

RK610 配置

版本	日 期	描 述	作 者	审核
1.0	2012-03-8	RK610Hdmi、LCD、LVDS、TVE、CODEC 简要配置	姚智情、郑 阳、邱建斌	
1.1	2012-3-30	更新 Rk610 的代码框架及一些配置设定	姚智情	
1.2	2012-4-18	修复一些 bug 及优化	姚智情	

目录

1. 版本历史.....	2
2. RK610.....	2
3. RK610 HDMI 及与 LVDS 或 LCD 双屏设置.....	4
4. LVDS、LCD 与 hdmi 双屏配置.....	5
5. Lcd 双屏调试.....	9
6. RK610 TVE.....	10

1. 版本历史

V1.0 2012-3-8

RK610 hdmi、tvout、lvds、lcd1、codec 模块功能初版。

V1.1

更新 Rk610 的代码框架及一些配置设定。

V1.2

修复问题：

- 1、hdmi 1080p 切换分辨率概率性无显示
- 2、lvds 双屏到 bypass 单屏显示时，黑屏，无输出
- 3、加入 lcd、lvds 睡眠唤醒处理（主要是关 lvds 模块，lcd 功耗影响不大）
- 4、优化 hdmi 电源管理
- 5、修复 edid 读取时的 bug,加入解析 edid，判断电视是否支持该分辨率。

2. RK610

应用平台：RK2918 kernel develop-3.0

驱动文件：

drivers/mfd/rk610-core.c	RK610 主要控制
drivers/video/hdmi/chips/rk610	rk610 hdmi 驱动
drivers/video/display/lcd/rk610_lcd.c	rk610 LCD 控制
drivers/video/display/screen/lcd_hdmi_xxx.c	双屏显示的 lcd 时序定义
drivers/video/display/tve	TVE 驱动
sound/soc/codecs/rk610_codec.c	codec 核心驱动代码
sound/soc/codecs/rk610_codec.h	头文件定义
sound/soc/rk29/rk29_jetta_codec.c	平台设置代码，主要设置 I2S clk

Board-xxx.c 配置：

```
#ifdef CONFIG_MFD_RK610
{
    .type          = "rk610_ctl",
    .addr          = 0x40,
    .flags         = 0,
},
#endif
#ifdef CONFIG_RK610_TVOUT
{
    .type          = "rk610_tvout",
    .addr          = 0x42,
```

```

        .flags          = 0,
    },
#endif
#ifdef CONFIG_RK610_HDMI
    {
        .type            = "rk610_hdmi",
        .addr            = 0x46,
        .flags           = 0,
        .irq             = RK29_PIN1_PD7,
    },
#endif
#ifdef CONFIG_SND_SOC_RK610
    {
        .type            = "rk610_i2c_codec",
        .addr            = 0x60,
        .flags           = 0,
    },
#endif
#endif

```

各个功能宏配置:

1、RK610 功能支持宏

```

-> Device Drivers
  -> Multifunction device drivers (MFD_SUPPORT [=y])
    [*] RK610(Jetta) Power Management chip
    [*] RK610(Jetta) Multimedia support

```

2、Lcd、tvout

```

-> Device Drivers
  -> Graphics support
    -> Display device support
      LCD Panel Select (RGB Hannstar LCD_HDMI_800x480)
    [*] RK610(Jetta) lcd support (NEW)
    [*] RK610(Jetta) tvout support (NEW)
    [ ] support YPbPr output (NEW)
    [ ] support CVBS output (NEW)

```

3、Hdmi

```

[*] HDMI support
  HDMI chips select (RK610(Jetta) hdmi support)
  [ ] enable hdmi save data
  [*] dual display support

```

4、Codec

```

-> Device Drivers
  -> Sound card support (SOUND [=y])
    -> Advanced Linux Sound Architecture (SND [=y])
      -> ALSA for SoC audio support (SND_SOC [=y])
        <*> SoC I2S Audio support for rockchip - RK610

```

调试接口:

Debug 开关:

在 include/linux/mfd/rk610-core.h 中: #define RK610_DEBUG

在调试的 sys 节点:

busybox find /sys -name reg_ctl

./sys/devices/platform/rk29_i2c.1/i2c-1/1--40/reg_ctl //codec 部分控制 lcd lvds 控制

./sys/devices/platform/rk29_i2c.1/i2c-1/1--46/reg_ctl //hdmi 控制寄存器

cat reg_ctl //打印寄存器中的值

echo 0924 >reg_ctl //向寄存器的 09 地址写值 24

```
shell@android:/ # busybox find -name reg_ctl
./sys/devices/platform/rk29_i2c.2/i2c-2/2-0040/reg_ctl
./sys/devices/platform/rk29_i2c.2/i2c-2/2-0046/reg_ctl
shell@android:/ # cat ./sys/devices/platform/rk29_i2c.2/i2c-2/2-0040/reg_ctl
>>>rk610_ctl 0: 22 51 51 b8 e 1 42 64 5 24 ff 2 10 0 0 0
>>>rk610_ctl 10: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
>>>rk610_ctl 20: 0 0 0 0 0 0 0 0 0 0 0 0 7 7 0
>>>rk610_ctl 30: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
>>>rk610_ctl 40: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
>>>rk610_ctl 50: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
>>>rk610_ctl 60: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
>>>rk610_ctl 70: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
>>>rk610_ctl 80: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
>>>rk610_ctl 90: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
>>>rk610_ctl a0: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
>>>rk610_ctl b0: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
>>>rk610_ctl c0: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
>>>rk610_ctl d0: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
>>>rk610_ctl e0: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
shell@android:/ # cat ./sys/devices/platform/rk29_i2c.2/i2c-2/2-0046/reg_ctl #
>>>rk610_hdmi 0: 5 1 34 81 8 0 22 0 0 0 0 0 0 0 0
>>>rk610_hdmi 10: 0 0 0 0 0 0 0 0 4 0 6 4d 0 0 2 ca
>>>rk610_hdmi 20: 4 0 11 df 10 c0 0 54 4 0 0 0 7 6c 2 ee
>>>rk610_hdmi 30: 0 0 0 0 0 1 40 40 3c 0 f 0 0 0 0
>>>rk610_hdmi 40: 18 80 0 0 0 0 0 0 0 0 63 40 0 0 2
>>>rk610_hdmi 50: a 0 12 8 0 6 0 20 0 7f 0 0 0 0 0
>>>rk610_hdmi 60: 0 0 12 26 2c 0 0 0 0 0 10 0 0 0 0
>>>rk610_hdmi 70: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
>>>rk610_hdmi 80: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
>>>rk610_hdmi 90: 0 0 0 0 0 0 0 0 0 0 0 0 0 1 8
>>>rk610_hdmi a0: 84 1 a 0 11 9 0 0 1 0 0 0 0 0 0
>>>rk610_hdmi b0: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
>>>rk610_hdmi c0: c0 0 78 0 0 0 0 0 50 70 0 1 0 1 0
>>>rk610_hdmi d0: 0 0 0 3 3 9 3 0 ff ff 0 0 d0 20 0 0
>>>rk610_hdmi e0: 0 8c 4 2 3 1c 2c 4 10 0 0 0 0 0 0
>>>rk610_hdmi f0: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
shell@android:/ #
```

3. RK610 HDMI 及与 LVDS 或 LCD 双屏设置

1. 内核选项

RK610 HDMI 模块输出 1080p、720p、576p、480p

2. 应用程序操作

Hdmi 在 linux sysfs 下注册 sys 节点。

/sys/class/hdmi/hdmi-0

注: 该节点的权限为 664, 普通的 apk 权限无法写该节点

1、查看/sys/class/hdmi/hdmi-0 设备节点是否存在, 判定系统是否支持 hdmi

2、查看/sys/class/hdmi/hdmi-0/state

display_on = x

标明 hdmi 显示模式: 0: disp_on_lcd 1: disp_on_hdmi 2: disp_on_lcd_and hdmi

plug = x

标明 hdmi 插拔状态, 0: 拔出 1: 插入

dual_disp = x

标明是否支持双屏 0: 支持 1: 不支持

scale = x

标明缩放大小 80-100

scale_set = x

设置缩放大小, echo scale_set=85 >state 即将图像缩成原来的 85%

resolution = x

设置显示分辨率, echo resolution=1 >state 即将输出分辨率改为 1080p60hz

resolution support:\n"

"HDMI_1920x1080p_50Hz 0\n"

"HDMI_1920x1080p_60Hz 1\n"

"HDMI_1280x720p_50Hz 2\n"

"HDMI_1280x720p_60Hz 3\n"

"HDMI_720x576p_50Hz_4x3 4\n"

"HDMI_720x576p_50Hz_16x9 5\n"

"HDMI_720x480p_60Hz_4x3 6\n"

"HDMI_720x480p_60Hz_16x9 7\n"

3、/sys/class/hdmi/hdmi-0/enable

echo 0 >enable lcd 显示, 关闭 hdmi,

echo 1 >enable 关闭 lcd 显示, hdmi 电视显示。

echo 2 >enable lcd 与 hdmi 双屏显示。

插入时 hdmi 线时,

29 lcdc 输出 hdmi 时序, 分成两路, 一路 hdmi 输出, 一路经 scaler 给 lvds 或 lcd1, 实现双屏;

拔出 hdmi 线时, scaler bypass。

4. LVDS、LCD 与 hdmi 双屏配置

$$(Hin_total * Vin_act / Fin) == (Hout_total * Vout_act / Fout)$$

这个计算是有效显示时间, 误差范围: 8line/total line in

$$T_BP_in = Hin_total * (VBP_in + VPW_in) / Fin$$

$$T_BP_out = Hout_total * (VBP_out + VPW_out) / Fout$$

$$Vst_in = T_BP_in$$

$$Vst_out = (Fvst * Hout_total + Fhst) + T_BP_out$$

根据 PLL 输出的 Fout 的实际值, 判断(1)等式左右两边的值大小。如果左边小, 则输出有效行的起始位置为 $Vst_out = Vst_in + 2 * Htotal_in * 1 / Fin$ (如果缩小大于 2 倍, 则为 $Vst_out = Vst_in + 4 * Htotal_in * 1 / Fin$), 如果右边小, 则输出有效行的起始位置为 $Vst_out = Vst_in + 5 * Htotal_in * 1 / Fin$ (如果缩小大于 2 倍, 则为 $Vst_out = Vst_in + 10 * Htotal_in * 1 / Fin$)。

上述是计算 scaler 时序的起始位置。

实际计算:

根据 scale pll, 选择所需的 pll clk, 即 Fout.

$$Hout_total = Hin_total * Vin_act * Fout / (Fin * Vout_act)$$

$$Vout_total = Fout / (Hout_total * HZ)$$

$$Frame_st = (Hin_total * (VBP_in + VPW_in + X) * Fout) / Fin$$

$$Fvst + (Hout_total * (VBP_out + VPW_out)) = Frame_st / Hout_total \text{ (Frame_st 是整数部分)}$$

为 Fvst)

```
Fhst = (Frame_st%Hout_total)*Hout_total (Frame_st 是小数部分为 Fhst).
If (Hin_total*Vin_act/Fin)< (Hout_total*Vout_act/Fin
X = 2
Else (Hin_total*Vin_act/Fin) >=(Hout_total*Vout_act/Fin
X = 5
If Vin_act/Vout_act >= 2
X =X*2
```

根据最后所得的 Fout、Hout_total、Vout_total、Fvst、Fhst 填写屏驱动。

实例：

```
Hdmi 时序
/* 1080p@60Hz Timing */
#define OUT_CLK4      148500000
#define H_PW5        44
#define H_BP5         148
#define H_VD5         1920
#define H_FP5         88
#define V_PW5         5
#define V_BP5         36
#define V_VD5         1080
#define V_FP5         4

Lcd 时序
#define OUT_CLK        330000000
/* Timing */
#define H_PW           1
#define H_BP           88
#define H_VD           800
#define H_FP           40

#define V_PW           3
#define V_BP           29
#define V_VD           480
#define V_FP           13
```

选择 clk

```
clk = 33.000000 m=16 n=9 no=8
Hout_total = (44+148+1920+88)*1080*33000000/(148500000*480) = 1100
Vout_total = 33000000/(1100*60)= 500
```

因为 $2200*1080/148500000 == 1100*480/330000000$, $1080/480 > 2$

所以 $X = 10$.

```
Frame_st = (Hin_total*(VBP_in +VPW_in+ X)*Fout)/Fin
           = ((2200*(36+5+10)*330000000)/148500000
           = 24933
```

设调整值 $k= 2200$ (计算结果与实际略微不同, 需实际调整) .

```
Fvst + (V_PW+V_BP) = (Frame_st+k)/Hout_total
                   = (24933+2200) / 1100 (求整)
                   = 24
```

```
Fhst = ((Frame_st+k)%Hout_total)*Hout_total
      = (22000%1100) * 1100 (求余)
```

= 733

重新设定 LCD 时序, 括号内为实际调整结果

```
#define S_OUT_CLK      SCALE_RATE(148500000, 330000000)
#define S_H_PW        1
#define S_H_BP        88
#define S_H_VD        800
#define S_H_FP        211

#define S_V_PW        3
#define S_V_BP        10
#define S_V_VD        480
#define S_V_FP        7

#define S_H_ST        244
#define S_V_ST        11
```

计算好相应配置参数后，在 driver/video/display/screen 下加入屏驱动，仿照 driver/video/display/screen/lcd_hdmi800x480.c

其中 SCALE_RATE(148500000, 330000000) 这个的配置需注意 driver/video/display/screen/screen.h 中是否有相应的频率，如果没有，请按双屏配置工具中生成的 m, n, no 填写上相应的频率。

频率列表如下：

```
enum{
    SCALE_PLL(148500000,    66000000,    16, 9, 4),
    SCALE_PLL(148500000,    54000000,    16, 11, 4),
    SCALE_PLL(148500000,    33000000,    16, 9, 8),
    SCALE_PLL(148500000,    30375000,    18, 11, 8),
    SCALE_PLL(148500000,    29700000,    16, 10, 8),
    SCALE_PLL(148500000,    25312500,    15, 11, 8),

    SCALE_PLL(74250000,    66000000,    32, 9, 4),
    SCALE_PLL(74250000,    54000000,    32, 11, 4),
    SCALE_PLL(74250000,    33000000,    32, 9, 8),
    SCALE_PLL(74250000,    30375000,    36, 11, 8),
    SCALE_PLL(74250000,    25312500,    30, 11, 8),

    SCALE_PLL(27000000,    31500000,    28, 3, 8),
    SCALE_PLL(27000000,    30000000,    80, 9, 8),
};
```

双屏配置做了生成工具，具体详见：“rk610 双屏配置工具”

从配置工具中获得类似的配置：

```
*****config pll*****/
xin=74.250000
/      hdmi mode 1280*720*60      /
*****best lcd config *****/
bestline=0.000000
vtotal=500
best_htotal=1100
HZ=60.000000
clk=33.000000 m=32 n=9 no=8
fvst + VBP +VPW =20
fhst=0
```

bestline =0.000000

总有效行时间偏差，这个为 0 时最好

vtotal=500

令 $S_V_PW + S_V_BP + S_V_VD + S_V_FP = vtotal = 500$
 $best_htotal = 1100$
 令 $S_H_PW + S_H_BP + S_H_VD + S_H_FP = best_htotal = 1100$
 $HZ = 60.000000$
 场频, 与 hdmi 场频接近最好
 $xin = 74.2500000$
 $clk = 33.000000$ $m = 32$ $n = 9$ $no = 8$
 查看 driver/video/display/screen/screen.h 是否有该配置, 如没有加上 SCALE_PLL (74250000, 33000000, 32, 9, 8), 在屏驱动中用 SCALE_RATE (74250000, 33000000);
 scale clk 配置
 $fvst + VBP + VPW = 22$
 fvst 即屏驱动中的 S_V_ST, 如果按上面的数字配出的显示有问题, 适当增加数值, 一般显示就会正常。
 $fhst = 244$
 fhst 即屏驱动中的 S_H_ST, 该数值相当于 fvst 的小数部分, 如需细调, 可以调整该参数, 一般效果不明显。
 config success
 说明生成的配置比较不错
 waring
 说明生成的配置有一定偏差, 尽量让 HZ 与 bestline 接近对应的值, 如果偏差不大, 也是有可能能正常显示。

注:

RGB 屏定义:

```
#define OUT_TYPE    SCREEN_RGB
```

LVDS 屏 定义:

```
#define OUT_TYPE    SCREEN_LVDS    //LVDS 屏要配成 SCREEN_LVDS
```

```
#define OUT_FORMAT    LVDS_8BIT_2    //LVDS 的接线方式
```

Scaler 时钟反向:

```
#define DCLK_POL    0    //bypass 显示时, 画面抖动, 可以尝试将这个取反
```

```
#define S_DCLK_POL    0    //当双屏显示画面抖动时, 可以尝试将这个取反
```

/*		LVDS_8BIT_1	LVDS_8BIT_2	LVDS_8BIT_3	LVDS_6BIT
Y 0	TX0	R0	R2	R2	R0
	TX1	R1	R3	R3	R1
	TX2	R2	R4	R4	R2
	TX3	R3	R5	R5	R3
	TX4	R4	R6	R6	R4
	TX6	R5	R7	R7	R5
	TX7	G0	G2	G2	G0
Y 1	TX8	G1	G3	G3	G1
	TX9	G2	G4	G4	G2
	TX12	G3	G5	G5	G3
	TX13	G4	G6	G6	G4
	TX14	G5	G7	G7	G5
	TX15	B0	B2	B2	B0
	TX18	B1	B3	B3	B1
Y 2	TX19	B2	B4	B4	B2
	TX20	B3	B5	B5	B3
	TX21	B4	B6	B6	B4
	TX22	B5	B7	B7	B5
	TX24	HSYNC	HSYNC	HSYNC	HSYNC
	TX25	VSYNC	VSYNC	VSYNC	VSYNC
	TX26	ENABLE	ENABLE	ENABLE	ENABLE
Y 3	TX27	R6	R0	GND	GND
	TX5	R7	R1	GND	GND
	TX10	G6	G0	GND	GND
	TX11	G7	G1	GND	GND
	TX16	B6	B0	GND	GND

TX17	B7	B1	GND	GND
TX23	RSVD	RSVD	RSVD	RSVD

5. Lcd 双屏调试

设置好参数后，在实际中可能会用偏差，如果出现问题，可以通过串口来微调参数校准。

方法：

一、打开 Debug 开关：

在 include/linux/mfd/rk610-core.h 中：#define RK610_DEBUG

在调试的 sys 节点：

busybox find /sys -name reg_ctl

./sys/devices/platform/rk29_i2c.1/i2c-1/1--40/reg_ctl //codec 部分控制 lcd lvds 控制

./sys/devices/platform/rk29_i2c.1/i2c-1/1--46/reg_ctl //hdmi 控制寄存器

```
shell@android:/ # busybox find -name reg_ctl
./sys/devices/platform/rk29_i2c.2/i2c-2/2-0040/reg_ctl
./sys/devices/platform/rk29_i2c.2/i2c-2/2-0046/reg_ctl
shell@android:/ # cat /sys/devices/platform/rk29_i2c.2/i2c-2/2-0040/reg_ctl
>>>rk610_ctl 0: 22 51 51 b8 e 1 42 64 5 24 ff 2 10 0 0 0
>>>rk610_ctl 10: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
>>>rk610_ctl 20: 0 0 0 0 0 0 0 0 0 0 0 7 7 0 0
>>>rk610_ctl 30: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
>>>rk610_ctl 40: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
>>>rk610_ctl 50: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
>>>rk610_ctl 60: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
>>>rk610_ctl 70: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
>>>rk610_ctl 80: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
>>>rk610_ctl 90: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
>>>rk610_ctl a0: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
>>>rk610_ctl b0: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
>>>rk610_ctl c0: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
>>>rk610_ctl d0: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
>>>rk610_ctl e0: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
shell@android:/ # cat /sys/devices/platform/rk29_i2c.2/i2c-2/2-0046/reg_ctl
>>>rk610_hdmi 0: 5 1 34 81 8 0 22 0 0 0 0 0 0 0 0
>>>rk610_hdmi 10: 0 0 0 0 0 0 0 0 0 4 0 6 4d 0 0 2 ca
>>>rk610_hdmi 20: 4 0 11 df 10 c0 0 54 4 0 0 0 7 6c 2 ee
>>>rk610_hdmi 30: 0 0 0 0 0 0 1 40 40 3c 0 f 0 0 0 0
>>>rk610_hdmi 40: 18 80 0 0 0 0 0 0 0 0 0 63 40 0 0 2
>>>rk610_hdmi 50: a 0 12 8 0 6 0 20 0 7f 0 0 0 0 0 0
>>>rk610_hdmi 60: 0 0 12 26 2c 0 0 0 0 0 0 10 0 0 0 0
>>>rk610_hdmi 70: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
>>>rk610_hdmi 80: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
>>>rk610_hdmi 90: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 8
>>>rk610_hdmi a0: 84 1 a 0 11 9 0 0 1 0 0 0 0 0 0 0
>>>rk610_hdmi b0: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
>>>rk610_hdmi c0: c0 0 78 0 0 0 0 0 0 50 70 0 1 0 1 0
>>>rk610_hdmi d0: 0 0 0 3 3 9 3 0 ff ff 0 d0 20 0 0
>>>rk610_hdmi e0: 0 8c 4 2 3 1c 2c 4 10 0 0 0 0 0 0
>>>rk610_hdmi f0: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
shell@android:/ #
```

二、调试 scaler 参数（lcd 上显示有横线等时序出错的现象）

cd ./sys/devices/platform/rk29_i2c.1/i2c-1/1--40/

cat reg_ctl

屏驱动中的 S_V_ST = {reg[0x14],reg[13]};

echo 13xx >reg_ctl (向 0x13 寄存器写入值 xx);

echo 14xx >reg_ctl (向 0x14 寄存器写入值 xx);

三、调试 LVDS 屏（LVDS 单屏和双屏颜色显示均不正常时）

LVDS 四种接线：

0x09 [1:0] 00: 8 bit format-1

01: 8 bit format-2

10: 8 bit format-3

11: 6 bit mode

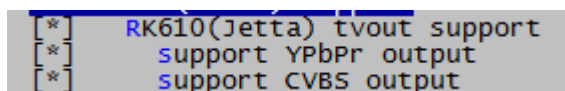
echo 09xx >reg_ctl (向 0x09 寄存器写入值 xx);

6. RK610 TVE

RK610 的 TVE 模块用于输出传统电视的 YPbPr 信号和 AV 信号。

3. 内核选项

RK610 TVE 模块支持 YPbPr 和 CVBS 输出，在内核中有三个相关选项需要配置，如下图示：



- RK610 (Jetta) tvout support: 支持 TVE 功能，使能该项后会开启以下两个选择
- Support YPbPr output: 支持 TVE 输出 YPbPr 信号，支持的格式有：
 - 1920x1080P@60Hz
 - 1920x1080P@50Hz
 - 1920x1080I@60Hz
 - 1280x720P@60Hz
 - 1280x720P@50Hz
 - 720Px576P@50Hz
 - 720Px480P@60Hz
- Support YPbPr output: 支持 TVE 输出 CVBS 信号，支持的格式有：
 - PAL
 - NTSC

4. 应用程序操作

开启 TVE 模块的 YPbPr 和 CVBS 输出后，会在 Linux 的 sysfs 系统中注册一个显示设备在 /sys/class/display 目录下会对对应生成 YPbPr 和 CVBS 目录。在每个目录中存在以下文件：

- type: 表征该显示接口的类型，只读。
- connect: 表征该显示接口是否与显示设备连接，只读。
- enable: 表征该显示接口是否使能，可读写。
- modes: 表征该显示接口支持的显示格式，只读。
- mode: 表征该显示接口当前的显示格式，可读写。

TVE 的输出操作通常有以下场景：

A. 使能（禁用）YPbPr 输出或 CVBS 输出。以 YPbPr 为例，操作步骤为：

- a) 使能：向 /sys/class/display/YPbPr/enable 文件写入字符串 “1”。此时，YPbPr 默认输出 720P@60Hz 格式。
- b) 禁用：向 /sys/class/display/YPbPr/enable 文件写入字符串 “0”。

注意：由于 CVBS 与 YPbPr 均由 RK610 输出，开启 YPbPr 输出前，应确认 CVBS 输出为禁用状态，反之亦然。

B. 设置 YPbPr (CVBS) 输出的显示格式。以 YPbPr 为例，操作步骤如下：

- a) 读取 /sys/class/display/YPbPr/modes 文件，获取 YPbPr 支持的显示格式。各视频格式分别对应一个字符串，对应关系如下表示：

显示接口	显示格式	字符串
------	------	-----

YPbPr	1080P@60Hz	1920x1080p-60
	1080P@50Hz	1920x1080p-50
	1080I@60Hz	1920x1080i-60
	720P@60Hz	1280x720p-60
	720P@50Hz	1280x720p-50
	576P@50Hz	720x576p-50
	480P@60Hz	720x480p-60
CVBS	PAL	720x576i-50
	NTSC	720x480i-60

- b) 从 modes 文件列出的视频格式中选取视频格式，比如 1080P@60Hz，将表征该格式的字符串 “1920x1080p-60” 写入 /sys/class/display/YPbPr/mode 文件。各视频格式与对应的字符串
- c) 向 /sys/class/display/YPbPr/enable 文件写入字符串 “1”。