

**Kramer Electronics, Ltd.**



# **Protocol 3000**

**Version 1.00 (Full Version)**

## Contents

|          |                              |          |
|----------|------------------------------|----------|
| <b>1</b> | <b>Protocol 3000 Syntax</b>  | <b>1</b> |
| 1.1      | Host Message Format          | 1        |
| 1.1.1    | Simple Command               | 1        |
| 1.1.2    | Command String               | 1        |
| 1.2      | Device Message Format        | 1        |
| 1.2.1    | Device Long Response         | 1        |
| 1.3      | Command Terms                | 2        |
| 1.4      | Entering Commands            | 3        |
| 1.5      | Command Forms                | 3        |
| 1.6      | Command Chaining             | 3        |
| 1.7      | Maximum String Length        | 3        |
| 1.8      | Backward Support             | 3        |
| <b>2</b> | <b>Commands</b>              | <b>4</b> |
| 2.1      | Help Commands                | 4        |
| 2.2      | Device Initiated Messages    | 4        |
| 2.3      | Result and Error Codes       | 4        |
| 2.4      | Basic Routing Commands       | 5        |
| 2.5      | Preset Commands              | 6        |
| 2.6      | Operation Commands           | 6        |
| 2.7      | Audio Parameters Commands    | 7        |
| 2.8      | Identification Commands      | 8        |
| 2.9      | Network Setting Commands     | 9        |
| 2.10     | Machine Information Commands | 9        |
| 2.11     | Advanced Switching Commands  | 10       |

---

# 1 Protocol 3000 Syntax

## 1.1 Host Message Format

| Start | Address (optional)     | Body    | Delimiter |
|-------|------------------------|---------|-----------|
| #     | <i>Destination_id@</i> | Message | <b>CR</b> |

### 1.1.1 Simple Command

Command string with only one command without addressing:

| Start | Address (optional) | Body   | Delimiter |
|-------|--------------------|--|-----------|
| #     |                    | Command <b>SP</b> <i>Parameter_1,Parameter_2,...</i> | <b>CR</b> |

### 1.1.2 Command String

Formal syntax with commands concatenation and addressing:

| Start | Address                | Body  | Delimiter |
|-------|------------------------|---|-----------|
| #     | <i>Destination_id@</i> | <b>Command_1</b> <i>Parameter1_1,Parameter1_2,.../</i><br><b>Command_2</b> <i>Parameter2_1,Parameter2_2,.../</i><br><b>Command_3</b> <i>Parameter3_1,Parameter3_2,.../...</i> | <b>CR</b> |

## 1.2 Device Message Format

| Start | Address (optional) | Body    | delimiter           |
|-------|--------------------|---------|---------------------|
| ~     | <i>Sender_id@</i>  | Message | <b>CR</b> <b>LF</b> |

### 1.2.1 Device Long Response

Echoing command:

| Start | Address (optional) | Body  | Delimiter           |
|-------|--------------------|---|---------------------|
| ~     | <i>Sender_id@</i>  | Command <b>SP</b> [ <i>Param1 ,Param2 ...</i> ] <b>result</b> | <b>CR</b> <b>LF</b> |

**CR** = Carriage return (ASCII 13 = 0x0D)

**LF** = Line feed (ASCII 10 = 0x0A)

**SP** = Space (ASCII 32 = 0x20)

---

## 1.3 Command Terms

### Command

A sequence of ASCII letters ('A'-'Z', 'a'-'z' and '-').  
Command and parameters must be separated by at least one space.

### Parameters

A sequence of alphanumeric ASCII characters ('0'-'9', 'A'-'Z', 'a'-'z' and some special characters for specific commands). Parameters are separated by commas.

### Message string

Every command entered as part of a message string begins with a **message starting character** and ends with a **message closing character**.

**Note:** A string can contain more than one command. Commands are separated by a pipe ( '| ' ) character.

### Message starting character

'#' – For host command/query  
'~' – For machine response

### Device address (Optional, for K-NET)

K-NET Device ID followed by '@'

### Query sign

'?' follows some commands to define a query request.

### All outputs sign

'\*' defines all outputs.

### Message closing character

**CR** – For host messages; carriage return (ASCII 13)

**CRLF** – For machine messages; carriage return (ASCII 13) + line-feed (ASCII 10)

### Command chain separator character

When a message string contains more than one command, a pipe ( '| ' ) character separates each command.

Spaces between parameters or command terms are ignored.

---

## 1.4 Entering Commands

You can directly enter all commands using a terminal with ASCII communications software, such as HyperTerminal, Hercules, etc. Connect the terminal to the serial, Ethernet, or USB port on the Kramer device. To enter **CR**, press the Enter key. (**LF** is also sent but is ignored by command parser).

For commands sent from some non-Kramer controllers like Crestron, some characters require special coding (such as, */X##*). Refer to the controller manual.

## 1.5 Command Forms

Some commands have short name syntax in addition to long name syntax to allow faster typing. The response is always in long syntax.

## 1.6 Command Chaining

Multiple commands can be chained in the same string. Each command is delimited by a pipe character ( '| ' ). When chaining commands, enter the **message starting character** and the **message closing character** only once, at the beginning of the string and at the end.

Commands in the string do not execute until the closing character is entered.

A separate response is sent for every command in the chain.

## 1.7 Maximum String Length

64 characters

## 1.8 Backward Support

Protocol 2000 is transparently supported by Protocol 3000. You can switch between protocols using a switch protocol command from either platform.

---

## 2 Commands

### 2.1 Help Commands

| Command              | Syntax      | Response        |
|----------------------|-------------|-----------------|
| Protocol handshaking | # <b>CF</b> | ~OK <b>CRLF</b> |

### 2.2 Device Initiated Messages

| Command       | Syntax   |
|---------------|--|
| Start message | Kramer Electronics LTD. . <b>Device Model</b> Version<br><b>Software Version</b> |

*Switcher actions:*

|   |                      |
|---|----------------------|
| Audio-video channel has switched (AFV mode) | AV <b>IN&gt;OUT</b>  |
| Video channel has switched (breakaway mode) | VID <b>IN&gt;OUT</b> |
| Audio channel has switched (breakaway mode) | AUD <b>IN&gt;OUT</b> |

### 2.3 Result and Error Codes

|                                     | Syntax                              |
|-------------------------------------|-------------------------------------|
| Command ran successfully, no error. | <b>COMMAND</b> <b>PARAMETERS</b> OK |

*Protocol Errors:*

|   |        |
|---|--------|
| Syntax error  | ERR001 |
| Command not available for this device                         | ERR002 |
| Parameter is out of range                                     | ERR003 |
| Unauthorized access (command run without the matching login). | ERR004 |

## 2.4 Basic Routing Commands

| Command                | Syntax   | Response  |
|------------------------|--|---|
| Switch audio and video | AV <u>IN</u> > <u>OUT</u> , <u>IN</u> > <u>OUT</u> , ...   | AV <u>IN</u> > <u>OUT</u> , <u>IN</u> > <u>OUT</u> ... <u>RESULT</u>    |
| Switch video only      | VID <u>IN</u> > <u>OUT</u> , <u>IN</u> > <u>OUT</u> , ...<br>Short form: V <u>IN</u> > <u>OUT</u> , <u>IN</u> > <u>OUT</u> , ... | VID <u>IN</u> > <u>OUT</u> , <u>IN</u> > <u>OUT</u> , ... <u>RESULT</u> |

**Note:**

When AFV mode is active, this command also switches audio. If audio is in breakaway mode, the device display mode changes to show the audio connection status.

|                   |  |   |
|-------------------|--|---|
| Switch audio only | AUD <u>IN</u> > <u>OUT</u> , <u>IN</u> > <u>OUT</u> , ...<br>Short form: A <u>IN</u> > <u>OUT</u> , <u>IN</u> > <u>OUT</u> , ... | AUD <u>IN</u> > <u>OUT</u> , <u>IN</u> > <u>OUT</u> , ... <u>RESULT</u> |
|-------------------|--|---|

**Note:**

When AFV mode is active, this command also switches video.

|                                   |  |   |
|-----------------------------------|--|---|
| Read video connection             | VID? <u>OUT</u><br>Short form: V? <u>OUT</u><br>VID? * | VID <u>IN</u> > <u>OUT</u><br><br>VID <u>IN</u> >1, <u>IN</u> >2, ... |
| Read audio connection             | AUD? <u>OUT</u><br>Short form: A? <u>OUT</u><br>AUD? * | AUD <u>IN</u> > <u>OUT</u><br><br>AUD <u>IN</u> >1, <u>IN</u> >2, ... |
| Reset video and audio connections | AV-RST   | AV-RST <u>RESULT</u>  |

**Parameter Description:**

IN = Input number or '0' to disconnect output.

'>' = Connection character between in and out parameters.

OUT = Output number or '\*' for all outputs.

**Examples:**

|  |  |  |
|--|--|--|
| Switch video and audio input 3 to output 7           | #AV 3>7 <u>CR</u>  | ~AV 3>7 OK <u>CRLF</u>   |
| Switch video input 2 to output 4                     | #V 2>4 <u>CR</u>   | ~VID 2>4 OK <u>CRLF</u>  |
| Switch video input 4 to output 2 in machine number 6 | #6@VID 4>2 <u>CR</u>   | ~6@VID 4>2 OK <u>CRLF</u>  |
| Disconnect video and audio output 4                  | #AV 0>4 <u>CR</u>  | ~AV 0>4 OK <u>CRLF</u>   |
| Switch video input 3 to all outputs                  | #V 3>* <u>CR</u>   | ~VID 3>* OK <u>CRLF</u>  |
| Chaining multiple commands                           | #AV 1>*   V 3>4, 2>2, 2>1, 0>2   V 3>9   A 0>1   V? * <u>CR</u><br>1. Switch audio and video from input 1 to all outputs.<br>2. Switch video input 3 to output 4, video input 2 to output 2, video input 2 to output 1 and disconnect video output 2.<br>3. Switch video input 3 to output 9 (non-existent).<br>4. Disconnect audio output 1.<br>5. Get status of all video links.<br>Command processing begins after entering <u>CR</u> . A response is sent for each command after processing. | ~AV 1>* OK <u>CRLF</u><br>~VID 3>4, 2>2, 2>1, 0>2 OK <u>CRLF</u><br>~VID <u>ERR003</u> <u>CRLF</u><br>~AUD 0>1 OK <u>CRLF</u><br>~VID 2>1, 0>2, 1>3, 3>4 <u>CRLF</u> |

## 2.5 Preset Commands

| Command                                  | Syntax  | Response   |
|--|---|--|
| Store current connections to preset      | PRST-STO <u>PRESET</u><br>Short form: PSTO <u>PRESET</u>  | PRST-STO <u>PRESET</u> <u>RESULT</u>   |
| Recall saved preset                      | PRST-RCL <u>PRESET</u><br>Short form: PRCL <u>PRESET</u>  | PRST-RCL <u>PRESET</u> <u>RESULT</u>   |
| Delete saved preset                      | PRST-DEL <u>PRESET</u><br>Short form: PDEL <u>PRESET</u>  | PRST-DEL <u>PRESET</u> <u>RESULT</u>   |
| Read video connections from saved preset | PRST-VID? <u>PRESET</u> , <u>OUT</u><br>Short form: PVID? <u>PRESET</u> , <u>OUT</u><br>PRST-VID? <u>PRESET</u> , * | PRST-VID <u>PRESET</u> , <u>IN</u> > <u>OUT</u><br><br>PRST-VID <u>PRESET</u> , <u>IN</u> >1, <u>IN</u> >2,... |
| Read audio connections from saved preset | PRST-AUD? <u>PRESET</u> , <u>OUT</u><br>Short form: PAUD? <u>PRESET</u> , <u>OUT</u><br>PRST-AUD? <u>PRESET</u> , * | PRST-AUD <u>PRESET</u> , <u>IN</u> > <u>OUT</u><br><br>PRST-AUD <u>PRESET</u> , <u>IN</u> >1, <u>IN</u> >2,... |
| Read saved presets list                  | PRST-LST?<br>Short form: PLST?  | PRST-LST <u>PRESET</u> , <u>PRESET</u> , ...   |

### Parameter Description:

PRESET = Preset number.

OUT = Output in preset to display, "\*" for all.

### Examples:

|   |                          |                              |
|---|--------------------------|------------------------------|
| Store current audio and video connections to preset 5 | #PRST-STR 5 <u>CR</u>    | ~PRST-STR 5 OK <u>CRLF</u>   |
| Recall audio and video connections from preset 3      | #PRCL 3 <u>CR</u>        | ~PRST-RCL 3 OK <u>CRLF</u>   |
| Show source of video output 2 from preset 3           | #PRST-VID? 3,2 <u>CR</u> | ~PRST-VID 3: 4>2 <u>CRLF</u> |

## 2.6 Operation Commands

| Command                       | Syntax   | Response                               |
|-------------------------------|--|--|
| Lock front panel              | LOCK-FP <u>LOCK-MODE</u><br>Short form: LCK <u>LOCK-MODE</u> | LOCK-FP <u>LOCK-MODE</u> <u>RESULT</u> |
| Get front panel locking state | LOCK-FP?   | LOCK-FP <u>LOCK-MODE</u>               |

### Parameter Description:

LOCK-MODE = Front panel locking state:

'0' or 'off' to unlock front panel buttons

'1' or 'on' to lock front panel buttons

|                          |       |          |
|--------------------------|-------|----------|
| Reset device             | RESET | RESET OK |
| Switch to protocol 2000* | P2000 | P2000 OK |

\* Protocol 2000 has a command to switch back to ASCII protocol (like Protocol 3000)

## 2.7 Audio Parameters Commands

| Command                                      | Syntax   | Response  |
|--|--|---|
| Set simple audio volume                      | VOLUME <u>VOLUME</u><br>Short form: VOL <u>VOLUME</u>  | VOLUME <u>VOLUME</u> <u>RESULT</u>                              |
| Increase/decrease simple audio volume        | VOLUME <u>+/-</u><br>Short form: VOL <u>+/-</u>  | VOLUME <u>+/-</u> <u>RESULT</u>                                 |
| Read simple audio level                      | VOLUME?<br>Short form: VOL?  | VOLUME <u>VOLUME</u>  |
| Set audio level in specific amplifier stage. | AUD-LVL <u>STAGE</u> <u>CHANNEL</u> <u>VOLUME</u><br>Short form: ADL <u>STAGE</u> <u>CHANNEL</u> <u>VOLUME</u> | AUD-LVL <u>STAGE</u> <u>CHANNEL</u> <u>VOLUME</u> <u>RESULT</u> |
| Read audio volume level                      | AUD-LVL? <u>STAGE</u> <u>CHANNEL</u><br>Short form: ADL? <u>STAGE</u>  | AUD-LVL <u>STAGE</u> <u>CHANNEL</u> <u>VOLUME</u>               |

Advanced commands for controlling each stage of audio amplification:

|                        |   |   |
|------------------------|---|---|
| Set audio bass level   | BASS <u>CHANNEL</u> <u>BASS</u><br>Short form: ADB <u>CHANNEL</u> <u>BASS</u>               | BASS <u>CHANNEL</u> <u>BASS</u> <u>RESULT</u>                 |
| Read audio bass level  | BASS? <u>CHANNEL</u><br>Short form: ADB? <u>CHANNEL</u>                                     | BASS <u>CHANNEL</u> <u>BASS</u>                               |
| Set audio treble level | TREBLE <u>CHANNEL</u> <u>TREBLE</u><br>Short form: ADT <u>CHANNEL</u> <u>TREBLE</u>         | TREBLE <u>CHANNEL</u> <u>TREBLE</u> <u>RESULT</u>             |
| Read audio treble      | TREBLE? <u>CHANNEL</u><br>Short form: ADT? <u>CHANNEL</u>                                   | TREBLE <u>CHANNEL</u> <u>TREBLE</u>                           |
| Set audio midrange     | MIDRANGE <u>CHANNEL</u> <u>MID_RANGE</u><br>Short form: ADM <u>CHANNEL</u> <u>MID_RANGE</u> | MIDRANGE <u>CHANNEL</u> <u>MID_RANGE</u> <u>RESULT</u>        |
| Read audio midrange    | MIDRANGE? <u>CHANNEL</u><br>Short form: ADM? <u>CHANNEL</u>                                 | MIDRANGE <u>CHANNEL</u> <u>MID_RANGE</u>                      |
| Set audio loudness     | LOUDNESS <u>CHANNEL</u> <u>LOUDNESS</u><br>Short form: ADS <u>CHANNEL</u> <u>LOUDNESS</u>   | LOUDNESS <u>CHANNEL</u> <u>LOUDNESS</u> <u>RESULT</u>         |
| Read audio loudness    | LOUDNESS? <u>CHANNEL</u><br>Short form: ADS? <u>CHANNEL</u>                                 | LOUDNESS <u>CHANNEL</u> <u>LOUDNESS</u>                       |
| Set audio mix          | MIX <u>MIX-MODE</u>   | MIX <u>MIX-MODE</u> <u>RESULT</u>                             |
| Read audio mix         | MIX?  | MIX <u>MIX-MODE</u>   |
| Mute audio             | MUTE <u>MUTE-MODE</u>   | MUTE <u>MUTE-MODE</u> <u>RESULT</u>                           |
| Read audio mute state  | MUTE?   | MUTE <u>MUTE-MODE</u>   |
| Set stereo mode        | STEREO <u>STEREO-MODE</u>   | STEREO <u>STEREO-MODE</u> <u>RESULT</u>                       |
| Read stereo mode       | STEREO?   | STEREO <u>STEREO-MODE</u>                                     |
| Set balance mode       | BALANCE <u>OUT-CHANNEL</u> <u>BALANCE-LEVEL</u>   | BALANCE <u>OUT-CHANNEL</u> <u>BALANCE-LEVEL</u> <u>RESULT</u> |
| Read balance mode      | BALANCE? <u>OUT-CHANNEL</u>   | BALANCE <u>OUT-CHANNEL</u> <u>BALANCE-LEVEL</u>               |

**Parameter Description:****STAGE** = 'IN, 'OUT'

or

Numeric value of present audio processing stage. For example: '0' for input level, '1' for pre-amplifier, '2' for amplifier (OUT) etc.

**CHANNEL** = Input or Output #**VOLUME** / **BASS** / **TREBLE** / **MID\_RANGE** = Audio parameter in Kramer units, minus sign precedes negative values.

++ increase current value,

-- decrease current value.

**MIX** =

'0' or 'OFF'

'1' or 'ON'

## 2.8 Identification Commands

| Command                                | Syntax                   | Response   |
|--|--------------------------|--|
| Protocol handshaking                   | # <b>CR</b>              | ~OK <b>CRLF</b>  |
| Read device model                      | MODEL?                   | MODEL <b>MACHINE_MODEL</b>                                     |
| Read device serial number              | SN?                      | SN <b>SERIAL_NUMBER</b>  |
| Read device firmware version           | VERSION?                 | VERSION <b>MAJOR</b> <b>MINOR</b> <b>BUILD</b> <b>REVISION</b> |
| Set machine name                       | NAME <b>MACHINE_NAME</b> | NAME <b>MACHINE_NAME</b> <b>RESULT</b>                         |
| Read machine name                      | NAME?                    | NAME <b>MACHINE_NAME</b>                                       |
| Reset machine name to factory default* | NAME-RST                 | NAME-RST <b>MACHINE_FACTORY_NAME</b> <b>RESULT</b>             |

\***Note:** 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).

**MACHINE\_NAME** = Up to 14 alphanumeric chars.

\* **Machine factory name** = Model name + last 4 digits from serial number.

|                       |                                   |   |
|-----------------------|-----------------------------------|---|
| Set machine ID number | MACH-NUM<br><b>MACHINE_NUMBER</b> | MACH-NUM <b>OLD_MACHINE_NUMBER</b><br><b>NEW_MACHINE_NUMBER</b> <b>RESULT</b> |
|-----------------------|-----------------------------------|---|

\* A response is sent after the machine number was changed. The response with the header is:

**NEW\_MACHINE\_NUMBER** @MACH-NUM **OLD\_MACHINE\_NUMBER** **NEW\_MACHINE\_NUMBER** OK

## 2.9 Network Setting Commands

| Command             | Syntax  | Response                                      |
|---------------------|---|---|
| Set IP address      | NET-IP <u>IP_ADDRESS</u><br>Short form: NTIP        | NET-IP <u>IP_ADDRESS</u> <u>RESULT</u>        |
| Read IP address     | NET-IP?<br>Short form: NTIP?                        | NET-IP <u>IP_ADDRESS</u>                      |
| Read MAC address    | NET-MAC?<br>Short form: NTMC                        | NET-MAC <u>MAC_ADDRESS</u>                    |
| Set subnet mask     | NET-MASK <u>SUBNET_MASK</u><br>Short form: NTMSK    | NET-MASK <u>SUBNET_MASK</u> <u>RESULT</u>     |
| Read subnet mask    | NET-MASK?<br>Short form: NTMSK?                     | NET-MASK <u>SUBNET_MASK</u>                   |
| Set gateway address | NET-GATE <u>GATEWAY_ADDRESS</u><br>Short form: NTGT | NET-GATE <u>GATEWAY_ADDRESS</u> <u>RESULT</u> |
| Read subnet mask    | NET-GATE?<br>Short form: NTGT?                      | NET-GATE <u>GATEWAY_ADDRESS</u>               |
| Set DHCP mode       | NET-DHCP <u>DHCP_MODE</u><br>Short form: NTDH       | NET-DHCP <u>DHCP_MODE</u> <u>RESULT</u>       |
| Read subnet mask    | NET-DHCP?<br>Short form: NTDH?                      | NET-DHCP <u>DHCP_MODE</u>                     |

DHCP\_MODE =

'0' – Don't use DHCP (Use IP set by factory or IP set command).

'1' – Try to use DHCP, if unavailable use IP as above.

|                                  |  |  |
|----------------------------------|--|--|
| Change protocol<br>Ethernet port | ETH-PORT <u>PROTOCOL</u> <u>PORT</u><br>Short form: ETHP | ETH-PORT <u>PROTOCOL</u> <u>PORT</u> <u>RESULT</u> |
| Read protocol<br>Ethernet port   | ETH-PORT? <u>PROTOCOL</u><br>Short form: ETHP?           | ETH-PORT <u>PROTOCOL</u> <u>PORT</u>               |

PROTOCOL = TCP/UDP (transport layer protocol)

PORT = Ethernet port that accepts Protocol 3000 commands

1-65535 = User defined port

0 - Reset port to factory default (50000 for UDP, 5000 for TCP)

## 2.10 Machine Information Commands

| Command   | Syntax                | Response  |
|---|-----------------------|---|
| Set device time and date  | TIME <u>DATE_TIME</u> | TIME <u>DATE_TIME</u> <u>RESULT</u>   |
| Read device time and date   | TIME?                 | TIME? <u>DATE_TIME</u>  |
| <b>Note:</b> Time setting commands require administrator authorization. |                       |   |
| Read in/out count   | INFO-IO?              | INFO-IO: IN <u>INPUTS_COUNT</u> , OUT <u>OUTPUTS_COUNT</u>                  |
| Read max preset count   | INFO-PRST?            | INFO-PRST: VID <u>PRESET_VIDEO_COUNT</u> ,<br>AUD <u>PRESET_AUDIO_COUNT</u> |
| Execute firmware upgrade*   | UPGRADE               | UPGRADE OK  |

Firmware usually uploads to a device via a command like LDFW. The device may need to be reset to complete the process.

| Command                                | Syntax  | Response              |
|--|---------|-----------------------|
| Reset to factory default configuration | FACTORY | FACTORY <u>RESULT</u> |

## 2.11 Advanced Switching Commands

| Command                     | Syntax              | Response                          |
|-----------------------------|---------------------|-----------------------------------|
| Set audio follow video mode | AFV <u>AFV-MODE</u> | AFV <u>AFV-MODE</u> <u>RESULT</u> |

**Note:** This command affects the device front-panel mode and AUD/VID command.

|                              |      |                     |
|------------------------------|------|---------------------|
| Read audio follow video mode | AFV? | AFV <u>AFV-MODE</u> |
|------------------------------|------|---------------------|

AFV-MODE = Front panel AFV mode

'0' or 'afv' sets front panel switching buttons to audio-follow-video state.

'1' or 'brk' sets front panel switching buttons to their previous audio state.