

IO-Domain 开发指南

前言

一般 IO 电源的电压有 1.8v, 3.3v, 2.5v, 5.0v 等, 有些 IO 同时支持多种电压, io-domain 就是配置 IO 电源域的寄存器, 依据真实的硬件电压范围来配置对应的电压寄存器, 否则无法正常工作; 下面有罗列出哪些 RK 芯片都需要配置 io-domain。

产品版本

芯片名称	内核版本
RK3188	4.4
RK3288	4.4
RK3036	4.4
RK312x	4.4
RK322x	4.4
RK3368	3.10
RK3368	4.4
RK3366	4.4
RK3399	4.4
RV1108	3.10
RV1108	4.4
RK3228H	3.10
RK3328	4.4
RK3326/PX30	4.4
RK3308	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

目录

IO-Domain 开发指南

1. 驱动文件与 DTS 节点
 - 1.1 驱动文件
 - 1.2 DTS 节点
2. TRM 中的描述
3. 驱动软件流程
 - 3.1 初始化配置
 - 3.2 动态配置
4. 如何配置 io-domain
 - 4.1 通过 rockchip-io-domain.txt 文档寻找名称
 - 4.2 通过硬件原理图寻找 io-domain 配置的真实电压
 - 4.3 通过 DTS 配置
5. 通过硬件 Pin 脚控制的电源域一般不做配置
6. DTS 中无定义 Regulator 情况处理
7. 常见问题
 - 7.1 如何确定某个 Pin 脚所在的电源域寄存器是否配置正确
 - 7.2 io-domain 的寄存器不正确

1. 驱动文件与 DTS 节点

1.1 驱动文件

驱动文件所在位置: `drivers/power/avs/rockchip-io-domain.c`

1.2 DTS 节点

- 内核 3.10 版本的 DTS 节点合并:

```
1  io-domains {
2      compatible = "rockchip,rk3368-io-voltage-domain";
3      rockchip,grf = <&grf>;
4      rockchip,pmugrf = <&pmugrf>;
5
6      /*GRF_IO_VSEL*/
7      dvp-supply = <&ldo7_reg>;      /* DVPIO_VDD */
8      wifi-supply = <&ldo7_reg>;      /* APIO2_VDD */
9      audio-supply = <&dcadc2_reg>;  /* APIO3_VDD */
10     sdcard-supply = <&ldo1_reg>;    /* SDMMC0_VDD */
11     gpio30-supply = <&dcadc2_reg>;  /* APIO1_VDD */
12     gpio1830-supply = <&dcadc2_reg>; /* ADIO4_VDD */
13
14     /*PMU_GRF_IO_VSEL*/
15     pmu-supply = <&ldo5_reg>;        /* PMUIO_VDD */
16     vop-supply = <&ldo5_reg>;        /* LCDC_VDD */
17 };
```

- 内核 4.4 版本的 DTS 节点 GRF 和 PMUGRF 分开:

```
1  &io_domains {
2      status = "okay";
3      dvp-supply = <&vcc_18>;
4      audio-supply = <&vcc_io>;
5      gpio30-supply = <&vcc_io>;
6      gpio1830-supply = <&vcc_io>;
7      sdcard-supply = <&vccio_sd>;
8      wifi-supply = <&vccio_wl>;
9  };
10
11  &pmu_io_domains {
12      status = "okay";
13
14      pmu-supply = <&vcc_io>;
15      vop-supply = <&vcc_io>;
16  };
```

2. TRM 中的描述

很多工程师反映在 TRM 中找不到 io-domain 相关的寄存器，可以通过 TRM 来搜索需要配置的 io-domain 寄存器描述，在 GRF/PMUGRF 章节搜索 'vsel'，'VSEL' 或者 'volsel' 索引，PMUGRF 中的 io-domain 是用来控制 PMU IO。

支持配置的两种电压 1.8v / 3.3v:

- 寄存器配置成 1，一般对应的电压范围是 1.62v ~ 1.98v，typical 电压 1.8v;
- 寄存器配置成 0，一般对应的电压范围是 3.00v ~ 3.60v，typical 电压 3.3v。

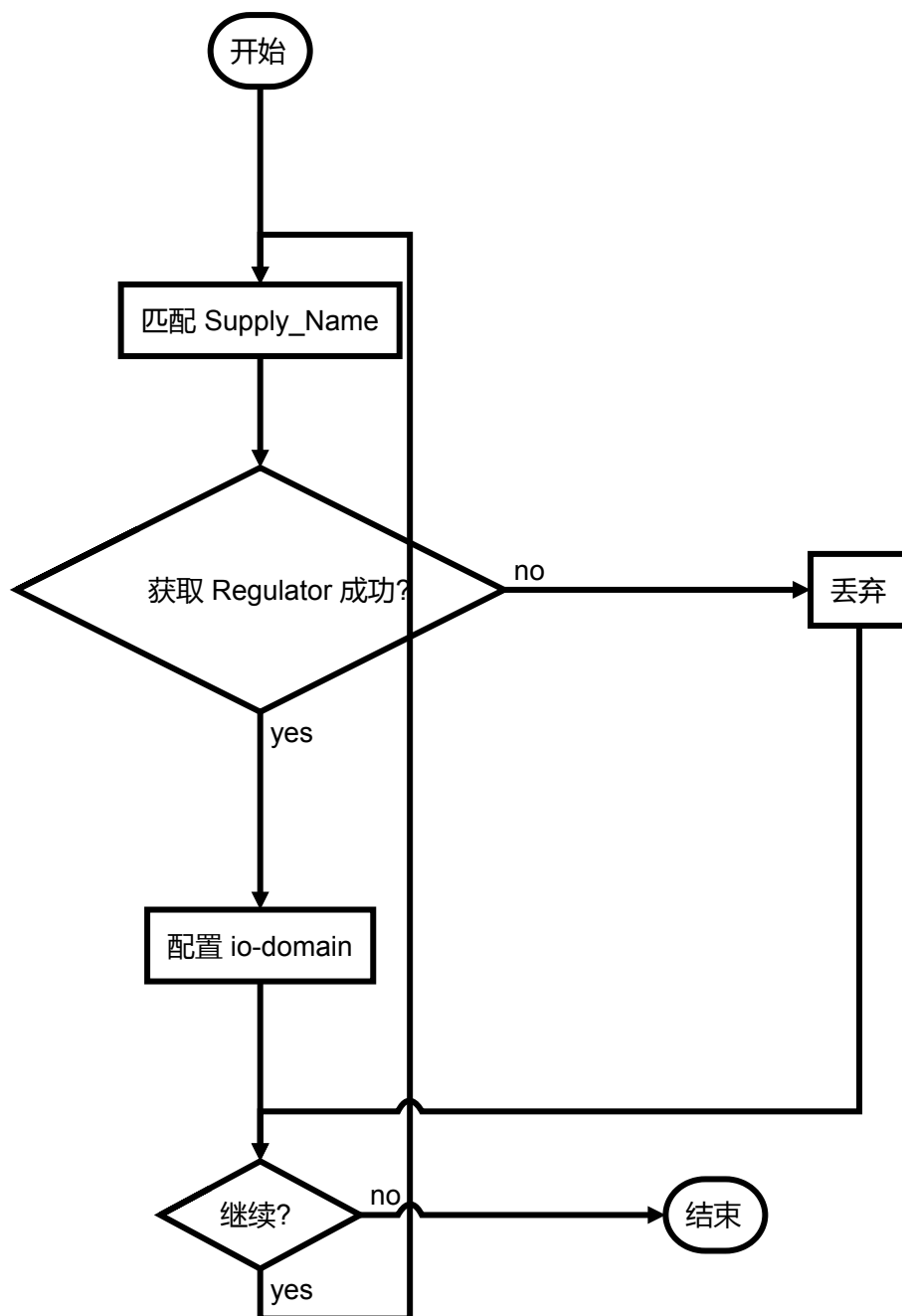
具体电压范围要以实际芯片的 Datasheet 为准。

3. 驱动软件流程

下面是 rockchip-io-domain.c 驱动的软件流程图，主要分为两个方面：

3.1 初始化配置

在驱动的 probe 函数中的 supply name，获取 dts 中对应 supply name 定义的 regulator，再根据 regulator 的电压配置 io-domain 寄存器，如果是 1.8v 那一档，该 bit 配置为 1；如果是 3.3v 那一档，该 bit 配置为 0。



3.2 动态配置

在初始化的过程中，会绑定 regulator，通过注册 notify 的方式，一旦这个 regulator 的电压发生变化，就会通知 io-domain 驱动更新成对应的寄存器，做到动态更新寄存器的效果。

4. 如何配置 io-domain

不是每个 IO 电源域都需要配置，有些 IO 的电源域是固定的，不需要配置。下面 3 个步骤描述如何通过软件配置 io-domain:

4.1 通过 rockchip-io-domain.txt 文档寻找名称

需要在软件上通过 dts 配置的 IO 电源域在 Linux Kernel 的目录下的文件都有描述：Documentation/devicetree/bindings/power/rockchip-io-domain.txt；由于 TRM 文档和硬件原理图上对同一个 io-domain 名称描述可能有差异，在 rockchip-io-domain.txt 文档上统一描述了 TRM 与 硬件原理图上 io-domain 名称的对应关系。

例如 RK3399 Soc，通过查看 rockchip-io-domain.txt 文档，我们知道了 RK3399 的电源域需要配置包含 bt565，audio，sdmmc，gpio1830，以及 PMUGRF 下面的 pmu1830 这几个 supply，后面的 The supply connected to “***_VDD” 表示在硬件原理图上对应的名称。

Possible supplies for rk3399:

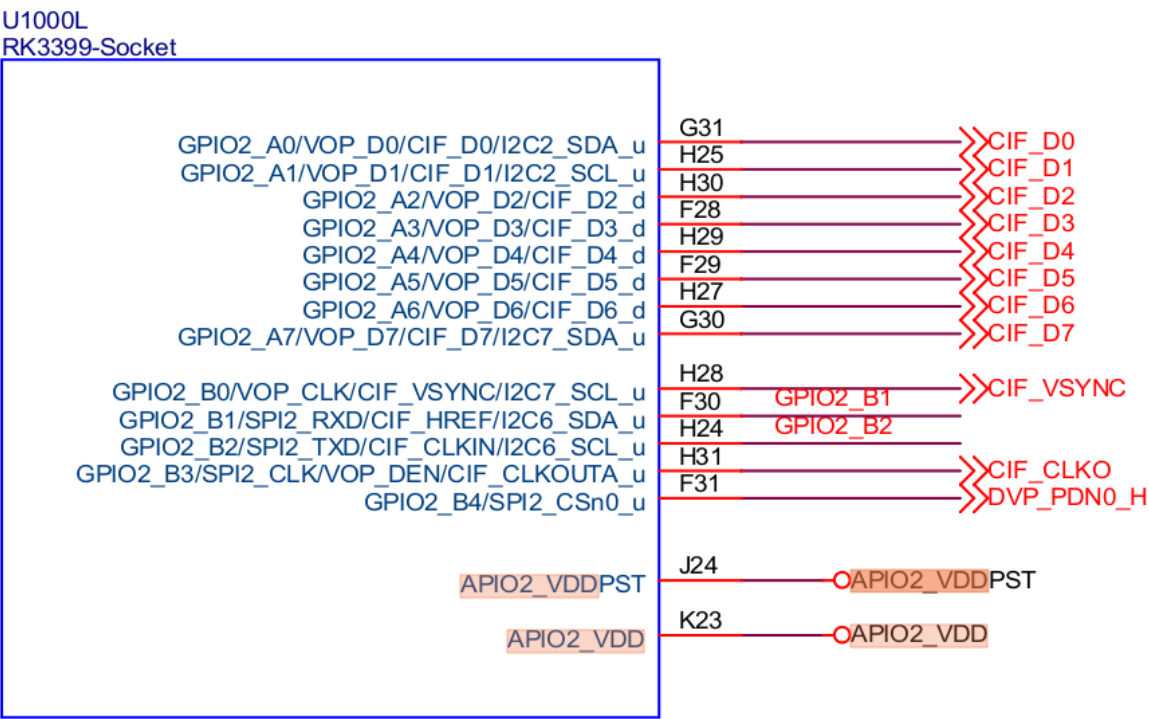
- bt656-supply: The supply connected to APIO2_VDD.
- audio-supply: The supply connected to APIO5_VDD.
- sdmmc-supply: The supply connected to SDMMC0_VDD.
- gpio1830-supply: The supply connected to APIO4_VDD.

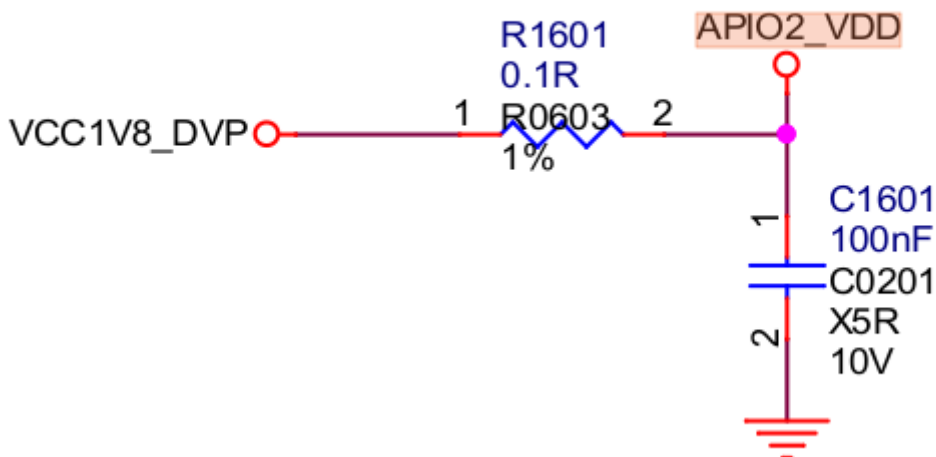
Possible supplies for rk3399 pmu-domains:

- pmu1830-supply:The supply connected to PMUIO2_VDD.

4.2 通过硬件原理图寻找 io-domain 配置的真实电压

仍以 RK3399-EVB 原理图 和 bt656 IO 电源域为例，我们在 rockchip-io-domain.txt 中找到了 bt656 对应的硬件原理图上表示为 APIO2_VDD。所以通过逆向搜索 ‘APIO2_VDD’ 得到 RK3399-EVB 硬件原理图上的 APIO2_VDD 电源是由 RK808 下的 VCC1V8_DVP 供给。





4.3 通过 DTS 配置

以上两步做完后，得到了配置的名称和供电源头，在 DTS 里面找到对应的 regulator: vcc1v8_dvp，就可以在 rk3399-evb.dtsi 配置上 “bt656-supply = <&vcc1v8_dvp>;”，其他的电源域配置类似。

5. 通过硬件 Pin 脚控制的电源域一般不做配置

在 RK Soc 中的一些 IO 电源域在硬件上已经通过某个 Pin 脚来控制的，这种情况下我们 kernel 的 DTS 一般不去配置，不破坏当前的硬件状态，像 flash 和 emmc 这些模块的 IO 电源域一般都是 Pin 脚来控制的。

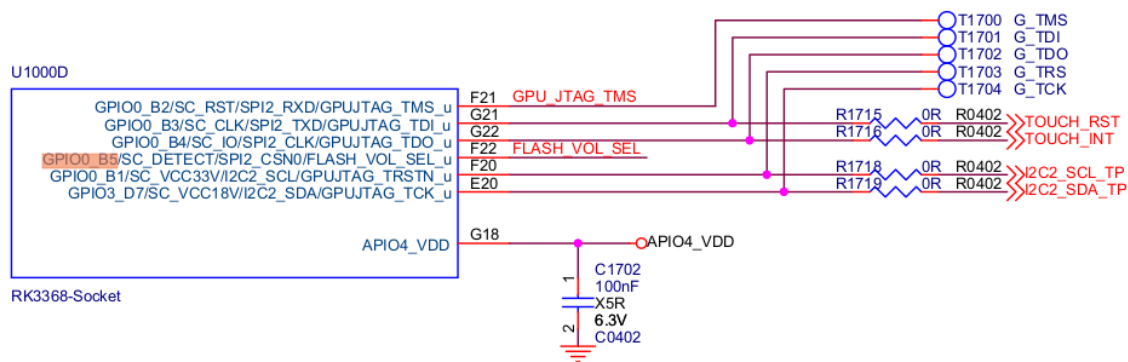
在 TRM 的 io-domain 寄存器描述中，我们可以看到哪些电源域是可以通过 Pin 脚来控制的，以及通过硬件上这个 Pin 脚的输入电压状态来确认当前这个电压域的配置；也可以通过 GRF 寄存器来配置，两种选择。

例如，RK3368 Soc 的 TRM 和 RK3368-evb 的硬件原理图上有下面寄存器的描述和硬件上 Pin 脚的配置。

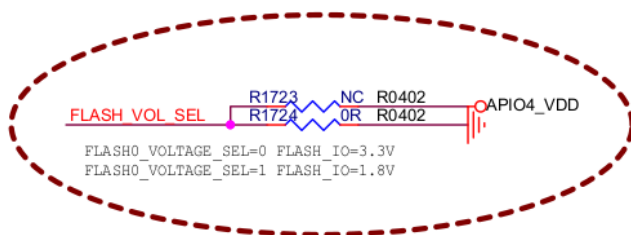
- TRM 寄存器描述：

14	RW	0x0	flash_poc_ctrol flash IO domain poc control selection 0: controled by gpio_0b5 pad 1: controled by bit 2 of IO_VSEL
2	RW	0x1	flash0_v18sel FLASH0 IO domain 1.8V voltage selection 1'b0: 3.3V/2.5V 1'b1: 1.8V

- 硬件原理图：



FLASH Driver IO SEL



FLASH_VOL_SEL	FLASH_IO Voltage
1.8V or 2.5V or 3.3V	1.8V Driver IO
0V	3.3V Driver IO

6. DTS 中无定义 Regulator 情况处理

在使用的过程中可能会遇到，你找不到相应的 regulator 来配置，可能项目上面未使用 pmic 等电源，只是简单的拉了一个电源过来，dts 上找不到 regulator 的定义，那么你需要在 dts 文件里面增加 fixed regulator 的定义，一般 3.3v 和 1.8v 两个 regulator 就够用了。

下面是 rk3229-evb.dts 的配置例子，确定硬件上的电压是用 1.8v 还是 3.3v，配置成相应的 regulator:

```

1      regulators {
2          compatible = "simple-bus";
3          #address-cells = <1>;
4          #size-cells = <0>;
5
6          vccio_1v8_reg: regulator@0 {
7              compatible = "regulator-fixed";
8              regulator-name = "vccio_1v8";
9              regulator-min-microvolt = <1800000>;
10             regulator-max-microvolt = <1800000>;
11             regulator-always-on;
12         };
13
14         vccio_3v3_reg: regulator@1 {
15             compatible = "regulator-fixed";
16             regulator-name = "vccio_3v3";
17             regulator-min-microvolt = <3300000>;
18             regulator-max-microvolt = <3300000>;
19             regulator-always-on;
20         };
21     };
22
23     &io_domains {
24         status = "okay";

```

```

25
26     vccio1-supply = <&vccio_3v3_reg>;
27     vccio2-supply = <&vccio_1v8_reg>;
28     vccio4-supply = <&vccio_3v3_reg>;
29 };
30

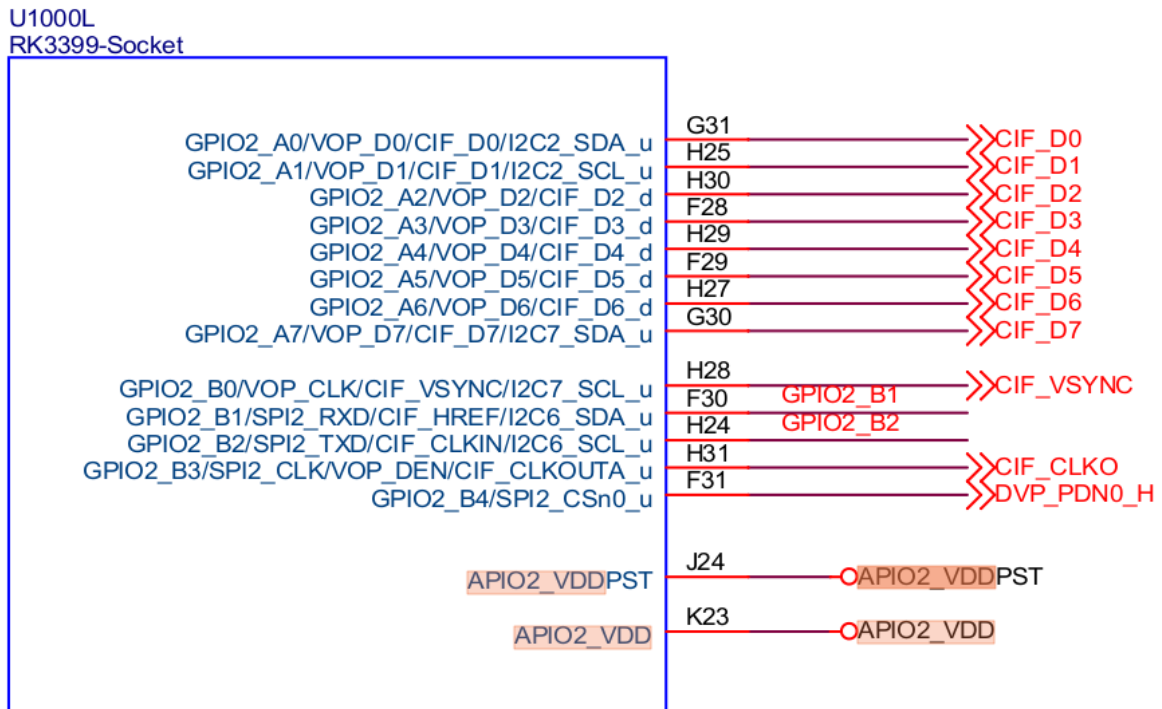
```

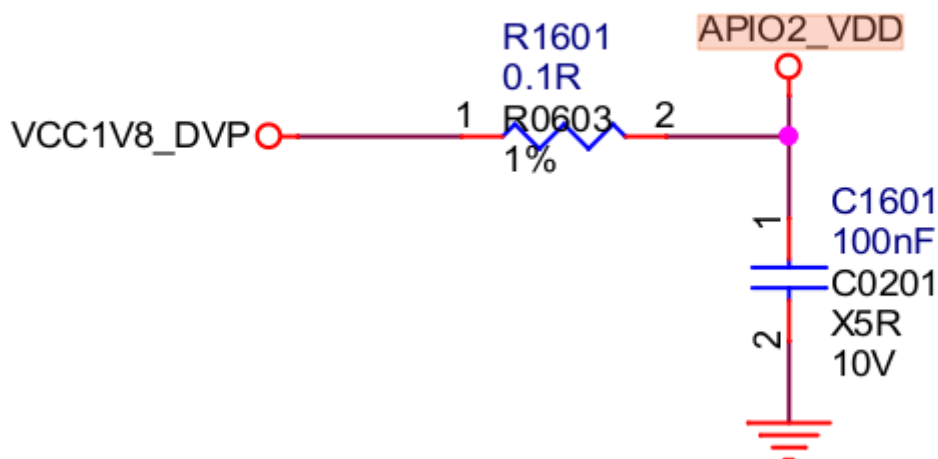
7. 常见问题

7.1 如何确定某个 Pin 脚所在的电源域寄存器是否配置正确

经常遇到客户报的问题是某 pin 脚的电压与所期望的不符，很有可能就是电源域配置问题。例如，在 RK3399 上，软件上代码已经让 GPIO2_B1 输出高，但是实际通过量测发现电压不对；通过读取寄存器已经确认该 pin 脚已经将 iomux 配置成 gpio，并且也设置成输出高，这就很有可能是 io-domain 没有配置正确。那么这时候就要确认电源域寄存器是否配置正确，方法就是上面介绍的如何配置电源域的反步骤。

- 先确定这个 io 所在的电源域，一般是看硬件原理图或者 Datasheet 来确定。例如，RK3399 下面通过硬件原理如图发现 GPIO2_B1 所在的电源域硬件上表示为 APIO2_VDD，并且 APIO2_VDD 是接的电压是 VCC1V8_DVP。





- 通过 rockchip-io-domain.txt 文档找到对应的名称。例如，在 rockchip-io-domain.txt 文档上找到的电源域对应的名称是“bt656”。

```
Possible supplies for rk3368 pmu-domains:
- pmu-supply: The supply connected to PMUIO_VDD.
- vop-supply: The supply connected to LCDC_VDD.

Possible supplies for rk3399:
- bt656-supply: The supply connected to APIO2_VDD.
- audio-supply: The supply connected to APIO5_VDD.
- sdmmc-supply: The supply connected to SDMMC0_VDD.
- gpio1830 The supply connected to APIO4_VDD.

Possible supplies for rk3399 pmu-domains:
- pmu1830-supply: The supply connected to PMUIO2_VDD.
```

- 在 TRM 上找到这个寄存器，通过 io 命令或者其他方式读取这个寄存器的值，一般基地址是 GRF 或者 PMUGRF。例如，在 TRM 文档上搜索到“bt656”寄存器描述，为 bit0，查看寄存器偏移为 0xe640，GRF 基地址为 0xff770000。在串口终端输入“io -4 0xff77e640”，得到 io-domain 寄存器值，如果该寄存器值 bit0 为 1，表示 1.8v，与硬件实际电压 VCC1V8_DVP，dts 中该项配置正确；如果 bit0 为 0，则表示 3.3v，与硬件实际电压 VCC1V8_DVP 不符，dts 中该项配置不正确。

3	RW	0x0	gpio1833_gpio4cd_ms
2	RW	0x0	sdmmc_gpio4b_ms
1	RW	0x0	audio_gpio3d4a_ms
0	RW	0x0	bt656_gpio2ab_ms

7.2 io-domain 的寄存器不正确

常见的寄存器错误，可能是以下几个问题

- 所配置的 regulator 电压不对；
- 未配置 Regulator 或 Regulator 未使能；
- Regulator 比 io-domain 驱动加载更慢，获取 regulator 失败。