Use holistic approach to battery powered designs

Richard Quinell - May 17, 2016

While attending ESC in Boston last month, I had the chance to listen to NXP's Allen Wilson talk about what he called "ultra low power malarkey." Basically, what he was saying was that the sleep mode specs that microprocessor vendors tout are pretty meaningless. You have to take a more holistic approach to designing for battery-powered operation if you want to get a true image of how long your system will run between battery replacements.

One of the first things to look at is the current draw that your design requires, not the power. If you're operating from a battery, after all, you can't really alter its voltage. Even if the processor is operating at a lower (or higher) voltage than the raw battery provides, it's the current drawn from the battery, coupled with the battery's capacity, that determines how long the system will endure.

Next, use the average current, not the sleep current or the peak current, to gauge battery life. Sure, peak and sleep currents matter, but they are simply two data points. Over the normal operating life of the device, how much total current does it draw in, say, 24 hours?

If you make the calculation, you might be surprised at how much impact the duty cycle of active to sleep mode can be. Take as an example an MKW01 from NXP, which has a Cortex M0+ processor and a sub-GHz transceiver in a single package, making it suitable for many kinds of wireless sensor systems. According to the data sheet, the MKW01's sleep mode current draw is 0.1 µA typical (1 µA maximum) and when transmitting it draws from 16 mA to 95 mA, depending on output power.

Let's say we're running at the low end of the RF output power. Allowing for other circuitry, power conversion inefficiency, and other factors, we'll use 20 mA as our active power level. Then, to be generous, we'll take the vendor's typical 0.1 µA value for the sleep mode draw. A high-capacity coin cell (which seems to be the most popular kind of battery used in development kits) like the CR3032 has a capacity of about 550 mAh.

With these figures, our design would be able to transmit continuously for about 27.5 hours before exhausting the battery. Hence, the need for putting the device to sleep in between transmissions. The more time it is asleep, the longer the battery will last - in theory, anyway. The table below shows how much battery life you get if you use these current draws at a variety of duty cycles.

<table>
<thead>
<tr>
<th>Duty Cycle</th>
<th>Seconds per day</th>
<th>Average current (mA)</th>
<th>Days of battery life</th>
<th>Years of battery life</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>8640</td>
<td>2.0001</td>
<td>11.46</td>
<td>0.03</td>
</tr>
</tbody>
</table>
If we want our design to run for ten years between battery changes, it looks like we can only transmit for about one second per hour. Depending on how long it takes for the system to wake up, establish contact, send its data package, and go back to sleep (say, 100 msec), we are looking at several minutes between samples. That may be fine for some systems, but not all, and in any case there is more to consider.

The processor's sleep current is not the only drain on the battery, though. The battery itself has a self-discharge rate, so its capacity is waning even with no load at all. Under load, depending on current draw, the battery's internal resistance will also steal some of the cell's capacity. (For a more thorough analysis of achievable battery capacity, see this article in Embedded.com.)

Then, too, batteries all have a limited operating life because the chemistry they use breaks down over time, which is reflected in their specification of shelf life. The CR3032, a long-life cell, only claims a 7- to 10-year shelf life at best. Many batteries offer far less. So, even if your circuit is optimized for battery life, the battery itself might not live as long as you require.

Then there is the load of any additional system circuits running while the processor is asleep to consider, or their sleep currents if they are shut down between transmissions. There is also the leakage current (or insulation resistance) of the PCB's bypass capacitors to think about. They're draining the battery when nothing else is running.

The bottom line is that looking at low-power processors when considering a battery-powered design is only the tip of the iceberg. You need a holistic, system-wide approach that factors in the battery characteristics, behavior of other circuits, current profiles, and other factors that can waste current or affect battery capacity. Without that in-depth look, your system's operating life between battery changes can turn out to be much less than desired.