School of Engineering Science
Simon Fraser University
8888 University Drive
Burnaby, BC
V5A 1S6

March 5, 2009

Mr. Steve Whitmore
School of Engineering Science
Simon Fraser University
Burnaby, British Columbia
V5A 1S6

RE: ENSC 440 Design Specifications for Watchbird™ Home Monitoring System.

Dear Mr. Whitmore,

"The Watchbird™ Remote Peace of Mind System: Design Specifications", attached to this letter, details the design of the Watchbird™ system.

These design specifications first summarize the Watchbird™ system and give an overview of its functional specifications. An overview of the full system design is then given, including the system's timing requirements and its environmental impact. The design of the individual components of the system is then discussed, providing reasoning for the design choices that Chickadee Tech has made for the system.

The document then presents the second part of Chickadee Tech's test plan, which describes in detail how we will test the prototype and the production model to ensure that the design is satisfactory and the system's functional requirements are met in full.

Please let me know if you have any questions, comments, or concerns about this document. I can be contacted at 604.837.4009 or by email (smg2@sfu.ca). Thank you very much for your time in reviewing the design of the Watchbird™ system.

Regards,

Samantha Grist
President
Chickadee Tech

The Watchbird™ Remote Peace of Mind System:

Design Specifications

Samantha Grist
Derek Lowes
Andra St. Quintin
Shiou-Min (Eric) Shen

March 5, 2009

Executive Summary

This document presents the design specifications for implementing the Watchbird™ system. As described in our functional specifications, the Watchbird™ system will provide users with a method for remotely checking the status of household components such as locks, windows, and ovens, as well as the ability to lock or unlock their front door. In summary, the user will initiate a query or request using a cell phone SMS text message, and receive a status message back. These messages will be relayed by a central Chickadee Tech server to the base station at the user's home, where sensors and actuators will be contacted prior to relaying a message back through the same path.

With the functional specifications of our prototype in mind, we first present the overall system design, describing how individual components of the system will interface with each other, in terms of software, timing logistics, and physical interfacing. Message transmission protocols are also discussed: communication between cell phones and the server will take place using text messages relayed via Twitter; communication between the server and base station will use TCP, and communications between the base station and sensor and actuator locations will take place using a fixed 418 MHz signal. The designs of the server, base station, sensor-end circuitry, and web-based user configuration are described based on how they will implement these protocols. At the component level, we explain the selection and design of individual components such as the RF transceivers, the contact switch sensors, and the lock actuators.

In order to verify the operation of our design, Chickadee Tech has devised a test plan. It lists intermediate tests that will be conducted during the development of sections of the system, as well as overall system tests that will be conducted once the entire system has been integrated. The intermediate tests will ensure development is proceeding correctly, and will help pinpoint the causes of any problems. The system wide tests will provide confidence that the entire system is fully functional in a user's home.

TOC

List of Figures

List of Tables

Glossary

Term/Acronym	**Definition**
A to D	Analog to Digital
IP	Internet Protocol
LCD	Liquid Crystal Display
LR	Long Range
MOSFET	Metal Oxide Semiconductor Field Effect Transistor
RF	Radio Frequency
RJ-45	Standard Ethernet connector
RSSI	Received Signal Strength Indicator
SAP	Sensor-Actuator Pair
SEC	Sensor End Circuitry
SPDT	Single-Pole, Double-Throw
SQL	Structured Query Language
TCP	Transmission Control Protocol
WMR	Waiting Message Request

1 Introduction

Watchbird™ is a home monitoring system that allows homeowners to request and obtain home status updates through cell phone text messages. The monitoring process involves the interaction between 3 components: the sensor and actuator units, the base station which controls the sensor and actuator units, and the server which bridges the communication between cell phones and Watchbird™. This document details the specifications for designing a proof-of-concept prototype, which will be demo-ready by April 2009. The prototype will support a single user account, 2 cell phone numbers, 3 sensors, and 1 actuator.

1.1 Scope

This document provides a general overview of the full Watchbird™ system and specifies the design for each component in detail. The design specifications for the prototype Watchbird™ fulfill the functional requirements for building a prototype and are an essential step in Chickadee Tech's realization of the Watchbird™ prototype.

1.2 Intended Audience

This document is written exclusively for the design team of Chickadee Tech and is to be used as a guideline for constructing the Watchbird™ prototype. In addition, this document is a tool for measuring the progress of the implementation process.

2 System Functionality/Specifications

As detailed in the Watchbird™ functional specifications, a user will send a text message to query the status of their sensors or to request their door be locked or unlocked. Chickadee Tech's server will check each incoming message for validity, parse the message into its expected parts, and relay the message to the appropriate base station. The base station will query the status of each sensor in turn, and actuate the door lock if necessary. If communication to any sensor is unsuccessful on the first try, the base station will make a second attempt at communication. Once it has collected the status of all sensors, it will relay a message back to the server, and the server will match it to the corresponding waiting request to send a reply.

From the user's point of view, all sensors will be easy to install in existing locations. In the case of a lock actuator, the locking system will be easy to install in a standard door. The user will then be able to initialize each sensor and actuator using a web-based interface connected to the central server.

3 Full System Design

The Watchbird™ full-system design will be broken up into hardware and software components. This section gives a high-level overview of the system functionality in both of these areas.

The Watchbird™ black-box problem analysis (as presented in the Watchbird™ functional specification) is given below.

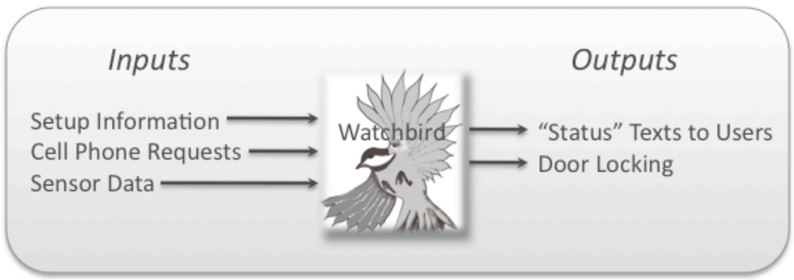

Figure 1: Watchbird™ black-box analysis.

The system was designed to provide maximal efficiency in dealing with these necessary inputs and outputs.

3.1 Full-System Hardware

3.1.1 Full-System Block Diagram

The Watchbird™ hardware will consist of several different components, since the Watchbird™ system itself can be broken up into several different subsystems. Figure 2 outlines the many components of the prototype Watchbird™, and the orange text indicates an overview of the hardware design choices for each of the components. The detailed design of each component is presented in Section 4.

A preexisting Chickadee Tech company laptop was chosen for the prototype server to reduce development costs of the system. The functional requirements of the server are minimal enough to make this an ideal choice. A microcontroller with Ethernet was chosen for the base station to minimize costs for the end user while still fulfilling all of the required functionality of the base station. Linx Technologies RF receivers and transmitters were deemed ideal for the RF transmission due to their low cost, simple design, long range, and low power consumption. Electrical contact switch sensors were chosen for the home sensors because they provide simple, bi-state functionality while consuming very little power. Finally, a modified keypad door lock was chosen for the prototype lock actuator because it provided a pre-integrated lock and lock actuator, as well as providing keypad door locking functionality.

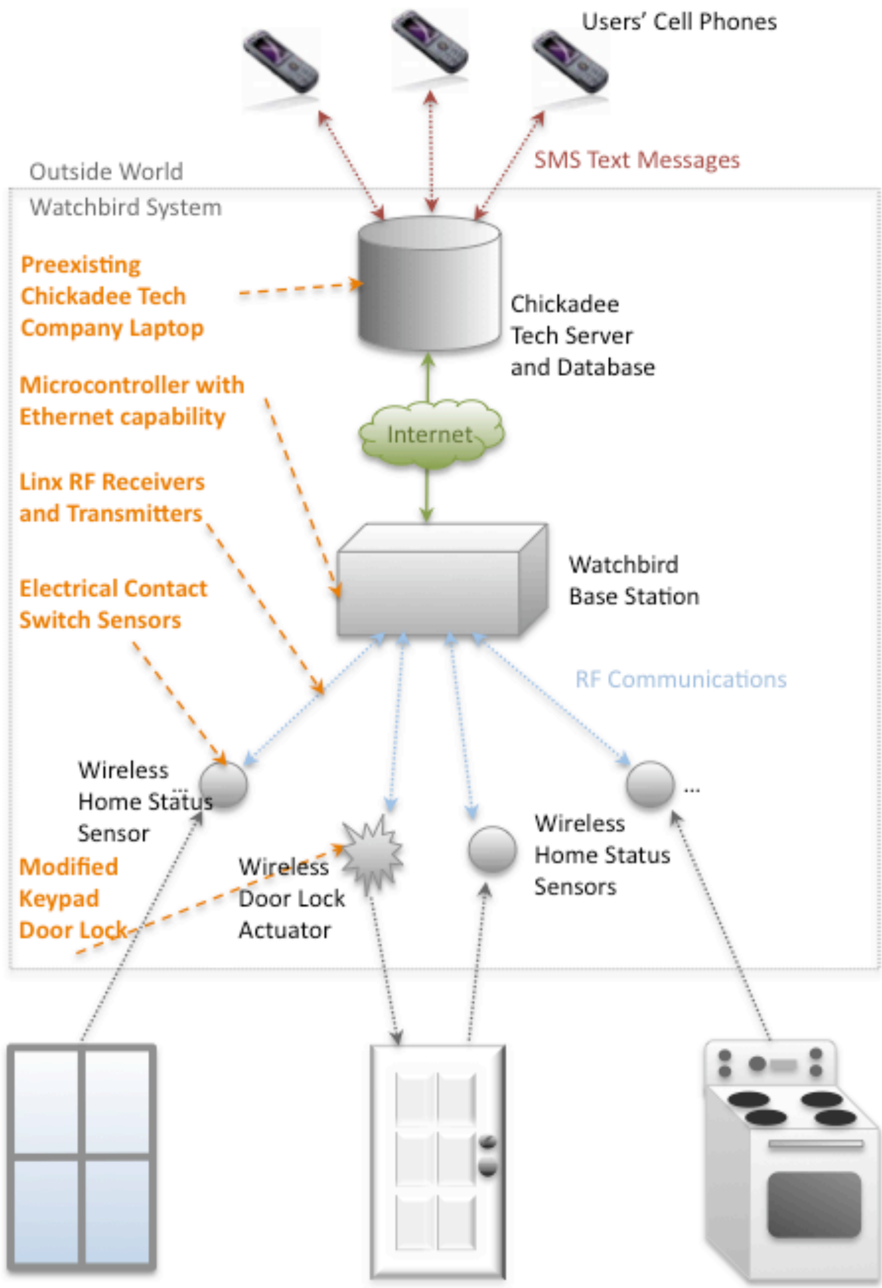

Figure 2: Full-system block diagram with design notes.

The flow chart in Figure 3 shows the connections between the individual components outlined above within the black box problem analysis shown in Figure 1. The system inputs are highlighted in blue, while the system outputs are highlighted in purple.

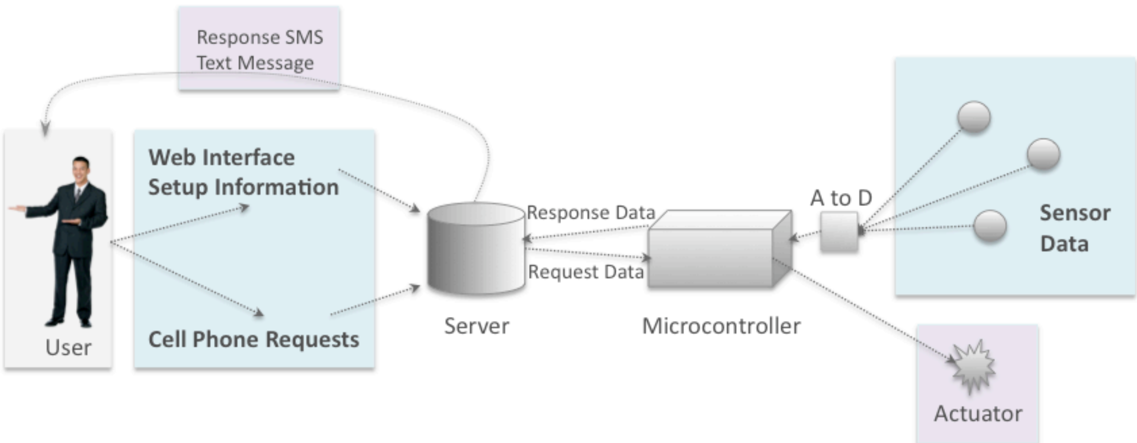

Figure 3: Input/Output Diagram.

3.1.2 Casing and Mounting Design

The casings for the individual Watchbird™ prototype's components will be made from durable plastic. Plastic was chosen because it will minimally interfere with the wireless communications, unlike a metal case. Additionally, it is inexpensive and the Watchbird™ components do not need to withstand large forces.

The entire base station (containing the RF receiver and transmitter there) will be encased. At the sensor end, there will be two components: the RF section, logic, and batteries, encased in plastic, and the sensor itself. The sensor itself will be connected to the RF section by two wires. At a location with both a sensor and an actuator, the actuator, keypad, and controller circuitry for the keypad will be contained in a third plastic housing.

The sensors and their circuitry will be mounted using Command™ adhesive [1]. This type of adhesive adheres firmly to smooth surfaces (i.e. doors, windows, walls, and stoves), and may also be easily removed without damage to the surface in most cases. These qualities make Command™ adhesive ideal for mounting the Watchbird™ sensor and sensor-end circuitry. The actuator, keypad, and controller circuitry will be mounted on the door itself, in a non-easily removable fashion.

3.1.3 Environmental Impact

The Watchbird™ system will have a minimal environmental impact during its product lifetime. The system's power consumption will be comparable to that of a standard desktop computer, and the system itself will not produce any harmful emissions. Furthermore, all batteries in the system will be rechargeable or have a rechargeable option.

When the product's lifetime has ended, Chickadee Tech also aims for the disposal of its products to be as environmentally friendly as possible. The production casings of the different components will be made from recyclable materials, and the components themselves may be returned to Chickadee Tech. After product return, Chickadee Tech will salvage all parts possible to be recycled or reused. Finally, Chickadee Tech will provide a repair service to its customers so that the lifetime of each Watchbird™ product will be as long as possible.

3.2 Full-System Software Overview

The Watchbird™ system will be driven by a number of software components. There will be software running the base station, a server to manage communication between various parts of the system, a web-site for configuring the system, and a database to store information. This section will provide a brief overview of the purpose of each component. Further detail is provided in their individual subsections of Section 4.

3.2.1 Server

All communication between the base station and the user will go through the server. When the user sends a text to query, lock, or unlock their system, the server will verify the cell phone number and password, determine the correct base station to contact, and pass along the request. When the base station sends a reply with the status, the server then formats the correct response text message and sends it back to the user. The server will also handle requests through the website to add or remove sensors from the system and pass these requests to the base station as necessary.

3.2.2 Base Station

The base station software manages the sensors and communication from the server. When it receives a request through the server, it queries the sensors or sends commands to lock the door as appropriate to the request, and then replies to the server. Additionally, each time the base station is powered on, it is responsible for creating the connection to the server for use in further communication.

3.2.3 Website

The website is used to manage configuration of the system. This includes choosing what name to display for each sensor and actuator, as well as adding and removing sensors and actuators from the system. When the user makes changes, the website will update the database directly.

3.2.4 Database

The database is used to store system information. It also acts as the communication pathway between the website and the server.

3.3 Timing Design

The Watchbird™ system will have strict timing requirements. Not including text message delivery times, each sensor status query request will take a maximum of 31 seconds, while each lock or unlock request will take a maximum of 35 seconds. The breakdown of the individual times that make up this overall time for a sensor query request and a lock request are shown in Figure 4 and Figure 5, respectively. These timing overviews are for 10 sensors (the maximum supported by Watchbird™) as well as for the worst-case scenario. The base station allows one re-send of the request to each sensor and actuator after timing out with no reply. The typical time to process each request will be less than half of that outlined in the diagrams below.

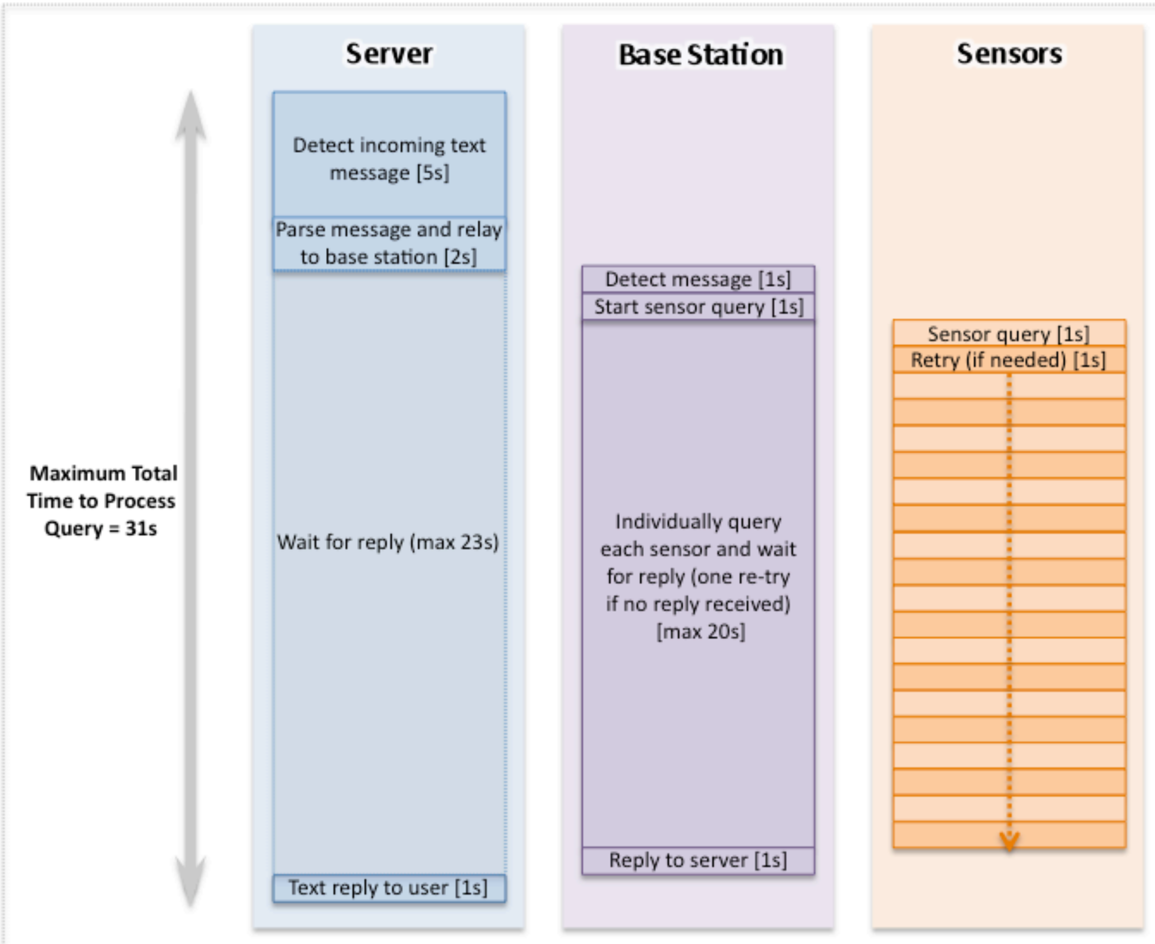

Figure 4: Worst-case timing breakdown for a sensor query request.

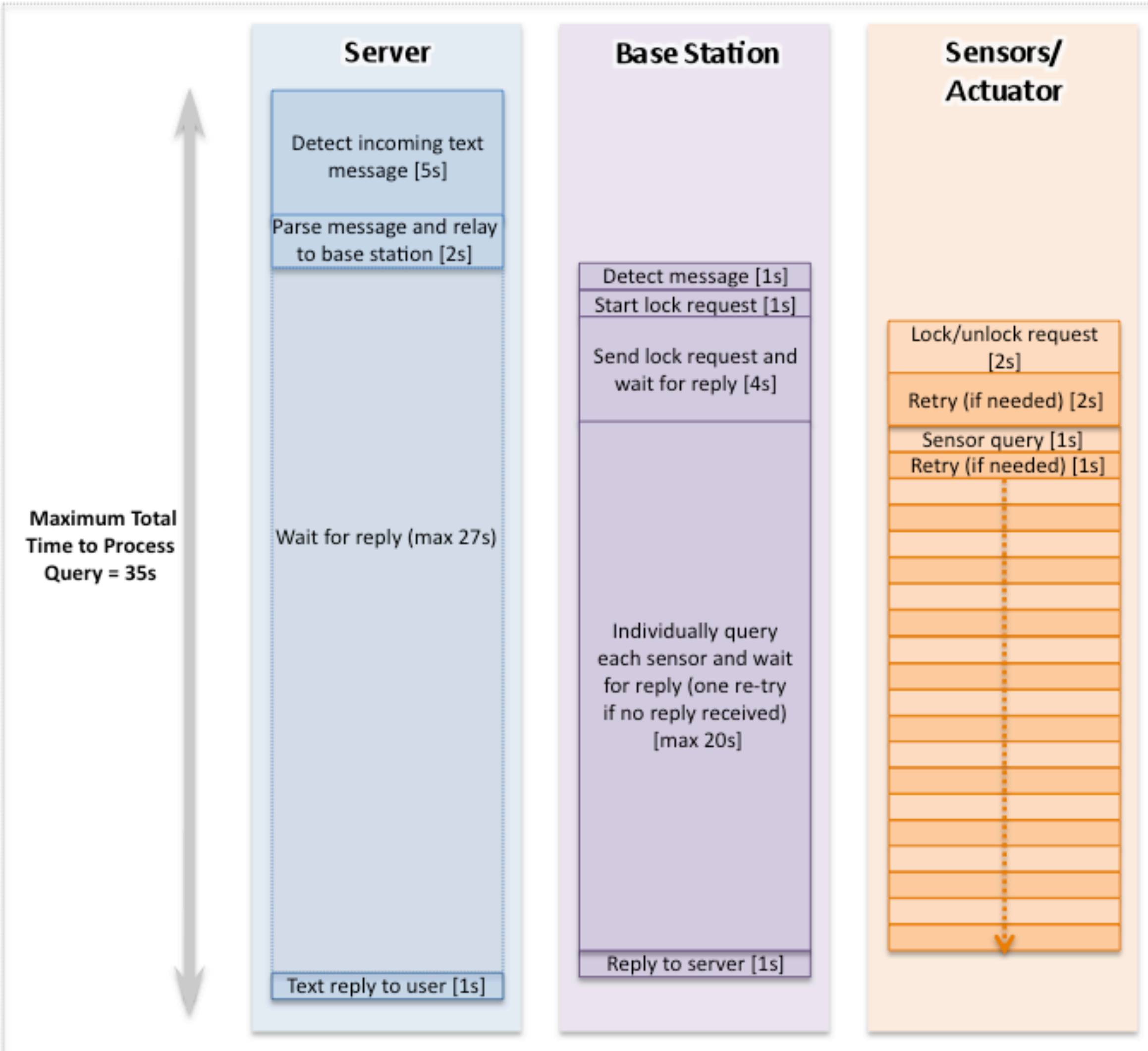

Figure 5: Worst-case timing breakdown for a lock/unlock request.

4 Design for Individual Components

4.1 Cell Phone – Server Communication Protocol

Watchbird™ is based on the concept of users checking the status of their homes at any time using text messages on their cell phones. The following protocol will be used for submitting queries from cell phones, transmitting responses to cell phones, and coordinating queries at the server.

4.1.1 Submitting Queries

At the production level, users will send a text message to a number directly associated with Chickadee Tech's server. However for the prototype, Twitter will serve as an existing intermediary between text messages and the server, as summarized in the following steps:

- The user sends a text message to the Twitter number (21212).
- Twitter automatically sends an email to Chickadee Tech's query management email address.
- Chickadee Tech's server checks the email account every 5 seconds for new queries.

The user's text message must be in the form described by Table 1, which describes the example message "d Watchbird 11235813FibSeq status".

Table 1: Components of a user's text message.

Message Component	Example	Notes
d Watchbird	d Watchbird	This sequence is necessary for Twitter to automatically send an email rather than just posting an online Twitter update.
password	11235813FibSeq	Password will be between 7 and 20 ASCII characters.
request	status	Can be "status", "lock", or "unlock". The latter two options automatically also provide a status update.

4.1.2 Transmitting Responses

Within 31 seconds of receiving a query, or within 35 seconds of receiving a locking or unlocking request, the server will send a text message reply to the user. The user should (depending on their cell phone provider's text message delivery delays) receive this reply shortly thereafter. The server will compose a response message, which will be sent directly as a text message using Chickadee Tech's query management email address. The email address for the text message is determined by the user's cell phone number and service provider, both of which will be supplied by the user when they set up their Watchbird™ system and subsequently stored in the database. In general, the address is of the form "10digitnumber@provider.com". Domains for some common Canadian cell phone providers are listed in Table 2.

Table 2: List of common provider domains for sending email-to-text messages to cell phones [2].

Provider	Domain
Bell Mobility Canada (and Solo Mobile)	@txt.bellmobility.ca
Fido Canada	@fido.ca
Koodo	@msg.koodomobile.com
Rogers Canada	@pcs.rogers.com
TELUS Mobility	@msg.telus.com
Virgin Mobile Canada	@vmobile.ca

The types of permissible messages are listed in Table 3, along with examples illustrating their structure. The exact message will reflect the names given to individual sensors by users, as well as their status as determined by the base station.

Table 3: List of types of text message responses with examples.

Message Type	Example Message
status report	Status Report: door – unlocked; oven – off; bedroom window – open; kitchen window – unresponsive.
locking confirmation	Locking confirmed. Status Report: door – locked; oven – off; bedroom window – open; kitchen window – unresponsive.
locking error	Error: unable to lock. Status Report: door – unlocked; oven – off; bedroom window – open; kitchen window – unresponsive.
unlocking confirmation	Unlocking confirmed. Status Report: door – unlocked; oven – off; bedroom window – open; kitchen window – unresponsive.
unlocking error	Error: unable to unlock. Status Report: door – locked; oven – off; bedroom window – open; kitchen window – unresponsive.
unresponsive base station	Error: no base station response.
incorrect password	Error: incorrect password.
invalid request	Error: invalid request.

4.1.3 Coordinating Queries

As mentioned above, the server will check Chickadee Tech's query management email address every 5 seconds. Each valid incoming email will be stored as a waiting message request (WMR), which will be handled by the server as shown in Figure 6.

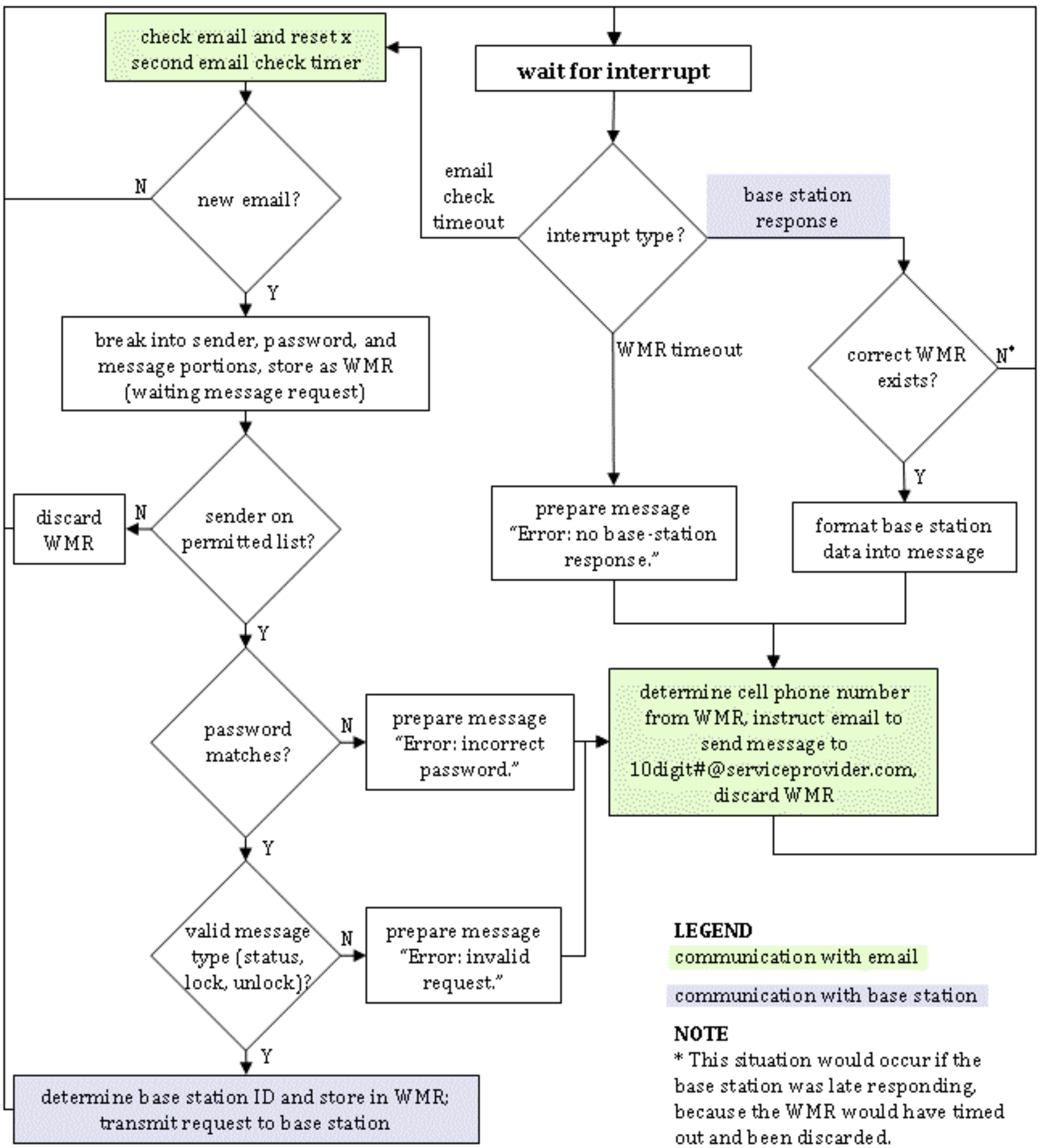

Figure 6: Software flowchart for server management of text message requests.

4.2 Server

4.2.1 Hardware

The Chickadee Tech server does not have strict hardware requirements. The server software for the prototype will be simple enough to run on an existing company laptop, thus preventing the group from incurring further costs for this unit in development.

4.2.2 Software

The server will manage communication between the users' cell phones, the web interface, the database, and all base stations. This includes text message driven requests such as querying the status of the sensors or locking a door, website driven requests such as adding new sensors, and responses from the base station.

The various tasks performed by the server are mostly independent of each other, and have fairly short durations. As such, the server will be designed to spend much of its time in a ready state where it periodically checks for requests or other tasks to perform. When it detects a request, it will then complete the task before returning to the ready state. The various tasks are described in Table 4.

Table 4: General server software components.

Event	Source	Actions to Take
Base station powered on	Base station	• Verify that the base station is entered in the database
Sensor status request	User (via text)	• Verify that the source cell phone is a valid number • Check the password against the database • Establish a connection with the correct base station • Send the base station a request for the sensor status • Store the request source for later use
Sensor configuration (Add/Remove)	Website	• Establish a connection with the correct base station • Send the base station information on the sensor to add or remove • Store the request for subsequent reply
Sensor status update	Base station	• Close the connection to the base station • Determine the request source • Send a text message reply to the initiating cell phone number

4.3 Server/Base Station Communication Protocol

The server and base station will communicate with each other over the Internet using Transmission Control Protocol (TCP). TCP communication is connection-based, so in order to send data between the server and base station, a connection must first be established.

One connection will be established between the server and the base station to handle all communication. Firstly, when the base station is powered on, it will establish the connection with the server and inform the server of its existence and serial number. Subsequently, whenever the server is processing a query or locking request, it will use the existing connection. This connection will be used by the server to send the request to the base station, as well as for the base station to reply with the sensor status. The flow of information through the connection between the server and base station is show in Figure 7, and a summary of the types of messages and the information they contain is in Table 5.

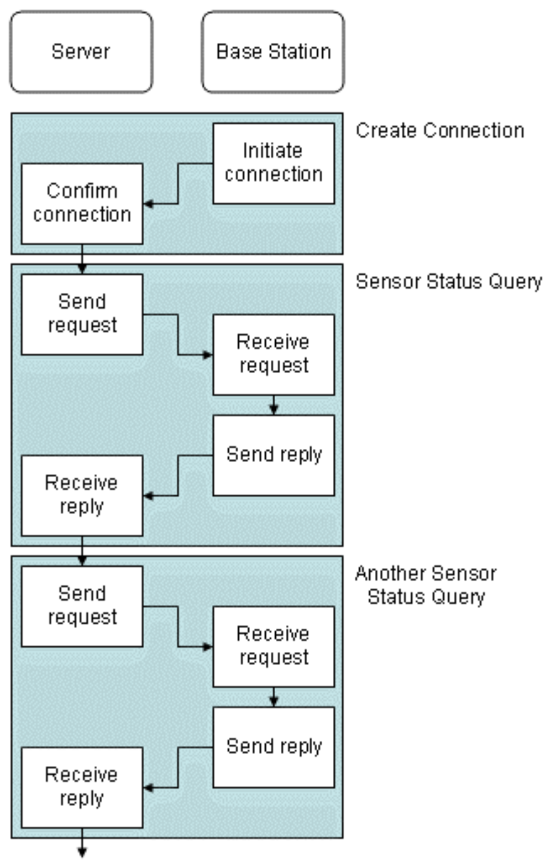

Figure 7: Information flow for connection between Base Station and Server

Table 5: List of messages for base station/server communication

Message Type	Source	Information
Powered On	Base Station	• Base station serial number.
Sensor Query Request	Server	
Lock/Unlock Request	Server	• The code of the actuator. • Lock or unlock
Status Update	Base Station	• Lock/unlock success (if applicable). • The status of each sensor
Sensor Configuration Request	Server	• The codes of all sensors and actuators
Sensor Configuration Confirmation	Base Station	• Success or error message

4.4 Base Station

For the prototype, the PICDEM.net 2 development kit has been chosen to act as the base station. This kit fulfils the necessary hardware functional requirements of our system while allowing our custom-designed production hardware (based upon the same PIC18F97J60 microprocessor present in the prototype) to be much less expensive than our prototype. Therefore, we can reduce the cost of the system for the end user.

4.4.1 Hardware

The base station hardware will be based around the PICDEM.net 2 development kit. The features of this kit that are relevant to our design are shown in Table 6.

Table 6: Features of PICDEM.net 2 kit [3].

Feature	Model Name/Type	Notes
Microcontroller	PIC18F97J60	• IEEE 802.3 compatible Ethernet controller. • Analog and digital I/O pins.
Ethernet Controller	ENC28J60	• Standalone Ethernet controller.
Ethernet Interface	2xRJ-45 connector	• One port controlled by the PIC18F97J60. • Second port controlled by the ENC28J60.
User Interface	16x2 character LCD	

The Ethernet port controlled by the PIC18F97J60 must have an Internet connection in order for the system to communicate with the server.

The microcontroller's I/O pins will be used to interface with the RF receiver and transmitter used by the base station to communicate with the sensors and actuators in the home. The receiver and transmitter at the base station end will be from the Linx LR (Long Range) series [4], operating at a frequency of 418 MHz. Details of these modules and the reasons for choosing them are discussed in Section 4.5. A diagram of the interface between the base station and its RF section is shown in Figure 8.

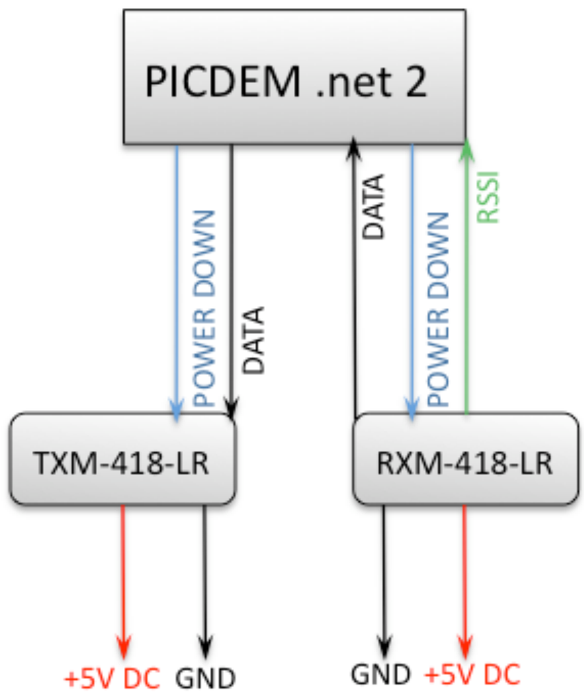

Figure 8: Base station-RF interface

When it does not need to send a request to the sensors or actuators, The PICDEM.net 2 can use one of its IO pins to hold the power down line of the transmitter low. This action places the transmitter in a low-current state. When it wishes to send a request, the base station brings the power down line of the transmitter high and sends encoded digital data to the data line of the transmitter via another of its IO pins.

At the receiver side, the base station does not listen for incoming data unless it has just sent out a request, so it can hold the receiver's power down line low except when waiting for a reply. When it waits for a reply from one of the sensors, the base station holds the power down line high and receives the incoming encoded data on its IO pin connected to the receiver's Data pin. The data pin of the receiver will be connected to an analog input of the PICDEM.net 2, and the RSSI (Received Signal Strength Indicator) pin of the receiver allows the analog to digital converter on the PICDEM.net 2 to properly distinguish between highs and lows in the data.

Both the receiver and transmitter circuitry are powered off of the PICDEM.net 2's on-board +5 V power pins, and are connected to the board's ground. The high frequency RF communication necessitates mounting the receiver, transmitter, and antennas on a custom-made PCB [5]. For prototyping purposes, Chickadee Tech will mount these components onto a PCB with a good ground plane (eliminating the problems of improper grounding and parasitic capacitances which would detriment a prototyping board design) and run the other pins of the receiver and transmitter out to through-hole pins. These through-hole pins will be connected to the prototyping area of the PICDEM.net 2, along with decoupling capacitors and other necessary circuitry such as voltage dropping resisters.

The RF receiver and transmitter circuitry at the base station is shown in Figure 9. The part of the circuit that will be mounted on the PCB is indicated by the green square.

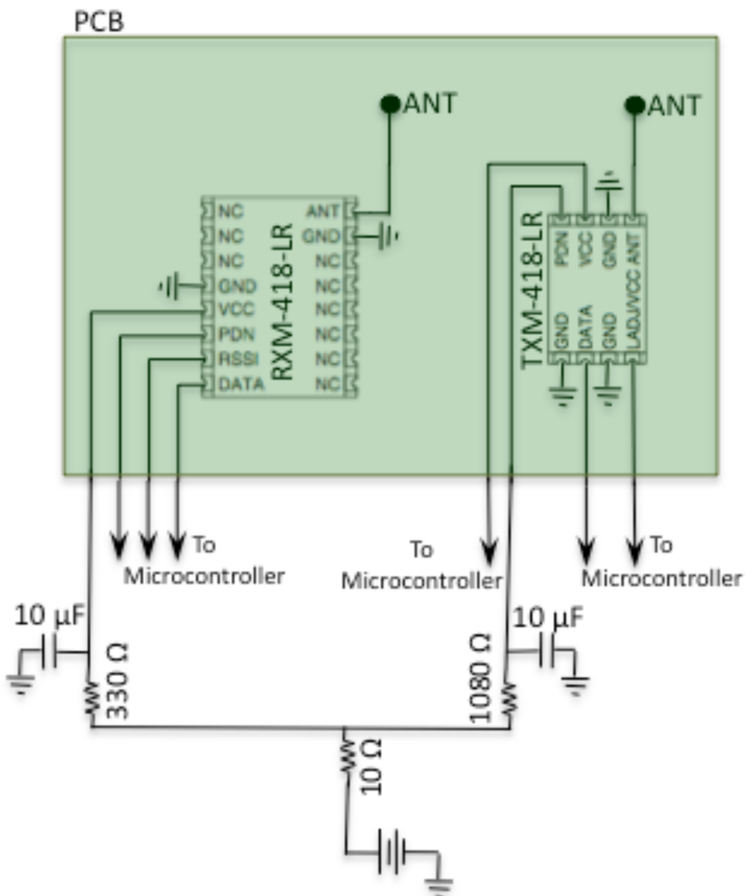

Figure 9: RF circuitry at base station.

4.4.2 Software

The base station software must contact the server (at a fixed IP address for each prototype test, and at a designated domain name for the production model) at boot up and then listen constantly for requests from the server. After it gets a request, it must take the appropriate action (i.e. query the sensors or send a lock/unlock request and then query the sensors) and return information to the server. An overview of the base station's software is presented in Figure 10.

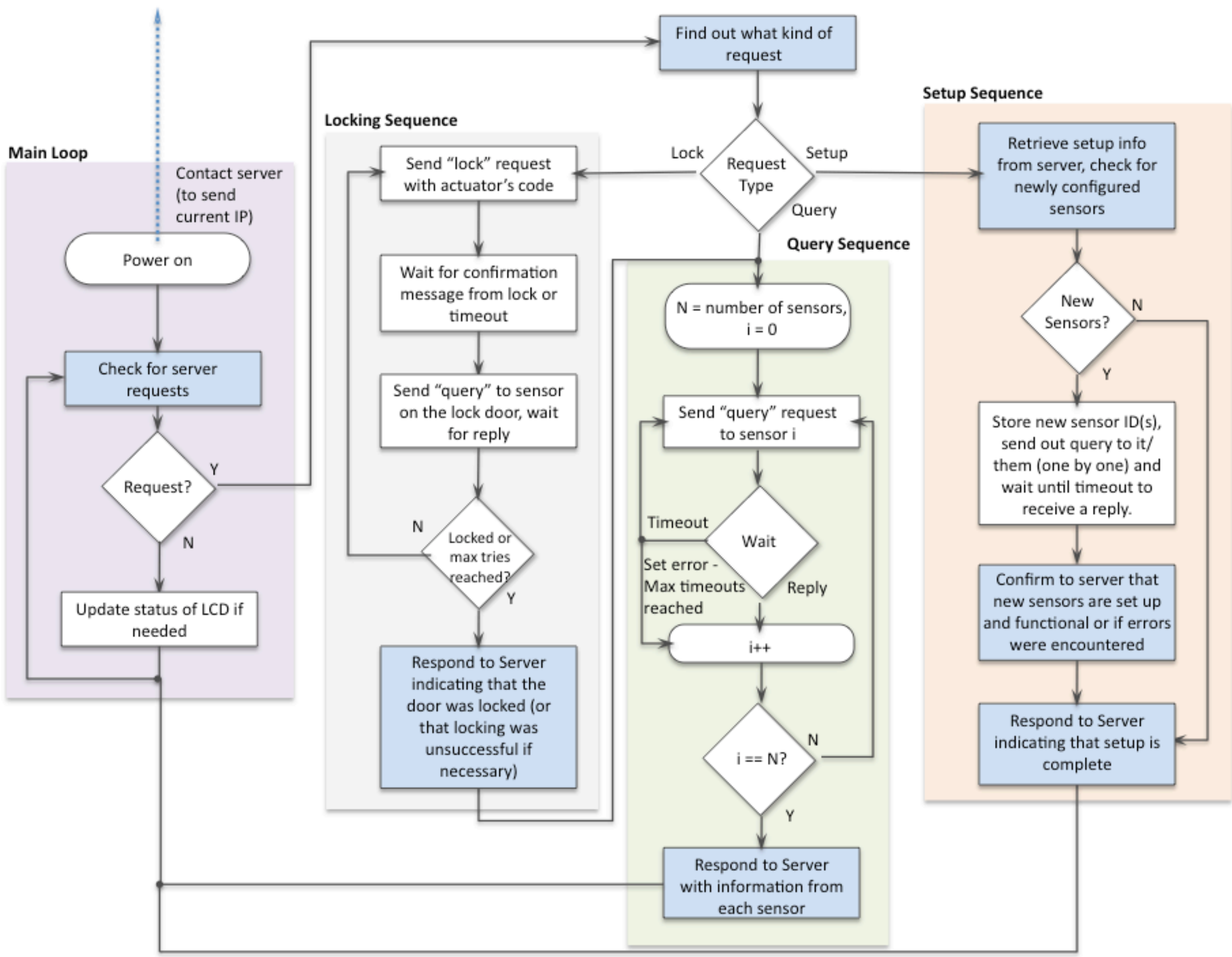

Figure 10: Base station main software outline

The main software loop is indicated by the purple box, the locking sequence is indicated by the grey box, the sensor query sequence is indicated by the green box, and the setup sequence is indicated by the orange box. The blue boxes and shapes indicate a step in which the base station must communicate with the server.

The base station's LCD will be updated if necessary after every loop to show the system's status. If any of the sensors are found to be unresponsive during a query, this will be indicated to the user on the LCD, as well as in the response text message. Additionally, if the base station loses its Internet connection, this will be indicated as an error on the LCD. For the production model, other errors shown on the LCD will include low battery warnings.

The software block diagrams for the base station sending a sensor query and an actuator lock/unlock request are shown in Figure 11. To communicate effectively with the sensors and actuators, the base station must encode the data it sends to the RF transmitter and decode the data that it receives. The format of these encoded messages is discussed in detail in Section 4.5.2. Additionally, the base station must take the receiver and transmitter out of power-down mode before starting communication and return them to power-down mode after communication has completed.

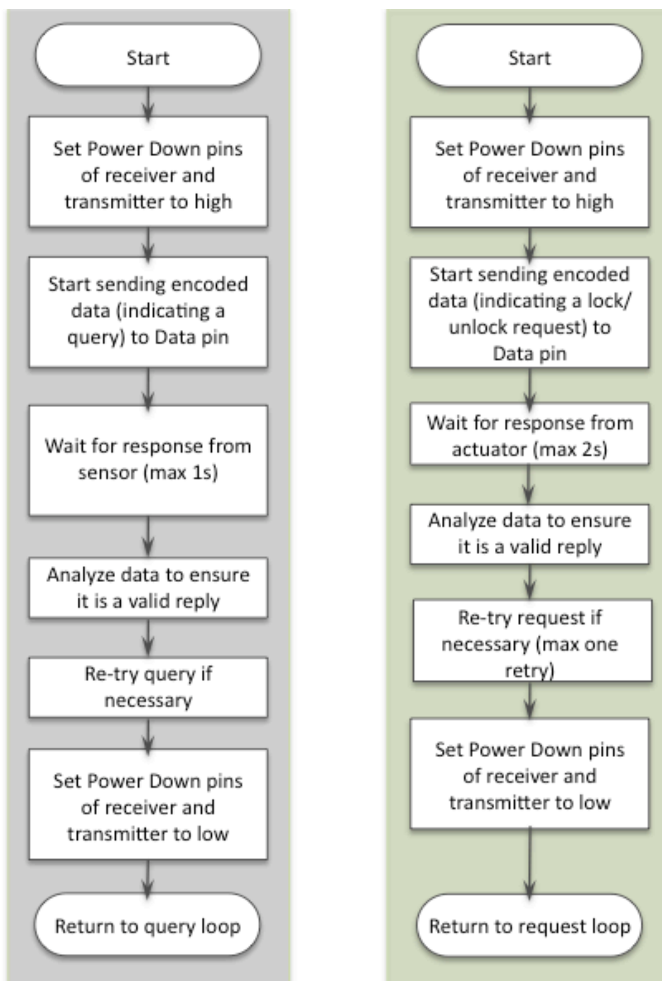

Figure 11: Software block diagram of sensor query and actuator lock/unlock request

4.5 RF Transceivers

The base station and sensor locations will communicate using RF transmitters and receivers, jointly referred to as transceivers. The details of the circuits interfacing the RF transceivers to the base station and sensors are discussed in Section 4.4.1 and Section 4.6 respectively. This section addresses the basic properties of the transceivers and the protocol for transmission.

4.5.1 Transceiver Properties

The Watchbird™ system will use receivers and transmitters from Linx Technologies. As discussed in 4.4.1, the LR series will be used at the base station. When connected to V_{cc} and ground, these modules will allow the microcontroller to transmit data on a single output pin and receive data on a single input pin. The receiver module includes an RSSI pin to allow for calibration in the analog to digital conversion process, and both modules have power down pins to conserve power when not in use.

At the sensor locations, transmissions will be handled by the KH2 series [6], [7]. These modules provide receiver-decoder and encoder-transmitter combinations, which will eliminate the need for a microcontroller at the sensor end. As discussed in detail in Section 4.6, the decoder checks whether a received message is intended for a particular sensor, and if so, decodes data onto individual data pins. The encoder performs the reverse function, taking data from address and message pins and relaying it to the transmitter. The receiver and transmitter portions of these modules are comparable to the LR modules used at the base station. At the base station, the microcontroller will perform the encoding and decoding functions that the KH2 series performs at the sensor end.

The transceivers will all operate at 418 MHz, which has relatively few other devices that would create interference problems [8]. An alternative option had been to give each sensor its own operating frequency; however, this choice was rejected because of the difficulty in having enough distinct frequencies for up to 10 sensors.

Data transmission will occur at a rate of approximately 384 Hz, as dictated by the KH2 encoder-transmitter module. This data rate is sufficient for the small amounts of data that must be transmit ted as well as the system's time constraints. The microcontroller at the base station will therefore send output data to its LR transmitter and process data from its LR receiver at the same rate.

4.5.2 Address and Message Protocol

Given that all the transceivers are operating at the same frequency, there must be a way of distinguishing the different sensors and types of transmissions. The KH2 series described above uses 10 address pins to uniquely identify individual transceivers. The first 10 bits of each received message are compared to the receiver's address, and if they do not match, the rest of the message is ignored. These address pins will be used as described in Table 7 and Table 8.

Table 7: Summary of address pin usage.

Pins	Function
0-1	start of transmission (always 1)
2-3	identification of message origin (1 for base station, 0 for sensor)
4-9	sensor identification (unique combinations)

Table 8: Details of address pin usage. For ease of reading, blank entries replace logical "0".

Message Destination	Pin									
	0	1	2	3	4	5	6	7	8	9
sensor location	1	1	1	1	sensor address					
base station	1	1			sensor address					

The Linx combination receivers are also designed to output eight bits of received data on eight data pins. Similarly, the transmitters have eight input data lines that get converted into 8 bits of the message. The type of message being sent by the base station will be encoded as described in Table 9, and then decoded accordingly at the sensor. The sensor circuitry will maintain the same message it received if the sensor is high, and create the ones' complement of the message if the sensor is low.

At a sensor and actuator location, the system confirms it actuated the lock by replying with the same message it received if the sensor is high, or with the one's complement of the message if the sensor is low. However, in this case the message will simply confirm the lock is actuated, and a separate sensor query is required to ensure an accurate reading. This part protocol is reflected with "x" in Table 9 to indicate that it is the message transmission, not the message content that is significant in this case.

This protocol creates distinct messages, with minimal circuitry, as discussed in the Section 4.6.

Table 9: Details of message pin usage. For ease of reading, blank entries replace logical "0".

Message Origin	Message Type	Pin							
		0	1	2	3	4	5	6	7
base station	check sensor	1	1	1		1	1	1	
sensor location	sensor high	1	1	1		1	1	1	
sensor location	sensor low				1				1
base station	lock door	1	1			1	1		
sensor + actuator location	confirm lock	1	1	x	x	x	x	x	x
base station	unlock door	1	1					1	1
sensor + actuator location	confirm unlock	1	1	x	x	x	x	x	x

4.6 RF Transceiver/Sensor/Actuator Circuitry

The sensor end circuitry (SEC) will be capable of communicating with the base station without the use of a microcontroller. This design choice was made based on the simplicity of the required functions at the sensors, and the availability of receiver-decoder and encoder-transmitter combination modules. A flowchart describing the basic steps for the SEC to receive and respond to a request is shown in Figure 12. The SEC will be essentially the same for a single sensor as for a sensor-actuator pair (SAP), but the SAP requires additional components to determine what type of message it is receiving.

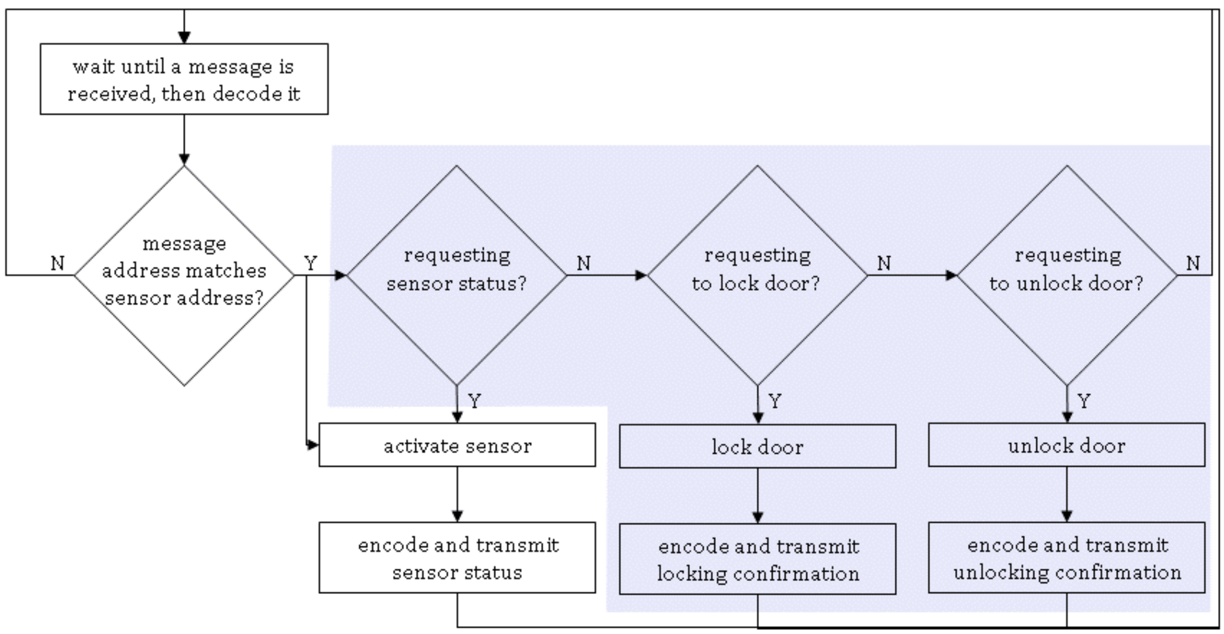

Figure 12: Flowchart of steps required to receive and respond to a base station request. The area shaded in blue is only present at SAP locations.

4.6.1 Single Sensor Circuitry

A schematic of the SEC for a single sensor is shown in Figure 13. Each receiver will be connected to a unique combination of 10 address pins, based on the protocol described in Table 7 and Table 8. The first four pins will always be connected to V_{cc}: the first two simply to define the start of transmission, and the second two to confirm the message has been sent by the base station. The remaining six pins will provide each receiver's specific address. Given that the Watchbird™ system supports a maximum of 10 sensors or actuators, this number of pins provides more than enough binary combinations. If the first 10 bits of the received message match the address pins, the "valid signal" line on the receiver-decoder will go high, and the remaining eight message bits will be outputted onto 8 data pins.

The data pins and valid signal will then feed in to a delay circuit, discussed in further detail in Section 4.6.3. This step is necessary because the transmitter must not begin broadcasting a signal until the receiver is no longer receiving a signal.

The encoder-transmitter will begin the reply using 10 address pins: the first two high indicating the start of a transmission, the second two low to indicate a message sent by a sensor, then the remaining six as defined by the sensor's ID. Following these 10 bits will be the sensor response. The protocol described in Section 4.5.2 will be implemented using an 8-bit XNOR gate, which replicates the received message if the sensor is high, and complements it if the sensor is low. Details of the sensor are provided in Section 4.7.

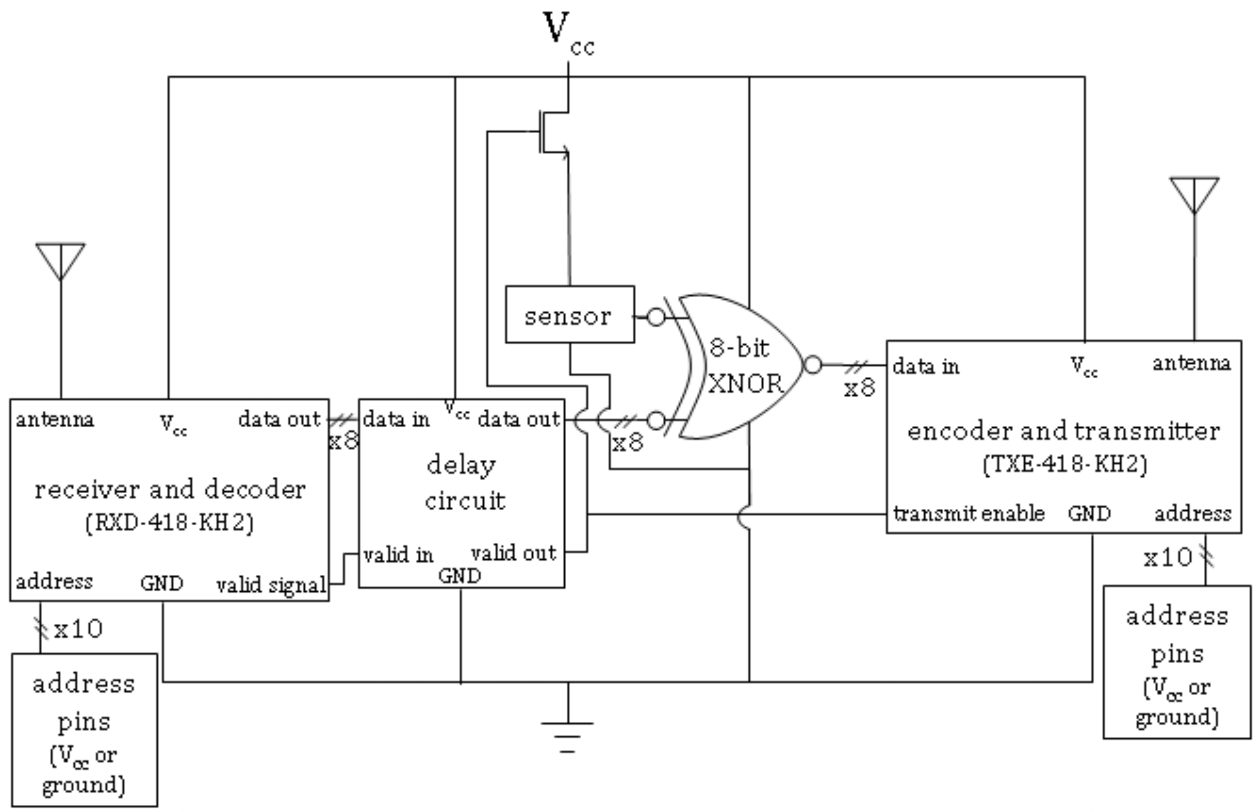

Figure 13: Schematic of the SEC for a single sensor.

4.6.2 Sensor and Actuator Pair Circuitry

The schematic for a SAP shown in Figure 14 is more complex, owing to the need to determine what type of message has been sent. Again, the receiver-decoder will only output data if the incoming address bits match the 10 address pins, and again the output data lines and the valid signal line will be passed through a timing circuit. The 8 data lines will then go to input A of three separate 8-bit comparators to determine if the message was a sensor check, door lock, or door unlock request. Input B of each comparator will be distinct combinations of V_{cc} and ground connections to define the message types listed in Table 9, although these connections are omitted in Figure 14 for clarity.

If the incoming message matches a comparator's message type, its enable line will go high, allowing the appropriate action to be taken. If the message is a sensor query, the circuit will function as in Section 4.6.1 for a single sensor. If the message is a lock or unlock request, the enable line will be passed through a pulse extending circuit and trigger the corresponding line on the H-bridge to provide current in the correct direction for the motor, as illustrated in Figure 15. If it is a lock request, $S_{lock\ 1}$ and $S_{lock\ 2}$ will be closed; and if it is an unlock request, $S_{unlock\ 1}$ and $S_{unlock\ 2}$ will be closed. Further details of this locking mechanism are discussed in Section 4.8. The pulse extending circuit (discussed further in Section 4.6.3) is required to extend the duration of the signal: the received message duration will be a fraction of a second, but the motor must be actuated for a full second to turn the lock sufficiently.

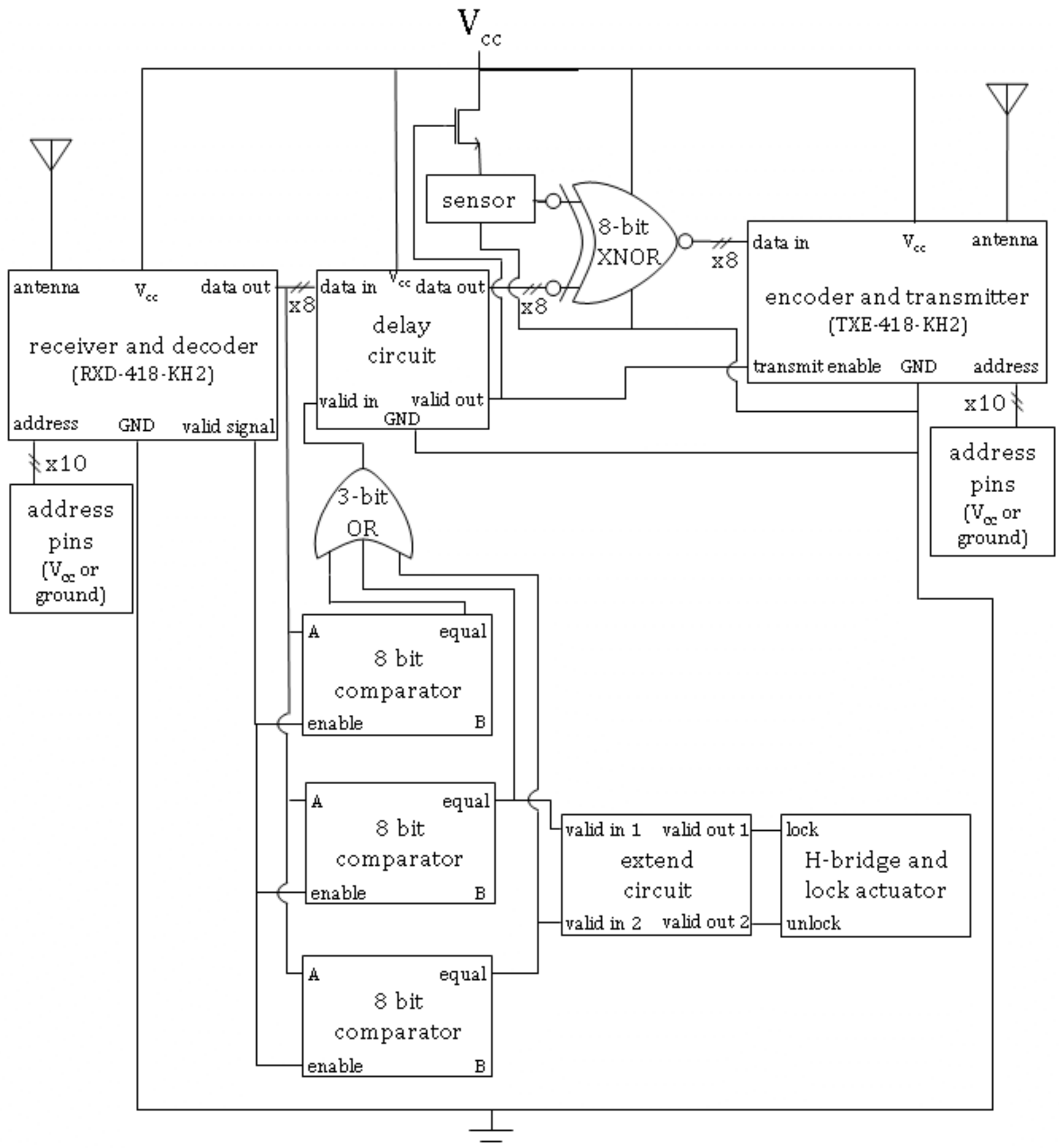

Figure 14: Schematic of the SEC for a SAP. Input B's to the comparators are connected to the combination of ground and V_{cc} that will give the correct message code for comparison, but are not shown for clarity. V_{cc} and ground connections for the comparators, extend circuit, and lock are also omitted for clarity.

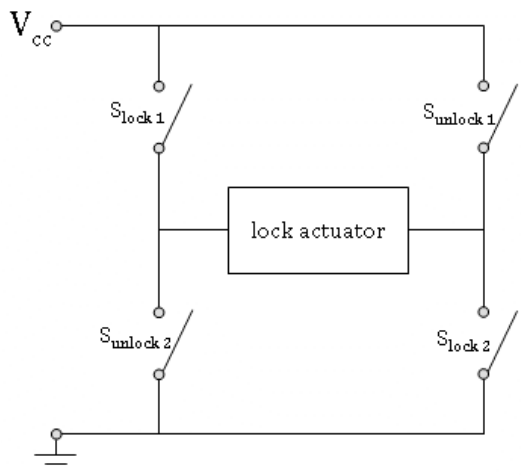

Figure 15: H-bridge circuit used for RF controlled locking and unlocking.

4.6.3 Timing Circuitry

As mentioned above, timing elements are required for two purposes: coordinating received and transmitted messages, and enabling the actuator long enough to turn the lock fully. Both functions will be implemented using 555 timer circuits in a monostable ("one-shot") configuration. The basic concept of this circuit is shown in Figure 16.

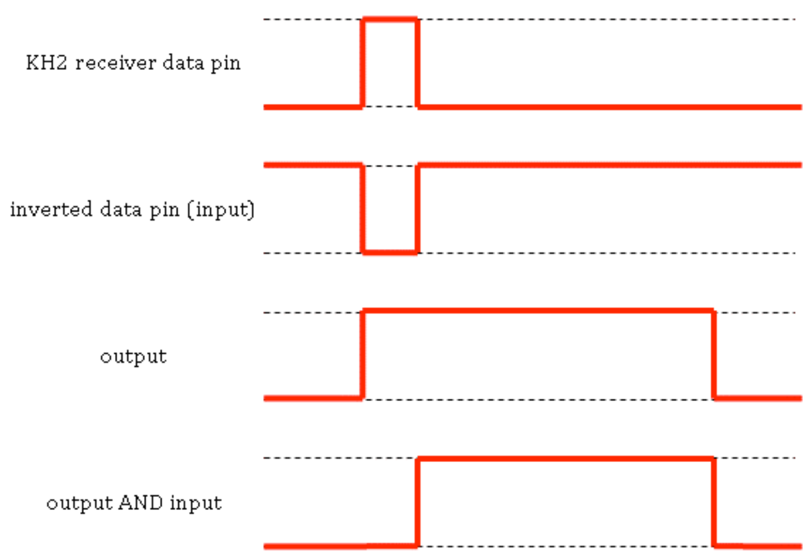

Figure 16: Concept of the 555 timer circuit in monostable mode.

The delay circuit shown in Figure 13 and Figure 14 needs to delay the received data signals until the receiver is finished receiving. This delay will be implemented by connecting a mono-stable timer [9] (also referred to as a "one-shot" timer) to the valid signal line and each data line. The circuit will be constructed using two NTE 925s, which consist of four 555 timers [10], in addition to a single 555 timer, as shown in Figure 17. Each timer circuit must have its own resistor and capacitor to provide the correct time constant for the output pulse. The RC time constant for the valid signal line will be slightly shorter than the time constants for the data lines to account for any mismatches in the remaining resistors and capacitors. In this fashion, incorrect data will not accidentally be transmitted, because the transmit enable line will always turn off before any of the data lines do.

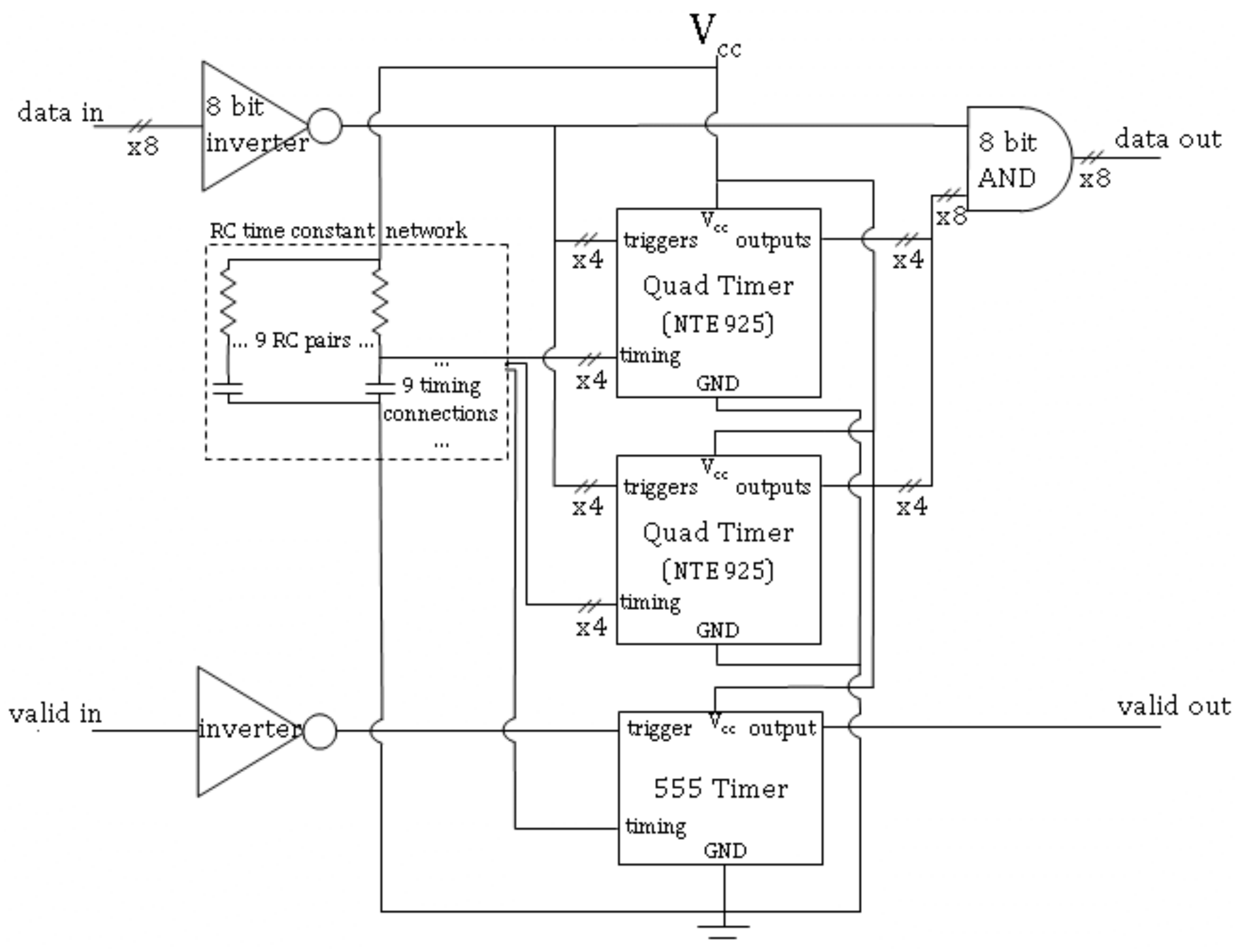

Figure 17: Delay circuit in the SEC. V_{cc} and ground connections for the inverters are omitted for clarity.

The extend circuit shown for the lock actuator in Figure 14 will be similar to the delay circuit, but will have a longer RC time constant to keep the enable signal high for long enough to actuate the lock. Its schematic is shown in Figure 18.

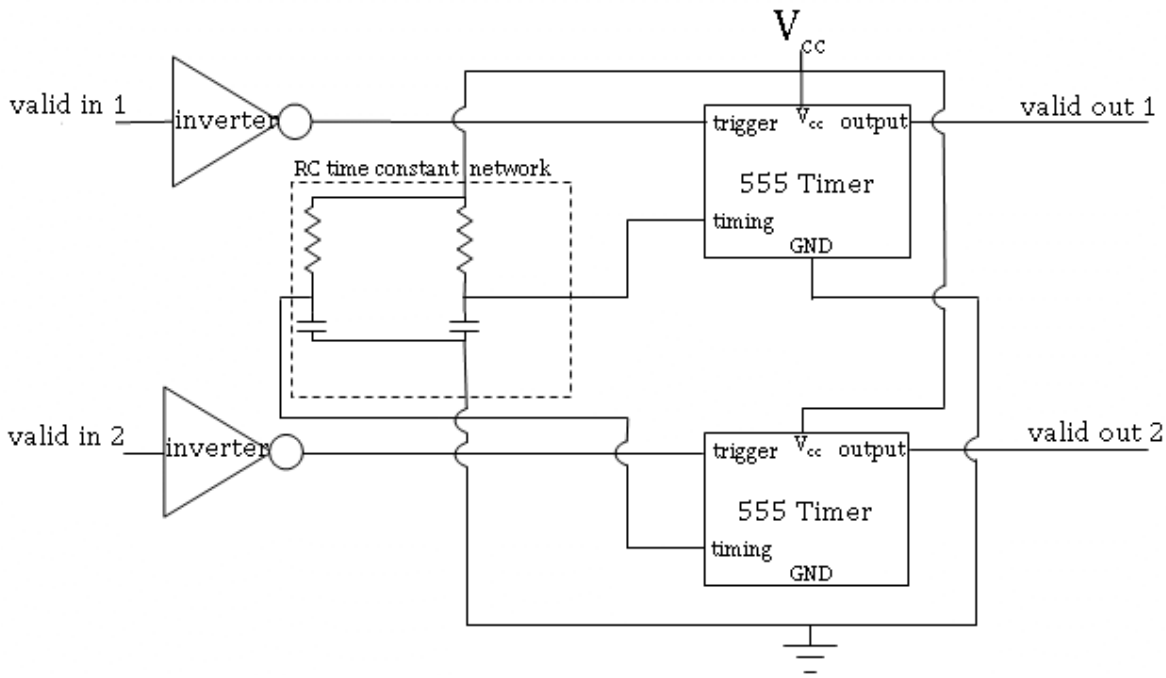

Figure 18: Extend circuit in the SEC. V_{cc} and ground connections for the inverters are omitted for clarity.

4.7 Sensor Modules

Several requirements guide the design of the sensor modules. Firstly, the sensor must be able to fit into the space beside the door deadbolts. Furthermore, the sensors only need to distinguish between two states, such as locked and unlocked for doors and windows, or on and off for switches and stoves. In addition, the sensor and transceiver packages operate on battery packs and should be small, light, and low-impact. The bi-state nature of the switch as well as the power consumption and size requirements suggest that a single-pole, double-throw (SPDT) contact switch with level would best suit the Watchbird™ requirements. The sensor module circuit will consist of both the SPDT switch and a switch-debouncing flipflop.

4.7.1 Single Pole, Double Throw Contact Switch

The bi-state switch from the Omron SS5 series [11], as shown in Figure 19, has a small contact resistance of 50 mΩ and current rating of 5 A for 8 V to 14 V while carrying a resistive load. These characteristics allow the switch to perform its duty within the operating requirement of the battery powered transceiver and sensor combination. This physical switch will be integrated into the overall SEC as shown in Figure 13 and Figure 14.

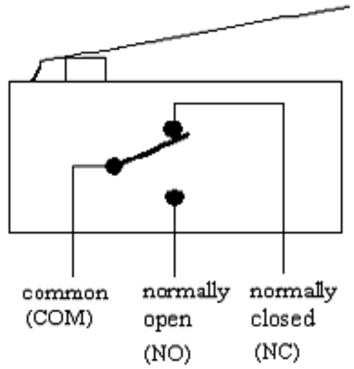

Figure 19: Schematic of SPDT switch.

4.7.2 Switch Debouncing With Flip-Flop

Switch debouncing was initially not included under the assumption that the sensor status requests would rarely happen at the time the switch position was changed, and so the switch would have reached a stable position at the time of status request. However, this assumption cannot hold true under all cases, and therefore debouncing is mandatory to ensure that the sensor reading is correct at all times.

Many methods of switch debouncing have been explored, and the flip-flop has been selected for its many advantages. Previously, the classic capacitor debouncing method was investigated, however this design requires a charging time before the capacitor can reach the required voltage. Because the capacitor only charges when the sensor module is enabled by the "valid signal" line, the duration of the transmission cycle is increased and becomes undesirable. Furthermore, the capacitor (a passive component) can be damaged after a number of sudden step voltage changes. The solution is to use active NAND gates; these components have good response time and have safety circuits built in to deal with sudden voltage changes. This type of debouncing circuit is shown in Figure 20.

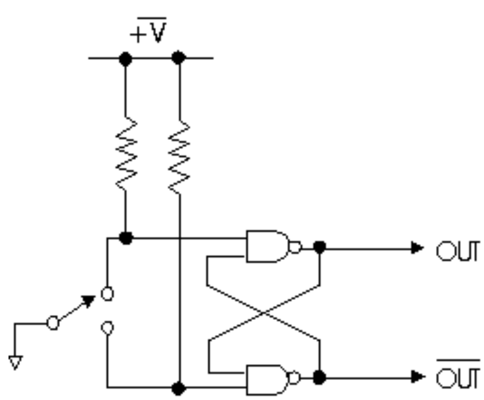

Figure 20: Schematic for the overall sensor module [12].

4.8 Lock Actuator

The door lock actuator for the prototype Watchbird™ will be based on a commercial keypad door lock. This door lock was chosen for our design because it provides an actuator already interfaced to a deadbolt lock and keypad. The actuator in the commercial keypad lock is controlled with a microcontroller, with the keypad connected to the microcontroller input.

When the user locks the door with the keypad, a positive 300 mA current with a potential of 6 V is sent to the actuator. When the user unlocks the door with the keypad, a negative 300 mA current with a potential of -6 V is sent to the actuator. When the lock is in an idle state, the two wires running from the microcontroller circuit to the actuator are shorted.

Our design will insert a circuit between the microcontroller and the actuator, which will allow the RF section at the sensors to control the actuator. It will send current to lock and unlock the deadbolt, and a MOSFET (Metal Oxide Semiconductor Field Effect Transistor) switch system will also be implemented so that the current sent by the RF section does not short through the microcontroller side of the circuit. This design is shown in Figure 21.

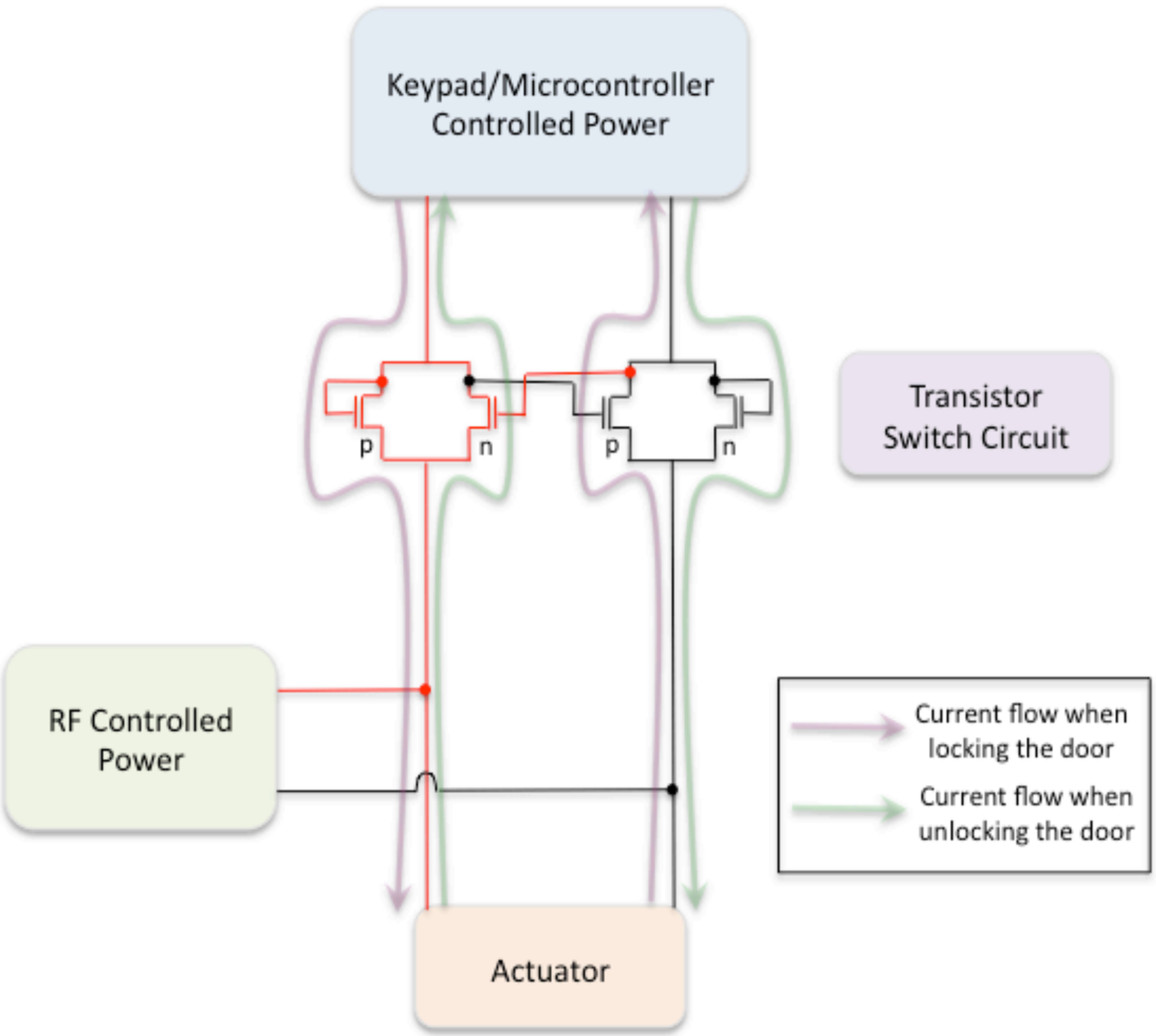

Figure 21: Door lock actuator circuitry

Power MOSFETs will be used in this circuit, so that the circuitry is capable of withstanding the large current the MOSFETs must pass to the actuator. The chosen MOSFETs are the AOP607 dual power MOSFETs [13]. Each transistor contains one n-channel and one p-channel MOSFET, with a maximum drain-source voltage (V_{DS})of 60 V and a maximum drain current (I_D) of 3.4A. The transistor switch circuit will allow the actuator to be controlled by either the keypad or the RF circuit at the sensor. It provides two parallel paths for current on each wire, which cannot both be active at the same time. The switches allow current to flow from the microcontroller to the actuator only when the microcontroller is providing positive current on either of the wires. The purple and green arrows on Figure 21 indicate current flow when the door is being locked and unlocked, respectively, and the small "n" or "p" beside each MOSFET indicates whether it is n- or p-channel.

A similar switching circuit to the one shown above will be implemented in the RF section to ensure that the keypad-controlled circuit maintains sufficient power to lock and unlock the door. As previously shown in Figure 15, the SEC will use an H-bridge to actuate the lock for remote requests. When the RF section is not active, all switches in the H-bridge will remain open, and locking from the keypad will be unaffected by the RF section's circuitry.

4.9 Setup and User Interface

4.9.1 User Interface

The user interface is the main tool for the user to configure their Watchbird™ system. To make the process easy for the user, the user interface is completely web based, thus allowing the user access to the configuration menu through the company website. The user interface will deal with logins and information editing (and registration for production level), and the user information will be stored in a database on the company server.

The user interface will be implemented using Microsoft Visual Web Developer Express and Microsoft SQL Server Express, and the web site itself will be designed using ASP.net, SQL, and C#. The user interface website and the database will both be hosted on the company server. The user interface can communicate with the database directly, and is independent of the operation of the server program, as shown in Figure 22. This separation ensures that the main server program can operate in a task-based manner, focusing on individual tasks in its queue from user cell phones and the web interface without handling the web user interface itself.

Figure 22: Relationship of server program, database, and user interface

4.9.1.1 Login Menu

This menu is the gateway to the Watchbird™ setup. The users enter their user ID and password to access their respective Watchbird™ menus. This is an important step that limits the users to accessing only their own Watchbird™ system.

Table 10: Login menu components

Instance Name	Function
User Name	Text field for users to enter their own user ID.
Password	Text field for users to enter their own passwords, each character is masked.
Login button	Submits user name and password for authentication

4.9.1.2 User Menu

When the users pass the authentication process, the user menu is loaded. This is the menu where users will view the current config uration for their Watchbird™ and their member settings for logins. Users will also be able to change their preference options from this menu. In addition, users will be able to view their product ID, the list of their Watchbird™'s sensors, and other operation details of their Watchbird™ in the user menu,

The user menu is split into three sections, User Information, Sensor Information, and Action Options, described in Table 11, Table 12, and Table 13, respectively.

Table 11: User information

Instance Name	Display	Edit
Account Password	Text field with masked current password	Changes account password
Cell Phone Numbers	Text fields with the phone numbers that are used to communicate with Watchbird™	Changes and/or adds phone numbers
Text Message Password	Text field with masked current password	Changes text messaging password
Product Serial Number	Text with the distinct Watchbird™ product serial number which is used to communicate with each user's Watchbird™	None

Table 12: Sensor information

Instance Name	Display	Edit
Sensor Designation	Display the sensor number (sensor 1-10)	None
Sensor Presence	Yes/No to indicate the existence of the sensor	None
Actuator Presence	Yes/No to indicate if the sensor unit includes an actuator	None
Sensor Name	Text field with the name of the sensor which is used in the cell phone text messages	Change the name of the sensor

Table 13: Action options

Instance Name	Function
Add Sensor Button	Upon button press, brings up a prompt to enter the new sensor number
Save Button	Upon button press, every change administered by the user will be written to the database, and the User Menu page will be refreshed to display current user configurations.
Remove Sensor Button	Upon button press, brings up a prompt asking the user to confirm removal of the sensor from their system. "Yes" will remove the sensor, "No" will return to the User Menu page.

5 System Test Plan

To ensure that the above design specifications are met, Chickadee Tech has devised the following system test plan.

5.1 Server

- Test to ensure that the server can receive and reply to text message requests.
    - Send a text message request.
    - Confirm that a text message reply is received.
- Test to ensure that a request is rejected if the source cell-phone number is not valid.
    - Send a text message query from a cell phone number that isn't in the list of accepted numbers.
- Test to ensure that a request is rejected if the password is not included.
    - Send a text message query that doesn't include a password.
    - Confirm that a response is sent telling the user that their password was incorrect.
- Test to ensure that a request is rejected if the password is incorrect.
    - Send a text message query with an incorrect password.
    - Confirm that a response is sent telling the user their password was incorrect.
- Test to ensure that the server will detect when base station is unreachable.
    - Disconnect base station, send a text message query, and check for correct error message from server.
- Test to ensure that the server will detect when base station fails to respond before timing out.
    - Send a text message query, have the base station not respond, and check for correct error message from server after the correct amount of time has elapsed.

5.2 Base Station

- Test to ensure that the base station can send digital data to the Linx LR transmitter of the correct frequency.
    - Program the microcontroller to send out digital data of the correct frequency to one of its output pins.
    - Monitor the output pin with an oscilloscope.
- Test to ensure that the Linx LR Transmitter can transmit data given to it by the base station.
    - Connect the LR transmitter to the base station.
    - Program the microcontroller to send out digital data of the correct frequency to the output pin connected to the LR transmitter.
    - Use the LR receiver connected to the lab power supply and an oscilloscope to ensure that it receives the correct data.
- Test to ensure that the base station can receive data from the Linx LR Receiver.
    - Connect the LR transmitter and receiver to the base station.

  - Program the microcontroller to send out digital data of the correct frequency to the output pin connected to the LR transmitter and read in data from its analog input (through its A to D).
  - Use debugging statements to the LCD to confirm that the base station is receiving the correct data.
- Test to ensure that the base station will send its serial number to the server on power up.
  - Disconnect and reconnect power to the base station.
  - Confirm at the server that a message has been received.
- Test to ensure that the base station and server can communicate using the designated message format.
  - Have the server initiate a request, and the server immediately reply (without contacting sensors/actuators).
  - Confirm at the server end that the reply was correct.
- Test to ensure that an error message is displayed on the LCD screen if the unit is not connected to the Internet.
  - Unplug the unit's Ethernet connection and watch the LCD screen.
- Test to ensure that an error message is displayed on the LCD screen if the unit cannot contact one of the set-up sensors/actuators.
  - Remove the batteries from one of the set-up sensor/actuator units and watch the LCD screen.
- Test to ensure that the timing requirements of the system are met and that base station/server connectivity is fully functional.
  - From the server, record the base station's response time over multiple instances of several tests:
    - Query with all sensors connected properly.
    - Query with one or more sensors disconnected.
    - Lock/Unlock request with all sensors and actuator connected properly.
    - Lock/Unlock request with all sensors connected properly and actuator disconnected.
    - Lock/Unlock request with one or more sensors disconnected and actuator connected properly.
- Test to ensure that the Analog to Digital converter is properly distinguishing between highs and lows.
  - Ensure that the response message to queries is correct while varying the range of RF transmission and the obstacles in the path of the transmission.

5.3 Lock Actuator

- Test to affirm locking and unlocking functionality via key, keypad, and RF transmission.
  - Lock and unlock the unit with the key, keypad, and RF transmission.
- Test to ensure that the lock and actuator are protected when the actuator tries to open the lock while already open or close it while already closed.
  - Send lock request when lock already locked.
  - Send unlock request when lock already unlocked.

5.4 Sensor Module

- Test to ensure the sensor can detect position correctly
  - Attach the sensor module to various windows, doors, and stoves and compare detection results.
  - Measure the output voltage of the sensor module during position change to ensure the debouncing mechanism is working properly.
- Test to ensure compatibility with the transceiver circuitry
  - Find a working input and output range at which the sensor module can operate and compare with the transceiver before fitting the sensor module with transceiver.

5.5 RF Transceivers

- Test communication between LR transmitter and KH2 receiver.
  - Set up the LR transmitter to send a TTY signal at 2 kHz, using a function generator and power supply (not batteries at this stage).
  - Set up the KH2 receiver on the far side of the lab, with address pins in the "don't care" condition, using a power supply instead of batteries.
  - Connect an oscilloscope to the data pin, and observe received waveform prior to decoder.
  - Repeat test using battery packs.
- Test communication between KH2 transmitter and LR receiver.
  - Set up KH2 transmitter using the power supply: place address pins in the "don't care" condition, connect even numbered data inputs to V_{cc}, and odd numbered data pins to ground.
  - Set up LR receiver using the power supply, and observe the output waveform on an oscilloscope.
  - Repeat test using battery packs.
- Test address pin functionality.
  - Set up KH2 transmitter using the power supply: place address pins in a 1100011000 configuration, connect even numbered data inputs to V_{cc}, and odd numbered data pins to ground.
  - Set up KH2 receiver using the power supply: place address pins in a 1100011000 configuration.
  - Connect an oscilloscope to the data pin, and observe received waveform prior to decoder.
  - Now change one address pin at the receiver. No output waveform should be observed.
  - Test various combinations of address pins for consistency.
  - Repeat tests using battery packs.
- Test message pin functionality.
  - Set up KH2 transmitter using the power supply: place address pins in a 1100011000 configuration, connect even numbered data inputs to V_{cc}, and odd numbered data pins to ground.

- Set up KH2 receiver using the power supply: place address pins in a 1100011000 configuration.
- Connect an oscilloscope to the data pin and observe received waveform prior to decoder.
- Measure the voltage on each individual output data pin, and check it matches the transmitter configuration.
- Test various message pin combinations.
- Repeat tests using battery packs.

5.6 Sensor End Circuitry

- Test individual 555 timer circuit.
    - Connect to power supply, attach input and output locations to oscilloscope; set oscilloscope to capture and hold a short waveform pulse.
    - Momentarily provide a high input; confirm on oscilloscope that output has remained high for desired length.
    - Test with appropriate RC values for both sensor relay, and locking/unlocking functionality.
    - Repeat tests using battery pack.
- Test comparators in SAP circuit.
    - Connect to power supply.
    - Apply message codes for sensor query, lock, and unlock requests to each comparator.
    - Ensure that only the correct comparator's enable line goes high.
    - Repeat tests using battery pack.
- Test entire sensor only circuit except for receiver and transmitter.
    - Connect to power supply. In place of receiver data pins, connect pins to V_{cc} or ground to emulate a message.
    - Ensure contact switch is open, and confirm that message is inverted at pins that will connect to transmitter.
    - Now ensure contact switch is closed, and confirm that message is not inverted at pins that will connect to transmitter.
    - Repeat tests using battery pack.
- Test entire sensor and actuator circuit except for receiver and transmitter.
    - Connect to power supply. In place of receiver data pins, connect pins to V_{cc} or ground to emulate a query sensor message.
    - Ensure contact switch is open, and confirm that message is inverted at pins that will connect to transmitter.
    - Now ensure contact switch is closed, and confirm that message is not inverted at pins that will connect to transmitter.
    - Adjust message type for a lock request, and ensure the lock is actuated.
    - Adjust message type for an unlock request, and ensure the lock is actuated.
    - Repeat tests using battery pack.
- Test entire circuit using receiver and transmitter.

  - Repeat above test, but use actual incoming messages as sent by the LR transmitter, and confirm LR receiver gets correct response.

5.7 User Interface

- Test the interaction with the server
  - Send an add sensor request and verify that the server program monitors the database and performs required duties.
- Test the interaction with the database
  - Check if user configuration changes will be reflected in cell phone status messages
  - When the server applies changes in the database, the web interface can update the changes accordingly.
- Test the capability of the user interface
  - Unauthorized users will not be able to access user configuration menu without valid log in information
  - Information changes will not apply if the format does not comply with allowed format.

5.8 Overall System

- Set up the system from scratch as a new user would.
  - Provide the base station with power and an Ethernet connection. Turn power on, and confirm that server registers a new base station.
  - Install door lock SAP, and stove dial sensor. Initialize new sensors using the web-based UI. Set up two allowed cell phone numbers and passwords. Confirm at the server that these parameters now exist.
- Test typical queries and requests.
  - Test each sensor in all possible situations (on/off or open/closed, as well as no battery), with a query for each.
  - Send a lock request, and ensure the lock is actuated.
  - Send an unlock request, and ensure the lock is actuated.
  - Send text messages from two different phones as simultaneously as possible, and ensure they are both dealt with.
- Install a new sensor.
  - Install a window sensor, and initialize it using the web-based UI. Confirm at the server that the sensor has been added without affecting previously existing information.
  - Query the new sensor via cell phone.
- Remove an existing sensor.
  - Use the web-based UI to remove the stove dial sensor. Confirm at the server that the sensor has been removed, and that no other information has been altered.
  - Query the remaining sensors via cell phone.

6 Conclusion

This document provides the detailed design specification for the prototype of the Watchbird™ system, and it will be followed closely by the Chickdee Tech design team to ensure that the prototype complies with the specification. The proof-of-concept prototype fulfilling the requirements listed in the functional specifications and designed according to this document will be fully tested and functional by April 2009.

7 References

[1] 3M, "3M Canada: Command™ Home", 2009. [Online]. Available: http://solutions.3m.com/wps/portal/3M/en_CA/CA-Command/Home/. [Accessed: March 4, 2009].

[2] D.F. Tweney, "How to send email to a phone," *SMS 411*, July 2, 2008. [Online]. Available: http://www.sms411.net/2006/07/how-to-send-email-to-phone.html. [Accessed: February 20, 2009].

[3] Microchip Technology, Inc. "PICDEM.net 2 Internet/Ethernet Development Board User's Guide", 2007. [Online]. Available: http://www.microchip.com/Microchip.WWW.SecureSoftwareList/secsoftwaredownload.aspx?device=en028217&lang=en&ReturnURL=http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1406&dDocName=en028217#. [Accessed: March 1, 2009].

[4] Linx Technologies, Inc. "LR Series RF Transmitter Module Data Guide", 2008. [Online]. Available: http://www.linxtechnologies.com/Documents/TXM-xxx-LR_Data_Guide.pdf. [Accessed: March 1, 2009].

[5] Linx Technologies, Inc. "Does Linx have sockets for the surface mount modules so that they can be mounted onto a protoboard without being soldered?", 2003. [Online]. Available: http://www.linxtechnologies.com/Support/Knowledgebase/RF-Modules/?Article=Does+Linx+have+sockets+for+the+surface+mount+modules+so+that+they+can+be+mounted+onto+a+protoboard+without+being+soldered%3F. [Accessed: March 1, 2009].

[6] Linx Technologies, Inc. "KH2 Series - RF Transmitter Module with Integrated Encoder Chip Data Guide", 2008. [Online]. Available: http://www.linxtechnologies.com/Documents/TXE-xxx-KH2_Data_Guide.pdf. [Accessed: February 10, 2009].

[7] Linx Technologies, Inc. "KH2 Series - RF Receiver Module with Integrated Decoder Chip Data Guide", 2008. [Online]. Available: http://www.linxtechnologies.com/Documents/RXD-xxx-KH2_Data_Guide.pdf. [Accessed: February 10, 2009].

[8] Linx Technologies, Inc. "Which is the best frequency to use for my application: 315, 418, 433.92MHz, or 900MHz?", *Linx Knowledgebase*, July 14, 2003. [Online]. Available: http://www.linxtechnologies.com/Support/Knowledgebase/RF-Modules/?Article=Which+is+the+best+frequency+to+use+for+my+application%2C+315%2C+418%2C+433.92MHz%2C+or+900MHz%3F. [Accessed: February 18, 2009].

[9] A.S. Sedra and K.C Smith, *Microelectronic Circuits*, 5th ed. New York, NY: Oxford University Press, Inc., 2004.

[10] NTE Electronics, Inc., "NTE926 Integrated Circuit Quad Timer," *Timing Circuits*, publication date unknown. [Online]. Available: http://www.nteinc.com/specs/900to999/pdf/nte926.pdf. [Accessed: March 1, 2009].

[11] Omron Electronic Components LLC., "Subminiature Snap Action Switch," January, 2007[Online]. Available: http://www.components.omron.com/components/web/pdflib.nsf/0/80FC48E7A8A557A885257201007DD493/$file/SS_0607.pdf. [Accessed: February 28, 2009].

[12] Electronix Express A Division of R.S.R. Electronics, Inc., "Contact Bounce and Debouncing," *Tips from Our Tech Department*, publication date unknown. [Online]. *Available:* http://www.elexp.com/t_bounc.htm. [Accessed: March. 1, 2009].

[13] Alpha and Omega Semiconductor, "AOP607 Complementary Enhancement Mode Field Effect Transistor", 2008. [Online]. Available: http://www.aosmd.com/pdfs/datasheet/AOP607.pdf'. [Accessed: February 27, 2009].