The Systems Hacker's Guide to the Galaxy
Energy Usage in a Modern Smartphone

Aaron Carroll and Gernot Heiser
NICTA and University of New South Wales
{aaron.carroll,gernot}@nicta.com.au

Abstract

Modern smartphones are increasingly performant and feature-rich but remain highly power-constrained. Effective energy management demands an informed policy, and to that end we present a detailed power analysis of the Samsung Galaxy S III smartphone. We measure power consumption by instrumentation at the circuit level, and present a breakdown by the major components including CPU, RAM, display, GPU, wireless radios, camera, GPS and environmental sensors.

1 Introduction

In 2010, we published a power analysis of the Openmoko FreeRunner mobile phone [2]. While perhaps not a typical example of the mass-market smartphone, this device had the advantage of documentation and hardware support that made a fine-grained power analysis reasonably straightforward. We claimed that such an analysis necessarily required both access to circuit schematics, and hardware support. To overcome the limitation of an atypical device, we compared the measurements with full-system power data from two additional devices which are considered more representative of the commodity smartphone.

We now show that module-level power instrumentation on an off-the-shelf mass-market smartphone is feasible after all, as we demonstrate with the Samsung Galaxy S III. We are able to measure CPU, RAM, GPU, SoC, display, WiFi and cellular wireless, audio, camera, GPS and environmental sensors; some directly and others indirectly. This enables us to analyse smartphone features which were not available on the FreeRunner, including GPU 3D acceleration, an OLED display, 3G cellular network connectivity, a multi-core CPU, and a camera.

We measure the power consumption of these components under a number of real-world benchmarks, including: sleep, gaming, video and audio playback, phone calling, SMS messaging, emailing, web browsing and photography. We then compare the results with the previously published measurements of earlier generation smartphones.

In summary, our contributions are threefold:

- We demonstrate that fine-grained power measurement is feasible on an off-the-shelf mass-market smartphone (Section 2.2).
- We analyse the energy consumption of smartphone components not covered in prior work, including the GPU, hardware video codec, 3G network, camera, and environmental sensors (Section 3).
- We update the literature with data for a popular, latest-generation smartphone featuring multi-core CPUs and highly-capable GPUs. We compare this with analyses of previous-generation devices and examine the trend in smartphone power consumption (Section 5).

2 Methodology

2.1 Device under test

We conduct our tests on the Samsung Galaxy S III GT-I9300 (S3) smartphone. This is a high-end device, released in mid 2012. Its main components are summarised in Table 1, and the software is listed in Table 2.

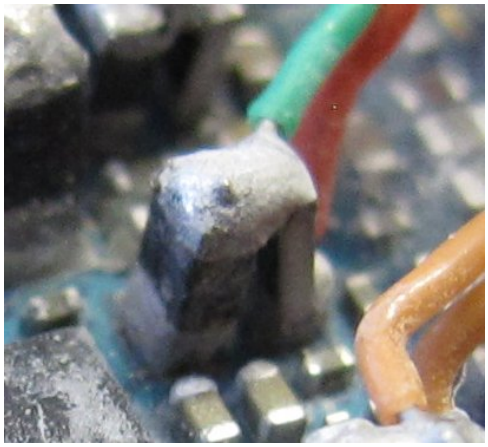

Figure 1: Instrumentation setup for one of the power supplies. The SMPS inductor is on the left, partially lifted off the board and soldered to a sense resistor on the right. Connected to the resistor is a twisted-pair wire leading to the data acquisition system.

Component	Specification
SoC	Samsung Exynos 4412
CPU	ARM Cortex-A9 quad-core, 1.4 GHz
RAM	1 GiB LP-DDR2
GPU	ARM Mali-400 MP
Display	Super AMOLED, 4.8", 720×1280
Battery	2100 mAh

Table 1: S3 hardware technical specifications.

2.2 Instrumentation

Modern consumer devices such as smartphones have a large number of power supplies for various components, which, in theory, enable per-component analysis of power consumptions. However, there are generally no schematics available which would help identify the supplies powering each component, nor are they equipped with measurement capability.

Fortunately, a common design feature helps. Smartphone designs utilise switch-mode power supplies (SMPS) to achieve good energy efficiency in converting the battery voltage to the various required supply voltages. Inherent to an SMPS is an output inductor, which is typically large and discrete (i.e. not integrated with other packages). This not only helps identifying power rails, but the inductor provides a convenient point at which the circuit can be opened and a series current-sense resistor inserted for power measurement. An example of such an instrumentation is shown in Figure 1.

We identified the relevant components on the circuit board by observing changes in the SMPS output waveforms while executing different subsystems from software. We also discovered some useful information by correlating measurements with data in the Linux source code and sysfs files. We discovered a partial OLED supply by searching for voltages above the battery supply, with the knowledge that OLED requires a higher voltage.

Software	Version
Android OS	4.0.4
Linux kernel	3.0.15
Baseband	I9300XXLEF
Build	IMM76D

Table 2: S3 software.

We apply this methodology to the S3, which allows us to directly measure: the CPU cores, RAM, GPU and several general system-on-chip (SoC) supplies. For the SoC, we distinguish three supplies: *MIF* (memory interface), *INT* (internal), and remaining miscellany *SoC*. Measuring power supplies to the radios is less straightforward. Radios are very sensitive to noise in the power supply, and therefore, some of their power is supplied from linear regulators, which are typically less energy efficient but show superior noise performance [6]. As these have no discrete components, we cannot insert a sense resistor. We therefore determine WiFi and cellular power by subtracting the sum of measured components from total power. We also use this approach to determine camera and audio power, which we were unable to measure at all.[1] Furthermore, we were only able to measure one of several power supplies for the OLED display. However, we found that

$$P_{display} = 223\,\text{mW} + P_O \times 1.57$$

is a fairly accurate model of the overall display power $P_{display}$, where P_O is the part we could measure directly. This model exhibits less than 3% error when validated against several images at varying brightness levels.

Similarly, we could only measure part of the power drawn by flash storage, but in this case we were not able to determine an accurate model. Instead we use the partial measured value to predict the worst-case flash power consumption under a number of different access patterns. This becomes relevant only for the camera scenarios; in all others, flash power consumption is negligible (< 8 mW).

Full instrumentation of the smartphone, including circuit analysis, took about 2 person-weeks of work.

In all cases, the power values we report are what we measure for a relevant subsystem, multiplied by a fixed scaling factor to account for the energy loss in voltage regulation. We determined experimentally the efficiency

to be approximately 83%, resulting in a scaling factor of 1.2. We also report total power consumption of the device, P_t, as measured at the battery connector.

3 Results

In this section we present the results of a series of benchmarking scenarios, each of which is averaged from a minimum of three iterations. Where appropriate, the device is run in airplane mode to eliminate external sources of noise. The additional energy cost of an active network connection can be inferred from the data presented below under *idle and suspend*.

Furthermore, all results are reported at 50% display brightness. However, power at other brightness levels can be determined my multiplying the OLED component of display power (see Section 2.2) by between 0.16 (for minimum brightness) and 2.0 (for maximum brightness).

For scenarios based on interactive applications (such as gaming, web browsing etc.) we used a trace of touchscreen input events to drive the benchmark. This approach yields good repeatability across iterations.

3.1 Usage scenarios

Idle and suspend Figure 2 shows power consumption in the *suspend* state—the device's deepest sleep state—for four configurations: airplane mode (no radios enabled), with a 2G (i.e. GSM) and 3G (UMTS) cellular network connection, and with a WiFi connection only. Data transfer is disabled in the cellular case, and for WiFi, we discarded any iteration where a data transfer occurred. This yields the power required to maintain the connection, but does not include any background data traffic, which is highly user-dependent.

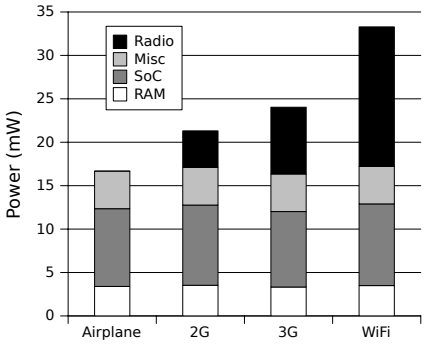

Figure 2: Suspend power in airplane mode, with a 2G and 3G cellular network connection, and a WiFi connection.

Misc is a small (< 10 mW) static power component (determined by subtractive analysis) which we could not attribute, as it is not provided by any of the instrumented supplies. It seems essentially usage-independent and we therefore ignore it for the remainder of the paper. *Radio* is the respective WiFi or 3G subsystem.

Figure 3 shows a breakdown of the device power in the *idle* state; that is, fully powered up but not executing any applications. We present the airplane-mode data, but note that the cost of maintaining a 2G/3G network connection is similar to the *suspend* case (5–10 mW), and that WiFi costs an additional 50 mW above airplane mode.

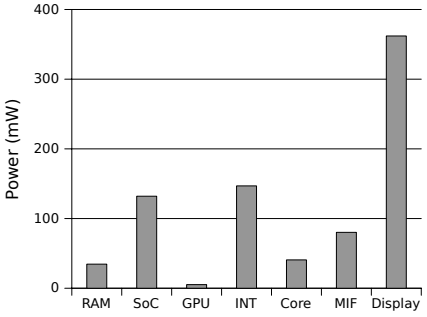

Figure 3: Idle power in airplane mode. $P_t = 805$ mW.

Gaming Two gaming scenarios are shown in Figure 4. The first is the 2D game *Angry Birds*. The workload consists of loading the application and playing through two levels, for a total time of 90 s. The second is the graphics-intensive 3D game *Need for Speed Most Wanted*. Data is collected during a game for a period of 110 s.

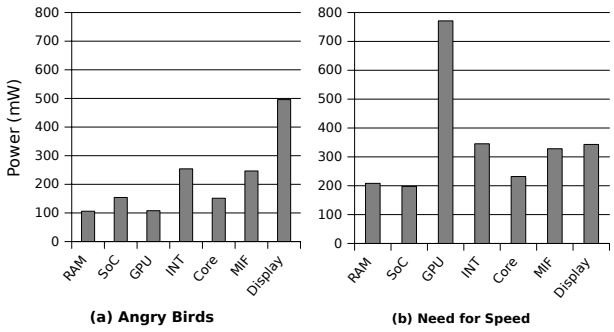

Figure 4: Gaming power consumption for: (a) Angry Birds (2D), $P_t = 1516$ mW; and (b) Need for Speed (3D) $P_t = 2425$ mW.

Video In Figure 5 we plot video playback of two 1280x720 (720p) H.264 videos; one at low quality (1013 kbps) and one at high quality (9084 kbps), played with the Android application *MX Player*, v1.6j. For each video we measured playback with the hardware video codec, and with a software decoder. Comparing software to hardware playback we note a decrease in all power supplies except *INT*, suggesting that this powers the hardware decode unit.

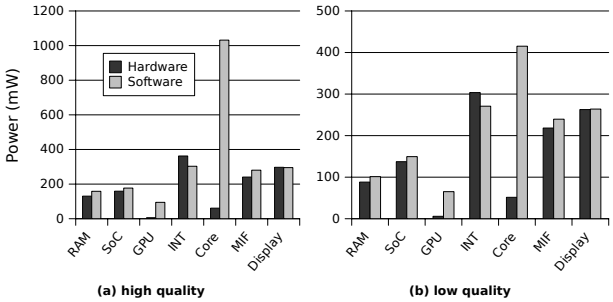

Figure 5: Video playback power consumption at two quality levels with hardware-acceleration and software decoding (a) high quality $P_{\mathrm{t,hw}} = 1270\,\mathrm{mW}$, $P_{\mathrm{t,sw}} = 2329\,\mathrm{mW}$; and (b) low quality $P_{\mathrm{t,hw}} = 1084\,\mathrm{mW}$, $P_{\mathrm{t,sw}} = 1571\,\mathrm{mW}$.

Audio Figure 6 shows the MP3 playback scenario through headphones at minimum volume. We see a 5 mW increase of total power for maximum volume, which is distributed across several supplies. The display is disabled throughout.

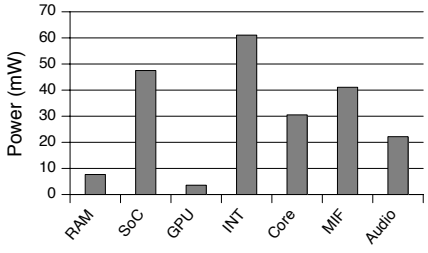

Figure 6: Audio playback $P_{\mathrm{t}} = 226\,\mathrm{mW}$.

Phone call and SMS In Figure 7 we show the power usage for placing a phone call and sending an SMS. The phone call consists of opening the dialer, a 10 s ring time, 40 s of talk time, then returning to the home screen. For SMS we include 45 s to load the messaging application

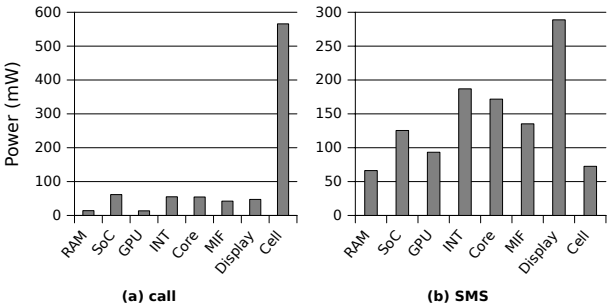

Figure 7: (a) phone call $P_{\mathrm{t,call}} = 865\,\mathrm{mW}$; and (b) SMS message $P_{\mathrm{t,sms}} = 1140\,\mathrm{mW}$, over 3G network.

and type a short message, plus additional time to transmit the message via the 3G cellular network.

Web browsing Our web browsing workload consists of loading the BBC news mobile website, browsing the front page headlines, and reading three articles, for a total of 180 seconds. Figure 8 shows the results for both WiFi and 3G data networks.

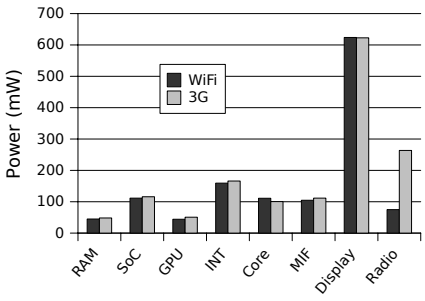

Figure 8: Web browsing over WiFi and 3G network. $P_{\mathrm{t,wifi}} = 1275\,\mathrm{mW}$, $P_{\mathrm{t,3g}} = 1479\,\mathrm{mW}$.

Email Figure 9 shows the results for our email benchmark. In this scenario we use Android's built-in email application to fetch and read five emails, one of which contains a 60 KiB image, and replying to two messages. The total run-time is 140 s, and we repeat the scenario for both 3G and WiFi networks.

Camera In Figure 10 we plot the power consumption under two camera workloads: still and video. The still image scenario consists of loading the camera application, focusing on the near-ground object, taking the photograph (no camera flash), and finally viewing the image.

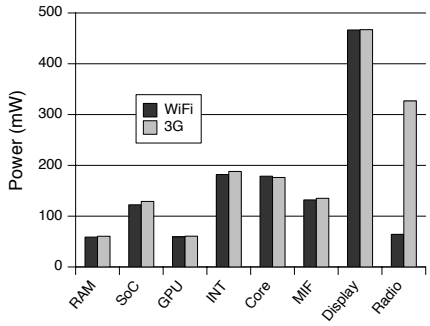

Figure 9: Email over WiFi and 3G network. $P_{\text{t,wifi}} = 1264\,\text{mW}$, $P_{\text{t,3g}} = 1543\,\text{mW}$.

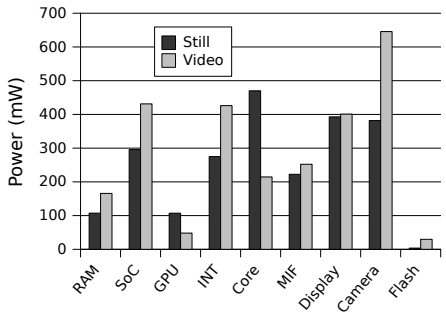

Figure 10: Camera still and video. $P_{\text{t,still}} = 2256\,\text{mW}$, $P_{\text{t,video}} = 2614\,\text{mW}$.

For the video capture scenario, we load the camera application and take a 20 s video.

3.2 Network

In Table 3 we show the throughput and power performance of the 3G subsystem under a microbenchmark—the *speedtest.net* Android application. Since controlling for environmental factors such as network operator load or signal strength is difficult, we ran this benchmark 16 times spread over a period of 24 hours.

	Upload	Download
Throughput (kbps)	547 ± 149	1317 ± 814
Cell power (mW)	1137 ± 372	768 ± 64
Efficiency (kb/J)	481	1715

Table 3: 3G network performance (average $\pm$ std. dev.)

3.3 Sensors

We measured the energy consumption of most of the device's environmental sensors, including: accelerometer, gyroscope, light meter, magnetometer, barometer and proximity sensor. We used a modified version of the `Data Monitor` application to sample each sensor at 50 Hz. We then compared the total power consumption against the same application running with a "dummy" sensor, which delivers software events to the application at a rate of 50 Hz, but involves no hardware. In Table 4, we tabulate the increase in power consumption for each sensor, compared with the dummy, averaged across 3 runs of 60 seconds each.

Sensor	Power (mW)
Accelerometer	5 ± 2.3
Gyroscope	30 ± 1.3
Light	3 ± 1.7
Magnetometer	12 ± 0.6
Barometer	1 ± 0.7
Proximity	7 ± 2.2

Table 4: Sensor power consumption (average $\pm$ std. dev.) above the "dummy" sensor. $P_{\text{dummy}} = 573\,\text{mW}$.

3.4 GPS

We were unable to measure GPS power consumption on our instrumented device, since an open case disconnects the antenna. Instead, we estimate GPS power by measuring the total power consumption on a different Galaxy S3 (this time a GT-I9305 model). Again, the `Data Monitor` application is used.

In Table 5 we report the power consumption above the idle power of 573 mW, separated into two parts: "acquisition" is the time until a position is reported (fix), "tracking" is the time thereafter. We experimented with both hot and cold GPS state, but found no significant difference in average power consumption. However, with cold GPS state, the device took approximately 400 seconds to acquire a fix, consuming 154 J of energy. For warm state (5–10 minutes old), this reduced to approximately 30 seconds.

Mode	Power (mW)
Acquisition	386 ± 19.5
Tracking	433 ± 21.5

Table 5: GPS power consumption (average $\pm$ std. dev.) above idle ($P_{\text{idle}} = 573\,\text{mW}$).

3.5 Maximum power

In addition to the realistic usage scenarios, we ran a series of tests to determine the maximum power consumption for some of the interesting components. The results of this, and the workload that evoked that behaviour, are summarised in Table 6.

Component	Power (mW)	Workload
Core	2845	AnTuTu Benchmark
RAM	208	Need for Speed
GPU	1415	AnTuTu 3DRating
3G	1137	Speedtest.net (upload)
Display	1124	White screen, bright

Table 6: Maximum power.

4 Analysis

Display power is a significant contributor in every scenario where the display is active. Even at the 50% brightness level it contributes 300–600 mW, highly dependent on the image content—an inherent property of OLED technology. Just varying the colour scheme can save up to 300 mW at medium brightness, or 600 mW at maximum brightness. In scenarios where a data network is used, namely web browsing and email, WiFi networking is approximately four times more efficient than 3G. This corresponds to a 20% reduction in total energy. However, the cost to maintain an unused data connection is less in the 3G case ($\approx$ 10 mW) than the WiFi case (50 mW).

Our video results demonstrate the efficacy of using the hardware video decode unit, rather than a software decoder. For a high-quality video, a saving of more than 1 W (45%) can be achieved, which translates to 2.6 hours of additional playback time.

The camera workloads show particularly high power consumption in the camera subsystem, but also across the various SoC supplies, the CPU core, and RAM. Surprisingly, these two scenarios are among the highest energy consumers across all our benchmarks.

While RAM power can be quite high ($>$ 100 mW in some scenarios), this always correlates with high CPU or GPU power consumption. Therefore as a percentage, RAM contributes on average only 6% of total power. In suspend mode however, this is up to 20%, since RAM must retain significant system state.

In every scenario, flash memory is a negligible contribution to power consumption. It peaks at 30 mW in the camera video recording benchmark (1.5% of total), and for all others is less than 0.5% of total.

Environmental sensors contribute little to overall power consumption, even at very high sampling rates. GPS however consumes significant power: around 40% of total. Moreover, our data demonstrate that the relative magnitude of network and GPS power consumption (approximately 2:1 for 3G) is such that the use of network-assisted GPS (such as A-GPS) is clearly a win in total system energy consumption.

The maximum power data demonstrate that, in principle, the S3 can consume in excess of 7 W. However, in realistic scenarios most, if not all components, are highly under-utilised. Consequently, efficient intermediate power states should be considered equally important to energy consumption as deep low-power states, as this disparity will likely continue.

5 Trends in Smartphone Power Use

How do the results for the latest-generation phone compare with the findings from three years ago [2]? There we analysed the then new Google Nexus One (N1) and the roughly one year older HTC Dream (G1). We also had the more detailed analysis of the Openmoko FreeRunner, but comparing to that device is less meaningful, since it was not a mass-market high-end smartphone.

In Table 7 we compare the total system energy consumption across the three generations of smartphone; the G1 and N1 data is reproduced from our earlier paper, and the S3 data is from the present work. In all three cases, the brightness-dependent component of display power has been subtracted out.

We observe very similar suspend power for the G1, N1 and S3 ($\approx$ 25 mW), suggesting that this may be an intrinsic lower bound. Idle power, however, appears to increase over time. We speculate that this may be due to increasing display resolution—the relationship between pixel count and idle power is approximately linear across these three devices. Moreover, in scenarios where the display is not active, we see either a reduction or no significant change in total power consumption, further supporting this conclusion. The power consumption for a voice call has not appreciably changed across the three devices; the G1, N1 and S3 are within 15% of each other, with the latest device consuming the highest power in this scenario. Audio playback power consumption has dropped steadily across the three generations, from 460 mW on the G1 to 226 mW on the S3. Since the display is disabled and audio levels unchanged, this shows a substantial increase in energy efficiency in the newer devices.

While average power consumption in typical working scenarios is not increasing substantially through generations of smartphones, maximum power is. This suggests that thermal management will increasingly become an important aspect of smartphone power management.

	Average System Power (mW)		
Benchmark	G1	N1	S3
Suspend	27	25	24
Idle	161	334	666
Phone call	822	747	854
Email (cell)	599	-	1299
Email (WiFi)	349	-	1020
Web (cell)	430	538	1080
Web (WiFi)	271	412	874
Audio	460	322	226

Table 7: Average system power for a range of usage scenarios across three smartphone generations.

6 Related work

Perrucci et al. [5] presents power and energy measurements for a range of microbenchmarks on a Nokia N95 smartphone, focusing on wireless interfaces. They use only total input power, and exercise subsystems in isolation with subsystem-oriented workloads to determine their power consumption. Such an approach is inherently limited for composite workloads, where multiple subsystems work in tandem for a single purpose. For instance, downloading a file via 3G exercises not only the wireless module, but requires the CPU to issue the commands and manage the network stack, and RAM to store the data. However, the micro-behavioural detail they provide, such as idle power for various radio technologies, can be easily applied to our full-system data.

There is significant work in the literature on modeling of power consumption. Recently for example, Pathak et al. [4] use an approach based on system call tracing to drive a state-machine model of devices. Our approach provides ground truth for such work. Moreover, McCullough et al. [3] argue that modeling of power consumption is becoming more error prone, due in part to hidden power states of devices, motivating a direct measurement approach. We describe a methodology for direct measurements on a smartphone, and provide concrete data from a real device.

7 Conclusions

We have shown that fine-grained power measurement is feasible on an off-the-shelf smartphone which is not designed for that purpose, even without hardware documentation. With the described methodology, we were able to measure or infer power consumption for all the major components of the Galaxy S III.

We have presented the power consumption of this device under a range of realistic workloads, and compared these data with previously reported measurements on earlier generations of smartphones. We found that idle power seems to be mostly determined by screen size, while the power draw of an idle application processor is negligible. This indicates that static power, which is becoming an increasing problem in servers [1] is still mostly a non-issue in phones. However, maximum power is steadily increasing. Hence, while compute power of smartphones continues to grow, it is becoming less affordable to use it for more than brief periods.

The raw data and information on instrumentation points is available at `http://ssrg.nicta.com.au/projects/energy-management`.

Acknowledgements

NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the Australian Research Council through the ICT Centre of Excellence program.

References

[1] BARROSO, L. A., AND HÖLZLE, U. The case for energy-proportional computing. *IEEE Comp. 40*, 12 (Dec 2007), 33–37.

[2] CARROLL, A., AND HEISER, G. An analysis of power consumption in a smartphone. In *2010 USENIX ATC* (Boston, MA, USA, Jun 2010).

[3] MCCULLOUGH, J. C., AGARWAL, Y., KUPPUSWAMY, J. C. S., SNOEREN, A. C., AND GUPTA, R. K. Evaluating the effectiveness of model-based power characterization. In *2011 USENIX ATC* (Portland, OR, Jun 2011).

[4] PATHAK, A., HU, Y. C., ZHANG, M., BAHL, P., AND WANG, Y.-M. Fine-grained power modeling for smartphones using system call tracing. In *6th EuroSys Conf.* (Apr 2011).

[5] PERRUCCI, G., FITZEK, F., AND WIDMER, J. Survey on energy consumption entities on the smartphone platform. In *73rd IEEE Vehicular Technology Conference* (May 2011).

[6] VOLK, K. R. Dealing with noise when powering RF sections in cellular handsets. `http://www.eetimes.com/document.asp?doc_id=1277716`, Jul 2002. [Online; accessed Jan 2013].

Notes

[1] This of course precludes measuring more than one of the WiFi, cellular, camera or audio subsystems simultaneously, but we believe this does not eliminate any significant benchmarking scenarios.