

RK3588 系统待机配置指南

文件标识: RK-KF-YF-452

发布版本: V1.0.0

日期: 2022-01-20

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 © 2022 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

本文档用于指导用户如何根据产品需求, 配置 RK3588 系统待机模式。

产品版本

芯片名称	内核版本
RK3588	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	黄小东	2022-01-20	初始版本

目录

RK3588 系统待机配置指南

系统待机

驱动文件

DTS 节点

DTS 配置

常规配置

唤醒配置

debug 配置

打印信息

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 RK3588 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 /vdd_log 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。

驱动文件

```
./drivers/soc/rockchip/rockchip_pm_config.c
./drivers/firmware/rockchip_sip.c
./include/dt-bindings/suspend/rockchip-rk3588.h
```

DTS 节点

```
rockchip_suspend: rockchip-suspend {
    compatible = "rockchip,pm-rk3588";
    status = "okay";
    // 休眠log开关配置, 0: 关闭打印, 1: 打开打印
    rockchip,sleep-debug-en = <1>;
    // 常规配置
    rockchip,sleep-mode-config = <
        0
        | RKPM_SLP_ARMOFF_DDRPD
        | RKPM_SLP_PMU_PMUALIVE_32K
        | RKPM_SLP_PMU_DIS_OSC
```

```

    | RKPM_SLP_32K_EXT
    | RKPM_SLP_PMU_DBG
)
>;
// 唤醒源配置
rockchip,wakeup-config = <
(0
    | RKPM_GPIO_WKUP_EN
)
>;
};

```

DTS 配置

目前已支持的配置选项都定义在:

```
./include/dt-bindings/suspend/rockchip-rk3588.h
```

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_SLP_ARMOFF BIT(1)
// 断电vdd_arm, 且DDR控制器断电, 需要硬件电路设计上能支持
#define RKPM_SLP_ARMOFF_DDRPD BIT(2)
// 断电vdd_arm和vdd_log, 需要硬件电路设计上能支持
#define RKPM_SLP_ARMOFF_LOGOFF BIT(3)
// 断电vdd_arm和vdd_log, 且PMU1电源域断电, 需要硬件电路设计上能支持
#define RKPM_SLP_ARMOFF_PMUOFF BIT(4)

// 休眠时使用32K时钟源作为系统时钟
#define RKPM_SLP_PMU_PMUALIVE_32K BIT(9)
// 关闭24M晶振, 最低功耗模式时可使能, 需要配合RKPM_SLP_PMU_PMUALIVE_32K使用
#define RKPM_SLP_PMU_DIS_OSC BIT(10)
// 休眠时的32K时钟源是否选用外部的32K钟源, 不配该选项则默认选用内部32K时钟源, 需要配合
RKPM_SLP_PMU_PMUALIVE_32K使用
#define RKPM_SLP_32K_EXT BIT(24)

```

注意事项:

- 需要根据具体产品对唤醒源的需求进行相关配置, 比如usb唤醒, 那休眠时就不能将usb的电源和时钟关闭, 所以不能配置RKPM_SLP_ARMOFF_LOGOFF、RKPM_SLP_PMU_DIS_OSC、RKPM_SLP_PMU_PMUALIVE_32K等选项。

唤醒配置

配置项:

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

配置源:

```
// 支持所有的中断唤醒（经过GIC管理的休眠可唤醒中断），一般由cpu0运行休眠流程，需要时只配
RKPM_CPU0_WKUP_EN即可
#define RKPM_CPU0_WKUP_EN BIT(0)
#define RKPM_CPU1_WKUP_EN BIT(1)
#define RKPM_CPU2_WKUP_EN BIT(2)
#define RKPM_CPU3_WKUP_EN BIT(3)
#define RKPM_CPU4_WKUP_EN BIT(4)
#define RKPM_CPU5_WKUP_EN BIT(5)
#define RKPM_CPU6_WKUP_EN BIT(6)
#define RKPM_CPU7_WKUP_EN BIT(7)
// GPIO0唤醒
#define RKPM_GPIO_WKUP_EN BIT(8)
// SDMMC唤醒
#define RKPM_SDMMC_WKUP_EN BIT(9)
// SDIO唤醒
#define RKPM_SDIO_WKUP_EN BIT(10)
// USB DEV 唤醒
#define RKPM_USB_WKUP_EN BIT(11)
// UART0唤醒
#define RKPM_UART0_WKUP_EN BIT(12)
// VAD唤醒
#define RKPM_VAD_WKUP_EN BIT(13)
// RK TIMER 唤醒
#define RKPM_TIMER_WKUP_EN BIT(14)
// 支持所有的中断唤醒（不经过GIC管理），不推荐使用
#define RKPM_SYSINT_WKUP_EN BIT(15)
// PMU内部timer唤醒（默认1s），用于测试和debug
#define RKPM_TIME_OUT_WKUP_EN BIT(16)
```

唤醒源注意事项:

- RKPM_GPIO_WKUP_EN (首选) :
GPIO0~4 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_CPU0_WKUP_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_GPIO_WKUP_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 1s 超时产生中断，一般仅用于开发阶段测试休眠唤醒使用。

debug 配置

配置项:

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

配置源:

```
#define RKPM_SLP_TIME_OUT_WKUP      BIT(25)
#define RKPM_SLP_PMU_DBG             BIT(26)
```

debug 注意点:

- RKPM_SLP_TIME_OUT_WKUP: 使能该配置后, 待机后1s左右会自动唤醒, 且只有pmu内部 timer才能唤醒系统, 该配置只用于休眠唤醒测试和debug。
- RKPM_SLP_PMU_DBG: 使能该配置后, 待机时 PMU 状态机会通过 GPIO0_A5 一直输出特定波形信号, 用于反馈当前 PMU 状态机内部状态, 该功能仅用于休眠唤醒测试和debug。

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。

配置项:

```
rockchip,sleep-debug-en = <...>;
```

- 0: 休眠时不会打印log, 1: 休眠时会打印log。

RK3588 系统待机打印:

```
//休眠所用trust bl31版本及commit信息
INFO:    BL31: v2.3():v2.3-264-g378cb8595:derrick.huang

//休眠模式配置及休眠次数打印
INFO:    enter: cfg=0x5000604, sleeptimes:1

//休眠模式打印
INFO:    armoff_ddrpd
INFO:    pmu_pmualive_32k
INFO:    pmu_dis_osc
INFO:    32k ext
INFO:    pmu debug

//休眠gpio中断配置状态打印
INFO:    GPIO0_INTEN: 0xffff 0xffff 0xff7f 0xffff 0x0 0xc81e142d
INFO:    GPIO1_INTEN: 0xffff 0xffff 0xffff 0xffff 0x0 0xe82863
INFO:    GPIO2_INTEN: 0xffff 0xffff 0xffff 0xffff 0x0 0xffefcef7
INFO:    GPIO3_INTEN: 0xffff 0xffff 0xfffe 0xffff 0x0 0xf0044483
INFO:    GPIO4_INTEN: 0xffff 0xffff 0xffff 0xffff 0x0 0xe0dc2003

//休眠关键寄存器信息打印
INFO:    PMU1_PWR_CON(0x1) PMU1_CRU_PWR_CON(0x23) PMU1_WAKEUP_INT_CON(0x100)
PMU2_BUS_IDLE_ST(0x27ffffff 0x0) PMU2_BUS_IDLE_ACK(0x27ffffff 0x0)
PMU2_PWR_GATE_ST(0x67ffffff 0x0)
PMU2_BUS_IDLE_CON(0x0 0xfd80 0xf007) PMU2_BIU_AUTO_CON(0xffff 0xffff 0x7)
PMU2_PWR_GATE_CON(0x0 0x9000 0x3)
PMU2_VOL_GATE_CON(0x7 0x0 0x3)
PMU2_QCHANNEL_PWR_CON(0x0) PMU2_QCHANNEL_STATUS(0xfe0007f)
PMU1_DDR_PWR_CON(0x747 0x747 0x747 0x747)
PMU1_DDR_PWR_SFTCON(0x900 0x900 0x900 0x900)
PMU1_PLLPD_CON(0xffff 0x3)
PMU2_DSU_PWR_CON(0x3)
PMU2_CORE_PWR_CON0(0x1 0x1)
```

```
PMU2_CORE_AUTO_PWR_CON(0x0 0x0)
PMU2_CLUSTER_IDLE_CON(0x75)
INFO:    PMU0_PWR_CON(0x0) PMU0_WAKEUP_INT_CON(0x0)
PMU0_DDR_RET_CON(0x0 0x0)
PMU1_GRF_SOC_CON2(0x7777) PMU0_GRF_OS_REGS9(0xd8394dc7)
```

RK3588 系统唤醒打印:

```
// 唤醒流程步骤打印
012376543edcba2

// 唤醒源
INFO:    gpio0_a7
INFO:    wake up status: 0x100
INFO:    the wake up information:
INFO:    GPIO0 interrupt wakeup
INFO:    GPIO0: 0x80
```