Prospector: Multiscale Energy Measurement of Networked Embedded Systems with Wideband Power Signals

Kenji R. Yamamoto and Paul G. Flikkema
Department of Electrical Engineering
Northern Arizona University
Flagstaff, Arizona 86001 USA
Email: {kry3,paul.flikkema}@nau.edu

Abstract*—Today's wirelessly networked embedded systems underlie a vast array of electronic devices, performing computation, communication, and input/output. A major design goal of these systems is energy efficiency. To achieve this goal, these systems are based on processors with numerous power and clock domains, variable clock rates, voltage scaling, and multiple hibernation states. These processors are designed into systems with sophisticated wireless transceivers and a diverse array of off-chip peripherals, and are linked through regulators to increasingly complex energy supplies. As a result, modern networked embedded systems are characterized by myriad power consumption states and significant power signal transients. Moreover, their power demands are multiscale in both magnitude and time, combining short bursts of high demand with long intervals of power-sipping sleep states. Thus the power supply signals have wideband spectra. In addition, due to noise, uniform relative precision across magnitude scales requires that measurement time increases with decreasing power. Tools are needed that support modeling, hardware/software optimization, and debugging for energy-centric embedded systems. This paper describes Prospector, an energy data acquisition system architecture for embedded systems that allows rapid, accurate, and precise assessment of system-level power usage. Prospector uses a distributed control architecture; each component contributes efficiently to control, precision and accuracy, analysis, and visualization. It is based on computer-based control of multimeters to maximize accuracy, precision, flexibility, and minimize target system overhead. Experimental results for a prototype Prospector system with a contemporary 16-bit ultra-low power microcontroller show that it can effectively measure power over the extreme time and magnitude scales found in today's embedded systems.

I. Introduction

A large percentage, if not the vast majority, of today's electronic devices are based on wirelessly networked embedded systems. Most of these systems tend to be portable and/or untethered to line power, and tethered systems are now expected to use minimum energy in numerous sleep and active states. Modern embedded systems integrate computation with sensing of the environment, are often part of networked, distributed systems, and usually energy-constrained. Moreover, in wirelessly-networked devices, a large portion of the energy budget is absorbed by communication costs. In these systems, the technological and human cost of energy and replenishment of energy supplies are dominant. Wireless sensor networks are paradigmatic examples of these systems. They are distributed instruments that are purposefully embedded in their environments, from which they acquire information. This information can range from detecting hazards and triggering an alarm response to data collection for inference of complex multivariate models. For these reasons, this paper uses wireless sensor network nodes as representative embedded systems.

With respect to energy efficiency, wireless sensor network (WSN) nodes are at the leading edge of energy-efficient networked embedded systems design. A sensor network must be designed to accomplish its tasks efficiently in terms of life-cycle cost. While installation, periodic transducer calibration, maintenance and recovery involve significant cost, perhaps the largest component of life-cycle cost is the labor to sustain the energy supplies of the network nodes, so battery lifetime is a primary design metric. The arrival of energy harvesting technologies increases this need, since maximizing energy efficiency makes them more technologically and economically feasible.

This paper uses the WSN application as a motivating example for the development of energy-centric analysis, design, test, and evaluation of networked embedded systems. While a great degree of effort is on-going in the modeling and optimization of the raw communication energy efficiency in sensor networks, understanding of the complete node as an energy-consuming agent is in its infancy. Each node can be considered a micro-scale power grid consisting of multiple consumers, including a microcontroller, one or more radio transceivers, memory chips, I/O peripherals and an array of transducers and/or actuators and their mixed-signal interfaces. Microcontrollers now have multiple sleep modes and numerous power and clock domains, so that power dissipation can vary over several orders of magnitude for just this device. As a result, the power consumption profile of a sensor node is the result of an enormous number of temporally-dependent and parameterized functional states. The complexity of networked embedded systems is growing rapidly in support of both greater functionality and energy efficiency.

Wireless sensors, like many other networked embedded systems, also have at least one power source, normally a battery pack that is itself a complex non-linear system whose state-of-charge and lifetime is a function of its current-voltage history. Between the power source and the consumers is a regulator, a control system that must deliver power over several orders of magnitude at a stable voltage level in the face of fast transients in demand. If the power supply includes energy harvesting and rechargeable power sources, then the regulator becomes a complex multi-input multi-output power manager.

For these reasons, the consumers, power supply, and regulator form a power distribution network of interacting time-variant non-linear dynamical systems that operate at diverse scales of current magnitude and time; a typical sensor node can spend milliseconds consuming over 20 mA followed by days where consumption is less than 1 μA. Hence power demands are multiscale in both time and amplitude.

II. Energy-Centric Design of Embedded Systems

Embedded systems software design is a complex enterprise; in the last decade, numerous approaches have been proposed to improve productivity. Energy-centric or "green" design only adds to this complexity; clearly, better methods are needed than "power debugging" after a functional design process that only informally captures energy efficiency. Specific aims of energy-centric design include battery lifetime of a node; network lifetime; optimization; feasibility of energy harvesting systems; and assessment of sleep strategies, including transient costs. Here we outline a five-phase iterative design process.

The first phase is the design of an energy consumption model for the system that balances simplicity and good prediction of energy consumption. The next phase, measurement, requires observation of a node's power consumption state and its integral, energy, leading to inference of models for short- and long-term energy consumption and battery lifetime. The next phase, model inference, is followed by testing and model verification, which also require measurement. Once models are developed and verified, the final phase is energy management and optimization. There are two basic approaches to optimization of energy consumption. In a fully on-line approach, the system autosenses—monitors itself—and adapts in real-time, which requires overhead in the form of hardware, software, and energy. Fully off-line approaches construct models of energy consumption and optimize the design prior to deployment, using fewer runtime resources but providing less flexibility. Except in extreme cases (e.g., where code space is so limited that only off-line approaches are practical), practical embedded systems will employ a hybrid approach. The number of potential solutions is essentially infinite; one choice that we are investigating is detailed measurement and modeling in the design and prototype phases, augmented by low-cost monitoring of battery voltage and simple on-line algorithms in deployments.

The measurement phase of this process requires an energy data acquisition system (EDAQS) to measure system power and energy consumption. Since the focus is energy (not just power), EDAQ systems must sense power as a function of time. The Prospector energy measurement architecture is general and can be applied to any embedded system. The ultimate goal—minimizing energy consumption—is different from recent approaches in power minimization techniques for high-end embedded systems such as data and storage servers, including power-aware scheduling, where the transient costs of power state switching can be neglected in some cases.

Numerous EDAQS have been proposed that require dedicated, integrated hardware or software components that are not easily removable; for example, embedded systems have been proposed that include current-sense resistors and use of on-chip or on-board analog-to-digital converters (ADCs). We simply refer to these systems as built-in.

III. Requirements

Understanding the energy efficiency and lifetime of networked embedded systems requires an EDAQS for measurement of the sensor node's power grid, followed by inference of system power states (and transitions between them), and optimization of power state trajectories for functional state sequences. To tackle the design phases requiring measurement, this paper describes an approach called Prospector, a current, power, and energy measurement approach that is designed to enable inference of accurate power consumption models that enable estimation of energy efficiency and node lifetime. Our goal is to make accurate measurement part of an automated workflow that designers can use to evaluate hardware and software designs and optimize the efficiency of sensor node hardware, firmware, and software. Prospector is a distributed controller/observer architecture for power and energy measurement with a human interface that enables automated multi-resolution measurements of time and current, and can be scaled to monitor multiple subsystems and even multiple nodes. It was designed to have the following properties:

Ability to characterize power usage in any embedded system. If embedded systems are anything, they are diverse; processor packages range from a handful of pins to hundreds, clocks from a few KHz to GHz, and they offer dozens of different on-chip peripherals. Energy supplies include batteries and a growing number of in situ energy scavenging options, and voltage regulators can be linear or switching. At the software level, many low-cost systems are based on elaborations of event loops; simple operating systems may be time-triggered, event-driven, and more complex operating systems are process-oriented and priority-driven. To extract maximum energy efficiency, it may also be useful to use RAM shadowing or caching of FLASH-resident system or application code. Finally, almost every embedded system design includes a unique complement of off-chip peripherals: volatile and non-volatile memory, digital I/O, and analog transducers and actuators.

Accurate measurement of power in all system states. There are two fundamental barriers to good models of power and energy use in embedded systems. First, the current usage of typical energy-autonomous systems in different states spans a range of 0.1 μA to over 50 mA—almost six orders of magnitude. Secondly, power consumption is a wideband process, since the time spent in states also varies. For example, a networked embedded device can spend dozens of microseconds consuming tens of milliamps responding to an event and several milliseconds transmitting a radio burst, followed by days of deep hibernation using microamps. The absolute (not relative) precision and accuracy must scale with the current magnitude, since long sleep intervals can dominate energy usage in many applications. This implies that the measurement range must be a function of the magnitude.

Measurement of current usage in ultra-low power states is prone to biases, since the act of power measurement consumes power. This makes measurement using in-system resources (e.g., an analog-to-digital converter) problematic. While this issue can be managed in these methods by careful experiments and compensation, there is a deeper problem: in some states, the entire processor might be asleep, preventing use of required measurement resources.

Precision. In the system power/energy characterization literature, usage of the terms resolution, accuracy and precision often differs, causing confusion. Resolution is the ratio of the measurement range and the number of discrete bins (or equivalently, digits) of the quantized measured value. Accuracy, or bias, describes the average difference between that value and the true value, and precision refers to the estimated variation of measured values about their mean. Put simply, high resolution guarantees neither high accuracy nor high precision, so that the number of effective digits (the effective resolution) is often less than the total. On the other hand, resolution often limits accuracy in EDAQS that use low-cost on-chip ADC's. To be descriptively precise in this paper, we use fidelity to refer to the combination of accuracy, precision, and resolution. Due to the high refinement level of contemporary analog-to-digital converter (ADC) designs, accuracy is not a serious issue in high-quality modern digital multi-meters (DMMs). The primary barrier to precise measurements is noise. In laboratory environments, the greatest noise source is 50 or 60 Hz electromagnetic interference from AC line power, which often is greater than thermal noise. While it is a predictable signal and hence not truly noise, it is a sum of a random number of signals with random amplitude and phase. This challenge is reflected in the test and measurement industry's use of the *power line cycle* ($\frac{1}{60}$ second in the US) as the standard time unit for the interval over which a measurement is taken. Maintaining relative precision over magnitude scales requires a communication-theoretic viewpoint in which the signal to be measured is integrated over an aperture T. Alternatively, if the signal is sampled instantaneously, multiple samples can be averaged. To maintain the same immunity to noise (i.e., the signal-to-noise ratio, or SNR) when measuring a 1 mA signal, a 1 μA signal requires an aperture 1000 times longer. In slope ADCs, the precision can be increased linearly with the integration time, or aperture T, prior to sampling, or by averaging N ADC samples (the noise variance is $\propto T, N$). Only the post-conversion option is available in instantaneous (or pure sampling) ADCs. Maintaining precision thus requires that the aperture or number of samples scales inversely with the magnitude of the current. Fortunately, deeply embedded systems tend to use small currents for comparatively long periods of time.

A conflicting requirement is measurement speed. If an instruction takes 0.02 microseconds to execute, what measurement rate is needed to capture the energy use of a function comprising just a few instructions? Evidence [1] gathered using a high-speed oscilloscope indicates power signal bandwidth is significantly less than the processor clock frequency; this is likely due to the combination of supply bypass capacitors, local bypass capacitors across chip supply and ground pins and the natural filtering of voltage regulator circuits.

Minimal target resource usage. Any power/energy measurement system incurs additional hardware in or attached to the target system (see Section VII). These approaches, when used on-line, also invoke software overhead to manage the ADC. This overhead has two classes of energy cost: the energy to simply power the hardware resources, which is relatively easy to characterize, and the energy for the software to configure and manage the resources, which is more difficult.

Hard real-time systems are particularly sensitive to this issue. In some cases, it may be possible to use compile switches to add/remove measurement code. However, since removal of measurement code can perturb the temporal behavior of the system, there may be no choice but to make this code a permanent feature to smooth the test and verification process. In this case, the energy and memory footprints of the code should be minimal.

Measurement Efficiency. One of the primary goals of Prospector is to enable convenient, comprehensive characterization of power profiles in support of modeling and design. Many current power measurement techniques require hand-coded task-level proxies of system- or application-level tasks (using loops to remove start-stop edge effects), and time-consuming manual power measurement procedures.

Minimal need for calibration. Many built-in EDAQ systems are limited by calibration challenges of two classes. First is the ability to accurately calibrate the design, meaning anything that requires a one-time measurement of the finished prototype. Second is unit-level calibration, where component variations force calibration of every unit. In both cases, a good EDAQS should not be based on current measurement designs for which it can be difficult to fully characterize biases or measurement noise.

Flexibility and Scalability. A EDAQS should be easily reconfigurable for multiple uses. In addition to characterizing the energy of power signal, it should also be useful for determining the quality of a reference signal (e.g., for an ADC). A good measurement solution should also make it easy to scale up to multi-signal current-voltage characterization for analysis of power flow in complex embedded system power grids. For example, it should be straightforward to characterize multiple currents and voltages to assess voltage regulator efficiency, or radio power usage as a function of transmit power control settings. Ideally, an EDAQS should also be quickly and completely removable, to eliminate all overhead

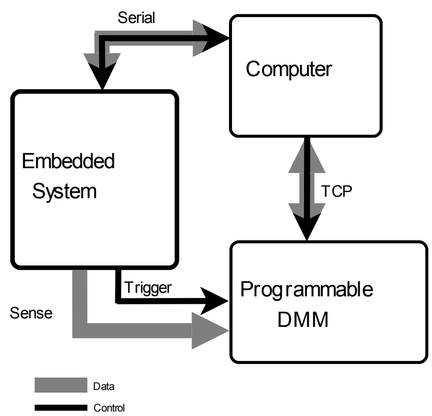

Fig. 1. Hardware block diagram of Prospector. High-fidelity energy measurements are enabled by integrating the native time scale of the embedded system under test with high-quality calibrated DMM measurements, managed by the computer workstation.

when required for optimum energy efficiency or minimum code space.

IV. The Prospector Architecture

Prospector is designed to relieve the target embedded system of all but the minimum hardware and software needed for energy measurement. It uses hardware common across nearly all microcontrollers, and exploits capabilities of new measurement technologies. As shown in the block diagram of Figure 1, Prospector consists of three hardware components: a programmable digital multimeter, a workstation consisting of a desktop or notebook computer, and the embedded system (or SUT—System Under Test). It is designed to measure time and current with high fidelity across scales from 20 microseconds to weeks and from single-digit μA to hundreds of mA.

Multimeter. Prospector achieves high-speed, high-fidelity current measurement by offloading the electrical measurement duties to a programmable DMM that meets the requirements of precision, accuracy, and resolution. Modern multimeters provide measurement ranges from 1 A down to 100 μA for accuracy, and apertures from 20 μs to 1 s for precision. Programmability means they can be dynamically configured remotely via IEEE-488, USB, and Ethernet LAN to optimize range and aperture. They can also be triggered externally, so that timing of measurements can be controlled by the SUT at its native time scale. Two meters that meet these requirements are the Agilent 34411A and Fluke 8845A/8846A. Both of these instruments allow autoranging to be turned off, improving measurement speed. Some also provide a pretriggering capability that can be used for capture of power transients occurring before the triggering event. Use of DMMs also removes the difficulty of calibrating the current transducer, since the DMMs can be factory-calibrated.

A standalone DMM—even a high performance unit—does not provide the needed capability for accurate and comprehensive measurement of embedded systems power signals. Autoranging is slow and invisible to the user, causing measurement gaps and unknown precision. The alternative, manual setting of the range, would be possible only if currents did not have wide bandwidth. Rapid changes over multiple orders of magnitude require near-instantaneous DMM range changes synchronized with associated changes in the embedded systems' power demand. This can be achieved by cooperative control of the DMM by the embedded system and a workstation: the former triggers measurements at its native time scales.

Computer workstation. The primary function of the workstation is to configure and acquire data from the DMM. It also serves as the broker between the SUT and the DMM, off-loading test management from the SUT. The computer provides a highly flexible platform for iterative analysis-driven measurement where code space and energy consumption are not limited. Functions include computation of optimum range and aperture, signal processing (e.g, removal of 60 Hz interference), visualization, and storage. Top-level control of measurement at the workstation allows convenient integration of measurement with the other phases of energy-centric design (cf. Section II).

Embedded system. The embedded system, or SUT, is also a participant, but Prospector is designed to minimize measurement overhead, protecting accuracy gained by using the multimeter and workstation. The only hardware resources it uses are a timer, a serial port, and one I/O pin. It is also quickly removable to eliminate all overhead when required for optimum energy efficiency or minimum code space.

Control Flow. Prospector requires accurate and precise measurement of time and current. Time is measured by the SUT at its native scale; the SUT also triggers measurements by the DMM that are planned, managed, and evaluated by the computer, thus decoupling the actual measurement from instrument configuration.

Prospector uses code fragments in the SUT software called *probes*; they are used to (i) measure time intervals (e.g., the duration of a particular task or functional state) with an on-chip timer and (ii) trigger the DMM with a digital I/O pin. Each probe requires only 4 lines of C code. Prospector first activates the SUT to measure time intervals between probes. The Prospector computer then programs the DMM and resets the SUT to collect measurement data over specific intervals. The reset and data collection cycle iterates, beginning with the lowest measurement range on the DMM and increasing to the highest range, maximizing precision and accuracy. The data sets are then time-synchronized, merged to maximize fidelity, and written to an output file for further analysis and display.

Component Integration. Measurement of current across scales is challenging; in this system, as in any, the act of measuring perturbs the measurement. The challenge is to make this perturbation negligible or compensate for it. In Prospector, one significant problem is that changing the measurement range of current meters causes a transient open and thus a momentary power loss to the embedded system that could cause a system reset. To avoid this, the control flow uses a single DMM range during each data collection cycle. As a consequence, the DMM series resistance at low ranges could cause measurement difficulties. At its lowest (100 μA) range, the Agilent DMM we used has 200 Ω resistance. While this causes a negligible voltage drop when the system is sleeping, it can be significant in active modes. For example, if both the MCU and radio wake up to send a packet in a wireless embedded system, the current demand is at least 20 mA, causing a potential 4 V drop across the DMM resistance resulting in system brownout or blackout.

This issue emphasizes the subtle difficulties in accurately and precisely measuring embedded systems with multiscale current demand: when current demand is high and the DMM range is low, the voltage drop across the DMM series resistance could cause collapse of the power supply. However, most embedded systems of interest include a voltage regulator, and this problem is easily addressed in these systems by using a battery or power supply with enough capacity to deliver the necessary voltage and current to the regulator and embedded system through the DMM. Note that this power source can be different than what the embedded system uses in the field, and will not cause corruption of the measured data since the actual demanded current is measured, not the capacity of the power supply. Also, the series resistance causes a significant voltage drop only when the demanded current is high. This causes no corruption of the measured data in Prospector, since this data is superseded by high-range data when the datasets are merged.

We note that some embedded system platforms, such as the one we used in Section VI, rely either on regulated power from a USB interface or unregulated battery power. In working with this hardware platform, we customized the Prospector implementation by manually adjusting the voltage source in each current range so that the voltage drop over the DMM's series resistance would not affect the operation of the system.

V. Proof of Concept Implementation

Our current implementation of Prospector uses Ethernet LAN instrument control via socket TCP communication and the industry-standard SCPI (Standard Commands for Programmable Instruments) control language. This allows simple extension to multiple instruments observing a node as well as simultaneous observation of multiple nodes. Among other possibilities, this approach can simultaneously measure regulator input and output current as well as battery voltage. Higher-level protocols were avoided to expose the full control language of the instrument and allow tuning of the TCP connection to eliminate buffer delays.

Prospector's Disadvantages. The greatest objection to Prospector might be its use of an expensive programmable DMM. However, the initial cost of the instrument is probably significantly less than the labor costs of design, test, and calibration of built-in EDAQS. Approaches based on built-in EDAQS can provide network-wide data for all nodes, a big plus, but they may be compromised in fidelity due to cost, complexity, or calibration issues. On the other hand, Prospector emphasizes high-quality measurements as part of a design process (Section II). When Prospector's data is coupled with good energy models, accurate characterization at a single node can be applied to all similar nodes in a network, amortizing the cost over the entire network, or more to the point, every network using the same hardware/software configuration. Model-based inference of energy consumption can save energy relative to direct measurement at all nodes. Prospector can also be quickly and easily used to characterize nodes with heterogeneous hardware, e.g., expensive, specialized transducers.

External, rather than built-in, EDAQ systems such as Prospector necessarily imply the potential of additional noise due to longer wiring runs for the sense signal. However, careful set-up can minimize noise, and the remaining noise can be easily mitigated with a change of aperture or post-sampling signal processing.

Prospector is bulky and not easily weatherproofed, limiting its applicability to laboratory and short-term in-field measurement campaigns. However, accurate measurements combined with good modeling should allow high-fidelity on-line estimation of energy use in the field.

VI. Results

To test Prospector we chose an embedded system based on an ultra-low power (ULP) microcontroller (Texas Instruments MSP430 family) used in numerous embedded and WSN node designs, and whose power supply signals have a wide dynamic range. TI claims current draw for this MCU can be as low as 100 nA with full RAM retention in the lowest power mode. In our first test, we used a board-level system (TI MSP430FG4618 Experimenter's Board) based on the MSP430FG4618 and LEDs to model peripherals. The MSP430 was programmed with a minimalist scheduler consisting of a timer interrupt and a task list. The timer interrupt wakes up the system every second to check the task list; any scheduled task for the time slot is executed and the device returns to ultra-low power sleep. For this example, we defined two tasks: first, a counter is incremented every second and is indicated by flashing a surface-mount LED. Secondly, every two seconds the MSP430 also sends a short packet to the computer via the serial communication interface.

As described in Section IV, a digital I/O pin on the MSP430 was used to drive the DMM's trigger port. The Agilent 34411A DMM we used requires a pulse of ≥ 1 μs to initiate an internal Schmitt trigger circuit. The embedded system uses a 1 MHz system clock, so here a probe with a two-cycle delay loop is sufficient, while having a minimum impact on the operation and scheduling of the SUT software.

For this SUT, the DMM is triggered by the probes based on a four-second period. Prospector calculates the sample rates and measurement apertures to maximize the data resolution for the 4 second interval. The results (Figures 2 and 3) clearly show the active (LED enabled) states embedded in the 2 μA sleep current, as well as the additional current used by the serial port communication (central peak in Figure 2 detailed in Figure 3).

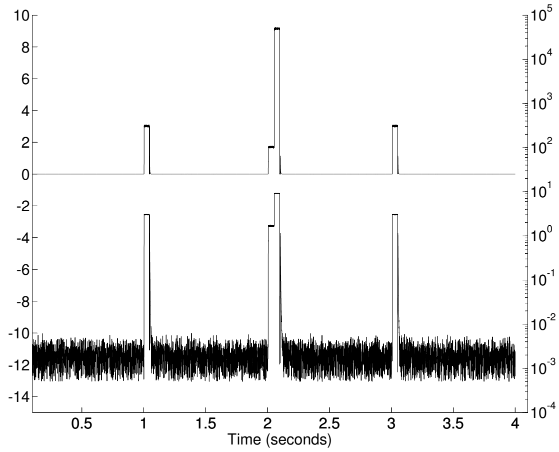

Fig. 2. Linear (upper) and log-scale (lower) current traces (in mA) of a ULP MCU-based embedded system over a 4 second interval.

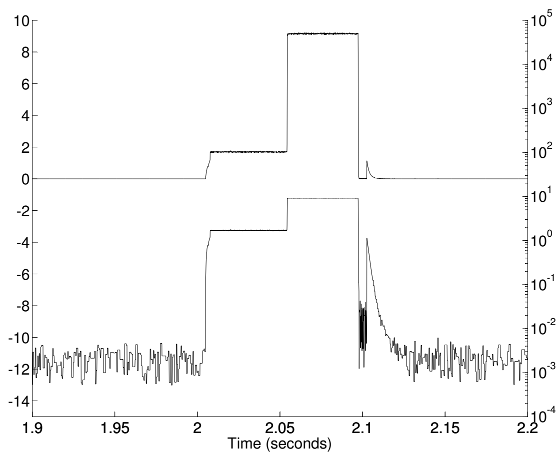

Fig. 3. Magnified view of current trace in Figure 2, showing sleep-task execution-sleep trajectory at 2 sec with almost four orders of magnitude and wideband dynamics (current in mA).

In most current embedded applications, including wireless sensor and geosensor networks, the energy cost of wireless communication is dominant. Current demands of up to 30 mA are routine, but these systems must drop into ultra-low power hibernation modes as much as possible to extend battery life or enable energy harvesting. To show the flexibility of Prospector, we also performed a second test using a wireless embedded node employing a TI CC2500 2.4 GHz ISM Radio Evaluation Module in the above-described Experimenter's board. The node is connected to a PC via serial communication.

The current measured by Prospector for this test is shown in Figure 4. In this test, the MCU and radio were programmed to sleep the majority of the time, drawing $\approx 2\ \mu$A (labeled by (5)), waking up periodically to transmit a 32-byte packet. The radio's current demand of 22 mA for packet transmission

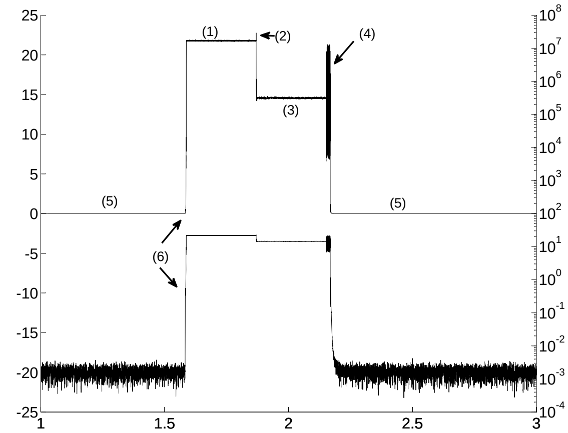

Fig. 4. Linear (upper) and log-scale (lower) current traces (in mA) of a wireless embedded node communicating with another node and a PC, showing data across four orders of magnitude. See the text for a detailed description.

is indicated by (1). During this transmission, the MCU is asleep until it is awakened by the radio chip following the transmission (2), at which point the radio is switched to receive mode and the MCU again returns to sleep mode (3). Finally, the MCU is awakened by an interrupt from the radio to indicate receipt of a packet, and the packet is decoded and sent to the PC (4). The large current fluctuations at (4) are caused by an on/off-keyed (OOK) LED that is part of board's optically-isolated serial link to the PC. The current fluctuations track the transmitted bit pattern used in this 19.2 Kb/s wired link. The plot also shows a very short spike (at (6), to ≈ 0.8 mA) just prior to radio turn-on that is caused by powering up the serial peripheral on the MCU for SPI communication with the radio, demonstrating Prospector's resolution.

VII. Related Work

Numerous low-power hardware platforms for wireless nodes (e.g., [2]), including those that specifically enable energy measurement [3], are under development. It has also been recognized that consideration of the regulator is important for understanding power efficiency in embedded systems [4]. Extensions to event-driven embedded cooperative operating systems for on-line measurement have been reported for on-line estimation [5] and management [6] of power consumption. Cycle-accurate current measurement approaches for laboratory testing of processors are described in [7], [8]. Built-in EDAQ systems typically use a current-sense resistor and an on-chip analog-to-digital converter. [9] describes a system with an add-on sense resistor and ADC at each of multiple nodes with centralized data collection. A built-in EDAQS that uses an amplifier and voltage-to-frequency converter for high-fidelity current measurements across scales is presented in [10]. [11] presents a novel power measurement approach that uses only a counter on the microcontroller; however it is dependent on the use of a switching regulator that does not automatically change switching modes. It also may require unit-level calibration due to inductor manufacturing variations. Prospector is complementary to this and all built-in and/or low-cost approaches by providing a convenient and high-quality design-stage and unit-level calibration tool. In our work, we use it exclusively, in tandem with modeling, to build energy-use models of embedded systems.

VIII. Conclusion

In this paper we introduced Prospector, an energy measurement architecture for characterization of networked devices with wideband power signals, and provided results from a prototype implementation. Recent advances in energy-efficient hardware and software are now driving the assessment of algorithms that trade the energy cost of trade of signal processing against communication energy cost in wirelessly-networked devices. Another aspect of energy efficiency ripe for optimization is the trade between information bit rate and transmit power for wireless links. It is well-known that RF power amplifier efficiency increases with transmit power; however, battery efficiency may deteriorate with high and pulsed current demands. Prospector forms a foundation for exploration of these and many other aspects of energy efficiency in embedded systems.

Acknowledgment

This work was supported by NAU TRIF ERDENE Grant EGRTA77 and National Science Foundation Grants CNS-0540414 and EF-0308498.

References

[1] R. Muresan and C. Gebotys, “Instantaneous current modeling in a complex VLIW processor core,” *Trans. on Embedded Computing Sys.*, vol. 4, no. 2, pp. 415–451, 2005.

[2] M. Hempstead, N. Tripathi, P. Mauro, G.-Y. Wei, and D. Brooks, “An ultra low power system architecture for sensor network applications,” *SIGARCH Comput. Archit. News*, vol. 33, no. 2, pp. 208–219, 2005.

[3] D. McIntire *et al.*, “The low power energy aware processing (LEAP) embedded networked sensor system,” in *IPSN 2006*, April 2006, pp. 449–457.

[4] Y. Choi, N. Chang, and T. Kim, “DC-DC converter-aware power management for battery-operated embedded systems,” in *DAC ’05: Proceedings of the 42nd annual conference on Design automation*. New York, NY, USA: ACM, 2005, pp. 895–900.

[5] A. Dunkels, F. Osterlind, N. Tsiftes, and Z. He, “Software-based on-line energy estimation for sensor nodes,” in *EmNets ’07: Proceedings of the 4th workshop on Embedded networked sensors*. New York, NY, USA: ACM, 2007, pp. 28–32.

[6] K. Klues, V. Handziski, C. Lu, A. Wolisz, D. Culler, D. Gay, and P. Levis, “Integrating concurrency control and energy management in device drivers,” in *SOSP ’07: Proceedings of twenty-first ACM SIGOPS symposium on Operating systems principles*. New York, NY, USA: ACM, 2007, pp. 251–264.

[7] N. Chang, K. Kim, and H. G. Lee, “Cycle-accurate energy measurement and characterization with a case study of the ARM7TDMI microprocessors,” *IEEE Trans. VLSI Syst.*, vol. 10, no. 2, pp. 146–154, 2002.

[8] V. Konstantakos, A. Chatzigeorgiou, S. Nikolaidis, and T. Laopoulos, “Energy consumption estimation in embedded systems,” *IEEE Transactions on Instrumentation and Measurement*, vol. 57, no. 4, pp. 797–804, 2008.

[9] I. Haratcherev, G. Halkes, T. Parker, O. Visser, and K. Langendoen, “PowerBench: A scalable testbed infrastructure for benchmarking power consumption,” in *Int’l Workshop on Sensor Network Engineering (IWSNE)*, June 2008.

[10] X. Jiang, P. Dutta, D. Culler, and I. Stoica, “Micro power meter for energy monitoring of wireless sensor networks at scale,” in *IPSN ’07: Proceedings of the 6th international conference on Information processing in sensor networks*. New York, NY, USA: ACM, 2007, pp. 186–195.

[11] P. Dutta, M. Feldmeier, J. Paradiso, and D. Culler, “Energy metering for free: Augmenting switching regulators for real-time monitoring,” in *SPOTS ’08*, April 2008.