

Tina

System Configuration 说明书 v1.2

文档履历

版本号	日期	制/修订人	制/修订记录
V1.0	2016/8/18		初始版本
V1.1	2017/10/24		
V1.2	2018/1/23		增加一些配置项，简化配置说明(<X>=1,2,3....)



目 录

1. 说明.....	6
1.1. 文档说明.....	6
1.2. 配置文件路径.....	6
2. 系统.....	7
2.1. [product].....	7
2.2. [platform].....	7
2.3. [target].....	7
2.4. [power_sply].....	8
2.5. [card_boot].....	9
2.6. [card0_boot_para].....	9
2.7. [card2_boot_para].....	10
2.8. [twi_para].....	11
2.9. [uart_para].....	11
2.10. [jtag_para].....	11
2.11. [clock].....	12
2.12. [pm_para].....	12
3. DRAM.....	13
3.1. [dram_para].....	13
4. Ethernet MAC Controller.....	14
4.1. [gmac_para].....	14
5. I2C 总线.....	15
5.1. [twi<X>].....	15
6. 串口(UART).....	16
6.1. [uart<X>].....	16
7. SPI 总线.....	17
7.1. [spi<X>].....	17
7.2. [spiX/spi_boardX].....	17
8. gpadc.....	18
8.1. [gpadc].....	18
9. 触摸屏配置.....	19
9.1. [rtp_para].....	19
9.2. [ctp].....	20
9.3. [acc_gpio].....	20
9.4. [ctp_list].....	21
10. 触摸按键.....	22
10.1. [tkey_para].....	22
11. 马达.....	23
11.1. [motor_para].....	23
12. 闪存.....	24
12.1. [nand<X>_para].....	24
13. 显示.....	26
13.1. [boot_disp].....	26
13.2. [disp].....	26
13.3. [edp<X>].....	27
13.4. [lcd<X>_suspend].....	28
13.5. [car_reverse].....	28
13.6. [lcd<X>].....	30
14. PWM.....	32
14.1. [pwm<X>].....	32
14.2. [pwm<X>_suspend].....	32
14.3. [spwm<X>].....	32
14.4. [spwm<X>_suspend].....	32
15. HDMI.....	33

15.1. [hdmi].....	33
16. tvd 摄像头.....	34
16.1. [tvd].....	34
17. vind 摄像头.....	35
17.1. [vind<X>].....	35
17.2. [vind<X>/csi<X>].....	35
17.3. [vind<X>/csi_cci<X>].....	35
17.4. [vind<X>/flash<X>].....	36
17.5. [vind<X>/actuator<X>].....	36
17.6. [vind<X>/sensor<X>].....	37
17.7. [vind<X>/vinc<X>].....	37
18. 摄像头(CSI).....	39
18.1. [csi<X>].....	39
18.2. [csi<X>/csi0_dev0].....	40
19. tvout/tvin.....	42
19.1. [tvout_para].....	42
19.2. [tvin_para].....	42
19.3. [di].....	42
20. SD / MMC.....	43
20.1. [sdc<X>].....	43
20.2. [smc].....	44
21. [gpio_para].....	45
22. USB 控制标志.....	46
22.1. [usbc<X>].....	46
23. [serial_feature].....	47
24. 重力感应(G Sensor).....	48
24.1. [gsensor_para].....	48
24.2. [gsensor_list].....	48
25. WiFi.....	49
25.1. [wlan].....	49
26. 蓝牙(blutetooth).....	50
26.1. [bt].....	50
26.2. [btlpm].....	50
27. 光感(light sensor).....	51
27.1. [ls_para].....	51
28. 陀螺仪传感器(gyroscope sensor).....	52
28.1. [gy_para].....	52
29. 罗盘 Compass.....	53
29.1. [compass_para].....	53
30. 数字音频总线 (SPDIF).....	54
30.1. [spdif] & [sndspdif].....	54
30.2. [audiohdmi] & [sndhdmi].....	54
30.3. [snddaudio<X>].....	54
30.4. [daudio<X>].....	55
31. 内置音频 codec.....	56
31.1. [sndcodec].....	56
31.2. [i2s].....	56
31.3. [codec].....	56
32. [s_cir0].....	57
33. PMU 电源.....	58
33.1. [pmu<X>].....	58
33.2. [charger<X>].....	58
33.3. [powerkey<X>].....	62
33.4. [regulator<X>].....	62
33.5. [axp_gpio<X>].....	63

34. [psensor_table].....	64
35. DVFS.....	65
35.1. [dvfs_table]&&[dvfs_table_[X]].....	65
36. s_uart<X>.....	66
37. s_twi<X>.....	67
38. s_jtag<X>.....	68
39. Virtual device.....	69
39.1. [Vdevice].....	69
40. Declaration.....	70



1. 说明

1.1. 文档说明

1. 蓝色字体为芯片引脚配置,黑色字体为芯片内部控制配置项;
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. 文中的<X>=0,1,2,3,4,5.....,如 twi0,twi1.....; uart0,uart1....
4. 部分模块的配置项目可能是多余的,同时配置举例仅供参考,不一定为真实可用的,实际使用时需向技术支持人员询问

1.2. 配置文件路径

系统配置文件路径:

tina/target/allwinner/<方案>/configs/sys_config.fex



2. 系统

2.1. [product]

配置项	配置项含义
version	配置的版本号
machine	方案名字
示例	
<pre>[product] version = "100" machine = "m2ultra"</pre>	

2.2. [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	PhoenixUSBPro 量产完成后: 1-不做任何动作, 2-重启, 3-关机, 4-量产, 5-正常启动, 6-量产结束进入关机关机充电
debug_mode	Uboot 阶段打印等级: 0-不打印, 1-打印
示例	
<pre>[platform] eraseflag = 0</pre>	

2.3. [target]

配置项	配置项含义
boot_clock	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	支持 DragonSN_V2.0 烧录 sn 号
示例	
<pre>[target] boot_clock = 1008 storage_type = -1 burn_key = 1</pre>	

2.4. [power_sply]

配置项	配置项含义
dcdc<X>_vol	dcdc<X>模块输出电压;
aldo<X>_vol	aldo<X> 模块输出电压;
dc1sw_vol	dc1sw 模块输出电压
dc5ldo_vol	dc5ldo 模块输出电压
dldo<X>_vol	dldo<X> 模块输出电压;
gpio<X>_vol	gpio<X> 的输出电压;

示例:

```
[power_sply]
dcdc1_vol      = 1003300
dcdc2_vol      = 1001160
dcdc3_vol      = 1001100
dcdc4_vol      = 1100
aldo1_vol      = 2800
aldo2_vol      = 1001500
aldo3_vol      = 1003000
dc1sw_vol      = 3000
dc5ldo_vol     = 1100
dldo1_vol      = 3300
dldo2_vol      = 3300
dldo3_vol      = 3300
dldo4_vol      = 2500
eldo1_vol      = 2800
eldo2_vol      = 1500
eldo3_vol      = 1200
gpio0_vol      = 3300
gpio1_vol      = 1800
```

补充说明:

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
 这里的电压值单位为 mV

2.5. [card_boot]

配置项	配置项含义
logical_start	启动卡逻辑起始扇区
sprite_gpio0	卡量产 gpio led 灯配置
next_work	卡量产完成后：1-不做任何动作，2-重启，3-关机，4-量产，5-正常启动
示例:	
<pre>[card_boot] logical_start = 40960 sprite_gpio0 = port:PH21<1><default><default><default></pre>	

2.6. [card0_boot_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	速度模式 0 为低速，1 为高速
card_line	代表 4 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
示例:	
<pre>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></pre>	

2.7. [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	1,4,8 线卡可以选择
sdc_cmd	sdc 命令信号的 GPIO 配置
sdc_clk	sdc 卡时钟信号的 GPIO 配置
sdc_d<X>	sdc 卡数据<X>线信号的 GPIO 配置
sdc_emmc_rst	sdc 卡 rst 引脚
sdc_ds	
sdc_ex_dly_used	
sdc_io_1v8	

示例:

```
[card2_boot_para]
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
;sd_io_1v8     =
```

2.8. [twi_para]

配置项	配置项含义
twi_port	Boot 的 twi 控制器编号
twi_scl	Boot 的 twi 的时钟的 GPIO 配置
twi_sda	Boot 的 twi 的数据的 GPIO 配置
示例:	
[twi_para]	
twi_port	= 0
twi_scl	= port:PB0<2><default><default><default>
twi_sda	= port:PB1<2><default><default><default>

2.9. [uart_para]

配置项	配置项含义
uart_debug_port	Boot 串口控制器编号
uart_debug_tx	Boot 串口发送的 GPIO 配置
uart_debug_rx	Boot 串口接收的 GPIO 配置
示例:	
[uart_para]	
uart_debug_port	= 0
uart_debug_tx	= port:PF02<3><1><default><default>
uart_debug_rx	= port:PF04<3><1><default><default>

2.10. [jtag_para]

配置项	配置项含义
jtag_enable	JTAG 使能
jtag_ms	测试模式选择输入(TMS) 的 GPIO 配置
jtag_ck	测试时钟输入(TMS) 的 GPIO 配置
jtag_do	测试数据输出(TDO) 的 GPIO 配置
jtag_di	测试数据输入 (TDI) 的 GPIO 配置
示例:	
[jtag_para]	
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>

2.11. [clock]

配置项	配置项含义
pll4	pll4 时钟频率(MHz)
pll8	pll8 时钟频率(MHz)
pll9	pll9 时钟频率(MHz)
pll12	pll12 时钟频率(MHz)
示例:	
[clock]	
pll4	= 297
pll8	= 297
pll9	= 384
pll12	= 297

2.12. [pm_para]

配置项	配置项含义
standby_mode	if 1 == standby_mode, then support super standby; else, support normal standby.
示例:	
[pm_para]	
standby_mode	= 1



3. DRAM

3.1. [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_mr<X>	DRAM 控制器内部参数，由原厂来进行调节，请勿修改

示例:

```
[dram_para]
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        = 0
dram_tpr10       = 0x0008
dram_tpr11       = 0x44450000
dram_tpr12       = 0x9777
dram_tpr13       = 0x4090950
```

4. Ethernet MAC Controller

4.1. [gmac_para]

配置项	配置项含义
gmac_used	是否使用 Ethernet
gmac_txd<X>	发送数据 GPIO 配置
gmac_txclk	发送时钟信号
gmac_txen	发送使能信号
gmac_gtxclk	gtx 时钟信号
gmac_rxd<X>	接收数据 GPIO 配置
gmac_rxdv	接收有效指示
gmac_rxclk	接收时钟信号
gmac_txerr	接收出错指示
gmac_col	冲突检测
gmac_crs	crs GPIO 配置
gmac_clkin	clkin GPIO 配置
gmac_mdc	配置接口时钟
gmac_mdio	配置接口 I/O
示例:	
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

5.1. [twi<X>]

配置项	配置项含义
twiX_used	TWI 使用控制：1 使用，0 不用
twiX_scl	TWI SCK 的 GPIO 配置
twiX_sda	TWI SDA 的 GPIO 配置
示例:	
[twi0]	
twiX_used	= 1
twiX_scl	= port:PB00<2><default><default><default>
twiX_sda	= port:PB01<2><default><default><default>



6. 串口(UART)

6.1. [uart<X>]

配置项	配置项含义
uart_used	UART 使用控制: 1 使用, 0 不用
uart_port	UART 端口号
uart_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;
uartX_tx	UART TX 的 GPIO 配置
uartX_rx	UART RX 的 GPIO 配置
uartX_rts	UART RTS 的 GPIO 配置
uartX_cts	UART CTS 的 GPIO 配置
uartX_dtr	UART DTR 的 GPIO 配置
uartX_dsr	UART DSR 的 GPIO 配置
uartX_dcd	UART DCD 的 GPIO 配置
uartX_ring	UART RING 的 GPIO 配置
示例:	
[uart1]	
uart1_used	= 0
uart1_port	= 1
uart1_type	= 8
uart1_tx	= port:PA10<4><1><default><default>
uart1_rx	= port:PA11<4><1><default><default>
uart1_rts	= port:PA12<4><1><default><default>
uart1_cts	= port:PA13<4><1><default><default>
uart1_dtr	= port:PA14<4><1><default><default>
uart1_dsr	= port:PA15<4><1><default><default>
uart1_dcd	= port:PA16<4><1><default><default>
uart1_ring	= port:PA17<4><1><default><default>

7. SPI 总线

7.1. [spi<X>]

配置项	配置项含义
spiX_used	SPI 使用控制：1 使用，0 不用
spiX_cs_number	spiX 片选个数，最多 2 个
spiX_cs_bitmap	由于 SPI 控制器支持多个 CS，这一个参数表示 CS 的掩码；
spiX_cs0	SPI CS0 的 GPIO 配置
spiX_cs1	SPI CS1 的 GPIO 配置
spiX_sclk	SPI CLK 的 GPIO 配置
spiX_mos	SPI MOSI 的 GPIO 配置
spiX_miso	SPI MISO 的 GPIO 配置
示例:	
<pre>[spi0] 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></pre>	

7.2. [spiX/spi_boardX]

配置项	配置项含义
compatible	设备名称
spi-max-frequency	工作最大频率
reg	片选
spi-cpha	时钟相位
spi-cpol	时钟极性
spi-cs-high	默认 0，为 1 表示 flash 的片选为 high active
示例:	
<pre>[spi0/spi_board0] compatible = "m25p80" spi-max-frequency = 1000000 reg = 0 ;spi-cpha ;spi-cpol ;spi-cs-high</pre>	

8. gpadc

8.1. [gpadc]

配置项	配置项含义
gpadc_used	
channel_num	
channel_select	
channel_data_select	
channel_compare_select	
channel_cld_select	
channel0_compare_lowdata	
channel0_compare_higdata	
key_cnt	
key<X>_vol	
示例:	
<pre>[gpadc] gpadc_used = 1 channel_num = 1 channel_select = 0x01 channel_data_select = 0 channel_compare_select = 0x01 channel_cld_select = 0x01 channel_chd_select = 0 channel0_compare_lowdata = 1700000 channel0_compare_higdata = 1200000 key_cnt = 5 key0_vol = 115 key0_val = 115 key1_vol = 240 key1_val = 114 key2_vol = 360 key2_val = 139 key3_vol = 480 key3_val = 28 key4_vol = 600 key4_val = 102</pre>	

9. 触摸屏配置

9.1. [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
示例:	
<pre>[rtp_para] 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</pre>	

9.2. [ctp]

配置项	配置项含义
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

示例:

```
[ctp]
ctp_used           = 1
ctp_twi_id         = 1
ctp_twi_addr       = 0x5d
ctp_screen_max_x   = 1280
ctp_screen_max_y   = 800
ctp_revert_x_flag  = 1
ctp_revert_y_flag  = 1
ctp_exchange_x_y_flag = 1
ctp_int_port       = port:PI10<6><default><default><default>
ctp_wakeup         = port:PH10<1><default><default><1>
ctp_power_ldo      = "vcc-ctp"
ctp_power_ldo_vol  = 3300
ctp_power_io       =
```

9.3. [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

示例:

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

9.4. [ctp_list]

配置项	配置项含义
ctp_det_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
gslX680	是否支持 gslX680 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 d5306_ts 模组
ctp_det_used	支持触摸屏 list
tu_ts	
gt818ts	
icn83xx_ts	
示例:	
<pre>[ctp_list] compatible = "allwinner,sun50i-ctp-list" ctp_det_used = 1 ft5x_ts = 1 gt82x = 1 gslX680 = 0 gslX680new = 1 gt9xx_ts = 1 gt9xxf_ts = 0 tu_ts = 0 gt818_ts = 0 zet622x = 0 aw5306_ts = 0 icn83xx_ts = 0</pre>	

10. 触摸按键

10.1. [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
示例:	
tkey_used	= 0
tkey_twi_id	=
tkey_twi_addr	=
tkey_int	=



11. 马达

11.1. [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
示例:	
[motor_para]	
motor_used	= 0
motor_shake	= port:power3<1><<default><default><1>
注意事项:	
motor_shake = port:power3<1><<default><default><1>	
默认 io 口的输出应该为 1，这样就不会初始化之后就开始震动了。	
假设 motor_shake = 0，说明没有指定 gpio 引脚，那么就会设置 axp 的引脚为马达供电，优先考虑 gpio 配置。	



12. 闪存

12.1. [nand<X>_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
nand0_we	nand0 写时钟信号的 GPIO 配置
nand0_ale	nand0 地址使能信号的 GPIO 配置
nand0_cle	nand0 命令使能信号的 GPIO 配置
nand0_ce1	nand0 片选 1 信号的 GPIO 配置
nand0_ce0	nand0 片选 0 信号的 GPIO 配置
nand0_nre	nand0 读时钟信号的 GPIO 配置
nand0_rb0	nand0 Read/Busy 1 信号的 GPIO 配置
nand0_rb1	nand0 Read/Busy 0 信号的 GPIO 配置
nand0_d[X]	nand0 数据总线信号的 GPIO 配置,[X]=0,1,2...
nand0_nwp	
nand0_ce[X]	nand0 片选[X]信号的 GPIO 配置,[X]=0,1,2...
nand0_ndqs	
nand0_regulator1	
nand0_regulator2	
nand0_cache_level	
nand0_flush_cache_num	
nand0_capacity_level	
nand0_id_number_ctl	
nand0_print_level	
nand0_p0	
nand0_p1	
nand0_p2	
nand0_p3	

示例:

```
[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
```



13.显示

13.1. [boot_disp]

配置项	配置项含义
output_disp	支持显示用户自定义 bootlogo
output_type = 1	1:LCD 2:TV 3:HDMI 4:VGA
output_mode = 10	(used for tv/hdmi output, 0:480i,1:576i,2:480p,3:576p,4:720p50,5:720p60,6:1080i50,7:1080i60,8:1080p24,9:1080p5,10:1080p60,11:pal 14:ntsc)
示例:	
<pre>[boot_disp] output_disp = 0 output_type = 1 output_mode = 10</pre>	

13.2. [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>
screen<X>_output_type	屏 0 输出类型(0:none; 1:lcd; 2:tv; 3:hdmi; 4:vga)
screen<X>_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)
screen<X>_output_format	0:RGB 1:yuv444 2:yuv422 3:yuv420
screen<X>_output_bits	0:8bit 1:10bit 2:12bit 2:16bit
screen<X>_output_eof	0:reserve 4:SDR 16:HDR10 18:HLG
screen[X]_output_cs	0:undefined 257:BT709 260:BT601 263:BT2020
fb<X>_format	fb<X>的格式(0:ARGB 1:ABGR 2:RGBA 3:BGRA)
fb<X>_width	fb<X>的宽度,为 0 时将按照输出设备的分辨率
fb<X>_height	fb<X>的高度,为 0 时将按照输出设备的分辨率
lcd<X>_backlight	lcd<X>的背光初始值, 0~255
lcd<X>_bright	lcd<X>的亮度值, 0~100
lcd<X>_contrast	lcd<X>的对比度, 0~100
lcd<X>_saturation	lcd<X>的饱和度, 0~100
lcd<X>_hue	lcd<X>的色度, 0~100
示例:	
<pre>[disp] disp_init_enable = 1 disp_mode = 0 screen0_output_type = 1 screen0_output_mode = 5</pre>	

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
lcd0_backlight	= 50
lcd1_backlight	= 50
lcd0_bright	= 50
lcd0_contrast	= 50
lcd0_saturation	= 57
lcd0_hue	= 50
lcd1_bright	= 50
lcd1_contrast	= 50
lcd1_saturation	= 57
lcd1_hue	= 50



13.3. [edp<X>]

配置项	配置项含义
-----	-------

全志科技版权所有，侵权必究

used	
edp_io_power	
edp_x	
edp_y	
edp_hbp	
edp_ht	
edp_hspw	
edp_vbp	
edp_vt	
edp_vspw	
edp_rate	
edp_lane	
edp_fps	
edp_colordepth	

示例:

```
[[edp0]
used=1
edp_io_power = "vcc-edp"
edp_x=2048
edp_y=1536
edp_hbp=10
edp_ht=2208
edp_hspw=5
edp_vbp=10
edp_vt=1570
edp_vspw=1
edp_rate=0
edp_lane=4
edp_fps=60
edp_colordepth=0
```

13.4. [lcd<X>_suspend]

配置项	配置项含义
lcdd<X>	lcd 数据<X>线信号休眠状态下的 GPIO 配置
示例:	
[lcd0_suspend]	
;lcdd0	= port:PD00<7><0><default><default>
;lcdd1	= port:PD01<7><0><default><default>
;lcdd2	= port:PD02<7><0><default><default>
;lcdd3	= port:PD03<7><0><default><default>
;lcdd4	= port:PD04<7><0><default><default>
;lcdd5	= port:PD05<7><0><default><default>
;lcdd6	= port:PD06<7><0><default><default>
;lcdd7	= port:PD07<7><0><default><default>
;lcdd8	= port:PD08<7><0><default><default>
;lcdd9	= port:PD09<7><0><default><default>

13.5. [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
tvd_id	倒车模块使用的 tvd 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

示例:

```
[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>
```



13.6. [lcd<X>]

配置项	配置项含义
lcd_used	是否使用 lcd0
lcd_driver_name	定义驱动名称
lcd_bl_0_percent	
lcd_bl_40_percent	
lcd_bl_100_percent	
cd_backlight	LCD 背光值
lcd_if	lcd 接口(0:hv(sync+de); 1:8080; 2:tft; 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_dsi_if	
lcd_dsi_lane	
lcd_dsi_format	
lcd_dsi_te	
lcd_dsi_eotp	
lcd_lvds_if	lcd lvds 接口, 0:single link; 1:dual link
lcd_lvds_colordepth	cd lvds 颜色深度 0:8bit; 1:6bit
lcd_lvds_mode	lcd 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_io_phase	
lcd_hv_clk_phase	lcd hv 时钟相位 0:0 degree; 1:90 degree; 2: 180 degree; 3: 270 degree
lcd_hv_sync_polarity	lcd io 属性, 0:not invert; 1:invert
lcd_gamma_en	lcdgamma 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
deu_mode	deu 模式 0:small lcd screen; 1:large lcd screen(larger than 10inch)
lcdgamma4ieq	使能背光参数, lcd gamma vale*10;decrease it while lcd is not bright enough; increase while lcd is too bright
lcd_dsi_port_num	
lcd_tcon_mode	
lcd_slave_stop_pos	
lcd_sync_pixel_num	

lcd_sync_line_num	
smart_color	丽色系统, 90:normal lcd screen 65:retina lcd screen(9.7inch)
lcd_bl_en	背光使能的 GPIO 配置
lcd_power	lcd 电源
lcd_gpio_<X>	lcd 数据<X>线信号的 GPIO 配置

```

示例:
[lcd0]
lcd_used                = 1

lcd_driver_name         = "S070WV20_MIPI_RGB"
lcd_backlight          = 50
lcd_if                  = 4
lcd_x                   = 800
lcd_y                   = 480
lcd_width               = 86
lcd_height              = 154
lcd_dclk_freq           = 20
lcd_pwm_used            = 1
lcd_pwm_ch              = 0
lcd_pwm_freq            = 50000
lcd_pwm_pol             = 1
lcd_pwm_max_limit      = 255
lcd_hbp                 = 88
lcd_ht                  = 928
lcd_hspw                = 48
lcd_vbp                 = 32
lcd_vt                  = 525
lcd_vspw                = 3

lcd_frm                 = 0
lcd_cmap_en             = 0
lcd_dsi_if              = 0
lcd_dsi_lane            = 4
lcd_dsi_format          = 0
lcd_dsi_te              = 0

deu_mode                = 0
lcdgamma4iep            = 22
smart_color             = 90

lcd_bl_en               = port:PH16<1><0><2><1>
lcd_power               = "vcc-3v"
;lcd_power              = "vcc-mipi"
lcd_gpio_0              = port:PH17<1><0><2><1>
lcd_gpio_1              = port:PH18<1><0><2><1>

```

14.PWM

14.1. [pwm<X>]

配置项	配置项含义
pwm_used	是否使用 PWM0
pwm_positive	PWM 输出 GPIO 配置
示例:	
[pwm0]	
pwm_used	= 1
pwm_positive	= port:PB2<3><0><default><default>

14.2. [pwm<X>_suspend]

配置项	配置项含义
pwm<X>_suspend	
示例:	
pwm_positive	= port:PB2<3><0><default><default>

14.3. [spwm<X>]

配置项	配置项含义
s_pwm<X>_used	是否使用 s_pwm<X>
pwm_positive	PWM 输出 GPIO 配置
示例:	
pwm_positive	= port:PL16<2><0><default><default>

14.4. [spwm<X>_suspend]

配置项	配置项含义
spwm0_suspend	
示例:	
pwm_positive	= port:PL16<7><0><default><default>

15.HDMI

15.1. [hdmi]

配置项	配置项含义
hdmi_used	是否使用 hdmi。1:使用;0:不使用
hdmi_hdcv_enable	是否使能 hdcp
hdmi_cts_compatibility	cts 兼容性使能设置
hdmi_power	内核阶段 hdmi 电源配置
示例:	
[hdmi]	
hdmi_used	= 1
hdmi_hdcv_enable	= 0
hdmi_cts_compatibility	= 0



16.tvd 摄像头

16.1. [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
示例:	
[tvd0]	
tvd_used	= 1
tvd_if	= 0
fliter_used	= 1
cagc_enable	= 1
agc_auto_enable	= 1



17.vind 摄像头

17.1. [vind<X>]

配置项	配置项含义
vind0_used	Vin 框架使能配置
示例:	
[tvd0]	
tvd_used	= 1
tvd_if	= 0
fliter_used	= 1
cagc_enable	= 1
agc_auto_enable	= 1

17.2. [vind<X>/csi<X>]

配置项	配置项含义
csi0_used	vin 框架对应的 csi 使能配置
csi0_pck	csi pclock 时钟 GPIO 配置
csi0_hsync	hsync 信号 GPIO 配置
csi0_vsync	vsync 信号 GPIO 配置
csi0_d<X>	csi 数据引脚 GPIO 配置
示例:	
[vind0/csi0]	
csi0_used	= 1
csi0_pck	= port:PE00<2><default><default><default>
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>

17.3. [vind<X>/csi_cci<X>]

配置项	配置项含义
csi_cci0_used	csi 的 cci 使能配置
csi_cci0_sck	cci 的 i2c 通信 sck GPIO 配置
csi_cci0_sda	cci 的 i2c 通信 sda GPIO 配置
示例:	
[vind0/csi_cci0]	
csi_cci0_used	= 1
csi_cci0_sck	= port:PE12<2><default><default><default>
csi_cci0_sda	= port:PE13<2><default><default><default>

17.4. [vind<X>/flash<X>]

配置项	配置项含义
flash0_used	vin 框架对应的闪光灯使能配置
flash0_type	闪光灯的类型
flash0_en	闪光灯使能
flash0_mode	闪光灯工作模式
flash0_flvdd	闪光灯电压配置
flash0_flvdd_vol	闪光灯电压值
示例:	
<pre>[vind0/flash0] flash0_used = 1 flash0_type = 2 flash0_en = flash0_mode = flash0_flvdd = "" flash0_flvdd_vol =</pre>	

17.5. [vind<X>/actuator<X>]

配置项	配置项含义
actuator0_used	对焦马达使能配置
actuator0_name	对焦马达名称
actuator0_slave	对焦马达的 i2c 地址
actuator0_af_pwdn	对焦马达的 pwm 控制
actuator0_afvdd	对焦马达电压配置
actuator0_afvdd_vol	对焦马达电压配置值
示例:	
<pre>[vind0/actuator0] actuator0_used = 0 actuator0_name = "ad5820_act" actuator0_slave = 0x18 actuator0_af_pwdn = actuator0_afvdd = "afvcc-csi" actuator0_afvdd_vol = 2800000</pre>	

17.6. [vind<X>/sensor<X>]

配置项	配置项含义
sensor1_used	sensor 使能控制
sensor1_mname	sensor 名称, 需要和驱动文件的对应
sensor1_twi_cci_id	sensor 通信使用的 twi 索引
sensor1_twi_addr	sensor 的 i2c 地址
sensor1_pos	sensor 索引
sensor1_isp_used	sensor 的 ISP 使能
sensor1_fmt	sensor 的数据格式
sensor1_stby_mode	sensor 的 stby 模式选择
sensor1_vflip	sensor 垂直镜像使能配置
sensor1_hflip	sensor 水平镜像使能配置
sensor1_iovdd	sensor io 电压配置
sensor1_iovdd_vol	sensor io 电压值
sensor1_avdd	sensor avdd 电压配置
sensor1_avdd_vol	sensor avdd 电压值
sensor1_dvdd	sensor dvdd 电压配置
sensor1_dvdd_vol	sensor dvdd 电压值
sensor1_power_en	sensor 电源使能
sensor1_reset	sensor reset GPIO 配置
sensor1_pwn	sensor pwn GPIO 配置
示例:	
sensor1_used	= 1
sensor1_mname	= "ov5647"
sensor1_twi_cci_id	= 0
sensor1_twi_addr	= 0x6c
sensor1_pos	= "front"
sensor1_isp_used	= 0
sensor1_fmt	= 0
sensor1_stby_mode	= 1
sensor1_vflip	= 0
sensor1_hflip	= 0
sensor1_iovdd	= "iovdd-csi"
sensor1_iovdd_vol	= 2800000
sensor1_avdd	= "avdd-csi"
sensor1_avdd_vol	= 2800000
sensor1_dvdd	= "dvdd-csi"
sensor1_dvdd_vol	= 1800000
sensor1_power_en	=
sensor1_reset	= port:PE16<0><0><1><0>
sensor1_pwn	= port:PE17<0><0><1><0>

17.7. [vind<X>/vinc<X>]

配置项	配置项含义
vinc<X>_used	vin core 使能配置
vinc<X>_csi_sel	vin core 对应的 csi 索引
vinc<X>_mipi_sel	vin core 对应的 mipi 索引
vinc<X>_isp_sel	vin core 对应的 isp 索引

vinc<X>_rear_sensor_sel	vin core 对应的 rear sensor 索引
vinc<X>_front_sensor_sel	vin core 对应的 front sensor 索引
vinc<X>_sensor_list	vin core 对应的 sensor 列表

示例:

```
[vind0/vinc1]
vinc1_used      = 1
vinc1_csi_sel   = 0
vinc1_mipi_sel  = 0xff
vinc1_isp_sel   = 0
vinc1_rear_sensor_sel = 0
vinc1_front_sensor_sel = 1
vinc1_sensor_list = 0
```



18. 摄像头(CSI)

18.1. [csi<X>]

配置项	配置项含义
csi<X>_used	摄像头使能配置
csi<X>_sensor_list	
csi<X>_pck	pclk 信号的 GPIO 配置。
csi<X>_mck	mclk 信号的 GPIO 配置。
csi<X>_hsync	hsync 信号的 GPIO 配置
csi<X>_vsync	vsync 信号的 GPIO 配置
csi<X>_d<X>	csi d<X> 信号的 GPIO 配置

示例:

```
[csi0]
csi0_used                = 1
csi0_sensor_list        = 0
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>
```

18.2. [csi<X>/csi0_dev0]

配置项	配置项含义
csi<X>_dev0_used	是否使用 csi0_dev0
csi<X>_dev0_mname	设置 sensor 0 名称
csi<X>_dev0_twi_addr	请参考实际原理图填写。
csi<X>_dev0_twi_id	请参考实际模组的 8bit ID 填写
csi<X>_dev0_pos	摄像头位置前置填"front",后置填"rear"。
csi<X>_dev0_isp_used	YUV 填 0。
csi<X>_dev0_fmt	YUV 填 0。
csi<X>_dev0_stby_mode	填 0。
csi<X>_dev0_vflip	Sensor 图像垂直翻转。
csi<X>_dev0_hflip	Sensor 图像水平翻转。
csi<X>_dev0_iovdd	IOVDD 配置,请参考实际原理图填写
csi<X>_dev0_iovdd_vol	IOVDD 电压值一般为 2.8V(2800000)。
csi<X>_dev0_avdd	AVDD 配置,如"csi-avdd"。
csi<X>_dev0_avdd_vol	AVDD 电压值,一般为 2.8V(2800000)。
csi<X>_dev0_dvdd	DVDD 配置,如"csi-dvdd"。
csi<X>_dev0_dvdd_vol	DVDD 电压值参考 datasheet,1.2/1.5/1.8V。
csi<X>_dev0_afvdd	Isp-dvdd 配置,如 isp-dvdd12
csi<X>_dev0_afvdd_vol	电压值为 1.2V。
csi<X>_dev0_power_en	Sensor power enable 引脚 GPIO 配置。
csi<X>_dev0_reset	Sensor reset 引脚 GPIO 配置。
csi<X>_dev0_pwdn	Sensor power down 引脚 GPIO 配置。
csi<X>_dev0_flash_used	填 0。
csi<X>_dev0_flash_type	填 0。
csi<X>_dev0_flash_en	不需填写
csi<X>_dev0_flash_mode	不需填写
csi<X>_dev0_flvdd	不需填写
csi<X>_dev0_flvdd_vol	不需填写
csi<X>_dev0_af_pwdn	不需填写
csi<X>_dev0_act_used	不需填写
csi<X>_dev0_act_name	不需填写
csi<X>_dev0_act_slave	不需填写

示例:

```
[csi0/csi0_dev0]
csi0_dev0_used          = 1
csi0_dev0_mname        = "ov5640"
csi0_dev0_twi_addr     = 0x78
csi0_dev0_twi_id      = 4
csi0_dev0_pos          = "rear"
csi0_dev0_isp_used     = 0
csi0_dev0_fmt          = 0
csi0_dev0_stby_mode    = 0
csi0_dev0_vflip        = 0
csi0_dev0_hflip        = 0
csi0_dev0_iovdd        = "csi-iovcc"
csi0_dev0_iovdd_vol    = 2800000
csi0_dev0_avdd         = "csi-avdd"
```



```
csi0_dev0_avdd_vol      = 2800000
csi0_dev0_dvdd         = "csi-dvdd"
csi0_dev0_dvdd_vol     = 1500000
csi0_dev0_afvdd        = "csi-afvcc"
csi0_dev0_afvdd_vol    = 2800000
csi0_dev0_power_en     =
csi0_dev0_reset        = port:PI07<1><0><1><0>
csi0_dev0_pwdn         = port:PI06<1><0><1><0>
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     = "ad5820_act"
csi0_dev0_act_slave    = 0x18
```



19.tvout/tvin

19.1. [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
示例:	
[tvout_para]	
tvout_used	=
tvout_channel_num	=
tv_en	=

19.2. [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
示例:	
[tvin_para]	
tvin_used	=
tvin_channel_num	=
tvin_en	=

19.3. [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
示例:	
[di]	
di_used	= 1

20.SD / MMC

20.1. [sdc<X>]

配置项	配置项含义
sdc<X>_used	SDC 使用控制：1 使用，0 不用
bus-width	位宽:1-1bit,4-4bit
sdc<X>_d<X>	SDC DATA<X>的 GPIO 配置
sdc<X>_clk	SDC CLK 的 GPIO 配置
sdc<X>_cmd	SDC CMD 的 GPIO 配置
sdc<X>_d<X>	SDC DATA<X>的 GPIO 配置
sd-uhs-sdr50	
sd-uhs-ddr50	
sd-uhs-sdr104	
broken-cd	
cd-inverted	
non-removable	
sdc<X>_emmc_rst	
cd-gpios	SDC 卡检测信号的 GPIO 配置
card-pwr-gpios	
data3-detect	
sunxi-power-save-mode	SDC CLK 信号无数据传输时暂停
sunxi-dis-signal-vol-sw	
mmc-ddr-1_8v	
mmc-hs200-1_8v	
mmc-hs400-1_8v	
max-frequency	
sdc_tm4_sm0_freq0	
sdc_tm4_sm0_freq1	
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	SDC 供电电源配置
vqmmc	SDC IO 供电电源配置
vdmcc	是否是 sdio card,0:不是, 1: 是
示例:	
[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:PH13<0><1><2><default>
sunxi-power-save-mode	=

vmmc	= "vcc-sdev"
vqmmc	= "vcc-sdevq33"
vdmmc	= "vcc-sdevd"

20.2. [smc]

配置项	配置项含义
smc_used	1:使用 0:不使用
smc_rst	
smc_vppen	
smc_vppp	
smc_det	
smc_vccen	
smc_sck	
smc_sda	

示例:

```
[smc]
smc_used      =
smc_rst       =
smc_vppen     =
smc_vppp     =
smc_det       =
smc_vccen    =
smc_sck       =
smc_sda       =
```



21.[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

示例:

```
[gpio_para]
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"
```



22.USB 控制标志

22.1. [usbc<X>]

配置项	配置项含义
usb_used	USB 使能标志(xx=1 or 0)。 1 表示系统中 USB 模块可用, 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_detect_mode	usb otg 的检测方法, 0-thread scan, 1-id gpio interrupt
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_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
示例:	
<pre>[usbc0] usb0_used = 1 usb_port_type = 2 usb_detect_type = 1 usb_detect_mode = 0 usb_id_gpio = port:PI4<0><1><default><default> usb_det_vbus_gpio = port:PI8<0><1><default><default> 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"</pre>	

23.[serial_feature]

配置项	配置项含义
sn_filename	该配置的名字
示例:	
[serial_feature] sn_filename	= "ULI/factory/snum.txt"



24.重力感应(G Sensor)

24.1. [gsensor_para]

配置项	配置项含义
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
示例:	
<pre>[gsensor_para] gsensor_used = 1 gsensor_twi_id = 2 gsensor_twi_addr = 0x18 gsensor_int1 = port:PA09<6><1><<default><default> gsensor_int2 =</pre>	

24.2. [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
示例:	
<pre>[gsensor_list_para] compatible = "allwinner,sun50i-gsensor-list-para" gsensor_list__used = 1 da380 = 1</pre>	

25. WiFi

25.1. [wlan]

配置项	配置项含义
wlan_used	是否要使用 wifi
compatible	wlan 名称
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_busnum	所使用的 SDIO 号,如使用的是 SDIO1,则此值为 1
wlan_regon	Wifi 使能脚
wlan_hostwake	wifi 唤醒主控脚
wlan_clk_gpio	wifi 模块 32K 时钟输出硬件

示例:

```
[wlan]
wlan_used          = 1
compatible         = "allwinner,sunxi-wlan"
clocks             = "outa"
wlan_power         = "vcc-wifi"
wlan_io_regulator  = "vcc-io-wifi"
wlan_busnum        = 1
wlan_regon         = port:PG10<1><1><1><0>
wlan_hostwake      = port:ower0<0><default><default><default>
```

26. 蓝牙(bluteeth)

26.1. [bt]

配置项	配置项含义
bt_used	蓝牙使用控制:1 使用,0 不用
compatible	"allwinner,sunxi-bt"
clocks	低功耗时钟,此值固定为&clk_outa
clock_io	32K 时钟的 clock io。
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	uart 电平转换芯片使能脚
示例:	
[bt]	
bt_used	= 1
compatible	= "allwinner,sunxi-bt"
clocks	= "outa"
pinctrl-names	= "default"
clock_io	= port:PI12<4><0><0><0>
bt_power	= "vcc-wifi"
bt_io_regulator	= "vcc-io-wifi"
bt_rst_n	= port:PH12<1><1><1><0>

26.2. [btlpm]

配置项	配置项含义
btlpm_used	蓝牙使用控制: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_host_wake	bt 唤醒主控引脚
示例:	
[btlpm]	
btlpm_used	= 0
compatible	= "allwinner,sunxi-btlpm"
uart_index	= 3
bt_wake	= port:PG11<1><1><1><0>
bt_host_wake	= port:power1<0><default><default><default>

27. 光感(light sensor)

27.1. [ls_para]

配置项	配置项含义
ls_used =xx	是否支持 ls
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
示例:	
[ls_para]	
ls_used	= 0
ls_twi_id	= 1
ls_twi_addr	= 0x23
ls_int	= port:PB07<4><1><default><default>



28. 陀螺仪传感器(gyroscope sensor)

28.1. [gy_para]

配置项	配置项含义
gy_used	是否支持 gyroscope
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
示例:	
[gy_para]	
gy_used	= 1
gy_twi_id	= 2
gy_twi_addr	= 0x6a
gy_int1	= port:PA10<6><1><default><default>
gy_int2	=



29. 罗盘 Compass

29.1. [compass_para]

配置项	配置项含义
compass_used	是否支持 compass
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	中断的 GPIO 配置
示例:	
[compass_para]	
compass_used	= 1
compass_twi_id	= 2
compass_twi_addr	= 0x0d
compass_int	= port:PA11<6><1><default><default>



30. 数字音频总线（SPDIF）

30.1. [spdif] & [sndspdif]

配置项	配置项含义
spdif_used	是否使用 R40 SPDIF 模块 0x1:使用,0x0:不使用
sndspdif_used	是否使用 R40 SPDIF 模块 0x1:使用,0x0:不使用
示例:	
[spdif] spdif_used	= 0
[sndspdif] sndspdif_used	= 0
注意:	
由于历史原因而存在功能相同的两项，两项需同时配置	

30.2. [audiohdm] & [sndhdm]

配置项	配置项含义
audiohdm_used	是否使用 V40 HDMI 音频模块 0x1:使用,0x0:不使用
sndhdm_used	是否使用 V40 HDMI 音频模块 0x1:使用,0x0:不使用
示例:	
[audiohdm] audiohdm_used	= 0
[sndhdm] sndhdm_used	= 0
注意:	
由于历史原因而存在功能相同的两项，两项需同时配置	

30.3. [snddaudio<X>]

配置项	配置项含义
snddaudio0_used	是否使用 V40 DAUDIO(I2S/PCM)音频模块 0x1:使用,0x0:不使用
示例:	
[snddaudio0] snddaudio0_used	= 0

30.4. [daudio<X>]

配置项	配置项含义
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	0: 16bit linear PCM; 1: 8bit linear PCM; 2: 8bit u-law; 3: 8bit a-law
rx_data_mode	0: 16bit linear PCM; 1: 8bit linear PCM; 2: 8bit u-law; 3: 8bit a-law
daudio_master	1: SND_SOC_DAIFMT_CBM_CFM(codec clk & FRM master) use; 2: SND_SOC_DAIFMT_CBS_CFM(codec clk slave & FRM master) not use; 3: SND_SOC_DAIFMT_CBM_CFS(codec clk master & frame slave) not use; 4: SND_SOC_DAIFMT_CBS_CFS(codec clk & FRM slave) use
audio_format	1: SND_SOC_DAIFMT_I2S(standard i2s format). use; 2: SND_SOC_DAIFMT_RIGHT_J(right justified format).; 3: SND_SOC_DAIFMT_LEFT_J(left justified format); 4: SND_SOC_DAIFMT_DSP_A(pcm. MSB is available on 2nd BCLK rising edge after LRC rising edge). use; 5: SND_SOC_DAIFMT_DSP_B(pcm. MSB is available on 1nd BCLK rising edge after LRC rising edge)
signal_inversion	1: SND_SOC_DAIFMT_NB_NF(normal bit clock + frame) use; 2: SND_SOC_DAIFMT_NB_IF(normal BCLK + inv FRM); 3: SND_SOC_DAIFMT_IB_NF(invert BCLK + nor FRM) use; 4: SND_SOC_DAIFMT_IB_IF(invert BCLK + FRM)
frametype	0: short frame = 1 clock width; 1: long frame = 2 clock width
tdm_config	0: pcm 1:i2s
mclk_div	0: not output(normal setting this) 1/2/4/6/8/12/16/24/32/48/64/96/128/176/192
daudio0_used	0: not use 1: use

示例:

```
[daudio0]
pcm_lrck_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
mclk_div              = 0x0
daudio0_used         = 0
```

31. 内置音频 codec

31.1. [sndcodec]

配置项	配置项含义
sndcodec_used	是否使用 R40 模拟音频输入输出 0x1:使用,0x0:不使用
aif2fmt	
aif3fmt	
aif2master	
hp_detect_case	
示例:	
[sndcodec] sndcodec_used = 0x1	

31.2. [i2s]

配置项	配置项含义
i2s_used	
示例:	
i2s_used = 0x1	

31.3. [codec]

配置项	配置项含义
codec_used	codec 是否使用, 1: 打开 (默认) 0: 关闭
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
hp_dirused	使能耳机 0:不使用;1:使用
headsetmicgain	
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
aif2config	
aif3config	
gpio-spk	喇叭 gpio 口配置
示例:	
[codec] 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:PG11<2><1><default><default>	

32.[s_cir0]

配置项	配置项含义
s_cir0_used	
ir_power_key_code	
ir_addr_code0	
ir_addr_cnt	

示例:

```
[s_cir0]
s_cir0_used      = 1
ir_power_key_code = 0x0
ir_addr_code0    = 0x04
ir_addr_cnt      = 0x1
```



33.PMU 电源

33.1. [pmu<X>]

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,0:not wake up 1:wakeup
pmu_hot_shutdown	是否允许 pmu 高温关机
pmu_inshort	启动是否检测电池电量
示例:	
[pmu0]	
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
pmu_start	= 0

33.2. [charger<X>]

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_bat_unused	是否使用电池,1:不使用,0:使用
pmu_chg_ic_temp	是否开启充电智能温度检测, 0 关闭, 1 开启
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_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	CHGLED 引脚控制。0:PMU 1:充电器
pmu_chgled_type	CHGLED 类型。0:Type A 1:Type B
pmu_ocv_en	
pmu_cou_en	
pmu_update_min_time	
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
power_start	<p>当充电状态下的关机动作。1:关机; 非 1:重启</p> <p>当有接入电池的情况下，插入外部电源时：</p> <p>0: 关机状态下，插入外部电源时，电池电量充足时，不允许开机，会进入充电模式；电池电量不足，则关机</p> <p>1: 关机状态下，插入外部电源，电池电量充足时，直接开机进入系统；电池电量不足，则关机</p> <p>2: 关机状态下，插入外部电源时，不允许开机，会进入充电模式；无视电池电量</p> <p>3: 关机状态下，插入外部电源，直接开机进入系统；无视电池电量。</p>

示例:

```
[charger0]
compatible = "axp221s-charger"
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_htf	= 388
pmu_bat_shutdown_ltf	= 3200
pmu_bat_shutdown_htf	= 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

33.3. [powerkey<X>]

配置项	相关说明
compatible	设备名字
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
pmu_hot_shutdown	是否允许 pmu 高温关机
pmu_inshort	启动是否检测电池电量
示例:	
[powerkey0]	
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

33.4. [regulator<X>]

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
regulator16	regulator16 对应的别名,请勿修改
regulator17	regulator17 对应的别名,请勿修改
regulator18	regulator18 对应的别名,请勿修改
regulator19	regulator19 对应的别名,请勿修改
regulator20	regulator20 对应的别名,请勿修改
示例:	
[regulator0]	
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-pe vcc-pd vcc-3v vcc-tvout vcc-tvin vcc-emmev vcc-sdev
vcc-sdevq33 vcc-sdevd vcc-nand vcc-sdev-p3 vcc-sdevq33-p3 vcc-sdevd-p3"
regulator2      = "axp221s_dcdc2 none vdd-cpua"
regulator3      = "axp221s_dcdc3 none vdd-sys vdd-gpu"
regulator4      = "axp221s_dcdc4 none"
regulator5      = "axp221s_dcdc5 none vcc-dram"
regulator6      = "axp221s_rtc none vcc-rtc"
regulator7      = "axp221s_aldo1 none vcc-25 csi-avdd"
regulator8      = "axp221s_aldo2 none vcc-pa ehpy-vdd25"
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"
regulator13     = "axp221s_dldo4 none vdd-sata-25 vcc-pf"
regulator14     = "axp221s_eldo1 none vcc-pe csi-iovc csi-afvcc"
regulator15     = "axp221s_eldo2 none csi-dvdd"
regulator16     = "axp221s_eldo3 none vdd-sata-12"
regulator17     = "axp221s_ldoio0 none vcc-ctp"
regulator18     = "axp221s_ldoio1 none vcc-i2s-18"
regulator19     = "axp221s_dc1sw none ephy-dvdd33"
regulator20     = "axp221s_dc5ldo none"

```

33.5. [axp_gpio<X>]

配置项	相关说明
compatible	设备名
示例:	
[axp_gpio0] compatible	= "axp221s-gpio"

34.[psensor_table]

配置项	配置项含义
psensor_count	
prange_min_2	
prange_max_2	
prange_min_1	
prange_max_1	
prange_min_0	
prange_max_0	
示例:	
<pre>psensor_count = 3 prange_min_2 = 4800 prange_max_2 = 6500 prange_min_1 = 4500 prange_max_1 = 4800 prange_min_0 = 0 prange_max_0 = 4500</pre>	



35.DVFS

35.1. [dvfs_table]&&[dvfs_table_[X]]

配置项	配置项含义
extremity_freq	极限频率
max_freq	最大运行频率
min_freq	最小运行频率
lv_count	VF 表项数
lvn_freq	对应的最大频率(n 表示级数)
lvn_volt	第 n 级的电压
示例:	
<pre>[dvfs_table] 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</pre>	
注意:	
vf 表(电压频率对应表)关乎系统稳定性,请勿私自修改!	

36.s_uart<X>

配置项	配置项含义
s_uart_used	是否使用 cpus 的 uart 模块：0-否，1-是
s_uart_tx	cpus TX GPIO 配置
s_uart_rx	cpus RX GPIO 配置
示例:	
[s_uart0]	
s_uart_used	= 1
s_uart_tx	= port:PL02<2><default><default><default>
s_uart_rx	= port:PL03<2><default><default><default>



37.s_twi<X>

配置项	配置项含义
s_twi0_used	0-否, 1-是
s_twi0_sck	cpus i2c sck GPIO 配置
s_twi0_sda	cpus i2c sda GPIO 配置
示例:	
[s_twi0]	
s_twi0_used	= 1
s_twi0_sck	= port:PL00<2><1><2><default>
s_twi0_sda	= port:PL01<2><1><2><default>



38.s_jtag<X>

配置项	配置项含义
s_jtag_used	0-否, 1-是
s_jtag_tms	cpus jtag 模式选择输入 GPIO 配置
s_jtag_tck	cpus jtag 时钟选择输入 GPIO 配置
s_jtag_tdo	cpus jtag 数据输出 GPIO 配置
s_jtag_tdi	cpus jtag 数据输入 GPIO 配置
示例:	
s_jtag_used	= 0
s_jtag_tms	= port:PL04<2><1><2><default>
s_jtag_tck	= port:PL05<2><1><2><default>
s_jtag_tdo	= port:PL06<2><1><2><default>
s_jtag_tdi	= port:PL07<2><1><2><default>



39.Virtual device

39.1. [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
示例:	
[Vdevice]	
Vdevice_used	= 1
Vdevice_0	= port:PB00<4><1><2><default>
Vdevice_1	= port:PB01<4><1><2><default>



40. 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 acknowledgment 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.

