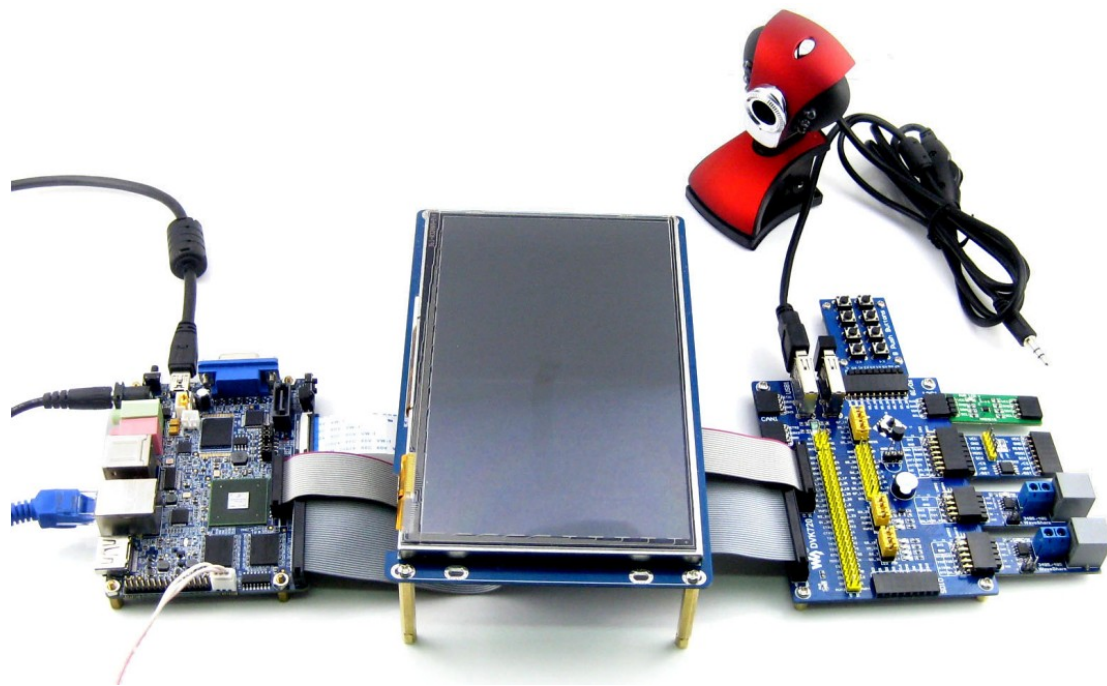


DVK720 扩展板

驱动移植手册

2014. 04. 03 V1. 0



版权声明

本手册所有权由深圳市微雪电子有限公司独家持有。未经本公司的书面许可，不得以任何方式或形式进行修改、分发或复制本文档的任何部分，否则一切后果由违者自负。

版本更新记录

版本	日期	说明
V1.0	2014.04.03	初始发布

目录

版权声明	I
版本更新记录	I
特殊说明	1
第一章 按键驱动移植.....	1
1.1 管脚配置	1
1.2 添加平台资源	2
第二章 PWM 驱动移植	3
2.1 管脚配置	3
2.2 添加平台资源	3
第三章 LED 驱动移植	4
3.1 配置管脚	4
3.2 配置内核	4
第四章 SPI 驱动移植.....	5
4.1 管脚配置	5
4.2 添加平台资源	6
第五章 UART 的移植.....	9
5.1 添加管脚配置	9
第六章 DS18B20 的移植	9
6.1 添加管脚配置	9
6.2 配置内核	10

第七章 RS485 的移植	10
7.1 添加管脚配置	10
7.2 配置内核	11
第八章 USB 网卡驱动移植	11
8.1 添加驱动源码	11
8.2 配置内核	11

特殊说明

为了区别在 PC 机和开发板上的操作，约定为：PC 机的 ubuntu 终端输入的命令前加“#”，并且是 root 用户权限；开发板终端输入的命令前加“\$”。

第一章 按键驱动移植

1.1 管脚配置

1) 在 arch/arm/mach-mx6/board-mx6q_sabresd.h 文件的

mx6q_sabresd_cstm_tq_pads 结构体中，添加管脚的初始化配置。

```
#vi arch/arm/mach-mx6/board-mx6q_sabresd.h
```

```
MX6Q_PAD_EIM_D17__GPIO_3_17,
```

```
MX6Q_PAD_EIM_D30__GPIO_3_30,
```

```
MX6Q_PAD_ENET_TX_EN__GPIO_1_28,
```

```
MX6Q_PAD_EIM_D20__GPIO_3_20,
```

```
MX6Q_PAD_CSI0_DAT10__GPIO_5_28,
```

```
MX6Q_PAD_EIM_D18__GPIO_3_18,
```

```
MX6Q_PAD_ENET_TXD0__GPIO_1_30,
```

```
MX6Q_PAD_EIM_D23__GPIO_3_23,
```

- 2) 在 arch/arm/mach-mx6/board-mx6q_sabresd.c 配置文件中添加宏定义:

#vi arch/arm/mach-mx6/board-mx6q_sabresd.c

```
#define SABRESD_BUTTON1 IMX_GPIO_NR(3,17)
```

```
#define SABRESD_BUTTON2 IMX_GPIO_NR(3,30)
```

```
#define SABRESD_BUTTON3 IMX_GPIO_NR(1,28)
```

```
#define SABRESD_BUTTON4 IMX_GPIO_NR(3,20)
```

```
#define SABRESD_BUTTON5 IMX_GPIO_NR(5,28)
```

```
#define SABRESD_BUTTON6 IMX_GPIO_NR(3,18)
```

```
#define SABRESD_BUTTON7 IMX_GPIO_NR(1,30)
```

```
#define SABRESD_BUTTON8 IMX_GPIO_NR(3,23)
```

1.2 添加平台资源

- 1) 在 board-mx6q_sabresd.c 中添加

#vi arch/arm/mach-mx6/board-mx6q_sabresd.c

```
static struct gpio_keys_button sabresd_buttons[] = {
```

```
GPIO_BUTTON(SABRESD_BUTTON1, KEY_1, 1, "button1", 0, 1),
```

```
GPIO_BUTTON(SABRESD_BUTTON2, KEY_2, 1, "button2", 0, 1),
```

```
GPIO_BUTTON(SABRESD_BUTTON3, KEY_3, 1, "button3", 0, 1),
```

```
GPIO_BUTTON(SABRESD_BUTTON4, KEY_4, 1, "button4", 0, 1),
```

```
GPIO_BUTTON(SABRESD_BUTTON5, KEY_5, 1, "button5", 0, 1),
```

```
GPIO_BUTTON(SABRESD_BUTTON6, KEY_6, 1, "button6", 0, 1),
```

```
GPIO_BUTTON(SABRESD_BUTTON7, KEY_7, 1, "button7", 0, 1),
GPIO_BUTTON(SABRESD_BUTTON8, KEY_8, 1, "button8", 0, 1),
}
```

第二章 PWM 驱动移植

2.1 管脚配置

在 arch/arm/mach-mx6/board-mx6q_sabresd.h 文件的 mx6q_sabresd_cstm_tq_pads 结构体中，添加管脚的初始化配置。

```
#vi arch/arm/mach-mx6/board-mx6q_sabresd.h
MX6Q_PAD_SD1_CMD__PWM4_PWMO,
```

2.2 添加平台资源

在 board-mx6q_sabresd.c 中添加平台资源。

```
#vi arch/arm/mach-mx6/board-mx6q_sabresd.c
static struct platform_pwm_backlight_data
mx6_marsboard_pwm3_backlight_data = {
    .pwm_id = 3,
    .max_brightness = 255,
    .dft_brightness = 128,
    .pwm_period_ns = 50000,
};
```

```
static void __init mx6_sabresd_board_init(void){  
  
    imx6q_add_mxc_pwm_backlight(3,&mx6_marsboard_pwm3_backlight_data);  
  
}
```

第三章 LED 驱动移植

3.1 配置管脚

在 board-mx6q_sabresd.h 文件的

mx6q_sabresd_cstm_tq_pads 结构体中，添加管脚的初始化配置。

```
#vi arch/arm/mach-mx6/board-mx6q_sabresd.h
```

```
MX6Q_PAD_CSI0_PIXCLK__GPIO_5_18,
```

```
MX6Q_PAD_CSI0_VSYNC__GPIO_5_21,
```

```
MX6Q_PAD_CSI0_DAT17__GPIO_6_3,
```

```
MX6Q_PAD_CSI0_DAT13__GPIO_5_31,
```

3.2 配置内核

拷贝源码/ws_driver 的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。

在 Kconfig 中添加

```
#vi driver/char/Kconfig
```

```
source "drivers/char/ws_driver/Kconfig"
```

```
#vi driver/char/Makefile
```

添加

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

```
#make menuconfig
```

```
Device Drivers --->
```

```
Character devices --->
```

```
ws_add_drivers --->
```

```
[*] LED support
```

第四章 SPI 驱动移植

4.1 管脚配置

在 board-mx6q_sabresd.h 文件的

mx6q_sabresd_cstm_tq_pads 结构体中，添加管脚的初始化配置。

```
#vi arch/arm/mach-mx6/board-mx6q_sabresd.h
```

```
MX6Q_PAD_EIM_CS0__ECSPI2_SCLK,
```

```
MX6Q_PAD_EIM_CS1__ECSPI2_MOSI,
```

```
MX6Q_PAD_EIM_OE__ECSPI2_MISO,
```

```
MX6Q_PAD_CSI0_DAT11__ECSPI2_SS0,
```

4.2 添加平台资源

1) 添加 SPI 平台资源

```
#vi arch/arm/mach-mx6/board-mx6q_sabresd.h
```

```
static int mx6q_sabrelite_spi2_cs[] = {  
    MX6Q_SABRELITE_ECSPi2_CS1,  
};
```

```
static const struct spi_imx_master mx6q_sabresd_spi2_data  
__initconst = {  
    .chipselect = mx6q_marsboard_spi2_cs,  
    .num_chipselect = ARRAY_SIZE(mx6q_marsboard_spi2_cs),  
};
```

```
imx6q_add_ecspi(0, &mx6q_sabrelite_spi2_data);
```

```
static struct mtd_partition imx6_sabrelite_spi_nor_partitions[] = {  
    {  
        .name = "bootloader",  
        .offset = 0,  
        .size = 0x00100000,  
    },  
    {  
        .name = "kernel",
```

```

        .offset = MTDPART_OFS_APPEND,

        .size = MTDPART_SIZ_FULL,
    },
};

static struct flash_platform_data imx6_sabrelite__spi_flash_data = {

    .name = "spidev",

    .parts = imx6_sabrelite_spi_nor_partitions,

    .nr_parts = ARRAY_SIZE(imx6_sabrelite_spi_nor_partitions),

    .type = "sst25vf016b",
};

static struct spi_board_info imx6_sabrelite_spi_nor_device[]
__initdata = {

    {

        .modalias = "spidev",

        .max_speed_hz = 12000000, /* max spi clock (SCK) speed
in HZ */

        .bus_num = 1,

        .chip_select = 0,

        .platform_data = &imx6_sabrelite__spi_flash_data,
    },
};

```

```
static void spi_device_init(void)
{
    spi_register_board_info(imx6_sabrelite_spi_nor_device,
        ARRAY_SIZE(imx6_sabrelite_spi_nor_device));
}
```

2) 配置内核

#make menuconfig

```
Device Drivers --->
    [*] SPI support --->
        <*> User mode SPI device driver support
```

3) 添加 SPI 片选信号驱动

3.2 配置内核章节拷贝的 `ws_driver` 文件夹包含的了 `spi` 片选信号驱动源码 `spi_io.c`。

在内核源码顶层目录下执行

#make menuconfig

```
Device Drivers --->
    Character devices --->
        ws_add_drivers --->
            [*] spi2_cs
```

第五章 UART 的移植

5.1 添加管脚配置

1) 在 board-mx6q_sabresd.h 文件的 mx6q_sabresd_cstm_tq_pads 结构体中，添加管脚的初始化配置。

```
#vi arch/arm/mach-mx6/board-mx6q_sabresd.h
```

```
MX6Q_PAD_CSI0_DAT14__UART5_TXD,
```

```
MX6Q_PAD_CSI0_DAT15__UART5_RXD,
```

2) 注释掉 static iomux_v3_cfg_t mx6q_sabresd_csi0_sensor_pads[] 结构体中的如下代码

```
MX6Q_PAD_CSI0_DAT12__IPU1_CSI0_D_12,
```

```
MX6Q_PAD_CSI0_DAT13__IPU1_CSI0_D_13,
```

```
MX6Q_PAD_CSI0_DAT14__IPU1_CSI0_D_14,
```

```
MX6Q_PAD_CSI0_DAT15__IPU1_CSI0_D_15,
```

```
MX6Q_PAD_CSI0_DAT16__IPU1_CSI0_D_16,
```

```
MX6Q_PAD_CSI0_DAT17__IPU1_CSI0_D_17,
```

第六章 DS18B20 的移植

6.1 添加管脚配置

1) 3.2 配置内核章节拷贝的 ws_driver 文件夹中包含动源码 ds18b20.c。

在 board-mx6q_sabresd.h 文件的

mx6q_sabresd_cstm_tq_pads 结构体中，添加管脚的初始化配置。

```
#vi arch/arm/mach-mx6/board-mx6q_sabresd.h
```

```
MX6Q_PAD_EIM_D21__GPIO_3_21,
```

6.2 配置内核

```
#make menuconfig
```

```
Device Drivers --->
```

```
Character devices --->
```

```
ws_add_drivers --->
```

```
[*] DS18B20 support
```

第七章 RS485 的移植

7.1 添加管脚配置

1) 3.2 配置内核章节拷贝的 ws_driver 文件夹中包含动源码 RS485.c。

在 board-mx6q_sabresd.h 文件的

mx6q_sabresd_cstm_tq_pads 结构体中，添加管脚的初始化配置。

```
#vi arch/arm/mach-mx6/board-mx6q_sabresd.h
```

```
MX6Q_PAD_CSI0_DAT18__GPIO_6_4,
```

```
MX6Q_PAD_CSI0_DAT19__GPIO_6_5,
```

7.2 配置内核

```
#make menuconfig
```

```
Device Drivers --->
```

```
Character devices --->
```

```
ws_add_drivers --->
```

```
[*] RS485 support
```

第八章 USB 网卡驱动移植

8.1 添加驱动源码

- 1) 拷贝源码/rtl8192cu 到内核 drivers/net/wireless，并修改相应的 Makefile 和 Kconfig 文件：

```
#vi drivers/net/wireless/Makefile
```

添加

```
obj-$(CONFIG_RTL8192CU) += rtl8192cu/
```

```
#vi drivers/net/wireless/Kconfig
```

添加

```
source "drivers/net/wireless/rtl8192cu/Kconfig"
```

8.2 配置内核

```
#make menuconfig
```

```
Device Drivers --->
```

[*] Network device support --->

[*] Wireless LAN --->

< * > RtalteX 8192c usb wifi