K9_Uboot 移植

K9 配套的软件 K9_Uboot 包含三个主要部分：
k9loader 暂时性存放并运行于 RAM 中。为运行 uboot 做必要的硬件初始化。
k9boot 固定存放并运行于 flash 中。为运行 uboot 做必要的硬件初始化。
k9Uboot 固定存放于 flash 中，运行在 SDRAM 里。

K9_Uboot 是从 uboot1.0.0 版本移植而成。

K9_Uboot 提供五个文件：cross.2.93.5.tar.gz k9loader.tar.gz k9boot.tar.gz k9Uboot.tar.gz k9bin.rar 。其中 k9bin.rar 里面包含最后编译好的 bin 文件：
k9loader.bin k9boot.bin k9Uboot.bin。

1．K9_Uboot 移植环境设置
   a) 交叉编译工具
   uboot1.0.0 采用 cross.2.93.5 交叉编译工具。
   b) 目录和文件
   将 cross.2.93.5.tar.gz 解压缩到 /usr/local/arm/2.93.5 下面。
   将 k9loader.tar.gz 解压缩到 /usr/local/arm/k9loader 下面。
   将 k9boot.tar.gz 解压缩到 /usr/local/arm/k9boot 下面。
   将 k9Uboot.tar.gz 解压缩到 /usr/local/arm/k9Uboot 下面。
2．K9_loader 编译
   在目录/usr/local/arm/k9loader/下执行
   make clean
   make
   生产 k9loader.bin 文件。
3．K9_boot 编译
   在目录/usr/local/arm/k9boot/下执行
   make clean
   make
   生产 k9boot.bin 文件。
4．K9_Uboot 编译
   在目录/usr/local/arm/k9Uboot/下执行
   make clean
   make at91rm9200dk_config
   make all
   生产 k9Uboot.bin 文件。

K9_loader 移植要点

1.将 Makefile 里面的 loader.bin 更改为 k9loader.bin，共二处.

2.修改 INIT.C 里面的 SDRAMINIT，设置为 32 位总线.

3.修改 main.h

```
#define AT91C_UBOOT_BASE_ADDRESS    0x20F00000
#define AT91C_UBOOT_MAXSIZE         0x20000
```

4.修改 main.c

修改打印信息

K9_boot 移植要点

1.将 Makefile 里面的 boot.bin 更改为 k9boot.bin，共二处.

2.修改 main.c

```
//decompress_image(SRC,DST,LEN);
 memcpy(DST,SRC,LEN);
 /*这里解决解压缩不成功的问题，所以 k9Flash 里面保存的是 k9uboot.bin 而不是 uboot.bin.gz */
```

3.修改 initboot.c 中 initflash 部分.

4.修改 initboot.c 中 initSDRAM 部分.

5.修改 entry.S

```
#define STACK                   0x20e00000    //0x22000000
```

K9_Uboot 移植要点

1.修改/borad/at91rm9200dk/flash.c

ａ. flash_init 增加

```
flash_info[i].flash_id=(INTEL_MANUFACT&FLASH_VENDMASK)|(INTEL_ID_28F320J3A&FLASH_TYPEMASK);
```

ｂ. flash_print_info 增加

```
case(INTEL_MANUFACT & FLASH_VENDMASK):printf ("Intel: ");break;
case(INTEL_ID_28F320J3A & FLASH_TYPEMASK):
printf ("E28F320J3A (32Mbit)\n");break;
```

ｃ. int flash_erase

```
注销
if(info->flash_id==FLASH_UNKNOWN) return ERR_UNKNOWN_FLASH_TYPE;
注销
if((info->flash_id&FLASH_VENDMASK)!=(INTEL_MANUFACT&FLASH_VEND
   MASK))   {return ERR_UNKNOWN_FLASH_VENDOR;}
```

2.修改/board/at91rm9200dk/config.mk

```
TEXT_BASE = 0x20f00000
```

3.修改 include/configs/at91rm9200dk.h

```
#define CFG_MAX_NAND_DEVICE    0     /* Max number of NAND devices          */
#define PHYS_SDRAM_SIZE 0x1000000    /* 16 megs */
#define CONFIG_HAS_DATAFLASH    0
#define PHYS_FLASH_SIZE 0x0400000    /* 4 megs main flash */
#define CFG_MAX_FLASH_SECT 32
#define CFG_LOAD_ADDR 0x20f00000    /* default load address */
```

4.修改/board/at91rm9200dk/config.mk

修改 boot env uboot 的地址空间分配

附：U-BOOT 常用命令简介

一 环境变量操作

（1） Setenv
设置环境变量
举例
setenv serverip 192.168.0.1
setenv ipaddr 192.168.0.56
setenv bootcmd ‘tftp 32000000 vmlinux; kgo 32000000’
（2） saveenv: 保存环境变量
在设置好环境变量以后，保存变量值
（3） printenv
举例
printenv，打印所有环境变量
printenv ipaddr，打印环境变量 ipaddr 的值

二 FLASH 操作

（1） flinfo
显示 Flash 芯片的相关信息，包括 Flash 容量、扇区起始地址以及是否保护等信息
命令格式：flinfo N，N 表示第 N 片 flash
（2） protect
保护（取消保护）Flash 的各扇区
命令格式：
protect on/off N:SF[-SL]，保护（取消保护）第 N 片 Flash 上的 SF 到 SL 之间的扇区
protect on/off bank N
protect on/off all
protect on/off start end
（4） erase
擦除 Flash 的各扇区
命令格式：
erase N:SF[-SL]，擦除第 N 片 Flash 上的 SF 到 SL 之间扇区
erase bank N，擦除第 N 片 Flash 所有扇区
erase all，作用同上
举例：
erase 1:0-2(就是对第一块 FLASH 的 0-2 扇区进行删除)

注：N 从 1 开始

三 内存操作

包括读取、比较、更改、写入等操作，通用命令如下
[.b, .w, .l] address [length]
其中，.b，.w，.l 分别表示 8/16/32 位操作
（1） md

显示某地址处的内存值。
命令格式：md[.b, .w, .l] address [length]
举例
md.b 10000000 16，表示以字节为单位显示地址 0x10000000 起始的 16 字节数据
（2） mm
更改某地址处的内存值
命令格式
mm[.b, .w, .l] address
（3） mw
向某地址写入数据
命令格式
mw[.b, .w, .l] address value [count]
（4） cp
将一个地址处数据拷贝到另一地址处。
命令格式
cp[.b, .w, .l] source target count
注：此命令支持从 SDRAM 拷贝数据到 Flash 中
（5） cmp
比较两地址处的数据
命令格式
cmp[.b, .w, .l] addr1 addr2 count

四 下载和运行控制

（1） go
从某地址处开始运行。
命令格式
go addr [arg ...]
（2） run
运行某环境变量里面的内容
命令格式
run var [...]
（3） bootm
执行存储在某地址处的 uboot 格式的 image 文件
命令格式
bootm [addr [arg ...]]
（4） tftp
通过网络 tftp 协议下载文件到内存某一地址处。
命令格式
tftp [loadAddress] [bootfilename]
（5） loadb
通过串口使用 KERMIT 协议下载文件到内存某一地址处。
命令格式
loadb [off] [baud]

五 其它

（1） help/?

举例

help，得到所有命令列表

help usb，列出 USB 功能的使用说明