Using ISO 15693 Compliant RFID Tags in an Inventory Control System

University: Louisiana State University, Baton Rouge, Louisiana

Course: Undergraduate Capstone Project

Student Team Members: Joseph Gates, Vishwajeet Potnis, Matthew Howell, William Tran

Faculty Advisor: Mr. John Scalzo

Abstract:

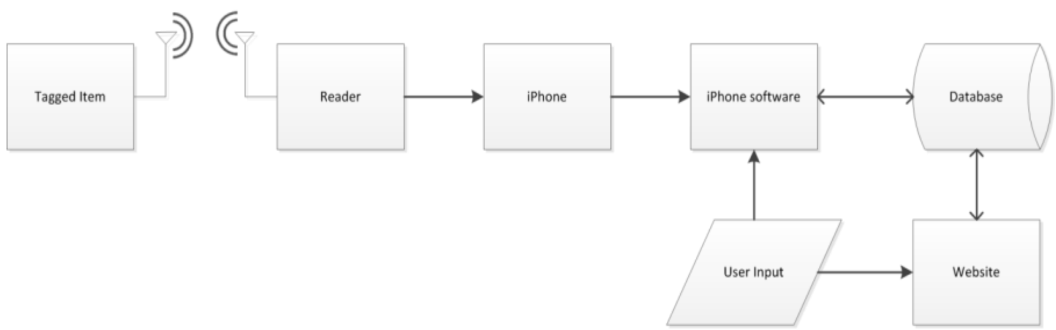

Figure 1: Top-level system summary

For an inventory control system to be effective, it must be scalable, cost-efficient, and easy to use. Currently, commonly used inventory control systems tend to be lacking in these areas, making the inventory process a time-consuming endeavor. For this project, an analysis of the inventory management system currently in place at Louisiana State University was used to define several goals for a new inventory system design. LSU's current system, as used by property management and the various departments of LSU, is an arduous process, involving printed notifications sent to each department listing the complete inventory of the department, the use of an aging database to store and access the item and department information, and manual entry of information into and retrieval of information from this database using an outdated interface. The complete inventory process currently requires a full year to complete. The major goal of this project is to streamline this process.

At the highest level, the inventory control system designed for this project can be described as 5 distinct parts: tagged items, reader, iOS application, database, and web application. The tags used to identify each item are passive RFIDs operating at 13.56MHz. The reader communicates with the tags, and the tags respond with their unique codes modulated onto this carrier frequency. Hardware on the reader demodulates and decodes the signal, and the resulting data is interfaced to an iOS device serially through the 3.5mm headphone jack on the device. The iOS device will be programmed, using the iOS programming interface, to interpret the data received through the headphone jack and to provide a simple user interface. A MySQL database will be used to store item, room, department, and user information. The device will maintain a persistent connection to the database using either Wi-Fi or 3G in order to provide real-time results while scanning items. Finally, a web site will be designed and implemented using Apache and PHP to provide administrative access to the database, as well as analyst access for generation of reports and notifications.

ISO Standard:

The ISO 15693 standard defines the operation of vicinity cards (passive RFID tags) operating at a frequency of 13.56MHz. Part 1 of the standard describes the physical characteristics of tags, part 2 describes the air interface and initialization between a tag and a reader, and part 3 describes the transmission protocol. An ISO 15693-2 and ISO 15693-3 compliant RFID tag from Texas Instruments is used to tag each item in the inventory system. Each tag has a 64-bit unique identification number (UID) and 256 bits of user programmable memory arranged is 8-bit x 32-bit blocks.

Since the RFID tags are passive, power transfer is achieved by coupling the magnetic field in the antenna of the reader with the antenna of the tag. The field required to activate the tag can range between 150mA/m to 5A/m.

Communication between the tag and the reader is a three step process. The tag is first activated by the 13.56MHz RF field of the reader, and then begins to wait for a command. The reader then transmits a command to the tag and awaits a response. If a valid command is received by the tag, it determines the appropriate response and transmits it back to the reader. All data transmission is achieved through modulation of the 13.56MHz carrier frequency.

For communication from the reader to the tag, data is modulated using amplitude shift keying (ASK) using either 10% or 100% modulation depths. Data is coded using either 1 out of 4 or 1 out of 256 pulse position modulation and is delimited by start of frame and end of frame signals. For communication from the tag to the reader, data is modulated using a subcarrier with Manchester coding. More details can be found in sections 7 and 8 of the ISO-15693-2 standard.

In order to obtain the unique identification number (UID) of an RFID tag, the correct command must be sent to the tag, in accordance with the ISO standard, specifically sections 7.2.2, 7.3, 7.4, 10, and annex C of the ISO-15693-3 standard. For a basic inventory command, 40 bits must be transmitted to the tag. The first 8-bits in the sequence define the flags sent to the tag that specify the actions to be performed by the tag. The next 8-bits specify the 'inventory' command followed by an 8-bit mask. A 16-bit CRC is then transmitted for error checking. All bytes are transmitted with the least significant bit first.

Hardware:

Antenna:

The antenna used to couple the magnetic field produced by the reader with the antenna built into the tag was created on a PCB designed for this project. The artwork used for this board produced an antenna with a calculated inductance of about 1µH. A variable capacitor and two fixed value capacitors were added to create a tank circuit with a resonance frequency of 13.56MHz. Another network of capacitors was added in series to the circuit to facilitate tuning the input impedance of the antenna to 50Ω. The completed antenna schematic is shown in figure 2, and the antenna board itself in figure 3.

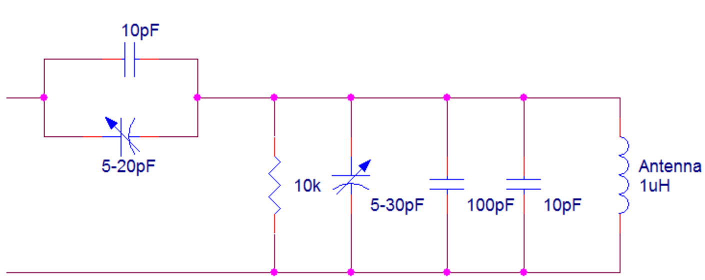

Figure 2: Antenna schematic

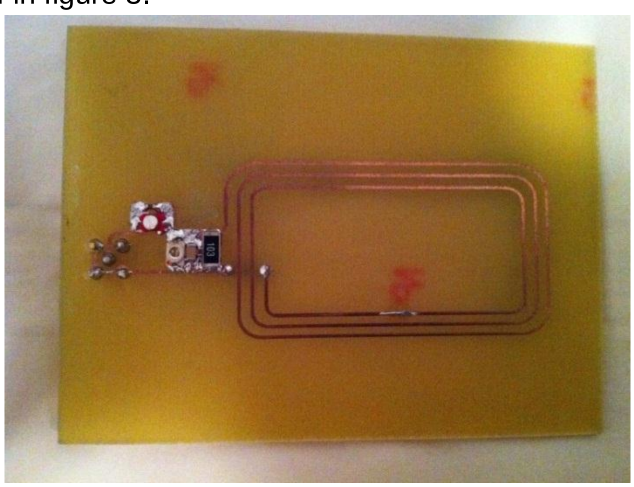

Figure 3: Complete antenna

In order to accurately tune the antenna to 13.56MHz, it was connected to a network analyzer for a one port S11 test, in order to measure the input reflection coefficient. Ideally, the result of the one port test would be to have the antenna tuned such that the minimum S11 value occurs at 13.56MHz. This would indicate that the maximum amount of the power of the transverse electromagnetic signal propagating to the antenna is being absorbed and hence radiated by the antenna at this frequency. By adjusting the variable capacitor present in the antenna circuit, we were able to tune the antenna to the point shown in figure 4 at marker 1, which represents the desired point of minimal S11.

The input impedance of the antenna was adjusted similarly by tuning the variable capacitor connected in series with the circuit. Figure 5 is a Smith chart of the input impedance, where marker 1 shows the impedance to be about 66Ω at 13.56MHz, which is not ideal, but is within the acceptable range.

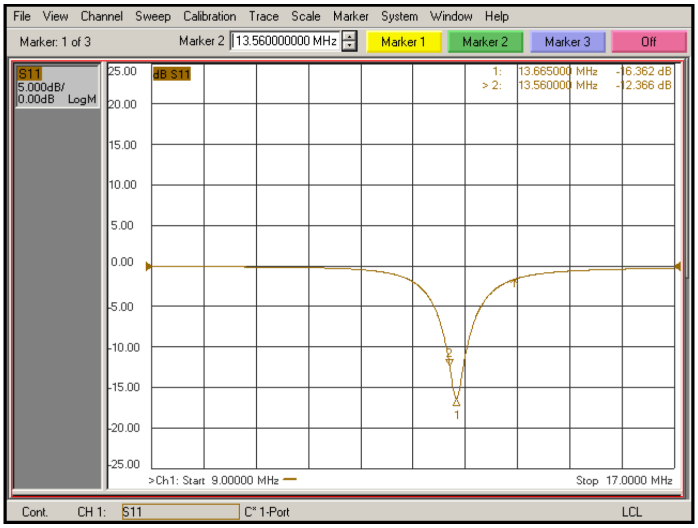

Figure 4: S11 result prior to tuning

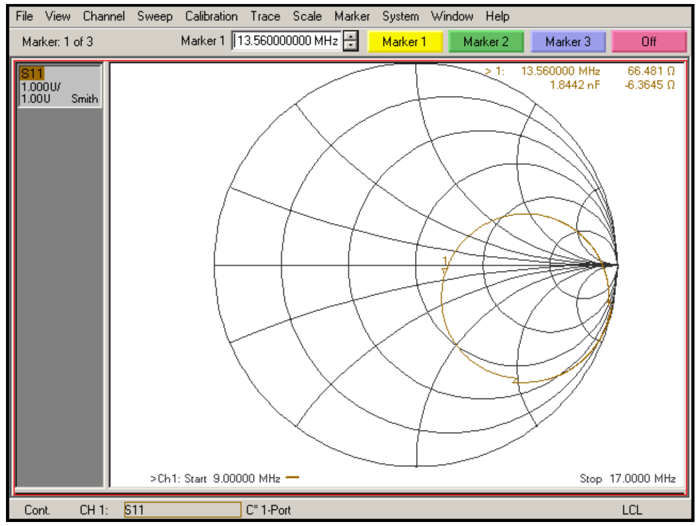

Figure 5: Antenna circuit Smith chart

Transceiver IC:

To simplify the circuitry involved for modulating and demodulating the data to and from the RFID tag, a multi-protocol transceiver IC from Texas Instruments was used for this project. This IC, the RI-R6C-001A, is a low power device designed for RFID systems operating in the 13.56MHz range. The chip interfaces with the antenna and a microcontroller to create a complete RFID reader/interrogator system. Figure 6 shows the TI recommended application schematic that was used for the construction of the reader. An Atmel ATMega328 with the Arduino bootloader installed was the microcontroller of choice for this project.

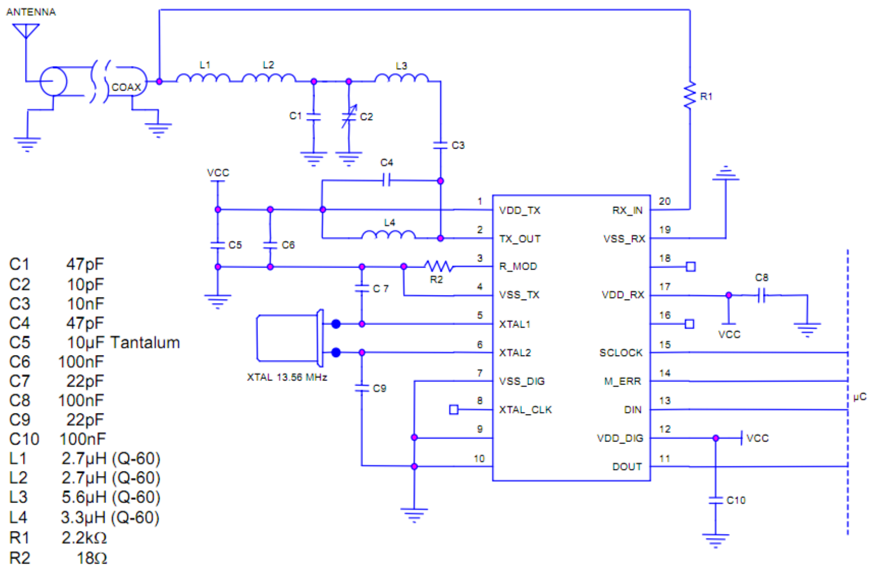

Figure 6: Recommended application schematic for RI-R6C-001A (Source: Texas Instruments)

The communication protocol used by the transceiver IC utilizes a simple three wire serial link between the transceiver and a microcontroller to transmit and set up data. There are 4 pins on the transceiver IC that interface with a microcontroller for communication, namely, SCLOCK, M_ERR, DIN, and DOUT. The SCLOCK pin is used as a bi-directional serial clock, while the DIN and DOUT pins are used as data input and output to the chip, respectively.

For each communication transaction, the microcontroller sets the necessary logic levels on the DIN and SCLOCK lines to send an appropriate sequence of commands to the IC and awaits a response from it. A typical command is structured as a start bit, a command byte, a variable-length string of binary data to be relayed to the RFID tag (the length of which depends on the message being sent), and a stop bit. The command byte sets the communication protocol used by the transceiver, setting parameters such as the modulation index (10% or 100%), pulse position modulation coding (1 out of 256 or 1 out of 4), etc.

Figure 7 shows a timing diagram that illustrates the relationship between the SCLOCK and DIN pin while a command is being transmitted to the IC. The start bit (S1) defines the start of a command sequence and is characterized by a low-to-high transition on the DIN line while SCLOCK is held high. The stop bit (ES1) defines the end of the command sequence and is a high-to-low transition on the DIN line while SCLOCK is held high. Each data bit between S1 and ES1 is latched in the IC on the rising edge of SCLOCK. The value of the data bit on DIN should remain constant for the duration of SLOCK being held high.

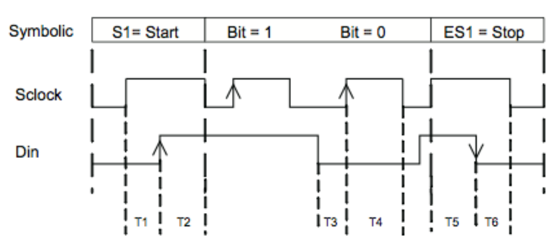

Figure 7: Timing diagram for the communication protocol (Source: Texas Instruments)

Once the entire command sequence has been sent, the transceiver IC modulates the data sent to the tag on a 13.56MHz carrier in accordance with the protocol defined in the ISO 15693 standard. The signal is transmitted through the attached external antenna. The tag responds appropriately to the request and transmits data back to the transceiver IC. The received signal is demodulated by a diode envelope detector and then processed inside the IC. The sequence of data from the tag's response is transmitted by the IC between a start and a stop bit, indicated by SCLOCK and DOUT. The transceiver IC clocks the data into the microcontroller using the SCLOCK and DOUT lines. Each data bit is latched at the rising edge of the clock. A timing diagram of the entire process of communication between the microcontroller and the transceiver IC is shown in figure 8.

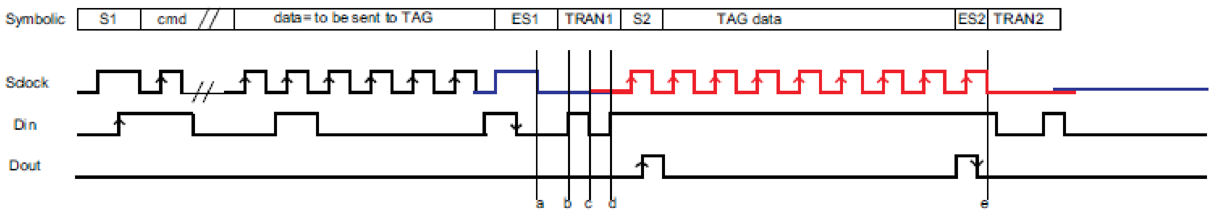

Figure 8: Full communication protocol (Source: Texas Instruments)

The microcontroller transmits data to the transceiver IC by changing the logic levels on the SCLOCK and DIN lines at the appropriate times, as described in the communication protocol. The microcontroller first sends the IC a command to initialize its transceiver. Figure 9 shows the voltage across the antenna once the transmitter is switched on. The output waveform is a very clean 13.56MHz sinusoidal wave that is used as a carrier to transmit data to the tag.

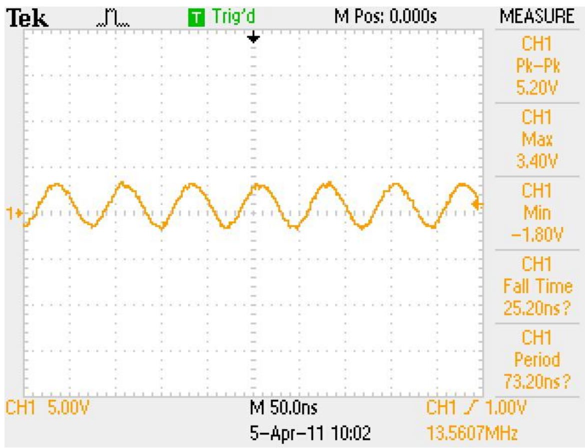

Figure 9: Voltage across the antenna

Once the transmitter has been initialized, the microcontroller sends the sequence of bits required to obtain the unique ID of the tag, as described in the ISO-15693 standard. Figure 10 shows the output signal from the microcontroller on the SCLOCK and DIN lines. The diagram has been labeled to show the appropriate sections of the sequence.

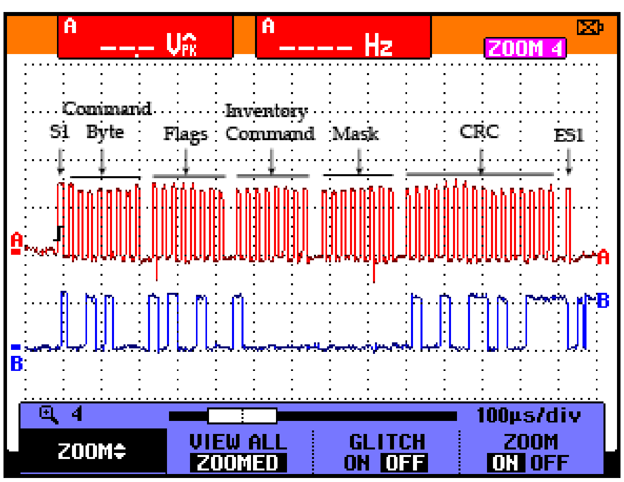

Figure 10: Command sequence sent to the transceiver IC

If a tag is present in the field of the antenna, the transceiver IC demodulates the response from the tag and transmits it to the microcontroller. The message sent by the transceiver IC contains a start bit, 96 bits of data from the tag (containing flags, the DSFID, and the UID as defined in ISO 15693-3), 2 bits indicating an end of file, and a stop bit. Since each data bit is latched onto the rising edge of SCLOCK, the microcontroller checks for interrupts on SCLOCK and reads the logic level on the DOUT line in the ISR for that interrupt.

Figure 11 shows the full response to the microcontroller of a sample tag that was used for testing. The red waveform is the DOUT line and the blue waveform is the SCLOCK line. A zoomed in version of the same signal, as shown in figure 12, demonstrates how the data bits are clocked into the microcontroller.

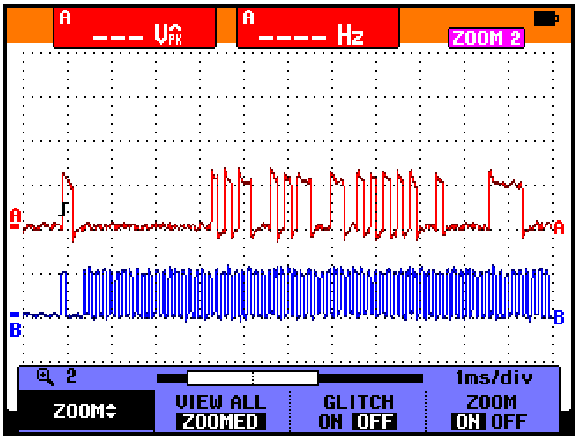

Figure 11: Response from RFID tag

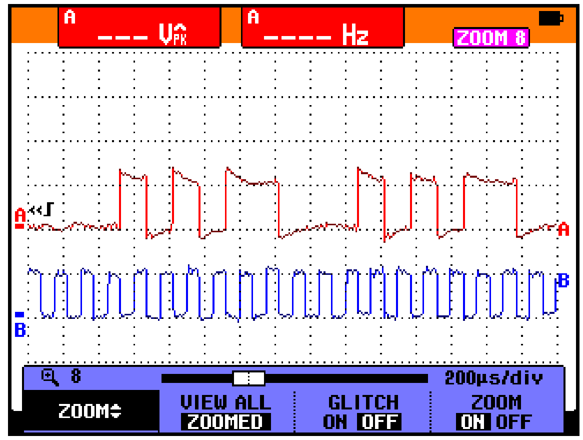

Figure 12: Response from RFID tag (zoomed)

The response from the transceiver IC is transmitted with the least significant bit first. Details of the format of this data string can be obtained from section 7.3 of the ISO-15693 standard. Functions for decoding the response received from the tag were programmed into the microcontroller, and the full 64 bit UID is stored in its memory as 8 separate bytes.

iOS device Interface:

A TRRS (tip-ring-ring-sleeve) headset connector was used for the physical layer of communication between the microcontroller and the iOS device. The TRRS connector, shown in figure 7.23, has 4 different connections, one each for the left speaker, right speaker, microphone in, and ground. The microphone line is used to transmit the UID that has been received by the microcontroller to the iOS device. The microphone line must present a particular impedance in order for the iOS device to detect that a microphone has been connected.

For the purpose of this project, the iOS application was programmed to accept a 32-bit FSK input from the microcontroller. The microcontroller uses a “bit banging” method for transmitting the UID serially through a digital output pin using FSK. For a 0 bit, the microcontroller toggles the logic level of the output pin at a rate of 1kHz for 1/32 of a second, and for a 1 bit the logic level of the output pin is toggled at a rate of 4kHz for 1/32 of a second. The least significant 32 bits of the UID are transmitted to the iOS device using this method. It therefore takes 1 second to transmit the entire string of data. Since the microcontroller uses TTL logic levels to output signals, the signal was attenuated to acceptable levels of the microphone input at around 50mV using a diode clipping circuit and an op-amp with an attenuation of 350. Figure 13 shows the plot and the Fourier transform of a sample FSK signal transmitted from the microcontroller in MATLAB. The large peaks visible in this Fourier transform occur at the desired frequencies of 1 kHz and 4 kHz, indicating that the signal generated by the microcontroller is as expected.

Figure 14 shows a photograph of the implemented hardware of the reader system.

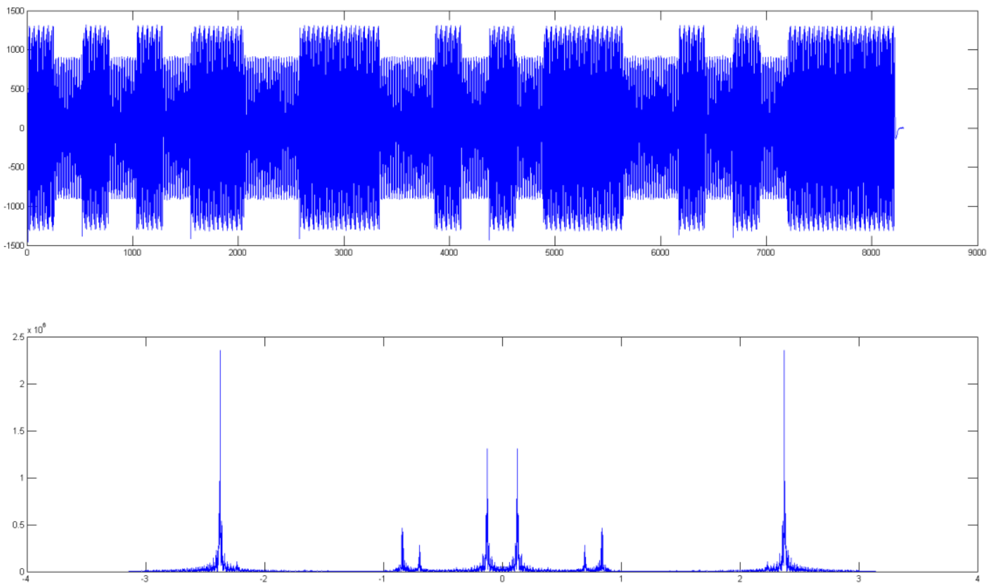

Figure 13: FSK signal sent to iOS device

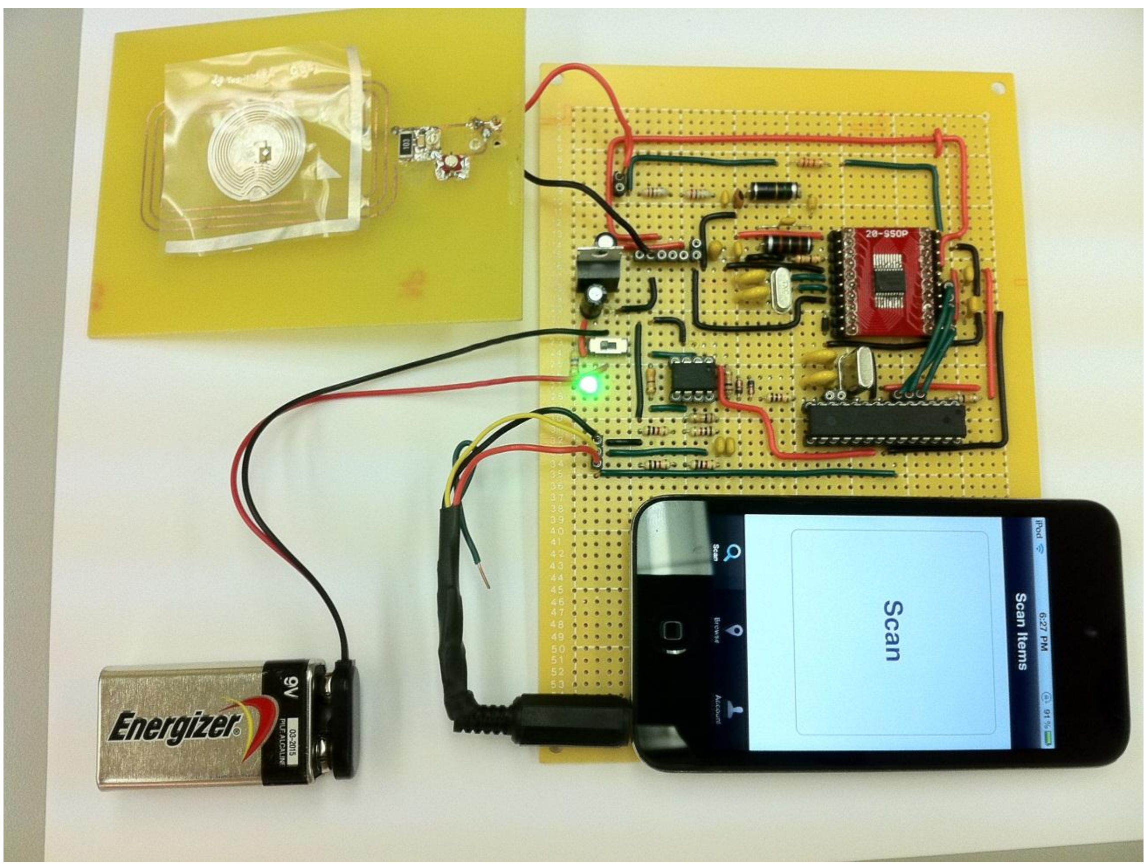

Figure 14: Complete reader system

Database and Website:

All user, department, and item data is stored in a single, centralized database. The database management system used for this project is MySQL. For security reasons, this database can be accessed only from the local server, and is available to sever-side application code through the MySQL PHP extension. Apache is used to host several PHP scripts that not only generate the content for the website, but also provide a secure API used by the iOS device to access the database.

The iOS database API is secured by requiring user authentication before any actions affecting the database can be performed. This authentication process, if successful, generates a unique authentication token for the user. Every request following this login will be validated by the API using this authentication token. After the API parses a valid request, it queries the database and returns the results to the iOS device using JavaScript Object Notation.

The website provides separate functionality for each level of user: administrator, analyst, or department custodian. Administrators are given access to various control panels which provide easy methods of managing departments, items, and users. Analysts are given access to report and notification generation systems, which together reduce the amount of paper required as compared to common inventory control systems. Finally, department custodians are only allowed to view their account information and the list of items in their assigned department. The administrator and analyst user classes are supersets of the department custodian class. An items listing for a test user is shown below in figure ~.

Figure 15: The items page offers a sortable table of all items tied to a user's department

The website is implemented in a modular fashion, with all requests first passing through the index file. All pages, with the exception of the login page, require authentication. Before any request is interpreted, the authentication is verified by checking the PHP $_SESSION array for an authentication variable. After it is verified that a user is authenticated, the URL is parsed and the file containing the HTML elements for the requested page is loaded. All variables that are passed into SQL queries are sanitized to prevent SQL injection attacks. To provide security in the event of a database compromise, all passwords are salted and hashed before being stored in the database.

iOS Application:

In order to provide a powerful and user-friendly interface to both the reader hardware and the inventory database, an iOS application was developed. This application serves as the primary user interface to the RFID tag reader hardware by receiving and interpreting the UID values read from tags by the hardware and passing them along to the server for processing. The UID data sent from the microcontroller is interpreted in the iOS application by recording the signal as a set of audio samples, dividing this list of samples into sections, one for each bit, and then performing a Fourier transform on each section in order to determine the dominant frequency. The application allows for the interpreted UID value to be used in three ways; to simply mark the item associated with a UID as scanned in the database, to associate a tag UID with an item that does not currently have one, or to create an entirely new item entry associated with the scanned UID in the database.

In addition to this tag reading functionality, the iOS application also serves as a general client to the database. Using the application, users can see listings of the buildings, rooms, and items which they are responsible for in the inventory database. They can also easily evaluate their progress through the current inventory cycle, thanks to the presence of counts of unscanned items in each building and room location on the associated listings screens for those locations. A sample building listing is shown below in figure 16. Users can also see details for an individual item, and request that the database mark an item as scanned without actually scanning its tag, if necessary. This screen is shown in figure 17.

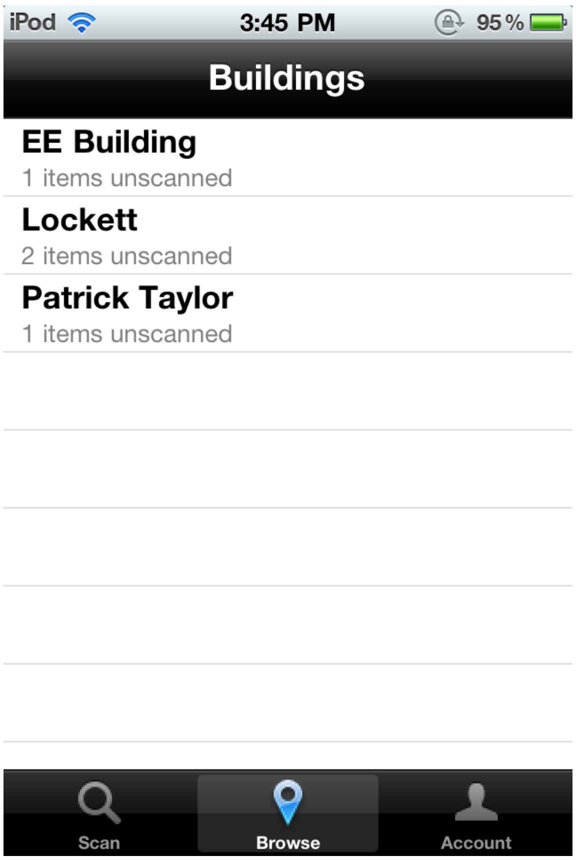

Figure 16: Sample items listing

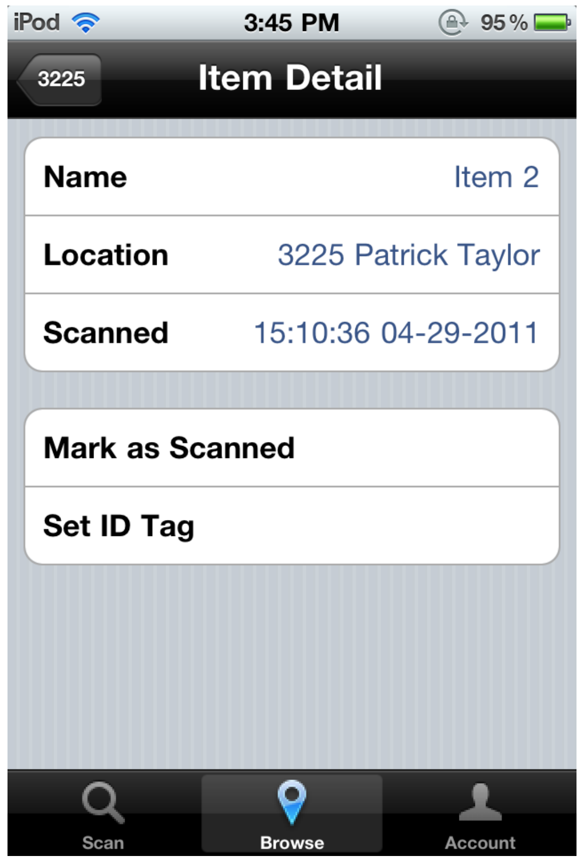

Figure 17: Sample item details

Conclusion:

In conclusion, this inventory tracking system has been designed with the goals of reducing waste and reducing effort as compared to the arduous processes which are currently common, involving printed notifications listing the complete inventory, aging databases to store and access items and department information, and tedious, manual entry of information into and retrieval of information from the database from an outdated user interface. It is believed that the paperless system featuring wireless tag scanning designed for this project meets these goals, while also being simple to use and incorporating a centralized database system featuring real-time data feeds. Industry standards contributed to streamlining the implementation of this project and also ensuring interoperability between our project and existing devices on the market throughout the world.

References:

[ISO15693-2]
ISO/IEC *stage* 15693-2: Identification cards -contact-less integrated circuit cards - vicinity cards – Part 2: Air interface and initialization, 2000-05-01, International Organization for Standardization, Geneva, Switzerland.

[ISO15693-3]
ISO/IEC *stage* 15693-3: Identification cards -contact-less integrated circuit cards - vicinity cards – Part 3: Anticollision and Transmission Protocol, 2001-04-01, International Organization for Standardization, Geneva, Switzerland.

“HF Antenna cookbook.” Texas Instruments.
<http://www.ti.com/rfid/docs/manuals/appNotes/HFAntennaCookbook.pdf>

“HF Reader System Series 6000” Texas Instruments. <http://focus.ti.com/lit/ug/scbu030/scbu030.pdf>