

云手机服务器

SDK 参考

文档版本 01
发布日期 2023-07-20



版权所有 © 华为技术有限公司 2024。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

商标声明



HUAWEI和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受华为公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为公司对本文档内容不做任何明示或暗示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

安全声明

漏洞处理流程

华为公司对产品漏洞管理的规定以“漏洞处理流程”为准，该流程的详细内容请参见如下网址：

<https://www.huawei.com/cn/psirt/vul-response-process>

如企业客户须获取漏洞信息，请参见如下网址：

<https://securitybulletin.huawei.com/enterprise/cn/security-advisory>

目录

1 简介	1
2 限制与约束	2
3 触控接口	3
3.1 触控注入.....	3
3.2 按键注入.....	5
4 音频接口	7
4.1 启动音频服务.....	7
4.2 停止音频服务.....	8
5 视频接口	10
5.1 启动视频服务.....	10
5.2 停止视频服务.....	13
5.3 设置物理分辨率或屏幕密度.....	14
5.4 动态设置视频参数.....	15
5.5 强制 I 帧.....	16
6 扩展接口	18
6.1 获取版本号.....	18
6.2 获取媒体服务支持能力.....	18
7 错误码	20

1 简介

本文档描述云手机libCPHMediaEngine.so提供的触控和音视频接口，方便您调用云手机的媒体能力，结合您的业务进行二次开发。

首先建议您参考我们提供的外部开源代码，该代码提供了完整的端侧和云侧服务部署方案，能够让您以最快的速度部署真机客户端与云手机接入交互的整套服务，相关开源代码路径如下：

- 端云协同的端侧SDK，请参考：[端云协同的端侧 SDK](#)。

2 限制与约束

- 出于安全考虑，云手机提供的音视频服务不支持真机侧SDK多实例连接；如果有同屏互动需求，请在端云协同的云侧接入引擎进行视频流分发。
- 本文档[SetPhysicalResolution接口](#)不能在出流过程中设置物理分辨率。

3 触控接口

3.1 触控注入

接口介绍

```
int InjectTouchData(const InputTouch &inputTouch);
```

功能描述

触控数据注入。

输入参数

表 3-1 输入参数说明

参数名称	说明
inputTouch	触控注入参数结构体，详见表3-2。

表 3-2 触控注入参数结构体说明

参数名称	参数类型	说明
pointerID	number	真机触点编号，多指触控下会有多个触点，取值范围[0,9]，即最多支持10个触点。
orientation	number	真机屏幕旋转角度，默认为0，可设置为1(90°)，2(180°)，3(270°)。
action	number	真机触控行为，如ACTION_DOWN, ACTION_UP，取值范围请参考： https://developer.android.com/reference/android/view/MotionEvent
pressure	number	真机触控压力值，默认为129。

参数名称	参数类型	说明
physicalWidth	number	真机屏幕分辨率宽，默认为0，取值范围[0,2160]。
physicalHeight	number	真机屏幕分辨率高，默认为0，取值范围[0,4096]。
touchX	number	真机触点坐标，取值范围[0,physicalWidth]。
touchY	number	真机触点坐标，取值范围[0,physicalHeight]。
touchTime	number	真机触控时间与上一次ACTION_DOWN时间的偏移量。

📖 说明

- 真机端触控事件的采集方式，详见端云协同的端侧SDK开源代码：[端云协同的端侧SDK](#)。
- 真机端触控事件采集相关代码，请参考：[onTouch方法](#)。

输出参数

无

返回值

0: 成功

小于0: 数据注入失败

错误码

请参见[错误码](#)。

调用示例

```
// 触控数据结构体
struct InputTouch {
    uint8_t pointerID;
    uint8_t orientation;
    uint8_t action;
    uint16_t pressure;
    uint16_t touchX;
    uint16_t touchY;
    uint16_t physicalWidth;
    uint16_t physicalHeight;
    int32_t touchTime;
    uint8_t reserved[16]; // 保留字段
};

InputTouch inputTouch;
inputTouch.pointerID = 0;
inputTouch.orientation = 0;
inputTouch.action = 0; // ACTION_DOWN
inputTouch.pressure = 129;
inputTouch.physicalWidth = 720;
inputTouch.physicalHeight = 1280;
inputTouch.touchX = 52;
```

```
inputTouch.touchY = 157;
inputTouch.touchTime = 0;
int ret = InjectTouchData(inputTouch);
//注意
要确保数据类型是本地字节序，数据来自网络则需要使用ntohl()方法进行本地字节序转换
```

3.2 按键注入

接口介绍

```
int InjectKeyData(const InputKey &inputKey);
```

功能描述

按键数据注入。

输入参数

表 3-3 输入参数说明

参数名称	说明
inputKey	按键注入参数结构体。键值表和按键行为设置范围，详见： https://developer.android.com/reference/android/view/KeyEvent 。

输出参数

无

返回值

0: 成功
小于0: 数据注入失败

错误码

请参见[错误码](#)。

调用示例

```
// 键值数据结构体
struct InputKey {
    uint16_t keyCode; // 真机触控键值
    uint16_t action; // 按键行为
    uint8_t reserved[8]; // 保留字段
}

InputKey inputKey;
// keyCode和action对应安卓框架中KeyEvent类的keyCode和action
// 音量加键keyCode=24, 按下action=0
inputKey.keyCode = 24;
inputKey.action = 0;
```

```
int ret = InjectKeyData(index, inputKey);    // 音量加键按下
usleep(100000);
// 音量加键keyCode=24, 抬起action=1
inputKey.keyCode = 24;
inputKey.action = 1;
int ret = InjectKeyData(inputKey);          // 音量加键抬起
```

4 音频接口

4.1 启动音频服务

接口介绍

```
int OpenAudio(const char *jsonConfig, AudioCallback callback);
```

功能描述

打开音频采集服务。

输入参数

表 4-1 输入参数说明

参数名称	说明
jsonConfig	音频初始化参数json串（推荐设置为空，使用默认参数），详见 表4-2 。
callback	音频数据回调函数。

表 4-2 jsonConfig 参数说明

参数名称	参数类型	说明
audioType	number	音频类型，默认为1（PCM），可设置为0（OPUS）。
sampleRate	number	采样率（单位：Hz），默认为48000，可设置为48000/8000/44100。
channels	number	采样通道数，默认为2（双声道），可设置为1（单声道）。

参数名称	参数类型	说明
interval	number	采样间隔，默认为10，PCM下默认为10不可修改，OPUS下可设置为10/20。
bitrate	number	音频编码码率（单位：bps/s），默认为2*channels*sampleRate，OPUS下可以修改，取值范围[500, 512000]，PCM下不可修改。

输出参数

无

返回值

0: 成功

小于0: 音频服务初始化失败

错误码

请参见[错误码](#)。

调用示例

```
static void TestAudioCallback(uint8_t* buffer, uint32_t length, AudioCallbackExt &audioCallbackExt)
{
    std::cout << "audio_Datasize = " << length << std::endl;
    std::cout << "frameTime is : " << audioCallBackExt.timestamp << std::endl;
    return 0;
}
//若需要获取pcm类型音频数据
int ret = OpenAudio(nullptr, TestAudioCallback);

//若需要获取opus类型音频数据
char *jsonstr = "{ \"audioType\":0, \"samplerate\":48000, \"channels\":2, \"interval\":10}";
int ret = OpenAudio(jsonstr, TestAudioCallback);
```

4.2 停止音频服务

接口介绍

```
int CloseAudio(void);
```

功能描述

关闭音频采集服务。

输入参数

无

输出参数

无

返回值

0: 成功

小于0: 停止音频服务失败

错误码

请参见[错误码](#)。

5 视频接口

5.1 启动视频服务

接口介绍

```
int OpenVideo(const char* jsonConfig, VideoCallback callback);
```

功能描述

启动视频采集服务。

如需视频编码后出流，推荐使用外置编码方案，有助于降低编码时延和云手机资源消耗，并且外置编码可支持h265编码出流。

输入参数

表 5-1 输入参数说明

参数名称	说明
jsonConfig	视频初始化参数json串，详见 表5-2 。
callback	视频数据回调函数。

表 5-2 jsonConfig 参数说明

参数名称	参数类型	说明
frame_type	string	出流帧类型，默认值为“h264”，可根据需要设置为“yuv”，“h265”。

参数名称	参数类型	说明
encode_type	number	<p>编码类型，默认值为0（AUTO模式）。</p> <p>在AUTO模式下，系统会自动选择编码类型：优先使用外置编码，当外置编码不可用时，将自动切换云手机CPU软编码。</p> <p>AUTO模式下自动切换CPU软编码存在以下两种例外情况：</p> <ol style="list-style-type: none">1. 由于CPU软编仅支持h264，当frame_type设置为“h265”时，无法自动切换；2. 当使用AUTO模式，但资源池elb ip或port配置非法，系统将返回错误码，无法自动切换。 <p>支持指定为1（CPU软编码）</p>
fps	number	<p>编码帧率，默认值为云手机规格的帧率，设置范围[10, 云手机规格的帧率]，支持设置为10的倍数。</p>
bitrate	number	<p>编码码率，720p默认值：3M，1080p默认值：6M，设置范围[100000, 10000000]</p> <p>基于编码画质最大保障和传输带宽最大节约，编码码率建议值如下：</p> <p>游戏多开场景（30fps）：</p> <p>720p/h264: 2M（高清）1.3M（标清）1M（流畅）</p> <p>720p/h265: 1.3M（高清）1M（标清）0.7M（流畅）</p> <p>1080p/h264: 4M（高清）2.6M（标清）2M（流畅）</p> <p>1080p/h265: 2.6M（高清）2M（标清）1.4M（流畅）</p> <p>云游戏场景（30fps）：</p> <p>720p/h264: 3M（高清）2M（标清）1.3M（流畅）</p> <p>720p/h265: 2M（高清）1.3M（标清）1M（流畅）</p> <p>1080p/h264: 6M（高清）4M（标清）2.6M（流畅）</p> <p>1080p/h265: 4M（高清）2.6M（标清）2M（流畅）</p>
stream_width	number	<p>出流帧宽度，默认为云手机分辨率规格宽度，设置范围[240, 云手机分辨率规格宽度]。</p> <p>出流帧宽度需要为8的倍数，且宽高比需要与物理分辨率宽高比保持一致。</p>

参数名称	参数类型	说明
stream_height	number	出流帧高度，默认为云手机分辨率规格高度，设置范围[240, 云手机分辨率规格高度]。 出流帧高度且需要为8的倍数，且宽高比需要与物理分辨率宽高比保持一致。
rcmode	number	码率控制模式，默认为2。 0: ABR（平均码率） 1: CRF（画质优先） 2: CBR（恒定码率） 3: CAPPED_CRF（画质优先，但限制码率）
gop	number	视频编码GOP长度，默认为3000，设置范围[30, 5000]。
profile	number	编码规格，默认为66（Baseline），可设置为77（Main），100（High）。 客户端为手机屏幕，建议选择66。 客户端为电视大屏，建议选择77或100。
repeat_fps	number	补帧帧率，默认为0（不开补帧），设置范围[0, 云手机规格的帧率]。
remote_scheduling_elb_ip	string	外置编码*调度资源池elb ip，默认为空，需配置为购买elb时分配的内网ip地址。
remote_scheduling_elb_port	number	外置编码*调度资源池elb端口，默认值为-1，需配置为购买elb时设置的port。

📖 说明

- 外置编码是采用与云手机服务器同VPC下的ECS为云手机提供编码能力的技术方案，支持H.264/H.265硬件编码出流，不占用云手机资源，具有编码时延低、画质质量高的特点。
- 具体接入及配置方法，详见：[华为云手机编码池](#)。

输出参数

无

返回值

0: 成功

小于0: 视频服务初始化失败

错误码

请参见[错误码](#)。

调用示例

json串的配置如下,参数可选,如果不设置,使用默认值:

```
{
  "frame_type": "h264",
  "remote_scheduling_elb_ip": "127.0.0.1",
  "remote_scheduling_elb_port": 10001
}
```

调用示例:

```
static void TestVideoCallback(uint8_t* buffer, uint32_t length, VideoCallbackExt &videoCallbackExt)
{
    std::cout <<"video_Datasize = "<< length <<std::endl;
    int orientation = (int)videoCallbackExt.orientation;
    // orientation代表视频出流方向,需要根据此回调值对客户端进行适配
    // orientation=0: 竖屏
    // orientation=8: 横屏,需要对回调后的视频内容做逆时针90度的旋转
    return;
}

// 场景一: 请求h264编码出流,采用默认AUTO模式
const char *jsonStr = "{\"frame_type\":\"h264\",\"remote_scheduling_elb_ip\":"
    "\"127.0.0.1\",\"remote_scheduling_elb_port\":\"10001\"}";
int ret = OpenVideo(jsonStr, TestVideoCallback);

// 场景二: 请求h265编码出流,采用默认AUTO模式,并判断返回值错误码(错误码详情描述见文档第6部分)
// 如果没有外置编码资源则切换h264 AUTO模式出流
const char *jsonStr = "{\"frame_type\":\"h265\",\"remote_scheduling_elb_ip\":"
    "\"127.0.0.1\",\"remote_scheduling_elb_port\":\"10001\"}";
int ret = OpenVideo(jsonStr, TestVideoCallback);
if (ret == -3300) {
    jsonStr = "{\"frame_type\":\"h264\",\"remote_scheduling_elb_ip\":"
        "\"127.0.0.1\",\"remote_scheduling_elb_port\":\"10001\"}";
    int ret = OpenVideo(jsonStr, TestVideoCallback);
}
```

5.2 停止视频服务

接口介绍

```
int CloseVideo(void);
```

功能描述

关闭视频采集服务。

输入参数

无

输出参数

无

返回值

0: 成功

小于0: 停止视频服务失败

错误码

请参见[错误码](#)。

5.3 设置物理分辨率或屏幕密度

接口介绍

- 设置物理分辨率：int SetPhysicalResolution (int width, int height);
- 设置物理分辨率和屏幕密度：int SetPhysicalResolution(int width, int height, int dpi);

功能描述

设置云手机的物理分辨率或屏幕密度（在出流过程中调用该接口设置物理分辨率会失败）。

输入参数

表 5-3 输入参数说明

参数名称	说明
width	目标屏幕宽度，默认值为当前云手机规格分辨率宽度，设置范围 [240, 云手机分辨率规格宽度]。
height	目标屏幕高度，默认值为当前云手机规格分辨率高度，设置范围 [240, 云手机分辨率规格高度]。
dpi	目标屏幕密度，默认值为当前云手机规格分辨率密度，设置范围 [140, 600]。 若设置为0或不设置，表示目标屏幕密度将根据目标屏幕宽度和目标屏幕高度自动计算后设置。

输出参数

无

返回值

0: 成功

小于0: 设置物理分辨率失败

错误码

请参见[错误码](#)。

5.4 动态设置视频参数

接口介绍

```
int SetParams(const char* jsonConfig);
```

功能描述

动态设置视频服务参数配置。

输入参数

表 5-4 输入参数说明

参数名称	说明
jsonConfig	动态设置视频参数json串，详见表5-5。

表 5-5 jsonConfig 参数说明

参数名称	参数类型	说明
fps	number	编码帧率，默认为云手机规格的帧率。设置范围[10, 云手机规格的帧率]，支持设置为10的倍数。
bitrate	number	码率，设置范围[100000, 10000000]，码率推荐设置值见表5-2。
gop	number	视频编码I帧间隔，设置范围[30, 5000]，建议设置为300。
profile	number	编码画质级别，可设置为66（Baseline），77（Main），100（High），手机屏幕编码建议选择66。
rcmode	number	码率控制模式。 0: ABR（平均码率） 1: CRF（画质优先） 2: CBR（恒定码率） 3: CAPPED_CRF（画质优先，但限制码率）
stream_width	number	出流帧宽度，设置范围[240, 云手机分辨率规格宽度]，并且出流帧宽度需要为8的倍数。
stream_height	number	出流帧高度，设置范围[240, 云手机分辨率规格高度]，并且出流帧高度需要为8的倍数。

参数名称	参数类型	说明
repeat_fps	number	补帧帧率，默认为0（不开补帧），设置范围[0, 云手机规格帧率]。

输出参数

无

返回值

0: 成功
小于0: 设置视频参数失败

错误码

请参见[错误码](#)。

调用示例

```
// 该接口需要在OpenVideo成功后调用
// 游戏多开场景720p/h264/30fps切换流畅等级出流
const char *pStr = "{\"bitrate\":1000000}";
if (0 != mediaEngine.SetParams(pStr)) {
    cout << "SetParams json version failed" << endl;
}
```

5.5 强制 I 帧

接口介绍

```
int GenerateKeyFrame(void);
```

功能描述

强制下一帧为I帧。

输入参数

无

输出参数

无

返回值

0: 成功
小于0: 设置I帧失败

错误码

请参见[错误码](#)。

6 扩展接口

6.1 获取版本号

接口介绍

```
const char * GetVersion();
```

功能描述

获得当前版本号。

输入参数

无

输出参数

无

返回值

版本号字符串。

错误码

请参见[错误码](#)。

调用示例

```
const char * version = GetVersion();
```

6.2 获取媒体服务支持能力

接口介绍

```
const char * GetCPHMediaFeatures();
```

功能描述

获得当前云手机媒体服务的支持能力。

输入参数

无

输出参数

无

返回值

云手机支持能力json串

表 6-1 获取媒体服务支持能力返回参数说明

参数名称	说明
H264	是否支持H264编码出流。true（支持）/false（不支持）
H265	是否支持H265编码出流。true（支持）/false（不支持）
YUV	是否支持YUV出流。true（支持）/false（不支持）
HardEncode	是否支持硬件GPU编码出流。true（支持）/false（不支持）
SoftEncode	是否支持软件CPU编码出流。true（支持）/false（不支持）
FPS_MAX	当前云手机规格帧率最大值。int
Width_MAX	当前云手机规格分辨率最大值。int
Height_MAX	当前云手机规格分辨率最大值。int

错误码

请参见[错误码](#)。

调用示例

```
const char * mediaFeatures = GetCPHMediaFeatures();
```

7 错误码

表 7-1 错误码说明

错误模块	错误类型	错误码	说明
-	未知错误	-1	未定义的错误类型。
触控模块	参数错误	-1000	触控参数设置错误。
		-1001	按键参数设置错误。
	注入错误	-1100	触控注入错误。
		-1101	按键注入错误。
音频模块	参数错误	-2000	音频参数设置错误。
	启动错误	-2100	音频服务启动错误。
	关闭错误	-2200	音频服务关闭错误。
视频模块	参数错误	-3000	视频参数设置错误。
		-3001	屏幕分辨率不符合要求。 屏幕宽度和高度不能小于240；屏幕分辨率不能小于规格的标准分辨率。
		-3002	DPI数值不符合要求，要求DPI数值在 [140,600]范围。
		-3003	出流过程不能修改屏幕分辨率。
	启动错误	-3100	视频服务启动错误。
	关闭错误	-3200	视频服务关闭错误。
	池化错误	-3300	elb ip/端口合法但配置错误或worker无空闲资源，无法正常出流； 建议客户排查elb ip/端口的正确性和worker剩余资源量。

错误模块	错误类型	错误码	说明
		-3301	设置了非法的elb ip/端口, 无法正常出流; 建议客户排查elb ip/端口的合法性。
		-3302	请求的帧率/分辨率与worker不匹配, 无法正常出流; 建议客户排查worker和请求的帧率/分辨率配置。
	系统错误	-3400	系统内部错误, 请联系技术支持。