

# RK3308 系统待机配置指南

---

发布版本：1.1

作者邮箱：[chenjh@rock-chips.com](mailto:chenjh@rock-chips.com)

日期：2019.11

文件密级：公开资料

## 前言

## 概述

本文档用于指导用户如何根据产品需求，配置 RK3308 系统待机模式。

## 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	4.4、4.19

## 修订记录

日期	版本	作者	修改说明
2019-05-01	V1.0	陈健洪	初始版本
2019-11-11	V1.1	陈健洪	支持列表：增加4.19内核

## **RK3308 系统待机配置指南**

1. 系统待机
  - 1.1 驱动文件
  - 1.2 DTS 节点
2. DTS 配置
  - 2.1 常规配置
  - 2.2 电源配置
  - 2.3 唤醒配置
  - 2.4 debug 配置
  - 2.5 reboot 复位配置
3. 打印信息

# 1. 系统待机

凡是带有 trust 的 SoC 平台，系统待机（system suspend）的工作都在 trust 中完成。因为各个平台的 trust 对于系统待机实现各不相同，所以不同平台之间的待机配置选项/方法没有任何关联性和参考性，本文档仅适用于 RK3308 平台。

系统待机流程一般会有如下操作：关闭 power domain、模块 IP、时钟、PLL、ddr 进入自刷新、系统总线切到低速时钟（24M 或 32K）、vdd\_arm 断电、配置唤醒源等。为了满足不同产品对待机模式的需求，目前都是通过 DTS 节点把相关配置在开机阶段传递给 trust。

## 1.1 驱动文件

```
./drivers/soc/rockchip/rockchip_pm_config.c
./drivers/firmware/rockchip_sip.c
./include/dt-bindings/suspend/rockchip-rk3308.h
```

## 1.2 DTS 节点

```
rockchip_suspend: rockchip-suspend {
    compatible = "rockchip,pm-rk3308";
    status = "okay";
    // 常规配置
    rockchip,sleep-mode-config = <
        (0
            | RKPM_PMU_HW_PLLS_PD
            | RKPM_DBG_FSM_SOUT
        )
    >;
    // 唤醒源配置
    rockchip,wakeup-config = <
        (0
            | RKPM_GPIO0_WAKEUP_EN
        )
    >;
    // 电源配置
    rockchip,pwm-regulator-config = <
        (0
            | RKPM_xxx
        )
    >;
    // reboot复位配置
    rockchip,apios-suspend = <
        (0
            | RKPM_xxx
        )
    >;
};
```

## 2. DTS 配置

目前已支持的配置选项都定义在：

```
./include/dt-bindings/suspend/rockchip-rk3308.h
```

### 2.1 常规配置

配置项：

```
rockchip,sleep-mode-config = <...>;
```

配置源：

```
// 断电vdd_arm, 需要硬件电路设计上能支持
#define RKPM_ARMOFF BIT(0)
// 关闭vad模块, 不需要VAD唤醒时可关闭
#define RKPM_VADOFF BIT(1)
// 默认必选
#define RKPM_PMU_HW_PLLS_PD BIT(3)
// 关闭24M晶振, 最低功耗模式时可使能, 需要配合选中下面的xxx_32K时钟源配置
#define RKPM_PMU_DIS_OSC BIT(4)
// 使用PMU内部的32K时钟源作为系统时钟(相比外部32K时钟, 推荐此方法)
#define RKPM_PMU_PMUALIVE_32K BIT(5)
// 使用外部32K晶振作为系统时钟, 不推荐
#define RKPM_PMU_EXT_32K BIT(6)
// 默认不选
#define RKPM_DDR_SREF_HARDWARE BIT(7)
// 默认不选
#define RKPM_DDR_EXIT_SRPD_IDLE BIT(8)
// RKPM_ARMOFF的情况下关闭PDM的clk时钟
#define RKPM_PDM_CLK_OFF BIT(9)
// 待机时pwm-regulator设置和maskrom一样的电压(否则会使用更低的电压, 目前仅宽温芯片需要选择此项)
#define RKPM_PWM_VOLTAGE_DEFAULT BIT(10)
```

目前 RK3308 支持的待机模式可分为 2 类：

- VAD 类产品：待机时需要支持 VAD 唤醒源，不会关闭 VAD/ACODEC/PDM 等相关模块 IP 和时钟，需保持 24M 晶振和相关 PLL 正常工作。目前待机时 trust 会先检测 VAD 相关模式是否在 kernel 阶段已经关闭，如果没有关闭则默认是 VAD 类产品，待机时切到支持 VAD 唤醒的低功耗模式。
- 非 VAD 类产品：待机时没有需要维持工作的模块 IP，所有的模块和时钟几乎都可以关闭，是一种最低功耗的模式。这种模式下，系统时钟可以切到 32K 或者 24M。

### 2.2 电源配置

配置项：

```
rockchip,pwm-regulator-config = <...>;
```

配置源:

```
// 使用pwm-regulator
#define RKPM_PWM_REGULATOR BIT(2)
```

电源注意点:

- 根据外部硬件电路设计确定是否使用了 pwm-regulator。

## 2.3 唤醒配置

配置项:

```
rockchip,wakeup-config = <...>;
```

配置源:

```
// 支持所有的中断唤醒（不经过GIC管理），不推荐使用
#define RKPM_ARM_PRE_WAKEUP_EN BIT(11)
// 支持所有的中断唤醒（经过GIC管理的休眠可唤醒中断）
#define RKPM_ARM_GIC_WAKEUP_EN BIT(12)
// SDMMC唤醒
#define RKPM_SDMMC_WAKEUP_EN BIT(13)
#define RKPM_SDMMC_GRF_IRQ_WAKEUP_EN BIT(14)
// RK TIMER唤醒
#define RKPM_TIMER_WAKEUP_EN BIT(15)
// USB唤醒
#define RKPM_USBDEV_WAKEUP_EN BIT(16)
// PMU内部timer唤醒（默认5s），一般用于测试休眠唤醒使用
#define RKPM_TIMEOUT_WAKEUP_EN BIT(17)
// GPIO0唤醒
#define RKPM_GPIO0_WAKEUP_EN BIT(18)
// VAD唤醒
#define RKPM_VAD_WAKEUP_EN BIT(19)
```

唤醒源注意点:

- RKPM\_GPIO0\_WAKEUP\_EN（首选）：  
GPIO0~3 中仅支持 GPIO0 这组 pin 脚作为唤醒源，该模式下 GPIO0 上的 pin 脚中断信号被直接送往 PMU 状态机，不经过 GIC。在硬件设计上，建议用户把需要的唤醒源尽量都放到 GPIO0 这组 pin 脚上。
- RKPM\_ARM\_GIC\_WAKEUP\_EN（次选）：  
支持所有在 kernel 阶段用 enable\_irq\_wake()注册到 GIC 的可唤醒中断，适用的唤醒中断源数量比 RKPM\_GPIO0\_WAKEUP\_EN 更多。但这种方式相当于把唤醒源的管理权分散交给了 kernel 各个模块，待机时系统有可能被不期望的中断唤醒。
- RKPM\_TIMEOUT\_WAKEUP\_EN：  
PMU 内部的 timer 唤醒，默认 5s 超时产生中断，一般仅用于开发阶段测试休眠唤醒使用。

## 2.4 debug 配置

配置项:

```
rockchip,sleep-mode-config = <...>;
```

配置源:

```
// 忽略
#define RKPM_DBG_INT_TIMER_TEST      BIT(22)
#define RKPM_DBG_WOARKAROUND        BIT(23)
#define RKPM_DBG_VAD_INT_OFF        BIT(24)
// 休眠时常开所有的clk
#define RKPM_DBG_CLK_UNGATE          BIT(25)
// 忽略
#define RKPM_DBG_CLKOUT              BIT(26)
// PMU状态机信号输出
#define RKPM_DBG_FSM_SOUT            BIT(27)
// 忽略
#define RKPM_DBG_FSM_STATE           BIT(28)
// DUMP某些寄存器: gpio/grf/sgrf...
#define RKPM_DBG_REG                 BIT(29)
// 忽略
#define RKPM_DBG_VERBOSE             BIT(30)
#define RKPM_CONFIG_WAKEUP_END       BIT(31)
```

debug 注意点:

- **RKPM\_DBG\_CLK\_UNGATE**: 如果怀疑待机阶段某些 clk 被关闭而引起系统/模块唤醒异常, 可启用该配置。
- **RKPM\_DBG\_REG**: 如果怀疑待机阶段某些寄存器值被 trust 修改, 可启用该配置。
- **RKPM\_DBG\_FSM\_SOUT**: 启用该配置后, 待机时 PMU 状态机会通过 GPIO4\_D5 一直输出特定波形信号, 用于反馈当前 PMU 状态机内部状态, 该功能仅在发生系统待机时 PMU 状态机本身死机的情况下有用处。

## 2.5 reboot 复位配置

配置项:

```
rockchip,apios-suspend = <...>;
```

配置源:

```
#define GLB1RST_IGNORE_PWM0         BIT(23)
#define GLB1RST_IGNORE_PWM1         BIT(24)
#define GLB1RST_IGNORE_PWM2         BIT(25)
#define GLB1RST_IGNORE_GPIO0        BIT(26)
#define GLB1RST_IGNORE_GPIO1        BIT(27)
#define GLB1RST_IGNORE_GPIO2        BIT(28)
#define GLB1RST_IGNORE_GPIO3        BIT(29)
```

reboot 复位注意点:

目前 RK3308 默认使用的是 first global software reset, reboot 时所有模块 IP 都会被复位。如果需要保持某些 IP 不被复位, 那么需要配置上面的选项, 目前支持: pwm0~3/gpio0~3 不复位。

GPIO 不复位的需求示例:

某些硬件电路设计上会提供“power hold”电源控制引脚, 需要在系统上电早期阶段由软件拉高/低保证系统电源工作正常, 在 reboot 过程中“power hold”引脚也不能被复位, 否则会出现系统下电的情况。

## 3. 打印信息

如下简要介绍系统待机和唤醒时的 trust 打印信息含义。为注释方便, 如下对一些打印内容进行分行, 不同的待机功耗模式同样也会带来不同的打印, 所有打印信息内容以实际显示为主。

**RK3308 系统待机打印:**

```
// 具备当唤醒源的pin脚
GPIO0_INTEN: 00000041
GPIO4_INTEN: 00001000
// kernel配置信息打印
v1.3(release):005c64b, cnt=1, config=0x8040009:armoff-hwplldown-ddrsw-gating-
24M-sout-
// 休眠流程步骤打印: 每个字符都代表trust里的一个休眠步骤
0123a4
// vad相关模块的使能状态, 1: 使能, 0: 关闭
Enabling: vad(1) acodec(1) pdm(0) i2s_2(1)
// 各个模块占用的PLL情况, "Enabling"表示系统正在使用的PLL(因为不一定所有PLL都会开着)
DDR: vp110 | VOICE(sum): vp110 | I2S: vp110 | PWM: dp11 | Enabling: ap11 dp11
vp110 | CRU_MODE: 3955
// "Disabling"表示待机时会被关闭的PLL
PMU Disabling: ap11 dp11 vp111
// PMU寄存器值(忽略)。“24Mhz”表示当前的系统时钟, 如果当前是32K情况, 则打印也随之变化体现
PMU: pd-000e wake-000c core-0bfb lo-180d hi-000e if-4001 24Mhz
// 休眠流程步骤打印。注意: 打印完“wfi”就表示系统已经完全待机下去了!
5bRc678wfi
```

**RK3308 系统唤醒打印:**

```
// 唤醒流程步骤打印
876ab543210
// 唤醒源
IRQ=89
PMU wakeup int: vad
VAD int=00000113
// 本次系统深度待机时长
Wfi total: 2.419s(this: 2.419s)
```