

# RK3399Pro IO 电源域配置说明

---

文档标识: RK-SM-YF-910

发布版本: V1.0.0

日期: 2021-05-15

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](http://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](mailto:fae@rock-chips.com)

---

## 前言

### 概述

主控电源域的IO电平要与对接外设芯片的IO电平保持一致, 还要注意软件的电压配置要跟硬件的电压一致, 否则, 最坏的情况可能会导致IO的损坏。

本文主要描述了RK3399Pro平台Linux SDK配置IO电源域的方法, 旨在帮助开发者正确配置IO的电源域。

### 产品版本

芯片名称	内核版本
RK3399Pro	Linux 4.4

## 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	Caesar Wang	2021-05-15	初始版本

## 目录

### RK3399Pro IO 电源域配置说明

- 第一步：获取硬件原理图并确认硬件电源的设计方案
- 第二步：查找对应的内核dts配置文件
- 第三步：修改内核dts的电源域配置节点pmu\_io\_domains
- 第四步：SDK查看当前固件电源域配置
- 第五步：烧录固件后确认寄存器值是否正确

## 第一步：获取硬件原理图并确认硬件电源的设计方案

本文以RK\_EVB\_RK3399PRO\_LP3S178P332SD8\_V14\_20191108 EVB板为例进行介绍。

硬件原理图：RK\_EVB\_RK3399PRO\_LP3S178P332SD8\_V14\_20191108.pdf

电源方案：从硬件原理图分析，EVB板RK\_EVB\_RK3399PRO\_LP3S178P332SD8\_V14\_20191108是带PMU (RK809-3) 方案。

## 第二步：查找对应的内核dts配置文件

由第一步可知，该EVB板的硬件电源设计是带PMU方案的，对应的内核dts配置文件位于：  
arch/arm64/boot/dts/rockchip/rk3399pro-evb-v11.dtsi（本文讨论的方案）

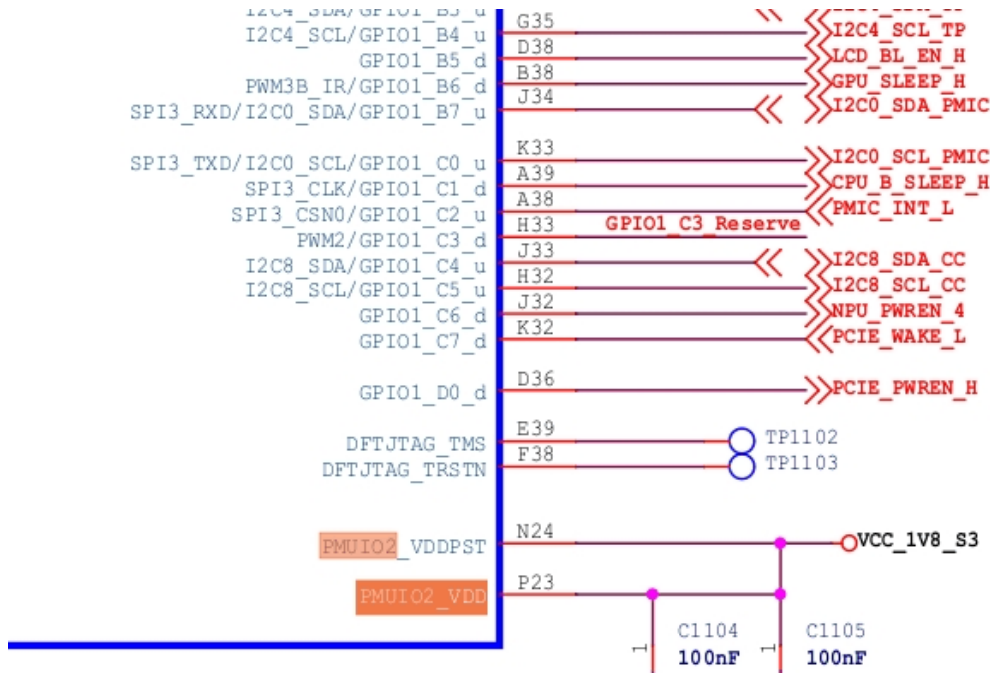
## 第三步：修改内核dts的电源域配置节点pmu\_io\_domains

```
&io_domains {
    status = "okay";
    bt656-supply = <&vcca_1v8>;
    audio-supply = <&vcca_1v8>;
    sdmmc-supply = <&vccio_sd>;
    gpio1830-supply = <&vcc_3v0>;
};

&pmu_io_domains {
    status = "okay";
    pmu1830-supply = <&vcc_1v8>;
};
```

以pmu1830-supply为例，首先查看硬件原理图确认pmuio2电源域（PMUIO2）的配置如图所示。

PMUIO2配置的电源域为VCC\_1V8\_S3（即1.8v）。



其中bt656-supply对应硬件原理图上是APIO2\_VDD，  
audio-supply对应硬件原理图上是APIO5\_VDD，  
gpio1830-supply对应硬件原理图上是APIO4\_VDD。

## 第四步：SDK查看当前固件电源域配置

命令：`./build.sh info`

```
PLEASE CHECK BOARD GPIO POWER DOMAIN CONFIGURATION !!!!!
<<< ESPECIALLY WI-FI/Flash/Ethernet IO power domain >>> !!!!!
Check Node [pmu_io_domains] in the file: /home/wxt/linux-develop/rk3399pro/kernel/arch/arm64/boot/dts/rockchip/rk3399pro-evb-v14-linux.dts

请再次确认板级的电源域配置!!!!!!
<<< 特别是WI-FI, FLASH, 以太网这几路io电源的配置 >>> !!!!!!!
检查内核文件 /home/wxt/linux-develop/rk3399pro/kernel/arch/arm64/boot/dts/rockchip/rk3399pro-evb-v14-linux.dts 的节点 [pmu_io_domains]

pmu1830-supply
regulator-min-microvolt = 1800mV
regulator-max-microvolt = 1800mV

bt656-supply
regulator-min-microvolt = 1800mV
regulator-max-microvolt = 1800mV

audio-supply
regulator-min-microvolt = 1800mV
regulator-max-microvolt = 1800mV

sdmmc-supply
regulator-min-microvolt = 1800mV
regulator-max-microvolt = 3300mV
```

## 第五步：烧录固件后确认寄存器值是否正确

以RK3399Pro芯片为例，根据手册获取PMU\_SOC\_CON0寄存器（0xFF320180）和GRF\_IO\_VSEL寄存器（0xFF77E640）说明如下：

**PMUGRF\_SOC\_CON0**

Address: Operational Base + offset (0x00180)  
SoC control register 0

Bit	Attr	Reset Value	Description
-----	------	-------------	-------------

**RK3399 TRM**

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:10	RO	0x0	reserved
9	RW	0x1	pmu1830_vol pmu IO 1.8v/3.0v select. 0: 3.0v ; 1: 1.8v ;
8	RW	0x1	pmu1830_volssel pmu GPIO1 1.8v/3.0v control source select. 0: controlled by IO_GPIOB1 ; 1: controlled by PMUGRF.SOC_CON0.pmu1830_vol
7	RO	0x0	reserved

**GRF\_IO\_VSEL**

Address: Operational Base + offset (0x0e640)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:4	RO	0x0	reserved
3	RW	0x0	gpio1833_gpio4cd_ms
2	RW	0x0	sdmmc_gpio4b_ms
1	RW	0x0	audio_gpio3d4a_ms
0	RW	0x0	bt656_gpio2ab_ms

```
# io -r -4 0xff320180
```

```
ff320180: 00000300
```

```
# io -r -4 0xff77e640
```

```
ff77e640: 00000003
```