

MEDIATEK

MT6765_62 ETT test & stress test reference for LP4X V1.6



Agenda

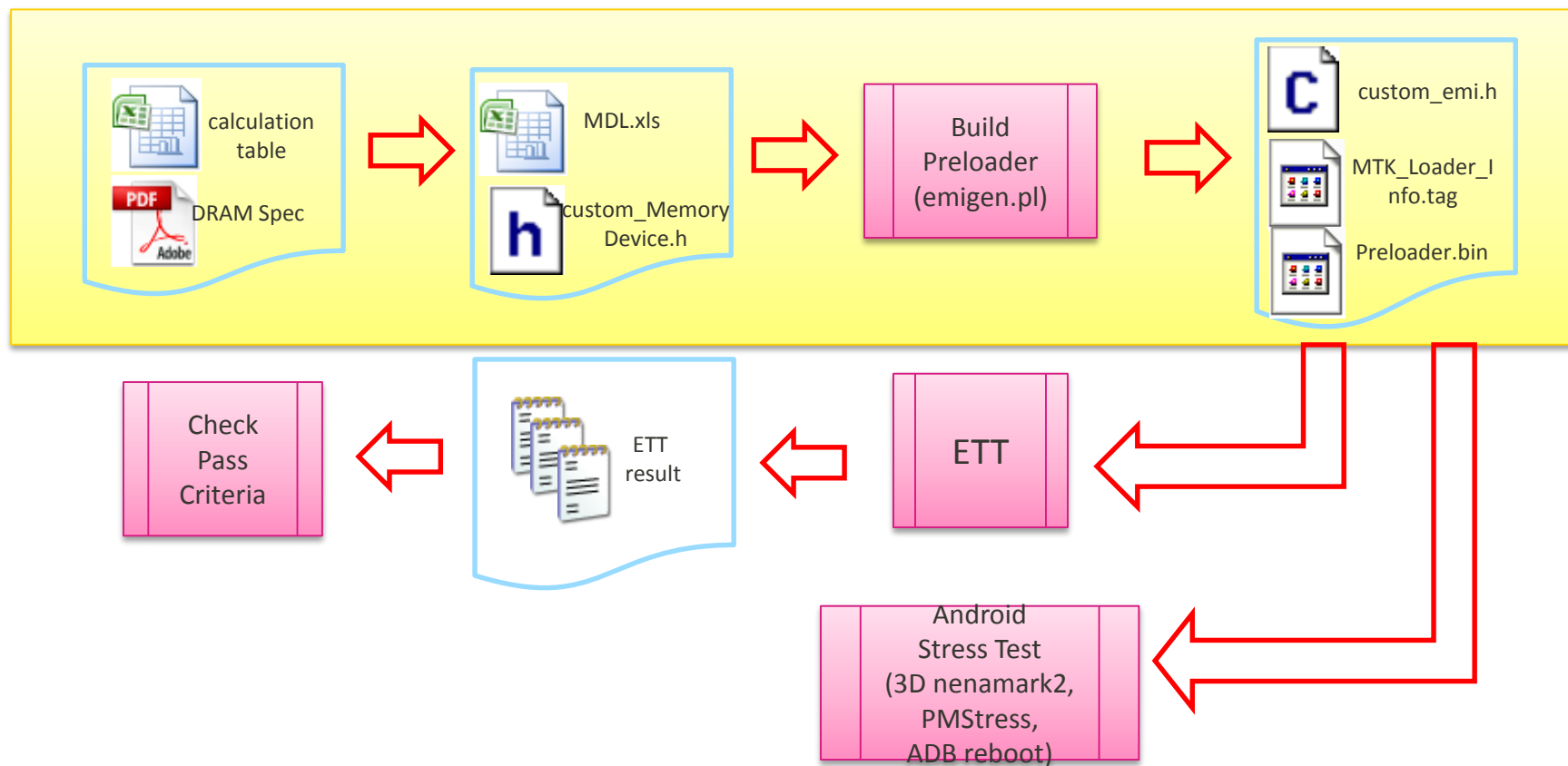
- ETT test step by step
- MTK Eye-Scan Function
- DRAM Stress Test step by step
- Nenamark2 + DVFS for Fast-K
- Suspend/Resume
- Reboot(DDR Reserve mode,Full-K,Fast-k)
- 判断测试pass的方法
- 常见问题处理
- 高温关机问题处理

Chang list

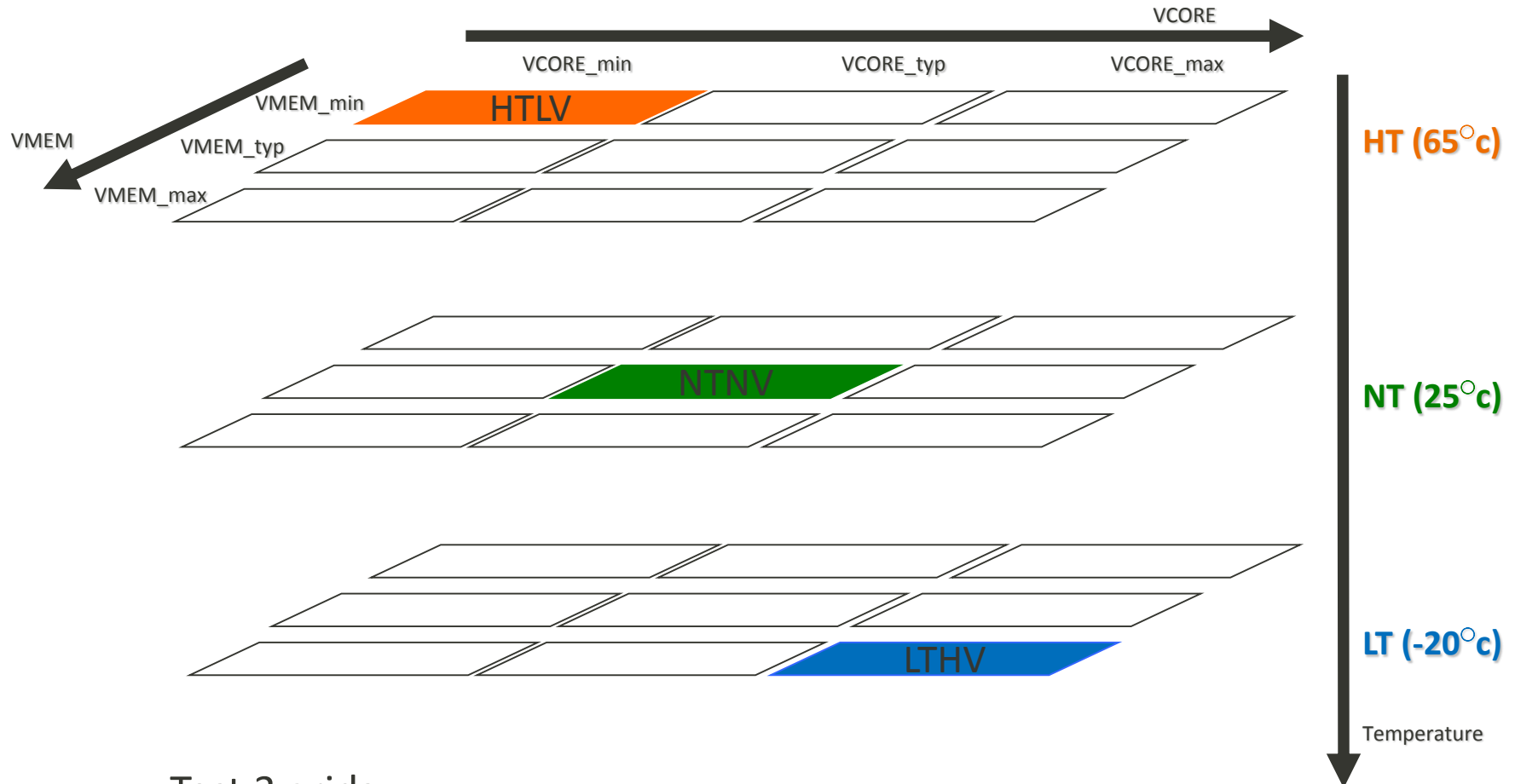
Vison	changlist	date
1	Original version	2018年4月11日
1.1	判断pass路径修改	2018年4月20日
1.2	修改LP4X ETT判断标准	2018年7月9号
1.3	Fix script device id	2018/09/12
1.4	Page 49 添加note提示Log检查	2018/12/10
1.5	Page42 修改note描述	2018/12/20
1.6	Page7、Page26 change VM(LV) from 1.09 to 1.06	2018/12/25

ETT TEST STEP BY STEP

MT6765_62 DRAM Validation Flow



ETT Test Environment Setup (1/2)



- Test 3 grids
 - HTLV, NTNV, LTHV

ETT Test Environment Setup (2/2)

For LPDDR4X 3200MHz

Condition	Temperature (°C)	Vcore (V)	Vm (V)	Vddq(V)
HTLV	65	0.75625	1.06	0.58
NTNV	25	0.80	1.12	0.6
LTHV	-20	0.84375	1.17	0.65

ETT 测试(1/12)

- 工具:支持MT6765_62平台的Flash tool(W1748以后版本).
 - 下载过程序的板子请务必先**Format whole flash**;
 - ETT BIN说明:
https://online.mediatek.com/qvl/_layouts/15/mol/qvl/ext/QVLHomeExternal.aspx下载对应平台、对应memory的ETT bin。
-
- **Note1. 必须用电源给VBAT供电，高低温环境下测试不能使用电池！**
 - **Note2. 手机上的NTC需要下拉10K电阻到GND，模拟电池本身的NTC！**

ETT 测试(2/12)

- **Step1.** UART Cable连接PC和手机的uart0;
- **Step2.** 打开 "超级中端"



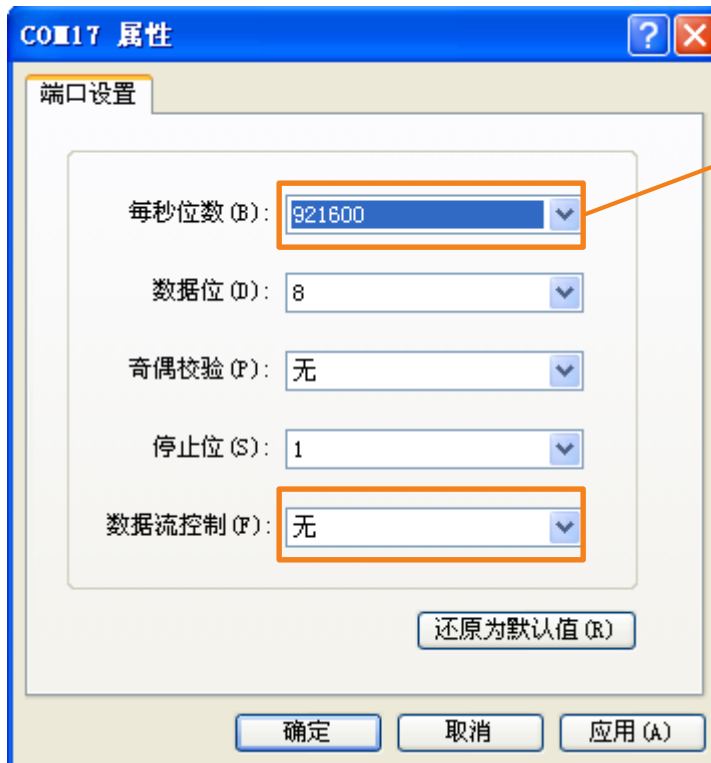
ETT 测试(3/12)

- **Step3:**正确选择UART0对应的COM port（MT6765_62平台的ETT log从uart0打出），点ok按钮；



ETT 测试(4/12)

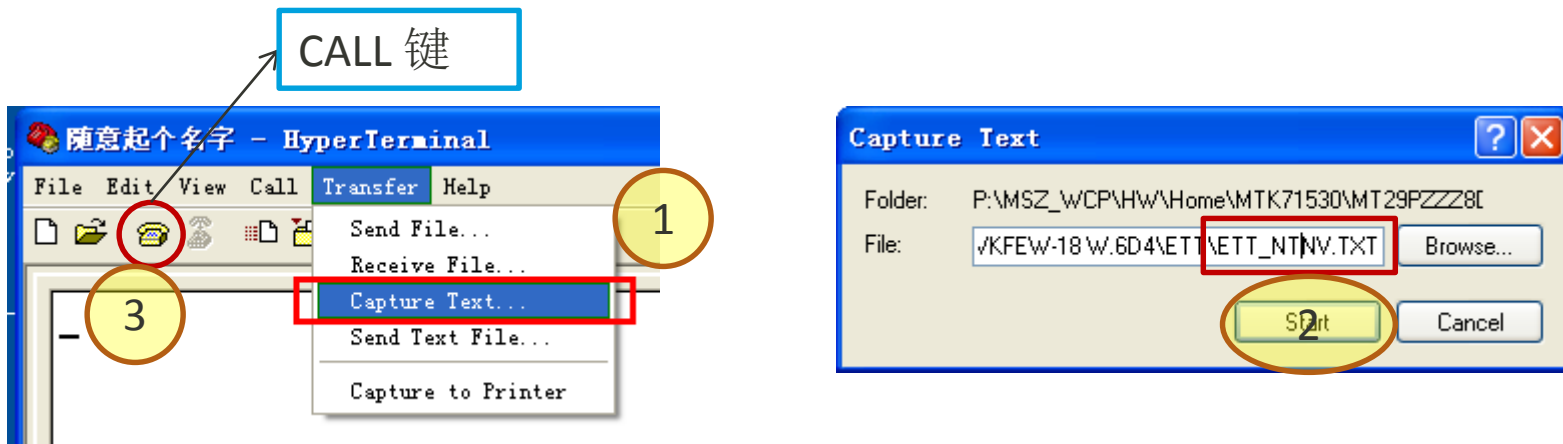
- Step4:波特率设置为” 921600”,数据流为 “无” → 点击确定按钮;



注意!

ETT 测试(5/12)

- **Step5:**建立接收ETT raw data的文件→按下Start→按下CALL键.



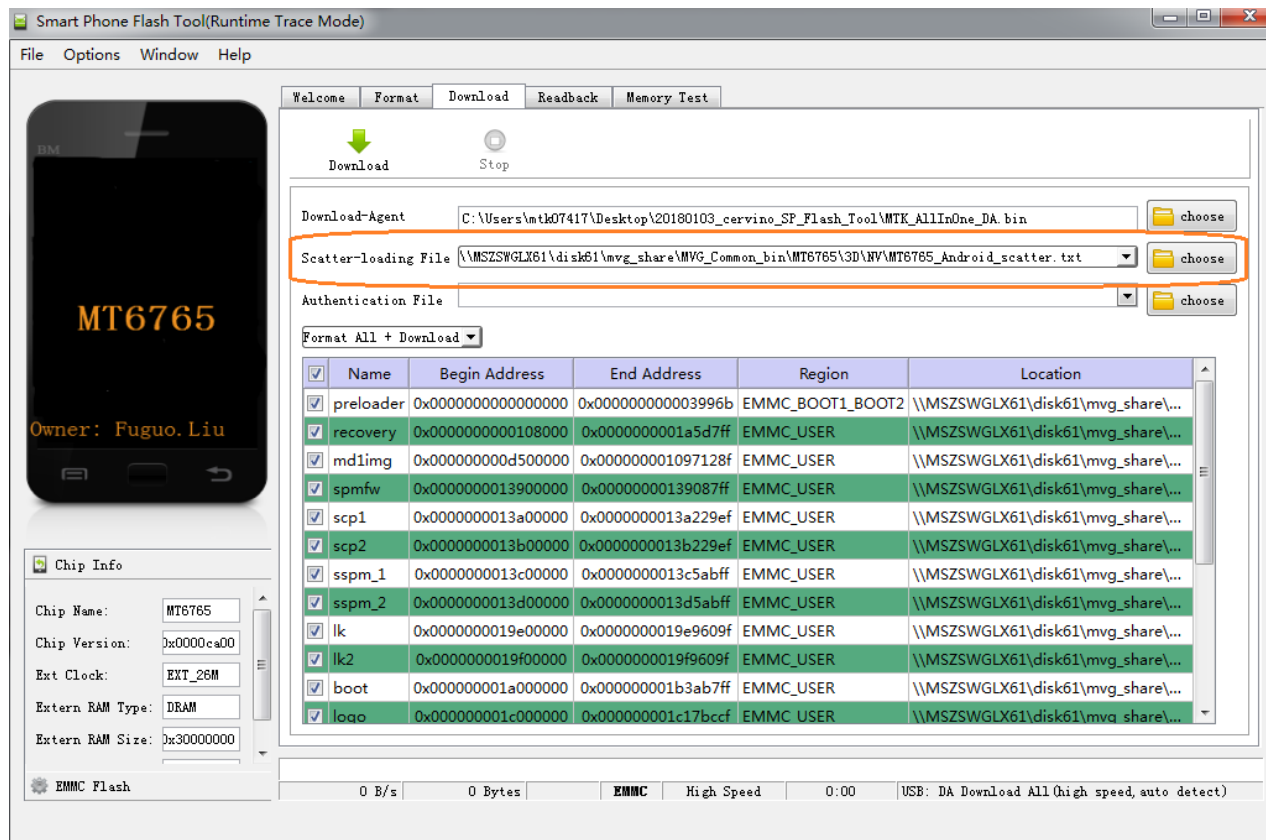
Note1:一种测试条件对应一个单独的文件

Note2:如果中途接收数据异常或中断,需要重新建立一个新的接收文件重做一遍,不要在旧文件上继续做.

注:一定要完整的ETT LOG

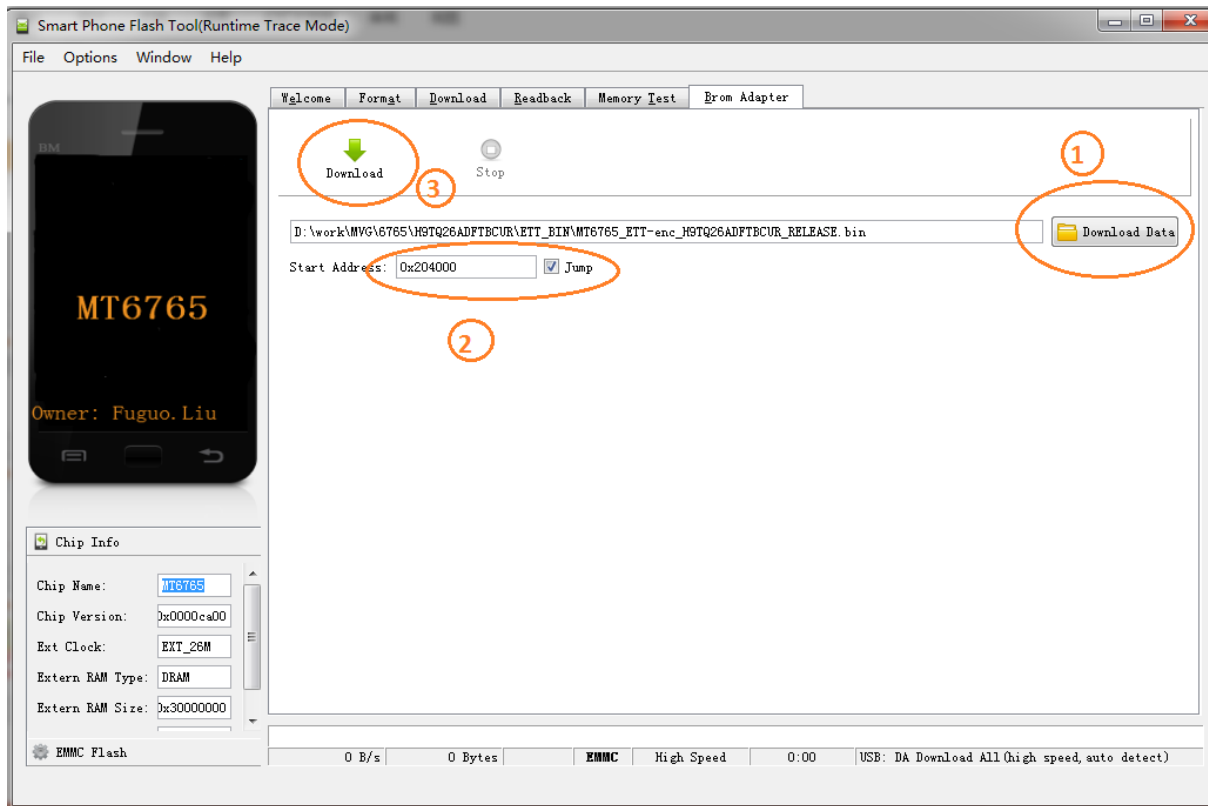
ETT 测试(6/12)

- **Step6:**打开Flash tool，在download 页面加载对应平台的scatter file;
- 下载过程序的板子请务必先**Format whole flash**



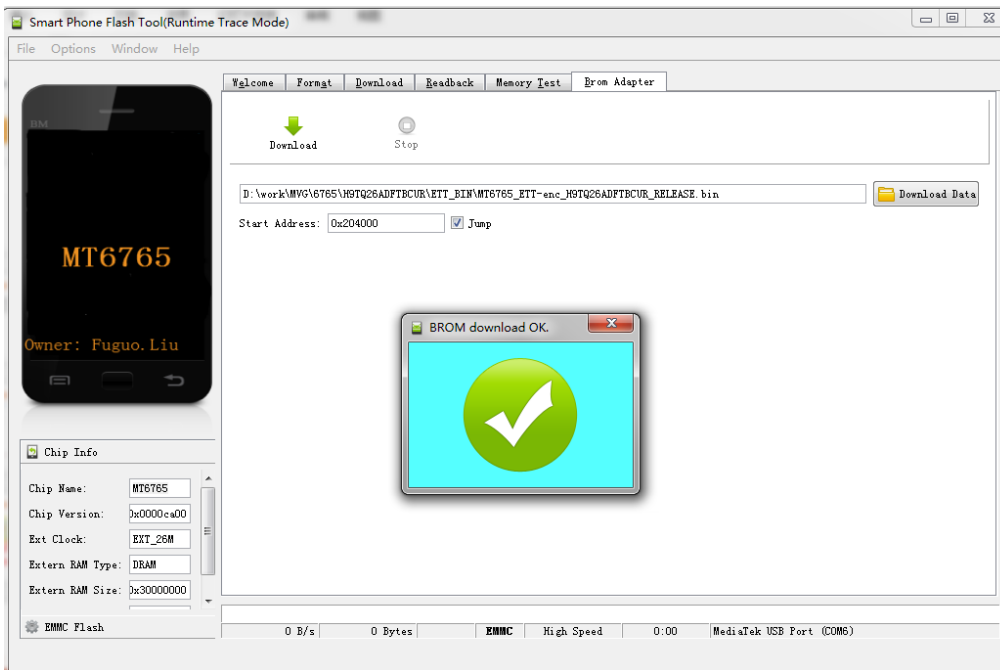
ETT 测试(7/12)

- **Step7:**组合键Ctrl+Alt+A调出flash tool的brom Adapter→选择ETT bin→设置start address(**0x204000**)→勾上Jump→点击download;



ETT 测试(8/12)

- **Step8:**连接USB完成ETT bin下载→转到超级终端调电压；



```
[MT6765] Welcome to ETT's world ... LPDDR3
PMIC TRAP GET DDR TYPE: 0x3
Vcore = 800000
Vdram = 1220000

G : Start the ETT test.
P : Print voltage settings
V : Voltage adjustment.
Please enter selection: v
Vdram (HV, NV, LV)=(1.17, 1.1, 1.06)
Vddq (HV, NV, LV)=(0.65, 0.6, 0.57)
1 : (Vcore HV, Vdram HV, Vddq HV)
2 : (Vcore NV, Vdram NV, Vddq NV)
3 : (Vcore LV, Vdram LV, Vddq LV)
4 : (Vcore HV, Vdram LV, Vddq LV)
5 : (Vcore LV, Vdram HV, Vddq HV)
6 : (Vcore) ++ ...
7 : (Vcore) -- ...
8 : (Vdram) ++ ...
9 : (Vdram) -- ...
10: (Vddq) ++ ...
11: (Vddq) -- ...
12: Max Vdram for heavy load test
```

ETT 测试(9/12)

- Step9. 电压粗调(HV, NV, LV), 此时测量的电压值不准确, 需要在Step11用万用表测试电压

```
G : Start the ETT test.
P : Print voltage settings
V : Voltage adjustment.
Please enter selection: v.
Vdram (HV, NV, LV)=(1.17, 1.1, 1.06)
Vddq (HV, NV, LV)=(0.65, 0.6, 0.57)
1 : (Vcore HV, Vdram HV, Vddq HV)
2 : (Vcore NV, Vdram NV, Vddq NV)
3 : (Vcore LV, Vdram LV, Vddq LV)
4 : (Vcore HV, Vdram LV, Vddq LV)
5 : (Vcore LV, Vdram HV, Vddq HV)
6 : (Vcore) ++ ...
7 : (Vcore) -- ...
8 : (Vdram) ++ ...
9 : (Vdram) -- ...
10: (Vddq) ++ ...
11: (Vddq) -- ...
12: Max Vdram for heavy load test
13: Min Vdram for heavy load test
Please enter pattern selection: 2 No VDDQ for LP3, return
```


ETT 测试(10/12)

- **Step10:**在超级终端主菜单选G开始ETT 测试;

```
[PMIC] pmic_voltage_read:
[HQA] Vcore = NV + 0
[HQA] Vdram = NV + 0

G : Start the ETT test.
P : Print voltage settings
V : Voltage adjustment.
Please enter selection gStart the test...
ett_before_k_test();
ret = Init_DRAM((emi_set->type & 0xF), emi_se
[HQA] hqa_set_voltage_by_freq() for 600
[HQA] Set Vcore = 675000 + 0
[HQA] Set Vdram = 1220000 + 0
```

ETT 测试(11/12)

- **Step11:**ETT 测试时间不长，完成之后 ETT log就不会再有log记录，此时不要下电请用万用表测试对应环境温度的电压，确认Vcore,Vmem, Vddq电压在20mV范围以内即可。

For LPDDR4X 3200MHz

Condition	Temperature (°C)	Vcore (V)	Vm (V)	Vddq(V)
HTLV	65	0.75625	1.09	0.58
NTNV	25	0.80	1.12	0.6
LTHV	-20	0.84375	1.17	0.65

ETT 测试(12/12)

Step12: 判读ETT 是否pass的标准: 能跑到3200且没有fail只有pass则ETT测试通过

```
[HQA] information for measurement,      Dram Data rate = 3200
[Read Voltage]
[HQALOG] 3200 Vcore_HQA = 756250
[HQALOG] 3200 Vdram_HQA = 1060000
[HQALOG] 3200 Vddq_HQA = 570000

[Cmd Bus Training window]
VrefCA Range : 1
VrefCA
[HQALOG] 3200 VrefCA Channel0 Rank0 32
[HQALOG] 3200 VrefCA Channel0 Rank1 32
[HQALOG] 3200 VrefCA Channel1 Rank0 30
[HQALOG] 3200 VrefCA Channel1 Rank1 32
CA_Window
[HQALOG] 3200 CA_Window Channel0 Rank0 46 (bit 3)
[HQALOG] 3200 CA_Window Channel0 Rank1 47 (bit 2)
[HQALOG] 3200 CA_Window Channel1 Rank0 49 (bit 4)
[HQALOG] 3200 CA_Window Channel1 Rank1 50 (bit 3)
CA Min Window(%)
[HQALOG] 3200 CA_Window(%) Channel0 Rank0 72% (PASS)
[HQALOG] 3200 CA_Window(%) Channel0 Rank1 74% (PASS)
[HQALOG] 3200 CA_Window(%) Channel1 Rank0 77% (PASS)
[HQALOG] 3200 CA_Window(%) Channel1 Rank1 79% (PASS)
```

从ETT log中直接查看ETT 结果是否pass。
请做NTNV, HTLV, LTHV三份ETT测试, 直接从log中直接查看结果。
不需要使用parsetool产生ETT out查看。

MTK EYE-SCAN FUNCTION

ETT结果显示眼图功能(1/2)

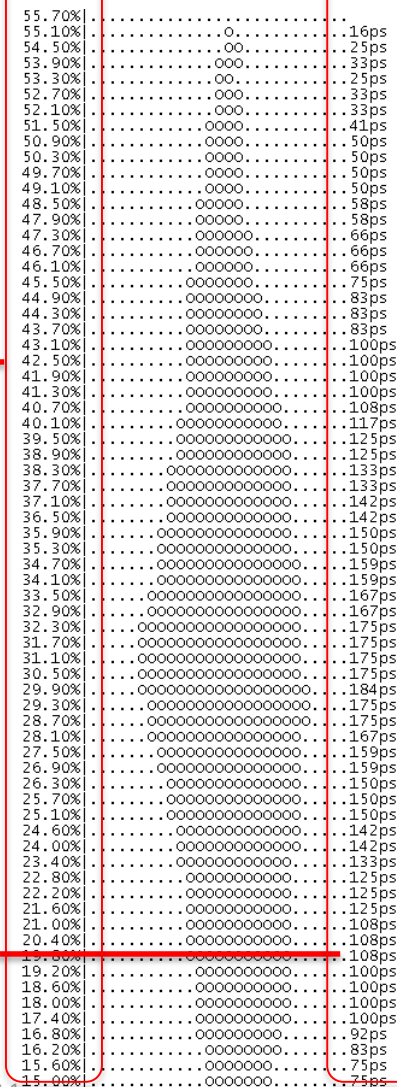
MT6765_62在ETT测试LP4X的时候新增加眼图扫描功能:

- 请注意, 此功能只适用于LPDDR4X的memory, 不能用于LPDDR3
- 眼图扫描功能是用图形表示随着Vref的变化每一个DQ 位上可以顺利pass的数据量显示成的窗口,

如右图由0组成的图形如果横着看像张开的人眼, 简称眼图

对眼图功能还有疑问请看下一页

MTK Eye-Scan for each bit



VREF Setting

PASS timing window

ETT结果显示眼图功能(2/2)

FAQ

■ 眼图是否有判断标准

→目前眼图没有判断标准，它的存在只是给客户多一个参考资料和debug工具，在怀疑板子有问题疑似和dram SI有关，可以在该条件下执行Eye Scan，可以多跑几次(建议跑5次以上)确认讯号的完整性。Debug过程中建议的判断方式是：1.通过Eye 看是否有破洞，垂直或横向是否有缺 2.用不同板子做对比，确认单一bit/byte的高度宽度和其他板子的不同

■ 10.眼图的原理是什么

→眼图原理就是不断的调整Vref，在每一档位的Vref设定值下面去扫每一根DQ的pass window，Vref是JEDEC里面有定义的 JEDEC Vref Setting table for LP4X

Table 2 Vref Settings for Range[0] and Range[1]

Step: 0.6%
(1/167)

Function	Operand	Range[0] Values (% of V_{DDQ})	Range[1] Values (% of V_{DDQ})	Notes
VREF Settings for MR12	OP [5:0]	000000 _B : 15.0% 011010 _B : 30.5%	000000 _B : 32.9% 011010 _B : 48.5%	1,2,3
		000001 _B : 15.6% 011011 _B : 31.1%	000001 _B : 33.5% 011011 _B : 49.1%	
		000010 _B : 16.2% 011100 _B : 31.7%	000010 _B : 34.1% 011100 _B : 49.7%	
		000011 _B : 16.8% 011101 _B : 32.3%	000011 _B : 34.7% 011101 _B : 50.3%	
		000100 _B : 17.4% 011110 _B : 32.9%	000100 _B : 35.3% 011110 _B : 50.9%	
		000101 _B : 18.0% 011111 _B : 33.5%	000101 _B : 35.9% 011111 _B : 51.5%	
		000110 _B : 18.6% 100000 _B : 34.1%	000110 _B : 36.5% 100000 _B : 52.1%	
		000111 _B : 19.2% 100001 _B : 34.7%	000111 _B : 37.1% 100001 _B : 52.7%	
		001000 _B : 19.8% 100010 _B : 35.3%	001000 _B : 37.7% 100010 _B : 53.3%	
		001001 _B : 20.4% 100011 _B : 35.9%	001001 _B : 38.3% 100011 _B : 53.9%	
		001010 _B : 21.0% 100100 _B : 36.5%	001010 _B : 38.9% 100100 _B : 54.5%	
		001011 _B : 21.6% 100101 _B : 37.1%	001011 _B : 39.5% 100101 _B : 55.1%	
		001100 _B : 22.2% 100110 _B : 37.7%	001100 _B : 40.1% 100110 _B : 55.7%	
		001101 _B : 22.8% 100111 _B : 38.3%	001101 _B : 40.7% 100111 _B : 56.3%	
		001110 _B : 23.4% 101000 _B : 38.9%	001110 _B : 41.3% 101000 _B : 56.9%	
		001111 _B : 24.0% 101001 _B : 39.5%	001111 _B : 41.9% 101001 _B : 57.5%	
		010000 _B : 24.6% 101010 _B : 40.1%	010000 _B : 42.5% 101010 _B : 58.1%	
		010001 _B : 25.1% 101011 _B : 40.7%	010001 _B : 43.1% 101011 _B : 58.7%	
		010010 _B : 25.7% 101100 _B : 41.3%	010010 _B : 43.7% 101100 _B : 59.3%	
		010011 _B : 26.3% 101101 _B : 41.9%	010011 _B : 44.3% 101101 _B : 59.9%	
		010100 _B : 26.9% 101110 _B : 42.5%	010100 _B : 44.9% 101110 _B : 60.5%	
		010101 _B : 27.5% 101111 _B : 43.1%	010101 _B : 45.5% 101111 _B : 61.1%	
		010110 _B : 28.1% 110000 _B : 43.7%	010110 _B : 46.1% 110000 _B : 61.7%	
		010111 _B : 28.7% 110001 _B : 44.3%	010111 _B : 46.7% 110001 _B : 62.3%	
		011000 _B : 29.3% 110010 _B : 44.9%	011000 _B : 47.3% 110010 _B : 62.9%	
		011001 _B : 29.9% All Others: Reserved	011001 _B : 47.9% All Others: Reserved	

Notes:

1. These values may be used for MR12 OP[5:0] to set the $V_{REF}(CA)$ levels in the LPDDR4-SDRAM.
2. The range may be selected in the MR12 register by setting OP[6] appropriately.
3. The MR12 registers represents either FSP[0] or FSP[1]. Two frequency-set-points each for CA and DQ are provided to allow for faster switching between terminated and un-terminated operation, or between different high-frequency setting which may use different terminations values.

MTK Eye-Scan for each bit

VREF Setting ←

PASS timing window ←

55.70%0.....	16ps
54.10%00.....	15ps
53.90%000.....	13ps
53.30%00.....	12ps
52.70%000.....	13ps
52.10%000.....	13ps
51.50%0000.....	12ps
50.90%0000.....	10ps
50.30%0000.....	10ps
49.70%0000.....	10ps
49.10%0000.....	10ps
48.50%0000.....	10ps
47.90%00000.....	8ps
47.30%00000.....	6ps
46.70%00000.....	6ps
46.10%00000.....	6ps
45.50%00000.....	7ps
44.90%00000.....	8ps
44.30%000000.....	8ps
43.70%000000.....	8ps
43.10%000000.....	10ps
42.50%000000.....	10ps
41.90%000000.....	10ps
41.30%000000.....	10ps
40.70%000000.....	10ps
40.10%000000.....	11ps
39.50%000000.....	12ps
38.90%000000.....	12ps
38.30%000000.....	13ps
37.70%000000.....	13ps
37.10%000000.....	14ps
36.50%000000.....	14ps
35.90%000000.....	15ps
35.30%000000.....	15ps
34.70%000000.....	15ps
34.10%000000.....	15ps
33.50%000000.....	16ps
32.90%000000.....	16ps
32.30%000000.....	17ps
31.70%000000.....	17ps
31.10%000000.....	17ps
30.50%000000.....	17ps
29.90%000000.....	18ps
29.30%000000.....	17ps
28.70%000000.....	17ps
28.10%000000.....	16ps
27.50%000000.....	15ps
26.90%000000.....	15ps
26.30%000000.....	15ps
25.70%000000.....	15ps
25.10%000000.....	15ps
24.50%000000.....	14ps
24.00%000000.....	14ps
23.40%000000.....	13ps
22.80%000000.....	12ps
22.20%000000.....	12ps
21.60%000000.....	12ps
21.00%000000.....	10ps
20.40%000000.....	10ps
19.80%000000.....	10ps
18.60%000000.....	10ps
18.00%000000.....	10ps
17.40%000000.....	10ps
16.80%000000.....	9ps
16.20%000000.....	8ps
15.60%000000.....	7ps
15.00%000000.....	7ps

DRAM STRESS TEST STEP BY STEP

MT6765_62 DRAM Stress Test 电压设置

- Stress Test电源不需要外灌，通过调整PMIC的输出来设定电压。
- 需要根据下一页的说明分别修改和编译NTNV，HTLV， LTHV对应的三份preloader bin，测试的时候需要选择对应的preloader bin。
- 只需要修改preloder的code即可
- 不需要修改任何Linux kernel code
- VDD2和VDDQ电压由preloader设定后，在linux kernel不会再去修改
- Vcore电压在Linux kernel会被动态设定和调整(Vcore DVFS)
- 注意：1.做Memory压力测试的版本需要在系统其他模块尽量稳定的版本上进行，以免频繁导致测试中断，影响测试。2. 编译root版本：即build版本时增加MTK_BUILD_ROOT=yes

Note: 请务必使用ENG版本，不能使用userdebug版本和user版本，否则测试脚本无法正常运行。

MT6765_62 DRAM Stress Test SW Configuration Android preloader

- Modify “vendor/mediatek/proprietary/bootable/bootloader/preloader/platform/MT6765/src/drivers/inc/emi.h”

1. Enable “DRAM_HQA” macro

```
//=====
//=====pmic related api for ETT HQA
//=====
-#if __ETT__
+#if 1
#define DRAM_HQA
#endif
```

2. Select voltage condition

- HVCORE_HVDRAM / NVCORE_NVDRAM / LVCORE_LVDRAM
- The 3 macros are exclusive for each other, please enable just one macro for each voltage condition.

```
465
466 #ifdef DRAM_HQA
467
468 // #define HVCORE_HVDRAM
469 #define NVCORE_NVDRAM
470 // #define LVCORE_LVDRAM
471 // #define HVCORE_LVDRAM
472 // #define LVCORE_HVDRAM
473
474 #endif
```

MT6765_62 Stress Test 电压测量（1/3）

- 为了验证DRAM的稳定性，接下来便做Stress test.
- MT6765_62平台3D TEST测试环境标准（与ETT的对应）：stress test前测量Vcore电压需要在开机后测量，并参考以下说明，设定固定的Vcore档位后再用示波器测量。

For LPDDR4X：

LP4X	DRAM Data Rate	HV	NV	LV
Vcore	3200(opp1)	0.84375	0.8	0.75625
	2400(opp3)	0.84375	0.8	0.75625
	2400(opp8)	0.7375	0.7	0.6625
	1534(opp10)	0.7375	0.7	0.6625
	1534(opp15)	0.6875	0.65	0.6125
VDRAM(VDD2)		1.17	1.12	1.06
VDDQ		0.65	0.6	0.58

MT6765_62 Stress Test电压测量（2/3）

特别提醒：为了避免不同 PMIC 的差异引起电压不准，run ETT 的主板和进行 Stress test 的主板必须是同一个。

Note1: Vcore DVFS 指的是测试过程中 Vcore 的跳变范围，由 script 控制。

Note2: 需要输入下一页的命令才能把 Vcore 电压频率固定住，固定住之后再测量 Vcore 电压。

Note3: VDD2 电压直接由 preloader 设定，不受 Vcore 档位的影响，可以直接开机后测量

Note4: 测量电压时，请注意分别 download 对应 HV / NV / LV 的 preloader bin 开机后测试

Note5: 开机后，请先不要执行任何脚本，脚本会以 1ms 频率自动调整 Vcore 电压，压力测试过程中测量 Vcore 电压是测不准的。

1. 需要打开 dvfs，使用命令查看，结果为 0 是关闭，结果为 1 是打开。

adb shell cat /sys/devices/platform/10012000.dvfsrc/helio-dvfsrc/dvfsrc_enable---查看 dvfs

adb shell "echo 1 > /sys/devices/platform/10012000.dvfsrc/helio-dvfsrc/dvfsrc_enable"

-----打开 dvfs

```
C:\Users\mtk07417>adb shell cat /sys/devices/platform/10012000.dvfsrc/helio-dvfsrc/dvfsrc_enable
0

C:\Users\mtk07417>adb shell "echo 1 > /sys/devices/platform/10012000.dvfsrc/helio-dvfsrc/dvfsrc_enable"

C:\Users\mtk07417>adb shell cat /sys/devices/platform/10012000.dvfsrc/helio-dvfsrc/dvfsrc_enable
1
```

MT6765_62 Stress Test电压测量 (3/3)

2. 查看当前vcore的DVFS OPP table, 如下图所示显示NV电压, 有的cat出来OPP13, OPP14是0.675V或0.65V, 以cat出来的值为准, 测量电压只要与cat值一致即可。

cat /sys/devices/platform/10012000.dvsrc/helio-dvsrc/dvsrc_opp_table

```
C:\Users\mtk07417>adb shell
1 > /sys/devices/platform/10012000.dvsrc/helio-dvsrc/dvsrc_enable
s/platform/10012000.dvsrc/helio-dvsrc/dvsrc_enable
sys/devices/platform/10012000.dvsrc/helio-dvsrc/dvsrc_opp_table
[OPP0 ]: 0          uv 0          khz
[OPP1 ]: 800000    uv 3200000    khz
[OPP2 ]: 800000    uv 3200000    khz
[OPP3 ]: 800000    uv 2400000    khz
[OPP4 ]: 800000    uv 3200000    khz
[OPP5 ]: 800000    uv 2400000    khz
[OPP6 ]: 800000    uv 3200000    khz
[OPP7 ]: 800000    uv 2400000    khz
[OPP8 ]: 700000    uv 2400000    khz
[OPP9 ]: 700000    uv 2400000    khz
[OPP10]: 700000    uv 1534000    khz
[OPP11]: 700000    uv 2400000    khz
[OPP12]: 700000    uv 1534000    khz
[OPP13]: 700000    uv 1534000    khz
[OPP14]: 700000    uv 1534000    khz
[OPP15]: 650000    uv 1534000    khz
```

3. 如果需要测试HV的电压, 打开DVFS_Nenamark_Memtester_Script文件夹, 双击里面的set_opp_table_HV.bat (先设置device id)来设置电压。如果需要测试LV的电压, 双击set_opp_table_LV.bat。设置完毕可以看到opp table电压已经变化。如右图所示。

```
[OPP0 ]: 0          uv 0          khz
[OPP1 ]: 843750    uv 3200000    khz
[OPP2 ]: 843750    uv 3200000    khz
[OPP3 ]: 843750    uv 2400000    khz
[OPP4 ]: 843750    uv 3200000    khz
[OPP5 ]: 843750    uv 2400000    khz
[OPP6 ]: 843750    uv 3200000    khz
[OPP7 ]: 843750    uv 2400000    khz
[OPP8 ]: 737500    uv 2400000    khz
[OPP9 ]: 737500    uv 2400000    khz
[OPP10]: 737500    uv 1534000    khz
[OPP11]: 737500    uv 2400000    khz
[OPP12]: 737500    uv 1534000    khz
[OPP13]: 737500    uv 1534000    khz
[OPP14]: 737500    uv 1534000    khz
[OPP15]: 687500    uv 1534000    khz
```

MT6765_62 Stress Test电压测量（3/3）

4. 执行adb命令分别固定5档位，每执行一条命令固定一个档位，测量一次Vcore电压，共执行5次命令测量5次，看和设定的电压差异多大，默认测量到的是Opp0的Vcore电压。后续测量到的电压要覆盖到三种频率（3200,2400,1534），三种电压（0.84375，0.7375,0.6875）即可，例如：频率1866电压0.84375可以测试OPP1也可以测试OPP2

`echo OPP_ID > /sys/devices/platform/10012000.dvfsrc/helio-dvfsrc/dvfsrc_force_vcore_dvfs_opp`

例如：下载HV的preload，设置OPP8为0.7375V，命令如下，此时可以用万用表测量到Vcore电压为0.7375V，VDD2为1.12V

```
evb6765_64_emmc:/ # echo 8 > /sys/devices/platform/10012000.dvfsrc/helio-dvfsrc/>
/sys/devices/platform/10012000.dvfsrc/helio-dvfsrc/dvfsrc_opp_table <
```

5. 每设定一次档位，可用” `cat /sys/devices/platform/10012000.dvfsrc/helio-dvfsrc/dvfsrc_dump`”命令确认设定是否生效。

如下图，命令输出后，可以catch到软件设置的电压和寄存器内的参数值，请比较实际测试到的电压值是否和软件设置的一样

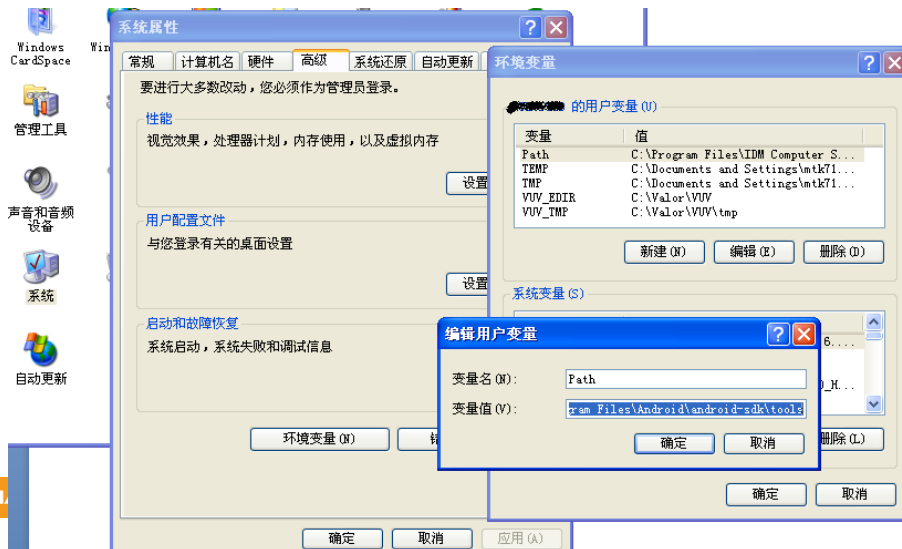
```
sys/devices/platform/10012000.dvfsrc/helio-dvfsrc/dvfsrc_dump
Vcore      : 843750 uv (PMIC: 0x34)
DDR        : 1866000 khz
```

DRAM Stress test测试步骤 (1/8)

Step1 : 搭建环境1

Only need to do this for once on your PC

- Download and install JAVA:
 - <http://www.java.com>
- Install Android SDK to have ADB.
 - <http://developer.android.com/sdk/index.html>
 - Remember to add ADB in your PATH.
 - EX: C:\ path = %path%; YOU_ADB_PATH
 - 获取ADB之后, 在ADB安装位置的tool文件夹下产生一个monkeyrunner.bat的文件, 请把该文件的路径添加到计算机的PATH, EX: ADB装在D:\Program Files, monkeyrunner.bat位于D:\Program Files\Android\android-sdk\tools, 把D:\Program Files\Android\android-sdk\tools添加到PC系统的PATH;
- Install a python environment to be able to run python programs.
 - For example, download Python 2.7.3 Windows Installer (Windows binary -- does not include source) from <http://www.python.org/getit/>



控制版面→系统→高级
→环境变量→Path

在原有的路径后面添加
monkeyrunner.bat的路径, 用
分号与原来的隔开。

DRAM Stress test测试步骤（2/8）

- **Step2. 环境搭建2**

- 选择解压缩**Test_Tools.rar**到根目录
- 解压Test_Tools
- 如无法连接adb, 请先安装SP_Driver_V2.0

Note1. MOL下载MTK_MVG_TOOLS.rar, 包含必要脚本文件;

Note2. Test_Tools和reboot_script, DVFS_Nenamark_Memtester_Script, suspend_and_DVFS script在OA的路径当中不能包含空格和中文。

请特别注意Stess Test测试不需要抓UART log

DRAM Stress test测试步骤（3/8）

- **Step3.**根据测试场景选择对应的load (HTLV/LTHV/NTNV),下载后第一次开机进行如下设置
- 设置→显示→休眠→30分钟,
- 设置→Developer options→stay awake 设置为不休眠
- **Step4.**手机设置为飞行模式, 关闭MTK mobile log

注意: Nenamark的测试有可能上传手机资讯, 测试时请不要连接网络,如有疑虑, 请不要进行此项测试, 脚本会关闭MD和wifi

DRAM Stress test测试步骤(4/8)

■ Step5.设置device id

- 电脑调出cmd命令框，在cmd窗口输入如下命令
- `adb devices`
- 先找出所测试手机的device id，如下图是0123456789ABCDEF 注意没有0x

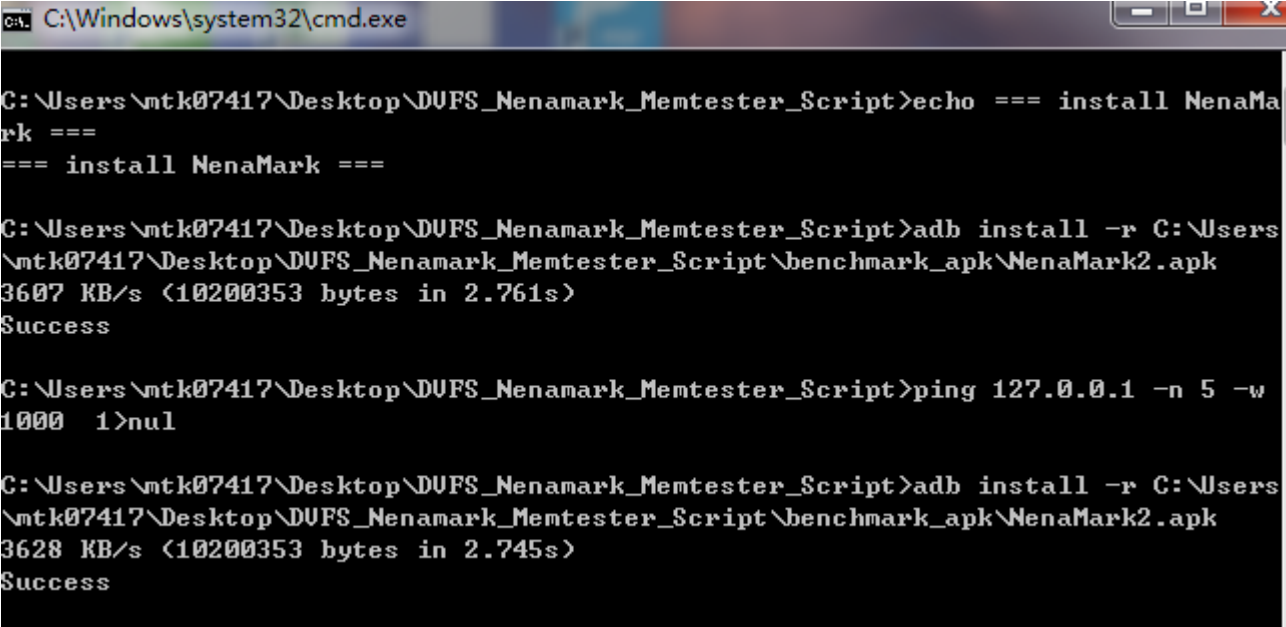
```
C:\Users\ntk07417>adb devices
List of devices attached
0123456789ABCDEF    device
```

- DVFS_Nenamark_memtest_script_LPDDR4文件夹中编辑start_DVFS_N2_Mem_test.bat脚本。把 `set deviceid=-s 0x??` 设置成所测试手机的device id, 如下图所示, 然后保存

```
1_start_DVFS_N2_Mem_test.bat - 记事本
文件(F)  编辑(E)  格式(O)  查看(V)  帮助(H)
@echo off
setlocal EnableDelayedExpansion
set deviceid=-s 0123456789ABCDEF
set memtester=/data/memtester
set chk_status=/data/memtest_check_status.sh
set test_rank=no
set test_num=6
```

DRAM Stress test测试步骤(5/8)

Step6. Install_Nenamark2.bat



```
C:\Windows\system32\cmd.exe

C:\Users\mtk07417\Desktop\DUFS_Nenamark_Memtester_Script>echo === install NenaMark2 ===
=== install NenaMark2 ===

C:\Users\mtk07417\Desktop\DUFS_Nenamark_Memtester_Script>adb install -r C:\Users\mtk07417\Desktop\DUFS_Nenamark_Memtester_Script\benchmark_apk\NenaMark2.apk
3607 KB/s (10200353 bytes in 2.761s)
Success

C:\Users\mtk07417\Desktop\DUFS_Nenamark_Memtester_Script>ping 127.0.0.1 -n 5 -w 1000 1>nul

C:\Users\mtk07417\Desktop\DUFS_Nenamark_Memtester_Script>adb install -r C:\Users\mtk07417\Desktop\DUFS_Nenamark_Memtester_Script\benchmark_apk\NenaMark2.apk
3628 KB/s (10200353 bytes in 2.745s)
Success
```

- 安装完毕会自动结束，可以在手机里面找到一个NenaMark2的图标，双击会出现 **choose what to allow NenaMark2 to access** 的讯息，点击 **continue** 给权限。然后双击Nenamark2的图标，点击 **run** **需要先跑一次3D动画**。

DRAM tress test测试步骤(6/8)

- Step7. 编辑Push.bat脚本，把set deviceid=-s 0x?? 设置成所测试手机的device id，双击运行Push.bat(出现下图并退出)

```
adbd is already running as root
remount succeeded
"=== push run.sh, start, back ==="
6 KB/s (527 bytes in 0.078s)
3 KB/s (202 bytes in 0.062s)
3 KB/s (192 bytes in 0.062s)
"=== install DUPS script ==="
238 KB/s (330976 bytes in 1.357s)
258 KB/s (330952 bytes in 1.248s)
21 KB/s (1036 bytes in 0.046s)
6 KB/s (1025 bytes in 0.156s)
```

- Step8. 双击hqa_test_cmd_fix_gpu.bat

```
C:\Users\ntk10621\Desktop\DUPS_test_script_LP4_desense_20180329
0180329>adb shell "cat /proc/gpufreq/gpufreq_var_dump"
g_cur_opp_idx = 2, g_cur_opp_cond_idx = 2
g_cur_opp_freq = 400000, g_cur_opp_volt = 65000, g_cur_opp_vsram_v
real freq = 400000, real volt = 80000, real vsram_volt = 0
current vcore opp = 3
clock freq = 400052
g_segment_id = 2
g_volt_enable_state = 1
g_opp_stress_test_state = 0
g_DUPS_off_by_ptpod_idx = 2
g_max_limited_idx = 0
g_opp_springboard_idx = 0
gpu_loading = 0
g_limited_idx_array[0] = 0
g_limited_idx_array[1] = 0
g_limited_idx_array[2] = 0
g_limited_idx_array[3] = 0
g_limited_idx_array[4] = 0
```

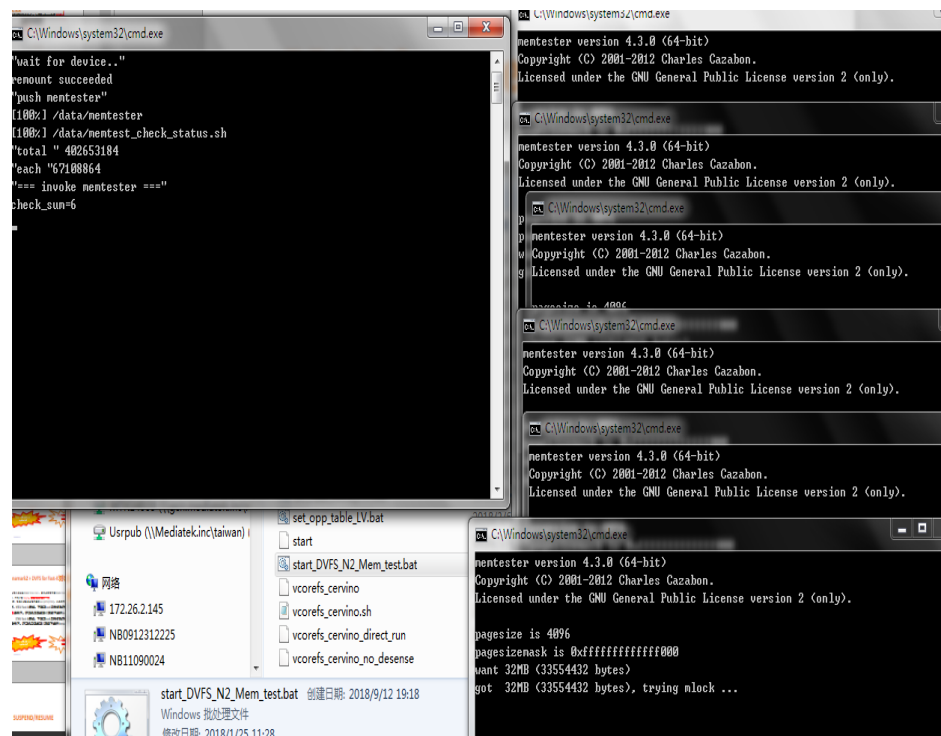
DRAM Stress test测试步骤(7/8)

- Step9. 如果做HV测试先双击set_opp_table_HV.bat进行配置HV OPP table
- 如果做LV测试先双击set_opp_table_LV.bat进行配置LV OPP table
- 如果做NV测试不需要配置OPP table

注意：上述脚本都需要编辑填入手机正确的device id。

- Step10: adb shell
- sh /data/run.sh &
- Step11: adb shell
- sh /data/vcorefs_cervino.sh &
- Step12:

DVFS_Nenamark_memtest_script_LPDDR4文件夹中双击start_DVFS_N2_Mem_test.bat脚本，会弹出n个框图,同时手机屏幕上3D动画在自动跑动,如果3D动画无法自动run请查看文档display the x and y point.pdf进行设置.



DRAM Stress test测试步骤(8/8)

- Step13. 此时千万不要拔掉USB，需要跑8小时Stress test
- Step14. 请参看‘判断测试pass的方法’对测试结果进行分析
- Step15. 重复上述步骤将>3部手机的HTLV\LTHV\NTNV所有条件下的Stress test完成。
- 高温环境测试换用低压preloader bin, 低温环境测试换用高压preloader bin, 常温环境用常温preloader bin.
- 每次变化环境都要重新download.



New: 不需要抓UART
log进行分析



**USB千万不能
拔掉**

Nenamark2 + DVFS for Fast-K测试步骤

注意：此项测试也是DRAM Stress test，操作步骤按前面DRAM Stress Test进行，不同点在 **Step3，首次开机环境不一样**。

新增测试项

Step 3改为：根据测试场景选择对应的load (HTLV/LTHV)，测试共两项。

操作：HTLV fast-k测试，下载完load后的首次开机需要在低温条件下。开完机后再放到HT环境下进行Stress Test。

LTHV fast-k测试，下载完load后的首次开机需要在高温条件下。开完机后再放到LT环境下进行Stress Test。

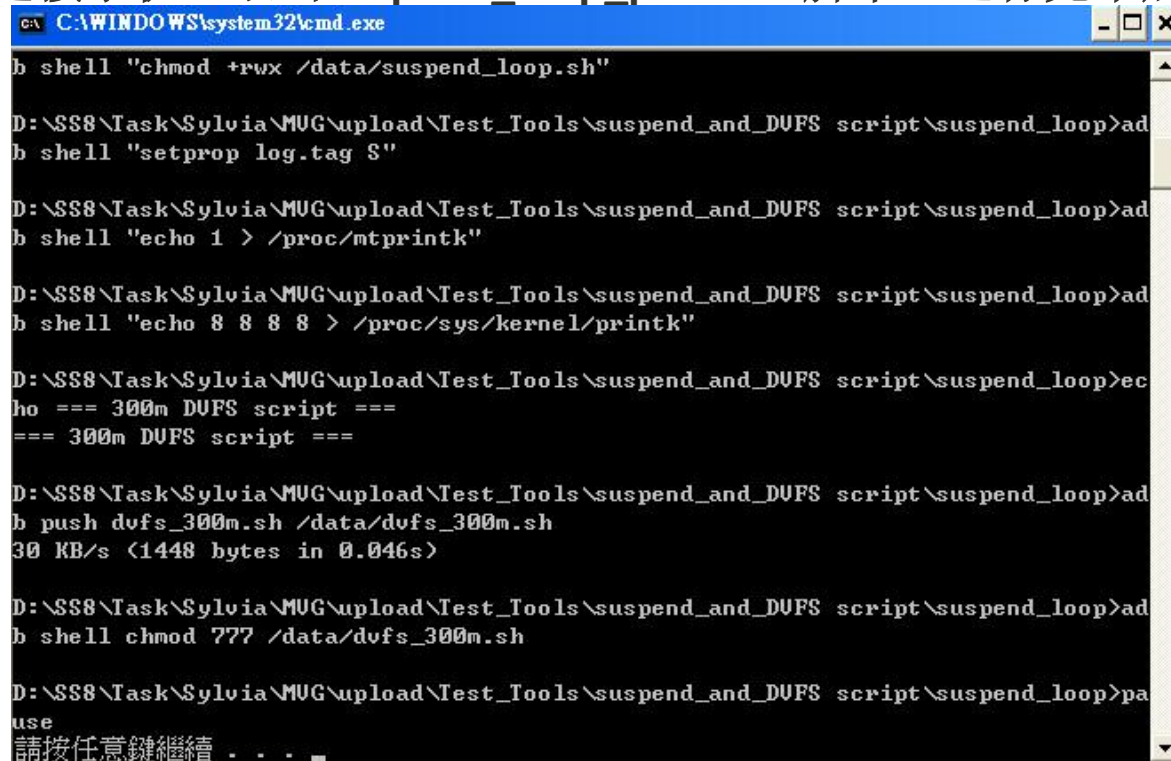
New：不需要抓UART log进行分析

USB千万不能拔掉

SUSPEND/RESUME

Suspend/Resume 测试步骤(1/3)

- **Preloader bin**使用默认的即可.
- 屏幕**不要**设置为always on, 手机设置为飞行模式, 关闭MTK mobile log
- **Step1.** 打开suspend_loop文件夹, 手机开debug功能
- **Step2.**连接手机, 双击suspend_loop_push.bat脚本, 运行完毕后按任意键结束



```
C:\WINDOWS\system32\cmd.exe

b shell "chmod +rwx /data/suspend_loop.sh"

D:\SS8\Task\Sylvia\MUG\upload\Test_Tools\suspend_and_DVFS script\suspend_loop>ad
b shell "setprop log.tag S"

D:\SS8\Task\Sylvia\MUG\upload\Test_Tools\suspend_and_DVFS script\suspend_loop>ad
b shell "echo 1 > /proc/mtdprintk"

D:\SS8\Task\Sylvia\MUG\upload\Test_Tools\suspend_and_DVFS script\suspend_loop>ad
b shell "echo 8 8 8 8 > /proc/sys/kernel/printk"

D:\SS8\Task\Sylvia\MUG\upload\Test_Tools\suspend_and_DVFS script\suspend_loop>ec
ho === 300m DVFS script ===
=== 300m DVFS script ===

D:\SS8\Task\Sylvia\MUG\upload\Test_Tools\suspend_and_DVFS script\suspend_loop>ad
b push dvfs_300m.sh /data/dvfs_300m.sh
30 KB/s <1448 bytes in 0.046s>

D:\SS8\Task\Sylvia\MUG\upload\Test_Tools\suspend_and_DVFS script\suspend_loop>ad
b shell chmod 777 /data/dvfs_300m.sh

D:\SS8\Task\Sylvia\MUG\upload\Test_Tools\suspend_and_DVFS script\suspend_loop>pa
use
請按任意鍵繼續 . . .
```


Suspend/Resume 测试步骤(2/3)

- Step3. 调出一个cmd命令框，输入如下命令

- adb shell
- sh /data/suspend_loop.sh &

```
D:\adb_1_031>adb shell
k71v1_64_bsp:/ # sh /data/suspend_loop.sh &
sh /data/suspend_loop.sh &
[1] 3309
k71v1_64_bsp:/ # ???[???]???_wakeup_???[???]???
```

- Step4. 拔掉usb, 拔掉usb, 拔掉usb, 重要的事情说三次

- 可接power monitor，也可以接usb adapter，不要直接用电脑供电给手机
- 测试时间12个小时

New: 不需要抓UART
log进行分析

USB要拔掉

Suspend/Resume 测试步骤(3/3)

- **Step5.** 测试的时候请确认板子是做了有效休眠唤醒的动作，使用如下命令进行查看

`adb shell cat /sys/kernel/debug/cpuidle/spm/spm_sleep_count`

例如开始测试的时候查看到如下进行了0次测试

```
evb6765_64_emmc:/ # cat /sys/kernel/debug/cpuidle/spm/spm_sleep_count
0
```

拔掉USB后进行测试一段时间再次连上USB，看到如下进行了662次测试，

有效测试标准：1.USB必须拔掉 2.测试过程中测试次数有在不断增加

```
C:\Users\ntk07417>adb shell cat /sys/kernel/debug/cpuidle/spm/spm_sleep_count
662
```

- **Step6.**请参看‘判断测试pass的方法1和3’对测试结果进行分析
- **Step7.**重复上述步骤将>3部手机在**常温**条件下休眠唤醒。



New: 不需要抓UART
log进行分析



USB要拔掉

REBOOT

(DDR RESERVE MODE , FULL-K, FAST K)

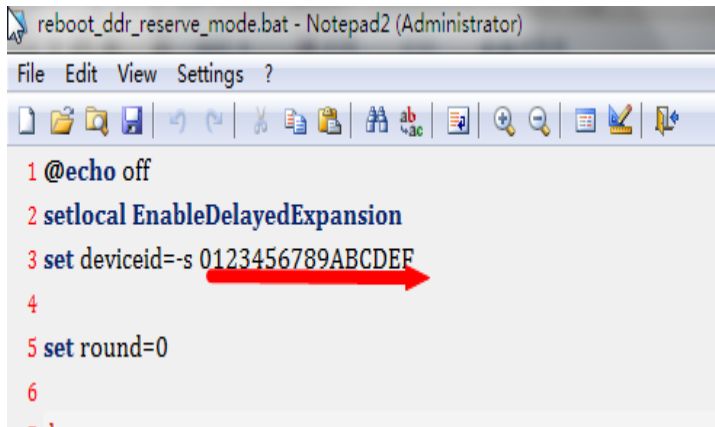
DDR reserve mode Reboot测试步骤(1/2)

- DDR Reserve mode test: 就是在打开DDR reserve mode 功能情况下测试reboot, DDR reserve mode是在reboot前让DRAM 进入self refresh, 确保reboot时DRAM资料可以保留, 在reboot后再把资料捞出来debug用。
- Step1. Preloader bin使用默认的即可.**

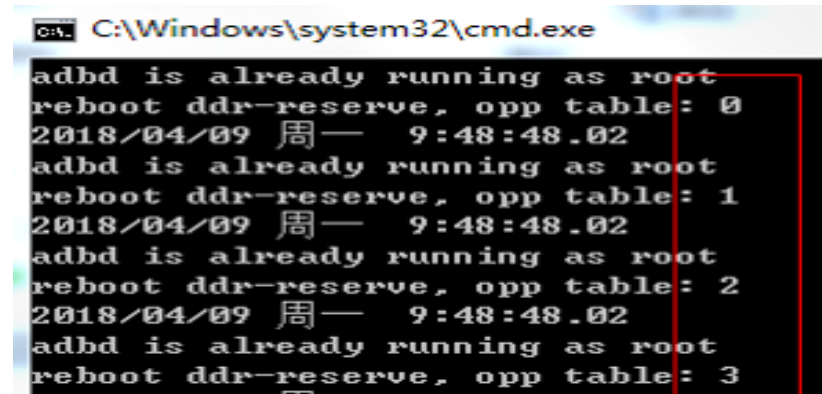
屏幕不要设置为always on, 手机设置为飞行模式, 关闭MTK mobile log

- Step2.**打开reboot_script文件夹, 编辑reboot_ddr_reserve_mode.bat把device id改成测试机器id后保存, 双击该脚本出现下图的框图, 和其他reboot比较区别在于有 "reboot with ddr reserve mode"字样

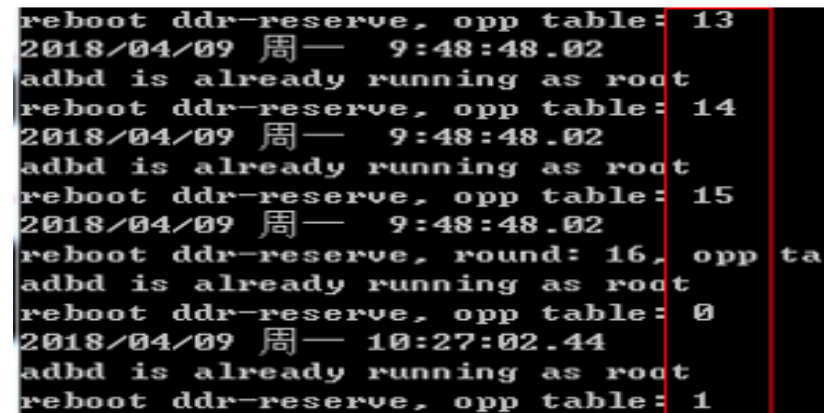
Note:Opp table 每16次循环



```
reboot_ddr_reserve_mode.bat - Notepad2 (Administrator)
File Edit View Settings ?
1 @echo off
2 setlocal EnableDelayedExpansion
3 set deviceid=-s 0123456789ABCDEF
4
5 set round=0
6
7
```



```
C:\Windows\system32\cmd.exe
addb is already running as root
reboot ddr-reserve, opp table: 0
2018/04/09 周一 9:48:48.02
addb is already running as root
reboot ddr-reserve, opp table: 1
2018/04/09 周一 9:48:48.02
addb is already running as root
reboot ddr-reserve, opp table: 2
2018/04/09 周一 9:48:48.02
addb is already running as root
reboot ddr-reserve, opp table: 3
```



```
reboot ddr-reserve, opp table: 13
2018/04/09 周一 9:48:48.02
addb is already running as root
reboot ddr-reserve, opp table: 14
2018/04/09 周一 9:48:48.02
addb is already running as root
reboot ddr-reserve, opp table: 15
2018/04/09 周一 9:48:48.02
reboot ddr-reserve, round: 16, opp ta
addb is already running as root
reboot ddr-reserve, opp table: 0
2018/04/09 周一 10:27:02.44
addb is already running as root
reboot ddr-reserve, opp table: 1
```

DDR reserve mode Reboot测试步骤(2/2)

- **Step4.**需要抓UART log进行分析，需要抓UART log进行分析，需要抓UART log进行分析，重要的事情说三遍，测试时间12小时
- **Step5.**请参看‘判断测试pass的方法1和3’对测试结果进行分析
- **Step6.**请打开UART log，对UART log进行关键字搜索, 如果log太大请用File-Splitter（网上下载）把log分割成小份进行分析
- 测试Fail的关键字搜索“[RGU] WDT DDR reserve mode FAIL!”
- 测试PASS的关键字搜索“DDR RESERVED”会出现如下log表示测试pass

```
[RGU] WDT DDR reserve mode success! 1387F1
[DDR Reserve] DCS/DVFSRC success! (dcs_en=0, dvfsrc_en=1)
[RGU] WDT DDR reserve mode success! 1387F1
[RGU] DDR RESERVE Success 1
[DDR Reserve] release dram from self-refresh PASS!
```
- **Step7.**重复上述步骤将>3部手机在常温条件下reboot完成。



要抓URART LOG



**USB千万不能
拔掉**

Full-K Reboot测试步骤(1/2)

- **Step1. Preloader 使用full k bin.**
- **生成Full k bin方法:**
- 1: 关闭 #define DRAM_HQA
- 2: 在dramc_pi_api.h中修改
- #define SUPPORT_SAVE_TIME_FOR_CALIBRATION **CFG_DRAM_CALIB_OPTIMIZATION**
- 改为 #define SUPPORT_SAVE_TIME_FOR_CALIBRATION **0**

Step 2.屏幕不要设置为always on，手机设置为飞行模式，关闭MTK mobile log

- **Step3.编辑 reboot_full_k.bat把device id**
- **改成测试机器id后保存，双击该脚本。**

- Note1: Full k 每5次更新。
- Note2: 脚本若无device id，则不需修改。

```
DDR : 3200000 khz
reboot full-k, opp table: 1
2018/09/12 周三 19:05:08.24
Ucore : 800000 uv <PMIC: 0x2d>
DDR : 2400000 khz
reboot full-k, opp table: 7
2018/09/12 周三 19:05:08.24
Ucore : 700000 uv <PMIC: 0x1d>
DDR : 2400000 khz
reboot full-k, opp table: 8
2018/09/12 周三 19:05:08.24
Ucore : 700000 uv <PMIC: 0x1d>
DDR : 1534000 khz
reboot full-k, opp table: 14
2018/09/12 周三 19:05:08.24
Ucore : 650000 uv <PMIC: 0x15>
DDR : 1534000 khz
reboot full-k, opp table: 15
2018/09/12 周三 19:05:08.24
reboot full-k, round: 5
Ucore : 800000 uv <PMIC: 0x2d>
DDR : 3200000 khz
reboot full-k, opp table: 1
2018/09/12 周三 19:07:41.00
```

Full-K Reboot测试步骤(2/2)

- **Step4. opp table**显示当前频率档位
- 测试时间**12**个小时
- **Step5.**请参看‘判断测试pass的方法1和3’对测试结果进行分析
- **Step6.**重复上述步骤将**>3**部手机在**常温**条件下reboot完成。



New: 不需要抓UART
log进行分析



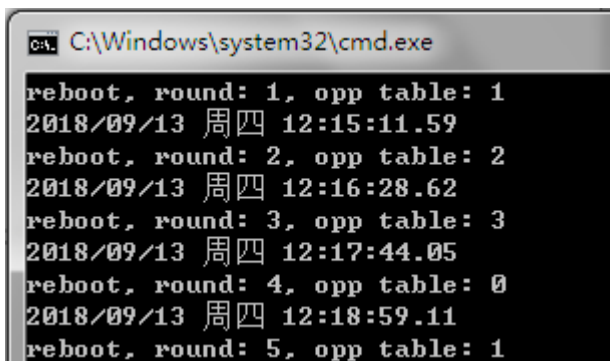
**USB千万不能
拔掉**

Fast-k Reboot测试步骤(1/2)

- 此测试项目只对LPDDR4，LPDDR3不需要做这个项目，Preloader bin使用LV bin在常温下测试
- Fast-k Stress test目的：下载软件后第一次开机的校准数据需要保存下来，为后续测试所用。
- Step1.打开reboot_script文件夹，编辑reboot_fast_K.bat把device id改成测试所用板子id

Fast-k Reboot测试步骤(2/2)

- Step2. 双击reboot_fast_K.bat进行测试，测试需要12小时。



```
C:\Windows\system32\cmd.exe
reboot, round: 1, opp table: 1
2018/09/13 周四 12:15:11.59
reboot, round: 2, opp table: 2
2018/09/13 周四 12:16:28.62
reboot, round: 3, opp table: 3
2018/09/13 周四 12:17:44.05
reboot, round: 4, opp table: 0
2018/09/13 周四 12:18:59.11
reboot, round: 5, opp table: 1
```

- Step3.请参看‘判断测试pass的方法’对测试结果进行分析
- Step4.请打开UART log，对UART log进行关键字搜索，如果log太大请用File-Splitter（网上下载）把log分割成小份进行分析

测试Fail的关键字搜索“Save calibration result to emmc”

测试PASS的关键字搜索“Bypass saving calibration result to emmc”

- Step5. 重复上述步骤将>3部手机在NT条件下reboot完成。

Note: 若上述Fail和PASS关键字均没有搜索到, 请先自行检查code,看是否有修改掉对应log。



要抓UART LOG



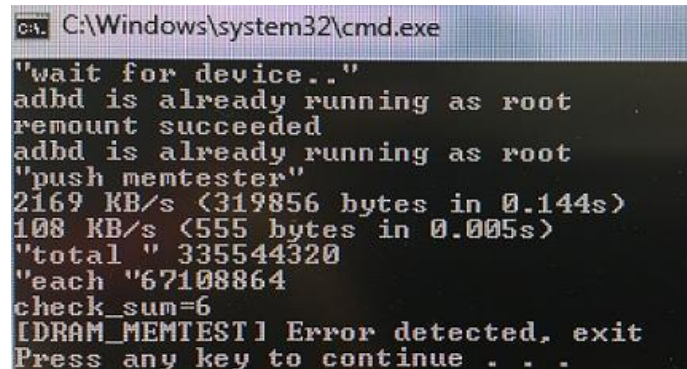
USB千万不能
拔掉

判断测试PASS的方法

判断测试pass的方法(1/3)

1. 连续测试后仍能正常运行：测试N小时后还在连续跑测试；
2. 如果有跑memtester的项目：连续测试后memtester还是在跑，如果memtester停止下来就是fail。

memtester的窗口停掉了会出现Error detected，如下图所示



```
C:\Windows\system32\cmd.exe
"wait for device.."
adbd is already running as root
remount succeeded
adbd is already running as root
"push memtester"
2169 KB/s <319856 bytes in 0.144s>
108 KB/s <555 bytes in 0.005s>
"total " 335544320
"each "67108864
check_sum=6
[DRAM_MEMTEST] Error detected, exit
Press any key to continue . . .
```

请运行collect_memtest_log.bat，会出现一个log文件夹把memtester fail log直接抓出来，打包上传给MTK分析即可

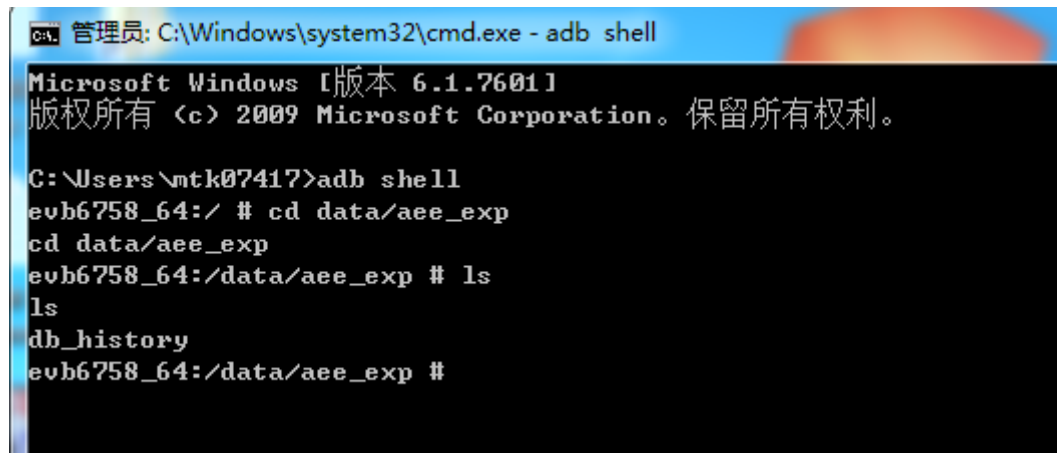
判断测试pass的方法(2/3)

3.在下面两个路径下检查是否有这4个字样：KE,HWT， HW_reboot, MEMTEST,
如果下面两个路径中有任何一个字样表示fail,没有那3个字样表示pass

(1) /data/aee_exp (2) /data/vendor/mtklog/aee_exp

上述3个条件任一不满足即可判定为 fail, 这时请将log发送给MTK分析。

例子1：如右图，打开cmd
adb shell
cd data/aee_exp
ls
可以看到aee_exp文件夹里
面并没有那4个字样



```
管理员: C:\Windows\system32\cmd.exe - adb shell
Microsoft Windows [版本 6.1.7601]
版权所有 (c) 2009 Microsoft Corporation。保留所有权利。

C:\Users\mtk07417>adb shell
evb6758_64:/ # cd data/aee_exp
cd data/aee_exp
evb6758_64:/data/aee_exp # ls
ls
db_history
evb6758_64:/data/aee_exp #
```

判断测试pass的方法(3/3)

例子2：如下图/data/vendor/mtklog/aee_exp文件夹里面有KE/HWT/HW_Reboot/的字样，表示发生了异常产生了DB需要把db文件用pull_log.bat pull出来发给MTK分析。

```
C:\Users\mtk10621>adb shell
evb6765_64_emmc:/ # cd /data/vendor/mtklog/aee_exp
evb6765_64_emmc:/data/vendor/mtklog/aee_exp # ls -al
total 16
drwxr-xr-x 3 root root 4096 2010-01-02 20:57 .
drwxr-xr-x 3 root root 4096 2018-04-11 05:36 ..
-rw----- 1 root root    0 2010-01-01 04:04 aee.lck
drwxrwxrwx 2 root root 4096 2010-01-01 04:04 db.00.KernelAPI
-rw-r--r-- 1 root root 103 2010-01-01 04:04 db_history
evb6765_64_emmc:/data/vendor/mtklog/aee_exp #
```

如何把aee_exp文件夹pull出来发给MTK

- 1.首先在D盘建立1个名字为mtklog的文件夹
- 2.使用把脚本pull_log.bat放到mtklog的文件夹里面，手机连上usb，双击一下脚本pull_log.bat即可，等所有log导入到文件夹完毕，打包mtklog上传给MTK进行分析即可

常见问题处理

常见问题处理

- ETT BIN无法下载
→ 确认有做Format whole flash操作；是否有勾上Jump, Address填写正确。尝试按下Download key.
- 超级终端无输出
→ 确认UART cable电平是否为1.8V, 手机正常开机是否能抓到log. 排除UART cable问题及线路连接问题。
- 超级终端无法敲入命令
→ 确认UART线的RX, TX和GND都有连接。
- ETT跑到一半中断
→ 确认手机是否有正确连接电源。
- adb无法连接
→ 确认ADB环境及驱动是否安装成功。
- APK无法安装
→ 脚本文件路径不能有中文和空格，APK命名是否正确。
- eMMC R/W测试无法运行
→ 确认软件版本32bit还是64bit, 与脚本是否匹配
- 手机进入待机模式，按Power KEY可唤醒
→ 请检查是否有设置显示->休眠->30分钟

高温关机问题处理

问题： NTNV/LTHV时3D测试正常，但是在HTLV时会出现自动关机的问题，且在Log中能找到Thermal 关机的线索。

原因：

由于客户手机在开发阶段还没有做散热处理，如没有散热胶，或PA 的Thermal ADC没有校准，在65度的高温下做压力测试，PCB的温度很快上升到100度，就会触发Thermal关机；也有可能是用了带NTC的电池做测试。

解决方法：

1. 用假电池，并把NTC电阻换成10K。
2. 给手机加散热胶或其它散热装置
3. 把全部的Thermal Throttling的功能关掉（下页的方法1或2）
4. 只关闭LTE 的Thermal Throttling机制（下页的方法3）

当发现有死机现象时，请打开Mobile Log 复制问题，并在Log中查看是否因为LTE Throttling 导致关机，如果只是LTE throttling导致关机，仅需要把LTE的Throttling 关掉，AP的Throttling可以保留(即采用下面的方法3)。

How To Modify Thermal Policy

- 关闭thermal throttling方法如下，目前的平台都会适用

(1) User sw load关闭thermal throttling

(a) Get root permission

(b) adb shell "/system/bin/thermal_manager /etc/.tp/.ht120.mtc" (for Android M)

adb shell "/vendor/bin/thermal_manager /vendor/etc/.tp/.ht120.mtc"(for Android N)

(c)是否有效修改: cat /data/.tp/.settings 是

/etc/.tp/.ht120.mtc 或 /vendor/etc/.tp/.ht120.mtc代表修改成功

(2) Eng sw load关闭thermal throttling

(a) 工程模式下others-thermal 替换thermal policy 为high temp 120deg c, apply后即可

(b) adb shell "/system/bin/thermal_manager /etc/.tp/.ht120.mtc" (for Android M)

adb shell "/vendor/bin/thermal_manager /vendor/etc/.tp/.ht120.mtc"(for Android N)

(c)是否有效修改: cat /data/.tp/.settings 是

/etc/.tp/.ht120.mtc 或 /vendor/etc/.tp/.ht120.mtc代表修改成功

(3) 僅取消LTE throttling,保留其餘的thermal throttling功能

(a)adb shell "echo 1 120000 0 mtk-cl-shutdown02 0 0 no-cooler 0 0 no-cooler 0 0 no-cooler 0 0 no-cooler 0 0 no-cooler 0 0 no-cooler 0 0 no-cooler 0 0 no-cooler 1000 > /proc/driver/thermal/tzbtspa"

(b)是否有效修改 cat /proc/driver/thermal/tzbtspa, 如果

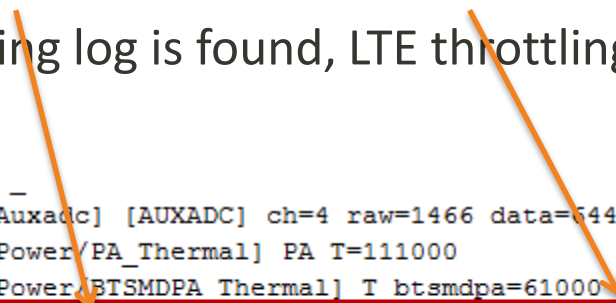
cooldev1=mtk-cl-mutt02,cooldev2=mtk-cl-mutt01,cooldev3=mtk-cl-mutt00, 变成cooldev1=no-cooler,cooldev2=no-cooler,cooldev3=no-cooler代表修改成功

How to check LTE throttling is triggered?

如何确认是Thermal的原因导致关机？也可以在Mobile Log中查到线索：

- How to Check if LTE throttling is triggered from mobilelog:
 - searching keyword in kernel_log
“thermal/cooler/mutt “ or “mtk_cl_mutt_set_mutt_limit”
 - If the following log is found, LTE throttling is triggered.

```
[0] [481:kworker/0:2] [Auxadc] [AUXADC] ch=4 raw=1466 data=544
[0] [481:kworker/0:2] [Power/PA_Thermal] PA T=111000
[0] [481:kworker/0:2] [Power/BTSMDDPA Thermal] T btssmdpa=61000
[0] [481:kworker/0:2] thermal/cooler/mutt [mtk_cl_mutt_set_mutt_limit] ret 0 param 20101 bcnt 11
[0] [481:kworker/0:2] [Power/BTS_Thermal] T_AP=56000
[0] [55:cfinteractive] [Power/cpufreq] _mt_cpufreq_power_limited_verify(): idx = 10, limited_max_
```



有时LTE Thermal Throttling 被触发的原因是Transceiver 的温度定义有错误而计算出了高于实际的温度，导致重启。这部分的Debug 有相关的SOP参考，如果需要请提交Eservice。



everyday genius