

HDTV Supply IP AV
Sender IP Command
Document V0. 5

Contents

| | |
|--|-----------|
| 1. FORWARD..... | 1 |
| 1.1 PURPOSE..... | 1 |
| 1.2 APPLICATION SCOPE..... | 1 |
| 1.3 APPLICATION PEOPLE..... | 1 |
| 2. IP AV SENDER IP COMMAND DESCRIPTION..... | 2 |
| 2.1 HDTV Supply IP AV SENDER IP CONTROL COMMAND INTRODUCTION..... | 2 |
| 2.2 HDTV Supply IP AV SENDER TCP COMMAND SCHEMATIC DIAGRAM I INTRODUCTION:..... | 2 |
| 2.3 HDTV Supply IP AV SENDER QUERY COMMAND SCHEMATIC DIAGRAM INSTROCTIONS:..... | 3 |
| 2.4 IP AV SENDER DEFINITION COMMAND BYTE STREAM FORMAT..... | 4 |
| 2.5 COMMAND PROTOTYPE..... | 4 |
| 2.6 COMMAND OVER IP FORMAT..... | 4 |
| 3. IP AV SENDER TX/RX SHARED TCP COMMAND INSTRUCTIONS..... | 5 |
| 3.1 TX AND RX SHARED COMMAND ID LIST..... | 5 |
| 3.2 ABNORMAL REPNSE COMMAND..... | 5 |
| 3.3 SET MULTICAST GROUP ID COMMAND (COMMAND_SET_GROUP_ID & COMMAND_SET_GROUP_ID_RET)..... | 5 |
| 3.4 GET GROUP ID COMMAND (COMMAND_GET_GROUP_ID & COMMAND_GET_GROUP_ID_RET)..... | 6 |
| 3.5 GET MAC ADDRESS COMMAND (COMMAND_GET_MAC_ADDRESS & COMMAND_GET_MAC_ADDRESS_RET)..... | 7 |
| 3.6 SET DEVICE NAME COMMAND (COMMAND_SET_DEVICE_NAME & COMMAND_SET_DEVICE_NAME_RET)..... | 8 |
| 3.7 GET DEVICE NAME COMMAND (COMMAND_GET_DEVICE_NAME & COMMAND_GET_DEVICE_NAME_RET)..... | 9 |
| 4. IP AV SENDER TX/RX QUERY UDP COMMAND INSTRUCTION..... | 10 |
| 4.1 UDP QUERY COMMAND ID LIST..... | 10 |
| 4.2 QUERY DEVICE INFORMATION COMMAND (COMMAND_DEV_INFO_QUERY & COMMAND_DEV_INFO_QUERY_RET). .. | 11 |

1. Forward

1.1 Purpose
HDTV Supply IP AV Sender IP Command instructions.

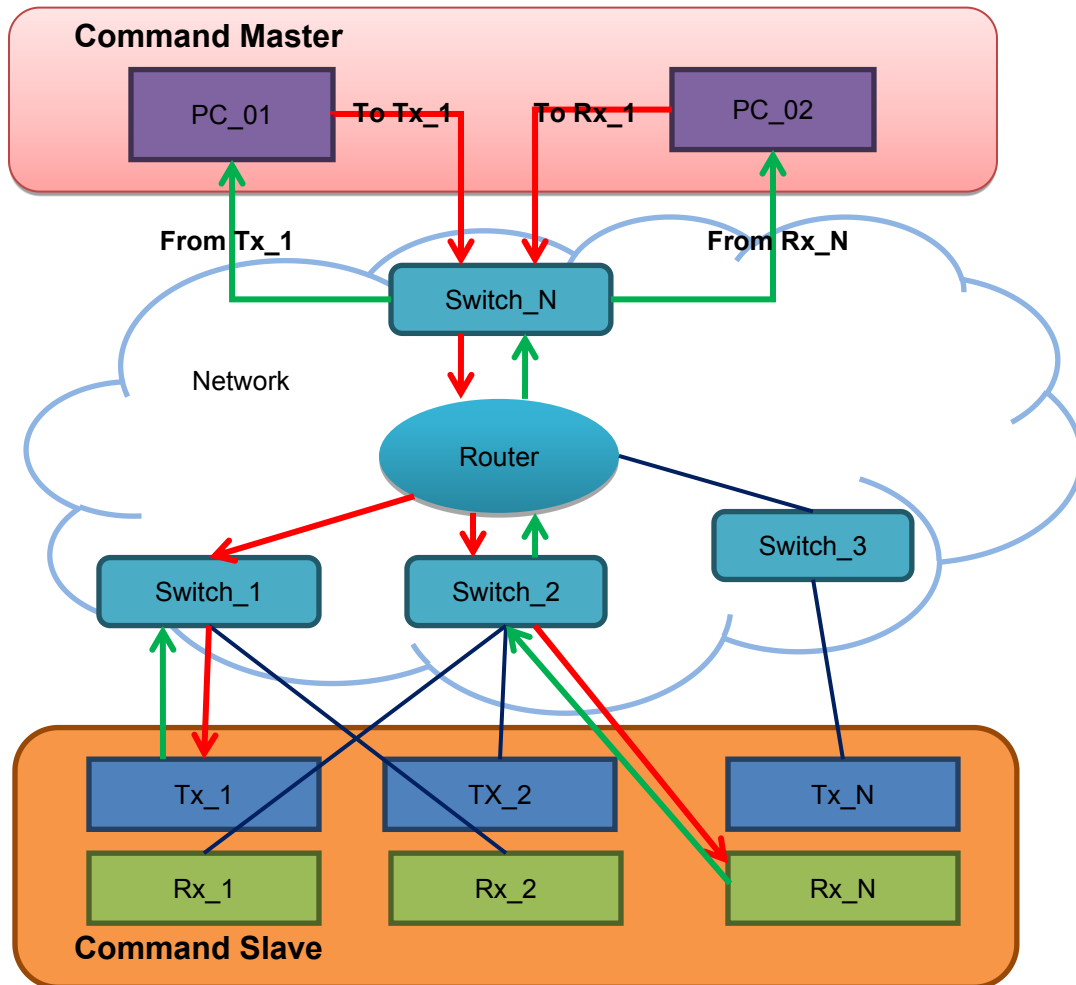
1.2 Application scop
HDTV Supply IP AV Sender related Tx/Rx products.

2. IP AV Sender IP Command description:

2.1 HDTV Supply IP AV Sender IP Control Command introduction

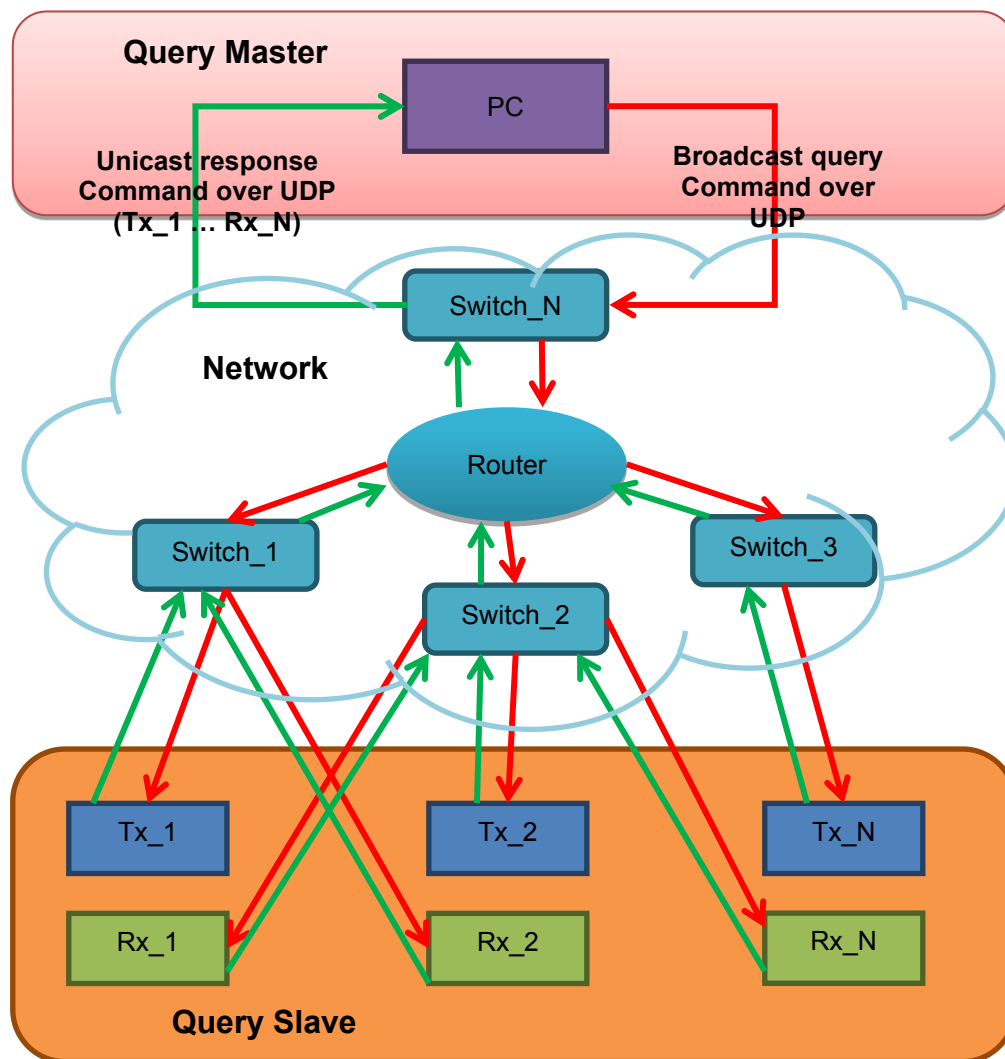
- Get/Set Command via TCP/IP, to set Tx/Rx Device function by network, get the related information of present set up status.
- Device Query Command via UDP/IP, to collect related device information on network.

2.2 HDTV Supply IP AV Sender TCP Command schematic diagram instructions:



- ALL Control Command(Get/Set) ONLY send by Command Master, just like above schematic diagram, Command Master send Request Command by TCP/IP to specific Device, and also received and replied Response Command by the Device. That is won't happen Slave send Command to Master automatically.

2.3 HDTV Supply IP AV Sender Query Command schematic diagram instructions:



- Query Command like above schematic diagram, Query Master according to setted Query Rule Broadcast Query Request Command over UDP/IP to all Devices on same network. If the received Device conformed Query Rule, then it will send Unicast Response Command over UDP/IP to Query Master.

2.4 IP AV Sender definition of Command Byte Stream Format

| | | | | | |
|-----------------------|------------------------|--------------------------|--------------------------------|------------------------------------|--|
| Sync Word (1 Byte) | Command Id (2 Byte) | Payload Size (2 Byte) | Header Checksum (1 Byte) | Command Payload (1...N Byte) | Command Payload Checksum (1 Byte) |
|-----------------------|------------------------|--------------------------|--------------------------------|------------------------------------|--|

- Sync Word: to identify Command's start code, Length: 1 Byte, content value: fixed 0x74.
- Command Id: to identify transmitted Command type, Length: 2 Byte, usually receiver's Command Id is the transmitter's Command Id +1.
- Payload Size: For record subsequent transmission length of Command Payload, Length is the actual transmission Command Payload's length + Command Payload Checksum's length. That is to say content value is N+1.
- Header Checksum: To make sure Command Id and Payload Size is normal value. Length is 1 Byte, Content value is Byte[1] + Byte[2] + Byte[3] + Byte[4].
- Command Payload: Realted Command's Payload. Length is uncertain , decided by each Command.
- Command Payload Checksum: To make sure Command Payload content is correct, Length is 1 Byte, For example, Payload has N Byte in total, the content value is Payload[0] + Payload[1] +.....Payload[N-1].

2.5 Command Prototype

- Through analysed Command Byte Stream, basically Command is consist of Command Id and Command Payload. The following structure is the basic prototype for each Command solution.

| Syntax | Byte Size |
|----------------------|-----------|
| command_prototype(){ | |
| command_id | 2 |
| command_payload | 1...N |
| } | |

- command_id: To recognize command's Identification code.
- command_payload: the contents of this command.

2.6 Command Over IP Format

| | | | |
|----------------------------------|-------------------------|-----------------------------------|----------------------------------|
| IP Command Sync Word (8 Byte) | Sender's IP (4 Byte) | Sender's Receive Port (2 Byte) | IP AV Sender Command (N Byte) |
|----------------------------------|-------------------------|-----------------------------------|----------------------------------|

- IP Command Sync Word: Used to recognize the start code of IP AV Sender Command Over IP, Length : 8 Byte, Content value: "IPTV_CMD".
- Sender's IP: Command source IP.
- Sender's Receive Port: Source IP's receiving Command Port Number.
- IPTV Command: Like the defined Command Format on 2.4.

3. IP AV Sender Tx/Rx common instructions on TCP Command

3.1 Command Id List shared Tx and Rx instructions

| Request Command | Mapped Command Id | Response Command | Mapped Command Id |
|-------------------------|-------------------|---------------------------------|-------------------|
| COMMAND_SET_GROUP_ID | 0x0005 | COMMAND_SET_GROUP_ID_RET | 0x0006 |
| COMMAND_GET_GROUP_ID | 0x0013 | COMMAND_GET_GROUP_ID_RET | 0x0014 |
| COMMAND_GET_MAC_ADDRESS | 0x001B | COMMAND_GET_MAC_ADDRESS_RE T | 0x001C |
| COMMAND_SET_DEVICE_NAME | 0x001D | COMMAND_SET_DEVICE_NAME_RE T | 0x001E |
| COMMAND_GET_DEVICE_NAME | 0x001F | COMMAND_GET_DEVICE_NAME_RE T | 0x0020 |
| | | TIMEOUT_COMMAND_ID | 0xFFFD |
| | | UNSUPPORT_COMMAND_ID | 0xFFFE |
| | | CORRUPTED_COMMAND_ID | 0xFFFF |

3.2 Abnormal Reponse Command

- Above table(on 3.1) has three special Command Id, each represents each abnormal results.

- Abnormal Response Command:

| Syntax | Byte Size |
|------------------------------|-----------|
| abnormal_response_command(){ | |
| command_id | 2 |
| null_payload | 1 |
| } | |

- command_id:
When content value is 0xFFFD represents Master did not receive the send back message via Slave.
When content value is 0xFFFE represents Slave can't recognize Request Command Id.
When content value is 0xFFFF represents Slave compared failure on Master's Command message of checksum.
- null_payload: content value is 0x0, in order to accord with basic command prototype's definition.

3.3 Set Multicast Group Id Command (COMMAND_SET_GROUP_ID & COMMAND_SET_GROUP_ID_RET)

- Function: to set the transmit Multicast Group Id (0~1023) on TX, to set the wanted received Multicast Group Id on RX.
- Request Command(COMMAND_SET_GROUP_ID) format:

| Syntax | Byte Size |
|-----------------------------------|-----------|
| set_multicast_group_id_command(){ | |
| command_id | 2 |
| group_id | 2 |
| } | |

- command_id: content value is 0x0005
- group_id: transmitting (Tx) or receiving (Rx)'s Multicast Group Id, legal range is 0~1023.

- Response Command(COMMAND_SET_GROUP_ID_RET) format:

| Syntax | Byte Size |
|---------------------------------------|-----------|
| set_multicast_group_id_command_ret(){ | |
| command_id | 2 |
| set_result | 1 |
| } | |

- command_id: content value is 0x0006.
- set_result: 0 –Set success, 0xFF – HDMI Sink non-existent, 0xFE – Non connection on TX Device.

3.4 Get Group ID Command (COMMAND_GET_GROUP_ID & COMMAND_GET_GROUP_ID_RET)

- Function: Get current Device setted Multicast Group ID.

- Request Command(COMMAND_GET_GROUP_ID) format:

| Syntax | Byte Size |
|-------------------------|-----------|
| get_group_id_command(){ | |
| command_id | 2 |
| null_payload | 1 |
| } | |

- command_id: content value is 0x0013
- null_payload: content value is 0x0, in order to accord with basic command prototype definition.

- Response Command(COMMAND_GET_GROUP_ID_RET) format:

| Syntax | Byte Size |
|-----------------------------|-----------|
| get_group_id_command_ret(){ | |
| command_id | 2 |
| get_result | 1 |
| group_id | 2 |
| } | |

- command_id: content value is 0x0014
- get_result: 0 –Read successfully, -1 –Read failed.
- group_id: Current Device used Group id.

3.5 Get MAC Address Command (COMMAND_GET_MAC_ADDRESS & COMMAND_GET_MAC_ADDRESS_RET)

- Function: Get current Device's MAC Address.

- Request Command(COMMAND_GET_MAC_ADDRESS) format:

| Syntax | Byte Size |
|----------------------------|-----------|
| get_mac_address_command(){ | |
| command_id | 2 |
| null_payload | 1 |
| } | |

- command_id: content value is 0x001B
- null_payload: content value is 0x0, in order to accord with basic command prototype definition.

- Response Command(COMMAND_GET_MAC_ADDRESS_RET) format:

| Syntax | Byte Size |
|--------------------------------|-----------|
| get_mac_address_command_ret(){ | |
| command_id | 2 |
| get_result | 1 |
| mac_address | 6 |
| } | |

- command_id: Content value is 0x001C
- get_result: 0 –Read successfully, -1 –Read failed..
- mac_address: Current Device's MAC Address.

3.6 Set Device Name Command (COMMAND_SET_DEVICE_NAME & COMMAND_SET_DEVICE_NAME_RET)

- Function: set current Device identification name.

- Request Command(COMMAND_SET_DEVICE_NAME) format:

| Syntax | Byte Size |
|----------------------------|-----------|
| set_device_name_command(){ | |
| command_id | 2 |
| device_name | 32 |
| } | |

- command_id: content value is 0x001D
- device_name: Pre set Device identification name, longest length is 32 Bytes.

- Response Command(COMMAND_SET_DEVICE_NAME_RET) format:

| Syntax | Byte Size |
|--------------------------------|-----------|
| set_device_name_command_ret(){ | |
| command_id | 2 |
| set_result | 1 |
| } | |

- command_id: Content value is 0x001E
- set_result: 0 –Set success, -1 –Set fail.

3.7 Get Device Name Command (COMMAND_GET_DEVICE_NAME & COMMAND_GET_DEVICE_NAME_RET)

- Function: Get current Device identification name.

- Request Command(COMMAND_GET_DEVICE_NAME) format:

| Syntax | Byte Size |
|----------------------------|-----------|
| get_device_name_command(){ | |
| command_id | 2 |
| null_payload | 1 |
| } | |

- command_id: content value is 0x001F
- null_payload: content value is 0x0, in order to accord with basic command prototype definition.

- Response Command(COMMAND_GET_DEVICE_NAME_RET) format :

| Syntax | Byte Size |
|--------------------------------|-----------|
| get_device_name_command_ret(){ | |
| command_id | 2 |
| device_name | 32 |
| } | |

- command_id: content value is 0x0020
- device_name: The recognition name of the Device, longest Length is 32 Bytes.

4. IP AV Sender Tx/Rx Query UDP Command instructions:

4.1 UDP Query Command Id List

| Request Command | Mapped Command Id | Response Command | Mapped Command Id |
|------------------------|-------------------|----------------------------|-------------------|
| COMMAND_DEV_INFO_QUERY | 0x00FE | COMMAND_DEV_INFO_QUERY_RET | 0x00FF |

4.2 Query Device Information Command (COMMAND_DEV_INFO_QUERY & COMMAND_DEV_INFO_QUERY_RET)

- Function: Through the setted search conditions, to obtain the related information on current network IP AV Sender in accordanced with the device.
- Request Command(COMMAND_DEV_INFO_QUERY) format:

| Syntax | Byte Size |
|-------------------------|-----------|
| query_device_command(){ | |
| command_id | 2 |
| check_device_type | 1 |
| device_type_rule | 1 |
| check_group_id | 1 |
| group_id_rule | 1 |
| check_device_ip | 1 |
| device_ip_rule | 4 |
| } | |

- command_id: content value is 0x00FE.
- check_device_type: content value is 0 – not open Device Type distinguish.
1 – open distinguish Device Type rules.
- device_type_rule: When check_device_type is 1. Content value 0 – Rx Device, 1 – Tx Device.
- check_group_id: Content value 0 – not open Group ID distinguish. 1 –open distinguish Group ID rules.
- group_id_rule: When check_group_id is 1. Content value is Pre search Group ID.
- check_device_ip: content value is 0 – not open particular IP distinguish. 1 –open distinguish particular IP rules.
- device_ip_rule: When check_device_ip is 1. Content value is Pre search Device IP.

- Response Command(COMMAND_DEV_INFO_QUERY_RET) format:

| Syntax | Byte Size |
|-----------------------------|-----------|
| query_device_command_ret(){ | |
| command_id | 2 |
| device_name | 32 |
| device_ip | 4 |
| device_tcp_cmd_receive_port | 2 |
| device_group_id | 2 |
| device_type | 1 |
| device_state | 1 |
| device_stream_type | 1 |
| device_streaming_mode | 1 |
| } | |

- command_id: content value is 0x00FF.
- device_name: Device identification name.
- device_ip: Device used IP.
- device_tcp_cmd_receive_port: Device TCP Command receiving Port Number.
- device_group_id: Current Device's Group ID.
- device_type: Content value is 0 – Rx Device, 1 – Tx Device.
- device_state: Content value is 0 – Idle State, 1 – Running State.
- device_stream_type: Content value is 0 – AV Over UDP, 1 – AV Over RTP.
- device_streaming_mode: Content value is 1 – Streaming Through Multicast.