



版本: 1.0.0 2012 年 1 月

## EXNFTL 库接口说明

## 声 明

本手册的版权归安凯技术公司所有，受相关法律法规的保护。未经安凯技术公司的事先书面许可，任何人不得复制、传播本手册的内容。

本手册所涉及的知识产权归属安凯技术公司所有（或经合作商授权许可使用），任何人不得侵犯。

本手册不对包括但不限于下列事项担保：适销性、特殊用途的适用性；实施该用途不会侵害第三方的知识产权等权利。

安凯技术公司不对由使用本手册或执行本手册内容而带来的任何损害负责。

本手册是按当前的状态提供参考，随附产品或本书内容如有更改，恕不另行通知。

## 联 系 方 式

**安凯（广州）微电子有限公司**

地址：广州科学城科学大道 182 号创新大厦 C1 区 3 楼

电话: (86)-20-3221 9000

传真: (86)-20-3221 9258

邮编: 510663

**销售热线:**

(86)-20-3221 9499

**电子邮箱:**

[sales@anyka.com](mailto:sales@anyka.com)

**主页:**

<http://www.anyka.com>

## 版本变更说明

以下表格对于本文档的版本变更做一个简要的说明。版本变更仅限于技术内容的变更，不包括版式、格式、句法等的变更。

版本	说明	完成日期
V1.0.0	正式发布	2012-1-9

## EXNFTL 库接口说明与库对应的关系

Version	The Corresponding Lib
V1.0.0	EXNFTL_V1.0.04

## 目 录

<b>1 EXNFTL库的介绍</b>	<b>4</b>
1.1 功能概述	4
1.2 适用范围	4
1.3 注意事项	4
<b>2 EXNFTL接口说明</b>	<b>5</b>
2.1 功能概述	5
2.2 数据结构	5
2.3 接口函数列表	6

Anyka Confidential For  
BOMEI Use Only

# 1 EXNFTL库的介绍

## 1.1 功能概述

EXNFTL(Extend NAND Flash Translation Layer)是为文件系统的逻辑地址与 NAND FLASH 物理地址建立映射关系。该库源代码为标准 C 语言，相对于平台有独立性，可直接运行于 PC，而且能够直接移植到其他嵌入式系统。

## 1.2 适用范围

EXNFTL 因使用全内存的缓冲方式，所以在内存使用方面比较多，其公式为  $8 * \text{TotalBlock} + \text{PagePerBlock} * 8 + 2k$ ，所以不适应于对内存要求比较高的系统。

## 1.3 注意事项

EXNFTL 库工作于多任务，在库内部会创建线程，用来进行垃圾回收，如果系统是工作于单任务系统，不能创建线程的话，将对回调函数设置为空。

## 2 EXNFTL接口说明

### 2.1 功能概述

在满足基本的内存要求后，使用对应版本的库，系统只要包括头文件 medium.h、mtdlib.h、nandflash.h。

### 2.2 数据结构

struct tag\_Medium

```
{
    T_OBJECT object;
    T_U8 type;      // Corresponding to E_MEDIUM type.
    T_U8 SecBit;    // the sector bytes, it is bits, eg:SectorPerBytes = 1024, it is 10(2^10)
    T_U16 BufFlag;  /*it is the DataBuf valid information, the current design, the page buffer
max value is 2. so the buffer valid data may be is 1,2,(1+2),if the previos page of the databuf need
be updated, it will be 1, the second page need be update, it will be 2, it will be 3 to all page */
    T_U32 PageNum;  //the page address of the Databuf data
    T_U32 capacity; // the medium room, it's unit is the sector size
    F_ReadSector read; //the read function
    F_WriteSector write; //the write function
    F_Flush flush;    //the flush function
    T_U8 *DataBuf;    // the buffer when the page size and the sector size is different,
                    //every mtd will have one,
    T_U8* msg;
};
```

typedef struct tag\_MtdCallBackFunc

```
{
    /* The folling 3 callback functions are about Ram operation. */
    F_MtdRamAlloc fRamAlloc;
    F_MtdRamRealloc fRamRealloc;
    F_MtdRamFree fRamFree;
```

```

F_MtdMemCpy fMemCpy;
F_MtdMemSet fMemSet;
F_MtdMemMove fMemMove;
F_MtdMemCmp fMemCmp;

/* The following 4 callback functions are used for Multi-Task. If the
   running-enviroment is single-task, set them into empty function. */
F_MtdOsCrtSem fCrtSem;
F_MtdOsDelSem fDelSem;
F_MtdOsObtSem fObtSem;
F_MtdOsRelSem fRelSem;

F_MtdPrintf fPrintf;
F_MtdGetChipID fGetChip;
F_MtdSysRst fMtdSysRst;
F_MtdRandSeed fMtdRandSeed;
F_MtdGetRand fMtdGetRand;
F_CreateThread fMtdCreateThread;
F_KillThead fMtdKillThread;
F_SystemSleep fSystemSleep;
}T_MTDCALLBACK, *T_PMTDCALLBACK;

```

## 2.3 接口函数列表

原 型	T_U32 NandMtd_Format(T_PNANDFLASH nand, T_U32 StartBlock, T_U32 BlkCnt)
功能概述	格式化 MTD
参数说明	T_PNANDFLASH nand: 物理 NAND FLASH 的结构体指针
	T_U32 StartBlock: 起始块号
	T_U32 BlkCnt: 块的数目
返回值说明	返回坏块数目

原 型	T_U32 NandMtd_Format(T_PNANDFLASH nand, T_U32 StartBlock, T_U32 BlkCnt)
注意事项	在第一次使用 NAND 时,要先格式化 MTD,才能使用.
调用示例	NandMtd_Format(pNand, StartBlock, BlkCnt);

原 型	T_U32 Medium_GetMTDRsvInfo(T_PNANDFLASH nand, T_U32 StartBlock, T_U32 TolBlock, T_U16 RsvBlockPerMTD, T_U16 *RsvBlock, T_U16 *RsvNum);
功能概述	当 NFTL 被格式化后,需要调用此接口,得到保留块的信息以便以后在 Nand_CreateMedium 时使用..
参数说明	T_PNANDFLASH nand: 物理 NAND FLASH 的结构体指针
	T_U32 StartBlk: 对应 NAND 上物理块号,MTD 的开始地址
	T_U32 BlkCnt :MTD 的大小
	T_U16 RsvBlockPerMTD: 每 2K 个物理块需要的保留块数目.
	T_U16 *RsvBlock:将返回保留块的信息,他可能会是一个数组,在小内存模式将是 (BlkCnt+2048)/2048 个元素,在大内存模式将是一个元素.
	T_U16 *RsvNum:返回 RsvBlock 元素的个数.
返回值说明	MTD 的实际大小,以块为单位.
注意事项	保留块信息必需在 NandMtd_Format 后调用得到,而且要保存起来,以后在每次调用 Nand_CreateMedium 时都不能改变,如果这个被改变,将会有不可预期的错误发生.
调用示例	BlkCnt = Medium_GetMTDRsvInfo(pNand, 64, BlkCnt, RsvBlkNum, RsvTbl, &RsvNum);

原 型	T_PMEDIUM Nand_CreateMedium(T_PNANDFLASH nand, T_U32 StartBlk, T_U32 BlkCnt, T_U16 *RsvBlkNum)
功能概述	创建 Medium, 也就是初始化 NFTL 信息。
参数说明	T_PNANDFLASH nand: 物理 NAND FLASH 的结构体指针
	T_U32 StartBlk: 对应 NAND 上物理块号,MTD 的开始地址
	T_U32 BlkCnt :MTD 的大小
	T_U16 *RsvBlkNum:MTD 的备用块数目,这个参数是通过调用 Medium_GetMTDRsvInfo 得到的.
返回值说明	Medium 结构体,以便可以创建分区.



原 型	T_PMEDIUM Nand_CreateMedium(T_PNANDFLASH nand, T_U32 StartBlk, T_U32 BlkCnt, T_U16 *RsvBlkNum)
注意事项	BlkCnt 不能小于 RsvBlkNum 块数目,否则创建会失败,而且系统必须能提供初始化的内存要求.
调用示例	pMedium = Nand_CreateMedium(pNand, 64, T_U32 BlkCnt, T_U16* RsvBlkNum)

原 型	T_PMEDIUM Medium_CreatePartition(T_PMEDIUM large, T_U32 StartPage, T_U32 PageCnt, T_U32 SecSize, T_U32 ProtectLevel)
功能概述	创建逻辑分区，也就是初始化用户使用盘
参数说明	T_PMEDIUM large: 创建 Medium 后的返回指针
	T_U32 StartPage: 分区对应 MTD 上的起始逻辑页地址
	T_U32 PageCnt:分区容量,以页为单位
	T_U32 SecSize:分区扇区大小
	T_U32 ProtectLevel:分区属性,只读或可写
返回值说明	Medium 结构体,分区结构体信息.
注意事项	PageCnt 必须少于或等于 MTD 的容量,SecSize 必须是 512 的倍数.
调用示例	pMedium = Medium_CreatePartition(pLarge, StartPage, PageCnt, 512, MEDIUM_PROTECT_LEVEL_NORMAL)

原 型	T_VOID Medium_Destroy(T_PMEDIUM obj)
功能概述	销毁 Medium，释放所有其消耗的内存空间
参数说明	T_PMEDIUM obj
返回值说明	无
注意事项	销毁后，不能再访问其上的任何对象，对象空间已经被自动释放。
调用示例	Medium_Destroy y(obj);

原 型	T_VOID MtdLib_SetCallBack(T_PMTDCALLBACK pMtdConfig);
功能概述	初始库使用的相关回调函数。
参数说明	T_PMTDCALLBACK pMtdConfig :EXNFTL 内部使用的回调函数

原 型	T_VOID MtdLib_SetCallBack(T_PMTDCALLBACK pMtdConfig);
返回值说明	无
注意事项	无
调用示例	MtdLib_SetCallBack(pMtdConfig);

原 型	void Medium_ConnectFS(T_PMEDIUM medium, const T_U8 data[], T_U32 sector,T_U32 size);
功能概述	如果是写 FAT 表数据,将分析,并解除相应的被删除文件的映射关系。
参数说明	T_PMEDIUM medium:分区指针
	const T_U8 data[] :系统写的的数据
	T_U32 sector:系统写的扇区 地址
	T_U32 size :系统写的扇区数目
返回值说明	无
注意事项	此功能仅仅在 U 盘方式使用小内存模式使用,也就是 U 盘方式系统不能工作在大内存方式。
调用示例	Medium_ConnectFS (pMediumC, buf, sector, size);

原 型	T_S8* MtdLib_GetVersion(T_VOID);
功能概述	返回 EXNFTL 的版本号信息
参数说明	无
返回值说明	返回版本号字符串
注意事项	无
调用示例	Printf(“%d\n”, MtdLib_GetVersion() )