

## 【UBoot 移植】当 bootdelay 误设置为 0 时的解决办法 2010-05-14 17:34:33

分类： LINUX

在开发板上配置 UBOOT 环境变量时，不小心误把 BOOTDELAY 设置为 0，这样就不能在开发板启动后进入配置环境，也就无法将 BOOTDELAY 修改回来。

首先想利用 AXD 在线下载原始 UBOOT 镜像到 SDRAM，在线运行后来进入配置环境修改参数，但是实际进行后发现，虽然 AXD 在线在 SDRAM 中跑正常 UBOOT，但是 UBOOT 启动的时候还是读取了存储在 FLASH 中的 BOOTARGS 参数信息，依然时 BOOTDELAY 为 0，所以这样子也行不通。

这时候我想到去 UBOOT 源代码中修改关于 BOOTDELAY 的相关代码功能，以让其在启动过程中不读取 FLASH 中的 BOOTARGS 参数。

PS：在请教师兄后，师兄也提出了和我想的一样的方法，同时还有一个方法就是在 AXD 中对 FLASH 进行擦除操作，但是这里就要涉及到 FLASH 操作的很多内容，也比修改代码要复杂，如果以后要研究 FLASH 的操作的话，可以考虑尝试下。

BOOTDELAY 参数的功能函数位置在 /common/main.c 中，找到：

```
static __inline__ int abortboot(int bootdelay)
```

注意：在 main.c 中有两段同名的 abortboot 函数，通过预处理来进行选择，分析函数内容可以知道是第二个函数。

首先修改了判断是否自启动的信号变量 abort，修改如下：

```
int abort = 1;          /* wuwei have changed this : the old is "abort = 0" ! */
                        /* The value of abort define autoboot by 0 and unautoboot by 1 */
```

然后通过预处理去掉接在后面的 bootdelay 判断处理函数：

```
#if 0          /*wuwei closed blow code , let uboot not to juge bootdelay*/
#ifdef CONFIG_MENU_PROMPT
.....
printf ("\b\b\b%2d ", bootdelay);
#endif
```

这样修改后 UBOOT 就在启动后停着等待用户输入命令，不进行 BOOTDELAY 的倒计时过程了。

\*\*\*\*\*

更 新

\*\*\*\*\*

根据师兄的指导，了解到参数的位置在 0x20000 ~ 0x3ffff 之间，也就是接在 uboot 程序所在的 0x0 ~ 0x1ffff 区间的后面，uboot 启动时先读取该区间段的参数，如果没有，就按照源码中 config 文档中的默认配置。