



Customized_FlashOperationLib


文档版本	V3.0
发布日期	2023-02-03

版权所有 © 紫光展锐（上海）科技有限公司。保留一切权利。

本文件所含数据和信息都属于紫光展锐（上海）科技有限公司（以下简称紫光展锐）所有的机密信息，紫光展锐保留所有相关权利。本文件仅为信息参考之目的提供，不包含任何明示或默示的知识产权许可，也不表示有任何明示或默示的保证，包括但不限于满足任何特殊目的、不侵权或性能。当您接受这份文件时，即表示您同意本文件中内容和信息属于紫光展锐机密信息，且同意在未获得紫光展锐书面同意前，不使用或复制本文件的整体或部分，也不向任何其他方披露本文件内容。紫光展锐有权在未经事先通知的情况下，在任何时候对本文件做任何修改。紫光展锐对本文件所含数据和信息不做任何保证，在任何情况下，紫光展锐均不负任何与本文件相关的直接或间接的、任何伤害或损失。

请参照交付物中说明文档对紫光展锐交付物进行使用，任何人对紫光展锐交付物的修改、定制化或违反说明文档的指引对紫光展锐交付物进行使用造成的任何损失由其自行承担。紫光展锐交付物中的性能指标、测试结果和参数等，均为在紫光展锐内部研发和测试系统中获得的，仅供参考，若任何人需要对交付物进行商用或量产，需要结合自身的软硬件测试环境进行全面的测试和调试。

商标声明

紫光展锐、**UNISOC**、、展讯、Spreadtrum、SPRD、锐迪科、RDA 及其他紫光展锐的商标均为紫光展锐（上海）科技有限公司及/或其子公司、关联公司所有。

Bluetooth®文字商标和徽标为蓝牙技术联盟的注册商标，紫光展锐（上海）科技有限公司对此类商标的任何使用均已获得许可。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

免责声明

本文档可能包含第三方内容，包括但不限于第三方信息、软件、组件、数据等。紫光展锐不控制且不对第三方内容承担任何责任，包括但不限于准确性、兼容性、可靠性、可用性、合法性、适当性、性能、不侵权、更新状态等，除非本文档另有明确说明。在本文档中提及或引用任何第三方内容不代表紫光展锐对第三方内容的认可、承诺或保证。

用户有义务结合自身情况，检查上述第三方内容的可用性。若需要第三方许可，应通过合法途径获取第三方许可，除非本文档另有明确说明。

紫光展锐（上海）科技有限公司



前 言

概述

FlashOperationLib 是 Download 工具下提供的客制化模块，可以读取、写入或擦除 DUT 中的数据，适用于所有 PC 上位机程序开发人员和对外合作开发人员。

读者对象

本文档主要适用于有相关需求的开发人员，必须具备以下经验和技能：



- 熟悉 C++ 编程语言。
- 了解需要操作的数据所在分区和偏移。

缩略语

缩略语	英文全称	中文解释
DUT	Device under test	待测设备

符号约定

在本文中可能出现下列符号，每种符号的说明如下。

符号	说明
 说明	用于突出重要或关键信息、补充信息和小窍门等。 “说明”不是安全警示信息，不涉及人身、设备及环境伤害。
 注意	用于突出容易出错的操作。 “注意”不是安全警示信息，不涉及人身、设备及环境伤害。

变更信息

文档版本	发布日期	作者	修改说明
V3.0	2023-02-03	UNISOC	更新文档，适配 R27 工具
V2.1	2022-02-17	UNISOC	优化代码结构
V2.0	2022-01-05	UNISOC	增加全擦和写分区功能
V1.0	2021-11-18	UNISOC	草稿

关键字

ReadFlash、EraseFlash、WriteFlash

目 录

1 FlashOperationLib	1
1.1 功能概述	1
1.2 目录结构	1
1.3 接口说明	1
1.3.1 Release	1
1.3.2 GetErrorMsg	2
1.3.3 SetProperty	2
1.3.4 GetProperty	3
1.3.5 EnableOperation	3
1.3.6 CheckReadBuffer	4
1.3.7 PrepareReadFlashAfterFDL2	4
1.3.8 PrepareEraseFlashAfterFDL2	5
1.3.9 PrepareWriteFlashAfterFDL2	5
1.3.10 PrepareWriteFlashAfterDIEnd	6
1.3.11 CreateFlashOperationObject	7
1.3.12 SaveFileToLocal	7
1.4 案例参考	8
1.4.1 案例一：UpgradeDownload 读取 SN/SN2	8
1.4.2 案例二：先读取 Flash 信息，后修改，最后写入	10
2 参考文档	13

图目录

图 1-1 源码路径	1
------------------	---

1 FlashOperationLib

1.1 功能概述

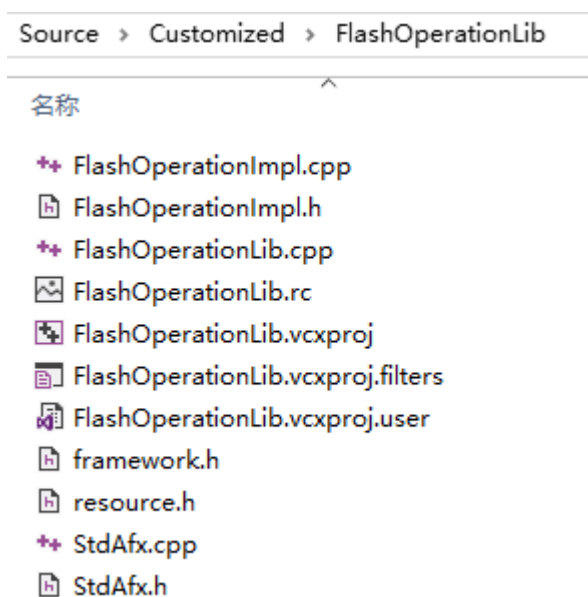
FlashOperationLib 是在下载过程中，执行读取、写入或擦除 DUT 中数据的操作。它可以操作所有分区，并且可以在一次下载过程中操作多个分区，只需指定操作分区的 FileID、size 和 offset。

1.2 目录结构

动态库路径：Bin\App\Customized\FlashOperationLib\FlashOperationLib.dll

源码路径：Source\Customized\FlashOperationLib

图1-1 源码路径



1.3 接口说明

1.3.1 Release

释放资源。

【接口定义】

```
virtual void Release( void );
```

【参数说明】

NA

【接口说明】

NA

【接口返回】

NA

1.3.2 GetErrorMsg

出现错误时，返回错误信息。

【接口定义】

```
virtual void GetErrorMsg( LPTSTR lpErrMsg, DWORD dwCount );
```

【参数说明】

参数名称	输入/输出	说明
lpErrMsg	OUT	错误信息
dwCount	OUT	错误信息字节数

【接口说明】

NA

【接口返回】

NA

1.3.3 SetProperty

设置属性。

【接口定义】

```
virtual SPRESULT SetProperty( INT nProperty, INT nFlags, LPCVOID lpValue );
```

【参数说明】

参数名称	输入/输出	说明
nProperty	IN	属性类别

nFlags	IN	属性标记
lpValue	IN	属性值，根据 nProperty 不同数据结构不同

【接口说明】

NA

【接口返回】

执行成功返回 SP_OK，否则返回错误代码。

1.3.4 GetProperty

获取属性。预留接口。

【接口定义】

```
virtual SPRESULT GetProperty( INT nProperty, INT nFlags, LPVOID lpValue );
```

【参数说明】

参数名称	输入/输出	说明
nProperty	IN	属性类别
nFlags	IN	属性标记
lpValue	OUT	属性值，根据 nProperty 不同数据结构不同

【接口说明】

NA

【接口返回】

执行成功返回 SP_OK，否则返回错误代码。

1.3.5 EnableOperation

功能开关。

【接口定义】

```
virtual SPRESULT EnableOperation( DL_TYPE_ENUM eType, LPFLASH_OPERATION_SWITCH lpOperation );
```

【参数说明】

参数名称	输入/输出	说明
eType	IN	工具类型，DL_TYPE_ENUM
lpOperation	IN	开关设置

【接口说明】

开启功能，将 lpOperation 指向值设为 1。

【接口返回】

执行成功返回 SP_OK。

1.3.6 CheckReadBuffer

读取数据校验。

【接口定义】

```
virtual SPRESULT CheckReadBuffer( LPCSTR lpReadBuffer, LPFLASH_READ_FILE lpReadFile );
```

【参数说明】

参数名称	输入/输出	说明
lpReadBuffer	OUT	输出读取的分区数据
lpReadFile	OUT	输出读取的分区信息

【接口说明】

可在该分区实现修改从 UE 侧读取的数据，再写入 UE 侧。

【接口返回】

执行成功返回 SP_OK，否则返回错误代码。

1.3.7 PrepareReadFlashAfterFDL2

FDL2 之后读取数据。

【接口定义】

```
virtual SPRESULT PrepareReadFlashAfterFDL2( LPFLASH_READ_FILE lpReadFile, INT& nCount );
```

【参数说明】

参数名称	输入/输出	说明
lpReadFile	IN	读取的分区信息
nCount	IN	读取的分区数量

【接口说明】

ReadFlash 操作在 FDL2 之后，读取的数据是 DUT 中上次的数据，而不是本次下载的数据。

【接口返回】

执行成功返回 SP_OK。

1.3.8 PrepareEraseFlashAfterFDL2

FDL2 之后擦除数据。

【接口定义】

```
virtual SPRESULT PrepareEraseFlashAfterFDL2( LPFLASH_ERASE_FILE lpEraseFile, INT& nCount );
```

【参数说明】

参数名称	输入/输出	说明
lpEraseFile	IN	擦除的分区信息
nCount	IN	擦除的分区数量

【接口说明】

NA

【接口返回】

执行成功返回 SP_OK。

1.3.9 PrepareWriteFlashAfterFDL2

FDL2 之后写入数据。

【接口定义】

```
virtual SPRESULT PrepareWriteFlashAfterFDL2( LPFLASH_WRITE_FILE lpWriteFile, INT& nCount );
```

【参数说明】

参数名称	输入/输出	说明
lpWriteFile	IN	需要写入的分区信息
nCount	IN	写入的分区数量

【接口说明】

NA

【接口返回】

执行成功返回 SP_OK。

1.3.10 PrepareWriteFlashAfterDIEnd

下载结束前写入数据。

【接口定义】

```
virtual SPRESULT PrepareWriteFlashAfterDIEnd( LPFLASH_WRITE_FILE lpWriteFile, INT& nCount ) = 0;
```

【参数说明】

参数名称	输入/输出	说明
lpWriteFile	IN	需要写入的分区和信息
nCount	IN	写入的分区数量

【接口说明】

写入之后，可以回读该分区，检查是否写入。

【接口返回】

执行成功返回 SP_OK。

1.3.11 CreateFlashOperationObject

创建类对象。

【接口定义】

```
SP_EXPORT IFlashOperation* SP_API CreateFlashOperationObject( LPVOID pLogUtil );
```

【参数说明】

参数名称	输入/输出	说明
pLogUtil	IN	类对象

【接口说明】

NA

【接口返回】

执行成功返回类对象。

1.3.12 SaveFileToLocal

保存文件到本地。

【接口定义】

```
BOOL SaveFileToLocal( LPCSTR pszFileName, LPBYTE pBuf, DWORD dwSize );
```

【参数说明】

参数名称	输入/输出	说明
pszFileName	IN	文件路径
pBuf	IN	待写入的分区数据
dwSize	IN	分区数据大小

【接口说明】

NA

【接口返回】

写入成功返回 TRUE。

1.4 案例参考

1.4.1 案例一：UpgradeDownload 读取 SN/SN2

1.4.1.1 需求说明

将 Download 工具嵌套进 MES 系统中，使用 UpgradeDownload 工具，通过读取 SN/SN2，判断是否需要进行升级。

1.4.1.2 操作步骤

1. 开启功能。

```
SPRESULT CFlashOperationImpl::EnableOperation( DL_TYPE_ENUM eType,  
LPFLASH_OPERATION_SWITCH lpOperation )
```

```
{  
    CTrLog log( this, _T( "EnableOperation" ) );  
    if( UPGRADE_TYPE == eType )  
    {  
        lpOperation->u8ReadFlashAfterFDL2 = 1;  
    }  
    lpOperation->u8EraseFlashAfterFDL2 = 0;  
    lpOperation->u8WriteFlashAfterFDL2 = 0;  
    lpOperation->u8WriteFlashAfterDIEnd = 0;  
    return SP_OK;  
}
```

2. 使用 PrepareReadFlashAfterFDL2 功能读取 miscdata 分区。

```
SPRESULT CFlashOperationImpl::PrepareReadFlashAfterFDL2( LPFLASH_READ_FILE lpReadFile, INT&  
nCount )
```

```
{  
    CTrLog log( this, _T( "ReadFlashAfterFDL2" ) );  
    if( 2 > nCount )  
    {  
        LogFmtStr( SPLOGLV_ERROR, _T( "The operation cannot be larger than %d" ), nCount );  
        return SP_E_FAIL;  
    }  
    nCount = 0;  
    wcsncpy_s( lpReadFile[nCount].szFileId, L"miscdata" );  
    lpReadFile[nCount].dwFileOffset = 0x00;
```

```
lpReadFile[nCount].dwFileSize = 0x50;  
nCount++;  
  
return SP_OK;  
}
```

3. 使用回调函数接口获取 SN/SN2。

```
BOOL CCallbackHelper::__cbHandleStringInfo( LPCCALLBACKDATA_STRING_INFO lpcbData )  
{  
    .....  
    case CBK_STRING_SN1:  
    {  
        m_strSN1.clear();  
        m_strSN1 = lpcbData->lpInfo;  
    }  
    break;  
    case CBK_STRING_SN2:  
    {  
        m_strSN2.clear();  
        m_strSN2 = lpcbData->lpInfo;  
    }  
    break;  
    .....  
}  
  
return TRUE;  
}
```

4. 使用回调函数接口获取下载结果。

```
BOOL CCallbackHelper::__cbHandleDIEnd( LPCCALLBACKDATA_DL_END lpcbData )  
{  
    if ( NULL == lpcbData )  
    {  
        ASSERT( 0 );  
        return FALSE;  
    }  
}
```

```

m_tBmDIEnd.lpErrMsg = lpcbData->lpErrMsg;
m_tBmDIEnd.dwErrCode = lpcbData->u32ErrCode;
m_tBmDIEnd.u32Stage = lpcbData->u32Stage;
m_tBmDIEnd.bDLEnd = TRUE;

PostMessage( g_hWndView, BM_END, lpcbData->u32Port, ( LPARAM )&m_tBmDIEnd );
PostMessage( g_hWndFrame, WM_STOP_ONE_PORT, lpcbData->u32Port, NULL );
return TRUE;
}

```

说明

Callback 接口可参考：UNISOC DLFramework 集成指导说明.pdf。

1.4.2 案例二：先读取 Flash 信息，后修改，最后写入

1.4.2.1 需求说明

在下载 FDL2 后，读取 miscdata 分区信息，然后修改该部分信息，最后写入到 miscdata 分区。

1.4.2.2 操作步骤

1. 开启功能。

```

SPRESULT CFlashOperationImpl::EnableOperation( DL_TYPE_ENUM eType,
LPFLASH_OPERATION_SWITCH lpOperation )
{
    CTrLog log( this, _T( "EnableOperation" ) );
    if( RESEARCH_TYPE == eType )
    {
        lpOperation->u8ReadFlashAfterFDL2 = 1;
    }
    lpOperation->u8EraseFlashAfterFDL2 = 0;
    lpOperation->u8WriteFlashAfterFDL2 = 1;
    lpOperation->u8WriteFlashAfterDIEnd = 0;
    return SP_OK;
}

```

2. 设置读分区。


```
SPRESULT CFlashOperationImpl::PrepareReadFlashAfterFDL2( LPFLASH_READ_FILE lpReadFile, INT&
nCount )
{
    CTrLog log( this, _T( "ReadFlashAfterFDL2" ) );
    if ( 2 > nCount )
    {
        LogFmtStr( SPLOGLV_ERROR, _T( "The operation cannot be larger than %d" ), nCount );
        return SP_E_FAIL;
    }
    nCount = 0;
    wcsncpy_s( lpReadFile[nCount].szFileId, L"miscdata" );
    lpReadFile[nCount].dwFileOffset = 0x00;
    lpReadFile[nCount].dwFileSize = 0x1C;
    nCount++;

    return SP_OK;
}
```

3. 设置写分区。

```
SPRESULT CFlashOperationImpl::PrepareWriteFlashAfterFDL2( LPFLASH_WRITE_FILE lpWriteFile, INT&
nCount )
{
    CTrLog log( this, _T( "WriteFlashAfterFDL2" ) );
    INT nOperation = 1;
    if ( nOperation > nCount )
    {
        LogFmtStr( SPLOGLV_ERROR, _T( "The operation cannot be larger than %d" ), nCount );
        return SP_E_FAIL;
    }
    m_arrWriteFdl2.Clear();

    nCount = 0;
    wcsncpy_s( lpWriteFile[nCount].szFileId, L"miscdata" );
    lpWriteFile[nCount].dwFileOffset = 0x00;
    lpWriteFile[nCount].dwFileSize = 0x1C;

    memcpy( lpWriteFile[nCount].u8FileValue, "99999999999999999999999999999999",
lpWriteFile[nCount].dwFileSize );    //该值可在CheckReadBuffer中被替换
}
```

```
nCount++;  
  
return SP_OK;  
  
}
```

4. 修改从 UE 侧读取的数据，再写入 UE 侧。

```
SPRESULT CFlashOperationImpl::CheckReadBuffer( LPCSTR lpReadBuffer, LPFLASH_READ_FILE  
lpReadFile )  
{  
    .....  
    // 实现修改从UE侧读取的数据，再写入UE侧;  
    // 注意：根据判断条件确认修改的位置  
    if (dwToatlSize == (dwFileSize + dwFileOffset))  
    {  
        LPFLASH_WRITE_FILE_VALUE lpValue = m_arrWriteFd12.Find( L"miscdata" );  
        if ( NULL != lpValue && NULL != lpReadBuffer )  
        {  
            memcpy_s(lpValue->pFileValue, lpValue->dwFileValueSize, "55555555555555555555555555555555",  
(*lpValue->pFileSize));  
            memcpy_s( lpValue->pFileValue, lpValue->dwFileValueSize, lpReadBuffer,  
(*lpValue->pFileSize ) );           //此时写入值为lpReadBuffer  
        }  
    }  
    .....  
    return SP_OK;  
}
```

2

参考文档

1. 《UNISOC DLFramework 集成指导说明.pdf》