

# A64

## System Configuration 说明书

Confidential

# 文档履历

版本号	日期	制/修订人	制/修订记录
V1.0	2015-06-08	王伟	初始版本

Confidential

# 1. 目录

A64.....	1
目录.....	3
2. 系统.....	6
2.1. [product].....	6
2.2. [platform].....	6
2.3. [target].....	6
2.4. [key_detect_en].....	6
2.5. [power_sply].....	6
2.6. [card_boot].....	7
2.7. [pm_para].....	8
2.8. [card_boot0_para].....	8
2.9. [card_boot2_para].....	9
2.10. [twi_para].....	10
2.11. [uart_para].....	10
2.12. [jtag_para].....	10
2.13. [clock].....	10
3. SDRAM.....	11
3.1. [dram_para].....	11
4. 以太网.....	13
4.1. [gmac_para].....	13
5. I2C 总线.....	14
5.1. [twi0].....	14
5.2. [twi1].....	15
5.3. [twi2].....	15
6. 串口(UART).....	15
6.1. [uart0].....	15
6.2. [uart1].....	16
6.3. [uart2].....	16
6.4. [uart3].....	17
6.5. [uart4].....	17
7. SPI 总线.....	18
7.1. [spi0].....	18
7.2. [spi1].....	18
8. 电容屏.....	19
8.1. [ctp].....	19
8.2. [ctp_list].....	20
9. 闪存.....	20
9.1. [nand0_para].....	20
10. 显示.....	22
10.1. [disp].....	22

11. LCD 屏	23
11.1. [lcd0]	23
12. HDMI	25
12.1. [hdmi]	25
13. PWM	26
13.1. [pwm]	26
13.2. [pwm_suspend]	26
14. [boot_disp]	26
14.1. [boot_disp]	26
15. 摄像头(CSI)	27
15.1. [csi0]	27
15.2. [csi0/csi0_dev0]	28
15.3. [csi0/csi0_dev1]	29
16. SD / MMC	31
16.1. [sdc0]	31
16.2. [sdc1]	32
16.3. [sdc2]	33
17. USB	35
17.1. [usbc0]	35
17.2. [usbc1]	36
18. serial_feature	37
19. 重力感应(G Sensor)	37
19.1. [gsensor]	37
19.2. [gsensor_list]	38
20. WiFi	40
20.1. [wlan]	40
21. 蓝牙(Bluetooth)	40
21.1. [bt]	40
21.2. [btlpm]	41
22. Gyro scope sensor	41
22.1. [gyroscopesensor]	41
22.2. [gy_list]	42
23. 光感(light sensor)	42
23.1. [lightsensor]	42
23.2. [ls_list]	43
24. 罗盘 Compass	43
24.1. [compasssensor]	43
24.2. [ls_list]	44
25. 数字音频总线(spdif)	44
25.1. [spdif]、[sndspdif]	44
25.2. [sndspdif]、[sndhdmi]	45
25.3. [daudio0]	45
26. 内置音频 codec	47
26.1. [sndcodec]	47

26.2. [codec].....	47
26.3. [i2s].....	48
27. 红外（IR）.....	49
27.1. [s_cir0].....	49
28. PMU 电源.....	49
28.1. [pmu0].....	49
28.2. [pmu0_regu].....	55
29. Recovery 键配置.....	55
30. DVFS.....	56
30.1. CPU DVFS.....	56
31. Pinctrl 测试.....	58
32. [s_uart0].....	58
33. [s_rsb0].....	58
34. [s_jtag0].....	59
35. mali400.....	59
36. Declaration.....	59

sonfidentia

## 2. 系统

### 2.1. [product]

配置项	配置项含义
Version = "100"	配置的版本号
machine = "evb"	方案名字

配置举例：

```
[product]
version = "100"
machine = "evb"
```

### 2.2. [platform]

配置项	配置项含义
eraseflag=1	量产时是否擦除。0：不擦，1：擦除（仅对量产工具，升级工具无效）

配置举例：

```
[platform]
eraseflag = 1
```

### 2.3. [target]

配置项	配置项含义
boot_clock=xx	启动频率(A7启动频率); xx 表示多少MHZ
storage_type = -1	启动介质选择 0 : nand, 1: card0,2: card2,-1 (default) 自动扫描启动介质。

配置举例：

```
[target]
boot_clock = 1008
storage_type = -1
```

### 2.4. [key\_detect\_en]

配置项	配置项含义
keyen_flag = 1	当 keyen_flag = 1 时，支持按键检测； 当 keyen_flag = 0 时，不支持按键检测

### 2.5. [power\_sply]

配置项	配置项含义
dc1_vol = 1003100	dc1 的输出电压，mV

dcdc2_vol = 1001100	dcdc2 的输出电压, mV,
dcdc6_vol = 1001100	dcdc6 的输出电压, mV,
aldo1_vol = 2800	aldo1 的输出电压, mV,
aldo2_vol = 1001800	aldo2 的输出电压, mV,
aldo3_vol = 1003000	aldo3 的输出电压, mV,
dldo1_vol = 3300	dldo1 的输出电压, mV,
dldo2_vol = 3300	dldo2 的输出电压, mV,
dldo3_vol = 2800	dldo3 的输出电压, mV,
dldo4_vol = 1003300	dldo4 的输出电压, mV,
eldo1_vol = 1001800	eldo1 的输出电压, mV,
eldo2_vol = 1800	eldo2 的输出电压, mV,
eldo3_vol = 1800	eldo3 的输出电压, mV,
fldo2_vol = 1001100	fldo2 的输出电压, mV,
gpio0_vol = 3100	gpio0 的输出电压, mV,

说明:

电压名称= 100XXXX : 表示把该路电压设置为 XXXX 指定的电压值, 同时打开输出开关

电压名称= 000XXXX : 表示把该路电压设置为 XXXX 指定的电压值, 同时关闭输出开关, 当有需要时由内核驱动打开

电压名称 = 0 : 表示关闭该路电压输出开关, 不修改原有的值

配置举例:

[power\_sply]

```

dcdc1_vol          = 1003100
dcdc2_vol          = 1001100
dcdc6_vol          = 1001100
aldo1_vol          = 2800
aldo2_vol          = 1001800
aldo3_vol          = 1003000
dldo1_vol          = 3300
dldo2_vol          = 3300
dldo3_vol          = 2800
dldo4_vol          = 1003300
eldo1_vol          = 1001800
eldo2_vol          = 1800
eldo3_vol          = 1800
fldo2_vol          = 1001100
gpio0_vol          = 3100

```

## 2.6.[card\_boot]

配置项	配置项含义
logical_start = 40960	启动卡逻辑起始扇区
sprite_gpio0 =	卡量产 gpio led 灯配置
next_work = 2	1-不做任何动作, 2-重启, 3-关机, 4-量产, 5-正常启动

举例配置:

```
[card_boot]
logical_start    = 40960
sprite_gpio0    =
next_work        = 2
```

## 2.7. [pm\_para]

配置项	配置项含义
standby_mode = x	if 1 == standby_mode, then support super standby; else, support normal standby.

配置举例:

```
[pm_para]
standby_mode    = 1
```

## 2.8. [card\_boot0\_para]

配置项	配置项含义
card_ctrl=0	卡量产相关的控制器选择 0
card_high_speed=xx	速度模式 0 为低速, 1 为高速
card_line=4	代表 4 线卡
sdc_d1=xx	sdc 卡数据 1 线信号的 GPIO 配置
sdc_d0=xx	sdc 卡数据 0 线信号的 GPIO 配置
sdc_clk=xx	sdc 卡时钟信号的 GPIO 配置
sdc_cmd=xx	sdc 命令信号的 GPIO 配置
sdc_d3=xx	sdc 卡数据 3 线信号的 GPIO 配置
sdc_d2=xx	sdc 卡数据 2 线信号的 GPIO 配置

配置举例:

```
[card0_boot_para]
card_ctrl      = 0
card_high_speed = 1
card_line      = 4
sdc_d1         = port:PF0<2><1><2><default>
sdc_d0         = port:PF1<2><1><2><default>
sdc_clk        = port:PF2<2><1><2><default>
sdc_cmd        = port:PF3<2><1><2><default>
sdc_d3         = port:PF4<2><1><2><default>
sdc_d2         = port:PF5<2><1><2><default>
```



## 2.9. [card\_boot2\_para]

配置项	配置项含义
sdc_io_1v8=1	表示 eMMC IO 电平是 1.8V
card_ctrl=2	卡启动控制器选择 2
card_high_speed=xx	速度模式 0 为低速, 1 为高速
card_line=4	4 线卡
sdc_cmd =xx	sdc 命令信号的 GPIO 配置
sdc_clk =xx	sdc 卡时钟信号的 GPIO 配置
sdc_d0 =xx	sdc 卡数据 0 线信号的 GPIO 配置
sdc_d1 =xx	sdc 卡数据 1 线信号的 GPIO 配置
sdc_d2=xx	sdc 卡数据 2 线信号的 GPIO 配置
sdc_d3=xx	sdc 卡数据 3 线信号的 GPIO 配置
sdc_d4=xx	sdc 卡数据 4 线信号的 GPIO 配置
sdc_d5=xx	sdc 卡数据 5 线信号的 GPIO 配置
sdc_d6=xx	sdc 卡数据 6 线信号的 GPIO 配置
sdc_d7=xx	sdc 卡数据 7 线信号的 GPIO 配置
sdc_emmc_rst	emmc 复位信号的 GPIO 配置
sdc_ex_dly_used	采样模式控制, 值等于 2 时, 要 tune 采样点, 其它值使用驱动内部默认的采样点配置

配置举例:

[card2\_boot\_para]

```

sdc_io_1v8      = 1
card_ctrl      = 2
card_high_speed = 1
card_line      = 8
sdc_ds         = port:PC1<3><1><3><default>
sdc_clk        = port:PC5<3><1><3><default>
sdc_cmd        = port:PC6<3><1><3><default>
sdc_d0         = port:PC8<3><1><3><default>
sdc_d1         = port:PC9<3><1><3><default>
sdc_d2         = port:PC10<3><1><3><default>
sdc_d3         = port:PC11<3><1><3><default>
sdc_d4         = port:PC12<3><1><3><default>
sdc_d5         = port:PC13<3><1><3><default>
sdc_d6         = port:PC14<3><1><3><default>
sdc_d7         = port:PC15<3><1><3><default>
sdc_emmc_rst   = port:PC16<3><1><3><default>
sdc_ex_dly_used = 2

```

## 2.10. [twi\_para]

配置项	配置项含义
twi_port= xx	Boot 的 twi 控制器编号
twi_scl=xx	Boot 的 twi 的时钟的 GPIO 配置
twi_sda=xx	Boot 的 twi 的数据的 GPIO 配置

配置举例:

[twi\_para]

```
twi_port          = 0
twi_scl           = port:PH0<2><default><default><default>
twi_sda           = port:PH1<2><default><default><default>
```

## 2.11. [uart\_para]

配置项	配置项含义
uart_debug_port=xx	Boot 串口控制器编号
uart_debug_tx=xx	Boot 串口发送的 GPIO 配置
uart_debug_rx=xx	Boot 串口接收的 GPIO 配置

配置举例:

[uart\_para]

```
uart_debug_port = 0
uart_debug_tx   = port:PB8<4><1><default><default>
uart_debug_rx   = port:PB9<4><1><default><default>
```

## 2.12. [jtag\_para]

配置项	配置项含义
jtag_enable=xx	JTAG 使能 1:使能 0: 不使能
jtag_ms=xx	测试模式选择输入(TMS) 的 GPIO 配置
jtag_ck=xx	测试时钟输入(TMS) 的 GPIO 配置
jtag_do=xx	测试数据输出(TDO) 的 GPIO 配置
jtag_di=xx	测试数据输入 (TDI) 的 GPIO 配置

配置举例:

[jtag\_para]

```
jtag_enable      = 1
jtag_ms          = port:PB0<4><default><default><default>
jtag_ck          = port:PB1<4><default><default><default>
jtag_do          = port:PB2<4><default><default><default>
jtag_di          = port:PB3<4><default><default><default>
```

## 2.13. [clock]

配置项	配置项含义
Pl14 =300	Ve 时钟频率

Pl16 =600	Peripherals 时钟频率
Pl18 =360	VEDIO1 时钟频率
Pl19 =297	GPU 时钟频率
Pl110=264	MIPI 时钟频率

配置举例:

[clock]

pll4               = 300  
 pll6               = 600  
 pll8               = 360  
 pll9               = 297  
 pll10             = 264

## 3. SDRAM

### 3.1. [dram\_para]

配置项	配置项含义
dram_clk =xx	DRAM 的时钟频率, 单位为 MHz; 它为 24 的整数倍, 最低不得低于 120,
dram_type =xx	DRAM 类型: 2 为 DDR2 3 为 DDR3
dram_zq=xx	DRAM 控制器内部参数, 由原厂来进行调节, 请勿修改
dram_odt_en=xx	ODT 是否需要使能 0: 不使能 1: 使能 一般情况下, 为了省电, 此项为 0
dram_para1=xx	DRAM 控制器内部参数, 由原厂来进行调节, 请勿修改
dram_para2 =xx	DRAM 控制器内部参数, 由原厂来进行调节, 请勿修改
dram_mr0 =xx	DRAM CAS 值, 可为 6,7,8,9; 具体需根据 DRAM 的规格书和速度来确定
dram_mr1 =xx	DRAM 控制器内部参数, 由原厂来进行调节, 请勿修改
dram_mr2 =xx	DRAM 控制器内部参数, 由原厂来进行调节, 请勿修改
dram_mr3 =xx	DRAM 控制器内部参数, 由原厂来进行调节, 请勿修改
dram_tpr0=xx	DRAM 控制器内部参数, 由原厂来进行调节, 请勿修改

dram_tpr1=xx	DRAM 控制器内部参数，由原厂来进行调节，请勿修改
dram_tpr2=xx	DRAM 控制器内部参数，由原厂来进行调节，请勿修改
dram_tpr3=xx	DRAM 控制器内部参数，由原厂来进行调节，请勿修改
dram_tpr4=xx	DRAM 控制器内部参数，由原厂来进行调节，请勿修改
dram_tpr5=xx	DRAM 控制器内部参数，由原厂来进行调节，请勿修改
dram_tpr6=xx	DRAM 控制器内部参数，由原厂来进行调节，请勿修改
dram_tpr7=xx	DRAM 控制器内部参数，由原厂来进行调节，请勿修改
dram_tpr8=xx	DRAM 控制器内部参数，由原厂来进行调节，请勿修改
dram_tpr9=xx	DRAM 控制器内部参数，由原厂来进行调节，请勿修改
dram_tpr10=xx	DRAM 控制器内部参数，由原厂来进行调节，请勿修改
dram_tpr11=xx	DRAM 控制器内部参数，由原厂来进行调节，请勿修改
dram_tpr12=xx	DRAM 控制器内部参数，由原厂来进行调节，请勿修改
dram_tpr13=xx	DRAM 控制器内部参数，由原厂来进行调节，请勿修改

配置举例：

[dram\_para]

dram\_clk = 672

dram\_type = 3

dram\_zq = 0x3b3bbb

dram\_odt\_en = 0x1

dram\_para1 = 0x10E410E4

dram\_para2 = 0x1000

dram\_mr0 = 0x1840

dram\_mr1 = 0x40

dram\_mr2 = 0x18

dram\_mr3 = 0x2

dram\_tpr0 = 0x004A2195

dram\_tpr1 = 0x02424190

dram\_tpr2 = 0x0008B060

dram\_tpr3 = 0x04b005dc

dram\_tpr4 = 0x0

dram\_tpr5 = 0x0  
 dram\_tpr6 = 0x0  
 dram\_tpr7 = 0x0  
 dram\_tpr8 = 0x0  
 dram\_tpr9 = 0x0  
 dram\_tpr10 = 0x8808  
 dram\_tpr11 = 0x20250000  
 dram\_tpr12 = 0x0  
 dram\_tpr13 = 0x04000800

## 4. 以太网

### 4.1. [gmac\_para]

配置项	配置项含义
gmac_used	gmac 使用控制: 1 使用, 0 不用
gmac_txd0	gmac tx GPIO 配置
gmac_txd1	gmac tx GPIO 配置
gmac_txd2	gmac tx GPIO 配置
gmac_txd3	gmac tx GPIO 配置
gmac_txd4	gmac tx GPIO 配置
gmac_txd5	gmac tx GPIO 配置
gmac_txd6	gmac tx GPIO 配置
gmac_txd7	gmac tx GPIO 配置
gmac_txclk	gmac tx clk GPIO 配置
gmac_txen	gmac tx enable GPIO 配置
gmac_gtxclk	gmac gtxclk GPIO 配置
gmac_rxd0	gmac rx GPIO 配置
gmac_rxd1	gmac rx GPIO 配置
gmac_rxd2	gmac rx GPIO 配置
gmac_rxd3	gmac rx GPIO 配置
gmac_rxd4	gmac rx GPIO 配置
gmac_rxd5	gmac rx GPIO 配置
gmac_rxd6	gmac rx GPIO 配置
gmac_rxd7	gmac rx GPIO 配置
gmac_rxdv	gmac rxdv GPIO 配置
gmac_txerr	gmac tx err GPIO 配置
gmac_rxerr	gmac rx err GPIO 配置
gmac_col	gmaccol GPIO 配置
gmac_crs	gmac crs GPIO 配置

gmac_clkin	gmac clkin GPIO 配置
gmac_mdc	gmac mdc GPIO 配置
gmac_mdio	gmac mdio GPIO 配置

配置举例:

```
[gmac_para]
gmac_used           = 0
gmac_txd0          = port:PA00<2><default><default><default>
gmac_txd1          = port:PA01<2><default><default><default>
gmac_txd2          = port:PA02<2><default><default><default>
gmac_txd3          = port:PA03<2><default><default><default>
gmac_txd4          = port:PA04<2><default><default><default>
gmac_txd5          = port:PA05<2><default><default><default>
gmac_txd6          = port:PA06<2><default><default><default>
gmac_txd7          = port:PA07<2><default><default><default>
gmac_txclk         = port:PA08<2><default><default><default>
gmac_txen          = port:PA09<2><default><default><default>
gmac_gtxclk        = port:PA10<2><default><default><default>
gmac_rxd0          = port:PA11<2><default><default><default>
gmac_rxd1          = port:PA12<2><default><default><default>
gmac_rxd2          = port:PA13<2><default><default><default>
gmac_rxd3          = port:PA14<2><default><default><default>
gmac_rxd4          = port:PA15<2><default><default><default>
gmac_rxd5          = port:PA16<2><default><default><default>
gmac_rxd6          = port:PA17<2><default><default><default>
gmac_rxd7          = port:PA18<2><default><default><default>
gmac_rxdv          = port:PA19<2><default><default><default>
gmac_rxclk        = port:PA20<2><default><default><default>
gmac_txerr         = port:PA21<2><default><default><default>
gmac_rxerr         = port:PA22<2><default><default><default>
gmac_col           = port:PA23<2><default><default><default>
gmac_crs           = port:PA24<2><default><default><default>
gmac_clkin         = port:PA25<2><default><default><default>
gmac_mdc           = port:PA26<2><default><default><default>
gmac_mdio          = port:PA27<2><default><default><default>
```

## 5. I2C 总线

主控有以下几个 I2C (twi) 控制器

### 5.1. [twi0]

配置项	配置项含义
twi0_used =xx	TWI 使用控制: 1 使用, 0 不用
twi0_scl =xx	TWI SCK 的 GPIO 配置

<code>twi0_sda=xx</code>	TWI SDA 的 GPIO 配置
--------------------------	-------------------

配置举例:

```
[twi0]
twi0_used      = 1
twi0_scl      = port:PH0<2><default><default><default>
twi0_sda      = port:PH1<2><default><default><default>
```

## 5.2. [twi1]

配置项	配置项含义
<code>twi1_used =xx</code>	TWI 使用控制: 1 使用, 0 不用
<code>twi1_scl =xx</code>	TWI SCK 的 GPIO 配置
<code>twi1_sda=xx</code>	TWI SDA 的 GPIO 配置

配置举例:

```
twi1_used      = 1
twi1_scl      = port:PH2<2><default><default><default>
twi1_sda      = port:PH3<2><default><default><default>
```

## 5.3. [twi2]

配置项	配置项含义
<code>twi2_used =xx</code>	TWI 使用控制: 1 使用, 0 不用
<code>twi2_scl =xx</code>	TWI SCK 的 GPIO 配置
<code>twi2_sda=xx</code>	TWI SDA 的 GPIO 配置

配置举例:

```
[twi2]
twi2_used      = 0
twi2_scl      = port:PE14<3><default><default><default>
twi2_sda      = port:PE15<3><default><default><default>
```

# 6. 串口(UART)

主控有 5 路 uart 接口, 5 路支持 4 线或者 2 线通讯 (但十分不建议用 uart0 作为控制台以外的用途), 实例中, 有些路仅仅写出 2 路的配置形式, 但实际使用时只要将其按照 4 路的格式补全, 也能支持 4 线通讯

## 6.1. [uart0]

配置项	配置项含义
<code>uart_used =xx</code>	UART 使用控制: 1 使用, 0 不用
<code>uart_port =xx</code>	UART 端口号
<code>uart_type = xx</code>	UART 类型

<code>uart_tx =xx</code>	UART TX 的 GPIO 配置
<code>uart_rx=xx</code>	UART RX 的 GPIO 配置

配置举例:

```
[uart0]
uart_used      = 1
uart_port      = 0
uart_type      = 2
uart0_tx       = port:PB8<4><1><default><default>
uart0_rx       = port:PB9<4><1><default><default>
```

## 6.2. [uart1]

配置项	配置项含义
<code>uart_used</code>	UART 使用控制: 1 使用, 0 不用
<code>uart_port</code>	UART 端口号
<code>uart_type</code>	UART 类型
<code>uart_tx</code>	UART TX 的 GPIO 配置
<code>uart_rx</code>	UART RX 的 GPIO 配置
<code>uart_rts</code>	UART RTS 的 GPIO 配置
<code>uart_cts</code>	UART CTS 的 GPIO 配置

配置举例:

```
[uart1]
uart1_used     = 1
uart1_port     = 1
uart1_type     = 4
uart1_tx       = port:PG6<2><1><default><default>
uart1_rx       = port:PG7<2><1><default><default>
uart1_rts      = port:PG8<2><1><default><default>
uart1_cts      = port:PG9<2><1><default><default>
```

## 6.3. [uart2]

配置项	配置项含义
<code>uart_used =xx</code>	UART 使用控制: 1 使用, 0 不用
<code>uart_port =xx</code>	UART 端口号
<code>uart_type =xx</code>	UART 类型
<code>uart_tx =xx</code>	UART TX 的 GPIO 配置
<code>uart_rx =xx</code>	UART RX 的 GPIO 配置
<code>uart_rts=xx</code>	UART RTS 的 GPIO 配置
<code>uart_cts=xx</code>	UART CTS 的 GPIO 配置

配置举例:

```
[uart2]
uart2_used     = 0
```



```

uart2_port      = 2
uart2_type     = 4
uart2_tx       = port:PB0<2><1><default><default>
uart2_rx       = port:PB1<2><1><default><default>
uart2_rts      = port:PB2<2><1><default><default>
uart2_cts      = port:PB3<2><1><default><default>

```

## 6.4. [uart3]

配置项	配置项含义
uart_used =xx	UART 使用控制：1 使用，0 不用
uart_port =xx	UART 端口号
uart_type =xx	UART 类型
uart_tx =xx	UART TX 的 GPIO 配置
uart_rx =xx	UART RX 的 GPIO 配置
uart_rts =xx	UART RTS 的 GPIO 配置
uart_cts =xx	UART CTS 的 GPIO 配置

配置举例：

```

[uart3]
uart3_used      = 0
uart3_port     = 3
uart3_type     = 4
uart3_tx       = port:PH4<2><1><default><default>
uart3_rx       = port:PH5<2><1><default><default>
uart3_rts      = port:PH6<2><1><default><default>
uart3_cts      = port:PH7<2><1><default><default>

```

## 6.5. [uart4]

配置项	配置项含义
uart_used =xx	UART 使用控制：1 使用，0 不用
uart_port =xx	UART 端口号
uart_type =xx	UART 类型
uart4_tx =xx	UART TX 的 GPIO 配置
uart4_rx =xx	UART RX 的 GPIO 配置

配置举例：

```

[uart4]
uart4_used      = 0
uart4_port     = 4
uart4_type     = 4
uart4_tx       = port:PD2<3><1><default><default>
uart4_rx       = port:PD3<3><1><default><default>
uart4_rts      = port:PD4<3><1><default><default>
uart4_cts      = port:PD5<3><1><default><default>

```

## 7. SPI 总线

### 7.1. [spi0]

配置项	配置项含义
spi_used =xx	SPI 使用控制：1 使用，0 不用
spi_cs_bitmap =xx	由于 SPI 控制器支持多个 CS，这一个参数表示 CS 的掩码；
spi_cs0 =xx	SPI CS0 的 GPIO 配置
spi_sclk =xx	SPI CLK 的 GPIO 配置
spi_mosi =xx	SPI MOSI 的 GPIO 配置
spi_miso =xx	SPI MISO 的 GPIO 配置

配置举例：

```
[spi0]
spi0_used      = 0
spi0_cs_number = 1
spi0_cs_bitmap = 1
spi0_cs0       = port:PC3<4><1><default><default>
spi0_sclk      = port:PC2<4><default><default><default>
spi0_mosi      = port:PC0<4><default><default><default>
spi0_miso      = port:PC1<4><default><default><default>
```

### 7.2. [spi1]

配置项	配置项含义
spi_used =xx	SPI 使用控制：1 使用，0 不用
spi_cs_bitmap =xx	由于 SPI 控制器支持多个 CS，这一个参数表示 CS 的掩码；
spi_cs0 =xx	SPI CS0 的 GPIO 配置
spi_sclk =xx	SPI CLK 的 GPIO 配置
spi_mosi =xx	SPI MOSI 的 GPIO 配置
spi_miso =xx	SPI MISO 的 GPIO 配置

配置举例：

```
[spi1]
spi1_used      = 0
spi1_cs_number = 1
spi1_cs_bitmap = 1
spi1_cs0       = port:PD0<4><1><default><default>
spi1_sclk      = port:PD1<4><default><default><default>
spi1_mosi      = port:PD2<4><default><default><default>
```

```
spi1_miso = port:PD3<4><default><default><default>
```

## 8. 电容屏

### 8.1. [ctp]

配置项	配置项含义
compatible	Device tree 中 compatible 设置
ctp_used	该选项为是否开启电容触摸，支持的话置 1，反之置 0
ctp_name	tp 的 name，必须配，与驱动保持一致
ctp_twi_id	用于选择 i2c adapter, 可选 1, 2
ctp_twi_addr	指明 i2c 设备地址，与具体硬件相关
ctp_screen_max_x	触摸板的 x 轴最大坐标
ctp_screen_max_y	触摸板的 y 轴最大坐标
ctp_revert_x_flag	是否需要翻转 x 坐标，需要则置 1，反之置 0
ctp_revert_y_flag	是否需要翻转 y 坐标，需要则置 1，反之置 0
ctp_exchange_x_y_flag	是否需要 x 轴 y 轴坐标对换
ctp_int_port	电容屏中断信号的 GPIO 配置
ctp_wakeup	电容屏唤醒信号的 GPIO 配置
ctp_power_ldo	电容屏供电 ldo
ctp_power_ldo_vol	电容屏供电 ldo 电压
ctp_power_io	电容屏供电 gpio

配置举例：

```
[ctp]
compatible = "allwinner,sun50i-ctp-para"
ctp_used = 1
ctp_name = "gsl_t1_v2"
ctp_twi_id = 0
ctp_twi_addr = 0x40
ctp_screen_max_x = 800
ctp_screen_max_y = 1280
ctp_revert_x_flag = 0
ctp_revert_y_flag = 0
ctp_exchange_x_y_flag = 0

ctp_int_port = port:PH04<6><default><default><default>
ctp_wakeup = port:PH08<1><default><default><1>
```

```

ctp_power_ldo      = "vcc-ctp"
ctp_power_ldo_vol  = 3300
ctp_power_io       =

```

## 8.2. [ctp\_list]

配置项	配置项含义
compatible	Device tree 中 compatible 设置
ctp_list_used	0 or 1, 是否使用自动检测功能
gslX680new	0 or 1, 检测时是否扫描此类触屏
gt9xx_ts	0 or 1, 检测时是否扫描此类触屏
gt9xxnew_ts	0 or 1, 检测时是否扫描此类触屏
gt82x	0 or 1, 检测时是否扫描此类触屏
zet622x	0 or 1, 检测时是否扫描此类触屏
aw5306_ts	0 or 1, 检测时是否扫描此类触屏

配置举例:

```

[ctp_list]
compatible      = "allwinner,sun50i-ctp-list"
ctp_list_used   = 1
gslX680new     = 1
gt9xx_ts       = 0
gt9xxnew_ts    = 1
gt82x          = 1
zet622x        = 1
aw5306_ts      = 1

```

## 9. 闪存

### 9.1. [nand0\_para]

配置项	配置项含义
nand_support_2ch	nand0 是否使能双通道
nand0_used =xx	nand0 模块使能标志
nand0_we =xx	nand0 写时钟信号的 GPIO 配置
nand0_ale =xx	nand0 地址使能信号的 GPIO 配置
nand0_cle =xx	nand0 命令使能信号的 GPIO 配置
nand0_ce1 =xx	nand0 片选 1 信号的 GPIO 配置
nand0_ce0 =xx	nand0 片选 0 信号的 GPIO 配置
nand0_nre =xx	nand0 读时钟信号的 GPIO 配置
nand0_rb0=xx	nand0 Read/Busy 1 信号的 GPIO 配置
nand0_rb1 =xx	nand0 Read/Busy 0 信号的 GPIO 配置

nand0_d0=xx	nand0 数据总线信号的 GPIO 配置
nand0_d1=xx	/
nand0_d2=xx	/
nand0_d3=xx	/
nand0_d4=xx	/
nand0_d5=xx	/
nand0_d6=xx	/
nand0_d7=xx	/
nand0_ce2=xx	nand0 片选 2 信号的 GPIO 配置
nand0_ce3=xx	nand0 片选 3 信号的 GPIO 配置
nand0_ndqs=xx	nand0 ddr 时钟信号的 GPIO 配置
nand0_regulator1	nand0 电源 1, 3V, 从[pmu0_regu]中查找对应的 regulator, 比如"vcc-nand"
nand0_regulator2	nand0 电源 2, 1.8V 一般不用
nand0_cache_level	配置默认值 0x55aaaa55
nand0_flush_cache_num	配置默认值 0x55aaaa55
nand0_capacity_level	配置默认值 0x55aaaa55
nand0_id_number_ctl	配置默认值 0x55aaaa55
nand0_print_level	配置默认值 0x55aaaa55
nand0_p0	配置默认值 0x55aaaa55
nand0_p1	配置默认值 0x55aaaa55
nand0_p2	配置默认值 0x55aaaa55
nand0_p3	配置默认值 0x55aaaa55

配置举例:

```
[nand0_para]
nand_support_2ch    = 0
nand0_used          = 1
nand0_we            = port:PC00<2><default><default><default>
nand0_ale           = port:PC01<2><default><default><default>
nand0_cle           = port:PC02<2><default><default><default>
nand0_ce1           = port:PC03<2><default><default><default>
nand0_ce0           = port:PC04<2><default><default><default>
nand0_nre           = port:PC05<2><default><default><default>
nand0_rb0           = port:PC06<2><default><default><default>
nand0_rb1           = port:PC07<2><default><default><default>
nand0_d0            = port:PC08<2><default><default><default>
nand0_d1            = port:PC09<2><default><default><default>
nand0_d2            = port:PC10<2><default><default><default>
nand0_d3            = port:PC11<2><default><default><default>
nand0_d4            = port:PC12<2><default><default><default>
nand0_d5            = port:PC13<2><default><default><default>
nand0_d6            = port:PC14<2><default><default><default>
nand0_d7            = port:PC15<2><default><default><default>
```

```

nand0_ndqs          = port:PC16<2><default><default><default>
nand0_ce2           = port:PC17<2><default><default><default>
nand0_ce3           = port:PC18<2><default><default><default>
nand0_regulator1    = "vcc-nand"
nand0_regulator2    = "none"
nand0_cache_level   = 0x55aaaa55
nand0_flush_cache_num = 0x55aaaa55
nand0_capacity_level = 0x55aaaa55
nand0_id_number_ctl = 0x55aaaa55
nand0_print_level   = 0x55aaaa55
nand0_p0             = 0x55aaaa55
nand0_p1             = 0x55aaaa55
nand0_p2             = 0x55aaaa55
nand0_p3             = 0x55aaaa55

```

## 10. 显示

### 10.1. [disp]

配置项	配置项含义
disp_init_enable=xx	是否进行显示的初始化设置
disp_mode =xx	显示模式: 0:screen0<screen0,fb0>
screen0_output_type=xx	屏 0 输出类型(0:none; 1:lcd; 3:hDMI; )
screen0_output_mode =xx	屏 0 输出模式(used for hDMI output, 0:480i 1:576i 2:480p 3:576p 4:720p50 5:720p60 6:1080i50 7:1080i60 8:1080p24 9:1080p50 10:1080p60)
screen1_output_type=xx	屏 1 输出类型(0:none; 1:lcd; 3:hDMI; )
screen1_output_mode=xx	屏 1 输出模式(used for hDMI output, 0:480i 1:576i 2:480p 3:576p 4:720p50 5:720p60 6:1080i50 7:1080i60 8:1080p24 9:1080p50 10:1080p60)
fb0_format=xx	fb0 的格式 0:ARGB 1:ABGR 2:RGBA 3:BGRA 5:RGB565 8:RGB888 12:ARGB4444 16:ARGB1555 18:RGBA5551
fb0_width=xx	fb0 的宽度,为 0 时将按照输出设备的分辨率
fb0_height=xx	fb0 的高度, 为 0 时将按照输出设备的分辨率
fb1_format=xx	0:ARGB 1:ABGR 2:RGBA 3:BGRA

	5:RGB565 8:RGB888 12:ARGB4444 16:ARGB1555 18:RGBA5551
fb1_width=xx	Fb1 的宽度,为 0 时将按照输出设备的分辨率
fb1_height=xx	Fb1 的高度, 为 0 时将按照输出设备的分辨率

配置举例:

[disp]

```

disp_init_enable      = 1
disp_mode              = 0
screen0_output_type   = 1
screen0_output_mode   = 4
screen1_output_type   = 3
screen1_output_mode   = 4
fb0_format             = 0
fb0_width              = 0
fb0_height             = 0
fb1_format             = 0
fb1_width              = 0
fb1_height             = 0

```

## 11. LCD 屏

### 11.1. [lcd0]

配置项	配置项含义
lcd_used=xx	是否使用 lcd0
lcd_driver_name = "xx"	定义驱动名称
lcd_backlight	lcd init backlight
lcd_if =xx	lcd 接口 (0:lv(sync+de); 1:8080; 2:ttl; 3:lvds, 4:dsi; 5:edp)
lcd_x=xx	Lcd 分辨率 x
lcd_y =xx	Lcd 分辨率 y
lcd_width = xx	Lcd 屏宽度
lcd_height = xx	Lcd 屏高度
lcd_dclk_freq = xx	Lcd 频率
lcd_pwm_used =	Pwm 是否使用
lcd_pwm_ch =	Pwm 通道
lcd_pwm_freq=xx	Pwm 频率
lcd_pwm_pol =xx	pwm 属性, 0:positive; 1:negative

lcd_pwm_max_limit	lcd backlight PWM max limit(<=255)
lcd_hbp=xx	Lcd 行后沿时间
lcd_ht=xx	Lcd 行时间
lcd_hspw = xx	Lcd 行同步脉宽
lcd_vbp=xx	Lcd 场后沿时间
lcd_vt=xx	Lcd 场时间
lcd_vspw=xx	Lcd 场同步脉宽
lcd_lvds_if=xx	Lcd lvds 接口, 0:single link; 1:dual link
lcd_lvds_colordepth =xx	Lcd lvds 颜色深度 0:8bit; 1:6bit
lcd_lvds_mode=xx	Lcd lvds 模式, 0:NS mode; 1:JEIDA mode
lcd_frm=xx	Lcd 格式, 0:disable; 1:enable rgb666 dither; 2:enable rgb656 dither
lcd_hv_clk_phase	Lcd hv 时钟 相位 0:noraml; 1:intert phase(0~3bit: vsync phase;4~7bit:hsync phase; 8~11bit:dclk phase; 12~15bit:de phase)
lcd_hv_sync_polarity	Lcd io 属性, 0:not invert; 1:invert
lcd_gamma_en=xx	Lcdgamma 校正使能
lcd_bright_curve_en=xx	Lcd 亮度曲线校正使能
lcd_cmap_en=xx	Lcd 调色板函数使能
lcd_bl_en=xx	背光使能的 GPIO 配置
lcd_bl_en_power=xx	背光电源
lcd_power=xx	Lcd power 电源
lcd_fix_power=xx	Lcd fix power 电源

配置举例:

[lcd0]

lcd\_used = 1

lcd\_driver\_name = "dx0960be40a1"

lcd\_backlight = 50

lcd\_if = 4

lcd\_x = 800

lcd\_y = 1280

lcd\_width = 129

lcd\_height = 206

lcd\_dclk\_freq = 103

lcd\_pwm\_used = 1

lcd\_pwm\_ch = 0

lcd\_pwm\_freq = 50000

lcd\_pwm\_pol = 1

lcd\_pwm\_max\_limit = 250

lcd\_hbp = 60



```

lcd_ht                = 1300
lcd_hspw              = 20
lcd_vbp               = 21
lcd_vt                = 1332
lcd_vspw              = 10
lcd_lvds_if           = 0
lcd_lvds_colordepth = 0
lcd_lvds_mode         = 0
lcd_frm               = 0
lcd_hv_clk_phase     = 0
lcd_hv_sync_polarity= 0
lcd_dsi_if            = 0
lcd_dsi_lane          = 4
lcd_dsi_format        = 0
lcd_dsi_te            = 0
lcd_gamma_en          = 0
lcd_bright_curve_en = 0
lcd_cmap_en           = 0

lcd_bl_en              = port:PD23<1><<0><default><<1>
lcd_bl_en_power        = "none"
lcd_power              = "vcc-mipi"
lcd_fix_power          = "vcc-dsi-33"

```

## 12. HDMI

### 12.1. [hdmi]

配置项	配置项含义
hdmi_used =xx	是否启用 hdmi，启用置 1，反之置 0
hdmi_power =xx	Hdmi 电源
hdmi_hdcp_enable	HDCP 功能使能标记，使能置 1，反之置 0
hdmi_cts_compatibility	

配置举例：

```

hdmi_used = 1
hdmi_power = "vcc-hdmi-33"
hdmi_hdcp_enable = 0
hdmi_cts_compatibility = 0

```

## 13. PWM

### 13.1. [pwm]

配置项	配置项含义
pwm_used =xx	是否启用 pwm, 启用置 1, 反之置 0
pwm_positive=xx	Pwm positive GPIO 配置

配置举例:

```
pwm_used          = 1
pwm_positive      = port:PD22<2><0><default><default>
```

### 13.2. [pwm\_suspend]

配置项	配置项含义
pwm_positive	Pwm positive GPIO 配置

配置举例:

```
[pwm_suspend]
pwm_positive      = port:PD22<7><0><default><default>
```

## 14. [boot\_disp]

### 14.1. [boot\_disp]

配置项	配置项含义
output_disp	0: screen0 1:screen1
output_type	0:none; 1:lcd; 3:hdmi;
output_mode	used for hdmi output, 0:480i 1:576i 2:480p 3:576p 4:720p50 5:720p60 6:1080i50 7:1080i60 8:1080p24 9:1080p50 10:1080p60

配置举例:

```
[boot_disp]
output_disp = 0
output_type = 1
output_mode = 4
```

## 15. 摄像头(CSI)

### 15.1. [csi0]

配置项	配置项含义
csi0_used	CSI 使用控制：1 使用，0 不用
csi0_sensor_list	如果配置了 system/etc/hawkview/sensor_list_cfg.ini 文件，填 1，默认填 0。
csi0_pck	
csi0_mck	Mipi mclk 信号的 GPIO 配置。
csi0_hsync	
csi0_vsync	
csi0_d0	csi0 数据线 GPIO 配置
csi0_d1	csi0 数据线 GPIO 配置
csi0_d2	csi0 数据线 GPIO 配置
csi0_d3	csi0 数据线 GPIO 配置
csi0_d4	csi0 数据线 GPIO 配置
csi0_d5	csi0 数据线 GPIO 配置
csi0_d6	csi0 数据线 GPIO 配置
csi0_d7	csi0 数据线 GPIO 配置
csi0_sck	CSI0 CCI 时钟信号的 GPIO 配置。 如果使用 CSI0 内部 CCI 需要配置该项
csi0_sda	CSI0 CCI 数据信号的 GPIO 配置。 如果使用 CSI0 内部 CCI 需要配置该项

配置举例：

[csi0]

csi0\_used = 1

csi0\_sensor\_list = 1

csi0\_pck = port:PE00<2><default><default><default>

csi0\_mck = port:PE01<0><0><1><0>

csi0\_hsync = port:PE02<2><default><default><default>

csi0\_vsync = port:PE03<2><default><default><default>

csi0\_d0 = port:PE04<2><default><default><default>

csi0\_d1 = port:PE05<2><default><default><default>

csi0\_d2 = port:PE06<2><default><default><default>

csi0\_d3 = port:PE07<2><default><default><default>

csi0\_d4 = port:PE08<2><default><default><default>

csi0\_d5 = port:PE09<2><default><default><default>

csi0\_d6 = port:PE10<2><default><default><default>

csi0\_d7 = port:PE11<2><default><default><default>  
csi0\_sck = port:PE12<2><default><default><default>  
csi0\_sda = port:PE13<2><default><default><default>

## 15.2. [csi0/csi0\_dev0]

配置项	配置项含义
csi0_dev0_used	0:disable 1:enable
csi0_dev0_mname	设置 sensor 0 名称,
csi0_dev0_twi_addr	请参考实际模组的 8bit ID 填写
csi0_dev0_pos	摄像头位置前置填"front", 后置填"rear"。
csi0_dev0_isp_used	0:not use isp 1:use isp
csi0_dev0_fmt	0:yuv 1:bayer raw rgb
csi0_dev0_stby_mode	0:not shut down power at standby 1:shut down power at standby
csi0_dev0_vflip	flip in vertical direction 0:disable 1:enable
csi0_dev0_hflip	flip in horizontal direction 0:disable 1:enable
csi0_dev0_iovdd	camera module io power handle string, pmu power supply
csi0_dev0_iovdd_vol	camera module io power voltage, pmu power supply
csi0_dev0_avdd	camera module analog power handle string, pmu power supply
csi0_dev0_avdd_vol	camera module analog power voltage, pmu power supply
csi0_dev0_dvdd	camera module core power handle string, pmu power supply
csi0_dev0_dvdd_vol	camera module core power voltage, pmu power supply
csi0_dev0_afvdd	camera module vcm power handle string, pmu power supply
csi0_dev0_afvdd_vol	camera module vcm power voltage, pmu power supply
csi0_dev0_power_en	Sensor power enable 引脚 GPIO 配置。
csi0_dev0_reset	Sensor reset 引脚 GPIO 配置
csi0_dev0_pwn	Sensor power down 引脚 GPIO 配置
csi0_dev0_flash_en	闪光灯 enable 引脚 GPIO 配置
csi0_dev0_flash_mode	闪光灯 flash mode 引脚 GPIO 配置
csi0_dev0_af_pwn	VCM driver power down 引脚 GPIO 配置

csi0_dev0_act_used	模组包含 VCM driver 时候填 1。
csi0_dev0_act_name	VCM driver 名字，如 “ad5820_act”
csi0_dev0_act_slave	VCM driver slave 地址

说明：fill voltage in uV, e.g. iovdd = 2.8V, csi0\_dev0\_iovdd\_vol = 2800000

配置举例：

```
[csi0/csi0_dev0]
csi0_dev0_used          = 1
csi0_dev0_mname        = "gc2155"
csi0_dev0_twi_addr     = 0x78
csi0_dev0_pos          = "rear"
csi0_dev0_isp_used     = 1
csi0_dev0_fmt          = 0
csi0_dev0_stby_mode    = 1
csi0_dev0_vflip        = 0
csi0_dev0_hflip        = 0
csi0_dev0_iovdd        = "iovdd-csi"
csi0_dev0_iovdd_vol    = 2800000
csi0_dev0_avdd         = "avdd-csi"
csi0_dev0_avdd_vol     = 2800000
csi0_dev0_dvdd         = "dvdd-csi-18"
csi0_dev0_dvdd_vol     = 1800000
csi0_dev0_afvdd        = ""
csi0_dev0_afvdd_vol    =
csi0_dev0_power_en     =
csi0_dev0_reset        = port:PE14<0><0><1><0>
csi0_dev0_pwrn         = port:PE16<0><0><1><0>
csi0_dev0_flash_en     =
csi0_dev0_flash_mode   =
csi0_dev0_af_pwrn      =
csi0_dev0_act_used     = 0
csi0_dev0_act_name     = "ad5820_act"
csi0_dev0_act_slave    = 0x18
```

### 15.3. [csi0/csi0\_dev1]

配置项	配置项含义
csi0_dev1_used	0:disable 1:enable
csi0_dev1_mname	设置 sensor 0 名称，
csi0_dev1_twi_addr	请参考实际模组的 8bit ID 填写
csi0_dev1_pos	摄像头位置前置填"front"，后置填"rear"。
csi0_dev1_isp_used	0:not use isp 1:use isp
csi0_dev1_fmt	0:yuv 1:bayer raw rgb
csi0_dev1_stby_mode	0:not shut down power at standby 1:shut

	down power at standby
csi0_dev1_vflip	flip in vertical direction 0:disable 1:enable
csi0_dev1_hflip	flip in horizontal direction 0:disable 1:enable
csi0_dev1_iovdd	camera module io power handle string, pmu power supply
csi0_dev1_iovdd_vol	camera module io power voltage, pmu power supply
csi0_dev1_avdd	camera module analog power handle string, pmu power supply
csi0_dev1_avdd_vol	camera module analog power voltage, pmu power supply
csi0_dev1_dvdd	camera module core power handle string, pmu power supply
csi0_dev1_dvdd_vol	camera module core power voltage, pmu power supply
csi0_dev1_afvdd	camera module vcm power handle string, pmu power supply
csi0_dev1_afvdd_vol	camera module vcm power voltage, pmu power supply
csi0_dev1_power_en	Sensor power enable 引脚 GPIO 配置。
csi0_dev1_reset	Sensor reset 引脚 GPIO 配置
csi0_dev1_pwn	Sensor power down 引脚 GPIO 配置
csi0_dev1_flash_en	闪光灯 enable 引脚 GPIO 配置
csi0_dev1_flash_mode	闪光灯 flash mode 引脚 GPIO 配置
csi0_dev1_af_pwn	VCM driver power down 引脚 GPIO 配 置
csi0_dev1_act_used	模组包含 VCM driver 时候填 1。
csi0_dev1_act_name	VCM driver 名字, 如 “ad5820_act “
csi0_dev1_act_slave	VCM driver slave 地址

配置举例:

```
[csi0/csi0_dev1]
csi0_dev1_used           = 1
csi0_dev1_mname         = "gc0328c"
csi0_dev1_twi_addr      = 0x42
csi0_dev1_pos           = "front"
csi0_dev1_isp_used      = 1
csi0_dev1_fmt           = 0
csi0_dev1_stby_mode     = 1
csi0_dev1_vflip         = 0
csi0_dev1_hflip         = 0
csi0_dev1_iovdd         = "iovdd-csi"
csi0_dev1_iovdd_vol     = 2800000
```

```

csi0_dev1_avdd           = "avdd-csi"
csi0_dev1_avdd_vol      = 2800000
csi0_dev1_dvdd         = "dvdd-csi-18"
csi0_dev1_dvdd_vol     = 1800000
csi0_dev1_afvdd        = ""
csi0_dev1_afvdd_vol    =
csi0_dev1_power_en     =
csi0_dev1_reset        = port:PE14<0><0><1><0>
csi0_dev1_pwdn         = port:PE15<0><0><1><0>
csi0_dev1_flash_en     =
csi0_dev1_flash_mode   =
csi0_dev1_af_pwdn      =
csi0_dev1_act_used     = 0
csi0_dev1_act_name     = "ad5820_act"
csi0_dev1_act_slave    = 0x18

```

## 16. SD / MMC

### 16.1. [sdc0]

配置项	配置项含义
Sdc0_used=xx	SDC 使用控制: 1 使用, 0 不用
bus-width=xx	位宽: 1-1bit, 4-4bit
sdc_d1=xx	SDC DATA1 的 GPIO 配置
sdc_d0=xx	SDC DATA0 的 GPIO 配置
sdc_clk=xx	SDC CLK 的 GPIO 配置
sdc_cmd=xx	SDC CMD 的 GPIO 配置
sdc_d3=xx	SDC DATA3 的 GPIO 配置
sdc_d2=xx	SDC DATA2 的 GPIO 配置
cd-gpios	card detect pin 的 GPIO 配置
sunxi-power-save-mode	For sdio wifi, should not be set when use sdio wifi
vmmc	power for card vdd, 从[pmu0_regu]中查找对应的 regulator
vqmmc	power for card io, 从[pmu0_regu]中查找对应的 regulator
vdmmc	power for card detect pin io, 从[pmu0_regu]中查找对应的 regulator

配置举例:

```

[sdc0]
sdc0_used           = 1

```

```

bus-width                = 4
sdc0_d1                  = port:PF00<2><1><2><default>
sdc0_d0                  = port:PF01<2><1><2><default>
sdc0_clk                  = port:PF02<2><1><2><default>
sdc0_cmd                  = port:PF03<2><1><2><default>
sdc0_d3                   = port:PF04<2><1><2><default>
sdc0_d2                   = port:PF05<2><1><2><default>
cd-gpios                  = port:PF06<0><1><2><default>
sunxi-power-save-mode =
vmmc                      = "none"
vqmmc                     = "none"
vdmmmc                     = "vcc-sdc"
    
```

## 16.2. [sdc1]

配置项	配置项含义
Sdc1_used=xx	SDC 使用控制: 1 使用, 0 不用
bus_width=xx	位宽: 1-1bit, 4-4bit
Sdc1_clk=xx	SDC CLK GPIO 配置
Sdc1_cmd=xx	SDC CMD GPIO 配置
Sdc1_d0=xx	SDC DATA0 GPIO 配置
Sdc1_d1=xx	SDC DATA1 GPIO 配置
Sdc1_d2=xx	SDC DATA2 GPIO 配置
Sdc1_d3=xx	SDC DATA3 GPIO 配置
sd-uhs-sdr50	card/sdio speed mode
sd-uhs-ddr50	card/sdio speed mode
sd-uhs-sdr104	card/sdio speed mode
cap-sdio-irq	sdio interrupt enable
keep-power-in-suspend	used for sdio wifi
ignore-pm-notify	used for sdio wifi
max-frequency	max clk sdc use,unit HZ

配置举例:

```

[sdc1]
sdc1_used                = 1
bus-width                = 4
sdc1_clk                  = port:PG00<2><1><3><default>
sdc1_cmd                  = port:PG01<2><1><3><default>
sdc1_d0                   = port:PG02<2><1><3><default>
sdc1_d1                   = port:PG03<2><1><3><default>
sdc1_d2                   = port:PG04<2><1><3><default>
sdc1_d3                   = port:PG05<2><1><3><default>
sd-uhs-sdr50              =
sd-uhs-ddr50              =
sd-uhs-sdr104            =
    
```



```

cap-sdio-irq          =
keep-power-in-suspend =
ignore-pm-notify=
max-frequency         = 150000000

```

### 16.3. [sdc2]

配置项	配置项含义
Sdc2_used=xx	SDC 使用控制: 1 使用, 0 不用
bus_width=xx	位宽: 1-1bit, 4-4bit, 8-8bit
non-removable	use for boot media,for example eMMC
sdc2_ds	
sdc_clk=xx	SDC CLK GPIO 配置
sdc_cmd=xx	SDC CMD GPIO 配置
sdc_d0=xx	SDC DATA0 GPIO 配置
sdc_d1=xx	SDC DATA1 GPIO 配置
sdc_d2=xx	SDC DATA2 GPIO 配置
sdc_d3=xx	SDC DATA3 GPIO 配置
sdc_d4=xx	SDC DATA4 GPIO 配置
sdc_d5=xx	SDC DATA5 GPIO 配置
sdc_d6=xx	SDC DATA6 GPIO 配置
sdc_d7=xx	SDC DATA7 GPIO 配置
sdc2_emmc_rst =xx	Emmc 的 reset 管脚
cd-gpios	card detect pin 的 GPIO 配置
sunxi-power-save-mode	For sdio wifi,should not be set when use sdio wifi
sunxi-dis-signal-vol-sw	Not allow to change io voltage
mmc-ddr-1_8v	speed mode for eMMC
mmc-hs200-1_8v	speed mode for eMMC
mmc-hs400-1_8v	speed mode for eMMC
max-frequency	max clk sdc use,unit HZ
sdc_tm4_sm0_freq0	host timing setting
sdc_tm4_sm0_freq1	host timing setting
sdc_tm4_sm1_freq0	/
sdc_tm4_sm1_freq1	/
sdc_tm4_sm2_freq0	/
sdc_tm4_sm2_freq1	/
sdc_tm4_sm3_freq0	/
sdc_tm4_sm3_freq1	/
sdc_tm4_sm4_freq0	/
sdc_tm4_sm4_freq1	/
vmmc	power for eMMC vcc, 从[pmu0_regu]中找到对应的字符串, 比如"vcc-emmc"
vqmmc	power for eMMC vccq, 从[pmu0_regu]

	中找到对应的字符串，比如"vcc-lpddr"
vdmmc	power for card detect pin io,not used for emmc，可配成"none"

配置举例：

```
[sdc2]
sdc2_used          = 1
non-removable     =
bus-width         = 8
sdc2_ds           = port:PC01<3><1><3><default>
sdc2_clk          = port:PC05<3><1><3><default>
sdc2_cmd         = port:PC06<3><1><3><default>
sdc2_d0          = port:PC08<3><1><3><default>
sdc2_d1          = port:PC09<3><1><3><default>
sdc2_d2          = port:PC10<3><1><3><default>
sdc2_d3          = port:PC11<3><1><3><default>
sdc2_d4          = port:PC12<3><1><3><default>
sdc2_d5          = port:PC13<3><1><3><default>
sdc2_d6          = port:PC14<3><1><3><default>
sdc2_d7          = port:PC15<3><1><3><default>
sdc2_emmc_rst    = port:PC16<3><1><3><default>
cd-gpios         =
sunxi-power-save-mode =
sunxi-dis-signal-vol-sw =
mmc-ddr-1_8v     =
mmc-hs200-1_8v  =
mmc-hs400-1_8v  =
max-frequency    = 100000000
sdc_tm4_sm0_freq0 = 0
sdc_tm4_sm0_freq1 = 0
sdc_tm4_sm1_freq0 = 0x00000000
sdc_tm4_sm1_freq1 = 0
sdc_tm4_sm2_freq0 = 0x00000000
sdc_tm4_sm2_freq1 = 0
sdc_tm4_sm3_freq0 = 0x05000000
sdc_tm4_sm3_freq1 = 0x00000405
sdc_tm4_sm4_freq0 = 0x00050000
sdc_tm4_sm4_freq1 = 0x00000408
vmmc             = "vcc-emmc"
vqmmc           = "vcc-lpddr"
vdmmc           = "none"
```

## 17. USB

### 17.1. [usbc0]

配置项	配置项含义
usb_used =xx	USB 使能标志(xx=1 or 0)。置 1，表示系统中 USB 模块可用，置 0，则表示系统 USB 禁用。此标志只对具体的 USB 控制器模块有效。
usb_port_type =xx	USB 端口的使用情况。(xx=0/1/2) 0: device only 1: host only 2: OTG
usb_detect_type=xx	USB 端口的检查方式。 0: 无检查方式 1: vbus/id 检查
usb_id_gpio=xx	USB ID pin 脚配置。具体请参考 gpio 配置说明。《配置与 GPIO 管理.doc》
usb_det_vbus_gpio=xx	USB DET_VBUS pin 脚配置。如果 GPIO 提供 pin，请参考 gpio 配置说明《配置与 GPIO 管理.doc》。如果的 AXP 提供 pin，则配置为: "axp_ctrl"。
usb_drv_vbus_gpio=xx	USB DRY_VBUS pin 脚配置。具体请参考 gpio 配置说明。《配置与 GPIO 管理.doc》
usb_restrict_gpio=xx	USB 限流控制 pin 脚 USB RESTRICT_GPIO pin 脚配置。具体请参考 gpio 配置说明。《配置与 GPIO 管理.doc》
usb_host_init_state=xx	host only 模式下，Host 端口初始化状态。 0: 初始化后 USB 不工作 1: 初始化后 USB 工作
usb_regulator_io	给控制器供电的 IO 域，从[pmu0_regu] 获取
usb_wakeup_suspend	Usb 唤醒功能标志位，1: 开启 0:关闭
usb_luns	PC 盘符个数
usb_serial_unique	usb 串口序列号是否唯一的标志，1: 唯一，0: 从 boot 获取 chipid
usb_serial_number	usb_serial_unique 为 1 时，该字符串作为 usb 的串口序列号
rndis_wceis	当 win8 系统上使用 rndis 功能时，要置 1

配置举例：

```

[usbc0]
usbc0_used          = 1
usb_port_type       = 2
usb_detect_type     = 1
usb_id_gpio         = port:PH09<0><1><default><default>
usb_det_vbus_gpio   = "axp_ctrl"
usb_drv_vbus_gpio   = port:power3<1><0><default><0>
usb_host_init_state = 0
usb_regulator_io    = "nocare"
usb_wakeup_suspend  = 0
;---      USB Device
usb_luns            = 3
usb_serial_unique   = 1
usb_serial_number   = "20080411"
rndis_wceis        = 1

```

## 17.2. [usbc1]

配置项	配置项含义
Usbc1_used =xx	USB 使能标志(xx=1 or 0)。置 1，表示系统中 USB 模块可用，置 0，则表示系统 USB 禁用。此标志只对具体的 USB 控制器模块有效。
usb_drv_vbus_gpio usb_host_init_state=xx	host_only 模式下，Host 端口初始化状态。 0: 初始化后 USB 不工作 1: 初始化后 USB 工作
usb_regulator_io	给 usb1 控制器供电的 IO 域，从 [pmu0_regu]获取
usb_wakeup_suspend	Usb 唤醒功能标志位，1: 开启 0:关闭
usb_hsic_used	是否使用 HSIC 功能
usb_hsic_regulator_io	给 HSIC 接口供电的 IO 域，从 [pmu0_regu]获取
usb_hsic_ctrl	是否需要软件控制 HSIC
usb_hsic_rdy_gpio	hsic 就绪状态的 GPIO 配置，usb_hsic_ctrl 为 1 时生效
usb_hsic_usb3503_flag	Hsic 接口是否支持 usb3503
usb_hsic_hub_connect_gpio	Usb3503 HSIC 的连接 gpio 配置
usb_hsic_int_n_gpio	Usb3503 HSIC 的中断 gpio 配置
usb_hsic_reset_n_gpio	Usb3503 HSIC 的复位 gpio 配置

配置举例：

```

[usbc1]
usbc1_used          = 0

```

```

usb_drv_vbus_gpio    = port:PB06<1><0><default><0>
usb_host_init_state = 1
usb_regulator_io     = "nocare"
usb_wakeup_suspend   = 0
;--- HSIC config
usb_hsic_used        = 0
usb_hsic_regulator_io = "vcc-hsic-12"
;--- Marvell 4G HSIC
usb_hsic_ctrl        = 0
usb_hsic_rdy_gpio    =
;--- SMSC usb3503 HSIC HUB
usb_hsic_usb3503_flag = 0
usb_hsic_hub_connect_gpio =
usb_hsic_int_n_gpio   =
usb_hsic_reset_n_gpio =

```

## 18. serial\_feature

配置项	配置项含义
sn_filename	Private 分区中, 存放有序列号的文件名

配置举例:

```
[serial_feature]
```

```
sn_filename      = "sn.txt"
```

## 19. 重力感应(G Sensor)

### 19.1. [gsensor]

配置项	配置项含义
compatible	Device tree 中 compatible 字段
gsensor_used=xx	1: 支持, 0: 不支持
gsensor_twi_id=xx	I2C 的 BUS 控制选择, 0 : TWI0;1:TWI1;2:TWI2
gsensor_twi_addr=xx	芯片的 I2C 地址
gsensor_vcc_io	供电配置, 从[pmu0_regu]中找到对应的字符串, 比如"vcc-deviceio"
gsensor_vcc_io_val	电压配置, 单位: mV
gsensor_int1=xx	中断 1 的 GPIO 配置
gsensor_int2=xx	中断 2 的 GPIO 配置

配置举例：

```
[gsensor]
compatible          = "allwinner,sun50i-gsensor-para"
gsensor_used       = 1
gsensor_twi_id     = 1
gsensor_twi_addr   = 0x1d
gsensor_vcc_io     = "vcc-deviceio"
gsensor_vcc_io_val = 3300
gsensor_int1       = port:PH05<6><1><default><default>
gsensor_int2       = port:PH06<6><1><default><default>
```

注：目前方案中支持 gsensor 的类型有以下列表，作为自动检测加载的时候使用，为 ‘1’ 时检测加载，为 ‘0’ 时不检测。

## 19.2. [gsensor\_list]

配置项	配置项含义
compatible	Device tree 中 compatible 字段
gsensor_list_used	0 or 1, 是否使用自动检测功能
lsm9ds0_acc_mag	0 or 1, 检测时是否扫描此类触屏
bma250	0 or 1, 检测时是否扫描此类触屏
mma8452	0 or 1, 检测时是否扫描此类触屏
mma7660	0 or 1, 检测时是否扫描此类触屏
mma865x	0 or 1, 检测时是否扫描此类触屏
afa750	0 or 1, 检测时是否扫描此类触屏
lis3de_acc	0 or 1, 检测时是否扫描此类触屏
lis3dh_acc	0 or 1, 检测时是否扫描此类触屏
kxtik	0 or 1, 检测时是否扫描此类触屏
dmard10	0 or 1, 检测时是否扫描此类触屏
dmard06	0 or 1, 检测时是否扫描此类触屏
mx622x	0 or 1, 检测时是否扫描此类触屏
fxos8700	0 or 1, 检测时是否扫描此类触屏
lsm303d	0 or 1, 检测时是否扫描此类触屏
sc7a30	0 or 1, 检测时是否扫描此类触屏

配置举例：

```
[gsensor_list]
compatible          = "allwinner,sun50i-gsensor-list-para"
gsensor_list_used  = 1
lsm9ds0_acc_mag    = 1
bma250             = 1
mma8452            = 1
mma7660            = 1
mma865x            = 1
```

afa750	= 1
lis3de_acc	= 1
lis3dh_acc	= 1
kxtik	= 1
dmard10	= 0
dmard06	= 1
mxc622x	= 1
fxos8700	= 1
lsm303d	= 0
sc7a30	= 1

confidential

## 20. WiFi

### 20.1. [wlan]

配置项	配置项含义
wlan_used	0-not use, 1- use
wlan_busnum	sdio/usb index
clocks	External low power clock input (32.768KHz)
wlan_power	input supply voltage, 从[pmu0_regu]中找到对应的字符串
wlan_io_regulator	IO 域供电配置, 从[pmu0_regu]中找到对应的字符串, 比如"vcc-wifi-io"
wlan_regon	power up/down internal regulators used by wifi section
wlan_hostwake	wlan to wake-up host

配置举例:

[wlan]

wlan\_used = 1

wlan\_busnum = 1

clocks =

wlan\_power =

wlan\_io\_regulator = "vcc-wifi-io"

wlan\_regon = port:PL02<1><default><default><0>

wlan\_hostwake = port:PL03<6><default><default><0>

## 21. 蓝牙(Bluetooth)

### 21.1. [bt]

配置项	配置项含义
bt_used	0-not use, 1- use
clocks	external low power clock input (32.768KHz)
bt_power	input supply voltage , 从[pmu0_regu]中找到对应的字符串
bt_io_regulator	供电配置, 从[pmu0_regu]中找到对应的字符串, 比如"vcc-wifi-io"
bt_rst_n	power up/down internal regulators used by BT section



配置举例:

```
[bt]
bt_used          = 1
clocks           =
bt_power         =
bt_io_regulator  = "vcc-wifi-io"
bt_rst_n         = port:PL04<1><default><default><0>
```

## 21.2. [btlpm]

配置项	配置项含义
btlpm_used =xx	0-not use, 1- use
uart_index	0- uart0, 1- uart1, 2- uart2
bt_wake	host wake-up bluetooth device, GPIO 配置
bt_hostwake	bt device wake-up host, GPIO 配置

配置举例:

```
[btlpm]
btlpm_used      = 1
uart_index      = 1
bt_wake         = port:PL06<1><default><default><1>
bt_hostwake     = port:PL05<6><default><default><0>
```

## 22. Gyro scope sensor

### 22.1. [gyroscopesensor]

配置项	配置项含义
compatible	Device tree 中 compatible 字段
gyroscopesensor_used	0-not use, 1- use
gy_twi_id	I2C 的 BUS 控制选择, 0 : TWI0;1:TWI1;2:TWI2
gy_twi_addr	芯片的 I2C 地址
gy_int1	中断 1 的 GPIO 配置
gy_int2	中断 2 的 GPIO 配置

配置举例:

```
[gyroscopesensor]
compatible       = "allwinner,sun50i-gyr_sensors-para"
gyroscopesensor_used = 0
```

```

gy_twi_id           = 2
gy_twi_addr        = 0x6a
gy_int1            = port:PA10<6><1><default><default>
gy_int2            =

```

## 22.2. [gy\_list]

配置项	配置项含义
compatible	Device tree 中 compatible 字段
gy_list_used	0-not use, 1- use
lsm9ds0_gyr	postposition 1 said detection, 0 means no detection.
l3gd20_gyr	postposition 1 said detection, 0 means no detection.
bmg160_gyr	postposition 1 said detection, 0 means no detection.

配置举例:

```

[gy_list]
compatible           ="allwinner,sun50i-gyr_sensors-list-para"
gy_list_used        = 0
lsm9ds0_gyr         = 1
l3gd20_gyr          = 0
bmg160_gyr          = 1

```

## 23. 光感(light sensor)

### 23.1. [lightsensor]

配置项	配置项含义
compatible	Device tree 中 compatible 字段
lightsensor_used	0-not use, 1- use
ls_twi_id=xx	I2C 的 BUS 控制选择, 0 : TWI0;1:TWI1;2:TWI2
ls_twi_addr =xx	芯片的 I2C 地址
ls_int=xx	中断的 GPIO 配置

配置举例:

```

[lightsensor]
compatible           ="allwinner,sun50i-lsensors-para"

```

```
lightsensor_used    =0
ls_twi_id           = 2
ls_twi_addr         = 0x23
ls_int              = port:PA12<6><1><default><default>
```

## 23.2. [ls\_list]

配置项	配置项含义
compatible	Device tree 中 compatible 字段
ls_list_used	Whether startup automatic inspection function. 1:used,0:unused
ltr_501als	Module name postposition 1 said detection, 0 means no detection
jsa1212	Module name postposition 1 said detection, 0 means no detection
jsa1127	Module name postposition 1 said detection, 0 means no detection
stk3x1x	Module name postposition 1 said detection, 0 means no detection

配置举例：

```
[ls_list]
compatible    ="allwinner,sun50i-lsensors-list-para"
ls_list_used  =0
ltr_501als    = 1
jsa1212      = 0
jsa1127      = 1
stk3x1x      = 0
```

## 24. 罗盘 Compass

### 24.1. [compasssensor]

配置项	配置项含义
compatible	Device tree 中 compatible 字段
compasssensor_used	1:used,0:unused
compass_twi_id=xx	I2C 的 BUS 控制选择， 0 : TWI0;1:TWI1;2:TWI2
compass_twi_addr =xx	芯片的 I2C 地址
compass_int =xx	中断的 GPIO 配置

配置举例：

```
[compasssensor]
compatible          ="allwinner,sun50i-compass-para"
compasssensor_used  = 0
compass_twi_id      = 2
compass_twi_addr    = 0x0d
compass_int         = port:PA11<6><1><<default><default>
```

## 24.2. [ls\_list]

配置项	配置项含义
compatible	Device tree 中 compatible 字段
compass_list_used	Whether startup automatic inspection function. 1:used,0:unused
lsm9ds0	Module name postposition 1 said detection, 0 means no detection
lsm303d	Module name postposition 1 said detection, 0 means no detection

配置举例：

```
[compass_list]
compatible          ="allwinner,sun50i-compass-list-para"
compass_list_used   = 0
lsm9ds0             = 1
lsm303d             = 0
```

## 25. 数字音频总线(spdif)

### 25.1. [spdif]、[sndspdif]

```
-----
;      NOTE :Make sure spdif_used = 0x1, sndspdif_used = 0x1,
;      if register the sound card spdif.
-----
```

配置举例：

```
[spdif]
spdif_used          = 0
[sndspdif]
sndspdif_used       = 0
```

## 25.2. [sndspdif]、[sndhdmi]

```

;-----
;      NOTE :Make sure daudio2_used = 0x1,sndhdmi_used = 0x1,
;      if register the sound card hdmi.
;-----

```

配置举例:

```

[daudio2]
daudio2_used = 1
[sndhdmi]
sndhdmi_used = 1

```

## 25.3. [daudio0]

配置项	配置项含义
daudio0_used	0:not use 1:use
pcm_lrck_period	16/32/64/128/256
pcm_lrckr_period	no use
slot_width_select	16bits/20bits/24bits/32bits
pcm_lsb_first	0: msb first; 1: lsb first
tx_data_mode	发送模式: 0: 16bit linear PCM; 1: 8bit linear PCM; 2: 8bit u-law; 3: 8bit a-law
rx_data_mode	接收模式: 0: 16bit linear PCM; 1: 8bit linear PCM; 2: 8bit u-law; 3: 8bit a-law
daudio_master	1: SND_SOC_DAI_FMT_CB_M_CFM(codec clk & FRM master) use 2: SND_SOC_DAI_FMT_CB_S_CFM(codec clk slave & FRM master) not use 3: SND_SOC_DAI_FMT_CB_M_CFS(codec clk master & frame slave) not use 4: SND_SOC_DAI_FMT_CB_S_CFS(codec clk & FRM slave) use
audio_format	1:SND_SOC_DAI_FMT_I2S(standard i2s format). 2:SND_SOC_DAI_FMT_RIGHT_J(right justified format). 3:SND_SOC_DAI_FMT_LEFT_J(left justified format) 4:SND_SOC_DAI_FMT_DSP_A(pcm. MSB is available on 2nd BCLK rising edge after LRC rising edge). 5:SND_SOC_DAI_FMT_DSP_B(pcm. MSB is available on 1nd BCLK rising edge after LRC
signal_inversion	1:SND_SOC_DAI_FMT_NB_NF(normal bit clock + frame) 2:SND_SOC_DAI_FMT_NB_IF(normal BCLK + inv FRM) 3:SND_SOC_DAI_FMT_IB_NF(invert BCLK + nor FRM) 4:SND_SOC_DAI_FMT_IB_IF(invert BCLK + FRM)
frametype	0: long frame = 2 clock width; 1: short frame

tdm_config	0:pcm 1:i2s
------------	-------------

配置举例:

```

;-----
;      NOTE :Make sure snddaudio0_used = 0x1,daudio1_used = 0x1,
;      if register the sound card DAUDIO1.
;-----

```

```

[snddaudio0]
snddaudio0_used = 0

```

```

;-----
[daudio0]
daudio0_used = 0
pcm_lrck_period = 0x20
pcm_lrckr_period = 0x01
slot_width_select = 0x20
pcm_lsb_first = 0x0
tx_data_mode = 0x0
rx_data_mode = 0x0
daudio_master = 0x04
audio_format = 0x01
signal_inversion = 0x01
frametype = 0x0
tdm_config = 0x01

```

```

;-----
;      NOTE :Make sure snddaudio1_used = 0x1,daudio0_used = 0x1,
;      if register the sound card DAUDIO0.
;-----

```

```

[snddaudio1]
snddaudio1_used = 0

```

```

;-----
[daudio1]
daudio1_used = 0
pcm_lrck_period = 0x20
pcm_lrckr_period = 0x01
slot_width_select = 0x20
pcm_lsb_first = 0x0
tx_data_mode = 0x0
rx_data_mode = 0x0
daudio_master = 0x04
audio_format = 0x01
signal_inversion = 0x01
frametype = 0x0
tdm_config = 0x01

```

## 26. 内置音频 codec

### 26.1. [sndcodec]

配置项	配置项含义
sndcodec_used	
aif2fmt	1:SND_SOC_DAI_FMT_I2S(standard i2s format). use 2:SND_SOC_DAI_FMT_RIGHT_J(right justified format). 3:SND_SOC_DAI_FMT_LEFT_J(left justified format) 4:SND_SOC_DAI_FMT_DSP_A(pcm. MSB is available on 2nd BCLK rising edge after LRC rising edge). use 5:SND_SOC_DAI_FMT_DSP_B(pcm. MSB is available on 1nd BCLK rising edge after LRC rising edge)
aif3fmt	1:SND_SOC_DAI_FMT_I2S(standard i2s format). use 2:SND_SOC_DAI_FMT_RIGHT_J(right justified format). 3:SND_SOC_DAI_FMT_LEFT_J(left justified format) 4:SND_SOC_DAI_FMT_DSP_A(pcm. MSB is available on 2nd BCLK rising edge after LRC rising edge). use 5:SND_SOC_DAI_FMT_DSP_B(pcm. MSB is available on 1nd BCLK rising edge after LRC rising edge)
aif2master	1: SND_SOC_DAI_FMT_CBM_CFM(codec clk & FRM master) use 2: SND_SOC_DAI_FMT_CBS_CFM(codec clk slave & FRM master) 3: SND_SOC_DAI_FMT_CBM_CFS(codec clk master & frame slave) 4: SND_SOC_DAI_FMT_CBS_CFS(codec clk & FRM slave)
hp_detect_case	0: low 1: high

### 26.2. [codec]

配置项	配置项含义
codec_used	Audiocodec 是否使用, 1: 打开 (默认) 0: 关闭
headphonevol	headphone volume:0x0--0x3f 0db--(-62db) 1db/step
spkervol	speaker volume:0x0--0x1f 0db-(-43.5db) 1.5db/step
earpiecevol	earpiece volume:0x0--0x1f 0db-(-43.5db) 1.5db/step
maingain	mainmic gain:0x0--0x7 0x0-0db 0x1:24db 3db/step
headsetmicgain	headphonemic gain:0x0--0x7 0x0-0db 0x1:24db 3db/step
adcag_cfg	1:use adcag 0:no use

adcdrc_cfg	1:use adcdrc 0:no use
adchpf_cfg	1:use adchpf_0:no use
dacdrdc_cfg	1:use dacdrdc 0:no use
dachpf_cfg	1:use dachpf 0:no use
aif2config	1:use aif2 0:no use
aif3config	1:use aif3 0:no use
aif1_lrlk_div	freq_blk/freq_lrck
aif2_lrlk_div	freq_blk/freq_lrck
pa_sleep_time	sleep ms after enable pa
gpio-spk	config gpio for enable/disable pa

## 26.3. [i2s]

配置项	配置项含义
i2s_used	0:not use 1:use

配置举例:

```

;-----
;      NOTE :Make sure sndcodec_used = 0x1,i2s_used = 0x1
;      codec_used = 0x1,if register the sound card audiocodec.
;-----
[sndcodec]
sndcodec_used = 0x1
aif2fmt = 0x3
aif3fmt = 0x3
aif2master = 0x1
hp_detect_case = 0x0
;-----
[i2s]
i2s_used = 0x1
;-----
[codec]
codec_used = 0x1
headphonevol = 0x3b
spkervol = 0x19
earpiecevol = 0x1e
maingain = 0x4
headsetmicgain = 0x4
adcage_cfg = 0x0
adcdrc_cfg = 0x0
adchpf_cfg = 0x0
dacdrdc_cfg = 0x0
dachpf_cfg = 0x0
aif2config = 0x0

```



```

aif3config = 0x0
aif1_lrlk_div = 0x40
aif2_lrlk_div = 0x40
pa_sleep_time = 0x15e
gpio-spk = port:PH07<2><1><default><default>

```

## 27. 红外 (IR)

### 27.1. [s\_cir0]

配置项	配置项含义
s_cir0_used =xx	是否支持 ir
ir_power_key_code	ir 遥控器电源键的编码值
ir_addr_code	ir 遥控器设备地址

配置举例:

```

[s_cir0]
s_cir0_used          = 0
ir_power_key_code    = 0x0
ir_addr_code         = 0x0

```

## 28. PMU 电源

### 28.1. [pmu0]

配置项	相关说明
used	是否使用 AXPxx: 0:不使用,1:使用
pmu_twi_addr	AXPxx 通信 I2C 地址
pmu_twi_id	AXPxx 挂接在主控的哪个 I2C 控制口 (0, 1, 2 ...)
pmu_irq_id	irq 号 (0 irq0,1 irq1,.....)
pmu_IRQ_wakeup	在关机和休眠状态下 IRQ 为低电平时是否触发开机和唤醒控制 0: 不开机或不唤醒 1: 开机或唤醒
pmu_chg_ic_temp	intelligence charge pmu temperature. when it is 0, this function is closed.
pmu_battery_rdc	电池通路内阻, 单位 mΩ
pmu_battery_cap	电池容量,单位 mAh, 如果配置改值, 计量方式为库仑计方式, 否则为电压方式
pmu_batdeten	电池检查使能控制: 0:使能 1:使能
pmu_runtime_chgcur	设置开机时充电电流大小, 单位 mA, 仅支

	持:300/450/600/750 /900/1050/1200/1350/1500/1650/1800/1950/2100
pmu_earllysuspend_chgcur	设置关屏时充电电流大小, 单位 mA, 仅支持: 300/4500/600/750/900/1050/1200/1350/1500/1650/1800/ 1950/ 2100
pmu_suspend_chgcur	设置待机时充电电流大小, 单位 mA, 仅支持: 300/4500/600/750 /900/1050/1200/1350/1500/1650/1800/1950/2100
pmu_shutdown_chgcur	设置关机时充电电流大小, 单位 mA, 仅支持: 300/4500/600/750 /900/1050/1200/1350/1500/1650/1800/1950/2100
pmu_init_chgvol	设置充电完成时电池目标电压, 仅支持: 4100/4200/4220/4240mV
pmu_init_chgend_rate	设置充电结束时电流占恒流值的百分比: 10/15
pmu_init_chg_enabled	开机后充电使能初始值: 0: 不开充电, 1: 开充电
pmu_init_adc_freq	ADC 采样频率设定值: 100/200/400/800 Hz
pmu_init_adcts_freq	TS ADC 采样频率设定值: 100/200/400/800 Hz
pmu_init_chg_pretime	涓流充电超时时间: 40/50/60/70 分钟
pmu_init_chg_csttime	恒流超时时间: 360/480/600/720 分钟
pmu_batt_cap_correct	满足电池容量校正条件后是否校正电池容量控制 0: 不校正 1: 校正
pmu_bat_regu_en	充电结束时, 充电开关是否关闭: 0: 关闭 1: 不关闭
pmu_bat_para1	电池空载电压为 3.13V 对应的电量值
pmu_bat_para2	电池空载电压为 3.27V 对应的电量值
pmu_bat_para3	电池空载电压为 3.34V 对应的电量值
pmu_bat_para4	电池空载电压为 3.41V 对应的电量值
pmu_bat_para5	电池空载电压为 3.58V 对应的电量值
pmu_bat_para6	电池空载电压为 3.52V 对应的电量值
pmu_bat_para7	电池空载电压为 3.55V 对应的电量值
pmu_bat_para8	电池空载电压为 3.57V 对应的电量值
pmu_bat_para9	电池空载电压为 3.59V 对应的电量值
pmu_bat_para10	电池空载电压为 3.61V 对应的电量值
pmu_bat_para11	电池空载电压为 3.63V 对应的电量值
pmu_bat_para12	电池空载电压为 3.64V 对应的电量值
pmu_bat_para13	电池空载电压为 3.66V 对应的电量值
pmu_bat_para14	电池空载电压为 3.7V 对应的电量值
pmu_bat_para15	电池空载电压为 3.73V 对应的电量值
pmu_bat_para16	电池空载电压为 3.77V 对应的电量值
pmu_bat_para17	电池空载电压为 3.78V 对应的电量值
pmu_bat_para18	电池空载电压为 3.8V 对应的电量值
pmu_bat_para19	电池空载电压为 3.82V 对应的电量值
pmu_bat_para20	电池空载电压为 3.84V 对应的电量值

pmu_bat_para21	电池空载电压为 3.85V 对应的电量值
pmu_bat_para22	电池空载电压为 3.87V 对应的电量值
pmu_bat_para23	电池空载电压为 3.91V 对应的电量值
pmu_bat_para24	电池空载电压为 3.94V 对应的电量值
pmu_bat_para25	电池空载电压为 3.98V 对应的电量值
pmu_bat_para26	电池空载电压为 4.01V 对应的电量值
pmu_bat_para27	电池空载电压为 4.05V 对应的电量值
pmu_bat_para28	电池空载电压为 4.08V 对应的电量值
pmu_bat_para29	电池空载电压为 4.1V 对应的电量值
pmu_bat_para30	电池空载电压为 4.12V 对应的电量值
pmu_bat_para31	电池空载电压为 4.14V 对应的电量值
pmu_bat_para32	电池空载电压为 4.15V 对应的电量值
pmu_usbvol_limit	USB 适配器限压功能控制 0: 不使能 1: 使能
pmu_usbcur_limit	USB 适配器限流功能控制 0: 不使能 1: 使能
pmu_usbvol	设置 USB 适配器限压值： 4000/4100/4200/4300/4400/4500/4600 4700 mV, 0-不限压
pmu_usbcur	设置 USB 适配器限流值: 500/900mA, 0-不限流
pmu_usbvol_pc	设置 USB 连接 PC 时限压值： 4000/4100/4200/4300/4400/4500 4600/4700 mV, 0-不限压
pmu_usbcur_pc	设置 USB 连接 PC 时限流值: 500/900mA, 0-不限流
pmu_pwroff_vol	PMU 关机时, 硬件低电保护电压设置值： 2600/2700/2800/2900 /3000/3100/3200/3300 mV
pmu_pwron_vol	PMU 开机后, 硬件低电保护电压设置值： 2600/2700/2800/2900 /3000/3100/3200/3300 mV
pmu_pekoff_time	长按键关机时间设置值: 4000/6000/8000/10000 ms
pmu_pekoff_func	长按键功能配置项: 0: 长按键后关机 1: 长按键后重启
pmu_pekoff_en	长按键后是否关闭 PMU: 0: 不关闭 1: 关闭
pmu_pekoff_delay_time	长按键关机激活时间设置, 0/10/20/30/40/50/60/70 秒
pmu_peklong_time	报长按键消息时间设定值: 1000/1500/2000/2500 ms
pmu_pekcon_time	关机情况下按键多长时间后启动设置： 128/1000/2000/3000 ms
pmu_pwrok_time	PWROK 启动延时时间设置值: 8/16/32/64 ms
pmu_pwrok_shutdown_en	长按 PWROK 键 6s 是否关机, 使能位
pmu_battery_warning_level1	低电报警门限 level 1 设置值百分比: 5~20, 每步设置 1%
pmu_battery_warning_level2	低电报警门限 level 2 设置值百分比: 0~15, 每步设置 1%

pmu_restvol_time	电池电量更新时间设置值：30/60/120 s
pmu_ocv_cou_adjust_time	根据 OCV 校正电池电量更新时间值：30/60/120 s
pmu_chgled_func	CHGLED 功能控制：0：马达驱动 1：充电状态指示
pmu_chgled_type	CHGLED 作为充电状态指示时指示功能控制：0：方式 A 1：方式 B
pmu_vbusen_func	N_VBUSEN 工作方式控制：0：作为输入脚 1：作为输出脚
pmu_reset	长按键 16s 后 PMU 是否重启控制：0：不重启 1：重启
pmu_hot_shutdown	PMU 过温后是否关机 0：不关机 1：关机
pmu_inshort	是否手动设置 ACIN/VBUS 短路控制 0：PMU 自动检测 1：手动设置 ACIN 和 VBUS 为短路
power_start	<p>火牛开机选择</p> <p>0：不允许插火牛直接开机，必须通过判断：满足以下条件可以直接开机：长按 power 按键，前次是系统状态，如果电池电量过低，则不允许开机</p> <p>1：任意状态下，允许插火牛直接开机，同时要求电池电量足够高</p> <p>2：不允许插火牛直接开机，必须通过判断：满足以下条件可以直接开机：长按 power 按键，前次是系统状态，不要求电池电量</p> <p>3：任意状态下，允许插火牛直接开机，不要求电池电量</p>
pmu_temp_enable	电池温度检测使能控制：0：disable 1：enable
pmu_charge_ltf	充电下限电池温度对应的电压
pmu_charge_hft	充电上限电池温度对应的电压
pmu_discharge_ltf	关机下限电池温度对应的电压
pmu_discharge_hft	关机上限电池温度对应的电压
pmu_temp_para1	电池温度-25 度对应的电压
pmu_temp_para2	电池温度-15 度对应的电压
pmu_temp_para3	电池温度-10 度对应的电压
pmu_temp_para4	电池温度-5 度对应的电压
pmu_temp_para5	电池温度 0 度对应的电压
pmu_temp_para6	电池温度 5 度对应的电压
pmu_temp_para7	电池温度 10 度对应的电压
pmu_temp_para8	电池温度 20 度对应的电压
pmu_temp_para9	电池温度 30 度对应的电压
pmu_temp_para10	电池温度 40 度对应的电压
pmu_temp_para11	电池温度 45 度对应的电压
pmu_temp_para12	电池温度 50 度对应的电压
pmu_temp_para13	电池温度 55 度对应的电压
pmu_temp_para14	电池温度 60 度对应的电压

pmu_temp_para15	电池温度 70 度对应的电压
pmu_temp_para16	电池温度 80 度对应的电压

配置举例：

[pmu0]

```
used                = 1
pmu_id              = 6
pmu_twi_addr        = 0x34
pmu_twi_id          = 1
pmu_irq_id          = 64
pmu_IRQ_wakeup      = 1
```

```
pmu_chg_ic_temp     = 0
pmu_battery_rdc     = 88
pmu_battery_cap     = 4800
pmu_runtime_chgcur  = 450
pmu_suspend_chgcur  = 1500
pmu_shutdown_chgcur = 1500
pmu_init_chgvol     = 4200
pmu_ac_vol          = 4000
pmu_ac_cur          = 3500
pmu_usbpc_vol       = 4400
pmu_usbpc_cur       = 500
pmu_battery_warning_level1 = 15
pmu_battery_warning_level2 = 0
pmu_chgled_func     = 0
pmu_chgled_type     = 0
```

```
pmu_bat_para1       = 0
pmu_bat_para2       = 0
pmu_bat_para3       = 0
pmu_bat_para4       = 0
pmu_bat_para5       = 0
pmu_bat_para6       = 0
pmu_bat_para7       = 1
pmu_bat_para8       = 1
pmu_bat_para9       = 2
pmu_bat_para10      = 3
pmu_bat_para11      = 4
pmu_bat_para12      = 10
pmu_bat_para13      = 17
pmu_bat_para14      = 26
pmu_bat_para15      = 41
pmu_bat_para16      = 46
```

pmu_bat_para17	= 51
pmu_bat_para18	= 56
pmu_bat_para19	= 59
pmu_bat_para20	= 65
pmu_bat_para21	= 69
pmu_bat_para22	= 75
pmu_bat_para23	= 79
pmu_bat_para24	= 83
pmu_bat_para25	= 89
pmu_bat_para26	= 95
pmu_bat_para27	= 98
pmu_bat_para28	= 100
pmu_bat_para29	= 100
pmu_bat_para30	= 100
pmu_bat_para31	= 100
pmu_bat_para32	= 100

pmu_bat_temp_enable	= 0
pmu_bat_charge_ltf	= 2261
pmu_bat_charge_hft	= 388
pmu_bat_shutdown_ltf	= 3200
pmu_bat_shutdown_hft	= 237
pmu_bat_temp_para1	= 7466
pmu_bat_temp_para2	= 4480
pmu_bat_temp_para3	= 3518
pmu_bat_temp_para4	= 2786
pmu_bat_temp_para5	= 2223
pmu_bat_temp_para6	= 1788
pmu_bat_temp_para7	= 1448
pmu_bat_temp_para8	= 969
pmu_bat_temp_para9	= 664
pmu_bat_temp_para10	= 466
pmu_bat_temp_para11	= 393
pmu_bat_temp_para12	= 333
pmu_bat_temp_para13	= 283
pmu_bat_temp_para14	= 242
pmu_bat_temp_para15	= 179
pmu_bat_temp_para16	= 134

pmu_powkey_off_time	= 6000
pmu_powkey_off_func	= 0
pmu_powkey_off_en	= 1
pmu_powkey_long_time	= 1500
pmu_powkey_on_time	= 1000

power\_start = 0

## 28.2. [pmu0\_regu]

regulator_count	Regulator 数目
regulator1	第 1 个 regulator 配置
regulator2	第 1 个 regulator 配置
regulatorX	第 X 个 regulator 配置

具体说明请参考《AXP8XX-regulator 使用文档-v1.0.doc》。

配置举例：

[pmu0\_regu]

regulator\_count = 23

```

regulator1      = "axp81x_dcdc1 none vcc-nand vcc-emmc vcc-sdc vcc-usb-30 vcc-io
vcc-pd"
regulator2      = "axp81x_dcdc2 none vdd-cpua"
regulator3      = "axp81x_dcdc3 none"
regulator4      = "axp81x_dcdc4 none"
regulator5      = "axp81x_dcdc5 none vcc-dram"
regulator6      = "axp81x_dcdc6 none vdd-sys"
regulator7      = "axp81x_dcdc7 none"
regulator8      = "axp81x_rtc none"
regulator9      = "axp81x_aldo1 none vdd-csi-led"
regulator10     = "axp81x_aldo2 none vcc-pl"
regulator11     = "axp81x_aldo3 none vcc-avcc vcc-pll"
regulator12     = "axp81x_dldo1 none vcc-hdmi-33 vcc-dsi-33"
regulator13     = "axp81x_dldo2 none vcc-mipi"
regulator14     = "axp81x_dldo3 none avdd-csi"
regulator15     = "axp81x_dldo4 none vcc-deviceio"
regulator16     = "axp81x_eldo1 none vcc-cpvdd vcc-wifi-io vcc-pc vcc-pg"
regulator17     = "axp81x_eldo2 none vcc-lcd-0"
regulator18     = "axp81x_eldo3 none dvdd-csi-18"
regulator19     = "axp81x_flldo1 none vcc-hsic-12"
regulator20     = "axp81x_flldo2 none vdd-cpus";
regulator21     = "axp81x_gpio0ldo none vcc-ctp"
regulator22     = "axp81x_gpio1ldo none iovdd-csi vcc-pe"
regulator23     = "axp81x_dc1sw none vcc-wifi-8089"

```

## 29. Recovery 键配置

key_min	= 3	作为 recovery 功能的按键的键值范围下限
key_max	= 5	作为 recovery 功能的按键的键值范围上限

配置举例：

[recovery\_key]

key\_min = 3

key\_max = 5

说明：

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。该键值配置用于作为 recovery 功能的按键的键值范围落在 key\_min 到 key\_max 之间。

## 30. DVFS

### 30.1. CPU DVFS

配置项	配置项含义
extremity_freq	极限超频模式频率上限
max_freq	最大运行频率
min_freq	最小运行频率
LV_count	VF 表项数
LV1_freq	LV1 对应频率段分界上限，0 表示结束符
LV1_volt	LV1 频率段的电压值，表示(LV2_freq, LV1_freq]范围类电压设置
LV2_freq	LV2 对应频率段分界上限，0 表示结束符
LV2_volt	LV2 频率段的电压值，表示(LV3_freq, LV2_freq]范围类电压设置
LV3_freq	LV3 对应频率段分界上限，0 表示结束符
LV3_volt	LV3 频率段的电压值，表示(LV2_freq, LV1_freq]范围类电压设置
LV4_freq	LV4 对应频率段分界上限，0 表示结束符
LV4_volt	LV4 频率段的电压值，表示(LV2_freq, LV1_freq]范围类电压设置
LV5_freq	LV5 对应频率段分界上限，0 表示结束符
LV5_volt	LV5 频率段的电压值，表示(LV2_freq, LV1_freq]范围类电压设置
LV6_freq	LV6 对应频率段分界上限，0 表示结束符



LV6_volt	LV6 频率段的电压值，表示(LV2_freq, LV1_freq]范围类电压设置
LV7_freq	LV7 对应频率段分界上限，0 表示结束符
LV7_volt	LV7 频率段的电压值，表示(LV2_freq, LV1_freq]范围类电压设置
LV8_freq	LV8 对应频率段分界上限，0 表示结束符
LV8_volt	LV8 频率段的电压值，表示(LV2_freq, LV1_freq]范围类电压设置

配置举例：

[dvfs\_table]

max\_freq = 1152000000

min\_freq = 480000000

lv\_count = 8

lv1\_freq = 1152000000

lv1\_volt = 1300

lv2\_freq = 1104000000

lv2\_volt = 1260

lv3\_freq = 1008000000

lv3\_volt = 1200

lv4\_freq = 816000000

lv4\_volt = 1100

lv5\_freq = 648000000

lv5\_volt = 1040

lv6\_freq = 0

lv6\_volt = 1040

lv7\_freq = 0

lv7\_volt = 1040

lv8\_freq = 0

lv8\_volt = 1040

## 31. Pinctrl 测试

配置项	配置项含义
Vdevice_used	作为 pinctrl test 的虚拟设备, 为 1 使能
Vdevice_0	虚拟设备的 gpio0 脚设置
Vdevice_1	虚拟设备的 gpio1 脚设置

配置举例:

```
[Vdevice]
Vdevice_used      = 1
Vdevice_0        = port:PA01<5><1><2><default>
Vdevice_1        = port:PA02<5><1><2><default>
```

## 32. [s\_uart0]

配置项	配置项含义
s_uart_used	使能 cpus 的 uart, 为 1 使能, 为 0 关闭
s_uart_tx	Uart 口发送引脚配置
s_uart_rx	Uart 接收引脚配置

配置举例:

```
[s_uart0]
s_uart0_used      = 1
s_uart0_tx        = port:PL02<2><default><default><default>
s_uart0_rx        = port:PL03<2><default><default><default>
```

## 33. [s\_rsb0]

配置项	配置项含义
s_rsb_used	使能 cpus 使用 rsb 总线, 为 1 使能, 为 0 关闭
s_rsb_sck	Rsb 时钟引脚设置
s_rsb_sda	Rsb 数据引脚设置

配置举例:

```
[s_rsb0]
s_rsb0_used      = 1
s_rsb0_sck       = port:PL00<2><1><2><default>
s_rsb0_sda       = port:PL01<2><1><2><default>
```

## 34. [s\_jtag0]

配置项	配置项含义
s_jtag0_used=xx	JTAG 使能
s_jtag0_tms=xx	测试模式选择输入(TMS) 的 GPIO 配置
s_jtag0_tck=xx	测试时钟输入(TMS) 的 GPIO 配置
s_jtag0_tdo=xx	测试数据输出(TDO) 的 GPIO 配置
s_jtag0_tdi=xx	测试数据输入 (TDI) 的 GPIO 配置

配置举例:

```
[s_jtag0]
s_jtag0_used          = 0
s_jtag0_tms           = port:PL04<2><1><2><default>
s_jtag0_tck           = port:PL05<2><1><2><default>
s_jtag0_tdo           = port:PL06<2><1><2><default>
s_jtag0_tdi           = port:PL07<2><1><2><default>
```

## 35. mali400

配置项	配置项含义
normal_freq	the normal frequency of gpu
scene_ctrl_status	scene control status, if this is enabled, android layer can ask gpu driver to change frequency in certain scene
temp_ctrl_status	temperature control status, if this is enabled, the gpu frequency will drop down if gpu's temperature is too high

配置举例:

```
[gpu_mali400_0]
normal_freq          = 408
scene_ctrl_status    = 0
temp_ctrl_status     = 1
```

## 36. Declaration

This document 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.

Confidential