

Table of Contents

[Table of Contents](#)

[Revision History](#)

[Overview](#)

[Supported Platform](#)

[Supported FW Version](#)

[Terminology](#)

[Preface](#)

[How to Access the Console](#)

[How to get AST15XX board's IP address](#)

[Command Category](#)

[Command Syntax Conventions](#)

[Console APIs](#)

[Link Manager Control](#)

[Access Settings in Flash - astparam](#)

[Send Event to Link Manager](#)

[Get/Set Link Manager Internal Parameters](#)

[Get Link Manager State](#)

[Simulate Button Press](#)

[Link Management](#)

[Switch Channel \[Client Only\]](#)

[Switch Channel \[Host Only\]](#)

[Stop Link \[Client Only\]](#)

[Stop Link \[Host Only\]](#)

[Get Current Channel Setting](#)

[IP Network Commands](#)

[Change Host Name ID](#)

[Get Host Name](#)

Get IP Address

Get MAC Address

Get Ethernet Link Status and Mode

Node Discovery and Query

Node Discovery and Query

Video Commands

Configure Graphic/Video Mode [Host Only]

Set Anti-Dither Mode [Host Only]

Enable/Disable Loopback Port [Host Only]

Frame Rate Control [Host Only]

Bit Rate Control [Host Only]

Detach/Attach Video Input Port [Host Only]

Upload/Copy Video Downstream Sink's EDID

Get Current EDID Reported to Upstream [Host Only]

Get Video Source Timing Information

Get Video Engine State

Get Video Sink Information

Turn Off/On Video Output [Client Only]

Pause Video Stream [Client Only]

Select Video Input Port [Host Only]

Scale Video Output [Client Only]

Video Snapshot

Video Wall Commands

Show Picture When Link Down

OSD Commands

Show String OSD [Client Only]

Show Picture OSD [Client Only]

Audio over IP Commands

Audio Source Port Selection [Host Only]

Analog Audio Volume Control

USB/IP/KMoIP Commands

Manually Start/Stop USB/IP and KMoIP [Client Only]

Manually Start/Stop USB/IP [Client Only]

Manually Start/Stop KMoIP [Client Only]

Get USB/IP Status [Client Only]

Get KMoIP Status [Client Only]

Get Attached Client IP [Host Only]

USB/IP Device Exporting Policy [Client Only]

I2C Access

Initial I2C Bus

Select I2C Bus and Device

Set Range for I2C Read

Read/Write I2C Slave

Misc.

Reboot Device

Turn on/off debug console's debug message

Migrate From Legacy Console APIs

Ethernet Switch Configuration

Incompatible

Command Changes

astparam Changes

Appendix A. Channel to Multicast IP Map

Appendix B. IP Port Usage List

Appendix C. Link Manager Event List

Host

Link Control

Button Events

Client

Link Control

Button Events

Product: AST15XX
Date: 9/12/2017
FW Rev:

Doc Type:

OSD Control

Appendix D. List of Imparam

Common

Host

Client

Appendix E. Sample Code

Revision History

- 2017/09/12:
 - [Switch Channel (Client Only)] Update Ex7.
- 2017/08/01:
 - [Analog Audio Volume Control] Add "Analog Audio Volume Control" information.
- 2017/07/24:
 - [Switch Channel (Client Only)] Add more 'CH_SELECT' details.
- 2017/07/18:
 - [Manually Start/Stop USB0IP and KMoIP] Add Ex3.
- 2017/07/14:
 - [Node Discovery and Query] Add new '--intf' option for 'node_query' command.
- 2017/07/12:
 - Revised "Migrate From Legacy Console APIs::Ethernet Switch Configuration".
 - Revised "Overview::Supported Platform".
- 2017/07/07:
 - First version.

Overview

The "Console APIs" interface is so called "Command Line Interface" (CLI). This document describes the commands which can be used under AST15XX series' platform's console interface. AST15XX uses Linux OS and the console is derived by Bash shell. Console can be accessed through:

- Debug console: is set to UART port 2 using baud rate at 115200-8n1.
- Telnet port 24: There is built-in telnet server to connect to AST15XX devices.
- Web CGI: Controlled through web protocol. Mapping CGI command to console command.

Through console APIs, developers can get more control on AST15XX firmware and extend product's features and capability.

Supported Platform

- AST1520
- AST1525
- AST1510: Only supports commands introduced before A7.0.0. FW \geq A7.0.0 doesn't support AST1510 platform.
- AST1500: Only supports commands introduced before A7.0.0. FW \geq A7.0.0 doesn't support AST1500 platform.

Supported FW Version

Supported FW version is listed in each command. This document is based on A7.0.0.

Terminology

Term.	Description
AST15XX	Refers to all series of AST1500, AST1510, AST1520.. platform.
AST152X	Refers to AST1520 and AST1525.
host	The AST15XX video encoder board.
Tx	Same as 'host'.
client	The AST15XX video decoder board.
Rx	Same as 'client'.
node	Generic term for 'host' and 'client'.
device	Same as 'node'.
FW	Short of 'firmware'. Normally refers to AST15XX's firmware.
HW	Short of 'hardware'. Normally refers to AST15XX's HW platform.
console	FW's command line interface (CLI).
console API	FW provides commands through console as API.
Link Manager	A major FW script/program handling most of user space link management.
astparam	A command to access parameters in flash. Also refer to those parameters accessed through this command.
Imparam	A command to access internal parameters of 'Link Manager'. Also refer to those parameters accessed through this command.
event	Normally refers to control commands sending to 'Link manager'.
service	AST15XX provide many over IP extension functions. It is called 'service' in general. For example, video over ip, usb over IP....
channel	Most of over IP streaming 'services' using different multicast IP/group. To simplify the process of choosing multicast IP, we use 'channel' to represent/mapping each service's multicast IP/group. Normally, each host has its own unique channel setting. So that client can choose which host to connect to by specifying the host's 'channel' number.
switch channel	The process of changing 'channel' number.
sink	Video sink. Refer to TV, monitor which attached to AST15XX's client video output port or AST15XX host's video loopback output port.

Product: AST15XX
Date: 9/12/2017
FW Rev:

Doc Type:

source	Video source. Refer to video player whichever feed HDMI/DVI video signal to AST15XX host.
OSD	Stands for "On Screen Display". Client display HW engine 'overlay' function to overlay 16bits RGB565 data onto video decode picture. FW provide 'String OSD' and 'Picture OSD' APIs for overlay string or picture onto video decode picture.
USBoIP	Short of 'USB over IP' extension function. Specially refers to 'transparent' type of 'USB over IP' extension function. Comparing to 'KMoIP' which is 'emulation' type.
KMoIP	Short of 'Keyboard/Mouse over IP' extension function. It is a kind of USB extension, but uses 'emulation' technology instead of 'transparent' USBoIP.

Preface

How to Access the Console

There are 3 ways to access the console:

1. **UART2 debug port:** It is a RS232 interface using 115200-8n1 baudrate setting. Developer can attach to this debug port to access console.
2. **Use telnet:** FW has built-in telnetd server and telnet client built in. Developer can use any telnet client to connect to any AST15XX board. Or write their own program using telnet protocol to connect to any AST15XX board through IP network. NOTE: **Default use telnet port 24 instead of 23. Use "root" to login. No password is required for login.**
3. **Use Web CGI:** FW has built-in web server and has a CGI command to mapping console commands to CGI. Check the Web UI ⇒ System ⇒ Utilities ⇒ Console API Command.

Use telnet or web protocol are recommended way for developing control software.

UART2 debug port is usually used for engineering test and debug. By default, debug port will output a lot of debug messages. You can [turn it off](#).

How to get AST15XX board's IP address

FW default runs "autoip" and uses 169.254.xxx.xxx private IP domain. The IP of target AST15XX board is resolved by its hostname using mDNS protocol. If developers want to use "static ip" or "DHCP client", developer will be responsible to maintain the "IP to board" mapping by themselves. Here are the methods to resolve the IP address using "autoip" mode.

- **Client's GUI:** The client's GUI will display the client's IP address and connected host's IP address.
- **Console APIs:** Following console APIs are provided:
 - [FW >= A7.0.0] **node_query**: An enhanced version of node_list. See "[Node Discovery and Query](#)" section for details.
 - [FW < A7.0.0] **node_list**: use this command to list all AST15XX boards' hostname and IP address.
 - **astresname**: use this command to resolve a target's IP address by providing target's hostname.
- **From PC:** Following methods are provided:
 - For end users, please reference to "Documents/Web UI/How to Connect to Web UI" document.
 - Developer can use/implement ASPEED proprietary discovery protocol. Reference to "**node_query**" or "node_list" sample code. There is pre-compiled execute files for Windows OS under "Documents/Console APIs/SampleCode/".
 - Use the mDNS libraries. For example:
 - Bonjour: <http://developer.apple.com/networking/bonjour/>
 - pyZeroConfig: <http://sourceforge.net/projects/pyzeroconf/files/>

Command Category

Different usage scenario is achieved by different type of commands. "[Console APIs](#)" section describes all those commands by usage type. However, by implementation type, Commands can be classified into following types:

- Link manager control:
 - 'e' command. Send event to link manager
 - 'Imparam' command. Query link manager parameters.
- Flash access control:
 - 'astparam' command. Used to read/write parameters into flash. See "Documents/Console APIs/All about astparam" for details.
- Driver control:
 - 'sysfs' driver files. Drivers use Linux 'sysfs' filesystem to export proprietary control interface to user.
- Bash shell commands:
 - echo, cat.... Normally used to access 'sysfs'.

Command Syntax Conventions

Following table describes the syntax used with the commands in this document.

Convention	Description
boldface	Commands and keywords.
<i>italic</i>	Command input that is supplied by you.
[x]	Keywords or parameters that appear within square brackets are optional.
{x}	Keywords or parameters within braces must be entered.
x y	Keywords or parameters separated by a vertical bar require you to choose one option.
x y	Keywords or parameters separated by a double vertical bar allow you to choose any or all of the options.

Note:

- All commands and parameters are case sensitive.
- **/ #** in command examples means command prompt. It is not part of command.

Console APIs

Link Manager Control

After FW boots up, the process hands over to 'link manager' script. Link manager is the main process controlling the reset of FW's life cycle. It maintains connection state machine. Responsible for taking and response events from low level drivers and users. To communicate with 'link manager', user send event to it by a command named 'e'.

Product: AST15XX
Date: 9/12/2017
FW Rev:

Doc Type:

Access Settings in Flash - astparam

Please read "Documents/Console APIs/All about astparam" for details.

Send Event to Link Manager

This is the most used command to control the system. This command has two syntax format. 'e' is just a shortcut of 'ast_send_event -1'. They are basically the same. The detail usage of events will be described in the reset of this document. And a list of available events are listed in **Appendix C**.

Syntax

e {event[::arg1][::arg2]...}

ast_send_event -1 {event[::arg1][::arg2]...}

Parameters

- event: A case sensitive string to represent a event. Usually starts with 'e_'.
 - Ex: e e_reconnect
- arg: If an event need some arguments, it is usually attached after event string with '::' token. Some events uses '_' to separate arguments.
 - Ex: e e_reconnect::9999::v

Usage Information

- See **Appendix C** for the list of available events.

Version History

- FW >= A5.0.0

Get/Set Link Manager Internal Parameters

Imparam is usually used to query some of link manager's internal parameters. This command also provide a method to set the value of specified parameter. However, it is rarely useful and may confuse link manager script results in unexpected problem.

Appendix D has a list of useful Imparam.

Syntax

Imparam {g|s} {key [value]}

Imparam dump

Parameters

- g: Get value of specified key. When 'g', no [value] is needed.
- s: Set specified key to value. [value] is only required when 's'.
- key: **Appendix D** has a list of useful Imparam keys.
- dump: Dump all Imparam key/value.

Usage Information

- Don't use 'set' command unless you know what you are doing.
- In FW >= A7.0.0, Link manager is divided into sub-link managers per services. Like "video link manager (vLM)". And they also has similar commands like Imparam. For example, vLM has '**vlmparam**' command. uLM has '**ulmparam**' command.
- Link manager internal parameters is implementation dependent. Which means their define may changes when FW changes. Please avoid using it, especially parameters that are not mentioned in this document.

Version History

- FW >= A6.0.0

Examples

Ex1: Get 'STATE'

```
/ # lmparam g STATE  
  
s_srv_on/ #
```

Ex2: Set 'TEST' to value 'on'

```
/ # lmparam s TEST on  
/ #
```

Ex3: Dump all Imparam key/value

Product: AST15XX
Date: 9/12/2017
FW Rev:

Doc Type:

```
/ # lmparam dump  
IS_HOST=y  
STATE=s_srv_on  
ACCESS_ON=y  
DBG=0  
SHARE_USB=y  
SHARE_USB_AUTO_MODE=n  
SHARE_USB_ON_FIRST_PEER=y  
CH_SELECT=9999  
.....  
REMOTE_EDID_PATCH=00000001  
LOOPBACK_EDID_PATCH=00000000  
LOOPBACK_DEFAULT_ON=y  
V_RX_DRV=it6802  
V_FRAME_RATE=0  
/ #
```

Get Link Manager State

Syntax

Imparm g STATE

Result

- Host
 - s_init: System is under initialization.
 - s_idle: System is idle. Services are stopped.
 - s_attaching: Services are started, but no client attached or no video source.
 - s_srv_on: Services are started.
 - s_error: Fatal error.
- Client
 - s_init: System is under initialization.
 - s_idle: System is idle. Services are stopped.
 - s_srv_on: Services are started.
 - s_error: fatal error.

Example

Ex1: Get link manager current state.

```
/# lmparm g STATE  
  
s_srv_on  
/#
```

Simulate Button Press

Evaluation board has built-in two button. FW defined a set of button behavior. User can use following 'e' commands to trigger those button behavior with the need to actually press the button. Button's behavior can be customized. For details, please read document "**LED and Button/AST1500 Button Function Descriptions**".

Syntax

e {*button_event*}

Parameters

- button_event: See **Appendix C** for available button events.

Usage Information

- See document "LED and Button/AST1500 Button Function Descriptions" for details.

Version History

- FW >= A5.0.0

Examples

Ex1: Simulate button 1 short press.

```
/ # e e_button_link  
/ #
```

Product: AST15XX
Date: 9/12/2017
FW Rev:

Doc Type:

Link Management

Switch Channel [Client Only]

Command client to connect to a specified channel. For API backward compatibility and easier usage, this command has 3 syntax format. Channel can be specified by channel number (*ch-select*) or host's ip address (*host-ip-addr*).

Syntax [legacy, refresh current connection]

e e_reconnect

Syntax [specify channel, apply to all services]

e e_reconnect::{*ch-select*|*host-ip-addr*}

Syntax [specify both channel and service(s)]

e e_reconnect::{*ch-select*|*host-ip-addr*}::**{z|Z}|{v|u|a|r|s|p}**}

Parameters

- *ch-select*:
 - Enter 4 digit decimal number from 0000 to 9999. It must be in 4 digit format prepending with 0.
 - Example: 0000, 0001, 0100, 1000, 3456, 1342, 8888, 9999....
 - Default value: Got from on-board 4-bits DIP switch. Since it is 4-bits 0 or 1, it can only represent 16 channels in total. They are 0000, 0001, 0010, 0011, 0100, 0101,..., 1111. Default values are also interpreted as decimal number instead of binary value.
- *host-ip-addr*:
 - Enter an IP address in dotted decimal format. Used to specify the host's IP address to connect to.
 - Example: 169.254.11.22
- **{z|Z}|{v|u|a|r|s|p}**:
 - Used to specify which function to apply.
 - v: video over IP
 - u: USB over IP
 - a: Audio over IP
 - r: IR over IP
 - s: Serial over IP
 - p: GPIO over IP
 - z: all features. == **vuasrp**
 - Z: all features except video over IP. == **uasrp**
 - {z|Z} is **exclusive** from {v|u|a|r|s|p}
 - {v|u|a|r|s|p} **can be any combination and order** which indicate connection order of specified service.
 - For example: "uar" means first "USB over IP", then "Audio over IP" and "IR over IP" at last.

Usage Information

- By specifying different channel to different services, you can 'free route' each services to different host as you want.
- Command supports *ch-select* from 0000 to 9999, which means there are maximum 10000 host channels can be used.
- For supporting 'free routing', each service has its own multicast IP and port number. So, there are multiple multicast IP setting on a host machine, not just one multicast IP. To simplify configuration process, FW will automatically map 'channel' to 'multicast IP'. User don't have to specify 'multicast IP'. See **Appendix A** for 'channel to multicast IP' mapping.
- Instead of configure 'ch_select' astparam in FW < A7.0.0, FW >= A7.0.0 will automatically update corresponding 'ch_select_x' astparam on receiving 'e_reconnect' command with {ch-select|host-ip-addr} specified. **Manually change astparam 'ch_select_x' is NOT recommended.**
- To persist channel setting through system reboot, please set astparam, *reset_ch_on_boot*, to 'n'.
- FW will NOT save start/stop state into flash (*access_on_x*). Set astparam, *access_on_x*, to manually set the state if you need.
- To configure your own 'channel to multicast IP' mapping, you can change astparam, *multicast_ip_prefix*. If changing prefix is not enough, you need SDK to modify FW to do your own mapping.
- [FW >= A7.0.0] In FW < A7.0.0, client maps *ch_select* to host's hostname in order to find and connect to host. In FW >= A7.0.0, use new approach to discover host's IP address and host's hostname is no more bond to channel resolution process. Which means host's astparam, *hostname_id*, can be any value as user's wish in FW >= A7.0.0. However, in order to backward compatible with old FW behavior, new FW still link host's 'hostname_id' and 'ch_select' together. To break this link, please set host's astparam, *hostnamebydipswitch*, to 'n'.
- For a trick to improve video switching time, please see **Ex7** in following Examples section.
- Legacy "client channel switching" command is still working. BUT following new astparam must be removed to avoid conflict. *ch_select_v*, *ch_select_u*, *ch_select_a*, *ch_select_r*, *ch_select_s*, *ch_select_p*.
 - Which means you can't mix legacy and new channel switching commands together in one system. Otherwise new command's setting will overwrite legacy setting.

Related astparam

- *ch_select*: default value of all following 'ch_select_x'. If not defined, *ch_select*'s default value is got from on-board 4-bits DIP switch.
- FW will automatically maintain following 'ch_select_x' astparam:
 - *ch_select_v*: {ch-select|host-ip-addr} value for video over IP.
 - *ch_select_u*: {ch-select|host-ip-addr} value for USB over IP.
 - *ch_select_a*: {ch-select|host-ip-addr} value for Audio over IP.
 - *ch_select_r*: {ch-select|host-ip-addr} value for IR over IP.
 - *ch_select_s*: {ch-select|host-ip-addr} value for Serial over IP.
 - *ch_select_p*: {ch-select|host-ip-addr} value for GPIO over IP.
 - *ch_select_i2s*: default value of 'ch_select_a'. Exists for legacy reason.

Don't use it.

- ch_select_soip2: default value of 'ch_select_s'. Exists for legacy reason. Don't use it.

- NOTE: **modify above 'ch_select_x's is not allowed and will cause un-expected FW behavior.**

- reset_ch_on_boot: To persist channel setting through system reboot, please set astparam, reset_ch_on_boot, to 'n'.
- multicast_ip_prefix:
- hostnamebydipswitch: y|n. Host set to 'n' if you don't want to link host's 'hostname_id' with host's 'ch_select'.

Related Imparam

- CH_SELECT: From astparam, ch_select. Default value of following CH_SELECT_X. **NOT** current configured channel number. This is different from host.
- CH_SELECT_V, CH_SELECT_U, CH_SELECT_A, CH_SELECT_R, CH_SELECT_S, CH_SELECT_P
- RESET_CH_ON_BOOT
- ACCESS_ON: From astparam, astaccess.
- ACCESS_ON_V, ACCESS_ON_U, ACCESS_ON_A, ACCESS_ON_R, ACCESS_ON_S, ACCESS_ON_P

Version History

- FW >= A7.0.0

Examples

Ex1: Start connect all services to 'current' channel. 'current' channel is by default got from on-board 4-bits DIP switch. **Only this command will trigger FW to re-read 4-bits DIP switch value.**

```
/ # e e_reconnect  
/ #
```

Ex2: Start connect all services to channel '9810'.

```
/ # e e_reconnect::9810  
/ #
```

Ex3: Above command equals to following command:

```
/ # e e_reconnect::9810::z  
/ #
```

Ex4: Start connect 'video over IP' to channel '9810'. The others will use 'current' channel.

```
/ # e e_reconnect::9810::v  
/ #
```

Ex5: Start connect 'video over IP' and 'audio over IP' to channel '1234'. Then connect 'usb over IP' to channel '8888'.

```
/ # e e_reconnect::1234::va  
/ # e e_reconnect::8888::u  
/ #
```

Ex6: Start connect 'video over IP' to channel '1222'. Then connection others to '1222'

```
/ # e e_reconnect::1222::v  
/ # e e_reconnect::1222::z  
/ #
```

Ex7: [FW < A7.1.0] Connect to channel '1222' fast video switching in one command. 'Video over IP' switch first. By 'sleep 2', this command let CPU focus on video switching first, then the others. So that we get faster video switching time comparing to 'e e_reconnect::1222'. **(video switching takes about only 1 seconds using this trick.)**

```
/ # e e_reconnect::1222::v;sleep 2;e e_reconnect::1222::z  
/ #
```

[FW >= A7.1.0] Implement and enhanced this trick into FW default behavior. Just use following command (without specifying 'functions') can switch video under only 1 second. You can still use above trick for free routing commands. Sleep 1 second will be enough.

```
/ # e e_reconnect::1222  
/ #
```

Ex8: Connect 'USB over IP' to '169.254.2.3'

```
/ # e e_reconnect::169.254.2.3::u  
/ #
```


Switch Channel [Host Only]

Host command. Used to setup host's multicast IP per channel setting. Issue this command also starts/re-starts host services. This command has 2 syntax format.

Syntax

e e_reconnect

e e_reconnect::{*ch-select*}

Parameters

- *ch-select*:
 - Enter 4 digit decimal number from 0000 to 9999. It must be in 4 digit format prepending with 0.
 - Example: 0000, 0001, 0100, 1000, 3456, 1342, 8888, 9999....
 - Default value: Got from on-board 4-bits DIP switch. Since it is 4-bits 0 or 1, default value can only represent 16 channels in total. They are 0000, 0001, 0010, 0011, 0100, 0101,..., 1111. Default values are also interpreted as decimal number instead of binary value.
 - **CAN NOT accept IP address in dotted decimal format.** Only 4 digit decimal number is valid.

Usage Information

- Although client also has 'e_reconnect' command, syntax and usage is a little bit different.
- Command supports *ch-select* from 0000 to 9999, which means there are maximum 10000 host channels can be used.
- Unlike client, **all services uses the same ch-select value.**
- To persist channel setting through system reboot, please set *astparam, reset_ch_on_boot*, to 'n'.
- For supporting 'free routing', each service has its own multicast IP and port number. So, there are multiple multicast IP setting on a host machine, not just one multicast IP. To simplify configuration process, FW will automatically map 'channel' to 'multicast IP'. User don't have to specify 'multicast IP'. See **Appendix A** for 'channel to multicast IP' mapping.
- To configure your own 'channel to multicast IP' mapping, you can change *astparam, multicast_ip_prefix*. If changing prefix is not enough, you need SDK to modify FW to do your own mapping.
- [FW >= A7.0.0] In FW < A7.0.0, client maps *ch_select* to host's hostname in order to find and connect to host. In FW >= A7.0.0, use new approach to discover host's IP address and host's hostname is no more bonded to channel resolution process. Which means host's *astparam, hostname_id*, can be any value as user's wish in FW >= A7.0.0. However, FW by default will still update hostname based on *ch_select*.
- [Host] For backward compatible with legacy "host channel switching" APIs,
 - host's **hostname_id MUST be the same as host's channel setting**,

ch_select.

■ **Must be the same format as 'ch_select', 0000~9999.** FW will use '0000' default value if invalid format detected.

- host's 'e_chg_hostname' event will automatically trigger channel setting change, 'e_reconnect::{ch_select}'.
- 'e_reconnect::{ch_select}' will automatically change 'hostname_id' to specified 'ch_select' value.
- If you want to break this legacy behavior and use different 'hostname_id' from 'ch_select', please set host astparam, hostnamebydipswitch, to 'n'.

Related astparam

- ch_select: default *ch-select* value. If not defined, ch_select's default value is got from on-board 4-bits DIP switch.
- astaccess
- reset_ch_on_boot: If this value is 'n', 'e_reconnect' command with {ch-select} specified will save the specified value to 'ch_select' astparam.
- multicast_ip_prefix:
- hostnamebydipswitch:
- hostname_id:

Related Iparam

- ACCESS_ON: Default from astparam, astaccess. Toggled through 'e_reconnect' and 'e_stop_link' commands.
- CH_SELECT

Version History

- FW >= A7.0.0

Examples

Ex1: Start all services as 'current' channel. 'current' channel is by default got from on-board 4-bits DIP switch. **Only this command will trigger FW to re-read 4-bits DIP switch value.**

```
/ # e e_reconnect
/ #
```

Ex2: Start/re-start all services as channel '7811'.

```
/ # e e_reconnect::7811
/ #
```

Stop Link [Client Only]

Used to stop services.

Syntax [stop all]

e e_stop_link

Syntax [specify services]

e e_stop_link::{{z|Z}|{v||u||a||r||s||p}}}

Parameters

- {{z|Z}|{v||u||a||r||s||p}}:
 - Used to specify which function to apply.
 - v: video over IP
 - u: USB over IP
 - a: Audio over IP
 - r: IR over IP
 - s: Serial over IP
 - p: GPIO over IP
 - z: all features. == vuasrp
 - Z: all features except video over IP. == uasrp
 - {z|Z} is exclusive from {v||u||a||r||s||p}
 - For example: 'Zuarsp', 'zvua', 'zp'... are all invalid format.
 - {v||u||a||r||s||p} can be any combination and order which indicate stop order of specified service.
 - For example: "uar" means first "USB over IP", then "Audio over IP" and "IR over IP" at last.

Usage Information

- Only services specified will be stopped.
- Without specifying services, all services will be stopped.
- Client's 'e_stop_link' state will NOT persist between client reboot. You need to manually save astparam, access_on_x, if you want to persist the state between client reboot.
- If astparam, reset_ch_on_boot, is set to 'y', the setting of 'access_on_x' will be reset to the default value, astaccess. So, when you set 'access_on_x', remember to set 'astaccess', also.

Related astparam

- reset_ch_on_boot: Set to 'y' will reset following 'access_on_x' value back to default value, astaccess.
- astaccess: Default value of following 'access_on_x'. Default is 'y'.
- access_on_v: y|n. Set to 'n' to not starting video service on boot.
- access_on_u: y|n. Set to 'n' to not starting USB service on boot.
- access_on_a: y|n. Set to 'n' to not starting audio service on boot.
- access_on_r: y|n. Set to 'n' to not starting IR service on boot.

Product: AST15XX
Date: 9/12/2017
FW Rev:

Doc Type:

- access_on_s: y|n. Set to 'n' to not starting Serial over IP service on boot.
- access_on_p: y|n. Set to 'n' to not starting GPIO over IP service on boot.

Related Imparam

- RESET_CH_ON_BOOT
- ACCESS_ON: From astparam, astaccess.
- ACCESS_ON_V, ACCESS_ON_U, ACCESS_ON_A, ACCESS_ON_R, ACCESS_ON_S, ACCESS_ON_P

Version History

- FW >= A7.0.0

Examples

Ex1: Stop all services.

```
/ # e e_stop_link  
/ #
```

Ex2: Stop USB services.

```
/ # e e_stop_link::u  
/ #
```

Ex3: Stop audio and video services.

```
/ # e e_stop_link::av  
/ #
```

Stop Link [Host Only]

Used to stop all services.

Syntax

e e_stop_link

Parameters

No parameter required.

Usage Information

- All services will be stopped. Can't specify which service to stop.

Version History

- FW >= A5.0.0

Examples

Ex1: Stop all services.

```
/ # e e_stop_link  
/ #
```

Get Current Channel Setting

Syntax

Imparam g CH_SELECT[_who]

Parameters

Use following Imparam to get current channel setting:

Host:

Imparam	Description
CH_SELECT	All service uses this channel.

Client:

Imparam	Description
CH_SELECT	System default channel. Used if following CH_SELECT_X is invalid/not defined.
CH_SELECT_V	Video over IP channel.
CH_SELECT_U	USBoIP/KMoIP channel.
CH_SELECT_A	Audio over IP channel.
CH_SELECT_R	IR over IP channel.
CH_SELECT_S	Serial over IP channel.
CH_SELECT_P	GPIO over IP channel.

Related astparam

- ch_select: default value of all following 'ch_select_x'. If not defined, ch_select's default value is got from on-board 4-bits DIP switch.
- FW will automatically maintain following 'ch_select_x' astparam:
 - ch_select_v: {ch-select|host-ip-addr} value for video over IP.
 - ch_select_u: {ch-select|host-ip-addr} value for USB over IP.
 - ch_select_a: {ch-select|host-ip-addr} value for Audio over IP.
 - ch_select_r: {ch-select|host-ip-addr} value for IR over IP.
 - ch_select_s: {ch-select|host-ip-addr} value for Serial over IP.
 - ch_select_p: {ch-select|host-ip-addr} value for GPIO over IP.
 - ch_select_i2s: default value of 'ch_select_a'. Exists for legacy reason. Don't use it.

Product: AST15XX
Date: 9/12/2017
FW Rev:

Doc Type:

- ch_select_soip2: default value of 'ch_select_s'. Exists for legacy reason. Don't use it.
- NOTE: **modify above 'ch_select_x's is not allowed and will cause un-expected FW behavior.**
- reset_ch_on_boot: If this value is 'n', 'e_reconnect' command with {ch-select|host-ip-addr} specified will save the specified value to specified 'ch_select_x' astparam.

Version History

- FW >= A7.0.0

Examples

Ex1: Client get current video over IP's channel

```
/ # lmparam g CH_SELECT_V  
9999/ #
```

IP Network Commands

Change Host Name ID

Syntax

astparam s hostname_id id;e e_chg_hostname

Parameters

- id: The host name id to be configured.

Usage Information

- See 'all about astparam' document for astparam command details.
- A device's host name is constructed by a set of astparams.
 - host: {hostname_prefix}{hostname_tx_middle}{hostname_id}
■ Ex: ast3-gateway0000
 - client: {hostname_prefix}{hostname_rx_middle}{hostname_id}
■ Ex: ast3-client82EAA841BDF1
- [Host] For backward compatible with legacy "host channel switching" APIs,
 - host's **hostname_id MUST be the same as host's channel setting, ch_select.**
■ **Must be the same format as 'ch_select', 0000~9999.** FW will use '0000' default value if invalid format detected.
 - host's 'e_chg_hostname' event will automatically trigger channel setting change, 'e_reconnect::{ch_select}'.
 - 'e_reconnect::{ch_select}' will automatically change 'hostname_id' to specified 'ch_select' value.
 - If you want to break this legacy behavior and use different 'hostname_id' from 'ch_select', please set host astparam, hostnamebydipswitch, to 'n'.

Related astparam

- hostname_id
- hostname_prefix
- hostname_tx_middle
- hostname_rx_middle
- hostnamebydipswitch

Related Imparam

- HOSTNAME_PREFIX
- HOSTNAME_TX_MIDDLE
- HOSTNAME_RX_MIDDLE
- HOSTNAME_ID

Version History

- FW >= A7.0.0

Examples

Product: AST15XX
Date: 9/12/2017
FW Rev:

Doc Type:

Ex1: Change client's host name to 'ast3-Rx-test1234'. And save it into flash.

```
/# astparam s hostname_id test1234;e e_chg_hostname;astparam save  
/#
```

Get Host Name

There are two commands which return different format of host name. First command returns full host name string. Second command returns only the ID part of host name.

Syntax

hostname

Imparam g HOSTNAME_ID

Related astparm

- hostname_id
- hostname_prefix
- hostname_tx_middle
- hostname_rx_middle
- hostnamebydipswitch

Version History

- FW >= A5.0.0

Examples

Ex1: Get full host name

```
/ # hostname  
ast3-Tx-9999  
/ #
```

Ex2: Get the ID part of host name

```
/ # lmparam g HOSTNAME_ID  
9999/ #
```

Get IP Address

Syntax

Imparam g MY_IP

Usage Information

- For other setting of IP, see document "Console APIs/All About astparam".

Related astparam

- ip_mode
- ipaddr
- netmask
- gatewayip

Version History

- FW >= A6.0.0

Examples

Ex1: Get IP address

```
/ # lmparam g MY_IP  
169.254.10.133/ #
```

Get MAC Address

Syntax

lmparam g MY_MAC

Usage Information

- For other setting of MAC address, see document "Console APIs/All About astparam".

Related astparam

- ethaddr
- random_mac

Version History

- FW >= A6.0.0

Examples

Ex1: Get MAC address

```
/ # lmparam g MY_MAC  
02C7C324D7E3/ #
```

Get Ethernet Link Status and Mode

Syntax

Imparam g ETH_LINK_STATE

Imparam g ETH_LINK_MODE

Parameters

- ETH_LINK_STATE: Get Ethernet link status. Result can be 'on' or 'off'.
- ETH_LINK_MODE: Get Ethernet link mode. Result can be '10M', '100M' or '1G'.
Only valid when ETH_LINK_STATE is 'on'.

Version History

- FW >= A5.0.0

Examples

Ex1: Get Link State

```
/ # lmparam g ETH_LINK_STATE  
on/ #
```

Ex2: Get Link Mode

```
/ # lmparam g ETH_LINK_MODE  
1G/ #
```

Node Discovery and Query

Node Discovery and Query

This command is an upgrade version of legacy 'node_list' command.

Use this command to query and list devices node information (node_info) in the network. node_info is a binary file containing a list of key/value pair. User can use this node_query command to add/modify any node_info file. Currently, FW uses only one node_info file named 'essential'. 'essential' node_info is the default node_info file used by FW's link manager.

key/value pairs can be dynamically created by users or FW itself. FW's key/value pairs in 'essential' node_info file are usually mapped from FW's link manager. Those key's meaning is explained in **Appendix D List of Imparam**.

Syntax

```
node_query [--reply_type reply_file_name] [--match_key key=value] [--set_key key=value] [--max n] [--if input_file_name] [--of output_file_name] [--period ms] [{--get_key key}|--dump] [--json] [--intf local_ip_addr]
```

Parameters

- reply_type:
 - Specify which node_info file to query. node_responder will reply the content of 'reply_file_name'.
 - Default value: *essential*
- match_key:
 - Enter a key/value pair to match replied node_info. Only matched node_info will be considered query matched.
 - There is no default value.
 - Can has up to **8** match_key key/value pairs in a single command. ALL match_key matched will be considered query matched.
 - Example:
 - node_query --match_key IS_HOST=y --match_key MY_IP=169.254.11.22
- set_key:
 - Enter a key/value pair to add into output node_info.
 - there is no default value.
 - Can has up to **8** set_key key/value pairs in a single command. All key/value pairs will be added into output node_info.
- get_key:
 - Enter a *key* to output. Instead of '--dump' which will print out all key/value pairs of node_info, only specified key's value will be printed out.
 - There is no default value.
 - Can has up to **8** get_key key/value pairs in a single command. ALL get_key matched will be printed out.

- If node_info doesn't contain the query key, an empty string "" will be printed out like 'THE_KEY=""'.
- max:
 - Enter a decimal number. query process stops when matched node_info count over this max value.
 - Default value: 600
- if:
 - If specified, node_query query node_info from local *input_file_name* file instead of query from network.
 - Default value: from network if not specified.
- of:
 - If specified, matched node_info will be written into *output_file_name*.
 - There is no default value.
 - If there are multiple matched node_info, previous match will be overwritten by new one. *output_file_name* will only contain the last matched one, NOT all matched node_info.
- dump:
 - Print all key/value pairs of matched node_info to stdout.
 - If has '--get_key' parameter, '--dump' will be ignored.
- period:
 - Enter a millisecond decimal value to specify query period. Default value is automatically calculated according to '--max' value. If you need a longer query period with a low '--max' value, then set the parameter to overwrite default query period.
- intf: [FW > b3015]
 - In 'xxx.xxx.xxx.xxx' ip format. Used to specify local network interface for outgoing node query packets.
 - This is only required when running on a machine with multiple network interfaces. ⇒
For Windows version node_query.
- json:
 - Print output using JSON format. Normally used for Web protocol.
 - Applies to '--dump' or '--get_key's output format.
 - NOT apply to '--if' or '--of's file format.
 - All JSON attributes are in 'string' type. Except the last 'count' attribute which is an integer number. See **Ex11/Ex12** for sample result.
 - The JSON object contains following attributes:
 - **nodes**: An array. Contains a list of objects. Each object represents a device's node_info. The content of device's node_info will depends on what you queried (--get_key or --dump) and what 'reply_type' you specified. It is recommended to use MAC address, 'MY_MAC', to identify which node_info belongs to which node.
 - **count**: An integer. The 'count' attribute is the number of matched and listed nodes. It should be the same as the array size of 'nodes' array attribute. If not, there is something wrong.
 - Example: (Two nodes listed)

Product: AST15XX
Date: 9/12/2017
FW Rev:

Doc Type:

```
{
  "nodes": [
    {
      "IS_HOST": "y",
      "MY_MAC": "02C7C324D7E3",
      "HOSTNAME": "ast3-gateway0000",
      "MY_IP": "169.254.10.133",
      "STATE": "s_srv_on"
    },
    {
      "IS_HOST": "n",
      "MY_MAC": "82EAA841BDF1",
      "HOSTNAME": "ast3-client82EAA841BDF1",
      "MY_IP": "169.254.10.167",
      "STATE": "s_srv_on"
    }
  ],
  "count": 2
}
```

Query Result

Those key's meaning is explained in **Appendix D List of Imparam.**

Host Example [Without --json]

```
MY_IP=169.254.10.133
IS_HOST=y
MULTICAST_ON=n
NO_VIDEO=n
NO_USB=n
NO_KMOIP=n
NO_I2S=n
NO_SOIP=n
SOIP_GUEST_ON=n
SOIP_TYPE=1
NO_IR=n
NO_PWRBTN=n
HOSTNAME=ast3-Tx-9999
CH_SELECT=9999
STATE=s_srv_on
REPLY_TYPE=essential
MY_MAC=02C7C324D7E3
```

Client Example [Without --json]

```
MY_IP=169.254.10.167
IS_HOST=n
HOSTNAME=ast3-Rx-82EAA841BDF1
MULTICAST_ON=n
CH_SELECT=0000
```


Product: AST15XX
Date: 9/12/2017
FW Rev:

Doc Type:

```
NO_VIDEO=n
NO_USB=n
NO_KMOIP=n
NO_I2S=n
NO_SOIP=n
SOIP_GUEST_ON=n
SOIP_TYPE=2
NO_IR=y
NO_PWRBTN=y
STATE=s_srv_on
REPLY_TYPE=essential
MY_MAC=82EAA841BDF1
```

Usage Information

- The maximum size of node_info file must < 64KB. Key/value pairs are saved as string characters.

Version History

- FW >= A7.0.0

Examples

Ex1: Print node_info of all devices to stdout.

```
/ # node_query --dump

-----
MY_IP=169.254.10.133
IS_HOST=y
HOSTNAME=ast3-Tx-0000
MULTICAST_ON=n
CH_SELECT=0000
NO_VIDEO=n
NO_USB=n
NO_KMOIP=n
NO_I2S=n
NO_SOIP=n
SOIP_GUEST_ON=n
SOIP_TYPE=1
NO_IR=n
NO_PWRBTN=n
STATE=s_attaching
REPLY_TYPE=essential
MY_MAC=02C7C324D7E3

-----
MY_IP=169.254.11.46
IS_HOST=n
HOSTNAME=ast3-Rx-82EA88FCD397
MULTICAST_ON=n
CH_SELECT=1111
NO_VIDEO=n
```

Product: AST15XX
Date: 9/12/2017
FW Rev:

Doc Type:

```
NO_USB=n
NO_KMOIP=n
NO_I2S=n
NO_SOIP=n
SOIP_GUEST_ON=n
SOIP_TYPE=2
NO_IR=n
NO_PWRBTN=y
STATE=s_srv_on
REPLY_TYPE=essential
MY_MAC=82EA88FCD397
/ #
```

Ex2: List all nodes' MY_MAC value.

```
/ # node_query --get_key MY_MAC

MY_MAC=02C7C324D7E3
MY_MAC=82EA88FCD397
/ #
```

Ex3: List all host nodes' STATE value. (host node's IS_HOST=y). There is only one host in this example. So, only one node's STATE is printed out.

```
/ # node_query --get_key STATE --match_key IS_HOST=y

STATE=s_attaching
/ #
```

Ex4: Discover a node which 'IS_HOST=y' and 'CH_SELECT=0000'. Only print one matched. And print out both 'NO_VIDEO' and 'MULTICAST_ON' keys.

```
/ # node_query --match_key IS_HOST=y --match_key CH_SELECT=0000 --max 1 -
-get_key NO_VIDEO --get_key MULTICAST_ON

NO_VIDEO=n
MULTICAST_ON=n
/ #
```

Ex5: Discover a node which 'IS_HOST=y' and 'CH_SELECT=0000'. Only print one matched. Print out its 'MY_IP' keys. ⇒ Find host with channel set to 0000's IP address.

```
/ # node_query --match_key IS_HOST=y --match_key CH_SELECT --max 1 --
get_key MY_IP
```

```
MY_IP=169.254.10.133  
/ #
```

Ex6: Create a local node_info file named "my_node_info" and add "MY_KEY=123" and "MY_KEY2=abc" into it.

```
/ # node_query --if my_node_info --of my_node_info --set_key MY_KEY=123 -  
-set_key MY_KEY2=abc  
/ #
```

Ex7: Dump my_node_info node_info file from previous example.

```
/ # node_query --dump --if my_node_info  
  
-----  
MY_KEY=123  
MY_KEY2=abc  
/ #
```

Ex8: Add "MY_KEY=123" into the default "essential" local node_info file.

```
/ # node_query --if essential --of essential --set_key MY_KEY=123  
/ #
```

Ex9: Create a local node_info file named "my_essential" based on default "essential" local node_info file. Add "MY_KEY=my_value" into it.

```
/ # node_query --if essential --of my_essential --set_key MY_KEY=my_value  
/ #
```

Ex10: Query a node's multicast channel.

- The node's IP address is known as 'MY_IP=169.254.10.133'.
- multicast channel's key is 'CH_SELECT'.

```
/# node_query --match_key MY_IP=169.254.10.133 --get_key CH_SELECT  
  
CH_SELECT=9999  
/ #
```

Ex11: Print node_info of all devices to stdout using JSON format.

Product: AST15XX
Date: 9/12/2017
FW Rev:

Doc Type:

```
/ # node_query --dump --json
{
  "nodes": [
    {
      "MY_IP": "169.254.10.167",
      "IS_HOST": "n",
      "HOSTNAME": "ast3-client82EAA841BDF1",
      "MULTICAST_ON": "y",
      "NO_VIDEO": "n",
      "NO_USB": "n",
      "NO_KMOIP": "n",
      "NO_I2S": "n",
      "NO_SOIP": "n",
      "SOIP_GUEST_ON": "n",
      "SOIP_TYPE": "2",
      "NO_IR": "n",
      "NO_PWRBTN": "y",
      "ACCESS_ON_P": "y",
      "ACCESS_ON_U": "y",
      "CH_SELECT_P": "0000",
      "CH_SELECT_V": "0000",
      "CH_SELECT_U": "0000",
      "CH_SELECT_A": "0000",
      "CH_SELECT_R": "0000",
      "CH_SELECT_S": "0000",
      "STATE": "s_srv_on",
      "ACCESS_ON_A": "n",
      "ACCESS_ON_S": "n",
      "ACCESS_ON_R": "n",
      "ACCESS_ON_V": "n",
      "REPLY_TYPE": "essential",
      "MY_MAC": "82EAA841BDF1"
    },
    {
      "MY_IP": "169.254.10.133",
      "IS_HOST": "y",
      "HOSTNAME": "ast3-gateway0000",
      "MULTICAST_ON": "y",
      "NO_VIDEO": "n",
      "NO_USB": "n",
      "NO_KMOIP": "n",
      "NO_I2S": "n",
      "NO_SOIP": "n",
      "SOIP_GUEST_ON": "n",
      "SOIP_TYPE": "1",
      "NO_IR": "n",
      "NO_PWRBTN": "n",
      "CH_SELECT": "0000",
      "ACCESS_ON": "y",
      "STATE": "s_srv_on",
      "REPLY_TYPE": "essential",
      "MY_MAC": "02C7C324D7E3"
    }
  ],
  "count": 2
}
```

```
/ #
```

Ex12: Query all node's following info and output using json format.

- IS_HOST
- MY_MAC
- HOSTNAME
- MY_IP
- STATE

There are total two nodes replied.

```
/# node_query --get_key IS_HOST --get_key MY_MAC --get_key HOSTNAME --  
get_key MY_IP --get_key STATE --json  
  
{  
  "nodes": [  
    {  
      "IS_HOST": "y",  
      "MY_MAC": "02C7C324D7E3",  
      "HOSTNAME": "ast3-gateway0000",  
      "MY_IP": "169.254.10.133",  
      "STATE": "s_srv_on"  
    },  
    {  
      "IS_HOST": "n",  
      "MY_MAC": "82EAA841BDF1",  
      "HOSTNAME": "ast3-client82EAA841BDF1",  
      "MY_IP": "169.254.10.167",  
      "STATE": "s_srv_on"  
    }  
  ],  
  "count": 2  
}  
/ #
```

Video Commands

Configure Graphic/Video Mode [Host Only]

Syntax [LM Get]

Imparam g V_QUALITY_MODE

Syntax [LM Toggle]

e e_btn_toggle_video_profile

Syntax [LM Set Video Mode]

e e_btn_toggle_video_profile_video

Syntax [LM Set Graphic Mode]

e e_btn_toggle_video_profile_graphic

Syntax [Driver Get]

cat /sys/devices/platform/videoip/QualityMode

Syntax [Driver Set]

echo {quality_mode} > /sys/devices/platform/videoip/QualityMode

Parameters

- quality_mode: Can be '0' or '-1'. Integer values from 1 to 5 are also valid.
 - 0: Graphic mode. Picture quality first. May drop frame.
 - 1: Video mode. Smooth frame rate first. Will change video quality dynamically.
 - 1 ~ 5: The higher the value, the lower the video quality.

Usage Information

- Driver command is used to toggle Graphic/Video mode in runtime. The setting will be overwritten by link manager once video input timing changed.
- Use link manager event, e_btn_toggle_video_profile, e_btn_toggle_video_profile_graphic and e_btn_toggle_video_profile_video, to toggle between Video/Graphic mode. Toggle through link manager is a recommended way because link manager will handle video state correctly and auto save the setting into flash.
- Use astparam, ast_video_quality_mode, to save the setting into flash. (need reset to take effect)

Related Document

- Documents/Video over IP/Notes on Fast Video Switching

Related event

- e_btn_toggle_video_profile
- e_btn_toggle_video_profile_graphic
- e_btn_toggle_video_profile_video

Related Imparam

- V_QUALITY_MODE

Related astparam

- ast_video_quality_mode

Version History

- [FW >= A5.0.0] e_btn_toggle_video_profile
- [FW >= A7.0.0] e_btn_toggle_video_profile_graphic,
e_btn_toggle_video_profile_video

Examples

Ex1: Get current mode and toggle change. (Through link manager)

```
/ # lmparam g V_QUALITY_MODE  
0/ #  
/ # e e_btn_toggle_video_profile  
/ #
```

Ex2: Force to Graphic mode.

```
/ # e e_btn_toggle_video_profile_graphic  
/ #
```

Set Anti-Dither Mode [Host Only]

Syntax [LM Get]

Imparam g V_BCD_THRESHOLD

Syntax [LM Toggle]

e e_btn_toggle_video_anti_dither

Syntax [LM Set]

e e_btn_toggle_video_anti_dither_{mode}

Syntax [Driver Get]

cat /sys/devices/platform/videoip/bcd_threshold

Syntax [Driver Set]

echo {mode} > /sys/devices/platform/videoip/bcd_threshold

Parameters

- mode: Can be '0', '1' or '2'.
 - 0: Turn off anti-dither.
 - 1: Mode 1.
 - 2: Mode 2. Use Mode 2 if Mode 1 can't resolve dithering problem.

Usage Information

- Driver command is used to toggle Graphic/Video mode in runtime. The setting will be overwritten by link manager once video input timing changed.
- Use link manager event, e_btn_toggle_video_anti_dither, and e_btn_toggle_video_anti_dither_{mode}, to toggle between Anti-Dither mode. Toggle through link manager is a recommended way because link manager will handle video state correctly and auto save the setting into flash.
- Use astparam, v_bcd_threshold, to save the setting into flash. (need reset to take effect)

Related event

- e_btn_toggle_video_anti_dither
- e_btn_toggle_video_anti_dither_0
- e_btn_toggle_video_anti_dither_1
- e_btn_toggle_video_anti_dither_2

Related Imparam

- V_BCD_THRESHOLD

Related astparam

- v_bcd_threshold

Version History

Product: AST15XX
Date: 9/12/2017
FW Rev:

Doc Type:

- [FW >= A5.0.0] e_btn_toggle_video_anti_dither
- [FW >= A7.0.0] e_btn_toggle_video_anti_dither_{mode}

Examples

Ex1: Get current mode and toggle change. (Through link manager)

```
/ # lmparam g V_BCD_THRESHOLD  
0/ #  
/ # e e_btn_toggle_video_anti_dither  
/ #
```

Ex2: Force to mode 2.

```
/ # e e_btn_toggle_video_anti_dither_2  
/ #
```

Enable/Disable Loopback Port [Host Only]

Syntax [LM Get]

Imparam g V_LOOPBACK_ENABLED

Syntax [LM Toggle]

e e_btn_toggle_snoop

Syntax [LM Turn On]

e e_btn_toggle_snoop_on

Syntax [LM Turn Off]

e e_btn_toggle_snoop_off

Usage Information

- Not all HW platform support this feature.

Related event

- e_btn_toggle_snoop
- e_btn_toggle_snoop_on
- e_btn_toggle_snoop_off

Related Imparam

- LOOPBACK_DEFAULT_ON: Boot up default status. From astparam, loopback_default_on.
- V_LOOPBACK_ENABLED: Current runtime status.

Related astparam

- loopback_default_on

Version History

- [FW >= A5.0.0] e_btn_toggle_snoop
- [FW >= A7.0.0] e_btn_toggle_snoop_on, e_btn_toggle_snoop_off

Examples

Ex1: Get current mode and toggle change. (Through link manager)

```
/ # lmparam g V_LOOPBACK_ENABLED
y/ #
/ # e e_btn_toggle_snoop
/ #
```

Ex2: Force to mode off.

Product: AST15XX
Date: 9/12/2017
FW Rev:

Doc Type:

```
/ # e e_btn_toggle_snoop_off  
/ #
```

Frame Rate Control [Host Only]

Used to control video encode's frame rate.

Syntax

```
echo {frame_rate} > /sys/devices/platform/videoip/frame_rate_control
```

Parameters

- frame_rate: Default set to '0' means 'no limitation' use best effort. Set to other value will limit the maximum video encode frame rate.
 - Valid range: 0,1,2,...,60.
 - Frame rate = (v_frame_rate / 60) * input refresh rate
 - For example, set to '30' under 1080p60Hz:
 - Frame rate = (30/60) * 60 = 30 FPS

Usage Information

- To keep the setting applied after system reboot, save this value into astparam, v_frame_rate.
- Some video profile may already apply lower encode frame rate. This command will 'add' more frame rate control based on video profile setting. For example, if video profile already set to 30fps, then set '30' in this command will results in '15' fps.

Related astparam

- v_frame_rate

Version History

- FW >= A6.0.0

Examples

Ex1: Set 1080p60Hz's encode frame rate to 30fps

```
/ # echo 30 > /sys/devices/platform/videoip/frame_rate_control
/ #
```

Bit Rate Control [Host Only]

Bit rate control is done by changing video profile. To change video profile, set astparam, profile, and reboot system to take effect. See 'Documents/Console APIs/All About astparam' document for details.

Usage Information

- Please contact with ASPEED if you need to runtime change profile.

Related astparam

- profile

Version History

- FW >= A6.0.0

Examples

Ex1: Set bit rate to 150Mbps.

```
/ # astparam s profile 150M;astparam save;reboot
/ #
```

Detach/Attach Video Input Port [Host Only]

Used to toggle upstream video port hotplug pin. Only valid for HDMI or DVI input.

Syntax

```
echo {0|1} > /sys/devices/platform/videoip/detach_input_port
```

Parameters

- {0|1}
 - 1: detach
 - 0: attach

Usage Information

- This command is typically used to trigger upstream video port hotplug event. Detach then attach will cause video source re-read EDID, re-do HDCP and change video sync.

Version History

- FW >= A5.0.0

Examples

Ex1: Detach upstream video port.

```
/ # echo 1 > /sys/devices/platform/videoip/detach_input_port  
/ #
```

Upload/Copy Video Downstream Sink's EDID

This command can be used to dynamically trigger EDID upload/copy.

Syntax

```
echo {primary|secondary} > /sys/devices/platform/videoip/edid_use
```

Parameters

- primary: The attached sink's EDID will be uploaded to host's EEPROM.
- secondary: The attached sink's EDID will not be taken by host.

Usage Information

- When applied to host, this setting is applied to host's loopback downstream sink.
- Under multicast mode, multiple clients can set to 'primary'. However, host can only take the last one uploaded. The best practice is to set ONLY one client to 'primary'. So that host won't get confused.
- FW has EDID patch function and default is enabled. So, uploaded EDID will be patched first before been written into EEPROM.
- For other EDID related commands and topic, please reference to 'Documents/Video over IP/AST15XX EDID Customization Guide' and 'Documents/Video over IP/AST15XX EDID Update Policy'.

Related Document

- Documents/Video over IP/AST15XX EDID Update Policy
- Documents/Video over IP/AST15XX EDID Customization Guide

Related astparam

- edid_use
- remote_edid_patch
- loopback_edid_patch

Version History

- FW >= A5.0.0

Examples

Ex1: Trigger EDID upload. (Assume this client's default value is 'secondary'.)

```
/ # echo primary > /sys/devices/platform/videoip/edid_use  
/ #
```

Get Current EDID Reported to Upstream [Host Only]

Syntax

cat /sys/devices/platform/videoip/edid_cache

Version History

- FW >= A5.0.0

Examples

Ex1: Get current used EDID

```
/ # cat /sys/devices/platform/videoip/edid_cache
00 ff ff ff | ff ff ff 00 | 10 ac c0 a0 | 4c 4e 33 30 |
06 19 01 03 | 80 35 1e 78 | ea e2 45 a8 | 55 4d a3 26 |
0b 50 54 a5 | 4b 00 71 4f | 81 80 a9 c0 | a9 40 d1 c0 |
e1 00 01 01 | 01 01 a3 66 | 00 a0 f0 70 | 1f 80 30 20 |
35 00 0f 28 | 21 00 00 1a | 00 00 00 ff | 00 44 33 43 |
38 58 35 32 | 34 30 33 4e | 4c 0a 00 00 | 00 fc 00 44 |
45 4c 4c 20 | 50 32 34 31 | 35 51 0a 20 | 00 00 00 fd |
00 1d 4c 1e | 8c 1e 00 0a | 20 20 20 20 | 20 20 01 dc |
02 03 2a f1 | 53 90 05 04 | 02 07 16 01 | 14 1f 12 13 |
27 20 21 22 | 03 06 11 15 | 23 09 07 07 | 6d 03 0c 00 |
10 00 30 3c | 20 00 60 03 | 02 01 02 3a | 80 18 71 38 |
2d 40 58 2c | 25 00 0f 28 | 21 00 00 1f | 01 1d 80 18 |
71 1c 16 20 | 58 2c 25 00 | 0f 28 21 00 | 00 9e 04 74 |
00 30 f2 70 | 5a 80 b0 58 | 8a 00 0f 28 | 21 00 00 1e |
56 5e 00 a0 | a0 a0 29 50 | 30 20 35 00 | 0f 28 21 00 |
00 1a 00 00 | 00 00 00 00 | 00 00 00 00 | 00 00 00 f9 |

/ #
```


Get Video Source Timing Information

Syntax

cat /sys/devices/platform/videoip/timing_info

Version History

- FW >= A6.2.0

Examples

Ex1: Get video source timing information

```
/ # cat /sys/devices/platform/videoip/timing_info
Timing Table: Serial Number[0x000D] [1920]X[1080] [60]Hz
              Pixel Rate: 148351KHz, Htotal: 2200, Vtotal: 1125
              Hbp: 148, Vbp: 36, Hsw: 44, Vsw: 5
              Progressive, HPos, VPos
Color Depth: [0]
HDCP: [Off]
HDCP Convert: Disable
Capture Windows: [1920]X[1080] [60]Hz
Compress Windows: [1920]X[1080] [60]Hz
Active Windows: [1920]X[1080] [60]Hz
CRT Windows: [1920]X[1080]
Scan Mode: Progressive
Signal Type: HDMI 16:9
/ #
```

Get Video Engine State

Syntax

cat /sys/devices/platform/videoip/State

Result

- Host:
 - INITIALIZING: Initializing.
 - OPERATING: Encoding and streaming out.
 - operating: Encoding without streaming. (No client attached yet.)
 - DETECTING_MODE: Detecting video input timing.
 - HOST_STATE_SUSPENDING: Engine suspended.
- Client:
 - INITIALIZING: Initializing.
 - OPERATING: Decoding video stream.
 - WAITING_HOST_MODE: Waiting for video stream.
 - CLIENT_STATE_SUSPENDING: Engine suspended.

Version History

- FW >= A5.0.0

Examples

Ex1: Get current video engine state

```
/ # cat /sys/devices/platform/videoip/State
OPERATING
/ #
```

Get Video Sink Information

Syntax

cat /sys/devices/platform/display/monitor_info

Usage Information

- [Host] This command shows loopback port video sink's information. NOT client's sink information.
- Under dual port output situation, this command shows current primary output port's sink information.

Version History

- FW >= A5.0.0

Examples

Ex1: Get video sink information. (Monitor attached.)

```
/ # cat /sys/devices/platform/display/monitor_info
attached=y

Checksum: ok
CEA Ext: y
CEA Ext Checksum: ok
HDMI 2.0: n
Input Signal: digital
Support Features:
    yuv: y
    hdr: n
Preferred Timing:
    3840x2160@30Hz,Prog (idx 107, sn 107)

EDID:
  00 ff ff ff | ff ff ff 00 | 10 ac c0 a0 | 4c 4e 33 30 |
  06 19 01 03 | 80 35 1e 78 | ea e2 45 a8 | 55 4d a3 26 |
  0b 50 54 a5 | 4b 00 71 4f | 81 80 a9 c0 | a9 40 d1 c0 |
  e1 00 01 01 | 01 01 a3 66 | 00 a0 f0 70 | 1f 80 30 20 |
  35 00 0f 28 | 21 00 00 1a | 00 00 00 ff | 00 44 33 43 |
  38 58 35 32 | 34 30 33 4e | 4c 0a 00 00 | 00 fc 00 44 |
  45 4c 4c 20 | 50 32 34 31 | 35 51 0a 20 | 00 00 00 fd |
  00 1d 4c 1e | 8c 1e 00 0a | 20 20 20 20 | 20 20 01 dc |
  02 03 2a f1 | 53 90 05 04 | 02 07 16 01 | 14 1f 12 13 |
  27 20 21 22 | 03 06 11 15 | 23 09 07 07 | 6d 03 0c 00 |
  10 00 30 3c | 20 00 60 03 | 02 01 02 3a | 80 18 71 38 |
  2d 40 58 2c | 25 00 0f 28 | 21 00 00 1f | 01 1d 80 18 |
  71 1c 16 20 | 58 2c 25 00 | 0f 28 21 00 | 00 9e 04 74 |
  00 30 f2 70 | 5a 80 b0 58 | 8a 00 0f 28 | 21 00 00 1e |
  56 5e 00 a0 | a0 a0 29 50 | 30 20 35 00 | 0f 28 21 00 |
  00 1a 00 00 | 00 00 00 00 | 00 00 00 00 | 00 00 00 f9 |

/ #
```

Product: AST15XX
Date: 9/12/2017
FW Rev:

Doc Type:

Ex2: Get video sink information. (Monitor not attached.)

```
/ # cat /sys/devices/platform/display/monitor_info  
attached=n
```

Turn Off/On Video Output [Client Only]

Manually force video output off.

Syntax

```
echo {1|0} > /sys/devices/platform/display/screen_off
```

Parameters

- 1: Turn off.
- 0: Turn on.

Related astparam

- v_src_unavailable_timeout
- v_turn_off_screen_on_pwr_save

Version History

- FW >= A5.0.0
- [BUG] A6.4.12 is buggy. Will dead lock and cause client reboot.

Examples

Ex1: Turn off client video output.

```
/ # echo 1 > /sys/devices/platform/display/screen_off  
/ #
```

Ex2: Turn client video output back on.

```
/ # echo 0 > /sys/devices/platform/display/screen_off  
/ #
```

Pause Video Stream [Client Only]

Syntax [Get]

cat /sys/devices/platform/videoip/pause

Syntax [Set, Pause]

echo {mode} > /sys/devices/platform/videoip/pause

Parameters

- mode: Can be '0', '1' or '2'.
 - 0: Resume video streaming.
 - 1: Pause.
 - 2: Pause and black out.

Usage Information

- This is a runtime command. Pause status will be reset after client reboot.

Version History

- FW >= A6.4.3

Examples

Ex1: Pause screen, then resume.

```
/ # echo 1 > /sys/devices/platform/videoip/pause
/ #
/ # echo 0 > /sys/devices/platform/videoip/pause
/ #
```

Ex2: Black screen, then resume.

```
/ # echo 2 > /sys/devices/platform/videoip/pause
/ #
/ # echo 0 > /sys/devices/platform/videoip/pause
/ #
```

Product: AST15XX
Date: 9/12/2017
FW Rev:

Doc Type:

Select Video Input Port [Host Only]

Please read "Document/Video over IP/Notes on Video Dual Input Auto Selection" for details.

Scale Video Output [Client Only]

Please read "Document/Video over IP/Notes on Video Hybrid Mode" for details.

Video Snapshot

Please read "Document/Console APIs/Video over IP snapshot APIs" for details.

Product: AST15XX
Date: 9/12/2017
FW Rev:

Doc Type:

Video Wall Commands

Please read document under Documents/Video over IP/Video Wall/, for details:

- AST1500 Video Wall Web Configuration Without RS232 Chain
- AST1520 Video Wall API v2.docx

Show Picture When Link Down

Please read document under Documents/Console APIs/GUI Change Background Picture, for details.

OSD Commands

Show String OSD [Client Only]

Syntax [long command format]

```
e "e_osd_on_str::{YSTART::YMAX::ALPHA::MASK::OFF_TIME::STRING::FSIZE::FCOLOR}"
```

Syntax [shortcut, center aligned]

```
osd_on.sh "STRING"
```

Syntax [shortcut, top-left aligned, big font]

```
osd_on_tl.sh "STRING"
```

Syntax [turn off OSD]

```
e e_osd_off_str::{WHEN_OFF}
```

Syntax [shortcut, turn off OSD]

```
osd_off.sh {WHEN_OFF} &
```

Parameters

- YSTART:
 - Specify starting line number of OSD source buffer.
 - This number normally is '0'.
- YMAX:
 - Specify how many line number of OSD source buffer to be taken.
 - '200' is a typical value.
 - Don't set a number over the size of OSD source buffer which is 480(640x480) by default.
- ALPHA:
 - Alpha value. From 1 to 31.
 - 31 means no transparent.
- MASK:
 - Usually set to '1'. Means take off background color. Otherwise, set to '0'.
- OFF_TIMER:
 - Specify how many seconds the OSD string shows up.
 - Integer decimal value. 0, 1, 2, 3, 4,....
 - A special value, 'n', means never timeout.
- STRING: The string to display. 'printf' like string format.
 - Use following code for reserved characters:
 - \x3A: ':'
 - \x22: ""
 - \n: Line feed
 - \t: TAB
 - Any special characters can use its ASCII code to represent using '\xHH'

format.

- SIZE: String font size in pixels.
- COLOR: String font color in 32bits hex XRGB888 format, 0XXRRGGBB:
 - XX: Special code. Can be:
 - FF: Center aligned.
 - F0: Aligned to top-left.
 - RR: 8-bits red color.
 - GG: 8-bits green color.
 - BB: 8-bits blue color.
 - Example:
 - 0xFF00FF00: Green color. Center aligned.
 - 0xF0FF0000: Red color. Aligned to top-left.
- WHEN_OFF:
 - A decimal integer value.
 - Turn off OSD after 'WHEN_OFF' seconds.
 - Special value 'now' means turn off immediately

Usage Information

- Parameter "*STRING*" takes 'printf' like special character input. For example:
 - `osd_on.sh "Line 1 \n Line 2"`
- Use double quote,"", to quote OSD strings.
- The maximum number of characters of OSD string can show is software limited to 128 characters. See
SDK\application\ast_app\applet\msgd\msgd.h::OSD_STR_BUF_SIZE.

Related astparam

- `ui_default_res`

Version History

- FW >= A6.0.0

Examples

Ex1: Show OSD string "Hello World". Use default settings.

```
/ # osd_on.sh "Hello World"  
/ #
```

Ex2: Show OSD string "Hello World" at top-left edge of screen. Use default settings.

```
/ # osd_on_tl.sh "Hello World"  
/ #
```

Ex3: Turn off Ex1 and Ex2's OSD string after 3 seconds.

```
/ # osd_off.sh 3 &
```

Product: AST15XX
Date: 9/12/2017
FW Rev:

Doc Type:

```
/ #
```

Ex4: Show OSD string "Hello OSD". With:

- '25' transparency
- turn off after '5' seconds
- font size '30'
- font color 'red'

```
/ # e "e_osd_on_str::0::200::25::1::5::Hello OSD::30::0xFFFF0000"  
/ #
```

Ex5: Show OSD string "Test:123". Use default settings. Note: ':' is a reserved character.

```
/ # osd_on.sh "Test\x3A123"  
/ #
```

Product: AST15XX
Date: 9/12/2017
FW Rev:

Doc Type:

Show Picture OSD [Client Only]

Please read document, "Documents/Console APIs/OSD Control APIs", for details.

Product: AST15XX
Date: 9/12/2017
FW Rev:

Doc Type:

Audio over IP Commands

Audio Source Port Selection [Host Only]

Please read "Documents/Audio over IP (I2S)/AST1520 Audio Usage Scenario" for details.

Analog Audio Volume Control

Please read "Documents/Console APIs/All about astparam" for details. Search for "a_analog_in_vol" and "a_analog_out_vol".

USBoIP/KMoIP Commands

Manually Start/Stop USBoIP and KMoIP [Client Only]

Syntax [Start]

e e_reconnect::{*ch-select|host-ip-addr*}::**u**

Syntax [Stop]

e e_stop_link::u

Usage Information

- See "Switch Channel [Client Only]" and "Stop Link [Client Only]" sections for details.

Related astparam

- no_usb
- no_kmoip
- share_usb

Version History

- FW >= A7.0.0

Examples

Ex1: Manually stop USBoIP and KMoIP services.

```
/ # e e_stop_link::u  
/ #
```

Ex2: Manually re-start USBoIP and KMoIP services and connect to channel '0020'.

```
/ # e e_reconnect::0020::u  
/ #
```

Ex3: Client disable/enable USBoIP and KMoIP without reboot the system and persist between system reboot.

```
/ # # To Disable:  
/ # astparam s access_on_u n;astparam save  
/ # e e_stop_link::u  
/ #  
/ # # To Enable and connect to '0020' channel:  
/ # astparam s access_on_u y;astparam save  
/ # e e_reconnect::0020::u  
/ #
```

Product: AST15XX
Date: 9/12/2017
FW Rev:

Doc Type:

Manually Start/Stop USB over IP [Client Only]

Syntax [Start]

e e_start_usb

Syntax [Stop]

e e_stop_usb

Syntax [Toggle]

e e_btn_request_usb

Usage Information

- This command is for USB over IP, not KModIP.
- This command is used for runtime control. If you want to disable USB over IP permanently, please use `astparam, no_usb`, instead.

Related `astparam`

- `no_usb`
- `share_usb`
- `share_usb_auto_mode`
- `share_usb_on_first_peer`

Version History

- FW \geq A7.0.0

Examples

Ex1: Manually stop USB over IP.

```
/ # e e_stop_usb  
/ #
```


Manually Start/Stop KMoIP [Client Only]

Syntax [Start]

e e_start_kmoip

Syntax [Stop]

e e_stop_kmoip

Usage Information

- This command is for KMoIP.
- This command is used for runtime control. If you want to disable KMoIP permanently, please use `astparam, no_kmoip`, instead.

Related `astparam`

- `no_kmoip`

Version History

- FW \geq A7.0.0

Examples

Ex1: Manually stop KMoIP.

```
/ # e e_stop_kmoip  
/ #
```

Get USBIP Status [Client Only]

Syntax

ulmparam g U_USBIP_STATE

Result

- on: USBIP access permission is granted.
- off: USBIP access permission is not granted or is off.
- request: Requesting USBIP access permission.

Usage Information

- 'ulmparam' is a command similar to 'lmparam'. It is used to query USB Link Manager (uLM)'s internal parameters.

Version History

- FW >= A7.0.0

Examples

Ex1: Get current USBIP status. Current status result is 'on'.

```
/ # ulmparam g U_USBIP_STATE
on/ #
```

Get KMoIP Status [Client Only]

Syntax

ulmparam g U_KMOIP_STATE

Result

- on: KMoIP is enabled.
- off: KMoIP is disabled.

Usage Information

- 'ulmparam' is a command similar to 'lmparam'. It is used to query USB Link Manager (uLM)'s internal parameters.

Version History

- FW >= A7.0.0

Examples

Ex1: Get current KMoIP status. Current status result is 'on'.

```
/ # ulmparam g U_KMOIP_STATE  
on/ #
```

Get Attached Client IP [Host Only]

Syntax

Imparam g USB_CLIENT_IP

Result

- xxx.xxx.xxx.xxx: Return the IP address of last requested USBoIP client. If there is no one requested, will return '0.0.0.0'.

Usage Information

Version History

- FW >= A7.0.0

Examples

Ex1: Get current USBoIP status. Current requested client's IP is '169.254.10.167'.

```
/ # lmparam g USB_CLIENT_IP  
169.254.10.167/ #
```

Product: AST15XX
Date: 9/12/2017
FW Rev:

Doc Type:

USBoIP Device Exporting Policy [Client Only]

Please read document, "Documents/USB over IP/USBoIP Device Exporting Policy", for details.

I2C Access

FW provide a simple sysfs interface for simple I2C read/write. It is normally used for debug. Read/Write I2C takes and blocks CPU for a long time. So, if you have special application which access I2C quite a lot, please consider write a driver instead of use those commands.

Initial I2C Bus

Syntax

```
echo {BUS_NUM} {SPEED} > /sys/devices/platform/i2c/bus_init
```

Parameters

- BUS_NUM: I2C bus number. Starts from '1' to '7'.
- SPEED: Specify bus clock speed in Hz. Normally, I2C slave device accepts clock from 40000 to 100000 Hz.

Usage Information

- If there are multiple I2C slave devices attached to the same bus, the lowest bus speed configured will be applied to all slave devices.

Version History

- FW >= A5.0.0

Examples

Ex1: Set I2C bus #1 to use 40,000Hz bus speed.

```
/ # echo 1 40000 > /sys/devices/platform/i2c/bus_init  
/ #
```

Select I2C Bus and Device

Used to select which I2C slave device to access. User must specify which BUS_NUM and the device's DEV_ADDR. Further I2C read/write commands are targeting at a I2C device selected by this command.

Syntax

```
echo {BUS_NUM} {DEV_ADDR} > /sys/devices/platform/i2c/io_select
```

Parameters

- BUS_NUM: I2C bus number. Starts from '1' to '7'.
- DEV_ADDR: Device address to be selected. In 0xHH hex format.

Usage Information

Version History

- FW >= A5.0.0

Examples

Ex1: Select a I2C slave device located at bus#3 and using 0xA3 device address.

```
/ # echo 1 0xa3 > /sys/devices/platform/i2c/io_select  
/ #
```

Set Range for I2C Read

Used to specify I2C read range. If not specified, offset is 0 and read one byte.

Syntax

```
echo {OFFSET} {BYTES} > /sys/devices/platform/i2c/i_range
```

Parameters

- OFFSET: Register offset to read.
- BYTES: How many bytes to read.

Usage Information

Version History

- FW >= A5.0.0

Examples

See examples in next section.

Read/Write I2C Slave

Syntax [Byte Read]

cat /sys/devices/platform/i2c/io_value

Syntax [Byte Write]

echo {OFFSET} {VALUE} > /sys/devices/platform/i2c/io_value

Parameters

- OFFSET: Address offset to write. In one hex byte format.
- VALUE: Hex byte value to write. In one hex byte format.

Usage Information

- Read operation will read from offset and bytes specified by 'i_range' command in previous section.

Version History

Examples

Ex1: Read I2C Bus#1, Device address 0xA3, Offset 0x20, Read 3 bytes.

```
/ # echo 1 40000 > /sys/devices/platform/i2c/bus_init
/ # echo 1 0xA3 > /sys/devices/platform/i2c/io_select
/ # echo 0x20 3 > /sys/devices/platform/i2c/i_range
/ # cat /sys/devices/platform/i2c/io_value
Rd Offset:0x20, Length:0x3
20: 0x11 0x22 0x33
/ #
```

Ex2: Follow up Ex1. Now, read offset 0x10, read 0x10 bytes.

```
/ # echo 0x10 0x100 > /sys/devices/platform/i2c/i_range
/ # cat /sys/devices/platform/i2c/io_value
Rd Offset:0x20, Length:0x3
20: 0x11 0x22 0x33 0x44 0x55 0x66 0x77 0x88...
```

Ex3: Follow up Ex1. Now write value 0x1F to offset 0x40.

```
/ # echo 0x40 0x1F > /sys/devices/platform/i2c/io_value
/ #
```

Product: AST15XX
Date: 9/12/2017
FW Rev:

Doc Type:

Misc.

Reboot Device

Syntax

reboot

Related astparam

- reset_ch_on_boot

Version History

- FW >= A5.0.0

Examples

Ex1: Reboot system

```
/ # reboot  
/ #
```

Turn on/off debug console's debug message

Syntax

log.sh [**off**|**on**]

Parameters

- on: Turn on debug message
- off: Turn off debug message. Default behavior.

Related astparam

- en_log: Configure FW boot up default.

Version History

- FW >= A5.0.0

Examples

Ex1: Turn off debug message

```
/# log.sh  
/#
```

Ex2: Turn on debug message

```
/# log.sh on  
/#
```

Migrate From Legacy Console APIs

This section list out Console APIs difference between A7.x.x FW and older FW (A6.x.x, A5.x.x...). We call old FW Console APIs 'legacy' commands.

Developers must read this section carefully for migrating your control SW to support A7.x.x and newer FW.

New APIs introduced in A7.x.x FW majorly focus on:

- Support 'free routing' feature. For example client can connect to host#0001's video and host#0002's audio at the same time.
- Better support for large matrix installation. For example 1000x1000.
- Provide simplified new APIs while maintaining legacy APIs and style.

Most of changes are related to above field.

Ethernet Switch Configuration

Due to the introduction of 'free routing', Ethernet switch's IGMP snooping loading becomes heavier when devices switching multicast channel. Some low performance Ethernet switch may not performs well under this situation. So, user may see 'video jitter', 'video lost' or 'slow channel switching' problems during channel switching.

FW >= A7.x.x implements some tricks to improve the reliability of switch's IGMP snooping function. To avoid above problems, please configure Ethernet switch as:

- **[Mandatory] Enable IGMP snooping for IGMP V2.**
- **[Optional] Enable 'IGMP querier'.** (Different from FW <= A6.x.x)
 - [FW>=A7.0.0] AST15XX FW don't need 'IGMP querier'. It is recommended to disable it. However enable 'IGMP querier' should do no harm.
- **[Optional] Enable 'IGMP immediate/fast leave'**
 - Enabling 'IGMP immediate leave' may improve channel switching speed.
 - NOTE: Enable immediate leave **only** on ports where just one device is connected to each port. If immediate leave is enabled on ports where multiple devices are connected to a port, some devices might be dropped inadvertently.

Incompatible

- A7.x.x FW is not compatible with any older FW including A6.x.x and A5.x.x. Please don't mix host and client with different versions of FW.
- Audio free routing: Commands described in "Documents/Audio over IP (I2S)/Independent Audio Channel" is now replaced by commands described in "[Link Management:Switch Channel \[Client Only\]](#)" and "[Link Management:Switch Channel \[Host Only\]](#)".
- A7.x.x FW doesn't support SoC backward compatibility mode. Which means setting astparam, soc_op_mode, to '1' or '2' may cause unexpected result.

Command Changes

[C] For client specific. [H] For host specific.

- [C] Legacy "client channel switching" command is still working. BUT following new astparam must be removed to avoid conflict.
 - ch_select_v, ch_select_u, ch_select_a, ch_select_r, ch_select_s, ch_select_p
- [H] Legacy "host channel switching" command is still working.
- [H] For backward compatible with legacy "host channel switching" command,
 - **host's hostname_id MUST be the same as host's channel setting, ch_select.** ⇒
hostname_id must be the same format as 'ch_select', 0000~9999. FW will use '0000' default value if invalid format detected. FW will automatically append '0' to short digit. For example, '16' will be convert to '0016'.
 - host's 'e_chg_hostname' event will automatically trigger channel setting change, 'e_reconnect::{ch_select}'.
 - 'e_reconnect::{ch_select}' will automatically change hostname to specified 'ch_select' value. BUT will not save astparam, 'hostname_id'.
 - If you want to break this legacy behavior and use different 'hostname_id' from 'ch_select', please set host astparam, hostnamebydipswitch, to 'n'.
- [H/C] 'node_query' command is an upgrade version of legacy 'node_list' command. 'node_list' command is still available but is not recommended for large installation (nodes over 100).

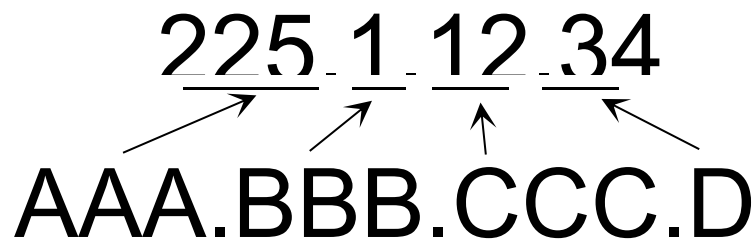
astparam Changes

astparam	Changes and Impact	Apply to Host or Client
multicast_ip multicast_ip_i2s	Changes: No more used. Multicast IP is now automatically mapped by FW. Check Appendix A. Channel to Multicast IP Mapping for details. Impact: Multicast IP can't be explicitly specified. Configure it has no effect, but won't impact FW operation. FW auto layout multicast IPs.	H/C
ch_select	Changes: Now, only serves as default value of other ch_select_x. This is a change for supporting 'free routing' feature. Impact: No. If you ever create 'ch_select_x' astparam by using new 'e_reconnect::{ch_select}' command, just remove those 'ch_select_x' astparam. Otherwise,	C

	'ch_select_x''s setting will be taken.	
ch_select_i2s	<p>Changes: astparam 'ch_select_i2s' is now 'ch_select_a'. 'ch_select_a' uses 'ch_select_i2s' as its default value.</p> <p>Impact: No. Use 'ch_select_a' is recommended.</p>	C
ch_select_soip2	<p>Changes: astparam 'ch_select_soip2' is now 'ch_select_s'. 'ch_select_s' uses 'ch_select_soip2' as its default value.</p> <p>Impact: No. Use 'ch_select_s' is recommended.</p>	C
multicast_ip_prefix	<p>Changes: astparam 'multicast_ip_prefix' has different format. New FW will automatically ignore invalid setting and use default value '225' instead.</p> <p>Impact: Legacy/invalid format setting will not be taken and default '225' will be used.</p>	H/C
hostname_id	<p>Changes: Behaves the same. New FW uses new 'ch_select' as the default value of 'hostname_id'. New FW actually don't need 'hostname_id' to be the same as client's 'ch_select' or 'ch_select_x'. However, in order to maintain backward compatibility, FW makes sure host's 'hostname_id' will always be the same as 'ch_select'. And trigger channel switching command if necessary. For new APIs, 'hostname_id' is not important anymore and will be the same as 'ch_select' value. If user want to use new APIs and specify a 'hostname_id' different from 'ch_select', please set astparam, hostnamebydipswitch, to 'n'.</p> <p>Impact: No. If the value of 'hostname_id' is not valid (0000~9999 format), then default '0000' value will be used. FW will automatically append '0' to short digit. For example, '16' will be convert to '0016'.</p>	H

Appendix A. Channel to Multicast IP Map

Different channel need different multicast IP. Also, for supporting free routing, each service has its own multicast IP. So, we map a multicast IP into following 4 parts:



- **AAA**: Fixed value. Valid range from 224 to 239. Use 225 to 238 is recommended. Default value is 225. Use astparam, **multicast_ip_prefix**, to change it.
- **BBB**: Maps to services. Different service has different value and depends on astparam **'free_routing'** and **'mcip_srv_map'** setting. Check 'BBB' list table.
- **CCC**: Maps to higher 2 digit of channel number. Ex: if channel number is '1234', then CCC will be '12'. Prepending '0' will be removed. Ex: channel number is '0102', then CCC will be '1'. In decimal format.
- **DDD**: Maps to lower 2 digit of channel number. Ex: if channel number is '1234', then DDD will be '34'. Prepending '0' will be removed. Ex: channel number is '0102', then DDD will be '2'. In decimal format.

'BBB' list table:

Function	Multicast IP "BBB"		Notes
	free_routing == y mcip_srv_map == 2:3:4:5:6:7	free_routing == n mcip_srv_map == 2:2:2:2:2:2	
Link Manager	225.1.0.0	225.1.0.0	node_list / name_service
	225.1.0.1	225.1.0.1	node_query / node_response
Video over IP Host encode	225.2.xxx.xxx	225.2.xxx.xxx	
Audio over IP Host encode	225.3.xxx.xxx	225.2.xxx.xxx	
IR over IP Host encode	225.4.xxx.xxx	225.2.xxx.xxx	1 Tx to N Rx
Serial over IP	225.5.xxx.xxx	225.2.xxx.xxx	

Product: AST15XX
Date: 9/12/2017
FW Rev:

Doc Type:

GPIO over IP	225.6.xxx.xxx	225.2.xxx.xxx	
USB over IP/KMoIP	225.7.xxx.xxx	225.2.xxx.xxx	

'**CCC.DDD**' maps from Channel 0000 to 9999:

Ch **1234**: 225.BBB.12.34

Ch **0000**: 225.BBB.0.0

Ch **0001**: 225.BBB.0.1

Ch **0002**: 225.BBB.0.2

...

Ch **0009**: 225.BBB.0.9

Ch **0010**: 225.BBB.0.10

...

Ch **0098**: 225.BBB.0.98

Ch **0099**: 225.BBB.0.99

Ch **0100**: 225.BBB.1.0

Ch **0101**: 225.BBB.1.1

...

Ch **0254**: 225.BBB.2.54

Ch **0255**: 225.BBB.2.55

Ch **0256**: 225.BBB.2.56

...

Ch **5000**: 225.BBB.50.0

...

Ch **9999**: 225.BBB.99.99

Channel to Multicast IP Mapping Examples: (**free_routing == y**)

Channel	Function	Multicast IP
0000	Video over IP	225.2.0.0
	Audio over IP	225.3.0.0
	IR over IP	225.4.0.0
	Serial over IP	225.5.0.0
	GPIO over IP	225.6.0.0

Product: AST15XX
Date: 9/12/2017
FW Rev:

Doc Type:

	USBoIP/KMoIP	225.7.0.0
0001	Video over IP	225.2.0.1
	Audio over IP	225.3.0.1
	IR over IP	225.4.0.1
	Serial over IP	225.5.0.1
	GPIO over IP	225.6.0.1
	USBoIP/KMoIP	225.7.0.1
0099	Video over IP	225.2.0.99
	Audio over IP	225.3.0.99
	IR over IP	225.4.0.99
	Serial over IP	225.5.0.99
	GPIO over IP	225.6.0.99
	USBoIP/KMoIP	225.7.0.99
0100	Video over IP	225.2.1.0
	Audio over IP	225.3.1.0
	IR over IP	225.4.1.0
	Serial over IP	225.5.1.0
	GPIO over IP	225.6.1.0
	USBoIP/KMoIP	225.7.1.0
1234	Video over IP	225.2.12.34
	Audio over IP	225.3.12.34

	IR over IP	225.4.12.34
	Serial over IP	225.5.12.34
	GPIO over IP	225.6.12.34
	USBoIP/KMoIP	225.7.12.34
9999	Video over IP	225.2.99.99
	Audio over IP	225.3.99.99
	IR over IP	225.4.99.99
	Serial over IP	225.5.99.99
	GPIO over IP	225.6.99.99
	USBoIP/KMoIP	225.7.99.99

Channel to Multicast IP Mapping Examples: (**free_routing == n**)

Channel	Function	Multicast IP
0000	Video over IP Audio over IP IR over IP Serial over IP GPIO over IP USBoIP/KMoIP	225.2.0.0
0001	Video over IP Audio over IP IR over IP Serial over IP GPIO over IP USBoIP/KMoIP	225.2.0.1
0099	Video over IP Audio over IP IR over IP Serial over IP GPIO over IP	225.2.0.99

Product: AST15XX
Date: 9/12/2017
FW Rev:

Doc Type:

	USB/IP/KMoIP	
0100	Video over IP Audio over IP IR over IP Serial over IP GPIO over IP USB/IP/KMoIP	225.2.1.0
1234	Video over IP Audio over IP IR over IP Serial over IP GPIO over IP USB/IP/KMoIP	225.2.12.34
9999	Video over IP Audio over IP IR over IP Serial over IP GPIO over IP USB/IP/KMoIP	225.2.99.99

Appendix B. IP Port Usage List

Category	Type	Host -> Client Host <- Client	Note
Heartbeat/Unicast/UDP	video	X → 59002 59002 ← X	
	audio	X → 59003 59002 ← X	
	ir	X → 59004 59004 ← X	
	serial	X → 59005 59005 ← X	
	gpio	X → 59006 59006 ← X	
	usb	X → 59007 59007 ← X	
Heartbeat/Multicast/UDP	video 225.2.xxx.xxx	X → 59002 59002 ← X	
	audio 225.3.xxx.xxx	X → 59003 59003 ← X	
	ir 225.4.xxx.xxx	X → 59004 59004 ← X	
	serial 225.5.xxx.xxx	X → 59005 59005 ← X	
	gpio 225.6.xxx.xxx	X → 59006	

Product: AST15XX
Date: 9/12/2017
FW Rev:

Doc Type:

		59006 ← X	
	usb 225.7.xxx.xxx	X → 59007 59007 ← X	
node_query → node_response	225.1.0.1	X→ 59101	
node_response → node_query		X→ 59100	
VideolP Host encode	225.2.xxx.xxx	X → 59200 59201 ← X	
AudiolP Host encode	225.3.xxx.xxx	X → 59300 59300 ← X	
IRolP Host encode	225.4.xxx.xxx	X → 59400 59400 ← X	
SolP Type 1	TCP	X → 6752	backward compatible
SolP Type 2	TCP	6752 ← X	backward compatible
SolP Type 3	225.5.xxx.xxx	X → 59500 59500 ← X	
GPIO over IP	225.6.xxx.xxx	X → 59600 59600 ← X	Re-use heartbeat's msg_channel, so, no extra port needed.
USBoIP	TCP	59700 ← X	USBoIP always use TCP
KMoIP	225.7.xxx.xxx	X → 59702 59703 ← X	KMoIP always use multicast UDP. In FW after A7.x.x, KMoIP uses unicast UDP for unicast mode.

Product: AST15XX
Date: 9/12/2017
FW Rev:

Doc Type:

Appendix C. Link Manager Event List

Host

Link Control

Event	Description
e_reconnect	Switch channel.
e_stop_link	Stop link.
e_chg_hostname	Change hostname.

Button Events

Event	Description
e_button_link	Short press button 1.
e_button_link_1	Long press button 1.
e_button_pairing	Short press button 2.
e_button_pairing_1	Long press button 2.
e_btn_toggle_link	Toggle between link On and Off.
e_btn_toggle_video_profile	Toggle between Video/Graphic mode.
e_btn_toggle_video_profile_video	Set to Video mode.
e_btn_toggle_video_profile_graphic	Set to Graphic mode.
e_btn_toggle_video_anti_dither e_btn_toggle_video_anti_dither_{mode}	Toggle video Anti-Dither setting.
e_btn_toggle_snoop e_btn_toggle_snoop_on e_btn_toggle_snoop_off	Host video loopback On/Off.

Client

Link Control

Event	Description
e_reconnect	Switch channel.
e_stop_link	Stop link.
e_start_usb e_stop_usb	Manually Start/Stop USB/IP.
e_start_kmoip e_stop_kmoip	Manually Start/Stop KMoIP.
e_chg_hostname	Change hostname.

Button Events

Event	Description
e_button_link	Short press button 1.
e_button_link_1	Long press button 1.
e_button_pairing	Short press button 2.
e_button_pairing_1	Long press button 2.
e_btn_toggle_link	Toggle between link On and Off.
e_btn_toggle_video_profile	Toggle between Video/Graphic mode.
e_btn_toggle_video_profile _video	Set to Video mode.
e_btn_toggle_video_profile _graphic	Set to Graphic mode.
e_btn_toggle_video_anti_di ther e_btn_toggle_video_anti_di ther_{mode}	Toggle video Anti-Dither setting.

Product: AST15XX
Date: 9/12/2017
FW Rev:

Doc Type:

e_btn_toggle_snoop e_btn_toggle_snoop_on e_btn_toggle_snoop_off	Host video loopback On/Off.
e_btn_request_usb	Manually Start/Stop USB/IP.

OSD Control

Event	Description
e_osd_on_str	Show string OSD.

Appendix D. List of Imparam

Most of Imparam are not allowed to be written from user. Unless special noted, write to Imparam may cause unexpected result. User should use it only for querying link manager's internal flags.

Common

Imparam	Description
STATE	Link Manager State .
HOSTNAMEBYDIPSWITCH	Hostname control. Maps from astparam, hostnamebydipswitch.
HOSTNAME_PREFIX HOSTNAME_TX_MIDDLE HOSTNAME_RX_MIDDLE HOSTNAME_ID	Hostname .
IP_MODE	Maps from astparam, ip_mode.
MY_IP	IP address.
MY_MAC	Ethernet MAC address.
ETH_LINK_STATE	Ethernet PHY link status.
ETH_LINK_MODE	Ethernet PHY link mode. Only valid when 'ETH_LINK_STATE' is on.
ACCESS_ON	Maps from astparam, astaccess.
RESET_CH_ON_BOOT	Maps from astparam, reset_ch_on_boot.
FREE_ROUTING	Maps from astparam, free_routing.
MCIP_SRV_MAP	Maps from astparam, mcip_srv_map.

Host

Imparam	Description
CH_SELECT	Service channel selection. All services uses the same channel, but maps to different multicast IP address.
V_QUALITY_MODE	Current video quality mode .
V_BCD_THRESHOLD	Current video Anti-Dither mode .

V_LOOPBACK_ENABLED	Current video loopback port on/off state .
LOOPBACK_DEFAULT_ON	Boot up default status. From astparam, loopback_default_on.
USB_CLIENT_IP	Attached USBoIP client IP address .

Client

Imparam	Description
CH_SELECT CH_SELECT_V CH_SELECT_U CH_SELECT_A CH_SELECT_R CH_SELECT_S CH_SELECT_P	Channel selection. In client, each service can choose it own channel to connect to. It is so called 'free routing'. CH_SELECT will be used if not specified.
ACCESS_ON_V ACCESS_ON_U ACCESS_ON_A ACCESS_ON_R ACCESS_ON_S ACCESS_ON_P	In client, each service can be start/stop independently. Each ACCESS_ON_X represents the start/stop state of corresponding service.
U_USBIP_STATE	Current USBoIP status . Get by ulmparam.
U_KMOIP_STATE	Current KMoIP status . Get by ulmparam.

Appendix E. Sample Code

You can find some useful reference source code under Documents/Console APIs/SampleCode/.

Folder/File Name	Description
node_query_windows	'node_query' source code and Windows version executable file.
node_list_windows	'node_list' source code and Windows version executable file. Use 'node_query' is recommended in FW >= A7.0.0.
serial_ip*.zip	'Serial over IP Type 1' source code.
serial_ip2*.zip	'Serial over IP Type 2' source code.