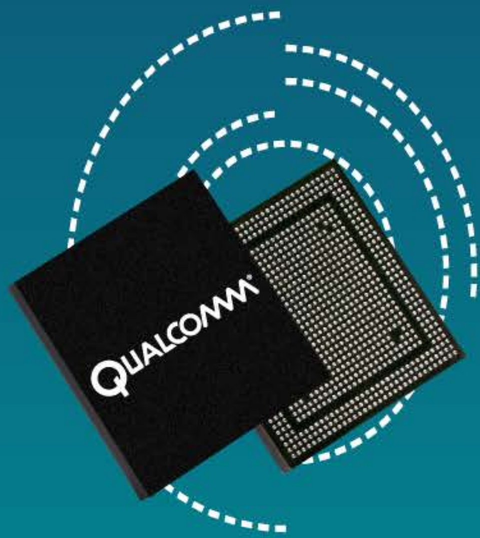


Thermal Tuning Procedure

80-N9649-1 D

Qualcomm Technologies, Inc.
5775 Morehouse Drive
San Diego, CA 92121
U.S.A.

Revision History

Revision	Date	Description
A	Feb 2012	Initial release
B	Feb 2013	Numerous changes were made to this document; it should be read in its entirety.
C	Mar 2014	Added software thermal requirement/sanity check, tuning procedures for dynamic algorithms. Added thermal calibration procedure for speaker coil protection. This document is only for the thermal engine-based solution.
D	Aug 2014	Numerous changes were made to this document; it should be read in its entirety.

Contents

Overview

Part 1 – Thermal Tuning Overview

- Thermal Tuning Overview
- Thermal Tuning Example
- Thermal Tuning Process Flow

Part 2 – Thermal Tuning Preparation

- Thermal Tuning Prerequisites
- Thermal Tuning Test Equipment
- How to Read Tsens (On-die Temperature)
- Adding External Thermistors
- Measuring Skin Temperature on Handheld Devices
- Example Use Cases for Thermal Tuning

Related document – [Q14]

Part 3 – Thermal Hardware and Software Functionality Verification

- Verification – Thermal Sensors
- Verification – Boot Thermal Monitor (BTM) Functionality
- Verification – Kernel Thermal Monitor (KTM) Functionality
- Verification – Complete Thermal Check

Part 4 – Thermal Tuning

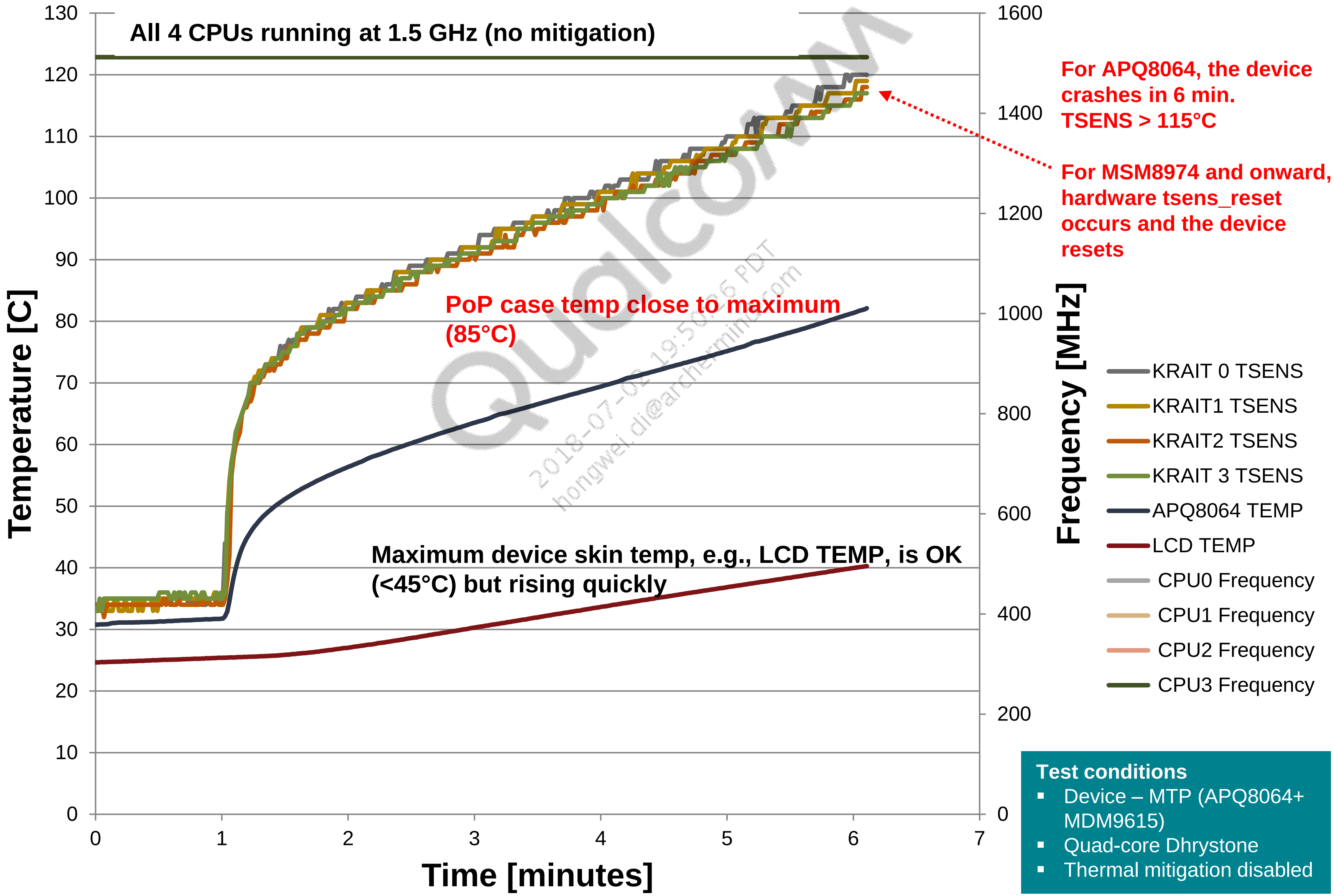
- Thermal Tuning Objectives
- Thermal Mitigation Algorithms
- DTM – Thermal Tuning Procedure
- Monitor – Thermal Tuning Procedure
- Thermal Tuning Example – Egypt 2.5 HD
- Sensor Aliases
- Thermal Calibration for Speaker Protection

Part 1 – Thermal Tuning Overview

Thermal Tuning Overview

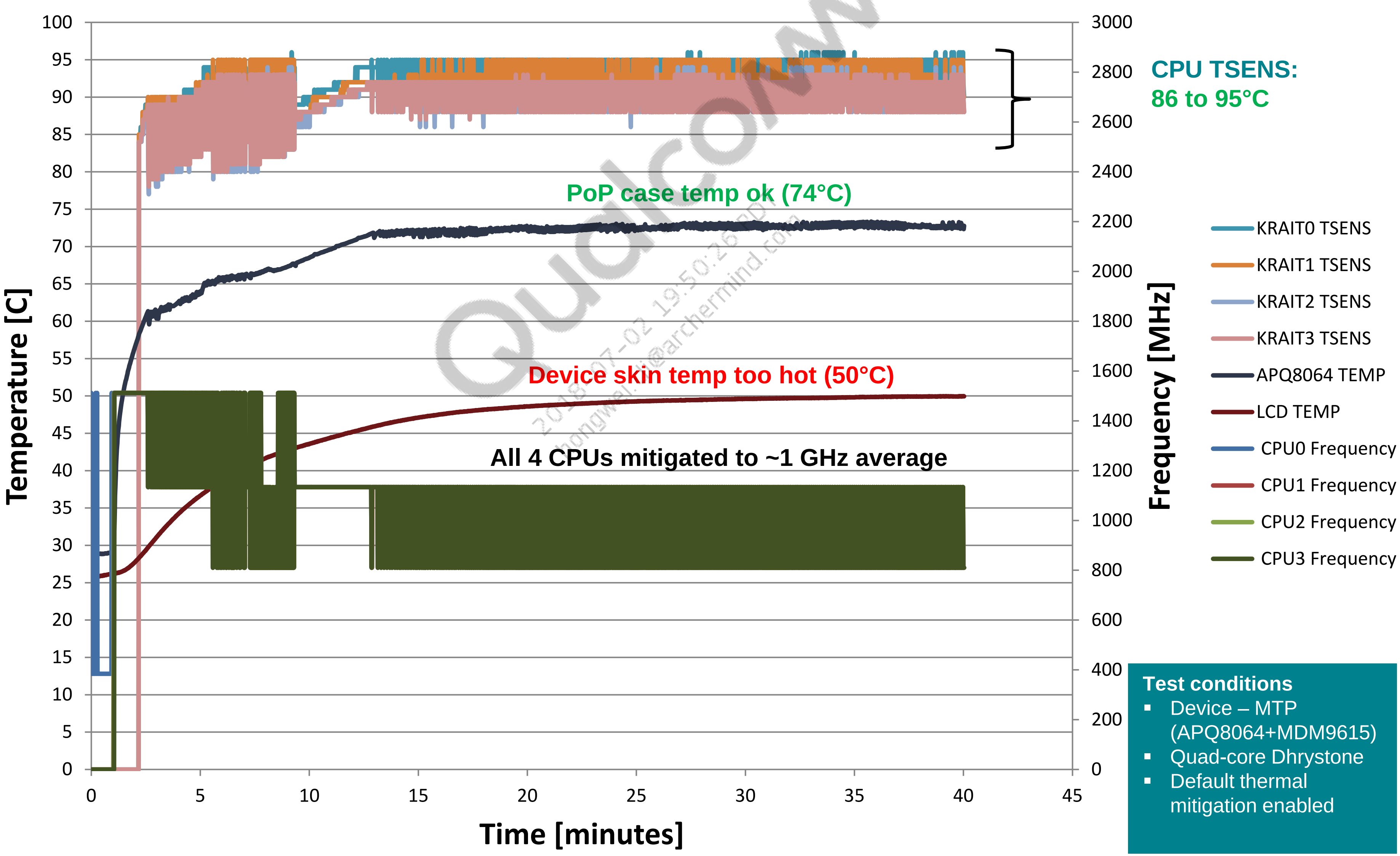
- What it is
  - Thermal tuning is the experimental process of determining the optimal thermal software configuration to manage device and OEM thermal specifications, while minimizing the impact on device performance. This is achieved through device testing across a variety of use cases.
  - **Note:** After the MD/ID is defined, software thermal mitigation is the last remaining method to manage excessive heat generation.
- Why it is needed
  - Maintains maximum chipset temperature limit; typically 85°C to 90°C
  - Maintains device skin temperature; typically 45°C
  - Allows the device to sustain the highest level of performance, e.g., highest CPU frequency, fps, and data rates, when thermal mitigation is necessary
- When it is needed
  - When Feature Complete (FC) software is available on chipsets that have open-loop AVS (also known as PVS) settings
  - **Note**: Waiting too long to verify thermal Key Performance Indicators (KPIs) may result in a delay of the customer launch. Starting too early, i.e., without open-loop AVS enabled or without FC software, may result in damaged devices or unnecessary time wasted in thermal and power debug.
  - After power optimizations are complete
  - After any change to the mechanical design
- The following slides show the result of thermal tuning to reduce on-die and skin temperatures.

Example – Thermal Mitigation Disabled



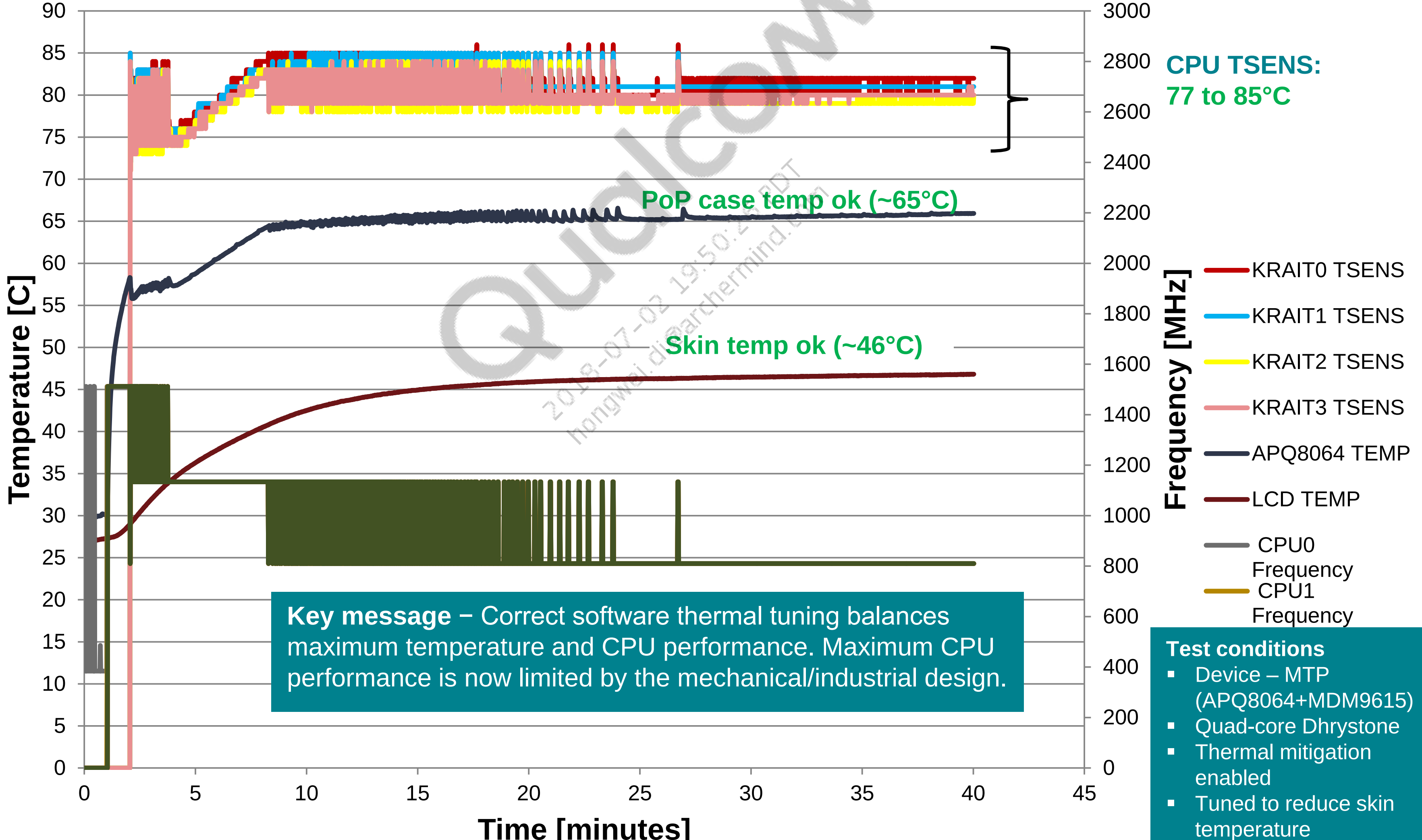
Test conditions
- Device – MTP (APQ8064+ MDM9615)
- Quad-core Dhrystone
- Thermal mitigation disabled

Example – Default Thermal Mitigation (Pretuning)



Test conditions
- Device – MTP (APQ8064+MDM9615)
- Quad-core Dhrystone
- Default thermal mitigation enabled

Example – Thermal Tuning Complete

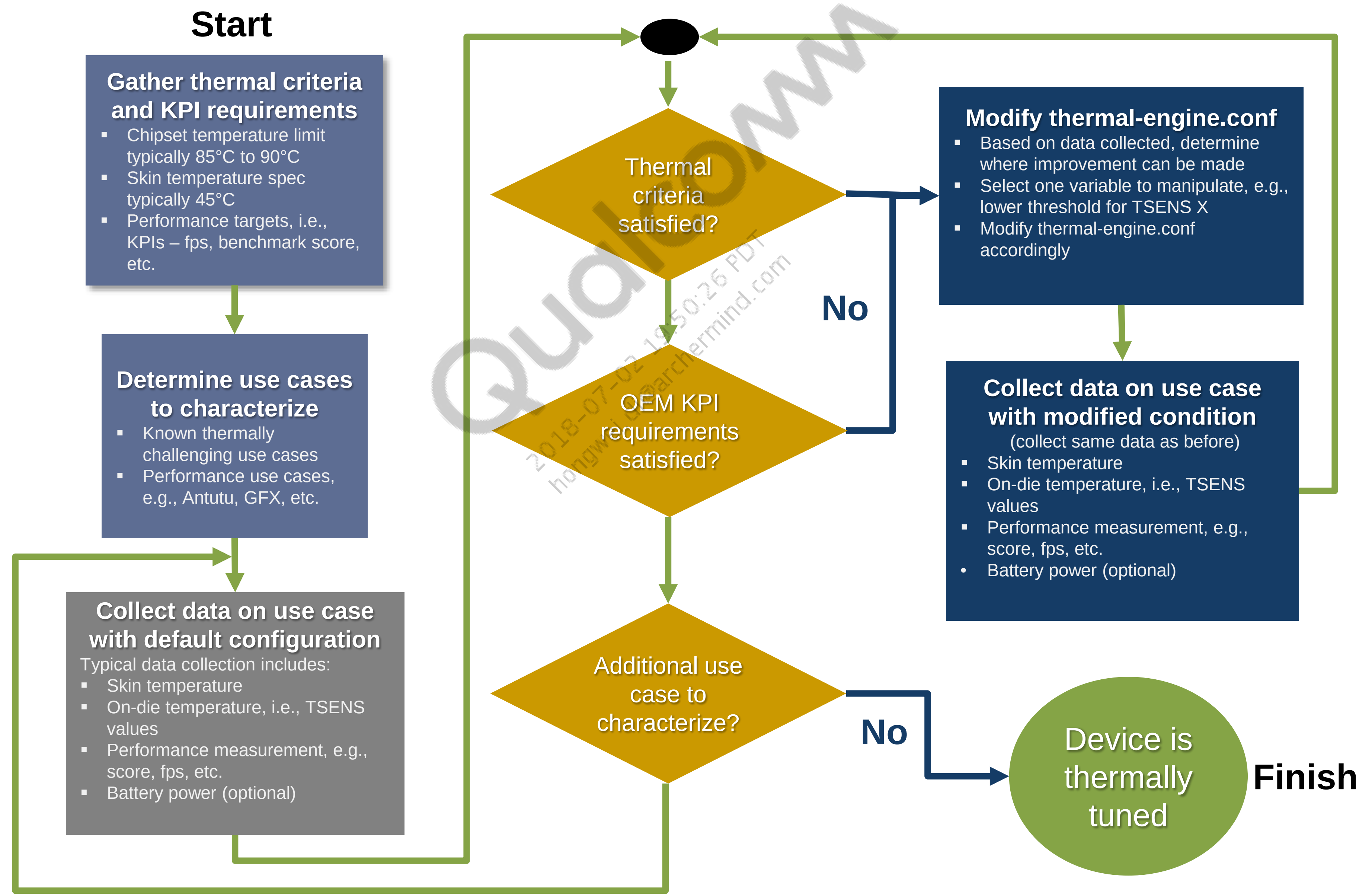


Key message – Correct software thermal tuning balances maximum temperature and CPU performance. Maximum CPU performance is now limited by the mechanical/industrial design.

Test conditions

- Device – MTP (APQ8064+MDM9615)
- Quad-core Dhrystone
- Thermal mitigation enabled
- Tuned to reduce skin temperature
- CPU average is 800 MHz

Thermal Tuning Process Flow



Part 2 – Thermal Tuning Preparation

Thermal Tuning Prerequisites

- Key factors required *prior* to thermal tuning
  - Power optimization efforts complete
  - PCB installed with all component shields into the final device skin; battery installed
  - Qualcomm Technologies, Inc. (QTI) devices with open-loop AVS enabled
    - Attempts to perform thermal tuning with initial ES devices (open–loop AVS not enabled) may result in exceeding chipset temperature limits, system crashes, and damage to the device. See the Device Revision Guide Release Notes to determine which devices are appropriate to use for thermal tuning.
  - QTI FC software
    - Thermal stress testing is used to identify hotspots on the Device Under Test (DUT) skin
    - Attempts to perform thermal tuning with non-FC software may result in exceeding chipset temperature limits, system crashes, and damage to the device. See the AMSS Release Notes to determine which builds are appropriate to use for thermal tuning.
  - **Note**: For a given chipset, use case, and operating frequency, the mechanical and industrial design (MD/ID) of each form factor has the most impact on the overall performance.

Thermal Tuning Test Equipment

- Below is a list of equipment required for thermal characterization and tuning. See Appendix B for an extended list of test equipment, materials, and tools.

Item	Description	Recommended brand, model
Data logger	Records values of thermocouples	Agilent, 34972A
Data module	Thermocouple module that inserts into the data logger	Agilent, 34901A
Thermocouples	Temperature-measuring sensor	Omega, 5TC-TT-K-36-72
IR camera with software	Thermal imaging for thermocouple attachment	FLIR, A655sc
Thermal chamber	For testing at specific ambient temperatures	Test Equity, 115A-B
Power monitor	Collects battery power consumption	Monsoon
Device mount	Suspends test device off table	PanaVise 201 Jr.
Kapton tape	Heat-tolerant tape used to attach thermocouples	3M, ¼ inch width

How to Read TSENS (On-Die Temperature)

- TSENS sensors measure on-die temperature at key points across the chip, e.g., each CPU, GPU, modem, etc. TSENS floor plans for each chipset are listed in the Linux Android Thermal Management document that corresponds to each chipset.
- The value of each TSENS sensor should be collected during profiling via the ADB.
  - cat /sys/devices/virtual/thermal/thermal_zoneN/temp displays the temperature of a TSENS sensor, where N is 0 through the maximum number of available TSENS, which varies by chipset.
  - cat /sys/devices/virtual/thermal/thermal_zoneN/type displays the name of the sensor, e.g., tsens_tz_sensor1.
  - The OEM should create a script to log available TSENS values at a rate of every 250 ms or higher.
- Additional thermal sensors may be available based on the chipset.
  - TSENS have a type of tsens_tz_sensorN and comprise thermal_zone0 through thermal_zoneN-1.
  - Starting at thermal_zoneN and onward, the following additional sensors may appear:
    - pmXXXX_tz – Thermal sensor for the PMIC, where XXXX is the PMIC number, commonly used in battery charge limiting
    - pmaXXXX_tz – Thermal sensor for the PMIC; only notifies thermal engine when specified threshold is crossed and does not provide real-time temperature data; commonly used for PMIC alarm monitoring, where sensor threshold is used to indicate when PMIC is drawing too much current
    - pa_therm0 – Used for modem-based mitigation
    - pa_therm1 – Used for modem-based mitigation
    - OEM-defined external thermistors; described on next slide

Adding External Thermistors

- External thermistors (recommended to achieve best overall thermal tuning and performance)
  - Placed by the OEM near key (hot) components, i.e., PMIC, PA, WLAN, and XO, on the PCB as part of board-level hardware design; OEM may also use Flex Printed Circuit (FPC) from thermistor vendors to attach thermistors directly onto the device skin
  - Preferred over TSENS for skin temperature management
  - *Not* preferred for on-die temperature management
  - Correlate better with the actual skin temperature that the user feels given that they react more slowly to on-die temperature fluctuations than TSENS
  - Enable greater sustained performance under common use cases
    - For example, the CPU may be mitigated less often given that the transient temperature fluctuations on-die have less impact.
- To enable external thermistors, there are required software changes in both the Linux kernel and thermal engine.
  - Linux kernel changes
    - Modify PMIC DTSI for the chipset, e.g., arch/arm/boot/dts/msm-pm8941.dtsi; example – PMIC for MSM8974
      - Add or replace the thermal node with a new channel (see example on next slide).
      - The explanation for the nodes is in the kernel/Documentation/devicetree/bindings/thermal/qpnp-adc-tm.txt file.
      - Thermistor can now be read through cat /sys/class/thermal/thermal_zoneN/type; N is the thermal zone number assigned for your sensor. When you cat type, the name that is added for the sensor appears.

Adding External Thermistors (cont.)

- Example using MSM8974 – Adding external thermistor to existing adc channel; see pm8941_vadc: vadc@3100 in arch/arm/boot/dts/msm-pm8941.dtsi
- An existing ADC channel is used by changing chan@b3 to the OEM-defined external thermistor settings.

Default

```
chan@b3 {
	label = "msm_therm";
	reg = <0xb3>;
	qcom,decimation = <0>;
	qcom,pre-div-channel-scaling = <0>;
	qcom,calibration-type = "ratiometric";
	qcom,scale-function = <2>;
	qcom,hw-settle-time = <2>;
	qcom,fast-avg-setup = <3>;
	qcom,btm-channel-number = <0x98>;
	qcom,thermal-node;
	};
```

Modified

```
chan@b3 {
	label = "<your sensor name>";
	reg = < your  setting>;
	qcom,decimation = < your  setting >;
	qcom,pre-div-channel-scaling = < your setting >;
	qcom,calibration-type = " your  setting ";
	qcom,scale-function = < your  setting >;
	qcom,hw-settle-time = < your  setting >;
	qcom,fast-avg-setup = < your  setting >;
	qcom,btm-channel-number = < your  setting >;
	qcom,thermal-node;
	};
```

Adding External Thermistors (cont.)

- Thermal engine changes
  - To enable the thermal engine to utilize the thermistor, it must be added to the sensors manager – /<android_branch>/vendor/qcom/proprietary/thermal-engine/sensors/sensors-8974.c
  - Case A – If sensor units are in Celsius, add to tsens_sensors[] within sensors-8974.c
  - Case B – If sensor units are in milliCelsius, add to gen_sensors[] within sensors-8974.c

Case A

```
static struct sensor_info tsens_sensors[] = {
 {
//add at end of sensors list…
     .name = "<your sensor name>",
     .setup = tsens_sensors_setup,
     .shutdown = tsens_sensors_shutdown,
     .get_temperature = tsens_sensor_get_temperature,
     .interrupt_wait = NULL,
     .update_thresholds = NULL,
     .tzn = 15,
     .data = NULL,
     .interrupt_enable = 0,
     },
};
```

Case B

```
static struct sensor_info gen_sensors[] = {
 {
//add at end of sensors list…
     .name = "<your sensor name>",
     .setup = gen_sensors_setup,
     .shutdown = gen_sensors_shutdown,
     .get_temperature = gen_sensor_get_temperature,
     .interrupt_wait = NULL,
     .update_thresholds = NULL,
     .tzn = 15,
     .data = NULL,
     .interrupt_enable = 0,
     }
};
```


Measuring Skin Temperature on Handheld Devices

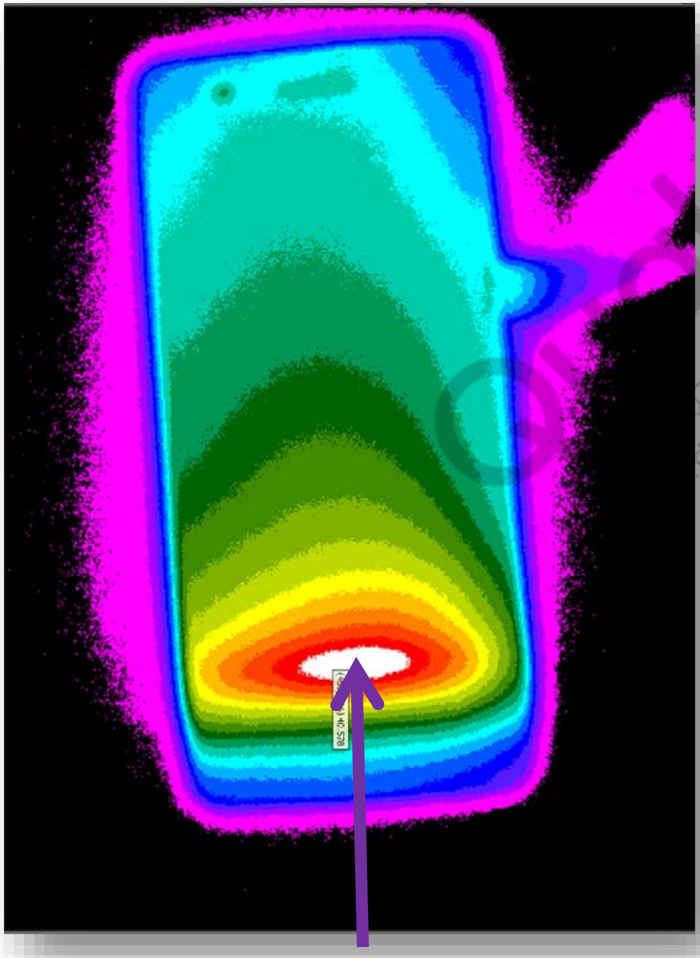
- Basic steps for setting up device to measure skin temperature
  1. For handheld devices, e.g., smartphones and tablets, it is necessary to measure device skin temperature during thermal testing/profiling to ensure user comfort and safety.
  2. Decide which use case to profile, e.g., for CPU-intensive, quad-core Dhrystone:
     - Adjust phone settings according to use case requirements, e.g., set device to Airplane mode, disable Wi-Fi, set LCD brightness, etc.
  3. Start the use case on the device.
  4. Identify hotspots on LCD (see example on next slide).
     - Capture thermal image with IR camera while running use case to identify hotspots
       - Tip: LCD is a very reflective surface that impacts the ability to identify a hotspot. Cover the LCD with nonreflective tape, e.g., black electrical tape. **Note**: Remove the tape before attaching thermocouples.
       - The use case should run long enough to develop easily identifiable hotspots (10+ min).
       - LCD usually has one to two hotspots, whereas the back cover usually has one hotspot.

Measuring Skin Temperature on Handheld Devices (cont.)

- Basic steps for setting up device to measure skin temperature (cont.)
  5. Attach thermocouples to LCD hotspots (see example on next slide)
     - Attach thermocouples to the exact center of the hotspots that were identified through the previous step.
     - **Note**: QTI uses the Agilent data logger to record thermocouple values; see Thermal Tuning Test Equipment for details.
  6. Repeat steps 3 and 4 for the back side of the phone.
  - Additional information
     - See Appendix A for comprehensive information about test setup, procedure, and equipment use.
     - Important – If skin temperature is measured exclusively with an IR camera, i.e., no thermocouples, see [Q14] for information about obtaining accurate skin temperature measurements.

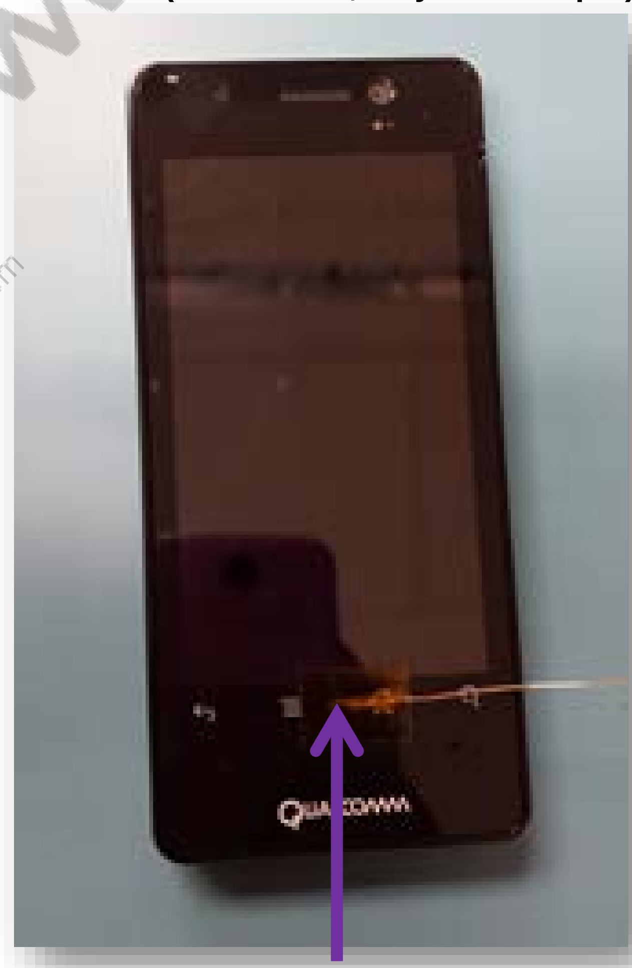
Measuring Skin Temperature on Handheld Devices (cont.)

Identify hotspots using IR camera



Hottest point on LCD side

Attach a thermocouple to each hotspot identified (in this case, only one hotspot).



Tip of thermocouple attached to center of hotspot

Example Use Cases for Thermal Tuning

- Tips for choosing use cases for thermal characterization/tuning
  - Choose use cases that are known to be thermally challenging, e.g., Dhrystone, camcorder
  - Thermal mitigation is not use case-specific, so it is important to choose and test a variety of different use cases, e.g., CPU intensive, GPU intensive, CPU/GPU mixed. This ensures that the resulting thermal configuration is not biased for a single or group of use cases.
  - Below are examples for reference; this does not comprise a comprehensive recommendation.

Use case	Type	Source	Notes
Dhrystone	CPU intensive	Open Source	▪ Dhrystone available through Docs and Downloads ▪ Run four instances to achieve quad-core condition, e.g., type adb shell and then type /data/dhrystone.sh & four times
TREX 2.7 (GFx)	GPU intensive, some CPU	Google Play	▪ Part of GFx Benchmark Suite (same as GL Benchmark) ▪ Run one iteration of TREX to produce benchmark score or repeat indefinitely to simulate intense graphics use case ▪ Capture fps for comparing changes in performance
Egypt 2.5 (GFx)	CPU/GPU intensive	Google Play	▪ Part of GFx Benchmark Suite (same as GL Benchmark) ▪ Run one iteration of Egypt to produce a benchmark score or repeat indefinitely to simulate an intense graphics use case ▪ Capture fps for comparing changes in performance
Antutu	CPU intensive	Google Play	▪ Produces benchmark score ▪ QTI currently uses Antutu v. 4.1.1
Camcorder	CPU intensive, some GPU	Default camera app	▪ 1080p and 4 K resolution video are known to be thermally challenging ▪ Capture video fps for comparing changes in performance
Asphalt Assault	CPU/GPU intensive	Google Play	▪ Popular racing game ▪ For consistency between tests, only touchscreen to get car out of trouble; otherwise, let the car run the course with little to no interference; this ensures greater repeatability

Part 3 – Thermal Hardware and Software Verification

Thermal Software Requirements and Verification

- Software thermal requirements
  - QTI thermal software and OEM-specific thermal software should handle all the thermal threats from the boot.
  - QTI strongly recommends that the following software components are enabled correctly at all times:
    - Boot Thermal Monitor (BTM) – Optional
    - Kernel Thermal Monitor (KTM) – Mandatory
    - Thermal engine with correct configuration file – Mandatory unless OEM incorporates third-party thermal management software with equivalent functionality
- Thermal software verification
  - Verify internal/external sensor calibration
  - Verify BTM
  - Verify KTM and kernel-level emergency throttling
  - Verify unintended performance-boosting mechanisms
  - Verify full thermal software
  - The following slides show how to perform basic checks before thermal tuning.

Verification – Thermal Sensors

- Thermal sensors
  - Internal TSENS
    - Depending on chipsets, up to 11 internal TSENS sensors are located on or around hotspots on the MSM™/APQ/MDM.
    - All TSENS sensor values are accessible through the kernel TSENS driver and are exposed to sysfs nodes for user space accesses.
      - cat /sys/devices/virtual/thermal/thermal_zone[A-B]/temp, where A and B are numbers and vary with chipset and configuration
  - External thermistors
    - OEM-specific, but generally most external sensor nodes are exposed to sysfs nodes
    - For thermistors connected to the PMIC ADC, available sysfs nodes are:
      - cat /sys/devices/virtual/thermal/thermal_zone[C-D]/temp, where C and D are numbers and vary with chipset and configuration
- Check thermal sensor calibration status
  1. Place the DUT into the temperature chamber at 50°C.
  2. Ensure thermal mitigation is disabled and the device is idling.
  3. Soak for at least 30 min.
  4. Read sensor values and thermocouples; the temperature value should be 50°C with reasonable variations (±1.5ºC for TSENS).

Verification – BTM Functionality

- Verify BOOT_TEMP_CHECK_THRESHOLD_DEGC is defined in the boot loader.
- The boot procedure is delayed by MAX_TEMP_CHK_ITER*MAX_WAIT_TIME_MICROSEC ms, which is defined at /core/boot/secboot3/src/boot_thermal_management.c.
- Only when the temperature is not within the following two limits hardcoded at /boot_images/core/hwengines/tsens/config/8974 (target name)/BootTempCheckBsp.c:

```
const BootTempCheckBspType BootTempCheckBsp[] = {
   {
      /* .nUpperThresholdDegC */ 150,
      /* .nLowerThresholdDegC */ -150
   }
};
```

- Check BTM functionality
  1. Change nUpperThresholdDegC to 40°C and recompile and load the new image.
  2. Place the DUT into the temperature chamber at 50°C.
  3. Soak for at least 30 min and try to boot the device.
  4. If the device does not boot, which is a correct action, cool down the device at room temperature at least 30 min and try to boot the device again; the device must boot correctly.
  5. Reiterate the test for a low temperature case by updating nLowerThresholdDegC, if required.

Verification – KTM Functionality

- Verify KTM thermal configuration and operation with the kernel message
  - Defined at the qcom,msm-thermal section in the dtsi file (device tree) for respective chipsets; following is an example for MSM8974

```
qcom,msm-thermal {
        compatible = "qcom,msm-thermal";
        qcom,sensor-id = <5>;
        qcom,poll-ms = <250>;
        qcom,limit-temp = <60>;
        qcom,temp-hysteresis = <10>;
        qcom,freq-step = <2>;
        qcom,freq-control-mask = <0xf>;
        qcom,core-limit-temp = <80>;
        qcom,core-temp-hysteresis = <10>;
        qcom,core-control-mask = <0xe>;
        qcom,hotplug-temp = <110>;
        qcom,hotplug-temp-hysteresis = <20>;
        qcom,cpu-sensors = "tsens_tz_sensor5", "tsens_tz_sensor6", "tsens_tz_sensor7",
        "tsens_tz_sensor8";
        qcom,freq-mitigation-temp = <110>;
        qcom,freq-mitigation-temp-hysteresis = <20>;
        qcom,freq-mitigation-value = <960000>;
        qcom,freq-mitigation-control-mask = <0x01>;
        qcom,vdd-restriction-temp = <5>;
        qcom,vdd-restriction-temp-hysteresis = <10>;
        qcom,pmic-sw-mode-temp = <85>;
        qcom,pmic-sw-mode-temp-hysteresis = <75>;
        qcom,pmic-sw-mode-regs = "vdd-dig";
        vdd-dig-supply = <&pm8841_s2_floor_corner>;
        vdd-gfx-supply = <&pm8841_s4_floor_corner>;
```

Some features are only for MSM8974 and later chipsets

Verification – KTM Functionality (cont.)

- Verify KTM thermal configuration and operation with the kernel message (cont.)
  - Defined at the qcom,msm-thermal section in the dtsi file (device tree) for respective chipsets; following is an example for MSM8974 (cont.)

```
qcom,vdd-dig-rstr{
           qcom,vdd-rstr-reg = "vdd-dig";
           qcom,levels = <5 7 7>; /* Nominal, Super Turbo, Super Turbo */
           qcom,min-level = <1>; /* No Request */
};
qcom,vdd-gfx-rstr{
           qcom,vdd-rstr-reg = "vdd-gfx";
           qcom,levels = <5 7 7>; /* Nominal, Super Turbo, Super Turbo */
           qcom,min-level = <1>; /* No Request */
};
           qcom,vdd-apps-rstr{
           qcom,vdd-rstr-reg = "vdd-apps";
           qcom,levels = <1881600 1958400 2265600>;
           qcom,freq-req;
};
};
```

Some features are only for MSM8974 and later chipsets

Verification – KTM Functionality (cont.)

- Verify KTM frequency mitigation
  - Reduce the threshold and determine if CPU mitigation occurs through the kernel message
    - Update qcom,limit-temp = <40>; and recompile and load the new image.
    - Reboot the device.
    - Every sampling period (qcom,poll-ms = <250>), CPU frequency is throttled down to min (keyword: msm_thermal).
      - <6>[2.243338] msm_thermal: Limiting cpu0 max frequency to 384000
      - <6>[2.243338] msm_thermal: Limiting cpu1 max frequency to 384000
      - <6>[2.243338] msm_thermal: Limiting cpu2 max frequency to 384000
      - <6>[2.243369] msm_thermal: Limiting cpu3 max frequency to 384000

Verification – KTM Functionality (cont.)

- Verify KTM thermal emergency throttling for MSM8974 and later chipsets only.
  - CPUs are hotplugged when the temperature exceeds the threshold (qcom, freq-mitigation-temp = <110>) and brought back if the temperature lowers (qcom,freq-mitigation-temp = <110> - qcom,freq-mitigation-temp-hysteresis = <20>).
  - Emergency CPU throttling is initiated to mitigate CPU frequency to (qcom,freq-mitigation-value = <960000>) when the temperature exceeds the threshold (qcom,freq-mitigation-temp = <110>); CPU mitigation ends after the temperature lowers (qcom,freq-mitigation-temp = <110> - qcom,freq-mitigation-temp-hysteresis = <20>).
- Emergency mitigation protects the system even without the thermal engine.
  - Do not run the thermal engine (stopping thermal-engine service).
    - #stop thermal-engine from adb shell
  - Run highly stressful apps, e.g., quad-Dhrystone, Antutu, or StabilityApp, for at least 10 min without TSENS reset; reset for other reasons, e.g., watchdog, can still occur.

Verification – Thermal Malware

- Various mechanisms exist that impact thermal characteristics by forcefully altering the operating conditions of each component (mostly CPU frequency/GPU frequency/bus frequency/power rail voltages).
- Beware of the following, which can impact thermal behaviors:
  - PerfLocks – Explicit user space or kernel space perflocks to boost CPU frequency or the number of CPU cores at runtime
  - PowerHAL updates – System configuration updates (for various hints) that impact power/performance/thermal
  - Performance boosters – Boost performance mostly from input events
  - Detectors – Boost performance mostly on specific set of apps or patterns of behaviors
  - Static/dynamic changes on configurations (power/thermal vs performance)
    - CPU frequency governor parameters
    - MP-decision parameters
    - Thermal engine parameters (dynamic)

Verification – Complete Thermal Sanity

- First verify if the thermal engine is running as service
  - ps | grep thermal-engine displays the current instance of the thermal engine. If not, the thermal engine must be started as service.
  - Try to kill the thermal engine by issuing kill -9 pid_of_thermal_engine and then check again with ps | grep thermal-engine to determine whether the thermal engine is automatically restarted.
- Thermal engine configuration
  - For older targets, there are two sources of default configurations – Hardcoded embedded rules and default thermal engine configuration file, /etc/thermal-engine.conf.
  - For APQ8084 and later targets, only embedded rules exist.
  - By performing thermal-engine –o, default configurations can be verified.
- Thermal engine + kernel emergency mitigation protects the system from thermal threats for MSM8974 and later chipsets only.
  1. Place the DUT into the temperature chamber at 50°C.
  2. Run highly stressful apps, e.g., quad-Dhrystone, Antutu or StabilityApp, for at least 5 min without a crash.
  3. Kill the thermal engine and run for an additional 5 min.
  4. Stop the thermal engine and run for an additional 5 min.
  5. Verify no tsens_reset occurs by looking at the reset reason if the device resets.

Part 4 – Thermal Tuning Procedure

Thermal Tuning Objectives

- Maintain chipset thermal limits per device spec, typically 85°C
- Maintain device skin temperature, typically 45°C
- Allow the device to sustain the highest level of performance, e.g., the highest CPU frequency, fps, and data rates, when thermal mitigation is necessary
- Also see *Thermal Tuning Process Flow* presented in *Thermal Tuning Overview* at the beginning of this document.

Thermal Mitigation Algorithms

- Dynamic Thermal Management (DTM) for CPU and GPU mitigation
  - OEM determines a single temperature at which to maintain, and the algorithm dynamically throttles an OEM-specified action, e.g., CPU frequency, accordingly
  - Used for skin temperature and on-die temperature control; requires less tuning effort than monitor and enables stricter enforcement of temperature limits
  - Only works for CPU and GPU mitigation, i.e., lowering maximum frequencies
  - Threshold changes for the DTM algorithm are made via set_point and set_point_clr in thermal-engine.conf
- Monitor algorithm for LCD, modem, camcorder, battery
  - OEM determines a series of temperature thresholds and precise corresponding action per threshold; many examples to follow
  - Monitor algorithm is used for LCD, modem, camcorder, and battery mitigation; not recommended for CPU and GPU mitigation, as it requires extensive tuning to find each setpoint
  - Threshold changes for the monitor algorithm are made via thresholds and threshold_clr in thermal-engine.conf
- Generally speaking, DTM is strongly recommended over monitor for CPU and GPU mitigation because it greatly relieves tuning effort, boosts performance, and more strictly maintains temperature to OEM setpoint
- For additional information, see Linux Android Power Management and Thermal Management Overview, which is released for each chipset

DTM – Thermal Tuning Procedure Using Dhrystone

1. Place the DUT into the temperature chamber at 25°C.
2. Ensure thermal mitigation is enabled, i.e., ps | grep thermal-engine from the adb shell.
3. Read the thermocouple; the temperature value should be ~25°C since the ambient is at 25°C.
4. Turn on the DUT.
5. Connect the micro USB cable.
6. Open a command prompt on the PC and enter the case-sensitive command adb root.
7. Disable Wi-Fi and enable Airplane mode.
8. Go to Settings→Display settings and turn on the brightness to maximum.
9. In Display settings, set Sleep mode/screen timeout to Never.
10. Open a command prompt, type adb root, adb shell, and look for root@android. Keep the USB cable plugged in.
11. Start TSENS and thermocouple logging simultaneously.
12. Start the Agilent Data Logger for thermocouple temperature logging.
13. Open another command prompt and type adb shell. Start the TSENS logging app, if available. Otherwise, try to grep temperature information from the thermal engine logcat log (logcat –v time –s ThermalEngine).
14. Wait 1 min to allow synchronization.

DTM – Thermal Tuning Procedure using Dhrystone

15. Run Dhrystone on each CPU core (this heats up the device as fast as possible), e.g., for a four-core device, run four instances of Dhrystone from an adb command line.
16. Log thermocouple and TSENS temperatures until:
    - The chipset exceeds its maximum temperature limit, typically 85°C or
    - The skin hotspot temperature rises more than acceptable limits, typically 45°C or
    - The system crashes.
17. If the system crashes before the maximum temperature limit or skin hotspot conditions are reached, go back to step 15. However, run Dhrystone on one less core to reduce the temperature, e.g., run Dhrystone on three cores instead of four.
18. Pick the TSENS that closely tracks the skin hotspot/chipset temperature limit and ensure this sensor is not on the CPU (Krait).

DTM – Thermal Tuning Procedure using Dhrystone (cont.)

19. Disable embedded rules if not needed

[PID-CPU0]
disable 1
[PID-CPU1]
disable 1
[PID-CPU2]
disable 1
[PID-CPU3]
disable 1
[PID-POPMEM]
disable 1

20. Edit the default thermal-engine.conf file as follows:

[SS-CPU0]

algo_type	ss	Algorithm type – SS
sensor	cpu0	Sensor – cpu0 maps to tsens_tz_sensor5 for MSM8974 based on alias table
sampling	65	65 ms sampling rate
device	cpu	Mitigation device: for all CPUs
set_point	90000	Mitigation target at 90°C and sampling starts once this threshold crosses over
set_point_clr	55000	Sampling ends after temperature goes below this threshold, 55°C

21. If the skin hotspot temperature is far below acceptable limits, increase set_point and set_point_clr values by 5°C and return to step 15. If the skin hotspot temperature exceeds acceptable limits, decrease set_point and set_point_clr values by 5°C and return to step 15.

DTM – Thermal Tuning Procedure using Dhrystone (cont.)

22. Edit the default thermal-engine.conf file as follows:
    - If temperature is low on the device skin, increase all temperature thresholds by 5°C.

[SS-CPU0]

algo_type	ss	Algorithm type – SS
sensor	cpu0	Sensor – cpu0 maps to tsens_tz_sensor5 for MSM8974 based on alias table
sampling	65	65 ms sampling rate
device	cpu	Mitigation device: for all CPUs
set_point	95000	Mitigation target at 95ºC and sampling starts after this threshold crosses over
set_point_clr	60000	Sampling ends after temperature goes below this threshold, 60°C

- If temperature is high on the device skin, decrease all temperature thresholds by 5°C.

[SS-CPU0]

algo_type	ss	Algorithm type – SS
sensor	cpu0	Sensor – cpu0 maps to tsens_tz_sensor5 for MSM8974 based on alias table
sampling	65	65 ms sampling rate
device	cpu	Mitigation device: for all CPUs
set_point	85000	Mitigation target at 85°C and sampling starts after this threshold crosses over
set_point_clr	50000	Sampling ends after temperature goes below this threshold, 50°C

23. If the TSENS temperature no longer increases over time and skin hotspot temperature limits are within acceptable limits, on-die tuning is complete.

Sample thermal-engine.conf File (DTM)

```
[PID-CPU0]
disable        1
[PID-CPU1]
disable        1
[PID-CPU2]
disable        1
[PID-CPU3]
disable        1
[PID-POPMEM]
disable        1

[SS-CPU0]
algo_type      ss
sensor         cpu0
sampling       65
device         cpu
set_point      90000
set_point_clr 55000

[SS-CPU1]
algo_type      ss
sensor         cpu1
sampling       65
device         cpu
set_point      90000
set_point_clr 55000
```

Embedded PID rules are disabled

DTM algorithm for CPU mitigation

```
[SS-CPU2]
algo_type ss
sensor         cpu2
sampling       65
device         cpu
set_point      90000
set_point_clr 55000

[SS-CPU3]
algo_type      ss
sensor         cpu3
sampling       65
device         cpu
set_point      90000
set_point_clr 55000

[SS-POPMEM]
algo_type      ss
sensor         pop_mem
sampling       65
device         cpu
set_point      80000
set_point_clr 50000
time_constant 16
```

Monitor – Thermal Tuning Procedure Using Dhrystone

1. Place the DUT into the temperature chamber at 25°C.
2. Ensure thermal mitigation is enabled, i.e., ps | grep thermal-engine from adb shell.
3. Read the thermocouple; the temperature value should be ~25°C since the ambient is at 25°C.
4. Turn on the DUT.
5. Connect the micro USB cable.
6. Disable Wi-Fi and enable Airplane mode.
7. Go to Settings→Display settings and turn on the brightness to maximum.
8. In Display settings, set Sleep mode/screen timeout to Never.
9. Open a command prompt and type adb root, adb shell, and look for root@android. Keep the USB cable plugged in.
10. Start TSENS and thermocouple logging simultaneously.
11. Start the Agilent Data Logger for thermocouple temperature logging.
12. Open another command prompt and type adb shell. Start the TSENS logging app, if available. Otherwise, try to grep temperature information from thermal-engine logcat log (logcat –v time –s ThermalEngine).
13. Wait 1 min to allow synchronization.

Monitor – Thermal Tuning Procedure Using Dhrystone (cont.)

14. Run Dhrystone on each CPU core (this heats up the device as quickly as possible), e.g., for a four-core device, run four instances of Dhrystone from the adb command line.
15. Log thermocouple and TSENS temperatures until the chipset exceeds its maximum temperature limit, typically 85°C, *or* until the skin hotspot temperature rises more than acceptable limits, typically 45°C, *or* the system crashes.
16. If the system crashes before maximum temperature limit or skin hotspot conditions are reached, go back to step 15. However, run Dhrystone on one less core to reduce the temperature, e.g., run Dhrystone on three cores instead of four.
17. Pick the TSENS that closely tracks the skin hotspot/chipset temperature limit and make sure this sensor is not on the CPU (Krait).
18. Edit the default thermal-engine.conf file as follows:

[CPU-MONITOR-EXAMPLE]					Rule name
algo_type	monitor				Algorithm type
sensor	tsens_tz_sensorX				Temp sensor # on the chipset die
sampling	1000				Temp sensor sampling rate in ms
thresholds	90000	95000	100000	120000	Enable temp in °mC
thresholds_clr	85000	90000	95000	115000	Disable temp in °mC
actions	cpu	cpu	cpu	shutdown	Thermal mitigation type
action_info	1512000	1296000	918000	5000	CPU clock frequency in Hz

19. If the skin hotspot temperature is far below acceptable limits, increase thresholds and threshold_clr values by 5°C and return to step 15. If the skin hotspot temperature exceeds acceptable limits, decrease thresholds and threshold_clr values by 5°C and return to step 15.

Monitor – Thermal Tuning Procedure Using Dhrystone (cont.)

20. Edit the default thermal-engine.conf file as follows:

- If temperature is low on the device skin, increase temperature thresholds by 5°C.

[CPU-MONITOR-EXAMPLE]					Rule name
algo_type	monitor				Algorithm type
sensor	tsens_tz_sensorX				Temp sensor # on the chipset die
sampling	1000				Temp sensor sampling rate in ms
thresholds	**95000**	**100000**	**105000**	120000	Enable temp in °mC
thresholds_clr	**90000**	**95000**	**100000**	115000	Disable temp in °mC
actions	cpu	cpu	cpu	shutdown	Thermal mitigation type
action_info	1512000	1296000	918000	5000	CPU clock frequency

- If temperature is high on the device skin, decrease temperature thresholds by 5°C.

[CPU-MONITOR-EXAMPLE]					Rule name
algo_type	monitor				Algorithm type
sensor	tsens_tz_sensorX				Temp sensor # on the chipset die
sampling	1000				Temp sensor sampling rate in ms
thresholds	**85000**	**90000**	**95000**	120000	Enable temp in °mC
thresholds_clr	**80000**	**85000**	**90000**	115000	Disable temp in °mC
actions	cpu	cpu	cpu	shutdown	Thermal mitigation type
action_info	1512000	1296000	918000	5000	CPU clock frequency in Hz

21. If TSENS temperature no longer increases over time and skin hotspot temperature limits are within acceptable limits, tuning is complete.

Monitor – thermal-engine.conf File Example

```
[tsens_tz_sensorX]
algo_type          monitor ← (Algorithm type)
sensor             tsens_tz_sensorX ← (Temp sensor # on the chipset die)
sampling           1000 ← (Temp sensor sampling rate in ms)
thresholds         75000   78000  81000  84000  87000  90000    ← Enable temp in °mC
thresholds_clr     72000   75000  78000  81000  84000  87000    ← Disable temp in °mC
actions            cpu     cpu    cpu    cpu    cpu    shutdown ← Mitigation type
action_info        1296000 1188000 918000 756000 648000 5000    ← CPU freq in Hz
```

Increasing temperature decreases CPU clock rates

Monitor – thermal-engine.conf File Example (Customer Must Edit)

```
debug
sampling         5000

[MODEM-EXAMPLE]
algo_type        monitor
sensor           pa_therm0
sampling         1000
thresholds       70000 80000  90000
thresholds_clr   65000 75000  85000
actions          modem    modem    modem
action_info      1    2    3

[CPU-USING-TSENS]
algo_type        monitor
sensor           tsens_tz_sensor0
sampling         1000
thresholds       65000   90000   93000   96000   99000   102000  105000
thresholds_clr   62000   87000   90000   93000   96000   99000   102000
actions          cpu     cpu     cpu     cpu     cpu     cpu     shutdown
action_info      1512000 1296000 1188000 918000  756000  648000  5000
```

Modem mitigation based on PA thermister

CPU mitigation based on internal temperature sensors

Thermal Tuning Example – Egypt, Graphics Use Case

- The following slides show an example of thermal tuning using the graphics use case Egypt 2.5 HD on the MSM8974 chipset.
- The example shows the results of three configuration changes:
  - Thermal Config 1 – CPU-only mitigation for managing on-die temperature
    - This config shows how using rules that are set *only* to manage on-die temperature allow skin temperature to reach extreme levels.
  - Thermal Config 2 – Add rule for skin temperature
    - This config addition shows how adding a rule for skin temperature based on an external thermistor maintains skin temperature at a target of 45°C.
  - Thermal Config 3 – Add GPU mitigation rule
    - This config addition shows how using a combination of CPU and GPU mitigation can improve performance over the CPU mitigation-only condition.

Config 1 – CPU-Only Mitigation with No Skin Temperature Mitigation

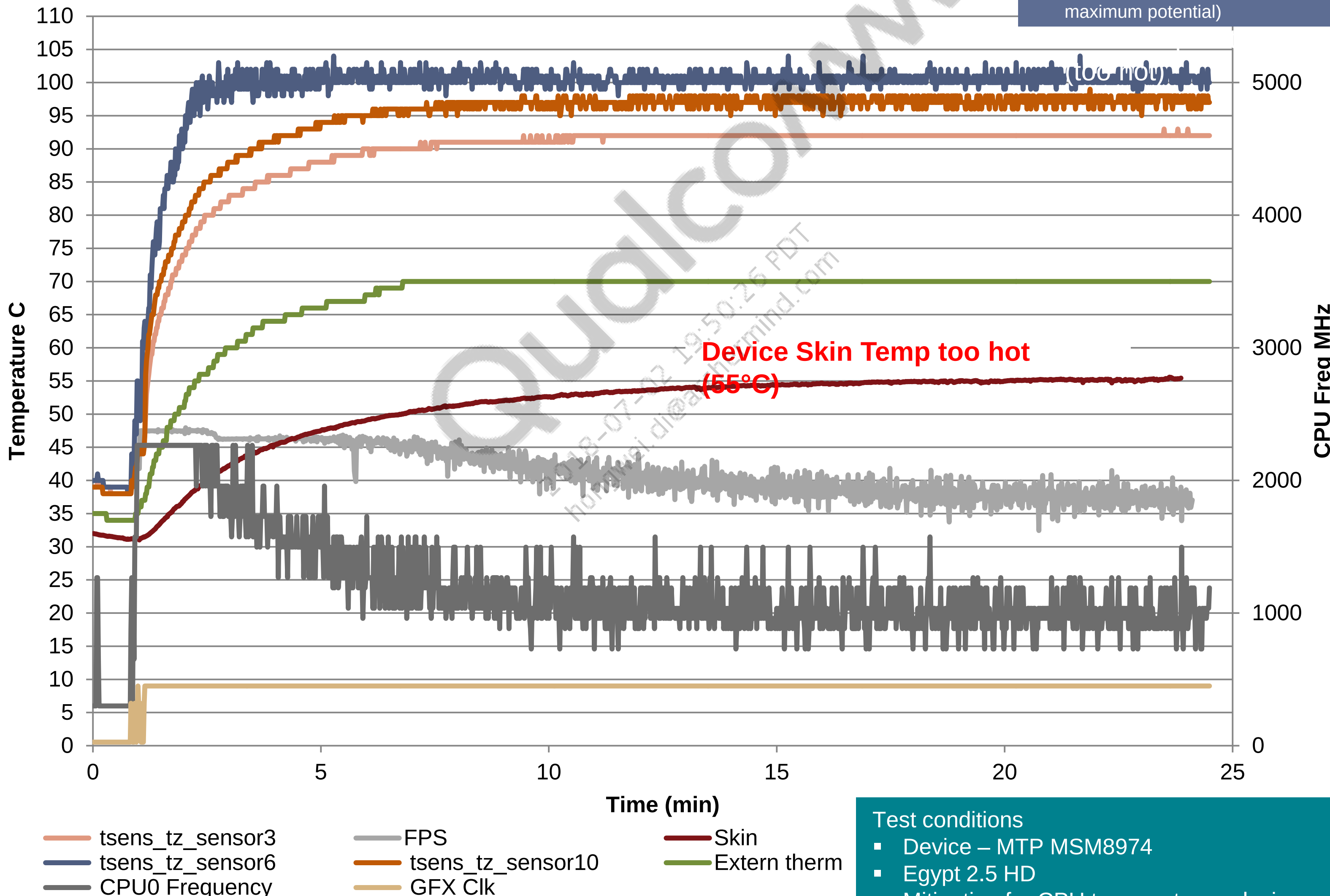
- CPU-only mitigation to maintain on-die temperature
- No skin temperature mitigation defined

[SS-CPUx]		For CPU 0 through CPU3
algo_type	ss	Algorithm type – SS
sampling	65	65 ms sampling rate
sensor	cpu0	
device	cpu	
set_point	95000	Target temperature of 95°C
set_point_clr	55000	

- This example only shows a rule for cpu0; however, the above rule is typically repeated in thermal-engine.conf for each CPU.
- The graph on the next slide displays thermal characteristics of the MSM8974 chipset for the Egypt 2.5 HD benchmark program without mitigation for skin.

Results Based on Config 1

Results of adding CPU-only rule:
- Average fps – 41.3 (degrades 12.86% from maximum potential)



Test conditions
- Device – MTP MSM8974
- Egypt 2.5 HD
- Mitigation for CPU temperature only, i.e., no skin temperature mitigation

Config 2 – Add Rule for Skin Temp Control

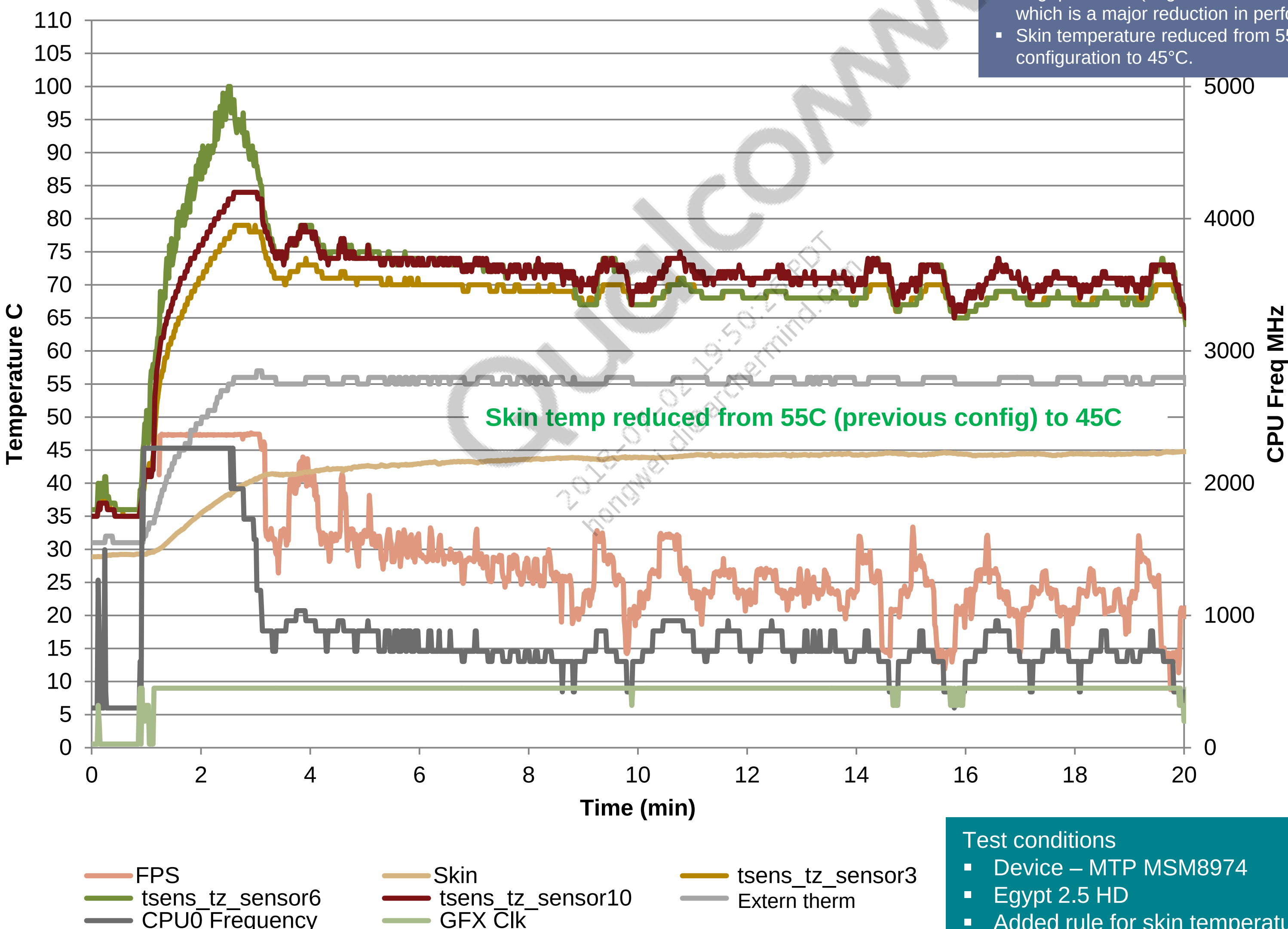
- Based on Config 1, skin temperature is too hot.
- The goal is to reduce skin temperature from 55°C to 45°C.
- For this additional rule, an external thermistor named skin is employed.
  - External thermistors are placed by the OEM and are better than TSENS sensors for correlating with the actual skin temperature that the user feels. One reason for this is that external thermistors react more slowly to on-die temperature fluctuations.
  - Notice the large time_constant (154) used in the config. The time_constant is a multiplier of the sampling period for holding off adjustments when the current and last error sample are equivalent. This improves performance by not mitigating on transient temperature spikes.
- The following rule is in addition to the rules from Config 1.

```
[SS-SKIN]
algo_type                 ss        Algorithm type – SS
sampling                  65        65 ms sampling rate
sensor                    skin      External thermistor on PCB
device                    cpu       Mitigation device – for all CPUs
set_point                 55000     Mitigation target at 55°C
set_point_clr    45000    Sampling ends after temperature goes below this threshold
time_constant     154     154*sampling = 10 sec of sampling if no change on sensor
```

- The graph on the next slide displays thermal characteristics of the MSM8974 chipset for the Egypt 2.5 HD benchmark program with CPU mitigation for skin.

Results Based on Config 2

Results of adding skin temp rule
- Avg fps – 26.3 (degrades 44% from maximum potential), which is a major reduction in performance
- Skin temperature reduced from 55°C in previous configuration to 45°C.



Test conditions
- Device – MTP MSM8974
- Egypt 2.5 HD
- Added rule for skin temperature mitigation

Config 3 – Add Rule for GPU Control

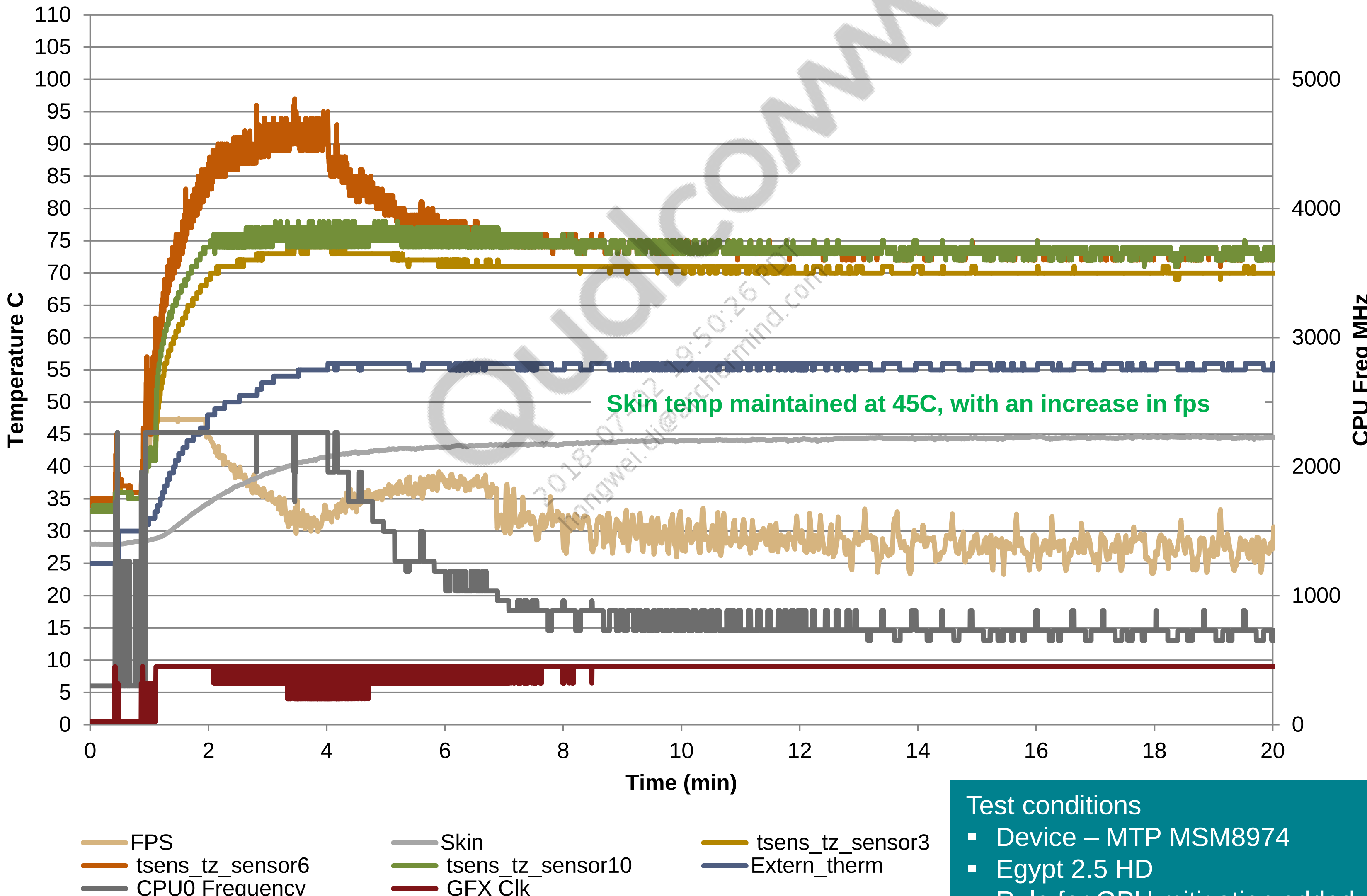
- Results based on Config 2 show that managing for skin temperature results in a major performance reduction, i.e., lower fps.
- GPU mitigation with CPU mitigation can help boost performance for GPU-intensive apps. **Note**: Need to profile which combination can help the most (CPU only mitigation, GPU only mitigation or CPU+GPU mitigation.
- This rule is in addition to the rules from Config 1 and 2.

[SS-GPU]

algo_type	ss	Algorithm type – SS
sampling	65	65 ms sampling rate
Sensor	tsens_tz_sensor10	Sensor – tsens_tz_sensor10 in MSM8974
device	gpu	Mitigation device – for GPU
set_point	75000	Mitigation target at 75°C
set_point_clr	55000	Sampling ends after temperature goes below this threshold

- The graph on the next slide displays thermal characteristics of the MSM8974 chipset for the Egypt 2.5 HD benchmark program with CPU+GPU mitigation for skin.

Results Based on Config 3



Test conditions
- Device – MTP MSM8974
- Egypt 2.5 HD
- Rule for GPU mitigation added

Results Based on Config 3 (cont.)

- Results of adding GPU mitigation
  - Avg fps – 30.54 (degrades 35.65% maximum potential), which is a considerable improvement from Config 2.
  - Skin temperature stays at 45°C, even with improvement in performance.
  - **Note:** This configuration is optimal for Egypt; however, additional use cases should be run to ensure the configuration can be applied generally.

Sensor Aliases

- Sensor aliases can be used in place of the specific TSENS sensor number in the sensor field of the thermal-engine.conf file. Below are the aliases by chipset.
- MSM8960
  - "tsens_tz_sensor0" = "cpu0"
  - "tsens_tz_sensor2" = "cpu1“
  - "tsens_tz_sensor3" = "pop_mem"
- MSM8930
  - "tsens_tz_sensor9" = "cpu0"
  - "tsens_tz_sensor6" = "cpu1"
  - "tsens_tz_sensor3" = "pop_mem"
- APQ8064
  - "tsens_tz_sensor7" = "cpu0"
  - "tsens_tz_sensor8" = "cpu1"
  - "tsens_tz_sensor9" = "cpu2"
  - "tsens_tz_sensor10" = "cpu3"
  - "tsens_tz_sensor6" = "pop_mem"
- MSM8974
  - "tsens_tz_sensor5" = "cpu0"
  - "tsens_tz_sensor6" = "cpu1"
  - "tsens_tz_sensor7" = "cpu2"
  - "tsens_tz_sensor8" = "cpu3“
  - "tsens_tz_sensor3" = "pop_mem"
- MSM8226
  - "tsens_tz_sensor5" = "cpu0-1"
  - "tsens_tz_sensor2" = "cpu2-3"
  - "tsens_tz_sensor3" = "pop_mem“
- MSM8926
  - "tsens_tz_sensor5" = "cpu0-1"
  - "tsens_tz_sensor1" = "cpu2-3"
  - "tsens_tz_sensor3" = "pop_mem"

Sensor Aliases (cont.)

- MSM8610
  - "tsens_tz_sensor5" = "cpu0-1-2-3"
  - "tsens_tz_sensor0" = "pop_mem“
- APQ8084
  - "tsens_tz_sensor5" = "cpu0"
  - "tsens_tz_sensor6" = "cpu1"
  - "tsens_tz_sensor7" = "cpu2"
  - "tsens_tz_sensor8" = "cpu3"
  - "tsens_tz_sensor3" = "pop_mem“
- MSM8916
  - "tsens_tz_sensor5" = "cpu0-1“
  - "tsens_tz_sensor4" = "cpu2-3“
  - "tsens_tz_sensor2" = "pop_mem"

Thermal Calibration for Speaker Protection

Thermal Calibration Procedure for Speaker Coil Protection

- Overview
  - The goal of thermal calibration for the speaker coil is to accurately estimate the temperature of the speaker coil from available temperature sensors in the system.
  - This can be achieved through a three-part process
    1. Derive equation to convert resistance to temperature
    2. Determine offset value for thermal management algorithm
    3. Program offset into device
  - This process is completed prefactory and should be redone when there is an electrical or mechanical design change to the device.
    - Required equipment
    - Thermal chamber
    - Multimeter with four-wire setup for accuracy
    - DUT with speaker disconnected from drive circuitry and the coil wired out to the meter for resistance measurement.
    - **Note:** The DUT should be fully reassembled prior to testing.

Thermal Calibration Procedure for Speaker Coil Protection (cont.)

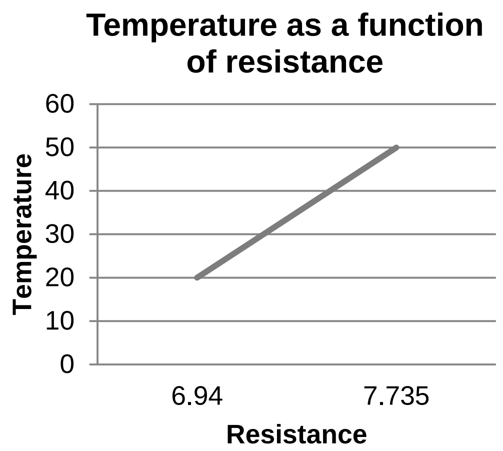
- Part 1 – Derive equation to convert resistance to temperature
  - Start by taking multiple resistance measurements manually, i.e., with multimeter at different temperatures. These measurements can be used to derive an equation (described on the next slide) that represents temperature as a function of resistance by taking advantage of the linear relationship between resistance and temperature.
  - Preconditions
    - DUT with speaker disconnected from drive circuitry and the coil wired out to the meter for resistance measurement
    - **Note:** DUT should be fully reassembled prior to testing.
    - Device is powered off
  - Procedure
    1. Set the thermal chamber to a known temperature, e.g., 20°C. The ambient temperature of the chamber should be monitored and verified by placing a thermocouple next to the device speaker.
    2. **Note:** The phone should be powered off during these steps.
    3. Place the phone in the chamber and allow it to soak for at least 30 min, allowing coil resistance to stabilize. Record coil resistance at 20°C.
    4. **Note:** Resistance variation should be less than 0.4% of nominal speaker resistance, e.g., for 8Ω, resistance variation should be < 0.03Ω, within 10 min.
    5. Repeat steps 1 and 2 for at least one other temperature, e.g., for 50°C for greater accuracy, the steps can be repeated a third time, e.g., 0°C.
    6. Resistance and temperature are assumed to have a linear relationship, so now any two points can be used to find the device's function for temperature. This function can then be used to find the temperature of the speaker coil by simply plugging in the value of the resistance (example shown on next slide).

Thermal Calibration Procedure for Speaker Coil Protection (cont.)

- Part 1 – Derive equation to convert resistance to temperature (cont.)
  - Here is an example of finding temperature as a function of resistance based on two sample data points, using the equation for a line $y = mx + b$.
  - Example data

Sample	Temp – T	Resistance – R
1	20°C	6.94
2	50°C	7.735

Graphed →



Procedure

1. Find the slope of line $m = \frac{T1-T2}{R1-R2}$, per the example, $m = \frac{20-50}{6.94-7.735}$ = 37.73585
2. Find the y-intercept $b = y - mx$, per the example, $b = 20 - 37.73585 * 6.94 =$ -241.887
3. Use the equation for a line to find the speaker coil temperature based on resistance $y = mx + b, where\ y = temp, and\ x = resistance,$ per the example, $y = 37.73585 * x - 241.887$

Note: Use linear regression to approximate a temperature function based on more than two data samples.

Thermal Calibration Procedure for Speaker Coil Protection (cont.)

- Part 2 – Determine offset value for thermal management algorithm
  - An equation to calculate temperature can be used to help determine the appropriate offset for the device. The following steps should be performed for at least three different temperatures and with the device running as many initial power-on scenarios as the manufacturer wishes. The most common temperatures, environments, and scenarios are the most important.
  - Preconditions
    - DUT with speaker disconnected from drive circuitry and the coil wired out to the meter for resistance measurement
    - **Note:** DUT should be fully reassembled prior to testing (same as Part 1)
    - Device software is installed
    - Device is powered on
  - Procedure
    1. Check for the file /data/misc/audio/audio.cal on the device. If this file exists, calibration already occurred. Remove the file and reboot the phone. Otherwise, proceed to step 2.
    2. Connect the USB debug command prompt to the dump logs. Start logging with the command adb logcat –s ThermalEngine | grep –i speaker.
    3. To obtain the necessary data for calibration, wait until the log displays min temperature (see the following logcat example). This takes 30 min or longer. The logs show a log sample count. When this count reaches 10, the speaker calibrates and the logs display the min temperature. **Note:** SpeakerCal temp is not used, because it has the default offset factored in.
    4. When this log is shown, use multimeter to measure the resistance of the speaker coil and calculate speaker temperature using the equation derived in Part 1. The difference between the logged min temperature and calculated temperature is the offset; e.g., if the measurement from the four-wire multimeter results in 27°C, the offset using the following logcat example is 27 to 32 = -5°C.

Thermal Calibration Procedure for Speaker Coil Protection (cont.)

- Part 2 – Determine offset value for thermal management algorithm
  - Procedure (cont.)
    5. To verify calibration success, check for existence of the file /data/misc/audio/audio.cal.
    6. Prepare the next temperature or scenario, remove the file /data/misc/audio/audio.cal, and reboot the phone to start the next thermal calibration (go to step 2).
  - After collecting offset data for the common ambient temperatures and scenario, an average of the offsets can be calculated and programmed into the phone.

```
01-10 04:47:19.159 E/ThermalEngine(  226): set_and_wait_alarm: Alarm Recieved 30000
01-10 04:47:19.159 E/ThermalEngine(  226): speaker_cal_algo: reading pm8941_tz 33168
mC
01-10 04:47:19.159 E/ThermalEngine(  226): speaker_cal_algo: reading[0] 33000 mC
01-10 04:47:19.159 E/ThermalEngine(  226): speaker_cal_algo: reading[1] 32000 mC
01-10 04:47:19.159 E/ThermalEngine(  226): speaker_cal_algo: reading[2] 33000 mC
01-10 04:47:19.159 E/ThermalEngine(  226): speaker_cal_algo: reading[3] 32000 mC
01-10 04:47:19.159 E/ThermalEngine(  226): speaker_cal_algo: reading[4] 32000 mC
01-10 04:47:19.159 E/ThermalEngine(  226): speaker_cal_algo: reading[5] 33000 mC
01-10 04:47:19.159 E/ThermalEngine(  226): speaker_cal_algo: reading[6] 33000 mC
01-10 04:47:19.159 E/ThermalEngine(  226): speaker_cal_algo: reading[7] 33000 mC
01-10 04:47:19.159 E/ThermalEngine(  226): speaker_cal_algo: min 32000 mC, max 33000
mC
01-10 04:47:19.159 I/ThermalEngine(  226): speaker_cal_algo: Speaker Cal temp 28C
01-10 04:47:19.159 E/ThermalEngine(  226): speaker_cal_algo: Success notifying spkr
clients
```

Thermal Calibration Procedure for Speaker Coil Protection (cont.)

- Part 3 – Program offset into device
  - The averaged offset can be used for the offset field of the thermal-engine.conf file, at /vendor/qcom/proprietary/thermal-engine/.
  - The speaker coil calibration instance is identified by the label SPEAKER-CAL (example below) in the thermal-engine.conf file.

```
[SPEAKER-CAL]
sampling     30000 30000 10 1800000
sensor       pm8941_tz
sensors      tsens_tz_sensor1 tsens_tz_sensor2 tsens_tz_sensor3
             tsens_tz_sensor4 tsens_tz_sensor7 tsens_tz_sensor8
             tsens_tz_sensor9 tsens_tz_sensor10
temp_range   6000 10000 2000
max_temp     45000
offset       -4000
```

  - If there is no preexisting instance of SPEAKER-CAL in the thermal-engine.conf file, the above example can be added and the offset value updated based on results from Part 2.
  - **Note:** The sensors line must be one contiguous line, without returns in the configuration file.
  - For field sampling, sensor, sensors, temp_range, and max_temp, see [R3] .

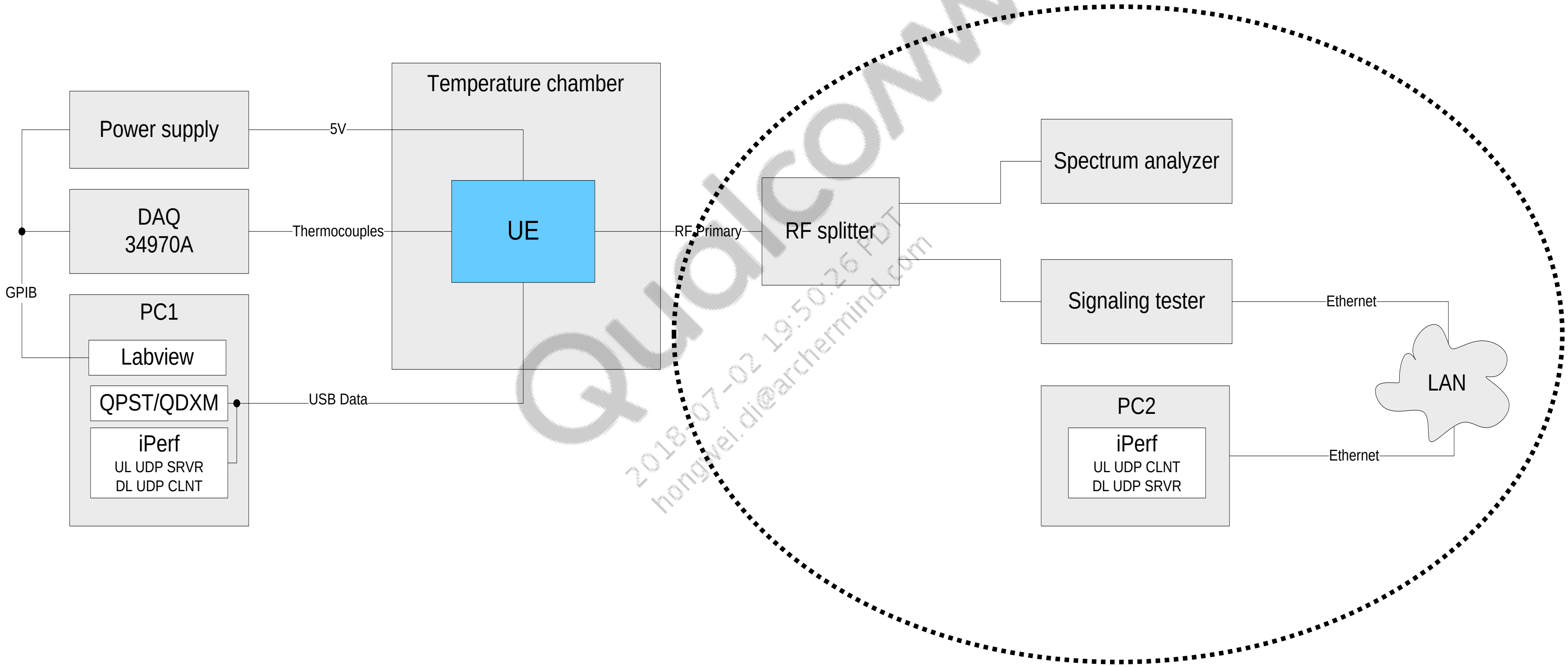
Appendix A – Thermal Test Setup

Thermal Lab Setup

1. Purchase thermal lab-related materials.
   - Data acquisition equipment as described below
   - Other required thermal equipment needed
2. Set up the environment.
   a. Place your device in a mechanical vice.
   b. Insert a micro SD card.
      - This is required to run the glBenchmark 2.5 Egypt HD in a fixed-frame loop.
   c. Disable USB charging; leave the USB plugged in.
   d. Set the screen timeout to NEVER.
   e. Set the display to maximum brightness.

Test Setup

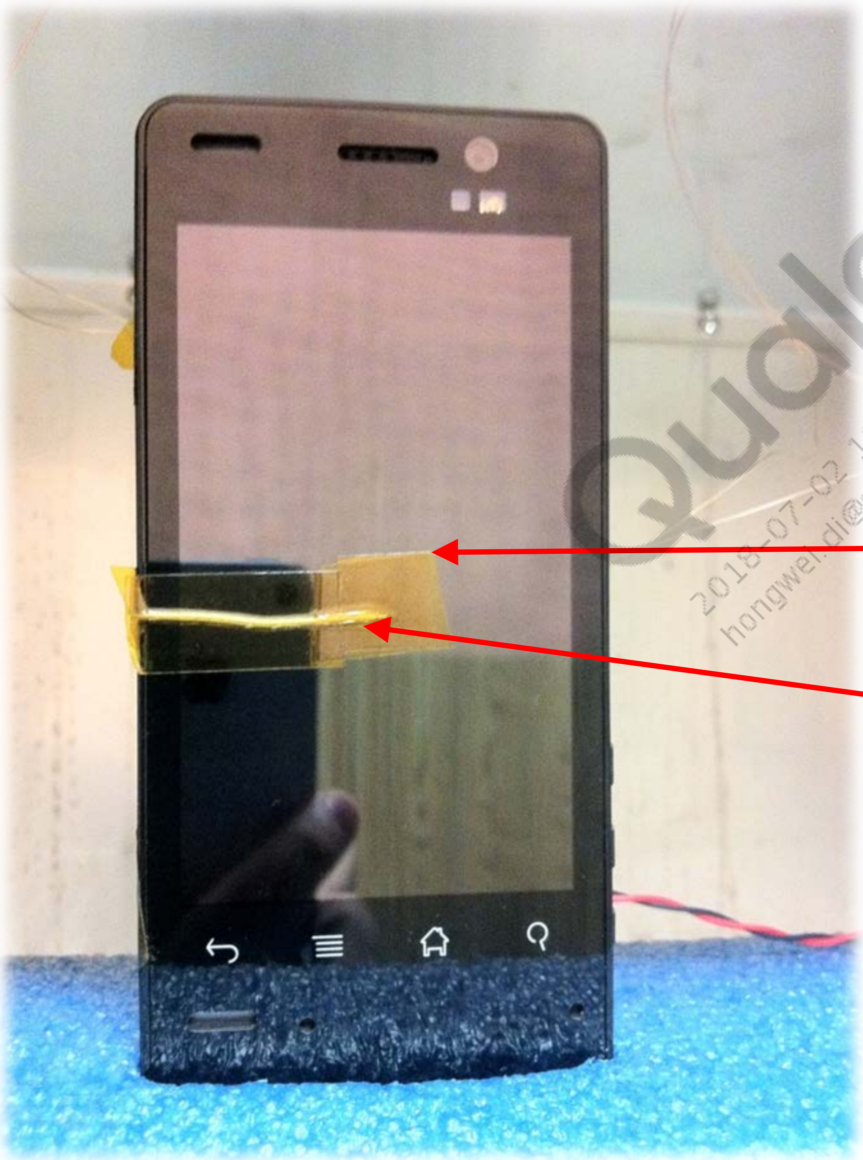
Required for modem-centric thermal use cases



Placement of Hotspot and Skin Thermocouples

1. Verify whether a thermocouple has a break/open before you begin.
   - Use a digital multimeter with a temperature setting. Connect each probe to each of the two dissimilar metals to verify functionality. The multimeter should read a temperature close to ambient (~25°C).
2. Untangle thermocouples and lay them out completely before you begin.
   - Create a label and attach it with Kapton tape along each thermocouple.
   - Place your device in a styrofoam pad or any holding structure with insulation between the chamber and DUT to secure it in an upright position (see slide 17).
3. Run a very CPU-intensive app, i.e., dual-Dhrystone, at maximum CPU frequency for a few minutes to identify hotspots on the LCD and back cover.
   - Use only QTI devices with open-loop AVS settings and FC software.
   - Continuously run for the period with no crashes/resets
4. With the app running, monitor the device skin with an IR camera and find the hotspots (hottest points) on the LCD panel and back cover.
   - Place the thermocouple tips on the exact hotspots on the LCD and back cover, respectively, one by one.
   - Place a few pieces of Kapton tape over each one. The tape should cover the length of the thermocouple.
5. Power down and allow the device to cool back down to ambient temperature.

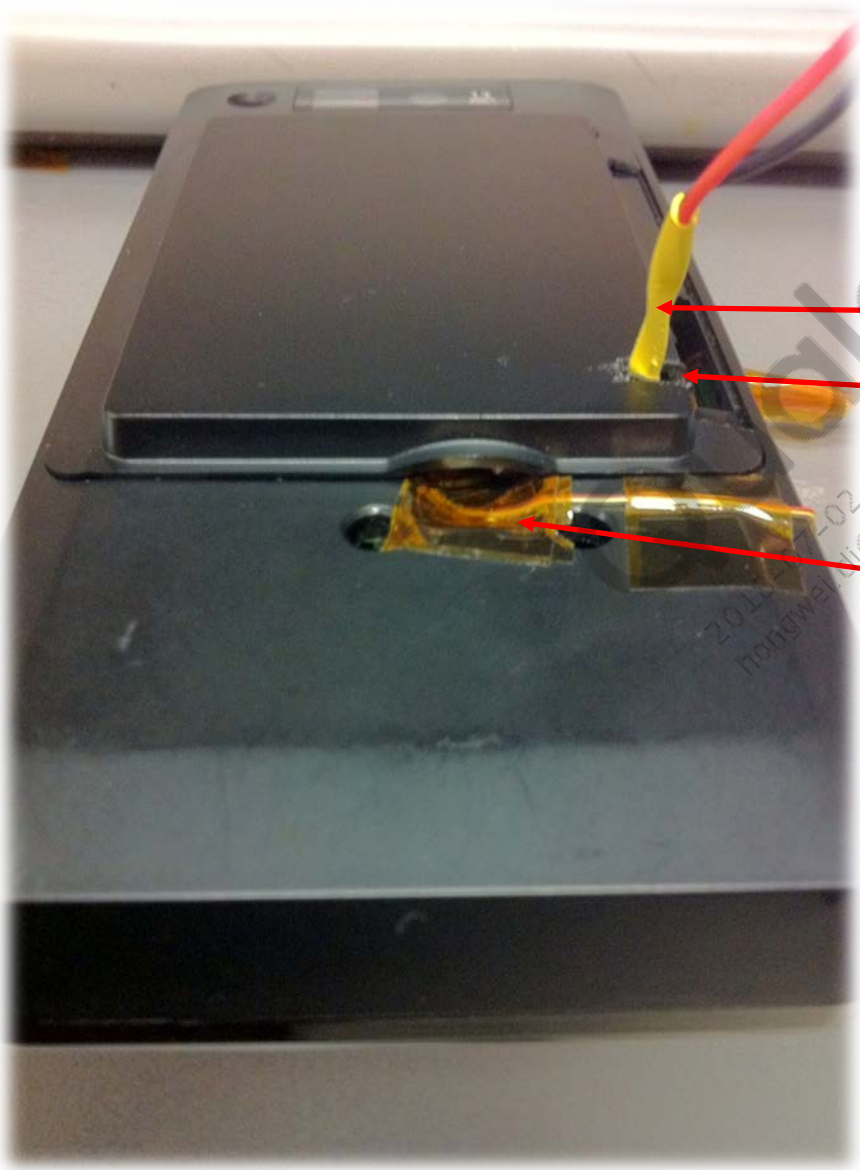
Correct LCD Thermocouple Placement



Layers of Kapton tape to hold thermocouples in place

Thermocouple tip

Correct Backside Thermocouple Placement



Fake battery leads

Hole cut into housing for fake battery leads

Back cover thermocouple with Kapton tape

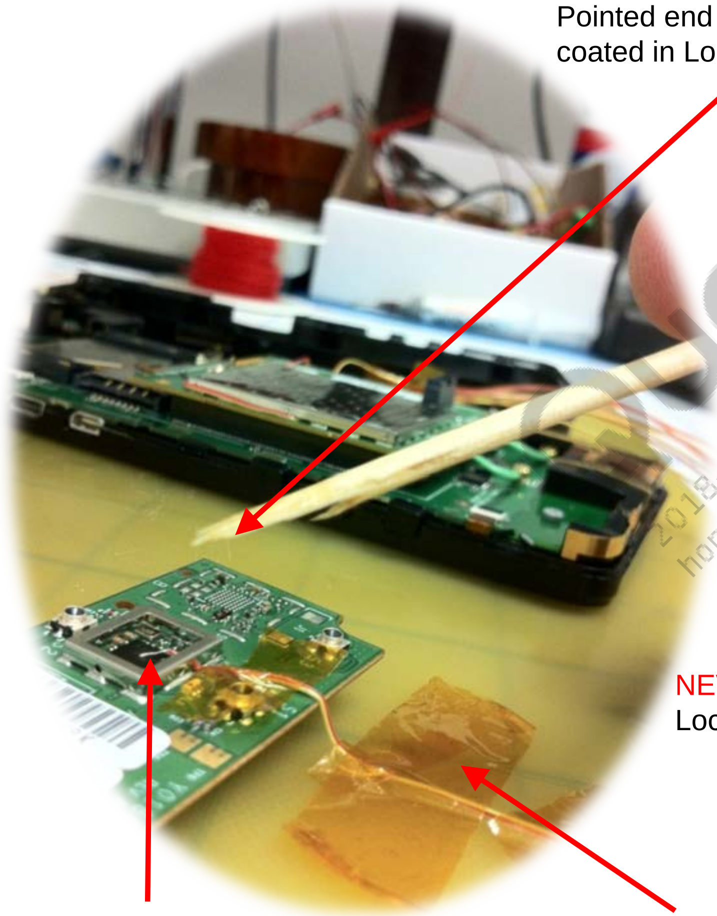
Placement of Internal Thermocouples

1. Completely disassemble the DUT. Remove the PCB from the housing assembly.
2. Plan thermocouple routing before you begin.
3. Using the IC package mechanical specs, find the exact middle of the device.
4. Place the thermocouple at that spot. It is very important that it is exactly in the middle and that it maintains complete contact with the component. Place each thermocouple flat against the board when you route, using the Kapton tape to securely hold it in place.
   - 36 AWG is thicker and more durable than 40 AWG, but it needs more space and may not be routed underneath some shielding.
   - Decide before placement; 40 AWG is fragile and if it is broken will completely ruin your work after it is bonded with Loctite 444.
5. With tweezers, grab the thermocouple shielding at the junction.
6. Make a crimp on the junction so it is no longer straight across but curved slightly downward.
   - It must be pointing downward so that when you place the Kapton tape around it, the thermocouple touches the IC package, i.e., PoP memory, on its own. Make sure the tip is in the exact middle and touching the component (see the next slide).
7. Use a microscope to verify.
8. Route cables so they make as little contact with other components as possible. Permanent mechanical rework may be necessary (cutting shielding bars or back cover of housing).

Note: The body of the thermocouple can be flattened after it is secured to the component (glue must be cured using the accelerant). The tip should *not* move.

Placement of Internal Thermocouples (cont.)

GOOD

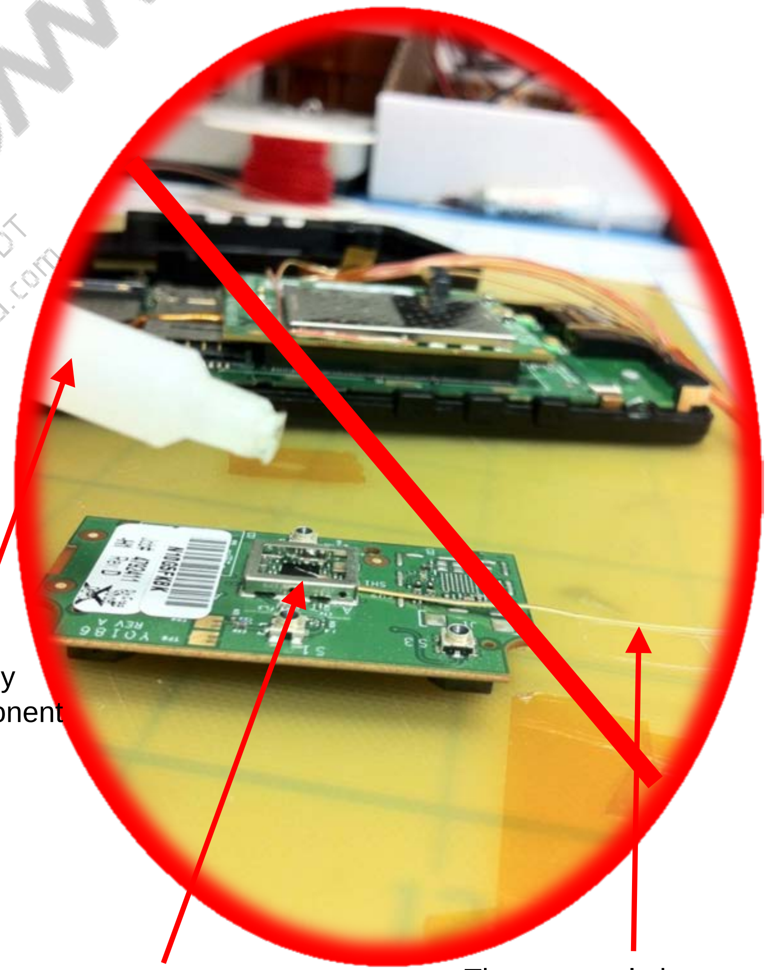


Pointed end of swab coated in Loctite 444

Junction bent downward to make complete contact with component

Kapton tape secures thermocouple

BAD



NEVER directly apply Loctite 444 to component

Junction is up in the air and straight across, not bent downward

Thermocouple is not secured; very loose

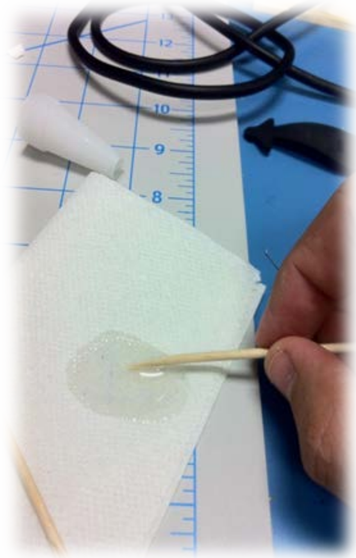
Placement of Internal Thermocouples (cont.)

9. After the thermocouple is secured with Kapton tape and makes complete contact with the exact middle of the component package:
   a. Squeeze Loctite 444 onto a clean piece of paper.
   b. Snap a wooden cotton swab in half, take the pointed end and coat by rolling its edge in Loctite 444 (see next slide).
   c. Use a microscope for the rest of the procedure.
   d. Place the smallest possible amount on the tip and the area where it touches the component. It should look like a small bubble under a microscope.

      This is extremely important. There must only be enough to hold the thermocouple tip rigidly in place, even if the device must be constantly moved. After applying Loctite 444, there must be ***NO space*** between the thermocouple tip and the package

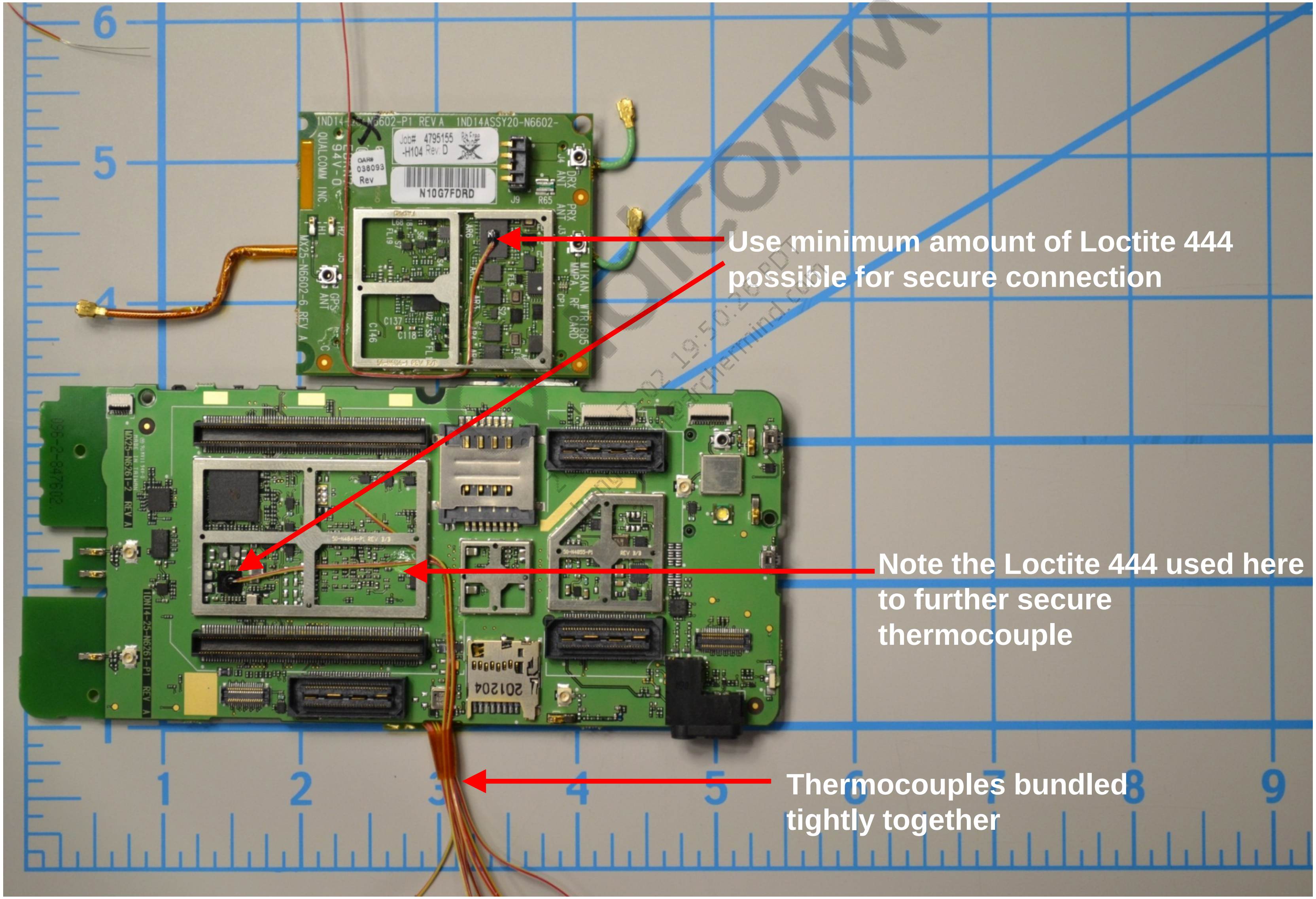
      If it must be reworked, scrape off the thermocouple, replace, and find another location. This may not be possible to remove, depending on the bond. It is best to start off with a new thermocouple. The prior location is ruined and ***cannot be used again*** for thermocouple placement. Loctite cannot be removed without destroying the PCB.
   e. Immediately open the accelerator fluid and very slightly touch the Loctite 444 on the thermocouple/ component with the brush. This quickly accelerates the curing process.
   f. Carefully remove the Kapton tape.
   g. Make sure the thermocouple junction that is not covered by glue does not come into contact with metal, i.e., the PCB shield, before reassembly.
   h. If needed, place the thermocouple against the board and add Loctite 444 so it is secured to the PCB. This strengthens the overall bond of the thermocouple (see the next slides).
   i. Place thermocouples in as many other locations as desired.
   j. If more than one thermocouple is used internally, route all to the same exit point *tightly* as close as possible and secure the outer bundle with the Kapton tape rolled around it, zip-tie, or both.

Thermocouple Materials

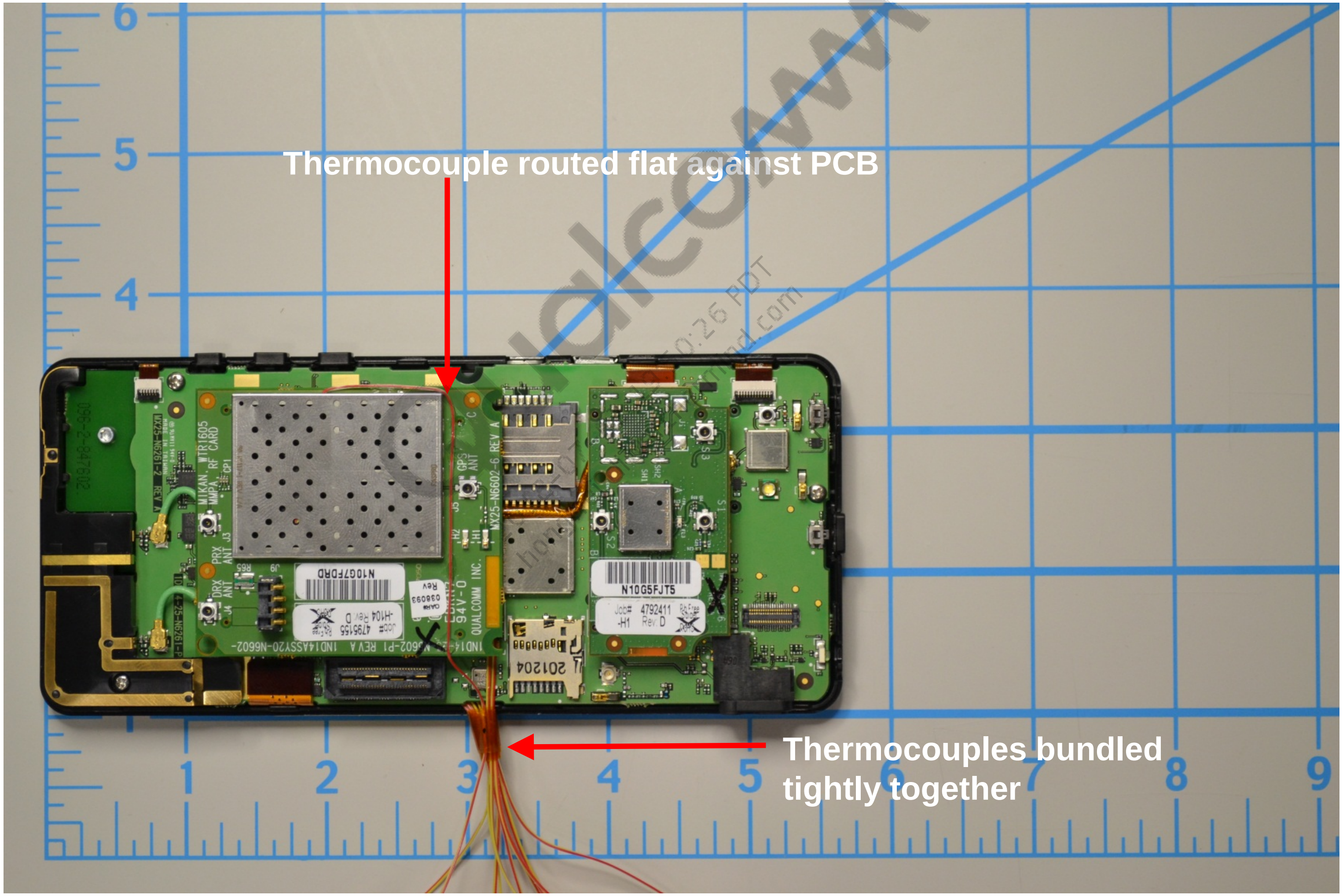


Omega K-Type 36AWG 30-inch thermocouples (pack of 5)

Internal Thermocouple Attachment



Thermocouple Routing



Reassembly and Data Logger

1. Reassemble PCB with the housing assembly and secure all connections
   - It may be necessary to cut a hole into the bundle exit point on the front or back housing. Do not snap the assembly together on the thermocouple; it may break and destroy your work.
   - Do not yet bundle the rest of the thermocouples together.
   - Determine the order for the thermocouples to be placed in the data logger. Note this order of placement.

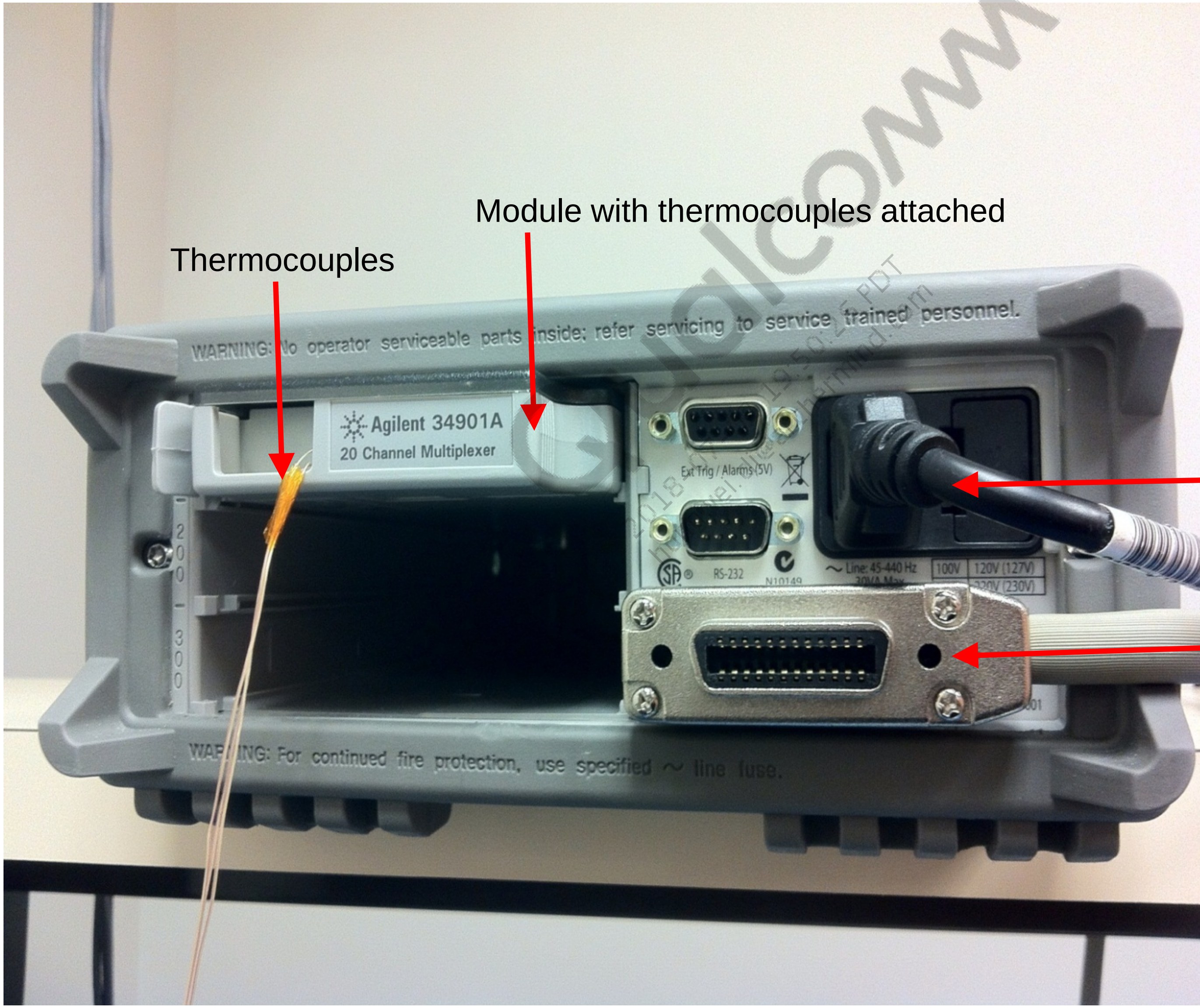
   Note: Also see the *Getting Started Guide* from Agilent.

2. Open the data logger 20-channel multiplexer cover with a screwdriver or pen, pushing the latch forward. Be careful or the module may need to be replaced.
3. With the correct mini-flathead screwdriver, unscrew the applicable screws in the Agilent multiplexer module.
   - Below each screw is an opening to which the metal end of the thermocouple is placed. Each opening contains a piece of metal that secures the thermocouple in place.
   - Unscrew until this piece of metal goes completely down.
   - Each channel in the multiplexer is labeled with an H (HI) and an L (LO). The yellow end of the thermocouple is HI and red is LO.
4. It is easier to insert both ends at once, holding the body of the thermocouple with one hand and using a mini screwdriver until the metal piece goes up and secures the thermocouple in place.
5. Do the same for the other lead while still holding the body with one hand.
6. Repeat for all thermocouples.

Verification and Common Setup Errors

- Verify with a digital multimeter
  - Place the hot and ground leads separately, with one for HI and the other for LO, for each channel. Polarity is not considered, i.e., the positive lead contacts the screw at CH01 HI and the ground lead contacts the screw at CH01 LO.
  - Only place multimeter leads at the screws, *not* at the thermocouples themselves.
  - You are only verifying that you see a temperature reading.
  - Check that the yellow lead is the first in the channel.
  - If there is an OPEN reading, the thermocouple HI and LO may not be in the same channel. Extract and correct.
  - It also may indicate a break in the thermocouple, which ruins your work and must be replaced.
  - If temperature readings on the data logger jump, it may indicate a loose thermocouple. Extract and correct.

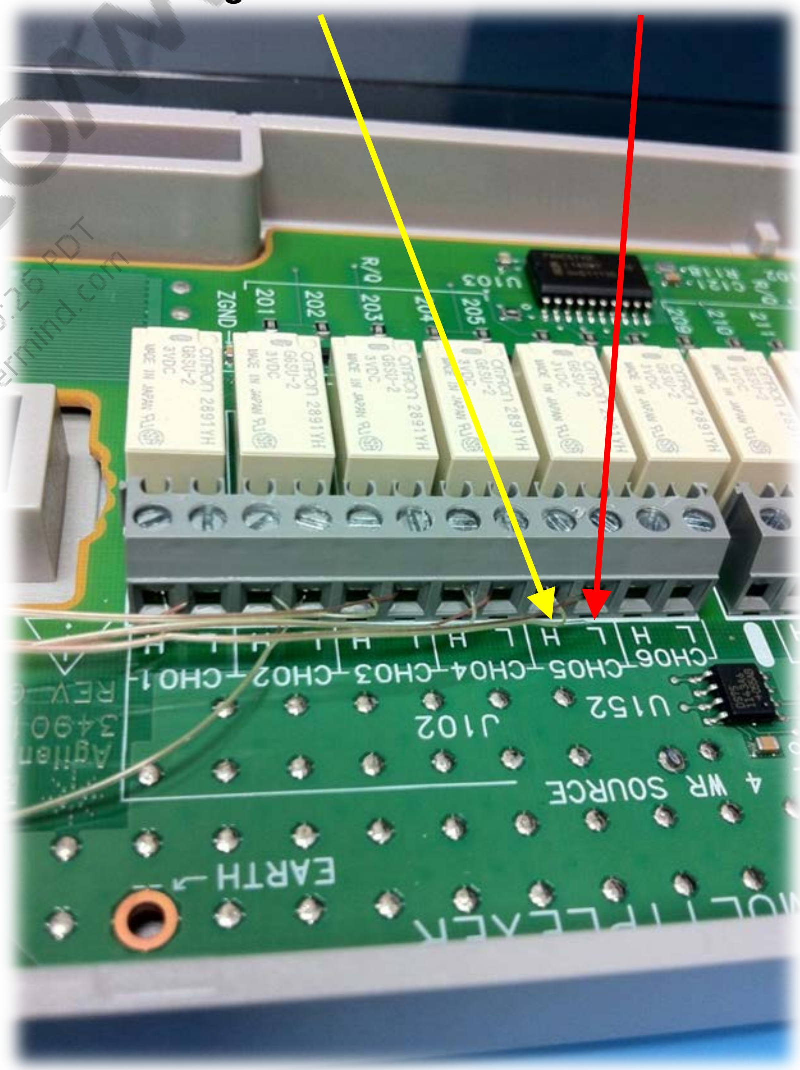
Agilent 34970A Data Logger with Agilent 34901A 20-Channel Multiplexer Module



Agilent 34901A 20-Channel Multiplexer Modules (Data Logger)

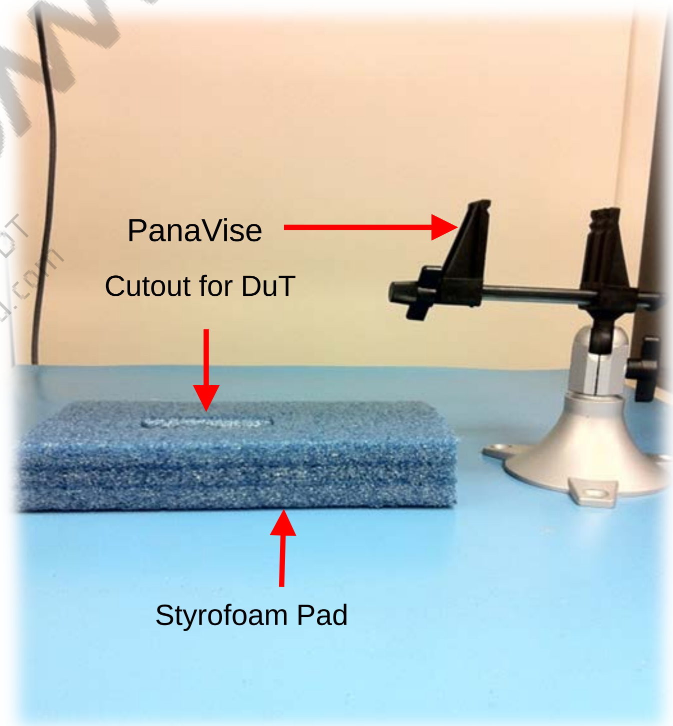
Yellow or light lead = H

Red or dark lead = L

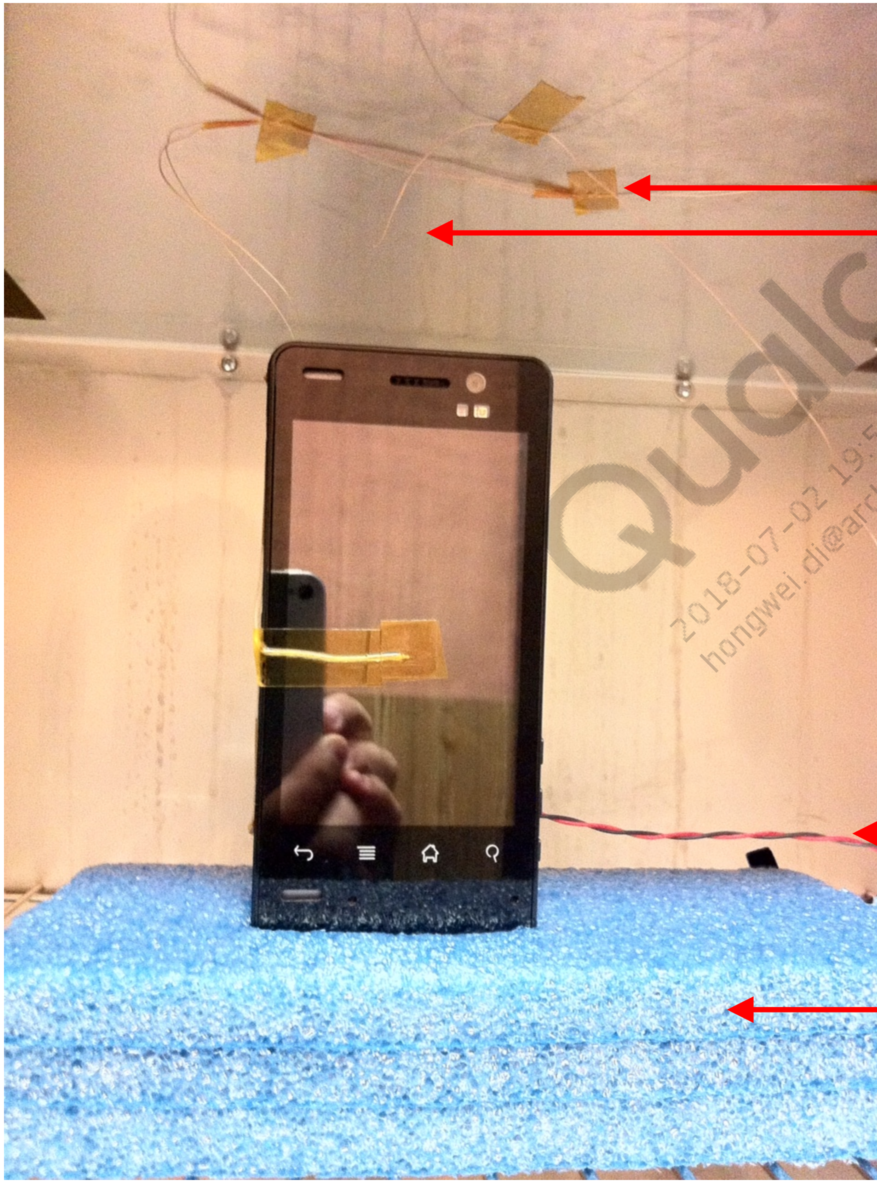


Temperature Chamber Setup

1. Use a large temperature chamber that has the ceiling clearance to hold the PanaVise (a holding/clamping tool) and DUT, with ambient thermocouple hanging down 1 inch above the device. If not, use a styrofoam block and cut out an area of the styrofoam so the device can be placed to secure it upright (it also works if you want to rest it in landscape orientation).
2. Place the fake battery, the device with thermocouples, and the USB cable inside the chamber.
3. Turn on the temperature chamber and set the temperature to 25°C.
4. Turn off the air flow before testing.



Temperature Chamber Setup (cont.)



Thermocouples taped to ceiling

Thermocouple measures ambient temp (~1 inch away from DUT and possibly place it on one side of DUT)

Fake battery leads from DUT to power supply

Styrofoam pad or holding structure with insulation between the chamber and DUT

Data Logger App for Agilent 34970A Data Logger

1. Insert as many multiplexer modules as you want to have available into the data logger (maximum of three, minimum of one).
2. Click **Configuration→New**. Label this configuration.
3. Select **Application Mode→Connected to Instrument**.
4. Select **Add or Remove Instruments→Add Instruments**.
   - Click **Find Instruments**.
   - A search begins for the module. After the modules are located correctly, check the box and click **Enter**.
5. A 34901A: 20-Channel Armature Multiplexer appears below.
6. The top module compartment is 100, middle is 200, and bottom is 300 on the data logger.
7. Check the boxes under the Scan column for the number of thermocouples you wish to monitor.
8. The order of thermocouples must correspond to the exact order in the module, e.g., 101 – LCD, channel 01 in the module must have the HI and LO leads of the thermocouple connected to the LCD.
9. Select the space corresponding to the desired channel under the Name column and name the thermocouple.

Note: Some of this information is available in [R3].

Data Logger Application for Agilent 34970A Data Logger

10. In the Function column, select **Temp (Type K)** for Type-K thermocouples. Verify that C (Celsius) appears in the Res column.
11. At least four thermocouples may be needed; one for LCD, back cover, internal thermocouple on PoP memory, and ambient.
12. The Scan and Log Data tab should now be populated with the information provided in step 11.
13. In the same tab, under the Scan Control heading, click **…** under the Set column.
14. In this dialog box, select **Immediately** and ensure Interval (Time Between Scans) under SS: is 1. Select **By Pressing Stop Scan** for the Stop Scanning option. Click **Apply to all instruments** at the bottom.

Note: Some of this information is available in [R3].

Appendix B – Thermal Lab Equipment (Extended List)

Thermal Lab Setup

Assembly/Disassembly/Rework Bench						
Equipment/tool	**Description**	**Manufacturer**	**Manufacturer model/part #**	**Vendor**	**Vendor part #**	**QTI contact**
Technician toolbox	**Total of 36 tools included**	SMS	1001-Q	Solder Master Supply	1001-Q	Test Equipment
Adjustable wrench	6" adjustable wrench with red vinyl grip, 15/16" capacity	Crescent	AC16C	Stanley Supply & Services	402-051	Test Equipment
Adjustable wrench	4" adjustable wrench	Iron Bull	NA	Solder Master Supply	NA	Test Equipment
Plier/cutter	4-1/2" transverse-end cutter pliers	Xcelite	EC54-J	Stanley Supply & Services	190-142	Test Equipment
Plier	Electronic pliers, round Jaw 4.5"	Xcelite	RN54	Stanley Supply & Services	114-809	Test Equipment
Plier	Combination slip joint plier 6"	Crescent	H26C	Stanley Supply & Services	403-684	Test Equipment
Plier	Groove joint plier 7"	Crescent	R27C	Stanley Supply & Services	114-808	Test Equipment
Wire stripper	T Stripper, 16 to 26 AWG, stranded wire	Ideal	45-121	Stanley Supply & Services	118-568	Test Equipment
Cutters	Angled cutter	Erem	2475E	Stanley Supply & Services	447-800	Test Equipment
Pick	Angle stainless steel probe, 5-1/2" long	Menda	35617	Stanley Supply & Services	151-166	Test Equipment
Forceps	Straight nose 5.5" seizer	Xcelite	42HV	Stanley Supply & Services	151-017	Test Equipment
Tape measure	1.5" x 10' measuring tape	Lufkin	L610	Solder Master Supply	NA	Test Equipment
Screwdriver	Regular Phillips screwdriver, #1 tip, 3" blade, 6-5/8" overall	Xcelite	X101	Stanley Supply & Services	115-589	Test Equipment
Screwdriver	Regular Phillips screwdriver, #2 tip, 4" blade, 8-1/8" overall	Xcelite	X102	Stanley Supply & Services	115-591	Test Equipment
Screwdriver	Slotted screwdriver, regular style, 1/8" x 2"	Xcelite	R182	Stanley Supply & Services	115-525	Test Equipment
Screwdriver	Regular Phillips screwdriver, #0 tip, 2" blade, 4-1/2" overall	Xcelite	X100	Stanley Supply & Services	115-588	Test Equipment
Screwdriver	Regular slotted, 1/4" tip, 4" blade, 8-1/8" overall	Xcelite	R144	Stanley Supply & Services	115-520	Test Equipment
Screwdriver	Regular slotted, 3/16" tip, 6" blade, 9 1/2" overall	Xcelite	R3166	Stanley Supply & Services	115-533	Test Equipment
Scissors	4 1/8" embroidery scissors	Cozic	KHS-105	Solder Master Supply	NA	Test Equipment
Tweezer	Tweezer, style 00, 4-3/4" long	CHP	00-SA	Solder Master Supply	NA	Test Equipment
Tweezer	Tweezer, style AA	CHP	AA-SA	Solder Master Supply	NA	Test Equipment

Thermal Lab Setup (cont.)

Assembly/Disassembly/Rework Bench (cont.)						
Equipment/tool	**Description**	**Manufacturer**	**Manufacturer model/part #**	**Vendor**	**Vendor part #**	**QTI contact**
Technician toolbox	**Total of 36 tools included**	SMS	1001-Q	Solder Master Supply	1001-Q	Test Equipment
Wire unwrapping tool	24-32 AWG/counter clock-wise direction	JDV Products	HU93	Solder Master Supply	NA	Test Equipment
PinVise	Double-end PinVise	Euro Tool	PIN219.00	Stanley Supply & Services	125-362	Test Equipment
Drill set	61-80 - .0390-.0135 drill set	Euro Tool	DRL-240.00	Solder Master Supply	N/A	Test Equipment
Chip puller	Static-dissipative PLCC extractor	C.K.	2371	Stanley Supply & Services	126-453	Test Equipment
Cutters	Slim tapered-head diagonal cutter	Swanstrom Tools	420-Jensen	Stanley Supply & Services	419-318	Test Equipment
Cutters	Miniature diagonal semi-flush electronic cutter, round nose	Xcelite	MS54	Stanley Supply & Services	115-074	Test Equipment
Wire stripper	No Nik wire stripper 32 AWG	No Nik	NN012	Stanley Supply & Services	4-303	Test Equipment
Desoldering pump	Static-free desoldering pump with aluminum barrel	Edsyn	SS350	Stanley Supply & Services	114-412	Test Equipment
Screwdriver set	6-pc miniature screwdriver set	Euro Tool	SCR-900.00	Solder Master Supply	NA	Test Equipment
Midget wrench set	10-pc midget combination wrench set (5/32-7/16")	Armstrong	25-600	Solder Master Supply	NA	Test Equipment
Precision knife	Precision knife and 5 blades	X-Acto	X3001	Stanley Supply & Services	119-336	Test Equipment
Steel ruler	6" precision rule with conversions (standard/metric/english)	Kristeel	401 A 5	Solder Master Supply	NA	Test Equipment
Needle file set	12-pc mini file set	Euro Tool	FIL-990.00	Stanley Supply & Services	401-442	Test Equipment
Long-nose locking pliers	6" long nose-locking pliers with wire cutter	Crescent	C6NV	Stanley Supply & Services	424-525	Test Equipment
Hex wrench set	L-wrench hex set 12-pc 050-5/16	Bondhus	12136	Stanley Supply & Services	174-435	Test Equipment
Needle-nose pliers	Electronic pliers, long nose, serrated	Xcelite	LN55	Stanley Supply & Services	114-781	Test Equipment

Thermal Lab Setup (cont.)

Equipment/tool	Description	Manufacturer	Manufacturer model/part #	Vendor	Vendor part #	QTI contact
Tools (not included with toolbox)						
Miniature torx set	6-piece miniature torx screwdriver set	Wiha	26790	Stanley Supply & Services	115-218	Test Equipment
Pen vac	Pen vac kit with four tips	Excelta	PV-HV	Stanley Supply & Services	435-541	ESOS
Cutting mat with scale	X-Acto 18“ x 24" self-healing cutting mat with 1" scale grid	X-Acto	X7762	Stanley Supply & Services	403-392	ESOS
Thermal test equipment						
Data module	20-channel multiplexer (2/4-wire) module for 34972A	Agilent	34901A	QTI		Test Equipment
Data logger	Data cquisition/data logger switch unit	Agilent	34972A	QTI		Test Equipment
Consumables and materials						
Pana vise 201 JR.	Vise head rotates a full 360° and pivots 210°	PanaVise	201	Stanley Supply & Services	400-231	Test Equipment
Thermocouples	36 gauge K-type 6' thermocouples	Omega	5TC-TT-K-36-72	Test Equipment Supply Supply WT-371		ESOS
Swabs	6" cotton-tipped applicators (Qty 100)	—	—	Test Equipment Supply Supply WT-371	NA	Test Equipment
Epoxy adhesive	50/50 epoxy/hardener (prepackaged) adhesive	AngstromBond	AB9226	Test Equipment Supply Supply WT-371	NA	Test Equipment
Instant adhesive	Cyanoacrylate (Loctite 444)	Loctite	12292	Test Equipment Supply Supply WT-371	NA	Test Equipment
Instant adhesive accelerator	Tak-pak accelerator 7452 (used with Loctite 444)	Loctite	18490	Test Equipment Supply Supply WT-371	NA	Test Equipment
Kapton tape	Kapton tape 1/2"	3M	5419	Test Equipment Supply Supply WT-371	NA	Test Equipment
Kapton tape	Kapton tape 1/4"	3M	5419	Test Equipment Supply Supply WT-371	NA	Test Equipment
Kimwipes	Lint-free cleaning wipes 4.4" x 8.4" (box contents 280 wipes)	Kimberly-Clark	NA	Test Equipment Supply Supply WT-371	NA	Test Equipment
Cleaning brushes	For cleaning at the PCB/PCA level (custom cut by KG)	NA	NA	Test Equipment Supply Supply WT-371	NA	Test Equipment
Wooden applicators	For applying liquid adhesives or support wire while adhesive cures	NA	NA	Test Equipment Supply Supply WT-371	NA	Test Equipment
Replacement x-Acto blades	Replacement X-Acto blades 100/pkg	X-Acto	X611	Stanley Supply & Services	119-362	Test Equipment
DMM	True-rms industrial logging multimeter with trendcapture	Fluke	Fluke 289	Stanley Supply & Services	444-379	Test Equipment

References

References

Ref.	Document	
Qualcomm Technologies		
Q1	*Application Note: Software Glossary for Customers*	CL93-V3077-1
Q2	*Thermal Design Checklist*	80-VU794-21
Q3	*Design for Thermal: Key Requirements Why, What, Where, When*	80-VU794-24
Q4	*Thermal Management of MSM8660/MSM8260/APQ8060 Devices*	80-VU872-16
Q5	*Thermal Protection Algorithm Overview*	80-VT344-1
Q6	*Application Note: MDM8200 Thermal Protection Algorithm*	80-VJ372-14
Q7	*Application Note: MDM9600 Thermal Protection Algorithm Details*	80-VP146-15
Q8	*Application Note: MDM9200 Thermal Protection Algorithm Details*	80-VP145-15
Q9	*Application Note: MDM8220 Thermal Protection Algorithm Details*	80-VP144-15
Q10	*MSM8960 Thermal Mitigation Algorithm*	80-N8633-1
Q11	*MDM9x15 Thermal Mitigation Algorithm*	80-N8633-2
Q12	*APQ8064+MDM9615 Thermal Management Algorithm*	80-N8633-3
Q13	*MSM8974 Thermal Management Algorithm*	80-N8633-6
Q14	*Skin Temperature Measurement Procedure Using IR Camera*	80-VU794-15

References (cont.)

Ref.	Document	
References		
R1	*DAQ*	http://www.home.agilent.com/en/pd-1756491-pn-34972A/lxi-data-acquisition-data-logger-switch-unit?cc=US&lc=eng
R2	*Free Software*	http://www.home.agilent.com/agilent/software.jspx?cc=US&lc=eng&ckey=778242&nid=-33257.922596&id=778242
R3	*Agilent Data Logger 3 Getting Started Guide*	http://www.home.agilent.com/agilent/software.jspx?id=778242&cc=US&lc=eng

Questions?

https://support.cdmatech.com