

# sdk 说明文档

作者	创建/修改日期	版本	备注
杜建斌	2018-06-02	V1.0	基础接口

# 目 录

1. 备注
2. 基本函数
3. 附录
4. 演示

## 备 注

当前SDK只支持V5版本彩卡软件的二次开发

1. SDK开发平台为Windows7，开发工具为VS2013，开发语言为C/C++； Demo示例程序开发平台为Windows7，开发工具为VS2013/Qt5.5，开发语言为C/C++；

2. 所有导出函数均为c函数，调用方式为\_declspec，Demo支持 c/c++、Qt5.5;
3. 需要传递的参数IP地址为异步卡联网后自动分配的IP。

## 基本函数

- 1、**功能**：获取 SDK 版本信息。  
**函数原型**：ZHSDK\_API int ZH\_SDKVer(void);  
**概述**：返回 SDK 版本信息
- 2、**功能**：获得上一次 SDK 函数调用的错误代码。  
**函数原型**：int ZHSDK\_API ZH\_GetSDKLastError();  
**概述**：返回上一次 SDK 函数调用的错误代码
- 3、**功能**：初始化系统。  
**函数原型**：int ZHSDK\_API ZH\_Initialize(void \*pExParamsBuf, int nBufSize);  
**概述**：  
pExParamsBuf：UI 控制面板指针；  
nBufSize：预留参数，可不传入。成功返回 1，失败返回 0。
- 4、**功能**：创建一个屏幕。  
**函数原型**：int ZHSDK\_API ZH\_CreateScreen(const std::wstring& name,int width, int height, int bgcolor, const std::wstring& imagePath, int nsize);  
**概述**：  
name：屏幕名称、  
width：屏幕宽度（0-1024）、  
height：屏幕高度（0-512）、  
bgcolor：屏幕背景颜色（0x000000-0xffffffff）、  
magePath：屏幕背景图像文件路径、  
nsize：额外参数缓冲区长度；  
返回所创建屏幕 ID(1-n),失败返回-1.
- 5、**功能**：添加一个节目到屏幕。  
**函数原型**：int ZHSDK\_API ZH\_AddProgram(int nScreenID, bool bGlobal, const std::wstring& name, int playmode, const std::string& endTime, int count, int duration, int bgcolor, const std::wstring& imagePath, int nsize);  
**概述**：  
nScreenID：指定屏幕的 ID(1-n),  
bGlobal：是否为全局节目的标志(0 或 1)、  
name：屏幕名称、  
playmode：播放模式（0-4 依次为默认<指定此模式时，参数 5, 6, 7 无效>、循环<指定此模式时，该节目在播放时保持循环播放状态，参数 5, 6, 7 无效>、指定时间结束 <当模式为指定时间结束时则参数 5--endTime 有效，参数 6-7 的 count 和 duration 无效>、指定次数 <当模式为指定次数时，参数 6--count 有效，参数 5, 7 无效>、指定播放时长 <当模式为指定时长时，参数 7--duration 有效，参数 5, 6 无效>、PlayDefault<指定此模式时，参数 5, 6, 7 无效>）、  
endTime：结束时间(格式："2018-6-20 20:25:18")、

count: 播放次数 (1-n)、  
duration: 节目播放时长 (1-n 单位:秒)、  
backcolor: 节目背景颜色 (0x000000-0xffffffff)、  
imagePath: 节目背景播放文件、  
nsize: 额外参数缓冲区长度;  
成功则返回节目 ID(1-n),失败返回-1.

**6、功 能:** 添加一个区域到指定节目。

**函数原型:** int ZHSDK\_API ZH\_AddArea(int nProgramID, const std::wstring& name,int x, int y, int width, int height, bool benable, int leftBorder, int topBorder, int speed, int style, int bordercolor, const std::wstring& imagePath, int iTransparency, int backcolor, std::vector<std::wstring> backgroundlist, int nsize);

**概 述:**

nProgramID: 指定的节目 ID (1-n)、  
name: 区域名称、  
x: 区域位置坐标 X (0-n, 小于实际创建的屏幕宽度)、  
y: 区域位置坐标 Y (0-n, 小于实际创建的屏幕高度)、  
width: 区域宽度 (0-n, 小于实际创建的屏幕宽度)、  
height: 区域高度 (0-n, 小于实际创建的屏幕高度)、  
benable: 是否启用边框 (bool 值)、  
leftBorder: 左右边框宽度 (1-10)、  
topBorder: 上下边框宽度 (1-10)、  
speed: 跑边速度 (0-10)、  
style: 边框风格 (0-3: 静态、逆时针、顺时针、闪烁)、  
bordercolor: 边框颜色 (0x000000-0xffffffff)、  
imagePath: 边框加载文件、  
iTransparency: 区域背景透明度 (0~255)、  
backcolor: 区域背景颜色值 (0x000000-0xffffffff)、  
backgroundlist: 区域背景加载文件列表、  
nsize: 额外参数缓冲区长度;  
成功则返回区域 ID (1-n)。失败返回-1。

**7、功 能:** 添加一个单行文本素材到指定区域。

**函数原型:** int ZHSDK\_API ZH\_AddSimpleText(int nAreaID, const std::wstring& name, int times, const std::wstring& content, const std::wstring& Font, bool beBold, bool beItalic, bool beUnderline, int ftSize ,int ftWidth, int ftColor, int ftSpace, int moveEffect, int moveSpeed, int duration, int colorMode, std::vector<int> fontColorList, int textureSpeed, int txtStyle, int txtColorTemplate, int bloop, int enterAnimationEffect, int enterAnimationDuration, int midAnimationEffect, int midAnimationDuration, int leaveAnimationEffect, int leaveAnimationDuration, int nsize);

**概 述:**

nAreaID: 指定区域 ID (1-n)、  
name: 单行文本元素名称、  
times: 播放次数 (1-100)、  
content: 文字内容、  
Font: 字体名称、  
beBol: 是否加粗 (bool 值)、  
beItalic: 是否斜体 (bool 值)、  
beUnderline: 是否下划线 (bool 值)、

ftSize: 字体大小 (8-200 注: 字体大小不得超过实际所在区域大小)、  
ftWidth: 字宽 (暂时不用)、  
ftColor: 字体颜色 (0x000000-0xffffffff)、  
ftSpace: 字间距 (0-99)、  
moveEffect: 移动模式 (402-406: 左移、静止、右移、上移、下移)、  
moveSpeed: 移动速度 (1-10)、  
duration: 停留时长 (不使用)、  
colorMode: 色彩模式 (见附录 1)、  
fontColorList: 色彩列表 (用于炫彩效果 注: 每个列表元素就是一个 RGB 值)、  
textureSpeed: 纹理速度 (不使用)、  
txtStyle: 文字特效 (0-3 默认、空心字、阴影字、立体字)、  
txtColorTemplate: 炫彩字模板 (见附录2)、  
bloop: 是否永久循环 (bool值)、  
enterAnimationEffect: 入场特效 (单行文本不使用)、  
enterAnimationDuration: 入场时间 (单行文本不使用)、  
midAnimationEffect: 停留特效 (单行文本不使用)、  
midAnimationDuration: 停留时间 (不使用)、  
leaveAnimationEffect: 出场特效 (单行文本不使用)、  
leaveAnimationDuration: 出场时间 (单行文本不使用)、  
nsize: 额外参数缓冲区长度;  
成功则返回自身ID。失败返回-1。

**8、功 能:** 添加一个图片素材到指定区域。

**函数原型:** int ZHSDK\_API ZH\_AddImage(int nAreaID, const std::wstring& name, const std::wstring& path, int fitMode, int bloop, int enterAnimationEffect, int enterAnimationDuration, int midAnimationEffect, int midAnimationDuration, int leaveAnimationEffect, int leaveAnimationDuration, int nsize);

**概 述:**

nAreaID: 指定区域 ID (1-n)、  
name: 图片素材名称、  
path: 图片文件路径、  
fitMode: 图片显示在指定区域内的填充模式(0-4 拉伸、原始大小、按宽适应、按高适应、自动缩放)、  
bloop: 是否永久循环 (bool值)、  
enterAnimationEffect: 入场特效 (20、21、22、23、40、81、140、141 上移、下移、左移、右移、淡入淡出、覆盖效果-向下、水平百叶窗效果、垂直百叶窗效果 )、  
enterAnimationDuration: 入场时间 (1000000-0 )、  
midAnimationEffect: 停留特效(0-4 拉伸、原始大小、按宽适应、按高适应、自动缩放)、  
midAnimationDuration: 停留时间 (1000000000-0)、  
leaveAnimationEffect: 出场特效 (20、21、22、23、40、81、140、141 上移、下移、左移、右移、淡入淡出、覆盖效果-向下、水平百叶窗效果、垂直百叶窗效果 )、  
leaveAnimationDuration: 出场时间 (1000000-0)、  
nsize: 额外参数缓冲区长度;  
成功则返回自身ID。失败返回-1。

**9、功 能:** 添加一个时钟素材到指定区域(注: 此版本 SDK 添加时钟, 程序运行的电脑必须支持 OPENGL)。

**函数原型:** int ZHSDK\_API ZH\_AddClock(int nAreaID, const std::wstring& name, bool bNumberStyle, int shape, bool bSingleLine, POINT position, int bloop, int enterAnimationEffect, int enterAnimationDuration, int midAnimationEffect, int midAnimationDuration, int leaveAnimationEffect, int leaveAnimationDuration, int nsize);

**概 述:**

nAreaID: 指定区域 ID (1-n)、  
name: 时钟素材名称、  
bNumberStyle: 是否以数字形式显示 (bool 值)、  
shape: 时钟形状 (0-3 圆形、椭圆、正方形、长方形)、  
bSingleLine: 是否单行显示 (bool 值)、  
position: 时钟中心点位置、  
bloop: 是否永久循环 (bool 值)、  
enterAnimationEffect: 入场特效 (不使用)、  
enterAnimationDuration: 入场时间 (不使用)、  
midAnimationEffect: 停留特效 (不使用)、  
midAnimationDuration: 停留时间 (不使用)、  
leaveAnimationEffect: 出场特效 (不使用)、  
leaveAnimationDuration: 出场时间 (不使用)、  
nsize: 额外参数缓冲区长度;  
成功则返回自身 ID。失败返回-1。

**10、功 能:** 添加一个传感器素材到指定区域。

**函数原型:** int ZHSDK\_API ZH\_AddSensor(int nAreaID, std::wstring name, int types, bool bSingleLine, int bloop, int enterAnimationEffect, int enterAnimationDuration, int midAnimationEffect, int midAnimationDuration, int leaveAnimationEffect, int leaveAnimationDuration, int nsize);

**概 述:**

nAreaID 指定区域 ID (1-n)、  
name: 传感器素材名称、  
types: 传感器类型 (0x0000: 所有类型、 0x0001: 温度、0x0002: 湿度、0x0004: 噪音、0x0008: PM2.5、0x0010:PM10、0x0020: co2、0x0040: 甲醛、 0x0080: 风向、0x0100: 风速、0x0200: 风力)、  
bSingleLine: 是否单行显示 (bool 值)、  
bloop: 是否永久循环 (bool 值)、  
enterAnimationEffect: 入场特效 (不使用)、  
enterAnimationDuration: 入场时间 (不使用)、  
midAnimationEffect: 停留特效 (不使用)、  
midAnimationDuration: 停留时间 (不使用)、  
leaveAnimationEffect: 出场特效 (不使用)、  
leaveAnimationDuration: 出场时间 (不使用)、  
nsize: 额外参数缓冲区长度;  
成功则返回自身 ID。失败返回-1。

**11、功 能:** 添加一个多行文本素材到指定区域。

**函数原型:** int ZHSDK\_API ZH\_AddRichText(int nAreaID, const std::wstring& name, const std::wstring& content, int bgColor, const std::wstring& Font, bool beBold, bool beItalic, bool beUnderline, int ftsize, int ftwidth, int ftcolor, int ftspc, int bloop, int enterAnimationEffect, int enterAnimationDuration, int midAnimationEffect, int midAnimationDuration, int leaveAnimationEffect, int leaveAnimationDuration, int nsize);

### 概 述:

nAreaID 指定区域 ID (1-n)、  
name: 多行文本素材名称、  
content: 多行文本内容、  
bgColor: 多行文本背景颜色值 (0x000000-0xffffffff)、  
Font: 字体名称、  
beBold: 是否加粗 (bool 值)、  
beItalic: 是否斜体 (bool 值)、  
beUnderline: 是否下划线 (bool 值)、  
ftsize: 字体大小 (8-200)、  
ftwidth: 字宽 (暂时不用)、  
ftcolor: 字体颜色 (0x000000-0xffffffff)、  
ftspace: 字间距 (0-99)、  
bloop 是否永久循环 (bool 值)、  
enterAnimationEffect 入场特效 (20、21、22、23、40、81、140、141 上移、下移、左移、右移、淡入淡出、覆盖效果-向下、水平百叶窗效果、垂直百叶窗效果 )、  
enterAnimationDuration 入场时间 (1000000-0 )、  
midAnimationEffect 停留特效 (不使用 )、  
midAnimationDuration 停留时间 (1000000000-0)、  
leaveAnimationEffect 出场特效 (20、21、22、23、40、81、140、141 上移、下移、左移、右移、淡入淡出、覆盖效果-向下、水平百叶窗效果、垂直百叶窗效果 )、  
leaveAnimationDuration 出场时间 (1000000-0 )、  
nsize: 额外参数缓冲区长度;  
成功则返回自身 ID。失败返回-1。

12、功 能: 添加一个视频素材到指定区域。

函数原型: int ZHSDK\_API ZH\_AddVideo(int nAreaID, const std::wstring& name, const std::wstring& path, int volume, int bloop, int enterAnimationEffect, int enterAnimationDuration, int midAnimationEffect, int midAnimationDuration, int leaveAnimationEffect, int leaveAnimationDuration, int nsize);

### 概 述:

nAreaID:指定区域 ID (0-n)、  
name: 视频素材名称、  
path: 视频文件路径、  
volume: 视频播放音量 (0-100)、  
bloop: 是否永久循环 (bool 值)、  
enterAnimationEffect: 入场特效 (20、21、22、23、40、81、140、141 上移、下移、左移、右移、淡入淡出、覆盖效果-向下、水平百叶窗效果、垂直百叶窗效果 )、  
midAnimationDuration: 入场时间 (1000000-0 )、  
midAnimationEffect: 停留特效 (不使用)、  
midAnimationDuration: 停留时间 (不使用)、  
leaveAnimationEffect: 出场特效 (20、21、22、23、40、81、140、141 上移、下移、左移、右移、淡入淡出、覆盖效果-向下、水平百叶窗效果、垂直百叶窗效果 )、  
leaveAnimationDuration: 出场时间 (1000000-0 )、  
nsize: 额外参数缓冲区长度;  
成功则返回自身 ID。失败返回-1。

13、功 能: 播放指定工程。

函数原型: int ZHSDK\_API ZH\_PlayProject(void \*pDirPath);

- 概述:** pDirPath: 项目路径; 成功则返回 0, 失败返回-1。
- 14、**功能:** 播放指定节目。  
**函数原型:** int ZHSDK\_API ZH\_PlayAct(int nProgramID);  
**概述:** nProgramID: 节目 ID (1-n); 成功则返回 0, 失败返回-1。
- 15、**功能:** 播放指定区域。  
**函数原型:** int ZHSDK\_API ZH\_PlayActContent(int nAreaID);  
**概述:** nAreaID: 区域 ID (1-n); 成功则返回 0, 失败返回-1。
- 16、**功能:** 播放指定素材。  
**函数原型:** int ZHSDK\_API ZH\_PlayMaterial(int nMaterialID);  
**概述:** nMaterialID: 素材 ID (1-n); 成功则返回 0, 失败返回-1。
- 17、**功能:** 获得指定屏幕位置及大小。  
**函数原型:** void ZHSDK\_API ZH\_LedScreenRect (int nScreenID,RECT\* rct);  
**概述:** nScreenID: 指定屏幕的 ID (1-n)、rct: RECT 指针;
- 18、**功能:** 设置指定屏幕位置及大小。  
**函数原型:** void ZHSDK\_API ZH\_SetLedScreenRect (int nScreenID, const RECT& rct);  
**概述:** nScreenID: 指定屏幕的 ID (1-n)、rct: RECT 位置大小数据;
- 19、**功能:** 删除指定虚拟屏幕。  
**函数原型:** int ZHSDK\_API ZH\_RemoveScreen(int nScreenID);  
**概述:** 传入指定屏幕的 ID; 成功则返回 0, 失败返回-1。
- 20、**功能:** 删除指定节目。  
**函数原型:** int ZHSDK\_API ZH\_RemoveProgram(int nProgramID);  
**概述:** nScreenID: 指定节目的 ID (1-n); 成功则返回 0, 失败返回-1。
- 21、**功能:** 删除指定区域。  
**函数原型:** int ZHSDK\_API ZH\_RemoveArea(int nAreaID);  
**概述:** nAreaID: 指定区域的 ID (1-n); 成功则返回 0, 失败返回-1。
- 22、**功能:** 删除指定素材。  
**函数原型:** int ZHSDK\_API ZH\_RemoveMaterial(int nMaterialID);  
**概述:** nMaterialID: 指定素材的 ID (1-n); 成功则返回 0, 失败返回-1。
- 23、**功能:** 保存项目到指定目录。  
**函数原型:** int ZHSDK\_API ZH\_SaveProject(const std::wstring& pDirPath);  
**概述:** pDirPath: 存放项目的路径。失败返回-1。
- 24、**功能:** 发送到指定 IP 的播放盒。  
**函数原型:** int ZHSDK\_API ZH\_SendToDevice(std::vector<ZH\_IP\_INFO>\* idList);  
**概述:** idList: 播放盒基本信息 ZH\_IP\_INFO 列表 (ZH\_IP\_INFO 结构体包含: 播放盒 ip 地址(std::string)、用户密码(std::string)、以及播放盒控制的 LED 屏幕 ID(int)。); 成功则返回 0, 失败返回-1。

- 25、功 能：清空异步卡节目。  
函数原型：int ZHSDK\_API ZH\_ClearProgram(std::vector<ZH\_IP\_INFO>\* idList);  
概 述：idList：播放盒基本信息 ZH\_IP\_INFO 列表（ZH\_IP\_INFO 结构体包含：播放盒 ip 地址(std::string)、用户密码(std::string)、以及播放盒控制的 LED 屏幕 ID(int)。)；成功则返回 0，失败返回-1。
- 26、功 能：获取 LED 异步卡的数量。  
函数原型：int ZHSDK\_API ZH\_DeviceCount();  
概 述：成功返回异步播放盒总数。
- 27、功 能：设置 LED 异步卡开关屏。  
函数原型：int ZHSDK\_API ZH\_LedScreenPower(bool powerOn, std::vector<ZH\_IP\_INFO>\* idList);  
概 述：  
powerOn：是否开启电源（bool 值 true 为开启，false 为关闭）  
idList：播放盒基本信息 ZH\_IP\_INFO 列表（ZH\_IP\_INFO 结构体包含：播放盒 ip 地址(std::string)、用户密码(std::string)、以及播放盒控制的 LED 屏幕 ID(int)。注：如果传入该列表为空，将根据参数 1 开关本地所有异步播放盒控制的远端屏幕）；成功则返回 0，失败返回-1。
- 28、功 能：设置 LED 屏的亮度。  
函数原型：int ZHSDK\_API ZH\_SetBrightness(int brightNessValue, std::vector<ZH\_IP\_INFO>\* idList);  
概 述：  
brightNessValue：亮度值（0-255）  
idList：播放盒基本信息 ZH\_IP\_INFO 列表（ZH\_IP\_INFO 结构体包含：播放盒 ip 地址(std::string)、用户密码(std::string)、以及播放盒控制的 LED 屏幕 ID(int)。注：如果为空，将所有异步播放盒亮度值全都设为参数 1 传递的值）；成功则返回 0，失败返回-1。
- 29、功 能：创建一个异步卡分组。  
函数原型：int ZHSDK\_API ZH\_CreateGroup(const std::wstring& name);  
概 述：name：组名；成功返回当前组 Id，失败返回-1。
- 30、功 能：添加一个异步卡到分组。  
函数原型：int ZHSDK\_API ZH\_AddOneDeviceToGroup(int groupId, const std::string& ip);  
概 述：groupId：组 Id（该 ID 值为创建异步卡分组的返回值）、ip：异步播放盒 IP 地址；成功返回 0，失败返回-1。
- 31、功 能：校时，包含单个异步卡或多个异步列表或分组。  
函数原型：int ZHSDK\_API ZH\_CalibrationTime(const std::string& timeArea, const std::string& currentTime, bool sumTm, bool autoSyn, std::vector<ZH\_IP\_INFO>\* idList);  
概 述：  
timeArea：时区，传入值见附录 3（不传则默认为当前时区，如果参数 4 为 true 则该参数无效）、  
currentTime：当前时间（不传则默认为系统当前时间，如果参数 4 为 true 则该参数无效）（格式："2018-6-20 9:25:18"）、  
sumTm：是否为夏令时（bool 值，如果参数 4 为 true 则该参数无效）、  
autoSyn：是否自动同步（bool 值）、

idList: 播放盒基本信息 ZH\_IP\_INFO 列表 (ZH\_IP\_INFO 结构体包含: 播放盒 ip 地址、用户密码、以及播放盒控制的 LED 屏幕 ID。 注: 如该列表传入为空则代表将所有异步播放盒的时间区域、当前时间设为参数 1、参数 2);

成功则返回 0, 失败返回-1。

32、功 能: 获取设备 IP 列表。

函数原型: int ZHSDK\_API ZH\_GetDeviceIdList(std::vector<ZH\_IP\_INFO>\* idList);

概 述:

idList: 播放盒 IP 等基本信息 ZH\_IP\_INFO 列表指针 (ZH\_IP\_INFO 结构体包含: 播放盒 ip 地址、用户密码、以及播放盒控制的 LED 屏幕 ID<注: 在此函数调用后该成员存放的是设备是否在线的值, 0 为离线, 1 为在线>。

成功则返回 0, 失败返回-1。

33、功 能: 清空本地节目。

函数原型: int ZHSDK\_API ZH\_ClearLocalActs();

概 述: 该函数调用后将清空本地所有节目/区域/元素。

成功则返回 0, 失败返回-1。

34、功 能: 云服务器登录。

函数原型: int ZHSDK\_API ZH\_CloudLogin(const std::string& userName, const std::string& password);

概 述:

userName: 用户名、

password: 用户密码。

成功则返回 0, 失败返回-1。

35、功 能: 云服务器登出。

函数原型: void ZHSDK\_API ZH\_CloudLogout();

概 述: 登出云服务器。

36、功 能: 云发送节目到异步卡, 包含单个异步卡或多个异步列表, 或者分组。

函数原型: int ZHSDK\_API ZH\_CloudSend(const std::wstring& proPath, std::vector<std::string>\* idList);

概 述:

proPath: 本地项目路径、

idList: 云端设备列表。

成功则返回 0, 失败返回-1。

37、功 能: 获取服务器设备列表, 以分组方式获取列表。

函数原型: int ZHSDK\_API ZH\_GetCloudDeviceIdList(std::vector<std::string>\* idList);

概 述:

idList: 返回云端设备列表。

成功则返回 0, 失败返回-1。

38、功 能: 设置云端设备控制屏幕的亮度值。

函数原型: int ZHSDK\_API ZH\_SetCloudDeviceBrightness(int brightNessValue, std::vector<std::string>\* idList);

概 述:

brightNessValue: 亮度值、

idList: 云端设备列表。

成功则返回 0, 失败返回-1。

39、功 能：云端设备校时。

函数原型：int ZHSDK\_API ZH\_CloudDeviceCalibrationTime(const std::string& timeArea, const std::string& currentTime, bool sumTm, bool autoSyn, std::vector<std::string>\* idList);

概 述：

timeArea: 时区，传入值见附录 3（不传则默认为当前时区，如果参数 4 为 true 则该参数无效）、

currentTime: 当前时间（不传则默认为系统当前时间，如果参数 4 为 true 则该参数无效）（格式："2018-6-20 9:25:18"）、

sumTm: 是否为夏令时（bool 值，如果参数 4 为 true 则该参数无效）、

autoSyn: 是否自动同步（bool 值）、

idList: 云端设备 ID 列表。注：如该列表指针传入为空则代表将该用户名下所有云端设备的时间区域、当前时间设为参数 1、参数 2。

成功则返回 0，失败返回-1。

40、功 能：云端设备开关屏。

函数原型：int ZHSDK\_API ZH\_CloudDeviceLedScreenPower(bool powerOn, std::vector<std::string>\* idList);

概 述：

powerOn: 是否开启电源（bool 值 true 为开启，false 为关闭）

idList: 云端设备 ID 列表。注：如该列表指针传入为空，将根据参数 1 开关当前用户名下所有云端设备控制的远端屏幕。

成功则返回 0，失败返回-1。

41、功 能：清空云端设备节目。

函数原型：int ZHSDK\_API ZH\_CloudDeviceClearPrograms(std::vector<std::string>\* idList);

概 述：

idList: 云端设备 ID 列表。

成功则返回 0，失败返回-1。

## 附 录

1.) colorMode: 色彩模式（

0: 默认、

1: 水平渐变 2 色、

2: 水平渐变 3 色、

3: 水平渐变 4 色、

4: 水平渐变 5 色、

5: 水平渐变 6 色、

6: 垂直渐变 2 色、

7: 垂直渐变 3 色、

8: 垂直渐变 4 色、

9: 垂直渐变 5 色、

10: 垂直渐变 6 色、

- 11: 斜角渐变 2 色、
- 12: 斜角渐变 3 色、
- 13: 斜角渐变 4 色、
- 14: 斜角渐变 5 色、
- 15: 斜角渐变 6 色、
- 16: 反斜角渐变 2 色、
- 17: 反斜角渐变 3 色、
- 18: 反斜角渐变 4 色、
- 19: 反斜角渐变 5 色、
- 20: 反斜角渐变 6 色、
- 21: 动画渐变 2 色、
- 22: 动画渐变 3 色、
- 23: 动画渐变 4 色、
- 24: 动画渐变 5 色、
- 25: 动画渐变 6 色、
- 26: 闪动 2 色、
- 27: 闪动 3 色、
- 28: 闪动 4 色、
- 29: 闪动 5 色、
- 30: 闪动 6 色、
- 31: 彩虹流光字（水平）、
- 32: 彩虹流光字（垂直）、
- 33: 彩虹流光字（斜角）、
- 34: 彩虹流光字（反斜角）、
- 35: 一字一色 2 色、
- 36: 一字一色 3 色、
- 37: 一字一色 4 色、
- 38: 一字一色 5 色、
- 39: 一字一色 6 色、
- 40: 2 色过渡、
- 41: 淡入淡出）、

## 2.) txtColorTemplate: 炫彩字模板（

- 0: 无、
- 1: 静止、
- 2: D旋转字上移、
- 3: 3D旋转字左移、
- 4: 3D旋转字右移、
- 5: 飞入、
- 6: 爆炸1、
- 7: 爆炸2（轨迹）、
- 8: 烟花1、
- 9: 烟花2（轨迹）、
- 10: 分裂、
- 11: 溶解、
- 12: 斑点、
- 13: 扫描、
- 14: 单色描边、

- 15: 光线、
  - 16: 膨胀、
  - 17: 旗帜飘扬、
  - 18: 落下、
  - 19: 升起、
  - 20: 走近、
  - 21: 左翻滚、
  - 22: 右翻滚、
  - 23: 波浪、
  - 24: 波动、
  - 25: 推挤、
  - 26: 顺时针循环、
  - 27: 逆时针循环、
  - 28: 左移圆环、
  - 29: 右移圆环、
  - 30: 探照灯、
  - 31: 放大镜、
  - 32: 闪烁、
  - 33: 打字、
  - 34: 涟漪、
  - 35: 模糊、
  - 36: 镜像、
  - 37: 原地旋转X、
  - 38: 原地旋转Y、
  - 39: 原地旋转Z、
- 注: 异步软件暂时不支持炫彩字)、

### 3.) 允许设置的时区:

<timezones>

```
<timezone id="Pacific/Majuro">马朱罗</timezone>
<timezone id="Pacific/Midway">中途岛</timezone>
<timezone id="Pacific/Honolulu">檀香山</timezone>
<timezone id="America/Anchorage">安克雷奇</timezone>
<timezone id="America/Los_Angeles">美国太平洋时间 (洛杉矶)</timezone>
<timezone id="America/Tijuana">美国太平洋时间 (提华纳)</timezone>
<timezone id="America/Phoenix">美国山区时间 (凤凰城)</timezone>
<timezone id="America/Chihuahua">奇瓦瓦</timezone>
<timezone id="America/Denver">美国山区时间 (丹佛)</timezone>
<timezone id="America/Costa_Rica">美国中部时间 (哥斯达黎加)</timezone>
<timezone id="America/Chicago">美国中部时间 (芝加哥)</timezone>
<timezone id="America/Mexico_City">美国中部时间 (墨西哥城)</timezone>
<timezone id="America/Regina">美国中部时间 (里贾纳)</timezone>
<timezone id="America/Bogota">哥伦比亚时间 (波哥大)</timezone>
<timezone id="America/New_York">美国东部时间 (纽约)</timezone>
```

<timezone id="America/Caracas">委内瑞拉时间 (加拉加斯)</timezone>  
<timezone id="America/Barbados">大西洋时间 (巴巴多斯)</timezone>  
<timezone id="America/Manaus">亚马逊标准时间 (马瑙斯)</timezone>  
<timezone id="America/Santiago">圣地亚哥</timezone>  
<timezone id="America/St\_Johns">纽芬兰时间 (圣约翰)</timezone>  
<timezone id="America/Sao\_Paulo">圣保罗</timezone>  
<timezone id="America/Argentina/Buenos\_Aires">布宜诺斯艾利斯</timezone>  
<timezone id="America/Godthab">戈特霍布</timezone>  
<timezone id="America/Montevideo">乌拉圭时间 (蒙得维的亚)</timezone>  
<timezone id="Atlantic/South\_Georgia">南乔治亚</timezone>  
<timezone id="Atlantic/Azores">亚述尔群岛</timezone>  
<timezone id="Atlantic/Cape\_Verde">佛得角</timezone>  
<timezone id="Africa/Casablanca">卡萨布兰卡</timezone>  
<timezone id="Europe/London">格林尼治标准时间 (伦敦)</timezone>  
<timezone id="Europe/Amsterdam">中欧标准时间 (阿姆斯特丹)</timezone>  
<timezone id="Europe/Belgrade">中欧标准时间 (贝尔格莱德)</timezone>  
<timezone id="Europe/Brussels">中欧标准时间 (布鲁塞尔)</timezone>  
<timezone id="Europe/Sarajevo">中欧标准时间 (萨拉热窝)</timezone>  
<timezone id="Africa/Windhoek">温得和克</timezone>  
<timezone id="Africa/Brazzaville">西部非洲标准时间 (布拉扎维)</timezone>  
<timezone id="Asia/Amman">东欧标准时间 (安曼)</timezone>  
<timezone id="Europe/Athens">东欧标准时间 (雅典)</timezone>  
<timezone id="Asia/Beirut">东欧标准时间 (贝鲁特)</timezone>  
<timezone id="Africa/Cairo">东欧标准时间 (开罗)</timezone>  
<timezone id="Europe/Helsinki">东欧标准时间 (赫尔辛基)</timezone>  
<timezone id="Asia/Jerusalem">以色列时间 (耶路撒冷)</timezone>  
<timezone id="Europe/Minsk">明斯克</timezone>  
<timezone id="Africa/Harare">中部非洲标准时间 (哈拉雷)</timezone>  
<timezone id="Asia/Baghdad">巴格达</timezone>  
<timezone id="Europe/Moscow">莫斯科</timezone>  
<timezone id="Asia/Kuwait">科威特</timezone>  
<timezone id="Africa/Nairobi">东部非洲标准时间 (内罗毕)</timezone>  
<timezone id="Asia/Tehran">伊朗标准时间 (德黑兰)</timezone>  
<timezone id="Asia/Baku">巴库</timezone>  
<timezone id="Asia/Tbilisi">第比利斯</timezone>  
<timezone id="Asia/Yerevan">埃里温</timezone>  
<timezone id="Asia/Dubai">迪拜</timezone>  
<timezone id="Asia/Kabul">阿富汗时间 (喀布尔)</timezone>  
<timezone id="Asia/Karachi">卡拉奇</timezone>  
<timezone id="Asia/Oral">乌拉尔</timezone>  
<timezone id="Asia/Yekaterinburg">叶卡捷林堡</timezone>  
<timezone id="Asia/Calcutta">加尔各答</timezone>  
<timezone id="Asia/Colombo">科伦坡</timezone>  
<timezone id="Asia/Katmandu">尼泊尔时间 (加德满都)</timezone>

```
<timezone id="Asia/Almaty">阿拉木图</timezone>
<timezone id="Asia/Rangoon">缅甸时间 (仰光)</timezone>
<timezone id="Asia/Krasnoyarsk">克拉斯诺亚尔斯克</timezone>
<timezone id="Asia/Bangkok">曼谷</timezone>
<timezone id="Asia/Shanghai">中国标准时间 (北京)</timezone>
<timezone id="Asia/Hong_Kong">香港时间 (香港)</timezone>
<timezone id="Asia/Irkutsk">伊尔库茨克时间 (伊尔库茨克)</timezone>
<timezone id="Asia/Kuala_Lumpur">吉隆坡</timezone>
<timezone id="Australia/Perth">佩思</timezone>
<timezone id="Asia/Taipei">台北时间 (台北)</timezone>
<timezone id="Asia/Seoul">首尔</timezone>
<timezone id="Asia/Tokyo">日本时间 (东京)</timezone>
<timezone id="Asia/Yakutsk">雅库茨克时间 (雅库茨克)</timezone>
<timezone id="Australia/Adelaide">阿德莱德</timezone>
<timezone id="Australia/Darwin">达尔文</timezone>
<timezone id="Australia/Brisbane">布里斯班</timezone>
<timezone id="Australia/Hobart">霍巴特</timezone>
<timezone id="Australia/Sydney">悉尼</timezone>
<timezone id="Asia/Vladivostok">海参崴时间 (符拉迪沃斯托克)</timezone>
<timezone id="Pacific/Guam">关岛</timezone>
<timezone id="Asia/Magadan">马加丹时间 (马加丹)</timezone>
<timezone id="Pacific/Auckland">奥克兰</timezone>
<timezone id="Pacific/Fiji">斐济</timezone>
<timezone id="Pacific/Tongatapu">东加塔布</timezone>
</timezones>
```

## 演 示

演示工程在随附目录：LEDPlayerV5\_SDK\demo 下，可以使用 VS2013 打开 demo\_c\_test.sln 编译运行；演示程序是利用 Qt5.5 开发的，工程配置里可能有开发时用的 PC 机 Qt5.5 安装路径（例如：D:/Qt/Qt5.5.0/5.5/msvc2013/lib），这可能与调试时的环境不匹配，使用前请自行下载 Qt5.5 安装，并修改工程 Qt 配置路径。