Luggage Link

Project Documentation

Senior Design I

Group C
David Farrell
Evan Husk
Jose Mousadi

April 23rd, 2012

Table of Contents

Tables

Figures

1 Executive Summary

This project is a Global Positioning Satellite (GPS) based luggage tracking system designed to provide the capability to track one or more pieces of luggage anywhere in the world. The core components of this project consist of Luggage Tracker Units (LTUs) operating in unison with a software package that is based around a central GPS server. Each individual piece of luggage requires a corresponding Luggage Tracker Unit designated to that specific luggage item. In order to track multiple pieces of luggage, multiple LTUs will be necessary. These tracking units will receive GPS signals and transmit their current location to the user via cellular network. Users will be able to log into a web portal and view the location of any or all of their Luggage Tracker Units on a map, with their positions being updated in real time.

The main objective of this project is to demonstrate tracking capability for a single Luggage Tracker Unit. The most basic method in which this can be accomplished is to send the unit's GPS coordinates to a user's phone via text message. Once this capability is achieved, more advanced features and communication methods will be integrated into the design. It should be noted that the additional features will seek to expand upon the basic functionality of the system. That is, the basic functionality of the system will not be significantly changed. The additional features will simply increase the system's ease of use.

The majority of these additional features will be implemented in software. While the final hardware product will remain relatively unchanged from the initial prototype, the final software product will feature a GPS server, a web interface, and an Apple operating system (iOS) application, all of which will work in harmony to provide an efficient, user-friendly way of tracking multiple pieces of luggage. To ensure cross-platform functionality of the system, the web interface will remain simplified so that any system, including portable devices, can support its functionality. Because any device capable of supporting iOS is also capable of supporting any iOS application, the iOS application will include a relatively elegant design and layout to incorporate the same functionality offered by the web interface.

The technical approach being taken is one in which components and features are implemented with two factors in mind: necessity and risk. In the initial prototype, only the essential components needed for basic functionality will be designed and implemented. Once this prototype has been developed and basic tracking functionality has been achieved, additional features and components will be incorporated. These features are not necessary to the basic functionality of the product, but serve to improve its overall functionality and design in various ways. By executing this particular approach, our goal is to eliminate any unnecessary risks that would jeopardize completion of the project, and at the same time, to mitigate the risk of any components which are essential to the project.

2 Project Description

2.1 Statement of Motivation

According to the March 2012 Air Travel Consumer Report published by the U.S. Department of Transportation, in January of 2012 there were 139,118 reports filed for mishandled luggage in the United States. While this statistic has decreased slightly since January of 2011, with a total of 181,608 reports filed, the problem of lost, delayed, or otherwise mishandled luggage is a significant one.

It should be noted that “mishandled luggage” includes not just reports of lost luggage, but also reports of damaged luggage, missing items from luggage, and the like. However, it should also be noted that this statistic only represents claims of mishandled luggage for domestic flights within the United States. Thus, these numbers represent only a fraction of total number of incidents occurring each month on international flights from the U.S., as well as the countless other flights happening worldwide around the clock.

The figure below provides a more comprehensive perspective on the issue of mishandled baggage with a comparison of annual statistics from 1990 to 2010. As mentioned previously, even though the numbers do seem to be on the decline, they are still exorbitant and unacceptable given the resources and technology available today to correct such problems.

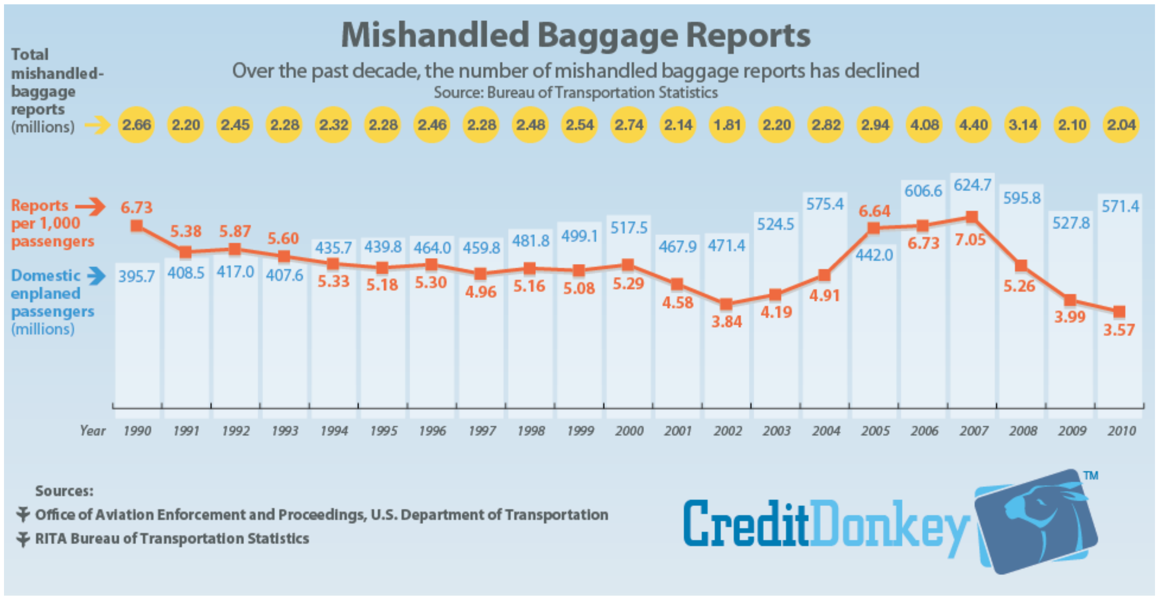


Figure 2-1 Mishandled Baggage Reports
Used with permission from CreditDonkey.com

The inability to fully rely on companies to process and handle luggage has become a significant problem, as well as a major inconvenience, for travelers.

The motivation for this project is to help solve the problem of lost luggage and set those who use the system at ease by providing a simple, reliable way for travelers to locate and track their luggage at anytime, anywhere in the world.

2.2 Goals and Objectives

2.2.1 Top Level Goals and Objectives

The Luggage Tracker Unit should be able to support the tracking of multiple pieces of luggage reliably, at any time and location. At certain time intervals, status updates and notifications will be sent via text message and/or email. These notifications will contain the information necessary to properly identify the location and current status of the item. This information will include the current coordinates of the item, its velocity, and any other relevant information, so the user can easily identify the location of their luggage.

With the help of cellular networks, a reliable connection can be established and maintained so that the correct coordinates will be received in a timely manner. That is, the user will not be receiving invalid coordinates and information based on the item's previous locations. Also, the coordinates will provide a sufficient degree of accuracy so the article's position can be identified even when inside of the airport.

An extremely important feature that is absolutely necessary to provide a reliable system is a robust power supply that can support the basic functionality of the LTU for an extensive period of time. The LTU must be able to maintain its functionality for a number of days to allow any lost or misplaced luggage to be located and retrieved.

2.2.2 Hardware Goals and Objectives

From a hardware perspective, a highly portable, affordable, and reliable LTU must be designed which provides an accurate position, current velocity, and can send Short Message Service (SMS) text messages, as well as emails, to the specified user(s). In order to properly design an efficient and accurate unit, many hardware sub-goals must be met.

Since the small size of the unit is a vital concern, an important hardware design sub-goal is to ensure that all components be placed in the housing in a compact, orderly manner to minimize the overall dimensions of the unit. Another sub-goal is finding and integrating a reliable, yet, somewhat inexpensive GPS module which can be easily integrated into the relatively small, portable encasement. In order to successfully send SMS text messages and emails, a Global System for Mobile Communications (GSM) module must be incorporated in the hardware design. Again, these units should be small and affordable so they can easily integrate into the compact design.

Another important hardware sub-goal is to find and incorporate a power source that can support the multiple modules in the system. It is imperative that the battery is reliable and able to work for extended periods of time in between charges. With the accomplishment of each hardware sub-goal, the main hardware objective will inevitably fall into place.

2.2.3 Software Goals and Objectives

In order to develop a system which supports the functionality discussed in the previous sections, software components must be integrated. The most important piece of software will be the on-chip software, or firmware. The on-chip software must allow all of the components and modules on the chip to exchange information in an efficient manner. It will also direct the functionality of each of the components (sending SMS text messages, emails, etc.). A logging mechanism will also be set in place so times and coordinates can be stored for later retrieval if the cellular connection has been lost and data cannot be transmitted. In order for the user to control the basic settings of the system, a web interface will be made available, and an iOS application will be developed. It should be noted that security features will be implemented in each of these software packages to ensure the user’s identity, location of LTUs, and other sensitive information are protected.

The iOS application will allow for a portable and user-friendly way of interacting with the LTU. Within the app, the user will be able to view and select various preferences. These preferences will allow the user to control the following alert settings:

- Type of alert
- Frequency of alerts
- Destination of alerts
- Factors/conditions that trigger alerts

The web interface will be similar in functionality to the iOS application, but will provide several additional features and capabilities. An additional advantage to implementing a web interface is that it is not limited to users with an Apple iPhone or iPod touch. It can be accessed by anyone with a personal computer or mobile device which has access to the internet.

2.3 Project Function

The luggage tracking system will allow users to implant a small, lightweight LTU and easily track their luggage through multiple user-friendly software interfaces. The LTU will utilize GPS technology to provide real-time locating and tracking of the user’s item(s). The design of the actual LTU will be a highly portable, lightweight device that can be easily placed within any item the user would like to track. It will utilize available GSM networks to send notifications at regular time intervals to keep the user updated on the status and location of their luggage.

There will also be three software interfaces that will allow the user to interact with each LTU. An iOS application will allow the user to track their item(s) from any location via their iOS supporting device. A web portal will allow the user to track their item(s) in much the same way as the iOS application, except the user can access the web portal through any device that is connected to the internet, not just an iOS device. While the desktop software will also provide notifications and information concerning the item(s) being tracked, it will be used more extensively for the purpose of maintaining the LTU.

3 Specifications and Requirements

3.1 Hardware Requirements

3.1.1 PCB

All major internal components shall be mounted to a printed circuit board (PCB). The PCB shall provide all data storage and processing components for the LTU. The physical PCB shall be designed and manufactured to fit within the maximum size requirement of the enclosure listed in section 3.1.9. The PCB shall also provide capability to handle all power regulation and distribution from the LTU's power source. If it is decided that a port will be used to manually reprogram the LTU's settings, the PCB shall provide such a port in an easily accessible location.

3.1.2 GPS Module

The LTU shall contain a GPS module to provide real-time position tracking. The GPS shall provide position information accurate to within +/- 15 meters. The GPS shall provide velocity information accurate to within +/- 0.2 meters per second.

3.1.3 GSM Module

The LTU shall contain a GSM module. Based on settings determined by the user, the GSM module shall transmit SMS text message and email alerts with GPS coordinates and other pertinent information. The GSM module shall contain an externally accessible slot for a subscriber identity module (SIM) card.

3.1.4 Antennae

The LTU shall contain separate antennas for the GPS and GSM components. Both antennas shall provide enough receptivity to operate inside of a multi-storied building. This is for the purpose of ensuring that the LTU will be able to receive GPS satellite and GSM network signals when inside an airport terminal.

3.1.5 Memory

The LTU shall contain a rewriteable form of memory such as Flash, RAM, or EEPROM. This memory component shall be mounted on the PCB and interfaced with the GPS Module, GSM Module, and other data processing components as necessary. The memory shall be used to store the GPS data log, alert settings, and all other relevant data.

3.1.6 Indicators

The LTU shall contain three externally-viewable Light Emitting Diodes (LEDs), one to indicate power, one to indicate GPS status, and one to indicate GSM status. The power indicator LED shall be turned on when the LTU is powered on and remain on until the LTU is powered off. The GPS status LED shall turn on and flash at intervals based on its current mode of operation. The GSM status LED shall turn on and flash at intervals when searching for a GSM network or when trying to register on a GSM network. The LED shall blink at 3-second intervals when the LTU has established a connection with a GSM network. The GSM status LED shall remain in this state until the connection is broken or until the LTU is powered off, at which point the LED will be turned off.

3.1.7 Power

The LTU shall contain a rechargeable battery capable of powering the system for a minimum of 72 hours of normal operation without recharging. A power adapter and charge controller shall be implemented in order to recharge the battery.

3.1.8 External Controls

The LTU shall include a normally-open momentary push button which turns the device on and off. The switch should be resistant to pressure from incidental contact, but easy for the user to depress intentionally. A normally-open momentary push button will also be included as a manual reset button.

3.1.9 Enclosure

A two-piece enclosure shall be utilized to house the LTU. The enclosure shall contain apertures for the Mini Universal Serial Bus (USB) port, power switch, and LED indicators. The maximum dimensions of the enclosure shall be 4” × 4” × 0.5”. The enclosure must have a durable, robust design, yet be easy to open for access to internal components. If possible, the enclosure should be waterproof or at least water resistant in order to protect the internal components.

3.1.10 Power Port

The LTU shall have an external power port into which the power adapter may be plugged in order to recharge the battery. The power port shall be designed in order to provide minimal possibility of damage or entrance of debris or other foreign objects.

3.2 Software Requirements

3.2.1 GPS Server

A GPS server shall be implemented as the central data point between the LTU and the user. It will be configurable with multiple LTUs and will provide the capability to receive and process data from those LTUs. The GPS server will also distribute data to the web portal and iOS application for retrieval and analysis by the user.

3.2.1.1 Alert Control

Alert control software shall be developed to determine what alerts are sent to the user and when they are sent. This software will monitor the incoming GPS data and compare the LTU's location with its intended destination. If conditions are met that qualify an alert to be sent, the appropriate data will be gathered and sent to the GSM module for transmittal.

3.2.2 iOS Application

A smart phone application for iOS platforms shall be developed. The application shall have the following features:

- Login screen for secure use
- Map feature that displays the current location of all LTUs registered to the user
- Alert center with customizable settings for text message and email alerts

3.2.3 Web Portal

A web portal shall be implemented at which users can login to their account and view the current location of any or all of their LTUs. This feature is primarily for users who do not own an Apple iPhone or iPod but have some other device with internet access. The web portal will provide the capability to generate reports and establish geozone boundaries for monitoring the position of LTUs.

3.2.4 On-Chip Software

The on-chip software shall be implemented using the Python programming language. It shall be able to correctly interpret and process information concerned with the GPS server. The basic processes of the LTU's hardware shall be correctly handled by the on-chip software, certifying correct communication between modules.

3.2.4.1 Interface Control

Interface control software shall be implemented to handle the interaction of the various components on the PCB. This software shall include any necessary features such as clock generators, watchdog timers, and commands for sending and receiving data to and from different hardware components.

3.2.4.2 Data Logger

A data logger shall be implemented on the PCB to store events and other usage data. Because a wireless connection cannot be guaranteed, it is necessary to store the events occurring within the LTU for later transmittal to the GPS server and/or directly to the user. Potential events are loss of GSM signal, loss of GPS link, beginning of sleep mode, beginning of active mode, attempt to send SMS message, and attempt to send email message.

3.2.4.3 Power Control

A power control program shall be developed to regulate the power modes of the internal components of the LTU. Algorithms will be developed to minimize power consumption.

3.3 Functional Requirements

3.3.1 Multi-device Tracking

The luggage tracking system shall provide the capability to track two or more LTUs simultaneously. The system shall provide a web-based interface in which users may view the real-time location and status of multiple LTUs.

3.3.2 User Alerts

The luggage tracking system shall provide alerts in the form of SMS text messages and emails for specific events determined by the user, such as entering or exiting a geozone area. Alert settings shall initially be predefined and not editable by the user. Capability for the user to specify custom alert settings will be added if time permits.

4 Research

4.1 Existing Similar Projects

While researching for this project, several products were found that are similar to our concept. One that stood out the most was a product called PocketFinder®. This product is marketed as a “personal GPS locator” useful for tracking people, pets and vehicles. In addition to the obvious similarity of being GPS-based, this product is similar to ours in that it also features a GSM module, web interface, and complementary iOS application.

While the PocketFinder is an excellent product, we believe our product is unique in the fact that it is designed specifically for tracking luggage. In all of our research, no product was found that was designed for this specific use. As technology in the GPS market improves and costs are driven down, we believe our product will have the potential to be marketed as an affordable option to be purchased and placed into luggage. Because the cost of GPS technology is still relatively expensive, the long-term plan for our product would be to integrate it into a line of luggage. The PocketFinder tracking unit is shown in Figure 4-1 below:

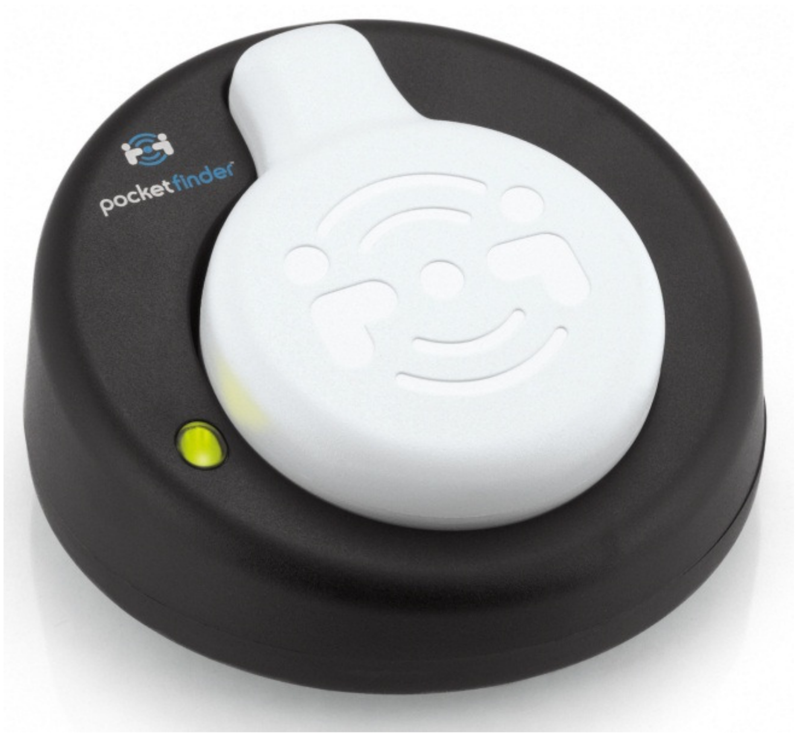


Figure 4-1 PocketFinder® GPS Tracking Unit

Used with permission from Location Based Technologies, Inc.

Luggage is already a very expensive commodity, so adding another roughly 100 dollars to the cost would not be as significant of an expense to prospective consumers as a small electronic device that must be purchased separately. Luggage manufactures would, of course, advertise the fact that their luggage lines feature built-in GPS tracking units to help provide a worry-free travel experience.

Another similar project found while researching is the UberTracker. The UberTracker is a GPS tracker produced by Sparkfun Electronics™ and is marketed as a potential project component for hobbyists and electronics enthusiasts. The UberTracker is much more “open source” than the PocketFinder product in the fact that the UberTracker has significantly greater amount of information available regarding its internal components and features.

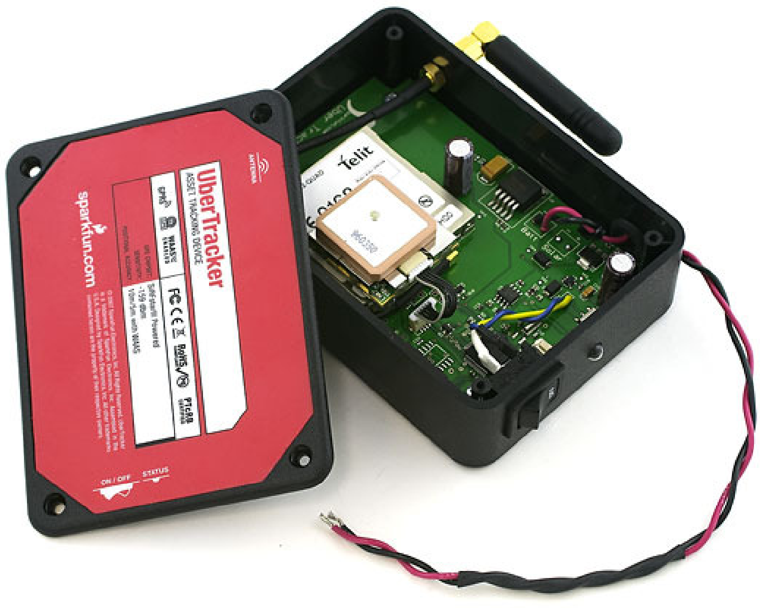


Figure 4-2 UberTracker GPS Tracking Unit
Used with permission pending from Sparkfun Electronics™

We plan to use the UberTracker as a guideline as we develop our own project, as it is very similar in both design and implementation to what we hope to accomplish.

4.2 Relevant Technologies

4.2.1 GPS

The technology most relevant to our project is most certainly GPS technology. Because GPS technology is satellite-based, it provides the most widely-available and cost-effective solution for wirelessly tracking a device anywhere in the world.

4.2.2 GPS Server

GPS servers are a relatively recent technology, but thankfully they have progressed swiftly to a level of reliability acceptable for this project. In addition, there are several open-source GPS servers available for free download and use, which is of great benefit to us both technically and financially. The main purpose of a GPS server is to receive incoming messages from a GPS tracking unit and distribute the relevant information to the user in various forms. Data is typically sent to the GPS server via GSM or General Packet Radio Service (GPRS) communication. Upon receipt by the server, the data is processed and often stored in a data logger for later retrieval. The data is then sent to devices such as a user's cell phone or email account. The data is also frequently used to populate a map so the user may view the current locations of his or her tracking units as well as their previous paths of travel.

4.2.3 GSM

GSM technology is another crucial part of this project. GSM was chosen as the preferred method of wireless communication because of its global coverage. The next best alternative to GSM for wireless data transmission would have been CDMA, which is currently available only in North America and parts of Asia. Because luggage might be traveling anywhere in the world, it is vital to select a technology with global communication capabilities.

4.3 Strategic Components

4.3.1 GPS Module

The GPS module is a strategic component in our design for two main reasons: connectivity and power. Although these two features are proportional in amount, unfortunately the affects they have on the system are inversely proportional. In other words, GPS modules vary in receptivity. The better the reception capabilities, the better connectivity the device will possess. The negative aspect is that the better the receptivity of the GPS module, the more power it consumes. Therefore, a balance must be achieved between the two which provides a high level of reception while consuming as little power as possible.

4.3.2 GSM Module

The GSM module is a strategic component in our design for reasons similar to that of the GPS module. The tradeoff between reception capability and power consumption is a delicate one that must be balanced and optimized. Another important feature of the GSM module is its ability to communicate in different protocols.

For the purposes of sending SMS text messages straight to a user's cellular phone, the standard GSM feature is sufficient. However, for the purpose of communicating with a GPS server, the GSM module will most likely need to have GPRS capabilities. GPRS is a data service built on the GSM network that allows internet protocol (IP) packets to be sent to external sources, achieving an effect similar to that of a Wi-Fi network. This capability would allow the GSM module to communicate with a GPS server located on the internet or on a personal computer.

4.3.3 On-Chip Software

Because our project is being developed in phases, the software components we design will play a very strategic role. If not designed properly, the software may possibly need to be redesigned and rewritten in order to allow for additional features to be implemented.

This is especially true for the on-chip software. The on-chip software must be designed with future capabilities in mind. Initially, the data will be received from the GPS module, processed into a text message, and then sent to the GSM module for transmission. However, in the later phases of development, a GPS server will be implemented and data will need to be processed and distributed in a different manner. Instead of being compiled into a simple text message, outgoing data will need to be generated in a format compatible with the GPS server.

With this in mind, the on-chip software design should be as object-oriented as possible. This will allow for relatively easy changes to be made without affecting the software as a whole. The principle of object-oriented design is one that will be applied to the entire project, but especially to the software packages being developed by the team.

4.3.4 GPS Server

As mentioned previously, the plan for this project is to implement a GPS server. The GPS server is a strategic component primarily because it will allow for multiple LTUs to be tracked and viewed simultaneously. Without a GPS server, the extent of the user's capability to manage his or her devices would be limited to receiving and viewing individual text messages and emails from each individual LTU. With a GPS server, the user has the ability to view the status of any or all LTUs from one source, greatly reducing the interaction and work required on the user's end. A GPS server will also provide the ability to store and process data at a central location, as all LTUs belonging to the user will send their data to the same server.

5 Design Details

5.1 Design and Development Strategy

In order to ensure a functional product is developed, our strategy is to design and develop prototypes in phases. These phases will start with a design that meets the basic requirements of the project and then increase in complexity and robustness with each subsequent phase. As the phases of development progress, the initial prototype will be expanded upon and at least one additional prototype will be produced in order to incorporate additional features. However, regardless of whether the subsequent development phases succeed or fail, it is ensured that a working prototype will be available to demonstrate at the end of the Senior Design course.

The current path forward is to have three phases of development: A, B, and C. These phases will be as described in the table below:

Table 5-1 Design and Development Phases

Phase	Capabilities and Features
A	Ability to receive GPS signal and send GPS coordinates via SMS text message. In this phase we are simply making something that works and can be demonstrated, nothing more.
B	All capabilities of Phase A plus a GPS server interface which receives data from one or more LTUs and displays a map with the location of the user's LTUs. Explore transmission protocol options such as GSM vs. GPRS. Explore web interface options and features such as geozones, customized alerts, etc.
C	All capabilities of Phase B plus an iOS app interfaced with the GPS server that provides a map of LTU locations updated in real time. Also look into the option of integrating the LTU with actual luggage rather than having it as a standalone unit.

With respect to hardware prototypes, our strategy is to leave the Phase A prototype untouched once it has been completed. In doing this, we eliminate the risk of damaging or causing errors within the unit. The Phase A prototype will be the last-resort plan if implementation of the subsequent phases is unsuccessful. Should an acceptable level of confidence in the implementation of the GPS server component be achieved in Phase B, steps may be taken to alter the Phase A prototype to also be compatible with the GPS server.

For Phase C, the current plan is to not have an additional prototype. This is mainly for two reasons. First, the only significant changes in Phase C will be software-based, so there is not a pressing need for another hardware unit. Second, the cost of producing a third hardware prototype is too expensive to be worthwhile. Having only two prototypes still provides the ability to demonstrate tracking of multiple units, which is one goal we hope to achieve in the this project.

5.2 Risk Assessment

The overall scope of this project contains several features which present significant risk factors in the design and development process. The main cause for these items being a high risk is due to a lack of experience in those areas. Although our team has a wide variety of background and experience, several areas remain in which our team has little or no experience. The amount of risk involved with these items contributed significantly to the decision to divide the development process into phases. Only the most critical functions were included in the initial phase, while each subsequent phase adds more features with increasing development risk.

5.2.1 Phase A Risks

Because Phase A is the most crucial phase of our development process, a great amount of attention must be focused on any issues that present a risk to the successful completion of this phase. One area that presents a significant concern is the software control of the GSM/GPS module. The GSM and GPS components within this module must be able to communicate in order to achieve the basic function of transmitting GPS information to the user. Data gathered by the GPS module must be processed into an SMS text message format and sent to the GSM module for transmission. How this will be accomplished is an issue that must be addressed.

5.2.2 Phase B Risks

The only high risk item in the Phase B development period is the GPS server. None of the members of our team currently have any relevant experience with databases development, setup, or application. For this reason, the GPS server used for this project will be a free, open-source software package that is largely pre-configured.

However, even with the benefit of using a pre-configured software package, there is a significant amount of risk involved with simply integrating the server into our system and interfacing it with LTUs. This task has the potential to be a very time-consuming undertaking, as it is currently not known how the integration will be accomplished. A large amount of research will likely need to be done before an LTU is successfully configured to communicate with the GPS server. The GPS server is discussed in greater detail in section 0.

5.2.3 Phase C Risks

Another at-risk item, and the one that presents the greatest risk of failure to achieve implementation, is the iOS application. As is the case with the GPS server, our team currently has no experience in the realm of mobile application development. In order to develop an iOS application, the iOS programming language and developer environment will have to be learned before development can even begin. The amount of time required to accomplish this is largely unknown, so the feasibility of attempting this feature of the project is yet to be determined.

The path forward is to begin initial research and investigation during the months of July and August while the main focus of the development effort is on the Phase A prototype and the GPS server implementation. Once the GPS server has been implemented and proven functional, development on the iOS application will begin immediately and continue until a successful implementation is achieved. The iOS application is discussed in greater detail in section 5.6.4.

5.3 Functional Architecture

5.3.1 GSM Module

When the LTU is powered on, the GSM module will automatically power on and enter full power mode. The first task of the module will be to check that the conditions necessary for message transmission are met. These conditions will include factors such as power, altitude, and pressure. If the GSM module is not receiving sufficient power from the battery, this is an obvious condition that will prevent the GSM module from sending. Therefore, the module should not attempt to send messages unless sufficient power is available. Failing to check this condition could potentially result in messages being lost. This is a possibility because the module could send a transmit command and assume the messages were sent. This in turn would cause them to be deleted from the transmit buffer, when in reality the messages were not transmitted due to insufficient power.

Altitude is another condition that must be checked before allowing the GSM module to connect to a network or transmit messages. Because the LTU is designed for tracking luggage, it will often be the case that the luggage being tracked is on an airplane. Most airlines have strict rules prohibiting the operation of cellular devices while onboard the aircraft. To maintain compliance with these guidelines, the GSM module must not be allowed to power on when onboard an aircraft. The best way to determine whether or not the LTU is on an aircraft is to monitor its altitude. The GPS receiver in the GM862 chip provides altitude data as one of its standard readings. A maximum altitude ceiling will be programmed into the LTU memory and compared with the most recent GPS reading. If the GPS reading shows an altitude greater than the determined ceiling, the GSM module will not be prohibited from sending messages and will be returned to a low power mode.

An additional safety mechanism that may be implemented to prohibit the GSM module from transmitting while on an airplane is to integrate a pressure sensor within the LTU. Since most luggage is carried in unpressurized storage compartments underneath the aircraft, a barometer may be used to take readings of the surrounding air pressure. If the pressure readings are significantly lower than the average air pressure at ground level, it may be assumed that the LTU is currently on an aircraft and should not be allowed to transmit messages.

If all conditions for transmission are met, the GSM module will attempt to connect to and register with any available GSM network. If the module successfully connects to a network, it will transmit all messages in the transmit buffer at that time. Once the messages have been sent, the module will return to a low power mode. If a network is not found, or if a network is found but the signal strength is too weak to connect, the module will continue to try to connect for a maximum time of 60 seconds. At the end of the 60 second time limit, if the module has still been unsuccessful in connecting to a network, it will return to low power mode.

Once the module enters the low power mode, a 15-minute timer will begin. When the timer reaches zero, the GSM module will reenter full power mode and begin the network acquisition and message transmission process again.

The activities performed by the GSM module are summarized in the activity diagram in Figure 5-1 below:

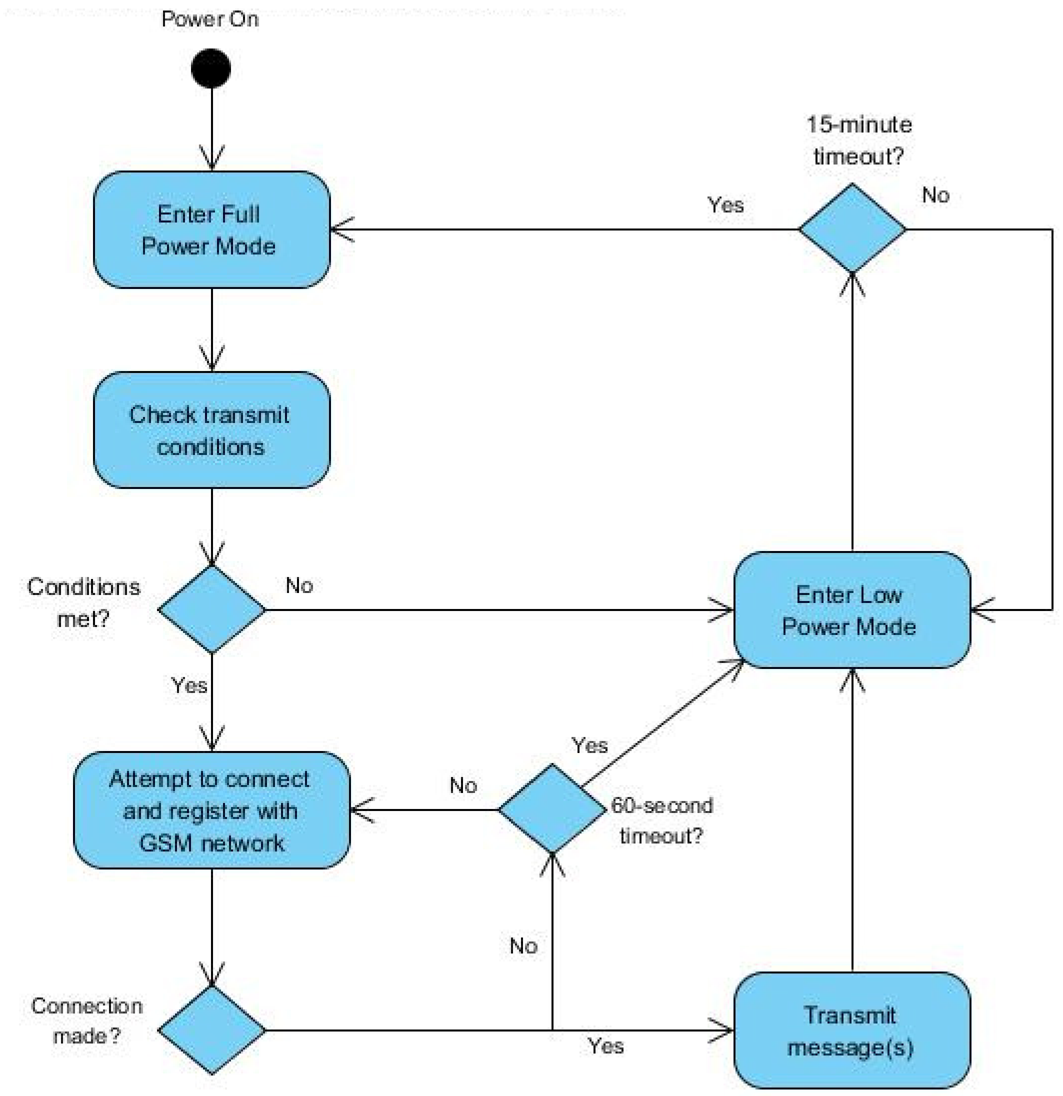


Figure 5-1 GSM Module Activity Diagram

5.3.2 GPS Receiver

The activity cycle of the GPS receiver is very similar to that of the GSM module. The GPS receiver will be powered on when the LTU is powered on. Upon being powered on, the receiver will be in full power mode and will immediately attempt to acquire a GPS fix. Unlike the GSM module the GPS receiver has no restrictions for use onboard a commercial airplane. For this reason, there is no need for any conditions to be checked before the receiver is allowed to search for available GPS signals.

If a sufficient number of GPS satellites are in range enabling a successful GPS fix, the data obtained will be stored in memory where it will later be compiled into a message to be transmitted by the GSM module. Once the data has been stored in memory, the receiver will enter a lower power mode to minimize power consumption. In the case that the receiver is not able to immediately acquire a GPS fix, it will continue to attempt to acquire a fix for 60 seconds. If at the end of this time a connection has still not been made, the receiver will enter a low power mode. Upon entering the low power mode, a 10-minute timer will be started. Once the timer reaches zero, the GPS receiver will reenter full power mode and begin the GPS satellite acquisition and data storage process again.

The activities performed by the GSM module are summarized in the activity diagram in Figure 5-2 below:

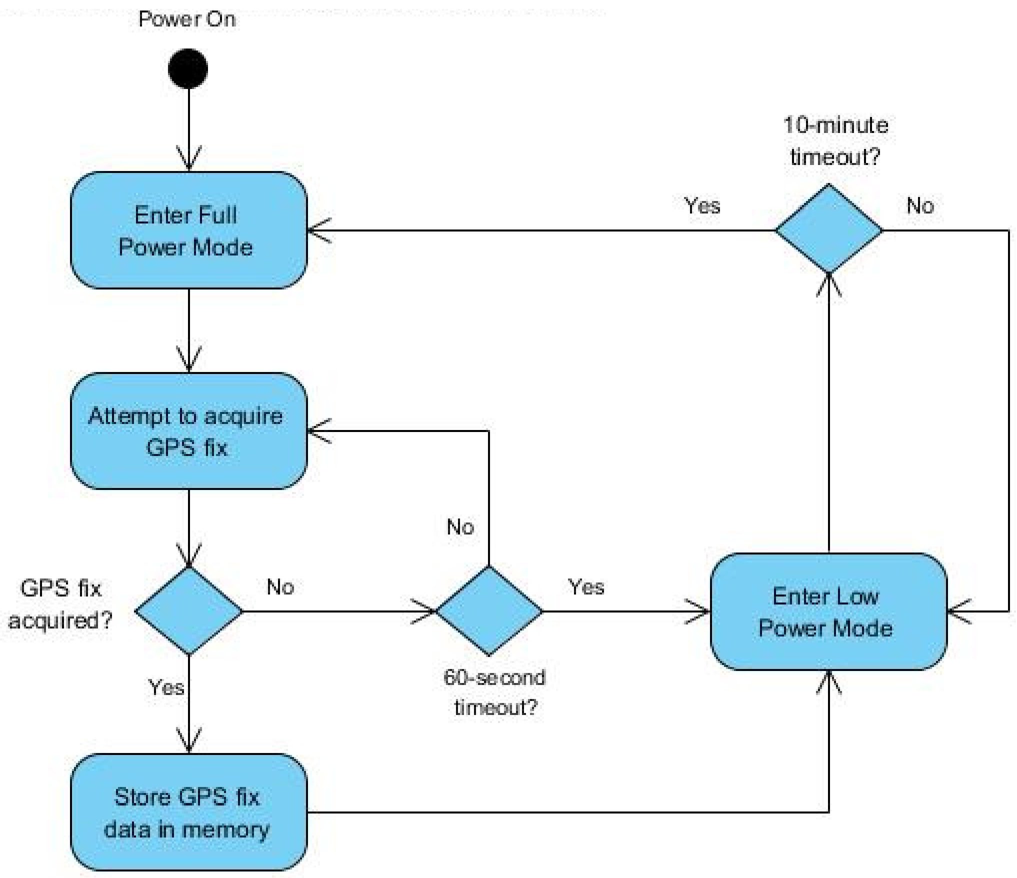


Figure 5-2 GPS Receiver Activity Diagram

5.4 Hardware Architecture Diagrams

Figure 5-3 shows an overview of interconnections for the hardware components.

The microcontroller will serve as the master controller for handling all communications. Moreover, the microcontroller will monitor the subsystems in order coordinate sequence of operation and minimize power consumption. The microcontroller will interface via serial peripheral interface (SPI) serial communication bus with the GM862 and the flash memory. For Network status and GPS status, the outputs provided by the GM862 will be utilized. These outputs energize two LEDs. A green LED for Network status and availability and a red LED for GPS status. A third yellow LED will be activated by the microcontroller to indicate that the unit is On. The microcontroller will also monitor the Barometric sensor via I^{2}C serial communication. The GSM and GPS antennas are connected to the SMA receptacles on the GM862. The connection is made via a coax cable and an MCCX male connector. The push buttons are for user interface to power up or shut down the LTU and to reset the GM862 unit. A light sensor will be used to indicate that the luggage has been opened. This sensor will be connected directly to the microcontroller.

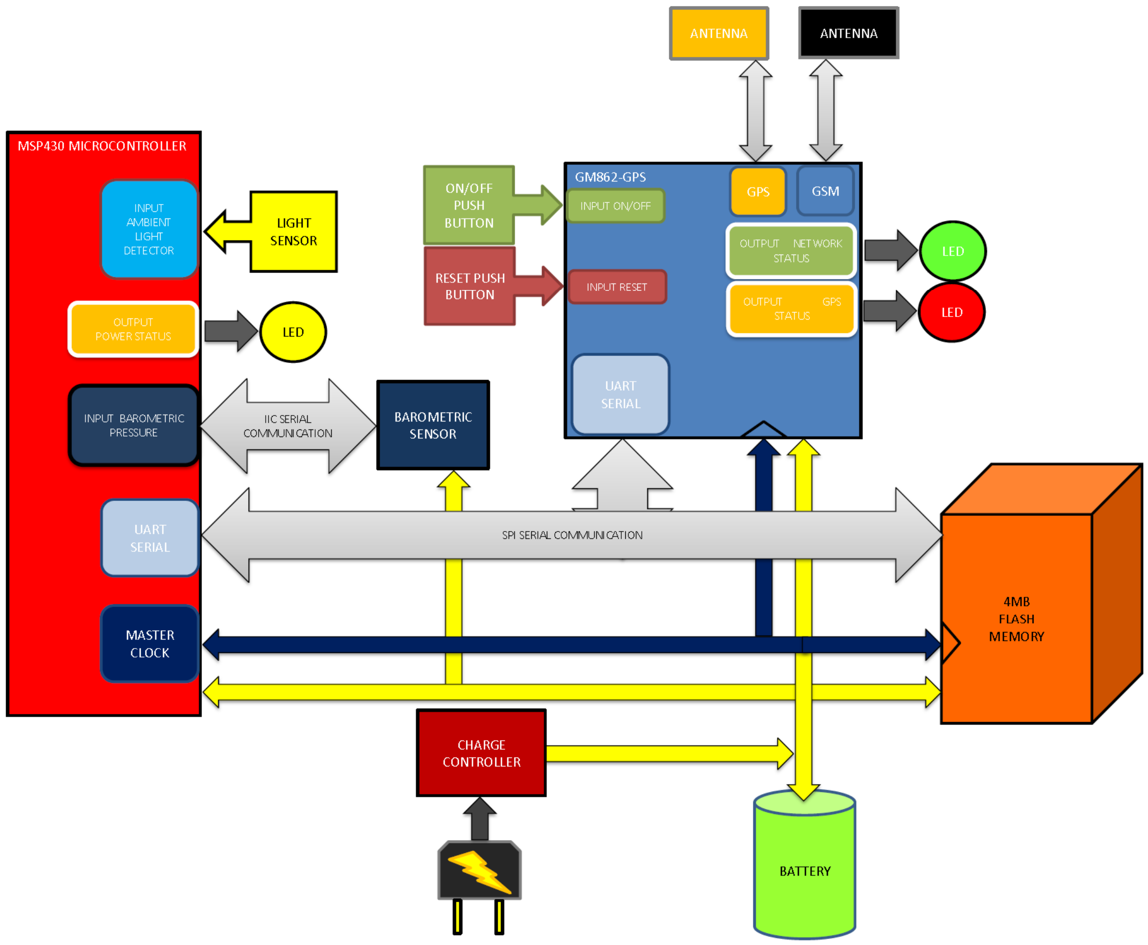


Figure 5-3 Hardware Architecture Diagram

5.5 Hardware Components

5.5.1 GPS and GPRS/GSM Module

Device: GPS receiver and GPRS digital communication GM862-GPS.
Part Number: GM862-GPS
Manufacturer: Telit.
Description: Quad-band GSM/GPRS modem with 20-channel high sensitivity silicon radio frequency (SiRF) Star III GPS receiver.

Figure 5-3 shows the general dimensions for the GM862 module. In order to interface with this module, a male connector SMD 50-pin will be used on the PCB. The GM862 also has a Mini-SIM card holder or receptacle, an SMA female connector for GSM/GPRS antenna and an SMA female connector for a GPS antenna.

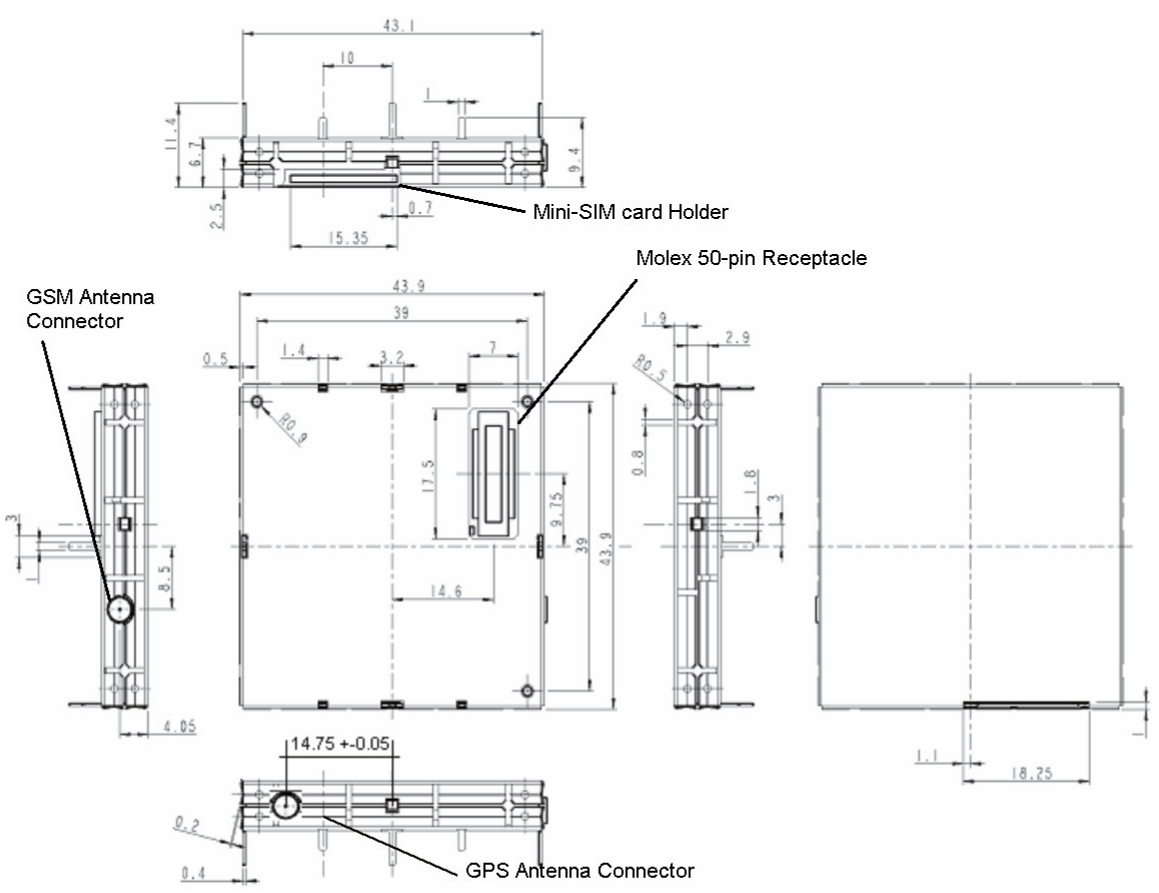


Figure 5-3 Telit GSM/GPRS/GPS Module GM862-GPS
Used with permission pending from Telit® Wireless Solutions.

5.5.2 GSM Antenna

Device: GSM/GPRS Antenna.

Part Number: S181FL-5-RMM-2450S
Manufacturer: Nearson.
Description: 2.4GHz Fixed Mount Swivel Antenna.

In order to select a suitable GSM/GPRS antenna for the GM862 module, mobile network operating frequencies have to be defined. Since the GM862 supports the following frequencies: 80 MHz in EGSM 900, 70 MHz if GSM 850, 170 MHz in DCS, 140 MHz PCS band. For the LTU the frequency selected is 70 MHz which corresponds to GSM 850. The mobile service provider that supports GSM 850 is AT&T.

Moreover, GSM antenna has been selected following the specifications of the GM862. 50 Ω impedance, less than 3dBi gain, and input power higher than 2 W peak. In order to match the female RF connector of the GM862, a coax cable with MMCX Angle Plug Crimp is used.

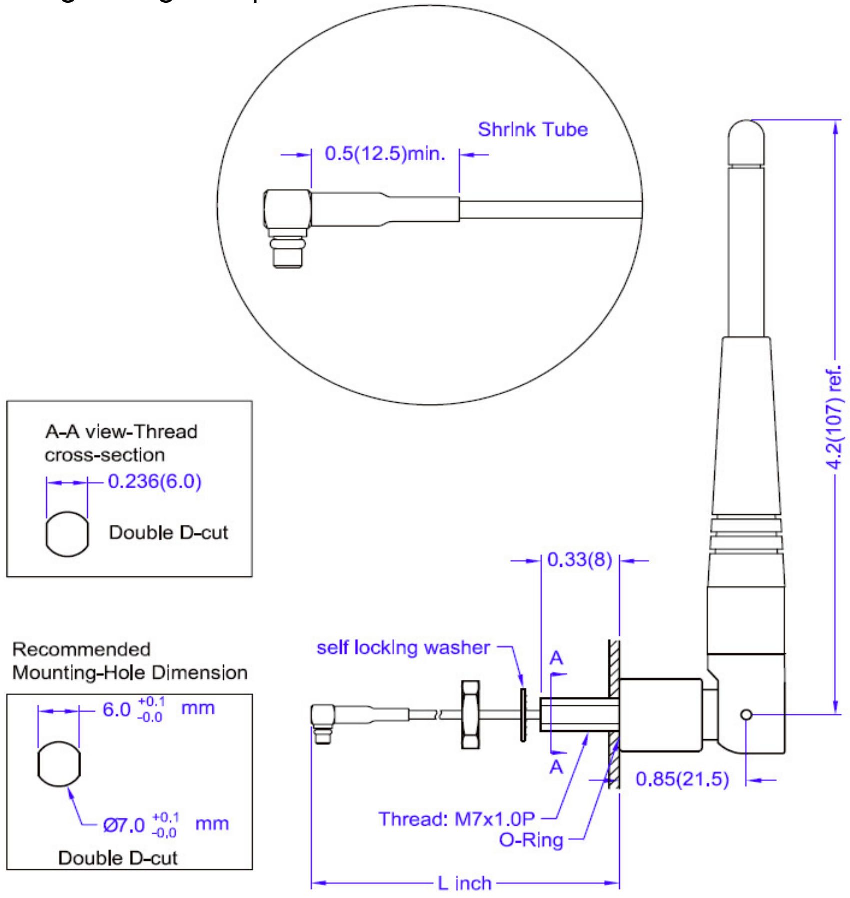


Figure 5-4 Nearson 2.4GHz Fixed Mount Swivel Antenna
Used with permission from Nearson, Inc.

The GSM antenna is connected to the side of the GM862 module, and the cable is routed to the side wall of the housing and extends through the outlet hole. The antenna cable is secured using a nut and star washer that is provided with the bulkhead/swivel antenna.

5.5.3 GPS Antenna

Device: GPS Antenna.
Part Number: VTGPSIA-3
Manufacturer: V.Torch.
Description: GPS Internal Active Antenna.

GPS antenna has been selected according to the specifications of the GM862. The frequency of GPS L1 is at 1575.42 MHz, bandwidth of +/- 1.023 MHz, between 1.5 dBi and 4.5 dBi gain, 50 Ω impedance, amplification of 25 dB typical, between 3 to 5 V DC supply voltage, and 20 mA typical current consumption.

The GPS antenna is connected to the SMA plug located on the side of the GM862 module. In order to save some space, the antenna is attached to the top of the GM862 module and the cable is routed around the module.

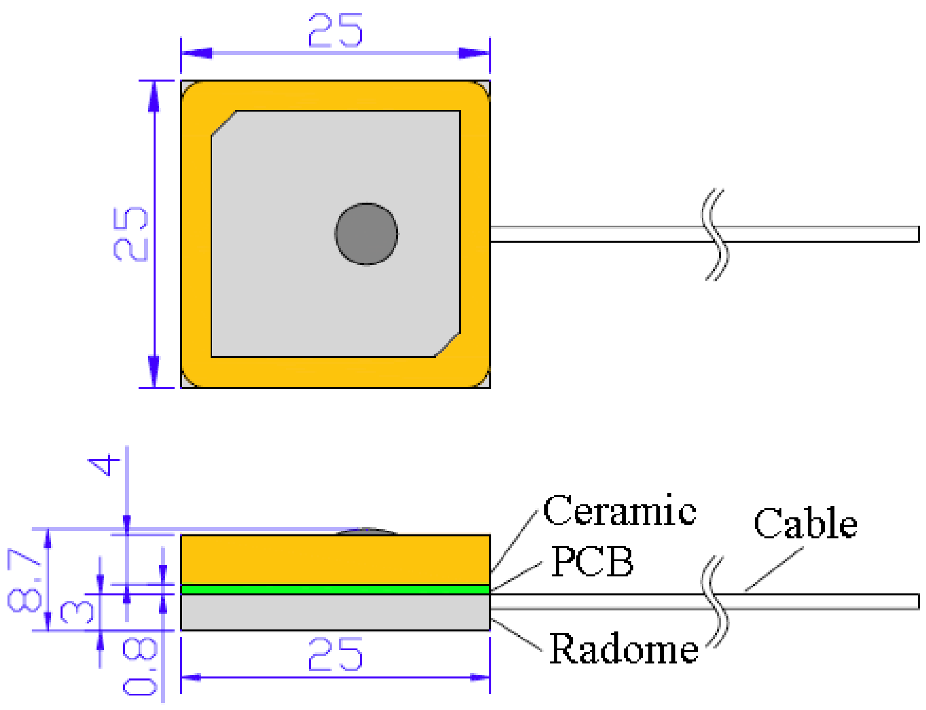


Figure 5-5 V. Torch GPS Internal Active Antenna. VTGPSIA-3
Used with permission pending from V. Torch.

5.5.4 PCB and Supporting Components

A PCB will be designed in order to hold all circuitry components that will support the power supply, on/off button, signaling, input and output interfaces, as well as antennas.

5.5.4.1 Connectors

5.5.4.1.1 50-pin board-to-board connector

GM862-GPS module has a board-to-board male connector type CSTP 50 pin vertical SMD Molex 52991-0508. The required mating connector that will be mounted to the PCB is the Molex 501920-5001 and its layout is shown in Figure 5-6. This connector is labeled “CON1” on the schematic diagrams.

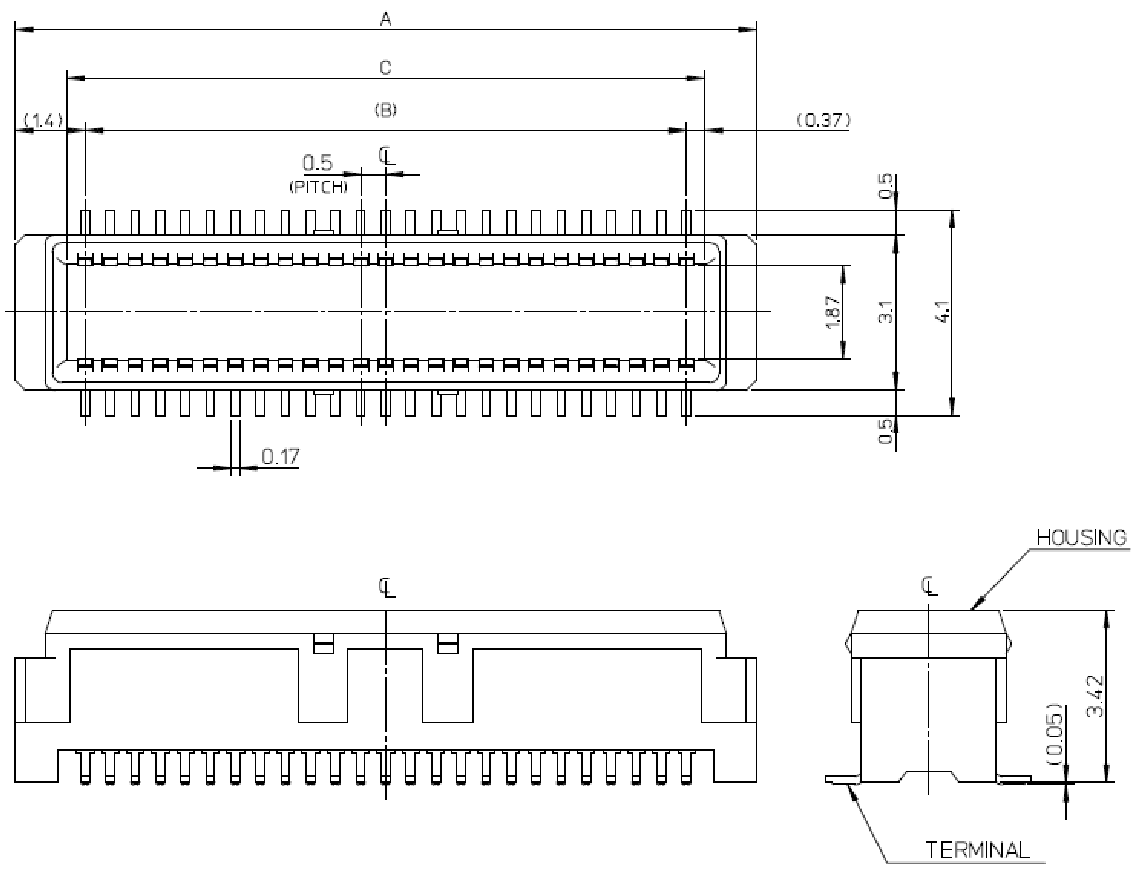


Figure 5-6 Molex 501920-5001dimensions
Used with permission from Molex®.

5.5.4.1.2 14 pin IC socket

In order to facilitate programming of the MSP430 microcontroller, a 14-pin IC socket will be used for the first PCB unit.

5.5.4.1.3 Mini-serial Receptacle

The Mini-USB receptacle allows connection to the Universal Power Charger in order to charge the battery with the on board controller located on pin 8

CHARGE of the GM862. It also can be used as a serial communication port for programming and parameter-setting of the microcontroller and the GM862-GPS module. One of the benefits of using this connector and its correspondent cable and charger, is that since 2009 several cellphone manufacturing companies have agreed to embrace to a Universal Charger Solution which defines the Micro and Mini connectors as the choice for future design. This allows users to utilize chargers across different products and whenever they change cellphone devices. This connector is labeled “CON2” on the schematic diagrams. Figure 5-7 shows dimensions of the Mini-USB receptacle.

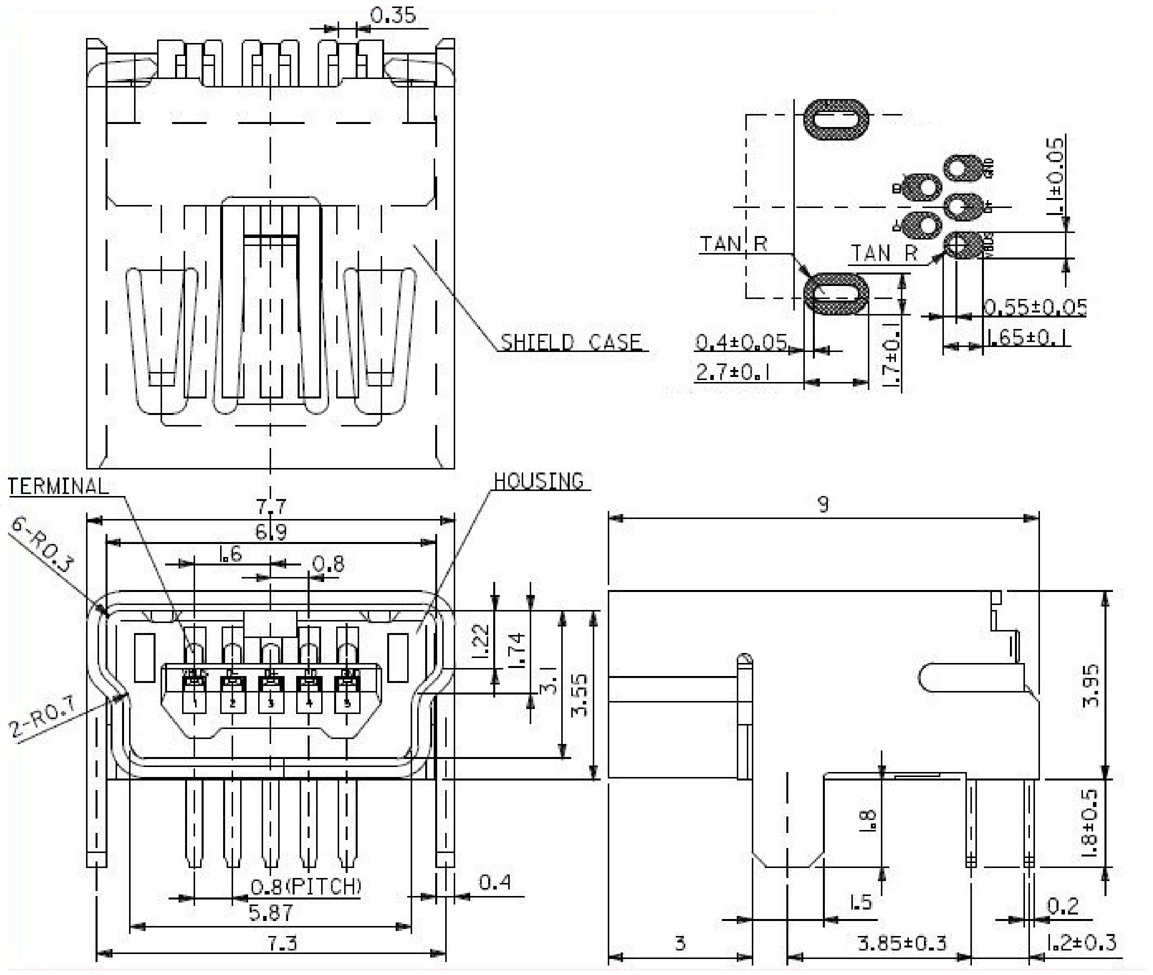


Figure 5-7 Molex 54819-0519 dimensions
Used with permission from Molex®.

5.5.4.1.4 Battery Receptacle

A high current, compact, two-connector receptacle designed for surface mount will be used to connect the battery to VBATT pins 1, 3, 5, and 7. This connector corresponds to the proper gender mate for the battery and allows for interchange in case of malfunction or damage to the battery. This connector is labeled “CON3” on the schematic diagrams.

5.5.4.2 Input Signals and Push Buttons

5.5.4.2.1 ON/OFF PB1 Push Button

A switch mounted on the side of the housing allows the device to be turned on and off. This switch is a normally-open push button with momentary close when actuated. When the push button is depressed, it connects the base of the NPN transistor to 3.7V VBATT. This actuation should be for more than 1 second. Then a low level signal is forced on the internally pulled up pin 17 ON/OFF of the GM862. This allows the module to be turned on or off properly. Moreover, it correctly shuts down the GSM module by issuing a detach request to the network where it is registered.

5.5.4.2.2 RESET PB2 Push Button

A small momentary push button is located on the PCB board to allow the user to reset or stop any operation. This switch is a normally-open push button with momentary close when actuated. When the push button is depressed, it connects the base of the NPN transistor to 3.7V VBATT. This actuation should be for more than 0.2 seconds. Then a low level signal is forced on the internally pulled up pin 23 RESET of the GM862. For the GM862-GPS, this reset function acts as an unconditional shut down, which means that no detach request is sent to the network.

5.5.4.3 Output Signals and Status Indicators

5.5.4.3.1 Network Status LED

A green LED is used to indicate Network Service Availability. This LED is connected to the open collector pin 35 STAT_LED of the GM862. The output of this pin is the inverse of the status signal. A 470Ω resistor is used in series to the 2V LED and is pulled up to 3.7V VBATT for correct operation.

5.5.4.3.2 GPS Status LED

A red LED is used to indicate GPS status and availability. This LED is connected to pin 31 GPIO1 of the GM862. The output of this pin is the inverse of the status signal. A 470Ω resistor is used in series to the 2V LED and is pulled up to 3.7V VBATT for correct operation. The GM862 does not offer an output with GPS status. Therefore, with the use of attention (AT) commands, the output pin GPIO1 will need to be programmed in order to provide GPS signal status and availability.

5.5.5 Mini-SIM card

The GM862 module has an internal SIM card interface that accepts and reads a Mini-SIM 1.8V and 3V (ISO/IEC 7810:2003, ID000). The SIM card plays an important role in establishing communication and allowing access to the mobile network provider. The SIM card's memory will be used to save the server access number in order to upload data and download settings and parameters changes.

Also, the LTU user's phone number will be saved on the SIM card to be used for SMS messaging. The SIM card is inserted into the GM862 module. Insertion side and direction is shown in Figure 5-8.

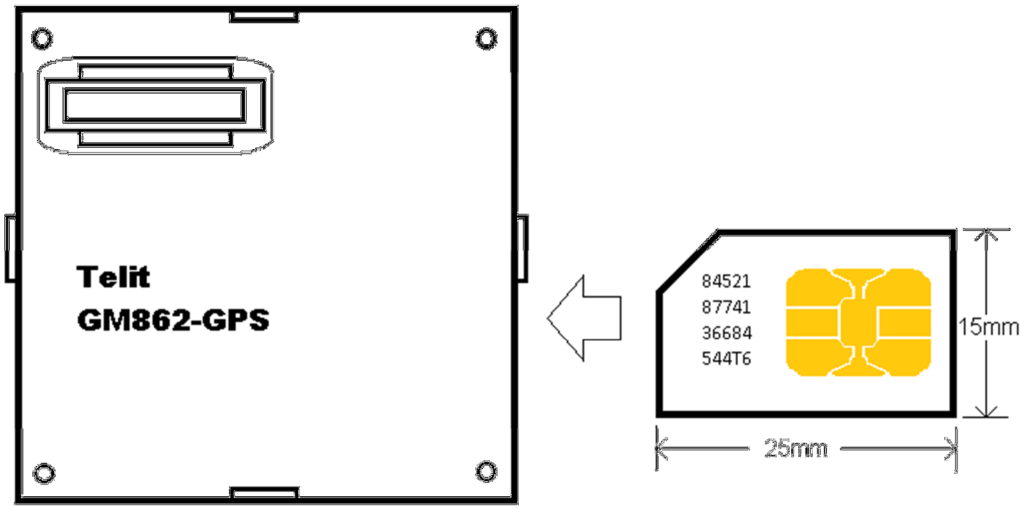


Figure 5-8 Mini-SIM card dimensions and insertion into the GM862-GPS
Designed by Jose Mousadi

5.5.6 Enclosure

A plastic two piece enclosure will be used. Preliminary dimension are for a 4" wide by 4" high by 0.5" deep enclosure. The wall thickness is still to be defined; however, a thick enough material will be necessary to guarantee the proper mounting of a push button. It will also have provisions for attaching the PCB board to the enclosure while providing enough space for routing the mini-coax cables for the two antennas and mounting of one of the push buttons to the side wall. The enclosure shall be made of material that can be machine-drilled to create holes for push buttons and a Mini-B USB power receptacle. A translucent enclosure will allow for visualization of the LED indicators without having to open more holes.

5.5.7 Battery and Power Supply

5.5.7.1 Battery Selection

Since all the peripherals will be driven by the GM862 module, the battery is chosen according to the requirements of the module. Modules with a SW release 7.03.x00 or newer require a nominal supply voltage of 3.8 V. Its normal operation voltage range is between 3.4 V to 4.20 V. This module also has an extended operating voltage range between 3.22 V and 4.50 V. The supporting circuitry will have a 100µF installed in parallel to the VBATT pin of the GM862.

5.5.7.2 Battery Charger

A charge controller will be utilized to charge the Lithium-Ion battery following the recommended charging current curve. This charge controller will allow the user of the LTU to connect a wall adaptor used for cellular phones to charge the battery via a Mini-B USB receptacle. The battery charger is designed to support a 3.7 V Lithium-Ion rechargeable battery with a suggested capacity of 1000 mAh. A zener diode is also implemented in order to protect against inverted voltages and voltages over 6.4 VDC. Also, a current limiter circuit will be implemented in order to protect the GM862 module. This current limiter will provide up to 400 mA max input current, with input voltage of 5.0 V min, 5.5 V typical, and 7.0 V max.

5.5.8 I/O Components and Sensors

5.5.8.1 Barometric Pressure Sensor

A pre-calibrated barometric pressure sensor with a Form-A contact will be used to shut down the LTU while in transit inside an airplane. This feature will accomplish two functions. First, the operation of the unit will follow the FAA guidelines and rules for electronic equipment in aircrafts. Second, the barometric sensor will prohibit the unit from attempting to access GPS data, logging the data or trying to establish transmission via GSM.

The barometric pressure sensor will be calibrated to an altitude of 1000m above sea level. This sensor will provide low signal while below the altitude limit and a high level signal while over 1000m. This signal will be connected to the GPIO2 which will be programmed to execute interrupts to the module and properly shut down the LTU via the microcontroller.

5.6 Software Components

5.6.1 On-Chip Software

The language used to write and simulate the on-chip software will be Python. This is the required language to correctly interface the LTU with the GPS module and server. Python’s lack of explicit types will allow our system to communicate efficiently, reducing the number of potential errors. Because our system contains a high level of integration, it is important to establish reliable communication lines between the several, independent modules. Python makes this communication straightforward because of its lack of types. That is, data can be conveniently sent from one module to another without having to parse the data before sending.

The user settings and potential firmware updates will also make use of Python. Although these settings and information may be developed in a different language, it will eventually be converted to a form such that Python can fully understand the information. The iOS software will also be written in a different language. However, in order to effectively communicate with the Luggage Tracker Unit, it too will need to be converted to Python so the on-chip software can properly process and make use of the data.

5.6.1.1 Memory Constraints

There are strict memory limitations on Telit Python modules. In order to avoid RAM overflow and other potential memory errors, the defined memory bounds on the modules must be explicitly understood. The total memory available on Telit Python modules is comprised of the following:

- approximately 2MB of non-volatile memory for user scripts and data
- 1.2 MB RAM reserved exclusively for the Python engine
- 16KB of memory for each variable

The effect of these constraints can be further grasped by understanding the individual limits for each type of variable in the Python language. The table below contains the estimated memory limits associated with each variable type. It should be noted that each element of dictionary, list, range, or tuple has up to 16KB of available memory.

Table 5-2 Memory Constraints Based on Variable Types

Type	Number of Elements	Sample
dictionary	682 (worst condition)	{‘xxx’:1000, ‘yyy’:1001}
list	4,000	[12,’data’,’a’]
range	4,000	range(3)=[0,1,2]
string	16,000	‘data’
tuple	4,000	[12,’data’,’a’]

5.6.1.1.1 Python Startup

To avoid memory errors, the startup process associated with the Python task should be further examined to understand what information is loaded into the system’s memory. Each time the system is started, the Python task loads a list consisting of the following items:

- method names
- modules names
- strings delimited by ‘ ‘ or “ “ (if not terminated with \r)
- variable names

These names are all included in the main script of the Python task, as well as all of the *.py files referenced directly or indirectly by the main method. Approximately 500 overall names can be loaded at each startup initiated by the Python task. Therefore, variable names will remain consistent for each *.py file associated with this project. Applying this method will allow the system to only allocate enough memory for one variable name associated with numerous *.py files, rather than using memory to simply rename variables that seek to serve the same purpose within the system. The final compiled *.pyo file should not be more than 16KB.

5.6.1.1.2 Data Logger and Flash Limits

The Luggage Tracker Unit will continually log data to keep track of its current position and time throughout its journey. However, an external memory module will be necessary to log this data because the flash memory used on the Telit module has a limited number of writing and deleting cycles. The limits of the non-volatile memory are described in the table below. These limits should be considered when developing the on-chip software and data logger to ensure no errors are encountered when logging data.

Table 5-3 Non-Volatile Memory Limits

Maximum length of file name	16 characters
Maximum number of files open	16
Maximum number of files saved	255

Some of the other Python limits that should be considered when developing the Python script to run the on-chip software of the Luggage Tracker Unit are listed below:

- Writing in non-volatile memory over 2MB may cause a decrease in writing speed
- The AT#LSCRIPT may not show an exact number of bytes that can be used for non-volatile memory due to dynamic memory reorganization
- A 2 second pause (MOD.sleep(20)) should be inserted after each AT#DSCRIPT command
- A 50 millisecond pause should occur each second during the Python task activity as to avoid interference with the GSM and GPRS standard operations
- The general purpose I/O (GPIO) polling frequency is <100Hz
- SPI and I2C speed is from 10Kb/s to 20Kb/s

5.6.1.2 GSM/GPS Module Software

The Telit GM862 module, which contains both a GSM modem and a GPS receiver, will operate on an internally-stored Python script designed and programmed by our team. The Telit module is equipped to store and run Python scripts to provide the developer with a high-level control interface rather than requiring all communication to be performed through raw AT commands via the serial communication ports. In essence, the Python script serves as the mediator and translator between what the designer wants the component to do and how this is actually accomplished by the component.

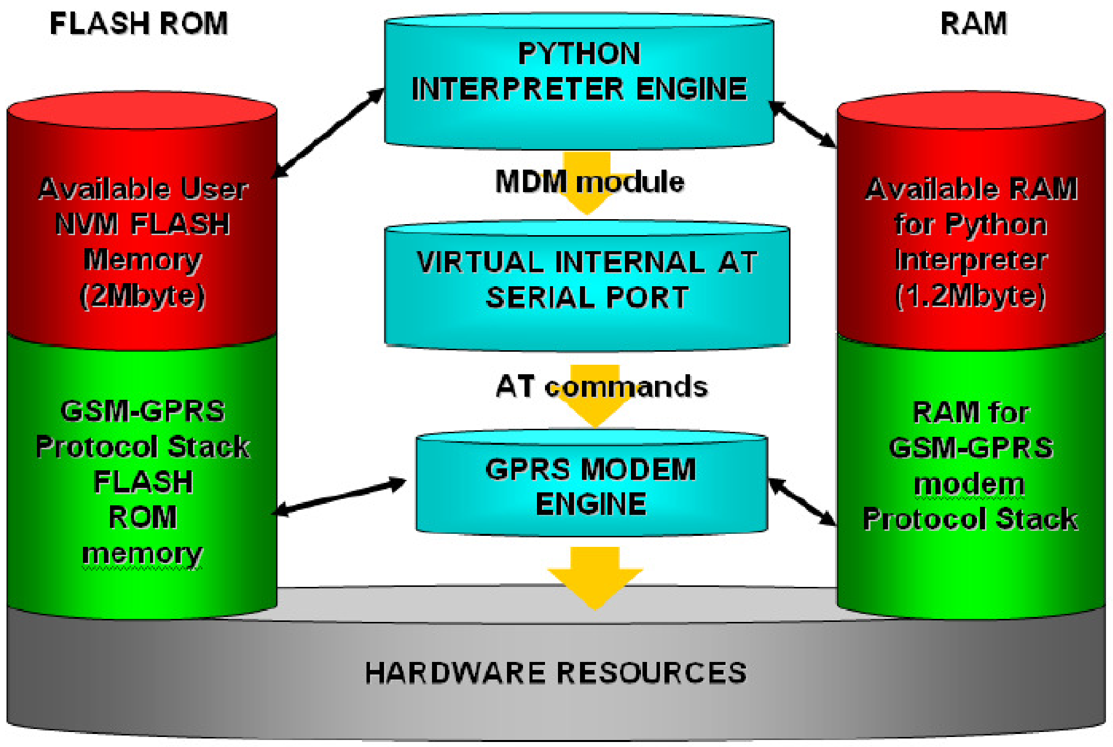

FLASH ROM
RAM
PYTHON INTERPRETER ENGINE
MDM module
VIRTUAL INTERNAL AT SERIAL PORT
AT commands
GPRS MODEM ENGINE
Available User NVM FLASH Memory (2Mbyte)
GSM-GPRS Protocol Stack FLASH ROM memory
Available RAM for Python Interpreter (1.2Mbyte)
RAM for GSM-GPRS modem Protocol Stack
HARDWARE RESOURCES

Figure 5-9 Python Engine Communication Diagram
Used with permission pending from Telit® Wireless Solutions.

The Python script engine interacts with the GM862 module through several different interfaces. These interfaces will be utilized by the design team to command the module into different power modes and perform tasks such as acquiring GPS fixes and sending status messages from the GSM module. The main interface that will be used most frequently is the MDM interface. This interface allows the Python script to both send and receive AT commands and data with the GM862 module. This communication is completely internal to the GM862, which frees the actual serial port on the hardware for other use.

Another interface that will be frequently used in this design is the GPS interface. This interface allows the Python script to communicate directly with the GPS receiver rather than having to communicate via the MDM interface. This will assist in reducing the response time of commands and increasing the efficiency of the software.

The SER and SER2 interfaces provide the capability for the Python script to read from and write to the ASC0 and ASC1 serial ports respectively. These interfaces may be used as an alternate method of communication should the internal software interface fail for some reason during operation.

The GPIO interface allows direct control of and communication with the GPIO pins on the GM862 module, increasing the overall efficiency of the module's operation. This interface will be used to control such items as the status LED, which is used as an indicator of whether or not the module has established a connection with a GSM network and if so, what the signal strength of that connection is.

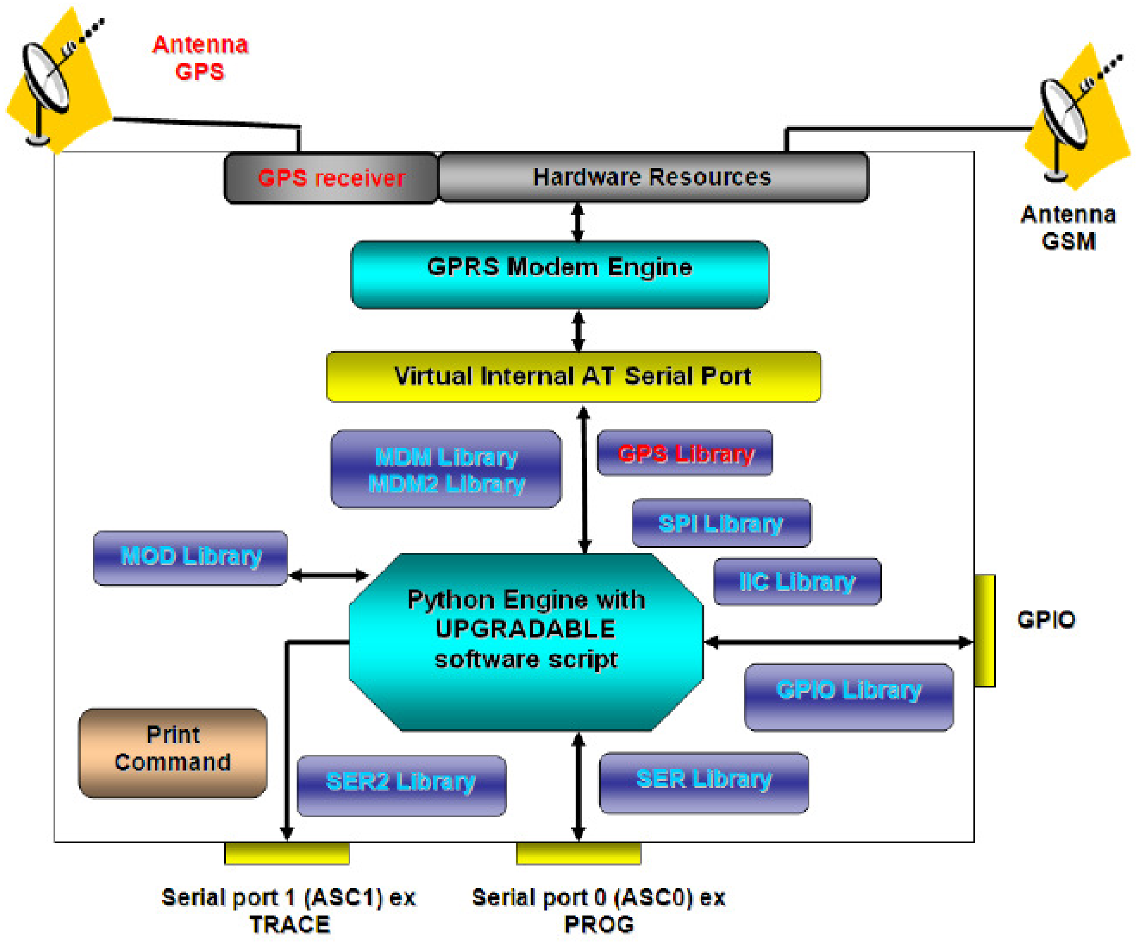


Figure 5-10 Python Engine Interface Diagram
Used with permission pending from Telit® Wireless Solutions.

5.6.2 GPS Server

The GPS server is the most critical component for the purpose of meeting the multi-device tracking requirement. Without a GPS server, communication capabilities are reduced to direct messages sent from each individual LTU to the user. The downsides to this method are significant. The main issue is that the user would have no central access point at which multiple devices could be viewed. Instead, the user would have to rely on messages arriving from the LTU's GSM module and then manually enter the data in a program such as Google Maps in order to view the location of their luggage.

While this would not be too much trouble if the user only had one piece of luggage to track, this is usually not the case. Most people travel with more than one piece of luggage and would, therefore, need the ability to track the locations of all of these pieces of luggage. If the user wished to view more than one device, messages would then need to be received from all of those LTUs and entered into individual Google Maps pages or in another similar mapping program. This method of viewing the location of luggage would be extremely time-consuming and frustrating for the user.

With a GPS server, however, the data from all active LTUs can be sent to a single location for processing and distribution. As an analogy, it is helpful to envision having multiple email accounts all receiving important messages. Without a central access point, it would be necessary to login to each individual account to view any messages sent to that address. This process becomes increasingly time-intensive as more and more email accounts are created. However, if all but one of those email accounts were configured to forward arriving messages on to a single account, then all the messages could be consolidated into one location and viewed with ease. That feature is the function of the GPS server: to provide the capability for all messages to be sent to a single location for efficient and simplified processing and access.

Due to the fact that our team does not have any experience with databases or servers, it was determined that, given the time constraints of this project, a pre-made GPS server would have to be downloaded from the internet rather than attempting to design and develop one from scratch.

After extensive research, Open Source GPS Tracking System (OpenGTS) was selected as the best option for this project. OpenGTS is a free, open source software package developed by Geo Telematic Solutions Incorporated. According to the project website, OpenGTS is “the first available open source project designed specifically to provide web-based GPS tracking services for a ‘fleet’ of vehicles.” In the case of our project, of course, the “fleet of vehicles” will be luggage.

While the OpenGTS software itself is coded entirely in Java, it utilizes a number of supporting software packages and features in order to function as a working GPS server. Figure 5-11 shows the overall architecture of OpenGTS as a system.

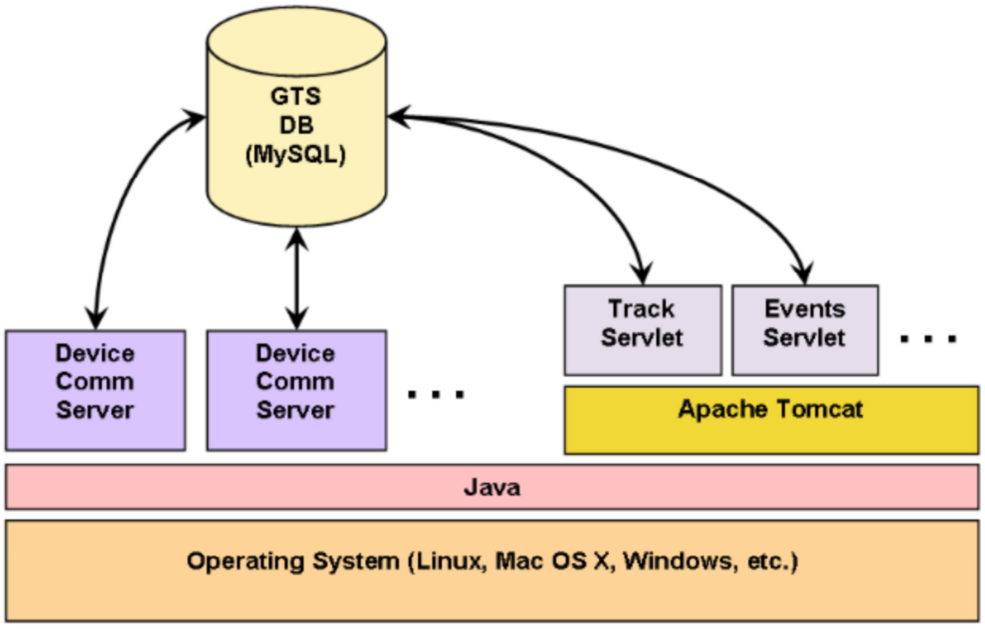


Figure 5-11 OpenGTS System Architecture
Used with permission from GeoTelematic Solutions Inc., copyright 2012, all rights reserved.

Although OpenGTS is developed in Java, as the figure above indicates, there are additional software packages to be implemented which complement the core OpenGTS software. Apache Tomcat software is used for the web interface, and MySQL functions as the database backend. With the support of these additional software components, OpenGTS offers a number of desirable features that will serve to enhance our project.

One such feature is the highly customizable, front end web portal. This webpage is preconfigured to work in unison with the server and database backend, which is of great benefit to our team as we have very limited experience in the realm of web development. However, the OpenGTS developers have provided a variety of options which the user can customize as desired. For example, reports can be generated to show the status of one or more tracking devices. The reports themselves can be customized to show certain types of data and certain levels of detail. Another useful feature is the geozone or geozone option. Geozones are virtual boundaries that can be set, usually as a certain radius around a given point, and alerts can be set to notify the user when a tracking device enters and/or leaves a geozoned area.

Another feature of OpenGTS, and one that is crucial to its implementation in our project, is the fact that it is device independent. While a number of GPS tracking devices are already supported by the OpenGTS software, their website claims that “with custom coding, other devices can also be integrated.” Since a GPS tracking device is what we are developing, it will be necessary for us to research and produce whatever custom coding is necessary to integrate our device with the OpenGTS software.

Yet another benefit of OpenGTS is that it offers support a variety of different mapping programs such as OpenStreetMap, Google Maps, Microsoft Visual Earth, and Mapstraction. This provides a great deal of flexibility in choosing how we wish to utilize data from the GPS server and distribute it to devices and programs for viewing by the user.

5.6.3 Web Portal

As mentioned previously, the web portal that will be implemented in this project is one that has already been developed by the OpenGTS project specifically for the use of interfacing with and pulling data from the GPS server and database.

The home screen of the website will feature four main tabs: Main, Mapping, Reports, and Administration. Under each of these tabs are numerous features and options which can be customized by the website administrator(s) to provide or not provide certain options to the user.

5.6.3.1 Maps

The main feature of the web portal will be the map page, which allows for viewing of multiple devices and their paths of movement. The mapping page will offer two main maps: vehicle map and group map. The vehicle map will display the current location and previous path of movement for a single LTU. A dropdown menu will provide the option to select other LTUs if desired. The group map will display the current locations of all active LTUs on a map identical to that used for the vehicle map.

For both maps, a calendar is displayed on the right side of the screen. This calendar provides the option to select a range of dates to display on the map. For example, if the user wishes to view the history of a certain LTU over the past week, the range of days can be specified on the calendar feature and the path of travel and other statistics will be shown for the given time frame.

At the bottom of both the vehicle and group map is a link called “Show Location Details.” Selecting this link will display a list of details on the current device or group of devices. The information displayed includes date, time, status, latitude, longitude, speed, heading, and address. The figure below shows an example of how the Vehicle Map page will appear to the user:

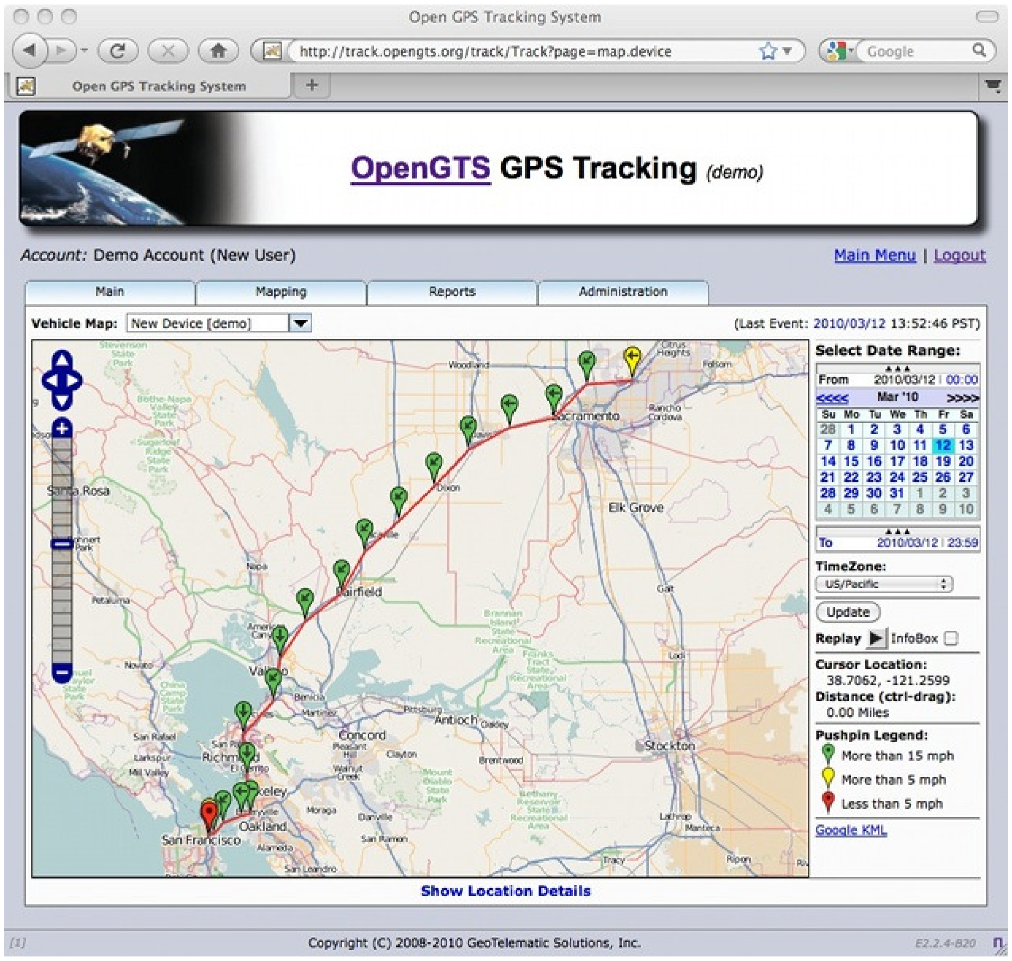

Open GPS Tracking System
http://track.opengts.org/track/Track?page=map.device
OpenGTS GPS Tracking (demo)
Account: Demo Account (New User)
Main Menu | Logout
Main
Mapping
Reports
Administration
Vehicle Map: New Device [demo]
(Last Event: 2010/03/12 13:52:46 PST)
Select Date Range:
From 2010/03/12 | 00:00
To 2010/03/12 | 23:59
TimeZone: US/Pacific
Update
Replay InfoBox
Cursor Location: 38.7062, -121.2599
Distance (ctrl-drag): 0.00 Miles
Pushpin Legend:
More than 15 mph
More than 5 mph
Less than 5 mph
Google KML
Show Location Details
Copyright (C) 2008-2010 GeoTelematic Solutions, Inc.

Figure 5-12 OpenGTS Web Portal Map Feature
Used with permission from GeoTelematic Solutions Inc., copyright 2012, all rights reserved.

The home screen of the web portal, if the user is not already logged in to an account, will be a login screen. The option will be given to either create a new account or login to an existing account. This feature provides the benefit of safety for users who wish to keep the location of their devices confidential.

5.6.3.2 Reports

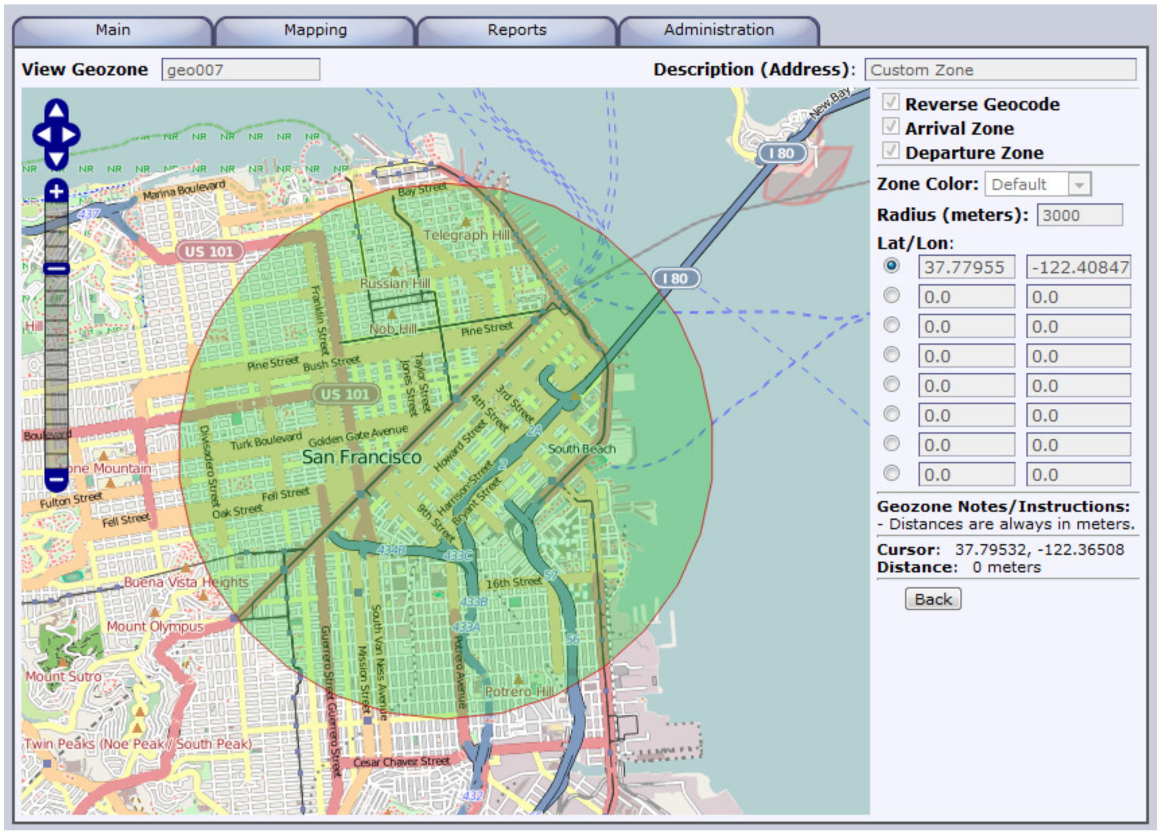
The reports option will feature its own menu at the top of the main screen. Four main reports will be offered: Device Detail, Group Detail, Group Summary, and Performance. Within each of these reports, specific report types can be selected to provide a greater level of specification. The report options are shown in the table below:

Table 5-4 LTU Report Summary

Report Type	**Available Reports**
Device Detail	- Event Detail - Property Values - Error/Diagnostic Values
Group Detail	- Event Detail (by group) - Speeds Over (speed specified by user)
Group Summary	- Last Known Luggage Location - Receive Event Counts
Performance	- Speeds Over (speed specified by user) - Moving/Stopped Time Summary

5.6.3.3 Geozones

Geozones are another main feature of the website and will provide a unique, customizable service for users. Found under the “Administration” menu, the “Geozone Admin” link will open a summary screen of all of the user’s current geozones. Geozones can be created with a variety of sizes and shapes, which provides tremendous flexibility for the user. The two main types of Geozones are Point Radius and Polygon. Point Radius is defined by a single point with a radius of coverage around it, both of which can be specified by the user. Figure 5-13 below provides an example of how the point radius geozone will appear to the user.



View Geozone geo007 **Description (Address):** Custom Zone

☑ **Reverse Geocode**
☑ **Arrival Zone**
☑ **Departure Zone**

Zone Color: Default
Radius (meters): 3000
Lat/Lon:

●	37.77955	-122.40847
○	0.0	0.0
○	0.0	0.0
○	0.0	0.0
○	0.0	0.0
○	0.0	0.0
○	0.0	0.0
○	0.0	0.0

Geozone Notes/Instructions:
- Distances are always in meters.

Cursor: 37.79532, -122.36508
Distance: 0 meters

Figure 5-13 Point Radius Geozone
Used with permission from GeoTelematic Solutions Inc., copyright 2012, all rights reserved.

As its name suggests, the polygon geozone can be implemented in virtually any two-dimensional shape. This is useful if, for example, the user wishes to specify an airport terminal as a geozone. If the terminal happens to be long and rectangular in shape, a point radius geozone may cover too much additional area when stretched to encompass the entire terminal. A polygon geozone, on the other hand, can be drawn around the exact borders of the terminal, thus providing a higher level of accuracy when an LTU enters or exits that geozone.

5.6.4 iOS Application

The iOS application is the last major feature that will be implemented on this project. This is mainly due to the fact that a smart phone application is not necessary for the functionality of the project. It is being implemented chiefly for convenience and marketability purposes. Another reason for the decision to implement this iOS application last is that it presents a high development risk, as none of our group members have experience developing smart phone applications.

That said, the iOS application will be an excellent feature for improving the user-friendly design of our product. The interface will take on a very clean, simplistic appearance with as little on-screen clutter as possible. The goal for this application, as it should be for virtually any smart phone application, is for the average person to easily navigate its features and menus for the first time. A well-designed, intuitive smart phone application should require no extensive learning or poking around by the user in order to operate it correctly.

With this design philosophy in mind, we plan to model our application after the design of the PocketFinder smart phone application. The PocketFinder software developers did an excellent job creating an application that both looks and functions as a user friendly interface. Their design features a clean layout with a small number of buttons per screen, yet each button clearly indicates which features or options it controls. This is the goal for our design as well.

One feature we plan to implement in the iOS application is the ability for the user to set geozones directly from their handheld device. Figure 5-14 shows an example of the geozone feature in the PocketFinder application.

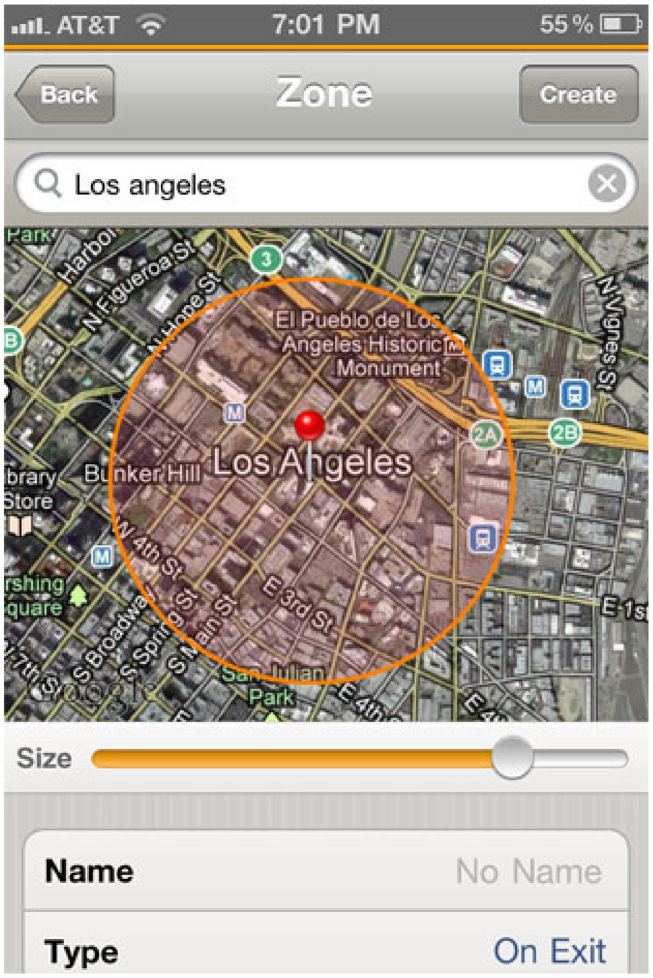


Figure 5-14 PocketFinder® Geozone Feature
Used with permission from Location Based Technologies, Inc.

In terms of appearance and simplicity, we plan to design our iOS application's geozone feature in a fashion similar to that of the PocketFinder application. The main issue with the implementation of this feature is the problem of transferring the user's input back to the GPS server. The geozone settings and specifications will need to be translated back to the GPS server and updated in the user's account. If this action is not taken, the settings in the smart phone application will not match the settings in the actual GPS server, and any alerts or updates will continue to be sent based on the settings stored in the user's account on the GPS server.

Another issue that must be considered is the transfer of settings and data not only from the application to the GPS server, but also from the GPS server to the application. If a user configures an alert, geozone, or other setting via the web portal, this information must be made available to the application immediately or at the very least upon startup or restart of the application.

5.7 Interface Control and Communication

5.7.1 GPS Satellite to LTU

In an ideal situation, the LTU would be constantly receiving signals from one or more GPS Satellites. However, in reality this is certainly not the case. It is entirely possible that at certain locations and for varying lengths of time, the LTU will not be receiving GPS satellite signals. This could be due to a dense building which blocks the signal, being in a remote or low-lying area with poor reception, or any number of other reasons. However, even if the LTU were to be in constant connection with GPS satellites, this would be an enormous drain on the power supply because the GPS receiver and microprocessor would be constantly operating and handling data.

In order for a GPS coordinate to be calculated, the LTU must be in range of at least four GPS satellites. This allows for the longitude and latitude to be determined based on the intersection of the four satellites' coverage areas.

The LTU will have a dedicated antenna to receive GPS satellite signals. It is yet to be determined whether this antenna will be internal or external to the LTU enclosure. The preferred option is to have the antenna inside the enclosure, but due to factors such as space and receptivity, the antenna may have to be placed outside the enclosure. As no member of our team has experience in antenna design, the GPS antenna will be purchased as an "off-the-shelf" part from a vendor.

5.7.2 Microprocessor to GPS Receiver

The chief function of the microprocessor in relation to the GPS receiver is to control the following actions:

- startup and shutdown
- power modes changes
- data collection

Because the GPS receiver is a major consumer of power, it is imperative that the GPS receiver is used only when necessary. When data is not being collected, the GPS receiver should be placed in the lowest reasonable power state. It is important to note that the lowest reasonable power state does not necessarily mean turning the receiver completely off. Depending on the type of receiver, the actions required for a full boot up, and the amount of time between reboots, it may actually require more power to perform a full shutdown and restart than to bring the receiver in and out of a low power mode.

Incoming GPS signals are processed internally by the GPS receiver, and the resulting data is made available on external data ports as well as directly to the GSM module via internal serial line. There are two external data ports on the GPS receiver, a modem serial port and a GPS serial port. These ports are described in detail in Table 5-5 below:

Table 5-5 GM862-GPS External Serial Ports

Port	Format	Data Rate (bps)	Description
Modem Serial	SiRF Binary	57,600	Control GPS and GSM via AT commands in a user-generated script.
GPS Serial	NMEA	4,800	Access NMEA sentences from GPS receiver.

It should be noted that while the Modem Serial port is external with respect to the GPS receiver, it is internal with respect to the GM862 chip itself. The Modem Serial port is, however, externally accessible on the GSM modem. A visual representation of these ports and their sources is provided in the figure below:

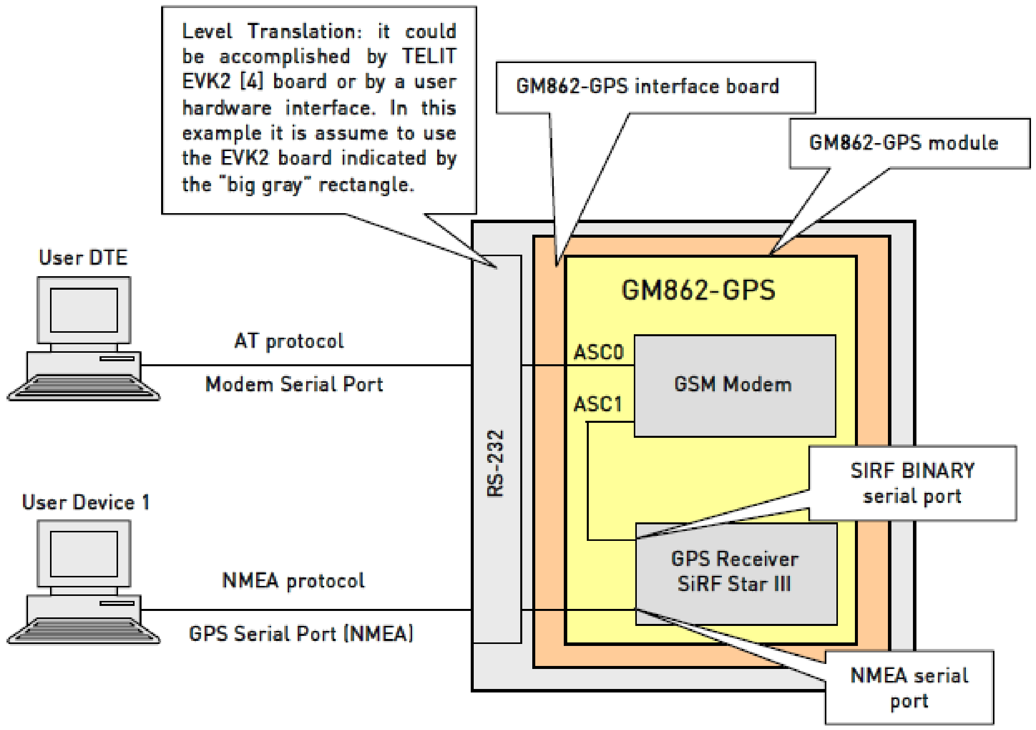

Level Translation: it could be accomplished by TELIT EVK2 [4] board or by a user hardware interface. In this example it is assume to use the EVK2 board indicated by the "big gray" rectangle.
GM862-GPS interface board
GM862-GPS module
User DTE
AT protocol
Modem Serial Port
User Device 1
NMEA protocol
GPS Serial Port (NMEA)
RS-232
GM862-GPS
ASC0
ASC1
GSM Modem
SIRF BINARY serial port
GPS Receiver SiRF Star III
NMEA serial port

Figure 5-15 GM862-GPS Serial Ports
Used with permission pending from Telit® Wireless Solutions

The outer gray rectangle in this figure represents Telit's EVK2 evaluation board. This board, or a similar evaluation board, will not be used in the final design, but may be used for test and integration purposes.

The primary mode of communication with the GE862 for this project will be the SiRF Binary protocol. This method implements the use of AT commands, which are used to control the GSM modem directly and the GPS receiver indirectly. Thus the microprocessor will send commands to the GPS receiver via the ASC1 serial line connecting the GSM module to the GPS receiver.

5.7.2.1 GPS Receiver Control

In order to control the GPS receiver, the GSM module's ASC1 port must be set to Controlled Mode. The AT command used to set the ASC1 port to Controlled Mode is AT$GPSD=<device type>. In this case, the device type is 2, so the actual command sent to the GPS receiver will be AT$GPSD=2. In order to activate control of the GPS receiver, this command must be sent first. Any AT commands sent to the GPS receiver before Controlled Mode is activated will not be received.

5.7.2.2 GPS Receiver Power

The AT$GPSP command is used to turn the GPS receiver on or off. Sending the command AT$GPSP=0 will power off the GPS receiver, and sending

AT$GPSP=1 will power on the GPS receiver. The GPS power command will, in general, only be used when the LTU itself is powered on or off by the user. When the LTU is on, the GPS receiver will typically be in either an active/ready state or a low power state. The power state of the GPS receiver will be determined by the settings programmed in the on-chip software designed by our team members.

For the majority of the time the LTU is powered on, the GPS receiver will be in a low power mode. However, at predetermined intervals, the receiver will be activated to a full power mode and allowed to search for and acquire GPS signals. Once a sufficient number of GPS signals have been acquired and a GPS coordinate has been calculated, the GPS receiver will be commanded back into a low power mode.

The low power mode that will be used for the GPS receiver is called Power Saving mode. The AT generic command for Power Saving mode is AT$GPSPS=<mode> [,<PTF_Period>]. The mode variable specifies what type of power mode to set the GPS receiver.

For this project, the default time between GPS fix attempts will be 10 minutes. Every 10 minutes, the GPS receiver will enter full power mode and attempt to acquire GPS satellites and generate a fix. Upon entering full power mode, the GPS receiver will be allowed 60 seconds to accomplish this task. If a GPS fix is not acquired within 60 seconds, the receiver will return to a low power mode.

The GPS receiver can be woken from a low power mode using one of two possible commands: the AT$GPSPS command or the AT$GPSWK command. Sending the command AT$GPSPS=0 will set the GPS receiver to full power mode. Sending the AT$GPSWK command will also set the GPS receiver to full power mode, but only if the current power mode is not full power. If the current power mode is zero (full power mode), sending the AT$GPSWK command will return the following error message: "+CME ERROR: operation not supported." Thus, the AT$GPSWK command should only be used when the GPS receiver is not in full power mode.

5.7.2.3 GPS Position Data

The most recent GPS position can be retrieved from the GPS receiver via the SiRF Binary port or the National Marine Electronics Association (NMEA) port. If using the SiRF Binary port, the AT$GPSACP command should be used. This command returns the latest GPS position as follows:

```
$GPSACP:<UTC>,<latitude>,<longitude>,<hdop>,<altitude>,<fix>,<cog>,<spkm
>, <spkn>,<date>,<nsat>
```

The variables in the GPSACP status message are described in detail in the table below:

Table 5-6 GPSACP Status Message Details

Variable	**Format/Values**	**Description**
UTC	hhmmss.sss	Universal Coordinated Time (UTC)
latitude	ddmm.mmmm	Latitude (d = degrees; m = minutes)
longitude	dddmm.mmmm	Longitude (d = degrees, m = minutes)
hdop	x.x	Horizontal Dilution of Precision
altitude	x.x	Altitude (in meters)
fix	0,2,3	0 = Invalid fix; 2 = 2D fix; 3 = 3D fix
cog	ddd.mm	Course Over Ground (d = degrees; m = minutes)
spkm	x.x	Speed over ground (in km/hr)
spkn	x.x	Speed over ground (in knots)
date	ddmmyy	Date of Fix (d = day; m = month; y = year)
nsat	nn	Total number of satellites in use

If using the NMEA port, the information provided by the GPSACP message above can be accessed, but this data must be pulled from a number of different NMEA output messages. By default, while the GPS receiver is in full power mode it continually broadcasts a four different messages over the NMEA serial port. These messages are GGA, GSA, GSV, and RMC.

Other status messages can be queried by invoking the $PSRF103 command. This command provides access to the GLL and VTG messages in addition to the four standard messages listed above. The data provided in the GLL and VTG messages is as shown in the table below:

Table 5-7 NMEA Output Message Content

Message Type	Message Content
GGA	- Time of fix - Position - Fix type
GSA	- Operating mode of GPS receiver - Satellites used - DOP values
GSV	- Number of GPS satellites in view - Satellite ID numbers - Elevation - Azimuth - SNR values
RMC	- Time - Date - Position - Course - Speed
GLL	- Latitude - Longitude - UTC time and status
VTG	- Course - Speed

A summary of the AT Commands used to control the GPS receiver are listed in the table below:

Table 5-8 GPS Receiver Commands

Command	Description	Variable Definition
AT$GPSD=<device_type>	Sets the ASC1 port to controlled mode.	<device type> 2 = Controlled Mode
AT$GPSP=<status>	Turns the GPS receiver on or off.	<status> 0 = Turn off GPS 1 = Turn on GPS
AT$GPSR=<reset_type>	Resets and restarts the GPS receiver.	<reset type> 0 = hardware reset 1 = cold start 2 = warm start 3 = hot start
AT$GPSRST	Restores the GPS factory settings.	N/A
AT$GPSACP	Reads the acquired GPS position.	N/A
AT$GPSPS=<mode> [,<PTF_Period>]	Puts the GPS receiver in Power Saving mode.	<mode> 0 = Full power mode (disable Power Saving mode) 1 = Tricklepower mode 2 = Push-to-fix mode 3 = N/A (GE864 only) <PTF_Period> (0-300000) # = number of seconds
AT$GPSWK	Brings the GPS receiver out of Power Saving mode and into full power mode.	N/A

5.7.3 Microprocessor to GSM Module

5.7.3.1 Power Modes

When the GM862 chip is powered on, the GSM module is turned on automatically. In order for the GM862 to be turned on, the ON pin must be tied low for a minimum of one second. When the low signal is released, the GM862 powers on.

To help reduce power consumption and extend battery life, the GM862 module comes equipped with several different power saving modes. The command used to control the power mode is the AT+CFUN command. This command provides the option of selecting a number of different power modes which can be used to mitigate the overall power consumption of the GSM module.

Calling AT+CFUN=0 will set the GSM module to non-cyclic sleep mode. This mode is the lowest power mode available without turning the GSM module completely off. It provides only minimum functionality, and it is important to note that even the AT command interface is shut down in this mode. The only way for the mode to be exited and the GSM module brought back to full power mode is through the occurrence of a wake-up event or by the receipt of a high RTS signal. Either of these two events will bring the unit out of the non-cyclic sleep mode and into full power mode. Because the non-cyclic sleep mode offers the smallest power consumption, this mode will be activated when the GSM module is not attempting to transmit a message. This means the GSM module will be non-cyclic sleep mode for 15-minute intervals, coming back to full power mode after each interval to attempt to transmit a message or multiple messages.

Calling AT+CFUN=1 will set the GSM module to full functionality and will also disable power saving. When power saving is disabled, the module does not reduce its power usage during the time it is idle. This mode is the default setting of the GM862 module. While this mode provides full functionality, it should only be entered when necessary because it is also the mode with the highest rate of power consumption.

Calling AT+CFUN=2 will disable the transmitting capability of the GSM module. In other words, the module will only be able to receive incoming signals and messages while in this mode. For this project, the sole purpose of the GSM module is to transmit messages, so this mode will not be utilized.

Calling AT+CFUN=4 will disable both the transmitting and receiving capabilities of the GSM module. This mode may be used to deregister the module from a network or to deactivate the SIM card. While this will not be a planned occurrence during the normal operation of the LTU, this mode may be implemented as a troubleshooting measure if errors occur with the SIM card or a cellular network.

Calling AT+CFUN=5 will bring the GSM module into a full power mode identical to that of setting number 1 with the exception of the power saving capability. Contrary to mode 1, mode 5 enables the power saving option. This means the GSM module will go into a low power state to reduce power consumption when it is idle. When the module is not idle, it is in its full power state with all functionality available. This mode will be the mode of choice when the GSM module is brought out of non-cyclic sleep mode to transmit messages. Because there may be some idle time while messages are prepared for sending or other processes are occurring, the module can go into a low power state if necessary before sending the messages in the send buffer and returning to non-cyclic sleep mode.

Calling AT+CFUN=7 will activate the cyclic sleep mode. This mode is similar to the non-cyclic sleep mode, but the main difference between the two is that, in cyclic sleep mode, the AT command interface remains active. This allows the GSM module to be awakened with an AT command rather than waiting for a wake-up event or a high RTS signal. Another unique feature of the cyclic sleep mode is that it allows the GSM module to sense when data is being transmitted on the serial interface. As long as the serial interface is in use, the GSM module will remain active. Cyclic sleep mode can only be deactivated by sending the AT+CFUN=1 command, which brings the GSM module to the full power mode. From there, other power modes may be selected as desired.

Calling AT+CFUN=9 will put the GSM module in a mode identical to that of mode zero with the exception of the behavior when a GPRS packet is received by the module. When in mode zero, an incoming GPRS packet event will not have any effect on the power mode. However, when in mode 9, an incoming GPRS packet event will bring the GSM module out of its current power mode and into mode 1, which is the full power mode with power saving disabled. Mode 9 will most likely not be used in this project as the GSM module should not be receiving any GPRS packets at any point during its operation cycle.

The GM862 can be shut down using either the AT#SHDN command or by tying the ON pin low. As with the power on procedure, the ON pin must be tied low for at least one second and then released in order to initiate the shutdown procedure. This command disconnects the GSM module from any networks and then powers off the module.

5.7.3.2 Band Selection

The GSM module is a quad-band modem, meaning it can operate on four different bands: 850/1900 and 900/1800. Rather than attempting to manually connect to a specific cellular band, the GSM module will be allowed to automatically select a band. This is accomplished using the following command: AT#AUTOBND=<value>. To disable the automatic band selection, send the following command: AT#AUTOBND=0.

5.7.3.3 SIM Phonebook

A critical function of the GSM module is to provide access to both its internal phonebook and the phonebook stored on the SIM card. The phone numbers to which text message alerts are to be sent must be stored either on the SIM card or in the internal memory of GSM module itself. This section will discuss the commands used to access the SIM card installed in the GSM module.

To add a phonebook entry to the SIM card phonebook, the SIM phonebook must first be selected by entering the command AT+CPBS=”SM”. Next, the new entry can be added to the phonebook using the AT+CPBW command. The phone number can be stored in either a national or international format. This format is specified in the “type” variable. Additional information on these two signals is provided in Table 5-9. The GSM module should return the “OK” message if the new entry was stored successfully.

5.7.3.4 Storing SMS Messages

Because a cellular network may not be available every time the GSM module wishes to send a text message, all text messages will be stored in memory before attempting to send. Doing this will ensure that messages are not lost when a network is not available, but instead messages will be ready for transmission when a network is found and a connection is established.

The command used to store an SMS message in memory is the AT+CMGW command. Once a message is stored in memory, it can be sent using the AT+CMSS command. This command allows for a specific message to be loaded from memory and sent to a given destination.

5.7.3.5 SMS Messaging via GSM

The GSM module offers two formats for working with SMS messages: Packet Data Unit (PDU) mode and Text mode. For this project, the Text mode will be used exclusively as it is much more user-friendly and does not require the learning of a new syntax, which would be necessary for the PDU mode to be used. In order to set the module to Text mode, the AT+CMGF command must be used. Sending the command AT+CMGF=1 will set the GSM module to Text mode.

Before this method of sending an SMS message is implemented, the following settings must be the current settings of the GSM module:

- AT#SELINT=2
- AT#SMSMODE=1

To send an SMS message, use the AT+CMGS command. This command alerts the network that a message is going to be sent and specifies both the type of destination address and the destination address itself. After this command is sent, the device should reply back with a standard message signifying it is ready for the actual SMS message to be sent.

At this point, depending on the current settings of the GSM module, the actual content of the SMS message should be entered in one of two formats. The first format, and the format that will be used most frequently in this project, is the more traditional of the two formats. This format allows characters to be entered as they would normally be entered by a user. Upon being entered, the characters are converted into the GSM alphabet by the module’s internal software. If the second format is used, the message content must be entered as IRA character long hexadecimal numbers. These numbers are then internally converted into 8-bit octets by the GSM module. This format will most likely not be implemented in this project.

By using the AT+CSMP command, the current SMS format setting can be both viewed and changed as desired. This command has four different variables, but the variables that control the SMS format are the “fo” and “dcs” variables. This command and its variables are described in more detail in Table 5-9.

The GSM module provides the capability to send up to 10 concatenated SMS messages. This feature is handled automatically by the module. An error message will be returned if more than 10 messages are attempted to be sent as a concatenated message thread.

5.7.3.6 SMS Messaging via GPRS

The primary means of communication between the GSM modem and the GPS server will be through the GPRS data service. Although GPRS is technically part of the GSM system, for the purposes of this project they are considered to be separate services. Because GPRS communication is IP based, data can be transmitted to any entity with an IP address. In this case, the entity is our GPS server.

In order to activate GPRS messaging capability, the SMS mode must be set to 1 using the command AT#SMSMODE=1. In addition, the AT command interface must be set to 2 by sending the command AT#SELINT=2. Next, the GSM module must be set to transmit via GPRS rather than GSM. The module’s default setting is to transmit via GSM only. The AT+CGSMS command allows this setting to be changed to GPRS only, GPRS preferred, or GSM preferred. The use of the term “preferred” signifies that if that particular service is not available, the alternate service will be used to transmit the SMS message. So for example, if GPRS preferred is chosen, the module will transmit data via GPRS unless that service is not available, in which case it will transmit via GSM.

After ensuring the GSM module is in text mode by sending the command AT+CMGF=1, the message can be sent using the AT+CMGS command. This command allows the destination and content of the message to be specified. Further details are provided in Table 5-9.

A summary of the AT Commands used to control the GSM module are listed in table below:

Table 5-9 GSM Module Commands

Command	**Description**	**Variable Definition**
AT#AUTOBND=<value>	Controls automatic band selection.	<value> (0-2) 0 = disable automatic band selection 1 = enable automatic band selection (at next power-up) 2 = enable automatic band selection (immediately)
AT#SHDN	Initiates shutdown procedure.	N/A
AT#SLED=<mode>, <on_duration>, <off_duration>	Sets the function of the status LED GPIO pin.	<mode> 0 = tied low 1 = tied high 2 = controlled by module 3 = turned off and on intermittently <on_duration> (mode = 3) *integer* = number in tenths of seconds that pin is tied high <off_duration> (mode = 3) *integer* = number in tenths of seconds that pin is tied low
AT#SMSMODE=<mode>		
AT+CFUN=<fun>	Sets the power mode.	<fun> 0 = non-cycle sleep mode 1 = full power mode (power saving disabled) 2 = transmit disabled 4 = transmit and receive disabled 5 = full power mode (power saving enabled)
AT+CGSMS=<service>	Sets the type of service for transmitting SMS messages.	<service> 0 = GPRS 1 = GSM 2 = GPRS preferred 3 = GSM preferred
AT+CMGF=<mode>	Controls the SMS format.	<mode> 0 = PDU mode 1 = Text mode

AT+CMGS=<da>,<toda>	Sends an SMS message via GPRS	<da> *string* = destination address <toda> 129 = national number 145 = international number
AT+CMGW=<da>,<toda>, <stat>	Stores an SMS message in memory.	<da> *string* = destination address <toda> 129 = national number 145 = international number <stat> “REC UNREAD” “REC READ” “STO UNSENT” “STO SENT”
AT+CMSS=<index>,<da>, <toda>	Sends an SMS message from memory.	<index> *string* = location of message in memory <da> *string* = destination address <toda> 129 = national number 145 = international number
AT+CPBS=<storage>	Selects which phonebook storage to access.	<storage> “SM” = SIM phonebook “FD” = SIM fixed-dialing phonebook “LD” = SIM last-dialed phonebook “MC” = SIM missed calls phonebook “RC” = SIM received calls phonebook
AT+CPBW=<index>, <number>,<type>,<text>	Writes an entry to the SIM phonebook.	<index> *string* = record position <number> *string* = phone number <type> 129 = national number 145 = international number <text> *string* = text ID

AT+CSMP=<fo>,<vp>, <pid>,<dcs>	Sets parameters and for SMS messages	<fo> *integer* = first octet of 3GPP TS 23.040 SMS <vp> *integer* = validity period <pid> *integer* = 3GPP TS 23.040 TP Protocol Identifier <dcs> *integer* = data coding scheme

5.7.4 Microprocessor to Memory

The microprocessor will serve as the control mechanism and data router for the purposes of the on-chip memory. The main purpose of the memory component is to serve as a GPS data log. While GPS fix data will be stored in the GPS module’s internal memory, it will also be relayed to the microprocessor, which will in turn store that data on the memory chip. The capacity of the memory chip is significantly greater than the internal memory of the GPS module, thus providing a backup storage space should the GPS module’s memory reach capacity. While this should not occur during normal operation of the unit, it is possible for this to occur if there is a malfunction in the GSM transceiver or if the LTU is in a location without GSM network coverage for an extended period of time.

The language used to control communication between the microprocessor and memory chip will be C. While the actual code for this reading and writing process has yet to be developed, it is anticipated that the total amount of code will be very concise and relatively simple in nature. The microprocessor will be alerted any time a GPS module obtains a new GPS fix. The software controlling the microprocessor will retrieve the GPS data from the GPS module and store it in memory. Data should not need to be retrieved from the memory chip unless there is a serious malfunction or failure within the GSM/GPS module. One possible scenario would be an error in the GSM/GPS module’s internal memory resulting in its data being erased. If this occurs, the data stored on the memory chip may be used to restore the lost data. In the worst case scenario, the GSM/GPS module will need to be replaced with a new one. In this situation, the GPS data stored on the memory chip would be retrieved by the microprocessor and sent to the new GSM module for transmission.

In addition to storing GPS coordinates and data, the memory chip will also store the software on which the microprocessor runs. This code will be developed by the design team, but its actual implementation will be handled by the microprocessor itself. The microprocessor will know at which locations in memory the embedded software is stored and will run this software when the LTU is powered on.

5.7.5 GPS/GSM Module to LED Indicator

The GPS/GSM module used in this project provides a GPIO pin which can be used for a GSM status LED. One of the design requirements for this project is to provide an external indication of the current network connection, or lack thereof. This can be accomplished through the use of this GPIO pin.

The function of this pin is controlled by the AT#SLED command. While several different functions are available for this pin, the default setting for this pin will be utilized as it provides a simple yet informative signal pattern for the LED. The default setting for the status LED output signals is as shown in the table below:

Table 5-10 Status LED Signals

Device Status	LED Status
Off	Off
Searching / Not registered / Shutting down	Blinking (every 1 second)
Connected and registered	Blinking (every 3 seconds)
Call in progress	On (solid)

It should be noted that, because the GSM module will not be used in this project to make calls, the status LED should never reach the solid “On” state. After changing the status LED setting, the new setting must be saved using the #SLEDSAV command. This command saves the status LED setting to the nonvolatile memory (NVM).

5.7.6 LTU to GPS Server

The LTU will communicate with the GPS server through the GPRS service. GPRS is a packet oriented service that allows data to be sent to a device such as a server or other internet-based terminal. The primary command used to transmit data from the GSM/GPS module will be the AT+CMGS message. This command will be used every 15 minutes when the GSM module is brought into full power mode and the GPS data messages in the transmission buffer are sent. The content of each message will conform to a predefined standard and will include the following data: LTU ID, time of fix, longitude, latitude, altitude, and velocity.

With the Phase A prototype these messages will only be sent directly to a cellular phone number or email address, as the GPS server will not yet be integrated into the system. With a Phase B prototype, however, the GPS server will be integrated and all messages will be sent to the server. On the server end of the communication, incoming messages will be processed and immediately distributed according to settings specified by the user. The data in the messages will also be used by the server to populate maps and generate reports.

5.7.7 GPS Server to Cell Phone

The GPS server will interface with the user's cellular phone in two ways: via the iOS application (if the user owns an iPhone) and via SMS text messages. These two methods of communication are discussed at length in the following sections.

5.7.7.1 iOS Application

The GPS server will interface with the iOS application by publishing LTU data to the application. This content will be made available via internet connection on the user's smartphone, which can be accessed through a Wi-Fi connection or through a 3G or 4G connection. The software in the iOS application will access the data stored in the GPS server and populate a map accordingly. The user will be able to select one or more LTUs whose locations will be displayed on the map. The user will also be able to create and/or remove geozones through the iOS application. Because this data is ultimately stored on the GPS server, the user's selections must be sent from the smartphone to the GPS server for the server to be updated.

5.7.7.2 Text Messages

Once the GPS server has been integrated into the luggage tracking system, SMS text message alerts may be sent to the user's cellular phone according to the settings stored on the server. Text message alert content will be generated in part from data transmitted by LTUs and, in part, by the GPS server itself. Transmitting text messages from the GPS server will be accomplished using GPRS communication.

5.7.8 GPS Server to Personal Computer

The GPS server will interface with the user's personal computer in two main ways: via the OpenGTS web portal and email alerts. The web portal method of communication is based solely on the action taken by the user to login from their computer and access information through the web portal. Emails will be sent to the user's personal computer (PC) based on the alert settings chosen by the user. The content of these emails will be generated, in part, from data transmitted by LTUs and, in part, by the GPS server itself. They will contain information such as LTU coordinates, time, and velocity. They will also contain a statement detailing the reason for the email alert being sent, such as an LTU leaving or entering a geozone. The interaction between the GPS server and personal computer does not involve any interaction beyond internet-based communication and interfacing and does not contain any elements that need to be designed in the scope of this project. Thus they will not be discussed further at this time.

5.7.9 GPS Server to Web Portal

Because the web portal is developed in conjunction with the OpenGTS GPS server, the interface between the two is handled internally by the OpenGTS software. For a diagram of the OpenGTS architecture and the web portal's place within that architecture, see Figure 5-11. The features and applications of the web portal itself are discussed in depth in section 5.6.3.

6 Design Summary

6.1 Overview

The overall function of our system is as follows. The GPS receiver within the LTU will receive a GPS signal from the network of GPS satellites. The GPS receiver will process this signal and generate GPS coordinates and other relevant data such as time, velocity, and altitude. This data will be sent via the GSM module from the LTU to the GPS server. The GPS server will receive this data, process it, and distribute updates to the user according to predefined settings as emails and text messages. The GPS server will also publish LTU data to the web portal for viewing by the user. Figure 6-1 shows a top level functional flow of the luggage tracking system.

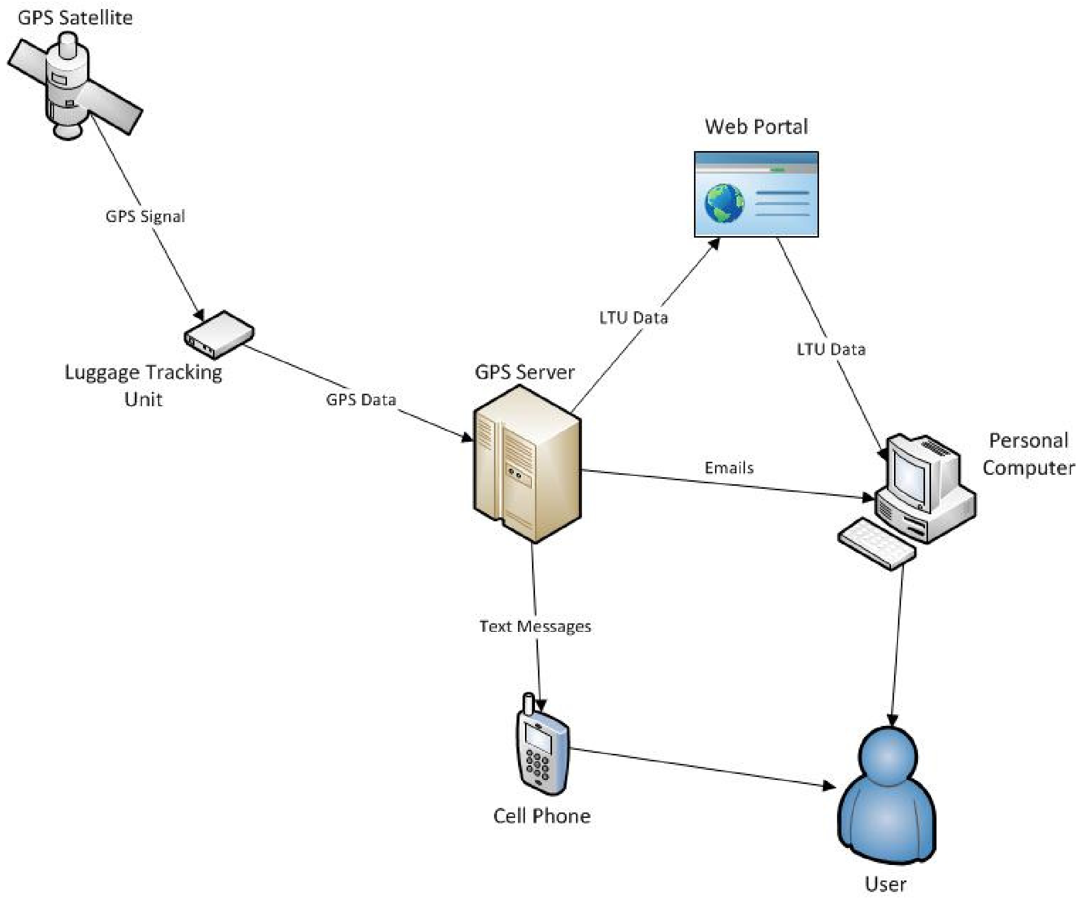


Figure 6-1 Luggage Tracking System

From the time it is powered on, the LTU will remain in a continuous operation cycle until it is powered off by the user. For the majority of the time the LTU is powered on, the internal communication components, namely the GSM module and GPS receiver, will be kept in a low power state in order to minimize power consumption and extended the battery life as much as possible.

After it is powered on, the LTU will first acquire a GPS fix if a sufficient number of GPS satellites are in range. If a fix is successfully acquired, the data will be stored in memory where it will be compiled into a message to be transmitted by the GSM module. If a GPS fix is not successfully obtained, the LTU will return to a low power mode for fifteen minutes before trying again.

The actual generation of the status message will be handled by the microprocessor. The microprocessor will assemble the content of the message based on a predetermined format, add the necessary destination information such as a cellular phone number or IP address, and place the message in the transmit buffer to be sent by the GSM module. Next, the GSM module will attempt to connect to a GSM network if one is available. If the module is successful in connecting to a network, it will send any messages stored in the transmit buffer. Once all messages in the buffer have been sent, the LTU will return to a lower power mode. After fifteen minutes have passed, the LTU will reenter full power mode and begin its operation cycle again. The overall function cycle of the LTU is captured in the activity diagram in the figure below:

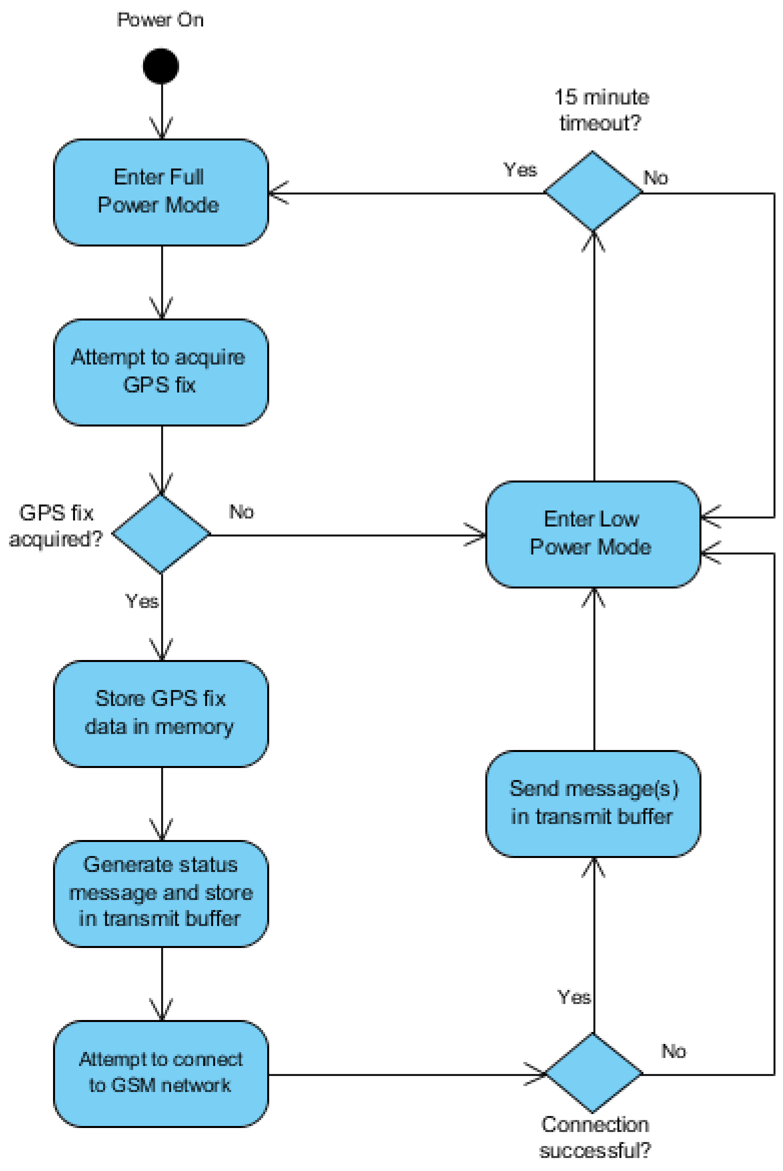

Power On
Enter Full Power Mode
Attempt to acquire GPS fix
GPS fix acquired?
No
Yes
Store GPS fix data in memory
Generate status message and store in transmit buffer
Attempt to connect to GSM network
Connection successful?
Yes
No
Send message(s) in transmit buffer
Enter Low Power Mode
15 minute timeout?
Yes
No

Figure 6-2 Top Level Activity Diagram

6.2 Hardware Summary

6.2.1 Electrical Schematics

The following figures show the electrical diagrams for power source management via charger, charge controller, battery, input diagram for on/off power, reset, photoelectric switch, barometric measurement, output diagram for network status and GPS status, and serial communication for the microcontroller and memory.

6.2.1.1 Charger Power Source Diagram

Figure 6-3 shows the electrical schematic for connections to the power source. A zener diode has been placed in order to protect the load from incorrect reverse voltages to the input as well as to protect against short circuits. Also a 0.1µF has been included in order to satisfy any transient high current demands from the module. A charger controller has been included in order to satisfy correct charging requirements for Li-Ion batteries. A current limiting circuit is also included in order to maintain charging current to 0.2 Amps.

6.2.1.2 Battery Power Source Diagram

Figure 6-5 shows the electrical schematic for connections to the battery. Since Ni/Cd or Ni/MH battery types can be over-charged resulting in the voltage surpassing the allowed maximum for the GM862, a Lithium Ion rechargeable battery has been selected in order to guarantee voltage stability of 3.7 V. Also, a 1000mAh battery capacity has been selected in order to satisfy any potential surge requirements of the GM862 module when simultaneously handling multiple processes (GPS, GSM, logging, etc). Also a 100µF has been included in order to satisfy any transient high current demands from the module.

6.2.1.3 Input Diagram

Error! Reference source not found. shows input signals for on/off, reset and ambient light sensor. On/off and reset input pins are internally pulled up, and they operate when forcing a low signal. The push buttons are connected to an inverter via an NPN transistor. This provides a low signal with low current draw. For the on/off operation the low level signal must be maintained more than 1 second. For the reset operation the low level signal must be maintained for at least 0.2 seconds.

6.2.1.4 Output Diagram

Figure 6-6 shows output signals for network status and availability. The green LED is connected in series with a 470Ω resistor. Since these output pins are open collectors, and the output voltage is the inverse of the signal status, the connection is pulled up to VBATT 3.7 V.

6.2.1.5 Microcontroller, GM862 and Memory SPI Serial Diagram

The microcontroller MSP430 will be able to request and collect GPS data from the GM862, add a time stamp and log into the flash memory via universal serial communication interface using SPI mode. This serial bus is shown in Figure 6-7. The microcontroller will function as a Master for all communications. The GM862 and the flash memory will be set as Slaves. During transmission through the SPI serial lines, the control signal CS1 will enable the GM862 to operate and capture all data. For the flash memory, the CS2 will be enabled by the microcontroller in order to upload data or retrieve data. See Figure 6-7.

It is important to highlight that the data rate for communication to the flash memory has to be less than 66 MHz maximum clock frequency. Since the flash memory has the slowest clock frequency requirement in comparison to the GM862, the recommended clock frequency for all communications will be set between 50MHz and 60MHz. This master clock will be generated and handled by the MSP430 microprocessor.

6.2.1.6 Barometric Pressure Sensor I2C Serial Diagram

The barometric pressure sensor shown in Figure 6-7 will be connected via I^2C serial communication to the microcontroller. Any signal over the threshold limit for 3000 meters will interrupt the microcontroller which in turns will disconnect the GM862 from the network and deactivate the GPS or any other RF working at the moment. Moreover, the microcontroller will be put into low power state and monitor the barometric sensor. Once the barometric sensor drops below the threshold, the microcontroller will wake up the GM862 and resume operations with the GPS.

6.2.1.7 Microcontroller Input and Output

Figure 6-8 shows the light sensor input and LTU power status LED. A photoelectric sensor or ambient light sensor has been included in order to trigger a signal in case the luggage is opened during transit. The sensor utilizes I^2C serial communication providing logarithmic values of light waves between 500nm and 600nm. This signal will be monitored by the microcontroller. Once a preset threshold is reached, the microcontroller will generate a time stamp of the event with a GPS location that can be logged into memory. A yellow LED has been added to one of the outputs of the microcontroller in order to indicate when the LTU is powered on.

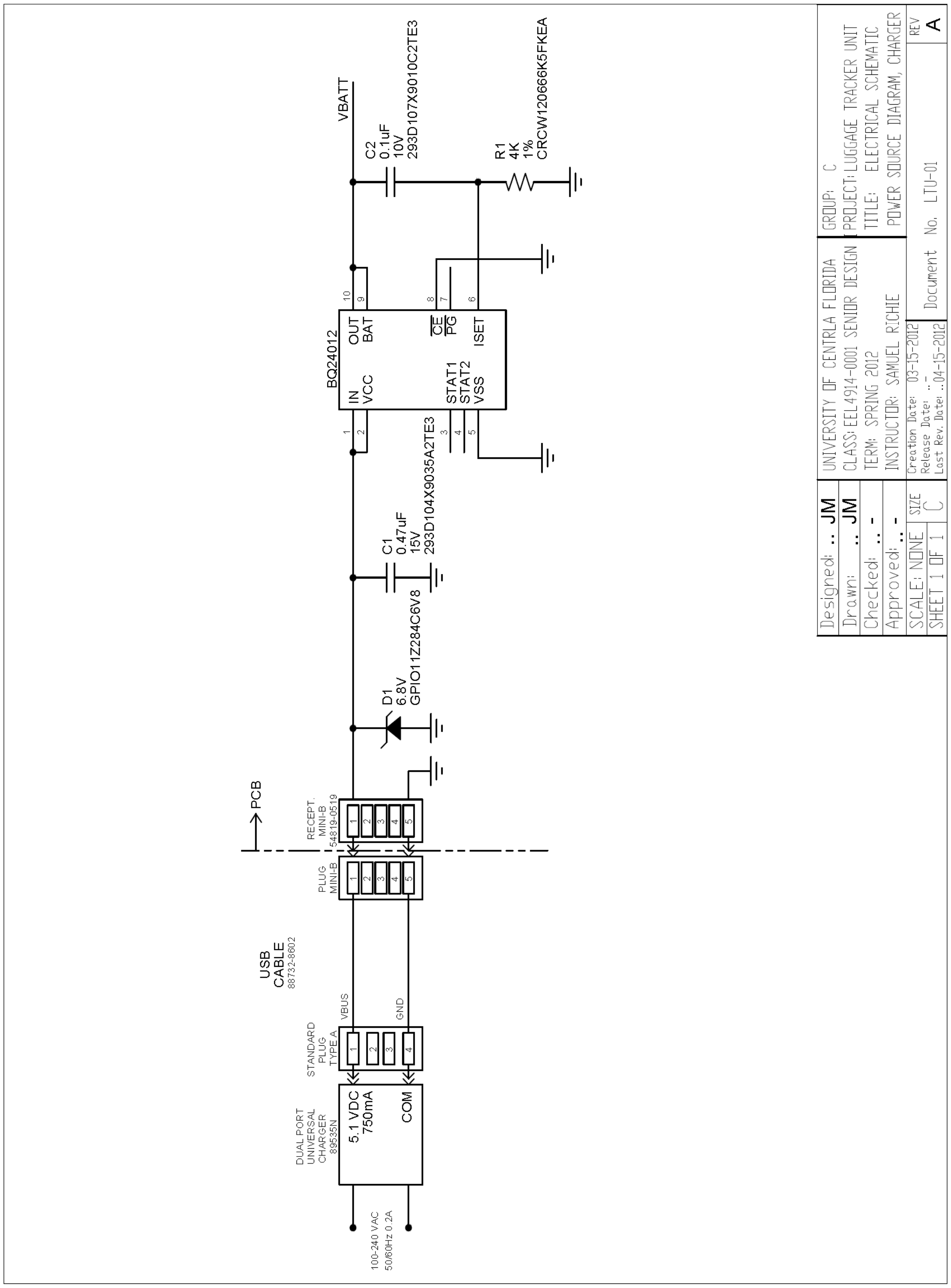

100-240 VAC
50/60Hz 0.2A
DUAL PORT UNIVERSAL CHARGER 89535N
5.1 VDC 750mA
COM
STANDARD PLUG TYPE A
VBUS
GND
USB CABLE 88732-8602
PLUG MINI-B
RECEPT. MINI-B 54819-0519
PCB
D1 6.8V GPIO11Z284C6V8
C1 0.47uF 15V 293D104X9035A2TE3
BQ24012
IN
VCC
STAT1
STAT2
VSS
OUT
BAT
CE
PG
ISET
VBATT
C2 0.1uF 10V 293D107X9010C2TE3
R1 4K 1% CRCW120666K5FKEA
Designed: .. JM
Drawn: .. JM
Checked: .. -
Approved: .. -
SCALE: NONE
SHEET 1 OF 1
SIZE C
UNIVERSITY OF CENTRLA FLORIDA
CLASS: EEL4914-0001 SENIOR DESIGN
TERM: SPRING 2012
INSTRUCTOR: SAMUEL RICHIE
Creation Date: 03-15-2012
Release Date: .. -
Last Rev. Date: ..04-15-2012
GROUP: C
PROJECT: LUGGAGE TRACKER UNIT
TITLE: ELECTRICAL SCHEMATIC
POWER SOURCE DIAGRAM, CHARGER
Document No. LTU-01
REV A

Figure 6-3 Charger Power Source Diagram

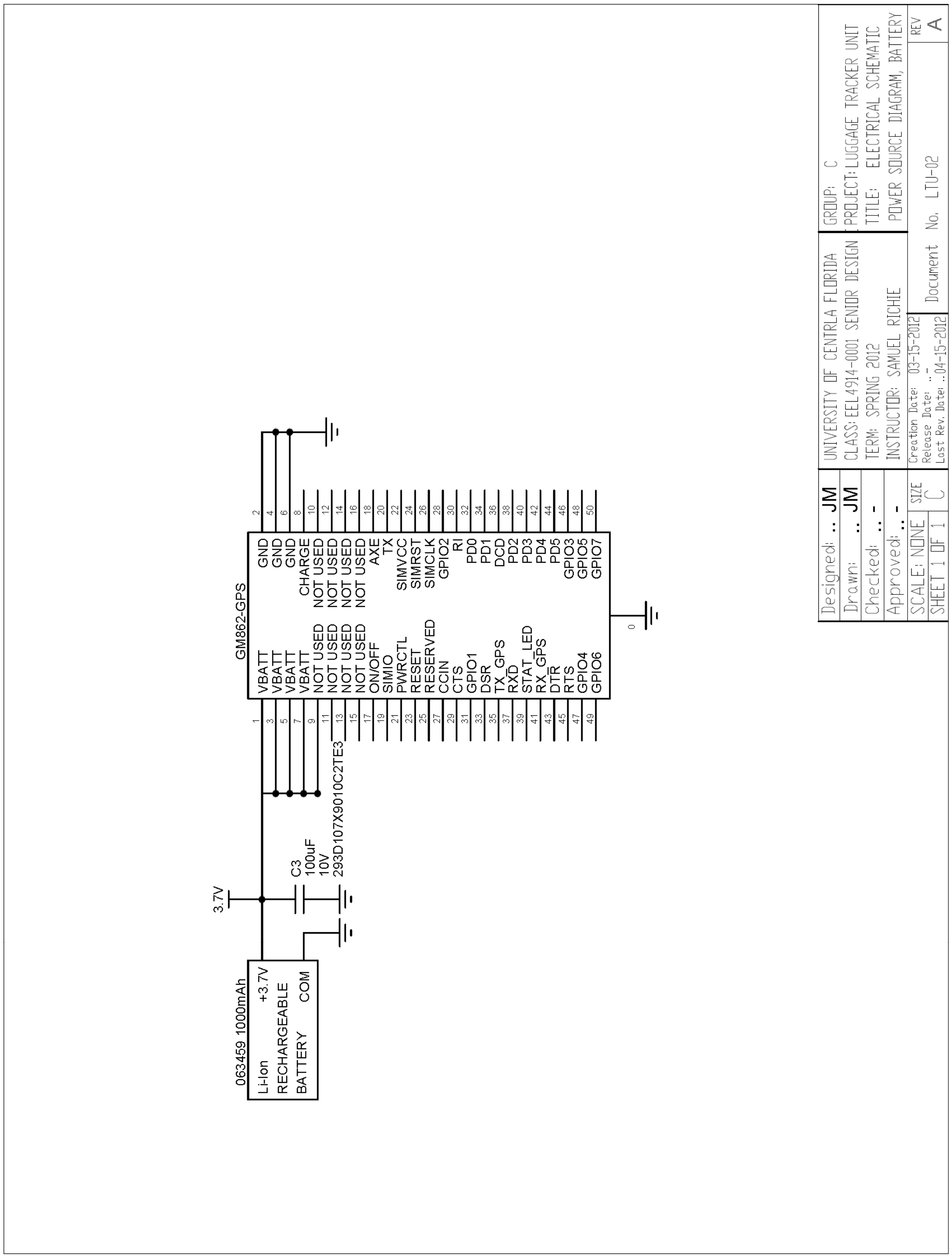


Figure 6-4 Battery Power Source Diagram

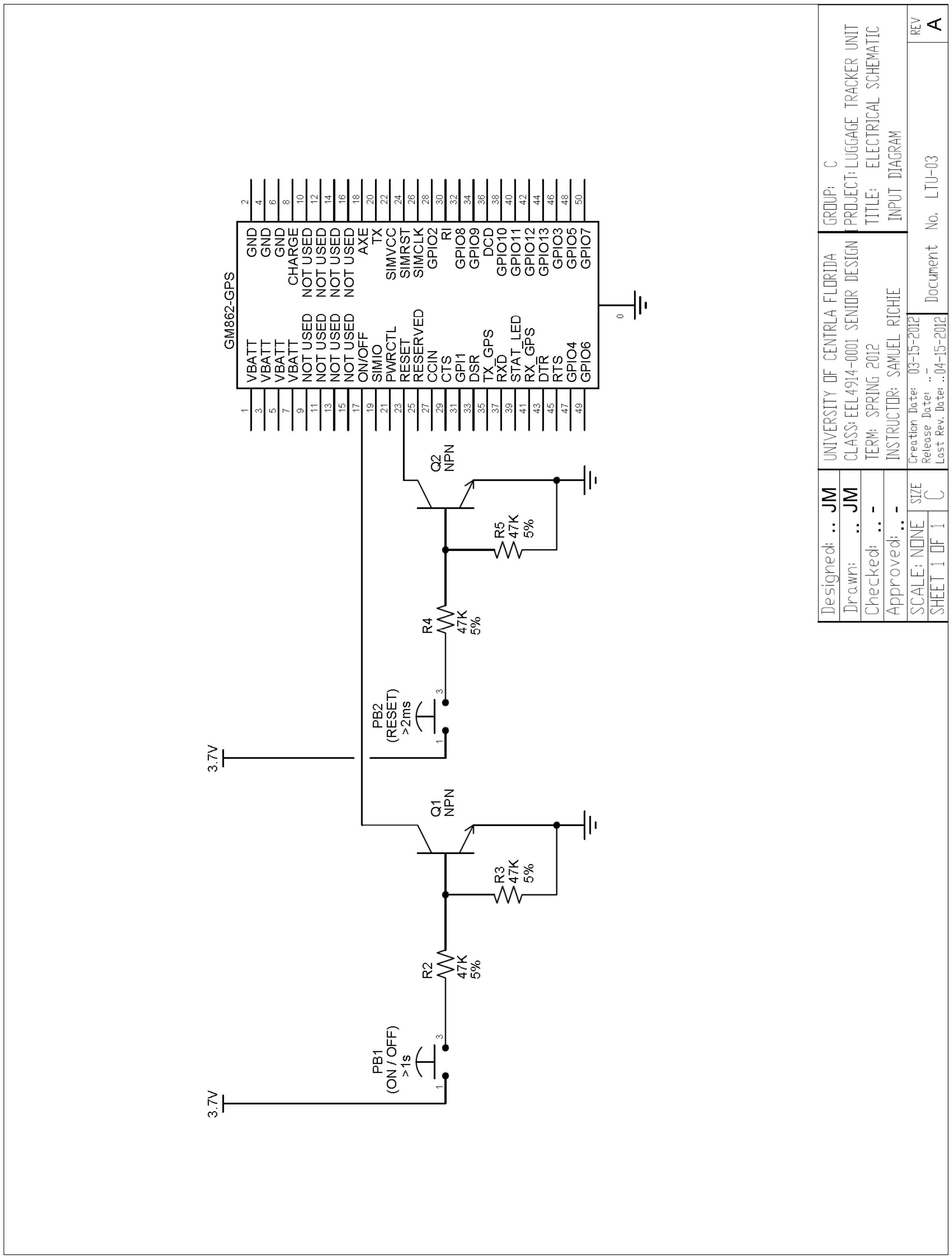


Figure 6-5 Input Diagram

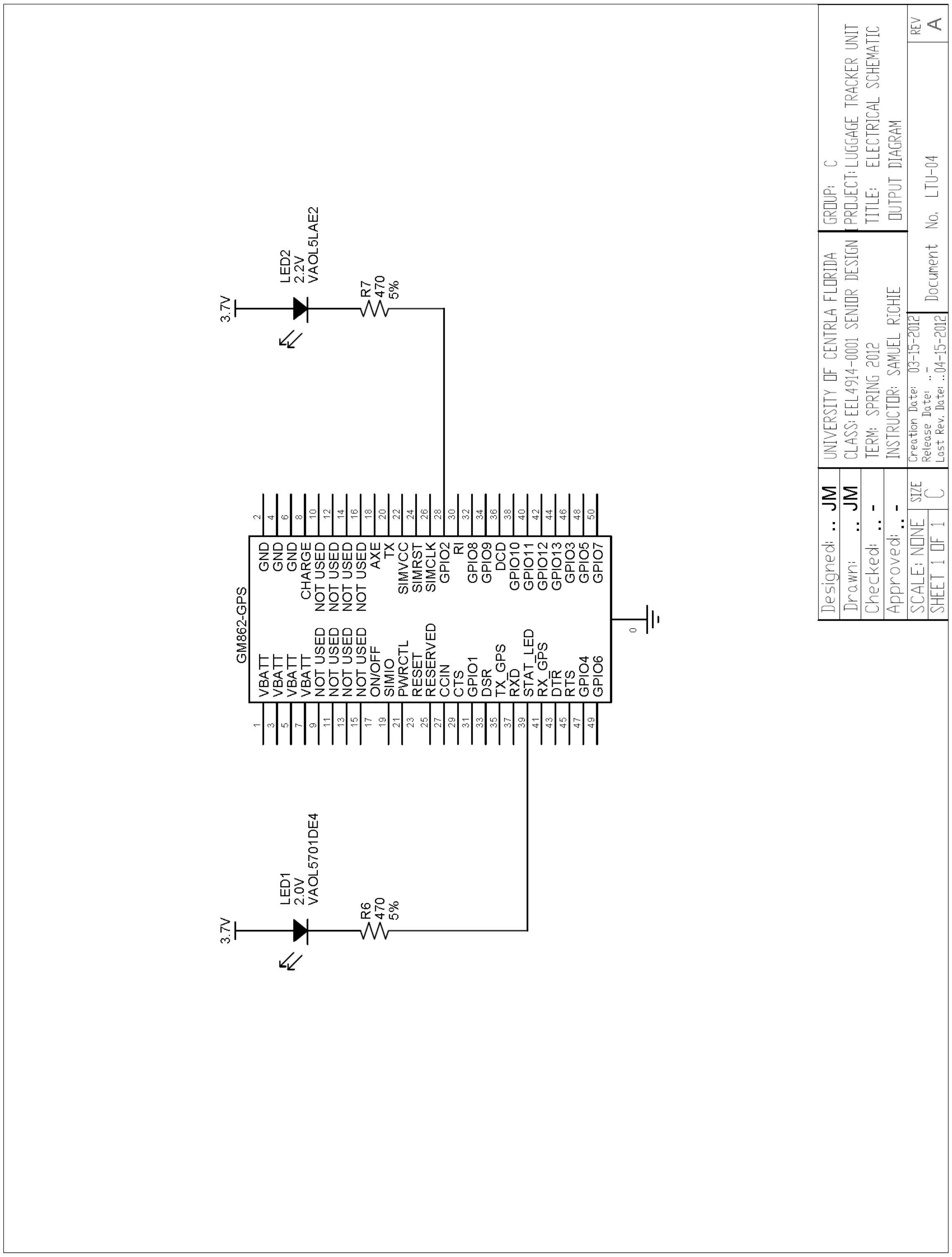


Figure 6-6 Output Diagram

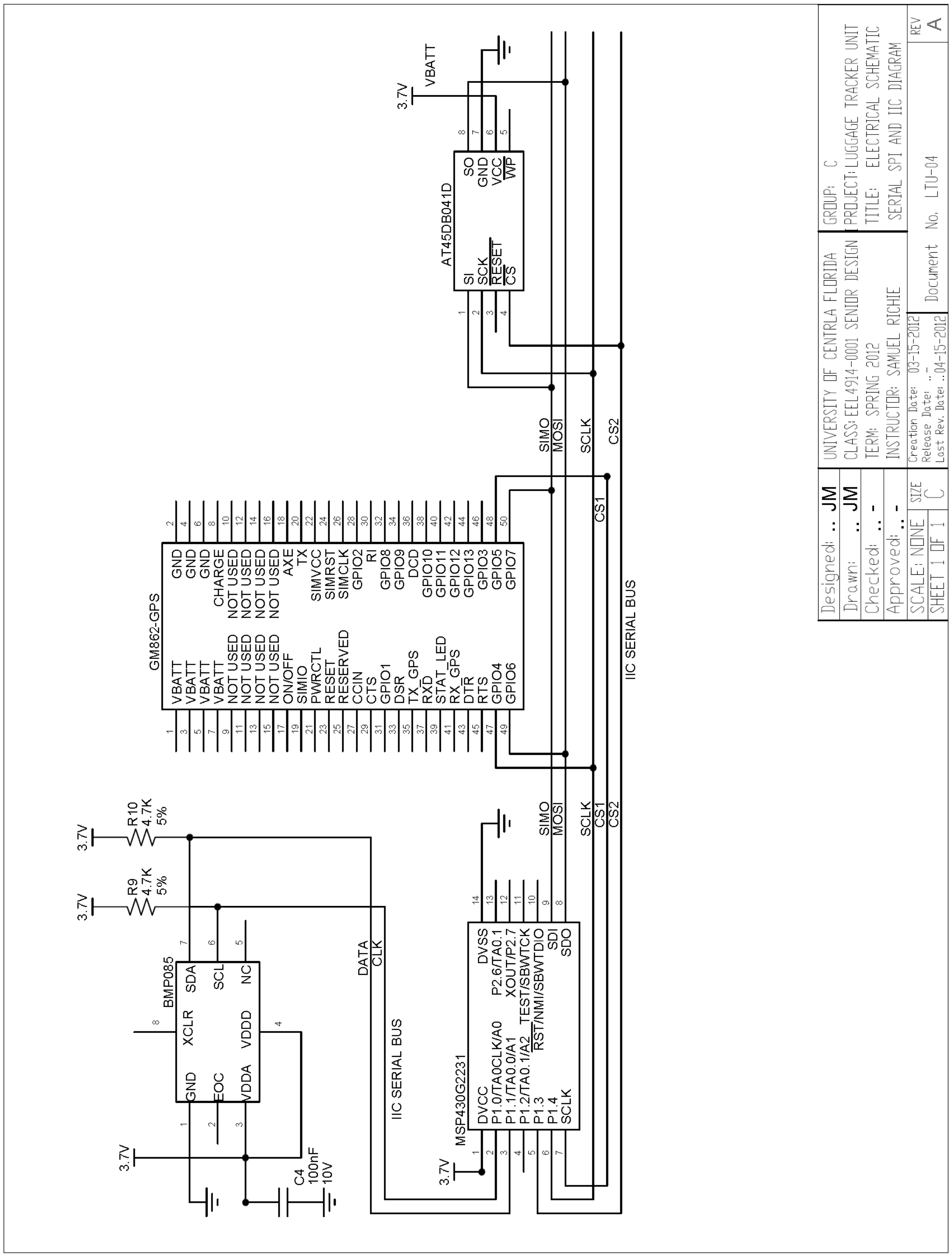


Figure 6-7 Microcontroller, GM862, Memory SPI Serial Diagram and Barometric Pressure Sensor I^2C Serial Diagram

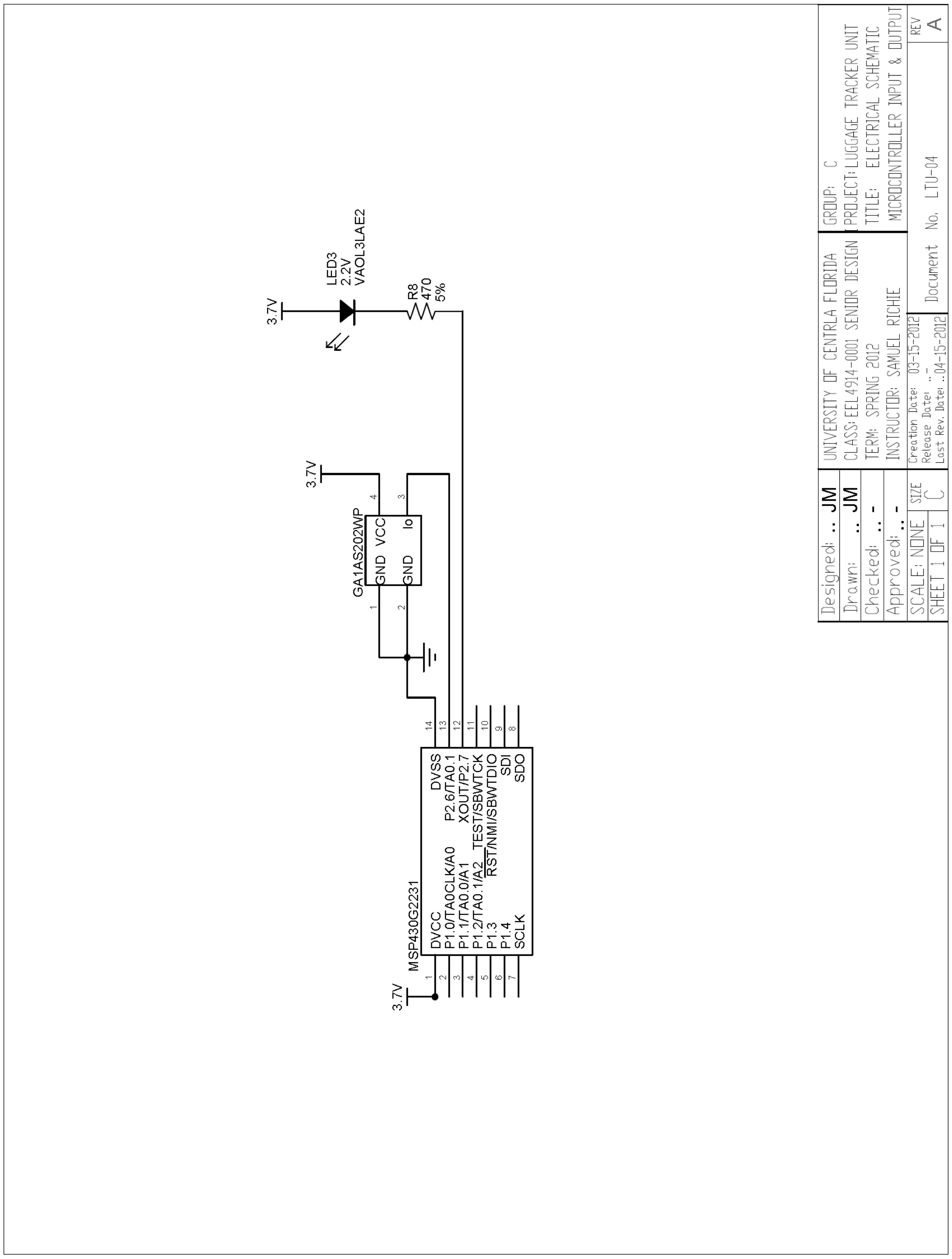


Figure 6-8 Microcontroller Input and Output

6.3 Software Summary

The software used to control the Luggage Tracker Unit consists of multiple software modules carefully integrated to allow direct communication between the components. Because of the complexity of the software for the Luggage Tracker Unit, it can be better understood when broken down into specific modules. At the top level, the overall functionality of the LTU's software depends on the proper functionality of its underlying software modules. These underlying software modules can be broken up into two basic categories: on-chip software and iOS application.

The on-chip software is designed to directly control the communication between the LTU and the GPS server. The on-chip software will receive and process data from the GPS module. The GPS coordinates will then be processed into a text message and transmitted to the GPS server via the GSM module. Thus, the on-chip software's main objective is to successfully transmit information concerning the LTU's current position while also maintaining communication lines between the multiple modules within the LTU.

The iOS application will serve as a user-friendly means of communicating with the LTU. The iOS application will directly communicate with the LTU to allow the user to update user setting and preferences. The application will also serve as a type of notification center for the LTU. That is, the iOS application will receive current GPS coordinates, usage data, and alerts from specified LTU(s). The iOS application will communicate with the LTU via the on-chip software. Therefore, the interface between the iOS application and the on-chip software must be reliable and efficient.

To better understand the high-level functionality of the software used to control the LTU, a class diagram is provided below:

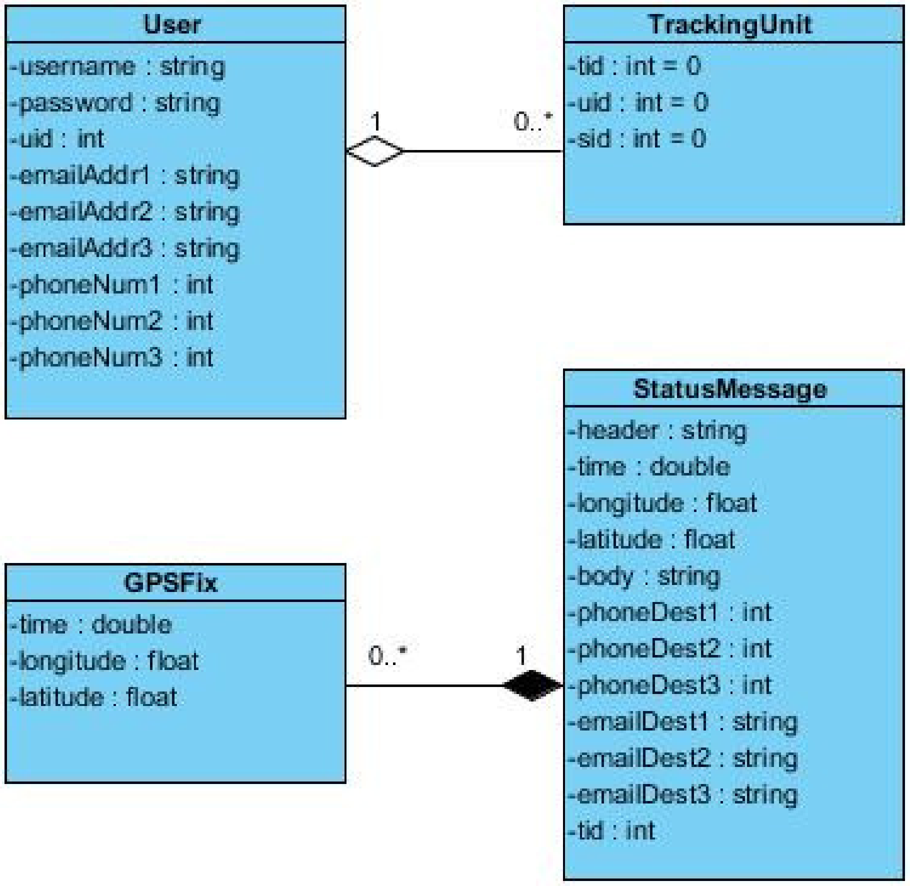


Figure 6-9 LTU Class Diagram

The LTU class diagram contains four basic classes: User, TrackingUnit, GPSFix, and StatusMessage. The User class is responsible for storing and verifying the user's information. Because each user can have many LTUs, there is a one-to-many relationship between the User class and the TrackingUnit class. The TrackingUnit class is responsible for storing the identification numbers associated with each individual LTU. The GPSFix class contains the information associated with each LTU's current location, while the StatusMessage class is responsible for formatting each alert in the form of an email or SMS text message. There is a one-to-many relationship between the StatusMessage class and the GPSFix class because there can be multiple sets of GPS coordinates associated with each status message (i.e. the user is tracking multiple LTUs).

The User class will be used to store the basic information of the user in a single User object. The username and password will allow the user to login to the system to directly connect to the LTU(s) associated with their account. The uid (user identification) field will contain the user's identification number, which will be directly linked to each of the user's LTU(s). There are three fields for the user's email addresses and phone numbers. These fields will allow the user to be verified in multiple ways, using numerous email addresses and phone numbers as notification destinations.

The TrackingUnit class assigns each LTU a unique set of identification numbers. An exclusive tid (tracking unit identification) will be assigned to each LTU and used to identify it. The uid (user identification) will be used to associate each LTU will a specific user, as to not confuse alerts and notifications with other users. The sid (status message identification) field will be used to track each status message to ensure it is sent to the proper LTU. This number will also be used to track the status message's timestamp so the LTU's location information can be retrieved by tracing the status messages.

The GPSFix class contains all the necessary information to correctly identify the location of an LTU at a specified time. The time parameter will be used to log the time once a request is sent to determine the location of an LTU. The longitude and latitude fields will then be populated with the correct GPS coordinate information based on the LTU's current position. The objects created by this class will be associated with the alert and status messages sent to the user to provide the current location(s) of their LTU(s).

The StatusMessage class will be used as a blueprint to format each alert message. Once the alert has been completely formatted, it will be processed and sent as an email and/or SMS text message to the destinations specified by the user, or sent to the GPS server for distribution. The header will serve as a title field in text format. The time field will contain the time at which the alert message was processed and sent. The longitude and latitude fields will contain the current GPS coordinates of the LTU at the time the alert message was sent. The body of the alert message will contain important information regarding the LTU in text format. The three emails and three phone numbers will be the destination(s) of each alert message, while the tid (tracking unit identification) field will notify the user as to which LTU the alert message is referring.

7 Prototype Construction

7.1 Bill of Materials

Line #	Item	Qty	Description	Manufacturer	Vendor	Part Number	Unit Cost	Lead Time
1	MOD1	1	GM862-GPS	Telit	Semiconductor Store.com	GM862-GPS V.00 (GM862GPS730-021/ SW#07.03.X00)	$108.33	In stock
2	AN1	1	GSM ANTENNA, 5" long, swivel, MMCX	Nearson, Inc	Digi-Key	S181FL-5-RMM-2450S	$18.75	In stock
3	AN2	1	GPS ANTENNA, 5" long, MMCX	V.Torch	Sparkfun.com	VTGPSIA-3	$11.95	In stock
4	BATT	1	BATTERY Li-ION, 3.7V, 1000mAh	Unionfortune	Sparkfun.com	063459 1000mAh	$11.95	In stock
5	CON1	1	BOARD TO BOARD CONNECTOR 50 PIN	MOLEX	MOUSER	501920-5001	$6.64	In stock
6	CON2	1	USB MINI-B RECEPTACLE	MOLEX	MOUSER	54819-0519	$1.50	In stock
7	CABLE	1	CABLE ASSEMBLY USB A / MINI-B 1 M	MOLEX	MOUSER	88732-8602	$3.52	In stock
8	CON3	1	JST VERTICAL PH BATTERY CONNECTOR	JST	Sparkfun.com	B2B-PH-K-S / PRT-08613	$0.95	In stock
9	CG	1	DUAL PORT UNIVERSAL CHARGER	Motorola	Motorola	89535N	$29.99	In stock
10	CASE	1	PROJECT CASE	Sparkfun	Sparkfun.com	WIG-08601 black WIG-08632 clear	$9.95 $11.95	In stock
11	C1	1	0.1µF Tantalum Capacitor, 35V	VISHAY/SPRAGUE	MOUSER	293D104X9035A2TE3	$0.67	In stock
12	C2, C3	2	100µF Tantalum Capacitor, 10V	VISHAY/SPRAGUE	MOUSER	293D107X9010C2TE3	$1.06	In stock
13	C4	1	100nF Tantalum Capacitor, 10V	VISHAY/SPRAGUE	MOUSER	293D102X9010C2TE3	$0.54	In stock
14	D1	1	Zener Diode 500mW, 6.8V	DIODES INC.	MOUSER	PD3Z284C6V8-7	$0.50	In stock
15	R1	1	66.5KΩ 1% Resistor, 1/4W	VISHAY/DALE	MOUSER	CRCW120666K5FKEA	$0.10	In stock
16	R2, R3, R4, R5	4	47KΩ 5% Resistor, 1/4W	VISHAY/DALE	MOUSER	CRCW060347K0JNEAHP	$0.19	In stock
17	R9, R10	2	4.7KΩ 5% Resistor, 1/4W	VISHAY/DALE	MOUSER	CRCW060347R0JNEAHP	$0.19	In stock
18	R6, R7, R8	2	470Ω 5% Resistor, 1/4W	VISHAY/DALE	MOUSER	CRCW0603470RJNEAHP	$0.19	In stock
19	Q1, Q2	2	NPN BJT, VCEO=45V, VEBO=6V, GAINCB=200, 2Ma, 5V	DIODES INC.	MOUSER	BC847BW-7-F	$0.40	In stock
20	LED1	1	LED GREEN 2.0V, 100mcd, 5mm	VCC	MOUSER	VAOL-5701DE4	$0.16	In stock
21	LED2	1	LED RED 2.2V, 100mcd, 5mm	VCC	MOUSER	VAOL-5LAE2	$0.15	In stock

22	LED3	1	LED YELLOW 2.2V, 100mcd, 5mm	VCC	MOUSER	VAOL-3LAE2	$0.15	In stock
23	PB1	1	Tactile & Jog Switch NO momentary	TE Connectivity / Alcoswitch	MOUSER	FSM104	$0.85	In stock
24	PB2	1	Tactile & Jog Switch NO momentary	ALPS	MOUSER	SKHRAAA010	$0.54	In stock
25	PCB	1	4"x4" PCB Board and population services	Advanced Circuits	Advanced Circuits	Custom Order	$33 + shipping	Lead Time 7 Days
26	MC	1	Microcontroller, 16-bit RISC CPU, 16-bit registers, ten I/O pins, 10-bit A/D converter and built-in communication capability using synchronous protocols (SPI or I2C).	TI	MOUSER	MSP430G2231IN14	$1.62	3 Days
27	MCS	1	14 pin IC socket	TE Connectivity	MOUSER	1825094-3	$0.93	In stock
28	Mem	1	Memory. Flash 4M 8 I/O Pin RapidS SPI 264B 2.5V 2.7V	Atmel	MOUSER	AT45DB041D-SU	$0.92	In stock
29	LS	1	Surface Mount Light Detector (555nm) 3.2V, 1mA	Sharp Microelectronics	MOUSER	GA1A1S202WP	$0.91	In stock
30	BPS	1	Digital Barometric Pressure Sensor. I^{2}C Serial Bus	BOSCH	Digi-Key	BMP085	$8.19	In stock
31	EVAL KIT	1	GM862 Evaluation Kit - USB w/GM862-GPS	Telit	Sparkfun	KIT-00280	$199.95	In stock
32	DEV KIT	1	TI Development Kit for MSP430	TI	TI	MSP-EXP430G2	$4.30	In stock

7.2 Procurement

7.2.1 Hardware

7.2.1.1 Prototype Components

The majority of the components will be purchased from vendors. Some of the vendors that we will utilize are as follows: Mouser Electronics, Digi-Key, Motorola, Sparkfun, and Semiconductor Store. These vendors are distributors for the major manufacturers of electronic components.

7.2.1.2 Prototype PCB

The PCB will be ordered through a board house called Advanced Circuits located in Aurora, CO and Tempe, AZ. Since the budget has been set for two LTU, the first unit or PCB will be ordered as soon as the design phase is complete. The second unit will be ordered after basic testing has been done to the first PCB. This will reveal any flaws not foreseen during the design phase that can be corrected. It will also reveal other potential improvements that can be incorporated into the second PCB unit.

The standard lead time for an order of a 2-Layer design PCB with less than 60 sq. inches is about 7 days, not including shipping time. A total of two weeks will be used as a standard for ordering the PCB units. This will allow ample time for any special component that will need to be sent to the board house.

7.2.2 Software

7.2.2.1 PCB Design Software

Two possible PCB layout software platforms have been identified in order to generate the Gerber files. The first one is a free software provided by the board house, Advanced Circuits. The PCB Artist software can be downloaded on any computer running Microsoft Windows® OS. This software package includes a library of over 270,000 parts and auto-routing tool, which selects the most effective route to interconnect terminals and components. These connections are made via tracks which are segments of a conductor. Free software provided by Advanced Circuits is called design file check (DFC). Once the Gerber files are generated, a DFC can be done in order to identify any potential problems related to physical connections of the components and files issues.

The second PCB layout software is called EAGLE PCB. It is provided by CadSoft Computer. The EAGLE PCB software offers a schematic editor, a layout editor and autorouter. The software also provides a library editor that allows for creation or custom adjustment of components. Another feature is the ability to create a bill of material and check pricing and availability of components in addition to the automatic creation of a bill of material. Moreover, an order can be completed entirely through the selected board house, so there is no need to purchase items before sending them to the board house.

7.2.2.2 On-Chip Software and iOS Application

The on-chip software will be developed in the Python language, which is supported by the Telit module. Therefore, there will be no additional steps necessary to specifically procure the on-chip software aside from obtaining the Telit module. However, to develop an iOS application, a developer's license is necessary. Each license costs $99 per year and gives the user access to XCode and other tools necessary to begin development of an iOS application. Upon completion of development, the application must be submitted to Apple's App Store for review. Once the application has been approved, it can be made available for download through Apple's App Store.

7.3 Prototype Assembly

Once the PCB unit has returned from the boarding house, interfacing with the case will be necessary. Part of the design of the board will include holes on the board that will be used to screw the PCB to the case. The case will then be modified in order to allow for the on/off push button to protrude through one of the sides. For the reset push button, a small diameter window will allow the user to insert the tip of a pen. We have to remember that the reset function only has to be used as a last resort for removing power and cycling the LTU. Another two small holes will be added for LEDs. These last two holes can be avoided if a translucent case with the desired dimensions can be found.

The GSM antenna will be installed through the side of the case. The antenna will be fastened with a nut provided with the bulkhead of the swivel antenna. The final routing and bending of the antenna cable around the PCB and GM862 will be done, and the MCCX plug will be connected to the SMA of the GM862.

8 Testing

8.1 Master Test Plan

The master test plan shall ensure that the system functions properly and efficiently, as originally designed. The ultimate goal of the testing is to eliminate all errors from the system. More specifically, errors that stem from the integration of the hardware and software components should be exposed in the master test plan. From the perspective of low-level testing, the hardware and software test plans shall seek to expose the errors within their respective regions and the modules associated with those regions. The low-level functionality of the system will be tested with more precise tests geared toward specific modules. However, the master test plan shall focus on the high-level functionality of the system, rather than specific, low-lying problems.

8.1.1 Simulation Test Environment

In order to avoid as many issues as possible in the build and prototype phase of development, we will attempt to simulate as many aspects of our design in a software simulation program before actually implementing them. Simulation provides the benefit of being able to experiment with an idea or design before spending the time, energy, and money to actually integrate that component into the product. This application of simulation applies to both hardware and software as individual components, and can even be applied to hardware and software simultaneously.

A complete hardware/software co-design and simulation will be implemented using SystemC. SystemC is a C++ based software that provides the ability to model a system's hardware and software components, as well as their interfaces, all in one software program. This program will rigorously test the general functionality of the system. The simulation will verify the basic functionality of the unit, excluding the influence of environmental factors. The simulation test environment shall establish benchmarks to help direct and define the real-world test environment. Once the simulation testing has been completed, the physical prototype will undergo much of the same testing in a real-world environment.

8.1.2 Real-World Test Environment

In order to yield realistic outcomes when testing the LTU, the test environment will be setup so that it resembles a real-world setting in which the system will operate. First, the basic functionality of the device will be tested to ensure the device is working properly under ideal conditions. If the results from the real-world test environment resemble those found in the simulated test environment, further testing will commence to test the robustness of the device under non-ideal conditions.

At the time of writing this document, there have been no set limitations or constraints on the potential destinations or locations of the unit. Therefore, it is assumed that the LTU will experience many environmental changes throughout its various journeys. The environmental changes can be, and will likely be, erratic. The device will continually experience rapid changes in altitude while in flight. The unit may also experience rapid changes in temperature and humidity when traveling internationally or to remote locations.

Since the environments in which the device will be taken are somewhat unpredictable, general environmental factors will be taken into consideration when defining the test environments. That is, the device will be tested by exposing it to significant changes in altitude, pressure, humidity, temperature, and many other environmental factors to determine which conditions allow for ideal operation of the unit. The test environments will also be setup to verify the environmental limits on the LTU. The environmental factors that can be controlled will be the focus when defining test case environments, while the uncontrollable factors will be considered but not altered. The defined testing environments shall seek to test each environmental factor individually as well as holistically. It should be noted that when forming the test environments, the specified limits of the modules will not be deliberately exceeded in order to ensure the safety of the prototype.

8.1.2.1 Testing Environmental Factors Individually

When testing individual environmental factors, only one environmental variable will be allowed to fluctuate. All other aspects will be controlled such that they experience no significant or erratic changes. For example, the device will be tested with varying temperatures while all other factors remain controlled and unchanged. Once the data is collected on the responsiveness and functionality of the system under different temperatures, the temperature will be fixed, along with all other factors, and another individual factor can be tested. This process of individually testing each environmental factor will continue until all aspects have been thoroughly tested and unambiguous data has been collected.

8.1.2.2 Testing Environmental Factors Holistically

Once the individual environmental factors have been tested, a holistic test shall take place. This testing environment shall test multiple factors in one, universal test environment. This testing environment shall encompass varying changes in a majority of the environmental factors such that it resembles a real-world scenario. This testing environmental will help form a conclusion on whether the device will successfully operate in extreme conditions with erratic changes in the environment.

8.1.3 Altitude Testing

The most important environmental factor to consider in our system is altitude. Since the primary use of the Luggage Tracker Unit will be on distant travels, the unit will likely spend the majority of its operation time on an airplane. Thus, the system must be able to undergo erratic and significant changes in altitude and still function properly. The system must also be able to handle the pressure changes that come with variations in altitude. Variations in pressure can cause the physical components of the unit to respond differently, resulting in invalid operation.

Testing variations in altitude in a real-world scenario proves to be quite a challenge. Most means of getting to a high altitude are expensive and time consuming. Thus, being able to test the system at a typical operating altitude is not feasible. Therefore, a software simulation will be used to mimic changes in altitude. That is, the system will simply be programmed to function as if it is at a certain altitude. Using this simulation method tests certain aspects of the unit that are dependent on altitude without physically having the device at that altitude. For example, to test the power saving functionality of the device, a high altitude will be hardcoded into the LTU, and the device will be checked to confirm that it entered a power saving mode.

Although software simulation can be used to generally test the system, some form of physically testing variations in altitude must be performed to properly test the robustness of the device. To physically simulate slight variations in altitude, a helium balloon will be used to elevate the LTU into the air. The system will be elevated to predetermined, specified heights and tested individually before readjusting the altitude. Testing and reconfiguring the system in this manner shall continue until the system can be said to successfully operate at various altitudes.

8.1.4 SIM Card Assessment

In order to holistically test and demonstrate the Luggage Tracker Unit's full functionality, a valid SIM card must be procured to send SMS text messages. An AT&T SIM card will be used for this portion of the testing. Before testing, the SIM card will be verified to have a valid phone and data plan. This will ensure that SMS text messages can be reliably sent using the Luggage Tracker Unit's GSM module. Because multiple LTUs can be used simultaneously to track multiple pieces of luggage, the system must also be tested with multiple LTUs, each with their own SIM card.

8.1.4.1 Verification of Single SIM Card

To ensure overall functionality of the communication aspect of the system, a single LTU, containing an AT&T SIM card, will be tested. First, the basic functionality of the system will be tested to ensure the GSM module is properly sending and receiving data via the SIM card. Individual test cases will be designed to test the transmission of all necessary data required to efficiently communicate with the GPS server. Once these individual test cases validate the proper functionality of the SIM card in conjunction with the GSM module, an overarching test will be conducted to test all aspects of the system that utilize the GSM module and SIM card to send and receive essential data.

8.1.4.2 Verification of Multiple SIM Cards

Once a single LTU is shown to successfully transmit data using an AT&T SIM card, multiple LTUs will be tested simultaneously. It should be noted that the SIM cards will each have their own cellular plan. That is, although the AT&T SIM cards may be on the same cellular plan, the cards will have their own unique contact number, and they will be able to operate independent of each other. Each LTU will contain its own SIM card, and the test cases will be designed to specifically address the problem of operating multiple units simultaneously.

Specifically, the communication of each LTU with the GPS server will be examined. The GPS coordinates shall be sent to each of the LTUs simultaneously to ensure the communication with each LTU is stable, despite noise contributed by the other. Once each LTU processes the coordinates, the SMS text messages, sent by each of the GSM modules, will be verified to contain the correct information regarding the LTU's current position. Once the test has shown that two LTUs work properly while operating simultaneously, it can be assumed that operating more than two LTUs, each with their individual SIM cards, can also operate simultaneously. If time permits, the testing process will continue to verify this assumption.

8.1.5 Stopping Criteria

Throughout testing, errors will inevitably be encountered. A predetermined system for handling those errors must be in place prior to testing. A plan to know when to stop testing and either announce success or go back to development shall be established. However, it should be noted that the stopping criteria can and will vary across test cases. Thus, the specific test case should be evaluated in depth before a stopping criteria is agreed upon by the group.

8.1.5.1 Error Discovery

There are two actions that can be taken when errors are encountered during testing:

1. Immediately stop the testing when the error is detected and attempt to fix that problem before continuing.
2. Continue testing and record all of the encountered errors in a sequential manner. If a fatal error is encountered when practicing this method, stop testing and attempt to solve the errors encountered up to that point.

The method used when testing will be discussed amongst the group members prior to executing the specified test. Depending on the components and modules being tested, the appropriate testing procedure will be decided on. Once the majority of the errors have been detected, a new testing procedure will be introduced. This testing procedure will consist of testing the LTU for a specified amount of time to ensure proper functionality of all systems under time intensive conditions.

When performing time-specific testing, a group meeting will be held prior to testing to determine the amount of time to test the basic functionality of the system. Throughout testing, errors will be recorded, much like the method described in (2) above. Once the time limit has been reached, testing will be halted and a team meeting will be held to determine how to address the encountered errors. If the group decides to address the errors immediately before testing, the system will be taken out of the testing environment, and diagnoses of the errors will begin. After the errors have dealt with, testing will commence in the same manner. However, if the group decides that the errors encountered are not fatal, an additional testing time limit will be determined and testing will continue until the time limit has been reached.

8.1.5.2 Error Handling

Once the errors have been detected in the system, a solution to properly eliminate the errors will be discussed amongst the group. The approach will vary depending on the location and severity of the errors. If the error is found to be in any of the hardware modules, the system will be taken to the lab and diagnoses will begin to pinpoint the error. An attempt to immediately eliminate the error will then ensue. Conversely, if the errors are found to be within any of the software components of the system, the error will be dealt with immediately to effectively eliminate it while in the testing environment. If a solution to the error cannot be found while in the testing environment, the LTU will be taken to the lab and an attempt to resolve the software error will ensue. After the errors are resolved, testing will continue in order to expose additional errors in the system.

8.1.5.3 Adequate for Delivery

Once the identified errors have been resolved, the system will undergo the same testing process once more to ensure no further errors exist. This verification testing process will expose the potential errors that may arise while attempting to resolve other errors. Throughout the verification testing, detected errors will be resolved in the same way as mentioned in the previous error handling section. However, unlike the other testing methods, once an error is resolved in the verification test, the entire testing procedure will restart. This process will ensure that while eliminating detected errors, other errors will not arise, and if they do, they will be detected and eliminated immediately.

Once the LTU has been through the verification testing completely with no detected errors, the system will undergo one last test. One final, overarching test will be conducted to ensure the full functionality of the basic features as well as the additional, nonessential features. The system will be analyzed once more, ensuring no errors have occurred that will jeopardize the functionality of the system under any conditions. It should be noted that cosmetic errors associated with the Graphical User Interface (GUI) and the format of the software tools will, by definition, not cause the system to fail any tests. Once the system has met all of these expectations, it can be said to be adequate for delivery.

8.2 Hardware Test Plan

8.2.1 Overall Objective of Hardware Testing

In order to alleviate troubleshooting and check functionality of the LTU, several simpler test and verifications should be performed for each of the individual hardware components. These tests and verifications should be done following procedures designed to ensure that all components are isolated from each other and that potential external interference is minimized.

8.2.2 Testing Individual Hardware Components

Each component shall be tested and verified its proper functionality. In order to accomplish proper isolation of key components, evaluation and development kits will be utilized. The biggest challenge will be found when testing those components that use serial communication like the barometric pressure sensor that utilizes I^2C serial communication protocol and the flash memory that utilizes SPI serial communications protocol. The following sections define the approach for functional testing and verification.

8.2.2.1 GM862-GPS Functional Test Plan

Verification of correct functionality of the GPS will be done by powering the GM862 while mounted to the evaluation kit. With the use of the Python script and the AT commands, the values of the coordinates in NMEA format can be retrieved from the GPS module. The unit will be powered on through the evaluation kit, which will be connected to either a wall outlet via power adapter or laptop via USB. The coordinates and timestamp will be recorded and compared to the unit's actual location to check for accuracy and functionality.

Since the GM862 is in charge of all RF communications via GSM and GPS, it is the largest consumer of power. A test will be developed in order to find the best power saving mode. A fully charged battery will be used and the microcontroller will be programmed to indicate the GM862 to go into trickle power mode. The laps of time until the battery discharges will be recorded. The same procedure will be done but in this case the microprocessor will be set for push-to-fix mode. The better of the last two will be used for the final LTU product.

8.2.2.2 MSP430 Functional Test Plan

The microcontroller will require extensive testing. All of these tests will be performed on the MSP430 microcontroller. Some of the tests will involve the use of an oscilloscope to check for correct frequency operation of the master clock, which should be set between 50MHz and 60MHz. Also, the correct voltages levels for outputs will be verified to meet the 2.1 V minimum for high signal level and the 0.5 V maximum for low signal level.

Inputs will be simulated using the onboard power provided by the power supply (charger) and the battery. Verification of each digital input will be done by checking values in the microcontroller registers while placing direct jumpers into each pin. The usage of internal power will guarantee that voltage and current levels will be met when the LTU is in operation.

8.2.2.3 Flash Memory Test Plan

Since the flash memory is connected to the microprocessor via serial communication SPI, a test will be developed in order to send and retrieve data utilizing the microcontroller and its developing software. A temporary setup will be constructed using a breadboard. A small program will be written in order to upload a series of data values structured in a similar fashion to the NMEA protocol to the flash memory. A different program will then be used to retrieve the data from the flash memory. Finally, these two series of data will be compared.

8.2.2.4 Battery Test Plan

The battery functionality and performance will be done in several areas. Voltage level, current level, charge performance, temperature performance shall be evaluated. The first verification will be to check for voltage output after it has been fully charged. A minimum of 3.4 V and a maximum of 4.2 V will guarantee correct functionality of all components.

A second test will be to power on the unit and observe how long it takes for the battery to discharge completely. This can be accomplished by connecting all of the components including LEDs, barometric sensor, light sensor, GM862, and MSP430, and observing how long the battery lasts.

A third test will be to write a short program that forces all systems functions to work simultaneously (GPS, GSM, memory data logging and retrieval, activation of LEDs and collection of data from barometric sensor and light sensor) and check if there are any failures in the functionality of each component while observing and measuring any temperature change on the battery. A handheld infrared temperature gun is recommended. This last test is done not only to verify functionality, moreover it is done for safety reasons in the event the demands from the LTU's microprocessor and the GM862 are so high that the battery overheats and the potential for a fire could result.

8.2.2.5 Push Buttons Test Plan

Each push button will be tested by using a multi-meter and checking for continuity when depressed. This test is done while the power is not present. A second test could be performed while the PCB has power. The push button activation can be verified by measuring the voltage at the respective input pin on the GSM/GPS module

8.2.2.6 Barometric Sensor Test Plan

Since the barometric pressure sensor provides an analog signal in reference to the altitude, the data collected by the microcontroller will be used to verify the functionality of the sensor. A lower threshold will be set on the microcontroller, and a simulation will be performed by elevating the LTU. A possible test would be to attach the LTU to a weather balloon or sounding balloon that will, in turn, be attached to a string. The test will be performed controlling the ascent by timely releasing 50 meters of string at a time until an altitude that surpasses the preset threshold by at least 25%. There will be a time delay of 5 minutes between releases in order to clearly distinguish field measurements and data collected by the LTU. All field data will be recorded and then compared to the logged data retrieved from the flash memory.

8.2.2.7 Light Sensor Test Plan

In order to test the light sensor, a test will be implemented in which the light source will be removed from the LTU. Since this device communicates directly with the microcontroller via serial I^2C communication, access to the data collected by the microcontroller will be required. In order to test the sensor, while the MSP430 microcontroller is mounted on the development board, a direct reading of the registers can be made. Later, to verify correct functionality of the sensor when mounted to the PCB, data must be retrieved from the flash memory by implementing a small program on the microcontroller.

8.3 Hardware Test Procedure

The test procedures described below are based on the hardware test plans described in section 8.2. Each subsection focuses on the test procedure for a particular hardware module. The table within each subsection provides the step-by-step procedure that will be taken when testing the specified module. A detailed description, along with the conditions under which the test will take place, is provided for each step. The expected results are also listed to serve as stopping criteria for each test. That is, after each test is conducted, if the expected results are not met, modifications to the system will ensue and the testing for that step will continue until the expected results are obtained.

It should be noted that some test cases closely resemble others. Although some steps within the testing procedures may seem repetitive, the somewhat redundant steps are necessary to properly test specific variables. For example, after performing straightforward tests to ensure basic functionality, the same test is conducted with a variation in conditions. Again, this may seem redundant, but it is important that this factor be tested independently of other variable changes. This form of testing will make it easier to pinpoint the errors that arise throughout the testing of that specific module.

8.3.1 GM862-GPS Test Procedure

A description of the step-by-step procedure for testing the GM862-GPS is provided in the table below. The procedure is based on the GM862-GPS test plan described in section 8.2.2.1.

Table 8-1 GM862-GPS Test Procedure

Step #	Description	Conditions	Expected Result
1	Basic functionality of the GPS will be tested using sample inputs to achieve corresponding outputs. This will serve as a proof-of-concept test.	The test will be conducted with the GM862 mounted on the evaluation kit.	The GM862 should provide the correct output for each of corresponding inputs on the GPS.
2	The module will be powered on at different, random locations to ensure the location is properly identified.	The test will be conducted with the GM862 mounted on the evaluation kit.	The module should correctly identify its current location.
3	The module will be powered on at different, random locations and the current location and timestamp will be logged in the data log.	The test will be conducted with the GM862 mounted on the evaluation kit.	The module should correctly identify its current location and log the GPS coordinates and timestamp into the data log.
4	The module will be powered on at different, random locations to ensure the location is properly identified.	The test will be conducted with the GM862 mounted onto a Luggage Tracker Unit.	The module should correctly identify its current location.
5	The module will be powered on at different, random locations and the current location and timestamp will be logged in the data log.	The test will be conducted with the GM862 mounted onto a Luggage Tracker Unit.	The module should correctly identify its current location and log the GPS coordinates and timestamp into the data log.

8.3.2 MSP430 Test Procedure

A description of the step-by-step procedure for testing the MSP430 is provided in the table below. The procedure is based on the MSP430 test plan described in section 8.2.2.2.

Table 8-2 MSP430 Test Procedure

Step #	Description	Conditions	Expected Result
1	The frequency of the MSP430's master clock will be tested with an oscilloscope.	The test will use an oscilloscope connected to the MSP430 microcontroller.	The frequency of the MSP430's master clock should be set between 50MHz and 60MHz.
2	The correct output voltage of the MSP430 will be analyzed.	The test will use a voltmeter connected to the MSP430 microcontroller.	The 2.1 VDC minimum for high signal level should be met.
3	The correct output voltage of the MSP430 will be analyzed.	The test will use a voltmeter connected to the MSP430 microcontroller.	The 0.5 VDC maximum for low signal level should be met.
4	The flash memory will be analyzed to ensure proper functionality. Inputs will be simulated using the onboard power provided by the power supply and the battery.	The test will be conducted on the MSP430 microcontroller.	Direct jumpers will be placed into each pin of the microcontroller's registers. The inputted values will then be verified.
5	The serial communication SPI, which connects the microprocessor and the flash memory, will be tested by sending and retrieving data from the memory.	The test will be conducted on the MSP430 microcontroller.	The microcontroller should successfully transmit data to and from the flash memory.

8.3.3 Flash Memory Test Procedure

A description of the step-by-step procedure for testing the flash memory is provided in the table below. The procedure is based on the flash memory test plan described in section 8.2.2.3.

Table 8-3 Flash Memory Test Procedure

Step #	Description	Conditions	Expected Result
1	The basic functionality of the flash memory will be tested by sending and retrieving data via the serial communication ISP.	The microcontroller and its developing software will be used to write and read from the memory module. A breadboard will be initially used for testing.	The data should be successfully written to and read from the memory module.
2	The basic functionality of the flash memory will be tested by uploading a series of data values, structured similar to the NMEA protocol, to the flash memory.	The microcontroller and its developing software will be used to write and read from the memory module. Two short programs will be used: one to write the data to the module and one to read the data from the module.	The data written to the memory module should identically match the data read from it.
3	The basic functionality of the flash memory will be tested by uploading a series of data resembling the log data format used in this system.	The memory module will be integrated and mounted to the LTU. The log data will be written to and read from the module resembling typical usage.	The data written to the memory module should identically match the data read from it.

8.3.4 Battery Test Procedure

A description of the step-by-step procedure for testing the battery used in each LTU is provided in the table below. The procedure is based on the battery test plan described in section 8.2.2.4.

Table 8-4 Battery Test Procedure

Step #	Description	Conditions	Expected Result
1	After the battery has been fully charged, the output voltage will be checked.	The test will be conducted under basic operating conditions of the battery.	The voltage output should meet the requirements of the LTU.
2	After the battery has been fully charged, the LTU will be left powered on to determine the battery life of the system.	The test will be conducted with the battery fully mounted to the LTU and a program will be written to continually flash the LTU's LEDs.	The battery should power the system for a minimum of five hours. This will ensure that without a power-saver mode, the LTU can remain on through the duration of an average flight.
3	All of the system's individual components will remain at full load to determine the maximum temperature of the battery. This will serve as a safety test.	The test will be conducted with the battery fully mounted to the LTU and in a safe environment in case of extreme temperatures.	The battery should not exceed a safe operating temperature.

8.3.5 Push Buttons Test Procedure

A description of the step-by-step procedure for testing the push buttons is provided in the table below. The procedure is based on the push buttons test plan described in section 8.2.2.5.

Table 8-5 Push Buttons Test Procedure

Step #	Description	Conditions	Expected Result
1	The push buttons will be first pressed in a procedural manner to ensure they return to the original position. This test will ensure proper physical functionality of each push button.	The test will be conducted under basic operating conditions of the push buttons.	Each push button should return to its original position once it is released.
2	Each push button's continuity will be tested.	The test will be conducted under basic operating conditions of the push buttons.	Using a multi-meter, each push button's continuity should be verified.

8.3.6 Barometric Sensor Test Procedure

A description of the step-by-step procedure for testing the barometric sensor is provided in the table below. The procedure is based on the barometric sensor test plan described in section 8.2.2.6.

Table 8-6 Barometric Sensor Test Procedure

Step #	Description	Conditions	Expected Result
1	The basic functionality of the barometric sensor will be tested by simulating an elevation in altitude.	The LTU will be elevated using a weather balloon. The balloon will be raised to a specified altitude every 5 minutes.	The incremental elevations will be tracked and compared to the data retrieved from the flash memory via the barometric sensor.
2	The basic functionality of the barometric sensor will be tested by simulating an elevation in altitude.	The LTU will be elevated using a weather balloon. The balloon will be raised to a height of 50 meters and then retrieved at a steady rate. This will simulate a small-scale flight.	The elevations will be tracked and compared to the data retrieved from the flash memory via the barometric sensor.

8.3.7 Light Sensor Test Procedure

A description of the step-by-step procedure for testing the light sensor is provided in the table below. The procedure is based on the light sensor test plan described in section 8.2.2.7.

Table 8-7 Light Sensor Test Procedure

Step #	Description	Conditions	Expected Result
1	The light sensor's basic functionality will be tested by simply exposing it to a light source and then removing the light.	The test will be conducted under controlled conditions to ensure the light sensor is either directly exposed to light or there is no light exposure to the sensor.	The light sensor should successfully detect whether it is being exposed to light or not.
2	The light sensor's basic functionality will be tested by simply exposing it to a light source and then removing the light.	The light sensor will be integrated and mounted on the LTU. The test will be conducted under controlled conditions to ensure the light sensor is either directly exposed to light or there is no light exposure to the sensor.	The light sensor should successfully detect whether it is being exposed to a light source or not.
3	The serial I^2C communication line between the microcontroller and the light sensor will be tested.	The light sensor will be integrated and mounted on the LTU. The test will be conducted under controlled conditions to ensure the light sensor is either directly exposed to light or there is no light exposure to the sensor.	The data collected by the microcontroller will be accessed to verify the light sensor correctly identified if it was exposed to a light source or not.

8.4 Software Test Plans

8.4.1 Overall Objective of Software Testing

The overall software test effort shall ensure that all software components within the system work properly and efficiently. The software testing should effectively expose all errors within the system. This includes any previously identified errors and potential errors that can stem from those previously identified errors. Along with exposing high-level problems, the software testing shall also expose low-level problems within specific modules.

For this system, high-level problems are defined as problems that cause the overall process of the system to fail. This includes problems with the integration of the modules and communication issues between individual modules. Since a high level of integration is required to develop the LTU, all interfacing shall be rigorously tested to ensure proper communication is established between each, individual module.

Low-level problems are defined as the errors that occur within the module itself. Both high-level and low-level problems will be approached and resolved in a similar manner. However, in order to identify the source of the problem, different tests will be run on both the high-level and low-level functionality of the system to pinpoint the specific error. Overall, rigorous testing should eliminate all errors within the system, allowing the system to run flawlessly and as it was initially designed.

8.4.2 Testing Individual Software Components

In order to pinpoint errors within this complex system, individual test cases will be setup to test specific modules independently. From a software perspective, the on-chip software will be tested first in a simulated environment to ensure there is no potential damage done to the hardware due to defective software. Next the GPS server software will be tested to ensure its proper communication and functionality with the LTU. Finally, the iOS application will be tested to make sure there are no miscommunications between the GPS server and the iOS application.

8.4.2.1 On-Chip Software Test Plan

In order to avoid dependence upon the availability of hardware, the on-chip software will be simulated entirely in a software test environment before being implemented on hardware. This approach will also serve to protect against potential damage to the hardware due to faulty software. The simulation testing will continue until it has successfully passed all tests with no discovered errors.

The on-chip software will then be tested in collaboration with the hardware components. It should be noted that the testing procedures used when interfacing the on-chip software and hardware will be vigilant as to not damage any of the hardware components. The tests will incrementally analyze the functionality of individual modules, accumulating additional modules as the test successfully progresses. Throughout testing, if communication issues between modules, specifically the GPS and GSM modules, are discovered, the components will be individually analyzed to reveal low-lying errors.

8.4.2.2 GPS Server Software Test Plan

Because the GPS server is not implemented on actual hardware but rather resides, for all intents and purposes, on the internet, there is no concern for interfacing the GPS server itself with any hardware components. Thus, the GPS server will only require testing from a purely software perspective. Using this form of simulation testing will also ensure none of the hardware components experience significant damage through the duration of testing.

Furthermore, because an already existing GPS server software package is being implemented for this project, only the interfaces between this server and the other software components will be tested. The level of knowledge required to test any internal components of the server exceeds the knowledge base of our group, and would require an amount of time that is outside the scope of this course. Therefore, communication between the already existing GPS server and the LTU will be the focus of the testing, and the intrinsic functionality of the GPS server will be assumed to function properly.

The two main GPS server interfaces that must be tested are the interface with the LTU's on-chip software and the interface with the iOS application. As mentioned in section 5.6.3, the web portal is integrated as part of the GPS server software package and is already configured to interface with the server and database. Thus, the on-chip software must properly communicate with the iOS application to ensure proper functionality of the system.

8.4.2.2.1 GPS Server and On-Chip Software Interface

The interface between the GPS server and on-chip software will be tested using a simulated environment to mimic typical usage. The communication between the modules, specifically the GSM and GPS server, will be carefully analyzed to ensure the modules are correctly sending and receiving data. Specific test cases shall be designed to test the basic functionality of the system that utilizes the GPS server to process location information. Designing individual test cases in this manner will allow the group to pinpoint any encountered errors. As the testing advances, the interface between the GPS server and on-chip software will be holistically tested to ensure the modules can support full functionality of the LTU.

8.4.2.2.2 GPS Server and iOS Application Interface

The interface between the GPS server and iOS application will be tested in much the same way the GPS server and on-chip software will be tested. However, there is one key difference that designates the two testing methods. When testing the interface between the GPS server and the iOS application, the communication between the on-chip software and the iOS application will be analyzed first. The iOS application must be able to successfully communicate with the on-chip software to allow the on-chip software to properly communicate with the GPS server. Thus, the on-chip software acts as an intermediary to relay the information from the iOS application to the GPS server. Once a proper communication line is established between the iOS application and the on-chip software, the communication between the on-chip software and the GPS server will be analyzed to ensure the data is not altered throughout the transition from one module to the next. Individual test cases will be established to test these communication lines individually to correctly identify any errors that are discovered throughout the testing process.

8.4.2.3 iOS Application Software Test Plan

The iOS application software test plan will be considered in two phases. In the first phase, the basic operation of the software, as it relates to the actual operating system, will be considered. That is, the procurement of the application through Apple’s App Store, and the installment of the application on a device running iOS will be tested. In the second phase of testing, the integration and communication with the LTU will be examined. This will ensure the iOS application is correctly interfaced with the system.

8.4.2.3.1 iOS Application Test Plan: Part 1

In Part 1 of the iOS application testing, the application will be tested for errors contained exclusively within the application and its interface with the iOS operating system. The specific test cases will not address the interfacing of the iOS application with the LTU. Instead, the iOS application will be tested to ensure proper procurement and installment of the application on devices running iOS. At the conclusion of Part 1, the iOS application shall be readily available through Apple’s App Store, and any device capable of running iOS shall be able to successfully install and run the application.

8.4.2.3.2 iOS Application Test Plan: Part 2

Part 2 of the iOS application testing shall test the integration and communication with the LTU. Unlike Part 1, the basic functionality of the application being able to successfully run on an iOS device will not be considered. Instead, the communication lines that allow the iOS application to correctly transmit data to the LTU via the on-chip software will be analyzed. Specific test cases will provide sample inputs through the iOS application and those inputs will be analyzed as outputs once they are sent to the LTU and processed by the on-chip software. Once the communication lines are approved and data can be successfully transferred and processed, more complex test cases will assess the overall functionality of the iOS application as it communicates with the LTU.

8.5 Software Test Procedure

The test procedures described below are based on the software test plans described in section 8.4. Each subsection focuses on the test procedure for a particular software module. The table within each subsection provides the step-by-step procedure that will be taken when testing the specified module. A detailed description, along with the conditions under which the test will take place, is provided for each step. The expected results are also listed to serve as stopping criteria for each test. That is, after each test is conducted, if the expected results are not met, modifications to the system will ensue and the testing for that step will continue until the expected results are obtained.

It should be noted that some test cases closely resemble others. Although some steps within the testing procedures may seem repetitive, the somewhat redundant steps are necessary to properly test specific variables. For example, after performing straightforward tests to ensure basic functionality, the same test is conducted with a variation in altitude. Again, this may seem redundant, but it is important that this factor be tested independently of other variable changes. This form of testing will make it easier to pinpoint the errors that arise throughout testing that specific module.

8.5.1 On-Chip Software Test Procedure

A description of the step-by-step procedure for testing the on-chip software is provided in the table below. The procedure is based on the on-chip software test plan described in section 8.4.2.1. It should be noted that in order to avoid dependence upon availability of hardware, the on-chip software will be simulated entirely in a software test environment before being implemented on hardware. This testing method will also help protect against potential damage to the hardware due to faulty software.

Table 8-8 On-Chip Software Test Procedure

Step #	Description	Conditions	Expected Result
1	Basic functionality of the on-chip software will be tested using sample inputs to achieve corresponding outputs. This will serve as a proof-of-concept test.	The test will be conducted under basic simulated conditions.	The on-chip software should provide the correct output for each of its corresponding inputs.
2	The communication between the GPS and GSM modules will be tested by processing a sample text message containing coordinate information.	The test will be conducted under basic simulated conditions.	The on-chip software should properly process the GPS coordinates into a text message and send that information to the GSM module for transmission.
3	Basic functionality of the on-chip software will be tested using inputs resembling real-world scenarios (i.e. GPS coordinates, user settings, etc.) to achieve corresponding outputs.	The test will be conducted under basic simulated conditions.	The on-chip software should correctly receive and process data from the GPS module. The GPS coordinates should be processed into a text message and then transmitted via the GSM module.
4	Basic functionality of the on-chip software will be tested using inputs resembling real-world scenarios (i.e. GPS coordinates, user settings, etc.) to achieve corresponding outputs.	The test will be conducted under a simulated high altitude scenario, with other varying environmental factors.	The on-chip software should correctly receive and process data from the GPS module. The GPS coordinates should be processed into a text message and then transmitted via the GSM module.
5	Basic functionality of the on-chip software will be tested using inputs resembling real-world scenarios (i.e. GPS coordinates, user settings, etc.) to achieve corresponding outputs.	The test will be conducted under a simulated environment with varying altitude to simulate a typical flight path.	The on-chip software should correctly receive and process data from the GPS module. The GPS coordinates should be processed into a text message and then transmitted via the GSM module.

6	Full functionality of the on-chip software will be tested using extensive, real-world cases, which utilize all its features.	The test will be conducted under a simulation environment with varying altitude to simulate a typical flight path.	The on-chip software should correctly receive and process data from the GPS module. The GPS coordinates should be processed into a text message and then transmitted via the GSM module.
7	The interface between the on-chip software and the iOS application will be tested to ensure proper communication from the on-chip software to the iOS application.	The test will be conducted under basic simulated conditions.	The on-chip software should correctly receive and process data from the iOS application.
8	The interface between the on-chip software and the iOS application will be tested to ensure proper communication from the iOS application to the on-chip software.	The test will be conducted under basic simulated conditions.	The iOS application should correctly receive and process data from the on-chip software.
9	The interface between the on-chip software and the iOS application will be tested to ensure proper bidirectional communication between the two software modules.	The test will be conducted under basic simulated conditions.	The bidirectional communication between the on-chip software and the iOS application should successfully transmit data.
10	Basic functionality of the on-chip software will be tested using sample inputs to achieve corresponding outputs while interfacing with the hardware modules.	The on-chip software will be integrated and interfaced with the hardware components and no simulation will take place.	The on-chip software should correctly receive and process data from the GPS module. The GPS coordinates should be processed into a text message and then transmitted via the GSM module.

11	Full functionality of the on-chip software will be tested using extensive, real-world cases, which utilize all its features.	The on-chip software will be integrated and interfaced with the hardware components and no simulation will take place.	The on-chip software should correctly receive and process data from the GPS module. The GPS coordinates should be processed into a text message and then transmitted via the GSM module.
12	Full functionality of the on-chip software will be tested using extensive, real-world cases, which utilize all its features.	The on-chip software will be integrated and interfaced with the hardware components and a flight path with varying environmental factors with be simulated.	The on-chip software should correctly receive and process data from the GPS module. The GPS coordinates should be processed into a text message and then transmitted via the GSM module.

8.5.2 GPS Server Software Test Procedure

A description of the step-by-step procedure for testing the GPS server software is provided in the table below. The procedure is based on the GPS server software test plan described in section 8.4.2.2. It should be noted that because the GPS server is not implemented on the LTU's hardware and resides solely on the internet, there is no concern with interfacing the GPS server with any of the hardware components. Therefore, the GPS server will only be tested from a software perspective.

Table 8-9 GPS Server Software Test Procedure

Step #	Description	Conditions	Expected Result
1	The interface between the on-chip software and the GPS server will be tested to ensure proper communication from the on-chip software to the GPS server.	The test will be conducted under normal operating conditions.	The on-chip software should successfully send data regarding the LTU's current location to the GPS server.
2	The interface between the on-chip software and the GPS server will be tested to ensure proper communication from the GPS server to the on-chip software.	The test will be conducted under normal operating conditions.	The on-chip software should successfully receive and process data regarding the LTU's current location from the GPS server.
3	The interface between the on-chip software and the GPS server will be tested to ensure proper bidirectional communication between the two software modules.	The test will be conducted under normal operating conditions.	The bidirectional communication between the on-chip software and the GPS server should successfully transmit data.
4	The interface between the iOS application and the GPS server will be tested to ensure proper communication from the iOS application to the GPS server.	The test will be conducted under normal operating conditions.	The iOS application should successfully send data regarding the LTU's current location to the GPS server via the on-chip software.

5	The interface between the iOS application and the GPS server will be tested to ensure proper communication from the GPS server to the iOS application.	The test will be conducted under normal operating conditions.	The iOS application should successfully receive and process data regarding the LTU's current location from the GPS server via the on-chip software.
6	The interface between the iOS application and the GPS server will be tested to ensure proper bidirectional communication between the two software modules.	The test will be conducted under normal operating conditions.	The bidirectional communication between the iOS application and the GPS server should successfully transmit data via the on-chip software.
7	The interface between the on-chip software and the GPS server will be tested to ensure proper bidirectional communication between the two software modules.	The test will be conducted under a simulation environment with varying altitude to simulate a typical flight path.	The bidirectional communication between the on-chip software and the GPS server should successfully transmit data.
8	The interface between the iOS application and the GPS server will be tested to ensure proper bidirectional communication between the two software modules.	The test will be conducted under a simulation environment with varying altitude to simulate a typical flight path.	The bidirectional communication between the iOS application and the GPS server should successfully transmit data via the on-chip software.

8.5.3 iOS Application Software Test Procedure

A description of the step-by-step procedure for testing the iOS application software is provided in the tables below. The procedure is based on the iOS application software test plan described in section 8.4.2.3.

Table 8-10 iOS Application Software Test Procedure: Part 1

Step #	Description	Conditions	Expected Result
1	The iOS application will be procured through Apple's App Store.	The test will be conducted under basic operating conditions on an iOS supporting device.	The iOS application should be readily available through Apple's App Store.
2	After obtaining the necessary files from Apple's App Store, the installation of the application will be tested.	The test will be conducted under basic operating conditions on an iOS supporting device.	The iOS application should successfully install the application and all of its necessary features.
3	After successfully installing the iOS application, an attempt to run the application, without testing its intrinsic functionality, will ensue.	The test will be conducted under basic operating conditions on an iOS supporting device.	The iOS application should successfully open and remain idle at the main menu, awaiting input from the user.
4	The iOS application will be tested to ensure general functionality of the menus.	The test will be conducted under basic operating conditions on an iOS supporting device.	The iOS application should successfully navigate its menus without any menu navigational issues.
5	The iOS application will be tested for errors contained exclusively within the application and its interface with iOS operating system.	The test will be conducted under basic operating conditions on an iOS supporting device.	The iOS application should successfully communicate with the iOS operating system.
6	All aspects and functionality of the iOS application will be tested to ensure no errors occur while navigated the menus or communicating with the iOS operating system.	The test will be conducted under basic operating conditions on an iOS supporting device.	The iOS application should successfully operate as if it were under typical use. It should not encounter any menu navigational errors or communication errors with the iOS operating system.

Table 8-11 iOS Application Software Test Procedure: Part 2

Step #	Description	Conditions	Expected Result
1	The interface between the iOS application and the LTU will be tested to ensure proper communication from the iOS application to the LTU.	The test will be conducted under basic operating conditions on an iOS supporting device that communicates with a designated LTU.	The iOS application should successfully send data containing the user's settings and preferences to the LTU.
2	The interface between the iOS application and the LTU will be tested to ensure proper communication from the LTU to the iOS application.	The test will be conducted under basic operating conditions on an iOS supporting device that communicates with a designated LTU.	The LTU should successfully send data containing the LTU's current GPS coordinates, usage data, and alerts to the iOS application.
3	The interface between the iOS application and the LTU will be tested to ensure proper bidirectional communication.	The test will be conducted under basic operating conditions on an iOS supporting device that communicates with a designated LTU.	The bidirectional communication between the iOS application and the LTU should successfully transmit data.
4	The translation of the data from the iOS application to the on-chip software will be tested.	The test will be conducted under basic operating conditions on an iOS supporting device that communicates with a designated LTU.	The data should be successfully translated to ensure the data is properly interpreted by the on-chip software.
5	The translation of the data from the on-chip to the iOS application will be tested.	The test will be conducted under basic operating conditions on an iOS supporting device that communicates with a designated LTU.	The data should be successfully translated to ensure the data is properly interpreted by the iOS application.

6	The interface between the iOS application and the on-chip will be tested to ensure proper bidirectional communication.	The test will be conducted under basic operating conditions on an iOS supporting device that communicates with a designated LTU.	The bidirectional translation of data between the iOS application and the on-chip software should be flawless to ensure proper communication.
7	All aspects and functionality of the iOS application will be tested to ensure no errors occur during the bidirectional transmission of data between the iOS application and the LTU.	The test will be conducted under basic operating conditions on an iOS supporting device that communicates with a designated LTU.	The iOS application should successfully operate as if it were under typical use. It should not encounter any communication errors between the iOS application and the LTU.

9 Administrative Content

9.1 Milestones

The following are key milestones that should be met in order to keep the project moving ahead at a desirable pace. Although the dates provided are not completely fixed and can fluctuate slightly, they serve as a general timeline to hold our group accountable for the work necessary to complete the project.

Table 9-1 Project Milestones

Date	Milestone
01/20/12	Team members selected
02/03/12	Project selected
03/24/12	Senior Design I documentation 50% complete
04/06/12	GSM/GPS module selected
04/23/12	Battery selected
04/25/12	Senior Design I documentation complete
04/27/12	GPS server integration begun
04/30/12	PCB design begun
05/04/12	Microprocessor selected
05/07/12	PCB design completed
05/14/12	All parts selected and ordered for Phase A prototype
05/28/12	Phase A prototype assembled and testing begun
06/08/12	GPS server integration complete
07/06/12	All parts received for Phase B prototype
07/20/12	Phase B prototype assembled and testing begun
08/03/12	iOS application development begun
08/17/12	Phase A prototype complete
09/24/12	iOS application development complete and integration begun
10/15/12	Phase B prototype complete
11/05/12	iOS integration complete
11/26/12	Phase C prototype complete

In order to ensure the project maintains a smooth and continuous process, group meetings will be continually held, even while class sessions are not. Continuous communication over the summer and into the fall will allow any potential problems to be resolved in a timely manner. This will help alleviate some of the potential last minute, unforeseen problems that will inevitably come about when developing the system.

As seen in the chart above, the assembly and testing of the prototypes will take place over the summer. This timing will ensure that at the commencement of Senior Design II, a working prototype will be completed. This will provide our group with ample time to continually expand the system already in place. If time permits, the scalability of the system will allow for the addition of other features to further enhance the system's capabilities.

Before adding additional features to the system, a group meeting will be held to discuss the monetary cost and amount of time necessary to incorporate these features. If a consensus is made to include these features, the timeline and milestones chart will be further expanded. As the final deadline approaches, the timeline and milestones will be continually assessed and reconfigured to include more specific, steadfast dates that must be met to ensure the completion of the system.

9.2 Budget and Financing

At the time of the writing of this document, no major corporations or significant sponsors have been identified to cover the full budget of the project. However, Ms. Sandy Cook, President of Decimal Point Accounting Services in Jacksonville, Florida, has agreed to contribute $500.00 to help with the funding of the project. This will cover a significant portion of the overall cost, which can be seen in the table below.

Although there is no direct relationship between Decimal Point Accounting Services and the need of an LTU, Mrs. Cook was still interested in the process used to develop such a device and the overall functionality of the device. Her interest was sparked by her own personal experiences of losing luggage while on travel for business. After the idea of using a small device to continually track the luggage throughout its journey was pitched, she agreed to make a donation.

When discussing the system with Ms. Cook, a general, overarching description was used. The discussion took place at the beginning of the project development. Therefore, no specific details or specifications were known. However, as a project sponsor, Ms. Cook is continually kept up-to-date with new information concerning the overall process and design of the system. Documents are regularly sent to her via email to provide her with updates on how her contribution is being distributed within the system.

The estimated budget for this project is shown in the table below:

Table 9-2 Project Budget

Part	Cost	Qty	Total Cost
GM862 Module	120.00	2	240.00
GM862 Evaluation Board	100.00	1	100.00
GSM Antenna	20.00	2	40.00
GPS Antenna	15.00	2	30.00
Mini USB cable & port	5.00	2	10.00
Battery, charger, & accessories	45.00	2	90.00
Microprocessor	10.00	2	20.00
Memory	10.00	2	20.00
Printed Circuit Board	80.00	2	160.00
50-pin connector	10.00	2	20.00
Enclosure	15.00	2	30.00
iOS Developer Membership	100.00	2	200.00
Miscellaneous	50.00	1	50.00
Total			**$ 1,010.00**

A project briefing is currently being developed to present to potential sponsors. Potential sponsors include those directly or indirectly associated with the buying, selling, or use of luggage. Specific information on how the contributions will be applied to the system will be explicitly defined to make sure each contributor is clear on how their donation will be used. A communication line will be established with all contributors to ensure they receive continuous updates on the development of the system. If no further sponsors or donors are found to cover the remaining budget of the system, the cost of the project will be divided evenly amongst the team members. Unanticipated costs will inevitably be encountered throughout the development of the system. Unexpected expenses may be encountered if additional features are added to the system after all key components have been successfully developed. They may also come about if certain modules are damaged in the process of building either of the prototypes. Nonetheless, each group member has agreed to evenly distribute any unexpected expenses to ensure a working system can be developed.

10 Personnel and Bibliography

10.1 David Farrell

David Farrell is a senior Computer Engineering student at the University of Central Florida. For the last 3 years, he has worked as a student intern at Lockheed Martin Missiles and Fire Control in Orlando. After graduating in December of 2012, he plans to pursue a career in systems engineering or a related field. He hopes to work in either the defense industry, the intelligence community, or in full time missions. Outside the classroom, David leads the college ministry at his church and enjoys participating in short term mission trips, playing sports, and spending time with family.

10.2 Evan Husk

Evan Husk is a senior Computer Engineering student at the University of Central Florida. He is currently conducting research and writing an undergraduate thesis under the supervision of Dr. Avelino Gonzalez. The research and thesis is on *Imitating Individualized Facial Expressions in a Human-Like Avatar through Machine Learning*. After completing his thesis and graduating in December of 2012, he plans on furthering his education by pursuing a master's degree in systems engineering at the University of Florida. He wishes to work as a systems engineer in the defense industry or in the field of communications. In his free time, Evan likes to play sports, fish, and spend time with his family.

10.3 Jose Mousadi

Jose Mousadi is a senior Electrical Engineering student at the University of Central Florida. For the last 5 years he has been working as a Project Engineer and Electrical Designer at Gas Turbine Efficiency, Inc. in Orlando, Fl. Previously, he worked as a Panel Builder at G&T Conveyor, Inc. in Tavares, Fl. This hand-on approach has aid him acquired extensive knowledge in industrial control systems and instrumentation geared towards the power generation and energy sector. Lately he has designed, developed and managed auxiliary systems used in Gas and Steam Turbines as well as other OEM products. In his free time, Jose likes to travel, spend time with family, and play basketball.

11 Facilities and Equipment

The primary facility for development and testing of our system will be the Senior Design Laboratory at the University of Central Florida (UCF). Because our team does not reside in the same general area, it was necessary to choose a designated location at which we may meet to work on the development of the project. The UCF campus was the logical choice for this need as it provides ready access to both engineering equipment in the senior design lab as well as engineering knowledge in the form of instructors and fellow classmates. The lab is equipped with lockers in which equipment and other supporting components and devices may be stored. Due to the small size of our project, we anticipate being able to store all of our hardware and other equipment in these lockers.

The lab is also equipped with an ample amount of engineering equipment such as computers, oscilloscopes, power supplies, function generators, and bread boards. If any testing needs to be performed that is not supported by our evaluation board or PCB, we intend to use the equipment in the lab to perform the desired test activities. In general, it is anticipated that the evaluation board purchased by our team will accommodate the majority of our testing needs. The board will be equipped with a serial communication interface, power port, and auxiliary I/O ports for interfacing with other devices and components. In other words, the evaluation board will be the hub around which all other supporting components will revolve and operate in the testing and development phases of this project.

12 Suppliers and Service Providers

Most of the suppliers will be contacted via email or through orders placed online through their websites. The following table lists the suppliers and their websites:

Table 12-1 Vendors, Suppliers and Service Providers

Supplier Name	Website
Mouser Electronics	www.Mouser.com
Digi-Key	www.digikey.com
Sparkfun Electronics	www.sparkfun.com
Semiconductor Store	www.semiconductorstore.com
Advance Circuits	www.4pcb.com
Motorola	www.motorola.com
Newark	www.newark.com
AT&T	www.att.com

The services of Advanced Circuits located in Aurora, CO and Tempe, AZ will be required in order to fabricate and populate the PCB. The services of AT&T Mobility will be required in order to provide wireless cellular network access. The SIM card will purchased along with a prepaid payment plan. The service plan that will fit best for the LTU is the $2 Daily Unlimited Plan.

13 Appendices

13.1 Acronyms

Abbreviation	Description
AT	Attention
COG	Course Over Ground
DFC	Design File Check
GPIO	General Purpose Input/Output
GPRS	General Packet Radio Service
GPS	Global Positioning System
GSM	Global System for Mobile Communications
HDOP	Horizontal Dilution Of Precision
I^2C	Inter-Integrated Circuit
iOS	(Apple) Operating System
IP	Internet Protocol
LED	Light Emitting Diode
LTU	Luggage Tracker Unit
NMEA	National Marine Electronics Association
NVM	Non Volatile Memory
PC	Personal Computer
PCB	Printed Circuit Board
PDU	Packet Data Unit
SIM	Subscriber Identity Module
SiRF	Silicon Radio Frequency
SMS	Short Message Service
SPI	Serial Peripheral Interface
UCF	University of Central Florida
USB	Universal Serial Bus
UTC	Coordinated Universal Time

13.2 Works Cited

"Air Travel Consumer Report." *Aviation Consumer Protection and Enforcement*. March 2012. U.S. Department of Transportation. 25 Mar. 2012 <http://airconsumer.dot.gov/reports/2012/March/2012MarchATCR.pdf>.

The OpenGTS Project. 2011. Geo Telematic Solutions Incorporated. 25 Mar. 2012 <http://opengts.sourceforge.net/index.html>.

"GM, GE/GL Families GPS Solutions User Guide." 12 Dec. 2011. Telit® Wireless Solutions. 30 Mar. 2012 <http://www.telit.com/module/infopool/download.php?id=4099>.

"Telit Modules Software User Guide." 26 Mar. 2012. Telit® Wireless Solutions. 30 Mar. 2012 <http://www.telit.com/module/infopool/download.php?id=522>.

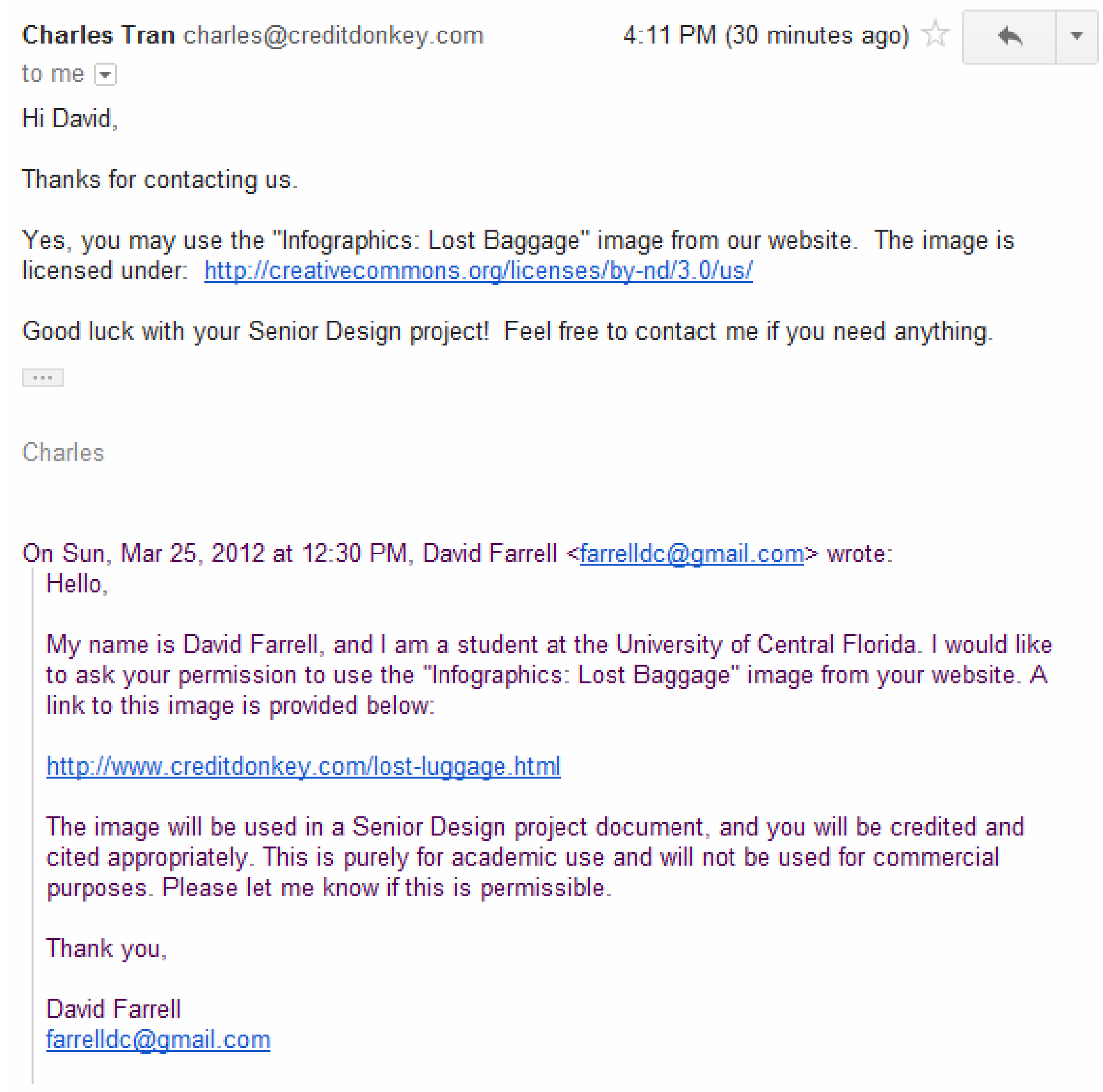
"AT Commands Reference Guide." 20 Mar. 2012. Telit® Wireless Solutions. 30 Mar. 2012 <http://www.telit.com/module/infopool/download.php?id=542>.

"GM862 Family Hardware User Guide." 15 Sept. 2011. Telit® Wireless Solutions. 30 Mar. 2012 <http://www.telit.com/module/infopool/download.php?id=537>.

"Easy Script in Python." 3 Dec. 2010. Telit® Wireless Solutions. 30 Mar. 2012 <http://www.telit.com/module/infopool/download.php?id=617>.

13.3 Copyright Permissions

13.3.1 Lost Luggage Graph (Figure 2-1)



Charles Tran charles@creditdonkey.com 4:11 PM (30 minutes ago)
to me

Hi David,

Thanks for contacting us.

Yes, you may use the "Infographics: Lost Baggage" image from our website. The image is licensed under: http://creativecommons.org/licenses/by-nd/3.0/us/

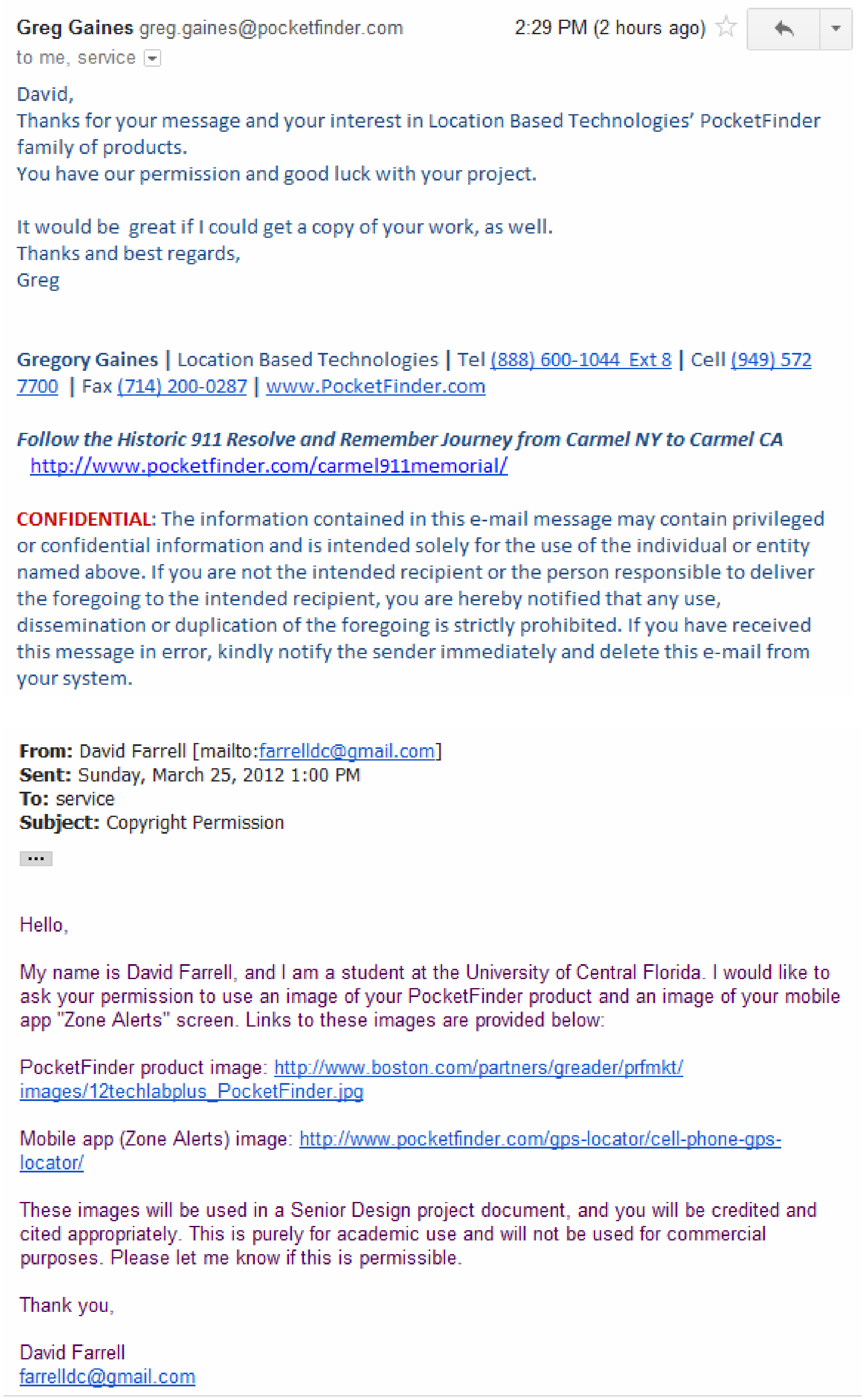
Good luck with your Senior Design project! Feel free to contact me if you need anything.

Charles

On Sun, Mar 25, 2012 at 12:30 PM, David Farrell <farrelldc@gmail.com> wrote:

> Hello,
>
> My name is David Farrell, and I am a student at the University of Central Florida. I would like to ask your permission to use the "Infographics: Lost Baggage" image from your website. A link to this image is provided below:
>
> http://www.creditdonkey.com/lost-luggage.html
>
> The image will be used in a Senior Design project document, and you will be credited and cited appropriately. This is purely for academic use and will not be used for commercial purposes. Please let me know if this is permissible.
>
> Thank you,
>
> David Farrell
> farrelldc@gmail.com

13.3.2 PocketFinder Image (Figure 4-1)



Greg Gaines greg.gaines@pocketfinder.com 2:29 PM (2 hours ago)
to me, service

David,
Thanks for your message and your interest in Location Based Technologies' PocketFinder family of products.
You have our permission and good luck with your project.

It would be great if I could get a copy of your work, as well.
Thanks and best regards,
Greg

Gregory Gaines | Location Based Technologies | Tel (888) 600-1044 Ext 8 | Cell (949) 572 7700 | Fax (714) 200-0287 | www.PocketFinder.com

Follow the Historic 911 Resolve and Remember Journey from Carmel NY to Carmel CA
http://www.pocketfinder.com/carmel911memorial/

CONFIDENTIAL: The information contained in this e-mail message may contain privileged or confidential information and is intended solely for the use of the individual or entity named above. If you are not the intended recipient or the person responsible to deliver the foregoing to the intended recipient, you are hereby notified that any use, dissemination or duplication of the foregoing is strictly prohibited. If you have received this message in error, kindly notify the sender immediately and delete this e-mail from your system.

From: David Farrell [mailto:farrelldc@gmail.com]
Sent: Sunday, March 25, 2012 1:00 PM
To: service
Subject: Copyright Permission

...

Hello,

My name is David Farrell, and I am a student at the University of Central Florida. I would like to ask your permission to use an image of your PocketFinder product and an image of your mobile app "Zone Alerts" screen. Links to these images are provided below:

PocketFinder product image: http://www.boston.com/partners/greader/prfmkt/images/12techlabplus_PocketFinder.jpg

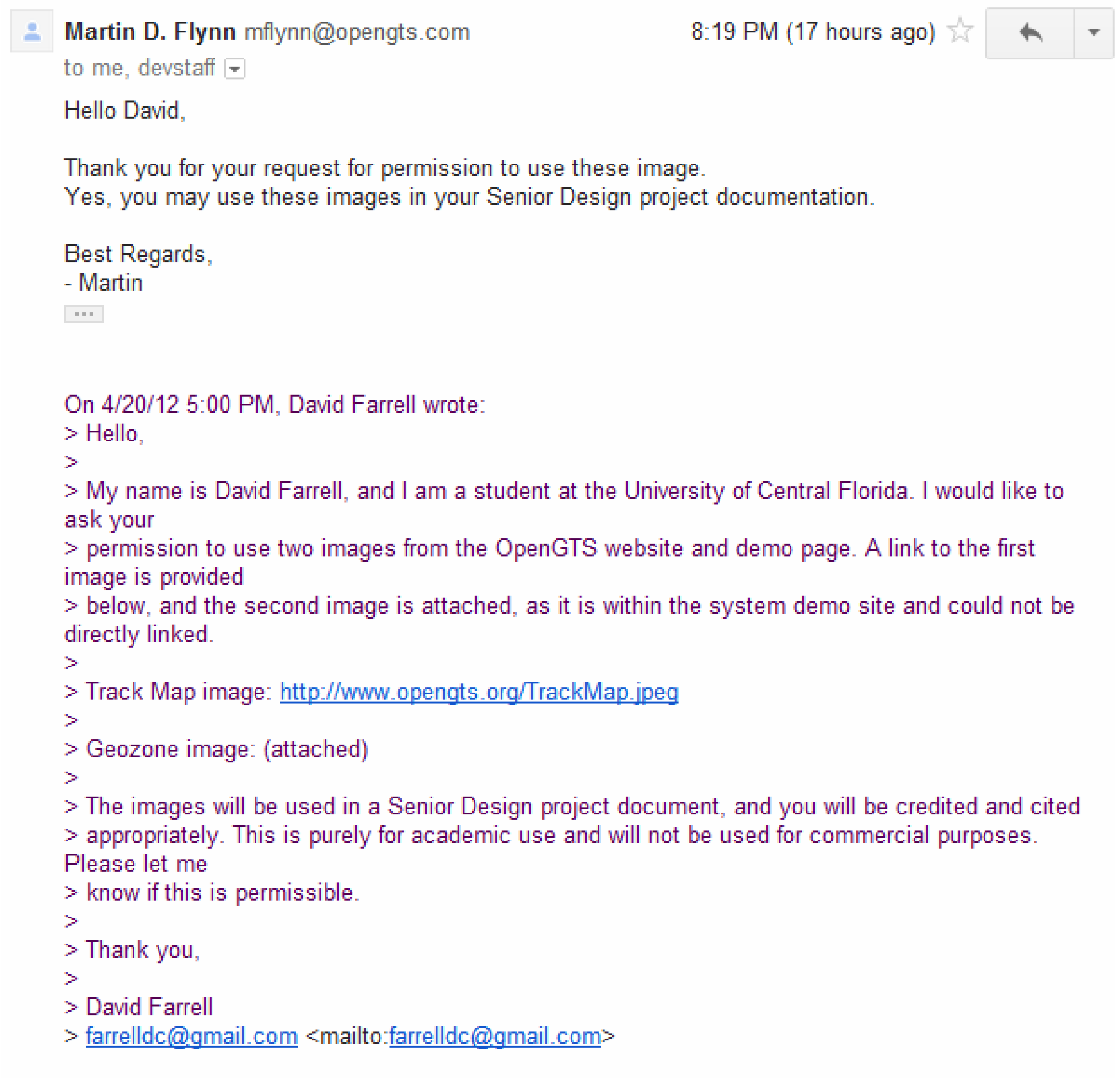
Mobile app (Zone Alerts) image: http://www.pocketfinder.com/gps-locator/cell-phone-gps-locator/

These images will be used in a Senior Design project document, and you will be credited and cited appropriately. This is purely for academic use and will not be used for commercial purposes. Please let me know if this is permissible.

Thank you,

David Farrell
farrelldc@gmail.com

13.3.3 OpenGTS Figures (Figure 5-12 and Figure 5-13)



Martin D. Flynn mflynn@opengts.com 8:19 PM (17 hours ago)
to me, devstaff

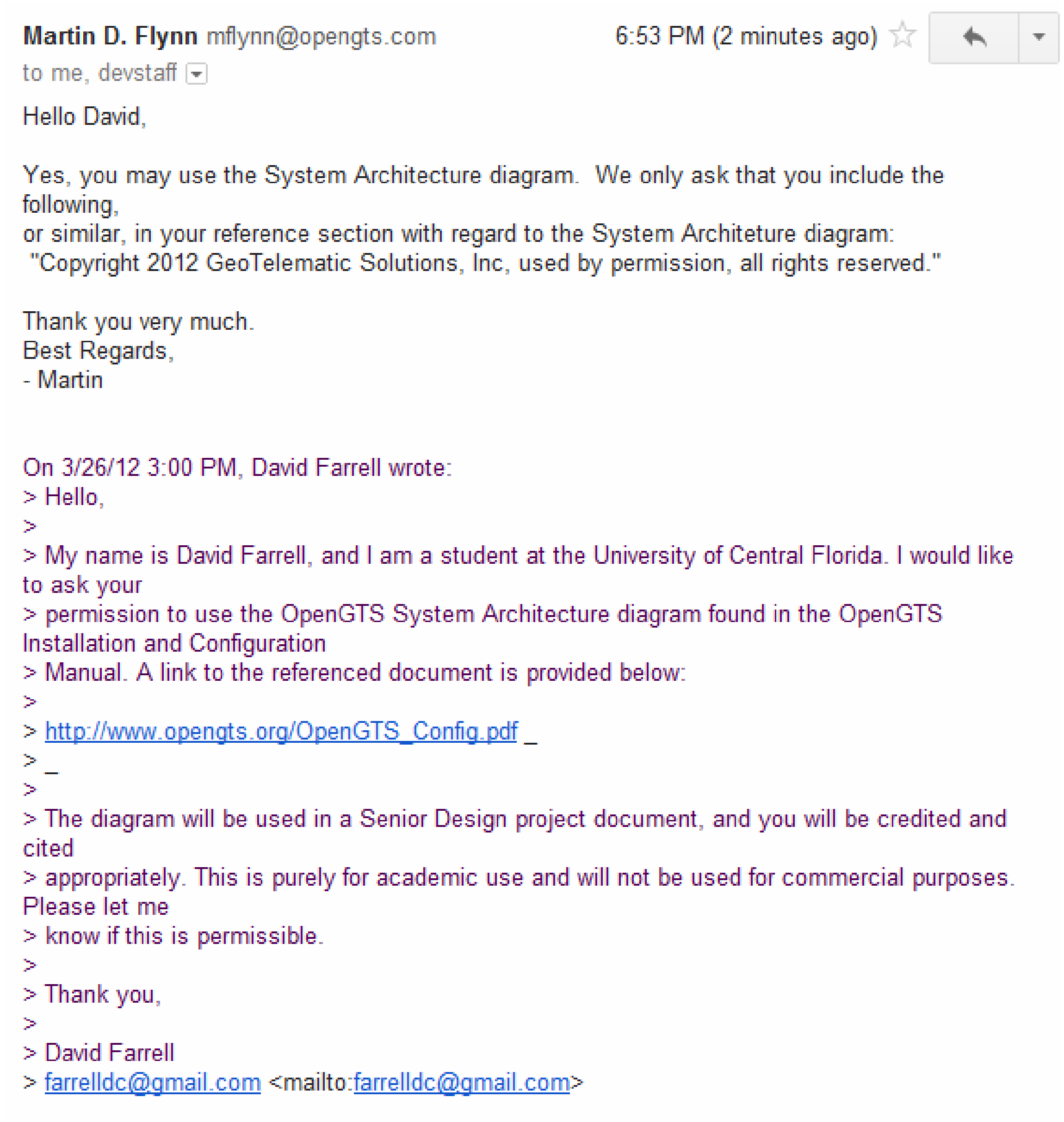
Hello David,

Thank you for your request for permission to use these image.
Yes, you may use these images in your Senior Design project documentation.

Best Regards,
- Martin

On 4/20/12 5:00 PM, David Farrell wrote:
> Hello,
>
> My name is David Farrell, and I am a student at the University of Central Florida. I would like to ask your
> permission to use two images from the OpenGTS website and demo page. A link to the first image is provided
> below, and the second image is attached, as it is within the system demo site and could not be directly linked.
>
> Track Map image: http://www.opengts.org/TrackMap.jpeg
>
> Geozone image: (attached)
>
> The images will be used in a Senior Design project document, and you will be credited and cited
> appropriately. This is purely for academic use and will not be used for commercial purposes. Please let me
> know if this is permissible.
>
> Thank you,
>
> David Farrell
> farrelldc@gmail.com <mailto:farrelldc@gmail.com>

13.3.4 OpenGTS System Architecture Diagram (Figure 5-11)



Martin D. Flynn mflynn@opengts.com 6:53 PM (2 minutes ago)
to me, devstaff

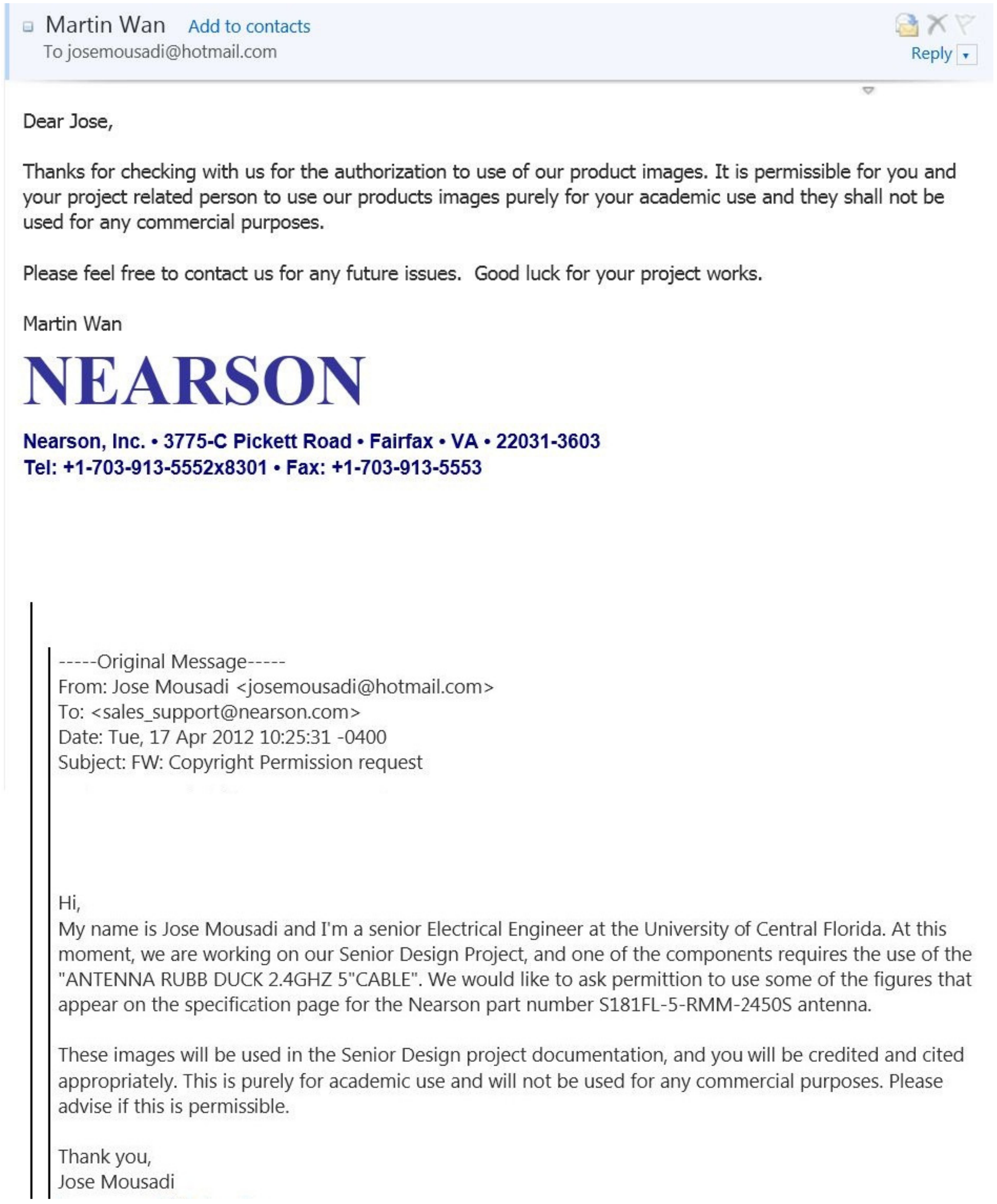
Hello David,

Yes, you may use the System Architecture diagram. We only ask that you include the following,
or similar, in your reference section with regard to the System Architeture diagram:
"Copyright 2012 GeoTelematic Solutions, Inc, used by permission, all rights reserved."

Thank you very much.
Best Regards,
- Martin

On 3/26/12 3:00 PM, David Farrell wrote:
> Hello,
>
> My name is David Farrell, and I am a student at the University of Central Florida. I would like to ask your
> permission to use the OpenGTS System Architecture diagram found in the OpenGTS Installation and Configuration
> Manual. A link to the referenced document is provided below:
>
> http://www.opengts.org/OpenGTS_Config.pdf _
> _
>
> The diagram will be used in a Senior Design project document, and you will be credited and cited
> appropriately. This is purely for academic use and will not be used for commercial purposes. Please let me
> know if this is permissible.
>
> Thank you,
>
> David Farrell
> farrelldc@gmail.com <mailto:farrelldc@gmail.com>

13.3.5 Nearson 2.4GHz Fixed Mount Swivel Antenna (Figure 5-4)



Martin Wan Add to contacts
To josemousadi@hotmail.com

Reply

Dear Jose,

Thanks for checking with us for the authorization to use of our product images. It is permissible for you and your project related person to use our products images purely for your academic use and they shall not be used for any commercial purposes.

Please feel free to contact us for any future issues. Good luck for your project works.

Martin Wan

NEARSON

Nearson, Inc. • 3775-C Pickett Road • Fairfax • VA • 22031-3603
Tel: +1-703-913-5552x8301 • Fax: +1-703-913-5553

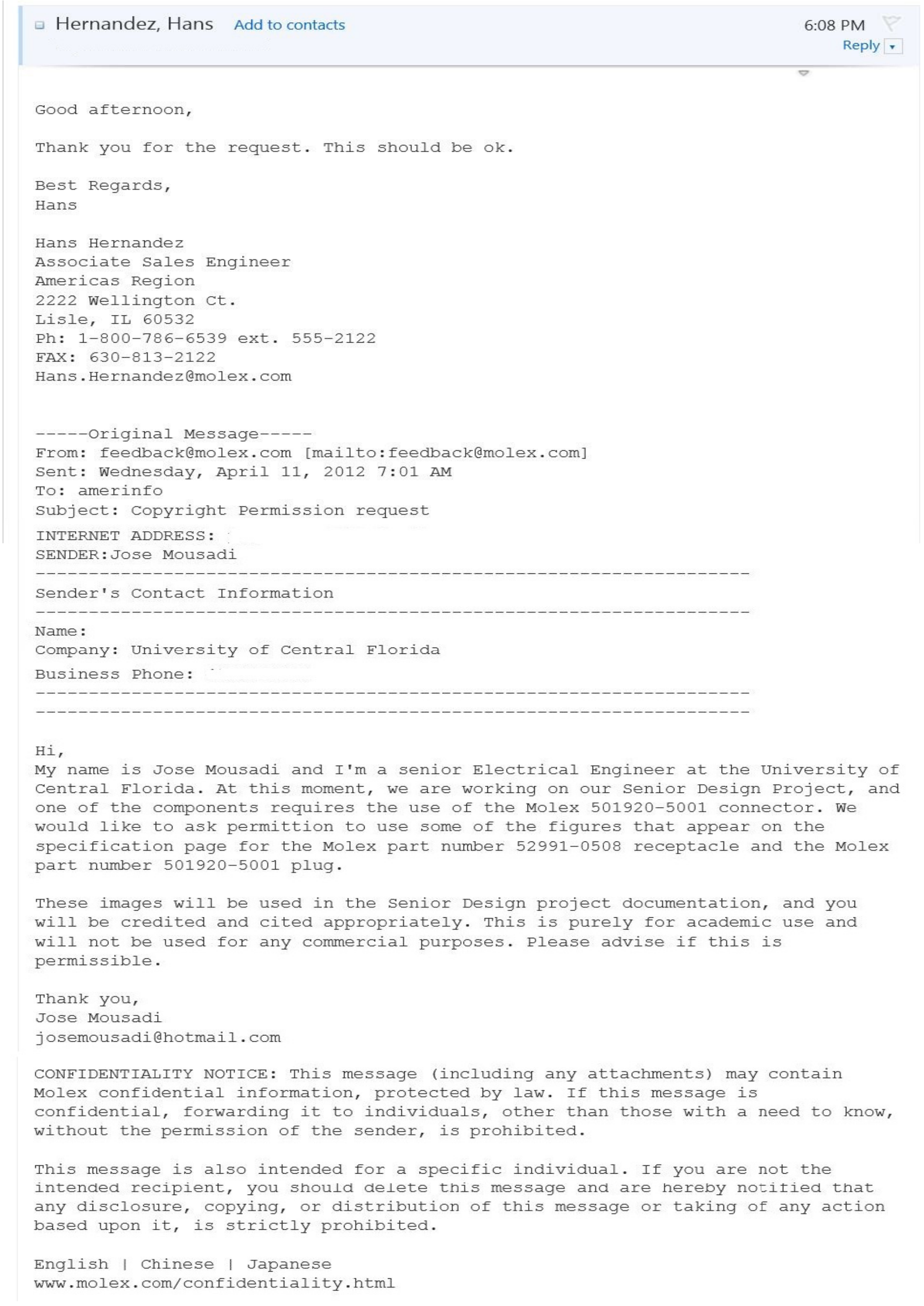
-----Original Message-----
From: Jose Mousadi <josemousadi@hotmail.com>
To: <sales_support@nearson.com>
Date: Tue, 17 Apr 2012 10:25:31 -0400
Subject: FW: Copyright Permission request

Hi,
My name is Jose Mousadi and I'm a senior Electrical Engineer at the University of Central Florida. At this moment, we are working on our Senior Design Project, and one of the components requires the use of the "ANTENNA RUBB DUCK 2.4GHZ 5"CABLE". We would like to ask permittion to use some of the figures that appear on the specification page for the Nearson part number S181FL-5-RMM-2450S antenna.

These images will be used in the Senior Design project documentation, and you will be credited and cited appropriately. This is purely for academic use and will not be used for any commercial purposes. Please advise if this is permissible.

Thank you,
Jose Mousadi

13.3.6 Molex Figures (Figure 5-6 and Figure 5-7)



Hernandez, Hans Add to contacts 6:08 PM
Reply

Good afternoon,

Thank you for the request. This should be ok.

Best Regards,
Hans

Hans Hernandez
Associate Sales Engineer
Americas Region
2222 Wellington Ct.
Lisle, IL 60532
Ph: 1-800-786-6539 ext. 555-2122
FAX: 630-813-2122
Hans.Hernandez@molex.com

-----Original Message-----
From: feedback@molex.com [mailto:feedback@molex.com]
Sent: Wednesday, April 11, 2012 7:01 AM
To: amerinfo
Subject: Copyright Permission request
INTERNET ADDRESS:
SENDER:Jose Mousadi

Sender's Contact Information

Name:
Company: University of Central Florida
Business Phone:

Hi,
My name is Jose Mousadi and I'm a senior Electrical Engineer at the University of Central Florida. At this moment, we are working on our Senior Design Project, and one of the components requires the use of the Molex 501920-5001 connector. We would like to ask permition to use some of the figures that appear on the specification page for the Molex part number 52991-0508 receptacle and the Molex part number 501920-5001 plug.

These images will be used in the Senior Design project documentation, and you will be credited and cited appropriately. This is purely for academic use and will not be used for any commercial purposes. Please advise if this is permissible.

Thank you,
Jose Mousadi
josemousadi@hotmail.com

CONFIDENTIALITY NOTICE: This message (including any attachments) may contain Molex confidential information, protected by law. If this message is confidential, forwarding it to individuals, other than those with a need to know, without the permission of the sender, is prohibited.

This message is also intended for a specific individual. If you are not the intended recipient, you should delete this message and are hereby notified that any disclosure, copying, or distribution of this message or taking of any action based upon it, is strictly prohibited.

English | Chinese | Japanese
www.molex.com/confidentiality.html