此版本uboot在XC2440\_uboot\_V1.3版本上修改得来，是为我们的一个大客户做的量产工具和智能升级工具。

    以前给XC2440核心板烧程序用电脑的USB口，每次还要接上串口，按选项选择烧写功能，电脑上也要频繁操作选择下载的文件。小量的话还好说，就是要费点时间。量大就不能这么干了，有的客户一次要1千套，这么搞会搞死人的。

   同时还有个问题一直困扰我，用2440做的产品，如果要更新系统怎么办。而且东西已经卖到了国外，不可能是让客户把东西发过来让我们给更新吧，也不可能让用户自己接个串口输命令吧。必须想个让用户自己就能方便操作的办法，做个智能的升级工具。

   首先我想到了网络，因为设备上是带有网口的，但是这还需要做个上位机软件，很麻烦，不愿投入开发这个。因为用过uboot的U盘烧写系统镜像的功能，就想到用U盘更新系统比较方便，设备上有USB HOST接口，U盘也方便操作，往里面放入要更新的镜像即可。

    因为设备一开机就直接就系统，只有在接上串口后，在开机时输入个键值才能进uboot，因为设备上没有留出串口，即使有串口和usb device，我们也不可能让用户去学习操作这些对他们没用的东西。于是我就根据我们做手机开发时，在手机开机的同时按住一个按键，即可进入 bootloader的下载模式得到启发，也做一个这样的检测机制。这样开机时只要按住一个按键若干秒后即可进入下载模式，然后再用按一个按键就开始自动 执行我设置好了从U盘烧写镜像的程序。

**操作方法：**

一、烧写uboot镜像到Nandflash的bios分区，也可以烧写到Norflash中，在Norflash中烧写Nandflash

二、准备一个U盘或SD/TF卡读卡器，接在电脑上，先将准备烧写的u-boot.bin，zImage.bin，rootfs.bin三个文件复制到U盘根目录下

注意：文件名不能错，否则找不到文件。U盘中尽量不要装太多文件，否则影响查找文件的速度

三、将U盘插在开发板的USB HOST接口上，按如下步骤操作（其中3～8是自动执行的）

1. 开机的同时按住KEY1，大概3秒后，看到LED1亮起即可松开按键，此时进入udisk download模式

2. 按一下Key2键，隔1秒后观察LED2会快速闪烁两下，此时开始烧写uboot镜像

3. 烧写完uboot后，LED2会长亮一秒后熄灭，此时已经烧写完uboot镜像

4. 接着LED3会快速闪烁两下，此时开始烧写zImage镜像

5. 烧写完zImage后，LED3会长亮一秒后熄灭，此时已经烧写完zImage镜像

6. 接着LED4会快速闪烁两下，此时开始烧写rootfs镜像

7. 烧写完rootfs后，LED4会长亮一秒后熄灭，同时LED1也熄灭，此时已经烧写完rootfs镜像

8. 自动重启系统，启动linux系统

**程序的大体思路是这样：**

    开机程序跑到main.c中的main\_loop函数，在判断bootdelay，读取连接按键的GPIO管脚状态，看否按键按下，如果没有按键按下，就照常往下跑，直接进系统。

    如果按键被按下，程序就停下来，用一个while等待另一个按键按下。当另一个按键按下了，开始执行后面的程序，就是用run\_command依次执行 usb start，fatload，nand erase，nand write这些命令把boot，kernel，rootfs都烧写完成后reset uboot即可，中间还可以加入LED闪烁和蜂鸣器发声的动作表示烧写状态，利用GPIO的操作即可

基本过程为：uboot 启动后，通过命令将U盘的的文件读取到内存中，再通过uboot 的flash 写入命令将读取到内存中的升级文件写入到flash的固定位置。

（一）usb常用命令

    uboot一般支持对usb的操作命令有：start,reset,stop,tree,info,storage,dev,part,read,write等。

Usage:

usb start - start (scan) USB controller

usb reset - reset (rescan) USB controller

usb stop [f] - stop USB [f]=force stop

usb tree - show USB device tree

usb info [dev] - show available USB devices

usb storage - show details of USB storage devices

usb dev [dev] - show or set current USB storage device

usb part [dev] - print partition table of one or all USB storage devices

usb read addr blk# cnt - read `cnt' blocks starting at block `blk#' to memory address `addr'

usb write addr blk# cnt - write `cnt' blocks starting at block `blk#' from memory address `addr'

1.usb start

Wisdom # usb start

(Re)start USB...

USB: scanning bus for devices...

2 USB Device(s) found

scanning bus for storage devices... iVendor 0 iProduct 0

iVendor 5E3 iProduct 727

ss->subclass : 0x6

usb\_stor\_get\_info->1496,blksz:512

1 Storage Device(s) found

Wisdom #

2.usb tree

Wisdom # usb tree

Device Tree:

1 Hub (12 Mb/s, 0mA)

| OHCI Root Hub

|

+-2 Mass Storage (12 Mb/s, 500mA)

Generic USB Storage 000000000250

3.usb infor

Wisdom #

Wisdom # usb info

1: Hub, USB Revision 1.10

- OHCI Root Hub

- Class: Hub

- PacketSize: 8 Configurations: 1

- Vendor: 0x0000 Product 0x0000 Version 0.0

Configuration: 1

- Interfaces: 1 Self Powered 0mA

Interface: 0

- Alternate Setting 0, Endpoints: 1

- Class Hub

- Endpoint 1 In Interrupt MaxPacket 2 Interval 255ms

2: Mass Storage, USB Revision 2.0

- Generic USB Storage 000000000250

- Class: (from Interface) Mass Storage

- PacketSize: 64 Configurations: 1

- Vendor: 0x05e3 Product 0x0727 Version 2.80

Configuration: 1

- Interfaces: 1 Bus Powered 500mA

Interface: 0

- Alternate Setting 0, Endpoints: 2

- Class Mass Storage, Transp. SCSI, Bulk only

- Endpoint 1 In Bulk MaxPacket 64

- Endpoint 2 Out Bulk MaxPacket 64

4.usb part

Wisdom # usb part

index: 0

Partition Map for USB device 0 -- Partition Type: DOS

Partition Start Sector Num Sectors Type

1 8192 7618560 b

index: 1

Wisdom #

5.usb storage

Wisdom # usb storage

Device 0: Vendor: Generic Rev: 0250 Prod: STORAGE DEVICE

Type: Removable Hard Disk

Capacity: 3724.0 MB = 3.6 GB (7626752 x 512)

Wisdom #

（二）U盘升级操作

（1）usb start  开始usb 操作

Wisdom # usb start

(Re)start USB...

USB: scanning bus for devices...

2 USB Device(s) found

scanning bus for storage devices... iVendor 0 iProduct 0

iVendor 5E3 iProduct 727

ss->subclass : 0x6

usb\_stor\_get\_info->1496,blksz:512

1 Storage Device(s) found

（2）fatls usb 0  查看usb 中的文件

Wisdom # fatls usb 0

index: 0

.trash-1000/

4.8.7/

5.9.0/

20190323/

dejavu/

fb/

install/

mouse/

qt\_4.8\_install/

rules.d/

935839 gui\_test

727 profile

3671576 uidemo18

3190568 uimage

4 file(s), 10 dir(s)

Wisdom #

（3）擦除内存中的一块空间用来放置升级文件：

    将0x82000000开始的 4M空间填上0xFF

mw.b 82000000 ff 400000

（4）加载usb的升级文件到内存

fatload usb 0 0x82000000 uimage

（5）擦除flash中原来的数据

nand erase 100000 400000

（6）将内存中的升级文件写入到flash中去

nand write 82000000 100000 400000

（3）~（6）操作过程

Wisdom # mw.b 82000000 ff 400000

Wisdom # fatload usb 0 0x82000000 uimage

index: 0

##########################################################################################################################################################################################

3190568 bytes read

Wisdom # nand erase 100000 400000

NAND erase: device 0 offset 0x100000, size 0x400000

Erasing at 0x4e0000 -- 100% complete.

OK

Wisdom # nand write 82000000 100000 400000

NAND write: device 0 offset 0x100000, size 0x400000

4194304 bytes written: OK

Wisdom #

全部升级操作命令：

usb start

fatls usb 0

mw.b 82000000 ff 400000

fatload usb 0 0x82000000 uimage

nand erase 100000 400000

nand write 82000000 100000 400000

      以上，仅供参考。

————————————————

版权声明：本文为CSDN博主「li\_wen01」的原创文章，遵循CC 4.0 BY-SA版权协议，转载请附上原文出处链接及本声明。

原文链接：https://blog.csdn.net/li\_wen01/article/details/88780481