



Tina Linux PMU 开发指南

版本号: 1.0
发布日期: 2021.11.25

版本历史

版本号	日期	制/修订人	内容描述
1.0	2021.11.25	AWA1611	1. 初始版本 2. 添加 AXP717 说明



目 录

1 前言	1
1.1 文档简介	1
1.2 目标读者	1
1.3 适用范围	1
2 模块介绍	2
2.1 模块功能介绍	2
2.2 相关术语介绍	2
2.3 模块配置介绍	3
2.3.1 Device Tree 配置说明	3
2.3.1.1 board.dts 配置说明	3
2.3.1.2 sys_config.fex 配置	15
2.3.2 kernel menuconfig 配置说明	18
2.4 源码结构介绍	22
2.5 模块框架介绍	23
3 模块使用范例	25
3.1 外部 sysfs 节点	25
3.1.1 Regulator	25
3.1.2 Virtual-consumer	26
3.1.3 Power_supply	27
3.2 Regulator 使用方法	29
3.2.1 内核代码调用 regulator 示例	29
3.2.2 Regulator shell 命令使用示例	29
3.2.3 usb_count 查看	30
4 FAQ	31
4.1 调试方法	31
4.1.1 调试工具	31
4.1.1.1 power key 调试方式	31
4.1.2 调试节点	31
4.1.2.1 /sys/kernel/debug/regulator/regulator_summary 节点	31
4.1.2.2 regmap registers 节点	33
4.1.2.3 axp_reg 节点	34
4.1.2.4 debug_mask 节点	34

插 图

2-1 pmu-control-config	18
2-2 regularot-config	19
2-3 charger-config	20
2-4 power-key-config	21
2-5 virtuaal-config	21
2-6 acin-config	22
2-7 AXP 框架图	23
2-8 AXP 软件框架图	23



1 前言

1.1 文档简介

介绍使用 Tina PMU 驱动的使用方法。

1.2 目标读者

适用于使用全志 AXP 平台 PMU 的开发人员。

1.3 适用范围

表 1-1: 适用 SOC 平台及 AXP 对应表

产品名称	内核版本	AXP 型号
R818	Linux-4.9	AXP717

2 模块介绍

2.1 模块功能介绍

PMU，负责系统各个模块供电、按键开关机、电池充放电管理。

2.2 相关术语介绍

表 2-1: 术语简介

术语	说明
PMU	电源管理单元，主要包括 regulator、power supply、gpio、power key 这四个子功能部分。
AXP	全志 PMU 平台的系列名称，如 AXP803、AXP717 等。
LDO	是 low dropout regulator，意为低压差线性稳压器。线性稳压器使用在其线性区域内运行的晶体管或 FET，从应用的输入电压中减去超额的电压，产生经过调节的输出电压。
DC-DC	是直流变直流，即不同直流电源值之间的转换，只要符合这个定义都可以叫 DC-DC 转换器，也包括 LDO。但是一般的说法是把直流变直流由开关方式实现的器件叫 DCDC。
regulator	Linux 内核对 LDO、DC-DC 的管理核心。
USB-Power-Supply	USB 接口对系统的供电。
ACIN-Power-Supply	适配器 ACIN 对系统的供电。
BAT-Power-Supply	电池 BAT 对系统的供电。
Power-Supply	Linux 内核对 USB、ACIN、BAT 供电的管理核心。
MFD	Multi Function Device，Linux 内核对多功能设备 PMU 的管理核心
regmap	Linux 内核用于管理片外模块寄存器的方法。

2.3 模块配置介绍

2.3.1 Device Tree 配置说明

在 Tina 系统中，有两种 dts 文件。一是用于保存芯片所有平台的模块配置 `${CHIP}.dtsi`，二是保存每一个板级平台的设备信息的 `board.dts`。两者的区别主要是：前者主要保存芯片相关的配置，不管芯片外围换什么硬件，芯片配置还是保持不变的。而后者是用于保存不同版型之间差异化的配置。

PMU 模块的 dts 配置是在 `board.dts` 中，`dtsi` 中无用户可用配置。

2.3.1.1 board.dts 配置说明

`board.dts` 路径为：`${ROOT_DIR}/device/config/chips/${PLATFORM}/configs/${TARGET}/board.dts`

说明

`${ROOT_DIR}`：是 *tina SDK* 根目录

`${PLATFORM}`：是芯片型号，如 *r818*

`${TARGET}`：是版级型号，如 *evb1*

技巧

`board.dts` 所在在版级配置目录，*tina SDK* 环境中，在 `source build/envsetup.sh` 后，可通过 “*cconfigs*” 命令直接跳转过去配置目录。

`board.dts` 就在版本配置目录的上一级，再通过 “*cd ..*” 即可到达 `board.dts` 所在的目录。

PMU 一共包含了 regulator, power supply, power key, PMU 在内核中的设备是多个设备同时存在，并存在层次对应关系。

PMU主设备(MFD)

```
|
+-----> regulator device
|
+-----> power key device
|
+-----> power supply device
|
+-----> wdt device
```

说明

AXP717 在内核中使用的是名为 *axp2202* 的软件框架，因此在 `dts` 文件中。型号采用软件框架的名字，而不配置为 *axp717*。同理，在 `sysconfig.fex` 和 `kernel menuconfig` 中也一样。

```
pmu0: pmu@34 {
    compatible = "x-powers,axp2202";
    reg = <0x34>;
    interrupts = <0 IRQ_TYPE_LEVEL_LOW>;
    interrupt-parent = <&nmi_intc>;
    x-powers,drive-vbus-en;
    pmu_reset = <0>;
```

```
pmu_irq_wakeup = <1>;
pmu_hot_shutdown = <1>;

wakeup-source;

//interrupt-controller;
//#interrupt-cells = <1>;

usb_power_supply: usb_power_supply {
    compatible = "x-powers,axp2202-usb-power-supply";

    pmu_usbpc_vol = <4600>;
    pmu_usbpc_cur = <500>;
    pmu_usbad_vol = <4000>;
    pmu_usbad_cur = <2500>;

    pmu_boost_vol = <5126>;
    pmu_bc12_en;
    pmu_cc_logic_en = <1>;
    /* pmu_boost_en; */

    pmu_usb_typec_used = <1>;
    wakeup_usb_in;
    wakeup_usb_out;

    status = "okay";
};

/* cvin */
gpio_power_supply: gpio_power_supply {
    compatible = "x-powers,gpio-supply";
    status = "disabled";

    wakeup_gpio;
};

bat_power_supply: bat-power-supply {
    compatible = "x-powers,axp2202-bat-power-supply";
    param = <&axp2202_parameter>;
    status = "okay";

    pmu_chg_ic_temp = <1>;

    pmu_battery_rdc = <147>;
    pmu_battery_cap = <1771>;
    pmu_runtime_chgcur = <1000>;
    pmu_suspend_chgcur = <1500>;
    pmu_shutdown_chgcur = <1500>;
    pmu_terminal_chgcur = <128>;
    pmu_init_chgvol = <4200>;
    pmu_battery_warning_level1 = <15>;
    pmu_battery_warning_level2 = <0>;
    pmu_chgled_func = <0>;
    pmu_chgled_type = <0>;
    pmu_bat_para1 = <0>;
    pmu_bat_para2 = <0>;
    pmu_bat_para3 = <0>;
    pmu_bat_para4 = <0>;
    pmu_bat_para5 = <0>;
    pmu_bat_para6 = <0>;
    pmu_bat_para7 = <2>;
```



```
pmu_bat_para8 = <3>;
pmu_bat_para9 = <4>;
pmu_bat_para10 = <6>;
pmu_bat_para11 = <9>;
pmu_bat_para12 = <14>;
pmu_bat_para13 = <26>;
pmu_bat_para14 = <38>;
pmu_bat_para15 = <49>;
pmu_bat_para16 = <52>;
pmu_bat_para17 = <56>;
pmu_bat_para18 = <60>;
pmu_bat_para19 = <64>;
pmu_bat_para20 = <70>;
pmu_bat_para21 = <77>;
pmu_bat_para22 = <83>;
pmu_bat_para23 = <87>;
pmu_bat_para24 = <90>;
pmu_bat_para25 = <95>;
pmu_bat_para26 = <99>;
pmu_bat_para27 = <99>;
pmu_bat_para28 = <100>;
pmu_bat_para29 = <100>;
pmu_bat_para30 = <100>;
pmu_bat_para31 = <100>;
pmu_bat_para32 = <100>;
pmu_bat_temp_enable = <0>;
pmu_bat_charge_ltf = <1105>;
pmu_bat_charge_hrf = <121>;
pmu_bat_shutdown_ltf = <1381>;
pmu_bat_shutdown_hrf = <89>;
pmu_bat_temp_para1 = <2814>;
pmu_bat_temp_para2 = <2202>;
pmu_bat_temp_para3 = <1737>;
pmu_bat_temp_para4 = <1381>;
pmu_bat_temp_para5 = <1105>;
pmu_bat_temp_para6 = <890>;
pmu_bat_temp_para7 = <722>;
pmu_bat_temp_para8 = <484>;
pmu_bat_temp_para9 = <332>;
pmu_bat_temp_para10 = <233>;
pmu_bat_temp_para11 = <196>;
pmu_bat_temp_para12 = <166>;
pmu_bat_temp_para13 = <141>;
pmu_bat_temp_para14 = <121>;
pmu_bat_temp_para15 = <89>;
pmu_bat_temp_para16 = <66>;

wakeup_bat_out;
/* wakeup_bat_in; */
/* wakeup_bat_charging; */
/* wakeup_bat_charge_over; */
/* wakeup_low_warning1; */
/* wakeup_low_warning2; */
/* wakeup_bat_unttemp_work; */
/* wakeup_bat_ovtemp_work; */
/* wakeup_bat_unttemp_chg; */
/* wakeup_bat_ovtemp_chg; */

};

powerkey0: powerkey@0 {
```

```
compatible = "x-powers,axp2101-pek";
pmu_powkey_off_time = <6000>;
pmu_powkey_off_func = <0>;
pmu_powkey_off_en = <1>;
pmu_powkey_long_time = <1500>;
pmu_powkey_on_time = <512>;
wakeup_rising;
wakeup_falling;
status = "okay";
};

regulator0: regulators@0 {
    reg_dcdc1: dcdc1 {
        regulator-name = "axp2202-dcdc1";
        regulator-min-microvolt = <500000>;
        regulator-max-microvolt = <1540000>;
        regulator-ramp-delay = <2500>;
        regulator-enable-ramp-delay = <1000>;
        regulator-boot-on;
        regulator-always-on;
    };
    reg_dcdc2: dcdc2 {
        regulator-name = "axp2202-dcdc2";
        regulator-min-microvolt = <500000>;
        regulator-max-microvolt = <3400000>;
        regulator-ramp-delay = <2500>;
        regulator-enable-ramp-delay = <1000>;
        regulator-boot-on;
        regulator-always-on;
    };
    reg_dcdc3: dcdc3 {
        regulator-name = "axp2202-dcdc3";
        regulator-min-microvolt = <500000>;
        regulator-max-microvolt = <1840000>;
        regulator-ramp-delay = <2500>;
        regulator-enable-ramp-delay = <1000>;
        regulator-always-on;
    };
    reg_dcdc4: dcdc4 {
        regulator-name = "axp2202-dcdc4";
        regulator-min-microvolt = <1000000>;
        regulator-max-microvolt = <3700000>;
        regulator-ramp-delay = <2500>;
        regulator-enable-ramp-delay = <1000>;
    };
    reg_rtclldo: rtclldo {
        /* RTC_LD0 is a fixed, always-on regulator */
        regulator-name = "axp2202-rtclldo";
        regulator-min-microvolt = <1800000>;
        regulator-max-microvolt = <1800000>;
        regulator-boot-on;
        regulator-always-on;
    };
    reg_aldo1: aldo1 {
        regulator-name = "axp2202-aldo1";
        regulator-min-microvolt = <500000>;
        regulator-max-microvolt = <3500000>;
        regulator-enable-ramp-delay = <1000>;
    };
    reg_aldo2: aldo2 {
```

```
        regulator-name = "axp2202-aldo2";
        regulator-min-microvolt = <500000>;
        regulator-max-microvolt = <3500000>;
        regulator-enable-ramp-delay = <1000>;
    };
    reg_aldo3: aldo3 {
        regulator-name = "axp2202-aldo3";
        regulator-min-microvolt = <500000>;
        regulator-max-microvolt = <3500000>;
        regulator-enable-ramp-delay = <1000>;
        regulator-always-on;
        regulator-boot-on;
    };
    reg_aldo4: aldo4 {
        regulator-name = "axp2202-aldo4";
        regulator-min-microvolt = <500000>;
        regulator-max-microvolt = <3500000>;
        regulator-enable-ramp-delay = <1000>;
        regulator-always-on;
        regulator-boot-on;
    };
    reg_bldo1: bldo1 {
        regulator-name = "axp2202-bldo1";
        regulator-min-microvolt = <500000>;
        regulator-max-microvolt = <3500000>;
        regulator-enable-ramp-delay = <1000>;
    };
    reg_bldo2: bldo2 {
        regulator-name = "axp2202-bldo2";
        regulator-min-microvolt = <500000>;
        regulator-max-microvolt = <3500000>;
        regulator-enable-ramp-delay = <1000>;
        regulator-boot-on;
        regulator-always-on;
    };
    reg_bldo3: bldo3 {
        regulator-name = "axp2202-bldo3";
        regulator-min-microvolt = <500000>;
        regulator-max-microvolt = <3500000>;
        regulator-enable-ramp-delay = <1000>;
    };
    reg_bldo4: bldo4 {
        regulator-name = "axp2202-bldo4";
        regulator-min-microvolt = <500000>;
        regulator-max-microvolt = <3500000>;
        regulator-enable-ramp-delay = <1000>;
    };
    reg_cldo1: cldo1 {
        regulator-name = "axp2202-cldo1";
        regulator-min-microvolt = <500000>;
        regulator-max-microvolt = <3500000>;
        regulator-enable-ramp-delay = <1000>;
    };
    reg_cldo2: cldo2 {
        regulator-name = "axp2202-cldo2";
        regulator-min-microvolt = <500000>;
        regulator-max-microvolt = <3500000>;
        regulator-enable-ramp-delay = <1000>;
    };
    reg_cldo3: cldo3 {
```

```
        regulator-name = "axp2202-cldo3";
        regulator-min-microvolt = <500000>;
        regulator-max-microvolt = <3500000>;
        regulator-ramp-delay = <2500>;
        regulator-enable-ramp-delay = <1000>;
        regulator-boot-on;
    };
    reg_cldo4: cldo4 {
        regulator-name = "axp2202-cldo4";
        regulator-min-microvolt = <500000>;
        regulator-max-microvolt = <3500000>;
        regulator-enable-ramp-delay = <1000>;
    };
    reg_cpusldo: cpusldo {
        /* cpus */
        regulator-name = "axp2202-cpusldo";
        regulator-min-microvolt = <500000>;
        regulator-max-microvolt = <1400000>;
        regulator-boot-on;
        regulator-always-on;
    };
    reg_drivevbus: drivevbus {
        regulator-name = "axp2202-drivevbus";
        regulator-enable-ramp-delay = <1000>;
    };
};

virtual-dcdc1 {
    compatible = "xpower-vregulator,dcdc1";
    dcdc1-supply = <&reg_dcdc1>;
};
virtual-dcdc2 {
    compatible = "xpower-vregulator,dcdc2";
    dcdc2-supply = <&reg_dcdc2>;
};
virtual-dcdc3 {
    compatible = "xpower-vregulator,dcdc3";
    dcdc3-supply = <&reg_dcdc3>;
};
virtual-dcdc4 {
    compatible = "xpower-vregulator,dcdc4";
    dcdc4-supply = <&reg_dcdc4>;
};
virtual-rtcldo {
    compatible = "xpower-vregulator,rtcldo";
    rtcldo-supply = <&reg_rtcldo>;
};
virtual-aldol {
    compatible = "xpower-vregulator,aldol";
    aldol-supply = <&reg_aldol>;
};
virtual-aldol2 {
    compatible = "xpower-vregulator,aldol2";
    aldol2-supply = <&reg_aldol2>;
};
virtual-aldol3 {
    compatible = "xpower-vregulator,aldol3";
    aldol3-supply = <&reg_aldol3>;
};
};
```

```

virtual-ald04 {
    compatible = "xpower-vregulator,ald04";
    ald04-supply = <&reg_ald04>;
};
virtual-bl0d01 {
    compatible = "xpower-vregulator,bl0d01";
    bl0d01-supply = <&reg_bl0d01>;
};
virtual-bl0d02 {
    compatible = "xpower-vregulator,bl0d02";
    bl0d02-supply = <&reg_bl0d02>;
};
virtual-bl0d03 {
    compatible = "xpower-vregulator,bl0d03";
    bl0d03-supply = <&reg_bl0d03>;
};
virtual-bl0d04 {
    compatible = "xpower-vregulator,bl0d04";
    bl0d04-supply = <&reg_bl0d04>;
};
virtual-cl0d01 {
    compatible = "xpower-vregulator,cl0d01";
    cl0d01-supply = <&reg_cl0d01>;
};
virtual-cl0d02 {
    compatible = "xpower-vregulator,cl0d02";
    cl0d02-supply = <&reg_cl0d02>;
};
virtual-cl0d03 {
    compatible = "xpower-vregulator,cl0d03";
    cl0d03-supply = <&reg_cl0d03>;
};
virtual-cl0d04 {
    compatible = "xpower-vregulator,cl0d04";
    cl0d04-supply = <&reg_cl0d04>;
};
virtual-cpusl0d0 {
    compatible = "xpower-vregulator,cpusl0d0";
    cpusl0d0-supply = <&reg_cpusl0d0>;
};
virtual-drivevbus {
    compatible = "xpower-vregulator,drivevbus";
    drivevbus-supply = <&reg_drivevbus>;
};
axp_gpio0: axp_gpio@0 {
    gpio-controller;
    #size-cells = <0>;
    #gpio-cells = <6>;
    status = "okay";
};
};
/{

axp2202_parameter:axp2202-parameter {
    select = "battery-model";
    battery-model {
        parameter = /bits/ 8 <0x01 0xf5 0x40 0x00 0x1b 0x1e 0x28 0x0f
        0x0c 0x1e 0x32 0x02 0x14 0x05 0x0a 0x04
        0x74 0xfb 0xc8 0x0d 0x43 0x10 0x36 0xfb
        0x46 0x01 0xea 0x0d 0x2a 0x06 0x36 0x05
        0xf4 0x0a 0xb5 0x0f 0x42 0x0e 0xe6 0x09

```

```

0x9a 0x0e 0x42 0x0e 0x3b 0x04 0x2d 0x04
0x23 0x09 0x18 0x0e 0x09 0x0e 0x04 0x08
0xf7 0x0d 0xda 0x0d 0xd0 0x03 0xbb 0x03
0x9d 0x08 0x7f 0x0d 0x6a 0x0d 0x55 0x07
0xc2 0x57 0x2b 0x27 0x1e 0x0d 0x14 0x08
0xc5 0x98 0x7e 0x66 0x4e 0x44 0x38 0x1a
0x12 0x0a 0xf6 0x00 0x00 0xf6 0x00 0xf6
0x00 0xfb 0x00 0x00 0xfb 0x00 0x00 0xfb
0x00 0x00 0xf6 0x00 0x00 0xf6 0x00 0xf6
0x00 0xfb 0x00 0x00 0xfb 0x00 0x00 0xfb
0x00 0x00 0xf6 0x00 0x00 0xf6 0x00 0xf6>;
};
};
};

```

说明

PMU 为 I2C 设备，PMU 设备配置需要写在 i2c 节点内。

• PMU 属性配置

```

reg <u32>
    i2c寄存器地址

interrupts <args>
    中断配置，参考内核中断配置文档

interrupt-parent <phandler>
    上级中断控制器结点

wakeup-source <bool>
    是否作为唤醒源
    0:disable
    1:enable

x-powers,drive-vbus-en <bool>
    set N_VBUSEN pin as an output pin to control an external regulator to drive VBus

pmu_reset <bool>
    when power key press longer than 16s, PMU reset or not.
    0: not reset
    1: reset

pmu_irq_wakeup
    press irq wakeup or not when sleep or power down.
    0: not wakeup
    1: wakeup

pmu_hot_shutdown
    when PMU over temperature protect or not.
    0: disable
    1: enable

```

• power_supply 配置

power supply 属性配置，包括 usb-power-supply、gpio-power-supply 和 battery-power-supply。

对于 usb-power-supply 属性配置如下:

```
pmu_usbpc_vol <u32>
    usb pc输入电压限制值, 单位为mV

pmu_usbpc_cur <u32>
    usb pc输入电流限制值, 单位为mA

pmu_usbad_vol <u32>
    usb adaptor输入电压限制值(vimdpm), 单位为mV

pmu_usbad_cur <u32>
    usb adaptor输入电流限制值, 单位为mA

pmu_boost_vol <u32>
    打开boost给usb口供电的电压值, 单位为mV

pmu_bc12_en <bool>
    是否打开BC1.2协议功能

pmu_cc_logic_en <bool>
    是否打开cc协议功能

pmu_boost_en <bool>
    是否在初始化时打开boost功能

pmu_usb_typec_used <bool>
    usb接口为type-c

wakeup_usb_in <bool>
    usb插入唤醒使能

wakeup_usb_out <bool>
    usb拔出唤醒使能
```

📖 说明

在使用 **type-c** 时, 还需将 **usb** 驱动中的 “**usb_detect_type**” 节点同步配置成 **2**, 才能使用 **typec** 识别 **usb** 设备的功能。

对于 gpio-power-supply: 是控制 ac in 的配置, 在 AXP717 上默认不使用。

对于 battery-power-supply 属性配置如下:

```
param <string>
    电池参数, 与axp2202_parameter对应

pmu_chg_ic_temp <u32>
    1: TS current source always on
    0: TS current source off

pmu_battery_rdc <u32>
    电池内阻, 单位为mΩ

pmu_battery_cap <u32>
    电池容量, 单位为mAh

pmu_runtime_chgcur <u32>
    运行时constant充电电流限制, 单位为mA
```

pmu_suspend_chgcur <u32>
休眠时constant充电电流限制，单位为mA

pmu_shutdown_chgcur <u32>
关机时constant充电电流限制，单位为mA

pmu_terminal_chgcur <u32>
截止电流，停止充电的标志位之一，单位为mA

pmu_init_chgvol <u32>
电池满充电压，单位为mV

pmu_battery_warning_level1 <u32>
5-20 5% - 20% warning level1
电池低电量警告，当芯片检测电池电量从高到低跌到了设置的level1的值，capacity < warning_level1，就会触发warning_level1中断，从而再进行对应操作。
如果capacity ≥ warning_level1则会清掉该中断。
如AXP717，当触发warning1中断时，默认是不发生操作，需要修改对应代码进行定制化操作。

pmu_battery_warning_level2 <u32>
0-15 0% - 15% warning level2
意义同level1，当电池电量从从高低跌落，capacity < warning_level2，就会触发warning_level2中断。
当电池电量 warning_level2 ≤ capacity < warning_level1 则会清掉 level2中断。②

pmu_chgled_func <u32>;
CHGLED pin control
0: controlled by pmu
1: controlled by Charger

pmu_chgled_type <u32>
CHGLED Type select when pmu_chgled_func is 0
0: display with type A function
1: display with type B function
3: output controlled by the register of chgled_out_ctrl

pmu_bat_para1 <u32>
pmu_bat_para2 <u32>
...
pmu_bat_para32 <u32>
电池曲线参数
电池参数根据使用的电池不同，通过仪器测量出来

pmu_bat_temp_enable <u32>
设置电池温度检测、ntc是否使能

pmu_bat_charge_ltf <u32>
触发电池低温停充的TS pin电压阈值，单位：mV
默认：1105mV
范围：0-8160mV

pmu_bat_charge_hrf <u32>
触发电池高温停充的TS pin电压阈值，单位：mV
默认：121mV
范围：0-510mV

pmu_bat_shutdown_ltf <u32>
非充电模式下，触发电池低温中断的TS pin电压阈值，单位：mV
默认：1381mV

pmu_bat_shutdown_hrf <u32>

默认：89mV
范围：0-510mV

pmu_bat_temp_para1 <u32>
电池包-25度对应的TS pin电压，单位：mV

pmu_bat_temp_para2 <u32>
电池包-15度对应的TS pin电压，单位：mV

pmu_bat_temp_para3 <u32>
电池包-10度对应的TS pin电压，单位：mV

pmu_bat_temp_para4 <u32>
电池包-5度对应的TS pin电压，单位：mV

pmu_bat_temp_para5 <u32>
电池包0度对应的TS pin电压，单位：mV

pmu_bat_temp_para6 <u32>
电池包5度对应的TS pin电压，单位：mV

pmu_bat_temp_para7 <u32>
电池包10度对应的TS pin电压，单位：mV

pmu_bat_temp_para8 <u32>
电池包20度对应的TS pin电压，单位：mV

pmu_bat_temp_para9 <u32>
电池包30度对应的TS pin电压，单位：mV

pmu_bat_temp_para10 <u32>
电池包40度对应的TS pin电压，单位：mV

pmu_bat_temp_para11 <u32>
电池包45度对应的TS pin电压，单位：mV

pmu_bat_temp_para12 <u32>
电池包50度对应的TS pin电压，单位：mV

pmu_bat_temp_para13 <u32>
电池包55度对应的TS pin电压，单位：mV

pmu_bat_temp_para14 <u32>
电池包60度对应的TS pin电压，单位：mV

pmu_bat_temp_para15 <u32>
电池包70度对应的TS pin电压，单位：mV

pmu_bat_temp_para16 <u32>
电池包80度对应的TS pin电压，单位：mV
不同电池包的温敏电阻特性不一样，根据电池包的TS温敏电阻手册，找到pmu_bat_temp_para[1-16]对应温度点的电阻阻值，将阻值除以20得到的电压数值（单位：mV），将电压数值填进pmu_bat_temp_para[1-16]的节点中即可

wakeup_bat_out <bool>
电池拔出唤醒使能

wakeup_bat_charging <bool>
电池充电唤醒使能

```
wakeup_bat_charge_over <bool>
    电池充电结束唤醒使能

wakeup_low_warning1 <bool>
    电池低电量告警唤醒使能

wakeup_low_warning2 <bool>
    电池低电量告警2唤醒使能

wakeup_bat_untemp_chg <bool>
    电池低温充电唤醒使能

wakeup_bat_ovtemp_chg <bool>
    电池超温充电唤醒使能

wakeup_bat_untemp_work <bool>
    电池低温工作唤醒使能

wakeup_bat_ovtemp_work <bool>
    电池高温工作唤醒使能
```

- power key 属性配置

power key 设备为按键设备，具体的说为电源按键设备。power key 属性配置：

```
pmu_powkey_off_time <u32>
    控制按下多长时间响应poweroff事件
    可选的值为：
    4000 4s
    6000 6s
    8000 8s
    10000 10s

pmu_powkey_off_func <u32>
    控制power_off事件功能, 如果不配置，默认为power-off
    0: power_off
    1: 复位系统

pmu_powkey_off_en <bool>
    控制按键关机使能
    1: PWRON > OFFLEVEL AS poweroff source enable
    0: PWRON > OFFLEVEL as poweroff source disable

pmu_powkey_long_time <u32>
    控制ponlevel 寄存器0x27[5:4]
    1000 1s
    1500 1.5s
    2000 2s
    2500 2.5s

pmu_powkey_on_time <u32>
    控制按钮按下多长时间开机
    128 0.128s
    512 0.512s
    1000 1s
    2000 2s

wakeup_rising <bool>
```

控制是否弹起按钮唤醒系统

wakeup_falling <bool>

控制是否按下按钮唤醒系统

- regulator 属性配置

regulator 为系统 regulator_dev 设备，每个 regulator_dev 代表一路电源，设备通过对 regulator_dev 的引用建立 regulator，用来实现对电源的电压设置等功能。regulator 属性配置，参考内核原生 regulator 使用文档：Documentation/devicetree/bindings/regulator/regulator.txt。

regulator 配置如下：

```
reg\_ald01: ald01{
    regulator-name = "axp2101-dcdc1";
        为电源设备的名称

    regulator-min-microvolt = <1500000>;
        电源的最小值，单位：uV

    regulator-max-microvolt = <3400000>;
        电源的最大值，单位：uV

    regulator-ramp-delay = <2500>;
        电源的调压延时，单位：us

    regulator-enable-ramp-delay = <1000>;
        电源从关闭到开启的使能延时，单位：us

    regulator-boot-on;
        电源从启动时开启，在内核启动时，就按照dts配置加载了该路regulator。同时，进入系统后通过软件读取也能获知该路regulator被加载了。

    regulator-always-on;
        电源保持常开，不会由于调用regulator等API接口而关闭。
};
```

2.3.1.2 sys_config.fex 配置

在 sysconfig 中定义了 PMU 的 regulator 输出信息及板型 PMU 类型，在 boot0 和 uboot 会通过解析这部分属性来执行调压等操作。

```
-----
;[target] system bootup configuration
;boot_clock      = CPU boot frequency, Unit: MHz
;storage_type    = boot medium, 0-nand, 1-card0, 2-card2, -1(default)auto scan
;advert_enable   = 0-close advert logo    1-open advert logo    (只有多核启动下有效)
;power_mode      = axp_type,    0:axp81X, 1:dummy, 2:axp806, 3:axp2202, 4:axp858
;-----
[target]
boot_clock      = 1008
```

```

storage_type      = -1
advert_enable     = 0
burn_key          = 1
dragonboard_test  = 0
power_mode        = 3

;-----
;  system configuration
;  ?
; dcdc1_vol          ---set dcdc1 voltage,mV
;    ,500-1200,10mV/step
;
;    1220-3400,20mV/step
; dcdc2_vol          ---set dcdc2 voltage,mV
;    ,500-1200,10mV/step
;
;    1220-1540,20mV/step
; aldo1_vol          ---set aldo1 voltage,mV
;    ,500-3500,100mV/step
; dldo1_vol          ---set dldo1 voltage,mV
;    ,500-3500,100mV/step
;-----
[power_sply]
dcdc3_vol          = 1001200
aldo3_vol          = 1003300
aldo4_vol          = 1001800
blldo2_vol         = 1002500
clldo1_vol         = 1001800
clldo3_vol         = 1003300
cpusldo_vol        = 100900
dcdc1_mode         = 1
dcdc2_mode         = 1
;-----
; gpio_bias
; set gpio group withstand voltage
; pc_bias = 1800 is emmc
; pc_bias = 3300 is nand
;-----
[gpio_bias]
device_type        = "gpio_bias"
pl_bias            = 3300
pl_supply          = "aldo3_vol"
pc_bias            = 1800
pc_supply          = "clldo1_vol"
;pc_bias           = 3300
;pc_supply         = "clldo3_vol"

[power_delay]
device_type        = "power_delay"
aldo3_vol_delay    = 20000

```

下面将按照不同模块来解析 sys_config 中各个模块的配置含义。

- target 属性配置：

在此配置下，与 PMU 相关的主要是 power_mode。power mode 这个节点就是为了告诉平台

当前使用的是哪个 PMU 的哪种方案（有时在同一 SOC 平台同一 PMU 也可能出现电源树配置不一样的情况），在 boot0 阶段就会解析出该属性并调用调压接口进行调压。

power mode <u32>

属性决定了当前SOC板型使用哪个PMU，需boot0代码支持解析，目前仅R818/MR813方案支持解析该节点。后续别的平台需支持时应根据boot0代码更新sysconfig中的版型选择说明。

0: axp81X,
1: dummy
2: axp806
3: axp2202
4: axp858

说明

AXP717 与其他 **PMU** 使用同一套代码框架，使用 **AXP717** 则选上3: **axp2202**即可。

• power_sply 属性配置

xxxx_vol <u32>

uboot阶段xxxx这路电是否开关及输出电压配置，其中xxxx为供电输出名。属性由前缀(100/000)和后缀组成。未配置的电在uboot阶段不会进行开关电和调压操作。

前缀: 100, 这路电在uboot阶段打开
前缀: 110, 这路电在uboot阶段打开，但是烧写时会关闭
前缀: 000, 这路电在uboot阶段关闭
后缀: 3300, 这路电输出电压设置为3300 mV

dcdcx_mode <u32>

uboot阶段强制将dcdcx设置为fpwm开关模式，提高这路抗负载扰动能力。该属性不配置默认为0。目前仅AXP806/AXP305/AXP81X/AXP803/AXP2202/AXP717支持该功能。

0: pfm-pwm模式自由切换
1: 强制pwm模式

battery_exist <u32>

强制电池存在状态，uboot阶段根据该属性决定是否做电池状态的相关判断。如果该属性不进行配置，默认为1。适用于部分无电持方案或factory_mode的无电池场景的调试

0: 强制认为无电池存在，uboot阶段不做电池相关状态判断
1: 认为电池存在，uboot阶段正常进行电池状态的相关判断

charge_mode <u32>

配置充电页面，根据该属性决定是否进入关机充电页面。

如果该属性不进行配置，默认为1。

适用于不需要充电页面，或者适配器唤醒直接开机的需求。

0: 不进入充电页面，适配器唤醒直接进入开机流程
1: 适配器唤醒进入充电流程

• power_delay 属性配置

xxxx_vol_delay <u32>

uboot阶段xxxx这路电调压后的延时时间，单位us。用于部分调压后需等电压稳定才能操作的模块。

如: twi, 在上电时twi的电压不是3.3V，在uboot阶段需要将其升至3.3V，在电压抬升阶段是不能允许twi进行通信，不然就会产生错误。因此需要在调整电压的时候停止twi的功能。

该属性需与**power_sply**中的xxx_vol属性一块使用，当输出需要开关或调压时才需要进行延时。

• gpio_bias 属性配置

```
xx_bias <u32>
```

GPIOx口的耐压值设置，单位：mV。用于调整GPIOx口的耐压值，使其与GPIOx模块挂载的电压匹配，避免I/O口损坏，提升信号质量。

```
xx_supply <char>
```

GPIOx模块挂载的输出电压名，名字格式需与**power_sply**中的xxx_vol一致。该属性如果配上，在GPIOx挂载的输出改编后，会将对应的GPIOx bias耐压值修改过来。

2.3.2 kernel menuconfig 配置说明

在 Tina SDK 根目录运行 `make kernel_menuconfig`，进行内核配置修改，进入配置界面按以下步骤进行修改。

AXP717 与 AXP2101 公用同一份控制器、按键以及 regularotr 的代码，因此在 `kernel_menuconfig` 里选的是 AXP2101 的选项。

- PMU 控制器

```
-> Device Drivers
    -> Multifunction device drivers
        <*> X-Powers AXP2101 PMICs with I2C
```

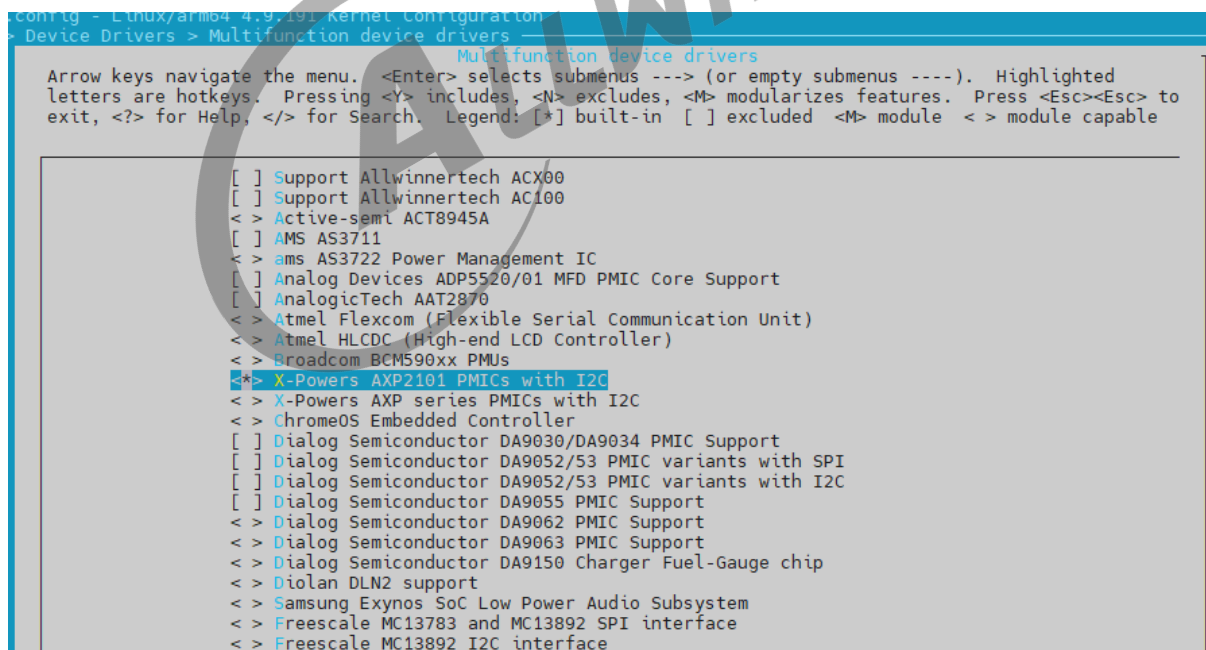


图 2-1: pmu-control-config

- regulator

```
-> Device Drivers
    -> Voltage and Current Regulator Support
        <*> X-POWERS AXP2101 PMIC Regulators
```

```

Voltage and Current Regulator Support
Arrow keys navigate the menu. <Enter> selects submenus ---> (or empty submenus ----). Highlighted
letters are hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to
exit, <?> for Help, </> for Search. Legend: [*] built-in [ ] excluded <M> module < > module capable

--- Voltage and Current Regulator Support
[ ] Regulator debug support
<*> Fixed voltage regulator support
<*> Virtual regulator consumer support
< > Userspace regulator consumer support
< > Active-semi act8865 voltage regulator
< > Analog Devices AD5398/AD5821 regulators
< > Freescale i.MX on-chip ANATOP LDO regulators
<*> X-POWERS AXP2101 PMIC Regulators
< > Dialog Semiconductor DA9210 regulator
< > Dialog Semiconductor DA9211/DA9212/DA9213/DA9214/DA9215 regulator
< > Fairchild FAN53555 Regulator
< > GPIO regulator support
< > Intersil ISL9305 regulator
< > Intersil ISL6271A Power regulator
< > National Semiconductors LP3971 PMIC regulator driver
< > National Semiconductors LP3972 PMIC regulator driver
< > TI/National Semiconductor LP8720/LP8725 voltage regulators
< > TI LP8755 High Performance PMU driver
< > LTC3589 8-output voltage regulator

```

图 2-2: regulatorot-config

- charger

```
-> Device Drivers
    -> Power supply class support
        <*> AXP2202 power supply driver
```

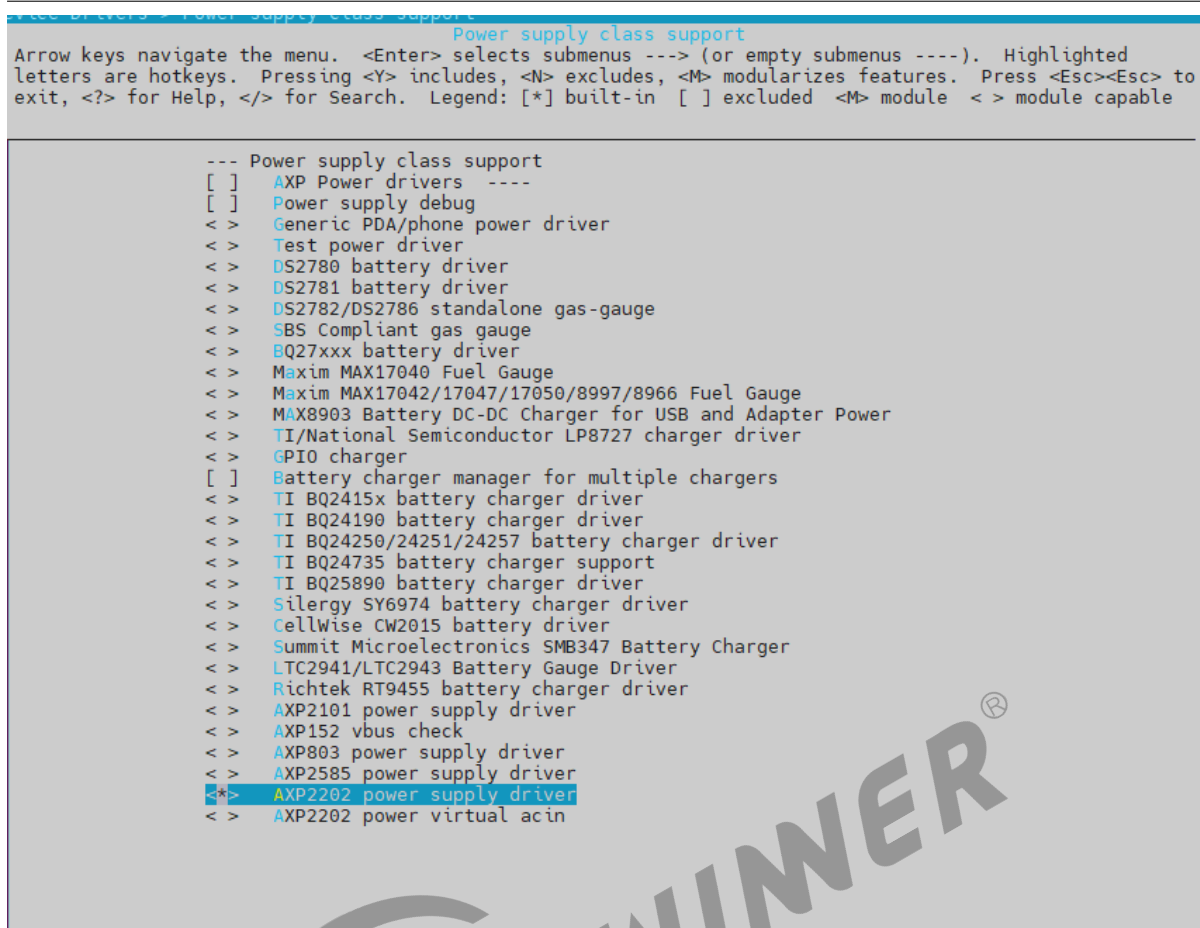


图 2-3: charger-config

- power key

```

-> Device Drivers
  -> Input device support
    -> Miscellaneous devices
      <*> X-Powers AXP2101 power button driver
  
```



```

Miscellaneous devices
Arrow keys navigate the menu. <Enter> selects submenus ---> (or empty submenus ----). Highlighted
letters are hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to
exit, <?> for Help, </> for Search. Legend: [*] built-in [ ] excluded <M> module < > module capable

--- Miscellaneous devices
<> Analog Devices AD714x Capacitance Touch Sensor
<> Atmel Capacitive Touch Button Driver
<> BMA150/SMB380 acceleration sensor support
<> NI Ettus Research USRP E3xx Button support.
<> MMA8450 - Freescale's 3-Axis, 8/12-bit Digital Accelerometer
<> MPU3050 Triaxial gyroscope sensor
<> Sharp GP2AP002A00F I2C Proximity/Opto sensor driver
<> Generic GPIO Beeper support
<> Polled GPIO tilt switch
<> Polled GPIO Decoder Input driver
<> ATI / Philips USB RF remote control
<> Keyspan DMR USB remote control
<> Kionix KXTJ9 tri-axis digital accelerometer
<> Griffin PowerMate and Contour Jog support
<> Yealink usb-p1k voip phone
<> C-Media CM109 USB I/O Controller
<> Regulator haptics support
[*] X-Powers AXP2101 power button driver
<> User level driver support
<> GPIO driver support
<> PCF8574 Keypad input device
<> PWM beeper support
<> Rotary encoders connected to GPIO pins
<> Analog Devices ADXL34x Three-Axis Digital Accelerometer
<> IMS Passenger Control Unit driver
<> VTI CMA3000 Tri-axis accelerometer
<> Windows-compatible SoC Button Array
<> TI DRV260X haptics support
<> TI DRV2665 haptics support
<> TI DRV2667 haptics support
<> stl3x1x Light driver support
<> hall_och175 hall driver support
<> LTR553ALS LIGHT SENSOR driver

```

图 2-4: power-key-config

- virtual regulator

```

-> Device Drivers
  -> Voltage and Current Regulator Support
    [*] Virtual regulator consumer support

```

```

Voltage and Current Regulator Support
Arrow keys navigate the menu. <Enter> selects submenus ---> (or empty submenus ----). Highlighted
letters are hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to
exit, <?> for Help, </> for Search. Legend: [*] built-in [ ] excluded <M> module < > module capable

--- Voltage and Current Regulator Support
[ ] Regulator debug support
[*] Fixed voltage regulator support
[*] Virtual regulator consumer support
<> Userspace regulator consumer support
<> Active-semi act8865 voltage regulator
<> Analog Devices AD5398/AD5821 regulators
<> Freescale i.MX on-chip ANATOP LDO regulators
[*] X-POWERS AXP2101 PMIC Regulators
<> Dialog Semiconductor DA9210 regulator
<> Dialog Semiconductor DA9211/DA9212/DA9213/DA9214/DA9215 regulator
<> Fairchild FAN53555 Regulator
<> GPIO regulator support
<> Intersil ISL9305 regulator
<> Intersil ISL6271A Power regulator
<> National Semiconductors LP3971 PMIC regulator driver
<> National Semiconductors LP3972 PMIC regulator driver
<> TI/National Semiconductor LP8720/LP8725 voltage regulators
<> TI LP8755 High Performance PMU driver

```

图 2-5: virtuaal-config

- acin

```
-> Device Drivers
    -> Power supply class support
        < > AXP2202 power virtual acin
```

Power supply class support

Arrow keys navigate the menu. <Enter> selects submenus ---> (or empty submenus ----). Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*] built-in [] excluded <M> module < > module capable

```
--- Power supply class support
[ ] AXP Power drivers ----
[ ] Power supply debug
< > Generic PDA/phone power driver
< > Test power driver
< > DS2780 battery driver
< > DS2781 battery driver
< > DS2782/DS2786 standalone gas-gauge
< > SBS Compliant gas gauge
< > BQ27xxx battery driver
< > Maxim MAX17040 Fuel Gauge
< > Maxim MAX17042/17047/17050/8997/8966 Fuel Gauge
< > MAX8903 Battery DC-DC Charger for USB and Adapter Power
< > TI/National Semiconductor LP8727 charger driver
< > GPIO charger
[ ] Battery charger manager for multiple chargers
< > TI BQ2415x battery charger driver
< > TI BQ24190 battery charger driver
< > TI BQ24250/24251/24257 battery charger driver
< > TI BQ24735 battery charger support
< > TI BQ25890 battery charger driver
< > Silergy SY6974 battery charger driver
< > Cellwise CW2015 battery driver
< > Summit Microelectronics SMB347 Battery Charger
< > LTC2941/LTC2943 Battery Gauge Driver
< > Richtek RT9455 battery charger driver
< > AXP2101 power supply driver
< > AXP152 vbus check
< > AXP803 power supply driver
< > AXP2585 power supply driver
< * > AXP2202 power supply driver
< * > AXP2202 power virtual acin
```

图 2-6: acin-config

2.4 源码结构介绍

- AXP717

```
${ROOT_DIR}/lichee/{KERNEL_VERSION}/
drivers/mfd/axp2101.c
drivers/mfd/axp2101-i2c.c
drivers/regulator/axp2101-regulator.c
drivers/input/misc/axp2101-pek.c
drivers/power/supply/axp2202_battery.c
drivers/power/supply/axp2202_charger.c
drivers/power/supply/axp2202_gpio_power.c
drivers/power/supply/axp2202_usb_power.c
```

2.5 模块框架介绍

AXP 的多功能设备驱动采用 i2c 总线跟主控进行交互，使用 regmap 方式注册访问接口。将 AXP 按照功能抽象出数个子设备模块，共同使用父设备 axp mfd 的资源（bus、irq）。基本的软件结构图如下图所示。

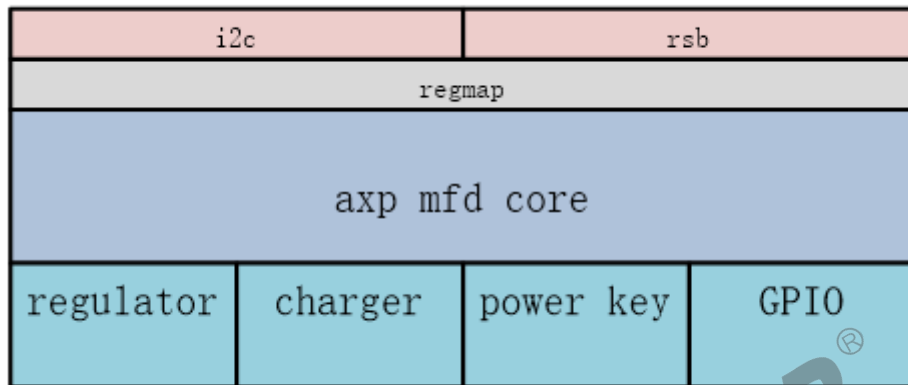


图 2-7: AXP 框架图

将 axp 按照功能划分为几个子设备，分别是 regulator、charger、power key、gpio。每个子设备作为一个 cell，使用父设备的资源（bus，irq），与不同的内核子系统交互，实现完整的电源管理功能。

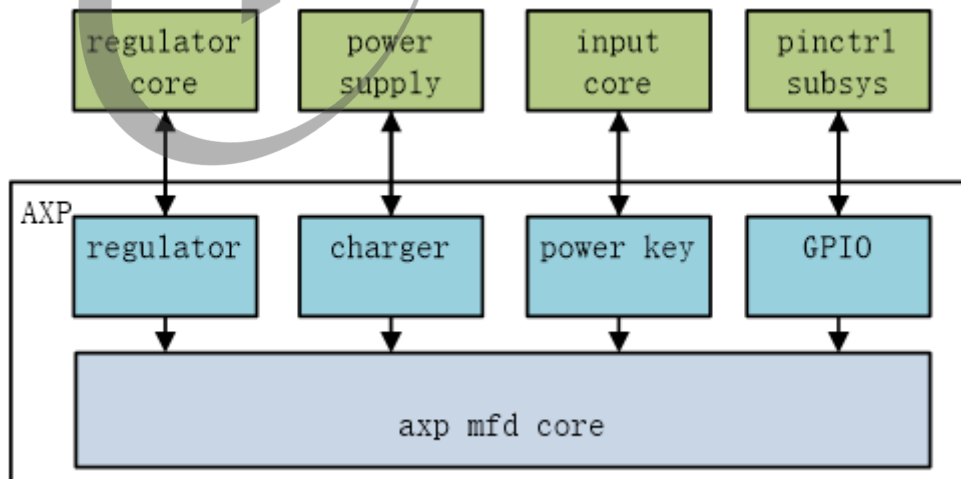


图 2-8: AXP 软件框架图

不同款的 AXP 型号功能框架也会是不一致的，详情各个型号的功能列表如下表所示。

产品名称	regulator	charger	power key	gpio
AXP717	有	有	有	无



3 模块使用范例

3.1 外部 sysfs 节点

AXP 注册了许多外部 sysfs 节点可供调试使用。

3.1.1 Regulator

有关于 regulator 调试节点，主要调整各路电压输出的。

在路径 `/sys/class/regulator/` 下，有关于单路 regulator 属性的节点可以读取数值。

文件路径	功能	属性	设置值
<code>regulator.X/name</code>	各路输出名字	r	对应的输出的名字
<code>regulator.X/max_microvolts</code>	各路输出最大电压	r	对应的最大电压值，单位 uV
<code>regulator.X/min_microvolts</code>	各路输出最小电压	r	对应的最小电压值，单位 uV
<code>regulator.X/state</code>	各路输出状态	r	对应的输出状态， enabled/disabled: 开启\关闭
<code>regulator.X/num_users</code>	各路输出对应的设备个数	r	0\1\2\.....

因为不同 AXP 型号 regulator 对应的节点不一样，因此下面将分开不同 AXP 进行介绍。

- AXP717

电源名称对应表如下：

节点名字	原理图名称
<code>regulator.0</code>	dummy
<code>regulator.1</code>	usb0-vbus
<code>regulator.2</code>	usb1-vbus
<code>regulator.3</code>	dcdc1
<code>regulator.4</code>	dcdc2

节点名字	原理图名称
regulator.5	dcdc3
regulator.6	dcdc4
regulator.7	aldo1
regulator.8	aldo2
regulator.9	aldo3
regulator.10	aldo4
regulator.11	blldo1
regulator.12	blldo2
regulator.13	blldo3
regulator.14	blldo4
regulator.15	clldo1
regulator.16	clldo2
regulator.17	clldo3
regulator.18	clldo4
regulator.19	rtclldo
regulator.20	cpusldo
regulator.21	drivevbus

3.1.2 Virtual-consumer

在路径 /sys/devices/platform/soc/7081400.s_twi/i2c-6/6-0034/ 下则有修改 regulator 输出电压的节点。

文件路径	功能	属性	设置值
reg-virt-consumer.X/max_microvolts	设置输出最大电压	rw	对应的最大电压值，单位 uV
reg-virt-consumer.X/min_microvolts	设置输出最小电压	rw	对应的最小电压值，单位 uV
of_node/name	各路输出名字	r	无

因为不同 AXP 型号 Virtual-consumer 对应的节点不一样，因此下面将分开不同 AXP 进行介绍。

- AXP717

virt-consumer 对应的电路的对应表如下：

节点名字	原理图名称
reg-virt-consumer.1	dcdc1
reg-virt-consumer.2	dcdc2
reg-virt-consumer.3	dcdc3
reg-virt-consumer.4	dcdc4
reg-virt-consumer.5	aldo1
reg-virt-consumer.6	aldo2
reg-virt-consumer.7	aldo3
reg-virt-consumer.8	aldo4
reg-virt-consumer.9	blldo1
reg-virt-consumer.10	blldo2
reg-virt-consumer.11	blldo3
reg-virt-consumer.12	blldo4
reg-virt-consumer.13	clldo1
reg-virt-consumer.14	clldo2
reg-virt-consumer.15	clldo3
reg-virt-consumer.16	clldo4
reg-virt-consumer.17	rtclldo
reg-virt-consumer.18	cpuslldo

3.1.3 Power_supply

在路径 `/sys/class/power_supply/` 下有 `power_supply` 相关的调试节点，可以读出电池以及供电当前状态和设置属性。

下表是电池相关的调试节点，路径为：`/sys/class/power_supply/axp2202-battery/`

文件路径	功能	属性	设置值
capacity	电池剩余电量	r	百分比，0\1\2\.....\100
capacity_alert_min	低电量警告阈值	r	百分比
capacity_level	当前充电等级	r	“UNKNOWN” 未知， “Critical” 接近没电，“LOW” 低电量，“NORMAL” 正常电量， “HIGH” 高电量，“FULL” 满电
charge_counter	当前电池容量	r	单位 mAh
charge_full	充满电的电池容量	r	单位 mAh
constant_charge_current	恒定充电电流	r	单位 mA
energy_full_design	充满电的电池容量	r	单位 mAh

文件路径	功能	属性	设置值
health	电池状况	r	“Unknown” 未知, “Good” 好, “Overheat” 过温, “Dead” 坏掉, “Over voltage” 过压, “Unspecified failure” 错误, “Cold” 冷
present	电池存在	r	0\1: 存在\不存在
serial_number	pmu 型号	r	string
status	电池当前状态	r	“Unknown” 未知, “Charging” 正在充电, “Discharging” 放电, “Not charging” 未在充电, “Full” 满
temp	电池温度	r	单位 °C
temp_alert_min	电池高温预警阈值	r	单位 °C
time_to_empty_now	放电剩余时间	r	单位 min
time_to_full_now	充电剩余时间	r	单位 min
type	设备类别	r	“battery” 电池, “Mains” 火牛, “USB” USB
voltage_now	当前电压	r	单位 uA

下表则是供电相关的节点，路径为：/sys/class/power_supply/axp2202-usb

文件路径	功能	属性	设置值
input_current_limit	输入电流限流值	r	单位 mA
online	USB 是否在使用	r	0\1: 未使用\正在使用
present	USB 是否接上	r	0\1: 插上\没插上
serial_number	pmu 型号	r	string
type	设备类别	r	“battery” 电池, “Mains” 火牛, “USB” USB
voltage_min_design	DC 供电最小设计电压	r	单位 uV
voltage_now	DC 供电时的电压大小	r	单位 uV

说明

AXP717 没有使用 **acin**，因此接入 **usb** 或者适配器都是使用 **usb-power-supply** 的驱动，节点也是公用的。

3.2 Regulator 使用方法

3.2.1 内核代码调用 regulator 示例

以 DCDC1 为例，需要设置 DCDC1 最大输出电压值为 3.4V，需要设置目标电压值为 3V。

```
#include <linux/regulator/consumer.h>

struct regulator *regu= NULL;
int ret = 0;
regu= regulator_get(NULL, "axp2202_dcdc1");
if (IS_ERR(regu)) {
    pr_err("%s: some error happen, fail to get regulator \n", __func__);
    goto exit;
}

//set output voltage to 3V
ret = regulator_set_voltage(regu, 3000000, 3400000);
if (0 != ret) {
    pr_err("%s: some error happen, fail to set regulator voltage!\n", __func__);
    goto exit;
}

//enalbe regulator
ret = regulator_enable(regu);
if (0 != ret) {
    pr_err("%s: some error happen, fail to enable regulator!\n", __func__);
    goto exit;
}

//disalbe regulator
ret = regulator_disable(regu);
if (0 != ret) {
    pr_err("%s: some error happen, fail to disable regulator!\n", __func__);
    goto exit;
}

//put regulator, when module exit
regulator_put(regu);
```

💡 技巧

`regulator_get` 调用有两种方法：

第一种就是直接获取 `dtb` 中 `regulator` 的句柄，比如说 `axp2202_dcdc1`。调用 `regulator_get` 时第一个参数不需要配置设备，直接写 `NULL`。

第二种是在 `dtb` 的设备里配置 `regulator` 的节点，比如说 `dtb` 某个设备里配置 `regulator0 = "axp2202\dc1"`；，该设备调用时写法为：`regulator_get(dev, "regulator0")`；。

3.2.2 Regulator shell 命令使用示例

AXP regulator 可以通过 shell 命令控制和设置其开关以及输出电压，各路文件节点建立在在 `/sys/devices/platform` 目录下

- AXP717

AXP717 可控的节点可参考上面 virtual 章节 [Virtual-consumer](#)

以设置 DCDC1 输出最大电压为 3.3V，设置目标电压为 3.0V 为例做说明。

```
cd /sys/devices/platform/soc/7081400.s_twi/i2c-6/6-0034/reg-virt-consumer.1
cat of_node/name //确认是否为DCDC1

//设置输出电压为3.0V
echo 3300000 > max_microvolts
echo 3000000 > min_microvolts

//关闭输出
echo 3300000 > max_microvolts
echo 3000000 > min_microvolts
echo 0 > min_microvolts
```

3.2.3 usb_count 查看

根据上面 Regulator 章节 [Regulator](#) 找到对应的 regulator 节点，这里以 dcdc1 为例，其节点名称为 regulator.3，则在/sys/class/regulator 目录下就有个 regulator.3 目录，regulator.3 目录有个 num_users 节点，cat 此节点就可以获得当前 use_count 值。

```
cat /sys/class/regulator/regulator.3/num_users
```

num_uesrs 代表当前有多少设备使用了 regulator 节点用来控制输出电压。

4 FAQ

4.1 调试方法

在设备进行开发过程中，难免需要对各路电源进行调试，控制电源各路电压等操作，内核中提供了对电源调试的方式。

4.1.1 调试工具

4.1.1.1 power key 调试方式

在用户空间调用 `getevent` 命令，通过标准 input 系统上报的 input 事件，可以确认 power key 是否能正常工作，是否能正常上报 input 事件。

```
add device 2: /dev/input/event1
name:      "axp2202-pek"
poll 4, returned 1
/dev/input/event1: 0001 0074 00000001
poll 4, returned 1
/dev/input/event1: 0000 0000 00000000
poll 4, returned 1
/dev/input/event1: 0001 0074 00000000
poll 4, returned 1
/dev/input/event1: 0000 0000 00000000
poll 4, returned 1
/dev/input/event1: 0001 0074 00000001
poll 4, returned 1
/dev/input/event1: 0000 0000 00000000
poll 4, returned 1
/dev/input/event1: 0001 0074 00000000
poll 4, returned 1
/dev/input/event1: 0000 0000 00000000
```

按下按钮为 1，弹起为 0。0074 为 power 事件。

4.1.2 调试节点

4.1.2.1 /sys/kernel/debug/regulator/regulator_summary 节点

shell 命令查询 regulator 状态。

kernel 提供调试结点供电源进行调试进行，我们可以通过 kernel 的调试结点获取各路电源的各个详细状态。以 AXP2101 的设备举例，首先需要 mount debugfs 文件系统。

```
mount -t debugfs none /sys/kernel/debug
```

```
cat /sys/kernel/debug/regulator/regulator_summary
```

regulator	use	open	bypass	voltage	current	min	max
regulator-dummy	0	8	0	0mV	0mA	0mV	0mV
dmic						0mV	0mV
uart2						0mV	0mV
uart1						0mV	0mV
twi3						0mV	0mV
twi2						0mV	0mV
twi1						0mV	0mV
twi0						0mV	0mV
twi6						0mV	0mV
usb0-vbus	0	0	0	5000mV	0mA	5000mV	5000mV
usb1-vbus	1	2	0	5000mV	0mA	5000mV	5000mV
5200000.ohci1-controller						0mV	0mV
5200000.ehci1-controller						0mV	0mV
axp2202-dcdc1	0	2	0	900mV	0mA	500mV	1540mV
cpu0						900mV	900mV
reg-virt-consumer.1						0mV	0mV
axp2202-dcdc2	0	1	0	950mV	0mA	500mV	3400mV
reg-virt-consumer.2						0mV	0mV
axp2202-dcdc3	0	1	0	1200mV	0mA	500mV	1840mV
reg-virt-consumer.3						0mV	0mV
axp2202-dcdc4	0	1	0	1000mV	0mA	1000mV	3700mV
reg-virt-consumer.4						0mV	0mV
axp2202-aldo1	0	3	0	2800mV	0mA	500mV	3500mV
sensor1						2800mV	3300mV
sensor0						2800mV	3300mV
reg-virt-consumer.5						0mV	0mV
axp2202-aldo2	1	4	0	1800mV	0mA	500mV	3500mV
sensor1						1800mV	3300mV
sensor0						1800mV	3300mV
sensor0						1800mV	3300mV
reg-virt-consumer.6						0mV	0mV
axp2202-aldo3	0	1	0	3300mV	0mA	500mV	3500mV
reg-virt-consumer.7						0mV	0mV
axp2202-aldo4	0	2	0	1800mV	0mA	500mV	3500mV
codec						1800mV	1800mV
reg-virt-consumer.8						0mV	0mV
axp2202-blldo1	2	3	0	3300mV	0mA	500mV	3500mV
soc@03000000:wlan@0						3300mV	3300mV
soc@03000000:wlan@0						3300mV	3300mV
reg-virt-consumer.9						0mV	0mV
axp2202-blldo2	0	1	0	2500mV	0mA	500mV	3500mV
reg-virt-consumer.10						0mV	0mV
axp2202-blldo3	1	1	0	2800mV	0mA	500mV	3500mV
reg-virt-consumer.11						0mV	0mV
axp2202-blldo4	0	3	0	1500mV	0mA	500mV	3500mV
sensor1						1500mV	1800mV
sensor0						1200mV	1800mV
reg-virt-consumer.12						0mV	0mV
axp2202-clldo1	3	3	0	1800mV	0mA	500mV	3500mV
codec						1800mV	1800mV
1-0036						1800mV	1800mV

reg-virt-consumer.13						0mV	0mV
axp2202-cldo2	0	1	0	3300mV	0mA	500mV	3500mV
reg-virt-consumer.14						0mV	0mV
axp2202-cldo3	3	4	0	3300mV	0mA	500mV	3500mV
1-0036						3300mV	3300mV
sdcc2						0mV	0mV
uart0						0mV	0mV
reg-virt-consumer.15						0mV	0mV
axp2202-cldo4	0	1	0	3300mV	0mA	500mV	3500mV
reg-virt-consumer.16						0mV	0mV
axp2202-rtcldo	0	1	0	1800mV	0mA	1800mV	1800mV
reg-virt-consumer.17						0mV	0mV
axp2202-cpusldo	0	1	0	900mV	0mA	500mV	1400mV
reg-virt-consumer.18						0mV	0mV
axp2202-drivevbus	0	0	0	0mV	0mA	0mV	0mV

4.1.2.2 regmap registers 节点

shell 命令读写寄存器。

寄存器调试是指直接对 PMIC 的寄存器进行读写操作，此操作应该对寄存器有了解的情况下进行操作，不正确的操作方式将会导致芯片烧毁。在终端中，对抛出的调试结点进行读写操作，即可对寄存器进行读写操作。无论是读还是写寄存器，都应该首先挂载 debugfs 文件系统。

由于 PMIC 是通过 regmap 进行读写操作，应该可以使用 regmap 的调试结点进行对 PMIC 的读写访问操作。regmap 的调试结点在 debugfs 文件系统下面，通过对 regmap 调试结点的操作可以对 PMIC 的寄存器进行读写访问操作。

• 写操作

寄存器调试挂载在debugfs文件系统。

```
mount -t debugfs none /sys/kernel/debug
echo ${reg} ${value} > /sys/kernel/debug/regmap/${dev-name}/registers
```

实例：

```
echo 0xff 0x01 > /sys/kernel/debug/regmap/4-0034/registers
写0xff寄存器值为0x01
```

• 读操作

寄存器调试挂载在debugfs文件系统。

```
mount -t debugfs none /sys/kernel/debug
cat /sys/kernel/debug/regmap/${dev-name}/registers
```

实例：

```
cat /sys/kernel/debug/regmap/4-0034/registers
读取pmic所有寄存器
```

4.1.2.3 axp_reg 节点

另外，还支持 axp 驱动自定义节点 axp_reg 读写寄存器。但是这种用法是不推荐的，因为有标准 regmap 方式来读写寄存器，根本没必要用私有非标的方式。示例如下。

```
往axp寄存器0x0f写入值0x55:
echo 0x0f55 > /sys/class/axp/axp_reg
读出axp寄存器0x0f的值:
echo 0x0f > /sys/class/axp/axp_reg
cat /sys/class/axp/axp_reg
```

4.1.2.4 debug_mask 节点

axp 驱动自定义节点 debug_mask 打开和关闭调试信息。相关调试信息参考具体的 PMIC 驱动。示例如下。

```
系统打印等级设置为8:
echo 8 > /proc/sys/kernel/printk
打开所有axp调试信息:
echo 0xf > /sys/class/axp/debug_mask
关闭所有axp调试信息:
echo 0x0 > /sys/class/axp/debug_mask
```

调试信息一般如下。

```
[ 712.458412] ic_temp = 45
[ 712.461311] vbat = 3977
[ 712.464082] ibat = -779
[ 712.466280] healthd: battery l=96 v=3977 t=30.0 h=2 st=3 c=-779 fc=5066880 chg=
[ 712.475174] charge_ibat = 0
[ 712.478448] dis_ibat = 779
[ 712.481545] ocv = 4073
[ 712.484239] rest_vol = 96
[ 712.487182] rdc = 123
[ 712.489862] batt_max_cap = 5066
[ 712.493472] coulumb_counter = 4857
[ 712.497583] AXP803_COULOMB_CTL = 0xe0
[ 712.501803] ocv_percentage = 86
[ 712.505436] col_percentage = 96
[ 712.509061] bat_current_direction = 0
[ 712.513386] ext_valid = 0
```

著作权声明

版权所有 © 2022 珠海全志科技股份有限公司。保留一切权利。

本文档及内容受著作权法保护，其著作权由珠海全志科技股份有限公司（“全志”）拥有并保留一切权利。

本文档是全志的原创作品和版权财产，未经全志书面许可，任何单位和个人不得擅自摘抄、复制、修改、发表或传播本文档内容的部分或全部，且不得以任何形式传播。

商标声明

、 全志科技 （不完全列举）均为珠海全志科技股份有限公司的商标或者注册商标。在本文档描述的产品中出现的其它商标，产品名称，和服务名称，均由其各自所有人拥有。

免责声明

您购买的产品、服务或特性应受您与珠海全志科技股份有限公司（“全志”）之间签署的商业合同和条款的约束。本文档中描述的全部或部分产品、服务或特性可能不在您所购买或使用的范围内。使用前请认真阅读合同条款和相关说明，并严格遵循本文档的使用说明。您将自行承担任何不当使用行为（包括但不限于如超压，超频，超温使用）造成的不利后果，全志概不负责。

本文档作为使用指导仅供参考。由于产品版本升级或其他原因，本文档内容有可能修改，如有变更，恕不另行通知。全志尽全力在本文档中提供准确的信息，但并不确保内容完全没有错误，因使用本文档而发生损害（包括但不限于间接的、偶然的、特殊的损失）或发生侵犯第三方权利事件，全志概不负责。本文档中的所有陈述、信息和建议并不构成任何明示或暗示的保证或承诺。

本文档未以明示或暗示或其他方式授予全志的任何专利或知识产权。在您实施方案或使用产品的过程中，可能需要获得第三方的权利许可。请您自行向第三方权利人获取相关的许可。全志不承担也不代为支付任何关于获取第三方许可的许可费或版税（专利税）。全志不对您所使用的第三方许可技术做出任何保证、赔偿或承担其他义务。