thermal management debug SOP V1.0 (CH)

Programming Guide

Customer Support

MT6757/MT6580/MT6755

Doc No: DS6000-AW2G-DMT-V1.0ZH
Version: V1.0
Release date: 2017-03-28
Classification: Internal

thermal management debug SOP V1.0 (CH)

Template, Common Part

MediaTek Inc.

Postal address

No. 1, Dusing 1st Rd. , Hsinchu Science Park, Hsinchu City, Taiwan 30078

MTK support office address

No. 1, Dusing 1st Rd. , Hsinchu Science Park, Hsinchu City, Taiwan 30078

Internet

http://www.mediatek.com/

Document Revision History

Revision	Date	Author	Description
A1	2017-03-28	Haoran.Dong	Initial Release

Table of Contents

Document Revision History **3**
Table of Contents **4**
Lists of Tables **6**
Lists of Figures **7**
1 Introduction **8**
1.1 Purpose 8
1.2 Scope 8
1.3 Who Should Read This Document 9
1.4 How to Use This Manual 9
1.4.1 Terms and Conventions 9
2 References **10**
3 Definitions **11**
4 Abbreviations **12**
5 Overview **13**
6 Architecture **14**
6.1 Architecture 14
6.2 Files for android “L” 14
6.3 Files for android “M” 15
7 debug 手法-通过 Proc fs 进行 debug **16**
7.1 通过 Proc fs 进行 debug(1) 16
7.2 通过 Proc fs 进行 debug (2) 16
7.3 通过 Proc fs 进行 debug (3) 17
7.4 通过 Proc fs 进行 debug (4) 17
7.5 常用的 Proc node (1) 18
7.6 常用的 Proc node (2) 18
7.7 常用的 Proc node (3) 19
7.8 常用的 Proc node (4) 19
7.9 常用的 Proc node (5-1) 20
7.10 常用的 Proc node (5-2) 20

7.11 常用的 Proc node (6) 21
8 debug 手法-通过 Sys fs 进行 debug 22
8.1 通过 sys fs 进行 debug (1) 22
8.2 通过 sys fs 进行 debug (2) 22
8.3 通过 sys fs 进行 debug (3) 23
8.4 通过 sys fs 进行 debug (3) 23
9 debug 手法-Elephant stress tool 24
9.1 Elephant Stress tool (1) 24
9.2 Elephant Stress tool (2) 24
9.3 Elephant Stress tool (3) 25
9.4 Elephant Stress tool (4) 25
9.5 Elephant Stress tool (5) 26
10 案例分析 27
10.1 hardware reboot (1) 27
10.2 hardware reboot (2) 27
10.3 hardware reboot (3) 28
10.4 mtktsAP 温度采样不准 (1) 28
10.5 mtktsAP 温度采样不准 (2) 29
10.6 mtktsAP 温度采样不准 (3) 29
10.7 mtktsAP 温度采样不准 (4) 30
10.8 thermal 引发 kernel panic (1) 30
10.9 thermal 引发 kernel panic (2) 31
10.10 thermal 引发 kernel panic (3) 31
10.11 thermal 引发 kernel panic (4) 32

Lists of Tables

Table 1-1. Reference Information beyond Scope 8
Table 1-2. Chapter Overview 9
Table 1-3. Conventions 9
Table 4-1. Abbreviations 12

Lists of Figures

1 Introduction

☞ [Random filler text. Not intended for actual reading.] Must keep the chapter even it have empty content.

1.1 Purpose

This document provides the programming guidelines for thermal management debug and associated modules. It describes how to bring up thermal management debug on the Android platform.

[Random filler text. Not intended for actual reading.] This section should describe about the following items:

- What does this document provide?
- What should the reader get after reading the document?
- Any concrete outcome (gains, or applications) can get after step-by-step following the document?

1.2 Scope

The document provide the programming details of the how to handle Chinese contact name in heteronym.

Table 1-1 presents the reference information of the modules which are used but beyond the scope.

Table 1-1. Reference Information beyond Scope

Modules	Reference information

[Random filler text. Not intended for actual reading.] This section should describe about the following items:

- What hardware and version is this document applying? The hardware can be MTK chipset or external module. Please don't put multiple MTK chipset information here. If this document can apply multiple MTK chipset, please duplicates this document for each chipset and modify the chipset information.
- What platform and version is this document applying? For example, Android platform version and kernel version.
- What hardware and software module is used but beyond the scope? Please describe them and add reference after the module.

1.3 Who Should Read This Document

This document is primarily intended for:

- Engineers with technical knowledge of the heteronym contacts
- Customers who handle Chinese contact name in heteronym

1.4 How to Use This Manual

This segment explains how information is distributed in this document, and presents some cues and examples to simplify finding and understanding information in this document. Table 1-2 presents an overview of the chapters and appendices in this document.

Table 1-2. Chapter Overview

#	Chapter	Contents

1.4.1 Terms and Conventions

This document uses special terms and typographical conventions to help you easily identify various information types in this document. These cues are designed to simply finding and understanding the information this document contains.

Table 1-3. Conventions

Convention	Usage	Example

2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

需要先阅读的文档

[1] Thermal_Management_MTxxxx

[2] Thermal Policy Tuning Guide V1.0

[3] The thermal_conf file introduce

上述文档，可以在敝司 DCC 网站上获取

☞ [Random filler text. Not intended for actual reading.] Must keep the chapter even it have empty content.

3 Definitions

For the purposes of the present document, the following terms and definitions apply:

☞ [Random filler text. Not intended for actual reading.] Must keep the chapter even it have empty content.

4 Abbreviations

Please note the abbreviations and their explanations provided in Table 4-1. They are used in many fundamental definitions and explanations in this document and are specific to the information that this document contains.

Table 4-1. Abbreviations

Abbreviations	Explanation
MTK	MediaTek, Asia's largest fabless IC design company.

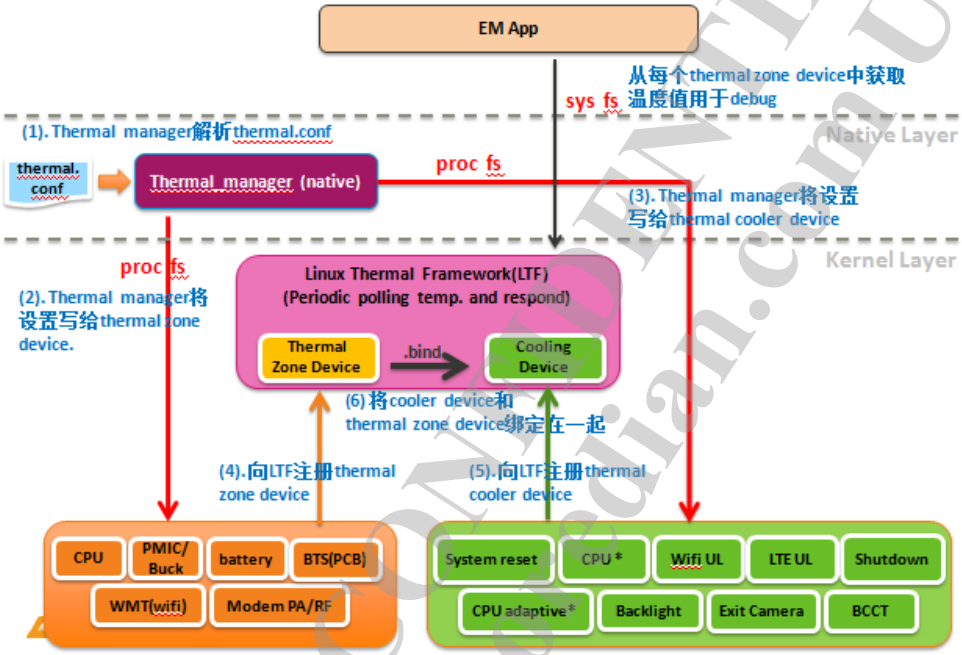
☞ [Random filler text. Not intended for actual reading.] Must keep the chapter even it have empty content.

5 Overview

This chapter first gives a brief description of the ??? functions and then a description of the external interfaces of ???.

6 Architecture

6.1 Architecture



6.2 Files for android "L"

LTF files:
- /kernel-3.10/driver/thermal/thermal_core.c
- /kernel-3.10/driver/thermal/backward_compatible.c

MTK files:
- /kernel-3.10/drivers/misc/mediatek/thermal/mtk_thermal_monitor.c
- /kernel-3.10/drivers/misc/mediatek/thermal/mtk_cooler_shutdown.c
- /kernel-3.10/drivers/misc/mediatek/thermal/mtk_cooler_kshutdown.c
- /kernel-3.10/drivers/misc/mediatek/thermal/mtk_cooler_backlight.c
- /kernel-3.10/drivers/misc/mediatek/thermal/[$Platform]/mtk_thermal_platform.c
- /kernel-3.10/drivers/misc/mediatek/thermal/[$Platform]/mtk_ts_pa_thput.c
- /kernel-3.10/drivers/misc/mediatek/thermal/[$Platform]/mtk_ts_pmic.c
- /kernel-3.10/drivers/misc/mediatek/thermal/[$Platform]/mtk_ts_bts.c
- /kernel-3.10/drivers/misc/mediatek/thermal/[$Platform]/mtk_ts_bstmdpa.c
- /kernel-3.10/drivers/misc/mediatek/thermal/[$Platform]/mtk_ts_all_ts.c
- /kernel-3.10/drivers/misc/mediatek/thermal/[$Platform]/mtk_ts_pa.c
- /kernel-3.10/drivers/misc/mediatek/thermal/[$Platform]/mtk_ts_battery.c
- /kernel-3.10/drivers/misc/mediatek/thermal/[$Platform]/mtk_ts_cpu.c
- /kernel-3.10/drivers/misc/mediatek/thermal/[$Platform]/mtk_cooler_bcct.c
- /kernel-3.10/drivers/misc/mediatek/thermal/[$Platform]/mtk_cooler_mutt.c
- /kernel-3.10/drivers/misc/mediatek/thermal/[$Platform]/mtk_cooler_amutt.c

6.3 Files for android "M"

LTF files:
- /kernel-3.18/driver/thermal/thermal_core.c
- /kernel-3.18/driver/thermal/backward_compatible.c

MTK files:
- /kernel-3.18/drivers/misc/mediatek/thermal/mtk_thermal_monitor.c
- /kernel-3.18/drivers/misc/mediatek/thermal/mtk_cooler_shutdown.c
- /kernel-3.18/drivers/misc/mediatek/thermal/mtk_cooler_kshutdown.c
- /kernel-3.18/drivers/misc/mediatek/thermal/mtk_cooler_backlight.c
- /kernel-3.18/drivers/misc/mediatek/thermal/common/mtk_thermal_platform.c
- /kernel-3.18/drivers/misc/mediatek/thermal/common/thermal_zones/mtk_ts1.c
... ...
- /kernel-3.18/drivers/misc/mediatek/thermal/common/thermal_zones/mtk_ts_bts.c
- /kernel-3.18/drivers/misc/mediatek/thermal/common/thermal_zones/mtk_ts_cpu.c
- /kernel-3.18/drivers/misc/mediatek/thermal/common/thermal_zones/mtk_ts_pa.c
- /kernel-3.18/drivers/misc/mediatek/thermal/common/thermal_zones/mtk_ts_pmic.c
- /kernel-3.18/drivers/misc/mediatek/thermal/common/coolers/mtk_cooler_amutt.c
- /kernel-3.18/drivers/misc/mediatek/thermal/common/coolers/mtk_cooler_mutt.c
- /kernel-3.18/drivers/misc/mediatek/thermal/common/coolers/mtk_cooler_bcct.c
- /kernel-3.18/drivers/misc/mediatek/thermal/common/coolers/mtk_cooler_atm.c
- /kernel-3.18/drivers/misc/mediatek/thermal/common/coolers/mtk_cooler_dtm.c
- /kernel-3.18/drivers/misc/mediatek/thermal/common/coolers/mtk_cooler_sysstst.c

7 debug 手法-通过 Proc fs 进行 debug

7.1 通过 Proc fs 进行 debug(1)

- 在每个thermal zone device和cooler device init的时候，会创建相应的proc fs (**/proc/driver/thermal/**)
- thermal manager就是通过这些proc fs来将thermal.conf中的设置写给每个device的。
- 对于thermal policy的修改，虽然可以通过Proc fs来做。但是，不直观。建议直接对thermal.conf文件进行修改。
(关于thermal.conf的修改方法，参考文档" The thermal_conf file introduce")

在debug过程中，Proc fs的主要用途：

1. 查看当前thermal相关的各device的参数设置
2. 更新thermal.conf文件后，查看新的参数是否有正确下给各device

7.2 通过 Proc fs 进行 debug (2)

ls /proc/driver/thermal/ 可以看的所有的device的proc node

```
clamutt_asparam
clamutt_dbg
clamutt_param
clatm
clatm_gpu_threshold
clatm_setting
clbcct
clctm
clmutt
clsd_dbt
clsd_pid
clsd_rst
clwmt_pid
clwmt_val
clwmt_wfdstat
mdm_mdinfo
mdm_mdinfoex
mdm_mdinfoex_thre
mdm_sw
mdm_timeout
mdm_value
mtm_indicator
```

```
mtm_indicator
mtm_monitor
mtm_scen_call
tzbattery
tzbts
tzbts_param
tzbtspa
tzbtspa_param
tzcpu
tzcpu_Tj_out_via_HW_pin
tzcpu_cal
tzcpu_fastpoll
tzcpu_log
tzcpu_read_temperature
tzcpu_set_temperature
tzcpu_talking_flag
tzpa
tzpmic
tzpmic_log
tzts1
tzts2
tzts3
tzwmt
wifi_tx_thro
```

7.3 通过 Proc fs 进行 debug (3)

Thermal cooler device name和proc node之间的对应关系

mtk-cl-bcct00	/proc/driver/thermal/clbcct
mtk-cl-bcct01	/proc/driver/thermal/clbcct
mtk-cl-bcct02	/proc/driver/thermal/clbcct
mtktswmt-pa1	/proc/driver/thermal/clwmt_val
mtktswmt-pa2	/proc/driver/thermal/clwmt_val
cpu_adaptive_0	/proc/driver/thermal/clatm_setting
cpu_adaptive_1	/proc/driver/thermal/clatm_setting
cpu_adaptive_2	/proc/driver/thermal/clatm_setting
mtktsAP	/proc/driver/thermal/tzbts_param
mtktsbtsmdpa	/proc/driver/thermal/tzbtspa_param
cl-amutt-upper	/proc/driver/thermal/clamutt_param
cl-amutt-lower	/proc/driver/thermal/clamutt_param
cl-amutt-asparam	/proc/driver/thermal/clamutt_asparam
mtk-cl-mutt00	/proc/driver/thermal/clmutt
mtk-cl-mutt01	/proc/driver/thermal/clmutt
mtk-cl-mutt02	/proc/driver/thermal/clmutt
mtk-cl-cam00	/proc/driver/cl_cam
ctm	/proc/driver/thermal/clctm

7.4 通过 Proc fs 进行 debug (4)

Thermal zone device name和proc node之间的对应关系

mtktscpu	/proc/driver/thermal/tzcpu
mtktsabb	/proc/driver/thermal/tzabb
mtktspmic	/proc/driver/thermal/tzpmic
mtktsbattery	/proc/driver/thermal/tzbattery
mtktspa	/proc/driver/thermal/tzpa
mtktswmt	/proc/driver/thermal/tzwmt
mtktsAP	/proc/driver/thermal/tzbts
mtktsbuck	/proc/driver/thermal/tz6311
mtkts1	/proc/driver/thermal/tzts1
mtkts2	/proc/driver/thermal/tzts2
mtkts3	/proc/driver/thermal/tzts3
mtkts4	/proc/driver/thermal/tzts4
mtktsbtsmdpa	/proc/driver/thermal/tzbtspa

7.5 常用的 Proc node (1)

查看charging限流的cooler "clbcct"的参数
cat /proc/driver/thermal/clbcct

当前限制的电流大小。
65535表示没有限流。

```
klog 0
curr_limit 65535
mtk-cl-bcct00 650 mA, state 0
mtk-cl-bcct01 450 mA, state 0
mtk-cl-bcct02 300 mA, state 0
```

state表示对应的cooler是有active。
0表示没有active
1表示有active

mtk-cl-bcctXX的limit电流，跟excel中
mtk-cl-bcct00 table中的设置是一样的。

Cooler Name	mtk-cl-bcct00	ENABLE
Extra	klog on	0
Extra	mtk-cl-bcct00 limit (mA)	650
Extra	mtk-cl-bcct01 limit (mA)	450
Extra	mtk-cl-bcct02 limit (mA)	300

7.6 常用的 Proc node (2)

查看ctm的设置
cat /proc/driver/thermal/clctm

ctm打印出的数据跟excel中ctm table是一一对应的。

```
ctm 1
Target_Tj_0 85000
Target_Tj_2 72000
Tpcb_1 39999
Tpcb_2 41999
Exit_Tj_0 75000
Exit_Tj_2 62000
Enter_a 344994
Enter_b 6500
Exit_a 334994
Exit_b 6500
```

Cooler Name	ctm	ENABLE
Extra	ctm on	1
Extra	Target Tj 0	85000
Extra	Target Tj 2	72000
Extra	Tpcb 1	39999
Extra	Tpcb 2	41999
Extra	Exit Tj 0	75000
Extra	Exit Tj 2	62000
Extra	Enter_a	344994
Extra	Enter_b	6500
Extra	Exit_a	334994
Extra	Exit_b	6500

7.7 常用的 Proc node (3)

查看LTE uplink限速cooler “clmutt”的设置

cat /proc/driver/thermal/clmutt

```
cat clmutt
klog 0
curr_limit 0
mtk-cl-mutt00 4 1 10401, state 0
mtk-cl-mutt01 1 2 20101, state 0
mtk-cl-mutt02 1 4 40101, state 0
```

当前限制的LET uplink data rate大小。
0表示没有限速。

state表示对应的cooler是有active。
0表示没有active
1表示有active

Cooler Name	mtk-cl-mutt00	ENABLE
Extra	klog on	0
Extra	mutt00 active ms	400
Extra	mutt00 suspend ms	100
Extra	mutt01 active ms	100
Extra	mutt01 suspend ms	200
Extra	mutt02 active ms	100
Extra	mutt02 suspend ms	400

7.8 常用的 Proc node (4)

查看atm的设置（即：cpu_adaptive_XX的设置）

cat /proc/driver/thermal/clatm_setting

```
cat clatm_setting
cpu_adaptive_00
 first_step = 1600
 theta rise = 15
 theta fall = 30
 min_budget_change = 1
 n cpu = 800
 M cpu = 1600
 n gpu = 600
 M gpu = 900
cpu_adaptive_01
 first_step = 0
 theta rise = 0
 theta fall = 0
 min_budget_change = 0
 n cpu = 0
 M cpu = 0
 n gpu = 0
 M gpu = 0
cpu_adaptive_02
 first_step = 0
 theta rise = 0
 theta fall = 0
 min_budget_change = 0
 n cpu = 0
 M cpu = 0
 n gpu = 0
```

Cooler Name	cpu_adaptive_0	ENABLE
Extra	id	0
Extra	First Step (mW)	1600
Extra	Theta(ja) Fall	15
Extra	Theta(ja) Rise	30
Extra	Min Budget Change	1
Extra	Min CPU Power (mW)	800
Extra	Max CPU Power (mW)	1600
Extra	Min GPU Power (mW)	600
Extra	Max GPU Power (mW)	900

在使用cATM算法的情况下，
cpu_adaptive_01和cpu_adaptive_02是disable的。
因此，这两个cooler的参数全为0

7.9 常用的 Proc node (5-1)

查看thermal zone "tzbts"的设置
cat /proc/driver/thermal/tzbts

[mtkts_bts_read]
trip_0_temp=95000,trip_1_temp=50000,trip_2_temp=48000,trip_3_temp=34000,
trip_4_temp=80000,trip_5_temp=70000,trip_6_temp=65000,trip_7_temp=60000,
trip_8_temp=55000,trip_9_temp=50000, trip point temp的设定

g_THERMAL_TRIP_0=0,g_THERMAL_TRIP_1=0,g_THERMAL_TRIP_2=0,g_THERMAL_TRIP_3=0,
g_THERMAL_TRIP_4=0,g_THERMAL_TRIP_5=0,g_THERMAL_TRIP_6=0,g_THERMAL_TRIP_7=0,
g_THERMAL_TRIP_8=0,g_THERMAL_TRIP_9=0, 没有实际意义，debug时请忽略

cooldev0=mtktspa-sysrst,cooldev1=mtk-cl-bcct02,cooldev2=mtk-cl-bcct01,
cooldev3=mtk-cl-bcct00,
cooldev4=no-cooler,cooldev5=no-cooler,cooldev6=no-cooler,cooldev7=no-cooler,
cooldev8=no-cooler,cooldev9=no-cooler, 其下绑定的cooler device

time_ms=1000 预设的polling time

7.10 常用的 Proc node (5-2)

[mtkts_bts_read]
trip_0_temp=95000,trip_1_temp=50000,trip_2_temp=48000,trip_3_temp=34000,
trip_4_temp=80000,trip_5_temp=70000,trip_6_temp=65000,trip_7_temp=60000,
trip_8_temp=55000,trip_9_temp=50000,

cooldev0=mtktspa-sysrst,cooldev1=mtk-cl-bcct02,cooldev2=mtk-cl-bcct01,
cooldev3=mtk-cl-bcct00,
cooldev4=no-cooler,cooldev5=no-cooler,cooldev6=no-cooler,cooldev7=no-cooler,
cooldev8=no-cooler,cooldev9=no-cooler,

其中，trip point temp的设置和cooldev的设置，
跟excel中是一样的。

mtktsAP
95000
mtktspa-sysrst
50000
mtk-cl-bcct02
48000
mtk-cl-bcct01
34000
mtk-cl-bcct00

7.11 常用的 Proc node (6)

查看AP的NTC设置（即：mtktsAP的NTC设置）
cat /proc/driver/thermal/tzbts_param

```
cat tzbts_param
390000
1800
4251000
6
0
```

NTC	mtktsAP	ENABLE
Extra	Param	PUP_R
Extra	Value	390000
Extra	Param	PUP_VOLT
Extra	Value	1800
Extra	Param	OVER_CRITICAL_L
Extra	Value	4251000
Extra	Param	NTC_TABLE
Extra	Value	6
Extra	AP ADC Channel	0

这个proc可以用于检测NTC设置。
对于处理mtktsAP温度采样不准的问题，很有帮忙。

8 debug 手法-通过 Sys fs 进行 debug

8.1 通过 sys fs 进行 debug (1)

在thermal zone device driver和thermal cooler device driver向 thermal core做register的时候，thermal core会为每一个device创建sys fs (**/sys/class/thermal/**)

在debug过程中，sys fs的主要用途：
查看各个thermal zone的即时温度值

8.2 通过 sys fs 进行 debug (2)

ls /sys/class/thermal/

```
cooling_device58
cooling_device59
cooling_device6
cooling_device60
cooling_device61
cooling_device62
cooling_device7
cooling_device8
cooling_device9
thermal_zone0
thermal_zone1
thermal_zone2
thermal_zone3
thermal_zone4
thermal_zone5
thermal_zone6
thermal_zone7
thermal_zone8
thermal_zone9
```

cooler device，debug时通常不使用。
可以忽略

在LTF中注册的thermal zone device
(可以看到有注册了10个thermal zone device)

这里的thermal_zoneX不是file，是folder。

8.3 通过 sys fs 进行 debug (3)

使用如下方法，
可以知道thermal_zoneX具体是哪个device；以及其即时温度值

cd /sys/class/thermal/
cat thermal_zone*/type

```
cat thermal_zone*/type
mtktswmt
mtktspmic
mtkts1
mtkts2
mtkts3
mtktspa
mtktsbattery
mtktscpu
mtktsAP
mtktsbtsmdpa
```

cd /sys/class/thermal/
cat thermal_zone*/temp

```
cat thermal_zone*/temp
36000
38705
47400
52000
44500
-127000
29000
52000
31600
32000
```

单位：毫摄氏度

-127度表示无效的温度（是modem没有注册网络）。

8.4 通过 sys fs 进行 debug (3)

9 debug 手法- Elephant stress tool

9.1 Elephant Stress tool (1)

为了减小内核的log量，
thermal management default将kernel部分的debug log都关闭了。

只留下了一些，error和warning log。

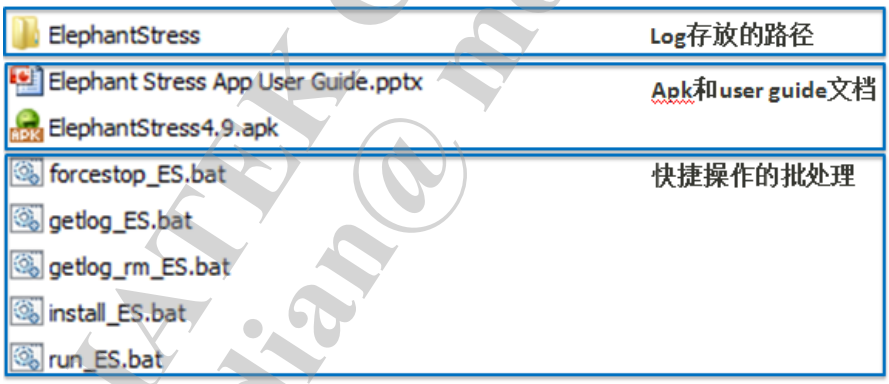
而有些问题的分析和thermal policy的tuning是需要完整的thermal log的。
特别是各个thermal zone device的温度值。

为此可以使用Elephant Stress工具来抓取完整的thermal log。
(user load也可以使用)

注：Elephant stress tool可以在敝司DDC网站上下载。

9.2 Elephant Stress tool (2)

在elephant stress工具包中，有如下文件：



9.3 Elephant Stress tool (3)

使用elephant stress tool抓取log，简介SOP：

1. 运行工具中的脚本文件"install_ES.bat",来向手机中安装该APK
2. 点击该APK图标（该图标是一个"重"字），并在右上角的设置中，设置no CPU stress，并且时间设置为no timeout
3. 点击start
4. 跑所需的测试项
5. 点击stop
6. 运行工具中的脚本文件"getlog_ES.bat",会将log捞到工具目录下的elephant stress文件夹下

按照上述SOP操作后，在elephant stress文件夹下，可以看到如下形式的文件。

2010-1-1-2-56-28.csv
2015-1-1-0-1-57.csv
2015-1-1-19-18-48.csv
2015-1-1-19-19-42.csv
2015-1-12-17-55-54.csv

PS: 该文件已手机当前时间的时间戳命名

9.4 Elephant Stress tool (4)

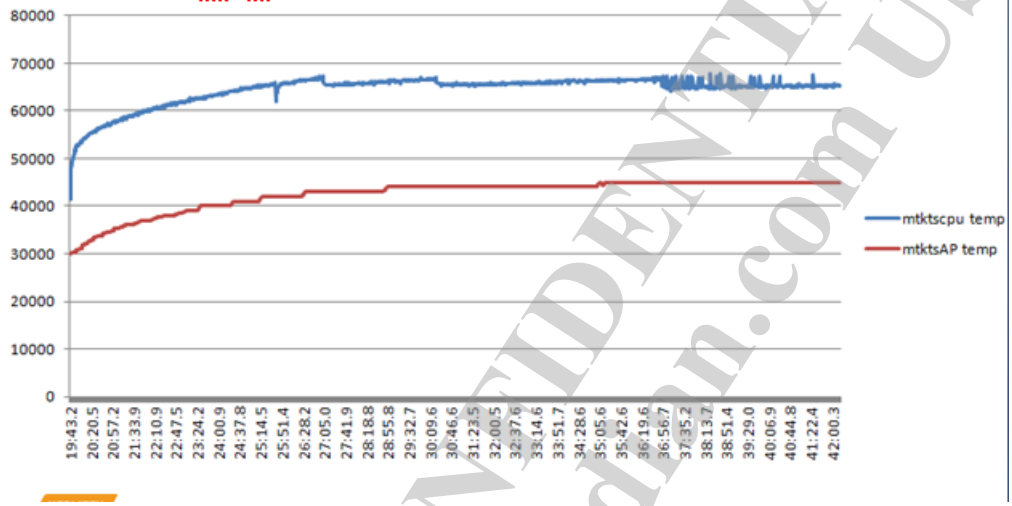
打开log文件，可以看到如下信息：

Time	WU	mtktswmt	mtktspmic	mtkts1	mtkts2	mtkts3	mtktspa	mtktsbatt	mtktscpu	mtktsAP	mtktsbtsr
19:43.2	214	33000	37096	41200	41500	38900	-127000	29500	41500	30000	31000
19:44.3	352.9	33000	37096	41200	41500	38900	-127000	29500	47800	30000	31000
19:45.5	342.4	33000	37096	44800	48900	41600	-127000	29750	48900	30200	31000
19:46.7	250.4	33000	37096	44800	48900	41600	-127000	29750	49500	30200	31000
19:47.8	276.1	33000	37827	45700	50000	42600	-127000	29750	50000	30200	31000
19:49.0	299.5	33000	37827	45700	50000	42600	-127000	29750	50400	30200	31000
19:50.1	331.5	33000	37827	46900	50900	43300	-127000	30000	50900	30400	31000
19:51.2	321.8	33000	37827	47400	51800	43900	-127000	30000	51800	30400	31000
19:52.3	338.2	33000	37827	47400	51800	43900	-127000	30000	51600	30400	31000
19:53.4	338.2	33000	38412	47900	52000	44500	-127000	30000	52000	30400	31000
19:54.5	354.5	33000	38412	47900	52000	44500	-127000	30000	52600	30800	32000
19:55.5	339.6	33000	38412	48300	52600	44700	-127000	30000	52600	30800	32000
19:56.6	349.6	33000	38412	48300	52600	44700	-127000	30000	52900	30800	32000
19:57.7	341.9	33000	38412	48400	52700	45000	-127000	30000	52700	30800	32000
19:58.8	341.3	33000	39143	48400	52700	45000	-127000	30000	53000	30800	32000
19:59.8	340.4	33000	39143	48700	53200	45600	-127000	30000	53200	31200	32000
20:00.9	322	33000	39143	48700	53200	45600	-127000	30000	53200	31200	32000
20:02.0	315.1	33000	39143	49400	53600	45800	-127000	30000	53600	31200	32000
20:03.1	333.3	33000	39581	49400	53600	45800	-127000	30000	53300	31200	32000

9.5 Elephant Stress tool (5)

可以借助excel的图表功能，绘制变化曲线。这样可以更直观的观察。

下图为所绘制的，cpu和bts的温度关系曲线：



10 案例分析

10.1 hardware reboot (1)

在发生这类问题时，会在aee_exp folder下面生成db文件

db.fatal.00.HW_Reboot
db.fatal.01.HW_Reboot
db.fatal.02.HW_Reboot
db.fatal.03.HW_Reboot

使用QAAT来对db进行解析

__exp_main.txt
INTERRUPT_NUM_NAME_MAPS
KERNEL_CONFIG.gz
PROC_CUR_TSK
SYS_KERNEL_LOG_RAW
SYS_LAST_KMSG
SYS_MINI_RDUMP
SYS_RAMCONSOLE_RAW
SYS_REBOOT_REASON
SYS_VERSION_INFO
ZAEE_LOG
ZZ_INTERNAL

```
Exception Class: Hardware Reboot
Exception Type: Thermal Reboot

WDT status: 32 fiq step: 0 exception type: 0
[LAST PC] CORE_0 PC = 0x7ff0004( + 0x0), FP = 0x7f12ef0, SP = 0x0
[LAST PC] CORE_1 PC = 0xffffffc0003233dc(do_raw_spin_trylock + 0x2c)
[LAST PC] CORE_2 PC = 0xffffffc0003233e4(do_raw_spin_trylock + 0x34)
[LAST PC] CORE_3 PC = 0xffffffc000083200(vectors + 0x200), FP = 0x0,
[LAST PC] CORE_4 PC = 0xffffffc000083200(vectors + 0x200), FP = 0xff
```

断定是thermal引发hardware reboot有两个条件：
1. Exception Type是Thermal Reboot
2. WDT status是32

这两个条件必须都满足，才能断定是thermal引发的hardware reboot.

10.2 hardware reboot (2)

由于console存在已知的bug(已有patch修正)，会误报Thermal reboot

如在__exp_main.txt中发现

```
Exception Class: Hardware Reboot
Exception Type: Thermal Reboot

WDT status: 28 fiq step: 0 exception type: 0
[LAST PC] CORE_0 PC = 0xa0a0a0a0a0a0a0a( + 0x0), FP = 0x
[LAST PC] CORE_1 PC = 0xa0a0a0a0a0a0a0a( + 0x0), FP = 0x
[LAST PC] CORE_2 PC = 0xa0a0a0a0a0a0a0a( + 0x0), FP = 0x
[LAST PC] CORE_3 PC = 0xa0a0a0a0a0a0a0a( + 0x0), FP = 0x
[LAST PC] CORE_4 PC = 0xa0a0a0a0a0a0a0a( + 0x0), FP = 0x

thermal_temp1 = 31
thermal_temp2 = 32
thermal_temp3 = 29
thermal_temp4 = 31
thermal_temp5 = 31
thermal status: 2
```

(1) Exception type虽然是Thermal reboot，但是，WDT status是28

(2) thermal_tempX的数值在较小的范围内
PS: 如果是255的，表示这个thermal_temp没有被使用

满足上面2个条件的话，就可以断定是误报的thermal reboot。
需要先上patch，再来复测。

10.3 hardware reboot (3)

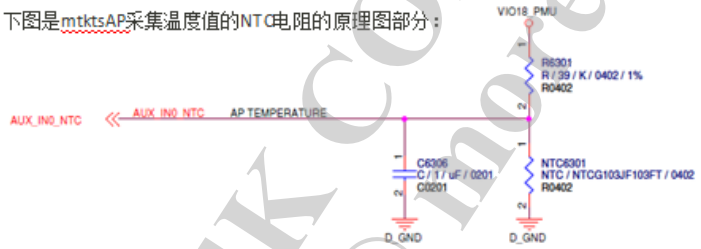
修正console误报Thermal reboot的patch如下：

Patch ID	SW version
ALPS02141523	L0.MP6 L0.MP8 L1.MP2 L1.MP3 L1.MP3.TC7SP L1.MP6
ALPS02074854:	L1.MP5 L1.MP2.TC9SP

10.4 mtktsAP 温度采样不准 (1)

敝司当前的thermal policy是基于PCB (即:mtktsAP)的温度来做调整的。
因此，mtktsAP采集的温度准确与否，直接关系到thermal policy是否可以正常工作。

下图是mtktsAP采集温度值的NTC电阻的原理图部分：



有如下几点需要**特别留意**的：
1. 软件上default是使用AUX_IN0
2. 上拉电阻和NTC电阻要**同数量级**的

NTC电阻是100K，则上拉电阻要用390K;
NTC电阻是10K，则上拉电阻要用39K

10.5 mtktsAP 温度采样不准 (2)

excel中NTC的table需要根据HW的设计情况来填写。

NTC	mtktsAP	ENABLE
Extra	Param	PUP_R
Extra	Value	390000
Extra	Param	PUP_VOLT
Extra	Value	1800
Extra	Param	OVER_CRITICAL_L
Extra	Value	4251000
Extra	Param	NTC_TABLE
Extra	Value	6
Extra	AP ADC Channel	0

上拉电阻，单位:欧姆

上拉电压，单位:毫伏

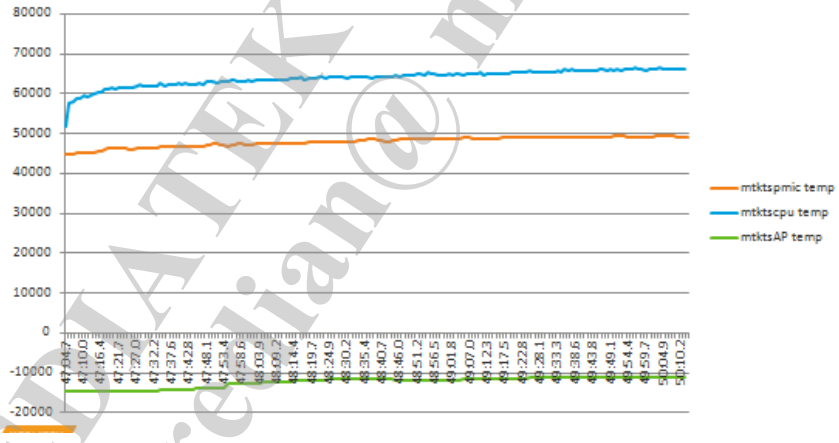
NTC电阻.
6表示100K; 4表示10K

ADC channel.

10.6 mtktsAP 温度采样不准 (3)

如果通过thermal log绘制的温升曲线中，
发现mtktsAP温度是负值，或者明显比mtktspmic低很多(超过20度)
则说明NTC table中设置的小了。

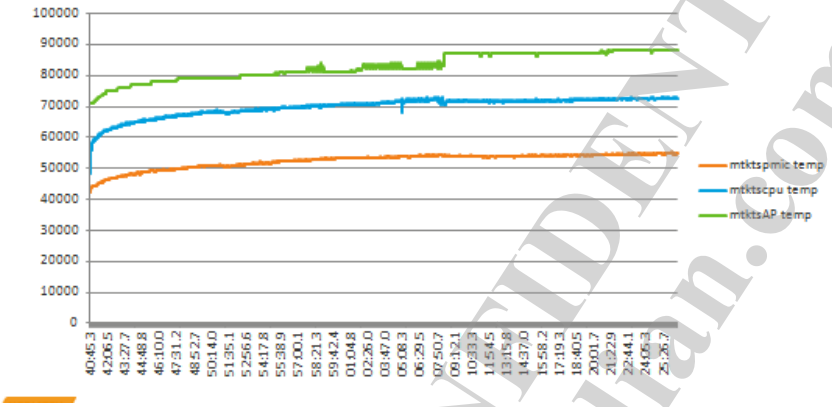
即：NTC table中设置的是4，而实际中贴的NTC电阻是100K的。



10.7 mtktsAP 温度采样不准 (4)

如果通过thermal log绘制的温升曲线中，发现mtktsAP温度明显高于mtktscpu的问题
则说明NTC table中设置的大了。

即：NTC table中设置的是6，而实际中贴的NTC电阻是10K的。



10.8 thermal 引发 kernel panic (1)

在thermal cooler device中有这样的一组cooler:

Cooler name	Binded in Thermal zone	Trip temp
mtktscpu-sysrst	mtktscpu	117000
mtktspmic-sysrst	mtktspmic	145000
mtktspa-sysrst	mtktsAP	95000
mtktsbattery-sysrst	mtktsbattery	60000
mtktswmt	mtktswmt	120000

这组cooler被active的时候，
会通过触发一个data abort的KE，来引发系统重启

通常，该问题发生时，在aee_exp folder下面会有KE的db生成：

10.9 thermal 引发 kernel panic (2)

使用QAAT解析KE的db后，打开__exp_main.txt文件。
按照如下的flow进行分析：

```
Exception Class: Kernel (KE)
PC is at [<ffffffc000637ef8>] tsbat_sysrst_set_cur_state+0x64/0x6c

Current Executing Process:
[thermal_manager, 320]

Backtrace:
[<ffffffc000991694>] __do_kernel_fault.part.5+0x70/0x84
[<ffffffc000094708>] do_page_fault+0x218/0x364
[<ffffffc000094940>] do_translation_fault+0x40/0x4c
[<ffffffc000081380>] do_mem_abort+0x38/0x9c
[<ffffffc000083c58>] el1_da+0x1c/0x88
[<ffffffc000627610>] mtk_cooling_wrapper_set_cur_state+0x124/0x1d4
[<ffffffc000736e64>] thermal_cdev_update+0x88/0xa0
[<ffffffc000737888>] backward_compatible_throttle+0x8c/0xc4
[<ffffffc000736b64>] handle_thermal_trip+0x4c/0x154
```

(2) 再在Exception Class下的PC打印出现的信息来确认是哪个cooler device。
这份log中可以看到是battery这个system reset cooler device被触发

(1) 在backtrace中找到"mtk_cooling_wrapper_set_cur_state"。
这个表面是thermal的system reset cooler打出的KE

(3) 在去对照上一页的mapping table来判断是哪个thermal zone的温度过高了。
这份log是mtktsbattery这个thermal zone的温度超过了60度。

10.10 thermal 引发 kernel panic (3)

再举一个例子：

```
Exception Class: Kernel (KE)
PC is at [<ffffffc000614574>] tspa_sysrst_set_cur_state+0x64/0x6c

Current Executing Process:
[kworker/0:2, 24421][kthreadd, 2]

Backtrace:
[<ffffffc0009a569c>] __do_kernel_fault.part.5+0x70/0x84
[<ffffffc000095248>] do_page_fault+0x218/0x364
[<ffffffc000095480>] do_translation_fault+0x40/0x4c
[<ffffffc0000813f8>] do_mem_abort+0x38/0x9c
[<ffffffc000083c58>] el1_da+0x1c/0x88
[<ffffffc0006038e4>] mtk_cooling_wrapper_set_cur_state+0x138/0x434
[<ffffffc00072b7c4>] thermal_cdev_update+0x88/0xa0
[<ffffffc00072c4e0>] backward_compatible_throttle+0x8c/0xc4
[<ffffffc00072a54c>] handle_thermal_trip+0x4c/0x154
```

这份log中，可以知道是mtktsAP的温度超过95度引起的。

10.11 thermal 引发 kernel panic (4)

在知道了是哪一个thermal zone的温度过高引起的之后，
就需要找办法将温度降下来。

通常有两类方法：

1. 软件上通过调整thermal policy来降低热的产生
（具体请参考Thermal Policy Tuning Guide V1.0这份文档来进行调整）

2. 添加导热辅材或更改机构设计来将热更快的散出去
（通常在SW上已经将thermal policy调整很低的情况下，仍然无改善的时候使用）