CargoNet: Micropower Sensate Tags for Supply-Chain Management and Security

by

Mateusz Ksawery Malinowski

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2007

Author ....
Department of Electrical Engineering and Computer Science
February 2, 2007

Certified by ....
Joseph A. Paradiso
Associate Professor
Program in Media Arts and Sciences
Thesis Supervisor

Accepted by ....
Arthur C. Smith
Chairman, Department Committee on Graduate Students

CargoNet: Micropower Sensate Tags for Supply-Chain Management and Security

by

Mateusz Ksawery Malinowski

Submitted to the Department of Electrical Engineering and Computer Science
on February 2, 2007, in partial fulfillment of the
requirements for the degree of
Master of Engineering in Electrical Engineering and Computer Science

Abstract

This thesis describes the development of a system of sensate active RFID tags for supply-chain management and security applications, necessitated by the current lack of commercial platforms capable of monitoring the state of shipments at the crate and case level. To make a practical prototype, off-the-shelf components and custom-designed circuits that minimize power consumption and cost were assembled and integrated into an interrupt-driven, quasi-passive system that can monitor, log, and report environmental conditions inside a shipping crate while consuming only 23.7 microwatts of average power. To prove the feasibility of the system, the tags were tested in the laboratory and aboard transport conveyances.

Thesis Supervisor: Joseph A. Paradiso
Title: Associate Professor
Program in Media Arts and Sciences

Acknowledgments

Foremost, I would like to thank my thesis supervisor, Professor Joseph Paradiso, who has provided me with many opportunities and challenges over the years. He has consistently given great ideas and criticism, and regularly demanded the (almost?) impossible, putting great faith in my rising to the task. At the same time, he has offered support when I have struggled, at even the darkest hours of the night.

Day to day, I have been relying on the other students in the Responsive Environments Group, and I would like to thank them for their strategic advice and quick hacks. I cannot imagine working with a more skilled and creative group of people, and have greatly enjoyed working with you all as an undergraduate and again for the past year and a half. Special thanks go out to Mark Feldmeier, who first showed me The Way of the Transistor and who has continued to help with circuit design, and Joshua Lifton and Mathew Laibowitz for their programming help.

Additional thanks go out to PepsiCo and the Things That Think consortium for funding this research, and to Mary Murphy-Hoye of Intel and Julius Akinyemi of PepsiCo for providing the initial inspiration for CargoNet and for their continuing excitement and suggestions throughout the course of the project. I would also like to thank again the team involved in Intel's Intelligent Container Project—Richard Tyo, George Cavage, and Scott Thomas—for letting me participate in their tests in New Jersey and Singapore.

Lisa Lieberson helped with crucial, last-minute logistics, while Zoz Brooks and Bo Morgan helped edit this thesis. Matthew Moskwa tested many of the CargoNet sensor circuits. I would like to thank you all.

Finally, I would like to thank my friends, family, and especially parents, all of whom I promise to visit now that this is finished.

Contents

List of Figures 9

List of Tables 11

1 Introduction 13

1.1 Synopsis . . . 13

1.2 A Brief Introduction to RFID . . . 14

1.3 The Impetus for Active RFID in the Logistics Industry . . . 15

1.3.1 Globalization and the Need for Visibility . . . 16

1.3.2 New Government Directives . . . 18

1.4 Commercial Applications of Active RFID to the Supply Chain . . . 20

1.4.1 Savi Technology: EchoPoint and SensorTag . . . 20

1.4.2 Sensitech: ColdStream and TempTaleRF . . . 20

1.4.3 Safefreight Technology: EnCompass and Qualcomm: OmniTracs . . 21

1.4.4 Motes and Other Wireless Networked Sensors . . . 22

1.5 Low-Power Strategies for a New System of Active Tags . . . 23

2 Hardware Description 27

2.1 Project History . . . 28

2.2 System Overview . . . 29

2.3 Core Components . . . 31

2.3.1 MSP430 Microcontroller . . . 31

2.3.2 CC2500 RF Transceiver . . . 32

2.3.3 Power . . . 33

2.4 Sensors and Other Components . . . 36

2.4.1 Quasi-Passive Wakeup . . . 36

2.4.2 Dynamic Thresholds . . . 38

2.4.3 Real-Time Clock . . . 39

2.4.4 Humidity and Temperature Sensors . . . 41

2.4.5 Breach Sensors . . . 46

2.4.6 Tilt Sensor . . . 48

2.4.7 Shock Sensor . . . 48

2.4.8 Vibration Dosimeter . . . 51

2.4.9 RF Detector . . . 52

2.4.10 Piezoelectric Microphone . . . 54

2.4.11 External Flash Memory . . . 55

2.5 Summary . . . 55

3 Firmware Design **59**
3.1 Operating Principles 59
3.2 Firmware Implementation 60
3.2.1 Sensor Object Abstraction 61
3.2.2 Dynamic Thresholds 61
3.2.3 Callbacks 62
3.2.4 Scheduler 62
3.2.5 Flash Memory Writes 64
3.2.6 Analog-to-Digital Converter 67
3.3 Sensor Control Firmware 68
3.3.1 Switch Sensors 69
3.3.2 Sampled Sensors 69
3.4 Summary 70

4 Testing and Analysis **73**
4.1 Singapore Test 73
4.1.1 Test Setup 75
4.1.2 Data Collection and Import 77
4.1.3 Real-Time Clock Performance 79
4.1.4 Battery Life 80
4.1.5 Tilt and Other Switched Sensors 83
4.1.6 Vibration Dosimeter 85
4.1.7 Piezoelectric Microphone 86
4.1.8 Temperature Sensors 87
4.1.9 Summary of Singapore Tests 90
4.2 Additional Laboratory Testing 90
4.2.1 Low-Cost Humidity Sensor Calibration 92
4.2.2 Humidity Sensor Temperature Dependence 93
4.3 Express Courier Service Test 95
4.3.1 Experimental Setup 95
4.3.2 Test Results 96
4.4 Summary and Future Work 99

5 Conclusion **103**

A Board Layouts and Schematic **105**

Bibliography **108**

List of Figures

1.1 ST-676 container tag from Savi Technology 21
1.2 TempTaleRF tag from Sensitech 22

2.1 Photographs of the CargoNet active tag at various stages of its development 30
2.2 Diagram of the CargoNet system 30
2.3 CC2500 daughter board photograph. 33
2.4 Schematic of CC2500 daughter board designed by Mathew Laibowitz. . . . 34
2.5 Discharge curves of various battery chemistries. 35
2.6 Quasi-passive wakeup scheme block diagram 37
2.7 Photograph of low-cost humidity sensor. 42
2.8 Detail of humidity sensors. 42
2.9 Response of a resistive humidity sensor 43
2.10 Linearization circuit for low-cost humidity sensor. 43
2.11 Output of linearizing circuit for low-cost humidity sensor 44
2.12 Temperature response of resistive humidity sensor. 45
2.13 Circuit schematic for phototransistor-based light detector/meter. 47
2.14 Breach sensor circuit schematic 48
2.15 Shock detector circuit diagram 50
2.16 Vibration dosimeter circuit diagram 52
2.17 RF detector circuit schematic 53
2.18 A piezoelectric microphone can wake the microcontroller on loud sounds. . 54
2.19 Frequency response of the 25LM025 piezoelectric microphone. 54

3.1 A diagram of the firmware state machine. 60
3.2 Example of a sensor object abstraction. 61
3.3 Definition of `bat` object and its initialization procedure. 62
3.4 ADC sampling code (with callbacks). 63
3.5 Timer_A scheduler task addition code. 64
3.6 Timer_A scheduler task removal code. 65
3.7 Timer_A interrupt handler code. 66
3.8 Flash memory double-buffering example code. 67
3.9 The `adc12.sample(...)` function prepares the ADC for collecting samples. 68
3.10 ADC interrupt request handler. 69
3.11 Tilt sensor interrupt-handler code. 70
3.12 Low-cost humidity sensor measurement algorithm. 71
3.13 Humidity sensor ADC callback code. 71

4.1 Shipping containers arranged in a stack in Kearny, New Jersey. 74

4.2 An active tag packaged for testing aboard a ship in East Asia. 75
4.3 A sketch of the placement of tags inside the shipping container. 76
4.4 A photograph of the tags inside the shipping container. 76
4.5 Continuity of RTC time can be restored despite resets. 79
4.6 Tag battery voltages. 81
4.7 An example of a firmware trap. 82
4.8 Current consumption of the active tag during the active part of its cycle. . 84
4.9 Tilt switch data for Tag #7. 85
4.10 Raw vibration dosimeter data collected by Tag #7. 86
4.11 Running sum of the dosimeter vibrations. 87
4.12 Responses of different sensors to large shocks. 88
4.13 Illustration of dynamic threshold in microphone sensor. 89
4.14 Comparison between temperature sensors. 91
4.15 Comparison of internal temperature sensor offsets. 91
4.16 Results of low-cost humidity sensor calibration tests. 94
4.17 Average current consumption of a CargoNet tag during an overnight journey. 97
4.18 Increased current consumption caused by vibration dosimeter output rising between resets. 98
4.19 Comparison of quasi-passive wakeup to sampling: shock sensor. 100
4.20 Comparison of quasi-passive wakeup to sampling: microphone. 101

A.1 CargoNet version 4 PCB top layer. 105
A.2 CargoNet version 4 PCB bottom layer. 106
A.3 CargoNet version 4 PCB internal ground plane (negative). 106
A.4 CargoNet version 4 PCB internal V_{CC} plane (negative). 106
A.5 CargoNet version 4 schematic diagram. 107

List of Tables

1.1 Characteristics of popular active RFID platforms for asset management. . . 24

2.1 TelosB partial bill of materials 28
2.2 Partial overview of available real-time clock integrated circuits. 40
2.3 Bill of materials for CargoNet version 4. 56
2.4 Quiescent power budget 57

3.1 Identification codes and data fields used in the event log. 66

4.1 The distribution of tags between the three placement groups. 77
4.2 Longevity of tags during the Singapore test. 81
4.3 Current and power consumption of CargoNet during DHL test. 96

Chapter 1

Introduction

This thesis describes the development of the CargoNet system of micropower sensate active radio-frequency tags for supply-chain management and security applications. By using micropower components and circuits and by keeping the tag's microcontroller in a sleep state for most of the time through quasi-passive wakeup [32] and dynamic threshold techniques, which allow the system to adapt to repeated stimuli, the tags should able to measure and log multiple environmental parameters for years, while running off a single coin-cell battery. This performance has been achieved at a reasonable cost—the components, in small quantities, cost only USD 40—and in a small form factor.

The CargoNet system spans the divide between traditional radio-frequency identification (RFID) and wireless sensor networks (WSN), with tags providing wireless access to both identification information and sensor data. The micropower components and efficient operation of the tags extend their maximum deployment time past that achieved by most wireless sensor networks, while a full environmental sensor suite allows multimodal monitoring of tagged objects in addition to identification and tracking. By combining the strengths of the RFID and WSN fields, the CargoNet system enables a variety of novel applications such as supply-chain visibility.

1.1 Synopsis

This chapter begins with a brief overview of RFID technology and develops the motivation behind the current research on active tags, while contrasting the approach taken by the author with related efforts currently in testing and production elsewhere.

Chapters 2 and 3 discuss the development of the system hardware and firmware, respectively, which were designed specifically for this active tagging application with the triple goals of minimizing cost, extending battery lifetime, and maintaining a small device size.

Chapter 4 describes the tests performed with the hardware, both in the laboratory and in the outside world (on a container ship traveling through the South China Sea and aboard an express courier aircraft), which provided an opportunity to assess the capabilities of the tag as a sensor platform and explore the potential of the tag-reader-database system as a whole.

All technical drawings are available in the appendices, which follow the conclusion.

1.2 A Brief Introduction to RFID

Radio-frequency identification (RFID) technology seeks to continue the revolution initiated by bar codes and fully automate the identification and tracking of items, whether at the checkout aisle, in the warehouse, or in countless other places. In contrast to bar code systems and other passive optical identification systems—which use reflected light, requiring a direct line of sight between a reader and the tagged object—RFID systems employ radio-frequency electromagnetic radiation or magnetic fields. These fields permeate through most materials, and therefore do not require a human operator to align the tagged object and the reader [20].

The most commonly used types of RFID tags are composed of an antenna that receives power and interrogation signals from a reader, and an integrated circuit that responds back through the antenna with a unique identification code. Because of the presence of an integrated circuit, an RFID tag can store much more information than a bar code and even dynamically change the data depending on external conditions. These capabilities further extend the advantages due to the use of radio waves instead of reflected light, such as the ability to read numerous tags within a short time span, to read tags continuously, even as the tagged object is moving, or to read the contents of large packages without opening them and sorting their contents [58].

Due to the potential of RFID, the past decade has seen gradual improvements in the capabilities of the technology and great strides in its commercialization, leading to an industry growth rate that surpasses that of cellular telephones [20]. Recent popular excitement about RFID has further been stoked by retailer Wal-Mart's announcement that it will require 100 of its largest suppliers to use RFID tags by 2005, a decision that was echoed by the United

States Department of Defense and other large retailers several months later. Despite the excitement and large investment in the technology in some quarters, many of the potential benefits of RFID have not yet been widely distributed. Many suppliers buy the simplest passive tags and apply them right as products are about to ship to the customer, without consideration to the outsize benefits that can result from integrating the technology into their internal processes [37]. Part of the hesitation in embracing RFID more fully comes from analyst forecasts warning that the technology will completely change within two years and recommending against embracing hardware that will likely be superseded [69].

Sure enough, rather than simply improving read ranges and success rates, researchers are transforming the field by, among other things, implementing sensing on-board passive tags. Passive tags have been developed that are capable of sensing high temperatures with fuses that melt above a particular threshold [74] or with tilt switches that can detect when household objects are manipulated (monitoring of the elderly is one application) [48]. Other platforms that promise to passively sense bacterial growth and corrosive chemicals are also under development [73, 74]. Although the above tags implement single-bit sensing because it is easier and requires little or no power scavenged from the reader's interrogation pulse, there has also been recent work in multi-bit reporting of such environmental conditions as ambient light. This work is doubly important because rather than using a dedicated IC for communication and sensing, the platform employed a fully programmable, low-power microcontroller, opening the door to quicker innovation and myriad new applications [59].

It is in the context of this thriving but still malleable technological landscape that the work presented in this thesis has been performed, in the hope of demonstrating the feasibility and potential benefit of active sensate RFID tags in the logistics industry.

1.3 The Impetus for Active RFID in the Logistics Industry

Supply-chain management is expected to be among the fastest-growing applications of RFID technology [20], and there are numerous examples of logistics companies evolving from passive to active tags, in order to utilize additional capabilities such as localization and monitoring [3].

1.3.1 Globalization and the Need for Visibility

An increasing number of companies are migrating their operations overseas. Companies that relied on global trade for only 5% of their business 10 years ago may be dependent on the overseas markets for as much as 50% today [71]. Despite this apparent explosion in global trade, many companies are failing to capitalize on its promise. In a 2005 survey, 91% of the companies polled were not realizing the expected cost reductions of operating internationally, mostly due to "lead times [inhibiting their] ability to respond to market demands" and "product cost savings eroded by unanticipated supply chain costs" [68].

Practicing logisticians and analysts recommend flexibility in supply chain operations: an ability to switch suppliers or destination ports allows operations to survive shortages, strikes, and other crises [68]. As supply-chain management shifts its attention from ever-decreasing costs and improvements in quality to managing risk and the continuity of supply [71], many operations have been insuring against shortages by rebuilding safety stock—a turnaround from the zero-inventory approach pioneered several years ago by electronics companies like Dell and Cisco Systems [51]. Maintaining safety stock is not a long term strategy, however; authors agree that companies need to invest in technology improvements to their supply chain to increase efficiency [71, 68, 51, 70].

International transport usually involves multiple handoffs as the goods pass from one tier of supplier to another, then on to carriers, and finally to various tiers of customers. The whole system is only as good as the weakest link, and traditionally, errors would only be detected when customers called to complain [51]. In order to preempt missed deliveries and otherwise improve supply chain efficiency, managers require end-to-end visibility: the ability to track all items currently traveling through the supply chain in real time. Unfortunately, efforts to track goods down the length of the supply chain have been constrained by the unwillingness of competing carriers to share information and the costs of fully integrating disparate operations, many of which still rely on paper records which must be keyed into a computer system at every step [51].

RFID tags are heralded as the revolutionary technology that will finally make supply-chain integration and end-to-end visibility possible. Because tagged items can be interrogated and their identity read at every step of the journey without any human intervention, reports can be collected with arbitrary granularity. Since they begin in electronic form, they can be easily distributed around the world. Such real-time data can prevent unwelcome scheduling surprises, while former “black holes” in the supply chain can be carefully examined in search of inefficiencies and potential adjustments [71].

Besides their disruptive potential as an investigative tool for managing the entire supply chain, RFID tags also provide an immediate, low-level advantage over bar codes, the previous method of tracking goods. According to studies performed by Marks and Spencer, a British supermarket chain, RFID tags decreased the read times of dollies loaded with multiple trays of goods by 80% and cut the time to unload a truck from 18 minutes to three, due to the ability of RFID technology to perform multiple reads in short succession and to execute a greater number of correct reads [24].

RFID technology fully automates the identification of goods in the supply chain, accelerating loading and unloading as well as allowing tracking of packages as they travel through the supply chain. Unfortunately, shipped goods are susceptible to undesirable environmental conditions such as shock, tilt, and extremes of temperature and humidity, and passive RFID tags cannot monitor these conditions; they cannot peer *inside* the package and make inferences about the state of its contents. While one can preempt damage through ever thicker padding or by packaging the goods with a desiccant to maintain low humidity or with dry ice to maintain low temperature, these preventive systems can still fail.

Damage indicators become particularly desirable when transporting sensitive items along a route with multiple segments. If damage is suspected along any segment, the goods can be diverted mid-shipment for inspection, which may save money (damaged goods are not transported along the remainder of the route) as well as time (a new shipment can be dispatched, reducing the delay and surprise the customer would otherwise experience).

In a recent review of successful strategies in global logistics [51], the CEO of a company that performs analyses in support of pharmaceutical trials revealed that his company does not use temperature monitoring tags when transporting samples, instead trusting the integrity of the shipments to his express carrier and a block of dry ice. Apparently damage is so unlikely that the tags are not worth the cost. When multiple carriers are involved, however, and damage does occur, monitoring can be an important tool for assigning responsibility. Even when using only one carrier, monitoring may be beneficial from the carrier's perspective in finding faulty links in the supply chain where goods are damaged.

A successful implementation of the above requires a damage monitor that logs abnormal environmental parameters together with a time stamp, making it possible to ascertain where the damage occurred by cross-referencing the RFID tracking records. The most elegant and efficient solution would combine the damage-monitoring and tracking functions on one device, reading off the internal state at the same time as the identification number used for tracking. Such an active RFID tag, consisting of integrated communication, sensing, and processing components, would finally allow logisticians to know not only the location of an item in the supply chain, but also its state—finally achieving full visibility.

1.3.2 New Government Directives

In addition to the pursuit of ever finer visibility, new directives and research solicitations issued by the United States Department of Homeland Security (DHS) and its Homeland Security Advanced Research Programs Agency (HSARPA) in the wake of the 11 September 2001 terrorist attacks have been a powerful driver in the development of active RFID.

Leveraging the already widespread adoption of electronic data systems for supply-chain visibility (as mentioned in Section 1.3.1 above), DHS mandated the submission of electronic manifests for all cargo entering or leaving the United States. The proposed rule of 23 July 2003 would require carriers to electronically submit information on the provenance and contents of all shipments, allowing the Bureau of Customs and Border Protection (CBP) to assess risk and better focus inspections of the over 5.7 million containers entering the United States every year [52], with the intent of combating smuggling and ensuring the safety and security of the cargo. The information is to be submitted to CBP 24 hours before the cargo is loaded aboard a ship [11], with appropriate exceptions (for example, two hours before reaching the border) in the case of short-haul rail and truck shipments and express airplane shipments [12]. Although carriers are not obligated to maintain their records in electronic form, the electronic filing requirement becomes much less burdensome in the case of fully automated systems, providing an incentive for carriers to expand visibility in their operations.

Although any partially or fully automated system for shipment tracking (such as those that use bar codes or RFID) would permit easier compliance with the above DHS directives, there are other recent DHS efforts specifically promoting active tagging of shipments. Under the Free and Secure Trade (FAST) program, participating importers receive expedited processing at the United States-Canada and United States-Mexico borders in exchange for verifiably "enhancing the security of their manufacturing plants, warehouses and shipping systems". To qualify for this preferential processing, all shipments must be sealed by the manufacturers before leaving their secure facilities for the border [4]. Similarly, maritime carriers cooperating with CBP are enjoying less frequent noninvasive and invasive inspections [15].

Seals are also a crucial component of CBP's Container Security Initiative, which aims to pre-screen maritime containers for radiological and other risks before departure for the United States [72]. Once a container has been screened, its integrity is protected by a seal consisting of a metal bolt or loop of cable that—once secured—cannot be released without breaking the seal (as per ISO standard 17712). Although dependable, these seals need to be visually inspected by a person and their status logged—a time- and labor-intensive process. To increase the number of containers that it can process and to lessen the regulatory burden on carriers, CBP has been promoting the development and adoption of electronic seals [72], which would wirelessly and automatically report on the integrity of the container, allowing for quicker inspections and providing carriers with an added layer of protection against pilfering.

Some of these electronic seals or "e-seals" are already in use, though widespread adoption is unlikely in the near future due to lack of a global standard (eventually to be called ISO 18185), data encryption procedures, or even a globally available frequency to be used for communication of seal status. Another concern is the divergence "between cargo security requirements and the supply chain management goals of protecting and efficiently utilizing assets". Crucially, e-seals must not be rewritable, as that would make it possible to disguise signs of tampering, even though rewritability would enable the tag to store manifests and other dynamic data. The International Standards Organization is working on a separate standard, ISO 17363, that describes a rewritable RFID tag for just this purpose [15].

Finally, mechanical and electronic seals can only reveal entry or attempted entry via the doors of the container. DHS is currently funding research into an "Advanced Container Security Device" which can detect intrusion through any of the six walls of a container [15], while the Secure Carton Initiative seeks to bring monitoring down to the level of the carton (or case), and would ensure cargo security before it enters and after it leaves the container [46].

1.4 Commercial Applications of Active RFID to the Supply Chain

As can be seen in Section 1.3.2, the Department of Homeland Security has been promoting active RFID for a number of applications in the supply chain in addition to the ones engendered by the natural push towards greater efficiency mentioned in Section 1.3.1. This potential demand has prompted numerous companies to invest in active RFID research, and while widespread acceptance awaits the development of international standards [15], their products are finding application in various specialized contexts.

1.4.1 Savi Technology: EchoPoint and SensorTag

Savi Technology offers a range of active RFID products, from simple tags that can only store an ID, through rich data tags capacious enough to store an entire container manifest, to sensate active RFID tags that can monitor the environmental conditions inside a container [54].

The ST-676, for example, is a sensate tag designed specifically for monitoring of shipping containers. As can be seen in Figure 1.1, its form factor allows it to be clamped around the door of an ISO-standard shipping container, allowing the tag to monitor breach in addition to exposing the tag antennas outside the steel body of the container for better signal reception. The core of the tag remains inside, where it monitors ambient light (another indication of breach), temperature, humidity, and shock, logging these environmental conditions for later analysis or sending alerts when they exceed programmed thresholds [55].

Rather than employing a standard tag-reader system architecture, Savi has introduced a third component, the “signpost”, a fixed or mobile beacon that wakes the tag and notifies it of its location via a short-range, 123 kHz inductive link. The tag stores the wakeup information in its log, facilitating the mapping of environmental records to geographic location. Upon waking, data is transferred to nearby readers using a longer-range, high-frequency channel. Finally, given an alarm condition such as large shocks or high temperatures, the tag is able to wake independently and alert any readers that happen to be in range [55, 54].

1.4.2 Sensitech: ColdStream and TempTaleRF

Sensitech is a company that develops solutions for the cold chain (refrigerated transport) that has recently produced a series of radio-enabled monitoring tags under the brand name

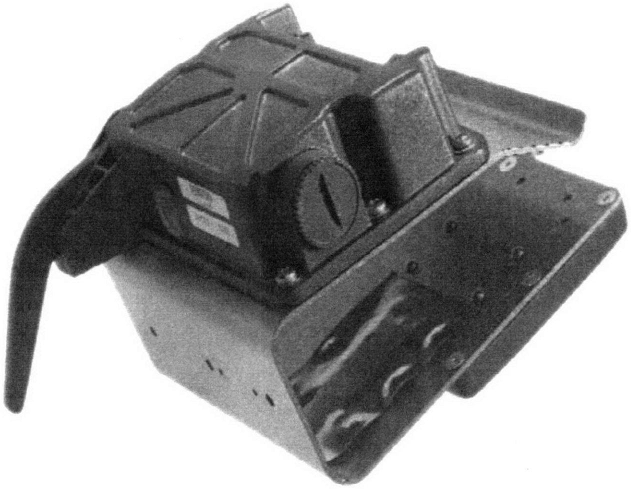


Figure 1.1: The Savi Technology ST-676 tag mounts onto container doors [55].

TempTaleRF. These tags monitor the temperature of a shipment, logging the data or transmitting it to any readers in the vicinity. Like the Savi EchoPoint system described above, the ColdStream infrastructure uses signposts to provide the monitoring tags with location information, though as the system operates on one frequency (either 915 or 866 MHz), the tags apparently wake up periodically and “listen” for the presence of nearby signposts and readers. The tags’ batteries last one year, even though they come equipped with a liquid crystal display of the current temperature, as can be seen in Figure 1.2 [57].

1.4.3 Safefreight Technology: EnCompass and Qualcomm: OmniTracs

Rather than depend on stationary signposts and readers to alert managers of the location of assets, the EnCompass tag from Safefreight Technology uses cellular data networks and GPS satellite positioning to ascertain and communicate its location [63]. This and similar platforms, such as Qualcomm’s OmniTracs [49], are intended primarily for fleet management where instantaneous position is necessary to calculate driver performance and other parameters; such fine detail is unlikely to be necessary at the level of the container, the movements of which are much more constrained.

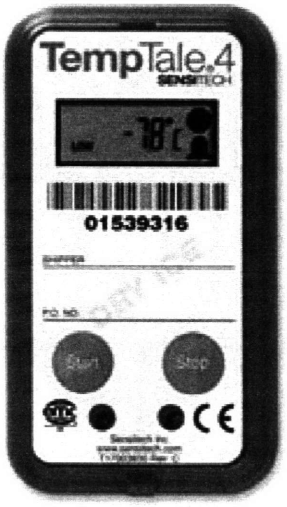


Figure 1.2: The TempTale tag for cold chain monitoring from Sensitech [57].

1.4.4 Motes and Other Wireless Networked Sensors

The decreasing size and price of wireless sensor platforms has been fueling their spread, and supply-chain management is one possible application. Wireless tags are already tracking high-value mobile assets, such as intravenous (IV) pumps or defibrillators in hospitals around North America [43, 61]. The systems offered use a variety of frequencies and protocols—some capable of using pre-existing 802.11b wireless ethernet for location, others requiring an all-new network of readers—but most are only able to convey identification information, without performing any sensing.

Moteiv, a company that implements wireless sensor networks, has recently combined localization and sensing in a location and navigation system for firefighters called FIRE [39]. Each firefighter carries a Tmote Sky wireless sensor node, which uses the ZigBee networking protocol to communicate with similar motes embedded in the building to determine its location and broadcast it to other firefighters and their commanders. Using embedded sensors, the mote is able to provide details regarding the firefighters' surroundings, such as light levels, smoke, and temperature [60].

Motes equipped with a diverse suite of sensors and mesh networking capability are usually too expensive and power-hungry for long-term supply chain applications, especially at the crate and container level. For instance, the Tmote Sky (and the equivalent TelosB mote from Crossbow Technology), which includes an IEEE 802.15.4-compliant[1] transceiver and integrated humidity and temperature sensors, requires 2 AA 1.5 V batteries for power and costs USD 130 in small quantities [41, 10]. The large battery pack and high cost make its use prohibitive for tracking smaller packages.

The same comments apply to the other platforms reviewed in this section: those that were specifically designed to perform supply-chain monitoring are either too bulky, too power-hungry, too expensive, or too specialized. While some can successfully secure and monitor a container, they cannot scale down to the level of the package or crate, where sensor information most closely corresponds to the conditions experienced by an item in transport. A summary of the platforms described in this section, along with several others, appears in Table 1.1.

1.5 Low-Power Strategies for a New System of Active Tags

Overcoming the shortfalls of current commercial platforms described in Section 1.4—such as high cost, large size, and insufficient battery life—requires novel techniques. Some researchers have rejected active RFID and are pursuing the passive sensing strategies described in Section 1.2. One example is the logistics giant DHL, which has recently unveiled passive RFID temperature sensors for use with such sensitive cargo as pharmaceuticals [62]. But not all active RFID approaches have been exhausted, and recent work on low-cost and low-power sensing at the MIT Media Laboratory's Responsive Environments group has led to the development of several strategies that are applicable to the problem at hand.

One related project required the distribution of hundreds of coarse movement sensors for crowd interaction at a large event. In order to make the platform economical (under USD 1 in large quantities), amplifiers and other active components were eliminated, and the output from a piezoelectric-film vibration sensor was used to directly drive a one-shot multivibrator, which then sent a $50\,\mu$s pulse via the on-board, single-transistor 433 MHz radio. This "quasi-passive" sensing method, whereby the energy of the stimulus is used to trigger the rest of the system, is not only less expensive, but much more power efficient. Thanks also to the platform's low-leakage CMOS logic, the single on-board lithium button

[1] 802.15.4 is the standard defining the physical and network layers used in the ZigBee low-power, low-data rate mesh networking protocol.

Platform Name	Manufacturer	Primary Application	Sensors	Communication Frequencies	Battery Life
SensorTag ST-676	Savi Technology	Container monitoring	Shock Humidity Temperature Breach	123 kHz 433 MHz	4 years (2 collections per day)
TempTaleRF	Sensitech	Cold chain monitoring	Temperature	915/866 MHz	1 year
ActiveTag	Axcess Inc.	Asset and personnel tracking	Temperature Tamper	126 kHz 315 MHz	1–3 years
EnCompass	Safefreight Technology	Fleet management	—	GPS/GSM/ CDMA	N/A

Table 1.1: Characteristics of popular active RFID platforms for asset and supply-chain management.

cell has already lasted for 4 years with light use (about 10 transmissions a day), and is expected to last for at least 5 to 10 years (i.e. the shelf life of the battery) [18, 19].

This same quasi-passive wakeup strategy, whereby the energy of a stimulus wakes up a dormant system without any amplification, was elaborated in the FindIT Flashlight project, a system of active tags that respond to modulated infrared (IR) radiation from a handheld flashlight/reader. The power of the interrogation signal, although not enough to supply the tag, was able to activate it from sleep, by tripping a nanopower comparator. The modulated signal was differentiated from background noise by a passive filter. Again, the lack of a linear amplifier at the signal input led to a current consumption of less than $2\,\mu$A and a projected battery life of seven years [32, 2, 33].

Researchers at Northwestern University have also used quasi-passive wakeup (under the moniker "lucid dreaming") for the more pertinent application of vibration detection for autonomous crack monitoring. Using a geophone as the input sensor, their platform wakes up and records aperiodic shocks to ensure structural integrity of buildings. Although the analog front end consumes only $16.5\,\mu$W on average, the processing and communication is performed by a Mica2 mote [8], which adds an additional $105\,\mu$W to the power budget [22]. Clearly, low-power processing must go hand-in-hand with low-power sensing, and that will be a focus of the work at hand.

The FindIT optical tags and the Lucid Dreaming acoustic monitoring platform use only one input modality for wakeup. There is no reason, however, why the system architecture could not be extended to include more micropower sensors. All that is necessary is that the environmental, human, or other stimuli of interest are powerful enough (vis-a-vis the background) to wake the system via interrupt, either by directly driving a microcontroller I/O pin above the logic threshold, by tripping a nanopower comparator [47], or by tripping the comparator after amplification by a micropower op-amp [2].

Chapter 2

Hardware Description

Although all the commercial platforms presented in Section 1.4 are intended for asset tracking and monitoring, none are suitable for use at the case and crate level. There are a number of barriers preventing their application in this space, such as high cost, large size, and short battery life[1].

To overcome these barriers, low-cost components must be used. Although markup constitutes a large portion of the cost of the commercial platforms (which cost up to USD 130 apiece, in the case of a sensor-equipped TelosB/Tmote Sky mote [41, 10]), the cost of the parts is also high. Table 2.1 includes a partial bill of materials for the Crossbow Technology TelosB mote with on-board sensors, including prices for small quantities (around 25 pieces), as listed in popular mail-order electronic components catalogs, such as Digikey, Newark, and Mouser, in January 2007. With only a fraction of the components included in the table, the parts cost already exceeds USD 40.

Part of the problem with the TelosB or Tmote Sky motes is their generality of purpose. In order to satisfy a wide range of applications, they are equipped with a powerful processor (the most powerful in Texas Instruments' low-power MSP430F1*XXX* series [66]), an RF transceiver that supports the IEEE 802.15.4-standard physical layer for mesh networking, and factory-calibrated temperature and humidity sensors. A sensor platform intended specifically for supply-chain monitoring applications can sacrifice speed and precision to achieve lower cost, and that tradeoff will be apparent throughout the chapter.

The other two barriers to the use of current commercial platforms in supply-chain monitoring—large size and short battery life—can be condensed into one. Since batter-

[1] Although Table 1.1 lists up to four years of battery life for some of the platforms, those figures assume extremely low sample rates [57] that may miss crucial changes in the state of the tagged object.

Component	Part Number	Price at 25 Units [USD]
Microcontroller	MSP430F1611	12.98
Temperature/Humidity Sensor	SHT11	18.62
RF Transceiver	CC2400	5.92
8 Mbit Flash Memory	M25P80	4.15
Batteries	2×AA	1.10
32,768 kHz Crystal	CMR200TB	0.62
6 MHz Crystal	CSTCR	0.45
⋮	⋮	⋮
Total		**> 40.00**

Table 2.1: A partial bill of materials for the Crossbow TelosB mote, including environmental sensors [9, 14, 42].

ies constitute the largest portion of a wireless sensor node's volume, attempts to reduce the tag's size will invariably lead to shorter battery life. Overcoming this tradeoff requires low-power components, circuits and operation. The first two—components that are not only cheap but also miserly, and appropriate interface circuits—will be described in the remainder of this chapter, while low-power operation will be presented in Chapter 3.

2.1 Project History

The hardware description that follows pertains to the fourth revision of the CargoNet active tags, a "patch" meant to correct minor errors discovered in CargoNet version 3. That revision was designed between July and October of 2006, following trials of the system at Intel Corporation facilities in Chandler, Arizona, and consultations with Intel staff involved in the Intelligent Container Project.

The initial inspiration for the project came from Mary Murphy-Hoye of Intel and Julius Akinyemi of Pepsico who saw supply-chain management and asset tracking as good applications of quasi-passive wakeup and other low-power sensing strategies under development at the MIT Media Laboratory's Responsive Environments group. The first revision of the active tag platform, modeled after some of the platforms in Section 1.4, was completed in the Spring of 2007, and included the following sensors and modules:

- Real-time clock,
- shock sensor,
- light sensor,

- reed switch (magnetic breach sensor),
- humidity and temperature sensor,
- CC2500 radio transceiver daughter board.

The experiences during Summer 2006 led to the specification and implementation of a number of new sensors, including:

- an RF wakeup/detector circuit,
- a low-cost humidity sensor,
- a vibration dosimeter for measuring the sum of small vibrations,
- a tilt switch,
- a piezoelectric microphone,
- the addition of dynamic thresholds (described in Section 2.4.2) to some sensor interface circuits.

To accommodate the additional components and traces, the microcontroller had to be replaced with a model with additional I/O pins, and the board was upgraded from two layers to four. The full schematic and printed-circuit-board layouts are included in Appendix A. Photographs of the different revisions detailing the evolution of the project can be seen in Figure 2.1.

2.2 System Overview

The CargoNet system of active tags for supply-chain monitoring is based on an architecture similar to that of many commercial systems, illustrated in block-diagram form in Figure 2.2. Active tags remain in a sleep state, monitoring environmental conditions inside a shipping crate or case. Exceptional conditions trigger an alarm that is broadcast to other tags and any readers that may be in range; the event is also logged. A reader can download logged events from the tags by first interrogating them (thereby waking them up) and then receiving the log via a high-frequency radio link.

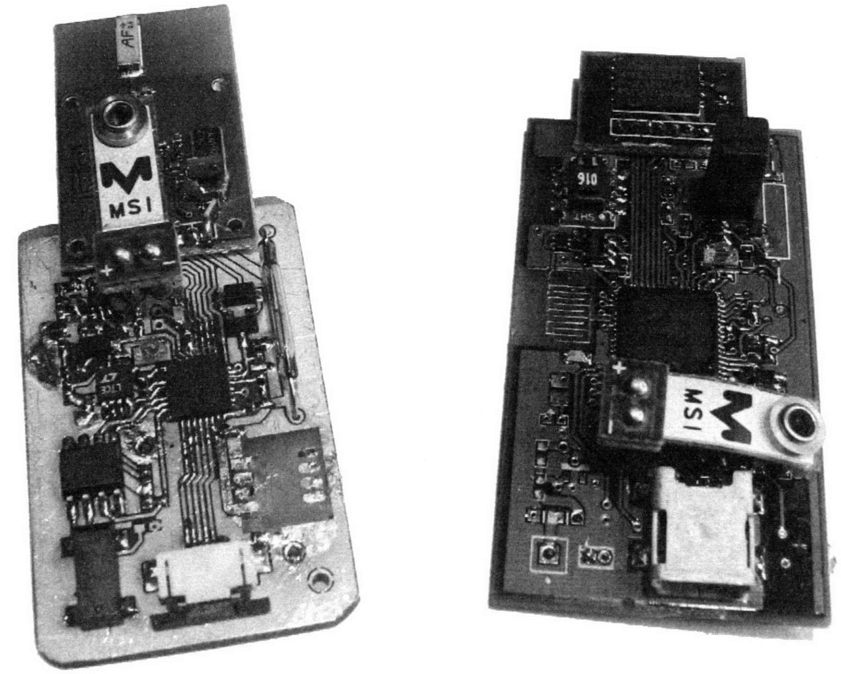


Figure 2.1: Photographs of the CargoNet active tag versions 2 (left) and 4 (right) show an increase in the number of sensors and associated complexity.

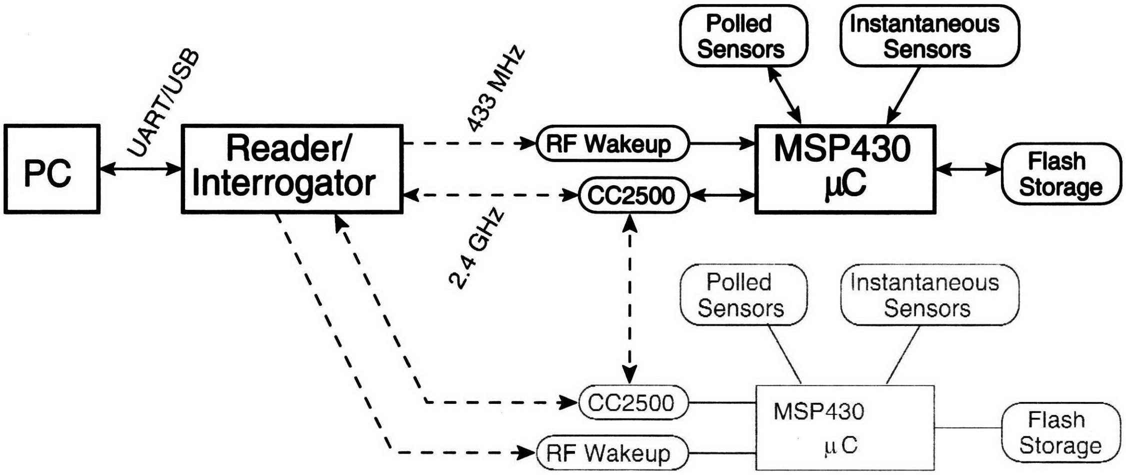


Figure 2.2: Diagram of the CargoNet system. The focus of this thesis is on the tag component to the right of the figure.

2.3 Core Components

The core components of the CargoNet active tag platform have been adapted from other successful platforms and represent a basic system necessary to perform low-power wireless sensing.

2.3.1 MSP430 Microcontroller

Although Texas Instruments' MSP430-series microcontrollers are not cheap (small-quantity prices run around USD 5 for models of intermediate capability [14]), nor particularly new, they remain the microcontroller of choice for ultra-low-power applications. The lowest power mode of operation (LPM4) provides RAM retention and I/O pin interrupt capability while consuming down to $0.1\,\mu$A of current and waking up into active mode operation within $6\,\mu$s [65].

Additional beneficial features of the MSP430-series of microcontrollers are an integrated digitally-controlled oscillator (DCO) and self-writable flash memory. The DCO is responsible for the fast startup from low-power modes; whereas a crystal oscillator may require milliseconds to stabilize [67], the DCO starts up in 6 microseconds, and is capable of driving the main processor clock at up to 5.4 MHz. Although the initial offset (actual frequency is ±25% from nominal) and drift with temperature (up to $-0.43\%/\,^\circ$C) and supply voltage (up to $10\%/\,$V) are atrocious [65], great accuracy is unnecessary as the oscillator controls sampling rates and timeout timers, which need only be loosely specified.

The flash-based MSP430 microcontrollers (part numbers MSP430F*XXXX*) also have an on-chip charge pump capable of generating the higher flash programming voltage, allowing the microcontroller to write to its internal flash during operation [67]. Any flash left over after code programming can therefore be used to log data, saving the board space, cost, and power that would otherwise need to be spent on an external flash memory.

Of the MSP430-series, the MSP430F1232 was initially chosen as the smallest and least expensive member of the series that included an analog-to-digital converter (ADC) and serial peripheral interface (SPI) module for high-data-rate communication with radio, external memory, and other peripherals [64]. As the system evolved, and more sensors were added to the tag, the microcontroller was upgraded to the MSP430F135. The larger package provided more I/O pins as well as twice the flash memory for data storage [65]. A table summarizing pin assignments is included at the end of Section 2.4.

2.3.2 CC2500 RF Transceiver

The active tags need to wirelessly communicate the state of objects inside a shipping crate or case to the outside world, and therefore require a radio transmitter. To make communications more robust, and enable wireless commissioning, re-tasking, and "marking" (whereby the current location or identity of the handler is recorded in the tag's log), the radio must be a fully bidirectional transceiver. The 2.4 GHz CC2500 transceiver from Chipcon is ideal for this application due to its low cost and high data rate: the IC costs USD 3.15 at low quantities [14] and can transmit at rates up to 500 kbps [6]. High data rates diminish average power consumption, and they usually also limit the range of transmission, but the CC2500 has been tested over a range of 100 m outdoors at a rate of 250 kbps. Other distinguishing features of the CC2500 are its easy interface (which simplifies the circuit, thereby reducing cost) and hardware packet handling [6].

The CC2500 is part of a recent wave of "digital radios" [27] that combine a high-data-rate RF transceiver with a state machine for packet handling and manipulation. This extra hardware frees the microcontroller from performing tasks like encoding/decoding, calculating checksums, and filtering packets based on length or destination address, all of which the dedicated hardware can perform more efficiently. The microcontroller is still able to maintain control over these functions, however, through a series of register settings, accessible via SPI.

The digital radios offered by companies such as Chipcon and Nordic Semiconductor differ from each other by the frequencies and modulation schemes they employ, and also by how many and what kind of additional network stack layers they include on-chip. For example, the CC2420 radio from Chipcon implements not only the physical layer (implemented by all radios) and data-link layer (packet integrity), but also medium-access control (MAC) compatible with the ZigBee standard for low-power, low data rate communications between wireless devices [7]. Because the envisioned application of supply-chain monitoring does not require full multi-hop mesh networking between tags, radios with elaborate built-in MAC layers—like the CC2400 present on the TelosB platform [9]—are unnecessary.

Some tag-to-tag communication is desirable, however, to synchronize clocks and confirm shared observations, and the CC2500 stands out as a compromise between simplicity and ability. While it is small (20-pin QFN package) and inexpensive (USD 3.15 in quantity of 25) [14], it is capable of supporting full multi-hop mesh networking, if need be [28]. The

CC2500 communicates with the microcontroller via SPI; several of the CC2500's general purpose I/O pins are also connected to the microcontroller's interrupt pins to alert it once a packet has been successfully transmitted or received.

Because the MSP430F122 and MSP430F135 microcontrollers have only one SPI port, the CC2500 resides on a daughter board, which can be removed and replaced with another SPI peripheral (such as additional flash memory) during project development and testing. The radio can be more fully integrated onto the active tag circuit board in a future revision, saving additional space and cost. A schematic of the CC2500 daughter board, developed by Responsive Environments Group Research Assistant Mathew Laibowitz, and its photograph are included in Figures 2.4 and 2.3.

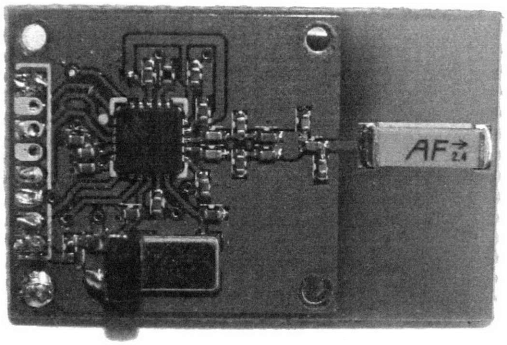


Figure 2.3: This CC2500 daughter board allows the radio to be swapped in or out during prototyping.

2.3.3 Power

The CC2500 radio and a MSP430-series microcontroller can operate over the 1.8 V to 3.6 V supply voltage range, but the microcontroller requires at least 2.7 V for flash memory programming. The goal of the power subsystem, therefore, is to maintain the supply voltage between 2.7 V and 3.6 V for the longest possible time, in a small form factor, and at a reasonable price.

In the TelosB mote, two standard AA-sized alkaline-zinc/manganese dioxide (alkaline) batteries provide the nominal 3 V necessary for operation. As can be seen in Figure 2.5, however, the discharge curve of alkaline batteries is much steeper than that of competing chemistries, and two of them are only able to support a voltage above 2.7 V over the first 20% of their utilization [21]. One solution to this problem uses additional alkaline cells to

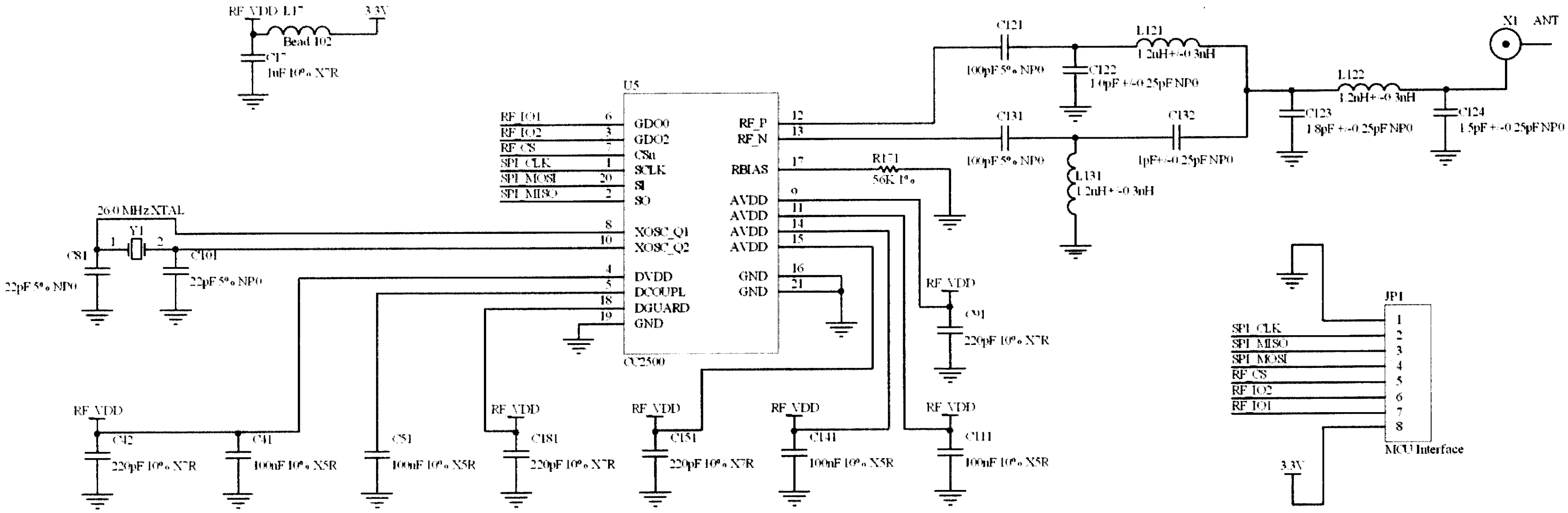


Figure 2.4: Schematic of CC2500 daughter board designed by Mathew Laibowitz.

achieve a higher supply voltage, which is then brought down to the nominal 3 V using a regulator. Linear series regulators, however, waste power in the form of quiescent supply current and dropout voltage, and although switching regulators can be highly efficient, they are expensive, making a truly low-power regulator a costly proposition. The space required by the additional alkaline cells is a further concern.

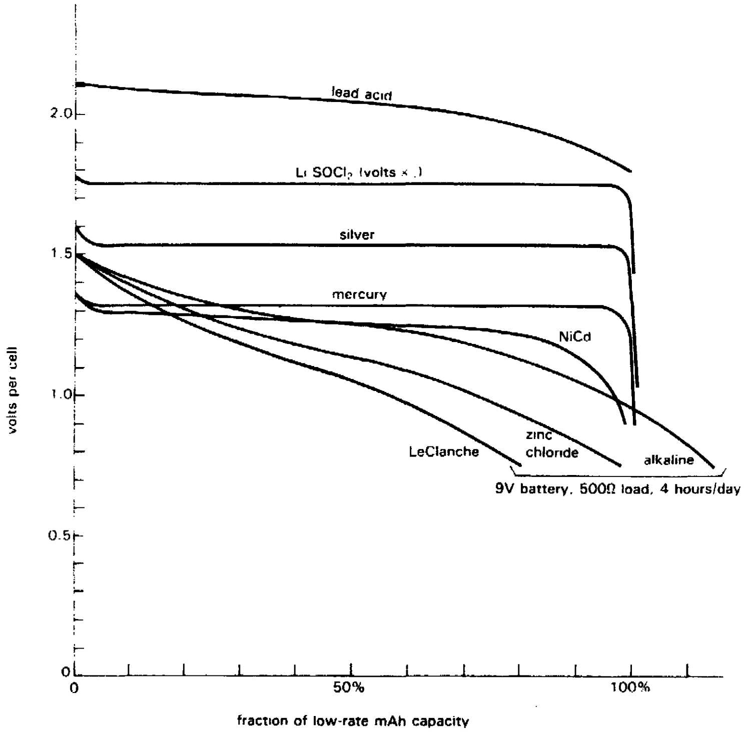


Figure 2.5: Alkaline batteries exhibit a much steeper discharge curve than many other competing chemistries, making them unsuitable for unregulated applications [21].

In contrast to alkaline cells, lithium batteries offer a nominal 3.3 V from a single cell and exhibit a much flatter discharge curve [44], with the voltage starting to fall only after the battery has been 90% discharged. Recent advances in lithium technology have brought to the market high-density *lithium-ion* batteries used in laptop computers and cellular telephones, and small, lightweight *lithium-polymer* prismatic batteries commonly used in model airplanes. Although both of these battery chemistries provide high energy density, they are secondary (rechargeable) chemistries. Since one of the key design goals of this project is to make a low-maintenance platform where the battery life approaches the device lifetime, obviating battery replacement, rechargeable batteries are of little benefit and not worth their substantially higher costs.

A cost-sensitive application such as supply-chain monitoring therefore requires that one of the less glamorous but time-tested solid-state lithium chemistries be used: lithium poly-carbonmonofluoride and lithium manganese dioxide. Both chemistries are available in small coin- and button-shaped cells, but the former has a much wider temperature range, operating from −30 °C to +80 °C, versus −30 °C to +60 °C for the latter [45]. For comparison, the temperature inside a sealed shipping container can easily reach 50 °C on a hot day. Unfortunately, the lithium poly-carbonmonofluoride batteries, as exemplified by the BR-series coin cells from Panasonic, do not provide the high instantaneous currents necessary to power the radio transceiver (up to 20 mA) [44]. Even when using the higher-current CR series, a large charge-storage capacitor must be placed in close proximity to the radio, as the battery cannot source the current required.

2.4 Sensors and Other Components

In addition to low cost and low power consumption, the additional feature lacking in the platforms described in Section 1.4 was an adequate suite of sensors. Even the Savi ST-676, which measures several pertinent environmental parameters [55], does not implement a tilt sensor, which is important when dealing with cargo in units smaller than the container. The lack of a comprehensive sensor suite requires the design of a new one, tailored specifically to the demands of crate- and case-level cargo monitoring, with the twin demands of low cost and low power consumption.

2.4.1 Quasi-Passive Wakeup

The quasi-passive wakeup strategy developed during the course of the FindIT Flashlight project allows for the monitoring of certain environmental parameters without resorting to linear amplification or polling [32]. Periodic polling is well suited to slowly-varying environmental phenomena, such as temperature or humidity, and usually involves transduction, linear amplification, analog-to-digital conversion, and digital manipulation. For signals that change on the time scale of minutes, this can be an extremely low-duty-cycle operation: for example, an SHT11 polling sequence with 12 bits of accuracy lasts 55 ms, during which the sensor draws 550 μA, resulting in a 0.092% duty cycle and 1.5 μW average power consumption. If the sensor were sampled at 10 Hz, that same collection process would require a duty cycle of 55% and consume 900 μW of power, which would drain a 235 mAh lithium coin cell in

$$235\,\mathrm{mAh} \cdot \frac{1\,\mathrm{day}}{24\,\mathrm{h}} \cdot \frac{1}{55\% \cdot 550\,\mu\mathrm{A}} = 32.4\,\mathrm{days}. \tag{2.1}$$

In the realm of active tagging, high-frequency stimuli are often fleeting so do not need to be sampled continuously; instead, they must be detected and analyzed only when they suddenly occur. The quasi-passive wakeup approach uses a passive (or micropower) transducer to convert the stimulus to an electrical signal, followed by a passive RLC filter to separate the high-frequency signal of interest from the background, followed finally by a nanopower comparator for thresholding [32]. The comparator is able to boost the usually weak signal from the sensor to logic levels very efficiently because it acts as a class-D amplifier, consuming power mainly during the state transition[2]. The output of the comparator is in turn connected to a microcontroller pin with interrupt capability, which restarts the microcontroller to begin processing. This basic architecture, with some elaborations, is presented in Figure 2.6.

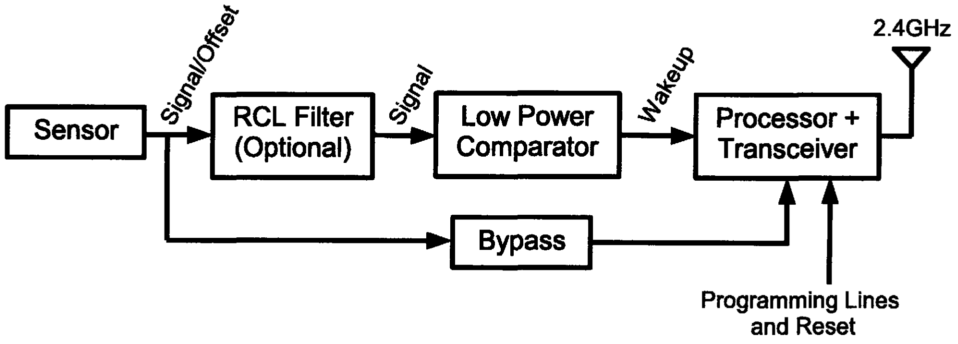


Figure 2.6: Block diagram of quasi-passive wakeup scheme used in the CargoNet system [47].

It is important to distinguish "wakeup" circuits on many sensor platforms from the quasi-passive wakeup described above. The Tmote Invent prototyping platform [40], for example, has a comparator that wakes up the microcontroller via interrupt when the accelerometer or microphone outputs surpass a certain level. Although the microcontroller could be put to sleep while awaiting stimuli, there is little power-savings advantage as the sensors themselves are active. The accelerometer sensor and circuitry draws 490 μA, while

[2] The LTC1540 nanopower comparator used in the CargoNet and FindIT flashlight projects typically requires a supply current of only 0.28 μA [29]

the (electret) microphone and circuitry draws 2.3 mA! For comparison, an MSP430F135 microcontroller draws 420 μA when running at 1 MHz from a 3 V supply.

2.4.2 Dynamic Thresholds

The quasi-passive wakeup approach described in the previous section has been implemented in the CargoNet tags to alert the microcontroller to large shocks that may damage the shipment. In theory, the presence of shocks should be easy to differentiate from their absence, but the residual vibrations from a large jolt can persist for over a second. During this time, the microcontroller consumes power in vain, as no new information is imparted by these secondary stimuli. They are part of the same event and are of significantly less interest than the initial shock. Similarly, ships, trains, and trucks all vibrate, and if the noise threshold has been incorrectly set, the vibration sensor will continually wake the tag, eventually discharging its battery and preventing it from catching important events that may happen later.

In order to prevent these false positives, the microcontroller could institute a timeout period by disabling the interrupt on the affected pin for one, two, four, etc., minutes, much the same way that Ethernet cards avoid collisions through exponential backoff. In the bumpy truck scenario presented above, however, this scheme would have the effect of completely desensitizing the tag to all shocks, big or small, even though the battery would stay charged and other sensors could function normally.

Ignoring small vibrations while maintaining the ability to respond to large shocks requires that the tag be *gradually* desensitized to the stimulus by dynamically increasing the threshold of its low-power comparator. Exponential decreases in sensitivity upon successive wakeups provide a quick immunity to repeated shocks. The sensitivity is gradually restored over time once the shock has passed.

The implementation particulars are described in the sensor-specific sections that follow, but in two of the three cases, a digitally-controlled potentiometer provides a variable load to a high-impedance sensor (piezoelectric microphone or shock sensor), attenuating the signal before comparison with a fixed threshold. This is equivalent to changing an actual threshold while keeping the signal attenuation constant, as has been done in several other other research and commercial platforms [22, 40]. The literature describing the operation of those platforms, however, assumes that the thresholds will be fixed at the beginning of the mission, and does not mention varying the thresholds dynamically, in response to incoming stimuli. Such a strategy will allow the tag to adapt to varying conditions, decreasing power consumption while logging a constant amount of pertinent information. The firmware that implements this dynamic threshold strategy with various sensors is described in Section 3.2.2.

2.4.3 Real-Time Clock

Inferring that a shipment may have been damaged due to shock, tilt, or extremes of temperature and humidity can already be accomplished by using stick-on mechanical or chemical tags that change color when an exceptional event occurs. One crucial advantage of active RFID is that it can also record *where* along the supply chain the damage has occurred, which permits easier identification of the responsible party or for future prevention. However, because it is too expensive and power-intensive to monitor the location of each crate or case by placing a GPS or cellular telephone receiver on it, its current location must be inferred indirectly by keeping track of the time elapsed from a known locale.

The MSP430 contains an on-chip oscillator that can be used with a low-frequency, 32,768 kHz watch crystal [67]. This is the lowest-part-count solution and, together with the appropriate firmware, could serve as the platform's real-time clock for a mere USD 0.60, the small-quantity cost of a crystal sourced from Digikey [14]. To use the low-frequency oscillator, however, the microcontroller must leave the sleep state (LPM4) for low-power mode 3 (LPM3). In this state, it typically consumes $1.6\,\mu$A at room temperature [65], an order of magnitude above its $0.1\,\mu$A LPM4 current.

Fortunately, the proliferation of digital watches, cellular telephones and other timekeeping devices has made available a host of real-time clock circuits that are both inexpensive and consume very little power. A partial summary of available parts and their characteristics appears in Table 2.2; of these, the PCF8563-04 from Phillips was selected for its extremely low current consumption ($0.25\,\mu$A typical), and low price.

Because certain environmental stimuli, such as changes in temperature or humidity, occur on a large time scale, they cannot be differentiated from background noise and drive a quasi-passive wakeup circuit of the kind described in Section 2.4.1. Instead, these stimuli must be sampled periodically utilizing sensors with very low duty cycles. Activation of these sensors, in turn, requires that the microcontroller be woken by a real-time clock interrupt.

Part Number	Manufacturer	Typ. Supply Current [μA]	Package	Communication Protocol	Notes	Price at 25 Units [USD]
Use μC as RTC	—	1.6	—	—	—	0.00
RX-4581	Epson	0.4	SON-22	4-pin SPI	16-byte RAM	??
DS1302	Maxim-Dallas	0.3	8-SOIC	3-pin serial	No interrupts!	2.63
DS13-5	Maxim-Dallas	> 0.3	20-TSSOP	4-pin SPI		4.02
MAX6902	Maxim-Dallas	> 0.4	8-TDFN	4-pin SPI	No interrupts!	4.04
NJU6355	New Japan Radio	$> 3.$	8-DMP	4-pin serial	No interrupts!	1.07
PCF8563	Philips	0.25	SO8	I^2C	Alarms and countdown timers	1.06

Table 2.2: Partial overview of available real-time clock integrated circuits.

Extensive interrupt capabilities are another reason why the PCF8563 was chosen over the other clock chips: alarms can be generated at a particular minute, hour, day of the week, or day of the month, or through an eight-bit countdown timer.

2.4.4 Humidity and Temperature Sensors

Temperature and humidity are important parameters to monitor as they can vary widely over the course of a journey. The natural diurnal variations can be exacerbated by poorly insulated cargo holds on airplanes or leaky maritime containers, making monitoring a must.

SHT11

The Sensirion SHT11 was used as the sole humidity and temperature sensor in the early versions of the CargoNet active tag. Unlike many of the offerings currently on the market, the SHT11 is not just a bare humidity sensor, but an integrated environmental monitoring subsystem that contains temperature and humidity sensors, a 14-bit ADC, a serial communication interface, a ROM, and all drive and control circuitry. The SHT11 simplifies the task of the system designer (no need for separate drive circuitry or ADC) and also provides higher accuracy, as the unit comes factory calibrated, with the calibration coefficients stored in the on-chip memory, a scheme that guarantees 3.5% accuracy between 20 and 80% relative humidity (RH), 0.03% RH resolution, and full interchangeability [56].

Low-cost Humidity Sensor

Unfortunately, the SHT11 is the single most expensive component in the CargoNet bill of materials, costing USD 18.62 in quantities of 25 [42], and singlehandedly defeating the goal of assembling a platform with a sub-USD 20 parts cost. The less expensive humidity sensors are no bargain either, since the cost of the necessary external drive circuitry consumes much of the savings. The low-cost Senser-HUM33 resistive humidity sensor from Erlich Industrial Development Corporation provides some inspiration, however. Resistive humidity sensors work by measuring the resistance across a hydrophilic polymer or ceramic; the HUM33 however, seems to use the bare surface of a PCB instead of polymer, and two interdigitated traces as the electrodes [16], as can be seen in Figure 2.7. By etching this design directly onto the PCB, it should be possible to achieve a *free* humidity sensor. A photograph of the sensor integrated into the CargoNet revision 4 PCB can be seen in Figure 2.8.

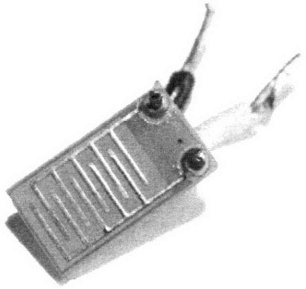


Figure 2.7: The Senser-HUM33 consists of nothing more than traces on a PCB.

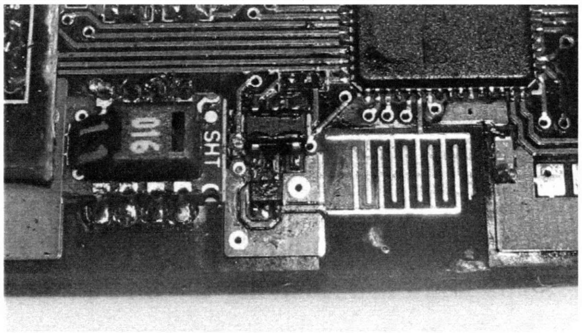


Figure 2.8: Detail of a photograph of the CargoNet version 4 PCB shows the low-cost resistive humidity sensor (right) alongside the SHT11.

Resistive humidity sensors must be driven with an alternating signal with zero bias, as polarization may damage the sensor. Furthermore, the resistance of the sensor is not linearly proportional to relative humidity (see Figure 2.9), and can range from less than $1\,\mathrm{k}\Omega$ to over $10\,\mathrm{M}\Omega$ [53], far surpassing the dynamic range of a common ADC. An interface circuit for the sensor must therefore address these issues, in addition to the requirements of low cost and low power consumption.

The circuit pictured in Figure 2.10 meets the humidity sensor requirements of AC drive, zero bias, and low cost. Low power consumption is automatically assured through low-duty-cycle polled operation; a microcontroller I/O pin sources the current to the amplifier and voltage divider. At the beginning of the polling cycle, the microcontroller powers on the sensor, while the drive pin oscillates between 0 and 3 V. As long as this driving square wave is symmetric (50% duty cycle), the average voltage at the left side of the humidity sensor will be 1.5 V. At the same time, the negative feedback of the op-amp will set $V_- = V_+$, elevating the voltage at the right side of the sensor to 1.5 V as well, resulting in zero bias across the sensor. Because of the high input impedance of the op-amp (a FET-input op-amp

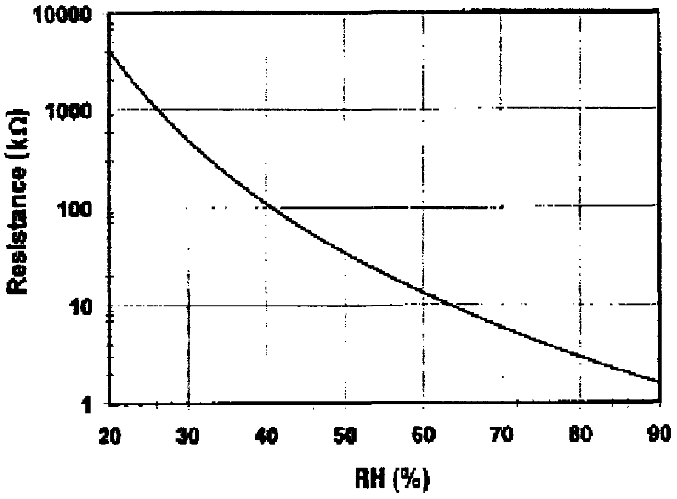


Figure 2.9: The logarithmic response of a resistive humidity sensor requires a linearizing circuit [53].

must be used), the current through the sensor $I = \frac{V_{DRIVE} - V_{-}}{R_{SENSE}}$ passes through one of the two diodes—depending on the polarity of the drive pin with respect to 1.5 V—and produces a logarithmic voltage drop at the output of the op-amp, linearizing the sensor resistance. As can be seen in Figure 2.11, for a decade increase in resistance, the peak-to-peak amplifier output voltage decreases by approximately 100 mV.

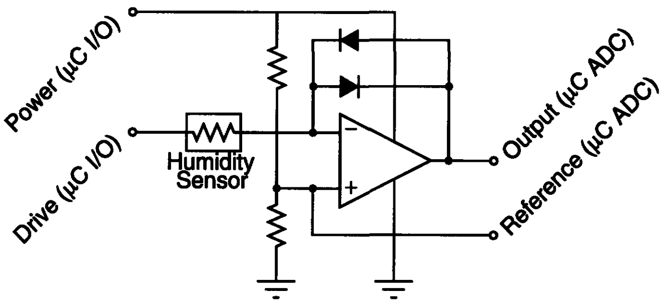


Figure 2.10: This linearization circuit should make humidity sensing affordable.

Although the accuracy of the low-cost humidity sensor has yet to be tested, the manufacturer of the sensor upon which this one was based claims 2.7% accuracy between 3 and 97% RH [16]. Assuming that the dynamic range of the sensor output amplitude extends from 0 to 300 mV, and the 12-bit ADC uses a dedicated mid-supply reference during conversion,

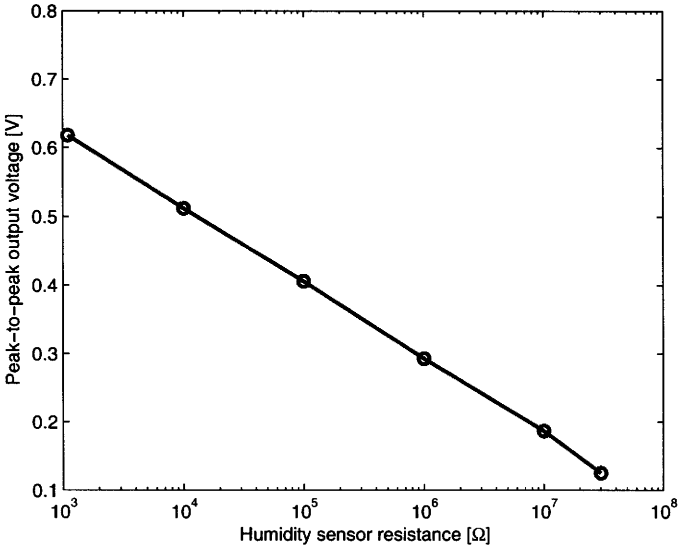


Figure 2.11: The peak-to-peak output voltage (centered around 1.5 V) of the logarithmic converter of Figure 2.10 linearizes the low-cost humidity sensor.

the sensor could achieve a resolution of

$$R = \frac{100\%\mathrm{RH}}{\frac{0.3\mathrm{V}}{1.5\mathrm{V}} \cdot 4096} = 0.12\%\mathrm{RH}. \tag{2.2}$$

In addition, like all resistive humidity sensors, this one should be fully interchangeable [53], requiring only one calibration that will be valid across all units.

Both resistive and capacitive humidity sensors exhibit a strong temperature dependence, an effect that can be corrected by simultaneously reading a nearby temperature sensor. The SHT11 humidity sensor, for example, has its own temperature sensor on-chip—a major strength. But even if the ambient temperature can be measured, the accuracy of the humidity reading depends on how easy it is to model and correct the sensor's temperature dependence. Greater accuracy requires more complex models, which in turn require more time and power to compute. Moving beyond this power/accuracy tradeoff therefore requires a sensor design with a simple temperature dependence.

The temperature dependence of a typical, low-cost resistive humidity sensor (Sencera K25K5A) is shown in Figure 2.12. Assuming that the HUM33 adaptation used on the CargoNet tags exhibits a similar response, as the resistance decreases exponentially with temperature the output of the circuit in Figure 2.10 will increase linearly. A 50 °C increase in temperature, which tends to lower the resistance of the sensor by a factor of 10, will increase the linearizing circuit output by 100 mV as shown in Figure 2.11.

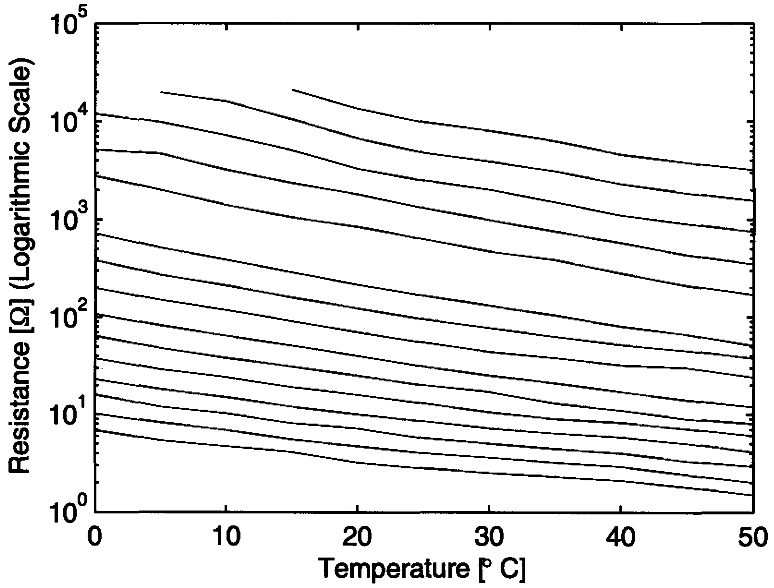


Figure 2.12: Resistive humidity sensors exhibit exponentially-decreasing resistance with temperature.

The diodes that set the gain in the linearizing circuit, however, exhibit their own temperature dependence. Fortunately, their temperature dependence will tend to cancel the effect just described. By starting with the diode equation [21],

$$I = I_S\left(e^{V\cdot\frac{q}{mkT}} - 1\right) \tag{2.3}$$

which establishes the relationship between voltage and current at a p-n junction, then solving for the voltage V, and differentiating with respect to temperature, one can show that given constant current, the diode voltage decreases with temperature at a rate of $\approx -2\,\mathrm{mV}/\,^\circ\mathrm{C}$ [31]. This result is counterintuitive, as increasing the temperature in Equation 2.3 while holding the current constant should result in an *increasing* voltage V. The saturation current I_S, however, also has a temperature dependence, which tends to dominate, resulting in a negative dependence for the voltage.

This $-2\,\mathrm{mV}/\,^\circ\mathrm{C}$ approximation will not hold over a wide range of diode forward voltages and temperatures. Nonetheless it is reassuring to note that at $-2\,\mathrm{mV}/\,^\circ\mathrm{C}$, the temperature dependence of the diodes will cancel the temperature dependence of the sensor almost exactly, as the diodes will decrease the output voltage by 100 mV over 50 °C at the same time as the humidity sensor attempts to increase it by 100 mV.

To obtain the temperature dependence at a wider range of temperatures and voltages, the following equation can be used [21, 31]:

$$\left.\frac{\partial V}{\partial T}\right|_I = \frac{V - V_{GO}}{T} - \frac{\gamma m k}{q}, \tag{2.4}$$

where V_{GO} is the bandgap of silicon, γ is a manufacturing parameter, m is a diode modeling factor between 1 and 2, and k and q are Boltzmann's constant and electron charge, respectively.

Internal Temperature Sensor

The low-cost humidity sensor is essentially free because all the other components of the SHT11 system are already in place: additional traces on the PCB can be included at no cost; resistive humidity sensors are interchangeable, which obviates the need for expensive calibration; and the microcontroller provides the drive, analog-to-digital conversion, and an on-chip temperature reference for temperature compensation of the humidity reading. This reference can also be used in place of the SHT11's integrated temperature sensor, resulting in a temperature and humidity measurement package that costs only USD 0.81 versus 18.62 for the SHT11. The SHT11 was, however, included in the final revision of the CargoNet tag to evaluate how well the low-cost components performed against the calibrated sensor. The results of these tests are discussed in Sections 4.2.1 and 4.1.8.

2.4.5 Breach Sensors

The CargoNet tag has two sensors to detect breach: a light meter and a magnetic reed switch.

Phototransistor Light Meter

As any person inspecting or pilfering from a box or crate will most likely need light to see what is inside, a log of the light meter condition will provide a good indication of the integrity of the shipment. Because a breach can happen at any time, the light meter cannot be a polled sensor; rather, it must follow the quasi-passive strategy developed previously in Section 2.4.1. Quasi-passive wakeup is simple to implement in the case of a light detector/meter, as can be seen in Figure 2.13.

No collector current flows through the NPN phototransistor in the absence of light, so if the circuit is powered, the output floats to the positive supply. The high load resistance (1 MΩ) was selected such that the minimum amount of light necessary to see in an otherwise dark room would bring the output of the circuit low and request a microcontroller interrupt. The output is also connected to the on-chip ADC, which allows for the microcontroller to quantify the intensity of light falling on the sensor. After the breach event has been logged and the illumination measured, the phototransistor will continue to conduct, so power to the sensor should be turned off to extend battery life.

A dynamic threshold was not implemented in this circuit because the amount of illumination in a sealed box is assumed to be zero, hence there is no "ambient" light level to which the circuit must adjust. If demanded by the application, however, a dynamic threshold could be easily added by inserting a digipot and nanopower comparator at the output of the phototransistor.

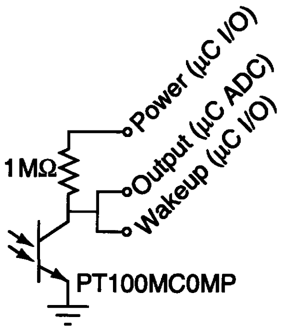


Figure 2.13: Circuit schematic for phototransistor-based light detector/meter.

Magnetic Reed Switch

A magnetic switch provides another method to detect breach. The field from a permanent magnet placed on one of the flaps of a box or the lid of a crate holds the reeds of the switch together, closing the circuit. When the box or container is opened and the magnet moved, the reeds spring apart, opening the switch and waking the microcontroller, which logs the event and can sound an alarm. Because the switch is normally closed, it cannot be connected directly to the microcontroller like the phototransistor; a p-channel FET serves as a second, normally-open switch driven by the first magnetic switch. A reverse-biased diode connected to the gate conducts minimal reverse current when the switch is closed, but acts as a pulldown when the switch opens. The schematic for the circuit can be seen in Figure 2.14.

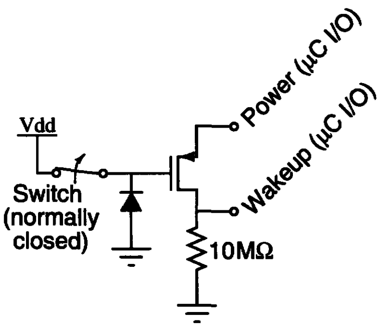


Figure 2.14: Breach sensor circuit schematic. The p-channel FET conducts only when the switch opens, ensuring essentially zero current consumption under ordinary circumstances.

2.4.6 Tilt Sensor

Because of the toxicity of mercury, tilt switches are usually implemented with a ball bearing that moves along a vertical shaft with a set of electrodes on one end. The tilt sensor used on the CargoNet active tags is a one-axis switch (part number 107-1007 from Mountain Switch), mounted on the top layer of the PCB, such that the bearing rests on the electrodes when the tag faces up. This results in a switch that is closed under normal circumstances. A tilt of more than 90° starts the ball rolling down the shaft, opening the switch and alerting the microcontroller. The behavior of this sensor is identical to that of the reed-switch breach sensor, so the circuit in Figure 2.14 was reused for this application.

2.4.7 Shock Sensor

Most inertial measurement applications require an accelerometer to precisely measure shocks. Unfortunately, IC accelerometers are still expensive and consume too much power to be used in common supply-chain applications, especially since the high resolution they provide is unnecessary when answering the question, “Did a shock occur?” In addition, multi-axis sensing is unnecessary since the primary threat to the integrity of a shipment comes from drops. Instead, as in the Ultra-Low-Cost Sensors for Large-Crowd Interaction project [18], a passive, one-axis shock sensor can be used, interfaced such that the energy from the shock alerts the microcontroller without any preamplification.

There are several types of sensors that can act as passive shock sensors, from a simple cantilevered switch to various piezoelectric materials that generate a voltage when deformed by accelerations. Kyocera and Murata have recently begun manufacturing piezoelectric sensors for use in laptop computer hard drives to detect drops and move the write heads away from the disk platters so as not to damage the disk. These sensors (PSAC-series from Kyocera and PKGS from Murata) are only around $3 \times 6.4 \times 1.5$ mm in size, with a resonant frequency above the audible range, and come in various sensitivities and primary-axis alignments [26].

Although the primary application of the CargoNet tag is supply-chain monitoring, the tags could also be repurposed for closely related security purposes: the platform already contains a light detector and magnetic (reed) switch and a radio transmitter to sound the alarm. All that is needed is a vibration sensor to detect motion of the tagged object as it is being stolen. The chip shock sensors described above are too small and therefore not sensitive enough for this security application, so to ensure some measure of generality a more sensitive bendable-piezoelectric-film shock sensor was chosen.

The LDTC MiniSense 100 vibration sensor (a.k.a. vibratab) from Measurement Specialties is a cantilevered strip of piezoelectric film encased in polyurethane with a small mass at the end; shocks set the mass in motion, and the piezoelectric film transduces the vibrations to voltage [36]. Although the sensor's resonant frequency is 75 Hz [35], the sensor has been found to respond to vibrations up to approximately 1 kHz (hence the envelope detector at the output in Figure 2.15). Another possible limitation of the sensor is that the piezo film is constrained to move up and down. But the worst shocks that a shipment is likely to experience—from drops—will be aligned in the vertical direction. Furthermore, shocks along the other axes have been found to couple to the primary axis, making this inexpensive but sensitive sensor a good match for this application.

The output of the sensor passes through a peak detector that strips the envelope from the high-frequency oscillations generated by the shock, reducing the sample-rate requirements that would otherwise have to be placed on the ADC to prevent aliasing when measuring the amplitude of the waveform. Neither the ADC nor the microcontroller is active when the shock first occurs; instead, the filtered waveform is compared against a fixed threshold at the input of an LTC1540 nanopower comparator, the output of which wakes the microcontroller and starts the ADC sampling sequence via an I/O pin interrupt.

Although the actual threshold of the comparator is fixed at 40 mV, the shock sensor wakeup circuit should use dynamic thresholding to decrease the number of unnecessary wakeups, as described in Section 2.4.2. To limit bias currents, high-valued ($\approx 5\,\mathrm{M}\Omega$) resistors are used in resistor dividers to set the comparator thresholds; but since digitally-controlled potentiometers (or digipots) are only manufactured in values up to $200\,\mathrm{k}\Omega$, one cannot directly control the threshold by placing the digipot in the resistor divider. Instead, as can be seen in Figure 2.15, the digipot serves as a load resistor in the envelope detector: the smaller its resistance, the higher is the attenuation of the vibratab signal, and the higher the effective threshold that the vibration must overcome before the comparator trips and wakes the microcontroller.

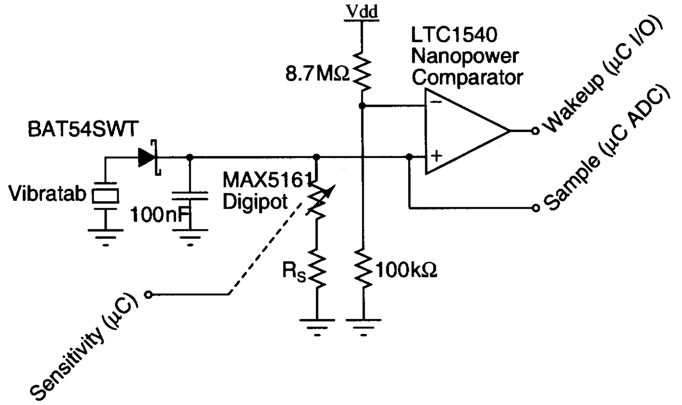


Figure 2.15: The digitally-controllable variable load resistor in the shock detector circuit effectively implements a dynamic threshold for shocks.

There are two parameters that can be set in the circuit: the comparator threshold V_T and the resistance in series R_S with the MAX5161 digipot; these will determine the initial sensitivity of the circuit as well as its ability to filter unwanted shocks up to some maximum value of shock. The initial threshold must be determined empirically from drop tests. Because of its high output impedance, the vibratab sensor can be modeled as a current source, producing an output voltage across a resistive load; the circuit parameters V_T and R_S must be set such that the minimum shock current of interest times the maximum load produces a voltage higher than V_T, while any current less than the maximum conceivable current times the minimum load stays under V_T. Put more concisely:

$$\begin{aligned} I_{MIN} \cdot R_{MAX} &> V_T + V_H && (2.5)\\ I_{MAX} \cdot R_{MIN} &< V_T - V_H, && (2.6)\end{aligned}$$

where $R_{MAX} = R_S + 200\,\mathrm{k\Omega}$, while $R_{MIN} = R_S + 0\,\mathrm{k\Omega} = R_S$, and V_H is the desired "hysteresis" voltage—usually several millivolts—which allows the designer to specify different high-current and low-current thresholds with only one fixed threshold. V_H does not physically exist anywhere in the circuit; rather it influences the derivation of the appropriate V_T and R_S as can be seen in the following equations:

$$\begin{aligned} R_S &= \frac{V_T - V_H}{I_{MAX}} && (2.7)\\ V_T &= I_{MIN} \cdot R_{MAX} - V_H && (2.8)\\ &= I_{MIN}\left(\frac{V_T - V_H}{I_{MAX}} + 200\,\mathrm{k\Omega}\right) - V_H && (2.9)\\ &= \frac{I_{MIN}\left(\frac{-V_H}{I_{MAX}} + 200\,\mathrm{k\Omega}\right) - V_H}{1 - \frac{I_{MIN}}{I_{MAX}}} && (2.10)\\ &= \frac{-V_H\left(1 + \frac{I_{MIN}}{I_{MAX}}\right) + I_{MIN} \cdot 200\,\mathrm{k\Omega}}{1 - \frac{I_{MIN}}{I_{MAX}}}, && (2.11)\end{aligned}$$

where I_{MIN} and I_{MAX} must be determined empirically by performing drop tests and measuring the output voltage across a fixed load. The initial series of CargoNet tests were performed with $R_S = 12\,\mathrm{k\Omega}$ and $V_T = 40\,\mathrm{mV}$.

2.4.8 Vibration Dosimeter

The shock sensor described in the previous section responds to and logs sudden, powerful vibrations, but lesser ones are ignored, as are—when a dynamic threshold is used—subsequent equally powerful vibrations that happen within a short time span after the initial one. All these vibrations have an effect, and even though they will not damage the shipment through sheer impact, they may contribute to the loosening of screws and other mechanical connections.

The so-called "vibration dosimeter" included in CargoNet consists of an active integrator built around a micropower op-amp, with low-leakage reset circuitry [21] and polyethylene capacitor, as can be seen in Figure 2.16. The microcontroller periodically samples and then resets the capacitor voltage to reduce the effects of leakage; additionally, the capacitor is connected to a microcontroller I/O pin with interrupt capability, which is able to wake the microcontroller and request a reading/reset if the voltage suddenly climbs above the microcontroller logic threshold of approximately 1.5 V.

The vibration dosimeter is connected to the same physical sensor as the shock detector above, but due to the reverse-connected Schottky diode at the input, it uses only the negative portion of the sensor's vibration signal, whereas the shock detector uses the positive. Because an active integrator inverts the input, the output will be positive and equal to:

$$v_{OUT} = \frac{1}{RC} \int v_{IN} + C = \frac{1}{10\,\mathrm{k}\Omega \cdot 100nF} \int v_{IN}. \tag{2.12}$$

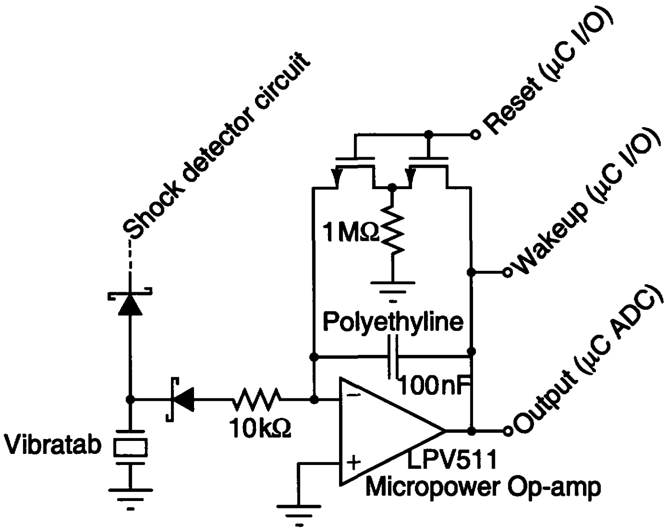


Figure 2.16: The vibration dosimeter integrates the negative swings of the vibratab, keeping track of small vibrations.

2.4.9 RF Detector

Most of the commercial active tags described in Section 1.4 include a high-frequency, long-range radio for data communication with a reader, and a secondary lower-frequency, short-range communication channel for interrogation and passing of location information. Because of the shorter range of the low-frequency link, the RF power delivered to the tag is high enough to wake it up; only then is the high-frequency radio—which consumes up to 20 mA of current in the case of the CC2500, due to amplification and high data rates—powered on.

In contrast to this common division of functions between two radios, it was the author's intention to use an interrogator operating at 2.4 GHz for both interrogation and data communication, by adding a novel detector front end to the on-board CC2500 radio module. A prototype operating at 433 MHz was developed but would not scale up to 2.4 GHz because of insufficient RF fabrication capabilities. At 2.4 GHz, the wavelength is 12.5 cm, and distances on the order of centimeters ($\approx \frac{1}{10}\lambda$) become electrically significant, making design difficult.

The 433 MHz prototype has therefore been laid out on the CargoNet PCB for demonstration purposes and also to help with the development of a CargoNet reader[3] and the testing of interrogation and communication protocols. As can be seen from the schematic in Figure 2.17, the detector circuit consists of an LC tank with autotransformer, which doubles the amplitude of the signal received at the antenna, followed by an envelope detector and a micropower amplifier with a digipot in the feedback loop for dynamic thresholding. The envelope detector (or AM demodulator) consists of a Schottky diode that is biased up to 100 mV, to further decrease the received signal strength required for detection. The input impedance of the op-amp serves as the envelope detector load.

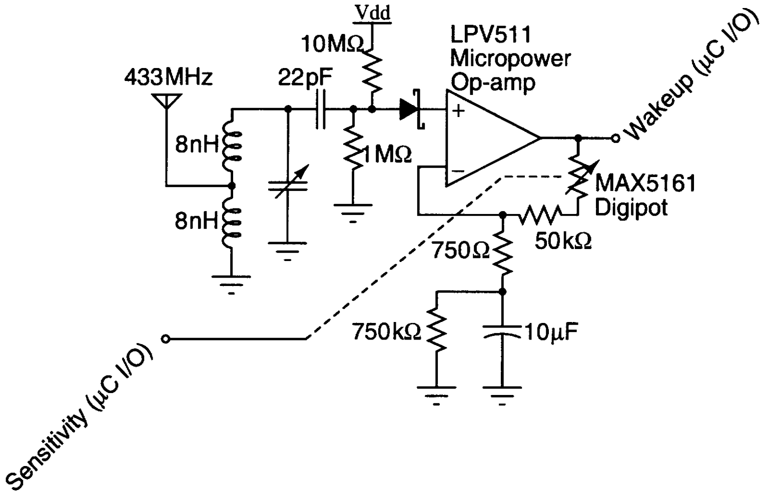


Figure 2.17: The RF detector circuit demodulates incoming 433 MHz radiation and boosts the signal to logic levels to wake the microcontroller.

The amplifier operates on the demodulated 10 kHz-range wakeup signal, boosting it by a factor of up to 300, so that it can serve as a logic-level wakeup input to the microcontroller. This high gain is attenuated at low frequencies to prevent amplification of offset voltages

[3] A reader/interrogator for use with CargoNet active tags is currently under development at the MIT Media Laboratory by Matthew Moskwa [38].

and other nonidealities. The gain is further controlled by a MAX5161 digitally-controlled potentiometer, which can lower the maximum gain to 80 to prevent false wakeups in noisy environments.

2.4.10 Piezoelectric Microphone

A 2.5 cm-diameter piezoelectric microphone (see photograph in Figure 2.18) was the last sensor added to the tag. Although the microphone is too large to be mounted to the tag PCB, it can be easily connected to it via a pair of wires and situated nearby to pick up sudden sounds. Data collected by the microphone, as by the shock sensor and vibration dosimeter, indicate sudden movement and impacts. Although it seems redundant at first, the microphone can provide information about shocks that do not occur along the vibratab's primary axis of sensitivity, or at high frequencies (see Figure 2.19). Furthermore, by analyzing the microphone waveform after wakeup, it may even be possible to determine what caused the impact, or whether an object inside the tagged shipment shattered.

Figure 2.18: A piezoelectric microphone can wake the microcontroller on loud sounds.

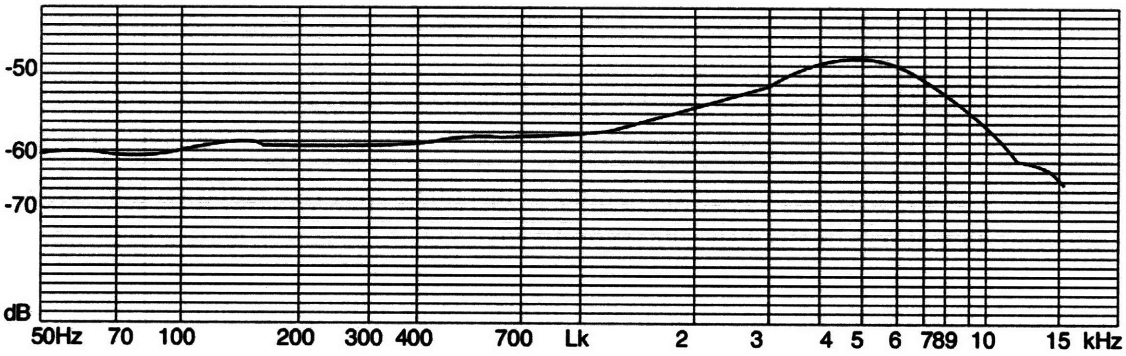


Figure 2.19: Frequency response of the 25LM025 piezoelectric microphone [25].

The interface circuit for the microphone is similar to that for the shock sensor, and consists of a MAX5161 digipot as the microphone load, followed by an LTC1540 nanopower comparator, which alerts the microcontroller when a sound loud enough to pass the threshold has occurred.

2.4.11 External Flash Memory

The MSP430F135 microcontroller only has 16 kB of internal flash memory, part of which must be occupied by the tag firmware. For some of the system tests involving large amounts of data collection, such as a month-long exercise in East Asia described in Section 4.1.1, the internal memory would prove insufficient, so the CC2500 daughter card was removed from the CargoNet board and replaced with the AT45DB081B 8 Mbit SPI flash.

This memory is small and inexpensive, and furthermore consumes only 2 μA of current in standby mode [1]. This is significantly more than the microcontroller's LPM4 consumption, but still less than that of competing flash memories. Therefore, while this flash memory was only included in the project for testing and calibration, its selection is consistent with the general goals of the project. It may be used for many low-power applications that require extensive data storage, and because the SPI bus can be shared by several peripherals, it can even be used in addition to the radio.

2.5 Summary

A schematic and printed-circuit board layouts of the complete active tag can be found in the Appendices. As can be seen from the descriptions in the preceding section, extensive efforts have been made to find a balance between the competing requirements of low cost and low power consumption. The results of these optimizations are summarized in Tables 2.3 and 2.4, which detail the bill of materials and the power budget for the system, as specified by the individual device datasheets.

The total parts cost of the system surpasses 40 dollars and approaches that of the TelosB presented in Table 2.1. It is important to note, however, that Table 2.1 is only a *partial* bill of materials: TelosB's full component price is likely to surpass CargoNet's, without providing the additional sensors pertinent to supply-chain management.

The power budget in Table 2.4 is a crude estimate of power consumption, since it is based on quiescent operation, and much will depend on the duty cycle of the sensors. This balance

Component	Part No.	Qty. / Tag	Total Qty.	Price at 25 Units [USD]	Total per Tag [USD]
Microcontroller	MSP430F135IRTDT	1	25	7.20	7.20
Comparator	LTC1540CMS8	2	50	2.24	4.48
Op-amp	LPV511MG	2	50	2.20	4.40
Piezo Microphone	25LM025	1	25	3.74	3.74
Digipot	MAX5161	3	75	1.24	3.72
Radio Transceiver	CC2500RGP	1	25	3.15	3.15
26 MHz Crystal	405C35B26M00000	1	25	1.56	1.56
Lithium Battery	CR2032FH-LF	1	25	1.36	1.36
Piezo Vibratab	LDTC-Horizontal	1	25	1.09	1.09
Real-Time Clock	PCF8563TD-T	1	25	1.06	1.06
Mini USB-B Jack	UX60A-MB-5ST	1	25	1.01	1.01
32,768 kHz Xtal	CM200S32.768KDZB	1	25	0.75	0.75
Tilt Switch	107-1007	1	25	0.70	0.70
Reed Switch	RI-01C	1	25	0.69	0.69
Dual Diode	BAS16DXV6T1G	2	50	0.30	0.59
Op-amp	MCP601T-I/OT	1	25	0.54	0.54
Dual pFET	NTJD2152PT1G	1	25	0.48	0.48
Phototransistor	PT100MC0MP	1	25	0.30	0.30
Dual nFET	2N7002DW-TP	1	25	0.20	0.20
Dual Schottky	BAT54AWT-TP	1	25	0.13	0.13
LED	SML-LX0603IW-TR	1	25	0.13	0.13
Schottky Diode	BAT54WT-TP	1	25	0.11	0.11
Other Passives					4.15
Total Cost					41.53

Table 2.3: The bill of materials for version 4 of the CargoNet active tag, with part numbers and small-quantity prices.

between active and quiescent states is controlled by the firmware, which is the subject of Chapter 3. An additional confounding factor is the external flash memory described in Section 2.4.11, which was attached to the active tag in place of the radio, and draws an additional 2 μA over the figure in the table.

Module	Component	Typ. Quiescent Current [μA]
Microcontroller	MSP430F135 Microcontroller	0.1
Radio Transceiver	CC2500 RF Transceiver IC	0.4
Real-Time Clock	PCF8563 Real-Time Clock IC	0.25
Low-Cost Humidity Sensor	(Polled sensor)	0
Light Sensor	PT100MC0MP Phototransistor	0.1
Reed Switch	NTJD2152 FET	0.01
	BAS16DXV Diode	0.004
Tilt Switch	NTJD2152 FET	0.01
	BAS16DXV Diode	0.004
Shock Sensor	LTC1540 Comparator	0.3
	MAX5161 Digipot	0.2
	Resistor Divider	0.36
Vibration Dosimeter	LPV511 Op-amp	0.8
	2N7002DW FET	0.025
RF Detector	LPV511 Op-amp	0.8
	MAX5161 Digipot	0.2
	Resistor Divider	0.27
Piezoelectric Microphone	LTC1540 Comparator	0.3
	MAX5161 Digipot	0.2
	Resistor Divider	0.5
Total Current at 3 V		4.83

Table 2.4: The quiescent power budget for the CargoNet active tag, based on manufacturers' typical figures.

Chapter 3

Firmware Design

Because of cost and power constraints, the CargoNet platform has at its core a Texas Instruments MSP430 16-bit processor with 512 bytes of RAM and 16 kB of storage, which must control the sensors described in Section 2.4, process and store received data, and respond to readers and communicate with other tags nearby. Although the hardware has already been designed to maintain low power consumption in its quiescent state, it rests on the firmware to control the duty cycle and active-mode operation.

The firmware for the CargoNet active tag was written in C with some macros provided by Rowley Crossworks 1.3, the integrated development environment used throughout the project.

3.1 Operating Principles

The primary principle motivating the design of the active tag firmware has been to keep the hardware in its low-power quiescent state for as much time as possible. This mandates that the system be completely interrupt driven, with the processor switching to the next waiting task or retreating to one of the low-power modes whenever it is not being used. Since most of the sensors have already been designed to interrupt the microcontroller if an important event occurs, the structure of program execution is straightforward: wake up, process data, return to sleep. What is not straightforward, however, is how to manage the conflicts that are bound to arise as a result of numerous sensors vying for processor resources.

CargoNet is an omnibus platform, containing sensors that measure light, shock, vibration, tilt, temperature, humidity, and breach. Since some users may not need all the sensors for a particular type of shipment—for example, if the items in the shipment are individually sealed and packaged with a desiccant, so there is no need for humidity monitoring—or may want to repurpose the tag altogether, the firmware support for the sensors must be modular. Encapsulating the variables and procedures pertinent to each sensor in their own software module makes the platform more flexible, reduces the potential for conflict, and also hides some of the underlying differences between the sensors, making the platform easier to manage.

The overall state diagram for the system appears below in Figure 3.1, while the implementation details that follow from the operating principles are described in the following section.

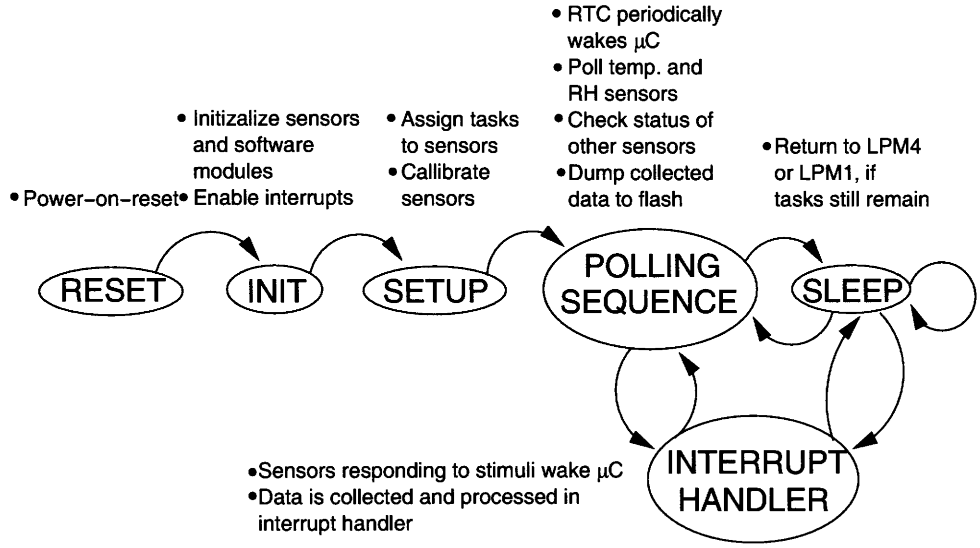

• Power-on-reset
• Initizalize sensors and software modules
• Enable interrupts
• Assign tasks to sensors
• Callibrate sensors
• RTC periodically wakes μC
• Poll temp. and RH sensors
• Check status of other sensors
• Dump collected data to flash
• Return to LPM4 or LPM1, if tasks still remain
RESET
INIT
SETUP
POLLING SEQUENCE
SLEEP
INTERRUPT HANDLER
•Sensors responding to stimuli wake μC
•Data is collected and processed in interrupt handler

Figure 3.1: A diagram of the firmware state machine.

3.2 Firmware Implementation

The microcontroller firmware has been designed to directly support the above principles of interrupt-driven processing and modularity. As mentioned previously in Section 2.4.1, all the sensors that respond to sudden events, such as sound, vibration, or light, are implemented passively and receive attention from the microcontroller—via interrupt—only when an above-threshold event occurs. These thresholds are furthermore adjusted dynamically to respond to changing conditions, limiting the number of unnecessary wakeups. Interrupts, along with a scheduler, are also used throughout the firmware to move the microcontroller between different tasks and between active and low-power modes.

3.2.1 Sensor Object Abstraction

Using multiple sensors on one platform requires software isolation to prevent conflicts. Each sensor or other system component is therefore represented by a software object that maintains variables and procedures influenced by and impacting the state of the sensor hardware. All the benefits of this abstraction—the reuse of application code for different hardware configurations or quicker implementation and debugging of new applications thanks to hardware libraries—are not yet realized since CargoNet involves only one sensor configuration and only one application. Nonetheless, the abstraction carries a small cost in storage space, but proved useful when on-chip storage was replaced by a larger external flash memory during a data-collection experiment.

The objects are implemented as C `structs` that contain function pointers to procedures such as initialization or data collection, and variables that maintain local state (an example appears in Figure 3.2). The object definition initializes the `struct` with an `init` procedure, which is then used by the main application to assign the other procedures to the object's function pointers and to initialize the physical sensor, as can be seen in Figure 3.3.

```c
/* Bat object. */
extern struct bat_s {
  void (*init) (void);
  U8 is_init;
  U8 (*measure) (void);
  U16 voltage;
  void (*adc12_callback) (void);
} bat;
```

Figure 3.2: A `struct` object that provides an abstraction to the battery voltage sensor.

3.2.2 Dynamic Thresholds

The dynamic thresholding strategy for limiting the number of successive tag wakeups to the same or a closely-related phenomenon was previously described in Section 2.4.2. This strategy is implemented by adjusting the apparent sensor signal strength with a MAX5161 digitally-controlled potentiometer. The digipot consists of a ladder of fixed resistors and switches that are able to adjust the resistance between 0 and 200 kΩ in 32 discrete steps [34].

```
/* Bat object.  Initialize with init function. */
struct bat_s bat = {bat_init};

static void bat_init (void) {
  /* Populate bat object: */
  bat.measure = bat_measure;
  bat.adc12_callback = bat_adc12_callback;
  bat.voltage = 0;              /* Start off with no information. */

  bat.is_init = TRUE;
}
```

Figure 3.3: Definition of `bat` object and its initialization procedure.

The tag begins operation with the digipot at the maximum resistance setting, to detect the smallest signal of interest. With each successful wakeup, the interrupt-handling routine for the sensor just stimulated "numbs" the sensor by doubling the resistance of the MAX5161, in an approach reminiscent of exponential backoff in Ethernet and other communication protocols. The resistance of the MAX5161 is increased by one step whenever the tag wakes up periodically to poll temperature and humidity sensors, gradually readying the sensor for new stimuli. The system is also amenable to more complex adaptation schemes: for example, the tags could vary the dynamic thresholds so as to maintain an average wakeup rate—and by extension power consumption—at some preset level [30].

3.2.3 Callbacks

The processor, which consumes 0.4 mA in active mode [65], is prevented from idling through liberal application of low-power modes and callbacks. Callbacks allow the processor to begin an independent task such as ADC sampling, then leave to perform another task or enter a low-power mode, but resume the initial task after the samples have been collected. Such a sampling example is illustrated in Figure 3.4. In this case, an interrupt handler for a sensor initiates ADC sampling, passing in the parameter `callback` a pointer to the function that should process the data after it is collected. This function is automatically called by the ADC's own interrupt handler (shown in Figure 3.10) after sampling has been completed, freeing the microcontroller to sleep or perform other tasks in the meantime.

3.2.4 Scheduler

Callbacks were also used to implement a scheduler for performing repetitive tasks. Rather than crafting a specific solution every time a function required a timer, a standard interface

```
/* Setup repeat-single-channel sampling.  Interrupt routine copies the
   sampled values from Memory 0 to the supplied buffer, and calls
   the callback function upon completion. */
static void adc12_sample (U16 *p_buffer, U8 count,
                          U16 time_interval, U16 delay,
                          void (*callback)(void)) {
  U16 prev_conversion;

  adc12.callback = callback;     /* Store callback in ADC12 object. */

  adc12.p_buffer = p_buffer;
  adc12.sample_count = count;

  /* Setup Timer_B */
  /* Timer will be run in UP mode, so it will count up to TBCCR0,
     setting OUT1 after reaching TBCCR1 (=1) and resetting at
     TBCCR0. */
  TBCCR0 = time_interval;        /* Time interval is nominally in
                                    microseconds (timer runs at 1MHz). */
  TBR = time_interval-delay;     /* Roll over to TBCCR0 value after
                                    initial sample delay. */

  TBCCTL1 &= ~OUT;               /* Start with output low. */
  ADC12IFG &= ~BIT0;             /* Clear IFG for memory slot 0. */
  ADC12IE |= BIT0;               /* Enable interrupt. */

  ADC12CTL0 |= (ADC12ON |        /* ADC12 on. */
                ENC |          /* Enable conversion——wait for trigger. */
                SHT0_7           /* Sample and hold time ~ 51uS. */
               );
  adc12.is_on = TRUE;
}
```

Figure 3.4: ADC sampling routine called by sensor interrupt handler accepts a callback—a pointer to a function which is called when the ADC finishes sampling.

to the on-chip 16-bit Timer_A was developed. By calling the function `timera.schedule()` with a pointer to a function to be executed and its desired period, a parent function can have a second function added to the scheduler. If this is the first task, Timer_A is started, and the function is called after the specified interval. For tasks that must be executed a specific number of times, it is the function's responsibility to `timera.remove()` itself from the scheduler following the final iteration. This scheduler simplifies the use of Timer_A through modularity (the tasks do not need to be aware of each other, and the scheduler resolves any potential conflicts between them), and allows it to be used more often for timekeeping, which permits the microcontroller to execute other tasks or go to sleep. Code pertaining to the scheduler appears in Figures 3.5, 3.6, and 3.7.

```
/* Schedule a repeated task to happen after x counts. */
static U16 timera_schedule (void (*task) (void), U16 period) {
  /* If timer is off, start: */
  if (timera.is_on == 0)
      timera.start();
  /* Find free capture/compare register. */
  if (timera.p_task_0 == 0) {   /* If CCR 0 unused . . . */
    timera.p_task_0 = task; /* Schedule task to be executed on int. */
    timera.period_0 = period;
    TACCR0 = TAR + period;        /* Set int to occur after period. */
    TACCTL0 &= ~CCIFG;            /* Clear interrupt flag. */
    TACCTL0 |= CCIE;              /* Enable CCR interrupt. */
    return ((U16) &timera.p_task_0); /* Return pointer to task LOC. */
  }
  else if (timera.p_task_1 == 0) {
    timera.p_task_1 = task; /* Schedule task to be executed on int. */
    timera.period_1 = period;
    TACCR1 = TAR + period;        /* Set int to occur after period. */
    TACCTL1 &= ~CCIFG;            /* Clear interrupt flag. */
    TACCTL1 |= CCIE;              /* Enable CCR interrupt. */
    return ((U16) &timera.p_task_1); /* Return pointer to task LOC. */
  }
  else if (timera.p_task_2 == 0)  {
    timera.p_task_2 = task; /* Schedule task to be executed on int.*/
    timera.period_2 = period;
    TACCR2 = TAR + period;        /* Set int to occur after period. */
    TACCTL2 &= ~CCIFG;            /* Clear interrupt flag. */
    TACCTL2 |= CCIE;              /* Enable CCR interrupt. */
    return ((U16) &timera.p_task_2); /* Return pointer to task LOC. */
  }
  else {                          /* No free CCR's! */
    return ERROR;                 /* Let calling function deal with it.*/
  } // Alternative is to also execute the task immediately?
}
```

Figure 3.5: The Timer_A scheduler accepts function pointers for tasks that must be executed periodically and provides a modular interface to the timer.

3.2.5 Flash Memory Writes

Programming flash memory requires a higher voltage than is provided by the 3 V lithium cell, but both the on-chip as well as external flash memories provide a charge pump to raise the battery voltage to the required level. The charge pump requires time to both ramp up the voltage and to power down, consuming 3 mA in the process in the case of the MSP430 on-chip flash [65]. This setup time is considerable (up to 66% of the word program time or an additional 38 ms [65]), but can fortunately be amortized over multiple bytes by writing an entire block of data at once.

During the testing of the platform aboard a merchant ship, a large amount of data had to be recorded for later analysis, so an 8 megabit external flash was used for extra capacity,

```
/* Remove a previously scheduled task. */
static U8 timera_remove (void (*task) (void)) {
  if (timera.p_task_0 == task) {          /* If it's task 0. */
    TACCTL0 &= ~CCIE;                  /* Disable interrupt. */
    timera.p_task_0 = 0;                      /* Reset. */
  }
  else if (timera.p_task_1 == task) { /* If it's task 1. */
    TACCTL1 &= ~CCIE;                    /* Disable interrupt. */
    timera.p_task_1 = 0;                        /* Reset. */
  }
  else if (timera.p_task_2 == task) { /* If it's task 2. */
    TACCTL2 &= ~CCIE;                    /* Disable interrupt. */
    timera.p_task_2 = 0;                        /* Reset. */
  }
  else          {                  /* It's none of the tasks! */
    return ERROR; /* Since task apperently wasn't scheduled, not doing
                     anything is equivalent to removal. */
  }
  /* If all the tasks are zero——no remaining tasks.  Stop timer.  */
  if ((timera.p_task_0 || timera.p_task_1 || timera.p_task_2) == 0)
    timera.stop();                 /* Stop timer. */
  /* Else keep it running! */
  return NOERROR;
}
```

Figure 3.6: Tasks are removed from the scheduler using the `timera.remove()` function call.

as it can provide over a month of data storage at a rate of 20 bytes per minute. Although this flash memory requires full page (256 + 8 bytes) programming, the IC also includes two RAM buffers for transfers of less than a page, so theoretically one could transfer single bytes to the buffers, synchronizing with the flash when they fill up.

Communication with the flash via SPI carries so much overhead, however, that it is economical to maintain a *third* local buffer in the microcontroller memory, and copy it to one of the flash RAM buffers all at once. Otherwise, each single-byte transfer would require the transfer of *seven* auxiliary bytes containing address, opcode, and status data, wasting not only power, but also the limited time available for the microcontroller to respond to sudden events.

By making available a third 256-byte buffer, stored in MSP430 RAM as `flash.temp_buffer`, individual bytes can be easily copied after sensor data processing. This process is illustrated in the fragment of low-cost humidity sensor code in Figure 3.8, where the continuity of the data written to the flash is protected by temporarily disabling interrupts. After the microcontroller has completed all its tasks and is about to return to sleep, the contents of `flash.temp_buffer` are copied to the flash.

All log entries are of fixed length and are preceded by a one-byte code that identifies

```
void timera0_irq_handler (void) __interrupt[TIMERA0_VECTOR] {
  if (timera.task_0 == timera.delay_task)
      LPM0_EXIT;                      /* Blank task——exit LPM0. */
  else {                              /* Reschedule task and run it. */
    TACCR0 = TAR+timera.period_0;
    TACCTL0 &= ~CCIFG;                /* Clear IFG. */
    TACCTL0 |= CCIE;                  /* Reenable interrupts. */

    timera.task_0();                  /* Run task. */
  }
}
```

Figure 3.7: Timer_A interrupt handlers execute the scheduled tasks when their respective periods elapse. The next execution time is then calculated by adding the period of the task to the current timer value (stored in the TAR register, modulo the period of the timer itself: 2^{16}). This code repeats for the other two Capture/Compare Registers (CCR1/2).

the source of the entry. Most of the log entries are also preceded by a six-byte time stamp, which is identified by its own code. When data is later collected for analysis, the parser separates the entries by sensor and groups, each one with the most recent time stamp, while using the fixed entry length associated with the code for error checking. The list of codes appears in Table 3.1.

Sensor	Code	Payload
Tilt Switch Closed	0x10	None
Tilt Switch Open	0x20	None
Reed Switch Closed	0x30	None
Reed Switch Open	0x40	None
Light Absent	0x50	None
Light Present	0x60	None
Shock Sensor	0x70	Maximum MSB Maximum LSB Sensitivity
Vibration Dosimeter	0x80	Level MSB Level LSB
RTC Time Stamp	0x90	Year Month Day Hour Minute Second
SHT11 Humidity	0xA0	Humidity MSB Humidity LSB
SHT11 Temperature	0xB0	Temperature MSB Temperature LSB
Low-cost Humidity	0xC0	Max. MSB Max. LSB Avg. LSB Avg. LSB
Internal Temperature	0xD0	Temperature MSB Temperature LSB
Piezo Microphone	0xE0	Maximum MSB Maximum LSB Sensitivity
Battery Monitor	0xF0	Level MSB Level LSB

Table 3.1: Identification codes and data fields used in the event log.

To ensure data integrity, the firmware loaded onto the tags during testing did not allow flash erasure. Instead, the flash memory was erased by a separate program that was loaded and removed from memory immediately prior to loading the test code. Without the capacity

```
/* Measure cheap humidity sensor. */
hum.measure();              /* Measure humidity using cheap sensor. */
_BIS_SR(LPM1_bits+GIE);        /* Low power mode. */
_NOP();

_DINT();                           /* Make writes atomic. */
if(flash.index < FLASH_TEMP_BUFFER_SIZE)
  flash.temp_buffer[flash.index++] = HUM_OPCODE;
if(flash.index < FLASH_TEMP_BUFFER_SIZE)
  flash.temp_buffer[flash.index++] = (U8) (hum.max >> 8);
if(flash.index < FLASH_TEMP_BUFFER_SIZE)
  flash.temp_buffer[flash.index++] = (U8) (hum.max & 0xFF);
if(flash.index < FLASH_TEMP_BUFFER_SIZE)
  flash.temp_buffer[flash.index++] = (U8) (hum.avg >> 8);
if(flash.index < FLASH_TEMP_BUFFER_SIZE)
  flash.temp_buffer[flash.index++] = (U8) (hum.avg & 0xFF);
/* Write peak and average reading for regressions. */
_EINT();
```

Figure 3.8: Humidity data collected through `hum.measure()` are written to a temporary buffer before being copied en-masse to flash memory.

to erase any existing data, the flash memory initialization routine scans successive memory pages at startup until encountering an empty one. This strategy also ensures that data will be preserved and continuous despite a reset, which clears the write pointer and any other data in microcontroller RAM.

3.2.6 Analog-to-Digital Converter

The analog-to-digital converter (ADC) is employed by many of CargoNet's on-board sensors, including the shock sensor, vibration dosimeter, low-cost humidity sensor, internal temperature reference, battery monitor, and others. Because the ADC is used by so many other modules, the microcontroller firmware has been extended to provide a standard interface. The ADC is initialized with the common settings (use of Timer_B as sampling trigger, Timer_B source clock and frequency, single-channel multiple conversion mode, etc.) at startup, so that each module that wishes to use the ADC must only specify the input channel (the ADC pin to which the sensor is connected) and choice of positive and negative reference.

The sampling is initiated with a call to `adc12.sample(...)`, the code for which appears in Figure 3.9, and which takes five standard arguments: a pointer to a data buffer, the number of samples to be taken, the sampling frequency, the sampling delay, and finally a pointer to the function that must be called after the samples are collected. As can be seen in Figure 3.9, the function parameters are written to the pertinent ADC registers and the ADC core is powered on. Control returns to the sensor's `measure()` function, which starts Timer_B to begin conversions.

```
/* Setup repeat-single-channel sampling.  Interrupt routine copies the
   sampled values from Memory 0 to the supplied buffer, and calls
   the callback function upon completion. */
static void adc12_sample (U16 *p_buffer, U8 count,
                          U16 time_interval, U16 delay,
                          void (*callback)(void)) {
  U16 prev_conversion;

  adc12.callback = callback;      /* Store callback in ADC12 object. */
  adc12.p_buffer = p_buffer;
  adc12.sample_count = count;
  /* Setup Timer_B */
  /* Timer will be run in UP mode, so it will count up to TBCCR0,
     setting OUT1 after reaching TBCCR1 (=1) and resetting at
     TBCCR0. */
  TBCCR0 = time_interval; /* Time interval is nominally in microseconds
                             (timer runs at 1MHz). */
  TBR = time_interval-delay; /* Roll over to TBCCR0 value after
                                initial sample delay. */

  TBCCTL1 &= ~OUT;                /* Start with output low. */
  ADC12IFG &= ~BIT0;              /* Clear IFG for memory slot 0. */

  ADC12IE |= BIT0;                /* Enable interrupt. */

  ADC12CTL0 |= (ADC12ON |         /* ADC12 on. */
                ENC |          /* Enable conversion——wait for trigger. */
                SHT0_7            /* Sample and hold time ~ 51uS. */
                );
  adc12.is_on = TRUE;
}
```

Figure 3.9: The `adc12.sample(...)` function prepares the ADC for collecting samples.

An ADC interrupt is requested after every sample has been collected, and the IRQ handler counts down the number of samples left and restarts sampling, or in the case of the final sample, shuts down the ADC core and executes the sensor callback to process the collected data. The code for the ADC interrupt handler appears in Figure 3.10.

3.3 Sensor Control Firmware

The firmware written to control each sensor module utilizes many of the above techniques, such as callbacks or scheduled tasks, but it also employs strategies appropriate to a particular sensor.

```
void adc12_irq_handler (void) __interrupt[ADC12_VECTOR] {
  if (--adc12.sample_count) {  /* If additional samples to convert. */
    /* Read data from memory. */
    *adc12.p_buffer++ = ADC12MEM0;
  }
  else {                           /* Final sample already read. */
    *adc12.p_buffer++ = ADC12MEM0; /* Read data from memory.  */
    ADC12CTL0 &= ~ENC;              /* Disable further conversions. */
    TBCTL &= ~(MC0 | MC1);         /* Stop Timer_B. */

    ADC12IE &= ~BIT0;              /* Disable interrupt. */
    ADC12CTL0 &= ~ADC12ON;         /* Power down ADC. */

    LPM1_EXIT;                     /* Leave low-power mode. */
    TBR = 0xFFFF;                  /* Reset timer. */

    if (adc12.callback) /* If pointer nonzero--it's implemented. */
      adc12.callback(); /*  Call! */
    /* Else, return. */
    adc12.is_on = FALSE; /* ADC finished. Other callers can now use.*/
  }
}
```

Figure 3.10: ADC interrupt request handler.

3.3.1 Switch Sensors

As described in Sections 2.4.5 and 2.4.6, the hardware for the switch sensors that detect tilt and breach conditions has been designed such that they consume minimal power in the absence of those conditions (the vast majority of the time). If the switch eventually opens (in the case of the tilt or magnetic breech sensors) or closes (light sensor), a non-negligible current will flow through the active element of the sensor circuit (either PFET or phototransistor), causing a voltage change across the load resistor and waking the microcontroller.

As can be seen in the interrupt-handler code in Figure 3.11, the microcontroller logs the event (tilt in this case) and powers down the sensor, as the sensor will now consume power as long as the fault condition persists. The sensor is then periodically polled to see if the fault condition has been eliminated (the shipment has been set back upright); if so, it is powered on and re-armed.

3.3.2 Sampled Sensors

After waking up the microcontroller, the shock sensor and piezoelectric microphone are sampled numerous times in an effort to find the peak of the incoming stimulus. The lower and upper bytes of this maximum reading are written to the event log, as specified in

```
else if (P2IE & TILT_INT_MASK && P2IFG & TILT_INT_MASK) {
  P2OUT &= ~TILT_PWR_MASK;
  P2IE &= ~TILT_INT_MASK;
  /* Tilt switch interrupt stays disabled until next
  timer polling sequence. */

  rtc.sync_time();              /* Get latest time.  */
  if(flash.index < FLASH_TEMP_BUFFER_SIZE)
    /* Schedule write of time of vibration to flash. */
    flash.temp_buffer[flash.index++] = RTC_OPCODE;
  for (i=0; i<sizeof(rtc.time); i++)
    if(flash.index < FLASH_TEMP_BUFFER_SIZE)
      /* Copy rtc.timer to temp buffer */
      flash.temp_buffer[flash.index++] = ((U8 *) &rtc.time)[i];
  if(flash.index < FLASH_TEMP_BUFFER_SIZE)
    flash.temp_buffer[flash.index++] = TILT_OPEN_OPCODE;
}
```

Figure 3.11: Tilt sensor interrupt-handler code.

Table 3.1. In each case, the sampling frequency has been selected to be high enough to prevent aliasing: around 1 kHz for the shock sensor and 10 kHz for the microphone.

The low-cost humidity sensor is operated similarly, except that rather than responding to sudden events in the environment, the microcontroller chooses when to poll the sensor. As Figure 3.12 illustrates, upon a call to `hum.measure()`, the microcontroller powers on the humidity measurement circuit, and begins to drive the input to the circuit with a square wave via a special Timer_A task. After the transients die out (around 250 ms), the output of the humidity sensor is sampled 16 times at a frequency of 3.33 kHz via a call to `adc12.sample(...)`.

The microcontroller is free to perform other tasks while the ADC collects the samples, but returns to process the humidity data collection after receiving an ADC interrupt request. The samples had been stored by the ADC in the 16-bit `hum.buffer[]`, and as can be seen in Figure 3.13, the microcontroller steps through this buffer collecting statistics on the samples. The average and maximum are then logged to flash.

3.4 Summary

The firmware of the CargoNet active tag has been designed to limit the proportion of time the microcontroller and the sensors spend in their active versus quiescent modes, while maintaining modularity that permits the easy addition and reconfiguration of sensors when tailoring the platform to a specific application. The strategies described in this chapter

```
static void hum_measure (void) {
  if (hum.is_init) {
    while (adc12.is_on == TRUE); /* Wait for ADC12 to free up. */
    P3OUT |= HUM_PWR_MASK;       /* Power on. */
    /* Toggle driving pin every half-period.  HUM_DRIVE_PERIOD is in
       miliseconds and TIMERA_FREQ is in kilohertz. */
    timera.schedule(hum.in_toggle, (HUM_DRIVE_PERIOD*TIMERA_FREQ/2) );

    /* Wait for transients to settle. HUM_TRANSIENT_TIME in ms,
       TIMERA_FREQ in kHz.*/
    timera.delay(HUM_TRANSIENT_TIME*TIMERA_FREQ);
    /* Save data in memory location 0, with AVcc as positive reference
       and VeREF- as negative, and using input channel A0. */

    ADC12MCTL0 = (SREF_0 | INCH_0); // Test without negative reference
    /* HUM_SAMPLING_TIME is in *MICROSECONDS (31us)*, while
       TIMERB_FREQ is in kHz, so divide by 1000.*/
    adc12.sample(hum.buffer, HUM_BUFFER_SIZE, HUM_TIMERB_COUNT,
                 HUM_SAMPLING_DELAY, hum_adc12_callback);
    /* Sampling over one period.  Will sort out positives and
       negatives in software. */
    while ((P4OUT & HUM_IN_MASK));
    while ((P4OUT & HUM_IN_MASK) == 0); /* Wait for rising edge. */
    /* Start ADC! */
    TBCTL |= MC_1;          /* Start timerB in up mode. */
  }
}
```

Figure 3.12: Low-cost humidity sensor measurement algorithm.

```
void hum_adc12_callback (void) {
  U8 i;
  U16 avg;
  U16 max;

    timera.remove(hum.in_toggle);
    P3OUT &= ~HUM_PWR_MASK;       /* Power down sensor. */

    /* Now do processing and conversion. */
    /* Calculate avg: */
    for (i = 0, avg = 0, max = 0; i < HUM_BUFFER_SIZE;) {
      /* ADC readings are around 2^11: don't worry about overflow.  */
      hum.buffer[i] = 0x0800 - hum.buffer[i]; /* Sub. from midrail. */
      if (hum.buffer[i]>max) /* Reading can't be min and max at
                                the same time. */
        max = hum.buffer[i];
      avg += hum.buffer[i++];
    }
    avg >>= 4;          /* Divide by sixteen——size of buffer. */
    hum.avg = avg;
    hum.max = max;
}
```

Figure 3.13: The humidity sensor ADC callback processes the sampled data before writing it to flash.

build upon a solid foundation of low-power hardware described in Chapter 2, but their performance is not guaranteed; in Chapter 4 they will be put to the test both in the laboratory and in the field, potentially resulting in some changes. But even if the firmware (or hardware) implementation must undergo further revision in reaching the goals of low cost at a low price, the techniques described in this chapter can guide further developments toward those goals.

Chapter 4

Testing and Analysis

The CargoNet tags were briefly tested in the laboratory and in a container yard in Kearny, New Jersey, where they were subjected to the shocks common in the moving and stacking of containers (see Figure 4.1). Soon afterwards, the fourth revision of CargoNet active tags were augmented with an external 8 Mbit flash memory and two AAA alkaline batteries, placed inside small ABS plastic cases for safety, and sent to Singapore as part of Intel Corporation's tests of the Intelligent Container Project. Upon arrival in Singapore, seven tags were placed inside an empty shipping container and proceeded to record the conditions inside the container. The container was loaded onto a cargo ship and travelled for a week between Singapore and Kaohsiung, Taiwan, and the tags continued to record for up to three additional weeks as they made their way back to the United States for analysis. A description of the Singapore test follows, along with descriptions of additional tests performed later after changes had been made to the tag.

4.1 Singapore Test

The Singapore test sought to establish the basic functionality of the low-power sensors and the firmware, to evaluate the performance of the system after an extended period of time and under strenuous conditions, and to record the broad range of stimuli to which the tags can be subjected. It was also hoped that data collected during the test would help tune the sensors for future assignments. The tags performed reasonably well: most of the tags initialized correctly after their journey to Singapore, the majority of their sensors working and recording data. Several of the tags operated for the entire month in the field.

Figure 4.1: Shipping containers arranged in a stack in Kearny, New Jersey.

The failures were more informative than the successes, however, and the latter half of this chapter describes the efforts to correct the shortcomings of the system through adjustments and additional tests.

4.1.1 Test Setup

The seven tags assembled for the Singapore test were each equipped with an external 8 Mbit flash memory, a piezoelectric microphone, and a larger battery pack, before being enclosed in a hard ABS plastic case as portrayed in Figure 4.2. The tags were then sent—with their batteries disconnected—to George Cavage from APL Logistics, who delivered them to Singapore and placed them inside an empty 20-foot shipping container with the help of Richard Tyo from Intel. To measure the effect of packaging on the exposure of the tags to environmental stimuli, the tags were divided into three groups—unenclosed, enclosed, enclosed and cushioned—and distributed throughout the container. The unenclosed tags were taped to the wooden floor of the shipping container, while the enclosed tags were placed inside a cardboard box that was taped shut. The enclosed and cushioned tags were wrapped in packing foam before placement in the cardboard box. Table 4.1 records the distribution of tags between the different groups, while Figures 4.3 and 4.4 show their locations inside the container.

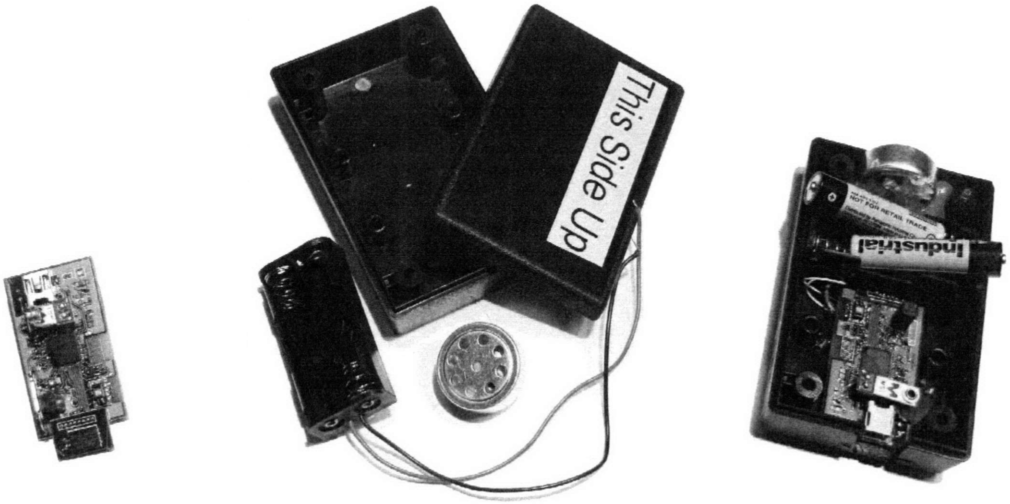


Figure 4.2: An active tag packaged for testing aboard a ship in East Asia.

This sea journey was undertaken as part of Intel Corporation's Intelligent Container

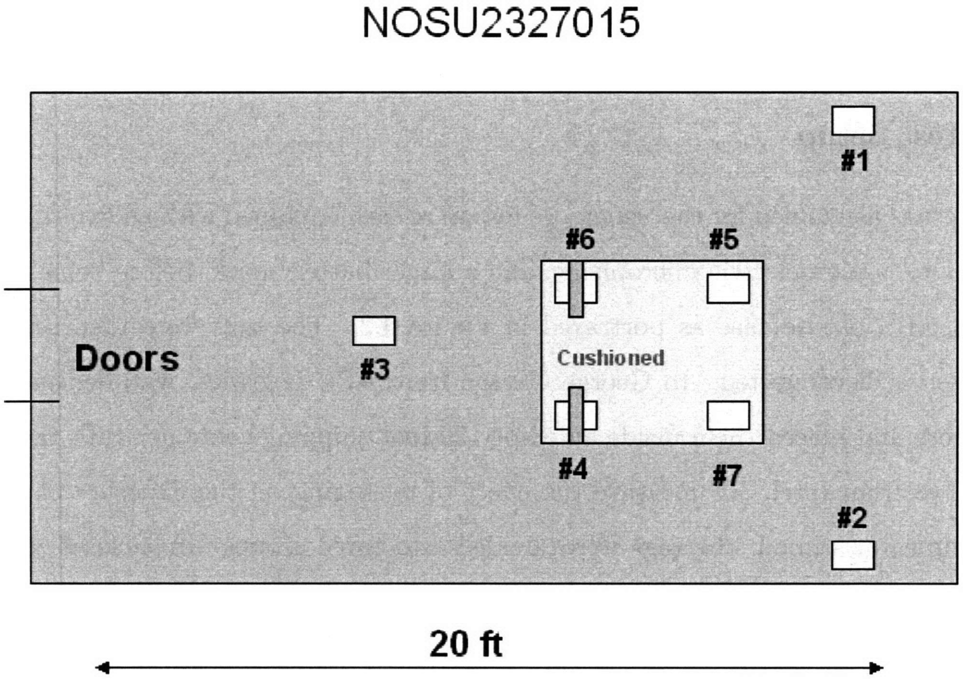


Figure 4.3: A sketch of the placement of tags inside the shipping container [5].

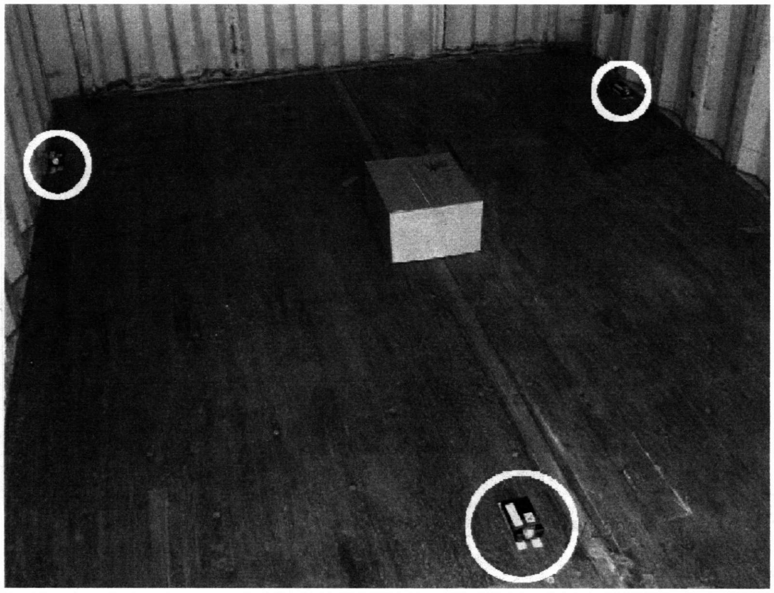


Figure 4.4: A photograph of the tags inside the shipping container [5].

Tag No.	Group
1	Unenclosed
2	Unenclosed
3	Unenclosed
4	Enclosed and cushioned
5	Enclosed
6	Enclosed and cushioned
7	Enclosed

Table 4.1: The distribution of tags between the three placement groups.

Project [50], whose participants were testing data collection and networking inside and outside shipping containers stacked aboard a ship. CargoNet fits neatly inside the Intelligent Container Project framework, for while the Project's sensors perform many of the same functions as CargoNet, they are too expensive (the cheapest components of the system are TelosB motes) and too power-hungry to be deployed inside crates and cases. Furthermore, some future incarnation of CargoNet sensors may use the backbone provided by TelosB and other nodes resident in the container to communicate alerts to the ship's crew or even to orbiting satellites.

After being placed inside empty containers, the CargoNet tags and Intel's sensors were loaded aboard a ship, which departed Singapore on the morning of 4 November 2006. The tags arrived in Kaohsiung, Taiwan, on the afternoon of 9 November, and they were taken from the ship and put in storage for a week before being sent back by ship to Singapore, where they arrived about one week later. They were finally airmailed back to MIT in Cambridge, Mass., directly from Singapore, on 27 November. Some of the tags continued to record data for these additional weeks, up to their return to MIT on 1 December.

The following sections detail the successes of the CargoNet low-power active tag platform, as evidenced by data collected during the voyage between Singapore and Kaohsiung. Although not all the sensors worked as expected, and several tags failed suddenly, the test established the validity of the CargoNet concept and pointed the way to further evaluations in the laboratory and in the field.

4.1.2 Data Collection and Import

Before discussing the performance of the tags at length, it is important to note how the data were stored in flash memory and later imported to a computer for analysis, as errors in this process will negatively impact the perceived operation of the on-board sensors.

Writes to Flash

As mentioned in Section 3.2.5, writes to flash were double buffered, first in the microcontroller RAM and then in the flash's on-chip RAM, to amortize the power costs of SPI transfer and flash charge-pump initialization across multiple bytes. Under this scheme, sensor data were written to a local 64 byte buffer, which was periodically transferred to the flash RAM. Although buffer-size checks were written into the firmware to prevent buffer overflows, values exceeding the size of the buffer were simply dropped. A secondary buffer or some other method of storing data that did not fit into the local buffer was *not* implemented; instead, the excess data were simply dropped. This was due to a lack of time before the test and also in the belief that stimuli occurring in the later part of an extended event are of much less value to the tag than the initial stimulus, but carry the same power cost to process and store. Because instances where the data did not fit into the buffer were not recorded, glitches and other discontinuities in the results can therefore be attributed to the concatenation (in flash and later by the parser) of a data sequence which was disrupted by the end of the buffer to the first bytes of an unrelated sequence recorded soon afterwards.

Computer Import

After the tags were received at MIT, the data in their external flash memories were copied to a personal computer via the JTAG programming and debugging interface. A code, as per Table 3.1, precedes each sensor record, and not only specifies how the data should be processed but also the length of the record. After reading in a code, the parser checks whether the first byte of what should be the next record is itself a valid code; if not, it is discarded. This method of error checking can detect records that are too short or too long. Additional functions check the data fields for sensical values; for example, humidity values must be less than 100 while days of the month must number fewer than 32. Thanks to these safeguards, the parser works fairly well in detecting the overrun errors described above. The next step in data integrity would require checksums or cyclic redundancy checks (CRCs), and the overhead, both in terms of computation and storage space, was not deemed worth the benefit.

4.1.3 Real-Time Clock Performance

The real-time clock (RTC) is the basis for meaningful operation of all the other sensors: without correct time stamps, it is impossible to tell when (and consequently where) a phenomenon occurred and how long it persisted. Glitches can be mistaken for significant events and vice versa. Correlation between sensing modalities becomes impossible.

Data revealing the correct operation of the real-time clock were therefore greeted with some relief. Without exception, however, the data collected by each of the tags were not continuous. As can be seen in Figure 4.5, which shows the time stamp data for Tag #2, the raw time stamps increase for five days after the beginning of the test, but then return to the initial value. This discontinuity was due to a reset of the tag, which caused the microcontroller to reprogram the RTC with the starting date and time, and was easily corrected during analysis (in addition to adjusting for the actual start date of the experiment), as can be seen in the figure.

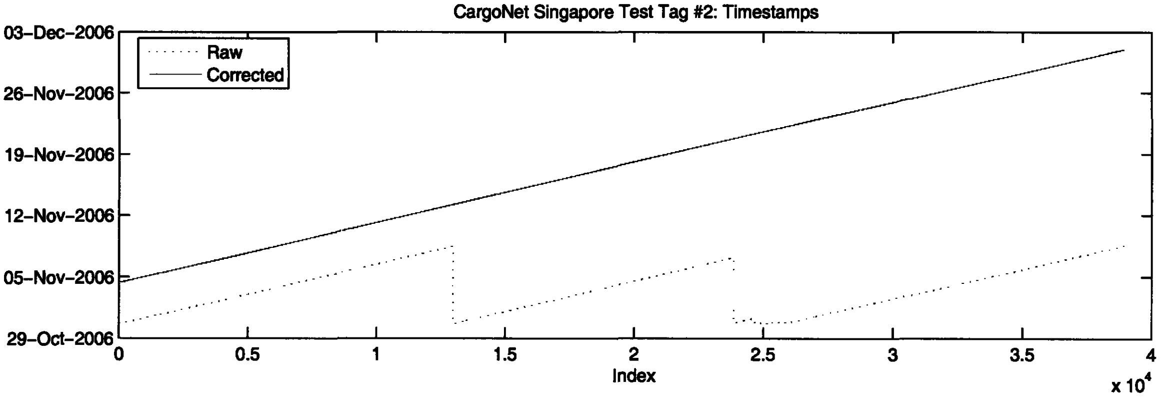


Figure 4.5: Tag time stamps were corrected during analysis ensuring a continuous and consistent timeline despite resets. The tag can be seen resetting numerous times in succession around point 25,000.

Such corrections would be unmanageable in a production environment, as not all time stamps are likely to be stored in a severely constrained tag memory. Furthermore, a mobile reader would not have the time necessary to parse each tag's history to assemble a coherent timeline across discontinuities. Optimally, the tag should not reset during operation (the code that most likely caused the resets, proactive recovery from an I^2C interface timeout, has since been removed from the firmware), but it is impossible to proscribe resets or crashes altogether.

Fortunately, there are ways to ensure time continuity despite resets. First, because the

RTC is external to the microcontroller, software resets of the latter are unlikely to affect the former's state, as long as the microcontroller does not automatically overwrite the RTC's registers (as happened in this case). Such a solution may even prove sufficient in the face of brief power interruptions, such as glitches due to faulty battery contacts, since the tag consumes so little power in quiescent mode that the voltage across the bypass capacitors may not drop below 1.8 V (the lower limit for RTC operation) during the interruption, preserving the RTC's state.

Even if power were removed from the tag circuit for an extended period of time, the microcontroller could recover the current time by performing the very corrections illustrated in Figure 4.5 at initialization time. The code would have to scan through memory and extract the latest time stamp, and write that to the RTC. Of course, such a strategy would depend on writing the time stamps to flash with satisfactory resolution, an action that may be too costly either in terms of power or storage space.

A final possibility involves nearby tags waking simultaneously following a shared stimulus, such as a powerful shock, and establishing radio communication to exchange data, for example the current time. Although tags could vote on which time is correct, a tag's microcontroller can know whether or not it had been reset, so erroneous times could be prevented from propagating during this exchange. Furthermore, incorrect time stamps recorded in the past could also be corrected, ensuring a coherent timeline when the tag is next interrogated by a reader.

4.1.4 Battery Life

The RTC time stamps, after the corrections described in the previous section, provide a quick measure of the longevity of the tags. By subtracting the first time stamp from the last, the longevities listed in Table 4.2 were obtained. The reason for the sudden expiration of certain tags can be seen in Figure 4.6, which plots the battery voltage of the tags with time.

The sudden expiration of the batteries in some tags, but not in others, indicates the presence of a bug, such as a mistake during assembly or a race condition in the firmware, rather than a design flaw. The very absence of battery voltage measurements for part of the time in the logs of four of the tags is due to a bug[1], so it is not too difficult to imagine that

[1] The readings for some of the sensors had to be dropped after they were corrupted during importing by an end-of-line character conversion.

Tag No.	Final Voltage [V]	Longevity
1	0.7	5 days 13 h 40 min
2	2.93	26 days 14 h 23 min
3	0.6	6 days 0 h 47 min
4	0	N/A[a]
5	2.5	0 days 4 h 24 min
6	3.1	5 days 15 h 29 min
7	2.98	27 days 19 h 41 min

[a] The tag collected no data.

Table 4.2: Longevity of tags during the Singapore test.

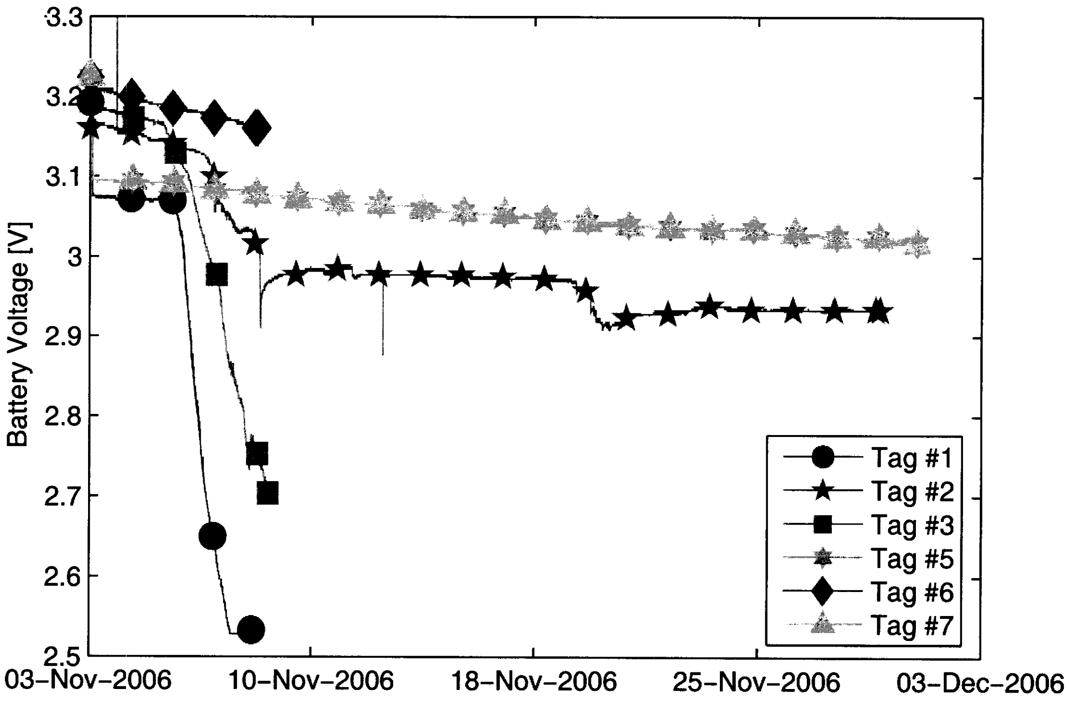


Figure 4.6: Tag battery voltages recorded during the Singapore test. Tags #4 and #6 recorded no data, whereas #2, #3, and #5 experienced large glitches during the first several days.

another bug was responsible for incorrect power management. One obvious source of error that could have led to the demise of the tags was the presence of traps in the tag firmware.

Traps are infinite `while` loops useful when debugging to catch a seemingly impossible case in a decision tree. For example, Figure 4.7 shows a trap in the `default` clause of a `switch` statement: any unspecified condition would land the processor in the trap, where it would cycle endlessly. Needless to say, all the traps should have been expunged from the firmware before deployment, and replaced with fault-tolerant code. As a second line of defence, and because pernicious loops can also arise in other situations, a watchdog timer should have been implemented, which would have made it possible for the microcontroller to leave the trap via a software reset.

```
default :
  while (1);    //XXX: TRAP
  break;
}
```

Figure 4.7: An example of a firmware trap that should have been removed prior to release.

Although tags #2 and #7 did not suffer from a bug which quickly drained their batteries, the voltage dropped far sooner than expected, as can be seen in Figure 4.6. According to the power budget outlined in Table 2.4, the tags should consume around 5 μA in standby, perhaps 7 μA with the external flash memory, which means they should operate for

$$225\,\mathrm{mAh} \cdot \frac{1\,\mathrm{day}}{24\,\mathrm{h}} \cdot \frac{1}{7\,\mu\mathrm{A}} = 1340\,\mathrm{days}, \tag{4.1}$$

or over three and a half years on a single 225 mAh lithium coin cell. Periodic active-mode operation will decrease that number somewhat, but the rate of discharge exhibited by tags #2 and #7 in Figure 4.6 is striking: if it were to continue at the rate in the graph, the battery voltage would drop to 2.7 V (the minimum voltage required for flash programming and low-power op-amp operation) within *four months.* Seemingly making matters worse is the fact that the AAA battery pack used during the Singapore test has a capacity of 1150 mAh [14], almost five times that of the lithium coin cell for which the tag was designed.

Further study following the test has revealed the problem with battery life to be twofold. As in the case of the above tags which expired after only several days, the increased power consumption of the tags that survived for the duration of the tests can also be blamed on insufficient cleanup of development code. Although the quiescent-mode current of the tags measures only 7 μA, the active-mode part of its periodic polling cycle is dominated by an 8.43 mA pulse which lasts 290 ms. This outsize current is used to illuminate the onboard LED to indicate correct operation during debugging, and the pertinent code should have been eliminated before the test, as it is responsible for increasing the average current consumption to

$$< i >= 8.43\,\mathrm{mA}\left(\frac{0.29\,\mathrm{s}}{60\,\mathrm{s}}\right) + 7.1\,\mu\mathrm{A}\left(\frac{60\,\mathrm{s} - 0.29\,\mathrm{s}}{60\,\mathrm{s}}\right) = 47.8\,\mu\mathrm{A}. \tag{4.2}$$

At first glance, it seems that the roughly 7× higher average current consumption caused by the LED should have been countered by the more capacious AAA batteries. As discussed previously in Section 2.3.3, however, alkaline cells are not well-suited for unregulated applications. So despite its low cost and high current capability, only 20% of a cell's capacity can be utilized before its voltage falls below 1.35 V (2.7 V/2), rendering it marginally useful for this application as the capacity of the two-battery pack is equivalent to that of the lithium coin cell.

Eliminating the LED-flashing code results in the active-mode current consumption pictured in Figure 4.8, which is equivalent to a 281 μA pulse for 584 ms, and which results in a cycle-average current consumption of

$$< i >= 281\,\mu\mathrm{A}\left(\frac{0.584\,\mathrm{s}}{60\,\mathrm{s}}\right) + 7.1\,\mu\mathrm{A}\left(\frac{60\,\mathrm{s} - 0.584\,\mathrm{s}}{60\,\mathrm{s}}\right) = 9.76\,\mu\mathrm{A}, \tag{4.3}$$

which should allow the tag to last up to

$$225\,\mathrm{mAh} \cdot \frac{1\,\mathrm{day}}{24\,\mathrm{h}} \cdot \frac{1}{9.76\,\mu\mathrm{A}} = 960\,\mathrm{days} \tag{4.4}$$

on a single lithium coin cell. The longevity will of course depend on the stimuli to which the tag is subjected, not only because of the power required to process the sensor data, but also to store it in flash and/or communicate it to tags and readers in the environment.

4.1.5 Tilt and Other Switched Sensors

The tilt switches on board the tags performed without failure during the Singapore test, though they tended to be overly sensitive. As can be seen in Figure 4.9, the switches tended to open immediately after being re-armed by the microcontroller during the active part of

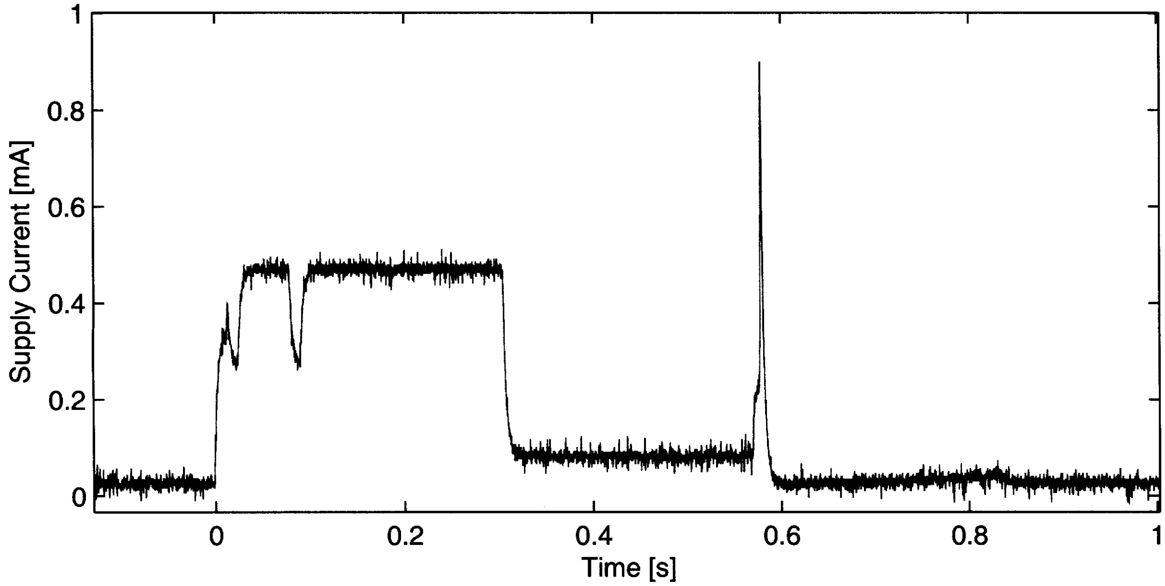


Figure 4.8: The current consumption of the CargoNet active tag during the active part of its one minute polling cycle. The short spike at the end of the waveform corresponds to writing data to flash.

the polling cycle. This indicates that the sensor was responding not to tilt but to sudden shocks that bounced the ball bearing inside the switch up from the contacts, breaking the circuit. A future revision of the board should have the switch placed on the underside of the board (i.e. normally open), so that when the ball bounces, it will have a much smaller chance of reaching the contacts and changing the state of the switch. In addition, as can be seen from Figure 4.9, the current firmware logs only the time when the switch first stays open as well as when it is finally re-armed. Since the logging procedure is far from foolproof, and data points may be lost, it would be beneficial to periodically log switch data, no matter what its state, to protect the data from the loss of a single point and also to better differentiate temporary shocks from longer-term tilts.

The data obtained from the phototransistor light sensor were mostly uninteresting: when the batteries were inserted, the sensor armed and immediately tripped, re-arming only after the tag was packaged in its opaque plastic case[2]. Again, periodic logging of sensor status or data integrity measures needs to be implemented to eliminate glitches in the data and to make it easier to verify the presence or absence of trigger phenomena.

No magnet was attached to the lid of the sensor case, so the magnetic reed switch recorded no data during this test.

[2] Actual applications will require that the sensor be packaged in a transparent or translucent case.

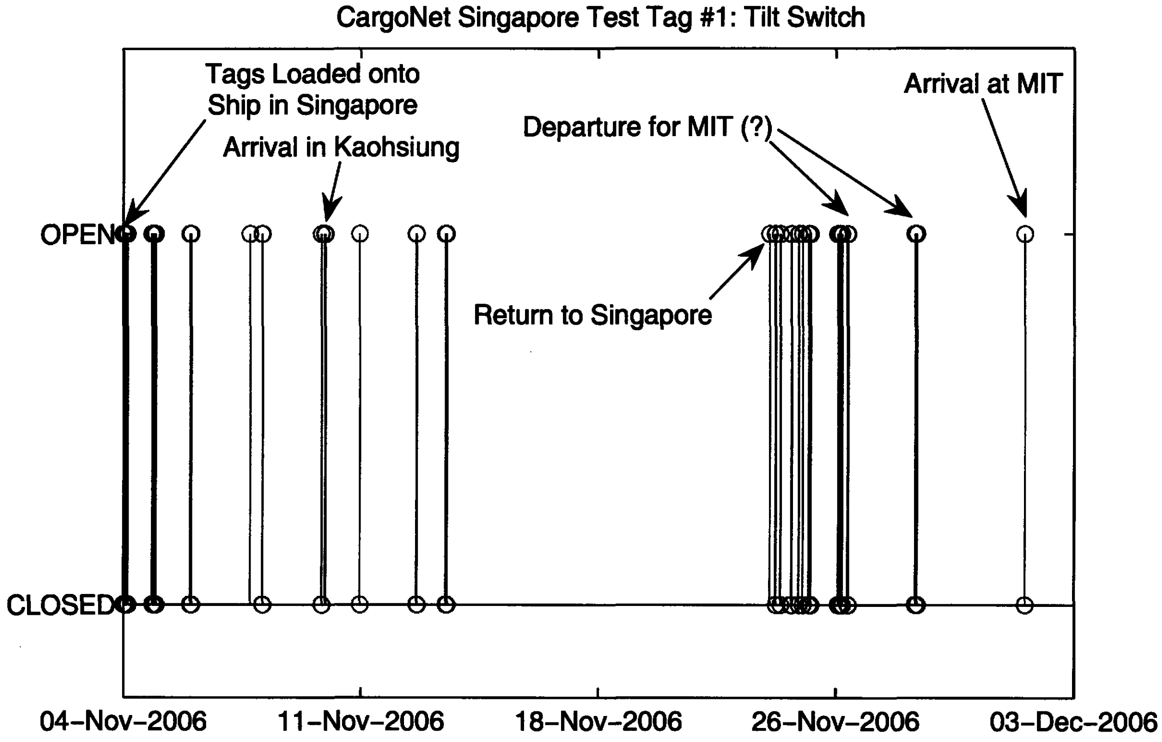


Figure 4.9: The tilt switch acted like a shock sensor, opening after large vibrations.

4.1.6 Vibration Dosimeter

The vibration dosimeter was designed to log the small vibrations that, over time, can cause the loosening of screws and other damage to a shipment, by integrating the negative portion of the vibration waveform onto a low-leakage capacitor. The capacitor is periodically sampled and discharged, and a quickly rising voltage (as in the case of a big shock) can also trigger an interrupt. Figure 4.10 shows the raw dosimeter data, as collected by the ADC once per minute or more. One can clearly see spikes at times when large shocks occurred, which correspond to the tilt-switch records in Figure 4.9.

Looking closer at Figure 4.10, it becomes clear that the small-scale vibrations at the lower range of the scale exhibit diurnal cycles. These variations are proportional to temperature, and were most likely caused by temperature-dependent offset voltages in the integrator op-amp; these offsets (which will vary between tags) can be corrected by subtracting a temperature-dependent offset from the waveform during analysis, and eventually on board the tag during data collection. A more precise op-amp in the integrator could also lessen the magnitude of this error. Figure 4.11 shows the results of these corrections followed by

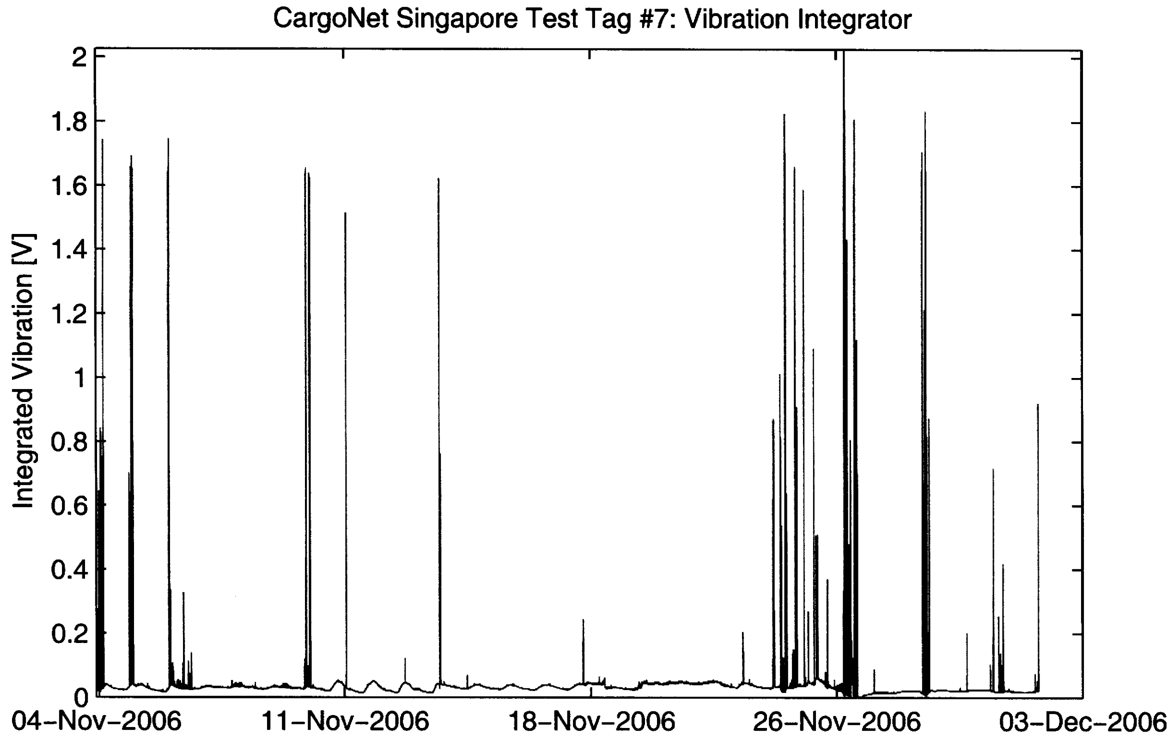


Figure 4.10: Raw vibration dosimeter data collected by Tag #7.

summing the vibrations collected over each interval, to better compare the impact of small versus large vibrations on the shipment. Because the vibratab shock sensor output voltage is proportional to the acceleration of the small mass mounted at the end of the sensor, and hence the force, the running sum of the dosimeter readings can be interpreted as a measure of the work being performed on the sensor by the vibrations.

4.1.7 Piezoelectric Microphone

The piezoelectric microphone was the only sensor that used dynamic thresholds during this test (the vibratab shock sensor worked too sporadically due to high thresholds, and the RF detector part of the tag was not populated), and it worked surprisingly well at picking up the large vibrations that also tripped the tilt switch and vibration dosimeter. Figure 4.12 shows the responses of the different sensors to shocks. Although the sensors seem to respond in unison to the large shocks experienced by the tag, there is considerable diversity to how they record the event. Capturing all aspects of the tag's environment therefore requires a multimodal sensor suite.

The effectiveness of the dynamic threshold mechanism is more difficult to evaluate, as

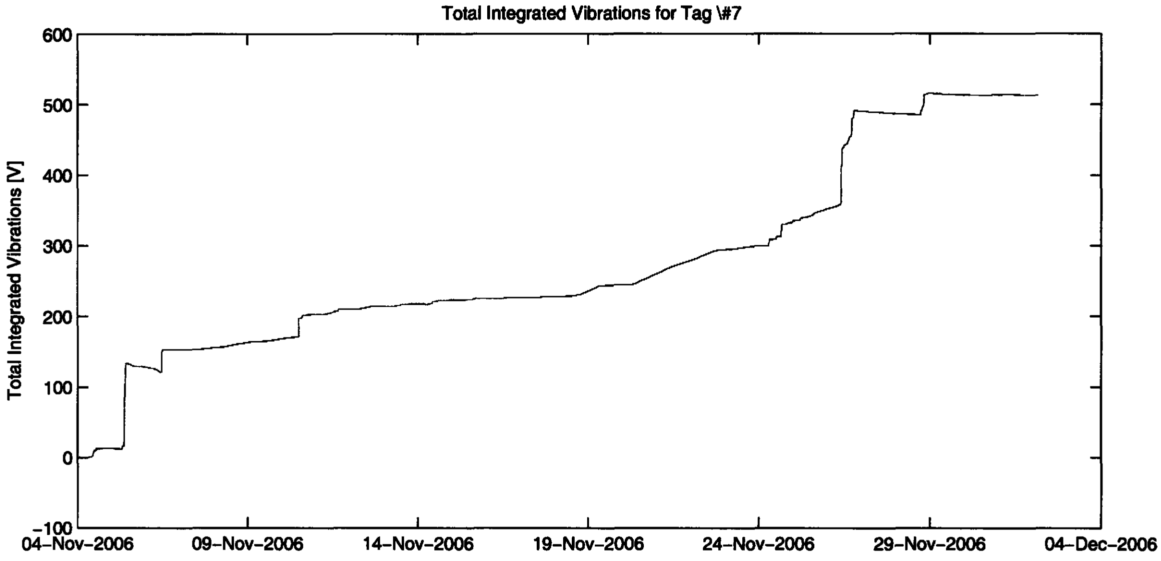


Figure 4.11: Running sum of the dosimeter vibrations collected over the course of the test, after temperature corrections.

there were no tags without dynamic thresholds implemented to serve as a control group. The tag logged decreasing sensitivity with each successive wakeup, but large shocks would still wake the microcontroller at the lowest sensitivity level. Figure 4.13 shows the problem from a different perspective by transforming the logged data to create an inverse—but easier to understand—dynamic threshold scenario. Instead of plotting a fixed threshold and a variable signal (the actual setup), the figure compares a variable threshold with a fixed signal. This theoretical signal is the microphone voltage that would have been present at the comparator had it not been attenuated by the digipot, and was obtained by measuring the microphone output voltage when driving various loads.

4.1.8 Temperature Sensors

Two sets of temperature readings were taken during the Singapore test, one using the SHT11 calibrated temperature sensor, and the other using the uncalibrated temperature sensor internal to the MSP430 ADC module. The external sensor has a resolution of 0.01 °C over the range of -40 to 123.8 °C, but is able to provide only ±2 °C accuracy between 0 and 85 °C [56]. The MSP430's internal temperature sensor maps the microcontroller's operating temperature range of -40 to 85 °C [65] to voltages between 0.85 and 1.3 V [67]. Assuming the internal 1.5 V reference is used for analog-to-digital conversion, the MSP430 internal

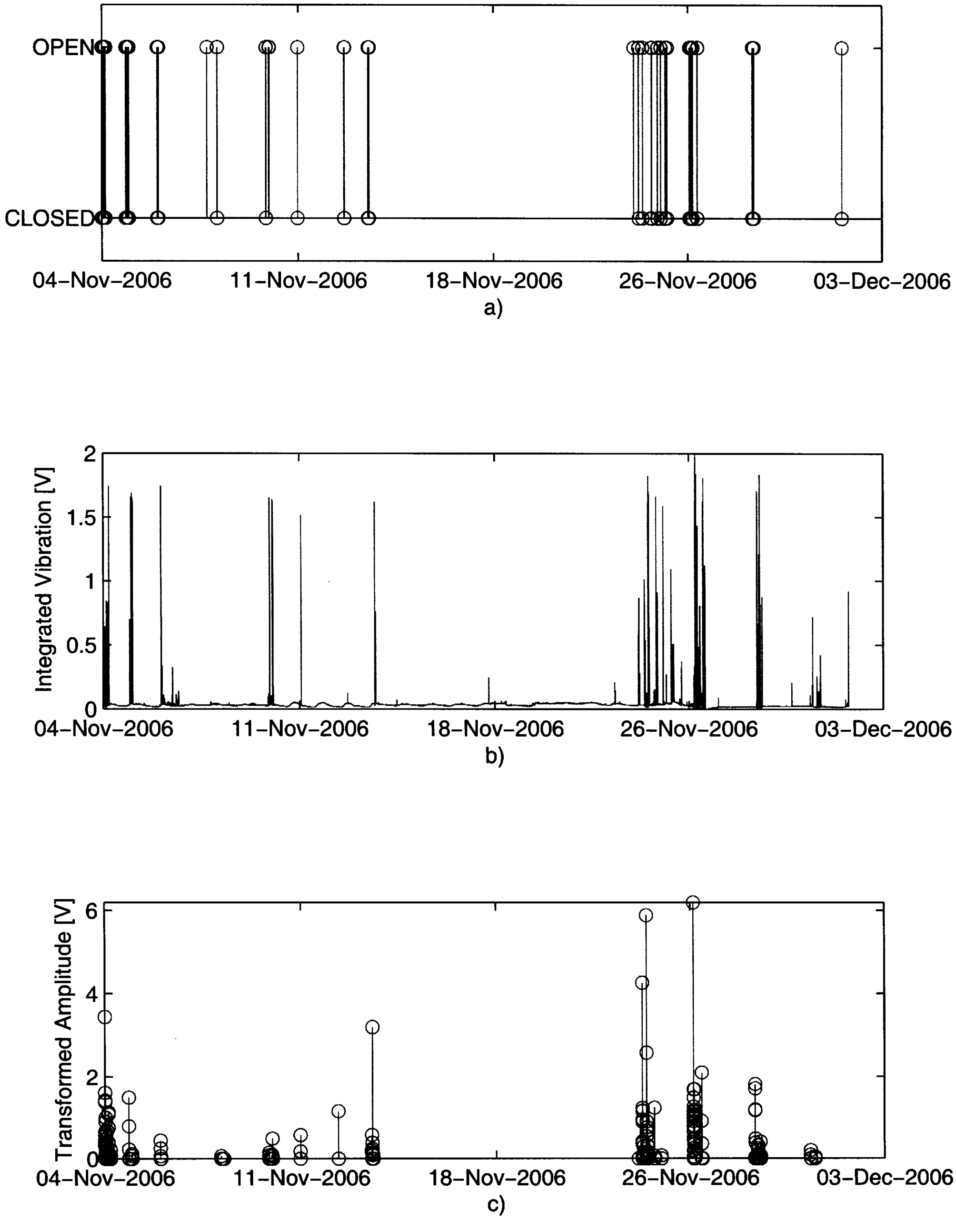


Figure 4.12: Comparison of the responses of the a) tilt switch, b) vibration dosimeter, and c) piezoelectric microphone, to large shocks.

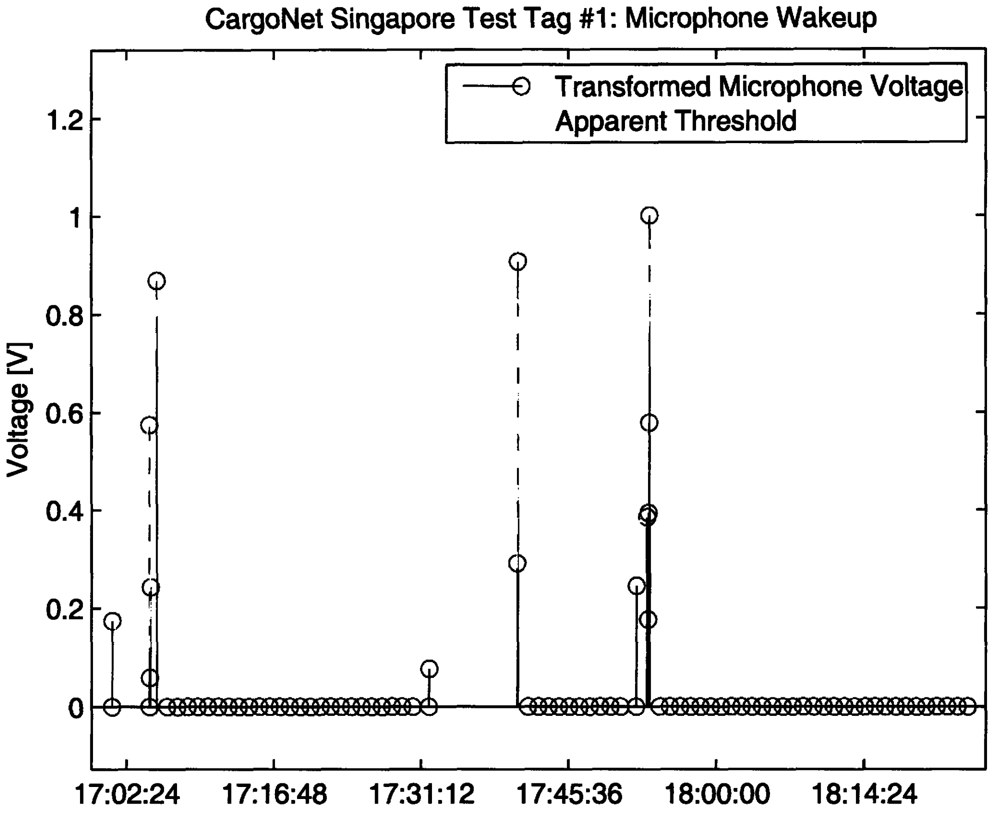


Figure 4.13: Illustration of dynamic threshold in microphone sensor. The plotted quantities were obtained by transforming the physical quantities of *voltage as measured across a variable load* and *fixed threshold* to the equivalent *voltage into a fixed load* and *variable threshold.*

temperature sensor has a resolution of

$$\begin{aligned} R &= \frac{T_{MAX} - T_{MIN}}{\frac{Range_{TEMP}}{Range_{ADC}} \cdot ...N_{SAMPLES}} && (4.5)\\ &= \frac{85\,^{\circ}\mathrm{C} - -4 - \,^{\circ}\mathrm{C}}{\frac{1.3\,\mathrm{V} - 0.85\,\mathrm{V}}{1.5\,\mathrm{V} - 0\,\mathrm{V}} \cdot 2^{12}} && (4.6)\\ &= 0.10\,^{\circ}\mathrm{C}. && (4.7) \end{aligned}$$

Furthermore, the MSP 430 datasheet projects[3] that a temperature of 0 °C will result in an output voltage of $986 \pm 5\%\,\mathrm{mV}$, but since the gain coefficient of the sensor is approximately 3.6 mV/ °C, this offset error can amount to a total offset up to

$$T_{OFF} = 5\% \cdot 986\,\mathrm{mV}/3.6\,\mathrm{mV}/\,^{\circ}\mathrm{C} = 13.7\,^{\circ}\mathrm{C}. \tag{4.8}$$

This offset in the internal temperature sensor readings can be seen in Figure 4.14, which shows the outputs of both the internal and external sensors. The offset is constant, with deviations less than ±1 °C, as can be seen in Figure 4.15, but varies noticeably from tag to tag. Notwithstanding its alarming magnitude, the offset can be easily corrected in a controlled environment during the commissioning or initial programming of the tag.

4.1.9 Summary of Singapore Tests

Although not all the sensors worked, and one tag failed to function entirely, the test aboard a cargo vessel as it travelled from Singapore to Kaohsiung helped in the discovery of bugs in hardware and firmware, and recorded baseline values that will be useful in tuning the platform to better measure events of interest.

4.2 Additional Laboratory Testing

After the tags returned from Singapore for analysis, it was clear that some of the sensors did not collect usable data (vibratab shock sensor), or did not work as expected (tilt switch). Furthermore, even the performance of some sensors that appeared to work correctly could not be evaluated, either because of incorrect calibration or because there was no control group against which the performance could be measured.

[3] "Not production tested, limits characterized" [65]

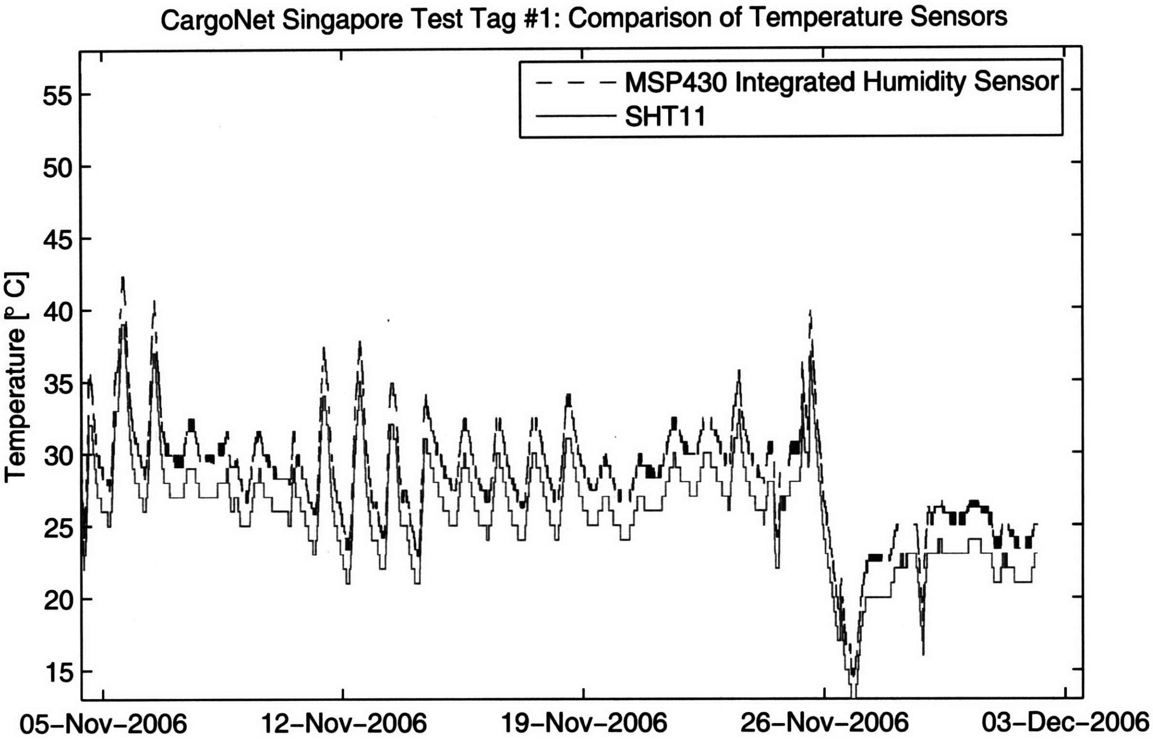


Figure 4.14: A comparison of temperature readings provided by an expensive calibrated sensor and the MSP430 internal temperature reference.

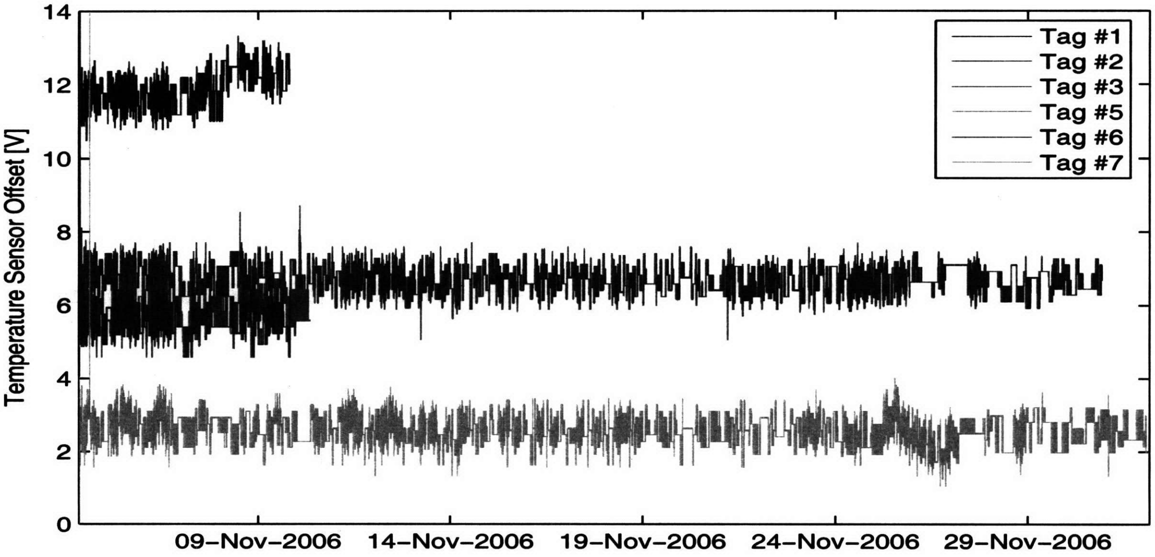


Figure 4.15: The uncalibrated (internal) temperature sensors exhibit fixed offsets with respect to the calibrated (external) sensor readings. These offsets vary from tag to tag.

The problems with non- or malfunctioning sensors such as the SHT11 humidity sensor were quickly corrected in firmware, as were the battery discharge problems described in Section 4.1.4, caused by remnants of debug code. Some other sensors, such as the microphone, functioned more or less correctly: they logged data that matched expectations and that correlated with data collected through additional sensing modalities. These data, collected under realistic conditions, can serve to improve the performance of these sensors through adjustments to thresholds, gains, and similar parameters. Even so, the impact of any tuning cannot be evaluated without additional tests, and these should involve a control of some sort: for example, the performance of the free internal temperature reference was compared against that of the expensive SHT11 to determine whether it could provide comparable performance. This sort of test needs to be performed for each of the tag's sensors, to evaluate the effectiveness of the hardware and firmware design.

4.2.1 Low-Cost Humidity Sensor Calibration

While the microcontroller's internal temperature reference was evaluated against the SHT11 during the course of the test, no such comparison was possible for the low-cost humidity sensor, as the SHT11 humidity sensor did not log any data due to a timing error in the firmware. As soon as the error was corrected, additional tests were performed in the laboratory.

To subject the sensors to a wide range of temperatures and relative humidities the tag under test was placed inside a ceramic container, which was packaged in open-cell foam. The ceramic walls of the container distributed the heat evenly throughout, while the foam kept the temperature and humidity stable enough for measurements. The sensors were subjected to various conditions by pouring hot water inside the container and heating it with hot air.

As was described in Sections 2.4.4 and 3.3.2, the low-cost humidity sensor operates by driving the resistive sensing element with a 100 Hz, 3 V_{P-P} square wave and measuring the logarithm of the current through the sensor, which should be proportional to relative humidity (RH). These are sampled every 300 μs, resulting in 16 samples over each half-period of the drive waveform. The microcontroller then calculates the series average and maximum, which are saved in memory.

Figure 4.16 shows the results of a typical test performed at room-temperature (21 °C–

23 °C). As can be seen in the second part of the graph, where the maximum and average low-cost humidity sensor readings are plotted against the SHT11 relative humidity, the maximum reading provides higher resolution (steeper slope) than the average reading at low humidities. Because of this advantage, the maximum will be the only reading considered from now on[4].

A second feature of note in Figure 4.16 is the quadratic nature of the relationship between the low-cost humidity sensor readings and the RH, as measured by the SHT11. The SHT11 sensor, however, does not return the RH value directly: its readout must instead be converted using a formula that compensates for its own *quadratic* relationship to relative humidity [56]. The nonlinearity exhibited in Figure 4.16 is therefore not unusual nor particularly burdensome, as the microcontroller would have to perform a linearization calculation with the more expensive sensor as well.

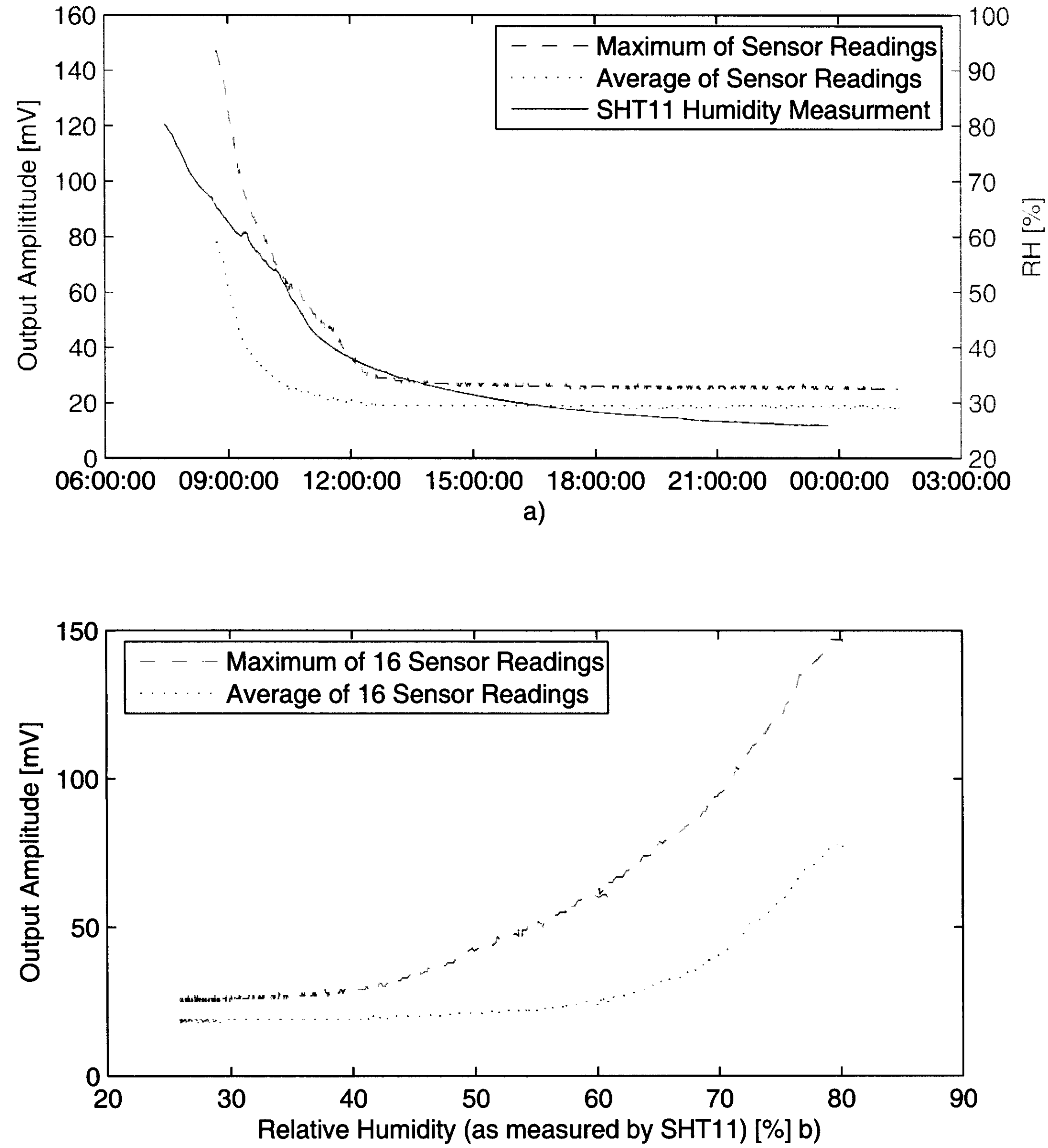
4.2.2 Humidity Sensor Temperature Dependence

Section 4.2.1 mapped the output of the low-cost humidity sensor to relative humidity, as measured by the calibrated SHT11 humidity sensor, but only at room temperature. Several tests were performed to calibrate the sensor's response at different temperatures and to evaluate the effect of the tendency of diodes in the linearizing circuit to compensate for the temperature dependence of the sensor element, as explained in Section 2.4.4.

Unfortunately, because of the lack of a proper test chamber, the calibration data obtained were ambiguous. The tag was placed in a ceramic container containing a small amount of liquid water and insulated from three sides. The container was heated from the outside with a hot-air gun, causing the air inside the container to heat, and some of the water to vaporize. However, because this test was performed manually, without closed-loop feedback adjusting temperature and humidity, the results were sparse. Furthermore, the sudden increases in temperature and humidity from the hot-air gun were not uniform throughout the container, resulting in different conditions at the two sensors, an effect that exhibited itself as hysteresis in the ensuing data.

A more thorough analysis with accurate test equipment would need to be conducted to obtain results similar to those presented in Figure 2.12. This data could then be used to

[4] Because the "resistive" humidity-sensing element is actually a resistance in parallel with a capacitance, the current waveforms are decaying exponentials, so the first reading will invariably be the maximum. The 15 extraneous samples were removed from later revisions of the code, reducing complexity and power consumption.



a)

b)

Figure 4.16: The output of the low-cost humidity sensor is sampled numerous times during each reading. The maximum and average amplitudes of those readings during a room-temperature test are shown in (a), with the low-cost, uncalibrated humidity sensor output voltage on the left axis and RH, as measured by the SHT11, on the right. The low-cost sensor readings are plotted directly against the SHT11-measured relative humidity in (b). Note the seemingly quadratic relationship of low-cost humidity sensor amplitude to RH.

calibrate the low-cost humidity sensor and use it exclusively in place of the SHT11, greatly decreasing the cost of the CargoNet tag. As mentioned in Section 2.4.4, resistive humidity sensors are interchangeable, so this calibration would only have to be performed once for each manufacturing run.

4.3 Express Courier Service Test

As mentioned in Section 1.5, the logistics company DHL has successfully tested passive temperature-sensing RFID tags for pharmaceutical cold chain management, and it is inevitable that sensate tags will eventually be added by courier services to their list of consumer express delivery offerings. To resolve some of the questions left unanswered during the Singapore test and also to experiment in this space, several CargoNet tags were packaged and sent across the United States, from Cambridge, Mass., to San Francisco, Calif., via DHL's overnight service.

4.3.1 Experimental Setup

Because of the short time frame of this test (less than 48 hours), the measured battery voltage was unlikely to decrease appreciably, so a dedicated circuit was developed to directly measure tag current consumption. Developed by Mark Feldmeier at the MIT Media Laboratory, and based on [23], this current meter achieves better than 0.5% accuracy over seven decades of current by using a running integral of tag supply current to gate periodic pulses [17]. These pulses are counted by a microcontroller (a repurposed CargoNet tag), and the number of pulses collected over a period of time (500 ms) corresponds (at 18.329 nJ/pulse) to the total energy consumed by the tag during that time. Further, dividing by the length of the period results in the average power consumption over that period.

All the sensors except for the phototransistor and reed switch were initialized and running on the device-under-test (DUT), though the firmware had been slightly changed to reflect the lessons learned as a result of the Singapore test. In addition to the DUT, current measuring circuit, and current-pulse logging tag, an additional CargoNet tag was included in the shipment, programmed to sample the state of the microphone, vibratab, and tilt switch at 2 Hz. This record would be used to confirm the ability of the quasi-passive wakeup and dynamic threshold mechanisms to sufficiently monitor those transient phenomena while conserving power.

4.3.2 Test Results

According to DHL's tracking Web site [13], the package containing the tags was picked up by DHL at 18:32 on 1 February 2007 and arrived in San Francisco, by way of Ohio, at 8:20 local time on 2 February 2007. Because delivery was attempted without success, the package stayed in the DHL delivery van until it was picked up at the depot at 17:52. Inside the package was another box with a return label already affixed, so the the inner box was removed and put back into the system for its return voyage to Cambridge. Before it left the DHL facility, however, some event incapacitated the DUT, causing it to draw over 90 mA during one 500 ms interval!

The cause of the failure is still unclear, but the analog output of the vibration dosimeter, which directly drives one of the microcontroller's I/O pins, stayed latched at the 1.5 logic threshold, consuming 22.8 μA of steady-state class AB current. Because the current-meter circuit supplying the DUT has a high output impedance, an extended period of high current draw could have brought the supply voltage below the threshold of operation of some components of the tag, in particular the LPV511 micropower op-amp inside the vibration dosimeter, which could have malfunctioned, letting its output float to the microcontroller's logic threshold, thereby reinforcing the malfunction. Whatever the cause of the malfunction, the DUT operated satisfactorily during the first half of the test. Its current consumption, averaged over 10-minute intervals to remove variations caused by once-per-minute writes to flash, can be seen in Figure 4.17.

The average current consumption prior to failure was 7.88 μA—slightly less than the laboratory estimate of 9 μA, most likely due to improvements in the efficiency of humidity sampling code—resulting in an average power consumption of 23.7 μW. These results and other statistics are summarized in Table 4.3.

	Current [μA]	Power [μW]
Average	7.88	23.7
Maximum	556	1670
Standard Dev.	28.3	85.2

Table 4.3: Current and power consumption of CargoNet during DHL test. All statistics calculated from data sampled at 500 ms.

As was shown in Figure 4.17, periods of higher activity noticeably increased the current consumption of the tag. Part of this increase was due to the unscheduled wakeups caused by

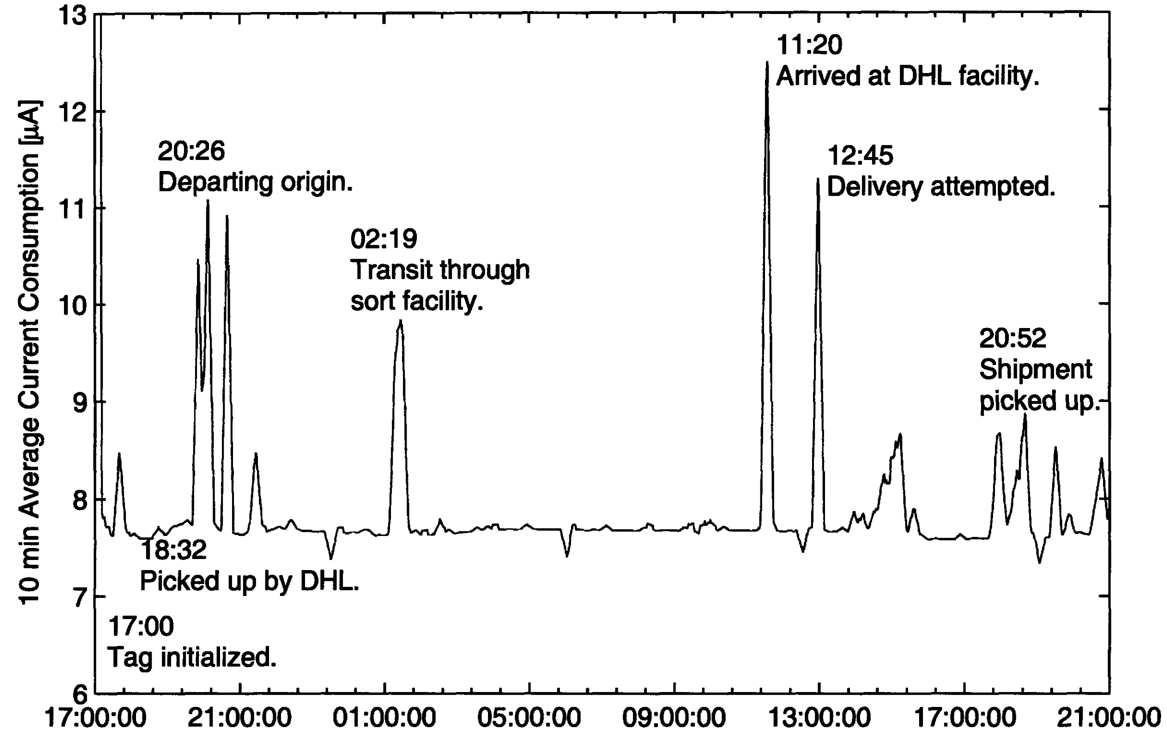


Figure 4.17: Average current consumption of a CargoNet tag during an overnight journey from Cambridge, Mass. to San Francisco, Calif. Periods of higher current clearly correspond to episodes in its history during which it was handled or loaded from one conveyance to another [13].

microphone, vibratab, and tilt switch sensor interrupts which woke the microcontroller and in some cases prompted a write to flash. However, these wakeups should have been limited by dynamic thresholds in the case of the microphone and vibratab, and could be completely eliminated in the case of the tilt switch if it were mounted to the underside of the tag. As can be seen from Figure 4.18, the current consumption of the tag rises continuously during the course of a one-minute polling period, doubling or even tripling from approximately $4\,\mu$A to $12\,\mu$A. This increased current corresponds to the output of the vibration dosimeter, as vibrations are continuously integrated between resets. This effect is most likely due to class AB conduction in the microcontroller's pin I/O buffer as the pin voltage is slowly increased from a logical LOW of 0 V. The MSP430 User Guide [67] specifically warns against using digital I/O pins with slowly-changing signals, and if the increased power consumption is a high enough concern in additional tests, another LTC1540 nanopower comparator can be inserted in the signal path between the dosimeter output and the microcontroller's I/O pin.

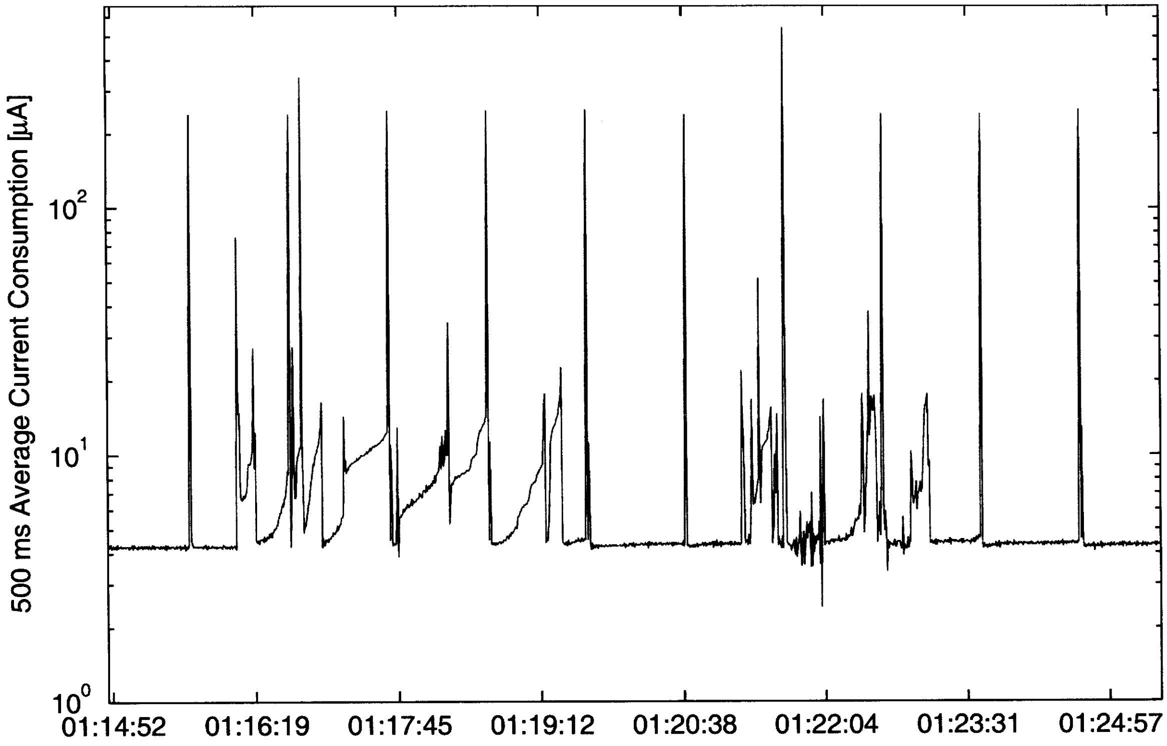


Figure 4.18: Increased current consumption caused by vibration dosimeter output rising between resets.

The vibration dosimeter is a micropower sensor that is continuously enabled, so that it can record the smallest of vibrations. The piezoelectric microphone and vibratab shock sensor respond to signals powerful enough to trigger the sensors and wake up the microcontroller. But a few questions remain: are there intermediate signals that are too powerful for the dosimeter, that overwhelm it, causing large current draws, but that are insufficiently strong to trigger the quasi-passive sensors? Also, what are the effects of the dynamic thresholds; can they still capture the important events while limiting power consumption? Although these questions will require more testing to answer conclusively, some inferences may be drawn from the results of this test.

Figures 4.19 and 4.20 show the efficacy of using quasi-passive wakeup with dynamic thresholds versus periodic sampling when logging sudden shocks and sounds. As can be seen in the figures, the quasi-passive wakeup scheme is able to capture the same events with often a better level of detail than the periodic sampling method, while consuming orders of magnitude less power. In all fairness, the 2 Hz sampling frequency was chosen based on the available storage capacity, and is often too slow to capture the events of interest. A more intelligent scheme, which samples at a higher frequency but logs only meaningful samples, could capture more data but would require even more power.

Finally, the DUT also monitored tilt, temperature, and humidity. The sensors for these modalities had worked previously during the Singapore tests, so their performance was not discussed here. The only exception was the SHT11 humidity sensor, which failed to work due to a firmware bug. This time the SHT11 operated correctly, but because it was Winter in the United States when the test was performed, the humidity stayed below 20 percent. As can be seen in Figure 4.16, however, the low-cost sensor's does not extend that low, so once again, its performance could not be evaluated in the field.

4.4 Summary and Future Work

The tests described in this chapter, performed at the MIT Media Laboratory and during journeys across the South China Sea and the continental United States, proved the efficacy of the hardware and firmware approaches described in Chapters 2 and 3. Although several portions of the hardware and software have been improved, the system still requires more testing and development before a full deployment. The best values at which to set the thresholds and their dynamic ranges have not been selected. The backoff protocol has not been conclusively tested, either, to ensure that a high proportion of interesting events are captured.

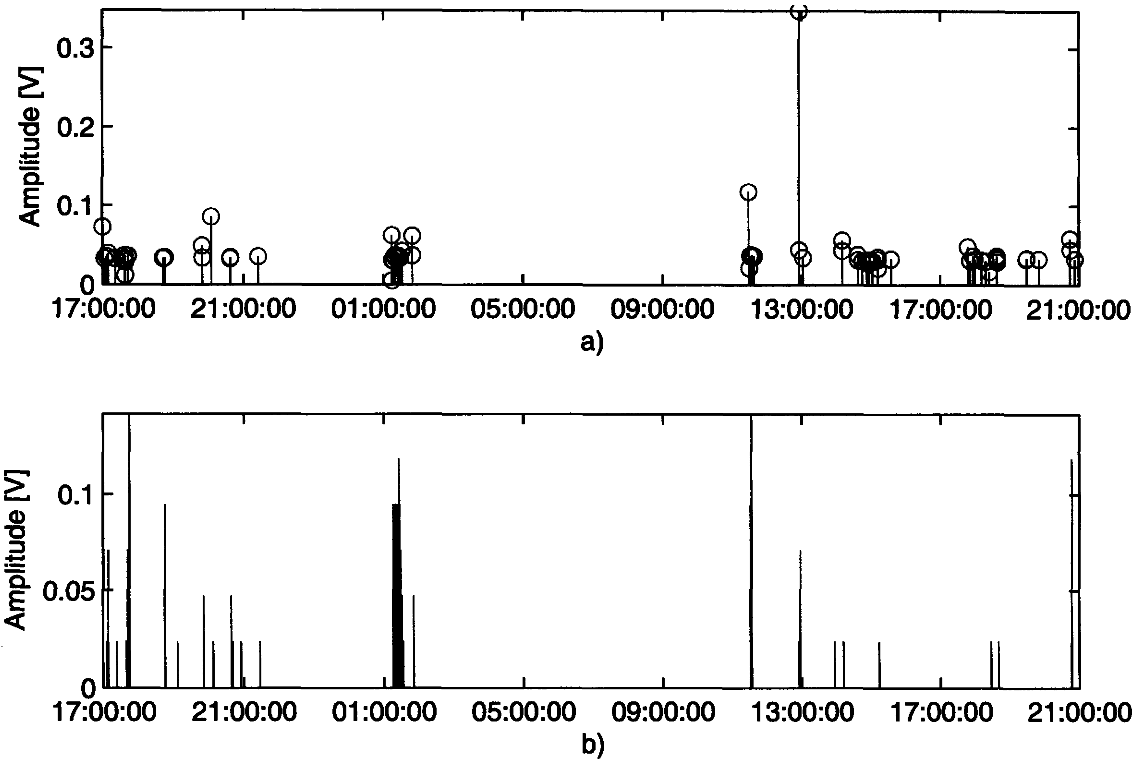


Figure 4.19: Comparison of vibratab shock sensor readings collected with quasi-passive wakeup with dynamic thresholds (a) to those collected by sampling the sensor every 500 ms (b).

Communication links between the tags and readers and other tags, both the high-bandwidth data links, and the short-distance RF wakeup, still remain to be integrated into the system and tested. A prototype of the reader/interrogator has recently been completed [38], and developing strategies for power-efficient and useful communication with the device will be topic for future work.

Additional testing and development must still be performed before the tags can be widely deployed. The on-tag sensors must be fully characterized in the laboratory to accurately gauge shocks and vibrations, as well as ambient light and humidity. A calibration procedure for all the sensors would need to be developed to guarantee the desired accuracy without drastically increasing the cost of using the tags.

Also of interest is the field performance of the lithium coin cell battery—which has not yet been tested—specifically its ability to source sufficient current and its longevity under the demanding temperatures of the global supply chain. But even if the battery proves to

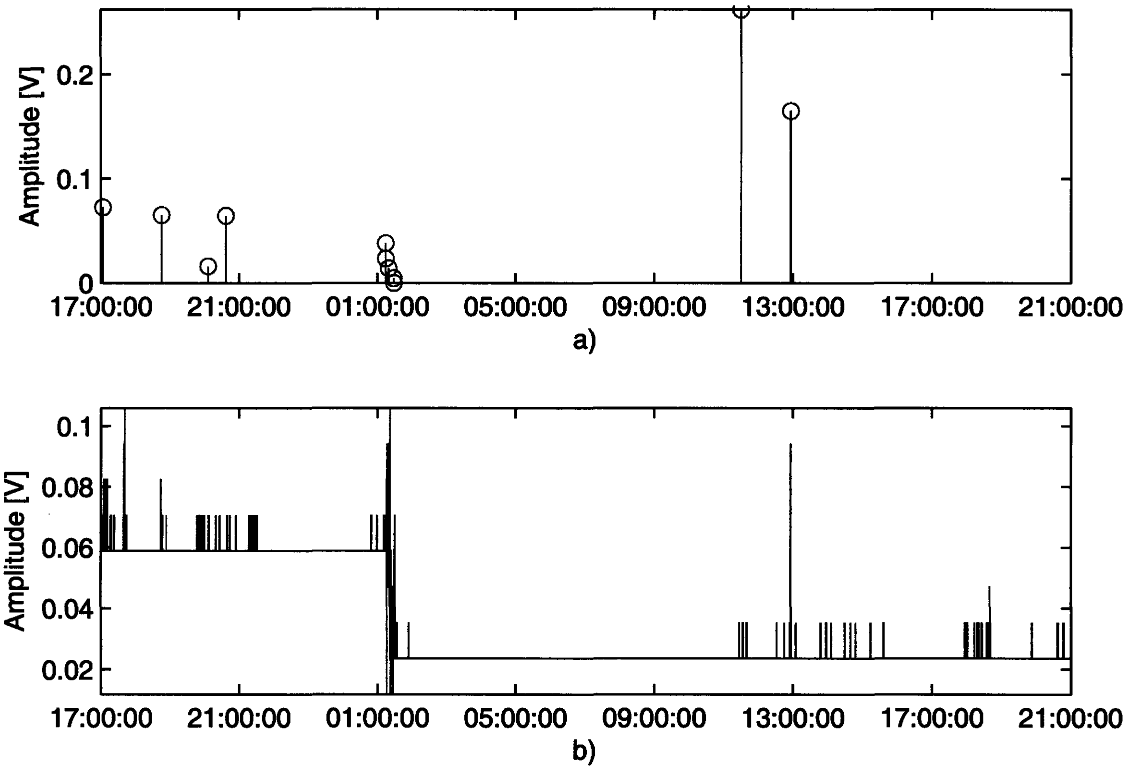


Figure 4.20: Comparison of piezoelectric microphone readings collected with quasi-passive wakeup with dynamic thresholds (a) to those collected by sampling the microphone every 500 ms (b). The origin of the offset at the beginning of the data series in (b) is unknown.

be a good fit, the power can still fail. Mechanisms ensuring recovery after crashes and data integrity under difficult conditions would also need to be developed and tested.

Hence, though the system would benefit from another cycle of design and testing, a great beginning has been forged through the development of the CargoNet platform and its successful operation in various contexts.

Chapter 5

Conclusion

The previous chapters have described the development of the CargoNet system of micropower active tags, from inspiration, through hardware and firmware design, to testing in realistic settings aboard ships and aircraft. This work fills an important niche in the current efforts to add visibility to the supply chain, as no active tags that contain multiple sensing modalities, while keeping price and power consumption low, have yet been developed. Currently available active tags are too application-specific or too powerful—and therefore power-hungry and expensive—to be broadly used at the crate and case level. Passive tags, on the other hand, still lag behind active tags in the number of sensing modalities they can provide. Furthermore, they suffer from the fundamental disadvantage of sensing, processing, and communicating only in the presence of a reader that provides power. Thorough logs of the state of the tagged object, like the ones analyzed in Chapter 4, are instead reduced to a handful of samples collected at checkpoints.

Through the framework of quasi-passive wakeup (developed in Section 2.4.1), which was further extended with dynamic thresholds, multiple spontaneous stimuli could be efficiently captured by the tag. Micropower sensors and interface circuits were a crucial component of this framework, as described in the remaining parts of Section 2.4. These same circuits were also the reason why the CargoNet tag was able to meet the low-cost and low-power requirements. In particular, the essentially free humidity sensor (described in Section 2.4.4), will greatly lower the price barrier of environmental sensing, once it is sufficiently tested and characterized.

The performance of dynamic thresholds depends to a great extent on the firmware that controls them, as does the performance of all sampled sensors, which rely on the firmware to manage their duty cycles. Chapter 3 described the implementation of a modular and interrupt-driven firmware that allowed the tag to be easily reconfigured for the numerous tests. This firmware enables a fully modular architecture that allows users of the system to use only the sensors needed to monitor a particular shipment, or alternatively, to repurpose the tag for an entirely different application.

The performance of the CargoNet tags was put to the test in Chapter 4, when they were sent to Singapore to monitor the conditions inside a container aboard a cargo ship. The test results (described in Section 4.1) revealed a number of improvements that needed to be made—mostly to the firmware—for the tags to work reliably. Although some of the failures experienced during the Singapore test were the result of easily-fixable bugs, others, such as the development of a more careful flash data-transfer mechanism, required additional development. The changes were incorporated into the tag firmware and tested again in the laboratory and by airmailing a tag across the United States, finally demonstrating 23.7 μW average power consumption of the CargoNet tags while capturing important environmental parameters such as shock and vibration.

This is the current state of CargoNet active tag system. Many challenges remain, but the payoff promises to be great: once they are fully operational and working consistently over long stretches of time and under a wide range of conditions, CargoNet active tags will be the bridge between RFID and wireless sensor networks, providing much-needed visibility to the global supply chain and paving the way towards inexpensive sensing and communication for everyday objects.

Appendix A

Board Layouts and Schematic

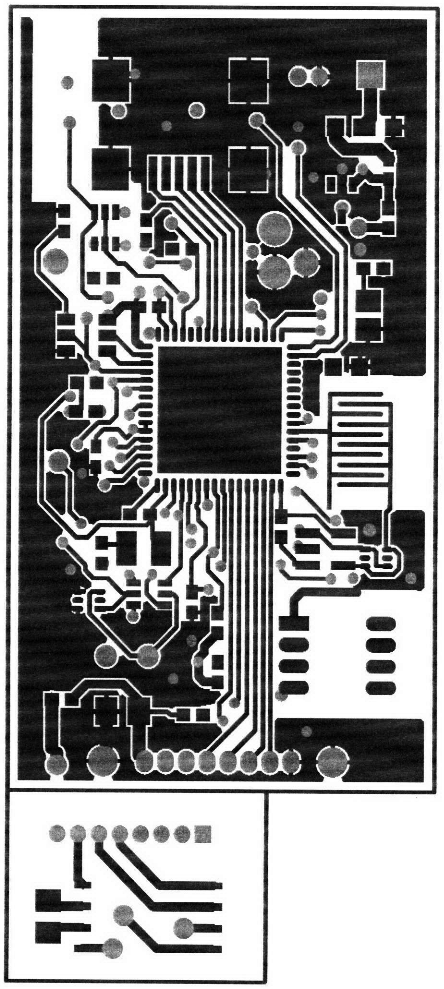


Figure A.1: CargoNet version 4 PCB top layer. (Layout includes SPI flash daughter board.)

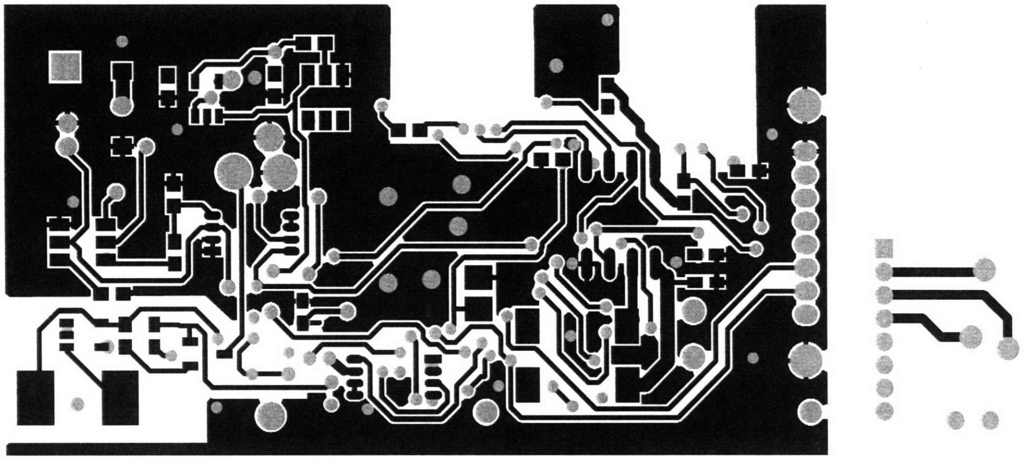


Figure A.2: CargoNet version 4 PCB bottom layer.

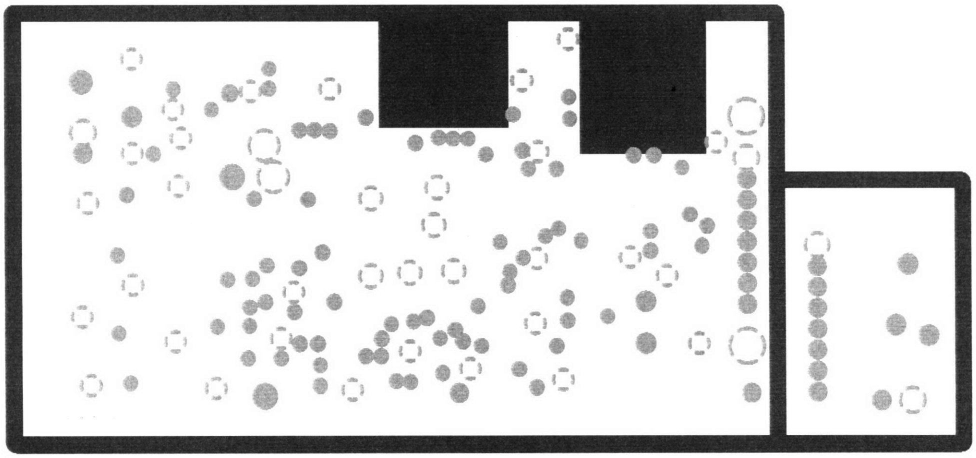


Figure A.3: CargoNet version 4 PCB internal ground plane (negative).

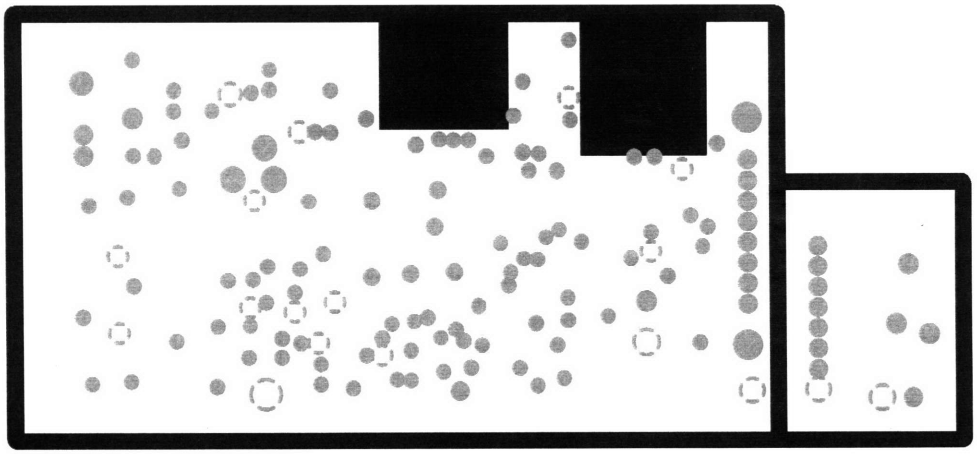


Figure A.4: CargoNet version 4 PCB internal V_{CC} plane (negative).

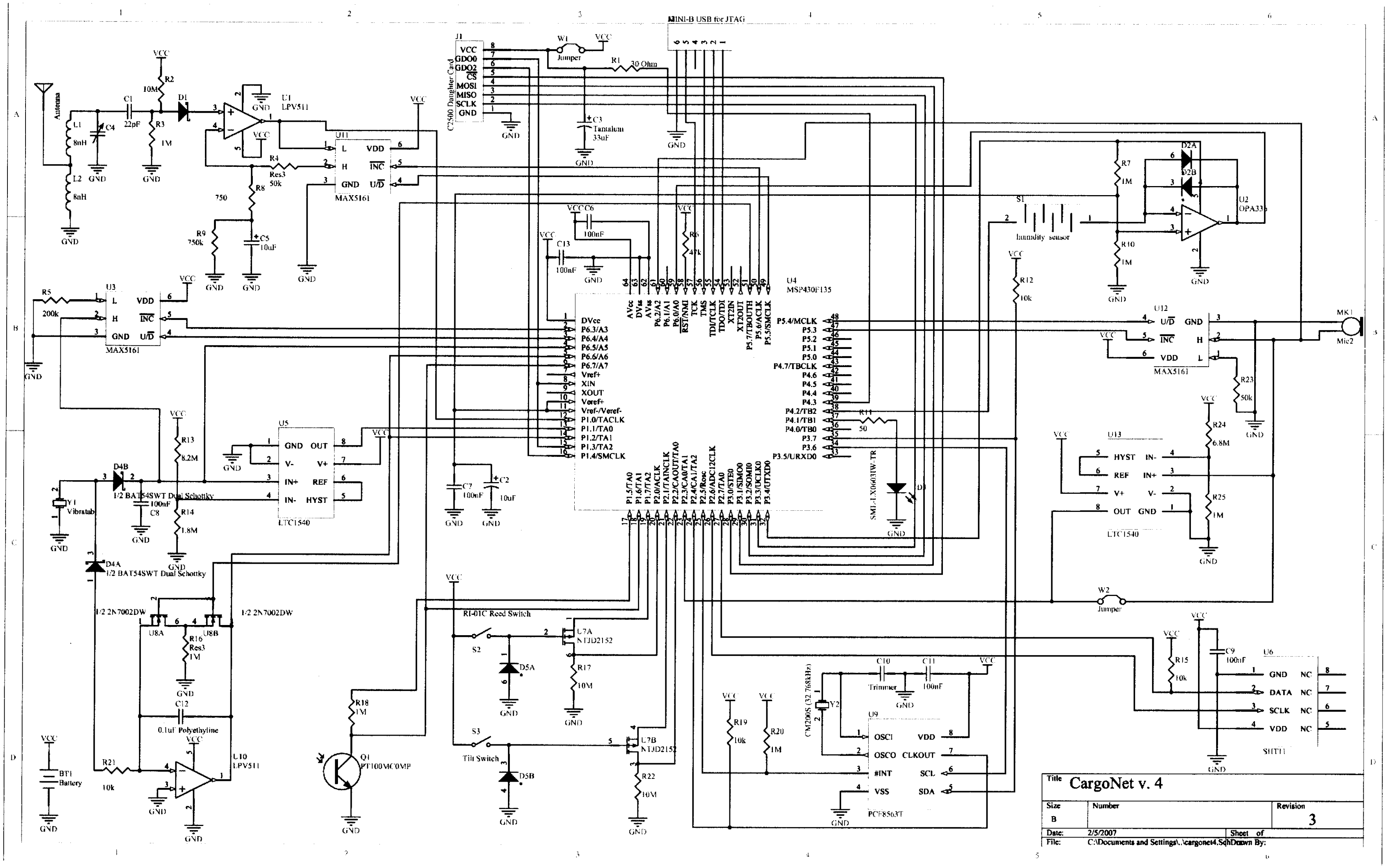


Figure A.5: CargoNet version 4 schematic diagram.

Bibliography

[1] Atmel. AT45DB081B: 8-megabit 2.5-volt only or 2.7-volt only Dataflash. Datasheet, September 2005. http://www.atmel-grenoble.com/dyn/resources/prod_documents/doc2225.pdf.

[2] Gerardo Barroeta Perez, Mateusz Malinowski, and Joseph A. Paradiso. An ultra-low power, optically-interrogated smart tagging and identification system. In *Fourth IEEE Workshop on Automatic Identification Advanced Technologies (AutoID2005)*, Buffalo, NY, October 2005.

[3] Dave Blanchard. RFID at the gates. *Logistics Today*, 45(9):1–16, September 2004.

[4] Bureau of Customs and Border Protection. Free and secure trade (FAST) implementation on the US/Mexico border. Press release, 4 December 2003. http://www.dhs.gov/xnews/releases/press_release_0309.shtm.

[5] George Cavage. Via email, 13 November 2006.

[6] Chipcon. Single chip low cost low power RF-transceiver [CC2500]. Datasheet, 2005. http://www.chipcon.com/files/CC2500_Data_Sheet_1_1.pdf.

[7] Chipcon. 2.4 GHz IEEE 802.15.4 / ZigBee-ready RF transceiver [CC2420]. Datasheet, April 2006. http://www.chipcon.com/files/CC2420_Data_Sheet_1_4.pdf.

[8] Crossbow Technology. Mica2 wireless measurement system. Datasheet, 5 January 2006. http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/MICA2_Datasheet.pdf.

[9] Crossbow Technology. TelosB mote platform. Datsheet, January 2006. http://xbow.com/Products/Product_pdf_files/Wireless_pdf/TelosB_Datasheet.pdf.

[10] Crossbow Technology. TelosB order form. Web page, 5 January 2007. http://www.xbow.com/Products/productdetails.aspx?sid=175.

[11] Department of Homeland Security: Bureau of Customs and Border Protection. Required advance electronic presentation of cargo information. *Federal Register*, 68(141):43574–43606, 23 July 2003. Proposed rule.

[12] Department of Homeland Security: Bureau of Customs and Border Protection. Required advance electronic presentation of cargo information. *Federal Register*, 68(234):68139–68177, 5 July 2003. Final rule.

[13] DHL. Tracking results detail for 8968906612. Web site, 3 February 2007. http://track.dhl-usa.com/TrackByNbr.asp?nav=Tracknbr.

[14] Digikey.com. Web site, January 2007. http://www.digikey.com/.

[15] Leslie Downey. International cargo conundrum. *RFID Journal*, 6 February 2006. http://www.rfidjournal.com/article/articleprint/2120/-1/82/.

[16] Erlich Industrial Development Corporation. Resistive humidity sensor board. Web site, 11 January 2006. http://www.eidusa.com/Interface_Boards_Humidity_Sensors.htm.

[17] Mark Feldmeier. Personal conversation., 29 January–3 February 2007.

[18] Mark Feldmeier, Mateusz Malinowski, and Joseph A. Paradiso. Large group musical interaction using disposable wireless motion sensors. In *Proceedings of the 2002 International Computer Music Conference (ICMC)*, pages 83–87, San Francisco, September 2002. International Computer Music Association.

[19] Mark Feldmeier and Joseph A. Paradiso. An interactive music environment for large groups with giveaway wireless motion sensors. *Computer Music Journal*, 31(1):50–67, Spring 2007.

[20] K. Finkenzeller. *The RFID Handbook—Radio-Frequency Identification Fundamentals and Applications*. John Wiley & Sons, Ltd., New York, 2000.

[21] Paul Horowitz and Winfield Hill. *The Art of Electronics*. Cambridge University Press, Cambridge, UK, second edition, 1989.

[22] Sasha Jevtic, Mat Kotowsky, Robert P. Dick, Peter A. Dinda, and Charles Dowding. Lucid dreaming: Reliable analog event detection for energy-constrained applications. In *Information Processing in Sensor Networks (IPSN/SPOTS) '07*, 2007. Forthcoming.

[23] Xiaofan Jiang, Prabal Dutta, David Culler, and Ion Stoica. Micro power meter for energy monitoring of wireless sensor networks at scale. In *Information Processing in Sensor Networks (IPSN/SPOTS) '07*, 2007. Forthcoming.

[24] Ajit Kambil. RFID: Retail's 800-pound gorilla. *Logistics Today*, 44(10):34–37, October 2003. Supplement.

[25] Kobitone Audio Company. 25LM025 crystal microphone. Datsheet, 14 July 2005. http://www.mouser.com/catalog/specsheets/KT-400026.pdf.

[26] Kyocera. Piezoelectric shock sensors. Datsheet. http://global.kyocera.com/prdct/electro/pdf/piezo/shock-ce.pdf.

[27] Mathew Laibowitz and Joseph Paradiso. Wireless wearable transceivers. *Circuit Cellar*, (143):28–30, February 2004. http://www.media.mit.edu/resenv/pubs/papers/2003-12-circuit_cellar5.1.pdf.

[28] Joshua Lifton, Mark Feldmeier, Yasuhiro Ono, Cameron Lewis, and Joseph A. Paradiso. A platform for ubiquitous sensor deployment in occupational and domestic environments. In *Information Processing in Sensor Networks (IPSN/SPOTS) '07*, 2007. Forthcoming.

[29] Linear Technology. LTC1540: Nanopower comparator with reference. Datasheet, 7 December 2004. http://www.linear.com/pc/downloadDocument.do?navId=H0,C1,C1154,C1004, C1139,P1593,D1777.

[30] Haiyang Liu, Abhishek Chandra, and Jaideep Srivastava. eSENSE: Energy efficient stochastic sensing framework for wireless sensor platforms. In *Information Processing in Sensor Networks (IPSN/SPOTS) '06*, Nashville, Tenn., 19–21 April 2006.

[31] Kent Lundberg. *Become One with the Transistor.* Number 2.2.20. Self-published, 5 February 2003. Notes for MIT course 6.301: Solid State Circuits.

[32] Hongshen Ma and Joseph A. Paradiso. The FindIT Flashlight: Responsive tagging based on optically triggered microprocessor wakeup. In *UBICOMP 2002*, pages 160–167, Berlin, 2002. Springer Verlag.

[33] Mateusz Malinowski. Optical wakeup of micropower tags for object location and identification. Advanced undergraduate project, Massachusetts Institute of Technology, Cambridge, Mass., May 2005.

[34] Maxim Integrated Products. MAX5160/MAX5161 low power potentiometers. Datasheet, February 2001. http://www.maxim-ic.com/getds.cfm?qv_pk=1976.

[35] Measurement Specialties. Measurement specialties 2006 capabilities brochure. Brochure, 21 April 2006. http://www.meas-spec.com/myMeas/images/catalog/MEASbro_14_FINAL2.pdf.

[36] Measurement Specialties, Inc. Minisense 100 vibration sensor. Datasheet, April 2006. http://www.meas-spec.com/myMeas/download/pdf/english/piezo/minisense100_vibration.pdf.

[37] Roger Morton. RFID compliance: Year two. *Logistics Today*, 47(1):6–9, January 2006.

[38] Matthew Moskwa. A base station for data collection for sensor-equipped active RFID tags. Undergraduate advanced project, Massachusetts Institute of Technology, Cambridge, Mass., February 2007.

[39] Moteiv Corporation. Moteiv Corporation introduces first intelligent wireless system for first responders. Press release, 27 September 2006. http://www.moteiv.com/pr/2006-09-27-fire.php.

[40] Moteiv Corporation, 55 Hawthorne St., Suite 550, San Francisco, CA 94105. *Tmote Invent User Guide*, 1.0 edition, 27 February 2006.

[41] Moteiv Corporation. Tmote Sky order form. Web page, 5 Jan 2007. http://www.moteiv.com/xcart/product.php?productid=2.

[42] Newark InOne. Web site, January 2007. http://www.newark.com/jsp/home/ homepage.jsp.

[43] Mary Catherine O'Connor. Toronto-area hospital kicks off asset-tracking pilot. *RFID Journal*, 26 November 2006. http://www.rfidjournal.com/article/articleview/2848/.

[44] Panasonic. Selecting the right lithium battery: BR vs. CR chemistries. Application note, 15 January 2001. http://www.panasonic.com/industrial/battery/oem/images/pdf/Selecting%20BR%20or%20CR.pdf.

[45] Panasonic. Lihium coin type. Web site, 10 January 2007. http://www.panasonic.com/industrial/battery/oem/chem/lith/coin1.htm.

[46] Joe Pappalardo. Researchers seek the "perfect shipping container". *National Defense Magazine*, 90(624):30–31, November 2005.

[47] Joseph A. Paradiso, Ari Benbasat, and Mark Felmeier. General wireless sensor with low-power wake-up. Meeting notes, 29 November 2003.

[48] Matthai Philipose, Joshua R. Smith, Bing Jiang, Alexander Mamishev, Sumit Roy, and Kishore Sundara-Rajan. Battery-free wireless identification and sensing. *IEEE Pervasive Computing*, 4(1):37–45, January–March 2005.

[49] QUALCOMM. OmniTRACS mobile communication system. Web page, January 2007. http://www.qualcomm.com/qwbs/solutions/prodserv/omnitracs.shtml.

[50] Jim Rice, Mary Gibson, Mary Murphy-Hoye, and George Cavage. Intelligent containers: End-to-end security investment for supply network collateral benefit. Research proposal, Intel Corporation, 9 March 2006. Confidential.

[51] Helen Richardson. Globalization: What works today . . . *Logistics Today*, 45(6):2A–7A, June 2004.

[52] Tom Ridge. Statement of secretary Tom Ridge before the U.S. House of Representatives Select Committee on Homeland Security. Press release, 12 February 2004. http://www.dhs.gov/xnews/testimony/testimony_0019.shtm.

[53] Denes J. Roveti. Choosing a humidity sensor: A review of three technologies. *Sensors*, 1 July 2001. http://www.sensorsmag.com/sensors/article/articleDetail.jsp?id=322590.

[54] Savi Technology. Savi products and technology overview. Web page, 3 January 2006. http://www.savi.com/products/overview.shtml.

[55] Savi Technology. Savi SensorTag ST-676. Datasheet, 11 June 2006. http://www.savi.com/products/SensorTag_676.pdf.

[56] Sensirion. SHT1x/SHT7x humidity and temperature sensor. Datasheet, March 2006. http://www.sensirion.com/en/pdf/product_information/Data_Sheet_humidity_sensor_SHT1x_SHT7x_E.pdf.

[57] Sensitech. Coldstream infrastructure. Datsheet, March 2006. www.sensitech.com/applications/coldstream_pts/ColdStream_Infra_DataSheet.pdf.

[58] Yossi Sheffi. RFID and the innovation cycle. *The International Journal of Logistics Management*, 15(1):1–10, 2004.

[59] Joshua R. Smith, Alasnos Sample, Pauline Powledge, Alexander Mamishev, and Sumit Roy. A wirelessly powered platform for sensing and computation. In *Ubicomp 2006: Eighth International Conference on Ubiquitous Computing*, pages 495–506, Orange County, Calif., 17–21 September 2006.

[60] Claire Swedberg. Chicago Fire Dept. tests ZigBee-based RFID system. *RFID Journal*, 11 October 2006. http://www.rfidjournal.com/article/articleview/2717/1/1/.

[61] Claire Swedberg. PCTS, Radianse team for hospital tracking solution. *RFID Journal*, 1 Septmber 2006. http://www.rfidjournal.com/article/articleview/2632/.

[62] Claire Swedberg. DHL expects to launch "sensor tag" service by midyear. *RFID Journal*, 19 January 2007. http://www.rfidjournal.com/article/articleview/2986/.

[63] Safefreight Technology. EnCompass series. Datasheet, November 2006. http://www.safefreight.com/images/stories/documents/encompass.pdf.

[64] Texas Instruments. MSP430F11x2, MSP430F12x2 mixed signal microcontroller (Rev. D). Datasheet, 31 August 2004. http://www.ti.com/lit/gpn/msp430f1232.

[65] Texas Instruments. MSP430x13xx, MSP430x14xx mixed signal microcontroller. Datasheet, 3 June 2004. http://www.ti.com/lit/gpn/msp430f135.

[66] Texas Instruments. Msp430 ultra-low-power microcontrollers. Brochure, August 2006. http://www.ti.com/litv/pdf/slab034l.

[67] Texas Instruments. *MSP430x1xx Family User's Guide (Rev. F)*, 28 February 2006. http://www.ti.com/litv/pdf/slau049f.

[68] Perry A. Trunick. Breaking the global logjam. *Logistics Today*, 46(5):1–11, May 2005. See figure on p. 1.

[69] Perry A. Trunick. Stay loose for the RFID strecth run. *Logistics Today*, 46(3):35–37, March 2005.

[70] Perry A. Trunick. Stop playing the waiting game. *Logistics Today*, 46(11):29–32, November 2005.

[71] Perry A. Trunick. Vision and visibility govern global supply chains. *Logistics Today*, 47(7):24–26, July 2006.

[72] U.S. Customs and Border Protection. Container security initiative: 2006–2011 strategic plan, 29 September 2006. http://www.cbp.gov/linkhandler/cgov/border_security/international_activities/csi/csi_strategic_plan.ctt/csi_strategic_plan.pdf.

[73] Roy Want. Enabling ubiquitous sensing with RFID. *Computer*, 37(4):84–86, April 2004.

[74] D. G. Watters, P. Jayaweera, A. J. Bahr, and D. L. Huestis. Design and performance of wireless sensors for structural health monitoring. In D. O. Thompson and D. E. Chimenti, editors, *AIP Conf. Proc. 615: Quantitative Nondestructive Evaluation*, pages 969–976, May 2002.