

[置顶] 亮仔移植u-boot系列之-- S3c2440在最新版本U-boot-2015.10移植(支持SPL模式启动) -- 1

标签： 移植 u-boot

2015年12月04日 21:23:39

1945人阅读

评论(0)

收藏

举报

目录(?)

[+]

我的qq：631223432 欢迎交流，转载请注明

第一步:下载最新的2015-10版本并配置

首先在u-boot官网上下载最新的U-boot-2015.10版本，通过命令 `sudo tar xvf u-boot-2015.10.tar.bz2` 后解压，进入u-boot目录，执行:

```
make smdk2410_config
```

出现lib/asm-offsets.c:1: error: bad value (armv4) for -march= switch

这是因为没有指定交叉编译器工具所致，因此输入

```
sudo gedit Makefile命令
```

编辑u-boot根目录下的MakeFile:

```
ifeq ($(HOSTARCH),$(ARCH))
```

```
CROSS_COMPILE ?=
```

```
endif
```

```
++CROSS_COMPILE=arm-linux- //加入交叉编译工具链的支持
```

重新make，出现：

Error: selected processor does not support `bx lr'的错误，显然，这是arm-linux-gcc版本过低引起的不能识别bx命令引起的错误，需要安装高版本的编译器。

在友善之臂的官网上下载arm-linux-gcc4.4.3，详细安装arm-linux-gcc4.4.3过程网上教程很多，本文就不介绍怎么安装了。重新输入命令：

```
make smdk2410_config
```

make

代码编译没有问题，编译环境设置成功.

第二步: 添加并编辑适合自己开发板的目录文件

1. 在board\samsung\目录下添加yl2440目录，将smdk2410目录下的所有文件全部拷贝到yl2440目录中.

1.1 修改目录下的makefile：

[cpp]  

1. `obj-y := yl2440.o boot_init.o`
2. `obj-y += lowlevel_init.o`

1.2 修改目录下的Kconfig：

[cpp]  

1. `if TARGET_YL2440`
- 2.
3. `config SYS_BOARD`
4. `default "yl2440"`
- 5.
6. `config SYS_VENDOR`
7. `default "samsung"`
- 8.
9. `config SYS_SOC`
10. `default "s3c24x0"`
- 11.
12. `config SYS_CONFIG_NAME`
13. `default "yl2440"`
- 14.
15. `endif`

2. 在include\configs\目录下添加yl2440.h目录，将smdk2410.h内容全部拷贝到yl2440.h目录下

3.在configs\目录下加入yl2440_deconfig文件

[cpp]  

1. CONFIG_ARM=y
2. CONFIG_TARGET_YL2440=y
3. CONFIG_SYS_PROMPT="YL2440 #"

4.进入arch\arm目录

4.1 修改目录下的Kconfig文件：

[cpp]  

1. config TARGET_YL2440
2. bool "Support yl2440"
3. select CPU_ARM920T

[cpp]  

1. source "board/samsung/yl2440/Kconfig"

重新执行

make clean

make yl2440_config

make

后编译成功，此时针对本开发板的文件修改添加成功.

第三步: 修改编译适合自己板卡的u-boot-spl.bin(BL1.bin)

前言:为什么需要将原先u-boot-1.1.6版本一个bin拆分成两个不同功能的bin——bl1.bin&bl2.bin

之前u-boot-1.1.6版本前4k映射到ARM S3C2440的steppingstone内,即将NAND Flash的前4K自动映射到CPU的4K IRAM里面，执行CPU的相关初始化和代码搬移到外部SDRAM的工作，假如这些工作相关的代码链接后超出了4K范围外，那么整

个代码无法搬移到SDRAM中得到正常执行.

u-boot-2015.10版本通过CONFIG_SPL_BUILD宏来控制MAKE编译产生uboot.bin或uboot-spl.bin.我之前网上查到的资料写的有点不对,并非是通过在代码中加入:

```
#define CONFIG_SPL_BUILD
```

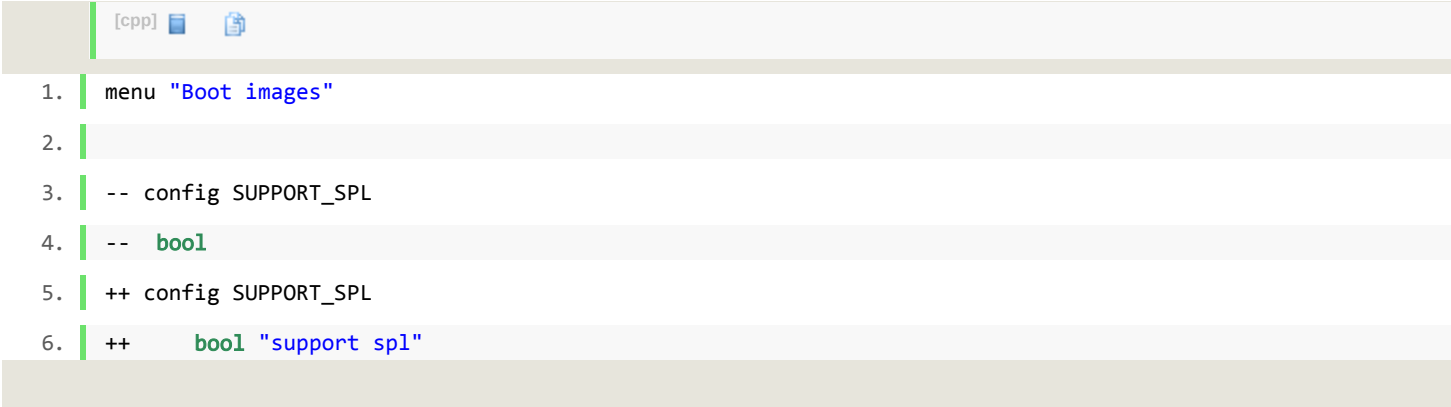
来控制的,因为我在定义了CONFIG_SPL_BUILD后MAKE会产生语法错误,正确的方法为,输入命令:

```
make menuconfig
```

选择选项

```
boot images --->
```

发现并没有Enable SPL的选项,退出menu后编辑uboot根目录下的Kconfig文件,修改如下代码:



```
1. menu "Boot images"
2.
3. -- config SUPPORT_SPL
4. -- bool
5. ++ config SUPPORT_SPL
6. ++ bool "support spl"
```

重新make menuconfig后可以看到

```
boot images --->
```

```
[ ] support spl
```

```
[ ] Enable SPL
```

```
[ ] Enable SDRAM location for SPL stack
```

三个选项.将头两个选项勾选,保存退出后make在SPL目录下生成u-boot-spl.bin.将头两个选项去掉后make,则在uboot根目录下生成

u-boot.bin.我们看到u-boot-spl.bin即本文之前提到的BL1,大小为2K,u-boot.bin为BL2,大小为196K.

yl2440目标板针对BL1,BL2任务功能的划分

在yl2440.h文件注释掉宏,并添加以下宏:

```
1. -- #define CONFIG_S3C2410
2. -- #define CONFIG_SMDK2410
3. ++ #define CONFIG_S3C2440 //目标板为2440
4. ++ #define CONFIG_YL2440
5. ++ #define CONFIG_SYS_TEXT_BASE 0x30008000 //定义BL2.bin的链接地址为0x30008000
6. ++ #define CONFIG_SPL_TEXT_BASE 0x0 //定义BL21.bin的链接地址为0x0
7. ++ #define CONFIG_SPL_MAX_SIZE 0x1000 //定义BL1的大小最大4K，由S3C2440 CPU决定，超出会报编译错
8. ++ #define CONFIG_SPL_STACK 0x1000 //定义BL1堆栈起始为0x1000
9. ++ #define BL2_MTD_OFFSET 0x20000 //定义BL2在NAND Flash128K偏移，即起始Block为Block1，128K
10. ++ #define BL2_MTD_LEN 0x100000 //定义BL2在NAND Flash大小占用1M
```

BL1任务划分：

1. 看门狗，中断，CPU时钟初始化
2. 设置在BL1模式下的SP指针
3. 搬移位于Nand Flash BL2位置的代码到SDRAM 0x30008000处
4. 将PC指针指向0x30008000，即BL2 Start.S程序入口地址

BL2任务划分：

1. 将BL2代码段0x30008000进行动态重定位
2. 做u-boot-1.1.6 _start_arm_boot类似NAND初始化，串口初始化，网卡初始化等工作并接收命令引导Linux系统

下一章将说明如何修改BL1.