RFID Home or Office Automation System

Based on Localized Room Entry

By

Nolan Clark

Senior Project

ELECTRICAL ENGINEERING DEPARTMENT

California Polytechnic State University

San Luis Obispo

2011

Table of Contents

Table of Figures 3
Acknowledgements 4
Abstract 5
Introduction 6
Background 9
Requirements 11
Design 12
  Sirit Long Range RFID Reader 12
  Localized RFID Reader 16
  125 kHz Scale Model 23
Test Plans 24
Development and Construction 26
Integration and Test Results 31
Conclusion 33
Appendices 35
  Appendix A 35
  Appendix B 36
  Appendix C 37
Appendix D 43
Bibliography 47

Table of Figures

Figure 1: Example Home RFID Lighting Setup 8
Figure 2: Sirit Reader Hardware Design........ 14
Figure 3: Sirit Reader Program Flow........ 15
Figure 4: Hardware Design for localized RFID readers........ 16
Figure 5: Lighting Control Node........ 17
Figure 6: Melexis Hardware Layout........ 19
Figure 7: Localized RFID Reader Firmware Platform 20
Figure 8: Network Connections 21
Figure 9 125 kHz Model 23
Figure 10: Melexis RFID reader with Xbee Node........ 26
Figure 11: ISO Protocols Used for Testing 28
Figure 12: 120VAC Lighting Control Node 29
Figure 13: Logic Analyzer output of RFID Reader........ 32
Figure 14: Original Gantt Chart 35
Figure 15: Updated Gantt chart for localize readers 35
Figure 16: Sirit Reader Hardware Setup 36
Figure 17: Melexis Internal Hardware........ 36

Acknowledgements

I would like to extend my greatest appreciation to Professor John Oliver for all the assistance and opportunities he has provided me over the last year. Without his level of commitment and desire to watch his students succeed I wouldn't be where I am today. I would also like to thank my mother, father and brother for the four years of financial and emotional support; you really made my life a lot easier and more interesting. And of course I would like to thank the love of my life, for more than three years of support, and for providing me with the drive to exceed expectations and move forward as a confident individual, you helped me become who I am today, and I thank you each and every day.

Abstract

Design and build a proof of concept model demonstrating an application of Radio Frequency Identification (RFID) technology for use in a home or office lighting automation system. The system provides new means for reducing energy consumption through active light switching based on simplified location tracking. Often lighting system use motion sensors and timers to determine when a person enter or leaves a room, but these platforms are based purely on motion meaning “no motion, no light.” The RFID based system utilizes medium range RFID readers and tags located at doorways or narrow passage ways in the environment to provide a general location and direction an individual moves through the area. As an individual enters a room the lights adjust to their personal preference and the lights remain on only while the person is in the room. The system provides a higher level of functionality and control compared to the typical motion sensor most commonly used in controlled lighting environments. The RFID based system provides a very effective methodology for lighting control with only minor setbacks for real world application.

Introduction

People often forget to turn the lights off when they leave a room and they never enjoy walking into a dark room looking for a light switch, that’s just human nature. The RFID based lighting control system provides means to eliminate this problem without the use of cumbersome and irritating motion detectors. Each user in the environment receives a relatively inexpensive unique RFID tag to associate them with a room or area in the home or workplace. This not only allows for individual tracking, but also creates the ability to adjust lighting based on who is in the room. The design of the RFID based lighting control system consists of several small, shorter range RFID readers located in the entryways and narrow passages of a home. Many factors played into this decision, all of which are overviewed in the “Design” portion of this document. The system consists of RFID nodes, used to monitor RFID tags, and lighting nodes, used to remotely control lights in the environment.

The RFID node basically controls the RFID reader and monitors incoming data for either processing or sending data to the primary controller for processing. The system contains as many RFID nodes necessary to provide the level of control the user wishes for the application. The RFID nodes operate at 13.56 MHz and provide a range of 1-3 feet depending on the location of the reader and attenuation in the respected environment. The proof of concept scale model uses 125 kHz readers with 1 inch of range. The design for the lighting node provides enough control and power to toggle 120VAC light without noticeable delay from the system. The entire system communicates over an established in home network.

The network used to communicate with all the nodes in the environment can either be wired or wireless depending on the application size. A wired network communicates via serial data transmission from one node to the next. This network form provides the highest reliability but is most beneficial for environments under construction, allowing the wires to hind within the walls. The wireless network uses an Xbee, wireless microcontroller, to communicate with a microcontroller via serial communication and sends the data by wireless modulation to the primary node for final processing. The wireless node can span between 10-30 meters, depending on interference, which is sufficient for most home or office applications.

One of the most important aspects of the design involves RFID reader placement in the environment in order to provide sufficient tracking for proper lighting control. The readers are designed for a one to three foot range, which severely limits how they get deployed through a home. For a typical application in a home or office environment the readers must be relatively close to where the tags are on an individual. For this reason the readers are placed in naturally-narrowed paths in the environment. This includes placement in doorways and narrow hallways in order to provide the reader the best opportunity to read the tag, as indicated in Figure 1. The number of readers and lighting nodes in use is entirely dependent on how the user decides to set up the system. Figure 1 provides a general idea of how a typical home may setup the different nodes to accommodate the space. The primary goal of the system is to assist the end user or company in saving money by reducing energy consumption throughout the home by eliminating wasted energy.

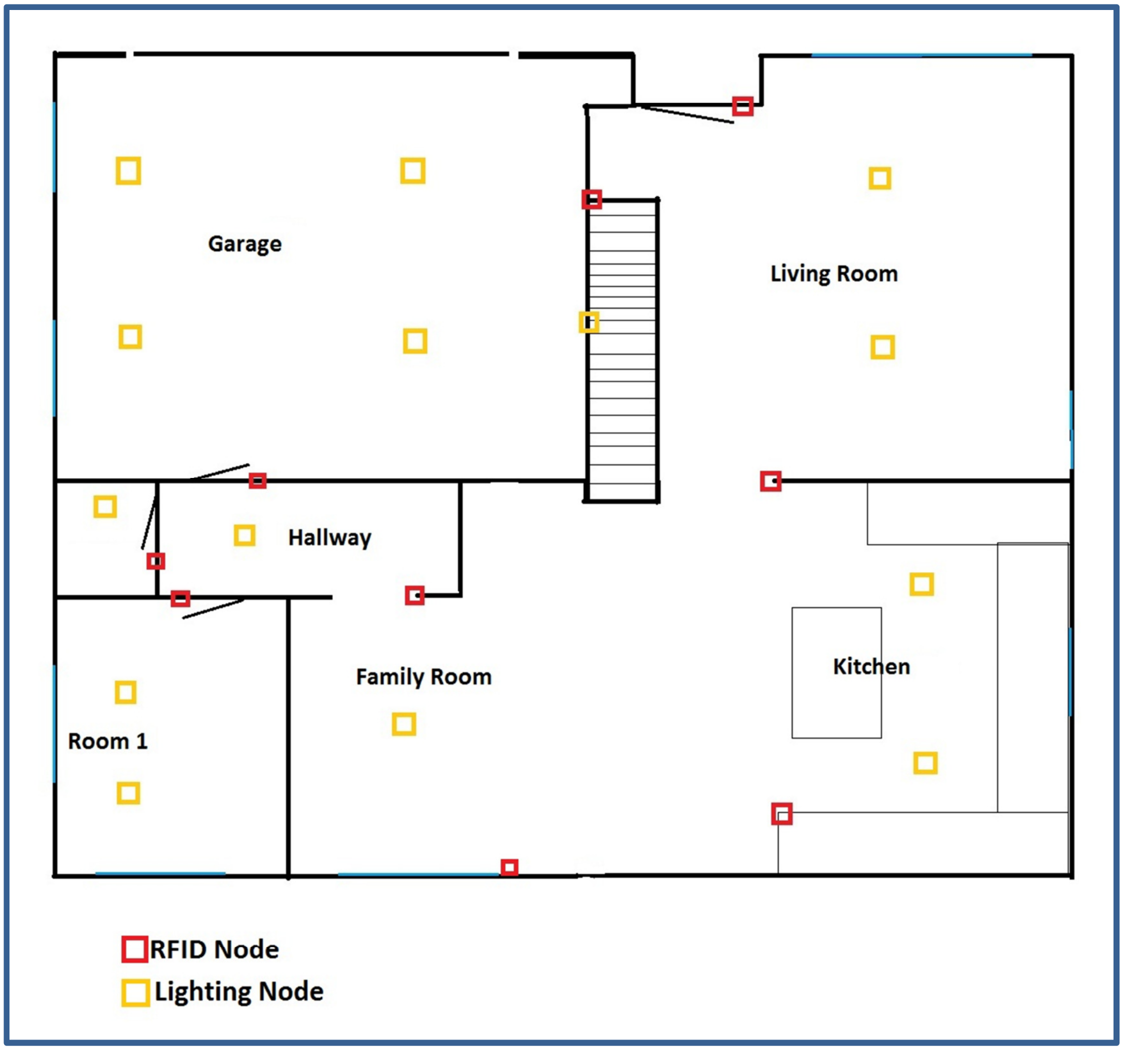

Figure 1: Example Home RFID Lighting Setup

The system initially under went several full design changes to create a solution that is both cost effective and potential marketable. All the design changes and modifications are fully detailed in the "Design" section. Appendix A provides detailed gantt charts and project development. The optimum solution provides a truly scalable low power design allowing it to accommodate a wide variety of applications.

Background

Versatile home automation systems have existed for nearly a decade and continue to work their way into homes across the country especially with the strong push for wireless devices and internet connectivity. The RFID based home automation provides a higher level on control not often seen in typical lighting automation systems.

RFID technology has come a long way over the last decade allowing strong personnel and inventory tracking for large scale corporations. Personnel RFID tags are most commonly used to grant access to a secure room or building and have an operating range of approximately 3-4 inches. A reader of this sort gets placed at every entrance to the area and a system logs the user identification number along with the time of entrance. The other most common application of RFID readers comes from warehouse inventory tracking. Often companies need to track inventory to determine either the location of the item or if it is in stock [5]. These readers often have extremely high power demands but have the ability to read tags from several hundred feet away without many issues. The inventory level readers often have the ability to read in several hundred RFID tags at a time making it a great system for a warehouse [6]. The first design for the RFID based home automation system uses a warehouse level RFID reader, a Sirit Infinity long-range RFID reader with an approximate cost of $2000 without the antennas or software. This type of RFID reader lack any level of versatility because each home requires a dedicated program due to attenuation between the walls making it difficult for determining location. For this reason the

design was changed to create a more versatile system. The reader used for the design falls in a category of its own between personnel and warehouse readers with an approximate range of 1-3 feet. The RFID reader is a 90121 Melexis RFID reader evaluation module operating at 13.56MHz and able to operate with a variety of ISO protocols [3]. The Melexis reader provides a real world transition from the small scale model operating at 125 kHz. All details pertaining to the development of the different RFID system is described in the “Design” section.

The entire network communicates by either a wired or wireless network. For the wired network each node connects to the next node creating a large network web to help stabilize data transmission. The wireless transmission uses Xbee wireless microcontrollers that operate at 2.4 GHz with a line of sight range of 30 meters [2]. The Xbee provides a simple interface allowing modifications and updates to operate relatively smoothly. Regardless of the network format, both communicate through a microcontroller to a USART bus for the network. The RFID and lighting nodes communicate over the network to provide rapid system response.

Requirements

The design must demonstrate the ability to create a RFID based home lighting automation system and demonstrate the advantages of such a system. This includes creating a scale model of the system to demonstrate how the system operates and prove the scalability of the design for a real world application. The proof of concept design contains two RFID readers and a lighting system to demonstrate a communication network. The design needs to prove scalability and potential for marketability in a wide range of applications. At least one light node must be completed to prove how high voltage switching operates in the system.

Design

The RFID based home lighting control system went through several major design changes as issues arose with the different readers rendering them impractical. The first design uses a long range RFID reader, but problems with attenuation and cost led to using a cheaper short range Melexis RFID reader. Full detail for each design is located in the subsequent sections. The scale prototype uses 125 kHz RFID reader, which better accommodates understanding of how the system operates.

Sirit Long Range RFID Reader

The original design uses a Sirit Infinity 510 Long Range RFID reader most commonly used for inventory tracking in a warehouse. The system can read in hundreds of tags at a time and utilizes signal attenuation and modulation technique allowing a computer program to run triangulation algorithms to determine the approximate location of the tag. The device contains 4 antenna connections, an Ethernet (RJ-45) and RS-232 for communication, see Appendix B[6]. In order for the lighting system to operate correctly the design calls for using one to two Sirit readers in the environment with 3-4 antennas each to provide some level of in home triangulation. Several major issues arose with this reader preventing it from moving forward with the implementation process.

One major issue turned out to be overall cost of the system, at 2000$ per reader without any antennas or software, development and implementation cost would be far too high for marketability. Another major issue comes from how the reader interacts with the environment. The reader is great for large open spaces, but

when walls, furniture, and other typical home amenities come into effect, attenuation between the reader and the tags creates an extremely complicated system. Creating a triangulation algorithm to compensate for each of these loses is very time consuming and must be custom for each home. The algorithm requires modification or the tag will never be where the reader/ program think it is in the environment. There was one) final issue that ended up costing huge amounts of time and effort in order to get the RFID reader to read a single tag and store that information. The Sirit reader was provided by the Cal Poly RFID lab, but during pick up was not verified for proper operation. Initially the reader appeared to be fully operational, but after several days of attempting to connect to the device using the "RealTerm" terminal, along with the provided software, the reader only blinked a light during start-up then appeared to turn off. The reader was returned to the lab and after several weeks retested. The reader is operational with the provided software, but still had issues connecting using "RealTerm." The combination of all these problems instigated a change in the final design to use several low power readers, although the long range system did have a full design completed.

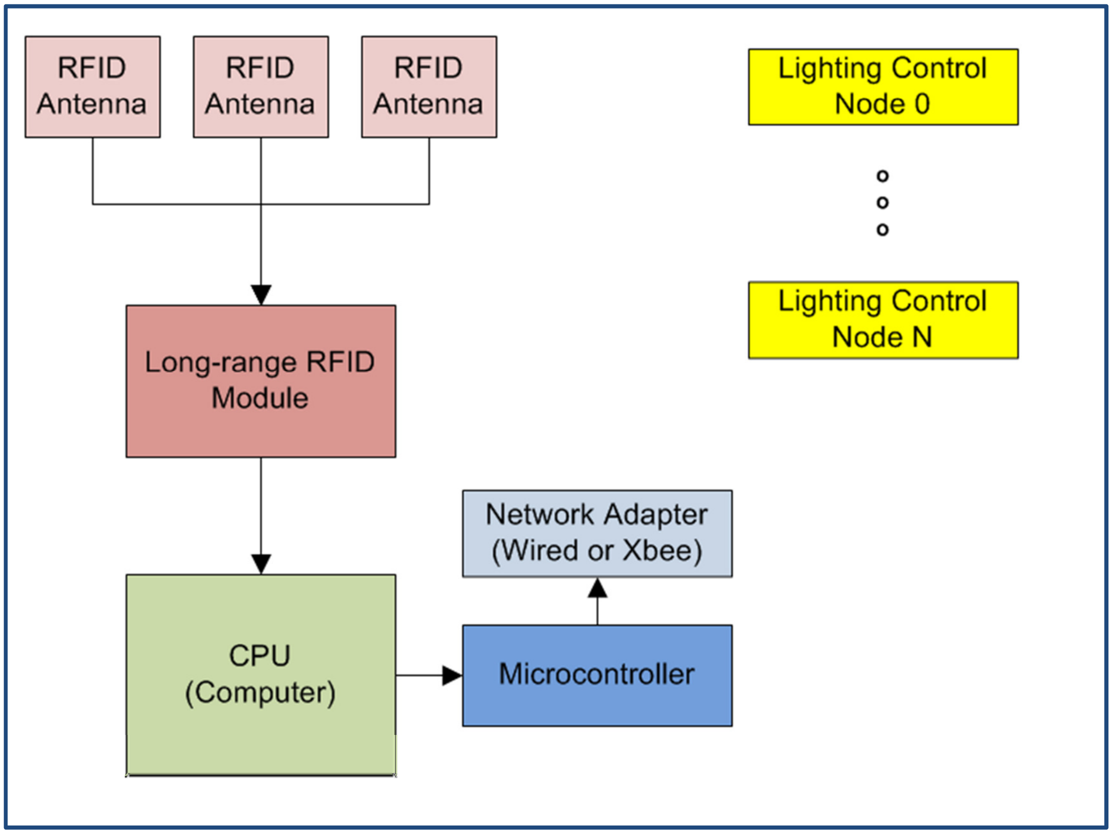

Figure 2: Sirit Reader Hardware Design

For the hardware portion of the design the system uses three RFID antennas connected to the reader, as see in figure 2. The reader communicates with a computer via an Ethernet connection and the computer performs most of the processing for output to the lighting system. The reader modulates through each antenna to prevent crosstalk, and gather both the tag data as well as signal strength. This data is sent back to the computer for processing. The computer runs the triangulation algorithm that, if properly calibrated, returns the location of an individual in the environment. The computer keeps track of each individual RFID tag and turns on the lights based on where they are in the home by sending the lighting data out from the network node to the lighting node, as see in figure 3. This system, although it works in theory, will not

operate well in real life because of the attenuation issues. If the layout of the home changes, or even someone walks by the reader, this may offset all the tags and cause the lights to flicker or completely change states. For this reason the design changed to using shorter range, more localized RFID readers.

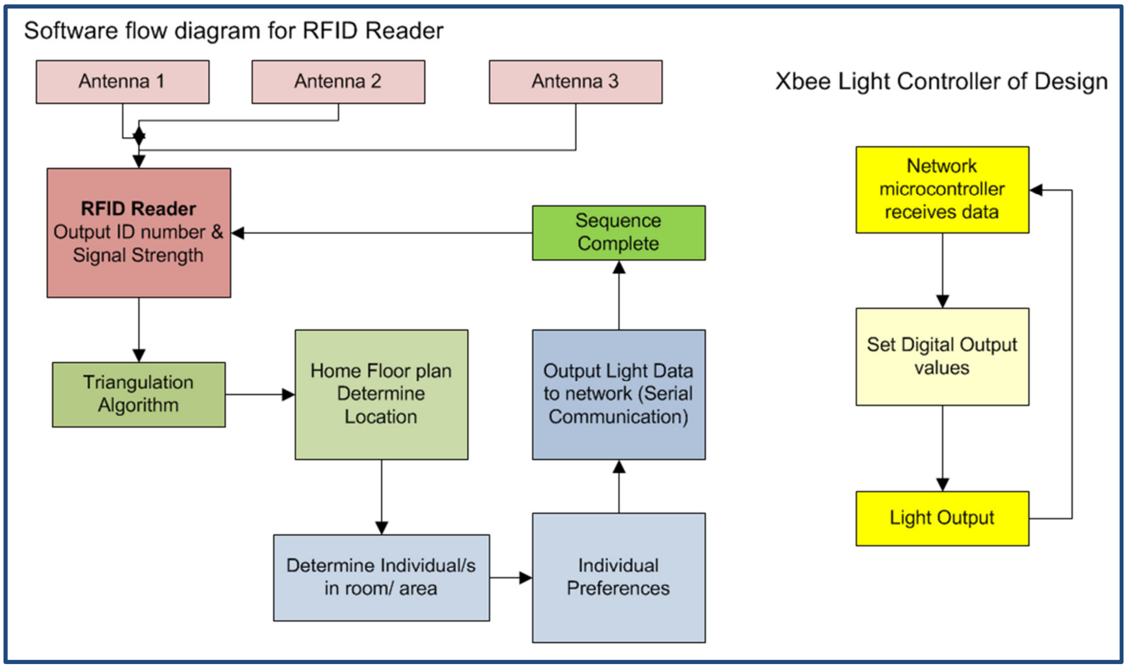

Figure 3: Sirit Reader Program Flow

Localized RFID Reader

The Melexis reader provides a cheap and versatile approach to create an automated home lighting system. The system uses multiple low power RFID readers with approximately 1-3ft range operating at 13.56 MHz. The lighting nodes use a microcontroller and some form of network connection, this can either be wired or the Xbee wireless microcontroller. A full system hardware design is in figure 4. The system is capable of using as few as 1-2 RFID nodes and up to several dozen lighting nodes, allowing the system to fit to the application. Although the system detailed here is dedicated to the Melexis RFID reader, the hardware layout and programming actually function with a variety of RFID readers with minor hardware and software changes.

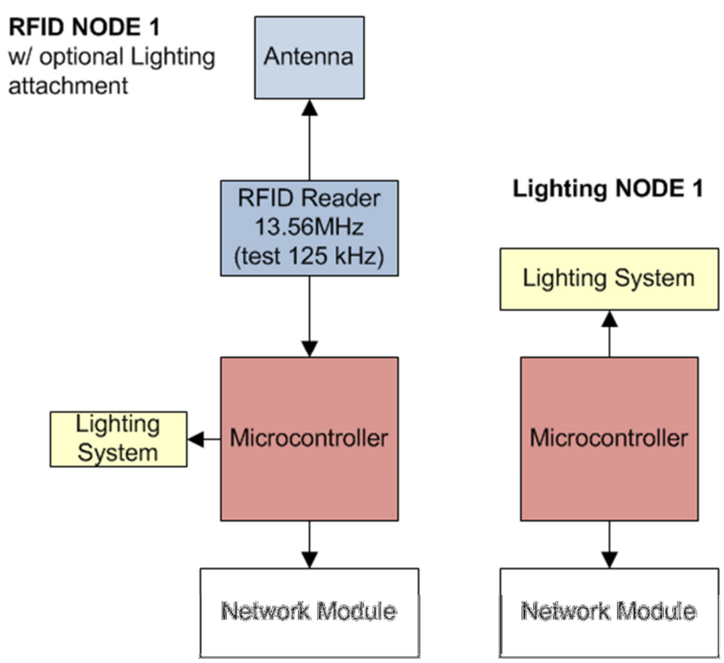

Figure 4: Hardware Design for localized RFID readers

The hardware portion of the design includes two components, the RFID nodes and the lighting nodes. Lighting nodes control the high voltage lighting in the environment using solid state relays (SSR), a microcontroller, and the Xbee wireless communicator. The lighting node program remains simple, compared to the complexity of the RFID node, which allows for quick design changes to meet the demands of the application. The node is fully adaptable for connection via wired or wireless network and provides complete control over the light attached to the node. Nodes for lighting control consist of solid state relays capable of drive 120V 8A loads from the 3.3V switching voltage of the microcontroller, as seen in figure 5. Power input for the nodes comes from a 120VAC input and transmits power out through a common United States AC receptacle. A 3.3V input control line toggles the relay to turn the light on and off. Under the current design the lighting control node and the high voltage switching use separate power sources, but with more development the lighting control node can connect to the 120VAC sources with an AC-DC converter, allowing for further integration.

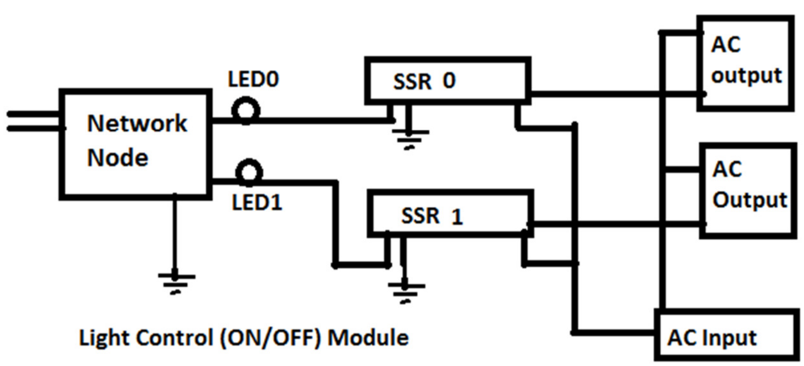

Figure 5: Lighting Control Node

A single node serves as the primary controller for the entire system and contains a RFID reader. The primary system controller consists of an ATMEGA2560 microcontroller, the RFID reader (Melexis for the full scale design), and a network connection either wired or wireless. A Melexis RFID reader operating at 13.56 MHz with 200mW of antenna power serves as the identification input device for the system. The high frequency system allows for longer range antennas and tags, which provides this reader with a reading distance of 1-3 feet. The antenna power output of the board is around 250mW, but the antenna provided with the development kit use approximately 200mW [3]. The Melexis RFID Evaluation Board uses a Melexis 90121 SoC with the capabilities of providing raw digital antenna data bits, or processed data. The reader is capable of using ISO 15693, and ISO 14443B RFID tags operating between 10-100% modulations. Digital output of the system can be AM (direct), FM (direct), FSK (Dual subcarrier), or PSK for ISO14443B [4]. The reader communicates with the microcontroller via serial peripheral interface (SPI) acting as the slave device, as seen in figure 6. A microcontroller acts as the master controlling the input clock frequency and sending the reader the initialization code to set the protocol and turn on the device. The reader also has two control lines to determine the mode of the devices.

An ATMEGA2560 microcontroller on the STK-600 development platform communicates with all the other readers lighting nodes. The microcontroller contains 256KB of memory, which is more than sufficient space for the program. The ATMEGA does support a variety of interrupts for the SPI and serial communication

as well as timers to allow solid program operation. ATMEGA2560 supports up to 86 digital I/O pins and the design (including testing ports) uses a total of 6 ports.

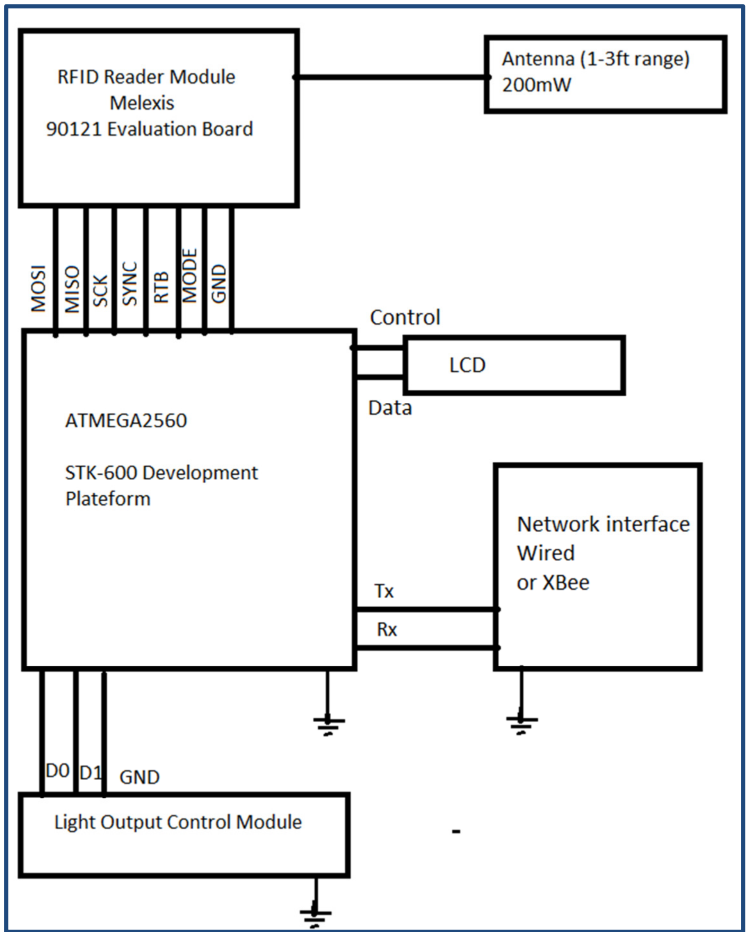

Figure 6: Melexis Hardware Layout

Testing and prototyping for the system uses a LCD display to visually show the ID number of the user to determine if the system is operating as expected. The primary module also uses LEDs for visual verification of data transmission and reception.

In a real world setting the system uses an RFID reader at all doorways in the home and a lighting node connected to many lights throughout the home. The system is designed to register users in a room or area in the environment as the reader

evaluates the id number. The lighting adjusts according to who is in the room and in the case of multiple users the system automatically adjusts to a default setting. The primary controller in the system mitigates all the data to output the lighting data as quickly as possible to prevent delay when switching a light on. A single primary controller helps make future development and production more affordable to the end user. The nodes remain small enough to reduce visual impact in the environment, making the system as streamline as possible while still reducing energy consumption.

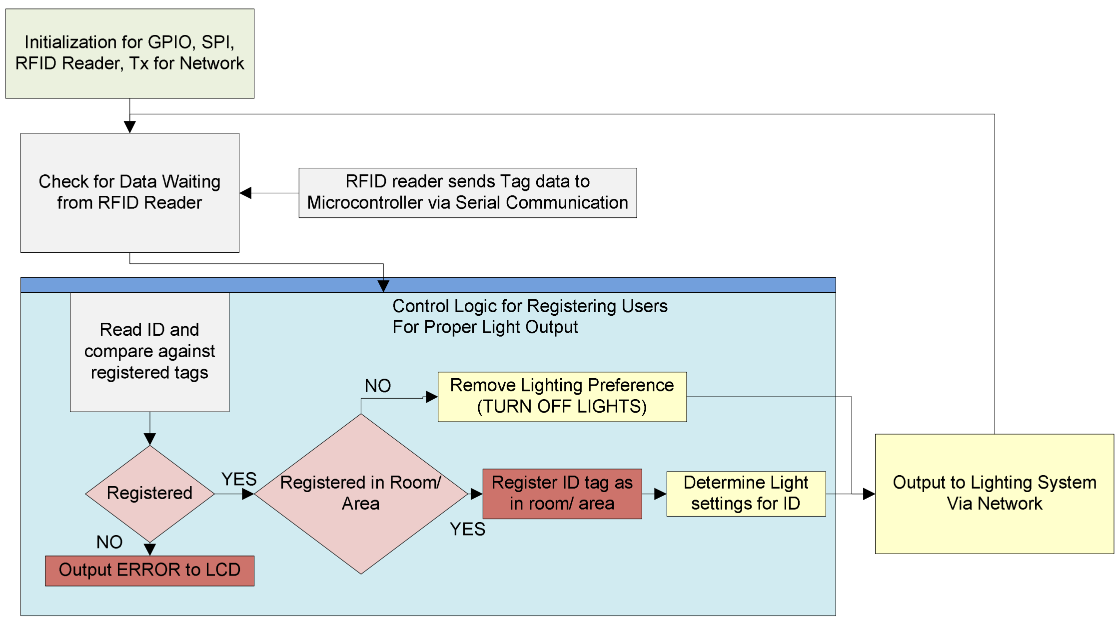

Figure 7: Localized RFID Reader Firmware Platform

The firmware for the program is written in C for the Atmel ATMEGA2560 microcontroller. The main program runs in a continuous loop cycle, preventing the program from ever exiting and disabling the system and contains two RFID identification numbers hard programmed into the system along with their particular lighting selection. An important feature for the RFID lighting system firm is

portability, which indicates the firmware can operate effectively under a variety of different implementations. This includes operating with both the Melexis RFID reader as well as the small scale Parallax readers.

The program uses a series of interrupts to determine if there is data waiting to input into the system and to properly output lighting data. The primary reader waits for data generated from an interrupt, collects the identification number which is clocked in using either UART or SPI. Once the system reads an ID number the next step involves registering and reregistering the system from the room/ area in the environment, as shown in figure 7. Once the ID is registered in a room the system checks for other users in the area and outputs the lighting data accordingly. It is necessary to check for other users in order to output a default setting if there are multiple people in the room. Figure 7 provides a visual explanation of how the system operates. The final step in the process involves output the light data onto the network where the lighting modules will output the correct light setting.

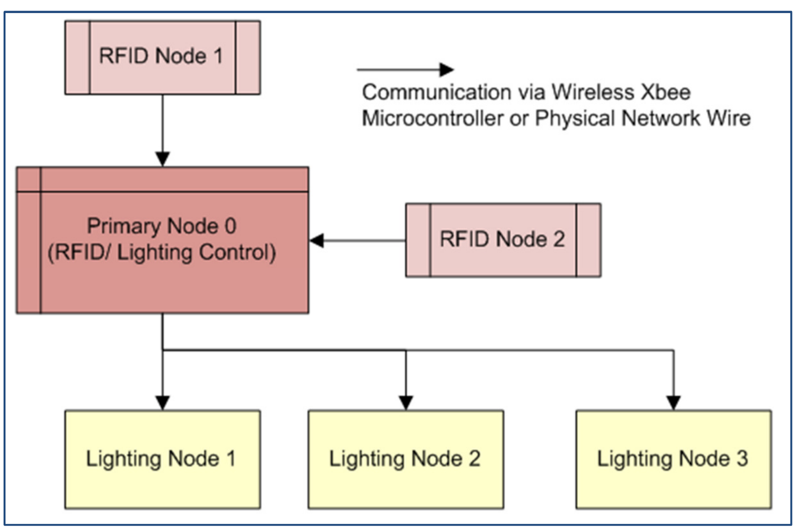

Figure 8: Network Connections

The communication network consists of a wireless microcontroller, Xbee, or a physical wire connection depending on the application. The main controller computes the logic for the lighting system and registers users in a room or area. The system can contain several additional RFID nodes that run on individual platforms and transmit the data back to the primary node. Using a centralized processor helps reduce the cost and simplifies the design as a whole, especially when programming lighting logic. The secondary RFID nodes only have the capability of sending data to the primary node, whereas the lighting nodes can only receive transmitted data from the primary. The primary node sends and receives data from/to the secondary nodes. Using unidirectional data transmission on the nodes prevents erroneous data from interrupting the lighting output causing the lights to malfunction. The system presented here opens the opportunity for reducing the energy consumption of the nodes, by allowing them to sleep when a user is not in close proximity to a reader. The network sampling rate can decrease and the RFID reader can go into standby mode reducing the overall energy consumption of the system.

125 kHz Scale Model

The 125 kHz scale model provides an in depth example of how the system operates in a model home. Minor issues with the 13.56MHz reader did hamper development but it does operate with the code provided in Appendix C. More detail is provided in the “Testing” portion of the report. The model uses 2 125 kHz RFID reader operating using the firmware flow diagram provided in figure 7. The home model contains 2 doorways that people (RFID cards) enter through, as seen in figure 9. The system contains two completely independent microcontrollers to simulate a real-world implementation. The primary node houses the logic and control for the system and the auxiliary RFID node provides communication to the reader. The two independent systems communicate over a UART network. How the system operates matches directly with a full scale system, the only modifications are for the initialization code to connect to the RFID reader. The model demonstrates the true versatility of the system.

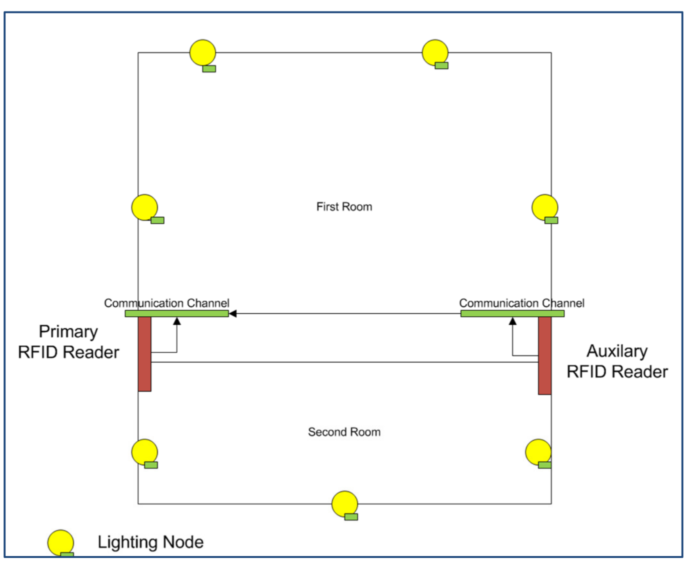

Figure 9 125 kHz Model

Test Plans

The test plan for the RFID system involves both reader functionality testing and system level testing to ensure the complete system operates as expected. The first test involves testing if the Melexis RFID reader operates as expected and is fully capable of outputting understandable consistent data. Melexis evaluation readers come with test points for the internal clock and the antenna allowing the use of an oscilloscope to test for functionality. For proper operation of the RFID is set to use majority voting to filter noise, compensate for jitter, and correct distortion. The C code for testing the RFID reader implements a very basic test function with wait loops and LEDs to ensure the code executes as predicted and does not lock up. One of the most challenging aspects about using the 13.56MHz RFID reader comes from different ISO, or communication protocols, for the 13.56MHz reader tags. The reader in this experiment is capable of reading 3 different protocols including, ISO 15693 100% mod, FSK ISO 15693 10% modulation, and B Type ISO14443; the ISO protocols are not interchangeable which can make reading the tags challenging if the protocol for the tag is unknown [4]. Since the RFID tags are not labeled with their ISO protocol the only way to determine if they work with the RFID reader is to test each one individually. To determine if the ID tags works an oscilloscope is attached to both the antenna and digital output test points. The output should contain a waveform representing the id number, if not, the tag did not operate correctly.

The communication network also requires testing to verify data is transmitted as expected. The primary tool to test network communication is a logic analyzer. A

special program, X-CTU, tests the Xbee, wireless microcontrollers for functionality [8]. The software allows easy setup of each wireless node to act as either an end node or a router. To test transmission and reception the nodes are set up using the software allowing the on-board LEDs to blink as the system transmits and receives data. The communication channel must also be check to ensure the data going into the primary controller contains both an id number and a location. The most important test verifies the logic of the main program operates as expected. The scale model provides an excellent example of how the entire system operates in a real world setting.

Development and Construction

Initial development for the RFID based lighting control system uses a high power Sirit RFID reader. The first reader received from the Cal Poly RFID lab did not communicate or power up correctly after several weeks of attempting to connect to the reader. The reader was returned to the lab and retested and, after approximately 1 week, the reader was tested and did communicate with the computer. As development for the project moved forward issues began to arise relating to the real functionality of using such a high power reader. The primary issue with the reader had to do with recreating the entire program every time it is implemented in a new environment. Due to a lack of attenuation consistency between the walls, the system would not be consistently reliable, and could inadvertently change the lights to the wrong room. This is where the localized RFID reader seemed far more practical for a typical home or office setting. Several days of searching resulted in the Melexis 13.56MHz RFID reader with approximately 1-3 feet of range depending on the tags.

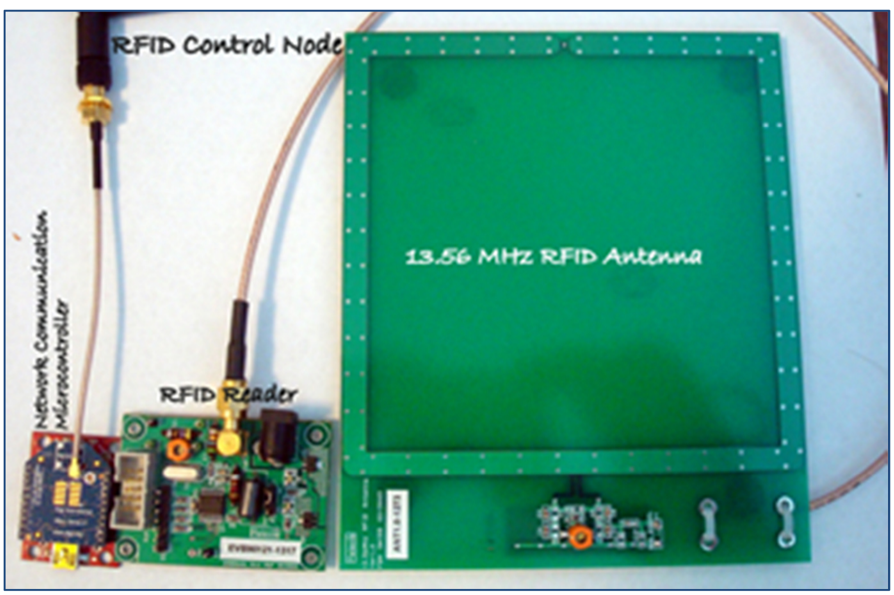

Figure 10: Melexis RFID reader with Xbee Node

Connecting the 13.56 MHz reader into the system uses an ATMEGA2560 microcontroller attached to the STK-600 through the on board SPI bus. The RFID reader data is either processed on the microcontroller or sent out via the network connection, depending on the type of node. The reader is connected to the microcontroller via the SPI bus along with 2 control lines for programming type (setup or data) and antenna (reception or transmission). Figure 11 provides an idea of how the registers were set up to test the ISO protocols for the reader because the actual tag type was unknown. The high frequency reader uses a hardware reception/ transmission encoding and majority voting to control all 13 control registers set up for ISO ASK 15693. The final test uses ISO 15693 FSK because of the tags type and the slightly long range of the reader. Using the 13.56 MHz reader created several major issues in the design because of actual problems with the reader communicating with the microcontroller. The first issue came directly from the RFID reader's indicator lights, which were extremely dim when the system was on. The two red indicator lights never changed brightness and made it appear as though the reader was not working correctly. After several days of troubleshooting the reader did appear to operate correctly, but the lights did remain very dim. The next major issue was the lack of ability to get the right RFID tags for the system. The RFID tags are not clearly labeled with their ISO protocol requiring constant testing and modification to determine if the system was operating correctly. The final confirmation that the system operates correctly came from modulation on the antenna after the program initialized on the microcontroller. The antenna was modulating at the same frequency

as the internal clock providing confirmation the system was operating correctly. Lack of time and resources prevent full system implementation on the larger scale, but the small scale does provide an excellent example of how the main program operates.

	Norm	ISO15693		ISO14443
Address	Register	ASK High Baud Rate 100% modulation	FSK High Baud Rate 10% modulation	B Type
0	AnalogConfig	47	63	73
1	PowerState*	01	01	01
2	Reserved	00	00	00
3	DigitalConfig	09	0D	0F
4	EncoderSym0	BF	BF	00
5	EncoderSym1	EF	EF	FF
6	EncoderSym2	FB	FB	00
7	EncoderSym3	FE	FE	00
8	EncoderSym4	7B	7B	00
9	EncoderSym5	DF	DF	00
10	EncoderTimeRef	1F	1F	03
11	DecoderTimeRef	7F	7E	3F
12	LTC	3F	00	00

Figure 11: ISO Protocols Used for Testing

The program went through a build and test cycle during development to ensure the system operates within parameters. To create the program the first step involves initializing the RFID reader, depending on the RFID reader this is done with either setting up the SPI bus for transmission and reception or setting up the microcontroller for reception of 2400baud UART data transfer. Once the reader is properly initialized, data reception and IDs are tested using a known tag id along with a logic analyzer and LCD screen. If everything operates correctly the associated ID is displayed on the screen without errors. For the scale model this took several attempts because the functions often received erroneous data and stored that data as actual id information. The next major step involved adding each id number to the microcontroller and developing a methodology to compare it against other saved id numbers. In order to make the comparison only a portion of the id is compared, 2 specific digits, to save processing clock cycles. If the id matches any of the registered tags a flag is set in order to indicate the presence of a person in a room. Once that

function appears to be fully operational the next step involves adding an additional RFID reader to the system. This involves setting up the primary device to receive RFID id data from either the established network or from the wireless Xbee. The additional microcontroller reads in the RFID data using the same code developed for the first reader, the only difference is that the second reader does not do any processing; it is only designed to send the data out. The primary reader processes the external RFID data via an external interrupt. The light data only gets output after processing, based on the tags in room flags. Full detail of how the code operates is in the code located in appendix E. The program is very versatile allowing different RFID readers to be added to the system with only minor code modifications.

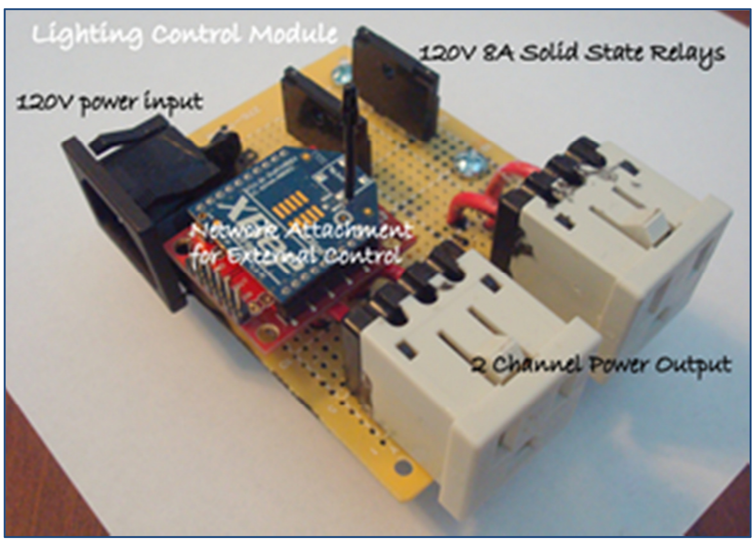

Figure 12: 120VAC Lighting Control Node

The second major component for the system is the lighting node which provides high voltage lighting control for in home use. When designing the high voltage controller special care is taken to ensure the high and low voltages systems are properly separated. The system also uses several grounding points to provide an

extra level of protection. The lighting node to the right contains 2 channels for lighting control. For high voltage switching the system uses 2 SHARP solid state relays rated for 120VAC with a maximum load of 8A allowing for high power demanding applications to connect to the system. The light ON/OFF logic connects to a microcontroller (not shown). The data can either come from the Xbee wireless microcontroller or from a wired connection depending on the application.

In order to program the wireless Xbee microcontroller the module must first connect to the computer via a USB connection and be programmed for the desired set up. The primary control unit in the system is set up as a coordinator in the network and the other nodes are set up as end nodes. The lighting nodes are set to output data to the digital IO pins on the device based on the received data. The RFID reader nodes are set to send data to the primary node based on information provided by the microcontroller. To send data, the microcontroller outputs the control codes for transmission and the system automatically sends the data. The Xbee modules are very straight forward to work with and there is sufficient online data if there is any assistance needed for setup.

The final proof of concept design uses a scale model of a home and two 125 kHz RFID readers connected to two independent microcontrollers. The primary controller outputs the light data for the rooms and is set according to the tag in the room. In the event multiple cards are present in a room the system automatically defaults to a standard light setting. The small scale model provides a great example of the versatility and portability of the system.

Integration and Test Results

The small scale system provides an excellent example of how the system operates in the real world. It uses a 125 kHz RFID reader arranged in the same fashion the larger scale system would be arranged in. The primary reason for using a small scale system comes from issues arising from the 13.56MHz reader communicating with the RFID tags. The main issue came from different ISO protocols for the system and the lack of ability to really test functionality without knowing if the tags were correct or if the tags even had ID numbers on them because many IDs require programming. After continued testing the antenna on the Melexis RFID reader does modulate as expected and will read in RFID data. The verification came from the antenna output and clock output on the oscilloscope, providing a communication path between the reader and the microcontroller, see appendix D for C code.

The small scale system operates very cleanly although it still requires the tags to be placed in very close proximity to the reader even though they are placed in room doorways. The system does lock up on occasion because of false data reception causing the system to wait for data that will not show up until the following card is read.

Overall the small scale design demonstrates how the system operates very well by changing the room lighting according to the id number registered in the room. The small scale system makes it very easy to quickly demonstrate how the larger scale system operates. This also demonstrates how truly scalable the system can be,

operating on a dollhouse level all the way up to real world implementation. Each node is fully configurable for either an RFID reader with or without lighting, and the lighting can be up to 8 channels.

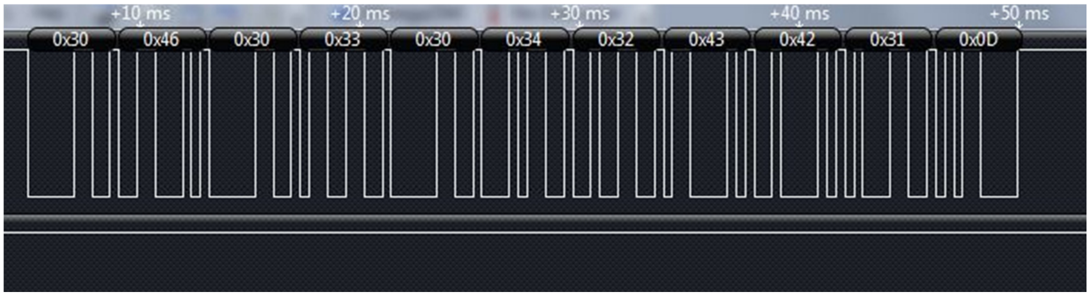

Figure 13: Logic Analyzer output of RFID Reader

Figure 13 shows the output of the 125 kHz RFID reader under proper operating conditions. Power usage for the readers was relatively high, approximately 130mA, which initially made it challenging to find a 5V power supply to power both the readers. Once the microcontroller is properly inputting data, the logic for the system is programmed to accommodate several RFID tags. The small scale model operates just as expected and provides a gateway for large scale development. Overall the design provides a truly scalable approach making it marketable for any size home.

Conclusion

The RFID based lighting control system design proves how RFID technology is effective for home lighting control in order to reduce home energy consumption. The proof of concept design demonstrates a versatile system capable of scalable implementations. The design can be anywhere from a single node to several dozen with minor programming changes creating a mobile marketable product. The design does have several limitation such as lighting control during the day verses night as well as the issue that the users of the system must always have an RFID tags on them for the system to work properly. With further advancements in the design as well as in RFID technology this issues become minor compared to the advantages of using such a system. The RFID home automation system provides a new level of control to home lighting, reducing energy consumption and increasing human comfort.

The system presented in this paper is far from production but does provide a gateway into full scale development. Future designs will include some form of time monitor in order to save money and not have the lights on during the day. The system will also include a computer connection to allow for tacking and easier registering for the RFID tags and lighting conditions. Their also needs to be some form of switch control to override the system when necessary. One large step forward is using a longer range RFID reader. The Melexis does operate well under limited conditions, but an RFID reader with 3-5 foot range would operate much more effectively for the design. A RFID based lighting control system has potential to make it into the home and office environment especially as RFID technology improves. As time moves on,

automated systems will continue to work their way into the lives of people, reducing energy consumption and increasing our personal comfort.

Appendices

Appendix A

The Gantt charts reviewing project progression throughout the development cycle.

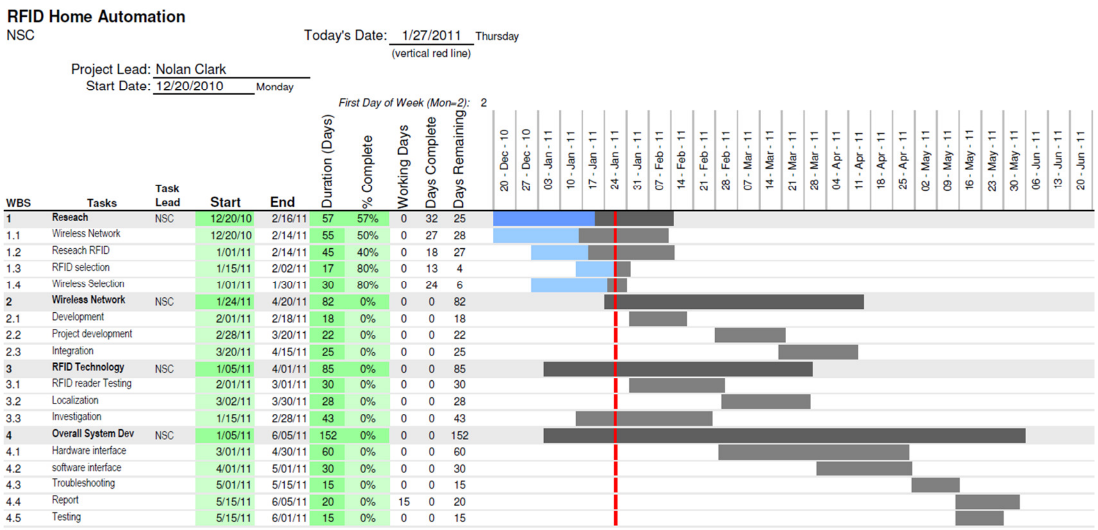

Figure 14: Original Gantt Chart

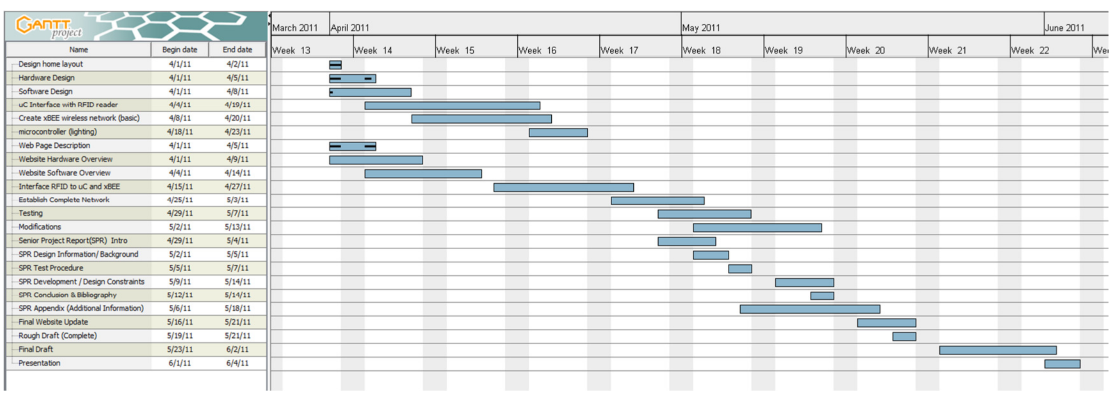

Figure 15: Updated Gantt chart for localize readers

Appendix B

Sirit reader hardware set up used for testing. The system communicated with the computer using the Ethernet connection and a single antenna for reading the tags.

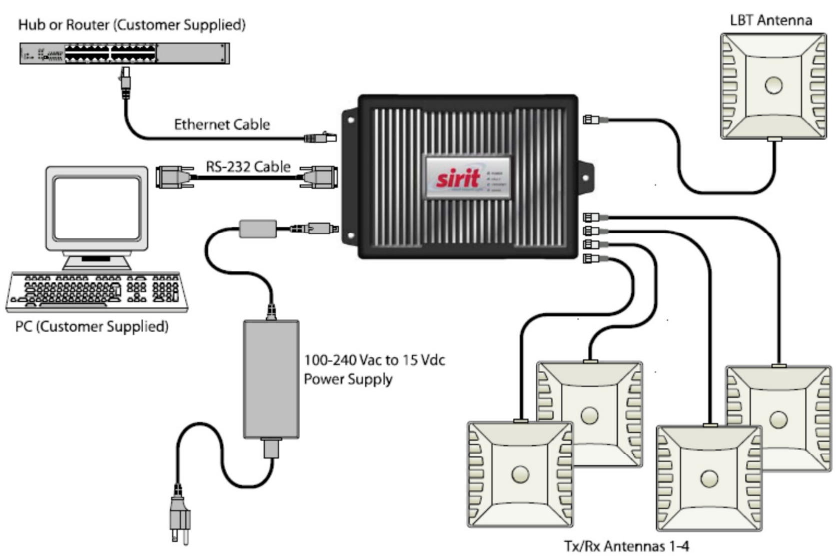

Figure 16: Sirit Reader Hardware Setup

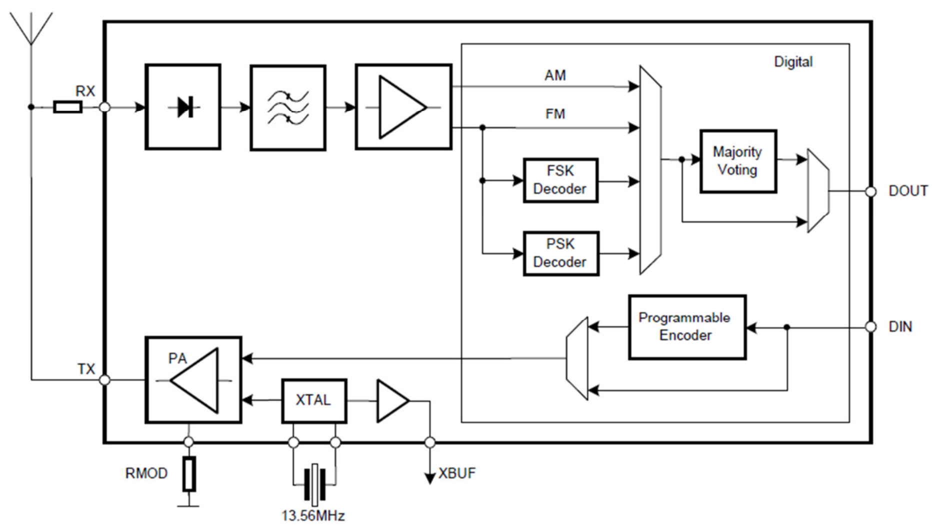

Figure 17: Melexis Internal Hardware

The reader is setup by the programmable encoder and the data is processed through the FSK decoder and majority voting according to the set up code.

Appendix C

C code for the primary control node with attached RFID reader.

```
/*
 * AVRGCC125.c
 *
 * Created: 5/1/2011 12:09:14 PM
 *  Author: Nolan CLark

 * 125 kHz solution for RFID based Home Automation System
 */
#define F_CPU 8000000
#define FOSC 8000000            // Clock Speed
#define BAUD 2400
#define UBRR FOSC/16/BAUD-1

#include <avr/io.h>
#include <util/delay.h>
#include <avr/interrupt.h>

//- initialization functions
void LCD_communication(int data);
void LCD_character(char character);

void LCD_initialization(void);
void UART_initialization(void);
void EXT_INT_initialization(void);

unsigned char UART_Rx1(void);
unsigned char UART_Rx0(void);

//- GLOBAL VARIABLES
unsigned char                   ID_Aux[12];
char                            no_info[40] =       "Not Registered";
unsigned char                   ID0[10]=            "0F03040B86";
unsigned char                   ID1[10]=            "0F03028F0B";
unsigned char                   ID2[10]=            "0F03042CB1";
unsigned short int      in_area01FLAG[2];
unsigned short int      in_area02FLAG[2];

int main(void)
{
        int count = 0;
        int i = 0;
        unsigned char data;
        unsigned char ID[10];

        unsigned char temp_data;

        DDRA = 0xff;
        DDRB = 0xff;
        DDRC = 0xff;
        DDRD = 0xff;
        DDRH = 0xff;
        DDRJ = 0xff;
        DDRF = 0xff;
```

```
PORTA = 0x00;

LCD_initialization();
UART_initialization();
EXT_INT_initialization();

PORTF = 0xFF;
PORTB = 0x00;

LCD_communication(0x01); //0x01

//- Enable interrupts
sei();

while(1)
{
        _delay_ms(800);

        PORTF = 0x00;

                temp_data = UART_Rx1();
                LCD_communication(0x01); //0x01

                for(i = 0; i < 11; i++){
                        temp_data = UART_Rx1();
                        //- Remove Start and Stop Bits
                        if(temp_data == 0x0A || temp_data == 0x0D){
                                //- do nothing (do not save)
                        } else {
                                ID[i] = temp_data;
                                _delay_us(40);
                        }

                        PORTF = 0xFF;
                }

        //- Check ID number room presences
        //- Set lighting output
        //- ID0
        if(ID[8] == ID0[8] && ID[7] == ID0[7]){

                for(i=0;i<10;i++){
                        LCD_character(ID0[i]);
                }

                in_area01FLAG[0] ^= 1;

                if(in_area01FLAG[0] == 1){
                        PORTA = 0x29;
                }else{
                        PORTA = 0x02;
                }

                if(in_area01FLAG[0] == 1 && in_area01FLAG[1] == 1){
                        PORTA = 0x69;
                } else if(in_area01FLAG[0] == 0 && in_area01FLAG[1] == 0){
```

```
                        PORTA = 0x16;
                } else if(in_area01FLAG[0] == 0 && in_area01FLAG[1] == 1){
                        PORTA ^= 0x72;
                } else if(in_area01FLAG[0] == 1 && in_area01FLAG[1] == 0){
                        PORTA ^= 0x0D;
                }
        //- ID1

        } else if(ID[8] == ID1[8] && ID[7] == ID1[7]){
                for(i=0;i<10;i++){
                        LCD_character(ID1[i]);
                }

                in_area01FLAG[1] ^= 1;
                if(in_area01FLAG[1] == 1){
                        PORTA = 0x48;
                }else{
                        PORTA = 0x04;
                }

                if(in_area01FLAG[0] == 1 && in_area01FLAG[1] == 1){
                        PORTA = 0x69;
                } else if(in_area01FLAG[0] == 0 && in_area01FLAG[1] == 0){
                        PORTA |= 0x16;
                } else if(in_area01FLAG[0] == 0 && in_area01FLAG[1] == 1){
                        PORTA ^= 0x72;
                } else if(in_area01FLAG[0] == 1 && in_area01FLAG[1] == 0){
                        PORTA ^= 0x0D;
                }
        //- ID2
        } else if(ID[8] == ID2[8] && ID[7] == ID2[7]){
                for(i=0;i<10;i++){
                        LCD_character(ID2[i]);
                }

                in_area01FLAG[2] ^= 1;
                if(in_area01FLAG[2] == 1){
                        PORTA = 0x09;
                }else{
                        PORTA = 0x12;
                }

        }else{
                for(i=0;i<14;i++){
                        LCD_character(no_info[i]);
                }
        }
        //- change indicator lights on stk-600
        PORTC = (in_area01FLAG[0]<< PC0) | (in_area01FLAG[1]<< PC1) | (in_area01FLAG[2]<< PC2);
    }
    return 0;
}

void EXT_INT_initialization(void)
{
    DDRE = 0x00;
    EICRA = (1 << ISC71) | (1 << ISC70);
```

```
	EIMSK = (1 << INT7);
}

void UART_initialization(void)
{
	//- RFID reader
	/* Set baud rate */
	UBRR1 = UBRR;
	/* Enable receiver */
	UCSR1B = (1<<RXEN1);
	/* Set frame format: 8data, 2stop bit */
	UCSR1C = (0<<USBS1)|(3<<UCSZ10);

	//- AUX UNIT
	/* Set baud rate */
	UBRR0 = UBRR;
	/* Enable receiver */
	UCSR0B = (1<<RXEN0);
	/* Set frame format: 8data, 2stop bit */
	UCSR0C = (0<<USBS0)|(3<<UCSZ00);
}

unsigned char UART_Rx1(void)
{
	/* Wait for data to be received */
	while ( !(UCSR1A & (1<<RXC1)));
	/* Get and return received data from buffer */
	return UDR1;
}

unsigned char UART_Rx0(void)
{
	/* Wait for data to be received */
	while ( !(UCSR0A & (1<<RXC0)) );
	/* Get and return received data from buffer */
	return UDR0;
}

//- PE7
ISR(INT7_vect)
{
	short int i;
	unsigned char temp_data;
	unsigned char ID[10];

	LCD_communication(0x01); //0x01
	//- read in ID
	for(i = 0; i < 12; i++){
		temp_data = UART_Rx0();
		ID_Aux[i] = temp_data;
	}
	for(i = 0; i < 10; i++){
			ID[i] = ID_Aux[i+1];
	}
	_delay_ms(1);
		//- Check ID number room presences
		//- Set lighting output
```

```
//- ID0
if(ID[8] == ID0[8] && ID[7] == ID0[7]){

        for(i=0;i<10;i++){
                LCD_character(ID0[i]);
        }

        in_area01FLAG[0] ^= 1;

        if(in_area01FLAG[0] == 1){
                PORTA = 0x29;
        }else{
                PORTA = 0x02;
        }

        if(in_area01FLAG[0] == 1 && in_area01FLAG[1] == 1){
                PORTA = 0x69;
        } else if(in_area01FLAG[0] == 0 && in_area01FLAG[1] == 0){
                PORTA = 0x16;
        } else if(in_area01FLAG[0] == 0 && in_area01FLAG[1] == 1){
                PORTA ^= 0x72;
        } else if(in_area01FLAG[0] == 1 && in_area01FLAG[1] == 0){
                PORTA ^= 0x0D;
        }
//- ID1
//- check and output light settings
} else if(ID[8] == ID1[8] && ID[7] == ID1[7]){
        for(i=0;i<10;i++){
                LCD_character(ID1[i]);
        }

        in_area01FLAG[1] ^= 1;
        if(in_area01FLAG[1] == 1){
                PORTA = 0x48;
        }else{
                PORTA = 0x04;
        }
        if(in_area01FLAG[0] == 1 && in_area01FLAG[1] == 1){
                PORTA = 0x69;
        } else if(in_area01FLAG[0] == 0 && in_area01FLAG[1] == 0){
                PORTA |= 0x16;
        } else if(in_area01FLAG[0] == 0 && in_area01FLAG[1] == 1){
                PORTA ^= 0x72;
        } else if(in_area01FLAG[0] == 1 && in_area01FLAG[1] == 0){
                PORTA ^= 0x0D;
        }


//- ID2
} else if(ID[8] == ID2[8] && ID[7] == ID2[7]){
        for(i=0;i<10;i++){
                LCD_character(ID2[i]);
        }

        in_area01FLAG[2] ^= 1;
        if(in_area01FLAG[2] == 1){
```

```
                        PORTA = 0x09;
                }else{
                        PORTA = 0x12;
                }

        }else{
                for(i=0;i<14;i++){
                        LCD_character(no_info[i]);
                }
        }
        //- change indicator lights on stk-600
        PORTC = (in_area01FLAG[0]<< PC0) | (in_area01FLAG[1]<< PC1) | (in_area01FLAG[2]<< PC2);
    }

void LCD_communication(int data)
{
        //communicate with LCD display
        PORTJ = 0x00;
        PORTJ = 0x04;
        PORTH = data;
        PORTJ = 0x00;
        _delay_us(40);}

void LCD_initialization()
{
        //initialization sequence--14/2 LCD display
        _delay_ms(20);
        //function set
        LCD_communication(0x38); //0x38
        _delay_us(40);
        //display ctrl
        LCD_communication(0x0F); //0x0F
        _delay_us(40);
        //clear display
        LCD_communication(0x01); //0x01
        _delay_us(40);
        //entry mode
        LCD_communication(0x06); //0x06
        _delay_ms(2);

        //initialization complete}
void LCD_character(char character)
{
        //send character data
        PORTJ = 0x01;
        PORTJ = 0x05;
        PORTH = character;
        PORTJ = 0x01;
        _delay_us(40);}
```

Appendix D

Code for Melexis RFID reader for antenna modulation

```
/*
 * AVRGCC1.c
 * 13.56 MHz RFID reader with LCD attachment
 * Created: 4/6/2011 10:31:06 PM
 * Author: Nolan Clark
 */

//- mods
/*
 * Change SPI to slave? (No)
 * Determine clocking frequency -- ok (5/14)
 * set up MOSI & MISO                          --
ok (5/14)
 * analyze ISO protocol
 *
*/

//- Communication with RFID reader
/* Drive by MODE & RTB (Operating mode)
 * MODE = 0 Config : MODE = 1 Communication
 * MODE RTB Function
 * 0      0      Configuration
 * 1      0      Transmission
 * 0      1      Reserved
 * 1      1      Reception
 * DIN & CK used to control chip & transmit data
 * MODE assert high each time sending register config
info (5us)
 * uC --> Direction Transmission & Reception
 * >>CONFIG MODE
 *       MSB first
 *       (4-bit Address & 8-bit Data fields)
 *       CK 12 pulses
 * Assert MODE H after every 12-bit write cycle
 * Select ANalog modulation depth (TModIndex bit)
Analog Config register
 *
*/

//- IO PIN ASSIGNMENT
/*      PORTH (0-7)     LCD DATA
 *      PORTJ (0-3)     LCD CNTRL
 *      PORTA (0-7)     SWITCHES
 *      PORTD (0-7) LEDs
 * PB0  SS/             SYNC
 * PB1  SCK             SCK
 * PB2  MOSI    MOSI
 * PB3  MISO    MISO
 * PC0  OUT             MODE
 * PC1  OUT             RTB
(Transmission/Reception SEL)
 */

//- Define library
#define F_CPU 8000000
#include <avr/io.h>
#include <util/delay.h>

#define MISO 3
#define MOSI 2
#define SCK  1
#define SS   0

//- Function definitions
int LCD_communication(int data);
int LCD_initialization(void);
int LCD_character(char character);
void Initialize_SPI_Master(void);
void Transmit_SPI_Master(int Data);
void Initialize_IO(void);
int Recieve_SPI_Master(void);

int main(void)
{
        char hello[40] = "RFID Proto!!!";
        char data_type0[5] = "data0";
        int count = 0;
        int i = 0;
        int input_data;

        Initialize_IO();

        for(i=0;i<14;i++)
        {
                LCD_character(hello[i]);
                _delay_us(40);
        }

        ////SET MODE///
        //- Set Configuration mode for RFID reader
        //- MODE = 0 RTB = 0
        _delay_ms(500);
        PORTD = 0xDD;
        _delay_ms(10);

        PORTC = 0x00;

        //- Output configuration data to RFID
        //- ISO configuration for ISO15693 ASK
(100% Modulation)
        //- Determined from MLX90121 data sheet
section 13, pg. 21
```

```
//- 12-bit data (address,data)
PORTC = 0x01;
PORTC = 0x00;
Transmit_SPI_Master(0x047);
PORTC = 0x01;
PORTC = 0x00;
Transmit_SPI_Master(0x101);
PORTC = 0x01;
PORTC = 0x00;
Transmit_SPI_Master(0x200);
PORTC = 0x01;
PORTC = 0x00;
Transmit_SPI_Master(0x309);
PORTC = 0x01;
PORTC = 0x00;
Transmit_SPI_Master(0x4BF);
PORTC = 0x01;
PORTC = 0x00;
Transmit_SPI_Master(0x5EF);
PORTC = 0x01;
PORTC = 0x00;
Transmit_SPI_Master(0x6FB);
PORTC = 0x01;
PORTC = 0x00;
Transmit_SPI_Master(0x7FE);
PORTC = 0x01;
PORTC = 0x00;
Transmit_SPI_Master(0x87B);
PORTC = 0x01;
PORTC = 0x00;
Transmit_SPI_Master(0x9DF);
PORTC = 0x01;
PORTC = 0x00;
Transmit_SPI_Master(0xA1F);
PORTC = 0x01;
PORTC = 0x00;
Transmit_SPI_Master(0xB7F);
PORTC = 0x01;
PORTC = 0x00;
Transmit_SPI_Master(0xC3F);
PORTC = 0x01;
PORTC = 0x00;

//- READER To Reception Mode
PORTC = 0x03;

PORTD = 0xFF;
while(1)
{
	//_delay_ms(500);
	//input_data = Recieve_SPI_Master();

	input_data = 0;
	if(input_data != 0){
		for(i=0;i<4;i++)
		{
			LCD_character(data_type0[i]);

			_delay_us(40);
		}
	}
	//count and space transmissions
	PORTD = ~count>>1;
	count++;


	Transmit_SPI_Master(0x056);
}

//- PROJECT pseudo CODE

/*
TAGs stored into memory

Read in ID number from RFID reader 1
determine person associated with tag (light setting)
1st read (room entry)

Check if reader 2 has ID data waiting for transmission

2nd read on (same tag) any reader (room exit)

reset room lighting

output light information to XBee via Tx

*/

	return 0;
}

void Initialize_IO(void)
{
	DDRA = 0x00;
	DDRC = 0xFF;
	DDRH = 0xFF;
	DDRJ = 0xFF;
	DDRD = 0xFF;
	PORTD = 0xFF;

	LCD_initialization();

	//- make sure SPI is not in low power mode
	PRR0 = 0<<PRSPI;

	//- make MOSI, SCK and SS outputs
	DDRB = 1<<MOSI | 1<<SCK | 0<<SS | 0<<MISO;

	Initialize_SPI_Master();
}
```

```
int LCD_communication(int data)
{
	//communicate with LCD display
	PORTJ = 0x00;
	PORTJ = 0x04;
	PORTH = data;
	PORTJ = 0x00;

	_delay_us(40);

	return 0;
}

int LCD_initialization(void)
{
	//initialization sequence--14/2 LCD display
	_delay_ms(20);
	//function set
	LCD_communication(0x38); //0x38
	_delay_us(40);
	//display ctrl
	LCD_communication(0x0F); //0x0F
	_delay_us(40);
	//clear display
	LCD_communication(0x01); //0x01
	_delay_us(40);
	//entry mode
	LCD_communication(0x06); //0x06
	_delay_ms(2);

	//initialization complete

	return 0;
}

int LCD_character(char character)
{
	//send character data
	PORTJ = 0x01;
	PORTJ = 0x05;
	PORTH = character;
	PORTJ = 0x01;
	_delay_us(40);

	return 0;
}

void Initialize_SPI_Master(void)
{
   SPCR = (0<<SPIE) |		//No interrupts
   (1<<SPE) |			//SPI enabled
   (1<<DORD) |			//shifted out
MSB (verified 5/15/11)
   (1<<MSTR) |			//master
   (0<<CPOL) |			//rising
leading edge
   (1<<CPHA) |			//sample
leading edge
   (0<<SPR1) | (0<<SPR0); //clock speed
   SPSR = (0<<SPIF) |		//SPI interrupt flag
   (0<<WCOL) |			//Write
collision flag
   (0<<SPI2X) ;			//Doubles SPI clock

   // PORTB = 1 << SS;  // make sure SS is high
}


void Transmit_SPI_Master(int Data)
{
	// assert the slave select
   //PORTB = 0 << SS;

   // Start transmission
   // send high byte first
   SPDR = (Data >> 8) & 0xF;

   // Wait for transmission complete
   while (!(SPSR & (1<<SPIF)));

	// Wait for transmission complete
   //while ( (SPSR & (1<<7)) == 0){}

	// send low byte next
   SPDR = 0xFF & Data;

   // Wait for transmission complete
   while (!(SPSR & (1<<SPIF)));

   // de-assert slave select
   //PORTB = 1 << SS;
}


int Recieve_SPI_Master(void)
{
	PORTD = 0xFE;

	while(!(SPSR & (1<<SPIF)));

	PORTD = 0xEE;

	return SPDR;
}
```

Bibliography

[1]"Datasheet." *Parallax*. Feb. 2006. Web. 5 May 2011. <http://www.parallax.com/dl/docs/prod/audiovis/RFID-Reader-v1.2.pdf>.

[2]Digi. "XBee-PRO® 802.15.4 OEM RF Modules - Digi International." *Digi International - Making Wireless M2M Easy*. Digi, 2011. Web. 06 June 2011. <http://www.digi.com/products/wireless-wired-embedded-solutions/zigbee-rf-modules/point-multipoint-rfmodules/xbee-series1-module.jsp>.

[3]Melexis. "EVB90121-ND Evaluation Board." *Melexis.com*. Melexis, Sept. 2009. Web. 06 June 2011. <http://search.digikey.com/scripts/DkSearch/dksus.dll?Detail>.

[4]MELEXIS. "MLX90121." *Www.melexis.com*. Melexis, Sept. 2009. Web. Apr. 2011. <http://www.melexis.com/Assets/MLX90121-Datasheet-4755.aspx>.

[5]"Radio-frequency Identification." *Wikipedia*. Web. 5 Apr. 2011. <http://en.wikipedia.org/wiki/Radio-frequency_identification>.

[6]Sirit. *Sirit Infinity 510 Quick Start Guide*. <www.Sirit.com>. Web. Mar. 2011.

[7]Sirit. *Sirit Infinity 510 User Guide*. <www.Sirit.com>. Web. Mar. 2011.

[8]"XBee Wireless Interface — MakingThings." *XBee Wireless Interface*. MakingThings, 2011. Web. 15 May 2011. <http://www.makingthings.com/documentation/tutorial/xbee-wireless-interface/tutorial-all-pages>.