



Hi3559A/C V100 SDK 安装及升级使用说明

文档版本 00B02

发布日期 2018-02-10

版权所有 © 深圳市海思半导体有限公司 2018。保留一切权利。

未经本公司书面许可，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

本文为 Hi3559A/C V100 SDK 的安装及升级使用说明，方便使用者能快速在 Hi3559A V100 或 Hi3559C V100 DEMB 板上搭建好 SDK 运行环境。



说明

本文以 Hi3559AV100 为例，未有特殊说明，Hi3559CV100 与 Hi3559AV100 内容一致。

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
Hi3559A	V100
Hi3559C	V100

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。

文档版本 00B02 (2018-02-10)

1.6 和 2.2.1 小节涉及修改。



添加 2.2.2 小节。

文档版本 00B01 (2018-01-15)

第 1 次临时版本发布。



目 录

前 言.....	i
1 首次安装 SDK.....	1
1.1 Hi3559AV100 SDK 包位置.....	1
1.2 解压缩 SDK 包.....	1
1.3 展开 SDK 包内容.....	1
1.4 在 linux 服务器上安装交叉编译器.....	1
1.5 编译 osdrv.....	2
1.6 SDK 目录介绍.....	2
2 安装、升级 Hi3559AV100 DEMO 板开发环境.....	5
2.1 准备工作.....	5
2.2 操作步骤.....	5
2.2.1 单系统 Linux 方案烧写步骤.....	
2.2.2 双系统 Linux+Huawei LiteOS 方案烧写步骤.....	7
3 开发前环境准备.....	10
3.1 管脚复用.....	10
3.2 连接串口.....	10
3.3 NFS 环境.....	10
4 使用 SDK 和 DEMO 板进行开发.....	11
4.1 开启 Linux 下的网络.....	11
4.2 使用 NFS 文件系统进行开发.....	11
4.3 开启 telnet 服务.....	11
4.4 运行 MPP 业务.....	12
5 地址空间分配与使用.....	13
5.1 DDR 内存管理说明.....	13
5.2 DDR 内存分配.....	13



1 首次安装 SDK

如果您已安装过 SDK，可以直接参看 2 “[安装、升级 Hi3559AV100 DEMO 板开发环境](#)”。

1.1 Hi3559AV100 SDK 包位置

在"Hi3559AV100***/01.software/board"目录下，您可以看到一个 Hi3559AV100_SDK_Vx.x.x.x.tgz 的文件，该文件就是 Hi3559AV100 的软件开发包。

1.2 解压缩 SDK 包

在 linux 服务器上（或者一台装有 linux 的 PC 上，主流的 linux 发行版本均可以），使用命令：`tar -zxf Hi3559AV100_SDK_Vx.x.x.x.tgz`，解压缩该文件，可以得到一个 Hi3559AV100_SDK_Vx.x.x.x 目录。

1.3 展开 SDK 包内容

返回 Hi3559AV100_SDK_Vx.x.x.x 目录，运行 `./sdk.unpack` (请用 root 或 sudo 权限执行) 将会展开 SDK 包打包压缩存放的内容，请按照提示完成操作。

如果您需要通过 WINDOWS 操作系统中转拷贝 SDK 包，请先运行 `./sdk.cleanup`，收起 SDK 包的内容，拷贝到新的目录后再展开。

1.4 在 linux 服务器上安装交叉编译器

在发布包 Hi3559A V100R001C02SPCxxx.rar 所在的目录中下载工具链文件。

注意：安装交叉编译器需要有 sudo 权限或者 root 权限。

1) 安装 aarch64 交叉编译器：



解压 `tar -xzf aarch64-himix100-linux.tgz`，运行 `chmod +x aarch64-himix100-linux.install`，然后运行 `./aarch64-himix100-linux.install` 即可。

2) 安装 arm-none-eabi 交叉编译器：

解压 `tar -xzf gcc-arm-none-eabi-4_9-2015q3.tgz`，运行 `chmod +x gcc-arm-none-eabi-4_9-2015q3.install`，然后运行 `./gcc-arm-none-eabi-4_9-2015q3.install` 即可。

3) 执行 `source /etc/profile`，安装交叉编译器的脚本配置的环境变量就可以生效了，或者请重新登陆也可。

1.5 编译 osdrv

参见 osdrv 目录下 readme。

1.6 SDK 目录介绍

Hi3559AV100_SDK_Vx.x.x.x 目录结构如下：

```
-- drv                                # drv 目录
|   |-- extdrv                        # 板级外围驱动源代码
|   |-- interdrv                     # 片内其它模块 mipi,cipher 等驱动源代码
|-- image_glibc_multi-core_arm64     # 单 Linux 方案的烧写镜像文件
|-- image_glibc_big-little_arm64     # Linux+Huawei LiteOS 双系统方案的烧写镜像文件
|-- mpp                               # 存放单核媒体处理平台的目录
|   |-- component                    # mpp 组件
|   |   |-- isp                     # isp firmware 源代码
|   |   |-- gpu                     # gpu 驱动源代码
|   |   |-- pci                     # pciv 驱动源代码
|   |-- out                          # 发布文件
|   |   |-- linux                   # linux 发布文件
|   |   |   |-- multi-core          # 在 A73MP 单 Linux 方案上运行的驱动文件
|   |   |   |   |-- init            # 内核模块的初始化源代码
|   |   |   |   |-- obj            # 内核模块的 obj 文件
|   |   |   |   |-- include        # 头文件
|   |   |   |   |-- ko             # 内核 ko 模块
|   |   |   |   |-- lib            # 用户态 lib 库
```



-- big-little	# 在 Linux+LiteOS 双系统方案中 A73MP Linux 上运行的驱动文件
-- init	# 内核模块的初始化源代码
-- obj	# 内核模块的 obj 文件
-- include	# 头文件
-- ko	# 内核 ko 模块
-- lib	# 用户态 lib 库
-- liteos	# 内核模块的 obj 文件
-- single	# 在 Linux+Huawei LiteOS 双系统方案中 A53UP 的 Huawei LiteOS 上运行的驱动文件
-- dsp	# 在 DSP 的 Huawei LiteOS 上运行的驱动文件
-- sample	# 各种业务的样例源代码
-- tools	# 媒体处理相关工具
-- cfg.mak	# mpp 配置文件
-- Makefile.param	# mpp 全局编译选项
-- Makefile.linux.param	# mpp linux 编译选项
-- Makefile.liteos.param	# mpp liteos 编译选项
-- osdrv	# 存放操作系统及相关驱动的目录
-- components	# 组件源代码
-- ipcm	# 核间通信驱动源码
-- pcie_mcc	# pcie_mcc 驱动源码
-- opensource	# opensource 源代码
-- arm-trusted-firmware	# ATF 源代码
-- busybox	# busybox 源代码
-- kernel	# linux 内核的 patch
-- uboot	# uboot 源代码
-- platform	# 平台文件
-- liteos_a53	# A53UP 上运行的 Huawei LiteOS 操作系统
-- liteos_m7	# M7 上运行的 Huawei LiteOS 操作系统 工具、
-- pub	# 编译好的镜像、drv 驱动等
-- tools	# 工具源代码
-- readme_cn.txt	# osdrv 中文使用说明



	-- readme_en.txt	# osdrv 英文使用说明
	-- Makefile	# osdrv Makefile
--	osal	# 存放操作系统适配层的头文件和源文件的目录
	-- include	# 存放操作系统适配层的头文件的目录
	-- linux	# 存放 linux 系统适配层的源文件的目录
	-- liteos	# 存放 Huawei LiteOS 系统适配层的源文件的目录
--	package	# 存放 SDK 各种压缩包的目录
	-- drv.tgz	# drv 压缩包
	-- mpp.tgz	# 媒体处理平台软件压缩包
	-- osal.tgz	# 操作系统适配层源码压缩包
	-- hisyslink.tgz	# 核间通信模块压缩包
	-- osdrv.tgz	# linux 内核/uboot/rootfs/tools 源码压缩包
--	scripts	# 存放 shell 脚本的目录
--	sdk.cleanup	# SDK 清理脚本
--	sdk.unpack	# SDK 展开脚本



2 安装、升级 Hi3559AV100 DEMO 板开发环境

本章节以 Hi3559AV100 DEMO 板为例，介绍烧写 u-boot、内核以及文件系统的方法。

2.1 准备工作

首先，请阅读文档《Hi3559AV100 Demo 单板用户指南》或《Hi3559CV100 Demo 单板使用指南》，了解 Hi3559AV100 DEMO 板硬件的功能、结构、接口等信息。

然后，请阅读文档《多核 使用指南》，了解 Hi3559AV100 上的多核分布、业务部署、调试方法等信息。

可以按照以下操作来烧写 u-boot、内核以及文件系统，以下操作均使用网络来更新。

1. 如果您拿到的单板没有 u-boot，就需要使用 HiTool 工具进行烧写。HiTool 工具位置放在 Hi3559A***/01.software/pc/HiTool，使用说明请参见该目录下的《HiBurn 工具使用指南》。
2. 如果您拿到的单板中已经有 u-boot，可以按照以下步骤使用网口烧写 u-boot、kernel 及 rootfs 到 Flash 中。

本章所有的烧写操作都在 DEMO 板上的串口座上进行。烧写到 SPI NAND Flash 上。

2.2 操作步骤

2.2.1 单系统 Linux 方案烧写步骤

步骤 1. 配置 tftp 服务器

可以使用任意的 tftp 服务器，将发布包 image_glibc_multi-core_arm64 目录下的相关文件拷贝到 tftp 服务器目录下。

步骤 2. 参数配置



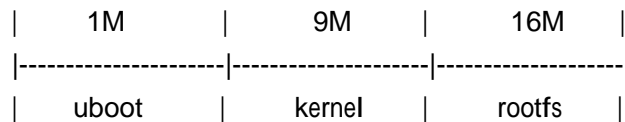
单板上电后，敲任意键进入 u-boot。设置 ipaddr (单板 ip)、ethaddr (单板的 MAC 地址) 和 serverip (即 tftp 服务器的 ip)。

```
setenv ipaddr 10.67.208.170
setenv ethaddr 00:10:ab:20:81:70
setenv netmask 255.255.254.0
setenv gatewayip 10.67.208.1
setenv serverip 10.67.209.239
ping serverip · 确保网络畅通。
以上为举例，IP 以实际为准。
```

步骤 3. 烧写 multi-core 版本映像文件到 SPI NAND

注意：单 Linux 方案要烧写 image_glibc_multi-core_arm64 目录中的镜像文件！

1. 地址空间说明



以下操作基于图示的地址空间分配，也可以根据实际情况进行调整。

2. 拨码选择主 CPU

通过拨码开关 SW1.4 设置选择主 CPU：

0：从 A53MP Core0 启动；

1：从 A53UP 启动。

3. 烧写 u-boot

```
mw.b 0x44000000 0xff 0x100000
tftp 0x44000000 u-boot-hi3559av100.bin
nand erase 0x0 0x100000
nand write 0x44000000 0x0 0x100000
```

4. 烧写内核

```
mw.b 0x44000000 0xff 0x900000
tftp 0x44000000 ulmage_hi3559av100_multi-core
nand erase 0x100000 0x900000
nand write 0x44000000 0x100000 0x900000
```

5. 烧写文件系统

```
mw.b 0x44000000 0xff 0x1000000
tftp 0x44000000 rootfs_hi3559av100_2k_24bit.yaffs2
```



```
nand erase 0xA00000 0x1000000
```

```
nand write.yaffs 0x44000000 0xA00000 0xcfeb00 ( 0xcfeb00 为 rootfs 文件实际大小 )
```

6. 设置启动参数

```
setenv bootargs 'mem=512M console=ttyAMA0,115200 root=/dev/mtdblock2 rw rootfstype=yaffs2 mtdparts=hinand:1M(boot),9M(kernel),16M(rootfs)'
```

```
setenv bootcmd 'nand read 0x44000000 0x100000 0x900000;bootm 0x44000000'
```

```
saveenv
```

7. 重启系统

```
reset
```

----结束

2.2.2 双系统 Linux+Huawei LiteOS 方案烧写步骤

步骤 1. 准备 Huawei LiteOS 端的镜像

Huawei LiteOS 端的烧写镜像是由客户在 Huawei LiteOS 端的业务代码编译生成的。如果是初次调试环境，可以用发布包中的 sample 编译出来的镜像，A53 端的镜像建议使用 mpp/sample/vio/sample_vio.bin；M7 端的镜像建议使用 osdrv/platform/liteos_m7/liteos/sample/sample_osdrv sample.bin。

步骤 2. 配置 tftp 服务器

可以使用任意的 tftp 服务器，将发布包 image_glibc_big-little_arm64 目录下的相关文件拷贝到 tftp 服务器目录下。将上面步骤 1 准备的 Huawei LiteOS 镜像也拷贝到 tftp 服务器目录下。

步骤 3. 参数配置

单板上电后，敲任意键进入 u-boot。设置 ipaddr (单板 ip)、ethaddr (单板的 MAC 地址) 和 serverip (即 tftp 服务器的 ip)。

```
setenv ipaddr 10.67.208.170
```

```
setenv ethaddr 00:10:ab:20:81:70
```

```
setenv netmask 255.255.254.0
```

```
setenv gatewayip 10.67.208.1
```

```
setenv serverip 10.67.209.239
```

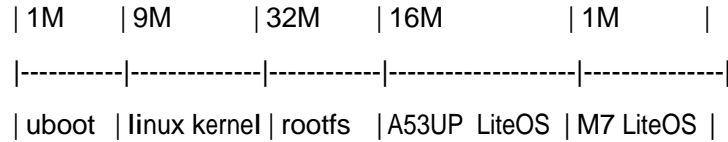
ping serverip，确保网络畅通。

以上为举例，IP 以实际为准。

步骤 4. 烧写 single+big.little 版本映像文件到 SPI NAND

注意：双系统方案要烧写 image_glibc_big-little_arm64 目录中的镜像文件！

1. 地址空间说明



以下操作基于图示的地址空间分配，也可以根据实际情况进行调整。

2. 拨码选择主 CPU

通过拨码开关 SW1.4 设置选择主 CPU：

0：从 A53MP Core0 启动；

1：从 A53UP 启动。

需要拨码到 0，从 A53MP 启动。

3. 烧写 u-boot

```
mw.b 0x52000000 0xff 0x100000
tftp 0x52000000 u-boot-hi3559av100.bin
nand erase 0 0x100000
nand write 0x52000000 0 0x100000
```

4. 烧写 Linux 内核

```
mw.b 0x52000000 0xff 0x1000000
tftp 0x52000000 ulmage_hi3559av100_big-little
nand erase 0x100000 0x900000
nand write 0x52000000 0x100000 0x900000
```

5. 烧写文件系统

```
mw.b 0x52000000 0xff 0x2000000
tftp 0x52000000 rootfs_hi3559av100_2k_24bit.yaffs2
(注：此示例 Flash 为 2KPage 24bitECC，应以实际使用器件选择相应 rootfs)

nand erase 0xA00000 0x2000000
nand write.yaffs 0x52000000 0xA00000 0xd32c00
(注：nand write.yaffs 的最后一个参数(size)，应该以实际下载的 rootfs 长度为准
如打印：“Bytes transferred = 13839360 (d32c00 hex)”，则为：0xd32c00)
```

6. 烧写 A53UP Huawei LiteOS

```
mw.b 0x52000000 0xff 0x1000000
tftp 0x52000000 sample_vio.bin
nand erase 0x2A00000 0x1000000
nand write 0x52000000 0x2A00000 0x1000000
```



7. 烧写 Cortex-M7 Huawei LiteOS (此步选做)

```
mw.b 0x52000000 0xff 0x100000
tftp 0x52000000 sample.bin
nand erase 0x3A00000 0x100000
nand write 0x52000000 0x3A00000 0x100000
```

8. 设置启动参数

```
setenv bootargs 'mem=512M console=ttyAMA0,115200 clk_ignore_unused rw
root=/dev/mtdblock2 rootfstype=yaffs2
mtdparts=hinand:1M(boot),9M(kernel),32M(rootfs)'; sa
```

不启动 Cortex-M7:

```
setenv bootcmd 'nand read 0x45000000 0x2A00000 0x1000000; go_a53up
0x45000000; nand read 0x52000000 0x100000 0x900000;bootm 0x52000000'; sa
```

带 Cortex-M7 启动:

```
setenv bootcmd 'nand read 0x45000000 0x2A00000 0x1000000; go_a53up
0x45000000; config_m7; nand read 0x19000000 0x3A00000 0x100000; go_m7; nand
read 0x52000000 0x100000 0x900000;bootm 0x52000000'; sa
```

9. 重启系统

```
reset
```

----结束



3 开发前环境准备

3.1 管脚复用

无

3.2 连接串口

通过 DEMO 板上的串口连接主 CPU。

3.3 NFS 环境

通过 DEMO 板上的网口连接 NFS。



4 使用 SDK 和 DEMO 板进行开发

4.1 开启 Linux 下的网络

步骤 1. 设置网络

```
ifconfig eth0 hw ether 00:10:67:20:81:70;  
ifconfig eth0 10.67.208.170 netmask 255.255.254.0;  
route add default gw 10.67.208.1
```

以上为举例，IP 以实际为准。

步骤 2. 然后 ping 一下其他 IP，正常情况下网络将能正常工作。

----结束

4.2 使用 NFS 文件系统进行开发

步骤 1. 在开发阶段，推荐使用 NFS 作为开发环境，可以省去重新制作和烧写根文件系统的工作。

步骤 2. 挂载 NFS 文件系统的操作命令：

```
mount -t nfs -o nolock -o tcp -o rsize=32768,wsiz=32768 xx.xx.xx.xx:/your-nfs-path /mnt
```

步骤 3. 然后就可以在/mnt 目录下访问服务器上的文件，并进行开发工作。

----结束

4.3 开启 telnet 服务

网络正常后，运行命令 `telnetd&` 就可以启动单板 telnet 服务，然后才能使用 telnet 登录到单板。



4.4 运行 MPP 业务

步骤 1. 进入 mpp/out/linux/multi-core/ko 目录，加载驱动，例：

```
cd mpp/out/linux/multi-core/ko
./load3559av100_multicore -i -sensor0 imx477 -sensor1 imx477
```

步骤 2. 进入各 sample 目录下执行相应样例程序(sample 需要先在服务器上成功编译过)

```
cd mpp/sample/vio
./sample_vio 0 0
----结束
```



5 地址空间分配与使用

5.1 DDR 内存管理说明

1. 所有 DDR 内存中，一部分由操作系统管理，称为 OS 内存；另一部分由 osal 模块管理，供媒体业务单独使用，称为 MMZ 内存。
2. Single 端的 OS 内存起始地址为 0x42000000，内存大小可通过 bootargs 进行配置，例如第三章中的 setenv bootargs 'mem=256M ...'，表示分配给 Single OS 操作系统内存为 256M，您可以根据实际情况进行调整。
3. MMZ 内存由 osal 内核模块管理（mpp/ko 目录下的 hi_osal.ko），加载 osal 模块时，通过模块参数指定其起始地址及大小，可在 load 脚本中修改 MMZ 的起始地址 mmz_start 及大小 mmz_size。
4. 请注意任何区域的内存划分都不能重叠。

5.2 DDR 内存分配

参考《Hi3559A/C V100 开发环境用户指南.pdf》第 9 章 Hi3559AV100 内存分配。

注意：

- (1) 任何用途的内存区域地址空间都不能重叠。
- (2) OS 的管理内存存在配置启动参数时设置，默认为 512M，“setenv bootargs 'mem=512M ...'”。
- (3) MMZ 区域内存在 load 脚本中设定，如果有特殊区域需求，可以单独分区。例如
"insmod hi_osal.ko anony=1 mmz_allocator=hisi
mmz=anonymous,0,0x64000000,2048M:jpeg,0,0xe4000000,16M || report_error "。