

bxDual_InitSdk

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：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_InitSdk();
```

bxDual_ReleaseSdk

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：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern void bxDual_ReleaseSdk();
```

bxDual_set_screenNum_G56

返回值：成功返回0；失败返回错误号

参数：

参数	说明
usDstAddr	屏号，2个字节长度，默认值0xfffe 为地址通配符

说明：设置目标地址，即设置屏号/设置屏地址/设置控制器的屏号

函数：

```
[DllImport("bx_sdk_dual_server.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_set_screenNum_G56(ushort usDstAddr);
```

bxDual_set_packetLen

返回值：成功返回0；失败返回错误号

参数：

参数	说明
packetLen	数据包长度 注：5E, 6E, 6Q系列最大数据长度64K（建议最大不要超过63*1024） 其他系列最大长度1K（1204）

说明： 用于设置控制各种通讯方式每一包最大长度

函数：

```
[DllImport("bx_sdk_dual_server.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_set_packetLen(ushort packetLen);
```

bxDual_cmd_searchController

返回值： 成功返回0；失败返回错误号

参数：

参数	说明
retData	控制卡相关参数数据 Ping_data

说明： 搜索控制器命令

函数：

```
[DllImport("bx_sdk_dual_server.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_cmd_searchController(ref Ping_data retData);
```

bxDual_cmd_uart_searchController

返回值： 成功返回0；失败返回错误号

参数：

参数	说明
retData	控制卡相关参数数据 Ping_data
uartPort	串口号，例如“COM1”

说明： 搜索控制器命令-串口

函数：

```
[DllImport("bx_sdk_dual_server.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_cmd_uart_searchController(ref Ping_data retData,
byte[] uartPort);
```

bxDual_cmd_uart_search_Net_6G【】

返回值：成功返回0；失败返回错误号

参数：

参数	说明
retData	NetSearchCmdRet
uartPort	串口号,如: "COM3"
nBaudRateType	1: 9600; 2: 57600;

说明：网络搜索命令，返回：温度传感器，空气，PM2.5等信息，详见 NetSearchCmdRet:参考结构体声明中的注释；

函数：

```
[DllImport("bx_sdk_dual_server.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_cmd_uart_search_Net_6G(ref NetSearchCmdRet
retData, byte[] uartPort, ushort nBaudRateType);
[DllImport("bx_sdk_dual_server.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_cmd_uart_search_Net_6G_web(ref
NetSearchCmdRet_web retData, byte[] uartPort, ushort nBaudRateType);
```

bxDual_cmd_uart_ofsFormat

返回值：成功返回0；失败返回错误号

参数：

参数	说明
uartPort	串口号,如: "COM3"
baudRate	1: 9600; 2: 57600;

说明：文件系统格式化

函数：

```
[DllImport("bx_sdk_dual_server.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_cmd_uart_ofsFormat(byte[] uartPort, byte
baudRate);
```

bxDual_cmd_uart_ofsStartFileTransf

返回值：成功返回0；失败返回错误号

参数：

参数	说明
uartPort	串口号,如: "COM3"
baudRate	1: 9600; 2: 57600;

说明： 开始批量写文件----串口

函数：

```
[DllImport("bx_sdk_dual_server.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_cmd_uart_ofsStartFileTransf(byte[] uartPort, byte
baudRate);
```

bxDual_cmd_uart_ofsEndFileTransf

返回值： 成功返回0；失败返回错误号

参数：

参数	说明
uartPort	串口号,如: "COM3"
baudRate	1: 9600; 2: 57600;

说明： 写文件结束----串口

函数：

```
[DllImport("bx_sdk_dual_server.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_cmd_uart_ofsEndFileTransf(byte[] uartPort, byte
baudRate);
```

bxDual_cmd_uart_ofsDeleteFormatFile

返回值： 成功返回0；失败返回错误号

参数：

参数	说明
uartPort	串口号,如: "COM3"
baudRate	1: 9600; 2: 57600;
fileNub	要删除的文件个数
fileName	要删除的文件名

说明： 删除文件----串口

函数：

```
[DllImport("bx_sdk_dual_server.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_cmd_uart_ofsDeleteFormatFile(byte[] uartPort,
byte baudRate, short fileNub, byte[] fileName);
```

bxDual_cmd_uart_confDeleteFormatFile 【】

返回值：成功返回0；失败返回错误号

参数：

参数	说明
uartPort	串口号,如: "COM3"
baudRate	1: 9600; 2: 57600;
fileNub	要删除的文件个数
fileName	要删除的文件名

说明：删除文件---串口

- 注：此函数用于对存储在固定位置的文件进行处理，例如：Firmware 文件、控制器参数配置文件、扫描配置文件等。

函数：

```
[DllImport("bx_sdk_dual_server.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_cmd_uart_confDeleteFormatFile(byte[] uartPort,
byte baudRate, short fileNub, byte[] fileName);
```

bxDual_cmd_uart_ofsGetMemoryVolume

返回值：成功返回0；失败返回错误号

参数：

参数	说明
uartPort	串口号,如: "COM3"
baudRate	1: 9600; 2: 57600;
totalMemVolume	全部空间大小
availableMemVolume	剩余空间大小

说明：获取控制空间大小和剩余空间---串口

函数：

```
[DllImport("bx_sdk_dual_server.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_cmd_uart_ofsGetMemoryVolume(byte[] uartPort, byte
baudRate, int[] totalMemVolume, int[] availableMemVolume);
```

bxDual_cmd_uart_ofsWriteFile 【】

返回值：成功返回0；失败返回错误号

参数：

参数	说明
uartPort	串口号,如: "COM3"
baudRate	1: 9600; 2: 57600;
fileName	文件名
fileType	文件类型
fileLen	文件长度
overwrite	是否覆盖控制上的文件 1覆盖 0不覆盖 建议发1
fileAddre	文件所在的缓存地址

说明：写文件到控制----串口

* 注：用于对存储在 OFS 中的文件的处理，例如：节目文件，字库文件、播放列表文件等

函数：

```
[DllImport("bx_sdk_dual_server.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_cmd_uart_ofsWriteFile(byte[] uartPort, byte
baudRate, byte[] fileName, byte fileType, uint fileLen, byte overwrite, IntPtr
fileAddre);
```

bxDual_cmd_uart_confWriteFile 【】

返回值：成功返回0；失败返回错误号

参数：

参数	说明
uartPort	串口号,如: "COM3"
baudRate	1: 9600; 2: 57600;
fileName	文件名
fileType	文件类型
fileLen	文件长度
overwrite	是否覆盖控制上的文件 1覆盖 0不覆盖 建议发1
fileAddre	文件所在的缓存地址

说明: 写文件到控制

* 注: 此函数用于对存储在固定位置的文件进行处理, 例如: Firmware 文件、控制器参数配置文件、扫描配置文件等。

函数:

```
[DllImport("bx_sdk_dual_server.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_cmd_uart_confwriteFile(byte[] uartPort, byte
baudRate, byte[] fileName, byte fileType, int fileLen, byte overwrite, byte[] fileAddre);
```

bxDual_cmd_uart_ofsStartReedFile

返回值: 成功返回0; 失败返回错误号

参数:

参数	说明
uartPort	串口号,如: "COM3"
baudRate	1: 9600; 2: 57600;
fileName	需要读取的文件名
fileSize	回读文件大小
fileCrc	回读的文件CRC

说明: 开始读文件

函数:

```
[DllImport("bx_sdk_dual_server.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_cmd_uart_ofsStartReedFile(byte[] uartPort, byte
baudRate, byte[] fileName, int[] fileSize, int[] fileCrc);
```

bxDual_cmd_uart_confStartReedFile 【】

返回值：成功返回0；失败返回错误号

参数：

参数	说明
uartPort	串口号,如: "COM3"
baudRate	1: 9600; 2: 57600;
fileName	需要读取的文件名
fileSize	回读文件大小
fileCrc	回读的文件CRC

说明：开始读文件 * 注：此函数用于对存储在固定位置的文件进行处理，例如：Firmware 文件、控制器参数配置文件、扫描配置文件等。

函数：

```
[DllImport("bx_sdk_dual_server.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_cmd_uart_confStartReedFile(byte[] uartPort, byte
baudRate, byte[] fileName, int[] fileSize, int[] fileCrc);
```

bxDual_cmd_uart_ofsReedFileBlock 【】

返回值：成功返回0；失败返回错误号

参数：

参数	说明
uartPort	串口号,如: "COM3"
baudRate	1: 9600; 2: 57600;
fileName	需要读取的文件名
fileAddre	传入读文件写的位置

说明：读文件

* 注：用于对存储在 ofs 中的文件的处理，例如：节目文件，字库文件、播放列表文件等

函数：

```
[DllImport("bx_sdk_dual_server.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_cmd_uart_ofsReedFileBlock(byte[] uartPort, byte
baudRate, byte[] fileName, byte[] fileAddre);
```


bxDual_cmd_uart_confReedFileBlock 【】

返回值：成功返回0；失败返回错误号

参数：

参数	说明
uartPort	串口号,如: "COM3"
baudRate	1: 9600; 2: 57600;
fileName	需要读取的文件名
fileAddre	传入读文件写的位置

说明：读文件

* 注：此函数用于对存储在固定位置的文件进行处理， 例如： Firmware 文件、 控制器参数配置文件、 扫描配置文件等。

函数：

```
[DllImport("bx_sdk_dual_server.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_cmd_uart_confReedFileBlock(byte[] uartPort, byte
baudRate, byte[] fileName, byte[] fileAddre);
```

bxDual_cmd_uart_ofsReedDirBlock

返回值：成功返回0；失败返回错误号

参数：

参数	说明
uartPort	串口号,如: "COM3"
baudRate	1: 9600; 2: 57600;
dirBlock	GetDirBlock_G56

说明：下面两条命令搭配使用可以获取所有文件名

函数：

```
[DllImport("bx_sdk_dual_server.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_cmd_uart_ofsReedDirBlock(byte[] uartPort, byte
baudRate, ref GetDirBlock_G56 dirBlock);
```

bxDual_cmd_uart_ofsFreeDirBlock 【】

返回值：成功返回0；失败返回错误号

参数：

参数	说明
dirBlock	GetDirBlock_G56

说明：释放cmd_ofsReedDirBlock所创建的节目列表dirBlock 注：

* dirBlock 上述两条命令调用完成后dirBlock不再使用时用此函数释放文件列表

函数：

```
[DllImport("bx_sdk_dual_server.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_cmd_uart_ofsFreeDirBlock(ref GetDirBlock_G56
dirBlock);
```

bxDual_cmd_uart_ofsGetTransStatus 【*】

返回值：成功返回0；失败返回错误号

参数：

参数	说明
uartPort	串口号,如: "COM3"
baudRate	1: 9600; 2: 57600;
r_w	
fileName	文件名
fileCrc	文件CRC
fileOffset	

说明：

函数：

```
[DllImport("bx_sdk_dual_server.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_cmd_uart_ofsGetTransStatus(byte[] uartPort, byte
baudRate, byte[] r_w, byte[] fileName, int[] fileCrc, int[] fileOffset);
```

bxDual_cmd_uart_sendConfigFile 【】

返回值：成功返回0；失败返回错误号

参数：

参数	说明
uartPort	串口号,如: "COM3"
baudRate	1: 9600; 2: 57600;
configData	ConfigFile

说明： 发送配置文件到控制器

函数：

```
[DllImport("bx_sdk_dual_server.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_cmd_uart_sendConfigFile(byte[] uartPort, byte
baudRate, ref ConfigFile configData);
```

bxDual_cmd_uart_programLock

返回值： 成功返回0；失败返回错误号

参数：

参数	说明
uartPort	串口号,如: "COM3"
baudRate	1: 9600; 2: 57600;
nonvolatile	状态是否掉电保存 0x00 -掉电不保存 0x01 -掉电保存
locked	0x00 -解锁, 0x01 -锁定
name	节目名称4 (byte) 个字节
lockDuration	节目锁定时间长度, 单位为 10 毫秒 例如当该值为 100 时表示锁定节目 1 秒.注意: 当该值为 0xffffffff 时表示节目锁定无时间长度限制

说明： 节目锁定

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_cmd_uart_programLock(byte[] uartPort, byte
baudRate, byte nonvolatile, byte locked, byte[] name, int lockDuration);

[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_cmd_uart_programLock_6G(byte[] uartPort, byte
baudRate, byte nonvolatile, byte locked, byte[] name, int lockDuration);
```

bxDual_cmd_AT_setWifiSsidPwd 【】

返回值：成功返回0；失败返回错误号

参数：

参数	说明
ssid	控制器WIFI ssid
pwd	控制WIFI密码

说明：设置wifi卡的 ssid pwd

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_cmd_AT_setWifiSsidPwd(byte[] ssid, byte[] pwd);
```

bxDual_cmd_AT_getWifiSsidPwd 【】

返回值：成功返回0；失败返回错误号

参数：

参数	说明
ssid	控制器WIFI ssid
pwd	控制WIFI密码

说明：获取WIFI卡ssid pwd

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_cmd_AT_getWifiSsidPwd(byte[] ssid, byte[] pwd);
```

bxDual_cmd_udpNetworkSearch

返回值：成功返回0；失败返回错误号

参数：

参数	说明
retData	heartbeatData

说明：网络搜索

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_cmd_udpNetworkSearch(ref heartbeatData retData);
```

bxDual_cmd_udpNetworkSearch_6G

返回值：成功返回0；失败返回错误号

参数：

参数	说明
retData	NetSearchCmdRet

说明：网络搜索命令，返回：温度传感器，空气，PM2.5等信息，详见 NetSearchCmdRet:参考结构体声明中的注释；

* 注： 针对 6代卡 的网络搜索命令

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_cmd_udpNetworkSearch_6G(ref NetSearchCmdRet
retData);
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_cmd_udpNetworkSearch_6G_web(ref NetSearchCmdRet_web
retData);
```

bxDual_cmd_udpPing

返回值：成功返回0；失败返回错误号

参数：无

参数	说明
retData	回读参数都会通过结构体回调 Ping_data

说明：UDP ping 命令 * 注： * 此命令用来搜索加屏使用

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_cmd_udpPing(ref Ping_data retData);
```

bxDual_cmd_udpSetMac 【】

返回值：成功返回0；失败返回错误号

参数：无

参数	说明
mac	传入的MAC地址

说明： 设置 MAC 地址命令

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_cmd_udpSetMac(byte[] mac);
```

bxDual_cmd_udpSetIP 【】

返回值： 成功返回0；失败返回错误号

参数：

参数	说明
mode	控制器连接模式： 0x00 -单机直连（PC 与控制器直接连接） 0x01 -自动获取IP（DHCP） 0x02 -手动设置IP（Static IP） 0x03 -服务器模式（动态 IP）
ip	要设置的IP地址
subnetMask	子网掩码
gateway	默认网关
port	端口号
serverMode	Bit[0]表示服务器模式是否使能： 1 -使能，0 -禁止 Bit[1]表示服务器模式： 1 -web 模式，0 -普通模式
serverIP	服务IP
serverPort	服务器端口号
password	服务器访问密码
heartbeat	心跳间隔时间单位秒 默认值20
netID	控制器网络ID，12个数据长度

说明： 设置 IP 地址相关参数命令

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_cmd_udpSetIP(byte mode, byte[] ip, byte[]
subnetMask, byte[] gateway, short port, byte serverMode, byte[] serverIP, short serverPort,
byte[] password, short heartbeat, byte[] netID);
```

bxDual_cmd_sysReset

返回值：成功返回0；失败返回错误号

参数：

参数	说明
ip	控制器IP
port	控制器端口

说明：让系统复位 * 注： * 此命令调用后所有参数全部会丢失

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_cmd_sysReset(byte[] ip, ushort port);
```

bxDual_cmd_tcpPing

返回值：成功返回0；失败返回错误号

参数：

参数	说明
ip	控制器IP
port	控制器端口
retData	Ping_data

说明：通过TCP方式获取到控制器相关属性和IP地址 * 注： * 和UDP PING命令获取到的参数相同

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_cmd_tcpPing(byte[] ip, ushort port, ref Ping_data
retData);
```

bxDual_cmd_check_time

返回值：成功返回0；失败返回错误号

参数：

参数	说明
ip	控制器IP
port	控制器端口

说明：校时，让控制器和当前上位机所在系统时间一致

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_cmd_check_time(byte[] ip, ushort port);
```

bxDual_cmd_tcpNetworkSearch_6G

返回值：成功返回0；失败返回错误号

参数：

参数	说明
ip	控制器IP
port	控制器端口
retData	NetSearchCmdRet

说明：网络搜索命令，返回：温度传感器，空气，PM2.5等信息，详见 NetSearchCmdRet:参考结构体声明中的注释；

* 注： 针对 6代卡 的网络搜索命令

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_cmd_tcpNetworkSearch_6G(byte[] ip, ushort port, ref
NetSearchCmdRet retData);
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_cmd_tcpNetworkSearch_6G_Web(byte[] ip, ushort port,
ref NetSearchCmdRet_Web retData);
```

bxDual_cmd_coerceOnOff

返回值：成功返回0；失败返回错误号

参数：

参数	说明
ip	控制器IP
port	控制器端口
onOff	控制器状态：0x01 -开机 0x00 -关机

说明： 强制开关机命令

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_cmd_coerceOnOff(byte[] ip, ushort port, byte onOff);
```

bxDual_cmd_timingOnOff

返回值： 成功返回0；失败返回错误号

参数：

参数	说明
ip	控制器IP
port	控制器端口
groupNum	有几组定时开关机
data	TimingOnOff 结构体的地址

说明： 定时开关机命令 * 注： * groupNum值是n组情况,data大小 = n * TimingOnOff

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_cmd_timingOnOff(byte[] ip, ushort port, byte groupNum, byte[] data);
```

bxDual_cmd_cancelTimingOnOff

返回值： 成功返回0；失败返回错误号

参数：

参数	说明
ip	控制器IP
port	控制器端口

说明： 取消定时开关机

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_cmd_cancelTimingOnOff(byte[] ip, ushort port);
```

bxDual_cmd_setBrightness

返回值： 成功返回0；失败返回错误号

参数：

参数	说明
ip	控制器IP
port	控制器端口
brightness	亮度度表 Brightness

说明： 设置亮度和相关模式

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_cmd_setBrightness(byte[] ip, ushort port, ref
Brightness brightness);
```

bxDual_cmd_readControllerID 【】

返回值： 成功返回0；失败返回错误号

参数：

参数	说明
ip	控制器IP
port	控制器端口
ControllerID	传回控制器ID

说明： 读控制器ID * 注： * ControllerID是8个字节 请定义char data[8];

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_cmd_readControllerID(byte[] ip, ushort port, byte[]
ControllerID);
```

bxDual_cmd_screenLock

返回值：成功返回0；失败返回错误号

参数：

参数	说明
ip	控制器IP
port	控制器端口
nonvolatile	状态是否掉电保存 0x00-掉电不保存 0x01-掉电保存
locked	0x00-解锁 0x01-锁定

说明：屏幕锁定

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_cmd_screenLock(byte[] ip, ushort port, byte
nonvolatile, byte locked);
```

bxDual_cmd_programLock

返回值：成功返回0；失败返回错误号

参数：

参数	说明
ip	控制器IP
port	控制器端口
nonvolatile	状态是否掉电保存 0x00-掉电不保存 0x01-掉电保存
locked	0x00-解锁 0x01-锁定
name	节目名称4 (byte) 个字节，格式P***，例如“P000”
lockDuration	节目锁定时间长度，单位为 10 毫秒，例 如当该值为 100 时表示锁定节目 1 秒.注意：当该值为 0xffffffff 时表示节目锁定无时间长度限制

说明：节目锁定

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_cmd_programLock(byte[] ip, ushort port, byte
nonvolatile, byte locked, byte[] name, uint lockDuration);
```

bxDual_cmd_check_controllerStatus 【】

返回值：成功返回0；失败返回错误号

参数：

参数	说明
ip	控制器IP
port	控制器端口
controllerStatus	请参考结构体 ControllerStatus_G56

说明：读控制器状态

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_cmd_check_controllerStatus(byte[] ip, ushort port,
ref ControllerStatus_G56 controllerStatus);
```

bxDual_cmd_setPassword 【】

返回值：成功返回0；失败返回错误号

参数：

参数	说明
ip	控制器IP
port	控制器端口
oldPassword	老密码
newPassword	新密码

说明：设置控制器密码 * 注：* 设置后一定要记住，设置后就不在能明码通讯

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_cmd_setPassword(byte[] ip, ushort port, byte[]
oldPassword, byte[] newPassword);
```

bxDual_cmd_deletePassword 【】

返回值：成功返回0；失败返回错误号

参数：

参数	说明
ip	控制器IP
port	控制器端口
password	输出当前控制密码

说明：删除当前控制器密码

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_cmd_deletePassword(byte[] ip, ushort port, byte[] password);
```

bxDual_cmd_setDelayTime 【】

返回值：成功返回0；失败返回错误号

参数：

参数	说明
ip	控制器IP
port	控制器端口
delayTime	开机延时，单位秒

说明：设置控制开机延时时间

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_cmd_setDelayTime(byte[] ip, ushort port, short delayTime);
```

bxDual_cmd_setBtnFunc 【】

返回值：成功返回0；失败返回错误号

参数：

参数	说明
ip	控制器IP
port	控制器端口
btnMode	按钮模式 0x00-测试按钮 0x01-沿触发切换节目 0x02-电平触发切换节目

说明： 设置控制测试按钮功能

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_cmd_setBtnFunc(byte[] ip, ushort port, byte
btnMode);
```

bxDual_cmd_setTimingReset 【】

返回值： 成功返回0；失败返回错误号

参数：

参数	说明
ip	控制器IP
port	控制器端口
cmdData	参考结构体 TimingReset

说明： 设置控制重启重启时间

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_cmd_setTimingReset(byte[] ip, ushort port, ref
TimingReset cmdData);
```

bxDual_cmd_setDispMode 【】

返回值： 成功返回0；失败返回错误号

参数：

参数	说明
ip	控制器IP
port	控制器端口
dispMode	控制器的显示模式（目前只针对 BX-5E系列控制器） Bit0 –串/并行， 0 表示并行， 1 表示并行 Bit1–同步使能， 1 使能同步， 0 禁止同步

说明： 设置控制重启重启时间

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_cmd_setDispMode(byte[] ip, ushort port, byte
dispMode);
```

bxDual_cmd_battieTime 【】

返回值：成功返回0；失败返回错误号

参数：

参数	说明
ip	控制器IP
port	控制器端口
mode	战斗时间控制命令 0x00:启动战斗时间 0x01:暂停战斗时间 0x02:复位战斗时间
battieData	命令回读参数请参考结构体 BattleTime

说明：战斗时间管理命令

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_cmd_battieTime(byte[] ip, ushort port, byte mode,
ref BattleTime battieData);
```

bxDual_cmd_getStopwatch 【】

返回值：成功返回0；失败返回错误号

参数：

参数	说明
ip	控制器IP
port	控制器端口
mode	秒表控制命令 0x00:启动秒表 0x01:暂停秒表 0x02:复位秒表
timeValue	回读回来的当前秒表，时间单位毫秒

无

说明：秒表控制并获取秒表时间

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_cmd_getStopwatch(byte[] ip, ushort port, byte mode,
int[] timeValue);
```

bxDual_cmd_getSensorBrightnessValue 【】

返回值：成功返回0；失败返回错误号

参数：

参数	说明
ip	控制器IP
port	控制器端口
brightnessValue	当前亮度传感器值

说明： 获取亮度读传感器值

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_cmd_getSensorBrightnessValue(byte[] ip, ushort
port, int[] brightnessValue);
```

bxDual_cmd_setSpeedAdjust 【】

返回值：成功返回0；失败返回错误号

参数：

参数	说明
ip	控制器IP
port	控制器端口
speed	速度微调参数值 该值以 0.1 毫秒为单位，共 256 级，上位机下发时该值为 0-255，这样刚好使用一个低位字节，高位字节为 0，留作以后扩展使用。下位机根据该参数在每次循环中延时相应的时间，以改善 LED 屏幕的显示效果。 当该参数为 0 时，下位机延时为 0， 该参数为 1 时，下位机延时 0.1 毫秒，以此类推

说明： 速度微调命令

函数：


```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_cmd_setSpeedAdjust(byte[] ip, ushort port, short speed);
```

bxDual_cmd_setScreenAddress 【】

返回值：成功返回0；失败返回错误号

参数：

参数	说明
ip	控制器IP
port	控制器端口
address	屏幕号

说明：设置屏幕号

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_cmd_setScreenAddress(byte[] ip, ushort port, short address);
```

bxDual_cmd_ofsFormat

返回值：成功返回0；失败返回错误号

参数：

参数	说明
ip	控制器IP
port	控制器端口

说明：文件系统格式化

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_cmd_ofsFormat(byte[] ip, ushort port);
```

bxDual_cmd_ofsStartFileTransf

返回值：成功返回0；失败返回错误号

参数：

参数	说明
ip	控制器IP
port	控制器端口

说明： 开始批量写文件

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_cmd_ofsStartFileTransf(byte[] ip, ushort port);
```

bxDual_cmd_ofsEndFileTransf

返回值： 成功返回0；失败返回错误号

参数：

参数	说明
ip	控制器IP
port	控制器端口

说明： 写文件结束

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_cmd_ofsEndFileTransf(byte[] ip, ushort port);
```

bxDual_cmd_ofsDeleteFormatFile

返回值： 成功返回0；失败返回错误号

参数：

参数	说明
ip	控制器IP
port	控制器端口
fileNub	要删除的文件个数
fileName	要删除的文件名

说明： 删除文件 * 注： * fileName是4个字节 fileNub值为N就要把N个fileName拼接 fileName大小 = fileName (4byte) *N

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_cmd_ofsDeleteFormatFile(byte[] ip, ushort port,
short fileNub, byte[] fileName);
```

bxDual_cmd_confDeleteFormatFile 【】

返回值：成功返回0；失败返回错误号

参数：

参数	说明
ip	控制器IP
port	控制器端口
fileNub	要删除的文件个数
fileName	要删除的文件名

说明：删除文件 * 注：此函数用于对存储在固定位置的文件进行处理，例如：Firmware 文件、控制器参数配置文件、扫描配置文件等。 * fileName是4个字节 fileNub值为N就要把N个fileName拼接 fileName大小 = fileName (4byte) *N

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_cmd_confDeleteFormatFile(byte[] ip, ushort port,
short fileNub, byte[] fileName);
```

bxDual_cmd_ofsGetMemoryVolume

返回值：成功返回0；失败返回错误号

参数：

参数	说明
ip	控制器IP
port	控制器端口
totalMemVolume	全部空间大小
availableMemVolume	剩余空间大小

说明：获取控制空间大小和剩余空间 * 注： * 发节目前需要查询防止空间不够用

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_cmd_ofsGetMemoryVolume(byte[] ip, ushort port, ref
int totalMemVolume, ref int availableMemVolume);
```

bxDual_cmd_ofsWriteFile

返回值：成功返回0；失败返回错误号

参数：

参数	说明
ip	控制器IP
port	控制器端口
fileName	文件名
fileType	文件类型
fileLen	文件长度
overwrite	文件所在的缓存地址
fileAddre	是否覆盖控制上的文件 1覆盖 0不覆盖 建议发1

说明：写文件到控制 * 注：用于对存储在 OFS 中的文件的处理，例如：节目文件，字库文件、播放列表文件等

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_cmd_ofswriteFile(byte[] ip, ushort port, byte[]
fileName, byte fileType, uint fileLen, byte overwrite, IntPtr fileAddre);
```

bxDual_cmd_confWriteFile 【】

返回值：成功返回0；失败返回错误号

参数：

参数	说明
ip	控制器IP
port	控制器端口
fileName	文件名
fileType	文件类型
fileLen	文件长度
overwrite	文件所在的缓存地址
fileAddre	是否覆盖控制上的文件 1覆盖 0不覆盖 建议发1

说明： 写文件到控制 * 注：此函数用于对存储在固定位置的文件进行处理，例如：Firmware 文件、控制器参数配置文件、扫描配置文件等。

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_cmd_confWriteFile(byte[] ip, ushort port, byte[]
fileName, byte fileType, int fileLen, byte overwrite, byte[] fileAddre);
```

bxDual_cmd_ofsStartReedFile 【】

返回值： 成功返回0；失败返回错误号

参数：

参数	说明
ip	控制器IP
port	控制器端口
fileName	需要读取的文件名
fileSize	回读文件大小
fileCrc	回读的文件CRC

说明： 开始读文件

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_cmd_ofsStartReedFile(byte[] ip, ushort port, byte[]
fileName, int[] fileSize, int[] fileCrc);
```

bxDual_cmd_confStartReedFile 【】

返回值：成功返回0；失败返回错误号

参数：

参数	说明
ip	控制器IP
port	控制器端口
fileName	需要读取的文件名
fileSize	回读文件大小
fileCrc	回读的文件CRC

说明：开始读文件 * 注：此函数用于对存储在固定位置的文件进行处理，例如：Firmware 文件、控制器参数配置文件、扫描配置文件等。

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_cmd_confStartReedFile(byte[] ip, ushort port,
byte[] fileName, int[] fileSize, int[] fileCrc);
```

bxDual_cmd_ofsReedFileBlock 【】

返回值：成功返回0；失败返回错误号

参数：

参数	说明
ip	控制器IP
port	控制器端口
fileName	需要读取的文件名
fileAddre	传入读文件写的位置

说明：读文件 * 注：用于对存储在 OFS 中的文件的处理，例如：节目文件，字库文件、播放列表文件等 * fileAddre大小根据bxDual_cmd_ofsStartReedFile函数回调值确定

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_cmd_ofsReedFileBlock(byte[] ip, ushort port, byte[]
fileName, byte[] fileAddre);
```

bxDual_cmd_confReedFileBlock 【】

返回值：成功返回0；失败返回错误号

参数：

参数	说明
ip	控制器IP
port	控制器端口
fileName	需要读取的文件名
fileAddre	传入读文件写的位置

说明： 读文件 * 注：此函数用于对存储在固定位置的文件进行处理，例如：Firmware 文件、控制器参数配置文件、扫描配置文件等。 * fileAddre大小根据bxDual_cmd_ofsStartReedFile函数回调值确定

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_cmd_confReedFileBlock(byte[] ip, ushort port,
byte[] fileName, byte[] fileAddre);
```

bxDual_cmd_ofsReedDirBlock

返回值： 成功返回0；失败返回错误号

参数：

参数	说明
ip	控制器IP
port	控制器端口
dirBlock	读回的文件列表，具体的具体参考 GetDirBlock_G56 结构体

说明： 获取文件列表

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_cmd_ofsReedDirBlock(byte[] ip, ushort port, ref
GetDirBlock_G56 dirBlock);
```

bxDual_cmd_getFileAttr

返回值： 成功返回0；失败返回错误号

参数：

参数	说明
dirBlock	读回的文件列表，具体的具体参考 GetDirBlock_G56 结构体
number	要获取的第几个文件的属性
fileAttr	获取到的文件属性存放位置参考结构体 FileAttribute_G56

说明： 获取指定文件的属性 * 注： * number：此参数值小于fileAttr.fileName 从0开始

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_cmd_getFileAttr(ref GetDirBlock_G56 dirBlock,
ushort number, ref FileAttribute_G56 fileAttr);
```

bxDual_cmd_ofs_freeDirBlock 【】

返回值： 成功返回0；失败返回错误号

参数：

参数	说明
dirBlock	读回的文件列表，具体的具体参考 GetDirBlock_G56 结构体

说明： 释放bxDual_cmd_ofsReedDirBlock所创建的节目列表dirBlock * 注： * dirBlock 上述两条命令调用完成后dirBlock不再使用时用此函数释放文件列表

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_cmd_ofs_freeDirBlock(ref GetDirBlock_G56 dirBlock);
```

bxDual_cmd_ofsGetTransStatus 【*】

返回值： 成功返回0；失败返回错误号

参数：

参数	说明
ip	控制器IP
port	控制器端口
r_w	
fileName	
fileCrc	
fileOffset	

说明： 初始化动态库

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_cmd_ofsGetTransStatus(byte[] ip, ushort port,
byte[] r_w, byte[] fileName, int[] fileCrc, int[] fileoffset);
```

bxDual_cmd_firmwareActivate 【】

返回值： 成功返回0；失败返回错误号

参数：

参数	说明
ip	控制器IP
port	控制器端口
firmwareFileName	要激活的固件名称，缺省值为4个字节字符串“F001”

说明： 激活指定固件

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_cmd_firmwareActivate(byte[] ip, ushort port, byte[]
firmwareFileName);
```

bxDual_cmd_sendConfigFile 【】

返回值： 成功返回0；失败返回错误号

参数：

参数	说明
ip	控制器IP
port	控制器端口
configData	请参考结构体 ConfigFile

说明： 发送5代卡配置文件到控制器

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_cmd_sendConfigFile(byte[] ip, ushort port, ref
ConfigFile configData);
```

bxDual_cmd_sendConfigFile_G6 【】

返回值： 成功返回0；失败返回错误号

参数：

参数	说明
ip	控制器IP
port	控制器端口
configData	请参考结构体 ConfigFile_G6

说明： 发送5代卡配置文件到控制器

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_cmd_sendConfigFile_G6(byte[] ip, ushort port, ref
ConfigFile_G6 configData);
```

bxDual_get_crc16 【】

返回值： 成功返回0；失败返回错误号

参数：

参数	说明
crc16	获取CRC16值

说明： 用来计算CRC16值

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_get_crc16(ref FileCRC16_G56 crc16);
```

bxDual_get_crc32 【】

返回值：成功返回0；失败返回错误号

参数：

参数	说明
crc32	获取CRC32值

说明：用来计算CRC32值

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_get_crc32(ref FileCRC32_G56 crc32);
```

bxDual_program_deleteProgram

返回值：成功返回0；失败返回错误号

参数：无

说明：删除节目

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_program_deleteProgram();
```

bxDual_program_freeBuffer

返回值：成功返回0；失败返回错误号

参数：

参数	说明
program	EQprogram

说明：删除节目

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_program_freeBuffer(ref EQprogram program);
```

bxDual_program_pictureArea 【】

返回值：成功返回0；失败返回错误号

参数：

参数	说明
programID	节目的ID号
ip	控制器IP
port	控制器端口

说明：只是用来测试图文区 * 注： * 屏幕大小为1024X80 输出26个字母

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_program_pictureArea(int programID, byte[] ip,
ushort port);
```

bxDual_program_setScreenParams_G56

返回值：成功返回0；失败返回错误号

参数：

参数	说明
color	屏型 E_ScreenColor_G56
ControllerType	控制卡型号值
doubleColor	点阵类型 E_DoubleColorPixel_G56

说明：设置屏相关属性

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_program_setScreenParams_G56(E_ScreenColor_G56
color, ushort ControllerType, E_DoubleColorPixel_G56 doubleColor);
```

bxDual_program_addProgram

返回值：成功返回0；失败返回错误号

参数：

参数	说明
programH	参考结构体 EQprogramHeader

说明： 添加节目句柄

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_program_addProgram(ref EQprogramHeader programH);
```

bxDual_program_changeProgramParams 【】

返回值： 成功返回0；失败返回错误号

参数：

参数	说明
programH	参考结构体 EQprogramHeader

说明： 修改已添加节目的一些参数

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_program_changeProgramParams(ref EQprogramHeader
programH);
```

bxDual_program_addPlayPeriodGrp

返回值： 成功返回0；失败返回错误号

参数：

参数	说明
header	EQprogramppGrp_G56

说明： 添加节目播放时段

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_program_addPlayPeriodGrp(ref EQprogramppGrp_G56
header);
```

bxDual_program_AddArea

返回值： 成功返回0；失败返回错误号

参数：

参数	说明
areaID	区域的ID号
aheader	参考结构体 EQareaHeader

说明： 添加区域句柄

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_program_AddArea(ushort areaID, ref EQareaHeader aheader);
```

bxDual_program_deleteArea 【】

返回值： 成功返回0；失败返回错误号

参数：

参数	说明
areaID	区域的ID号

说明： 用来删除编号为areaID的区域

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_program_deleteArea(ushort areaID);
```

bxDual_program_picturesAreaAddTxt

返回值： 成功返回0；失败返回错误号

参数：

参数	说明
areaID	区域的ID号
str	需要画的字符
fontName	字体名称
pheader	参考结构体 EQpageHeader

说明： 画字符到图文区

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_program_picturesAreaAddTxt(ushort areaID, byte[]
str, byte[] fontName, ref EQpageHeader pheader);
```

bxDual_program_picturesAreaChangeTxt 【】

返回值：成功返回0；失败返回错误号

参数：

参数	说明
areaID	区域的ID号
str	需要画的字符
pheader	参考结构体 EQpageHeader

说明：修改图文区域内容

- 注：
- 只可以修改文字内容和EQpageHeader结构体里面的参数，不可以修改字体，如需修改，需要删除区域后重新添加文本设置字体

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_program_picturesAreaChangeTxt(ushort areaID, byte[]
str, ref EQpageHeader pheader);
```

bxDual_program_fontPath_picturesAreaAddTxt

返回值：成功返回0；失败返回错误号

参数：

参数	说明
areaID	区域的ID号
str	需要画的字符
fontName	字体绝对路径加字库文件名称
pheader	参考结构体 EQpageHeader

说明：图文区添加字符串--使用字库

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_program_fontPath_picturesAreaAddTxt(ushort areaID,
byte[] str, byte[] fontPathName, ref EQpageHeader pheader);
```

bxDual_program_fontPath_picturesAreaChangeTxt 【】

返回值：成功返回0；失败返回错误号

参数：

参数	说明
areaID	区域的ID号
str	需要画的字符
pheader	参考结构体 EQpageHeader

说明：图文区修改字符串--使用字库

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_program_fontPath_picturesAreaChangeTxt(ushort
areaID, byte[] str, ref EQpageHeader pheader);
```

bxDual_program_IntegrateProgramFile

返回值：成功返回0；失败返回错误号

参数：

参数	说明
program	参考结构体 EQprogram

说明：合成节目文件返回节目文件属性及地址 * 注： * EQprogram 结构体是用来回调发送文件所需要参数

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_program_IntegrateProgramFile(ref EQprogram
program);
```

bxDual_program_picturesAreaAddFrame

返回值：成功返回0；失败返回错误号

参数：

参数	说明
areaID	区域的ID号
afHeader	参考结构体 EQareaframeHeader
picPath	边框图片文件的路径

说明： 区域添加边框

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_program_picturesAreaAddFrame(ushort areaID, ref
EQareaframeHeader afHeader, byte[] picPath);
```

bxDual_program_picturesAreaAddFrame_G6

返回值： 成功返回0；失败返回错误号

参数：

参数	说明
areaID	区域的ID号
afHeader	参考结构体 EQscreenframeHeader_G6
picPath	边框图片文件的路径

说明： 区域添加边框

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_program_picturesAreaAddFrame_G6(ushort areaID, ref
EQscreenframeHeader_G6 afHeader, byte[] picPath);
```

bxDual_program_pictureAreaGetOnePage 【】

返回值： 成功返回0；失败返回错误号

参数：

参数	说明
areaID	区域的ID号
pageNum	区域ID
pageData	第几页，从0开始计算

说明： 返回区域第n张图片

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_program_pictureAreaGetOnePage(ushort areaID, int
pageNum, ref getPageData pageData);
```

bxDual_program_pictureAreaAddPic

返回值： 成功返回0；失败返回错误号

参数：

参数	说明
areaID	区域的ID号
picID	图片的ID号
pheader	参考结构体 EQpageHeader
picPath	添加的图片路径

说明： 添加图片到区域

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_program_pictureAreaAddPic(ushort areaID, ushort
picID, ref EQpageHeader pheader, byte[] picPath);
```

bxDual_program_addFrame

返回值： 成功返回0；失败返回错误号

参数：

参数	s说明
sfHeader	参考结构体 EQscreenframeHeader
picPath	添加的边框图片路径

说明： 节目添加边框

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_program_addFrame(ref EQscreenframeHeader sfHeader,
byte[] picPath);
```

bxDual_program_changeFrame 【】

返回值：成功返回0；失败返回错误号

参数：

参数	s说明
sfHeader	参考结构体 EQscreenframeHeader
picPath	边框图片路径

说明：节目修改已添加边框的一些参数

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_program_changeFrame(ref EQscreenframeHeader
sfHeader, byte[] picPath);
```

bxDual_program_removeFrame 【】

返回值：成功返回0；失败返回错误号

参数：无

说明：节目去掉边框

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_program_removeFrame();
```

bxDual_program_pictureAreaRemoveFrame 【】

返回值：成功返回0；失败返回错误号

参数：

参数	说明
areaID	区域的ID号

说明：区域去掉边框

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_program_pictureAreaRemoveFrame(ushort areaID);
```

bxDual_program_MoveArea 【】

返回值：成功返回0；失败返回错误号

参数：

参数	说明
areaID	区域的ID号
x	区域left坐标
y	区域top坐标
width	区域宽度
height	区域高度

说明：改变区域坐标大小

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_program_MoveArea(ushort areaID, int x, int y, int width, int height);
```

bxDual_program_timeAreaAddContent

返回值：成功返回0；失败返回错误号

参数：

参数	说明
areaID	区域的ID号
timeData	详情请见时间区数据格式结构体 EQtimeAreaData_G56

说明：时间分区添加内容

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_program_timeAreaAddContent(ushort areaID, ref EQtimeAreaData_G56 timeData);
```

bxDual_program_fontPath_timeAreaAddContent 【】

返回值：成功返回0；失败返回错误号

参数：

参数	说明
areaID	区域的ID号
timeData	详情请见时间区数据格式结构体 EQtimeAreaData_G56

说明： 时间分区添加内容EQtimeAreaData::fontName == 字库名称

* 注：ios下无法使用program_timeAreaAddContent请使用program_fontPath_timeAreaAddContent()

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_program_fontPath_timeAreaAddContent(ushort areaID,
ref EQtimeAreaData_G56 timeData);
```

bxDual_program_timeAreaChangeContent 【】

返回值： 成功返回0；失败返回错误号

参数：

参数	说明
areaID	区域的ID号
timeData	详情请见时间区数据格式结构体 EQtimeAreaData_G56

说明： 时间分区修改内容EQtimeAreaData::fontName == 字库的路径加字库文件名（字库地址）

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_program_timeAreaChangeContent(ushort areaID, ref
EQtimeAreaData_G56 timeData);
```

bxDual_program_timeAreaGetOnePage 【】

返回值： 成功返回0；失败返回错误号

参数：

参数	说明
areaID	区域的ID号
pageData	getPageData

说明： 返回时间区域第n张图片

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_program_timeAreaGetOnePage(ushort areaID, ref
getPageData pageData);
```

bxDual_program_timeAreaAddAnalogClock

返回值：成功返回0；失败返回错误号

参数：

参数	说明
areaID	区域的ID号
header	详情见 EQAnalogClockHeader_G56 结构体
cStyle	表盘样式，详情见 E_ClockStyle
cColor	表盘颜色，详情见 E_Color_G56 通过此枚举值可以直接配置七彩色，如果大于枚举范围使用RGB888模式

说明：时间分区添加模拟时钟

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_program_timeAreaAddAnalogClock(ushort areaID, ref
EQAnalogClockHeader_G56 header, E_ClockStyle cStyle, ref ClockColor_G56 cColor);
```

bxDual_program_timeAreaChangeAnalogClock 【】

返回值：成功返回0；失败返回错误号

参数：

参数	说明
areaID	区域的ID号
header	详情见 EQAnalogClockHeader_G56 结构体
cStyle	表盘样式，详情见 E_ClockStyle
cColor	表盘颜色，详情见 E_Color_G56 通过此枚举值可以直接配置七彩色，如果大于枚举范围使用RGB888模式

说明：时间分区修改模拟时钟的一些设置参数

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_program_timeAreaChangeAnalogClock(ushort areaID,
ref EQAnalogClockHeader_G56 header, E_ClockStyle cStyle, ref ClockColor_G56 cColor);
```

bxDual_program_timeAreaChangeDialPic 【】

返回值：成功返回0；失败返回错误号

参数：

参数	说明
areaID	区域的ID号
picPath	表盘图片位置

说明：时间分区从外部添加表盘图片

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_program_timeAreaChangeDialPic(ushort areaID, byte[]
picPath);
```

bxDual_program_timeAreaChangeDialPicAdd_G56 【*】 ()

返回值：成功返回0；失败返回错误号

参数：

参数	说明
areaID	区域的ID号
picAdd	
picLen	

说明：时间分区从外部添加表盘图片

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_program_timeAreaChangeDialPicAdd_G56(ushort areaID,
byte[] picAdd, int picLen);
```

bxDual_program_timeAreaRemoveDialPic 【】

返回值：成功返回0；失败返回错误号

参数：

参数	说明
areaID	区域的ID号

说明： 时间分区移除外部添加的表盘图片

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_program_timeAreaRemoveDialPic(ushort areaID);
```

bxDual_dynamicArea_SetDualPixel

返回值： 成功返回0；失败返回错误号

参数：

参数	说明
ePixelRGorGR	点阵类型 E_DoubleColorPixel_G56

说明： 设置动态区颜色像素类型：R+G 或 G+R

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_dynamicArea_SetDualPixel(E_DoubleColorPixel_G56
ePixelRGorGR);
```

bxDual_dynamicArea_AddAreaTxt_6G

返回值： 成功返回0；失败返回错误号

参数：

参数	说明
pIP	控制器IP
nPort	控制器端口
color	LED屏颜色类型，详见 E_ScreenColor_G56 声明
uAreaId	区域的ID号
uAreaX	区域left坐标
uAreaY	区域top坐标
uWidth	区域宽度
uHeight	区域高度
fontName	字体名称，如"宋体"
nFontSize	字体大小，如12
strAreaTxtContent	要显示的文本内容

无

说明：6代更新动态区最基本功能：仅显示动态区：即不与节目一起显示，如果当前有节目显示，调用此函数后，LED屏幕上会清空原来的内容，显示此函数中 strAreaTxtContent 参数的内容；如果要与屏幕上原来显示的节目一起显示，请调用下面的 动态区文本关关节目 函数；与节目一起显示时，要注意动态区域与原来的节目区域不能重叠！

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_dynamicArea_AddAreaTxt_6G(byte[] pIP, int nPort,
E_ScreenColor_G56 color, byte uAreaId, ushort uAreaX, ushort uAreaY,
ushort uwidth, ushort uHeight, byte[] fontName, byte nFontSize, byte[]
strAreaTxtContent);
```

bxDual_dynamicArea_AddAreaTxtDetails_6G

返回值：成功返回0；失败返回错误号

参数：

参数	说明
pIP	控制器IP
nPort	控制器端口
color	LED屏颜色类型，详见 E_ScreenColor_G56 声明
uAreaId	区域的ID号
oAreaHeader_G6	区域属性 EQareaHeader_G6
stPageHeader	显示属性 EQpageHeader_G6
fontName	字体名称，如"宋体"
strAreaTxtContent	要显示的文本内容

说明：6代更新动态区详细功能：仅显示动态区

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_dynamicArea_AddAreaTxtDetails_6G(byte[] pIP, int
nPort, E_ScreenColor_G56 color, byte uAreaId, ref EQareaHeader_G6 oAreaHeader_G6,
ref EQpageHeader_G6 stPageHeader, byte[] fontName, byte[] strAreaTxtContent);
```

bxDual_dynamicArea_AddAreaTxtDetails_6G_Serial

返回值：成功返回0；失败返回错误号

参数：

参数	说明
pSerialName	串口号字符串；如:byte pSerialName[] = "COM3"
nBaudRateIndex	波特率；取值为1时，代表波特率为9600; 取值为2时，代表波特率为57600
color	LED屏颜色类型，详见 E_ScreenColor_G56 声明
uAreaId	区域的ID号
oAreaHeader_G6	区域属性 EQareaHeader_G6
stPageHeader	显示属性 EQpageHeader_G6
fontName	字体名称，如"宋体"
strAreaTxtContent	要显示的文本内容

说明：6代更新动态区详细功能：仅显示动态区使用串口发送

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_dynamicArea_AddAreaTxtDetails_6G_Serial(byte[]
pSerialName, byte nBaudRateIndex, E_ScreenColor_G56 color, byte uAreaId, ref
EQareaHeader_G6 oAreaHeader_G6,
ref EQpageHeader_G6 stPageHeader, byte[] fontName, byte[] strAreaTxtContent);
```

bxDual_dynamicArea_AddAreaTxtDetails_WithProgram_6G

返回值：成功返回0；失败返回错误号

参数：

参数	说明
pIP	控制器IP
nPort	控制器端口
color	LED屏颜色类型，详见 E_ScreenColor_G56 声明
uAreaId	区域的ID号
oAreaHeader_G6	区域属性 EQareaHeader_G6
stPageHeader	显示属性 EQpageHeader_G6
fontName	字体名称，如"宋体"
strAreaTxtContent	要显示的文本内容
RelateProNum	RelateProNum = 0 时，关联所有节目，与所有节目一起播放，如果没有节目，则不播放该动态区； > 0 时，指定关联系节目，要关联的节目ID存放在RelateProSerial[]中；
RelateProSerial	关联系节目编号

说明：动态区文本关联系节目：

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int
bxDual_dynamicArea_AddAreaTxtDetails_WithProgram_6G(byte[] pIP, int nPort,
E_ScreenColor_G56 color, byte uAreaId, ref EQareaHeader_G6 oAreaHeader_G6,
ref EQpageHeader_G6 stPageHeader, byte[] fontName, byte[] strAreaTxtContent,
ushort RelateProNum, ushort[] RelateProSerial);
```

bxDual_dynamicArea_AddAreaTxtDetails_WithProgram_6G_Serial

返回值：成功返回0；失败返回错误号

参数：

参数	说明
pSerialName	串口号字符串；如:byte pSerialName[] = "COM3"
nBaudRateIndex	波特率；取值为1时，代表波特率为9600;取值为2时，代表波特率为57600
color	LED屏颜色类型，详见 E_ScreenColor_G56 声明
uAreaId	区域的ID号
oAreaHeader_G6	区域属性 EQareaHeader_G6
stPageHeader	显示属性 EQpageHeader_G6
fontName	字体名称，如"宋体"
strAreaTxtContent	要显示的文本内容
RelateProNum	RelateProNum = 0 时，关联所有节目，与所有节目一起播放，如果没有节目，则不播放该动态区； > 0 时,指定关联系节目，要关联的节目ID存放在RelateProSerial[]中；
RelateProSerial	关联系节目编号

说明： 动态区文本关联系节目: 串口

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int
bxDual_dynamicArea_AddAreaTxtDetails_withProgram_6G_Serial(byte[] pSerialName, int
nBaudRateIndex, E_ScreenColor_G56 color, byte uAreaId, ref EQareaHeader_G6
oAreaHeader_G6,
ref EQpageHeader_G6 stPageHeader, byte[] fontName, byte[] strAreaTxtContent,
ushort RelateProNum, ushort[] RelateProSerial);
```

bxDual_dynamicArea_AddAreaPic_6G

返回值： 成功返回0；失败返回错误号

参数：

参数	说明
pIP	控制器IP
nPort	控制器端口
color	LED屏颜色类型，详见 E_ScreenColor_G56 声明
uAreaId	区域的ID号
uAreaX	区域left坐标
uAreaY	区域top坐标
uWidth	区域宽度
uHeight	区域高度
pheader	显示属性 EQpageHeader_G6
picPath	显示图片路径

说明： 更新动态区图片：仅显示动态区;

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_dynamicArea_AddAreaPic_6G(byte[] pIP, int nPort,
E_ScreenColor_G56 color, byte uAreaId, ushort AreaX, ushort AreaY,
    ushort Areawidth, ushort AreaHeight, ref EQpageHeader_G6 pheader, byte[]
picPath);
```

bxDual_dynamicArea_AddAreaPic_6G_Serial

返回值： 成功返回0；失败返回错误号

参数：

参数	说明
pSerialName	串口号字符串；如:byte pSerialName[] = "COM3"
nBaudRateIndex	波特率；取值为1时，代表波特率为9600;取值为2时，代表波特率为57600
color	LED屏颜色类型，详见 E_ScreenColor_G56 声明
uAreaId	区域的ID号
uAreaX	区域left坐标
uAreaY	区域top坐标
uWidth	区域宽度
uHeight	区域高度
pheader	显示属性 EQpageHeader_G6
picPath	显示图片路径

说明： 更新动态区图片：仅显示动态区;串口

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_dynamicArea_AddAreaPic_6G_Serial(byte[]
pSerialName, int nBaudRateIndex, E_ScreenColor_G56 color, byte uAreaId, short AreaX, ushort
AreaY,
        ushort Areawidth, ushort AreaHeight, ref EQpageHeader_G6 pheader, byte[]
picPath);
```

bxDual_dynamicArea_AddAreaPic_WithProgram_6G

返回值： 成功返回0；失败返回错误号

参数：

参数	说明
pIP	控制器IP
nPort	控制器端口
color	LED屏颜色类型，详见 E_ScreenColor_G56 声明
uAreaId	区域的ID号
uAreaX	区域left坐标
uAreaY	区域top坐标
uWidth	区域宽度
uHeight	区域高度
pheader	显示属性 EQpageHeader_G6
picPath	显示图片路径
RelateProNum	RelateProNum = 0 时，关联所有节目，与所有节目一起播放，如果没有节目，则不播放该动态区； > 0 时，指定关联系节目，要关联的节目ID存放在RelateProSerial[]中；
RelateProSerial	关联系节目编号

说明： 动态区图片关联系节目

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_dynamicArea_AddAreaPic_WithProgram_6G(byte[] pIP,
int nPort, E_ScreenColor_G56 color, byte uAreaId, ushort AreaX, ushort AreaY,
ushort Areawidth, ushort AreaHeight, ref EQpageHeader_G6 pheader, byte[]
picPath, ushort RelateProNum, ushort[] RelateProSerial);
```

bxDual_dynamicArea_AddAreaPic_WithProgram_G6_Serial

返回值： 成功返回0；失败返回错误号

参数：

参数	说明
pSerialName	串口号字符串；如:byte pSerialName[] = "COM3"
nBaudRateIndex	波特率；取值为1时，代表波特率为9600;取值为2时，代表波特率为57600
color	LED屏颜色类型，详见 E_ScreenColor_G56 声明
uAreaId	区域的ID号
uAreaX	区域left坐标
uAreaY	区域top坐标
uWidth	区域宽度
uHeight	区域高度
pheader	显示属性 EQpageHeader_G6
picPath	显示图片路径
RelateProNum	RelateProNum = 0 时，关联所有节目，与所有节目一起播放，如果没有节目，则不播放该动态区； > 0 时，指定关联节目，要关联的节目ID存放在RelateProSerial[]中；
RelateProSerial	关联节目编号

说明： 动态区图片关联节目:串口

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_dynamicArea_AddAreaPic_withProgram_G6_Serial(byte[]
pSerialName, int nBaudRateIndex, E_ScreenColor_G56 color, byte uAreaId, ushort AreaX,
ushort AreaY,
ushort AreaWidth, ushort AreaHeight, ref EQpageHeader_G6 pheader, byte[]
picPath, ushort RelateProNum, ushort[] RelateProSerial);
```

bxDual_dynamicAreaS_AddTxtDetails_6G

返回值： 成功返回0；失败返回错误号

参数：

参数	说明
pIP	控制器IP
nPort	控制器端口
color	LED屏颜色类型，详见 E_ScreenColor_G56 声明
uAreaCount	更新区域个数
pParams	区域数组 DynamicAreaParams

说明：同时更新多个动态区:仅显示动态区，不显示节目

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_dynamicAreaS_AddTxtDetails_6G(byte[] pIP, int
nPort, E_ScreenColor_G56 color, byte uAreaCount, DynamicAreaParams[] pParams);
```

bxDual_dynamicAreaS_AddTxtDetails_6G_Serial

返回值：成功返回0；失败返回错误号

参数：

参数	说明
pSerialName	串口号字符串；如:byte pSerialName[] = "COM3"
nBaudRateIndex	波特率；取值为1时，代表波特率为9600;取值为2时，代表波特率为57600
color	LED屏颜色类型，详见 E_ScreenColor_G56 声明
uAreaCount	更新区域个数
pParams	区域数组 DynamicAreaParams

说明：同时更新多个动态区文本:仅显示动态区，不显示节目;串口

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_dynamicAreaS_AddTxtDetails_6G_Serial(byte[]
pSerialName, int nBaudRateIndex, E_ScreenColor_G56 color, byte uAreaCount,
DynamicAreaParams[] pParams);
```

bxDual_dynamicAreaS_AddTxtDetails_WithProgram_6G

返回值：成功返回0；失败返回错误号

参数：

参数	说明
pIP	控制器IP
nPort	控制器端口
color	LED屏颜色类型，详见 E_ScreenColor_G56 声明
uAreaCount	更新区域个数
pParams	区域数组 DynamicAreaParams
RelateProNum	RelateProNum = 0 时，关联所有节目，与所有节目一起播放，如果没有节目，则不播放该动态区； > 0 时，指定关联系节目，要关联的节目ID存放在RelateProSerial[]中；
RelateProSerial	关联系节目编号

说明：同时更新多个动态区文本:并与节目关联，即与节目一起显示

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_dynamicAreaS_AddTxtDetails_WithProgram_G6(byte[]
pIP, int nPort, E_ScreenColor_G56 color, byte uAreaCount, DynamicAreaParams[] pParams,
ushort RelateProNum, ushort[] RelateProSerial);
```

bxDual_dynamicAreaS_AddTxtDetails_WithProgram_G6_Serial

返回值：成功返回0；失败返回错误号

参数：

参数	说明
pSerialName	串口号字符串；如:byte pSerialName[] = "COM3"
nBaudRateIndex	波特率；取值为1时，代表波特率为9600;取值为2时，代表波特率为57600
color	LED屏颜色类型，详见 E_ScreenColor_G56 声明
uAreaCount	更新区域个数
pParams	区域数组 DynamicAreaParams
RelateProNum	RelateProNum = 0 时，关联所有节目，与所有节目一起播放，如果没有节目，则不播放该动态区； > 0 时，指定关联系节目，要关联的节目ID存放在RelateProSerial[]中；
RelateProSerial	关联系节目编号

说明：同时更新多个动态区文本:并与节目关联，即与节目一起显示 串口

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int
bxDual_dynamicAreaS_AddTxtDetails_WithProgram_G6_Serial(byte[] pSerialName, int
nBaudRateIndex, E_ScreenColor_G56 color, byte uAreaCount, DynamicAreaParams[] pParams,
ushort RelateProNum, ushort[] RelateProSerial);
```

bxDual_dynamicAreaS_AddAreaPic_6G

返回值：成功返回0；失败返回错误号

参数：

参数	说明
pIP	控制器IP
nPort	控制器端口
color	LED屏颜色类型，详见 E_ScreenColor_G56 声明
uAreaCount	更新区域个数
pParams	区域数组 DynamicAreaParams

说明：同时更新多个动态区图片：仅显示动态区图片;不与节目关联/不与节目一起显示；

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_dynamicAreaS_AddAreaPic_6G(byte[] pIP, int nPort,
E_ScreenColor_G56 color, byte uAreaCount, DynamicAreaParams[] pParams);
```

bxDual_dynamicAreaS_AddAreaPic_6G_Serial

返回值：成功返回0；失败返回错误号

参数：

参数	说明
pSerialName	串口号字符串；如:byte pSerialName[] = "COM3"
nBaudRateIndex	波特率；取值为1时，代表波特率为9600;取值为2时，代表波特率为57600
color	LED屏颜色类型，详见 E_ScreenColor_G56 声明
uAreaCount	更新区域个数
pParams	区域数组 DynamicAreaParams

说明：同时更新多个动态区图片;仅显示动态区图片/不与节目关联/不与节目一起显示；串口

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_dynamicAreaS_AddAreaPic_6G_Serial(byte[]
pSerialName, int nBaudRateIndex, E_ScreenColor_G56 color, byte uAreaCount,
DynamicAreaParams[] pParams);
```

bxDual_dynamicAreaS_AddAreaPic_WithProgram_6G

返回值：成功返回0；失败返回错误号

参数：

参数	说明
pIP	控制器IP
nPort	控制器端口
color	LED屏颜色类型，详见 E_ScreenColor_G56 声明
uAreaCount	更新区域个数
pParams	区域数组 DynamicAreaParams
RelateProNum	RelateProNum = 0 时，关联所有节目，与所有节目一起播放，如果没有节目，则不播放该动态区； > 0 时，指定关联系节目，要关联的节目ID存放在RelateProSerial[]中；
RelateProSerial	关联系节目编号

说明：同时更新多个动态区图片，并与节目关联，即与节目一起显示；

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_dynamicAreaS_AddAreaPic_WithProgram_6G(byte[] pIP,
int nPort, E_ScreenColor_G56 color, byte uAreaCount, DynamicAreaParams[] pParams, ushort
RelateProNum, ushort[] RelateProSerial);
```

bxDual_dynamicAreaS_AddAreaPic_WithProgram_6G_Serial

返回值：成功返回0；失败返回错误号

参数：

参数	说明
pSerialName	串口号字符串；如:byte pSerialName[] = "COM3"
nBaudRateIndex	波特率；取值为1时，代表波特率为9600;取值为2时，代表波特率为57600
color	LED屏颜色类型，详见 E_ScreenColor_G56 声明
uAreaCount	更新区域个数
pParams	区域数组 DynamicAreaParams
RelateProNum	RelateProNum = 0 时，关联所有节目，与所有节目一起播放，如果没有节目，则不播放该动态区； > 0 时, 指定关联系目，要关联的节目ID存放在RelateProSerial[]中；
RelateProSerial	关联系目编号

说明： 同时更新多个动态区图片，并与节目关联，即与节目一起显示；串口

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int
bxDual_dynamicAreaS_AddAreaPic_withProgram_6G_Serial(byte[] pSerialName, int
nBaudRateIndex, E_ScreenColor_G56 color, byte uAreaCount, DynamicAreaParams[] pParams,
ushort RelateProNum, ushort[] RelateProSerial);
```

bxDual_dynamicArea_AddAreaInfos_6G

返回值： 成功返回0；失败返回错误号

参数：

参数	说明
pIP	控制器IP
nPort	控制器端口
color	LED屏颜色类型，详见 E_ScreenColor_G56 声明
uAreaId	区域的ID号
RunMode	动态区运行模式 0— 动态区数据循环显示。 1— 动态区数据显示完成后静止显示最后一页数据。 2— 动态区数据循环显示，超过设定时间后数据仍未更新时不再显示 3— 动态区数据循环显示，超过设定时间后数据仍未更新时显示 Logo 信息,Logo 信息即为动态区域的最后一页信息 4— 动态区数据顺序显示，显示完最后一页后就不再显示
Timeout	动态区数据超时时间，单位为秒
RelateAllPro	当该字节为 1 时，所有异步节目播放时都允许播放该动态区域；为 0 时，由接下来的规则来决定
RelateProNum	动态区域关联了多少个异步节目 一旦关联了某个异步节目，则当该异步节目播放时允许播放该动态区域，否则，不允许播放该动态区域 以下的节目编号根据 RelateProNum 的值来确定，当该值为 0 时不发送
RelateProSerial	节目编号
ImmePlay	是否立即播放 该字节为 0 时，该动态区域与异步节目一起播放 该字节为 1 时，异步节目停止播放，仅播放该动态区域 注意： 当该字节为 0 时，RelateAllPro 到 RelateProSerialN-1 的参数才有效，否则无效 当该参数为 1 时，由于不与异步节目同时播放，为控制该动态区域能及时结束，可选择 RunMode 参数为 2 或 4，当然也可通过删除该区域来实现
uAreaX	区域左上角横坐标(Top Left)，单位 Pixel
uAreaY	区域左上角纵坐标(Top Left)，单位 Pixel
uWidth	区域宽度，单位 Pixel
uHeight	区域高度，单位 Pixel
oFrame	区域边框属性 EQareaframeHeader
nInfoCount	数据数量
pInfo	显示数据数组 DynamicAreaBaseInfo_5G

说明： 增加多条信息（文本/图片）到指定的动态区，并可以关联这个动态区到指定的节目；

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_dynamicArea_AddAreaInfos_6G(byte[] pIP, int nPort,
E_ScreenColor_G56 color,
    byte uAreaId,
    byte RunMode,
    ushort Timeout,
    byte RelateAllPro,
    ushort RelateProNum,
    ushort[] RelateProSerial,
    byte ImmePlay,
    ushort uAreaX, ushort uAreaY, ushort uWidth, ushort uHeight,
    EQareaframeHeader oFrame,

    byte nInfoCount,
    DynamicAreaBaseInfo_5G[] pInfo
);
```

bxDual_dynamicArea_AddAreaInfos_G6_Serial

返回值: 成功返回0; 失败返回错误号

参数:

参数	说明
pSerialName	串口号字符串；如:byte pSerialName[] = "COM3"
nBaudRateIndex	波特率；取值为1时，代表波特率为9600;取值为2时，代表波特率为57600
color	LED屏颜色类型，详见 E_ScreenColor_G56 声明
uAreaId	区域的ID号
RunMode	动态区运行模式 0— 动态区数据循环显示。 1— 动态区数据显示完成后静止显示最后一页数据。 2— 动态区数据循环显示，超过设定时间后数据仍未更新时不再显示 3— 动态区数据循环显示，超过设定时间后数据仍未更新时显示 Logo 信息,Logo 信息即为动态区域的最后一页信息 4— 动态区数据顺序显示，显示完最后一页后就不再显示
Timeout	动态区数据超时时间，单位为秒
RelateAllPro	当该字节为 1 时，所有异步节目播放时都允许播放该动态区域；为 0 时，由接下来的规则来决定
RelateProNum	动态区域关联了多少个异步节目 一旦关联了某个异步节目，则当该异步节目播放时允许播放该动态区域，否则，不允许播放该动态区域 以下的节目编号根据 RelateProNum 的值来确定，当该值为 0 时不发送
RelateProSerial	节目编号
ImmePlay	是否立即播放 该字节为 0 时，该动态区域与异步节目一起播放 该字节为 1 时，异步节目停止播放，仅播放该动态区域 注意： 当该字节为 0 时，RelateAllPro 到 RelateProSerialN-1 的参数才有效，否则无效 当该参数为 1 时，由于不与异步节目同时播放，为控制该动态区域能及时结束，可选择 RunMode 参数为 2 或 4，当然也可通过删除该区域来实现
uAreaX	区域左上角横坐标(Top Left)，单位 Pixel
uAreaY	区域左上角纵坐标(Top Left)，单位 Pixel
uWidth	区域宽度，单位 Pixel
uHeight	区域高度，单位 Pixel
oFrame	区域边框属性 EQareaframeHeader
nInfoCount	数据数量
pInfo	显示数据数组[DynamicAreaBaseInfo_5G](

说明： 增加多条信息（文本/图片）到指定的动态区，并可以关联这个动态区到指定的节目；串口

函数：


```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_dynamicArea_AddAreaInfos_G6_Serial(byte[]
pSerialName, byte nBaudRateIndex, E_ScreenColor_G56 color,
    byte uAreaId,
    byte RunMode,
    ushort Timeout,
    byte RelateAllPro,
    ushort RelateProNum,
    ushort[] RelateProSerial,
    byte ImmePlay,
    ushort uAreaX, ushort uAreaY, ushort uWidth, ushort uHeight,
    EQareaframeHeader oFrame,

    byte nInfoCount,
    DynamicAreaBaseInfo_5G[] pInfo
);
```

bxDual_dynamicArea_AddAreaInfos_6G_V2

返回值: 成功返回0; 失败返回错误号

参数:

参数	说明
pIP	控制器IP
nPort	控制器端口
color	LED屏颜色类型，详见 E_ScreenColor_G56 声明
uAreaId	区域的ID号
RunMode	动态区运行模式 0— 动态区数据循环显示。 1— 动态区数据显示完成后静止显 示最后一页数据。 2— 动态区数据循环显示，超过设 定时间后数据仍未更新时不再 显示 3— 动态区数据循环显示，超过设 定时间后数据仍未更新时显示 Logo 信息,Logo 信息即为 动 态区域的最后一页信息 4— 动态区数据顺序显示，显示完 最后一页后就不再显示
Timeout	动态区数据超时时间，单位为秒
RelateAllPro	当该字节为 1 时，所有异步节目 播放时都允许播放该动态区域； 为 0 时，由接下来的规则 来决定
RelateProNum	动态区域关联了多少个异步节目 一旦关联了某个异步节目，则当 该异步节目播放时允许播 放该动 态区域，否则，不允许播放该动 态区域 以下的节目编号根据 RelateProNum 的值来确定，当该 值为 0 时不发送
RelateProSerial	节目编号
ImmePlay	是否立即播放 该字节为 0 时，该动态区域与异 步节目一起播放 该字节为 1 时，异步节目停止播 放，仅播放该动态区域 注意： 当该字节为 0 时，RelateAllPro 到 RelateProSerialN-1 的参数才有 效，否则无效 当该参数为 1 时，由于不与异步 节目同时播放，为控制该动态区 域能及时结束，可选择 RunMode 参数为 2 或 4，当然也可通过删除 该区域来实现
uAreaX	区域左上角横坐标(Top Left)，单 位 Pixel
uAreaY	区域左上角纵坐标(Top Left)，单 位 Pixel
uWidth	区域宽度，单位 Pixel
uHeight	区域高度，单位 Pixel
oFrame	区域边框属性 EQareaframeHeader
nInfoCount	数据数量
pInfo	显示数据数组 DynamicAreaBaseInfo_5G
pSoundData	语音设置EQSound_6G

说明：一次向一个动态区发送/更新多条信息（文字或图片）及语音

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_dynamicArea_AddAreaInfos_6G_V2(byte[] pIP, int
nPort, E_ScreenColor_G56 color,
    byte uAreaId,
    byte RunMode,
    ushort Timeout,
    byte RelateAllPro,
    ushort RelateProNum,
    ushort[] RelateProSerial,
    byte ImmePlay,
    ushort uAreaX, ushort uAreaY, ushort uwidth, ushort uHeight,
    BxAreaFrmae_Dynamic_G6 oFrame,

    byte nInfoCount,
    DynamicAreaBaseInfo_5G[] pInfo,
    ref EQSound_6G pSoundData
);
```

bxDual_dynamicArea_AddAreaInfos_6G_V2_Serial 【*】

返回值： 成功返回0；失败返回错误号

参数：

参数	说明
pSerialName	串口号字符串；如:byte pSerialName[] = "COM3"
nBaudRateIndex	波特率；取值为1时，代表波特率为9600;取值为2时，代表波特率为57600
color	LED屏颜色类型，详见 E_ScreenColor_G56 声明
uAreaId	区域的ID号
RunMode	动态区运行模式 0— 动态区数据循环显示。 1— 动态区数据显示完成后静止显示最后一页数据。 2— 动态区数据循环显示，超过设定时间后数据仍未更新时不再显示 3— 动态区数据循环显示，超过设定时间后数据仍未更新时显示 Logo 信息,Logo 信息即为动态区域的最后一页信息 4— 动态区数据顺序显示，显示完最后一页后就不再显示
Timeout	动态区数据超时时间，单位为秒
RelateAllPro	当该字节为 1 时，所有异步节目播放时都允许播放该动态区域；为 0 时，由接下来的规则来决定
RelateProNum	动态区域关联了多少个异步节目 一旦关联了某个异步节目，则当该异步节目播放时允许播放该动态区域，否则，不允许播放该动态区域 以下的节目编号根据 RelateProNum 的值来确定，当该值为 0 时不发送
RelateProSerial	节目编号
ImmePlay	是否立即播放 该字节为 0 时，该动态区域与异步节目一起播放 该字节为 1 时，异步节目停止播放，仅播放该动态区域 注意： 当该字节为 0 时，RelateAllPro 到 RelateProSerialN-1 的参数才有效，否则无效 当该参数为 1 时，由于不与异步节目同时播放，为控制该动态区域能及时结束，可选择 RunMode 参数为 2 或 4，当然也可通过删除该区域来实现
uAreaX	区域左上角横坐标(Top Left)，单位 Pixel
uAreaY	区域左上角纵坐标(Top Left)，单位 Pixel
uWidth	区域宽度，单位 Pixel
uHeight	区域高度，单位 Pixel
oFrame	区域边框属性 EQareaframeHeader
nInfoCount	数据数量
pInfo	显示数据数组 DynamicAreaBaseInfo_5G
pSoundData	语音设置EQSound_6G

说明：一次向一个动态区发送/更新多条信息（文字或图片）及语音串口

函数:

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_dynamicArea_AddAreaInfos_6G_V2_Serial(byte[]
pSerialName, byte nBaudRateIndex, E_ScreenColor_G56 color,
    byte uAreaId,
    byte RunMode,
    ushort Timeout,
    byte RelateAllPro,
    ushort RelateProNum,
    ushort[] RelateProSerial,
    byte ImmePlay,
    ushort uAreaX, ushort uAreaY, ushort uwidth, ushort uHeight,
    EQareaframeHeader oFrame,

    byte nInfoCount,
    DynamicAreaBaseInfo_5G[] pInfo
);
```

bxDual_dynamicArea_DelArea_6G

返回值: 成功返回0; 失败返回错误号

参数:

参数	说明
pIP	控制器IP
nPort	控制器端口
uAreald	区域ID, 0xff:删除所有区域

说明: TCP方式删除动态区 删除动态区: 删除单个动态区: uAreald = 0xff:删除所有区域

函数:

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_dynamicArea_DelArea_6G(byte[] pIP, int nPort, byte
uAreaId);
```

bxDual_dynamicArea_DelAreas_6G

返回值: 成功返回0; 失败返回错误号

参数:

参数	说明
pIP	控制器IP
nPort	控制器端口
uAreaCount	动态区ID数组中的个数
uAreaId	存放要删除的动态区ID数组

说明： TCP方式删除多个动态区：

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_dynamicArea_DelAreas_6G(byte[] pIP, int nPort, byte
uAreaCount, byte[] pAreaID);
```

bxDual_dynamicArea_DelArea_G6_Serial

返回值： 成功返回0；失败返回错误号

参数：

参数	说明
pSerialName	串口号字符串；如:byte pSerialName[] = "COM3"
nBaudRateIndex	波特率；取值为1时，代表波特率为9600;取值为2时，代表波特率为57600
uAreaId	区域ID, 0xff:删除所有区域

说明： 串口方式删除动态区

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)] public static extern int
bxDual_dynamicArea_DelArea_G6_Serial(byte[] pSerialName, byte nBaudRateIndex, byte
uAreaId);
```

bxDual_dynamicArea_DelAreas_G6_Serial

返回值： 成功返回0；失败返回错误号

参数：

参数	说明
pSerialName	串口号字符串；如:byte pSerialName[] = "COM3"
nBaudRateIndex	波特率；取值为1时，代表波特率为9600;取值为2时，代表波特率为57600
uAreaCount	动态区ID数组中的个数
uAreaId	存放要删除的动态区ID数组

说明： 串口方式删除多个动态区：

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_dynamicArea_De1Areas_G6_Serial(byte[] pSerialName,
byte nBaudRateIndex, byte uAreaCount, byte[] pAreaID);
```

bxDual_dynamicArea_InsertSoundIndepend

返回值： 成功返回0；失败返回错误号

参数：

参数	说明
pIP	控制器IP
nPort	控制器端口
stSoundData	指向存放EQSoundDepend_6G结构的一段内存首地址指针
VoiceFlg	语音属性 0：此条语音从头插入队列，且停止当前正在播放的语音 1：此条语音从头插入队列，不停止当前播报的语音 2：此条语音从尾插入队列
StoreFlag	该值为 1 表示需要存储到 FLASH 中，掉电信息不丢失;该值为 0 表示需要存储到 RAM 中，掉电信息丢失

说明： 插入独立语音

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_dynamicArea_InsertSoundIndepend(byte[] pIP, int
nPort, ref EQSoundDepend_6G stSoundData, byte VoiceFlg, byte StoreFlag);
```

bxDual_dynamicArea_UpdateSoundIndepend

返回值： 成功返回0；失败返回错误号

参数：

参数	说明
pIP	控制器IP
nPort	控制器端口
stSoundData	指向存放EQSoundDepend_6G结构的一段内存首地址指针
nSoundDataCount	指示stSoundData指向内存地址空间中存放EQSoundDepend_6G个数
StoreFlag	该值为 1 表示需要存储到 FLASH 中，掉电信息不丢失;该值为 0 表示需要存储到 RAM 中，掉电信息丢失

说明： 更新独立语音命令

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_dynamicArea_UpdateSoundIndepend(byte[] pIP, int
nPort, ref EQSoundDepend_6G stSoundData, ushort nSoundDataCount, byte StoreFlag);
```

bxDual_dynamicArea_AddAreaWithTxt_5G

返回值： 成功返回0；失败返回错误号

参数：

参数	说明
pIP	控制器IP
nPort	控制器端口
color	LED屏颜色类型，详见 E_ScreenColor_G56 声明
uAreaId	区域的ID号
RunMode	动态区运行模式 0— 动态区数据循环显示。 1— 动态区数据显示完成后静止显示最后一页数据。 2— 动态区数据循环显示，超过设定时间后数据仍未更新时不再显示 3— 动态区数据循环显示，超过设定时间后数据仍未更新时显示 Logo 信息,Logo 信息即为动态区域的最后一页信息 4— 动态区数据顺序显示，显示完最后一页后就不再显示
Timeout	动态区数据超时时间，单位为秒
RelateAllPro	当该字节为 1 时，所有异步节目播放时都允许播放该动态区域；为 0 时，由接下来的规则来决定
RelateProNum	动态区域关联了多少个异步节目 一旦关联了某个异步节目，则当该异步节目播放时允许播放该动态区域，否则，不允许播放该动态区域 以下的节目编号根据 RelateProNum 的值来确定，当该值为 0 时不发送
RelateProSerial	节目编号
ImmePlay	是否立即播放 该字节为 0 时，该动态区域与异步节目一起播放 该字节为 1 时，异步节目停止播放，仅播放该动态区域 注意： 当该字节为 0 时，RelateAllPro 到 RelateProSerialN-1 的参数才有效，否则无效 当该参数为 1 时，由于不与异步节目同时播放，为控制该动态区域能及时结束，可选择 RunMode 参数为 2 或 4，当然也可通过删除该区域来实现
uAreaX	区域左上角横坐标(Top Left)，单位 Pixel
uAreaY	区域左上角纵坐标(Top Left)，单位 Pixel
uWidth	区域宽度，单位 Pixel
uHeight	区域高度，单位 Pixel
oFrame	区域边框属性 EQareaframeHeader
DisplayMode	显示方式，附录
ClearMode	退出方式/清屏方式，固定为0
Speed	速度等级
StayTime	停留时间，单位为 10ms

参数	说明
RepeatTime	重复次数，固定为1
oFont	字体属性EQfontData
fontName	字体
strAreaTxtContent	显示内容

说明： 发送一条文本信息到指定的动态区，并可以关联这个动态区到指定的节目；

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_dynamicArea_AddAreaWithTxt_5G(byte[] pIP, int
nPort, E_ScreenColor_G56 color,
    byte uAreaId,
    byte RunMode,
    ushort Timeout,
    byte RelateAllPro,
    ushort RelateProNum,
    ushort[] RelateProSerial,
    byte ImmePlay,
    ushort uAreaX, ushort uAreaY, ushort uwidth, ushort uHeight,
    EQareaframeHeader oFrame,
    //PageStyle begin-----
    byte DisplayMode,
    byte ClearMode,
    byte Speed,
    ushort StayTime,
    byte RepeatTime,
    //PageStyle End.
    //显示内容和字体格式 begin-----
    EQfontData oFont,
    byte[] fontName,
    byte[] strAreaTxtContent
    //end.
);
```

bxDual_dynamicArea_AddAreaWithTxt_5G_Serial

返回值： 成功返回0；失败返回错误号

参数：

参数	说明
pSerialName	串口号字符串；如:byte pSerialName[] = "COM3"
nBaudRateIndex	波特率；取值为1时，代表波特率为9600; 取值为2时，代表波特率为57600
color	LED屏颜色类型，详见 E_ScreenColor_G56 声明
uAreaId	区域的ID号
RunMode	动态区运行模式 0— 动态区数据循环显示。 1— 动态区数据显示完成后静止显示最后一页数据。 2— 动态区数据循环显示，超过设定时间后数据仍未更新时不再显示 3— 动态区数据循环显示，超过设定时间后数据仍未更新时显示 Logo 信息,Logo 信息即为动态区域的最后一页信息 4— 动态区数据顺序显示，显示完最后一页后就不再显示
Timeout	动态区数据超时时间，单位为秒
RelateAllPro	当该字节为 1 时，所有异步节目播放时都允许播放该动态区域；为 0 时，由接下来的规则来决定
RelateProNum	动态区域关联了多少个异步节目 一旦关联了某个异步节目，则当该异步节目播放时允许播放该动态区域，否则，不允许播放该动态区域 以下的节目编号根据 RelateProNum 的值来确定，当该值为 0 时不发送
RelateProSerial	节目编号
ImmePlay	是否立即播放 该字节为 0 时，该动态区域与异步节目一起播放 该字节为 1 时，异步节目停止播放，仅播放该动态区域 注意： 当该字节为 0 时，RelateAllPro 到 RelateProSerialN-1 的参数才有效，否则无效 当该参数为 1 时，由于不与异步节目同时播放，为控制该动态区域能及时结束，可选择 RunMode 参数为 2 或 4，当然也可通过删除该区域来实现
uAreaX	区域左上角横坐标(Top Left)，单位 Pixel
uAreaY	区域左上角纵坐标(Top Left)，单位 Pixel
uWidth	区域宽度，单位 Pixel
uHeight	区域高度，单位 Pixel
oFrame	区域边框属性 EQareaframeHeader
DisplayMode	显示方式，附录
ClearMode	退出方式/清屏方式，固定为0
Speed	速度等级
StayTime	停留时间，单位为 10ms

参数	说明
RepeatTime	重复次数，固定为1
oFont	字体属性EQfontData
fontName	字体
strAreaTxtContent	显示内容

说明： 发送一条文本信息到指定的动态区，并可以关联这个动态区到指定的节目；串口

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_dynamicArea_AddAreaWithTxt_5G_Serial(byte[]
pSerialName, int nBaudRateIndex, E_ScreenColor_G56 color,
    byte uAreaId,
    byte RunMode,
    ushort Timeout,
    byte RelateAllPro,
    ushort RelateProNum,
    ushort[] RelateProSerial,
    byte ImmePlay,
    ushort uAreaX, ushort uAreaY, ushort uwidth, ushort uHeight,
    EQareaframeHeader oFrame,
    //PageStyle begin-----
    byte DisplayMode,
    byte ClearMode,
    byte Speed,
    ushort StayTime,
    byte RepeatTime,
    //PageStyle End.
    //显示内容和字体格式 begin-----
    EQfontData oFont,
    byte[] fontName,
    byte[] strAreaTxtContent
//end.
);
```

bxDual_dynamicArea_AddAreaWithTxt_Point_5G

返回值： 成功返回0；失败返回错误号

参数：

参数	说明
pIP	控制器IP
nPort	控制器端口
color	LED屏颜色类型，详见 E_ScreenColor_G56 声明
uAreaId	区域的ID号
RunMode	<p>动态区运行模式</p> <p>0— 动态区数据循环显示。</p> <p>1— 动态区数据显示完成后静止显 示最后一页数据。</p> <p>2— 动态区数据循环显示，超过设 定时间后数据仍未更新时不再 显示</p> <p>3— 动态区数据循环显示，超过设 定时间后数据仍未更新时显示 Logo 信息,Logo 信息即为动 态区域的最后一页信息</p> <p>4— 动态区数据顺序显示，显示完 最后一页后就不再显示</p>
Timeout	动态区数据超时时间，单位为秒
RelateAllPro	当该字节为 1 时，所有异步节目 播放时都允许播放该动态区域； 为 0 时，由接下来的规则来决定
RelateProNum	<p>动态区域关联了多少个异步节目 一旦关联了某个异步节目，则当 该异步节目播放时允许播放该动 态区域，否则，不允许播放该动 态区域</p> <p>以下的节目编号根据 RelateProNum 的值来确定，当该 值为 0 时不发送</p>
RelateProSerial	节目编号
ImmePlay	<p>是否立即播放</p> <p>该字节为 0 时，该动态区域与异 步节目一起播放</p> <p>该字节为 1 时，异步节目停止播 放，仅播放该动态区域</p> <p>注意：</p> <p>当该字节为 0 时，RelateAllPro 到 RelateProSerialN-1 的参数才有 效，否则无效</p> <p>当该参数为 1 时，由于不与异步 节目同时播放，为控制该动态区 域能及时结束，可选 择 RunMode 参数为 2 或 4，当然也可通过删除 该区域来实现</p>
uAreaX	区域左上角横坐标(Top Left)，单 位 Pixel
uAreaY	区域左上角纵坐标(Top Left)，单 位 Pixel
uWidth	区域宽度，单位 Pixel
uHeight	区域高度，单位 Pixel
oFrame	区域边框属性 EQareaframeHeader
DisplayMode	显示方式，附录
ClearMode	退出方式/清屏方式，固定为0
Speed	速度等级
StayTime	停留时间，单位为 10ms

参数	说明
RepeatTime	重复次数，固定为1
oFont	字体属性EQfontData
fontName	字体
strAreaTxtContent	显示内容

说明：发送一条文本信息到指定的动态区，并可以关联这个动态区到指定的节目；

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_dynamicArea_AddAreaWithTxt_Point_5G(byte[] pIP, int
nPort, E_ScreenColor_G56 color,
    byte uAreaId,
    byte RunMode,
    ushort Timeout,
    byte RelateAllPro,
    ushort RelateProNum,
    ushort[] RelateProSerial,
    byte ImmePlay,
    ushort uAreaX, ushort uAreaY, ushort uwidth, ushort uHeight,
    ref EQareaframeHeader oFrame,
    //PageStyle begin-----
    byte DisplayMode,
    byte ClearMode,
    byte Speed,
    ushort StayTime,
    byte RepeatTime,
    //PageStyle End.
    //显示内容和字体格式 begin-----
    ref EQfontData oFont,
    byte[] fontName,
    byte[] strAreaTxtContent
    //end.
);
```

bxDual_dynamicArea_AddAreaWithTxt_Point_5G_Serial

返回值：成功返回0；失败返回错误号

参数：

参数	说明
pSerialName	串口号字符串；如:byte pSerialName[] = "COM3"
nBaudRateIndex	波特率；取值为1时，代表波特率为9600;取值为2时，代表波特率为57600
color	LED屏颜色类型，详见 E_ScreenColor_G56 声明
uAreaId	区域的ID号
RunMode	动态区运行模式 0— 动态区数据循环显示。 1— 动态区数据显示完成后静止显示最后一页数据。 2— 动态区数据循环显示，超过设定时间后数据仍未更新时不再显示 3— 动态区数据循环显示，超过设定时间后数据仍未更新时显示 Logo 信息,Logo 信息即为动态区域的最后一页信息 4— 动态区数据顺序显示，显示完最后一页后就不再显示
Timeout	动态区数据超时时间，单位为秒
RelateAllPro	当该字节为 1 时，所有异步节目播放时都允许播放该动态区域；为 0 时，由接下来的规则来决定
RelateProNum	动态区域关联了多少个异步节目 一旦关联了某个异步节目，则当该异步节目播放时允许播放该动态区域，否则，不允许播放该动态区域 以下的节目编号根据 RelateProNum 的值来确定，当该值为 0 时不发送
RelateProSerial	节目编号
ImmePlay	是否立即播放 该字节为 0 时，该动态区域与异步节目一起播放 该字节为 1 时，异步节目停止播放，仅播放该动态区域 注意： 当该字节为 0 时，RelateAllPro 到 RelateProSerialN-1 的参数才有效，否则无效 当该参数为 1 时，由于不与异步节目同时播放，为控制该动态区域能及时结束，可选择 RunMode 参数为 2 或 4，当然也可通过删除该区域来实现
uAreaX	区域左上角横坐标(Top Left)，单位 Pixel
uAreaY	区域左上角纵坐标(Top Left)，单位 Pixel
uWidth	区域宽度，单位 Pixel
uHeight	区域高度，单位 Pixel
oFrame	区域边框属性 EQareaframeHeader
DisplayMode	显示方式，附录
ClearMode	退出方式/清屏方式，固定为0
Speed	速度等级
StayTime	停留时间，单位为 10ms

参数	说明
RepeatTime	重复次数，固定为1
oFont	字体属性EQfontData
fontName	字体
strAreaTxtContent	显示内容

说明： 发送一条文本信息到指定的动态区，并可以关联这个动态区到指定的节目；串口

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_dynamicArea_AddAreaWithTxt_Point_5G_Serial(byte[]
pSerialName, int nBaundRateIndex, E_ScreenColor_G56 color,
    byte uAreaId,
    byte RunMode,
    ushort Timeout,
    byte RelateAllPro,
    ushort RelateProNum,
    ushort[] RelateProSerial,
    byte ImmePlay,
    ushort uAreaX, ushort uAreaY, ushort uwidth, ushort uHeight,
    ref EQareaframeHeader oFrame,
    //PageStyle begin-----
    byte DisplayMode,
    byte ClearMode,
    byte Speed,
    ushort StayTime,
    byte RepeatTime,
    //PageStyle End.
    //显示内容和字体格式 begin-----
    ref EQfontData oFont,
    byte[] fontName,
    byte[] strAreaTxtContent
//end.
);
```

bxDual_dynamicArea_AddAreaWithPic_5G

返回值： 成功返回0；失败返回错误号

参数：

参数	说明
pIP	控制器IP
nPort	控制器端口
color	LED屏颜色类型，详见 E_ScreenColor_G56 声明
uAreaId	区域的ID号
RunMode	动态区运行模式 0— 动态区数据循环显示。 1— 动态区数据显示完成后静止显 示最后一页数据。 2— 动态区数据循环显示，超过设 定时间后数据仍未更新时不再 显示 3— 动态区数据循环显示，超过设 定时间后数据仍未更新时显示 Logo 信息,Logo 信息即为 动 态区域的最后一页信息 4— 动态区数据顺序显示，显示完 最后一页后就不再显示
Timeout	动态区数据超时时间，单位为秒
RelateAllPro	当该字节为 1 时，所有异步节目 播放时都允许播放该动态区域； 为 0 时，由接下来的规则 来决定
RelateProNum	动态区域关联了多少个异步节目 一旦关联了某个异步节目，则当 该异步节目播放时允许播 放该动 态区域，否则，不允许播放该动 态区域 以下的节目编号根据 RelateProNum 的值来确定，当该 值为 0 时不发送
RelateProSerial	节目编号
ImmePlay	是否立即播放 该字节为 0 时，该动态区域与异 步节目一起播放 该字节为 1 时，异步节目停止播 放，仅播放该动态区域 注意： 当该字节为 0 时，RelateAllPro 到 RelateProSerialN-1 的参数才有 效，否则无效 当该参数为 1 时，由于不与异步 节目同时播放，为控制该动态区 域能及时结束，可选择 RunMode 参数为 2 或 4，当然也可通过删除 该区域来实现
uAreaX	区域左上角横坐标(Top Left)，单 位 Pixel
uAreaY	区域左上角纵坐标(Top Left)，单 位 Pixel
uWidth	区域宽度，单位 Pixel
uHeight	区域高度，单位 Pixel
oFrame	区域边框属性 EQareaframeHeader
DisplayMode	显示方式，附录
ClearMode	退出方式/清屏方式，固定为0
Speed	速度等级
StayTime	停留时间，单位为 10ms

参数	说明
RepeatTime	重复次数，固定为1
filePath	图片路径

说明： 发送一个图片到指定的动态区，并可以关联这个动态区到指定的节目；

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_dynamicArea_AddAreaWithPic_5G(byte[] pIP, int
nPort, E_ScreenColor_G56 color,
    byte uAreaId,
    byte RunMode,
    ushort Timeout,
    byte RelateAllPro,
    ushort RelateProNum,
    ushort[] RelateProSerial,
    byte ImmePlay,
    ushort uAreaX, ushort uAreaY, ushort uwidth, ushort uHeight,
    EQareaframeHeader oFrame,
    //PageStyle begin-----
    byte DisplayMode,
    byte ClearMode,
    byte Speed,
    ushort StayTime,
    byte RepeatTime,
    //PageStyle End.
    //图片路径 begin-----
    byte[] filePath
//end.
);
```

bxDual_dynamicArea_AddAreaWithPic_5G_Serial

返回值： 成功返回0；失败返回错误号

参数：

参数	说明
pSerialName	串口号字符串；如:byte pSerialName[] = "COM3"
nBaudRateIndex	波特率；取值为1时，代表波特率为9600;取值为2时，代表波特率为57600
color	LED屏颜色类型，详见 E_ScreenColor_G56 声明
uAreaId	区域的ID号
RunMode	动态区运行模式 0— 动态区数据循环显示。 1— 动态区数据显示完成后静止显示最后一页数据。 2— 动态区数据循环显示，超过设定时间后数据仍未更新时不再显示 3— 动态区数据循环显示，超过设定时间后数据仍未更新时显示 Logo 信息,Logo 信息即为动态区域的最后一页信息 4— 动态区数据顺序显示，显示完最后一页后就不再显示
Timeout	动态区数据超时时间，单位为秒
RelateAllPro	当该字节为 1 时，所有异步节目播放时都允许播放该动态区域；为 0 时，由接下来的规则来决定
RelateProNum	动态区域关联了多少个异步节目 一旦关联了某个异步节目，则当该异步节目播放时允许播放该动态区域，否则，不允许播放该动态区域 以下的节目编号根据 RelateProNum 的值来确定，当该值为 0 时不发送
RelateProSerial	节目编号
ImmePlay	是否立即播放 该字节为 0 时，该动态区域与异步节目一起播放 该字节为 1 时，异步节目停止播放，仅播放该动态区域 注意： 当该字节为 0 时，RelateAllPro 到 RelateProSerialN-1 的参数才有效，否则无效 当该参数为 1 时，由于不与异步节目同时播放，为控制该动态区域能及时结束，可选择 RunMode 参数为 2 或 4，当然也可通过删除该区域来实现
uAreaX	区域左上角横坐标(Top Left)，单位 Pixel
uAreaY	区域左上角纵坐标(Top Left)，单位 Pixel
uWidth	区域宽度，单位 Pixel
uHeight	区域高度，单位 Pixel
oFrame	区域边框属性 EQareaframeHeader
DisplayMode	显示方式，附录
ClearMode	退出方式/清屏方式，固定为0
Speed	速度等级
StayTime	停留时间，单位为 10ms

参数	说明
RepeatTime	重复次数，固定为1
filePath	图片路径

说明：发送一个图片到指定的动态区，并可以关联这个动态区到指定的节目；串口

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_dynamicArea_AddAreaWithPic_5G_Serial(byte[]
pSerialName, int nBaudRateIndex, E_ScreenColor_G56 color,
    byte uAreaId,
    byte RunMode,
    ushort Timeout,
    byte RelateAllPro,
    ushort RelateProNum,
    ushort[] RelateProSerial,
    byte ImmePlay,
    ushort uAreaX, ushort uAreaY, ushort uwidth, ushort uHeight,
    EQareaframeHeader oFrame,
    //PageStyle begin-----
    byte DisplayMode,
    byte ClearMode,
    byte Speed,
    ushort StayTime,
    byte RepeatTime,
    //PageStyle End.
    //图片路径 begin-----
    byte[] filePath
//end.
);
```

bxDual_dynamicArea_AddAreaInfos_5G

返回值：成功返回0；失败返回错误号

参数：

参数	说明
pIP	控制器IP
nPort	控制器端口
color	LED屏颜色类型，详见 E_ScreenColor_G56 声明
uAreaId	区域的ID号
RunMode	动态区运行模式 0— 动态区数据循环显示。 1— 动态区数据显示完成后静止显示最后一页数据。 2— 动态区数据循环显示，超过设定时间后数据仍未更新时不再显示 3— 动态区数据循环显示，超过设定时间后数据仍未更新时显示 Logo 信息,Logo 信息即为动态区域的最后一页信息 4— 动态区数据顺序显示，显示完最后一页后就不再显示
Timeout	动态区数据超时时间，单位为秒
RelateAllPro	当该字节为 1 时，所有异步节目播放时都允许播放该动态区域；为 0 时，由接下来的规则来决定
RelateProNum	动态区域关联了多少个异步节目 一旦关联了某个异步节目，则当该异步节目播放时允许播放该动态区域，否则，不允许播放该动态区域 以下的节目编号根据 RelateProNum 的值来确定，当该值为 0 时不发送
RelateProSerial	节目编号
ImmePlay	是否立即播放 该字节为 0 时，该动态区域与异步节目一起播放 该字节为 1 时，异步节目停止播放，仅播放该动态区域 注意： 当该字节为 0 时，RelateAllPro 到 RelateProSerialN-1 的参数才有效，否则无效 当该参数为 1 时，由于不与异步节目同时播放，为控制该动态区域能及时结束，可选择 RunMode 参数为 2 或 4，当然也可通过删除该区域来实现
uAreaX	区域左上角横坐标(Top Left)，单位 Pixel
uAreaY	区域左上角纵坐标(Top Left)，单位 Pixel
uWidth	区域宽度，单位 Pixel
uHeight	区域高度，单位 Pixel
oFrame	区域边框属性 EQareaframeHeader
nInfoCount	数据数组个数
pInfo	数据数组DynamicAreaBaseInfo_5G

说明： 发送多条信息（文本/图片）到指定的动态区，并可以关联这个动态区到指定的节目；

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_dynamicArea_AddAreaInfos_5G(byte[] pIP, int nPort,
E_ScreenColor_G56 color,
    byte uAreaId,
    byte RunMode,
    ushort Timeout,
    byte RelateAllPro,
    ushort RelateProNum,
    ushort[] RelateProSerial,
    byte ImmePlay,
    ushort uAreaX, ushort uAreaY, ushort uWidth, ushort uHeight,
    EQareaframeHeader oFrame,

    byte nInfoCount,
    DynamicAreaBaseInfo_5G[] pInfo
);
```

bxDual_dynamicArea_AddAreaInfos_5G_Point

返回值: 成功返回0; 失败返回错误号

参数:

参数	说明
pIP	控制器IP
nPort	控制器端口
color	LED屏颜色类型，详见 E_ScreenColor_G56 声明
uAreaId	区域的ID号
RunMode	动态区运行模式 0— 动态区数据循环显示。 1— 动态区数据显示完成后静止显示最后一页数据。 2— 动态区数据循环显示，超过设定时间后数据仍未更新时不再显示 3— 动态区数据循环显示，超过设定时间后数据仍未更新时显示 Logo 信息,Logo 信息即为动态区域的最后一页信息 4— 动态区数据顺序显示，显示完最后一页后就不再显示
Timeout	动态区数据超时时间，单位为秒
RelateAllPro	当该字节为 1 时，所有异步节目播放时都允许播放该动态区域；为 0 时，由接下来的规则来决定
RelateProNum	动态区域关联了多少个异步节目 一旦关联了某个异步节目，则当该异步节目播放时允许播放该动态区域，否则，不允许播放该动态区域 以下的节目编号根据 RelateProNum 的值来确定，当该值为 0 时不发送
RelateProSerial	节目编号
ImmePlay	是否立即播放 该字节为 0 时，该动态区域与异步节目一起播放 该字节为 1 时，异步节目停止播放，仅播放该动态区域 注意： 当该字节为 0 时，RelateAllPro 到 RelateProSerialN-1 的参数才有效，否则无效 当该参数为 1 时，由于不与异步节目同时播放，为控制该动态区域能及时结束，可选择 RunMode 参数为 2 或 4，当然也可通过删除该区域来实现
uAreaX	区域左上角横坐标(Top Left)，单位 Pixel
uAreaY	区域左上角纵坐标(Top Left)，单位 Pixel
uWidth	区域宽度，单位 Pixel
uHeight	区域高度，单位 Pixel
oFrame	区域边框属性 EQareaframeHeader
nInfoCount	数据数组个数
pInfo	数据数组DynamicAreaBaseInfo_5G

说明： 增加多条信息（文本/图片）到指定的动态区，并可以关联这个动态区到指定的节目；

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_dynamicArea_AddAreaInfos_5G_Point(byte[] pIP, int
nPort, E_ScreenColor_G56 color,
    byte uAreaId,
    byte RunMode,
    ushort Timeout,
    byte RelateAllPro,
    ushort RelateProNum,
    ushort[] RelateProSerial,
    byte ImmePlay,
    ushort uAreaX, ushort uAreaY, ushort uWidth, ushort uHeight,
    EQareaframeHeader oFrame,

    byte nInfoCount,
    DynamicAreaBaseInfo_5G[] pInfo

);
```

bxDual_dynamicArea_AddAreaInfos_5G_Serial

返回值: 成功返回0; 失败返回错误号

参数:

参数	说明
pSerialName	串口号字符串；如:byte pSerialName[] = "COM3"
nBaudRateIndex	波特率；取值为1时，代表波特率为9600;取值为2时，代表波特率为57600
color	LED屏颜色类型，详见 E_ScreenColor_G56 声明
uAreaId	区域的ID号
RunMode	动态区运行模式 0— 动态区数据循环显示。 1— 动态区数据显示完成后静止显示最后一页数据。 2— 动态区数据循环显示，超过设定时间后数据仍未更新时不再显示 3— 动态区数据循环显示，超过设定时间后数据仍未更新时显示 Logo 信息,Logo 信息即为动态区域的最后一页信息 4— 动态区数据顺序显示，显示完最后一页后就不再显示
Timeout	动态区数据超时时间，单位为秒
RelateAllPro	当该字节为 1 时，所有异步节目播放时都允许播放该动态区域；为 0 时，由接下来的规则来决定
RelateProNum	动态区域关联了多少个异步节目 一旦关联了某个异步节目，则当该异步节目播放时允许播放该动态区域，否则，不允许播放该动态区域 以下的节目编号根据 RelateProNum 的值来确定，当该值为 0 时不发送
RelateProSerial	节目编号
ImmePlay	是否立即播放 该字节为 0 时，该动态区域与异步节目一起播放 该字节为 1 时，异步节目停止播放，仅播放该动态区域 注意： 当该字节为 0 时，RelateAllPro 到 RelateProSerialN-1 的参数才有效，否则无效 当该参数为 1 时，由于不与异步节目同时播放，为控制该动态区域能及时结束，可选择 RunMode 参数为 2 或 4，当然也可通过删除该区域来实现
uAreaX	区域左上角横坐标(Top Left)，单位 Pixel
uAreaY	区域左上角纵坐标(Top Left)，单位 Pixel
uWidth	区域宽度，单位 Pixel
uHeight	区域高度，单位 Pixel
oFrame	区域边框属性 EQareaframeHeader
nInfoCount	数据数组个数
pInfo	数据数组DynamicAreaBaseInfo_5G

说明： 初始化动态库

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_dynamicArea_AddAreaInfos_5G_Serial(byte[]
pSerialName, int nBaudRateIndex, E_ScreenColor_G56 color,
    byte uAreaId,
    byte RunMode,
    ushort Timeout,
    byte RelateAllPro,
    ushort RelateProNum,
    ushort[] RelateProSerial,
    byte ImmePlay,
    ushort uAreaX, ushort uAreaY, ushort uWidth, ushort uHeight,
    EQareaframeHeader oFrame,

    byte nInfoCount,
    DynamicAreaBaseInfo_5G[] pInfo
);
```

bxDual_dynamicArea_DelArea_5G

返回值：成功返回0；失败返回错误号

参数：

参数	说明
pIP	控制器IP
nPort	控制器端口
uAreaId	区域ID, 0xff:删除所有区域

说明：删除动态区

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_dynamicArea_DelArea_5G(byte[] pIP, int nPort, byte
uAreaId);
```

bxDual_dynamicArea_DelAreaS_5G

返回值：成功返回0；失败返回错误号

参数：

参数	说明
pIP	控制器IP
nPort	控制器端口
uAreaCount	动态区ID数组中的个数
uAreaId	存放要删除的动态区ID数组

说明：删除多个动态区

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_dynamicArea_DelAreaS_5G(byte[] pIP, int nPort, byte
uAreaCount, byte[] pAreaID);
```

bxDual_dynamicArea_DelArea_G5_Serial

返回值：成功返回0；失败返回错误号

参数：

参数	说明
pSerialName	串口号字符串；如:byte pSerialName[] = "COM3"
nBaudRateIndex	波特率；取值为1时，代表波特率为9600;取值为2时，代表波特率为57600
uAreaId	区域ID，0xff:删除所有区域

说明：串口方式删除动态区

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_dynamicArea_DelArea_G5_Serial(byte[] pSerialName,
byte nBaudRateIndex, byte uAreaId);
```

bxDual_dynamicArea_DelAreaS_G5_Serial

返回值：成功返回0；失败返回错误号

参数：

参数	说明
pSerialName	串口号字符串；如:byte pSerialName[] = "COM3"
nBaudRateIndex	波特率；取值为1时，代表波特率为9600; 取值为2时，代表波特率为57600
uAreaCount	动态区ID数组中的个数
uAreaId	存放要删除的动态区ID数组

说明： 串口方式删除多个动态区：

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)] public static extern int
bxDual_dynamicArea_DeAreaS_G5_Serial(byte[] pSerialName, byte nBaudRateIndex, byte
uAreaCount, byte[] pAreaID);
```

bxDual_program_addProgram_G6

返回值： 成功返回0；失败返回错误号

参数：

参数	说明
programH	参考结构体EQprogramHeader_G6

说明： 添加节目

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_program_addProgram_G6(ref EQprogramHeader_G6
programH);
```

bxDual_program_addPlayPeriodGrp_G6

返回值： 成功返回0；失败返回错误号

参数：

参数	说明
programH	参考结构体EQprogramHeader_G6

说明： 添加节目播放时段

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_program_addPlayPeriodGrp_G6(ref EQprogramppGrp_G56
header);
```

bxDual_program_deleteProgram_G6

返回值：成功返回0；失败返回错误号

参数：无

说明：删除节目

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_program_deleteProgram_G6();
```

bxDual_program_freeBuffer_G6

返回值：成功返回0；失败返回错误号

参数：

参数	说明
programH	参考结构体EQprogram_G6

说明：释放生成节目文件的缓冲区

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_program_freeBuffer_G6(ref EQprogram_G6 program);
```

bxDual_program_changeProgramParams_G6 【】

返回值：成功返回0；失败返回错误号

参数：

参数	说明
programH	参考结构体EQprogramHeader_G6

说明：修改已添加节目的一些参数

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_program_changeProgramParams_G6(ref
EQprogramHeader_G6 programH);
```

bxDual_program_addFrame_G6

返回值：成功返回0；失败返回错误号

参数：

参数	s说明
sfHeader	参考结构体 EQscreenframeHeader_G6
picPath	添加的边框图片路径

说明：节目添加边框

* 注：节目添加边框后，区域的坐标随即发生变化，添加区域的时候需注意

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_program_addFrame_G6(ref EQscreenframeHeader_G6
sfHeader, byte[] picPath);
```

bxDual_program_changeFrame_G6 【】

返回值：成功返回0；失败返回错误号

参数：

参数	s说明
sfHeader	参考结构体 EQscreenframeHeader_G6
picPath	添加的边框图片路径

说明：节目修改已添加边框的一些参数

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_program_changeFrame_G6(ref EQscreenframeHeader_G6
sfHeader, byte[] picPath);
```

bxDual_program_removeFrame_G6 【】

返回值：成功返回0；失败返回错误号

参数：无

说明：节目去掉边框

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_program_removeFrame_G6();
```

bxDual_program_addArea_G6

返回值：成功返回0；失败返回错误号

参数：

参数	说明
areaID	区域的ID号
aheader	参考结构体[EQareaHeader_G6](

说明： 节目添加区域

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_program_addArea_G6(ushort areaID, ref
EQareaHeader_G6 aheader);
```

bxDual_program_deleteArea_G6 【】

返回值：成功返回0；失败返回错误号

参数：

参数	说明
areaID	区域的ID号

说明： 节目删除已添加的区域

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_program_deleteArea_G6(ushort areaID);
```

bxDual_program_MoveArea_G6 【】

返回值：成功返回0；失败返回错误号

参数：

参数	说明
areaID	区域的ID号
x	区域left坐标
y	区域top坐标
width	区域宽度
height	区域高度

说明： 改变区域坐标大小

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_program_MoveArea_G6(ushort areaID, int x, int y,
int width, int height);
```

bxDual_program_picturesAreaAddTxt_G6

返回值： 成功返回0；失败返回错误号

参数：

参数	说明
areaID	区域的ID号
str	需要画的字符
fontName	字体名称
pheader	参考结构体 EQpageHeader_G6

说明： 画文字到图文区域

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_program_picturesAreaAddTxt_G6(ushort areaID, byte[]
str, byte[] fontName, ref EQpageHeader_G6 pheader);
```

bxDual_program_picturesAreaChangeTxt_G6 【】

返回值： 成功返回0；失败返回错误号

参数：

参数	说明
areaID	区域的ID号
str	需要画的字符
pheader	参考结构体 EQpageHeader_G6

说明： 修改图文区域已添加过的文字内容及EQpageHeader_G6结构体中的参数 * 注： * 如需修改字体，需要将区域删除，重新添加区域和文字

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_program_picturesAreaChangeTxt_G6(ushort areaID,
byte[] str, ref EQpageHeader_G6 pheader);
```

bxDual_program_fontPath_picturesAreaAddTxt_G6

返回值： 成功返回0；失败返回错误号

参数：

参数	说明
areaID	区域的ID号
str	需要画的字符
fontName	字体绝对路径加字库文件名称
pheader	参考结构体[EQpageHeader_G6](

说明： 图文区添加字符串--使用字库

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_program_fontPath_picturesAreaAddTxt_G6(ushort
areaID, byte[] str, byte[] fontPathName, ref EQpageHeader_G6 pheader);
```

bxDual_program_fontPath_picturesAreaChangeTxt_G6 【】

返回值： 成功返回0；失败返回错误号

参数：

参数	说明
areaID	区域的ID号
str	需要画的字符
pheader	参考结构体EQpageHeader_G6(

说明： 图文区修改字符串--使用字库

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_program_fontPath_picturesAreaChangeTxt_G6(ushort
areaID, byte[] str, ref EQpageHeader_G6 pheader);
```

bxDual_program_pictureAreaAddPic_G6

返回值： 成功返回0；失败返回错误号

参数：

参数	说明
areaID	区域的ID号
picID	图片的ID号
pheader	参考结构体 EQpageHeader_G6
picPath	添加的图片路径

说明： 添加图片到图文区域

* 注：下位机播放图片的次序与picID一致，即最先播放picID为0的图片，依次播放

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_program_pictureAreaAddPic_G6(ushort areaID, ushort
picID, ref EQpageHeader_G6 pheader, byte[] picPath);
```

bxDual_program_backGroundPic_G6 【】

返回值： 成功返回0；失败返回错误号

参数：

参数	说明
areaID	区域的ID号
picID	图片的ID号
pheader	参考结构体 EQpageHeader_G6
picPath	添加的图片路径

说明： 添加图片到图文区域

* 注：下位机播放图片的次序与picID一致，即最先播放picID为0的图片，依次播放

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_program_backGroundPic_G6(ushort areaID, ushort
picID, ref EQpageHeader_G6 pheader, byte[] picPath);
```

bxDual_program_backGroundColor_G6 【】

返回值： 成功返回0；失败返回错误号

参数：

参数	说明
areaID	区域的ID号
pheader	参考结构体 EQpageHeader_G6
BGColor	背景色

说明： 添加图片到图文区域

* 注：下位机播放图片的次序与picID一致，即最先播放picID为0的图片，依次播放

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_program_backGroundColor_G6(ushort areaID, ref
EQpageHeader_G6 pheader, int BGColor);
```

bxDual_program_pictureAreaChangePic_G6 【】

返回值： 成功返回0；失败返回错误号

参数：

参数	说明
areaID	区域的ID号
picID	图片的ID号
pheader	参考结构体 EQpageHeader_G6
picPath	添加的图片路径

说明：修改当前picID对应的图片和一些参数

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_program_pictureAreaChangePic_G6(ushort areaID,
ushort picID, ref EQpageHeader_G6 pheader, byte[] picPath);
```

bxDual_program_pictureAreaEnableSound_G6()

返回值：成功返回0；失败返回错误号

参数：

参数	说明
areaID	区域的ID号
picID	图片的ID号
pheader	参考结构体 EQpageHeader_G6
picPath	添加的图片路径

说明：图文分区使能语音播放

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_program_pictureAreaEnableSound_G6(ushort areaID,
EQPicAreaSoundHeader_G6 shheader, byte[] soundData);
```

bxDual_program_pictureAreaChangeSoundSettings_G6 【】

返回值：成功返回0；失败返回错误号

参数：

参数	说明
areaID	区域ID
EQPicAreaSoundHeader_G6	参考结构体EQPicAreaSoundHeader_G6
soundData	语音数据

说明： 图文分区修改语音播放的一些参数或数据

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_program_pictureAreaChangeSoundSettings_G6(ushort
areaID, EQPicAreaSoundHeader_G6 shheader, byte[] soundData);
```

bxDual_program_pictureAreaDisableSound_G6 【】

返回值： 成功返回0；失败返回错误号

参数：

参数	说明
areaID	区域ID

说明： 图文分区取消语音播放

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_program_pictureAreaDisableSound_G6(ushort areaID);
```

bxDual_program_timeAreaSetBattleTime_G6 【】

返回值： 成功返回0；失败返回错误号

参数：

参数	说明
areaID	区域ID
header	EQTimeAreaBattle_G6

说明： 时间分区设置战斗时间和战斗时间的启动模式

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_program_timeAreaSetBattleTime_G6(ushort areaID, ref
EQTimeAreaBattle_G6 header);
```

bxDual_program_timeAreaCancleBattleTime_G6 【】

返回值：成功返回0；失败返回错误号

参数：

参数	说明
areaID	区域ID

说明：取消后的时间分区将作为普通时间

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_program_timeAreaCancleBattleTime_G6(ushort areaID);
```

bxDual_program_timeAreaAddContent_G6

返回值：成功返回0；失败返回错误号

参数：

参数	说明
areaID	区域的ID号
timeData	详情请见时间区数据格式结构体[EQtimeAreaData_G56](

说明：时间分区添加时间等内容

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_program_timeAreaAddContent_G6(ushort areaID, ref
EQtimeAreaData_G56 timeData);
```

bxDual_program_timeAreaChangeContent_G6 【】

返回值：成功返回0；失败返回错误号

参数：

参数	说明
areaID	区域的ID号
timeData	详情请见时间区数据格式结构体[EQtimeAreaData_G56](

说明： 时间分区修改时间等内容

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_program_timeAreaChangeContent_G6(ushort areaID, ref
EQtimeAreaData_G56 timeData);
```

bxDual_program_fontPath_timeAreaAddContent_G6

返回值： 成功返回0；失败返回错误号

参数：

参数	说明
areaID	区域的ID号
timeData	详情请见时间区数据格式结构体[EQtimeAreaData_G56](

说明： 时间分区添加模拟时钟

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_program_fontPath_timeAreaAddContent_G6(ushort
areaID, ref EQtimeAreaData_G56 timeData);
```

bxDual_program_timeAreaChangeAnalogClock_G6 【】

返回值： 成功返回0；失败返回错误号

参数：

参数	说明
areaID	区域的ID号
header	详情见 EQAnalogClockHeader_G56 结构体
cStyle	表盘样式，详情见 E_ClockStyle
cColor	表盘颜色，详情见 E_Color_G56 通过此枚举值可以直接配置七彩色，如果大于枚举范围使用RGB888模式

说明： 时间分区修改模拟时钟的一些设置参数

函数:

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_program_timeAreaChangeAnalogClock_G6(ushort areaID,
ref EQAnalogClockHeader_G56 header, E_ClockStyle cStyle, ref C_ClockColor_G56 cColor);
```

bxDual_program_timeAreaChangeDialPic_G6 【】

返回值: 成功返回0; 失败返回错误号

参数:

参数	说明
areaID	区域的ID号
picPath	表盘图片位置

说明: 时间分区从外部添加表盘图片

函数:

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_program_timeAreaChangeDialPic_G6(ushort areaID,
byte[] picPath);
```

bxDual_program_timeAreaRemoveDialPic_G6 【】

返回值: 成功返回0; 失败返回错误号

参数:

参数	说明
areaID	区域的ID号

说明: 时间分区移除添加的表盘图片

函数:

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_program_timeAreaRemoveDialPic_G6(ushort areaID);
```

bxDual_program_IntegrateProgramFile_G6

返回值: 成功返回0; 失败返回错误号

参数:

参数	说明
program	参考结构体[EQprogram_G6](

说明： 合成节目文件返回节目文件属性及地址

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_program_IntegrateProgramFile_G6(ref EQprogram_G6
program);
```

4.1 bxDual_Start_Server

返回值： 成功返回服务器启动句柄

参数：

参数	说明
port	服务器访问端口

说明： 启动服务器

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_Start_Server(int port);
```

4.1 bxDual_Stop_Server

返回值： 成功返回0；失败返回错误号

参数：

参数	说明
pServer	服务器启动句柄

说明： 关闭服务器

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_Stop_Server(int pServer);
```

4.1 bxDual_Get_CardList

返回值： 成功返回控制卡上线个数

参数：

参数	说明
cards	一个控制卡数据20个长度，前16位数据是控制卡网络ID编号或者条形码，后4位是分配端口

说明： 获取上线列表数据

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_Get_CardList(byte[] cards);
```

4.1 Get_Port_Barcode

返回值： 成功返回通信端口

参数：

参数	说明
barcode	控制卡条形码或者网络ID编号

说明： 获取动态库分配通信端口

函数：

```
[DllImport("bx_sdk_dual.dll", CharSet = CharSet.Unicode)]
public static extern int bxDual_Get_Port_Barcode(byte[] barcode);
```

附录一 枚举参数

屏型 E_ScreenColor_G56

```
public enum E_ScreenColor_G56
{
    eSCREEN_COLOR_SINGLE = 1,    //单基色
    eSCREEN_COLOR_DOUBLE,        //双基色
    eSCREEN_COLOR_THREE,         //七彩色
    eSCREEN_COLOR_FULLCOLOR,     //全彩色
}
```

屏型 E_DoubleColorPixel_G56

```
public enum E_DoubleColorPixel_G56 : int
{
    eDOUBLE_COLOR_PIXTYPE_1 = 1, //双基色1: G+R
    eDOUBLE_COLOR_PIXTYPE_2,     //双基色2: R+G
}
```

单多行 E_arrMode

```
public enum E_arrMode : int
{
    eSINGLELINE,    //单行
    eMULTILINE,    //多行
}
```

日期格式 E_DateStyle

```
public enum E_DateStyle : int
{
    eYYYY_MM_DD_MINUS,    //YYYY-MM-DD
    eYYYY_MM_DD_VIRGURE,  //YYYY/MM/DD
    eDD_MM_YYYY_MINUS,    //DD-MM-YYYY
    eDD_MM_YYYY_VIRGURE,  //DD/MM/YYYY
    eMM_DD_MINUS,         //MM-DD
    eMM_DD_VIRGURE,       //MM/DD
    eMM_DD_CHS,           //MM月DD日
    eYYYY_MM_DD_CHS,      //YYYY年MM月DD日
}
```

时间格式 E_TimeStyle

```
public enum E_TimeStyle : int
{
    eHH_MM_SS_COLON,    //HH:MM:SS
    eHH_MM_SS_CHS,      //HH时MM分SS秒
    eHH_MM_COLON,       //HH:MM
    eHH_MM_CHS,         //HH时MM分
    eAM_HH_MM,          //AM HH:MM
    eHH_MM_AM,          //HH:MM AM
};
```

星期格式 E_WeekStyle

```
public enum E_WeekStyle : int
{
    eMonday = 1,        //Monday
    eMon,               //Mon.
    eMonday_CHS,        //星期一
}
```

颜色取值 E_Color_G56

```
public enum E_Color_G56
{
    eBLACK,      //黑色
    eRED,        //红色
    eGREEN,      //绿色
    eYELLOW,     //黄色
    eBLUE,       //蓝色
    eMAGENTA,    //品红/洋红
    eCYAN,       //青色
    eWHITE,      //白色
} //5代时间区只支持四种颜色
```

表盘样式 E_ClockStyle

```
public enum E_ClockStyle
{
    eLINE,      //线形
    eSQUARE,    //方形
    eCIRCLE,    //圆形
};//表盘样式
```

图文区文字方向---暂不支持 E_txtDirection

```
public enum E_txtDirection
{
    pNORMAL,      //正常
    pROTATERIGHT, //向右旋转
    pMIRROR,      //镜像
    pROTATELEFT,  //向左旋转
};//图文区文字方向---暂不支持
```

附录二 结构体参数详细说明

ConfigFile

```
[StructLayoutAttribute(LayoutKind.Sequential, CharSet = CharSet.Ansi, Pack = 1)]
public struct ConfigFile
{
    public byte FileType;
    public byte[] ControllerName;
    ushort Address;
    public byte Baudrate;
    ushort Screenwidth;
    ushort ScreenHeight;
    public byte Color;
    public byte MirrorMode;
    public byte OEPol;
    public byte DAPol;
    public byte RowOrder;
```

```
    public byte FreqPar;  
    public byte OEwidth;  
    public byte OEAngle;  
    public byte FaultProcessMode;  
    public byte CommTimeoutValue;  
    public byte RunningMode;  
    public byte LoggingMode;  
    public byte GrayFlag;  
    public byte CascadeMode;  
    public byte Default_brightness;  
    public byte HUBConfig;  
    public byte Language;  
    public byte Backup;  
    ushort CRC16;  
}
```

[illegible]

