

# V40 项目

sys\_config 配置说明文档

## 文档履历

[illegible]

## 说明

1. 蓝色为模块芯片引脚配置，黑色为模块内部控制配置项；
2. 描述 GPIO 配置的形式：  
Port:端口+组内序号<功能分配><内部电阻状态><驱动能力><输出电平状态>
3. 配置举例中的管脚不一定为真实可用的，实际使用时需向技术支持人员询问。

# 目 录

V40 项目 .....	1
sys_config 配置说明文档 .....	1
1. 系统 (SYSTEM) .....	5
1.1. [product] .....	5
1.2. [platform] .....	5
1.3. [target] .....	5
1.4. [power_sply] .....	5
1.5. [card_boot] .....	6
1.6. [pm_para] .....	6
1.7. [card0_boot_para] .....	7
1.8. [card2_boot_para] .....	7
1.9. [twi_para] .....	8
1.10. [uart_para] .....	8
1.11. [jtag_para] .....	8
1.12. [clock] .....	9
2. dram 配置 .....	10
2.1. [dram_para] .....	10
3. I2C 总线 .....	12
3.1. [twi0] .....	12
3.2. [twi1] .....	12
3.3. [twi2] .....	12
3.4. [twi3] .....	12
3.5. [twi4] .....	13
4. 串口 (UART) .....	14
4.1. [uart0] .....	14
4.2. [uart2] .....	14
4.3. [uart3] .....	14
5. SPI 总线 .....	16
5.1. [spi0] .....	16
5.2. [spi1] .....	16
5.3. [spi2] .....	17
5.4. [spi3] .....	17
6. 触摸屏配置 .....	18
6.1. [rtp_para] .....	18
6.2. [acc_gpio] .....	18
6.3. [ctp] .....	18
6.4. [ctp_list] .....	19
7. 触摸按键 .....	21
7.1. [tkey_para] .....	21
8. 马达 .....	22
8.1. [motor_para] .....	22
9. 闪存 (nand flash) .....	23
9.1. [nand0_para] .....	23
10. 显示 .....	25

10.1. [boot_disp] .....	25
10.2. [disp].....	25
10.3. [lcd0] .....	26
10.4. [lcd0_suspend] .....	28
10.5. [car_reverse].....	28
11. PWM.....	30
11.1. [pwm0] .....	30
11.2. [pwm0_suspend].....	30
12. HDMI.....	31
12.1. [hdmi] .....	31
13. tvd 摄像头 .....	32
13.1. [tvd] .....	32
14. 摄像头 .....	34
14.1. [csi0].....	34
14.2. [csi0/csi0_dev0] .....	34
15. tvout/tvin.....	37
15.1. [tvout_para] .....	37
15.2. [tvin_para] .....	37
15.3. [di] .....	37
16. SD/MMC .....	38
16.1. [sdc0].....	38
16.2. [sdc1].....	38
16.3. [sdc2].....	39
16.4. [sdc3].....	41
17. [gpio_para].....	42
18. USB 控制器标志.....	43
18.1. [usbc0].....	43
18.2. [usbc1].....	43
18.3. [usbc2].....	44
19. 重力感应器 .....	45
19.1. [gsensor] .....	45
19.2. [gsensor_list] .....	45
20. WIFI.....	46
20.1. [wlan] .....	46
21. 蓝牙 .....	47
21.1. [bt].....	47
21.2. [btlpm].....	47
22. Gps.....	48
22.1. [gps].....	48
23. gyroscopesensor.....	49
23.1. [gyroscopesensor].....	49
24. 光感(light sensor).....	50
24.1. [lightsensor].....	50
25. 罗盘 Compass.....	51
25.1. [compasssensor] .....	51
26. 数字音频总线（SPDIF） .....	52

26.1. [audiospdif] .....	52
26.2. [spdif_machine] .....	52
26.3. [audiohdm] .....	52
26.4. [hdm_machine] .....	52
26.5. [snddaudio0] .....	52
26.6. [daudio0] .....	52
26.7. [snddaudio1] .....	54
26.8. [daudio1] .....	54
27. Codec 配置 .....	57
27.1. [sndcodec] .....	57
27.2. [codec] .....	57
28. PMU 电源 .....	58
28.1. [pmu0] .....	58
28.2. [charger0] .....	58
28.3. [powerkey0] .....	62
28.4. [regulator0] .....	62
28.5. [axp_gpio0] .....	63
29. Vf 表设置 .....	64
29.1. [dvfs_table] .....	64
30. Virtual device .....	65
30.1. [Vdevice] .....	65
Declaration .....	66

## 1. 系统 (SYSTEM)

### 1.1. [product]

配置项	配置项含义
version = "100"	sdk 版本号 1.00
machine = "magton-p1"	sdk 代号

配置举例：

```
version      = "100"
machine      = "magton-p1"
```

### 1.2. [platform]

配置项	配置项含义
eraseflag	量产时是否擦除。0：不擦，1：擦除（仅仅对量产工具，升级工具无效）
next_work	USB 量产完成后状态。1:不做任何动作 2：重启 3:关机 4：量产

配置举例：

```
eraseflag    = 1
```

### 1.3. [target]

配置项	配置项含义
boot_clock	启动频率(A7启动频率); xx 表示多少 MHZ
storage_type	启动介质选择 0 : nand, 1: card0,2: card2,-1 (default) 自动扫描启动介质:
burn_key	开机启动 usb 烧 key 检测 0: 不启动 1: 启动
dragonboard_test	是否编译支持卡启动的 dragonboard 固件。1: 是 0: 否

配置举例：

```
boot_clock      = 1008
storage_type     = -1
burn_key        = 0
dragonboard_test =0
```

### 1.4. [power\_sply]

配置项	配置项含义
dc1_vol	dc1 模块输出电压
dc2_vol	dc2 模块输出电压
dc3_vol	dc3 模块输出电压
dc4_vol	dc4 模块输出电压
aldo1_vol	aldo1 模块输出电压
aldo2_vol	aldo2 模块输出电压
aldo3_vol	aldo3 模块输出电压
dc1sw_vol	

dc5ldo_vol	
dldo1_vol	dldo1 模块输出电压
dldo2_vol	dldo2 模块输出电压
dldo3_vol	dldo3 模块输出电压
dldo4_vol	dldo4 模块输出电压
eldo1_vol	eldo1 模块输出电压
eldo2_vol	eldo2 模块输出电压
eldo3_vol	eldo3 模块输出电压

配置举例：

```

dc5ldo_vol      = 1003300
dc5ldo_vol      = 1001200
dc5ldo_vol      = 1001100
dc5ldo_vol      = 1100
aldol_vol       = 1003300
aldol_vol       = 1001800
aldol_vol       = 1003000
dc1sw_vol       = 3000
dc5ldo_vol      = 1100
dldol_vol       = 3300
dldol_vol       = 3300
dldol_vol       = 3300
dldol_vol       = 3300
eldo1_vol       = 1200
eldo2_vol       = 1500
eldo3_vol       = 2800

```

## 1.5. [card\_boot]

配置项	配置项含义
logical_start	启动卡逻辑起始扇区
sprite_work_delay	正常卡量产和一键 recovery 指示灯的闪烁间隔
sprite_err_delay	非正常卡量产和一键 recovery 指示灯的闪烁间隔
<a href="#">sprite_gpio0</a>	卡量产，一键 recovery led 指示灯 GPIO 配置
next_work	卡量产完成后状态：1:不做任何动作 2：重启 3:关机 4：量产

配置举例：

```

logical_start    = 40960
sprite_gpio0     =
next_work        = 3

```

## 1.6. [pm\_para]

配置项	配置项含义
standby_mode	1: super standby other: normal standby

配置举例：

```
standby_mode     = 1
```



## 1.7. [card0\_boot\_para]

配置项	配置项含义
card_ctrl	卡量产相关的控制器选择 0
card_high_speed	速度模式 0 为低速, 1 为高速
card_line	4: 4 线卡, 8: 8 线卡
sdc_d1	sdc 卡数据 1 线信号的 GPIO 配置
sdc_d0	sdc 卡数据 0 线信号的 GPIO 配置
sdc_clk	sdc 卡时钟信号的 GPIO 配置
sdc_cmd	sdc 命令信号的 GPIO 配置
sdc_d3	sdc 卡数据 3 线信号的 GPIO 配置
sdc_d2	sdc 卡数据 2 线信号的 GPIO 配置

配置举例:

```

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>

```

## 1.8. [card2\_boot\_para]

配置项	配置项含义
card_ctrl	卡启动控制器选择 2
card_high_speed	速度模式 0 为低速, 1 为高速
card_line	4: 4 线卡, 8: 8 线卡
sdc_clk	sdc 卡时钟信号的 GPIO 配置
sdc_cmd	sdc 命令信号的 GPIO 配置
sdc_d0	sdc 卡数据 0 线信号的 GPIO 配置
sdc_d1	sdc 卡数据 1 线信号的 GPIO 配置
sdc_d2	sdc 卡数据 2 线信号的 GPIO 配置
sdc_d3	sdc 卡数据 3 线信号的 GPIO 配置
sdc_d4	sdc 卡数据 4 线信号的 GPIO 配置
sdc_d5	sdc 卡数据 5 线信号的 GPIO 配置
sdc_d6	sdc 卡数据 6 线信号的 GPIO 配置
sdc_d7	sdc 卡数据 7 线信号的 GPIO 配置
sdc_emmc_rst	
sdc_ds	
sdc_ex_dly_used	
sdc_io_1v8	

配置举例:

```

card_ctrl      = 2

```

```

card_high_speed = 1
card_line       = 8
sdc_clk         = port:PC7<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:PC24<3><1><3><default>
sdc_ds          = port:PC5<3><1><3><default>
sdc_ex_dly_used = 2
sdc_io_1v8      = 1

```

## 1.9. [twi\_para]

配置项	配置项含义
twi_port	Boot 的 twi 控制器编号
<a href="#">twi_scl</a>	Boot 的 twi 的时钟的 GPIO 配置
<a href="#">twi_sda</a>	Boot 的 twi 的数据的 GPIO 配置

配置举例：

```

twi_port      = 0
twi_scl       = port:PB0<2><default><default><default>
twi_sda       = port:PB1<2><default><default><default>

```

## 1.10.[uart\_para]

配置项	配置项含义
uart_debug_port	Boot 串口控制器编号
<a href="#">uart_debug_tx</a>	Boot 串口发送的 GPIO 配置
<a href="#">uart_debug_rx</a>	Boot 串口接收的 GPIO 配置

配置举例：

```

uart_debug_port = 0
uart_debug_tx   = port:PB22<2><1><default><default>
uart_debug_rx   = port:PB23<2><1><default><default>

```

## 1.11.[jtag\_para]

配置项	配置项含义
jtag_enable	JTAG 使能
<a href="#">jtag_ms</a>	测试模式选择输入(TMS) 的 GPIO 配置
<a href="#">jtag_ck</a>	测试时钟输入(TMS) 的 GPIO 配置
<a href="#">jtag_do</a>	测试数据输出(TDO) 的 GPIO 配置
<a href="#">jtag_di</a>	测试数据输入 (TDI) 的 GPIO 配置

配置举例：

```
jtag_enable      = 1
jtag_ms          = port:PB14<3><default><default><default>
jtag_ck          = port:PB15<3><default><default><default>
jtag_do          = port:PB16<3><default><default><default>
jtag_di          = port:PB17<3><default><default><default>
```

1.12.[clock]

配置项	配置项含义
pll4	pll4 时钟频率
pll8	pll8 时钟频率
pll9	pll9 时钟频率
pll12	pll12 时钟频率

配置举例：

```
pll4             = 297
pll8             = 297
pll9             = 297
pll12            = 297
```

## 2. dram 配置

### 2.1. [dram\_para]

配置项	配置项含义
dram_clk	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	DRAM 类型： 2 为 DDR2 3 为 DDR3
dram_zq	DRAM 控制器内部参数，由原厂来进行调节，请勿修改
dram_odt_en	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	DRAM 控制器内部参数，由原厂来进行调节，请勿修改
dram_para2	DRAM 控制器内部参数，由原厂来进行调节，请勿修改
dram_mr0	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	DRAM 控制器内部参数，由原厂来进行调节，请勿修改
dram_mr2	DRAM 控制器内部参数，由原厂来进行调节，请勿修改
dram_mr3	DRAM 控制器内部参数，由原厂来进行调节，请勿修改
dram_tpr0	DRAM 控制器内部参数，由原厂来进行调节，请勿修改
dram_tpr1	DRAM 控制器内部参数，由原厂来进行调节，请勿修改
dram_tpr2	DRAM 控制器内部参数，由原厂来进行调节，请勿修改
dram_tpr3	DRAM 控制器内部参数，由原厂来进行调节，请勿修改
dram_tpr4	DRAM 控制器内部参数，由原厂来进行调节，请勿修改
dram_tpr5	DRAM 控制器内部参数，由原厂来进行调节，请勿修改
dram_tpr6	DRAM 控制器内部参数，由原厂来进行调节，请勿修改
dram_tpr7	DRAM 控制器内部参数，由原厂来进行调节，请勿修改
dram_tpr8	DRAM 控制器内部参数，由原厂来进行调节，请勿修改
dram_tpr9	DRAM 控制器内部参数，由原厂来进行调节，请勿修改
dram_tpr10	DRAM 控制器内部参数，由原厂来进行调节，请勿修改
dram_tpr11	DRAM 控制器内部参数，由原厂来进行调节，请勿修改
dram_tpr12	DRAM 控制器内部参数，由原厂来进行调节，请勿修改
dram_tpr13	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_clk          = 648
dram_type         = 7
dram_zq           = 0x3b3bfb
dram_odt_en       = 0x31
dram_para1        = 0x10e410e4
dram_para2        = 0x1000
dram_mr0          = 0x1840

```

dram_mr1	= 0x40
dram_mr2	= 0x18
dram_mr3	= 0x2
dram_tpr0	= 0x0048A192
dram_tpr1	= 0x01b1a94b
dram_tpr2	= 0x00061043
dram_tpr3	= 0xB47D7D96
dram_tpr4	= 0x0000
dram_tpr5	= 0x198
dram_tpr6	= 0x21000000
dram_tpr7	= 0x2406C1E0
dram_tpr8	= 0x0
dram_tpr9	= 480
dram_tpr10	= 0x0008
dram_tpr11	= 0x44450000
dram_tpr12	= 0x9777
dram_tpr13	= 0x4090950

### 3. I2C 总线

主控有 5 个 I2C（twi）控制器

#### 3.1. [twi0]

配置项	配置项含义
twi0_used	twi 使用控制：1 使用，0 不用
<a href="#">twi0_scl</a>	twi SCK 的 GPIO 配置
<a href="#">twi0_sda</a>	twi SDA 的 GPIO 配置

配置举例：

```
twi0_used      = 1
twi0_scl       = port:PB00<2><default><default><default>
twi0_sda       = port:PB01<2><default><default><default>
```

#### 3.2. [twi1]

配置项	配置项含义
twi1_used	twi 使用控制：1 使用，0 不用
<a href="#">twi1_scl</a>	twi SCK 的 GPIO 配置
<a href="#">twi1_sda</a>	twi SDA 的 GPIO 配置

配置举例：

```
twi1_used      = 1
twi1_scl       = port:PB18<2><default><default><default>
twi1_sda       = port:PB19<2><default><default><default>
```

#### 3.3. [twi2]

配置项	配置项含义
twi2_used	twi 使用控制：1 使用，0 不用
<a href="#">twi2_scl</a>	twi SCK 的 GPIO 配置
<a href="#">twi2_sda</a>	twi SDA 的 GPIO 配置

配置举例：

```
twi2_used      = 1
twi2_scl       = port:PB20<2><default><default><default>
twi2_sda       = port:PB21<2><default><default><default>
```

#### 3.4. [twi3]

配置项	配置项含义
twi3_used	twi 使用控制：1 使用，0 不用
<a href="#">twi3_scl</a>	twi SCK 的 GPIO 配置
<a href="#">twi3_sda</a>	twi SDA 的 GPIO 配置

配置举例：

```
twi3_used      = 1
twi3_scl       = port:PI0<3><default><default><default>
```

```
twi3_sda          = port:PI1<3><default><default><default>
```

### 3.5. [twi4]

配置项	配置项含义
twi4_used	twi 使用控制：1 使用，0 不用
<a href="#">twi4_scl</a>	twi SCK 的 GPIO 配置
<a href="#">twi4_sda</a>	twi SDA 的 GPIO 配置

配置举例：

```
twi4_used          = 0
twi4_scl           = port:PI2<3><default><default><default>
twi4_sda           = port:PI3<3><default><default><default>
```

## 4. 串口(UART)

### 4.1. [uart0]

配置项	配置项含义
uart0_used	UART 使用控制：1 使用，0 不用
uart0_port	UART 端口号
uart0_type	UART 类型，有效值为：2/4/8; 2：2 线模式; 4：4 线模式; 8：8 线模式。
uart0_tx	UART TX 的 GPIO 配置
uart0_rx	UART RX 的 GPIO 配置

配置举例：

```

uart0_used      = 1
uart0_port      = 0
uart0_type      = 2
uart0_tx        = port:PB22<2><1><default><default>
uart0_rx        = port:PB23<2><1><default><default>

```

### 4.2. [uart2]

配置项	配置项含义
uart3_used	UART 使用控制：1 使用，0 不用
uart3_port	UART 端口号
uart3_type	UART 类型，有效值为：2/4/8; 2：2 线模式; 4：4 线模式; 8：8 线模式。
uart3_tx	UART TX 的 GPIO 配置
uart3_rx	UART RX 的 GPIO 配置
uart3_rts	UART RTS 的 GPIO 配置
uart3_cts	UART CTS 的 GPIO 配置

配置举例：

```

uart3_used      = 1
uart3_port      = 2
uart3_type      = 4
uart3_tx        = port:PI18<3><1><default><default>
uart3_rx        = port:PI19<3><1><default><default>
uart3_cts       = port:PI17<3><1><default><default>
uart3_rts       = port:PI16<3><1><default><default>

```

### 4.3. [uart3]

配置项	配置项含义
-----	-------



uart3_used	UART 使用控制：1 使用，0 不用
uart3_port	UART 端口号
uart3_type	UART 类型，有效值为：2/4/8; 2：2 线模式; 4：4 线模式; 8：8 线模式
uart3_tx	UART TX 的 GPIO 配置
uart3_rx	UART RX 的 GPIO 配置
uart3_rts	UART RTS 的 GPIO 配置
uart3_cts	UART CTS 的 GPIO 配置

配置举例：

[uart3]

uart3\_used = 1

## 5. SPI 总线

### 5.1. [spi0]

配置项	配置项含义
spi0_used	SPI 使用控制：1 使用，0 不用
spi0_cs_number	
spi0_cs_bitmap	由于 SPI 控制器支持多个 CS，这一个参数表示 CS 的掩码；
spi0_cs0	SPI CS0 的 GPIO 配置
spi0_cs1	SPI CS1 的 GPIO 配置
spi0_sclk	SPI CLK 的 GPIO 配置
spi0_mosi	SPI MOSI 的 GPIO 配置
spi0_miso	SPI MISO 的 GPIO 配置

配置举例：

```

spi0_used      = 0
spi0_cs_number = 2
spi0_cs_bitmap = 3
spi0_cs0       = port:PC23<3><1><default><default>
spi0_cs1       = port:PI14<2><1><default><default>
spi0_sclk      = port:PC2<3><default><default><default>
spi0_mosi      = port:PC0<3><default><default><default>
spi0_miso      = port:PC1<3><default><default><default>s

```

### 5.2. [spi1]

配置项	配置项含义
spi1_used	SPI 使用控制：1 使用，0 不用
spi1_cs_number	
spi1_cs_bitmap	由于 SPI 控制器支持多个 CS，这一个参数表示 CS 的掩码；
spi1_cs0	SPI CS0 的 GPIO 配置
spi1_cs1	SPI CS1 的 GPIO 配置
spi1_sclk	SPI CLK 的 GPIO 配置
spi1_mosi	SPI MOSI 的 GPIO 配置
spi1_miso	SPI MISO 的 GPIO 配置

配置举例：

```

spi1_used      = 0
spi1_cs_number = 2
spi1_cs_bitmap = 3
spi1_cs0       = port:PA0<3><1><default><default>
spi1_cs1       = port:PA4<3><1><default><default>
spi1_sclk      = port:PA1<3><default><default><default>
spi1_mosi      = port:PA2<3><default><default><default>
spi1_miso      = port:PA3<3><default><default><default>

```

### 5.3. [spi2]

配置项	配置项含义
spi0_used	SPI 使用控制：1 使用，0 不用
spi0_cs_number	
spi0_cs_bitmap	由于 SPI 控制器支持多个 CS，这一个参数表示 CS 的掩码；
spi0_cs0	SPI CS0 的 GPIO 配置
spi0_cs1	SPI CS1 的 GPIO 配置
spi0_sclk	SPI CLK 的 GPIO 配置
spi0_mosi	SPI MOSI 的 GPIO 配置
spi0_miso	SPI MISO 的 GPIO 配置

配置举例：

```

spi2_used      = 0
spi2_cs_number = 2
spi2_cs_bitmap = 3
spi2_cs0       = port:PB14<2><1><default><default>
spi2_cs1       = port:PB13<2><1><default><default>
spi2_sclk      = port:PB15<2><default><default><default>
spi2_mosi      = port:PB16<2><default><default><default>
spi2_miso      = port:PB17<2><default><default><default>

```

### 5.4. [spi3]

配置项	配置项含义
spi0_used	SPI 使用控制：1 使用，0 不用
spi0_cs_number	
spi0_cs_bitmap	由于 SPI 控制器支持多个 CS，这一个参数表示 CS 的掩码；
spi0_cs0	SPI CS0 的 GPIO 配置
spi0_cs1	SPI CS1 的 GPIO 配置
spi0_sclk	SPI CLK 的 GPIO 配置
spi0_mosi	SPI MOSI 的 GPIO 配置
spi0_miso	SPI MISO 的 GPIO 配置

配置举例：

```

spi3_used      = 0
spi3_cs_number = 2
spi3_cs_bitmap = 3
spi3_cs0       = port:PA5<3><1><default><default>
spi3_cs1       = port:PA9<3><1><default><default>
spi3_sclk      = port:PA6<3><default><default><default>
spi3_mosi      = port:PA7<3><default><default><default>
spi3_miso      = port:PA8<3><default><default><default>

```

## 6. 触摸屏配置

### 6.1. [rtp\_para]

配置项	配置项含义
rtp_used	该模块在方案中是否启用，
rtp_screen_size	屏幕尺寸设置，以斜对角方向长度为准，以寸为单位
rtp_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	是否开启压力的们门限制，建议选 0 不开启
rtp_press_threshold	这配置项当 rtp_press_threshold_enable 为 1 时才有效，其数值可以是 0 到 0xFFFFFFFF 的任意数值，数值越小越敏感，推荐值为 0xF
rtp_sensitive_level	敏感等级，数值可以是 0 到 0xF 之间的任意数值，数值越大越敏感，0xF 为推荐值
rtp_exchange_x_y_flag	当屏的 x,y 轴需要转换的时候，这个项目该置 1，一般情况下则该置 0

配置举例：

```

rtp_used                = 0
rtp_screen_size         = 5
rtp_regidity_level      = 5
rtp_press_threshold_enable = 0
rtp_press_threshold      = 0x1f40
rtp_sensitive_level     = 0xf
rtp_exchange_x_y_flag   = 0

```

### 6.2. [acc\_gpio]

配置项	配置项含义
compatible	设备名字
acc_gpio_used	改选项是否开启，1：开启，0：关闭
acc_int	acc gpio 配置引脚，用作判断是否需要进入睡眠

配置举例：

```

compatible              = "allwinner,sunxi-acc-det"
acc_gpio_used           = 1
acc_int                 = port:power0<6><default><default><default>

```

### 6.3. [ctp]

配置项	配置项含义
compatible	设备名字
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	用作区别 gd 系列，适配屏幕分辨率
ctp_gesture_wakeup	

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	触摸屏电源配置
ctp_power_ldo_vol	触摸屏电压配置
ctp_power_io	电源使能脚

配置举例:

```

compatible      = "allwinner,sun50i-ctp-para"
ctp_used        = 1
ctp_name        = "gt911_784"
ctp_gesture_wakeup = 0
ctp_twi_id      = 3
ctp_twi_addr    = 0x5d
ctp_screen_max_x = 400
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:PH15<6><default><default><default>
ctp_wakeup      = port:PH14<1><default><default><1>

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

```

## 6.4. [ctp\_list]

配置项	配置项含义
compatible	配置名称
ctp_list_used	支持触摸屏 list
ft5x_ts	是否支持 ft5x_ts 模组
gt82x	是否支持 gt82x 模组
gsIX680	是否支持 gsIX680 模组
gt9xx_ts	是否支持 gt9xx_ts 模组
gt9xxnew_ts	是否支持 gt9xxnew_ts 模组
gt811	是否支持 gt811 模组
zet622x	是否支持 zet622x 模组
aw5306_ts	是否支持 aw5306_ts 模组

配置举例：

```
compatible          = "allwinner,sun50i-ctp-list"
ctp_list_used       = 1
gt82x               = 1
gt9xxnew_ts         = 1
```

## 7. 触摸按键

### 7.1. [tkey\_para]

配置项	配置项含义
tkey_used	支持触摸按键的置 1，反之置 0
tkey_twi_id	用于选择 i2c adapter，可选 1，2
tkey_twi_addr	指明 i2c 设备地址，与具体硬件相关
tkey_int	触摸按键中断信号的 GPIO 配置

配置举例：

```
tkey_used          = 0
tkey_twi_id        =
tkey_twi_addr      =
tkey_int           =
```

## 8. 马达

### 8.1. [motor\_para]

配置项	配置项含义
motor_used	是否启用马达，启用置 1，反之置 0
motor_shake	马达使用的 GPIO 配置

配置举例：

motor\_para\_used = 0

motor\_shake = port:power3<1><default><default><1>



## 9. 闪存(nand flash)

### 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	nand0 模块使能标志
<a href="#">nand0_we</a>	nand0 写时钟信号的 GPIO 配置
<a href="#">nand0_ale</a>	nand0 地址使能信号的 GPIO 配置
<a href="#">nand0_cle</a>	nand0 命令使能信号的 GPIO 配置
<a href="#">nand0_cel</a>	nand0 片选 1 信号的 GPIO 配置
<a href="#">nand0_ce0</a>	nand0 片选 0 信号的 GPIO 配置
<a href="#">nand0_nre</a>	nand0 读时钟信号的 GPIO 配置
<a href="#">nand0_rb0</a>	nand0 Read/Busy 1 信号的 GPIO 配置
<a href="#">nand0_rb1</a>	nand0 Read/Busy 0 信号的 GPIO 配置
<a href="#">nand0_d0</a>	nand0 数据总线信号的 GPIO 配置
<a href="#">nand0_d1</a>	/
<a href="#">nand0_d2</a>	/
<a href="#">nand0_d3</a>	/
<a href="#">nand0_d4</a>	/
<a href="#">nand0_d5</a>	/
<a href="#">nand0_d6</a>	/
<a href="#">nand0_d7</a>	/
<a href="#">nand0_nwp</a>	
<a href="#">nand0_ce2</a>	nand0 片选 2 信号的 GPIO 配置
<a href="#">nand0_ce3</a>	nand0 片选 3 信号的 GPIO 配置
<a href="#">nand0_ce4</a>	
<a href="#">nand0_ce5</a>	
<a href="#">nand0_ce6</a>	
<a href="#">nand0_ce7</a>	
<a href="#">nand0_ndqs</a>	
<a href="#">nand0_regulator1</a>	
<a href="#">nand0_regulator2</a>	
<a href="#">nand0_cache_level</a>	
<a href="#">nand0_flush_cache_num</a>	
<a href="#">nand0_capacity_level</a>	
<a href="#">nand0_id_number_ctl</a>	
<a href="#">nand0_print_level</a>	
<a href="#">nand0_p0</a>	
<a href="#">nand0_p1</a>	
<a href="#">nand0_p2</a>	
<a href="#">nand0_p3</a>	

配置举例：

```

[nand0_para]
nand0_support_2ch      = 0

nand0_used             = 1
nand0_we               = port:PC00<2><0><1><default>
nand0_ale              = port:PC01<2><0><1><default>
nand0_cle              = port:PC02<2><0><1><default>
nand0_ce1              = port:PC03<2><1><1><default>
nand0_ce0              = port:PC04<2><1><1><default>
nand0_nre              = port:PC05<2><0><1><default>
nand0_rb0              = port:PC06<2><1><1><default>
nand0_rb1              = port:PC07<2><1><1><default>
nand0_d0               = port:PC08<2><0><1><default>
nand0_d1               = port:PC09<2><0><1><default>
nand0_d2               = port:PC10<2><0><1><default>
nand0_d3               = port:PC11<2><0><1><default>
nand0_d4               = port:PC12<2><0><1><default>
nand0_d5               = port:PC13<2><0><1><default>
nand0_d6               = port:PC14<2><0><1><default>
nand0_d7               = port:PC15<2><0><1><default>
nand0_nwp              = port:PC16<2><1><1><default>
nand0_ce2              = port:PC17<2><1><1><default>
nand0_ce3              = port:PC18<2><1><1><default>
nand0_ce4              = port:PC19<2><1><1><default>
nand0_ce5              = port:PC20<2><1><1><default>
nand0_ce6              = port:PC21<2><1><1><default>
nand0_ce7              = port:PC22<2><1><1><default>
nand0_ndqs             = port:PC24<2><0><1><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.[boot\_disp]

配置项	配置项含义
advert_disp	支持显示用户自定义 bootlogo

配置具体：

```
advert_disp = 0
```

### 10.2.[disp]

配置项	配置项含义
disp_init_enable	是否进行显示的初始化设置
disp_mode	显示模式： 0:screen0<screen0,fb0> 1: screen1<screen1,fb0>
screen0_output_type	屏 0 输出类型(0:none; 1:lcd; 2:tv; 3:hdmi; 4:vga)
screen0_output_mode	屏 0 输出模式(used for tv/hdmi output, 0:480i 1:576i 2:480p 3:576p 4:720p50 5:720p60 6:1080i50 7:1080i60 8:1080p24 9:1080p50 10:1080p60 11:pal 14:ntsc)
screen1_output_type	屏 1 输出类型(0:none; 1:lcd; 2:tv; 3:hdmi; 4:vga)
screen1_output_mode	屏 1 输出模式(used for tv/hdmi output, 0:480i 1:576i 2:480p 3:576p 4:720p50 5:720p60 6:1080i50 7:1080i60 8:1080p24 9:1080p50 10:1080p60 11:pal 14:ntsc)
fb0_format	fb0 的格式(0:ARGB 1:ABGR 2:RGBA 3:BGR)
fb0_width	fb0 的宽度,为 0 时将按照输出设备的分辨率
fb0_height	fb0 的高度, 为 0 时将按照输出设备的分辨率
fb1_format	fb1 的格式(0:ARGB 1:ABGR 2:RGBA 3:BGR)
fb1_width	Fb1 的宽度,为 0 时将按照输出设备的分辨率
fb1_height	Fb1 的高度, 为 0 时将按照输出设备的分辨率

配置举例：

```
disp_init_enable = 1
```

```
disp_mode = 0
```

```
screen0_output_type = 3
```

```
screen0_output_mode = 4
```

```
screen1_output_type = 2
```

```
screen1_output_mode = 11
```

```
fb0_format = 0
```

```
fb0_width = 1280
```

```
fb0_height = 720
```

```
fb1_format = 0
```

```
fb1_width = 0
```

fb1\_height = 0

### 10.3.[lcd0]

配置项	
lcd_used	是否使用 lcd0。 1：使用；0：不适用
lcd_driver_name	定义驱动名称
lcd_backlight	LCD 背光值
lcd_if	lcd 接口(0:hv(sync+de); 1:8080; 2:ttl; 3:lvds, 4:dsi; 5:edp)
lcd_x	LCD 分辨率 x
lcd_y	LCD 分辨率 y
lcd_width	LCD 屏宽度
lcd_height	LCD 屏高度
lcd_dclk_freq	LCD 频率
lcd_pwm_used	Pwm 是否使用
lcd_pwm_ch	Pwm 通道
lcd_pwm_freq	Pwm 频率
lcd_pwm_pol	pwm 属性, 0:positive; 1:negative
lcd_pwm_max_limit	Pwm 最大值
lcd_hbp	LCD 行后沿时间
lcd_ht	LCD 行时间
lcd_hspw	LCD 行同步脉宽
lcd_vbp	LCD 场后沿时间
lcd_vt	LCD 场时间
lcd_vspw	LCD 场同步脉宽
lcd_lvds_if	lvds 接口, 0:single link; 1:dual link
lcd_lvds_colordepth	lvds 颜色深度 0:8bit; 1:6bit
lcd_lvds_mode	lvds 模式, 0:NS mode; 1:JEIDA mode
lcd_frm	LCD 格式, 0:disable; 1:enable rgb666 dither; 2:enable rgb656 dither
lcd_hv_clk_phase	hv 时钟相位: 0:0 degree; 1:90 degree; 2: 180 degree; 3: 270 degree
lcd_hv_sync_polarity	io 属性, 0:not invert; 1:invert
lcd_gamma_en	gamma 校正使能
lcd_bright_curve_en	LCD 亮度曲线校正使能
lcd_cmap_en	LCD 调色板函数使能
lcd_bl_en	背光使能的 GPIO 配置
lcd_bl_en_power	背光使用管脚配置
lcd_power	LCD 电源
lcd_fix_power	LCD 电源
lcdd0	LCD 数据 0 线信号的 GPIO 配置
Lcdd1	LCD 数据 1 线信号的 GPIO 配置
Lcdd2	LCD 数据 2 线信号的 GPIO 配置
Lcdd3	LCD 数据 3 线信号的 GPIO 配置
Lcdd4	LCD 数据 4 线信号的 GPIO 配置
Lcdd5	LCD 数据 5 线信号的 GPIO 配置

Lcdd6	LCD 数据 6 线信号的 GPIO 配置
Lcdd7	LCD 数据 7 线信号的 GPIO 配置
Lcdd8	LCD 数据 8 线信号的 GPIO 配置
Lcdd9	LCD 数据 9 线信号的 GPIO 配置
lcd_pin_power	LCD 所使用的 GPIO 的供电电源

配置举例：

```
lcd_used = 0
```

```
lcd_driver_name = "default_lcd"
```

```
lcd_backlight = 50
```

```
lcd_if = 3
```

```
lcd_x = 1280
```

```
lcd_y = 800
```

```
lcd_width = 150
```

```
lcd_height = 94
```

```
lcd_dclk_freq = 70
```

```
lcd_pwm_used = 1
```

```
lcd_pwm_ch = 0
```

```
lcd_pwm_freq = 50000
```

```
lcd_pwm_pol = 1
```

```
lcd_pwm_max_limit = 255
```

```
lcd_hbp = 20
```

```
lcd_ht = 1418
```

```
lcd_hspw = 10
```

```
lcd_vbp = 10
```

```
lcd_vt = 814
```

```
lcd_vspw = 5
```

```
lcd_lvds_if = 0
```

```
lcd_lvds_colordepth = 1
```

```
lcd_lvds_mode = 0
```

```
lcd_frm = 1
```

```
lcd_hv_clk_phase = 0
```

```
lcd_hv_sync_polarity = 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-lcd-0"
```

```
lcd_fix_power = "vcc-dsi-33"
```

```
lcdd0 = port:PD12<3><0><default><default>
```

```
lcdd1 = port:PD13<3><0><default><default>
```

```
lcdd2 = port:PD14<3><0><default><default>
```

```

lcdd3          = port:PD15<3><0><default><default>
lcdd4          = port:PD16<3><0><default><default>
lcdd5          = port:PD17<3><0><default><default>
lcdd6          = port:PD18<3><0><default><default>
lcdd7          = port:PD19<3><0><default><default>
lcdd8          = port:PD20<3><0><default><default>
lcdd9          = port:PD21<3><0><default><default>
lcd_pin_power  = "vcc-pd"

```

## 10.4.[lcd0\_suspend]

配置项	配置项含义
lcdd0	LCD 数据 0 线信号休眠状态下的 GPIO 配置
lcdd1	LCD 数据 1 线信号休眠状态下的 GPIO 配置
lcdd2	LCD 数据 2 线信号休眠状态下的 GPIO 配置
lcdd3	LCD 数据 3 线信号休眠状态下的 GPIO 配置
lcdd4	LCD 数据 4 线信号休眠状态下的 GPIO 配置
lcdd5	LCD 数据 5 线信号休眠状态下的 GPIO 配置
lcdd6	LCD 数据 6 线信号休眠状态下的 GPIO 配置
lcdd7	LCD 数据 7 线信号休眠状态下的 GPIO 配置
lcdd8	LCD 数据 8 线信号休眠状态下的 GPIO 配置
lcdd9	LCD 数据 9 线信号休眠状态下的 GPIO 配置

配置举例：

```

lcdd0          = port:PD12<7><0><default><default>
lcdd1          = port:PD13<7><0><default><default>
lcdd2          = port:PD14<7><0><default><default>
lcdd3          = port:PD15<7><0><default><default>
lcdd4          = port:PD16<7><0><default><default>
lcdd5          = port:PD17<7><0><default><default>
lcdd6          = port:PD18<7><0><default><default>
lcdd7          = port:PD19<7><0><default><default>
lcdd8          = port:PD20<7><0><default><default>
lcdd9          = port:PD21<7><0><default><default>

```

## 10.5.[car\_reverse]

配置项	配置项含义
compatible	匹配设备的 token
used	模块使用配置项
tv_d_id	倒车模块使用的 tv_d 通道
screen_width	倒车预览图像宽度
screen_height	倒车预览图像高度
rotation	是否使能旋转
reverse_pin	倒车信号输入管脚

配置举例：

[car\_reverse]

```
compatible      = "allwinner,sunxi-car-reverse"  
used            = 1  
tvd_id         = 0  
screen_width   = 720  
screen_height  = 480  
rotation       = 1  
reverse_pin    = port:PH20<6><0><default><default>
```

## 11.PWM

### 11.1.[pwm0]

配置项	配置项含义
pwm_used	是否使用 PWM0
pwm_positive	PWM 输出 GPIO 配置

配置举例：

```
pwm_used          = 1
pwm_positive      = port:PB2<3><0><default><default>
```

### 11.2.[pwm0\_suspend]

配置项	配置项含义
pwm_positive	PWM 输出 GPIO 配置

配置举例：

```
[pwm0_suspend]
pwm_positive      = port:PB2<7><0><default><default>
```



## 12.HDMI

### 12.1.[hdmi]

配置项	配置项含义
hdmi_used	是否使用 hdmi。1： 使用； 0： 不使用
hdmi_power	内核阶段 hdmi 电源配置
hdmi_hdcp_enable	是否使能 hdcp
hdmi_cts_compatibility	cts 兼容性使能设置
hdmi_hpd_mask	hpd 掩码设置
hdmi_cec_support	是否支持 CEC 功能

配置举例：

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

## 13.tvd 摄像头

### 13.1.[tvd]

配置项	配置项含义
tvd_used	是否使用 TVD。1：使用；0：不使用
tvd_if	tvd interface 0:CVBS , 1:YPBPRI , 2: YPBPRP
fliter_used	使能 3D 滤波功能，设置为 1
cagc_enable	使能 cagc 功能，设置为 1
agc_auto_enable	使能 agc 功能，设置为 1
tvd_power0	AXP power，具体参考原理图配置
tvd_hot_plug	支持 TVD 动态插拔功能， 1 to enable hot plug function , 0 to disable ,default disable
tvd_gpio0	gpio control power output or not

配置举例：

[tvd0]

tvd\_used = 1

tvd\_if = 0

fliter\_used = 1

cagc\_enable = 1

agc\_auto\_enable = 1

[tvd1]

tvd\_used = 0

tvd\_if = 0

fliter\_used = 1

cagc\_enable = 1

agc\_auto\_enable = 1

[tvd2]

tvd\_used = 0

tvd\_if = 0

fliter\_used = 1

cagc\_enable = 1

agc\_auto\_enable = 1

[tvd3]

tvd\_used = 0

tvd\_if = 0

fliter\_used = 1

cagc\_enable = 1

agc\_auto\_enable = 1

```
[tvd]
tvd_hot_plug = 1
tvd_power0 = "vcc-tvd"
tvd_gpio0 = port:PH27<1><0><default><1>
```

## 14.摄像头

### 14.1.[csi0]

配置项	配置项含义
csi0_used	摄像头使能配置
csi0_pck	pclk 信号的 GPIO 配置。
csi0_mck	mclk 信号的 GPIO 配置。
csi0_hsync	hsync 信号的 GPIO 配置
csi0_vsync	vsync 信号的 GPIO 配置
csi0_d0	csi d0 信号的 GPIO 配置
csi0_d1	csi d1 信号的 GPIO 配置
csi0_d2	csi d2 信号的 GPIO 配置
csi0_d3	csi d3 信号的 GPIO 配置
csi0_d4	csi d4 信号的 GPIO 配置
csi0_d5	csi d5 信号的 GPIO 配置
csi0_d6	csi d6 信号的 GPIO 配置
csi0_d7	csi d7 信号的 GPIO 配置

配置举例：

```

csi0_used           = 1
csi0_pck            = port:PE00<3><default><default><default>
csi0_mck            = port:PE01<1><0><1><0>
csi0_hsync          = port:PE02<3><default><default><default>
csi0_vsync          = port:PE03<3><default><default><default>
csi0_d0             = port:PE04<3><default><default><default>
csi0_d1             = port:PE05<3><default><default><default>
csi0_d2             = port:PE06<3><default><default><default>
csi0_d3             = port:PE07<3><default><default><default>
csi0_d4             = port:PE08<3><default><default><default>
csi0_d5             = port:PE09<3><default><default><default>
csi0_d6             = port:PE10<3><default><default><default>
csi0_d7             = port:PE11<3><default><default><default>
csi0_sck            = port:PE12<3><default><default><default>
csi0_sda            = port:PE13<3><default><default><default>

```

### 14.2.[csi0/csi0\_dev0]

配置项	配置项含义
csi0_dev0_used	是否使用 csi0_dev0
csi0_dev0_mname	设置 sensor 0 名称，如 “ov2710_aw6131 “。
csi0_dev0_twi_id	请参考实际原理图填写
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	YUV 填 0。
csi0_dev0_fmt	YUV 填 0。
csi0_dev0_stby_mode	填 0。
csi0_dev0_vflip	Sensor 图像垂直翻转。
csi0_dev0_hflip	Sensor 图像水平翻转。
csi0_dev0_iovdd	IOVDD 配置，请参考实际原理图填写。
csi0_dev0_iovdd_vol	IOVDD 电压值一般为 2.8V(2800000)。
csi0_dev0_avdd	AVDD 配置，如"csi-avdd"。
csi0_dev0_avdd_vol	AVDD 电压值，一般为 2.8V(2800000)。
csi0_dev0_dvdd	DVDD 配置，如"csi-dvdd"。
csi0_dev0_dvdd_vol	DVDD 电压值参考 datasheet，1.2/1.5/1.8V。
csi0_dev0_afvdd	Isp-dvdd 配置，如 isp-dvdd12
csi0_dev0_afvdd_vol	电压值为 1.2V。
csi0_dev0_power_en	Sensor power enable 引脚 GPIO 配置。
csi0_dev0_reset	Sensor reset 引脚 GPIO 配置。
csi0_dev0_pwdn	Sensor power down 引脚 GPIO 配置。
csi0_dev0_flash_used	填 0。
csi0_dev0_flash_type	填 0。
csi0_dev0_flash_en	不需填写
csi0_dev0_flash_mode	不需填写
csi0_dev0_flvdd	不需填写
csi0_dev0_flvdd_vol	不需填写
不需填写	不需填写
csi0_dev0_act_used	不需填写
csi0_dev0_act_name	不需填写
csi0_dev0_act_slave	不需填写

配置举例：

```
[csi0/csi0_dev0]
csi0_dev0_used      = 1
csi0_dev0_mname     = "ov2710_aw6131"
csi0_dev0_twi_addr  = 0x36    ;请参考实际模组的 8bit ID 填写
csi0_dev0_twi_id    = 1
csi0_dev0_pos       = "rear"
csi0_dev0_isp_used  = 0    ; 必须填 0
csi0_dev0_fmt       = 0    ; YUV 格式必须填 0
csi0_dev0_stby_mode = 0
csi0_dev0_vflip     = 0
csi0_dev0_hflip     = 0
csi0_dev0_iovdd     = "csi-iovdd" ;电源请参考实际原理图填写
csi0_dev0_iovdd_vol = 2800000 ;电压值非特殊 sensor 一般为 2.8V
csi0_dev0_avdd      = "csi-avdd"    ;电源请参考实际原理图填写
csi0_dev0_avdd_vol  = 3300000; 电压值非特殊 sensor 一般为 3.3V
csi0_dev0_dvdd      = "csi-dvdd"    ;电源请参考实际原理图填写
csi0_dev0_dvdd_vol  = 1500000    ;电压值非特殊 sensor 一般为 1.5V
```

```

csi0_dev0_afvdd      = "isp-dvdd12" ;电源请参考实际原理图填写
csi0_dev0_afvdd_vol  = 1200000  ;电压值非特殊 sensor 一般为 1.2V
csi0_dev0_power_en   =
csi0_dev0_reset      = port:PH17<1><0><1><0>
csi0_dev0_pwdn       =
csi0_dev0_flash_used = 0
csi0_dev0_flash_type = 2
csi0_dev0_flash_en   =
csi0_dev0_flash_mode =
csi0_dev0_flvdd      = ""
csi0_dev0_flvdd_vol  =
csi0_dev0_af_pwdn    =
csi0_dev0_act_used   = 0
csi0_dev0_act_name   =
csi0_dev0_act_slave  =

```

## 15.tvout/tvin

### 15.1.[tvout\_para]

配置项	配置项含义
tvout_used	是否使用 tvout。 1：使用 0：不使用
tvout_channel_num	使用的 tvout 通道号
tv_en	tvout 通道使能

配置举例：

```
tvout_used          = 0
tvout_channel_num   =
tv_en               =
```

### 15.2.[tvin\_para]

配置项	配置项含义
tvin_used	是否使用 tvint。 1：使用 0：不使用
tvin_channel_num	使用的 tvin 通道号

配置举例：

```
tvin_used          = 0
tvin_channel_num   =
```

### 15.3.[di]

配置项	配置项含义
di_used	是否使用反交错。1：使用 0：不使用

配置举例：

```
di_used            = 1
```

## 16.SD/MMC

### 16.1.[sdc0]

配置项	配置项含义
sdc0_used	SDC 使用控制：1 使用，0 不用
bus-width	位宽：1-1bit，4-4bit
sdc0_d1	SDC DATA1 的 GPIO 配置
sdc0_d0	SDC DATA0 的 GPIO 配置
sdc0_clk	SDC CLK 的 GPIO 配置
sdc0_cmd	SDC CMD 的 GPIO 配置
sdc0_d3	SDC DATA3 的 GPIO 配置
sdc0_d2	SDC DATA2 的 GPIO 配置
cd-gpios	SDC 卡检测信号的 GPIO 配置
sunxi-power-save-mode	SDC CLK 信号无数据传输时暂停
vmmc	SDC 供电电源配置
vqmmc	SDC IO 供电电源配置
vdmmc	SDC 卡检测信号上拉电阻的电源配置

举例说明：

```

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:PH18<6><1><2><default>
;wp-gpios          = port:PG01<0><1><2><default>
;wp-inverted        =
broken-cd          =
sunxi-power-save-mode =
sunxi-dis-signal-vol-sw =
vmmc="vcc-sdcv"
vqmmc="vcc-sdcvq33"
vdmmc="vcc-sdcvd"

```

### 16.2.[sdc1]

配置项	配置项含义
sdc1_used	SDC 使用控制：1 使用，0 不用
bus-width	位宽：1-1bit，4-4bit
sdc1_d1	SDC DATA1 的 GPIO 配置
sdc1_d0	SDC DATA0 的 GPIO 配置



<a href="#">sdc1_clk</a>	SDC CLK 的 GPIO 配置
<a href="#">sdc1_cmd</a>	SDC CMD 的 GPIO 配置
<a href="#">sdc1_d3</a>	SDC DATA3 的 GPIO 配置
<a href="#">sdc1_d2</a>	SDC DATA2 的 GPIO 配置
sunxi-power-save-mode	SDC CLK 信号无数据传输时暂停
sd-uhs-sdr50	支持 SDR50 速度模式
sd-uhs-ddr50	支持 DDR50 速度模式
sd-uhs-sdr104	支持 SDR104 速度模式
cap-sdio-irq	目前只用于 SDIO WIFI 驱动，表示控制器支持 SDIO 中断。
keep-power-in-suspend	目前只用于 SDIO WIFI 驱动，表示休眠时器件供电保持不变
ignore-pm-notify	目前只用于 SDIO WIFI 驱动，表示休眠唤醒时忽略内核的 pm notify
max-frequency	最高接口配置频率配置

举例说明：

```

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>
;sunxi-power-save-mode =
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	SDC 使用控制：1 使用，0 不用
non-removable	SDC 连接 Device 具备不可移除属性
bus-width	位宽：1-1bit，4-4bit
sdc2_ds	SDC eMMC Data Strobe 的 GPIO 配置
<a href="#">sdc2_d1</a>	SDC DATA1 的 GPIO 配置
<a href="#">sdc2_d0</a>	SDC DATA0 的 GPIO 配置
<a href="#">sdc2_clk</a>	SDC CLK 的 GPIO 配置
<a href="#">sdc2_cmd</a>	SDC CMD 的 GPIO 配置
<a href="#">sdc2_d3</a>	SDC DATA3 的 GPIO 配置
<a href="#">sdc2_d2</a>	SDC DATA2 的 GPIO 配置
<a href="#">sdc2_d4</a>	SDC DATA4GPIO 配置

<a href="#">sdc2_d5</a>	SDC DATA5 GPIO 配置
<a href="#">sdc2_d6</a>	SDC DATA6 GPIO 配置
<a href="#">sdc2_d7</a>	SDC DATA7 GPIO 配置
sdc2_emmc_rst	SDC eMMC Hardware Reset 的 GPIO 配置
cd-gpios	SDC 卡检测信号的 GPIO 配置
sunxi-power-save-mode	SDC CLK 信号无数据传输时暂停
sunxi-dis-signal-vol-sw	MMC 驱动支持开关 IO 电压但不修改 IO 电压值
sdc_tm4_sm0_freq0	SDC 采样参数控制
sdc_tm4_sm0_freq1	SDC 采样参数控制
sdc_tm4_sm1_freq0	SDC 采样参数控制
sdc_tm4_sm1_freq1	SDC 采样参数控制
sdc_tm4_sm2_freq0	SDC 采样参数控制
sdc_tm4_sm2_freq1	SDC 采样参数控制
sdc_tm4_sm3_freq0	SDC 采样参数控制
sdc_tm4_sm3_freq1	SDC 采样参数控制
sdc_tm4_sm4_freq0	SDC 采样参数控制
sdc_tm4_sm4_freq1	SDC 采样参数控制
vmmc	SDC 供电电源配置
vqmmc	SDC IO 供电电源配置
vdmmc	SDC 卡检测信号上拉电阻的电源配置

配置举例：

```

sdc2_used          = 1
non-removable     =
bus-width         = 8
sdc2_ds           = port:PC05<3><1><3><default>
sdc2_clk          = port:PC07<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:PC24<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      = 150000000
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-emmcv"
;vqmmc="vcc-emmcvq33"
vqmmc="vcc-emmcvq18"
vdmmc="none"

```

## 16.4.[sdc3]

```

sdc3_used          = 1
bus-width          = 4
sdc3_clk            = port:PI05<2><1><2><default>
sdc3_cmd            = port:PI04<2><1><2><default>
sdc3_d0             = port:PI06<2><1><2><default>
sdc3_d1             = port:PI07<2><1><2><default>
sdc3_d2             = port:PI08<2><1><2><default>
sdc3_d3             = port:PI09<2><1><2><default>

cd-gpios            = port:PH19<6><1><2><default>
sunxi-power-save-mode =
sunxi-dis-signal-vol-sw =
;mmc-ddr-1_8v       =
;mmc-hs200-1_8v     =
;mmc-hs400-1_8v     =
;max-frequency      = 150000000
vmmc="vcc-sdcv-p3"
vqmmc="vcc-sdcvq33-p3"
vdmmc="vcc-sdcvd-p3"

```

## 17.[gpio\_para]

配置项	配置项含义
compatible	该配置的名字
gpio_used	内核 GPIO 初始化使能功能，1：开启 0：禁用
gpio_num	GPIO 引脚数目
gpio_pin_1	GPIO 引脚配置
gpio_pin_2	GPIO 引脚配置
normal_led	正常状态灯使用的 GPIO
standby_led	休眠状态灯使用的 GPIO

配置举例：

```
compatible      = "allwinner,sunxi-init-gpio"
gpio_used       = 1
gpio_num        = 2
gpio_pin_1      = port:PL08<1><default><default><1>
gpio_pin_2      = port:power0<1><default><default><0>
normal_led      = "gpio_pin_1"
standby_led     = "gpio_pin_2"
```

## 18.USB 控制器标志

### 18.1.[usbc0]

配置项	配置项含义
usb0_used	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	USB 端口的使用情况。(xx=0/1/2) 0: device only 1: host only 2: OTG
usb_detect_type	USB 端口的检查方式。 0: 无检查方式 1: vbus/id 检查
usb_id_gpio	USB ID pin 脚配置
usb_det_vbus_gpio	USB DET_VBUS pin 脚配置
usb_drv_vbus_gpio	USB DRY_VBUS pin 脚配置
usb_host_init_state	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	usb 供电的 regulator GPIO
usb_wakeup_suspend	支持 usb 唤醒功能 0: 关闭 usb 唤醒功能 1: 当进入 normal standby 时候，支持 usb 唤醒（例如鼠标等外设）
USB device	
usb_luns	使用 mass storage 功能时的盘符数量
usb_serial_unique	usb device 的序列号是否唯一。1: 唯一，使用 chip id; 0: 相同: 由 usb_serial_number 指定
usb_serial_number	usb device 的序列号

配置举例：

```

usb0_used          = 1
usb_port_type      = 2
usb_detect_type    = 1
usb_id_gpio        = port:PH12<0><1><default><default>
usb_det_vbus_gpio  = "axp_ctrl"
usb_drv_vbus_gpio  = "axp_ctrl"
usb_host_init_state = 0
usb_regulator_io   = "nocare"
usb_regulator_vol   = 0
usb_wakeup_suspend = 0
;--- USB Device
usb_luns            = 3
usb_serial_unique   = 0
usb_serial_number   = "20080411"

```

### 18.2.[usbc1]

配置项	配置项含义
usb1_used	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 DRY_VBUS pin 脚配置。具体请参考 gpio 配置说明
usb_host_init_state	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 usb 供电的 regulator GPIO
usb_wakeup_suspend	支持 usb 唤醒功能 0：关闭 usb 唤醒功能        1：当进入 normal standby 时候，支持 usb 唤醒（例如鼠标等外设）

配置举例：

```
usb1_used          = 0
usb_drv_vbus_gpio  = port:PI1<0><1><default><default>
usb_host_init_state = 1
usb_regulator_io   = "nocare"
usb_wakeup_suspend = 0
```

### 18.3.[usbc2]

配置项	配置项含义
usb1_used	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 DRY_VBUS pin 脚配置。具体请参考 gpio 配置说明
usb_host_init_state	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 usb 供电的 regulator GPIO
usb_wakeup_suspend	支持 usb 唤醒功能 0：关闭 usb 唤醒功能        1：当进入 normal standby 时候，支持 usb 唤醒（例如鼠标等外设）

```
usbc2_used          = 0
usb_drv_vbus_gpio   =
usb_host_init_state = 1
usb_regulator_io    = "nocare"
usb_wakeup_suspend  = 0
```

## 19.重力感应器

### 19.1.[gsensor]

配置项	配置项含义
compatible	配置名字
gsensor_used	是否支持 gsensor
gsensor_twi_id	I2C 的 BUS 控制选择, 0: twi0;1:twi1;2:twi2
gsensor_twi_addr	芯片的 I2C 地址
gsensor_int1	中断 1 的 GPIO 配置
gsensor_int2	中断 2 的 GPIO 配置

配置举例:

```
compatible      = "allwinner,sun50i-gsensor-para"
gsensor_used    = 1
gsensor_twi_id  = 2
gsensor_twi_addr = 0x27
gsensor_int1    = port:power1<6><default><default><default>
gsensor_int2    =
```

### 19.2.[gsensor\_list]

配置项	配置项含义
compatible	配置名字
gsensor_list_used	是否支持 gsensor list
da380	是否支持 da380 模组

配置举例:

```
compatible      = "allwinner,sun50i-gsensor-list-para"
gsensor_list_used = 1
da380           = 1
```

## 20.WIFI

### 20.1.[wlan]

配置项	配置项含义
wlan_used	是否要使用 wifi
wlan_busnum	所使用的 SDIO 号，如使用的是 SDIO1，则此值为 1
clocks	低功耗时钟，此值固定为&clk_outa
wlan_power	wifi 模组使用哪一路 AXP 供电
wlan_io_regulator	wifi 模组 io 使用哪一路 AXP 供电
wlan_hostwake	wifi 唤醒主控脚
wlan_regon	Wifi 使能脚
wlan_clk_gpio	wifi 模块 32K 时钟输出硬件

配置举例：

```
[wlan]
wlan_used          = 1
wlan_busnum        = 3
wlan_usbnum        = 3
;clocks            = "&clk_outa"
wlan_power         = ""
wlan_io_regulator  = ""
wlan_regon         = port:PB07<1><default><default><0>
wlan_hostwake      = port:PI17<6><default><default><0>
wlan_clk_gpio      = port:PI12<4><default><default><0>
```



## 21. 蓝牙

### 21.1. [bt]

配置项	配置项含义
bt_used	蓝牙使用控制：1 使用，0 不用
clocks	低功耗时钟，此值固定为&clk_outa
bt_power	bt 模组使用哪一路 AXP 供电（通常情况下和 wifi 相同）
bt_io_regulator	bt 模组 io 使用哪一路 AXP 供电（通常情况下和 wifi 相同）
bt_rst_n	bt 使能脚

配置举例：

```

    bt_used          = 1
    ;clocks          = "&clk_outa"
    bt_power         = "vcc-wifi"
    bt_io_regulator  = ""
    bt_rst_n         = port:PG10<1><default><default><0>

```

### 21.2. [btlpm]

配置项	配置项含义
btlpm_used	蓝牙低功耗使用控制：1 使用，0 不用
uart_index	使用的串口序号，如使用 ttyS1，则此值为 1
bt_wake	主控唤醒 bt 引脚
bt_hostwake	bt 唤醒主控引脚

配置举例

```

    btlpm_used      = 1
    uart_index      = 1
    bt_wake         = port:PL06<1><default><default><1>
    bt_hostwake     = port:PL05<6><default><default><0>

```

## 22. Gps

### 22.1. [gps]

配置项	配置项含义
gps_used	是否要使用 GPS
gps_power	GPS 模块使用哪一路 AXP 供电
gps_nstandby	GPS 模块功能使能引脚，
gps_clk_gpio	GPS 模块 32k 时钟输出引脚

```
gps_used          = 1
;clocks           = "&clk_outb"
gps_power         = "vcc-gps"
gps_nstandby      = port:PI15<1><default><default><0>
gps_clk_gpio      = port:PI13<4><default><default><0>
```

## 23.gyrosopesensor

### 23.1.[gyrosopesensor]

配置项	配置项含义
compatible	配置名字
gyrosopesensor_used	是否支持 gyr
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:

```
compatible          = "allwinner,sun50i-gyr_sensors-para"
gyrosopesensor_used = 0
gy_twi_id           = 2
gy_twi_addr         = 0x6a
gy_int1             = port:PA10<6><1><default><default>
gy_int2             =gy_int2          =
```

## 24.光感(light sensor)

### 24.1.[lightsensor]

配置项	配置项含义
compatible	配置名字
ls_used	是否支持光传感器
ls_twi_id	I2C 的 BUS 控制选择，0: twi0;1:twi1;2:twi2
ls_twi_addr	芯片的 I2C 地址
ls_int	中断 1 的 GPIO 配置

配置举例：

```
compatible          = "allwinner,sun50i-lsensors-para"
lightsensor_used    = 0
ls_twi_id           = 2
ls_twi_addr         = 0x23
ls_int              = port:PA12<6><1><default><default>
```

## 25.罗盘 Compass

### 25.1.[compasssensor]

配置项	配置项含义
compatible	配置名称
compasssensor_used	是否支持罗盘
compass_twi_id	I2C 的 BUS 控制选择，0: twi0;1:twi1;2:twi2
compass_twi_addr	芯片的 I2C 地址
compass_int	中断 1 的 GPIO 配置

配置举例：

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

## 26. 数字音频总线（SPDIF）

### 26.1. [audiospdif]

配置项	配置项含义
audiospdif_used	是否开启 spdif codec，1：开启，0：不开启

配置举例：

```
audiospdif_used = 0
```

### 26.2. [spdif\_machine]

配置项	配置项含义
spdif_machine_used	是否开启 spdif machine，1：开启，0：不开启

配置举例：

```
spdif_machine_used = 0
```

注意：要生成并注册 spdif 声卡，就必须要把 audiospdif\_used 和 spdif\_machine\_used 都设置为 1。

### 26.3. [audiohdm]i

配置项	配置项含义
audiohdm]i_used	是否开启 hdmi codec，1：开启，0：不开启

配置举例：

```
audiohdm]i_used = 1
```

### 26.4. [hdmi\_machine]

配置项	配置项含义
hdmi_machine_used	是否开启 hdmi machine，1：开启，0：不开启

配置举例：

```
hdmi_machine_used = 1
```

注意：要生成并注册 HDMI 声卡，就必须要把 audiohdm]i\_used 和 hdmi\_machine\_used 都设置为 1。

### 26.5. [snddaudio0]

配置项	配置项含义
snddaudio0_used	是否使用该接口，默认配置为 0 1：使用 0：不使用

配置举例：

```
snddaudio0_used = 1
```

### 26.6. [daudio0]

配置项	配置项含义
daudio0_used	是否使用 daudio0 接口，默认要配置为 1 1: 使用 0: 不使用
pcm_lrck_period	每声道 bclk 个数/lrck 个数，设置如下： PCM mode: Number of BCLKs within(Left + Right)channel width 注意在 pcm 模式下，pcm_lrck_period 代表左和右声道相加，2 个声道的大小； I2S/Left-Justified/Right-Justified mode: Number of BCLKs within each individual channel width(Left or Right), pcm_lrck_period 代表左或者右声道，一个声道的大小； 在 i2s 模式下，一个 lrck 的宽度：2*32。假如 fs=48k，那么需要的 bclk 是 $3.072M = 2*32*48k$ ; bclk_div = $24.576M/3.072M=8$ ; 在 pcm 模式下，一个 lrck 的宽度就是 32。假如 fs=8k，那么需要的 bclk 是： $32*8k=256k$
pcm_lrckr_period	未使用
slot_width_select	数据 word 的宽度，对 i2s 模式，pcm 模式都有效。 16bits/20bits/24bits/32bits
pcm_lsb_first	数据 endian, 0: msb first; 1: lsb first
tx_data_mode	数据格式，0: 16bit linear PCM; 1: 8bit linear PCM; 2: 8bit u-law; 3: 8bit a-law
rx_data_mode	数据格式，0: 16bit linear PCM; 1: 8bit linear PCM; 2: 8bit u-law; 3: 8bit a-law
daudio_master	Master/slave 模式： 1:daudio0 slave; 4:daudio0 master
audio_format	1 SND_SOC_DAIFMT_I2S(standard i2s format). use 表示标准 i2s 格式； 2 SND_SOC_DAIFMT_RIGHT_J(right justified format). 表示右对齐格式； 3 SND_SOC_DAIFMT_LEFT_J(left justified format) 表示左对齐格式； 4 SND_SOC_DAIFMT_DSP_A 短帧模式 并设置 frame_width 为 0. 短帧； 5 SND_SOC_DAIFMT_DSP_B 长帧模式 并设置 frame_width 为 1.长帧；
signal_inversion	信号的翻转，比如标准的 I2S 模式，如果 lrck 翻转是模式，那么用示波器测量，左右声道是跟标准 i2s 模式相反的。如果 bclk 是翻转模式，那么用示波器测量，BCLK 信号是翻转的。 1 SND_SOC_DAIFMT_NB_NF(normal bit clock + frame) use 表示 bclk 采用正常模式，lrck 也正常模式 2 SND_SOC_DAIFMT_NB_IF(normal BCLK + inv FRM) 表示 bclk 采用正常模式，lrck 采用翻转模式 3 SND_SOC_DAIFMT_IB_NF(invert BCLK + nor FRM) use 表示 bclk 采用翻转模式，lrck 采用正常模式 4 SND_SOC_DAIFMT_IB_IF(invert BCLK + FRM) 表示 bclk 采用翻转模式，lrck 采用翻转模式
frametype	长帧或短帧 0: long frame = 2 clock width; 1: short frame

tdm_config	I2S 或 PCM 选择 0:pcm 1:i2s
------------	--------------------------

配置举例：

```
pcm_lrck_period = 0x40
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 = 0x04
signal_inversion = 0x04
frametype = 0x0
tdm_config = 0x00
mclk_div = 0x0
daudio0_used = 1
;daudio0_mclk = port:PB5<2><0><1><default>
daudio0_sync = port:PB7<2><0><1><default>
daudio0_bclk = port:PB6<2><0><1><default>
daudio0_dout = port:PB8<2><0><1><default>
daudio0_din = port:PB12<2><0><1><default>
```

注意：要生成并注册 Daudio0 声卡，就必须要把 snddaudio0\_used 和 daudio0\_used 都设置为 1。

## 26.7.[snddaudio1]

配置项	配置项含义
snddaudio1_used	是否使用该接口，默认配置为 0 1：使用 0：不使用

配置举例：

```
snddaudio1_used = 1
```

## 26.8.[daudio1]

配置项	配置项含义
daudio1_used	是否使用 daudio1 接口 1：使用 0：不使用
pcm_lrck_period	每声道 bclk 个数/lrck 个数，设置如下： PCM mode: Number of BCLKs within(Left + Right)channel width 注意在 pcm 模式下，pcm_lrck_period 代表左和右声道相加，2 个声道的大小； I2S/Left-Justified/Right-Justified mode: Number of BCLKs within each individual channel width(Left or Right), pcm_lrck_period 代表左或者右声道，一个声道的大小； 在 i2s 模式下，一个 lrck 的宽度：2*32。假如 fs=48k，那么需要的 bclk 是 3.072M = 2*32*48k; bclk_div = 24.576M/3.072M=8;



	在 pcm 模式下，一个 lrck 的宽度就是 32。假如 fs=8k，那么需要的 bclk 是：32*8k=256k
pcm_lrckr_period	未使用
slot_width_select	数据 word 的宽度，对 i2s 模式，pcm 模式都有效。 16bits/20bits/24bits/32bits
pcm_lsb_first	数据 endian，0: msb first; 1: lsb first
tx_data_mode	数据格式，0: 16bit linear PCM; 1: 8bit linear PCM; 2: 8bit u-law; 3: 8bit a-law
rx_data_mode	数据格式，0: 16bit linear PCM; 1: 8bit linear PCM; 2: 8bit u-law; 3: 8bit a-law
daudio_master	Master/slave 模式：1:daudio0 slave; 4:daudio0 master
audio_format	1 SND_SOC_DAIFMT_I2S(standard i2s format). use 表示标准 i2s 格式; 2 SND_SOC_DAIFMT_RIGHT_J(right justified format). 表示右对齐格式; 3 SND_SOC_DAIFMT_LEFT_J(left justified format) 表示左对齐格式; 4 SND_SOC_DAIFMT_DSP_A 短帧模式 并设置 frame_width 为 0. 短帧; 5 SND_SOC_DAIFMT_DSP_B 长帧模式 并设置 frame_width 为 1.长帧;
signal_inversion	信号的翻转，比如标准的 I2S 模式，如果 lrck 翻转是模式，那么用示波器测量，左右声道是跟标准 i2s 模式相反的。如果 bclk 是翻转模式，那么用示波器测量，BCLK 信号是翻转的。 1 SND_SOC_DAIFMT_NB_NF(normal bit clock + frame) use 表示 bclk 采用正常模式，lrck 也正常模式 2 SND_SOC_DAIFMT_NB_IF(normal BCLK + inv FRM) 表示 bclk 采用正常模式，lrck 采用翻转模式 3 SND_SOC_DAIFMT_IB_NF(invert BCLK + nor FRM) use 表示 bclk 采用翻转模式，lrck 采用正常模式 4 SND_SOC_DAIFMT_IB_IF(invert BCLK + FRM) 表示 bclk 采用翻转模式，lrck 采用翻转模式
frametype	长帧或短帧 0: long frame = 2 clock width; 1: short frame
tdm_config	I2S 或 PCM 选择 0:pcm 1:i2s

配置举例：

```

pcm_lrckr_period = 0x20
pcm_lrckr_period = 0x01
slot_width_select = 0x10
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

```

```
daudio1_used = 1
```

注意：要生成并注册 Daudio1 声卡，就必须要把 snddaudio1\_used 和 daudio1\_used 都设置为 1。

## 27.Codec 配置

### 27.1.[sndcodec]

配置项	配置项含义
sndcodec_used = 0x1	是否使用 V40 模拟音频输入输出 0x1：使用，0x0：不使用

配置举例：

```
sndcodec_used = 0x1
```

### 27.2.[codec]

配置项	配置项含义
codec_used	是否使用内置 codec。 1：使用 0：不使用
hp_dirused	使能耳机 0：不使用；1：使用
headphonevol	耳机音量 0x0--0x3f, 0：静音，(-62db)-0db, 1db/step
spkervol	喇叭音量 0x0--0x1f, -4.5db-6db, 1.5db/step
maingain	主 mic 增益 0x0--0x7, -4.5db-6db, 1.5db/step
adcagc_cfg	是否使用录音 agc, 0：不使用；1：使用
adcdrc_cfg	是否使用录音 drc, 0：不使用；1：使用
adchpf_cfg	是否使用录音 hpf, 0：不使用；1：使用
dacdrc_cfg	是否使用放音 drc, 0：不使用；1：使用
dachpf_cfg	是否使用放音 hpf, 0：不使用；1：使用
gpio-spk	喇叭 pa gpio 口配置

配置举例：

```

codec_used = 0x1
headphonevol =0x3b
spkervol =    0x1b
maingain =    0x4
hp_dirused = 0x1
adcagc_cfg =   0x0
adcdrc_cfg =   0x0
adchpf_cfg =   0x0
dacdrc_cfg =   0x0
dachpf_cfg =   0x0
gpio-spk = port:PB05<1><1><default><default>

```

## 28.PMU 电源

### 28.1.[pmu0]

配置项	相关说明
compatible	AXP 名字
used	是否使用 AXPxx: 0:不使用,1:使用
pmu_id	Pmu 的 id 号
reg	Twi id 号
pmu_vbusen_func	Vubs 引脚 0: 输出 1: 输入
pmu_reset	长按 16s, 0: 不操作 1: 重启
pmu_irq_wakeup	是否允许中断唤醒
pmu_hot_shutdown	是否允许 pmu 高温关机
pmu_inshort	启动是否检测电池电量

配置举例:

```

compatible      = "axp221s"
used            = 1
pmu_id          = 2
reg             = 0x34
pmu_vbusen_func = 0
pmu_reset       = 0
pmu_irq_wakeup  = 1
pmu_hot_shutdown = 1
pmu_inshort     = 0

```

### 28.2.[charger0]

配置举例:

配置项	相关说明
compatible	设备名
used	是否使用 AXPxx: 0:不使用,1:使用
pmu_bat_unused	是否使用电池, 1:不使用,0:使用
pmu_chg_ic_temp	
pmu_battery_rdc	电池通路内阻, 单位 $m\Omega$
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_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_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_ac_vol	usb-ac 限制电压

pmu_ac_cur	usb-ac 限制电流
pmu_usbpc_vol	usb-pc 限制电压
pmu_usbpc_cur	usb-pc 限制电流
pmu_battery_warning_level1	低电量警告 level1
pmu_battery_warning_level2	低电量警告 level2
pmu_chgled_func	CHGKED 引脚控制。0: PMU 1: 充电器
pmu_chgled_type	CHGLED 类型。0: Type A 1: Type B
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_bat_temp_enable	电池温度检测使能
pmu_bat_charge_ltf	电池充电低温门限电压
pmu_bat_charge_htf	电池充电高温门限电压
pmu_bat_shutdown_ltf	关机电池低温门限电压
pmu_bat_shutdown_htf	关机电池高温门限电压

pmu_bat_temp_para1	电池温度-25 度对应的电压
pmu_bat_temp_para2	电池温度-15 度对应的电压
pmu_bat_temp_para3	电池温度-10 度对应的电压
pmu_bat_temp_para4	电池温度-5 度对应的电压
pmu_bat_temp_para5	电池温度 0 度对应的电压
pmu_bat_temp_para6	电池温度 5 度对应的电压
pmu_bat_temp_para7	电池温度 10 度对应的电压
pmu_bat_temp_para8	电池温度 20 度对应的电压
pmu_bat_temp_para9	电池温度 30 度对应的电压
pmu_bat_temp_para10	电池温度 40 度对应的电压
pmu_bat_temp_para11	电池温度 45 度对应的电压
pmu_bat_temp_para12	电池温度 50 度对应的电压
pmu_bat_temp_para13	电池温度 55 度对应的电压
pmu_bat_temp_para14	电池温度 60 度对应的电压
pmu_bat_temp_para15	电池温度 70 度对应的电压
pmu_bat_temp_para16	电池温度 80 度对应的电压
pmu_powkey_off_time	Power key 关机或者重启响应时间
pmu_powkey_off_func	Power key 功能选择, 0: 关机; 1 重启
pmu_powkey_off_en	Power key 使能
pmu_powkey_long_time	Power key 长按响应时间
pmu_powkey_on_time	Power key 开机响应时间
power_start	当充电状态下的关机动作。1: 重启 2: 关机

compatible = "axp221s-charger"

pmu\_bat\_unused = 1

pmu\_chg\_ic\_temp = 0

pmu\_battery\_rdc = 100

pmu\_battery\_cap = 0

pmu\_runtime\_chgcur = 450

pmu\_suspend\_chgcur = 1500

pmu\_shutdown\_chgcur = 1500

pmu\_init\_chgvol = 4200

pmu\_ac\_vol = 4000

pmu\_ac\_cur = 0

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

power\_start = 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	= 0
pmu_bat_para8	= 0
pmu_bat_para9	= 5
pmu_bat_para10	= 8
pmu_bat_para11	= 9
pmu_bat_para12	= 10
pmu_bat_para13	= 13
pmu_bat_para14	= 16
pmu_bat_para15	= 20
pmu_bat_para16	= 33
pmu_bat_para17	= 41
pmu_bat_para18	= 46
pmu_bat_para19	= 50
pmu_bat_para20	= 53
pmu_bat_para21	= 57
pmu_bat_para22	= 61
pmu_bat_para23	= 67
pmu_bat_para24	= 73
pmu_bat_para25	= 78
pmu_bat_para26	= 84
pmu_bat_para27	= 88
pmu_bat_para28	= 92
pmu_bat_para29	= 93
pmu_bat_para30	= 94
pmu_bat_para31	= 95
pmu_bat_para32	= 100

pmu_bat_temp_enable	= 0
pmu_bat_charge_ltf	= 2261
pmu_bat_charge_hrf	= 388
pmu_bat_shutdown_ltf	= 3200
pmu_bat_shutdown_hrf	= 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

```

### 28.3.[powerkey0]

配置项	相关说明
compatible	设备名
pmu_powkey_off_timet	系统起来后，长按关机时间
pmu_powkey_off_func	系统起来后，长按功能 0:shutdown,1:restart
pmu_powkey_off_en	系统起来后，是否使用长按功能
pmu_powkey_long_time	短按响应时间
pmu_powkey_on_time	关机后，长按开机时间

```

compatible      = "axp221s-powerkey"
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

```

### 28.4.[regulator0]

配置项	相关说明
compatible	设备名
regulator_count	regulator 数量
regulator1	regulator1 对应的别名，请勿修改
regulator2	regulator2 对应的别名，请勿修改
regulator3	regulator3 对应的别名，请勿修改
regulator4	regulator4 对应的别名，请勿修改
regulator5	regulator5 对应的别名，请勿修改
regulator6	regulator6 对应的别名，请勿修改
regulator7	regulator7 对应的别名，请勿修改
regulator8	regulator8 对应的别名，请勿修改
regulator9	regulator9 对应的别名，请勿修改
regulator10	regulator10 对应的别名，请勿修改
regulator11	regulator11 对应的别名，请勿修改
regulator12	regulator12 对应的别名，请勿修改
regulator13	regulator13 对应的别名，请勿修改
regulator14	regulator14 对应的别名，请勿修改
regulator15	regulator15 对应的别名，请勿修改

```
compatible      = "axp221s-regulator"
```

```
regulator_count = 20
```

```

regulator1      = "axp221s_dcdc1 none vcc-hdmi vcc-io vcc-dsi vcc-usb vdd-efuse vcc-hp vcc-audio
vcc-emmc vcc-card vcc-pa vcc-pd vcc-sdcv vcc-sdcvq33 vcc-sdcvd vcc-ctp vcc-sdcv-p3 vcc-sdcvq33-p3
vcc-sdcvd-p3"

```



regulator2 = "axp221s\_dcdc2 none vdd-cpua"  
regulator3 = "axp221s\_dcdc3 none vdd-sys"  
regulator4 = "axp221s\_dcdc4 none"  
regulator5 = "axp221s\_dcdc5 none vcc-dram"  
regulator6 = "axp221s\_rtc none vcc-rtc"  
regulator7 = "axp221s\_aldo1 none vcc-25 vcc-tvd"  
regulator8 = "axp221s\_aldo2 none vcc-pc vcc-emmcvq18"  
regulator9 = "axp221s\_aldo3 none avcc vcc-pll"  
regulator10 = "axp221s\_dldo1 none vcc-io-wifi vcc-pg"  
regulator11 = "axp221s\_dldo2 none vcc-wifi"  
regulator12 = "axp221s\_dldo3 none vcc-gps"  
regulator13 = "axp221s\_dldo4 none csi-avdd"  
regulator14 = "axp221s\_eldo1 none isp-dvdd12"  
regulator15 = "axp221s\_eldo2 none csi-dvdd"

## 28.5.[axp\_gpio0]

配置项	相关说明
compatible	设备名

compatible = "axp221s-gpio"

## 29.Vf 表设置

注意：vf 表（电压频率对应表）关乎系统稳定性，请勿私自修改！

### 29.1.[dvfs\_table]

配置项	配置项含义
extremity_freq	极限频率
max_freq	最大频率
min_freq	最小频率
lv_count	vf 表的级数
lvn_freq	对应的最大频率(n 表示级数)
lvn_volt	第 n 级的电压

配置举例：

```
max_freq = 1200000000
```

```
min_freq = 240000000
```

```
lv_count = 8
```

```
lv1_freq = 1200000000
```

```
lv1_volt = 1300
```

```
lv2_freq = 1104000000
```

```
lv2_volt = 1240
```

```
lv3_freq = 1008000000
```

```
lv3_volt = 1160
```

```
lv4_freq = 912000000
```

```
lv4_volt = 1100
```

```
lv5_freq = 720000000
```

```
lv5_volt = 1000
```

```
lv6_freq = 0
```

```
lv6_volt = 1000
```

```
lv7_freq = 0
```

```
lv7_volt = 1000
```

```
lv8_freq = 0
```

```
lv8_volt = 1000
```

## 30.Virtual device

### 30.1.[Vdevice]

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_used      = 1
Vdevice_0         = port:PB00<4><1><2><default>
Vdevice_1         = port:PB01<4><1><2><default>
```

## 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.