

Hi3518E V20X 穿戴式 DV

Flash+DDR 小型化指南

文档版本 00B03

发布日期 2016-05-20

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

(上) 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

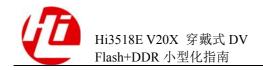
地址: 深圳市龙岗区坂田华为基地华为电气生产中心 邮编: 518129

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

本文为基于 Hi3518EV20X 做小型化开发的程序员而写,目的是介绍在 Hi3518EV20X 单板上进行 Linux 和 Huawei LiteOS 小系统开发、裁剪、优化及使用注意事项等内容。

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
Hi3518E	V200
Hi3518E	V201

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

日期	修订版本	描述	作者
2016-05-20	00B03	增加 8M SPI flash+32M DDR 相关优化	马君慧
2016-03-18	00B02	增加 Huawei LiteOS 相关优化	李坚
2015-12-03	00B01	初稿完成	马君慧

目 录

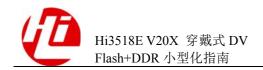
前	f 言f	i
1	Hi3518EV20X Flash+DDR 小型化综述	1
	裁剪及说明	
	2.1 uboot 裁剪	
	2.2 内核裁剪(Linux)	
	2.3 ROOTFS 的一些说明	4
	2.4 KO 加载注意事项及 VB 分配(Linux)	4
	2.5 WIFI 驱动裁剪(Linux)	5
3	优化	6
	3.1 应用程序优化(Linux)	6
	3.2 内存优化经验(Huawei LiteOS)	7
	33 各模块占用内存统计	7

插图目录

图 1-1 板 A 中 Linux 系统内存分配图	1
图 1-2 板 A 中 Linu 镜像 Flash 地址空间分配	1
图 1-3 板 B 中 Linux 系统内存分配图	1
图 1-4 板 B 中 Linu 镜像地址空间分配	2
图 1-5 Huawei LiteOS 整体内存分配图	2
图 3-1 MMZ 占用统计(Huawei LiteOS)	8
图 3.2 MM7 占田统计(Linux)	c

表格目录

表 2-1 A 板内核裁剪前后占用内存统计	3
表 2-2 B 板内核裁剪前后占用内存统计	
表 2-3 不需要加载 KO 列表	4
表 2-4 裁剪 ko 前后占用内存统计	5
表 2-5 VB 分配	5
表 2-6 WiFi 驱动裁剪前后大小统计	5
表 3.1 OSMEM 占田统计	7



1

Hi3518EV20X Flash+DDR 小型化综述

基于 Hi3518EV20X 的 SDK 可以分别运行 Linux 和 Huawei LiteOS 两种操作系统,它 们在 OS 内存占用和镜像大小上有较大的不同(Huawei LiteOS 占用内存更小,镜像更小),不过 MMZ 部分是一致的,本文中默认以 Linux 为基础进行描述,Huawei LiteOS 的特别之处会单独说明。

Hi3518EV20X 的 Linux 系统小型化基于两块单板实现,EMMC 板: 32G EMMC+32M DDR, SPI 板: 8M SPI Flash+32M DDR (下文会以 EMMC 板和 SPI 板描述), 会导致内存占用和裁剪手段的略微差异,以下会分别说明。

H3518EV20X 的 Huawei LiteOS 系统本身镜像较小,所以下文仅会涉及到 DDR 优化方案。

图1-1 EMMC 板中 Linux 系统内存分配图



图1-2 EMMC 板中 Linu 镜像 Flash 地址空间分配

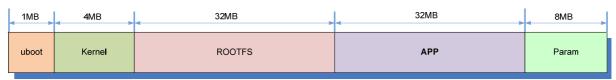


图1-3 SPI 板中 Linux 系统内存分配图

4		OS: 22MB		•	MMZ: 10MB	>
3260KB	6364KB	3848KB	1064KB	7992KB	8308KB	636KB
OS解压占用	OS运行占用	Media KO占用	WiFi KO 占用	其他应用占用	Media MMZ占用	REC占用

图1-4 SPI 板中 Linu 镜像地址空间分配

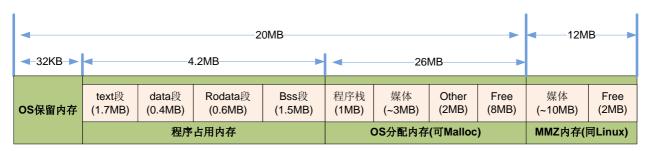
256KB	2048KB	4846KB	512KB	512KB
uboot	Kernel	ROOTFS	APP	Param

Linux 小系统可正常运行场景如下:

场景 1: 一路 720p@H.264@30fps@2Mbps+3dnr +图像质量自适应+一路 720p@JPEG@6fps+DCF(exif)+AAC 16K+一路录像(MP4 封装)+WIFI_AP+单用户 APK 客户端点播+APK 客户端参数修改和文件管理

场景 2: 一路 720p@H.264@30fps@2Mbps+一路 720p@JPEG@6fps+3dnr+ AAC 16K + 一路录像(MP4 封装)+WIFI_STA(包含零配置)+单用户 APK 客户端点播+ APK 客户端参数修改和文件管理

图1-5 Huawei LiteOS 整体内存分配图



目前来说,在 32M DDR 系统上剩余内存还较多,OS 部分约 6MB,MMZ 也还剩余近 1.5MB,可供产品层做其它扩展,因此在 Huawei LiteOS 上 32M 内存规格与 64M 内存规格是一致的,具体如下:

一路 720p@H.264@30fps@3Mbps+3dnr +图像质量自适应+一路 720p@JPEG@6fps+DCF(exif)+AAC 16K+一路录像(MP4 封装) +WIFI_AP/WIFI_STA+单用户 APK 客户端点播+APK 客户端参数修改和文件管理

2 裁剪及说明

2.1 uboot 裁剪

此裁剪是为了在 SPI 板上节省 Flash 空间,基于 SDK 发布包中 uboot 配置 hi3518ev201_config 修改:

- 1. 去掉 nand flash 初始化;
- 2. 修改环境变量偏移地址和环境变量占用大小;
- 3. 去掉 DDR training。

具体可以参照《hi3518ev200 dvs.h》。

2.2 内核裁剪(Linux)

EMMC 板由于使用了 EMMC,推荐使用 ext4 文件系统,此处将内核中其他文件系统、调试信息以及其他不支持的规格都裁剪,其余一些细微的裁剪请参看文件《hi3518ev20x_mini_emmc_config》。

表2-1 EMMC 板内核裁剪前后占用内存统计

	裁剪前(单位: KB)	裁剪后(单位: KB)
内核解压后占用内存	5556	2948
内核运行占用内存	8748	5560

SPI 板使用 8M SPI Flash,要求镜像 size 比较小,而且应用程序和 WiFi 需要读写参数,所以选择支持 jffs2 和 squashfs 文件系统,squashfs 选用 xz 压缩方式,其余裁剪参照文件《hi3518ev20x mini spi config》。

由于 SPI 板上必须支持两种文件系统 jffs2 和 squashfs,并且 sdio 驱动当前未能裁剪优化,SPI 板上的 kernel 大小比 EMMC 板大一些,为了保持 EMMC 板和 SPI 板应用规格一致,更多优化请参照章节 2.4 KO 加载注意事项及 VB 分配(Linux)和章节 2.5 WIFI 驱动裁剪(Linux)。

表2-2 SPI 板内核裁剪前后占用内存统计

	裁剪前(单位: KB)	裁剪后(单位: KB)
内核解压后占用内存	5556	3260
内核运行占用内存	8748	6364

□ 说明

- 内核的裁剪与对比均是基于 SDK 的正式发布版本。
- 以上内核均不支持 USB、不支持 framebuf、不支持以太网及 nfs、不支持进程间通信。

2.3 ROOTFS 的一些说明

- 1. ROOTFS 的打包格式一定要注意与 kernel 中的 config 匹配: A 板采用的是 ext4 文件系统, block 大小根据 ROOTFS 实际大小决定; B 板中 kernel 中支持 jffs2 和 squashfs 两种文件系统, squashfs 采用 xz 压缩格式。
- 2. B 板 Flash 较小, busybox 中一些不常用的命令尽量裁剪。比如 add-shell、fgconsole、flock、groups、lsof、lspci、lsusb、lzcat、lzma 等。

2.4 KO 加载注意事项及 VB 分配(Linux)

- 1. SDK 发布的 ko 在 Hi3518E 的穿戴式 DV 中并不全部需要,所以可将没有需要的 ko 加载去掉,节省 ko 加载需要分配的 os 内存以及 mmz。
- 2. 采用单 buffer 低延时模式可以节省 VB, 具体使用方式及注意事项: Linux 系统请参考《HiMPP IPC V2.0 媒体处理软件开发参考》, Huawei LiteOS 请参考《HiMPP IPC V3.0 媒体处理软件开发参考》

表2-3 不需要加载 KO 列表

KO 名称	删除原因简述	删除后内存收益(KB)
hi3518e_adec	无对讲及 HDMI 等实时输出需求	28
hi3518e_ao	无对讲及 HDMI 等实时输出需求	188
hi3518e_tde.ko	无屏无 UI	296
hi3518e_region.ko	不需要时间标签	48
hi3518e_vou.ko	无屏无 UI	60
hifb.ko	无屏无 UI	1620
hi3518e_ive.ko	不需要智能算法	200

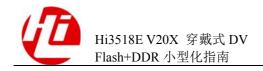


表2-4 裁剪 ko 前后占用内存统计

	裁剪前(单位: KB)	裁剪后(单位: KB)
加载 ko 占用内存	5684	3848

表2-5 VB 分配

VB 组编号	VB 块大小	VB 块数
0	1280×720×1.5	2
1	320×240×1.5	1

□ 说明

- ko 的加载细节请参考文件《hi3518ev20x_mini_loadko》。
- 如果不需要缩略图,则 hi3518e vgs.ko 可以不加载, VB 也只需要分配第0组。

2.5 WIFI 驱动裁剪(Linux)

通过修改 WiFi 驱动的 Makefile,裁剪 WIFI-Direct(P2P 直连)和 BT 蓝牙模块,去除调试打印信息,编译成 release 版本。缩小驱动 kzalloc 和 kmalloc 动态分配的 buffer 大小。

表2-6 WiFi 驱动裁剪前后大小统计

	裁剪前(单位: KB)	裁剪后(单位: KB)
KO大小	752	295

□ 说明

WiFi 驱动具体的小型化措施请参考文档《WiFi 驱动小型化方案用户指南.pdf》。

3 优化

3.1 应用程序优化(Linux)

1. 限制栈大小为 1024KB 或者更小

默认的栈大小为 **8092KB**,如果内存较小时,会导致线程创建不成功。根据实际的业务需要的栈空间,修改栈的限制大小为 **1024KB**,如果业务更少则可以改成 512KB 或者更小。

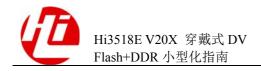
有两种方法可以修改栈大小:

- 使用 ulimit -s 1024 命令,在应用程序启动之前调用一次。
- 在 main 函数最开始调用函数 pthread_attr_setstacksize 也可实现对单个应用程序修改栈空间的目的。
- 2. 修改系统最低保留虚拟内存 min_free_kbytes

系统的最低保留虚拟内存 min_free_kbytes 默认为 **512KB**,在保证业务在更小剩余内存下也可以正常运行的前提下,此值也可修改的更小。此处裁减较为保守,仍然采用默认值。

可以通过 echo 570 > /proc/sys/vm/min free kbytes 修改。

- 3. 优化代码中内存使用(栈、堆、常量、全局变量)
 - 避免应用程序代码中申请后不用的变量;
 - 不要随意申请大块的内存,够用即可;
 - 冗余的内存使用还包含不用的功能模块的初始化;
- 4. 由于主进程数据段占用内存已经接近 **70%**,如果 fork 子进程必定消耗大量内存而且会有大概率失败,所以应用程序中禁用 fork、system 等进程创建函数:如 himm调用、mkfs.vfat 调用等。
- 5. 用到 himm 配置寄存器的地方也可放到 uboot 里配置,可以节省一些内存;根据统计,himm 配置拉高 WiFi 的 regon 引脚就会消耗 10KB 左右内存,如果配置的寄存器很多的话,会消耗更多内存。
- 6. WIFI-STA 模式下用到的 wpa_supplicant 和 udhcpc 等开源软件编译成库的形式直接调用函数。



3.2 内存优化经验(Huawei LiteOS)

Huawei LiteOS 内存分配都是按物理内存直接分配,不像 LinuxOSE 有虚拟内存。如在 Huawei LiteOS 上某模块只需要 100KB 的内存,但按照 1MB 的内存去分配时,系统可用内存就实实在在的少了 1MB,但在 Linux 上,尽管也分配了 1MB 的内存,但是最后的物理内存占用是根据实际内存来分配的,像上面这个例子中,系统就只会减少 100KB 的可用内存。因此在 Huawei LiteOS 上对内存分配的总原则是:按需分配。

1. 线程堆栈大小的设置

Linux 上的栈一般都是以 M 为单位设置的,但在 Huawei LiteOS 上,需要根据线程的实际使用大小来配置(比该线程的最大栈使用值大一点就行),目前 OS 中默认的栈大小为 0x6000,即 24KB,分多了就会浪费内存,如果分少了就会踩内存导致异常。

2. 根据产品需求去掉不必要的模块 例如裁剪掉 Huawei LiteOS 中不需要的动态加载、JFFS2 与 NFS 文件系统。

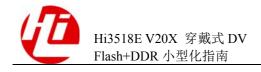
为避免内存碎片和性能损耗,对于某些需要频繁申请和释放的内存使用需要转换成静态内存或在初始化时一次性分配。

3.3 各模块占用内存统计

内存使用主要分两大块管理,由于内存较小,防止内存分的过碎,不做多余的分区。 一是 MMZ,media+rec 专用,一是 OSMEM,os+其他应用程序模块使用。

表3-1 OSMEM 占用统计

模块名称	OSMEM 占用(单位 KB)
Kernel 镜像解压	3092
Kernel 运行	7764
加载 sdk 提供 ko	3864
加载 WIFI 的 ko	2052 (包含了 ap 和 sta 的功能)
单独运行 WIFI 的 ap 模式	2296
单独运行 WIFI 的 sta 模式	2516
MCU	140
参数区	250
按键管理	140
日志	20
音视频	750
图像自适应	32



模块名称	OSMEM 占用(单位 KB)
存储管理+文件管理	210
抓拍管理	40
录像管理	20
点播管理	400
录像 buf 管理	50

□ 说明

由于 WIFI 的 ko 同时包含了 ap 和 sta 的功能,所以在统计"单独运行 WIFI 的 ap 模式"和"单独运行 WIFI 的 sta 模式"两部分时,会把不属于本模块功能的部分统计进去,两个数字体现的应该是处理加载 ko 之外运行功能消耗的内存。

图3-1 MMZ 占用统计(Linux)

```
ZONE: PHYS (0x81600000,
                           0x81FFFFFF), GFP=0, nBYTES=10240KB, NAME="anonymous
                           0x81603FFF), kvirt=0x (null), flags=0x00000000, length=16KB,
   -MMB: phys(0x81600000,
                                                                                                       name="sys_scale_coef"
                                                                                                      name="sys_defliscale_"
name="ACM LUTS"
   -MMB: phys (0x81604000,
                                                     (null), flags=0x00000000,
                           0x81607FFF), kvirt=0x
                                                                                 length=16KB,
   -MMB: phys(0x81608000,
                           0x8160FFFF), kvirt=0x
                                                     (null), flags=0x00000000,
                                                                                 length=32KB,
                                                     (null), flags=0x00000000,
   -MMB: phys (0x81610000,
                           0x81610FFF), kvirt=0x
                                                                                                    "ACM Coef
                                                                                 length=4KB, name=
                                                                                                      name="ISP shadow mem"
name="VbPool"
   -MMB: phys (0x81611000,
                            0x81614FFF), kvirt=0x
                                                     (null), flags=0x00000000,
                                                                                 length=16KB
   -MMB: phys (0x81615000,
                           0x818B8FFF), kvirt=0x
                                                     (null), flags=0x00000000,
                                                                                 length=2704KB,
                                                                                                      name="VirtReg0"
name="ISPStat"
    -MMB: phys (0x818B9000,
                           0x818C8FFF), kvirt=0x
                                                     (null), flags=0x00000000,
                                                                                 length=64KB,
                                                     (null), flags=0x00000000,
   -MMB: phys(0x818C9000, 0x818CFFFF), kvirt=0x
                                                                                 length=28KB,
                                                     (null), flags=0x000000000,
                                                                                                       name="ISPDRCStat"
                                                                                 length=24KB,
   -MMB: phys(0x818D0000,
                           0x818D5FFF), kvirt=0x
                                                                                 length=4KB, name="ISPDCFInfo
                                                     (null), flags=0x00000000,
   -MMB: phys(0x818D6000,
                           0x818D6FFF), kvirt=0x
                                                     (null), flags=0x00000000,
                                                                                 length=4KB, name="VirtReg1
   -MMB: phys(0x818D7000,
                           0x818D7FFF), kvirt=0x
                                                                                             name="VirtReg2"
   -MMB: phys (0x818D8000,
                           0x818D8FFF), kvirt=0x
                                                     (null), flags=0x000000000,
                                                                                 length=4KB,
                                                                                                      name="recbuf"
name="Ref&Mad buf"
   -MMB: phys (0x818D9000,
                           0x8198EFFF), kvirt=0x
                                                     (null), flags=0x000000000,
                                                                                 length=728KB,
    -MMB: phys(0x8198F000,
                           0x81B26FFF), kvirt=0x
                                                     (null), flags=0x00000000,
                                                                                 length=1632KB,
                                                     (null), flags=0x000000000,
   -MMB: phys(0x81B27000, 0x81B27FFF), kvirt=0x
                                                                                 length=4KB, name=
                                                                                                    RegionLuma buf'
                                                     (null), flags=0x000000000,
                                                                                                      name="model buf"
name="h264e0_Str"
name="h264e0_Ct"
   -MMB: phys(0x81B28000,
                           0x81B2DFFF), kvirt=0x
                                                                                 length=24KB.
   -MMB: phys(0x81B2E000,
                           0x81C10FFF), kvirt=0x
                                                     (null), flags=0x00000000,
                                                                                 length=908KB,
                                                     (null), flags=0x00000000,
                                                                                 length=12KB,
   -MMB: phys(0x81C11000,
                           0x81C13FFF), kvirt=0x
                                                     (null), flags=0x00000000,
   -MMB: phys (0x81C14000,
                           0x81CF6FFF), kvirt=0x
                                                                                 length=908KB,
                                                                                                       name="h264e0_Rcn
   -MMB: phys (0x81CF7000,
                           0x81DD8FFF), kvirt=0x
                                                     (null), flags=0x000000000,
                                                                                 length=904KB,
                                                                                                       name="h264e0_RcnVb"
                                                                                                      name="Jpege9"
name="Jpege10"
   -MMB: phys(0x81DD9000,
                           0x81EBCFFF), kvirt=0x
                                                     (null), flags=0x00000000,
                                                                                 length=912KB,
  -MMB: phys(0x81EBD000, 0x81ED2FFF), kvirt=0x
                                                     (null), flags=0x00000000,
                                                                                 length=88KB,
 -MMZ_USE_INFO:
total size=10240KB(10MB), used=9036KB(8MB + 844KB), remain=1204KB(1MB + 180KB), zone_number=1, block_number=22
```

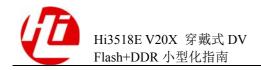


图3-2 MMZ 占用统计(Huawei LiteOS)

```
HuaweiLite OS # +---ZONE: PHYS(0X82000000, 0X82FFFFFF), GFP=0, nBYTES=16384KB, NAME="anonymous" |
|-MMB: phys(0X82000000, 0X82003FFF), kvirt=0X0X0, flags=0X00000000, length=16KB, name="sys_scale_coef" |
|-MMB: phys(0X82008000, 0X8203FFF), kvirt=0X0X0, flags=0X00000000, length=16KB, name="sys_defliscale_" |
|-MMB: phys(0X8203E000, 0X82045FFF), kvirt=0X0X0, flags=0X00000000, length=216KB, name="Coeff |
|-MMB: phys(0X8204000, 0X82046FFF), kvirt=0X0X0, flags=0X00000000, length=4KB, name="ACM LUTS" |
|-MMB: phys(0X82047000, 0X82046FFF), kvirt=0X0X0, flags=0X00000000, length=16KB, name="ACM COEff |
|-MMB: phys(0X82048000, 0X82834FFF), kvirt=0X0X0, flags=0X00000000, length=16KB, name="TSP shadow mem |
|-MMB: phys(0X82835000, 0X82831FFF), kvirt=0X0X0, flags=0X00000000, length=16KB, name="Vbpool" |
|-MMB: phys(0X82835000, 0X82861FFF), kvirt=0X0X0, flags=0X00000000, length=16KB, name="TSP shadow mem |
|-MMB: phys(0X82835000, 0X8286FFF), kvirt=0X0X0, flags=0X00000000, length=16KB, name="TSP shadow mem |
|-MMB: phys(0X8286000, 0X8286FFF), kvirt=0X0X0, flags=0X00000000, length=2KB, name="TSP shadow mem |
|-MMB: phys(0X8286000, 0X8286FFF), kvirt=0X0X0, flags=0X00000000, length=2KB, name="TSP shadow mem |
|-MMB: phys(0X8286000, 0X8287FFFF), kvirt=0X0X0, flags=0X00000000, length=2KB, name="TSP shadow mem |
|-MMB: phys(0X82887000, 0X8287FFFF), kvirt=0X0X0, flags=0X00000000, length=4KB, name="TSP shadow mem |
|-MMB: phys(0X8288000, 0X8288FFF), kvirt=0X0X0, flags=0X00000000, length=4KB, name="TSP shadow mem |
|-MMB: phys(0X8288000, 0X8288FFF), kvirt=0X0X0, flags=0X00000000, length=4KB, name="TSP shadow mem |
|-MMB: phys(0X8288000, 0X8288FFF), kvirt=0X0X0, flags=0X00000000, length=4KB, name="TSP shadow mem |
|-MMB: phys(0X8288000, 0X8288FFF), kvirt=0X0X0, flags=0X00000000, length=4KB, name="TSP shadow mem |
|-MMB: phys(0X8288000, 0X8288FFF), kvirt=0X0X0, flags=0X00000000, length=4KB, name="ISPDCFINFO" |
|-MMB: phys(0X8288000, 0X8288FFF), kvirt=0X0X0, flags=0X00000000, length=2KB, name="NirtReg2" |
|-MMB: phys(0X8288000, 0X8288FFF)
```