



PhoenixUSBPro 二次开发 SDK 使用指南

版本号：1.3

发布时间：2021-01-15

版本历史

版本	日期	责任人	版本描述
1.3	2021-01-15	AWA1695	1.增加第 1 章。 2.更新文档模板和排版。
1.2	2019-12-10	Allwinner	USBProProc_IF 新增 Release 接口, 释放 usb 消息的注册。
1.1	2016-10-28	Allwinner	OnBurnOver 回调接口增加 wParam 参数返回设备信息。
1.0	2014-09-05	Allwinner	创建文档。



目录

版本历史	1
目录	2
表格目录	4
1 前言	5
1.1 文档简介.....	5
1.2 目标读者.....	5
1.3 适用范围.....	5
1.4 文档约定.....	5
1.4.1 标志说明.....	5
2 插件使用操作说明.....	6
2.1 包含插件头文件.....	6
2.2 拷贝插件库至工程输出目录.....	6
2.3 引入头文件.....	6
2.4 定义插件变量.....	6
2.5 加载插件.....	7
2.6 捕获设备插拔消息.....	8
3 接口说明	9
3.1 接口结构体定义.....	9
3.2 各接口作用.....	9
3.2.1 pInitalizeProc.....	9
3.2.2 pSetImageFile.....	9
3.2.3 pSetKeyFile.....	9
3.2.4 pPreLiveMessage	10
3.2.5 pStopWork	10
3.2.6 pStartWork.....	10
3.2.7 pGetDynamicIF.....	10
3.2.8 pGetDynamicHandle	10
3.2.9 pLoadCfg	11
3.2.10 pGetVersion	11
3.2.11 pSetReboot.....	11
3.2.12 pSetBurnPrivate	11
3.2.13 pRelease	11
4 回调接口说明.....	12

5 信息号说明.....	14
6 目录说明	16



表格目录

表 3-1	pInitIizeProc 接口说明	9
表 3-2	pSetImageFile 接口说明.....	9
表 3-3	pSetKeyFile 接口说明.....	9
表 3-4	pPreLiveMessage 接口说明	10
表 3-5	pStopWork 接口说明	10
表 3-6	pStartWork 接口说明.....	10
表 3-7	pGetDynamicIF 接口说明.....	10
表 3-8	pGetDynamicHandle 接口说明	10
表 3-9	pLoadCfg 接口说明	11
表 3-10	pGetVersion 接口说明	11
表 3-11	pSetReboot 接口说明	11
表 3-12	pSetBurnPrivate 接口说明.....	11
表 3-13	pRelease 接口说明.....	11

1 前言

1.1 文档简介

本文档介绍了 PhoenixUSBPro 针对 USBProcPro 插件的二次开发。

1.2 目标读者

需要对 PhoenixUSBPro 进行二次开发的开发人员。





1.3 适用范围

仅针对 PhoenixUSBPro 中的 USBProcPro 插件。

1.4 文档约定

1.4.1 标志说明

本文档采用各种醒目的标志来表示在操作过程中应该特别注意的地方，这些标志的含义如下：

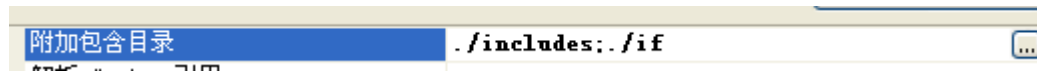
标识	说明
 警告	该标志后的说明应给予格外关注，如果不遵守，可能会导致人员受伤或死亡。
 注意	提醒操作中应注意的事项。不当的操作可能会损坏器件，影响可靠性、降低性能等。
 说明	为准确理解文中指令、正确实施操作而提供的补充或强调信息。
 窍门	一些容易忽视的小功能、技巧。了解这些功能或技巧能帮助解决特定问题或者节省操作时间。

2 插件使用操作说明

2.1 包含插件头文件

把 SDK 目录下 if 目录，includes 目录拷贝至工程目录下，并设置工程包含 if,includes 目录，如图 2-1 所示。

图 2-1 工程设置示意图



2.2 拷贝插件库至工程输出目录

将如图 2-2 所示插件拷贝至工程输出目录下。

图 2-2 插件库示意图



2.3 引入头文件

开发者使用插件库前，需要引入相关头文件：

```
#include "PluginVect.h"
#include "Dynamicdata_if.h"
#include "UsbProProc_if.h"
#include "lang_if.h"
```

2.4 定义插件变量

引入头文件后，使用各插件的接口前，还需要定义相关的插件变量：

```
Plugin_Man_Fun*    m_PlgMan;
USBProProc_IF*    m_USBProciF;
Lang_Plg_Interface* m_pLanIF;
```

ProcCallbackFun	g_procCbkJun;
-----------------	---------------

2.5 加载插件



说明

参考 Demo 代码 BOOL CUSBProProcDemoDlg::InitPlugin

```

/*获取插件管理器绝对路径*/
GetAbsPath(PLUGIN_VECTOR_NAME, szPluMng);
/*获取插件语言包绝对路径*/
GetAbsPath("lang.ini", szLangPath);
/*加载插件管理器*/
hModule = ::LoadLibraryA(szPluMng);
if (hModule){
/*获取插件管理器管理接口*/
    pGetPluginMan pFun = (pGetPluginMan)GetProcAddress(hModule, "GetPluginMan");
    /*创建管理器实例*/
    m_PlgMan = pFun();
    /*初始化量产插件信息*/
    Plugin_info plgInfo[] =
    {
        {USB_PROC_PLG_ID, USB_PROC_PLG_NAME},
        {LANG_PLG_ID, LANG_PLG_NAME}
    };
    int ns = sizeof(plgInfo)/ sizeof(Plugin_info);
    for(int i = 0; i < ns ; i++)
    {
        /*使用插件管理器加载量产插件*/
        m_PlgMan->LoadPlugin(plgInfo[i]);
    }
}
/*从插件管理器中获取量产插件接口*/
m_USBProIF = (USBProProc_IF*)m_PlgMan->QueryInterface(USB_PROC_PLG_ID);
m_pLanIF = (Lang_Plug_Interface*)m_PlgMan->QueryInterface(LANG_PLG_ID);
/*加载插件语言包*/
m_pLanIF->LoadStringEx(szLangPath);
/*设置量产各种事件回调*/
g_procCbkJun.OnBurnProcess          = OnBurnProcess;
g_procCbkJun.OnBurnError           = OnBurnError;
g_procCbkJun.OnBurnWarning         = OnBurnError;
g_procCbkJun.OnBurnOver            = OnBurnOver;
g_procCbkJun.OnUpdateTipInfo       = OnUpdateTipInfo;
g_procCbkJun.OnLoadDyamic          = NULL;
g_procCbkJun.OnMessageBox          = OnMessageBox;
g_procCbkJun.OnUpdateDevInfo       = OnUpdateDeviceInfo;
g_procCbkJun.ConfigDynamic         = NULL;
g_procCbkJun.EnableStart           = EnableStart;
g_procCbkJun.LightOnOff            = LightOnOff;

```

```
g_procCbkJun.OnUpdateResultInfo = OnUpdateResultInfo;  
g_procCbkJun.OnFesIn          = OnFesIn;  
/*设备量产插件回调*/  
m_USBProclF->InitIizeProc(this->m_hWnd, &g_procCbkJun);
```

2.6 捕获设备插拔消息

1. 重载主窗口类虚函数:

```
virtual LRESULT WindowProc(UINT message, WPARAM wParam, LPARAM lParam);
```

2. 在函数中加入插件对消息的捕获:

```
if(m_USBProclF)  
    m_USBProclF->PreLiveMessage(this->m_hWnd, message, wParam, lParam);  
return CDialog::WindowProc(message, wParam, lParam);
```



3 接口说明

3.1 接口结构体定义

```

typedef struct _Tga_USBProProc_IF
{
    pInitlizeProc      InitlizeProc;
    pSetImageFile     SetImageFile;
    pSetKeyFile       SetKeyFile;
    pPreLiveMessage   PreLiveMessage;
    pStartWork        StartWork;
    pStopWork         StopWork;
    pGetDynamicIF     GetDynamicIF;
    pGetDynamicHandle GetDynamicHandle;
    pLoadCfg          LoadCfg;
    pGetVersion       GetVersion;
    pSetReboot        SetReboot;
    pSetBurnPrivate   SetBurnPrivate;
    pRelease          Release;//新增 2019.12
}USBProProc_IF;
    
```

3.2 各接口作用

3.2.1 pInitlizeProc

表 3-1 pInitlizeProc 接口说明

接口原型	typedef LRESULT (*pInitlizeProc)(HWND hWnd, PProcCallbackFun pCbKFunList);
功能	初始化插件
参数	hWnd 消息窗口 pCbKFunList 回调函数结构体
返回值	0 SUCCESS,非 0 FAIL

3.2.2 pSetImageFile

表 3-2 pSetImageFile 接口说明

接口原型	ypedef LRESULT (*pSetImageFile)(LPCTSTR lpFileName);
功能	设置固件路径
参数	lpFileName 固件路径
返回值	0 SUCCESS,非 0 FAIL

3.2.3 pSetKeyFile

表 3-3 pSetKeyFile 接口说明

接口原型	typedef LRESULT (*pSetKeyFile)(LPCTSTR lpFileName);
------	---

功能	已失效
----	-----

3.2.4 pPreLiveMessage

表 3-4 pPreLiveMessage 接口说明

接口原型	typedef LRESULT (*pPreLiveMessage)(HWND hWnd, UINT message, WPARAM wParam, LPARAM lParam);
功能	捕获设备消息
参数	hWnd 消息窗口 message 消息 ID wParam, lParam 消息参数
返回值	0 SUCCESS, 非 0 FAIL

3.2.5 pStopWork

表 3-5 pStopWork 接口说明

接口原型	typedef LRESULT (*pStopWork)();
功能	暂停插件
参数	无
返回值	0 SUCCESS, 非 0 FAIL

3.2.6 pStartWork

表 3-6 pStartWork 接口说明

接口原型	typedef LRESULT (*pStartWork)();
功能	启动插件
参数	无
返回值	0 SUCCESS, 非 0 FAIL

3.2.7 pGetDynamicIF

表 3-7 pGetDynamicIF 接口说明

接口原型	typedef DynamicData_Fun* (*pGetDynamicIF)();
功能	已失效

3.2.8 pGetDynamicHandle

表 3-8 pGetDynamicHandle 接口说明

接口原型	typedef HANDLE (*pGetDynamicHandle)();
功能	已失效

3.2.9 pLoadCfg

表 3-9 pLoadCfg 接口说明

接口原型	typedef LRESULT (*pLoadCfg)(void* pCfgsystem, void* pCfgconfig);
功能	已失效

3.2.10 pGetVersion

表 3-10 pGetVersion 接口说明

接口原型	typedef LRESULT (*pGetVersion)(PHOENIX_VER_t* pVersion);
功能	获取插件版本信息
参数	pVersion 存放版本信息结构体地址
返回值	0 SUCCESS, 非 0 FAIL

3.2.11 pSetReboot

表 3-11 pSetReboot 接口说明

接口原型	typedef void (*pSetReboot)(int rebootFlag);
功能	设置固件烧录完毕后是否重启设备
参数	rebootFlag 是否重启
返回值	无

3.2.12 pSetBurnPrivate

表 3-12 pSetBurnPrivate 接口说明

接口原型	typedef void (*pSetBurnPrivate)(int burnPrivateFlag);
功能	已失效

3.2.13 pRelease

表 3-13 pRelease 接口说明

接口原型	typedef void (*pRelease)();
功能	调用 UnRegisterDeviceMessage, 去除 usb 消息注册

4 回调接口说明

定义消息回调函数为全局函数，全局函数类型如下：

```
// 烧录过程回调函数
LRESULT OnBurnProcess (HWND hWnd, WPARAM wParam, LPARAM lParam )
// wParam 进度值 0~100
// lParam 返回量产设备 id,类型为 int 型
// 烧录错误回调函数
LRESULT OnBurnError(HWND hWnd, WPARAM wParam, LPARAM lParam )
// wParam 返回量产操作错误提示, 类型为 int
// lParam 返回量产设备 id,类型为 int 型

// 烧录警告回调函数
LRESULT OnBurnWarning(HWND hWnd, WPARAM wParam, LPARAM lParam )
// wParam 返回量产操作警告提示, 类型为 int
// lParam 返回量产设备 id,类型为 int 型

// 烧录完毕回调函数
LRESULT OnBurnOver(HWND hWnd, WPARAM wParam, LPARAM lParam )
// wParam mp_info_t 类型指针, 包含 chipid 等信息
// lParam 返回量产设备 id,类型为 int 型

// 烧录 更新提示信息回调函数
LRESULT OnUpdateTipInfo(HWND hWnd, WPARAM wParam, LPARAM lParam )
// wParam 返回量产操作警告提示, 类型为 int
// lParam 返回量产设备 id,类型为 int 型

// 弹窗提示信息回调函数, 暂未使用
LRESULT OnMessageBox(HWND hWnd, WPARAM wParam, LPARAM lParam)
// wParam 返回量产操作警告号, 类型为 int,
// lParam 返回量产设备 id,类型为 int 型

// 更新指定 ID 信息时回调函数
LRESULT OnUpdateDeviceInfo(HWND hWnd, WPARAM wParam, LPARAM lParam)
// wParam 不为 0 时重置, 为 0 时无意义, 类型为 int
// lParam 返回量产设备 id,类型为 int 型

//初始化完成时回调
LRESULT EnableStart(HWND hWnd, BOOL bEnable)
// hWnd, 消息窗口
// bEnable 返回是否可开始监听。
```

```
//提示可插拔设备时回调  
LRESULT LightOnOff(HWND hWnd, BOOL bOnOff)  
//hWnd,  消息窗口  
//bEnable  返回是否可插拔  
  
//设备开始烧写前回调  
LRESULT OnFesIn(HWND hWnd, WPARAM wParam, LPARAM lParam)  
//hWnd 消息窗口  
//wParam 返回量产设备 sn 号 const char*  
//lParam 返回量产设备 id,类型为 int 型  
  
//已失效,直接返回 0 即可  
LRESULT OnUpdateResultInfo(HWND hWnd, WPARAM wParam, LPARAM lParam)
```



5 信息号说明



说明

参考 SDK 头文件 includes/tool_if_ex.h

```

#define TOOL_STEP                0x10    //UI 进度
#define TOOL_ERROR               0x20    //错误
#define TOOL_WARN                0x30    //警告
#define TOOL_OVER                0x40    //完成
#define TOOL_TIP                 0x50    /*[8/4/2011 Sam.Wu]Added */

//#define TOOL_CANCEL            0x50    //取消升级操作

enum _TOOL_TIP_E{
    TOOLS_TIP_HW_SCAN    = TOOL_TIP + 1, //phase: storage media scan
    TOOLS_TIP_DOWN_FED,           //phase: down and run fed
    TOOLS_TIP_DOWN_MBR,          //phase: down MBR
    TOOLS_TIP_DOWN_UPDATE_BOOT,  //phase: down update boot0/boot1
    TOOLS_TIP_NEAR_FINISH,       //Phase: Finish to down and restore system
    TOOLS_TIP_DOWN_PART_0,       //phase: down download0
    TOOLS_TIP_DOWN_PART_1,       //phase: down download1
    TOOLS_TIP_DOWN_PART_2,       //phase: down download2
    TOOLS_TIP_DOWN_PART_3,       //phase: down download3
    TOOLS_TIP_DOWN_PART_4,       //phase: down download4
    TOOLS_TIP_DOWN_PART_5,       //phase: down download5
    TOOLS_TIP_DOWN_PART_6,       //phase: down download0
    TOOLS_TIP_DOWN_PART_7,       //phase: down download1
    TOOLS_TIP_DOWN_PART_8,       //phase: down download2
    TOOLS_TIP_DOWN_PART_9,       //phase: down download3
    TOOLS_TIP_DOWN_PART_A,       //phase: down download4
    TOOLS_TIP_DOWN_PART_B,       //phase: down download5
};

#define TOOLS_TIP_STR_HW_SCAN    _T("Scanning the storage media.")
#define TOOLS_TIP_STR_DOWN_FED   _T("Downing the FED.")
#define TOOLS_TIP_STR_DOWN_MBR   _T("Downing the MBR.")
#define TOOLS_TIP_STR_DOWN_UP_BOOT _T("Downing update boot.")
#define TOOLS_TIP_STR_NEAR_FINISH _T("Finish to down and restore system")
#define TOOLS_TIP_STR_DOWN_PART0 _T("Downing the PART0.")
#define TOOLS_TIP_STR_DOWN_PART1 _T("Downing the PART1.")
#define TOOLS_TIP_STR_DOWN_PART2 _T("Downing the PART2.")
#define TOOLS_TIP_STR_DOWN_PART3 _T("Downing the PART3.")
#define TOOLS_TIP_STR_DOWN_PART4 _T("Downing the PART4.")
#define TOOLS_TIP_STR_DOWN_PART5 _T("Downing the PART5.")
#define TOOLS_TIP_STR_DOWN_PART6 _T("Downing the PART6.")
#define TOOLS_TIP_STR_DOWN_PART7 _T("Downing the PART7.")
#define TOOLS_TIP_STR_DOWN_PART8 _T("Downing the PART8.")
#define TOOLS_TIP_STR_DOWN_PART9 _T("Downing the PART9.")
#define TOOLS_TIP_STR_DOWN_PARTA _T("Downing the PARTA.")
    
```

```

#define TOOLS_TIP_STR_DOWN_PARTB    _T("Downing the PARTB.")

//-----烧写失败错误号-----
//PC 工具自检  //100~119
#define FES_ERR_NUM_100    "检测失败"
#define FES_ERR_NUM_101    "打开固件包失败"
#define FES_ERR_NUM_102    "关闭固件包失败"
#define FES_ERR_NUM_103    "打开 USB 设备失败"
#define FES_ERR_NUM_104    "关闭 USB 设备失败"
#define FES_ERR_NUM_105    "初始化 MBR 失败"

//硬件扫描  //120~139
#define FES_ERR_NUM_120    "硬件失败"    /* nand */

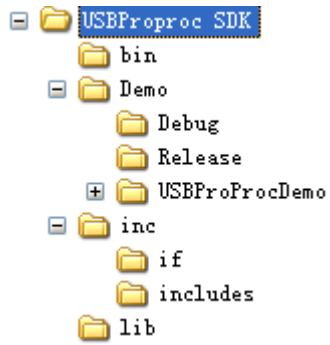
//固件烧写  //160~189
#define FES_ERR_NUM_160    "烧写失败"    /* 烧写 boot0 失败 */
#define FES_ERR_NUM_161    "烧写失败"    /* 烧写 boot1 失败 */
#define FES_ERR_NUM_162    "烧写失败"    /* 烧写 fed_nand 失败 */
#define FES_ERR_NUM_163    "烧写失败"    /* 烧写 rootfs 失败 */
#define FES_ERR_NUM_164    "烧写失败"    /* 保留 */
#define FES_ERR_NUM_165    "烧写失败"    /* 创建 SPM 失败 */
#define FES_ERR_NUM_166    "烧写失败"    /* 烧写 SPM 失败 */
#define FES_ERR_NUM_167    "烧写失败"    /* 烧写 MBR 失败 */
#define FES_ERR_NUM_168    "烧写失败"    /* 烧写量产结束标识符失败 */

//烧写后处理  //190~199
#define FES_ERR_NUM_190    "系统复位失败"    /* 小机重启失败 */
    
```

6 目录说明

SDK 包的目录如图 6-1 所示。

图 6-1 SDK 包目录示意图



lib 目录为插件相关可执行文件。

Demo 目录为插件的使用 demo 源代码参考目录。

bin 目录为 demo 的可执行示例。

inc/if 和 inc/includes 两个目录为接口声明文件。

著作权声明

版权所有©2020 珠海全志科技股份有限公司。保留一切权利。

本档及内容受著作权法保护，其著作权由珠海全志科技股份有限公司（“全志”）拥有并保留一切权利。

本档是全志的原创作品和版权财产，未经全志书面许可，任何单位和个人不得擅自摘抄、复制、修改、发表或传播本档内容的部分或全部，且不得以任何形式传播。

商标声明

、、、（不完全列举）均为珠海全志科技股份有限公司的商标或者注册商标。在本档描述的产品中出现的其它商标，产品名称，和服务名称，均由其各自所有人拥有。

免责声明

您购买的产品、服务或特性应受您与珠海全志科技股份有限公司（“全志”）之间签署的商业合同和条款的约束。本档中描述的全部或部分产品、服务或特性可能不在您所购买或使用的范围内。使用前请认真阅读合同条款和相关说明，并严格遵循本档的使用说明。您将自行承担任何不当使用行为（包括但不限于如超压，超频，超温使用）造成的不利后果，全志概不负责。

本档作为使用指导仅供参考。由于产品版本升级或其他原因，本档内容有可能修改，如有变更，恕不另行通知。全志尽全力在本档中提供准确的信息，但并不确保内容完全没有错误，因使用本档而发生损害（包括但不限于间接的、偶然的、特殊的损失）或发生侵犯第三方权利事件，全志概不负责。本档中的所有陈述、信息和建议并不构成任何明示或暗示的保证或承诺。

本档未以明示或暗示或其他方式授予全志的任何专利或知识产权。在您实施方案或使用产品的过程中，可能需要获得第三方的权利许可。请您自行向第三方权利人获取相关的许可。全志不承担也不代为支付任何关于获取第三方许可的许可费或版税（专利税）。全志不对您所使用的第三方许可技术做出任何保证、赔偿或承担其他义务。