密级状态：绝密（ ） 秘密（ ） 内部（ √ ） 公开（ ）

固件工厂内核设置

（技术部，MID 组）

文件状态： [ ] 正在修改 [√] 正式发布	当前版本：	V1.0
	作　　者：	杨文杰
	完成日期：	2013-07-23
	审　　核：	
	完成日期：	

福州瑞芯微电子有限公司

Fuzhou Rockchips Semiconductor Co . , Ltd

版本历史

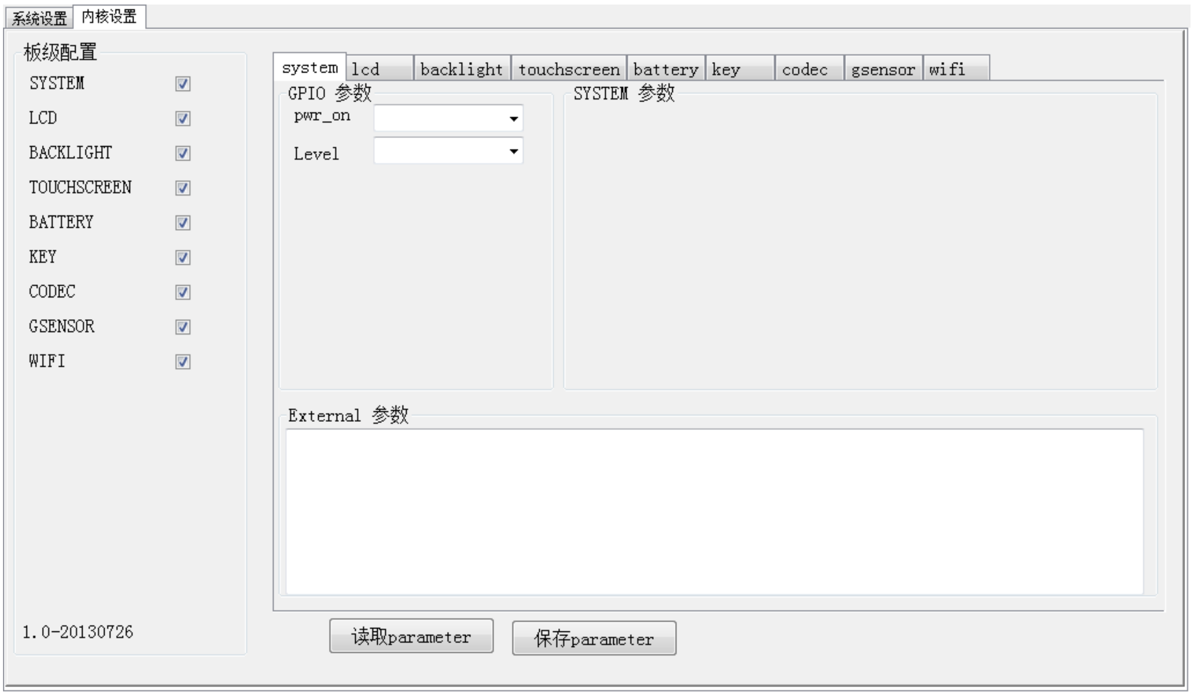
版本号	作者	修改日期	修改说明	备注
V1.0	杨文杰	20130723	建立文档	
V1.0	杨文杰	20130830	增加 rk3026 支持	

目 录

1 软件说明

内核设置目前只适用 RK3168，RK3188，RK3026，git 更新到 20130723 及其以后发布的版本

2 使用说明



SYSTEM ☑
LCD ☑
BACKLIGHT ☑
TOUCHSCREEN ☑
BATTERY ☑
KEY ☑
CODEC ☑
GSENSOR ☑
WIFI ☑

1.0-20130726

在固件工厂解压完固件之后，点选 读取parameter ，加载工具目录 Temp\Android\parameter,加载成功后选择相应的标签页进行修改，如果只想修改某一项功能，可以勾选板级配置。

在工具目录下 config\ 目录下有参考的 parameter_rk3188_616， parameter_rk3168_616 and parameter_rk3026_86v

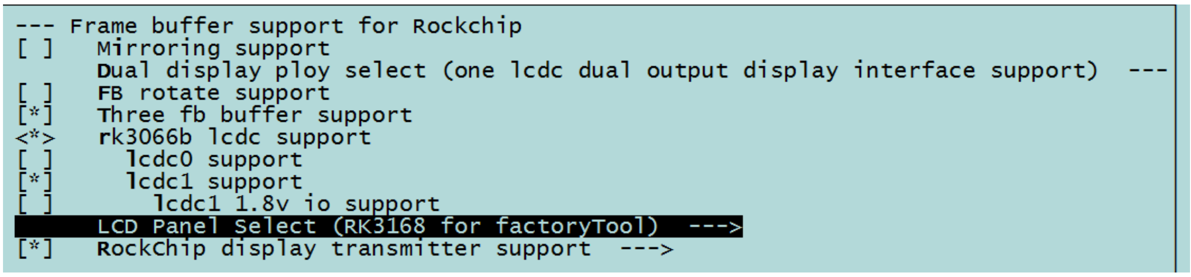
编辑完成之后可以覆盖 Temp\Android\parameter

只配置其中的某些驱动，可以勾选左边的复选框

配置中不需要一些配置项，可以空置或者直接出入空白

2.1 LCD 配置

Menuconfig 中配置



```
--- Frame buffer support for Rockchip
[ ]   Mirroring support
      Dual display ploy select (one lcdc dual output display interface support)  ---
[ ]   FB rotate support
[*]   Three fb buffer support
<*>   rk3066b lcdc support
[ ]     lcdc0 support
[*]     lcdc1 support
[ ]       lcdc1 1.8v io support
      LCD Panel Select (RK3168 for factoryTool)  --->
[*]   RockChip display transmitter support  --->
```

把以前调试过的 lcd 的相关参数填写即可替换

```
#define SCREEN_TYPE          SCREEN_RGB
#define LVDS_FORMAT          LVDS_8BIT_2
#define OUT_FACE         OUT_P888
#define DCLK         50000000
#define LCDC_ACLK            500000000//312000000            //29 lcdc axi DMA  频率
/* Timing */
#define H_PW         30
#define H_BP         10
#define H_VD         1024
#define H_FP         210
#define V_PW         13
#define V_BP         10
#define V_VD         600
#define V_FP         22
#define LCD_WIDTH            154
#define LCD_HEIGHT          85
#define DCLK_POL          1
#define SWAP_RB           0
```

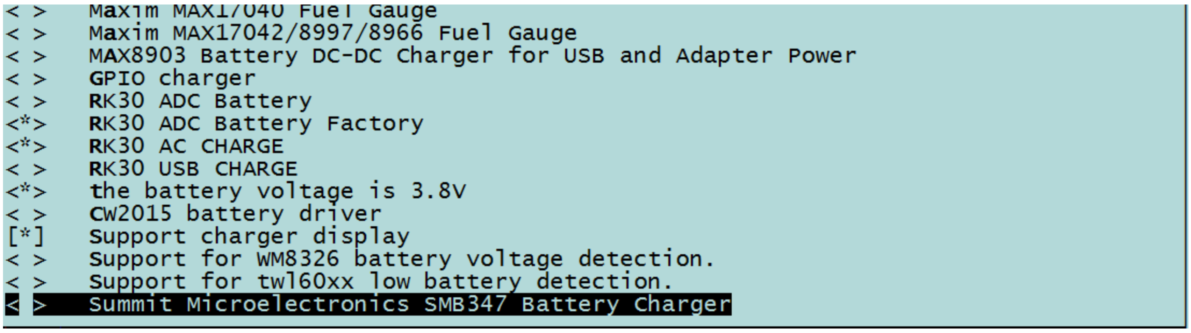
2.2 BackLight 配置

把 board 文件中 backlight 的参数写入即可，比如 board-rk3168-86v.c 中

```
static struct rk29_bl_info rk29_bl_info = {

        .min_brightness = 33,

        .max_brightness=255,

        .brightness_mode =1,

        .pre_div = 20 * 1000,    // pwm output clk: 20k;

        .pwm_id = PWM_ID,

        .bl_ref = PWM_EFFECT_VALUE,

        .io_init = rk29_backlight_io_init,

        .io_deinit = rk29_backlight_io_deinit,

        .pwm_suspend = rk29_backlight_pwm_suspend,

        .pwm_resume = rk29_backlight_pwm_resume,

};
```

2.3 battery 配置

Menuconfig 中配置



```
< >   Maxim MAX17040 Fuel Gauge
< >   Maxim MAX17042/8997/8966 Fuel Gauge
< >   MAX8903 Battery DC-DC Charger for USB and Adapter Power
< >   GPIO charger
< >   RK30 ADC Battery
<*>   RK30 ADC Battery Factory
<*>   RK30 AC CHARGE
< >   RK30 USB CHARGE
<*>   the battery voltage is 3.8V
< >   CW2015 battery driver
[*]   Support charger display
< >   Support for WM8326 battery voltage detection.
< >   Support for twl60xx low battery detection.
< >   Summit Microelectronics SMB347 Battery Charger
```

把 board 文件中 battery 的参数写入即可，比如 board-rk3168-86v.c 中

```
static struct rk30_adc_battery_platform_data rk30_adc_battery_platdata = {

            .dc_det_pin          = RK30_PIN0_PB2,

            .batt_low_pin        = INVALID_GPIO,
            .charge_set_pin    = INVALID_GPIO,
            .charge_ok_pin      = RK30_PIN0_PA6,
            .usb_det_pin = INVALID_GPIO,
            .dc_det_level        = GPIO_LOW,
            .charge_ok_level = GPIO_HIGH,

           .reference_voltage = 1800, // the rK2928 is 3300;RK3066 and rk29 are 2500;rk3066B is 1800;
           .pull_up_res = 200,          //divider resistance ,    pull-up resistor
           .pull_down_res = 120, //divider resistance , pull-down resistor

           .is_reboot_charging = 1,
           .save_capacity      = 1 ,
           .low_voltage_protection = 3600,        };
```

2.4 key 配置

把 board 文件中 key 的参数写入即可，比如 board-rk3168-86v.c 中

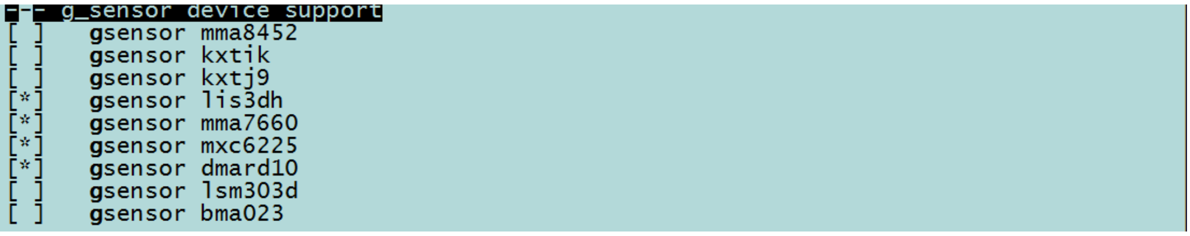
```
static struct rk29_keys_button key_button[] = {
    {
        .desc       = "play",
        .code       = KEY_POWER,
        .gpio       = RK30_PIN0_PA4,
        .active_low = PRESS_LEV_LOW,
        .wakeup   = 1,
    },
    {
                .desc      = "vol-",
                .code       = KEY_VOLUMEDOWN,
                .adc_value          = 150,
                .gpio      = INVALID_GPIO,
                .active_low = PRESS_LEV_LOW,
        },
        {
                .desc      = "vol+",
                .code       = KEY_VOLUMEUP,
                .adc_value          = 1,
                .gpio = INVALID_GPIO,
                .active_low = PRESS_LEV_LOW,
        },
};
struct rk29_keys_platform_data rk29_keys_pdata = {
    .buttons    = key_button,
    .nbuttons = ARRAY_SIZE(key_button),
    .chn = 1,    //chn: 0-7, if do not use ADC,set 'chn' -1
};
```

1 如果机子上没有缺少相应的按键， 请填写 0x80000000,

2 如果知道按键的 adc 值，请换算成 16 进制写入，比如 vol-按键 adc 的值是 150，十六进制是 0x96，填写的值为 0x80000096

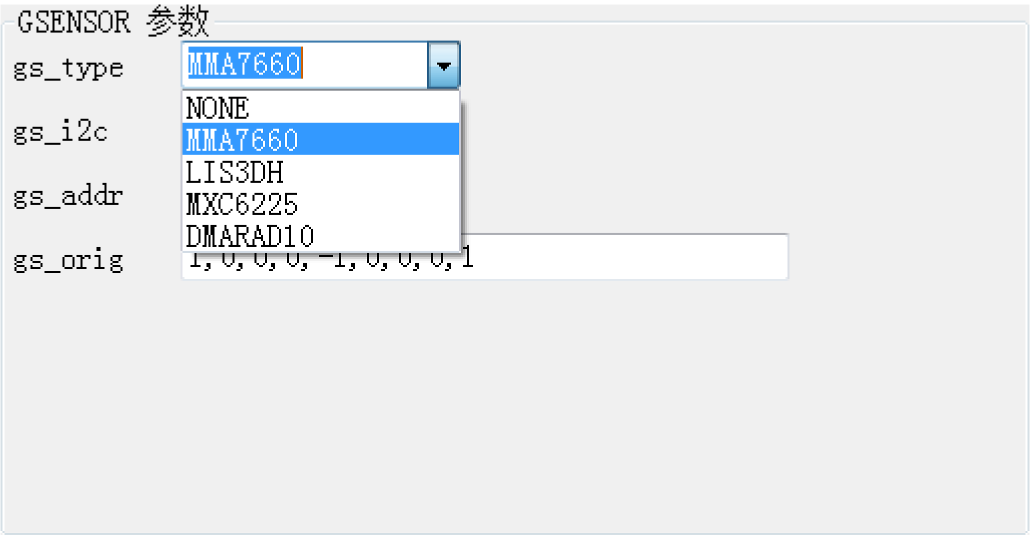
2.5 Gsensor 配置

Menuconfig 中配置



```
--- g_sensor device support
[ ]    gsensor mma8452
[ ]    gsensor kxtik
[ ]    gsensor kxtj9
[*]    gsensor lis3dh
[*]    gsensor mma7660
[*]    gsensor mxc6225
[*]    gsensor dmard10
[ ]    gsensor lsm303d
[ ]    gsensor bma023
```

目前支持四种 gsensor 的配置



GSENSOR 参数

gs_type MMA7660

gs_i2c

gs_addr

gs_orig 1, 0, 0, 0, -1, 0, 0, 0, 1

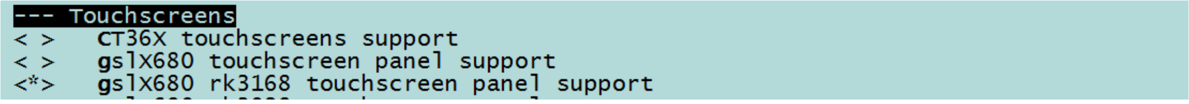
Gs_orig 代表 Gsensor 的方向数组

```
static struct sensor_platform_data mma8452_info = {
        .type = SENSOR_TYPE_ACCEL,
        .irq_enable = 1,
        .poll_delay_ms = 30,
                .init_platform_hw = mma8452_init_platform_hw,
                .orientation = {-1, 0, 0, 0, 0, 1, 0, 1, 0},
};
```

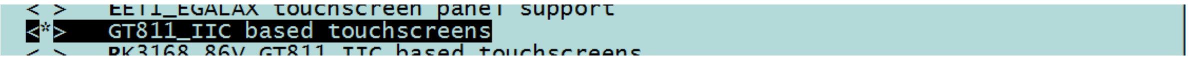
新变量和新驱动的添加参考第四节和第五节

2.6 TP 配置

Menuconfig 中配置

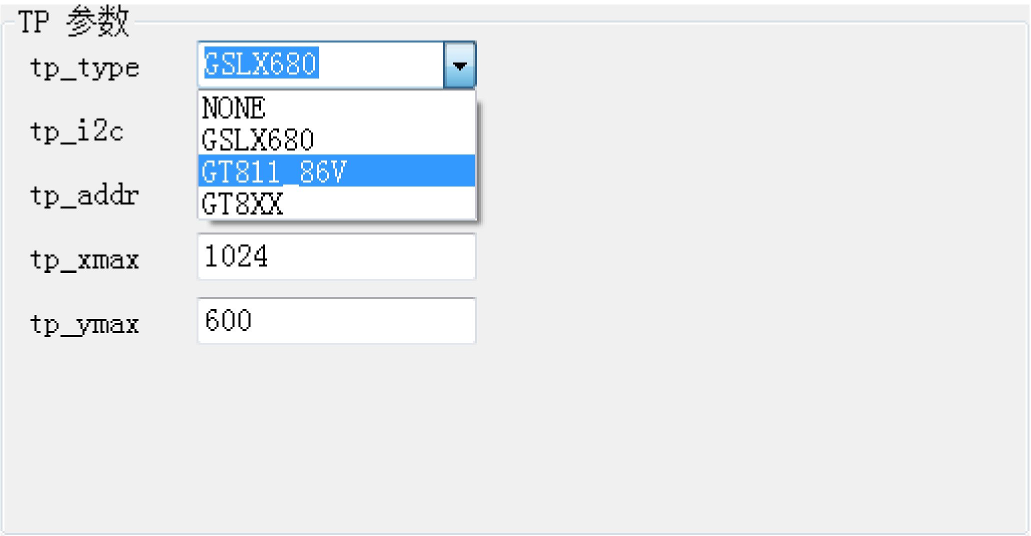


```
--- Touchscreens
< >    CT36X touchscreens support
< >    gslX680 touchscreen panel support
<*>    gslX680 rk3168 touchscreen panel support
```



```
< >   EETI_EGALAX touchscreen panel support
<*>   GT811_IIC based touchscreens
< >   RK3168_86V GT811_IIC based touchscreens
```

目前支持两中 TP，（GSLX680 和 GT811_86V） 或者（GSLX680 和 GT8XX）



TP 参数

tp_type: GSLX680

tp_i2c

tp_addr

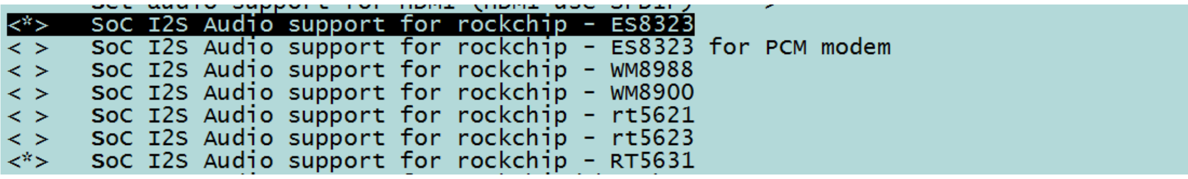
tp_xmax: 1024

tp_ymax: 600

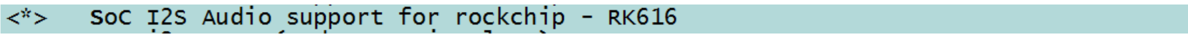
新变量和新驱动的添加参考第四节和第五节

2.7 codec 配置

Menuconfig 中配置

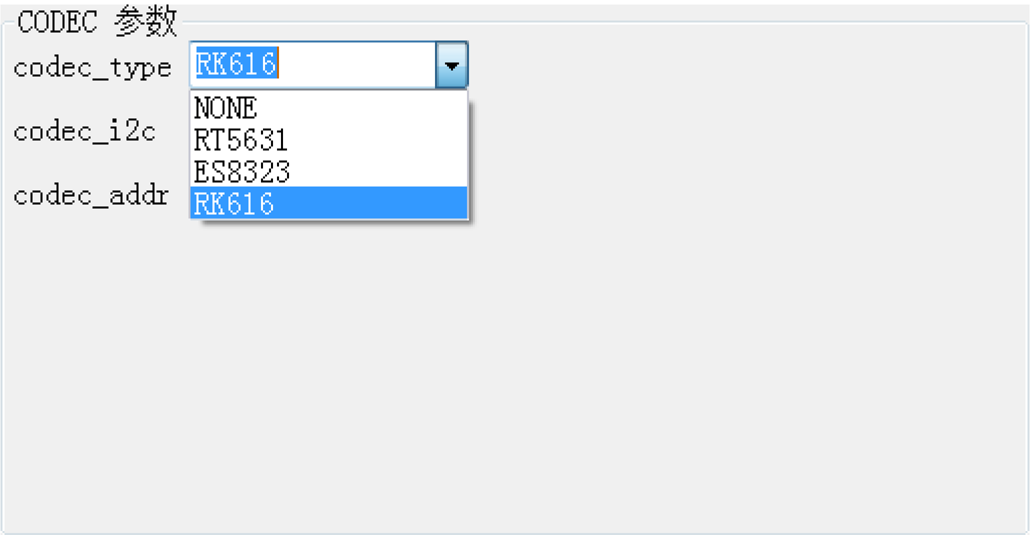


```
      Set audio support for HDMI (HDMI use SPDIF)  --->
<*>   SoC I2S Audio support for rockchip - ES8323
< >   SoC I2S Audio support for rockchip - ES8323 for PCM modem
< >   SoC I2S Audio support for rockchip - WM8988
< >   SoC I2S Audio support for rockchip - WM8900
< >   SoC I2S Audio support for rockchip - rt5621
< >   SoC I2S Audio support for rockchip - rt5623
<*>   SoC I2S Audio support for rockchip - RT5631
```



```
<*>   SoC I2S Audio support for rockchip - RK616
```

目前支持三种 codec 的配置



CODEC 参数

codec_type RK616

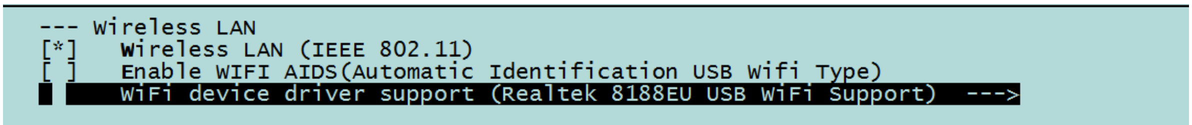
codec_i2c

codec_addr

新变量和新驱动的添加参考第四节和第五节

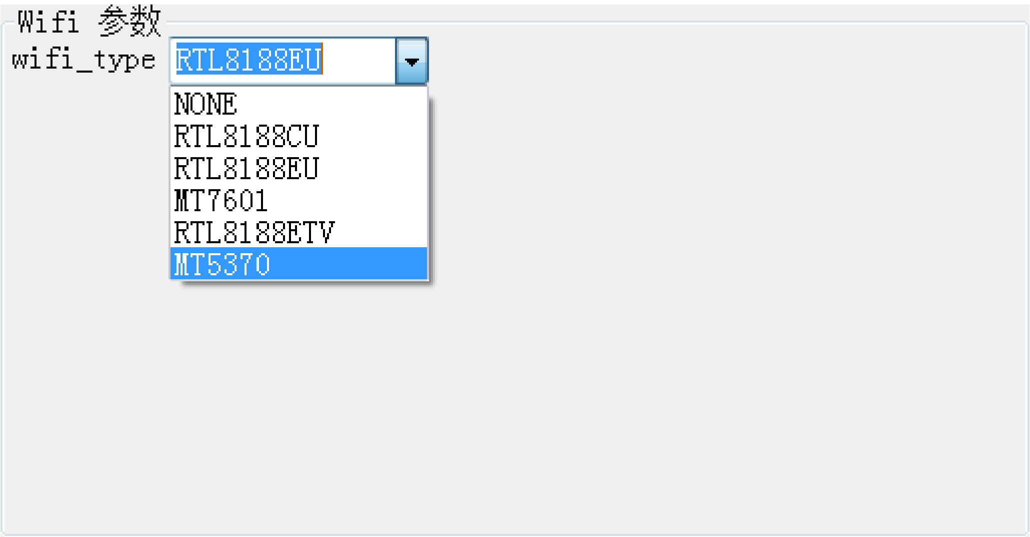
2.8 wifi 配置

Menuconfig 中配置



```
--- Wireless LAN
[*]   Wireless LAN (IEEE 802.11)
[ ]   Enable WIFI AIDS(Automatic Identification USB wifi Type)
      WiFi device driver support (Realtek 8188EU USB WiFi Support)  --->
```

目前支持 5 种 USBwifi



Wifi 参数

wifi_type RTL8188EU

新驱动的添加只需要在 kernel\drivers\net\wireless\wifi_sys\rkwifi_sys_iface.c 中，比如添加 MT5370

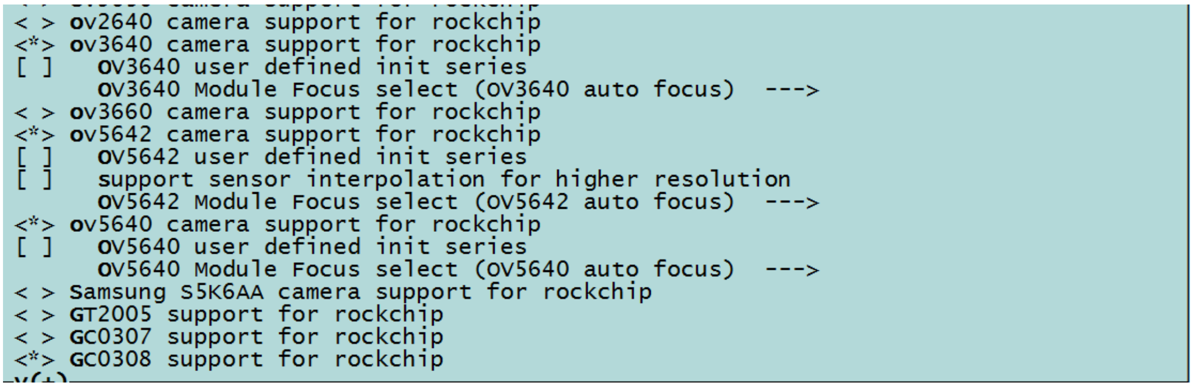
#ifdef CONFIG_MACH_RK_FAC

if(wifi_type==WIFI_TYPE_RTL8188CU) {

```
        count = sprintf(_buf, "%s", "RTL8188CU");

        printk("Current WiFi chip is RTL8188CU.\n");

        return count;

    } else if(wifi_type==WIFI_TYPE_RTL8188EU) {

        count = sprintf(_buf, "%s", "RTL8188EU");

        printk("Current WiFi chip is RTL8188EU.\n");

        return count;

    } else if(wifi_type==WIFI_TYPE_MT7601) {

        count = sprintf(_buf, "%s", "MT7601");

        printk("Current WiFi chip is MT7601.\n");

        return count;

    } else if(wifi_type==WIFI_TYPE_RTL8188ETV) {

        count = sprintf(_buf, "%s", "RTL8188ETV");

        printk("Current WiFi chip is RTL8188ETV.\n");

        return count;

    } else if(wifi_type==WIFI_TYPE_MT5370) {

        count = sprintf(_buf, "%s", "MT5370");

        printk("Current WiFi chip is MT5370.\n");

        return count;

    } else {

        //printk("NOT surpport type %d\n",wifi_type);

    }

#endif
```

2.9 camera 配置

Menuconfig 中配置



```
< > ov2640 camera support for rockchip
<*> ov3640 camera support for rockchip
[ ]   OV3640 user defined init series
      OV3640 Module Focus select (OV3640 auto focus)  --->
< > ov3660 camera support for rockchip
<*> ov5642 camera support for rockchip
[ ]   OV5642 user defined init series
[ ]   support sensor interpolation for higher resolution
      OV5642 Module Focus select (OV5642 auto focus)  --->
<*> ov5640 camera support for rockchip
[ ]   OV5640 user defined init series
      OV5640 Module Focus select (OV5640 auto focus)  --->
< > Samsung S5K6AA camera support for rockchip
< > GT2005 support for rockchip
< > GC0307 support for rockchip
<*> GC0308 support for rockchip
```

可支持内核配置中的任意四款 camera

可参考文档 Camera_for_RockChipSDK 参考说明_v4.1.pdf

3 内核配置

配置文件可以参考

rk3168_86V_fac_616_defconfig

rk3188_rk618_fac_defconfig

4 普通配置项的添加

如果你自己有其他的参数需要添加，参考如下，以 tp_firmVer 为例

4.1 工具中添加

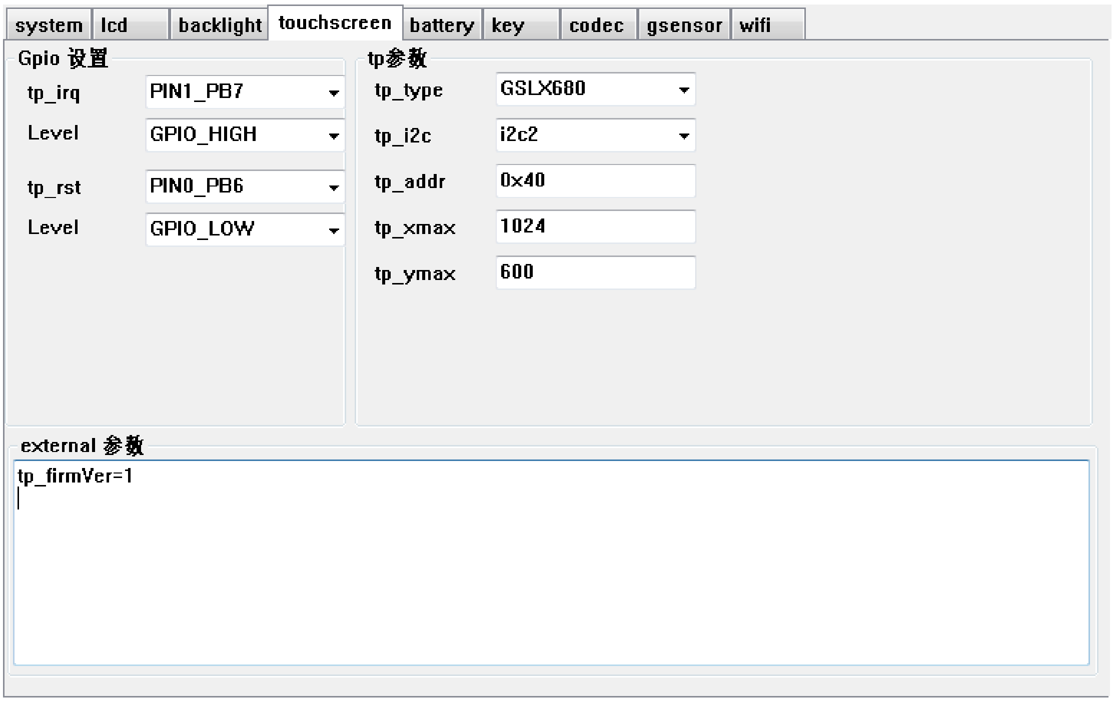
在工具 config\KernelConfig.ini 中添加

backlight={bl_en(io),bl_pwmid,bl_mode,bl_div,bl_ref,bl_min,bl_max}

touchscreen={tp_type,tp_irq(io),tp_rst(io),tp_i2c,tp_addr,tp_xmax,tp_ymax,tp_firmVer} battery={dc_det(io),chg_ok(io),ref_vol,up_res,down_res,root_chg,save_cap,low_vol,bat_num,bat_charge,bat_discharge}

在原始的 config\parameter_616 中添加

board.tp_xmax=1024 board.tp_ymax=600 board.tp_firmVer=1 board.gs_type=1 board.gs_i2c=0



Gpio 设置

tp_irq PIN1_PB7

Level GPIO_HIGH

tp_rst PIN0_PB6

Level GPIO_LOW

tp参数

tp_type GSLX680

tp_i2c i2c2

tp_addr 0x40

tp_xmax 1024

tp_ymax 600

external 参数

tp_firmVer=1

4.2 内核中添加

在 arch/arm/mach-rk30/board-rk3168-fac-config.c 中添加，其中默认值可以自己填写

static int tp_ymax = DEF_Y_MAX;

module_param(tp_ymax, int, 0644);

static int tp_firmVer= DEF_FIRMVER;

module_param(tp_firmVer, int, 0644);

在 arch/arm/plat-rk/include/plat/board.h

struct tp_platform_data {

```
    int model;

    int x_max;

    int y_max;

    int reset_pin;

    int irq_pin ;

    int firmVer;

    int (*get_pendown_state)(void);

    int (*init_platform_hw)(void);

    int (*platform_sleep)(void);

    int (*platform_wakeup)(void);

    void (*exit_platform_hw)(void);

};
```

arch/arm/mach-rk30/board-rk3168-fac.c 中添加

```
#if defined (CONFIG_TOUCHSCREEN_GSLX680_RK3168)

    if(tp_type == TP_TYPE_GSLX680){

        gslx680_data.irq_pin = irq_port.gpio;

        gslx680_info.addr = tp_addr;

        gslx680_data.reset_pin= rst_port.gpio;

        gslx680_data.x_max=tp_xmax;

        gslx680_data.y_max=tp_ymax;

        gslx680_data.firmVer=tp_firmVer;

        i2c_register_board_info(tp_i2c, &gslx680_info, 1);

    }

#endif
```

然后在 GSLX680 驱动中引入 firmVer 即可实现对 tp 固件版本的控制

5 GPIO 配置项的添加

如果你自己有其他的 GPIO 口需要添加控制，参考如下，以 example(io)为例

5.1 工具中添加

在工具 config\KernelConfig.ini 中添加

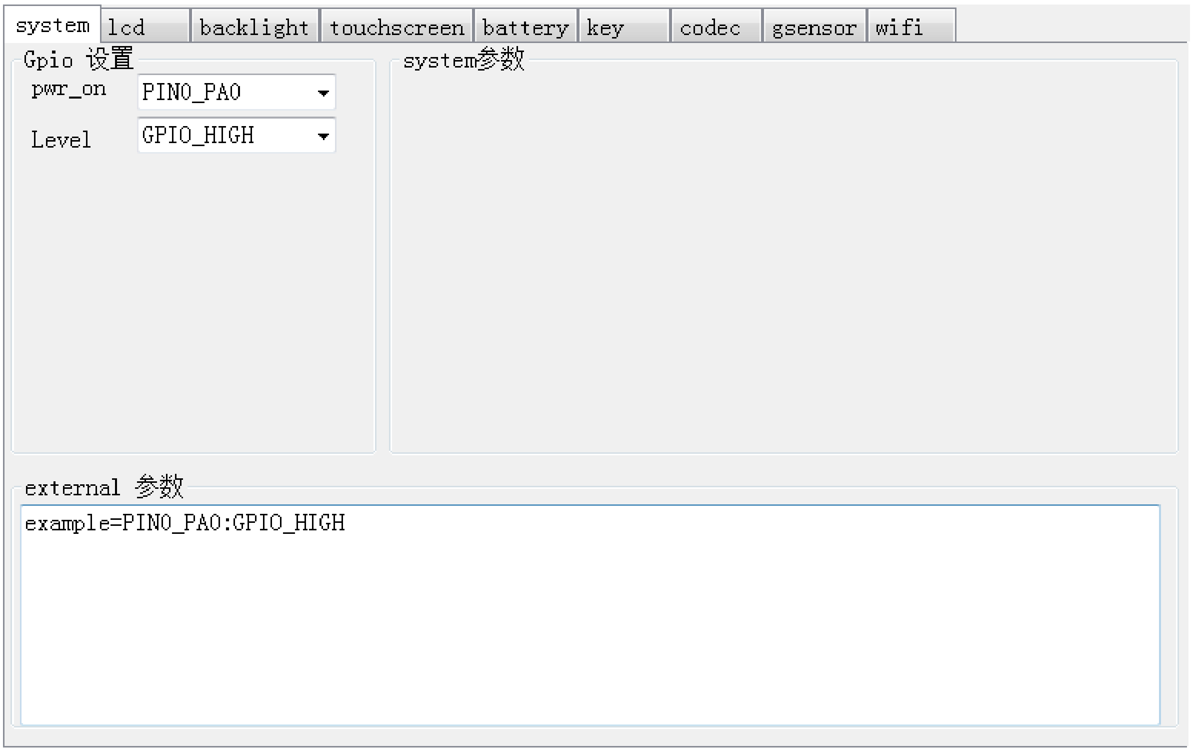
backlight={bl_en(io),bl_pwmid,bl_mode,bl_div,bl_ref,bl_min,bl_max}

lcd={lcd_std(io),lcd_en(io),lcd_cs(io),lcd_param}

system={pwr_on(io),example(io)}

在原始的 config\parameter_616 中添加

board.example=0x000000a0,board.pwr_on=0x000000a0,board.lcd_param=1,0,1,50000000,500000000,3



pwr_on: PIN0_PA0

Level: GPIO_HIGH

```
example=PIN0_PA0:GPIO_HIGH
```

如果需要换 GPIO 口和电平，可以修改为：example=PIN0_PA3:GPIO_LOW

5.2 内核中添加

在 arch/arm/mach-rk30/include/mach/config.h

```
enum {
        DEF_PWR_ON = 0x000000a0,
};
enum {
        DEF_EXA_ON = 0x000000a0,
};
```

在 arch/arm/mach-rk30/board-rk3168-fac-config.c 中添加，其中默认值可以自己填写

```
static int pwr_on = DEF_PWR_ON;
module_param(pwr_on, int, 0644);
static int example = DEF_EXA_ON;
module_param(example, int, 0644);
static inline int rk_power_on(void)
{
        int ret;
        ret=port_output_init(pwr_on, 1, "pwr_on");
        if(ret<0)
                CONFIG_ERR(pwr_on, "pwr_on");
        port_output_on(pwr_on);
        ret=port_output_init(example, 1, "example");
        if(ret<0)
                CONFIG_ERR(example, "example");
                port_output_on(example);
}
```

这样就可以实现对 GPIO 的控制,也可以参考 bl_en(io)的相关代码做参考

6 新驱动的添加

新驱动的添加，以 Gsensor D10 为例：

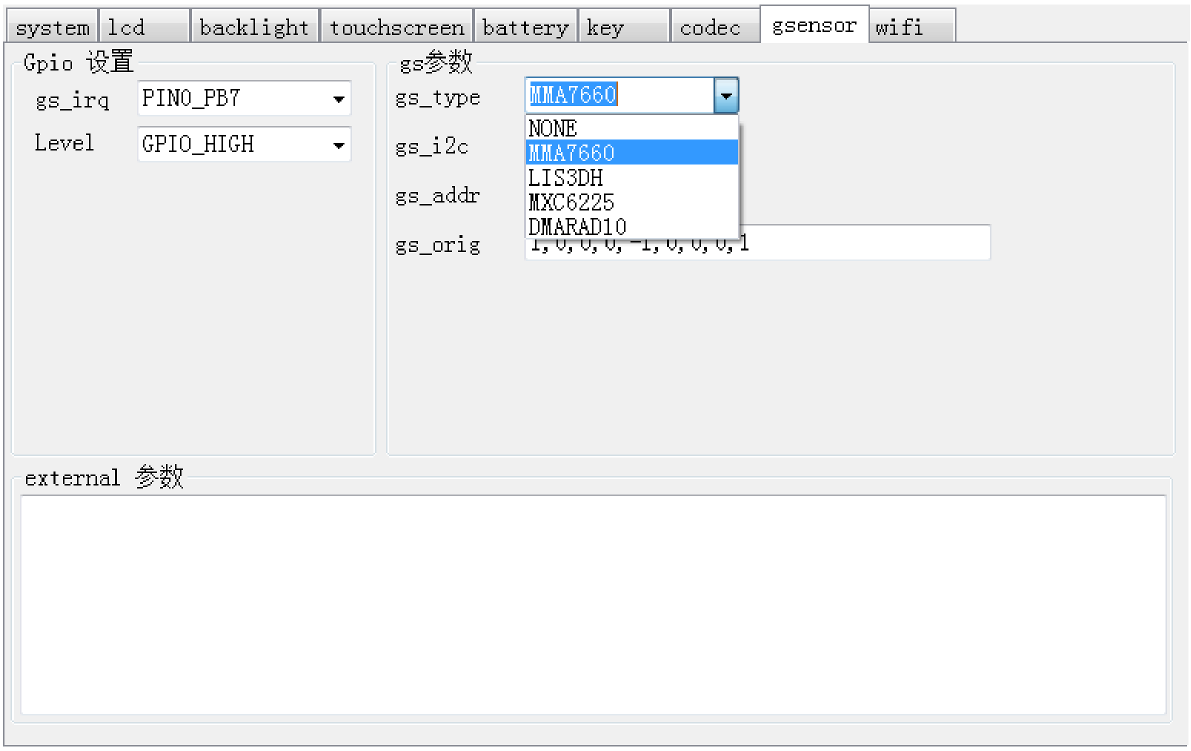
6.1 工具中添加

在工具 config\KernelConfig.ini 中添加

enum codec_type {NONE,RT5631,ES8323,RK616}

enum gs_type {NONE,MMA7660,LIS3DH,MXC6225,DMARAD10}

enum wifi_type {NONE,RTL8188CU,RTL8188EU,MT7601,RTL8188ETV,MT5370}



Gpio 设置

gs_irq PIN0_PB7

Level GPIO_HIGH

gs参数

gs_type MMA7660

gs_i2c

gs_addr

gs_orig 1, 0, 0, 0, -1, 0, 0, 0, 1

external 参数

6.2 内核中添加

arch/arm/mach-rk30/include/mach/config.h 中添加

```
enum {

    GS_TYPE_NONE = 0,

    GS_TYPE_MMA7660,
    GS_TYPE_LIS3DH,

    GS_TYPE_MXC6225,

    GS_TYPE_DMARAD10,

    GS_TYPE_MAX,

};
```

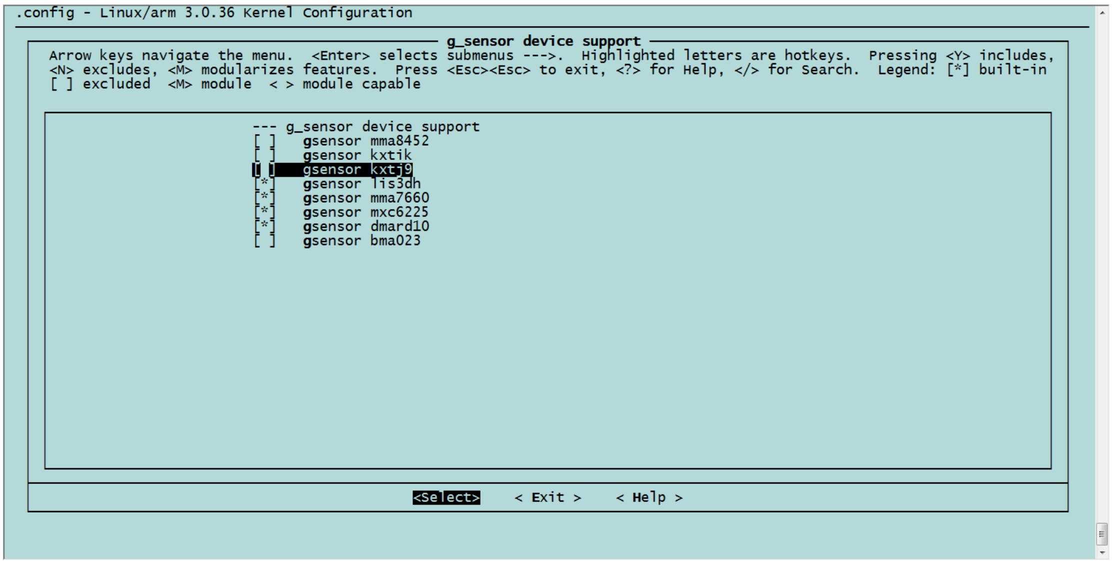
arch/arm/mach-rk30/board-rk3168-fac.c 中添加

```
#if defined (CONFIG_GS_MXC6225)

static int mxc6225_init_platform_hw(void)

{

//          rk30_mux_api_set(GPIO1B1_SPI_TXD_UART1_SOUT_NAME, GPIO1B_GPIO1B1);

        return 0;

}


static struct sensor_platform_data mxc6225_data = {

        .type = SENSOR_TYPE_ACCEL,

        .irq_enable = 0,

        .poll_delay_ms = 30,

        .init_platform_hw = mxc6225_init_platform_hw,

};

struct i2c_board_info __initdata mxc6225_info = {

        .type                   = "gs_mxc6225",

        .flags                  = 0,

        .platform_data          =&mxc6225_data,

};

#endif
```

```
#if defined (CONFIG_GS_DMT10)

static int dmt10_init_platform_hw(void)

{

        return 0;

}

static struct sensor_platform_data dmt10_data = {

        .type = SENSOR_TYPE_ACCEL,

        .irq_enable = 0,

        .poll_delay_ms = 30,

        .init_platform_hw = dmt10_init_platform_hw,

};

struct i2c_board_info __initdata dmt10_info = {

        .type                       = "gs_dmard10",

        .flags                      = 0,

        .platform_data              =&dmt10_data,

};

#endif

#if defined (CONFIG_GS_MXC6225)

    if(gs_type == GS_TYPE_MXC6225){

        mxc6225_info.irq = port.gpio;

        mxc6225_info.addr = gs_addr;

        for(i = 0; i < 9; i++)

            mxc6225_data.orientation[i] = gs_orig[i];

        i2c_register_board_info(gs_i2c, &mxc6225_info, 1);

    }

#endif
```

```
#if defined (CONFIG_GS_DMT10)
    if(gs_type == GS_TYPE_DMARAD10){
        dmt10_info.irq = port.gpio;
        dmt10_info.addr = gs_addr;
        for(i = 0; i < 9; i++)
            dmt10_data.orientation[i] = gs_orig[i];
        i2c_register_board_info(gs_i2c, &dmt10_info, 1);
    }
#endif
```

6.3 驱动配置



```
.config - Linux/arm 3.0.36 Kernel Configuration
                              g_sensor device support
  Arrow keys navigate the menu.  <Enter> selects submenus --->.  Highlighted letters are hotkeys.  Pressing <Y> includes,
  <N> excludes, <M> modularizes features.  Press <Esc><Esc> to exit, <?> for Help, </> for Search.  Legend: [*] built-in
  [ ] excluded  <M> module  < > module capable

                      --- g_sensor device support
                      [ ]   gsensor mma8452
                      [ ]   gsensor kxtik
                      [ ]   gsensor kxtj9
                      [*]   gsensor lis3dh
                      [*]   gsensor mma7660
                      [*]   gsensor mxc6225
                      [*]   gsensor dmard10
                      [ ]   gsensor bma023

                         <Select>    < Exit >    < Help >
```