Energy Metering for Free: Augmenting Switching Regulators for Real-Time Monitoring

Prabal Dutta[†], Mark Feldmeier[‡], Joseph Paradiso[‡], and David Culler[†]
{prabal,culler}@cs.berkeley.edu {carboxyl,joep}@mit.edu

[†]Computer Science Division
University of California, Berkeley
Berkeley, CA 94720

[‡]The Media Laboratory
Massachusetts Institute of Technology
Cambridge, MA 02139

Abstract

We present iCount, a new energy meter design. For many systems that have a built-in switching regulator, adding a single wire between the regulator and the microcontroller enables real-time energy metering. iCount measures energy usage by counting the switching cycles of the regulator. We show that the relationship between load current and switching frequency is quite linear and demonstrate that this simple design can be applied to a variety of regulators. Our particular implementation exhibits a maximum error of less than ±20% over five decades of current draw, a resolution exceeding 1 μJ, a read latency of 15 μs, and a power overhead that ranges from 1% when the node is in standby to 0.01% when the node is active, for a typical workload. The basic iCount design requires only a pulse frequency modulated switching regulator and a microcontroller with an externally-clocked counter.

1 Introduction

Almost all modern rechargeable devices have a battery life indicator onboard, usually based solely on the battery voltage, but in more sophisticated devices on the average of the energy removed from the cells. These energy data are valuable to the user, as they allow her to modify behavior in order to maximize the utility gained from the device given the limited energy resource. In the area of distributed sensor networks, this information is even more vital, as there is often no opportunity for recharge, and the device itself must make the decisions necessary to accomplish its tasks before its energy is depleted. A number of approaches are taken to minimize this energy consumption, usually involving low-power components which are duty cycled at the lowest possible rates. But there is no good way to verify the effectiveness of these algorithms in an actual large-scale deployment, as measuring the current draw of many distributed nodes is cumbersome and costly.

In order for these devices to be able to make informed decisions, detailed energy usage information must be present for all operating conditions. To date, this has either been accomplished with predictions based on laboratory tests and manufacturer datasheets, or with onboard current measuring hardware [9, 11, 19], which itself draws considerable power. The former solution is lower power, but not as accurate as the latter, and is not as adaptable to conditions which were not foreseen in the testing phase. This problem is particularly challenging because of the bursty nature of sensor node power draws. The measuring device needs to be able to sense sleep currents of around 5 μA and active currents in the range of 50 mA, all with a time response of a few microseconds to catch short wakeup scenarios.

In order to meet the needs of *in situ* current measurement, to validate sensor network power saving algorithms, and to facilitate adaptive power usage, we have developed a system that can measure over the necessary five decades of current draw with negligible increase in total system power consumption and total system size and part count. This design, called iCount, takes advantage of the constant power metering inherent in a pulse frequency modulated (PFM) switching regulator. Since the inductor pulses are limited to a fixed current, the total $\frac{1}{2}Li^2$ energy transferred from the input to the output is also fixed, and the switching frequency varies linearly with the output power (since the output voltage is constant, a linear increase in current leads to a linear increase in power). Therefore, by merely connecting the PFM regulator switch output to a microcontroller counter pin, the total energy consumed at any point in time can be known and recorded. This counting method

of current accumulation gives large range and accuracy as arbitrarily long time windows can be used to average out certain errors.

This paper compares four different PFM regulators, focuses on the Maxim MAX1724 regulator [12], and deeply evaluates an energy meter design built around this regulator. Our implementation exhibits a maximum error less than $\pm 20\%$ over five decades of current draw, a resolution exceeding 1 μJ, a read latency of 15 μs, and an energy overhead that ranges from 1% when the node is in standby to 0.01% when the node is active. Despite these benefits, there are limitations to the system, as it shows variations with both input voltage and temperature, both of which can be compensated for with periodic measurements. Also, the current profile of the PFM regulator requires initial calibration, but its savings in terms of size, part count, and power consumption make it a promising option for real time power measurement in large scale deployments.

Despite these limitations, iCount enables many new measurement-based scenarios for sensor network operation. For example, hardware power profiles can generated by running a simple benchmarking application, long-term power draw characteristics can be validated *in situ*, energy profiles of application software can be generated, real-time current draws can be measured, and energy quanta-based scheduling can be implemented. These new capabilities promise new directions in sensor network research.

2 Background and Related Work

The sensor network community is very focused on the reduction of power in both individual nodes, and in the entire network as a whole. Unfortunately, the majority of this work is based on power measurements which are made on the laboratory bench, and not in a real world use scenario. Instead, shunt resistors, difference amplifiers, and oscilloscopes [4] or current mode digital multimeters are used. These solutions are both bulky and power hungry, making them impractical for wireless sensor nodes which must run on batteries and be deployed in the field.

Although there is a large demand for energy consumption monitoring hardware, the majority of this demand comes from either the portable electronics market or the building energy metering market. Both of these applications have generated many solutions, none of which are particularly suited to the needs of distributed sensor nodes. For portable electronic devices, merely the total energy remaining needs to be known, so the user can plan accordingly. As a result, these devices tend to have low resolution with slow response times, as the user does not need to know precisely how many joules have been used, and remaining battery life will not change appreciably over a few seconds. Almost every major integrated circuit manufacturer produces a battery monitor solution. The majority of these fail to meet the five decades of dynamic range required for sensor nodes. For example, Maxim's DS2438 [13] only has a 10-bit ADC for current measurement while Analog Device's ADM1191 [2] only has a 12-bit ADC and no measurement accumulation.

More complicated battery fuel gauges come closer to solving this problem, but still do not meet the demanding needs of this particular application. The BQ2019 [16] from Texas Instruments obtains very high accuracy and a wide dynamic range by using a current to frequency converter, but has effective sampling rates which only reach a maximum of 30 Hz, which is much too slow to capture the microsecond operations which might be occuring in a sensor node. The BQ27500 [18], also from Texas Instruments, has a lower, but almost adequate, dynamic range of a 14-bit ADC, but it does not save this information. Rather, it keeps a lower resolution accumulation of total battery life remaining via sophisticated algorithms based on voltage and temperature. Although very accurate at predicting battery life, this solution would not be able to give detailed temporal information.

The 120V AC power line measuring devices provide an intriguing option, as they are much more accurate, due to their use for billing purposes. Devices such as the Microchip MCP3096 [14] and the Analog Devices ADE775x [1] line have 16-bit ADC's, and up to 14 kHz frequency response, both of which would be adequate for sensor node use, but these integrated circuits are not designed for low power operation: the MCP3096 [14] draws 4 mA of quiescent current, and the ADE7753 [1] draws 7 mA, as they are not expected to run on battery power. These sorts of continuous current draws would become the dominant drain on the battery in most wireless sensor applications.

The SPOT system by Jiang, et al. provides an example of energy metering that is targeted directly at distributed sensor nodes. Using a current to frequency converter similar to the BQ2019, it provides a dynamic range of 45000:1, a resolution of less than 1 μA, and onboard measurement accumulation. However, SPOT uses the AD627 instrumentation amplifier, the AD7740 voltage-to-frequency converter, and FM3104 processor companion, which draw up to 85 μA, 1.5 mA, and 150 μA, respectively. Although this system meets the accuracy requirements, it requires the addition of a number of integrated circuits, and adds to the current drain on the battery.

An improvement to this system is shown in Malinowski, et al. [11], where the same current to frequency method of charge accumulation is used, but it is incorpo-

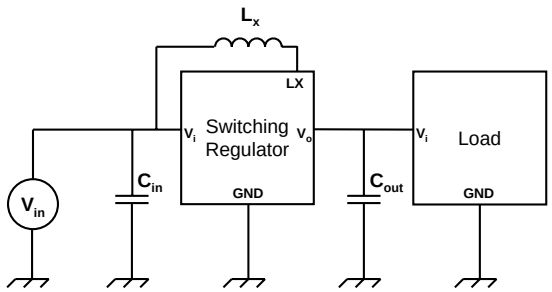

Figure 1. A typical system with switching regulator-based power supply, including the regulator, input capacitor, output capacitor, inductor, and load.

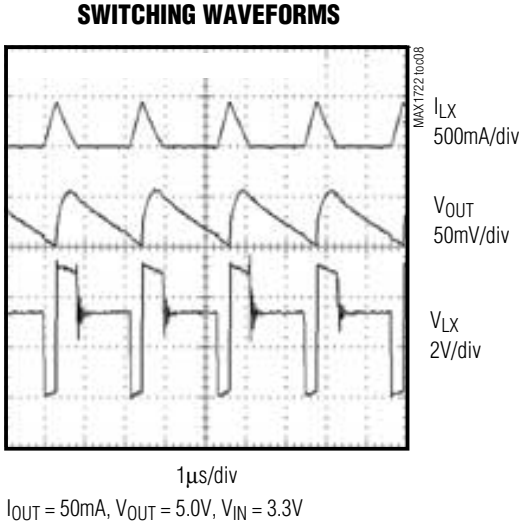

Figure 2. The switching waveform of the Maxim MAX1724 switcher. The inductor voltage, V_{LX}, alternates between the input voltage (3.3 V), ground (0 V), and output voltage (5.0 V). Source: Maxim [12].

rated into a specially designed linear voltage regulator, minimizaing the added quiescent current losses. A difference amplifier maintains the required output voltage by metering out precise pulses of current into a smoothing capacitor. By counting these pulses, the total energy consumed can be known. Although this last solution gives good results, it still has the headroom losses associated with all linear regulators, requires extra components to generate the pulses, and cannot operate from a single cell. These problems are eliminated in our solution which uses an off-the-shelf PFM switching regulator to feed these pulses through an inductor, eliminating the headroom losses, and merely monitors the switching cycles, eliminating the need for extra components.

3 The iCount Design

Many battery-operated devices use a switching regulator, as shown in Figure 1. Such regulators provide a constant output voltage and high conversion efficiency across a range of input voltages and load currents. Although a variety of regulator topologies (boost, buck, buck-boost), control modes (current-mode, voltage-mode) and modulation schemes (pulse-frequency modulated, pulse-width modulated) exist, we focus on boost regulators that employ current-mode control using pulse frequency modulation. Such regulators allow single-cell operation, can supply high currents, and draw ultra-low quiescent currents, making them ideal for low-power, battery-operated systems that exhibit a wide dynamic range in power draws. Unless otherwise noted, we use the terms *switcher* and *regulator* interchangeably in the remainder of this paper to describe PFM regulators.

A switcher goes through three stages during a switching *cycle*, as Figure 2 shows. Each cycle delivers $\frac{1}{2}Li^2$ J, where i is the peak inductor current (*i.e.* the max value of I_{LX}). A cycle begins when the switcher senses that the output has fallen below the regulation threshold. During the first stage of a cycle, the switch-side of the inductor (LX) is connected to ground. The resulting potential difference across the inductor gives rise to a steadily increasing current. When this current reaches a limit (or some maximum on-time has passed), the switch-side of the inductor is disconnected from ground. During the second stage of the cycle, the switch-side of the inductor is connected to the switcher output, which discharges the inductor energy into the output capacitor. When the inductor current ramps to zero, the discharge stage is complete. Sometimes, the inductor and output capacitor form a resonant circuit and this stage ends with ringing oscillations. During the third stage, the switcher maintains a quiescent state while the load draws current from the output capacitor. A cycle repeats when the output falls below the regulation threshold.

The iCount design is motivated by the simple observation that many switchers exhibit a nearly linear relationship between switching frequency and load current over a wide dynamic range. Figure 3 shows how the switching frequency changes with load current for several different commercial switchers. In this figure, the bias, or no-load switching frequency, has been subtracted from each data point. Fortunately, such biases can be subtracted easily in software. Although the data

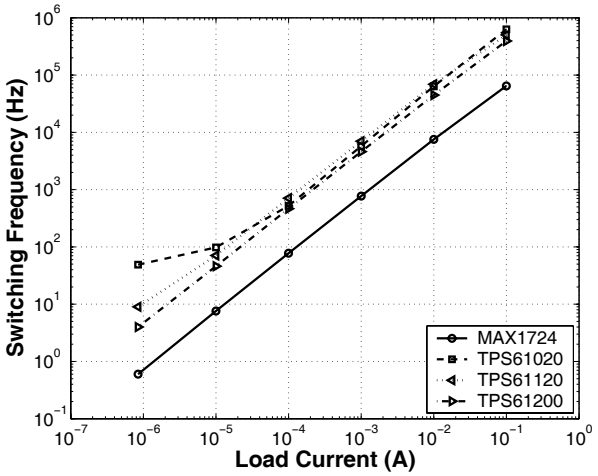

Figure 3. The relationship between load current and switching frequency for several switchers, after bias compensation. Some switchers are more linear than others (and sometimes over a wider dynamic range). Plotted on a log-log scale.

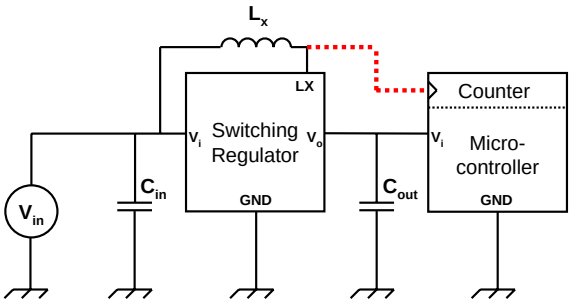

Figure 4. A typical circuit with a switching regulator and microcontroller. Adding a single wire (the dashed line) enables real-time energy metering.

appear linear over five decades, the data are linear only if the lines have unit slope since they are plotted on a log-log scale. We investigate the linearity in Section 5.

Since switchers provide a constant output voltage, the nearly linear relationship between switching frequency and load current implies a fixed amount of energy is delivered per cycle. Therefore, simply counting switching cycles approximates the total energy used over the counting interval, and dividing the number of counts by the counting interval gives the average power over that interval. However, because switchers do not expose their internal control logic and signals, one challenge is determining when the switcher actually cycles using only the signals that are observable externally. Our solution is to monitor the switch-side voltage, V_{LX}, of the inductor since it alternates between the input voltage, ground, and output voltage, as shown in Figure 2. Since the inductor voltage can exhibit ringing, care must be taken to ensure that each cycle is counted exactly once.

The basic iCount design follows directly from the data in Figures 2 and 3. Counting the rising edges of the V_{LX} signal is all that is required to accumulate energy usage. Since many battery-operated systems include a microcontroller, and since most microcontrollers support counters that can be externally-clocked, simply adding a single wire between the switcher and a the microcontroller's counter enables real-time energy metering. Figure 4 shows the changes to Figure 1 needed to implement iCount.

Returning to the topic of ringing, note that the underlying problem is that an analog power signal is being interpreted as a digital signal. In the digital domain, only high and low signals are defined while intermediate voltages are undefined. Ringing occurs when the inductor cannot deliver additional energy to the output capacitor and the two elements form a resonant circuit that oscillates. This ringing in the V_{LX} signal can be either mitigated or ignored. Some switchers, like the MAX1724 we use in this work, include an integrated damping switch that detects and eliminates the ringing. It may be safe to ignore ringing if the input and output voltages are sufficiently close since the signal would oscillate near the supply voltage rather than around an undefined midpoint voltage in the digital domain. An alternate approach is to use hysteresis on the counter's input provided by, for example, a Schmitt trigger. Fortunately, many microcontrollers have a Schmitt action on their inputs and clamping to limit input voltage excursions, so additional components may be avoided.

The basic iCount design can be enhanced by using common timer capture and compare blocks. These blocks allow specialized timing functionality without requiring the microcontroller to be active, and hence without increasing the system power draw. Capture mode is used to record the timestamp of an event relative to an underlying timer. For example, the event might be a rising edge on a microcontroller input pin that corresponds to the beginning of the discharge stage of a switcher cycle, and the timer might be the free-running system clock. By taking the timestamp of two such successive rising edges, the instantaneous load current can be estimated. The instantaneous frequency is just the reciprocal of the time difference between the two timestamps, and the instantaneous current is the frequency multiplied by an appropriate scaling factor.

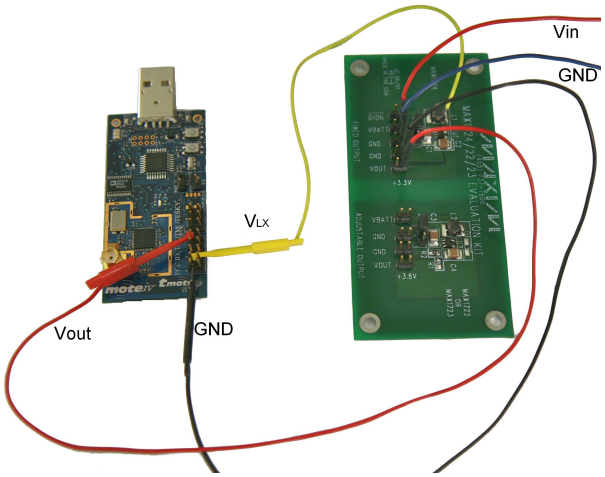

Figure 5. An illustration of how simple iCount is to implement: a Maxim MAX1724 switcher is connected to a Moteiv Tmote.

```
interface iCount {

  // Basic interface
  command error_t reset();
  command error_t start();
  command error_t pause();
  command uint32_t read();

  // Cycle-to-cycle feedback
  command error_t enableCapture();
  command errot_t disableCapture();
  async event void captured(uint32_t timestamp);

  // Quanta-based feedback
  command error_t compare(uint32_t delta)
  async event void matched(uint32_t now);
}
```

Figure 6. The iCount TinyOS API.

Compare mode is used to generate an interrupt when the value of an underlying counter/timer matches the value in the compare register. Compare mode can be used to implement an energy-based scheduler. For example, in an equal-energy operating system, each process is scheduled to run until it uses up either an energy quanta or a time quanta. If the scheduler loads the compare register with the quanta allotted to a process for its current slice just before starting the process, the compare register will generate an interrupt when the energy quanta has been used.

4 Implementation

To evaluate the feasibility and performance of our design, we implemented iCount using off-the-shelf hardware and a small amount of driver software.

4.1 Hardware

Our implementation uses the Maxim MAX1724 [12] as the switcher and the Moteiv Tmote [15] sensor platform's MSP430 as the microcontroller. We chose the Maxim MAX1724 because of its low bias and ultra-low quiescent current. The majority of our experiments are based on the Maxim MAX1724 evaluation kit. We used a 3.3 V, fixed-output switching regulator (MAX1724EZK33); a pair of 10 μF, 16 V X7R ceramic capacitors for Cin and Cout (TDK C3225X7R1C106MT); and a 10 μH inductor (Sumida CDR43-100MC).

iCount requires a dedicated hardware counter to accumulate the switcher's cycles. We use Timer A on the Tmote's MSP430F1611 [17] for this purpose, since it is normally unused. In addition to connecting the Tmote's power supply lines to the switcher's output and ground, we added a single wire between the switch-side of the inductor and the Tmote's port U2.7, as shown in Figure 5. We populated resistor R16 with a zero-ohm resistor on the Tmote to connect port U2.7 to the TAINCLK line, the external clock for the MSP430's Timer A subsystem. Using this approach, each cycle of the switcher increments the hardware counter automatically and does not require the microcontroller to execute any software. We also removed diode D22 which ensures that only the Tmote USB interface is powered by USB when plugged into a computer and the microcontroller, radio, flash, sensors, and LEDs can be powered from a different source – in this case the switcher.

4.2 Software

We implemented our driver software in TinyOS. Our driver exposes the application programming interface shown in Figure 6. The driver also handles hardware counter overflows, wraps the underlying 16-bit counter in software to increase its width to 32-bits (or more), and ensures that counter state is accessed and modified safely.

If available, iCount can use capture and compare registers attached to the dedicated hardware counter to provide additional functionality. Since a capture records the value of the timebase whenever a rising edge corresponding to a regulator cycle occurs, it can be used to measure the interval between two successive cycles, and hence the near-instantaneous current of the hardware. A compare register, in contrast, allows an interrupt to be generated when a certain quanta of energy has been used. Although the MSP430 supports these features, the capture and compare functionality is currently unimplemented.

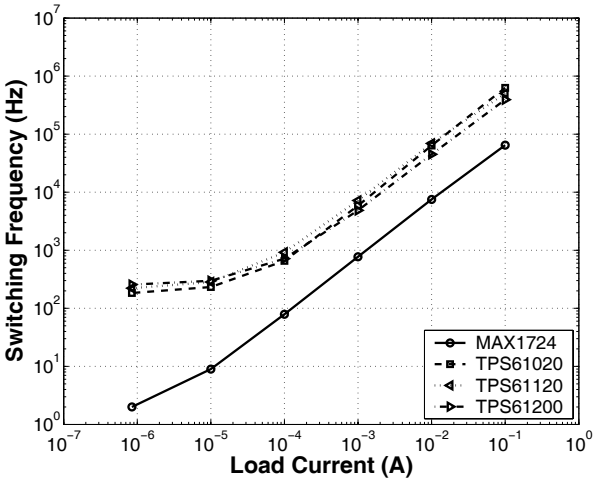

Figure 7. The relationship between load current and switching frequency without bias compensation shows significant non-linearity after three to four decades. Plotted on a log-log scale.

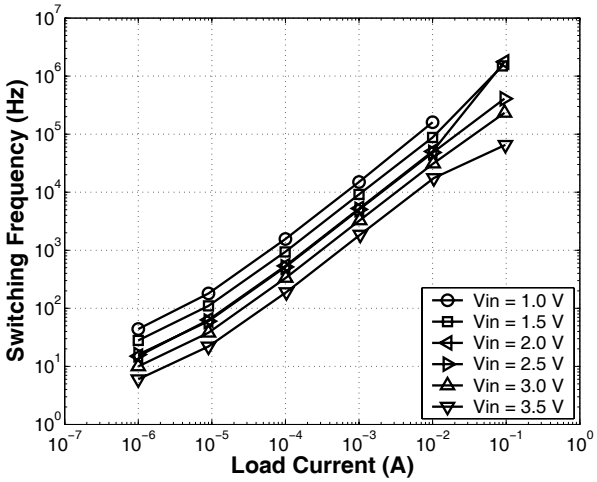

Figure 8. The relationship between input voltage and switching frequency. An order of magnitude variation in switching frequency occurs as input voltage goes from 1.0 V to 3.5 V. Plotted on a log-log scale.

Switcher	Bias	Std Dev.	Min	Max
MAX1724	1.4 Hz	0.5 Hz	1 Hz	2 Hz
TPS61020	135 Hz	1 Hz	132 Hz	138 Hz
TPS61120	213 Hz	15 Hz	187 Hz	228 Hz
TPS61200	252 Hz	1 Hz	252 Hz	254 Hz

Table 1. The no-load minimum, average and maximum bias, and its standard deviation. At least 30 samples were taken, each with a gate period of 1 second.

4.3 Calibration

Viewing iCount as an instrument, calibration is necessary to establish the relationship between the input current and output frequency. Calibration requires measuring the bias and fitting a line to a set of current-frequency measurement pairs to minimize error.

Bias, or a switcher's no-load switching frequency, must be compensated. Removing the bias is necessary to ensure high accuracy measurements at small load currents. The bias is determined by disconnecting all loads from the switcher and measuring the output frequency at the switch-side of the inductor. An Agilent Technologies 53132 Universal Counter was used to measure the bias frequency using a gate period of 1 second. Figure 7 shows the effect of bias on the switching frequency vs. load current. The no-load bias for these switchers is shown in Table 1.

The relationship between input voltage and switching frequency is shown in Figure 8. An order of magnitude variation occurs as the switcher's input voltage is swept from 1.0 V to 3.5 V. This voltage-dependent relationship can be used to adjust the meter sensitivity at runtime as battery voltage slowly changes.

Linearity describes how well a line approximates the relationship between switching frequency and load current. A linear relationship is important for at least two reasons. First, iCount will be used in systems that exhibit a wide dynamic range in power draws, so linearity across this range is needed to faithfully capture the energy usage across the operating spectrum. Second, since iCount accumulates cycles, small non-linearities can result in large errors over time. These errors are bounded by the maximum non-linearity, so this figure is important.

To calibrate our implementation, we loaded the switcher in round-robin fashion using six resistors, measured the switching frequency for each value of the load resistance, and fit a line to the data using linear regression. We used the MAX1724 switcher with input voltage of 3.0 V and output voltage of 3.3 V. The six resistors had values of 33 Ω, 330 Ω, 3.3 kΩ, 33 kΩ, 330 kΩ, and 3.9 MΩ, which provided load currents ranging from about 1 μA to 100 mA. To weight these measurements equally during regression, we first took the logarithm of the current and frequency values, performed the linear regression, and then applied the exponential to complete the process.

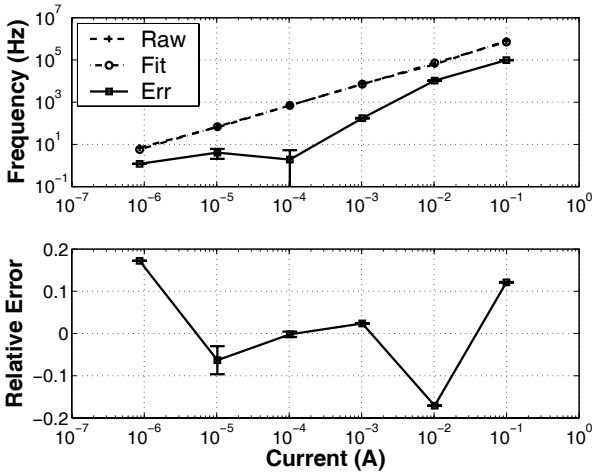

Figure 9. The accuracy and relative error. The relative error is less than ±20% over five decades of current draw. Plotted on log-log and lin-log scales, respectively.

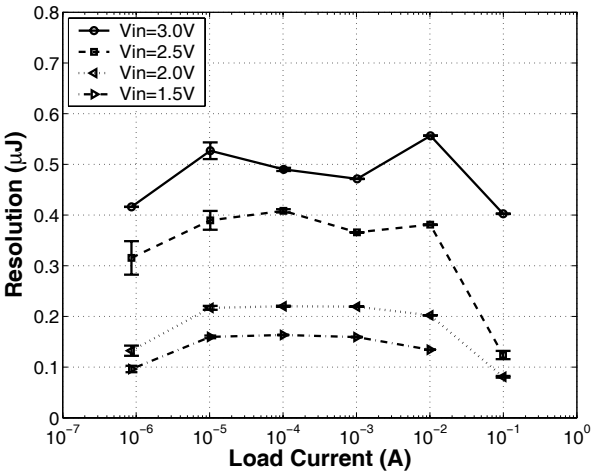

Figure 10. The resolution varies with input voltage and load current and exhibits high non-linearities at extreme load currents. Plotted on a lin-log scale.

5 Evaluation

In this section, we evaluate the performance of our iCount implementation using a set of microbenchmarks. These microbenchmarks assess the bias, accuracy, overhead, resolution, responsiveness, read latency, precision, and stability of our implementation.

5.1 Accuracy

Figure 9 shows the accuracy of our implementation using the calibration coefficients previously described. We subtract the true measurement from the line of best fit to determine the absolute estimation error. To determine the relative error, we divide the absolute error by the true value. The relative error is less than ±20% across five decades of current draw. If we consider the typical operating range of a mote, from 5 μA to 50 mA, the relative error falls within ±15% range.

5.2 Resolution

Resolution refers to the smallest quanta of energy that the system can measure. To determine our implementation's resolution, we loaded the switcher in a round-robin fashion using six resistors, measured the switching frequency for each value of load resistance, and repeated across a total of four different voltages. The six resistors had values of 33 Ω, 330 Ω, 3.3 kΩ, 33 kΩ, 330 kΩ, and 3.9 MΩ, which provided load currents ranging from about 1 μA to 100 mA. We used an Agilent Technologies 53132 Universal Counter to measure the output frequency for each value of load resistance over a 1 second gate period. A Keithley 2400 sourcemeter, configured to use 4-wire sense mode to eliminate resistive losses along the power supply lines, provided 1.5 V, 2.0 V, 2.5 V, and 3.0 V. We estimated the energy delivered during each trial by squaring the output voltage, dividing by the load resistance, and multiplying the gate time:

$$E_{trial} = \frac{V_{out}^2}{R_{load}} T_{gate} \quad (1)$$

To determine the resolution (in μJ) or sensitivity (in μJ/cycle), we divided E_{trial} by the number of cycles, minus the no-load bias, during that trial. Figure 10 shows that the resolution of the system ranges from about 0.1μJ to 0.5μJ, depending primarily on input voltage and somewhat on load current. Note the significant reduction in resolution at high load currents. The root cause has not been determined but may be due to resistive losses in the switcher's current limiting circuitry, inductor saturation, or other losses associated with the damping circuitry. Since systems usually have a safety factor in their design, and do not operate close to or at the rated limits of the power supply, these non-linearities may not affect actual measurement performance but it is important to be aware of their existence.

The meter resolution depends on the value of the switching inductor. Each cycle of the switcher delivers $\frac{1}{2}Li^2$ J, where L is the inductance and i is the peak inductor current. Our experiments are based on a 10 μH inductor but note that a smaller inductor will provide a

higher resolution, and vice versa. Also, note that manufacturing variations of $\pm 10\%$ are common, suggesting that individualized calibration is useful.

5.3 Overhead

There are three primary sources of overhead in the iCount design. First, the microcontroller-based hardware counter contributes to a fixed increase in the power draw. Second, the act of counting itself contributes a frequency-dependent increase in the power draw of the counter hardware due to gates changing state at increasing rates and the resulting charge/discharge cycles of the gate capacitance. Third, the microcontroller must service counter overflows periodically, with an average period equal to the average frequency divided by the maximum value of the hardware counter.

To determine the fixed overhead due to enabling the counter hardware, we programmed our mote with the TinyOS `Null` application, which forces all of the hardware components into their lowest power states. We powered the system using a Keithley 2400 sourcemeter configured to use 4-wire sense mode, to detect and eliminate any resistive losses in the power supply lines. Using a Fluke 189 digital multimeter, we measured the `Null` app current draw at 8.87 μA (using the DC-μA range setting) and 0.006 mA (on the DC-mA range setting), both averaged over one minute. This experiment established the baseline current draw of the mote.

We then added the iCount driver logic to the `Null` application, added a wire between the switch-side of the inductor and the counter clock input. Repeating the measurements, we found the current draw to be *the same* 8.87 μA as before via the microcontroller's power supply lines. But, we also found that a 90 nA current was being supplied by the switcher via the wire we added. This shows that the fixed overhead in our implementation is about 1% (90 nA/8.87 μA) when the system is in its lowest power state.

We also found that in some cases, the current draw via the power supply lines actually decreased, suggesting that some power was supplied via the counter wire. Figure 11 shows the overhead current across four decades of load current. The total current, I_{total}, is slightly lower than the current passing through the wire connecting the switcher and microcontroller, I_{wire}, because the total includes the decrease in current draw via the microcontroller power supply lines.

The iCount counter is implemented using the MSP430 microcontroller's 16-bit Timer A. Each time this hardware counter overflows, the microcontroller must increment a larger-width software counter by the maximum value of the hardware counter. Since this process requires writing the value of a variable in interrupt

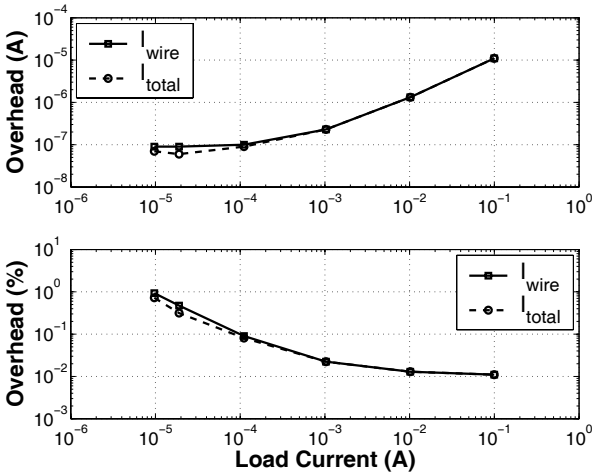

Figure 11. The overhead ranges from 0.01% for a 100 mA load to 1% for a 10 μA load, with a 3.0 V input. Plotted on a log-log scale.

context, the write should be protected with an atomic section. All interrupts are turned off during the atomic section but since the hardware counter is clocked asynchronously, no cycles are missed during execution of the interrupt handler.

Since an overflow occurs after 65,536 cycles, and each cycle delivers approximately 0.5 μJ, the counter overflows once per 32,768 μJ of energy consumed. A typical mote draws about 25 mA at 3 V, when active, and would therefore draw about 750 μW running at a 1% duty cycle. At this 1% workload, the counter would overflow every 43 seconds (32,768 μJ/750 μW). In contrast, when fully active at a 100% duty cycle, the counter would overflow less than three times per second, making the overhead of handling counter overflows negligible.

5.4 Read Latency

Application software reads the hardware counter by calling `iCount.read`. This function, executed in an atomic section, disables the hardware counter, takes a snapshot, reenables the hardware counter, performs a 32-bit addition, and returns. The total time required to invoke this function, from call to return, is 15 μs. With this read latency, the counter could be read at over 66 kHz. We determined the read latency by setting an I/O pin on the microcontroller immediately before the call, setting a second I/O pin immediately after the call, and taking their difference. The overhead of setting an I/O pin was measured by setting two pins in succession, measuring their difference (1 μs), and subtracting this value from the prior uncalibrated measurement (16 μs).

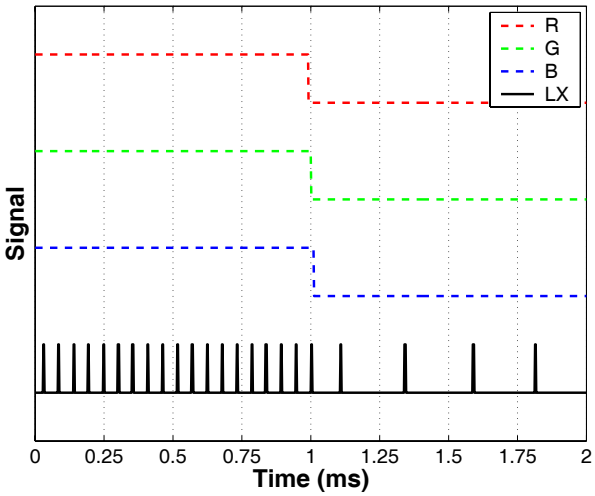

Figure 12. The system responds quickly to changes in the load current. The system adjusts in less than 125 μs to a load current decrease from about 10 mA to 2 mA.

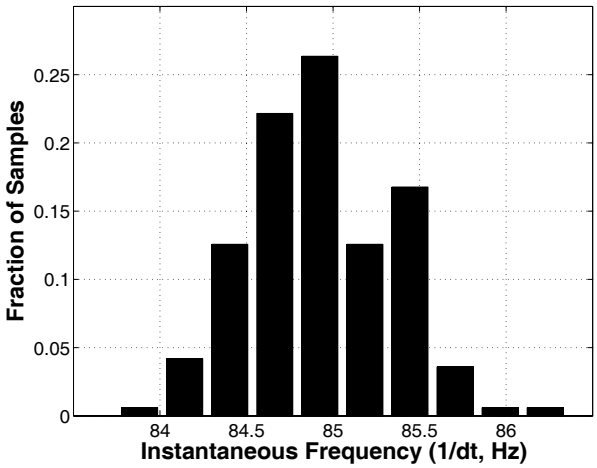

Figure 13. Measurement precision is better than $\pm$1.5%. The distribution of instantaneous (cycle-by-cycle) frequency measurements over a two second window.

5.5 Responsiveness

Responsiveness is a measure of how quickly the switching frequency settles when the load changes. To illustrate the responsiveness of the system, we instrumented the TinyOS `Blink` application and monitored the state of the red (R), green (G), and blue (B) LEDs as well as the cycles of the switcher. The greatest change in the power draw of this application occurs when the three LEDs are all turned off simultaneously. Figure 12 shows the change in the cycle rate of the switcher for a two millsecond period centered at the point when the LEDs are turned off. We see that the switching frequency adjusts to the new rate in less than 125 μs as the current draw falls from approximately 10 mA to 2 mA.

5.6 Precision

Precision characterizes the degree of mutual agreement among a series of individual measurements, or the ability to produce the same result given the same input conditions. To measure the precision of our implementation, we loaded the switcher with a fixed resistor and captured the inductor switching waveform over a two second window. This waveform was post-processed to extract the cycle-to-cycle switching period, and its reciprocal, the instantaneous frequency. Figure 13 shows the distribution of these frequency values ($N = 167$). The variations in the values are small, within $\pm$1.5%, so a small number of samples provides an accurate estimate of the near-instantaneous current.

5.7 Stability

While precision characterizes the degree of mutual agreement among a series of individual measurements over the short term, stability refers to the agreement of these readings over a much longer term. To evaluate the stability of our implementation, we loaded iCount with a fixed resistor and used a mote-based application to track the switching frequency over a one week period. The measurement setup was placed near a window and experienced daily temperature variations of more than 10° C. Figure 14 illustrates how the switching frequency changed over the course of the week and Figure 15 shows the distribution of these values. These figures shows that the range in variation falls within $\pm$1% of the mean.

The data show both daily fluctuations in frequency as well as a one-time "warmup" period. The daily variations may be due to temperature- or humidity-dependent changes in the inductance or temperature-dependent changes in the measurement system (an uncalibrated 32 kHz crystal oscillator was used). The one-time warmup period may be an artifact of the experimental setup: the iCount system was located near a laptop computer that vented warm air.

It might seem odd that our implementation exhibits a stability of $\pm$1% over one week while only exhibiting a precision of $\pm$1.5% over a two second window. One plausible explanation is that for the precision measurements, we presented the instantaneous (or cycle-by-cycle) frequency but for the stability measurements, we averaged over a one second period.

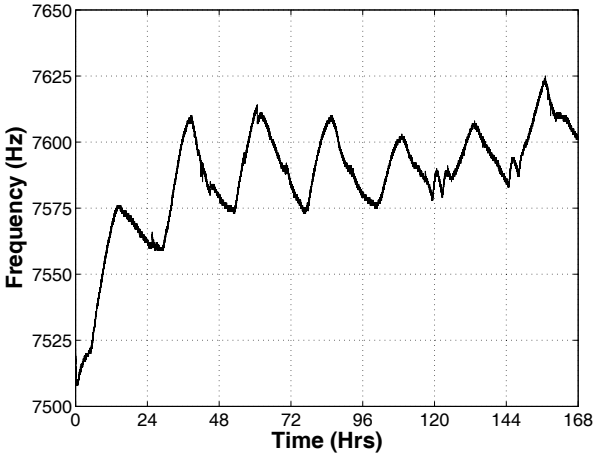

Figure 14. The system is stable. The range in variation falls within $\pm$1% of the mean over a one week period for a fixed load.

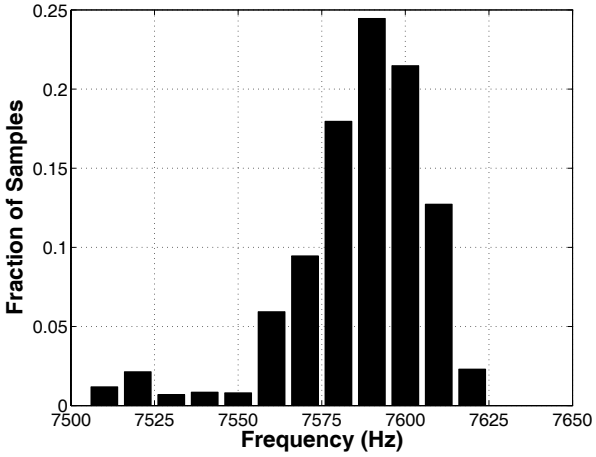

Figure 15. A histogram of the observerd switching frequencies over one week for a fixed load.

6 Discussion

Having presented and evaluated the iCount design, we now discuss some of the new opportunities it enables as well as some challenges that merit further study.

6.1 Opportunities

Hardware profiling. A common requirement when evaluating a new platform is to generate its hardware power profile [6]. Traditional techniques use oscilloscopes or digital multimeters but correlating node activity with power traces is cumbersome and difficult to automate. In addition, the read latency of a meter influences how well-correlated the measurement is to the underlying phenomenon. Using iCount, these measurements can be automated by simply running a benchmark application. In addition, variations from node-to-node or across wide supply voltages can be faithfully captured in a manner that is impossible today.

Gas gauge. A battery gas gauge is an indicator of the amount of energy already drawn from the battery and, if the battery capacity is known, the remaining energy. Since iCount increments a counter each time a fixed amount of energy is consumed, it would be simple to add a gas gauge accessible to application software. One wrinkle would be adjusting for the regulator inefficiency which is non-linear over the regulator operating range.

Model validation. A model is often used to predict the energy usage and lifetime of a network but earlier work shows that these models can be inaccurate. Efforts to validate the energy usage models *in situ* are difficult because earlier meter designs draw too much power to be practical – the instruments would dominate the power budget – but iCount is sufficiently low-power that it could be deployed alongside the application. This would finally allow model validation in realistic environments and at scale.

Software energy profiling. Much like typical software profileres track where the CPU cycles go [3], the goal with software energy profiling is to understand where the joules go, helping to identify energy hotspots [5]. This requires periodically sampling the program counter and energy counter. Earlier implementation required an expensive, external DMM with triggering and serial I/O [8].

Real-time current metering. Since a PFM switching regulator's switching interval is inversely proportional to the "instantaneous" current draw, measuring this interval allows a system to determine its own current draw. Measurements can be taken in two ways: measure the elapsed time between two (or more) cycles or measure the number of the cycles over a time interval (gate period).

Runtime adaptation. By monitoring the energy (supply and) demand, applications can select the appropriate tradeoff between energy conservation and application quality [8]. For example, on a micro scale, the system can impose limits on energy usage over short intervals (e.g. equal-energy scheduling). On a macro scale, nodes can adapt the routing tree, duty cycle, or other parameters to meet lifetime goals based on average power over a moving window or remaining battery energy [7, 10].

6.2 Challenges

Calibration. Calibration may present the greatest challenge in adopting iCount. Inductor tolerance, temperature and humidity variations, and changes in the regulator input voltage can all affect system sensitivity. Each regulator cycle stores $\frac{1}{2}Li^2$ joules in the inductor. The energy stored depends on the inductance, L, which can vary due to manufacturing tolerances and temperature and humidity changes. The actual energy stored in the inductor also depends on the inductor saturation current, which is the value above which energy is no longer efficiently stored in the magnetic core, and the regulator state machine (the MAX1724 has a 5 μs charging timeout per cycle). These factors must be considered when selecting an inductor and calibrating the system.

The node battery voltage is a function of battery temperature over short time frames and slow drooping over long time frames, this variation should be measured and compensated. We do note, however, that even over a one week period in which the system experienced over a 10° C variation in temperature, the output changed by less than $\pm 1\%$. Such errors may be negligible for some applications but might require periodic calibration for others. The battery droop over long periods is worse for alkaline batteries than it is for Lithium batteries, which have a flatter discharge curve, but since the time scales are long, it is possible to deal with these variations by taking occasional measurements and compensating.

The meter sensitivity (μJ/Hz) changes with input voltage. This can affect accuracy under high current drains, and introduce quantization errors, when operating from batteries for long periods of time. Under light current loads, the system should operate without problems but under prolonged heavy currents, I^2R losses in the battery output impendance and power supply lines can cause transient input voltage depressions which momentarily increase the resolution of each cycle.

Constant power operation. Switching regulators are constant-power devices. This means that the supply current *increases* as the supply voltage *decreases*, accelerating battery depletion at low voltages. In contrast, for a linear (resistive) load, direct connection to a battery implies proportional-current: current draw is proportional to the voltage, so the supply current decreases as the supply voltage decreases – exactly the opposite behavior from a switching regulator. It may be possible to force a switching regulator to have Vout track Vin (and hence make it a proportional-current device) by adjusting the regulator's feedback signal. If possible, there would be two benefits: first, a regulator could be used to track energy in systems which would not otherwise need one and second, the regulator precision/sensitivity would not be dependent on input voltage.

Efficiency. The efficiency (P_{out}/P_{in}) of many switchers varies widely as a function of input voltage and load current; no single input voltage is the most efficient across all load currents. Although incorporating a regulator into the power supply decreases efficiency, a regulator does allow the node to operate over an extended range of battery voltages, but at increased current draws. One practical solution to sidestepping inefficiencies at very small load currents is to simply turn off the regulator when the device enters a sleep state. Many modern microcontrollers can operate the core at very low voltages and currents, and primarily use a regulator to power peripherals like flash memory, analog electronics, or sensors which may require higher voltages.

7 Conclusion

Because energy is a critical resource in micro-power embedded systems like wireless sensor networks, much attention has been paid in the research community to energy-efficient and power-aware system designs. Typically, however, these designs are not evaluated either *in situ* or at scale. This is because traditional instrument-based power measurements, which are useful for design-time laboratory testing, are impractical for everyday *in situ* use due to the cost of instruments, their physical size, and their poor system integration. Dedicated power metering hardware can enable run-time adaptation but this approach results in increased hardware costs and power draws.

To address these shortcomings, we have designed, implemented, and demonstrated iCount, a practical energy meter design for *in situ* profiling and run-time adaptation in low-power systems. The iCount design is based on the simple observation that many switching regulators cycle at a frequency proportional to load current over many decades. Therefore, simply counting the cycles of certain switching regulators provides a measure of the energy usage. Since switching regulators are common in many battery-operated systems, and iCount requires no additional hardware beyond the existing regulator and a spare microcontroller counter, we believe our approach holds promise as a practical solution to many energy metering problems, both at bench scales and in larger deployments. iCount should simplify hardware power profiling, enable battery gas gauges, allow empirical validation of sensornet energy models, support real-time current metering, and provide the framework for software energy profiling and runtime adaptation. We show that with the addition of a single wire, iCount enables a device to introspect its own energy usage with virtually no cost or energy overhead.

8 Acknowledgments

This material is based upon work supported by the National Science Foundation under grants #0435454 ("NeTS-NR") and #0454432 ("CNS-CRI"). This work was also supported by a National Science Foundation Graduate Research Fellowship as well as generous gifts from Aginova Inc., Hewlett-Packard Company, Intel Research, Microsoft Corporation, and Sharp Electronics. The authors would also like to acknowledge the support of the Things that Think Consortium, as well as all the sponsors of the MIT Media Laboratory.

References

[1] Analog Devices. Single Phase Multifunction Energy Metering IC with di/dt Input. http://www.analog.com/UploadedFiles/Data_Sheets/ADE7753.pdf, Aug. 2003.

[2] Analog Devices, Inc. Digital Power Monitor with Convert Pin and ALERTB Output. http://www.analog.com/UploadedFiles/Data_Sheets/ADM1191.pdf, Apr. 2007.

[3] J. Anderson, L. Berc, J. Dean, S. Ghemawat, M. Henzinger, S. Leung, D. Sites, M. Vandevoorde, C. Waldspurger, and W. Weihl. Continuous profiling: Where have all the cycles gone. Technical Note 1997-016, Digital Equipment Corporation Systems Research Center, 1997.

[4] A. Benbasat. *An Automated Framework for Power-Efficient Detection in Embedded Sensor Systems*. PhD thesis, Massachusetts Institute of Technology, 2007.

[5] F. Chang, K. Farkas, and P. Ranganathan. Energy-driven statistical profiling: Detecting software hotspots. In *Workshop on Power-Aware Computer Systems*, Feb. 2002.

[6] K. I. Farkas, J. Flinn, G. Back, D. Grunwald, and J. M. Anderson. Quantifying the energy consumption of a pocket computer and a Java virtual machine. *SIGMETRICS Perform. Eval. Rev.*, 28(1):252–263, 2000.

[7] J. Flinn and M. Satyanarayanan. Energy-aware adaptation for mobile applications. In *Symposium on Operating Systems Principles (SOSP'99)*, pages 48–63, 1999.

[8] J. Flinn and M. Satyanarayanan. Powerscope: A tool for profiling the energy usage of mobile applications. In *WMCSA '99: Proceedings of the Second IEEE Workshop on Mobile Computer Systems and Applications*, page 2, Washington, DC, USA, 1999. IEEE Computer Society.

[9] X. Jiang, P. Dutta, D. Culler, and I. Stoica. Micro power meter for energy monitoring of wireless sensor networks at scale. In *Proceedings of the 6th International Conference on Information Processing in Sensor Networks (IPSN '07)*, pages 186–195, New York, NY, USA, 2007. ACM Press.

[10] S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong. The design of an acquisitional query processor for sensor networks. In *Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data (SIGMOD '03)*, pages 491–502, New York, NY, USA, 2003. ACM Press.

[11] M. Malinowski, M. Moskwa, M. Feldmeier, M. Laibowitz, and J. A. Paradiso. CargoNet: A low-cost micropower sensor node exploiting quasi-passive wakeup for adaptive asychronous monitoring of exceptional events. In *Proceedings of the 5th ACM Conference on Embedded Networked Sensor Systems (SenSys '07)*, Nov. 2007.

[12] Maxim Integrated Products. 1.5uA IQ, Step-Up DC-DC Converters in Thin SOT23-5. http://datasheets.maxim-ic.com/en/ds/MAX1722-MAX1724.pdf, July 2001.

[13] Maxim Integrated Products. DS2438 smart battery monitor. http://datasheets.maxim-ic.com/en/ds/DS2438.pdf, July 2005.

[14] Microchip. Energy metering ic. http://ww1.microchip.com/downloads/en/DeviceDoc/21948c.pdf, Aug. 2005.

[15] J. Polastre, R. Szewczyk, and D. Culler. Telos: Enabling ultra-low power research. In *Proceedings of the 4th International Conference on Information Processing in Sensor Networks (IPSN '05)*, Apr. 2005.

[16] Texas Instruments. Advanced battery monitoring ic. http://www.ti.com/lit/gpn/bq2019, Feb. 2003.

[17] Texas Instruments. Mixed signal microcontroller (rev. e). http://www.ti.com/lit/gpn/msp430f1611, Aug. 2006.

[18] Texas Instruments. System-side impedance track fuel gauge. http://www.ti.com/lit/gpn/bq27500, Oct. 2007.

[19] T. Trathnigg and R. Weiss. Towards runtime support for energy awareness in wireless sensor networks. In *Proceedings of the Workshop on Wireless Sensor Networks*, June 2007.