Linux3.0.8平台搭建移植文档——SPI移植

1. SPI 子系统

1）添加 S3C64xx SPI 控制器驱动的支持

由于我们当前架构是 S5PV210，其 spi 控制器驱动依赖于 S3C64xx 架构，而在 make menuconfig 时无法看到 spi 控制器驱动选项，因此我们要修改其 Kconfig，让其支持 S5PV210架构（可以在 arch/arm/kconfig 目录下找到该架构的宏名“ARCH_S5PV210”）

#vi driver/spi/kconfig

找到 config SPI_S3C64XX，修改内容如下：

```
config SPI_S3C64XX
	tristate "Samsung S3C64XX series type SPI"
	depends on (ARCH_S3C64XX || ARCH_S5P64X0 || ARCH_S5PV210)
	select S3C64XX_DMA if ARCH_S3C64XX
	help
	  SPI driver for Samsung S3C64XX and newer SoCs.
```

2）添加 SPI 控制器 platform 资源

spi 控制器的 platform 资源定义在 arch/arm/mach-s5pv210/dev-spi.c 文件中，而该文件当前未被编译进内核，因此我们要将其加入编译队伍，通过 makefile 文件查看出 dev-spi.c 文件的编译依赖宏 S3C64XX_DEV_SPI，因此我们要在 kconfig 中添加该宏。

#vi arch/arm/mach-s5pv210/kconfig

找到 config MACH_SMDKV210选项，添加内容如下：

```
config MACH_SMDKV210
    bool "SMDKV210"
    ...
    select S3C64XX_DEV_SPI
    help
      Machine support for Samsung SMDKV210
```

#vi arch/arm/mach-s5pv210/mach-smdkv210.c

添加 platform 资源到资源列表（smdkv210_devices）中，如下所示：

```
static struct platform_device *smdkv210_devices[] __initdata = {
        ......
        &s3c_device_i2c2,
        &s5pv210_device_spi0,
        &s5pv210_device_spi1,

        ......
```

```
};
```

3）修改 linux3.0.8内核有关 spi 子系统的参数

在 bsp 中调用 s5pv210_spi_set_info 函数对 spi 控制器进行参数设置时会为其设置 s3c64xx_spi_info 参数，在该参数中有 cfg_gpio 配置函数的指定（s5pv210_spi_cfg_gpio 函数）。

#vi arch/arm/mach-s5pv210/dev-spi.c

找到 s5pv210_spi_cfg_gpio()函数，对函数中以下内容进行修改：

```
 static int s5pv210_spi_cfg_gpio(struct platform_device *pdev)
{
     ...
     s3c_gpio_cfgall_range(base, 4,
                           S3C_GPIO_SFN(2), S3C_GPIO_PULL_UP);
//对 io 口进行批量设置，需要配置的 io 口为4个，所以第二个参数应当为4
     return 0;
}
```

4）配置 SPI 子系统支持

#make menuconfig

```
Device Drivers  --->
          [*] SPI support  --->
                    --- SPI support
                      [ ]   Debug support for SPI drivers
                       *** SPI Master Controller Drivers ***
                      < >   Altera SPI Controller
                      -*-   Utilities for Bitbanging SPI masters
                      <*>   GPIO-based bitbanging SPI Master
                      < >   OpenCores tiny SPI
                      <*>   Samsung S3C64XX series type SPI
                      < >   Xilinx SPI controller common module
                      < >   DesignWare SPI controller core support
                     *** SPI Protocol Masters ***
                      <*>   User mode SPI device driver support
                      < >   Infineon TLE62X0 (for power switching)
```

5)make

将在 arch/arm/boot/下生成编译好的可执行程序 zImage 下载到开发板即可

2. bct3288a 呼吸灯移植

1）添加 bct3288a 驱动代码

bct3288a 为呼吸灯模组通过 SPI 接口与处理器连接，因此移植 bct3288a 必须保证系统支持 SPI

控制器。这次我们采用动态插入的方式添加该驱动，即系统运行起来后通过 insmod 进行 bct3288a 驱动的添加，此外我们还需要在 mach-smdkv210.c 文件中添加呼吸灯模组的资源信息。

编译提供好的 bct3288a 模组驱动程序最终生成.ko 文件。

使用方式：等待内核运行起来进行动态添加。

2）添加 bct3288a 的 spi_board_info 资源

#vi arch/arm/mach-s5pv210/mach-smdkv210.c

添加 spi_board_info 及 s3c64xx_spi_csinfo 相关资源，具体操作如下：

<a>在 static struct i2c_board_info smdkv210_i2c_devs2[]的后面，添加 spi_board_info 资源，具体内容如下：

```
static struct s3c64xx_spi_csinfo smdk_spi0_csi[] = {
        [0] = {
                .line = S5PV210_GPB(1),
                .set_level = gpio_set_value,
                .fb_delay = 0x0,
        },
};
static struct s3c64xx_spi_csinfo smdk_spi1_csi[] = {
        [0] = {
                .line = S5PV210_GPB(5),
                .set_level = gpio_set_value,
                .fb_delay = 0x0,
        },
};
static  struct spi_board_info smdkv210_spi_devs[] __initdata = {
        [0] = {
                .modalias        = "bct3288a",      /* device node name */
                .mode            = SPI_MODE_0,      /* CPOL=0, CPHA=0 */
                .max_speed_hz    = 10000000,
                /* Connected to SPI-1 as 1st Slave */
                .bus_num         = 1,
                .irq             = IRQ_SPI1,
                .chip_select     = 0,
                .controller_data = &smdk_spi1_csi[0],
        },
};
```

<b>在 smdkv210_machine_init 函数中在 i2c_register_board_info 函数调用的后面，添加以下函数调用，完成 spi_board_info 的注册,具体内容如下：

```
static void __init smdkv210_machine_init(void)
{
        ...
                s5pv210_spi_set_info(0, S5PV210_SPI_SRCCLK_PCLK,
                        ARRAY_SIZE(smdk_spi0_csi));
                s5pv210_spi_set_info(1, S5PV210_SPI_SRCCLK_PCLK,
```

```
                ARRAY_SIZE(smdk_spi1_csi));
        spi_register_board_info(smdkv210_spi_devs,
                ARRAY_SIZE(smdkv210_spi_devs));
        ...
}
```

<c>在文件开始位置添加 spi 相关头文件，如下所示：

```
#include <linux/spi/spi.h>
#include <plat/s3c64xx-spi.h>
#include <mach/spi-clocks.h>
```

3)make

将在 arch/arm/boot/下生成编译好的可执行程序 zImage 下载到开发板即可