

A20 System Configuration

V1.0

2013-03-15

Revision History

Version	Date	Section/ Page	Changes
V1.0	2013-03-15		Initial version



目录

目录.....	3
1 系统(System).....	7
1.1 [platform].....	7
1.2 [target].....	7
1.3 [pm_para].....	7
1.4 [card_boot].....	8
1.5 [card0_boot_para].....	8
1.6 [card2_boot_para].....	9
1.7 [twi_para].....	9
1.8 [uart_para].....	9
1.9 [uart_force_debug].....	10
1.10 [jtag_para].....	10
1.11 [clock].....	11
2 SDRAM.....	12
2.1 [dram_para].....	12
3 EMAC.....	14
4 GMAC.....	16
4.1 [gmac_para].....	16
5 I2C 总线.....	18
5.1 [twi0_para].....	18
5.2 [twi1_para].....	18
5.3 [twi2_para].....	18
5.4 [twi3_para].....	19
6 UART 总线.....	20
6.1 [uart_para0].....	20
6.2 [uart_para1].....	20
6.3 [uart_para2].....	21
6.4 [uart_para3].....	21
6.5 [uart_para4].....	22
6.6 [uart_para5].....	22
6.7 [uart_para6].....	23
6.8 [uart_para7].....	23
7 SPI 总线.....	25
7.1 [spi0_para].....	25
7.2 [spi1_para].....	25
7.3 [spi2_para].....	26

7.4	[spi3_para].....	26
7.5	[spi_devices].....	27
7.6	[spi_board0].....	27
8	电阻屏(rtp).....	28
8.1	[rtp_para].....	28
9	电容屏(ctp).....	29
9.1	[ctp_para].....	29
10	触摸按键(touch key).....	30
10.1	[tkey_para].....	30
11	马达(motor).....	31
11.1	[motor_para].....	31
12	闪存(nand flash).....	32
12.1	[nand_para].....	32
13	显示初始化(disg init).....	34
13.1	[disg_init].....	34
14	LCD 屏 0.....	36
14.1	[lcd0_para].....	36
15	LCD 屏 1.....	40
15.1	[lcd1_para].....	40
16	sata.....	41
17	TV.....	42
17.1	[tv_out_dac_para].....	42
17.2	[tvout_para].....	42
17.3	[tvinn_para].....	42
18	HDMI.....	44
18.1	[hdmi_para].....	44
19	摄像头(CSI).....	45
19.1	[camera_list_para].....	45
19.2	[csi0_para].....	46
19.3	[csi1_para].....	46
20	SD/MMC.....	51
20.1	[mmc0_para].....	51
20.2	[mmc1_para].....	52
20.3	[mmc2_para].....	52
20.4	[mmc3_para].....	53
21	SIM 卡.....	55
21.1	[smc_para].....	55
22	USB 控制标志.....	56
22.1	[usb0].....	56
22.2	[usb1].....	57



22.3	[usbc2].....	58
23	USB Device.....	59
23.1	[usb_feature].....	59
23.2	[msc_feature].....	59
24	重力感应(G-Sensor).....	60
24.1	[gsensor_para].....	60
25	WIFI.....	61
25.1	[wifi_para].....	61
25.2	sdio 接口 wifi rtl8723as demo.....	61
25.3	usb 接口 wifi rtl8188eu demo.....	62
26	3G.....	63
26.1	[3g_para].....	63
27	gyroscope.....	64
27.1	[gy_para].....	64
28	光感(light sensor).....	65
28.1	[ls_para].....	65
29	罗盘(Compass).....	66
29.1	[compass_para].....	66
30	蓝牙(blueetooth).....	67
30.1	[bt_para].....	67
31	数字音频总线(I2S).....	68
31.1	[i2s_para].....	68
32	数字音频总线(S/PDIF).....	69
32.1	[spdif_para].....	69
33	内置音频(codec).....	70
33.1	[audio_para].....	70
34	红外(ir).....	71
34.1	[ir_para].....	71
35	PMU 电源.....	72
35.1	[pmu_para].....	72
36	动态电压频率(dvfs).....	77
36.1	[dvfs_table].....	77

备注

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. GPIO 配置的形式描述：
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；

1 系统(System)

1.1 [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)

配置举例:

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

1.2 [target]

配置项	配置项含义
boot_clock=xx	启动频率, xx 表示多少 MHZ
dcdc2_vol=1400	Dcdc2(CPU)的输出电压, mV
dcdc3_vol=1400	Dcdc3(GPU)的输出电压, mV
storage_type = -1	启动介质选择 0: nand 1: card0 2: card2 -1: (default) 自动扫描启动介质:

配置举例:

```
[target]
boot_clock = 1008
dcdc1_vol = 300
dcdc2_vol = 1400
dcdc3_vol = 1250
storage_type = -1
```

1.3 [pm_para]

配置项	配置项含义
-----	-------

standby_mode = 1	1: super standby 2: normal standby
------------------	---------------------------------------

配置举例:

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

1.4 [card_boot]

配置项	配置项含义
logical_start=40960	启动卡逻辑起始扇区
sprite_gpio0=	卡量产 gpio led 灯配置

配置举例:

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

1.5 [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=1	速度模式 0 为低速, 1 为高速
card_line=4	代表 4 线卡
sdc_d1= port:PF0<2><1><default><default>	sdc 卡数据 1 线信号的 GPIO 配置
sdc_d0= port:PF1<2><1><default><default>	sdc 卡数据 0 线信号的 GPIO 配置
sdc_clk= port:PF2<2><1><default><default>	sdc 卡时钟信号的 GPIO 配置
sdc_cmd= port:PF3<2><1><default><default>	sdc 命令信号的 GPIO 配置
sdc_d3= port:PF4<2><1><default><default>	sdc 卡数据 3 线信号的 GPIO 配置
sdc_d2= port:PF5<2><1><default><default>	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><default><default>
sdc_d0             = port:PF1<2><1><default><default>
sdc_clk            = port:PF2<2><1><default><default>
sdc_cmd            = port:PF3<2><1><default><default>
sdc_d3             = port:PF4<2><1><default><default>
sdc_d2             = port:PF5<2><1><default><default>
```


sdc_d3 = port:PF4<2><1><default><default>
sdc_d2 = port:PF5<2><1><default><default>

1.6 [card2_boot_para]

配置项	配置项含义
card_ctrl=2	卡启动控制器选择 2
card_high_speed=xx	速度模式 0 为低速, 1 为高速
card_line=4	4 线卡
sdc_cmd = port:PC6<3><1>	sdc 命令信号的 GPIO 配置
sdc_clk = port:PC7<3><1>	sdc 卡时钟信号的 GPIO 配置
sdc_d0 = port:PC8<3><1>	sdc 卡数据 0 线信号的 GPIO 配置
sdc_d1 = port:PC9<3><1>	sdc 卡数据 1 线信号的 GPIO 配置
sdc_d3= port:PC10<3><1>	sdc 卡数据 3 线信号的 GPIO 配置
sdc_d2= port:PC11<3><1>	sdc 卡数据 2 线信号的 GPIO 配置

配置举例:

card_ctrl = 2
card_high_speed = 1
card_line = 4
sdc_cmd = port:PC6<3><1>
sdc_clk = port:PC7<3><1>
sdc_d0 = port:PC8<3><1>
sdc_d1 = port:PC9<3><1>
sdc_d2 = port:PC10<3><1>
sdc_d3 = port:PC11<3><1>

1.7 [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

配置举例:

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

1.8[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

配置举例:

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

1.9[uart_force_debug]

配置项	配置项含义
uart_debug_port	
uart_debug_tx	
uart_debug_rx	

配置举例:

```
[uart_force_debug]
uart_debug_port      = 0
uart_debug_tx       = port:PF2<4><1><default><default>
uart_debug_rx       = port:PF4<4><1><default><default>
```

1.10[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

配置举例:

```
[jtag_para]
jtag_enable          = 1
jtag_ms              = port:PB14<3>
```

jtag_ck = port:PB15<3>
jtag_do = port:PB16<3>
jtag_di = port:PB17<3>

1.11 [clock]

配置项	配置项含义
pll3 =297	Video0 时钟频率
pll4 =300	Ve 时钟频率
pll6 =600	Peripherals 时钟频率
pll7 =297	Video1 时钟频率
pll8 =360	GPU（通信）时钟频率

配置举例：

[clock]

pll3 = 297
pll4 = 300
pll6 = 600
pll7 = 297
pll8 = 360

2 SDRAM

2.1 [dram_para]

配置项	配置项含义
dram_baseaddr =xx	DRAM 的基地址
dram_clk =xx	DRAM 的时钟频率, 单位为 MHz; 它为 24 的整数倍, 最低不得低于 120,
dram_type =xx	DRAM 类型: 2 为 DDR2 3 为 DDR3
dram_rank_num =xx	DRAM 控制器内部参数, 由原厂来进行调节, 请勿修改
dram_chip_density =xx	DRAM 控制器内部参数, 由原厂来进行调节, 请勿修改
dram_io_width =xx	DRAM 控制器内部参数, 由原厂来进行调节, 请勿修改
dram_bus_width =xx	DRAM 控制器内部参数, 由原厂来进行调节, 请勿修改
dram_cas =xx	DRAM 控制器内部参数, 由原厂来进行调节, 请勿修改
dram_zq=xx	DRAM 控制器内部参数, 由原厂来进行调节, 请勿修改
dram_odt_en=xx	DRAM 控制器内部参数, 由原厂来进行调节, 请勿修改
dram_size =xx	DRAM 控制器内部参数, 由原厂来进行调节, 请勿修改
dram_tpr0=xx	DRAM 控制器内部参数, 由原厂来进行调节, 请勿修改
dram_tpr1=xx	DRAM 控制器内部参数, 由原厂来进行调节, 请勿修改
dram_tpr2=xx	DRAM 控制器内部参数, 由原厂来进行调节, 请勿修改
dram_tpr3=xx	DRAM 控制器内部参数, 由原厂来进行调节, 请勿修改
dram_tpr4=xx	DRAM 控制器内部参数, 由原厂来进行调

	节, 请勿修改
dram_tpr5=xx	DRAM 控制器内部参数, 由原厂来进行调节, 请勿修改
dram_emr1 =xx	DRAM 控制器内部参数, 由原厂来进行调节, 请勿修改
dram_emr2 =xx	DRAM 控制器内部参数, 由原厂来进行调节, 请勿修改
dram_emr3 =xx	DRAM 控制器内部参数, 由原厂来进行调节, 请勿修改

配置举例:

[dram_para]

```

dram_baseaddr      = 0x40000000
dram_clk           = 408
dram_type          = 3
dram_rank_num      = 0xffffffff
dram_chip_density  = 0xffffffff
dram_io_width      = 0xffffffff
dram_bus_width     = 0xffffffff
dram_cas           = 9
dram_zq            = 0x7f
dram_odt_en        = 0
dram_size          = 0xffffffff
dram_tpr0          = 0x42d899b7
dram_tpr1          = 0xa090
dram_tpr2          = 0x22a00
dram_tpr3          = 0x0
dram_tpr4          = 0x0
dram_tpr5          = 0x0
dram_emr1          = 0x4
dram_emr2          = 0x10
dram_emr3          = 0x0

```

3 EMAC

配置项	配置项含义
emac_used	
emac_rxd3	
emac_rxd2	
emac_rxd1	
emac_rxd0	
emac_txd3	
emac_txd2	
emac_txd1	
emac_txd0	
emac_rxclk	
emac_rxerr	
emac_rxdV	
emac_mdc	
emac_mdio	
emac_txen	
emac_txclk	
emac_crs	
emac_col	
emac_reset	

配置举例：

```
[emac_para]
emac_used          = 0
emac_rxd3          = port:PA00<2><default><default><default>
emac_rxd2          = port:PA01<2><default><default><default>
emac_rxd1          = port:PA02<2><default><default><default>
emac_rxd0          = port:PA03<2><default><default><default>
emac_txd3          = port:PA04<2><default><default><default>
emac_txd2          = port:PA05<2><default><default><default>
emac_txd1          = port:PA06<2><default><default><default>
emac_txd0          = port:PA07<2><default><default><default>
emac_rxclk         = port:PA08<2><default><default><default>
emac_rxerr         = port:PA09<2><default><default><default>
emac_rxdV         = port:PA10<2><default><default><default>
```



emac_mdc	= port:PA11<2><default><default><default>
emac_mdio	= port:PA12<2><default><default><default>
emac_txen	= port:PA13<2><default><default><default>
emac_txclk	= port:PA14<2><default><default><default>
emac_crs	= port:PA15<2><default><default><default>
emac_col	= port:PA16<2><default><default><default>
emac_reset	= port:PA17<1><default><default><default>

Confidential

4 GMAC

4.1 [gmac_para]

配置项	配置项含义
gmac_used=0	Gmac 模块是否使能: 1: enable 0: disable
gmac_txd0=xx	Gmac tx0 的 GPIO 配置
gmac_txd1=xx	Gmac tx1 的 GPIO 配置
gmac_txd2=xx	Gmac tx2 的 GPIO 配置
gmac_txd3=xx	Gmac tx3 的 GPIO 配置
gmac_txd4=xx	Gmac tx4 的 GPIO 配置
gmac_txd5=xx	Gmac tx5 的 GPIO 配置
gmac_txd6=xx	Gmac tx6 的 GPIO 配置
gmac_txd7=xx	Gmac tx7 的 GPIO 配置
gmac_txclk=xx	Gmac MII 接口发送时钟
gmac_txen=xx	Gmac 发送使能 GPIO 配置
gmac_gtxclk=xx	Gmac GMII 接口发送时钟
gmac_rxd0=xx	Gmac rx0 的 GPIO 配置
gmac_rxd1=xx	Gmac rx1 的 GPIO 配置
gmac_rxd2=xx	Gmac rx2 的 GPIO 配置
gmac_rxd3=xx	Gmac rx3 的 GPIO 配置
gmac_rxd4=xx	Gmac rx4 的 GPIO 配置
gmac_rxd5=xx	Gmac rx5 的 GPIO 配置
gmac_rxd6=xx	Gmac rx6 的 GPIO 配置
gmac_rxd7=xx	Gmac rx7 的 GPIO 配置
gmac_rxdv=xx	Gmac 接收数有效使能
gmac_rxclk=xx	Gmac 接收时钟
gmac_txerr=xx	Gmac 发送错误使能
gmac_rxerr=xx	Gmac 接收错误使能
gmac_col=xx	Gmac 冲突检测(仅用于半双工)
gmac_crs=xx	Gmac 载波监测(仅用于半双工)
gmac_clkin=xx	Gmac GMII 外部时钟
gmac_mdc=xx	Gmac 配置接口时钟
gmac_mdio=xx	Gmac 配置接口数据 I/O

配置举例:



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

5 I2C 总线

主控有 4 个 I2C (twi) 控制器

5.1 [twi0_para]

配置项	配置项含义
twi0_used =xx	TWI 使用控制：1 使用，0 不用
twi0_scl =xx	TWI SCK 的 GPIO 配置
twi0_sda=xx	TWI SDA 的 GPIO 配置

配置举例：

```
twi0_used          = 1
twi0_scl           = port:PH14<2><default><default><default>
twi0_sda           = port:PH15<2><default><default><default>
```

5.2 [twi1_para]

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

配置举例：

```
[twi1_para]
twi1_used          = 1
twi1_scl           = port:PH16<2><default><default><default>
twi1_sda           = port:PH17<2><default><default><default>
```

5.3 [twi2_para]

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

配置举例:

```
[twi2_para]
twi2_used          = 1
twi2_scl           = port:PH18<2><default><default><default>
twi2_sda           = port:PH19<2><default><default><default>
```

5.4[twi3_para]

配置项	配置项含义
twi3_used =xx	TWI 使用控制: 1 使用, 0 不用
twi3_scl =xx	TWI SCK 的 GPIO 配置
twi3_sda=xx	TWI SDA 的 GPIO 配置

配置举例:

```
[twi2_para]
twi2_used          = 1
twi2_scl           = port:PB05<4><default><default><default>
twi2_sda           = port:PB06<4><default><default><default>
```

6 UART 总线

主控有 8 路 uart 总线，其中 uart1 支持完整的 8 线通讯，而其他 7 路支持 4 线或者 2 线通讯（但十分不建议用 uart0 作为控制台以外的用途），实例中，有些路仅仅写出 2 路的配置形式，但实际使用时只要将其按照 4 路的格式补全，也能支持 4 线通讯。

6.1 [uart_para0]

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

配置举例：

[uart_para0]

```

uart_used          = 1
uart_port          = 0
uart_type          = 2
uart_tx            = port:PB22<2> <1><default><default>
uart_rx            = port:PB23<2> <1><default><default>
  
```

6.2 [uart_para1]

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

配置举例:

[uart_para1]

```

uart_used          = 0
uart_port          = 1
uart_type          = 8
uart_tx            = port:PA10<4><1><default><default>
uart_rx            = port:PA11<4><1><default><default>
uart_rts           = port:PA12<4><1><default><default>
uart_cts           = port:PA13<4><1><default><default>
uart_dtr           = port:PA14<4><1><default><default>
uart_dsr           = port:PA15<4><1><default><default>
uart_dcd           = port:PA16<4><1><default><default>
uart_ring          = port:PA17<4><1><default><default>
    
```

6.3 [uart_para2]

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

配置举例:

[uart_para2]

```

uart_used          = 0
uart_port          = 2
uart_type          = 4
uart_tx            = port:PI18<3><1><default><default>
uart_rx            = port:PI19<3><1><default><default>
uart_rts           = port:PI16<3><1><default><default>
uart_cts           = port:PI17<3><1><default><default>
    
```

6.4[uart_para3]

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

配置举例：

[uart_para3]

```

uart_used           = 0
uart_port           = 3
uart_type           = 4
uart_tx             = port:PH00<4><1><<default><<default>
uart_rx             = port:PH01<4><1><<default><<default>
uart_rts            = port:PH02<4><1><<default><<default>
uart_cts            = port:PH03<4><1><<default><<default>
  
```

6.5[uart_para4]

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

配置举例：

[uart_para4]

```

uart_used           = 0
uart_port           = 4
uart_type           = 2
uart_tx             = port:PH04<4><1><<default><<default>
uart_rx             = port:PH05<4><1><<default><<default>
  
```

6.6[uart_para5]

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

配置举例：

[uart_para5]

```

uart_used          = 0
uart_port          = 5
uart_type          = 2
uart_tx            = port:PH06<4><1><<default><default>
uart_rx            = port:PH07<4><1><<default><default>
    
```

6.7[uart_para6]

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

配置举例：

[uart_para6]

```

uart_used          = 0
uart_port          = 6
uart_type          = 2
uart_tx            = port:PA12<4><1><<default><default>
uart_rx            = port:PA13<4><1><<default><default>
    
```

6.8[uart_para7]

配置项	配置项含义
uart_used =xx	UART 使用控制：1 使用，0 不用

uart_port =xx	UART 端口号
uart_type =xx	UART 类型
uart_tx =xx	UART TX 的 GPIO 配置
uart_rx =xx	UART RX 的 GPIO 配置

配置举例:

[uart_para7]

```

uart_used          = 0
uart_port          = 7
uart_type          = 2
uart_tx            = port:PA14<4><default><default><default>
uart_rx            = port:PA15<4><default><default><default>

```


7 SPI 总线

7.1 [spi0_para]

配置项	配置项含义
spi_used =xx	SPI 使用控制：1 使用，0 不用
spi_cs_bitmap =xx	启用哪路 cs，bit 位表示
spi_cs0 =xx	SPI CS0 的 GPIO 配置
spi_cs1 =xx	SPI CS1 的 GPIO 配置
spi_sclk =xx	SPI CLK 的 GPIO 配置
spi_mosi =xx	SPI MOSI 的 GPIO 配置
spi_miso =xx	SPI MISO 的 GPIO 配置

配置举例：

[spi0_para]

```
spi_used           = 0
spi_cs_bitmap     = 1
spi_cs0           = port:PI10<2><default><default><default>
spi_cs1           = port:PI14<2><default><default><default>
spi_sclk          = port:PI11<2><default><default><default>
spi_mosi          = port:PI12<2><default><default><default>
spi_miso          = port:PI13<2><default><default><default>
```

7.2 [spi1_para]

配置项	配置项含义
spi_used =xx	SPI 使用控制：1 使用，0 不用
spi_cs_bitmap =xx	启用哪路 cs，bit 位表示
spi_cs0 =xx	SPI CS0 的 GPIO 配置
spi_cs1 =xx	SPI CS1 的 GPIO 配置
spi_sclk =xx	SPI CLK 的 GPIO 配置
spi_mosi =xx	SPI MOSI 的 GPIO 配置
spi_miso =xx	SPI MISO 的 GPIO 配置

配置举例：

[spi1_para]

```

spi_used           = 0
spi_cs_bitmap     = 1
spi_cs0           = port:PA00<4><default><default><default>
spi_sclk          = port:PA01<4><default><default><default>
spi_mosi          = port:PA02<4><default><default><default>
spi_miso          = port:PA03<4><default><default><default>

```

7.3 [spi2_para]

配置项	配置项含义
spi_used =xx	SPI 使用控制：1 使用，0 不用
spi_cs_bitmap =xx	启用哪路 cs，bit 位表示
spi_cs0 =xx	SPI CS0 的 GPIO 配置
spi_cs1 =xx	SPI CS1 的 GPIO 配置
spi_sclk =xx	SPI CLK 的 GPIO 配置
spi_mosi =xx	SPI MOSI 的 GPIO 配置
spi_miso =xx	SPI MISO 的 GPIO 配置

配置举例：

```

spi_used           = 0
spi_cs_bitmap     = 1
spi_cs0           = port:PB14<2><default><default><default>
spi_sclk          = port:PB15<2><default><default><default>
spi_mosi          = port:PB16<2><default><default><default>
spi_miso          = port:PB17<2><default><default><default>

```

7.4 [spi3_para]

配置项	配置项含义
spi_used =xx	SPI 使用控制：1 使用，0 不用
spi_cs_bitmap =xx	启用哪路 cs，bit 位表示
spi_cs0 =xx	SPI CS0 的 GPIO 配置
spi_cs1 =xx	SPI CS1 的 GPIO 配置
spi_sclk =xx	SPI CLK 的 GPIO 配置
spi_mosi =xx	SPI MOSI 的 GPIO 配置
spi_miso =xx	SPI MISO 的 GPIO 配置

配置举例：

```
[spi3_para]
spi_used           = 0
spi_cs_bitmap     = 1
spi_cs0           = port:PA05<3><default><default><default>
spi_sclk          = port:PI06<3><default><default><default>
spi_mosi          = port:PI07<3><default><default><default>
spi_miso          = port:PI08<3><default><default><default>
spi_cs1           = port:PA09<3><default><default><default>
```

7.5 [spi_devices]

配置项	配置项含义
spi_dev_num=xx	该项目直接和下面的[spi_board0]相关，它指定主板连接 spi 设备的数目，假如有 N 个 SPI 设备 那么 [spi_devices] 中就要有 N 个（[spi_board0]到[spi_board (N-1)]）配置

7.6 [spi_board0]

配置项	配置项含义
modalias=xx	Spi 设备名字
max_speed_hz =xx	最大传输速度（HZ）
bus_num =xx	Spi 设备控制器序号
chip_select=xx	理论上可以选 0, 1, 2, 3, 目前只支持 1, 2（芯片没引出接口）
mode=xx	SPI MOSI 的 GPIO 配置可选值 0-3

8 电阻屏(rtp)

8.1 [rtp_para]

配置项	配置项含义
rtp_used=xx	该模块在方案中是否启用，
rtp_screen_size =xx	屏幕尺寸设置，以斜对角方向长度为准，以寸为单位
rtp_regidity_level=xx	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=xx	是否开启压力的们门限制，建议选 0 不开启
rtp_press_threshold= xx	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=xx	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=xx	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

9 电容屏(ctp)

9.1 [ctp_para]

配置项	配置项含义
ctp_used=xx	该选项为是否开启电容触摸，支持的话置 1，反之置 0
ctp_name	ctp 的名字
ctp_twi_id=xx	用于选择 i2c adapter, 可选 1, 2
ctp_twi_addr =xx	指明 i2c 设备地址，与具体硬件相关
ctp_screen_max_x=xx	触摸板的 x 轴最大坐标
ctp_screen_max_y=xx	触摸板的 y 轴最大坐标
ctp_revert_x_flag=xx	是否需要翻转 x 坐标，需要则置 1，反之置 0
ctp_revert_y_flag=xx	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=xx	电容屏中断信号的 GPIO 配置
ctp_wakeup=xx	电容屏唤醒信号的 GPIO 配置

配置举例：

[ctp_para]

```

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:PA03<6><default><default><default>
ctp_wakeup         = port:PA02<1><default><default><1>
  
```

注意事项：

若要支持新的电容触控 ic，在原有电容触控 ic 的代码基础上，须结合 A31 bsp 层的配置情况，作相应修改。具体说来，

1. 在 sys_config 中：ctp_twi_id 应与硬件连接一致；
2. 在驱动部分代码中：sysconfig 中的其他子键也要正确配置，在程序中，要对这些配置进行相应的处理；

10 触摸按键(touch key)

10.1[tkey_para]

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

配置举例：

```
tkey_used          = 0
tkey_twi_id        = 2
tkey_twi_addr      = 0x62
tkey_int           = port:PI13<6><default><default><default>
```

注意事项：

若支持，则将 tkey_used 置 1 并配置相应子键值；否则，tkey_used 置 0；

11 马达(motor)

11.1 [motor_para]

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

配置举例：

motor_used = 1

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。

12 闪存(nand flash)

12.1[nand_para]

配置项	配置项含义
nand_used =xx	nand 模块使能标志
nand_we =xx	nand 写时钟信号的 GPIO 配置
nand_ale =xx	nand 地址使能信号的 GPIO 配置
nand_cle =xx	nand 命令使能信号的 GPIO 配置
nand_ce1 =xx	nand 片选 1 信号的 GPIO 配置
nand_ce0 =xx	nand 片选 0 信号的 GPIO 配置
nand_nre =xx	nand 读时钟信号的 GPIO 配置
nand_rb0=xx	nand Read/Busy 1 信号的 GPIO 配置
nand_rb1 =xx	nand Read/Busy 0 信号的 GPIO 配置
nand_d0=xx	nand 数据总线信号的 GPIO 配置
nand_d1=xx	/
nand_d2=xx	/
nand_d3=xx	/
nand_d4=xx	/
nand_d5=xx	/
nand_d6=xx	/
nand_d7=xx	/
nand_wp	/
nand_ce2=xx	nand 片选 2 信号的 GPIO 配置
nand_ce3=xx	nand 片选 3 信号的 GPIO 配置
nand_ce4=xx	
nand_ce5=xx	
nand_ce6=xx	
nand_ce7=xx	
nand_spi=xx	
nand_ndqs=xx	nand ddr 时钟信号的 GPIO 配置
good_block_ratio=xx	

配置举例:

[nand_para]

nand_used = 1



```
nand_we          = port:PC00<2><default><default><default>
nand_ale         = port:PC01<2><default><default><default>
nand_cle        = port:PC02<2><default><default><default>
nand_ce1        = port:PC03<2><default><default><default>
nand_ce0        = port:PC04<2><default><default><default>
nand_nre        = port:PC05<2><default><default><default>
nand_rb0        = port:PC06<2><default><default><default>
nand_rb1        = port:PC07<2><default><default><default>
nand_d0         = port:PC08<2><default><default><default>
nand_d1         = port:PC09<2><default><default><default>
nand_d2         = port:PC10<2><default><default><default>
nand_d3         = port:PC11<2><default><default><default>
nand_d4         = port:PC12<2><default><default><default>
nand_d5         = port:PC13<2><default><default><default>
nand_d6         = port:PC14<2><default><default><default>
nand_d7         = port:PC15<2><default><default><default>
nand_wp         = port:PC16<2><default><default><default>
nand_ce2        = port:PC17<2><default><default><default>
nand_ce3        = port:PC18<2><default><default><default>
nand_ce4        =
nand_ce5        =
nand_ce6        =
nand_ce7        =
nand_spi        = port:PC23<3><default><default><default>
nand_ndqs       = port:PC24<2><default><default><default>
good_block_ratio = 0
```

13 显示初始化(**disp init**)

13.1[**disp_init**]

配置项	配置项含义
disp_init_enable=xx	是否进行显示的初始化设置
disp_mode =xx	显示模式: 0:screen0<screen0,fb0>
screen0_output_type=xx	屏 0 输出类型 (0:none; 1:lcd; 2:tv; 3:hdm; 4:vga)
screen0_output_mode =xx	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=xx	屏 1 输出类型 (0:none; 1:lcd; 2:tv; 3:hdm; 4:vga)
screen1_output_mode=xx	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=xx	fb0 的格式(4:RGB655 5:RGB565 6:RGB556 7:ARGB1555 8:RGBA5551 9:RGB888 10:ARGB8888 12:ARGB4444)
fb0_pixel_sequence=xx	fb0 的 pixel sequence(0:ARGB 1:BGRA 2:ABGR 3:RGBA)
fb0_scaler_mode_enable=xx	fb0 是否使用 scaler mode, 即使用 FE
fb0_width=xx	fb0 的宽度,为 0 时将按照输出设备的分辨率
fb0_height=xx	fb0 的高度, 为 0 时将按照输出设备的分辨率
fb1_format=xx	fb1 的格式(4:RGB655 5:RGB565 6:RGB556 7:ARGB1555 8:RGBA5551 9:RGB888 10:ARGB8888 12:ARGB4444)
fb1_pixel_sequence=xx	fb1 的 pixel sequence(0:ARGB 1:BGRA 2:ABGR 3:RGBA)
fb1_scaler_mode_enable=xx	fb1 是否使用 scaler mode, 即使用 FE
fb1_width=xx	Fb1 的宽度,为 0 时将按照输出设备的分辨率
fb1_height=xx	Fb1 的高度, 为 0 时将按照输出设备的分辨率

lcd0_backlight	Lcd0 的背光初始值, 0~255
lcd1_backlight	Lcd1 的背光初始值, 0~255
lcd0_bright	Lcd0 的亮度值, 0~100
lcd0_contrast	Lcd0 的对比度, 0~100
lcd0_saturation	Lcd0 的饱和度, 0~100
lcd0_hue	Lcd0 的色度, 0~100
lcd1_bright	Lcd1 的亮度值, 0~100
lcd1_contrast	Lcd1 的对比度, 0~100
lcd1_saturation	Lcd1 的饱和度, 0~100
lcd1_hue	Lcd1 的色度, 0~100

配置举例:

[disp_init]

```

disp_init_enable          = 1
disp_mode                 = 0
screen0_output_type      = 1
screen0_output_mode      = 4
screen1_output_type      = 1
screen1_output_mode      = 4
fb0_format                = 10
fb0_pixel_sequence       = 0
fb0_scaler_mode_enable   = 0
fb0_width                 = 0
fb0_height                = 0
fb1_format                = 10
fb1_pixel_sequence       = 0
fb1_scaler_mode_enable   = 0
fb1_width                 = 0
fb1_height                = 0
lcd0_backlight           = 197
lcd1_backlight           = 197
lcd0_bright               = 50
lcd0_contrast             = 50
lcd0_saturation           = 57
lcd0_hue                  = 50
lcd1_bright               = 50
lcd1_contrast             = 50
lcd1_saturation           = 57
lcd1_hue                  = 50

```

14 LCD 屏 0

14.1[lcd0_para]

配置项	配置项含义
lcd_used=xx	是否使用 lcd0
lcd_if=xx	lcd interface(0:hv(sync+de); 1:8080; 2:tft; 3:lvds, 4:dsi; 5:edp)
lcd_x=xx	lcd active width
lcd_y=xx	lcd active height
lcd_dclk_freq=xx	pixel clock, in MHZ unit
lcd_pwm_freq=xx	pwm freq, in HZ unit
lcd_pwm_pol=xx	pwm polarity, 0:positive; 1:negative
lcd_pwm_max_limit=xx	Lcd pwm max limit(<=255)
lcd_hbp=xx	hsync back porch
lcd_ht=xx	hsync total cycle
lcd_vbp=xx	vsync back porch
lcd_vt=xx	vsync total cycle
lcd_hv_vspw=xx	vsync plus width
lcd_hv_hspw=xx	hsync plus width
lcd_hv_if=xx	hv interface(0:parallel; 8:serial(8bit/3cycle); 10:dummyrgb(8bit/4cycle);11:rgbdummy(8bit/4cycle); 12: ccir656)
lcd_hv_srgb_seq=xx	serial RGB output sequence
lcd_hv_syuv_seq=xx	serial YUV output sequence
lcd_hv_syuv_fdly	serial YUV output F line delay(0: no delay;1: delay 2line[CCIR NTSC]; 2: delay 3line[CCIR PAL])
lcd_lvds_if=xx	0:single channel; 1:dual channel
lcd_lvds_colordepth=xx	0:8bit; 1:6bit
lcd_lvds_mode=xx	0:NS mode; 1:JEIDA mode
lcd_lvds_io_polarity=xx	0:normal; 1:pn cross
lcd_dsi_if=xx	0:video mode; 1:command mode
lcd_dsi_lane=xx	1/2/3/4lane
lcd_dsi_format=xx	0:RGB888; 1:RGB666; 2:RGB666P; 3:RGB565
lcd_dsi_eotp=xx	0:no ending symbol 1:insert ending symbol;



lcd_dsi_te=xx	0:disable te mode; 1:rising te mode; 2:falling te mode
lcd_cpu_if=xx	cpu i/f mode(0:18bit; 1:16bit mode0; 2:16bit mode1; 3:16bit mode2;4:16bit mode3; 5:9bit; 6:8bit 256K; 7:8bit 65K;)
lcd_cpu_te=xx	0:disable te mode; 1:enable rising te mode; 2:enable falling te mode
lcd_frm=xx	0:disable; 1:enable rgb666 dither; 2:enable rgb656 dither
lcd_edp_tx_ic=xx	0:anx9804; 1:anx6345
lcd_edp_tx_rate=xx	1:1.62G; 2:2.7G; 3:5.4G
lcd_edp_tx_lane=xx	1/2/4lane
lcd_io_phase=xx	0:noraml; 1:intert phase(0~3bit: vsync phase; 4~7bit:hsync phase;8~11bit:dclk phase; 12~15bit:de phase)
deu_mode=xx	Parameter for deu. 0:smoll lcd screen; 1:large lcd screen(larger than 10inch)
lcdgamma4iep=xx	Smart Backlight parameter, lcd gamma vale * 10;
smart_color=xx	90:normal lcd screen 65:retina lcd screen(9.7inch) (0~100)
lcd_bl_en=xx	LCD_BL_EN 的 GPIO 配置
lcd_power=xx	LCD_VCC control 的 GPIO 配置
lcd_pwm=xx	lcd PWM 的 GPIO 配置 (PWM0 固定使用 PB02, PWM1 固定使用 PI03,用户无需修改该项)
lcd_gpio_scl	iic SCL
lcd_gpio_sda	iic SDA
lcd_gpio_0/1/2/3=xx	LCD 额外需要使用的 GPIO 配置
lcd0~23=xx	lcd 数据的 GPIO 配置
lcdclk=xx	lcd 信号的 GPIO 配置(具体信号与实际电路相关)
lcdde=xx	lcd 信号的 GPIO 配置(具体信号与实际电路相关)
lcdhsync=xx	lcd 信号的 GPIO 配置(具体信号与实际电路相关)
lcdvsync=xx	lcd 信号的 GPIO 配置(具体信号与实际电路相关)

配置举例:



```
[lcd0_para]
lcd_used          = 1
lcd_if           = 0
lcd_x            = 1280
lcd_y            = 800
lcd_dclk_freq    = 70
lcd_pwm_freq     = 50000
lcd_pwm_pol      = 0
lcd_pwm_max_limit = 150
lcd_hbp          = 20
lcd_ht           = 1418
lcd_hspw         = 10
lcd_vbp          = 10
lcd_vt           = 814
lcd_vspw         = 5
lcd_hv_if        = 0
lcd_hv_srgb_seq  = 0
lcd_hv_syuv_seq  = 0
lcd_hv_syuv_fdly = 0
lcd_lvds_if      = 0
lcd_lvds_colordepth = 1
lcd_lvds_mode    = 0
lcd_lvds_io_polarity = 0
lcd_dsi_if       = 0
lcd_dsi_lane     = 0
lcd_dsi_format   = 0
lcd_dsi_eotp     = 0
lcd_dsi_te       = 0
lcd_cpu_if       = 0
lcd_cpu_te       = 0
lcd_frm          = 1
lcd_edp_tx_ic    = 0
lcd_edp_tx_rate  = 0
lcd_edp_tx_lane  = 0
lcd_io_phase     = 0x00
deu_mode         = 0
lcdgamma4iep    = 22
Smart_color      = 90
lcd_bl_en        = port:PA25<1><0><default><1>
lcd_power        = port:power2<1><0><default><1>
```



```
lcd_pwm          = port:PH13<2><0><default><default>
lcd_gpio_scl     =
lcd_gpio_sda     =
lcd_gpio_0       =
lcd_gpio_1       =
lcd_gpio_2       =
lcd_gpio_3       =
lcd0             = port:PD00<2><0><default><default>
lcd1             = port:PD01<2><0><default><default>
lcd2             = port:PD02<2><0><default><default>
lcd3             = port:PD03<2><0><default><default>
lcd4             = port:PD04<2><0><default><default>
lcd5             = port:PD05<2><0><default><default>
lcd6             = port:PD06<2><0><default><default>
lcd7             = port:PD07<2><0><default><default>
lcd8             = port:PD08<2><0><default><default>
lcd9             = port:PD09<2><0><default><default>
lcd10            = port:PD10<2><0><default><default>
lcd11            = port:PD11<2><0><default><default>
lcd12            = port:PD12<2><0><default><default>
lcd13            = port:PD13<2><0><default><default>
lcd14            = port:PD14<2><0><default><default>
lcd15            = port:PD15<2><0><default><default>
lcd16            = port:PD16<2><0><default><default>
lcd17            = port:PD17<2><0><default><default>
lcd18            = port:PD18<2><0><default><default>
lcd19            = port:PD19<2><0><default><default>
lcd20            = port:PD20<2><0><default><default>
lcd21            = port:PD21<2><0><default><default>
lcd22            = port:PD22<2><0><default><default>
lcd23            = port:PD23<2><0><default><default>
lcdclk           = port:PD24<2><0><default><default>
lcdde            = port:PD25<2><0><default><default>
lcdhsync         = port:PD26<2><0><default><default>
lcdvsync         = port:PD27<2><0><default><default>
```

15 LCD 屏 1

15.1[lcd1_para]

所有配置跟 lcd0 一样

16 sata

配置项	配置项含义
sata_used	
sata_power_en	

配置举例:

[sata_para]

sata_used = 1

sata_power_en =

17 TV

17.1[`tv_out_dac_para`]

配置项	配置项含义
<code>dac_used</code>	
<code>dac0_src</code>	
<code>dac1_src</code>	
<code>dac2_src</code>	
<code>dac3_src</code>	

配置举例：

[`tv_out_dac_para`]

```

dac_used           = 1
dac0_src          = 4
dac1_src          = 5
dac2_src          = 6
dac3_src          = 0
    
```

17.2[`tvout_para`]

配置项	配置项含义
<code>tvout_used=xx</code>	
<code>tvout_channel_num=xx</code>	

配置举例：

[`tvout_para`]

```

tvout_used         = 1
tvout_channel_num = 1
    
```

17.3[`tvin_para`]

配置项	配置项含义
<code>tvin_used</code>	

tvn_channel_num	
-----------------	--

配置举例:

[tvn_para]

tvn_used = 0

tvn_channel_num = 4

18 HDMI

18.1 [hdmi_para]

配置项	配置项含义
para_used =xx	是否使用 hdmi

19 摄像头(CSI)

19.1 [camera_list_para]

配置项	配置项含义

配置举例：

[camera_list_para]

camera_list_para_used = 1

ov7670 = 0
gc0308 = 1
gt2005 = 0
hi704 = 0
sp0838 = 0
mt9m112 = 0
mt9m113 = 0
ov2655 = 0
hi253 = 0
gc0307 = 0
mt9d112 = 0
ov5640 = 1
gc2015 = 0
ov2643 = 0
gc0329 = 0
gc0309 = 0
tvp5150 = 0
s5k4ec = 0
ov5650_mv9335 = 0
siv121d = 0

19.2[csi0_para]

留空，不要填写，如下：

[csi0_para]

csi_used = 0

19.3[csi1_para]

特别注意事项：

在 A31 以及后续项目中（因为内核对 GPIO 资源的管理有修改），如果两个 sensor 制作 2 合 1 模组的时候**请注意将两个模组的 reset 控制脚分开（包括），stby 控制脚也分开**，仅有电源，数据线、clock 线、地可以复用。如果是使用 RAW 格式的 sensor，硬件上需要 CSI_D[11:2]共 10 条数据线，请不要将 CSI_D3 和 CSI_D2 用做 GPIO 功能，模组上的 D[3:2]也要注意从 sensor 端引出来。

配置项	配置项含义
csi_used =xx	是否使用 csi1
csi_twi_id =xx	csi 使用的 IIC 通道序号，查看具体方案原理图，使用 twi0 填 0
csi_mname=xx	csi 使用的模组名称，需要与驱动匹配，可以查看驱动目录里面的 readme 目前有 gc0307, gc0308, gc2035, gt2005, hi253, ov5640, s5k4ec 可选
csi_twi_addr=xx	csi 使用的模组的 IIC 地址（8bit 地址），可以查看驱动目录里面的 readme
csi_if	配置目前使用模组的接口时序： 0:8bit 数据线，带 Hsync,Vsync 1:16bit 数据线，带 Hsync,Vsync 2:24bit 数据线，带 Hsync,Vsync 3:8bit 数据线,BT656 内嵌同步,单通道



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



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



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

配置举例：

[csi1_para]

```
csi_used           = 1
csi_mode           = 0
csi_dev_qty       = 2
csi_stby_mode     = 0

csi_mname         = "ov5640"
csi_twi_id        = 0
csi_twi_addr      = 0x78
csi_if            = 0
csi_vflip         = 0
csi_hflip         = 1
csi_iovdd         = "axp22_eldo3"
csi_avdd          = "axp22_dldo4"
csi_dvdd          = "axp22_eldo2"
csi_vol_iovdd     = 2800
csi_vol_avdd      = 2800
csi_vol_dvdd      = 1800
csi_flash_pol     = 1

csi_mname_b       = "gc0307"
csi_twi_id_b      = 0
csi_twi_addr_b    = 0x42
csi_if_b          = 0
csi_vflip_b       = 1
csi_hflip_b       = 1
csi_iovdd_b       = "axp22_eldo3"
csi_avdd_b        = "axp22_dldo4"
csi_dvdd_b        = "axp22_eldo2"
csi_vol_iovdd_b   = 2800
```



```
csi_vol_avdd_b      = 2800
csi_vol_dvdd_b      = 1800
csi_flash_pol_b     = 1

csi_pck              = port:PE00<2><default><default><default>
csi_mck              = port:PE01<2><default><default><default>
csi_hsync            = port:PE02<2><default><default><default>
csi_vsync            = port:PE03<2><default><default><default>
csi_d0               =
csi_d1               =
csi_d2               =
csi_d3               =
csi_d4               = port:PE08<2><default><default><default>
csi_d5               = port:PE09<2><default><default><default>
csi_d6               = port:PE10<2><default><default><default>
csi_d7               = port:PE11<2><default><default><default>
csi_d8               = port:PE12<2><default><default><default>
csi_d9               = port:PE13<2><default><default><default>
csi_d10              = port:PE14<2><default><default><default>
csi_d11              = port:PE15<2><default><default><default>
csi_reset            = port:PE04<1><default><default><0>
csi_power_en         =
csi_stby              = port:PE05<1><default><default><1>
csi_flash            =
csi_af_en            =
csi_reset_b          = port:PE06<1><default><default><0>
csi_power_en_b       =
csi_stby_b           = port:PE07<1><default><default><1>
csi_flash_b          =
csi_af_en_b          =
```

20 SD/MMC

20.1 [mmc0_para]

配置项	配置项含义
sdc_used=xx	SDC 使用控制: 1 使用, 0 不用
sdc_detmode=xx	检测模式: 1-gpio 检测, 2-data3 检测, 3-无检测, 卡常在(不卡拔插), 4 - manual mode(from proc file system node)
bus_width=xx	位宽: 1-1bit, 4-4bit
sdc_d1=xx	SDC DATA1 的 GPIO 配置
sdc_d0=xx	SDC DATA0 的 GPIO 配置
sdc_clk=xx	SDC CLK 的 GPIO 配置
sdc_cmd=xx	SDC CMD 的 GPIO 配置
sdc_d3=xx	SDC DATA3 的 GPIO 配置
sdc_d2=xx	SDC DATA2 的 GPIO 配置
sdc_det=xx	SDC DET 的 GPIO 配置
sdc_use_wp=xx	SDC 写保护配置: 1 使用, 0 不用
sdc_wp=xx	SDC WP 的 GPIO 配置
sdc_isio=xx	是否是 sdio card,0:不是, 1: 是
sdc_regulator=xx	假如过卡支持 SD3.0 或者 emmc4.5 的 UHS-I/DDR、HS200, 这里就要写成 sdc_regulator = "axp22_eldo2"

配置举例:

[mmc0_para]

```

sdc_used          = 1
sdc_detmode       = 1
bus_width         = 4
sdc_d1            = port:PF0<2><1><default><default>
sdc_d0            = port:PF1<2><1><default><default>
sdc_clk           = port:PF2<2><1><default><default>
sdc_cmd           = port:PF3<2><1><default><default>
sdc_d3            = port:PF4<2><1><default><default>
sdc_d2            = port:PF5<2><1><default><default>
sdc_det           = port:PH1<0><1><default><default>
sdc_use_wp        = 0
  
```

sdc_wp =

20.2[mmc1_para]

配置项	配置项含义
sdc_used=xx	SDC 使用控制: 1 使用, 0 不用
sdc_detmode=xx	检测模式: 1-gpio 检测, 2-data3 检测, 3-无检测, 卡常在(不卡拔插), 4 - manual mode(from proc file system node)
bus_width=xx	位宽: 1-1bit, 4-4bit
sdc_d1=xx	SDC DATA1 GPIO 配置
sdc_d0=xx	SDC DATA0 GPIO 配置
sdc_clk=xx	SDC CLK GPIO 配置
sdc_cmd=xx	SDC CMD GPIO 配置
sdc_d3=xx	SDC DATA3 GPIO 配置
sdc_d2=xx	SDC DATA2 GPIO 配置
sdc_det=xx	SDC DET GPIO 配置
sdc_use_wp=xx	SDC 写保护配置: 1 使用, 0 不用
sdc_wp=xx	SDC WP GPIO 配置

配置举例:

[mmc1_para]

```

sdc_used          = 1
sdc_detmode      = 1
bus_width        = 4
sdc_cmd          = port:PH22<5><1><default><default>
sdc_clk          = port:PH23<5><1><default><default>
sdc_d0           = port:PH24<5><1><default><default>
sdc_d1           = port:PH25<5><1><default><default>
sdc_d2           = port:PH26<5><1><default><default>
sdc_d3           = port:PH27<5><1><default><default>
sdc_det          = port:PH2<0><1><default><default>
sdc_use_wp       = 0
sdc_wp           =
    
```

20.3[mmc2_para]

配置项	配置项含义
sdc_used=xx	SDC 使用控制: 1 使用, 0 不用

sdc_detmode=xx	检测模式：1-gpio 检测，2-data3 检测，3-无检测，卡常在(不卡拔插)，4 - manual mode(from proc file system node)
bus_width=xx	位宽：1-1bit, 4-4bit
sdc_d1=xx	SDC DATA1 GPIO 配置
sdc_d0=xx	SDC DATA0 GPIO 配置
sdc_clk=xx	SDC CLK GPIO 配置
sdc_cmd=xx	SDC CMD GPIO 配置
sdc_d3=xx	SDC DATA3 GPIO 配置
sdc_d2=xx	SDC DATA2 GPIO 配置
sdc_det=xx	SDC DET GPIO 配置
sdc_use_wp=xx	SDC 写保护配置：1 使用，0 不用
sdc_wp=xx	SDC WP GPIO 配置

配置举例：

[mmc2_para]

```

sdc_used                = 1
sdc_detmode            = 1
bus_width               = 4
sdc_cmd                = port:PH22<5><1><default><default>
sdc_clk                = port:PH23<5><1><default><default>
sdc_d0                 = port:PH24<5><1><default><default>
sdc_d1                 = port:PH25<5><1><default><default>
sdc_d2                 = port:PH26<5><1><default><default>
sdc_d3                 = port:PH27<5><1><default><default>
sdc_det                = port:PH2<0><1><default><default>
sdc_use_wp             = 0
sdc_wp                 =
    
```

20.4[mmc3_para]

配置项	配置项含义
sdc_used=xx	SDC 使用控制：1 使用，0 不用
sdc_detmode=xx	检测模式：1-gpio 检测，2-data3 检测，3-无检测，卡常在(不卡拔插)，4 - manual mode(from proc file system node)
bus_width=xx	位宽：1-1bit, 4-4bit
sdc_d1=xx	SDC DATA1 GPIO 配置
sdc_d0=xx	SDC DATA0 GPIO 配置

<code>sdc_clk=xx</code>	SDC CLK GPIO 配置
<code>sdc_cmd=xx</code>	SDC CMD GPIO 配置
<code>sdc_d3=xx</code>	SDC DATA3 GPIO 配置
<code>sdc_d2=xx</code>	SDC DATA2 GPIO 配置
<code>sdc_det=xx</code>	SDC DET GPIO 配置
<code>sdc_use_wp=xx</code>	SDC 写保护配置：1 使用，0 不用
<code>sdc_wp=xx</code>	SDC WP GPIO 配置

配置举例：

[mmc3_para]

```

sdc_used                = 1
sdc_detmode            = 1
bus_width               = 4
sdc_cmd                 = port:PH22<5><1><default><default>
sdc_clk                 = port:PH23<5><1><default><default>
sdc_d0                  = port:PH24<5><1><default><default>
sdc_d1                  = port:PH25<5><1><default><default>
sdc_d2                  = port:PH26<5><1><default><default>
sdc_d3                  = port:PH27<5><1><default><default>
sdc_det                 = port:PH2<0><1><default><default>
sdc_use_wp              = 0
sdc_wp                  =

```

21 SIM 卡

21.1 [smc_para]

配置项	配置项含义
smc_used =xx	
smc_rst=xx	
smc_vppen=xx	
smc_vppp=xx	
smc_det=xx	
smc_vccen=xx	
smc_sck=xx	
smc_sda=xx	

配置举例：

[smc_para]

```

smc_used          = 0
smc_rst           = port:PH13<5><default><default><default>
smc_vppen        = port:PH14<5><default><default><default>
smc_vppp         = port:PH15<5><default><default><default>
smc_det          = port:PH16<5><default><default><default>
smc_vccen        = port:PH17<5><default><default><default>
smc_sck          = port:PH18<5><default><default><default>
smc_sda          = port:PH19<5><default><default><default>
    
```

22 USB 控制标志

22.1 [usbc0]

配置项	配置项含义
usb_used =xx	USB 使能标志(xx=1 or 0)。置 1，表示系统中 USB 模块可用，置 0，则表示系统 USB 禁用。此标志只对具体的 USB 控制器模块有效。
usb_port_type =xx	USB 端口的使用情况。(xx=0/1/2) 0: device only 1: host only 2: OTG
usb_detect_type=xx	USB 端口的检查方式。 0: 无检查方式 1: vbus/id 检查
usb_id_gpio=xx	USB ID pin 脚配置。具体请参考 gpio 配置说明。《配置与 GPIO 管理.doc》
usb_det_vbus_gpio=xx	USB DET_VBUS pin 脚配置。如果 GPIO 提供 pin，请参考 gpio 配置说明《配置与 GPIO 管理.doc》。如果的 AXP 提供 pin,则配置为: "axp_ctrl"。
usb_drv_vbus_gpio=xx	USB DRY_VBUS pin 脚配置。具体请参考 gpio 配置说明。《配置与 GPIO 管理.doc》
usb_restrict_gpio=xx	USB 限流控制 pin 脚 USB RESTRICT_GPIO pin 脚配置。具体请参考 gpio 配置说明。《配置与 GPIO 管理.doc》
usb_host_init_state=xx	host only 模式下，Host 端口初始化状态。 0: 初始化后 USB 不工作 1: 初始化后 USB 工作
usb_restric_flag=xx	Usb 限流标志位 0: 不使能限流功能 1: 使能限流功能
usb_restric_voltage=xx	限流开启的条件 电压值小于设置值，则开启限流
usb_restric_capacity=xx	限流开启的条件 电量值小于设置值，则开启限流

配置举例:

[usbc0]

usb_used = 1

usb_port_type = 2


```

usb_detect_type          = 1
usb_id_gpio              = port:PH4<0><1><default><default>
usb_det_vbus_gpio        = port:PH5<0><0><default><default>
usb_drv_vbus_gpio        = port:PB9<1><0><default><0>
usb_restrict_gpio        = port:PH26<1><0><default><0>
usb_host_init_state      = 0
usb_restric_flag         = 0
usb_restric_voltage      = 3550000
usb_restric_capacity     = 5
  
```

22.2[usbc1]

配置项	配置项含义
usb_used=xx	USB 使能标志(xx=1 or 0)。置 1，表示系统中 USB 模块可用，置 0，则表示系统 USB 禁用。此标志只对具体的 USB 控制器模块有效。
usb_port_type=xx	USB 端口的使用情况。(xx=0/1/2) 0: device only 1: host only 2: OTG
usb_detect_type=xx	USB 端口的检查方式。 0: 无检查方式 1: vbus/id 检查
usb_id_gpio=xx	USB ID pin 脚配置。具体请参考 gpio 配置说明。《配置与 GPIO 管理.doc》
usb_det_vbus_gpio=xx	USB DET_VBUS pin 脚配置。具体请参考 gpio 配置说明。《配置与 GPIO 管理.doc》
usb_restrict_gpio=xx	USB 限流控制 pin 脚 USB RESTRICT_GPIO pin 脚配置。具体请参考 gpio 配置说明。《配置与 GPIO 管理.doc》
usb_drv_vbus_gpio=xx	USB DRY_VBUS pin 脚配置。具体请参考 gpio 配置说明。《配置与 GPIO 管理.doc》
usb_host_init_state=xx	host only 模式下，Host 端口初始化状态。 0: 初始化后 USB 不工作 1: 初始化后 USB 工作
usb_restric_flag=xx	Usb 限流标志位 0: 表不设限流，1 开启限流

配置举例：

```

[usbc1]
usb_used          = 1
usb_port_type     = 1
  
```

```
usb_detect_type      = 0
usb_id_gpio          =
usb_det_vbus_gpio    =
usb_drv_vbus_gpio    = port:PH6<1><0><default><0>
usb_restrict_gpio    = port:PH26<1><0><default><0>
usb_host_init_state  = 1
usb_restric_flag     = 0
```

22.3[usbc2]

配置项	配置项含义
usb_used =xx	USB 使能标志(xx=1 or 0)。置 1，表示系统中 USB 模块可用，置 0，则表示系统 USB 禁用。此标志只对具体的 USB 控制器模块有效。
usb_port_type =xx	USB 端口的使用情况。(xx=0/1/2) 0: device only 1: host only 2: OTG
usb_detect_type=xx	USB 端口的检查方式。 0: 无检查方式 1: vbus/id 检查
usb_id_gpio=xx	USB 限流控制 pin 脚 USB ID pin 脚配置。具体请参考 gpio 配置说明。《配置与 GPIO 管理.doc》
usb_det_vbus_gpio=xx	USB DET_VBUS pin 脚配置。具体请参考 gpio 配置说明。《配置与 GPIO 管理.doc》
usb_drv_vbus_gpio=xx	USB DRY_VBUS pin 脚配置。具体请参考 gpio 配置说明。《配置与 GPIO 管理.doc》
usb_restrict_gpio=xx	USB RESTRICT_GPIO pin 脚配置。具体请参考 gpio 配置说明。《配置与 GPIO 管理.doc》
usb_host_init_state=xx	host only 模式下，Host 端口初始化状态。 0: 初始化后 USB 不工作 1: 初始化后 USB 工作
usb_restric_flag=xx	Usb 限流标志位 0: 表不设限流，1 开启限流

配置举例：

```
[usbc2]
usb_used      = 1
usb_port_type = 1
usb_detect_type = 0
usb_id_gpio   =
```

```

usb_det_vbus_gpio      =
usb_drv_vbus_gpio     = port:PH3<1><0><default><0>
usb_restrict_gpio     = port:PH26<1><0><default><0>
usb_host_init_state   = 1
usb_restric_flag      = 0
  
```

23 USB Device

23.1[usb_feature]

配置项	配置项含义
vendor_id=xx	USB 厂商 ID
mass_storage_id =xx	U 盘 ID
adb_id =xx	USB 调试桥 ID
manufacturer_name=xx	USB 厂商名
product_name=xx	USB 产品名
serial_number=xx	USB 序列号

配置举例：

```

[usb_feature]
vendor_id      = 0x18D1
mass_storage_id = 0x0001
adb_id        = 0x0002
manufacturer_name = "USB Developer"
product_name   = "Android"
serial_number  = "20080411"
  
```

23.2[msc_feature]

配置项	配置项含义
vendor_name=xx	U 盘 厂商名
product_name=xx	U 盘产品名
release=xx	发布版本
luns=xx	U 盘逻辑单元的个数(PC 可以看到的 U 盘盘符的个数)

配置举例：



```
[msc_feature]
vendor_name      = "USB 2.0"
product_name     = "USB Flash Driver"
release         = 100
luns             = 2
```

Confidential

24 重力感应(G-Sensor)

24.1 [gsensor_para]

配置项	配置项含义
gsensor_used=xx	是否支持 gsensor
gsensor_twi_id=xx	I2C 的 BUS 控制选择，0 : TWI0;1:TWI1;2:TWI2
gsensor_twi_addr=xx	芯片的 I2C 地址
gsensor_int1=xx	中断 1 的 GPIO 配置
gsensor_int2=xx	中断 2 的 GPIO 配置

配置举例：

```
[gsensor_para]
gsensor_used          = 1
gsensor_twi_id        = 2
gsensor_twi_addr      = 0x18
gsensor_int1          = port:PA09<6><1><<default><<default>
gsensor_int2          =
```

25 WIFI

25.1[wifi_para]

配置项	配置项含义
wifi_used =xx	是否要使用 wifi
wifi_sdc_id =xx	sdio wifi 选用的是哪个 sdc 作为接口
wifi_usbc_id =xx	usb wifi 选用的是哪个 usb 作为接口
wifi_usbc_type=xx	usb 接口类型，1 为 ehci，0 为 ohci
wifi_mod_sel =xx	具体选择哪一款模组 1-bcm40181; 2-bcm40183; 3-rtl8723as; 4-rtl8189es; 5 - rtl8192cu; 6 - rtl8188eu; 7 - rtl8723au;
wifi_power=xx	给模组供电的 axp 引脚名

说明：[wifi_para]下的配置项是 usb 和 sdio 接口 wifi 共用的。

25.2sdio 接口 wifi rtl8723as demo

```
[wifi_para]
wifi_used          = 1
wifi_sdc_id        = 1
wifi_usbc_id       = 1
wifi_usbc_type     = 1
wifi_mod_sel       = 3
wifi_power         = "axp22_ald01"

rtk_rtl8723as_wl_dis = port:PG10<1><<default><default><0>
rtk_rtl8723as_bt_dis = port:PG11<1><<default><default><0>
rtk_rtl8723as_wl_host_wake = port:PG12<0><<default><default><0>
rtk_rtl8723as_bt_host_wake = port:PG17<0><<default><default><0>
```

以上配置意思是要使用序号为 3 的 SDIO 接口 rtl8723as 模组，选用 SDC1 接口。
SDC1 对应是 mmc1，需要确定[mmc1_para]配置项如下：

```
[mmc1_para]
sdc_used          = 1
sdc_detmode      = 4
sdc_buswidth     = 4
sdc_clk          = port:PG00<2><1><2><default>
sdc_cmd          = port:PG01<2><1><2><default>
sdc_d0           = port:PG02<2><1><2><default>
sdc_d1           = port:PG03<2><1><2><default>
sdc_d2           = port:PG04<2><1><2><default>
sdc_d3           = port:PG05<2><1><2><default>
sdc_det          =
sdc_use_wp       = 0
sdc_wp           =
sdc_isio         = 1
sdc_regulator    = "none"
```

25.3usb 接口 wifi rtl8188eu demo

```
[wifi_para]
wifi_used        = 1
wifi_sdc_id      = 1
wifi_usbc_id     = 1
wifi_usbc_type   = 1
wifi_mod_sel     = 6
wifi_power       = "axp22_aldo1"
```

以上配置意思是要使用序号为 6 的 ehci USB 接口 rtl8188eu 模组，选用 usb1 接口。需要确定 [usbc1]配置项如下：

```
[usbc1]
usb_used        = 1
usb_port_type   = 1
usb_detect_type = 0
usb_id_gpio     =
usb_det_vbus_gpio =
usb_drv_vbus_gpio =
usb_restrict_gpio =
usb_host_init_state = 0
usb_restric_flag = 0
```

26 3G

26.1[3g_para]

配置项	配置项含义
3g_used	3G 使能标志位。 0: 禁用; 1: 使能
3g_usbc_num	3G 使用到的 USB 控制器编号。 0: USB0; 1: USB1; 2: USB2; 3: USB3 等
3g_uart_num	3G 使用到的 UART 控制器编号。 0: UART0; 1: UART1; 2: UART2; 3: UART3 等
3g_pwr	
3g_wakeup	
3g_int	

配置举例:

[3g_para]

3g_used = 1

3g_usbc_num = 2

3g_uart_num = 0

3g_pwr =

3g_wakeup =

3g_int =

27 gyroscope

27.1 [gy_para]

配置项	配置项含义
gy_used=xx	是否支持 gyr
gy_twi_id=xx	I2C 的 BUS 控制选择，0 : TWI0;1:TWI1;2:TWI2
gy_twi_addr=xx	芯片的 I2C 地址
gy_int1=xx	中断 1 的 GPIO 配置
gy_int2=xx	中断 2 的 GPIO 配置

配置举例:

[gy_para]

```

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

28 光感(light sensor)

28.1 [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

配置举例:

[ls_para]

ls_used = 1

ls_twi_id = 2

ls_twi_addr = 0x23

ls_int = port:PA12<6><1><default><default>

29 罗盘(Compass)

29.1 [compass_para]

配置项	配置项含义
compass_used=xx	是否支持 compass
compass_twi_id=xx	I2C 的 BUS 控制选择，0 : TWI0;1:TWI1;2:TWI2
compass_twi_addr =xx	芯片的 I2C 地址
compass_int =xx	中断的 GPIO 配置

配置举例：

[compass_para]

```
compass_used           = 1
compass_twi_id        = 2
compass_twi_addr      = 0x0d
compass_int           = port:PA11<6><1><<default><<default>
```

30 蓝牙(blutetooth)

30.1 [bt_para]

配置项	配置项含义
bt_used=xx	BLUETOOTH 使用控制：1 使用，0 不用
bt_uart_id=xx	BLUETOOTH 使用的 UART 控制器号
bt_wakeup =xx	BT WAKEUP GPIO 配置
bt_gpio=xx	BT 可选 GPIO 配置
bt_rst=xx	BT RESET GPIO 配置

配置举例：

[bt_para]

bt_used = 0

bt_uart_id = 2

bt_wakeup = port:PI20<1><default><default><default>

bt_gpio = port:PI21<1><default><default><default>

bt_rst = port:PB05<1><default><default><default>

31 数字音频总线(I2S)

31.1 [i2s_para]

配置项	配置项含义
i2s_used=xx	xx 为 0 时加载该模块，为 0 是不加载
i2s_channel=xx	声道控制
i2s_mclk =xx	I2sMCLK 信号的 GPIO 配置
i2s_bclk=xx	I2sBCLK 信号的 GPIO 配置
i2s_lrclk =xx	I2sLRCK 信号的 GPIO 配置
i2s_dout0	I2S out0 的 GPIO 配置
i2s_dout1	暂不使用
i2s_dout2	暂不使用
i2s_dout3	暂不使用
i2s_din	I2sIN 信号的 GPIO 配置

配置举例:

```

i2s_used           = 0
i2s_channel        = 2
i2s_mclk           = port:PB5<2><1><<default><default>
i2s_bclk           = port:PB6<2><1><<default><default>
i2s_lrclk          = port:PB7<2><1><<default><default>
i2s_dout0          = port:PB8<2><1><<default><default>
i2s_dout1          =
i2s_dout2          =
i2s_dout3          =
i2s_din            = port:PB12<2><1><<default><default>
  
```

32 数字音频总线(S/PDIF)

32.1 [spdif_para]

配置项	配置项含义
spdif_used=xx	xx 为 0 时加载该模块，为 1 是不加载
spdif_dout =xx	Spdif out 的 gpio 控制
spdif_din=xx	

配置举例：

[spdif_para]

spdif_used = 1

spdif_mclk =

spdif_dout = port:PH28<3><1><default><default>

spdif_din =

33 内置音频(codec)

33.1 [audio_para]

配置项	配置项含义
audio_used =xx	Audiocodec 是否使用， 1：打开（默认）0：关闭
audio_pa_ctrl=xx	喇叭的 gpio 口控制。

配置举例：

[audio_para]

audio_used = 1

audio_pa_ctrl = port:PA18<1><<default><default><0>

34 红外(ir)

34.1 [ir_para]

配置项	配置项含义
ir_used=xx	是否支持 ir
ir0_rx =xx	ir 的接收管脚 GPIO 配置

配置举例:

[ir_para]

ir_used = 1

ir_rx = port:PL04<2><1><default><default>

35 PMU 电源

35.1 [pmu_para]

配置项	配置项含义
pmu_used=xx	Pmu 使能标志(xx=1 or 0), 0: 不使用, 1: 使用
pmu_twi_addr=xx	Pmu 设备地址
pmu_twi_id=xx	Pmu 挂载的 i2c 控制器号, 0: twi0, 1: twi1, 2: twi2
pmu_irq_id=xx	Pmu 中断号, 0: NMI, 1: 1 号中断 2: 2 号中断……
pmu_battery_rdc=xx	电池内阻, mΩ, 根据实际测试填写
pmu_battery_cap=xx	电池容量, mAh, 根据实际测试填写
pmu_batdeten	PMU 电池检测功能使能, 0: 不自动检测 1: 自动检测
pmu_runtime_chgcur=xx	设置开机充电电流, mA, 300/450/600/750/900/1050/1200 /1350/1500/1650/1800/1950
pmu_earlysuspend_chgcur=xx	设置关屏充电电流, mA, 300/450/600/750/900/1050/1200 /1350/1500/1650/1800/1950
pmu_suspend_chgcur=xx	设置休眠充电电流, mA, 300/450/600/750/900/1050/1200 /1350/1500/1650/1800/1950
pmu_shutdown_chgcur=xx	设置关机充电电流, mA 300/450/600/750/900/1050/1200 /1350/1500/1650/1800/1950
pmu_init_chgvol=xx	设置充电目标电压, mV, 4100/4220/4200/4240
pmu_init_chgend_rate=xx	设置结束充电电流的比率, %, 10, 15
pmu_init_chg_enabled=xx	设置充电功能, 0: 关闭, 1: 打开
pmu_init_adc_freq=xx	设置 adc 采样率, Hz, 100/200/400/800
pmu_init_adc_freqts=xx	设置 TS 引脚采样率, Hz, 100/200/400/800
pmu_init_chg_pretime=xx	设置预充电超时时间, min, 40/50/60/70
pmu_init_chg_cstime=xx	设置恒流充电超时时间, min, 360/480/600/720
pmu_batt_cap_correct	完成一次完成充放电后, 电池容量是否自校正功

	能启用，0：关闭 1：开启
Pmu_bat_regu_en = x	充电完成后，bat regulator 是否常开，0：关闭，1：常开
pmu_bat_para1=xx	设置空载电池电压为 3.13V 对应的百分比，%
pmu_bat_para2=xx	设置空载电池电压为 3.27V 对应的百分比，%
pmu_bat_para3=xx	设置空载电池电压为 3.34V 对应的百分比，%
pmu_bat_para4=xx	设置空载电池电压为 3.41V 对应的百分比，%
pmu_bat_para5=xx	设置空载电池电压为 3.48V 对应的百分比，%
pmu_bat_para6=xx	设置空载电池电压为 3.52V 对应的百分比，%
pmu_bat_para7=xx	设置空载电池电压为 3.55V 对应的百分比，%
pmu_bat_para8=xx	设置空载电池电压为 3.57V 对应的百分比，%
pmu_bat_para9=xx	设置空载电池电压为 3.59V 对应的百分比，%
pmu_bat_para10=xx	设置空载电池电压为 3.61V 对应的百分比，%
pmu_bat_para11=xx	设置空载电池电压为 3.63V 对应的百分比，%
pmu_bat_para12=xx	设置空载电池电压为 3.64V 对应的百分比，%
pmu_bat_para13=xx	设置空载电池电压为 3.66V 对应的百分比，%
pmu_bat_para14=xx	设置空载电池电压为 3.7V 对应的百分比，%
pmu_bat_para15=xx	设置空载电池电压为 3.73V 对应的百分比，%
pmu_bat_para16=xx	设置空载电池电压为 3.77V 对应的百分比，%
pmu_bat_para17=xx	设置空载电池电压为 3.78V 对应的百分比，%
pmu_bat_para18=xx	设置空载电池电压为 3.8V 对应的百分比，%
pmu_bat_para19=xx	设置空载电池电压为 3.82V 对应的百分比，%
pmu_bat_para20=xx	设置空载电池电压为 3.84V 对应的百分比，%
pmu_bat_para21=xx	设置空载电池电压为 3.85V 对应的百分比，%
pmu_bat_para22=xx	设置空载电池电压为 3.87V 对应的百分比，%
pmu_bat_para23=xx	设置空载电池电压为 3.91V 对应的百分比，%
pmu_bat_para24=xx	设置空载电池电压为 3.94V 对应的百分比，%
pmu_bat_para25=xx	设置空载电池电压为 3.98V 对应的百分比，%
pmu_bat_para26=xx	设置空载电池电压为 4.01V 对应的百分比，%
pmu_bat_para27=xx	设置空载电池电压为 4.05V 对应的百分比，%
pmu_bat_para28=xx	设置空载电池电压为 4.08V 对应的百分比，%
pmu_bat_para29=xx	设置空载电池电压为 4.1V 对应的百分比，%
pmu_bat_para30=xx	设置空载电池电压为 4.12V 对应的百分比，%
pmu_bat_para31=xx	设置空载电池电压为 4.14V 对应的百分比，%
pmu_bat_para32=xx	设置空载电池电压为 4.15V 对应的百分比，%
pmu_usbvol_limit=xx	设置 usb 限压功能，0：关闭，1：打开
pmu_usbvol=xx	设置 usb 限压电压，mV，4000/4100/4200/4300/4400/4500/4600/4700
pmu_usbvol_pc =xx	设置连接至 PC 时 USB 限压值，mV，



	4000/4100/4200/4300/4400/4500/4600/4700
pmu_usbcur_limit=xx	设置 usb 限流功能，0：关闭，1：打开
pmu_usbcur=xx	设置 usb 限流电流，mA，500/900,若设置为 0，则不限流
pmu_usbcur_pc =xx	设置连接至 PC 时 USB 限流值，mA。500/900,若设置为 0，则不限流
pmu_pwroff_vol=xx	设置启动过程中硬件保护电压，mV，2600/2700/2800/2900/3000/3100/3200/3300
pmu_pwron_vol=xx	设置开机状态下的硬件保护电压，mV，2600/2700/2800/2900/3000/3100/3200/3300
pmu_pekoff_time=xx	设置硬件关机时长，ms，4000/6000/8000/10000
pmu_pekoff_func=xx	设置长按键强制关机后是否自动启动功能，0：不自动启动 1：自动启动
pmu_pekoff_en=xx	设置长按键硬件关机功能，0：关闭，1：打开
pmu_peklong_time=xx	设置长按键中断时间，ms，1000/1500/2000/2500
pmu_pekcon_time=xx	设置开机时间，ms，128/1000/2000/3000
pmu_pwrok_time=xx	设置电源启动完成后 pwrok 信号延时，ms，8/16/32/64
pmu_battery_warning_level1 =xx	设置电池低电第一级报警门限：可设范围 5%~20%，1%/step
pmu_battery_warning_level2=xx	设置电池低电第二级报警门限：可设范围 1%~15%，1%/step
pmu_restvol_time=xx	设置电池电量更新时间间隔值，可设范围 30s/60s/120s
pmu_restvol_adjust_time =xx	设置校正电池电量时间间隔值 1，可设范围 30s/60s/120s
pmu_ocv_cou_adjust_time=xx	设置校正电池电量时间间隔值，可设范围 30s/60s/120s
pmu_chgled_func=xx	设置 CHGLED 引脚功能，0：由驱动程序控制，1：由充电逻辑控制
pmu_chgled_type	设置 CHGLD 由充电逻辑控制时，只是方式。0：方式 A 1：方式 B
pmu_vbusen_func	设置 N_VBUEN 引脚功能，0：作为输出功能 1：作为输入功能
pmu_reset	设置长按键 16s 后是否复位 PMU，0：不复位 1：复位
pmu_IRQ_wakeup	设置在 PMU 待机和关机情况下，IRQ 是否唤醒 PMU 功能：0：不唤醒 1：唤醒
pmu_hot_shutdownm	设置 PMU 内部温度过高是否自动关机保护功能，0：不关机 1：关机

pmu_inshort	设置 ACIN 和 VBUS 是否短路功能，0-由 PMU 自动检测 1：驱动设置为短路功能
-------------	--

配置举例：

```

pmu_used = 1
pmu_twi_addr = 0x34
pmu_twi_id = 1
pmu_irq_id = 0
pmu_battery_rdc = 100
pmu_battery_cap = 0
pmu_batdeten = 1
pmu_runtime_chgcur = 600
pmu_earlysuspend_chgcur = 900
pmu_suspend_chgcur = 1500
pmu_shutdown_chgcur = 1500
pmu_init_chgvol = 4200
pmu_init_chgend_rate = 15
pmu_init_chg_enabled = 1
pmu_init_adc_freq = 800
pmu_init_adcts_freq = 800
pmu_init_chg_pretime = 70
pmu_init_chg_csstime = 720
pmu_batt_cap_correct = 1
pmu_bat_regu_en = 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 = 2
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 = 26

```



pmu_bat_para16	= 36
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_usbvol_limit	= 1
pmu_usbcur_limit	= 0
pmu_usbvol	= 4400
pmu_usbcur	= 0
pmu_usbvol_pc	= 4400
pmu_usbcur_pc	= 900
pmu_pwroff_vol	= 3300
pmu_pwron_vol	= 2600
pmu_pekoff_time	= 6000
pmu_pekoff_func	= 0
pmu_pekoff_en	= 1
pmu_peklong_time	= 1500
pmu_pekcon_time	= 1000
pmu_pwrok_time	= 64
pmu_battery_warning_level1	= 15
pmu_battery_warning_level2	= 0
pmu_restvol_adjust_time	= 30
pmu_ocv_cou_adjust_time	= 60
pmu_chgled_func	= 0
pmu_chgled_type	= 0
pmu_vbusen_func	= 1
pmu_reset	= 0



pmu_IRQ_wakeup	= 0
pmu_hot_shutdownm	= 1
pmu_inshort	= 0
power_start	= 0

Confidential

36 动态电压频率(dvfs)

36.1 [dvfs_table]

配置项	配置项含义
max_freq	系统最高运行频率
min_freq	系统最低运行频率
LV_count	动态电压频率等级
LVx_freq	x: 从 1 到 LV_count; 当前等级最高可运行频率
LVx_volt	x: 从 1 到 LV_count; 当前等级最高可运行电压

说明:

一般的方案不要随便修改这组参数。

配置举例:

[dvfs_table]

max_freq = 912000000

min_freq = 60000000

LV_count = 7

LV1_freq = 1008000000

LV1_volt = 1450

LV2_freq = 912000000

LV2_volt = 1400

LV3_freq = 864000000

LV3_volt = 1300

LV4_freq = 720000000

LV4_volt = 1200



LV5_freq = 528000000

LV5_volt = 1100

LV6_freq = 312000000

LV6_volt = 1000

LV7_freq = 144000000

LV7_volt = 1000

Confidential