



# A10 Android4.0 内容定制说明文档

V1.0

2012-7-11



## Revision History

Version	Date	Section/ Page	Changes compared to previous issue
1.0	20120711		initialition



## 目录

<b>V1.0</b> .....	1
<b>2012-7-11</b> .....	1
<b>Revision History</b> .....	2
1. A10 crane 定制化内容说明 .....	4
1.1 Logo & Animation .....	4
1.1.1. Boot 阶段 Logo .....	4
1.1.2. Android Logo .....	4
1.1.3. Boot Animation .....	4
1.1.4. Android Logo & Animation 的编译 .....	5
1.2 设备信息修改 .....	5
1.2.1. 设备型号及软件版本 .....	5
1.2.2. USB 相关 .....	6
1.3 APK 预装 .....	6
1.4 预设桌面方案 .....	7
1.5 自定义按键 .....	7
1.6 Wifi 设置 .....	9
1.7 LCD Pannel 修改 .....	13
1.8 Touch Pannel 配置 .....	16
1.9 G-Sensor 配置 .....	18
1.10 Camera 的配置 .....	20
1.11 震动马达的配置 .....	29
1.12 GPS 配置 .....	29
1.13 Boot 电流充电图标配置 .....	31
1.14 Recovery 功能使用 .....	31
2. Android 标准常用修改 .....	34
2.1 预定义动态壁纸 .....	34



# 1. A10 crane 定制化内容说明

## 1.1 Logo & Animation

### 1.1.1. Boot 阶段 Logo

位置: exdroid\lichee\tools\pack\chips\sun4i\wboot\bootfs\linux, 采用 argb24bit 或 32bit 的图替换原图即可。

### 1.1.2. Android Logo

位置: android4.0\device\softwinner\ crane-xx\initlogo.rle

制作过程: 将 logo 图片转化成 24bpp 或 32bpp bmp 图片, 打开 LogoGen 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: ARGB 值是针对 fb0\_pixel\_sequence=0 的配置值, 如果 fb1\_pixel\_sequence = 2, 则选择 ABGR, fb0\_pixel\_sequence 在 sys\_config1.fex 中填写, 是 disp\_init 的一个子键;

```

;-----
;disp init configuration
;fbx pixel sequence (0:ARGB 1:BGRA 2:ABGR 3:RGBA) --- 0 for linux, 2 for android
;-----
[disp_init]
disp_init_enable = 1
.....
fb0_pixel_sequence = 2
.....
fb1_pixel_sequence = 2

```

### 1.1.3. Boot Animation

位置: android4.0\device\softwinner\ crane-xx\media\bootanimation.zip

(根文件系统位置: /system/media/bootanimation.zip)

bootanimation 格式: bootanimation.zip 包含 part1 part2 文件夹和 desc.txt, 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：

1. windows 使用 winrar 打包，选择 ZIP 格式，压缩标准要选“储存”；
2. 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

注意：打包时不要带上层目录；

打包完之后发到指定目录，修改其权限值：

```
chmod 777 bootanimation.zip
```

### 1.1.4. Android Logo & Animation 的编译

加入 initlogo.rle 和 bootanimation.zip 后，要修改编译文件的拷贝方法，在 android4.0\device\softwinner\ crane-xx\crane\_xx.mk 文件中加入：

```
#initlogo and bootanimation
PRODUCT_COPY_FILES += \
    device/softwinner/crane-xx/media/bootanimation.zip:system/media/bootanimation.zip \
    device/softwinner/crane-xx/initlogo.rle:root/initlogo.rle
```

分号前面标示源文件位置，后面标示编译后拷贝到的根文件系统位置；

## 1.2 设备信息修改

### 1.2.1. 设备型号及软件版本

在编译后生成的 android4.0\out\target\product\crane-xx\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

内核版本是动态读取出来的~

修改方法：

1. 编译后直接在 out\target\product\crane-xx\system\build.prop 文件中修改以上信息再 mking 打包即可；

2. 源码中修改 android4.0\device\softwinner\ crane-xx\crane\_xx.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 ONDA MID（等效于在 build.prop 文件中修改 o.product.model=ONDA MID）
```

```
PRODUCT_MODEL := ONDAMID
```



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；

## 1.2.2. USB 相关

### ➤ flash 盘符的修改：

在 android4.0\device\softwinner\crane-xx\init.sun4i.rc 中 ‘format\_userdata’ shell 命令后面接格式化用的盘符，如：

```
format_userdata ONDA_MID #刷机第一次开机时格式化 flash 盘符为
```

```
ONDA_MID，当打开 USB 设备后即可在电脑上看到给盘符；
```

如果用户手动格式化 flash 时，盘符信息将被删除掉，那么需要重新将盘符信息写到 flash，所以还需要在 android4.0\device\softwinner\crane-xx\crane\_xx.mk 中添加如下：

```
PRODUCT_PROPERTY_OVERRIDES += \
```

```
ro.udisk.lable= ONDA_MID
```

### ➤ 连接电脑时，显示驱动力的修改：

如修改设备连接电脑时，电脑显示驱动为 “ONDA MID USB 2.0 Driver”，则在

```
sys_config1.fex
```

文件中修改如下：

```
[msc_feature]
```

```
vendor_name = "ONDA MID"
```

```
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\_config1.fex 文件中：

```
[usb_feature]
```

```
serial_number = "20080411"
```

## 1.3 APK 预装

预装的 APK 存放位置在 android4.0\device\softwinner\crane-xx\apk\目录下，将需要预装的 APK 文件放在该目录下面，该目录下的 apk 安装有两种方法：

(1) 安装到 system/app 目录下：

修改 android4.0\device\softwinner\ crane-xx\crane\_xx.mk 如下：

```
# app apks
```

```
PRODUCT_COPY_FILES += \
```

```
$(call find-copy-subdir-files,*.apk,$(LOCAL_PATH)/apk,system/app)
```

优点：对启动速度无影响（影响第一次烧录后启动速度）；

缺点：对于 apk 中有 lib 库文件的应用，预装在该目录后，不能正常使用该应用；

(2) 安装到 system/preinstall 目录下：

```
# pre-installed apks
```

```
PRODUCT_COPY_FILES += \
```



```
$(call find-copy-subdir-files,*.*apk,$(LOCAL_PATH)/apk,system/preinstall)
```

优点：在系统烧录后第一次启动进入系统之后才会安装这些应用，不影响开机时间，可以有效解决预装 apk（带 lib 库）不能使用的问题；

缺点：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；

综合以上两点，对于带 lib 库的应用，想在一开机系统就使用的，如开机动态壁纸，则可以按以下来做，将 apk 的 lib 库文件提取出来预安装在 /system/lib 目录下，将 apk 安装到 system/app 目录下，这样可以保证系统在进入系统之前可以正常安装这些带有 lib 库文件的应用：

```
# pre-installed start apks
```

```
PRODUCT_COPY_FILES += \
```

```
    $(call find-copy-subdir-files,*.*apk,$(LOCAL_PATH)/apk_start,system/app)
```

```
# pre-installed start apks
```

```
PRODUCT_COPY_FILES += \
```

```
    $(call find-copy-subdir-files,*.*so,$(LOCAL_PATH)/lib,system/lib)
```

## 1.4 预设桌面方案

目前桌面使用 google 默认的 Launcher2，源码位置为 android4.0/packages/apps/Launcher2，桌面的快捷方式在 android4.0/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
```

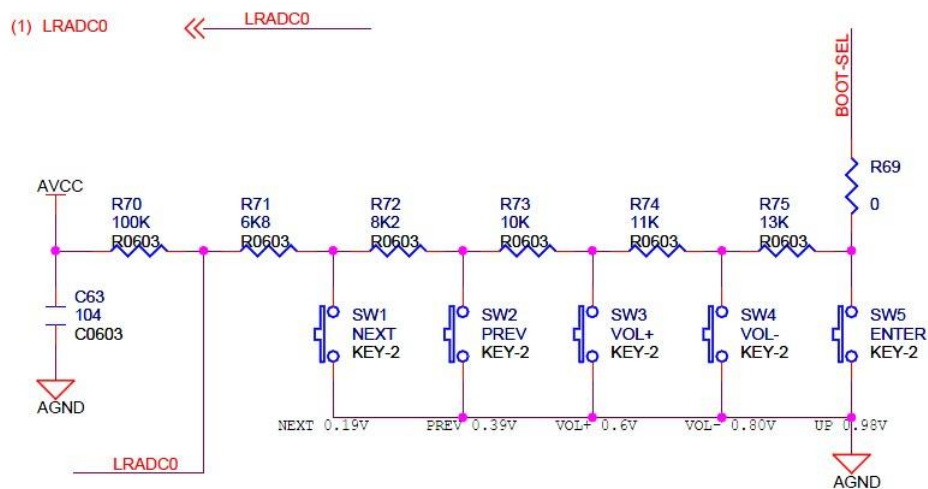
```
    launcher:x="1" //x\y 坐标
```

```
    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

## 1.5 自定义按键

- (1) 硬件：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；



(2) 驱动与硬件的对应关系:

Key 的驱动实现文件位置: kernel/linux3.0/drivers/input/keyboard/sun4i-keyboard.c;

实现原理:

a.通过检测电压值的数字量来区分当前是第几个按键:

```
//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,	//key 2, 7 个, 8-14
2,2,2,2,2,2,2,	//key 3, 7 个, 15-21
3,3,3,3,3,3,	//key 4, 6 个, 22-27
4,4,4,4,4,4,	//key 5, 6 个, 28-33
5,5,5,5,5,5,	//key 6, 6 个, 34-39
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

```
};
```

b.按键对应的键值:

```
static unsigned int sun4i_scankeycodes[KEY_MAX_CNT]=
```

```
{
```

```
[0] = 163, //KEY_NEXTSONG,
[1] = 165, //KEY_PREVIOUSSONG,
[2] = KEY_VOLUMEUP,
[3] = KEY_VOLUMEDOWN,
[4] = KEY_ENTER,
[5] = KEY_RESERVED,
[6] = KEY_RESERVED,
[7] = KEY_RESERVED,
[8] = KEY_RESERVED,
```





```
[9] = KEY_RESERVED,
[10] = KEY_RESERVED,
[11] = KEY_RESERVED,
[12] = KEY_RESERVED,
};
```

- c. 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(sun3ikbd_dev, sun3i_scankeycodes[scancode], 1);
input_sync(sun3ikbd_dev);
```

(3) Android 键值功能映射：

映射文件为 android4.0\device\softwinner\crane\_xx\ sun4i-keyboard.kl，举例如下：

```
key 102    HOME           WAKE
key 115    BACK            WAKE
key 114    MENU            WAKE_DROPPED
```

key后面的数字为驱动中上报的键值，后面为对应的功能，自定义按键时仅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）

注：如果仅想快速地匹配按键，请按照培训测验题目第一道题的做法来实现快速配置（进入 adb shell，用 getevent 命令获取按键映射值，重填 sun4i-keyboard.kl 文件）；

## 1.6 Wifi 设置

(1) USB Wifi 的配置：

- 配置文件：sys\_config1.fex

根据硬件原理图配置 ‘USB 控制标志’ 和 ‘usb\_wifi\_para’ 部分，前者标志 usb vcc 电平控制引脚及逻辑，后者标志使用哪一组 usb，举例如下：

```
;-----
;---      USB0 控制标志
;-----
[usbc0]                                     ;usb0 一般用于 otg 功能
usb_used          = 1
usb_port_type     = 2
usb_detect_type   = 1
usb_id_gpio       = port:PH4<0><1><default><default>
usb_det_vbus_gpio = port:PH5<0><0><default><default>
usb_drv_vbus_gpio = port:PB9<1><0><default><0>
usb_host_init_state = 0
;-----
```



```

;---          USB1 控制标志
;-----
[usbc1]                                     ;usb1 根据原理图决定是否需要打开
usb_used                                     = 0
usb_port_type                               = 1
usb_detect_type                             = 0
usb_id_gpio                                 =
usb_det_vbus_gpio                           =
usb_drv_vbus_gpio                           = port:PH6<1><0><default><0>
usb_host_init_state                         = 1

;-----
;---          USB2 控制标志
;-----
[usbc2]                                     ;usb2 打开，用于 usb wifi
usb_used                                     = 1
usb_port_type                               = 1
usb_detect_type                             = 0
usb_id_gpio                                 =
usb_det_vbus_gpio                           =
;设置 wifi vcc 控制 gpio 为 PH3，且配置控制逻辑为高电平打开；
usb_drv_vbus_gpio                           = port:PH3<1><0><default><0>
usb_host_init_state                         = 0 ;设置为 0，代表 usb_drv_vbus_gpio 会随着 wifi 的打开或关
闭而有电平高低的变化，设置为 1，则该电平无法动态控制，会一直保持初始化的电平
状态；

;-----
;usb wifi configuration
;usb_wifi_id  --- 0- USB0, 1- USB1, 2- USB2
;-----
[usb_wifi_para]                             ;设置 usb wifi，且选择使用 usb2
usb_wifi_used                               = 1
usb_wifi_usbc_num                           = 2

```

共有三组 USB，USB0 用于 OTG，USB1/USB2 用于 HOST，wifi 可以使用 USB1 或 USB2，usb\_wifi\_usbc\_num 标志 wifi 使用哪一组 usb，usb\_drv\_vbus\_gpio 标志 wifi VCC 是哪个 GPIO 控制，可根据自己的硬件原理图情况来配置该部分；

usb\_drv\_vbus\_gpio 对 wifi power 的控制分以下两种情况：

- 1) 该 GPIO 为高时 wifi vcc 上电，反之 wifi vcc 拉低：
  - 》usb\_drv\_vbus\_gpio = port:PH3<1><0><default><0>
- 2) 该 GPIO 为低时 wifi vcc 上电，反之 wifi vcc 拉低：
  - 》usb\_drv\_vbus\_gpio = port:PH3<1><0><default><1>



➤ Android 层配置:

默认在 android4.0\device\softwinner\crane-common\ BoardConfigCommon.mk 文件中有以下配置:

```
# wifi
AVE_CUSTOM_WIFI_DRIVER_2 := true
BOARD_WPA_SUPPLICANT_DRIVER := WEXT
WPA_SUPPLICANT_VERSION := VER_0_6_X
# usb wifi "rtl8192cu"; sdio wifi "nanowifi"
BOARD_USR_WIFI := nanowifi
```

配置成 usb wifi 请在 android4.0\device\softwinner\crane\_xx\ BoardConfig.mk 中重写 BOARD\_USR\_WIFI 变量 (这样做是为了维护平台的方便, 当然可以直接更改 BoardConfigCommon.mk 中变量):

```
# WIFI
BOARD_USR_WIFI := rtl8192cu
```

(2) SDIO wifi 配置:

➤ 配置文件: sys\_config1.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 mmc3 为例):

```
[mmc3_para]
sdc_used                = 1
sdc_detmode             = 4 ; sdio wifi 没有热插拔, 请配置检测模式为 4
bus_width               = 4
sdc_cmd                 = port:PI04<2><1><2><default>
sdc_clk                 = port:PI05<2><1><2><default>
sdc_d0                  = port:PI06<2><1><2><default>
sdc_d1                  = port:PI07<2><1><2><default>
sdc_d2                  = port:PI08<2><1><2><default>
sdc_d3                  = port:PI09<2><1><2><default>
sdc_det                 =
sdc_use_wp              = 0
sdc_wp                  =
;-----
;sdio wifi configuration
;wifi_sdc_id    ---  0- SDC0, 1- SDC1, 2- SDC2, 3- SDC3
;sdio_wifi_mod_sel    ---  0- none, 1- swl-n20(wifi), 2- usi bm-01(wifi+bt+fm),
;                      3- ar6302qfn, 4- apm6xxx, 5- swb b23(wifi+bt+fm)
;-----
[sdio_wifi_para]
sdio_wifi_used          =0
sdio_wifi_sdc_id        = 3      ; SDC3 用于 wifi 功能
```



sdio\_wifi\_mod\_sel = 1 ;选择 sdio wifi 模组的型号，根据需要打开下面相关配置项

```
; samsung swl-n20 sdio wifi gpio config
swl_n20_shdn = port:PH09<1><default><default><0>
swl_n20_host_wakeup = port:PH10<1><default><default><1>
swl_n20_vdd_en = port:PH11<1><default><default><0>
swl_n20_vcc_en = port:PH12<1><default><default><0>
```

```
; usi bm01a sdio wifi gpio config
;usi_bm01a_wl_pwr = port:PH12<1><default><default><0>
;usi_bm01a_wlbt_regon = port:PH11<1><default><default><0>
;usi_bm01a_wl_rst = port:PH10<1><default><default><0>
;usi_bm01a_wl_wake = port:PH09<1><default><default><0>
;usi_bm01a_bt_rst = port:PB05<1><default><default><0>
;usi_bm01a_bt_wake = port:PI20<1><default><default><0>
;usi_bm01a_bt_hostwake = port:PI21<0><default><default><0>
```

```
; swb-b23/usibm01 sdio wifi gpio config
;swbb23_wl_pwr = port:PH12<1><default><default><0>
;swbb23_wl_shdn = port:PH09<1><default><default><0>
;swbb23_wl_wake = port:PB10<1><default><default><0>
;swbb23_bt_shdn = port:PB05<1><default><default><0>
;swbb23_bt_wake = port:PI20<1><default><default><0>
;swbb23_bt_hostwake = port:PI21<0><default><default><0>
```

```
; apm sdio wifi gpio config
;apm_6981_vcc_en = port:PA09<1><default><default><0>
;apm_6981_vdd_en = port:PA10<1><default><default><0>
;apm_6981_wakeup = port:PA11<1><default><default><0>
;apm_6981_rst_n = port:PA12<1><default><default><0>
;apm_6981_pwd_n = port:PA13<1><default><default><0>
```

mmc3\_para 部分配置 sdio 通信 data 及 clk 信号，一般不改动，sdio\_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：

如上一部分所述，默认在 android4.0\device\softwinner\crane-common\BoardConfigCommon.mk 文件中已将 BOARD\_USR\_WIFI 设置为 nanowifi，所以不需在 android4.0\device\softwinner\crane\_xx\BoardConfig.mk 修改；

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。

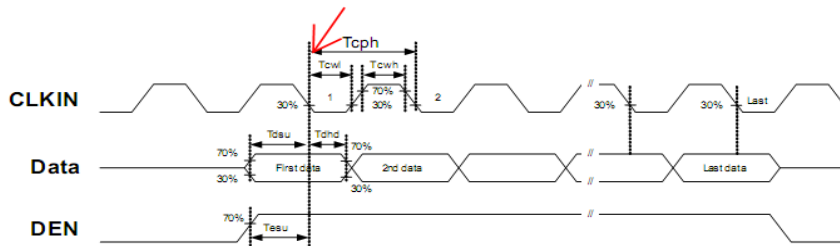
## 1.7 LCD Panel 修改

### 1) 基本的配置及调整

调试 lcd panel 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）

Item	Symbol	Min.	Typ.	Max.	Unit	Note
DCLK cycle time	Tclk	25			ns	
DCLK frequency	fclk		33	40	MHz	
DCLK pulse duty	Tcwh	40	50	60	%	
VSYNC setup time	Tvst	8			ns	
VSYNC hold time	Tvhd	8			ns	
HSYNC setup time	Thst	8			ns	
HSYNC hold time	Thhd	8			ns	
Data setup time	Tdasu	8			ns	
Data hold time	Tdahd	8			ns	
DE setup time	Tdesu	8			ns	
DE hold time	Tdehd	8			ns	
Horizontal display area	Thd		800		Tcph	
HSYNC period time	Th		928		Tcph	
HSYNC width	Thwh	1	48		Tcph	
HSYNC back porch	Thbp		40		Tcph	
HSYNC front porch	Thfp		40		Tcph	
Vertical display area	Tvd		480		th	
VSYNC period time	Tv		525		th	
VSYNC width	Tvwh		3		th	
VSYNC back porch	Tvbp		29		th	
VSYNC front porch	Tvfp		13		th	



```

;-----
;lcd0 configuration
;lcd_dclk_freq:      in MHZ unit
;lcd_pwm_freq:      in HZ unit
;lcd_if:             0:lv(sync+de); 1:8080; 2:ttl; 3:lvds
;lcd_lvds_ch:        0:single channel; 1:dual channel
;lcd_lvds_mode:      0:NS mode; 1:JEIDA mode
;lcd_lvds_bitwidth:  0:24bit; 1:18bit
;lcd_lvds_io_cross:  0:normal; 1:pn cross
;lcd_cpu_if:         0:18bit; 1:16bit mode0; 2:16bit mode1; 3:16bit mode2; 4:16bit mode3; 5:9bit;
6:8bit 256K; 7:8bit 65K
;lcd_io_cfg0:        0x10000000 - MAGIC_VER_A, 0x04000000 - MAGIC_VER_B
;-----
[lcd0_para]
lcd_used              = 1

```



```
lcd_x                = 800
lcd_y                = 480
lcd_dclk_freq        = 33                ;dclk 频率 (MHz)
lcd_pwm_freq         = 12000            ;pwm 频率 (Hz)
lcd_pwm_pol          = 1                ;该参数用于纠正亮度调节反向的问题
lcd_srgb             = 0x00202020
lcd_swap             = 0
lcd_if               = 0
lcd_hbp              = 40                ;hbp
lcd_ht               = 928
lcd_vbp              = 29                ;vbp
lcd_vt               = 1050            ;(525*2)
lcd_hv_if            = 0                ;屏接口, 0: hv, 3: lvds
lcd_hv_smode         = 0
lcd_hv_s888_if       = 0
lcd_hv_syuv_if       = 0
lcd_hv_vspw          = 0
lcd_hv_hspw          = 0
lcd_hv_lde_used      = 0
lcd_hv_lde_iovalue   = 0
lcd_ttl_stvh         = 0
lcd_ttl_stvdl        = 0
lcd_ttl_stvdp        = 0
lcd_ttl_ckvt         = 0
lcd_ttl_ckvh         = 0
lcd_ttl_ckvd         = 0
lcd_ttl_oevt         = 0
lcd_ttl_oevh         = 0
lcd_ttl_oevd         = 0
lcd_ttl_sthh         = 0
lcd_ttl_sthd         = 0
lcd_ttl_oehh         = 0
lcd_ttl_oehd         = 0
lcd_ttl_revd         = 0
lcd_ttl_datarate     = 0
lcd_ttl_revsel       = 0
lcd_ttl_datainv_en   = 0
lcd_ttl_datainv_sel  = 0
lcd_lvds_ch          = 0
lcd_lvds_mode        = 0
lcd_lvds_bitwidth    = 0
```



```

lcd_lvds_io_cross      = 0
lcd_cpu_if            = 0
lcd_cpu_da           = 0
lcd_frm              = 0
lcd_io_cfg0          = 0x10000000      ;clk 上升沿有效或下降沿有效
lcd_io_cfg1          = 0
lcd_io_strength      = 0

```

;背光控制，电路图如果没有 **bl enable gpio**，请设置为 0；否则请正确设置 **gpio** 及控制逻辑（0：低电平打开，1：高电平打开）；

```

lcd_bl_en_used       = 1
lcd_bl_en            = port:PH07<1><0><default><1>

```

;lcd power 控制，请正确设置 **gpio** 及控制逻辑（0：低电平打开，1：高电平打开）；

```

lcd_power_used       = 1
lcd_power            = port:PH08<1><0><default><1>

```

;lcd pwm 控制，根据原理图配置使用的哪组 **pwm**，一般不需修改；

```

lcd_pwm_used         = 1
lcd_pwm              = port:PB02<2><default><default><default>

```

```

lcd_gpio_0           =
lcd_gpio_1           =
lcd_gpio_2           =
lcd_gpio_3           =

```

```

lcdd0                = port:PD00<2><default><default><default>
lcdd1                = port:PD01<2><default><default><default>
lcdd2                = port:PD02<2><default><default><default>
lcdd3                = port:PD03<2><default><default><default>
lcdd4                = port:PD04<2><default><default><default>
lcdd5                = port:PD05<2><default><default><default>
lcdd6                = port:PD06<2><default><default><default>
lcdd7                = port:PD07<2><default><default><default>
lcdd8                = port:PD08<2><default><default><default>
lcdd9                = port:PD09<2><default><default><default>
lcdd10               = port:PD10<2><default><default><default>
lcdd11               = port:PD11<2><default><default><default>
lcdd12               = port:PD12<2><default><default><default>
lcdd13               = port:PD13<2><default><default><default>
lcdd14               = port:PD14<2><default><default><default>
lcdd15               = port:PD15<2><default><default><default>

```



```

lcdd16          = port:PD16<2><default><default><default>
lcdd17          = port:PD17<2><default><default><default>
lcdd18          = port:PD18<2><default><default><default>
lcdd19          = port:PD19<2><default><default><default>
lcdd20          = port:PD20<2><default><default><default>
lcdd21          = port:PD21<2><default><default><default>
lcdd22          = port:PD22<2><default><default><default>
lcdd23          = port:PD23<2><default><default><default>
lcdclk          = port:PD24<2><default><default><default>
lcdde           = port:PD25<2><default><default><default>
lcdhsync        = port:PD26<2><default><default><default>
lcdvsync        = port:PD27<2><default><default><default>

```

### (2) 屏驱动扩展及调试

关于屏的移植方法及驱动程序基本工作原理请查看提供的《A10 lcd 用户配置手册》文档；

## 1.8 Touch Panel 配置

### (1) 支持的TP模组

发布的SDK 中，有对电阻屏（两点）、FT5206/FT5406/FT5306/FT5606（敦泰）、GT801/GT813/GT827/GT828等的支持，详细请查看各个平台上面的TP支持列表，下表列出一些CTP模组使用时的驱动名称以及I2C地址；

CTP模组名称	驱动名称	CTP 设备名称	I2C 地址	IDC文件名称
FT5206/FT5306/FT5406/FT5606	ft5x_ts.ko	ft5x_ts	0x38	ft5x_ts.idc
GT801	Goodix_touch.ko	Goodix-TS	0x55	gt80x.idc
GT801二合一方案	gt801_2plus1.ko	Goodix-TS	0x55	gt80x.idc
GT813/GT827/GT828	gt82x.ko	Goodix-TS	0x5d	gt80x.idc
GT818	gt818_ts.ko	gt818_ts	0x5d	gt818_ts.idc
矽创	sitronix_st15xx_ts.ko	sitronix_ts	0x55	sitronix_ts.idc
电阻屏	sun4i-ts.ko			sun4i-ts.idc

### (2) 打包配置文件修改: sys\_config1.fex

电阻屏:

```

;-----
;rtp_screen_size: 表屏幕尺寸，以斜对角方向长度为准，以寸为单位
;tp_regidity_level: 表屏幕的硬度，以指覆按压，抬起时开始计时，多少个10ms 时间单位之后，硬件采集不到数据为准；
;通常，我们建议的屏，5寸屏设为5,7寸屏设为7，
;对于某些供应商提供的屏，硬度可能不合要求，需要适度调整
;rtp_press_threshold_enable: whether press threshold is enable or not. 0 is suggested.
;rtp_press_threshold : only when rtp_press_threshold_enable == 1, this para take effect.

```





between 0 and 0xFFFFFFFF is supported.more bigger, more sensitive,  
0x1f40 is suggested.

;rtp\_sensitive\_level : between 0 and 0xf is supported.more bigger, more sensitive, 0xf is  
suggested.

;rtp\_exchange\_x\_y\_flag : in some case, x, y need to be exchanged, at that situation, 1 is setted.  
normally, 0 is setted.

;------

[rtp\_para]

rtp\_used = 1

rtp\_screen\_size = 7 ;Touch panel 的尺寸

rtp\_regidity\_level = 7

rtp\_press\_threshold\_enable = 0

rtp\_press\_threshold = 0x1f40

rtp\_sensitive\_level = 0xf

rtp\_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:

**电容屏:**

;------

;capacitor tp configuration

;external int function

;wakeup output function

;notice --- tp\_int\_port & tp\_io\_port use the same port

;name should be compatible with the corresponding ctp driver.

;------

[ctp\_para]

ctp\_used =1

ctp\_name = "ft5x\_ts" ;驱动中设置的模组名称, 根据具体的模组进行配置

ctp\_twi\_id = 2 ;使用哪组I2C

ctp\_twi\_addr =0x38 ;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:PH21<6><default>

;ctp\_wakeup = port:PB13<1><default><default><1>

ctp\_reset = port:PH20<1><default>

ctp\_io\_port = port:PH21<0><default>

Touch panel 的GPIO 请正确配置中断引脚、wakeup pin 和reset pin, 一般情况下, wakeup 和  
reset pin

只需留一个就可以, 一般用reset, 此时请把wakeup 注释掉, 可以减少驱动中suspend 和  
resume 时耗费的



时间;

### (3) 驱动的拷贝

编译过程中使用的命令: `$extract-bsp` , 将把驱动拷贝到指定的目录中备用。

编译后的目录: `out\ target\product\...\system\vendor\modules`, 省略的部分为lunch时选择选择的配置文件夹名称。

机器中对应的目录目录为: `/system/vendor/modules`

### (4) Android 层驱动的加载:

根据使用的TP模组进行相应驱动的加载, 不使用的模组则注释掉或者是删除掉。加载驱动的目录文件为: `android4.0\device\softwinner\...\init.XXX.rc` , 省略的部分为lunch时选择的配置文件夹名称。

```
#insmod tp driver
```

```
#insmod /drv/sun4i-ts.ko
```

```
insmod /drv/ft5x_ts.ko
```

```
#insmod /drv/ goodix_touch.ko
```

### (5) idc文件

Android4.0中需要使用idc文件来识别是否为触摸屏, 点idc的源文件存放目录 :

`android4.0\device\softwinner\...` , 省略的部分为lunch时选择的配置文件夹名称。

机器里面的存放目录为`system/usr/idc` 。

点idc文件以注册的设备名称命名。如果不清楚模组使用的idc文件名称, 可以进入`adb shell`之后使用`getevent`命令查看。

在`android4.0\device\softwinner\...\ crane_xx.mk`或者是`nuclear_xx.mk`文件中, 省略的部分为lunch时选择的配置文件夹名称。修改为相应模组使用的idc文件, 以`ft5x_ts.ko`为例:

```
#input device config
```

```
PRODUCT_COPY_FILES += \
```

```
device/softwinner/crane-evb-v13/sun4i-keyboard.kl:system/usr/keylayout/sun4i-keyboard.kl \
```

```
device/softwinner/crane-evb-v13/ft5x_ts.idc:system/usr/idc/ft5x_ts.idc
```

### (6) CTP 驱动移植及调试:

该部分请参考提供的《`ctp` 调试文档》及相关文档;

## 1.9 G—Sensor 配置

### (1) 支持的GSENSOR模组

发布的SDK 中已添加了对`DMDRD06`、`MMA7660`、`MXC6225` 及`BMA250 G-Sensor` 的支持, 需要客户根据需要做如下配置:

### (2) 驱动的拷贝

编译过程中使用的命令: `$extract-bsp` , 将把驱动拷贝到指定的目录中备用。

编译后的目录: `out\ target\product\...\system\vendor\modules`, 省略的部分为lunch时选择选择的配置文件夹名称。



机器中对应的目录目录为： /system/vendor/modules

### (3) 打包配置文件修改： sys\_config1.fex

```
-----  
; G sensor configuration  
;("mx622x", 0x15) or ("mma7660", 0x4C) or ("bma250", 0x18) or (dmard06,0x1c)//参考设备名及I2C  
地址  
; gs_twi_id --- TWI ID for controlling Gsensor (0: TWI0, 1: TWI1, 2: TWI2)  
-----  
[gsensor_para]  
gsensor_used = 1 //标示使用gsensor  
//标示使用mma7660, 可为"mx622x"、"mma7660"、"bma250"  
gsenser_name = "mma7660"  
gsensor_twi_id = 1 //使用哪组i2c  
gsensor_twi_addr = 0x4c //i2c 设备地址  
gsensor_int1 = port:PH00<6><1><default><default> //中断引脚, 预留  
gsensor_int2 =  
根据实际使用的gsensor 来做配置, mx6225、mma7660、dmard06 及bma250 设备名及I2C 地  
址如上注释所示;
```

### (4) 驱动的加载

根据使用的Gsensor模组进行相应驱动的加载, 不使用的模组则注释掉或者删掉。加载驱动的目录文件为: android4.0\device\softwinner\..\init.XXX.rc ,省略的部分为lunch时选择的配置文件夹名称。

如下:

```
使用bma250时: insmod /system/vendor/modules/bma250.ko  
使用mma7660时: insmod /system/vendor/modules/mma7660.ko  
使用dmard06时: insmod /system/vendor/modules/dmard06.ko  
使用mx6225时: insmod /system/vendor/mecs.ko  
insmod /system/vendor/mx622x.ko  
mkdir /data/misc/sensors 0777 system system  
service memsicc /system/bin/logwrapper /system/bin/memsicc
```

### (5) 方向的调整

使用gsensor时, 发现方向反了, 需要进行方向的调整, 需要修改的文件为:

android4.0\device\softwinner\..\BoardConfig.mk, 省略的部分为lunch时选择的配置文件夹名

```
# Gsensor board config  
# we can use this string : mma7660, mx622x, bma250, dmard06  
BOARD_USES_GSENSOR_TYPE := mma7660 //标示用mma7660 gsensor, 根据使用的模组进行选择,  
如注释所示  
BOARD_GSENSOR_DIRECT_X := true //如果x 轴反向, 则置false
```



```
BOARD_GSENSOR_DIRECT_Y := true //如果y 轴反向，则置false
BOARD_GSENSOR_DIRECT_Z := false //如果z 轴反向，则置false
BOARD_GSENSOR_XY_REVERT := true//如果x 轴当y 轴用，y 轴当x 轴，则置true
```

### (6) 修改相关设备节点

在android4.0\device\softwinner\..\ueventd.XXX.rc，省略的部分为lunch时选择的配置文件夹名.

```
bma250 ,mma7660: /sys/devices/virtual/input/input* name 0660 root input
/sys/devices/virtual/input/input* max 0660 root input
/sys/devices/virtual/input/input* min 0660 root input
/sys/devices/virtual/input/input* poll 0666 root input
dmard06: /dev/dmard06 0777 system system
mxc6225:/dev/mxc622x 0777 system system
/dev/ecompass_ctrl 0777 system system
```

## 1.10 Camera 的配置

- (1) 发布的 SDK 中已添加了对 SP0838、GC0308、OV7670、GT2005、HI704 和 MT9M112 的支持;
- (2) 打包配置文件修改: sys\_config1.fex

```

;-----
;csi gpio configuration
;csi_if: 0:hv_8bit 1:hv_16bit 2:hv_24bit 3:bt656 1ch 4:bt656 2ch 5:bt656 4ch
;csi_mode: 0:sample one csi to one buffer 1:sample two csi to one buffer
;csi_dev_qty: The quantity of devices linked to csi interface
;csi_vflip: flip in vertical direction 0:disable 1:enable
;csi_hflip: flip in horizontal direction 0:disable 1:enable
;csi_stby_mode: 0:not shut down power at standby 1:shut down power at standby
;csi_iovdd: camera module io power , pmu power supply
;csi_avdd: camera module analog power , pmu power supply
;csi_dvdd: camera module core power , pmu power supply
;pmu_lldo3: fill "axp20_pll"
;pmu_lldo4: fill "axp20_hdmi"
;fill "" when not using any pmu power supply
;csi_flash_pol: the active polarity of the flash light IO 0:low active 1:high active
; ( "sp0838" ,0x30) ("gt2005", 0x78) ("gc0308", 0x42) ("mt9m112", 0xba)
; ("hi704",0x60) ("mt9m113", 0x78) ("ov2655", 0x60)
;-----
[csi0_para]
csi_used = 1
csi_mode = 0
csi_dev_qty = 1 ;配置 csi0 接口上挂载 camera 的个数

```



```
csi_stby_mode          = 0          ;0:standby 时不关电源， 1: 关电源

csi_mname              = "gc0308"   ;设备名
csi_twi_id             = 1          ;使用哪组 I2C
csi_twi_addr          = 0x42        ;设备 I2C 地址
csi_if                = 0
csi_vflip             = 0
csi_hflip             = 1
csi_iovdd             = ""
csi_avdd              = ""
csi_dvdd              = ""
csi_flash_pol         = 1

csi_mname_b           = ""          ;设备名
csi_twi_id_b          = 1          ;使用哪组 I2C
csi_twi_addr_b       = 0x78        ;设备 I2C 地址
csi_if_b              = 0
csi_vflip_b           = 1
csi_hflip_b           = 0
csi_iovdd_b           = ""
csi_avdd_b            = ""
csi_dvdd_b            = ""
csi_flash_pol_b      = 1

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_d8                =
csi_d9                =
csi_d10               =
csi_d11               =
csi_d12               =
csi_d13               =
```



```

csi_d14           =
csi_d15           =
csi_reset         = port:PH13<1><default><default><0>
csi_power_en      =
csi_stby          = port:PH16<1><default><default><0>
csi_flash         =                               ;闪光灯的 gpio 配置
csi_af_en         =
csi_reset_b       =
csi_power_en_b    =
csi_stby_b        =
csi_flash_b       =                               ;闪光灯电平控制逻辑
csi_af_en_b       =

```

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；

该部分配置详细说明如下：

配置项	配置项含义
csi_used =xx	是否使用 csi0
csi_twi_id =xx	csi0 使用的 IIC
csi_mname=xx	csi0 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 ov7670, gc0308, gt2005, hi704,sp0338,mt9m112 可选
csi_twi_addr=xx	csi0 使用的模组的 IIC 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 standby io
csi_iovdd	配置 csi io 电源来源： “axp20_pll”: 表示电源来自 pmu 的 ldo3 “axp20_hdmi”: 表示电源来自 pmu 的 ldo4 “”: 表示电源来自非 pmu 的 ldo 一般来说，EVB 配置成“” 多数方案，配置成“axp20_pll”
csi_avdd	配置 csi avdd 电源来源： “axp20_pll”: 表示电源来自 pmu 的 ldo3 “axp20_hdmi”: 表示电源来自 pmu 的 ldo4 “”: 表示电源来自非 pmu 的 ldo 一般来说，EVB 配置成“” 多数方案，配置成“axp20_pll”
csi_dvdd	配置 csi dvdd 电源来源： “axp20_pll”: 表示电源来自 pmu 的 ldo3 “axp20_hdmi”: 表示电源来自 pmu 的 ldo4 “”: 表示电源来自非 pmu 的 ldo 一般来说，EVB 配置成“” 多数方案，配置成“”
csi_pck=xx	模组送给 csi0 的 clock GPIO 配置
csi_ck=xx	csi0 送给模组的 clock GPIO 配置
csi_hsync=xx	模组送给 csi0 的行同步信号 GPIO 配置
csi_vsync=xx	模组送给 csi0 的帧同步信号 GPIO 配置
csi_d0=xx ... csi_d15=xx	模组送给 csi0 的 8bit/16bit 数据 GPIO 配置
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）



<b>csi_reset_b=xx</b>	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）
<b>csi_power_en_b=xx</b>	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 csi_stby_mode 配置成 1，则 csi_power_en 的默认值一般配置成 0。
<b>csi_stby_b=xx</b>	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) Android 层的配置:

➤ 设备相关驱动模块:

相关驱动模块为 videobuf-core.ko、videobuf-dma-contig.ko、sun4i\_csi0.ko、sun4i\_csi1.ko 及 sp0838.ko/ gc0308.ko/ gt2005.ko/ mt9m112.ko,

➤ Android:

在 android4.0\device\softwinner\crane-xx\init.sun4i.rc 文件中加入装载驱动模块的语句:

单摄像头:

#csi module

```
insmod /system/vendor/modules/videobuf-core.ko
insmod /system/vendor/modules/videobuf-dma-contig.ko
insmod /system/vendor/modules/mt9m112.ko
insmod /system/vendor/modules/sun4i_csi0.ko
```

双摄像头:

#csi module

```
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/gt2005.ko
insmod /system/vendor/modules/sun4i_csi1.ko
insmod /system/vendor/modules/sun4i_csi0.ko
```

若一个 CSI 接口上挂载两个 camera，则只需要 insmod 该 csi 模块即可;

在 android4.0\device\softwinner\crane-xx\ueventd.sun4i.rc 文件中改变相关设备节点的权限:

```
/dev/video0          0777  system  system
/dev/video1          0777  system  system (双摄像头时加入)
```

(4) Camera 参数配置





➤ camera.cfg

路径: android4.0\device\softwinner\crane-xxx\ camera.cfg

内容简介:

```
-----  
; 1 for single camera, 2 for double camera  
-----  
number_of_camera = 1 ;camera device 的个数，如果有两个 camera device，则以下参数需要  
                        写两份，分别对应不同的设备参数  
-----  
; CAMERA_FACING_BACK  
; gt2005  
-----  
camera_id = 0      ;camera id  
-----  
; 1 for CAMERA_FACING_FRONT  
; 0 for CAMERA_FACING_BACK  
-----  
camera_facing = 0  ;标志是该 camera 是前置还是后置  
-----  
; driver device name  
-----  
camera_device = /dev/video0 ;标志该 camera 是挂载的 csi0 接口还是 csi1 (/dev/video1) 接口  
-----  
; device id  
; for two camera devices with one CSI  
-----  
device_id = 0      ;当两个 camera device 公用一个 CSI Interface 时的 device id 号  
                    ;device_id 需要和 sysconfig1.fex 中的配置项对应；  
                    ;device_id =0 对应 csi_mname 代表的 camera 设备  
                    ;device_id=1 对应 csi_mname_b 代表的 camera 设备；  
  
used_preview_size = 1//是否支持预览，及预览像素大小  
key_support_preview_size = 1600x1200,640x480  
key_default_preview_size = 640x480  
  
used_picture_size = 1//是否支持照相，及像素大小  
key_support_picture_size =  
2592x1936,2048x1536,1600x1200,1280x1024,1024x768,800x600,640x480  
key_default_picture_size = 1600x1200  
  
used_flash_mode = 0//闪光配置，目前只支持 on、off 属性  
key_support_flash_mode = on,off
```



key\_default\_flash\_mode = off

used\_color\_effect=1//色彩效果

key\_support\_color\_effect = none,mono,negative,sepia,aqua

key\_default\_color\_effect = none

used\_frame\_rate = 1//帧率

key\_support\_frame\_rate = 15, 30

key\_default\_frame\_rate = 30

used\_focus\_mode = 0//自动对焦

key\_support\_focus\_mode = auto,infinity,macro,fixed

key\_default\_focus\_mode = auto

used\_scene\_mode = 0//场景模式

key\_support\_scene\_mode =

auto,auto,portrait,landscape,night,night-portrait,theatre,beach,snow,sunset,steadyphoto,fireworks,sports,party,candlelight,barcode

key\_default\_scene\_mode = auto

used\_white\_balance = 1//白平衡

key\_support\_white\_balance =

auto,incandescent,fluorescent,warm-fluorescent,daylight,cloudy-daylight

key\_default\_white\_balance = auto

used\_exposure\_compensation = 1//曝光补偿

key\_max\_exposure\_compensation = 4

key\_min\_exposure\_compensation = -4

key\_step\_exposure\_compensation = 1

key\_default\_exposure\_compensation = 0

; only for facing back camera

used\_zoom = 1//是否支持 zoom 功能，仅后置摄像头支持该功能

key\_zoom\_supported = true

key\_smooth\_zoom\_supported = false

key\_zoom\_ratios = 100,120,150,200,230,250,300

key\_max\_zoom = 6

key\_default\_zoom = 0

➤ media\_profiles.xml

路径: android4.0\device\softwinner\crane-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>
```

```
<!-- Each camcorder profile defines a set of predefined configuration parameters -->
```

```
<!-- Back Camera -->
```

```
<CamcorderProfiles cameraId="0">
```

```
<EncoderProfile quality="low" fileFormat="mp4" duration="30">
```

```
<Video codec="h264"
```

```
    bitRate="1000000"
```

```
    width="640"
```

```
    height="480"
```

```
    frameRate="30" />
```

```
<Audio codec="aac"
```

```
    bitRate="12200"
```

```
    sampleRate="8000"
```

```
    channels="1" />
```

```
</EncoderProfile>
```

```
<EncoderProfile quality="high" fileFormat="mp4" duration="30">
```

```
<Video codec="h264"
```

```
    bitRate="3000000"
```

```
    width="1600"
```

```
    height="1200"
```

```
    frameRate="15" />
```

```
<Audio codec="aac"
```

```
    bitRate="96000"
```

```
    sampleRate="44100"
```

```
    channels="1" />
```

```
</EncoderProfile>
```

```
<ImageEncoding quality="90" />
```

```
<ImageEncoding quality="80" />
```

```
<ImageEncoding quality="70" />
```

```
<ImageDecoding memCap="20000000" />
```

```
</CamcorderProfiles>
```

```
<!-- Front Camera -->
```

```
<CamcorderProfiles cameraId="1">
```

```
<EncoderProfile quality="high" fileFormat="mp4" duration="60">
```

```
<Video codec="h264"
```

```
    bitRate="1000000"
```

```
    width="640"
```

```
    height="480"
```

```
    frameRate="25" />
```



```
<Audio codec="aac"
    bitRate="12200"
    sampleRate="44100"
    channels="1" />
</EncoderProfile>
<EncoderProfile quality="low" fileFormat="mp4" duration="30">
    <Video codec="h264"
        bitRate="500000"
        width="320"
        height="240"
        frameRate="25" />
    <Audio codec="aac"
        bitRate="12200"
        sampleRate="8000"
        channels="1" />
</EncoderProfile>
<ImageEncoding quality="90" />
<ImageEncoding quality="80" />
<ImageEncoding quality="70" />
<ImageDecoding memCap="20000000" />
</CamcorderProfiles>
<EncoderOutputFileFormat name="mp4" />
<!--
    If a codec is not enabled, it is invisible to the applications
    In other words, the applications won't be able to use the codec
    or query the capabilities of the codec at all if it is disabled
-->
<VideoEncoderCap name="h264" enabled="true"
    minBitRate="64000" maxBitRate="8000000"
    minFrameWidth="320" maxFrameWidth="1600"
    minFrameHeight="240" maxFrameHeight="1200"
    minFrameRate="1" maxFrameRate="30" />

<AudioEncoderCap name="aac" enabled="true"
    minBitRate="8000" maxBitRate="320000"
    minSampleRate="8000" maxSampleRate="48000"
    minChannels="1" maxChannels="1" />

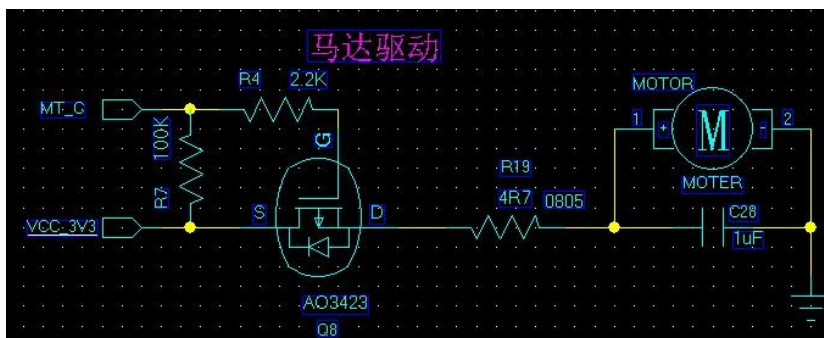
<VideoDecoderCap name="wmv" enabled="false"/>
<AudioDecoderCap name="wma" enabled="false"/>
</MediaSettings>
```



## 1.11 震动马达的配置

### (1) 震动马达典型电路:

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



### (2) Sys\_config1 文件配置:

```

;-----
;motor configuration
;-----

```

[motor\_para]

motor\_used = 1 ; 是否启用马达，启用置 1，反之置 0

motor\_shake = port:PB03<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) 驱动及 Android 的配置

在 android4.0\device\softwinner\crane-xx\init.sun4i.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/sun4i-vibrator.ko
chmod 777 /sys/class/timed_output/sun4i-vibrator/enable

```

## 1.12 GPS 配置

(1)发布的 SDK 从 1.8 版本集成了 GPS 功能，目前只支持海威讯的 GPS，默认该功能没有打开，如果客户版中加入了 GPS 功能，请做如下配置：

### (2) 打包配置文件：sys\_config1.fex

```

;-----
; gps gpio configuration
; gps_spi_id --- the index of SPI controller. 0: SPI0, 1: SPI1, 2: SPI2, 15: no SPI used
; gps_spi_cs_num --- the chip select number of SPI controller. 0: SPI CS0, 1: SPI CS1
; gps_lradc --- the lradc number for GPS used. 0 and 1 is valid, set 2 if not use lradc
;-----

```



```
[gps_para]
gps_used           = 1           ;是否使用 GPS 功能
gps_spi_id        = 2           ;使用的 SPI 接口
gps_spi_cs_num    = 0
gps_lradc         = 1
gps_clk           = port:PI00<2><default><default><default>
gps_sign         = port:PI01<2><default><default><default>
gps_mag          = port:PI02<2><default><default><default>
gps_vcc_en       = port:PC22<1><default><default><0>
gps_osc_en       = port:PI14<1><default><default><0>
gps_rx_en        = port:PI15<1><default><default><0>
```

以上请根据硬件原理图来配置；

### (3) Android 层的配置：

1) 在 android4.0\device\softwinner\crane-common\BoardConfigCommon.mk 文件中有对 GPS 的选择：BOARD\_USES\_GPS\_TYPE 值当没有 GPS 模块时用 simulator 关键字，用海威讯 GPS 模块时填写 haiweixun；

(BOARD\_USES\_GPS\_TYPE 可以在 android2.3\device\softwinner\crane\_xx\ BoardConfig.mk 文件中重写，以便于客户版的维护)

```
#gps
#"simulator":target board does not have a gps hardware module;"haiweixun":use the gps
module offer by haiweixun
```

```
BOARD_USES_GPS_TYPE := haiweixun
```

2) 在 android4.0\device\softwinner\crane-xx\init.sun4i.rc 文件中加入装载驱动模块的语句：

```
#gps module
insmod /system/vendor/modules/gps.ko
```

3) 在 android4.0\device\softwinner\crane-xx\ueventd.sun4i.rc 文件中改变相关设备节点的权限：

```
/dev/gps    0777    system    system
```

### (4) GPS 功能测试：

调试 GPS 时建议使用 GPS\_Test\_Plus\_v1.1.3.apk，配合串口打印可以初步确认 GPS 是否正常工作。GPS 正常工作时，通常有以下打印：

```
I/dalvikvm( 383): Total arena pages for JIT: 11
I/GBCLogMsg:( 109): $GPVER,GBCV:1.0015,Nov 6 2011,19:12:37;DRIVER:1.0008,Nov 6
2011,20:34:41
I/GBCLogMsg:( 109): $GPVER,GBCV:1.0015,Nov 6 2011,19:12:37;DRIVER:1.0008,Nov 6
2011,20:34:41
I/GBCLogMsg:( 109): $GPVER,GBCV:1.0015,Nov 6 2011,19:12:37;DRIVER:1.0008,Nov 6
2011,20:34:41
```

如果没有上述打印，从两方面确认问题原因：

1) 通过 adb shell，查看驱动是否加载，如果没有加载，则手动加载再看看结果；



- 2) 检查硬件是否正确，GPS 与 IC 的连线是否正确，GPS 模组的晶振是否有焊上去等等；
- 3) 如果硬件引脚使用跟参考设计不一致，请修改 `sys_config1.fex` 中 GPS 使用的 GPIO 配置。

## 1.13 Boot 电流充电图标配置

- (1) 在 `sys_config1.fex` 配置文件中，`target` 下增加一项 `power_start`，如下所示：

```
[target]
power_start          = 0
```

当 `power_start` 存在，并且值是 0 的时候，下面的功能不起作用；

当 `power_start` 不存在，或存在但是值不是 0 的时候，下面的功能将生效：

∅ 在有电池的情况下，系统处于关机状态，如果插上外部电源，会显示电池图标，3 秒后自动关机；

∅ 在有电池的情况下，系统处于关机状态，如果长按开机按键，会正常启动；

∅ 在有电池并且有外部电源的情况下，系统处于关机状态，如果长按开机按键，会正常启动；

∅ 在没有电池的情况下，系统处于关机状态，如果插上外部电源，会显示电池图标，3 秒后正常启动；

∅ 在没有电池但是连接有外部电源的情况下，系统处于关机状态，如果长按开机按键，会正常启动；

- (2) 电池图标

电池图标是一个 `bmp` 图片，是位于 `\wboot\bootfs\os_show` 目录下的一个名为 `startup.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。

## 1.14 Recovery 功能使用

- (1) Recovery 功能说明：

Recovery 是 Android 的专用升级模式，用于对 android 自身进行更新；进入 recovery 模式的方法是，在 android 系统开机时，按住一个特定按键，则会自动进入 android 的 recovery 模式。

- (2) 键值的查看

按键是通过 AD 转换的原理制成。当用户按下某个按键的时候，会得到这个按键对应的 AD 转换的值。同时，所有的按键的键值都不相同，并且，键值之间都有一定的间隔，没有相邻。比如，键值可能是 5,10,15,20，但是不可能是 5,11,12,13。为了方便用户查看不同按键的键值，这种方法要求连接上串口使用，因此适合于开发阶段使用。具体步骤是：

- 1) 把小机把 PC 通过串口线连接起来，设置屏幕焦点在串口调试软件上；
- 2) 用户开机之前，按住 PC 键盘上的数字键“3”；
- 3) 开机，等待，1 秒后可以松开电脑键盘；

经过这样的步骤，用户会看到屏幕上有如下的打印信息出现：



```
welcome to key value test
press any key, and the value would be printed
press power key to
```

这表示系统已经进入了按键的键值测试模式，这种模式下将一直等待用户按下按键，并在串口屏幕上把按键的键值打印出。这样，用户可以很方便的查看不同按键的键值。比如，当按下某一个按键，用户可以看到如下的打印信息。

```
key value = 8
key value = 8
key value = 8
key value = 63
```

由于 AD 采用的速度非常快，所以同一个按键按下，屏幕上会出现多个值。用户可以看出，这个按键的键值是 8。最后出现的 63 是由于松开按键的时候的采用，是需要去掉的干扰数据。因此，用户查看按键键值的时候只要关注前面打印出的数值，后面出现的应该忽略不计。

### (3) 按键选择

通常情况下，一块方案板上的按键个数不同，或者排列不同，这都导致了方案商在选择作为开机阶段 recovery 功能的按键有所不同。因此，系统中提供了一种方法用于选择进入 recovery 模式的按键：

在 `efex\sys_config1.fex` 配置脚本中，提供了一项配置，用于选择按键的键值，如下所示：

```
[recovery_key]
key_max          = 24
key_min          = 17
```

它表示，所选择用于作为 recovery 功能的按键的键值范围落在 `key_min` 到 `key_max` 之间，即 17 到 24 之间。由于所有按键的选择都可以通过前面介绍的方法查看，因此，假设用户要选的按键是 a，用户这里选择配置的方法是：

- 1) 按照前面介绍的方法，读出所有按键的键值；
- 2) 读出 a 的键值 a1，同时取出两个相邻于 a 的键值，记为 b1 和 c1，b1 大于 c1；
- 3) 计算出  $(a1 + b1)/2$ ， $(a1 + c1)/2$ ，分别填写到 `key_max` 和 `key_min` 处；
- 4) 如果 a1 刚好是所有按键的最小值，则取 `key_min` 为 0；如果 a1 刚好是所有按键的最大值，则取 `key_max` 为 63；

经过以上的步骤，就可以选择一个特定的按键进入 recovery 模式。取了一个平均值的原因是考虑到长时间的使用，电阻的阻值可能会略有变化导致键值变化，取范围值就可以兼容这种阻值变化带来的键值变化。

### (4) 功能使用

在 android 编译完毕之后，使用 `make otapackage` 命令就可以在 `android4.0\out\target\product\crane-xxx` 目录下生成一个 `crane_xxx-ota-eng.username.zip` 文件，将该文件更名为 `update.zip` 后拷贝的 SD 卡中；

在系统启动时，按住设定的特定按键进入 recovery 模式，进入该模式后，系统会在 SD 卡目录下寻找 `update.zip` 文件，找到该文件就会进入自动升级，如没有找到该文件，则系统会提示报错；







## 2. Android 标准常用修改

### 2.1 预定义动态壁纸

预安装动态壁纸 apk

1.1 将想要预定义动态壁纸 apk 按照“预安装 apk”中的方法安装到系统中；  
配置系统启动后加载该壁纸

在 android4.0\frameworks\base\core\res\res\values\config.xml 文件中修改如下：

```
<!-- Component name of the default wallpaper. This will be ImageWallpaper if not  
specified -->
```

```
<!--<string name="default_wallpaper_component">@null</string-->
```

```
<string
```

```
name="default_wallpaper_component">fishnoodle.koipond_free/fishnoodle.koipond_free.WallpaperSer  
vice</string>
```

将@null 内容更改为该动态壁纸的 package name 和 Service Class name，中间用 ‘/’ 连接；

## Declaration

This **A10 android4.0 doc** is the original work and copyrighted property of Allwinner Technology (“Allwinner”). Reproduction in whole or in part must obtain the written approval of Allwinner and give clear acknowledgement to the copyright owner.

The information furnished by Allwinner is believed to be accurate and reliable. Allwinner reserves the right to make changes in circuit design and/or specifications at any time without notice. Allwinner does not assume any responsibility and liability for its use. Nor for any infringements of patents or other rights of the third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of Allwinner. This datasheet neither states nor implies warranty of any kind, including fitness for any particular application.