

博聪人脸图片上传 SDK 说明

(版本 V1.9)

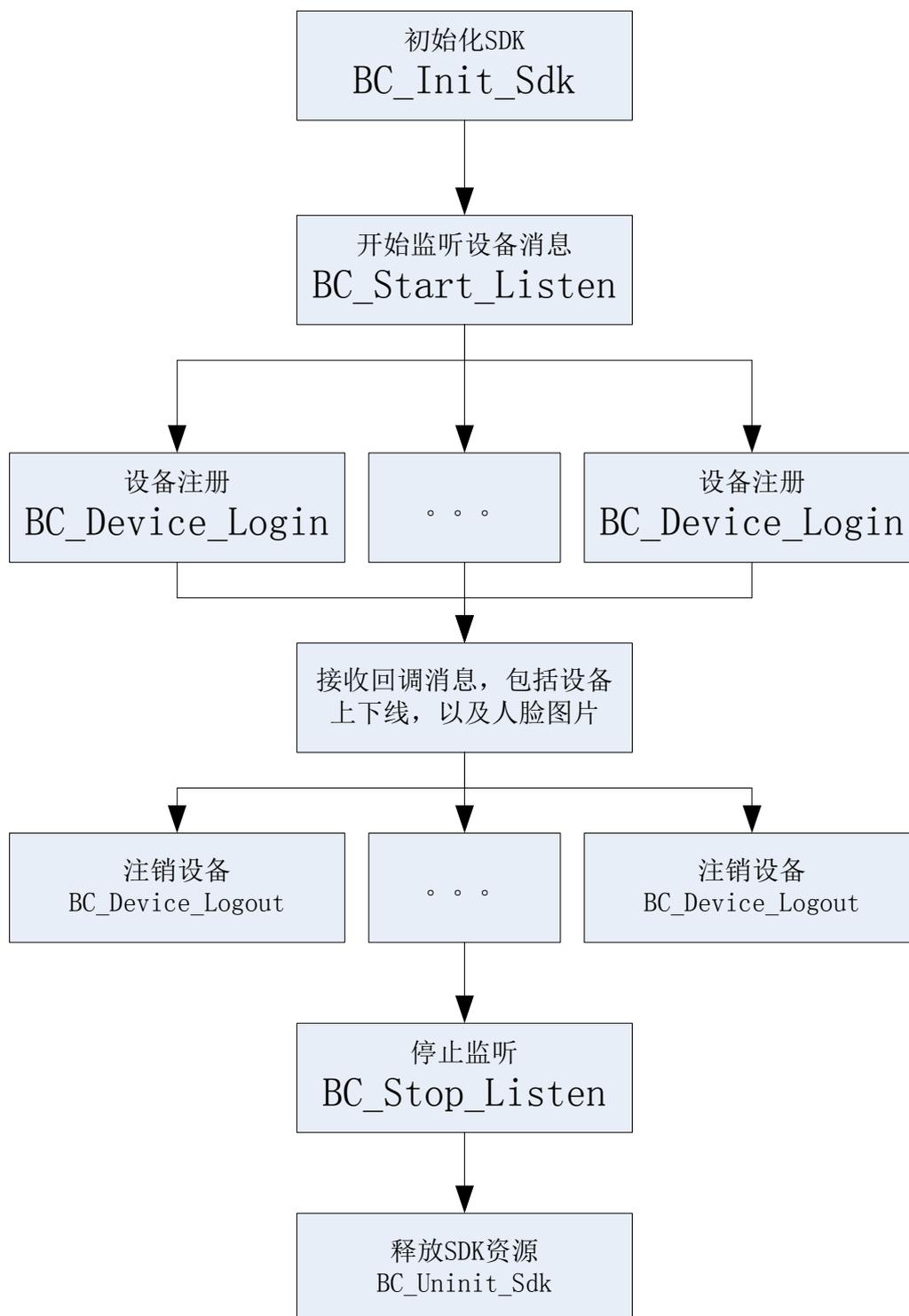
修改记录：

版本	日期	描述
V1.0	2019-02-14	第一版
V1.1	2019-02-15	将设备状态回调与业务回调 区分成两个回调
V1.5	2019-06-13	增加监听 ip 地址； 支持多个平台连接；
V1.9	2021-9-9	增加日志接口

目录

一、	SDK 接口调用主要流程	4
二、	函数说明	5
三、	结构体说明	11

一、 SDK 接口调用主要流程



应用层首先调用初始化函数，然后调用开始监听函数，允许SDK接收设备上

传的消息。应用层调用设备注册函数，完成设备注册，当有多个设备时，可以一直循环调用，直到所有设备注册完毕。

设备注册完后，应用层即可通过回调函数接收设备相关消息，包括设备状态，以及人脸图片。

对于不再希望接收消息的设备，可以调用注销函数注销该设备。

如果设备通知应用层设备离线，则应用层应先调用该设备的注销函数，然后间隔一段时间如15秒以后尝试注册一次设备，直到注册成功为止。

系统在退出时，先停止监听，再释放SDK资源。

二、 函数说明

1. 设置日志保存路径BC_Set_LogFullPath

SDK_API void BC_Set_LogFullPath(char *fullPath);

说明：

当需要保存 SDK 打印日志时调用,调用接口后 SDK 将日志保存在传入路径中。

不调用此接口将不保存日志。

参数：

[in] fullPath 保存日志全路径 比如: /home/sdk.txt

2. 初始化BC_Init_Sdk

SDK_API int BC_Init_Sdk(BC_FaceMsg_callback

pMsgcallback,BC_DeviceState_callback pDevcallback, unsigned long
ulUserData);

说明：

初始化。调用SDK其他函数的前提。

参数：

[in] BC_FaceMsg_callbackpMsgcallback:人脸图片信息回调函数.

[in] BC_DeviceState_callback pDevcallback:设备在线信息回调函数.

[in] unsigned long ulUserData:用户数据，在回调时，再回传给用户。

返回值：

0表示成功，1表示失败。

3. 业务回调函数

```
typedef void (*BC_FaceMsg_callback)( BC_STRUCT_FACEINFO  
*pMsgInfo,unsigned int iMsgLen,unsigned long ulUserData);
```

说明：

当设备有人脸图片上传时，通过本函数回传给应用层。

请不要在回调函数中做多余的操作,建议尽快的结束。

参数：

[out]pMsgInfo 详见结构体定义:BC_STRUCT_FACEINFO

[out]unsigned int iMsgLen：消息 pMsgInfo 的长度，以字节为单位。

[out]unsigned long ulUserData：应用层在调用 BC_Init_Sdk 时，传入的
ulUserData 值。

4. 设备状态回调函数

```
typedef void (*BC_DeviceState_callback)( BC_STRUCT_DEVICE_STATE  
*pMsgInfo,unsigned int iMsgLen,unsigned long ulUserData);
```

说明：

当设备状态改变时，通过本函数回传给应用层。状态没变则不调用。

请不要在回调函数中做多余的操作,建议尽快的结束。

当设备离线后，SDK 会通过本回调通知应用层，但不会主动尝试与设备建立连接，需要应用层主动调用设备注册函数。因此，当设备离线后，应用层不会再收到该设备业务数据，直到重新注册成功。

参数：

[out]pMsgInfo 详见结构体定义:BC_STRUCT_DEVICE_STATE

[out]unsigned int iMsgLen：消息 pMsgInfo 的长度，以字节为单位。

[out]unsigned long ulUserData：应用层在调用 BC_Init_Sdk 时，传入的 ulUserData 值。

5. 注册设备BC_Device_Login

```
SDK_API intBC_Device_Login(BC_STRUCT_LOGININFO *pstLogininfo,  
BC_STRUCT_DEVICEINFO *pstDeviceinfo);
```

说明：

1. 一台设备同时最多允许64个用户注册。

2. SDK支持同时注册2048个设备。
3. 该函数为同步函数，超时时间为结构体BC_STRUCT_LOGININFO中
iLoginTimeout。

参数：

[in]BC_STRUCT_LOGININFO *pstLogininfo：登录参数。包括设备地址，
登录用户，密码。详见结构体:BC_STRUCT_LOGININFO。

[out] BC_STRUCT_DEVICEINFO *pstDeviceinfo：设备返回的sessionID,
设备能力类型，站点名称等,详见结构体BC_STRUCT_DEVICEINFO。

当函数返回成功时，pstDeviceinfo内容有效。

返回值：

成功返回0，注册失败返回-1，用户名密码错误返回1，超时返回2，超过SDK
允许设备数返回3，超过设备最大连接数量返回4。

当设备不在线时，注册不成功。

6. 设备注销 BC_Device_Logout

函数: SDK_API int BC_Device_Logout(unsigned int uiSessionID);

说明：

设备注销，不再接收该设备的任何消息。

参数:

[in]uiSessionID BC_Device_Login返回的ID

返回值:

0表示成功，1表示失败

7. 启动监听，接收设备主动上报的图片等信息BC_Start_Listen

SDK_API int BC_Start_Listen(unsigned short usPort);

说明：

启动sdk服务端监听，开始允许接收设备的消息。

参数：

[in] unsigned short usPort:本机监听端口号。

返回值：

成功返回0。其他值表示失败。

8. 停止监听 BC_Stop_Listen

SDK_API int BC_Stop_Listen();

说明：

停止sdk服务端监听，不再接收设备的任何消息。

参数：

无

返回值：

0表示成功。1表示失败。

9. 释放SDK资源 BC_Uninit_Sdk

SDK_API int BC_Uninit_Sdk();

说明：

反初始化。系统退出时调用。

参数：

无

返回值：

0表示成功。1表示失败。

10. 获取设备能力

```
SDK_API intBC_Get_DeviceAbility (unsigned int uiSessionID,unsigned int *puiAbility);
```

说明：

获取设备能力类型。登录设备后可以调用，

参数：

[in] unsigned int uiSessionID:设备成功登录后获取的sessionID。

[out] unsigned int *puiAbility:设备能力值。

0表示人脸抓拍，1表示车辆 抓拍，2表示人体抓拍。详见

ENUM_DEVICEABILITY

当函数返回成功时，*puiAbility内容有效。

返回值：

0表示成功。1表示失败。

11. 获取SDK版本号BC_Get_SDKVERSION

```
SDK_API intBC_Get_SDKVERSION(char *pchVersion,int ichlen);
```

说明：

获取SDK版本号。

参数：

[in]int ichlen:pchVersion指向区域的空间大小。不小于64个字节。

[out]char *pchVersion:SDK版本号。

当函数返回成功时，*pchVersion内容有效。

返回值：

0表示成功。1表示失败。

12. 设置平台监听IP地址BC_Set_ListenIp

```
SDK_API intBC_Set_ListenIp(char *chIpAddr);
```

说明：设置平台监听IP，必须在BC_Init_Sdk之后调用，在 BC_Start_Listen之前调用。

如果不设置，则在平台所有IP上监听，如果设置，则只在设置的IP上监听。

参数：

[in]char *chIpAddr：平台IP地址

返回值：

0表示成功。-1表示失败。

三、 结构体说明

typedef struct

```
{  
  
    char sDeviceIP[64];  
  
    char sUserName[64];  
  
    char sPassword[64];  
  
    intiLoginTimeout;//同步 login 的超时时间，单位 ms.500 表示 500ms  
}BC_STRUCT_LOGININFO;
```

typedef struct

```
{  
  
    unsigned int uiDeviceSessionID;  
  
    unsigned int uiDeviceAbility;  
  
    char sDeviceName [128];  
}BC_STRUCT_DEVICEINFO;
```

typedef struct

```
{  
  
    int top;  
  
    int bottom;  
  
    int left;  
  
    int right;  
}BC_STRUCT_RECT;
```

说明：人脸图位于背景图中的坐标，单位为像素。背景图默认为 1920x1080.

```
typedef struct
{
    unsigned int uiSessionID;// BC_Device_Login 返回的 ID
    unsigned int uiState;//设备当前状态。=0，表示设备下线，=1，表示设备
    上线。
}BC_STRUCT_DEVICE_STATE
```

```
typedef struct{
    unsigned int uiSessionID;// BC_Device_Login 返回的 ID
    unsigned char *smallpicbuf;//人脸图数据指针,数据格式为 JPG
    int smallpicbufsize;//人脸图数据大小
    unsigned char *bigpicbuf;//背景图数据指针，数据格式为 JPG
    int bigpicbufsize;//背景图数据大小
    unsigned long faceID;//人脸图唯一标识号。前端设备启动后从 0 开始计
    数，在该设备下唯一。
    unsigned char timestring[64];//人脸图产生时间(精确到秒),格式如下：
    20190214131400，表示 2019 年 2 月 14 号下午 13 点 14 分 0 秒
    BC_STRUCT_RECT rect;//人脸图在背景图中的坐标。
}BC_STRUCT_FACEINFO;
```

```
typedef enum
```

```
{  
    FACE = 0,  
    VEHICLE,  
    PERSON,  
}ENUM_DEVICEABILITY;//设备能力类型定义 :0 表示人脸抓拍 ,1 表示车辆 抓  
拍 , 2 表示人体抓拍。
```