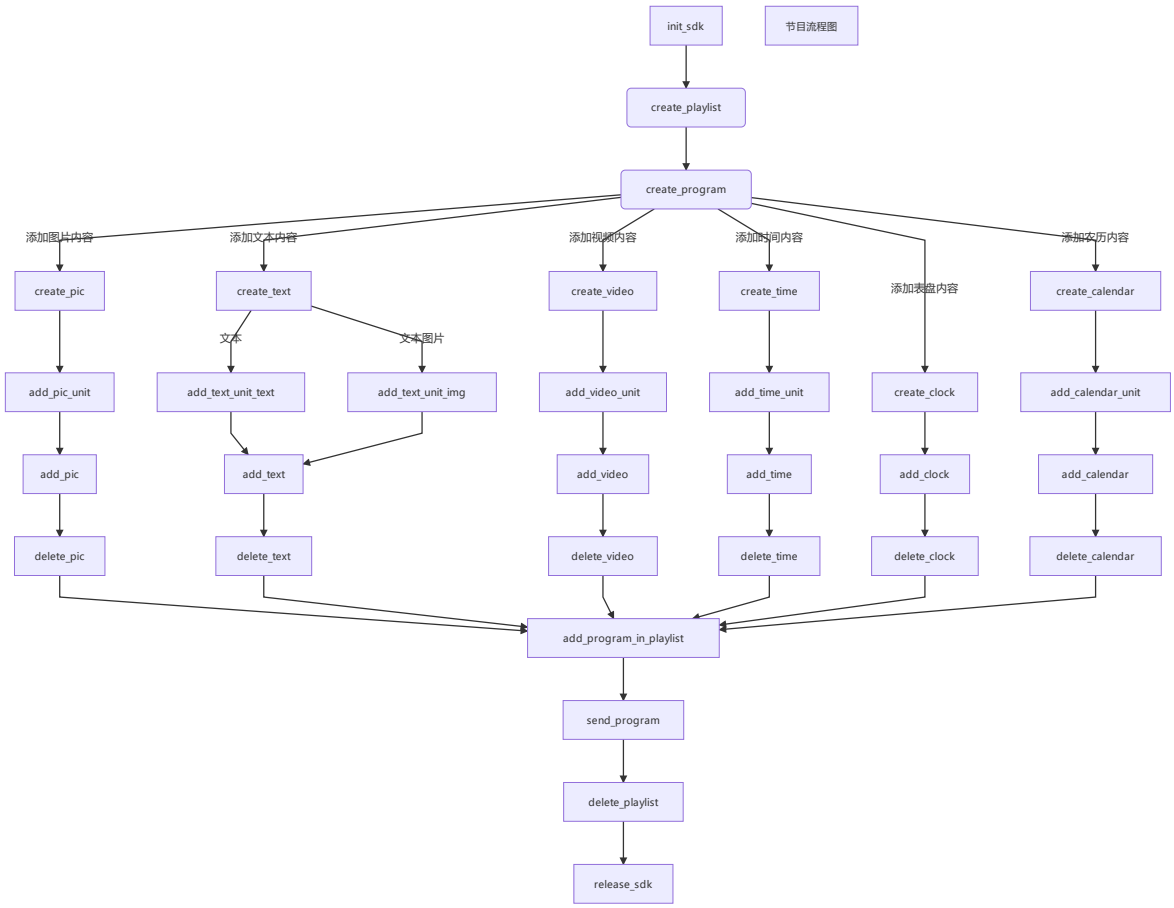
BX-Y系列动态库(YQNetCom.dll)使用说明

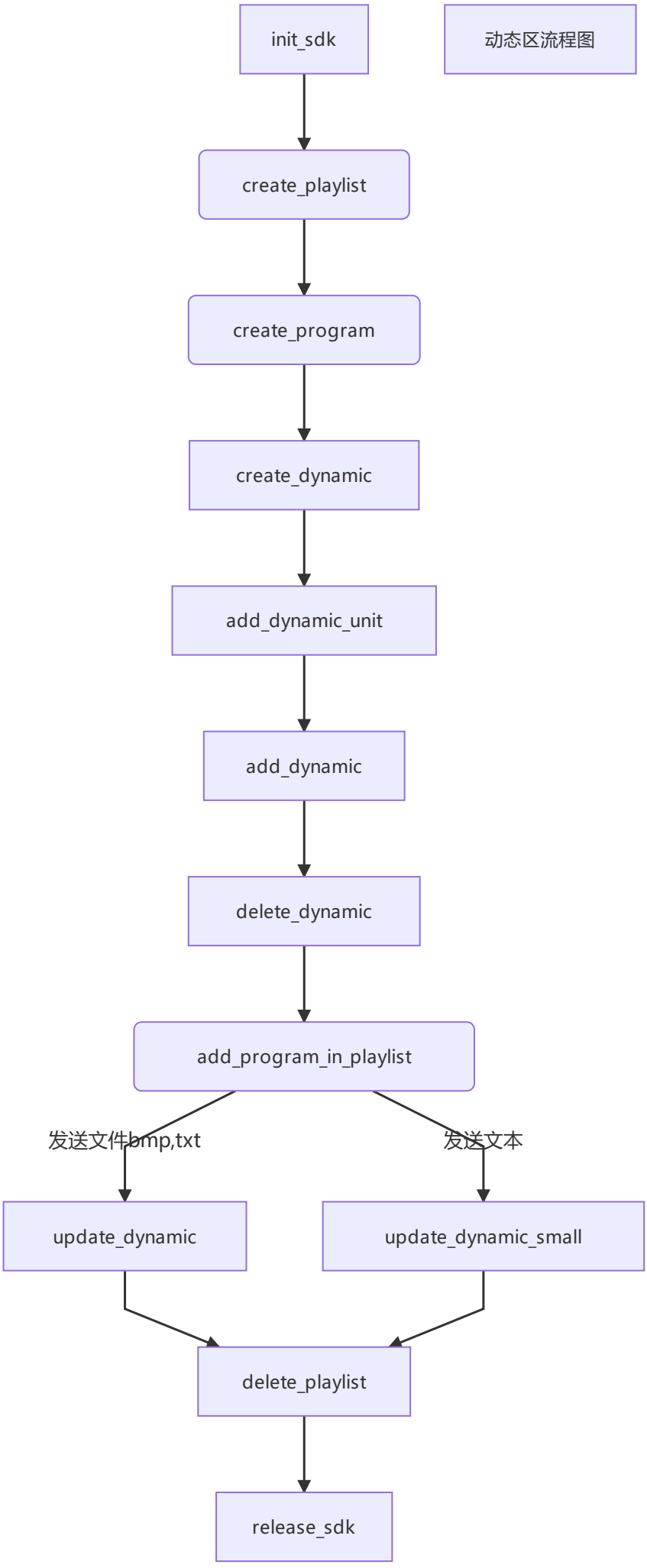
版本历史:

版本号	日期	作者	描述
19.11.19.1	2019-11-19	wanght	添加视频文件add_video_unit(),节目添加视频区add_video()添加参数：crop_type【HDMI输出指定位置 可以给空字符串】
20.06.08.0	2020-06-08	wanght	1.send_program()添加参数is_make 2.添加加密接口add_tls_md5() 3.update_dynamic/update_dynamic_small接口中immediately_play参数int类型改为string类型
21.04.18.0	2021-04-18	wanght	1.add_time,add_clock,add_calendar添加adjustment参数：用以调整时差；支持天数，格式“±dd:hh:mm:ss” 2.add_dynamic_unit添加 key_list 类型为"URLText"，且获取网络资源为json格式数据时有效。获取数据的字段 proxyService 动态区"URLText"类型，网络资源为代理服务时，该参数有效，且该参数是代理服务的标识参数（存在且非空） 3.添加节目富文本接口 4.添加节目炫彩字接口

节目调用流程图


init_sdk
节目流程图
create_playlist
create_program
添加图片内容
添加文本内容
添加视频内容
添加时间内容
添加表盘内容
添加农历内容
create_pic
create_text
create_video
create_time
create_calendar
文本
文本图片
add_pic_unit
add_text_unit_text
add_text_unit_img
add_video_unit
add_time_unit
create_clock
add_calendar_unit
add_pic
add_text
add_video
add_time
add_clock
add_calendar
delete_pic
delete_text
delete_video
delete_time
delete_clock
delete_calendar
add_program_in_playlist
send_program
delete_playlist
release_sdk

动态区调用流程图


init_sdk
动态区流程图
create_playlist
create_program
create_dynamic
add_dynamic_unit
add_dynamic
delete_dynamic
add_program_in_playlist
发送文件bmp,txt
发送文本
update_dynamic
update_dynamic_small
delete_playlist
release_sdk

接口说明

#define _CALL_STD __stdcall

1.常规API

1.1 初始化动态库 init_sdk

返回值：成功返回0；失败返回错误号

参数：无

说明： 初始化动态库

函数： int _CALL_STD init_sdk();

1.2 释放动态库 release_sdk

返回值：成功返回0；失败返回错误号

参数：无

说明： 释放动态库

函数： int *CALL_STD release_sdk();*

2.局域网模式获取控制卡API

2.1 搜索控制卡 search_card

返回值：成功返回0；失败返回错误号

参数：

参数	说明
cards	控制卡数据缓存空间,数据解析格式BroadCast2
card_count	控制卡数量

说明： 固定IP下搜索控制卡

函数： int _CALL_STD search_card(BYTE* cards, int* card_count);

3.服务器模式获取控制卡API

3.1 启动服务器 Start_Native_Server

返回值：成功返回0；失败返回错误号

参数：

参数	描述
port	服务器端口

说明： 启动服务器

函数： int _CALL_STD Start_Native_Server(int port);

3.2 停止服务器 Stop_Server

返回值：成功返回0；失败返回错误号

参数：无

说明： 停止服务器

函数：int _CALL_STD Stop_Server();

3.3 搜索控制卡 Get_CardList

返回值：成功返回0；失败返回错误号

参数：

参数	说明
cards	控制卡数据缓存空间BroadCast2
card_count	控制卡数量

说明： 服务器模式搜索控制卡

函数：int _CALL_STD Get_CardList(BYTE* cards, int* card_count);

3.4 获取控制卡使用端口 Get_Port_Barcode

返回值：成功返回控制卡使用端口

参数：

参数	说明
Barcode	控制卡条码

说明： 服务器模式，根据条码获取控制卡使用端口

函数：int CALL_STD Get_Port_Barcode(TEXT_CHAR* Barcode);

3.5 获取控制卡使用端口 Get_Port_Pid

返回值：成功返回控制卡使用端口

参数：

参数	说明
pid	控制卡pid码

说明： 服务器模式，根据PID码获取控制卡使用端口

函数：int CALL_STD Get_Port_Pid(TEXT_CHAR* pid);

4.节目API

4.1 创建节目清单 create_playlist

返回值：成功返回节目单句柄；失败返回错误号

参数：

参数	说明
W	屏幕宽度
H	屏幕高度
device_type	控制卡型号BX-Y04/8280 BX-Y08/8536 BX-Y2/8792 BX-Y2L/9304 BX-Y3/9048 BX-Y5E/10584

说明： 创建节目清单

函数：unsigned long _CALL_STD create_playlist(int w, int h, int device_type);

4.2 创建节目 create_program

返回值：成功返回节目句柄；失败返回错误号

参数：

参数	说明
name	节目名
bg_color	背景颜色

说明： 创建节目

函数：unsigned long CALL_STD create_program(TEXT_CHAR* name, _TEXT_CHAR* bg_color);

4.3 节目添加到节目单 add_program_in_playlist

返回值：成功返回0；失败返回错误号

参数：

参数	说明
playlist	播放列表
program	节目句柄
play_mode	播放模式，0 – 定长播放，1 – 定次播放
play_time	播放时长或者播放次数
aging_start_time	播放时效的开始日期 格式2012-06-20
aging_end_time	播放时效结束日期
period_ontime	播放时段开始时间 格式08:30:00
period_offtime	放时段结束时间
play_week	bit0 ~ bit6 依次表示星期一至星期天，0—表示该天节目不能播放，1—表示该天节目可以播放

说明： 节目添加到节目单

函数： int _CALL_STD add_program_in_playlist(unsigned long playlist, unsigned long program, int play_mode, int play_time, _TEXT_CHAR* aging_start_time, _TEXT_CHAR* aging_end_time, _TEXT_CHAR* period_ontime, _TEXT_CHAR* period_offtime, int play_week);

4.4 发送节目单 send_program

返回值： 成功返回0；失败返回错误号

参数：

参数	说明
ip	控制卡IP
port	端口
user_name	登录名 “guest”
user_pwd	登录密码 “guest
tmp_path	播放列表缓存路径
playlist	播放列表
send_style	固定值为0 不能改变
free_size	剩余空间 目前无效
total_size	总容量 目前无效

说明： 发送节目单

函数： int _CALL_STD send_program(char* ip, unsigned short port, _TEXT_CHAR* user_name, _TEXT_CHAR* user_pwd, _TEXT_CHAR* tmp_path, unsigned long playlist, int send_style, long long * free_size, long long * total_size);

4.5 销毁节目单 delete_playlist

返回值： 成功返回0；失败返回错误号

参数：

参数	说明
playlist	节目单句柄

说明： 销毁节目单

函数： void _CALL_STD delete_playlist(unsigned long playlist);

4.6 取消发送节目 cancel_send_program

返回值： 成功返回0；失败返回错误号

参数：

参数	说明
playlist	节目单句柄

说明： 取消发送节目单

函数： void _CALL_STD cancel_send_program(unsigned long playlist);

4.7 创建图文分区 create_pic

返回值： 成功返回分区句柄

参数： 无

说明： 创建图文分区

函数： unsigned long _CALL_STD create_pic();

4.8 图文分区添加图片 add_pic_unit

返回值：

参数：

参数	说明
pic_area	图片分区句柄
stay_time	图片停留时间，以秒为单位
display_effects	特技编号
display_speed	特技速度等级，1~16，1 为最快
src_path	图片文件全路径

说明： 图文分区添加显示图片

函数： int _CALL_STD add_pic_unit(unsigned long pic_area, int stay_time, int display_effects, int display_speed, _TEXT_CHAR* src_path);

4.9 图文分区添加到节目 add_pic

返回值：

参数：

参数	说明
program	节目句柄
pic_area	图文区句柄
X	坐标起始X
Y	坐标起始Y
W	区域宽度
H	区域高度
transparency	分区透明度(0~100)：100 为完全不透明，默认 100

说明： 图文分区添加到节目

函数： int _CALL_STD add_pic(unsigned long program, unsigned long pic_area, int x, int y, int w, int h, int transparency);

4.10 创建视频分区 create_video

返回值： 成功返回句柄

参数： 无

说明： 创建视频分区

函数： unsigned long _CALL_STD create_video();

4.11 添加视频文件 add_video_unit

返回值：

参数：

参数	说明
video_area	src_path
volume	音量 0-100
scale_mode	缩放模式0-按原始比例进行缩放，1-按窗口比例进行缩放
source	输入视频源（包括"CVBS/0"、“HDMI/1”，此属性只对外部输入视频类型有效） 0
play_time	播放时长以秒为单位： 0 表示在节目时效内一直播放或由文件时长决定
src_path	视频绝对路径
crop_type	HDMI输出指定位置 可以给空字符串

说明： 添加视频文件

函数： int _CALL_STD add_video_unit(unsigned long video_area, int volume, int scale_mode, int source, int play_time, _TEXT_CHAR* src_path, _TEXT_CHAR* crop_type);

4.12 节目添加视频分区 add_video

返回值：

参数：

参数	说明
program	节目句柄
video_area	视频区句柄
X	坐标起始X
Y	坐标起始Y
W	区域宽度
H	区域高度
volume_mode	是否静音：‘Unmute’/0 - 非静音；‘Mute’/1 - 静音
video_type	‘local’/0 - 本地视频或网络流媒体 ‘capture’/1 - 外部输入视频(Y 系列暂不支持)
ratation_mode	逆时针旋转角度：仅支持 0/90/180/270 四种，默认 0
clone_str	“”
crop_type	HDMI输出指定位置 可以给空字符串

说明： 节目添加视频分区

函数： int _CALL_STD add_video(unsigned long program, unsigned long video_area, int x, int y, int w, int h, int volume_mode, int video_type, int ratation_mode, _TEXT_CHAR* clone_str, _TEXT_CHAR* crop_type);

4.13 创建时间分区 create_time

返回值： 成功返回句柄

参数： 无

说明： 创建时间分区

函数： unsigned long _CALL_STD create_time();

4.14 添加时间数据 add_time_unit

返回值： 成功返回0；失败返回错误号

参数：

参数	说明
time_area	时间分区句柄
content	显示内容
font_color	字体颜色
font_name	字体名称
font_size	字体大小
x	x坐标(相对窗口)
y	y坐标(相对窗口)
font_attributes	字体属性（包括“bold”、“italic”、“normal”，其中“bold”和“italic”可以通过“&”进行组合）

说明： 添加时间数据

函数： int _CALL_STD add_time_unit(unsigned long time_area, _TEXT_CHAR* content, _TEXT_CHAR* font_color, _TEXT_CHAR* font_name, int font_size, int x, int y, _TEXT_CHAR* font_attributes);

4.15 节目添加时间分区 add_time

返回值：

参数：

参数	说明
program	节目句柄
time_area	时间分区句柄
X	坐标起始X
Y	坐标起始Y
W	区域宽度
H	区域高度
transparency	分区透明度(0~100)：100 为完全不透明，默认 100
bg_color	背景颜色，默认’0x00000000’ - 透明黑色
time_equation	时差，格式“hh:mm:ss”
positive_te	正，负时差：”true“，”false“
adjustment	用以调整时差；支持天数，格式“±dd:hh:mm:ss”

说明： 节目添加时间分区

函数： int _CALL_STD add_time(unsigned long program, unsigned long time_area, int x, int y, int w, int h, int transparency, _TEXT_CHAR* bg_color, _TEXT_CHAR* time_equation, _TEXT_CHAR* positive_te, _TEXT_CHAR* adjustment);

4.16 创建表盘分区 create_clock

返回值：成功返回句柄

参数：无

说明： 创建表盘分区

函数：unsigned long _CALL_STD create_clock();

4.17 节目添加表盘分区 add_clock

返回值：

参数：

参数	说明
program	节目句柄
clock_area	表盘区域句柄
X	坐标起始X
Y	坐标起始Y
W	区域宽度
H	区域高度
transparency	分区透明度(0~100)：100 为完全不透明，默认 100
time_equation	时差，格式“hh:mm:ss”
positive_te	正，负时差：”true“，”false“
adjustment	用以调整时差；支持天数，格式“±dd:hh:mm:ss”
hour_color	时针颜色
minute_color	分针颜色
second_color	秒针颜色
bg_image	表盘图片路径

说明： 节目添加表盘分区

函数：int _CALL_STD add_clock(unsigned long program, unsigned long clock_area, int x, int y, int w, int h, int transparency, _TEXT_CHAR* time_equation, _TEXT_CHAR* positive_te, _TEXT_CHAR* adjustment, _TEXT_CHAR* hour_color, _TEXT_CHAR* minute_color, _TEXT_CHAR* second_color, _TEXT_CHAR* bg_image);

4.18 创建农历分区 create_calendar

返回值：成功返回句柄

参数：无

说明：创建农历分区

函数：unsigned long _CALL_STD create_calendar();

4.19 添加农历信息 add_calendar_unit

返回值：成功返回0；失败返回错误号

参数：

参数	说明
calendar_area	农历分区句柄
mode	显示类型包括"heavenlystem"(天干)、“lunarcalendar”(农历)、“solarterms”(节气)、“text”(固定文本)
font_color	字体颜色
font_name	字体名称
font_size	字体大小
X	x坐标(相对窗口)
Y	y坐标(相对窗口)
font_attributes	字体属性（包括“bold”、“italic”、“normal”，其中“bold”和“italic”可以通过“&”进行组合）
text_content	mode为"text"时的文本内容

说明： 添加农历信息

函数：int _CALL_STD add_calendar_unit(unsigned long calendar_area, _TEXT_CHAR* mode, _TEXT_CHAR* font_color, _TEXT_CHAR* font_name, int font_size, int x, int y, _TEXT_CHAR* font_attributes, _TEXT_CHAR* text_content);

4.20 节目添加农历分区 add_calendar

返回值：成功返回0；失败返回错误号

参数：

参数	说明
program	节目句柄
calendar_area	农历区句柄
X	坐标起始X
Y	坐标起始Y
W	区域宽度
H	区域高度
transparency	分区透明度(0~100)：100 为完全不透明，默认 100
bg_color	背景颜色
time_equation	时差，格式“hh:mm:ss”
positive_te	正，负时差：”true“，”false“
adjustment	用以调整时差；支持天数，格式“±dd:hh:mm:ss”

说明： 节目添加农历分区

函数： int _CALL_STD add_calendar(unsigned long program, unsigned long calendar_area, int x, int y, int w, int h, int transparency, _TEXT_CHAR* bg_color, _TEXT_CHAR* time_equation, _TEXT_CHAR* positive_t, _TEXT_CHAR* adjustment);

4.21 节目添加计时分区 add_count

返回值：

参数：

参数	说明
program	节目句柄
X	坐标起始X
Y	坐标起始Y
W	区域宽度
H	区域高度
transparency	分区透明度(0~100)：100 为完全不透明，默认 100
bg_color	背景颜色
time_equation	时差，格式“hh:mm:ss”
positive_te	正，负时差：”true“，”false“
target_date	目标日期,格式“yyyy-MM-dd”
target_time	目标时间,格式“hh:mm:ss”
content	显示内容
font_color	字体颜色
font_name	字体名称
font_size	字体大小
content_x	x坐标(相对窗口)
content_y	y坐标(相对窗口)
font_attributes	字体属性（包括“bold”、“italic”、“normal”，其中“bold”和“italic”可以通过“&”进行组合）
add_enable	是否显示“yes” “no"

说明： 节目添加计时分区

函数： int _CALL_STD add_count(unsigned long program, int x, int y, int w, int h, int transparency, _TEXT_CHAR* bg_color, _TEXT_CHAR* time_equation, _TEXT_CHAR* positive_te, _TEXT_CHAR* target_date, _TEXT_CHAR* target_time, _TEXT_CHAR* content, _TEXT_CHAR* font_color, _TEXT_CHAR* font_name, int font_size, int content_x, int content_y, _TEXT_CHAR* font_attributes, _TEXT_CHAR* add_enable);

4.22 清除所有节目 clear_all_program

返回值：成功返回节目单句柄；失败返回错误号

参数：

参数	说明
ip	控制卡IP
port	端口
user_name	登录名 "guest"
user_pwd	登录密码 "guest"

说明： 清除所有节目

函数： unsigned long _CALL_STD clear_all_program(char* ip, unsigned short port, _TEXT_CHAR* user_name, _TEXT_CHAR* user_pwd);

4.23 停止播放节目 stop_play_program

返回值： 成功返回节目单句柄；失败返回错误号

参数：

参数	说明
ip	控制卡IP
port	端口
user_name	登录名 "guest"
user_pwd	登录密码 "guest"

说明： 停止播放节目

函数： unsigned long _CALL_STD stop_play_program(char* ip, unsigned short port, _TEXT_CHAR* user_name, _TEXT_CHAR* user_pwd);

4.24 创建字幕分区 create_text

返回值： 成功返回分区句柄

参数： 无

说明： 创建字幕分区

函数： unsigned long _CALL_STD create_text();

4.25 字幕分区添加图片 add_text_unit_img

返回值： 成功返回0；失败返回错误号

参数：

参数	说明
text_area	字幕分区句柄
stay_time	图片停留时间，以秒为单位
display_speed	特技速度等级，1~16，1 为最快
last_move_width	整条字幕最后一张图的移动宽/高度 当多幅图时，不为最后一张，该值必须为 0 整条字幕只有一张图，该值固定为窗口宽/高度
src_path	图片文件全路径

说明： 字幕分区添加显示图片

函数： int _CALL_STD add_text_unit_img(unsigned long text_area, int stay_time, int display_speed, int last_move_width, _TEXT_CHAR* src_path);

4.26 字幕分区添加文本 add_text_unit_text

返回值： 成功返回0；失败返回错误号

参数：

参数	说明
text_area	字幕分区句柄
stay_time	图片停留时间，以秒为单位
display_speed	特技速度等级，1~16，1 为最快
font_name	字体名称
font_size	字体大小
font_attributes	字体附加属性： 包括'bold', 'italic', 'normal', 'underline', 'strikeout'等五 种; 可通过'&'组合,例如： 'bold& italic& underline'
font_alignment	待定
font_color	字体颜色，默认 '0xFFFFFFFF' - 不透明白色
bg_color	背景颜色，默认 '0x00000000'- 透明黑色
content	文本内容

说明： 字幕分区添加显示图片

函数： int _CALL_STD add_text_unit_text(unsigned long text_area, int stay_time, int display_speed, _TEXT_CHAR* font_name, int font_size, _TEXT_CHAR* font_attributes, _TEXT_CHAR* font_alignment, _TEXT_CHAR* font_color, _TEXT_CHAR* bg_color, _TEXT_CHAR* content);

4.27 字幕分区添加到节目 add_text

返回值：

参数：

参数	说明
program	节目句柄
text_area	字幕区句柄
X	坐标起始X
Y	坐标起始Y
W	区域宽度
H	区域高度
transparency	分区透明度(0~100)：100 为完全不透明，默认 100
display_effects	特技类型
unit_type	子单元类型选择： ‘image’/0 - 渲染好的文本图片，上位机渲染 ‘text’/1 - 未渲染的文字，控制卡自渲染

说明： 字幕分区添加到节目

函数： int _CALL_STD add_text(unsigned long program, unsigned long text_area, int x, int y, int w, int h, int transparency, int display_effects, int unit_type);

4.28 创建炫彩字分区 create_colortext

返回值： 成功返回分区句柄

参数： 无

说明： 创建炫彩字分区

函数： unsigned long _CALL_STD create_colortext();

4.29 炫彩分区添加显示内容 add_colorful_fontunit

返回值：

参数：

参数	说明
colorful_subtitle_area	炫彩字分区句柄
file	文字图片路径
display_effects	显示特技，附录8
display_speed	特技速度，1-16，1最快
stay_time	特技停留时间，单位秒
wave_effects	叠加波动效果类型索引（不在以下范围当‘0’处理）， 0 - 无 58 - 水平静止波； 59 - 水平移动波60 - 垂直静止波； 61 - 垂直移动波当文本掩模移动（50~53 号特技）时，波动方向须与其方向一致，否则用 0 替代，当文本掩模静止（0 号特技）时，无此限制
wave_count	波峰数目： 最小个数为 1，最大个数与波动方向的像素数相关。水平波取决于分区宽度，垂直波取决于分区高度。小于 200 时，为不超过 N/12 的最大偶数；大于 1000 时，为不超过 N/60 的最大偶数；其他为 16
wave_speed	波动速度等级，1~16；移动波时有效，1 为最快
wave_amplitude	波峰幅度占比：波峰幅度在分区宽度（水平波）或分区高度（垂直波）中的所占的千分比，1~250

说明： 炫彩分区添加显示内容

函数： int _CALL_STD add_colorful_fontunit(unsigned long colorful_subtitle_area, _TEXT_CHAR* file, int display_effects, int display_speed, int stay_time, int wave_effects, int wave_count, int wave_speed, int wave_amplitude);

4.30 炫彩分区添加背景 add_colorful_hollowunit

返回值：

参数：

参数	说明
colorful_subtitle_area	炫彩字分区句柄
display_effects	字芯特技，只支持 0、3~6，这 5 种特技类型，其他类型索引会使用 0 替代
display_speed	字芯特技速度等级，1~16，默认为 1 最快
stay_time	单元停留时间，以秒为单位
file	字芯图片路径

说明： 添加炫彩字分区字芯项

函数： int _CALL_STD add_colorful_hollowunit(unsigned long colorful_subtitle_area, int display_effects, int display_speed, int stay_time, _TEXT_CHAR* hollow_file);

4.30 炫彩分区添加到节目 add_colorful_subtitle

返回值：

参数：

参数	说明
program	节目句柄
colorful_subtitle_area	炫彩字区分区句柄
x	分区x坐标
y	分区y坐标
w	分区宽度
h	分区高度

说明：炫彩分区添加到节

函数：int _CALL_STD add_colorful_subtitle(unsigned long program, unsigned long colorful_subtitle_area, int x, int y, int w, int h);

4.31 创建富文本分区 create_rich_text

返回值：成功返回分区句柄

参数：无

说明： 创建富文本分区

函数：unsigned long _CALL_STD create_rich_text();

4.32 富文本分区添加显示内容 add_rich_text_unit

返回值：

参数：

参数	说明
rich_text_area	富文本分区句柄
stay_time	特技停留时间，以秒为单位
display_speed	特技速度等级，1~16,1为最快
bg_image_path	背景图片文件路径，上位机需要将背景图片和节目文件一起下发，该目录为控制卡存储背景图片的目录。
bg_color	背景色
content	文本内容；需要显示的内容和文字属性，如字体类型、颜色、大小、行间距，如：上海仰邦 格式参考链接：https://developer.gnome.org/pango/stable/pango-Markup.html

说明： 富文本分区添加显示内容

函数： int _CALL_STD add_rich_text_unit(unsigned long rich_text_area, int stay_time, int display_speed, _TEXT_CHAR* bg_image_path, _TEXT_CHAR* bg_color, _TEXT_CHAR* content);

4.33 富文本分区添加到节目 add_rich_text

返回值：

参数：

参数	说明
long program	节目句柄
text_area	富文本区分区句柄
x	分区x坐标
y	分区y坐标
w	分区宽度
h	分区高度
transparency	分区透明度(0~100)：100 为完全不透明，默认100
display_effects	显示特技;建议只使用0，50，51这三种特技类型
alignment_h	居左0，居中2，居右1

说明： 富文本分区添加到节目

函数： int _CALL_STD add_rich_text(unsigned long program, unsigned long rich_text_area, int x, int y, int w, int h, int transparency, int display_effects, int alignment_h);

4.34 销毁图文区句柄 delete_pic

返回值：

参数：

参数	说明
area_tree	图文区句柄

说明： 销毁图文区句柄

函数： void _CALL_STD delete_pic(unsigned long area_tree);

4.35 销毁视频区句柄 delete_video

返回值：

参数：

参数	说明
area_tree	视频分区句柄

说明： 销毁视频分区

函数： unsigned long _CALL_STD delete_video(area_tree);

4.36 销毁时间区句柄 delete_time

返回值：

参数：

参数	说明
area_tree	时间分区句柄

说明： 销毁时间分区

函数： void _CALL_STD delete_time(unsigned long area_tree);

4.37 销毁表盘区句柄 delete_calendar

返回值：

参数：

参数	说明
area_tree	表盘分区句柄

说明： 销毁表盘分区

函数： void _CALL_STD delete_calendar(unsigned long area_tree);

4.38 销毁字幕区句柄 delete_text

返回值：

参数：

参数	说明
area_tree	字幕分区句柄

说明： 销毁字幕分区

函数： void _CALL_STD delete_text(unsigned long area_tree);

4.39 销毁炫彩区句柄 delete_colortext

返回值：

参数：

参数	说明
area_tree	炫彩字幕分区句柄

说明： 销毁炫彩字幕分区

函数： void _CALL_STD delete_colortext(unsigned long area_tree);

4.40 销毁富文本区句柄 delete_rich_text

返回值：

参数：

参数	说明
area_tree	富文本分区句柄

说明： 销毁富文本分区

函数：void _CALL_STD delete_rich_text(unsigned long area_tree);

5.动态区API

5.1 创建动态区分区 create_dynamic

返回值：成功返回句柄

参数：无

说明：创建动态区分区

函数：unsigned long _CALL_STD create_dynamic();

5.2 添加动态区文件 add_dynamic_unit

返回值：成功返回0；失败返回错误号

参数：

参数	说明
dynamic_area	动态区句柄
dynamic_type	动态单元类型 图片0 txt/文本1
display_effects	特技类型
display_speed	特技速度等级，1~16，1 为最快
stay_time	特技停留时间
file_path	元素绝对路径；文本类的支持txt（utf-8）格式；字符串要使用Base64编码格式；JSON链接
gif_flag	0非GIF；1GIF类型（暂不支持动态播放，作为普通图片播放）
bg_color	背景颜色
font_size	字体大小
font_name	字体
font_color	字体颜色
font_attributes	包括“bold”、“italic”、“normal”，其中“bold”和“italic”可以通过“&”进行组合
align_h	水平对齐方式0居左/1居右/2居中
align_v	垂直对齐方式0居上/1居下/2居中 以下三个为视频属性，暂不支持视频
volumn	音量
scale_mode	缩放模式，窗口比例0/原始比例1
rolation_mode	旋转角度
key_list	类型为"URLText"，且获取网络资源为json格式数据时有效。获取数据的字段，base64加密
proxyService	动态区"URLText"类型，网络资源为代理服务时，该参数有效，且该参数是代理服务的标识参数（存在且非空）

说明： 添加动态区文件

函数： int CALL_STD add_dynamic_unit(unsigned long dynamic_area, int dynamic_type, int display_effects, int display_speed, int stay_time,TEXT_CHAR* file_path, int gif_flag, _TEXT_CHAR* bg_color, int font_size, _TEXT_CHAR* font_name, _TEXT_CHAR* font_color, _TEXT_CHAR* font_attributes, _TEXT_CHAR* align_h, _TEXT_CHAR* align_v, int volumn, int scale_mode, int rolation_mode, _TEXT_CHAR* key_list, _TEXT_CHAR* proxyService);

5.3 节目添加动态分区 add_dynamic

返回值： 成功返回0；失败返回错误号

参数：

参数	说明
program	
dynamic_area	动态区句柄
dynamic_id	动态区编号0-31
X	坐标起始X
Y	坐标起始Y
W	区域宽度
H	区域高度
relative_program	关联的节目，即所要关联的节目序号(节目列表中的 order 字段(“0”,“1”,...))
run_mode	动态区运行方式： "0"：全局播放动态区，所有 unit 循环显示 "1"：全局动态区节目，所有 unit 顺序显示，显示完一遍后停止播放 该动态区节目 "2"：全局动态区节目，所有 unit 顺序显示，显示完一遍后静止显示 该动态区最后一个 unit(以 unit 的 order 为序) "3"：绑定播放动态区，关联节目开始播放时播放动态区，所有 unit 按 order 排序循环显示，直到关联节目播放完毕 "4"：绑定播放动态区，关联节目开始播放时播放动态区，所有 unit 顺序显示，显示完一遍后本轮不再显示 "5"：绑定播放动态区，关联节目开始播放时播放动态区，所有 unit 顺序显示，显示完一遍后静止显示该动态区最后一个 unit "6"：插播动态区，关联节目播放完后播放动态区，所有 unit 顺序 显示一遍后本轮不再显示 ，继续播放关联节目后的节目 注意： Ⅰ：未关联任何普通节目时，该字段只能取 0、1、2，指定关联节目时，该字段只能取 3、4、5、6，越界取值无效 Ⅱ：3、4、5、6 的运行模式用于动态区关联普通节目时，动态区的 播放方式，动态区参与关联节目的轮
update_frequency	放空""即可
transparency	分区透明度(0~100)：100 为完全不透明，默认100

说明： 节目添加动态分区

函数： int _CALL_STD add_dynamic(unsigned long program, unsigned long dynamic_area, int dynamic_id, int x, int y, int w, int h, int relative_program, int run_mode, _TEXT_CHAR* update_frequency, int transparency);

5.4 销毁动态分区句柄 delete_dynamic

返回值： 成功返回0；失败返回错误号

参数：

参数	说明
dynamic_area	动态分区句柄

说明： 销毁动态分区句柄

函数： void _CALL_STD delete_dynamic(unsigned long dynamic_area);

5.5 更新动态区-图文文件 update_dynamic

返回值：成功返回0；失败返回错误号

参数：

参数	说明
ip	控制卡IP
port	端口
user_name	登录名 “guest”
user_pwd	登录密码 “guest”
playlist	播放列表
immediately_play	指定一个关联了普通节目的动态区 ID（必须是 dynamics 参数中存在的 id）
conver	是否覆盖普通节目，即是否只播放动态区，“0”：动态区和普通节目共存播放，“1”：停止播放普通节 目，只播放动态区节目
onlyUpdate	是否只更新动态区；只更新动态区"0"；清空原来的动态区后再更新动态区"1" 目前无效，固定为0

说明： 更新动态区-图文文件

函数：int _CALL_STD update_dynamic(char* ip, unsigned short port, _TEXT_CHAR* user_name, _TEXT_CHAR* user_pwd, unsigned long playlist, int immediately_play, int conver, int onlyUpdate);

5.6 更新动态区-文本 update_dynamic_small

返回值：成功返回0；失败返回错误号

参数：

参数	说明
ip	控制卡IP
port	端口
user_name	登录名 “guest”
user_pwd	登录密码 “guest”
playlist	播放列表
immediately_play	指定一个关联了普通节目的动态区 ID（必须是 dynamics 参数中存在的 id）
conver	是否覆盖普通节目，即是否只播放动态区，“0”：动态区和普通节目共存播放，“1”：停止播放普通节 目，只播放动态区节目
onlyUpdate	是否只更新动态区；只更新动态区"0"；清空原来的动态区后再更新动态区"1" 目前无效，固定为0

说明： 更新动态区-文本

函数： int _CALL_STD update_dynamic(char* ip, unsigned short port, _TEXT_CHAR* user_name, _TEXT_CHAR* user_pwd, unsigned long playlist, int immediately_play, int conver, int onlyUpdate);

5.7 更新动态区-图文素材 update_dynamic_unit

返回值： 成功返回0；失败返回错误号

参数：

参数	说明
ip	控制卡IP
port	端口
user_name	登录名 “guest”
user_pwd	登录密码 “guest”
playlist	播放列表

说明： 更新动态区-图文素材，显示屏上有动态区时使用，素材类型不能变动，图片换图片，txt换txt

函数： int _CALL_STD update_dynamic_unit(char* ip, unsigned short port, _TEXT_CHAR* user_name, _TEXT_CHAR* user_pwd, unsigned long playlist);

5.8 更新动态区-文本素材 update_dynamic_unit_small

返回值： 成功返回0；失败返回错误号

参数：

参数	说明
ip	控制卡IP
port	端口
user_name	登录名 “guest”
user_pwd	登录密码 “guest”
playlist	播放列表

说明： 更新动态区-文本素材，显示屏上有动态区时使用

函数： int _CALL_STD update_dynamic_unit_small(char* ip, unsigned short port, _TEXT_CHAR* user_name, _TEXT_CHAR* user_pwd, unsigned long playlist);

5.9 删除所有动态区 clear_dynamic

返回值： 成功返回0；失败返回错误号

参数：

参数	说明
ip	控制卡IP
port	端口
user_name	登录名 “guest”
user_pwd	登录密码 “guest”

说明： 删除所有动态区

函数： int _CALL_STD clear_dynamic(char* ip, unsigned short port, _TEXT_CHAR* user_name, _TEXT_CHAR* user_pwd);

6.屏幕状态API

6.1 查询固件版本 get_firmware_version

返回值： 成功返回0；失败返回错误号

参数： 无

参数	说明
ip	控制卡IP
port	端口
user_name	登录名 “guest”
user_pwd	登录密码 “guest”
firmware_version	程序版本号
app_version	APP版本号
fpga_version	fpga版本号

说明： 查询固件版本

函数： int _CALL_STD get_firmware_version(char* ip, unsigned short port, _TEXT_CHAR* user_name, _TEXT_CHAR* user_pwd, _TEXT_CHAR* firmware_version, _TEXT_CHAR* app_version, _TEXT_CHAR* fpga_version);

6.2 校时命令 check_time

返回值： 成功返回0；失败返回错误号

参数：

参数	说明
ip	控制卡IP
port	端口
user_name	登录名 “guest”
user_pwd	登录密码 “guest”

说明： 校时命令

函数： int _CALL_STD check_time(char* ip, unsigned short port, _TEXT_CHAR* user_name, _TEXT_CHAR* user_pwd);

6.3 取得屏幕参数 get_screen_parameters

返回值： 成功返回0；失败返回错误号

参数：

参数	说明
ip	控制卡IP
port	端口
user_name	登录名 “guest”
user_pwd	登录密码 “guest”
cards	屏幕参数缓存空间，ControllerInfo

说明： 取得屏幕参数

函数： int _CALL_STD get_screen_parameters(char* ip, unsigned short port, _TEXT_CHAR* user_name, _TEXT_CHAR* user_pwd, BYTE* cards);

6.4 节目解锁/锁定 lock_program

返回值： 成功返回0；失败返回错误号

参数：

参数	说明
ip	控制卡IP
port	端口
user_name	登录名 “guest”
user_pwd	登录密码 “guest”
flag	锁定：1 解锁：0
program_name	节目名

说明： 节目解锁/锁定

函数： int _CALL_STD lock_program(char* ip, unsigned short port, _TEXT_CHAR* user_name, _TEXT_CHAR* user_pwd, int flag, _TEXT_CHAR* program_name);

6.5 格式化 reboot

返回值： 成功返回0；失败返回错误号

参数：

参数	说明
ip	控制卡IP
port	端口
user_name	登录名 “guest”
user_pwd	登录密码 “guest”

说明： 格式化

函数： int _CALL_STD reboot(char* ip, unsigned short port, _TEXT_CHAR* user_name, _TEXT_CHAR* user_pwd);

6.6 设置系统音量 set_screen_volumn

返回值： 成功返回0；失败返回错误号

参数：

参数	说明
ip	控制卡IP
port	端口
user_name	登录名 “guest”
user_pwd	登录密码 “guest”
volumn	音量0-100

说明： 设置系统音量

函数： int _CALL_STD set_screen_volumn(char* ip, unsigned short port, _TEXT_CHAR* user_name, _TEXT_CHAR* user_pwd, int volumn);

6.7 手动调亮 set_screen_brightness

返回值： 成功返回0；失败返回错误号

参数：

参数	说明
ip	控制卡IP
port	端口
user_name	登录名 “guest”
user_pwd	登录密码 “guest”
brightness	亮度值 0-255

说明： 手动调亮

函数： int _CALL_STD set_screen_brightness(char* ip, unsigned short port, _TEXT_CHAR* user_name, _TEXT_CHAR* user_pwd, int brightness);

6.8 定时调亮 set_screen_cus_brightness

返回值： 成功返回0；失败返回错误号

参数： 无

参数	说明
ip	控制卡IP
port	端口
user_name	登录名 “guest”
user_pwd	登录密码 “guest”
brightness	48组时间00:00:00-00:29:59 00:30:00-00:59:59…23:30:00-23:59:59 亮度值
data_count	固定值48 不能改变

说明： 定时调亮

函数： int _CALL_STD set_screen_cus_brightness(char* ip, unsigned short port, _TEXT_CHAR* user_name, _TEXT_CHAR* user_pwd, unsigned short* brightness, int data_count);

6.9 手动设置开关机 set_screen_turnonoff

返回值： 成功返回0；失败返回错误号

参数：

参数	说明
ip	控制卡IP
port	端口
user_name	登录名 “guest”
user_pwd	登录密码 “guest”
turnonoff_status	开关机状态(1:开机 0:关机)

说明： 手动设置开关机

函数：int _CALL_STD set_screen_turnonoff(char* ip, unsigned short port, _TEXT_CHAR* user_name, _TEXT_CHAR* user_pwd, int turnonoff_status);

6.10 创建定时开关机分区 create_turnonoff

返回值：成功返回句柄

参数：无

说明： 创建定时开关机分区

函数：unsigned long _CALL_STD create_turnonoff();

6.11 添加定时开关机时间 add_turnonoff

返回值：成功返回0；失败返回错误号

参数：

参数	说明
turnonoff	开关机句柄
action	开关机(1:开机 0:关机)
time	时间“hh:mm:ss”

说明： 添加定时开关机时间，一组：一个开机时间，一个关机时间

函数：void _CALL_STD add_turnonoff(unsigned long turnonoff, int action, _TEXT_CHAR* time);

6.12 设置定时开关机set_screen_cus_turnonoff

返回值：成功返回0；失败返回错误号

参数：

参数	说明
ip	控制卡IP
port	端口
user_name	登录名 “guest”
user_pwd	登录密码 “guest”
turnonoff	定时开关机分区句柄

说明： 设置定时开关机

函数：int _CALL_STD set_screen_cus_turnonoff(char* ip, unsigned short port, _TEXT_CHAR* user_name, _TEXT_CHAR* user_pwd, unsigned long turnonoff);

6.13 销毁句柄delete_turnonoff

返回值：成功返回0；失败返回错误号

参数：

参数	说明
turnonoff	定时开关机分区句柄

说明： 销毁句柄

函数：void _CALL_STD delete_turnonoff(unsigned long turnonoff);

6.14 取消定时开关机 cancel_screen_cus_turnonoff

返回值：成功返回0；失败返回错误号

参数：

参数	说明
ip	控制卡IP
port	端口
user_name	登录名 “guest”
user_pwd	登录密码 “guest”

说明： 取消定时开关机,保留取消时屏幕状态

函数：int _CALL_STD cancel_screen_cus_turnonoff(char* ip, unsigned short port, _TEXT_CHAR* user_name, _TEXT_CHAR* user_pwd);

6.15 设置屏幕大小 set_screen_size

返回值：成功返回0；失败返回错误号

参数：

参数	说明
ip	控制卡IP
port	端口
user_name	登录名 “guest”
user_pwd	登录密码 “guest”
screen_w	屏幕宽度
screen_h	屏幕高度
screenrotation	0

说明： 设置屏幕大小

函数：int _CALL_STD set_screen_size(char* ip, unsigned short port, _TEXT_CHAR* user_name, _TEXT_CHAR* user_pwd, int screen_w, int screen_h, int screenrotation);

7.功能API

7.1 屏幕截取 get_screen_capture

返回值：成功返回0；失败返回错误号

参数：

参数	说明
ip	控制卡IP
port	端口
user_name	登录名 “guest”
user_pwd	登录密码 “guest”
file_path	截屏保存的文件 例：“D:\1.png" "F:\1.bmp" gif暂不支持
captureW	截取后图片的宽度
captureH	截取后图片的高度，如果宽高与屏幕宽高不一样，将会做拉伸或缩小处理

说明： 屏幕截取

函数： int _CALL_STD get_screen_capture(char* ip, unsigned short port, _TEXT_CHAR* user_name, _TEXT_CHAR* user_pwd, _TEXT_CHAR* file_path, int captureW, int captureH);

5.9 清除素材 clear_material

返回值：成功返回0；失败返回错误号

参数：

参数	说明
ip	控制卡IP
port	端口
user_name	登录名 “guest”
user_pwd	登录密码 “guest”

说明： 清除素材

函数：int _CALL_STD clear_material(char* ip, unsigned short port, _TEXT_CHAR* user_name, _TEXT_CHAR* user_pwd);

附录

1.字体名称规范

- 中文：楷体-'KaiTi'; 黑体-'SimHei'; 宋体-'SimSun'; 新宋体-'NSimSun';
- 西文：'Arial'; 'Times New Roman'; 'Tahoma'; 'Georgia'; 'Verdana';

2.字体大小规范

字体采用的是矢量字体，大小在理论上没有限制。实际使用中，肯定不能大于对应分区的宽度或高度；也不能太小，否则显示效果不好

建议：一般纯西文字体最小不小于8pixel，中文不小于16pixel。
节目文件中字体大小参数有两个：

- fontSizeType指定单位类型0 - pixel，1 - point
- fontSize 指定大小。point为单位的会比pixel为单位的字大

3.背景颜色和字体颜色规范

没有特殊说明，所有与颜色相关的字段，均以以 0xAARRGGBB 字符串格式给出

4.控制卡型号

- BX-Y04 = 8280
- BX-Y08 = 8536
- BX-Y2 = 8792
- BX-Y2L = 9304
- BX-Y3 = 9048
- BX-Y5E = 10584
- BX-Y1 = 9560
- BX-Y3X =9816
- BX-Y1L =10072
- BX-Y5E/OVP-Y5E =10584
- BX-YL5 = 10840
- BX-Y1A = 11608
- BX-Y08A = 11352
- BX-Y3A = 10328
- BX-Y3E = 11096

5.在线搜索命令返回数据解析格式

```
typedef struct ControllerData
{
    int is_dhcp;
    int wifi_is_dhcp;
    int output_type;   //输出方式 LCD/DVI
    int port;  //端口
    int screen_w;  //屏幕宽度
    int screen_h;  //屏幕高度
    int screen_volume;  //音量
    int screen_brigtness;  //亮度值
    int screen_brigtness_mode;  //亮度模式
    int screen_rotation;
    unsigned short screen_type;  //屏幕类型
    _TEXT_CHAR storagemedia[10];  //存储介质：emmc,sd,usb1
    _TEXT_CHAR barcode[17];  //条形码
    _TEXT_CHAR ip[20];        //以太网卡 IP 地址
    _TEXT_CHAR source_ip[20];  //控制卡IP
    _TEXT_CHAR sub_mark[20];   //以太网卡子网掩码
    _TEXT_CHAR gateway[20];    //以太网卡网关
    _TEXT_CHAR mac[20];//MAC地址
    _TEXT_CHAR local_ip[20];
    _TEXT_CHAR pid[33];  //PID码
    _TEXT_CHAR local_net[64];
    _TEXT_CHAR ap_wifi_ip[20];
    _TEXT_CHAR wifi_ip[20];      //WiFi IP 地址
    _TEXT_CHAR wifi_sub_mark[20];  //WiFi 子网掩码
```

```
    _TEXT_CHAR wifi_gateway[20];   //WiFi 网关
    _TEXT_CHAR dns_server[20];  //域名服务器地址
    _TEXT_CHAR network_device[20];
    _TEXT_CHAR controller_name[128];
}BroadCast2;
```

6.屏参查询返回数据解析格式

```
typedef struct ControllerInfo
{
    int is_dhcp;
    int wifi_is_dhcp;
    int output_type;
    int port;
    int screen_w;
    int screen_h;
    int screen_volume;
    int screen_brigtness;
    int screen_brigtness_mode;
    int fold_type;
    int fold_count;
    int logic_width;
    int logic_height;
    int screen_rotation;
    unsigned short screen_type;
    _TEXT_CHAR storagemedia[10];
    _TEXT_CHAR barcode[17];
    _TEXT_CHAR ip[20];
    _TEXT_CHAR source_ip[20];
    _TEXT_CHAR sub_mark[20];
    _TEXT_CHAR gateway[20];
    _TEXT_CHAR mac[20];
    _TEXT_CHAR local_ip[20];
    _TEXT_CHAR pid[33];
    _TEXT_CHAR local_net[64];
    _TEXT_CHAR ap_wifi_ip[20];
    _TEXT_CHAR wifi_ip[20];
    _TEXT_CHAR wifi_sub_mark[20];
    _TEXT_CHAR wifi_gateway[20];
    _TEXT_CHAR dns_server[20];
    _TEXT_CHAR network_device[20];
    _TEXT_CHAR controller_name[128];
    _TEXT_CHAR fold_width[128];
    _TEXT_CHAR fold_height[128];
}controllerInfo;
```

7.错误码列表

错误码	定义	错误描述
1	ERR_HTTP_REQUEST_EMPTY HTTP	请求参数为空
2	ERR_HTTP_REQUEST_METHOD HTTP	请求方法错误
3	ERR_PROTOCOL_PARSE	协议解析错误
4	ERR_PROTOCOL_NAME	协议名错误
5	ERR_PROTOCOL_VERSION	协议版本错误
6	ERR_PID PID	错误
7	ERR_BARCODE	控制器 barcode 错误
8	ERR_HTTP_REQUEST_PARAMETER_KEY	协议请求报文键错误
9	ERR_CONFIG_PARSE	配置文件解析错误
10	ERR_PERMISSION	权限不够
11	ERR_INVALID_AUTHENTICATION	用户认证失效
12	ERR_ACCESS_VIOLATION	非法访问
13	ERR_IO_READ_WRITE	输入输出操作错误
14	ERR_COMMAND_PARAMETER_KEY	请求命令参数错误
15	ERR_COMMAND_CALL	请求命令调用错误
16	ERR_COMMAND_PROCESS	请求命令处理错误
17	ERR_COMMAND_NOT_EXISTS	请求命令不存在
18	ERR_COMMAND_PARAMETER_EMPTY	请求命令参数为空
19	ERR_COMMAND_EXECUTE	系统命令执行错误
20	ERR_COMMAND_PARAMETER_VALUE	请求命令参数值错误
21	ERR_USER_NOT_EXISTS	用户不存在
22	ERR_USER_PASSWORD	密码错误
23	ERR_STORAGE_MEDIA_NOT_EXISTS	媒体存储介质不存在
24	ERR_FILE_PATH	文件路径错误
25	ERR_MAC_FORMAT MAC	地址格式错误
26	ERR_UDP_TRANSMIT UDP	转发错误
27	ERR_VERIFICATION_CODE	验证码错误

错误码	定义	错误描述
28	ERR_NO_FIRMWARE	固件不存在
29	ERR_USER_WORK_PATH	用户工作目录创建失败
30	ERR_PLAYER_CMD	播放器执行指令出错
31	ERR_GET_WIFI_LIST	获取热点列表失败
32	ERR_WIFI_CONNECT_TIMEOUT	热点连接超时
33	ERR_HOTSPOT_NOT_FOUND	热点未找到
34	ERR_WIFI_PASSWORD	热点密码错误
35	ERR_NETWORK_RESTART	网络正在重启中
255	ERR_OTHER	其他错误

8. 图/文区特技列表

类型索引	特技描述
0	快速打出
1	随机特技
2	静止显示
3	向上推入
4	向下推入
5	向左推入
6	向右推入
7	向下移入
8	向上移入
9	向右移入
10	向左移入
11	向上堆积
12	向下堆积
13	向左堆积
14	向右堆积
15	向上拉幕
16	向下拉幕
17	向左拉幕
18	向右拉幕
19	左上角拉幕
20	右上角拉幕
21	右下角拉幕
22	右下角拉幕
23	四周往中心拉幕（矩形）
24	四角往中心拉幕（十字）
25	中心往四角拉幕（十字）
26	左右交叉拉幕
27	上下交叉拉幕
28	垂直百叶拉幕
29	水平百叶拉幕

类型索引	特技描述
30	上下闭合
31	上下对开
32	左右对开
33	左右闭合
34	中心放大
35	马赛克
36	淡入淡出
38	条式擦除
39	中心往四周拉幕（矩形）
40	中心缩小
41	中心缩小（拖尾）
42	向左拉伸
43	向右拉伸
50	平滑向上推入
51	平滑向下推入
52	平滑向左推入
53	平滑向右推入
54	平滑向下移入
55	平滑向上移入
56	平滑向右移入
57	平滑向左移入