XR806 文件系统工具

使用指南

版本号：1.0

发布时间：2021-02-20

版本历史

版本	日期	责任人	版本描述
1.0	2021-02-20	AWA 1680	创建文档。

目录

1 前言

1.1 文档简介

本文档主要介绍文件系统工具的功能说明和使用示例。

1.2 目标读者

使用 XR806 SDK 的开发人员。

1.3 适用范围

此文档适用于 XR806 SDK，支持 XR806 系列芯片产品。

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。

1.4.2 地址与数据描述方法约定

本文档在描述地址、数据时遵循如下约定：

符号	例子	说明
0x	0x0200，0x79	地址或数据以 16 进制表示。
0b	0b010，0b00 000 111	数据采用二进制表示(寄存器描述除外)。
X	00X，XX1	数据描述中，X 代表 0 或 1。 例如，00X 代表 000 或 001；XX1 代表 001，011，101 或 111。

1.4.3 数值单位约定

本文档在描述数据容量（如 NAND 容量）时，单位词头代表的是 1024 的倍数；描述频率、数据速率等时则代表的是 1000 的倍数。具体如下：

类型	符号	对应数值
数据容量（如 NAND 容量）	1 K	1024
	1 M	1 048 576
	1 G	1 073 741 824
频率，数据速率等	1 k	1000
	1 M	1 000 000
	1 G	1 000 000 000

2 概述

2.1 背景说明

XR806 SDK 支持 VFS（Virtual File System）中间件模块，与此模块对接的文件系统包括 littleFS 和 SPIFFS，为方便用户在 PC 上对特定文件进行文件系统镜像打包、文件系统镜像解包等操作，XR806 SDK 分别移植了 mklittlefs 3.0.0 版本（以下简称 littleFS 工具）和 mkspiffs 0.2.3 版本（以下简称 SPIFFS 工具）的文件系统工具。

通过使用此工具，可以满足用户的以下三种需求：

1. 用户可以在 PC 上将自定义的文件打包生成对应的文件系统镜像（二进制 bin 文件），将此 bin 文件烧录进 Flash 的指定区域，即可正常运行（例如将某一音频 MP3 文件打包进文件系统镜像，烧录后进行 MP3 播放）；
2. 用户也可以对文件系统镜像（bin 文件）进行解包，在 PC 上获取镜像内部的文件数据。例如需要将录音生成的 AMR 文件在 PC 上播放，则可以先将整个文件系统镜像 dump 出来，然后使用文件系统工具对文件系统镜像进行解包，获取 AMR 文件。（Flash 上的文件系统镜像可通过 phoenixMC 烧录工具 dump 出来，对 Flash 上特定区域的烧写与读取可参考《XR806_phoenixMC 工具_使用指南》）。
3. 用户可选择在 menuconfig 中配置完文件系统参数时，编译期间自动生成一个空的文件系统 bin 文件打包至 xr_system.img 固件中，避免用户向 Flash 中重新烧写固件时，需要手动对原来烧写的文件系统区域进行擦除，方便用户操作。

2.2 规格特性

littleFS 文件系统工具支持以下功能特点：

- 支持正向打包，将文件夹、文件等内容打包为 littleFS 镜像。
- 支持反向解包，将从 Flash 中读出来的 littleFS 镜像文件解包，生成该镜像中相同的文件目录结构。
- 支持显示镜像中文件目录。

SPIFFS 文件系统工具支持以下功能特点：

- 支持正向打包，将文件夹、文件等内容打包为 SPIFFS 镜像。
- 支持反向解包，将从 Flash 中读出来的 littleFS 镜像文件解包，生成该镜像中相同的文件目录结构。
- 支持显示镜像中文件目录。
- 支持显示当前 SPIFFS 镜像中对 Flash 的区域（Block 与 Page）使用情况。

2.3 文件位置

以 SDK 包为根目录，文件系统工具所在目录位置如下：./tools/fs_img_tools。

```
.
├── mklittlefs
├── mklittlefs.exe
├── mkspiffs
└── mkspiffs.exe
```

关键文件说明如下。

表 2-1 文件系统工具文件件说明

文件名	文件说明
mklittlefs	Linux 版本的可执行文件，64 位系统
mklittlefs.exe	Windows 版本的可执行文件，64 位系统
mkspiffs	Windows 版本的可执行文件，32/64 位系统
mkspiffs.exe	Linux 版本的可执行文件，64 位系统

说明

XR806 SDK 可在以下 GitHub 仓库获取：https://github.com/XradioTech/xr806_sdk.git

说明

文件系统工具包可在芯之联文档中心获取：https://docs.xradiotech.com/。

3 技术说明

3.1 工具基本原理

littleFS 工具与 SPIFFS 工具的实现基本原理一致，工具主要基于对应的 littleFS/SPIFFS 工程源码，在 PC 上借助内存模拟 Flash 区域，在 PC 上挂载一个相同的文件系统，进行操作。以 SPIFFS 工具为例，其基本工作原理如下：

1. 在 PC 上使用一块内存区域模拟对 Flash 区域的操作（擦、写、读）。
2. 在 SPIFFS 配置时，将文件系统的读、写、擦接口与内存区域的读、写、擦接口对接。
3. 根据用户输入的配置信息，在内存中挂载一个 SPIFFS 文件系统。
4. 打包操作时，流程如下：
   A. 根据输入配置，在内存中挂载一个 SPIFFS；
   B. 将目标目录和文件在 SPIFFS 中逐个创建；
   C. 将模拟的整个 Flash 区域数据 dump 出来，生成镜像文件。
5. 解包操作时，流程如下：
   A. 打开输入的镜像文件；
   B. 将镜像文件二进制数据按一定格式写入到模拟的 Flash 区域中；
   C. 调用 SPIFFS 接口进行挂载；
   D. 读出此文件系统中的目录和对应的文件，逐个 dump 出来。
   E. 在 PC 上获得解包后的目录与文件。

4 应用说明

4.1 应用简述

XR806 文件系统工具主要为 CLI 类型，在 linux 环境或 windows 环境下的应用步骤基本相同，如下。

1. 打开 cmd/shell 终端。
2. 进入到文件系统工具的可执行文件所在目录。
3. 检查/确认需要打包的文件和目录路径、需要解包的镜像路径。
4. 输入命令进行镜像打包/解包操作。
5. 获取打包后的镜像或解包后的目录文件。

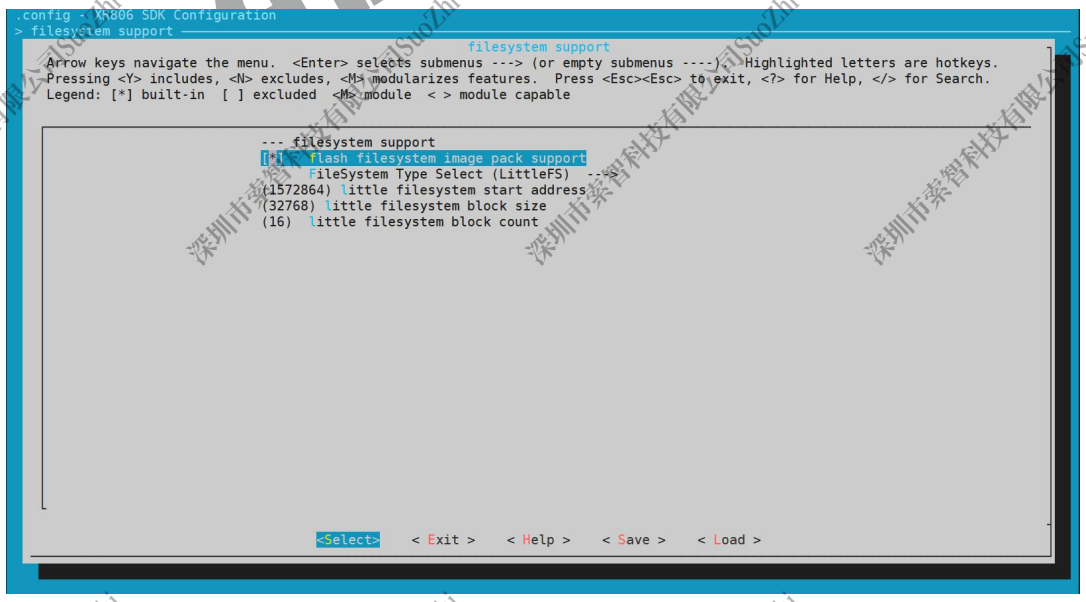
4.2 配置说明

文件系统工具的配置包括两个部分：

1. 工程编译前 menuconfig 中的配置，主要用于选择是否生成空文件系统并自动打包进固件（可参考“2.1 背景说明”章节中的需求 3）。
2. 工具命令参数的配置，主要用于配置工具在进行手动打包/解包时所使用的文件系统的参数（可参考“2.1 背景说明”章节中的需求 1、2）。

打开文件系统自动打包功能，可按如下配置：执行 make menuconfig，打开并进入 filesystem support，选中 flash filesystem image pack support。

图 4-1 打开文件系统自动打包功能



flash filesystem image pack support 选中后，SDK 工程在编译时，会根据用户对文件系统的配置（起始地址、block 大小、block 数量等）生成一个与之对应的空文件系统镜像（littleFS 或 SPIFFS），并自动打包至生成的固件中（默认为 xr_system.img）。

注意

1) 打包的文件系统类型与用户所选择的文件系统类型一致，若选择文件系统为 SPIFFS，则编译时 SDK 会根据 menuconfig 中 SPIFFS 的配置，自动生成 SPIFFS 空文件系统镜像并打包至固件。
2) 文件系统的选择与配置可参见《XR806_VFS 中间件_开发指南》文档。

工具命令参数的配置则直接在打包、解包等命令中体现，详细可参见“4.4 命令说明”章节。

4.3 格式说明

文件系统工具包括 littleFS 工具和 SPIFFS 工具，支持镜像打包、镜像解包、查看帮助信息和工具版本等功能，其命令的语法约定如下表。

表 4-1 文件系统工具命令语法约定

符号	说明
=	参数引导符
,	参数分隔符
<>	强制使用
[]	选择使用

4.4 命令说明（littleFS 工具）

4.4.1 镜像打包

信息项	说明
原型	mklittlefs.exe -c <pack_dir> -d <dbg_levl> -b <b_size> -p <p_size> -s <s_size> <image_file>
功能	将指定目录下的内容打包为 littleFS 镜像(bin 文件)
参数	<pack_dir>：需要被打包的目录（源目录需要保证已存在） <dbg_levl>：工具 debug 等级选择，一般为 0 <b_size>：block 大小（单位为字节），需要与 SDK 中 littleFS 配置一致（默认 4096） <p_size>：page 大小（单位为字节），工具内部使用，一般为 256 <s_size>：littleFS 镜像大小，即文件系统占用 Flash 空间大小（单位为字节），需要与 SDK 中 littleFS 配置一致（默认 524288，即 512KB） <image_file>：需要生成的 littleFS 镜像文件名，一般为“.bin”后缀

信息项	说明
响应	打包成功，会打印镜像中的目录结构
示例	将当前路径下 fs 文件夹中的所有内容打包至 lfs.bin 镜像： mklittlefs.exe -c ./fs/ -d 0 -b 4096 -p 256 -s 524288 ./lfs.bin
备注	<image_file>为生成的目标镜像名，若<image_file>参数中包含路径信息，需要保证路径中的目录事先已被创建

注意

以上命令格式均以 windows 的 cmd 终端为例，使用的可执行文件为 mklittlefs.exe；若在 linux 终端下使用，则只需将命令中的“mklittlefs.exe”替换为“./mklittlefs”即可，下同。

说明

XR806 SDK 中文件系统的选择与默认参数配置可参见《XR806_VFS 中间件_开发指南》文档，下同。

4.4.2 镜像解包

信息项	说明
原型	mklittlefs.exe -u <dest_dir> -d <dbg_levl> -b <b_size> -p <p_size> -s <s_size> <image_file>
功能	将 littleFS 镜像(bin 文件)解包为对应的目录文件，并存放在指定目录下
参数	<dest_dir>：解包获取到的目录文件的存放路径 <dbg_levl>：工具 debug 等级选择，一般为 0 <b_size>：block 大小（单位为字节），需要与 SDK 中 littleFS 配置一致（默认 4096） <p_size>：page 大小（单位为字节），工具内部使用，一般为 256 <s_size>：littleFS 镜像大小，即文件系统占用 Flash 空间大小（单位为字节），需要与 SDK 中 littleFS 配置一致（默认 524288，即 512KB） <image_file>：需要进行解包的 littleFS 镜像文件
响应	解包成功，会输出获取到的文件列表和文件大小
示例	将当前路径下的 flash_lfs.bin 解包到 dump 目录下： mklittlefs.exe -u ./dump/ -d 0 -b 4096 -p 256 -s 524288 ./flash_lfs.bin
备注	<dest_dir>目录若不存在会被自动创建

4.4.3 查看镜像中目录结构信息

信息项	说明
原型	mklittlefs.exe -l -b <b_size> -p <p_size> -s <s_size> <image_file>
功能	列出 littleFS 镜像中的目录结构信息
参数	<b_size>：block 大小（单位为字节），需要与 SDK 中 littleFS 配置一致（默认 4096） <p_size>：page 大小（单位为字节），工具内部使用，一般为 256 <s_size>：littleFS 镜像大小，即文件系统占用 Flash 空间大小（单位为字节），需要与 SDK 中 littleFS 配置一致（默认 524288，即 512KB） <image_file>：需要进行查看的 littleFS 镜像文件
响应	打印输出该镜像中的目录列表和文件大小
示例	查看当前路径下 flash_lfs_dmp.bin 镜像中的目录结构： mklittlefs.exe -l -b 4096 -p 256 -s 524288 ./flash_lfs_dmp.bin
备注	N/A

4.4.4 查看工具版本

信息项	说明
原型	mklittlefs.exe --version
功能	查看 littleFS 工具版本
参数	N/A
响应	输出版本信息
示例	N/A
备注	N/A

4.4.5 查看帮助信息

信息项	说明
原型	mklittlefs.exe -h
功能	查看 littleFS 工具帮助信息
参数	N/A
响应	输出各命令参数的帮助信息
示例	N/A
备注	N/A

4.5 命令说明（SPIFFS 工具）

4.5.1 镜像打包

信息项	说明
原型	mkspiffs.exe -c <pack_dir> -d <dbg_levl> -b <b_size> -p <p_size> -s <s_size> <image_file>
功能	将指定目录下的内容打包为 SPIFFS 镜像(bin 文件)
参数	<pack_dir>：需要被打包的目录（源目录需要保证已存在） <dbg_levl>：工具 debug 等级选择，一般为 0 <b_size>：block 大小（单位为字节），需要与 SDK 中 SPIFFS 配置一致（默认 4096） <p_size>：page 大小（单位为字节），需要与 SDK 中 SPIFFS 配置一致（默认 256） <s_size>：SPIFFS 镜像大小，即文件系统占用 Flash 空间大小（单位为字节），需要与 SDK 中 SPIFFS 配置一致（默认 131072，即 128KB） <image_file>：需要生成的 SPIFFS 镜像文件名，一般为“.bin”后缀
响应	打包成功，会打印镜像中的目录结构
示例	将当前路径下 fs 文件夹中的所有内容打包至 spiffs.bin 镜像： mkspiffs.exe -c ./fs/ -d 0 -b 4096 -p 256 -s 524288 ./spiffs.bin
备注	若<image_file>参数中包含路径信息，需要保证路径中的目录事先已被创建

注意

以上命令格式均以 windows 的 cmd 终端为例，使用的可执行文件为 mkspiffs.exe；若在 linux 终端下使用，则只需将命令中的“mkspiffs.exe”替换为“./mkspiffs”即可，下同。

4.5.2 镜像解包

信息项	说明
原型	mkspiffs.exe -u <dest_dir> -d <dbg_levl> -b <b_size> -p <p_size> -s <s_size> <image_file>
功能	将 SPIFFS 镜像(bin 文件)解包为对应的目录文件，并存放在指定目录下
参数	<dest_dir>：解包获取到的目录文件的存放路径 <dbg_levl>：工具 debug 等级选择，一般为 0 <b_size>：block 大小（单位为字节），需要与 SDK 中 SPIFFS 配置一致（默认 4096） <p_size>：page 大小（单位为字节），需要与 SDK 中 SPIFFS 配置一致（默认 256）

信息项	说明
	<s_size>：SPIFFS 镜像大小，即文件系统占用 Flash 空间大小（单位为字节），需要与 SDK 中 SPIFFS 配置一致（默认 131072，即 128KB） <image_file>：需要进行解包的 SPIFFS 镜像文件
响应	解包成功，会输出获取到的文件列表和文件大小
示例	将当前路径下的 flash_spiffs.bin 解包到 dump 目录下： mkspiffs.exe -u ./dump/ -d 0 -b 4096 -p 256 -s 524288 ./flash_spiffs.bin
备注	<dest_dir>目录若不存在会被自动创建

4.5.3 查看镜像中文件信息

信息项	说明
原型	mkspiffs.exe -l -b <b_size> -p <p_size> -s <s_size> <image_file>
功能	列出 SPIFFS 镜像中的文件信息
参数	<b_size>：block 大小（单位为字节），需要与 SDK 中 SPIFFS 配置一致（默认 4096） <p_size>：page 大小（单位为字节），需要与 SDK 中 SPIFFS 配置一致（默认 256） <s_size>：SPIFFS 镜像大小，即文件系统占用 Flash 空间大小（单位为字节），需要与 SDK 中 SPIFFS 配置一致（默认 131072，即 128KB） <image_file>：需要进行查看的 SPIFFS 镜像文件
响应	打印输出该镜像中的目录列表和文件大小
示例	查看当前路径下 flash_spiffs_dmp.bin 镜像中的目录结构： mklittlefs.exe -l -b 4096 -p 256 -s 524288 ./flash_spiffs_dmp.bin
备注	N/A

4.5.4 查看镜像对 Flash 的使用情况

信息项	说明
原型	mkspiffs.exe -i -b <b_size> -p <p_size> -s <s_size> <image_file>
功能	输出显示 SPIFFS 镜像中对 Flash 区域的使用情况
参数	<b_size>：block 大小（单位为字节），需要与 SDK 中 SPIFFS 配置一致（默认 4096） <p_size>：page 大小（单位为字节），需要与 SDK 中 SPIFFS 配置一致（默认 256） <s_size>：SPIFFS 镜像大小，即文件系统占用 Flash 空间大小（单位为字节），需要与 SDK 中 SPIFFS 配置一致（默认 131072，即 128KB）

信息项	说明
	<image_file>：需要进行查看的 SPIFFS 镜像文件
响应	打印输出此文件系统镜像所占用的 Flash 区域中 block 和 page 的使用情况
示例	查看当前路径下 flash_spiffs_dmp.bin(SPIFFS 镜像)对 Flash 的 block 和 page 使用情况： mkspiffs.exe -i -b 4096 -p 256 -s 131072 ./flash_spiffs_dmp.bin
备注	N/A

说明

不同的文件系统镜像对 Flash 物理区域的使用不尽相同。SPIFFS 源码中提供了 SPIFFS_vis()接口可将 block 和 page 使用情况进行打印输出，输出的结果即为此文件系统所在的 Flash 物理区域使用情况。因此，SPIFFS 工具也使用了此接口，通过“-i”命令在终端进行可视化输出。

4.5.5 查看工具版本

信息项	说明
原型	mkspiffs.exe --version
功能	查看 SPIFFS 工具版本
参数	N/A
响应	输出版本信息
示例	N/A
备注	N/A

4.5.6 查看帮助信息

信息项	说明
原型	mkspiffs.exe -h
功能	查看 SPIFFS 工具帮助信息
参数	N/A
响应	输出各命令参数的帮助信息
示例	N/A
备注	N/A

5 示例说明

XR806 文件系统工具支持用户手动打包(或解包)和编译时自动打包两种功能，下面分别对手动打包（5.1 章节、5.2 章节）和自动打包（5.3 章节）两种用法进行使用说明。

5.1 LittleFS 工具使用示例

5.1.1 示例简介

本示例主要介绍 littleFS 工具的手动打包、解包等功能的操作。打包过程可以将指定的目录文件生成文件系统镜像(bin 文件)；解包可以获取文件系统镜像(bin 文件)中的目录文件数据。

因此，借助 PhoenixMC 工具的调试功能，用户可以灵活的将文件系统镜像(bin 文件)烧写至 Flash 指定区域，供 XR806 直接使用。或者使用 PhoenixMC 工具读取 Flash 文件系统镜像(bin 文件)，dump 出目录文件数据后，供用户在 PC 上直接查阅。

说明

PhoenixMC 工具调试功能的使用可参考《XR806_PhoenixMC 工具_使用指南》文档。

5.1.2 准备工作

使用 XR806 文件系统工具(littleFS)的准备工作如下：

1. XR806 文件系统工具包
2. 需要进行打包的目录与文件
3. 需要进行解包的 littleFS 文件系统镜像(bin 文件)

本示例假设工具软件和需要打包的目录在同一路径，文件结构如下（flash_dmp.bin 中的目录结构与 fs 目录一致）：

```
.
├── fs
│   ├── book1
│   │   └── text2.txt
│   └── music3
│       └── sound4.mp3
├── mklittlefs
├── mklittlefs.exe
└── flash_dmp.bin
```

5.1.3 镜像打包示例

说明

示例中的参数以 SDK 中 littleFS 默认配置为参考，即：block_size: 4K; block_count: 128; filesystem_size: 512K，下同。

镜像打包示例主要完成功能为：将 fs 目录下的内容打包为 lfs.bin 镜像文件。

5.1.3.1 操作步骤

1）Windows 版本：

A）打开 cmd 终端并进入到 mklittlefs.exe 所在目录

B）执行命令：

```
mklittlefs.exe -c ./fs/ -d 0 -b 4096 -p 256 -s 524288 ./lfs.bin
```

2）Linux 版本：

A）打开 shell 终端并进入到 mklittlefs 所在目录

B）执行命令：

```
./mklittlefs -c ./fs/ -d 0 -b 4096 -p 256 -s 524288 ./lfs.bin
```

5.1.3.2 效果展示

Windows 版本和 Linux 版本执行后效果一致，终端输出如下：

```
/book1/text2.txt
/music3/sound4.mp3
```

输出内容表示在此 lfs.bin 中，根目录下存在 book1 目录和 music3 目录，且分别存在 text2.txt 文件与 sound4.mp3 文件，符合预期，打包成功。将此 lfs.bin 文件，烧写至 Flash 指定区域，littleFS 文件系统可正常运行。

说明

Windows 版本和 Linux 版本的工具使用除了使用的可执行文件名称不一样外，其余参数均相同，因此后文仅以 Windows 版本的 cmd 终端为例进行介绍，下同。

5.1.4 镜像解包示例

镜像解包示例主要完成功能为：将 flash_dmp.bin 镜像解包为对应的目录文件，并存放在当前路径下的 dump 文件夹中。

（flash_dmp.bin 文件可借助 PhoenixMC 工具的调试功能从 Flash 中读取）

5.1.4.1 操作步骤

A）打开 cmd 终端并进入到 mklittlefs.exe 所在目录

B）执行命令：

```
mklittlefs.exe -u ./dump/ -d 0 -b 4096 -p 256 -s 524288 ./flash_dmp.bin
```

若目标文件夹 dump 不存在，会被自动创建。

5.1.4.2 效果展示

```
Directory ./dump/ does not exists. Try to create it.
book1/text2.txt    > ./dump/book1/text2.txt            size: 35 Bytes
music3/sound4.mp3              > ./dump/music3/sound4.mp3        size: 43 Bytes
```

终端输出显示：将 book1/text2.txt 文件拷贝到了./dump/book1 目录，将 music3/sound4.mp3 拷贝到了./dump/music3/目录，解包后输出了文件列表和文件大小，且查阅 PC 端的目录后，dump 出的目录与文件真实存在，符合预期。

5.1.5 查看镜像中目录结构示例

此示例主要打印镜像 flash_dmp.bin 中的文件目录结构信息。

5.1.5.1 操作步骤

A）打开 cmd 终端并进入到 mklittlefs.exe 所在目录

B）执行命令：

```
mklittlefs.exe -l -b 4096 -p 256 -s 524288 ./flash_dmp.bin
```

5.1.5.2 效果展示

```
<dir>     /book1
35          /book1/text2.txt
<dir>     /music3
43          /music3/sound4.mp3
```

成功输出镜像中的目录和对应的文件信息，其中 text2.txt 的文件大小 35 字节，sound4.mp3 的文件大小为 43 字节，符合预期。

5.1.6 查看版本与帮助示例

本示例功能主要为查看 littleFS 工具的版本和命令帮助信息。

5.1.6.1 操作步骤

A）打开 cmd 终端并进入到 mklittlefs.exe 所在目录

B）执行命令：

```
mklittlefs.exe --version
mklittlefs.exe -h
```

5.1.6.2 效果展示

执行命令后，终端输出如下：

```
./mklittlefs    version: 0.2.3-31-g295fe9b

USAGE:
    ./mklittlefs    {-c <pack_dir>|-u <dest_dir>|-l} [-d <0-5>] [-a] [-b <number>] [-p <number>] [-s
                    <number>] [--] [--version] [-h] <image_file>

Where:
    -c <pack_dir>,    --create <pack_dir>
      (OR required)    create littlefs image from a directory
            -- OR --
    -u <dest_dir>,    --unpack <dest_dir>
      (OR required)    unpack littlefs image to a directory
            -- OR --
    -l,    --list
      (OR required)    list files in littlefs image

    -d <0-5>,    --debug <0-5>
      Debug level. 0 means no debug output.

    -a,    --all-files
      when creating an image, include files which are normally ignored;
      currently only applies to '.DS_Store' files and '.git' directories

    -b <number>,    --block <number>
      fs block size, in bytes

    -p <number>,    --page <number>
      fs page size, in bytes

    -s <number>,    --size <number>
      fs image size, in bytes

    --,    --ignore_rest
      Ignores the rest of the labeled arguments following this flag.

    --version
      Displays version information and exits.
```

```
-h,   --help
    Displays usage information and exits.

<image_file>
    (required)    littlefs image file
```

版本和帮助信息输出成功，符合预期。

5.2 SPIFFS 工具使用示例

5.2.1 示例简介

同“5.1.1 示例简介”章节，本示例主要介绍 SPIFFS 工具的手动打包、解包等功能的操作。

5.2.2 准备工作

同“5.1.2 准备工作”章节类似，使用 XR806 文件系统工具(SPIFFS)的准备工作如下：

1．XR806 文件系统工具包

2．需要进行打包的目录与文件

3．需要进行解包的 SPIFFS 文件系统镜像(bin 文件)

本示例假设工具软件和需要打包的目录在同一路径，文件结构如下（flash_dmp.bin 中的目录结构与 fs 目录一致）：

```
.
├── fs
│   ├── book1
│   │   └── text2.txt
│   └── music3
│       └── sound4.mp3
├── mkspiffs
├── mkspiffs.exe
└── flash_dmp.bin
```

5.2.3 镜像打包示例

说明

示例中的参数以 SDK 中 SPIFFS 默认配置为参考，即：block_size: 4K; page_size: 256Byte; filesystem_size: 128K，下同。

镜像打包示例主要完成功能为：将 fs 目录下的内容打包为 spiffs.bin 镜像文件。

5.2.3.1 操作步骤

A）打开 cmd 终端并进入到 mkspiffs.exe 所在目录

B）执行命令：

```
mkspiffs.exe -c ./fs/ -d 0 -b 4096 -p 256 -s 131072 ./spiffs.bin
```

5.2.3.2 效果展示

终端输出如下：

```
/book1/text2.txt
/music3/sound4.mp3
```

输出内容表示在此 spiffs.bin 中，存在两个文件，其文件名分别为“/book1/text2.txt”与“/music3/sound4.mp3”，符合预期，打包成功。将此 lfs.bin 文件，烧写至 Flash 指定区域，SPIFFS 文件系统可正常运行。

5.2.4 镜像解包示例

镜像解包示例主要完成功能为：将 flash_dmp.bin 镜像解包为对应的目录文件，并存放在当前路径下的 dump 文件夹中。

5.2.4.1 操作步骤

A）打开 cmd 终端并进入到 mkspiffs.exe 所在目录

B）执行命令：

```
mkspiffs.exe -u ./dump/ -d 0 -b 4096 -p 256 -s 131072 ./flash_dmp.bin
```

5.2.4.2 效果展示

终端输出如下：

```
Directory ./dump/ does not exists. Try to create it.
book1/text2.txt    > ./dump/book1/text2.txt          size: 35 Bytes
music3/sound4.mp3          > ./dump/music3/sound4.mp3          size: 43 Bytes
```

终端输出显示：将 book1/text2.txt 文件拷贝到了./dump/book1 目录，将 music3/sound4.mp3 拷贝到了./dump/music3/目录，解包后输出了文件列表和文件大小，且查阅 PC 端的目录后，dump 出的目录与文件真实存在，符合预期。

5.2.5 查看镜像中的文件示例

此示例主要打印镜像 flash_dmp.bin 中的文件目录结构信息。

5.2.5.1 操作步骤

A）打开 cmd 终端并进入到 mkspiffs.exe 所在目录

B）执行命令：

```
mkspiffs.exe -l -b 4096 -p 256 -s 131072 ./flash_dmp.bin
```

5.2.5.2 效果展示

终端输出如下：

```
35        /book1/text2.txt
43        /music3/sound4.mp3
```

成功输出镜像中对应的文件信息，其中 text2.txt 的文件大小 35 字节，sound4.mp3 的文件大小为 43 字节，符合预期。

注意

SPIFFS 不支持真实目录，因此与 littleFS 不同的是，此处不会有“<dir> /book1”等目录信息。

5.2.6 查看镜像对 Flash 使用情况示例

本示例主要演示打印 SPIFFS 镜像（flash_dmp.bin）中对 Flash 的使用情况。

5.2.6.1 操作步骤

A）打开 cmd 终端并进入到 mkspiffs.exe 所在目录

B）执行命令：

```
mkspiffs.exe -i -b 4096 -p 256 -s 131072 ./flash_dmp.bin
```

5.2.6.2 效果展示

终端输出如下：

```
0000 idid_____________ era_cnt: 0
0001 _____________ era_cnt: 0
0002 _____________ era_cnt: 0
0003 _____________ era_cnt: 0
0004 _____________ era_cnt: 0
0005 _____________ era_cnt: 0
0006 _____________ era_cnt: 0
0007 _____________ era_cnt: 0
0008 _____________ era_cnt: 0
0009 _____________ era_cnt: 0
000a _____________ era_cnt: 0
000b _____________ era_cnt: 0
000c _____________ era_cnt: 0
000d _____________ era_cnt: 0
000e _____________ era_cnt: 0
000f _____________ era_cnt: 0
0010 _____________ era_cnt: 0
0011 _____________ era_cnt: 0
```

```
0012 ________________    era_cnt: 0
0013 ________________    era_cnt: 0
0014 ________________    era_cnt: 0
0015 ________________    era_cnt: 0
0016 ________________    era_cnt: 0
0017 ________________    era_cnt: 0
0018 ________________    era_cnt: 0
0019 ________________    era_cnt: 0
001a ________________    era_cnt: 0
001b ________________    era_cnt: 0
001c ________________    era_cnt: 0
001d ________________    era_cnt: 0
001e ________________    era_cnt: 0
001f ________________    era_cnt: 0
era_cnt_max: 1
last_errno:  0
blocks:      32
free_blocks: 31
page_alloc:  4
page_delet:  0
used:        1004 of 113201
total: 113201
used: 1004
```

从输出中可以看出，SPIFFS 系统中使用了 1 个 block 中的 4 个 page(2 个 ipage 和 2 个 dpage)，与“/book1/text2.txt”文件以及“/music3/sound4.mp3”文件大小可以对应，符合预期。

5.2.7 查看版本与帮助示例

本示例演示的主要功能为查看 SPIFFS 工具的版本和命令帮助信息。

5.2.7.1 操作步骤

A）打开 cmd 终端并进入到 mkspiffs.exe 所在目录

B）执行命令：

```
mkspiffs.exe --version
mkspiffs.exe -h
```

5.2.7.2 效果展示

终端输出如下：

```
mkspiffs ver. 0.2.3
Build configuration name: generic
SPIFFS ver. 0.3.7-5-gf5e26c4
Extra build flags: (none)
```

```
SPIFFS configuration:
   SPIFFS_OBJ_NAME_LEN: 32
...

USAGE:
    ./mkspiffs    {-c <pack_dir>|-u <dest_dir>|-l|-i} [-d <0-5>] [-a] [-b
                  <number>] [-p <number>] [-s <number>] [--] [--version] [-h]
                  <image_file>

Where:

    -c <pack_dir>,    --create <pack_dir>
       (OR required)    create spiffs image from a directory
             -- OR --
    -u <dest_dir>,    --unpack <dest_dir>
       (OR required)    unpack spiffs image to a directory
             -- OR --
    -l,    --list
       (OR required)    list files in spiffs image
             -- OR --
    -i,    --visualize
       (OR required)    visualize spiffs image


    -d <0-5>,    --debug <0-5>
       Debug level. 0 means no debug output.

    -a,    --all-files
       when creating an image, include files which are normally ignored;
       currently only applies to '.DS_Store' files and '.git' directories

    -b <number>,    --block <number>
       fs block size, in bytes

    -p <number>,    --page <number>
       fs page size, in bytes

    -s <number>,    --size <number>
       fs image size, in bytes

    --,    --ignore_rest
       Ignores the rest of the labeled arguments following this flag.

    --version
```

```
    Displays version information and exits.

-h,   --help
    Displays usage information and exits.

<image_file>
    (required)   spiffs image file
```

版本和帮助信息输出成功，符合预期。

5.3 镜像自动打包示例

5.3.1 示例简介

本示例主要以 littleFS 为例，借助 XR806 SDK 的 project/demo/hello_demo 工程，通过在 menuconfig 中配置选择自动打包功能，来演示如何进行空文件系统的自动打包。

自动打包的目的是为了避免用户每一次烧写前都需要手动擦除文件系统所在的 Flash 区域（若不对此区域擦除，则烧写前的文件可能会残留在 Flash 区域中，影响下一次使用）。

说明

XR806 SDK 可在以下 GitHub 仓库获取：https://github.com/XradioTech/xr806_sdk.git

5.3.2 准备工作

镜像自动打包示例的硬件准备有如下。

1. 评估板：运行工程代码。
2. 串口线：连接评估板的 UART0 插针，用于 console 控制台的输入输出。
3. PC 机：用于镜像烧录和 console 控制的输入输出。

本示例的软件准备，包括烧写工具、代码编译和烧写操作，请参见《XR806_SDK_快速入门指南》。

基于 hello_demo 工程，需要准备的代码如下：

1. 在 command.c 文件的 g_main_cmds 数组中添加一行代码以支持 fs 命令

```
static const struct cmd_data g_main_cmds[] = {
...
   { "fs",         cmd_fs_exec },
...
};
```

2. 在 main.c 文件的 main 函数中注释掉打印“Hello world”的语句，方便控制台输出信息查看。

```
int main(void)
{
    platform_init();
    return 0;
}
```

5.3.3 操作步骤

1. 在 menuconfig 中打开 filesystem support 选项，并选择文件系统类型为 littleFS，其他配置默认。
2. 编译并烧写固件 1 至开发板运行。
3. 挂载文件系统，使用控制台创建一个名为“data/test.txt”的文件。
4. 向文件中写入内容“abcd123456789”并关闭文件。
5. 再次编译烧写固件 1 至开发板运行，查看文件“data/test.txt”是否存在。
6. 在 menuconfig 的 filesystem support 选项中打开 flash filesystem image pack support 选项，其他配置不变。
7. 编译并烧写固件 2 至开发板运行，查看文件“data/test.txt”是否存在。

说明

若步骤 1 中选择文件系统类型为 SPIFFS，则步骤 6 中的打包功能会自动与 SPIFFS 对应，生成一个空的 SPIFFS 镜像(fs_spiffs.bin)打包至 xr_system.img 固件。

5.3.4 效果展示

按照上述操作，编译后固件 1 的 Flash Layout 如下：

```
Flash Layout:
sec bin 0 boot_40M.bin      :    flash_offs: 0x00000000(    0K)    data_size: 0x000025F4(   10K)
sec bin 1 app.bin           :    flash_offs: 0x00004000(   16K)    data_size: 0x00006470(   26K)
sec bin 2 app_xip.bin       :    flash_offs: 0x00029800( 166K)    data_size: 0x000211F0( 133K)
```

挂载文件系统后，创建“data/test.txt”文件控制台显示如下：

```
$ fs open data/test.txt
[cmd] open success
<ACK> 200 OK
$ fs write abcd123456789
<ACK> 200 OK
$ fs close
<ACK> 200 OK
$ fs opendir data/
open directory data/ success
```

```
<ACK> 200 OK
$ fs readdir
., type:directory
.., type:directory
test.txt, type:file, file size:13
<ACK> 200 OK
$ fs closedir
close directory success
<ACK> 200 OK
$ fs open data/test.txt
[cmd] open success
<ACK> 200 OK
$ fs read 256
abcd123456789
<ACK> 200 OK
$ fs close
<ACK> 200 OK
```

可以看到，文件创建与写入成功。

再次编译烧写固件 1 至开发板后，查看该文件状态：

```
$ fs opendir data/
open directory data/ success
<ACK> 200 OK
$ fs readdir
., type:directory
.., type:directory
test.txt, type:file, file size:13
<ACK> 200 OK
$ fs closedir
close directory success
<ACK> 200 OK
$ fs open data/test.txt
[cmd] open success
<ACK> 200 OK
$ fs read 256
abcd123456789
<ACK> 200 OK
$ fs close
<ACK> 200 OK
```

发现仍然存在，表明固件重新烧写后，并没有擦除到文件系统所在区域，符合预期。

在 menuconfig 的 filesystem support 选项中打开 flash filesystem image pack support 选项后，查看编译生成固件 2 的 Flash Layout 如下：

```
Flash Layout:
sec bin 0 boot_40M.bin      :    flash_offs: 0x00000000(     0K)    data_size: 0x000025F4(    10K)
sec bin 1 app.bin           :    flash_offs: 0x00004000(    16K)    data_size: 0x00006470(    26K)
sec bin 2 app_xip.bin       :    flash_offs: 0x00029800( 166K)    data_size: 0x000211F0( 133K)
raw bin 0 fs_littlefs.bin   :    flash_offs: 0x00180000(1536K)    data_size: 0x00080000( 512K)
```

多了 fs_littlefs.bin 文件，符合预期。

编译并烧写固件 2 至开发板运行，再次查看文件“data/test.txt”状态：

```
$ fs opendir data/
open directory data/ success
<ACK> 200 OK
$ fs readdir
., type:directory
.., type:directory
<ACK> 200 OK
$ fs closedir
close directory success
<ACK> 200 OK
```

控制台输出显示，data/目录下的文件已经被清空，“data/test.txt”文件已不存在，符合预期。

6 常见问题

6.1 文件系统镜像解包错误

使用文件系统工具时，对 bin 文件进行手动解包操作可以获取此文件系统镜像中的目录和文件信息，若出现报错，如“error: lfs error: 3409：Unsupported name_max(255 > 32)”等等，可能是 bin 文件不干净造成的。

用户可以优先检查开发板上文件系统的区域（比如 Flash 尾部的 512K 区域）是否在烧写前被手动擦除。

6.2 无法打开镜像文件

使用文件系统工具进行打包时，若出现“error: failed to open image file.”等错误提示，用户可以优先检查目标路径是否被事先创建。

附录 A： 术语表

表 A-1 术语表

A		
AMR	Adaptive multi-Rate compression	自适应多速率音频压缩
C		
CLI	Command Line Interface	命令行界面
V		
VFS	Virtual File System	虚拟文件系统

著作权声明

商标声明

免责声明