Measuring Power Values

Device manufacturers must provide a component power profile in `/frameworks/base/core/res/res/xml/power_profile.xml`.

To determine values for power profiles, use hardware that measures the power being used by the device and perform the various operations for which information is needed. Measure the power use during those operations and compute the values (deriving differences from other baseline power uses as appropriate).

As the goal of a power profile is to estimate battery drain appropriately, power profile values are given in current (amps). The Android framework multiplies the current by the time for which the subsystem was active and computes the mAh value, which is then used to estimate the amount of battery drained by the application/subsystem.

Devices with Bluetooth, modem, and Wi-Fi controllers running Android 7.0 and higher can provide additional power values obtained from chipset data.

Devices with heterogeneous CPUs

The power profile for devices with CPU cores of heterogeneous architecture must include the following additional fields:

- Number of total CPUs for each cluster (expressed in cpu.clusters.cores).
- CPU speeds supported by each cluster.
- Active CPU power consumption for each cluster.

To differentiate between active CPU power consumption and supported CPU speeds for clusters, append the cluster number to the name of the array. Cluster numbers are assigned in the order of CPU cores in the kernel device tree. For example, in a heterogeneous architecture that has two (2) clusters with four (4) cores:

- cluster0 consists of cpu0-3
- cluster1 consists of cpu4-7

The Android framework uses these CPU core numbers when it reads statistics from the `sysfs` files in: `/sys/devices/system/cpu/cpu<number>/cpufreq/stats`.

Example of cluster CPUs and speeds:

```
<array name="cpu.active.cluster0">
<value>200</value>
<value>300</value>
<value>400</value>
</array>
<array name="cpu.speeds.cluster0">
<value>600000</value>
<value>800000</value>
<value>1200000</value>
</array>

<array name="cpu.active.cluster1">
<value>400</value>
<value>500</value>
<value>600</value>
</array>
<array name="cpu.speeds.cluster1">
<value>800000</value>
<value>1200000</value>
<value>1400000</value>
</array>
```

Power values

The following table describes available power value settings. To view the sample file in AOSP, see power_profile.xml (https://android.googlesource.com/platform/frameworks/base/+/master/core/res/res/xml/power_profile.xml).

Name	Description	Example Value	Notes
none	Nothing	0	
screen.on	Additional power used when screen is turned on at minimum brightness.	200mA	Includes touch controller and display backlight. At 0 brightness, not the Android minimum which tends to be 10 or 20%.
screen.full	Additional power used when screen is at maximum brightness, compared to screen at minimum brightness.	100mA-300mA	A fraction of this value (based on screen brightness) is added to the screen.on value to compute the power usage of the screen.
wifi.on	Additional power used when Wi-Fi is turned on but not receiving, transmitting, or scanning.	2mA	
wifi.active	Additional power used when transmitting or receiving over Wi-Fi.	31mA	
wifi.scan	Additional power used when Wi-Fi is scanning for access points.	100mA	
dsp.audio	Additional power used when audio decoding/encoding via DSP.	14.1mA	Reserved for future use.
dsp.video	Additional power used when video decoding via DSP.	54mA	Reserved for future use.
camera.avg	Average power use by the camera subsystem for a typical camera application.	600mA	Intended as a rough estimate for an application running a preview and capturing approximately 10 full-resolution pictures per minute.
camera.flashlight	Average power used by the camera flash module when on.	200mA	
gps.on	Additional power used when GPS is acquiring a signal.	50mA	
radio.active	Additional power used when cellular radio is transmitting/receiving.	100mA-300mA	
radio.scanning	Additional power used when cellular radio is paging the tower.	1.2mA	
radio.on	Additional power used when the cellular radio is on. Multi-value entry, one per signal strength (no signal, weak, moderate, strong).	1.2mA	Some radios boost power when they search for a cell tower and do not detect a signal. Values can be the same or decrease with increasing signal strength. If you provide only one value, the same value is used for all strengths. If you provide two values, the first is used for no-signal, the second value is used for all other strengths, and so on.
bluetooth.controller.idle	Average current draw (mA) of the Bluetooth controller when idle.	-	These values are not estimated, but taken from the data sheet of the controller. If there are multiple receive or transmit states, the average of those states is taken. In addition, the system now collects data for Low Energy (LE) and Bluetooth scans (#le-bt-scans). Android N and later no longer use the Bluetooth power values for bluetooth.active (used when playing audio via Bluetooth A2DP) and bluetooth.on (used when Bluetooth is on but idle).
bluetooth.controller.rx	Average current draw (mA) of the Bluetooth controller when receiving.	-	
bluetooth.controller.tx	Average current draw (mA) of the Bluetooth controller when transmitting.	-	
bluetooth.controller.voltage	Average operating voltage (mV) of the Bluetooth controller.	-	
modem.controller.idle	Average current draw (mA) of the modem controller when idle.	-	These values are not estimated, but taken from the data sheet of the controller. If there are multiple receive or transmit states, the average of those states is taken.
modem.controller.rx	Average current draw (mA) of the modem controller when receiving.	-	
modem.controller.tx	Average current draw (mA) of the modem controller when transmitting.	-	
modem.controller.voltage	Average operating voltage (mV) of the modem controller.	-	

wifi.controller.idle	Average current draw (mA) of the Wi-Fi controller when idle.	-	These values are not estimated, but taken from the data sheet of the controller. If there are multiple receive or transmit states, the average of those states is taken.
wifi.controller.rx	Average current draw (mA) of the Wi-Fi controller when receiving.	-	
wifi.controller.tx	Average current draw (mA) of the Wi-Fi controller when transmitting.	-	
wifi.controller.voltage	Average operating voltage (mV) of the Wi-Fi controller.	-	
cpu.speeds	Multi-value entry that lists each possible CPU speed in KHz.	125000KHz, 250000KHz, 500000KHz, 1000000KHz, 1500000KHz	The number and order of entries must correspond to the mA entries in cpu.active.
cpu.idle	Total power drawn by the system when CPUs (and the SoC) are in system suspend state.	3mA	
cpu.awake	Additional power used when CPUs are in scheduling idle state (kernel idle loop); system is not in system suspend state.	50mA	Your platform might have more than one idle state in use with differing levels of power consumption; choose a representative idle state for longer periods of scheduler idle (several milliseconds). Examine the power graph on your measurement equipment and choose samples where the CPU is at its lowest consumption, discarding higher samples where the CPU exited idle.
cpu.active	Additional power used by CPUs when running at different speeds.	100mA, 120mA, 140mA, 160mA, 200mA	Value represents the power used by the CPU rails when running at different speeds. Set the max speed in the kernel to each of the allowed speeds and peg the CPU at that speed. The number and order of entries correspond to the number and order of entries in cpu.speeds.
cpu.clusters.cores	Number of cores each CPU cluster contains.	4, 2	Required only for devices with heterogeneous CPU architectures (#multiple-cpus). Number of entries and order should match the number of cluster entries for the cpu.active and cpu.speeds. The first entry represents the number of CPU cores in cluster0, the second entry represents the number of CPU cores in cluster1, and so on.
battery.capacity	Total battery capacity in mAh.	3000mAh	

Low Energy (LE) and Bluetooth scans

For devices running Android 7.0, the system collects data for Low Energy (LE) scans and Bluetooth network traffic (such as RFCOMM and L2CAP) and associates these activities with the initiating application. Bluetooth scans are associated with the application that initiated the scan, but batch scans are not (and are instead associated with the Bluetooth application). For an application scanning for N milliseconds, the cost of the scan is N milliseconds of rx time and N milliseconds of tx time; all leftover controller time is assigned to network traffic or the Bluetooth application.