D1 Tina Linux 温度控制使用指南

版本号: 1.0
发布日期: 2021.04.07

版本历史

版本号	日期	制/修订人	内容描述
1.0	2021.04.07	AWA1610	1. 初始版本

目　录

1 概述

1.1 编写目的

介绍 Tina 温度控制机制实现及相关的模块的实现与配置。

1.2 适用范围

适用于 D1 平台。

1.3 相关人员

适用 Tina 平台的广大客户和对 Linux thermal 框架感兴趣的同事。

2 Linux 温控框架简介

2.1 基础框架

温控（thermal）系统的核心功能就是将目标温度控制在一个合理的范围。温度过高，则会快速消耗寿命，体验不佳，严重时还会带来安全隐患。

而对于嵌入式系统来说，降温的一般方法是降频，温度降低越厉害，系统频率越低，对应的性能也越差。而如果不降温，则系统可能温度会越来越高，导致系统寿命降低，体验不好。因此选择合适的降温状态是非常重要的一点。

要达到此目的，首先需要思考两个关键点：

（1）如何降低温度

（2）何时降低温度

为了实现上述功能需求，thermal framework 抽象出 Thermal Zone Device、Thermal Cooling Device、Thermal Governor、Thermal Core 四个部分。

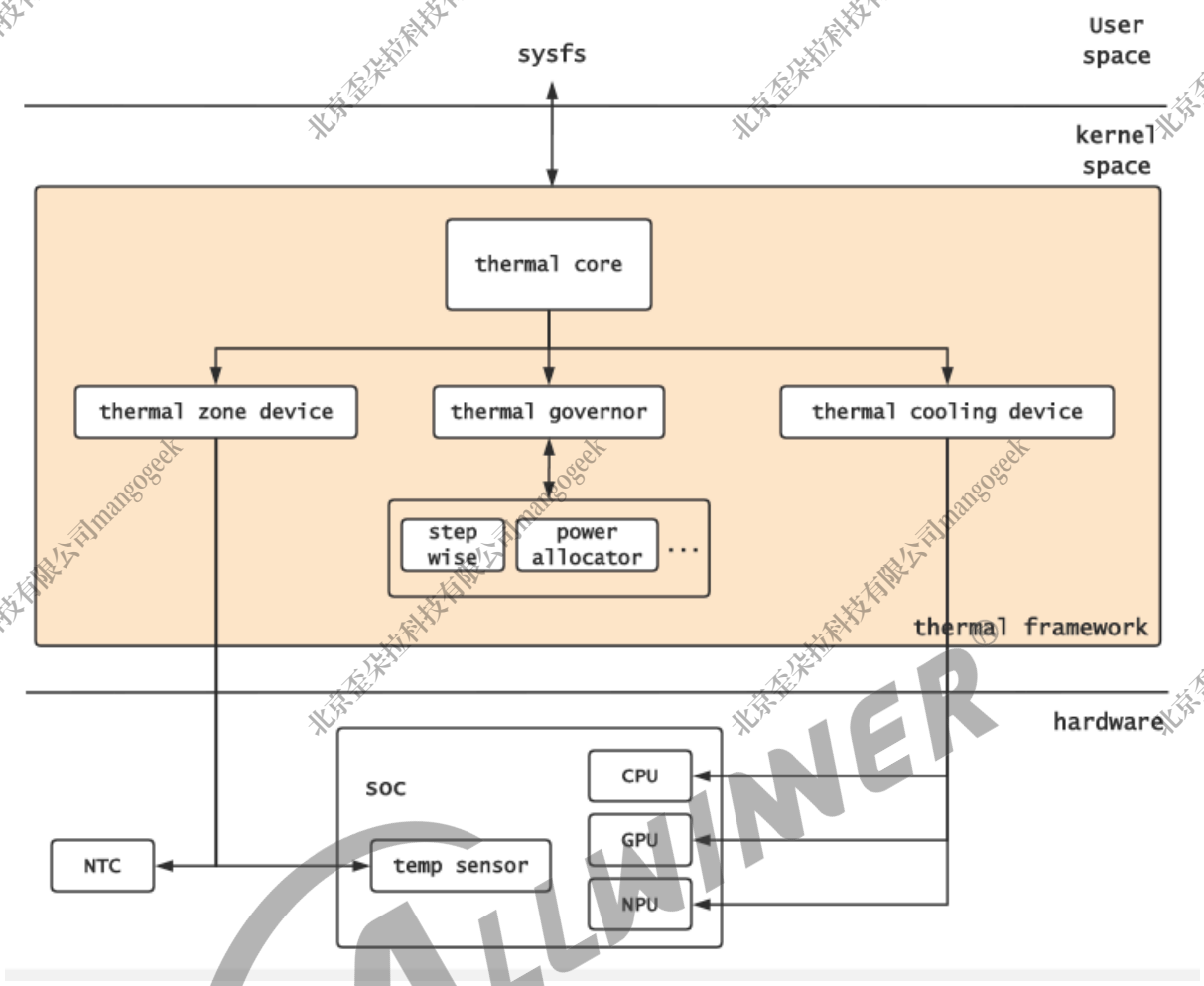

图 2-1: thermal framework

如何降低温度是由 Thermal Cooling Device 处理，它负责向 thermal 框架提供不同级别的降温状态，级别越大降温越深，一般对系统性能和功耗的影响就会越大。
何时降低温度是由 Thermal Zone Device 和 Thermal Governor 模块负责，前者负责对系统温度采样，为 thermal 框架提供实时数据反馈，而后者根据这些数据计算并选择一个合适的降温状态。

2.1.1 Thermal Zone Device——获取温度的设备

Thermal Zone Device 是在 thermal framework 中充当测温设备。由于获取温度的设备有很多，例如 cpu, gpu, ddr 内部集成的 sensor，还有电池温敏电阻，外置温度传感器等等。这些获取温度的设备形态，功能，原理，分布的位置等各有差异，有的内嵌在 IC 中，有的独立于 IC 之外，有的是数字的，有些又是模拟的，甚至有的只是一个虚拟出来的设备。为了方便管理他们，所以内核将获取温度的特性抽象出来统一管理，至于对温度获取的具体实现由对应的设备驱动完成。只有当驱动将设备注册到 thermal framework 中成为 Thermal Zone Device 后，该设备才能参与温控过程。

2.1.2 Thermal Cooling Device——控制温度的设备

与测温设备相同，控制温度的设备也各式各样，常见的有散热片，风冷，水冷等等，他们都是通过增加散热效率降温的。而在嵌入式系统中，一般采用降低产热量的方式，例如降低设备运行频率等等，这种方式通过对某种设备属性的操作，达到温度控制的目的，因此内核也需要将其特性抽象出来，这便是 Thermal Cooling Device。同样，各厂商需要实现自己的降温设备驱动，然后将降温功能注册成 Thermal Cooling Device。

值得注意的是由于在电子系统中，发热是固有的属性，设备常常会过热，而且高温比低温对系统的危害要高得多，因此大多数产品对温度的控制都是专注于如何降温。同样地，在本文中，对温度的控制主要是指降低温度。

在 D1 中，降温的主要方法是限制 CPU 的最高运行频率。

2.1.3 Thermal Governor——温控系统的调度

有了测温和控温的设备，那么何时控温，怎样控温，采取何种策略控温就成了 Thermal 框架最重要的问题了。于是内核提供了 Thermal Governor 模块来抽象这些工作。

1，Thermal Governor 负责对温度系统的调用，如何时采样，采样后根据什么判断是否更新降温状态，如果更新降温状态，采用什么样的策略来计算最终的降温状态等等，这些均由 Thermal Governor 负责管理，但 Governor 只能提供框架，并不实现具体的计算操作。

2，Thermal policy 温控策略主要是根据当前温度来选择合适的降温状态的计算方法。举个简单的例子，当前的温度升高速很快，选择风扇 3 档风，温度升高不快，选择 1 档风，这就是一个温控策略。其中，policy 负责为 Governor 提供具体的的计算工作，Governor 可通过切换不同的 policy 达到不同的计算结果。

2.1.3.1 thermal governor 相关的基础知识

- 常用`policy`说明

用户态可以通过`cat /sys/devices/virtual/thermal/thermal_zone0/available_policies`查看支持的温控策略

```
user_space：用户模式，将温控区间的控制权移交给用户空间，用户可以客制化相关的热功耗参数。（在kernel
    menuconfig选中配置即可）

step_wise：逐步调整模式，该策略属于开环控制，基于温度阈值和趋势，逐步浏览每个cooling设备的cooling等
    级。（下文主要讲述此 policy）

power_allocator：功率分配模式，该策略属于闭环控制，基于每个相关设备的功率，温度和当前功耗通过PID算法进行闭
    环控制。（D1 暂不支持）
```

- trips 和 binds 说明

什么是 trips? 和 binds? 在 thermal policy 的实现中，它需要了解哪一个测温设备，对应哪一个降温状态，以及温度和降温状态间的转换关系才能正确的计算。为了简化这个关系的描述，Linux thermal 提供了 trips 和 cooling-maps 的概念。policy 的实现需要依赖于 trips 和 binds 配置。
trips 一般理解为触发点，当测温设备达到一定的温度时，就需要更新降温状态，而此时对应的温度值就是一个 trip 。一个测温设备可以有多个触发点，这个 trip 集合就称为 trips。
cooling-maps 有时也称做 binds，一般理解为绑定。他的含义是为一个 trip 绑定一个或多个降温状态的集合（这个集合下文称为 states），当测温设备温度达到 trip 时，会触发 thermal governor 调度一次 thermal policy，而 thermal policy 将会从这个降温状态的集合（states）中，计算出一个合适的降温状态（state）返回给 thermal governor 去设置，这样就完成一次温控动作。

另外指出的是，并不是每一个 trip 都会触发 thermal governor 调度，因为 Linux 定义了四种 trip 类型，如下：

```
* active: 激活主动冷却
* passive: 启动被动冷却
* hot: 系统过热
* critical: 系统不可靠
```

只有 active，passive 类型的 trip 会触发 thermal governor 调度一次 thermal policy，完成降温状态的更新。而 hot，critical 类型的 trip 属于过热保护的系统警告，触发他们后，不会执行更新降温状态的操作，而是发送 notify 到 thermal_zone_device 中，由 thermal_zone_device 驱动执行过温保护的操作来保护系统。而且，critical 类型的 trip 会在 notify 发送完成后，直接执行关机操作。因此可以通过配置 trip 的类型为 critical，来设置过温关机保护。

2.1.4 Thermal Core——内核及用户空间接口

有了测温控温设备，有了控制策略，剩下的问题就是如何将其联系在一起了。内核抽象了 Thermal core，这是 Thermal 的核心。主要功能是关联测温控温设备和控制策略，并请求内核调度执行；向其他模块提供统一的注册，使用接口，并向用户空间开放 sysfs 调试接口。

2.2 温控策略

2.2.1 step-wise policy

在 D1 中，一般我们采用的温控策略是 step-wise, 这是一种最基础，也是最简单有效的温控策略。它的核心思想是根据当前温度的趋势（上升、下降、平稳），以及当前温度是否高于 trip(触

发点)，来决定 cpu 下一次的 Cooling Device 状态。

Linux thermal 定义的温度趋势有以下五种，都是根据上一次与这一次温度的变化关系而变化的。

```
enum thermal_trend {
    THERMAL_TREND_STABLE, /* temperature is stable */
    THERMAL_TREND_RAISING, /* temperature is raising */
    THERMAL_TREND_DROPPING, /* temperature is dropping */
    THERMAL_TREND_RAISE_FULL, /* apply highest cooling action */
    THERMAL_TREND_DROP_FULL, /* apply lowest cooling action */
};
```

step-wise 主要逻辑：

如果当前温度比一个 trip 温度高，且
a. 如果趋势是 THERMAL_TREND_RAISING，则使用这个 trip 对应的更高的降温状态；
b. 如果趋势是 THERMAL_TREND_DROPPING，则不做任何事情；
c. 如果趋势是 THERMAL_TREND_RAISE_FULL，则使用这个 trip 对应的最高的降温状态；
d. 如果趋势是 THERMAL_TREND_DROP_FULL，则使用这个 trip 对应的最低的降温状态；
e. 如果趋势是 THERMAL_TREND_STABLE，则不做任何事情；

如果当前温度比一个 trip 温度低，且
a. 如果趋势是 THERMAL_TREND_RAISING，不做任何事；
b. 如果趋势是 THERMAL_TREND_DROPPING，则使用此触发点对应的较低的降温状态，如果冷却状态已经等于下限，则停用任何降温措施；
c. 如果趋势是 THERMAL_TREND_RAISE_FULL，则不采取任何措施；
d. 如果趋势是 THERMAL_TREND_DROP_FULL，则使用下限，如果冷却状态已经等于下限，则停用任何降温措施；

温度变化趋势是由这次测量值与上次测量值间的比较确定，若本次测量值高于上次，则温度为 THERMAL_TREND_RAISING 趋势；若本次测量值低于上次，则温度为 THERMAL_TREND_DROPPING 趋势；若两次相同，则为 THERMAL_TREND_STABLE 趋势。另外，THERMAL_TREND_RAISE_FULL，THERMAL_TREND_DROP_FULL 这两种趋势状态，在 Allwinner 平台未使用。

3 Step-Wise 温控机制配置

3.1 DTS 配置

3.1.1 获取温度模块

在实现中，获取温度模块不再分为两个部分，而是直接配置成一个设备。这个设备包含 cpu，gpu 等所有的温度传感器，并为所有的传感器执行初始化，注册为 Thermal Zone Device 的操作。

```
# 版本发布时部分配置可能会存在差异，该示例仅供参考，请以实际为准
ths: ths@02009400 {
    compatible = "allwinner,sun20iw1p1-ths"; # 匹配驱动
    reg = <0x0 0x02009400 0x0 0x400>; # 寄存器资源
    clocks = <&ccu CLK_BUS_THS>;  # 时钟资源
    clock-names = "bus";
    resets = <&ccu RST_BUS_THS>;
    nvmem-cells = <&ths_calib>;  # 一个nvmem设备引用，用于读取校准值
    nvmem-cell-names = "calibration"; # 校准值的引用名
    #thermal-sensor-cells = <1>;
};
```

以上描述由 Allwinner 配置，均不需要修改。

3.1.2 控制温度模块

在新版实现中，控制温度的方式仍然是使用 cpu 降频降压等方式。但新实现中直接使用 Linux 通用的 cpu-cooling，dev-cooling 设备驱动，不再使用 Allwinner 实现的 sunxi_cooling_device 模块。

因此，不需要在 dts 中描述并抽象 cpu_budget_cooling，gpu_cooling 设备。这一部分内容也由 Allwinner 配置，且由于调频调压涉及系统稳定性，客户不可修改。

如需知道当前系统支持哪些调频的频率点，可通过读取 sysfs 节点获取，如下：

```
# 版本发布时部分配置可能会存在差异，该示例仅供参考，请以实际为准
root@TinaLinux:/# cat /sys/devices/system/cpu/cpufreq/policy0/scaling_available_frequencies
408000 720000 1008000
```

3.1.3 温度控制策略

在新版本中，温度控制策略描述仍然使用 Linux Thermal 提供的 of_thermal 模块来构建关系，故此处配置，仍然可参考上文旧版说明或 Linux Document 说明。

```
# 版本发布时部分配置可能会存在差异，该示例仅供参考，请以实际为准
thermal-zones {
    cpu_thermal_zone {
        polling-delay-passive = <500>; # 当温控策略激活时，即触发第一个trip后，温度采样间隔
        polling-delay = <1000>; # 当温控策略未激活时，温度采样间隔
        thermal-sensors = <&ths 0>; # 此 cpu_thermal_zone 对应的sensor

        trips{
            cpu_trip0:t0{ #第一个 trip 点
                temperature = <80000>;
                type = "passive";
                hysteresis = <0>;
            };
            cpu_trip1:t1{
                temperature = <90000>;
                type = "passive";
                hysteresis = <0>;
            };
            cpu_trip2:t2{
                temperature = <105000>;
                type = "passive";
                hysteresis = <0>;
            };
            crt_trip:shutdown{  # 最后一个 trip 点， 用于过温关机保护
                temperature = <110000>;
                type = "critical";
                hysteresis = <0>;
            };
        };
        cooling-maps {
            bind0{ #  trip0  对应的 降温状态
                contribution = <0>;
                trip = <&cpu_trip0>;
                cooling-device = <&CPU0 1 1>;  #cpufreq 的频点从最高频率到最小频点排序，从0开始标
    注，0对应最高频率
            };
            bind1{
                contribution = <0>;
                trip = <&cpu_trip1>;
                cooling-device = <&CPU0 2 2>;
            };
            bind2{
                contribution = <0>;
                trip = <&cpu_trip2>;
                cooling-device = <&CPU0 3 3>;
            };
        };
    };
};
```

3.2 内核配置

进入 tina 源码目录，执行 make kernel_menuconfig 进入配置主界面，并按以下步骤操作：

1、首先，进入到Device Drivers -> NVMEM Support，如下图所示：

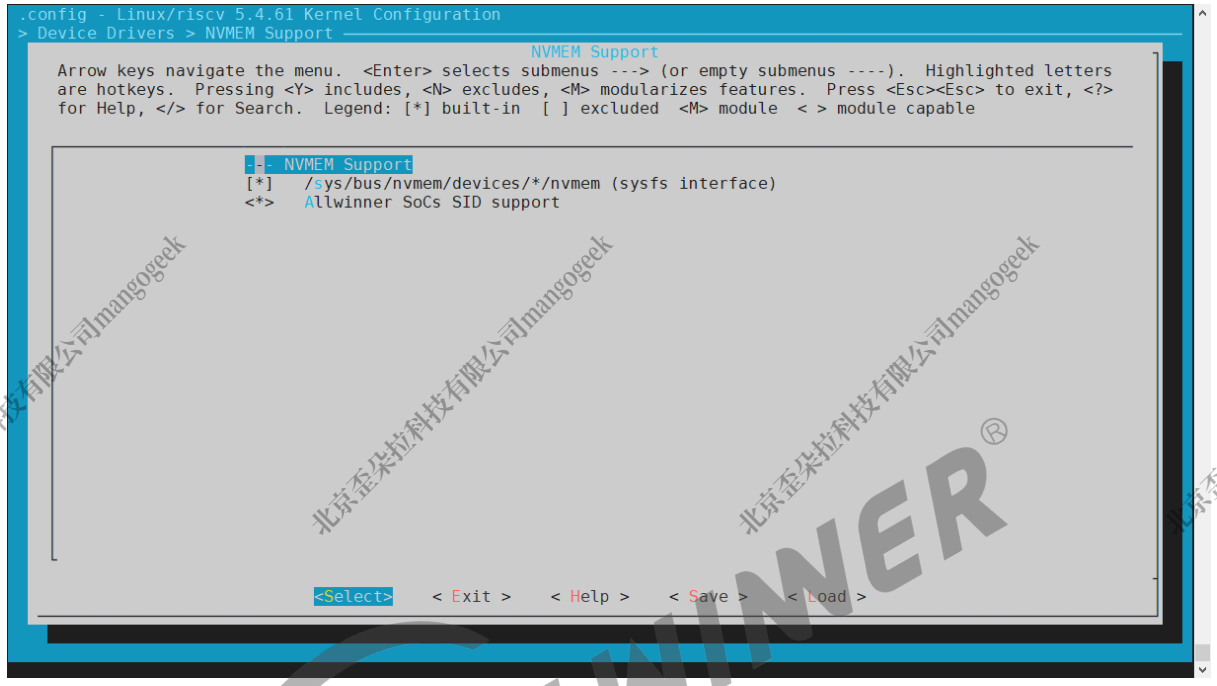

图 3-1: Thermal_menuconfig_D1_001

配置项说明：

```
[*]   /sys/bus/nvmem/devices/*/nvmem (sysfs interface) #sysfs 调试节点
<*>   Allwinner SoCs SID support  #使能 SID 驱动支持，主要用于读取校准值
```

2、进入到Device Drivers ->Generic Thermal sysfs driver，如下图所示：

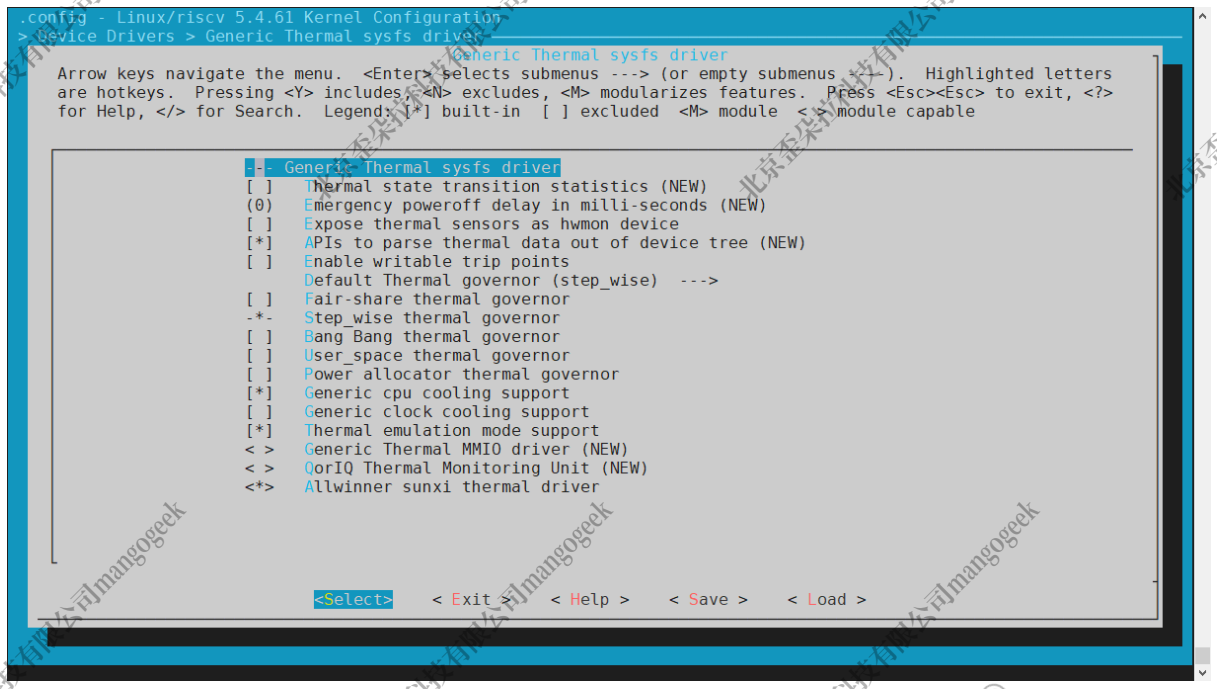

图 3-2: Thermal_menuconfig_D1_002

配置项说明：

```
    Default Thermal governor (step_wise)  ---> # 默认配置项，根据具体情况选择
[*]   Step_wise thermal governor # step-wise调频策略，配置后默认配置才能选择此策略
[*]   Generic cpu cooling support # 使能cpu降频
[*]   Thermal emulation mode support # 使能 模拟温度，调试用
<*>   Allwinner sunxi thermal driver  #使能 thermal driver
```

4 温控框架调试

4.1 基础说明

4.1.1 sysfs 节点

thermal_core会在/sys/devices/virtual/thermal目录下创建相应的节点，主要有thermal_zone和cooling_device两部分。

cooling_device的主要节点含义如下：

- type：降温设备的类型（处理器/风扇/...）
- max_state：降温设备的最大降温状态（以处理器为例，将 VF 表的数量划分成对应等级，最大降温等级就是 cpu 的最低电压和频率所在的等级）
- cur_state：降温设备的当前降温状态（以处理器为例，即当前的 cpu 电压和频率所在的等级）

thermal_zone的主要节点含义如下：

- type：温控区间的类型，比如cpu、gpu、ddr等
- temp：温控区间的当前温度
- mode：温控区间的工作状态，比如 enabled、disabled
- policy：温控区间的当前策略
- available_policies：温控区间支持的策略
- trip_point_[0-*]_temp：触发点[0-*]的温度
- trip_point_[0-*]_type：触发点[0-*]的类型，active、passive、hot、critical（没有风扇等主动设备的情况下一般使用passive类型表示被动降频降压降温，以及使用critical类型表示需要 shutdown 的温度）
- trip_point_[0-*]_hyst：触发点[0-*]的热滞系数，环境快速变化时的温度数据滞后现象，热滞系数和测温元件的质量，元件的比热容，热交换系数，元件的有效表面积有关，通常配置为 0
- emul_temp：模拟温度的节点，默认值是 0（设置成 0 就关闭模拟温度的功能）

4.2 基础操作

- 查看 thermal_zone 参数

不同平台温度 sensor 的个数及温度监控区域 thermal_zone 是不一样的。多个温度监控区域在/sys/class/thermal目录下就会有多个 thermal_zone*。当然，这些目录是/sys/devices/virtual/thermal/thermal_zone*的软连接。

下面以 thermal_zone0 为例，介绍几种常用的操作：

1、查看 thermal_zone0 的类型 (cpu_thermal_zone 为 cpu 温度域)

查看 cpu 温度时，先通过此节点确定其与thermal_zone*设备的对应关系，然后查看对应thermal_zone*设备的temp值即可。

```
#cat /sys/class/thermal/thermal_zone0/type
cpu_thermal_zone
```

2、查看 thermal_zone0 的温度

```
// 一般是5位数值，单位 千分之一摄氏度
#cat /sys/class/thermal/thermal_zone0/temp
24522
```

3、thermal_zone0 的温控 (开：enabled；关：disabled)

```
#echo disabled > /sys/class/thermal/thermal_zone0/mode
```

4、过温关机温度配置

配置关机温度只需要建立一个 trip，并将其类型配置为 critical 即可，具体可查看上文trips和binds说明一节描述。

```
crt_trip:t8{
    temperature = <110000>;
    type = "critical";
    hysteresis = <0>;
};
```

- 模拟温度

thermal 有温度模拟功能，可以通过模拟温度校验温度策略是否符合预期。

1、设置 thermal_zone0 的模拟温度

```
#echo 80 > /sys/class/thermal/thermal_zone0/emul_temp
// 注:  这里的数值单位与/sys/class/thermal/thermal_zone0/temp节点单位一致
```

2、关闭 thermal_zone0 的模拟温度功能

```
#echo 0 > /sys/class/thermal/thermal_zone0/emul_temp
```

著作权声明

本文档及内容受著作权法保护，其著作权由珠海全志科技股份有限公司（“全志”）拥有并保留一切权利。

本文档是全志的原创作品和版权财产，未经全志书面许可，任何单位和个人不得擅自摘抄、复制、修改、发表或传播本文档内容的部分或全部，且不得以任何形式传播。

商标声明


、 
、全志科技、  （不完全列举）均为珠海全志科技股份有限公司的商标或者注册商标。在本文档描述的产品中出现的其它商标，产品名称，和服务名称，均由其各自所有人拥有。

免责声明

您购买的产品、服务或特性应受您与珠海全志科技股份有限公司（“全志”）之间签署的商业合同和条款的约束。本文档中描述的全部或部分产品、服务或特性可能不在您所购买或使用的范围内。使用前请认真阅读合同条款和相关说明，并严格遵循本文档的使用说明。您将自行承担任何不当使用行为（包括但不限于如超压，超频，超温使用）造成的不利后果，全志概不负责。

本文档作为使用指导仅供参考。由于产品版本升级或其他原因，本文档内容有可能修改，如有变更，恕不另行通知。全志尽全力在本文档中提供准确的信息，但并不确保内容完全没有错误，因使用本文档而发生损害（包括但不限于间接的、偶然的、特殊的损失）或发生侵犯第三方权利事件，全志概不负责。本文档中的所有陈述、信息和建议并不构成任何明示或暗示的保证或承诺。

本文档未以明示或暗示或其他方式授予全志的任何专利或知识产权。在您实施方案或使用产品的过程中，可能需要获得第三方的权利许可。请您自行向第三方权利人获取相关的许可。全志不承担也不代为支付任何关于获取第三方许可的许可费或版税（专利税）。全志不对您所使用的第三方许可技术做出任何保证、赔偿或承担其他义务。