Huawei LiteOS 分散加载应用指南

文档版本　00B01

发布日期　2016-05-10

版权所有 © 深圳市海思半导体有限公司 2016。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

商标声明

 、HISILICON、海思和其他海思商标均为深圳市海思半导体有限公司的商标。

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。

注意

您购买的产品、服务或特性等应受海思公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，海思公司对本文档内容不做任何明示或默示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

深圳市海思半导体有限公司

地址：　深圳市龙岗区坂田华为基地华为电气生产中心　邮编：518129

网址：　http://www.hisilicon.com

客户服务电话：　+86-755-28788858

客户服务传真：　+86-755-28357515

客户服务邮箱：　support@hisilicon.com

前 言

概述

本文档主要介绍基于 Huawei LiteOS 的分散加载 Sample 的使用。

产品版本

与本文档相对应的产品版本如下。

产品名称	产品版本
Hi3516A	V100
Hi3516D	V100
Hi3518E	V200
Hi3518E	V201
Hi3516C	V200
Hi3519	V100
Hi3519	V101

读者对象

本文档（本指南）主要适用于以下工程师：

- 技术支持工程师
- 软件开发工程师

修订记录

修订记录累积了每次文档更新的说明。最新版本的文档包含以前所有文档版本的更新内容。

海思专有和保密信息
版权所有 © 深圳市海思半导体有限公司

修订日期	版本	修订说明
2016-05-10	00B01	第一次临时版本发布

海思专有和保密信息
版权所有 © 深圳市海思半导体有限公司

目 录

海思专有和保密信息
版权所有 © 深圳市海思半导体有限公司

插图目录

海思专有和保密信息
版权所有 © 深圳市海思半导体有限公司

表格目录

海思专有和保密信息
版权所有 © 深圳市海思半导体有限公司

1 分散加载概述

1.1 分散加载应用场景

1. 对启动速度有要求。

   以 DV 为例，为了保证尽快出图像，将媒体等模块作为第一部分，在 uboot 里读取。待媒体模块加载完成，在 APP 里读取第二部分镜像，这一部分主要是录像、拍照、网络等模块。

2. WiFi 待机唤醒。

   WiFi 待机唤醒是分散加载的一个特殊应用。这种情况下，把 OS 和 WiFi 作为第一部分，其余的作为第二部分。

1.2 原理说明

详细原理说明请参考《Huawei LiteOS Kernel Deverloper Guide》（下文简称为开发手册）中相关章节。本文档主要针对分散加载 Sample 进行说明，是对开发手册的补充。用户需要先阅读开发手册，了解分散加载的原理后，再阅读本文档。

1.3 本文档及附属 Sample 作用

因为分散加载功能比较复杂，如果只是阅读开发手册，难以真正理解分散加载。为了帮助用户加深对这个功能的理解以及方便用户进行移植，所以提供本文档和 Sample 工程。

海思专有和保密信息
版权所有 © 深圳市海思半导体有限公司

2 分散加载 Sample 使用说明

2.1 Sample 工程说明

2.1.1 支持的硬件平台

Sample 与具体硬件关联不大，只要 Huawei LiteOS 支持的硬件，都可以运行这个 Sample。

2.1.2 目录结构说明

Sample 的代码在 Huawei LiteOS 的发布包的 sample 目录下，所在目录名为 sample_scatter，目录下的内容如表 2-1 所示。

表2-1 目录说明

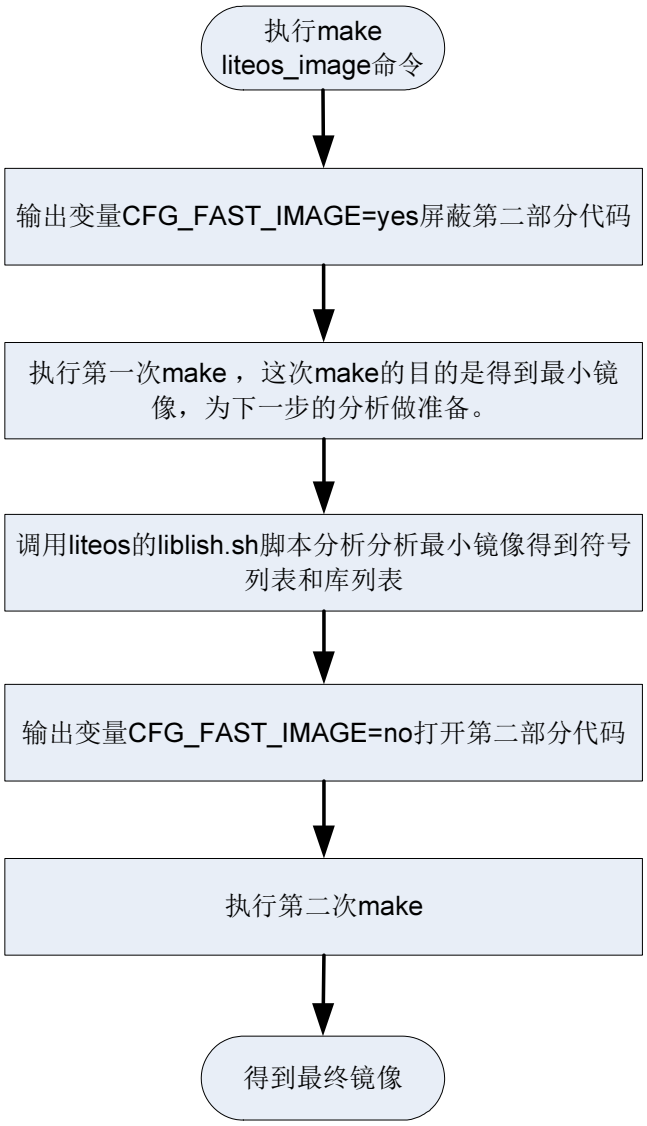
目录或文件名	说明
out	输出的 o 文件、lib 文件、elf 文件和烧录镜像。
script	工程相关脚本。
src	源代码目录。
cfg.mak	Sample 工程的配置文件。
Makefile	Sample 工程的 Makefile。

2.2 Sample 编译流程

在 sample_scatter 目录下，执行 make liteos_image 命令。该命令调用脚本 script/gen_image.sh，编译过程的流程如图 2-1 所示，这个流程就是 gen_image.sh 脚本的内容，用户可以参照脚本内容进行理解。

海思专有和保密信息
版权所有 © 深圳市海思半导体有限公司

图2-1 Sample 的编译过程



2.3 Sample 操作步骤

Sample 操作步骤如下：

步骤 1. 进入 sample_scatter 目录，输入 make liteos_image 命令。

步骤 2. 等编译完成，在 shell 窗口的最后会打印长度值，如图 2-2 所示。0x55000 表示的就是关键业务部分的字节长度，这个长度值需要被设置到 uboot 的 bootcmd 中。

海思专有和保密信息
版权所有 © 深圳市海思半导体有限公司

图2-2 编译完成时的关键业务长度提示

```
###### ###### ##### ##### ##### ##### #####
#######Calculate the size of scatter#######
the size is 0x55000
####################end####################
```

步骤 3. 用户可以用 arm-hisivxxx-linux-readelf -S out/bin/sample（hisivxxx 根据用户使用的具体工具链版本来定，例如 hisiv300、hisiv500 等）来读取可执行的 Section 信息。当前读取的情况如图 2-3 所示。可见除了一般常见的 text、data、bss 之外，还多了 fast_text 等新增的 Section。

上面得到的 0x55000 就是根据 text 段地址前面的长度经过对齐得到的，详细的算法请参考 liteos/tools/scripts/scatter_sr 目录下的 scatter_size.sh 脚本。

图2-3 用 readelf 命令读取可执行文件的 Section 信息

```
There are 24 section headers, starting at offset 0x126948:

Section Headers:
  [Nr] Name              Type            Addr     Off    Size   ES Flg Lk Inf Al
  [ 0]                   NULL            00000000 000000 000000 00      0   0  0
  [ 1] .debug_aranges    PROGBITS        00000000 0ed430 0006b8 00      0   0  8
  [ 2] .debug_info       PROGBITS        00000000 0edae8 015e44 00      0   0  1
  [ 3] .debug_abbrev     PROGBITS        00000000 10392c 003fe3 00      0   0  1
  [ 4] .debug_line       PROGBITS        00000000 10790f 0067de 00      0   0  1
  [ 5] .debug_frame      PROGBITS        00000000 10e0f0 001514 00      0   0  4
  [ 6] .debug_str        PROGBITS        00000000 10f604 005cf2 01  MS  0   0  1
  [ 7] .debug_loc        PROGBITS        00000000 1152f6 00defe 00      0   0  1
  [ 8] .comment          PROGBITS        00000000 1231f4 000032 01  MS  0   0  1
  [ 9] .ARM.attributes   ARM_ATTRIBUTES  00000000 123226 000036 00      0   0  1
  [10] .rom_vectors      PROGBITS        80008000 0000d4 000048 00  AX  0   0  4
  [11] .fast_rodata      PROGBITS        80009000 000120 004b98 00   A  0   0  8
  [12] .fast_init_array  INIT_ARRAY      8000db98 004cb8 000004 00  WA  0   0  4
  [13] .fast_text        PROGBITS        8000e000 005120 03e9f0 00  AX  0   0  4
  [14] .fast_data        PROGBITS        8004d000 044120 00e454 00  WA  0   0  4
  [15] .got              PROGBITS        8005c000 052578 000084 04  WA  0   0  4
  [16] .text             PROGBITS        8005d000 053578 05c650 00  AX  0   0  4
  [17] .rodata           PROGBITS        800ba000 0b0578 03b688 00   A  0   0  8
  [18] .data             PROGBITS        800f6000 0ec578 000eb4 00  WA  0   0  8
  [19] .bss              NOBITS          800f7000 0ed42c 27ae34 00 WAT  0   0  8
  [20] .debug_ranges     PROGBITS        00000000 12325c 0035f0 00      0   0  1
  [21] .shstrtab         STRTAB          00000000 12684c 0000f9 00      0   0  1
  [22] .symtab           SYMTAB          00000000 126d08 01f080 10     23 5888  4
  [23] .strtab           STRTAB          00000000 145d88 00e2ad 00      0   0  1
Key to Flags:
```

步骤 4. 烧录镜像并设置 uboot 环境变量。

- 将 out/burn/sample.bin 烧录到 Flash 地址 0x100000 的位置。
- 设置 bootcmd：setenv bootcmd 'sf probe 0; sf read 0x80008000 0x100000 0x55000; go 0x80008000';sa。
- 0x55000 这个值就是编译完成时提示的长度值。如果代码有改动，这个长度可能会变动，为了避免频繁修改 bootcmd，可以把读取的长度设置为略长于 0x55000，例如 0x60000，读取长度多一些没有问题，如果读取长度不够，则系统无法正常运行。

步骤 5. 烧录完成后，对板子重新上电，如果可以正常启动，则说明分散加载过程正常。

----结束

海思专有和保密信息
版权所有 © 深圳市海思半导体有限公司

3 注意事项

3.1 APP 代码修改

涉及的代码文件只有一个，就是 sample_scatter.c。主体内容如图 3-1 所示。

图3-1 sample_scatter.c 主要代码

```
void app_init(void)
{
    misc_driver_init();
    //TODO:
    //User can add code here. This part is loaded in  uboot.
#ifdef CFG_SCATTER_FLAG
    LOS_ScatterLoad(0x100000, image_read, 1);//"0x100000" is the
#endif
#ifndef CFG_FAST_IMAGE
    //TODO:
    //User can add code here. This part is loaded in  liteos.
    //Now I use the shell as an example.
    osShellInit();
    shell_cmd_register();

    dprintf("------------------start finish-----------------
#endif

    while (1)
    {
        sleep(1);
    }
} ? end app init ?
```

需要在 uboot 里加载的代码，放在 LOS_ScatterLoad 函数前面调用。需要在 APP 里加载的代码，放在 LOS_ScatterLoad 函数后面的 CFG_FAST_IMAGE 条件里。

LOS_ScatterLoad 的函数原型：

```
VOID LOS_ScatterLoad(UINT32 uwImageFlashAddr, FLASH_READ_FUNC
pfFlashReadFunc, UINT32 uwReadAlignSize);
```


海思专有和保密信息
版权所有 © 深圳市海思半导体有限公司

- uwImageFlashAddr：APP 镜像烧录的 Flash 地址。
- pfFlashReadFunc：APP 中读取 Flash 的函数。
- uwReadAlignSize：Flash 读取的最小单位。

pfFlashReadFunc 在 Sample 里的实现如图 3-2 所示。如果用户的电路板使用其他类型的 Flash，则需要根据实际情况修改这个函数的实现。

图3-2 Sample 里 pfFlashReadFunc 的实现

```
static int image_read(void* memaddr, unsigned long start, unsigned long size)
{
    return hispinor_read(memaddr, start, size);
}
```

3.2 开发效率提升

如前文所述，make liteos_image 过程中，执行了 2 次 make，且都先进行了 make clean，在代码较多的时候，编译过程非常耗时。所以在不调试分散加载功能的时候，可以通过只输入 make 命令来进行编译。这种机制是通过 Makefile 里的变量来控制的。

海思专有和保密信息
版权所有 © 深圳市海思半导体有限公司