

VS-88H2A: Protocol 3000

Commands

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:**

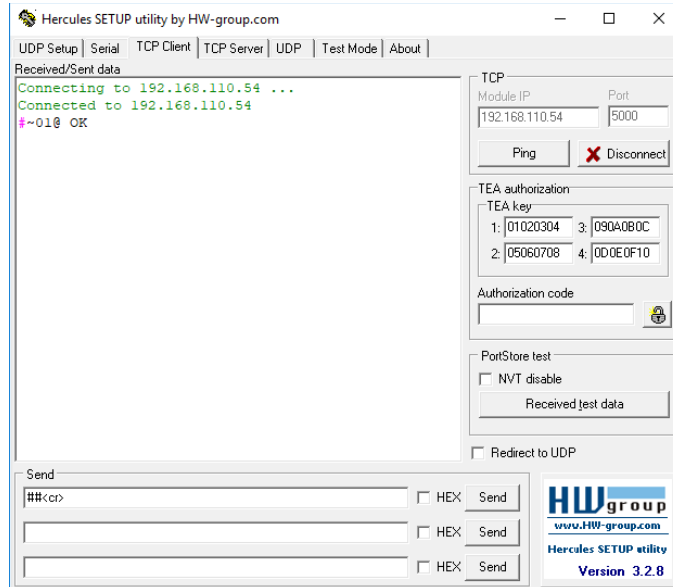
Prefix	Command Name	Constant (Space)	Parameter(s)	Suffix
#	Command	_	Parameter	<CR>

- **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-88H2A**. 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
#	<p>Protocol handshaking.</p> <p>① Validates the Protocol 3000 connection and gets the machine number.</p> <p>Step-in master products use this command to identify the availability of a device.</p>	<p>COMMAND</p> <p>#<CR></p> <p>FEEDBACK</p> <p>~nn@_ok<CR><LF></p>		#<CR>
AFV	<p>Set audio follow video/audio breakaway mode.</p> <p>① When the unit moves from breakaway to audio follow video switching mode, all audio switch settings reset according to the video switch settings.</p>	<p>COMMAND</p> <p>#AFV_afv_mode<CR></p> <p>FEEDBACK</p> <p>~nn@AFV_afv_mode<CR><LF></p>	<p>afv_mode – Front panel AFV mode</p> <p>0 – afv – sets the unit to the audio-follow-video switching mode</p> <p>1 – brk – sets the unit to the audio breakaway switching mode</p>	Set audio breakaway mode: #AFV_1<CR>
AFV?	<p>Get audio follow video mode status.</p> <p>① When the unit moves from breakaway to audio follow video switching mode, all audio switch settings reset according to the video switch settings.</p>	<p>COMMAND</p> <p>#AFV?_<CR></p> <p>FEEDBACK</p> <p>~nn@AFV_afv_mode<CR><LF></p>	<p>afv_mode – Front panel AFV mode</p> <p>0 – afv – sets the unit to the audio-follow-video switching mode</p> <p>1 – brk – sets the unit to the audio breakaway switching mode</p>	Get audio follow video mode status: #AFV?_<CR>

Function	Description	Syntax	Parameters/Attributes	Example
AUD	<p>LEGACY COMMAND. Set audio switch state.</p> <p>ⓘ When AFV switching mode is active, this command cannot switch video.</p>	<p>COMMAND</p> <pre>#AUD_in>out_id,in>out_id,...<CR></pre> <p>FEEDBACK</p> <pre>~nn@AUD_in>out_id<CR><LF> ~nn@AUD_in>out_id<CR><LF></pre>	<p>in – Input number 0 – disconnect output 1 – HDMI IN 1 2 – HDMI IN 2 3 – HDMI IN 3 4 – HDMI IN 4 5 – HDMI IN 5 6 – HDMI IN 6 7 – HDMI IN 7 8 – HDMI IN 8 9 – Analog IN 1 10 – Analog IN 2 11 – Analog IN 3 12 – Analog IN 4 13 – Analog IN 5 14 – Analog IN 6 15 – Analog IN 7 16 – Analog IN 8</p> <p>> – Connection character between in and out parameters</p> <p>out_id – Output number * – All outputs 1 – HDMI OUT 1 2 – HDMI OUT 2 3 – HDMI OUT 3 4 – HDMI OUT 4 5 – HDMI OUT 5 6 – HDMI OUT 6 7 – HDMI OUT 7 8 – HDMI OUT 8 9 – Analog OUT 1 10 – Analog OUT 2 11 – Analog OUT 3 12 – Analog OUT 4 13 – Analog OUT 5 14 – Analog OUT 6 15 – Analog OUT 7 16 – Analog OUT 8</p>	<p>Switch embedded audio HDMI IN 1 to HDMI OUT 3: #AUD_1>3<CR></p>
AUD?	<p>LEGACY COMMAND. Get audio switch state.</p> <p>ⓘ When AFV switching mode is active, this command cannot switch video.</p>	<p>COMMAND</p> <pre>#AUD?_out_id<CR> #AUD?_*<CR></pre> <p>FEEDBACK</p> <pre>~nn@AUD_in>out_id<CR><LF> ~nn@AUD_in>1,in>2,...<CR><LF></pre>	<p>in – Input number 1 – HDMI IN 1 2 – HDMI IN 2 3 – HDMI IN 3 4 – HDMI IN 4 5 – HDMI IN 5 6 – HDMI IN 6 7 – HDMI IN 7 8 – HDMI IN 8 9 – Analog IN 1 10 – Analog IN 2 11 – Analog IN 3 12 – Analog IN 4 13 – Analog IN 5 14 – Analog IN 6 15 – Analog IN 7 16 – Analog IN 8</p> <p>> – Connection character between in and out parameters</p> <p>out_id – Output number * – All outputs 1 – HDMI OUT 1 2 – HDMI OUT 2 3 – HDMI OUT 3 4 – HDMI OUT 4 5 – HDMI OUT 5 6 – HDMI OUT 6 7 – HDMI OUT 7 8 – HDMI OUT 8 9 – Analog OUT 1 10 – Analog OUT 2 11 – Analog OUT 3 12 – Analog OUT 4 13 – Analog OUT 5 14 – Analog OUT 6 15 – Analog OUT 7 16 – Analog OUT 8</p>	<p>Get audio switch state for HDMI OUT 3: #AUD?_3<CR></p>
AUD-LVL	<p>Set volume level.</p>	<p>COMMAND</p> <pre>#AUD-LVL_io_mode,io_index,vol_level<CR></pre> <p>FEEDBACK</p> <pre>~nn@AUD-LVL_io_mode,io_index,vol_level<CR><LF></pre>	<p>io_mode – Output 2 – Output</p> <p>io_index – Number that indicates the specific output port: 1 – Analog OUT 1 2 – Analog OUT 2 3 – Analog OUT 3 4 – Analog OUT 4 5 – Analog OUT 5 6 – Analog OUT 6 7 – Analog OUT 7 8 – Analog OUT 8</p> <p>vol_level – Volume level 0 to 100%; ++ (increase current value by 1%); -- (decrease current value by 1%)</p>	<p>Set analog audio OUT 1 level to -50dB: #AUD-LVL_2,1,50<CR></p>

Function	Description	Syntax	Parameters/Attributes	Example
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 – Output 2 – Output io_index – Number that indicates the specific output port: 1 – Analog OUT 1 2 – Analog OUT 2 3 – Analog OUT 3 4 – Analog OUT 4 5 – Analog OUT 5 6 – Analog OUT 6 7 – Analog OUT 7 8 – Analog OUT 8 vol_level – Volume level 0 to 100%.	Get analog OUT 8 level: #AUD-LVL?_2,8<CR>
AV	Switch audio and video.	COMMAND #AV_in>out_id,in>out_id,...<CR> FEEDBACK ~nn@AV_in>out_id,in>out_id,...<CR><LF>	in – Number that indicates the specific input: 0 – disconnect output 1 – HDMI IN 1 2 – HDMI IN 2 3 – HDMI IN 3 4 – HDMI IN 4 5 – HDMI IN 5 6 – HDMI IN 6 7 – HDMI IN 7 8 – HDMI IN 8 > – Connection character between in and out parameters out_id – Output number * – All outputs 1 – HDMI OUT 1 2 – HDMI OUT 2 3 – HDMI OUT 3 4 – HDMI OUT 4 5 – HDMI OUT 5 6 – HDMI OUT 6 7 – HDMI OUT 7 8 – HDMI OUT 8	Switch IN 1 to OUT 4: #AV_1>4<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 OUT 1 2 – HDMI OUT 2 3 – HDMI OUT 3 4 – HDMI OUT 4 5 – HDMI OUT 5 6 – HDMI OUT 6 7 – HDMI OUT 7 8 – HDMI OUT 8 connection_mode – Connection mode 0 – manual 1 – priority switch 2 – last connected switch	Set input auto switch mode (per output) for HDMI 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 OUT 1 2 – HDMI OUT 2 3 – HDMI OUT 3 4 – HDMI OUT 4 5 – HDMI OUT 5 6 – HDMI OUT 6 7 – HDMI OUT 7 8 – HDMI OUT 8 connection_mode – Connection mode 0 – manual 1 – priority switch 2 – last connected switch	Get the input audio switch mode for HDMI Out: #AV-SW-MODE?_1,1<CR>
AV-SW-TIMEOUT	Set auto switching timeout.	COMMAND #AV-SW-TIMEOUT_switching_mode,time_out<CR> FEEDBACK ~nn@AV-SW-TIMEOUT_switching_mode,time_out<CR><LF>	switching_mode – Switching mode 0 – Video signal lost 4 – Disable 5V on video output if no input signal detected time_out – Timeout in seconds 0 - 999	Set the auto switching timeout to 5 seconds in the event of 5V disable when no input signal is detected: #AV-SW-TIMEOUT_4,5<CR>
AV-SW-TIMEOUT?	Get auto switching timeout.	COMMAND #AV-SW-TIMEOUT?_switching_mode<CR> FEEDBACK ~nn@AV-SW-TIMEOUT_switching_mode,time_out<CR><LF>	switching_mode – Switching mode 0 – Video signal lost 4 – Disable 5V on video output if no input signal detected time_out – Timeout in seconds 0 - 999	Get the Disable 5V on video output if no input signal detected timeout: #AV-SW-TIMEOUT?_4<CR>

Function	Description	Syntax	Parameters/Attributes	Example
BALANCE	Set balance level.	COMMAND #BALANCE_out_index,balance_level<CR> FEEDBACK ~nn@BALANCE_out_index,balance_level<CR><LF>	out_index – Number that indicates the specific output: 1 – Analog OUT 1 2 – Analog OUT 2 3 – Analog OUT 3 4 – Analog OUT 4 5 – Analog OUT 5 6 – Analog OUT 6 7 – Analog OUT 7 8 – Analog OUT 8 balance_level – 0 to 100; ++ increase current value – decrease current value	Set the speaker output balance to +12: #BALANCE_1,12<CR>
BALANCE?	Get balance level.	COMMAND #BALANCE?_out_index<CR> FEEDBACK ~nn@BALANCE_out_index,balance_level<CR><LF>	out_index – Number that indicates the specific output: 1 – Analog OUT 1 2 – Analog OUT 2 3 – Analog OUT 3 4 – Analog OUT 4 5 – Analog OUT 5 6 – Analog OUT 6 7 – Analog OUT 7 8 – Analog OUT 8 balance_level – 0 to 100 ++ increase current value – decrease current value	Get balance level for channel 1: #BALANCE?_1<CR>
BAUD	Set protocol serial port baud rate. ① 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 (if ERR 002 is returned, the default baud rate is used).	COMMAND #BAUD_baud_rate<CR> FEEDBACK ~nn@BAUD_baud_rate<CR><LF> Option 1: ~nn@BAUD_current_baud_rate<CR><LF> Option 2: ~nn@BAUD_baud_rate1,baud_rate2,...<CR><LF>	baud_rate – 9600 / 115200 / else - new baud rate to set current_baud_rate – 9600 / 115200 / else - current protocol serial port baud rate baud_param – 0 - get the list of supported baud rates baud_rate1,baud_rate2, ... – List of supported baud rates	Set the baud rate to 9600: #BAUD_9600<CR>
BAUD?	Get protocol serial port baud rate. (Option 1 - for current baud rate. Option 2 - for list of supported baud rates). ① 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 (if ERR 002 is returned, the default baud rate is used).	COMMAND #BAUD?_<CR> #BAUD?_baud_param<CR> FEEDBACK ~nn@BAUD_baud_rate<CR><LF> Option 1: ~nn@BAUD_current_baud_rate<CR><LF> Option 2: ~nn@BAUD_baud_rate1,baud_rate2,...<CR><LF>	baud_rate – 9600 / 115200 / else - new baud rate to set current_baud_rate – 9600 / 115200 / else - current protocol serial port baud rate baud_param – 0 - get the list of supported baud rates baud_rate1, - Baud_rate2, ... - list of supported baud rates	Get protocol serial port baud rate: #BAUD?_<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>

Function	Description	Syntax	Parameters/Attributes	Example
CPEDID	<p>Copy EDID data from the output to the input EEPROM.</p> <p> 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>COMMAND</p> <pre>#CPEDID_edid_io,src_id,dst_type,dest_bitmap<CR></pre> <p>or</p> <pre>#CPEDID_edid_io,src_id,dst_type,dest_bitmap,safe_mode<CR></pre> <p>FEEDBACK</p> <pre>~nn@CPEDID_edid_io,src_id,dst_type,dest_bitmap<CR><LF></pre> <pre>~nn@CPEDID_edid_io,src_id,dst_type,dest_bitmap,safe_mode<CR><LF></pre>	<p>edid_io – EDID source type (usually output)</p> <ul style="list-style-type: none"> 0 – Input 1 – Output 2 – Default EDID <p>src_id – Number of chosen source stage</p> <p>For input source:</p> <ul style="list-style-type: none"> 1 – HDMI IN 1 2 – HDMI IN 2 3 – HDMI IN 3 4 – HDMI IN 4 5 – HDMI IN 5 6 – HDMI IN 6 7 – HDMI IN 7 8 – HDMI IN 8 <p>For output source:</p> <ul style="list-style-type: none"> 0 – Default EDID source 1 – HDMI OUT 1 2 – HDMI OUT 2 3 – HDMI OUT 3 4 – HDMI OUT 4 5 – HDMI OUT 5 6 – HDMI OUT 6 7 – HDMI OUT 7 8 – HDMI OUT 8 <p>dst_type – EDID destination type (usually input)</p> <ul style="list-style-type: none"> 0 – Input <p>dest_bitmap – Bitmap representing destination IDs. Format: XXXX...X, where X is hex digit. The binary form of every hex digit represents corresponding destinations.</p> <ul style="list-style-type: none"> 0 – indicates that EDID data is not copied to this destination. 1 – indicates that EDID data is copied to this destination. <p>safe_mode – Safe mode</p> <ul style="list-style-type: none"> 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>Copy the EDID data from the Output 1 (EDID source) to the Input:</p> <pre>#CPEDID_1,1,0,0x1<CR></pre> <p>Copy the EDID data from the default EDID source to the Input:</p> <pre>#CPEDID_2,0,0,0x1<CR></pre>
DIR	List files in device.	<p>COMMAND</p> <pre>#DIR<CR></pre> <p>FEEDBACK</p> <p>Multi-line:</p> <pre>~nn@DIR<CR><LF></pre> <pre>file_name TABfile_size_bytes,id:_file_id<CR><LF></pre> <pre>TABfree_size_bytes.<CR><LF></pre>	<p>file_name – Name of file</p> <p>file_size – File size in bytes. A file can take more space on device memory</p> <p>file_id – Internal ID for file in file system</p> <p>free_size – Free space in bytes in device file system</p>	#DIR<CR>
DISPLAY?	Get output HPD status.	<p>COMMAND</p> <pre>#DISPLAY?_out_index<CR></pre> <p>FEEDBACK</p> <pre>~nn@DISPLAY_out_index,status<CR><LF></pre>	<p>out_index – Number that indicates the specific output:</p> <ul style="list-style-type: none"> 1 – HDMI OUT 1 2 – HDMI OUT 2 3 – HDMI OUT 3 4 – HDMI OUT 4 5 – HDMI OUT 5 6 – HDMI OUT 6 7 – HDMI OUT 7 8 – HDMI OUT 8 <p>status – HPD status according to signal validation</p> <ul style="list-style-type: none"> 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 OUT 1:
DPSW-STATUS? N/A	Get the DIP-switch state.	<p>COMMAND</p> <pre>#DPSW-STATUS?_dip_id<CR></pre> <p>FEEDBACK</p> <pre>~nn@DPSW-STATUS_dip_id,status<CR><LF></pre>	<p>dip_id – 1 to 8 (number of DIP switches)</p> <p>status – Up/down</p> <ul style="list-style-type: none"> 0 – Up 1 – Down 	get the DIP-switch 2 status:
ETH-PORT	Set Ethernet port protocol.	<p>COMMAND</p> <pre>#ETH-PORT_port_type,port_id<CR></pre> <p>FEEDBACK</p> <pre>~nn@ETH-PORT_port_type,port_id<CR><LF></pre>	<p>port_type – TCP/UDP</p> <p>port_id – TCP/UDP port number (2000 – 65535)</p>	Set the Ethernet port protocol for TCP to port 12457:
ETH-PORT?	Get Ethernet port protocol.	<p>COMMAND</p> <pre>#ETH-PORT?_port_type<CR></pre> <p>FEEDBACK</p> <pre>~nn@ETH-PORT_port_type,port_id<CR><LF></pre>	<p>port_type – TCP/UDP</p> <ul style="list-style-type: none"> 0 – TCP 1 – UDP <p>port_id – TCP / UDP port number (2000 – 65535)</p>	Get the Ethernet port protocol for UDP:

Function	Description	Syntax	Parameters/Attributes	Example
FACTORY	<p>Reset device to factory default configuration.</p> <p>i 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>	COMMAND #FACTORY<CR> FEEDBACK ~nn@FACTORY_ok<CR><LF>		Reset the device to factory default configuration: #FACTORY<CR>
FPGA-VER?	<p>Get current FPGA version.</p>	COMMAND #FPGA-VER?_fpga_id<CR> FEEDBACK ~nn@FPGA-VER_fpga_id,expected_ver,ver<CR><LF>	fpga_id – FPGA id 1 expected_ver – Expected FPGA version for current firmware ver – Actual FPGA version	Get current FPGA version: #FPGA-VER?_1<CR>
GEDID	<p>Get EDID support on certain input/output.</p> <p>i For old devices that do not support this command, ~nn@ERR 002<CR><LF> is received.</p>	COMMAND #GEDID_io_mode,in_index<CR> FEEDBACK ~nn@GEDID_io_mode,in_index,size<CR><LF>	io_mode – Input/Output 0 – Input 1 – Output 2 – Default EDID in_index – Number that indicates the specific input: 1 – HDMI IN 1 2 – HDMI IN 2 3 – HDMI IN 3 4 – HDMI IN 4 5 – HDMI IN 5 6 – HDMI IN 6 7 – HDMI IN 7 8 – HDMI IN 8 size – Size of data to be sent from device, 0 means no EDID support	Get EDID support information for input 1: #GEDID_0,1<CR>
HDCP-MOD	<p>Set HDCP mode.</p> <p>i Set HDCP working mode on the device input:</p> <p>HDCP supported - HDCP_ON [default].</p> <p>HDCP not supported - HDCP OFF.</p>	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 – HDMI IN 1 2 – HDMI IN 2 3 – HDMI IN 3 4 – HDMI IN 4 5 – HDMI IN 5 6 – HDMI IN 6 7 – HDMI IN 7 8 – HDMI IN 8 mode – HDCP mode: 0 – HDCP Off 1 – HDCP On	Set the input HDCP-MODE of HDMI IN 1 to Off: #HDCP-MOD_1,0<CR>
HDCP-MOD?	<p>Get HDCP mode.</p> <p>i Set HDCP working mode on the device input:</p> <p>HDCP supported - HDCP_ON [default].</p> <p>HDCP not supported - HDCP OFF.</p>	COMMAND #HDCP-MOD?_in_index<CR> FEEDBACK ~nn@HDCP-MOD_in_index,mode<CR><LF>	in_index – Number that indicates the specific input: 1 – HDMI IN 1 2 – HDMI IN 2 3 – HDMI IN 3 4 – HDMI IN 4 5 – HDMI IN 5 6 – HDMI IN 6 7 – HDMI IN 7 8 – HDMI IN 8 mode – HDCP mode: 0 – HDCP Off 1 – HDCP On	Get the input HDCP-MODE of HDMI IN 1: #HDCP-MOD?_1<CR>
HDCP-STAT?	<p>Get HDCP signal status.</p> <p>i 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>	COMMAND #HDCP-STAT?_io_mode,in_index<CR> FEEDBACK ~nn@HDCP-STAT_io_mode,in_index,status<CR><LF>	io_mode – Input/Output 0 – Input 1 – Output in_index – Number that indicates the specific input: 1 – HDMI IN 1 2 – HDMI IN 2 3 – HDMI IN 3 4 – HDMI IN 4 5 – HDMI IN 5 6 – HDMI IN 6 7 – HDMI IN 7 8 – HDMI IN 8 status – Signal encryption status - valid values On/Off 0 – HDCP Off 1 – HDCP On	Get the output HDCP-STATUS of HDMI IN 1: #HDCP-STAT?_0,1<CR>
HELP	<p>Get command list or help for specific command.</p>	COMMAND #HELP<CR> FEEDBACK 1. Multi-line: ~nn@Device_cmd_name,_cmd_name...<CR><LF>	cmd_name – Name of a specific command	Get the command list: #HELP<CR>
IDV	<p>Set visual indication from device.</p> <p>i Using this command, some devices can light a sequence of buttons or LEDs to allow identification of a specific device from similar devices.</p>	COMMAND #IDV<CR> FEEDBACK ~nn@IDV_ok<CR><LF>		#IDV<CR>

Function	Description	Syntax	Parameters/Attributes	Example												
INFO-IO?	LEGACY COMMAND. Get in/out count.	COMMAND #INFO-IO?_<CR> FEEDBACK ~nn@INFO-IO_IN_in_count,OUT_out_count<CR><LF>	in_count – Number of inputs in the unit out_count – Number of outputs in the unit	Get inputs count: #INFO-IO?_<CR>												
INFO-PRST?	LEGACY COMMAND. 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.	COMMAND #INFO-PRST?_<CR> FEEDBACK ~nn@INFO-PRST_vid_video_preset_count,audio_audio_preset_count<CR><LF>	video_preset_count – Maximum number of video presets in the unit audio_preset_count – Maximum number of audio presets in the unit	Get number of video and audio presets: #INFO-PRST?_<CR>												
LABEL	Set input/output label.	COMMAND #LABEL_io_mode,io_index,switch,label_txt<CR> FEEDBACK ~nn@LABEL_io_mode,io_index,switch,label_txt<CR><LF>	io_mode – Input/Output 0 – Input 1 – Output io_index – Number that indicates the specific input or output port: 1 – HDMI IN 1 2 – HDMI IN 2 3 – HDMI IN 3 4 – HDMI IN 4 5 – HDMI IN 5 6 – HDMI IN 6 7 – HDMI IN 7 8 – HDMI IN 8 1 – HDMI OUT 1 2 – HDMI OUT 2 3 – HDMI OUT 3 4 – HDMI OUT 4 5 – HDMI OUT 5 6 – HDMI OUT 6 7 – HDMI OUT 7 8 – HDMI OUT 8 switch – On/Off (enable/disable) custom label label_txt – Custom label string	Set input/output label: #LABEL_1,1,1,1<CR>												
LABEL?	Get input/output label.	COMMAND #LABEL?_io_mode,io_index<CR> FEEDBACK ~nn@LABEL_io_mode,io_index,switch,label_txt<CR><LF>	io_mode – Input/Output 0 – Input 1 – Output io_index – Number that indicates the specific input or output port: 1 – HDMI IN 1 2 – HDMI IN 2 3 – HDMI IN 3 4 – HDMI IN 4 5 – HDMI IN 5 6 – HDMI IN 6 7 – HDMI IN 7 8 – HDMI IN 8 1 – HDMI OUT 1 2 – HDMI OUT 2 3 – HDMI OUT 3 4 – HDMI OUT 4 5 – HDMI OUT 5 6 – HDMI OUT 6 7 – HDMI OUT 7 8 – HDMI OUT 8 switch – On/Off (enable/disable) custom label label_txt – Custom label string	Get input/output label: #LABEL?_0,1<CR>												
LOAD	Load file to device.	COMMAND #LOAD_file_name,size<CR> FEEDBACK 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>	file_name – Name of file to save on device size – Size of file data that is sent Using the Packet Protocol 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: Packet ID (1, 2, 3...) (2 bytes in length) Length (data length + 2 for CRC) – (2 bytes in length) Data (data length -2 bytes) CRC – 2 bytes <table border="1"><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			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													

Function	Description	Syntax	Parameters/Attributes	Example
LOCK-FP	<p>Lock the front panel.</p> <p>i In NT-52N, this command includes the PortNumber (1-2) parameter.</p>	<p>COMMAND</p> <pre>#LOCK-FP_lock/unlock<CR></pre> <p>FEEDBACK</p> <pre>~nn@LOCK-FP_lock/unlock<CR><LF></pre>	<p>lock/unlock – On/Off 0 – Off unlocks EDID 1 – On locks EDID</p>	<p>Unlock front panel: #LOCK-FP_0<CR></p>
LOCK-FP?	<p>Get the front panel lock state.</p> <p>i In NT-52N, this command includes the PortNumber (1-2) parameter.</p>	<p>COMMAND</p> <pre>#LOCK-FP?_<CR></pre> <p>FEEDBACK</p> <pre>~nn@LOCK-FP_lock/unlock<CR><LF></pre>	<p>lock/unlock – On/Off 0 – Off unlocks EDID 1 – On locks EDID</p>	<p>Get the front panel lock state: #LOCK-FP?<CR></p>
LOGIN	<p>Set protocol permission.</p> <p>i The permission system works only if security is enabled with the "SECUR" command.</p> <p>LOGIN allows the user to run commands with an End User or Administrator permission level. When the permission system is enabled, LOGIN enables running commands with the User or Administrator permission level. When set, login must be performed upon each connection.</p> <p>It is not mandatory to enable the permission system in order to use the device.</p> <p>In each device, some connections allow logging in to different levels. Some do not work with security at all.</p> <p>Connection may logout after timeout.</p>	<p>COMMAND</p> <pre>#LOGIN_login_level,password<CR></pre> <p>FEEDBACK</p> <pre>~nn@LOGIN_login_level,password_ok<CR><LF></pre> <p>or</p> <pre>~nn@LOGIN_err_004<CR><LF></pre> <p>(if bad password entered)</p>	<p>login_level – Level of permissions required (User or Admin) password – Predefined password (by PASS command). Default password is an empty string</p>	<p>Set the protocol permission level to Admin (when the password defined in the PASS command is 33333): #LOGIN_admin,33333<CR></p>
LOGIN?	<p>Get current protocol permission level.</p> <p>i The permission system works only if security is enabled with the "SECUR" command.</p> <p>For devices that support security, LOGIN allows the user to run commands with an End User or Administrator permission level.</p> <p>In each device, some connections allow logging in to different levels. Some do not work with security at all.</p> <p>Connection may logout after timeout.</p>	<p>COMMAND</p> <pre>#LOGIN?_<CR></pre> <p>FEEDBACK</p> <pre>~nn@LOGIN_login_level<CR><LF></pre>	<p>login_level – Level of permissions required (User or Admin)</p>	<p>Get current protocol permission level: #LOGIN?<CR></p>
LOGOUT	<p>Cancel current permission level.</p> <p>i Logs out from End User or Administrator permission levels to Not Secure.</p>	<p>COMMAND</p> <pre>#LOGOUT<CR></pre> <p>FEEDBACK</p> <pre>~nn@LOGOUT_ok<CR><LF></pre>		<p>#LOGOUT<CR></p>
MODEL?	<p>Get device model.</p> <p>i This command identifies equipment connected to VS-88H2A and notifies of identity changes to the connected equipment. The Matrix saves this data in memory to answer REMOTE-INFO requests.</p>	<p>COMMAND</p> <pre>#MODEL?_<CR></pre> <p>FEEDBACK</p> <pre>~nn@MODEL_model_name<CR><LF></pre>	<p>model_name – String of up to 19 printable ASCII chars</p>	<p>Get the device model: #MODEL?_<CR></p>




Function	Description	Syntax	Parameters/Attributes	Example
MTX-MODE	LEGACY COMMAND. Set auto-switch mode. ⓘ Not recommended for new devices.	COMMAND #MTX-MODE_<out_id>,<connection_mode><CR> FEEDBACK ~nn@MTX-MODE_<out_id>,<connection_mode><CR><LF>	out_id – number of system outputs * – All outputs 1 – HDMI OUT 1 2 – HDMI OUT 2 3 – HDMI OUT 3 4 – HDMI OUT 4 5 – HDMI OUT 5 6 – HDMI OUT 6 7 – HDMI OUT 7 8 – HDMI OUT 8 connection_mode – Connection mode 0 – manual 1 – auto priority 2 – auto last connected	Set output to last connected: #MTX-MODE_1,2<CR>
MTX-MODE?	LEGACY COMMAND. Get auto-switch mode. ⓘ Not recommended for new devices.	COMMAND #MTX-MODE?_<out_id><CR> FEEDBACK ~nn@MTX-MODE_<out_id>,<connection_mode><CR><LF>	out_id – number of system outputs 1 – HDMI OUT 1 2 – HDMI OUT 2 3 – HDMI OUT 3 4 – HDMI OUT 4 5 – HDMI OUT 5 6 – HDMI OUT 6 7 – HDMI OUT 7 8 – HDMI OUT 8 connection_mode – Connection mode 0 – manual 1 – auto priority 2 – auto last connected	Get auto-switch mode: #MTX-MODE?_?<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 – Analog OUT 1 2 – Analog OUT 2 3 – Analog OUT 3 4 – Analog OUT 4 5 – Analog OUT 5 6 – Analog OUT 6 7 – Analog OUT 7 8 – Analog OUT 8 mute_mode – On/Off 0 – Off 1 – On	Set Output 1 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 – Analog OUT 1 2 – Analog OUT 2 3 – Analog OUT 3 4 – Analog OUT 4 5 – Analog OUT 5 6 – Analog OUT 6 7 – Analog OUT 7 8 – Analog OUT 8 mute_mode – On/Off 0 – Off 1 – On	Get mute status of output 1 #MUTE_1?<CR>
NAME	Set machine (DNS) name. ⓘ 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).	COMMAND #NAME_<machine_name><CR> FEEDBACK ~nn@NAME_<machine_name><CR><LF>	machine_name – String of up to 14 alpha-numeric chars (can include hyphen, not at the beginning or end)	Set the DNS name of the device to room-442: #NAME_?room-442<CR>
NAME?	Get machine (DNS) name. ⓘ 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).	COMMAND #NAME?_<CR> FEEDBACK ~nn@NAME_<machine_name><CR><LF>	machine_name – String of up to 14 alpha-numeric chars (can include hyphen, not at the beginning or end)	Get the DNS name of the device: #NAME?_<CR>
NAME-RST	Reset machine (DNS) name to factory default. ⓘ Factory default of machine (DNS) name is "KRAMER_" + 4 last digits of device serial number.	COMMAND #NAME-RST<CR> FEEDBACK ~nn@NAME-RST_ok<CR><LF>		Reset the machine name (S/N last digits are 0102): #NAME-RST_?kramer_0102<CR>

Function	Description	Syntax	Parameters/Attributes	Example
NET-DHCP	<p>Set DHCP mode.</p> <p>i 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 NAME 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>i For Backward compatibility, the id parameter can be omitted. In this case, the Network ID, by default, is 0, which is the Ethernet control port.</p>	<p>COMMAND</p> <pre>#NET-DHCP_dhcp_state<CR></pre> <p>FEEDBACK</p> <pre>~nn@NET-DHCP_dhcp_state<CR><LF></pre>	<p>dhcp_state –</p> <p>1 – Try to use DHCP. (If unavailable, use the IP address set by the factory or the net-ip command).</p>	<p>Enable DHCP mode for port 1, if available:</p> <pre>#NET-DHCP_1<CR></pre>
NET-DHCP?	<p>Get DHCP mode.</p> <p>i For Backward compatibility, the id parameter can be omitted. In this case, the Network ID, by default, is 0, which is the Ethernet control port.</p>	<p>COMMAND</p> <pre>#NET-DHCP?_<CR></pre> <p>FEEDBACK</p> <pre>~nn@NET-DHCP_netw_id,dhcp_mode<CR><LF></pre>	<p>dhcp_mode –</p> <p>0 – Do not use DHCP. Use the IP set by the factory or using the net-ip or net-config command.</p> <p>1 – Try to use DHCP. If unavailable, use the IP set by the factory or using the net-ip or net-config command.</p>	<p>Get DHCP mode for port 1:</p> <pre>#NET-DHCP?<CR></pre>
NET-GATE	<p>Set gateway IP.</p> <p>i 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>COMMAND</p> <pre>#NET-GATE_ip_address<CR></pre> <p>FEEDBACK</p> <pre>~nn@NET-GATE_ip_address<CR><LF></pre>	<p>ip_address – Format: xxx.xxx.xxx.xxx</p>	<p>Set the gateway IP address to 192.168.0.1:</p> <pre>#NET-GATE_192.168.000.001<CR></pre>
NET-GATE?	<p>Get gateway IP.</p> <p>i A network gateway connects the device via another network and maybe over the Internet. Be aware of security problems.</p>	<p>COMMAND</p> <pre>#NET-GATE?_<CR></pre> <p>FEEDBACK</p> <pre>~nn@NET-GATE_ip_address<CR><LF></pre>	<p>ip_address – Format: xxx.xxx.xxx.xxx</p>	<p>Get the gateway IP address:</p> <pre>#NET-GATE?_<CR></pre>
NET-IP	<p>Set IP address.</p> <p>i For proper settings consult your network administrator.</p>	<p>COMMAND</p> <pre>#NET-IP_ip_address<CR></pre> <p>FEEDBACK</p> <pre>~nn@NET-IP_ip_address<CR><LF></pre>	<p>ip_address – Format: xxx.xxx.xxx.xxx</p>	<p>Set the IP address to 192.168.1.39:</p> <pre>#NET-IP_192.168.001.039<CR></pre>
NET-IP?	<p>Get IP address.</p>	<p>COMMAND</p> <pre>#NET-IP?_<CR></pre> <p>FEEDBACK</p> <pre>~nn@NET-IP_ip_address<CR><LF></pre>	<p>ip_address – Format: xxx.xxx.xxx.xxx</p>	<p>Get the IP address:</p> <pre>#NET-IP?_<CR></pre>
NET-MAC?	<p>Get MAC address.</p> <p>i For backward compatibility, the id parameter can be omitted. In this case, the Network ID, by default, is 0, which is the Ethernet control port.</p>	<p>COMMAND</p> <pre>#NET-MAC?<CR></pre> <p>FEEDBACK</p> <pre>~nn@NET-MAC_mac_address<CR><LF></pre>	<p>mac_address – Unique MAC address. Format: XX-XX-XX-XX-XX where X is hex digit</p>	<pre>#NET-MAC?<CR></pre>
NET-MASK	<p>Set subnet mask.</p> <p>i For proper settings consult your network administrator.</p>	<p>COMMAND</p> <pre>#NET-MASK_net_mask<CR></pre> <p>FEEDBACK</p> <pre>~nn@NET-MASK_net_mask<CR><LF></pre>	<p>net_mask – Format: xxx.xxx.xxx.xxx</p>	<p>Set the subnet mask to 255.255.0.0:</p> <pre>#NET-MASK_255.255.000.000<CR></pre>
NET-MASK?	<p>Get subnet mask.</p>	<p>COMMAND</p> <pre>#NET-MASK?_<CR></pre> <p>FEEDBACK</p> <pre>~nn@NET-MASK_net_mask<CR><LF></pre>	<p>net_mask – Format: xxx.xxx.xxx.xxx</p>	<p>Get the subnet mask:</p> <pre>#NET-MASK?<CR></pre>

Function	Description	Syntax	Parameters/Attributes	Example
PASS	Set password for login level. ⓘ The default password is an empty string.	COMMAND #PASS_login_level,password<CR> FEEDBACK ~nn@PASS_login_level,password<CR><LF>	login_level – Level of login to set (End User or Admin): 0 – User 1 – Admin password – Password for the login_level. Up to 15 printable ASCII chars	Set the password for the Admin protocol permission level to 33333: #PASS_admin,33333<CR>
PASS?	Get password for login level. ⓘ The default password is an empty string.	COMMAND #PASS?_login_level<CR> FEEDBACK ~nn@PASS_login_level,password<CR><LF>	login_level – Level of login to set (User or Admin): 0 – User 1 – Admin password – Password for the login_level. Up to 15 printable ASCII chars	Get the password for the Admin protocol permission level: #PASS?_admin<CR>
PROG-ACTION	Set Step-In button action bitmap. ⓘ Programs matrix action as a response for external event (programmable button pressed).	COMMAND #PROG-ACTION_port_type,port_id,button_id,bitmap_actions_id<CR> FEEDBACK ~nn@PROG-ACTION_port_type,port_id,button_id,bitmap_actions_id<CR><LF>	io_mode – Input 0 – Input port_id – input number on the device: 1 – HDMI IN 1 2 – HDMI IN 2 3 – HDMI IN 3 4 – HDMI IN 4 5 – HDMI IN 5 6 – HDMI IN 6 7 – HDMI IN 7 8 – HDMI IN 8 button_id – External programmable button ID bitmap_actions_id – Bitmap representing actions to perform after receiving button_id. format: XXXX...X, where X is a hex digit. The binary form of every hex digit represents actions from the table 0 – Echo to controller 1 – Step-in out 1 2 – Step-in out 2 3 – Step-in out 3 4 – Step-in out 4 5 – Step-in out 5 6 – Step-in out 6 7 – Step-in out 7 8 – Step-in out 8 Setting '1' says that the corresponding action must be executed.	Set step-in button actions on input 3: #PROG-ACTION_0,3,1,0x07<CR>
PROG-ACTION?	Get step-in button action bitmap. ⓘ Programs matrix action as a response for external event (programmable button pressed).	COMMAND #PROG-ACTION?_port_type,port_id,button_id<CR> FEEDBACK ~nn@PROG-ACTION_port_type,port_id,button_id,bitmap_actions_id<CR><LF>	io_mode – Input 0 – Input port_id – input number on the device: 1 – HDMI IN 1 2 – HDMI IN 2 3 – HDMI IN 3 4 – HDMI IN 4 5 – HDMI IN 5 6 – HDMI IN 6 7 – HDMI IN 7 8 – HDMI IN 8 button_id – External programmable button ID bitmap_actions_id – Bitmap representing actions to perform after receiving button_id. format: XXXX...X, where X is a hex digit. The binary form of every hex digit represents actions from the table 0 – Echo to controller 1 – Step-in out 1 2 – Step-in out 2 3 – Step-in out 3 4 – Step-in out 4 5 – Step-in out 5 6 – Step-in out 6 7 – Step-in out 7 8 – Step-in out 8 Setting '1' says that the corresponding action must be executed.	Get step-in button action bitmap on input 3: #PROG-ACTION?_0,3,1<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>

Function	Description	Syntax	Parameters/Attributes	Example
PRST-AUD?	<p>LEGACY COMMAND. Get audio connections from saved preset.</p> <p>i In most units, video and audio presets with the same number are stored and recalled together by commands #PRST-STO and #PRST-RCL.</p>	<p>COMMAND #PRST-AUD?_preset,out<CR> #PRST-AUD?_preset,*<CR></p> <p>FEEDBACK ~@PRST-AUD_preset,>out<CR><LF> ~@PRST-AUD_preset,i>1,i>2,i>3,...<CR><LF></p>	<p>preset – Preset number – 1 – Preset 1 2 – Preset 2 3 – Preset 3 4 – Preset 4 5 – Preset 5 6 – Preset 6 7 – Preset 7 8 – Preset 8 9 – Preset 9 10 – Preset 10 11 – Preset 11 12 – Preset 12 13 – Preset 13 14 – Preset 14 15 – Preset 15 16 – Preset 16</p> <p>> – Connection character between in and out parameters out – Number that indicates the specific output: * – All outputs 1 – HDMI OUT 1 2 – HDMI OUT 2 3 – HDMI OUT 3 4 – HDMI OUT 4 5 – HDMI OUT 5 6 – HDMI OUT 6 7 – HDMI OUT 7 8 – HDMI OUT 8</p>	<p>Get audio connection IN 1 to OUT 3 from saved preset 1: #PRST-AUD?_1<CR></p>
PRST-LST?	<p>Get saved preset list.</p> <p>i In most units, video and audio presets with the same number are stored and recalled together by commands #PRST-STO and #PRST-RCL.</p>	<p>COMMAND #PRST-LST?_<CR></p> <p>FEEDBACK ~nn@PRST-LST_preset,preset,...<CR><LF></p>	<p>preset – Preset number</p>	<p>Show preset list: #PRST-LST?<CR></p>
PRST-RCL	<p>Recall saved preset list.</p> <p>i In most units, video and audio presets with the same number are stored and recalled together by commands #PRST-STO and #PRST-RCL.</p>	<p>COMMAND #PRST-RCL_preset<CR></p> <p>FEEDBACK ~nn@PRST-RCL_preset<CR><LF></p>	<p>preset – Preset number</p>	<p>Recall preset 1: #PRST-RCL_1<CR></p>
PRST-STO	<p>Store current connections, volumes and modes in preset.</p> <p>i In most units, video and audio presets with the same number are stored and recalled together by commands #PRST-STO and #PRST-RCL.</p>	<p>COMMAND #PRST-STO_preset<CR></p> <p>FEEDBACK ~nn@PRST-STO_preset<CR><LF></p>	<p>preset – Preset number</p>	<p>Store preset 1: #PRST-STO_1<CR></p>

Function	Description	Syntax	Parameters/Attributes	Example
PRST-VID?	<p>Get video connections from saved preset.</p> <p>i In most units, video and audio presets with the same number are stored and recalled together by commands #PRST-STO and #PRST-RCL.</p>	<p>COMMAND</p> <pre>#PRST-VID?_preset,out_id<CR> #PRST-VID?_preset,*<CR></pre> <p>FEEDBACK</p> <pre>~nn@PRST-VID,_preset,in_id>out_id<CR><LF> ~nn@PRST-VID,_preset,>1,>2,>3,...<CR><LF></pre>	<p>preset – Preset number –</p> <ul style="list-style-type: none"> 1 – Preset 1 2 – Preset 2 3 – Preset 3 4 – Preset 4 5 – Preset 5 6 – Preset 6 7 – Preset 7 8 – Preset 8 9 – Preset 9 10 – Preset 10 11 – Preset 11 12 – Preset 12 13 – Preset 13 14 – Preset 14 15 – Preset 15 16 – Preset 16 <p>In_id</p> <ul style="list-style-type: none"> 0 – disconnect output 1 – HDMI IN 1 2 – HDMI IN 2 3 – HDMI IN 3 4 – HDMI IN 4 5 – HDMI IN 5 6 – HDMI IN 6 7 – HDMI IN 7 8 – HDMI IN 8 <p>> – Connection character between in and out parameters</p> <p>out_id – Output number</p> <ul style="list-style-type: none"> * – All outputs 1 – HDMI OUT 1 2 – HDMI OUT 2 3 – HDMI OUT 3 4 – HDMI OUT 4 5 – HDMI OUT 5 6 – HDMI OUT 6 7 – HDMI OUT 7 8 – HDMI OUT 8 	<p>Get video connections from preset 3 for all outputs:</p> <pre>#PRST-VID?_3,*<CR></pre>
REMOTE-INFO?	<p>Get connected Step-in module information.</p> <p>i The matrix uses this command to notify about Step-in client changes.</p>	<p>COMMAND</p> <pre>#REMOTE-INFO?_io_mode,io_index<CR></pre> <p>FEEDBACK</p> <pre>~nn@REMOTE-INFO_io_mode,io_index,connected_state, model_name,in_selected,step-in_state,in_count, cntl_btn_count,in_src1,in_src2...<CR><LF></pre>	<p>io_mode – Input/Output</p> <ul style="list-style-type: none"> 0 – Input 1 – Output <p>io_index – Number that indicates the specific input or output port:</p> <ul style="list-style-type: none"> 1 – HDMI IN 1 2 – HDMI IN 2 3 – HDMI IN 3 4 – HDMI IN 4 5 – HDMI IN 5 6 – HDMI IN 6 7 – HDMI IN 7 8 – HDMI IN 8 1 – HDMI OUT 1 2 – HDMI OUT 2 3 – HDMI OUT 3 4 – HDMI OUT 4 5 – HDMI OUT 5 6 – HDMI OUT 6 7 – HDMI OUT 7 8 – HDMI OUT 8 <p>connected_state – 0/1 (if module connected)</p> <p>model_name – Model name string</p> <p>in_selected – Input, currently chosen on module</p> <p>step-in_state – Step-in state</p> <ul style="list-style-type: none"> 0 – module doesn't support Step-in 1 – module supports Step-in None <p>in_count – 8</p> <p>cntl_btn_count – Number of control buttons on module</p> <p>in_src – Type2... typeN – Input type according to num_of_inputs</p> <ul style="list-style-type: none"> 0 – Undefined 2 – HDMI 	<p>Get connected Step-in module information:</p> <pre>#REMOTE-INFO?_0,1<CR></pre>
RESET	<p>Reset device.</p> <p>i 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.</p>	<p>COMMAND</p> <pre>#RESET<CR></pre> <p>FEEDBACK</p> <pre>~nn@RESET_ok<CR><LF></pre>		<p>Reset the device:</p> <pre>#RESET<CR></pre>

Function	Description	Syntax	Parameters/Attributes	Example
SECUR	Start/stop security.  The permission system works only if security is enabled with the "SECUR" command.	COMMAND #SECUR security_state<CR> FEEDBACK ~nn@SECUR security_state<CR><LF>	security_state – Security state 0 – OFF (disables security) 1 – ON (enables security)	Enable the permission system: #SECUR_0<CR>
SECUR?	Get current security state.  The permission system works only if security is enabled with the "SECUR" command.	COMMAND #SECUR?_<CR> FEEDBACK ~nn@SECUR security_state<CR><LF>	security_state – Security state 0 – OFF (disables security) 1 – ON (enables security)	Get current security state: #SECUR?_<CR>
SET-IN-CAP	Set input EDID status.	COMMAND #SET-IN-CAP stage, stage_id, mode<CR> FEEDBACK ~nn@SET-IN-CAP stage, stage_id, mode<CR><LF>	stage – Input: 0 – Input stage_id – Number that indicates the specific input: 0 – Color Space 1 – Color Depth 2 – Two Audio Channels mode – 0 – Pass 1 – Set	Set the input EDID support to Two Audio Channels: #SET-IN-CAP_0, 2, 1<CR>
SET-IN-CAP?	Get input EDID status.	COMMAND #SET-IN-CAP? stage, stage_id<CR> FEEDBACK ~nn@SET-IN-CAP? stage, stage_id, mode<CR><LF>	stage – Input: 0 – Input stage_id – Number that indicates the specific input: 0 – Color Space 1 – Color Depth 2 – Two Audio Channels mode – 0 – Pass 1 – Set	Get the input EDID support to Color Depth: #SET-IN-CAP?_0, 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 – HDMI IN 1 2 – HDMI IN 2 3 – HDMI IN 3 4 – HDMI IN 4 5 – HDMI IN 5 6 – HDMI IN 6 7 – HDMI IN 7 8 – HDMI IN 8 status – Signal status according to signal validation: 0 – Off 1 – On	Get the input signal lock status of IN 1: #SIGNAL?_1<CR>
SIG-TYPE?	Get signal type on input/output.  "Set" command is not available for all devices (refer to device specifications).	COMMAND #SIG-TYPE? io_mode, io_index<CR> FEEDBACK ~nn@SIG-TYPE io_mode, io_index, signal_src<CR><LF>	io_mode – Input/Output 0 – Input 1 – Output io_index – Number that indicates the specific input or output port: 1 – HDMI IN 1 2 – HDMI IN 2 3 – HDMI IN 3 4 – HDMI IN 4 5 – HDMI IN 5 6 – HDMI IN 6 7 – HDMI IN 7 8 – HDMI IN 8 1 – HDMI OUT 1 2 – HDMI OUT 2 3 – HDMI OUT 3 4 – HDMI OUT 4 5 – HDMI OUT 5 6 – HDMI OUT 6 7 – HDMI OUT 7 8 – HDMI OUT 8 signal_src – Signal type 0 – No signal 2 – HDMI	Get signal type on input/output: #SIG-TYPE?_1, 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>
TUNNEL-CTRL	LEGACY COMMAND. Send an asynchronous command to a remote Step-in equipment.	COMMAND #TUNNEL-CTRL io_mode, io_index, cmd_name<CR> FEEDBACK None	io_mode – Input/Output 0 – Input 1 – Output io_index – Number that indicates the specific input or output port: 1-N (N= the total number of input or output ports) cmd_name – Command to send to the Step-in client	LEGACY COMMAND: #TUNNEL-CTRL_1, 1, 1<CR>

Function	Description	Syntax	Parameters/Attributes	Example
UART	<p>Set com port configuration.</p> <p>① In the FC-2x the serial port is selectable to RS-232 or RS-485 (usually serial port 1).</p> <p>If Serial is configured when RS-485 is selected, the RS-485 UART port automatically changes.</p> <p>The command is backward compatible, meaning that if the extra parameters do not exist, FW goes to. RS-232.</p> <p>Stop_bits 1.5 is only relevant for 5 data_bits.</p>	<p>COMMAND #UART_com_id,baud_rate,data_bits,parity,stop_bits_mode,serial_type,485_term<CR></p> <p>FEEDBACK ~nn@UART_com_id,baud_rate,data_bits,parity,stop_bits_mode,serial_type,485_term<CR><LF></p>	<p>com_id – 1 to n (machine dependent)</p> <p>baud_rate – 9600 - 115200</p> <p>data_bits – 5-8</p> <p>parity – Parity Type 0 – No 1 – Odd 2 – Even 3 – Mark 4 – Space</p> <p>stop_bits_mode – 1/1.5/2</p> <p>serial_type – 232/485 0 – 232 1 – 485</p> <p>485_term – 485 termination state 0 – disable 1 – enable (optional - this exists only when serial_type is 485)</p>	<p>Set baud rate to 9600, 8 data bits, parity to none and stop bit to 1: #UART_9600,8,node,1<CR></p>
UART?	<p>Get com port configuration.</p> <p>① In the FC-2x the serial port is selectable to RS-232 or RS-485 (usually serial port 1).</p> <p>If Serial is configured when RS-485 is selected, the RS-485 UART port automatically changes.</p> <p>The command is backward compatible, meaning that if the extra parameters do not exist, FW goes to. RS-232.</p> <p>Stop_bits 1.5 is only relevant for 5 data_bits.</p>	<p>COMMAND #UART?_com_id<CR></p> <p>FEEDBACK ~nn@UART_com_id,baud_rate,data_bits,parity,stop_bits_mode,serial_type,485_term<CR><LF></p>	<p>com_id – 1 to n (machine dependent)</p> <p>baud_rate – 9600 - 115200</p> <p>data_bits – 5-8</p> <p>parity – Parity Type 0 – No 1 – Odd 2 – Even 3 – Mark 4 – Space</p> <p>stop_bits_mode – 1/1.5/2</p> <p>serial_type – 232/485 0 – 232 1 – 485</p> <p>485_term – 485 termination state 0 – disable 1 – enable (optional - this exists only when serial_type is 485)</p>	<p>Set baud rate to 9600, 8 data bits, parity to none and stop bit to 1: #UART_1,9600,8,node,1<CR></p>
VERSION?	<p>Get firmware version number.</p>	<p>COMMAND #VERSION?_<CR></p> <p>FEEDBACK ~nn@VERSION_firmware_version<CR><LF></p>	<p>firmware_version – XX.XX.XXXX where the digit groups are: major.minor.build version</p>	<p>Get the device firmware version number: #VERSION?_<CR></p>
VID	<p>LEGACY COMMAND. Set video switch state.</p> <p>① The GET command identifies input switching on Step-in clients.</p> <p>The SET command is for remote input switching on Step-in clients (essentially via by the Web).</p> <p>This is a legacy command. New Step-in modules support the ROUTE command.</p>	<p>COMMAND #VID_in_id>out_id<CR></p> <p>FEEDBACK ~nn@VID_in_id>out_id<CR><LF></p>	<p>in_id – Indicates the ID of the input: 1-n (n= the total number of inputs) > – Connection character between in and out parameters out_id – Output number * for all outputs</p>	<p>Switch IN 1 to OUT 3: #VID_1>3<CR></p>
VID?	<p>LEGACY COMMAND. Get video switch state.</p> <p>① The GET command identifies input switching on Step-in clients.</p> <p>① The SET command is for remote input switching on Step-in clients (essentially via by the Web).</p> <p>This is a legacy command. New Step-in modules support the ROUTE command.</p>	<p>COMMAND #VID?_out_id<CR></p> <p>FEEDBACK ~nn@VID_in_id>out_id<CR><LF></p>	<p>in_id – Indicates the ID of the input: 1 – HDMI IN 1 2 – HDMI IN 2 3 – HDMI IN 3 4 – HDMI IN 4 5 – HDMI IN 5 6 – HDMI IN 6 7 – HDMI IN 7 8 – HDMI IN 8 > – Connection character between in and out parameters out_id – Output number: 1 – HDMI OUT 1 2 – HDMI OUT 2 3 – HDMI OUT 3 4 – HDMI OUT 4 5 – HDMI OUT 5 6 – HDMI OUT 6 7 – HDMI OUT 7 8 – HDMI OUT 8</p>	<p>Get video switch state: #VID?_2<CR></p>

Function	Description	Syntax	Parameters/Attributes	Example
VID-PATTERN	Set test pattern on output.	COMMAND #VID-PATTERN_out_index,pattern_id<CR> FEEDBACK ~nn@VID-PATTERN_out_index,pattern_id<CR><LF>	out_index – Number that indicates the specific output: 1 – HDMI OUT 1 2 – HDMI OUT 2 3 – HDMI OUT 3 4 – HDMI OUT 4 5 – HDMI OUT 5 6 – HDMI OUT 6 7 – HDMI OUT 7 8 – HDMI OUT 8 pattern_id – Number of system patterns: 1 – Color bars 2 – Ramp 3 – Solid White 4 – Solid Black 5 – Solid Red 6 – Solid Green	Switch PATTERN 1 to OUT 3: #VID-PATTERN_3,1<CR>
VID-PATTERN?	Get test pattern on output.	COMMAND #VID-PATTERN?_out_index<CR> FEEDBACK ~nn@VID-PATTERN_out_index,pattern_id<CR><LF>	out_index – Number that indicates the specific output: 1 – HDMI OUT 1 2 – HDMI OUT 2 3 – HDMI OUT 3 4 – HDMI OUT 4 5 – HDMI OUT 5 6 – HDMI OUT 6 7 – HDMI OUT 7 8 – HDMI OUT 8 pattern_id – Number of system patterns: 1 – Color bars 2 – Ramp 3 – Solid White 4 – Solid Black 5 – Solid Red 6 – Solid Green	Get test pattern on output: #VID-PATTERN?_3<CR>
VMUTE	Set enable/disable video on output. <i>i</i> 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 OUT 1 2 – HDMI OUT 2 3 – HDMI OUT 3 4 – HDMI OUT 4 5 – HDMI OUT 5 6 – HDMI OUT 6 7 – HDMI OUT 7 8 – HDMI OUT 8 flag – Video Mute 0 – Video enabled 1 – Video disabled 2 – Blank picture	Disable the video output on OUT 2: #VMUTE_2,0<CR>
VMUTE?	Get video on output status. <i>i</i> 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 OUT 1 2 – HDMI OUT 2 3 – HDMI OUT 3 4 – HDMI OUT 4 5 – HDMI OUT 5 6 – HDMI OUT 6 7 – HDMI OUT 7 8 – HDMI OUT 8 flag – Video Mute 0 – Video enabled 1 – Video disabled 2 – Blank picture	Get video on output status: #VMUTE?_2<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_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