8-1-2009

TelosRFID an ad-hoc wireless networking capable multi-protocol RFID reader system

Michael P. Lewis

TelosRFID
An Ad-Hoc Wireless Networking Capable Multi-Protocol RFID Reader System

by

Michael P. Lewis

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of
Master of Science in Computer Engineering

Supervised by

Professor, Department of Computer Engineering Dr. Kenneth Hsu
Department of Computer Engineering
Kate Gleason College of Engineering
Rochester Institute of Technology
Rochester, New York
August 2009

Approved By:

Dr. Kenneth Hsu
Professor, Department of Computer Engineering
Primary Adviser

Dr. Muhammad Shaaban
Associate Professor, Department of Computer Engineering

Dr. Andres Kwasinski
Assistant Professor, Department of Computer Engineering

Thesis Release Permission Form

Rochester Institute of Technology

Kate Gleason College of Engineering

Title: TelosRFID: An Ad-Hoc Wireless Networking Capable Multi-Protocol RFID Reader System

I, Michael P. Lewis, hereby grant permission to the Wallace Memorial Library to reproduce my thesis in whole or part.

Michael P. Lewis

Date

Dedication

I dedicate this work to my friends, family, co-workers and faculty who have helped me over my years at RIT. Special thanks go to my parents for their love and supportive in everything I do.

Acknowledgments

I would like to acknowledge the support of Dr. Kenneth Hsu, and my thesis committee. Without their input and assistance, this work would not have been possible.

Abstract

Radio Frequency IDentification (*RFID*) is rapidly being adopted as a powerful tool used in object tracking[1, 2, 3, 4], access control[5], telemedicine[6, 7] and inventory management[8, 9]. Its basic architecture endows *reader* devices with the capability to wirelessly read stored data off of RFID *tags*. Because of competing standards, there is no unified air protocol for RFID communication. The proliferation of competing standards, paired with the proprietary nature of commercial readers, can make maintaining and upgrading an RFID infrastructure expensive and time-consuming.

Part of the solution that this thesis proposes is an RFID reader which supports custom air protocol implementations. This provides the ability for system administrators to rapidly and inexpensively upgrade their RFID infrastructure to new standards and security policies[10], without equipment replacement. The flexibility of this reader also facilitates the development of innovative custom RFID protocols within academia.

To further reduce the costs associated with the adoption of a new infrastructure, RFID readers would benefit from supportting ad-hoc wireless networking[11, 12, 13, 14]. This feature mitigates the need for an installed infrastructure[15] and facilitates immediate deployment of RFID systems. The development of a multi-protocol RFID reader with ad-hoc wireless capabilities will be a boon for both the commercial and academic sectors[16, 17]. The negligible infrastructure will mitigate entry costs, and the ability to inexpensively upgrade equipment to the latest RFID standards will reduce lifecycle costs and improve equipment amortization schedules. The ease of re-deployment will further benefit network administrators by providing the ability for zero-cost system reconfiguration. This will result in more effective systems with lower maintenance costs.

To address these stated issues, this thesis outlines the design and implementation of an ad-hoc wireless networking capable multi-protocol RFID reader system, called TelosRFID. The name TelosRFID stems from the system's combination of Crossbow Telos rev. B (*TelosB*) ZigBee motes with a custom 13.56MHz RFID reader board.

The TelosB devices employ ZigBee protocol stack layers, implementing a 2.4GHz ad-hoc wireless network amongst themselves. Custom firmware enables the TelosB motes to construct an ad-hoc ZigBee network and relay RFID tag data from the custom reader board back to a client PC. The client PC runs custom software, developed for this thesis, to display pertinent information regarding tag activity.

The TelosRFID reader board is a custom hardware device that can communicate with 13.56MHz RFID tags. It runs custom firmware in order to control tag communications, manage tag presence monitoring, and relay tag information through the ZigBee network (via its attached TelosB mote.)

The system is designed to be demonstratably useful. It performs as described and can be immediately applied to research in the fields of systems engineering, information technology, and wireless air protocol development. Its functionality can be visibly confirmed, and configuration errors are easily detected at every component in the system. This framework provides a reliable and established baseline for future enhancements to the system's feature set.

Listings

Contents

List of Figures

Chapter 1

Thesis Overview and Motivation

To date, there has been some academic research which hypothesizes wireless sensor networks which employ a swarm of RFID readers[3, 18, 19, 13]. Despite this research, there have been almost no academic implementations of such a network[17]. In the work performed by Faschinger, *et al.*a system of RFID readers is attached and incorporated into a wireless sensor network. The system is used to develop workflow optimization techniques. Although an excellent tool for process engineering, the RFID readers used were commercial off the shelf (*COTS*) devices with limited programmability[17, 20]. This limits their applicability to the commercial air protocol that they were originally designed to process. The ability to adapt them to other protocols is non-existant, making such a system condemned to early obsolescence.

This thesis outlines a system designed to bridge RFID technology with the flexibility of wireless sensor network communications. It works by piggybacking a custom RFID reader onto a Crossbow Telos rev. B (*TelosB)* wireless sensor network (*WSN*) mote in order to supplement the reader with ad-hoc wireless networking capabilities. This reader will be user programmable to facilitate the development and testing of new protocols created by the academic community. This system as a whole has been named TelosRFID; a conjunction of 'TelosB' and 'RFID Reader'. The custom circuit board designed for this thesis, with its custom firmware, will be referred to as the TelosRFID Reader Board.

Before describing the technical details of this thesis, some brief background on the technologies involved will be reviewed. The history and applications of RFID will be

explored. Nuances and terminology related to its technology are explained to provide a basis for subsequent development. Likewise, the architecture, components, standards, and features of the ZigBee wireless sensor network technology will be reviewed to establish a baseline for the expected capabilities of such a network.

Following the surveys of involved technology and previous work, the original contributions of this thesis will be explained in depth. For clarity, these contributions will be divided into three chapters. The first chapter will outline the system overview. It describes the high level goals and accomplishments of the system, as well as how each individual component fits into the system as a whole. The following chapter describes the hardware used in completion of this thesis, including the original contribution of a custom RFID reader board. This reader board was designed with intent of relaying RFID tag communications to the rest of the system. With the system hardware layed out, the software written for each device will be described and explained. Three individual suites of software, each in a different language, for a different hardware platform and microprocessor, had to be written to tie the system together. The program flow, caveats, and pseudo-code are elaborated upon in order to better convey the inner workings of the system.

Following the technical description of the system, the resulting product is analyzed and judged. The extensibility of its framework for future research is explained. Metrics regarding its performance are collected and suggestions for subsequent expansion are suggested. Additionally, current shortcomings of the system are described, as well as proposals for improvements in future designs.

Chapter 2

RFID Background

The concept of RFID has existed for over half a century[21], dating back to World War II. It is only recently that it has emerged as a viable commercial concept[22, 23]. With advances in device miniaturization techniques, low-power design, and wireless communications, RFID is rapidly being adopted as a powerful and low-cost object management tool. The basic principle of contemporary RFID infrastructures is that a reader (or readers) monitor their environments for tags that are within their proximity[24]. The information read from nearby tags can be used for any number of applications, including product tracking, access control, sensor monitoring, telemedicine supervision, *etc.*.

2.1 Readers

RFID readers are devices capable of communicating with RFID tags, with read or read-/write abilities[25, 26, 27]. They are the medium through which RFID tags can be tracked and referenced to a central database. They must be capable of communicating with a central client or database to be practically useful for object tracking[28, 29, 30].

2.2 Tags

RFID tags come in two basic varieties[31], depending upon the system's requirements. These versions are active or passive, with their hybrid variation being called semi-active.

Active

Active RFID tags consist of a microcontroller, an antenna, and a battery. The reader sends a wireless response request to a tag, which is captured by the tag's antenna. The tag's microcontroller (powered by a battery) detects the antenna's signal and reads this request . It then computes a response and uses the battery to power an amplifier, transmitting its response. The advantage of active tags is that they can operate at much greater distances than passive tags. Additionally, they can do more complex processing, store more tag data, and communicate at higher data rates than their passive counterparts[12]. The obvious drawback of active tags is that including a battery increases both the size and cost of the RFID tag, in addition to reducing shelf life. This makes active tags inappropriate for disposable or low-cost applications such as retail inventory management.

Passive

Passive RFID tags are similar in structure to active tags, but lack a battery and use a much simpler microcontroller or analog circuit in order to reduce power consumption[32]. This limits their computational ability, and can impose restrictions upon the protocols and security mechanisms they can use[33]. The premise of passive tags is that both the response computation and transmission are powered by the reader's original wireless signal. For near-field communication, this is accomplished by magnetic coupling of the reader and tag antennas. For far-field communication, the tag rapidly modifies the impedance of its antenna to modulate the backscatter pattern that is reflected back to the reader. Although the range and data size of passive tags is far more limited than active tags[34], they can be manufactured cheaply, and can even be printed or embedded into paper. Additionally, their shelf life is nearly infinite, making passive tags an ideal solution for product tracking.

Semi-Active

Semi-active tags are a hybrid concept combining properties of both active and passive tags. They consist of a microcontroller, an antenna and a battery, just like active tags. In an

effort to reduce size and costs, however, the battery is much smaller, and an RF amplifier is omitted. The battery is used to power the microcontroller, permitting complex security protocols and sizable data storage. Although the microcontroller is battery powered, RF communication is not powered, and is performed identically to passive tags, employing near-field magnetic coupling or far-field backscattering reflection.

Chapter 3

ZigBee Background

ZigBee is a network stack implementation that provides a set of standards for low-speed and low-power ad-hoc wireless communication among devices[35, 36]. It encompasses several layers, including routing, security and application frameworks[37, 38]. It operates on top of the IEEE 802.15.4 protocol, which includes PHY and MAC layers[39].

3.1 Architecture

ZigBee is a stack-based architecture composed of multiple functional layers[40]. The two primary specifications for these layers are the IEEE 802.15.4 hardware layer and the ZigBee Alliance defined layers.

3.1.1 IEEE 802.15.4 PHY and MAC Layers

The IEEE 802.15.4 Layer is compromised of the Physical (*PHY*) and the Medium Access Control (*MAC*) sub-layers. The PHY layer defines the physical properties of the communications link. This includes specifying the frequency and wireless signal properties used in the communication. For the IEEE 802.15.4 specification, the frequency used must be either 868/915 MHz or 2.4GHz, which are ISM (*industrial, scientific, and medical*) band frequencies. The MAC layer defines the radio access control protocols required to keep members of the wireless system communicating fluently. Its roles include controlling the modulation

scheme sent to the radio, data/time synchronization, beacon handling, and transmission speed regulation.

3.1.2 ZigBee Alliance Layer

The ZigBee Alliance Layer builds upon the IEEE 802.15.4 Layer by supplementing it with higher-level functionality. This higher-level abstraction is what allows rapid application development without requiring intimate manipulation of the underlying hardware. The ZigBee Alliance Layer actually consists of two major sub-layers, the Network (*NWK*) Layer and the Application (*APL*) Layer.

Network Layer

The Network layer is what governs the ZigBee network's construction and maintenance from a connection perspective. It is in this layer that the network's topology (star, tree, mesh, *etc.*) is defined. It is also the layer upon which nodes with specialized routing properties differentiate themselves. At a Network level, ZigBee motes can function as Coordinators, Routers, or End Devices. ZigBee Coordinator nodes control key aspects of network communication and are responsible for maintaining a healthy network. ZigBee Router nodes are another special entity in the NWK layer. Router nodes allow networks to expand by joining separate networks together. Joining networks is accomplished by enabling inter-network communication, and by sharing and relaying information between one another.

Application Layer

The Application layer is the name of an enveloping set of sub-layers which are designed to interface with the end user's software. These sub-layers include the Application Support (*APS*) Sublayer, the ZigBee Device Object (*ZDO*) and the Application Framework, which includes the Application Objects. The Application Support Sublayer acts as a central point of communication between the Application Layer, the Application Framework, the ZigBee

Device Object, the Network Layer, and the Security Service Provider. The role of the APS layer is to negotiate appropriate usage of the network layer by the user's Application Objects, without exposing direct control of the hardware. The ZigBee Device Object is a high level description of ZigBee parameters required by the node in order to participate in a ZigBee network. This includes the node's network ID, its role (ZigBee End Device, ZigBee Router, or ZigBee Coordinator), its Cluster Identifier, and other properties which will define its behavior in relation to other nodes. The Application Framework is what permits developers to design a *ZigBee Profile*. Each ZigBee Profile exists as an instance of an Application Object within the Application Framework. This concept of a device comprised of multiple objects allows implementors to extend standards based specifications with custom functionality, without sacrificing standards compliance.

3.2 Features

3.2.1 Low Power

ZigBee nodes are designed to be low-speed and low-power[41]. A 2.4 GHz node has a theoretical maximum data bandwidth of 250kbps and an optimal maximum range of 50 meters. Because of these characteristics, ZigBee technology is very attractive for event-based applications. Because every network behaves differently, it is impossible to determine exactly how long the batteries in a TelosB mote will last. In a worst-case scenario, the current draw while in receieve mode is 23mA[42]. At 23mA, a pair of Energizer E91 AA alkaline batteries batteries have an expected battery lifetime of 100 hours[43]. In a real world scenario, motes spend most of their time in sleep or idle modes, consuming current on the order of 21μA. In situations where information is not constantly streamed, nodes can run on battery power for exceptionally long lengths of time, dramatically reducing their associated maintenance costs[44].

3.2.2 Ad-Hoc Networking

A ZigBee network can be designed to be self-organizing using one of three topologies. The network uses ZigBee End Devices, ZigBee Routers, and ZigBee Coordinators in order to form a tree, mesh or star topology. If a star topology is used, the center node is always a ZigBee Coordinator. In a tree or mesh topology, ZigBee Router nodes can be used to incorporate a self-contained network into a larger network of sub-networks. Incorporating several sub-networks into a larger one requires that each sub-network be willing to share and route data packets in conjunction with other sub-networks.

3.2.3 Security

The ZigBee specification provides several security measures in order to guarantee the integrity of networks, data, and communications[45, 40]. Each network can implement key-based authentication to control which motes can join the network. In this situation, a node wishing to join the network must be authenticated by a *Trust Manager* device. It may join the network using a factory-installed (or over-the-air programmed) security key. Once this node is admitted into the network, it may establish link keys between itself and other nodes that it communicates with. This ensures that even within the network, no node is given full access. Additionally, all of this key authentication and communication is encrypted using the NIST approved 128-bit AES encryption algorithm.

The effect of implementing this stack of security mechanisms is that packets are protected from eavesdropping, man-in-the-middle, as well as replay attacks. Because unauthorized nodes are not permitted onto the network, data injection attacks are nearly impossible. Even on the off chance that a malicious node gains access to the network, if the network is set up for high security, the node would still be unable to eavesdrop on most network traffic due to the link keys which establish private connections between individual nodes.

Chapter 4

System Overview

The TelosRFID project involves hardware and software components, both of which operate on several distinct layers. Each component contributes to the system goal of enabling the wireless communication of RFID tag detection events. It accomplishes this specification by constructing a communications flow, connecting each device in a chain.

The system premise is that the RFID reader board will detect any 13.56Mhz RFID tags that enter its proximity field. When a tag is detected, a message containing tag information is sent from the TelosRFID reader board to the attached TelosB mote. The attached TelosB mote will then propagate that tag's data through a wireless ZigBee network to a client-side TelosB mote. The client-side TelosB mote feeds the tag information to a client computer [Figure 4.1], which can then display the tag information on a screen. In addition to just tag information, the client will also display the serial number of the mote which originally transmitted the information, letting the user know where the tag was read from.

4.1 Goals

The overarching goal of the TelosRFID system is to let a central PC client monitor RFID tags being detected at multiple wireless reader stations. Each of these TelosRFID tag reader stations are portable, battery powered devices capable of inter-communication. Multiple devices need to be capable of forming ad-hoc networks, cooperating to improve overall performance of the network. The central PC must be able to display information about

tag presence. This information includes the IDs of detected tags, what reader they were detected from, when they entered the reader's field, and when they left. This is the basic functionality required to operate a useful system. This degree of functionality will provide all of the essential building blocks for future researchers to implement localization algorithms, tracking programs, custom RFID tag development, security mechanisms, data management or other sorts of academic pursuit[46].

4.2 System Components

The TelosRFID system is composed of several hardware and software sub-systems that communicate amongst each other. Each component of this communication ecosystem exists for a specific purpose. The functional requirements for each sub-system is detailed as follows.

4.2.1 TelosRFID Reader Board

The TelosRFID reader board is a custom piece of hardware that incorporates a microcontroller with an RF front-end, running custom RFID protocol and communications firmware. It is battery powered off of 2-AA batteries. The reader board is capable of detecting 13.56MHz ISO/IEC 14443 RFID tags. It can discriminate several tags within its field using the ISO/IEC 14443-3 anti-collision algorithm[47]. It can also determine the serial ID of every tag within its antenna's proximity. The firmware is also capable of being reprogrammed to support other (non ISO/IEC 14443) RFID air protocols, provided that they operate at 13.56MHz.

When tags enter or leave the antenna's proximity, detection information needs to be relayed to the attached TelosB mote. This communication is initiated by the reader board pulling high an interrupt line on the TelosB mote. This interrupt line lets the TelosB mote know that a detection message is about to come from the reader board and that it should start listening to its UART port. Shortly after the inturrupt line is pulled high, the TelosRFID

reader board communicates the tag message via UART at 112500 baud to the attached TelosB mote.

For verification of proper functionality, the reader board's red LED reflects the state of RFID tag detections. Whenever a tag enters or leaves the reader's field, the onboard red LED will toggle. This is an important feature for debugging a malfunctioning system, as it verifies that tag information is being detected and transmitted to the attached TelosB mote. Additionally, the green LED toggles at a constant 0.5 second period. This is a visual cue indicating continued functionality of the device.

4.2.2 Reader-Side TelosB Mote

A battery-powered TelosB mote is attached to the TelosRFID reader board. Using custom firmware, it is used to relay communication with, and to, the rest of the ZigBee network. Under normal conditions, it acts as an ad-hoc router within the ZigBee network. It helps propogate general traffic through the network, relaying packets from other motes to the central PC client.

When the TelosRFID reader board detects a tag, it will signal this event to the attached TelosB mote by pulling high an interrupt line. This interrupt signals the TelosB mote to disable its radio and prepare itself for an incoming tag communication. As detection data arrives at the TelosB's microcontroller it is queued for re-transmission. After the entire detection message has been received, the TelosB re-enables its radio and transmits the detection packet out onto the ZigBee network. This packet's destination is the client-side TelosB mote.

4.2.3 Client-Side TelosB Mote

The client-side TelosB mote is connected to the USB port of the client PC. Because the PC's USB port is powered with a 5.0V supply, the client-side mote does not require batteries. Instead, it uses the PC's internal power supply. When ZigBee communication packets

arrive at the client-side mote, it immediatelly relays them to the PC via its USB port. Likewise, if the PC wants to send out a packet to the ZigBee network, the client-side mote will function as an outgoing radio. Any time the client-side mote receives communication via its USB port, it will relay the packets, unmodified, out onto the ZigBee network.

4.2.4 PC Client

The PC client can be any computer running the Linux operating system, either natively or through a virtual machine. It runs custom software which will monitor its USB port for mote communication. When the attached client-side mote communicates a tag detection packet through the USB port, the PC client is responsible for processing this data. The software written for this thesis will parse the incoming packet, turning its binary representation into useful information. It will then display on the screen the nature of the detection (entering or leaving a field), the ID of the tag and the unique ID of the reader board that detected the tag.

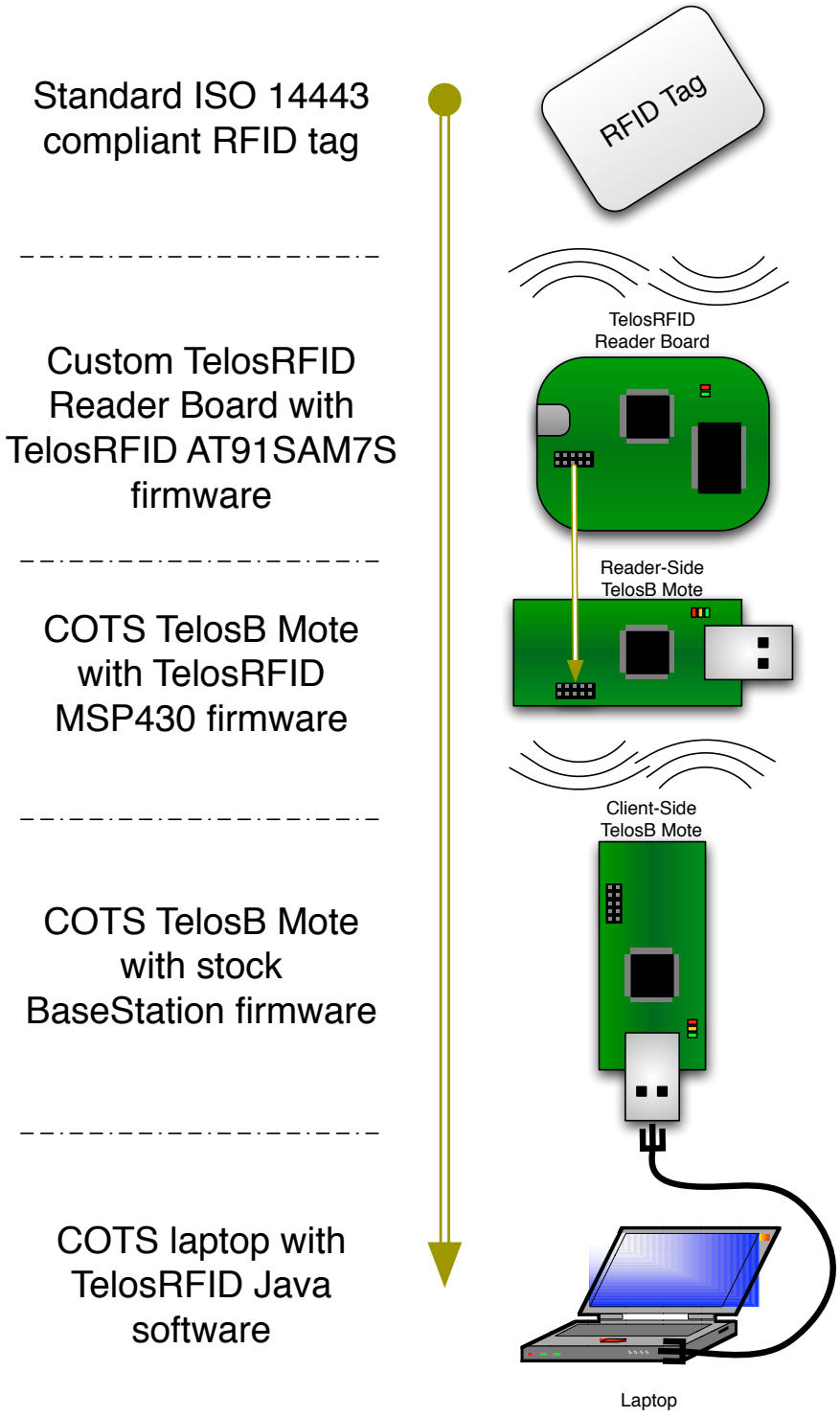

Figure 4.1: System Overview Diagram

4.3 Testability and Validation

For the TelosRFID system, a series of basic requirements are laid out and are used to validate the performance of the system as a whole.

The boot sequence of the TelosRFID reader board can be viewed by connecting the 112500 baud DBGU port to an ASCII terminal program. The boot sequence is reported in plaintext to verify that the system started up properly. Once the system is booted, its continued operation is visibly established by a continuously blinking ”heartbeat” using the green LED.

When the TelosRFID reader board detects a tag, it will communicate the detection information to its attached TelosB mote via the DBGU port. In addition to the serial communication, the reader board will toggle its red LED to visually indicate the detection and communication.

When the reader-side TelosB mote receives a detection message interrupt from its attached TelosRFID reader board, it will toggle its red LED. After queuing the message from its UART port, it re-transmits it out to the ZigBee network. As it does this it also toggles its blue LED. Because this is performed so quickly, both LEDs should appear to toggle at the same time. Both LEDs toggling in unison visually indicate proper functionality of the TelosB’s communications capabilities.

When the client-side TelosB mote receives a detection message from the ZigBee network, it will communicate this detection to the USB port. In addition to the serial communication, it will toggle its LEDs to visually indicate receipt of the communication.

As the client PC receives a communication via the USB port, the Java application will be waiting for messages from the ZigBee network. It detects the incoming message and immediately displays information about the detection. When tag detections are displayed in the PC’s console it validates that the communications chain is operating as expected. If a detection is not displayed on the screen, then the previously described validation tools must be used to investigate and fix the source of the problem.

Chapter 5

Hardware Implementation

The TelosRFID project involves several distinct hardware components. Most are commerically available off the shelf. This includes the following components:

- 13.56 MHz ISO/IEC 14443 compliant RFID tags
- Two (2) Crossbow Telos rev. B wireless sensor devices
- A client PC running a distribution of the Linux operating system

In addition to the COTS devices, an original custom hardware device was created as an essential component of the TelosRFID system. Dubbed the *TelosRFID Reader Board*, its design and firmware are significant contributions of this thesis.

5.1 Academic Contributions

The RFID tags used are commercial off the shelf components purchased from an RF technology retailer. They are not considered academic contributions in this thesis.

The TelosRFID tag reader board is a new, fully custom piece of hardware designed by the author. It was designed in Altium Designer^{TM}using component datasheets and several reference designs[48, 49, 50]. Once designed, the PCB layout was submitted to Sunstone Circuits for manufacture. After the PCB itself was made, the board was sent to Screaming Circuits, along with all required components, and the final product was assembled using

SMT (*surface mount technology*) techniques. The completed work is unique, useful, and will contribute to the RFID and wireless research community by assisting with protocol and process development. As detailed above, this custom hardware is needed in order to tie together two distinct wireless technologies, RFID and WSN. It's flexible programming and communication features allow for research and applications that have previously not been possible.

The TelosB wireless sensor devices used are commercial off the shelf components purchased from an RF technology retailer. They are not considered academic contributions in this thesis.

The client PC can be any standard desktop or laptop computer capable of running Linux. They are not considered academic contributions in this thesis.

5.2 13.56 MHz ISO/IEC 14443 RFID tags

For the purpose of this thesis, 13.56 MHz ISO/IEC 14443 compliant RFID tags were chosen. The specific type of tag used in the development of the communications software stack is sold under the name *Mifare Classic*. The Mifare Classic 1k tags[51] were chosen because they are based upon the ISO/IEC 14443 specification stack, are in widespread use (over 1 billion tags)[52], and are readily available.

The ISO/IEC 14443 standard is used as an example RFID stack implementation due to its well-defined specification[53, 54, 47, 55], as well as its widespread adoption in industry[56, 52, 57, 58]. It provides an immediate practical use, while demonstrating software layers used to access the hardware. The example implementation developed for this thesis provides an example framework for developing communication stacks. These programmable communication stacks can be implementations of existing standards, or could be used to prototype 13.56 MHz RFID protocols that have yet to be developed or standardized.

5.3 TelosRFID Reader Board

The TelosRFID reader board is a custom piece of hardware that was developed specifically for this thesis. The overarching goal behind the hardware design of the TelosRFID reader board was to create a device that could be battery-powered, and entirely wireless. It had to be able to detect 13.56 MHz RFID tags, and transmit that tag's ID to the connected TelosB mote. One of the most difficult design constraints is that is had to be incredibly small, so as to permit portability.

5.3.1 Components

From a high level perspective, the TelosRFID reader board is a small circuit board with a microcontroller and a 13.56 MHz wireless front-end integrated circuit (*IC*). The microcontroller (μC) used is an Atmel AT91SAM7S256 and the wireless front-end IC is a Philips NXP CL RC632. The microcontroller communicates via external interfaces in order to facilitate programming and communication, as well as communication with the CL RC632, which powers and modulates the embedded antenna. The board also uses two power management ICs, the TPS61202 and the IRU1502-33, in order to negotiate voltage level between power sources and components which need to operate at different voltage levels. It requires requires an external 5.0V power source in order to operate.

Atmel AT91SAM7S256

The Atmel AT91SAM7S256 is a 32-bit ARM-based RISC microcontroller[59]. It has the following characteristics that make it a good choice for the TelosRFID platform.

- 256 Kb internal high-speed flash memory
- 64 Kb internal high-speed SRAM
- Periodic Interval Timer (*PIT*)

- Advanced Interrupt Controller (*AIC*)
- Debug Unit (*DBGU*)
- Programmable Parallel Input/Output Controller (*PIOA*)
- USB 2.0 Full Speed Device Port
- 2 Universal Synchronous/Asynchronous Receiver Transmitters (*USART*)
- Master/Slave Serial Peripheral Interface (*SPI*)
- Two Wire Interface (*TWI*), usable as an I^2C interface
- Small profile 64-pad QFN package
- at91lib - Atmel's freely available C library for use onboard AT91 series microcontrollers

These varied properties each solved distinct design problems faced at the onset of this thesis. The copious flash memory was needed in order to accommodate RFID protocol stacks of varying complexity. Although the example firmware implementation provided with this thesis takes up a fraction of this program space, it is not inconceivable that a more complicated protocol, possibly one with layers of security and/or encryption, may demand a significantly greater program footprint. Likewise with the SRAM, only a portion of the provided 64 Kb was needed, but protocol implementations involving encryption algorithms will use procedures that require much more memory to execute.

The PIT was used to drive the heartbeat of the TelosRFID board, as well as to control all timing related functionality. Tying the PIT to the AIC permitted the implementation of an LED heartbeat, without any modifications to the main loop of the program. The PIT interrupt vector handler was also used to implement highly accurate and reliable time-based functions, such as delays. The AIC, although not entirely necessary, served to make event-based processing within the firmware much easier. By providing flexibility in what sources

can trigger an interrupt (*i.e.*external PIO pins, internal embedded peripherals), designing interrupt-driven firmware became a simpler and more straightforward task. This results in a system that is much more responsive in nature.

The DBGU, PIOA, USB 2.0, USART, SPI, and I^2C peripherals are all I/O devices capable of performing communications tasks. The original design was architected to allow as much redundancy of interfaces as possible. Because of the inavailabilty of dual inline packages (*DIP*) for many components used, breadboarding a prototype was not a reasonable option. The hardware was designed and implemented entirely by datasheet, so redundancy and backup plans were always considered a high priority. A direct result of this requirement was the selection of a microcontroller with numerous communications interfaces to use and expose.

In the final implementation, the USB 2.0 port is used to power the board, using its 5 volt supply. The DBGU interface is used to program the microcontroller, as well as to communicate with the TelosB device. Additionally, when connected to a serial port at 115200 baud, the DBGU port can be used to view diagnostics from the system's boot sequence, as well as outgoing tag communications. The PIOA controller is used to control two output pins on the AT91SAM7S256. Those output pins (PA7 and PA8) are used to pull up an interrupt line on the connected TelosB. This signals to the TelosB that a communication is about to arrive at its UART0 peripheral. The necessity of this interrupt line is discussed further in section 5.4. The SPI interface is used for serial communication to and from the NXP CL RC632 device. The USART and I^2C peripherals were not ultimately needed for the final deliverable, but a pin header to access the I^2C is provided if users need it in order to extend functionality.

The availability of a 64-pad QFN package for the AT91SAM7S256 allows for a very small footprint on the board. With side dimensions of 9.0mm $\times$ 9.0mm, this highly capable microcontroller takes up less than $1cm^2$ on the final PCB.

The final point on the list is a publicly available free library called at91lib that is provided by Atmel for their AT91 series microcontrollers. This C library provides basic accessor functionality to most of the embedded peripherals in the chip. It also provides abstraction of some low level register accesses, reducing the possibility of errors during the development cycle. The use of this library and its sample projects saved dozens of hours in development time. It also undoubtedly improved the reliability of the firmware by shifting the development focus from micro-implementation details to program flow and protocol development. Since the library was open source, it ended up being modified. All NXP CL RC632 communications functionality and ISO/IEC 14443 protocol stack implementations were developed into the library framework.

Philips NXP CL RC632

The Philips NXP CL RC632 IC is a device meant to simplify the front-end design for 13.56 MHz RFID systems[60, 61]. It is a highly integrated RFID reader IC used for wireless communication at 13.56 MHz. It supports every layer of the ISO/IEC 14443 protocol stack, holistically or as distinct layers. That ability was essential for this thesis, as it allows for the implementation of algorithms that use the standards-compliant ISO/IEC 14443-1[53] and ISO/IEC 14443-2[54] low level layers, while implementing custom higher level layers. This is required in order to design and implement custom anti-collision algorithms and communications protocols.

The CL RC632 is able to bi-directionally communicate with the AT91SAM7S256 using an SPI bus, drastically reducing the number of traces, as compared to a parallel communication interface. Because the CL RC632 is digitally controlled, the microcontroller does not have to directly manage the analog frontend. This abstraction allows the microcontroller to issue commands at its own pace, while the CL RC632 handles the strict wireless timing requirements of RFID communication. This simplifies firmware design and makes the implementation of computationally complex protocols and algorithms less challenging (as compared to if the microcontroller directly manipulated the analog front-end).

The reader IC is important because it is a specialized chip designed to simplify the analog design of the RF circuit. It ensures a clean, jitter-free 13.56 MHz baseband frequency, and flexibly handles all of the modulation operations specified in ISO/IEC 14443-2[54]. To further simplify timing requirements on the microcontroller, the CL RC632 even has a built in 64 byte send/receive buffer. This provides the AT91SAM7256 with the flexibility to handle tag communications on its own timing schedule, even with performance delays.

In order to identify each reader board, every CL RC632 has a unique 32-bit serial number. This serial number can be read via the SPI bus and can then be embedded into all outgoing packets so that the client PC can identify exactly which TelosRFID board each communication originated from.

Texas Instruments TPS61202

The Texas Instruments TPS61202 IC is a charge pump voltage converter[50]. Its basic purpose is to take in a power supply at 3.3V DC and output a power supply at 5V DC. The original goal behind this choice was the desire to be able to power the entire TelosRFID reader board off of the TelosB's batteries. The boards have jumpers that were intended to allow the reader board to change its power source back and forth between its onboard USB port and the attached TelosB's battery supply. Post-fabrication, this proved to be a poor solution, as several problems crippled its success. The most significant implementation problem was power quality from the TPS61202 IC. The nature of a boost converter is that it uses capacitors to store charge and then rapidly switches voltage sources in order to increase output voltage to levels above it's original input voltage. The problem with a switching power supply is the transient line noise that it tends to produce. In the case of the TPS61202, it oscillates between 1.25 MHz and 1.65 MHz, which will have massive negative repercussions on the analog front-end's high frequency performance at 13.56 MHz.

Given 3V DC input from the TelosB device, the reader board will generate a 5V DC supply. Unfortunately, due to insufficient power filtering, this 5V supply is sub-optimal for

RF wireless communication. As a result of this, the TelosRFID reader board must have its own 5V DC power supply in order to operate as specified.

International Rectifier IRU1502-33

The International Rectifier IRU1502-33 is a linear regulator which converts power from the USB's 5V supply into a 3.3V supply needed by the microcontroller. Since it is a linear regulator, rather than a switching charge pump, the supplied voltage is much cleaner than that provided by the Texas Instruments TPS61202. This configuration is aided by the fact that the 5V supplies used (PC USB host controller or a 5V battery pack) tend to be less noisy sources of power.

5.3.2 Electrical Schematics

The electrical design of the TelosRFID board was completed virtually, before it was ever physically implemented. The two major obstacles to breadboard prototyping were the radio frequncies being used, as well as the lack of available components in through-hole packages. Because of the electrical sensitivities associated with wireless RF frequencies, a breadboard prototype could not be relied upon to give accurate results. Due to the inavailability of most components in through hole (DIP) packages, breadboarding would not have been possible without SMT adapter daughterboards. Because of the difficulty of prototype SMT work, the decision was made to design the entire board using Altium Designer, an electrical computer-aided design (*CAD*) tool.

The hardware circuitry, although custom designed, was somewhat derived from application notes by Atmel, Philips NXP, and Texas Instruments[48, 49, 50]. Some of the microcontroller portions of the board, including the USB 2.0 circuit were adapted from the Atmel AT91SAM7S-EK evaluation board's design files[48]. Likewise, the CL RC632 connectivity and RF tuning portions of the circuit were influenced by application notes from Philips NXP[62, 49, 63, 61, 64].

The following electrical schematics are organized into 5 diagrams, each representing a distinct role within the design. For full-page images, see Appendix **??**.

Top Level View

The top level view shows how each of the subsystems are related to each other. As can be seen, the microcontroller provides communication buses to the I/O subsystem. The microcontroller also controls the RF subsystem through an interrupt line and the SPI bus.

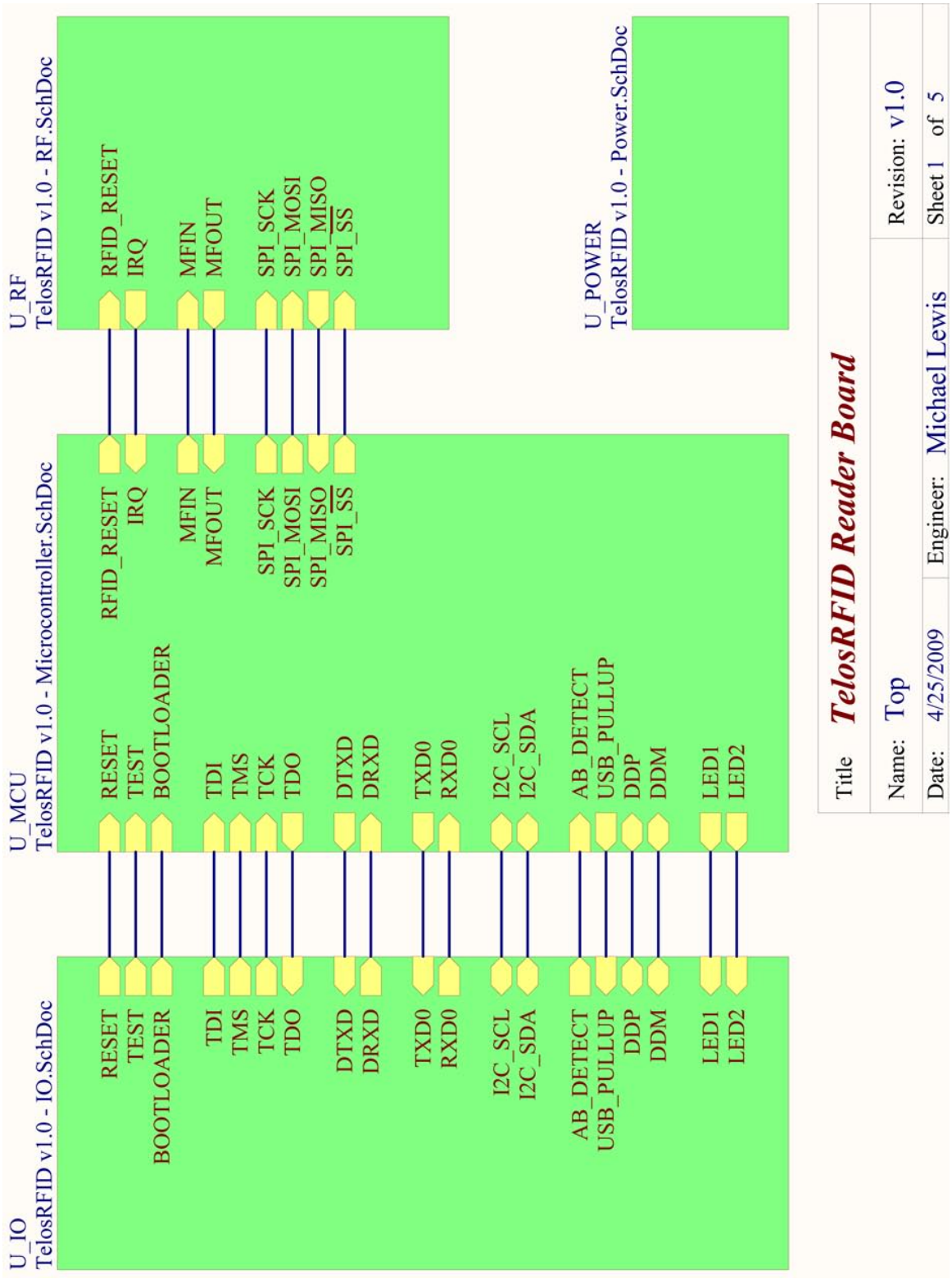

Figure 5.1: Design Schematic Page 1 of 5: Top Level

Microcontroller Circuitry

The microcontroller schematic shows what is needed to make the AT91SAM7S256 function, as well as what external peripherals are used.

- The circuit on the left of the schematic (U1A) describes the I/O capabilities that are used on the reader board. I^2C [pins 36 & 43], USART0 [pins 31 & 32], DBGU (USART) [pins 29 & 30], SPI [pins 21, 22, 27 & 28], USB 2.0 [pins 16 & 19], and general purpose I/O pins [pins 9, 10, 25, 26, 35, 37 & 42] all provide the capability for the microcontroller to communicate with other devices as well as to control operations performed on the reader board.

- The circuit in the upper right of the schematic (U1B) is the clock/oscillator circuit, which is responsible for generating a 48 MHz clock signal from crystal X1.

- The circuit on the lower right corner of the schematic (U1C) describes the JTAG interface for the AT91SAM7S256 IC. Although it was not used in my development, external pins are exposed so that future developers of the system can use the JTAG interface to load and debug custom firmware.

- Finally, the circuit on the bottom-center (U1D) of the schematic describes the power feeds upon which the microcontroller relies. The AT91SAM7S256 requires a 1.8V and a 3.3V power supplies. It has a built-in 1.8V regulator so that it can generate its own 1.8V supply (on pin 8, its VDDOUT). Decoupling capacitors are also included in order to reduce line noise in the power supply.

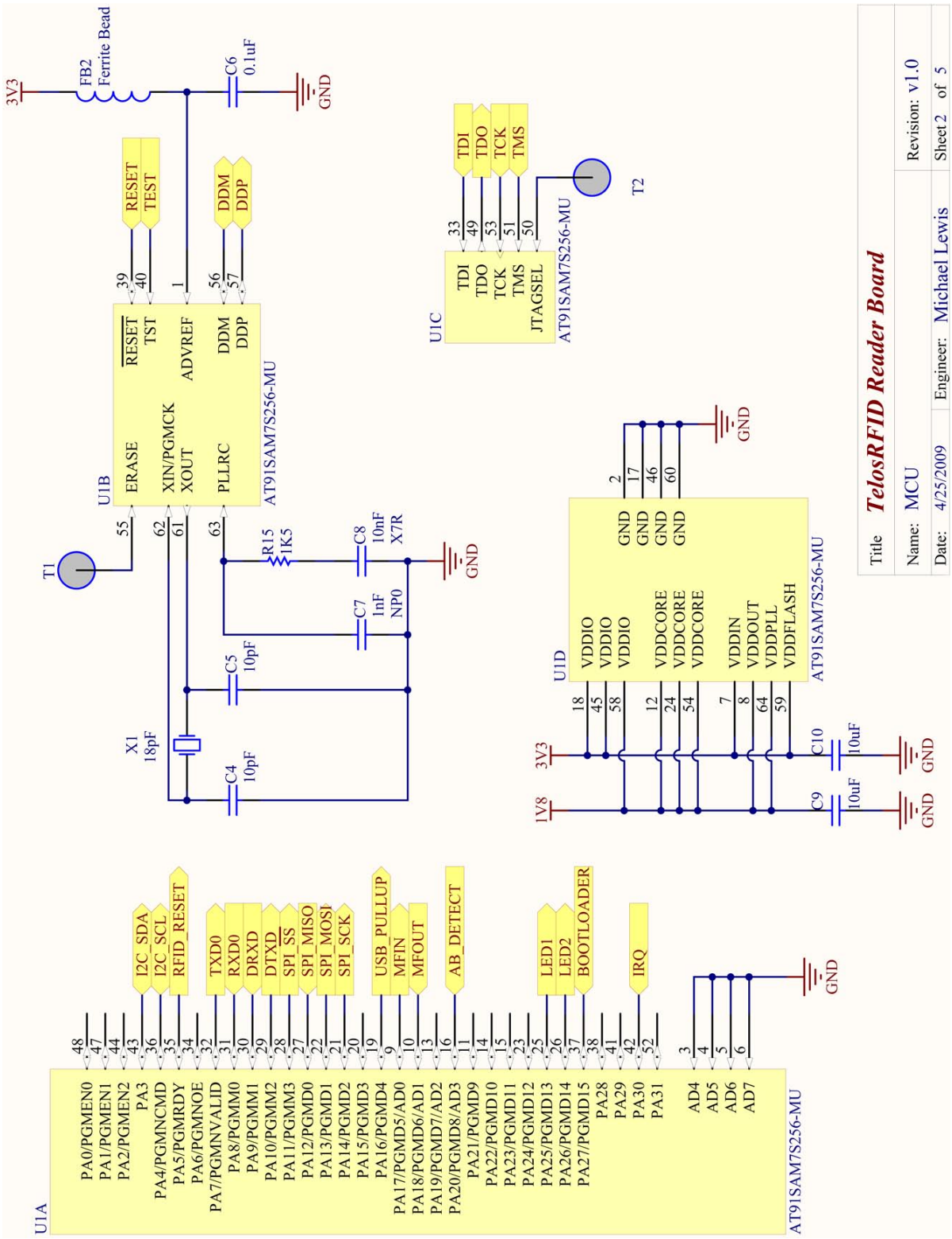

Figure 5.2: Design Schematic Page 2 of 5: Microcontroller Circuitry

RF and CL RC632 Circuitry

The RF schematic breaks down into two sections: the CL RC632 connection diagram, and the RF tuning circuit. The CL RC632 connection digram is relatively straightforward. The SPI bus is used to communicate with the AT91SAM7S256, the IRQ line signals interrupts, the RFID_RESET line permits the microcontroller to power cycle the CL RC632, and the MFIN and MFOUT lines expose low-level communication details. Aside from the crystal oscillator and some power filtering components, there are very few parts needed in this section.

One feature that was included in the hardware but has yet to be used in software is the MFIN and MFOUT lines. These stand for ”MIFARE in” and ”MIFARE out” and they can provide the capability to directly manipulate the data being sent and received from RFID tags.

The RF tuning portion of the circuit is significantly more complicated than the communications circuit. The RF circuit design incorporated knowledge from NXP[49] and Fotopoulou[65]. The design procedure entailed constructing the circuit as shown, leaving empty pads so that components whose values had yet to be determined could be placed at a later time. After the boards were manufactured and populated, the antennas were measured with respect to their inductance and impedance. Determining the values of the ”NC” components involved applying formulas found in the aforementioned application note. These formulas were based upon the measured impedance and inductance of the antenna trace. The tuning procedure is outlined in [49, 62]. In short, it involves tuning capacitor and resistor values on the board in order to obtain the lowest impedance and phase skew for the final filter stage. Because the above documents are protected under non-disclosure agreements, their details won't be outlined here. If needed, they are available from Philips NXP, with permission.

The final component values used were 10pF for C16 and C24, and 56pF for C18. C21. C17, C25, C19, and C22 were left empty. These components were hand-soldered onto the board once the calibration routine was completed. For further configurability, R16, R17,

and R18 are potentiometers. Modifying their resistance will change behavioral characteristics of the RF signal. The most dramatic effect is obtained by reducing the values of R16 and R17 to increase the output power. This occurs because the overall impedance of the antenna is reduced, increasing current through the coils.

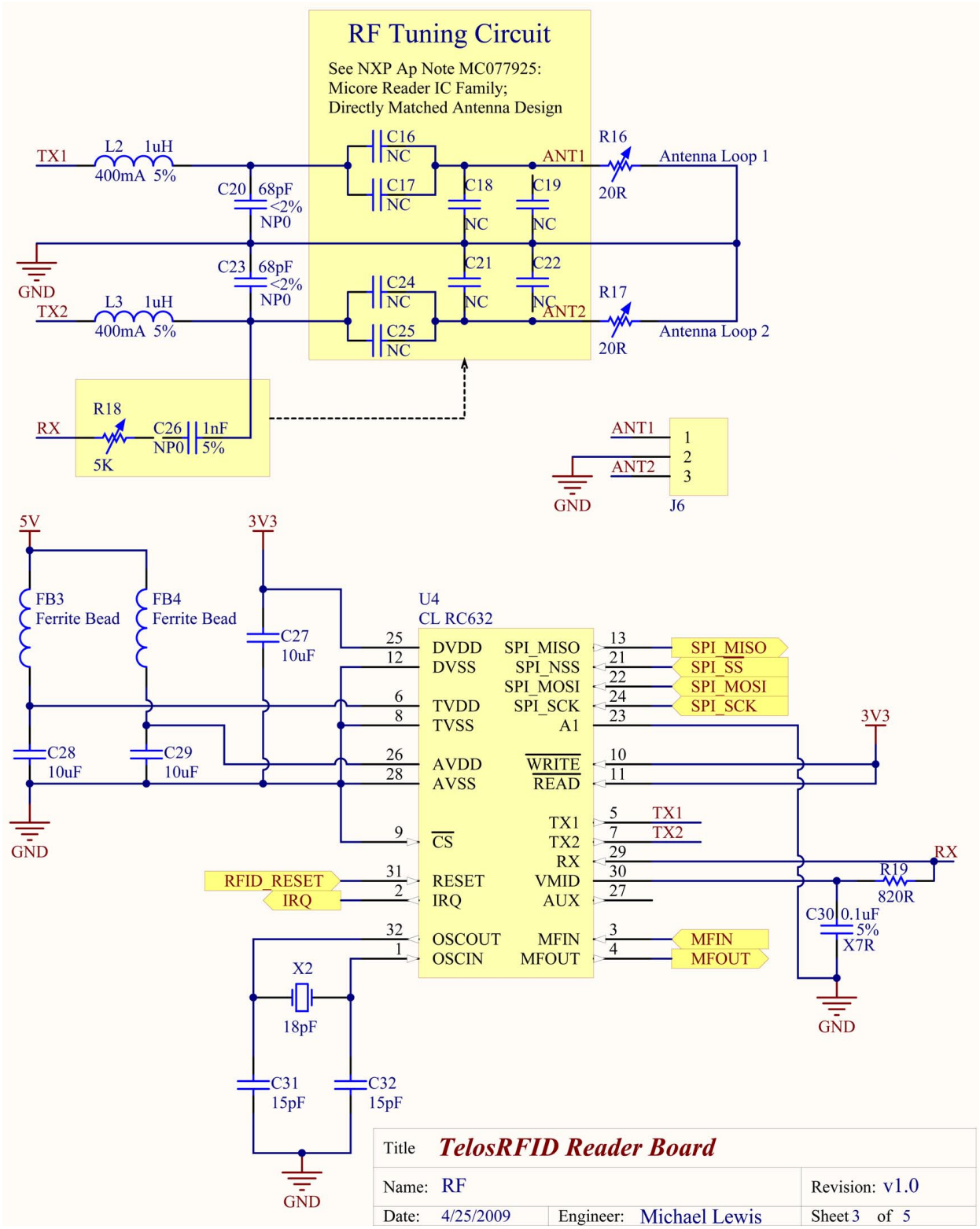

Figure 5.3: Design Schematic Page 3 of 5: RF and CL RC632 Circuitry

I/O Circuitry

The I/O schematic shows the interfaces that are externally exposed by the TelosRFID reader board.

- The USB 2.0 port powers the reader board, and could theoretically be used to communicate with a host computer.
- Two LEDs are used to indicate the status of the board. The green LED is used as a system heartbeat, with a 0.5 second period. The red LED is used to indicate RFID tag activity, such as a transponder entering or leaving the reader's field.
- The serial debug port is used to program the AT91SAM7S256. It is also used to print boot information to a terminal, and to report tag activity to the attached TelosB mote.
- The JTAG header can be used to program and debug the microcontroller, if an appropriate JTAG device is available.
- Finally, the expansion port is used to expose multiple communications lines to external sources. From conception, it was originally designed to plug directly into the attached TelosB mote. Due to compromises made during software development, this is no longer the case. Several pins of this header are still used, but are connected with the TelosB mote via a communications harness.

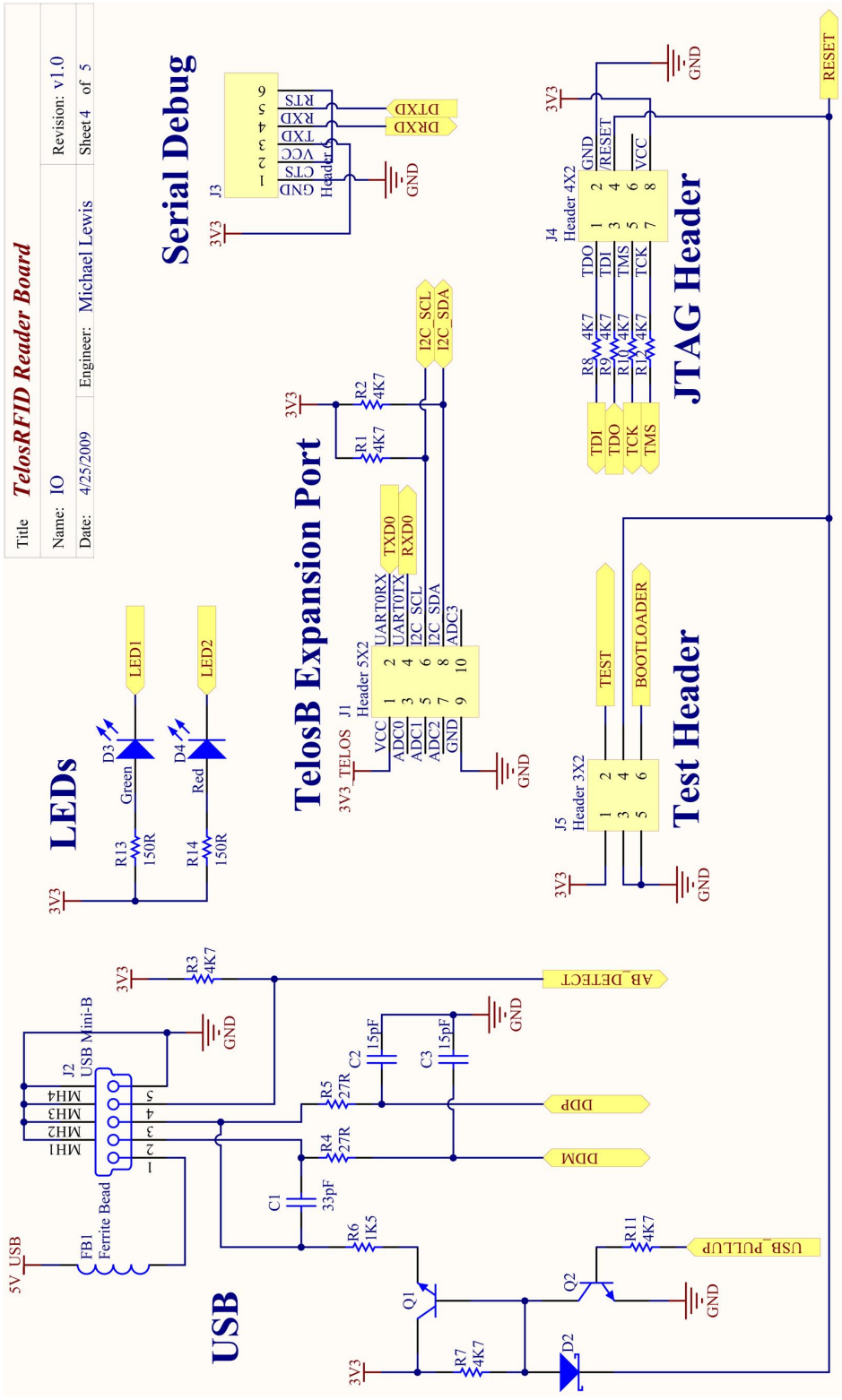

Figure 5.4: Design Schematic Page 4 of 5: I/O Circuitry

Power Circuitry

The power schematic shows the wiring of the TPS61202 and IRU1502-33 power ICs described in 5.3.1. It includes jumper pins which enable the user to configure where the TelosRFID reader board would draw its power from. Since it has been determined that TelosB sourced power does not perform as desired, these jumpers must be left connecting the board's power lines to the USB's 5.0V power source.

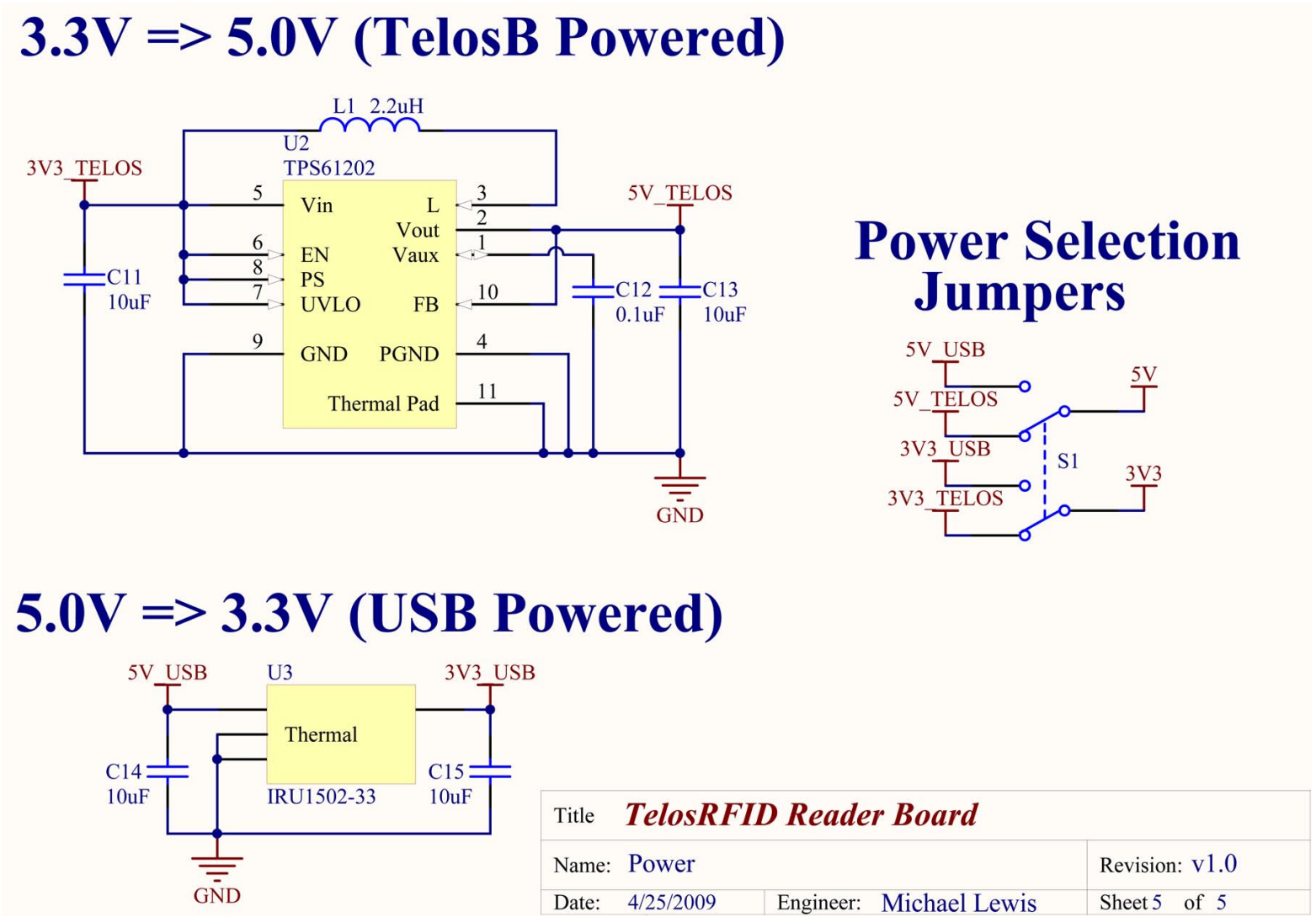

Figure 5.5: Design Schematic Page 5 of 5: Power Circuitry

5.3.3 Board Layout

The TelosRFID Reader Board is a 4-layer PCB measuring $2\frac{5}{8}$in $\times$ $1\frac{15}{16}$in. The 4 layers, Top, Ground, Power, and Bottom, are described in more detail in the following layout diagrams.

One interesting characteristic of the board that is not obvious from the previous schematics is that the antenna is actually formed using a copper PCB trace that surrounds the functional circuit. The active portion of the antenna resides in the middle two layers of the board. Each layer includes two loops of the antenna, for a total of four loops. The top and bottom layer each include shields enclosing the active portion of the antenna. This helps to reduce electrical noise produced by the board's components from corrupting the wireless signal emitted and recieived by the antenna.

The board itself was manufactured by Sunstone Circuits[66]. Sunstone Circuits accepted Gerber output files of the electronic design and manufactured a 4-sided PCB to the required specifications. Once manufactured, the naked board, along with all necessary components, was sent to Screaming Circuits[67] for PCB assembly. Screaming Circuits also accepted Gerber files of the design and used them to accurately populate the PCB with the components that they received.

Top Layer

The top layer image includes the silkscreen layer, indicating the placement of components on the board. All components are placed on the top, to avoid the complexity and cost of double-sided board population. The top layer can be divided into two logical sections. The center portion of the circuit consists of copper traces used to connect components together. The trace that borders the circuit, however, is actually part of the ground net, and serves to shield the RFID antenna that resides in the middle two layers.

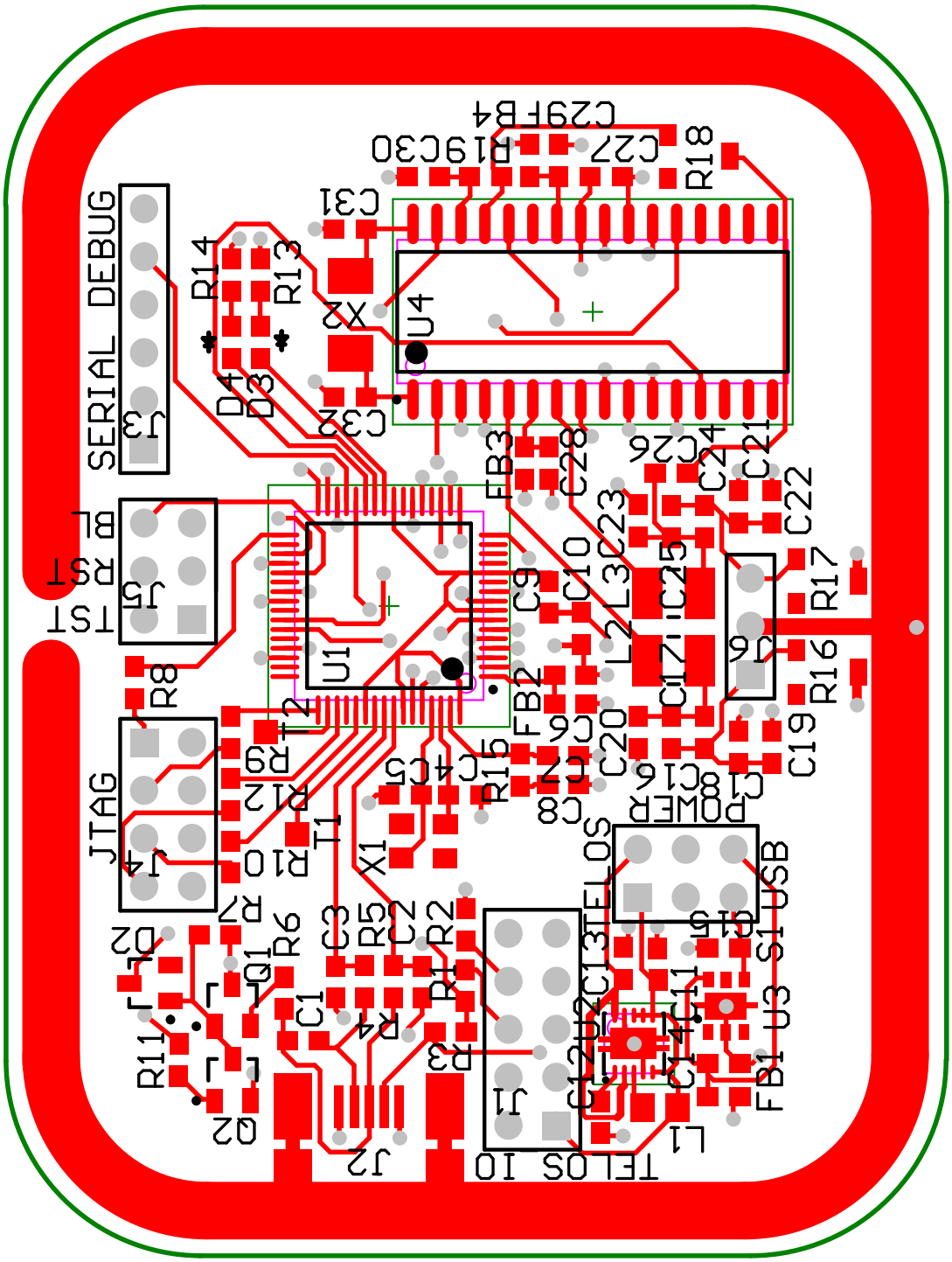

Figure 5.6: PCB Layout Page 1 of 4: Top Layer

Ground Layer

The layer just below the top layer in the board is the ground layer. The ground layer can be divided into two logical sections as well. The center portion of the circuit forms a ground plane that is used underneath the entire component portion of the board. This provides shielding between the top and bottom layers for high frequency signal traces, as well as providing a low resistance, loop free means of grounding all components at the same voltage potential. This low resistance is important for reducing electrical noise within the board. The trace that borders the circuit, although logically part of the ground net, is actually one half of the 13.56 MHz antenna used to communicate with RFID tags. This trace was designed using knowledge from an application note by NXP[49].

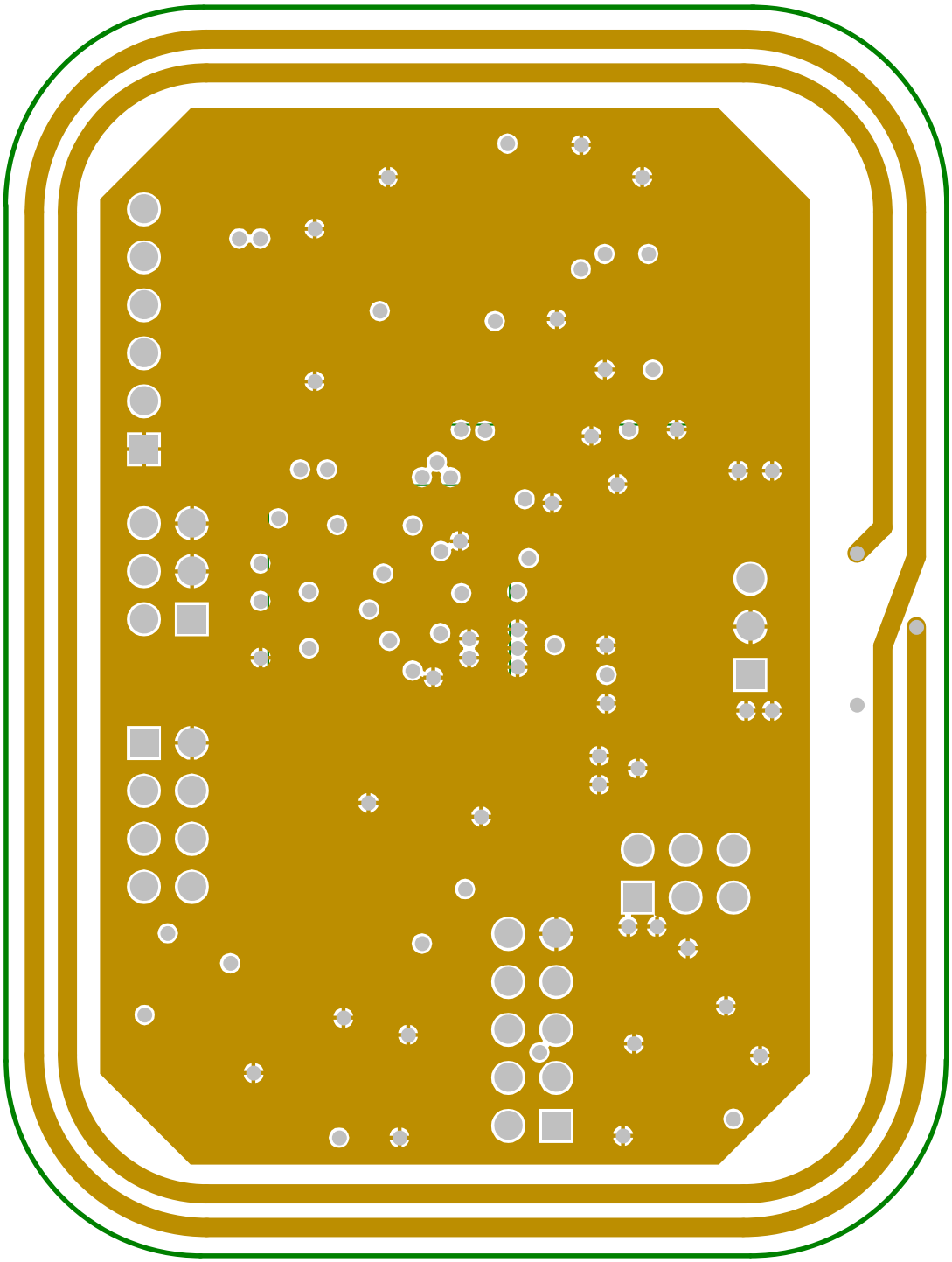

Figure 5.7: PCB Layout Page 2 of 4: Ground Layer

Power Layer

The layer just below the ground plane in the board is the power layer. The power layer can also be divided into two logical sections. The center portion of the circuit forms a 3.3V power plane that is used underneath the entire component portion of the board. This provides shielding between the top and bottom layers for high frequency signal traces, as well as providing a low resistance, loop free means of powering all 3.3V components at the same voltage potential. This low resistance is important for reducing electrical noise within the board. The trace that borders the circuit, although logically part of the ground net, is actually the second half of the 13.56 MHz antenna used to communicate with RFID tags. This trace was designed using knowledge from an application note by NXP[49].

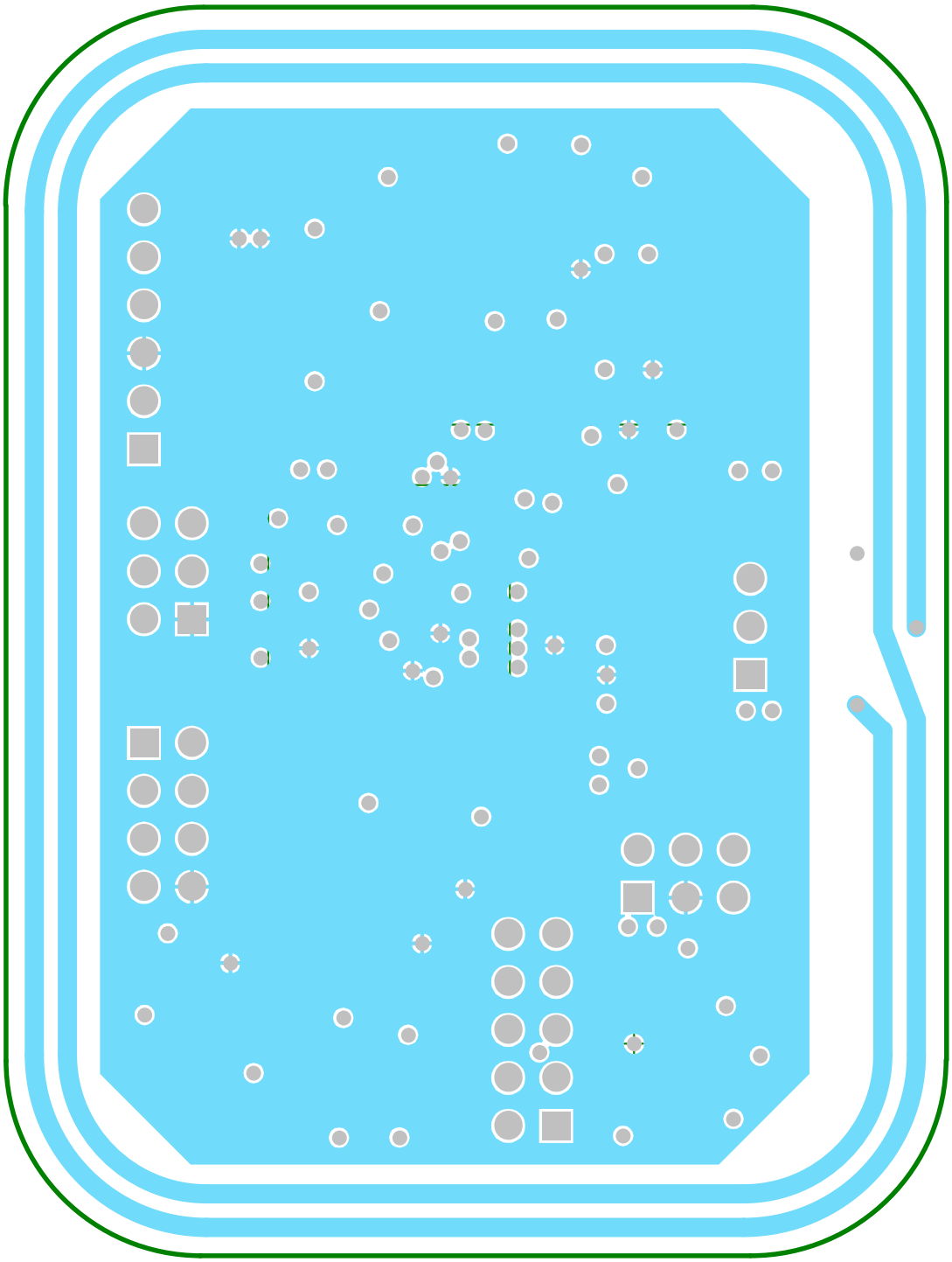

Figure 5.8: PCB Layout Page 3 of 4: Power Layer

Bottom Layer

The layer on the underside of the board, the bottom layer, can be divided into two logical sections. The center portion of the circuit consists of copper traces used to connect components together. The trace that borders the circuit is actually part of the ground net, and serves to shield the RFID antenna that resides in the middle two layers.

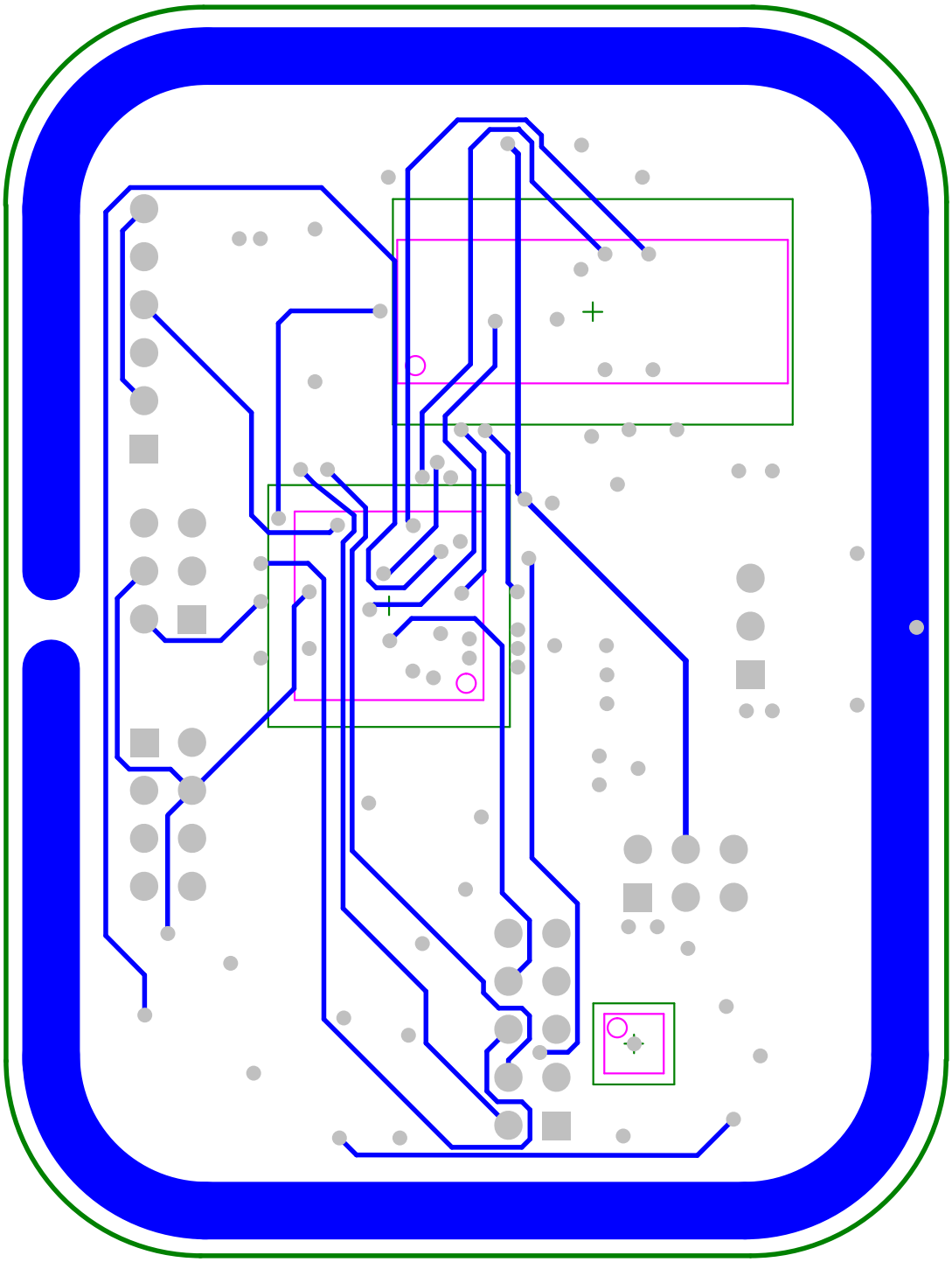

Figure 5.9: PCB Layout Page 4 of 4: Bottom Layer

5.4 Telos rev. B Mote

The Telos rev. B (*TelosB*) device is a freely available wireless sensor mote[68]. It was designed by the University of California, Berkeley, and subsequently released to the public as a free design[69]. Implemented physical boards are available from several manufacturers[70, 42] and are readily available as a commercial off the shelf (*COTS*) product.

The TelosB mote was originally intended to perform the role of a sensor data capture device. It has built in sensors for monitoring temperature, light, and humidity, as well as including an expansion port to facilitate the use of additional third party or custom sensor module[42]. This sensor data is generally then transmitted over the built-in IEEE 802.15.4 compliant wireless radio[71]. All of this functionality is controlled by a low-power 16-bit MSP430F1611 microcontroller, manufactured by Texas Instruments[72, 73]. The mote itself can be programmed from a PC, using a USB port.

Chapter 6

Software/Firmware Design

6.1 Academic Contributions

As part of this thesis, the TelosRFID project involves several software components that are unique academic contributions.

The TelosRFID Reader Board firmware is a suite of original embedded software for the Atmel AT91SAM7S256 microcontroller. It's source code was written in the ANSI C programming language. It was built as a collaboration of several custom software modules, structured to make use of the low-level open-source library *at91lib*[74], released by the Atmel Corporation®. Several unique firmware contributions are delivered in support of this thesis. The most significant of these contributions is an original ISO/IEC 14443 RFID communication stack implementation, a tag pressence management methodology, and an iterrupt-driven wired communication protocol design.

The reader-side TelosB firmware is a suite of original embedded firmware for the Texas Instruments MSP430 microcontroller. The firmware source code was written in the NesC programming language. It was built as a collaboration of several custom software modules, structured on top of the TinyOS[75] framework and operating system. The TinyOS development ecosystem was produced by the University of California, Berkeley. The unique contribution built into this framework is an original interrupt-driven wired communications control protocol permitting the TelosB mote to communicate with the TelosRFID

reader board, without corrupting wireless communications. Additionally, this communication protocol uses the TinyOS ZigBee library in order to broker messages between the TelosRFID reader and the ZigBee wireless network.

The client-side TelosB software is not an original contribution. The client-side TelosB uses the BaseStation application that is included in the standard TinyOS distribution.

The client-side application software is an original Java program that communicates with the attached TelosB mote. It leverages TinyOS libraries in order to read mote traffic and display RFID detection data into a console on the client PC's screen.

6.2 TelosRFID Reader Board Firmware

The TelosRFID firmware resides in the flash memory of the AT91SAM7S256 microcontroller. It is responsible for coordinating all functionality on the reader board. The firmware itself is written in C, and is debugged, compiled and linked through the YAGARTO tool chain[76]. It is responsible for several tasks.

The TelosRFID firmware controls the RFID air protocol implementation. It realizes the ISO-IEC 14443-2 and ISO/IEC 14443-3 protocol logic necessary to communicate with MIFARE tags. The firmware accomplishes this through digital control of the CL RC632 IC. A CL RC632 driver was written to interface with the RF frontend. It consists of several layers, from basic read/write of registers, up to higher-level functions, such as configuring the RF control settings. Built on top of the RC632 driver is an ISO/IEC 14443 driver. This protocol driver leverages the low level control functions of the RC632 driver in order to implement the ISO/IEC 14443 RFID air protocol, including tag detection and anti-collision.

The TelosRFID firmware also controls other onboard functionality. It controls board initialization, including setting up the microcontroller's ports, the onboard timer module and drivers, and printing out boot diagnostics via the DBGU port. It produces the green LED heartbeat, and toggles the red LED during tag detection. Additionally, it runs the routine which continuously scans the air for nearby tags, and reports its findings to the RFID tag presence monitoring module (*TagMonitor*). The TagMonitor module was written for this thesis in order to keep track of what tags are present near the reader. When a tag first enters the reader's field it is reported to the TagMonitor, and a "tag found" message is sent to the central PC. This is considered the tag *entering* the reader's proximity. The air is then continuously scanned for tags, every time the original tag reports back, its ID is sent to the tag monitor. As long as the tag's ID keeps being reported to the TagMonitor, it is considered *present*. If the tag does not report in for 200ms, then it is assumed that the tag left the reader's field and is no longer present. This results in a "tag left" message being sent to the central PC.

To facilitate communication with the attached TelosB mote, a driver module named

TelosComm was created. The purpose of this driver is to abstract away some of the complexities of communicating with the TelosB mote. When a message is ready to be sent, it pulls the TelosB's interrupt line high. After pulling high, it waits 2ms for the TelosB to gain control of it's UART line before transmitting the tag detection message out of the TelosRFID board's DBGU port. The reason that the TelosB mote needs advanced warning is that its UART line is shared between the TelosRFID board and the TelosB's onboard radio. If the TelosB is communicating wirelessly, any communication from the TelosRFID board will corrupt communication with the radio and never reach the MSP430.

6.2.1 Implementation Overview

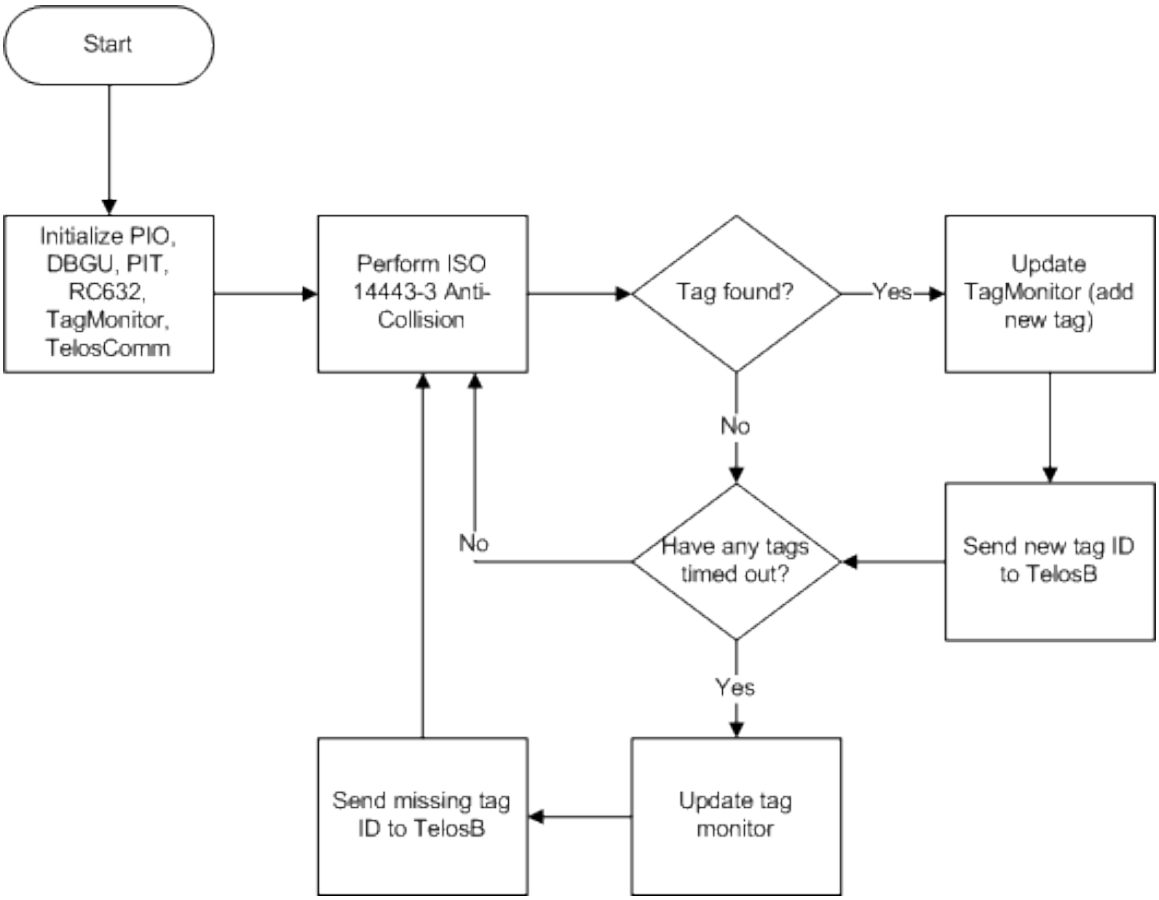

Figure 6.1: Main Loop in the TelosRFID Reader Board Software

The overall architecture for the TelosRFID software is relatively straightforward. It consists of an infinite loop that continuously scans for tags using the ISO/IEC 14443-3 anti-collision routine[47]. When a tag is detected, its ID is added to the tag presence management module, called TagMonitor. This module keeps track of how long it has been since each tag was seen. This allows tags to 'leave' the space by timing out. Whenever a tag enters or leaves the space a message is sent to the attached TelosB mote, so that it can relay that data to the ZigBee network.

Listing 6.1: TelosRFID main loop pseudo-code

```
/* Initialize the system */
initPorts()
initDebugSerialPort()
initTimer()
initRC632Driver()
initTagMonitor()
initTelosComm()
printBootInfo()
enableRxCircuit()

/* main loop repeats forever */
FOREVER {
  bool tagFound = scanForTag()
  if (tagFound) {
    tellTagMonitorAboutTag(newTagId)
    if (tagIsNew(newTagId)) {
      sendTagDetectionToTelosB(newTagId)
      toggleRedLED()
    }
  }

  updateTagMonitor()
  if (tagsHaveExpired()) {
    sendTagLossToTelosB(expiredTagId)
    toggleRedLED()
  }
}
```

6.2.2 Programming

The TelosRFID microcontroller can be programmed via its attached DBGU port. Firmware programming allows it to support custom 13.56MHz RFID protocols.

Programming is accomplished using the SAM-BA utility provided by Atmel[77]. In order to program the AT91SAM7S256, the following steps must be followed.

1.) Install Atmel's AT91-ISP v1.11 software[77] onto a Windows-based computer.

2.) From the files included with this thesis, copy the contents of /Thesis/Data Files/Software Design/TelosRFID SAM-BA 2.7 Programming Profile.zip to C:/Program Files/ATMEL Corporation/AT91-ISP v1.11/SAM-BA v2.7/lib

3.) Place a jumper across the pins labeled *TST*. This shorts the *TEST* net to 3.3V.

4.) Connect 5V power to the USB 2.0 port.

5.) Leave the device powered for 10 seconds. This loads bootloader code into the microcontroller's program execution flash memory space.

6.) Remove power, then remove the *TST* jumper.

7.) Reconnect 5V power to the USB 2.0 port.

8.) Connect the Serial Debug (*DBGU*) port on the board to a serial port on the programming PC.

9.) Launch the SAM-BA 2.7 software from the Atmel AT91-ISP v1.11 suite. Select the serial port that you are using and select TelosRFID as your board.

10.) Select 'Connect'.

11.) Select the *.bin file you wish to load in the 'Send File Name' dialog. Select the 'Send File' button.

12.) If dialog boxes appear, click yes to 'unlock', and then later 'lock', protected sections of flash memory.

13.) More advanced capabilities are explained in [78].

6.3 Reader-Side TelosB Software

The reader-side TelosB software resides in the flash memory of the Texas Instruments MSP430 microcontroller. Software for the reader-side TelosB mote was written in the NesC language[79] using the TinyOS 2.1.0 embedded framework[75]. It provides the reader-side TelosB mote with the capability to be a data hub by providing the following features.

- Communication with the attached TelosRFID board via UART and an interrupt line
- The ability to relay tag communication data from TelosRFID reader board to the ZigBee network
- Communication with a ZigBee network of TelosB motes

6.3.1 Implementation Overview

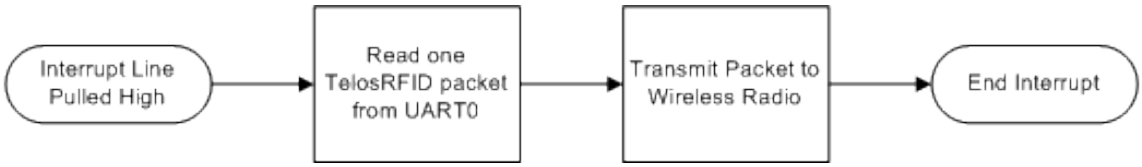

Figure 6.2: Data Interrupt Routine in the TelosB Reader Mote Software

Since TinyOS is an event based operating system, the program flow for this software is interrupt driven, unlike the loop-driven firmware in the TelosRFID reader board. When the interrupt line is pulled high by the TelosRFID board disables its radio, then reads its UART0 port and queues the packet. When it has the packet queued, it regains control of the radio IC in order to transmit the data out onto the ZigBee network. For testing and verification purposes, the red LED is toggled when an interrupt is recieved. When the packet is subsequently sent out the radio to the ZigBee network, the blue LED is also toggled. Verifying that both LEDs toggle at the same time demonstrates the successful arrival and retransmission of data from the MSP430 microcontroller.

Listing 6.2: Reader-side TelosB pseudo-code

```
/* intitalization code */
init() {
  initQueues()
  startRadio()
  setupInterruptPort()
}

/* interrupt handling routine */
interrupt_handler() {
  toggleRedLED()
  stopRadio()
  enableUARTListener()
  queueIncomingPacket()
  disableUARTListener()
  enableRadio()
  transmitPacketOutRadio()
  toggleBlueLED()
}
```

6.3.2 Programming

The official TinyOS website has excellent tutorials on how to compile and program firmware using the TinyOS tool chain[75].

6.4 Client-Side TelosB Software

The client-side TelosB software resides in the flash memory of the Texas Instruments MSP430 microcontroller. Software for the client-side TelosB mote was written in the NesC language using the TinyOS 2.1.0 embedded framework[75]. It provides the client-side TelosB mote with the capability to be a data hub by providing the following features.

- Communicate with a ZigBee network of TelosB motes
- Relay tag communication from the ZigBee network to the client PC

6.4.1 Implementation Overview

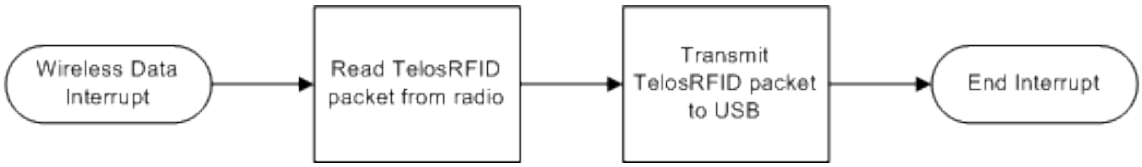

Figure 6.3: Data Interrupt Routine in the TelosB Client Mote Software

The TinyOS software package has a prebuilt program called BaseStation that fits the requirements for this mote exactly. It relays ZigBee network traffic through the USB port and to the client PC. This firmware is being used to channel ZigBee network packets to the attached PC for subsequent client-side processing. It is interrupt driven, waiting for data to become available before performing any tasks.

Listing 6.3: Client-side TelosB pseudo-code

```
1  /* USB -> Radio */
2  packetReceivedFromUSB(packet) {
3    transmitPacketOutRadio(packet)
4    toggleBlueLED()
5  }
6
7  /* Radio -> USB */
8  packetReceivedFromRadio(packet) {
9    transmitPacketOutUSB(packet)
10   toggleGreenLED()
11 }
```

6.4.2 Programming

The official TinyOS website has excellent tutorials on how to compile and program firmware using the TinyOS tool chain[75].

6.5 Client-Side Application Software

The client PC software provides the capability to view tag communications reported by the system. Written in Java, the primary role of this application is to read tag information from the attached client-side TelosB mote via a USB port. Tag information received at the client will be displayed on a computer screen so that the user knows when a tag has been detected. It displays pertinent information such as the ID of the tag read, the ID of the TelosRFID reader that saw the tag, and sequential message number.

6.5.1 Implementation Overview

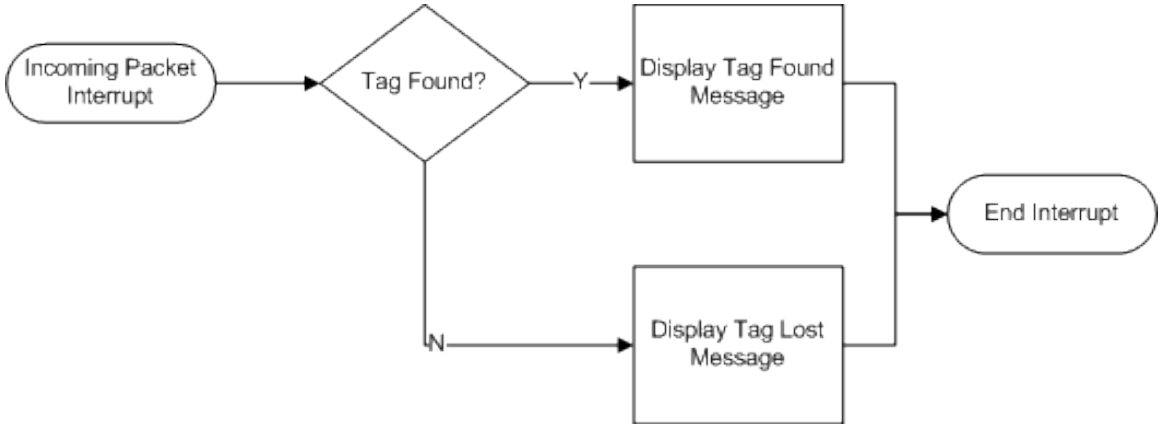

Figure 6.4: Interrupt Routine in the Java Client Software

The client-side Java application is event driven. It initializes a mote listening object, and registers a messageReceived() method to it. As incoming packets arrive, the message-handling method is called. This method is responsible for parsing the binary data sent from the TelosRFID reader board. Once parsed, it prints the data in a human-readable format into the console. The message format processing was done using generated code. The code was generated using TinyOS's *mig* tool on the custom *TelosRFID.h* file.

Listing 6.4: Client-side application pseudo-code

```
1  /* initialize the application */
2  main() {
3     validateInputParameters()
4     createMoteCommunicationObject()
5     registerCommunicationListener()
6  }
7
8  /* process messages as they come in */
9  messageReceived() {
10    parseBinaryPacket()
11    validatePacketContents()
12    formatAndDisplayPacketInfo()
13 }
```

Chapter 7

Results and Analysis

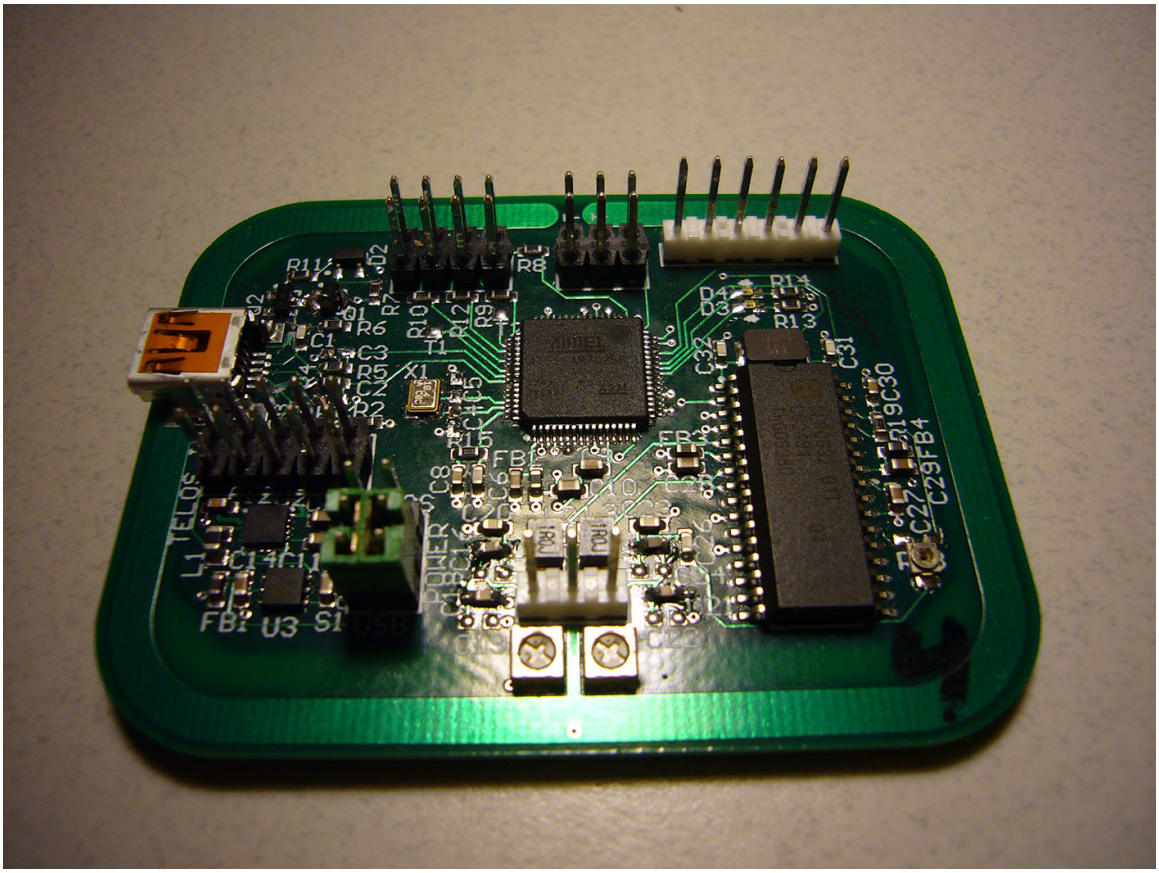

Figure 7.1: Completed TelosRFID Board

As each individual component was completed, it was integrated into the system as a whole. The TelosRFID reader board was the first device finished, being first developed as a standalone unit, and later implementing communication features. Following that, the

reader-side TelosB device firmware was written, debugged, and integrated with the reader board. After the reader subsystem was assembled, mote-to-mote communication was enabled using the TinyOS ZigBee stack. Finally, the client-side mote and client-side Java application were programmed to converse with each other. As each step was completed, the system slowly grew to completion.

Since the TelosRFID system consists of several components, analysis can be performed on each device individually, as well as on the system as a whole. Final thoughts, results, and analysis will be given for each significant contribution that this thesis makes.

7.1 TelosRFID Reader-side Subsystem

The TelosRFID reader-side subsystem ties together two distinct wireless technologies, 13.56MHz RFID and ZigBee. These technologies naturally complement each other from a technological, as well as systematic perspective. Technologically, ZigBee operates on top of the IEEE 802.15.4 protocol, which can function at 2.4GHz. Although RFID systems can operate at any number of frequencies, 13.56MHz was chosen because it is very far away from the TelosB's 2.4GHz. Due to the large difference between the spectrum bands, neither technology interferes with the other. From an end-user perspective, each technology behaves just as well as if it had been implemented by itself.

Systematically, RFID and ZigBee complement each other nicely. Each technology fills a gap in the offerings of the other, resulting in a more capable and robust system. RFID's ability to read passive, low-cost tags has lead to acceptance for industrial tracking applications. It's short range, however, necessitates the establishment of infrastructure in order to effecively deploy it on a widespread scale. ZigBee fills this need by providing longer range, paired with advanced ad-hoc networking features. The fact that ZigBee can only communicate with powered networking devices is irrelevant, since that need to read passive devices is handled by the RFID technology.

7.1.1 TelosRFID Reader Board

The TelosRFID reader board functions as described. When an ISO/IEC 14443 13.56 MHz RFID tag is placed next to the board, the tag's ID is detected by the system. Additionally, multiple tags can be differentiated from each other using the ISO/IEC 14443-3 anti-collision algorithm. When a tag is either found or lost, the board's red LED toggles and the detection data is sent via the DBGU port. Apart from the system, this can be verified by connecting the DBGU port to a computer's serial port, configured to display raw hexidecimal data at 112500 baud.

A rather noticable property of the board is that it's range is rather short. Detections are reliably made from just over 1cm from the antenna, which is less than expected. There are likely several contributing causes for this disappointing result. This first, and probably most significant, is a poorly tuned antenna tuning filter. Due to a lack of proper equipment and expertise available, the final filter is functional, but likely sub-optimal. Another possible cause of the limited range could be the CL RC632 IC used. A higher power or lower noise analog front end would most likely have a positive direct impact on the RFID range of the reader board.

Due to post-manufacturing design adjustments, the reader board does not fit directly onto the TelosB mote, but instead uses an adapter cable. Although the adapter cable works flawlessly, it would be beneficial for packaging the reader-side subsystem if they could connect directly. This would reduce costs, as well as the footprint, of the completed assembly.

7.1.2 Reader-side TelosB Mote

The TelosB device performed its role perfectly. Is served as an easily programmable ZigBee device. The ZigBee features enabled the construction of a low-power ad-hoc network. This network was used to reliably communcate with the client PC, providing it with updates concerning the state of RFID tag detections from its attached TelosRFID reader board. The

tag detections were read successfully from the reader board using a custom communications protocol. The custom protocol was required because the TelosB mote's radio and UART line share the same ports.

7.2 TelosRFID Client-side Subsystem

The TelosRFID client-side subsystem provides a user interface for the technological operations occurring on the reader-side. By displaying tag detection events, a system administrator is kept aware of tho state of the system. If the TelosRFID system were to be integrated into a larger system, it would be integrated at the PC section of the TelosRFID system. Such integration may be beneficial for extending the functionality. Enabling database comparison of tag IDs would permit the implementation of object tracking or access management applications.

7.2.1 Client-side TelosB Mote

As with the reader-side TelosB mote, its client-side counterpart performed as expected. It provided easily implementable ZigBee functionality. With a range of up to 250 feet, it can control a sizeable network of motes. The ability to transmit data to the attached PC via its USB port was essential to the success of the client-side subsystem.

7.2.2 Client-side Java Application

The linux operating system, paired with the TinyOS programming framework, proved to be an effective developing environment for the client-side Java application. The TinyOS framework provided classes to enable communication with an attached TelosB mote. Additionally, it provided a utility to simplify message parsing by reading the message format from the NesC code used to program the reader-side TelosB mote's firmware.

7.3 TelosRFID System

Whenever a tag is placed next to the reader board, a message prints to the PC's console, informing the user of the tag's ID, as well as the serial number of the reader. When the tag is taken away from the reader, another message is printed, this one also including the tag's ID an the reader's serial number. Each message displayed includes information regarding if the tag was just lost or found.

The system, as a whole, functions as expected. Despite some difficulties during the implementation phase, the system satisfies all requirements laid out in 4. Although far from a commercial product, it demonstrates the necessary attributes needed to implement a demo system. With a basic feature-set, it provides the minimum framework needed for the system to prove its utility and functionality. Future development in the form of additional RFID air protocols and networking features would greatly expand the usefulness of the system.

Chapter 8

Conclusion

The TelosRFID system provides the initial implementation of a complete ZigBee and RFID ecosystem. It combines both technologies in order to get the best of each and to supplement the weaknesses of each. The system is provided in an open, research-friendly manner in order to encourage ongoing development. Continued research is supported through the reprogrammability of device firmware and client software.

By providing programmable control over two widely researched technologies, flexibility is created. Not only can each technology be manipulated and extended to the researcher's desire, but the possibilities created by synergizing both expands exponentially. The most flexible portion of the system is the ability to read RFID tags. Because the TelosRFID reader board's firmware is open-source and reprogrammable, the implementation of custom 13.56MHz RFID air protocols is possible. The ability to test custom air protocols enables research to be taken from theory to application.

The ability to study process and system engineering is enabled by the ZigBee capabilities of the system. From the perspective of a system administrator, the ability to monitor numerous remote points, without the cost of infrastructure deployment is appealing. A rapidly deployable network of tag monitoring devices can be used to track object movement and process flow.

The most obvious extensions to this research are to extend limitations of the existing implementation. The TelosRFID reader board would benefit from a hardware rework. The

antenna tuning filter needs adjustment in order to improve the range of the tag reader. Additionally, the IO port on the reader board should be re-arranged to permit direct connection of the reader board and its attached TelosB mote, thus doing away with the adapter cable.

From a research perspective, several iterative improvements upon the system can be suggested. The first is the implementation of other RFID air protocols. The development of a library of protocols would make headway on the goal of a ”universal RFID reader”, at least for 13.56MHz tags. Expanding upon the ZigBee aspect of the system, increasing the number of reader-side subsystems would increase the scale of the system greatly. If a ”swarm” of reader-side sub-systems were created and networked, localization algorithms could be employed to track movement within a network of scattered readers.

Beyond the intended research extensions, the fact that every device in the system can be reprogrammed opens even greater possibilites. The research community is continuously researching innovate new ways to extend existing technology. It would not be inconceivable for other academics to apply this system to problems that the author never knew existed or even envisioned.

Both ad-hoc networking and RFID are technologies that, while experiencing rapid industrial adoption, are also the topics of significant academic research. Because of our society’s increasing dependence upon convenience and security, often conflicting goals, improvements in these technologies are always being developed. The TelosRFID system provides researchers with an innovate tool for exploring these technological improvements.

Appendix A

TelosRFID Reader Board Source Code

The following source files were created or modified in support of this thesis. They are required to compile firmware for the TelosRFID reader board. Also required to compile the firmware is Atmel's at91lib library[74].

- at91lib/boards/telosrfid/board.h
- at91lib/peripherals/pio/pio.h
- at91lib/peripherals/pio/pio.c
- at91lib/peripherals/pit/pit.h
- at91lib/peripherals/pit/pit.c
- at91lib/peripherals/spi/spi.c
- at91lib/utility/debug.h
- at91lib/utility/debug.c
- at91lib/utility/trace.h
- at91lib/std.h
- rc632/rc632.h
- rc632/rc632.c
- rc632/rc632_defines.h
- rc632/rc632_iso14443a.h
- rc632/rc632_iso14443a.c
- rc632/rc632_macros.h

- rc632/rc632_macros.c
- rc632/rc632_mifare.h
- rc632/rc632_mifare.c
- heartbeat.h
- heartbeat.c
- init.h
- init.c
- isr_handlers.h
- isr_handlers.c
- main.h
- main.c
- tag_monitor.h
- tag_monitor.c
- telos_comm.h
- telos_comm.c
- time.h
- time.c
- Makefile.make

Appendix B

Reader-side TelosB Source Code

The following source files were created or modified in support of this thesis. They are required to compile firmware for the client-side TelosB mote. Also required to compile the firmware is the TinyOS framework and libraries[75].

- RfidReaderC.nc
- RfidReaderP.nc
- TelosRFID.h
- TelosRFIDReader.h
- TelosRFIDReaderAppC.nc
- TelosRFIDReaderC.nc
- Makefile.make

Appendix C

Client-side Java Application Source Code

The following source files were created or modified in support of this thesis. They are required to compile and run software for the client-side console application. Also required to compile the firmware is the TinyOS framework and libraries[75].

- TelosRFID.h
- TelosRFIDApp.java

Bibliography

[1] Lee, H. and Kim, J. Privacy threats and issues in mobile rfid. *Availability, Reliability and Security, 2006. ARES 2006. The First International Conference on*, April 2006. doi:10.1109/ARES.2006.96.

[2] Xiao, Q., Boulet, C., and Gibbons, T. Rfid security issues in military supply chains. *Availability, Reliability and Security, 2007. ARES 2007. The Second International Conference on*, pp. 599–605, April 2007. doi:10.1109/ARES.2007.127.

[3] Jiang, W., Yu, D., and Ma, Y. A tracking algorithm in rfid reader network. *Frontier of Computer Science and Technology, 2006. FCST '06. Japan-China Joint Workshop on*, pp. 164–171, Nov. 2006. doi:10.1109/FCST.2006.7.

[4] Alfonsi, B. Privacy debate centers on radio frequency identification. *Security and Privacy, IEEE*, 2(2):12, Mar.-Apr. 2004. ISSN 1540-7993. doi:10.1109/MSECP.2004.1281237.

[5] Li, Y.Z., Cho, Y.B., Um, N.K., and Lee, S.H. Security and privacy on authentication protocol for low-cost rfid. *Computational Intelligence and Security, 2006 International Conference on*, 2:1101–1104, Nov. 2006. doi:10.1109/ICCIAS.2006.295432.

[6] Takahashi, D., Xiao, Y., Hu, F., and Lewis, M. A survey of insulin-dependent diabetes—part i: Therapies and devices. *International Journal of Telemedicine and Applications*, 2008, 2008.

[7] Graafstra, A. Hands on. *Spectrum, IEEE*, 44(3):18–23, Mar. 2007. ISSN 0018-9235. doi:10.1109/MSPEC.2007.323420.

[8] Gao, X., Xiang, Z., Wang, H., Shen, J., Huang, J., and Song, S. An approach to security and privacy of rfid system for supply chain. *E-Commerce Technology for Dynamic E-Business, 2004. IEEE International Conference on*, pp. 164–168, Sept. 2004. doi:10.1109/CEC-EAST.2004.14.

[9] Evers, L., Havinga, P., Kuper, J., Lijding, M., and Meratnia, N. Sensorscheme: Supply chain management automation using wireless sensor networks. *Emerging Technologies and Factory Automation, 2007. ETFA. IEEE Conference on*, pp. 448–455, Sept. 2007. doi:10.1109/EFTA.2007.4416802.

[10] Phillips, T., Karygiannis, T., and Kuhn, R. Security standards for the rfid market. *Security and Privacy, IEEE*, 3(6):85–89, Nov.-Dec. 2005. ISSN 1540-7993. doi: 10.1109/MSP.2005.157.

[11] Lee, B.N., Kim, Y.W., and Kim, H.J. Evolution of rfid applications and its implications: Standardization perspective. *Management of Engineering and Technology, Portland International Center for*, pp. 903–910, Aug. 2007. doi:10.1109/PICMET. 2007.4349409.

[12] Sandoval-Reyes, S. and Soberanes Perez, J. Mobile rfid reader with database wireless synchronization. *Electrical and Electronics Engineering, 2005 2nd International Conference on*, pp. 5–8, Sept. 2005. doi:10.1109/ICEEE.2005.1529560.

[13] Zhang, L. and Wang, Z. Integration of rfid into wireless sensor networks: Architectures, opportunities and challenging problems. *Grid and Cooperative Computing Workshops, 2006. GCCW '06. Fifth International Conference on*, pp. 463–469, Oct. 2006. doi:10.1109/GCCW.2006.58.

[14] Xu, B., Hischke, S., and Walke, B. The role of ad hoc networking in future wireless communications. *Communication Technology Proceedings, 2003. ICCT 2003. International Conference on*, 2:1353–1358 vol.2, April 2003. doi:10.1109/ICCT.2003. 1209779.

[15] Subramanian, S.P., Sommer, J., Schmitt, S., and Rosenstiel, W. Sbil: Scalable indoor localization and navigation service. *Wireless Communication and Sensor Networks, 2007. WCSN '07. Third International Conference on*, pp. 27–30, Dec. 2007. doi: 10.1109/WCSN.2007.4475741.

[16] Leong, K.S., Ng, M.L., Grasso, A., and Cole, P. Synchronization of rfid readers for dense rfid reader environments. *Applications and the Internet Workshops, 2006. SAINT Workshops 2006. International Symposium on*, Jan. 2006. doi:10.1109/ SAINT-W.2006.39.

[17] Faschinger, M., Sastry, C.R., Patel, A.H., and Tas, N.C. An rfid and wireless sensor network-based implementation of workflow optimization. *World of Wireless, Mobile and Multimedia Networks, 2007. WoWMoM 2007. IEEE International Symposium on a*, pp. 1–8, June 2007. doi:10.1109/WOWMOM.2007.4351732.

[18] Bacheldor, B. Hospital tries zigbee to track patients. *RFID Journal*, May 2006. URL `http://www.rfidjournal.com/article/articleview/2509/1/1/`.

[19] Sung, J., Sanchez Lopez, T., and Kim, D. The epc sensor network for rfid and wsn integration infrastructure. *Pervasive Computing and Communications Workshops, 2007. PerCom Workshops '07. Fifth Annual IEEE International Conference on*, pp. 618–621, Mar. 2007. doi:10.1109/PERCOMW.2007.113.

[20] SkyeTek. *SkyeModule M1 Datasheet*. SkyeTek, 2005. URL `http://www.skyetek.com/Default.aspx?tabid=82`.

[21] Landt, J. The history of rfid. *Potentials, IEEE*, 24(4):8–11, Oct.-Nov. 2005. ISSN 0278-6648. doi:10.1109/MP.2005.1549751.

[22] Weinstein, R. Rfid: a technical overview and its application to the enterprise. *IT Professional*, 7(3):27–33, May-June 2005. ISSN 1520-9202. doi:10.1109/MITP.2005.69.

[23] Preradovic, S. and Karmakar, N. Rfid readers - a review. *Electrical and Computer Engineering, 2006. ICECE '06. International Conference on*, pp. 100–103, Dec. 2006. doi:10.1109/ICECE.2006.355300.

[24] Vun, N. and Law, C.L. Development of an embedded based rfid front end system. *Consumer Electronics, 2006. ISCE '06. 2006 IEEE Tenth International Symposium on*, 2006. doi:10.1109/ISCE.2006.1689444.

[25] Safarian, A., Shameli, A., Rofougaran, A., Rofougaran, M., and De Flaviis, F. An integrated rfid reader. *Solid-State Circuits Conference, 2007. ISSCC 2007. Digest of Technical Papers. IEEE International*, pp. 218–598, Feb. 2007. ISSN 0193-6530. doi:10.1109/ISSCC.2007.373372.

[26] Birari, S. and Iyer, S. Mitigating the reader collision problem in rfid networks with mobile readers. *Networks, 2005. Jointly held with the 2005 IEEE 7th Malaysia International Conference on Communication., 2005 13th IEEE International Conference on*, 1, Nov. 2005. ISSN 1531-2216. doi:10.1109/ICON.2005.1635526.

[27] Lu, L., Han, J., Hu, L., Liu, Y., and Ni, L.M. Dynamic key-updating: Privacy-preserving authentication for rfid systems. *Pervasive Computing and Communications, 2007. PerCom '07. Fifth Annual IEEE International Conference on*, pp. 13–22, Mar. 2007. doi:10.1109/PERCOM.2007.13.

[28] Osaka, K., Takagi, T., Yamazaki, K., and Takahashi, O. An efficient and secure rfid security method with ownership transfer. *Computational Intelligence and Security, 2006 International Conference on*, 2:1090–1095, Nov. 2006. doi:10.1109/ICCIAS.2006.295430.

[29] Park, N., Kim, H., Chung, K., and Sohn, S. Design of an extended architecture for secure low-cost 900mhz uhf mobile rfid systems. *Consumer Electronics, 2006. ISCE '06. 2006 IEEE Tenth International Symposium on*, 2006. doi:10.1109/ISCE.2006.1689490.

[30] Mason, A., Shaw, A., and Al-Shamma'a, A. Intelligent radio frequency identification positioning using wireless sensor networks. *Antennas and Propagation Conference, 2007. LAPC 2007. Loughborough*, pp. 145–148, April 2007. doi:10.1109/LAPC.2007.367452.

[31] Want, R. An introduction to rfid technology. *Pervasive Computing, IEEE*, 5(1):25–33, Jan.-Mar. 2006. ISSN 1536-1268. doi:10.1109/MPRV.2006.2.

[32] Chawla, V. and Ha, D. An overview of passive rfid. *Communications Magazine, IEEE*, 45(9):11–17, Sept. 2007. ISSN 0163-6804. doi:10.1109/MCOM.2007.4342873.

[33] Ith, P., Oyama, Y., Inomata, A., and Okamoto, E. Implementation of id-based signature in rfid system. *Communications, 2007. APCC 2007. Asia-Pacific Conference on*, pp. 233–236, Oct. 2007. doi:10.1109/APCC.2007.4433543.

[34] Hartmann, C. and Claiborne, L. Fundamental limitations on reading range of passive ic-based rfid and saw-based rfid. *RFID, 2007. IEEE International Conference on*, pp. 41–48, Mar. 2007. doi:10.1109/RFID.2007.346148.

[35] Alliance, Z. Zigbee alliance – home page. Electronic, May 2008. URL `http://www.zigbee.org/`.

[36] Alliance, Z. How does zigbee compare to other wireless standards? Electronic, May 2008. URL `http://www.zigbee.org/en/about/faq.asp#17`.

[37] Chen, B., Wu, M., Yao, S., and Binbin, N. Zigbee technology and its application on wireless meter-reading system. *Industrial Informatics, 2006 IEEE International Conference on*, pp. 1257–1260, Aug. 2006. doi:10.1109/INDIN.2006.275820.

[38] Sveda, M. and Trchalik, R. Zigbee-to-internet interconnection architectures. *Systems, 2007. ICONS '07. Second International Conference on*, pp. 30–30, April 2007. doi: 10.1109/ICONS.2007.58.

[39] Adams, J. An introduction to ieee std 802.15.4. *Aerospace Conference, 2006 IEEE*, pp. 8 pp.–, Mar. 2006. doi:10.1109/AERO.2006.1655947.

[40] ZigBeeAlliance. Zigbee specification. Tech. rep., ZigBee Alliance, 2008.

[41] Evans-Pughe, C. Bzzzz zzz [zigbee wireless standard]. *IEE Review*, 49(3):28–31, March 2003. ISSN 0953-5683.

[42] Crossbow. Telosb data sheet. Tech. rep., Crossbow, 2009.

[43] Energizer. Energizer aa e91 product datasheet, 2009. URL `http://data.energizer.com/PDFs/e91.pdf`.

[44] Sarkinen, J. Decentralized power, need for smarter products, total cost of ownership, and security. *Telecommunications Conference, 2005. INTELEC '05. Twenty-Seventh International*, pp. 131–136, Sept. 2005. doi:10.1109/INTLEC.2005.335081.

[45] Reddy, J.S. Zigbee security layer technical overview. Tech. rep., ZigBee Alliance, 2004.

[46] Lee, J.G., Hwang, S.J., Kim, S.W., Ahn, S., Park, K., Koo, J.H., and Kang, W.S. Software architecture for a multi-protocol rfid reader on mobile devices. *Embedded Software and Systems, 2005. Second International Conference on*, Dec. 2005. doi: 10.1109/ICESS.2005.86.

[47] ISO/IEC. Iso/iec 14443-3: Initialization and anticollision. Tech. rep., International Organization for Standardization, 2006.

[48] Atmel. At91sam7s-ek evaluation board user guide. Tech. rep., Atmel Corporation, 2007. URL `http://www.atmel.com/dyn/resources/prod_documents/doc6112.pdf`.

[49] NXPSemiconductors. Micore reader ic family; directly matched antenna design. Tech. rep., NXP Semiconductors, 2006.

[50] TexasInstruments. Tps61200, tps61201, tps61202 low input voltage synchronous boost converter with 1.3-a switches. Tech. rep., Texas Instruments, 2008.

[51] TrossenRobotics. Trossen robotics store, 2009. URL `http://www.trossenrobotics.com`.

[52] NXPSemiconductors. Mifare.net, 2009. URL `http://mifare.net`.

[53] ISO/IEC. Iso/iec 14443-1: Physical characteristics. Tech. rep., International Organization for Standardization, 2008.

[54] ISO/IEC. Iso/iec 14443-2: Radio frequency power and signal interface. Tech. rep., International Organization for Standardization, 2007.

[55] ISO/IEC. Iso/iec 14443-4: Transmission protocol. Tech. rep., International Organization for Standardization, 2008.

[56] Legic. Legic avant crypto transponder chips. Tech. rep., Legic, 2006.

[57] Calypso. Functional card application. Tech. rep., Calypso, 2005.

[58] HIDGlobal. 13.56 mhz contactless smart card systems, 2009. URL `http://www.hidglobal.com`.

[59] Atmel. Atmel at91sam7sx data sheet. Tech. rep., Atmel Corporation, 2007.

[60] Philips. Cl rc632 multiple protocol contactless reader ic short form specification. Tech. rep., Philips Semiconductors, 2005.

[61] Philips. Cl rc632 multiple protocol contactless reader ic long form specification. Tech. rep., Philips Semiconductors, 2005.

[62] NXPSemiconductors. mifare ultralight features and hints. Tech. rep., NXP Semiconductors, 2006.

[63] Philips. mifare (14443a) 13.56 mhz rfid proximity antennas. Tech. rep., Philips Semiconductors, 2002.

[64] Philips. Pegoda contactless smart card reader. Tech. rep., Philips Semiconductors, 2004.

[65] Fotopoulou, K. and Flynn, B. Optimum antenna coil structure for inductive powering of passive rfid tags. *RFID, 2007. IEEE International Conference on*, pp. 71–77, Mar. 2007. doi:10.1109/RFID.2007.346152.

[66] SunstoneCircuits. Sunstone circuits, 2009.

[67] Circuits, S. Screaming circuits, 2009.

[68] Polastre, J., Szewczyk, R., and Culler, D. Telos: Enabling ultra-low power wireless research. Tech. rep., University of California, Berkeley, 2005.

[69] University of California, B. Telos rev b hardware design files, 2004. URL `http://www.tinyos.net/scoop/special/hardware`.

[70] MoteivCorporation. Tmote sky datasheet. Tech. rep., Moteiv Corporation, 2006.

[71] Kiing-Ing, W. A light-weighted, low-cost and wireless ecg monitor design based on tinyos operating system. *Information Technology Applications in Biomedicine, 2007. ITAB 2007. 6th International Special Topic Conference on*, pp. 165–168, Nov. 2007. doi:10.1109/ITAB.2007.4407370.

[72] TexasInstruments. Msp430x1xx family user's guide. Tech. rep., Texas Instruments, 2006.

[73] TexasInstruments. Msp430x15x, msp430x16x, msp430x161x mixed signal microcontroller data sheet. Tech. rep., Texas Instruments, 2002.

[74] Atmel. At91 software package release notes, April 2008. URL `http://www.atmel.com/dyn/resources/prod_documents/softpack-1.4-releasenote.txt`.

[75] University of California, B. Tinyos community forum, 2009.

[76] Fischer, M. Yagarto - yet another gnu arm toolchain, 2009.

[77] Atmel. Atmel products - tools and software, 2009. URL `http://www.atmel.com/dyn/products/tools_card.asp?tool_id=3883`.

[78] Atmel. Sam boot assistant (sam-ba) user guide. Tech. rep., Atmel Corporation, 2006.

[79] NesC. nesc: A programming language for deeply networked systems, May 2008. URL http://nescc.sourceforge.net/.