

u-boot移植编译

将u-boot-2013.01.tar.bz2文件放到Ubuntu内解压
tar -xvf u-boot-2013.01.tar.bz2

1.抄板

仿照别人的板子修改一些信息,改成自己的

看boards.cfg, 复制284行, 修改origen为fs4412

u-boot-2013.01\$ vi boards.cfg

复制284行, yy复制, p粘贴, 将复制的285行中的两个origen修改为fs4412

u-boot-2013.01\$ cp include/configs/origen.h include/configs/fs4412.h

u-boot-2013.01\$ cp -r board/samsung/origen/ board/samsung/fs4412

u-boot-2013.01\$ mv board/samsung/fs4412/origen.c board/samsung/fs4412/fs4412.c

u-boot-2013.01\$ vi Makefile

修改185行: CROSS_COMPILE ?= arm-none-linux-gnueabi-

u-boot-2013.01\$ vi board/samsung/fs4412/Makefile

修改31行: COBJS += fs4412.o

u-boot-2013.01\$ make fs4412 //敲 fs4 用tab键补全

make完成后, 看看有没有报错信息, 没有的话就能生成 u-boot.bin 文件

2.添加设备相关代码

2.1 点亮led灯

在启动u-boot后点亮led灯能明确知道cpu能否在工作, 因为程序最先执行汇编程序进行启动, 所以修改汇编文件, 在 reset后加上以下代码, 大致在126行:

u-boot-2013.01\$ vi arch/arm/cpu/armv7/start.S

```
/* led3 init */
ldr r0, =0x11000c20
ldr r1, [r0]
bic r1, #0xf
orr r1, #0x1
str r1, [r0]

/* led3 on */
ldr r0, =0x11000c24
ldr r1, [r0]
orr r1, #0x1
str r1, [r0]
```

2.2 关闭看门狗

让系统不自动重启,跟在上面代码后面添加以下内容:

```
#if 1 /*for close watchdog */
/* PS-Hold high */
ldr r0, =0x1002330c
ldr r1, [r0]
orr r1, r1, #0x300
str r1, [r0]
ldr r0, =0x11000c08
ldr r1, =0x0
str r1, [r0]
/* clear MASK_WDT_RESET_REQUEST */
ldr r0, =0x1002040c
ldr r1, =0x00
str r1, [r0]
#endif
```

2.3 调整串口时钟配置

设置后才能通过串口输入输出, 修改uart_asm_init: 标签处, 大致在336行, 在 str r1, [r0, #EXYNOS4_GPIO_A1_CON_OFFSET] 后加入以下代码

u-boot-2013.01\$ vi board/samsung/fs4412/lowlevel_init.S

```
ldr r0, =0x10030000
ldr r1, =0x666666
ldr r2, =CLK_SRC_PERILO_OFFSET
str r1, [r0, r2]
ldr r1, =0x777777
ldr r2, =CLK_DIV_PERILO_OFFSET
str r1, [r0, r2]
```

禁用trustzone技术, 屏蔽 bl uart_asm_init下面内容, 大致是90行
在前面加上一个@符号

```
@ bl tzpc_init
```

2.4 网卡驱动和相关命令

2.4.1 添加网络初始化代码, 修改.c文件

u-boot-2013.01\$ vi board/samsung/fs4412/fs4412.c

在 struct exynos4_gpio_part2 *gpio2; 后添加以下内容, 大致在31行

```
#ifdef CONFIG_DRIVER_DM9000
#define EXYNOS4412_SROMC_BASE 0x12570000

#define DM9000_Tacs (0x1)
#define DM9000_Tcos (0x1)
```

```

#define DM9000_Tacc      (0x5)
#define DM9000_Tcoh     (0x1)
#define DM9000_Tah      (0xc)
#define DM9000_Tacp     (0x9)
#define DM9000_PMC      (0x1)

struct exynos_sromc {
    unsigned int bw;
    unsigned int bc[6];
};

/*
 * s5p_config_sromc() - select the proper SROMC Bank and configure the
 * band width control and bank control registers
 * srom_bank      - SROM
 * srom_bw_conf   - SMC Band width reg configuration value
 * srom_bc_conf   - SMC Bank Control reg configuration value
 */
void exynos_config_sromc(u32 srom_bank, u32 srom_bw_conf, u32 srom_bc_conf)
{
    unsigned int tmp;
    struct exynos_sromc *srom=(struct exynos_sromc *) (EXYNOS4412_SROMC_BASE);

    /* Configure SMC_BW register to handle proper SROMC bank */
    tmp = srom->bw;
    tmp&= ~(0xF << (srom_bank * 4));
    tmp |= srom_bw_conf;
    srom->bw = tmp;
    /* Configure SMC_BC register */
    srom->bc[srom_bank] = srom_bc_conf;
}

static void dm9000aep_pre_init(void)
{
    unsigned int tmp;
    unsigned char smc_bank_num = 1;
    unsigned int    smc_bw_conf=0;
    unsigned int    smc_bc_conf=0;

    /* gpio configuration */
    writel(0x00220020, 0x11000000 + 0x120);
    writel(0x00002222, 0x11000000 + 0x140);
    /* 16 Bit bus width */
    writel(0x22222222, 0x11000000 + 0x180);
    writel(0x0000FFFF, 0x11000000 + 0x188);
    writel(0x22222222, 0x11000000 + 0x1c0);
    writel(0x0000FFFF, 0x11000000 + 0x1c8);
    writel(0x22222222, 0x11000000 + 0x1e0);
    writel(0x0000FFFF, 0x11000000 + 0x1e8);
    smc_bw_conf &= ~(0xf<<4);
    smc_bw_conf |= (1<<7) | (1<<6) | (1<<5) | (1<<4);
    smc_bc_conf = ((DM9000_Tacs << 28)
                   | (DM9000_Tcos << 24)
                   | (DM9000_Tacc << 16)
                   | (DM9000_Tcoh << 12)
                   | (DM9000_Tah << 8)
                   | (DM9000_Tacp << 4)
                   | (DM9000_PMC));
}

```

```

        exynos_config_sromc(smc_bank_num, smc_bw_conf, smc_bc_conf);
    }
#endif

```

在 gd->bd->bi_boot_params = (PHYS_SDRAM_1 + 0x100UL); 后添加以下内容
大致在107行, 就在刚刚添加代码的下面一点点, 在 board_init() 函数内部

```

#ifdef CONFIG_DRIVER_DM9000
    //根据dm9000网卡的参数初始化外扩EBI总线（位宽等匹配）
    dm9000aep_pre_init();
#endif

```

在文件末尾添加

```

#ifdef CONFIG_CMD_NET
int board_eth_init(bd_t *bis)
{
    int rc = 0;

    #ifdef CONFIG_DRIVER_DM9000
        // 网卡初始化代码的调用
        rc = dm9000_initialize(bis);
    #endif

    return rc;
}
#endif

```

修改 checkboard 函数内的打印信息, 可以修改u-boot的启动信息, 可根据自己需求来修改, 大概在150行左右

也可以在printf内加上组号, 组员名, 一些个人的打印信息....

```

int checkboard (void)
{
    printf("\nBoard: FS4412\n");
    return 0;
}

```

2.4.2 修改配置文件添加网络相关配置, 修改.h文件

u-boot-2013.01\$ vi include/configs/fs4412.h

```

修改 #undef CONFIG_CMD_PING    //大致在85行
为   #define CONFIG_CMD_PING

修改 #undef CONFIG_CMD_NET     //大致在90行
为   #define CONFIG_CMD_NET

```

在文件末尾 #endif /* __CONFIG_H */ 前面 添加以下内容 (在倒数第2行后加入):

```

#ifdef CONFIG_CMD_NET
#define CONFIG_NET_MULTI
#define CONFIG_DRIVER_DM9000 1
#define CONFIG_DM9000_BASE 0x05000000

```

```
#define DM9000_IO                CONFIG_DM9000_BASE
#define DM9000_DATA              (CONFIG_DM9000_BASE + 4)
#define CONFIG_DM9000_USE_16BIT
#define CONFIG_DM9000_NO_SROM    1
#define CONFIG_ETHADDR          11:22:33:44:55:66
#define CONFIG_IPADDR            192.168.22.123
#define CONFIG_SERVERIP          192.168.22.111
#define CONFIG_GATEWAYIP         192.168.22.1
#define CONFIG_NETMASK           255.255.255.0
#endif
```

修改104行, 修改u-boot启动后的输出信息, 可以根据自己的来修改

```
#define CONFIG_SYS_PROMPT        "FS4412 # "
```

2.5 修改内存偏移量

修改环境参数保存位置, 不修改的话会造成环境变量保存位置错误

u-boot-2013.01\$ vi include/configs/fs4412.h

修改144行, 因为我们的 myuboot_sd.img 文件是516k的, 所以将16改为516

```
#define BL1_SIZE    (516 << 10)
```

3.编译生成 u-boot.bin 文件

u-boot-2013.01\$ make fs4412

编译完成注意有没有ERROR的存在, 没有就能生成u-boot.bin文件

u-boot.bin 不能直接用

4.生成 myuboot_sd.img 文件 或 u.bin文件

4.1 生成烧写到SD卡内的 myuboot_sd.img 镜像文件

将 tool_SD_img 文件夹复制到Ubuntu内u-boot-2013.01文件夹的上一级目录下

终端进入 tool_SD_img 文件夹内:

u-boot-2013.01\$ cd ../tool_SD_img

执行shell脚本, 生成 myuboot_sd.img 镜像文件

tool_SD_img\$./mk_uboot.sh

将生成的 myuboot_sd.img 文件复制到windows 桌面

通过 SD卡烧写工具(ImageWriter) 烧写到SD卡内

将SD卡插回开发板(断电拔插), 断电重启

4.2 生成烧写到EMMC内的 myuboot_emmc.bin 镜像文件

将 tool_EMCC_bin 文件夹复制到Ubuntu内u-boot-2013.01文件夹的上一级目录下

终端进入 tool_EMCC_bin 文件夹内:

```
u-boot-2013.01$ cd ../tool_EMCC_bin
```

执行shell脚本, 生成 myuboot_emmc.bin 镜像文件

```
tool_SD_img$ ./build.sh
```

将生成的 myuboot_emmc.bin 文件复制到 tftpboot 文件夹内

```
tool_SD_img$ cp myuboot_emmc.bin /home/hqyj/tftpboot
```

先SD卡启动, tftp 将 u.bin 下载到内存内 (开发板上操作)

```
tftp 41000000 myuboot_emmc.bin
```

下载完成后, 将文件搬到EMMC当中 (开发板上操作)

```
movi write u-boot 41000000
```

断电, 将拨码开关换到 EMMC , 再上电启动开发板