Hi35xx Huawei LiteOS 开发环境

用户指南

文档版本 **00B06**

发布日期 **2017-04-28**

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

本文档介绍 Huawei LiteOS 开发环境。Huawei LiteOS 开发环境的搭建、U-boot、Huawei LiteOS、文件系统，以及如何在 Huawei LiteOS 下进行开发。

本文档主要提供让客户更快地了解 Huawei LiteOS 开发环境指导。

说明
本文描述以 Hi3516AV100 为例。未有特殊说明，Hi3519V100/Hi3519V101、Hi3518EV20X、Hi3516CV200、Hi3516CV300 与 Hi3516AV100 一样。

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
Hi3516C	V300

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。

修订日期	版本	修订说明
2017-04-28	00B06	3.2 和 4.1 小节涉及修改
2017-01-20	00B05	1.3.1 和 3.3 小节涉及修改
2016-08-30	00B04	添加 Hi3516CV300 的相关内容
2016-06-20	00B03	第 3 次临时版本发布 添加 Hi3519V101 相关内容
2016-05-13	00B02	第 2 次临时版本发布 添加 Hi3519V100 相关内容
2015-12-10	00B01	第 1 次临时版本

目 录

插图目录

表格目录

1 开发环境

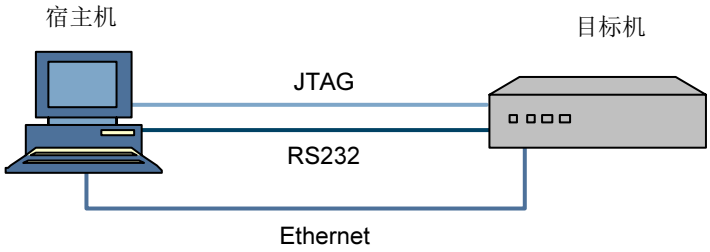
1.1 嵌入式开发环境

由于嵌入式单板的资源有限，不能在单板上运行开发和调试工具，通常需要交叉编译调试的方式进行开发和调试，即“宿主机＋目标机（评估板）”的形式。宿主机和目标机一般采用串口连接，也可同时通过网口或者 JTAG 连接，如图 1-1 所示。

宿主机和目标机的处理器一般不相同。宿主机需要建立适合于目标机的交叉编译环境。程序在宿主机上经过“编译－连接－定位”得到可执行文件。通过一定的方法将可执行文件烧写到目标机中，然后在目标机上运行。

目标机上的 Bootloader 启动后，目标机中的操作信息通过串口或者网口输出到宿主机上显示。在宿主机上的控制台中输入命令，可以控制目标机。

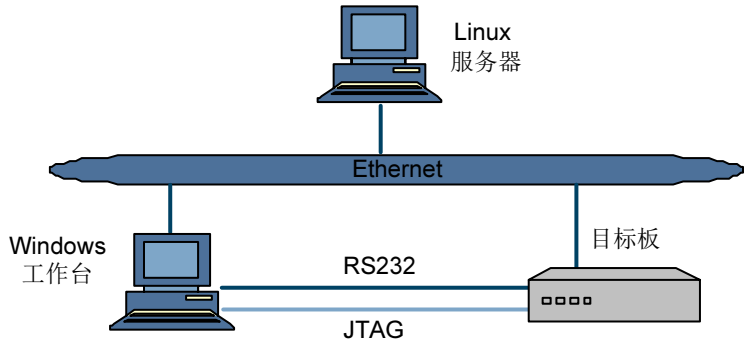
图1-1 嵌入式开发图例



1.2 Hi35xx Huawei LiteOS 开发环境

Hi35xx Huawei LiteOS 开发环境通常包括 linux 服务器、Windows 工作台和 Hi35xx DMEB（目标板），三者同处于一个网络中，如图 1-2 所示。

图1-2 Hi35xx Huawei LiteOS 开发环境



在 linux 服务器上建立交叉编译环境，Windows 工作台通过串口和网口与 Hi35xx 单板连接，开发人员可以在 Windows 工作台中进行程序开发或者远程登录到 linux 服务器进行程序开发。各部分具体软件介绍如表 1-1 所示。

说明

开发环境中使用了 Windows 工作台，实际上很多工作也可以在 Linux 服务器上完成，如使用 minicom 代替超级终端等，用户可自行选择。

表1-1 Hi35xx Huawei LiteOS 开发环境的各部分软件描述

软件		描述
Windows 工作台	操作系统	Windows 98/me/2000/XP/Windows7
	应用软件	putty、tftp 服务器、超级终端、DS-5 等软件。
Linux 服务器	操作系统	无特别要求，可为 Ubuntu、Redhat、Debian 等。内核版本支持 2.6.18 及以上版本。安装时建议选择完全安装。
	应用软件	arm 交叉编译环境（Gcc 版本 4.8.3），编译工具链，Vim、mkyaffs2image、mkfs.jffs2 等。
Hi35xx	引导程序	U-boot
	操作系统	Huawei LiteOS
	文件系统	yaffs2、fat32、jffs2 等。

1.3 搭建 Huawei LiteOS 开发环境

以下介绍 Huawei LiteOS 开发环境中 Linux 服务器、Windows 工作台与目标板三者开发环境的搭建与各自环境中开发工具的使用。

1.3.1 Linux 服务器环境搭建

Linux 服务器主要用于编译 Huawei LiteOS 代码。

- 服务器操作系统

  Linux 服务器建议选择常用的 Linux 发行版，便于寻找各类技术资源。例如：

  a. RedHat 较新的发行版如 RedHat Fedora Core 系列和 Redhat Enterprise Linux、Red Hat 3.4.4-2。

  b. RedHat 较老的发行版如 RedHat 9.0 等。

  推荐使用较新版本，以方便获取各类资源，如 Fedora Core 系列、Ubuntu10。

- 安装 Huawei LiteOS 编译工具链

  arm-hisiv500-linux 为发布包提供的编译工具链，用于编译、链接、处理和调试跨平台体系结构的程序代码。安装步骤如下：

  步骤 1. 解压工具链。在 Huawei LiteOS 发布包中，解压后会在 toolchain 目录下得到 arm-hisiv500-linux.tar.bz2 和 arm-hisiv500-linux.install 两个文件，其中 arm-hisiv500-linux.tar.bz2 是安装包，arm-hisiv500-linux.install 是 x86_64-linux-gnutar 是服务器上的安装脚本。

  步骤 2. 安装方法：

  a. 将 arm-hisiv500-linux.tar.bz2 和 arm-hisiv500-linux.install 放在同一级目录下；

  b. ./ arm-hisiv500-linux.install

  ----结束

执行完以上命令即完成安装。

说明

使用从网络等渠道得到的交叉编译工具可能存在与使用的内核并不配套，造成开发过程中出现一些不可预料的问题。

- 开发软件

在 Linux 服务器端，可使用一些工具进行有效率的开发。

a. Vim 文本编辑器。使用 Vim+ctags+cscope 工具可方便地对代码文件进行快速修改与编写。

b. mkyaffs2image。制作 yaffs2 文件系统镜像，相关使用参考第四节。

c. mkfs.jffs2。用于制作 jffs2 文件系统镜像，相关使用参考第四节。

1.3.2 Windows 工作台环境搭建

Window 工作台主要用于建立与目标板的硬件连接。同时远程到服务器上进行开发操作。

1.3.2.1 与目标板的连接

Windows 工作台工作环境需要与目标板的串口与网络端口分别进行连接。串口连接保证开发者能实时查看目标板提供的终端打印信息。而目标板的程序升级则需要通过网络端口快速下载。下面介绍两类连接需要用到的软件工具：

- 串口终端。Windows 下可提供的串口终端有很多，用户可用 windows 自带的超级终端工具，也可使用第三方软件如 SecureCRT、IPOP 等。
- tftp（Trivial File Transfer Protocol）服务器工具。Windows 平台可使用 Tftpd32 工具搭建 tftp 服务器。目标板在烧写过程需要从该服务器中获得烧写对象（关于目标板的程序更新，请参考 2 “U-boot”）。

1.3.2.2 与服务器的连接

由于代码在 linux 服务器上编译。Windows 工作台另一方面也须与服务器端建立网络连接，并使用相关工具远程连接到服务器中进行代码编辑、代码编译、交换文件等操作。常用工具有 putty,WinSCP 等。

1.3.3 目标板开发环境搭建

目标板环境包括 U-boot，Huawei LiteOS 与文件系统三大块，将在后面篇幅详细介绍。

2 U-boot

关于 U-boot 的介绍与使用请参见《Hi35xx U-boot 移植应用 开发指南》。

3 Huawei LiteOS

3.1 Huawei LiteOS 发布包

Huawei LiteOS 发布包中包含了 Huawei LiteOS 源代码，U-boot 源代码，tools 工具集等。对发布包进行解压后，可以看到以下文件目录架构：

图3-1 发布包文件目录架构

```
├── Makefile
├── opensource
├── pub
├── Readme.txt
├── toolchain
└── tools
```

以上目录解释如下：

- Makefile：用于配置编译镜像与文件系统制作工具的规则文件。
- opensource：存放 U-boot，Huawei LiteOS 源码。
- pub：存放镜像工具与输出镜像。
- toolchain：存放编译工具链。
- tools：存放其他工具。

3.2 Huawei LiteOS 源代码

Huawei LiteOS 源代码存放于目录 opensource 下。用户可直接进入目录进行相关操作。下图为源代码顶层目录结构：

图3-2 Huawei LiteOS 源代码顶层目录结构

```
liteos
├──── build
├──── compat
├──── .config
├──── config.mk
├──── config_xx.mk
├──── doc
├──── drivers
├──── fs
├──── kernel
├──── lib
├──── liteos.ld
├──── Makefile
├──── net
├──── NUTTX_COPYING
├──── out
├──── platform
├──── ReleaseNotes
├──── sample
├──── shell
└──── tools
```

对各个目录或文件的解释如下：

- build：编译选项与配置。
- compat：操作系统标准兼容接口，包含 CMSIS、Posix、Linux。
- config：默认配置文件。
- config.mk：系统级的头文件、系统编译宏和编译参数配置文件。
- config_xx.mk：芯片平台编译参数配置文件。
- doc: 系统描述相关文档路径。
- drivers：驱动文件目录。
- fs：文件系统目录，包含虚拟文件系统 VFS，支持 fat32、yaffs2、jffs2。
- lib：包含了标准函数 libm、libc 库。
- liteos.ld：链接脚本
- Makefile：顶层 Makefile
- net：网络相关模块代码。
- NUTTX_COPYING：nuttx license 声明。
- platform：包含了硬件平台相关代码，例如异常、硬中断等。
- ReleaseNotes：版本描述、修订记录文件。
- sample：包含了编程示例。
- shell：包含了 shell 命令调试框架。
- kernel：Huawei LiteOS 源码。
- tools：系统工具。

3.3 Huawei LiteOS 编译

Huawei LiteOS 的开发，编译工作首要任务是安装编译工具链。具体安装方法可以参考本文档 1.3.1 Linux 服务器环境搭建。

- 统一编译。用户需要在开发包解压根目录下输入：“make”。make 工具将会根据当前 Makefile 设置编译进行统一编译，统一编译结果将在 pub 目录下得到两个目录 boot 与 tools。两者分别保存 U-boot 镜像与文件系统镜像制作工具。由统一编译生成的 U-boot 镜像可直接用于烧写到单板，文件系统镜像工具可用于制作 yaffs 与 jffs2 文件系统镜像，具体使用可参考本文档 4.2.3 挂载文件系统镜像。
- U-boot 单独编译。在 linux 服务器环境下需要逐步完成以下步骤：

a. 在发布包根目录下，执行 cd opensource/u-boot/。

b. 解压源码，tar -xf u-boot-2010.06.tgz。

c. 切换目录，cd u-boot-2010.06。

d. 更新配置文件（命令需要根据芯片系列不同相应调整）：

```
make ARCH=arm CROSS_COMPILE=arm-hisiv500-linux- hi3516a_config
```

e. 编译：

```
make ARCH=arm CROSS_COMPILE=arm-hisiv500-linux-
```

f. 制作可烧写 U-boot 文件：

```
cp ../../../tools/pc/uboot_tools/mkboot.sh
```

复制表格文件（文件名根据芯片系列不同需要调整）：

```
cp ../../../tools/pc/uboot_tools/reg_info_hi3516a.bin
./mkboot.sh reg_info_hi3516a.bin u-boot-final.bin
```

完成以上步骤之后，当前目录下会生成 u-boot-final.bin 文件，u-boot-final.bin 即为可用的 u-boot 镜像。

说明

- 更新配置文件步骤，若使用其它芯片平台，则配置文件名称需要更改为所用芯片，例：

  ```
  Hi3518EV200: make ARCH=arm CROSS_COMPILE=arm-hisiv500-linux-
  hi3518ev200_config
  Hi3519: make ARCH=arm CROSS_COMPILE=arm-hisiv500-linux-
  hi3519_config
  ```

  其它芯片平台同理。

- 制作可烧写 U-boot 文件步骤，若使用其它芯片平台，则配置文件名称需要指定表格生成的配置文件名称。例：

  ```
  Hi3518EV200:
  cp ../../../tools/pc/uboot_tools/reg_info_hi3518ev200.bin
  Hi3519:
  cp ../../../tools/pc/uboot_tools/reg_info.bin
  Hi3516CV300:
  cp ../../../tools/pc/uboot_tools/reg_info.bin
  ```

4 文件系统

4.1 文件系统简介

当前 Huawei LiteOS 版本暂仅支持 yaffs2、jffs2、fat32 与 exFAT 文件系统。后续版本将会添加更多文件系统支持。

- yaffs2 是专门为 NAND Flash 设计的嵌入式文件系统。它是日志结构的文件系统，提供了损耗平衡和掉电保护，可以有效地避免意外掉电对文件系统一致性和完整性的影响。yaffs2 的优缺点如下：

a) 优点

- 专门针对 NAND Flash，软件结构得到优化，速度快。
- 使用硬件的 spare area 区域存储文件组织信息，启动时只需扫描组织信息，启动比较快。
- 采用多策略垃圾回收算法，能够提高垃圾回收的效率和公平性，达到损耗平衡的目的。

b) 缺点

没有采用压缩的文件格式。

说明

Hi3518V200/Hi3518V201/Hi3516CV200/Hi3516CV300 不支持 NAND Flash。

- JFFS2 是 RedHat 的 David Woodhouse 在 JFFS 基础上改进的文件系统，是用于微型嵌入式设备的原始闪存芯片的实际文件系统。JFFS2 文件系统是日志结构化的可读写的文件系统。JFFS2 的优缺点如下：

a) 优点

使用了压缩的文件格式。最重要的特性是可读写操作。

b) 缺点

JFFS2 文件系统挂载时需要扫描整个 JFFS2 文件系统，因此当 JFFS2 文件系统分区增大时，挂载时间也会相应的变长。使用 JFFS2 格式可能带来少量的 Flash 空间的浪费。这主要是由于日志文件的过度开销和用于回收系统的无用存储单元，浪费的空间大小大致是若干个数据段。JFFS2 的另一缺点是当文件系统已满或接近满时，JFFS2 运行速度会迅速降低。这是因为垃圾收集的问题。

- FAT32 起源于 70 年代末 80 年代初，用于微软的 MS-DOS 操作系统。现被广泛应用于 PC 机与各类存储介质如 U 盘、存储卡等。嵌入式系统对 fat32 文件系统的支

持，主要应用于对外设存储设备的挂载访问。exFAT 称为扩展 FAT，可支持 4G 及其更大文件。

a) 优点
   - 被广泛应用于 U 盘，存储卡等便捷性存储介质。

b) 缺点
   - 对嵌入式存储介质支持优势不明显。

4.2 添加文件系统支持到 Huawei LiteOS

Huawei LiteOS 暂支持 yaffs2、jffs2 与 fat32，分别应用于 Nand flash、Nor flash 与外设存储挂载。本章节分类说明 yaffs 与 jffs2 文件系统在 Huawei LiteOS 中的应用。以下是对应的操作步骤。

4.2.1 Flash 初始化

文件系统应用于存储介质上，因此，首要工作是需要对 flash 进行初始化。具体操作是在系统初始化过程中添加对 flash 控制器的初始化函数（开发者可参考 sample/sample.c 文件中该函数的使用）。

```
nand_init();  // nand flash 控制器初始化函数
spinor_init(); // spi flash 控制器初始化函数
```

成功调用函数后，即完成对应 flash 控制器的初始化。

4.2.2 分区与挂载

a. 分区，flash 控制器初始化后，即可对其进行分区操作。目前 Huawei LiteOS 版本内核提供以下函数用于文件系统分区。

```
int add_mtd_partition( char *type, uint32_t start_addr, uint32_t
length, uint32_t partition_num)
[type]:存储介质类型，指定字符串为"nand"与"spinor"。分别代表两类存储介质。
[start_addr]:分区操作flash的物理开始地址。
[length]:分区操作Nand flash的分区大小。
[partition_num]:分区盘符。
eg:
add_mtd_partition("nand", 0x900000, 16*0x100000, 0);
add_mtd_partition("spinor", 0x100000, 2*0x100000, 0);
```

以上例子操作，分别操作 nand 与 spinor 类型 flash，前者以 0x900000 为起始地址，划分大小为 16M 的 nand 分区作为 yaffs 分区，后者划分了以 0x100000 为起始地址，划分大小为 2M 的 jffs 分区。

说明

分区函数调用成功后，会于/dev 目录下生成对应挂载结点，用户可通过该结点把对应分区挂载文件系统。

b. 挂载。在完成了 Nand flash 的分区操作后，便可对该 Nand 分区进行挂载。内核提供 mount 函数来挂载完成已完成分区操作的设备节点。

```
int mount(FAR const char *source, FAR const char *target, FAR const
char *filesystemtype, unsigned long mountflags,
FAR const void *data)
```

[source]:挂载源设备。参数意义指需要挂载的设备结点。

[target]:挂载点目录。

[filesystemtype]:挂载文件系统类型。

[mountflags]:挂载标志。在此例挂载暂时无义，可传入0参数。后续版本会进行扩展。

[data]:挂载追加参数。此例挂载暂时无义。可传入参数0。

```
eg:
mount("/dev/nandblk0", "/yaffs0", "yaffs", 0, NULL);
mount("/dev/spinorblk0", "/jffs0", "jffs", 0, NULL);
```

以上例子，对已分区的 flash 分别挂载到指定目录中。

4.2.3 挂载文件系统镜像

用户可以挂载自己的文件系统镜像，Huawei LiteOS 没有像 Linux 相似的根文件系统，文件系统的目录结构请根据具体情况自行制作。以下说明文件系统制作与挂载步骤。

a. yaffs 文件系统镜像可由 mkyaffs2image 工具制作。使用时，需要根据器件的 pagesize 与 ecctype 类型确定传入参数。

```
eg:
 ./mkyaffs2image610 rootfs_jffs rootfs_yaffs.yaffs 1 2
```

该命令是把名为 rootfs_jffs 的文件夹制作成名为 rootfs_yaffs.yaffs 的文件系统镜像，适配 nand 器件 pagesize=2k，ecctype=4bit/512 ecc。用户可以通过不输入任何参数运行工具以查看工具支持的帮助信息。

b. jffs2 文件系统由工具 mkfs.jffs2 制作。

```
eg:
 ./mkfs.jffs2 -d rootfs_jffs -l -e 0x40000 -o rootfs_jffs.jffs2
```

参数说明：

-d：需要制作成文件系统的目录。

-e：指定擦除块大小。

-l：指定为小端格式。

-o：指定生成镜像名称。

用户可直接运行工具查看工具帮助说明。上述命令把名为 rootfs_jffs 的文件夹以 0x40000 为擦除块大小，小端格式制作成 jffs2 文件系统。

文件系统镜像制作好后，需要把文件系统烧写到对应的分区起始地址中。系统启动后，便可以访问制作的文件系统内容。

说明

Huawei LiteOS 支持 fat32/exFAT 格式，开发者可参考文档《》Huawei LiteOS 外围设备驱动操作指南》中“SD/MMC 卡操作指南”了解关于 fat32/exFAT 格式在 Huawei LiteOS 中的使用示例。

5 程序开发简介

5.1 编写代码

用户可根据个人习惯选择代码编写工具。通常在 Windows 环境下使用 Source Insight，在 Linux 环境下使用 Vim+ctags+cscope，功能也相当强大。

5.2 镜像烧写

要在 Huawei LiteOS 平台上运行应用程序，必须完成以下工作：

- 编译。请使用随 Huawei LiteOS 系统发布的工具链，使用其它工具链或会出现不兼容等错误。相关编译方法可参考本文档 Huawei LiteOS 编译。
- 烧写 U-boot。单板首次烧写需要烧写 U-boot 启动镜像。首次烧写 U-boot 需软件工具 HiTool 烧写，具体方法可参考 HiTools 帮助手册。这里简单说明单板已存在 U-boot 并且启动单板后顺得进入 U-boot 命令行后的 U-boot 镜像烧写方法。

a. 切换目录至代码包根目录，把./pub/boot/U-boot-hi3516a.bin 拷贝至 FTP 服务器。

b. 启动单板进入 U-boot 命令行模式,根据 FTP 服务器设置更新网络相关环境变量。

c. 在单板命令行执行：`tftp 0x82000000 U-boot-final.bin`

d. 擦除 flash 信息（以 nand 为例）：`nand erase 0x0 0x80000`

e. 烧写到 flash（以 nand 为例）：`nand write 0x82000000 0x0 0x80000`

- Huawei LiteOS 镜像烧写。

a. 切换目录至代码包根目录，把./opensource/Liteos/Liteos_ipc/out/sample.bin 拷贝至 FTP 服务器。

b. 启动单板进入 U-boot 命令行模式,根据 FTP 服务器设置更新网络相关环境变量。

c. 在单板命令行执行：`tftp 0x82000000 sample.bin`

d. 擦除 flash 信息（以 nand 为例）：`nand erase 0x100000 0x700000`

e. 烧写到 flash（以 nand 为例）：`nand write 0x82000000 0x100000 0x700000`

设置 U-boot 环境变量为：（注意：0x80008000 为 OS 固定的启动地址，不能修改）

```
setenv bootcmd 'nand read 0x80008000 0x100000 0x700000; go
0x80008000';saveenv
```

A 缩略语

A

ARM	Advanced RISC Machine	ARM 公司指令集

C

CRAMFS	Compressed RAM file system	压缩 RAM 文件系统

D

DMS	Digital Media Solution	媒体解决方案平台

E

ELF	Executable and Linkable Format	可执行连接格式文件

G

GCC	GNU Complier Collection	GNU 编译器集合
GNU	GNU’s Not UNIX	GNU

I

IP	Internet Porotocol	Internet 协议

J

JFFS2	Journalling FLASH File System v2	一种 Flash 文件系统
JTAG	Joint Test Action Group	联合测试行动组

P

PC	Personal Computer	个人计算机

S

SDRAM	Synchronous Dynamic Random Access Memory	同步动态随机存储器
SDK	Software Development Kit	软件开发工具集

Y

YAFFS2	Yet Another Flash File System	一直专门针对 NAND 闪存设计的文件系统