带RK816的RK3308开发指南

发布版本：1.0

作者邮箱：linjh@rock-chips.com

日期：2019.04

文件密级：公开资料

前言

概述

RK816 是一款高性能 PMIC，集成了多路大电流DCDC，多个LDO，1个线性开关，1个USB（5V）及 boost输出，还有开关充电，智能功率路径管理，库仑计，RTC 及可调上电时序等功能。对于RK816本身有详细的文档介绍，本文档仅对RK816应用配置进行简要介绍，旨在让读者理解RK3308如何使用RK816。

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

产品版本

芯片名称	内核版本
RK3308/RK816-3	4.4

修订记录

日期	版本	作者	修改说明
2019-04-26	V1.0	林建华	初始版本

带RK816的RK3308开发指南
- uboot
  - 1. 32位配置
  - 2. 64位配置
  - 3.打印信息识别
- kernel
  - 1. 32位配置
  - 2. 64位配置
  - 3. dts使能
    - 3.1 增加pmic pinctrl
    - 3.2 增加rk816节点

- 4. 参考板级配置
- uboot关机充电功能
  - 1. 内核板级dts配置
  - 2. 打印信息识别
  - 3. 充电灯控制
    - 3.1.dts配置
    - 3.2. uboot打印识别
    - 3.3.内核控制led灯
- 附录 RK816文档参考

uboot

1. 32位配置

使用rk3308-aarch32_defconfig

2. 64位配置

使用rk3308_defconfig

3.打印信息识别

```
U-Boot 2017.09-02378-gabe0aae (Apr 15 2019 - 14:37:44 +0800)

Model: Rockchip RK3308 EVB
PreSerial: 2
DRAM:  126 MiB
Relocation Offset is: 0791e000
Using default environment
Bootdev(atags): rknand 0
No.1 FLASH ID:c8 d1 80 95 42 7f
SFTL version: 5.0.50 20181227
PartType: EFI
boot mode: None
Load FDT from boot part
DTB: rk-kernel.dtb
I2c speed: 400000Hz
PMIC:  RK8160 (on=0x03, off=0x06)
vdd_core 1100000 uV
vdd_logic 1100000 uV
```

kernel

1. 32位配置

rk3308_linux_aarch32_debug_defconfig默认打开rk816的配置，因为考虑到有些特定的项目对固件的大小和开机速度有很严格的要求，所以没有把rk816的配置加到rk3308_linux_aarch32_defconfig文件里，如果需要用rk3308_linux_aarch32_defconfig，把如下的配置加上并且把USB相关的配置成m的都改成y。

```
menuconfig里对应的宏配置:

CONFIG_INPUT_RK8XX_PWRKEY
CONFIG_BATTERY_RK816
CONFIG_MFD_RK808
CONFIG_REGULATOR_RK808
CONFIG_RTC_DRV_RK808
CONFIG_BATTERY_RK816
```

2. 64位配置

rk3308_linux_defconfig

3. dts使能

3.1 增加pmic pinctrl

```
&pinctrl {
     pmic {
          pmic_int_l: pmic-int-l {
          rockchip,pins =
              <0 6 RK_FUNC_GPIO &pcfg_pull_up>;
          };
     };
};
```

3.2 增加rk816节点

```
&i2c1 {
        status = "okay";
        clock-frequency = <400000>;
        rk816: pmic@1a {
                status = "okay";
                compatible = "rockchip,rk816";
                reg = <0x1a>;
                interrupt-parent = <&gpio0>;
                interrupts = <6 IRQ_TYPE_LEVEL_LOW>;
                pinctrl-names = "default";
                pinctrl-0 = <&pmic_int_l>;
                rockchip,system-power-controller;
                wakeup-source;
                gpio-controller;
                #gpio-cells = <2>;
                #clock-cells = <1>;
                clock-output-names = "rk816-clkout1", "rk816-clkout2";
                extcon = <&u2phy>;
                vcc1-supply = <&vcc_sys>;
                vcc2-supply = <&vcc_sys>;
                vcc3-supply = <&vcc_sys>;
                vcc4-supply = <&vcc_sys>;
                vcc5-supply = <&vcc_io>;
                vcc6-supply = <&vcc_sys>;
                pwrkey {
                        status = "okay";
                };
                gpio {
                        status = "okay";
                };
                rtc {
                        status = "okay";
                };
                battery {
                        compatible = "rk816-battery";
                        ocv_table = <3500 3625 3685 3697 3718 3735 3748
                                     3760 3774 3788 3802 3816 3834 3853
                                     3877 3908 3946 3975 4018 4071 4106>;
                        design_capacity = <5000>;   //电池容量
                        design_qmax = <5200>;
                        bat_res = <100>;
                        max_input_current = <2000>; //最大输入电流
                        max_chrg_current = <1300>;  //最大充电电流
                        max_chrg_voltage = <4200>;  //最大充电电压
                        sleep_enter_current = <300>;
                        sleep_exit_current = <300>;
                        sleep_filter_current = <100>;
                        power_off_thresd = <3500>;    //低电关机电压
                        zero_algorithm_vol = <3850>;
                        max_soc_offset = <80>;
                        monitor_sec = <5>;
                        virtual_power = <0>;        //开发阶段没有接电池，这里配成1
                        power_dc2otg = <0>;
                        dc_det_adc = <0>;
                };
                regulators {
                        vdd_core: DCDC_REG1{
                                regulator-name = "vdd_core";
                                regulator-min-microvolt = <750000>;
                                regulator-max-microvolt = <1500000>;
                                regulator-ramp-delay = <6001>;
                                regulator-initial-mode = <1>;
                                regulator-always-on;
                                regulator-boot-on;
                                regulator-state-mem {
                                        regulator-off-in-suspend;
                                };
                        };
                        vdd_log: DCDC_REG2 {
                                regulator-name = "vdd_logic";
                                regulator-min-microvolt = <750000>;
                                regulator-max-microvolt = <1500000>;
```

```
                                        regulator-ramp-delay = <6001>;
                                        regulator-initial-mode = <1>;
                                        regulator-always-on;
                                        regulator-boot-on;
                                        regulator-state-mem {
                                                regulator-on-in-suspend;
                                                regulator-suspend-microvolt = <1000000>;
                                        };
                                };
                                vcc_ddr: DCDC_REG3 {
                                        regulator-name = "vcc_ddr";
                                        regulator-always-on;
                                        regulator-boot-on;
                                        regulator-initial-mode = <1>;
                                };
                                vcc_io: DCDC_REG4 {
                                        regulator-name = "vcc_io";
                                        regulator-min-microvolt = <3300000>;
                                        regulator-max-microvolt = <3300000>;
                                        regulator-initial-mode = <1>;
                                        regulator-always-on;
                                        regulator-boot-on;
                                        regulator-state-mem {
                                                regulator-on-in-suspend;
                                                regulator-suspend-microvolt = <3000000>;
                                        };
                                };
                                 vcc_1v8_codec: LDO_REG1 {
                                        regulator-name = "vcc_1v8_codec";
                                        regulator-min-microvolt = <1800000>;
                                        regulator-max-microvolt = <1800000>;
                                        regulator-always-on;
                                        regulator-boot-on;
                                        regulator-state-mem {
                                                regulator-on-in-suspend;
                                                regulator-suspend-microvolt = <1800000>;
                                        };
                                };
                                vccio_sdio: vcc_1v8: LDO_REG2 {
                                        regulator-name = "vcc_1v8";
                                        regulator-min-microvolt = <1800000>;
                                        regulator-max-microvolt = <1800000>;
                                        regulator-always-on;
                                        regulator-boot-on;
                                        regulator-state-mem {
                                                regulator-on-in-suspend;
                                                regulator-suspend-microvolt = <1800000>;
                                        };
                                };
                                vdd_10: LDO_REG3 {
                                        regulator-name = "vdd_10";
                                        regulator-min-microvolt = <1000000>;
                                        regulator-max-microvolt = <1800000>;
                                        regulator-state-mem {
                                                regulator-off-in-suspend;
                                        };
                                };
                                ldo4: LDO_REG4 {
                                        regulator-name = "ldo4";
                                        regulator-min-microvolt = <1800000>;
                                        regulator-max-microvolt = <3300000>;
                                        regulator-state-mem {
                                                regulator-off-in-suspend;
                                        };
                                };
                                vccio_3v3: LDO_REG5 {
                                        regulator-name = "vccio_3v3";
                                        regulator-min-microvolt = <1800000>;
                                        regulator-max-microvolt = <3300000>;
                                        regulator-state-mem {
                                                regulator-off-in-suspend;
                                        };
                                };
                                ldo6: LDO_REG6 {
                                        regulator-name = "ldo6";
                                        regulator-min-microvolt = <1800000>;
                                        regulator-max-microvolt = <3300000>;
                                        regulator-state-mem {
                                                regulator-off-in-suspend;
                                        };
                                };
                        };
                };
};
```

a) i2c挂载

整个完整的rk816节点挂在对应的i2c节点下面，并且配置status = "okay";

b) 主体部分

不可修改:

```
compatible = "rockchip,rk816";
reg = <0x1a>;
rockchip,system-power-controller;
wakeup-source;
gpio-controller;
#gpio-cells = <2>;
```

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;
pinctrl-names:不修改,固定为 "default";
pinctrl-0:引用pinctrl里定义好的pmic_int引脚;
```

c) rtc、pwrkey、gpio

如果menuconfig选中了这几个模块，但是实际又不需要使能这几个驱动，那么可以在dts里增加rtc、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；如果要使能驱动，则可以去掉相应的节点,或者设置状态为status = "okay"。

d) regulator

```
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-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-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。
```

4. 参考板级配置

RK3308的sdk中rk3308-dot-rk816-v10开发板是用rk186的pmic，板级配置在如下目录，64位的配置是通过导入32位的板级配置实现。

32位：arch/arm/boot/dts/rk3308-dot-rk816-v10-aarch32.dts

64位：arch/arm64/boot/dts/rockchip/rk3308-dot-rk816-v10.dts

---

uboot关机充电功能

RK816可以实现关机充电功能，通过在kernel板级dts配置charge-animation节点中增加rockchip,uboot-charge-on = <1>；属性实现。

说明: 使用rk3308-aarch32_defconfig和rk3308_defconfig配置的Uboot，在启动第二阶段是使用内核的dtb。

1. 内核板级dts配置

```
charge-animation {
    compatible = "rockchip,uboot-charge";
    rockchip,uboot-charge-on = <1>;
    rockchip,uboot-low-power-voltage = <3500>;
    status = "okay";
};
rockchip,uboot-low-power-voltage = <3500>; 该属性实现低电不开机功能，当电池低于3.5V不能开机。
```

2. 打印信息识别

```
Fuel gauge initialize = 1
rk816-bat: fuel gauge initialized... estv=4104, ch=1
rk816-bat: first power on: soc=96
Battery: soc=96%, cap=2635mAh, voltage=4114mv, Charger: DC
Enable charge animation display
Enter U-Boot charging mode
```

3. 充电灯控制

在Uboot充电模式 下还实现充电灯的控制，目前实现了充电状态和充满状态两种led灯常亮效果。只支持通过gpio控制led灯。

3.1.dts配置

Uboot在初始化的时候，会解析内核板级dts的Leds节点，通过label属性值来判断是否定义了battery_full（充满）和battery_charging（充电）led灯控制。

```
        leds {
                status = "okay";
                compatible = "gpio-leds";
                blue-led {
                        gpios = <&gpio2 RK_PA1 GPIO_ACTIVE_LOW>;
                        label = "battery_full";
                        default-state = "off";
                        linux,default-trigger = "battery-full";
                };
                red-led {
                        gpios = <&gpio2 RK_PA2 GPIO_ACTIVE_LOW>;
                        label = "battery_charging";
                        default-state = "off";
                        linux,default-trigger = "battery-charging";
                };
        };


```

3.2. uboot打印识别

```
    Found Charging LED
    Found Charging-Full LED
```

3.3.内核控制led灯

内核也实现充电灯的控制，但是内核是通过解析leds的“linux,default-trigger”属性来控制充电状态（battery-charging）和充满状态（battery-full）led灯。

附录 RK816文档参考

sdk\ docs\Develop reference documents\PMIC