

A20 sys_config.fex 配置说明

V2.1
2014-06-05

CONFIDENTIAL

历史版本

	修改人	时间	备注
V1.0		2013-03-15	建立初始版本
V2.1		2014-06-05	增加 axp15_para 配置说明; 增加 boot_init_gpio 配置说明; 增加 anrecovery_key 配置说明; 增加 i2s_para 配置说明; 增加 memory_stick 配置说明; 增加 recovery_key 配置说明; 增加 ir_boot_para 配置说明; 增加 boot_disp 配置说明; 修改对齐。

目录

历史版本.....	- 2 -
目录.....	- 3 -
1 系统(System).....	- 8 -
1.1 [product].....	- 8 -
1.2 [platform].....	- 8 -
1.3 [target].....	- 8 -
1.4 [axp15_para].....	- 9 -
1.5 [clock].....	- 10 -
1.6 [boot_init_gpio].....	- 10 -
1.7 [card_boot].....	- 10 -
1.8 [card_boot0_para].....	- 11 -
1.9 [card_boot2_para].....	- 11 -
1.10 [twi_para].....	- 12 -
1.11 [uart_para].....	- 12 -
1.12 [uart_force_debug].....	- 13 -
1.13 [jtag_para].....	- 13 -
1.14 [pm_para].....	- 14 -
2 SDRAM.....	- 15 -
2.1 [dram_para].....	- 15 -
3 mali.....	- 17 -
3.1 [mail_para].....	- 17 -
4 EMAC.....	- 18 -
4.1 [emac_para].....	- 18 -
5 I ² C 总线.....	- 20 -
5.1 [twi0_para].....	- 20 -
5.2 [twi1_para].....	- 20 -
5.3 [twi2_para].....	- 20 -
5.4 [twi3_para].....	- 21 -
5.5 [twi4_para].....	- 21 -
6 串口(UART).....	- 22 -
6.1 [uart_para0].....	- 22 -
6.2 [uart_para1].....	- 22 -
6.3 [uart_para2].....	- 23 -
6.4 [uart_para3].....	- 24 -

6.5	[uart_para4].....	- 24 -
6.6	[uart_para5].....	- 25 -
6.7	[uart_para6].....	- 25 -
6.8	[uart_para7].....	- 26 -
7	SPI 总线.....	- 27 -
7.1	[spi0_para].....	- 27 -
7.2	[spi1_para].....	- 27 -
7.3	[spi2_para].....	- 28 -
7.4	[spi3_para].....	- 28 -
8	电阻屏(resistance tp).....	- 30 -
8.1	[rtp_para].....	- 30 -
9	电容屏(capacitor tp).....	- 31 -
9.1	[ctp_para].....	- 31 -
9.2	[ctp_list_para].....	- 32 -
10	触摸按键(touch key).....	- 33 -
10.1	[tkey_para].....	- 33 -
11	马达(motor).....	- 34 -
11.1	[motor_para].....	- 34 -
12	闪存 (nand flash)	- 35 -
12.1	[nand_para].....	- 35 -
13	显示初始化(disg init).....	- 37 -
13.1	[disg_init].....	- 37 -
14	LCD 屏 0.....	- 40 -
14.1	[lcd0_para].....	- 40 -
15	LCD 屏 1.....	- 44 -
15.1	[lcd1_para].....	- 44 -
16	TVIN/TVOUT.....	- 45 -
16.1	[tv_out_dac_para].....	- 45 -
16.2	[tvout_para].....	- 46 -
16.3	[tvin_para].....	- 46 -
17	HDMI.....	- 47 -
17.1	[hdmi_para].....	- 47 -
18	音频格式配置.....	- 48 -
18.1	[i2s2_para].....	- 48 -
19	摄像头(CSI).....	- 50 -
19.1	[camera_list_para].....	- 50 -
19.2	[csi0_para].....	- 51 -
19.3	[csi1_para].....	- 54 -
20	SATA 配置.....	- 56 -
A20 sysconfig.fex 配置说明		- 4 -

20.1	[sata_para].....	- 56 -
21	SD / MMC.....	- 57 -
21.1	[mmc0_para].....	- 57 -
21.2	[mmc1_para].....	- 58 -
21.3	[mmc2_para].....	- 59 -
21.4	[mmc3_para].....	- 60 -
22	memory stick 配置.....	- 61 -
22.1	[ms_para].....	- 61 -
23	SIM 卡配置.....	- 62 -
23.1	[smc_para].....	- 62 -
24	PS2 配置.....	- 63 -
24.1	[ps2_0_para].....	- 63 -
24.2	[ps2_1_para].....	- 63 -
25	CAN BUS 配置.....	- 64 -
25.1	[can_para].....	- 64 -
26	矩阵键盘配置.....	- 65 -
26.1	[keypad_para].....	- 65 -
27	USB 控制标志.....	- 67 -
27.1	[usbc0].....	- 67 -
27.2	[usbc1].....	- 68 -
27.3	[usbc2].....	- 69 -
28	USB Device.....	- 71 -
28.1	[usb_feature].....	- 71 -
28.2	[msc_feature].....	- 71 -
29	重力感应(G-sensor).....	- 73 -
29.1	[gsensor_para].....	- 73 -
29.2	[gsensor_list_para].....	- 73 -
30	GPS 配置.....	- 75 -
30.1	[gps_para].....	- 75 -
31	WIFI.....	- 76 -
31.1	[wifi_para].....	- 76 -
31.2	bcm40181 GPIO 配置.....	- 77 -
31.3	bcm40183 GPIO 配置.....	- 77 -
31.4	rtl8723as GPIO 配置.....	- 77 -
31.5	rtl8189es GPIO 配置.....	- 77 -
31.6	rtl8188eu GPIO 配置.....	- 77 -
31.7	ap6210 GPIO 配置.....	- 77 -
32	3G.....	- 79 -
32.1	[3g_para].....	- 79 -

33	gyroscope.....	- 80 -
33.1	[gy_para].....	- 80 -
34	光感(light sensor).....	- 81 -
34.1	[ls_para].....	- 81 -
35	罗盘 Compass.....	- 82 -
35.1	[compass_para].....	- 82 -
36	蓝牙(bluteeth).....	- 83 -
36.1	[bt_para].....	- 83 -
37	数字音频总线 (I2S)	- 84 -
37.1	[i2s_para].....	- 84 -
38	数字音频总线(pcm).....	- 85 -
38.1	[pcm_para].....	- 85 -
39	数字音频总线 (SPDIF)	- 88 -
39.1	[spdif_para].....	- 88 -
40	内置音频 (codec)	- 89 -
40.1	[audio_para].....	- 89 -
41	红外(ir).....	- 90 -
41.1	[ir_para].....	- 90 -
42	gpio 配置.....	- 91 -
42.1	[gpio_para].....	- 91 -
43	PMU 电源.....	- 92 -
43.1	[pmu_para].....	- 92 -
44	dvfs 电压-频率配置表.....	- 97 -
44.1	[dvfs_table].....	- 97 -
45	recovery key 配置.....	- 99 -
45.1	[recovery_key].....	- 99 -
45.2	[system].....	- 99 -
46	boot 阶段红外配置.....	- 100 -
46.1	[ir_boot_para].....	- 100 -
47	boot 阶段显示输出配置.....	- 101 -
47.1	[boot_disp].....	- 101 -

名称说明

1. 文档描述中蓝色为模块芯片引脚配置，不可更改；

A20 sysconfig.fex 配置说明

- 6 -

Copyright © 2014Allwinner Technology. All Rights Reserved.

-- 6 --

2. 描述 GPIO 配置的形式:

Port: 端口+组内序号<功能分配><内部电阻状态><驱动能力><输出电平状态>

3. 配置举例中的管脚不一定为真实可用的, 实际使用时需向技术支持人员询问。

4. 以 sugar-ref001 方案的参考配置为例说明, 位置:

lichee/tools/pack_brandy/chips/sun7i/configs/android/sugar-ref001/sys_config.fex

CONFIDENTIAL

1 系统(System)

1.1 [product]

配置项	配置项含义
version=1	版本信息，仅作标识本文件的版本信息
machine = "sugar-ref001"	方案名称，仅作标识本文件对应的方案名称

配置举例：

[product]

version = "100"

machine = "sugar-ref001"

1.2 [platform]

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

配置举例：

[platform]

eraseflag = 1

1.3 [target]

配置项	配置项含义
boot_clock=xx	启动频率; xx 表示多少 MHZ
dc2c2_vol=1400	dc2c2 的输出电压, mV
dc2c3_vol=1250	dc2c3 的输出电压, mV
ldo2_vol=3000	ldo2 的输出电压, mV
ldo3_vol=2800	ldo3 的输出电压, mV
ldo3_vol=2800	ldo4 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:
usb_recovery = 1	默认为 1 (盒子未用, 配置无效)

配置举例:

[target]

```
boot_clock      = 912
dcdc2_vol       = 1400
dcdc3_vol       = 1250
ldo2_vol        = 3000
ldo3_vol        = 2800
ldo4_vol        = 2800
power_start     = 0
storage_type    = -1
usb_recovery    = 1
```

1.4 [axp15_para]

配置项	配置项含义
dcdc1_vol=3000	dcdc1 的输出电压, mV
dcdc2_vol=1250	dcdc2 的输出电压, mV
dcdc3_vol=1500	dcdc3 的输出电压, mV
dcdc4_vol=1230	dcdc4 的输出电压, mV
aldo1_vol=3000	aldo1 的输出电压, mV
aldo2_vol=3000	aldo2 的输出电压, mV
dldo1_vol=3300	dldo1 的输出电压, mV
dldo2_vol=3300	dldo2 的输出电压, mV

配置举例:

[axp15_para]

```
dcdc1_vol  = 3000
dcdc2_vol  = 1250
dcdc3_vol  = 1500
dcdc4_vol  = 1250
aldo1_vol  = 3000
aldo2_vol  = 3000
dldo1_vol  = 3300
dldo2_vol  = 3300
```


配置举例：

```
[card_boot]
logical_start          = 40960
sprite_gpio0           = port:PH13<1><default><default><0>
sprite_work_delay      = 500
sprite_err_delay       = 200
```

1.8 [card_boot0_para]

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

配置举例：

```
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>
```

1.9 [card_boot2_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=xx	sdc 卡命令信号的 GPIO 配置
sdc_clk=xx	sdc 卡时钟信号的 GPIO 配置
sdc_d0=xx	sdc 卡数据 0 线信号的 GPIO 配置
sdc_d1=xx	sdc 卡数据 1 线信号的 GPIO 配置
sdc_d3=xx	sdc 卡数据 3 线信号的 GPIO 配置
sdc_d2=xx	sdc 卡数据 2 线信号的 GPIO 配置

配置举例：

```
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>
```

1.10 [twi_para]

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

配置举例：

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

1.11 [uart_para]

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

配置举例：

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

1.12 [uart_force_debug]

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

配置举例：

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

1.13 [jtag_para]

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

配置举例：

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

1.14 [pm_para]

配置项	配置项含义
standby_mode = x	休眠模式配置： 0 -- normal standby 1 -- super standby
usbhid_wakeup_enable = 0	是否支持 usb 唤醒： 0 -- 不支持 usb 唤醒 1 -- 支持 usb 唤醒 注：使能 usb 唤醒，会导致待机功耗增加

配置举例：

[pm_para]

standby_mode = 0

usbhid_wakeup_enable = 0

2 SDRAM

2.1 [dram_para]

注意:

1. dram 参数直接影响系统的稳定性, 请勿随意修改, 如有疑问必须咨询 FAE 进行确认。

配置项	配置项含义
dram_baseaddr = 0x40000000	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 -- DDR2 3 -- 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

A20 sysconfig.fex 配置说明

- 15 -

Copyright © 2014 Allwinner Technology. All Rights Reserved.

-- 15 --

dram_clk	= 432
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	= 0x38D48893
dram_tpr1	= 0xa0a0
dram_tpr2	= 0x22a00
dram_tpr3	= 0x0
dram_tpr4	= 0x0
dram_tpr5	= 0x0
dram_emr1	= 0x4
dram_emr2	= 0x10
dram_emr3	= 0x0

3 mali

3.1 [mail_para]

配置项	配置项含义
mali_used = 1	是否使用 mali (GPU): 0 -- disable 1 -- enable
mali_clkdiv = 1	GPU 时钟分频设置

配置举例:

[mail_para]

mali_used = 1

mali_clkdiv = 1

4 EMAC

4.1 [emac_para]

配置项	配置项含义
emac_used=0	emac 模块是否使能：1：enable 0：disable
emac_txd0=xx	emac tx0 的 GPIO 配置
emac_txd1=xx	emac tx1 的 GPIO 配置
emac_txd2=xx	emac tx2 的 GPIO 配置
emac_txd3=xx	emac tx3 的 GPIO 配置
emac_rxd0=xx	emac rx0 的 GPIO 配置
emac_rxd1=xx	emac rx1 的 GPIO 配置
emac_rxd2=xx	emac rx2 的 GPIO 配置
emac_rxd3=xx	emac rx3 的 GPIO 配置
emac_rxdv=xx	emac 接收数有效使能
emac_rxclk=xx	emac 接收时钟
emac_rxerr=xx	emac 接收错误使能
emac_mdc=xx	emac 配置接口时钟
emac_mdio=xx	emac 配置接口数据 I/O
emac_txen	emac 发送使能 GPIO 设置
emac_txclk	emac 发送设置设置
emac_crs	emac 载波检测
emac_col	emac 冲突检测
emac_reset	emac 复位 GPIO 设置
emac_power	emac 的电源控制管脚 GPIO 配置

配置举例：

```

emac_used                = 1
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>

```

A20 sysconfig.fex 配置说明

- 18 -

Copyright © 2014Allwinner Technology. All Rights Reserved.

-- 18 --

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>
emac_power	= port:PH15<1><default><default><0>

CONFIDENTIAL

5 I²C 总线

主控有 5 个 I²C (TWI) 控制器，分别通过 twi0_para、twi1_para、twi2_para、twi3_para、twi4_para 进行配置。

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:PB0<2><default><default><default>
twi0_sda           = port:PB1<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:PB18<2><default><default><default>
twi1_sda           = port:PB19<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:PB20<2><default><default><default>
twi2_sda           = port:PB21<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：

[twi3_para]

```
twi3_used          = 1
twi3_scl           = port:PI0<3><default><default><default>
twi3_sda           = port:PI1<3><default><default><default>
```

5.5 [twi4_para]

配置项	配置项含义
twi4_used=xx	TWI 使用控制：1 使用，0 不用
twi4_scl=xx	TWI SCK 的 GPIO 配置
twi4_sda=xx	TWI SDA 的 GPIO 配置

配置举例：

[twi4_para]

```
twi4_used          = 1
twi4_scl           = port:PI2<3><default><default><default>
twi4_sda           = port:PI3<3><default><default><default>
```

6 串口(UART)

主控有 8 路 uart 接口，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（但强烈不建议使用用 **uart0** 作为控制台以外的用途，以方便调试定位问题）。

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
uart0_tx =xx	UART TX 的 GPIO 配置
uart0_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
uart1_tx =xx	UART TX 的 GPIO 配置
uart1_rx =xx	UART RX 的 GPIO 配置
uart1_rts=xx	UART RTS 的 GPIO 配置
uart1_cts=xx	UART CTS 的 GPIO 配置
uart1_dtr=xx	UART DTR 的 GPIO 配置
uart1_dsr=xx	UART DSR 的 GPIO 配置

uart1_dcd=xx	UART DCD 的 GPIO 配置
uart1_ring=xx	UART RING 的 GPIO 配置

配置举例：

[uart_para1]

```

uart_used          = 0
uart_port          = 1
uart_type          = 8
uart1_tx           = port:PA10<4><default><default><default>
uart1_rx           = port:PA11<4><default><default><default>
uart1_rts           = port:PA12<4><default><default><default>
uart1_cts           = port:PA13<4><default><default><default>
uart1_dtr           = port:PA14<4><default><default><default>
uart1_dsr           = port:PA15<4><default><default><default>
uart1_dcd           = port:PA16<4><default><default><default>
uart1_ring         = port:PA17<4><default><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
uart2_tx =xx	UART TX 的 GPIO 配置
uart2_rx =xx	UART RX 的 GPIO 配置
uart2_rts=xx	UART RTS 的 GPIO 配置
uart2_cts=xx	UART CTS 的 GPIO 配置

配置举例：

[uart_para2]

```

uart_used          = 1
uart_port          = 2
uart_type          = 4
uart2_tx           = port:PI18<3><default><default><default>
uart2_rx           = port:PI19<3><default><default><default>
uart2_rts           = port:PI16<3><default><default><default>
uart2_cts           = port:PI17<3><default><default><default>

```

A20 sysconfig.fex 配置说明

- 23 -

Copyright © 2014Allwinner Technology. All Rights Reserved.

-- 23 --

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
uart3_tx =xx	UART TX 的 GPIO 配置
uart3_rx =xx	UART RX 的 GPIO 配置
uart3_rts=xx	UART RTS 的 GPIO 配置
uart3_cts=xx	UART CTS 的 GPIO 配置

配置举例：

[uart_para3]

```

uart_used          = 0
uart_port          = 3
uart_type          = 4
uart3_tx           = port:PH00<4><default><default><default>
uart3_rx           = port:PH01<4><default><default><default>
uart3_rts          = port:PH02<4><default><default><default>
uart3_cts          = port:PH03<4><default><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
uart4_tx =xx	UART TX 的 GPIO 配置
uart4_rx =xx	UART RX 的 GPIO 配置

配置举例：

[uart_para4]

```

uart_used          = 0
uart_port          = 4
uart_type          = 2
uart4_tx           = port:PH04<4><default><default><default>

```


uart4_rx = port:PH05<4><default><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
uart5_tx =xx	UART TX 的 GPIO 配置
uart5_rx =xx	UART RX 的 GPIO 配置

配置举例：

[uart_para5]

```

uart_used          = 0
uart_port          = 5
uart_type          = 2
uart5_tx           = port:PH06<4><default><default><default>
uart5_rx           = port:PH07<4><default><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
uart6_tx =xx	UART TX 的 GPIO 配置
uart6_rx =xx	UART RX 的 GPIO 配置

配置举例：

[uart_para6]

```

uart_used          = 0
uart_port          = 6
uart_type          = 2
uart6_tx           = port:PA12<3><default><default><default>
uart6_rx           = port:PA13<3><default><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
uart7_tx =xx	UART TX 的 GPIO 配置
uart7_rx =xx	UART RX 的 GPIO 配置

配置举例：

[uart_para7]

uart_used = 0

uart_port = 7

uart_type = 2

uart7_tx = port:PA14<3><default><default><default>

uart7_rx = port:PA15<3><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	使用哪路 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	使用哪路 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<3><default><default><default>
spi_cs1                 = port:PA04<3><default><default><default>
spi_sclk                = port:PA01<3><default><default><default>
spi_mosi                = port:PA02<3><default><default><default>
spi_miso                = port:PA03<3><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	使用的片选选择，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:PC19<3><default><default><default>
spi_cs1                 = port:PB13<3><default><default><default>
spi_sclk                = port:PC20<3><default><default><default>
spi_mosi                = port:PC21<3><default><default><default>
spi_miso                = port:PC22<3><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	使用的片选选择，bit 位表示
spi_cs0=xx	SPI CS0 的 GPIO 配置

<code>spi_cs1=xx</code>	SPI CS1 的 GPIO 配置
<code>spi_sclk=xx</code>	SPI CLK 的 GPIO 配置
<code>spi_mosi=xx</code>	SPI MOSI 的 GPIO 配置
<code>spi_miso=xx</code>	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_cs1                 = port:PA09<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>
```

8 电阻屏(resistance tp)

8.1 [rtp_para]

注意：盒子方案可忽略

配置项	配置项含义
rtp_used=xx	是否启用电阻触摸屏： 0 -- disable 1 -- enable
rtp_screen_size=xx	屏幕尺寸设置（单位 inch）
rtp_regidity_level=xx	屏幕硬度
rtp_press_threshold_enable=xx	是否开启压力触发的阀门控制
rtp_press_threshold=xx	仅当 rtp_press_threshold_enable=1 时有效，敏感度设置
rtp_sensitive_level=xx	敏感等级
rtp_exchange_x_y_flag	是否需要 x 轴 y 轴坐标对换

配置举例：

[rtp_para]

rtp_used = 1
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

9 电容屏(capacitor tp)

注意：盒子方案可忽略

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_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              = 2
ctp_twi_addr            = 0x5d
ctp_screen_max_x        = 800
ctp_screen_max_y        = 480
ctp_revert_x_flag       = 0
ctp_revert_y_flag       = 0
ctp_exchange_x_y_flag   = 0
ctp_int_port            = port:PH21<6><default><default><default>
ctp_wakeup              = port:PB13<1><default><default><1>

```

9.2 [ctp_list_para]

配置项	配置项含义
ctp_det_used = 0	是否支持 ctp 驱动自动检测： 0 -- 不支持 1 -- 支持
ft5x_ts = 0	是否支持 ft5x_ts
gt82x = 0	是否支持 gt82x
gslX680 = 0	是否支持 gslX680
gt9xx_ts = 0	是否支持 gt9xx_ts
gt811 = 0	是否支持 gt811

配置举例：

[ctp_list_para]

ctp_det_used = 0

ft5x_ts = 0

gt82x = 0

gslX680 = 0

gt9xx_ts = 0

gt811 = 0

10 触摸按键(touch key)

注意：盒子方案可忽略

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：

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

11 马达(motor)

注意：盒子方案可忽略

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 = 0
motor_shake = port:PB03<1><default><default><1>

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	nandRead/Busy 1 信号的 GPIO 配置
nand_rb1=xx	nandRead/Busy 0 信号的 GPIO 配置
nand_d0=xx	nand 数据总线信号的 GPIO 配置, bit0
nand_d1=xx	nand 数据总线信号的 GPIO 配置, bit1
nand_d2=xx	nand 数据总线信号的 GPIO 配置, bit2
nand_d3=xx	nand 数据总线信号的 GPIO 配置, bit3
nand_d4=xx	nand 数据总线信号的 GPIO 配置, bit4
nand_d5=xx	nand 数据总线信号的 GPIO 配置, bit5
nand_d6=xx	nand 数据总线信号的 GPIO 配置, bit6
nand_d7=xx	nand 数据总线信号的 GPIO 配置, bit7
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 片选 4 信号的 GPIO 配置
nand_ce5=xx	nand 片选 5 信号的 GPIO 配置
nand_ce6=xx	nand 片选 6 信号的 GPIO 配置
nand_ce7=xx	nand 片选 7 信号的 GPIO 配置
nand_ndqs=xx	nand0 ddr 时钟信号的 GPIO 配置
good_block_ratio=xx	NAND 控制器内部参数, 由原厂来进行调节, 请勿修改
id_number_ctl	NAND 控制器内部参数, 由原厂来进行调节, 请勿修改
nand_p0	NAND 控制器内部参数, 由原厂来进行调节, 请勿修改
nand_p1	NAND 控制器内部参数, 由原厂来进行调节, 请勿修改

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
id_number_ctl              = 0x3
nand_p0                    = 0x28010020
nand_p1                    = 0x01eeeeee
```

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> 1: screen1<screen1,fb0> 2: two_diff_screen_diff_contents 3: two_same_screen_diff_contents 4: two_diff_screen_same_contents
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_framebuffer_num=xx	framebuffer 个数
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_framebuffer_num=xx	framebuffer 个数
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             = 4
screen0_output_type   = 3
screen0_output_mode   = 5
screen1_output_type   = 1
screen1_output_mode   = 4
fb0_framebuffer_num   = 2
fb0_format            = 10
fb0_pixel_sequence    = 0
fb0_scaler_mode_enable = 1
fb0_width             = 0
fb0_height            = 0
fb1_framebuffer_num   = 2
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

```

CONFIDENTIAL

14 LCD 屏 0

备注：盒子产品未使用 LCD 输出，默认关闭。

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, 输出接口类型 (0:hv(sync+de), 1:8080, 2:ttl, 3:lvds, 4:dsi, 5:edp)
lcd_x=xx	lcd active width, 输出宽度
lcd_y=xx	lcd active height, 输出高度
lcd_dclk_freq=xx	pixel clock, 像素时钟, in MHZ unit
lcd_pwm_freq=xx	pwm freq, 背光 PWM 频率, in HZ unit
lcd_pwm_pol=xx	pwm polarity, PWM 极性配置, 0:positive, 1:negative
lcd_pwm_max_limit=xx	Lcd pwm max limit(<=255)
lcd_hbp=xx	hsync back porch, hsync 消隐配置
lcd_ht=xx	hsync total cycle, hsync 总时钟
lcd_vbp=xx	vsync back porch, vsync 消隐配置
lcd_vt=xx	vsync total cycle, vsync 总时钟
lcd_hv_vspw=xx	vsync plus width, vsync 时钟宽度
lcd_hv_hspw=xx	hsync plus width, hsync 时钟宽度
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 = 0

lcd_if = 3

A20 sysconfig.fex 配置说明

- 41 -

Copyright © 2014Allwinner Technology. All Rights Reserved.

-- 41 --

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:PH07<1><0><default><1>
lcd_power	= port:PH08<1><0><default><1>
lcd_pwm	= port:PB02<2><0><default><default>
lcd_gpio_scl	=

```

lcd_gpio_sda      =
lcd_gpio_0        =
lcd_gpio_1        =
lcd_gpio_2        =
lcd_gpio_3        =
lcdd0             = port:PD00<2><0><default><default>
lcdd1             = port:PD01<2><0><default><default>
lcdd2             = port:PD02<2><0><default><default>
lcdd3             = port:PD03<2><0><default><default>
lcdd4             = port:PD04<2><0><default><default>
lcdd5             = port:PD05<2><0><default><default>
lcdd6             = port:PD06<2><0><default><default>
lcdd7             = port:PD07<2><0><default><default>
lcdd8             = port:PD08<2><0><default><default>
lcdd9             = port:PD09<2><0><default><default>
lcdd10            = port:PD10<2><0><default><default>
lcdd11            = port:PD11<2><0><default><default>
lcdd12            = port:PD12<2><0><default><default>
lcdd13            = port:PD13<2><0><default><default>
lcdd14            = port:PD14<2><0><default><default>
lcdd15            = port:PD15<2><0><default><default>
lcdd16            = port:PD16<2><0><default><default>
lcdd17            = port:PD17<2><0><default><default>
lcdd18            = port:PD18<2><0><default><default>
lcdd19            = port:PD19<2><0><default><default>
lcdd20            = port:PD20<2><0><default><default>
lcdd21            = port:PD21<2><0><default><default>
lcdd22            = port:PD22<2><0><default><default>
lcdd23            = 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，请参看 lcd0 配置。

CONFIDENTIAL

16 TVIN/TVOUT

16.1 [tv_out_dac_para]

配置项	配置项含义
dac_used=xx	是否使用 tv dac 输出, 1: 使用, 0: 不使用
dac0_src	dac 通道 0 的输出类型: 0: COMPOSITE 1: LUMA 2: CHROMA 4: Y 5: PB 6: PR
dac1_src	dac 通道 1 的输出类型: 0: COMPOSITE 1: LUMA 2: CHROMA 4: Y 5: PB 6: PR
dac2_src	dac 通道 2 的输出类型: 0: COMPOSITE 1: LUMA 2: CHROMA 4: Y 5: PB 6: PR
dac3_src	dac 通道 3 的输出类型: 0: COMPOSITE 1: LUMA 2: CHROMA 4: Y 5: PB 6: PR

配置举例：

[tv_out_dac_para]

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

16.2 [tvout_para]

配置项	配置项含义
tvout_used =xx	是否使用 tvout, 1: 使用, 0: 不使用
tvout_channel_num=xx	使用的 tvout 通道号

配置举例：

[tvout_para]

```
tvout_used      = 1
tvout_channel_num = 1
```

16.3 [tvvin_para]

配置项	配置项含义
tvvin_used =xx	是否使用 tvvin, 1: 使用, 0: 不使用
tvvin_channel_num=xx	使用的 tvvin 通道号

配置举例：

[tvvin_para]

```
tvvin_used      = 0
tvvin_channel_num = 4
```

17 HDMI

17.1 [hdmi_para]

配置项	配置项含义
hdmi_used =xx	是否使用 hdmi, 1: 使用, 0: 不使用
hdcp_enable=xx	是否使能 hdcp(数字内容保护), 1: enable, 0: disable

配置举例:

[hdmi_para]

hdmi_used = 1

hdcp_enable = 0

18 音频格式配置

18.1 [i2s2_para]

配置项	配置项含义
i2s_channel	
i2s_master	1: codec clk & FRM master 2: codec clk slave & FRM master 3: codec clk master & frame slave 4: codec clk & FRM slave
i2s_select	1: pcm 0: i2s
audio_format	1: standard i2s format 2: right justified format 3: left justified format 4: pcm. MSB is available on 2nd BCLK rising edge after LRC rising edge 5: pcm. MSB is available on 1nd BCLK rising edge after LRC rising edge
signal_inversion	1: normal bit clock + frame 2: normal BCLK + inv FRM 3: invert BCLK + nor FRM 4: invert BCLK + FRM
over_sample_rate	过采样设置: 128fs/192fs/256fs/384fs/512fs/768fs
sample_resolution	采样精度: 16bits/20bits/24bits
word_select_size	字宽度: 16bits/20bits/24bits/32bits
pcm_sync_period	同步周期: 16/32/64/128/256
msb_lsb_first	0: msb 1: lsb
sign_extend	0: zero pending; 1: sign extend
slot_index	接口索引
slot_width	solt 位宽:

	8/16
frame_width	0: long frame = 2 clock width; 1: short frame
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

配置举例:

[i2s2_para]

i2s_channel = 2

i2s_master = 4

i2s_select = 1

audio_format = 1

signal_inversion = 1

over_sample_rate = 256

sample_resolution = 16

word_select_size = 32

pcm_sync_period = 256

msb_lsb_first = 0

sign_extend = 0

slot_index = 0

slot_width = 16

frame_width = 1

tx_data_mode = 0

rx_data_mode = 0

19 摄像头(CSI)

19.1 [camera_list_para]

配置项	配置项含义
camera_list_para_used	是否支持 camera 自动检测, 0: 不支持, 1: 支持
ov7670	是否支持此型号摄像头, 0: 不支持, 1: 支持
gc0308	是否支持此型号摄像头, 0: 不支持, 1: 支持
gt2005	是否支持此型号摄像头, 0: 不支持, 1: 支持
hi704	是否支持此型号摄像头, 0: 不支持, 1: 支持
sp0838	是否支持此型号摄像头, 0: 不支持, 1: 支持
mt9m112	是否支持此型号摄像头, 0: 不支持, 1: 支持
mt9m113	是否支持此型号摄像头, 0: 不支持, 1: 支持
ov2655	是否支持此型号摄像头, 0: 不支持, 1: 支持
hi253	是否支持此型号摄像头, 0: 不支持, 1: 支持
gc0307	是否支持此型号摄像头, 0: 不支持, 1: 支持
mt9d112	是否支持此型号摄像头, 0: 不支持, 1: 支持
ov5640	是否支持此型号摄像头, 0: 不支持, 1: 支持
gc2015	是否支持此型号摄像头, 0: 不支持, 1: 支持
ov2643	是否支持此型号摄像头, 0: 不支持, 1: 支持
gc0329	是否支持此型号摄像头, 0: 不支持, 1: 支持
gc0309	是否支持此型号摄像头, 0: 不支持, 1: 支持
tvp5150	是否支持此型号摄像头, 0: 不支持, 1: 支持
s5k4ec	是否支持此型号摄像头, 0: 不支持, 1: 支持
ov5650_mv9335	是否支持此型号摄像头, 0: 不支持, 1: 支持
siv121d	是否支持此型号摄像头, 0: 不支持, 1: 支持

配置举例:

[camera_list_para]

```
camera_list_para_used    = 1
ov7670                  = 0
gc0308                  = 1
gt2005                  = 0
hi704                   = 0
sp0838                  = 0
```

A20 sysconfig.fex 配置说明

- 50 -

Copyright © 2014Allwinner Technology. All Rights Reserved.

-- 50 --

mt9m112	= 0
mt9m113	= 0
ov2655	= 0
hi253	= 0
gc0307	= 0
mt9d112	= 0
ov5640	= 1
gc2015	= 0
ov2643	= 0
gc0329	= 0
gc0309	= 0
tpv5150	= 0
s5k4ec	= 0
ov5650_mv9335	= 0
siv121d	= 0

19.2 [csi0_para]

配置项	配置项含义
csi_used=xx	是否使用，0：不使用，1：使用
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
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”axp xx _Xldo N ”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_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”axp xx _Xldo N ”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”axp 22 _Xldo N ”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 如果 csi_iovdd 配置不为空时会配置对应的 axp 电源为相应电压 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
csi_vol_avdd	配置 csi avdd 电源电压 如果 csi_avdd 配置不为空时会配置对应的 axp 电源为相应电压 配置为 2800 表示 2.8V ，一般不要修改此数值
csi_vol_dvdd	配置 csi dvdd 电源电压 如果 csi_dvdd 配置不为空时会配置对应的 axp 电源为相应电压 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
csi_pck=xx	模组送给 csi 的 clock 的 GPIO 配置
csi_ck=xx	csi 送给模组的 clock 的 GPIO 配置
csi_hsync=xx	模组送给 csi 的行同步信号 GPIO 配置
csi_vsync=xx	模组送给 csi 的帧同步信号 GPIO 配置
csi_d0=xx ... csi_d23=xx	模组送给 csi 的 8bit/16bit/24bit 数据的 GPIO 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 pwn/reset 信号，使用 RAW 格式的 sensor 只

	能用 csi_d0/d1 作 GPIO 用途。
csi_reset=xx	控制模组的 reset 的 GPIO 配置，默认值为 reset 有效（高或低有效需要取决于模组）
csi_power_en=xx	控制模组的电源的 GPIO 配置，若 csi_stby_mode 配置成 0，则 csi_power_en 的默认值一般配置成 1；若 csi_stby_mode 配置成 1，则 csi_power_en 的默认值一般配置成 0。
csi_stby=xx	控制模组的 standby 的 GPIO 配置，默认值为 standby 有效（高或低有效需要取决于模组）

配置举例：

```
csi_used                = 1
csi_dev_qty             = 1
csi_stby_mode           = 0
csi_mname               = "gc0308"
csi_if                  = 0
csi_iovdd               = ""
csi_avdd                = ""
csi_dvdd               = ""
csi_vol_iovdd           =
csi_vol_dvdd            =
csi_vol_avdd            =
csi_vflip               = 0
csi_hflip               = 0
csi_flash_pol           = 0
csi_facing              = 0
csi_twi_id              = 1
csi_twi_addr            = 0x42
csi_pck                 = port:PE00<3><default><default><default>
csi_ck                  = port:PE01<3><default><default><default>
csi_hsync               = port:PE02<3><default><default><default>
csi_vsync               = port:PE03<3><default><default><default>
csi_d0                  = port:PE04<3><default><default><default>
csi_d1                  = port:PE05<3><default><default><default>
csi_d2                  = port:PE06<3><default><default><default>
csi_d3                  = port:PE07<3><default><default><default>
csi_d4                  = port:PE08<3><default><default><default>
csi_d5                  = port:PE09<3><default><default><default>
csi_d6                  = port:PE10<3><default><default><default>
```

```
csi_d7          = port:PE11<3><default><default><default>
csi_reset       = port:PH13<1><default><default><0>
csi_power_en    =
csi_stby        = port:PH16<1><default><default><0>
```

19.3 [csi1_para]

配置项说明如[csi0_para].

配置举例:

[csi1_para]

```
csi_used        = 0

csi_dev_qty     = 1
csi_stby_mode   = 0
csi_mname       = "gc0308"
csi_if          = 0
csi_iovdd       = ""
csi_avdd        = ""
csi_dvdd        = ""
csi_vol_iovdd   =
csi_vol_dvdd    =
csi_vol_avdd    =
csi_vflip       = 0
csi_hflip       = 0
csi_flash_pol   = 0
csi_facing      = 1
csi_twi_id      = 1
csi_twi_addr    = 0x42
csi_pck         = port:PG00<3><default><default><default>
csi_ck          = port:PG01<3><default><default><default>
csi_hsync       = port:PG02<3><default><default><default>
csi_vsync       = port:PG03<3><default><default><default>
csi_d0          = port:PG04<3><default><default><default>
csi_d1          = port:PG05<3><default><default><default>
csi_d2          = port:PG06<3><default><default><default>
csi_d3          = port:PG07<3><default><default><default>
csi_d4          = port:PG08<3><default><default><default>
```

csi_d5	= port:PG09<3><default><default><default>
csi_d6	= port:PG10<3><default><default><default>
csi_d7	= port:PG11<3><default><default><default>
csi_reset	= port:PH14<1><default><default><0>
csi_power_en	=
csi_stby	= port:PH17<1><default><default><0>

CONFIDENTIAL

20 SATA 配置

20.1 [sata_para]

配置项	配置项含义
sata_used =xx	是否使用 sata, 1: 使用, 0: 不使用
sata_power_en=xx	SATA 电源控制 GPIO

配置举例:

[sata_para]

sata_used = 1

sata_power_en =

21 SD / MMC

21.1 [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, 4 - manual mode(from proc file system node)
bus_width=xx	数据总线位宽: 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	假如该卡支持 SD3.0 或者 emmc4.5 的 UHS-I/DDR、HS200, 这里就要写成供电电源

配置举例:

[mmc0_para]

```
sdc_used          = 1
sdc_detmode       = 1
sdc_buswidth      = 4
sdc_clk           = port:PF02<2><1><2><default>
sdc_cmd           = port:PF03<2><1><2><default>
sdc_d0            = port:PF01<2><1><2><default>
sdc_d1            = port:PF00<2><1><2><default>
sdc_d2            = port:PF05<2><1><2><default>
```

A20 sysconfig.fex 配置说明

- 57 -

Copyright © 2014Allwinner Technology. All Rights Reserved.

-- 57 --

```
sdc_d3          = port:PF04<2><1><2><default>
sdc_det         = port:PH1<0><1><default><default>
sdc_use_wp      = 0
sdc_wp          =
sdc_isio        = 0
sdc_regulator   = "none"
```

21.2 [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, 4 - manual mode(from proc file system node)
bus_width=xx	数据总线位宽: 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          = 0
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>
```

A20 sysconfig.fex 配置说明

- 58 -

Copyright © 2014Allwinner Technology. All Rights Reserved.

-- 58 --

```
sdc_det =
sdc_use_wp = 0
sdc_wp =
sdc_isio = 0
sdc_regulator = "none"
```

21.3 [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, 4 - manual mode(from proc file system node)
bus_width=xx	数据总线位宽: 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 = 0
sdc_detmode = 3
bus_width = 4
sdc_cmd = port:PC06<3><1><2><default>
sdc_clk = port:PC07<3><1><2><default>
sdc_d0 = port:PC08<3><1><2><default>
sdc_d1 = port:PC09<3><1><2><default>
sdc_d2 = port:PC10<3><1><2><default>
sdc_d3 = port:PC11<3><1><2><default>
sdc_use_wp = 0
```

A20 sysconfig.fex 配置说明

- 59 -

Copyright © 2014Allwinner Technology. All Rights Reserved.

-- 59 --

sdc_wp

=

21.4 [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, 4 - manual mode(from proc file system node)
bus_width=xx	数据总线位宽: 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:

[mmc3_para]

sdc_used = 1

sdc_detmode = 1

bus_width = 4

sdc_cmd = port:PI04<2><1><2><default>

sdc_clk = port:PI05<2><1><2><default>

sdc_d0 = port:PI06<2><1><2><default>

sdc_d1 = port:PI07<2><1><2><default>

sdc_d2 = port:PI08<2><1><2><default>

sdc_d3 = port:PI09<2><1><2><default>

sdc_use_wp = 0

sdc_wp =

22 memory stick 配置

22.1 [ms_para]

配置项	配置项含义
ms_used	是否使用 memory stick, 1: 使用, 0: 不使用
ms_bs	总线状态 GPIO 配置
ms_clk	时钟 GPIO 配置
ms_d0	数据线 bit0 配置
ms_d0	数据线 bit1 配置
ms_d0	数据线 bit2 配置
ms_d0	数据线 bit3 配置
ms_det	卡检测 GPIO 配置

配置举例:

[ms_para]

```

ms_used          = 0
ms_bs            = port:PH06<5><default><default><default>
ms_clk           = port:PH07<5><default><default><default>
ms_d0            = port:PH08<5><default><default><default>
ms_d1            = port:PH09<5><default><default><default>
ms_d2            = port:PH10<5><default><default><default>
ms_d3            = port:PH11<5><default><default><default>
ms_det           =
  
```

23 SIM 卡配置

注意：盒子方案可忽略

23.1 [smc_para]

配置项	配置项含义
smc_used	是否使用 SIM 卡，1：使用，0：不使用
smc_rst	SIM 卡复位 GPIO 设置
smc_vppen	SIM 卡电源使能 GPIO 设置
smc_vppp	SIM 卡电源配置
smc_det	SIM 卡检测 GPIO 设置
smc_vccen	SIM 卡电源使能 GPIO 设置
smc_sck	SIM 卡时钟 GPIO 设置
smc_sda	SIM 卡数据 GPIO 设置

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>
```

24 PS2 配置

24.1 [ps2_0_para]

配置项	配置项含义
ps2_used =xx	是否使用 ps2, 1: 使用, 0: 不使用
ps2_scl=xx	时钟管脚的 GPIO 配置
ps2_sda=xx	数据管脚的 GPIO 配置

配置举例:

[ps2_0_para]

ps2_used = 0

ps2_scl = port:PI20<2><1><default><default>

ps2_sda = port:PI21<2><1><default><default>

24.2 [ps2_1_para]

配置项	配置项含义
ps2_used =xx	是否使用 ps2, 1: 使用, 0: 不使用
ps2_scl=xx	时钟管脚的 GPIO 配置
ps2_sda=xx	数据管脚的 GPIO 配置

配置举例:

[ps2_1_para]

ps2_used = 0

ps2_scl = port:PI14<3><1><default><default>

ps2_sda = port:PI15<3><1><default><default>

25 CAN BUS 配置

注意：盒子方案可忽略

25.1 [can_para]

配置项	配置项含义
can_used =xx	是否使用 can, 1: 使用, 0: 不使用
can_tx=xx	can bus 发送 GPIO 配置
can_rx=xx	can bus 接受 GPIO 配置

配置举例:

[can_para]

can_used = 0

can_tx = port:PA16<3><default><default><default>

can_rx = port:PA17<3><default><default><default>

26 矩阵键盘配置

26.1 [keypad_para]

配置项	配置项含义
kp_used =xx	是否使用矩阵键盘，1：使用，0：不使用
kp_in_size	矩阵键盘输入（行）宽度
kp_out_size	矩阵键盘输出（列）宽度
kp_in0	矩阵键盘输入 bit0 配置
kp_in1	矩阵键盘输入 bit1 配置
kp_in2	矩阵键盘输入 bit2 配置
kp_in3	矩阵键盘输入 bit3 配置
kp_in4	矩阵键盘输入 bit4 配置
kp_in5	矩阵键盘输入 bit5 配置
kp_in6	矩阵键盘输入 bit6 配置
kp_in7	矩阵键盘输入 bit7 配置
kp_out0	矩阵键盘输出 bit0 配置
kp_out1	矩阵键盘输出 bit1 配置
kp_out2	矩阵键盘输出 bit2 配置
kp_out3	矩阵键盘输出 bit3 配置
kp_out4	矩阵键盘输出 bit4 配置
kp_out5	矩阵键盘输出 bit5 配置
kp_out6	矩阵键盘输出 bit6 配置
kp_out7	矩阵键盘输出 bit7 配置

配置举例：

[keypad_para]

```
kp_used          = 0
kp_in_size       = 8
kp_out_size      = 8
kp_in0           = port:PH08<4><1><default><default>
kp_in1           = port:PH09<4><1><default><default>
kp_in2           = port:PH10<4><1><default><default>
kp_in3           = port:PH11<4><1><default><default>
kp_in4           = port:PH14<4><1><default><default>
```

A20 sysconfig.fex 配置说明

- 65 -

Copyright © 2014Allwinner Technology. All Rights Reserved.

-- 65 --

kp_in5	= port:PH15<4><1><default><default>
kp_in6	= port:PH16<4><1><default><default>
kp_in7	= port:PH17<4><1><default><default>
kp_out0	= port:PH18<4><1><default><default>
kp_out1	= port:PH19<4><1><default><default>
kp_out2	= port:PH22<4><1><default><default>
kp_out3	= port:PH23<4><1><default><default>
kp_out4	= port:PH24<4><1><default><default>
kp_out5	= port:PH25<4><1><default><default>
kp_out6	= port:PH26<4><1><default><default>
kp_out7	= port:PH27<4><1><default><default>

CONFIDENTIAL

27 USB 控制标志

27.1 [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；
usb_port_type=xx	USB 端口的使用模式： 0: device only 1: host only 2: OTG
usb_detect_type=xx	USB 端口的检查方式： 0: 无检查 1: vbus/id 检查
usb_id_gpio=xx	USB ID pin 脚配置
usb_det_vbus_gpio=xx	USB DET_VBUS pin 脚配置： 如果 GPIO 提供 pin，请参考 gpio 配置说明进行配置； 如果是 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 配置说明进行配置。
usb_restrict_gpio=xx	USB 限流控制 pin 脚： 参考 gpio 配置说明进行配置
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：

A20 sysconfig.fex 配置说明

Copyright © 2014Allwinner Technology. All Rights Reserved.

-- 67 --

```
[usbc0]
usb_used          = 1
usb_port_type     = 2
usb_detect_type   = 1
usb_id_gpio       = port:PH04<0><1><default><default>
usb_det_vbus_gpio = "axp_ctrl"
usb_drv_vbus_gpio = port:PB09<1><0><default><0>
usb_restrict_gpio = port:PH00<1><0><default><0>
usb_host_init_state = 0
usb_restric_flag   = 0
usb_restric_voltage = 3550000
usb_restric_capacity = 5
```

27.2 [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；
usb_port_type =xx	USB 端口的使用模式： 0: device only 1: host only 2: OTG
usb_detect_type=xx	USB 端口的检查方式： 0: 无检查 1: vbus/id 检查
usb_id_gpio=xx	USB ID pin 脚配置
usb_det_vbus_gpio=xx	USB DET_VBUS pin 脚配置： 如果 GPIO 提供 pin，请参考 gpio 配置说明进行配置； 如果是 AXP 提供 pin，则配置为: "axp_ctrl"。
usb_restrict_gpio=xx	USB DRY_VBUS pin 脚配置： 具体请参考 gpio 配置说明进行配置。
usb_drv_vbus_gpio=xx	USB 限流控制 pin 脚： 参考 gpio 配置说明进行配置
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
--	-------------------------

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       =
usb_host_init_state     = 1
usb_restric_flag        = 0
```

27.3 [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；
usb_port_type=xx	USB 端口的使用模式： 0: device only 1: host only 2: OTG
usb_detect_type=xx	USB 端口的检查方式： 0: 无检查 1: vbus/id 检查
usb_id_gpio=xx	USB ID pin 脚配置
usb_det_vbus_gpio=xx	USB DET_VBUS pin 脚配置： 如果 GPIO 提供 pin，请参考 gpio 配置说明进行配置； 如果是 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 配置说明进行配置。
usb_restrict_gpio=xx	USB 限流控制 pin 脚： 参考 gpio 配置说明进行配置
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

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       =
usb_host_init_state     = 0
usb_restric_flag        = 0
```

CONFIDENTIAL

28 USB Device

28.1 [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，字符串
product_name=xx	USB 产品名，字符串
serial_number=xx	USB 序列号，字符串

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"

```

28.2 [msc_feature]

配置项	配置项含义
vendor_name=xx	U 盘厂商名，字符串
product_name=xx	U 盘产品名，字符串
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 = 3

CONFIDENTIAL

29 重力感应(G-sensor)

注意：盒子未使用该模块。

29.1 [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 控制器选择, 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          =
```

29.2 [gsensor_list_para]

配置项	配置项含义
gsensor_det_used=xx	是否支持 gsensor 自动检测
bma250	是否支持此型号, 0: 不支持, 1: 支持
mma8452	是否支持此型号, 0: 不支持, 1: 支持
mma7660	是否支持此型号, 0: 不支持, 1: 支持
mma865x	是否支持此型号, 0: 不支持, 1: 支持
afa750	是否支持此型号, 0: 不支持, 1: 支持

lis3de_acc	是否支持此型号, 0: 不支持, 1: 支持
lis3dh_acc	是否支持此型号, 0: 不支持, 1: 支持
kxtik	是否支持此型号, 0: 不支持, 1: 支持
dmard10	是否支持此型号, 0: 不支持, 1: 支持
dmard06	是否支持此型号, 0: 不支持, 1: 支持
mx622x	是否支持此型号, 0: 不支持, 1: 支持
fxos8700	是否支持此型号, 0: 不支持, 1: 支持
lsm303d	是否支持此型号, 0: 不支持, 1: 支持

配置举例:

[gsensor_list_para]

```
gsensor_det_used          = 0
bma250                    = 1
mma8452                    = 1
mma7660                    = 1
mma865x                    = 1
afa750                     = 1
lis3de_acc                 = 1
lis3dh_acc                 = 1
kxtik                      = 1
dmard10                    = 0
dmard06                    = 1
mx622x                     = 1
fxos8700                   = 1
lsm303d                    = 1
```

30 GPS 配置

注意：盒子未使用该模块。

30.1 [gps_para]

配置项	配置项含义
gps_used=xx	是否支持 gps
gps_spi_id	GPS 模块使用的 spi 控制器编号
gps_spi_cs_num	使用的 spi 片选，0 或 1
gps_clk	gps clk 脚 GPIO 设置
gps_sign	gps sign 脚 GPIO 设置
gps_mag	gps msg 脚 GPIO 设置
gps_vcc_en	gps 电源使能脚 GPIO 设置
gps_osc_en	gps 晶振使能脚 GPIO 设置
gps_rx_en	gps 输出管脚配置

配置举例：

[gps_para]

gps_used = 0
gps_spi_id = 2
gps_spi_cs_num = 0
gps_clk = port:PI00<2><default><default><default>
gps_sign = port:PI01<2><default><default><default>
gps_mag = port:PI02<2><default><default><default>
gps_vcc_en = port:PC22<1><default><default><0>
gps_osc_en = port:PI14<1><default><default><0>
gps_rx_en = port:PI15<1><default><default><0>

31 WIFI

31.1 [wifi_para]

配置项	配置项含义
wifi_used	是否要使用 wifi
wifi_sdc_id	sdio wifi 选用的是哪个 sdc 作为接口
wifi_usbc_id	usb wifi 选用的是哪个 usb 作为接口
wifi_usbc_type	usb 接口类型，1 为 ehci，0 为 ohci
wifi_mod_sel	具体选择哪一款模组 0- none, 1- bcm40181, 2- bcm40183, 3 - rtl8723as(wifi+bt), 4- rtl8189es, 5 - rtl8192cu, 6 - rtl8188eu, 7 - ap6210,
wifi_power	给模组供电的 axp 引脚名

配置举例：

[wifi_para]

```
wifi_used          = 1
wifi_sdc_id        = 3
wifi_usbc_id       = 2
wifi_usbc_type     = 1
wifi_mod_sel       = 6
wifi_power         = ""
```

说明：

1. [wifi_para]下的配置项是 usb 和 sdio 接口 wifi 共用的；
2. 对于 USB 接口类型的 wifi 模块，不需要额外的 GPIO 进行电源配置，如 rtl8192cu、rtl8188eu、rtl8723au 等模块；
3. 对于 SDIO 接口类型的 wifi 模块，需要配置 GPIO，请打开以下对应的 GPIO 配置项（打开方法：去掉配置项前的分号）。

31.2 bcm40181 GPIO 配置

```
;bcm40181_shdn      = port:PH09<1><default><default><0>  
;bcm40181_host_wake = port:PH10<0><default><default><0>
```

31.3 bcm40183 GPIO 配置

```
;bcm40183_wl_region = port:PH09<1><default><default><0>  
;bcm40183_wl_host_wake = port:PH10<0><default><default><0>  
;bcm40183_bt_rst     = port:PB05<1><default><default><0>  
;bcm40183_bt_region  = port:PB05<1><default><default><0>  
;bcm40183_bt_wake    = port:PI20<1><default><default><0>  
;bcm40183_bt_host_wake = port:PI21<0><default><default><0>
```

31.4 rtl8723as GPIO 配置

```
rtk_rtl8723as_wl_dis = port:PH09<1><default><default><0>  
rtk_rtl8723as_bt_dis = port:PB05<1><default><default><0>  
rtk_rtl8723as_wl_host_wake = port:PH10<0><default><default><0>  
rtk_rtl8723as_bt_host_wake = port:PI21<0><default><default><0>
```

31.5 rtl8189es GPIO 配置

```
;rtl8189es_shdn      = port:PG10<1><default><default><0>  
;rtl8189es_wakeup    = port:PG12<1><default><default><1>
```

31.6 rtl8188eu GPIO 配置

```
rtk_rtl8188eu_wl_dis = port:PH03<1><default><default><0>
```

31.7 ap6210 GPIO 配置

```
ap6xxx_wl_region = port:PH09<1><default><default><0>  
ap6xxx_wl_host_wake = port:PH10<0><default><default><0>  
ap6xxx_bt_region  = port:PB05<1><default><default><0>
```

ap6xxx_bt_wake	= port:PI20<1><default><default><0>
ap6xxx_bt_host_wake	= port:PI21<0><default><default><0>

CONFIDENTIAL

32 3G

注意：盒子未使用该模块。

32.1 [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
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	电源配置
3g_wakeup	唤醒 GPIO 设置
3g_int	中断输入 GPIO 配置

配置举例:

[3g_para]

3g_used = 0

3g_usbc_num = 2

3g_uart_num = 0

3g_pwr =

3g_wakeup =

3g_int =

33 gyroscope

注意：盒子未使用该模块。

33.1 [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 = 0

gy_twi_id = 1

gy_twi_addr = 0x00

gy_int1 = port:PH18<6><1><default><default>

gy_int2 = port:PH19<6><1><default><default>

34 光感(light sensor)

注意：盒子未使用该模块。

34.1 [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 = 0

ls_twi_id = 1

ls_twi_addr = 0x00

ls_int = port:PH20<6><1><default><default>

35 罗盘 Compass

注意：盒子未使用该模块。

35.1 [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 = 0

compass_twi_id = 1

compass_twi_addr = 0x00

compass_int = port:PI13<6><1><default><default>

36 蓝牙(blutetooth)

36.1 [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>

37 数字音频总线（I2S）

37.1 [i2s_para]

配置项	配置项含义
i2s_used=xx	为 1 时加载该模块，为 0 是不加载
i2s_channel=xx	声道控制
i2s_mclk=xx	I2s MCLK 信号的 GPIO 配置
i2s_bclk=xx	I2s BCLK 信号的 GPIO 配置
i2s_lrclk=xx	I2s LRCK 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>

```

38 数字音频总线(pcm)

38.1 [pcm_para]

配置项	配置项含义
pcm_used=xx	为 1 时加载该模块，为 0 是不加载
pcm_channel=xx	声道控制
pcm_master	1: codec clk & FRM master 2: codec clk slave & FRM master 3: codec clk master & frame slave 4: codec clk & FRM slave
pcm_select	1: pcm 0: i2s
audio_format	1: standard i2s format 2: right justified format 3: left justified format 4: pcm. MSB is available on 2nd BCLK rising edge after LRC rising edge 5: pcm. MSB is available on 1nd BCLK rising edge after LRC rising edge
signal_inversion	1: normal bit clock + frame 2: normal BCLK + inv FRM 3: invert BCLK + nor FRM 4: invert BCLK + FRM
over_sample_rate	过采样设置: 128fs/192fs/256fs/384fs/512fs/768fs
sample_resolution	采样精度: 16bits/20bits/24bits
word_select_size	字宽度: 16bits/20bits/24bits/32bits
pcm_sync_period	同步周期: 16/32/64/128/256
msb_lsb_first	0: msb 1: lsb
sign_extend	0: zero pending;

	1: sign extend
slot_index	接口索引
slot_width	solt 位宽: 8/16
frame_width	0: long frame = 2 clock width; 1: short frame
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
pcm_mclk=xx	暂不使用
pcm_bclk=xx	pcmBCLK 信号的 GPIO 配置
pcm_lrclk=xx	pcmLRCK 信号的 GPIO 配置
pcm_dout	pcm out 的 GPIO 配置
pcm_din	pcmIN 信号的 GPIO 配置

配置举例:

[pcm_para]

```

pcm_used           = 0
pcm_channel        = 2
pcm_master         = 4
pcm_select         = 1
audio_format       = 4
signal_inversion   = 1
over_sample_rate   = 256
sample_resolution  = 16
word_select_size   = 32
pcm_sync_period    = 256
msb_lsb_first      = 0
sign_extend        = 0
slot_index         = 0
slot_width         = 16
frame_width        = 1
tx_data_mode       = 0

```

rx_data_mode	= 0
pcm_melk	= port:PA09<6><1><default><default>
pcm_bclk	= port:PA14<6><1><default><default>
pcm_lrcclk	= port:PA15<6><1><default><default>
pcm_dout0	= port:PA16<6><1><default><default>
pcm_dout1	=
pcm_dout2	=
pcm_dout3	=
pcm_din	= port:PA17<6><1><default><default>

CONFIDENTIAL

39 数字音频总线（SPDIF）

39.1 [spdif_para]

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

配置举例：

[spdif_para]

spdif_used = 1

spdif_dout = port:PB13<4><1><default><default>

spdif_din =

40 内置音频（codec）

40.1 [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_used=xx	是否使能功放芯片的静音控制
audio_pa_ctrl=xx	功放芯片的静音 gpio 口控制。

配置举例：

[audio_para]

audio_used = 1

audio_pa_ctrl_used = 0

audio_pa_ctrl = port:PB03<1><default><default><1>

41 红外(ir)

41.1 [ir_para]

配置项	配置项含义
ir_used=xx	是否支持 ir
ir_rx =xx	ir 的接收管脚 GPIO 配置
ir_wakeup	0: 有 MCU 1: 无 MCU
power_key	ir 电源键对应的键值
ir_addr_code	ir 遥控器的地址码

配置举例:

[ir_para]

ir_used = 1

ir_rx = port:PB04<2><default><default><default>

ir_wakeup = 0

ir_power_key_code = 0x57

ir_addr_code = 0x9f00

42 gpio 配置

42.1 [gpio_para]

配置项	配置项含义
gpio_used	是否使用 GPIO 默认初始化模块
gpio_num	需要初始化的 GPIO 的数量
gpio_pin_1	GPIO 管脚配置 (可以有多个, 名称不能一样)

配置举例:

[gpio_para]

gpio_used = 1

gpio_num = 1

gpio_pin_1 = port:PH20<1><default><default><1>

注意: gpio 初始化是在 uboot 阶段做的, 用途比如可以做 led 灯的初始化。

43 PMU 电源

43.1 [pmu_para]

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, axp152 和 axp209 地址不一样
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_init_chgcur	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_resume_chgcur=xx	设置唤醒充电电流, 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_freqc=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_csttime=xx	设置恒流充电超时时间, min, 360/480/600/720

power_start	充电完成后, 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 3.41V 对应的百分比, %
pmu_bat_para4=xx	设置空载电池电压为 3.55V 对应的百分比, %
pmu_bat_para5=xx	设置空载电池电压为 3.62V 对应的百分比, %
pmu_bat_para6=xx	设置空载电池电压为 3.66V 对应的百分比, %
pmu_bat_para7=xx	设置空载电池电压为 3.69V 对应的百分比, %
pmu_bat_para8=xx	设置空载电池电压为 3.73V 对应的百分比, %
pmu_bat_para9=xx	设置空载电池电压为 3.76V 对应的百分比, %
pmu_bat_para10=xx	设置空载电池电压为 3.80V 对应的百分比, %
pmu_bat_para11=xx	设置空载电池电压为 3.83V 对应的百分比, %
pmu_bat_para12=xx	设置空载电池电压为 3.87V 对应的百分比, %
pmu_bat_para13=xx	设置空载电池电压为 3.94V 对应的百分比, %
pmu_bat_para14=xx	设置空载电池电压为 4.01V 对应的百分比, %
pmu_bat_para15=xx	设置空载电池电压为 4.08V 对应的百分比, %
pmu_bat_para16=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_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_peekon_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_intotp_en=xx	设置过温保护功能是否开启, 0: 关闭, 1: 开启
pmu_used2=xx	设置是否使用第二个 PMU 芯片, 0: 不使用, 1: 使用

pmu_adpdet=xx	设置第二个 PMU 芯片的使能脚，GPIO 配置
pmu_init_chgcur2=xx	设置第二个 PMU 芯片的默认充电电流
pmu_earlysuspend_chgcur2=xx	设置第二个 PMU 芯片的 earlysuspend 阶段的充电电流
pmu_suspend_chgcur2	设置第二个 PMU 芯片的 suspend 阶段的充电电流
pmu_resume_chgcur2	设置第二个 PMU 芯片的 resume 阶段的充电电流
pmu_shutdown_chgcur2	设置第二个 PMU 芯片的使能脚
pmu_suspendpwroff_vol	设置休眠状态下，关机电压，mV
pmu_batdeten	设置电池检测功能，0：关闭，1：开启

配置举例：

[pmu_para]

```
pmu_used = 1
pmu_twi_addr = 0x34
pmu_twi_id = 0
pmu_irq_id = 32
pmu_battery_rdc = 100
pmu_battery_cap = 3200
pmu_init_chgcur = 300
pmu_earlysuspend_chgcur = 600
pmu_suspend_chgcur = 1000
pmu_resume_chgcur = 300
pmu_shutdown_chgcur = 1000
pmu_init_chgvol = 4200
pmu_init_chgend_rate = 15
pmu_init_chg_enabled = 1
pmu_init_adc_freq = 100
pmu_init_adc_freqc = 100
pmu_init_chg_pretime = 50
pmu_init_chg_csttime = 720
power_start = 1
```

```
pmu_bat_para1 = 0
pmu_bat_para2 = 0
pmu_bat_para3 = 0
pmu_bat_para4 = 0
pmu_bat_para5 = 5
pmu_bat_para6 = 8
pmu_bat_para7 = 11
```

pmu_bat_para8	= 22
pmu_bat_para9	= 33
pmu_bat_para10	= 43
pmu_bat_para11	= 50
pmu_bat_para12	= 59
pmu_bat_para13	= 71
pmu_bat_para14	= 83
pmu_bat_para15	= 92
pmu_bat_para16	= 100
pmu_usbvol_limit	= 1
pmu_usbcur_limit	= 0
pmu_usbvol	= 4000
pmu_usbcur	= 0
pmu_usbvol_pc	= 4400
pmu_usbcur_pc	= 0
pmu_pwroff_vol	= 3300
pmu_pwrnol_vol	= 2900
pmu_peekoff_time	= 6000
pmu_peekoff_en	= 1
pmu_peeklong_time	= 1500
pmu_peekon_time	= 1000
pmu_pwrrok_time	= 64
pmu_pwrnoe_time	= 2000
pmu_intotp_en	= 1
pmu_used2	= 0
pmu_adpdet	= port:PH02<0><default><default><default>
pmu_init_chgcur2	= 400
pmu_earlysuspend_chgcur2	= 600
pmu_suspend_chgcur2	= 1200
pmu_resume_chgcur2	= 400
pmu_shutdown_chgcur2	= 1200
pmu_suspendpwroff_vol	= 3500
pmu_batdeten	= 1

CONFIDENTIAL

44 dvfs 电压-频率配置表

注意:

dvfs 参数直接影响系统的稳定性, 请勿随意修改, 如有疑问必须咨询 FAE 进行确认。

用户唯一可以修改的参数是 min_freq, 即 CPU 最低的运行频率, 默认为 60MHz, 按应用场景需求可以适当改为 312Mhz。其他任何参数禁止修改。

44.1 [dvfs_table]

配置项	配置项含义
max_freq	CPU 最高运行频率, 单位 Hz
normal_freq	CPU 正常运行频率, 单位 Hz
min_freq	CPU 最低运行频率, 单位 Hz 默认为 60Mhz, 用户唯一可以修改的选项。
LV_count	dvfs 表项数量, 最大为 16
LV1_freq	dvfs 表第一项的 CPU 频率, 单位 Hz
LV1_volt	dvfs 表第一项的 CPU 频率对应的供电电压, 单位 mV
LV2_freq	dvfs 表第二项的 CPU 频率, 单位 Hz
LV2_volt	dvfs 表第二项的 CPU 频率对应的供电电压, 单位 mV
LV3_freq	dvfs 表第三项的 CPU 频率, 单位 Hz
LV3_volt	dvfs 表第三项的 CPU 频率对应的供电电压, 单位 mV
LV4_freq	dvfs 表第四项的 CPU 频率, 单位 Hz
LV4_volt	dvfs 表第四项的 CPU 频率对应的供电电压, 单位 mV
LV5_freq	dvfs 表第五项的 CPU 频率, 单位 Hz
LV5_volt	dvfs 表第五项的 CPU 频率对应的供电电压, 单位 mV
LV6_freq	dvfs 表第六项的 CPU 频率, 单位 Hz
LV6_volt	dvfs 表第六项的 CPU 频率对应的供电电压, 单位 mV
LV7_freq	dvfs 表第七项的 CPU 频率, 单位 Hz
LV7_volt	dvfs 表第七项的 CPU 频率对应的供电电压, 单位 mV
LV8_freq	dvfs 表第八项的 CPU 频率, 单位 Hz
LV8_volt	dvfs 表第八项的 CPU 频率对应的供电电压, 单位 mV

配置举例:

[dvfs_table]

max_freq = 1008000000

A20 sysconfig.fex 配置说明

- 97 -

Copyright © 2014Allwinner Technology. All Rights Reserved.

-- 97 --

normal_freq = 1008000000
min_freq = 60000000

LV_count = 8

LV1_freq = 1008000000
LV1_volt = 1450

LV2_freq = 912000000
LV2_volt = 1400

LV3_freq = 864000000
LV3_volt = 1300

LV4_freq = 792000000
LV4_volt = 1250

LV5_freq = 720000000
LV5_volt = 1200

LV6_freq = 624000000
LV6_volt = 1150

LV7_freq = 528000000
LV7_volt = 1100

LV8_freq = 312000000
LV8_volt = 1050

备注:

目前公版方案的 CPU 频率最高是 1008Mhz，超频有风险，详细请咨询 FAE。

45 recovery key 配置

45.1 [recovery_key]

配置项	配置项含义
key_min	盒子未使用
key_max	盒子未使用

配置举例：

[recovery_key]

key_min =4

key_max =6

45.2 [system]

配置项	配置项含义
recovery_key	一键恢复功能(按键连接到的 GPIO); 此功能需要在 sys_partition.fex 把 sysrecovery 分区打开。
anrecovery_key	一键 OTA 功能(按键连接到的 GPIO); 此功能支持从 U 盘或者其他 外设 sd 卡进行 OTA 升级, 升级包命名必须为 update.zip 且放在根目 录。

配置举例：

[system]

recovery_key = port:PH16<0><default>

;recovery_key = port:PH16<0><default>

注意：两个功能只能二选一，如示例所示，选择一键恢复，则将 anrecovery_key 配置使用“;”注释掉。

46 boot 阶段红外配置

46.1 [ir_boot_para]

配置项	配置项含义
ir_used	设置是否使能，0：关闭，1：使能
ir_mode	模式配置，1：一键进入 OTA，2：一键恢复，其他：无效
ir_rx	ir 输入引脚配置
ir_recovery_key	按键码配置，默认 power 键
ir_addr_code	遥控器地址码配置

配置举例：

[ir_boot_para]

ir_used = 0

ir_mode = 1

ir_rx = port:PB04<2><default><default><default>

ir_recovery_key = 0x57

ir_addr_code = 0x9f00

注意：红外和休眠唤醒的关系说明请参考文档《A20_红外遥控与休眠唤醒配置 v1.0》。

47 boot 阶段显示输出配置

47.1 [boot_disp]

配置项	配置项含义
output_type	输出接口: 0 -- none 1 -- lcd 2 -- tv(cvbs/ypbpr) 3 -- hdmi 4 -- vga
output_mode	输出模式（仅对 tv/hdmi 输出接口有效）: 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
auto_hpd	推荐配置为 1，自动检测 hdmi/cvbs/ypbpr 插入

配置举例：

例子 1：自动检测

[boot_disp]

output_type = 3

output_mode = 5

auto_hpd = 1

例子 2：非自动检测，指定

如果配置为非自动检测，则务必注意 type 和 mode 需要配对，如上表所示，红色对应的是 hdmi 配置，蓝色对应 cvbs/ypbpr 配置。

[boot_disp]

output_type = 2

output_mode = 11

auto_hpd = 0