

HDTV925 UDP/TCP control API

UDP, port number is 6004

TCP, port number is 6006

*Note: when use TCP protocol, need to add 20 as the head, the command format as below:

{ Uint32 Ver, Uint32 msgType, Uint32 msgCode, Uint32 Jsonlen, Uint32 reserved}

MsgType = 0x6300, reply: 0x6301

Jsonlen = json

The reset is 0

* Suitable for users who knows about json protocol

Function	Example	Explanation
Factory Reset	Send: {"cmd":"factoryreset"} Reply: {"cmd":"factoryreset","code":0}	code:0 means command correct And the device will reboot
Device reboot	Send: {"cmd":"reset"} Reply: {"cmd":" reset","code":0}	code:0 means command correct And the device will reboot
Get device run time	Send: {"cmd":" getruntime"} Reply:{"cmd":"getruntime","code":0 "starttimes":3,"runtimelist":[{"runtime":348}{"runtime":3053},{"runtime":70483}]}	Starttimes: times of running ;启动次数 Runtime: the running time (s)
Get device info	Send: {"cmd":" getdevops"} Reply: {"cmd":"getdevops","code":0,"cpu":"6.36%","memall":"256.68 MB","memused":"41.36 MB","netin":"2.52Mbps","netout":"52.00Kbps"}	cpu: cpu usage memall: the storage BYTE memeused: in use storage netin: Network input flow netout: Network output flow
Get ID	Send: {"cmd":"getid"} Reply: {"cmd":"getid","code":0,"id":90}	
Set ID	Send: {"cmd":"setid","id":0} Reply: {"cmd":" setid ","code":0 }	Cmd: for command name
获取按键机锁	请求: {"cmd":"getkeylock"} 返回: {"cmd":"getkeylock","code":0,"lock":1}	获取设备锁命令
设置按键锁	请求: {"cmd":"setkeylock","lock":0} 回复: {"cmd":" setkeylock ","code":0 }	lock: 0-解锁按键, 1-锁定按键

获取网络设备信息	请求： <pre>{ "cmd": "getnet" }</pre> 返回： <pre>{ "cmd": "getnet", "code": 0, "ip": "192.168.1.90", "netmask": "255.255.255.0", "gateway": "192.168.1.1", "serverip": "0.0.0.0" }</pre>	获取网络参数命令
设置设备网络信息	请求： <pre>{ "cmd": "setnet", "ip": "192.168.1.100", "netmask": "255.255.255.0", "gateway": "192.168.1.1", "serverip": "192.168.1.1" }</pre> 回复： <pre>{ "cmd": "setnet", "code": 0 }</pre>	ip: ip 地址 netmask: 子网掩码地址 gateway: 网关 serverip: 服务器 IP 此命令会导致系统重新启动。
编码恢复默认值	请求： <pre>{ "cmd": "resetvencattr", "id": 0 }</pre> 回复： <pre>{ "cmd": "resetvencattr", "code": 0 }</pre>	恢复默认编码参数，id: [0-2] 0 主码流将会自适应前端输入 1 子码流变成 1280x720@30, 2 预览流: 720x5676@15
设置编码参数	请求： <pre>{ "cmd": "setvencattr", "id": 0, "enctype": 1, "width": 3840, "height": 2160, "fps": 30, "gop": 30, "bitrate": 1000000 }</pre> 回复： <pre>{ "cmd": "setvencattr", "code": 0 }</pre>	设置编码参数，注意编码参数被设置后，编码将无法跟随前端自适应。 id: 主码流 enctype: 0-H264, 1-H265 width, height 编码宽高 fps 帧率 gop 是 I 帧间隔 bitrate: 码率 (kps)
获取编码参数	请求： <pre>{ "cmd": "getvencattr" }</pre> 回复： <pre>{ "cmd": "getvencattr", "code": 0, "vencattr": [{ "id": 0, "enctype": 1, "profile": 2, "width": 3840, "height": 2160, "fps": 30, "gop": 30, "bitrate": 10000 }, { "id": 1, "enctype": 1, "profile": 2, "width": 1280, "height": 720, "fps": 30, "gop": 30, "bitrate": 4000 }, { "id": 2, "enctype": 0, "profile": 1, "width": 720, "height": 576, "fps": 15, "gop": 30, "bitrate": 1024 }] }</pre>	数组个数支持的码流数量 Id 是索引 enctype: 编码类型: 0-264, 1-265
获取裁剪参数	请求： <pre>{ "cmd": "getenccrop" }</pre>	获取裁剪参数命令， 0- 主码流

	回复： <pre>{ "cmd": "getenccrop", "code": 0, "enccrop": [{ "id": 0, "x": 0, "y": 0, "width": 0, "height": 0 }, { "id": 1, "x": 0, "y": 0, "width": 0, "height": 0 }, { "id": 2, "x": 0, "y": 0, "width": 0, "height": 0 }, { "id": 3, "x": 0, "y": 0, "width": 0, "height": 0 }] }</pre>	1- 辅码流 2- 预览流 3- 环出
设置裁剪参数	请求： <pre>{ "cmd": "setenccrop", "chn": 0, "x": 200, "y": 200, "width": 960, "height": 540 }</pre> 回复： <pre>{ "cmd": "setenccrop", "code": 0 }</pre>	设置编码裁剪参数 chn:通道号 [0-3] , 0 主码流 x,y,width 和,height 裁剪位置, 尺寸 x, y 必须是偶数, width 要对 8 对齐, height 是 4 对齐, 否则会出现异常。尺寸为 0, 取消裁剪
设置外设接口源	请求： <pre>{ "cmd": "setuartsel", "id": 0 }</pre> 回复： <pre>{ "cmd": "setuartsel", "code": 0 }</pre>	选择 UART 接收口, 对于 IP 矩阵 (无光口), 此命令无用 对于分布式 (有光口), 只能接收一个口, RS232 的 ID 为 0 或 RS485 的 ID 为 3;
获取外设接口源	请求: <pre>{ "cmd": "getuartsel", }</pre> 回复： <pre>{ "cmd": "getuartsel", "code": 0, "id": 0 }</pre>	获取 UART 接收口, 对于 IP 矩阵 (无光口), 此命令无用 对于分布式 (有光口), 只能接收一个口, RS232 的 ID 为 0 或 RS485 的 ID 为 3;
获取本地透传端口	请求： <pre>{ "cmd": "getlocal" }</pre> 回复： <pre>{ "cmd": "getlocal", "local": [{ "type": 0, "port": 16232 }, { "type": 1, "port": 16485 }] }</pre>	
设置 RS232 或 485 转发地址和端口	请求： <pre>{ "cmd": "setremote", "remote": [{ "type": 0, "remoteip": "192.168.1.98", "port": 16232 }, { "type": 1, "remoteip": "192.168.1.98", "port": 16485 }] }</pre> 回复：	设置 RS232 和 RS485 转发地址和端口 remote:转发地址和端口 type: 0-RS232 1-RS485

	<pre>{“cmd”:"setremote",“code”:0}</pre>	
获取 RS232 或 485 转发地址和端口	请求: <pre>{“cmd”:"getremote"}</pre> 回复: <pre>{“cmd”:"getremote",“code”:0,“remote”:[{"type":0,“remoteip”:"192.168.11.98",“port”:16232}, {"type":1,“remoteip”:"192.168.11.98",“port”:16458}, {"type":2,“remoteip”:"192.168.11.98",“port”:16457}]}</pre>	
获取 IO 信息	请求: <pre>“cmd”:"getio"</pre> 回复: <pre>{“cmd”:"getio",“code”:0,“ios”:[{"id":0,“dir”:0,“val”:0}, {"id":0,“dir”:0,“val”:0},]}</pre>	获取 IO 请求或获取 IP 返回, 请求是 ios 字段无效 <pre>{“pin”:0,“dir”:0,“val”:0}</pre> id:对外 IO 索引值 dir: io 方向, 0-输出, 1-输入 val:当前 io 输入或输出值, 0 对应低, 1 对应高
设置 IO	请求: <pre>{ “cmd”:"setio”, “ios”:[{"id":0,“dir”:0,“val”:0}, {"id":1 “dir”:0,“val”:0},]}</pre> 回复: <pre>{“cmd”:"setio”,“code”:0}</pre>	设置 IO <pre>{“pin”:0,“dir”:0,“val”:0}</pre> id:对外 IO 索引值 dir: io 方向, 0-输出, 1-输入 val:当前 io 输入或输出值, 0 对应低, 1 对应高 当 IO 设置为输入时, val 的值是无意义的。
获取 Uart	请求: <pre>{“cmd”:"getuart"}</pre> 回复: <pre>{“cmd”:"getuart",“code”:0,“uarts”:[{"id":0,“baudrate”:115200,“datawidth”:8,“stopbit”:1,“parity”:0}, {"id":3,“baudrate”:115200,“datawidth”:8,“stopbit”:1,“parity”:0}]}</pre>	获取 uart 属性 uarts 返回的所有 uart 信息 每一条代表一个串口的信息 Id 代表对外接口的索引值, uart 对外可能是 RS485 也可能是 RS2323 baudrate : 波特率, datawidth :数据位 stopbit :停止位, 支持: 1, 2 parity : 奇偶检验: 0-无, 1-奇, 2-偶 RS232 的 ID 为 0 或 RS485 的 ID 为 3;
设置 Uart	请求: <pre>{ “cmd”:"setuart”, “uarts”:[{"id":0,“baudrate”:115200, “datawidth”:0,“stopbit”:1,“parity”:0 }]}</pre>	设置 uart 属性

	回复： {"cmd":"setuart","code":0}	
设置节点拉流方式	请求： {"cmd":"setprotocol","protocol":0} 回复： {"cmd":"setprotocol","code":0}	设置默认拉流协议 0-udp 组播 1-tcp 单播 2-udp 单播 255-自动切换
获取节点拉流方式	请求： {"cmd":"getprotocol"} 回复： {"cmd":"getprotocol","code":0,"protocol":0}	获取默认拉流协议 0-udp 组播 1-tcp 单播 2-udp 单播 255-自动切换
选择节点	请求： {"cmd":"select","flag":0}	flag: 0-取消选中, 1-选中
绑定音频节点命令	请求： {"cmd":"bindaudio"} 回复： {"cmd":"bindaudio","code":0}	绑定音频命令
解除绑定音频节点命令	请求： {"cmd":"unbindaudio"} 回复： {"cmd":"unbindaudio","code":0}	解除绑定命令
添加矩阵矩阵命令	请求： {"cmd","newmatrix"}	添加矩阵
删除矩阵命令	请求： {"cmd","deletematrix"}	删除矩阵

<p>建立大屏</p>	<p>请求:</p> <pre>{ "cmd": "newwall", "id": 0, "type": "1", "timing": 0, "w": "1920", "h": "1080", "hz": "60", "rows": 3, "cols": 3, "ledw": 5120, "ledh": 2880, "row": 0, "col": 1, "multiaddr": "239.1.1.1", "multiport": 1100, "allres": [{"row": 0, "col": 0, "width": "1920", "height": "1080"}, {"row": 0, "col": 1, "width": "1280", "height": "1080"}, {"row": 0, "col": 2, "width": "1920", "height": "1080"}, {"row": 1, "col": 0, "width": "1920", "height": "720"}, {"row": 1, "col": 1, "width": "1280", "height": "720"}, {"row": 1, "col": 2, "width": "1920", "height": "720"}, {"row": 2, "col": 0, "width": "1920", "height": "1080"}, {"row": 2, "col": 1, "width": "1280", "height": "1080"}, {"row": 2, "col": 2, "width": "1920", "height": "1080"}] }</pre> <p>回复:</p> <pre>{"cmd": "newwall", "code": 0}</pre>	<p>id:屏组 id</p> <p>type: 0->LCD 屏组 1->LED 屏组</p> <p>timing:时序, 0-标准, 100-非标</p> <p>w:节点输出宽度</p> <p>h:节点输出高度</p> <p>hz:节点刷新率</p> <p>rows,cols:屏组行列数</p> <p>x,y:节点所在屏组的位置信息</p> <p>allres:屏组的所有节点信息</p> <p>规定: 屏组同一行的节点的高度一定是相同。 屏组同一列的节点的宽度一定是相同。</p>
<p>删除大屏</p>	<p>请求:</p> <pre>{"cmd": "deletewall", "id": 0}</pre> <p>回复:</p> <pre>{"cmd": "deletewall", "code": 0}</pre>	<p>id:屏组 id</p>
<p>打开视频窗口</p>	<p>请求:</p> <pre>{ "cmd": "openwindow", "id": 0, "x": 0, "y": 0, "width": 1920, "height": 1080, "type": 0, "protocol": 0, "devid": 20, "url": "192.168.1.71:9705/channel=0/stream=0", "suburl": "192.168.1.71:9705/channel=0/stream=1", "stream": 0 }</pre>	<p>id:窗口 id</p> <p>x,y,width,height:窗口位置信息</p> <p>type: 0-编码节点, 1-流媒体服务器, 2-rtsp 视频流, url 必须是 rtsp 地址。</p> <p>protocol:0-组播, 1-tcp, 2-udp 单播。</p> <p>devid:输入设备 ID, RTSP 无</p> <p>url:流的 url 信息 (必须)</p> <p>suburl:子码流的 url 信息 (必须)</p> <p>stream: 0-主码流, 1-子码流,</p>

	<pre> } 回复: {"cmd": "openwindow", "code": 0} </pre>	preview-预览流（编码器支持）
获取窗口信息	<pre> 请求: {"cmd": "getwindowinfo"} 回复: { "cmd": "getwindowinfo", "code": 0, "data": { "list": [{ "id": 0, "x": 0, "y": 0, "width": 1920, "height": 1080, "level": 0, "protocol": 0, "url": "192.168.1.71:9705/channel=0/stream=0", "suburl": "192.168.1.71:9705/channel=0/stream=1", "stream": 0, "type": 0, "devid": "fdfaefads0e1f12120fe" }] } } </pre>	除了 devid 外，其他字段和开窗字段意义一样。devid:表示信源唯一表示。
改变视频窗口位置	<pre> 请求: { "cmd": "changewindow", "id": 0, "x": 0, "y": 0, "width": 1920, "height": 1080 } 回复: {"cmd": "changewindow", "code": 0} </pre>	id:窗口 id x,y,width,height:窗口位置信息
视频窗口置顶	<pre> 请求: {"cmd": "top", "id": 0} 回复: {"cmd": "top", "code": 0} </pre>	id:窗口 id
视频窗口置底	<pre> 请求: {"cmd": "bottom", "id": 0} 回复: {"cmd": "bottom", "code": 0} </pre>	id:窗口 id
关闭视频窗口	<pre> 请求: {"cmd": "closewindow", "id": 0} 回复: {"cmd": "closewindow", "code": 0} </pre>	id:窗口 id

清除所有 视频窗口	请求： {“cmd”：“clearwindows”} 回复： {“cmd”：“clearwindows”,“code”:0}	清空视频，不会关闭声音。
打开声音	请求： { “cmd”：“openaudio”,“id”:0,“type”:0“vol”:60, “url”: “192.168.1.71:9705/channel=0/stream=0”, “protocol”:0 } 回复： {“cmd”：“openaudio”,“code”:0}	打开信源声音 type: 0-编码节点, 1-流媒体服务器, 2-rtsp 视频流, url 必须是 rtsp 地址。 Id: 窗口 ID, 选填。 url: 信源 url vol:声音大小 编器 url 默认是: ip:9705/channel=0/stream=0 Protocol:0-tcp, 1-udp 组播
获取声音	请求： {“cmd”：“getaudioinfo”} 回复： { “cmd”：“getaudioinfo”, “code”:0, “data”{ “url”: “192.168.1.71:9705/channel=0/stream=0”, “vol”:20,“id”:0,“mute”:0,“status”:1 }} } }	获取音频信息 code: 0-OK, 其他错误。 id:窗口 ID, -1 无效 vol:声音大小 status:音频开关状态, 1-开, 0-关 mute: 1-静音, 0-正常 url: 信源 url
关闭声音	请求： { “cmd”：“closeaudio”,“type”:0 “url”: “192.168.1.71:9705/channel=0/stream=0” } 回复： {“cmd”：“closeaudio”,“code”:0}	关闭信源声音 type: 0-编码节点, 1-流媒体服务器, 2-rtsp 视频流, url 必须是 rtsp 地址。 编器 url 默认是: ip:9705/channel=0/stream=0
调节声音	请求： { “cmd”：“adjustaudio”,“type”:0“vol”:60, “url”: “192.168.1.71:9705/channel=0/stream=0” } 回复： {“cmd”：“adjustaudio”,“code”:0}	调节声音大小 0-编码节点, 1-流媒体服务器 vol 是声音大小 url 是视频的地址信息 编器 url 默认是: ip:9705/channel=0/stream=0
同步盒子 视频窗口 信息	请求： { “cmd”：“check”,	"如果 ids 不存在,则表示上位机不存在 窗口, 则会自动清屏。

	<pre> “ids”: [{"id":0}, {"id":1}, {"id":2},] } 回复: {"cmd":"check","code":0} </pre>	
调用场景	<pre> 请求: {"cmd":"callscene","id":1, "type":0} 回复: {"cmd":"callscene","code":0} </pre>	<p>调用场景命令:</p> <p>id 场景的编号</p> <p>type:0-音视频, 1-视频, 2-音频</p> <p>id 场景的编号</p>
保存场景	<pre> 请求: {"cmd":"savescene","name":"scene-01", "id":1" type":0, "type":0 } 回复: {"cmd":"savescene","code": 0 } </pre>	<p>保存场景命令: 保存盒子的当前窗口作为场景。</p> <p>name:场景名称</p> <p>id 场景的编号</p> <p>type:0-音视频, 1-视频, 2-音频</p>
删除场景	<pre> 请求: {"cmd":"deletescene","id":1, "type":0} 回复: {"cmd":"deletescene", "code":0} </pre>	<p>删除场景命令:</p> <p>id 场景的编号</p> <p>type:0-音视频, 1-视频, 2-音频</p>
清除所有场景	<pre> 请求: {"cmd":" clearscenes", "type":0 } 回复: {"cmd":" clearscenes", "code":0} </pre>	<p>清除所有场景</p> <p>type:0-音视频, 1-视频, 2-音频</p>
场景视频同步	<pre> 请求: { “cmd”:"syncscenevideo", “sceneid”:1, “id”:0, “x”:0,“y”:0,“width”:1920,“height”:1080, “type”:0, “protocol”:0, “devid”:20, “url”: “192.168.1.71:9705/channel=0/stream=0”, “suburl”: “192.168.1.71:9705/channel=0/stream=1”, “stream”:0, } 回复: {"cmd":"syncscenevideo","code":0} </pre>	<p>保存场景窗口,调用此接口前应先调用删除场景命令,把对应的场景删除,在保存。否则会有异常。</p> <p>sceneid:场景 ID</p> <p>id:窗口 id</p> <p>x,y,width,height:窗口位置信息</p> <p>type:</p> <p>0-编码节点, 1-流媒体服务器,</p> <p>2-rtsp 视频流, url 必须是 rtsp 地址。</p> <p>protocol:0-组播, 1-tcp, 2-udp 单播。</p> <p>devid:输入设备 ID, RTSP 无</p> <p>url:流的 url 信息 (必须)</p> <p>suburl:子码流的 url 信息 (必须)</p> <p>stream:</p> <p>0-主码流,</p> <p>1-子码流,</p> <p>preview-预览流 (编码器支持</p>

<p>场景音频同步</p>	<p>请求： <pre>{ "cmd": "syncsceneaudio", "sceneid": 1, "type": 0, "vol": 60, "url": "192.168.1.71:9705/channel=0/stream=0", "protocol": 0 }</pre> 回复： <pre>{"cmd": "syncsceneaudio", "code": 0}</pre> </p>	<p>保存场景音频 type: 0-编码节点, 1-流媒体服务器, 2-rtsp 视频流, url 必须是 rtsp 地址。 编 器 url 默 认 是 : ip:9705/channel=0/stream=0</p>
<p>获取场景列表</p>	<p>请求： <pre>{"cmd": "getscenesinfo"}</pre> 回复： <pre>{ "cmd": "getscenesinfo", "code": 0, "data": { "list": [{"id": 0, "name": "null"}] } }</pre> </p>	
<p>获取场景窗口</p>	<p>请求： <pre>{"cmd": "getscenewindowinfo", "id": 0}</pre> 回复： <pre>{ "cmd": "getscenewindowinfo", "code": 0, "id": 0, "data": { "list": [[{"id": 0, "x": 0, "y": 0, "width": 1920, "height": 1080, "level": 0, "protocol": 0, "url": "192.168.1.71:9705/channel=0/stream=0", "suburl": "192.168.1.71:9705/channel=0/stream=1", "stream": 0, "type": 0, "devid": "fdfaefads0e1f12120fe"}]] } }</pre> </p>	<p>外层 id:场景 ID Code:0-OK, 其他错误。</p>

获取场景 音频	请求： {“cmd”：“getsceneaudioinfo”,“id”: 0} 回复： { “cmd”：“getsceneaudioinfo”, “code”:0, “id”:0, “data”: { “audio“:{ “url”：“192.168.1.71:9705/channel=0/stream=0”, “vol”:20,“id”:0,“mute”:0,“status”:1 } } }	Id-窗口 ID, -1 无效 Url-源 url 信息 Mute:0-非静音, 1-静音 Status: 0-关闭, 1-打开
------------	--	---