

# Rockchip

## IO-Domain 开发指南

发布版本:1.0

日期:2016.06

# 前言

## 概述

## 产品版本

芯片名称	内核版本
RK3399	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

日期	版本	作者	修改说明
2016-06-30	V1.0	WDC	初始版本

# 目录

1	电源域设置.....	1-1
1.1	驱动文件 与 DTS 节点: .....	1-1
1.2	使用 IO-Domain 驱动好处: .....	1-1
2	如何配置电源域 .....	2-1
3	无 PMIC 情况如何处理 .....	3-1
4	总结 .....	4-1

# 1 电源域设置

## 1.1 驱动文件与 DTS 节点：

驱动文件所在位置：

drivers/power/avs/rockchip-io-domain.c

Dts 节点：

```
io-domains {
    compatible = "rockchip,rk3399-io-voltage-domain";
    rockchip,grf = <&grf>;

    bt656-supply = <&vcc1v8_dvp>;
    audio-supply = <&vcca1v8_codec>;
    sdmmc-supply = <&vcc_sd>;
    gpio1830-supply = <&vcc_3v0>;
};

pmu-io-domains {
    compatible = "rockchip,rk3399-pmu-io-voltage-domain";
    rockchip,grf = <&pmugrf>;

    pmu1830-supply = <&vcc_1v8>;
};
```

## 1.2 使用 IO-Domain 驱动好处：

1. 在 IO-Domain 的 DTS 节点统一配置电压域，不需要每个驱动都去配置一次，便于管理；
2. 依照的是 Upstream 的做法，以后如果需要 Upstream 比较方便；
3. IO-Domain 的驱动支持运行过程中动态调整电压域，例如 PMIC 的某个 Regulator 可以 1.8v 和 3.3v 的动态切换，一旦 Regulator 电压发生改变，会通知 IO-Domain 驱动去重新设置电压域。

## 2 如何配置电源域

以下是 RK3399 原理图上的 Power Domain Map 表以及配置示例：

RK3399 Power Domain Map					
Part Port	Domain	Pin name in datasheet	I/O type	Power supply	Power source
Part C	PMUIO1	PMUIO1_gpio0ab	1.8V only	VCC1V8_PMU	RK808 VLDO3
Part E	PMUIO2	PMUIO2_gpio1abcd	1.8V(Default) 3.0V	VCC1V8_PMU	RK808 VLDO3
Part I	APIO1	GMAC_gpio3abc	3.3V only	VCC1V8_IO VCC3V3_IO	RK808 BUCK4 RK808 VSW2
Part L	APIO2	BT656_gpio2ab	1.8V(Default) 3.0V	VCC1V8_DVP	RK808 VLDO1
Part G	APIO3	WIFI/BT_gpio2cd	1.8V only	VCC1V8_WIFI	RK808 BUCK4
Part K	APIO4	GPIOD1833_gpio4cd	1.8V(Default) 3.0V	VCC_1V5 VCC3V0_IO	RK808 VLDO6 RK808 VLDO8
Part J	APIO5	AUDIO_gpio3d_gpio4a	1.8V(Default) 3.0V	VCCA1V8_CODEC	RK808 VLDO7
Part F	SDMMC0	SDMMC_gpio4b	1.8V(Default) 3.0V	VCC_SDIO	RK808 VLDO4

通过 RK3399 SDK 的原理图可以看到 bt656-supply 的电压域连接的是 vcc18\_dvp, vcc\_io 是从 PMIC RK808 的 VLDO1 出来的；

在 DTS 里面可以找到 vcc1v8\_dvp， 将 bt656-supply = <&vcc18\_dvp>。

其他路的配置也类似，需要注意的是如果这里是其他 PMIC，所用的 Regulator 也不一样,具体以实际电路情况为标准。

## 3 无 PMIC 情况如何处理

如果项目硬件上并没有使用 PMIC，可能你找不到相应的 Regulator 来配置，你可以在 DTS 文件里面增加两个固定的 Regulator，vccio\_1v8\_reg 和 vccio\_3v3\_reg，一般 3.3v 和 1.8v 就已经够用了。

例如 3228-SDK 的 IO-Domain 配置：

```
io-domains {
    status = "okay";
    vccio1-supply = <&vccio_3v3_reg>;
    vccio2-supply = <&vccio_1v8_reg>;
    vccio4-supply = <&vccio_3v3_reg>;
};

regulators {
    compatible = "simple-bus";
    #address-cells = <1>;
    #size-cells = <0>;

    vccio_1v8_reg: regulator@0 {
        compatible = "regulator-fixed";
        regulator-name = "vccio_1v8";
        regulator-min-microvolt = <1800000>;
        regulator-max-microvolt = <1800000>;
        regulator-always-on;
    };

    vccio_3v3_reg: regulator@1 {
        compatible = "regulator-fixed";
        regulator-name = "vccio_3v3";
        regulator-min-microvolt = <3300000>;
        regulator-max-microvolt = <3300000>;
        regulator-always-on;
    };
};
```

事实上可以灵活应用，只要可用的 Regulator 的电压与实际电路的电压相符合都可以拿来使用。

# 4 总结

IO-Domain 的配置是非常重要的，每个项目都可能不一样，在写新的 DTS 文件时，一定要根据硬件实际，去做正确的配置，或者可能会出现无法预知的问题。