

# Protocol 3000

Kramer devices can be operated using Kramer Protocol 3000 commands sent via serial or Ethernet ports.

---

## Understanding Protocol 3000

Protocol 3000 commands are a sequence of ASCII letters, structured according to the following.

- **Command format:**

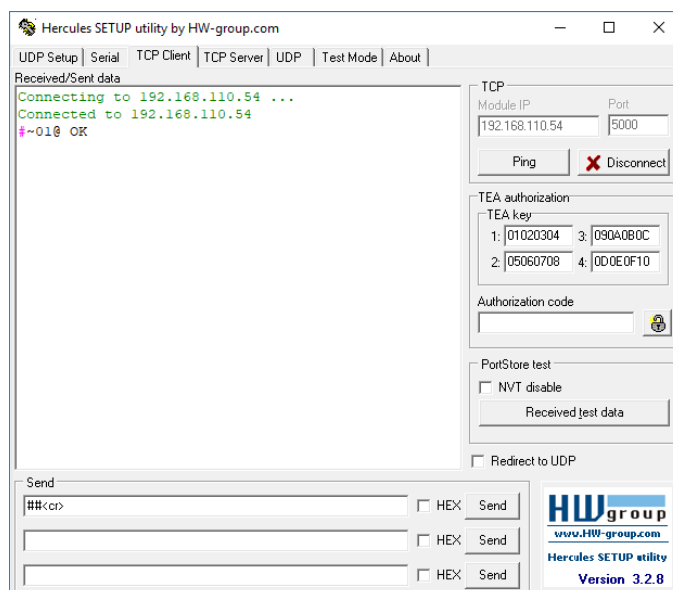
Prefix	Command Name	Constant (Space)	Parameter(s)	Suffix
#	Command	_	Parameter	<CR>

- **Feedback format:**



Prefix	Device ID	Constant	Command Name	Parameter(s)	Suffix
~	nn	@	Command	Parameter	<CR><LF>

- **Command parameters** – Multiple parameters must be separated by a comma (,). In addition, multiple parameters can be grouped as a single parameter using brackets ([ and ]).
- **Command chain separator character** – Multiple commands can be chained in the same string. Each command is delimited by a pipe character (|).
- **Parameters attributes** – Parameters may contain multiple attributes. Attributes are indicated with pointy brackets (<...>) and must be separated by a period (.).




The command framing varies according to how you interface with the **VS-88UT**. The following figure displays how the # command is framed using terminal communication software (such as Hercules):



# Protocol 3000 Commands

Function	Description	Syntax	Parameters/Attributes	Example												
#	Protocol handshaking.   Validates the Protocol 3000 connection and gets the machine number.	<b>COMMAND</b> #<CR> <b>FEEDBACK</b> ~nn@_ok<CR><LF>		#<CR>												
LOAD	Load file to device.	<b>COMMAND</b> #LOAD_file_name,size<CR> <b>FEEDBACK</b> Data sending negotiation: * Device - ~01@LOAD_file_name,size_ready<CR><LF> * End User (+Device)- Send file in Protocol Packets * Device - ~01@LOAD_file_name,size_ok<CR><LF>	<b>file_name</b> – Name of file to save on device <b>size</b> – Size of file data that is sent <b>Using the Packet Protocol</b> Send a command: LDRV, LOAD, IROUT, LDEDID Receive Ready or ERR### If Ready: a. Send a packet, b. Receive OK on the last packet, c. Receive OK for the command Packet structure: <b>Packet ID</b> (1, 2, 3...) (2 bytes in length) <b>Length</b> (data length + 2 for CRC) – (2 bytes in length) <b>Data</b> (data length -2 bytes) <b>CRC</b> – 2 bytes <table border="1" data-bbox="981 645 1337 689"> <tr> <td>01</td> <td>02</td> <td>03</td> <td>04</td> <td>05</td> <td></td> </tr> <tr> <td>Packet ID</td> <td>Length</td> <td>Data</td> <td>CRC</td> <td></td> <td></td> </tr> </table> Response: ~nnnn_ok<CR><LF> (Where <b>NNNN</b> is the received packet ID in ASCII hex digits.)	01	02	03	04	05		Packet ID	Length	Data	CRC			Load the file_response.dat file to the device: #LOAD_file_response.dat,5360<CR>
01	02	03	04	05												
Packet ID	Length	Data	CRC													
EXT-CFG-FILE?	Get configuration file status.	<b>COMMAND</b> #EXT-CFG-FILE?<CR> <b>FEEDBACK</b> ~nn@EXT-CFG-FILE_status<CR><LF>	<b>status</b> – 1 – configuration.json is ready 2 – configuration.json is not ready	Get the configuration file status: #EXT-CFG-FILE?_<CR>												
EXT-PSU?	Get the status of the power supply unit.	<b>COMMAND</b> #EXT-PSU?_ps_id<CR> <b>FEEDBACK</b> ~nn@EXT-PSU_ps_id,plug_status,ps_status<CR><LF>	<b>ps_id</b> – Power supply id number: 1-2 * – both PSUs <b>plug_status</b> – Physical connection status of the power supply: ON – Connected OFF – Disconnected <b>ps_status</b> – Power status of the power supply: OK – powered properly WARN – not powered properly and may need to be checked OFFLINE – not receiving power	Get the status of power supply 1: #EXT-PSU?_1<CR>												
EXT-PSU-BUZZER	Enable/disable PSU warning buzzer.	<b>COMMAND</b> #EXT-PSU-BUZZER_enabled_status<CR> <b>FEEDBACK</b> ~nn@EXT-PSU-BUZZER_enabled_status<CR><LF>	<b>enabled_status</b> – 0 – Disable 1 – Enable	Enable the PSU warning buzzer: #EXT-PSU-BUZZER?_1<CR>												
EXT-PSU-BUZZER?	Get the enable/disable status of the PSU warning buzzer.	<b>COMMAND</b> #EXT-PSU-BUZZER?_<CR> <b>FEEDBACK</b> ~nn@EXT-PSU-BUZZER?_enabled_status<CR><LF>	<b>enabled_status</b> – 0 – Disabled 1 – Enabled	Get the enable/disable status of the PSU warning buzzer: #EXT-PSU-BUZZER?_<CR>												
EXT-VOLT?	Get the voltage of the board.	<b>COMMAND</b> #EXT-VOLT?_slot_id<CR> <b>FEEDBACK</b> ~nn@EXT-VOLT?_slot_id:volt_checkpoint_id/volt_checkpoint_total : stdand value : actual value : status<CR><LF>	<b>slot_id</b> – Module ID (slot number): 1-17 <b>volt_checkpoint_id</b> – ID number of current voltage detection point, starting from 1 <b>volt_checkpoint_total</b> – Nnumber of voltage detection points <b>stdand value</b> – Hardware design value <b>actual value</b> – Current value <b>status</b> – WARN OK	Get the voltage for the card in slot 1: #EXT-VOLT?_slot_id<CR>												
EXT-POE	Enable/disable the +48V input for the PS-4812 PoE power supply add-on.   Only the input for the PS-4812 PoE power supply add-on can be enabled/disabled. The module card ports do not.	<b>COMMAND</b> #EXT-POE_0,enable_value<CR> <b>FEEDBACK</b> ~nn@EXT-POE_0,enable_value,48V_status<CR><LF>	<b>port_id</b> – 0 – 48V input connector <b>enable_value</b> – ON – 48V input is enabled OFF – 48V input is disabled <b>48v_status</b> – Operational status of 48V input: OK – Receiving power WARN – Not working POWERDOWN – Enabled, but receiving power	Enable the 48V input for the PS-4812 PoE power supply add-on: #EXT-POE_0,ON<CR>												
EXT-POE?	Get the PoE power operational status for one of the module card ports or for the +48V input for the PS-4812 PoE power supply add-on.	<b>COMMAND</b> #EXT-POE?_port_id<CR> <b>FEEDBACK</b> ~nn@EXT-POE?_0,enable_value,48V_status<CR><LF>	<b>port_id</b> – 0 – 48V connector 1-34 <b>enable_value</b> – ON – 48V input is enabled OFF – 48V input is disabled <b>48v_status</b> – Operational status of 48V input: OK – Receiving power WARN – Not working POWERDOWN – Disabled, but receiving power	Get the enable/disable status of the 48V input for port 2: #EXT-POE?2_<CR>												

Function	Description	Syntax	Parameters/Attributes	Example
<b>AUD</b>	Set audio switch state.  ⓘ When AFV switching mode is active, this command also switches video and unit replies with command -AV.	<b>COMMAND</b> #AUD_in>out_id,in>out_id,...<CR> <b>FEEDBACK</b> ~nn@AUD_in>out_id<CR><LF> ~nn@AUD_in>out_id<CR><LF>	<b>in</b> – Input number or '0' to disconnect output > – Connection character between in and out parameters <b>out_id</b> – Output number 1–34 * for all outputs	Switch embedded audio HDMI™ IN 1 to HDMI OUT 3: #AUD_1>3<CR>
<b>AUD?</b>	Get audio switch state.  ⓘ When AFV switching mode is active, this command also switches video and unit replies with command -AV.	<b>COMMAND</b> #AUD?_out_id<CR> #AUD?_*<CR> <b>FEEDBACK</b> ~nn@AUD_in>out_id,in>out_id,...<CR><LF> or ~nn@AUD_*>out<CR><LF>	<b>in</b> – Input number or '0' to disconnect output > – Connection character between in and out parameters <b>out_id</b> – Output number 1–34 * for all outputs	Get audio switch state for output 1: #AUD?_1<CR>
<b>AUD-LVL</b>	Set volume level.	<b>COMMAND</b> #AUD-LVL_io_mode,io_index,vol_level<CR> <b>FEEDBACK</b> ~nn@AUD-LVL_io_mode,io_index,vol_level<CR><LF>	<b>io_mode</b> – Input/Output 0 – Input 1 – Output <b>io_index</b> – Number that indicates the specific input or output port: 1–34 <b>vol_level</b> – Volume level 0–70	Set AUDIO OUT 2 volume level to 50: #AUD-LVL_1,1,50<CR>
<b>AUD-LVL?</b>	Get volume level.	<b>COMMAND</b> #AUD-LVL?_io_mode,io_index<CR> <b>FEEDBACK</b> ~nn@AUD-LVL_io_mode,io_index,vol_level<CR><LF>	<b>io_mode</b> – Input/Output 0 – Input 1 – Output <b>io_index</b> – Number that indicates the specific input or output port: 1–34 <b>vol_level</b> – Volume level 0–70	Get AUDIO OUT 1 volume level #AUD-LVL?_1,1<CR>
<b>AV</b>	Switch audio and video.	<b>COMMAND</b> #AV_in>out_id,in>out_id,...<CR> <b>FEEDBACK</b> ~nn@AV_in>out_id,in>out_id,...<CR><LF>	<b>in</b> – Number that indicates the specific input: 1–34 0 – disconnect output > – Connection character between in and out parameters <b>out_id</b> – Output number 1–34 * for all outputs	Switch IN 1 to OUT 4: #AV_1>4<CR>
<b>BALANCE</b>	Set balance level.	<b>COMMAND</b> #BALANCE_out_index,balance_level<CR> <b>FEEDBACK</b> ~nn@BALANCE_out_index,balance_level<CR><LF>	<b>out_index</b> – Number that indicates the specific output: 1–34 <b>balance_level</b> – Balance level 0–100 ++ increase current value – decrease current value	Set the speaker output balance to 50: #BALANCE_1,50<CR>
<b>BALANCE?</b>	Get balance level.	<b>COMMAND</b> #BALANCE?_out_index<CR> <b>FEEDBACK</b> ~nn@BALANCE_out_index,balance_level<CR><LF>	<b>out_index</b> – Number that indicates the specific output: 1–34 <b>balance_level</b> – Balance level 0–100	Get balance level for Output 1: #BALANCE?_1<CR>
<b>BASS</b>	Set audio bass level.	<b>COMMAND</b> #BASS_io_index,bass_level<CR> <b>FEEDBACK</b> ~nn@BASS_io_index,bass_level<CR><LF>	<b>io_index</b> – Number that indicates the specific input or output port: 1–34 <b>bass_level</b> – Bass level 0–15 ++ increase current value – decrease current value	Set audio bass level of channel 1 to 5: #BASS_1,5<CR>
<b>BASS?</b>	Get audio bass level.	<b>COMMAND</b> #BASS?_io_index<CR> <b>FEEDBACK</b> ~nn@BASS_io_index,bass_level<CR><LF>	<b>io_index</b> – Number that indicates the specific input or output port: 1–34 <b>bass_level</b> – Bass level 0–15	Get audio bass level of channel 1: #BASS?_1<CR>
<b>BAUD</b>	Set protocol serial port baud rate.  ⓘ The new defined baud rate is stored in the EEPROM and used when powering up.  Default baud rate is 115200 (on factory reset).  Only works with devices supporting this command.	<b>COMMAND</b> #BAUD_baud_rate<CR> <b>FEEDBACK</b> ~nn@BAUD_baud_rate<CR><LF> Option 1: ~nn@BAUD_current_baud_rate<CR><LF>	<b>baud_rate</b> – 9600, 19200, 38400, 57600, 115200 <b>current_baud_rate</b> – 9600, 19200, 38400, 57600, 115200, else - current protocol serial port baud rate <b>baud_param</b> – 0 - get the list of supported baud rates <b>baud_rate1,baud_rate2, ...</b> – List of supported baud rates	Set the baud rate to 9600: #BAUD_9600<CR>

Function	Description	Syntax	Parameters/Attributes	Example
<b>BAUD?</b>	<p>Get protocol serial port baud rate. (Option 1 - for current baud rate. Option 2 - for list of supported baud rates).</p> <p> The new defined baud rate is stored in the EEPROM and used when powering up.</p> <p>Default baud rate is 115200 (on factory reset).</p> <p>Only works with devices supporting this command.</p>	<p><b>COMMAND</b></p> <p><b>#BAUD?_&lt;CR&gt;</b></p> <p><b>#BAUD?_baud_param&lt;CR&gt;</b></p> <p><b>FEEDBACK</b></p> <p><b>~nn@BAUD_baud_rate&lt;CR&gt;&lt;LF&gt;</b></p> <p>Option 1: <b>~nn@BAUD_current_baud_rate&lt;CR&gt;&lt;LF&gt;</b></p>	<p><b>baud_rate</b> – 9600, 19200, 38400, 57600, 115200</p> <p><b>current_baud_rate</b> – 9600, 19200, 38400, 57600, 115200, else - current protocol serial port baud rate</p> <p><b>baud_param</b> – 0 - get the list of supported baud rates</p> <p><b>baud_rate1, – Baud_rate2, ...</b> - list of supported baud rates</p>	<p>Get protocol serial port baud rate: <b>#BAUD?_&lt;CR&gt;</b></p>
<b>BRIGHTNESS</b>	<p>Set image brightness per output.</p> <p> Value limits can vary for different module cards.</p> <p>Value is a property of input connected to current output. Changing input source might cause changes in this value (refer device definitions).</p> <p>In devices that enable showing multiple outputs on one display – each in a separate window – this command relates only to the window associated with the output indicated in the out-index parameter.</p>	<p><b>COMMAND</b></p> <p><b>#BRIGHTNESS_out_index,value&lt;CR&gt;</b></p> <p><b>FEEDBACK</b></p> <p><b>~nn@BRIGHTNESS_out_index,value&lt;CR&gt;&lt;LF&gt;</b></p>	<p><b>out_index</b> – Number that indicates the specific output: 1–34</p> <p><b>value</b> – Brightness value: 1–63</p>	<p>Set brightness for output 1 to 50: <b>#BRIGHTNESS_1,50&lt;CR&gt;</b></p>
<b>BRIGHTNESS?</b>	<p>Get image brightness per output.</p> <p> Value limits can vary for different devices.</p> <p>Value is a property of input connected to current output. Changing input source might cause changes in this value (refer device definitions).</p> <p>In devices that enable showing multiple outputs on one display – each in a separate window – this command relates only to the window associated with the output indicated in the out-index parameter.</p>	<p><b>COMMAND</b></p> <p><b>#BRIGHTNESS?_out_index&lt;CR&gt;</b></p> <p><b>FEEDBACK</b></p> <p><b>~nn@BRIGHTNESS_out_index,value&lt;CR&gt;&lt;LF&gt;</b></p>	<p><b>out_index</b> – Number that indicates the specific output: 1–34</p> <p><b>value</b> – Brightness value</p>	<p>Get brightness for output 1: <b>#BRIGHTNESS?_1&lt;CR&gt;</b></p>
<b>BUILD-DATE?</b>	<p>Get device build date.</p>	<p><b>COMMAND</b></p> <p><b>#BUILD-DATE?_&lt;CR&gt;</b></p> <p><b>FEEDBACK</b></p> <p><b>~nn@BUILD-DATE_date,time&lt;CR&gt;&lt;LF&gt;</b></p>	<p><b>date</b> – Format: YYYY/MM/DD where YYYY = Year MM = Month DD = Day</p> <p><b>time</b> – Format: hh:mm:ss where hh = hours mm = minutes ss = seconds</p>	<p>Get the device build date: <b>#BUILD-DATE?&lt;CR&gt;</b></p>

Function	Description	Syntax	Parameters/Attributes	Example
<b>CONTRAST</b>	<p>Set image contrast per output.</p> <p><b>i</b> Value limits can vary for different devices.</p> <p>Value is a property of input connected to current output. Changing the input source might cause changes in this value (refer to device definitions).</p> <p>In devices that enable showing multiple outputs on one display – each in a separate window – this command relates only to the window associated with the output indicated in the out-index parameter.</p>	<p><b>COMMAND</b></p> <pre>#CONTRAST_out_index,value&lt;CR&gt;</pre> <p><b>FEEDBACK</b></p> <pre>~nn@CONTRAST_out_index,value&lt;CR&gt;&lt;LF&gt;</pre>	<p><b>out_index</b> – Number that indicates the specific output: 1–34</p> <p><b>value</b> – Contrast value: 1–63</p>	<p>Set contrast for Output 1 to 40:</p> <pre>#CONTRAST_1,40&lt;CR&gt;</pre>
<b>CONTRAST?</b>	<p>Get image contrast per output.</p> <p><b>i</b> Value limits can vary for different devices.</p> <p>Value is a property of input connected to current window. Changing the window input source might cause changes in this value (refer to device definitions).</p> <p>In devices that enable showing multiple outputs on one display – each in a separate window – this command relates only to the window associated with the output indicated in the out-index parameter.</p>	<p><b>COMMAND</b></p> <pre>#CONTRAST?_out_index&lt;CR&gt;</pre> <p><b>FEEDBACK</b></p> <pre>~nn@CONTRAST_out_index,value&lt;CR&gt;&lt;LF&gt;</pre>	<p><b>out_index</b> – Number that indicates the specific output: 1–34</p> <p><b>value</b> – Contrast value</p>	<p>Get contrast for Output 1:</p> <pre>#CONTRAST?_1&lt;CR&gt;</pre>
<b>CPEDID</b>	<p>Copy EDID data from the output to the input EEPROM.</p> <p><b>i</b> Destination bitmap size depends on device properties (for 64 inputs it is a 64-bit word).</p> <p>Example: bitmap 0x0013 means inputs 1,2 and 5 are loaded with the new EDID.</p> <p>In certain products Safe_mode is an optional parameter. See the HELP command for its availability.</p>	<p><b>COMMAND</b></p> <pre>#CPEDID_edid_io,src_id,edid_io,dest_bitmap&lt;CR&gt;</pre> <p>or</p> <pre>#CPEDID_edid_io,src_id,edid_io,dest_bitmap,safe_mode&lt;CR&gt;</pre> <p><b>FEEDBACK</b></p> <pre>~nn@CPEDID_edid_io,src_id,edid_io,dest_bitmap&lt;CR&gt;&lt;LF&gt;</pre> <pre>~nn@CPEDID_edid_io,src_id,edid_io,dest_bitmap,safe_mode&lt;CR&gt;&lt;LF&gt;</pre>	<p><b>edid_io</b> – EDID source type (usually output) 0 – Input 1 – Output 2 – Default EDID</p> <p><b>src_id</b> – Port ID number of chosen input or output source 1–34 0 – Default EDID source</p> <p><b>edid_io</b> – EDID destination type (always input) 0 – Input</p> <p><b>dest_bitmap</b> – Bitmap representing destination IDs. Format: XXXX..X, where X is hex digit. The binary form of every hex digit represents corresponding destinations. 0 – indicates that EDID data is not copied to this destination. 1 – indicates that EDID data is copied to this destination.</p> <p><b>safe_mode</b> – Safe mode 0 – device accepts the EDID as is without trying to adjust 1 – device tries to adjust the EDID (default value if no parameter is sent)</p>	<p>Copy the EDID data from the Output 1 (EDID source) to the Input:</p> <pre>#CPEDID_1,1,0,0x1&lt;CR&gt;</pre> <p>Copy the EDID data from the default EDID source to the Input:</p> <pre>#CPEDID_2,0,0,0x1&lt;CR&gt;</pre>
<b>DETAIL-TIMING?</b>	<p>Get detail timing parameters.</p>	<p><b>COMMAND</b></p> <pre>#DETAIL-TIMING?_param,in_index&lt;CR&gt;</pre> <p><b>FEEDBACK</b></p> <pre>~nn@DETAIL-TIMING_param,in_index,value&lt;CR&gt;&lt;LF&gt;</pre>	<p><b>param</b> – Detail Timing 2 – H-De-Total 5 – V-De-Total</p> <p><b>in_index</b> – Number that indicates the specific input: 1–34</p> <p><b>value</b> – Video parameter in Kramer units, minus sign precedes negative values</p>	<p>Get detail timing parameters:</p> <pre>#@DETAIL-TIMING?_2,1&lt;CR&gt;</pre>
<b>DISPLAY?</b>	<p>Get output HPD status.</p> <p><b>i</b> After execution, response is sent to the com port from which the command was received. Response is sent after every change in output HPD status ON to OFF. Response is sent after every change in output HDP status OFF to ON.</p>	<p><b>COMMAND</b></p> <pre>#DISPLAY?_out_index&lt;CR&gt;</pre> <p><b>FEEDBACK</b></p> <pre>~nn@DISPLAY_out_index,status&lt;CR&gt;&lt;LF&gt;</pre>	<p><b>out_index</b> – Number that indicates the specific output: 1–34</p> <p><b>status</b> – HPD status according to signal validation 0 – HPD ON to OFF, 1 – HPD OFF to ON, 2 - EDID ready</p>	<p>Get the output HPD status of Output 1:</p> <pre>#DISPLAY?_1&lt;CR&gt;</pre>

Function	Description	Syntax	Parameters/Attributes	Example
EQ-LVL	Set equalization level.	<b>COMMAND</b> #EQ-LVL <u>io_mode</u> , <u>io_index</u> , <u>eq_type</u> , <u>eq_level</u> <CR> <b>FEEDBACK</b> ~nn@EQ-LVL <u>io_mode</u> , <u>io_index</u> , <u>eq_type</u> , <u>eq_level</u> <CR> <LF>	<b>io_mode</b> – Input/Output 0 – Input 1 – Output <b>io_index</b> – Number that indicates the specific input or output port: 1–34 <b>eq_type</b> – Equalizer Types 0 – Bass 2 – Treble <b>eq_level</b> – Equalizer level: 0–15	Set Bass EQ level of the speaker output to 12: #EQ-LVL <u>1</u> , <u>1</u> , <u>0</u> , <u>12</u> <CR>
EQ-LVL?	Get equalization level.	<b>COMMAND</b> #EQ-LVL? <u>io_mode</u> , <u>io_index</u> , <u>eq_type</u> <CR> <b>FEEDBACK</b> ~nn@EQ-LVL <u>io_mode</u> , <u>io_index</u> , <u>eq_type</u> , <u>eq_level</u> <CR> <LF>	<b>io_mode</b> – Input/Output 0 – Input 1 – Output <b>io_index</b> – Number that indicates the specific input or output port: 1–34 <b>eq_type</b> – Equalizer Types 0 – Bass 2 – Treble <b>eq_level</b> – Equalizer level: 0–15	Get Bass EQ level of the speaker output to 12: #EQ-LVL? <u>1</u> , <u>2</u> , <u>2</u> <CR>
ETH-PORT	Set Ethernet port protocol.  ⓘ If the port number you enter is already in use, an error is returned. The port number must be within the following range: 0-(2 <sup>16</sup> -1).	<b>COMMAND</b> #ETH-PORT <u>port_type</u> , <u>port_id</u> <CR> <b>FEEDBACK</b> ~nn@ETH-PORT <u>port_type</u> , <u>port_id</u> <CR> <LF>	<b>port_type</b> – TCP/UDP <b>port_id</b> – TCP/UDP port number: 1 – 65535	Set the Ethernet port protocol for TCP to port 12457: #ETH-PORT <u>0</u> , <u>12457</u> <CR>
ETH-PORT?	Get Ethernet port protocol.	<b>COMMAND</b> #ETH-PORT? <u>port_type</u> <CR> <b>FEEDBACK</b> ~nn@ETH-PORT <u>port_type</u> , <u>port_id</u> <CR> <LF>	<b>port_type</b> – TCP/UDP <b>port_id</b> – TCP / UDP port number: 1 – 65535	Get the Ethernet port protocol for UDP: #ETH-PORT? <u>TCP</u> <CR>
EXT-ETH-PORT	Set Ethernet port number for a specific module card.  ⓘ If the module card does not support Ethernet functions, ERR 034 is returned.	<b>COMMAND</b> #EXT-ETH-PORT <u>port_type</u> , <u>slot_id</u> , <u>port_id</u> <CR> <b>FEEDBACK</b> ~nn@EXT-ETH-PORT <u>port_type</u> , <u>slot_id</u> , <u>port_id</u> <CR> <LF>	<b>port_type</b> – TCP/UDP <b>slot_id</b> – Module ID (slot number): 0 or 100 – control module 1–17 <b>port_id</b> – TCP/UDP port number: 1 – 65535	Set the Ethernet TCP port number for the card in slot 1 to 12457: #EXT-ETH-PORT <u>TCP</u> , <u>1</u> , <u>12457</u> <CR>
EXT-ETH-PORT?	Get Ethernet port number for a specific module card.  ⓘ If the module card does not support Ethernet functions, ERR 034 is returned.	<b>COMMAND</b> #EXT-ETH-PORT? <u>port_type</u> , <u>slot_id</u> <CR> <b>FEEDBACK</b> ~nn@EXT-ETH-PORT? <u>port_type</u> , <u>slot_id</u> , <u>port_id</u> <CR> <LF>	<b>port_type</b> – TCP/UDP <b>slot_id</b> – Module ID (slot number): 0 or 100 – control module 1–17 <b>port_id</b> – TCP/UDP port number: 1 – 65535	Get the Ethernet TCP port number for the card in slot 1: #EXT-ETH-PORT? <u>TCP</u> , <u>1</u> <CR>
EXT-FAN-LEVEL	Set fan operation speed and mode.	<b>COMMAND</b> #EXT-FAN-LEVEL <u>fan_index</u> , <u>fan_speed</u> , <u>fan_mode</u> <CR> <b>FEEDBACK</b> ~nn@EXT-FAN-LEVEL <u>fan_index</u> , <u>fan_speed</u> , <u>fan_mode</u> , <u>fan_status</u> <CR> <LF>	<b>fan_index</b> – number that indicates the fan being set: 1–8 – Chassis fans 9–10 – Power supply fans <b>fan_speed</b> – 0–4 <b>fan_mode</b> – 0 – Fan operation automatically controlled based on the device temperature 1 – Continuous operation ⓘ The default is automatic. If a fan is set to continuous and the device is reset, the fan returns to automatic. <b>fan_status</b> – 0 – OK 1 – Error	Set fan #1 speed to 2 and mode to continuous: #EXT-FAN-LEVEL <u>1</u> , <u>2</u> , <u>1</u> <CR>
EXT-INFO-IO?	Get a list of ports according to signal type (video or audio) and direction (inputs or outputs).	<b>COMMAND</b> #EXT-INFO-IO? <u>signal_type</u> , <u>io_mode</u> <CR> <b>FEEDBACK</b> ~nn@Device <u>signal_type</u> , <u>io_mode</u> , <u>io_index</u> , <u>io_index</u> , ... <CR> <LF>	<b>signal_type</b> – 1 – video 2 – audio <b>io_mode</b> – Input/Output 0 – Input 1 – Output <b>io_index</b> – Number that indicates the specific input or output port: 1–34	Get the list of video input ports: #EXT-INFO-IO? <u>1</u> , <u>0</u> <CR>
EXT-NET-MAC?	Get the MAC address for a module card.	<b>COMMAND</b> #EXT-NET-MAC? <u>slot_id</u> <CR> <b>FEEDBACK</b> ~nn@EXT-NET-MAC? <u>slot_id</u> , <u>mac_address</u> <CR> <LF>	<b>slot_id</b> – Module ID (slot number): 0 or 100 – control module 1–17 <b>mac_address</b> – Unique MAC address. Format: XX-XX-XX-XX-XX (where X is a hex digit)	Get the MAC address for the card in slot 1: #EXT-NET-MAC? <u>1</u> <CR>
EXT-PING	Ping the IP address.	<b>COMMAND</b> #EXT-PING <u>ip_address</u> <CR> <b>FEEDBACK</b> ~nn@EXT-PING <u>ping_information</u> <CR> <LF>	<b>ip_address</b> – Format: xxx.xxx.xxx.xxx	Get the gateway IP address: #EXT-PING <u>192.168.1.39</u> <CR>

Function	Description	Syntax	Parameters/Attributes	Example
<b>FACTORY</b>	<p>Reset device to factory default configuration.</p> <p><b>i</b> This command deletes all user data from the device. The deletion can take some time.</p> <p>Your device may require powering off and powering on for the changes to take effect.</p>	<b>COMMAND</b> <b>#FACTORY&lt;CR&gt;</b> <b>FEEDBACK</b> ~nn@FACTORY_uk<CR><LF>		Reset the device to factory default configuration: <b>#FACTORY&lt;CR&gt;</b>
<b>FPGA-VER?</b>	<p>Get current FPGA version.</p>	<b>COMMAND</b> <b>#FPGA-VER?_fpga_id&lt;CR&gt;</b> <b>FEEDBACK</b> ~nn@FPGA-VER_fpga_id,expected_ver,ver<CR><LF>	<b>fpga_id</b> – FPGA id <b>expected_ver</b> – Expected FPGA version for current firmware <b>ver</b> – Actual FPGA version	Get current FPGA version: <b>#FPGA-VER?_1&lt;CR&gt;</b>
<b>GEDID</b>	<p>Get EDID support on certain input/output.</p> <p><b>i</b> For old devices that do not support this command, ~nn@ERR 002&lt;CR&gt;&lt;LF&gt; is received.</p>	<b>COMMAND</b> <b>#GEDID_io_mode,in_index&lt;CR&gt;</b> <b>FEEDBACK</b> ~nn@GEDID_io_mode,in_index,size<CR><LF>	<b>io_mode</b> – Input/Output 0 – Input 1 – Output 2 – Default EDID <b>io_index</b> – Number that indicates the specific input or output port: 1–34 <b>size</b> – Size of data to be sent from device, 0 means no EDID support	Get EDID support information for input 1: <b>#GEDID_111&lt;CR&gt;</b>
<b>GET</b>	<p>Get file contents.</p>	<b>COMMAND</b> <b>#GET_file_name&lt;CR&gt;</b> <b>FEEDBACK</b> Multi-line: ~nn@GET_file_name,file_size_ready<CR><LF> <b>contents</b> ~nn@GET_file_name_ok<CR><LF>	<b>file_name</b> – Name of file from which to get contents, includes: CONFIGURATION.JSON CONFIG_LOG.JSON METADATA.JSON, MESSAGES <b>contents</b> – Byte stream of file contents <b>file_size</b> – Size of file (device sends it in response to give user a chance to get ready)	<b>#GET_file_resp onse.dat&lt;CR&gt;</b>
<b>HDCP-MOD</b>	<p>Set HDCP mode.</p> <p><b>i</b> Set HDCP working mode on the device input:</p> <p>HDCP supported - HDCP_ON [default].</p> <p>HDCP not supported - HDCP OFF.</p> <p>HDCP support changes following detected sink - MIRROR OUTPUT.</p>	<b>COMMAND</b> <b>#HDCP-MOD_in_index,mode&lt;CR&gt;</b> <b>FEEDBACK</b> ~nn@HDCP-MOD_in_index,mode<CR><LF>	<b>in_index</b> – Number that indicates the specific input: 1–34 <b>mode</b> – HDCP mode: 0 – HDCP Off 1 – HDCP On	Set the input HDCP-MODE of IN 1 to Off: <b>#HDCP-MOD_1,0&lt;CR&gt;</b>
<b>HDCP-MOD?</b>	<p>Get HDCP mode.</p> <p><b>i</b> Set HDCP working mode on the device input:</p> <p>HDCP supported - HDCP_ON [default].</p> <p>HDCP not supported - HDCP OFF.</p> <p>HDCP support changes following detected sink - MIRROR OUTPUT.</p>	<b>COMMAND</b> <b>#HDCP-MOD?_in_index&lt;CR&gt;</b> <b>FEEDBACK</b> ~nn@HDCP-MOD_in_index,mode<CR><LF>	<b>in_index</b> – Number that indicates the specific input: 1–34 <b>mode</b> – HDCP mode: 0 – HDCP Off 3 – HDCP defined according to the connected output	Get the input HDCP-MODE of IN 1 HDMI: <b>#HDCP-MOD?_1&lt;CR&gt;</b>
<b>HDCP-STAT?</b>	<p>Get HDCP signal status.</p> <p><b>i</b> io_mode =1 – get the HDCP signal status of the sink device connected to the specified output.</p> <p>io_mode =0 – get the HDCP signal status of the source device connected to the specified input.</p>	<b>COMMAND</b> <b>#HDCP-STAT?_io_mode,in_index&lt;CR&gt;</b> <b>FEEDBACK</b> ~nn@HDCP-STAT_io_mode,in_index,status<CR><LF>	<b>io_mode</b> – Input/Output 0 – Input 1 – Output <b>in_index</b> – Number that indicates the specific input: 1–34 <b>status</b> – Signal encryption status - valid values On/Off 0 – HDCP Off 1 – HDCP On	Get the output HDCP-STATUS of IN 1: <b>#HDCP-STAT?_0,1&lt;CR&gt;</b>
<b>HELP</b>	<p>Get command list or help for specific command.</p>	<b>COMMAND</b> <b>#HELP&lt;CR&gt;</b> <b>#HELP_cmd_name&lt;CR&gt;</b> <b>FEEDBACK</b> 1. Multi-line: ~nn@Device_cmd_name,_cmd_name...<CR><LF> To get help for command use: HELP (COMMAND_NAME)<CR><LF> ~nn@HELP_cmd_name:<CR><LF> <b>description&lt;CR&gt;&lt;LF&gt;</b> <b>USAGE: usage&lt;CR&gt;&lt;LF&gt;</b>	<b>cmd_name</b> – Name of a specific command	Get the command list: <b>#HELP&lt;CR&gt;</b>  To get help for HDCP-MOD: HELP_hdcp-mod<CR>



Function	Description	Syntax	Parameters/Attributes	Example												
H-PHASE	Set H-phase.	<b>COMMAND</b> #H-PHASE_ <u>io_mode</u> , <u>io_index</u> , <u>value</u> <CR> <b>FEEDBACK</b> ~nn@H-PHASE_ <u>io_mode</u> , <u>io_index</u> , <u>value</u> <CR><LF>	<b>io_mode</b> – Input/Output 1 – Input 2 – Output <b>io_index</b> – Number that indicates the specific input or output port: 1–34 <b>value</b> – Video parameter in Kramer units: 0–63 ++ increase current value – decrease current value	Set H-phase: #H-PHASE_ <u>1</u> , <u>1</u> , <u>1</u> <CR>												
H-PHASE?	Get H-phase.	<b>COMMAND</b> #H-PHASE?_ <u>io_mode</u> , <u>io_index</u> <CR> <b>FEEDBACK</b> ~nn@H-PHASE_ <u>io_mode</u> , <u>io_index</u> , <u>value</u> <CR><LF>	<b>io_mode</b> – Input/Output 0 – Input 1 – Output <b>io_index</b> – Number that indicates the specific input or output port: 1–34 <b>value</b> – Video parameter in Kramer units, minus sign precedes negative values	Get H-phase: #H-PHASE?_ <u>1</u> , <u>1</u> <CR>												
HW-TEMP?	Get temperature of a specific region of the hardware.  ⓘ The Get command is not available for all parts of the hardware, and is device specific.	<b>COMMAND</b> #HW-TEMP?_ <u>region_id</u> <CR> <b>FEEDBACK</b> ~nn@HW-TEMP_ <u>region_id</u> , <u>temperature</u> <CR><LF>	<b>region_id</b> – ID of the region for which to get the temperature 0 – Control Board 1–17 – Module card slot number * – Matrix <b>temperature</b> – Temperature in Celsius of the HW region, rounded down to the closest integer	Get temperature of a specific region of the hardware: #HW-TEMP?_ <u>1</u> <CR>												
INFO-IO?	Get in/out count.	<b>COMMAND</b> #INFO-IO?_ <u>in_count</u> , <u>out_count</u> <CR> <b>FEEDBACK</b> ~nn@INFO-IO_ <u>IN</u> , <u>in_count</u> , <u>OUT</u> , <u>out_count</u> <CR><LF>	<b>in_count</b> – Number of inputs in the unit <b>out_count</b> – Number of outputs in the unit	Get inputs count: #INFO-IO?_ <u>1</u> <CR>												
INFO-PRST?	Get maximum preset count.  ⓘ In most units, video and audio presets with the same number are stored and recalled together by commands #PRST-STO and #PRST-RCL.	<b>COMMAND</b> #INFO-PRST?_ <u>vid</u> , <u>video_preset_count</u> , <u>aud</u> , <u>audio_preset_count</u> <CR><LF> <b>FEEDBACK</b> ~nn@INFO-PRST_ <u>vid</u> , <u>video_preset_count</u> , <u>aud</u> , <u>audio_preset_count</u> <CR><LF>	<b>video_preset_count</b> – Maximum number of video presets in the unit <b>audio_preset_count</b> – Maximum number of audio presets in the unit	Get number of video and audio presets: #INFO-PRST?_ <u>1</u> <CR>												
LDEDID	Write EDID data from external application to device.  ⓘ When the unit receives the LDEDID command it replies with READY and enters the special EDID packet wait mode. In this mode the unit can receive only packets and not regular protocol commands.  If the unit does not receive correct packets for 30 seconds or is interrupted for more than 30 seconds before receiving all packets, it sends timeout error ~nn@LDEDID_err01<CR><LF> and returns to the regular protocol mode. If the unit received data that is not a correct packet, it sends the corresponding error and returns to the regular protocol mode.	<b>COMMAND</b> Multi-step syntax <b>FEEDBACK</b> Step 1: #LDEDID_ <u>edid_io</u> , <u>dest_bitmask</u> , <u>edid_size</u> , <u>safe_mode</u> <CR> Response 1: ~nn@LDEDID_ <u>edid_io</u> , <u>dest_bitmask</u> , <u>edid_size</u> , <u>safe_mode</u> , <u>ready</u> <CR><LF> or ~nn@LDEDID_errnn<CR><LF> Step 2: If ready was received, send EDID_DATA Response 2: ~nn@LDEDID_ <u>edid_io</u> , <u>dest_bitmask</u> , <u>edid_size</u> , <u>safe_mode</u> , <u>ok</u> <CR><LF> or ~nn@LDEDID_errnn<CR><LF>	<b>edid_io</b> – EDID destination type (always input) 0 – Input <b>dest_bitmask</b> – Bitmap representing destination IDs. Format: 0x*****, where * is ASCII presentation of hex digit. The binary presentation of this number is a bit mask for destinations. Setting '1' means EDID data has to be copied to this destination <b>edid_size</b> – EDID data size <b>safe_mode</b> – Safe mode 0 – Device accepts the EDID as is without trying to adjust 1 – Device tries to adjust the EDID <b>edid_data</b> – Data in protocol packets <b>Using the Packet Protocol</b> Send a command: LDRV, LOAD, IROUT, LDEDID Receive Ready or ERR### If Ready: a. Send a packet, b. Receive OK on the last packet, c. Receive OK for the command Packet structure: <b>Packet ID</b> (1, 2, 3...) (2 bytes in length) <b>Length</b> (data length + 2 for CRC) – (2 bytes in length) <b>Data</b> (data length - 2 bytes) <b>CRC</b> – 2 bytes <table border="1" data-bbox="986 1512 1337 1556"> <tr> <td>01</td> <td>02</td> <td>03</td> <td>04</td> <td>05</td> <td></td> </tr> <tr> <td>Packet ID</td> <td>Length</td> <td>Data</td> <td>CRC</td> <td></td> <td></td> </tr> </table> 5. Response: ~nnnn_ok<CR><LF> (Where NNNN is the received packet ID in ASCII hex digits.)	01	02	03	04	05		Packet ID	Length	Data	CRC			Write the EDID data from an external application to the HDMI In 1 input without adjustment attempts: #LDEDID_0,0x1,2340,0<CR> Write the EDID data from an external application to HDMI In 1 and PC In inputs with adjustment attempts: #LDEDID_0,0x5,2340,1<CR>
01	02	03	04	05												
Packet ID	Length	Data	CRC													
LOAD	Load file to device.	<b>COMMAND</b> #LOAD_ <u>file_name</u> , <u>size</u> <CR> <b>FEEDBACK</b> Data sending negotiation: * Device - ~01@LOAD_ <u>file_name</u> , <u>size</u> , <u>ready</u> <CR><LF> * End User (+Device)- Send file in Protocol Packets * Device - ~01@LOAD_ <u>file_name</u> , <u>size</u> , <u>ok</u> <CR><LF>	<b>file_name</b> – Name of file to save on device <b>size</b> – Size of file data that is sent	Load the file_response.dat file to the device: #LOAD_ <u>file_respon</u> se.dat,5360<CR>												
LOCK-FP	Lock the front panel.	<b>COMMAND</b> #LOCK-FP_ <u>lock/unlock</u> <CR> <b>FEEDBACK</b> ~nn@LOCK-FP_ <u>lock/unlock</u> <CR><LF>	<b>lock/unlock</b> – On/Off 0 – unlock front panel 1 – lock front panel	Unlock front panel: #LOCK-FP_0<CR>												
LOCK-FP?	Get the front panel lock state.  ⓘ In NT-52N, this command includes the PortNumber (1-2) parameter.	<b>COMMAND</b> #LOCK-FP?_ <u>port</u> <CR> <b>FEEDBACK</b> ~nn@LOCK-FP_ <u>port</u> , <u>lock/unlock</u> <CR><LF>	<b>lock/unlock</b> – On/Off 0 – unlock front panel 1 – lock front panel	Get the front panel lock state: #LOCK-FP?_ <u>1</u> <CR>												



Function	Description	Syntax	Parameters/Attributes	Example
MIX	Set audio MIX.	<b>COMMAND</b> #MIX_out_index,mix_mode<CR> <b>FEEDBACK</b> ~nn@MIX_out_index,mix_mode<CR><LF>	<b>out_index</b> – Number that indicates the specific output: 1–34 <b>mix_mode</b> – On/Off 0 – Off 1 – On	Set audio MIX: #MIX_1,1<CR>
MIX?	Get audio MIX.	<b>COMMAND</b> #MIX?_out_index<CR> <b>FEEDBACK</b> ~nn@MIX_out_index,mix_mode<CR><LF>	<b>out_index</b> – Number that indicates the specific output: 1–34 <b>mix_mode</b> – On/Off 0 – Off 1 – On	Get audio MIX for output 1: #MIX?_1<CR>
MODEL?	Get device model.	<b>COMMAND</b> #MODEL?_<CR> <b>FEEDBACK</b> ~nn@MODEL_model_name<CR><LF>	<b>model_name</b> – String of up to 19 printable ASCII chars	Get the device model: #MODEL?_<CR>
MODULE-INFO?	Get module information.	<b>COMMAND</b> #MODULE-INFO?_slot_id<CR> <b>FEEDBACK</b> ~nn@MODULE-INFO_slot_id,m_direction,channel_start,channel_end,mod_type,fw_ver,upgradable,mod_status<CR><LF>	<b>slot_id</b> – Module ID (slot number): 0 – control module 1–17 – I/O cards 200 – test module 201 – keyboard software application 202 – keyboard hardware <b>m_direction</b> – Transmission direction 0 – input 1 – output 2 – OS system or applications in the control board or unknown <b>channel_start</b> – Start ID of the port in the device 1–34 <b>channel_end</b> – End ID of the port in the device 1–34 <b>mod_type</b> – Module type 18 – VGAA 32 – SDIA 47 – Control module 49 – H2 50 – H2A 57 – DTArC2 59 – DTxD2 203 – Sub-function for control board. 204 – InnerEthSwitcher in the matrix 205 – Power and fan monitor in the matrix 206 – Main board in the matrix <b>fw_ver</b> – Module firmware version XX.XX.XXXX where the digit groups are: major.minor.build version <b>upgradable</b> – Indicates whether the firmware can be upgraded 0 – not upgradable 1 – upgradable <b>mod_status</b> – Module status 0 – OK 1 – unknown error 2 – no communication 3 – module missing or offline	Get information for the module in slot 8: #MODULE-INFO?_8<CR>
MODULE-ISP	Set com port configuration.  <i>ⓘ</i> After running this command, power-cycle the device to return to normal operation.	<b>COMMAND</b> #MODULE-ISP_slot_id,baud_rate,data_bits,parity,stop,wait_time<CR> <b>FEEDBACK</b> ~nn@MODULE-ISP_slot_id,baud_rate,data_bits,parity,stop,wait_time<CR><LF>	<b>slot_id</b> – Module ID (slot number): 0 or 100 – control module 1–17 <b>baud_rate</b> – 9600 - 115200 <b>data_bits</b> – 7, 8 <b>parity</b> – Parity Type 0 – No 1 – Odd 2 – Even <b>stop</b> – 1, 2 <b>wait_time</b> – In seconds 1–60	Set baud rate for the module in slot 1 to 9600, 8 data bits, parity to none, stop bit to 1, and wait time to 30 seconds: #MODULE-ISP_1,9600,8,0,1,30<CR>
MODULE-TYPE?	Get module type.	<b>COMMAND</b> #MODULE-TYPE?_module_id<CR> <b>FEEDBACK</b> ~nn@MODULE-TYPE_module_id,mod_type,mod_status<CR><LF>	<b>module_id</b> – Number that identifies the module 1–17 <b>mod_type</b> – Module type 18 – VGAA 32 – SDIA 47 – Control module 49 – H2 50 – H2A 57 – DTArC2 59 – DTxD2 <b>mod_status</b> – Module status 0 – OK 1 – Unknown error 2 – No communication 3 – Module missing or offline	Get module type: #MODULE-TYPE?_1<CR>

Function	Description	Syntax	Parameters/Attributes	Example
<b>MODULE-VER?</b>	Get module version.  <i>i</i> Some devices do not set the new machine number until the device is restarted.  Some devices can change the machine number only from DIP-switches.	<b>COMMAND</b> <b>#MODULE-VER?_module_id&lt;CR&gt;</b>  <b>FEEDBACK</b> ~nn@MODULE-VER_module_id, fw_version<CR><LF>	<b>module_id</b> – Number that identifies the module 0 – control module 1–17 – I/O cards 220 – Linux OS <b>fw_version</b> – XX.XX.XXXX where the digit groups are: major.minor.build version	Get module version: <b>#MODULE-VER?_1&lt;CR&gt;</b>
<b>MUTE</b>	Set audio mute.	<b>COMMAND</b> <b>#MUTE_out_index,mute_mode&lt;CR&gt;</b>  <b>FEEDBACK</b> ~nn@MUTE_out_index,mute_mode<CR><LF>	<b>out_index</b> – Number that indicates the specific output: 1–34 <b>mute_mode</b> – On/Off 0 – Off 1 – On	Set Output 1 to mute: <b>#MUTE_1,1&lt;CR&gt;</b>
<b>MUTE?</b>	Get audio mute.	<b>COMMAND</b> <b>#MUTE?_out_index&lt;CR&gt;</b>  <b>FEEDBACK</b> ~nn@MUTE_out_index,mute_mode<CR><LF>	<b>out_index</b> – Number that indicates the specific output: 1–34 <b>mute_mode</b> – On/Off 0 – Off 1 – On	Get mute status of output 1 <b>#MUTE?_1&lt;CR&gt;</b>
<b>NAME</b>	Set machine (DNS) name.  <i>i</i> The machine name is not the same as the model name. The machine name is used to identify a specific machine or a network in use (with DNS feature on).	<b>COMMAND</b> <b>#NAME_machine_name&lt;CR&gt;</b>  <b>FEEDBACK</b> ~nn@NAME_machine_name<CR><LF>	<b>machine_name</b> – String of up to 15 alphanumeric chars (can include hyphen, not at the beginning or end)	Set the DNS name of the device to room-442: <b>#NAME_room-442&lt;CR&gt;</b>
<b>NAME?</b>	Get machine (DNS) name.  <i>i</i> The machine name is not the same as the model name. The machine name is used to identify a specific machine or a network in use (with DNS feature on).	<b>COMMAND</b> <b>#NAME?_&lt;CR&gt;</b>  <b>FEEDBACK</b> ~nn@NAME_machine_name<CR><LF>	<b>machine_name</b> – String of up to 15 alphanumeric chars (can include hyphen, not at the beginning or end)	Get the DNS name of the device: <b>#NAME?_&lt;CR&gt;</b>
<b>NAME-RST</b>	Reset machine (DNS) name to factory default.  <i>i</i> Factory default of machine (DNS) name is "KRAMER." + 4 last digits of device serial number.	<b>COMMAND</b> <b>#NAME-RST&lt;CR&gt;</b>  <b>FEEDBACK</b> ~nn@NAME-RST_ok<CR><LF>		Reset the machine name (S/N last digits are 0102): <b>#NAME-RST_kramer_0102&lt;CR&gt;</b>
<b>NET-CONFIG</b>	Set a network configuration.  <i>i</i> If the gateway address is not compliant to the subnet mask used for the host IP, the command will return an error. Subnet and gateway compliancy specified by RFC950.	<b>COMMAND</b> <b>#NET-CONFIG_netw_id,net_ip,net_mask,gateway&lt;CR&gt;</b> <b>&gt;</b>  <b>FEEDBACK</b> ~nn@NET-CONFIG_netw_id,net_ip,net_mask,gateway<CR><LF>	<b>netw_id</b> – Network ID—the device network interface 0 or 100 – control module 1–17 – module cards <b>net_ip</b> – Network IP <b>net_mask</b> – Network mask <b>gateway</b> – Network gateway	Set the device network parameters to IP address 192.168.113.10, net mask 255.255.0.0, and gateway 192.168.0.1: <b>#NET-CONFIG_0,192.168.113.10,255.255.0.0,192.168.0.1&lt;CR&gt;</b>
<b>NET-CONFIG?</b>	Get a network configuration.	<b>COMMAND</b> <b>#NET-CONFIG?_netw_id&lt;CR&gt;</b>  <b>FEEDBACK</b> ~nn@NET-CONFIG_netw_id,net_ip,net_mask,gateway<CR><LF>	<b>netw_id</b> – Network ID—the device network interface 0 or 100 – control module 1–17 – module cards <b>net_ip</b> – Network IP <b>net_mask</b> – Network mask <b>gateway</b> – Network gateway	Get network configuration for the module card in slot 2: <b>#NET-CONFIG?_2&lt;CR&gt;</b>

Function	Description	Syntax	Parameters/Attributes	Example
NET-DHCP	<p>Set DHCP mode.</p> <p><b>i</b> Only 1 is relevant for the mode value. To disable DHCP, the user must configure a static IP address for the device.</p> <p>Connecting Ethernet to devices with DHCP may take more time in some networks.</p> <p>To connect with a randomly assigned IP by DHCP, specify the device DNS name (if available) using the <b>NAME</b> command. You can also get an assigned IP by direct connection to USB or RS-232 protocol port, if available.</p> <p>For proper settings consult your network administrator.</p>	<p><b>COMMAND</b></p> <pre>#NET-DHCP_dhcp_state&lt;CR&gt;</pre> <p><b>FEEDBACK</b></p> <pre>~nn@NET-DHCP_dhcp_state&lt;CR&gt;&lt;LF&gt;</pre>	<p><b>dhcp_state</b> –</p> <p>0 – Do not use DHCP. Use the IP set by the factory or using the <b>NET-CONFIG</b> or <b>NET-IP</b> command.</p> <p>1 – Try to use DHCP. If unavailable, use the IP address set by the factory or the <b>NET-CONFIG</b> or <b>NET-IP</b> command.</p>	<p>Enable DHCP mode for port 1, if available:</p> <pre>#NET-DHCP_1,1&lt;CR&gt;</pre>
NET-DHCP?	Get DHCP mode.	<p><b>COMMAND</b></p> <pre>#NET-DHCP?_&lt;CR&gt;</pre> <p><b>FEEDBACK</b></p> <pre>~nn@NET-DHCP_dhcp_mode&lt;CR&gt;&lt;LF&gt;</pre>	<p><b>dhcp_mode</b> –</p> <p>0 – Do not use DHCP. Use the IP set by the factory or using the <b>NET-CONFIG</b> or <b>NET-IP</b> command.</p> <p>1 – Try to use DHCP. If unavailable, use the IP set by the factory or using the <b>NET-CONFIG</b> or <b>NET-IP</b> command.</p>	<p>Get DHCP mode for port 1:</p> <pre>#NET-DHCP?_1&lt;CR&gt;</pre>
NET-GATE	<p>Set gateway IP.</p> <p><b>i</b> A network gateway connects the device via another network and maybe over the Internet. Be careful of security issues. For proper settings consult your network administrator.</p>	<p><b>COMMAND</b></p> <pre>#NET-GATE_ip_address&lt;CR&gt;</pre> <p><b>FEEDBACK</b></p> <pre>~nn@NET-GATE_ip_address&lt;CR&gt;&lt;LF&gt;</pre>	<p><b>ip_address</b> – Format: xxx.xxx.xxx.xxx</p>	<p>Set the gateway IP address to 192.168.0.1:</p> <pre>#NET-GATE_192.168.0.00.001&lt;CR&gt;</pre>
NET-GATE?	<p>Get gateway IP.</p> <p><b>i</b> A network gateway connects the device via another network and maybe over the Internet. Be careful of security issues. For proper settings consult your network administrator.</p>	<p><b>COMMAND</b></p> <pre>#NET-GATE?_&lt;CR&gt;</pre> <p><b>FEEDBACK</b></p> <pre>~nn@NET-GATE_ip_address&lt;CR&gt;&lt;LF&gt;</pre>	<p><b>ip_address</b> – Format: xxx.xxx.xxx.xxx</p>	<p>Get the gateway IP address:</p> <pre>#NET-GATE?_&lt;CR&gt;</pre>
NET-IP	<p>Set IP address.</p> <p><b>i</b> For proper settings consult your network administrator.</p>	<p><b>COMMAND</b></p> <pre>#NET-IP_ip_address&lt;CR&gt;</pre> <p><b>FEEDBACK</b></p> <pre>~nn@NET-IP_ip_address&lt;CR&gt;&lt;LF&gt;</pre>	<p><b>ip_address</b> – Format: xxx.xxx.xxx.xxx</p>	<p>Set the IP address to 192.168.1.39:</p> <pre>#NET-IP_192.168.001.039&lt;CR&gt;</pre>
NET-IP?	Get IP address.	<p><b>COMMAND</b></p> <pre>#NET-IP?_&lt;CR&gt;</pre> <p><b>FEEDBACK</b></p> <pre>~nn@NET-IP_ip_address&lt;CR&gt;&lt;LF&gt;</pre>	<p><b>ip_address</b> – Format: xxx.xxx.xxx.xxx</p>	<p>Get the IP address:</p> <pre>#NET-IP?_&lt;CR&gt;</pre>
NET-MAC?	Get MAC address.	<p><b>COMMAND</b></p> <pre>#NET-MAC?_&lt;CR&gt;</pre> <p><b>FEEDBACK</b></p> <pre>~nn@NET-MAC_mac_address&lt;CR&gt;&lt;LF&gt;</pre>	<p><b>mac_address</b> – Unique MAC address. Format: XX-XX-XX-XX-XX-XX where X is hex digit</p>	<pre>#NET-MAC?_id&lt;CR&gt;</pre>
NET-MASK	<p>Set subnet mask.</p> <p><b>i</b> For proper settings consult your network administrator.</p>	<p><b>COMMAND</b></p> <pre>#NET-MASK_net_mask&lt;CR&gt;</pre> <p><b>FEEDBACK</b></p> <pre>~nn@NET-MASK_net_mask&lt;CR&gt;&lt;LF&gt;</pre>	<p><b>net_mask</b> – Format: xxx.xxx.xxx.xxx</p>	<p>Set the subnet mask to 255.255.0.0:</p> <pre>#NET-MASK_255.255.00.000&lt;CR&gt;</pre>
NET-MASK?	Get subnet mask.	<p><b>COMMAND</b></p> <pre>#NET-MASK?_&lt;CR&gt;</pre> <p><b>FEEDBACK</b></p> <pre>~nn@NET-MASK_net_mask&lt;CR&gt;&lt;LF&gt;</pre>	<p><b>net_mask</b> – Format: xxx.xxx.xxx.xxx</p>	<p>Get the subnet mask:</p> <pre>#NET-MASK?_&lt;CR&gt;</pre>
PORT-DIRECTION?	<p>Get port direction for video port.</p> <p><b>i</b> This command gets the direction of a bidirectional port.</p>	<p><b>COMMAND</b></p> <pre>#PORT-DIRECTION?_port_index&lt;CR&gt;</pre> <p><b>FEEDBACK</b></p> <pre>~nn@PORT-DIRECTION_port_index,direction&lt;CR&gt;&lt;LF&gt;</pre>	<p><b>port_index</b> – Port number 1–34</p> <p><b>direction</b> – 0 – Input 1 – Output</p>	<p>Get port direction for port 2:</p> <pre>#PORT-DIRECTION?_5&lt;CR&gt;</pre>
PROT-VER?	Get device protocol version.	<p><b>COMMAND</b></p> <pre>#PROT-VER?_&lt;CR&gt;</pre> <p><b>FEEDBACK</b></p> <pre>~nn@PROT-VER_3000:version&lt;CR&gt;&lt;LF&gt;</pre>	<p><b>version</b> – XX.XX where X is a decimal digit</p>	<p>Get the device protocol version:</p> <pre>#PROT-VER?_&lt;CR&gt;</pre>

Function	Description	Syntax	Parameters/Attributes	Example
PRST-AUD?	Get audio connections from saved preset.  Video and audio presets with the same number are stored and recalled together by commands #PRST-STO and #PRST-RCL.	<b>COMMAND</b> #PRST-AUD?_preset,out_index<CR> #PRST-AUD?_preset,*<CR> <b>FEEDBACK</b> ~@PRST-AUD_preset,in_index>out_index<CR><LF> ~@PRST-AUD_preset,in_index>1,in_index>2,in_index>3,...<CR><LF>	<b>preset</b> – Preset number <b>in_index</b> – Number that indicates the specific input: 1 – N (N= the total number of inputs) 0 – Output is disconnected > – Connection character between in and out parameters <b>out_index</b> – Number that indicates the specific output: 1 – 34 * – All outputs	Get audio connection for OUT 3 from saved preset 1: #PRST-AUD?_1,3<CR>
PRST-LST?	Get saved preset list.  Video and audio presets with the same number are stored and recalled together by commands #PRST-STO and #PRST-RCL.	<b>COMMAND</b> #PRST-LST?_<CR> <b>FEEDBACK</b> ~nn@PRST-LST_preset,preset,...<CR><LF>	<b>preset</b> – Preset number	Show preset list: #PRST-LST?<CR>
PRST-RCL	Recall saved preset list.  Video and audio presets with the same number are stored and recalled together by commands #PRST-STO and #PRST-RCL.	<b>COMMAND</b> #PRST-RCL_preset<CR> <b>FEEDBACK</b> ~nn@PRST-RCL_preset<CR><LF>	<b>preset</b> – Preset number	Recall preset 1: #PRST-RCL_1<CR>
PRST-STO	Store current connections, volumes, and modes in preset.  Video and audio presets with the same number are stored and recalled together by commands #PRST-STO and #PRST-RCL.	<b>COMMAND</b> #PRST-STO_preset<CR> <b>FEEDBACK</b> ~nn@PRST-STO_preset<CR><LF>	<b>preset</b> – Preset number	Store the current configuration under preset 1: #PRST-STO_1<CR>
PRST-VID?	Get video connections from saved preset.  In most units, video and audio presets with the same number are stored and recalled together by commands #PRST-STO and #PRST-RCL.	<b>COMMAND</b> #PRST-VID?_preset,out_index<CR> #PRST-VID?_preset,*<CR> <b>FEEDBACK</b> ~nn@PRST-VID_preset,in_index>out_index<CR><LF> ~nn@PRST-VID_preset,in_index>1,in_index>2,in_index>3,...<CR><LF>	<b>preset</b> – Preset number <b>in_index</b> – Number that indicates the specific input: 1 – 34 0 – Output is disconnected > – Connection character between in and out parameters <b>out_index</b> – Output number 1–34 * for all outputs	Get video connections from preset 3 for all outputs: #PRST-VID?_3,*<CR>
RESET	Reset device.  To avoid locking the port due to a USB bug in Windows, disconnect USB connections immediately after running this command. If the port was locked, disconnect and reconnect the cable to reopen the port.	<b>COMMAND</b> #RESET<CR> <b>FEEDBACK</b> ~nn@RESET_ok<CR><LF>		Reset the device: #RESET<CR>
ROUTE	Set layer routing.  This command replaces all other routing commands.	<b>COMMAND</b> #ROUTE_layer_type,out_index,in_index<CR> <b>FEEDBACK</b> ~nn@ROUTE_layer_type,out_index<CR><LF>	<b>layer_type</b> Layer Enumeration 1 – Video 2 – Audio <b>out_index</b> – Number that indicates the specific output 1–34 * for all outputs <b>in_index</b> – Number that indicates the specific input 0 – disconnect the output 1–34	Route HDMI IN 2 to HDMI OUT 4: #ROUTE_1,2,4<CR>
ROUTE?	Get layer routing.  This command replaces all other routing commands.	<b>COMMAND</b> #ROUTE?_layer_type,out_index<CR> <b>FEEDBACK</b> ~nn@ROUTE_layer_type,out_index,in_index<CR><LF>	<b>layer_type</b> Number that represents the type of signal 1 – Video 2 – Audio <b>out_index</b> – Number that indicates the specific output 1–34 * for all outputs <b>in_index</b> – Number that indicates the specific input 0 – output is disconnected 1–34	Get the video routing for output 2: #ROUTE?_1,2<CR>
SIGNAL?	Get input signal status.	<b>COMMAND</b> #SIGNAL?_in_index<CR> <b>FEEDBACK</b> ~nn@SIGNAL_in_index,status<CR><LF>	<b>in_index</b> – Number that indicates the specific input: 1–34 <b>status</b> – Signal status according to signal validation: 0 – Off 1 – On	Get the input signal status of IN 1: #SIGNAL?_1<CR>

Function	Description	Syntax	Parameters/Attributes	Example
SN?	Get device serial number.	<b>COMMAND</b> #SN?_<CR> <b>FEEDBACK</b> ~nn@SN_serial_num<CR><LF>	serial_num – 14 decimal digits, factory assigned	Get the device serial number: #SN?_<CR>
TEST-MODE	Put device into keyboard testing mode.  ⓘ Refer to User Manual for test instructions.	<b>COMMAND</b> #TEST-MODE<CR> <b>FEEDBACK</b> ~nn@TEST-MODE_result<CR><LF>	result – OK – Device has entered keyboard testing mode	Put device into keyboard testing mode: #TEST-MODE<CR>
TIME	Set device time and date.  ⓘ The year must be 4 digits.  The device does not validate the day of week from the date.  Time format - 24 hours.  Date format - Day, Month, Year.	<b>COMMAND</b> #TIME_day_of_week,date,time<CR> <b>FEEDBACK</b> ~nn@TIME_day_of_week,date,time<CR><LF>	day_of_week – One of {SUN,MON,TUE,WED,THU,FRI,SAT} date – Format: DD-MM-YYYY. time – Format: hh:mm:ss where hh = hours mm = minutes ss = seconds	Set device time and date to December 5, 2018 at 2:30pm: #TIME_mon_05-12-2018,14:30:00<CR>
TIME?	Get device time and date.  ⓘ The year must be 4 digits.  The device does not validate the day of week from the date.  Time format - 24 hours.  Date format - Day, Month, Year.	<b>COMMAND</b> #TIME?_<CR> <b>FEEDBACK</b> ~nn@TIME_day_of_week,date,time<CR><LF>	day_of_week – One of {SUN,MON,TUE,WED,THU,FRI,SAT} date – Format: DD-MM-YYYY where DD = Day MM = Month YYYY = Year time – Format: hh:mm:ss where hh = hours mm = minutes ss = seconds	Get device time and date: #TIME?<CR>
TREBLE	Set audio treble level.	<b>COMMAND</b> #TREBLE_io_index,treble_level<CR> <b>FEEDBACK</b> ~nn@TREBLE_io_index,treble_level<CR><LF>	io_index – Number that indicates the specific input or output port: 1-34 treble_level – Audio parameter in Kramer units, minus sign precedes negative values 0-15 ++ increase current value – decrease current value	Set audio treble level: #TREBLE_1,1<CR>
TREBLE?	Get audio treble level.	<b>COMMAND</b> #TREBLE?_io_index<CR> <b>FEEDBACK</b> ~nn@TREBLE_io_index,treble_level<CR><LF>	io_index – Number that indicates the specific input or output port: 1-34 treble_level – Audio parameter in Kramer units, minus sign precedes negative values 0-15	Get audio treble level: #TREBLE?_1<CR>
VERSION?	Get firmware version number.	<b>COMMAND</b> #VERSION?_<CR> <b>FEEDBACK</b> ~nn@VERSION_firmware_version<CR><LF>	firmware_version – XX.XX.XXXX where the digit groups are: major.minor.build version	Get the device firmware version number: #VERSION?_<CR>
VID	Set video switch state.	<b>COMMAND</b> #VID_in_id>out_id<CR> <b>FEEDBACK</b> ~nn@VID_in_id>out_id<CR><LF>	in_id – Indicates the ID of the input: 0 – Disconnect output 1-34 > – Connection character between in and out parameters out_id – Output number 1-34 * for all outputs	Switch IN 1 to OUT 3: #VID_1>3<CR>
VID?	Get video switch state.  ⓘ The GET command identifies input switching on Step-in clients.  ⓘ The SET command is for remote input switching on Step-in clients (essentially via by the Web).	<b>COMMAND</b> #VID?_out_id<CR> <b>FEEDBACK</b> ~nn@VID_>out_id<CR><LF>	in_id – Indicates the ID of the input: 0 – Output disconnected 1-34 > – Connection character between in and out parameters out_id – Output number 1-34 * for all outputs	Get video switch state: #VID?_2<CR>
X-MUTE	Set mute ON/OFF on a specific signal.  ⓘ This command is designed to Mute a Signal. This means that it could be applicable on any type of signal. Could be audio, video and maybe IR, USB, or data if this capability is supported by the product.	<b>COMMAND</b> #X-MUTE_<direction_type>.<port_format>.<port_index>.<signal_type>.<index>,state<CR> <b>FEEDBACK</b> ~nn@X-MUTE_<direction_type>.<port_format>.<port_index>.<signal_type>.<index>,state<CR><LF>	The following attributes comprise the signal ID: ▪ <direction_type> – Direction of the port: o IN – Input o OUT – Output ▪ <port_format> – Type of signal on the port: o ANALOG_AUDIO ▪ <port_index> – The port number as printed on the front or rear panel ▪ <signal_type> – Signal ID attribute: o AUDIO ▪ <index> – 1 state – OFF/ON (not case sensitive)	Mute the video on HDMI OUT 2: #X-MUTE_out_hdmi.2.video.1,o n<CR>

Function	Description	Syntax	Parameters/Attributes	Example
<code>X-MUTE?</code>	Get mute ON/OFF state on a specific signal.	<p><b>COMMAND</b></p> <pre>#X-MUTE?_&lt;direction_type&gt;. &lt;port_format&gt;. &lt;port_index&gt;.&lt;signal_type&gt;.&lt;index&gt;&lt;CR&gt;</pre> <p><b>FEEDBACK</b></p> <pre>~nn@X-MUTE_&lt;direction_type&gt;.&lt;port_format&gt;. &lt;port_index&gt;.&lt;signal_type&gt;. &lt;index&gt;,state&lt;CR&gt;&lt;LF&gt;</pre>	<p>The following attributes comprise the signal ID:</p> <ul style="list-style-type: none"> <li>▪ <b>&lt;direction_type&gt;</b> – Direction of the port: <ul style="list-style-type: none"> <li>○ IN – Input</li> <li>○ OUT – Output</li> </ul> </li> <li>▪ <b>&lt;port_format&gt;</b> – Type of signal on the port: <ul style="list-style-type: none"> <li>○ ANALOG_AUDIO</li> </ul> </li> <li>▪ <b>&lt;port_index&gt;</b> – The port number as printed on the front or rear panel</li> <li>▪ <b>&lt;signal_type&gt;</b> – Signal ID attribute: <ul style="list-style-type: none"> <li>○ AUDIO</li> </ul> </li> <li>▪ <b>&lt;index&gt;</b> – 1</li> </ul> <p><b>state</b> – OFF/ON (not case sensitive)</p>	<p>Get the mute ON/OFF state on a specific signal:</p> <pre>#X-MUTE?_out.h dmi.4.video.1&lt; CR&gt;</pre>

---

# Result and Error Codes

## Syntax

In case of an error, the device responds with an error message. The error message syntax:

- **~NN@ERR XXX<CR><LF>** – when general error, no specific command
- **~NN@CMD ERR XXX<CR><LF>** – for specific command
- **NN** – machine number of device, default = 01
- **XXX** – error code

## Error Codes

Error Name	Error Code	Description
P3K_NO_ERROR	0	No error
ERR_PROTOCOL_SYNTAX	1	Protocol syntax
ERR_COMMAND_NOT_AVAILABLE	2	Command not available
ERR_PARAMETER_OUT_OF_RANGE	3	Parameter out of range
ERR_UNAUTHORIZED_ACCESS	4	Unauthorized access
ERR_INTERNAL_FW_ERROR	5	Internal FW error
ERR_BUSY	6	Protocol busy
ERR_WRONG_CRC	7	Wrong CRC
ERR_TIMEOUT	8	Timeout
ERR_RESERVED	9	(Reserved)
ERR_FW_NOT_ENOUGH_SPACE	10	Not enough space for data (firmware, FPGA...)
ERR_FS_NOT_ENOUGH_SPACE	11	Not enough space – file system
ERR_FS_FILE_NOT_EXISTS	12	File does not exist
ERR_FS_FILE_CANT_CREATED	13	File can't be created
ERR_FS_FILE_CANT_OPEN	14	File can't open
ERR_FEATURE_NOT_SUPPORTED	15	Feature is not supported
ERR_RESERVED_2	16	(Reserved)
ERR_RESERVED_3	17	(Reserved)
ERR_RESERVED_4	18	(Reserved)
ERR_RESERVED_5	19	(Reserved)
ERR_RESERVED_6	20	(Reserved)
ERR_PACKET_CRC	21	Packet CRC error
ERR_PACKET_MISSED	22	Packet number isn't expected (missing packet)
ERR_PACKET_SIZE	23	Packet size is wrong
ERR_RESERVED_7	24	(Reserved)
ERR_RESERVED_8	25	(Reserved)
ERR_RESERVED_9	26	(Reserved)
ERR_RESERVED_10	27	(Reserved)
ERR_RESERVED_11	28	(Reserved)
ERR_RESERVED_12	29	(Reserved)
ERR_EDID_CORRUPTED	30	EDID corrupted
ERR_NON_LISTED	31	Device specific errors
ERR_SAME_CRC	32	File has the same CRC – no changed
ERR_WRONG_MODE	33	Wrong operation mode
ERR_NOT_CONFIGURED	34	Device/chip was not initialized