Channel in a box

Device Control Protocol

Version 1.9

* **Document Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Changes** | **Writer** |
| 1.0 | 02-10-2017 | First document for general distribution | Sangbong, Lee |
| 1.1 | 02-28-2017 | Add MCS(Master Switch) protocol | Sangbong, Lee |
| 1.2 | 06-27-2017 | Add get device number function  Modify MCS crosspoint take function  Add MCS key functions | Sangbong, Lee |
| 1.3 | 07-08-2019 | Add "command" to the beginning of every response packet. | Sangbong, Lee |
| 1.4 | 07-22-2019 | * Add MCS functions “get\_status\_key” assign key to bus response | Sangbong, Lee |
| 1.5 | 07-23-2019 | * Add MCS functions “key\_assign” assign key to bus “get\_key\_assign” get assigned key to bus * Modify MCS function “get\_crosspoint\_take” bus parameter delete “get\_status\_key” -> “get\_key\_take” * Add all play device function “get\_rate” get play rate | Sangbong, Lee |
| 1.6 | 07-24-2019 | * Modify MCS functions Delete “key\_assign, “get\_key\_assign” functions   “key\_take” set multi bus and key turn on/off  “get\_key\_take” get multi bus and key on/off information | Sangbong, Lee |
| 1.7 | 10-02-2019 | - Integrate the CG control protocol. | Sangbong, Lee |
| 1.8 | 10-07-2019 | - Set\_Page command is delete.  - Added object\_id to Set\_Data & Set\_Rate command | Sangbong, Lee |
| 1.9 | 01-17-2020 | - Organize document | Sangbong, Lee |
| 2.0 | 01-27-2020 | * Add video encoder/decoder functions “get\_clip\_exit” get clip exist state “get\_clip\_duration” get clip duration timecode | Sangbong, Lee |
|  |  |  |  |
|  |  |  |  |

1. **Introduction**

Defines a protocol for a device to be used in a channel in a box (CIB).

CIB Device consists of Video Encoder / Decoder, CG (Cut & Page), Sequence CG, Scroll CG, Logo, Closed Caption & Sub Title, MCS and describes control protocol for each device in this document.

1. **Interface System Overview**

* The protocol is based on JSON
* Confirming to UDP & RS-422
* Data is transmitted asynchronously, with data exchange between devices.
* Standard transmission UDP port on see the below :

Video Encoder/Decoder : 1000 ~ 1999

CG (Cut & Page, Sequence, Scroll, Logo, Closed Caption & Sub Title : 2000 ~ 2999

Master Control Switch : 3000~3999

1. **JSON Protocol Format**

* Command Default  
  {  
   “command” : “”   
  }
* Response Default  
  If success  
  {  
   “command” : “”  
   “response” : “success”  
  }  
    
  If fail  
  {  
   “command” : “”  
   “response” : “fail”,  
   “errorcode” : 0  
  }
* Depending on the type of command, the convention for JSON objects is different.
* Response also define additional JSON objects depending on the type of command.
* The command is classified according to the mode of operation. See the below :  
  System Control  
  Immediate Control  
  Preset/Select Control  
  Sens Request
* It should be noted that the response of all commands except “Sense Request” is not a result of the command but a command that is normally received.

1. **Command**
2. **Command Table (Video Encoder/Decoder)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Command** | | **Response** | |
| **JSON Object** | **Description** | **JSON Object** | **Description** |
| **System Control** | | | |
| open | Device open | Success | Command success |
| close | Device close | Success | Command success |
| **Immediate Control** | | | |
| stop | Stop | Success | Command success |
| play | Play or Record | Success | Command success |
| pause | Pause | Success | Command success |
| set\_rate | Set play rate | Success | Command success |
| set\_timecode | Go to timecode position | Success | Command success |
| **Preset/Select Control** | | | |
| play\_cue | Play cue | Success | Command success |
| play\_cue\_data | Play cue & Start & Duration | Success | Command success |
| record\_cue | Record cue | Success | Command success |
| record\_cue\_data | Record cue & Duration | Success | Command success |
| **Sense Request** | | | |
| get\_device\_no | Get device number | Success & Data | Command success & device number data |
| get\_status | Get device status | Success & Data | Command success & status data |
| get\_rate | Get play rate | Success & Data | Command success & play rate data |
| get\_timecode | Get timecode | Success & Data | Command success & timecode data |
| get\_clip\_exist | Get clip exist | Success & Data | Command success & clip exist data |
| get\_clip\_size | Get clip duration timecode | Success & Data | Command success & clip duration timecode data |

1. **Command Table (CG)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Command** | | **Response** | |
| **JSON Object** | **Description** | **JSON Object** | **Description** |
| **System Control** | | | |
| open | Device open | Success | Command success |
| close | Device close | Success | Command success |
| **Immediate Control** | | | |
| stop | Stop | Success | Command success |
| play | Play | Success | Command success |
| pause | Pause | Success | Command success |
| set\_page | Go to CG cut page | Success | Command success |
| set\_data | Set template or scroll data | Success | Command success |
| set\_rate | Set play rate | Success | Command success |
| set\_timecode | Go to timecode position | Success | Command success |
| **Preset/Select Control** | | | |
| play\_cue | Play cue | Success | Command success |
| play\_cue\_data | Play Cue & Start & Duration | Success | Command success |
| **Sense Request** | | | |
| get\_device\_no | Get device number | Success & Data | Command success & device number data |
| get\_status | Get device status | Success & Data | Command success & status data |
| get\_rate | Get play rate | Success & Data | Command success & play rate data |
| get\_timecode | Get timecode | Success & Data | Command success & timecode data |

1. **Command Table (MCS – Master Control Switch)**

|  |  |  |  |
| --- | --- | --- | --- |
| **“Command” JSON Object** | | **“Response” JSON Object** | |
| **JSON Value** | **Description** | **JSON Value** | **Description** |
| **System Control** | | | |
| open | Device open | Success | Command success |
| close | Device close | Success | Command success |
| **Immediate Control** | | | |
| next\_transition | Next transition | Success | Command success |
| start\_transition | Start transition | Success | Command success |
| crosspoint\_take | Crosspoint take | Success | Command success |
| key\_take | Key take | Success | Command success |
| **Preset/Select Control** | | | |
| set\_transition\_type | Set transition type | Success | Command success |
| set\_transition\_rate | Set transition rate | Success | Command success |
| set\_enable\_key | Set enable key | Success | Command success |
| **Sense Request** | | | |
| get\_transition\_type | Get transition type | Success & Data | Command success & transition type data |
| get\_transition\_rate | Get transition rate | Success & Data | Command success & transition rate data |
| get\_crosspoint\_take | Get crosspoint take | Success & Data | Command success & crosspoint take data |
| get\_enable\_key | Get enable key | Success | Command success & key enable data. |
| get\_status\_key | Get key status | Success & Data | Command success & key status data |
| get\_status | Get device status | Success & Data | Command success & status data |

1. **Command Description**
2. **Video Encoder/Decoder**

**System Control**

* **open**Open to use device.  
  No other commands can be executed before this command is executed.  
    
  {   
   “command” : “open”  
  }
* **close**  
  Close the device.  
  No other commands can be executed after this command.  
    
  {   
   “command” : “close”  
  }

**Immediate Control**

* **stop**The [stop] command stops playback or recording.  
    
  {   
   “command” : “stop”  
  }
* **play**Start playback or recording.  
  If the decoder in PAUSE state, the [play] command start playback or recording at current position.
* Play Cue before you start playback.
* Record Cue before you start record.

{   
 “command” : “play”  
}

* **pause**The [pause] command pause playback or recording.  
  If the decoder in PLAY or PLAY CUE state, the [pause] command causes the currently playing ID to pause and display a still video frame.  
  If the encoder in RECORD or RECORD CUE state, the PAUSE command causes the currently recording ID to pause.  
    
  {   
   “command” : “pause”  
  }
* **set\_rate**Specifies the playback speed  
  If it is positive, it is forward.  
  If it is negative, it is rewind.  
  If it is 0, it is pause.  
  This command can be used to implement the Jog & Shuttle function.

|  |  |  |  |
| --- | --- | --- | --- |
| **JSON Object** | **JSON Value** | **Value Type** | **Description** |
| command | set\_rate | String |  |
| rate | 0 | Number (float) | 0 : pause  Greater then 0 : forward play  Less then 0 : rewind play |

{   
 “command” : “set\_rate”,  
 “rate” : 0  
}

* **set\_timecode**If playback is in progress, position the video in timecode.  
  If the time code is larger than the video duration, it moves to the back of the video.

|  |  |  |  |
| --- | --- | --- | --- |
| **JSON Object** | **JSON Value** | **Value Type** | **Description** |
| command | set\_timecode | String |  |
| timecode | 00:00:00:00 | String | Timecode (DF & NDF) |

{   
 “command” : “set\_timecode”,  
 “timecode” : “00:00:00:00”  
}

**Preset/Select Control**

* **play\_cue**The [play\_cue] command prepares for playback.  
  If the ID is not found, an error occurs and state returns STOP.  
  When the CUE is complete, the CUE-DONE flag is set in status.

|  |  |  |  |
| --- | --- | --- | --- |
| **JSON Object** | **JSON Value** | **Value Type** | **Description** |
| command | play\_cue | String |  |
| id | id | String | Video material id |

{   
 “command” : “play\_cue”,  
 “id” : “id”  
}

* **play\_cue\_data**Performs similar to [play\_cue].  
  In addition, you can specify playback start timecode and playback duration.

|  |  |  |  |
| --- | --- | --- | --- |
| **JSON Object** | **JSON Value** | **Value Type** | **Description** |
| command | play\_cue\_data | String |  |
| id | id | String | Video material id |
| start | 00:00:00:00 | String | Playback start timecode |
| duration | 00:59:59:29 | String | Playback duration timecode |

{   
 “command” : “play\_cue\_data”,  
 “id” : “id”,  
 “start” : “00:00:00:00”,  
 “duration” : “00:59:59:29”  
}

* **record\_cue**The [record\_cue] command prepares for record.  
  If there is an existing ID, it overwrites it, and immediately the ID has a length of 0.  
  When the CUE is complete, the CUE-DONE flag is set in status.

|  |  |  |  |
| --- | --- | --- | --- |
| **JSON Object** | **JSON Value** | **Value Type** | **Description** |
| command | record\_cue | String |  |
| id | id | String | Video material id |

{   
 “command” : “record\_cue”,  
 “id” : “id”  
}

* **record\_cue\_data**Performs similar to [record\_cue].  
  In addition, you can specify record duration.  
  When recording is done for the specified duration, it stops automatically.

|  |  |  |  |
| --- | --- | --- | --- |
| **JSON Object** | **JSON Value** | **Value Type** | **Description** |
| command | play\_cue\_data | String |  |
| id | id | String | Video material id |
| duration | 00:59:59:29 | String | Playback duration timecode |

{   
 “command” : “play\_cue\_data”,  
 “id” : “id”,  
 “duration” : “00:59:59:29”  
}

**Sense Request**

* **get\_device\_no**The [get\_device\_no] command get device number.  
    
  {   
   “command” : “get\_device\_no”  
  }  
    
  **Response**

|  |  |  |  |
| --- | --- | --- | --- |
| **JSON Object** | **JSON Value** | **Value Type** | **Description** |
| command | get\_device\_no | String |  |
| response | success | String | Success or Fail |
| no | 0 | Number | Get PCS the set device number. |

* **get\_status**The [get\_status] command get device status.  
    
  {   
   “command” : “get\_status”  
  }  
    
  **Response**

|  |  |  |  |
| --- | --- | --- | --- |
| **JSON Object** | **JSON Value** | **Value Type** | **Description** |
| command | get\_status | String |  |
| response | success | String | Success or Fail |
| * status idle cueing cue\_done play pause | true  false  false  false  false | Boolean  Boolean  Boolean  Boolean  Boolean | Not doing anything or stopped  Preparing play or record  Completed prepare play or record  Playing  Paused |

* **get\_rate**The [get\_rate] command get current play rate.  
    
  {   
   “command” : “get\_rate”  
  }  
    
  **Response**

|  |  |  |  |
| --- | --- | --- | --- |
| **JSON Object** | **JSON Value** | **Value Type** | **Description** |
| command | get\_timecode | String |  |
| response | success | String | Success or Fail |
| rate | 0 | Number (float) | 0 : pause  Greater then 0 : forward play  Less then 0 : rewind play |

* **get\_timecode**The [get\_timecode] command get current timecode.  
    
  {   
   “command” : “get\_timecode”  
  }  
    
  **Response**

|  |  |  |  |
| --- | --- | --- | --- |
| **JSON Object** | **JSON Value** | **Value Type** | **Description** |
| command | get\_timecode | String |  |
| response | success | String | Success or Fail |
| timecode | 00:00:00:00 | String |  |

* **get\_clip\_exist**The [get\_clip\_exist] command get clip exist state.  
    
  {   
   “command” : “get\_clip\_exist”  
   “id” : “id”  
  }  
    
  **Response**

|  |  |  |  |
| --- | --- | --- | --- |
| **JSON Object** | **JSON Value** | **Value Type** | **Description** |
| command | get\_clip\_exist | String |  |
| response | success | String | Success or Fail |
| exists | true | Boolean |  |

* **get\_clip\_size**The [get\_clip\_size] command get clip duration timecode.  
    
  {   
   “command” : “get\_clip\_size”  
   “id” : “id”  
  }  
    
  **Response**

|  |  |  |  |
| --- | --- | --- | --- |
| **JSON Object** | **JSON Value** | **Value Type** | **Description** |
| command | get\_clip\_size | String |  |
| response | success | String | Success or Fail |
| duration | 00:00:00:00 | String |  |

1. **CG**

**System Control**

* **open**Open to use device.  
  No other commands can be executed before this command is executed.  
    
  {   
   “command” : “open”  
  }
* **close**  
  Close the device.  
  No other commands can be executed after this command.  
    
  {   
   “command” : “close”  
  }

**Immediate Control**

* **stop**The [stop] command stops playback.  
    
  {   
   “command” : “stop”  
  }
* **play**Start playback CG.
* Play Cue before you start playback.

{   
 “command” : “play”  
}

* **pause**The [pause] command pause playback. This command is only used [scroll, closed caption] types.   
  If the device in PLAY or PLAY CUE state, the [pause] command causes the currently playing ID to pause and display a still video frame.  
    
  {   
   “command” : “pause”  
  }
* **set\_data**Set the data of the cued scroll template.  
  Template data is string.

|  |  |  |  |
| --- | --- | --- | --- |
| **JSON Object** | **JSON Value** | **Value Type** | **Description** |
| command | set\_data | String |  |
| object\_id | object\_id | String | ID of object in template |
| data | data | String | ID of object in template data |

{   
 “command” : “set\_data”,  
 “object\_id” : “1”,  
 “data” : “data”  
}

* **set\_rate**Specifies the playback speed. This command is only used [scroll, closed caption] types.  
  If it is positive, it is forward.  
  If it is negative, it is rewind.  
  If it is 0, it is pause.  
  This command can be used to implement the Jog & Shuttle function.

|  |  |  |  |
| --- | --- | --- | --- |
| **JSON Object** | **JSON Value** | **Value Type** | **Description** |
| command | set\_rate | String |  |
| object\_id | object\_id | String | ID of object in template |
| rate | 0 | Number (float) | 0 : pause  Greater then 0 : forward play  Less then 0 : rewind play |

{   
 “command” : “set\_rate”,  
 “object\_id” : “1”,  
 “rate” : 0  
}

* **set\_timecode**If playback is in progress, position the video in timecode. This command is only used [sequence, scroll, closed caption] types.  
  If the time code is larger than the video duration, it moves to the back of the video.

|  |  |  |  |
| --- | --- | --- | --- |
| **JSON Object** | **JSON Value** | **Value Type** | **Description** |
| command | set\_timecode | String |  |
| timecode | 00:00:00:00 | String | Timecode (DF & NDF) |

{   
 “command” : “set\_timecode”,  
 “timecode” : “00:00:00:00”  
}

**Preset/Select Control**

* **play\_cue**The [play\_cue] command prepares for playback.  
  If the ID is not found, an error occurs and state returns STOP.  
  When the CUE is complete, the CUE-DONE flag is set in status.

|  |  |  |  |
| --- | --- | --- | --- |
| **JSON Object** | **JSON Value** | **Value Type** | **Description** |
| command | play\_cue | String |  |
| id | id | String | Template id |

{   
 “command” : “play\_cue”,  
 “id” : “id”  
}

* **play\_cue\_data**Performs similar to [play\_cue].   
  In addition, you can specify playback start timecode and playback duration.  
  This command is only used [sequence, scroll, closed caption] types.

|  |  |  |  |
| --- | --- | --- | --- |
| **JSON Object** | **JSON Value** | **Value Type** | **Description** |
| command | play\_cue\_data | String |  |
| id | id | String | Template id |
| start | 00:00:00:00 | String | Playback start timecode |
| duration | 00:59:59:29 | String | Playback duration timecode |

{   
 “command” : “play\_cue\_data”,  
 “id” : “id”,  
 “start” : “00:00:00:00”,  
 “duration” : “00:59:59:29”  
}

**Sense Request**

* **get\_device\_no**The [get\_device\_no] command get device number.  
    
  {   
   “command” : “get\_device\_no”  
  }  
    
  **Response**

|  |  |  |  |
| --- | --- | --- | --- |
| **JSON Object** | **JSON Value** | **Value Type** | **Description** |
| command | get\_device\_no | String |  |
| response | success | String | Success or Fail |
| no | 0 | Number | Get PCS the set device number. |

* **get\_status**The [get\_status] command get device status.  
    
  {   
   “command” : “get\_status”  
  }  
    
  **Response**

|  |  |  |  |
| --- | --- | --- | --- |
| **JSON Object** | **JSON Value** | **Value Type** | **Description** |
| command | get\_status | String |  |
| response | success | String | Success or Fail |
| * status idle cueing cue\_done play | true  false  false  false | Boolean  Boolean  Boolean  Boolean | Not doing anything or stopped  Preparing play  Completed prepare play  Playing |

* **get\_data**The [get\_data] command get data id of object in template.

|  |  |  |  |
| --- | --- | --- | --- |
| **JSON Object** | **JSON Value** | **Value Type** | **Description** |
| command | get\_data | String |  |
| object\_id | object\_id | String | ID of object in template |

{   
 “command” : “get\_data”  
 “object\_id: : “1”  
}  
  
**Response**

|  |  |  |  |
| --- | --- | --- | --- |
| **JSON Object** | **JSON Value** | **Value Type** | **Description** |
| command | get\_data | String |  |
| response | success | String | Success or Fail |
| data | data | String | ID of object in template data |

* **get\_rate**The [get\_rate] command get current play rate.  
  This command is only used [sequence, scroll, closed caption] types.

|  |  |  |  |
| --- | --- | --- | --- |
| **JSON Object** | **JSON Value** | **Value Type** | **Description** |
| command | get\_rate | String |  |
| object\_id | object\_id | String | ID of object in template |

{   
 “command” : “get\_rate”  
 “object\_id: : “1”  
}  
  
**Response**

|  |  |  |  |
| --- | --- | --- | --- |
| **JSON Object** | **JSON Value** | **Value Type** | **Description** |
| command | get\_rate | String |  |
| response | success | String | Success or Fail |
| rate | 0 | Number (float) | ID of object in template playing rate 0 : pause  Greater then 0 : forward scroll  Less then 0 : rewind scroll |

* **get\_timecode**The [get\_timecode] command get current timecode.  
  This command is only used [scroll, closed caption] types.  
    
  {   
   “command” : “get\_timecode”  
  }  
    
  **Response**

|  |  |  |  |
| --- | --- | --- | --- |
| **JSON Object** | **JSON Value** | **Value Type** | **Description** |
| command | get\_timecode | String |  |
| response | success | String | Success or Fail |
| timecode | 00:00:00:00 | String |  |

1. **MCS – Master Control Switch**

**System Control**

* **open**Open to use device.  
  No other commands can be executed before this command is executed.  
    
  {   
   “command” : “open”  
  }
* **close**  
  Close the device.  
  No other commands can be executed after this command.  
    
  {   
   “command” : “close”  
  }

**Immediate Control**

* **next\_transition**The [next\_transition] command sets the next transition of the background or key.

|  |  |  |  |
| --- | --- | --- | --- |
| **JSON Object** | **JSON Value** | **Value Type** | **Description** |
| command | next\_transition | String |  |
| background | true | Boolean | Background selected |
| key | true | Boolean | Keyers selected |

{   
 “command” : “next\_tranition”  
 “background” : “true”,  
 “key” : “true”  
}

* **start\_transition**The start of the next transition.
* Next Transition before you start transition.

{   
 “command” : “start\_transition”  
}

* **crosspoint\_take**This command set crosspoint take.  
  The [crosspoint\_take] command will clear any overs that may be selected.  
  The number of bus parameter can be change.

|  |  |  |  |
| --- | --- | --- | --- |
| **JSON Object** | **JSON Value** | **Value Type** | **Description** |
| command | crosspoint\_take | String |  |
| * bus program preset output1 output2 | true true true true | Boolean Boolean Boolean Boolean | Select program bus Select preset bus Select output1 Select output2 |
| crosspoint | 0 | Number (integer) | 0 : No change  1~16 : Crosspoint number (Device number) |

{   
 “command” : “crosspoint\_take”,  
 “bus” : {  
 “program” : true,  
 “preset” : true,  
 “output1” : true,  
 “output2” : true  
 },  
 “crosspoint” : 0  
}

* **key\_take**This command set key take.  
  The [key\_take] command turns the key on/off.  
  The number of key and bus parameter can be change.

|  |  |  |  |
| --- | --- | --- | --- |
| **JSON Object** | **JSON Value** | **Value Type** | **Description** |
| command | key\_take | String |  |
| * keys - key1  program  preset  output1  output2 - key2 - key3 - key4 - key5 - key6 - key7 - key8 | true true true true | Boolean Boolean Boolean Boolean | Set the key turns on/off of each bus. |

{   
 “command” : “key\_take”,  
 “keys” : {  
 “key1” : {  
 “program” : true,  
 “preset” : true,  
 “output1” : true,  
 “output2” : true  
 }  
 “key2” : {  
 “program” : true,  
 “preset” : false  
 }  
 }  
}

**Preset/Select Control**

* **set\_transition\_type**This command set transition type.

|  |  |  |  |
| --- | --- | --- | --- |
| **JSON Object** | **JSON Value** | **Value Type** | **Description** |
| command | set\_transition\_type | String |  |
| type | cut | String | cut  fade-fade  cut-fade  mix  fade-cut |

{   
 “command” : “set\_transition\_type”,  
 “type” : “cut”  
}

* **set\_transition\_rate**This command set transition rate.

|  |  |  |  |
| --- | --- | --- | --- |
| **JSON Object** | **JSON Value** | **Value Type** | **Description** |
| command | set\_transition\_rate | String |  |
| rate | slow | String | slow,  medium,  fast,  rate : 0 ~ 65535 frames |

{   
 “command” : “set\_transition\_rate”,  
 “rate” : “slow”  
}

* **set\_enable\_key**This command set enable key.  
  The [set\_enable\_key] command sets whether the key is transited together during transition.  
  Key can be assigned only CG (Cut & Page), Sequence CG, Scroll CG, Logo, Closed Caption & Sub Title devices. Allocate devices from the PCS.

|  |  |  |  |
| --- | --- | --- | --- |
| **JSON Object** | **JSON Value** | **Value Type** | **Description** |
| command | set\_enable\_key | String |  |
| * keys key1 key2 key3 key4 key5 key6 key7 key8 | true true true true true true true true | Boolean Boolean Boolean Boolean Boolean Boolean Boolean Boolean | Set the key enable. |

{   
 “command” : “set\_enable\_key”,  
 “keys” : {  
 “key1” : true,  
 “key2” : true,  
 “key3” : true,  
 “key4” : true,  
 “key5” : true,  
 “key6” : true,  
 “key7” : true,  
 “key8” : true  
 }  
}

**Sense Request**

* **get\_transition\_type**The [get\_transition type] command get currently set the transition type.  
    
  {   
   “command” : “get\_transition\_type”  
  }  
    
  **Response**

|  |  |  |  |
| --- | --- | --- | --- |
| **JSON Object** | **JSON Value** | **Value Type** | **Description** |
| command | get\_transition\_type | String |  |
| response | success | String | Success or Fail |
| type | cut | String | cut  fade-fade  cut-fade  mix  fade-cut |

* **get\_transition\_rate**The [get\_transition\_rate] command get currently set the transition rate.  
    
  {   
   “command” : “get\_transition\_rate”  
  }  
    
  **Response**

|  |  |  |  |
| --- | --- | --- | --- |
| **JSON Object** | **JSON Value** | **Value Type** | **Description** |
| command | get\_transition\_rate | String |  |
| response | success | String | Success or Fail |
| rate | slow | String | slow,  medium,  fast,  rate : 0 ~ 65535 frames |

* **get\_crosspoint\_take**The [get\_crosspoint\_take] command get currently set the crosspoint take information.

|  |  |  |  |
| --- | --- | --- | --- |
| **JSON Object** | **JSON Value** | **Value Type** | **Description** |
| command | get\_crosspoint\_take | String |  |

{   
 “command” : “get\_crosspoint\_take”  
}  
  
**Response**

|  |  |  |  |
| --- | --- | --- | --- |
| **JSON Object** | **JSON Value** | **Value Type** | **Description** |
| command | get\_crosspoint\_take | String |  |
| response | success | String | Success or Fail |
| program | 1 | Number  (integer) | 1~16 : Crosspoint number (Device number) |
| preset | 2 |
| output1 | 3 |
| output2 | 4 |

* **get\_enable\_key**The [get\_enable\_key] command get currently key enable information.

|  |  |  |  |
| --- | --- | --- | --- |
| **JSON Object** | **JSON Value** | **Value Type** | **Description** |
| command | get\_enable\_key | String |  |

{   
 “command” : “get\_enable\_key”

}  
  
**Response**

|  |  |  |  |
| --- | --- | --- | --- |
| **JSON Object** | **JSON Value** | **Value Type** | **Description** |
| command | get\_enable\_key | String |  |
| response | success | String | Success or Fail |
| * keys key1 key2 key3 key4 key5 key6 key7 key8 | true true true true true true true true | Boolean Boolean Boolean Boolean Boolean Boolean Boolean Boolean | Get the key enable information. |

* **get\_key\_take**The [get\_key\_take] command get key on/off status.  
    
  {   
   “command” : “get\_key\_take”  
  }  
    
  **Response**

|  |  |  |  |
| --- | --- | --- | --- |
| **JSON Object** | **JSON Value** | **Value Type** | **Description** |
| command | get\_key\_take | String |  |
| response | success | String | Success or Fail |
| * keys - key1  program  preset  output1  output2 - key2 - key3 - key4 - key5 - key6 - key7 - key8 | true true true true | Boolean Boolean Boolean Boolean | Get the key on/off information of each bus. |

* **get\_status**The [get\_status] command get device status.  
    
  {   
   “command” : “get\_status”  
  }  
    
  **Response**

|  |  |  |  |
| --- | --- | --- | --- |
| **JSON Object** | **JSON Value** | **Value Type** | **Description** |
| command | get\_status | String |  |
| response | success | String | Success or Fail |
| status | quiescent | String | quiescent, transitional, fadingtoblack, fadingfromblack, inblack |