

**RS-232 Control  
of the  
Advantage EQ281/8, EQ282/8  
and Advantage SMS200**

**advantage** 

## **Introduction**

This document contains technical information relating to computer control of the Biamp Advantage EQ282/8 dual channel digitally controlled graphic equalizer. The command set described in this manual also applies to the Advantage EQ281/8 (which is a single channel version of the EQ282/8) and the Advantage SMS200 Sound Masking System (which is an EQ281/8 with internal pink noise generator).

This information is intended for advanced users - in particular for those who wish to develop their own computer programs to control these equalizers. It is assumed that the reader is an experienced programmer and has some familiarity with standard programming practices, binary and hexadecimal numbers, the ASCII character set, asynchronous serial data communications, and RS-232 interfaces.

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

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

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

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

The Advantage EQ282/8 does not utilize standard ASCII-hex format. The EQ282/8 computer commands implement what I call "pseudo-hex" notation (also sometimes referred to as a "poor man's" ASCII-hex notation). Instead of representing the hexadecimal value using the ASCII characters '0' thru '9' and 'A', 'B', 'C', 'D', 'E', and 'F', the pseudo-hex format uses the ASCII characters '0' thru '9' and the characters ':', ';', '<', '=', '>', and '?'. As you can see by studying the ASCII code chart, a binary nibble may be converted to its equivalent pseudo-hex character by simply adding 0x30.

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

## **Data Communications Parameters**

The Advantage EQ282/8 has a rear-panel dip-switch which allows you to set its baud rate to either 2400, 9600, or 19200 bits per second. The factory default setting is 9600. The EQ282/8 communicates with 8 data bits, no parity, and 1 stop bit. The EQ282/8 utilizes a subset of the standard 7-bit ASCII character set.

The Advantage EQ282/8 utilizes interrupt-driven communications firmware with a 32-character receive buffer. When the EQ282/8 receives a command, it may become "busy" for as long as 250 milliseconds while it is executing the command. During this time, the EQ282/8 can successfully receive up to 32 additional characters. If more than 32 characters are sent to the EQ282/8 while it is busy executing a command, its input buffer will be overrun and some characters will be "lost".

To avoid overrunning the Advantage EQ282/8's input buffer, it is recommended that some form of flow control be implemented. Neither hardware (DTR or RTS) nor XON/XOFF (also known as DC1/DC3 or control-S/control-Q) handshaking is supported by the EQ282/8. However, each character which the EQ282/8 receives through its serial port will be "echoed" back to the computer. A simple form of flow control may be implemented by simply waiting for each character's echo from the EQ282/8 before transmitting the next character, since the EQ282/8 doesn't retrieve and echo an incoming character until it has finished processing the previous character.

## **Computer Control**

The Advantage EQ282/8 has an RS-232-compatible serial port which allows it to be controlled by a computer or by a system controller such as those provided by AMX or Crestron. The EQ282/8 offers the following two methods of computer control:

- **Control Button Emulation.** This method of computer control allows the computer to emulate Biamp's standard infrared remote control transmitter or wall-mount remote control panel. Using this method, the computer outputs ASCII characters which are equivalent to the characters which would be generated by a Biamp remote control. These ASCII characters are transmitted from the computer to the EQ282/8's serial port. In order to use Control Button Emulation with an EQ282/8, you must first use Biamp's Windows 95-based PC Control Software Package to establish "button" definitions for the device. Control Button Emulation is simple to perform, however, it only provides "one-way" control of the EQ282/8 - it allows the computer to send simple commands *to* the EQ282/8, but it does not provide any mechanism for requesting status information *from* the EQ282/8.

- Advanced Computer Control. This method of computer control provides advanced commands which allow "two-way" control of the Advantage EQ282/8. Using Advanced Computer Control commands, the computer may request status information *from* the device as well as send commands *to* the device. The computer communicates with the Advantage EQ282/8 using the Advantage EQ282/8's serial port.

## **Control Button Emulation**

Control Button Emulation is the simplest form of computer control of the Advantage EQ282/8. This method of operation allows the computer to emulate the operation of a standard Biamp remote control transmitter. This may seem a little strange since, unlike most of Biamp's other computer-controlled products, the EQ282/8 does not actually have a remote control port. However, the EQ282/8 still implements control button emulation for situations where the EQ282/8 is linked (via serial port) to other Biamp products which *do* have remote control capability.

For each button on a standard Biamp remote control, there is a corresponding ASCII character. In order to emulate a remote control button, the computer simply transmits the corresponding ASCII character to the Advantage EQ282/8's serial port. Each character received by the EQ282/8 will be echoed back to the computer.

The following table summarizes the ASCII character codes for Control Button Emulation corresponding to each of the 40 remote control buttons supported by the Advantage EQ282/8. These button codes are also summarized on the ASCII code chart provided at the end of this manual. The remote control buttons on the standard Biamp transmitter are numbered from left to right going from bottom to top with the lower left-hand button being button number 1. When the EQ282/8 is shipped from the factory, all remote control button definitions are set to "nop" (no operation) - in order for Control Button Emulation to be useful, you must first use Biamp's PC software to create some button definitions which actually do something (recall preset curves, for example).

### **Control Button Emulation ASCII Codes**

button 1	'B' (0x42)	button 15	'P' (0x50)	button 29	'^' (0x5E)
button 2	'C' (0x43)	button 16	'Q' (0x51)	button 30	'_' (0x5F)
button 3	'D' (0x44)	button 17	'R' (0x52)	button 31	'`' (0x60)
button 4	'E' (0x45)	button 18	'S' (0x53)	button 32	'b' (0x62)
button 5	'F' (0x46)	button 19	'T' (0x54)	button 33	'c' (0x63)
button 6	'G' (0x47)	button 20	'U' (0x55)	button 34	'd' (0x64)
button 7	'H' (0x48)	button 21	'V' (0x56)	button 35	'e' (0x65)
button 8	'I' (0x49)	button 22	'W' (0x57)	button 36	'f' (0x66)
button 9	'J' (0x4A)	button 23	'X' (0x58)	button 37	'g' (0x67)
button 10	'K' (0x4B)	button 24	'Y' (0x59)	button 38	'h' (0x68)
button 11	'L' (0x4C)	button 25	'Z' (0x5A)	button 39	'i' (0x69)
button 12	'M' (0x4D)	button 26	'[' (0x5B)	button 40	'j' (0x6A)
button 13	'N' (0x4E)	button 27	'\' (0x5C)		
button 14	'O' (0x4F)	button 28	']' (0x5D)		

## Device Select Prefix Characters

When using Advanced Computer Control, up to sixty-four Advantage EQ282/8s may be linked together and individually controlled by the computer (a rear-panel dip-switch is used to set a unique device number to each unit). When using Control Button Emulation, a limited subset of device addressing may be performed, which allows individual control of up to four Advantage EQ282/8s (with device numbers 1 thru 4). This is accomplished by transmitting a device select prefix code immediately prior to each control button ASCII character code. The device select prefix code is inspected by each device to determine whether or not the device should react to the button code which immediately follows. If a button code is not immediately preceded by a device select prefix character, then all Advantage EQ282/8s in the system will react to that button code. The following table summarizes the ASCII characters to use for selecting various devices. This information is also summarized in the ASCII code chart provided at the end of this manual.

## Device Select Prefix Codes

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

## Advanced Computer Control

The Advanced Computer Control command set includes commands which allow the Advantage EQ282/8 to return information about the system to the computer, unlike Control Button Emulation which is basically a one-way control mechanism. The following list summarizes the commands available using Advanced Computer Control, including the ASCII command character associated with each command:

- ) recall-preset                      recall a preset equalization curve
- \* get-status                         retrieve the status indicating which presets are currently selected
- / get-version                        retrieve the model information and firmware version date

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

Some of the commands cause the Advantage EQ282/8 to return information to the computer. For each string of information returned to the computer, the EQ282/8 terminates the string by transmitting the ASCII carriage return character (0x0D - represented in this document as ↵).

As mentioned earlier, the Advantage EQ282/8 will echo all characters it receives, regardless of whether or not the characters are valid commands or parameters. Characters greater than 0x7F are reserved and should not be transmitted by the computer. The EQ282/8 utilizes a subset of the standard ASCII character set. The following characters have meaning to the Advantage EQ282/8:

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

An ASCII code chart showing all Advantage EQ282/8 commands and codes is provided at the end of this document for your convenience. Please note that all ASCII *control*

characters (including space, tab, carriage return, and line feed) will be echoed by the Advantage EQ282/8 and then ignored - they have no special significance to the EQ282/8.

### **Device Type Bitmask and Device Number Parameters**

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

The Device Type Bitmask parameter byte for the Advantage EQ282/8 is always 0x80. This Device Type Bitmask is also shared with other Biamp devices, including the Advantage EQ281/8, Advantage SMS200, and Advantage DRI. It is anticipated that all new devices developed by Biamp will utilize 0x80 as their Device Type Bitmask. Other Device Type Bitmasks currently in use are:

- 0x01 Biamp Advantage DRC 4+4 digital remote control
- 0x02 Biamp Advantage EQ28X digitally-controlled graphic equalizer
- 0x04 Biamp Advantage SPM522D stereo preamp/mixer
- 0x08 Biamp Advantage PMX84 programmable matrix switch

The Device Number parameter byte provides support for up to sixty-four devices, numbered 0 thru 63 ('00' thru '3?' in pseudo-hex). The Advanced Computer Control command structure also supports a "broadcast" command, which allows you to send a command to all devices which match the Device Type Bitmask. To send a "broadcast" command, specify 255 ('??' in pseudo-hex) as the Device Number parameter.

Each Advantage EQ282/8's device number is assigned using rear-panel dip-switches. Except for broadcast commands, any particular EQ282/8 will respond to an Advanced Computer Control command only if bit 7 of the Device Type Bitmask is a '1' and the Device Number parameter matches its dip-switch settings.

## ) recall-preset

### Description:

The recall-preset command causes the Advantage EQ282/8 to recall a preset equalization curve for one or both channels. Each preset eq curve includes the high-pass filter corner frequency setting, the gain settings for all 28 1/3-octave filter bands, the low-pass filter corner frequency setting, and the signal level control setting. For the SMS200, each preset eq curve also includes the pink-noise level control setting.

### Syntax of Command:

*ab80dd)*

where:

- |           |   |   |
|-----------|---|---|
| <i>a</i>  | = | Preset for channel 2:<br>0 = NOP (no operation),<br>1 - 8 = recall preset 1 - 8,<br>> = recall prior channel 2 preset |
| <i>b</i>  | = | Preset for channel 1:<br>0 = NOP (no operation),<br>1 - 8 = recall preset 1 - 8,<br>> = recall prior channel 1 preset |
| 80        | = | Device Type Bitmask (pseudo-hex)  |
| <i>dd</i> | = | Device Number (pseudo-hex)  |
| )         | = | recall-preset command character (0x29)  |

### Syntax of Response:

(no response)

### Example:

**578001)**

This example causes Advantage EQ282/8 number 1 to recall preset eq curve 5 for channel 2 and recall preset eq curve 7 for channel 1.

### Comments:

For the Advantage EQ281/8 and SMS200 (which are both single-channel devices) always specify '0' (no operation) for channel 2.

The typical execution time for the recall-preset command is 125 milliseconds when recalling a preset for one channel and 250 milliseconds when recalling a preset for both channels.



**\* get-status**

Description:

The get-status command causes the Advantage EQ282/8 to return its model identifier and the preset number of the currently selected preset for each channel.

Syntax of Command:

0080*dd*\*

where:

00	=	command specifier
80	=	Device Type Bitmask (pseudo-hex)
<i>dd</i>	=	Device Number (pseudo-hex)
*	=	get-status command character (0x2A)

Syntax of Response:

*xxab*↵

where:

<i>xx</i>	=	Model Identifier: '0:' = EQ282/8, '0;' = EQ281/8, '0<' = SMS200
<i>a</i>	=	Currently selected preset for channel 2
<i>b</i>	=	Currently selected preset for channel 1

Example:

command:  
**008001\***

response:  
**0:57↵**

This example causes device number 1 to return its model identifier and preset status. In this example, the device identifies itself as an Advantage EQ282/8 with preset 5 selected for channel 2 and preset 7 selected for channel 1.

Comments:

## / **get-version**

### Description:

The get-version command causes the device to return its model identifier code and firmware version to the computer. The model identifier code is '0:' for the Advantage EQ282/8, '0;' for the EQ281/8, and '0<' for the SMS200. The firmware version number is simply the release date of the firmware, in the format of *mmddy*. These date values are decimal digits, not pseudo-hex notation. For example, December 31, 1997 would be represented as **123197**.

### Syntax of Command:

0080*dd*/

where:

00	=	'00' (command specifier)
80	=	Device Type Bitmask (pseudo-hex)
<i>dd</i>	=	Device Number Bitmask (pseudo-hex)
/	=	get-version command character (0x2F)

### Syntax of Response:

*xxmmddy*↵

where:

<i>xx</i>	=	Model Identifier
<i>mm</i>	=	2-digit decimal month number
<i>dd</i>	=	2-digit decimal day of the month
<i>yy</i>	=	2-digit decimal year number

### Example:

command:  
**008001/**

response:  
**0:070297↵**

This example causes device number 1 to return its model I.D. and firmware version. In this example, device number 1 is an Advantage EQ282/8 and its firmware version date is July 2, 1997.

# ASCII Code Chart

with Decimal & Hexadecimal Equivalents and Advantage EQ282/8 Commands

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

# HEXADECIMAL CONVERSION CHART

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