technische universiteit eindhoven

A Mobile Client Platform for Sensor Networks

Phillip Stanley-Marbell

ES Reports
ISSN 1574-9517

ESR-2008-04
24 January 2008

Eindhoven University of Technology
Department of Electrical Engineering
Electronic Systems

People who are really serious about software should make their own hardware. — Alan Kay

http://www.es.ele.tue.nl/esreports
esreports@es.ele.tue.nl

Eindhoven University of Technology
Department of Electrical Engineering
Electronic Systems
PO Box 513
NL-5600 MB Eindhoven
The Netherlands

A Mobile Client Platform for Sensor Networks

Phillip Stanley-Marbell
Technische Universiteit Eindhoven, Den Dolech 2, 5612 WB Eindhoven, The Netherlands

Abstract

Presented is the design, hardware implementation and evaluation of a mobile computing platform that is well suited for use as an interface to wireless sensor networks. The device incorporates an 802.15.4 radio interface common in contemporary sensor network platforms, a color 320×240 pixel low-power organic LED (OLED) graphical display, an input device, and computation resources. The system includes several sensors relevant to a mobile device in a sensor network — a digital compass, multiple temperature sensors, a humidity sensor, and a pressure sensor for altimeter or barometer applications. The system incorporates these facilities, along with a rechargeable 2000 mAh lithium-polymer battery, all within a form-factor of 2.1"×4"×0.5". The platform contains many hardware facilities to support current and future research directions in wireless sensor networks and their interactions with mobile computing systems, including a low-power signal strength monitoring circuit independent of the system's main radio, support for dynamic operating voltage setting, power gating of peripherals, and built-in power monitoring circuits for power consumption introspection.

1. Introduction

A large portion of the research activity in wireless sensor networks has focused on issues internal to the network. These include various research problems pertaining to the nodes that make up the network, low-power sensing devices, the medium-access, network and transport protocols for their interconnection, systems software, and algorithms for facilities such as localization and aggregation. The implicit assumption of the mode of access to such networks has been that users will employ a device such as an off-the-shelf personal digital assistant (PDA), mobile phone, laptop computer, or workstation, as the *sink* of network traffic. Such a sink is also often assumed to provide certain facilities to the network at large, such as being the location of implementation of congestion control mechanisms [29], initiating the setting-up of routing tables by the injection of requests [32], or the maintenance of a cluster-wide beacon. It is therefore of interest to begin to take a more serious look at these hardware platforms that form the interface to the network — the *sensor network clients*.

Existing approaches to interfacing with wireless sensor networks typically involve the use of a mobile phone, personal digital assistant (PDA) or laptop computer to interface to the network, through a gateway such as a wireless sensor node with a universal serial bus (USB) interface [25]. Laptops, PDAs, and mobile phones have lifetimes limited to between a few hours (for laptops) to a few days (for PDAs and mobile phones) during active usage, as illustrated in Table 1. There are however many applications envisaged for wireless sensor networks where it would be desirable to have a network access device with lifetimes of multiple days or even weeks. For example, an often proposed application of sensor networks is forest fire monitoring. Real-life deployments of fire fighters typically require them to setup a base camp at a safe vantage point from which they operate, and where they may not have access to electricity for days or weeks. Other scenarios for wireless sensor network deployments, such as in disaster recovery, also typically have rescue personnel working for several hours at a time, with the duration of their shifts sometimes exceeding the lifetimes for laptop, PDA and mobile phones listed in Table 1.

Another motivation for a dedicated sensor network client platform, is the need, in some applications, for a variety of sensors on the interface platform. For example, in applications where spatial information is being extracted from a sensor network, it is often useful to be able to determine the bearing / direction of the interface device relative to the deployed network — e.g., to know in which direction to head in a disaster recovery deployment. Examples of some of the assumptions appearing in the research literature about the capabilities of sensor network interface devices are presented in Table 2. While some of these capabilities may be retrofitted to existing platforms through dedicated expansion connectors or through

Table 1. Battery capacities and lifetimes for several contemporary mobile computing platforms that may be adapted for use as interfaces to sensor networks.

Device	Battery Capacity (mAH)	Batt. Lifetime (manufacturer reported)
Handheld Devices		
HP iPAQ hx2795	1440	4-5 h active
HP iPAQ hw6945	1200	4 h active, 7 d idle
Sony Ericsson P1i	1120	10 h active, 18 d idle
Nokia N810	1500	4 h active, 14 d idle
Apple iPhone	1400	8 h active, 10 d idle
Laptops		
OLPC XO	3150	~5–12 h
Asus Eee PC	4400	2.8 h
Dell XPS M1330	4774	2.5 h
Apple MacBook	5200	6 h

Table 2. Assumptions in the research literature about sensor network "sink" devices.

(Implicit) Assumption	Examples
Compute Resources	
Sufficient compute power for computation of network set-point	[16]
Communication Interface	
Compatible PHY layer	[28, 29, 32]
Sensors and Peripherals	
Absolute or relative localization	[32]
Accurate or real-time clock	[37]
Battery Lifetime	
Lifetime of days or weeks	
Display	
Graphical display for visualizing data	

interfaces such as *secure digital I/O (SDIO)* card slots, the base systems still suffer from the aforementioned limited battery life. It is unlikely to be possible to add multiple sensor peripherals at the same time in such retrofitting, and the addition of sensors will likely exacerbate the issues of limited energy resources.

This paper introduces a mobile hardware platform, the *Sunflower Mobile Client*, designed to facilitate new research directions into the client end-points of wireless sensor networks. The platform, shown in Figure 1, incorporates a radio interface compatible with most contemporary sensor network physical layers, a low-power graphical display, and several sensors that have direct relevance to mobile applications of sensor networks. The integrated sensors include multiple temperature sensors, a humidity sensor, a digital compass, and a pressure sensor for altimeter and barometer applications. The entire system, which includes a rechargeable 2000 mAh lithium-polymer battery capable of powering the system for weeks when idle, fits within a form-factor of 2.1"×4"×0.5" — barely larger than a credit card in area, and thinner than a deck of playing cards. The platform also includes hardware facilities specifically architected for research use, such as a custom receive signal strength indication sub-circuit with a miniature integrated antenna, separate from the system's radio, that enables channel monitoring at a fraction of the energy cost of the system's main communication radio. The platform incorporates facilities for dynamic introspective monitoring of the entire system's power consumption (a facility which has seen increased interest in recent sensor platforms [18, 31]), dynamic operating voltage setting under software control, and software-

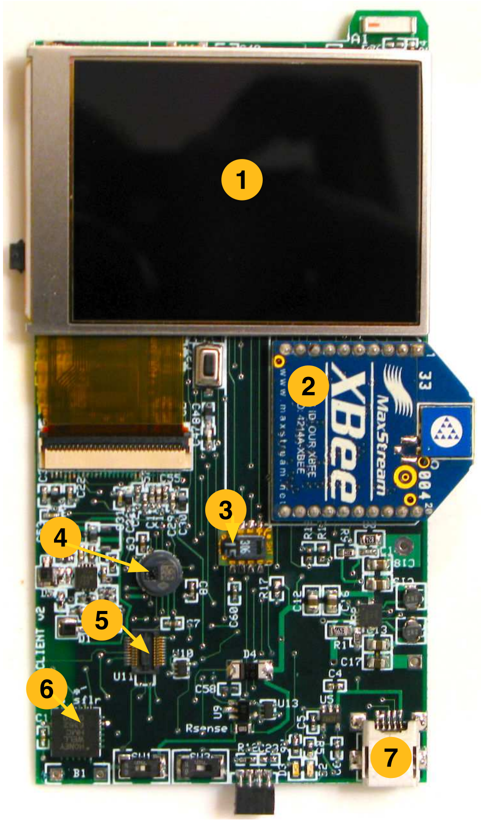


Figure 1. The Sunflower Mobile Client platform has a 320×240 pixel color display (1), humidity/temperature (3), and pressure sensors (4), and a digital compass (6). It includes a dedicated expansion connector (5), USB (7), and an 802.15.4 radio interface (2), as well as a microSD slot for flash memory or peripheral cards. The primary source of computing power is a 32-bit ARM7 implementation (AT91SAM7S256) with 64 KB of on-chip RAM and 256 KB of on-chip flash memory (on the rear side of the device), and the system is powered by a thin 2000 mAh rechargeable lithium polymer battery.

controlled gating of the power supplies of select sensors and peripherals.

Following a survey of relevant related research in Section 2, Section 3 overviews the system's architecture. The system implementation is detailed in Section 4, alongside a preliminary evaluation of the performance of several of its subsystems. Section 5 discusses insights gained in designing the first two generations of the hardware platform, and Section 6 concludes the paper with a discussion of possible avenues of research using the presented platform.

2. Related Research

Three classes of platforms are relevant to the system described in this paper. The idea of wireless sensor network *gateways* and their implementations, have been explored for many years. These gateways typically act as a bridge between the physical and network layers employed in wireless sensor networks, and those employed in traditional data networks. While *enabling* the processing and visualization of data from a network, they do not address the issue of a mobile interface that can be used to interact with the network, for example, in a disaster recovery scenario. Such gateways may however be used in conjunction with a mobile terminal such as a laptop, PDA or mobile phone.

Although off-the-shelf solutions may have the limited energy and sensing resources outlined in Section 1, there have been various research platforms for investigating energy-efficient mobile terminals. Most existing research on mobile terminal platforms has however not considered the specific needs of wireless sensor networks, such as the incorporation of the typical radio technologies used in existing deployments, the need for low-power radio channel monitoring facilities, or the need for relevant sensors and peripherals.

There have been very few attempts at creating a comprehensive mobile interface platform for sensor networks, and an overwhelming majority of existing interfaces to deployed networks occur via a personal computer connected to the network through a gateway device. In what follows, representative examples from each of the above three relevant domains are surveyed.

2.1 Wireless sensor network gateways

Wireless sensor network gateways such as the Stargate and the Stargate NetBridge platforms [5] do not directly provide hardware support for interfacing to wireless sensor network physical layers. Instead, they require a sensor node platform with such a radio interface to be connected to them, e.g., via an RS-232 connection or over USB. As gateways, their primary function is to enable the shuffling of bytes between their sensor node interface connections, and their legacy network connections.

Wireless sensor nodes with large computing resources, such as the Intel/Crossbow Imote [22] and Imote-II [1] platforms, are also often considered for use as gateways. Like the Stargate platforms, they provide significantly greater computing resources than typical sensor platforms, but unlike the Stargate platforms, the Imotes do not incorporate legacy network interfaces. While the Imote includes interfaces for connecting camera modules, and could also in principle be retrofitted with a display, it is not on its own a full-fledged platform for interfacing to a wireless sensor network.

2.2 Research mobile computing platforms

In the last decade, there have been several research platforms aimed at investigating issues relating to mobile computing. These platforms include the Berkeley Infopad [33, 2], the Active Badge system [36], the Xerox PARC smart badge systems [30], the DEC/Compaq Itsy [13] and the Delft LART [26] platforms. As research platforms, these systems enabled the investigation of new directions in hardware, systems software and applications for mobile computing. Of particular note, through the hardware facilities they provided, e.g., probe points for lab-based measurement of current drawn by system components, they enabled research directions that were otherwise cumbersome (if not impossible) with commercially available platforms. The work presented in this paper shares many of these motivations.

There have previously been many research efforts targeted at reducing the power consumption of display devices, which often form a large fraction of the total system power consumption. Early work in this area includes the observations by Flinn and Satyanarayanan, that hardware support for dimming the backlight of selective portions of a display would be useful in significantly reducing power consumption of mobile platforms [7]. Other studies of techniques for reducing display power dissipation range from adaptive backlight scaling [4] and the use of adaptive display color depths to trade off fidelity for power consumption, to techniques that take advantage of specific hardware characteristics [17, 15, 38], such as those of organic light emitting diode (OLED) displays. The display technology employed in the platform described in this paper — an OLED display — was motivated by the observations of Flinn and Satyanarayanan, as well as by the aforementioned research efforts to take advantage of novel display technologies. While these prior efforts have focused on novel *techniques* that can be implemented over a hardware platform, our contribution in this regard is to the design and implementation of a *hardware platform* that, among other things, enables the deployment of these prior research ideas.

2.3 Wireless sensor network interface platforms

Most of the existing interfaces to in-situ monitoring of wireless sensor networks (i.e., on a mobile terminal, without the use of a gateway), involve fairly simple interfaces such as a collection of light-emitting diodes (LEDs) [25].

Systems with with more sophisticated interfaces, such as the CMU eWatch [3] incorporate small liquid crystal displays (LCDs); unfortunately, the eWatch only provides a Bluetooth wireless interface, and would thus only be able to connect to typical sensor network deployments via a Bluetooth gateway. The most advanced interface for wireless sensor networks to date, that enables direct connection to deployed networks, is the SeeMote platform [31]. The SeeMote was motivated by many of the same concerns presented in Section 1. Unlike the Sunflower Mobile Client platform presented in this paper however, which was designed from the ground up to be a self-contained sensor network interface device, the SeeMote is only an add-on display board for the Micaz and Mica2 motes. While it also provides facilities for current monitoring such as those described in Section 3.1, such power monitoring facilities are only for the display, and not for the entire system. Furthermore, the SeeMote does not enable system-wide power adaptation or dynamic voltage scaling (described in Section 3.1), does not provide facilities for low-power channel monitoring (which we present in Section 3.4), does not provide the low-power system sleep facilities enabled by the use of a real-time clock (presented in Section 3.1), and does not incorporate sensors such as those we present in Section 3.5.

Examples of the use of PDAs and mobile phones as interfaces to sensor networks include the "Tricorder" platform [19]. As elaborated in Section 1, such systems are limited in lifetime, must operate using a separate gateway to the sensor network, and do not provide access to a variety of sensors such as those presented in Section 3.5.

3. System Architecture

To address the challenges noted in the foregoing sections, we designed and implemented two generations of a hardware platform, the Sunflower Mobile Client, intended for use as a dedicated client for wireless sensor networks. The platform integrates computation and sensing facilities, a radio interface, an input device and display, along with a rechargeable battery, in an ultra-portable form-factor, shown previously in Figure 1. The system architecture is illustrated in Figure 2. A summary of the hardware properties and the motivation for their incorporation in the design is presented in Table 3. These facilities directly address the common assumptions about the hardware properties of sensor network "sink" nodes or clients, previously listed in Table 2. The appropriate groupings of these facilities are discussed with respect to their research contributions and potential for enabling new research directions, in the following sections.

Table 3. Summary of properties of the Sunflower Mobile Client platform, in the context of the common assumptions of sensor network sink devices (Table 2).

	Motivation and Application
Compute Resources	
32-bit ARM7 processor at up to 60 MHz, with 256 KB of on-chip flash.	Sufficient resources for tasks such as computation of network set-points or signal processing.
Communication Interface	
IEEE 802.15.4 radio interface and dedicated MAC processor	Compatible with physical (PHY) layers of common sensor nodes, supports implementation of research MAC layers.
Dedicated receive signal strength indication circuit	Permits low-power channel monitoring.
Sensors and Peripherals	
Real-time clock	Permits the maintenance of an accurate time base, despite processor sleep and wakeup, and effects of temperature / time on oscillator crystal drift.
Digital compass peripheral	Enables implementation of relative positioning systems.
Pressure sensor	Provide sufficient resolution for altimeter applications.
Humidity sensor	Useful in monitoring applications executing on the handheld platform, and for calibration or debugging of deployed networks of nodes.
Temperature sensors	Applications similar to the humidity sensor, as well as in computing temperature-compensated sensor readings.
MicroSD card slot	Permits the use of removable memory media of up to 8 GB.
Power delivery, monitoring and management	
2000 mAh capacity ultra-thin lithium-polymer battery	Small size but large capacity (compare to entries in Table 1).
Built-in power monitoring	Enables algorithms that rely on on-line power estimation.
Software-controlled voltage regulator and voltage gating	Enables further reduction of system idle power dissipation.
Display	
320×240 18-bit color OLED graphic display	Permits the visualization of data such as spatial maps of monitored phenomena, and debugging (e.g., packet traces).
Expansion and debugging	
Expansion connector	Permits the development of expansion boards for new features.
JTAG interface	Hardware debug support.
USB 2.0 high speed	Interfacing to PCs.

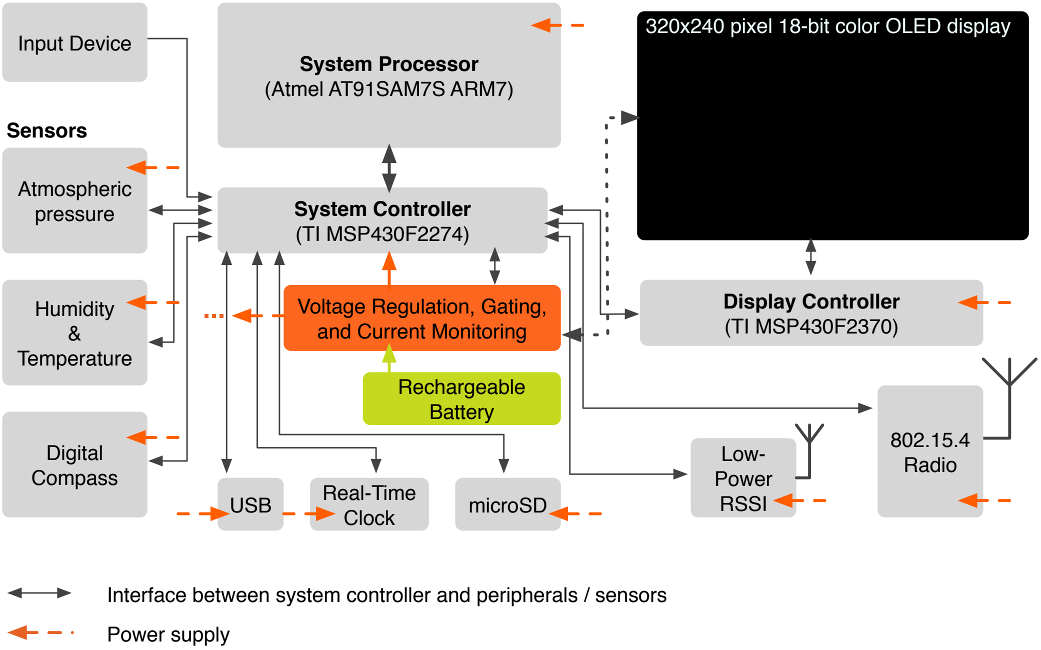


Figure 2. System architecture of the Sunflower Mobile Client platform. The *system controller* implements the low-level software interfaces to peripherals and sensors, and applications run over the *system processor*, an ARM processor running FreeRTOS.

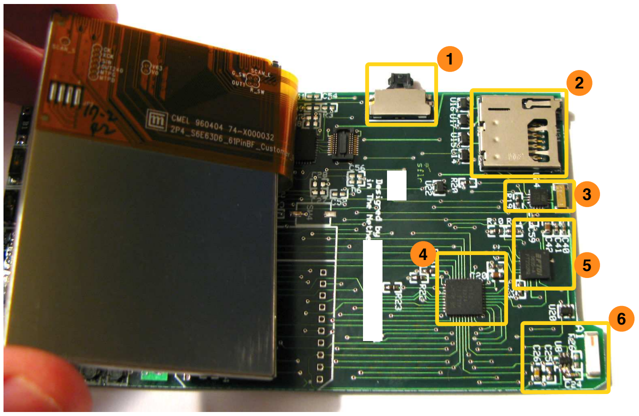


Figure 3. Underside of Sunflower Mobile Client display, showing: (1) the thumb input interface, (2) the microSD card slot, (3) the real-time clock IC and crystal, (4) the system controller, (5) the USB interface IC, and (6) the low-power receive signal strength indication (RSSI) circuit.

3.1 Low-power operation

The Sunflower Mobile Client platform employs an integrated hierarchy of techniques to achieve *both* a reduction of average power consumption, as well as a range of power-performance trade-off operating points. The hardware portion corresponding to the system power management is shown in Figure 3.

At the bottom of the hierarchy of techniques employed, is an ultra-low power real-time clock (RTC) integrated circuit. The RTC is a dedicated integrated circuit for performing accurate time-keeping over long time periods, and incorporates facilities for compensating for drifts in oscillator frequency that occur with time and temperature. The RTC consumes only 0.35 μA while maintaining time-keeping, and is the only system component that always remains active. In addition to maintaining an accurate time reference, the RTC can be configured to generate interrupts to wake up a device from a sleep mode.

Next in the hierarchy is the *system controller*, a Texas Instruments (TI) MSP430F2274 microcontroller, with 1 kB of on-chip RAM and 32 kB of on-chip flash memory. The system controller manages the power states of all devices in the system, implements the low-level details of interfacing with all peripherals in the platform.

The system is powered by a 2000 mAh rechargeable lithium-polymer battery, which is charged over USB. A programmable voltage regulator, the TI EasyScale TP62420, which supports dynamic setting of its output voltage via a control interface, is used to provide a stable supply voltage to the system. The default operating voltage of the system is set at 2.8 V, and the system controller may change this operating voltage dynamically. The use of a voltage regulator is necessary as the battery terminal voltage of the system's battery varies from 4.2 V (too high for some system components) when fully charged, to 2.5 V when depleted (too low for many system components). While most of the components of the Sunflower Client system must operate at a fixed voltage of 2.8 V, the voltage scaling facilities of the regulator can be used to further reduce system power consumption when only the system controller is active. This is made possible by the broad range of operating voltages and frequencies supported by the MSP430 family of processors, which can operate in a voltage range of 1.8 V–3.6 V.

The current leaving the battery is monitored using a combination of a high-accuracy, low-ohmic-value current sensing resistor and a current monitoring TI INA195 amplifier. The voltage output of the amplifier, which is proportional to the current being drawn from the battery, is read by the system controller's analog to digital converter (ADC) interface. In addition to enabling dynamic in-system measurement of the system's power dissipation, this facility also enables more accurate estimation of the system's remaining battery charge, compared to the often employed technique of estimating battery capacity by monitoring battery terminal voltage. The increased battery life estimation accuracy is because lithium-polymer and lithium-ion batteries have relatively flat battery discharge profiles, and when the battery voltage begins to drop appreciably, they are very near their depletion points. By continuously monitoring the current being drawn from the battery (a technique sometimes referred to as *Coulomb counting*), it is possible to more accurately predict the system's remaining battery charge.

All components of the system were chosen for their low power dissipation and provision of low-power idle or sleep modes. On some devices however, the power dissipation even in idle mode is not trivial, while other devices are so simple they do not have an interface for putting the device to sleep. For example, the current-monitoring amplifier, which typically draws a constant current of 700 μA, has no interface to enable shut-down. A potential technique that can be used to further reduce power consumption in these cases is to employ *power supply gating*, in which a transistor is placed in series with the device's power supply, disconnecting it completely under the control of a logic signal.

Employing gated supplies however involves a trade-off, as the gate transistor dissipates quiescent power (albeit small, of the order of tens of nano-Amperes per gate), and also introduces an additional resistance in the power supply path. The power dissipated by such a supply gating scheme, P_{gating}, can be expressed as a function of the supply voltage (V), the current drawn by the device whose supply is being gated (I_{load}), the on-resistance of the gating transistor (r_{on}) and the quiescent gate leakage current dissipated by the gate transistor regardless of its switch state (I_{gateleak}):

$$P_{\text{gating}} = I_{\text{load}}^2 \cdot r_{\text{on}}, \tag{1}$$

when the gate switch is "on" (device receiving power), and

$$P_{\text{gating}} = I_{\text{gate leak}} \cdot V, \tag{2}$$

when the device's power is disconnected (gate switch "off"). When the device whose supply is being gated draws large currents in active mode, the $I_{\text{load}}^2 \cdot r_{\text{on}}$ losses and quiescent power losses may overtake the gains obtained from reduced power in idle mode. It is therefore important to carefully consider the properties of both the gate switch used (in particular, its r_{on}), the typical load current, and the fraction of time a device is expected to spend active versus idle.

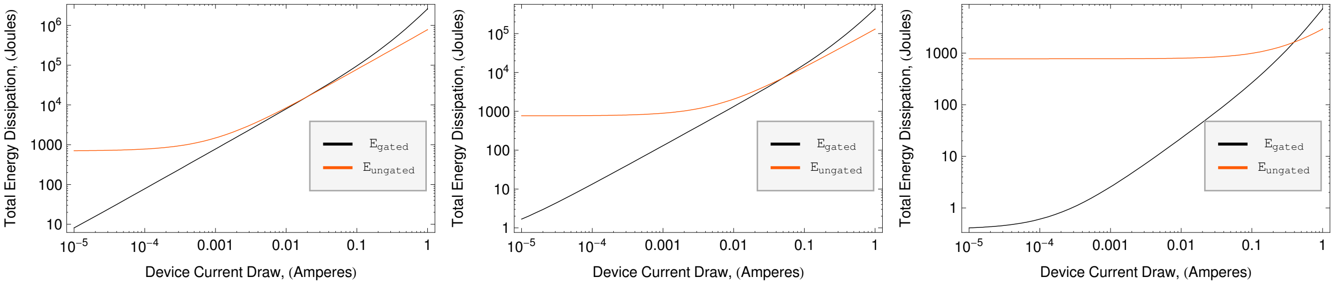
Over a duration of T seconds, for a device that spends fraction k of this time in an active state (drawing a current of I_{load}), the energy dissipation with and without power supply gating are given by

$$\begin{aligned} E_{\text{gated}} &= \left(I_{\text{load}} \cdot V + I_{\text{load}}^2 \cdot r_{\text{on}}\right) \cdot k \cdot t \\ &\quad + I_{\text{gate leak}} \cdot V \cdot (1-k) \cdot t, \\ E_{\text{ungated}} &= I_{\text{load}} \cdot V \cdot k \cdot t \\ &\quad + I_{\text{gate leak}} \cdot V \cdot (1-k) \cdot t. \end{aligned} \tag{3}$$

Figure 4 shows the gated and ungated device energy usage, for devices spending differing fractions of their time active versus idle. In the Sunflower Client platform, CMOS power gate switches with an on-resistance (r_{on}) of 10 Ω, and quiescent current of 50 nA are employed in gating the 2.8 V power supply. It can be seen from the figure, that, for example, over a one month period of operation, such power supply gating is barely beneficial for a device that draws 100 μA of current when idle and 10 mA when active, if it needs to be active for one second out of every ten seconds (Figure 4(a)). On the other hand, power supply gating is clearly useful for the same device when it spends less than one second a minute active. Based on these analysis, power supply gating is provided in the Sunflower Client platform for the system's USB controller, power monitoring amplifier, system ARM processor, microSD socket and secondary expansion header, but not for the other peripherals in the system.

3.2 System computation processor

The main computational resource of the system is a 32-bit ARM7 processor, and ATMEL AT91SAM7S256,



(a) 1/10 active/idle time ratio. (b) 1/60 active/idle time ratio. (c) 1/3600 active/idle time ratio.

Figure 4. Total energy usage over a 30 day period, with and without the overheads and benefits of power supply gating, for a device with 100 μA idle power dissipation. The energy usage is plotted for a range of active-mode currents drawn by the device being gated, at various proportions of active versus idle times.

referred to henceforth as the *system processor*, with 256 kB of flash memory and 64 kB of RAM. The system processor is mounted on a removable module on the rear of the Sunflower Client platform, and relies on the *system controller* described previously for access to all peripherals of the system. This design decision will enable the easy upgrading of the compute resources of the system, e.g., to employ a faster processor or one with more code storage flash memory, or more RAM.

In addition to the facilities within the ARM system processor for shutting down its internal core and peripherals (such as its internal voltage regulator, brown-out detector, flash memory, ADC and USB peripherals), the system controller may also gate the supply voltage of the system ARM processor, using the facilities described in the foregoing section, further reducing its idle power consumption.

3.3 Display and display power

The Sunflower Client platform employs an organic light emitting diode (OLED) display. Unlike liquid crystal displays (LCDs), which require a backlight for illumination of the display image, OLEDs employ *self-emissive* pixels — pixels emit light rather than filtering a backlight as in the case of LCDs. This construction enables high contrast ratios, wide viewing angles, and selective illumination of only portions of the display, and results in low power consumption. The low power consumption is due partly to the fact that only pixels which are lit draw a larger current, and the power consumption of the display can be dramatically affected by adapting the displayed image. For example, displaying monochrome images on a black background can be used to reduce power consumption compared to displaying color images. The ability to selectively illuminate only portions of the display make it a perfect fit for the implementation of techniques described in the research literature [7, 17, 15, 38].

The display employed in the Sunflower Client platform has a resolution of 320×240 pixels, and measures 2.2" diagonally. It is capable of displaying 18-bit color images (262,144 colors), and supports facilities such as selective update of sub-portions of the display. The display is controlled over a serial peripheral interface (SPI), by a dedicated microcontroller, a TI MSP430F2370, referred to henceforth as the *display controller*.

Employing a separate microcontroller for the display enables reduction in system power dissipation by only having to power up the minimum hardware needed for any situation. For example, at system initialization time, the system's ARM processor might remain powered down, while the system controller initializes peripherals and initiates power monitoring and control. In the absence of user input, the system processor and display microcontroller can remain powered down in their lowest power modes, awaiting external interrupts from the system controller to awaken them. For displaying simple system status information, the system controller powers-up the display controller and issues drawing commands to it. The system controller and display controller achieve pipeline parallelism since display commands issued to the display controller will be processed while the system controller is performing other control functions or preparing the next command. By splitting up the display task in this manner, the system can achieve twice the throughput (since the task is split across two stages), or conversely, can achieve the same throughput at half the operating frequency. This means the system can be operated at a lower dynamic power consumption, and is an application of a the well known technique of parallel composition for throughput-conserving power consumption reduction [23].

3.4 Low-power channel monitoring

In many wireless communication systems, and in sensor networks in particular, the radio communication interface accounts for a large fraction of the system's power consumption. The importance of the radio power dissipation is heightened by the use of carrier-sense multiple access (CSMA) medium access control protocols, which often require some amount of channel monitoring, even for nodes in the network that are not in the process of active communication. As a result, there have been many attempts to reduce this *idle listening*, with techniques ranging from the use of dedicated wakeup radio hardware or decoupled radio versus system processing [35, 27], to the use of software-driven low-power listening [24].

In some IEEE 802.15.4 radios, this problem is made seemingly worse at low transmit power settings, where the receive/listen power consumption might even *exceed* that of transmit power consumption. This is because, in order to enable extremely low power operation, the IEEE 802.15.4 physical layer (PHY) employs 16-ary orthogonal multi-level signaling [12]; due to the signal processing circuits that must thus be active in the receive versus transmit signal path [6], along with the power costs of the low noise amplifier (LNA) in the receive signal path, receive/listen power dissipation may end up being *larger* than transmit power dissipation.

When performing *channel monitoring* however, it is often sufficient to just gauge the amount of modulated signal energy in the communication channel, without the additional overhead of decoding the modulated signal. In order to simply ascertain a measure of channel occupancy, it is possible to employ very simple circuitry that functions at a fraction of the receive power consumption of a full-fledged radio, detecting channel occupancy, but not decoding the modulated signal. Such circuits can be easily built out of a radio frequency (RF) amplifier, antenna matching circuits, and an antenna. In the Sunflower Client, we employ a Linear Technologies LT5534 RF amplifier [20], in conjunction with a miniature chip antenna, in implementing low-power channel monitoring.

While active, the low-power channel monitor consumes approximately 7 mA (compared to 50 mA of the system's 802.15.4 radio in idle/listen mode). The output of the channel monitor is a voltage that is linearly related to the *logarithm* of the receive signal strength [20], thus enabling easy calculation of the receive signal strength in dBm (transmit power normalized to 1 mW, in decibels). This signal is connected to an analog to digital converter (ADC) input of the system controller. This input pin on the system controller can also be configured to generate interrupts on a voltage level trigger, and thus the receive signal strength indication subsystem can also in principle be used to wake the system from sleep. The entire system can thus be placed in a power-down state while the low-power channel monitoring is active. When not needed, the channel monitoring can be deactivated, with a power-down current drain of only 0.1μA.

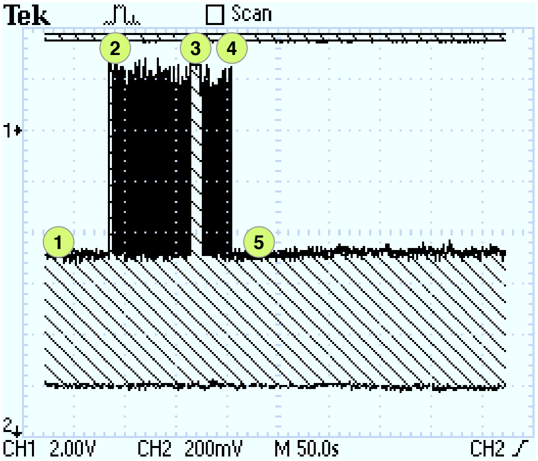


Figure 5. Output of RSSI circuit, measured on an oscilloscope, showing detected energy level during activity of a pair of nearby Bluetooth devices: (1) devices off, (2) Bluetooth device browse, (3) Bluetooth file transfer, (4) device browse, (5) devices off.

Figure 5 illustrates the output signal of the low power channel monitoring / receive signal strength indication (RSSI) circuit on the Sunflower Client platform, in the presence of activity between a Bluetooth handset and a mobile terminal. As can be seen in the Figure, the system can clearly register the presence of ongoing communications in the 2.4 GHz band.

This scheme for low-power channel monitoring is however not perfect. In practice, the 2.4 GHz band realization of the IEEE 802.15.4 physical layer splits the band into 16 channels, while the foregoing technique detects energy in the entire band. Our implementation may thus be seen as a low-power dedicated equivalent of the 802.15.4 receiver energy detection (ED) performed over the entire frequency band. Using this scheme may therefore provide false positives when the platform is operating on a specific channel. On the other hand, detection of energy across the entire frequency band may be used to detect situations where cross-modulation effects across channels may arise, or where wide-band interferers such as microwave ovens are being operated.

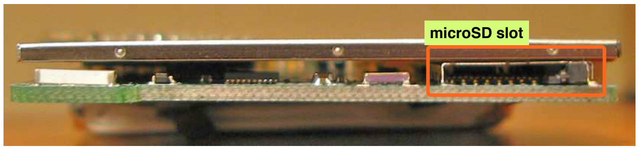


Figure 6. The Sunflower Client platform is equipped with a microSD flash card slot for memory expansion.

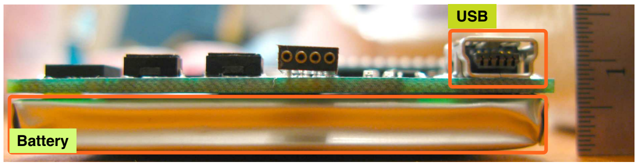


Figure 7. A USB 2.0 high speed interface is provided via a mini-USB connector.

3.5 Sensors, radio, USB, and expansion interfaces

The Sunflower Client platform includes a radio communication interface based on the IEEE 802.15.4 physical layer, and sensors for four different phenomena — temperature, humidity, atmospheric pressure and compass bearing. In addition to these built-in sensors, it includes a dedicated expansion interface for connecting other peripherals in the future, such as, e.g., a GPS module.

The radio interface is implemented with a commercially available radio module, the XBee module from Maxstream [21]. The radio module contains a Freescale MC13193 RF transceiver [10], and a Freescale MC9S08GT60 microcontroller [9] (referred to henceforth as the communication processor). The communication processor implements the IEEE 802.15.4 MAC layer, and can be configured as an end device, router, or coordinator, with a subset of the Zigbee protocol stack above the 802.15.4 MAC. It may also be reprogrammed with any alternative MAC layer, such as any of the MAC protocols reported in the research literature. One advantage of using such a module, as opposed to integrating radio transceiver circuitry on board, is that such modules can be obtained precertified to government regulatory standards, such as FCC certification in the United States, and ETSI certification in Europe. Independently obtaining such certification is costly, especially from the perspective of University-based platforms which do not yield the economies of scale of commercial platforms. Other advantages of self-contained radio hardware include upgradability of the radio subsystem.

The sensors in the system are connected to the system controller over a serial peripheral interface (SPI) bus for the dual temperature-humidity and pressure sensors, and over an inter integrated circuit communication (I2C) bus for the digital compass. All three sensors can be placed in a sleep mode when not active, and remain in this state by default. When a request for a sensor reading is provided to the system controller, it wakes up the appropriate sensor, takes the reading, and puts it back to sleep.

Code and data expansion storage is provided through a *microSD* flash card slot (Figure 6), enabling the use of memory expansion cards, which are available on the market in sizes of up to 8 GB. A USB interface is provided by means of a dedicated USB integrated circuit [11], and a mini-USB connector (Figure 7). Drivers are available for the USB controller employed, for the Linux, Windows and MacOS operating systems. We have developed utilities to enable interaction with the system controller and writing to the microSD slot, through this USB interface.

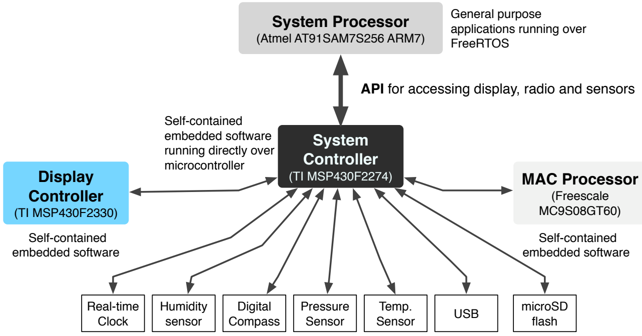


Figure 8. Implementation of low-level system software: interfacing with all hardware peripherals is implemented on the system controller, and the system ARM processor accesses these facilities through a simple API.

3.6 System software

The Sunflower Client hardware platform contains a default of 4 processors — the system controller, the display controller, the communication processor (part of the radio module), and the system ARM processor. The system and display controllers are currently programmed independently to implement their respective functionality. Likewise, the communication processor in the radio module is pre-programmed with an 802.15.4-compliant MAC layer implementation. The system's main computation processor, the ARM, currently runs a port of the FreeRTOS [8] operating system.

The system software architecture is illustrated in Figure 8. The interfaces to all the sensors in the system are implemented on the system controller, as self-contained embedded software running directly over the hardware (i.e., without an operating system). The system controller only handles the details of hardware interfacing, and is not used for the execution of applications. General purpose applications execute on the system ARM processor, and access peripherals via a simple API to the system controller. We are currently investigating the possibility of the deployment of operating systems, such as CoMOS [14], and programming languages and their runtimes, such as the *error-tolerant name generator model* [34], designed specifically for such heterogeneous multiprocessor systems.

3.7 Debugging support

The Sunflower Client hardware platform provides multiple facilities for software update and debug. Software updates for the system ARM processor, system controller, display controller and communication processor may be performed via the microSD card, either by loading the updates in a separate device, or through the system's USB interface, which permits direct writes to the microSD slot. Updates may also occur directly over USB without storing the code in the microSD flash, or over the radio interface.

To enable more flexible debugging, such as single-stepping the code running on the different processors, IEEE 1149.1 joint test action group (JTAG) interfaces are provided for the system controller, display controller, and system ARM processor.

4. System Implementation and Evaluation

The hardware implementation of the Sunflower Client platform contains over 130 components. As described in the foregoing sections, careful attention was given to ensuring that the hardware implementation can enable operation at the lowest possible power dissipation.

4.1 Implementation cost

To illustrate the viability of the Sunflower Client platform as an affordable research tool, a breakdown of the fabrication costs is presented in Table 4. The parts cost is approximately 200 USD, based on bulk purchases of the system components listed. For smaller quantities, and when including the costs of assembly and testing, the costs are likely to be slightly higher.

4.2 Per-component power breakdown

Table 5 shows the active and idle power breakdowns for all components of the system. The power consumption for the microSD interface and expansion devices are not included as they are zero in the absence of inserted microSD cards or expansion modules (there is no logic associated with them other than the voltage gates which are listed together), and in their presence, depend on the inserted cards / attached modules.

Table 4. Fabrication costs for the Sunflower Client platform, based on bulk purchases of system components. A subset of the sensors may be omitted to obtain an even less expensive implementation.

Component	Description	Quantity	Cost (USD)
MSP430F2274	System controller	1	4.30
MSP430F2370	Display controller	1	3.10
AT91SAM7S256	System ARM processor	1	12.60
XB24	802.15.4 radio module	1	19.00
PPT9999	OLED display	1	30.00
HMC3652	Digital compass	1	32.00
SCP1000D01	Pressure sensor	1	12.18
SHT1x	Humidity & temperature	1	28.00
LT3471	OLED Voltage regulator	1	2.98
TPS62420	System voltage regulator	1	3.65
TS5A1066	CMOS SPST switch	9	1.00
INA195	Current sense amplifier	1	1.03
LT5534	RF amplifier	1	5.35
FT232R	USB interface IC	1	6.70
M41T65	Real-time clock	1	0.89
MAX1555	USB charger IC	1	1.10
Diodes	Schottky diodes, LEDs	6	2
Passives	Resistors, Crystals, etc.	92	20
Miscellaneous	Connectors, switches	12	15
PCB	4-layer circuit board	1	5
Total			**205.88**

Table 5. Per-component power consumption breakdowns (at 3 V operating voltage), for the Sunflower Client platform. Devices whose power supplies are gated in a given mode are noted. The effective active power dissipation of the voltage regulators (the TPS62420 and LT3471) are dependent on their load currents.

Component	Active Power	Idle/sleep Power	Supply Gated ?
MSP430F2274	12 mW @12 MHz	0.3 μW	no
MSP430F2370	12 mW @12 MHz	0.3 μW	no
AT91SAM7S256	567 mW @60 MHz	114 μW	yes
XB24	150 mW	30 μW	no
OLED display	240 mW	3 mW	(shut off)
HMC3652	3 mW	3 μW	no
SCP1000D01	75 μW	0.6 μW	no
SHT1x	1.65 mW	1 μW	no
LT3471	n/a	3 μW	no
TPS62420	n/a	3.6 μW	no
TS5A1066	0.15 μW	n/a	no
INA195	2.1 mW	n/a	yes
LT5534	21 mW	0.3 uW	no
FT232R	45 mW	210 uW	yes
M41T65	105 μW	1 uW	no
MAX1555	8.75 mW	n/a	no

5. Design Insights and Discussion

The Sunflower Client platform was developed to enable the pursuit of research directions pertaining to interfacing to and tasking wireless sensor networks. The goals of such a platform include its use as a *sink* for traffic from the network, or a location for the computation of network configuration points. It was motivated in part by our involvement in a large multi-institution research project, funded by the European Union, on wireless sensor network hardware, software and applications. As part of this project, we realized the need for a platform for interfacing to the network, to perform tasks such as injecting queries into the network, displaying the results obtained from such queries, as well as more compute-intensive tasks such as the calculation of network set-points based on usage requirements or user constraints. Experience in using platforms such as PDAs and laptops, as well as the feedback obtained from potential users, led us to the idea of developing a custom research platform such as the Sunflower Client.

The platform presented in this paper is the second generation of the Sunflower Client platform. The largest change between the two generations was a shift from a 4-bit greyscale OLED display to the 18-bit color display, and the addition of the display microcontroller. From our experience using the previous generation, several implementation changes were also made, including several layout changes such as the positioning of the chip-scale antenna for the low-power RSSI circuit. The system software for access to all system components, except the display, however remained unchanged.

6. Summary and Future Directions

This paper presented the motivation for full-fledged, self-contained mobile computing platforms to serve as *clients* for wireless sensor networks, and presented the design and implementation of one such platform, the Sunflower Mobile Client. The Sunflower Client incorporates a 320×240 18-bit color OLED display, an ARM processor, rechargeable 2000 mAh lithium-polymer battery, temperature, humidity and atmospheric pressure sensors, a digital compass, an 802.15.4 radio communication interface and a separate independent low-power RSSI circuit in an small form factor. The entire system, including its rechargeable battery, is only 2.1"×4"×0.5", barely larger in area than a credit card, and thinner than a deck of playing cards. A dedicated low-power real-time clock circuit enables accurate time-keeping even in the presence of system sleep operation and the effects of temperature on oscillator drift. Through innovative system architecture design, the system provides a broad range of operating modes, at different points in the power-performance trade-off envelope. For future expansion, the system includes two expansion interfaces — a microSD card slot, and a peripheral expansion interface.

The platform implements several techniques, and incorporates several facilities, for energy-efficient and energy-adaptive operation. These facilities include the shutdown of all peripherals, the gating of the power supplies of select peripherals for which such action has a favorable trade-off, and the potential for software controlled dynamic voltage setting.

We have built two generations of the Sunflower Client platform, and are currently focusing on taking advantage of the research directions that the platform facilitates. Current applications being pursued both at our research institution and elsewhere, using the Sunflower Client platform, include the development of multi-protocol network debuggers for wireless sensor network protocol stacks, ubiquitous computing applications involving sensor networks such as location-aware games, the development of real-time schedulers for sensor network interface platforms, and the implementation of online trade-off and optimization algorithms for configuring wireless sensor networks.

References

[1] R. Adler, M. Flanigan, J. Huang, R. Kling, N. Kushalnagar, L. Nachman, C.-Y. Wan, and M. Yarvis. Intel mote 2: an advanced platform for demanding sensor network applications. In *SenSys '05: Proceedings of the 3rd international conference on Embedded networked sensor systems*, pages 298–298, New York, NY, USA, 2005. ACM Press.

[2] R. W. Brodersen. Infopad - past, present and future. *SIGMOBILE Mob. Comput. Commun. Rev.*, 3(1):1–7, 1999.

[3] Carnegie Mellon University. CMU eWatch `http://flat-earth.ece.cmu.edu/~eWatch/` (accessed November 2007).

[4] N. Chang, I. Choi, and H. Shim. Dls: dynamic backlight luminance scaling of liquid crystal display. *IEEE Trans. Very Large Scale Integr. Syst.*, 12(8):837–846, 2004.

[5] Crossbow. Stargate NetBridge `http://www.xbow.com` (accessed November 2007).

[6] J. Edgar H. Callaway. Low Power Consumption Features of the IEEE 802.15.4/ZigBee LR-WPAN Standard. In *Invited Talk, SenSys '03: Proceedings of the 3rd international conference on Embedded networked sensor systems*, 2003.

[7] J. Flinn and M. Satyanarayanan. Energy-aware adaptation for mobile applications. *SIGOPS Oper. Syst. Rev.*, 33(5):48–63, 1999.

[8] FreeRTOS. `http://www.freertos.org`, accessed November 2007.

[9] Freescale Semiconductor. MC9S08GT60 datasheet. 2004.

[10] Freescale Semiconductor. MC13193: 2.4 GHz, Low Power Transceiver for 802.15.4. 2007.

[11] Future Technology Devices International Ltd. Datasheet, FT232R — USB UART IC. 2005.

[12] J. A. Gutierrez, J. Edgar H. Callaway, and J. Raymond L. Barrett. In *Low-Rate Wireless Personal Area Networks: Enabling Wireless Sensors with IEEE 802.15.4*, chapter 4. IEEE Press, New york, 2003.

[13] W. R. Hamburgen, D. A. Wallach, M. A. Viredaz, L. S. Brakmo, C. A. Waldspurger, J. F. Bartlett, T. Mann, and K. I. Farkas. Itsy: Stretching the bounds of mobile computing. *Computer*, 34(4):28–36, 2001.

[14] C.-C. Han, M. Goraczko, J. Helander, J. Liu, B. Priyantha, and F. Zhao. Comos: An operating system for heterogeneous multi-processor sensor devices. Technical Report MSR-TR-2006-177, Microsoft Research, December 2006.

[15] T. Harter, S. Vroegindeweij, E. Geelhoed, M. Manahan, and P. Ranganathan. Energy-aware user interfaces: an evaluation of user acceptance. In *CHI '04: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 199–206, New York, NY, USA, 2004. ACM.

[16] R. Hoes, T. Basten, C.-K. Tham, M. Geilen, and H. Corporaal. Analysing qos trade-offs in wireless sensor networks. In *MSWiM '07: Proceedings of the 10th ACM Symposium on Modeling, analysis, and simulation of wireless and mobile systems*, pages 60–69, New York, NY, USA, 2007. ACM.

[17] S. Iyer, L. Luo, R. Mayo, and P. Ranganathan. Energy-adaptive display system designs for future mobile environments. In *MobiSys '03: Proceedings of the 1st international conference on Mobile systems, applications and services*, pages 245–258, New York, NY, USA, 2003. ACM.

[18] 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*, pages 186–195, New York, NY, USA, 2007. ACM.

[19] J. Lifton, M. Mittal, M. Lapinski, and J. A. Paradiso. Tricorder: A mobile sensor network browser. In *Proceedings of the ACM CHI 2007 Conference - Mobile Spatial Interaction Workshop*, April 2007.

[20] Linear Technology, Inc. Datasheet, LT5534 — 50MHz to 3GHz RF Power Detector with 60dB Dynamic Range. 2004.

[21] Maxstream, Inc. Datasheet, XBee ZigBee OEM RF Module. 2006.

[22] L. Nachman, R. Kling, R. Adler, J. Huang, and V. Hummel. The intel mote platform: a bluetooth-based sensor network for industrial monitoring. In *IPSN '05: Proceedings of the 4th international symposium on Information processing in sensor networks*, page 61, Piscataway, NJ, USA, 2005. IEEE Press.

[23] P. I. Pénzes and A. J. Martin. Energy-delay efficiency of vlsi computations. In *GLSVLSI '02: Proceedings of the 12th ACM Great Lakes symposium on VLSI*, pages 104–111, New York, NY, USA, 2002. ACM.

[24] J. Polastre, J. Hill, and D. Culler. Versatile low power media access for wireless sensor networks. In *SenSys '04: Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 95–107, New York, NY, USA, 2004. ACM.

[25] J. Polastre, R. Szewczyk, and D. Culler. Telos: enabling ultra-low power wireless research. In *IPSN '05: Proceedings of the 4th international symposium on Information processing in sensor networks*, page 48, Piscataway, NJ, USA, 2005. IEEE Press.

[26] J. Pouwelse, K. Langendoen, and H. Sips. Dynamic voltage scaling on a low-power microprocessor. In *MobiCom '01: Proceedings of the 7th annual international conference on Mobile computing and networking*, pages 251–259, New York, NY, USA, 2001. ACM.

[27] J. M. Rabaey, J. M. Ammer, J. J. L. da Silva, D. Patel, and S. Roundy. Picoradio supports ad hoc ultra-low power wireless networking. *Computer*, 33(7):42–48, 2000.

[28] N. Ramanathan, K. Chang, R. Kapur, L. Girod, E. Kohler, and D. Estrin. Sympathy for the sensor network debugger. In *SenSys '05: Proceedings of the 3rd international conference on Embedded networked sensor systems*, pages 255–267, New York, NY, USA, 2005. ACM.

[29] Y. Sankarasubramaniam, Özgür B. Akan, and I. F. Akyildiz. Esrt: event-to-sink reliable transport in wireless sensor networks. In *MobiHoc '03: Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing*, pages 177–188, New York, NY, USA, 2003. ACM Press.

[30] B. N. Schilit, N. Adams, R. Gold, M. Tso, and R. Want. The PARCTAB mobile computing system. In *Proceedings Fourth Workshop on Workstation Operating Systems (WWOS-IV)*, pages 34–39. IEEE, October 1993.

[31] L. Selavo, G. Zhou, and J. A. Stankovic. Seemote: In-situ visualization and logging device for wireless sensor networks. *Broadband Communications, Networks and Systems, 2006. BROADNETS 2006. 3rd International Conference on*, pages 1–9, 2006.

[32] R. C. Shah and J. M. Rabaey. Energy aware routing for low energy ad hoc sensor networks. In *IEEE Wireless Communications and Networking Conference*, pages 350–355, March 2002.

[33] S. Sheng, A. Chandrakasan, and R. W. Brodersen. A portable multimedia terminal. *Communications Magazine, IEEE*, 30(12):64–75, 1992.

[34] P. Stanley-Marbell and D. Marculescu. A Programming Model and Language Implementation for Concurrent Failure-Prone Hardware. In *Proceedings of the 2nd Workshop on Programming Models for Ubiquitous Parallelism, PMUP '06*, September 2006.

[35] V. Tsiatsis, S. Zimbeck, and M. Srivastava. Architecture strategies for energy-efficient packet forwarding in wireless sensor networks. In *ISLPED '01: Proceedings of the 2001 international symposium on Low power electronics and design*, pages 92–95, New York, NY, USA, 2001. ACM.

[36] R. Want and A. Hopper. Active badges and personal

interactive computing objects. *Consumer Electronics, IEEE Transactions on*, 38(1):10–20, 1992.

[37] W. Ye, J. Heidemann, and D. Estrin. Medium access control with coordinated adaptive sleeping for wireless sensor networks. *IEEE/ACM Trans. Netw.*, 12(3):493–506, 2004.

[38] L. Zhong and N. K. Jha. Energy efficiency of handheld computer interfaces: limits, characterization and practice. In *MobiSys '05: Proceedings of the 3rd international conference on Mobile systems, applications, and services*, pages 247–260, New York, NY, USA, 2005. ACM.