The Power of Smartphones

Roy Want, Google

There are several characteristics that distinguish mobile computers from static desktop computers—low-power operation, wireless networking, mobile sensing, low weight, small volume (fits in a pocket), and a constrained user interface. Of these characteristics, however, low-power operation trumps them all. Without power, nothing else really matters.

If you're new to power monitoring in the mobile design process, either when building mobile hardware or writing software-based applications, this article will point you in the right direction, helping you identify what characteristics to consider and what test equipment to use.

INCREASING THE OPERATIONAL LIFETIME

So how low is low-power? Power is energy per unit time. Ultimately, we judge a smartphone's energy consumption based on its operational life before it needs to be recharged. A design engineer can increase this operational lifetime by either increasing the energy capacity of the phone's battery, or reducing its average power consumption.

Improving Battery Technology

Over the last 30 years, rechargeable battery energy capacities have increased as they've evolved from Nickel Cadmium to Nickel Metal Hydride to Lithium and Lithium-ion. Today, a typical smartphone battery is based on Lithium-ion technology, with a capacity of approximately 2,000–2,500 mAh. This allows a modern phone to operate for, on average, at least a day before needing to be recharged. Even if we increase battery capacity, it's often the case that more power-hungry applications are created—as long as a day's use is possible. However, the improvement of battery technology has been slow compared to technology in other high-tech areas—such as memory capacity and processor performance, which have also been important to deliver the functionality we expect from a smartphone.

The reality is that it's hard to make a better battery that's both high capacity and safe. In fact, even Li-ion batteries, if not manufactured to high standards, can pose a fire risk. A simple YouTube search will turn up dramatic pictures of devices (usually laptops) whose Lithium batteries caught fire, although the exact circumstances that caused the problem are usually unclear. These images demonstrate the potential for disaster in all mobile devices. It's unlikely we'll develop higher capacity batteries that don't pose some sort of danger given certain conditions. This is because, by definition, you're storing more and more energy in a small volume device. Furthermore, a smartphone is often kept in a pocket close to your skin. Overheating, or even worse, a battery fire, are clearly undesirable outcomes.

Reducing Power Consumption

The alternative approach is to reduce the mobile device's average power consumption, and thus energy use, over time. Our industry has fortunately been through a golden age of semiconductor improvements, with progressively smaller lithography, purer materials, and a better understanding of solid state physics.

CMOS Field Effect Transistor (FET) technologies have been downwardly scalable to enable more transistors to be manufactured in the same area of silicon, year after year (Moore's Law).[1] This also reduces the dynamic power of these devices—that is, the power consumption resulting from transistors switching on and off, because the energy dissipated as heat at the gate of a switching CMOS transistor is proportional to the energy stored in the gate. This can be quantified as $1/2\ CV^2$, where C is the gate capacitance and V is the operating voltage. The power dissipation (energy per second) based on a full clock cycle at frequency f is therefore CV^2f. As gate area A decreases, so does capacitance $C = \varepsilon A/d$, where d is the separation of materials and ε the permittivity of the gap between them. Consequently, at a constant frequency, as the gate area decreases, the dynamic power being consumed per device also decreases.

In addition, the lower capacitance means that the gate charging time is smaller, enabling a higher maximum frequency of operation. Engineers have historically had the option of using next-generation devices (smaller lithography) to either increase performance by running the system clocks faster (resulting in higher power) or keeping the clock constant and operating at lower power (or some compromise in between).

Published by the IEEE CS ■ 1536-1268/14/$31.00 © 2014 IEEE

There is also the issue of static power consumption resulting from energy leakage from CMOS devices while not switching, due to the decreasing thickness of a FET's gate dielectric. In the more recent history of lithography and CMOS technology, the inherent increase of static power dissipation has been more significant than dynamic power. As a result, the design of a FET to control static power has resulted in an effective upper limit to the processor clock frequency, leading to the trend of increasing processor performance using a multicore design architecture rather than increasing the clock frequency.

Designing for Low Power

Designing for low power in the processor can also be addressed by considering the architectural implications of power. As I noted earlier, clocks driving the gate of an FET give rise to dynamic power consumption. If the system can determine that functional components inside the processor—for example, the components of the Arithmetic Logic Unit (ALU)—aren't being used at a particular time, then gating off the clock from these components will save power without impacting performance. This principle can be applied to all aspects of system design, including data buses between components. In fact, the golden rule of low-power architectures is, if it's not being used, turn it off.

Another important aspect of low-power design focuses on the supply voltage. Since the power dissipation at the gate of an FET is proportional to 1/2 CV^2, reducing the supply voltage reduces power by the square of the decrease in voltage. Over the years, we've seen microprocessor core designs evolve from a 5 volt supply to 3 volts and recently close to 1 volt, resulting in up to 25 times less power dissipation based on supply voltage alone. We're probably near the limit of supply voltage reduction possible with CMOS due to the inherent property of silicon semiconductor junctions and the need to accurately represent a binary logic 1 and 0 as two distinct states.

RELATED URLS

Monsoon Power Monitor: www.msoon.com/LabEquipment/PowerMonitor

Fluke NetDAQ 2456a: http://us.flukecal.com/products/data-acquisition-and-test-equipment/data-acquisition/netdaq-networked-data-acquisition-unit

Considering the Applications

So far, I've been considering the power characteristics of the underlying semiconductor technology. However, a major factor that determines the actual power consumed by a smartphone is the applications that are run and the resources that the application needs, such as cellular wireless technology (3G, LTE), Wi-Fi, Bluetooth, GPS, the display, user interaction, and the processing cycles required to perform the function. For example, playing a video continuously will consume more processor power than playing a game of tic-tac-toe. The latter lets the processor remain idle while the user is considering his or her next move.

When a developer creates a smartphone application, it's quite difficult to estimate the resulting battery drain just by inspecting the program. There are many nondeterministic factors, including network latency, server latency, and the whims of user interaction, which aren't easily predicted. Sometimes the best way to reduce power is to design the application, measure its average power use in operation, and then use this as a baseline to optimize components of the program that either activate energy hungry resources, such as GPS, or engage in prolonged processing periods, stopping the processor from idling or entering its sleep mode. Sometimes the application's algorithms can be optimized to reduce these issues. Other times, the overhead of interacting with a resource (for example, a network server) can be optimized by batching requests together[2] and making one network request rather than several.

There have been a number of articles published in this publication in prior years that have discussed the impact of an application or service in terms of power consumption. The ability to measure smartphone power has been a key tool for characterizing an application's operation and improving the final design.

MEASURING POWER

So how do you measure the power, and ultimately the energy consumed, by an electronic device and its applications? Power (Watts) is a measure of energy (Joules) consumed per second and it can be derived from the following equation:

power = potential difference across load (Volts) × current flowing through load (Amps).

In terms of a smartphone, the potential difference across the load is simply the supply voltage (V_s). In a perfect world, this should be the labeled battery voltage, but in practice it will vary depending on the remaining charge in the battery. In a power test, we can, however, pin this voltage at a constant value by using a regulated power supply to replace the battery and power the device.

Measuring the current drawn by the device is a little more complicated. We could use a standard handheld current meter, but these devices typically have their own averaging circuits that don't let us read the correct instantaneous current occurring during rapidly changing processing demands—it can, however, allow a rough estimate to be made.

For accurate measurements of power, it's necessary to rapidly and accurately sample the changing current drawn by the device. In practice, it's much easier to measure a changing voltage than to

SMARTPHONES

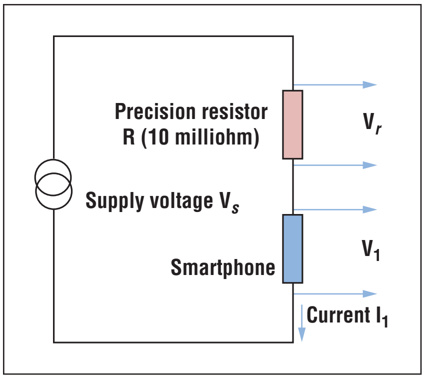


Figure 1. Using a series resistor to measure the current flowing in a load.

Figure 2. The Monsoon power monitor.

directly measure the current. Current (I) can be converted into a voltage (V_r) by using ohm's law. That is, $I = V_r / R$, where R is the resistance the current is flowing through and V_r is the voltage dropped across that resistance. Thus, by placing a known resistor in series with the power supply and the device, and measuring the instantaneous voltage drop across the resistor, we can calculate instantaneous current (see Figure 1).

One problem that arises is that the current flowing through the series resistor also reduces the supply voltage across our smartphone, which then affects the current drawn, often in a nonlinear way. To reduce this effect, the series resistance chosen must be very small and its value accurately known. Typical resistors used for this purpose are approximately 10 milliohms and can be purchased from common component suppliers, such as DigiKey, but are much more expensive than typical resistors found in everyday products with values of 10 ohm to 1 Mohm, mainly because of their precision and the small number of products that need them.

The instantaneous power consumed by the load will be the potential across the load V_l, multiplied by the current flowing through the load I_l:

$$V_l \times I_l = (V_s - V_r) \times (V_r/R).$$

If V_r can be sampled 1,000 times per second, it's possible to have reasonably accurate power measurements based on 1 millisecond time increments. These results can be captured with a standard digital oscilloscope found in most labs. To convert the sampled data into average power, we must record the data over a time window and then average the results. Average power can then be used to estimate energy consumption and battery lifetime for that application:

Lifetime [hours] = Battery Capacity [mAhr]/(Average Power [mW] / Battery Voltage [V]).

USING POWER MONITORING EQUIPMENT

With mobile devices beginning to dominate the consumer computer market, interest in power monitoring has become more common and several companies make affordable, easy-to-use power test equipment. These products remove many of the issues that hinder setting up power monitoring experiments.

The Monsoon Power Monitor

The power monitor, shown in Figure 2, is a good example of a device in this category, retailing for approximately US$750. The power monitor is a USB peripheral for a PC that can record the power consumption of a single device. It uses a PC to provide the user interface and displays the power consumption graphically in real time in a window. Numerical results can also be stored in a file and placed in a spreadsheet for later analysis. However, it also calculates the average power consumed over the sampling window and then displays it automatically in the user interface. This is usually the main quantity that we care about and often no further calculations are required.

The Monsoon user interface provides all the features you would expect from a standard piece of test equipment (Figure 3).

Supply voltage. The supply voltage is programmable up to 4.5 volts and can be set at a custom value or chosen from a set of standard values. The monitor's output voltage can be enabled or disabled under user control to help manage the experimental process, and record the power-on characteristics.

Current protection. This is designed to protect the test device and can be set up for both the power-on conditions (to limit an initial burst of current when the devices is turned on) and the maximum expected current while the device is continuously running. Sensible defaults are chosen for the maximum values, but a user can override them as needed.

Y-axis range. You can choose the range of values displayed on the vertical axis, such as current, power, and supply voltage. There are also check boxes to choose which quantities should be displayed in parallel, including their average, maximum, and minimum values.

X-axis range. The horizontal axis displays time. You can also set the sample rate, the time units associated with graduation tick marks along the axis and the number of ticks presented on the screen.

Trigger control. The sampling process can be triggered manually using start and stop buttons, or under various trigger conditions including exceeding a threshold power. Triggers can also be setup to stop data collection.

User interface. The interface provides buttons to allow the displayed graph to be copied to the copy buffer and pasted into other programs; this is particularly useful for capturing the results as an image that can be pasted into a summary document.

I've personally had experience with this device for over a year and found it a valuable tool for many day-to-day power experiments.

NetDAQ Data Logger

If you're interested in a more detailed power breakdown of a device, considering the many components found in a mobile system (for example, the processor, flash memory, dynamic memory, cellular radio, Wi-Fi radio, touchscreen, and display), you will need to capture multiple-power readings in parallel.

In the past, I have used another piece of equipment called a NetDAQ Data Logger made by Fluke (see Figure 4), which can record 20 analog channel potentials in parallel (expandable to 400). To sample power, you need to instrument your mobile device in the form of a test board with tiny series resistors, as described earlier, placed in series with the supply pins of each component in which you're interested. You can then make the connection across each resistor available as a set of test pins that can be used to connect to the various channels of the NetDAQ. Achieving this level of testing is only practical if you're implementing a completely new device and can design a printed circuit board that includes the required test points. It would be virtually impossible to retrofit an existing smartphone in this way.

The NetDAQ also uses a PC as its primary user interface, allowing graphs to be generated automatically from the data. A convenient on-board feature is a mechanism to apply a linear function to the captured data, enabling it to be automatically scaled to the desired power measurements. Thus, the values of the series resistors can be factored into the function, along with the supply voltage, resulting in a complete multichannel power monitoring solution.

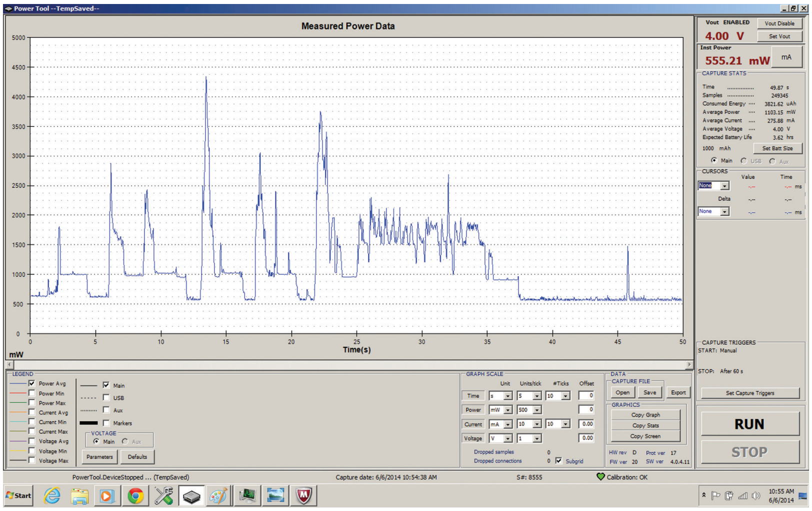
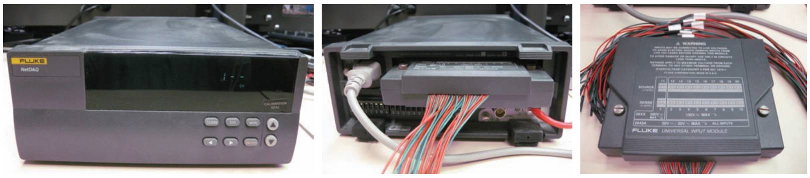


Figure 3. The Monsoon user interface and a typical power versus time graph.



(a) **(b)** **(c)**

Figure 4. NetDAQ Data Acquisition device: (a) front, (b) back, and (c) the 20-channel input module (inserts at backside).

Pervasive computing is about embedding computing everywhere and includes both mobile and fixed infrastructure devices. Although it's possible to harvest small amounts of energy from the environment—for example, from light, heat, and vibration,[3] or from ambient signals used for radio or television—the results are small compared to the energy consumption of the mobile applications we expect to run on a day-to-day basis. As a result, it is safe to conclude that understanding average power, the energy consumption over time, and its implications when designing an application will be an important part of pervasive computing research for the foreseeable future. ◼

REFERENCES

1. G.E. Moore, "Cramming More Components onto Integrated Circuits," *Electronics Magazine*, vol. 38, no. 8, 1965, p. 4.
2. A. Sharma, P. Eastham, and F. Nerieri, "Designing an Energy-Efficient Cloud Messaging Service for Smartphones," *IEEE Pervasive Computing*, Jan.–Mar. 2014, pp. 84–88.
3. R. Want, K.I. Farkas, and C. Narayaswami, "Energy Harvesting and Conservation," *IEEE Pervasive Computing*, Jan.–Mar. 2005, pp. 14–17.

Roy Want is a Research Scientist at Google. Contact him at roywant@google.com.