



API Guide

MODEL:

KDS-EN6, KDS-DEC6

Contents

Overview	1
Terms	1
How to Access the Console	2
Astparam Configuration API	3
Overview	3
How to Access astparam	4
List of astparams	5
Getting KDS-6 Board's IP Address	24
Application Scenario:	25
FAQs:	26
Generic Configuration:	26
Unicast (KVM over IP)	32
Multicast (M to 1 group)	32
Multicast (M to N group)	33
IIC	33
UART	34
Console APIs	36
Debug Console APIs	36
Link Manager APIs	37
Video Over IP Driver Statistic/Control APIs	37
Display Driver Statistic/Control APIs	38
OSD APIs	38
Configurations APIs	40
IIC APIs	41
UART Port 1 APIs	41
Configuration Option List	42

Overview

This document describes the commands that can be used for KDS-6 console. KDS-6 uses Linux OS and the console is operated by BusyBox shell.

Using console APIs, developers can control KDS-6 firmware and extend the product's features and capabilities.



Kramer KDS-6 is based on Aspeed A6.4.3 firmware.

Default is Telnet port 24.

Use "root" to log in. No password is required.

Terms

- Commands starting with "\$" – Console commands executed under the Linux shell console. The default value of multicast IP (multicast_ip) is 225.0.10x.xxx. Please use this multicast_ip value.
- The 'ast_send_event -1 xxx' command has a new shortcut named 'e'. All of the following 'ast_send_evnt -1 xxx' commands can be replaced with the shortcut command like 'e xxx'. They are exactly the same.

How to Access the Console

- Telnet

Every KDS-6 host/client FW has Telnet and a Telnet client built in. a developer can use one KDS-6's Telnet client to connect to any other KDS-6 board. Or write their own program using Telnet protocol to connect to any KDS-6 board through the Ethernet network.



Default is Telnet port 24.

Use "root" to log in. No password is required.

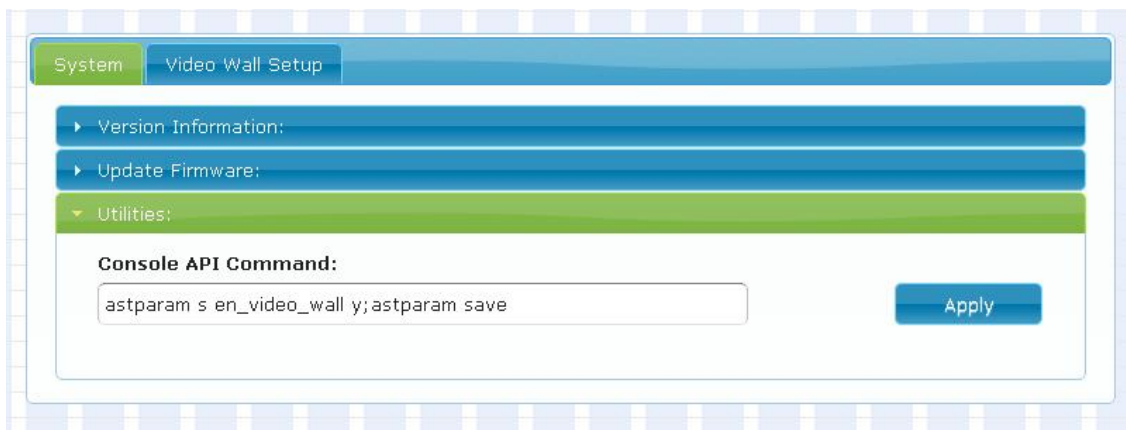
- Web UI

Use this approach when you need to modify astparam and prefer to enter commands via a web interface instead of a console.

To modify astparam via a web interface:

1. Connect to the device's Web UI
2. Enter astparam commands in "System->Utilities->Console API command:"

Note: You can enter multiple commands in a single line, separated by a semicolon (;).



Astparam Configuration API

Overview

KDS-6 firmware includes many configurable features. Most of the configuration are done using the “Configuration API” called *astparam*. *astparam* is the console command to access these system configurations.

astparam can be saved in flash ROM, so that system configuration is not lost when the system loses power. Typical applications of *astparam* are:

- Enable/disable firmware features like serial over IP or video wall.
- Configure factory default firmware features on a production line.
- Prepare an SPI flash image for production, including default *astparam* settings.
- Enable/Disable firmware features on end customer side.

The format of *astparam* is a key=value string pair. For example, a *astparam*: “en_video_wall=y” means *astparam* key “en_video_wall”’s value is “y”. Both “key” and “value” are ANSI string and case sensitive. The firmware decides how to use it.

Note: The firmware does not validate the *astparam*. So, any type error could cause unexpected results. The advantage is that a user can add any *astparam* key/value pair, even if KDS-6’s reference firmware doesn’t need it. Both RO and RW *astparams* can be modified or even removed.

The firmware reserves two flash partitions for system configuration. One is read only (RO) the other is read-write (RW).

Note: BOTH ‘RO’ and ‘RW’ partitions are actually read/writable. Their usage is just defined differently.

The firmware treats the setting in the RO partition as the system’s factory default settings. It is recommended to save non-factory default changes in the RW partition (including the random ethaddr). When firmware boots up, it takes *astparam* in the following priority order: RW > RO > firmware default. Firmware uses the value in RW if the *astparam* exists. If the *astparam* is not available in RW partition, then the value in RO partition is used. Firmware hard coded value is used only when the *astparam* doesn’t exist in both RO and RW partition.

When the firmware starts up, both RO and RW configurations are read from flash and saved in RAM.

Note: All configuration modifications are saved in RAM until the "astparam save [ro]" command is called.

The process of "reset_to_default.sh" erases the RW partition but keeps the 'ethaddr' value (by rewriting it after partition erase.). Since everything in the RW is gone, the firmware takes values from the RO partition, which is the 'factory default'.

How to Access astparam

The most fundamental and powerful way to access astparam is through the "astparam" Console API command. Most of alternative approaches are just a wrap of the "astparam" command.

```
$ astparam OPTIONS [KEY] [VALUE]
```

OPTIONS

- r : read from RO partition cache file. [KEY] is required.
- w: write to RO partition cache file (**not save to flash ROM yet**). [KEY] and [VALUE] are required. If [VALUE] is empty the [KEY] will be removed.
- g : read from RW partition cache file. [KEY] is required.
- s : write to RW partition cache file (**not save to flash ROM yet**). [KEY] and [VALUE] are required. If [VALUE] is empty the [KEY] will be removed.
- flush : clear all settings in RW partition cache file including random generated MAC address. "./astparam save" is needed to clear all settings in flash ROM.
- dump: dump all parameters in RW partition cache file.
- dump ro: dump all parameters in RO partition cache file.
- dd: dump all parameters in RW partition directly from flash ROM.
- dd ro: dump all parameters in RO partition directly from flash ROM.
- save: **save all parameters in RW partition cache file into flash ROM.**
- save ro: **save all parameters in RO partition cache file into flash ROM.**

Examples:

- Read "ip_mode" setting from RO partition cache file:
\$./astparam r ip_mode
- Read "ip_mode" setting from RW partition cache file:
\$./astparam g ip_mode
- Write "ip_mode" setting as "autoip" to RW partition cache file:
\$./astparam s ip_mode autoip
- Clear all settings in RW partition cache file:
\$./astparam flush
- Clear ip_mode setting in RW partition cache file:
\$./astparam s ip_mode
- Write RW partition cache file back to SPI flash ROM's RW partition:

```
$/astparam save
```

- Write RO partition cache file back to SPI flash ROM's RO partition:
\$/astparam save ro

List of astparams

Generic Settings

Key	Description	Value (bold is default)	Host/ Client
scenario		pc2tv: usb_only:	H/C
astaccess	Host: 'y' boot up default accept connection from client. Client: 'y' boot up default auto connect to host.	y: n:	H/C
multicast_on	Configure casting mode	y: Multicast mode n: Unicast mode	H/C
ui_feature		y: n:	C
ignore_e_button_link		y: n:	H/C
ignore_e_button_link_1		y: n:	H/C
pwr_led_type		default: share_usb:	H/C
en_log	'y' show debug message on debug console.	y: n:	H/C
stop_on_boot_test_fail		y: n:	H/C
telnetd_param	'-p 24': use IP port 24.	-p 24	H/C
lm_link_off_timeout		15: 15 seconds	C

Key	Description	Value (bold is default)	Host/ Client
cs_gpio_inv	[>= A5.3.3] Invert 4-bits DIP switch value. Default value is '0000', means all 4bits (CH0,CH1,CH2,CH3) no invert. To invert CH1, for example, set the value to '0100'.	0000	H/C
soc_op_mode	[>= AST1520][>= A6.2.0] Config backward compatibility mode.	0: as it is 1: AST1500 compatible 2: AST1510 compatible	H/C
web_ui_cfg	Used to customize Web UI. The default value is automatically generated based on enabled functions during first time boot up. The value is a series of characters: e: essential functions (can't be disabled) n: network v: video w: video wall a: audio (i2s) u: usb s: serial over ip r: IR Character order does not matter. Example 1: To enable all UI elements, set the value to 'envwausr'. Example 2: 'envas' means show 'essential functions', 'network', 'video', 'audio', and 'serial over ip' modules.		C
board_revision	[>= AST1520] Used to indicate firmware different kind of HW board design. 300: [Host] AST152x v1.x reference board. 301: [Host] AST1520 v2.x splitter loopback design with HDCP 2.2 support	300: 301:	H

Key	Description	Value (bold is default)	Host/ Client
ui_default_res	[>= A6.1.0] Change the bootup and GUI default resolution. Default resolution is 640x480@60Hz. Note: System reboot is required for ui_default_res to be applied.	640x480@60	C
ui_show_text	'y' to display diagnostic information on the bottom of the GUI. 'n' to hide it.	y: Display n: Hide	C

IP Network Setting

Key	Description	Value (bold is default)	Host/ Client
ip_mode		static: dhcp: autoip:	H/C
ipaddr			H/C
netmask			H/C
gatewayip	[> A 1.38]		H/C
hostname_prefix		ast: For AST1500 ast2: For AST1510 ast3: For AST1520	H/C
hostnamebydipswitch	Use default hostname_id based on 4-bits DIP switch. Host default is 'y' Client default is 'n'	y: n:	H/C
hostname_id	Overwrite "hostnamebydipswitch"		H/C
multicast_ip			H/C
ch_select			H/C
reset_ch_on_boot	Resets link related astparam on boot, to prevent the user from entering the wrong channel setting. The following astparam is cleared to the default value: ch_select multicast_ip hostname_id	y: enable n: disable	H/C
jumbo_mtu	[>= AST1510]	1500: 8000:	H/C
multicast_ip_prefix	Uses a different multicast IP prefix. Default value is '225.0.10' which means multicast IP is '225.0.10x.xxx' where x's value depends on 4-bits DIP switch.	225.0.10	H/C

Button: (> A3.2)

Key	Description	Value (bold is default)	Host/ Client
btn1_short			H/C
btn1_long			H/C
btn1_short_on_boot			H/C
btn1_long_on_boot			H/C
btn1_short_on_eth_of f			H/C
btn1_long_on_eth_off			H/C
btn2_short			H/C
btn2_long			H/C
btn2_short_on_boot			H/C
btn2_long_on_boot			H/C
btn2_short_on_eth_of f			H/C
btn2_long_on_eth_off			H/C
btn_init			H/C
btn1_delay			H/C
btn2_delay			H/C

Video over IP

Key	Description	Value (bold is default)	Host/ Client
no_video		y: disable VideoIP n: enable VideoIP	H/C
ast_video_quality_mode		-1: Video mode 0: Graphic mode	H
v_bcd_threshold		0: anti-dither off 1: anti-dither 1 2: anti-dither 2	H
v_1080i_to_720p			H/C
en_video_wall		y: n:	H/C
profile	[< A6.0.0] Select 'giga' or 'analog'. [>= A6.x.x] Revised the naming to: auto, 10M, 50M, 100M, 150M, 200M. SoC V1 and V2 only support 'auto'.	auto: 10M: 50M: 100M: 150M: 200M:	H
v_chunk_size	[OP_MODE < 3] Used to control video network stream's burst size. A smaller value smooths the Ethernet switch's loading, but may causes video streaming performance issues. Default value is '64512'. Use '7376' when you want to smooth the video stream packets. Don't touch this setting unless you know what you are doing.	64512	H
v_ignore_cts7_33	HDMI compliant test (CTS 7-33) requires the HDMI source to always output a DVI signal when the downstream sink is not an HDMI sink (EDID is not HDMI device). Set this option to 'y' to ignore this requirement.	n: follow CTS y: ignore CTS	H/C
v_type	[>= AST1520] Overwrite "video type" setting. Default value is received from the GPIO pin (HDMI or VGA).	0: VGA 1: disable 2: HDMI 3: DVI	H/C

Key	Description	Value (bold is default)	Host/Client
v_loopback_type	<p>[>= AST1520] Set video loopback port type. Overwrite HW GPIO setting. Default value is gotten from GPIO pin.</p> <p>Default value when GPIO pin is set to enable:</p> <p>AST1525: the same as v_type setting.</p> <p>AST1520:</p> <p>'board_revision' == 301, default 2 (HDMI).</p> <p>If the board is '301' type and has VGA loopback, please set v_loopback_type to '0'.</p> <p>v_dual_port to '1'.</p> <p>Otherwise</p> <p>[<= A6.2.6] Default as v_type setting.</p> <p>[> A6.2.6] Default as 0 (VGA).</p>	<p>0: VGA</p> <p>1: disable</p> <p>2: HDMI</p> <p>3: DVI</p>	H
v_output_timing_convert	<p>[>= AST1520] Force specific video output timing.</p> <p>Examples:</p> <p>00000000: Pass-Through</p> <p>80000010: Full HD 1080p60</p> <p>8000001F: Full HD 1080p50</p> <p>8000005F: Ultra HD 2160p30</p> <p>8000005E: Ultra HD 2160p25</p> <p>80000004: HD 720p60</p> <p>81000061: WXGA 1366x768@60</p> <p>81000040: WXGA+ 1440x900@60</p> <p>81000051: WUXGA 1920x1200@60</p> <p>8100003C: SXGA+ 1400x1050@60</p>	0: Pass-Through	C
v_eng_drv_option	<p>[>= AST1520] 32bits bitmap VE driver options to tweak driver behavior. Default value is 0.</p> <p>0x00000001: [>=A6.0.0][Host] Convert 2160p30Hz to 1080p</p> <p>0x00000002: [>=A6.0.0][Host] Convert 2160p60Hz (YUV420) to 1080p60Hz</p> <p>0x00010000: [>=A6.1.0][Client] Do not black out after switching to new video source</p>	0: Default value	H/C

Key	Description	Value (bold is default)	Host/ Client
v_dual_port	[>= AST1520] Dual output mode. Client set to '1' to output video signal to both VGA and HDMI port. Host set to '1' to output HDMI input to VGA loopback port. Default value: Host: 0 Client: 1	1: enable 0: disable	H/C
v_rx_drv	[>= A6.0.0] Used to specify HDMI receiver driver used. Default value for different SoC version: AST1500: cat6023 AST1510: cat6023 AST1520: it6802 AST1525: it6802	cat6023: for IT6604, IT6605 chip it6802: for IT6802 chip	H
v_src_unavailable_timeout	Wait for a v_src_unavailable_timeout ms before entering power save. Set to '-1' means never timeout.	10000: 10 seconds -1: never timeout	C
v_turn_off_screen_on_pwr_save	Turn off video output sync after connected host stops streaming video to client.	y: Turn off n: Do not turn off. Will switch back to GUI screen	C
v_rx_drv_option	Special bitmap options for HDMI receiver driver. (IT6604, IT6802...) 0x00000002: Always report HDMI audio as 2Ch 48KHz 0x00000004: Always report HDMI audio is ON	0: default	H
v_analog_edge_detect	[>= A6.1.12] For VGA analog video input only. Used to adjust edge detection threshold value. Default value should be fine in most of cases. Increase the value when edge detection is not accurate. Valid range is 0~255. Good value range: 15~64. Set the value too high or too low may cause bad result.	37: default value	H
loopback_edid_patch	Copy EDID patch option for local loopback video port.	00000000	H

Key	Description	Value (bold is default)	Host/ Client
remote_edid_patch	Copy EDID patch option for remote client. Default values: AST1500: 00230017 AST1510: 00000005 AST1520/AST1525: [FW < A6.3.0] 00000000 [FW >= A6.3.0] 00000001		H
v_input_select	[>=AST1520][>=A6.3.0] Configure different types of input select mode: auto: auto detect. always controlled by GPIO. detect_sync: auto switch after video lost fixed: fixed selection	auto detect_sync fixed	H
v_dual_input	[>=AST1520][>=A6.3.0] Specify if the HW board has dual input port or not. This astparam is optional. The FW will automatically detect the HW configuration based on the availability of EDID ROM of both ports.	y: has n: doesn't have	H

Key	Description	Value (bold is default)	Host/ Client
edid_use	<p>By design, all connected client and host loopback video port's EDID is received by host's VE driver. All EDID information also contains the 'edid_use' flag. The host VE driver will ONLY write the received EDID into EEPROM when the 'edid_use' is set to 'primary'. So, 'edid_use' flag is used to control whether a host should write a received EDID into EEPROM or not. Different casting mode has different 'edid_use' default value.</p> <p>By default:</p> <p>Unicast Mode: All client's 'edid_use' will ALWAYS be 'primary'. host loopback's 'edid_use' default is 'primary'.</p> <p>Multicast Mode: All host and client 'edid_use' default is 'secondary'. And "press and hold Button 2 on device reboot" can force the device to be 'primary'.</p> <p>In most of cases, leaving 'edid_use' as default is fine.</p>	<p>primary:</p> <p>secondary:</p>	H/C
v_hdmi_hdr_mode	[FW >= A6.3.0]	<p>0: HDR passthrough</p> <p>1: Force HDR off</p>	C

Video Wall

Key	Description	Value (bold is default)	Host/ Client
en_video_wall	Enable video wall feature. Please also check the value of astparam ,web_ui_cfg, for the video wall Web UI.	y: enable n: disable	H/C
vw_rotate	[>= AST1520] Clock-wise rotate output picture. Console API: echo 3 > /sys/devices/platform/display/vw_rotate	0: No rotate 3: 180 degree 6: 270 degree	C

HDCP

Key	Description	Value (bold is default)	Host/ Client
hdcp_always_on	[Host >=A5.3.1] [Client >=A6.1.0] Always enable HDCP 1.4	y: enable n: disable	H/C
hdcp_always_on_22	[Host >=A6.0.0] [Client >=A6.1.0] Always enable HDCP 2.2.	y: enable n: disable	H/C
v_reject_hdcp	Don't accept HDCP video source.	y: enable n: disable	H
hdcp_cts_option	[FW >= A6.2.0] 32bits bitmap options for HDCP CTS test. 0x80000000: Disable auto Sil9678/9679 firmware update. 0x00000001: HDCP 2.2 Only firmware 0x00000002: HDCP 1.x Only firmware	0x00000000:	H/C

USB over IP

Key	Description	Value (bold is default)	Host/ Client
no_usb		y: n:	H/C
share_usb	Default value: (when 'share_usb_auto_mode' is 'n') Multicast mode: 'share_usb = y'. Unicast mode: 'share_usb = n'.	y: n:	H/C
share_usb_auto_mode	When 'share_usb_auto_mode == y', 'share_usb' setting will automatically change depends on the setting of casting mode ('multicast_on').	y: n:	H/C
share_usb_on_first_peer	[>A6.1.8] Applied to 'share_usb == y' mode. When value is 'y', the first connected client automatically requests USBoIP function. By enabling this feature, USBoIP automatically enabled when there is only one client connected to host under multicast mode.	y: auto connect first peer n:	H
usb_set_addr_hack	It is a hack for some buggy BIOS to be able to recognize USBoIP devices. The problem is rarely seen.	0: disable 1: enable	H
usb_hid_urb_interval	Set to 35 to resolve some USB HID long latency issue. Applies to both KMoIP and USBoIP.	0: ASAP 35: 35ms	H
usb_quirk	[>=A5.3.3] 32bits bitmap options for resolving some vhub compatibility issues. VHUB_QUIRK_BULK_SSPLIT (0x1UL << 0)	0:	H
usb_fast_switch	Set to 'y' to enable 'usb fast switch' mode. This enhances about 2 seconds of USB switching time, but sacrifices video switching time from 2 to 5 seconds.	n: Disable y: Enable	C

USB over IP Exporting Policy: (> A 5.2.2)

Key	Description	Value (bold is default)	Host/ Client
usb_default_policy	The default/global exporting policy used when a specific policy is not available.	auto_export no_auto_export	C
usb_conflict_policy	Policy used when a device has different interfaces with conflict exporting policy.	auto_export no_auto_export	C
usb_disable_classes (>A5.1.0 b1701)	Policy setup to disable per USB classes. There is no default value. Each class should be separated by a space. For example, usb_disable_classes=audio video printer.	per_interface audio comm hid physical still_image printer mass_storage cdc_data cscid content_sec video personal_healthcar e audio_video diagnostic wireless_controller misc app_spec vendor_spec	C
usb_enable_classes	Policy setup to enable per USB classes. There is no default value. Each class should be separated by a space. For example, usb_enable_classes=audio video printer.	same as "usb_disable_class es"	C
usb_disable_devices	Policy setup to disable per USB's PnP ID. There is no default value. Each device should be separated by a space. For example, usb_disable_devices=0781:000c 0770:1234.		C
usb_enable_devices	Policy setup to enable per USB's PnP ID.		C

	<p>There is no default value. Each device should be separated by a space. For example, usb_enable_devices=0781:000c 0770:1234.</p>		
--	--	--	--

KMoIP: (> A 5.1.0)

Key	Description	Value (bold is default)	Host/ Client
no_kmoip	Enable/disable KMoIP function.	y: disable KMoIP n: enable KMoIP	H/C
kmoip_ports	Specify which client USB port can use KMoIP function when USBoIP session is running. The value can be 'all' or a list of port numbers, Ex: '1 2'. USB port number can be re-mapped by 'usb_busid_map' astparam.	all: all USB ports 1: USB port 1 2: USB port 2 3: USB port 3 4: USB port 4 5: USB port 5 Note: Firmware version A5.1.0 ~ A5.1.4 uses 'all' as default value. The value was changed to '1 2' starting from A5.2.1.	C
kmoip_hotkeys	Specify 8 set of hotkeys.		C
kmoip_token_interval	This is a per client setting. Set the idle timeout value (in milliseconds) for holding an access token. For example, the default value is 100ms, which means once client A acquires keyboard/mouse access, other clients can only successfully acquire the access (token) after client A is idle longer than 100ms. Normally, you don't need to set this value (leave it as default). It is used to work around a bad mouse, which keeps reporting pointer activities and hence blocks all other client keyboard/mouse input.	100	C
usb_busid_map	A mapping list to map USB busid to USB port. Use default value unless you have a different hardware layout.		C
kmoip_poll_interval	[FW >= A6.3.0] Used to set KMoIP's USB poll interval. Both host and client must be set to the same value. Unit is in ms. Default value should be good for most		H/C

	<p>cases: AST1500: 35 AST1510: 35 AST1520: 10 [FW >=A6.3.0], 35 [FW < A6.3.0]</p>		
--	---	--	--

Audio over IP (I2S over IP)

Key	Description	Value (bold is default)	Host/ Client
no_i2s		y: disable I2SoIP n: enable I2SoIP	H/C
a_io_select	<p>[FW >= b1733] Used to select audio input source.</p> <p>Host: auto: auto select analog when host's analog line in plugged hdmi: fixed hdmi audio analog: fixed analog audio/in</p> <p>Client: auto: auto dual output. hdmi: fixed hdmi audio out analog: fixed analog audio out/in</p> <p>[FW >= A6.1.1][>=AST152x] Set this on client to select audio output port. Default is 'auto' which means dual output.</p> <p>[FW >= A6.3.0][>=AST1510][Host Only] Host add extra 'auto_1' and 'auto_2' mode and redefined 'auto'. auto: When HDMI/DVI video port is used, auto select audio port per analog line in plug. When VGA video port is used, ALWAYS use analog audio port. auto_1: Always select audio port per analog line in plug. auto_2: Always select audio port per video port used. When HDMI/DVI video port is used, HDMI always use HDMI audio port. When VGA video port is used, always use analog audio port.</p>	auto: hdmi: analog: [FW>= A6.3.0][Host Only] auto_1: auto_2:	H/C

Key	Description	Value (bold is default)	Host/ Client
a_analog_in_vol	<p>[>=AST1510][FW >= A6.3.0] Set analog audio's input volume. Range from 0 to 100 %. Default value '-1' means use driver built-in default value.</p> <p>This setting requires system reset to take effect. To runtime change the volume, please use following command:</p> <pre>echo [value] > /sys/devices/platform/150 0_i2s/analog_in_vol</pre>	-1,0,1,2 ~,100	H/C
a_analog_out_vol	<p>[>=AST1510][FW >= A6.3.0] Set analog audio's output volume. Range from 0 to 100 %. Default value '-1' means use driver built-in default value.</p> <p>This setting requires system reset to take effect. To runtime change the volume, please use following command:</p> <pre>echo [value] > /sys/devices/platform/150 0_i2s/analog_out_vol</pre>	-1,0,1,2 ~,100	H/C

IR over IP

Key	Description	Value (bold is default)	Host/ Client
no_ir		y: n:	H/C

Serial over IP

Key	Description	Value (bold is default)	Host/ Client
no_soip		y: Disable SoIP n: Enable SoIP	H/C
soip_type2		y: n:	H/C
soip_guest_on		y: n:	H/C
s0_baudrate		115200-8n1	H/C
soip_type2_token_timeout	[>=A1.38]	1,2,3... seconds	H/C
ch_select_soip2	[>=A5.3.0]	N/A	C

GPIO over IP (Push Button over IP)

Key	Description	Value (bold is default)	Host/ Client
no_pwrbtn		y: n:	H/C

Getting KDS-6 Board's IP Address

Kramer KDS-6 FW default runs "static ip" and uses the following address: 192.168.1.39.

If the device is set to "autoip" and uses 169.254.xxx.xxx private IP domain, the IP of the target KDS-6 board can be resolved by its hostname using mDNS protocol..

The following are the methods of resolving the IP address using "autoip" mode:

- Client's GUI – The client's GUI displays the client's IP address and the connected host's IP address.
- Console APIs – Two console APIs are provided:
 - astresname: use this command to resolve a target's IP address by providing target's hostname.
 - node_list: use this command to list all KDS-6 boards' hostname and IP address.
- From PC – Use the mDNS libraries, such as:
 - Bonjour
 - pyZeroConfig
- From PC: Developer can reference to "node_list" sample code to implement PC version of node_list.

Application Scenario:

With Console APIs, developers have more controls over KDS-6 FW, enabling development of new application scenarios. Console APIs are designed to support following application scenarios:

- KVM over IP: Extension of built-in 1 to 1 unicast mode. The client is able to choose which host to connect to. Console APIs provide APIs for developers to enable a client to connect to a specific host.
- Multicast M to 1 group: Extension of built-in multicast mode. In this mode, there are multiple video sources but a single group of monitors. This group of clients always displays the same video source from one of the hosts. Console APIs provide APIs to enable a host to change video source for display.
- Multicast M to N group: Extension of built-in multicast M to N mode. In this mode, there are multiple video sources and multiple groups of monitors. Console APIs provide APIs to enable clients to join different multicast groups.
- Extra IIC remote communication:
- Extra UART port remote communication:

FAQs

Generic Configuration:

How do I discover all KDS-6 boards in the same LAN?

1. Use Telnet or the debug console to connect to the target board.
2. Start discover all: `./node_list`
3. Start discover hosts: `./node_list -t host`
4. Start discover clients: `./node_list -t client`

The results appear on the console.

How do I use the node_list feature without going through KDS-6 board? For example, using a PC.

Request a "node_list" sample code.

How do I find the target client board's current multicast group?

1. Use Telnet or the debug console to connect to the target board.
2. Get the current multicast ip: `./astparam g multicast_ip`.
3. To read the default value from the DIP switch: `./astconfig channel`

The result is a four bits value (B0B1B2B3). that maps to multicast ip 225.0.B0.B1B2B3.

How do I turn on/off the debug console's debug message?

- To turn off: `$dmesg -n 1 && setconsole /dev/ttyS1`
- To turn on: `$dmesg -n 8 && setconsole -r`

How do I configure IP settings?

KDS-6 firmware supports three kinds of ip mode settings:

- autoip: 169.254.xxx.xxx private IP domain is used and auto-generated on boot.
- dhcp client: Use DHCP client to get an IP address. A DHCP server is required to use this mode.
- static ip: Use a static IP address, that can be configured through the Console API.

1. To configure the IP mode: `./astparam s ip_mode mode` (where "mode" can be: autoip, dhcp, static)
2. To configure a static IP address:
 - IP: `./astparam s ipaddr xxx.xxx.xxx.xxx`
 - Netmask: `./astparam s netmask xxx.xxx.xxx.xxx`
 - Default gateway: `./astparam s gatewayip xxx.xxx.xxx.xxx`
3. Reboot the system to apply the above IP configuration changes.

How do I configure hostname_id?

Host:

1. hostname_id will overwrite "hostnamebydipswitch" setting.
2. Set hostname_id by: `./astparam s hostname_id XXXX`
3. If you want to save the setting into flash and keeps the setting after reboot:
 - `./astparam s reset_ch_on_boot n`
 - `./astparam save`
4. Trigger hostname change: `./ast_send_event -1 e_chg_hostname`

Client:

1. Set hostname_id by: `./astparam s hostname_id XXXX`
2. If you want to save the setting to flash and keep the setting after reboot:
 - `./astparam s reset_ch_on_boot n`
 - `./astparam save`
3. Trigger hostname change: `./ast_send_event -1 e_chg_hostname`
Note: See hostname_id in the [Configuration Option List](#) section on page [42](#) for details

What is the relationship between hostname and hostname_id?

KDS-6 board's "hostname" is constructed by "hostname_id". For example:

- For a host board with "hostname_id" as "ABCD", the board's "hostname" is "ast-gatewayABCD".
The board's IP can be resolved by `./astresname ast-gatewayABCD.local`
- For client board with "hostname_id" as "1234", the board's "hostname" is "ast-client1234".
The board's IP can be resolved by `./astresname ast-client1234.local`

How do I dynamically toggle between Graphic Mode and Video Mode?

1. Use Telnet or the debug console to connect to the host board.
2. Current quality mode can be get by: `$cat /sys/devices/platform/videoip/QualityMode`
3. Toggle quality mode by sending a "button 2" press event:
`./ast_send_event -1 e_button_pairing`

How do I dynamically change Anti-Dithering mode?

1. Use Telnet or the debug console to connect to the host board.
2. Get the current quality mode: `$cat /sys/devices/platform/videoip/bcd_threshold`
3. Toggle anti-dithering mode by sending a "button 2 long press" event:
`./ast_send_event -1 e_button_pairing_1`

How do I dynamically change snoop mode?

1. Use Telnet or the debug console to connect to the host board.
2. Get the current video loopback state:
`$cat /sys/devices/platform/videoip/LoopbackEnable`
 - 0: currently not enabled
 - 1: enabled
3. Toggle "snoop" mode by sending a "button 2" long press event:
`./ast_send_event -1 e_button_pairing_1`

How do I dynamically connect/disconnect a client?

1. Use Telnet or the debug console to connect to the client board.
2. Get the current link manager state: `$cat /var/ast_device_status`
See [Link Manager APIs](#) on page [37](#) for details.
3. Toggle the "link" mode by sending a "button 1" event UNDER stable state:
Wait state "s_idle" or "s_srv_on"
`./ast_send_event -1 e_button_link`

How do I dynamically enable/disable remote access (host)?

1. Use Telnet or the debug console to connect to the host board.
2. Get the current link manager state: `$cat /var/ast_device_status`
See [Link Manager APIs](#) on page [37](#) for details

3. Toggle the "link" mode by sending a "button 1" event UNDER stable state:
Wait state "s_idle" or "s_srv_on" or "s_attaching"
`$/ast_send_event -1 e_button_link`

How do I force update EDID of a target client?

1. Use Telnet or the debug console to connect to the client board.
2. Wait for link manager state "s_srv_on"
`$echo primary > /sys/devices/platform/videoip/edid_use`

How do I force update EDID of the video loopback port (host)?

1. Use Telnet or the debug console to connect to the host board.
2. Wait for link manager state "s_srv_on" or "s_attaching"
`$echo primary > /sys/devices/platform/videoip/edid_use`

Under multicast USB mode, how do I know the current USB state of a client?

1. Use Telnet or the debug console to connect to the client board.
2. Dump the link manager state by:
`$/ast_send_event -1 e_debug`
`/usr/local/bin$ cat lm_params`
[FW >= A3.20]
`lmparam g SHARE_USB_STATE`
3. The "SHARE_USB_STATE" is the current USB state.
s_idle: Stopped USB redirection
s_srv_on: Using USB redirection
others states: Not stable state

How do I find the status of the video source?

- When video source is not available:
 - Host: `$cat /sys/devices/platform/videoip/State` will be DETECTING_MODE
 - Client: `$cat /sys/devices/platform/videoip/State` will be WAITING_HOST_MODE
- When video is streaming
 - Host: `$cat /sys/devices/platform/videoip/State` will be OPERATING
 - Client: `$cat /sys/devices/platform/videoip/State` will be OPERATING

How do I check if the Ethernet cable is connected and works?

```
$cat /sys/devices/platform/ftgmac/link_state
```

How do I know which client (IP) is using the host's USB redirection under multicast mode?

1. Use Telnet or the debug console to connect to the host board.
2. Request link manager's status by: `./ast_send_event -1 e_debug`
3. Get the client's IP address by: `/usr/local/bin$ cat lm_params | grep USB_CLIENT_IP`

How do I know whether a monitor is connected?

1. Use Telnet or the debug console to connect to the host or client board in interests.
2. `$cat /sys/devices/platform/display/monitor_info | grep "attached="`

How do I dump the attached monitor's EDID?

1. Use Telnet or the debug console to connect to the host or client board in interests.
2. `$cat /sys/devices/platform/display/monitor_info`

How do I know if the video timing (resolution...) is in use?

1. Use Telnet or the debug console to connect to the client board.
2. `$cat /sys/devices/platform/display/timing_info`

How do I get the device's hostname?

1. Use Telnet or the debug console to connect to the device.
2. For the short version of hostname: `$lmparam g HOSTNAME_ID`
3. For the long version of hostname: `$hostname`

How do I get the device's IP address?

1. Use Telnet or the debug console to connect to the device.
2. `$lmparam g MY_IP`

How do I get the device's MAC address?

1. Use Telnet or the debug console to connect to the device.
2. `$lmparam g MY_MAC`
3. [A3.50] Query current connected host IP on client:
4. Use Telnet or the debug console to connect to the client.
5. `$lmparam g GWIP`

How do I stop a client's connection?

1. Use Telnet or the debug console to connect to the client.
2. `$e e_stop_link`

How can a client request USB redirection access under multicast mode?

1. Use Telnet or the debug console to connect to the client.
2. `$e msg_e_request_usb`

Note: If the client currently has USB redirection access, the above command stops the redirection session. You can query the current USB redirection status to avoid this condition.

How do I configure as multicast mode?

1. Use Telnet or the debug console to connect to the target device.
2. `$astparam s multicast_on y`
3. Config USB over IP mode:
 - `$astparam s share_usb_auto_mode y`
4. `$astparam save`
5. `$reboot`

How do I configure as unicast mode?

1. Use Telnet or the debug console to connect to the target device.
2. `#astparam s multicast_on n`
3. Config USB over IP mode:
 - [\geq A3.54]: `$astparam s share_usb_auto_mode y`
 - [$<$ A3.54]: `$astparam s share_usb n`
4. `$astparam save`
5. `$reboot`

Can I overwrite multicast IP?

There are 2 ways to overwrite the multicast IP:

- `astparam "multicast_ip"` – you must set `"reset_ch_on_boot"` to `"n"` and do the `"astparam save"` command to save it into flash ROM, so that the setting won't disappear after box reset.

- `astparam "multicast_ip_prefix"` – used to construct the default multicast ip value's prefix. When "multicast_ip" is not set, firmware is construct default multicast ip as `$(MULTICAST_IP_PREFIX){ch0}.{ch1}{ch2}{ch3}`. For example, the default value of "multicast_ip_prefix" is "225.0.10". So, for a DIP switch setting of 0110, the default multicast ip is 225.0.100.110.

Unicast (KVM over IP)

How can a client connect to a specified host under unicast mode?

1. Use Telnet or the debug console to connect to the client board.
2. [\geq A5.1.0] Change multicast ip: `./astparam s multicast_ip 225.0.B0.B1B2B3 =>`
where B0,B1,B2,B3 are mapped to the 4-bits DIP switch
3. Set target `ch_select` as target host's `hostname_id` or IP:
`hostname_id: ./astparam s ch_select hostname_id`
`IP: ./astparam s ch_select xxx.xxx.xxx.xxx`
4. Trigger client to re-connect: `./ast_send_event -1 e_reconnect`

Multicast (M to 1 group)

How to change Host's multicast group from XXXX to 1001?

1. Use Telnet or the debug console to connect to the host board.
2. Stop Link: `./ast_send_event -1 e_stop_link`
3. Change multicast ip: `./astparam s multicast_ip 225.0.1.001`
4. Change the `hostname_id`: `./astparam s hostname_id 1001`
5. Trigger hostname change: `./ast_send_event -1 e_chg_hostname`
6. If you want to save the setting into flash and keep the setting after reboot:
`./astparam s reset_ch_on_boot n`
`./astparam save`
7. Restart Link: `./ast_send_event -1 e_reconnect`

Multicast (M to N group)

How to change the target client board's multicast group?

1. Use Telnet or the debug console to connect to the target board.
2. Change multicast ip: `./astparam s multicast_ip 225.0.B0.B1B2B3 =>` where B0,B1,B2,B3 are mapped to the 4-bits DIP switch
3. Change the ch_select to connect to: `./astparam s ch_select B0B1B2B3`
4. If you want to save the setting into flash and keep the setting after reboot:
`./astparam s reset_ch_on_boot n`
`./astparam save`
5. Trigger client to re-connect: `./ast_send_event -1 e_reconnect`

IIC

How to read/write target board's IIC channel?

1. Use Telnet or the debug console to connect to the target board.
2. All following values are treated as hex number
3. [\geq A3.0] Initial IIC bus: `$echo BusNum BusSpeed > /sys/devices/platform/i2c/bus_init`
4. Setup the IIC slave's address: `$echo BusNum DevAddr > /sys/devices/platform/i2c/io_select`
5. To read offset X:
`$echo X 1 > /sys/devices/platform/i2c/i_range`
`$cat /sys/devices/platform/i2c/io_value`
6. To write offset X to value Y: `$echo X Y > /sys/devices/platform/i2c/io_value`

Why is the sysfs approach used for IIC access?

IIC access consumes a very high percentage of the CPU resources. Using the sysfs approach, we ensure that developers' IIC access won't impact system performance.

UART

How to read/write the target board's UART port 1 (/dev/ttyS0)?

1. Make sure "Serial over IP" feature is disabled. (Reboot is needed for taking effect.)
2. Use telnet or debug console to attach to target board.
3. [~A1.32]
 - Use stty to configure the UART port 1. Ex: `$stty 115200 -F /dev/ttyS0` => set UART port 1's baud rate to 115200.
 - To start read: `$cat /dev/ttyS0 &` => UART data will be redirected to console
 - To stop read: `$pkill cat`
 - To start write: `$cat > /dev/ttyS0` => console input will be redirected to UART
 - To stop write: press "ctrl+c" (0x03)
4. [A1.33~]
 - Use "microcom" to direct access /dev/ttyS0.
ex: `microcom -t 10000 -s 115200 -X /dev/ttyS0`
Please google "busybox microcom" for usage details

How to configure UART port 1 (/dev/ttyS0) for non-Serial over IP usage?

- UART port 1 is used for "Serial over IP" feature. Developers who want to manually control UART port 1 need to disable "Serial over IP" feature.
- To configure UART port 1, please use "stty" command. See [Console APIs](#) on page [36](#) for details.

How to enable debug console repeater mode?

1. Use the host debug console to Telnet to client.
2. Create pipe: `$mknod p1 p && mknod p2 p`
3. Enable bi-directional repeater mode by:

Host:

 - Start read: `$cat p1 &`
 - Start write: `$cat > p2`

Client:

 - Start read: `$cat p2 &`
 - Start write: `$cat > p1`

[A1.2 Firmware] How to have functions like Serial over IP without going through the host board's UART port 1? For example, controlling all host/client boards through a network attached PC.

We call it "Serial over IP guest mode".

Please request sample code if you need this feature.

Console APIs

Debug Console APIs

- Turn on/off application debug message
 - NAME: setconsole
 - EXAMPLE:
 - Turn off: `$setconsole /dev/ttyS1`
 - Turn on: `$setconsole -r`

- Turn on/off kernel debug message
 - NAME: dmesg
 - EXAMPLE:
 - Turn off: `$dmesg -n 1`
 - Turn on: `$dmesg -n 8`

- Reboot system
 - NAME: reboot
 - EXAMPLE: `$reboot`

- Telnet client
 - NAME: telnet
 - EXAMPLE:
 - telnet to IP 169.254.0.101 port 24
 - `$telnet 169.254.0.101 24`
 - telnet to ast-client0000.local port 24
 - `$telnet ast-client0000.local 24`

- List all KDS-6 devices:
 - NAME: node_list
 - OPTIONS
 - t : specify the device type to list
 - EXAMPLE
 - List all clients:
 - `$/node_list -t client`
 - List all hosts:
 - `$/node_list -t host`

- Get IP from hostname:
 - NAME: astresname
 - EXAMPLE:
 - Get "ast-gateway0001.local"'s IP address:

- `$/astresname ast-gateway0001.local`
- To kill a running process by process name:
 - NAME: pkill
 - EXAMPLE:
 - kill a process named "abc"
 - `pkill -9 abc`

Link Manager APIs

- Send event to link manager
 - NAME: `ast_send_event`
 - OPTIONS:
 - `e_stop_link`: stop link
 - `e_reconnect`: restart link
 - `e_chg_hostname`: notify hostname change
 - `e_button_link`: trigger a "button 1" press event
 - `e_button_link_1`: trigger a "button 1" long press event
 - `e_button_pairing`: trigger a "button 2" press event
 - `e_button_pairing_1`: trigger a "button 2" long press event
 - `e_debug`: save current link manager states into `./lm_params`
 - EXAMPLE:
 - `$/ast_send_event -1 e_chg_hostname`
- Get current link manager state
 - COMMAND: `$cat /var/ast_device_status`
 - RESULT: (Host)
 - `s_idle`: Client connection is not allowed. (loopback mode)
 - `s_attaching`: Client is connecting or video source is not available.
 - `s_srv_on`: All functions are ready to work or working.
 - other states: non-stable states
 - RESULT: (CLIENT)
 - `s_idle`: Client is not connected to Host
 - `s_srv_on`: Client is connected to Host
 - other states: non-stable states

Video Over IP Driver Statistic/Control APIs

The interface is presented as files and located under `/sys/devices/platform/videoip/`.

- Connected client list:
 - FILE: `unlink`
 - USAGE: `cat /sys/devices/platform/videoip/unlink`

Display Driver Statistic/Control APIs

The interface is presented as files and located under `/sys/devices/platform/display/`.

- Attached monitor information:
 - FILE: `monitor_info`
 - USAGE: `cat /sys/devices/platform/display/monitor_info`
- Video timing information:
 - FILE: `timing_info`
 - USAGE: `cat /sys/devices/platform/display/timing_info`
- [\geq A3.54] Turn on/off display:
 - FILE: `screen_off`
 - USAGE:
 - Turn off screen: `echo 1 > /sys/devices/platform/display/screen_off`
 - Turn on screen: `echo 0 > /sys/devices/platform/display/screen_off`
 - Show status: `cat > /sys/devices/platform/display/screen_off`
 - enable: means screen is off
 - disable: means screen is on

OSD APIs

By using the following script, the user can dynamically show OSD messages.

- Turn ON:
 - `./osd_on.sh`
 - USAGE:
 - To show "This is OSD string":
 - `./osd_on.sh "This is OSD string"`
 - The OSD string takes 'printf' like special character input, for example:
 - `./osd_on.sh "Line 1 \n Line 2"`
 - Use following code for reserved characters:
 - `\x3A: ':'`
 - `\n: CR`
 - `\t: tab`
 - Environment Variables: Please see the comments in `osd_on.sh`
 - `OSD_STR`

- OSD_FONT_SIZE
 - OSD_FONT_COLOR
 - OSD_TRANSPARENT
 - OSD_MAX_Y_SIZE
- Turn OFF:
 - `./osd_off.sh`
 - USAGE: To disable OSD message after 5 seconds:
 - `./osd_off.sh 5 &`

Note: The maximum number of characters an OSD string can show is software limited to 128 characters.

Configurations APIs

Current firmware reserved two flash sections for system configuration. One is read only (RO) the other is read-write (RW). By factory default, there are configurations saved in the flash read only (RO) section. When the firmware starts up, both RO and RW configurations are read from flash and saved in RAM. All configuration modifications are saved in RAM until `"/astparam save"` is called.

- Access configurations from RO/RW section:
 - NAME: `astparam`
 - OPTIONS
 - `r` : read from RO section
 - `g` : read from RW section
 - `s` : write to RW section
 - `flush` : clear all settings in RW section including randomly generated MAC address. `"/astparam save"` is needed to clear all settings in flash memory.
 - `dump`: dump all parameters in RW section
 - `dump ro`: dump all parameters in RO section
 - `dd`: dump all parameters in RW section directly from flash
 - `dd ro`: dump all parameters in RO section directly from flash
 - `save`: save all parameters in RW section into flash
 - EXAMPLE:
 - read "ip_mode" setting from RO section:
 - `"/astparam r ip_mode`
 - read "ip_mode" setting from RW section:
 - `"/astparam g ip_mode`
 - write "ip_mode" setting as "autoip" to RW section:
 - `"/astparam s ip_mode autoip`
 - clear all settings in RW section:
 - `"/astparam flush`
 - clear ip_mode setting in RW section:
 - `"/astparam s ip_mode`
- Reset setting to factory default:
 - NAME: `reset_to_default.sh`
 - DESCRIPTION:
 - reset RW section but keeps the random generated MAC address. A system reboot is required to take effect.
 - EXAMPLE:
 - `"/reset_to_default.sh`

IIC APIs

- [\geq A3.0] Initial IIC bus:
 - NAME: bus_init
 - DESCRIPTION: Used to initial IIC bus speed. Must be set at least once in order to work.
 - EXAMPLE: Set bus number 1 using 40000Hz
 - `$echo 1 40000 > /sys/devices/platform/i2c/bus_init`
- Select IIC bus and device:
 - NAME: io_select
 - DESCRIPTION: Used to set IIC slave device's bus number and device address
 - EXAMPLE: choose bus number 1 and device address 9a
 - `$echo 1 9a > /sys/devices/platform/i2c/io_select`
- Read/Write IIC slave:
 - NAME: io_value
 - DESCRIPTION: Used to read or write value. Set "i_range" for read range.
 - EXAMPLE:
 - To write 0xaa to IIC device offset 0xbb:
 - `$echo 0xbb 0xaa > /sys/devices/platform/i2c/io_value`
 - To read 5 bytes from offset 0xbb:
 - `$echo 0xbb 0x5 > /sys/devices/platform/i2c/i_range`
 - `$cat /sys/devices/platform/i2c/io_value`
- Setting range of IIC read:
 - NAME: i_range
 - DESCRIPTION: Used to set the range for read. See "io_value" for example.

UART Port 1 APIs

- Setup /dev/ttyS0:
 - NAME: stty
 - DESCRIPTION: http://linux.about.com/od/lna_guide/a/gdelna38t01.htm
 - Example: set baudrate to 115200
 - `$stty 115200 -F /dev/ttyS0`
-

Configuration Option List

Configuration options are settings saved in flash. These settings are used by KDS-6 FW. The KDS-6 FW reserves two flash partitions for this purpose, one is read only (RO) and the other is read-write (RW). Factory default settings are saved in the RO partition. Changes are saved in the RW partition. When the same configuration name is defined in both partitions, the setting in the RW partition is used (unless specially noted).

- ch_select
 - DESCRIPTION: Used by client FW only. If defined, client FW uses it to resolve target host's IP address, otherwise 4-bits DIP switch will be used as ch_select.
 - OPTIONS: IP or host's hostname_id
 - DEFAULT: 4-bits DIP switch
 - PARTITION: [A1.32]RO, RW
- hostname_id
 - DESCRIPTION: If defined, FW will use hostname_id to create the board's hostname (See FAQ: [What is the relationship between hostname and hostname_id?](#)). If not defined, default value will be used.
 - OPTIONS: any string without '.'
 - DEFAULT:
 - When hostnamebydipswitch == y: 4-bits DIP switch
 - When hostnamebydipswitch == n: mac address
 - PARTITION: [A1.32]RO, RW
- ip_mode
 - DESCRIPTION: Used to configure how FW allocates it's IP address.
 - OPTIONS:
 - autoip: Auto-generated. Use 169.254.xxx.xxx private IP domain.
 - dhcp: Use DHCP client. A DHCP server is required.
 - static: Use static assigned IP address. The flash configurations "ipaddr" and "netmask" are used.
 - DEFAULT: autoip
 - PARTITION: RO, RW
- ipaddr
 - DESCRIPTION: Used when "ip_mode" is "static". If not defined, default value is used.
 - DEFAULT:
 - Host: 169.254.0.222
 - Client: 169.254.0.111
 - PARTITION: RO, RW
- netmask
 - DESCRIPTION: Used when "ip_mode" is "static". If not defined, default value is used.

- DEFAULT: 255.255.0.0
- PARTITION: RO, RW

- gatewayip
 - DESCRIPTION: Used when "ip_mode" is "static". If not defined, default value is used.
 - DEFAULT: 169.254.0.254
 - PARTITION: RO, RW

- s0_baudrate
 - DESCRIPTION: The static baud rate used under RS-232 over IP Type 2 mode. For example: "115200-8n1" means using "115200" baud rate with data bits "8", parity "None" and stop bits "1".
 - DEFAULT: 115200-8n1
 - PARTITION: RO, RW

- multicast_ip
 - DESCRIPTION: Multicast group used under multicast mode. Always map the multicast_ip to 225.0.B0.B1B2B3 format where B0, B1, B2 and B3 are originally mapped from 4-bits DIP switch. Do not use a different multicast_ip, unless you know what you are doing.
[A3.50] Starting from firmware A3.50, the default value of multicast IP (multicast_ip) changes from 225.0.x.xxx to 225.0.10x.xxx. Please use the new multicast_ip value when using fw version A3.50 and up.
 - OPTIONS: any valid multicast IP
 - DEFAULT: mapping from 4-bits DIP switch:
 - [<A3.50] 225.0.B0.B1B2B3
 - [≥A3.50] 225.0.10B0.B1B2B3
 - PARTITION: [A1.32]RO, RW

- hostnamebydipswitch
 - DESCRIPTION: If "hostname_id" is not defined, the KDS-6 FW selects the "hostname_id" according to the definition of "hostnamebydipswitch". If 'y' the value of the 4-bits DIP switch is used. Otherwise, the MAC address is used.
 - OPTIONS: y , n
 - DEFAULT:
 - Host: y
 - Client: n
 - Client for USB only FW: y
 - PARTITION: RO, RW

- reset_ch_on_boot
 - DESCRIPTION: Set to 'y' to reset channel related settings to the setting of the 4-bits DIP switch after board reset. By using this feature, the modification of multicast group or unicast target disappears after board reset. The following settings are reset:

- multicast_ip
 - ch_select
 - hostname_id
- OPTIONS: y , n
- DEFAULT: y
- PARTITION: RO, RW
- ast_video_quality_mode
 - DESCRIPTION: Used to configure "Graphic Mode" or "Video Mode". This setting only takes effect each time video capturing starts. This setting only applies to the host board. Pressing "button 2" changes this setting. The recommended way of dynamically changing the quality mode is to use "button 2 event".
 - OPTIONS:
 - 0: graphic mode
 - -1: video mode
 - DEFAULT: -1
 - PARTITION: RO, RW
- v_bcd_threshold
 - DESCRIPTION: Used to configure "Anti-Dithering Mode". This setting only takes effect each time video capturing starts. This setting only applies on the host board. Pressing and holding "button 2" changes this setting. The recommended way of dynamically changing this setting is to use "button 2 event".
 - OPTIONS:
 - 0: Off
 - 1: Anti-dithering 1
 - 2: Anti-dithering 2
 - DEFAULT: 0
 - PARTITION: RO, RW
- no_soip
 - DESCRIPTION: Used to turn ON/OFF "Serial Over IP" feature. This setting only takes effect on system reboot.
 - OPTIONS:
 - y: Disable "Serial Over IP" feature
 - n: Enable "Serial Over IP" feature
 - DEFAULT: Default value is configured by manufacturer using "autoterm".
 - PARTITION: RO, RW
- soip_type2
 - DESCRIPTION: Used to configure "Serial Over IP" operating mode when "Serial Over IP" feature is enabled. This setting only takes effect on system reboot and "Serial Over IP" feature is enabled.
 - OPTIONS:
 - n: Use "Serial over IP type 1"
 - y: Use "Serial over IP type 2"
 - DEFAULT: Default value is configured by manufacturer using "autoterm"

- PARTITION: RO, RW
- [A1.2 Firmware] soip_guest_on
 - DESCRIPTION: Used to configure "Serial Over IP" operating mode to "guest mode". This setting only takes effect on system reboot and "Serial Over IP" feature is enabled.
 - OPTIONS:
 - n: Use typical "Serial Over IP" mode, in which host and client are paired together.
 - y: Use "guest mode", in which both host and client boards' UART port 1 are controlled by an external network controller (for example a PC).
 - DEFAULT: n. Can be configured by manufacturer using "autoterm"
 - PARTITION: RO, RW
- multicast_on
 - DESCRIPTION: Used to configure unicast/multicast operation mode. Since the operation mode of "USB over IP" will be different between multicast and unicast mode. So, when you change 'multicast_on', remember to config 'USB over IP' mode as well. See 'share_usb' and 'share_usb_auto_mode' below.
 - OPTIONS:
 - n: unicast mode
 - y: multicast mode
 - DEFAULT: n
 - PARTITION: RO, RW
- share_usb
 - DESCRIPTION: Used to configure USB over IP's operation mode. There are two USB over IP operation modes. 'share USB' mode is typically used in multicast mode. In this mode, only one client can request the USB over IP access at a time.
 - OPTIONS:
 - n: default mode for unicast mode
 - y: share USB mode. default mode for multicast mode
 - DEFAULT:
 - n
 - PARTITION: RO, RW
- [≥A3.54] share_usb_auto_mode
 - DESCRIPTION: A new setting used to overwrite the setting of 'share_usb' depending on the setting of 'multicast_on'.
 - OPTIONS:
 - y: enable. The value of 'share_usb' is always:
 - share_usb will be 'n' under unicast mode

- share_usb will be 'y' under multicast mode
- n: disable.
- DEFAULT:
 - n: For backward compatibility. The default value is 'n'.
- PARTITION: RO, RW



P/N: 2900-300839



Rev: 1



SAFETY WARNING

Disconnect the unit from the power supply before opening and servicing

For the latest information on our products and a list of Kramer distributors, visit our Web site where updates to this user manual may be found.

We welcome your questions, comments, and feedback.