U-LARM Bike Lock Accessory

By

Craig Cary

Senior Project

ELECTRICAL ENGINEERING DEPARTMENT

California Polytechnic State University

San Luis Obispo

2015

Table of Contents

List of Tables and Figures

Tables *Page*

Figures *Page*

Acknowledgements

After the last few months spent researching and designing this bike lock, I would like to thank a few people for their help in making everything come together. First, I would like to thank Devin Kerns of Kernstech LLC for providing me with this U-LARM bike lock idea. Without his vision, I wouldn't have had the opportunity to work on this project. I also wanted to thank my advisor Dr. Bridget Benson for her willingness to work with me on this senior project and her flexibility in timing with my unusual academic situation. Finally, I wanted to thank my family and friends for their encouragement and support throughout my time here at Cal Poly.

Abstract

Over the past few years, anti-pollution organizations and activists have encouraged society to use eco-friendly transportation like ride-sharing and biking. According to the San Luis Obispo Police Department, however, bike owners report approximately 269 bike thefts per year. This statistic illustrates the need for a more effective way to protect bicycle property.

The development of an alarm system for the U-Lock bike lock encourages bike owners to ride their bicycles by providing them with additional measures to protect their bicycle property. This U-Lock accessory, when cut, sounds an alarm to alert nearby persons of the theft-in-progress. In order to turn off the alarm, the bike owner must disconnect the bike lock and disable the alarm. This U-Lock alarm promotes biking as an effective means of transportation by deterring bike thieves.

Chapter 1: Introduction

Introduction

Over the past few years, anti-pollution organizations and activists have encouraged society to use eco-friendly transportation like ride-sharing and biking. According to the San Luis Obispo Police Department, however, bike owners report approximately 269 bike thefts per year [8]. This statistic illustrates the need for a more effective way to protect bicycle property.

The development of an alarm system for the U-Lock bike lock encourages bike owners to ride their bicycles by providing them with additional measures to protect their bicycle property. This U-Lock accessory, when cut, sounds an alarm to alert nearby persons of the theft-in-progress. In order to turn off the alarm, the bike owner must disconnect the bike lock and disable the alarm. This U-Lock alarm promotes biking as an effective means of transportation by deterring bike thieves.

M. A. Acena and Y. Ghabbour have a US patent that addresses the idea of isolated current sensing in circuit design [9]. Using their research and the "Making Sense of Current" application note, this project focuses on the use of resistor voltage measurements to trigger an alarm upon a 15 mV change [15]. This alarm manifests itself in multiple ways. In the immediate context, this alarm produces a 100 dB audible sound which acts as a deterrent to scare away the thief. In addition, this accessory also communicates wirelessly with an app on the owner's smart phone. This app raises an alarm in the vicinity of the bike owner to properly alert him or her of the theft in progress. A technological paper produced by Texas Instruments named *A Primer to Wi-Fi Provisioning for IoT Applications* provides background information regarding wireless connectivity in the internet of things (IoT) movement currently active in American society [6]. By joining this device with the IoT movement, the user connects more readily with his or her bicycle and can rest assured of the security of his or her property.

Through active theft prevention techniques, this device accounts for environmental, physical, and electrical tampering. By utilizing a waterproof housing technique, this accessory resists water flow and provides electrical isolation for the internal circuitry. In addition, a ¼" steel housing encases this circuitry to prevent the sawing and breaking of this device. Similarly, this encasing separates the thief from direct contact with the electrical components. By designing the circuit with unique resistive characteristics, this accessory also accounts for short circuiting techniques meant to bypass the sensing circuitry. Each and every one of these techniques come together to create a tamper-proof, effective product which meets the desired needs of the customer.

Chapter 2: Customer Needs, Requirements, and Specifications

Customer Needs Assessment

To determine my customer's needs, I first looked at customers who might buy this product and the companies who might invest in this product. The consumers who might buy this product include bike owners and their family and friends. When making design choices for this product and identifying the customer needs, I need to keep these users in mind. This U-Lock accessory needs the ability to effectively and reliably alert a bike owner during a theft-in-progress. With a simple, low cost design, this accessory should deter thieves from attempting to steal bicycles on the street by interfacing the accessory with the U-Lock device. In order to provide an easy-to-use system, this accessory's design should minimize additional weight and area requirements. In addition, it should not sacrifice the structural reliability of the U-Lock and should operate properly in case of the introduction of water into the system.

For the development of this assessment, I hope to interact with the biking community to find out what they would like to see in this product. With their experience and familiarity with the current bike lock systems, they can provide useful thoughts and suggestions to make this product more effective and desirable.

Requirements and Specifications

Based on the customer needs assessment above, I created the list of marketing requirements shown in *Table 1*. These requirements attempt to cover a wide range of user-desired applicable areas. Using the marketing requirements, I developed the engineering specifications list below. These specifications address many of the more technical and numerical sides of the U-LARM accessory including cost, size, weight, durability, battery life, and noise requirements. This table provides a list of reasonable design criteria to keep in mind when creating this accessory.

TABLE I

U-LARM REQUIREMENTS AND SPECIFICATIONS

Spec. Number	Marketing Requirements	Engineering Specifications	Justification
1	1, 2, 5, 7	The accessory costs less than 20 additional dollars per unit.	Since the U-Lock itself costs approximately $35, the accessory price needs to cost less than the normal price.
2	3, 4, 7	The battery power for this accessory lasts 25+ hours of constant operation before requiring a new battery.	Since continual battery replacement annoys users, this device's battery life needs to last a long time.
3	5, 7	The accessory notifies the owner upon low battery supply conditions.	In order to create an effective alarm, the system needs power. As a result, low battery notifications notify the user to change the battery before the product loses power.
4	5, 7, 8	This device wirelessly communicates over a distance of 50 yards.	This distance specifies the proximity range between the U-LARM transmitter and the receiver.
5	1, 5, 7, 8	The U-LARM mobile application doesn't add any additional costs to the product.	This provides an easy to use, low cost method of connecting the user with their bicycle lock.

6	2, 4	Under daily use, this accessory has a lifespan of 5+ years.	Basic vibrations, dings, and usage do not cause malfunctions in the system. Since users only desire to purchase this accessory once, the accessory lasts 5 years before needing replacement.
7	1, 5	This accessory weighs less than 12 ounces.	While biking along roads and trails, bikers dislike additional weight. To minimize this, my accessory emphasizes low weight.
8	1, 4, 5	This accessory maintains the size of the U-Lock within 1 centimeter on each side of the device.	Most bikers do not enjoy carrying around large objects on their bicycles. As a result, we minimized the additional area of our system.
9	4, 5	With experience, a user installs this device in under 10 seconds.	After practicing a few times, knowledgeable bikers should install this device in a short amount of time. The user does not spend a long time connecting the device to the U-Lock.
10	7, 9	When triggered, this accessory produces a constant alarm that outputs an alert at approximately 100 dB.	This accessory requires 100 dB output sounds for effective notification. It requires sufficient noise to alert people nearby.
11	7, 9	The alarm for this device sounds for 5 minutes before turning itself off.	After 5 minutes, a thief has already finished stealing the bike or has run away due to the sound of the alarm. As a result, the alarm deactivates after 5 minutes.
12	2, 6, 7, 8, 9	Upon the introduction of water into this system, the accessory remains electrically functional.	Thieves may tamper with the accessory using water. In addition, users may store their bikes outside during a rainstorm. As a result, this device properly operates under these conditions.
13	2, 6, 7	This device avoids compromise of the mechanical structure of the original U-Lock device.	All joints or additions to this device independently attach to the U-Lock and doesn’t structurally compromise the U-Lock.
14	1, 2, 3, 6, 7, 8, 9	This device notifies the bike owner after detecting a 10 percent voltage change across a voltage sensing resistor.	This voltage change results from electrical circuit tampering and notifies the owner of voltage changes.
15	2, 6, 7, 8, 9	¼” metal housing protects the additional U-LARM circuitry for physical standards comparable to the U-Lock.	In order to prevent thieves from disarming the alarm and accessing the circuitry, this housing isolates the circuitry from external influences.
	Marketing Requirements 1. Low Cost 2. Durable 3. Energy Efficient 4. Minimalistic Design 5. Easy to Use 6. Tamper Proof 7. Effective Theft Deterrent 8. Wireless Notification 9. Audible Notification		

TABLE II
U-LARM DELIVERABLES

Delivery Date	Deliverable Description
April 1st , 2015	Design Review
April 20th, 2015	Completion of Initial Design Decisions
May 10th, 2015	Finished Circuit Design
May 25th , 2015	EE 461 demo
May 29th , 2015	EE 461 report
June 5th ,2015	Built Design Prototype
Aug 15th, 2015	EE 462 demo
Aug 30th, 2015	ABET Sr. Project Analysis
Aug 30th, 2015	EE 462 Report

Chapter 3: Functional Decomposition (Level 0 and Level 1)

Level 0 Block Diagram

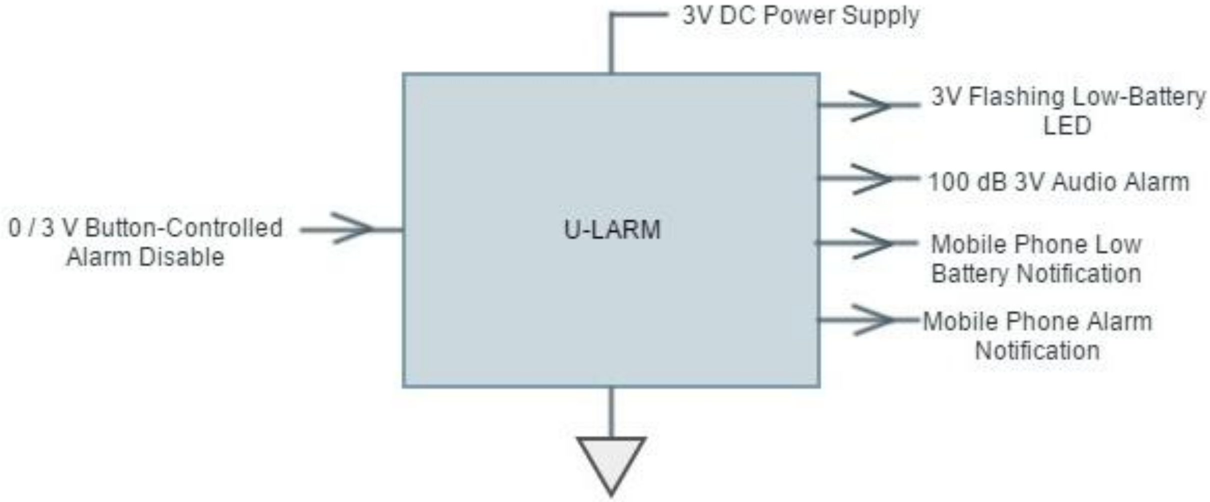


FIGURE 1: U-LARM LEVEL 0 BLOCK DIAGRAM

Figure 1 illustrates the level 0 block diagram of the U-LARM accessory. The system inputs and outputs, as listed in *Table III*, help define the functionality requirements of the system.

TABLE III
U-LARM LEVEL 0 BLOCK DIAGRAM FUNCTIONAL REQUIREMENTS

Module	U-LARM
Inputs	- User Button: Disables the system's alarm functionality using 3V and 0V button-controlled microcontroller detection. - Battery: Powers the circuit with a 3.7 V Lithium Polymer Battery. - System Ground: Provides ground for all the internal electrical components.
Outputs	- Audio Alarm: Sounds 100 dB audible alarm at 3V. - Low Battery Alert: Blinks at a rate of 12 flashes / minute at 3V when the battery falls below 20% capacity. - Mobile Phone Low Battery Notification: Alert on a mobile device when the battery falls below 20% capacity. - Mobile Phone Alarm Notification: Notifies the owner on his or her mobile device when resistive voltage sensor detects a 15 mV change.
Functionality	- Monitors the U-LARM system to detect electrical circuit tampering. Upon changes to this circuitry, accessory outputs an audible alarm until the depletion of the power supply or the disabling of the user. This circuit runs at 0V and 3V DC high and low voltages.

Level 1 Block Diagram

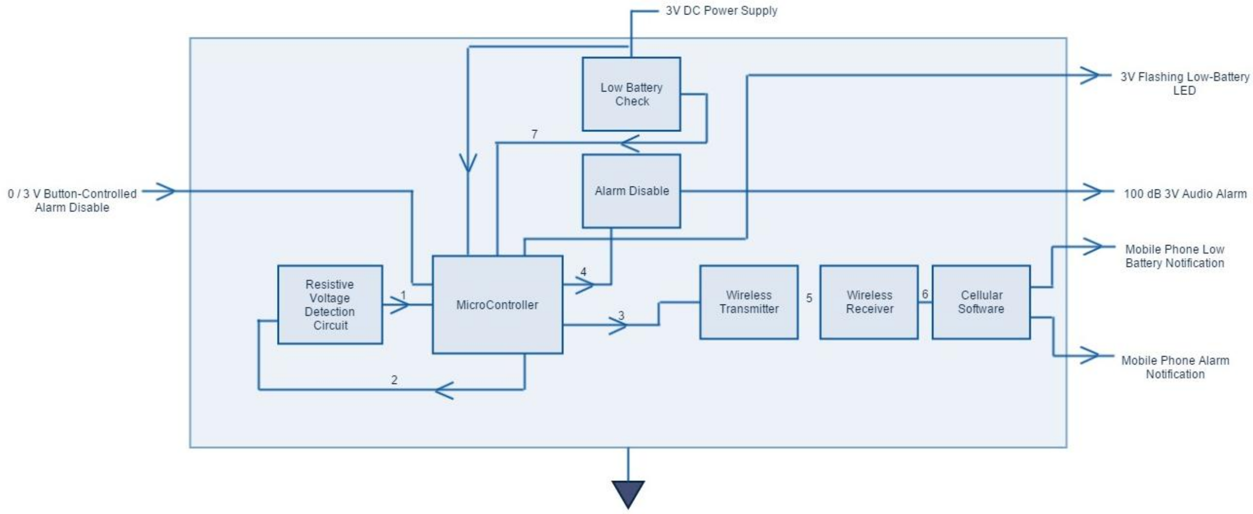


FIGURE 2: U-LARM LEVEL 1 BLOCK DIAGRAM

Figure 2 illustrates the level 1 block diagram of the U-LARM bike accessory. *Table III* contains the input and output signal descriptions. Numbers 1 through 7 designate the internal signals in the level 1 block diagram. *Table IV* describes the format and purposes of these signals in more detail.

TABLE IV
U-LARM LEVEL 0 BLOCK DIAGRAM FUNCTIONAL REQUIREMENTS

Signal Identification Number	Signal Description
1	- Sensed voltage across the resistive circuit with a 0-3 V analog value. - Microcontroller ADC input for circuit tampering detection.
2	- 3V signal from microcontroller for voltage detection circuit.
3	- Wireless transmitter power line. - Transmission signal for receiver.
4	- 3V enable signal for the alarm speaker.
5	- Wireless communication signal connecting transmitter and receiver.
6	- Internal cellular circuitry connecting receiver data to software.
7	- 0 to 3V analog low battery signal connecting voltage regulator to microcontroller analog input.

Chapter 4: Project Planning (Gantt Chart and Cost Estimates)

Project Timeline

TABLE V: EE 460 U-LARM TIMELINE

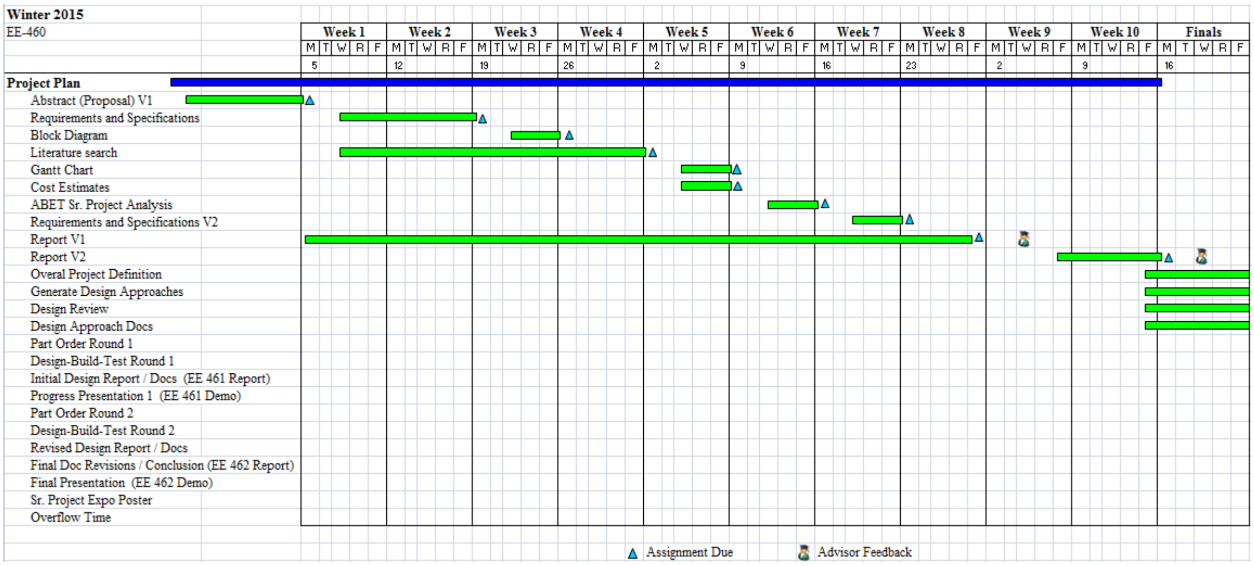


TABLE VI: EE 461 U-LARM TIMELINE

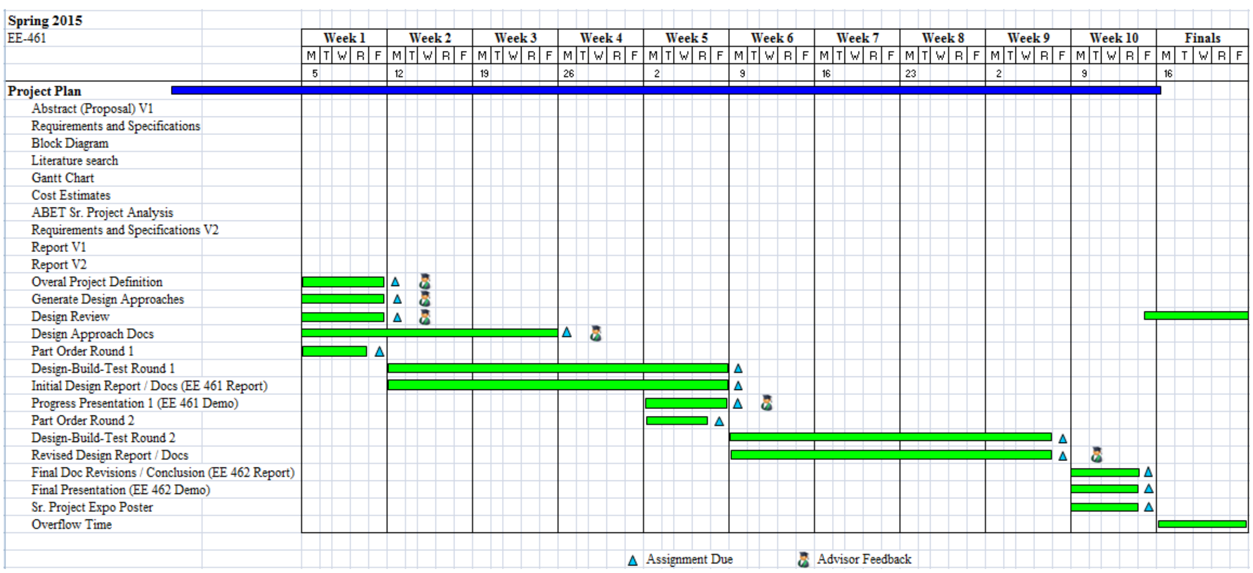
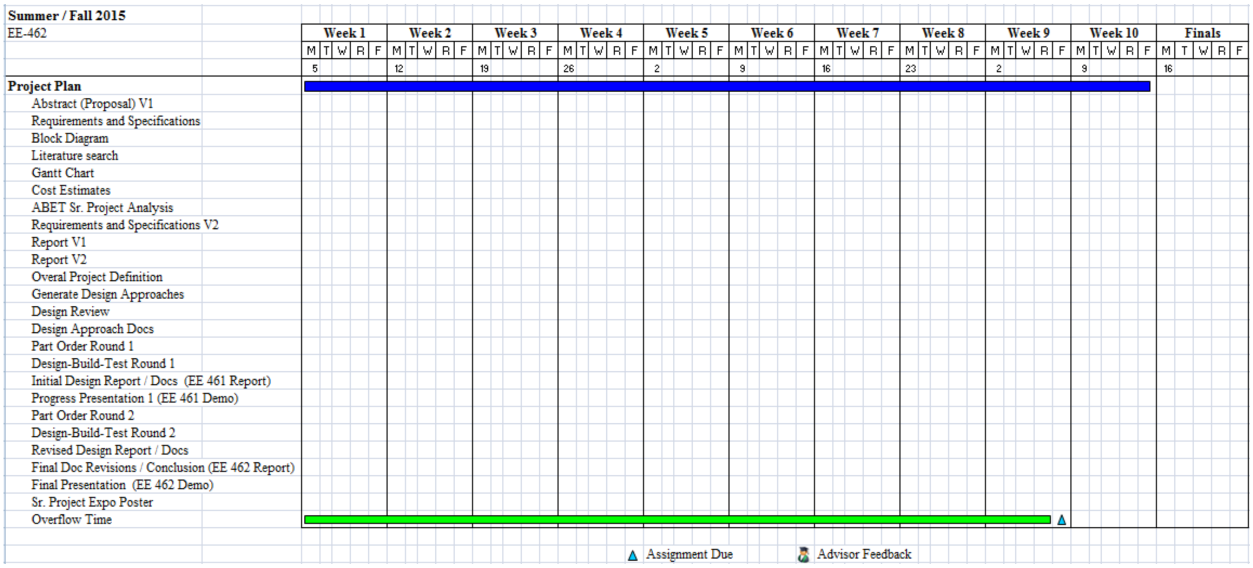


TABLE VII: EE 462 U-LARM TIMELINE



Tables V to VII show my senior project completion by the end of Cal Poly's spring quarter in 2015. This deadline allows for additional time in the summer or fall of 2015 to finish the project if I have fallen behind. The project consists of two design-build-test cycles. Each of these cycles produces its own documentation for inclusion in the final product report.

Cost Analysis

TABLE VIII
U-LARM Product Cost

Product Cost			
Material	Quantity	Price Per Unit (ppu)	Total Cost
Resistors	11	0.25	2.75
Capacitors	7	0.5	3.5
TL082 Op-Amp	1	0.83	0.83
Lithium Ion Battery	1	9.95	9.95
Atmega328P	1	4.2	4.2
Inductor	1	1	1
Solder	1	7.33	7.33
Micro USB Connector	1	2.26	2.26
16 MHz Crystal	1	0.36	0.36
Schottky Diode	1	0.34	0.34
Buttons	2	0.41	0.82
MCP73833	1	0.94	0.94
NCP1402	1	0.8	0.8
LEDs	6	0.07	0.42
3-point switch	1	2.49	2.49
Speaker	1	6.95	6.95
Spring	2	0.5	1
Construction Fabric	1	2.5	2.5
Conductive Thread	1	2.95	2.95
PCB Board	1	9.3	9.3
U-Lock	1	35	35
Wi-Fi Adaptor (opt)	1	30	30

TABLE IX

U-LARM Labor Cost

Labor Cost			
Personel	Total Hours	Hourly Rate	Total Cost
Designer	150	25	3750
Manufacturer	5	12.5	62.5

TABLE X

U-LARM Total Cost in Dollars

Total Cost	**3908.19**

Table VIII, IX, and X approximate the U-LARM product costs. The creation of the current / voltage sensor, the Wi-Fi communication hardware, and the audible alarm sound require each of the items listed in the product cost table. The costs in this table provide for the creation of one U-LARM accessory. The labor costs table includes the total number of design and manufacturing hours necessary for the completion of this project. By summing the costs of these tables, we arrive at a labor cost estimate of approximately $4,000.

Design

Basic Description

The U-LARM design utilizes resistive thread technology to monitor bike lock tampering. Through basic resistive divider principles, the Atmega328P microcontroller monitors the output voltage between the varying lock resistance and the fixed resister value. When the U-LOCK turns on, push-button calibration stores the output voltage and allows the microcontroller to sound the alarm and send a text message if the voltage change exceeds 10 percent of the calibrated voltage. A small, rechargeable 3.7V Lithium Ion battery and a custom PCB allow this design to utilize a small spatial footprint. The following paragraphs detail the specifics of each of the sub-components.

Conductive Sleeve

In order to run wire around the outside of the U-Lock, I needed to find a specific type of material. The small resistance of plain copper wire allows thieves to bypass the detection system through short-circuit tampering. By placing a wire between the two ends of the “U” portion of the lock, the thief opens up the opportunity to cut the lock’s steel without breaking the circuit. In a similar way, placing resistors around the lock fails to account for this type of tampering, it just makes it a bit more complicated. Through reading the resistor values, the thief can simply replicate the locks equivalent resistance. In addition, holding each of those resistors in place around the lock would require additional space through some sort of housing. To simplify the design, I decided to use conductive thread [29] around the lock. This thread, purchased at Sparkfun, maintains a fairly consistent resistance rating throughout the thread. For example, the stainless steel thread that I chose to use has approximately 28 Ω per foot. This allows me to have one big resistor around the “U” portion of the lock with approximately 265 Ω.

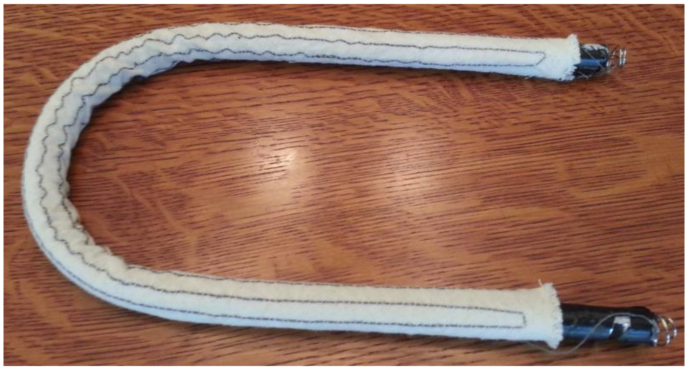


FIGURE 3: U-LARM FABRIC SLEEVE

To wrap this thread around the lock and isolate it from the elements, I developed a sleeve design to hold the thread in place. This design incorporates durable utility fabric for the sleeve and a simple zigzag stitch to hold the conductive thread to the sleeve. When designing this sleeve, I attempted to use a running stitch, but the sewing machine broke the conductive thread in the middle of each stitch attempt. *Figure 4* shows the final stitch layout, but I also tried zigzagging the conductive thread in the opposite direction on the opposite side of the material as is shown in *Figure 5*. Although this helped further increase the difficulty of finding a place to cut the lock, the conductive thread on each side resided too close to the thread on the opposite side and significantly decreased the resistance of the lock.

FIGURE 4: FINAL CONDUCTIVE THREAD PATTERN

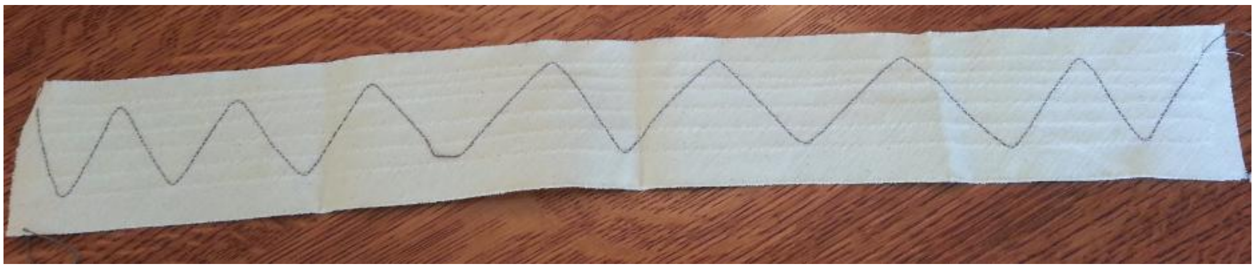


FIGURE 5: ATTEMPTED ADDITIONAL ZIGZAG PATTERN

To make the lock detachable, I needed to find a way to securely make an electrical connection to the microcontroller on the flat part of the lock. After brainstorming, using battery pack springs on the "U" portion of the lock and a conductive plate on the flat portion of the lock surfaced as the best method. A demonstration of this technique is shown in *Figure 3*.

Speaker

For this speaker of the U-LARM, I used a GT-0950RP3 magnetic transducer from Digilent Inc.'s analog parts kit. Although not waterproof, this speaker demonstrates the ability to produce an 89 dB alarm when provided a 5 V_p square wave at a frequency of 3200 Hz [28]. For more information on this speaker, see the part datasheet. The "Conclusions and Recommendations" section contains more information for future speaker development.

Power

Adafruit sells a 3.7 V Lithium Ion Polymer battery [30] which I chose to use for this U-Lock accessory. This battery utilizes compact packaging with the ability to supply 4.2 Wh of power. This rechargeable battery offers an eco-friendly solution to avoid unnecessary disposable batteries. In order to recharge this battery, though, the accessory requires some of the additional circuitry shown in Appendix E. This circuitry is based around the MCP73833 Li-Ion charge controller chip [19]. Removing some of the unnecessary components left me with the final circuit section shown in *Figure 9*. For user benefit, four LEDs are used in the final design that alert of the user of certain conditions. A red LED blinks for half of a second every 5 seconds to alert the user of a depleted battery supply that needs recharging. This blinking attempts to save power in a state where sustained battery life is of dire importance. A second red LED tells the user if a micro USB connection is connected to the battery. A third yellow LED and fourth green LED signal if the battery is being charged or is fully charged respectively.

For the GPRS text message functionality to work properly, the Atmega328P has to be able to supply a 5 V output. Using the NCP1402 PFM Step-Up Micropower Switching Regulator [20], the U-LARM accessory is able to meet this voltage requirement. Appendix F shows the recommended circuit layout which I utilized in our final design shown in *Figure 10*.

Hardware Components

The Atmega328P chip operates as the controller for the U-LARM accessory. After removing some of the unnecessary Arduinio Uno components, the final Atmega328P utilizes a reset button with a pull-up resistor and a 16 MHz crystal clock. The layout of this circuitry can be seen in *Figure 7*.

Additional hardware components include a 3-point On-Off switch, a calibration push button, a reset button, and a voltage follower. The 3-point switch, purchased from Radio Shack, allows the user to disconnect the battery from the rest of the circuitry. This reduces unnecessary battery consumption when the lock doesn't need to be armed. Since the switch lies between the battery junction and the 3.7V to 5V DC converter as seen in *Figure 6*, the battery can still charge even with a switch in the "Off" position. To allow for potential resistance changes in the conductive thread over time, this accessories uses a button to take in the voltage in the middle of the resistive divider each time the circuit is powered on. Upon receiving power, the Atmega328P illuminates a yellow LED to indicate the setup mode state of the controller where it awaits a resistive divider voltage. Rather than keeping the LED on at all times, the software blinks this LED with the purpose of saving power. The periodicity in which this LED illuminates can easily be altered in the code. Upon pressing the button, the yellow LED permanently turns off as the controller saves the analog voltage. The controller compares every other voltage check to this saved voltage until the user resets or turns off the accessory. This system also uses a reset button to reset the controller. In case a weird bug causes the alarm to sound, the user can reset the system using this button. Finally, the design of this system utilizes a voltage follower to pass the resistive divider voltage to the analog pin of the Atmega328P. In order to avoid resistive loading, I used the TL082 op-amp in the simple voltage follower configuration as seen in *Figure 8*. Each of these components helped create a fully functional system.

For the breadboard prototype, I designed a system that allows the user to send a text message to his or her cellular device. To accomplish this task, I used a GPRS text message shield purchased at Radio Shack. The visual graphics and schematics are located in Appendix G. By simply attaching this shield to the Arduino Uno and un-commenting the text message code in Appendix H, one can obtain working text message functionality. Through simple code alterations, the user can customize phone number and message information to their personal liking. This feature also allows the accessory to alert the user upon low battery conditions with a text message.

Software

The software component of this accessory contains two basic components. The system requires certain setup criteria in order to complete the system initialization. After initializing the circuit, the system goes into a main loop where it awaits an external change to the circuit. These two areas along with some supporting code make up the software portion of this system. Appendix H lists all of the finalized software used in this system.

The setup portion of this code performs a lot of the initialization required for a properly functioning alarm system. This includes the initialization of serial communication needed by the Sim900 chip of the GPRS shield as well as the input / output nature of each of the Atmega328P pins. In this section of code, the controller measures the calibrated voltage and the battery voltage levels. In addition to these obvious things, the setup code also initializes the Jeelab watchdog timer that helps optimize the system for low power applications. [32] This library helps keep the controller in a low power state for as long as possible to minimize the necessity of wasting energy. Once the code has completed all of these tasks, the controller enters the main loop of the code.

In the main loop, the code awaits an “alarm” state in which the battery voltage drops below 3.5 V or the resistive divider voltage varies from the calibrated voltage by a factor of 10 percent. If the system fails either of these conditions, the accessory sounds the alarm and sends the user a text message, assuming the system offers the text message functionality. Each iteration of the main loop calls a sleep cycle of 5 seconds. This low power setting decreases power consumption dramatically by essentially putting the Arduino to sleep until an internal timer wakes it up.

PCB Design

The PCB design process involved learning how to use the Eagle PCB software to create a custom PCB for this alarmed bike lock application. After finalizing a circuit design, I used the software to create an all-inclusive circuit schematic for the PCB using the breakout board schematics and my breadboard design. This process involved creating custom components in the software to match the individual components and their datasheet package specifications. After creating all of the individual parts and connecting them in the correct fashion, I finished the schematic portion of the PCB. Appendix C shows the finished design for this project.

By using the finished schematic, I generated the board layout for the PCB. This process involved orienting components in such a way as to minimize the area of the final design. By placing the parts in the layout shown in Appendix D, I simplified the amount of internal routing required for the PCB fabrication while maintaining a 2” by 2” spatial footprint.

Upon receiving the components and the fabricated board, I spent time soldering on all of the individual components. Most of the components I chose were through-hole components, but some of the parts used surface mount technology. Since the surface mount components are naturally smaller, soldering them onto the board correctly proved a bit challenging and required close attention.

Testing

Upon constructing the PCB the first time, the device didn't work as expected. The green light power light and the setup mode LED didn't illuminate as it was programmed to do. Similarly, two of the three charging LEDs (with the exception of the red power LED) didn't work as discussed in the design portion of this paper. In addition, the alarm didn't sound like it was supposed to do. These were some of the initial things that I needed to troubleshoot.

At this point in time, I decided to test some of the electrical connections to verify that the electrical connections were there as intended. I quickly found that I had a solder bridge on the MCP73833 charging chip. After fixing this solder bridge, I was able to make the yellow charge indicator LED illuminate when a micro USB charged the battery. Unfortunately, I still haven't been able to get the green LED to illuminate when the battery is fully charged.

When turning on the accessory with the switch, the green power LED illuminated temporarily but turned off shortly after. After a bit of voltage probing, I found out that, as I connected the battery to the circuit, the voltage at the battery terminals dropped to 5 mV. When I connected the circuit to the micro USB connector, this battery voltage didn't drop. As a result, I decided to remove the 5V DC converter by shorting the connection between the 3.7V battery node and the 5V "VDD" node. When I connected these nodes, the diode smoked a little bit and the device turned on. After disconnecting the micro USB, the device now worked properly but at a voltage of 3.7V as opposed to 5V. This led me to conclude that the parts that I chose for the 5V DC converter were causing the problems in my circuit. Since the PCB version doesn't require 5V to operate, the system now works correctly, but work needs to be done to figure out why the step-up converter doesn't work.

The original design my friend used with his initial version of this bike lock alarm utilized a vibration activated alarm. The goal for this second version hoped to remove some of the false alarms that caused customers satisfaction with his product to drop without affecting the effectiveness of the alarm. Throughout all of the testing phase of this product, no false alarms have occurred. By decreasing the 10% change specifications in the code, the user can make this device more sensitive to resistive tampering of the circuit.

Table XI shows the testing related to the specifications shown in *Table I*. This table lists whether the tests passed, failed or remain untested and discuss how I came to these testing conclusions.

TABLE XI
U-LARM REQUIREMENTS AND SPECIFICATIONS TESTING

Spec Number	Pass / Fail	Testing Description
1	Fail	By adding up the total costs in Table VIII, the additional cost excluding the U-LOCK costs sums to approximately 60 dollars. This price drops significantly if the manufacture purchases the components in bulk.
2	Pass	The U-LOCK battery lasts approximately 25 hours before needing recharging. This was tested through allowing the system to remain "On" without recharging the battery.
3	Pass	The U-LOCK blinks a red LED when the battery supply voltage drops below 3.5 V

4	Pass	The text message version of this device can transmit over a long distance far exceeding 50 yards due to the use of the cellular network. As long as the user has a data signal, the device will work properly.
5	Pass	Since this system uses a phone's built-in text message software, no additional costs arise through adding software to the phone.
6	Did Not Test	No stress or lifespan testing has been performed on this U-LARM accessory.
7	Pass	This U-LARM accessory weighs less than 3 ounces in total. This weight includes the battery, the PCB, and the sleeve.
8	Fail	The U-LARM PCB exceeds the 1 centimeter maximum by approximately 0.25 centimeters on each side of the device. The U-LOCK has a width of 3 centimeters while the PCB has a width of approximately 5.5 centimeters.
9	Pass	This U-LARM accessory does not alter the way in which the U-LOCK attaches to the bike. Since the user only has to turn on a switch and push a button, the 10 second time limit is easily achievable.
10	Fail	The datasheet for the chosen speaker can only produce an 89 dB alarm. In order to meet the 100 dB requirement, a new speaker must be chosen for this product.
11	Pass	This software for this code will cause the alarm to sound until the bike owner returns to their bike. As a result, this alarm meets the 5 minute requirement.
12	Did Not Test	This project focused primarily on the electronic components of the U-LARM. Further development needs to be made before performing this test.
13	Pass	Since this PCB attaches to the bike lock and the U-LOCK receives no mechanical alterations, the strength and structure of the U-LOCK meet this requirement.
14	Pass	The software of this code can change to allow the microcontroller to sense a voltage change less than 10% of the calibrated voltage.
15	Did Not Test	This project focused primarily on the electronic components of the U-LARM. Further development needs to be made before performing this test.

Conclusions and Recommendations

This project focused on the electrical design of a simple system. Through the use of basic circuit and programming concepts, I created an alarm that activates on resistive changes as opposed to mechanical vibration. This alarm sensing change significantly reduces the amount of false alarms triggered by the U-Lock. By removing some of the unnecessary Arduino Uno components, I decreased the overall size of the PCB. The U-LARM accessory also incorporates a rechargeable battery and the supporting circuitry for proper recharging techniques. This supports the project goal of creating an eco-friendly bike lock accessory that avoids waste from frequent battery disposal.

Although this project design works properly, multiple opportunities for improvement exist for the overall system. These areas include some of the electronics, mechanical components, overall system design, and PCB design. Each of these areas make important contributions to a well-rounded bike lock accessory. My primary focus lied in obtaining a U-LOCK accessory that utilized a non-vibration activated tamper sensing method. As a result, consumers need further improvements before they can really use this product.

Some improvements in the electronic components can improve the functionality and effectiveness of this system. Initial battery testing shows that the device is able to withstand 12+ hours of continual use without recharging the battery. Further developments can focus on increasing the battery life of this accessory through removing some of the LED indicators, altering resistance values to decrease power loss through the resistors, and using a more power efficient microcontroller. Another area of development focuses on improving the speaker effectiveness of the project. Increasing the volume or adding a voice alarm that signifies to bystanders of the theft in progress can effectively improve the alarm as a deterrent.

Without additional packaging developments, the only option for this accessory involves attaching it to a premade bike lock. Although this will work, it exposes much of the system to circuit tampering. A more optimal system would position the circuit inside part of the lock. The lock materials would then assist in isolating the circuit from the outside world. Since this circuit needs to be waterproof, encasing the PCB in metal will hide the circuit from elemental impacts. Similarly, improvements can occur in isolating the conductive thread and sleeve from the rain. Currently, the U-LARM makes use of electrical tape carefully wrapped around the entire "U" portion of the lock to protect it from harmful water effects. Through the use of some sort of plastic substance, this design could be optimized and made to look more like a standard bike lock.

Finally, the PCB design of this system can be optimized for this product. One can change the dimensions of the board to fit it inside a specific container. Similarly, smaller components can replace some of the ones that I chose to use for this project. Typically, surface mount components require a smaller spatial footprint on the PCB board but are typically harder to use. One could also use surface mount components for this system to further reduce the size of the board. Lastly, one can improve the design of this system by incorporating the text message functionality into the PCB hardware design. This would take the overall product to the next level for consumers.

Bibliography

Literature Search

[1] A. Baum, *A Link to the Internet of Things: IoT made easy with SimpleLink Wi-Fi solutions* [online]. Texas Instruments, Dallas, TX. Tech. Paper. Jul. 2014. Available: http://www.ti.com/lit/wp/swry009/swry009.pdf [Accessed: Feb 2, 2015].

[2] Atmel Corporation, “Low current Microcontroller for Wireless Communication,” ATARxxx datasheet, 2007. Available: http://www.atmel.com/Images/doc4535.pdf [Accessed: Feb 2, 2015].

[3] B. W. Kernighan and D. M. Ritchie, *The C Programming Language*, 2nd ed., Upper Saddle River, NJ: Prentice Hall PTR, 1988.

[4] *CC3100/CC3200 SimpleLink Wi-Fi Network Processor Subsystem: Programmer’s Guide* [online]. Texas Instruments, Dallas, TX. Tech. Report. Jun. 2014. Available: http://www.ti.com/lit/ug/swru368/swru368.pdf [Accessed: Feb 2, 2015].

[5] G. Perry, *Absolute Beginner’s Guide to C*, 2nd ed., Indianapolis, IN: Sams Publishing, 1994.

[6] G. Reiter, *A Primer to Wi-Fi Provisioning for IoT Applications* [online]. Texas Instruments, Dallas, TX. Tech. Paper. Jul. 2014. Available: http://www.ti.com/lit/wp/swry011/swry011.pdf [Accessed: Feb 2, 2015].

[7] *IEEE Std 1233, 1998 Edition*, p. 4 (10/36), DOI: 10.1109/IEEESTD.1998.88826

[8] Itech Solutions, ‘New Times SLO | News | Be gone, bike thieves', Newtimesslo.com, 2015. [Online]. Available: http://www.newtimesslo.com/news/9554/be-gone-bike-thieves/. [Accessed: 24- Feb- 2015].

[9] M. A. Acena and Y. Ghabbour, “Isolated Resistive Current Sensor,” U. S. Patent 8,836,525, 16 Sept., 2011.

[10] Mammano, Bob. *Current Sensing Solutions for Power Supply Designers* [online]. Texas Instruments, Dallas, TX. Tech. Paper. 2001. Available: http://www.ti.com/lit/ml/slup114/slup114.pdf [Accessed: Feb 2, 2015].

[11] P Bode, *Current Measurement Applications Handbook* [online]. Zetex Semiconductors, Oldham, UK, Tech. Paper. Jan 2008. Available:http://www.diodes.com/_files/products_appnote_pdfs/zetex/an39.pdf [Accessed: Feb 2, 2015].

[12] Perera, C.; Liu, C.H.; Jayawardena, S.; Chen, M., "A Survey on Internet of Things From Industrial Market Perspective," Access, IEEE , vol.2, no., pp.1660,1679, 2014
doi: 10.1109/ACCESS.2015.2389854. Available:
http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=7004894&refinements%3D4291944246%26queryText%3DInternet+of+Things [Accessed: Feb 2, 2015].

[13] R. Ford and C. Coulston, *Design for Electrical and Computer Engineers*, McGraw-Hill, 2007, p. 37

[14] *Sensing Elements for Current Measurements* [online]. Intersil Corporation, DE, Tech. Paper. 2014. Available: http://www.techonline.com/electrical-engineers/education-training/tech-papers/4432671/Sensing-Elements-for-Current-Measurements/viewpdf [Accessed: Feb 2, 2015].

[15] T. Regan, J. Munson, G. Zimmer, and M. Stokowski, "Making Sense of Current," in *Current Sense Circuit Collection* [online]. Linear Technology, Milpitas, CA, Appl. Note 105. Dec 2005. Available: http://cds.linear.com/docs/en/application-note/an105fa.pdf [Accessed: Feb 2, 2015].

[16] Texas Instruments, "AMC1305x High-Precision, Reinforced Isolated Delta-Sigma Modulators," ACM1305x datasheet, Jun. 2014 [Revised Dec. 2014]. Available: http://www.ti.com/lit/ds/symlink/amc1305l25.pdf [Accessed: Feb 2, 2015].

[17] "Wi-Fi," in *Wikipedia: The Free Encyclopedia* [online]. Updated: Feb 2, 2015. Available: http://en.wikipedia.org/wiki/Wi-Fi [Accessed: Feb 2, 2015].

[18] Y. Zhen, *Current Sensing Circuit Concepts and Fundamentals* [online]. Microchip Technology Incorporated, Chandler, AZ, Tech. Paper. 2010. Available: http://www.techonline.com/electrical-engineers/education-training/tech-papers/4204475/Current-Sensing-Circuit-Concepts-and-Fundamentals/viewpdf [Accessed: Feb 2, 2015].

[19] Microchip Technology Inc., "Stand-Alone Linear Li-Ion / Li-Polymer Charge Management Controller," MCP73833 datasheet, March 2009. Available: https://www.adafruit.com/datasheets/MCP73833.pdf [Accessed: Aug 7, 2015].

[20] ON Semiconductor, "200 mA, PFM Step-Up Micropower Switching Regulator," NCP1402 datasheet, May 2007. Available: http://cdn.sparkfun.com/datasheets/Tools/NCP1402-D.pdf [Accessed: Aug 7, 2015].

[21] Adafruit, "USB LiIon/LiPoly Charger – v1.2," Product ID: 259. Available: https://www.adafruit.com/products/259 [Accessed: Aug 7, 2015].

[22] SparkFun, "SparkFun 5V Step-Up Breakout – NCP1402," Product ID: PRT-10968 ROHS. Available: https://www.sparkfun.com/products/10968 [Accessed: Aug 7, 2015].

[23] Jimb0, "Using EAGLE: Schematic," SparkFun. [Online]. Available: https://learn.sparkfun.com/tutorials/using-eagle-schematic [Accessed: Aug 7, 2015].

[24] Jimb0, "Using EAGLE: Board Layout," SparkFun. [Online]. Available: https://learn.sparkfun.com/tutorials/using-eagle-board-layout [Accessed: Aug 7, 2015].

[25] Texas Instruments, "TL08xx JFET-Input Operational Amplifiers," TL082 datasheet, Feb. 1977 [Revised May 2015]. Available: http://www.ti.com/lit/ds/symlink/tl082.pdf [Accessed: Aug 7, 2015].

[26] Atmel Corporation, “Atmel 8-Bit Microcontroller with 4/8/16/32Kbytes In-System Programmable Flash,” Atmega328P datasheet, Oct. 2014. Available: http://www.atmel.com/Images/Atmel-8271-8-bit-AVR-Microcontroller-ATmega48A-48PA-88A-88PA-168A-168PA-328-328P_datasheet_Summary.pdf [Accessed: Aug 7, 2015].

[27] ArduinoFun, “Build Your Own Arduino,” Instructables. [Online]. Available: http://www.instructables.com/id/Build-Your-Own-Arduino/ [Accessed: Aug 7, 2015].

[28] Soberton Inc., “Magnetic Transducer,” GT-0950RP3 datasheet, Oct. 2008. Available: http://www.soberton.com/wp-content/uploads/2012/06/GT-0950RP3-Spec-Sheet.pdf [Accessed: Aug 15, 2015].

[29] SparkFun, “Conductive Thread Bobbin – 30 ft (Stainless Steel),” Product ID: DEV-10867 ROHS. Available: https://www.sparkfun.com/products/10867 [Accessed: Aug 15, 2015].

[30] Adafruit, “Lithium Ion Polymer Battery – 3.7v 1200 mAh,” Product ID: 258. Available: http://www.adafruit.com/products/258 [Accessed: Aug 15, 2015].

[31] Seeed Studio, “GPRS Shield V2.0,” in *Seeed Wiki*. [online document], 2015. Available: http://www.seeedstudio.com/wiki/GPRS_Shield_V2.0 [Accessed: Aug 28, 2015].

[32] JeeLib, “Sleepy Class Reference.” Available: http://jeelabs.net/pub/docs/jeelib/classSleepy.html [Accessed: Aug 28, 2015].

Appendix A: Hardware Layout / Configuration

Figure 6 displays the finalized hardware layout of the U-LARM accessory. The text message configuration with the GPRS Arduino Uno shield matches the non-text message layout of the product. *Figures 7 to 10* show the Eagle PCB circuit schematic used to fabricate the custom PCB.

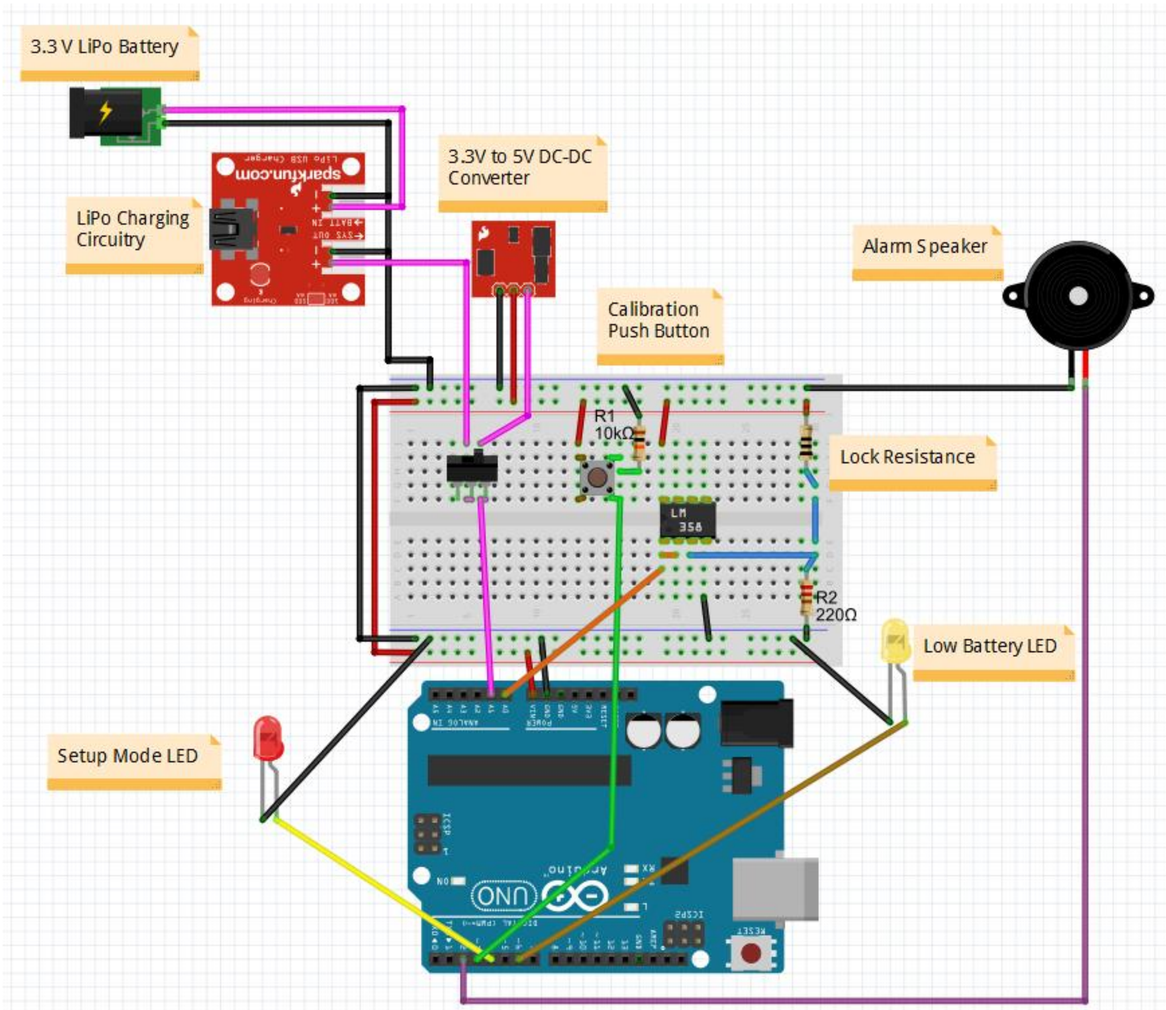


FIGURE 6: U-LARM HARDWARE LAYOUT

The Atmega328P chip and supporting circuitry, shown in Figure 10, simplifies the Arduino Uno design by removing unnecessary circuit hardware including a large number of I/O ports, a 5 V regulator, and the circuitry required to program the microcontroller. This allows for minimal PCB size for the U-LARM accessory. At the same time, this design utilizes many circuit concepts from the Arduino Uno including a reset button with a pull-up resistor, an LED illuminated when the microcontroller is powered, and a 16 MHz oscillator required by the Atmega328P. In addition to these components, I have added a low battery LED, a setup mode LED, a calibration button and a speaker to the circuit.

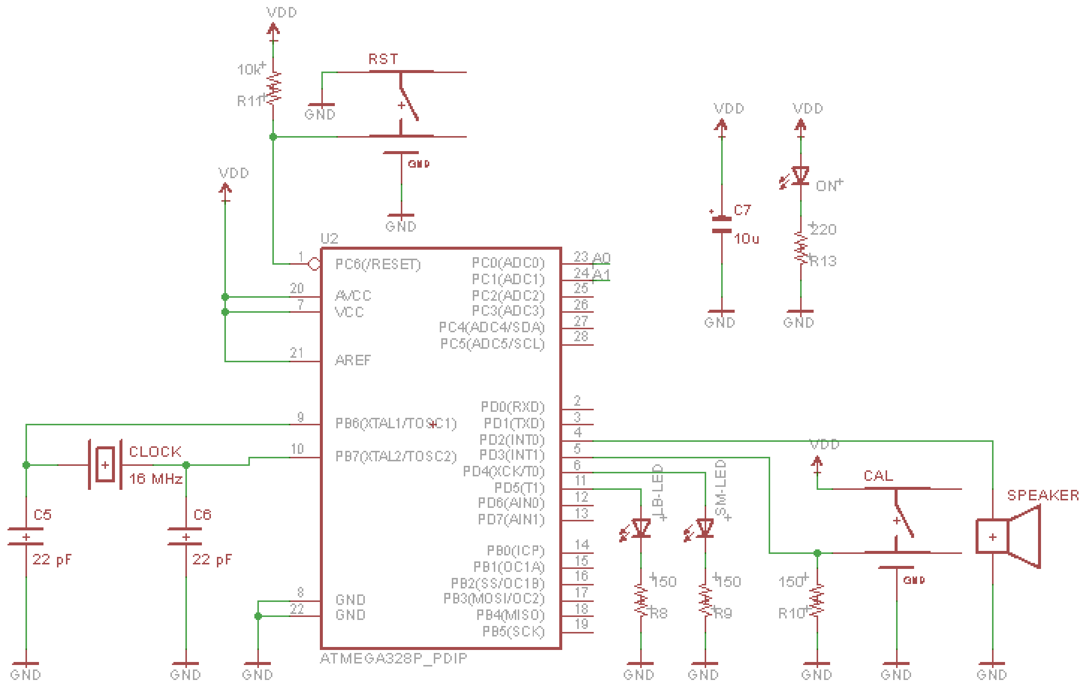


FIGURE 7: EAGLE PCB MICROCONTROLLER SCHEMATIC

Figure 11 shows the resistive divider of the U-LARM design with the TL082 Op-Amp voltage follower. The A0 connection of this figure passes the voltage between the resistors to the Analog 0 pin on the Atmega328P chip.

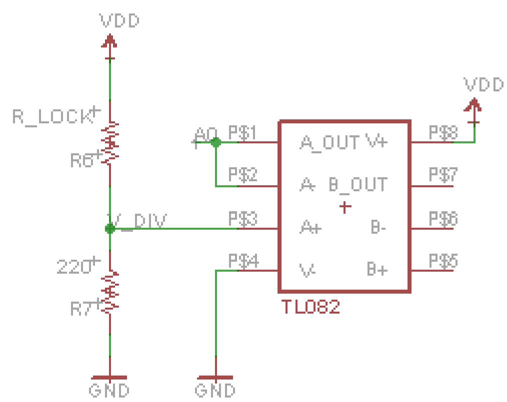


FIGURE 8: EAGLE PCB RESISTIVE DIVIDER SCHEMATIC

The circuit portion shown in Figure 12 follows the design of Adafruit's 3.7 V Lithium Polymer battery charging breakout which uses the MCP73833. This version of their design removes a few unnecessary JST connectors and resistors in an attempt to minimize PCB size criteria. The design still uses a Micro USB port to charge the LiPo battery connected to the 2-port symbol on the right side of the circuit. This circuit design has an LED that illuminates to inform the owner that the circuit has power from the Micro USB port. It also utilizes two state LEDs that inform the user of a charging and fully charged battery. Adafruit designed this circuit to charge the LiPo battery with 500 mA of charge current. More information can be obtained on adafruit's website [21] and on the MCP73833 datasheet [19]. See Appendix E for Adafruit's full circuit schematic.

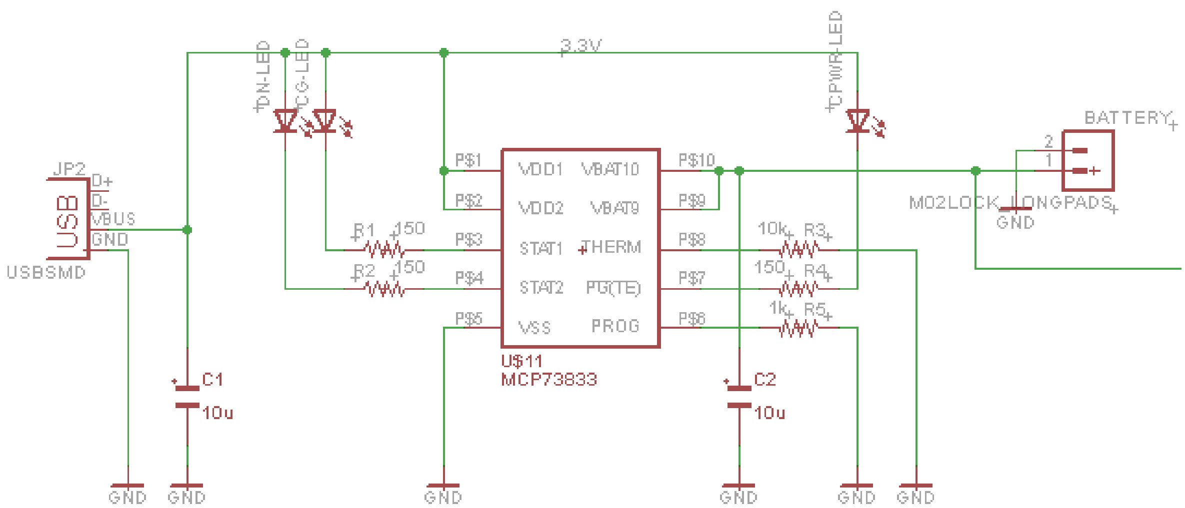


FIGURE 9: EAGLE PCB LIPO CHARGING SCHEMATIC

Figure 13 shows the final portion of the final U-LARM design. This portion of the schematic contains the On-Off switch and the 3.3V to 5V DC converter modeled after Sparkfun's step-up breakout. This chip uses ON Semiconductor's NCP1402 chip which has the capability of outputting 5V at a maximum of 200 mA. More information can be found on Sparkfun's website [22] and on the NCP1402 datasheet [20]. See Appendix F for Sparkfun's full circuit schematic.

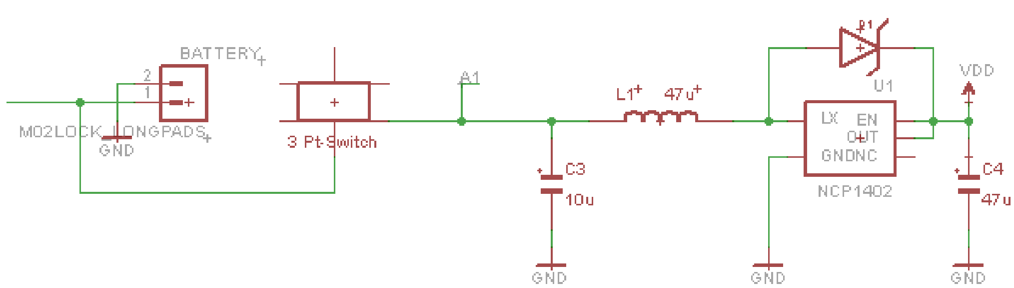


FIGURE 10: EAGLE PCB 5V STEP-UP CONVERTER SCHEMATIC

Appendix B: PCB Board Layout

Using the Eagle PCB software, I have obtained the board design shown below in *Figure 11*. This PCB contains all of the components shown in *Figures 7 to 10*. After meeting their design requirements, this design was sent to OSH Park for fabrication. *Figures 12 and 13* show the PCB without the components soldered onto the board. *Figure 14* shows the finished U-LARM PCB. *Table X1* identifies the parts shown in *Figure 14*.

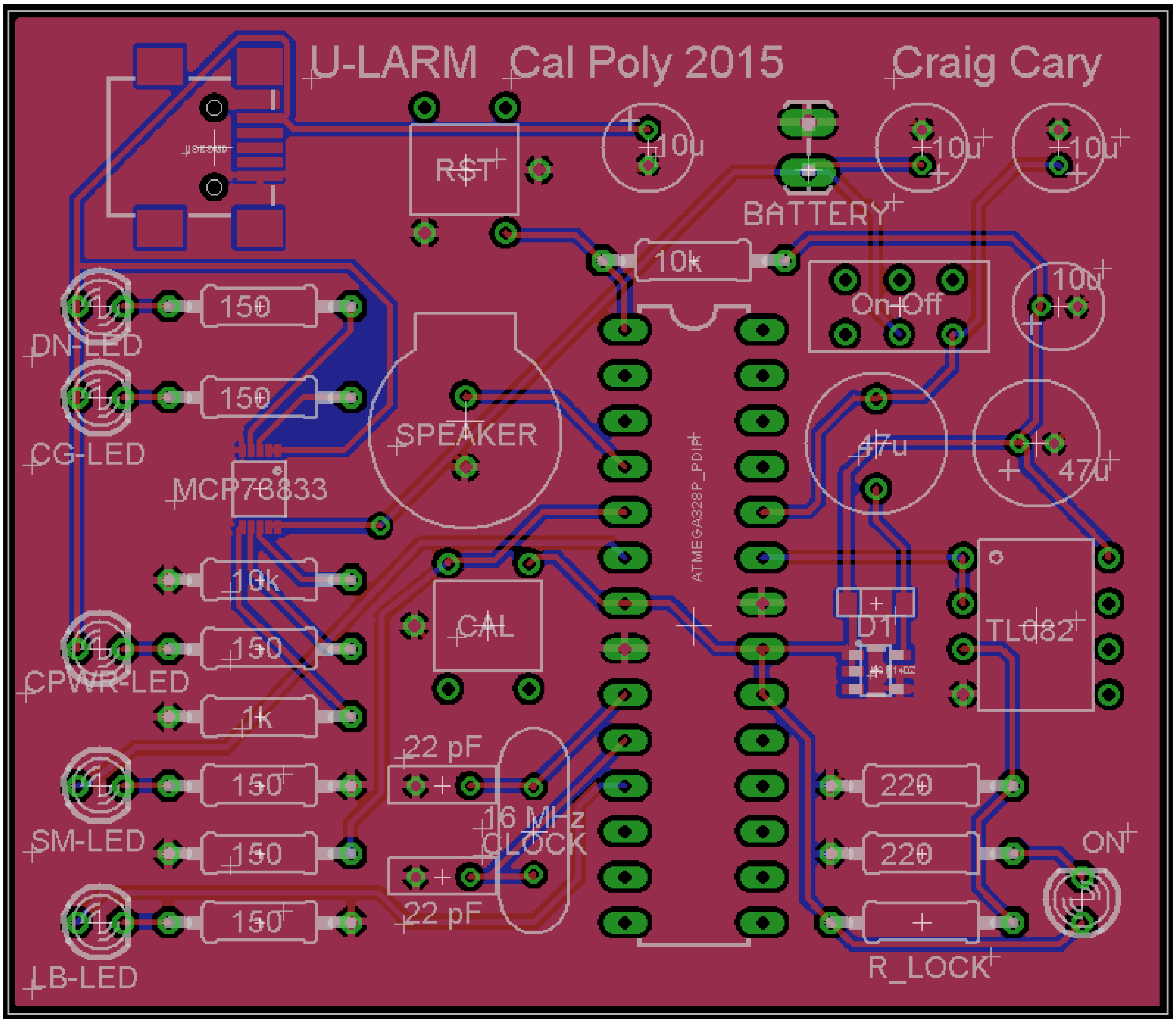


FIGURE 11: EAGLE PCB BOARD LAYOUT

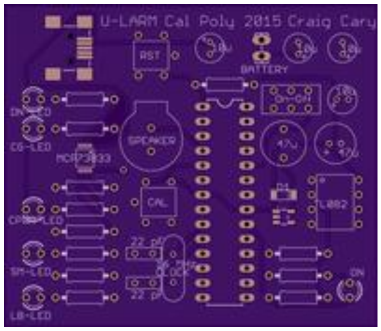


FIGURE 12: EAGLE PCB FRONT

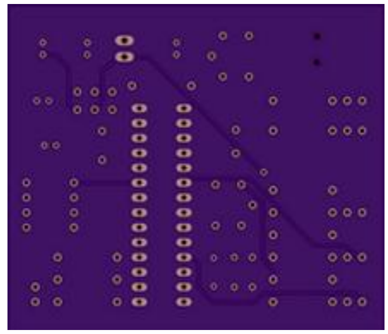


FIGURE 13: EAGLE PCB BACK

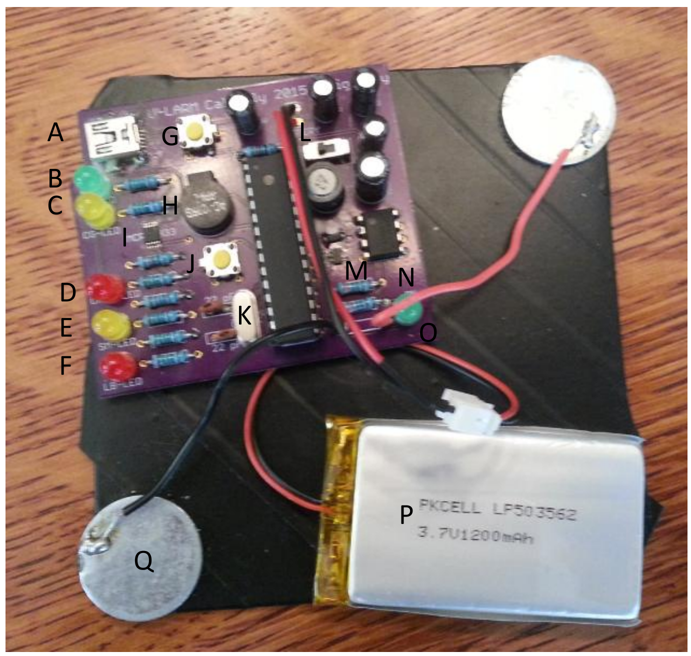


FIGURE 14: FULLY CONSTRUCTED PCB

TABLE XII
PCB PART IDENTIFIER

Letter	Part Description	Letter2	Part Description3
A	Micro USB Connection	J	Calibration Button
B	Fully Charged LED	K	16 Mhz Crystal Oscillator
C	Charging-in-Progress LED	L	On-Off Switch
D	Charge Circuitry Power LED	M	NCP1402 Step-up Converter
E	Setup Mode LED	N	TL082 Op-amp
F	Low Battery LED	O	Power LED
G	Reset Button	P	3.7V LiPo Battery
H	Speaker	Q	U-Lock Connection Points
I	MCP73833 Charging Chip		

Appendix C: Adafruit Lithium-Polymer Charging Circuitry

Figure 15 shows Adafruit's full circuit diagram for their Lithium-Polymer charging breakout board. [21]

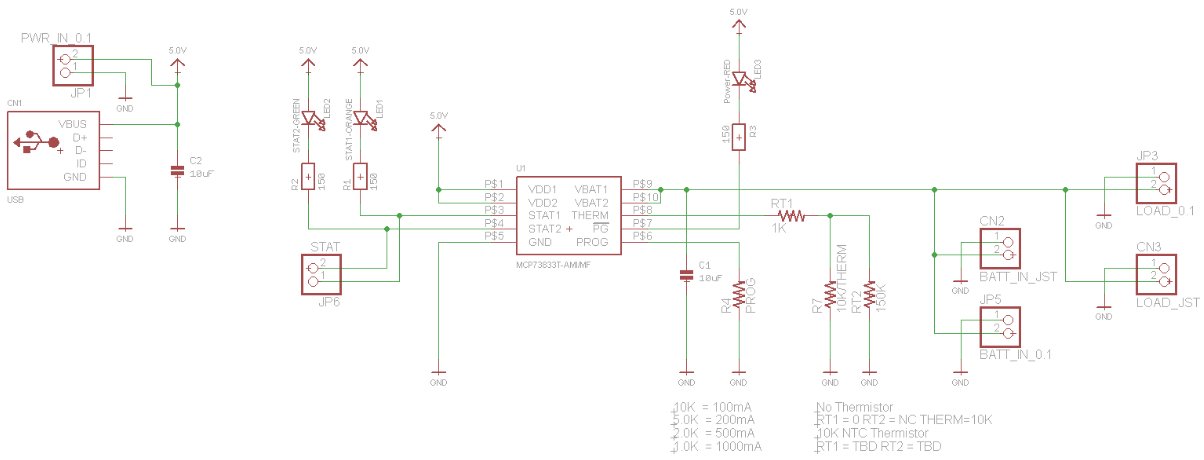


FIGURE 15: ADAFRUITS LIPO CHARGING SCHEMATIC

Appendix D: Sparkfun 5V Step-up Circuitry

Figure 16 shows Sparkfun's full circuit diagram for their 5V Step-up breakout board. [22]

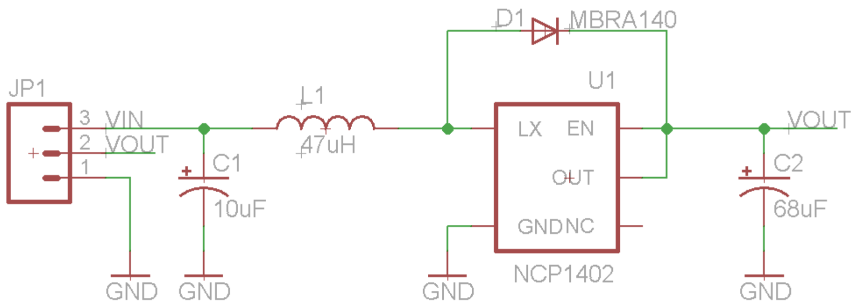


FIGURE 16: SPARKFUN'S 5V STEP-UP CONVERTER SCHEMATIC

Appendix E: Seeed Studio GPRS Shield V2.0

Figures 17 and 18 show Seeed Studio's GPRS shield that I used for this project. The smaller rectangular piece is the antenna for the shield. On the bottom of the shield, one can see the SIM card holder. Since modern phones have a Micro Sim card and this shield is compatible with the normal Sim card, I cut out a normal Sim Card sized holder for the Micro Sim card using an old credit card. Since the electrical connections are identical, this allowed me to create a simple adapter for the project.

FIGURE 17: SEED STUDIO GPRS SHIELD FRONT

FIGURE 18: SEEED STUDIO GPRS SHIELD BACK

Figures 19 and 20 show the circuit schematic used to create this text messaging shield. The eagle files of these schematics and these images can be found on Seeed Studio's GPRS wiki page. [31]

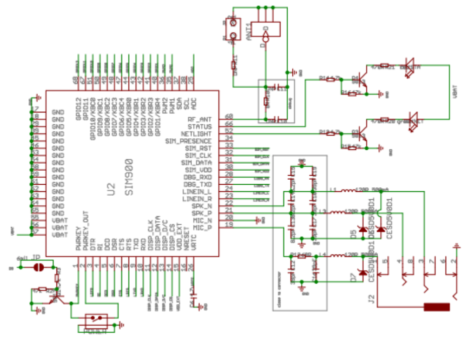


FIGURE 19: SEEED STUDIO GPRS SHIELD SCHEMATIC PART 1

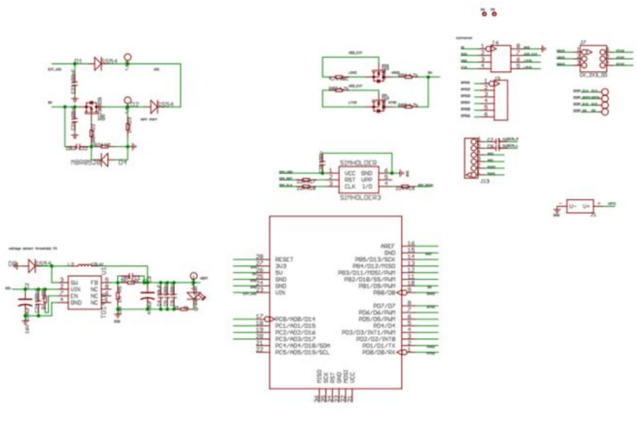


FIGURE 20: SEEED STUDIO GPRS SHIELD SCHEMATIC PART 2

Appendix F: Program Software with Alarm Functionality Commented

```
//-----------------------------------------------------
//------------------Includes---------------------------
//-----------------------------------------------------

#include <JeeLib.h>
#include <String.h>
//#include <SoftwareSerial.h>



//-----------------------------------------------------
//--------Variable Constants---------------------------
//-----------------------------------------------------

#define SleepModePin 6
#define LowBattCheckPin 5
#define AlarmPin 2
#define Button 3
#define AnalogInPin A0
#define LowBattPin A1
#define SetupModeLED 4
//#define SERIAL_BAUD 19200

//SoftwareSerial gprsSerial(7,8);



//-----------------------------------------------------
//--------Internal Variables---------------------------
//-----------------------------------------------------

double batVoltage;
double calVoltage;
int BattTextSentCheck = 0;
int StealTextSentCheck = 0;
//String number = "1----------";
//String StealText = "Hi Todd! This text means Bike Stolen";
//String LowBattText = "Hi Todd! This text means Low Battery";
```

```
//-------------------------------------------------------
//----------------------Setup--------------------------
//-------------------------------------------------------

// Initialize the Watchdog
ISR(WDT_vect) { Sleepy::watchdogEvent(); }

void setup() {
  // Initialize Pin 5 to Check for Incoming Signal
//  pinMode(StealTextSentCheck, INPUT);

  // Initialize LED Pin to On
  pinMode(SetupModeLED, OUTPUT);
  pinMode(LowBattCheckPin, OUTPUT);
  digitalWrite(SetupModeLED, HIGH);
  digitalWrite(LowBattCheckPin, LOW);

//  gprsSerial.begin(SERIAL_BAUD); // GPRS shield baud rate
//  Serial.begin(SERIAL_BAUD);

  // Set Up calVoltage and batVoltage Global Variable to check
  calVoltage = 0;
  batVoltage = -1;

  // Set Digital Ouput Pin for Resistive Sensing Circuit
  pinMode(AlarmPin, OUTPUT);
  pinMode(Button, INPUT);

  // Set Arduino Analog Pin Reference Voltage
  // AREF Connected to 3.3V Arduino Voltage Pin
  analogReference(DEFAULT);

  while(calVoltage == 0){
    // Blinks LED to Indicate Setup Mode
    blinkLED(SetupModeLED);

    // Wait for calVoltage to be Initialized
    if(voltCalibrate() != -1){
      // Serial.println("Taking in New Value");
      calVoltage = voltCalibrate();
      digitalWrite(SetupModeLED, LOW);
    }

    delay(500);
  }

  // Taking in Battery Information
  batVoltage = analogRead(LowBattPin)*(5.0/1023);

  delay(500);
}
```

```
//-------------------------------------------------------
//------------------Main Loop---------------------------
//-------------------------------------------------------

void loop() {

//  Serial.flush();
  Sleepy::loseSomeTime(5000);
//  Serial.begin(SERIAL_BAUD);

  // Variable Declaration
  int calCheck = 1;    // Check for Voltage Match

  // If Button is Pressed, Save new Voltage
  if(voltCalibrate() != -1){
    calVoltage = voltCalibrate();
  }

  // Checks to see if Current Analog Voltage Matches Calibrated Voltage
  calCheck = voltMismatch(calVoltage);

  // Checks for Low Battery State
  lowBattCheck();
}
```

```
//-------------------------------------------------------
//--------Voltage Mismatch Check-----------------------
//-------------------------------------------------------

// Returns -1 if the Analog Voltage Matches the Calibrated Voltage and 1 if
they do Match
// Sends Text Message if Mismatched
int voltMismatch(double calVolt){
  double AnRead = 0;

  // 1 Means Values Match
  double MatchCheck = 1;

  // Read in Analog Pin Voltage
  AnRead = analogRead(AnalogInPin);

  // Convert the Analog Reading into a Voltage
  double AnVoltage = AnRead*(5.0/1023);

  if( (AnVoltage > (1.1 * calVolt)) || (AnVoltage < (0.9 * calVolt)) ){
    // -1 Means Values don't Match
    MatchCheck = -1;

//    // Send Text Message
//    if(StealTextSentCheck == 0){
//      // Sound Alarm
    buzzerAlarm();
//
//      StealTextSentCheck = 1;
//      SendTextMessage(number,StealText); // send the text message
//    }
//  } else {
//    StealTextSentCheck = 0;
  }

  return MatchCheck;
}
```

```
//---------------------------------------------------------
//------Voltage Calibration Check-------------------------
//---------------------------------------------------------

// Returns -1 if Button isn't pressed or Returns Analog Voltage if Button is
Pressed
double voltCalibrate() {
  // Default Return Value
  double AnRead = 0;
  double AnVoltage = -1;

  // If Butotn is Pressed
  if( digitalRead(Button) ){
    digitalWrite(SetupModeLED, HIGH);

    // Read in Analog Pin Voltage
    AnRead = analogRead(AnalogInPin);

    // Convert the Analog Reading into a Voltage
    AnVoltage = AnRead*(5.0/1023);

    digitalWrite(SetupModeLED, LOW);
  }

  return AnVoltage;
}


//---------------------------------------------------------
//--------Low Battery Check Code--------------------------
//---------------------------------------------------------
void lowBattCheck() {
  // Default Return Value
  double lowBattRead = 0;
  double lowBattVolt = -1;

  // Read in Low Battery Analog Pin
  lowBattRead = analogRead(LowBattPin);

  // Convert the Analog Reading into a Voltage
  lowBattVolt = lowBattRead*(5.0/1023);

  if(lowBattVolt < 3.5) {
    // Send Text Message
    blinkLED(LowBattCheckPin);    // Blink LED for Half a Second

//    if(BattTextSentCheck == 0){
//      BattTextSentCheck = 1;
//      SendTextMessage(number,LowBattText); // send the text message
//    }
//  } else {
//      BattTextSentCheck = 0;
  }
}
```

```
//-------------------------------------------------------
//------------Buzzer Code--------------------------------
//-------------------------------------------------------

void buzzerAlarm(){
  for(int h = 0; h < 10; h++){
    for(int i = 0; i < 1000; i++){
      delayMicroseconds(156);
      digitalWrite(AlarmPin,HIGH);
      delayMicroseconds(156);
      digitalWrite(AlarmPin,LOW);
    }
  }
}



//-------------------------------------------------------
//------------Blinking LED Code--------------------------
//-------------------------------------------------------

void blinkLED(int pin){
  digitalWrite(pin,HIGH);
  // LED is Turned on for Half of a Second
  delay(500);
  digitalWrite(pin,LOW);
}
```

```
//-------------------------------------------------------
//-------- Text Message Code---------------------------
//-------------------------------------------------------

/*
* Name: SendTextMessage
* Description: Send a text message to a number
*/
//void SendTextMessage(String number, String text)
//{
//  Serial.println("Sending Text...");
//  gprsSerial.print("AT+CMGF=1\r"); // Set the shield to SMS mode
//  delay(100);
//
//  // Concatenate Number into String
//  String numberFormatString = "AT+CMGS = \"" + number + "\"";
//
//  // Convert String to Char Array
//  char numberFormatChar[25];
//  numberFormatString.toCharArray(numberFormatChar, 25);
//
//  // Send an SMS Message in the Form of 1xxxxxxxxxx
//  gprsSerial.println(numberFormatChar);
//  delay(100);
//
//  // Insert Message Content Here
//  gprsSerial.println(text); //the content of the message
//  delay(100);
//
//  gprsSerial.print((char)26);//the ASCII code of the ctrl+z is 26 (required
according to the datasheet)
//  delay(100);
//  gprsSerial.println();
//  Serial.println("Text Sent.");
//}
```