

Download_ReadFlash_动态库使用说明

Document Number:		Document Version:	2.1
Owner:	Danyang.zheng	Date:	2022-02-17
Document Type:			
NOTE:	<p>ALL MATERIALS INCLUDED HEREIN ARE COPYRIGHTED AND CONFIDENTIAL UNLESS OTHERWISE INDICATED. The information is intended only for the person or entity to which it is addressed and may contain confidential and/or privileged material. Any review, retransmission, dissemination, or other use of or taking of any action in reliance upon this information by persons or entities other than the intended recipient is prohibited.</p> <p>This document is subject to change without notice. Please verify that your company has the most recent specification.</p> <p>Copyright © 2019 UNISOC Communications Inc.</p>		

修订记录

版本	日期	作者	说明
V1.0	2021-11-18	Danyang.zheng	草稿
V2.0	2022-01-05	Danyang.zheng	增加全擦和写分区功能
V2.1	2022-02-17	Danyang.zheng	优化代码结构

目录

Download_ReadFlash_动态库使用说明.....	1
修订记录	2
目录.....	3
1. 目的.....	4
1.1. 缩略词说明.....	4
1.2. 参考文档.....	4
2. 环境配置	5
2.1. 生成 ReadFlash.dll 动态库	5
2.2. 添加 ReadFlash.dll	5
3. 动态库接口说明.....	6
3.1. 功能开关.....	6
3.2. 设置读取分区参数	7
3.3. 读取数据校验	7
3.4. 设置写分区参数	8
3.5. 设置擦除分区参数	9

1. 目的

本文介绍 Download 工具客制化读取分区功能 ReadFlash.dll 动态库的方法，适用于所有 PC 上位机程序开发人员和对外合作开发人员。

1.1. 缩略词说明

名称	全称	定义
OPR	Operation	功能

1.2. 参考文档

2. 环境配置

2.1. 生成 ReadFlash.dll 动态库

根据第 3 章接口说明内容，填充和配置接口实现，工具将根据接口的具体实现完成读取分区功能流程。

2.2. 添加 ReadFlash.dll

将编译出的 ReadFlash 动态库放到工具包/Bin/Customized/ReadFlash/ReadFlash.dll 路径上。

- 结构说明：

- Bin
 - |——Download.exe
 - |——Customized
 - |——ReadFlash
 - |——ReadFlash.dll
 - |——其余客制化模块

3. 动态库接口说明

3.1. 功能开关

【语法】

```
READFLASH_API DownloadResult EnableOperation(OperationSwitch* pEnable,  
Download_Verson_E Version);
```

【说明】

设置指定功能是否打开

【参数】

参数	IN/OUT	说明
OperationSwitch* pEnable	OUT	返回指定功能是否打开
Download_Verson_E Version	IN	当前工具版本

```
struct OperationSwitch  
{
```

```
    int ReadByFileID_FDL2 : 1;  
    int ReadSNByFileID_FDL2 : 1;  
    int WriteByFileID_END : 1;  
    int EraseByFileID_FDL2 : 1;  
    int Reserved0 : 12;  
    int Reserved1[3];
```

```
};
```

0 : 关闭功能

1 : 打开功能

【返回值】

```
enum DownloadResult_E
```

```
{
```

```
    DownloadResult_OK = 0x00,          /* 0x00 Setting success*/  
    DownloadResult_FALSE = 0x01,       /* 0x01 Setting error*/  
    DownloadResult_FAILED = 0x02       /* 0x02 Setting fail */  
    /*0x50 to 0xff are customer-defined error codes*/
```

```
};
```

3.2. 设置读取分区参数

【语法】

```
READFLASH_API DownloadResult ReadFileFlash(FLASH_READ_FILE_T**  
pReadFile, int* pCount);
```

【说明】

用户设置读取分区信息

【参数】

参数	IN/OUT	说明
FLASH_READ_FILE_T** pReadFile	IN	设置需要读取的分区信息
int* pCount	IN	设置需要读取的分区个数

```
typedef struct _FLASH_READ_FILE_T  
{  
    DWORD      FileSize = 0;           // File size  
    DWORD      FileOffset = 0;         // File offset  
    wchar_t    FileId[MAX_REP_ID_LEN] = { 0 }; // File ID  
} FLASH_READ_FILE_T, * PFLASH_READ_FILE_T;
```

【返回值】

```
enum DownloadResult_E  
{  
    DownloadResult_OK = 0x00,           /* 0x00 Setting success*/  
    DownloadResult_FALSE = 0x01,        /* 0x01 Setting error*/  
    DownloadResult_FAILED = 0x02        /* 0x02 Setting fail */  
    /*0x50 to 0xff are customer-defined error codes*/  
};
```

3.3. 读取数据校验

【语法】

```
READFLASH_API DownloadResult CheckReadBuffer(char* lpReadBuffer,  
FLASH_READ_FILE_T* pReadFile, int PortID);
```

【说明】

分批次读取分区数据，客户可自定义校验操作

【参数】

参数	IN/OUT	说明
char* lpReadBuffer	OUT	输出读取的分区数据
FLASH_READ_FILE_T* pReadFile	OUT	输出读取的分区信息
int PortID	OUT	输出当前端口 ID

【返回值】

```
enum DownloadResult_E
{
    DownloadResult_OK = 0x00,          /* 0x00 Setting success*/
    DownloadResult_FALSE = 0x01,       /* 0x01 Setting error*/
    DownloadResult_FAILED = 0x02       /* 0x02 Setting fail */
    /*0x50 to 0xff are customer-defined error codes*/
};
```

3.4. 设置写分区参数

【语法】

```
READFLASH_API DownloadResult WriteFileFlash(FLASH_WRITE_FILE_T**
pWriteFile, int* pCount)
```

【说明】

用户设置写分区信息

【参数】

参数	IN/OUT	说明
FLASH_WRITE_FILE_T** pWriteFile	IN	设置需要写的分区信息
int* pCount	IN	设置需要写的分区个数

```
typedef struct _FLASH_WRITE_FILE_T
{
    DWORD      FileSize = 0;          // File size
    DWORD      FileOffset = 0;        // File offset
    wchar_t    FileId[16] = { 0 };    // File ID
    char       FileValue[MAX_LEN] = { 0 }; // File value
} FLASH_WRITE_FILE_T, * PFLASH_WRITE_FILE_T;
```


【返回值】

```
enum DownloadResult_E
{
    DownloadResult_OK = 0x00,          /* 0x00 Setting success*/
    DownloadResult_FALSE = 0x01,       /* 0x01 Setting error*/
    DownloadResult_FAILED = 0x02       /* 0x02 Setting fail */
    /*0x50 to 0xff are customer-defined error codes*/
};
```

3.5. 设置擦除分区参数

【语法】

```
READFLASH_API DownloadResult EraseFileFlash(FLASH_ERASE_FILE_T**
pEraseFile, int* pCount);
```

【说明】

用户设置擦除分区信息

【参数】

参数	IN/OUT	说明
FLASH_ERASE_FILE_T** pEraseFile	IN	设置需要擦除的分区信息
int* pCount	IN	设置需要擦除的分区个数

```
typedef struct _FLASH_ERASE_FILE_T
{
    wchar_t    FileId[MAX_LIST_ID_LEN] = { 0 };    // File ID
} FLASH_ERASE_FILE_T, * PFLASH_ERASE_FILE_T;
```

【返回值】

```
enum DownloadResult_E
{
    DownloadResult_OK = 0x00,          /* 0x00 Setting success*/
    DownloadResult_FALSE = 0x01,       /* 0x01 Setting error*/
    DownloadResult_FAILED = 0x02       /* 0x02 Setting fail */
    /*0x50 to 0xff are customer-defined error codes*/
};
```