Design and Implementation of Wireless Sensor Networks for Habitat Monitoring

by Joseph Robert Polastre

Research Project

Submitted to the Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, in partial satisfaction of the requirements for the degree of **Master of Science, Plan II**.

Approval for the Report and Comprehensive Examination:

Committee:

Professor David Culler
Research Advisor

(Date)

* * * * * * *

Professor Eric Brewer
Second Reader

(Date)

Design and Implementation of Wireless Sensor Networks for Habitat Monitoring

Copyright Spring 2003

by

Joseph Robert Polastre

Abstract

Design and Implementation of Wireless Sensor Networks for Habitat Monitoring

by

Joseph Robert Polastre

Master of Science in Computer Science

University of California at Berkeley

Professor David Culler, Research Advisor

We provide an in-depth study of applying wireless sensor networks to real-world habitat monitoring. A set of system design requirements were developed that cover the hardware design of the nodes, the design of the sensor network, and the capabilities for remote data access and management. We propose a system architecture that addresses these requirements for habitat monitoring in general. We present an in-depth discussion of the implementation of the architecture for habitat monitoring. In the summer of 2002, 32 nodes were deployed on a small island off the coast of Maine streaming useful live data onto the web using our implementation. Results from the deployment show the profound impact software and hardware power management has on node longevity. The effectiveness of the system architecture is shown through the packet throughput and through the delivery of over 1.2 million readings logged at our database in Berkeley. The system operated for over four months; it provided data for two months after researchers had left the island for the winter due to poor weather conditions. The application-driven design exercise serves to identify important areas of further work in power management, data sampling, communications, network retasking, and health monitoring. We discuss the lessons learned from our deployment and provide a series of solutions that include new hardware, software, and protective enclosures.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The emergence of wireless sensor networks has enabled new classes of applications that benefit a large number of fields. Wireless sensor networks have been used for fine-grain distributed control [42], inventory and supply-chain management [39], and environmental and habitat monitoring [34].

Habitat and environmental monitoring represent a class of sensor network applications with enormous potential benefits for scientific communities. Instrumenting the environment with numerous networked miniature sensors can enable long-term data collection at scales and resolutions that are difficult, if not impossible, to obtain otherwise. Miniature sensor networks can instrument at the scale of the organism and be deployed in dense clusters. A sensor's intimate connection with its immediate physical environment allows each sensor to provide localized measurements and detailed information that is hard to obtain through traditional instrumentation. The integration of local processing and storage allows sensor nodes to perform complex filtering and triggering functions, as well as to apply application-specific or sensor-specific aggregation, filtering, and compression algorithms. The ability to communicate not only allows sensor data and control information to be communicated across the network of nodes, but nodes to cooperate in performing more complex tasks, such as statistical sampling, data aggregation, and system health and status monitoring [14, 15]. Low-power radios with well-designed protocol stacks allow generalized communications among network nodes, rather than point-to-point telemetry. The computing and networking capabilities allow sensor networks to be reprogrammed or retasked after deployment in the field. Nodes have the ability to adapt their operation over time in response to changes in the environment, the condition of the sensor network itself, or the scientific endeavor. Finally, sensor network nodes are built from low-cost components that benefit from Moore's Law–instead of increasing CPU power, Moore's Law applies to reduce the size and cost. This property directly affects sensor nodes, which one day may be the size of a dust mote and cost only a few cents [21, 28].

Biologists at the University of California and the College of the Atlantic propose that ultimately sensor networks should be scaled to the size of the organisms under study, sample data at frequencies equivalent to environmental changes that organisms encounter, and deployed to capture the full range of the organism's environmental exposure. Only sensor networks with these properties can provide the appropriate fine-grain information needed for accurate modeling and prediction [4, 20, 44]. In order to deploy dense wireless sensor networks capable of recording, storing, and transmitting microhabitat data, a complete system composed of communication protocols, sampling mechanisms, and power management must be developed. In order to develop wireless sensor networks for habitat monitoring, we let the application drive the system design agenda. A system may be designed based on the requirements of biologists that will utilize this system. The application-driven model includes an iterative design cycle to provide researchers with early feedback on system design. The result is frequent iterations that lead to a more sophisticated and powerful architecture for monitoring applications. Taking an application-driven approach quickly separates actual problems from potential ones, and relevant issues from irrelevant ones. The application context helps to differentiate problems with simple, concrete solutions from open research areas.

Our goal is to develop an effective sensor network architecture for the domain of monitoring applications, not just one particular instance. Collaboration with scientists in other fields helps to define the broader application space, as well as specific application requirements, allows field testing of experimental systems, and offers objective evaluations of sensor network technologies. The impact of sensor networks for habitat and environmental monitoring will be measured by their ability to enable new applications and produce new results otherwise too difficult to realize.

This thesis develops a specific habitat monitoring application, one that is largely representative of the domain. It presents a collection of requirements, constraints, and guidelines that serve as a basis for a general sensor network architecture for habitat and environmental monitoring. Included in this architecture are the hardware and sensor platforms, the distinct networks involved in data transport, their interconnection, and data management facilities. The design and implementation of essential network services, including power management, communications, retasking, and node management may be evaluated in this context. A simple solution to the habitat monitoring application that meets the requirements of life scientists and provides insight into the behavior of long term sensor network deployments will meet our research goals.

The rest of this thesis is organized as follows. Chapter 2 discusses the requirements of habitat monitoring and provides information about the specific deployment on Great Duck Island discussed at length in this thesis. Chapter 3 describes the system architecture we developed to satisfy the requirements of habitat monitoring applications. The implementation of the architecture from Chapter 3 is discussed in Chapter 4. We discuss the services needed by habitat monitoring applications and their impact on sensor node power management in Chapter 5. The results of the deployment on Great Duck Island and their relationship to the requirements of habitat monitoring applications is described in Chapter 6. Chapter 7 discusses components of the system that were changed after evaluating the results of the first habitat monitoring deployment. Chapter 8 discusses the related work that uses wireless sensor networks for habitat monitoring. Chapter 9 provides concluding remarks.

Chapter 2

Habitat Monitoring

Life scientists are increasingly concerned about the potential impacts of direct human interaction in monitoring plants and animals in field studies. At best, it is possible that chronic human disturbance may distort results by changing the behavioral patterns of the animal. In the worst case, anthropogenic disturbance can seriously reduce, destroy, or stress a breeding species. Although the effects of disturbances are usually immediately noticed in animals, plant populations are sensitive to trampling by even the most well-intended researchers.

Traditional data loggers for habitat monitoring are typically large in size and expensive. They require that intrusive probes be placed in the area of interest and the corresponding equipment be immediately adjacent. Life scientists typically use these data loggers since they are commercially available, supported, and provide a variety of sensors. One such data logger is the Hobo Data Logger [38] from Onset Corporation. Due to size, price, and organism disturbance, using these systems for fine-grained habitat monitoring is inappropriate.

Other habitat monitoring studies install one or a few sophisticated weather stations an “insignificant distance” (as much as tens of meters) from the area of interest. A major concern with this method is that biologists cannot gauge whether the weather station actually monitors a different microclimate due to its distance from the organism being studied [20]. Using the readings from the weather station, biologists make generalizations through coarse measurements and sparsely deployed weather stations.

For example, seabird colonies are extremely sensitive to human interaction. In Maine, biologists have shown that even a 15 minute visit to a colony can result in up to 20% mortality among eggs and chicks in a given breeding year [3]. If the disturbance is repeated, the entire colony may abandon their breeding site. Current methods for monitoring seabirds involve using invasive infrared cameras and physically disturbing or removing a seabird from its nest in order to determine the nest's occupancy. On Kent Island, Nova Scotia, researchers found that the Leach's Storm Petrel, the seabird that we monitor in this study, is likely to leave their nesting burrows if disturbed during their first two weeks of incubation. Additionally, the hatching success of petrel eggs was significantly influenced by the frequency of investigator disturbance. In a recent study by Alexis Blackmer, she found that the hatching success was reduced by 56% in petrel burrows that were disturbed on a daily basis (and 50% less for burrows disturbed on a weekly basis) compared to a control group that was not disturbed for the duration of their breeding cycle [6].

Instead of large data loggers, weather stations, and invasive physical monitoring, a revolution for biologists would be the ability to monitor the environment on the scale of the organism, not on the scale of the biologist [4, 44]. A dense deployment of sensors in the area of interest can provide high fidelity readings that were previously impossible to obtain.

In order to deploy sensors densely to monitor a habitat, the sensors must be small in size to minimize the disturbance effects caused by traditional methods. Disturbance effects are of particular concern in small island situation, where it may be physically impossible for researchers to avoid impacting an entire population. Islands often serve as refugia for species that cannot adapt to the presence of terrestrial mammals, or may hold fragments of once widespread populations.

Monitoring petrel activity must occur on a fine-grain spatial scale. Probes and traditional methods may create a "shadow effect"–a situation that occurs when an organism alters its behavioral patterns due to an interference in their space or lifestyle [37]. Instead, biologists argue for the miniaturization of devices that may be deployed on the surface, in burrows, or in trees. Since interference is such a large concern, the sensors must be inconspicuous. They should not disrupt the natural processes or behaviors under study [7]. To analyze the patterns of the organisms, the sensors must be deployed both in the organisms' burrows as well as on the surface surrounding the burrows to monitor the differences. In order to determine why petrels nest in specific patches, data should be gathered from both populated and unpopulated petrel patches.

Sensor networks represent a significant advance over traditional invasive methods of monitoring. Sensors can be deployed prior to the sensitive period (*e.g.*, breeding season for animals, plant dormancy period, or when the ground is frozen for botanical studies). Miniature sensors may be deployed on small islets where it would be unsafe or unwise to repeatedly attempt field studies. Miniature sensors are smaller in volume and do not intrude through the existing wall of the burrow like probes.

Deploying sensor networks is a substantially more economical method for conducting long-term studies than traditional personnel-rich methods. It is also more economical than installing many large data loggers. Currently, field studies require substantial maintenance in terms of logistics and infrastructure. Since sensors can be deployed and left, the logistics are reduced to initial placement and occasional servicing. Sensor networks may greatly increase access to a wider array of study sites that are often limited by concerns about disturbance due to frequent access and lack of habitability.

2.1 Great Duck Island

Great Duck Island (GDI), located at 44.09N, 68.15W, is a 237 acre island located 15 km south of Mount Desert Island, Maine. The Nature Conservancy, the State of Maine, and the College of the Atlantic hold much of the island in joint tenancy. Great Duck contains approximately 5000 pairs of petrels, nesting in discrete "patches" within the three major habitat types (spruce forest, meadow, and mixed forest edge) found on the island [1]. The College of the Atlantic (COA) is field testing in-situ sensor networks for habitat monitoring on GDI. COA has ongoing field research programs on several remote islands with well established on-site infrastructure and logistical support. John Anderson, a professor at COA, studies the Leach's Storm Petrel on GDI. Professor Anderson and other seabird researchers are interested in four major questions:

1. What is the usage pattern of nesting burrows over the 24-72 hour cycle when one or both members of a breeding pair may alternate incubation duties with feeding at sea?

2. What environmental changes occur inside the burrow and on the surface during the course of the seven month breeding season (April-October)?

3. What is the variation across petrel breeding sites? Which of these conditions yield an optimal microclimate for breeding, incubation, and hatching?

4. What are the differences in the micro-environments between areas that contain large numbers of nesting petrels and those areas that do not?

It is unlikely that any one parameter recorded by wireless sensors could determine why petrels choose a specific nesting site, rather predictive models may be developed by making multiple measurements of many variables or sensors. These models can be used to correlate which conditions seabirds prefer.

2.2 Monitoring Requirements

In order to build a system for monitoring the Leach's Storm Petrel on GDI, we must identify the requirements of the system and integrate technical solutions for each requirement.

The spatial scale of the instrumentation is a primary requirement for life scientists. By increasing the size of the monitoring area and the number of discrete sampling locations, we can obtain data at fidelities and densities not possible using traditional methods. Using data from a dense network of sensors, life scientists can develop predictive models describing organism behavior.

The temporal scale of the instrumentation must be on the order of the organism. Sensors should collect data at a rate equal or greater to changing environmental conditions that the organism experiences. Traditional data collection systems calculate the average, minimum, and maximum over 24 hour periods as well as time series data. Instead, data analysis must be able to identify not only changes but also the *duration* of events. Some examples applicable to GDI are:

- What is the duration of various temperature gradients or ambient light levels?
- How long does a petrel stay in the nest?
- What is the length of the incubation, hatching, and feeding periods?

Data analysis from sensor networks should be able to detect changes and log not only the values of the changes but the durations over which they occur.

The sensors should be deployed in areas that contain breeding petrel burrows approximately one month prior to the initial courtship period of the petrel reproductive cycle. After the petrel begins incubating the egg, there must be no human interference to the nest for an average of 43 days. After hatching, the nestling is brooded for an average of six days, after which the chick remains alone in the burrow for 55-65 days. Overall, the petrel cycle lasts approximately 5 months [24]. In order to monitor an entire field season (corresponding to a single petrel reproductive cycle), the sensors must budget their power to provide at least 7 months of continuous operation without changing batteries or physically accessing the sensors. It is important that the nodes efficiently manage their power consumption through low duty-cycle operation. All of the components in the deployed system must operate in accordance with the power budget in order to meet the required system lifetime. Node longevity is essential to a successful deployment. The system must operate autonomously during the deployment period providing reliable, predictable operation. Not only is it difficult to debug and maintain an unpredictable system, reliability and predictability are essential for developing trust in new technologies among life scientists.

The sensors should be deployed in a manner to cover as many petrel burrows as possible.

Burrows typically occur in discrete “patches”. The patches are geographically dispersed around the island. Each patch must relay its data to be collected. One sensor node per burrow is sufficient for data sampling. Each patch may contain over 50 burrows; a large number of these burrows should be populated with sensors. Some should be left unpopulated to be able to evaluate the disturbance effects caused by wireless sensors.

A variety of environmental monitoring sensors must be included on a small device. The device should be as small as possible so that disturbance effects are minimized. Biologists are primarily interested in monitoring the light, temperature, pressure, humidity, and occupancy changes that occur both inside and outside petrel burrows. These sensors may be used to analyze the difference between populated and unpopulated burrows. It may also be used to build predictive models of the burrow microclimate both spatially and temporally. The unique combination of sensors on a single device may be used for a variety of aggregate operations. An infrared sensor may be used in conjunction with temperature and light sensors to detect cloud cover [11]. Infrared sensors may also be used to detect occupancy, measure the temperature of a nearby object (for example, a bird or a nest), and sense changes in the object’s temperature over time. An absolute pressure sensor may be used as a relative altimeter, or absolute altimeter if the initial deployment altitude is known. Strategically placed sensor boards with pressure sensors can detect the wind speed and direction by modeling the wind as a fluid flowing over a series of apertures (one such method is described in [5]).

In order to verify the data received from the networked sensors, there must be a method of validation. Life scientists need to contrast new sensor technologies with traditional systems considered to be ground truth. The sensors must be calibrated and accurate, and some sensors should be deployed next to traditional equipment. On Great Duck Island, Professor Anderson advocates using miniature wired cameras to verify the output of an occupancy sensor.

The system must be able to tolerate disconnected operation at every level. The sensor nodes and satellite/WAN connection may periodically lose connectivity. Data must be stored reliably such that disconnected operation does not result in data loss.

The data received through the sensor network should be available to the larger scientific community for independent analysis. Streaming data to the web allows students and researchers to access and learn from remote sites that they may not otherwise be able to visit. Online live data shortens the design loop and iteration cycles since the results of changing sensing modalities is instantaneously available. A replicated database accessible via the Internet must archive the data for later analyses and protect against failure and loss.

2.3 Current Status

Thirty-two motes were deployed on Great Duck Island in the summer of 2002. Nine of these motes are monitoring underground burrows. After deployment, life scientists set out to verify the accuracy of the sensor readings. On several occasions, after noticing changes in the occupancy of a burrow, a recorded petrel call was played back above the instrumented burrows. A petrel called back to the recording, indicating that a bird was indeed present. Additional data analysis is provided in Chapter 6.

During the deployment, the data was graphed live on the web using a java applet and database at `http://www.greatduckisland.net`. The java applet is shown in Figure 1. The data is replicated in Berkeley for complex data mining and analysis. The live applet and replicated database serve as a tool for scientists to investigate the behavior of the Leach’s Storm Petrel and the environment on Great Duck Island, as well as a learning tool for students to see the direct effects of the current climate conditions on animal behavior (such as occupancy). The data is extremely useful for evaluating long term sensor network deployments. The remainder of this paper discusses the architecture, implementation, and system analysis of how the Great Duck Island habitat monitoring system was developed and deployed.

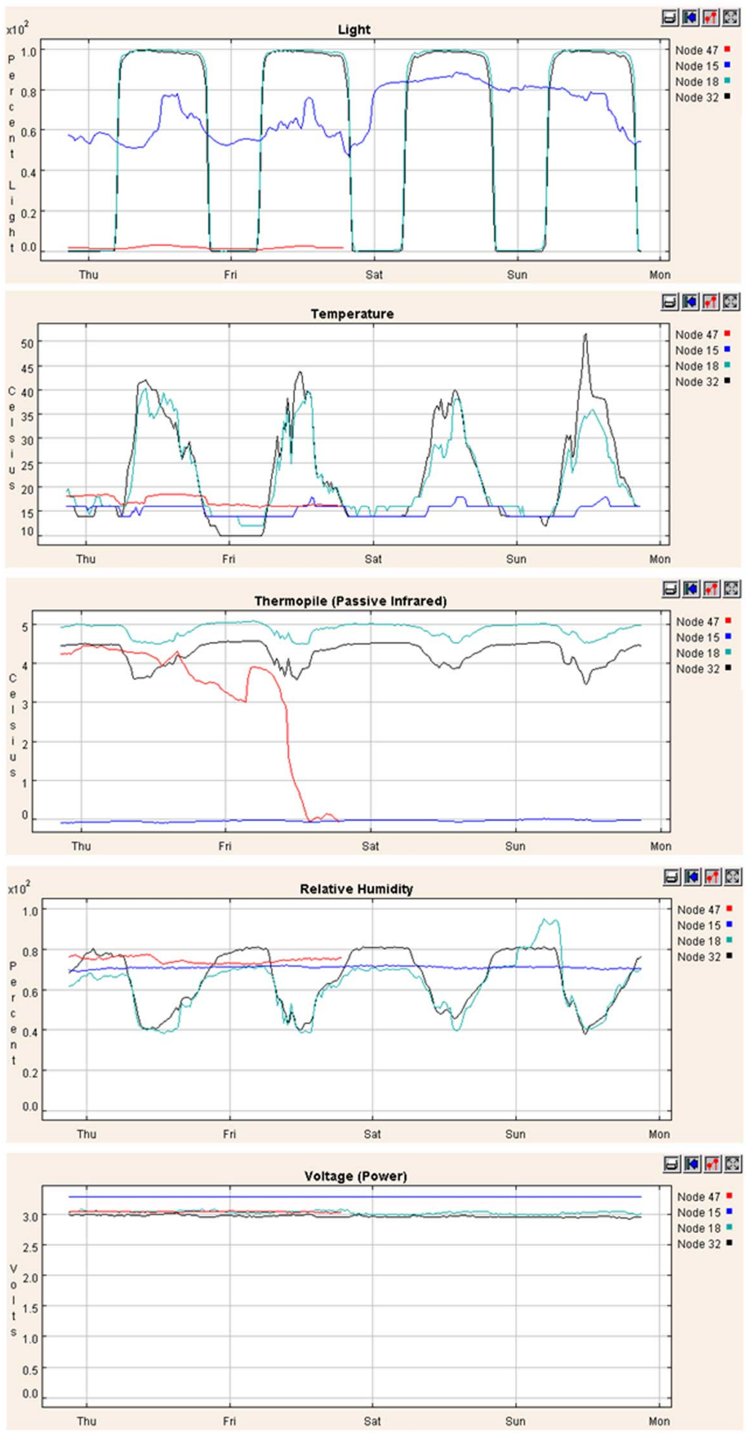


Figure 1: Java applets displaying the live readings of various sensors deployed on Great Duck Island (www.greatduckisland.net)

Chapter 3

System Architecture

In order to deploy a network that satisfies the requirements of Chapter 2, we developed a system architecture for habitat monitoring applications. Here, we describe the architecture, the functionality of individual components, and the interoperability between components.

The system architecture for habitat monitoring applications is a tiered architecture. Samples originate at the lowest level that consists of *sensor nodes*. These nodes perform general purpose computing and networking in addition to application-specific sensing. Sensor nodes will typically be deployed in dense *sensor patches* that are widely separated. Each patch encompasses a particular geographic region of interest. The sensor nodes transmit their data through the sensor patch to the sensor network *gateway*. The gateway is responsible for transmitting sensor data from the *sensor patch* through a local transit network to the remote *base station* that provides WAN connectivity and data logging. The base station connects to database replicas across the Internet. Finally, the data is displayed to scientists through any number of user interface. Mobile devices may interact with any of the networks–whether it is used in the field or across the world connected to a database replica. The full architecture is depicted in Figure 2.

The lowest level of the sensing application is provided by autonomous *sensor nodes*. These small, battery-powered devices are placed in areas of interest. Each sensor node collects

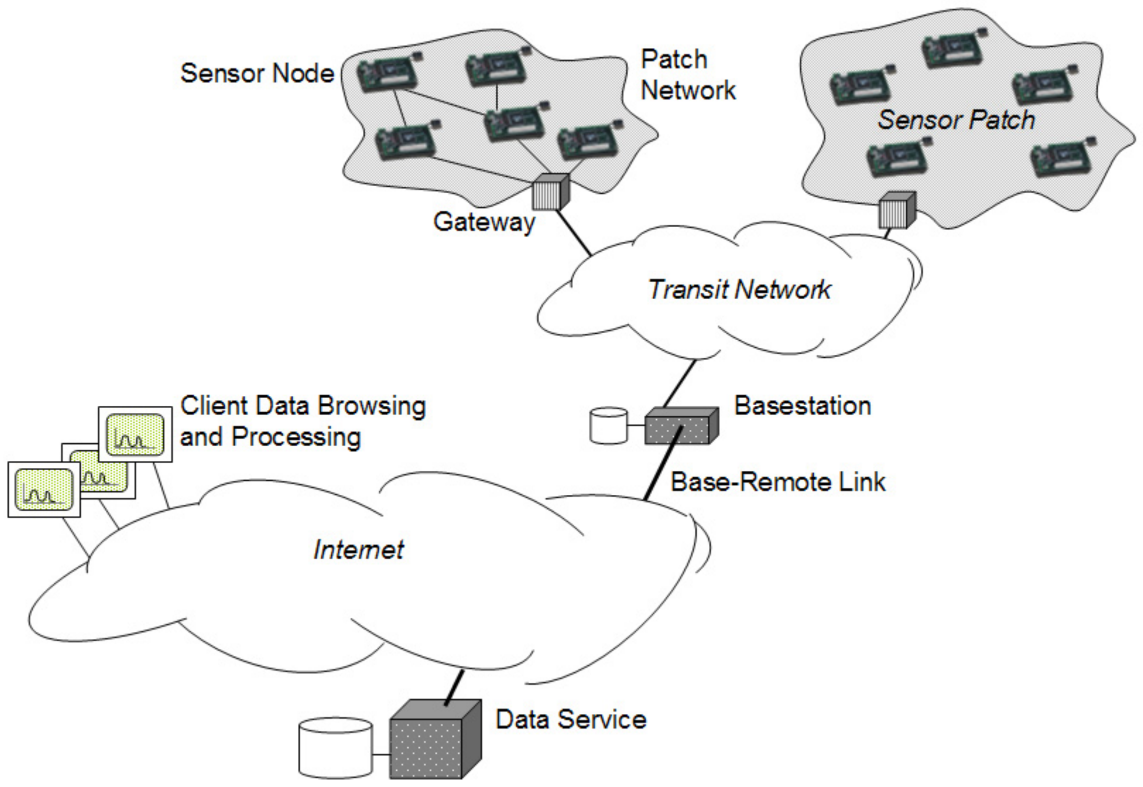


Figure 2: System architecture for habitat monitoring

environmental data primarily about its immediate surroundings. Because it is placed close to the phenomenon of interest, the sensors nodes may be built using small and inexpensive individual sensors. High spatial resolution can be achieved through dense deployment of sensor nodes. Compared with traditional approaches, which use a few high quality sensors with sophisticated signal processing, this architecture provides higher robustness against occlusions and component failures.

The sensor node computational module is a programmable unit that provides computation, storage, and bidirectional communication with other nodes in the system. It interfaces with analog and digital sensors on the sensor module, performs basic signal processing (*e.g.,* simple translations based on calibration data or threshold filters), and dispatches the data according to the application's needs. Compared with traditional data logging systems, it offers two major advantages: it can easily communicate with the rest of the system and it can be *retasked* in the field discussed further in Section 5.4.

Individual sensor nodes communicate and coordinate with one another in the same geographic region. This coordination makes up the *sensor patch*. The sensor patches are typically small in size (tens of meters in diameter); in our application they correspond to petrel nesting patches.

Using a multi-tiered network is particularly advantageous since each habitat involves monitoring several particularly interesting areas, each with its own dedicated sensor patch. Each sensor patch is equipped with a *gateway* which can communicate with the sensor network. The gateway provides a bridge to the commercial wireless local-area network (WLAN) for the sensor patch by connecting to the base station through the transit network. The gateway and transit network make it possible to deploy extremely small devices with miniscule batteries. By relying on the gateway, sensor nodes may extend their lifetime through extremely low duty-cycles. In addition to providing connectivity to the base station, the gateway coordinates the activity within the sensor patch, and provides additional computation and storage. The extra resources at the gateway come at a cost—providing enough energy to sustain the gateway unit requires photovoltaic cells roughly the size of a car battery.

Sensor nodes may provide transit network connectivity. A series of nodes are placed along the path between the sensor patch and the base station; each node in the series acts as a relay. Transit network robustness could be achieved though redundant connectivity. Each connectivity design has different characteristics with respect to expected robustness, bandwidth, energy efficiency, cost, and manageability.

Ultimately, data from each sensor needs to be propagated to the Internet. The propagated data may be raw, filtered, or processed. Bringing direct wide area connectivity to each sensor patch is not feasible–the equipment is too costly, it requires too much power and the installation of all required equipment is quite intrusive to the habitat. Instead, the wide area connectivity is brought to a *base station*, where adequate power and housing for the equipment is provided. The base station communicates with the sensor patch using a WLAN. The WLAN access point that communicates with gateways is co-located with the base station. To provide data to remote end-users, the *base station* includes wide area network (WAN) connectivity and persistent data storage for the collection of sensor patches. Since many habitats of interest are quite remote, we expect that the WAN connection will be wireless (*e.g.,* two-way satellite). The components must be reliable, enclosed in environmentally protected housing, and provided with adequate power. In many environments such conditions can be provided relatively easily at a ranger station.

Data reporting in our architecture may occur both spatially and temporally. In order to meet the network lifetime requirements, nodes may operate in a phased manner. Nodes primarily sleep, periodically sample, perform necessary calculations, and then send or relay readings through the network at intervals. Data may travel spatially through various routes in the sensor patch, transit network, or wide area network; it is then routed over long distances to the wide area infrastructure.

Each layer of the network architecture must address the possibility of disconnected operation. Provisioning for disconnected operation allows layers of the network to operate independently from the others. The sensor nodes, gateway, and base station contain non-volatile storage. Each node maintains its own local cache. In the case of disconnected operation, data is stored. Upon reconnection with other layers of the network architecture, the caches are synchronized such that data loss is minimized. At the sensor level, data management primitives may be primitive, taking the form of data logging. The base station will often offer full-fledged relational database service. The data management at the gateways will fall somewhere in between, offering some database services, but perhaps over limited window of data. Although many types of communication can be unreliable, when it comes to data collection, long-latency is preferable to data loss. For this kind of communication, a “custody transfer” model, similar to SMTP messages or bundles [16], may be applicable.

Users interact with the sensor network data in two ways. Remote users access the replica of the base station database (in the degenerate case they interact with the database directly). This approach allows for easy integration with data analysis and mining tools, while masking the potential wide area disconnections with the base stations. Remote control of the network is also provided through the database interface. Although this control interface is is sufficient for remote users, on-site users may often require a more direct interaction with the network. Small, PDA-sized devices enables such interaction. The data store replicates content across the WAN and its efficiency is integral to live data streams and large analyses.

Chapter 4

Implementation of the Habitat Monitoring Architecture

The architecture in Chapter 3 describes the components of the network needed to meet habitat monitoring requirements. In this chapter, the implementation of the habitat monitoring architecture is described. We discuss the sensor node including the hardware and software design. The sensors are enclosed in protective packaging and transmit their readings to a gateway node. We analyze two implementations for the gateway and transit network by examining power consumption and robustness; then we describe the base station installation and database structure.

4.1 Sensor Network Node

In our deployment, we are using UC Berkeley *motes* as the sensor nodes. The latest member of the mote family, called *Mica* [22] (shown in Figure 3 and Table 1), uses a single channel, 916MHz radio from RF Monolithics to provide bidirectional communication at 40kbps, an Atmel ATmega128 microcontroller running at 4MHz, and a considerable amount of nonvolatile storage (512 KB). A pair of conventional AA batteries and a DC boost converter provide a stable voltage source, though other renewable energy sources can be easily used. Small size (approximately

2.0 x 1.5 x 0.5 inches) and wireless communication capabilities allow us to deploy motes in remote locations with minimal interference with the existing habitat.

The UC Berkeley mote family (see Table 1) has evolved over the past three years into a stable platform for sensor networks research. There is a strong software base to build applications, a composible operating system called TinyOS [23, 45], and a programming language for networked embedded systems called nesC [17]. Due to the open availability of the hardware and software, and the match between the system requirements and the properties of the *Mica* mote, we chose to use the UC Berkeley platform running TinyOS.

An important aspect of the *Mica* mote running TinyOS is the ability to set low level hardware functionality to achieve low power sleep states. Since sensor nodes are expected to spend most of the time sleeping and periodically sample, compute, and communicate, minimizing the sleep current is essential (as shown in Section 5.1) to meeting the system lifetime requirements. Minimizing power in sleep mode involves turning off the sensors, the radio, and putting the processor into a deep sleep mode. I/O pins on the microcontroller need to be put in a pull-up state whenever possible, as they can contribute as much as 100 μA of leakage current. *Mica* architecture uses a DC booster to provide stable voltage from degrading alkaline batteries. With no load, the booster draws between 200 and 300 μA, depending on the battery voltage. Although this functionality is crucial for predictable sensor readings and communications, it is not needed in the sleep mode. Furthermore, the current draw of the microprocessor is proportional to the supply voltage. We modified *Mica* motes with a Schottky diode, which allows us to reliably bypass the DC booster while reducing the supply voltage in sleep modes. The modification allows us to achieve between 30 and 50 μA current draw (battery dependent).

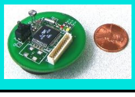
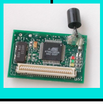
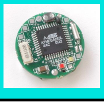
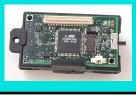
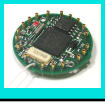
Mote Type	*WeC*	*René*	*René 2*	*Dot*	*Mica*	*MicaDot*
						
Microcontroller						
Type	AT90LS8535		ATmega163		ATmega128	
Program memory (KB)	8		16		128	
RAM (KB)	0.5		1		4	
Nonvolatile storage						
Chip	24LC256				AT45DB041B	
Connection type	I^{2}C				SPI	
Size (KB)	32				512	
Default power source						
Type	Lithium	Alkaline	Alkaline	Lithium	Alkaline	Lithium
Size	CR2450	2 x AA	2 x AA	CR2032	2 x AA	3B45
Capacity (mAh)	575	2850	2850	225	2850	1000
Communication						
Radio	TR1000					CC1000
Radio speed (kbps)	10	10	10	10	40	38.4
Modulation type	OOK				ASK	FSK

Table 1: The family of UC Berkeley motes and their capabilities

In our implementation, the communication protocols are simplified as much as possible while still meeting the data delivery requirements. Each sensor node acts as a transmit-only device in a single-hop broadcast network. The data is received by the gateway node that operates with significantly more energy capacity than the small sensor nodes. In order to extend the patch to more burrows beyond the single-hop broadcast range, the sensors may form a multihop wireless network by forwarding each other's messages. Using a multihop topology vastly extends connectivity options. If appropriate, the network can perform in-network aggregation (*e.g.*, reporting the average temperature across a region). A flexible communication structure allows enables network design that delivers the required data while meeting the energy requirement of operating for an entire field season. We discuss sensor node power management solutions and energy efficient communication protocols in Chapter 5.

4.2 Sensor Board

To provide meaningful data to scientists, we designed and manufactured an environmental monitoring sensor board, shown in Figure 3. The *Mica* Weather Board provides sensors that monitor changing environmental conditions with the same functionality as a traditional weather station. The *Mica* Weather Board integrates the required sensors into a single small package. The board operates at low duty-cycles and low sampling rates so that power may be conserved as much as possible. The *Mica* Weather Board Revision 1.0 includes temperature, photoresistor (light), barometric pressure, humidity, and passive infrared (occupancy) sensors.

The barometric pressure module is a digital sensor manufactured by Intersema [25]. The sensor is sensitive to 0.1 mbar of pressure and has an absolute pressure range from 300 to 1100 mbar. The module is calibrated during manufacturing and the calibration coefficients are stored in EEPROM persistent storage. The pressure module includes a calibrated temperature sensor to compensate raw barometric pressure readings. This calibrated module accelerates time to deployment since it is calibrated prior to delivery and operates using a standard digital bus (SPI).

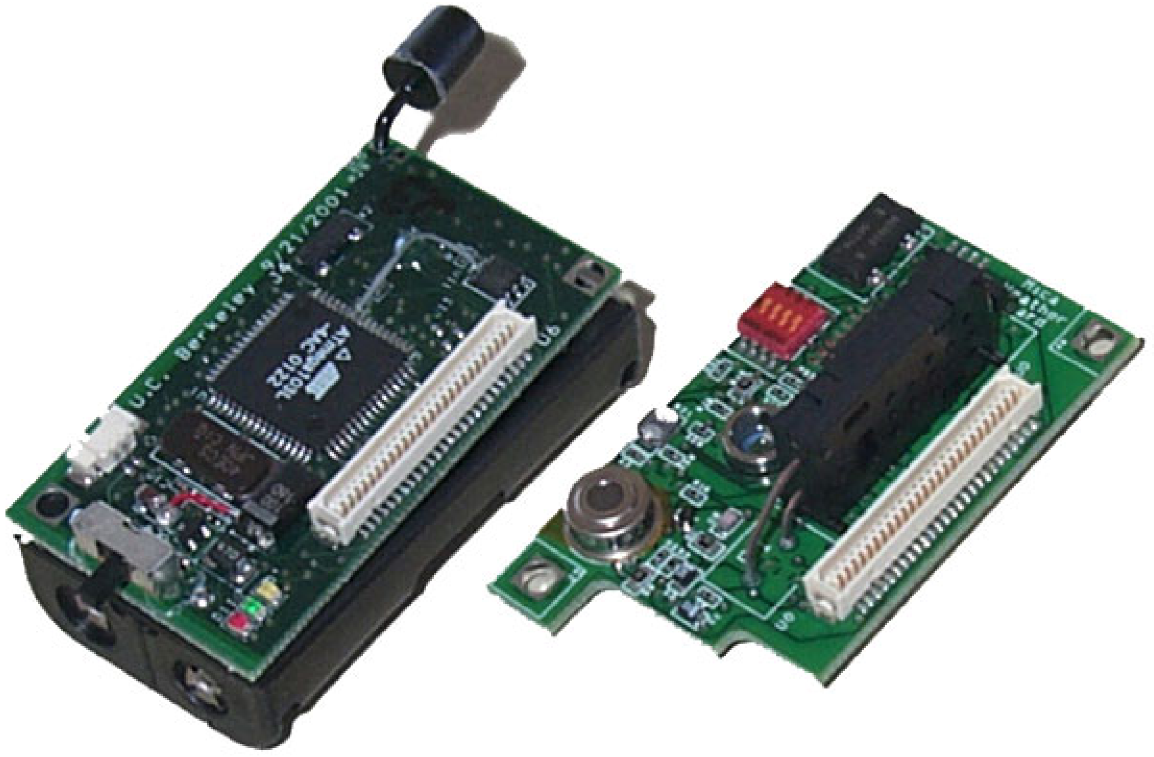


Figure 3: The *Mica* mote with the *Mica* Weather Board Revision 1.0

The humidity sensor is manufactured by General Eastern [18]. It is an analog polymer capacitive sensor that is factory calibrated to within 1 picofarad ($\pm 3\%$ relative humidity). The sensing element consists of an electrode metalization deposited over the humidity sensitive polymer. The sensor is modulated by a 555 CMOS timer to sense the charge in the capacitor, which is filtered through by RC circuit. The resulting voltage is amplified by an instrumentation amplifier for greater sensitivity over the range of 0% to 100% relative humidity. The humidity sensor may be used to monitor changes in burrow conditions and analyze conditions between populated and unpopulated burrows. Many other humidity sensors were inappropriate for use on the *Mica* Weather Board. These sensor either required high excitation voltages (5-18V) or relied on a large number of components (too many to fit on a single sensor board) to operate accurately.

The thermopile is a passive infrared sensor manufactured by Melexis [35]. Infrared radi-

Sensor	Accuracy	Change-ability	Max Rate (Hz)	Startup Time (ms)	Current (mA)
Photoresistor [10]	N/A	10%	2000	10	1.235
I^2C Temperature [12]	1 K	0.20 K	2	500	0.150
Barometric Pressure [25]	1.5 mbar	0.5%	28	35	0.010
Barometric Pressure Temp [25]	0.8 K	0.24 K	28	35	0.010
Humidity [18]	2%	3%	500	500-30000	0.775
Thermopile [35]	3 K	5%	2000	200	0.170
Thermistor [35]	5 K	10%	2000	10	0.126

Table 2: *Mica* Weather Board (Revision 1): Characteristics of each sensor.

ation produced by objects in the sensor's field of view causes a temperature difference between the thermopile's cold junction and the thermopile membrane. The temperature difference is converted to an electric potential by the thermo-electric effect in the thermopile junctions. The sensor does not require any supply voltage. The thermopile includes a thermistor in the silicon mass. The thermistor may be used to measure the temperature of the cold junction on the thermopile and accurately calculate the temperature of the black body. Passive infrared sensors may be used to detect burrow occupancy and object temperature. These readings may be filtered to report the entrance/exit times and duration of burrow occupancy.

The photoresistor is a variable resistor in a simple voltage divider circuit [10]. The divided voltage is measured by the ADC. The final temperature sensor is a 12-bit digital calibrated sensor that communicates over the I^2C bus [12]. The temperature sensor is small and is primarily used to verify readings from the temperature sensor on the barometric pressure module. The characteristics of each sensor can be seen in Table 2.

The sensors were chosen with great care to ensure high interchangeability (also known as changeability) and high accuracy. Each sensor has less than 3% variation when interchanged with others of the same model. The accuracy of each sensor is within 3% of the actual value. Through calibration, the interchangeability and accuracy can be reduced to below 1% depending on the requirements of the application. Out of the box, the nodes will be accurate for most applications. Interchangeability and accuracy allow the sensors to be deployed in the field more quickly since little or no calibration is needed prior to deployment. Accurate readings are essential for building biologists' trust in wireless sensor networks.

Use of the sensor board in low duty-cycle applications has shown that the start up time of each sensor dominates power consumption. Many sensor manufacturers assume that the sensor will be turned on once and powered indefinitely. They optimize their sensors to perform efficiently at high sample rates. Low power applications require that the sensor turns on and off quickly. The start up time is the time a sensor must be powered before its reading stabilizes. Since changes in the environment occur on the order of minutes, the sensors have a low duty-cycle and sampling rate. Typically sensors will only be powered every fifteen minutes for a few milliseconds. Sensors with long start up times (some require seconds before the readings stabilize) require current for a longer period of time, resulting in higher power consumption. Minimizing start up time yields more power per day to perform other tasks, such as routing and communication. Powering a sensor indefinitely, as the manufacturers' suggest, is not appropriate for long-lasting wireless sensor networks on a limited power budget. This property made the search for low duty-cycle sensors difficult, and resulted in two iterations of the *Mica* Weather Board to find the appropriate modules. Start up times for each sensor are listed in Table 2.

In addition to the sensors on the *Mica* Weather Board Revision 1.0, we included an I^2C analog to digital converter. Separating the ADC from the main *Mica* processing board provides greater flexibility in developing components to reduce power consumption. The ADC uses less power than the Atmel processor on the *Mica*, may be used in parallel with processing or radio transmission on the *Mica*, and can be operated in various low-power and sleep modes. Additionally, the sensor board includes an I^2C 8×8 power switch permitting individual components on the board to be turned on or off. Each switch can be operated independently of each other—further reducing power consumption.

Both the *Mica* and *Mica* Weather Board were designed with interoperability in mind. The *Mica* includes a 51-pin expansion connector. The connector has the ability to stack sensor boards on top of each other. Instead of allowing each board to compete for pins on the connector, we developed an access protocol. The protocol maintains a software queue of which boards are requesting use of the *Mica*'s input/output pins. Using an asynchronous interface, each board notifies the protocol of its completion and the next board is granted access. This method enables exclusive access to hardware input/output lines for each board. The protocol on the *Mica* will change the value of a switch on the sensor board using the I^2C bus. Changing the value of the switch triggers the sensor board to connect its sensors exclusively to the *Mica*'s resources. When a board has access, it may use the power, interrupt, ADC, and EEPROM lines that are directly connected to the microprocessor and components on the *Mica* processing board.

4.3 Sensor Nodes

To withstand the variable weather conditions on GDI, we designed protective packaging that minimally obstructs sensing functionality. *Mica* motes by their design are fairly robust mechanically, with the battery case firmly integrated with the main processing and sensor boards, and mounting holes for securing the sensor boards. To provide weather-proofing, we coated the entire sensor package with a 10-micron parylene sealant, which protects exposed electrical contacts from water. The sensors remain exposed to preserve their sensitivity. Each coated node is then enclosed in a transparent acrylic enclosure. The enclosure is ventilated to not distort the sensor readings; its primary function is to provide additional protection against mechanical failures and to raise the sensor off the ground. Acrylic packaging was chosen because it is transparent to infrared and radio frequency bands, which won't obstruct sensor readings or wireless communication. The package contains holes to enable sampling of the environment rather than conditions inside the enclosure.

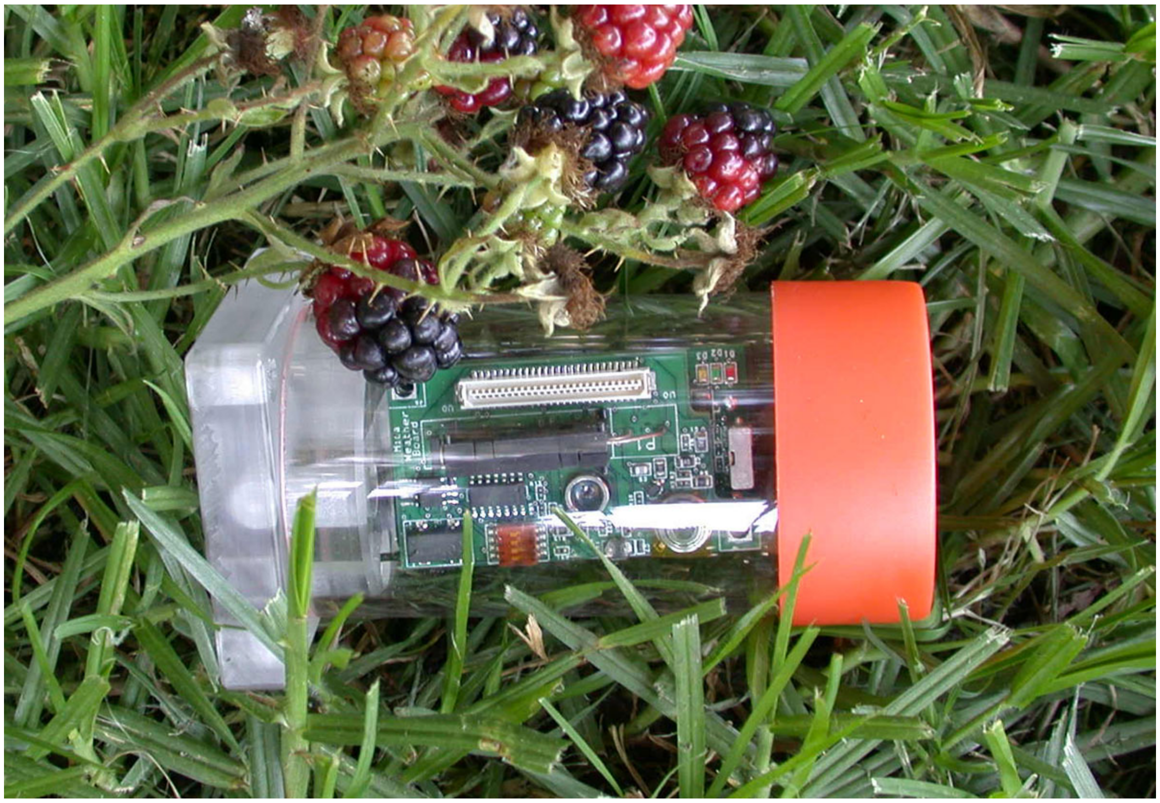


Figure 4: Acrylic enclosure used for deploying the *Mica* mote.

The acrylic enclosure shown in Figure 4 is used for deploying nodes above the ground on Great Duck Island. The size of the *Mica* mote itself was almost too large to fit in petrel burrows; therefore we placed the parylene sealed motes into the burrows without enclosures. Not using the enclosure is less robust; we've noticed expansion and contraction of connectors over the course of four weeks leading to faulty electrical connections.

4.4 Patch Gateways

Different choices of gateway nodes directly affects the implementation of the underlying transit network. We implemented two designs in parallel: an 802.11b single hop network with an embedded Linux system and a single hop mote-to-mote network. In comparison to the sensor patch which stretches only tens of meters, our solution for the transit network implementation must be able to stretch up to thousands of meters.

Initially, we chose CerfCube [26], a small, StrongArm-based embedded system, to act as the sensor patch gateway. Researchers from Intel Research and JPL have demonstrated delay-tolerant networking using CerfCubes and motes [16] which fits very well with the overall system architecture. Each gateway is equipped with a CompactFlash 802.11b adapter. Porting functionality to CerfCubes is fairly easy; they run an embedded version of the Linux operating system. Permanent storage is plentiful–the gateway can use the IBM MicroDrive which provides up to 1 GB of storage. Supplying adequate power for this device is a challenge, without power management features this device consumes about 2.5W (two orders of magnitude more than the sensor nodes). To satisfy the CerfCube power requirements, we connected it to a rechargeable battery with capacity between 50 and 100 Watt-hours (*e.g.,* sealed lead-acid). We deployed a CerfCube with a 12dbi omni-directional 2.4GHz antenna that provided a range of approximately 1000 feet

The mote-to-mote solution consisted of a mote connected to the base station and a mote in the sensor patch. Both motes were connected to a 14dbi directional 916MHz Yagi antenna. The range of the Yagi antennae is more than 1200 feet. The differences between the mote and the

CerfCube include not only a different communication frequency and power requirements, but also software components. Of particular interest to network connectivity is the MAC layer–the mote's MAC does not require a bidirectional link like 802.11b. Additionally, the mote sends raw data with a small packet header (four bytes) directly over the radio as opposed to overheads imposed by 802.11b and TCP/IP connections.

We experimented with a naïve solution that operated at 100% power duty-cycle in order to compare the reception rates and power consumption of the CerfCube compared to the mote. A 100% duty-cycle was chosen to give each solution maximum opportunity to receive and relay all sensor readings through the transit network. We discovered that both systems provided nearly identical packet reception rates, yet the CerfCube consumed two orders of magnitude more power. The packet reception rate is limited by the available bandwidth from the patch network; we can therefore assume that packets could not possibly arrive at a rate higher than the mote's radio can transmit. Despite the CerfCube's ability to forward packets reliably, we experienced many crashes caused by embedded Linux. In the same two-week time period, the mote did not need to be rebooted or managed.

Both devices were operated from a 6-volt rechargeable lead-acid battery. The CerfCube maintained a high packet reception rate through the 802.11b protocol. A TCP/IP connection was established with the base station and readings that arrived at the gateway were reliably sent over the WLAN. The combination of directional antennae for the mote resulted in reliable transmission over the single 1000 foot hop even though the mote's protocol does not feature reliable transfer like TCP/IP. We found the directional amplification of the mote's radio transmissions was adequate for delivering readings through the transit network; furthermore, readings did not collide with the sensor patch deployed in the opposite direction of the antenna.

At a remote locale, the obvious method to maintain a 100% duty-cycle for both CerfCube and mote implementations is to use solar power. A major concern about solar panels is their size and impact on the environment and habitat. As with the sensors in the sensor patch, our goal is to minimize the size of the solar panel thereby minimizing the affect on the habitat. A typical solar panel has an energy density of 0.065W/in^2. In order to determine the area of the solar panel for the gateway node, we must know the total energy consumption per day and the expected number of direct sunlight hours in the winter, the worst case scenario. The following equation is derived directly from these parameters:

$$\text{Panel Size in}^2 = \frac{\text{Total Watt Hours per Day}}{\text{Peak Winter Hours}} \times \frac{1}{0.065\text{W/in}^2}$$

The CerfCube with 802.11b requires 60Wh per day. Using the above equation, it would require a solar panel approximately 924 in^2, or 2.5×2.5 feet. Conversely, a mote gateway requires 2Wh per day for continuous operation and requires a panel only 6×6 inches in size. Before we left GDI, we decided to only use the mote solution for the gateway due to its power efficiency and small size.

4.5 Base-station installation

In order to provide remote access to the habitat monitoring network, the collection of sensor network patches is connected to the Internet through the transit network. The base station connects the transit network to the wide-area network. On Great Duck Island, we implemented the base station's Internet connection through a two-way satellite connection provided by Hughes and similar to a DirecTV system. The satellite system is connected to a laptop, which coordinates the sensor patches and provides a relational database service. We had to solve a number of challenges to turn a consumer-grade, web-oriented service into a highly reliable general-purpose network connection. The base station needs to function as a turn-key system, since it needs to run unattended. During that time we expect unscheduled system reboots. At this point we have resolved many of the engineering issues by installing additional monitoring software and running two laptops in parallel.

4.6 Database Management System

The base station currently uses a Postgres SQL database. The database stores time-stamped readings from the sensors, health status of individual sensors (*e.g.*, battery status) and the network as a whole (*e.g.*, connectivity and routing information) as well as metadata (*e.g.*, sensor locations). All of this information is specified by the database schema, which adds the ability to record both raw and compensated/calibrated sensor readings in addition to the attributes above. The particular schema used for GDI is shown in Appendix B. The GDI database is replicated every fifteen minutes over the wide-area satellite link to our Postgres database in Berkeley. Our replication implementation was simple; it dumped the data from the past fifteen minutes into a local file. The file is securely transferred over the satellite link to a server in Berkeley where it is inserted into the replicated database. Every fifteen minutes, `pg_dump` extracts the new readings to a file. During disconnections, the database update would not be successfully transferred yet the file would be overwritten fifteen minutes later when the next replication `cron` job is executed. Later in the deployment we initiated a query on the replicated database to determine which samples were present. The differences between databases was extracted to a unique file, then securely transferred to the replicated database.

Chapter 5

Sensor Network Dynamics

All of the components in the system must operate in accordance with the system's power budget. In a running system, the energy budget must be divided amongst several system services: sensor sampling, data collection, routing and communication, health monitoring and network retasking. This chapter discusses the power budget available to each mote and evaluates core services and their operation in the context of low duty-cycle operation. The power budget is essential to meeting the lifetime requirements. After an analysis of a mote's available energy, we discuss methods to further reduce power consumption. Data sampling and collection can take the form of aggregation and compression to reduce the number of packets transmitted. Low power communication enables multihop topologies to extend the size of the sensor patch. Health monitoring enables online analysis of a mote's behavior and retasking allows the mote to be reconfigured to better meet the power budget and sampling requirements of the researchers.

5.1 Energy budget

Our habitat monitoring application needs to run for seven to nine months–the length of a single field season. *Mica* runs on a pair of AA batteries, with a typical capacity of 2.5 ampere-hours (Ah). However we can neither use every drop of energy in the batteries, nor are the batteries

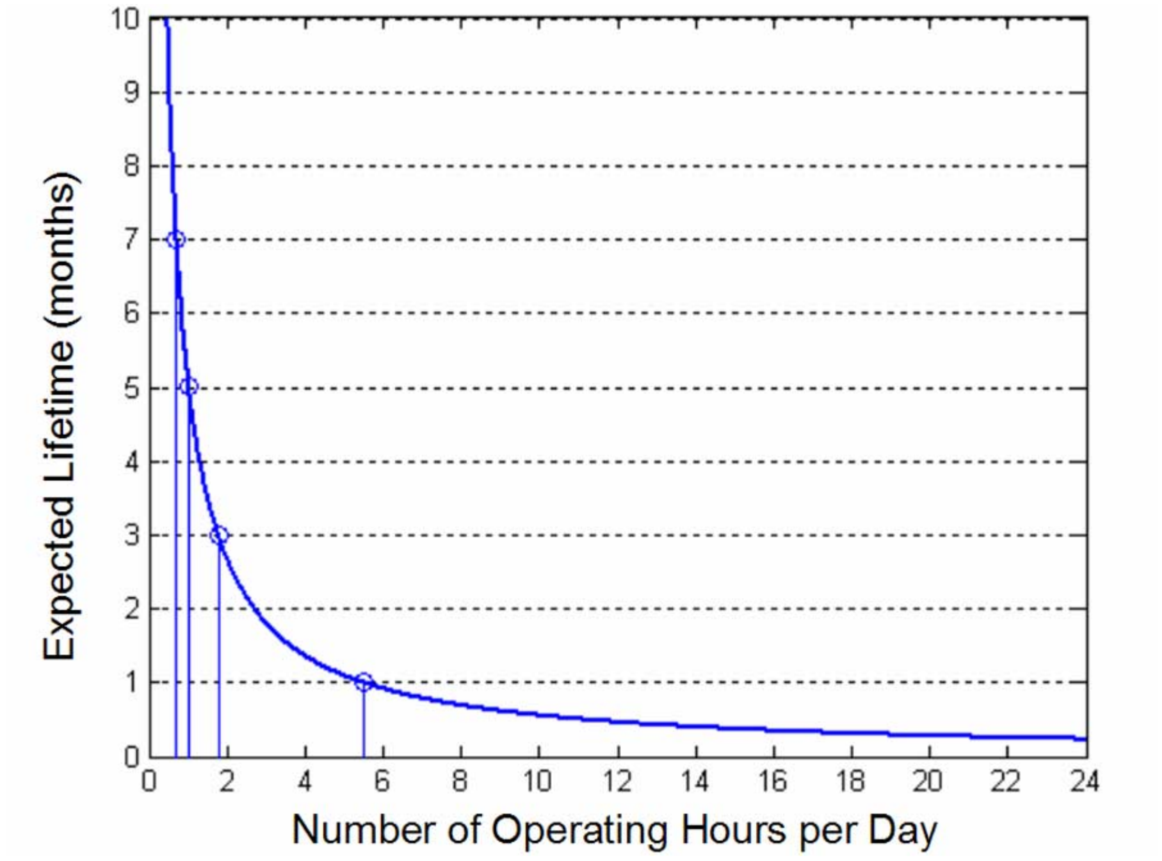


Figure 5: The expected lifetime of a sensor node as a function of duty-cycle

manufactured with identical capacities from batch to batch or from manufacturer to manufacturer. We make a conservative estimate that the batteries will be able to supply 2200 mAh at 3 volts. Since the majority of the motes are located underground, renewable energy sources, like solar power, are not available.

Assuming the system will operate uniformly over the deployment period, each mote has 8.148 mAh per day available for use in a nine month deployment. The application chooses how to allocate this energy budget between sleep modes, sensing, local calculations and communications. We note that since different motes in the network have different functions, they also may have very different power requirements. For example, motes near the gateway may need to forward all messages from a patch, whereas a mote in a nest may need to merely report its own readings. In any

Operation	**nAh**
Transmitting a packet	20.000
Receiving a packet	8.000
Radio listening for 1 millisecond	1.250
Operating sensor for 1 sample (analog)	1.080
Operating sensor for 1 sample (digital)	0.347
Reading a sample from the ADC	0.011
EEPROM Read Data	1.111
EEPROM Program/Erase Data	83.333

Table 3: Power required by various *Mica* operations.

network, there will be some set of power limited motes; when these motes exhaust their supplies, the network is disconnected and inoperable. Consequently, we need to budget our power with respect to the energy bottleneck of the network. To form an estimate of what is possible on a *Mica* mote with a pair of AA batteries, we tabulated the costs of various basic operations in Table 3.

The baseline lifetime of the mote is determined by the current draw in the sleep state. By minimizing sleep current to approximately 30μA discussed in Section 4.1, the energy available for

Operation	**Operating Time/Day**	**Duty-Cycle**	**Sample Rate**
Always Sleep	24 hours	0%	0 samples/day
+ μCPU on	52 minutes	3.61%	0 samples/day
+ Radio On (Listen)	28 minutes	1.94%	0 samples/day
+ Sample All Sensors	21 minutes	1.45%	630 samples/day
+ Transmit Samples	20 minutes	1.38%	600 samples/day

Table 4: The effect of each sensor node operation on duty-cycle and sample rate.

tasks is reduced to 6.9 mAh per day. The expected lifetime of a mote as a function of how long the mote operates per day can be seen in Figure 5. The exponential curve shows that cutting the operating time by a small amount when the duty-cycle is low results in a huge gain in expected lifetime. A mote will "live" the longest if it is always asleep. By optimizing the sleep current consumption, the expected lifetime exponential function shown in Figure 5 has a lower slope near the origin. Lowering the sleep current consumption yields more operating hours per day while the expected lifetime remains constant.

As each operation in Table 3 is added to the application running on a *Mica* mote, it directly affects the available energy in the power budget for sampling. If the mote is always asleep, it can operate 24 hours per day for 2081 days or 5.7 years. On the other hand, a mote running at a 100% power duty-cycle will only operate for 7 days or 0.02 years. We must carefully adhere to the budget of 6.9 mAh per day for operations. Table 4 shows the effect of adding each operation into the application on the duty-cycle and maximum sampling rate of the sensor node. By adding each operation, we lower the available power per day. In the Great Duck Island deployment, we chose to use a static energy budget based on the calculations and values shown in Table 4. The mote was programmed at deployment time with a sampling rate (one sample every 64 seconds) that meets the power budget according to the required sample rate and network lifetime.

5.2 Data sampling and collection

In habitat monitoring the ultimate goal is data collection; sampling rates and precision of measurements are often dictated by external specifications. For every sensor we can bound the cost of taking a single sample. By analyzing the requirements we can place a bound on the energy spent on data acquisition. We trade the cost of data processing and compression against the cost of data transmission. Results can be transmitted as raw data, transmitted data, filtered data, and in-network aggregated data.

To compare the difference between transmitting the raw data and the cost of compression,

Compression algorithm	Huffman (pack)	Lempel-Ziv (gzip)	Burrow-Wheeler (bzip2)	Uncompressed
8-bit sample	1128	611	681	1365
10-bit sample	1827	1404	1480	1707
16-bit sample	2074	1263	1193	2730
8-bit difference	347	324	298	1365
10-bit difference	936	911	848	1707
16-bit difference	839	755	769	2730

Table 5: Compression characteristics of typical indoor light signal.

we can analyze a dataset collected from indoor monitoring networks. We must decide whether it is worthwhile to log data and compress it, or is it more efficient to push each sample into the network. Let us consider an experiment where a mote collects a light sample every minute. The sample is represented as a 16-bit integer, but it contains a 10-bit ADC reading. Assuming that each packet can carry 25 bytes of payload, unprocessed data requires between 72 (for 10-bit samples) and 116 packets (for 16-bit samples). Although this service does not put a burden on the leaf nodes, the routing nodes near the root may need to retransmit the messages from every leaf in the network, roughly two orders of magnitude more. Anecdotal evidence presented in Table 5 suggests that this volume of data can be easily reduced by a factor of 2-4 by applying a delta compression and a standard compression algorithm (*e.g.,* Huffman coding or Lempel-Ziv). The compression performs even better when applied to a longer run of data. Better results can be obtained through lossy compression techniques and filtering–often times a life scientist is interested in duration of occupancy or average reading over a time window. In this case, the data is filtered locally and the results are transmitted over the network in small infrequent updates. Unfortunately the cost of writing a data sample to stable storage is extremely costly (83nAh per 4 bytes). In comparison, an entire 30-byte payload can be sent over the radio for 20nAh. On GDI, we chose to send uncompressed samples into the network since the power consumption is much lower.

Other methods include distributed compression involving correlating network data among similar nodes, using Coset codes [30], and using in-network aggregation. TinyDB [33], a streaming data service for sensor networks, features an in-network aggregator for sensor data delivery [32]. TinyDB treats the network as a single entity where a query is applied to that entity. The query does not specify how the data gets back from the sensors, rather it operates on the network to aggregate and efficiently deliver data. TinyDB attaches to the underlying power management and routing interfaces to transmit queries. It is a very large, complex application that provides users with a simple declarative language to retrieve specific sensor data. We intend to integrate TinyDB into future habitat monitoring applications, however for the first deployment, TinyDB's complexity is too great for conclusive analysis of basic operations, such as the effectiveness of various power management and routing algorithms.

Once we have allocated the energy for sampling the sensors and communicating the results, the remaining energy is devoted to maintaining the network – MAC protocols, health and status, routing tables, and forwarding network messages. These tasks can either be tightly scheduled or run on demand. On one extreme, the system is scheduled at every level, from TDMA access to the channel, through scheduled adaptation of routes and channel quality. Overhead costs are upfront and fixed. A TDMA system is expected to perform well if the network is relatively static. On the other extreme, we use a low-power hailing channel to create on-demand synchronization between a sender and a receiver. The service overhead is proportional to the use of the service. This approach can be more robust to unexpected changes in the network, at the expense of extra cost. Finally, a hybrid approach is possible, where each service runs in an on-demand fashion, but the time period for when the demand can occur is scheduled on a coarse basis.

Our deployment at GDI is sending raw data values that are logged at the basestation. As the biologists at the College of the Atlantic analyze the data, we intend to change the data sampling and collection according to their needs via network retasking (see Section 5.4).

5.3 Communications

Power-efficient communication paradigms for habitat monitoring must include a set of routing algorithms, media access algorithms, and managed hardware access. The routing algorithms must be tailored for efficient network communication, while maintaining connectivity when required to source or relay packets.

A simple routing solution for low duty-cycle sensor networks is simply transmitting data to a gateway during scheduled communication periods. Data is only communicated in one direction and there is no dependency on surrounding motes for relaying packets in a multihop manner. The routing deployed on GDI is a hierarchical model. The sensor nodes in burrows are transmit only with a low duty-cycle; they sample about once per minute. The gateway mote is fully powered by solar power, so it is always on and relaying packets to the base station. We intend to equip future deployments with energy harvesting capabilities to allow motes above ground to perform additional routing tasks with higher duty-cycles.

Many of the hard to reach biological research locations are beyond the range of a single wireless broadcast from mote to gateway. Accordingly, we intend to use a low-power multi-hop protocol to collect, aggregate, and communicate data.

Methods like GAF [48] and SPAN [9] have been used to extend the longevity of the network by selecting representatives to participate in the network; thereby these algorithms reduce the average per node power consumption. Although these methods provide factors of 2 to 3 times longer network operation, our application requires a factor of 100 times longer network operation– recall that our sensor nodes are on for at most 1.4 hours per day. GAF and SPAN do not account for infrequent sampling but rather continuous network connectivity and operation. Instead, we propose augmenting scheduled multihop routing or low power MAC protocols with GAF and/or SPAN to provide additional power savings. GAF and SPAN are independent of communication frequency, whereas our application requires increased power savings that may be achieved by adjusting the communication frequency.

Duty-Cycle	**Preamble Length**	**Radio Sampling**	**Max Packets**	**Throughput**
100%	18 bytes	N/A	42.93 packets/sec	12.364kbps
35.5%	84 bytes	20 ms	19.69 packets/sec	5.671kbps
11.5%	240 bytes	85 ms	8.64 packets/sec	2.488kbps
7.53%	361 bytes	135 ms	6.03 packets/sec	1.737kbps
5.61%	480 bytes	185 ms	4.64 packets/sec	1.336kbps
2.22%	1202 bytes	485 ms	1.94 packets/sec	0.559kbps
1.00%	2644 bytes	1085 ms	0.89 packets/sec	0.258kbps

Table 6: Effective channel utilization using low power listening on the *Mica* 2 mote.

Alternatively, we have experimented with using low power MAC protocols. By determining our duty-cycle, we can calculate the frequency with which the radio samples for a start symbol. By extending the start symbol when transmitting packets, we can match the length of the start symbol to the sampling frequency. We have implemented low power listening on the *Mica* 2 hardware platform. The Chipcon CC1000 radio on the *Mica* 2 requires 3 ms for the CC1000 radio to begin sampling, and then the radio is sampled for 8 ms. The effect of duty-cycling the radio can be seen in Table 6. For example, to achieve a 1% radio duty-cycle, transmitted packets require a 2644 byte preamble and limit the radio to an effective throughput of 258 bits per second of data being transmitted. Other low power MAC protocols, such as S-MAC [49] and Aloha with preamble sampling [13] employ similar techniques that turn off the radio during idle periods to reduce power consumption. The difference between scheduled communication and low power MACs is instead of having a large power and network overhead to set up a schedule, the overhead is distributed along the lifetime of the node. The decision for which to use depends on the end-user interactivity required by the application. A potential tradeoff of using a low power MAC is that transmitted packets potentially wake up every node within the cell. Other nodes periodically sampling for a preamble may pick up unneeded packets. Although early rejection can be applied, scheduling prevents unneeded nodes from wasting power processing a packet's headers.

5.4 Network Retasking

As the life scientists refine the experiment, it may be necessary to adjust the functionality of individual motes. Scientists may be interested in changing the sampling rates, duty-cycle, and filters running on each mote. As initial data is analyzed, life scientists may be interested in monitoring certain sensors more closely. For example, after examining raw thermopile occupancy data, the mote is retasked to report only the entrance and exit of an animal. By performing this particular refinement, the mote will consume less power and its lifetime will be extended.

Retasking can take several different forms. Scalar parameters, like duty-cycle or sampling rates, may be adjusted through the application manager. Even such simple adjustment allows researchers to focus their efforts in more interesting areas. Most of the time such updates can be encapsulated in network maintenance packets. More complex functionality adjustment may be implemented through virtual machines like Maté [31]. Virtual machine-based retasking seems ideal when the much of the underlying functionality is implemented through native functions, as is the case in making routing decisions, or processing data through a predefined set of filters. Virtual machine programs can be fairly small (many fit in a single packet). Instead of using virtual machines, the TinyDB query model may be used to retask the sampling modalities [33]. Life scientists may inject a new query into the network to change the functionality of the sensor motes. Simic and Sastry [41] have designed a distributed algorithm for estimating the gradient of a scalar field using a wireless sensor network. They bound the error of the distributed calculated gradient and show convergence of the algorithm. As new distributed algorithms for filtering and aggregating data are developed, we would like the ability to retask nodes with that functionality.

In the most drastic situation, the entire code image running on a mote may be replaced with a new one. One would use this method when an extreme retasking of the application is necessary; for example if it were necessary to install a new signal-specific compression algorithm to cope with the volume of data or a reorganization of how data is stored in a mote's flash memory. The reprogramming process is quite costly – it involves reliably transmitting the binary image of the code (approximately 10kB) to all motes that need to be reprogrammed, and invoking a reprogramming application which runs the motes for 2 minutes while drawing about 10 mA. To relate this to the energy budget: we can afford to reprogram the motes every day during the 9 month life cycle if reprogramming is the mote's only task. Although significantly more expensive in absolute terms than virtual machine reprogramming, it can pay off over the period of a few days since native code executes more efficiently.

5.5 Health and Status Monitoring

In order to verify that the mote is maintaining its power budget and local connectivity, a health monitoring component runs on each mote. Health monitoring is essential for a variety of purposes; *e.g.*, the duty-cycle of a mote may be dynamically adjusted to alter its lifetime. Monitoring the health of a mote provides information about how well the mote is meeting the requirements set forth in Section 2.2.

A simple monitoring implementation is deployed on GDI. Each mote periodically includes its battery voltage level with the sensor readings it transmits. The voltage is represented as a one byte discrete value in the range of 0 to 3.3V. Adding voltage measurements has greatly assisted us in remote analysis of mote failures, while the packet size has only minimally increased (see Appendix A for the TinyOS packet format).

Heath readings can assist computer science researchers when analyzing the algorithms running on a mote. These messages may provide power readings, information about the mote's neighbors, and profile the mote's packet loss.

Health and status readings sent to the gateway can be used to infer the validity of the mote's sensor readings. A mote may monitor values of sensor readings and send status messages when unusual values exist. Although the health readings are not critical for correct application execution, their use can be seen as preventive maintenance.

Chapter 6

Results from the GDI Deployment

Thirty-two motes were deployed on Great Duck Island, of which nine were in underground burrows. The sensor network logged data from July 18, 2002 through November 18, 2002. During this period, 1.2 million readings and over 1.8 million packets were logged and replicated back to the database in Berkeley. In this chapter, we analyze the performance and accuracy of the *Mica* Weather Board. We provide an analysis of the power consumption and overall system reliability during the deployment.

6.1 *Mica* Weather Board 1.0 Analysis

Throughout the deployment, we sought to verify the sensor network readings with the behavior of the Leach's Storm Petrels and the expected environmental conditions. Occupancy (thermopile) data was verified using recorded petrel calls. Figure 6 shows occupancy data collected from July 18, 2002 to August 5, 2002. The mote was placed several feet down a burrow tunnel, approximately 1500 feet from the lightkeeper's house on Great Duck Island. The plotted values indicate the difference between ambient temperature and the object in the thermopile's field of view. Figure 6 illustrates that a petrel left the burrow on July 21st and returned on July 23rd. This data was verified by playing recorded petrel calls above the burrow and listening for the petrel's response. The petrel

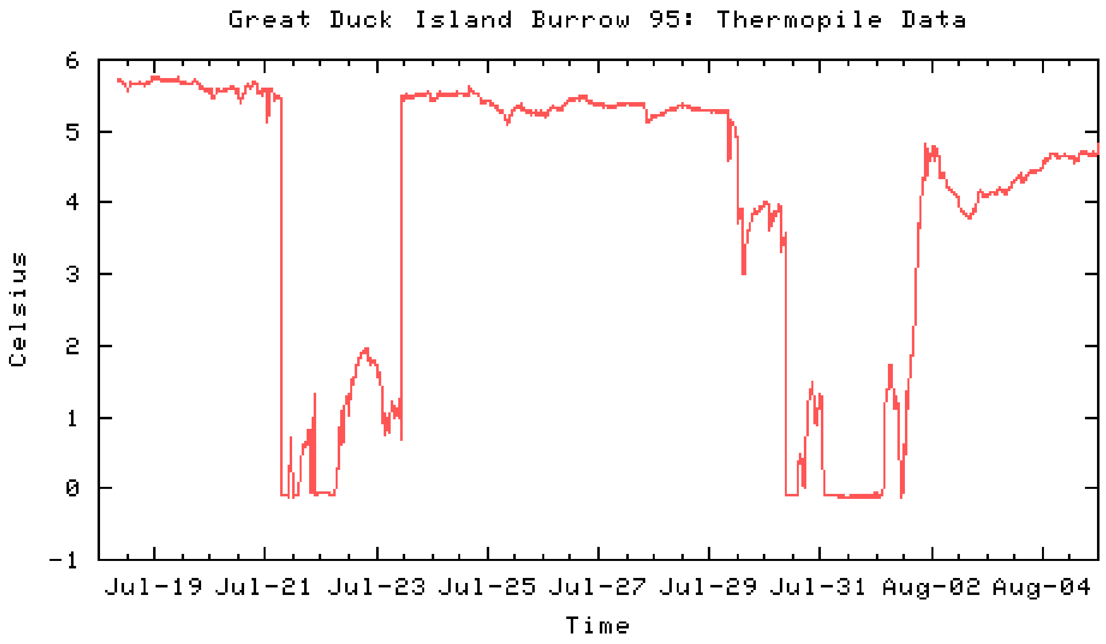


Figure 6: Raw thermopile data from Great Duck Island during a 19-day period (July 18, 2002 to August 5, 2002)

left again between July 30th and August 1st. Variations in the data during times that the petrel vacated can be attributed to the $\pm 3^{o}$C accuracy of the thermopile as well as changing environmental conditions between night and day.

Through the use of the humidity, temperature, and lights sensors, we were able to establish correlations between the sensors during day and night periods both inside and outside of petrel burrows. Figure 7 shows light, temperature, and humidity readings over a five day period for a mote in a burrow and a mote directly outside of a burrow. The burrow mote maintains constant light, temperature, and humidity levels over the five day period. In contrast, the mote outside of the burrow correlates light, humidity, and temperature data with time of day. During the day while the sun is out, the humidity drops, light increases, and temperature increases. The opposite effect occurs at night.

There were many issues concerning the data received from the first *Mica* Weather Board. Of primary concern was the capacitive humidity sensor. When the sensor was saturated with water, a

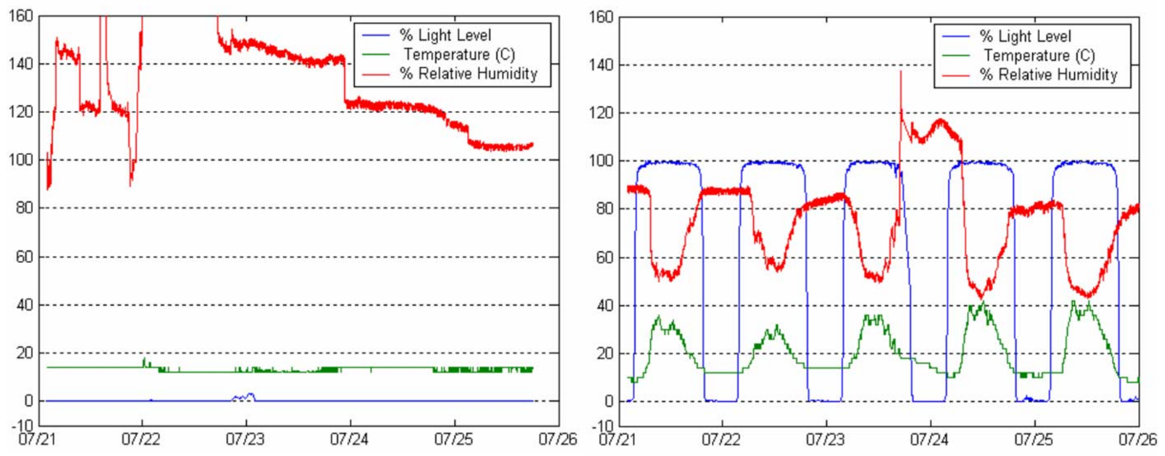


Figure 7: Light, Temperature and Humidity readings inside a burrow (left) and outside a burrow

short occurred between the two terminals of the capacitive element causing excessive battery drain. After the sensor dried, the readings were unusable for an extended period of time (sometimes up to 24 hours, others times for the remainder of the deployment).

A flaw in our deployment strategy was the lack of a pre-deployment calibration procedure. The sensors were not measured at a few points to determine the calibration curves for each sensor; instead we assumed the manufacturer's curves and interchangeability data was accurate. Figure 10(a) in Chapter 7 shows ambient relative humidity readings over an 800-hour deployment in our laboratory. It is obvious that the six humidity sensors shown do not exhibit identical behavior or $\pm 3\%$ interchangeability as we expected.

6.2 Power Management

A primary requirement of deploying wireless sensor networks for habitat monitoring is a system lifetime of at least one field season (7-9 months). In the initial deployment, we calculated that the nodes would operate for approximately 7 months. In the period from July 19th through September 19, 2002, the battery voltage dropped from 3.3V to 2.4V as shown in Figure 8. Typical

AA battery half-life occurs at 1.2V, or 2.4V for 2 AA batteries in series. The results of battery voltage show the batteries to be at their half life approximately 2 months into the deployment. The total lifetime of the nodes was recalculated with the initial deployment parameters to uncover any mistakes. Upon arrival on Great Duck Island, we changed the sampling rate from once every two minutes to once every 64 seconds (5.8% duty-cycle) for initial analysis of the incoming data. Before leaving GDI, we did not reset the sampling rate to once every 128 seconds (3% duty-cycle). This error partially accounts for the actual system lifetime equal to half of the expected lifetime. The other factor is the AA batteries do not operate according to typical battery curves provided by the manufacturer due to our periodic execution model. AA batteries are rated at a constant current discharge higher than the operating current consumption of the mote. The data shows that the discharge of the batteries under periodic operation (sample once every minute, then sleep) does not result in uniform energy consumption from the batteries. Additionally, as the voltage drops, the boost converter on the sensor node draws additional current to maintain the regulated 3.3V during operation.

6.3 System Analysis

The lifetime of the system must be achieved through the combination of efficient power management and unattended operation. In order for the system to run unattended, the behavior of the system must be predictable. Life scientists will only begin to trust the system through repeated results and confidence-building predictable operation.

The behavior of the system over a two-month period is shown in Figure 8. The spikes in the number of nodes in Figure 8 is due to researchers returning to Great Duck Island to revive the nodes. An initial deficiency was found with the plastic expansion connector between sensor board and the *Mica* processing platform. Humidity and temperature variations caused the connector to expand and contract thereby resulting in faulting connections. Other nodes had failed due to corroded battery terminals. Their batteries were replaced and redeployed, however the redeployed

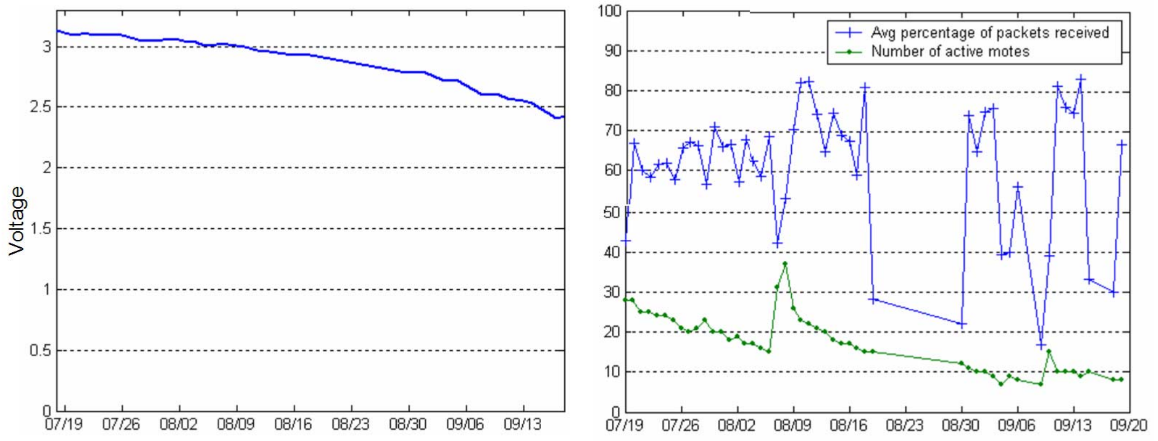


Figure 8: System performance: voltage degradation over time (left) and average number of packets compared to number of operating nodes (right)

nodes were no less susceptible to corrosion after replacement. Upon recovering nonfunctioning nodes, we noticed that petrels are not “mote-neutral”. In a few instances, the petrels had buried motes and packed dirt on top of the mote.

Later in the deployment, the variance in the average percentage of packets received increases yet the average packet loss during the deployment remains nearly constant. Since the average percent loss is nearly constant, the per node throughput over time is also constant. After two months of deployment, 30% of the initial nodes were still operating. The final node ceased reporting on November 20, 2002, over four months after the deployment began and two months after life scientists had abandoned the island for the winter season.

Chapter 7

System Evolution

Many issues arose during the initial 2002 deployment on Great Duck Island that have warranted a redesign of many components in preparation for future deployments. After analyzing the 2002 deployment in Chapter 6, we discuss the redesign of system components after learning from the successes and failures of the initial deployment.

7.1 *Mica* Weather Board Revision 1.5

The *Mica* Weather Board Revision 1.0 was deployed on Great Duck Island in the summer of 2002. Analysis and results from these sensors are described in Section 6.1. We noticed that some of the readings from the *Mica* Weather Board Revision 1.0 were out of range, whereas other motes appeared to excessively drain their batteries due to the capacitive humidity sensor. These flaws led to the design of a second revision called the *Mica* Weather Board Revision 1.5. In order to meet the needs of a greater scientific community and address concerns raised in the GDI deployment, we redesigned the *Mica* Weather Board with a new set of sensors and power management circuits. We were primarily interested in choosing familiar sensors that scientists could trust. To expedite design and deployment time, we choose to use all digital, calibrated sensors in the *Mica* Weather Board Revision 1.5 (shown in Figure 9). The schematic is available in Appendix D.

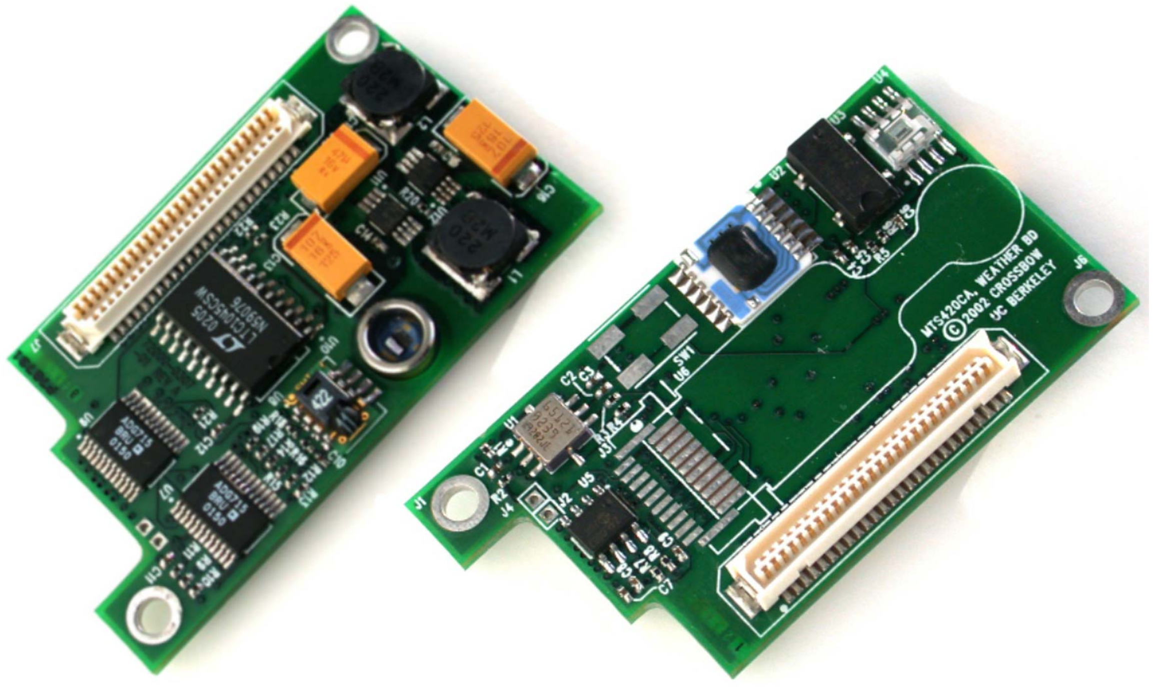


Figure 9: The *Mica* Weather Board Revision 1.5

Instead of a generic photoresistor, we choose to use a digital I^{2}C module from TAOS that provides readings in Lux units [43]. The sensor features two photodiodes sensitive over two different wavelengths. The intensity from each diode is read, and an interpolation function calculates the total Lux value. The sensor is sensitive over the range of total solar radiation (200-1100nm), the light range visible to humans and animals. Accurate Lux readings, in conjunction with occupancy readings, may help life scientists determine what light levels are acceptable for petrels to emerge from the burrow. We added a second photodiode from Hamamatsu that is sensitive to Photosynthetically Active Radiation (400-750nm) for deployments monitoring trees and plants [19].

Due to the complexity of the analog circuit design and poor functionality in the first revision, we moved to the Sensirion digital relative humidity module [40]. The sensor is small in size, calibrated, includes an integrated temperature sensor, and contains a 5^{o}C miniature heater to prevent condensation buildup on the sensing element. For the same reasons, we moved from the

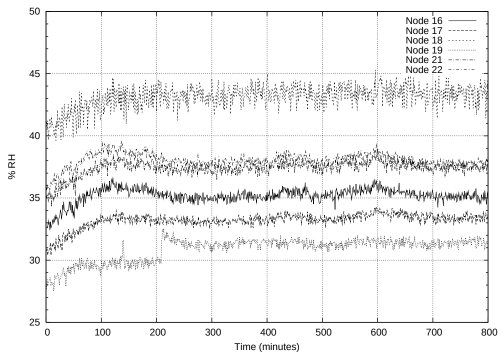
Sensor	Accuracy	Change-ability	Max Rate (Hz)	Startup Time (ms)	Current (mA)
Total Solar [43]	35%	5%	2	800	0.350
Photosynthetically Active [19]	0.3 A/W	5%	1000	0.1	0.025
Barometric Pressure [25]	1.5 mbar	0.5%	28	35	0.010
Barometric Pressure Temp [25]	0.8 K	0.24 K	28	35	0.010
Humidity [40]	2%	0.1%	3	11	0.500
Humidity Temp [40]	0.01 K	0.1 K	3	11	0.500
Thermopile [36]	2 K	0.1 K	3000	2	5.600
Thermistor [36]	1 K	0.1 K	3000	2	5.600
Accelerometer [2]	200 μg	2%	375	17	0.600

Table 7: *Mica* Weather Board (Revision 1.5): Characteristics of each sensor.

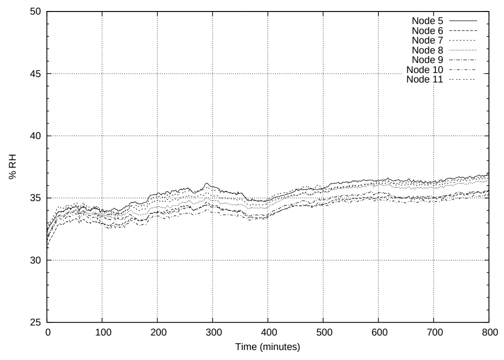
Melexis infrared detector to a Melexis infrared calibrated module [36]. The module is calibrated at two different temperature and includes a microprocessor to linearize, average, and filter the analog output of the thermopile.

The Intersema module appears on both revisions of the *Mica* Weather Board since it is calibrated, low-power, and has digital output. An accelerometer was added for environmental, seismic, and building monitoring. The accelerometer may be used to remotely determine the orientation of the sensor board. By monitoring the change in acceleration over time, the orientation may be determined by using gravity as a reference point. The sensors of the *Mica* Weather Board Revision 1.5 and their properties are shown in Table 7.

The significant differences between the two revisions are the reduction in startup time, the move to digital calibrated sensors, and an increase in sensor accuracy. When selecting components, we further reduced the size by minimizing the height such that the package may be smaller or other sensor boards may be stacked above or below the *Mica* Weather Board.



(a) Revision 1.0



(b) Revision 1.5

Figure 10: Comparison of relative humidity readings in a controlled environment for *Mica* Weather Board Revision 1.0 and Revision 1.5

On the *Mica* Weather Board Revision 1.5, each sensor contains its own ADC, optimized both in power and precision for that particular sensor. Readings are sent over the various digital data buses, maintaining the benefits of ADC separation from the *Mica* mote such that communication and sampling can be performed in parallel.

The second weather board includes two I^{2}C switches for additional power control–we found that even I/O pins pulled high caused reverse powering of some components and additional energy consumption. Testing of Revision 1.5 has shown that decoupling entire circuits from power lines and I/O lines has significantly reduced leakage current by 100μA. An onboard I^{2}C EEPROM permits identification of the board from other concurrent sensor boards. The schematic of the Revision 1.5 board can be found in Appendix D.

We have tested prototype Revision 1.5 boards to ensure that the sensors function as per specification. Initial data has shown that the new digital calibrated sensors have significant advantages over analog designs. We programmed six Revision 1.0 boards and seven Revision 1.5 boards with an application to read the value of the humidity sensor and transmit the sample wirelessly every two seconds. The boards were placed next to each other monitoring the environment in our lab.

There was no additional calibration performed on the boards other than the calibration performed by the sensor manufacturer. The application ran over a six day period and the results are shown in Figure 10. Each data point is the average of a minute consisting of 30 samples. The noise in the Revision 1.0 weather board was almost 5% with a 15% interchangeability error between boards. Traditional post-manufacturing calibration may only partially reduce the noise and interchangeability; the readings indicate there are uncorrelated changes in readings between nodes. The noise in the Revision 1.5 board was under 1%, and averaged 0.5%. The maximum interchangeability error was 2%, which is lower than the manufacturer's published error of 3.5%. Interestingly, the interchangeability error in the Revision 1.5 board is lower than the noise in the Revision 1.0 board. Note that the variance between individual samples shown in Figure 10 is lower than the actual variation due to smoothing caused by averaging samples. The power consumption of the *Mica* Weather Board Revision 1.5 was half that of Revision 1.0 during the test period due to two factors: (1) the Revision 1.5 sensor was turned off between samples due to low start up time and (2) the power-on current consumption of the Revision 1.5 sensor is two-thirds that of the Revision 1.0 sensor.

7.2 Power Management

To address the problems with AA alkaline batteries that was discussed in Section 6.2, our next deployment features a node without a 3.3V boost converter and will be powered by a lithium-sulfur dioxide battery. The terminal voltage of lithium batteries is very stable. The battery is able to maintain a relatively constant terminal voltage until the last 15% of its life for significant discharge rates. At lower discharge rates the terminal voltage will stay almost constant until the last 5% of the battery's life [29]. This is in direct contrast to alkaline batteries where the terminal voltage starts with a rapid drop as the internal battery resistance climbs, wasting much of the remaining power as shown in Figure 8 of Chapter 6. Lithium sulfur dioxide batteries are typically used for outdoor military operations that require long battery life in a small, light package.

7.3 Form Factor and Enclosures

Upon retrieving the sensor nodes from the 2002 GDI deployment, we immediately noticed the physical abuse that the sensors endured. The petrels chose to peck and bury the motes placed in their burrows. Many were packed extremely tightly with dirt and were found a few inches below their initial deployment position. One possible reason is a red LED was turned on for a few microseconds every minute to indicate to the life scientist that the device was still operating. Since the *Mica* mote was a tight fit for petrel burrows, it is possible the mote is simply too large and disruptive.

In response to these concerns, we have moved to a circular platform that is one inch in diameter. The *MicaDot* node, shown in Table 1 of Chapter 4, is much smaller in size. Since radio transmissions on the *MicaDot* are not dependent on input voltage, there is no DC-DC boost converter to excessively drain battery power. The radio also features higher gain and tunable frequency. We have manufactured the *Mica* Weather Board Revision 1.5 in the *MicaDot* form factor.

In order to protect the node from the environment more than the parylene and acrylic enclosures, we designed sealed enclosures from HDPE (shown in Figure 11 and Figure 12). An unfortunate side effect of the acrylic enclosures being infrared transparent is that the sun's radiation causes the enclosure to heat up more than the ambient temperature. The enclosures feature exposed sensors on the top (and bottom for the above ground design). The *MicaDot* and battery are sealed inside the enclosure using an o-ring that sandwiches the top sensor board and the cylinder together to make a seal. The enclosure is opaque to the infrared band while transparent to radio communication at 433MHz and 916MHz. We expect that new sensors, redesigned hardware, and redesigned enclosures with the experience of the 2002 GDI deployment will yield less disturbance and greater robustness in future deployments.

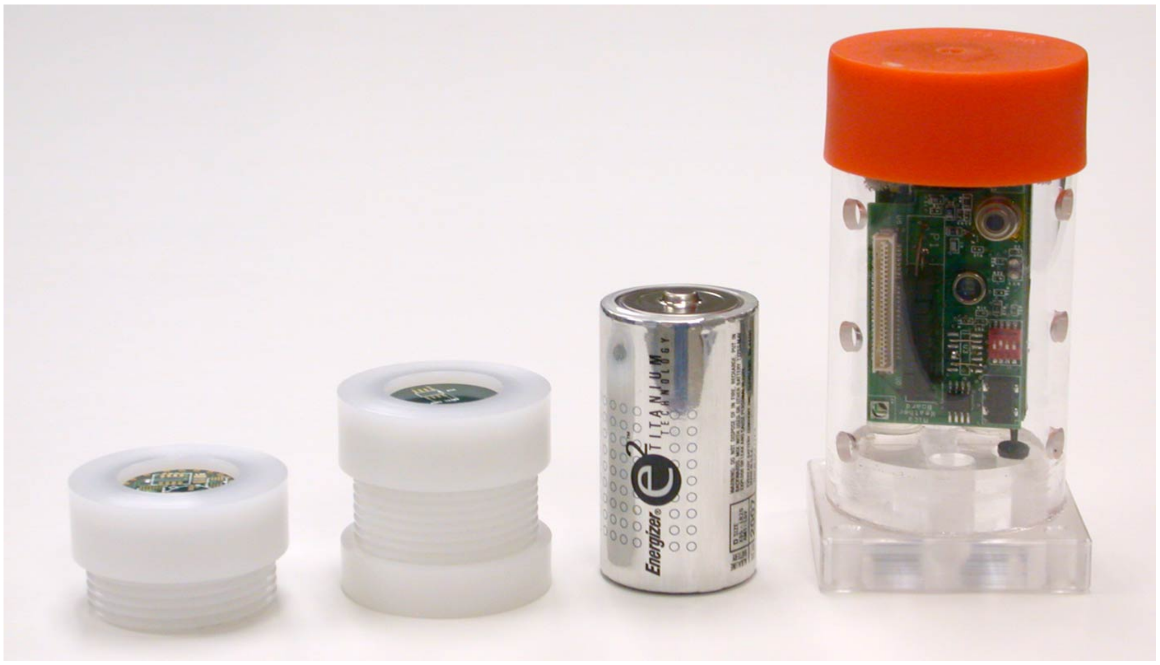


Figure 11: New enclosures for habitat monitoring deployment. From left: Burrow enclosure, above ground enclosure, D-cell battery for size comparison, and the above ground enclosure from the 2002 GDI deployment.

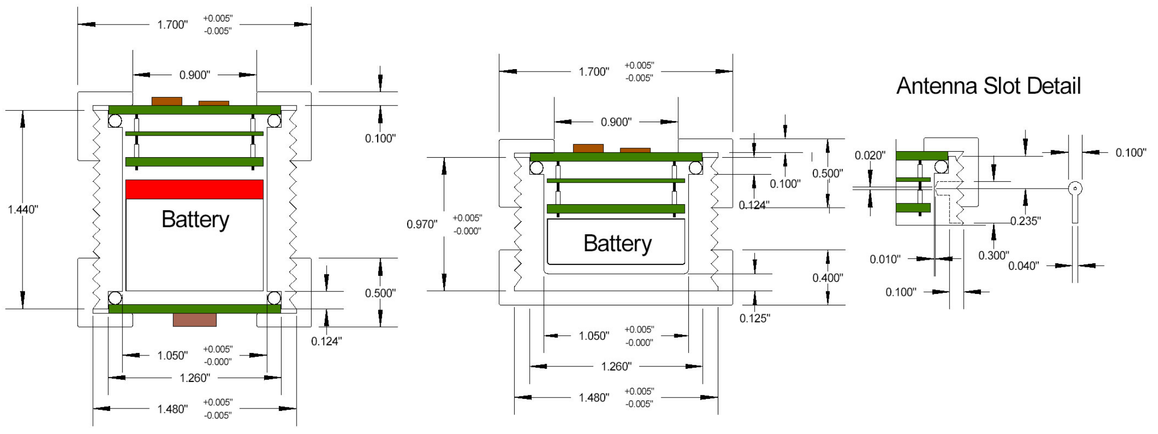


Figure 12: Design drawings of the 2003 GDI enclosures. From left: above ground enclosure, burrow enclosure, and antenna mounting design.

Chapter 8

Related Work

A discussion of the use of traditional data loggers for habitat monitoring in comparison to wireless sensor networks is provided in Chapter 2. Due to size, price, and organism disturbance, using these systems for fine-grained habitat monitoring is inappropriate.

Habitat monitoring for wireless sensor networks has been studied by a variety of other research groups. Cerpa et al. [8] propose a multi-tiered architecture for habitat monitoring. The architecture focuses primarily on wildlife tracking instead of habitat monitoring. Habitat monitoring applications may need other important services in addition to those mentioned in Chapter 5. These services include localization, time synchronization, and self configuration described in [8]. We intend to implement these services in future iterations of the application that includes more complex functionality. A PC104 hardware platform was used for the implementation with future work involving porting the software to motes. Experimentation using a hybrid PC104 and mote network has been done to analyze acoustic signals [46], but no long term deployment results or reliability data has been published.

Researchers at Northern Arizona University have built a sensor network platform called WISARD that they are using to monitor microclimate variations in a forest populated with Ponderosa pine trees [47]. West et al. designed hardware components to accomplish microclimate monitoring on a node level; however, it is assumed external sensors will be attached to the node and there is no discussion of sensor selection. Power management, retasking, and communication algorithms described are similar to those discussed here. They take a similar approach of building and evaluating a simple system with specialized hardware for computation, communication, and sampling. There is currently no analysis of the performance of their sensor nodes or the software they describe. The remainder of the system architecture (such as the flow of data) is not considered in their work.

Directly related to our deployment, ZebraNet [27] is a wireless sensor network design for monitoring and tracking wildlife. ZebraNet uses nodes significantly larger and heavier than motes. Their nodes are approximately 1151 grams compared to a 173 gram *Mica* mote and 28 gram *MicaDot*. The architecture is designed for a mobile, multi-hop wireless network. In many respects, this design does not fit with monitoring the Leach's Storm Petrel at static positions (burrows). ZebraNet operates at a 100% duty cycle with an expected lifetime of 5 days. The system uses solar panels to recharge batteries. ZebraNet acquires GPS data every 3 minutes to keep a log of a zebra's position. GPS time synchronization is leveraged to execute a time-slotted routing protocol. Unfortunately we cannot use the same software for underground burrow motes since there is no option of solar recharging or view of the sky for GPS signal. ZebraNet, at the time of this writing, has not yet had a full long-term deployment so there is currently no thorough analysis of the reliability of their sensor network algorithms and design.

Chapter 9

Conclusion

Habitat and environmental monitoring represent an important class of sensor network applications. We are collaborating with biologists at the College of the Atlantic to define the continually refine the core application requirements. Because end users are ultimately interested in the sensor data, the sensor network system must deliver the data of interest in a confidence-inspiring manner. The low-level energy constraints of the sensor nodes combined with the data delivery requirements leave a clearly defined energy budget for all other services. Tight energy bounds and the need for predictable operation guide the development of application architecture and services. We have shown that the requirements of habitat monitoring applications may be achieved through our architecture and simple software components.

The habitat monitoring architecture met the requirements of the life scientists and streamed sensor readings from underground burrows to databases across the Internet for over four months. By using habitat monitoring as a driving application for researching long term deployments and power efficient sensor network solutions, we were able to build, analyze, and refine the design to produce a more robust system. Building a simple solution and performing frequent iterations has yielded a successful architecture for habitat monitoring.

We believe GDI is representative of many applications in this domain, there may be significant differences. To evaluate our implementation, we deployed an initial prototype network at the James San Jacinto Mountains Reserve (JMR) (33.48N, 116.46W) in Idyllwild, California. JMR is a 29-acre ecological preserve, representing just one of the University of California Natural Reserve System's 34 land holdings. The deployment uses a basic sensor package instead of the *Mica* Weather Board. JMR's climate is significantly different from GDI; it is arid and weather changes occur over long periods of time. Researchers at JMR are interested in microclimate readings over a large area as opposed to animal monitoring on GDI. Another environmental monitoring deployment is scheduled for the University of California Botanical Gardens Redwood Grove in Berkeley, California. This deployment focuses on profiling the microclimate variations that occur over three-dimensional cross-sections of a redwood tree. The deployment will be expanded to infer greater microclimate variations across the forest of redwoods, including the ability to compare the microclimates in the middle of the forest to those at the edge and thereby evaluate "edge effects". This deployment will further help refine the requirements for habitat and environmental monitoring applications.

The ultimate goal of habitat monitoring applications for sensor networks is an easy-to-use kit for life scientists to deploy on their own. Users will be able to tailor a mote's operation to a variety of experimental configurations. By allowing life scientists, rather than computer scientists, to control and customize the sensor network, the sampled data will be inherently more useful to the users. Through wireless sensor networks, life scientists will be able to retrieve data reliably from locations previously unaccessible at a scale and resolution previously unattainable.

Acknowledgments

Special thanks to Professor David Culler and Alan Mainwaring who have largely been the driving force behind this work. Many thanks are owed to Rob Szewczyk who helped tremendously with the software, deployment, and analysis of data. Thank you to John Anderson for exposing the biological impact of sensor networks, helping to define and refine the requirements for habitat monitoring, and trusting us to deploy the sensors on Great Duck Island, Eric Paulos, Ken Smith, and David Koizumi for designing and manufacturing the acrylic enclosures, Phil Buonodonna for his networking expertise, Andrew Peterson and Amanda Witherell for boat and travel logistics to Great Duck Island, and the College of the Atlantic for the use of Great Duck Island. This work was supported by the Intel Research Laboratory at Berkeley, DARPA grant F33615-01-C1895 (Network Embedded Systems Technology "NEST"), the National Science Foundation, and the Center for Information Technology Research in the Interest of Society (CITRIS).

Bibliography

[1] Julia Ambagis. Census and monitoring techniques for Leach's Storm Petrel (Oceanodroma leucorhoa). Master's thesis, College of the Atlantic, Bar Harbor, ME, 2002.

[2] Analog Devices. ADXL202e 2-axis Accelerometer. `http://www.analog.com/UploadedFiles/Datasheets/567227477ADXL202E_a.pdf`, October 2000.

[3] John G. T. Anderson. Pilot survey of mid-coast maine seabird colonies: An evaluation of techniques. In *Report to the State of Maine Department of Inland Fisheries and Wildlife*, Bangor, ME, 1995.

[4] John G. T. Anderson. College of the Atlantic. Personal communication, January 2003.

[5] John D. W. Barrick, John A. Ritter, Catherine E. Watson, Mark W. Wynkoop, John K. Quinn, and Daniel R. Norfolk. Calibration of NASA turbulent air motion measurement system. NASA Technical Paper 3610, Langley Research Center, December 1996.

[6] Alexis L. Blackmer, Joshua T. Ackerman, and Gabrielle A. Nevitta. Effects of investigator disturbance on hatching success and nest-site fidelity in a long-lived seabird, Leach's Storm-Petrel. *Biological Conservation*, 2003.

[7] Karen Carney and William Sydeman. A review of human disturbance effects on nesting colonial waterbirds. *Waterbirds*, 22:68–79, 1999.

[8] Alberto Cerpa, Jeremy Elson, Deborah Estrin, Lewis Girod, Michael Hamilton, and Jerry Zhao. Habitat monitoring: Application driver for wireless communications technology. In *2001 ACM SIGCOMM Workshop on Data Communications in Latin America and the Caribbean*, San Jose, Costa Rica, April 2001.

[9] Benjie Chen, Kyle Jamieson, Hari Balakrishnan, and Robert Morris. Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks. In *Proceedings of the 7th ACM International Conference on Mobile Computing and Networking*, pages 85–96, Rome, Italy, July 2001.

[10] Clairex Technologies, Inc. CL9P4L epoxy-encapsulated photoconductor. `http://www.clairex.com/datasheets/cl9p_series.pdf`, July 2001.

[11] Roger W. Clay, N. R. Wild, D. J. Bird, B. R. Dawson, M. Johnston, R. Patrick, and A. Sewell. A cloud monitoring system for remote sites. *Publications of the Astronomical Society of Australia*, 15(3):332–335, August 1998.

[12] Dallas Semiconductor. MAX6633 12-bit plus sign temperature sensor. `http://pdfserv.maxim-ic.com/arpdf/MAX6633-MAX6635.pdf`, August 2001.

[13] Amre El-Hoiydi. Aloha with preamble sampling for sporadic traffic in ad hoc wireless sensor networks. In *Proceedings of IEEE International Conference on Communications*, New York, NY, USA, April 2002.

[14] Deborah Estrin, Lewis Girod, Greg Pottie, and Mani Srivastava. Instrumenting the world with wireless sensor networks. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2001)*, Salt Lake City, UT, May 2001.

[15] Deborah Estrin, Ramesh Govindan, John S. Heidemann, and Satish Kumar. Next century challenges: Scalable coordination in sensor networks. In *Mobile Computing and Networking*, pages 263–270, 1999.

[16] Kevin Fall. A delay-tolerant network architecture for challenged internets. Technical Report IRB-TR-03-003, Intel Research, Berkeley, CA, February 2003.

[17] David Gay, Phil Levis, Rob von Behren, Matt Welsh, Eric Brewer, and David Culler. The nesC language: A holistic approach to networked embedded systems. In *Proceedings of the ACM Conference on Programming Language Design and Implementation*, June 2003.

[18] General Eastern. G-CAP 2 relative humidity sensor. `http://www.geinet.com/html/humGCAPspecs.htm`.

[19] Hamamatsu Photonics K.K. S1087 ceramic package photodiode with low dark current. `http://usa.hamamatsu.com/hcpdf/parts_S/S1087-01.pdf`, April 2001.

[20] David Happold. The subalpine climate at smiggin holes, Kosciusko National Park, Australia, and its influence on the biology of small mammals. *Arctic & Alpine Research*, 30:241–251, 1998.

[21] Jason Hill. Spec: single chip mote integrating communication, computation, and sensing. `http://www.cs.berkeley.edu/~jhill/spec/`, March 2003.

[22] Jason Hill and David Culler. Mica: a wireless platform for deeply embedded networks. *IEEE Micro*, 22(6):12–24, November/December 2002.

[23] Jason Hill, Robert Szewczyk, Alec Woo, Seth Hollar, David Culler, and Kristofer Pister. System architecture directions for networked sensors. In *Proceedings of the International Conference on Architectural Support for Programming Languages and Operating Systems*, Boston, MA, USA, November 2000.

[24] Chuck Huntington, Ron Butler, and Robert Mauck. *Leach's Storm Petrel (Oceanodroma leucorhoa)*, volume 233 of *Birds of North America*. The Academy of Natural Sciences, Philadelphia and the American Orinthologist's Union, Washington D.C., 1996.

[25] Intersema. MS5534A barometer module. `http://www.intersema.com/pro/module/file/da5534.pdf`, October 2002.

[26] Intrinsyc Corporation. Cerfcube embedded StrongARM system. `http://www.intrinsyc.com/products/cerfcube/`.

[27] Philo Juang, Hidekazu Oki, Yong Wang, Margaret Martonosi, Li-Shiuan Peh, and Daniel Rubenstein. Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with zebranet. In *Proceedings of ASPLOS X*, San Jose, CA, October 2002.

[28] Joseph M. Kahn, Randy H. Katz, and Kristofer S. J. Pister. Emerging challenges: Mobile networking for smart dust. *Journal of Communications and Networks*, 2(3), September 2000.

[29] David Kiernan. Battery compartment design guidelines for equipment using lithium-sulfur dioxide batteries. Technical Report CECOM-TB-7 (REV A), U.S. Army Communications-Electronics Command, October 1997.

[30] Julius Kusuma, Lance Doherty, and Kannan Ramchandran. Distributed compression for wireless sensor networks. In *Proceedings of ICIP 2001*, Thessalonika, Greece, October 2001.

[31] Philip Levis and David Culler. Maté: A tiny virtual machine for sensor networks. In *Proceedings of the International Conference on Architectural Support for Programming Languages and Operating Systems*, San Jose, CA, October 2002.

[32] Samuel Madden, Michael Franklin, Joseph Hellerstein, and Wei Hong. TAG: a Tiny AGgregation service for ad-hoc sensor networks. In *Proceedings of Operating Systems Design and Implementation*, Boston, MA, December 2002.

[33] Samuel Madden, Wei Hong, Joseph Hellerstein, and Kyle Stanek. TinyDB: a query processing system for extracting information from a network of TinyOS sensors. `http://telegraph.cs.berkeley.edu/tinydb/`.

[34] Alan Mainwaring, Joseph Polastre, Robert Szewczyk, David Culler, and John Anderson. Wireless sensor networks for habitat monitoring. In *Proceedings of the ACM International Workshop on Wireless Sensor Networks and Applications (WSNA'02)*, Atlanta, GA, September 2002.

[35] Melexis Microelectronic Integrated Systems. MLX90247 discrete infrared thermopile detector. `http://www.melexis.com/prodfiles/mlx90247.pdf`, August 2002.

[36] Melexis Microelectronic Integrated Systems. MLX90601 infrared thermopile module. `http://www.melexis.com/prodfiles/mlx90601.pdf`, August 2002.

[37] Ian Nisbet. Disturbance, habituation, and management of waterbird colonies. *Waterbirds*, 23:312–332, 2000.

[38] Onset Computer Corporation. HOBO weather station. `http://www.onsetcomp.com`.

[39] Kris Pister, Barbara Hohlt, Jaein Jeong, Lance Doherty, and J.P. Vainio. Ivy: A sensor network infrastructure for the University of California, Berkeley College of Engineering. `http://www-bsac.eecs.berkeley.edu/projects/ivy/`.

[40] Sensirion. SHT11/15 relative humidity sensor. `http://www.sensirion.com/en/pdf/Datasheet_SHT1x_SHT7x_0206.pdf`, June 2002.

[41] Slobodan Simic and Shankar Sastry. Distributed environmental monitoring using random sensor networks. In *Proceedings of Workshop on Information Processing in Sensor Networks*, Palo Alto, CA, April 2003.

[42] Bruno Sinopoli, Cory Sharp, Luca Schenato, Shawn Schaffert, and Shankar Sastry. Distributed control applications within sensor networks. *Proceedings of the IEEE*, November 2003.

[43] Texas Advanced Optoelectronic Solutions. TSL2550 ambient light sensor. `http://www.taosinc.com/pdf/tsl2550-E39.pdf`, September 2002.

[44] Marco Toapanta, Joe Funderburk, and Dan Chellemi. Development of Frankliniella species (Thysanoptera: Thripidae) in relation to microclimatic temperatures in vetch. *Journal of Entomological Science*, 36:426–437, 2001.

[45] University of California, Berkeley. TinyOS: A component-based OS for the networked sensor regime. `http://webs.cs.berkeley.edu`, 2003.

[46] Hanbiao Wang, Deborah Estrin, and Lewis Girod. Preprocessing in a tiered sensor network for habitat monitoring. In *EURASIP JASP special issue on sensor networks*, 2003.

[47] Brent West, Paul Flikkema, Thomas Sisk, and George Koch. Wireless sensor networks for dense spatio-temporal monitoring of the environment: A case for integrated circuit, system, and network design. In *Proceedings of IEEE CAS Workshop on Wireless Communications and Networking*, Notre Dame, IN, August 2001.

[48] Ya Xu, John Heidemann, and Deborah Estrin. Geography-informed energy conservation for ad hoc routing. In *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking*, pages 70–84, Rome, Italy, July 2001.

[49] Wei Ye, John Heidemann, and Deborah Estrin. An energy-efficient mac protocol for wireless sensor networks. In *Proceedings of the 21st International Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2002)*, New York, NY, USA, June 2002.

Appendix A

Packet Structure

TinyOS Message

```
typedef struct TOS_Msg
{
    uint16_t addr;     // Destination address
    uint8_t  type;     // Active Message handler ID
    uint8_t  group;    // Receive group ID
    uint8_t  length;   // Length of the data section
    int8_t   data[29]; // Data packet up to 29 bytes
    uint16_t crc;      // CRC checksum
} TOS_Msg;
```

GDI Data Payload

```
typedef struct gdimsg_t
{
    uint16_t sender_id;
    uint16_t photo_data;
    uint16_t temp_data;
    uint16_t thermopile_data;
    uint16_t thermistor_data;
    uint16_t humidity_data;
    uint8_t  volts_data;
    uint32_t seqno;
    uint16_t intersema_temp_raw;
    uint16_t intersema_pressure_raw;
    int16_t  intersema_pressure;
    int16_t  intersema_temp;
} gdimsg_t;
```

Appendix B

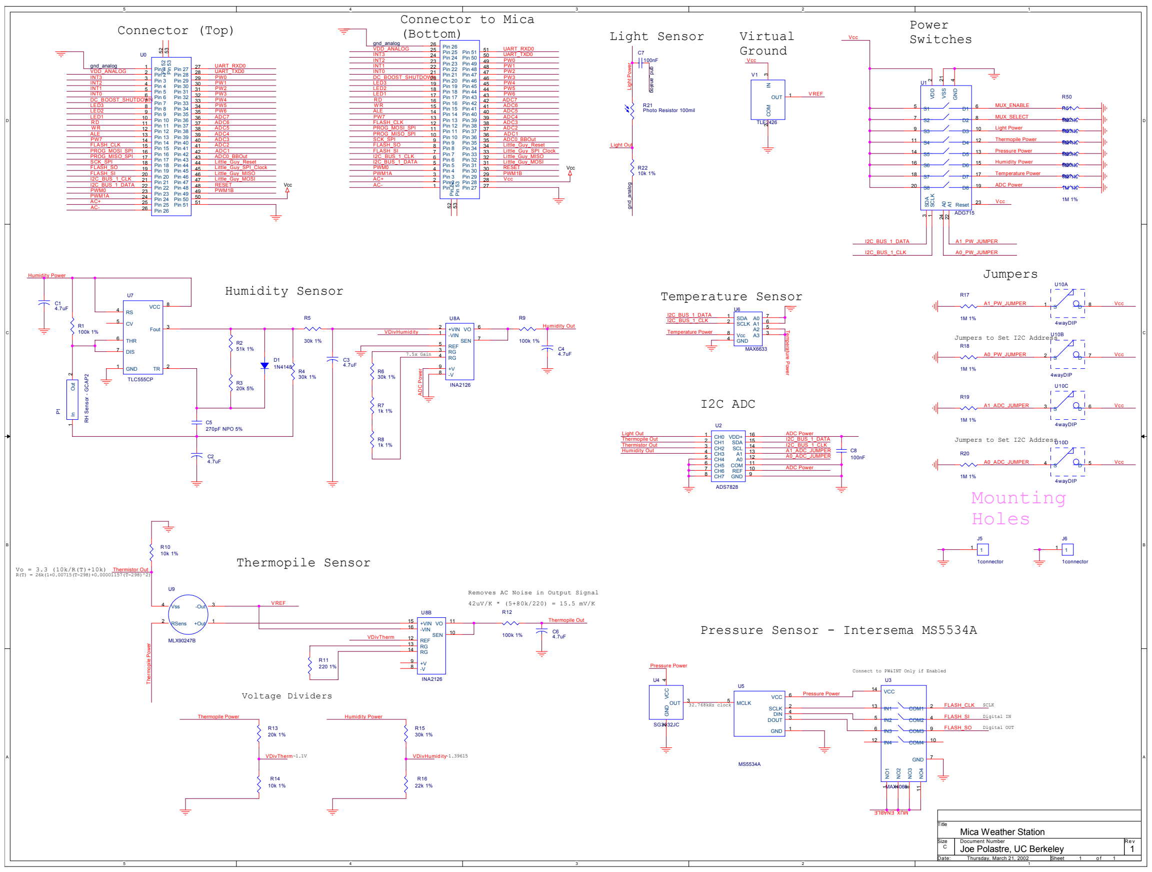
Database Schema

```
CREATE TABLE "weather" (
        "packet_time" timestamp with time zone,
        "node_id" integer,
        "light_reading" integer,
        "temp_reading" integer,
        "thermopile_reading" integer,
        "thermistor_reading" integer,
        "humidity_reading" integer,
        "intersema_pressure_reading" integer,
        "intersema_pressure_raw" integer,
        "intersema_temp_reading" integer,
        "intersema_temp_raw" integer,
        "voltage_reading" integer,
        "seqno" integer,
        "crc" integer,
        "packet" bytea
);

CREATE TABLE "mote_data" (
        "node_id" integer,
        "intersema_calibration" bytea,
        "gps_zone" integer,
        "gps_hemisphere" char,
        "easting" integer,
        "northing" integer,
        "location" varchar(80),
        "asset" integer,
        "wb_num" integer
);
```

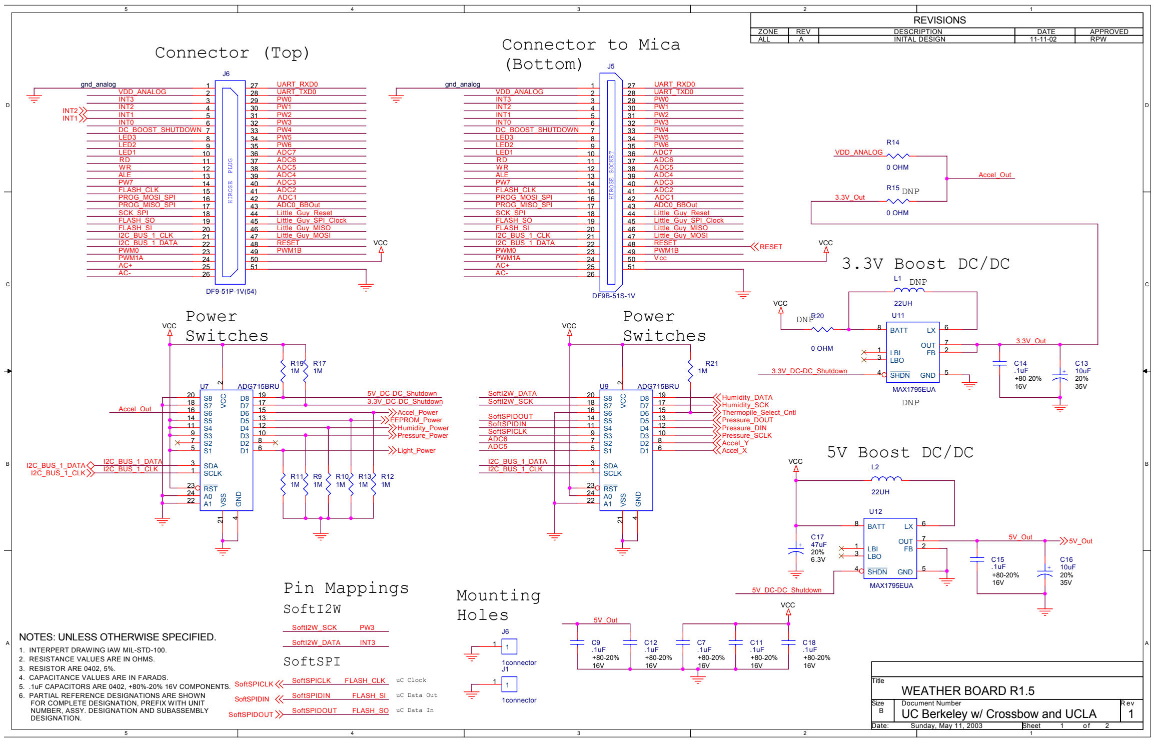
Appendix C

Mica Weather Board Revision 1.0 Schematic



Appendix D

Mica Weather Board Revision 1.5 Schematic

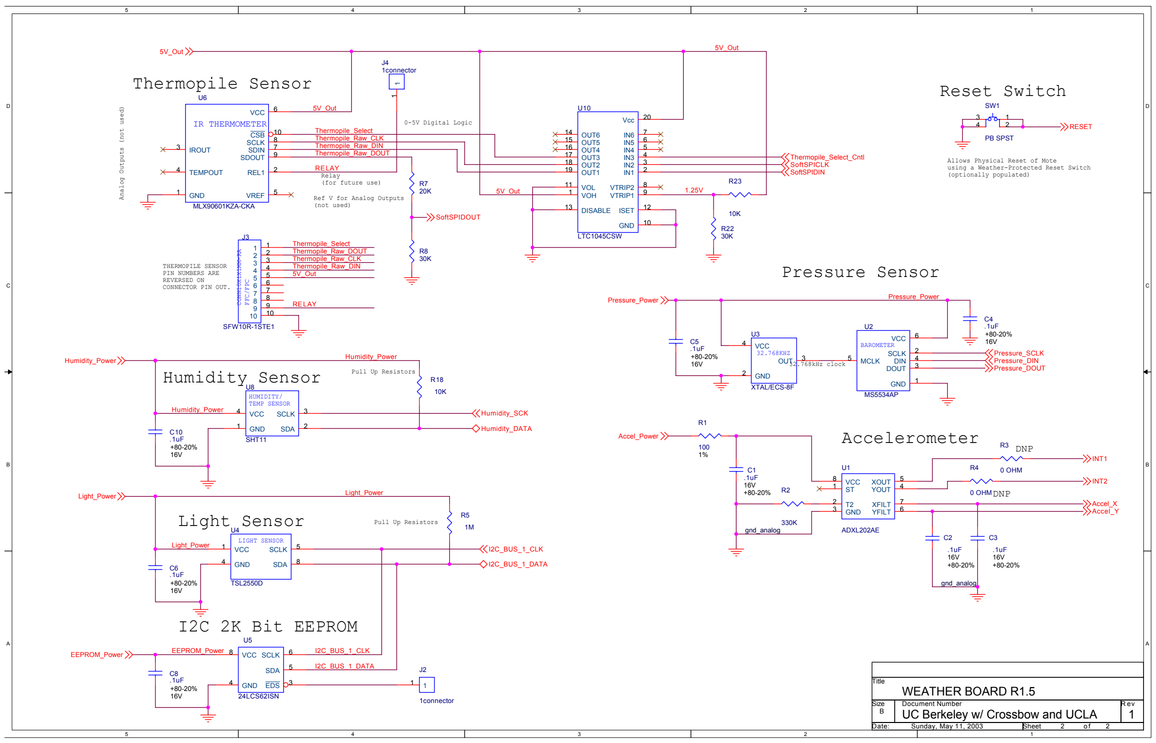


NOTES: UNLESS OTHERWISE SPECIFIED.

1. INTERPERT DRAWING IAW MIL-STD-100.
2. RESISTANCE VALUES ARE IN OHMS.
3. RESISTOR ARE 0402, 5%.
4. CAPACITANCE VALUES ARE IN FARADS.
5. .1uF CAPACITORS ARE 0402, +80%-20% 16V COMPONENTS.
6. PARTIAL REFERENCE DESIGNATIONS ARE SHOWN FOR COMPLETE DESIGNATION, PREFIX WITH UNIT NUMBER, ASSY. DESIGNATION AND SUBASSEMBLY DESIGNATION.

WEATHER BOARD R1.5

UC Berkeley w/ Crossbow and UCLA


Thermopile Sensor
U6
IR THERMOMETER
MLX90601KZA-CKA
5V_Out
J4
1connector
0-5V Digital Logic
Thermopile_Select
Thermopile_Raw_CLK
Thermopile_Raw_DIN
Thermopile_Raw_DOUT
RELAY
Relay (for future use)
Ref V for Analog Outputs (not used)
Analog Outputs (not used)
R7 20K
R8 30K
SoftSPIDOUT
U10
LTC1045CSW
Thermopile_Select_Cntl
SoftSPICLK
SoftSPIDIN
R23
1.25V
10K
R22 30K
J3
SFW10R-1STE1
THERMOPILE SENSOR PIN NUMBERS ARE REVERSED ON CONNECTOR PIN OUT.
Reset Switch
SW1
PB SPST
RESET
Allows Physical Reset of Mote using a Weather-Protected Reset Switch (optionally populated)
Pressure Sensor
Pressure_Power
C5 .1uF +80-20% 16V
U3
32.768KHZ
XTAL/ECS-8F
32.768kHz clock
U2
BAROMETER
MS5534AP
C4 .1uF +80-20% 16V
Pressure_SCLK
Pressure_DIN
Pressure_DOUT
Humidity Sensor
Humidity_Power
Pull Up Resistors
R18 10K
U8
HUMIDITY/ TEMP SENSOR
SHT11
C10 .1uF +80-20% 16V
Humidity_SCK
Humidity_DATA
Light Sensor
Light_Power
R5 1M
U4
LIGHT SENSOR
TSL2550D
C6 .1uF +80-20% 16V
I2C_BUS_1_CLK
I2C_BUS_1_DATA
I2C 2K Bit EEPROM
EEPROM_Power
U5
24LCS62ISN
C8 .1uF +80-20% 16V
J2
1connector
Accelerometer
Accel_Power
R1 100 1%
C1 .1uF 16V +80-20%
R2 330K
gnd_analog
U1
ADXL202AE
R3 DNP
R4 0 OHM DNP
INT1
INT2
Accel_X
Accel_Y
C2 .1uF 16V +80-20%
C3 .1uF 16V +80-20%
Title
WEATHER BOARD R1.5
Size B
Document Number
UC Berkeley w/ Crossbow and UCLA
Rev 1
Date: Sunday, May 11, 2003
Sheet 2 of 2