



AX620E 软件业务场景功耗调节指南

文档版本：V1.0

发布日期：2024/1/31

目 录

前 言	3
修订历史	4
1 AX620E 软件业务场景功耗调节概述	5
1.1 概述	5
1.2 软件调节方法概述	5
1.2.1 驱动层软件调节	5
2 驱动层软件调节方法	6
2.1 Devfreq	6
2.2 Cpufreq	10

权利声明

爱芯元智半导体股份有限公司或其许可人保留一切权利。

非经权利人书面许可，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。

注意

您购买的产品、服务或特性等应受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的购买或使用范围之内。除非商业合同另有约定，本公司对本文档内容不做任何明示或默示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

前言

适用产品

AX620E 系列产品（AX630C、AX620Q）

适读人群

- 软件开发工程师
- 技术支持工程师

符号与格式定义

符号/格式	说明
xxx	表示您可以参考的代码。
斜体	表示变量。如，“安装目录/AX620EA_SDK_Vx.x.x/build 目录”中的“安装目录”是一个变量，由您的实际环境决定。
说明/备注：	表示您在使用产品的过程中，我们向您说明的事项。
注意：	表示您在使用产品的过程中，需要您特别注意的事项。

修订历史

文档版本	发布时间	修订说明
V0.5	2023/8/15	文档初版
V1.0	2024/1/31	去掉应用层部分

1 AX620E 软件业务场景功耗调节概述

1.1 概述

芯片在某些业务场景下，由于功耗大导致发热量增加，若处于散热不足的环境中，就有可能会导致芯片越来越热而触发过热保护系统重启。为避免这种情况的出现，需要在检测到温度超过某个临界点时，用降低系统性能的方式来降低功耗，从而减小芯片发热，让芯片温度回落到安全的范围后，再将系统性能调整回初始的状态。这是业务场景下通过功耗调节来控制温度的基本方法，能起到增加系统稳定性，保护设备长期可靠运行的作用。

本文主要描述在 AX620E 业务场景下功耗调节的方法和需注意的事项。

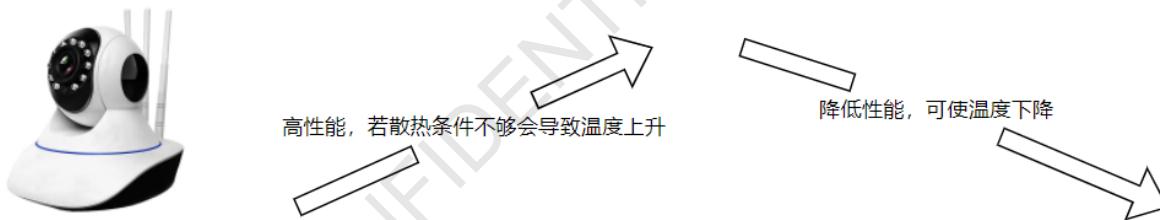


图1-1 AX620E 业务场景下功耗调节框图

1.2 软件调节方法概述

AX620E 业务场景功耗调节主要包括了温度监测、应用层软件功耗调节、驱动层软件功耗调节等几个方面。

1.2.1 驱动层软件调节

通过改变各模块工作频率和 DDR 速率的方法来改变系统功耗。

2 驱动层软件调节方法

2.1 Devfreq

复杂 SoC 由多个子模块协同工作组成。在执行各种用例的操作系统中，并非 SoC 中的所有模块都需要始终保持最高性能。为方便起见，将 SoC 中的子模块分组为域，从而允许某些域以较低的电压和频率运行，而其他域以较高的电压/频率对运行，对于这些设备支持的频率和电压对，称之为 OPP (Operating Performance Point)。对于具有 OPP 功能的非 CPU 设备，称之为 OPP device，需要通过 devfreq 进行动态的调频调压。devfreq 则支持多个设备，并且允许每个设备有自己对应的 governor。Linux 内核实现了完整的 devfreq 框架，AX620E 上采用了 userspace governor，在应用层创建了相应的文件节点来进行调频。

驱动层面的场景功耗调节，主要通过修改业务相关模块的工作频率，以及修改 ddr 速率来实现。基于 linux 的 devfreq 框架，各模块为应用层开辟了相应的功耗调节的文件节点。

模块名称	DFS 节点
CPU	/sys/devices/system/cpu/cpufreq/policy0
DDR	/sys/devices/platform/soc/soc:ddr_dfs/devfreq/soc:ddr_dfs/
ISP_TOP	/sys/devices/platform/soc/soc:isp_top/devfreq/soc:isp_top/
ISP_AXI	/sys/devices/platform/soc/soc:isp_axi/devfreq/soc:isp_axi/
ISP_IR	/sys/devices/platform/soc/soc:isp_ir/devfreq/soc:isp_ir/
VENC	/sys/devices/platform/soc/4010000.venc/devfreq/4010000.venc/
JENC	/sys/devices/platform/soc/4000000.jenc/devfreq/4000000.jenc/

VDEC	/sys/devices/platform/soc/19000000.vdec/devfreq/4020000.vdec
TDP	/sys/devices/platform/soc/4405000.tdp/devfreq/4405000.tdp/
GDC	/sys/devices/platform/soc/4404000.gdc/devfreq/4404000.gdc/
VPP	/sys/devices/platform/soc/4403000.vpp/devfreq/4403000.vpp/
DPU	/sys/devices/platform/soc/4407000.vo/devfreq/4407000.vo/
DPU_LITE	/sys/devices/platform/soc/4408000.vo/devfreq/4408000.vo/
NPU	/sys/devices/platform/soc/3800000.ax_npu/devfreq/3800000.ax_npu/
IVE	/sys/devices/platform/soc/4406000.ive/devfreq/4406000.ive/

表 2.1 AX620E 各模块 DFS 节点

通过 echo xxx(期望的频率值) > /sys/device/platform/soc/xxx/devfreq/xxx/userspace/set_freq 的方式，可以设置各模块的频点

通过 cat /sys/device/platform/soc/xxx/devfreq/xxx/available_frequencies 的方式，可获取各模块支持的频点

通过 cat /sys/device/platform/soc/xxx/devfreq/xxx/cur_freq 的方式，可获取各模块当前的频点

基于图像质量和硬件限制的考虑，各模块组合使用时，频点并不是随意设置的，在实验基础上，我们给出如下两组参考值，可保证在运行 BoxDemo、FRTDemo 下，动态切换频率，图像正常，各模块的参考频点设置如下：

模块名称	最高档(默认配置)	最低档
CPU	1200000000HZ	800000000HZ

DDR	3200000000HZ	1600000000HZ
ISP_TOP	416000000HZ	416000000HZ
ISP_AXI	533333333HZ	312000000HZ
ISP_IR	297000000HZ	100000000HZ
VENC	533333333HZ	208000000HZ
JENC	533333333HZ	208000000HZ
VDEC	533333333HZ	208000000HZ
TDP	533333333HZ	208000000HZ
GDC	533333333HZ	208000000HZ
VPP	533333333HZ	208000000HZ
DPU	533333333HZ	208000000HZ
DPU_LITE	533333333HZ	208000000HZ
NPU	800000000HZ	594000000HZ
IVE	533333333HZ	208000000HZ

表 2.2 AX620E 可选频点组合

各模块可选的频点值，可通过 dts 文件进行配置，kernel/linux/linux5.15.73/arch/arm64/boot/dts/axera/ AX620E_opptable.dtsi 下是当前各模块可选的频点

为方便应用层配置上述频点，我们在/usr/bin/下提供了 ax_clk 脚本，可以用来快速配置上述频点组合，可参考脚本修改添加需要的频点，脚本使用方法如下：

输入	功能
ax_clk	查询所有模块 clk
ax_clk -q	查询所有模块 clk
ax_clk -q <module>	查询指定模块 clk 模块名称(不区分大小写): CPU / VENC / JENC / VDEC / TDP / GDC/VPP/DPU/DPU_LITE/NPU/IVE /ISP_TOP/ISP_AXI/ISP_IR/DDR
ax_clk -s <groupX>	设置已配好组别的 clk, 详细查看/usr/bin/ax_clk 的定义
ax_clk -s <module> <clk>	设置指定模块 clk 模块名称(不区分大小写): CPU / VENC / JENC / VDEC / TDP / GDC/VPP/DPU/DPU_LITE/NPU/IVE /ISP_TOP/ISP_AXI/ISP_IR/DDR

表 2.3 ax_clk 使用方法

2.2 Cpufreq

针对 CPU，AX620E 上采用了 linux cpufreq 框架进行调频，在应用层生成了用于调频的文件节点

应用层可通过如下接口进行相应 CPU 的频率的设置

cat /sys/devices/system/cpu/cpufreq/policy0/scaling_available_frequencies, 来查看对应 cpu 支持的频率

echo xxx > /sys/devices/system/cpu/cpufreq/policy0/scaling_setspeed, 设置需要的 cpu 频率

cat /sys/devices/system/cpu/cpufreq/policy0/scaling_cur_freq, 来查看当前 CPU 的频率

各 CPU 共享同一套频率电压组合，详见 kernel/linux/linux5.15.73/arch/arm64

/boot/dts/axera/AX620E_opptable.dtsi

```
cpu_opp_table: cpu_opp_table {
    compatible = "operating-points-v2";
    opp-shared;

    opp00 {
        opp-hz = /bits/ 64 <24000000>;
    };
    opp01 {
        opp-hz = /bits/ 64 <208000000>;
    };
    opp02 {
        opp-hz = /bits/ 64 <500000000>;
    };
    opp03 {
        opp-hz = /bits/ 64 <800000000>;
    };
    opp04 {
        opp-hz = /bits/ 64 <1200000000>;
    };
};
```