

# Rockchip

# Thermal 开发指南

发布版本:1.0

日期:2017.02

# 前言

## 概述

本文档主要介绍 RK 平台 Thermal 配置的调试方法。

## 产品版本

产品名称	内核版本
RK3228H	Linux3.10
RK3328	Linux3.10

## 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

日期	版本	作者	修改说明
2017-02-16	V1.0	HYZ	初始发布

# 目录

前言 .....	I
目录 .....	II
1 概述 .....	1-1
2 重要概念 .....	2-1
3 配置方法 .....	3-1
3.1 Tsadc 配置 .....	3-1
3.1.1 Menuconfig 配置 .....	3-1
3.1.2 DTS 配置 .....	3-1
3.2 Normal 策略配置 .....	3-3
3.2.1 默认使用 Normal 策略，以 CPU 为例说明 .....	3-3
3.2.2 使用 Performance 策略，以 CPU 为例说明 .....	3-4
3.2.3 Thermal_zone 配置 .....	3-4
4 调试接口 .....	4-1
4.1 关温控 .....	4-1
4.2 获取当前温度 .....	4-1

# 1 概述

本文档主要描述 Thermal 的相关的重要概念、配置方法和调试接口。

# 2 重要概念

在 Linux 内核中，定义一套温控（thermal）框架，在 3.10 内核 arm64 版本，我们使用 thermal 框架的 sysfs 接口读取当前的温度。温控策略是自定义的方式：

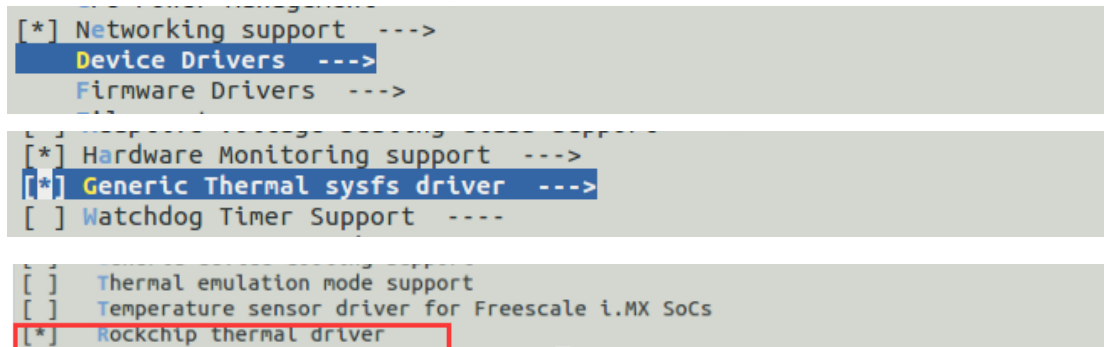
- **performance** 策略：当前温度超过了目标温度，CPU 会设定在固定的频率，具体的数值配置在芯片级 dtsi 文件。
- **normal** 策略：当前温度超过目标温度不同的温度值时，CPU 会降低相应的频率，具体的数值配置在芯片级 dtsi 文件。

# 3 配置方法

## 3.1 Tsadc 配置

### 3.1.1 Menuconfig 配置

make ARCH=arm64 menuconfig



### 3.1.2 DTS 配置

如下是芯片级的 DTSI 的配置:

```
tsadc: tsadc@ff250000 {
    compatible = "rockchip,rk322xh-tsadc";
    reg = <0x0 0xff250000 0x0 0x100>;
    interrupts = <GIC_SPI 58 IRQ_TYPE_LEVEL_HIGH>;
    clock-frequency = <50000>;
    clocks = <&clk_tsadc>, <&clk_gates16 14>;
    clock-names = "tsadc", "apb_pclk";
    pinctrl-names = "default", "tsadc_int";
    pinctrl-0 = <&tsadc_gpio>;
    pinctrl-1 = <&tsadc_int>;
    resets = <&reset RK322XH_SRST_TSADC_P>;
    reset-names = "tsadc-apb";
    hw-shut-temp = <120000>;
    tsadc-tshut-mode = <0>; /* tshut mode 0:CRU 1:GPIO */
    tsadc-tshut-polarity = <1>; /* tshut polarity 0:LOW 1:HIG */
    #thermal-sensor-cells = <1>;
    status = "disabled";
};
```

```
tsadc: tsadc@ff250000 {
    compatible = "rockchip,rk322xh-tsadc";
```

Tsadc 驱动加载的标识字符串

```
reg = <0x0 0xff250000 0x0 0x100>;
```

Tsadc 操作的寄存器基地址和寄存器地址总长度

```
interrupts = <GIC_SPI 58 IRQ_TYPE_LEVEL_HIGH>;
```

Tsadc 中断号

```
clocks = <&clk_tsadc>, <&clk_gates16 14>;
clock-names = "tsadc", "apb_pclk";
```

Tsadc 模块的两个 clock，其中"tsadc"是 Tsadc 的工作时钟，"apb\_pclk"是 Tsadc 的配置时钟

```
clock-frequency = <50000>;
```

配置 Tsadc 的工作时钟是 50000，Tsadc 的配置时间周期，是以这个时钟为基准的。

```
resets = <&reset RK322XH_SRST_TSADC_P >;
reset-names = "tsadc-apb";
```

Tsadc 的 reset 控制，用于 resetTsadc 模块

```
pinctrl-names = "default", "tsadc_int";
pinctrl-0 = <&tsadc_gpio>;
pinctrl-1 = <&tsadc_int>;
```

Tsadc 工作的 GPIO 口配置，这个配置与下面的配置对应。默认 Default 的时候，是 GPIO 口功能。tsadc\_int 配置请参考下面 rockchip,hw-tshut-mode 配置。

```
tsadc_pin {
    tsadc_int: tsadc-int {
        rockchip,pins =
        <2 GPIO_B5 RK_FUNC_2 &pcfg_pull_none>;
    };
    tsadc_gpio: tsadc-gpio {
        rockchip,pins =
        <2 GPIO_B5 RK_FUNC_GPIO &pcfg_pull_none>;
    };
};
```

```
#thermal-sensor-cells = <1>;
```

表示 Tsadc 可以向 ThermalZone 注册，而且对应的 ThermalZone 引用 Tsadc 的时候，带有一个参数。

```
如: thermal-sensors = <&tsadc 0>;
    hw-shut-temp = <120000>;
```

Tsadc 设定关机温度是 120 度，0 表示 tsadc 的通道

```
status = "disabled";
};
```

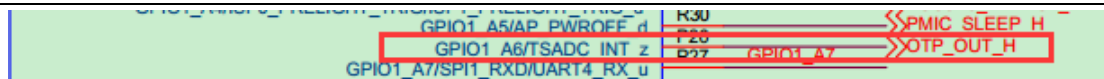
注：

1. Tsadc 模块，默认在 DTSI 中是 Disabled 状态，要启用的时候，需要在板级的 DTS 中配置，如：

```
&tsadc {
    rockchip,hw-tshut-mode = <1>; /* tshut mode 0:CRU 1:GPIO */
    rockchip,hw-tshut-polarity = <1>; /* tshut polarity 0:LOW 1:HIGHIGH */
    status = "okay";
};
```

2. rockchip,hw-tshut-mode = <1>

配置温度超过关机温度的复位方式，配置 0 是通过复位 SoC 的 CRU 模块，配置 1 是通过配置上文提到的 pinctrl-1 = <&tsadc\_int>;来实现，这个引脚功能 tsadc\_int 一般会接入 pmic 的 reset 引脚(如下图)，至于这个引脚是高有效还是低有效，需要配置 hw-tshut-polarity 来实现。



注：有的芯片，这个引脚没有引出来，具体请参考 TRM 手册

## 3.2 策略配置

### 3.2.1 默认使用 Normal 策略，以 CPU 为例说明

```
echo 1 > /sys/module/rockchip_pm/parameters/policy
```

在 init.rc 脚本里面配置 policy 参数为 1

温控参数配置在 dvfs 里面，具体配置参数如下：

```
temp-limit-enable = <1>;
tsadc-ch = <0>;
target-temp = <95>;
min_temp_limit = <600000>;
max_temp_limit = <1200000>;
normal-temp-limit = <
/*delta-temp    delta-freq*/
    3        96000
    6        144000
    9        192000
    15       384000
>;
```

```
temp-limit-enable = <1>;
```

使能温控

```
tsadc-ch = <0>;
```

温度采集的 tsadc 通道

```
target-temp = <95>;
```

设定温控目标温度是 95 度

```
min_temp_limit = <600000>;
```

温度超过目标温度后，CPU 的最小工作频率是 600M。比如，按照该温控策略，计算得知需要降频到 400M，但 CPU 还是会设定工作频率 600M。

```
max_temp_limit = <1200000>;
```

温度超过目标温度后，CPU 的最大工作频率是 1200M。例如，当前 CPU 的工作频率是 1400M，当温度超过目标温度 95 度，CPU 会急剧降到 1200M 以内。

```
normal-temp-limit = <
/*delta-temp    delta-freq*/
    3        96000
    6        144000
    9        192000
    15       384000
>;
```

温度超过设定的温控目标温度 3 度，就以 96M 的步进来降低 CPU 频率，其他值类似。具体请参考 static void dvfs\_temp\_limit\_normal(struct dvfs\_node \*dvfs\_node, int temp)。



### 3.2.2 使用 Performance 策略，以 CPU 为例说明

```
performance-temp-limit = <
    /*temp    freq*/
    110      816000
>;
echo 0 > /sys/module/rockchip_pm/parameters/policy
```

配置 policy 为 0，或者不用配置这个参数，policy 默认是 0

温度超过 110 度，频率会限制在 816M 以下。具体请参考 static void

dvfs\_temp\_limit\_performance(struct dvfs\_node \*dvfs\_node, int temp)。

### 3.2.3 Thermal\_zone 配置

以 RK3328 的配置为例

```
thermal-zones {
    cpu_thermal: cpu-thermal {
        polling-delay-passive = <1000>; /* milliseconds */
        polling-delay = <5000>; /* milliseconds */
        /* sensor      ID */
        thermal-sensors = <&tsadc      0>;
    };
};
```

Thermal-Zones 下的第一级子节点，对应的是不同的 Tsadc，RK3328 上有 1 个 Tsadc，在 CPU 旁边，我们可以看到 1 个子节点。

```
polling-delay-passive = <1000>;
```

温度超过阈值时，每隔 1000ms 查询温度，并限制频率。

```
polling-delay = <5000>;
```

温度未超过阈值时，每隔 5000ms 查询温度。

```
thermal-sensors = <&tsadc 0>;
```

当前 thermal\_zone 的温度是通过 tsadc0 获取的。我们如上简单配置，主要是复用 thermal 框架的 sysfs 接口，方便用户层读取当前的温度。

# 4 调试接口

## 4.1 关温控

主控默认开启温控，即 dvfs 的 dts 里面配置 temp-limit-enable = <1>。如果要关闭温控，dvfs 的 dts 里面配置 temp-limit-enable = <0>;或者不配置 temp-limit-enable 这个参数。

## 4.2 获取当前温度

以 RK3328 为例，获取 CPU 温度，在串口中输入如下命令：

```
cat /sys/class/thermal/thermal_zone0/temp
```