



www.enjoylinux.cn

UBOOT

谢伟 著

版权声明：本课件及其印刷物、视频的版权归成都国嵌信息技术有限公司所有，并保留所有权力：任何单位或个人未经成都国嵌信息技术有限公司书面授权，不得使用该课件及其印刷物、视频从事商业、教学活动。已经取得书面授权的，应在授权范围内使用，并注明“来源：国嵌”。违反上述声明者，我们将追究其法律责任。

Contents



BootLoader介绍

Uboot介绍

Uboot命令

Uboot工作流程

Uboot移植



Contents



BootLoader介绍

Uboot介绍

Uboot命令

Uboot工作流程

Uboot移植



概念

什么是BootLoader?



软件层次



www.enjoylinux.cn

一个嵌入式系统从软件角度来看分为三个层次：

1. 引导加载程序

包括固化在固件(firmware)中的 boot 程序(可选)，和 **BootLoader** 两大部分。

2. Linux 内核

特定于嵌入式平台的定制内核。

3. 文件系统

包括了系统命令和应用程序。

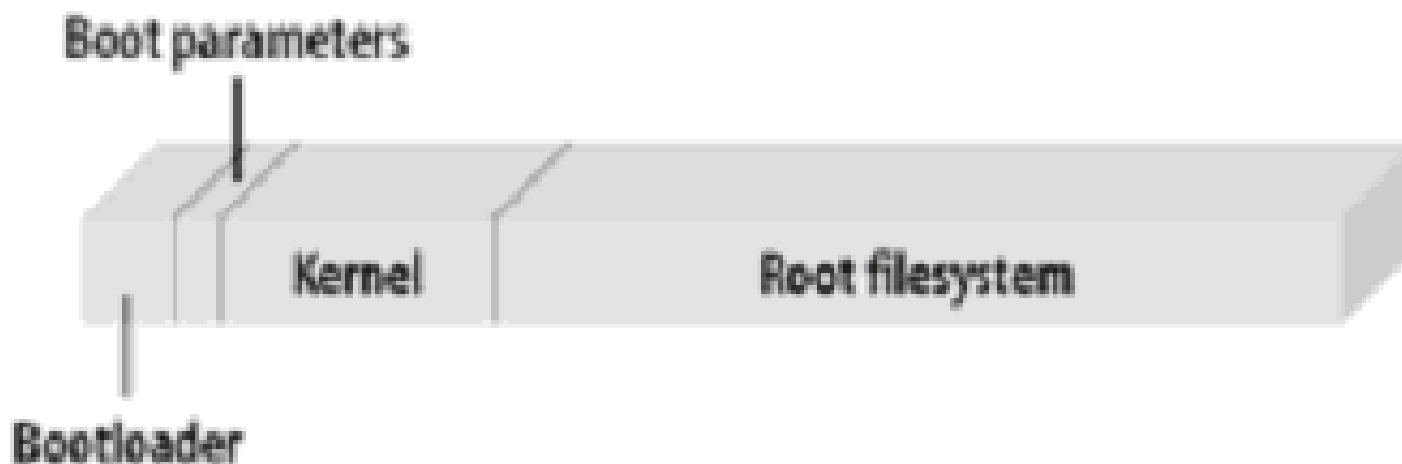


软件层次



www.enjoylinux.cn

一个同时装有 **BootLoader**、内核的启动参数、内核映像和根文件系统映像的固态存储设备的典型空间分配结构图：



回忆PC



www.enjoylinux.cn



PC机中的引导加载程序由**BIOS**（其本质是一段固件程序）和**GRUB或LILO**一起组成。**BIOS**在完成硬件检测和资源分配后，将硬盘中的引导程序读到系统内存中然后将控制权交给引导程序。引导程序的主要任务是将内核从硬盘上读到内存中，然后跳转到内核的入口点去运行，即启动操作系统。



定义



www.enjoylinux.cn

在嵌入式系统中，通常没有像BIOS那样的固件程序，因此整个系统的加载启动任务就完全由BootLoader来完成。比如在一个基于 ARM7TDMI core的嵌入式系统中，系统在上电或复位时都从地址 0x00000000开始执行。而在这个地址处安排的通常就是系统的BootLoader程序。



定义



www.enjoylinux.cn

简单地说，**BootLoader**就是在操作系统运行之前运行的一段小程序。通过这段小程序，可以**初始化硬件设备**，从而将系统的软硬件环境带到一个合适的状态，以便为最终**调用操作系统**做好准备。



安装



www.enjoylinux.cn



系统加电或复位后，所有的**CPU**通常都从**CPU**制造商预先安排地址开始执行。比如，**S3C2410**在复位后从地址**0x00000000**起开始执行。而嵌入式系统则将固态存储设备（比如：**FLASH**）安排在这个地址上，而**bootloader**程序又安排在固态存储器的最前端，这样就能保证在系统加电后，**CPU**首先执行**BootLoader**程序。



移植



为什么需要进行bootloader移植?



移植



每种不同的**CPU**体系结构都有不同的
BootLoader。除了依赖于**CPU**的体系结构外，
BootLoader 还依赖于**具体的嵌入式板级设备的配置**，比如板卡的硬件地址分配，外设芯片的类型等。这也就是说，对于两块不同的开发板而言，**即使它们是基于同一种CPU而构建的，但如果他们的硬件资源或配置不一致的话，要想在一块开发板上运行的BootLoader程序也能在另一块板子上运行，还是需要作修改。**



流程



www.enjoylinux.cn

BootLoader 的启动过程可分为**单阶段**（Single-Stage）和**多阶段**（Multi-Stage）两种，通常多阶段的 BootLoader 具有更复杂的功能，更好的可移植性。从固态存储设备上启动的 BootLoader 大多**采用两阶段**，即启动过程可以分为 stage 1 和 stage 2：stage 1 完成初始化硬件，为 stage 2 准备内存空间，并将 stage 2 复制到内存中，设置堆栈，然后跳转到 stage 2。



流程



www.enjoylinux.cn

BootLoader 的 stage1 通常包括以下步骤:

- 硬件设备初始化
- 为加载 BootLoader 的 stage2 准备 RAM 空间
- 拷贝 BootLoader 的 stage2 到 RAM 空间中
- 设置好堆栈 (why??)
- 跳转到 stage2 的 C 入口点



流程



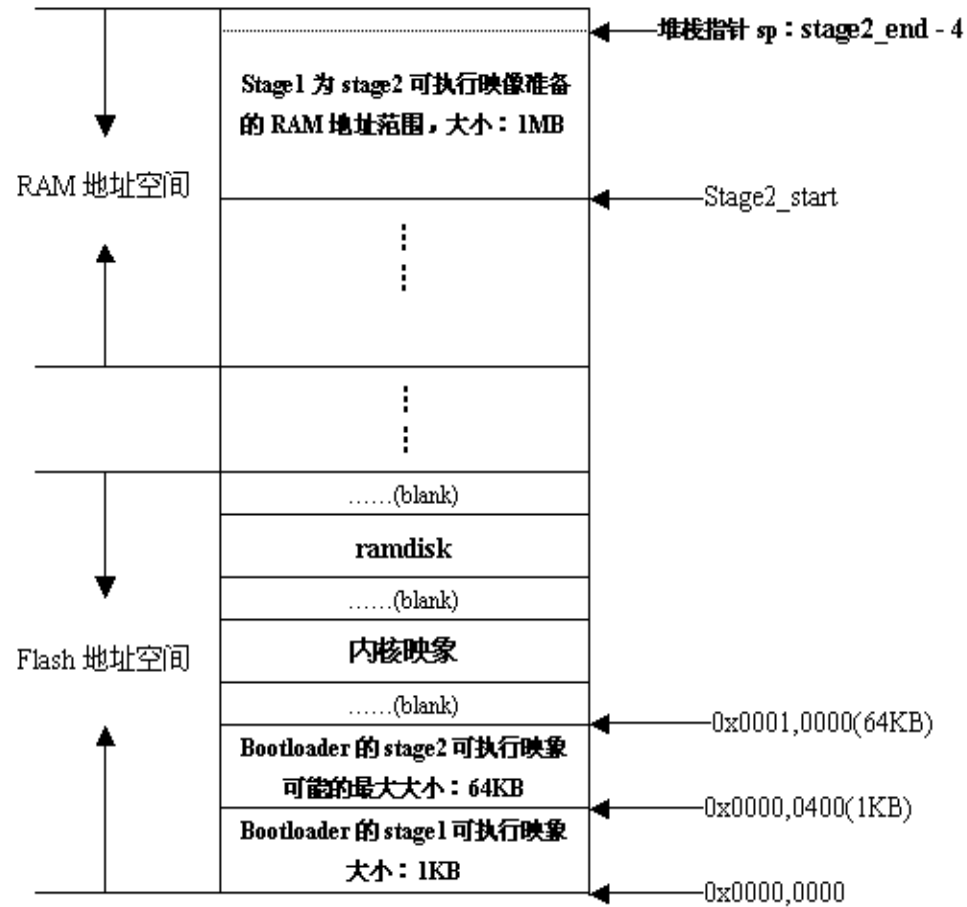
www.enjoylinux.cn

BootLoader 的 **stage2** 通常包括以下步骤:

- 初始化本阶段要使用到的硬件设备
- 将内核映像和根文件系统映像从 flash 上读到
RAM 中
- 调用内核



内存分布



Contents



BootLoader介绍

Uboot介绍

Uboot命令

Uboot工作流程

Uboot移植



作用



Uboot是德国**DENX**小组开发的用于**多种嵌入式CPU（MIPS、x86、ARM、XScale等）**的**bootloader**程序，**UBoot**不仅支持嵌入式**Linux**系统的引导，还支持**VxWorks, QNX等多种嵌入式操作系统**。



下载



www.enjoylinux.cn

从下面地址可以下载到uboot的源代码:

<ftp://ftp.denx.de/pub/u-boot/>



目录结构



进入到**UBOOT**目录，可以得到如下的**目录结构**：

- |-- board
- |-- common
- |-- cpu
- |-- disk
- |-- doc
- |-- drivers
- |-- dtb
- |-- examples
- |-- fs
- |-- include



目录结构



www.enjoylinux.cn

- |-- lib_arm
- |-- lib_generic
- |-- lib_i386
- |-- lib_m68k
- |-- lib_microblaze
- |-- lib_mips
- |-- lib_nios
- |-- lib_nios2
- |-- lib_ppc
- |-- net
- |-- post
- |-- rtc
- |-- tools



目录结构



www.enjoylinux.cn

✓ Board

和开发板有关的文件。每一个开发板都以一个子目录出现在当前目录中，比如：**SMDK2410**，子目录中存放与开发板相关的文件。

✓ Common

实现Uboot支持的命令。

✓ Cpu

与特定**CPU**架构相关的代码，每一款Uboot下支持的**CPU**在该目录下对应一个子目录，比如有子目录**arm920t**等。



目录结构



✓ Disk

对磁盘的支持。

✓ Doc

文档目录。Uboot有非常完善的文档，推荐大家参考阅读。

✓ Drivers

Uboot支持的设备驱动程序都放在该目录，比如各种网卡、支持CFI的Flash、串口和USB等。



目录结构



✓Fs

文件系统的支持。

✓Include

Uboot使用的头文件。该目录下**configs**目录有与开发板相关的配置头文件，如**smdk2410.h**。该目录下的**asm**目录有与CPU体系结构相关的头文件。



目录结构



✓Net

与网络协议栈相关的代码，例如：**TFTP**协议、**RARP**协议的实现。

✓Tools

生成Uboot的工具，如：**mkimage**, **crc**等等。



编译



www.enjoylinux.cn

Uboot的Makefile从**功能上**可以分成两个部分:

- 1、执行每种board相关的配置
- 2、编译生成uboot.bin文件



编译



www.enjoylinux.cn

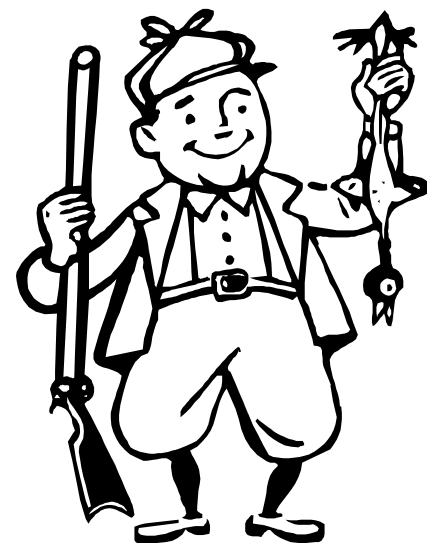
Uboot.bin的生成也分为两步，以smdk2410为例来说明，如下：

1. 对于board进行配置：

```
$make smdk2410_config
```

2. 进行编译生成u-boot.bin:

```
$make CROSS_COMPILE=arm-linux-
```



Contents



BootLoader介绍

Uboot介绍

Uboot命令

Uboot工作流程

Uboot移植



常用命令



www.enjoylinux.cn

尽管**UBOOT**提供了丰富的命令集，但不同的单板所支持的命令并不一定一样（**可配置，How?后面章节**），**help** 命令可用于察看当前单板所支持的命令。

【2410】 # help

autoscr -run script from memory

base -print or set address offset

bdinfo -print Board Info structure

boot -boot default,i.e.,run 'bootcmd'

bootm -boot application image from memory



环境变量相关



printenv 查看环境变量

usage:

printenv

- print values of all environment variables

printenv name ...

- print value of environment variable 'name'

Uboot> printenv

ipaddr=192.168.1.1

ethaddr=12:34:56:78:9A:BC

serverip=192.168.1.5



环境变量相关



setenv 添加、修改、删除环境变量

✓ **setenv** name value ...

- **set** environment variable 'name' to 'value ...'

✓ **setenv** name

- **delete** environment variable 'name'

```
Uboot> setenv myboard AT91RM9200DK
```

```
Uboot> printenv
```

```
serverip=192.168.1.5
```

```
myboard=AT91RM9200DK
```



环境变量相关



saveenv 保存环境变量

将当前定义的所有变量及其值存入flash中。



文件下载



www.enjoylinux.cn

tftp 通过网络下载文件

注意：使用tftp，需要先配置好网络

```
Uboot> setenv ethaddr 12:34:56:78:9A:BC
```

```
Uboot> setenv ipaddr 192.168.1.1
```

```
Uboot> setenv serverip 192.168.1.254 （tftp服务器的地址）
```

例：

```
Uboot> tftp 32000000 ulmage
```

把server（IP=环境变量中设置的serverip）中服务目录下的
ulmage通过TFTP读入到0x32000000处。



内存操作



www.enjoylinux.cn

md 显示内存区的内容。

md采用十六进制和**ASCII**码两种形式来显示存储单元的内容。

这条命令还可以采用长度标识符 **.l**, **.w**和**.b** :

md [.b, .w, .l] address

md.w 100000

00100000: 2705 1956 5050 4342 6f6f 7420 312e 312e

00100010: 3520 284d 6172 2032 3120 3230 3032 202d



内存操作



www.enjoylinux.cn

mm 修改内存，地址自动递增。

mm [.b, .w, .l] address

mm 提供了一种互动修改存储器内容的方法。它会显示地址和当前值，然后提示用户输入。如果你输入了一个合法的十六进制数，这个新的值将会被写入该地址。然后提示下一个地址。如果你没有输入任何值，只是按了一下回车，那么该地址的内容保持不变。如果想**结束输入**，则**输入空格，然后回车**。

=> mm 100000

00100000: 27051956 ? 0

00100004: 50504342 ? AABBCDD



Flash操作



www.enjoylinux.cn

flinfo 查看Flash扇区信息

Usage: Uboot> flinfo



Flash操作



www.enjoylinux.cn

protect Flash写保护

打开或关闭扇区写保护

用法:

protect off all

关闭所有扇区的写保护

protect on all

打开所有扇区的写保护

protect off start end

关闭从**start** 到 **end** 扇区的写保护(**start**为要关闭的第1个扇区的起始地址, **end**为要关闭的最后一个扇区的结束地址)

protect on start end

打开从**start** 到 **end** 扇区的写保护



Flash操作



www.enjoylinux.cn

erase 擦除flash扇区

用法: `erase start end`

擦除从start 到 end 的扇区，start 为要擦除的第1个扇区的起始地址，end 为要擦除的最后一个扇区的结束地址(在使用cp命令向Nor型Flash写入数据之前必须先使用erase 命令擦除flash，因为nor flash 按字节写入时，无法写入1，所以必须通过擦除来写入1)。

例: `erase 30000 1effff`

。



Flash操作



www.enjoylinux.cn

cp 数据拷贝。

cp [.b, .w, .l] saddress daddress len

cp 提供了一种内存与内存，内存与Flash之间数据拷贝的方法。

例：

cp.b 31000000 50000 d0000

将内存地址0x31000000处的数据（长度为0xd0000）拷贝到地址0x50000处（Flash中）

cp.b 32000000 120000 c0000

将内存地址0x32000000处的数据（长度为0xc0000）拷贝到地址0x120000处（Flash中）



执行程序



www.enjoylinux.cn

go 执行内存中的二进制代码，一个简单的
跳转到指定地址

go addr [arg ...]

- start application at address 'addr',
passing 'arg' as arguments



执行程序



www.enjoylinux.cn

bootm 执行内存中的二进制代码

bootm [addr [arg ...]]

- boot application image stored in memory
passing arguments 'arg ...'; when booting a
Linux kernel, 'arg' can be the address of an
initrd image

要求二进制代码有固定格式的文件头。



开发板信息



bdinfo – 显示开发板信息

bdinfo命令（简写为**bdi**）将在终端显示诸如**内存地址和大小、时钟频率、MAC地址**等信息。这些信息在传递给Linux内核一些参数时可能会用到。



自动启动



1. 设置自动启动

```
mini2440=>setenv bootcmd tftp  
31000000 ulmage \; bootm 31000000  
mini2440 =>saveenv
```



实验



www.enjoylinux.cn

Uboot命令

使用该小节介绍的命令



Contents



BootLoader介绍

Uboot介绍

Uboot命令

Uboot工作流程

Uboot移植



工作模式



大多数BootLoader都包含两种不同的操作模式：“启动模式”和“下载模式”，这种区别仅对于开发人员才有意义，但从最终用户的角度看，BootLoader的作用就是用来加载操作系统，而不存在所谓的启动模式与下载模式。



启动模式



这种模式也称为“自主”模式，是指 **BootLoader** 从目标机上的某个固态存储设备上将操作系统自动加载到 **RAM** 中运行，整个过程并没有用户的介入。这种模式是 **BootLoader** 的正常工作模式，因此在嵌入式产品发布的时候，**BootLoader** 显然必须工作在这种模式下。



下载模式



www.enjoylinux.cn

在这种模式下，目标机上的
BootLoader 将通过串口或网络等通
信手段从主机（**Host**）下载文件，然
后控制启动流程。



代码导读



参考文档 《uboot启动流程》



Contents



BootLoader介绍

Uboot介绍

Uboot命令

Uboot工作流程

Uboot移植



软硬件配置



为什么需要对Uboot进行移植?

BootLoader 依赖于:

具体的CPU体系、具体的板级设备配置
(芯片级移植、板级移植)



软硬件配置



www.enjoylinux.cn

具体的板级设备的配置在哪里？

板级设备的配置文件位于

include/configs/<board_name>.h

<board_name>用相应的**BOARD**定义代替（例：
smdk2410.h）



Smdk2410.h



```
#define CONFIG_ARM920T      1
```

```
/* CPU 类型 */
```

```
#define CONFIG_S3C2410    1
```

```
/* MCU类型 */
```

```
#define CONFIG_SMDK2410    1
```

```
/* 开发板型号 */
```

```
#define USE_920T_MMU    1
```

```
/* 使用MMU */
```



Smdk2410.h



```
#undef CONFIG_USE_IRQ
```

```
/* 不使用 IRQ/FIQ */
```

```
/* malloc 池大小*/
```

```
#define CFG_MALLOC_LEN (CFG_ENV_SIZE +  
128*1024)
```

```
/* 数据段大小 128字节 */
```

```
#define CFG_GBL_DATA_SIZE 128
```



Smdk2410.h



/* 使用 CS8900 网卡 */

#define CONFIG_DRIVER_CS8900 1

/* CS8900A 基地址 */

#define CS8900_BASE 0x19000300

/* 使用串口1 */

#define CONFIG_SERIAL1 1

/* 波特率 */

#define CONFIG_BAUDRATE 115200



Smdk2410.h



```
#define CONFIG_COMMANDS \
```

```
(CONFIG_CMD_DFL      |\
CFG_CMD_CACHE        |\
/*CFG_CMD_NAND       /*|\
/*CFG_CMD_EEPROM     /*|\
/*CFG_CMD_I2C         /*|\
/*CFG_CMD_USB         /*|\
CFG_CMD_REGINFO      |\
CFG_CMD_DATE         |\
CFG_CMD_ELF)
```

*/*定义使用的命令,可添加额外命令,如PING*/*



Smdk2410.h



www.enjoylinux.cn

/* 自动启动等待时间 */

#define CONFIG_BOOTDELAY 3

/* 内核启动参数 */

#define CONFIG_BOOTARGS

"root=ramfs devfs=mount console=ttySA0,9600"

#define CONFIG_ETHADDR 08:00:3e:26:0a:5b

#define CONFIG_NETMASK 255.255.255.0

#define CONFIG_IPADDR 10.0.0.110

#define CONFIG_SERVERIP 10.0.0.1



Smdk2410.h



```
#define CONFIG_BOOTCOMMAND    "tftp; bootm"
```

```
#define CFG_PROMPT            "SMDK2410 # "
```

```
#define PHYS_SDRAM_1  0x30000000 /* SDRAM Bank #1 */
```

```
#define PHYS_SDRAM_1_SIZE      0x04000000 /* 64 MB */
```

```
#define CFG_LOAD_ADDR  0x33000000
```

```
/* 默认的启动地址 */
```

```
#define CFG_BAUDRATE_TABLE    { 9600, 19200, 38400, 57600,  
    115200 } /*可用的波特率*/
```



Smdk2410.h



/* 有一片 SDRAM */

#define CONFIG_NR_DRAM_BANKS 1

/* FLASH No1的基地址 */

#define PHYS_FLASH_1 0x00000000

/* FLASH 的基地址 */

#define CFG_FLASH_BASE PHYS_FLASH_1



移植



www.enjoylinux.cn

怎么做Uboot的移植呢？



技术咨询QQ: 550491596 1327229087 技术咨询电话: 028-88820953 028-66501487

移植方法



www.enjoylinux.cn

开始移植之前，首先要**分析U-Boot已经支持的开发板，选择出硬件配置最接近的开发板**。选择的原则是，首先选择**MCU相同**的开发板，如果没有，则选择**MPU相同**的开发板。



移植范例



www.enjoylinux.cn

以mini2440开发板为例，该开发板采用s3c2440芯片。根据选择原则，首先选择MCU为s3c2440的开发板，但 UBoot 各版本均不支持采用s3c2440芯片的开发板。因此根据第二原则，选择MPU相同，即ARM核为arm920T的开发板，Uboot支持SMDK2410开发板,并且SMDK2410采用s3c2410芯片，s3c2410采用的正好是arm920T，因此选取SMDK2410开发板作为移植参考板。



移植步骤



移植U-Boot的基本步骤如下:

1. 在顶层Makefile中为开发板添加新的配置选项, 使用已有的配置项目为例

```
smdk2410_config :    unconfig
    @./mkconfig $(@:_config=) arm arm920t smdk2410
    NULL s3c24x0
```

参考上面2行, 添加下面2行:

```
mini2440_config :    unconfig
    @./mkconfig $(@:_config=) arm arm920t mini2440 NULL
    s3c24x0
```



移植步骤



www.enjoylinux.cn

arm: CPU 架构

arm920t: CPU 类型，对应cpu/arm920t目录

mini2440: 开发板型号，对应board/mini2440目录

NULL: 开发者

s3c24x0: 片上系统(SOC)



移植步骤



www.enjoylinux.cn

2. 在board目录中创建一个属于新开发板的目录，并添加文件：

```
mkdir -p board/mini2440
```

```
cp -rf board/smdk2410/* board/mini2440
```



移植步骤



www.enjoylinux.cn

3. 为开发板添加新的配置文件

先复制参考开发板的配置文件，**再修改**。例如：

```
$cp include/configs/smdk2410.h include/configs/mini2440.h
```

4. 配置开发板

```
$ make mini2440_config
```

5. 编译U-Boot

执行**make CROSS_COMPILE=arm-linux-**命令，编译成功可以得到U-Boot映像。



移植步骤



6.烧写 Uboot

参考<<友善之臂mini2440使用手册>>



实验一



www.enjoylinux.cn

移植Uboot

移植Uboot到mini2440板

- ✓ 具备Ping命令
- ✓ Nor flash写入支持



实验二



www.enjoylinux.cn

Uboot自主模式设置

- ✓ 将内核、ramdisk通过uboot写入flash
- ✓ 修改uboot相关参数，将uboot设置为自主模式，能使用flash中的内核、ramdisk自动启动

