

RK616（JettaB）开发文档

	文件标识	
	当前版本	V1.1
	作者	yxj、 yj、 xhc、 lb
	完成日期	2013-05-21

说明：该文档适用于 RK616 及其后续相关兼容芯片

目录

1、概述.....	3
2、程序框架及实现.....	3
2.1、核心控制部分代码.....	3
2.2、lvds 驱动代码.....	3
2.3、MIPI 驱动代码.....	3
2.4、HDMI 驱动代码.....	3
2.5、CODEC 驱动相关代码.....	4
2.6、板级配置参考代码.....	4
3、makemenuconfig 相关配置.....	4
4、板级配置.....	5
5. 系统启动相关 log 及 debug.....	9

1、概述

JettaB 是为 RK 平台设计的一颗配套芯片，他包含 LVDS、HDMI、MIPI、CODEC 等模块。该芯片对外提供如下接口：

- 两个 RGB 接口 LCD0、LCD1，其中 LCD0 只能用作输入，LCD1 是双向口，可以用作输入也可以用作输出。
- 一个 LVDS 输出口，该接口支持双通道 LVDS，并可复用为 RGB 输出接口，在软件上可配置，RK616 在该通道上不支持 RGB 输出。
- 一个 MIPI 输出接口，RK616 目前不支持该接口，将在后续封装的芯片上支持。
- 一个 HDMI 输出接口。
- 2 个 I2S/PCM 接口。一个是 4 通道，另一个单通道。
- 一个 I2C 控制接口，I2C 地址为 0x50(该地址是不带读写标志位的 7bit i2c 地址)。

2、程序框架及实现

由于 JettaB 是多功能设备，所以在软件上按照 MFD 框架组织相关驱动。

2.1、核心控制部分代码

```
drivers/mfd/rk616-core.c  
drivers/mfd/rk616-vif.c
```

2.2、lvds 驱动代码

```
drivers/video/rockchip/transmitter/rk616_lvds.c
```

2.3、MIPI 驱动代码

```
drivers/video/rockchip/transmitter/rk616_mipi_dsi.c
```

2.4、HDMI 驱动代码

```
drivers/video/rockchip/hdmi/chips/rk616/
```

2.5、CODEC 驱动相关代码

```
sound/soc/codecs/rk616_codec.c
sound/soc/codecs/rk616_codec.h
sound/soc/rk29/rk_rk616.c
```

2.6、板级配置参考代码

```
arch/arm/mach-rk30/board-rk3168-tb.c
```

3、makemenuconfig 相关配置

(1) 由于 jettaB 为 MFD 设备，所以首先要做如下选项

```
Device Drivers --->
[*] Multifunction device drivers --->
  [*] RK616(Jetta B) Multifunction device support
    [*] Enable 12M clock for timing reconstruction
      [ ] RK616(JettaB) debug enable
```

只有配置了该选项，后续 LVDS、HDMI、MIPPI、CODEC 的选项才会显示。

利用 12M MCLK 进行信号重建可以提高 HDMI 的稳定性，所以在使用 HDMI 的项目中要打开该选项，并且要确保在 HDMI 接入的时候 MCLK(i2s0)的时钟为 12M.

如果开发过程中遇到了问题，需要显示相关 DEBUG 信息，请打开 debug enable 配置，该选项默认不打开，以减少相关 log 的打印。

(2) LVDS 配置

```
Graphics support --->
Display device support --->
  <*> Frame buffer support for Rockchip --->
    [*] RockChip display transmitter support --->
      [*] RK616(JettaB) lvds,lcd,scaler vido interface support
```

如果有使用 JettaB 输出显示信号（RGB/LVDS）到 LCD 屏幕，要选上该选项。

(3) HDMI 配置

```
Graphics support --->
  [*] Rockchip HDMI support --->
    [*] RK616 HDMI support
```

如果有使用 JettaB 的 HDMI 功能，需要配上该选项

(4) CODEC 配置

```
Device Drivers --->
  <*> Sound card support --->
  <*> Advanced Linux Sound Architecture --->
```

```
<*> ALSA for SoC audio support --->
    Set audio support for HDMI (HDMI use SPDIF) --->
<*>SoC I2S Audio support for rockchip - RK616
```

如果有使用 jettaB 的 CODEC 功能，需要配上该选项。

如果在 HDMI 的时候，开启了 12M 信号重建功能，HDMI 状态下的音频要用 SPDIF 实现，所以要选中 HDMI use SPDIF 选项。

(5) MIPI 配置

如果使用 MIPI 接口，需要配置下面选项

```
Device Drivers --->
Graphics support --->
    <*> Frame buffer support for Rockchip --->
        [*] RockChip display transmitter support --->
            [*] Rockchip MIPI DSI support
    <*> RK616(JettaB) mipi dsi support
```

4、板级配置

板级配置以 board-rk3168-tb.c 为例，相关代码如下：

```
#if defined(CONFIG_MFD_RK616)
    #include <linux/mfd/rk616.h> //head file
#endif
#if defined(CONFIG_MFD_RK616)
#define RK616_RST_PIN        RK30_PIN3_PB2
#define RK616_PWREN_PIN      RK30_PIN0_PA3
#define RK616_SCL_RATE       (100*1000) //i2c scl rate
static int rk616_power_on_init(void)
{
    int ret;

    if(RK616_PWREN_PIN != INVALID_GPIO)
    {
        ret = gpio_request(RK616_PWREN_PIN, "rk616 pwren");
        if (ret)
        {
            printk(KERN_ERR "rk616 pwrengpio request fail\n");
        }
        else
        {
            gpio_direction_output(RK616_PWREN_PIN, GPIO_HIGH);
        }
    }
}
```

```

    if(RK616_RST_PIN != INVALID_GPIO)
    {
        ret = gpio_request(RK616_RST_PIN, "rk616 reset");
        if (ret)
        {
            printk(KERN_ERR "rk616 reset gpio request fail\n");
        }
        else
        {
            gpio_direction_output(RK616_RST_PIN, GPIO_HIGH);
            msleep(2);
            gpio_direction_output(RK616_RST_PIN, GPIO_LOW);
            msleep(10);
            gpio_set_value(RK616_RST_PIN, GPIO_HIGH);
        }
    }

    return 0;

}

static int rk616_power_deinit(void)
{
    gpio_set_value(RK616_PWREN_PIN, GPIO_LOW);
    gpio_set_value(RK616_RST_PIN, GPIO_LOW);
    gpio_free(RK616_PWREN_PIN);
    gpio_free(RK616_RST_PIN);

    return 0;
}

static struct rk616_platform_data rk616_pdata = {
    .power_init = rk616_power_on_init,
    .power_deinit = rk616_power_deinit,
    .scl_rate    = RK616_SCL_RATE,
    .lcd0_func = INPUT,                //port lcd0 as input
    .lcd1_func = INPUT,                //port lcd1 as input
    .lvds_ch_nr = 1,                  //the number of used lvds channel
    .hdmi_irq = RK30_PIN2_PD6,
    .spk_ctl_gpio = RK30_PIN2_PD7,
    .hp_ctl_gpio = RK30_PIN2_PD7,
};
#endif

```

```

#ifdef CONFIG_I2C4_RK30
static struct i2c_board_info __initdata i2c4_info[] = {
#ifdef CONFIG_MFD_RK616
{
    .type          = "rk616",
    .addr          = 0x50,
    .flags         = 0,
    .platform_data = &rk616_pdata,
},
#endif
};
#endif

```

其中 RK616_RST_PIN 、RK616_PWREN_PIN 为 JettaB 复位和电源控制 pin 脚，需要根据硬件设计进行相应的定义。

RK616_SCL_RATE 是 I2C 通信的速率定义，默认设置为 100k，如果板子质量不好，导致 I2C 通信出错，可以相应降低速率。

rk616_power_on_init 和 rk616_power_deinit 是 jettaB 对应的上电和下电控制函数。它们分别在系统启动和关机的时候被调用。如果硬件上有特殊的设计，可以根据实际需要调整这两个函数。

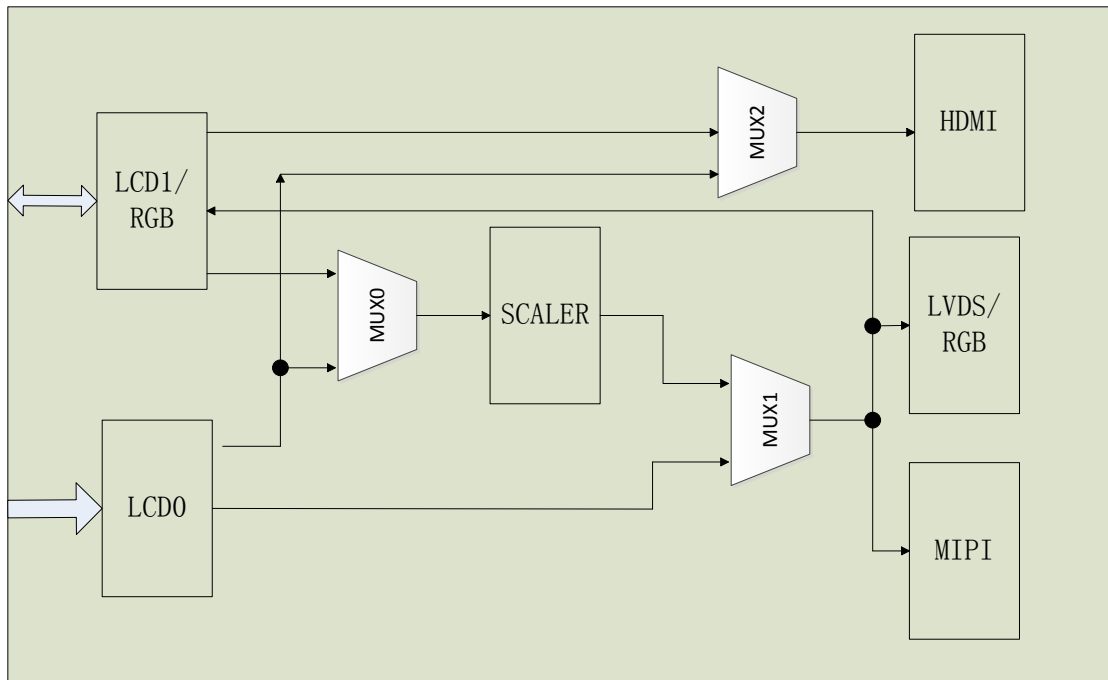
hdmi_irq 是 HDMI 检测中断引脚，如果该值定义为对应的 GPIO pin 脚，则使用中断的方式检测 HDMI 插拔，如果定义为 INVALID_GPIO 则使用轮询的方式检测 HDMI 插拔。

spk_ctl_gpio 为 codec speaker 输出使能管脚

hp_ctl_gpio 是耳机输出使能管脚。

lcd0_func、lcd1_func、lvds_ch_nr 为显示相关的配置选项。这个需要根据硬件连接和显示需要进行相关设置。

JettaB 显示系统框架图如图（1）所示。



图（1）JettaB 显示系统框图

如第一部分概述所描述，JettaB 内部集成了两个 LCD 口（LCD0、LCD1），一个 LVDS/RGB 复用接口，一个 MIPI 接口，一个 HDMI 接口。为软硬件提供了灵活的设计支持。

其中 LCD0 是单向接口，只接收主控输出的 RGB 数据，LCD1 为双向接口，可以接收主控输出的 RGB 数据，还可以作为输出口，输出数据给 RGB 接口的屏幕。所以我们在显示上有两种实现方案：

（1）双 LCDC 实现双显，显示通路如下

LCD1——》MUX2——》HDMI
LCD0——》MUX1——》LVDS/MIPI

（2）一个 LCDC 加 scaler 实现 双显，这个根据屏幕接的位置，有两种显示通路

a. 屏幕接在 LVDS 或者 MIPI 接口上

LCD0——》MUX2——》HDMI
LCD0——》MUX0——》SCALER——》MUX1——》LVDS/MIPI

b. RGB 屏幕接在 LCD1 上

LCD0——》MUX2——》HDMI
LCD0——》MUX0——》SCALER——》MUX1——》LCD1/RGB

对于这种使用 scaler 实现双显的方案，需要使用 jettaB_scaler_config 工具计算各种 HDMI 分辨率下对应的 pll、scaler 配置参数。然后填写到屏幕配置文件中，可参考 86v 的屏幕配置文件，或者 RK30 SDK 板子上的屏幕配置文

件 lcd_b101ew05.c.

LVDS 接口支持双通道和单通道 LVDS，该选项通过 lvds_ch_nr 配置，对于使用单通道 LVDS 的屏幕，该值设置为 1，对于使用双通道 LVDS 的屏幕，改值设置为 2，如果该接口不使用则设置为 0。在使用单通道 LVDS 的时候，只能使用 lvds 的 channel0。

LCD0 和 LCD1 的使用情况和功能，通过 lcd0_func 和 lcd1_func 设置。

由于 LCD0 是单向接口，所以如果 LCD0 有和主控的 LCDC 相连接，则 LCD0 设置为 INPUT，如果 LCD0 没有和主控的 LCDC 相连接，则设置为 UNUSED，即未使用。

LCD1 口是双向口，所以 LCD1 口有三种值，如果 LCD1 和主控的 LCDC 相连接，则 LCD1 作为输入口，接收主控输出的 RGB 数据，则 LCD1 应设置为 INPUT。如果 LCD1 口外接 RGB 屏幕，则 LCD1 作为输出口，他把 LCD0 口输入的数据输出到 RGB 屏幕，LCD1 应设置为 OUTPUT。如果 LCD1 没有使用，则设置为 UNUSED。

另外，对于使用到 jettaB 的 LVDS 或者 RGB 接口的项目，需要在屏幕配置文件中的 set_lcd_info 函数中做如下配置：

```
#if defined(CONFIG_RK610_LVDS) || defined(CONFIG_RK616_LVDS)
    screen->:sscreen_set = rk610_lcd_scaler_set_param;
#endif
}
```

更加详细的修改，请参考 lcd_b101ew05.c 中和 CONFIG_RK616_LVDS 宏相关的修改。

5. 系统启动相关 log 及 debug

下面是打开 Debug 选项后的正常内核 log:

```
[ 0.542737] rk616-lvds rk616-lvds: rk616 lvds probe success!
[ 0.542971] rk616 4-0050: rk616 core probe success!
[ 0.543199] Switching to clocksource rk_timer
[ 0.548487] Switched to NOHz mode on CPU #0
[ 0.549130] Switched to NOHz mode on CPU #2
[ 0.549152] Switched to NOHz mode on CPU #1
[ 0.549165] Switched to NOHz mode on CPU #3
[ 0.550305] lcdc1:reg_phy_base = 0x1010e000,reg_vir_base:0xf70a0000
[ 0.550387] fb0:win0
```

```

[ 0.550392] fb1:win1
[ 0.550396] fb2:win2
[ 0.550484] rk3188 lcdc1 clk enable...
[ 0.618344] rk3188 lcdc1 clk disable...
[ 0.647998] fb0:phy:90c00000>>vir:f8000000>>len:0xc00000
[ 0.648259] rk_fb_register>>>>fb0
[ 0.648538] rk_fb_register>>>>fb1
[ 0.648585] rk3188 lcdc1 clk enable...
[ 0.648672] lcdc1: dclk:66500000>>fps:58
[ 0.648707] rk616 4-0050: rk616 use dual input for dual display!
[ 0.683986] rk616 4-0050: rk616 vif0 disable
[ 0.723983] rk616 4-0050: rk616 vif1 disable
[ 0.726122] rk616 4-0050: rk616 use single lvds channel.....

```

```

[ 2.038984] asoc: rk616-hifi <-> rk29_i2s.1 mapping ok
[ 2.043738] asoc: rk616-voice <-> rk29_i2s.1 mapping ok
[ 2.044191] ALSA device list:
[ 2.044214] #0: RK_RK616

```

错误 log:

```

[ 1.563775] ALSA device list:
[ 1.563793] No soundcards found.

```

出现这种情况: 如果 I2C 没有报错, 可以确定是 sound/soc/rk29/rk_rk616.c 中的:

```

232 static struct snd_soc_dai_link rk_dai[] = {
233     {
234         .name = "RK616 I2S1",
235         .stream_name = "RK616 PCM",
236         .codec_name = "rk616-codec.4-0050",
237         .platform_name = "rockchip-audio",
238         .cpu_dai_name = "rk29_i2s.1",
239         .codec_dai_name = "rk616-hifi",
240         .init = rk616_init,
241         .ops = &rk616_hifi_ops,
242     },
243     {
244         .name = "RK616 I2S2",
245         .stream_name = "RK616 PCM",
246         .codec_name = "rk616-codec.4-0050",
247         .platform_name = "rockchip-audio",
248         .cpu_dai_name = "rk29_i2s.1",
249         .codec_dai_name = "rk616-voice",
250         .ops = &rk616_voice_ops,
251     },
252 }

```

rk616-codec.x-0050:x 为实际设备所在的 I2C 通道。

rk29_i2s.1: 为 I2S 的通道选择。

如果 I2S 配置为 CONFIG_SND_RK29_SOC_I2S_2CH: 则是 rk29_i2s.1

如果 I2S 配置为 CONFIG_SND_RK29_SOC_I2S_8CH: 则是 rk29_i2s.0

如果 I2S 配置为 CONFIG_SND_RK_SOC_I2S2_2CH: 则为 rk29_i2s.2