

# A Higher Capability Sensor Node Platform Suitable for Demanding Applications

Thomas Hammel  
Fantastic Data  
207 Prospect Avenue  
San Francisco, California 94110  
1 415 643 9555  
hammel@fandat.com

Mark Rich  
Fantastic Data  
207 Prospect Avenue  
San Francisco, California 94110  
1 415 643 9555  
rich@fandat.com

## ABSTRACT

A novel, autonomous, fully distributed sensor node platform designed and built for a continuous, wide-area surveillance and security system is described. Sensor nodes cooperate to detect and track intruders in the surveilled area. Analysis and simulation of the surveillance system indicates that while considerably more capability is required in many aspects (processing power, memory, latency, communication range, and so on) than is currently available in common “mote” designs, performance, energy consumption, node lifetime, and ease of use are enhanced by this approach. Because higher capability components are used, more careful scheduling and power control software is required to mitigate the impact on energy consumption. A full software suite was developed and instrumented to record true system usage during operation of the surveillance system. Measurements of actual usage have been made on a moderately oversized prototype platform. A second generation platform has been designed based on the measured usage data. The software suite is being ported to this platform. Lifetime of the second generation platform running the demanding surveillance application is expected to be about 90 days on 2 AA batteries (3000mAh at 1.5V). Applications with less stringent requirements should enjoy much longer lifetimes.

## Categories and Subject Descriptors

C.3 Special-Purpose and Application-Based Systems—Real-time and embedded systems, C.2.1 [Computer-Communication Networks]: Network Architecture and Design—Wireless Communication, and C.4 Performance of Systems—Design Studies.

## General Terms

Algorithms, Measurement, Performance, Design, Economics

## Keywords

wireless sensor network, networking protocols, mesh networking, low power wireless systems

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

*IPSN '07*, April 25–27, 2007, Cambridge, Massachusetts, USA.  
Copyright 2007 ACM 978-1-59593-638-7/07/0004...\$5.00.

## 1. INTRODUCTION

Experimental wireless sensor networks are being developed for a variety of purposes. In the last few years, a typical approach has been to use simple, low-power devices, often called motes [3, 4, 7, 8, 9, 18], organized in a simple, hierarchical communications architecture to collect all sensor information at a common location. The sensor information is then processed to yield the desired result. In the case of a surveillance system, sensor readings (or sometime detection events) are collected from all of the sensors and target detection and tracking is performed at a central location [1, 17]. The sensor node technology required for this type of solution is simple. Unfortunately, the scalability of the central processing solution is limited, the robustness of the communication architecture is weak, and the planning needed to establish the network makes it difficult to install.

We have taken a radically different approach. We have elected to place increased processing power and memory on each sensor node as have some others [8, 16]. This increased capability appears to have a considerable cost in terms of increased energy consumption. Our thesis, however, is that increased knowledge and computation can yield better control of the remaining elements of the sensor node and yield lower, overall energy consumption. Since embedded processors continue to become more capable, smaller, and use less energy, this approach has significant benefits for the future.

The surveillance system provides security and early warning of intrusion in a large area. Hasty, unplanned deployment is normal. The system is expected to detect any intrusion, although individuals are the prime concern. Target identification is not a major concern as few intrusions are expected and these can be investigated with other means. Thus the expected application data rate is close to zero, but when targets are identified, they must be communicated to the users in a few seconds.

We have developed a software suite that employs the increased processing power and memory to retain all data and control at the local level [5]. The nodes operate autonomously—there are no external controllers. None of the control or application data traverses the network. The communication schedule is computed locally and distributed only within a small local area by the *sas* medium access protocol. Application data is managed in a local area by the *data cache* distributed database software. Detection and tracking is performed cooperatively among the nodes in a small local area. All of these processes require more processing power and access to more data than for a simple mote executing a repetitive plan developed by an external controller. Collaborative



Figure 1. The data products of the sensor network.

tracking [2, 6, 12] has been performed previously on much larger and more capable platforms [13]. The challenge is to select the right level of platform capability and software sophistication.

Figure 1 shows the application data (node locations, detections, and tracks) developed by the sensor nodes and extracted in real-time to a display unit. In normal operation, only confirmed, high-quality tracks are extracted. Application processing on the nodes decreases the amount of data that must be communicated by eliminating the need for forwarding of messages to a central site. This local processing at all levels lessens the need for reliable communication paths and decreases the response time to changing conditions which in turn allows more extreme variations in service levels.

At the beginning of this project, we did not know how far we would be able to take this concept, how successful the incorporation of power saving measures into the application programs would be, or what sensors and sensor processing would be required. The prototype provided a lot of processing power to allow exploration of these concepts—more processing power than was ultimately required. The prototype is an excellent platform for software development and performance measurement, but with a deployed lifetime of 1 day, it does not constitute a suitable platform for deployment. Working from the measured performance data, we have redesigned the platform to fit. The resulting second generation has considerably more capability than typical motes and achieves lower energy consumption.

## 2. PROTOTYPE DESIGN

A block diagram of the typical subsystems of a sensor node is shown in Figure 2. These subsystems are usually considered independently and are often implemented with separate physical components selected from the ones with the lowest possible energy consumption. In the prototype design process, we have attempted to consider the system as a whole and make tradeoffs in capability among these subsystems and to share components when possible. Indeed, our central thesis is that the provision of a larger

than expected processor can allow the implementation of more intelligent software which allows the selection of more capable components without sacrificing energy consumption.

The sensor monitors the environment. For the particular surveillance system, the sensor must be able to detect a single person. We have elected to use a passive infra-red (PIR) motion detector. Other sensor networks might use a variety of environmental sensors, such as, acoustic, seismic, image, thermal, chemical, radiological, or biological. The selection of a sensor has implications beyond its packaging and powering, as the sensor modality and the intended use influence the amount of processing required to convert raw sensor data into higher level information.

For many sensor networks, the location of individual nodes may not be very important or may be determined at the time of deployment. For a surveillance system, on the other hand, knowledge of the location of the nodes is critical. For a remotely deployed system, this knowledge must be generated within the network. Considerable research on self-localization is underway [10]. All solutions, however, require some form of infrastructure at known locations. We have elected to use the GPS satellites.

Communication between the nodes is provided by the radio. A wide variety of options are available and the selection is based upon the application and the environment. Although the commercially available medium access protocols are not well matched to sensor network needs, the physical layer modules provide very good value. Enormous effort continues to be dedicated to reducing power consumption at the physical level. The deployment environment must be considered, as there are large differences between the requirements of an indoor building monitoring system and a wide area surveillance system, especially as related to link range, bandwidth, clutter, and latency.

Timing is a critical factor often overlooked in the design of low power communication systems. As many low-power communication systems rely on duty cycle management to reduce energy consumption, the ability to synchronize clocks across multiple nodes is required so that the nodes may complete a temporal rendezvous—one node must have its receiver on when the other node transmits. Broadcast timing beacons are frequently used, but these have severe repercussions for energy consumption, on the nodes that broadcast the beacons and on the nodes that must listen for them. We calibrate the local node clock and occasionally recalibrate the clock with an independent reference source (GPS).

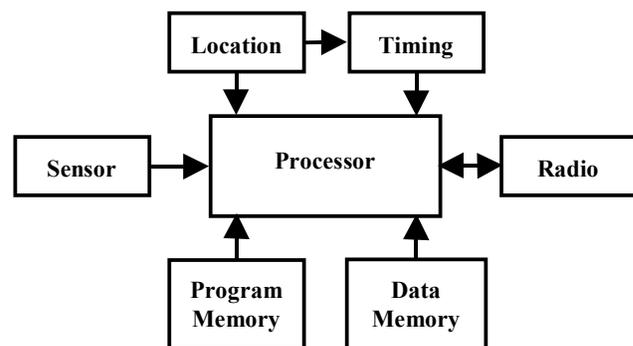


Figure 2. The major components of a sensor node.

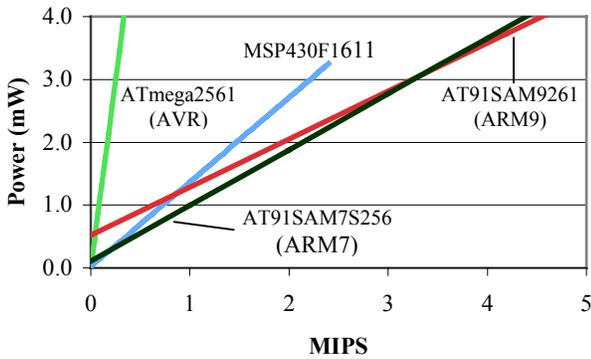


Figure 3. Power consumption vs. processing capability.

Finally, our approach relies upon the node having substantial processing power and memory, at least as compared to the traditional mote. Many mote manufacturers integrate communications with a trivial amount of processing power in order to advertise a long lifetime product. Such products are capable of executing the communication protocol, but can do little else. Indeed, some groups have added additional processors to the motes to satisfy their processing needs [11]. We prefer to equip the sensor node with one capable processor shared among the functions. As shown in Figure 3, it may be more efficient to use a high performance processor at a low duty cycle than to use a low power processor at a high percentage of its peak rate when more than a trivial amount of processing power is required. Benchmark tests performed on several processors have also indicated considerable energy savings with more capable processors [14, 15]. Since modern processor devices implement low power sleep states, the higher energy consumption of the faster devices is offset by the fact that they operate less and sleep more.

Hardware design and software development were performed in parallel. At the time of component selection for the prototype, estimates of processing requirements were fluid. Since the prototype was intended to support the exploration of the utility of the distributed approach, we chose to err on the high side when specifying hardware components. As shown in the section on usage measurements, in the selection of several components, the high side error was quite high. This either reflects poorly on our ability to estimate software requirements or well on our ability to implement efficient software. We prefer to believe the latter.

Initial exploration and simulation of the system architecture occurred during 2003. Major component selection was completed in 2004. The prototypes were designed, built, and debugged during 2005. Testing was complete by Spring 2006.

Figure 4 shows the completed prototype sensor node ready for deployment. Batteries and antennas are internal to the case. The prototype node measures 6cm x 8cm x 4cm and weighs 200g (including 60g of batteries). Figure 5 shows the 3 board electronics stack housed within the case.

The following sections describe the major components and their capabilities. These components are summarized in Table 1. The costs shown are for purchases in quantities of 100; purchases in large quantities receive substantially lower net prices as for most electronic components.



Figure 4. The finished prototype sensor node.



Figure 5. The internal elements of the prototype node.

## 2.1 Processor

A hybrid general purpose processor (GPP) and programmable logic device (PLD) was selected. The Altera Excalibur processor (EPXA1F484C3) provides a 130MHz ARM9 core and 100k APEX family logic gates operating at 20MHz.

The Excalibur processor measures 25mm x 25mm. Its energy consumption varies with the input clock rate and the settings of the internal phase lock loop (PLL). Using an input clock rate of 20MHz, the processor and memory consumes approximately 600mW when running at 130MHz (PLL configured for maximum speed) and 200mW when running at 20MHz (PLL shut off and bypassed). Unfortunately this processor chip does not implement a low power sleep state. It was selected because of its combination of programmable logic and a common computing core. It was expected to be more capable than required for the sensor node.

Real time functions are implemented in Verilog for the programmable logic. Non-real-time functions are implemented in C for the GPP. Substantial instantaneous computing power is available, but energy consumption remains relatively low due to processor duty cycle management.

**Table 1. Major components of the prototype and product nodes.**

Function	Prototype			Product		
	Part	Size (mm <sup>2</sup> )	Cost (\$)	Part	Size (mm <sup>2</sup> )	Cost (\$)
Processor	Altera EPXA1F484C3	625	60.00	Atmel AT91SAM7S256	81	8.70
Program Memory	Intel GE28F320C3BD70	49	6.08	included with processor		
Data Memory	Micron MT48V8M16LFB4-8	64	9.48	included with processor		
Timing	Epson SG-350-SCF 20.00 m-m	9	2.27	Abracon ABM8-16.000MHz-B2-T	8	1.50
Modem	Chipcon CC2400	49	3.98	Chipcon CC2500	16	2.66
T/R Switch	Hittite HMC484MS8G (2x)	18	9.96	NEC UPG2214TB (2x)	4	0.56
PA	RFMD RF5189SR	9	3.12	NEC UPG2314T5N	3	2.00
LNA	RFMD RF2373SR	12	2.00	Maxim MAX2641	9	0.80
Antenna	GigaAnt 3030A6111-01	48	3.00	GigaAnt 3030A6111-01	48	3.00
GPS	ublox TIM-LA	625	52.70	ublox NEOS-4S	200	55.00
GPS Antenna	Laipac GLP1-MMCX	625	13.00	Zhengyuan Electric DAM1575C	324	13.00
Compass	Honeywell HMC1052/HMC1051Z	38	11.06	deleted		
Sensor	GloLab PIR325/FL65	80	4.52	Panasonic AMN44122	150	26.74
ADC	Microchip MCP3208-CI/SL	90	2.63	included with processor		
Power Supply	Maxim MAX1566ETL	40	3.51	Maxim MAX8621	16	3.70
Other	340 miscellaneous parts		70.51	70 miscellaneous parts (est)		20.00
PCB Fabrication	3 boards: \$64.00+\$26.87+\$30.30		121.17	1 board (est)		30.00
PCB Assembly	3 boards: \$33.75+\$22.45+\$71.10		127.30	1 board (est)		30.00
Packaging	Case		117.00	Case (est)		60.00
TOTAL			623.29			257.66

## 2.2 Memory

Substantial external FLASH memory and dynamic random access memory (DRAM) are interfaced to the processor. The main consideration in selecting the memory components was to ensure that no constraints were placed on the experimental software.

Inexpensive, readily available memory chips were selected. The FLASH memory is an Intel GE28F320C3BD70 supplying 4MB of program and persistent data storage memory. The DRAM is a Micron MT48V8M16LFB4-8 supplying 32MB of memory for program execution. The Altera Excalibur processor loads the entire program image into the DRAM for execution.

Energy consumption of the FLASH memory during normal operation is negligible. It automatically enters a low power sleep state when not accessed. The DRAM consumes approximately 180mW when operating at full speed. A low power, self refresh state that preserves data values is theoretically possible but was not used during testing of the prototype.

## 2.3 Radio

The transceiver (Chipcon CC2400) is an integrated 1Mb/s FSK modem with a 2.4GHz radio frequency (RF) front end. The transceiver is augmented by an external low noise amplifier (LNA) (RF Micro Devices RF2373SR) and power amplifier (PA) (RF Micro Devices RF5189SR) to provide greater link margin for operation at ground level and at greater range (>100m). A compact antenna (GigaAnt 3030A6111-01) is included in the circuit for normal operation. An external antenna may be used through an MMCX connector.

Streamlined applications or those requiring a greater distance between nodes could use a lower frequency, lower bandwidth radio and achieve slightly lower power consumption. 2.4GHz was used in the prototype to provide adequate bandwidth margin for future application development. Use of a product designed for Bluetooth products ensured small size and low cost for this prototype effort. The sensor node does not, however, use the Bluetooth communication protocol.

## 2.4 Timing

Timing support is provided by an on-board  $\pm 20$ ppm, 20MHz oscillator (Epson SG-350-SCF 20.00 m-m). The node performs self calibration of the oscillator using either 1 pulse per second signals from the GPS receiver or time synchronization data embedded in the received packets from other nodes. Self calibration renders timing accuracy better than  $\pm 1$ ppm. Synchronization with an external source is still required, but at a much lower duty cycle.

The GPS receiver includes a built-in oscillator used for both received frequency tuning and for symbol recovery. The 1pps signal used for precision time synchronization is gated by the internal GPS clock. This leads to a 60ns uncertainty in the system time but this is more than an order of magnitude lower than the required timing accuracy.

## 2.5 Location

The prototype node uses a ublox TIM-LA receiver. This module measures 25mm x 25mm. It consumes 165mW in its normal

active state and 0.4mW in its sleep state. It yields a time to first fix of 4s with valid satellite ephemeris.

The GPS receiver is powered at node startup to provide the node location and to calibrate the node oscillator. It is then turned off to save energy. It is operated only occasionally as necessary to resynchronize the node clock. An absolute timing accuracy of  $\pm 500\mu\text{s}$  is required on each node to support a communication guard time of  $1000\mu\text{s}$ . Since the calibrated node clock maintains an accuracy of better than  $\pm 1\text{ppm}$ , resynchronization is required about every 500s or about once every 10 minutes.

The GPS receiver has an input sensitivity of -150dBm. An active antenna with 12dB gain is used to provide noise isolation from the processing circuitry. The prototype uses a Lairac generic ceramic patch antenna with embedded LNA.

The prototype node includes a 3-axis magnetic compass (Honeywell HMC1052 and HMC1051Z) so that pointing angles can be discerned for randomly emplaced nodes. Since the nodes are designed for static operation, the compass only needs to be used at startup. The compass requires manual calibration.

## 2.6 Sensor

Detections are provided by the passive infra-red (PIR) sensor (GloLab PIR325). The Fresnel lens (GloLab FL65) mounted in the node case forms a tightly focused detection beam with a range of approximately 20m for small, slow moving targets. Larger or faster targets, such as vehicles, can be detected at a greater range. Filtering, amplification, and acquisition of the signal are performed by custom circuitry. The 8-channel ADC (Microchip MCP3208) is also used for conversion of other internal signals, such as the battery voltage, temperature, and magnetic compass.

The detection processing on the prototype node is performed by software analysis of the signal sampled at 100Hz. Data is accumulated in a buffer and about once a second the processor wakes up and analyzes the accumulated data. The software detects the characteristic shape of a PIR signal when a moving object traverses its beam. A detection is declared which triggers many further activities, including the writing of a detection record into the data cache, the creation of a broadcast slot for communication to neighboring nodes, and the activation of the target tracker.

## 2.7 Power Supply

Commonly available nodes provide only the communication function, leaving other functions, including the power supply to the user. The sophistication of our nodes and the inclusion of the sensor warrant the inclusion of a more capable power supply. The power supply provides 1.8V, 2.5V, 3.3V, and 5V outputs.

The 3.3V supply is used for most elements. The low 1.8V supply powers the processor and radio cores to reduce their energy consumption. The 2.5V supply powers the DRAM. The 5V supply was used internally to compensate for varying battery voltage but could be used to supply external sensors as well.

Often the power supply functions are provided using simple voltage regulators but this approach may waste half the power. A DC/DC converter is typically 90% to 95% efficient at the required current levels. DC/DC converter technology allows us to provide these voltages efficiently. The prototype node uses a Maxim MAX1566ETL converter, repurposed from a digital camera.

**Table 2. Simulated radio usage summary.**

	Tx (%)	Rx good (%)	Rx collision (%)	Rx empty (%)
NetForm	0.129	0.039	0.011	0.180
Quiescent	0.007	0.000	0.000	0.247
Track (avg)	0.283	0.215	0.044	0.411
Track (max)	0.840	1.020	0.248	0.855

## 3. USAGE MEASUREMENTS

Node component usage varies with the node state—broadly defined as either “quiescent” or “tracking”. Node state is not a hard definition—there are many intermediate and mixed states—but these two named states illuminate the differences in node component usage requirements.

In the quiescent state the node is actively operating its sensor to search for intruders. The node is operating its transceiver about once per second in a scheduled fan-in receive slot in case a neighboring node may want to send it a message. There is little data traffic, most of which is overhead traffic to support network formation and maintenance.

In the tracking mode, the node or one of its neighbors has detected an intruder. Both detection and tracking processes run in this mode. The node is operating its transceiver more often to receive broadcast messages from each of its neighbors and to transmit broadcast messages to its neighbors in independent time slots. There are frequent data messages.

Simulations and field experiments were performed to determine the usage patterns of the node components. These measurements, reported in the following sections, are the basis for the redesign of the node hardware to closely match the system requirements.

### 3.1 Radio Usage

Simulations to determine radio usage and make tradeoffs in the configuration of the *sas* medium access protocol parameters were performed and previously reported [5]. These simulation results are summarized in Table 2. The *sas* protocol manages communication between the nodes using a time division multiple access (TDMA) scheme. Dynamic frequency selection (DFS) and transmit power control (TPC) are included. Frequency and time slot hopping are employed to avoid interference. As configured for the surveillance application, the duration of a time slot is  $2500\mu\text{s}$  repeated at a frame interval of 1s. A guard time of  $1000\mu\text{s}$  is employed to account for time synchronization error between nodes. These parameters yield a maximum message size of about 180Bytes over the 1Mb/s channel.

By maintaining local control, the *sas* protocol allows extreme variation in communication usage at any point in the network. It can switch the configuration of the local network from a very low energy consumption state to a very low message delivery delay state in about 1-2s. In the quiescent state the *sas* protocol maintains a single fan-in receive slot on each node to receive very infrequent data traffic from neighboring nodes. Any incoming messages are acknowledged. This state yields extremely low energy consumption for the radio portion of the node—it is

**Table 3. Summary of prototype node processor usage.**

	Quiescent	Tracking
account	cpu time (%)	cpu time (%)
<b>Operating System Services</b>		
task scheduler	0.28	0.48
sleep	95.57	89.54
waste	1.21	3.06
<b>Location and Timing Services</b>		
location	0.00	0.00
compass	0.00	0.00
time synchronization	0.00	0.01
gps	0.03	0.00
<b>Communication Services</b>		
communication scheduler	2.37	4.97
forward error correction	0.18	0.65
sas	0.31	0.43
<b>Information Management Services</b>		
data cache	0.21	1.25
<b>Application</b>		
detector	0.04	0.05
tracker	0.00	0.04
status	0.00	0.00
<b>Test</b>		
test	0.05	0.00
TOTAL	100.00	100.00

operated for only 0.25% of the time—the number of received data messages and thus the number of transmitted acknowledgements are negligible.

When the node enters the tracking state, a broadcast transmit slot is created that allows low latency delivery of data messages to all neighbors. The creation of this broadcast transmit slot forces neighboring nodes to operate their receivers during the same slot, leading to correspondingly higher energy consumption. Thus every node operates its receiver during an additional time slot for each of its neighbors and may operate its transmitter in another additional time slot. These slots are often used. Assuming that there are 6 neighboring nodes, the communication usage on a node expands to include 7 additional slots, one for transmission and 6 for reception (with accompanying acknowledgements), leading to average slot usage of 2% on a node involved in tracking an intruder.

### 3.2 Processor Usage

Processor usage in each of these states is summarized in Table 3. Processor usage is measured by code embedded in the node's task scheduler. These summaries are prepared from two data segments each approximately 5 minutes long taken when there was no intruder activity for the quiescent case and about a minute after an intruder traversed the sensor field for the tracking case.

The row labeled "waste" accounts for time that the processor could have been sleeping, except that the interval was smaller

**Table 4. Prototype memory usage summary.**

Function	Program Memory (kB)	Data Memory (kB)
Operating System	39	5
Location and Timing	43	1
Communication	100	11
Information Management	118	2
Application	35	1
SUBTOTAL	335	20
Test	43	13
TOTAL	378	33

than the time required to change the processor state. The prototype processor is slow to enact state changes (approximately 20ms). More modern processors with faster state changes may recover most of this time into the sleep state. This wasted time is about 27% of the time that the node is not sleeping.

In both states, the processor is underutilized. The processor is active for less than 5% of the time in the quiescent state and less than 10% of the time in the tracking state. The processor speed may be easily reduced to about 7-14% of the current speed with a substantial amount of headroom. The prototype processor executes at 130MHz, so this implies a need for a processor operating at about 9-18MHz (assuming similar ratios of Mips/MHz). Removal of test and logging code and other software improvements, including the off-loading of several functions into dedicated hardware, may further reduce the need to about 1-5MHz.

### 3.3 Memory Usage

Measured memory usage is shown in Table 4. Except for the information management data memory, memory usage is static. Information management memory may grow to fill the remainder of the available data memory. Its growth is dependent upon the number of detections and tracks processed and retained by the node. This amount is affected by the user requirements and the activity in the sensor field. The user can set the data retention time for detection and track records. The number of detection and track events is determined by real world activity. The longer data is retained within the network and the higher the intrusion rate, the greater the number of retained records.

It should be noted that no effort was expended during software development to produce a compact code or data size. The prototype effort was focused on producing a system with the desired performance free of size constraints. In addition to the test modules clearly called out in the table, considerable test and debugging code is sprinkled throughout the other modules.

### 3.4 Energy Consumption

The energy consumption of each of the major subsystems of the prototype node has been measured in laboratory tests. In these tests, a regulated laboratory supply was connected to the battery terminals of the completed node in place of the regular battery. This supply was regulated to a voltage of 2.27V and current consumption was measured.

**Table 5. Energy consumption of the prototype and product nodes.**

Item	State	Prototype			Product		
		Peak Energy (mW)	Duty Cycle	Average Energy (mW)	Peak Energy (mW)	Duty Cycle	Average Energy (mW)
Processing/Memory		603.820	0.357143	215.650	45.000	0.052322	2.355
processor	fast	603.820	0.050000	30.191	45.000	0.050000	2.250
	slow	195.220	0.950000	185.459	0.110	0.950000	0.105
Communication		599.280	0.000995	0.596	247.500	0.001007	0.249
modem	idle	43.130	0.002200	0.095	33.000	0.002200	0.073
	tx	65.830	0.000200	0.013	71.280	0.000200	0.014
	rx	86.260	0.002000	0.173	42.240	0.002000	0.084
power amplifier	ps=a	556.150	0.000200	0.111	214.500	0.000200	0.043
low noise amplifier		102.150	0.002000	0.204	17.490	0.002000	0.035
Location/Timing		351.850	0.022796	8.021	270.600	0.002141	0.579
gps receiver	acq	324.610	0.006667	2.164	231.000	0.000278	0.064
	track	313.260	0.016667	5.221	125.400	0.003056	0.383
antenna/lna		27.240	0.023333	0.636	39.600	0.003333	0.132
Sensing		1.135	1.000000	1.135	0.660	1.000000	0.660
pir		1.135	1.000000	1.135	0.660	1.000000	0.660
magnetometer		63.560	0.000000	0.000	0.000	0.000000	0.000
temperature		0.000	0.003333	0.000	0.000	0.003333	0.000
battery		0.000	0.003333	0.000	0.000	0.003333	0.000
TOTAL		1556.085	0.144852	225.402	563.760	0.006817	3.843

Using embedded test software, node subsystems were enabled and disabled independently. When the subsystem could operate in more than one active state, all active states were independently tested. Current consumption was measured in all states, and the difference between the active state and the inactive state was used to determine the current consumption for each of the active states. This measurement technique ensures that all energy consumption of the subsystem (major components, ancillary components, passive parts, and supporting circuitry) are included in the reported values, which may differ from the values reported by the manufacturer of the major component.

During actual deployments, the node changes state quickly, activating and deactivating subsystems independently. Average energy consumption for the subsystems over any period of time is difficult to measure. We have estimated duty cycles for the various subsystems based on the known responses of the node software to external events (such as an intruder entering the field) and our estimate of the frequency of these events. The measured energy consumption, estimated duty cycles, and resulting average energy consumption of the prototype node subsystems are shown in Table 5. Low duty cycle software acting on low power components yields the low average energy consumption.

The values shown in the summary rows (gray) of this table require explanation. The “Peak Energy” column shows the highest energy consumption of that subsystem in any of its possible states; the “Average Energy” column is the sum of the average energy consumption of the components; and the “Duty Cycle” column is the ratio of the Average Energy to the Peak

Energy, a measure of the successful management of the energy consumption.

As shown by the table, the major area that must be addressed in the redesign is processor energy consumption. In a distant second place is the GPS receiver energy consumption, and then the sensor and radio subsystem, both of which are quite small.

The lifetime of the prototype node is estimated at slightly more than 1 day. Field deployments of the sensor nodes have verified that these estimates are reasonable—the prototype nodes operate in excess of 24 hours using 2 AA batteries as predicted.

#### 4. PRODUCT DESIGN

The node hardware has been redesigned based on the measurements made with the prototype system and experience developing the prototype. At this stage the uncertainties about the software suite have been eliminated allowing the hardware design to be driven by known requirements. The redesign is affected by the following major factors:

- Components are sized to fit the measured requirements.
- New, better components are available.
- A single board design reduces complexity and cost.
- Unused components have been eliminated.

The major components of the second generation, product node are shown in Table 1. There are fewer major components than for the prototype platform. This simplification lowers costs in several ways. First the major component cost is lowered. Secondly, substantial cost savings result from the elimination of the many

passive parts required to interface the major components and in the simplification of the circuit board fabrication and assembly. Board fabrication and assembly costs are often hidden and are also often larger than the cost of many of the components.

The second generation node is currently in development. The redesign has reduced the part count from 355 to 82, the number of signal nets from 1345 to 208, and the number of printed circuit boards from 3 to 1. The node measures 4cm x 5cm x 2cm. Completed units are expected by the middle of 2007.

The estimated energy consumption of the product design is shown in Table 5. These estimates assume the use of the same protocols, algorithms, and software as were used on the prototype node. Projections have been made to account for the lower energy consumption of the new components, off-loading of certain software functions from the processor to dedicated hardware, and code efficiency improvements.

The processor remains the largest consumer of energy, although its consumption is now much closer to that of the other components. This change is mostly the result of the small sleep state energy consumption and the speed with which the processor can transition from the active to the sleep state. The GPS receiver duty cycle has been reduced significantly (5x) to account for fixes in the over-the-air time synchronization algorithm, which effectively shares resynchronization with neighboring nodes. Energy consumption of the other components has improved slightly over the last 3 years, accounting for the modest improvements in the other subsystems.

Code efficiency improvements are expected across all of the major software modules, including substantial reduction or elimination of performance calculation and logging code, conversion from floating point to fixed point arithmetic, and elimination of unused or repetitive calculations. Secondly, several functions performed in software on the prototype node are either removed or off-loaded to dedicated hardware components for the product version. These functions include the forward error correction code and the detection processing algorithm. We estimate that the remaining software processing requirements will be about 1-5Mips. These improvements allow us to maintain the same duty cycle estimate for the processor, in spite of the reduction in maximum processing speed.

At this level of energy consumption the product node operates for about 90 days on 2 AA batteries. As the protocol software has effectively reduced the energy consumption of the 2 highest energy consumers—the communication and location/timing subsystems—meeting the processor duty cycle estimate is the key to realizing the predicted lifetime. This lifetime satisfies many deployment plans—an operational lifetime of 30 days is frequently specified.

## 4.1 Processor

The existence of a complete software suite and the measured performance of that suite on the prototype nodes allow us to estimate with assurance the processing power required in the product node. We intend to operate the same software suite on the product node with a few exceptions—all of which lower the processing requirements.

A number of highly integrated microprocessor devices are available, which typically include a processing core, program and

data memory, digital I/O lines, ADC, PLLs, clocks, and other peripherals.

The projected processing requirements of the system (1-5Mips) lie in the portion of the curves where the choice between an ARM7 and an ARM9 core is not clear cut. Both provide sufficient processing power. The ARM7 is more efficient at the lower end of our estimated processing requirement, while an ARM9 is more efficient at the upper end. Although it is very popular in mote designs, the MSP430 may not provide enough processing power and is less efficient.

We have selected the Atmel AT91SAM7S256. This device features an ARM7 core, 256kB of FLASH and 64kB of SRAM memory, an 8 channel, 10-bit ADC, and internal PLL supporting a 55MHz maximum clock rate. It can switch between the active and the ultra-low power sleep state in 1.5ms, a prime consideration for our intended operation. Its maximum clock rate is 50MHz at which it draws 45mW and in its sleep state it draws only 110 $\mu$ W.

We have selected this particular part mainly because of the high level of integration—that is, the inclusion of memory, ADC, and clock circuitry on the same chip as the processor core. Similar integrated products including ARM9 cores are under development. When these products are available, they may represent a better choice, as they allow the use of additional processing power and memory capacity without substantially increasing cost or energy consumption.

## 4.2 Memory

Program memory (256kB FLASH) and data memory (64kB SRAM) for the product sensor node are integrated with the processor described above. Separate components are not required. The elimination of memory bus interfaces and signal paths greatly simplifies the circuit board design, fabrication, and assembly.

As expected, the prototype software is too large to fit in most integrated processor and memory devices that typically include only small memories. We have selected a device with one of the larger memories of current devices. It still is not quite enough. The prototype software is approximately 30% larger than the available program memory.

We expect to rework portions of the prototype software to reduce code and data memory requirements. In particular, debug messages, hardware test functions, and performance logging functions are removed in the product version. These removals produce considerable shrinkage. If necessary, additional reductions are possible by removing unused code—the information management module includes a general purpose transaction and query parser and support for data types unused in the application and the communication module includes several alternative, experimental error recovery algorithms.

## 4.3 Radio

We have selected the Chipcon CC2500 for the product node. This component provides approximately the same RF performance as the part used in the prototype in a smaller package (16mm<sup>3</sup> vs. 49mm<sup>3</sup>) and with lower energy consumption (approximately 50% less for receive and idle states). The maximum data rate of the CC2500 is 500kB/s versus 1Mb/s for the CC2400. The required communication bandwidth for the surveillance system is a small fraction of both values as shown in Table 2.

We plan to dispense with the software Reed-Solomon block forward error correcting code (FEC) used in the prototype. As a substitute, the 16bit cyclical redundancy code (CRC) implemented in the CC2500 hardware is used. While the CRC provides less error detection capability than the longer FEC and no error correction capability, it is much cheaper to compute and appears to be sufficient as the *sas* protocol supplies link level acknowledgements on all data messages.

The transceiver and the processor are interfaced through a 4 signal serial peripheral interface (SPI) bus. The interface between the transceiver and the remainder of the RF electronics is the same as that used in the prototype.

We include an external power amplifier, low noise amplifier, and T/R switches to enhance range as was done in the prototype. We are replacing the T/R switches, low noise amplifier, and power amplifier circuitry with more compact and lower power components. The selected PA is the NEC UPG2314T5N. The selected LNA is the Maxim MAX2641. The selected T/R switch is the NEC UPG2214TB. The same embedded communication antenna, GigaAnt 3030A6111-01, is used.

#### 4.4 Timing

The product node uses two separate oscillators. An extremely low power, 32kHz RC oscillator is used to maintain timing during sleep states. This oscillator is calibrated with the GPS 1pps or over-the-air timing information as was done in the prototype.

While the prototype node used an accurate crystal oscillator, the product node uses an RC oscillator internal to the processor. This is because the RC oscillator consumes very little energy and the short term stability is sufficient. Low energy consumption is critical since this is the only element that is always active.

A 16MHz crystal oscillator is used for precision processor timing. This clock is used directly in low demand, active states. During high demand, the 16MHz clock is multiplied up to as much as 50MHz using a PLL internal to the processor. The PLL is only used when necessary because of the higher current consumption and the longer transition time required to stabilize the PLL.

#### 4.5 Location

A compatible GPS receiver from the same manufacturer, ublox NEO-4S, has been selected for the product node. It is considerably smaller (200mm<sup>2</sup> vs. 625mm<sup>2</sup>), has higher sensitivity (8dB increase), and consumes less energy (125mW average). While its hot-start acquisition time is the same as the current element, re-acquisition time is reduced to less than 1s. Unfortunately it is not cheaper.

We intend to operate the GPS receiver in the same manner as in the prototype node—the receiver is deactivated completely except when needed. We also expect the deactivated periods will be 5x longer, as problems in the over-the-air synchronization algorithm have been fixed. Thus the GPS receiver sleep state energy consumption is immaterial. The time to first fix, however, is a very important parameter as this determines how long the GPS receiver must be operated to resynchronize the node clock.

At this time, the manufacturer has announced an even better GPS receiver module, NEO-5S, which is reportedly both hardware and software compatible with the NEO-4S. Its reported acquisition time is under 1s by virtue of the extensive array of built in correlators. Availability of this module is expected in the second

quarter of 2007. If all claims are verified, the NEO-5S may be seamlessly substituted into the product design.

A slightly smaller, compatible GPS antenna, Zhengyuan Electric DAM1575C, has been selected. The dimensions are reduced from 25mm x 25mm to 18mm x 18mm to fit the slimmer profile case of the product node.

The digital compass circuitry was not used in tracking experiments performed with the prototype and has been dropped from the product node. It was hoped that the inclusion of a compass would allow more accurate localization of the target from the directional PIR sensor. Inaccuracy in the GPS reported node location and difficulty estimating range to the target from the PIR signal rendered this approach ineffective. Final tracking experiments used the node location as the reported location of the initial detection and relied on the tracker to refine the target location from these inaccurate initial reports.

#### 4.6 Sensor

Since the prototype node was designed, an integrated PIR, lens assembly, and detection processing module has been developed and marketed. This module, Panasonic AMN44122, provides the required functionality in a module that is smaller, more convenient, and of lower energy consumption than the custom circuitry used in the prototype. Furthermore, the module provides a digital detection output that is used to trigger an interrupt on the processor. Analog sampling of the PIR signal and software detection processing is no longer required. This change allows the processor to sleep for longer periods in the quiescent state, whereas previously it woke up at least once a second for detection processing.

This module is available in 3 variations, for long range, wide angle, and high sensitivity. We have selected the long range version for the standard sensor node. It has a detection range of 10m, which is somewhat shorter than that achieved by the prototype, but still sufficient for the system. Variations may be built with the wide angle sensor for interior use, where the long range capability is negated by the existence of interior walls.

This integrated module appears to be far more expensive than the custom detector used in the prototype. However, there are a number of hidden costs associated with the prototype sensor that make the costs more equal. Most important is the case design. A PIR sensor relies on an accurately positioned lens to focus the light on the sensor. In the prototype design the lens is mounted in the case making that custom part more complicated. The integrated module includes the lens properly mounted in front of the sensor. Assembly complexity is also reduced from 15 parts to 1 part.

#### 4.7 Power Supply

The product node uses the same basic power supply circuit as the prototype node. The 5V and 2.5V supplies are eliminated since no components requiring those voltages are present or likely. The 1.8V and 3.3V supplies continue to be available and are used to power components. The sizes of ancillary components (such as, inductors) have been reduced to the correct sizes.

### 5. CONCLUSIONS

Considerable processing power can be provided in a low-power sensor node without negatively impacting node energy consumption and lifetime. Processing power is available when

needed to execute application level software or to execute more complex communications algorithms. With modern processor cores that change from an active mode to a low-power sleep mode very quickly, excess processing power does not invariably lead to excess energy consumption. Extra processing power actually lowers overall energy consumption, as the increased computation allows better decision making and thus less use of more expensive resources. The reduction of communication of false alarms through local, collaborative processing and the local decision making that allows the *sas* protocol to make rapid changes between communication schedule extremes to suit current conditions are two examples.

One of the controversial aspects of the node design may be the inclusion of a power amplifier in the radio circuit. Mote designs trying to achieve the smallest size, lowest cost, and lowest energy consumption do not include a power amplifier. However, these designs also exhibit very short communication ranges in surveillance deployments (5m), leading to very high node density requirements. We do not believe these deployments are realistic. Cost per area covered is a more appropriate cost measure for a surveillance system than node cost. Communication and detection range must be balanced as both impact the required node density. Using this cost measure and with the detection range of the node sensor, the power amplifier makes sense.

Another controversial aspect of the design may be the inclusion of a GPS receiver. GPS receivers are often viewed as large, expensive, high energy devices and many researchers are exploring alternative ways to determine node location and time synchronization. While it remains true that the GPS receiver is a significant budget issue, its energy consumption can be carefully controlled so that using a GPS receiver is an energy efficient method to provide node location and time synchronization. This is likely to become even more true in the future, as GPS receiver technology is rapidly becoming smaller, more sensitive, and cheaper in response to demand from the cell phone industry.

While we believe the sensor node design is appropriate for many missions, the design has been optimized for a particular surveillance system, and other tradeoffs may be more appropriate for other systems. In particular, if a system requires substantially more processing power for application processing, a higher capability processor may be appropriate. Integrated processing and memory devices similar to the one selected for this design with higher capability cores are under development and may be more suitable. Similarly, if the detection system is designed for a very short range, it may be desirable to adjust the communication link range by removing the LNA and PA from the radio circuit. This removal simplifies and shrinks the node design by eliminating many parts.

## 6. REFERENCES

- [1] Arora, A., Dutta, P., Bapat, S., Kulathumani, V., Zhang, H., Naik, V., Mittal, V., Cao, H., Demirbas, M., Gouda, M., Choi, Y-R., Herman, T., Kulkarni, S. S., Arumugam, U., Nesterenko, M., Vora, A. and Miyashita, M. A Line in the sand: A wireless sensor network for target detection, classification, and tracking. *Computer Networks (Elsevier)*, 46, 5, 2004, 605-634.
- [2] Brooks, R., Griffin, C., Friedlander, D., Self-Organized Distributed Sensor Network Entity Tracking. *International Journal of High Performance Computing Applications*, 16, 3, (2002), 207-219.
- [3] Crossbow Micaz Data Sheet, Crossbow Document Part Number: 6020-0060-03 Rev A, www.xbow.com.
- [4] Dust Networks' M2030 Data Sheet, www.dust-inc.com.
- [5] Hammel, T., Rich, M., Graff, C. Design of a Sensor Protocol Suite for VLSI Implementation. In *Military Communications Conference, 2005 (MILCOM 2005)* (Atlantic City, NJ, October 20, 2005). IEEE Press, 2005, Vol. 4, 2421-2426.
- [6] Hammel, T., Yetso, B., and Rogers, T., Fusing live sensor data into situational multimedia views. In *Proceedings of the 9th International Workshop on Multimedia Information Systems* (Ischia, Italy, 2003). 2003, 145-156.
- [7] Hill, J., Culler, D., Mica: A wireless platform for deeply embedded networks, *IEEE Micro*, 22, 6, (2002) 12-24.
- [8] Hill, J., Horton, M., Kling, R., Krishnamurthy, L., The platforms enabling wireless sensor networks, *Communications of the ACM*, 47, 6 (June 2004), 41-46.
- [9] IEEE Std 802.15.4™-2003, Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs), 1 October 2003, ISBN 0-7381-3677-5 SS95127.
- [10] Langendoen, K., Reijers, N., Distributed localization in wireless sensor networks: a quantitative comparison, *Computer Networks, special issue on Wireless Sensor Networks*, 2, (2003).
- [11] Ledeczi *et al.*, Countersniper system for urban warfare, *ACM Transactions on Sensor Networks*, 2, (Nov 2005), 153-177.
- [12] Li, D., Wong, K., Hu Y., Sayeed, A., Detection, Classification and tracking of targets in distributed sensor networks, *IEEE Signal Processing Magazine*, 19, 2, (March 2002) 17-29.
- [13] Merrill, W., Sohrabi, K., Girod, L., Elson, J., Newberg, F., Kaiser, W., Open Standard Development Platforms for Distributed Sensor Networks, *Proceedings of SPIE Aero Sense Conference 2002*.
- [14] McIntire, D. *Energy Benefits of 32-bit Microprocessor Wireless Sensing Systems*. Technical Note, Sensoria Corporation, Culver City, California.
- [15] Morton, G., *MSP430 Competitive Benchmarking*, Texas Instruments Application Report SLAA205A, January 2005.
- [16] Nachman, L., Kling, R., Adler, R., Huang, J., Hummel, V., The Intel Mote platform: a Bluetooth-based sensor network for industrial monitoring, *Proceedings of the 4th international symposium on Information processing in sensor networks*, April 24-27, 2005 (Los Angeles, California, USA, 2005) 61.
- [17] Oh, S., Chen, P., Manzo, M., and Sastry, S. Instrumenting Wireless Sensor Networks for Real-time Surveillance. In Proc. of the International Conference on Robotics and Automation (Orlando, FL, May 2006).
- [18] Zigbee Specification, ZigBee Document 053474r06, Version 1.0 December 14th, 2004.