

RK806 开发指南

文件标识：RK-KF-YF-199

发布版本：V1.0.0

日期：2021-12-24

文件密级：☐绝密 ☐秘密 ☐内部资料 ☒公开

免责声明

本文档按“现状”提供，瑞芯微电子股份有限公司（“本公司”，下同）不对本文档的任何陈述、信息和内容的准确性、可靠性、完整性、适销性、特定目的性和非侵权性提供任何明示或暗示的声明或保证。本文档仅作为使用指导的参考。

由于产品版本升级或其他原因，本文档将可能在未经任何通知的情况下，不定期进行更新或修改。

商标声明

“Rockchip”、“瑞芯微”、“瑞芯”均为本公司的注册商标，归本公司所有。

本文档可能提及的其他所有注册商标或商标，由其各自拥有者所有。

版权所有 © 2021 瑞芯微电子股份有限公司

超越合理使用范畴，非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

瑞芯微电子股份有限公司

Rockchip Electronics Co., Ltd.

地址：福建省福州市铜盘路软件园A区18号

网址：www.rock-chips.com

客户服务电话：+86-4007-700-590

客户服务传真：+86-591-83951833

客户服务邮箱：fae@rock-chips.com

前言

概述

本文档主要介绍 RK806 的各个子模块，介绍相关概念、功能、dts 配置和一些常见问题的分析定位。

产品版本

| 芯片名称 | 内核版本 |
|-------|------|
| RK806 | 5.10 |

读者对象

本文档（本指南）主要适用于以下工程师：

技术支持工程师

软件开发工程师

修订记录

| 版本号 | 作者 | 修改日期 | 修改说明 |
|--------|-----|------------|------|
| V1.0.0 | 许盛飞 | 2021-12-24 | 初始版本 |

目录

RK806 开发指南

1. 基础
 - 1.1 概述
 - 1.2 功能
 - 1.3 芯片引脚功能
 - 1.4 重要概念
 - 1.5 双PMIC协同工作
2. 配置
 - 2.1 驱动和 menuconfig
 - 2.2 DTS 配置
 - 2.3 函数接口
3. Debug
 - 3.1 内核
 - 3.2 Kernel 5.10 内核

1. 基础

1.1 概述

RK806 是一款高性能 PMIC，RK806 集成 10 个大电流 DCDC、5 个 NLDO、6 个 PLDO、可调上电时序等功能。

系统中各路电源总体分为两种：DCDC 和 LDO。两种电源的总体特性如下（详细资料请自行搜索）：

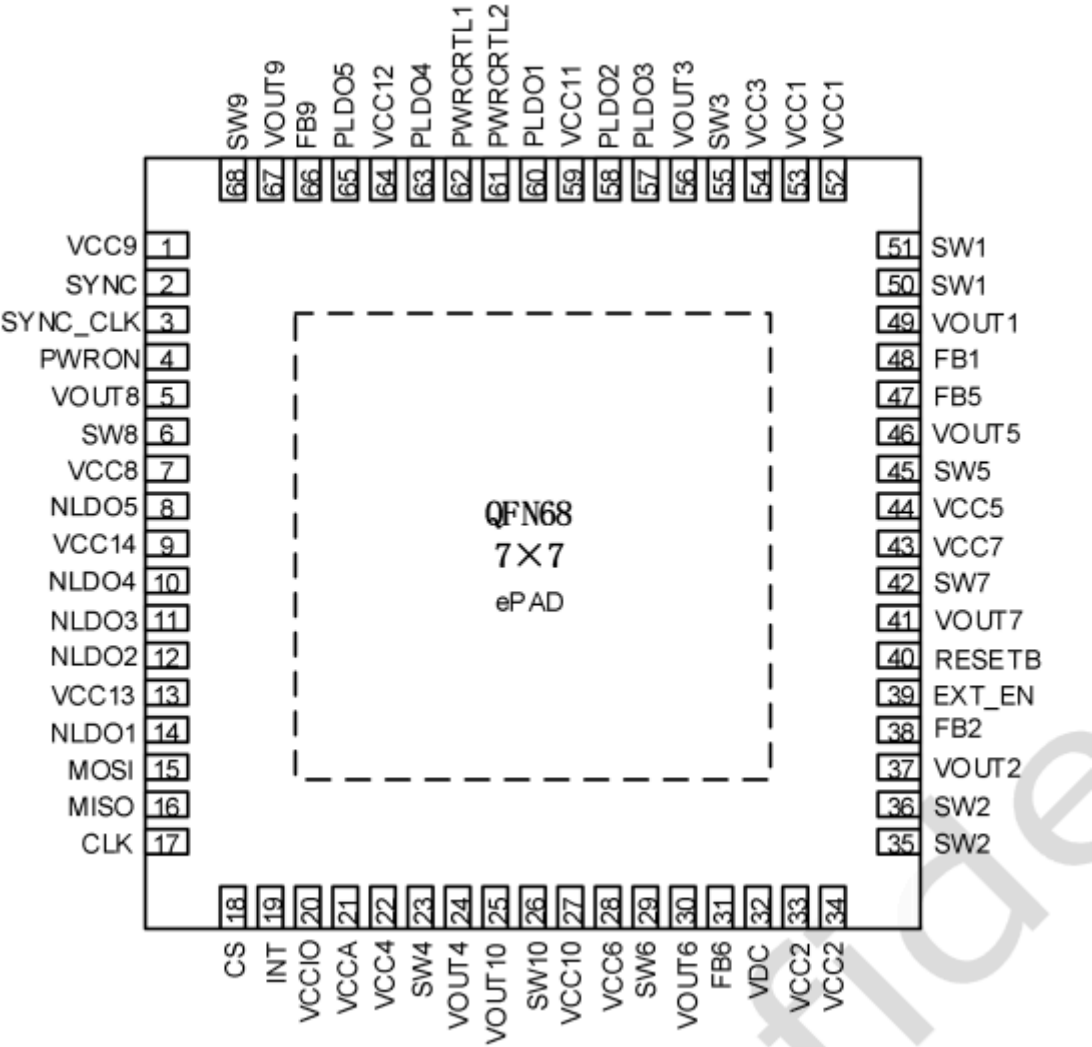
1. DCDC：输入输出压差大时，效率高，但是存在纹波比较大的问题，成本高，所以大压差，大电流负载时使用。一般有两种工作模式。PWM 模式：纹波瞬态响应好，效率低；PFM 模式：效率高，但是负载能力差。
2. PLDO：和以前 PMIC 的 LDO 一样（最低输入电压 2.0V，最高 5.5V）
3. NLDO：支持低压输入输出，损耗更小（比如 1.1V 输入 0.9V 输出）

1.2 功能

从使用者的角度看，RK806 的功能概况起来可以分为 3 个部分：

1. regulator 功能：控制各路 DCDC、LDO 电源状态；
2. gpio 功能：有 3 个 IO 可用，可以控制整个 PMIC 进待机，也可以单独分给指定电源进待机，也可当普通 gpio 使用；
3. pwrkey 功能：检测 power 按键的按下/释放，可以为 AP 节省一个 gpio。

1.3 芯片引脚功能



下面描述中，SLEEP 和 INT 引脚需要重点关注：

| PIN NO | PIN NAME | PIN DESCRIPTION | I/O |
|--------|-----------------------|---|-----|
| 1 | VCC9 | Power supply of buck9. | I |
| 2 | SYNC | Master and slave synchronization signal. | I/O |
| 3 | SYNC_CLK | 32k synchronization clk. | I/O |
| 4 | PWRON | Power on key. The internal pull-up resistance is about 45K. | I |
| 5 | VOUT8 | Output feedback voltage of buck8. | O |
| 6 | SW8 | Switching node of buck8. | O |
| 7 | VCC8 | Power supply of buck8. | I |
| 8 | NLDO5 | NMOS LDO5 output. | O |
| 9 | VCC14 | Power supply of NLDO3/4/5. | I |
| 10 | NLDO4 | NMOS LDO4 output. | O |
| 11 | NLDO3 | NMOS LDO3 output. | O |
| 12 | NLDO2 | NMOS LDO2 output. | O |
| 13 | VCC13 | Power supply of NLDO1/2. | I |
| 14 | NLDO1 | NMOS LDO1 output. | O |
| 15 | MOSI/SCL | SPI MOSI. I2C SCL. | I/O |
| 16 | MISO/SDA/ PWRCRTL3 | SPI MISO. I2C SDA. PWRCRTL3 control. | I/O |
| 17 | CLK | SPI CLK. | O |
| 18 | CS | CS is used to select the I2C and SPI functions, when it connected to VCCA is the I2C mode | I |
| 19 | INT | Interrupt. | O |
| 20 | VCCIO/ PLDO6 | PMOS LDO6 output. VCCIO for SPI/I2C interface. | O |
| 21 | VCCA | Analog power supply. Power supply of PLDO6 and system | I |

| PIN NO | PIN NAME | PIN DESCRIPTION | I/O |
|---------|----------|---|-----|
| | | logic. | |
| 22 | VCC4 | Power supply of buck4. | I |
| 23 | SW4 | Switching node of buck4. | O |
| 24 | VOU4 | Output feedback voltage of buck4. | O |
| 25 | VOU10 | Output feedback voltage of buck10. | O |
| 26 | SW10 | Switching node of buck10. | O |
| 27 | VCC10 | Power supply of buck10. | I |
| 28 | VCC6 | Power supply of buck6. | I |
| 29 | SW6 | Switching node of buck6. | O |
| 30 | VOU6 | Output feedback voltage of buck6. | O |
| 31 | FB6 | Externed divided resistor mode feedback voltage of buck6. | O |
| 32 | VDC | VDC power on signal. | I |
| 33 | VCC2 | Power supply of buck2. | I |
| 34 | VCC2 | Power supply of buck2. | I |
| 35 | SW2 | Switching node of buck2. | O |
| 36 | SW2 | Switching node of buck2. | O |
| 37 | VOU2 | Output feedback voltage of buck2. | O |
| 38 | FB2 | Externed divided resistor mode feedback voltage of buck2. | O |
| 39 | EXT_EN | Control externed DCDC enable. Master/Slave select. | I |
| 40 | RESETB | Reset the AP. | I/O |
| 41 | VOU7 | Output feedback voltage of buck7. | O |
| 42 | SW7 | Switching node of buck7. | O |
| 43 | VCC7 | Power supply of buck7. | I |
| 44 | VCC5 | Power supply of buck5. | I |
| 45 | SW5 | Switching node of buck5. | O |
| 46 | VOU5 | Output feedback voltage of buck5. | O |
| 47 | FB5 | Externed divided resistor mode feedback voltage of buck5. | O |
| 48 | FB1 | Externed divided resistor mode feedback voltage of buck1. | O |
| 49 | VOU1 | Output feedback voltage of buck1. | O |
| 50 | SW1 | Switching node of buck1. | O |
| 51 | SW1 | Switching node of buck1. | O |
| 52 | VCC1 | Power supply of buck1. | I |
| 53 | VCC1 | Power supply of buck1. | I |
| 54 | VCC3 | Power supply of buck3. | I |
| 55 | SW3 | Switching node of buck3. | O |
| 56 | VOU3 | Output feedback voltage of buck3. | O |
| 57 | PLDO3 | PMOS LDO3 output. | O |
| 58 | PLDO2 | PMOS LDO2 output. | O |
| 59 | VCC11 | Power supply of PLDO1/2/3. | I |
| 60 | PLDO1 | PMOS LDO1 output. | O |
| 61 | PWRCRTL2 | PWRCRTL2 control. | I/O |
| 62 | PWRCRTL1 | PWRCRTL1 control. | I/O |
| 63 | PLDO4 | PMOS LDO4 output. | O |
| 64 | VCC12 | Power supply of PLDO4/5. | I |
| 65 | PLDO5 | PMOS LDO5 output. | O |
| 66 | FB9 | Externed divided resistor mode feedback voltage of buck9. | O |
| 67 | VOU9 | Output feedback voltage of buck9. | O |
| 68 | SW9 | Switching node of buck9. | O |
| Exposed | ePAD | Ground | |

1.4 重要概念

- 支持SPI和I2C通信

当前软件上只支持SPI通信模式，默认SPI频率是1M。

- PMIC 有 3 种工作模式

1. PMIC normal 模式

系统正常运行时 PMIC 处于 normal 模式，此时 pmic_sleep 的有效电平可配置。

2. PMIC sleep 模式

系统休眠时需要待机功耗尽量低，PMIC 会切到 sleep 模式减低自身功耗，这时候一般会降低某些路的输出电压，或者直接关闭输出，这可以根据实际产品需求进行配置。系统待机时 AP 通过 SPI 指令把 pmic_sleep 配置成 sleep 模式，然后拉高 pmic_sleep 即可让 PMIC 进入 sleep 状态；当 SoC 唤醒时 pmic_sleep 恢复为低电平，PMIC 退出休眠模式。

3. PMIC shutdown 模式

当系统进入关机流程的时候，PMIC 需要完成整个系统的电源下电操作。AP 通过 SPI 指令把 pmic_sleep 配置成 shutdown 模式，然后拉高 pmic_sleep 即可让 PMIC 进入 shutdown 状态。

- int 引脚

常态为高电平，当有中断产生的时候变为低电平。如果中断没有被处理，则会一直维持低电平。

- pwrctrl1/pwrctrl2/pwrctrl3 引脚

这两个引脚可以当普通的 gpio 使用。

- pwrn 引脚

pwrkey 的功能需要硬件上将 power 按键接到这个引脚，驱动通过这个引脚来判断按下/释放。

- 各路 DCDC 的工作模式

DCDC 有 PWM（也叫 force PWM）、PFM 模式，但是 PMIC 有一种模式会动态切换 PWM、PFM，这就是我们通常所说的 AUTO 模式。PMIC 支持 PWM、AUTO PWM/PFM 两种模式，AUTO 模式效率高但是纹波瞬态响应会差。出于系统稳定性考虑，运行时都是设置为 PWM 模式，系统进入休眠时会选择切换到 AUTO PWM/PFM。

- PLDO6 电压调节

PMIC 与 AP 的 VCCIO 供电口，只要有用到这些 io 口就必须供电，且供电电压要大于等于 1.8V，即 PLDO6 通常应用中不能低于 1.8V，且待机时不能关闭。

- DCDC 和 LDO 的运行时电压调节范围

1. DCDC 电压范围不连续：

| 电压范围(V) | 步进值(mV) | 具体档位值(V) |
|---------------|---------|-----------------------|
| 0.7125 ~ 1.45 | 6.25 | 0.5、0.50625、..... 1.5 |
| 1.5~ 3.4 | 25 | 1.5、1.525、..... 3.4 |

2. NLDO/PLDO 电压连续：

| 电压范围(V) | 步进值(mV) | 具体档位值(V) |
|-----------|---------|------------------------|
| 0.5 ~ 3.4 | 12.5 | 0.5125、0.525、..... 3.4 |

1.5 双PMIC协同工作

- 主从：协同工作时两颗 PMIC 分主从模式，主从芯片配置通过第一次上电时 EXT_EN 引脚电平状态来区分。EXT_EN 和 VCCA 短接在一起的为 Slave，EXT_EN 悬空或有电阻下拉的为主机。
- 同步：双 RK806 协同工作，两颗芯片的 SYNC_CLK 和 SYNC 互连，主芯片提供时钟（SYNC_CLK，频率接近 32K）从芯片接收，SYNC 提供同步信号，产生同步脉冲用于实现：开机、关机、复位及上下电时序。

- PWRON、RESETB信号：主从的这两个信号分别短接在一起，用于PMIC的开机始能，和外接复位按键时的外部复位信号输入。
- VDC信号：主从的VDC脚可以短接在一起，也可以把SALVE的VDC接到MASTER的EXT_EN上。

2. 配置

2.1 驱动和 menuconfig

RK806 驱动文件：

```
drivers/mfd/rk806-core.c
drivers/mfd/rk806-spi.c
drivers/pinctrl/pinctrl-rk806.c
drivers/regulator/rk806-regulator.c
```

menuconfig 里对应的宏配置：

```
CONFIG_PINCTRL_RK806=y
CONFIG_MFD_RK806_SPI=y
CONFIG_REGULATOR_RK806=y
```

2.2 DTS 配置

5.10 内核配置

DTS 的配置包括：spi挂载、gpio、regulator 等部分。

单RK806 dts配置

```
&pinctrl {
    pmic {
        soc_slppin_gpio: soc_slppin_gpio {
            rockchip,pins = <0 RK_PA2
                RK_FUNC_GPIO &pcfg_output_low>;
        };
        soc_slppin_shutdown: soc_slppin_shutdown {
            rockchip,pins = <0 RK_PA2
                RK_FUNC_GPIO &pcfg_output_high>;
        };
    };
};

&spi2 {
    status = "okay";
    assigned-clocks = <&cru CLK_SPI2>;
    assigned-clock-rates = <200000000>;
    num-cs = <&spi2 {
```

```

status = "okay";
assigned-clocks = <&cru CLK_SPI2>;
assigned-clock-rates = <200000000>;
num-cs = <2>;

rk806single@0 {
    compatible = "rockchip,rk806";
    spi-max-frequency = <1000000>;
    reg = <0x0>;

    interrupt-parent = <&gpio0>;
    interrupts = <7 IRQ_TYPE_LEVEL_LOW>;

    pinctrl-names = "default", "pmic-sleep", "pmic-power-off", "pmic-
reset";

    pinctrl-0 = <&soc_slppin_gpio>, <&rk806_dvs1_null>,
<&rk806_dvs2_null>, <&rk806_dvs3_null>;
    pinctrl-1 = <&soc_slppin_gpio>, <&rk806_dvs1_slp>,
<&rk806_dvs2_null>, <&rk806_dvs3_null>;
    pinctrl-2 = <&rk806_dvs1_pwrtn>, <&rk806_dvs2_null>,
<&rk806_dvs3_null>;
    pinctrl-3 = <&rk806_dvs1_rst>, <&rk806_dvs2_null>,
<&rk806_dvs3_null>;

    /* 2800mv-3500mv */
    low_voltage_threshold = <3000>;
    /* 2700mv-3400mv */
    shutdown_voltage_threshold = <2700>;
    /* 140 160 */
    shutdown_temperature_threshold = <160>;
    hotdie_temperature_threshold = <115>;

    /* 0: restart PMU;
    * 1: reset all the power off reset registers,
    *     forcing the state to switch to ACTIVE mode;
    * 2: Reset all the power off reset registers,
    *     forcing the state to switch to ACTIVE mode,
    *     and simultaneously pull down the RESETB PIN for 5mS before
releasing
    */
    pmic-reset-func = <1>;

    vcc1-supply = <&vcc5v0_sys>;
    vcc2-supply = <&vcc5v0_sys>;
    vcc3-supply = <&vcc5v0_sys>;
    vcc4-supply = <&vcc5v0_sys>;
    vcc5-supply = <&vcc5v0_sys>;
    vcc6-supply = <&vcc5v0_sys>;
    vcc7-supply = <&vcc5v0_sys>;
    vcc8-supply = <&vcc5v0_sys>;
    vcc9-supply = <&vcc5v0_sys>;
    vcc10-supply = <&vcc5v0_sys>;
    vcc11-supply = <&vcc_2v0_pldo_s3>;
    vcc12-supply = <&vcc5v0_sys>;
    vcc13-supply = <&vcc_1v1_nldo_s3>;
    vcc14-supply = <&vcc_1v1_nldo_s3>;

```

```

vcca-supply = <&vcc5v0_sys>;2>;

regulators {
    vdd_gpu_s0: vdd_gpu_mem_s0: DCDC_REG1 {
        regulator-always-on;
        regulator-boot-on;
        regulator-min-microvolt = <550000>;
        regulator-max-microvolt = <950000>;
        regulator-ramp-delay = <12500>;
        regulator-name = "vdd_gpu_s0";
        regulator-state-mem {
            regulator-off-in-suspend;
        };
    };

    vdd_cpu_lit_s0: vdd_cpu_lit_mem_s0: DCDC_REG2 {
        regulator-always-on;
        regulator-boot-on;
        regulator-min-microvolt = <550000>;
        regulator-max-microvolt = <950000>;
        regulator-ramp-delay = <12500>;
        regulator-name = "vdd_cpu_lit_s0";
        regulator-state-mem {
            regulator-off-in-suspend;
        };
    };

    vdd_log_s0: DCDC_REG3 {
        .....
    };

    vdd_vdenc_s0: vdd_vdenc_mem_s0: DCDC_REG4 {
        .....
    };
    .....
};

};
};

```

双RK806 dts配置:

```

&pinctrl {
    pmic {
        soc_slppin_gpio: soc_slppin_gpio {
            rockchip,pins = <0 RK_PA2
                RK_FUNC_GPIO &pcfg_output_low>;
        };
        soc_slppin_shutdown: soc_slppin_shutdown {
            rockchip,pins = <0 RK_PA2
                RK_FUNC_GPIO &pcfg_output_high>;
        };
    };
};

&spi2 {

```

```

status = "okay";
assigned-clocks = <&cru CLK_SPI2>;
assigned-clock-rates = <200000000>;
num-cs = <2>;
rk806master@0 {
    compatible = "rockchip,rk806";
    spi-max-frequency = <1000000>;
    reg = <0x0>;

    interrupt-parent = <&gpio0>;
    interrupts = <7 IRQ_TYPE_LEVEL_LOW>;
    /* 0: restart PMU;
     * 1: reset all the power off reset registers,
     *     forcing the state to switch to ACTIVE mode;
     * 2: Reset all the power off reset registers,
     *     forcing the state to switch to ACTIVE mode,
     *     and simultaneously pull down the RESETB PIN for 5mS before
releasing
     */
    pmic-reset-func = <1>;

    vcc1-supply = <&vcc5v0_sys>;
    vcc2-supply = <&vcc5v0_sys>;
    .....
    vcca-supply = <&vcc5v0_sys>;
    regulators {
        vdd_gpu_s0: DCDC_REG1 {
            regulator-always-on;
            regulator-boot-on;
            regulator-min-microvolt = <550000>;
            regulator-max-microvolt = <950000>;
            regulator-ramp-delay = <12500>;
            regulator-name = "vdd_gpu_s0";
            regulator-state-mem {
                regulator-off-in-suspend;
            };
        };
        vdd_npu_s0: DCDC_REG2 {
            .....
        };
        .....
    };
};
rk806slave@1 {
    compatible = "rockchip,rk806";
    spi-max-frequency = <1000000>;
    reg = <0x01>;

    interrupt-parent = <&gpio0>;
    interrupts = <7 IRQ_TYPE_LEVEL_LOW>;
    /* 0: restart PMU;
     * 1: reset all the power off reset registers,
     *     forcing the state to switch to ACTIVE mode;
     * 2: Reset all the power off reset registers,
     *     forcing the state to switch to ACTIVE mode,

```

```

        *    and simultaneously pull down the RESETB PIN for 5mS before
releasing
    */
    pmic-reset-func = <1>;

    vcc1-supply = <&vcc5v0_sys>;
    vcc2-supply = <&vcc5v0_sys>;
    .....
    vcca-supply = <&vcc5v0_sys>;

    pwrkey {
        status = "disabled";
    };
    regulators {
        vdd_cpu_big1_s0: DCDC_REG1 {
            regulator-always-on;
            regulator-boot-on;
            regulator-min-microvolt = <550000>;
            regulator-max-microvolt = <950000>;
            regulator-ramp-delay = <12500>;
            regulator-name = "vdd_cpu_big1_s0";
            regulator-state-mem {
                regulator-off-in-suspend;
            };
        };
        vdd_cpu_big0_s0: DCDC_REG2 {
            .....
        };
        .....
    };
};
};
};

```

1. spi 挂载

整个完整的 rk806 节点挂在对应的spi 节点下面，并且配置 status = "okay";

2. 主体部分

- 不可修改：

```

compatible = "rockchip,rk806";
spi-max-frequency = <1000000>;
reg = <0x0>;

interrupt-parent = <&gpio0>;
interrupts = <7 IRQ_TYPE_LEVEL_LOW>;

```

- 可修改（按照 pinctrl 规则）

interrupt-parent: pmic_int 隶属于哪个 gpio;

interrupts: pmic_int 在 interrupt-parent 的 gpio 上的引脚索引编号和极性;

3. pwrkey、gpio

项目中若没有用到pwrkey或者gpio功能，可以在 dts 里增加pwrkey、gpio 节点，并且显式指明状态为 status = "disabled", 这样就不会使能驱动，但是开机信息会有错误 log 报出，可以忽略。

```
pwrkey {  
    status = "disabled";  
};
```

4. regulator

- `regulator-compatible`：驱动注册时需要匹配的名字，不能改动，否则会加载失败；
- `regulator-name`：电源的名字，建议和硬件图上保持一致，使用 `regulator_get` 接口时需要匹配这个名字；
- `regulator-init-microvolt`：u-boot阶段的初始化电压，kernel阶段无效；
- `regulator-min-microvolt`：运行时可以调节的最小电压；
- `regulator-max-microvolt`：运行时可以调节的最大电压；
- `regulator-initial-mode`：运行时 DCDC 的工作模式，一般配置为 1。1：force pwm，2：auto pwm/pfm；
- `regulator-mode`：休眠时 DCDC 的工作模式，一般配置为 2。1：force pwm，2：auto pwm/pfm；
- `regulator-initial-state`：suspend 时的模式，必须配置成 3；
- `regulator-boot-on`：存在这个属性时，在注册 regulator 的时候就会使能这路电源；
- `regulator-always-on`：存在这个属性时，表示运行时不允许关闭这路电源且会在注册的时候使能这路电源；
- `regulator-ramp-delay`：DCDC 的电压上升时间，固定配置为 12500；
- `regulator-on-in-suspend`：休眠时保持上电状态，想要关闭该路电源，则改成“regulator-off-in-suspend”；
- `regulator-suspend-microvolt`：休眠不断电情况下的待机电压。

5.10内核配置

请参考5.10内核DTS配置。

2.3 函数接口

如下几个接口基本可以满足日常使用，包括 regulator 开、关、电压设置、电压获取等：

1. 获取 regulator：

```
struct regulator *devm_regulator_get_optional(struct device *dev, const char *id)
```

dev 为当前设备，id 对应 dts 里的设置的-supply 属性。

2. 释放 regulator

```
void regulator_put(struct regulator *regulator)
```

3. 打开 regulator

```
int regulator_enable(struct regulator *regulator)
```

4. 关闭 regulator

```
int regulator_disable(struct regulator *regulator)
```

5. 获取 regulator 电压

```
int regulator_get_voltage(struct regulator *regulator)
```

6. 设置 regulator 电压

```
int regulator_set_voltage(struct regulator *regulator, int min_uV, int max_uV)
```

传入的参数时保证 min_uV <= max_uV，由调用者保证。

7. 范例

```
struct regulator *vdd_ana;

vdd_ana = devm_regulator_get_optional(dev, "power");
/* 从dts获取power-supply
   power-supply = <&vcc3v3_lcd0_n> */
regulator_enable(vdd_ana);
regulator_disable(vdd_ana);           // 关闭vdd_ana
regulator_put(vdd_ana);                // 释放vdd_ana
```

3. Debug

3.1 内核

因为 PMIC 涉及的驱动在使用逻辑上都不复杂，重点都体现在最后的寄存器设置上。所以目前常用的 debug 方式可以通过如下节点：

```
/sys/kernel/debug/regulator
```

```
console:/ # ls /sys/kernel/debug/regulator/
avcc1v8_codec_s0      vcc5v0_sys           vdd_0v75_s3
avcc1v8_s0            vcc5v0_usb           vdd_0v85_s0
avdd1v8_ddr_pll_s0    vcc5v0_usbdcin       vdd1v8_pll_s0
avdd0v75_s0           vcc1v1_nldo_s3       vdd2v0_pldo_s3
avdd0v85_s0           vcc1v8_cam_s0        vdd_cpu_big0_mem_s0
avdd1v2_cam_s0        vcc1v8_s0            vdd_cpu_big0_s0
avdd1v2_s0            vcc2v8_cam_s0        vdd_cpu_big1_mem_s0
pcie20_avdd0v85       vcc3v3_s0            vdd_cpu_big1_s0
pcie20_avdd1v8        vcc3v3_s3            vdd_cpu_lit_mem_s0
pcie30_avdd0v75       vcc3v3_sd_s0         vdd_cpu_lit_s0
pcie30_avdd1v8        vcc_mipicsi0         vdd_ddr_pll_s0
pldo6_s3              vcc_mipicsi1         vdd_ddr_s0
reg-dummy-regulator-dummy vcc_mipidcphy0       vdd_gpu_mem_s0
regulator_summary     vccio1v8_s3          vdd_gpu_s0
supply_map            vccio_sd_s0          vdd_log_s0
vbus5v0_typec         vdd11v8_ddr_s3       vdd_npu_mem_s0
vcc12v_dc_in          vdd2_ddr_s3          vdd_npu_s0
vcc3v3_lcd0_n         vdd21_0v9_ddr_s3     vdd_vdenc_mem_s0
vcc3v3_pcie30         vdd_0v75_hdmi_edp_s0 vdd_vdenc_s0
vcc5v0_host           vdd_0v75_pll_s0      vddq_ddr_s0
```

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

| regulator | use | open | bypass | opmode | voltage | current | min | max |
|----------------------|-----|------|--------|---------|---------|---------|---------|---------|
| regulator-dummy | 5 | 5 | 0 | unknown | 0mV | 0mA | 0mV | 0mV |
| fe210000.sata-target | 1 | | | | | 0mA | 0mV | 0mV |
| fe210000.sata-phy | 1 | | | | | 0mA | 0mV | 0mV |
| fe210000.sata-ahci | 1 | | | | | 0mA | 0mV | 0mV |
| backlight-power | 1 | | | | | 0mA | 0mV | 0mV |
| regulator-dummy | 0 | | | | | 0mA | 0mV | 0mV |
| vcc12v_dcin | 3 | 4 | 0 | unknown | 12000mV | 0mA | 12000mV | 12000mV |
| vcc12v_dcin | 0 | | | | | 0mA | 0mV | 0mV |
| vcc5v0_sys | 30 | 30 | 0 | unknown | 5000mV | 0mA | 5000mV | 5000mV |
| vcc5v0_sys | 0 | | | | | 0mA | 0mV | 0mV |
| vdd_gpu_s0 | 1 | 3 | 0 | normal | 750mV | 0mA | 550mV | 950mV |
| fb000000.gpu-mali | 0 | | | | | 0mA | 750mV | 950mV |
| fb000000.gpu-mali | 0 | | | | | 0mA | 0mV | 0mV |
| vdd_gpu_s0 | 0 | | | | | 0mA | 0mV | 0mV |
| vdd_npu_s0 | 1 | 1 | 0 | normal | 750mV | 0mA | 550mV | 950mV |
| vdd_npu_s0 | 0 | | | | | 0mA | 0mV | 0mV |
| vdd_log_s0 | 1 | 1 | 0 | normal | 750mV | 0mA | 750mV | 750mV |
| vdd_log_s0 | 0 | | | | | 0mA | 0mV | 0mV |
| vdd_vdenc_s0 | 1 | 1 | 0 | normal | 750mV | 0mA | 550mV | 950mV |
| vdd_vdenc_s0 | 0 | | | | | 0mA | 0mV | 0mV |
| vdd_gpu_mem_s0 | 1 | 3 | 0 | normal | 750mV | 0mA | 675mV | 950mV |
| fb000000.gpu-mem | 0 | | | | | 0mA | 750mV | 950mV |
| fb000000.gpu-mem | 0 | | | | | 0mA | 0mV | 0mV |
| vdd_gpu_mem_s0 | 0 | | | | | 0mA | 0mV | 0mV |
| vdd_npu_mem_s0 | 1 | 1 | 0 | normal | 750mV | 0mA | 675mV | 950mV |
| vdd_npu_mem_s0 | 0 | | | | | 0mA | 0mV | 0mV |
| vdd_2v0_pldo_s3 | 9 | 9 | 0 | normal | 2000mV | 0mA | 2000mV | 2000mV |
| vdd_2v0_pldo_s3 | 0 | | | | | 0mA | 0mV | 0mV |
| avcc_1v8_s0 | 1 | 3 | 0 | unknown | 1800mV | 0mA | 1800mV | 1800mV |
| avcc_1v8_s0 | 0 | | | | | 0mA | 0mV | 0mV |
| pcie20_avdd1v8 | 0 | 1 | 0 | unknown | 1800mV | 0mA | 1800mV | 1800mV |
| pcie20_avdd1v8 | 0 | | | | | 0mA | 0mV | 0mV |
| pcie30_avdd1v8 | 0 | 1 | 0 | unknown | 1800mV | 0mA | 1800mV | 1800mV |
| pcie30_avdd1v8 | 0 | | | | | 0mA | 0mV | 0mV |
| vdd1_1v8_ddr_s3 | 1 | 1 | 0 | unknown | 1800mV | 0mA | 1800mV | 1800mV |
| vdd1_1v8_ddr_s3 | 0 | | | | | 0mA | 0mV | 0mV |
| avcc_1v8_codec_s0 | 1 | 1 | 0 | unknown | 1800mV | 0mA | 1800mV | 1800mV |
| avcc_1v8_codec_s0 | 0 | | | | | 0mA | 0mV | 0mV |
| avdd_1v2_cam_s0 | 1 | 1 | 0 | unknown | 1200mV | 0mA | 1200mV | 1200mV |
| avdd_1v2_cam_s0 | 0 | | | | | 0mA | 0mV | 0mV |
| avdd_1v2_s0 | 1 | 1 | 0 | unknown | 1200mV | 0mA | 1200mV | 1200mV |
| avdd_1v2_s0 | 0 | | | | | 0mA | 0mV | 0mV |
| vcc_1v8_cam_s0 | 1 | 1 | 0 | unknown | 1800mV | 0mA | 1800mV | 1800mV |
| vcc_1v8_cam_s0 | 0 | | | | | 0mA | 0mV | 0mV |
| avdd1v8_ddr_pll_s0 | 1 | 1 | 0 | unknown | 1800mV | 0mA | 1800mV | 1800mV |
| avdd1v8_ddr_pll_s0 | 0 | | | | | 0mA | 0mV | 0mV |
| vdd_1v8_pll_s0 | 1 | 1 | 0 | unknown | 1800mV | 0mA | 1800mV | 1800mV |
| vdd_1v8_pll_s0 | 0 | | | | | 0mA | 0mV | 0mV |
| vdd_vdenc_mem_s0 | 1 | 1 | 0 | normal | 750mV | 0mA | 675mV | 950mV |
| vdd_vdenc_mem_s0 | 0 | | | | | 0mA | 0mV | 0mV |
| vdd2_ddr_s3 | 1 | 1 | 0 | normal | 1100mV | 0mA | 0mV | 0mV |
| vdd2_ddr_s3 | 0 | | | | | 0mA | 0mV | 0mV |
| vcc_1v1_nldo_s3 | 6 | 6 | 0 | normal | 1100mV | 0mA | 1100mV | 1100mV |
| vcc_1v1_nldo_s3 | 0 | | | | | 0mA | 0mV | 0mV |
| avdd_0v75_s0 | 1 | 2 | 0 | unknown | 750mV | 0mA | 750mV | 750mV |

:

每一路电源对应的属性及电压值可通过以下节点查找

```
console:/sys/kernel/debug/regulator/vdd_gpu_mem_s0 # ls
bypass_count  fb000000.gpu-mem  mode            vdd_gpu_mem_s0
consumers     force_disable     open_count      voltage
enable        load              use_count
```

3.2 Kernel 5.10 内核

请参考5.10内核命令。