

A20 Android 内容定制文档

V1.0

2013-03-15

Confidential

Revision History

Version	Date	Section/ Page	Changes
1.0	2013-03-15		初始版本

目录

一、概述.....	5
二、系统定制.....	5
2.1 启动 LOGO 修改.....	5
2.1.1 Android 初始化 LOGO.....	5
2.1.2 Android 引导动画.....	5
2.2 预装 APK.....	6
2.2.1 预装到 system/app 目录.....	6
2.2.2 预装到 system/preinstall 目录.....	6
2.3 预设桌面制定.....	6
2.4 boot 电池充电图标修改.....	7
2.5 设备相关信息修改.....	7
2.5.1 设备型号及软件版本修改.....	7
2.5.2 USB 相关信息修改.....	7
三、模块配置.....	8
3.1 自定义按键配置.....	8
3.1.1 KEY 的硬件原理.....	8
3.1.2 驱动与硬件对应的关系.....	8
3.1.3 Android 按键功能的映射.....	9
3.2 WIFI 配置.....	10
3.2.1 USB WIFI 配置.....	10
3.2.2 SDIO WIFI 模块配置.....	10
3.3 LCD Panel 配置.....	11
3.4 Touch Panel 配置.....	12
3.4.1 配置文件的修改.....	12
3.4.2 Android 层的配置修改.....	13
3.4.3 touch panel 驱动使用说明.....	14
3.5 G Sensor 配置.....	15
3.5.1 打包配置文件修改.....	15
3.5.2 Android 层配置修改.....	15
3.6 Camera 配置.....	16
3.6.1 打包配置文件修改.....	16
3.6.2 Android 层的配置修改.....	19
3.6.3 Camera 参数配置.....	20
3.7 震动马达配置.....	22
3.7.1 配置文件修改.....	23
3.7.2 Android 层配置修改.....	23

一、概述

本节用于介绍 A20 环境 Android 部分的定制方法以及常用模块的定制介绍。

二、系统定制

本小节主要介绍 android 部分系统定制的内容，其中涉及启动 LOGO 的修改，预装 APK 设置，预设桌面的制定，以及充电图标的修改等

2.1 启动 LOGO 修改

启动 LOGO 为初始引导阶段的 LOGO。

位置：lichee/tools/pack/chips/sun7i/wboot/bootfs/linux/linux.bmp

修改方法：替换 linux.bmp。根据小机屏幕分辨率大小，定制 Logo 图片的大小，替换即可

2.1.1 Android 初始化 LOGO

位置：android/device/softwinner/~~wing-xxx~~/initlogo.rle

修改方法：替换 initlogo.rle。initlogo.rle 是通过 windows 端工具制作而成的文件，制作过程：将 logo 图片转化成 24bpp 或 32bpp bmp 图片，打开 LogoGen(位于 lichee/tools/tools_win)目录下的 BmpConvert 工具，按照《BMPConvert_UserManual.doc》文档说明来生成所需的 initlogo.rle 文件；

说明：framebuffer 使用 ARGB 的格式。

2.1.2 Android 引导动画

位置：android/device/softwinner/~~wing-xxx~~/media/bootanimation.zip

bootanimation 格式：bootanimation.zip 包含 part0 part1 文件夹和 desc.txt 文件，part0, part1 文件夹里面放的是动画拆分的图片，格式为 png 或 jpg；

desc.txt 格式：

800 480 15
p 1 0 part0

p 0 0 part1

说明：800 为宽度，480 为高度，15 为帧数，即每秒播放动画 15 帧；第一项 p 为标志符，第二项为循环次数，1 为只播放 1 次，0 为无限循环，第三项为两次循环之间间隔的帧数，第四项为对应的目录名（图片放在 desc.txt 中目录名指定的目录中，目录中按字符顺序播放）

打包格式要求：windows 使用 winrar 打包，选择 ZIP 格式，压缩标准要选“储存”；linux 下，zip -0 -r ../bootanimation.zip ./ * linux 命令使用 -0 指定压缩等级为最低等级 stored，即只归档不压缩，否则可能由于包格式问题引起动画显示为黑屏。打包完之后发到指定目录 (wing-xxx/media)，修改其权限值：chmod 777 bootanimation.zip

2.2 预装 APK

预装 apk 安装有两种方法，可以安装到 system/app 目录下，也可以安装到 system/preinstall 目录下。

2.2.1 预装到 system/app 目录

预装的 APK 存放位置在 android/device/softwinner/wing-common/apk 目录下，将需要预装的 APK 文件放在该目录下面即可。如果该 apk 包含有 lib，则需要将 lib 库拷贝到 device/softwinner/wing-common/apklib 目录下。

注意：安装到 system/app 目录下的 apk 为系统应用，不能卸载。

2.2.2 预装到 system/preinstall 目录

预装的 APK 存放位置在 android/device/softwinner/wing-common/preinstallapk 目录下。

注意：

- 1) 存放在该目录下的 apk 可以被卸载
- 2) 安装之前会做标志位，如安装过程中出现异常，则其余应用不会自动安装。

注：apk 包的文件名中不能包含中文和空格

2.3 预设桌面制定

目前桌面使用 google 默认的 Launcher2，源码位置为 android/packages/apps/Launcher2，桌面的快捷方式在 android/packages/apps/Launcher2/res/xml/default_workspace.xml 文件中修改：

```
<appwidget
    launcher:packageName="com.google....genie.geniewidget" //widget package name
    launcher:className="com.google....MiniWidgetProvider" //widget class name
    launcher:screen="1" //第几屏
    launcher:x="0" //x\y 坐标
```

```

launcher:y="0"
launcher:spanX="4" //widget 占用 x\y 的格数
launcher:spanY="1" />
<favorite
  launcher:packageName="com.android.camera" //apk package name
  launcher:className="com.android.camera.Camera" //apk activity class name
  launcher:screen="1" //第几屏
  launcher:x="1" //x\y 坐标
  launcher:y="3" />

```

favorite 项为桌面快捷方式，appwidget 项为 widget 的设置，其各项的意义如上面标注所示；

2.4 boot 电池充电图标修改

电池图标是一个 bmp 图片，是位于 lichée/tools/pack/chips/sun7i/wboot/bootfs/os_show 目录下的十一张，名字分别为从 bat0.bmp 到 bat10.bmp 的文件；

如果希望更换这个显示的电池图标，直接用相应分辨率的同名文件替换掉这个文件即可。

2.5 设备相关信息修改

本小节主要讲解 A20 主控下 Android4.2 平台下的设备相关信息的修改。

2.5.1 设备型号及软件版本修改

在编译后生成的 android/out/target/product/*wing-xxx*/system/build.prop 文件中记录了设置中的显示信息，常用的信息（设置-->关于设备）如下：

```

型号 -->ro.product.model
固件版本--> ro.product.firmware
Android 版本-->ro.build.version.release
版本号 -->ro.build.display.id

```

修改方法：源码中修改 android/device/softwinner/wing-xxx/wing_xxx.mk，添加如下：

修改固件版本为 V1.3

```

PRODUCT_PROPERTY_OVERRIDES += \
ro.product.firmware=V1.3

```

修改设备型号为 SoftwinnerEvb

等效于在 build.prop 文件中修改 ro.product.model=SoftwinnerEvb

```

PRODUCT_MODEL := SoftwinnerEvb

```

注意：设备型号中应该全部为英文字符，不能出现中文，可以有空格；

2.5.2 USB 相关信息修改

第一次开机 flash 盘符的修改：在 android/device/softwinner/~~wing-xxx~~/init.sun7i.rc 的“format_userdata” shell 命令

后面接格式化用的盘符，如：

```
format_userdata /dev/block/UDISK WING
```

格式化 flash 盘符为 WING，当打开 USB 设备后即可在电脑上看到给盘符；

用户格式化 flash 盘符修改：修改 android/device/softwinner/wing-xxx/wing_xxx.mk 文件的 ro.udisk.lable=WING 属性。

连接电脑时，显示驱动的修改：

如修改设备连接电脑时，电脑显示驱动为“WING USB 2.0 Driver”，则在 lichee/tools/pack/chips/sun7i/configs/android/~~wing-xxx~~/sys_config.fex 文件中修改如下：

```
[msc_feature]
vendor_name  = "WING"
product_name = "USB 2.0 Driver"
```

设备序列号的修改：默认设备序列号为“20080411”，该序列号主要在豌豆夹手机精灵上有显示，设备的序列号最好为英文大写字母(A~F)及数字的组合，在 sys_config.fex 文件中：

```
[usb_feature]
serial_number = "20080411"
```

三、模块配置

本小节主要讲解 A20 主控下的 Android4.2 的一些外围模块的配置

3.1 自定义按键配置

3.1.1 KEY 的硬件原理

目前 KEY 检测使用了 ADC 转换的原理实现的，由于该原理的限制，所以不能区分组合键（功能键，不包括电源键）；按照目前公版原理图，0.2V 的电压变化可以区分一档，所以最多可以实现 10 个键，硬件原理如下：



```
//0.2V mode
static unsigned char keypad_mapindex[64] =
{
    0,0,0,0,0,0,0,0,          //key 1, 8 个,   0-7
    1,1,1,1,1,1,1,1,          //key 2, 7 个,   8-14
    2,2,2,2,2,2,2,2,          //key 3, 7 个,   15-21
    3,3,3,3,3,3,3,3,          //key 4, 6 个,   22-27
    4,4,4,4,4,4,4,4,          //key 5, 6 个,   28-33
    5,5,5,5,5,5,5,5,          //key 6, 6 个,   34-39
    6,6,6,6,6,6,6,6,6,6,      //key 7, 10 个,  40-49
    7,7,7,7,7,7,7,7,7,7,7,7  //key 8, 17 个,  50-63
};
```

```
static unsigned int sw_scankeycodes[KEY_MAX_CNT] = {
    [0] = KEY_VOLUMEUP,
    [1] = KEY_VOLUMEDOWN,
    [2] = KEY_MENU,
    [3] = KEY_ENTER,
    [4] = KEY_HOME,
    [5] = KEY_RESERVED,
    [6] = KEY_RESERVED,
    [7] = KEY_RESERVED,
    [8] = KEY_RESERVED,
    [9] = KEY_RESERVED,
    [10] = KEY_RESERVED,
    [11] = KEY_RESERVED,
    [12] = KEY_RESERVED,
```

```
};
```

当有按键事件时，通过以上两次映射将最终的键值上报：

```
scancode = keypad_mapindex[key_val&0x3f];
input_report_key(swkbd_dev, sw_scankeycodes[scancode], 1);
input_sync(swkbd_dev);
```

3.1.3 Android 按键功能的映射

映射文件为 android/device/softwinner/*wing-xxx*/sw-keyboard.kl，举例如下：

key 102	HOME	WAKE
key 1	BACK	WAKE
key 139	MENU	WAKE_DROPPED
key 115	VOLUME_UP	
key 114	VOLUME_DOWN	

key 后面的数字为驱动中上报的键值，后面为对应的功能，自定义按键时只需要将前面的映射值和后面的功能对应起来即可；（WAKE_DROPPED：唤醒屏幕，但是这个按键不会发给当前应用程序，WAKE：唤醒屏幕，但是这个按键需要发送给应用程序，后面不加的代表没有唤醒功能）

3.2 WIFI 配置

本小节主要讲解 USB WIFI 模块和 SDIO WIFI 模块的配置

3.2.1 USB WIFI 配置

配置文件：lichee/tools/pack/chips/sun7i/configs/android/*wing-xxx*/sys_config.fex 根据硬件原理图配置‘USB 控制标志’和‘wifi_para’部分，前者标志 usb vcc 电平控制引脚及逻辑，后者标志是 wifi 配置项，举例如下：

wifi_used	= 1	#1 表示使用；0 表示不使用
wifi_sdc_id	= 3	#0: SDC0; 1: SDC1; 2: SDC2; 3SDC3
wifi_usbc_id	= 2	#0: USB0; 1: USB1; 2: USB2
wifi_usbc_type	= 1	#1: ECHI; 2OHCI
wifi_mod_sel	= 7	#0: none; 1: bcm40181; 7: ap6210
wifi_power	= ""	

共有三组 USB，USB0 用于 OTG，USB1/USB2 用于 HOST。wifi 可以使用 USB1 或 USB2。wifi_used 标志平台是否使用 wifi；wifi_usbc_id 标志 wifi 使用哪一组 usb；wifi_usbc_type 标志 usb 接口支持的版本；wifi_mod_sel 标志选用哪一款 wifi 模组；wifi_power 标志 wifi VCC 是哪个 axp pin 控制；可根据自己的硬件原理图情况来配置该部分；

Android 层配置，默认在 android/device/softwinner/*wing-xxx*/BoardConfig.mk 文件中有以

下配置:

```

ifeq ($(BOARD_WIFI_VENDOR), realtek)
    WPA_SUPPLICANT_VERSION := VER_0_8_X
    BOARD_WPA_SUPPLICANT_DRIVER := NL80211
    BOARD_WPA_SUPPLICANT_PRIVATE_LIB := lib_driver_cmd_rtl
    SW_BOARD_USR_WIFI := rtl8723as
    BOARD_WLAN_DEVICE := rtl8723as
endif

```

SW_BOARD_USR_WIFI 和 BOARD_WLAN_DEVICE 是标志选用哪一款 wifi 模组, 这两个宏会直接影响到 hardware/libhardware_legacy/下的 wifi.c 的编译。

3.2.2 SDIO WIFI 模块配置

配置文件: lichee/tools/pack/chips/sun7i/configs/android/~~wing-xxx~~sys_config.fex 根据硬件原理图来确定 sdio wifi 使用哪一组 sdio 卡接口, 按照如下配置 (以 mmc1 为例):

```

[mmc1_para]
sdc_used                = 0      #1:使用; 0: 不使用
sdc_detmode             = 4      #1: 轮训检测; 2: 中断检测; 3: 引导卡; 4: 手动检测
sdc_buswidth            = 4      #1:1bit; 4:4bit; 8:8bit
sdc_clk                 = port:PG00<2><1><2><default>
sdc_cmd                 = port:PG01<2><1><2><default>
sdc_d0                  = port:PG02<2><1><2><default>
sdc_d1                  = port:PG03<2><1><2><default>
sdc_d2                  = port:PG04<2><1><2><default>
sdc_d3                  = port:PG05<2><1><2><default>
sdc_det                 =
sdc_use_wp              = 0      #1: 带有写保护; 0: 无写保护
sdc_wp                  =
sdc_isio                 = 0      #SDIO 卡
sdc_regulator           = "none" #电源控制
[wifi_para]
wifi_used               = 1
wifi_sdc_id             = 3
wifi_usbc_id            = 2
wifi_usbc_type          = 1
wifi_mod_sel            = 7
wifi_power              = ""
; 3 - rtl8723as sdio wifi + bt gpio config
rtk_rtl8723as_wl_dis     = port:PG10<1><default><default><0>
rtk_rtl8723as_bt_dis    = port:PG11<1><default><default><0>
rtk_rtl8723as_wl_host_wake = port:PG12<0><default><default><0>
rtk_rtl8723as_bt_host_wake = port:PG17<0><default><default><0>

```

mmc1_para 部分配置 sdio 通信 data 及 clk 信号, 一般不改动, wifi_para 部分配置 sdio wifi 使用哪个 sdc 卡接口、相关控制 GPIO 的选择, 请根据实际原理图来配置;

Android 层配置: 此部分跟 usb wifi 配置中的 android 层配置一样。

注: 因 wifi 部分和 android framework 配置层关联比较大, 更改 wifi 配置后, 建议 make clean 之后再编译, 否则会出现 wifi 无效的情况;

3.3 LCD Panel 配置

配置文件 : lichee/tools/pack/chips/sun7i/configs/android/~~wing-xxx~~/sys_config.fex

调试 lcd pannel 的参数: 如发现屏幕闪动或显示位置有偏差, 可以按照该 panel 的 spec 调整如下参数:

参数调整: 用户需要根据 panel 的 datasheet, 调整如下参数: (仅供参考)

```
[lcd0_para]
lcd_used                = 1          #1: 使能; 0 不使能
lcd_x                   = 800        #x 方向的分辨率
lcd_y                   = 480        #y 方向的分辨率
lcd_dclk_freq           = 33         #in MHZ unit
lcd_pwm_not_used       = 0
lcd_pwm_ch              = 0
lcd_pwm_freq           = 10000      #in HZ unit
lcd_pwm_pol            = 0
lcd_if                  = 0          #0: hv; 1:2020; 2: ttl; 3: lvds
lcd_hbp                 = 46
lcd_ht                  = 1055
lcd_vbp                 = 23
lcd_vt                  = 1050
lcd_vspw                = 0
lcd_hspw                = 0
lcd_hv_if               = 0
lcd_hv_smode            = 0
lcd_hv_s888_if          = 0
lcd_hv_syuv_if          = 0
lcd_lvds_ch             = 0
lcd_lvds_mode           = 0
lcd_lvds_bitwidth       = 0
lcd_lvds_io_cross       = 0
lcd_cpu_if              = 0
lcd_frm                 = 0
lcd_io_cfg0             = 0x10000000
lcd_gamma_correction_en = 0
lcd_gamma_tbl_0         = 0x00000000
```

```

lcd_gamma_tbl_1      = 0x00010101
lcd_gamma_tbl_255    = 0x00ffffff
lcd_bl_en_used       = 1
lcd_bl_en            = port:PH07<1><0><default><1>
lcd_power_used       = 1
lcd_power            = port:PH08<1><0><default><1>
lcd_pwm_used         = 1
lcd_pwm             = port:PB02<2><0><default><default>
...

```

3.4 Touch Panel 配置

发布的 SDK 中，默认有对电阻屏（两点）、FT5302/FT5406/FT5506/FT5606（敦泰）、GT813/GT827/GT828（惠鼎）、GSL2680/GSL3680(思立微)的支持；

3.4.1 配置文件的修改

打包配置文件修改：lichee/tools/pack/chips/sun7i/configs/android/**wing-xxx**/sys_config.fex
如果是电阻屏：

```

[ctp_para]
ctp_used           = 1
ctp_twi_id         = 2
ctp_twi_name       =
ctp_screen_max_x   = 800
ctp_screen_max_y   = 480
ctp_revert_x_flag  = 0
ctp_revert_y_flag  = 0
ctp_exchange_x_y_flag = 0

```

建议只修改 rtp_screen_size 和 rtp_regidity_level，其他参数请暂不要修改；
如果是电容屏：

```

[ctp_para]
ctp_used           = 1
ctp_twi_id         = 2           #使用哪组 I2C
;ctp_twi_addr      = 0x5d       #I2C 设备地址（7 位地址）
ctp_screen_max_x   = 800        #touch panel 的分辨率
ctp_screen_max_y   = 480
ctp_revert_x_flag  = 0           #如果 x 轴反向，请置为 1
ctp_revert_y_flag  = 0           #如果 y 轴反向，请置为 1
ctp_exchange_x_y_flag = 0       #如果 x、y 倒置，请置为 1
ctp_int_port       = port:PA03<6><default><default><default>
ctp_wakeup         = port:PA02<1><default><default><1>

```

Touch panel 的 GPIO 请正确配置中断引脚与 wakeup (reset)引脚;

3.4.2 Android 层的配置修改

在 android/device/softwinner/*wing-xxx*/init.sun7i.rc 文件中加入装载驱动模块的语句:

```
insmod /system/vendor/modules/gslX680.ko
```

IDC 文件修改

Android4.0 之后, 配置文件中需要一个 idc 文件来识别输入设备为触摸屏还是鼠标, 如果没有该文件, 则默认为鼠标, 因此需要添加该文件。在产品目录中, 需要将原有文件的文件名修改为与驱动注册的名称一致 (adb shell 中 getevent 显示的设备名称)。如 gslX680 中, 则使用的 idc 文件为: gslX680.idc。且在配置文件中修改相应的文件名。配置文件为 wing_xx.mk。idc 文件放置的目录为:system/usr/idc ,则在改目录中的拷贝语句以 onda809q9 为例进行说明, 如下所示:

```
PRODUCT_COPY_FILES += \
device/softwinner/wing-xxx/sw-keyboard.kl:system/usr/keylayout/sw-keyboard.kl \
device/softwinner/wing-xxx/tp.idc:system/usr/idc/tp.idc \
```

3.4.3 touch panel 驱动使用说明

gslX680 使用说明: gslX680 驱动兼容 gsl2680, gsl3680。为了区分下载的参数, 在文件配置说明如下:

ctp_tw_i_id	使用哪组 I2C
ctp_name	设备的名字, 用于同一驱动中, 区别不同型号的 TP
ctp_tw_i_addr	I2C 设备地址 (7 位地址)
ctp_screen_max_x, ctp_screen_max_y	touch panel 的分辨率
ctp_revert_x_flag	如果 x 轴反向, 请置为 1
ctp_revert_y_flag	如果 y 轴反向, 请置为 1
ctp_exchange_x_y_flag	如果 x、y 倒置, 请置为 1
ctp_int_port	触摸中断引脚
ctp_wakeup	触摸 wakeup 引脚
ctp_io_port	中断引脚配置

触摸模块中的 GPIO 需要正确的配置中断引脚、wakeup (reset), 根据平台中使用的引脚进行相应的配置。事例配置如下

```
[ctp_para]
ctp_used          = 1
ctp_name          = "gsl1680"
```

```
ctp_twi_id          = 2
ctp_twi_addr        = 0x40
ctp_screen_max_x     = 1024
ctp_screen_max_y     = 600
ctp_revert_x_flag    = 0
ctp_revert_y_flag    = 0
ctp_exchange_x_y_flag = 1
```

当更换 tp 屏厂，或者是修改 tp 屏参数时，将参数替换掉 gslX680.h 中的原有参数即可。该文件路径为：lichee/linux-3.3/drivers/input/touchscreen/gslX680.h

gsl2680 使用参数位于 GSL2680_FW 数组中，由于数据比较多，更换时请确保更换完整。
gsl3680 使用参数位于 GSL3680_FW 数组中，由于数据比较多，更换时请确保更换完整。

gt82x 使用说明：gt82x 为兼容 gt813，gt827，gt828。初始化时需要根据具体的 tp 屏下载相应的参数。参数位于 lichee/linux-3.3/drivers/input/touchscreen/gt82x.c 文件中的 goodix_init_panel 函数中，目前已有一组 813 以及 828 参数，根据 chip id 值进行相应参数的下载。当更换屏厂或者是更新 tp 屏参数时，修改该函数中相应的数组即可。

gt813 参数数组 data_info0，gt828 参数数组 data_info1。参数由 tp 屏厂提供，注意，前两个数据（0x0F、0x80）为寄存器地址，不要变动，只替换后面的 112 个数据即可。

3.5 G Sensor 配置

发布的 SDK 中已添加了对 MMA8452、MMA8652、MMA8653、LIS3DH、LIS3DE AFA750、dmard06 及 BMA250 G-Sensor 的支持，需要客户根据需要做如下配置：

3.5.1 打包配置文件修改

打包配置文件修改：lichee/tools/pack/chips/sun7i/configs/android/*wing-xxx*/sys_config.fex
G sensor 的配置文件事例如下，具体配置参考芯片硬件手册

```
[gsensor_para]
gsensor_used      = 1
gsensor_twi_id    = 1
gsensor_int1      =
gsensor_int2      =
```

只需要配置 gsensor_used 与 gsensor_twi_id 即可，gsensor_used 代表是否支持 gsensor，gsensor_twi_id 代表 I2C 总线号。

3.5.2 Android 层配置修改

以 MMA8452 为例：

Android 中，在 android/device/softwinner/*wing-xxx*/init.sun7i.rc 文件中加入装载驱动模块的语句：

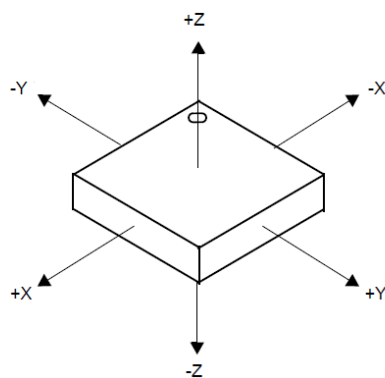
```
insmod /system/vendor/modules/mma8452.ko
```

方向的调整:

在 android/device/softwinner/*wing-xxx*/gsensor.cfg

```
gsensor_name = mma8452 //标示用 mma8452 gsensor
gsensor_direct_x = true//如果 x 轴反向, 则置 false
gsensor_direct_y = true//如果 y 轴反向, 则置 false
gsensor_direct_z = false//如果 z 轴反向, 则置 false
gsensor_xy_revert = true//如果 x 轴当 y 轴用, y 轴当 x 轴, 则置 true
```

MMA8452 实际方向如下 (参见 8452 Datasheet):



3.6 Camera 配置

发布的 SDK 中已添加了对 gc0307, gc0308, gc2035, gt2005, hi253, ov5640, s5k4ec 的支持;

3.6.1 打包配置文件修改

配置文件位置: lichee/tools/pack/chips/sun7i/configs/android/*wing-xxx*/sys_config.fex
A20 上面使用了 CSI0 和 CSI1, 驱动里面需要配置[csi0_para], [csi1_para]段落。

```
[csi0_para]
csi_used          = 1
csi_dev_qty       = 1
csi_stby_mode     = 0
csi_mname         = "gc0308"
csi_if            = 0
csi_iovdd         = ""
csi_avdd          = ""
csi_dvdd          = ""
```



```

csi_vol_iovdd      =
csi_vol_dvdd       =
csi_vol_avdd        =
csi_vflip           = 0
csi_hflip           = 0
csi_flash_pol       = 0
csi_facing          = 0
csi_twi_id          = 1
csi_twi_addr        = 0x42
csi_pck             = port:PE00<3><default><default><default>
csi_ck              = port:PE01<3><default><default><default>
csi_hsync           = port:PE02<3><default><default><default>
csi_vsync           = port:PE03<3><default><default><default>
csi_d0              = port:PE04<3><default><default><default>
csi_d1              = port:PE05<3><default><default><default>
csi_d2              = port:PE06<3><default><default><default>
csi_d3              = port:PE07<3><default><default><default>
csi_d4              = port:PE08<3><default><default><default>
csi_d5              = port:PE09<3><default><default><default>
csi_d6              = port:PE10<3><default><default><default>
csi_d7              = port:PE11<3><default><default><default>
csi_reset           = port:PH13<1><default><default><0>
csi_power_en        =
csi_stby            = port:PH16<1><default><default><0>

```

根据使用的 Camera 型号来设置如上子项，根据原理图来设置 reset、power 及 standby 引脚的 gpio 和控制逻辑；

配置项	配置项含义
csi_used=xx	是否使用 csi0
csi_twi_id=xx	csi 使用的 IIC 通道序号，查看具体方案原理图，使用 twi0 填 0
csi_mname=xx	csi 使用的模组名称，需要与驱动匹配，可以查看驱动目录里面的 readme 目前有 gc0307, gc0308, gc2035, gt2005, hi253, ov5640, s5k4ec 可选
csi_twi_addr=xx	csi 使用的模组的 IIC 地址（8bit 地址），可以查看驱动目录里面的 readme
csi_if	配置目前使用模组的接口时序： 0:8bit 数据线，带 Hsync,Vsync 1:16bit 数据线，带 Hsync,Vsync 2:24bit 数据线，带 Hsync,Vsync 3:8bit 数据线,BT656 内嵌同步,单通道

	4:8bit 数据线,BT656 内嵌同步,双通道 5:8bit 数据线,BT656 内嵌同步,四通道
csi_mode	配置 csi 接收 buffer 的模式: 0: 一个 CSI 接收对应一个 buffer 1: 两个 CSI 接收内容拼接成一个 buffer
csi_dev_qty	配置 csi 目前连接的器件数量,目前只能配置为 1 或 2
csi_vflip	配置 csi 接收图像默认情况下, 上下颠倒情况: 0: 正常 1: 上下颠倒
csi_hflip	配置 csi 接收图像默认情况下, 左右颠倒情况: 0: 正常 1: 左右颠倒
csi_stby_mode	配置 csi 在进入 standby 时的处理: 0: 不关闭电源, 只拉 standby io 1: 关闭电源, 同时拉 standy io
csi_iovdd	配置 csi iovdd 电源来源: 请查看对应方案原理图, 一般填写的名字为"axp22_XldoN"等 (注意带英文字符的双引号, 不使用 axp 电源供电时候请务必留空引号""") 如 EVB 上, 配置成"axp22_eldo3"
csi_avdd	配置 csi avdd 电源来源: 请查看对应方案原理图, 一般填写的名字为"axp22_XldoN"等 (注意带英文字符的双引号, 不使用 axp 电源供电时候请务必留空引号"""), 这个地方请特别注意, 因为此电源对于 sensor 图像质量关系较大, 对于高像素 sensor 建议使用 axp22_ldoio0 或 axp22_ldoio1 这两组电源或者采用外挂带 EN 控制的 LDO
csi_dvdd	配置 csi dvdd 电源来源: 请查看对应方案原理图, 一般填写的名字为"axp22_XldoN"等 (注意带英文字符的双引号, 不使用 axp 电源供电时候请务必留空引号""")
csi_vol_iovdd	配置 csi iovdd 电源电压 如果 csi_iovdd 配置不为空时会配置对应的 axp 电源为相应电压

	配置为 2800 表示 2.8V，范围不要超过 1800~2800，请查看具体 sensor 的 datasheet 填写此电压
csi_vol_avdd	配置 csi avdd 电源电压 如果 csi_avdd 配置不为空时会配置对应的 axp 电源为相应电压 配置为 2800 表示 2.8V，一般不要修改此数值
csi_vol_dvdd	配置 csi dvdd 电源电压 如果 csi_dvdd 配置不为空时会配置对应的 axp 电源为相应电压 配置为 1500 表示 1.5V，范围不要超过 1200~1800，请查看具体 sensor 的 datasheet 填写此电压
csi_pck=xx	模组送给 csi 的 clock 的 GPIO 配置
csi_ck=xx	csi 送给模组的 clock 的 GPIO 配置
csi_hsync=xx	模组送给 csi 的行同步信号 GPIO 配置
csi_vsync=xx	模组送给 csi 的帧同步信号 GPIO 配置
csi_d0=xx ... csi_d23=xx	模组送给 csi 的 8bit/16bit/24bit 数据的 GPIO 配置，使用 YUV 格式的 sensor 方案中，csi_d0/d1/d2/d3 会被配置成普通 GPIO，用来控制 sensor 的 pwn/reset 信号，使用 RAW 格式的 sensor 只能用 csi_d0/d1 作 GPIO 用途。
csi_reset=xx	控制模组的 reset 的 GPIO 配置，默认值为 reset 有效（高或低有效需要取决于模组）
csi_power_en=xx	控制模组的电源的 GPIO 配置，若 csi_stby_mode 配置成 0，则 csi_power_en 的默认值一般配置成 1；若 csi_stby_mode 配置成 1，则 csi_power_en 的默认值一般配置成 0。
csi_stby=xx	控制模组的 standby 的 GPIO 配置，默认值为 standby 有效（高或低有效需要取决于模组）
csi_reset_b=xx	如果有两个模组同时连接到一个 CSI，需要额外的 IO 控制；控制模组的 reset 的 GPIO 配置，默认值为 reset 有效（高或低有效需要取决于模组）
csi_power_en_b=xx	如果有两个模组同时连接到一个 CSI，需要额外的 IO 控制；控制模组的电源的 GPIO 配置，若 csi_stby_mode 配置成 0，则 csi_power_en 的默认值一般配置成 1；

	若 <code>csi_stby_mode</code> 配置成 1，则 <code>csi_power_en</code> 的默认值一般配置成 0。
<code>csi_stby_b=xx</code>	如果有两个模组同时连接到一个 CSI，需要额外的 IO 控制；控制模组的 standby 的 GPIO 配置，默认值为 standby 有效（高或低有效需要取决于模组）

3.6.2 Android 层的配置修改

Android 中，在 `android/device/softwinner/wing-xxx/init.sun7i.rc` 文件中加入装载驱动模块的语句：

```
单摄像头：
insmod /system/vendor/modules/videobuf-core.ko
insmod /system/vendor/modules/videobuf-dma-contig.ko
insmod /system/vendor/modules/gc0308.ko
insmod /system/vendor/modules/sunxi_csi0.ko
```

在 `android/device/softwinner/wing-xxx/ueventd.sun7i.rc` 文件中改变相关设备节点的权限：

<code>/dev/video0</code>	0666	media	media
--------------------------	------	-------	-------

3.6.3 Camera 参数配置

配置文件路径：`android/device/softwinner/wing-xxx/camera.cfg`

事例内容简介：

```
number_of_camera = 1    #camer 模块的数量(1/2)
camera_id = 0
camera_facing = 0       #1: 前置摄像头；0 后置摄像头
camera_orientation = 0  #camer 模块的方向(0/90/180/270)
camera_device = /dev/video0    #设备文件接口
device_id = 0           #设备 id，两个 camera 使用一个 CSI
used_preview_size = 1
key_support_preview_size = 640x480
key_default_preview_size = 640x480
used_picture_size = 1
key_support_picture_size = 640x480
key_default_picture_size = 640x480
used_flash_mode = 0
key_support_flash_mode = on,off,auto
key_default_flash_mode = on
used_color_effect=1
```

```
key_support_color_effect = none,mono,negative,sepia,aqua
key_default_color_effect = none
used_frame_rate = 1
key_support_frame_rate = 25
key_default_frame_rate = 25
used_focus_mode = 0
key_support_focus_mode = auto,infinity,macro,fixed
key_default_focus_mode = auto
used_scene_mode = 0
key_support_scene_mode =
auto,auto,portrait,landscape,night,night-portrait,theatre,beach,snow,sunset,ste
adyphoto,fireworks,sports,party,candlelight,barcode
key_default_scene_mode = auto
used_white_balance = 1
key_support_white_balance =
auto,incandescent,fluorescent,warm-fluorescent,daylight,cloudy-daylight
key_default_white_balance = auto
used_exposure_compensation = 1
key_max_exposure_compensation = 3
key_min_exposure_compensation = -3
key_step_exposure_compensation = 1
key_default_exposure_compensation = 0
; only for facing back camera in android2.3, should be set in android4.0
used_zoom = 1
key_zoom_supported = true
key_smooth_zoom_supported = false
key_zoom_ratios = 100,120,150,200,230,250,300
key_max_zoom = 30
key_default_zoom = 0
camera_orientation = 0
...
```

media_profiles.xml 的路径: android/device/softwinner/*wing-xxx*/media_profiles.xml

内容简介: 该文件主要保存 Camera 支持的摄像相关参数, 包括摄像质量, 音视频编码格式、帧率、比特率等等, 该参数主要有摄像头厂商提供: (以下 Demo 对应两个摄像头的情况, 如果只有一个 camera 则只需要一份参数)

```
<MediaSettings>
  <CamcorderProfiles>
    <EncoderProfile quality="480p" fileFormat="mp4" duration="60">
      <Video codec="h264"
        bitRate="1000000"
        width="640"
        height="480">
```

```

frameRate="30" />

    <Audio codec="aac"
        bitRate="12200"
        sampleRate="44100"
        channels="1" />
    </EncoderProfile>

    <EncoderProfile    quality="timelapse480p"    fileFormat="mp4"
duration="60">
        <Video codec="h264"
            bitRate="1000000"
            width="640"
            height="480"
            frameRate="30" />

        <Audio codec="aac"
            bitRate="12200"
            sampleRate="44100"
            channels="1" />
        </EncoderProfile>
        <ImageEncoding quality="90" />
        <ImageEncoding quality="80" />
        <ImageEncoding quality="70" />
        <ImageDecoding memCap="20000000" />
        <Camera previewFrameRate="0" />
    </CamcorderProfiles>
    <EncoderOutputFileFormat name="mp4" />
    <VideoEncoderCap name="h264" enabled="true"
        minBitRate="64000" maxBitRate="3000000"
        minFrameWidth="320" maxFrameWidth="640"
        minFrameHeight="240" maxFrameHeight="480"
        minFrameRate="1" maxFrameRate="30" />

    <AudioEncoderCap name="aac" enabled="true"
        minBitRate="5525" maxBitRate="12200"
        minSampleRate="8000" maxSampleRate="44100"
        minChannels="1" maxChannels="1" />

    <AudioEncoderCap name="amrwb" enabled="true"
        minBitRate="6600" maxBitRate="23050"
        minSampleRate="16000" maxSampleRate="16000"

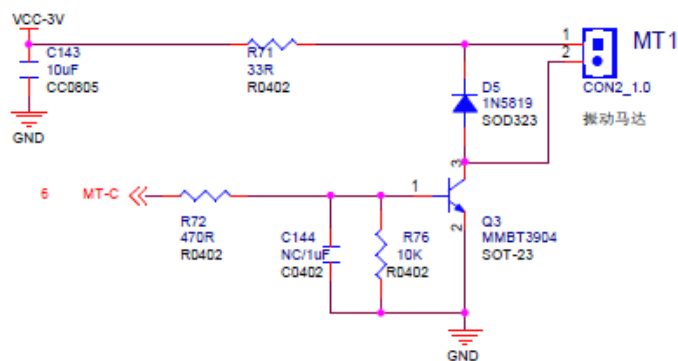
```

```
minChannels="1" maxChannels="1" />
```

```
<AudioEncoderCap name="amrnb" enabled="true"
  minBitRate="5525" maxBitRate="12200"
  minSampleRate="8000" maxSampleRate="8000"
  minChannels="1" maxChannels="1" />
<VideoDecoderCap name="wmv" enabled="true"/>
<AudioDecoderCap name="wma" enabled="true"/>
<VideoEditorCap maxInputFrameWidth="1920"
  maxInputFrameHeight="1080" maxOutputFrameWidth="1920"
  maxOutputFrameHeight="1080" maxPrefetchYUVFrames="10"/>
<ExportVideoProfile name="m4v" profile="1" level="128"/>
</MediaSettings>
```

3.7 震动马达配置

震动马达部分的电路比较单一，软件端只用控制 CHGLED 引脚的电平就可以打开和关闭马达震动；



3.7.1 配置文件修改

配置文件路径: `lichee/tools/pack/chips/sun7i/configs/android/wing-xxx/sys_config.fex`
根据硬件原理图进行相关参数的配置

[motor_para]

motor_used = 1 #是否启用马达，启用置 1，反之置 0

motor_shake = port:power3<1><default><default><1> #马达震动的 GPIO 选择及控制逻辑：0 代表低电平关闭，高电平打开；1 代表高电平关闭，低电平打开，请根据实际电路图来配置；

3.7.2 Android 层配置修改

在 android/device/softwinner/~~wing-xxx~~/init.sun7i.rc 文件中加入改变相关设备节点及装载驱动模块的语句:

```
# insmod vibrator  
insmod /system/vendor/modules/sun7i-vibrator.ko  
chmod 777 /sys/class/timed_output/sun7i-vibrator/enable
```