Protocol 3000 for VP-427X

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**:

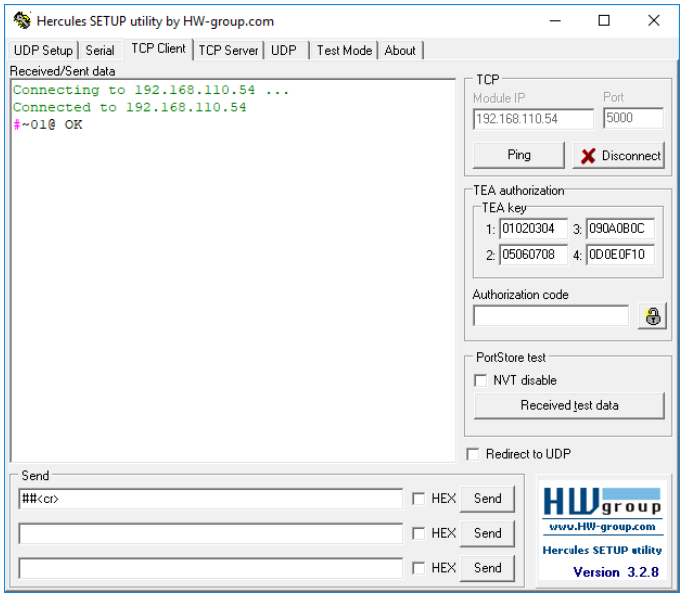
Prefix	Command Name	Constant (Space)	Parameter(s)	Suffix
#	Command	␣	Parameter	<CR>

- **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 **VP-427X**. 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. Step-in master products use this command to identify the availability of a device.	COMMAND #<CR> FEEDBACK ~nn@␣ok<CR><LF>		#<CR>
AUD-LVL	Set volume level.	COMMAND #AUD-LVL␣io_mode,io_index,vol_level<CR> FEEDBACK ~nn@AUD-LVL␣io_mode,io_index,vol_level<CR><LF>	io_mode – Input/Output 1 – Output io_index – Number that indicates the specific input or output port: 1 – AUDIO OUTPUT vol_level – Volume level 0 to 100; ++ (increase current value by 1); -- (decrease current value by 1)	Set AUDIO OUTPUT level to 50: #AUD-LVL␣1,1,50<CR>
AUD-LVL?	Get volume level.	COMMAND #AUD-LVL?␣io_mode,io_index<CR> FEEDBACK ~nn@AUD-LVL␣io_mode,io_index,vol_level<CR><LF>	io_mode – Input/Output 1 – Output io_index – Number that indicates the specific input or output port: 1 – AUDIO OUTPUT vol_level – Volume level 0 to 100	Get AUDIO OUT 1 level #AUD-LVL?␣1,1<CR>
AV-SW-MODE	Set input auto switch mode (per output).	COMMAND #AV-SW-MODE␣layer_type,out_index,connection_mode<CR> FEEDBACK ~nn@AV-SW-MODE␣layer_type,out_index,connection_mode<CR><LF>	layer_type – Number that indicates the signal type: 1 – Video out_index – Number that indicates the specific output: 1 – HDMI OUTPUT connection_mode – Connection mode 0 – manual 1 – auto scan 2 – last connected	Set input auto switch mode (per output) for audio 1 to manual: #AV-SW-MODE␣1,1,0<CR>
AV-SW-MODE?	Get input auto switch mode (per output).	COMMAND #AV-SW-MODE?␣layer_type,out_index<CR> FEEDBACK ~nn@AV-SW-MODE␣layer_type,out_index,connection_mode<CR><LF>	layer_type – Number that indicates the signal type: 1 – Video out_index – Number that indicates the specific output: 1 – HDMI OUTPUT connection_mode – Connection mode 0 – manual 1 – auto scan 2 – last connected	Get the input audio switch mode for HDBT Out: #AV-SW-MODE?␣1,1<CR>
BUILD-DATE?	Get device build date.	COMMAND #BUILD-DATE?␣<CR> FEEDBACK ~nn@BUILD-DATE␣date,time<CR><LF>	date – Format: YYYY/MM/DD where YYYY = Year MM = Month DD = Day time – Format: hh:mm:ss where hh = hours mm = minutes ss = seconds	Get the device build date: #BUILD-DATE?<CR>
CEC	Set display to ON/OFF	COMMAND #CEC␣state<CR> FEEDBACK ~nn@CEC␣state<CR><LF>	state – CEC state 0 – Off 1 – On	Set display to OFF via CEC: #CEC-ON<CR>
CEC-PASS	Set CEC device bypass.	COMMAND #CEC-PASS␣state<CR> FEEDBACK ~nn@CEC-PASS␣state<CR><LF>	state – CEC state 0 – Off 1 – On	Bypass device: #CEC-PASS␣1<CR>

Function	Description	Syntax	Parameters/Attributes	Example
CPEDID	Copy EDID data from the output to the input EEPROM. ⓘ Destination bitmap size depends on device properties (for 64 inputs it is a 64-bit word). Example: bitmap 0x0013 means inputs 1,2 and 5 are loaded with the new EDID. In certain products Safe_mode is an optional parameter. See the HELP command for its availability.	COMMAND **#CPEDID** edid_io,src_id,edid_io,dest_bitmap<CR> or **#CPEDID** edid_io,src_id,edid_io,dest_bitmap,safe_mode<CR> FEEDBACK **~nn@CPEDID** edid_io,src_id,edid_io,dest_bitmap<CR><LF> **~nn@CPEDID** edid_io,src_id,edid_io,dest_bitmap,safe_mode<CR><LF>	**edid_io** – *EDID* source type (usually output) 1 – Output **src_id** – Number of chosen source stage 1=Def. 1080P 2=Def. 4K2K(3G) 3=Def. 4K2K(3G-4:2:0) 4=USER1 5=USER2 6=OUTPUT 7= Def. 4K2K(6G) – for HDMI only (not relevant for HDBT) **edid_io** – *EDID destination* type (usually input) 0 – Input **dest_bitmap** – 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. **safe_mode** – 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)	Copy the EDID data from USER1 to the Input: **#CPEDID** 1,4,0,0x01<CR> Copy the EDID data from the default EDID source to the Input: **#CPEDID** 1,1,0,0x01<CR>
DISPLAY?	Get output HPD status.	COMMAND **#DISPLAY?** out_index<CR> FEEDBACK **~nn@DISPLAY** out_index,status<CR><LF>	**out_index** – Number that indicates the specific output: 1 – HDMI OUTPUT **status** – HPD status according to signal validation 0 – Signal or sink is not valid 1 – Signal or sink is valid 2 – Sink and EDID is valid	Get the output HPD status of HDMI OUTPUT: **#DISPLAY?** 1<CR>
FCT-SN	Set serial number.	COMMAND **#FCT-SN** serial_num<CR> FEEDBACK **~nn@FCT-SN** serial_num<CR><LF>	**serial_num** – 14 decimal digits	Set serial number: **#FCT-SN** 19763840581123<CR>
HDCP-MOD	Set HDCP mode. ⓘ Set HDCP working mode on the device input: HDCP supported - HDCP_ON [default]. HDCP not supported - HDCP OFF. HDCP support changes following detected sink - MIRROR OUTPUT. When you define 3 as the mode, the HDCP status is defined according to the connected output in the following priority: OUT 1, OUT 2. If the connected display on OUT 2 supports HDCP, but OUT 1 does not, then HDCP is defined as not supported. If OUT 1 is not connected, then HDCP is defined by OUT 2.	COMMAND **#HDCP-MOD** in_index,mode<CR> FEEDBACK **~nn@HDCP-MOD** in_index,mode<CR><LF>	**in_index** – Number that indicates the specific input: 1 – HDBT INPUT 1 – HDMI INPUT **mode** – HDCP mode: 0 – HDCP Off 1 – HDCP On	Set the input HDCP-MODE of HDBT INPUT to Off: **#HDCP-MOD** 1,0<CR>
HDCP-MOD?	Get HDCP mode. ⓘ Set HDCP working mode on the device input: HDCP supported - HDCP_ON [default]. HDCP not supported - HDCP OFF. HDCP support changes following detected sink - MIRROR OUTPUT.	COMMAND **#HDCP-MOD?** in_index<CR> FEEDBACK **~nn@HDCP-MOD** in_index,mode<CR><LF>	**in_index** – Number that indicates the specific input: 1 – HDBT INPUT 2 – HDMI INPUT **mode** – HDCP mode: 0 – HDCP Off 1 – HDCP On	Get the input HDCP-MODE of HDMI INPUT: **#HDCP-MOD?** 2<CR>

Function	Description	Syntax	Parameters/Attributes	Example
HELP	Get command list or help for specific command.	**COMMAND** **#HELP<CR>** **#HELP** cmd_name**<CR>** **FEEDBACK** 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>** **description<CR><LF>** **USAGE:usage<CR><LF>**	**cmd_name** – Name of a specific command	Get the command list: **#HELP<CR>** To get help for AV-SW-TIMEOUT: HELP **av-sw-timeout**<CR>
MODEL?	Get device model. ⓘ This command identifies equipment connected to VP-427X and notifies of identity changes to the connected equipment. The Matrix saves this data in memory to answer REMOTE-INFO requests.	**COMMAND** **#MODEL?** **<CR>** **FEEDBACK** **~nn@MODEL** model_name**<CR><LF>**	**model_name** – String of up to 19 printable ASCII chars	Get the device model: **#MODEL?** **<CR>**
MUTE	Set audio mute.	**COMMAND** **#MUTE** out_index,mute_mode**<CR>** **FEEDBACK** **~nn@MUTE** out_index,mute_mode**<CR><LF>**	**out_index** – Number that indicates the specific output: 1 – AUDIO OUTPUT **mute_mode** – On/Off 0 – Off 1 – On	Set AUDIO OUTPUT to mute: **#MUTE** 1,1**<CR>**
MUTE?	Get audio mute.	**COMMAND** **#MUTE?** out_index**<CR>** **FEEDBACK** **~nn@MUTE** out_index,mute_mode**<CR><LF>**	**out_index** – Number that indicates the specific output: 1 – AUDIO OUTPUT **mute_mode** – On/Off 0 – Off 1 – On	Get mute status of AUDIO OUTPUT **#MUTE** 1?**<CR>**
PROT-VER?	Get device protocol version.	**COMMAND** **#PROT-VER?** **<CR>** **FEEDBACK** **~nn@PROT-VER** 3000:version**<CR><LF>**	**version** – XX.XX where X is a decimal digit	Get the device protocol version: **#PROT-VER?** **<CR>**
RELAY-STATE	Set relay state.	**COMMAND** **#RELAY-STATE** relay_id,state**<CR>** **FEEDBACK** **~nn@RELAY-STATE** relay_id,state**<CR><LF>**	**relay_id** – Relay number 1 **state** – Relay state 0 – (open) 1 – (close)	Set relay to closed: **#RELAY-STATE** 1,1**<CR>**
RELAY-STATE?	Get relay state.	**COMMAND** **#RELAY-STATE?** relay_id**<CR>** **FEEDBACK** **~nn@RELAY-STATE** relay_id,relay_state**<CR><LF>**	**relay_id** – Relay number 1 **relay_state** – Relay state 0 – (open) 1 – (close)	Get relay state: **#RELAY-STATE?** 1**<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.	**COMMAND** **#RESET<CR>** **FEEDBACK** **~nn@RESET** ok**<CR><LF>**		Reset the device: **#RESET<CR>**
ROUTE	Set layer routing. ⓘ This command replaces all other routing commands.	**COMMAND** **#ROUTE** layer_type,out_index,in_index**<CR>** **FEEDBACK** **~nn@ROUTE** layer_type,out_index**<CR><LF>**	**layer_type** Layer Enumeration 1 – Video **out_index** (both selections are identical) 1 – HDMI OUTPUT * – ALL **in_index** – Source id 1 – HDBT INPUT 2 – HDMI INPUT	Route video HDBT INPUT to HDMI OUTPUT: **#ROUTE 1,1,1<CR>**
ROUTE?	Get layer routing. ⓘ This command replaces all other routing commands.	**COMMAND** **#ROUTE?** layer_type,out_index**<CR>** **FEEDBACK** **~nn@ROUTE** layer_type,out_index,in_index**<CR><LF>**	**layer_type** Layer Enumeration 1 – Video **out_index** (both selections are identical) 1 – HDMI OUTPUT * – ALL **in_index** – Source id 1 – HDBT INPUT 2 – HDMI INPUT	Get the layer routing: **#ROUTE?** 1,*<CR>
SCLR-AS	Set auto-sync features. ⓘ Sets the auto sync features for the selected scaler.	**COMMAND** **#SCLR-AS** scaler_index,sync_speed**<CR>** **FEEDBACK** **~nn@SCLR-AS** scaler_index,sync_speed**<CR><LF>**	**scaler_index** – Scaler Number: 1 – Scaler1 **sync_speed** – 0, 1 or 2 0 – off 1 – fast 2 – slow	Set auto-sync feature to fast: **#SCLR-AS** 1,1**<CR>**
SCLR-AS?	Get auto-sync features. ⓘ Gets the auto sync features for the selected scaler.	**COMMAND** **#SCLR-AS?** scaler_index**<CR>** **FEEDBACK** **~nn@SCLR-AS** scaler_index,sync_speed**<CR><LF>**	**scaler_index** – Scaler Number: 1 – Scaler1 **sync_speed** – 0, 1 or 2 0 – off 1 – fast 2 – slow	Get auto-sync features: **#SCLR-AS?** 1**<CR>**

Function	Description	Syntax	Parameters/Attributes	Example
SCLR-AUDIO-DELAY	Set the scaler audio delay. ⓘ Sets the audio delay for the selected audio output.	**COMMAND** **#SCLR-AUDIO-DELAY** scaler_index,delay**<CR>** **FEEDBACK** **~nn@SCLR-AUDIO-DELAY** scaler_index,delay**<CR><LF>**	**scaler_index** – Audio output number 1 – Scaler1 **delay** – 0 – Off 1 – 40ms 2 – 110ms 3 – 150ms	Set the scaler audio delay 40ms: **#SCLR-AUDIO-DELAY** 1,1**<CR>**
SCLR-AUDIO-DELAY?	Get the scaler audio delay. ⓘ Gets the audio delay for the selected audio output.	**COMMAND** **#SCLR-AUDIO-DELAY?** scaler_index**<CR>** **FEEDBACK** **~nn@SCLR-AUDIO-DELAY** scaler_index,delay**<CR><LF>**	**scaler_index** – Audio output number 1 – Scaler1 **delay** – 0 – Off 1 – 40ms 2 – 110ms 3 – 150ms	Get the scaler audio delay: **#SCLR-AUDIO-DELAY?** 1**<CR>**
SIGNAL?	Get input signal status.	**COMMAND** **#SIGNAL?** in_index**<CR>** **FEEDBACK** **~nn@SIGNAL** in_index,status**<CR><LF>**	**in_index** – Number that indicates the specific input: 1 – HDBT INPUT 2 – HDMI INPUT **status** – Signal status according to signal validation: 0 – Signal or sink is not valid (Off) 1 – Signal or sink is valid (On)	Get the input signal lock status of HDBT INPUT 1: **#SIGNAL?** 1**<CR>**
SN?	Get device serial number.	**COMMAND** **#SN?** **<CR>** **FEEDBACK** **~nn@SN** serial_num**<CR><LF>**	**serial_num** – 14 decimal digits, factory assigned	Get the device serial number: **#SN?** **<CR>**
VERSION?	Get firmware version number.	**COMMAND** **#VERSION?** **<CR>** **FEEDBACK** **~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>**
VFRZ	Set freeze on selected output.	**COMMAND** **#VFRZ** out_index,freeze_flag**<CR>** **FEEDBACK** **~nn@VFRZ** out_index,freeze_flag**<CR><LF>**	**out_index** – Number that indicates the specific output: 1 – HDMI OUTPUT **freeze_flag** – On/Off 0 – Off 1 – On	Set freeze on HDMI OUTPUT: **#VFRZ** 1,1**<CR>**
VFRZ?	Get output freeze status.	**COMMAND** **#VFRZ?** out_index**<CR>** **FEEDBACK** **~nn@VFRZ** out_index,freeze_flag**<CR><LF>**	**out_index** – Number that indicates the specific output: 1 – HDMI OUTPUT **freeze_flag** – On/Off 0 – Off 1 – On	Get output freeze status: **#VFRZ?** 1**<CR>**
VID-RES	Set output resolution. ⓘ "Set" command with is_native=ON sets native resolution on selected output (resolution index sent = 0). Device sends as answer actual VIC ID of native resolution. To use "custom resolutions" (entries 100-105 In View Modes), define them using the DEF-RES command.	**COMMAND** **#VID-RES** io_mode,io_index,is_native,resolution**<CR>** **FEEDBACK** **~nn@VID-RES** io_mode,io_index,is_native,resolution**<CR><LF>**	**io_mode** – Input/Output 1 – Output **io_index** – Number that indicates the specific output port: 1 – HDMI OUTPUT **is_native** – Native resolution flag 1 – On **resolution** – Resolution index 0=Native 1=640x480@60 2=800x600@60 3=1024x768@60 4=1280x768@60 5=1280x800@60 6=1280x1024@60 7=1360x768@60 8=1400x1050@60 9=1440x900@60 10=1600x1200@60 11=1680x1050@60 12=1920x1200@60 RB 13=2560x1600@60 RB 14=1920x1080@60 15=1280x720@60 16=2048x1080@50 17=2048x1080@60 18=2560x1440@60 RB 19=3440x1440@30 20=3440x1440@60 21=720x480P@60 22=720x576P@50 23=1280x720P@50 24=1280x720P@60 25=1920x1080P@24 26=1920x1080P@25 27=1920x1080P@30 28=1920x1080P@50 29=1920x1080P@60 30=2560x1080P@50 31=2560x1080P@60 32=3840x2160P@24 33=3840x2160P@25 34=3840x2160P@30 35=3840x2160P@50 36=3840x2160P@60	Set HDMI OUTPUT resolution to 1440x900: **#VID-RES** 1,1,1,9**<CR>**

Function	Description	Syntax	Parameters/Attributes	Example
VID-RES?	Get output resolution. ⓘ "Get" command with is_native=ON returns native resolution VIC, with is_native=OFF returns current resolution. To use "custom resolutions" (entries 100-105 In View Modes), define them using the DEF-RES command.	**COMMAND** **#VID-RES?** io_mode,io_index,is_native**<CR>** **FEEDBACK** **~nn@VID-RES?** io_mode,io_index,is_native,resolution**<CR><LF>**	**io_mode** – Input/Output 1 – Output **io_index** – Number that indicates the specific output port: 1 – HDMI OUTPUT **is_native** – Native resolution flag 1 – On **resolution** – Resolution index 0=Native 1=640x480@60 2=800x600@60 3=1024x768@60 4=1280x768@60 5=1280x800@60 6=1280x1024@60 7=1360x768@60 8=1400x1050@60 9=1440x900@60 10=1600x1200@60 11=1680x1050@60 12=1920x1200@60 RB 13=2560x1600@60 RB 14=1920x1080@60 15=1280x720@60 16=2048x1080@50 17=2048x1080@60 18=2560x1440@60 RB 19=3440x1440@30 20=3440x1440@60 21=720x480P@60 22=720x576P@50 23=1280x720P@50 24=1280x720P@60 25=1920x1080P@24 26=1920x1080P@25 27=1920x1080P@30 28=1920x1080P@50 29=1920x1080P@60 30=2560x1080P@50 31=2560x1080P@60 32=3840x2160P@24 33=3840x2160P@25 34=3840x2160P@30 35=3840x2160P@50 36=3840x2160P@60	Set output resolution: **#VID-RES?** 1,1,1**<CR>**
VMUTE	Set enable/disable video on output. ⓘ Video mute parameter 2 (blank picture) is not supported.	**COMMAND** **#VMUTE** out_index,flag**<CR>** **FEEDBACK** **~nn@VMUTE** out_index,flag**<CR><LF>**	**out_index** – Number that indicates the specific output: 1 – HDMI OUTPUT **flag** – Video Mute 0 – Video disabled, +5V low 1 – Video enabled, +5V high 2 – Blank picture, +5V high	Disable the video output on HDMI OUTPUT: **#VMUTE** 1,0**<CR>**
VMUTE?	Get video on output status. ⓘ Video mute parameter 2 (blank picture) is not supported.	**COMMAND** **#VMUTE?** out_index**<CR>** **FEEDBACK** **~nn@VMUTE** out_index,flag**<CR><LF>**	**out_index** – Number that indicates the specific output: 1 – HDMI OUTPUT **flag** – Video Mute 0 – Video disabled, +5V low 1 – Video enabled, +5V high 2 – Blank picture, +5V high	Get video on HDMI OUTPUT status: **#VMUTE?** 1**<CR>**
X-AUD-LVL	Set audio level of a specific signal. ⓘ This is an Extended Protocol 3000 command.	**COMMAND** **#X-AUD-LVL** <direction_type>.<port_format>.<port_index>.<signal_type>.<index>,audio_level**<CR>** **FEEDBACK** **~nn@X-AUD-LVL** <direction_type>.<port_format>.<port_index>.<signal_type>.<index>,audio_level**<CR><LF>**	The following attributes comprise the signal ID: ▪ **<direction_type>** – Direction of the port: 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:1 ▪ **<signal_type>** – Signal ID attribute: o AUDIO ▪ **<index>** – Indicates a specific channel number when there are multiple channels of the same type **audio_level** – Audio level (range between 0 to +100)	Set the AUDIO OUT level to 10: **#X-AUD-LVL** out.analog_audio.1.audio.1,10**<CR>**

Function	Description	Syntax	Parameters/Attributes	Example
X-AUD-LVL?	Get audio level of a specific signal. ⓘ This is an Extended Protocol 3000 command.	**COMMAND** **#X-AUD-LVL?**␣<direction_type>.<port_format>.<port_index>.<signal_type>.<index>**<CR>** **FEEDBACK** **~nn@X-AUD-LVL**␣<direction_type>.<port_format>.<port_index>.<signal_type>.<index>,audio_level**<CR><LF>**	The following attributes comprise the signal ID: ▪ **<direction_type>** – Direction of the port: 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:1 ▪ **<signal_type>** – Signal ID attribute: o AUDIO ▪ **<index>** – Indicates a specific channel number when there are multiple channels of the same type **audio_level** – Audio level (range between 0 to +100)	Get the audio level of a specific signal: **#X-AUD-LVL?**␣out.analog_audio.1.audio.1**<CR>**
X-ROUTE	Send routing command to matrix. ⓘ It is recommended to use the command **#SIGNALS-LIST** to get the list of all signal IDs available in the system and which can be used in this command. Video 1 is the default port in this command and is implied even if not written: **#X-ROUTE**␣out.sdi.5,in.sdi.1**<CR>** is interpreted as: **#X-ROUTE**␣out.sdi.5.video.1,in.sdi.1.video.1**<CR>** This is an Extended Protocol 3000 command.	**COMMAND** **#X-ROUTE**␣<direction_type1>.<port_type1>.<port_index1>.<signal_type1>.<index1>,<direction_type2>.<port_type2>.<port_index2>.<signal_type2>.<index2>**<CR>** **FEEDBACK** **~nn@X-ROUTE**␣<direction_type1>.<port_type1>.<port_index1>.<signal_type1>.<index1>,<direction_type2>.<port_type2>.<port_index2>.<signal_type2>.<index2>**<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 HDMI o HDBT ▪ **<port_index>** – The port number as printed on the front or rear panel: For the output: 1 – HDMI OUTPUT For the input: 1 – HDBT INPUT 2 – HDMI INPUT ▪ **<signal_type>** – Signal ID attribute: o VIDEO ▪ **<index>** – Indicates a specific channel number when there are multiple channels of the same type: 1	Route HDMI INPUT to HDMI OUTPUT: **#X-ROUTE**␣out.hdmi.1.video.1,in.hdmi.2.video.1**<CR>**
X-ROUTE?	Get routing status. ⓘ It is recommended to use the command **#SIGNALS-LIST** to get the list of all signal IDs available in the system and which can be used in this command. **VIDEO.1** are the default **<signal_type>** and **<index>** in this command and are implied even if not written: **#X-ROUTE**␣out.sdi.5,in.sdi.1**<CR>** is interpreted as: **#X-ROUTE**␣out.sdi.5.video.1,in.sdi.1.video.1**<CR>** This is an Extended Protocol 3000 command.	**COMMAND** **#X-ROUTE?**␣<direction_type1>.<port_type1>.<port_index1>.<signal_type1>.<index1>**<CR>** **FEEDBACK** **~nn@X-ROUTE**␣<direction_type1>.<port_type1>.<port_index1>.<signal_type1>.<index1>,<direction_type2>.<port_type2>.<port_index2>.<signal_type2>.<index2>**<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 HDMI o HDBT ▪ **<port_index>** – The port number as printed on the front or rear panel: For the output: 1 – HDMI OUTPUT For the input: 1 – HDBT INPUT 2 – HDMI INPUT ▪ **<signal_type>** – Signal ID attribute: o VIDEO ▪ **<index>** – Indicates a specific channel number when there are multiple channels of the same type: 1	Get the routing status: **#X-ROUTE?**␣out.hdmi.1.video.1**<CR>**
X-SIGNAL?	Get input signal status. ⓘ This is an Extended Protocol 3000 command.	**COMMAND** **#X-SIGNAL?**␣<direction_type>.<port_format>.<port_index>.<signal_type>.<index>**<CR>** **FEEDBACK** **~nn@X-SIGNAL**␣<direction_type>.<port_format>.<port_index>.<signal_type>.<index>,status**<CR><LF>**	The following attributes comprise the signal ID: ▪ **<direction_type>** – Direction of the port: o IN – Input ▪ **<port_format>** – Type of signal on the port: o HDMI o HDBT ▪ **<port_index>** – The port number as printed on the front or rear panel: 1 – HDBT INPUT 2 – HDMI INPUT ▪ **<signal_type>** – Signal ID attribute: o VIDEO ▪ **<index>** – Indicates a specific channel number when there are multiple channels of the same type: 1 **status** – Input Signal Status 0 – No signal 1 – There is a signal	**#X-SIGNAL?**␣in.hdbt.1.video.1**<CR>**

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_TIMEDOUT	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 – not changed
ERR_WRONG_MODE	33	Wrong operation mode
ERR_NOT_CONFIGURED	34	Device/chip was not initialized