MarsBoard RK3066 Pro 驱动移植手册

目录

特殊说明

命令前加“#”的表示 PC 机的 ubuntu 终端输入的，并且是 root 用户权限；命令前加“$”的表示开发板终端输入的。

1. 按键驱动移植

1.1 在 Board-rk30-box-key.c (arch\arm\mach-rk30) 这个文件 static struct rk29_keys_button key_button[]这个结构体中，第 19 行添加如下代码：

```
{
        .desc    = "esc",
        .code    = KEY_BACK,
        .adc_value    = 334,
        .gpio = INVALID_GPIO,
        .active_low = PRESS_LEV_LOW,
},{
        .desc    = "vol-",
        .code    = KEY_VOLUMEDOWN,
        .adc_value    = 135,
        .gpio = INVALID_GPIO,
        .active_low = PRESS_LEV_LOW,
},{
        .desc    = "vol+",
        .code    = KEY_VOLUMEUP,
        .adc_value    = 1,
        .gpio = INVALID_GPIO,
        .active_low = PRESS_LEV_LOW,
},
```

添加的按键对应开发板中 ESC，VOL-，VOL+

1.2 内核源码根目录下执行

```
#./build_marsboard_rk3066_mtd
```

编译内核，生 marsboard_rk3066_mtdboot.img，下载新内核到开发板中。

2. SPI 驱动移植

2.1 在 Board-rk30-box.c (arch\arm\mach-rk30)，
static struct spi_board_info board_spi_devices[] ={}
结构体中添加如下代码（添加 SPI 设备资源）：

```
{
        .modalias = "spidev",
        .chip_select =0,
        .max_speed_hz = 66000000,
```

```
        .bus_num = 0,
    }
```

2.2 配置内核支持 SPI，

```
#make menuconfig
    -> Device Drivers    --->
        ->    [*] SPI support    --->
            <*>      RK SPI master controller core support
                [*]      RK SPI0 master controller
```

2.3 内核源码根目录下执行

```
#./build_marsboard_rk3066_mtd
```

编译内核，生成 marsboard_rk3066_mtdboot.img，下载新内核到开发板中。

3. I2C 驱动移植

3.1 配置内核支持 I2C 通用驱动，

```
Device Drivers    --->
    {*} I2C support    --->
        <*>    I2C device interface
        [*]      Autoselect pertinent helper modules
```

3.2 内核源码根目录下执行

```
#./build_marsboard_rk3066_mtd
```

编译内核，生成 marsboard_rk3066_mtdboot.img，下载新内核到开发板中。

在开发板的/dev/目录下可以看到 i2c-0，i2c-1，i2c-2，i2c-3，i2c-4 五个设备文件。

4. UART 驱动移植

4.1 配置内核支持 UART

```
#. make menuconfig
    Device Drivers    --->
        Character devices    --->
        Serial drivers    --->
            [*] RockChip RK29/RK30 serial port support
            [*]      Serial port 0 support
            [*]      Serial port 2 support
            [*]      Serial port 3 support
```

4.2 内核源码根目录下执行

```
#./build_marsboard_rk3066_mtd
```

编译内核，生成 marsboard_rk3066_mtdboot.img，下载新内核到开发板中。

在开发板的/dev 目录下的 ttyS0,ttyS2,ttyS3 分别对应 UART0，UART 2，UART3。

5. USB CAMERA 驱动移植

5.1 添加驱动文件：

```
#make menuconfig
    Device Drivers    --->
        <*>   Multimedia support    --->
            <*>   Video For Linu
                [*]   Video capture adapters    --->
                    [*]   V4L USB devices    --->
                        <*> USB Video Class (UVC)
                        [*] UVC input events device support
```

5.2 内核源码目录下执行

```
#./build_marsboard_rk3066_mtd
```

编译内核，生成 marsboard_rk3066_mtdboot.img，下载新内核到开发板中。

6. LED 驱动移植

6.1 复制相关源码/ws_driver 整个文件夹到内核目录 drivers/char 下（ws_driver 目录下包含可 led.c 的驱动文件），修改 driver/char 下 Kconfig 和 Makefile 将 ws_driver 文件夹中的驱动文件添加到内核中。

6.2 在 Kconfig 中添加

```
#vim driver/char/Kconfig
source "drivers/char/ws_driver/Kconfig"
#vim driver/char/Makefile
```

添加

```
obj-y += ws_driver/
```

6.3 配置内核

```
#make menuconfig
Drivers     --->
    Character devices     --->
        ws_add_drivers     --->
            [*] LED support
```

6.4 内核源码根目录下执行

```
#./build_marsboard_rk3066_mtd
```

编译内核，生成 marsboard_rk3066_mtdboot.img，下载新内核到开发板中。

7. PWM 驱动移植

7.1 在 6.1 节中复制的 ws_driver 文件夹中包含动源码 Pwm1。

7.2 配置内核

```
#make menuconfig
Device Drivers     --->
```

```
Character devices      --->
ws_add_drivers      --->
[*] pwm11_test
```

7.3 内核源码根目录下执行

```
#./build_marsboard_rk3066_mtd
```

编译内核，生成 marsboard_rk3066_mtdboot.img，下载新内核到开发板中。

8. DS18B20 的添加

8.1 在 6.1 节中复制的 ws_driver 文件夹中包含动源码 ds18b20.c

8.2 配置内核

```
#make menuconfig
Device Drivers      --->
    Character devices      --->
        ws_add_drivers      --->
            [*] DS18B20 suppor
```

8.3 内核源码根目录下执行

```
#./build_marsboard_rk3066_mtd
```

编译内核，生成 marsboard_rk3066_mtdboot.img，下载新内核到开发板中。

9. SPI CS 的添加

9.1 修改 device.c

```
vim marsboard-rk3066-linux-3.0.8+/arch/arm/mach-rk30/ devices.c
```

9.2 修改第 729 行为

```
#define SPI_CHIPSELECT_NUM 1
```

9.3 注释第 769 到 774 行

```
/*{
    .name = "spi0 cs0",
    .cs_gpio = RK30_PIN1_PA4,
    .cs_iomux_name = GPIO1A4_UART1SIN_SPI0CSN0_NAME,
    .cs_iomux_mode = GPIO1A_SPI0_CSN0,
},
*/
```

9.4 注释第 833 到 838

```
/*{
    .name = "spi1 cs0",
    .cs_gpio = RK30_PIN2_PC4,
    .cs_iomux_name = GPIO2C4_LCDC1DATA20_SPI1CSN0_HSADCDATA1_NAME,
    .cs_iomux_mode = GPIO2C_SPI1_CSN0,
},
*/
```

9.5 在 6.1 节中复制的 ws_driver 文件夹中包含动源码 at45_cs.c

9.6 配置内核

```
#make menuconfig
Device Drivers    --->
    Character devices    --->
        ws_add_drivers    --->
            [*] spi_cs
```

9.7 内核源码根目录下执行

```
#./build_marsboard_rk3066_mtd
```

编译内核，生成 marsboard_rk3066_mtdboot.img，下载新内核到开发板中。

10. RS485 片选的添加

10.1 在 6.1 节中复制的 ws_driver 文件夹中包含动源码 RS485.c。

10.2 配置内核

```
#make menuconfig
Device Drivers      --->
    Character devices      --->
        ws_add_drivers      --->
            [*] RS485 support
```

10.3 内核源码根目录下执行

```
#./build_marsboard_rk3066_mtd
```

编译内核，生成 marsboard_rk3066_mtdboot.img，下载新内核到开发板中。