君正 **USB Boot** 工具用户手册

Revision: 1.0
Date: Oct. 2010

北京君正集成电路有限公司
Ingenic Semiconductor Co. Ltd

君正 **USB Boot** 工具用户手册

Release history

Date	Revision	Change
2010.10.14	1.0	Created.

Disclaimer

This documentation is provided for use with Ingenic products. No license to Ingenic property rights is granted. Ingenic assumes no liability, provides no warranty either expressed or implied relating to the usage, or intellectual property right infringement except as provided for by Ingenic Terms and Conditions of Sale.

Ingenic products are not designed for and should not be used in any medical or life sustaining or supporting equipment.

All information in this document should be treated as preliminary. Ingenic may make changes to this document without notice. Anyone relying on this documentation should contact Ingenic for the current documentation and errata.

北京君正集成电路股份有限公司

北京市海淀区东北旺西路 8 号中关村软件园信息中心 A 座 108 室
Tel: 86-10-82826661
Fax: 86-10-82825845
Http: //www.ingenic.cn

内容

1 概述 1

2 名称约定 3

3 使用准备 5

3.1 基本准备 5
3.2 配置文件 5
3.3 从USB启动 11
3.4 主机安装驱动程序 11

4 USB BOOT工具功能 13

4.1 复位状态与BOOT状态 13
4.2 烧录功能 13
4.3 测试功能 14

5 烧录实例 15

5.1 LEPUS板 15
5.2 烧录其它文件系统 15

6 烧录原理简介 17

6.1 BOOT流程介绍 17

1 概述

USB Boot 工具是北京君正集成电路有限公司研发的具有特色的开发、生产工具。可用于产品研发阶段构建开发环境，也可用于产品出厂时的大批量烧录和产品售后升级。USB Boot 工具将尽最大的努力不断升级完善，为用户提供更稳定、更便捷的功能，来进一步提高君正处理器平台的易用性。

USB Boot 工具有以下特点：

- 基于 USB 来传输数据，具有较高的数据传输速度和烧录速度；
- 工作条件简单；
- 一台主机可以同时操作多个设备；
- 支持君正处理器芯片 Jz4740、Jz4750、Jz4750d(Jz4755)和 Jz4760；
- 稳定支持页大小为 512/2048/4096 的 Nand Flash；
- 支持两种基本的开发板测试功能；

使用者请务必留意本文档的说明，尽量按照约定使用，从而减少因为误操作带来的麻烦，提高用户开发效率，也减少君正公司对重复问题支持的精力和时间。

2 名称约定

以下是本文使用的一些名称和术语的含义，只列举本文单独使用的名称，其他专业术语与原意相同。

主机：运行着 windows XP/2K 的兼容微型计算机，要求带有 USB 接口。

设备：搭载了支持 USB Boot 功能的君正处理器的开发板、产品板或者最小系统。

数据区：这是从驱动程序角度划分的区域，对应与 Nand Flash 物理结构中的一个区域。该区域的大小由 Nand Flash 的结构决定。驱动程序用这个区域存放实际数据。

OOB 区：与数据区相对应，可以理解为紧跟在数据区后面的一个小的区域，大小与数据区有固定的比例关系。用来存放由实际数据产生的 ECC 信息和文件系统的额外信息。

第一阶段：主机方程序对设备 PLL，SDRAM 和串口进行初始化的阶段，对应的代码成为第一阶段代码，在 cache 中运行一次然后返回。

第二阶段：能够响应用户命令，完成烧录、测试等实际功能的阶段，对应的代码成为第二阶段代码，运行在 SDRAM 中，是常驻的服务程序。

Boot 操作：指使设备进入可以交互状态的操作。实际就是主机方程序让第二阶段代码在设备上正常运行的整个过程。

复位状态：设备从 usb device 复位后的状态。

Boot 状态：设备在复位状态进行 Boot 操作之后的状态。

文件类型：指执行烧录命令时，待烧录文件的数据组织结构。这个结构与 Nand Flash 的存储结构相对应，分为无 OOB、有 OOB 无 ECC 和有 OOB 有 ECC 三种。

3 使用准备

3.1 基本准备

USB Boot 工具可以工作在 Windows XP/2000 平台上，主机 USB Host 接口支持 2.0 和 1.1。USB Boot 工具包括以下文件。

- USBbootTool.exe，这是运行在主机上的 MFC 应用程序，用户通过该程序获得所需要的功能。
- fw.bin，这是设备方第一阶段的可执行代码。进行 Boot 操作的初期，主机方的程序会把该文件里的代码通过 USB 发送到设备上执行。
- usb_boot.bin，这是设备方第二阶段代码，这部分代码实现烧录等具体功能。Boot 操作的后期，主机方的程序会把该文件里的代码通过 USB 发送到设备上执行。
- Usb_Boot_Driver.sys，Windows 驱动程序的执行代码，只有安装了这个驱动之后，USB Boot 的功能才能正常使用。
- Usb_Boot_Driver.inf，Windows 驱动程序配置信息，与 Usb_Boot_Driver.sys 配套。

除了以上资源，使用 USB Boot 工具还需要准备 USB 电缆和配置文件。

3.2 配置文件

3.2.1 配置说明

配置文件非常重要，主机方的程序通过配置文件获取大部分关键信息，如果配置文件设置错误，必然导致之后的操作产生错误。

用户请格外注意，配置文件必须随时根据设备的变化修改，在执行操作之前，必须保证配置文件中的信息与设备开发板的具体情况一致。

配置文件分别放在 Release/4725B/、Release/4740/、Release/4750/、Release/4755/、Release/4760/ 和 Release/tool_cfg 文件夹下。根据 CPU 型号修改相应文件夹下的配置文件即可。

配置文件的**模板**如下：

Release/4760/HardwareCfg.ini

```
;-------------------------------------------------
; USB boot configuration file for Jz47XX board
;-------------------------------------------------
```

```
[PLL]
EXTCLK=12                ;Define the external crystal in MHz
CPUSPEED=240             ;Define the PLL output frequency
PHMDIV=3                 ;Define the frequency divider ratio of PLL=CCLK:PCLK=HCLK=MCLK
BOUDRATE=115200              ;Define the uart boudrate
USEUART=1                ;Use which uart, 0 for jz4725B,1 for jz4755

[SDRAM]
BUSWIDTH=32              ;The bus width of the SDRAM in bits (16|32)
BANKS=4                  ;The bank number (2|4)
ROWADDR=12               ;Row address width in bits (11-13)
COLADDR=9                ;Column address width in bits (8-12)
ISMOBILE=1               ;Define whether SDRAM is mobile SDRAM, this only valid for
Jz4750 ,1:yes 0:no
ISBUSSHARE=0             ;Define whether SDRAM bus share with NAND 1:shared 0:unshared
DRAMTYPE=2                   ;Define SDRAM Type   0:sdram  1:ddr1    2:ddr2    3:mobile ddr
```

Release/4760/LinuxNandCfg.ini

```
[NAND]
BUSWIDTH=8                       ;The width of the NAND flash chip in bits (8|16|32)
ROWCYCLES=3                      ;The row address cycles (2|3)
FORCEERASE=0                 ;The force to erase flag (0|1)
PAGENUMBER=65536         ;The page number of the entire nand chip
ECCPOS=24                        ;ecc position
BADBLACKPOS=0                    ;bad black ID position
BADBLACKPAGE=127             ;bad black ID page
PLANENUM=1                       ;The planes number of target nand flash
BCHBIT=4                         ;Specify the hardware BCH algorithm for 4750 (4|8)
WPPIN=0                              ;Specify the write protect pin number
BLOCKPERCHIP=0               ;Specify the block number per chip,0 means ignore

BCHSTYLE=0                       ;BCH style,uCOS use 1, Linux use 0
[END]
```

Release/tool_cfg/LinuxFileCfg.ini

```
# 烧录文件位置设定，可设定：文件使用数、结束页地址、待烧录文件起始页地址
# 待烧录文件名、烧录文件 Flash 校验方式。
# ******请根据实际调整参数**********
#    NandOption          OOB_ECC      = 0,          //对应命令行参数“-o”
#                                     OOB_NO_ECC = 1,
#                                     NO_OOB       = 2,          //对应命令行参数“-n”
#                                     BOOT_BIN     = 3
#
```

[Fileset]
MaxFileCount=4 ;最大烧录
FileAutoFilter=1 ;是否打开文件自动匹配
EndPage=524280 ;烧录结束也地址

NandInfo1=61 ;未使用
NandInfo2=62 ;未使用
NandUID=63 ;未使用

FileName 烧录 bin 文件名，会显示于烧录界面，或用于按文件文件名自动匹配。
[File1]
FileName=u-boot-nand.bin
StartPage=0 ;文件起始地址，烧录 NAND 时此项表示页号，烧录 SD 卡时此项表示扇区号
NandOption=2 ;ECC 校验方式，烧录 SD 卡忽略此项

[File2]
FileName=uImage
StartPage=19456
NandOption=2

[File3]
FileName=system.img
StartPage=25600
NandOption=0

[File4]
FileName=userdata.img
StartPage=107520
NandOption=0

[File5]
FileName=minios.bin
StartPage=524288
NandOption=2

[File6]
FileName=res.bin
StartPage=524288
NandOption=2

[File7]
FileName=demo.bin

StartPage=0
NandOption=2

[Version]
Cur=(Linux 2.0.1)

Release/tool_cfg/Setting.ini

烧录历史记录文件。
CpuType: 0 - 25\40,1 - 4750 ， 2 - 4755 ， 3 - 4725B , 4 - 4760#
SystemUsed: 0 - miniCos ,1 - Linux , 2 - Other
StorageMedia: 0 – NAND, 1 – SD/MMC/iNAND
ForceErase/Format: 1 - Used ,0 - Unused
DDRType: 1 - mddr, 2 – ddr1, 3 – ddr2, 0 – others

[UserSetting]
CpuType=0
SystemUsed=1
StorageMedia=0
ForceErase=0
Format=0
ProgrammeCount=0
DDRType=3

注：

1. 如果打开 USBbootTool.exe 时弹出选择产品型号对话框，可以不选择任何产品，直接点击确定，否则配置文件可能被修改。
2. 烧录时需要 tool_cfg 和以相应 cpu 命名的文件夹，如烧录 4760 时需要 tool_cfg/和 4760/文件夹。4760 文件夹下的 fw_mddr.bin/fw_ddr.2bin/fw_ddr1.bin 和 usb_boot.bin 分别在/fw_usbboot/4760/stage1/fw 和/fw_usbboot/4760/stage2 下编译得到。
3. fw_mddr.bin 包含了 mobile ddr 的初始化，fw_ddr2.bin 包含了 ddr2 的初始化。如果开发板上使用 ddr2，将 fw_usbboot/4760/stage1/fw/文件夹下编译好的 fw_ddr2.bin 放在 Release/4760/文件夹下，Release/tool_cfg/Setting.ini 文件中 DDRType=3。
4. 如果需要烧录 4760B，创建 4760B 文件夹，将 HardwareCfg.ini、LinuxNandcfg.ing、MiniOSNandcfg.ini、OtherNandcfg.int、usb_boot.bin 和 fw.bin 拷贝到文件夹下，修改相应的配置即可。
5. 如果开发板上使用的 ddr 类型与我们开发板的标配类型不一致，可根据 ddr spec 说明修改 mddr.h 或 ddr2.h 文件后重新编译生成 fw_mddr.bin 或 fw_ddr2.bin，放在 Release/4760/文件夹下替换原来的文件。

3.2.2 PLL 段

PLL 段定义频率相关信息：

- EXTCLK：指定当前电路板 CPU 主晶体的频率，对于 Jz4740 和 Jz4760 应该是 12MHz，对于 Jz4750 可以是 12，13，19.2，24，26，27MHz 中的一个。注意，如果主晶体的频率是 19.2MHz，请把 EXTCLK 的值设置为 19。
- CPUSPEED：指定 Boot 之后 CCLK 的工作频率，注意这个频率必须能被 12 和 EXTCLK 整除，否则无法正确升频。
- PHMDIV：指定 Boot 之后 PCLK，HCLK，MCLK 与 CCLK 工作频率的比率。3 则表示 1：3：3：3。例如，EXTCLK=12，CPUSPEED=336，PHMDIV=3，那么 PCLK=HCLK=MCLK=112MHz=1：3=CCLK= CPUSPEED=336MHz。
- BOUDRATE：指定 Boot 之后串口的波特率，如果没有特殊，请将该值设置为 57600。
- USEUART：指定使用哪个串口作为输出。Jz4740 有两个串口，应该是 0 或者 1；Jz4750 有四个串口，可以 0，1，2，3；Jz4755 有三个串口，可以 0，1，2；Jz4760 有一个串口，串口 1。

3.2.3 SDRAM 段

SDRAM 段定义当前开发板 SDRAM 的情况，注意，该段必须正确设置，否则不能成功 Boot。该段的定义将动态反映到 fw.bin 中，如果 SDRAM 的大小改变了，不需要重新编译 fw.bin，只需要修改配置文件，然后重新 Boot 即可。注意，对于 Jz4760 需要重新编译 fw.bin。

- BUSWIDTH：定义 SDRAM 工作的数据宽度，32bit 或者是 16bit。如果实际硬件的 SDRAM 是 32bit 方式，BUSWIDTH 设为 32 和 16 都能正常工作；如果实际硬件的 SDRAM 是 16bit 方式，BUSWIDTH 必须设为 16。
- BANKS：定义 SDRAM 的 bank 个数，取值为 2 或者 4，这个值请参考所使用 SDRAM 的手册。对于 4bank 的 SDRAM，BANKS 可以设成 4 或者 2，都能正常工作；对于 2bank 的 SDRAM，BANKS 必须设成 2。
- ROWADDR：定义 SDRAM 的 row 地址位数，请参考所使用 SDRAM 的手册。
- COLADDR：定义 SDRAM 的 col 地址位数，请参考所使用 SDRAM 的手册。
- ISBUSSHARE 该项只对 jz4750 有效，指定当前 CPU 是否工作在 bus share 模式。

USB Boot 主机方程序将根据 BUSWIDTH，BANKS ，BANKS，COLADDR 这四个值计算当前设备的 SDRAM 的总大小，然后根据这个值确定第二阶段代码的执行地址。如果计算出来的值超过了设备实际 SDRAM 的大小，将导致第二阶段代码无法执行。计算的公式如下：

SDRAM size = 2^(ROWADDR + COLADDR) * BANKS * (BUSWIDTH / 8)

例如：

BUSWIDTH=32，BANKS=4，ROWADDR=13，COLADDR=9；则 SDRAM size=64MB；
BUSWIDTH=16，BANKS=2，ROWADDR=13，COLADDR=9；则 SDRAM size=16MB。

3.2.4 NAND 段

NAND 段定义与 Nand Flash 相关的信息，这些信息影响烧录过程，请确保这些设置与设备的 Nand Flash 相符合，否则会导致烧录错误。

- BUSWIDTH：定义 Nand 访问的数据宽度，可以是 8bit 或者 16bit 方式。
- ROWCYCLES：定义命令周期数，取值为 2 或者 3，该值请根据 Nand Flash 手册设定。
- PAGESIZE：定义 Nand Flash 的页大小，取值为 512，2048 或者 4096，该值请根据 Nand Flash 手册设定。
- PAGEPERBLOCK：定义 Nand Flash 每一块包含多少个页，取值为 32，64，128，该值请根据 Nand Flash 手册设定。
- FORCEERASE：指定是否强制擦除坏块，如果为 1，则忽略坏块标记擦除所有的块；如果为 0，则只擦除没有坏块标记的块。注意，如果该值为 1，则会丢失在使用过程中产生的坏块记录信息。
- OOBSIZE：定义 Nand Flash 的 OOB 区的大小，取值为 16，64，128。
- ECCPOS：定义 ECC 校验信息存放位置在 OOB 区中的偏移，该值的设置由未来读取 Nand Flash 数据的程序决定，如果不一致，将导致烧完之后不能正常使用。例如，Jz4740 Linux MTD 驱动把 ECC 信息放在 OOB 区中偏移 28 的地方，那么烧录文件系统(yaffs2，vfat)的时候必须把这个值设置为 28。Jz4750 Linux MTD 驱动、Jz4755 Linux MTD 驱动和 Jz4760 Linux MTD 驱动把 ECC 信息放在 OOB 区中偏移 24 的地方，那么烧录文件系统(yaffs2，vfat)的时候必须把这个值设置为 24。细节参考 linux/drivers/mtd/nand_base.c。
- BADBLACKPOS：定义坏块标记在 OOB 区中的偏移，例如，BADBLACKPOS=0 则表示 OOB 区中的第 0 个字节是坏块标记，如果该字节不等于 0xff，则表明这块是坏块。
- BADBLACKPAGE：定义坏块标记在块中的哪一页，例如，BADBLACKPAGE=0 则表示每块的第一页有坏块信息；BADBLACKPAGE=PAGEPERBLOCK 则表示最后页；如果 BADBLACKPAGE>PAGEPERBLOCK，则每块的头尾两页都包含坏块信息。
- PLANENUM ：定义是否使用 two-plane 的方式来烧录。请参考后文。
- BCHBIT：定义 BCH 算法的纠错能力，该值只对 Jz4750 和 Jz4760 有效，可选值为 4 或者 8。4 表示使用能纠 4 个 bit 错误的 BCH 算法，8 则能纠 8 个 bit。这个值的设定要跟操作系统驱动的设定一致。
- WPPIN：定义写保护管脚 GPIO 的编号。这个数值用来确定一个 GPIO，GPA0 的编号为 0，GPB0 的编号为 32，依此类推。如果这个值非 0，当访问 Nand Flash 的时候，程序会让对应的 GPIO 管脚输出低电平，以关闭写保护，访问结束，对应的管脚输出高。
- BLOCKPERCHIP：这个值暂时不使用，忽略。

3.2.5 Fileset 段

Fileset 段定义与待烧录文件相关的信息，这些信息影响烧录过程，请确保这些设置与设备相符合，否则会导致烧录错误。

- MaxFileCount：定义要烧录的文件个数。

- FileName：定义要烧录的文件。

- StartPage：定义烧录起始页。
- NandOption：定义烧录方式。0 表示 OOB_ECC；1 表示 OOB_NO_ECC；2 表示 NO_OOB；3 表示 BOOT_BIN。

3.3 从 USB 启动

使用 USB Boot 工具，开发板需要以 USB Device 的方式启动。启动最初期的代码已经固化在支持 USB Boot 的芯片内部（bootrom），使用时通过跳线设置就可以选择不同的启动方式。Bootrom 中的代码根据复位时 Boot_sel[0:1]的电位状况选择启动方式，当 Boot_sel[0:1] = 01 时，按下复位开关，处理器以 USB Device 的身份启动。操作时具体如何设置 Boot_sel[0:1]请参考具体开发板的原理图，以君正公司提供的 PAVO v1.3 开发板为例，按住 sw5 不放再按 reset 就可以从 USB Device 启动。 Jz4750 APUS v1.1.2，按下 sw6 不放再按 reset 就可以从 USB Device 启动。Jz4755 APUS v1.2，按下 sw10 不放，再向右按 sw8，再按 reset 就可以从 USB Device 启动。Jz4760 LEPUS v1.1，按下 sw8 不放，再按 reset 就可以从 USB Device 启动。

注意只有支持 USB Boot 的芯片才能从 USB Device 启动，Jz4730 不支持 USB Boot 功能，也无法使用本文介绍的所有功能。

后文所说的复位均与此同。

3.4 主机安装驱动程序

以 USB Device 的方式启动的设备第一次连接主机（复位后）需要安装主机方的驱动程序，以后连接不再需要安装驱动。主机方的驱动程序可在工具软件发布包中找到，即 Usb_Boot_Driver.inf 和 Usb_Boot_Driver.sys 这两个文件。安装驱动的方法与其他设备类似，安装成功之后应该能在 windows 的设备管理器中查看到 Usb_Boot 设备。

4 USB Boot 工具功能

经过以上准备，可以开始使用 USB Boot 工具。USB Boot 的功能分为：烧录功能和测试功能。其中每一类命令执行所需要的的条件不同，下面将详细介绍。

4.1 复位状态与 Boot 状态

Boot 是改变设备状态的唯一方法。

设备从 USB 启动并连接到主机之后之后处于复位状态，主机方程序成功执行 Boot 之后，设备处于 Boot 状态。处于 Boot 状态的设备如果要返回复位状态，请将设备复位。

烧录类功能必须工作在 Boot 状态。设备连接主机之后请首先执行 Boot 操作，然后才能烧录。

4.2 烧录功能

设备必须处在 Boot 状态才能使用这类功能。

4.2.1 写 Nand Flash

烧录功能能将三种类型的文件烧录到 Nand Flash 的指定位置中，烧录过程中带有校验。

文件类型的分类：

1） 无OOB：镜像文件中不包括OOB信息，文件内容的组织方式为连续的数据。编译生成的bootloader 和 kernel 属于这种类型。
2） 有 OOB 无 ECC：镜像文件中包括了 OOB 信息，但是没有包含 ECC 信息，文件内容的组织方式为数据和 OOB 间隔连续。某些文件系统镜像属于这种类型，如 yaffs2，用 mkyaffs2image 生成的镜像就是这种类型。
3） 有 OOB 也有 ECC：镜像文件中包括了 OOB 信息，也包含 ECC 信息，文件内容的组织方式为数据和 OOB 间隔连续。从 Nand Flash 母片生成的文件属于这种类型。

注意事项：

1） 烧录校验的方法是将数据从 Nand Flash 中读回来逐字节对比，如果检验没有报错，则说明烧录没有问题。
2） 如果烧录过程没有报错，但是系统启动之后报错了，很大的可能是烧录方法与操作系统的读取方法不匹配，请仔细检查 ECC，PLANE 等设置。
3） Nand Flash 最开始的 16KB 的烧录方法是固定的，不受配置文件影响。对 Jz4740，最开始的 16KB

代码使用 RS ECC，ECC 存放的偏移为 6。对于 Jz4750 和 Jz4755，最开始的 16KB 代码使用 BCH 8-bit ECC，ECC 偏移为 3。对于 Jz4760，最开始的 16KB 代码使用 BCH 24-bit ECC，ECC 保存在奇数页。

4） 烧录中包含了擦除的操作，烧录之前不需要先进行擦除。擦除的范围根据烧录文件的大小确定，如果需要擦除比镜像文件更大的空间，就需要先手动擦除。

烧录的实例参考后文。

4.3 测试功能

这类命令在 USB Boot 中包括读 Nand Flash。

4.3.1 读 Nand Flash

READ 用于读当前设备上的 Nand Flash。

格式：

from：起始页。

to：结束页。

注意事项：

结果保存在用户指定文件。

5 烧录实例

本节介绍实际的烧录方法，本节将以君正网站发布的 Linux BSP 的整个烧录过程为例，具体介绍烧录工具和命令的用法。下面涉及到的配置文件的改动都是针对君正发布的 Linux BSP，如果是别的系统，或者是修改过的 Linux BSP，这些配置需要根据实际情况修改。硬件方面，本节以君正的参考设计 LETUS 开发板为例。

请用户务必明确，烧录的根本原则是：烧录的设置（ECC 偏移，ECC 算法，plane 方式，坏块标记位置等）要与读取烧录内容的程序匹配。例如，烧录 bootloader 的设置要与芯片内 bootrom 代码定义匹配；烧录 kernel 要与 bootloader 的定义匹配；烧录文件系统要与 kernel 的文件系统驱动匹配。

5.1 LEPUS 板

从君正网站下载 USB Boot 工具包并解压。

LETUS 开发板搭载 Jz4760 处理器。

1）修改配置文件：Release/4760/HardwareCfg.ini，Release/4760/LinuxNandCfg.ini，Release/tool_cfg/LinuxFileCfg.ini，将 Release/4760/文件夹下 fw_ddr2.bin 改名为 fw.bin。
2）打开 USBbootTool.exe，配置 CPU 型号，选择烧录文件。
3）按下 SW8，最后按下复位键。如果成功，在 PC 上应该能看到一个新的 usb 设备，同时软件自动执行烧录。
4）烧录结束后设备自动重启。

5.2 烧录其它文件系统

其他文件系统的烧录也要两个过程：制作镜像和烧录。

由于文件系统的特点不同，制作镜像的过程可能不一样。有的可以通过专门的工具，有的需要通过母片的方式。无论哪种方法，最后都生成的镜像文件都必须归类到三种文件类型当中。USB Boot 工具只能烧录符合文件类型约定的镜像。

例如 FAT 文件系统，通过母片的方式把数据完整读出来，生成文件 root26.fat。以 PAVO 为例，这个文件的结构是 2KB 数据和 64 字节 OOB 交替，并且 OOB 中包含了 ECC 信息，属于有 OOB 又有 ECC 的类型。

修改配置文件，然后烧录到MTD最后一个分区。

6 烧录原理简介

6.1 Boot 流程介绍

Boot 过程需要经历两个阶段。其中第一个阶段负责初始化第二阶段程序运行的环境，第二阶段才能提供烧录等功能。

USB Boot 版本的 Boot 过程详细如下：

1） 用户 boot 设备，程序自动检测 usb 设备。
2） 主机方程序扫描配置文件，检查配置文件的正确性，然后设置第一阶段的参数。
3） 主机方程序将从配置文件中获得的参数和 fw.bin 一起发送给目标开发板。
4） 目标开发板在 cache 中执行 fw.bin，fw.bin 读取参数，初始化目标开发板。
5） 目标开发板从 fw.bin 返回，主机检查第一阶段是否运行成功。
6） 主机程序根据当前目标开发板的 SDRAM 大小计算第二阶段的执行位置，执行位置位于 SDRAM 中的最后 4M，然后将 usb_boot.bin 发送到目标板 SDRAM 中的执行位置。
7） usb_boot.bin 在目标板 SDRAM 中执行，其最初的代码根据执行位置修改程序本身的 GOT 表，使之能够在当前地址执行。
8） GOT 修改完毕，usb_boot.bin 跳转到主函数入口，usb_boot.bin 正式运行。
9） 主机程序将完整的配置文件参数通过第二阶段协议发送给目标板，目标板记录这些设置。
10） 主机程序验证 usb_boot.bin 是否正常运行，Boot 过程结束。