

# U-boot 下自动升级

# 使用手册移植说明

文档版本 08

发布日期 2016-08-31

#### 版权所有 © 深圳市海思半导体有限公司 2012-2016。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

本文档主要是指导 uboot 下 USB 或者 SD 卡升级的使用和移植,主要包括升级包的制作、USB 下自动升级、USB 下手动升级、SD 卡自动升级、SD 卡手动升级以及相关的注意事项。

## □ 说明

- Hi3520D/Hi3515A/Hi3535/Hi3536/Hi3521A/Hi3520DV300 平台仅支持 U 盘升级,不支持 SD 卡升级,本文未做特殊说明,Hi3516D 与 Hi3516A 完全一致。
- 本文未做特殊说明, Hi3520DV300 与 Hi3521A 完全一致; Hi3518EV201、Hi3516CV200 与 Hi3518EV200 完全一致。

# 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	说明
Hi3516C	V100	-
Hi3518	V100	包含 Hi3518A 和 Hi3518C
Hi3518E	V200	包含 Hi3518EV200、Hi3518EV201 和 Hi3516CV200
Hi3520A	V100	-
Hi3521	V100	-
Hi3531	V100	-
Hi3531A	V100	Hi3531A 不支持 SD 卡升级
Hi3520D	V100	Hi3520D 不支持 SD 卡升级
Hi3515A	V100	Hi3515A 不支持 SD 卡升级
Hi3535	V100	Hi3535 不支持 SD 卡升级
Hi3516A	V100	-
Hi3516D	V100	-



产品名称	产品版本	说明
Hi3536	V100	Hi3536 不支持 SD 卡升级
Hi3521A	V100	Hi3521A 不支持 SD 卡升级
Hi3520D	V300	Hi3520DV300 不支持 SD 卡升级
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
2016-08-31	08	添加 Hi3516CV300 的相关内容。
2016-08-10	07	添加 Hi3519V101 的相关内容。
2015-10-31	06	添加 Hi3519V100 的相关内容。
2015-09-01	05	添加 Hi3531A 的相关内容。
2015-08-05	04	添加 Hi3518EV200、Hi3518EV201 和 Hi3516CV200 的相 关内容。
2015-06-26	03	添加 Hi3521A/Hi3520DV300 的相关内容。
2015-04-01	02	添加 Hi3536 的相关内容。
2014-12-30	01	第 1 次正式版本发布,添加 Hi3516D 的相关内容。
2014-10-19	00B40	第 5 次版本发布,添加 Hi3516A 的相关内容。
2013-11-18	00B30	第 4 次版本发布。
2013-05-09	00B20	第3次版本发布。



修订日期	版本	修订说明
2012-08-30	00B10	第2次版本发布。
2012-08-15	00B01	第1次临时版本发布。



# 目录

1 U-boot 下系统升级使用说明	1
1.1 操作准备	1
1.2 升级流程	
1.3 操作过程	
1.4 操作示例	
1.4.1 编译支持系统升级的 U-boot 镜像	
1.4.2 制作系统升级包	
1.4.3 升级系统	
1.5 操作中需要注意的问题	
2 U-boot 下系统升级移植说明	
2.1 移植应用升级	7
2.1.1 移植应用升级操作步骤	7
2.1.2 应用升级移植示例	
2.2 升级系统的代码移植	
2.2.1 移植步骤	10
222 移植示例	10



# 插图目录



# 【 U-boot 下系统升级使用说明

# 1.1 操作准备

U-boot 下 SD 卡/U 盘升级操作准备如下:

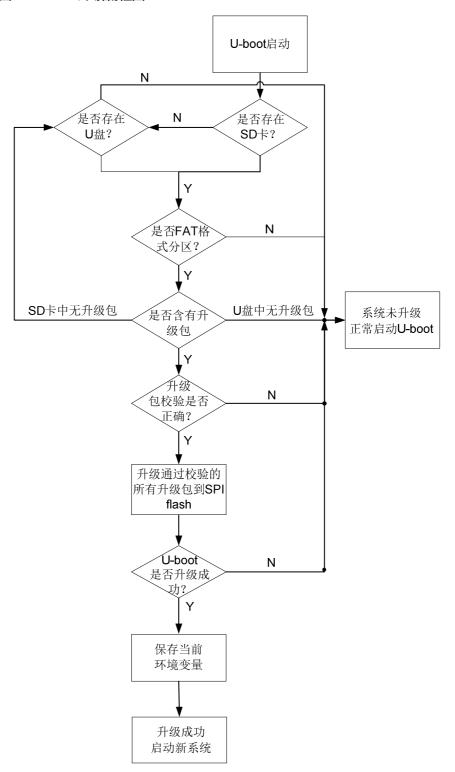
- 打开代码中的升级开关,编译支持系统升级的 U-boot 镜像
- 制作升级镜像
- 存储介质准备(FAT32 格式的 U 盘或 SD 卡)

# 1.2 升级流程

U-boot 升级流程图如图 1-1 所示。



#### 图1-1 U-boot 升级流程图



#### □ 说明

当同时插入 SD 卡和 U 盘时,若 SD 卡中没有有效升级包则系统自动转入 U 盘升级,若 SD 卡中含有有效升级包则直接使用 SD 卡中的升级包升级系统,不再扫描 U 盘中的内容。



# 1.3 操作过程

操作过程如下:

#### 步骤 1. 编译支持系统升级的 U-boot 镜像

• 修改芯片配置文件,开启升级开关

include/configs/hi3516c.h
include/configs/hi3518a.h
include/configs/hi3518c.h
hi3518c
hi3518c
hi3518c

- include/configs/hi3518ev200.h [hi3518ev200]

- include/configs/hi3518ev201.h 【hi3518ev201】

include/configs/hi3516cv200.h 【hi3516cv200】

include/configs/ godcare.h
 include/configs/ godarm.h
 include/configs/ godnet.h
 include/configs/ hi3535.h
 [hi3520a]
 [hi3521]
 [hi3531]

- include/configs/ hi3516a.h 和 include/configs/ hi3516a\_spinand.h 【hi3516a】

- include/configs/ hi3536.h 和 include/configs/ hi3536\_spinand.h 【hi3536】

include/configs/ hi3521a.h
include/configs/ hi3531a.h
hi3521A I
hi3531A I

- include/configs/hi3519.h 和 include/configs/hi3519\_nand.h 【hi3519v100】

include/configs/hi3519v101.h 和 include/configs/hi3519v101\_nand.h 【hi3519v101】

- include/configs/hi3516cv300.h [hi3516cv300]

- 编译 U-boot 镜像
- 烧写 U-boot 镜像到开发板

#### 步骤 2. 制作系统升级包

- 制作 U-boot 升级包
  - 编译生成正常的 U-boot 镜像
  - 使用 mkimage 工具制作升级包 u-boot
- 制作 kernel 升级包
  - 编译生成正常的内核 uImage
  - 将 uImage 重命名为 kernel
- 制作文件系统升级包
  - 编译生成正常的文件系统
  - 使用 mkimage 工具制作文件系统升级包 rootfs

步骤 3. 插入存放有升级包的 FAT32 格式的 SD 卡/U 盘,启动单板,自动升级系统。



#### ----结束

# 1.4 操作示例

此操作以 Hi3516A 开发环境为例,进行完整的 SD 升级操作。U 盘升级类似,只需要把升级包放在 U 盘中,插入 U 盘进行升级操作即可,这里不再举例。

## 1.4.1 编译支持系统升级的 U-boot 镜像

• 修改芯片配置文件开启自动升级功能

/\*-----

- \* sdcard/usb storage system update
- \* -----\*/

```
/* #define CONFIG_AUTO_UPDATE 1 */
#ifdef CONFIG_AUTO_UPDATE
    #define CONFIG_AUTO_SD_UPDATE 1
#define CONFIG_AUTO_USB_UPDATE 1
```

#endif

开启自动升级功能只需要将上述代码段的红色代码行中的注释去掉即可。

• 编译支持自动升级功能的 uboot

make ARCH=arm CROSS\_COMPILE=arm-hisiv300-linux- hi3516a\_config
make ARCH=arm CROSS\_COMPILE=arm-hisiv300-linux- -j

● 使用 mkboot.sh 以及 reg\_info.bin 制作可烧写运行的 uboot

./mkboot.sh reg\_info.bin u-boot-3516a-update.bin

#### M H

- Hi3516A 提供 arm-hisiv300-linux-和 arm-hisiv400-linux-两种交叉编译工具链,请根据具体情况进行选择。
- mkboot.sh 以及 reg\_info.bin 存放在发布包 /Hi3516A\_SDK\_V1.0.1.0/package/osdrv/tools/pc/uboot\_tools/中, 关于 uboot 编译的更多信息, 请查看《Hi3516A/Hi3516D U-boot 移植应用开发指南》。
- 最后生成的 uboot 名 u-boot-3516a-update.bin 可以任取,此镜像是具备自动升级功能的 U-boot。

# 1.4.2 制作系统升级包

升级包镜像与正常编译出来的镜像有差异,是使用 mkimage 工具处理过的,以便进行升级包有效性校验。

#### 步骤 1. Uboot 升级包制作。

mkimage -A arm -T firmware -C none -n hiboot -d u-boot-hi3516a.bin u-boot



#### □ 说明

- u-boot-hi3516a.bin 是指任意的可以直接烧写启动的 hi3516a u-boot 镜像,可以是不具备升级功能的 u-boot 也可以是支持自动升级的 u-boot。
- 最后生成的 u-boot 升级包必须以 u-boot 命名

#### 步骤 2. Kernel 升级包制作。

由于内核 uImage 原本就是使用 mkimage 工具制作出来的,直接将 uImage 重命名为 kernel 即可

#### 步骤 3. Rootfs 升级包制作。

mkimage -A arm -T filesystem -C none -n hirootfs -d rootfs\_256k.jffs2
rootfs

□ 说明

最后生成的文件系统升级包必须以 rootfs 命名。

----结束

## 1.4.3 升级系统

- 将 1.4.1 中编译得到的支持自动升级的 u-boot 镜像烧写到 SPI flash 中或者准备好 SD 卡/U 盘和升级包后直接下载到内存中运行。
- 格式化 SD 卡/U 盘为 FAT 格式。
- 将 1.4.2 中制作的升级包拷贝到格式化好的 SD 卡/U 盘中。格式化 SD 卡/U 盘具体方法请参考《外围设备驱动 操作指南.doc》附录。
- 开发板上电,uboot 启动,开始自动升级 SD 卡/U 盘中存放的升级包。
- 升级打印以及打印说明如下:

```
//升级 u-boot
reading u-boot
                    //扫描 u-boot 升级包
                    //复制 u-boot 升级包到内存
reading u-boot
                    //擦除 u-boot 将要升级的 SPI 区域
flash erase...
                    //将新的 u-boot 数据写入 SPI 指定区域
flash write...
//升级内核
reading kernel
reading kernel
flash erase...
flash write...
//升级文件系统
reading rootfs
reading rootfs
flash erase...
flash write...
//保存当前环境变量
Erasing SPI flash, offset 0x00080000 size 256K ...done
```



Writing to SPI flash, offset 0x00080000 size 256K ...done //接下来将自动启动新系统

# 1.5 操作中需要注意的问题

- 制作文件系统升级包时注意文件系统的块大小必须与SPI的擦除块大小一致。
- SD 卡或者 U 盘必须格式化成 FAT 格式。
- 若 SD 卡或 U 盘有多个分区时,升级包必须放在第一个分区,否则升级程序扫描不到升级包。
- 当 SD 卡和 U 盘同时插入开发板,若 U 盘和 SD 卡中均存放有有效升级包时,系 统将升级 SD 卡中的系统镜像而忽略 U 盘中的内容。当 SD 卡中不含有升级包或 者升级包无效时,系统将自动扫描 U 盘中的升级包进行系统升级。
- 系统升级完成后会自动保存原来的环境变量,手动重启系统后可能需要手动根据 新的系统烧写位置设置并保存环境变量,以便系统能够正常启动。



# 2 U-boot 下系统升级移植说明

□ 说明

以下增加的 app 文件升级只是一个举例说明,具体的地址设置和命名要根据自己使用的地址来设置。

# 2.1 移植应用升级

## 2.1.1 移植应用升级操作步骤

内核下使用网口的操作涉及到以下几个方面:

- 添加文件名信息
- 添加新的文件索引号
- 添加该文件在Flash 上的存储位置
- 添加新文件的校验过程

## 2.1.2 应用升级移植示例

以下移植修改均在文件./product/hiupdate/auto update.c 中修改

## 2.1.2.1 添加文件名信息

```
/* possible names of files on the medium. */
#define AU_FIRMWARE "u-boot"
#define AU_KERNEL "kernel"
#define AU_ROOTFS "rootfs"
#define AU_APP "app"
/* pointers to file names */
char *aufile[AU_MAXFILES] = {
    AU_FIRMWARE,
    AU_KERNEL,
    AU_ROOTFS
    AU_APP
```



};

### 2.1.2.2 添加新的文件索引号

```
/* index of each file in the following arrays */
#define IDX_FIRMWARE 0
#define IDX_KERNEL 1
#define IDX_ROOTFS 2
#define IDX_APP 3
/* max. number of files which could interest us */
#define AU_MAXFILES 4
```

## 2.1.2.3 添加该文件在 Flash 上的存储位置

有两种方法可以指定升级时各镜像存放在 Flash 上的位置: 在代码中指定和在环境变量中指定。如果环境变量中有相关参数指定存放位置,则优先选择环境变量中的设置。在调试时,若需要频繁变化烧写位置,建议直接使用环境变量指定的方式指定烧写位置。

#### 步骤 1. 在代码中修改

```
/* layout of the FLASH. ST = start address, ND = end address. */
#define AU FL FIRMWARE ST
                            0x0
#define AU FL FIRMWARE ND
                            0x7FFFF
#define AU FL KERNEL ST
                            0x100000
#define AU FL KERNEL ND
                            0x5FFFFF
#define AU FL ROOTFS ST
                            0x600000
#define AU FL ROOTFS ND
                            0xbFFFFF
#define AU FL APP ST
                            0x..... //Flash 上的起始位置
                           0x..... //Flash 上的结束位置
#define AU FL APP ND
/* sizes of flash areas for each file */
long ausize[AU MAXFILES] = {
      (AU FL FIRMWARE ND + 1) - AU FL FIRMWARE ST,
      (AU_FL_KERNEL_ND + 1) - AU_FL_KERNEL_ST,
      (AU FL ROOTFS ND + 1) - AU FL ROOTFS ST,
(AU FL APP ND + 1) - AU FL APP ST,
}:
/* array of flash areas start and end addresses */
struct flash layout aufl layout[AU MAXFILES] = {
      { AU_FL_FIRMWARE_ST, AU_FL_FIRMWARE_ND, },
      { AU FL KERNEL ST,
                           AU FL KERNEL ND, },
      { AU_FL_ROOTFS_ST, AU_FL_ROOTFS_ND,
{ AU FL APP ST, AU FL APP ND, },
};
```

步骤 2. 在环境变量中直接赋值



#### 找到函数 do auto update (void), 红色部分为修改内容

```
/*
* Get image layout from environment.
  * If the start address and the end address
  * were not definedin environment virables,
  * use the default value
  */
  get_update_env("firmware_st", "firmware_nd");
  get_update_env("kernel_st", "kernel_nd");
  get_update_env("rootfs_st", "rootfs_nd");
  get_update_env("app_st", "app_nd");
```

当前环境变量中设置有"app\_st"和"app\_nd"时升级位置将按照环境变量中设置的位置升级系统,否则按照 2.1.2.3 中代码的设置值升级。

#### ----结束

## 2.1.2.4 添加新文件的校验过程

找到函数 au check header valid(int idx, long nbytes), 红色部分为修改内容

```
/* check the type - could do this all in one gigantic if() */
      if ((idx == IDX FIRMWARE) && (hdr->ih type != IH TYPE FIRMWARE)) {
             printf("Image %s wrong type\n", aufile[idx]);
             return -1;
      if ((idx == IDX KERNEL) && (hdr->ih type != IH TYPE KERNEL)) {
             printf("Image %s wrong type\n", aufile[idx]);
             return -1;
      if ((idx == IDX_ROOTFS) &&
                    (hdr->ih type != IH TYPE RAMDISK) &&
                    (hdr->ih type != IH TYPE FILESYSTEM)) {
             printf("Image %s wrong type\n", aufile[idx]);
             ausize[idx] = 0;
             return -1;
if ((idx == IDX APP) &&(hdr->ih type != IH TYPE FILESYSTEM)) {
printf ("Image %s wrong type\n", aufile[idx]);
ausize[idx] = 0;
return -1;
```



## □ 说明

mkimage 说明:制作 APP 升级镜像时用mkimage 加入校验信息时建议采用'-T filesystem'这个参数。

# 2.2 升级系统的代码移植

## 2.2.1 移植步骤

- 在芯片配置文件中增加驱动和升级模块的编译控制开关
- 完善新芯片平台的 MMC 驱动
- 完善新芯片平台的 USB OHCI 驱动
- 增加 U-boot 下自动升级入口

## 2.2.2 移植示例

下面以 Hi3516A 芯片为例,完整移植自动升级功能到新平台,新芯片平台可参考移植。

## □ 说明

本示例是在完成 FAT 文件系统、MMC 驱动、USB OHCI 驱动移植后的操作请先保证在海思发布的最新 U-boot 版本下或者已完整移植好上述三部分驱动的 U-boot 版本。此三部分的具体移植详见《U-boot T U 盘 T SD 卡自动升级功能设计说明》。

• 增加编译控制开关

在芯片配置文件最后插入下面宏定义

/*			
* sdcard/usb storage system update			
*			-*/
#define CONFIG_AUTO_UPDATE	1 /	* 此为升级功能总开关,治	主释
此行则关闭系统自动升级功能 */			
#ifdef CONFIG_AUTO_UPDATE			
#define CONFIG_AUTO_SD_UPDATE	1		
<pre>#define CONFIG_AUTO_USB_UPDATE</pre>	1		
#endif			
<pre>#defineLITTLE_ENDIAN</pre>	1		
#define CONFIG_DOS_PARTITION	1		
#define CONFIG_CMD_FAT	1		
/*			-
* sdcard			
*			-*/
#ifdef CONFIG_AUTO_SD_UPDATE			
<pre>#define CONFIG_HIMCI_HI3516a</pre>	//此开关用于MMC	驱动编译	



```
#define REG BASE MCI
                                    0x206e0000
     #define CONFIG HIMCI V100
     #define CONFIG GENERIC MMC
     #define CONFIG MMC
                                    1
     #define CONFIG CMD MMC
#endif
注意: Hi 3516A自带两个SD卡控制器, 其寄存器基地址分别为0x206e0000和0x206f0000, 请根
据具体情况配置宏定义REG BASE MCI。
* usb
* -----*/
#define CONFIG USB OHCI
                                    1
#define CONFIG CMD USB
#define CONFIG USB STORAGE
                                     1
#define CONFIG LEGACY USB INIT SEQ
注: Hi3536中USB支持USB3.0自动升级,所以还需添加对USB3.0支持的宏,即:
#define CONFIG USB XHCI
#define CONFIG SYS USB XHCI MAX ROOT PORTS 2
```

#### ● 完善 MMC 驱动

增加 drivers/mmc/himciv100\_3516a.c 文件, 实现 Hi3516A 平台下 MMC 模块驱动差异性部分

```
#define PERI CRG49
                            (CRG REG BASE + 0xC4)
#define SDIOOCLK PCTRL
                             (0x1 << 4)
#define SDIOO_CLK_BIT_HIGH
                              (1U << 3)
#define SDIOO CLK BIT LOW
                              (1U << 2)
#define SDIO0 CKEN
                            (0x1 << 1)
                            (0x1 << 0)
#define SDIO RESET
#define SYS_PERIPHCTRL4 (0x20120004)
#define SDIO0_DET_MODE (0x1 << 2)</pre>
#define REG UPDATE MCI BASE REG BASE MCI
#define MMC UHS REG EXT 0x108
static void hi_mci_sys_init(void)
      unsigned int reg value;
      unsigned int value;
      himci writel(0x1010000, REG UPDATE MCI BASE + MMC UHS REG EXT);
      /* set detect polarity */
      reg value = himci readl(SYS PERIPHCTRL4);
      reg value &= ~SDIO0 DET MODE;
      himci writel(reg value, SYS PERIPHCTRL4);
      /* set clk polarity, mmc clk */
      reg value = 0;
```



```
reg value = himci readl(PERI CRG49);
     reg value &= ~(SDIOOCLK PCTRL);
     reg value &= ~(SDIOO CLK BIT HIGH);
     reg value &= ~(SDIO0 CLK BIT LOW);
     reg value |= SDIOO CKEN;
     himci_writel(reg_value, PERI_CRG49);
}
static void hi mci ctr reset(void)
{
     unsigned int reg value;
     reg value = himci readl(PERI CRG49);
     reg value |= SDIO RESET;
     himci writel(reg value, PERI CRG49);
}
static void hi_mci_ctr_undo_reset(void)
     unsigned int reg value;
     reg value = himci readl(PERI CRG49);
     reg value &= ~(SDIO RESET);
     himci_writel(reg_value, PERI_CRG49);
}
修改./drivers/mmc/himciv100.c 增加编译控制代码,红色部分为增加
#ifdef CONFIG HIMCI HI3518
#include "himciv100 3518.c"
#endif
#ifdef CONFIG_HIMCI_HI3516a
#include "himciv100 3516a.c"
完善 USB OHCI 驱动
   增加 drivers/usb/host/hiusb/hiusb-3516a.c 文件,实现 Hi3516A 平台下 USB OHCI 模
   块驱动差异性部分
#define HIUSB OHCI BASE
                       0x100a0000
#define HIUSB OHCI DEV NAME
                        "hiusb-ohci"
#define PERI CRG46
                      (CRG REG BASE + 0xb8)
#define USB CKEN
                      (1 << 7)
#define USB CTRL UTMI1 REG
                       (1 << 6)
```



```
#define USB CTRL UTMIO REG
                              (1 << 5)
#define USB CTRL HUB REG
                              (1 << 4)
#define USBPHY PORT1 TREQ
                              (1 << 3)
#define USBPHY PORTO TREQ
                              (1 << 2)
#define USBPHY REQ
                            (1 << 1)
#define USB AHB SRST REQ
                             (1 << 0)
#define PERI USB
                             (0x20120078)
#define WORDINTERFACE
                             (1 << 0)
#define SS BURST16 EN
                             (1 << 9)
#define USBOVR P CTRL
                              (1 << 17)
#define MISC USB
                             (0x20120080)
static void hiusb_ohci_enable_clk(void)
      int req;
      /* enable clock to EHCI block and HS PHY PLL */
      reg = readl(PERI CRG46);
      reg |= USB_CKEN;
      reg &= ~(USB CTRL UTMI1 REG);
      reg &= ~(USB_CTRL_UTMIO_REG);
      reg &= ~(USB_CTRL_HUB_REG);
      reg &= ~(USBPHY PORT1 TREQ);
      reg &= ~(USBPHY PORTO TREQ);
      reg &= ~(USBPHY REQ);
      reg &= ~(USB AHB SRST REQ);
      writel(reg, PERI_CRG46);
      udelay(100);
      /* open phy clk */
      writel(0xc06, MISC USB);
      udelay(10);
      writel(0xc26, MISC_USB);
      udelay(100);
      /* enable phy */
      reg = readl(PERI USB);
      reg &= ~ (WORDINTERFACE);
      /* disable ehci burst16 mode */
      reg &= ~(SS BURST16 EN);
      reg |= USBOVR P CTRL;
      writel(reg, PERI USB);
      udelay(100);
```



```
static void hiusb_ohci_disable_clk(void)
      int reg;
      /* Disable EHCI clock.
      If the HS PHY is unused disable it too. */
      reg = readl(PERI CRG46);
      reg &= ~(USB_CKEN);
      reg |= (USB CTRL UTMI1 REG);
      reg |= (USB_CTRL_UTMI0_REG);
      reg |= (USB_CTRL_HUB_REG);
      reg |= (USBPHY PORT1 TREQ);
      reg |= (USBPHY PORTO TREQ);
      reg |= (USBPHY REQ);
      reg |= (USB_AHB_SRST_REQ);
      writel(reg, PERI CRG46);
      udelay(100);
      /* enable phy */
      reg = readl(PERI USB);
      reg |= (WORDINTERFACE);
      reg |= (SS BURST16 EN);
      reg |= (USBOVR P CTRL);
      writel(reg, PERI USB);
      udelay(100);
}
修改 drivers/usb/host/hiusb/hiusb-ohci.c 增加编译控制代码,红色部分为增加
/*
* low level initalisation routine, called from usb.c
static char ohci inited;
#ifdef CONFIG HI3535
#include "hiusb-3535.c"
#endif
#ifdef CONFIG HI3516A
#include "hiusb-3516a.c"
#endif
   修改 board/hi3516a/board.c 增加自动升级功能模块在 U-boot 中的入口,红色部分
   为增加
int misc init r(void)
{
```



```
#ifdef CONFIG_RANDOM_ETHADDR
     random_init_r();
#endif
     setenv("verify", "n");
#ifdef CONFIG_AUTO_UPDATE
                              //系统升级功能总开关
     extern int do_auto_update(void);
#ifdef CFG_MMU_HANDLEOK
     dcache_stop();
#endif
     do_auto_update(); //升级函数入口
#ifdef CFG_MMU_HANDLEOK
     dcache_start();
#endif
#endif /* CONFIG_AUTO_UPDATE */
     return 0;
}
```