Chapter 5

Remote Control

Your CDQPrimas advanced remote control capabilities make controlling your codec from your desktop or from half-way around the world simple.

5. Remote Control Considerations

The powerful remote control features standard on all ***CDQPrima*** models make configuring and controlling your codec from an attached computer or terminal simple and easy. You can also configure and control a ***CDQPrima*** from anywhere in the world as easy as if the codec were next to you.

The ***CDQPrima*** provides remote control by way of two communications protocols. RS232 or RS485 communications protocols can be used when connected to the rear panel remote control port. The front panel remote control port on models 120, 220, and 230 provides an additional RS232 interface. The front and rear panel remote control ports do not need to be configured the same way and can be used simultaneously. The serial communication facility provides extensive control of the ***CDQPrima***.

The RS232 remote control port provides point-to-point communication between a ***CDQPrima*** and a host controller. This allows the host to communicate with the ***CDQPrima*** through a directly connected terminal or a dial-up modem connection. In addition, the **EasyCall *Prima***® remote keypad connects to this RS232 port. Any terminal or terminal emulation software can be used as long as VT100 emulation is possible. You do not need a PC compatible computer, terminal emulation packages are available for all major computer platforms, including Mac and Sun.

The RS485 port allows control of several ***CDQPrima*** units connected on a bus. There are two versions of RS485: two-wire and four-wire, both of which require addressing and proper terminations. The two-wire configuration allows all ***CDQPrima*** units to be interconnected with a single shielded twisted wire pair. The various RS485 configurations are discussed later in this chapter.

5.1 Communications PROTOCOL definition

All communication to the ***CDQPrima*** can be with or without a protocol. The protocol provides a level of protection and insures that the correct message is received even in the presence of errors on the communication channel. The protocol envelope is wrapped around the data message. The protocol envelope header consists of an SOH, length, unit address, block serial number (BSN) and a status byte. This header is followed by a message of length `len`. The message is followed by a checksum of the specified bytes and finally an EOT byte.

The BSN is the block serial number. The BSN byte is any number between 0 and 255 and is generated by the host. This byte is unchanged by the ***CDQPrima*** and returned to the master in the response. The host to associate the response with the command uses this byte. It is not used by the ***CDQPrima*** and may be set to 0 (or any number) by the host.

The unit address is used to address a particular unit. The unit address byte may be any number between 0 and 30. This number is used to address the proper ***CDQPrima*** unit. In the protocol mode, each ***CDQPrima*** is watching for any message with its unit address. The ***CDQPrima*** unit address is set by the **`CID`** command or the keypad sequence <Common><RPRmt Ctl><Set ID>.

The **CmdStat** is the command status. On messages to the ***CDQPrima***, the **CmdStat** byte is ignored. On messages from the ***CDQPrima***, this byte represents the status of the operation. If the **CmdStat** variable is non-zero in the ***CDQPrima*** response, then the **len** byte will always be 0 and there will never be any data since a non-zero value means an error. The status bytes are as follows:

0	=	command executed successfully
non-zero	=	Error in command execution (See next section)

The checksum is the CCITT-CRC16 16 bit Cyclic-Redundancy-Code checksum of the 8 bit characters in the message. The following shows the fields of a message using protocol:

SOH	1	
lenmsb	(msb of message length)	
lenlsb	(lsb of message length)	
unit address	0..255	*
BSN	0..255	*
CmdStat	not used on input, status of command from ***CDQPrima***	*
data 0		*
data 1		*
•		*
•		*
•		*
data len-1		*
checksum	msb of CCITT-16	
checksum	lsb of CCITT-16	

* - included in checksum calculation

Following is an example of a protocol message for the command **EBR ?** addressed to a ***CDQPrima*** encoder with a unit number of 0. All values are in hex:

01	SOH
00	length of message msb
05	length of message lsb
00	unit 0
00	BSN
00	CmdStat (unused by ***CDQPrima***)
45	ASCII E
42	ASCII B
52	ASCII R
20	ASCII blank
3F	ASCII ?
97	checksum
d8	checksum

The response from the ***CDQPrima*** is:

01	SOH
00	length of message msb
04	length of message lsb
00	unit 0

00	BSN
00	CmdStat (command successful)
31	ASCII 1
32	ASCII 2
38	ASCII 8
4B	ASCII K
6B	checksum msb
9A	checksum lsb

5.2 RS232 Considerations

Although, due to space constraints, the ***CDQPrima*** models do not use a standard RS232 connector, all other aspects of RS232 communications standards are implemented. RS232 has voltage levels symmetrical about 0 v (approximately +/- 10 volts) and is single-ended. RS232 is a point-to-point interface and can support data rates up to 100 kb/s. While RS232 is limited to 50 feet (15.2 meters), longer distances between terminal and ***CDQPrima*** can be achieved using RS232 line extenders. In addition, if modems are used, the ***CDQPrima*** and controlling terminal can be half a world apart, yielding true 'remote' control. Refer to Appendix A for the C-1800 cable used for rear panel RS232 remote control. Cable C-2000 is used for front panel RS232 remote control.

5.3 RS485 Considerations

RS485, on the other hand, uses balanced signals and voltages that are always positive (approximately 0 to 5 volts). The RS485 electrical interface is intended for communication over large distances (miles) at high data rates (megabits per second) While RS232 is strictly a point-to-point interface, RS485 can be used as a point-to-point interface as well as in multi-drop (point to multi-point) configurations. As in all point-to-multi-point architectures, terminating resistors are required at both ends of the bus. The factory default is set to have the RS485 termination 'ON'. Refer to Appendix A for the RS485 cable diagram

Fig. 5-1 shows two RS485 devices in a point-to-point connection. Notice that the transmit data port (TD) from one device is connected to the received data port (RD) of the other device.

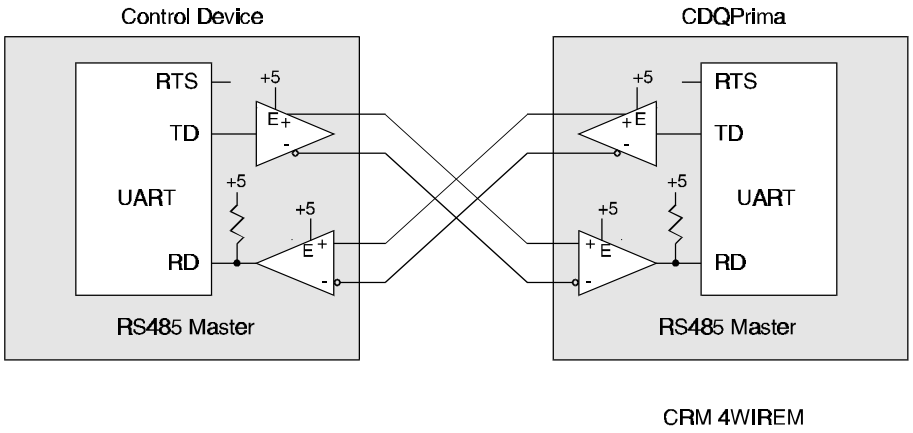

Figure 5-1 RS485 Master-to-Master interconnection

This configuration is called *FOUR WIRE MASTER* because there are four wires interconnecting each of the devices and each device has its transmitter and receiver permanently enabled. This is shown by a high (+5 volts) connected to the E (enable) lines. This mode of operation in the ***CDQPrima*** is enabled by the **`CRM 4WIREM`** command,

<Common><RPRmt Ctl><485 Mode><4WIREM>

from the keypad, or by entering a 'Direct Command' from the Windows control program.

The RTS signal is not used in the four wire master configuration. This signal is generated by the UART and goes high slightly before any data is transmitted and returns to low after the last character has been transmitted from the UART. See Fig. 5-2 for the timing of the RST signal relative the to TD signal. Note that the transmitted data (TD) may be high or low during the active period.

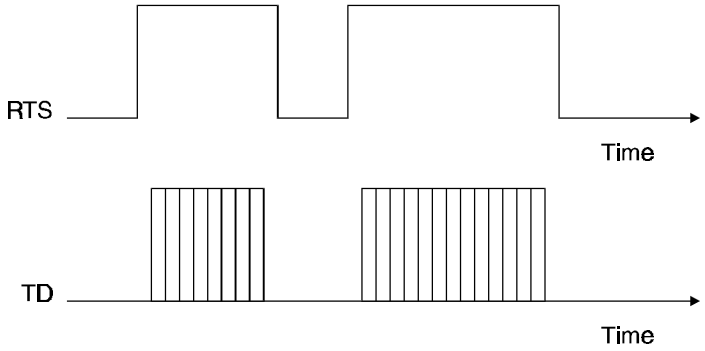

Figure 5-2 RTS/TD UART data timing

A four wire master device may be connected to a number of four wire slave devices, as shown in Fig. 5-3. In this case, the four wire slaves are always listening to the data transmitted by the four wire master controller. The transmitter (TD) of each slave is disconnected from the bus in normal operation and is enabled only when the four-wire slave has received a message to which it must respond. To respond, the slave must raise the RTS line high, transmit the response and then lower the RTS line. This master-to-slave communication assumes that each message from the master is tagged with an ID field so that only the proper slave may respond. This configuration is therefore an *Addressed* communications bus. If two slaves respond at the same time, the message received by the master would be garbled.

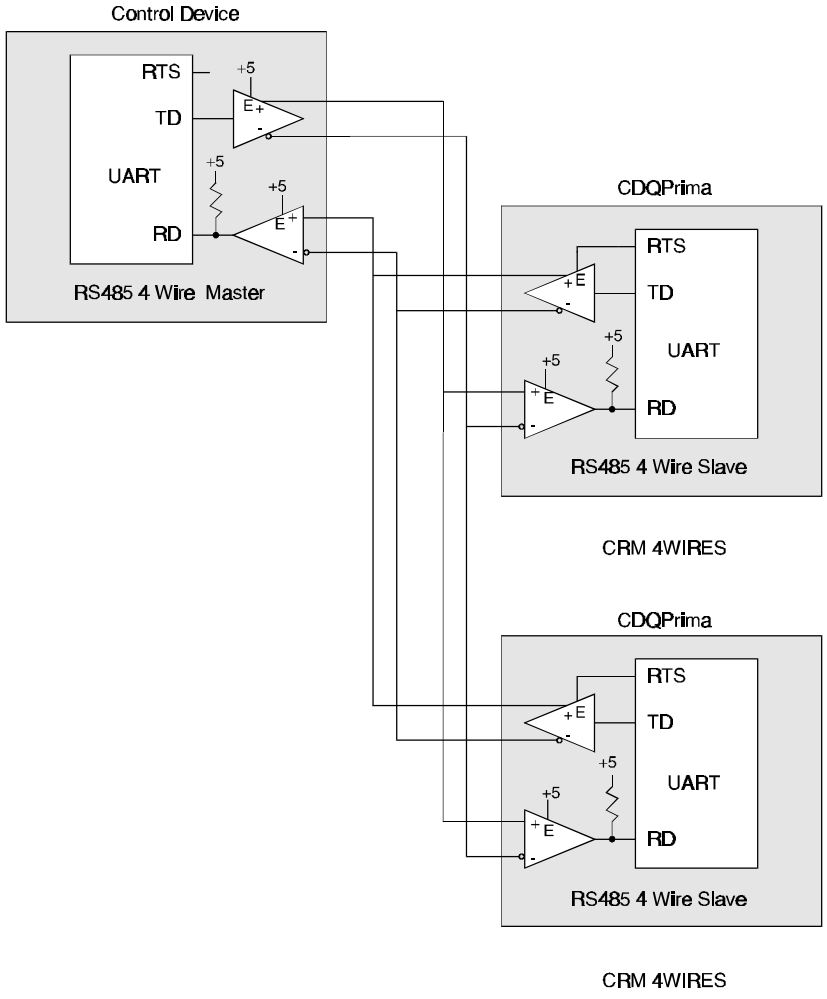

Figure 5-3 RS485 four wire master to four wire slaves

Fig. 5-4 shows a configuration in which only two wires are used for both transmission and reception. In this configuration all transmitters (TD) must be disconnected from the bus and all receivers are connected to the bus looking for a message with the correct ID. Whichever unit wants to access the bus must first enable its transmitter (TD), disable its receiver (RD) and transmit its message with the appropriate ID field, thus becoming a temporary' master. Once the message has been transmitted, it must disconnect its transmitter and enable its receiver. Notice the small circle above the E by the line receiver. This indicates that the RTS signal is inverted before connection to the line receiver. This circuit automatically forces the receiver to be off when the transmitter is on, and vice versa. This configuration is also an addressed communications bus.

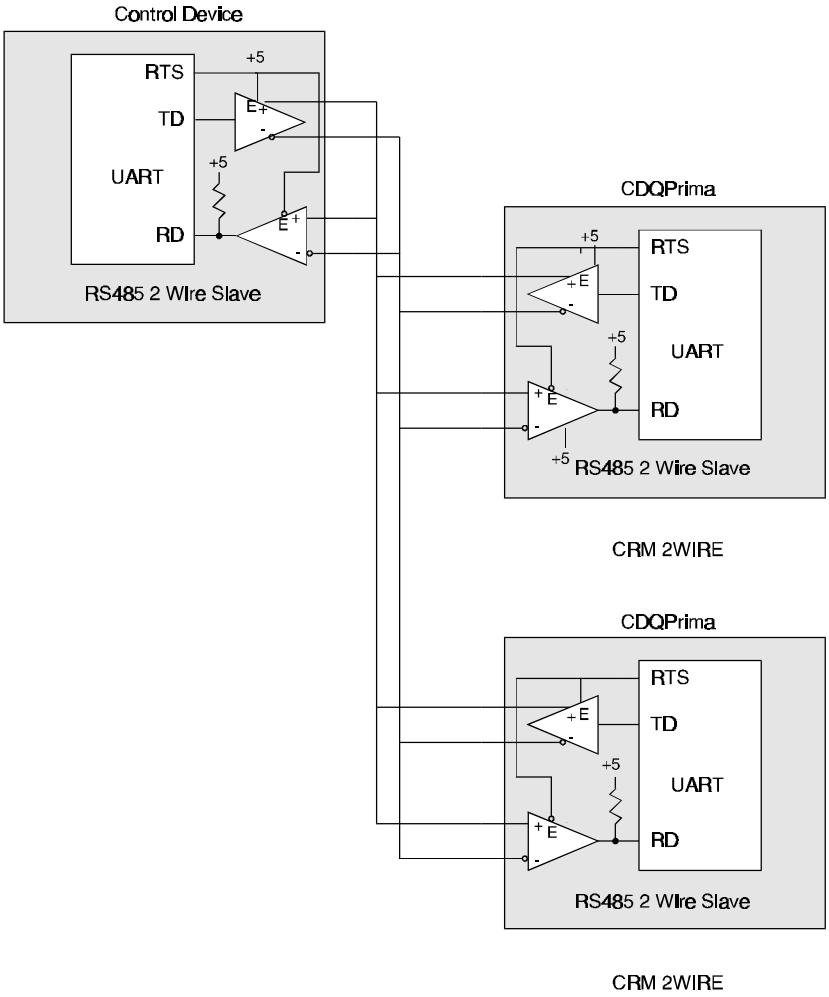

Figure 5-4 RS485 two wire slave interconnection

The following table summarizes the three possible ***CDQPrima*** RS485 configurations:

RS485 mode	Description
4 Wire Master	Transmitter and receiver are always enabled.
4 Wire Slave	Receiver always enabled. Transmitter active only when transmitting a message.
2 Wire Slave	Receiver enabled when transmitter not transmitting. Transmitter active only when transmitting a message.

Table 5-1 RS485 operational modes

Use the **CRI** command

<Common><RPRmt Ctl><Interface>

to set the rear port to RS485 operation, the **CRM** command

<Common><RPRmt Ctl><485 Mode>

to set the RS485 mode, the **CID** command

<Common><RPRmt Ctl><Set ID>

to set the RS485 ID (remember, this is an *addressed* bus), and the **CRB** command

<Common><RPRmt Ctl><Port baud>

to set the baud rate.

For the Windows control setup, most parameters of the RS485 bus are set at program load time, either automatically or via prompts.

There is one last issue with RS485 and that is termination. Since the RS485 lines are high speed, they must be properly terminated. RS485 requires 110 ohm twisted pair wire with 110-ohm termination at ***each end*** of the transmission line, a total of two terminations. There should not be any terminations in the middle of the transmission line. We have found, however, that if only a few ***CDQPrima***'s are connected and wires are short, you do not need to turn off any terminations. The ***CDQPrima*** comes from the factory with line terminations activated. To disable the RS485 terminations, there are 12 dip switches (2 switch blocks labeled DSW2 and DSW3, near the rear of the ***CDQPrima***). The ***CDQPrima*** must be opened to access these switches. The factory default position is ON, indicating active terminations. To de-activate the terminations, put all 12 dip switches to the OFF position. If your ***CDQPrima*** has been

!note:

equipped with the optional relay outputs, you will have to remove the relay circuit board to access the RS485 termination switches.

In the two-wire slave mode, no one is driving the transmission line. In this case, it might be advisable to pull the lines to a generally known state with pull-up and pull-down resistors. Fig. 5-5 shows a method of terminating an RS485 transmission line. It is important that the pull-up and pull-down resistors pull the signals to their natural state. There should not be any bus glitches when the enable (E) signals are activated. The ***CDQPrima*** has internal terminations and pull-ups/pull-downs for all RS485 ports. See Appendix B for details of the rear panel connections.

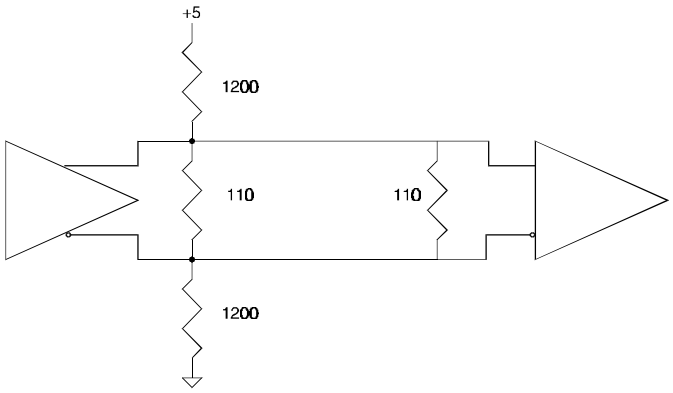

Figure 5-5 RS485 Termination and Pull-up/Pull down

5.4 Inband Control of Far End *CDQPrima*

The ***CDQPrima*** allows you to execute commands in the far end ***CDQPrima*** or ***Prima LT*** by an inband signaling technique. Far-end control is only possible if a terminal or Windows control is used. Far-end control is *not* possible from the keypad or when using the **EasyCall *Prima*** remote keypad, available from an outside vendor.

!note:

To enable inband communications, both near and far end units must have the following ancillary data configurations enabled:

- **MUX mode must be enabled**. Use the **`CAN MUX`** command or the keypress sequence

  <Common><Anc Data><Asy Adata><Mux Mode>

- Use any ISO/MPEG algorithm, MPEGL2, MPEGL3, CCSO or CCSN (see the **EAL** command). Far-end remote control is not possible if using the G.722 algorithm.
- If you are using MPEG Layer III (MPEGL3), you must set the encoder and decoder ancillary data format to CCS using the **EAH**

  <Common><Anc Data><Fmt/L3> <Enc Fmt>

  and **DAH**

  <Common><Anc Data><Fmt/L3><Dec Fmt>

  commands. Far end remote control will not work using Layer III with the CCS Protect mode. This mode is for using ancillary data with a Telos Zephyr codec.
- Once these conditions are met *and the units have both framed*, it is then possible to execute commands in a far end unit.

5.4.1 Far End *CDQPrima* Control Using a Terminal

To execute a command on the local ***CDQPrima***, you simply type the command with its required parameters, if any. To execute the same command on the far-end ***CDQPrima*** or ***Prima LT***, precede the command with the 'greater-than' sign (>). This tells the near end command processor that the command is to be sent to the far end unit.

For example, if you want to find out the command processor software version of the near end unit, type

MVN CP

To execute the same command in the far end unit and transmit the response back to the near end ***CDQPrima***, type:

>MVN CP

Notice the addition of the > before the command. The ***CDQPrimas*** unique ability to send inband commands to the far end unit opens many possibilities for programming, speed dial table maintenance, troubleshooting and control. For example, a remote operator can switch the audio source of the far-end encoder, or can even put the far-end codec into loopback. The operator *cannot*, however, get the studio unit out of loopback remotely, see the warning below:

***

!note:

Far end remote control is a very powerful feature available on all *CDQPrima* models. However, it is also a very *dangerous* feature. Far end remote control is only possible if the two *CDQPrimas* are framed to each other. It is possible to execute commands on the far end *CDQPrima* that would cause permanent frame loss, and therefore loss of far end remote control. The only way to recover is to get the local *CDQPrima* into the same configuration as the far end unit, if possible, and re-establish frame.

***

Another form of inband control of a far end ***CDQPrima*** is to use virtual actions and simulated switches to send commands and actions to another ***CDQPrima***. Virtual actions are pre-defined instructions that can be executed automatically when pre-arranged conditions are met. Virtual actions will be explained in the Prima Logic Language chapter.

5.5 Windows Remote Control Program

By far, the easiest method of controlling up to 31 attached ***CDQPrima*** codecs and all connected far-end ***CDQPrima*** codecs is with the optional Windows Remote Control program. All ***CDQPrima*** functions can be accessed using simple, intuitive point-and-click operations or direct commands. The Windows Remote Control program works with any PC compatible computer running Microsoft Windows 3.0 or later. Sorry, no Mac version is available, but it will run using most Windows emulation software.

The Windows program is self-installing, and its use is intuitive. No specific instruction manual is needed, and extensive, context sensitive on-line help is included. Screen examples can be found throughout this manual and in the ***CDQPrima*** Users Guide. There are also several control and convenience features that are available only when using the Windows Remote Control program, such as:

- A VCR-like timer with multiple events for automatic, unattended operation
- Hot-Key functions available in all ***CDQPrima*** models
- Simple point-and-click Prima Logic Language programming

- Simple storage and downloading of the Speed Dial and Quick Configuration directory
- Storage and recall Prima Logic Language programs and Virtual Actions

5.5.1 Far End *CDQPrima* Control Using Windows® Control

Once framed communications have been established with the far end ***CDQPrima***, you can click on the Extra' menu pull-down, and then click on Far end Prima test'. This will open up a dialog box, and by clicking on 'TEST', remote control communications with the far end prima are established. The Windows control screen would now look like:

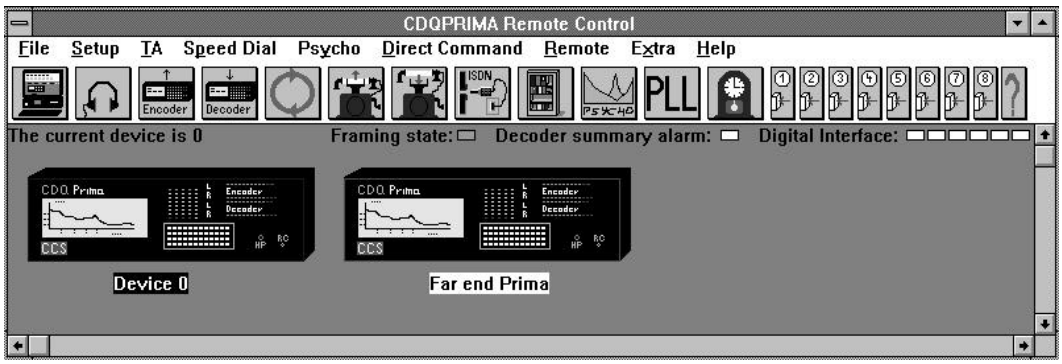

Figure 5-6 Windows screen with far end control activated

Any command can now be executed locally or sent to the far end by simply double-clicking on the appropriate picture before clicking on the desired command or icon. Please remember, however, that far end remote control is only possible as long as all conditions above have been met and the two units are framed to each other.

5.6 Remote Control Error Summary

Following is a list of ***CDQPrima*** error codes. Fatal errors requiring factory assistance are indicated with an asterisk (*):

1	Command unknown	9	Selected line has no interface assigned
2	Command missing 1 or more arguments	10	No lines have an interface selected
3*	Internal error	11	Cannot delete last line - RED or COM using it
4	Unknown argument		
5	No value to read	12	Cannot delete a line that is in use
6*	Could not talk to decoder DSP	13	Argument too long
7	One or more command arguments are bad	14	No response from far end
		15*	Flash ROM byte write failure
8	Too many arguments for command	16*	Flash page erase failure

17*	Flash address test failure	49*	Could not talk to encoder DSP
18*	Flash data test failure	50*	Hardware test failure
19	Down load record invalid	51	Could not talk to VU DSP
20*	Trouble writing to BBM	52	No response from far end ***CDQPrima***
21*	Checksum of flash disagrees with checksum	53	Not allowed in H.221 mode
		54	Decoder must not be dependent
22*	Thing failed to boot	55	SD description used
23	Speed dial directory full	56	SD list empty
24	FFT failure	57	Not enough phone numbers
25	Headphone muted	58	Too many phone numbers
26	Speed dial in use by front panel user	59	Bad bit rate for speed dial
27	Security level not sufficient for execution of the requested command	60	SD description not found
		61	Paren mismatch in expression
28	Password mismatch	62	Bad phone number
29	Line/bitrate mismatch	63*	Bad flash address
30	Not enough TA's	64	Ancillary data mode not in CAN MUX
31	Could not set parm	65	Decoder must not be independent
32*	No heap memory	66*	Could not talk to R-S DSP
33	SD ID not found	67	Bad psycho table number
34	DIF not defined	68	Decoder did not frame in required time
35	Connection failed	69	AES/EBU input not allowed in this mode
36	Bad bit rate		
37	Not enough empty TA lines	70	Must be in mono
38	DIF not a TA	71	DIF not an internal TA
39	Command not available	72	No response from TA
40	Unknown identifier	73	DIF not an external TA
41	Too many OR terms	74	Not enough digits in SPID
42	Expected identifier not found	75	Unknown TA type
43	Expected operator not found	76	TA config directory full
44	Event or action program too long	77	TA config list empty
45*	Trouble loading PA parms from flash	78	TA description used
46*	Trouble storing PA parms in flash	79	Not work in loopback
47	Digital I/F not a TA	80	Action completed
48	Command does not accept a ?	81	Action not taken

The ***CDQPrima*** has nearly 200 remote control commands available. References have been made to most available remote control commands throughout this document and the ***CDQPrima*** Users Guide. Command syntax help is available on-line to all remote control users by simply entering the command:

HELP ?

for a complete categorized listing. Once the category number is known, enter the command:

HELP *number*

for a listing of all commands within the category.

A complete listing of all commands is available in the ***CDQPrima*** Remote Control Manual, available from MUSICAM USA.