T.A.A.G

Tamper Automated Alert Gadget

GROUP 7:

Aiman Salih	EE
Daniel Gibney	CpE
Leaphar Castro	EE

TERM:

Fall 2015

FUNDING:

Dr. Jiann-Shiun Yuan
co-Director of NSF MIST Center

Contents

1 EXECUTIVE SUMMARY ---- 1

2 INTRODUCTION AND PROJECT SUMMARY ---- 2

2.1 OUR AIM AND MOTIVATION ---- 2
2.2 MEMBERS OF THE GROUP 7 ---- 2
2.3 FUNDING SOURCES AND OVERALL MILESTONES ---- 3
2.4 MIST CENTER ---- 3

3 THE PROJECT'S DESCRIPTION ---- 4

3.1 GOALS AND OBJECTIVES ---- 4
3.2 REQUIREMENTS SPECIFICATIONS ---- 4
3.2.1 Scalability ---- 5
3.2.2 Communication ---- 5
3.2.3 Power Usage and Charging ---- 5
3.2.4 Notification ---- 5
3.2.5 Sensitivity Settings and Trigger Thresholds ---- 5
3.2.6 Size and shape ---- 6
3.2.7 Mass ---- 6

4 DEFINING THE PROJECT ---- 6

4.1 EXISTING PRODUCTS ---- 6
4.1.1 Samsung SmartThings ---- 6
4.2 RELEVANT TECHNOLOGIES ---- 7
4.3 STRATEGIC COMPONENTS ---- 7
4.4 ARCHITECTURE CHOSEN AND BLOCK DIAGRAMS ---- 8

5 IMPORTANT PROJECT STANDARDS ---- 9

5.1 SAFETY STANDARDS ---- 9
5.2 BATTERY MONITORING STANDARDS ---- 10
5.2.1 Battery Recycling and Disposal Standards ---- 11
5.3 OTHER RELATED ELECTRICAL STANDARDS ---- 11
5.4 QUALITY STANDARDS ---- 12
5.5 GENERAL BATTERY STANDARDS ---- 12
5.5.1 Lithium Battery Standards ---- 13
5.5.2 Nickel Metal Hydride Battery Standard ---- 14

6 DESIGN CONSTRAINTS IN COMPLETING THE PROJECT ---- 14

6.1 ECONOMIC AND TIME CONSTRAINTS ---- 15
6.1.1 Time ---- 15
6.1.2 Money ---- 15
6.2 ENVIRONMENTAL, SOCIAL, AND POLITICAL CONSTRAINS ---- 16
6.3 ETHICAL, HEALTH, AND SAFETY CONSTRAINTS ---- 17
6.4 MANUFACTURABILITY AND SUSTAINABILITY CONSTRAINTS ---- 18

7 COMMUNICATIONS SYSTEM FROM A HARDWARE PERSPECTIVE ---- 18

7.1 THE INTERNET OF THINGS AS A SOLUTION ---- 18

7.2 INITIAL IDEA OF USING A BASE STATION FOR COMMUNICATION ---- 19

8 LIGHT SENSING FROM A HARDWARE PERSPECTIVE ---- 20

8.1 RESEARCH ON HARDWARE MEANS OF SENSING ---- 20
8.2 PARTS AND INTEGRATED CIRCUITS USED TO EMPLOY LIGHT SENSING ---- 22
8.2.1 TAOS TSL2561 LIGHT-TO-DIGITAL CONVERTER: ---- 23

9 MOTION SENSING FROM A HARDWARE PERSPECTIVE ---- 24

9.1 RESEARCH ON HARDWARE MEANS OF SENSING ---- 24
9.1.1 Bosch BMA222 3-AXIS ACCELEROMETER DIGITAL SMD. ---- 26

10 MICROCONTROLLER FROM A HARDWARE PERSPECTIVE ---- 27

10.1 INITIAL RESEARCH FINDING AN OPTIMAL MCU ---- 28
10.2 CC3200 SIMPLELINK™ WI-FI® AND INTERNET-OF-THINGS SOLUTION, A SINGLE-CHIP WIRELESS MCU 28
10.3 CC3200MOD OVER THE CC3200 ---- 29
10.4 HARDWARE FOR DEVELOPMENT ENVIRONMENT ---- 30
10.5 MCU PIN CONNECTIONS ---- 32

11 MONITORING SYSTEM ---- 36

11.1 SYSTEM-SIDE IMPEDANCE-TRACK FUEL GAUGE WITH DIRECT BATTERY CONNECTION-BQ27510-G3 ---- 36

12 BATTERY TYPES ---- 37

12.1 COIN CELL ---- 37
12.2 NICKEL METAL HYDRIDE (NIMH) ---- 38
12.3 ALKALINE ---- 39
12.4 LITHIUM POLYMER: ---- 40
12.4.1 Nominal Voltage ---- 40
12.4.2 Connectors ---- 40
12.4.3 Charging and Discharging ---- 40

13 BATTERY PROTECTION ---- 41

13.1 LI-ION/LI POLYMER BATTERY PROTECTION IC-BQ29700 ---- 41

14 BATTERY COMPARISON & CONCLUSION ---- 42

14.1 USB CONNECTOR ---- 43
14.2 CABLING ---- 43
14.3 SLEEP-AND-CHARGE PORTS ---- 44
14.4 TRANSMISSION RATES ---- 44
14.5 MINI-USB &MICRO-USB ---- 45

15 CHARGING CIRCUIT ---- 46

15.1.1 Texas Instruments BQ24210 ---- 46

16 VOLTAGE REGULATOR ---- 46

16.1 SWITCHING REGULATORS ---- 47
16.2 FUNDAMENTALS ---- 47
16.3 TOPOLOGIES ---- 47
16.3.1 Topology 1 ---- 48
16.3.2 Topology 2 ---- 48

16.3.3 Topology 3 ---- *49*
16.3.4 Topology conclusion ---- *49*

17 TESTING ---- **50**

17.1 ELECTRICAL SYSTEM TESTING ---- 50
17.1.1 Charging and Battery ---- *50*
17.1.2 Voltage Regulator ---- *50*

18 OVERVIEW OF THE SOFTWARE SYSTEM ---- **50**

19 MICROCONTROLLER SOFTWARE ---- **52**

19.1 OVERVIEW ---- 52
19.2 PROVISIONING AND TRANSMITTING ADDITIONAL DATA ---- 53
19.3 NETWORK COMMUNICATION ---- 54
19.4 NOTIFICATIONS ---- 55
19.5 POWER MANAGEMENT ---- 55
19.6 TO RTOS OR TO NOT RTOS? ---- 57
19.7 LONG TERM STORAGE THROUGH FILES ---- 57
19.8 INTERRUPT HANDLING ---- 59
19.9 GPIO ---- 59
19.10 SENSORS ---- 60
19.10.1 The Light Sensor ---- *60*
19.10.2 The Motion Sensor ---- *63*
19.11 BATTERY MONITORING ---- 64

20 WEB SERVICE ---- **65**

20.1 OVERVIEW ---- 65
20.2 DATA BASE ---- 65
20.3 ACCOUNT CREATION ---- 66
20.4 LOGGING ---- 67

21 MOBILE APPLICATION ---- **67**

21.1 OVERVIEW ---- 67
21.2 NOTIFICATIONS ---- 68
21.3 PROVISIONING AND SETTINGS TRANSFER ---- 68
21.4 USER INTERFACE ---- 69

22 SOFTWARE SYSTEM TESTING AND MILE STONES ---- **70**

22.1 DETECTOR ---- 70
22.2 MOBILE APP ---- 72
22.3 WEB SERVICE ---- 72

23 PROGRAMMING THE CC3200 ---- **73**

24 ENCRYPTION AND SECURITY ---- **74**

24.1 LIMITATIONS ---- 74

25 BATTERY LIFE ---- **75**

26 INITIAL PROJECT PROTOTYPING AND TESTING ---- **77**

27 FINAL PROTOTYPE ASSEMBLY ---- **80**

27.1 PCB DESIGN SOFTWARE ---- 80
27.1.1 OrCAD PCB Editor and Allegro ---- *81*
27.1.2 Altium Designer ---- *81*
27.1.3 EAGLE CAD 7.4.0 ---- *82*
27.2 EAGLE SCHEMATICS OF FINAL PROTOTYPE ---- 82
27.3 EAGLE BOARD OF FINAL PROTOTYPE ---- 88
27.4 BILL OF MATERIALS (BOM) OF THE SYSTEM ---- 90
27.5 PLASTIC 3-D PRINTED ENCASING ---- 92
27.6 MASS OF THE OVERALL SYSTEM ---- 93

28 FINAL PROTOTYPE TESTING ---- 94

29 CONCLUDING REMARKS ---- 94

APPENDIX A: REFERENCES ---- 95

APPENDIX B: STANDARDS REFERENCE NAME ---- 98

APPENDIX C: PERMISSION EMAILS ---- 100

1 Executive Summary

There are plenty of objects around a household which you may not want other people to touch. These may range from the serious, for instance, pharmaceuticals, a safe, or a gun cabinet, to the mundane items, like a roommate's goal of preventing someone from eating all of their breakfast cereal. The goal here is to solve all of these problems. This can be done by building a wireless, low cost, and easy to use tamper detection system which can be placed on a variety of items and can notify you on your phone when that item has been interacted with.

A major application of this technology may be for parents who are concerned with which items their children are interacting with. If this was the system's true niche, it might be tempting to dub the detectors "don't-touch-me's". The ability to know that your child has just picked up or moved something they're not supposed to and then to call them seems, although a devastating blow for the inner child inside all of us, a thing of value to the concerned parent. As such, the system should be consumer friendly and not require much, if any, networking knowledge to set up and should then require minimal interaction to maintain. Its user interface should be easily understood as well.

Aside from being low cost and easy to use here are some more technical details, it should be small and light weight (ideally enough so to adhered to some flat surface), battery powered with a long battery life, and easily adjusted so that it can effectively detect the tampering of any object in several ways (detecting motion and/or a change in light would be the primary means). The system as a whole should be expandable. In other words, there should be an option to add more detectors to a system when needed. Low-battery notifications seem necessary. Additional features might include an off-line mode which can record data while the device is unable to directly notify the user. Encryption can also be provided if it was desired for some purpose.

In terms of what exists that is similar to this, it can be said that although there are products for specific items such as gun cases, doors and windows, etc..., a quick on-line search turns up no results for such a small, general purpose device as this one. Moreover, the cost of these devices is typically several hundred dollars. A preliminary estimate of the cost of our system gives no indication of why it should cost this much. From there it is our goal to deliver a product with more efficiency and lower cost.

As we delved forward into the design phase of the system we discovered that a better name for the device must be employed in order for it to be more marketable, more memorable, and more recognizable. Hence we change it from the less interesting name of "don't-touch-me's" to a more catchy and robust name. That is where that name T.A.A.G came about, an acronym with both meanings when read as is and when broken down. The acronym can be broken down as Tamper Automated Alert Gadget, a name that is both descriptive and enables the acronym to make sense.

Users of the product can now simply state that "I have T.A.A.Ged" whatever object of interest they have. For example a user can now say "I have T.A.A.Ged this gun case, now I can get a notification on my phone if anyone moves it or opens it up." This can come

quite in handy as far as the marketability is concerned, it makes the product name easier to reference from the user's point of view.

2 Introduction and Project Summary

This section discusses some introductory information on the project.

2.1 Our AIM and Motivation

Upon forming our Senior Design group and choosing the Tamper Automated Alert Gadget as our project, as a group we made it such that our primary focus is delivering a product that was robust and complete in every sense of the word. We made it our AIM and focus, after researching and coming with the system design of the project, to deliver a product that was complete and has lowest possible amount of flaws.

By robust, complete, with the lowest possible amount of flaws we mean to have the system operate with very efficient power consumption, minimizing power losses within every possible angle. Also making sure that every feature that is available in the system has is fully functional and complete without any shortcomings that may hinder the user experience.

What was mentioned above is more on the system level. This robust system design as well stems from the innards of the device, the guts of the device, the components that it takes to make the device functional. Our group works on this project from start to finish, from coming up with the idea to choosing which components must go on the device, therefore to come up with the most optimal and robust operation of the system the group must be very mindful of the implications of choosing any component for the system. The propagation impact on one component can be quite impactful. In that, the group was very careful when selecting components looking at all of the specifications of the components from power draw to speed of operation.

2.2 Members of the Group 7

Senior Design Group 7 Consists of 3 members:-

Aiman Salih (majoring in EE), Daniel Gibney (majoring in CpE), and Leaphar Castro (majoring in EE).

The project under discussion requires a variety of expertise, backgrounds, and knowledge base. It can be shown that the project involves system design, hardware implementations of systems and hardware integration, it also involves power system design and power regulation of the different components of the system.

In addition there is a deep programing necessity for this project, since all of the analog components convert their data in to digital data that is processed and used by the MCU. Not to mention the use of a phone app that requires some understanding and development skills (all details to come later in this document).

The roles have more or less have been divided within the group as follows:

Aiman Salih: - Hardware, sensors, PCB layout, complete system integration.

Daniel Gibney: - Software, Programing MCU and sensors, App Development.

Leaphar Castro: - Hardware, power system, power regulation.

2.3 Funding Sources' Budget and Overall Milestones

With any project that requires a fair share of development the cost most likely always ends up being higher than anticipated due to the purchase of multiple parts for comparison, the purchase of development boards that normally have much more than what is needed, and other factors such as the damage of parts that can occur in the situation were the group falls into mistakes.

As such a source of funding is quite important in such a project, although it can bring an added layer of complexity to the design due to some requests that can be made by the funder, none the less it is overall most beneficial to have a person or a company taking care of the funds. Our group was able to secure a beneficiary that will be funding and covering all costs for this senior design project.

Funding for the senior design project comes from Dr. Jiann-Shiun Yuan who is a professor of electronics at the University of Central Florida and who is also the manager of a research lab on campus. Dr. Jiann-Shiun Yuan is the co-Director of the NSF's MIST Center at the University of Central Florida, he will be funding our project in total and it will come out the MIST Center budget that has been allocated towards funding senior design projects for student within the university.

The current budget as it stands is $700. As far as the overall milestones are concerned the current plan is to have the research and design complete by the middle of the semester (10/01/2015), the second milestone is to have the entire software system completed and the PCB fully designed and ready for printing by the end of the semester (December 2015). As for the milestones for the next semester we plan to have the PCB printed and the BOM purchased, and soldered on to the board by the 2nd week of the semester (January 2015), this will give us a sufficient amount of time to debug and make the sure everything is operational. Lastly we plan to have the entire project completed and ready by the middle of the semester (March 2015).

2.4 MIST Center

MIST Center stands for Multi-functional Integrated Systems Center, this center has a very special gathering of many cross-disciplinary experts. The range of diversity includes: -

Materials science, electronics, magnetics, acoustics, photonics, microfluidics, MEMS, power, and circuits.

The goal of this center is to tackle complex research challenges that are critical to the development of multi-functional integrated systems. This is where we believe our group's interest and our funder's interest intersected. The MIST Center has a heavy focus on the Internet of Things (IOT) as a solution to many of our everyday tasks, also the center has a focus on internet security and security algorithms. As such our project would provide an

excellent platform for some of the research students of the MIST Center at UCF to test and see their work on a real life system that users can interface with, rather than their daily simulation space that simply employs two points on the web and pings them back and forth.

3 The Project's Description

This portion of the document will focus more on the formal description of the project.

3.1 Goals and Objectives

Below is an itemized list of all of the goals that we have for our senior design project as a collective group:

- Robust and easy to use.
- Device to communicate over the internet to send tamper alerts to the user.
- Does not require much technical skill to setup.
- Have a companion app that is complementary with the product itself and can be used to interface with it and access all the features.
- Low cost. I.e. when the cost is compared to the other solutions that are available in the market as of the production of this document.
- Power efficient.
- Rechargeable battery.
- Can last a long time on single battery charge.
- Can give the user a live feed of the remaining battery life.
- Has a small size. Size small enough to fit onto many different everyday objects.
- Ergonomic feel.
- All electronics encased within a 3D printed case. Our current envisioning of the 3D printed case is for it to be that of a rectangular or circular shape.
- Pleasant and complete appearance.
- Employs different technology from what is out there in the market today.
- Uses the best and most cost efficient components.
- Completed on time for the final presentation.
- Satisfies all the requirements for Senior Design. This includes the use of a printed circuit board for the final prototype of the senior design project, and numerous other requirements.
  (More details will be given later in this document.)
- Satisfies our funder's requirements. The project funder requested that we add some encryption ability to the system, in order to increase security. (More details will be given later in this document.)

3.2 Requirements Specifications

Below is an itemized list of all of the requirements specifications that we have for our senior design project as a collective group:

3.2.1 Scalability

Up to 10 detectors can be used in a system. The user should be able to push a button on the detector and interface with the mobile app to have a new detector join the system. The user should be able select the detectors threshold settings during this process.

3.2.2 Communication

The detector or device must use wireless communication (Wi-Fi IEEE 802.11) to communicate tampering event and battery information over the Internet through a Wi-Fi access point.

Define a tampering event as a trigger threshold being crossed more than a minute apart in time from any other crossing of a trigger threshold.

3.2.3 Power Usage and Charging

The detector must be able to operate wirelessly for at least 720+ hours (30 days) with one or less tampering events being detected. The tapering event limitation comes from the necessity of the MCU to exit from its low power state and enter into full operation in order to push its notifications over the internet.

The re-powering process for the detector can use wires, but should charge to an operating voltage in less than 10 hours.

The charging must occur using a Universal Serial Bus (USB) standard-B micro connector (IEC 62680 "Universal Serial Bus interfaces for data and power" international standard) that can be powered from any source pushing USB Power-Delivery 1 (v1.0 or later), which has a capability of delivering up to 20V with 5A max.

3.2.4 Notification

The system must notify the user (connectivity permitting) of low power at least once while the detectors are within the range of 15% to 0% of needed operating power remaining. This must happen before a detector runs out of power.

A tampering event must trigger a notification for the user as long as Wi-Fi network connectivity is available and previously set up for the tamper detector device, and the user also has their mobile device connected to the internet with the companion application installed. In the case where there is no network connectivity for the user, the notification should be delivered to the user when possible, i.e. when internet connectivity has been reestablished.

3.2.5 Sensitivity Settings and Trigger Thresholds

A tamper detection device must be able to recognize when its dynamic acceleration crosses the user selectable variable threshold. Static acceleration (e.g. gravity, which give a constant towards the earth acceleration) should not trigger a tampering event, due its natural continual occurrence.

A detector must be able to detect when illuminance crosses a variable threshold. The user can select the ratio by which the light intensity changes from its startup value. Also, this

change in the intensity must last for a certain period of time, such that it can avoid triggering an interrupt due to slight spikes that may occur in the light intensity.

3.2.6 Size and shape

The detector can physically be one of the two forms:

-A cylindrical disk that is less than 70 mm in diameter (lateral dimensions) and less than15 mm in height (vertical dimension).
-A rectangle with sides less than 70 mm in length (lateral dimension), 50 mm (lateral dimension) in width, and a height less than 15 mm (vertical dimension).

3.2.7 Mass

The detector must be less than or equal to 50 grams in mass. Including all electronic components that are with the device and the plastic 3D printed incasing that holds all of the internal components.

4 Defining the Project

This portion will give the definition of the project and shed some light on some of the initial research that has been made for the project.

4.1 Existing Products

In the duration of the projects early research stages, the group surveyed the market to locate products that achieve similar results at what we are planning to design and build for our senior design project. Looking at what is out there in the market serves as a very insightful way to understand what is already fully developed and that is being used today and what can be done in order to further advance the design and build the product with more features, higher efficiency, and lower cost.

Below are a few products our survey has brought us to recognize the similarities to our design, but is already out there in the market:

4.1.1 Samsung SmartThings

Here is an excerpt from the official Amazon sale web link:

> *SmartThings lets you easily control, monitor, and secure your home from anywhere in the world. The heart of your smart home, the Samsung SmartThings Hub will connect all of your different smart locks, lights, outlets, thermostats, and more and let you control them from the free SmartThings app. Receive notifications about what's happening in and around your home and use your smartphone to remotely control your home's security, energy usage, lighting, and more. Since SmartThings is compatible with a wide variety of smart devices from different companies, once you have the SmartThings Hub and the free app for iOS, Android, or Windows, you can add as many additional SmartThings sensors or other popular Z-Wave, ZigBee, or Internet-connected products as you want to enhance your connected home. The Samsung SmartThings Hub works in the US and Canada.*

The SmartThings introduces a Hub which acts as a base station that collects information from all the other SmartThings sensors and channels that information to the user through their free mobile application.

SmartThings has one sensor that is similar to what we are constructing, it is called the Multipurpose Sensor. This sensor in summary employs a magnetic sensor that can be used to detect if a door or window has been opened, this occurs by placing or sticking the sensor on the door (or object of interest) and placing the magnet on the frame or any location where it is expected for the sensor and the magnet to be close to each other when the object of interest is considered to be closed. The Multipurpose Sensor also employs a vibration sensor and a temperature sensor for added awareness capabilities within the Samsung SmartThings environment.

Our group believes that it would be more efficient to use an accelerometer to detect motion rather that to a magnetic Reed switch type topology, this reduces the size and the overall cost of the product as well. We also believe that inclusion of the light sensor comes gives an additional degree of sensing that can be quite useful for the user.

The sensor itself has a cost of $40, this does not include the overhead cost of the Samsung SmartThings Hub which by itself costs $100. Altogether to employ a system that is similar to ours, it would cost the user around $140. This cost is much more than our group plans to build the product for.

4.2 Relevant Technologies

When doing researching for this senior design project our group acknowledged the technologies that were needed to be understood in more details in order to accomplish what was needed for the completion of the overall system.

Below is a list of all the technologies that are involved in the project:

- Wi-Fi IEEE 802.11 technology.
- USB IEC 62680 "Universal Serial Bus interfaces for data and power".
- Texas Instruments SimpleLink. IOT Wi-Fi capable MCU series.
- I^2C communication protocol between MCU and peripherals.
- Light Diodes, and their ability to measure light intensity.
- Accelerometers used as a means to sense sudden motion.

4.3 Strategic Components

In coming up with an optimal design for the project, throughout the research phase came components that were key to the completion of the project and that were needed to approach the end goal of making the system functional.

Below is the list of all of the key components that are needed to successfully achieve a system that is functional:

4.3.1 Development Boards

To be used for the initial proof of concept phase and the initial design phase.

4.3.2 Microcontroller

To be obtained separately from the development environment and to be placed with the final prototype (more details later in this document).

4.3.3 Breakout boards

Breakout boards with integrated components, or breakout boards with a surface mount chip installed to be used for the initial proof of concept phase and the initial design phase. More specifically:

- Light sensor (breakout board with components, or integrated circuit surface mount chip).
- Accelerometer (breakout board with components, or integrated circuit surface mount chip).
- Integrated circuit chips and surface mount parts are to be used in the final prototype (more details later in this document). This also more particularly includes:
    - Light sensor integrated circuit.
    - Accelerometer integrated circuit.

4.4 Architecture Chosen and Block Diagrams

The architecture of the system and how we have chosen to implement is summarized in this section below:

From a user's point of the view there are two major proponents that play and they are:-

1- The actual T.A.A.G. detector itself.
2- The phone app that is used as a main point of access for the user or as the user interface with the detector.

What occurs is a back and forth communication between that users mobile device (for example their cell phone or tablet that contains the companion app) and the tamper detection device. Where the tamper detection device would send information about tampering events and battery life information, it would also receive setup information form the user via the companion app. Similarly the user through their mobile device and companion app can receive information about tampering events and battery life information, and would also send setup information from the user to the tamper alert device.

The description above gives a "big picture" summary to the user's view of overall system and the direction we chose to go with it.

Finally here is a block diagram for that puts the words above in pictorial form (figure 1):

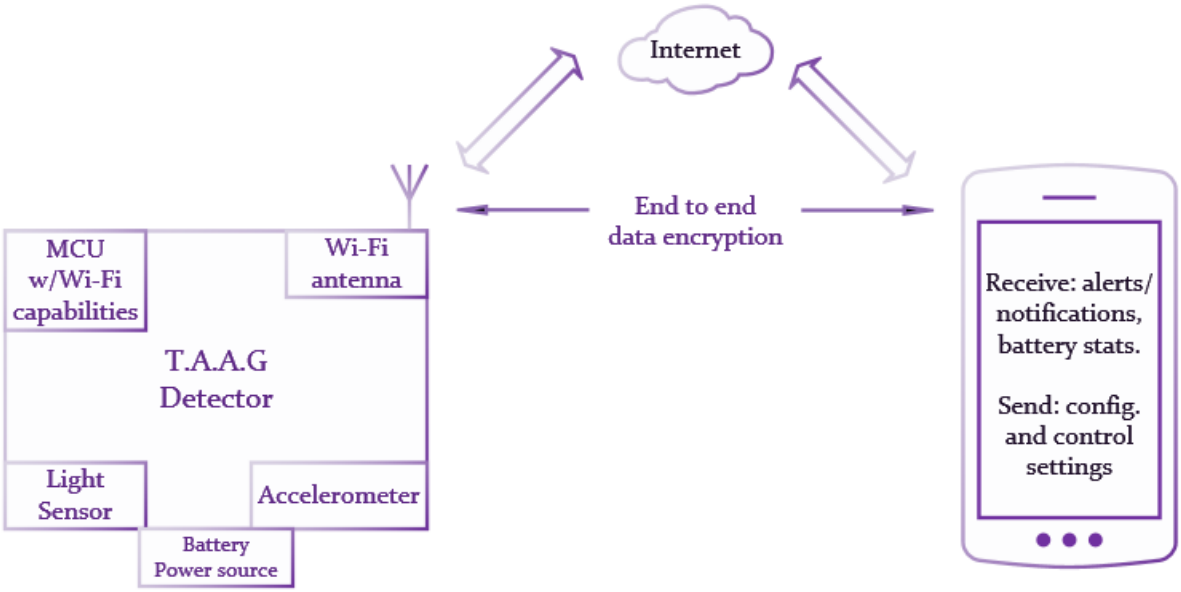

FIGURE 1: SYSTEM SUMMARY BLOCK DIAGRAM

5 Important Project Standards

This section details all of the standards that are relevant to our project.

5.1 Safety Standards

The table below give a list of the most important safety standards that must be taken into account in this senior design project.

TABLE 1: SAFETY STANDARDS

Standard Number	Title
IEC 61508	The IEC requirements for the functional safety of Electrical/Electronic/Programmable Electronic Safety Related Systems
BS EN 45510-2-3:2000	Guide for the procurement of power station equipment. Electrical equipment. Stationary batteries and chargers
BS EN 50272-2:2001	Safety requirements for secondary batteries and battery installations. Stationary batteries
BS EN60950-1: 2002	Low Voltage Directive (Safety)

Standard Number	Title
IEC/TR2 61430:1997	Test methods for determining the performance of devices designed for reducing explosion hazards - Lead-acid batteries
IEC62133:2002	Secondary batteries containing alkaline or other non-acid electrolytes - Safety requirements for portable sealed secondary cells, and for batteries made of them, for use in portable applications
IEC/TR2 61438:1996	Possible safety and health hazards in the use of alkaline secondary cells and batteries - Guide to equipment manufacturers and users
ANSI C18.2M	Safety Requirements for Portable Rechargeable Cells and Batteries
UL 2054	Safety Requirements for Household and Commercial Batteries
EN 45011	General requirements for bodies operating product certification schemes
EAS	The UK Electrotechnical Assessment Scheme (Electrical installation safety standards managed by the IEE)
BS 2754	Memorandum. Construction of electrical equipment for protection against electric shock
STRD Low Voltage Directive	The Electrical Equipment (Safety) Regulations 1994 SI 1994 No. 3260 Implementing Directive 73/23/EEC (The Low Voltage Directive - LVD)
IEC 479-1	Effects of current on human beings and livestock
IEEE 80 2000	IEEE Guide for Safety in AC Substation Grounding
HSG 85	HSE publication. Electricity at Work. Safe Working Practices
NFPA 70E-1995	Standard for Electrical Safety Requirements for Employee Workplaces
OSHA - 29 CFR 1910, Subpart S, Electrical	Electrical industry safe occupational working standards

5.2 Battery Monitoring Standards

We decided that the battery monitoring standards were an important part to add to the project because it is a key feature in the project battery management. These standards would help build the best and most efficient system so that we would not be wasting current and voltages, so that the battery can perform at its max capacity with harm to the system.

Meaning that if could maximize the performance of the system and less batteries would be needed. Table 2 shows these standards.

TABLE 2: BATTERY MONITORING STANDARDS

Standard Number	Title
IEC/TR 61431 1995	Guide for the use of monitor systems for lead-acid traction batteries
IEC/TR 62060 2001	Secondary cells and batteries - Monitoring of lead acid stationary batteries - User guide

5.2.1 Battery Recycling and Disposal Standards

The battery recycling and disposal standards are very important to use because as the name suggests, we wanted our product to have as little impact to the environment as possible. The only way to do so was to add as many environmental friendly standards to protect the environment and have the customer feel good about having a product like ours. This can be noted in table 3.

TABLE 3: BATTERY RECYCLING AND DISPOSAL STANDARDS

Standard Number	Title
EEC Directive 91/157	Batteries and accumulators containing certain dangerous substances. (Currently being revised)
BS EN 61429:1997, IEC 61429:1995	Marking of secondary cells and batteries with the international recycling symbol ISO 7000-1135

5.3 Other Related Electrical Standards

Table 4 gives some added electrical standards that are relevant to our project.

TABLE 4: OTHER RELATED STANDARDS

Standard Number	Title
BS 7671:2001	The IEE Wiring Regulations (UK)
NFPA 70 1993	National Electrical Code (USA)
UL 1310	Safety of Class 2 Power Supplies, AC Adapters and Battery Chargers – Testing
BS EN 60598-2-22:1999	Luminaires. Particular requirements. Luminaires for emergency lighting

5.4 Quality Standards

Table 5 give some of the quality standards that affected our project.

TABLE 5: QUALITY STANDARDS

Standard Number	Title
ISO 9000:2000	Quality management systems. Fundamentals and vocabulary
ISO 14001:1996	Environmental management systems. Specification with guidance for use
ISO 2859-0:1995	Sampling procedures for inspection by attributes
ANSI/ASQC Z1.4	Sampling Procedures and Tables for Inspection by Attributes

5.5 General Battery Standards

We decide to go with a few basic standards for each of the general types of batteries that we are considering for our project. Each of these standards were chosen for a reason, the most important being that without them we wouldn't be able to provide a safe and reliable product to a customer wanting to use our product. These standards (table 6) helps us in ensuring that our product will be perceived as reliable and help with consumer confidence.

TABLE 6: GENERAL BATTERY STANDARDS

Standard Number	Title
IEC 60050	International electro technical vocabulary. Chapter 486: Secondary cells and batteries.
IEC60086-1,BS 387	Primary Batteries – General
IEC 60086-2, BS	Batteries – General
ANSI C18.1M	Portable Primary Cells and Batteries with Aqueous Electrolyte - General and Specifications
ANSI C18.2M	Portable Rechargeable Cells and Batteries - General and Specifications
ANSI C18.3M	Portable Lithium Primary Cells and Batteries - General and Specifications
UL 2054	Safety of Commercial and Household Battery Packs - Testing
IEEE 1625	Standard for Rechargeable Batteries for Mobile Computers
USNEC Article 480	Storage Batteries

Standard Number	Title
ISO 9000	A series of quality management systems standards created by the ISO. They are not specific to products or services, but apply to the processes that create them.
ISO 9001: 2000	Model for quality assurance in design, development, production, installation and servicing.
ISO 14000	A series of environmental management systems standards created by the ISO.
ISO/IEC/EN 17025	General Requirements for the Competence of Calibration and Testing Laboratories

5.5.1 Lithium Battery Standards

TABLE 7: LITHIUM BATTERY STANDARDS

Standard Number	Title
BS EN 60086-4:2000, IEC 60086-4:2000	Primary batteries. Safety standard for lithium batteries
BS EN 61960-1:2001, IEC 61960-1:2000	Secondary lithium cells and batteries for portable applications. Secondary lithium cells
BS EN 61960-2:2002, IEC 61960-2:2001	Secondary lithium cells and batteries for portable applications. Secondary lithium batteries
02/208497 DC	IEC 61960. Ed.1. Secondary cells and batteries containing alkaline or other non-acid electrolytes. Secondary lithium cells and batteries for portable applications
02/209100 DC	IEC 62281. Ed.1. Safety of primary and secondary lithium cells and batteries during transport
BS EN 60086-4:1996, IEC 60086-4:1996	Primary batteries. Safety standard for lithium batteries
UL 1642	Safety of Lithium-Ion Batteries – Testing
ST/SG/AC.10/27/ Add.2	United Nations recommendations on the transport of dangerous goods

5.5.2 Nickel Metal Hydride Battery Standard

TABLE 8: NICKEL METAL HYDRIDE BATTERY STANDARDS

Standard Number	Title
BS EN 61436:1998, IEC 61436:1998	Secondary cells and batteries containing alkaline or other non-acid electrolytes. Sealed nickel-metal hydride rechargeable single cells
BS EN 61808:2001, IEC 61808:1999	Secondary cells and batteries containing alkaline or other non-acid electrolytes. Sealed nickel-metal hydride button rechargeable single cells
BS EN 61951-2:2001, IEC 61951-2:2001	Secondary cells and batteries containing alkaline or other non-acid electrolytes. Portable sealed rechargeable single cells. Nickel-metal hydride
BS EN 61951-2:2003	Secondary cells and batteries containing alkaline or other non-acid electrolytes. Portable sealed rechargeable single cells. Nickel-metal hydride
96/216533 DC	IEC 1808. Sealed nickel-metal hydride button rechargeable single cells (IEC Document 21A/207/CD)
97/204158 DC	IEC 1441. Secondary cells and batteries containing alkaline or other non-acid electrolytes. User-replaceable batteries containing more than one sealed nickel-metal hydride rechargeable cell for consumer electronic applications (21A/212/CD)
00/246138 DC	BS EN 61436 Ed 2. Sealed nickel-metal hydride rechargeable single cells (IEC Document 21A/303/CD)
GB/T18288-2000	Chinese National Standard for Nickel Metal Hydride batteries for mobile phones

6 Design Constraints in Completing the Project

Design constraints can come from many sources. Of the sources of these constraints are:

- Economic and Time constraints.
- Environmental, Social, and Political constrains.
- Ethical, Health, and Safety constraints.
- Manufacturability and Sustainability constraints.

This section will discuss each of these constraints in detail and how they impacted the overall design of the system at hand.

6.1 Economic and Time constraints

Time and money can alone be described as the two most impactful constraints that any team that is working on a project can face. Time being the most invaluable asset in the world, for it can never be returned once spent and is continuously being spent without the control of any person. And money being known for its scarcity in the world, and the need for it to be correctly allocated and not spent in ways that would lead to the lack of it.

The most interesting thing about the description that was given above is the fact that the two were described in a fashion that is almost completely interchangeable, this is why they can be grouped together as the two most impactful constraints in any project or system design that is made by any team. Below is how these constraints affected our project:

6.1.1 Time

Our senior design group has a very short period of time to complete this project. Our group has a period of two semesters, equivalent to a period of 8 moths to compete the project and make sure that it is fully functional.

As result of this constraint our group has an obligation to correctly utilize the allocated time, and to not waste much of it. This can be accomplished by organizational means, examples of measures taken to correctly use the time allocated is the setting of mile stones, weekly and/or biweekly meetings, and the use of checks and balance, where each member of the group is obliged to show the progress that they have made towards completing the project each time the group meets.

For our project, time represents the greatest of all constraints that the group has to face, wisdom and precaution must be taken.

6.1.2 Money

Money here is the second most important constraint that the team must deal with. As mentioned earlier in this document, our project is being funded by the MIST Center research lab at UCF, our senior design group has been allocated and nominal budget of around $700. In knowing this budget our team must be very careful and aware when purchasing any parts, equipment, and development kits. Exceeding this budget would could potentially lead to the need of the members of the group to start purchasing parts and services from their own personal funds, which a final resort option that our group would like to avoid.

With the overall direction the project is going in it does not seem as though the group will fall into any budgetary issues. The implementation method of the project was carefully chosen to not exceed the allocated budget in any way shape or form.

Another parameter that must not be neglected is the accounting of the project. The group must carefully keep track of all of the expenses to make sure that the budget has not been exceeded. Methods to keep track of this information is the storage of all the receipts and bills of sale, and to also keep another more organized list or spreadsheet that holds the

information of all the purchases made. This spread sheet can show the amount of budget spent and the amount that remains, this will keep a constant awareness of the economic situation of the group.

6.2 Environmental, Social, and Political constrains

As far as this category of constraints are concerned this project is not significantly impacted by any environmental, social, nor political constraints.

The structure and application of our project may have some environmental constraints as far as some of the components are concerned. Here is a list environmental constraints that were imposed:-

- Avoid the use of certain battery pack types that may have some harmful side effects when placed with other components in the system.
- Taking into consideration the temperature of operation of the system. The system that we have developed so far has a temperature of operation of room temperature with variations to accommodate temperature variations that are within a reasonable limit.
- Also when it comes to emissions and electromagnetic radiations produced, our group is committing to the standard (Wi-Fi certified standard). The Wi-Fi module that is within the MCU is the only source of emissions and electromagnetic radiations, other than that all the components communicate in a wired method that produces almost no effect to the external environment.
- There are also some other external environmental factors that may affect the operation of the system. Of these factors are:-
    - Magnetic fields: An introduction to an external magnetic field may cause the system to malfunction this constraint may lead to some modification to the overall system design.
    - Current surge: There also exists a possibility that when the device's battery is being charged that there must be some variation in the current that is being fed into the system. For that reason we needed to add some form of regulation to the system's powder supply.
    - Electro-Static Distortion: ESD can prove to be a very dangerous environmental occurrence which can happen when people or machinery touch the electronic components in the system and cause some distortion or permanent damage. This constraint obliges the group to design and incorporate some ESD protection circuitry.

Although our project has many different environmental constraints, we should consider our battery as having the most noticeable impact. Our dream is to have multiple small devices connected together and functioning on a single network. If hundreds of units are bought and are in use in offices or homes then that means all this batteries must be monitored. Without a doubt, nothing lasts forever and our battery will have to be changed out some day, depending on the technology we decide to go with whether it be rechargeable or non-rechargeable batteries.

When the monitoring system tells you the life of the battery has been reached its limit and to swap out the battery for new ones, then the old ones must be properly disposed of because of a potential impact to the environment. More specifically, the NiMH (nickel metal hydride) and Lithium ion batteries are slightly toxic and ought to be disposed of properly. We recommend recycling these batteries to minimize the environmental impact and decrease its footprint in the environment. More will be discussed about the different battery types and their implications later in this document.

The only social constraints that we may have is the acceptability of the product from a marketing perspective. The question of how this will appeal to a potential customer if this project were to be translated into a commercial product introduced some constraints as far as the user interface and the external appearance of the device is concerned. As such our group has decided to come up with a user experience that is intuitive and easy to use. The final external shell of the device will have an appealing look that is marketable and robust.

This senior design group does not believe that there are any significant political constraints that have actually affected the design or the implementation of the project.

6.3 Ethical, Health, and Safety constraints

This category of constraints are quite important. There are quite a few of this type of constraints that impacted the design and implantation of our project. Here are a few points on the matter.

An ethical constraint is that of surveillance. This project can be categorized under security and surveillance. Although when most common people hear surveillance they immediately picture the usage of camera and video equipment to record that actions that people are doing in order to protect public property or any object or place that is of value. Surveillance that employs cameras also acts as an excellent form of deterrence to crime, because surveillance cameras' recordings can be used as a means of conviction in the court of law.

The point being that although the popular means of surveillance is the use of cameras, it can also exist in various other forms as well. Our project fall into the sensory alarm surveillance category, for it in fact does act as an alarm, alerting the user that placed it in/on their valuable belonging that their valuable has been tampered with. From this other people may not know that whatever common object that they are going to touch may be "T.A.A.Ged" and could alert the user that placed the alert device on the common object, hence it becomes a subjective matter to whether the "assailant" should actually get in trouble for something touching something that may seem common, but is actually being protected.

For the reasons mentioned above if this project were to be commercialized our group would advise the user to place the tag on objects that are strictly theirs and to avoid placing them on objects that are more common unless they were to issue a warning out like a note or a notice.

As far as health and safety constraints are concerned this project does not have any particular issues that have had any significant effect on the design and implementation. The

use of standards when designing many of the portions of the system took care of all the health and safety constraints.

6.4 Manufacturability and Sustainability constraints

This category of constraints have to do with the transfer of the on paper design into a physical working system that does what is written of it to do. This category also touches up on the constraints that must be in place in order to ensure that the product would have an acceptable product life and would be reliable enough to work correctly for that life period.

Here are some points as to how this category of constraints affected our overall design and implementation of the system:-

- Designing the PCB layout is the biggest example for manufacturability constraints.
    - In designing the PCB some techniques must be used in order to optimize the number of layers that were used, such that the manufacturer can have ability to implement and synthesize our design. This can also fall into the economic category, but at this point that has already been determined by the budget. Hence, this becomes a manufacturability constraint.
    - When designing the PCB routing we must also make sure that we do not exceed the design rules that are set by the manufacturer. This adds some constraints to the overall design.
    - Also, when designing the layout the group must also be weary of the placement of all of the components, such that they are not placed too close to each other and that the soldering will not cause any issues. This can also fall into the design rules constraints.
- The issue with sustainability is that it is harder to determine, because it requires some time consuming and resource consuming methods such as stress testing which puts the product thought simulated harsh environments that would model what would happen to the product over a long period of time. Due to the time constraints and economic constraints our group doesn't have the means to come up with or figure out what there constraints exactly are. As a result, to put these constraints into consideration the group has employed methods of looking at each of the components sustainability results from the manufacturer and taking their word for it. In summary, sustainability constraints are not fully in place for the system as a whole.

7 Communications System from a Hardware Perspective

This portion of the document discusses the communication systems and what it takes for the tamper detection device to be connected to the internet from the hardware point of view.

7.1 The Internet of Things as a Solution

When needing a system to communicate to the other systems and to users, often the method taken for this communication to take place is through the internet. The Internet of Things

(IOT) often used as a buzz word to attract the attention of users and to denote extraction of results through the internet. With the ever expanding need for computational power and the push for the scaling of the devices held by users, a sort of solution for the continuation of Moore's Law in a fashion that has been dubbed "More than Moore" IOT systems can deliver this demand of high computational output through the use of remote servers which house monstrous commuters with high performance, capabilities, and size that would perform all of the resource consuming computational processes and simply transmit the results to the user over the internet. From this the device that the user holds is simply a terminal to which all the heavy duty is being loaded elsewhere this would allow for the scaling of the devices.

Not just for computation, IOT systems are now employed for control and monitoring, even doing so for parts of our everyday lives. With this we can in essence make our environment "smarter" and that is the principle of the IOT. Our project is considered to be a great application of the IOT, since it uses the internet to monitor everyday regular objects, giving the user the ability to add more security and more convenience to their lives.

7.2 Initial Idea of Using a Base Station for Communication

When our group first came up with the design for the project, we initially had the design to be in a similar fashion to that of the Samsung SmartThings. This design involved a base station (which would be a commuter or microcomputer) that would act a sort of server or link between the user mobile device's companion app and the tamper detection device, a middle man basically. This middle man would increase the cost of the system and make setting up the system a bit more involved, hence impeding the user experience. Figure 2 is a more descriptive block diagram of what we had in mind.

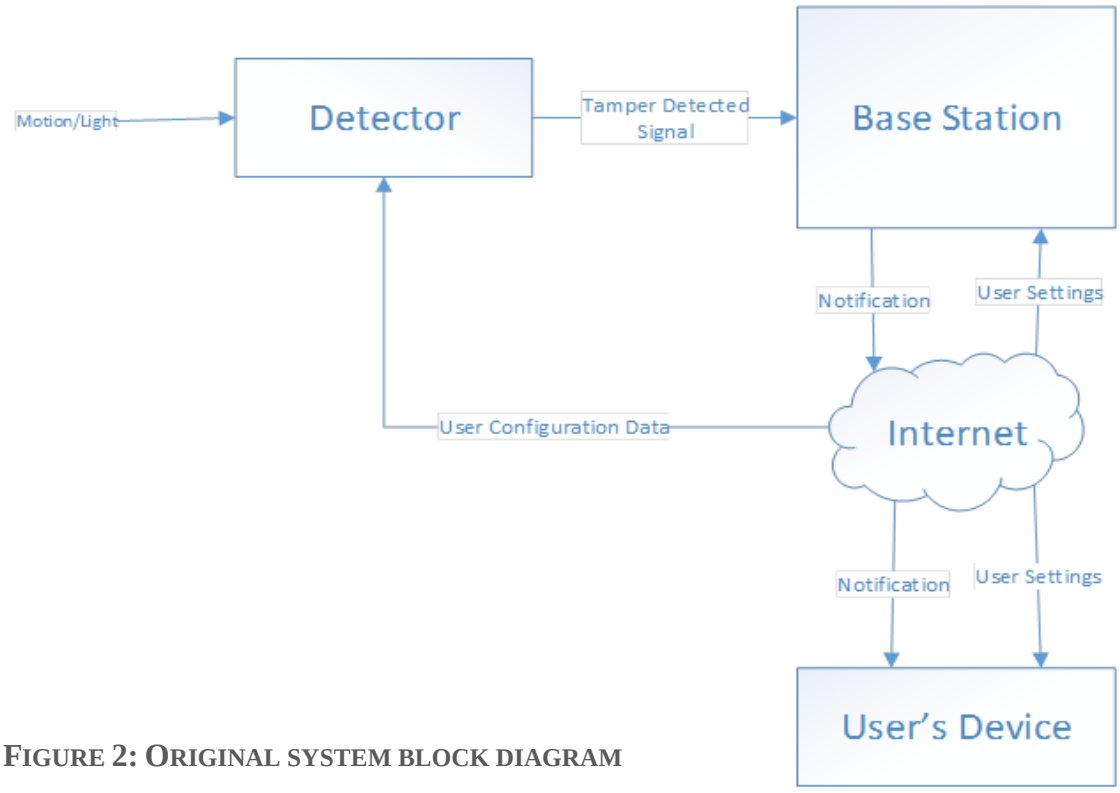

FIGURE 2: ORIGINAL SYSTEM BLOCK DIAGRAM

Although the design has changed from this, in retrospect we can tell that that there may have been some pros to this design. This design would have made the individual detectors more battery efficient, since their communication means would have been by a more power efficient means, and it may have made the software development of the feature of adding more detectors to the system more streamlined and straight forward. In the end we chose to go with a design that we believe will be more efficient, as in figure 1.

8 Light Sensing From a Hardware Perspective

The T.A.A.G system, as previously stated in this document, comes equipped with the ability to detect and sense light. This ability to sense light is what gives the detector the ability to detect a breach in the environment in which it has been placed in, for example a user can place it inside the back of a filing cabinet, and once one of the drawers on the light cabinet is opened the light will change hence the triggering the detector to push a notification to the user.

8.1 Research on Hardware Means of Sensing

From a physics perspective there can be many ways of sensing light. At its premise sensing light just means the use of a transducer to convert the light signal to an electrical signal which can then be later analyzed and converted into useful information in some way or the other. Below are some the initial techniques that our group has initially come up with in order to sense light and covert its intensity values to that which is meaningful:-

The use of Light Dependent Resistors (LDR): This technique involved the use of an LDR in series with a resistor in order to form a voltage divider network, as shown in figure 3.

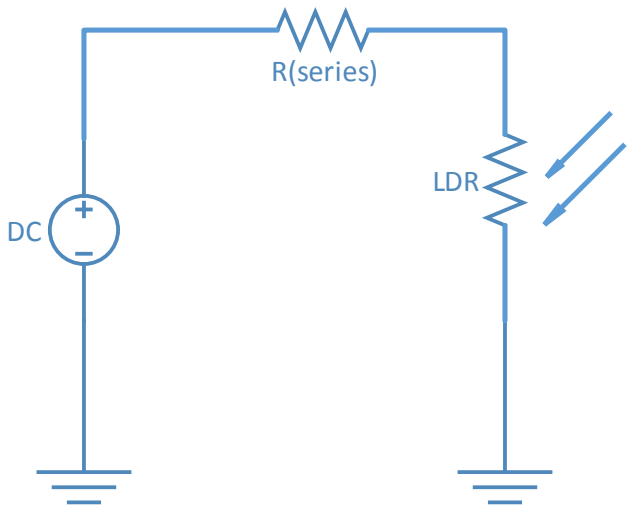

FIGURE 3: LDR VOLTAGE DIVIDER NETWORK

In the network above the output would be considered to be at LDR node, consequentially this output would be connected to the analog to digital converted to be converted to a digital value which can then be analyzed. With this, light can be sensed because the resistance

value of the LDR changes depending on the light that is incident upon it. When the LDR is exposed to different intensities of light its resistance value will change and, since it is a resistor in series with a voltage source and another resistor, the value of the voltage will change. This can be seen from the relation below.

$$V_{LDR} = \frac{LDR}{LDR + R(series)} * V_{DC}$$

From this relation we can tell that with each light intensity value we will get a different light value, hence this can be considered as a transducer.

There is a down side to using this technique especially with this topology. This circuit topology has passives that will perpetually consume current, this current consumed will lead to some power dissipation with can hurt the battery life of the system.

The use of Light Diode in a light sensing configuration:

This employs a similar circuit topology where the circuit contains a light diode in place of the LDR. The network can be seen below in figure 4.

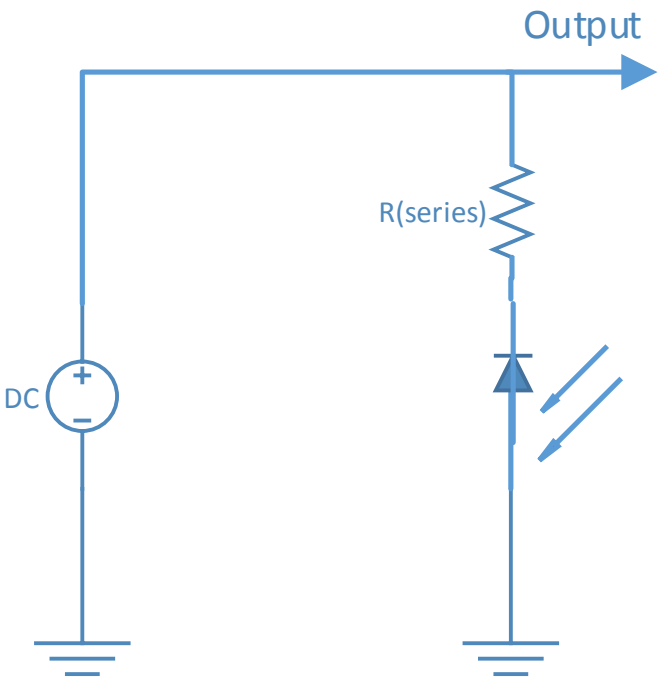

FIGURE 4: LIGHT DIODE NETWORK

In this configuration the light diode controls the amount of current flowing within the circuit. Looking at the circuit if it were to be an ordinary diode then there would be no current following the circuit would be open by virtue of the diode. Since here we have a light diode which is sensitive to the light that is incident upon it, the diode will turn on when it is hit by light, despite the fact that it is paced in a reverse bias configuration. With

this the series resistor will have current flowing through it creating a voltage drop at the output. The voltage drop can realized by:-

$$V_{output} = R * I_D$$

This voltage, as in the case with the LDR network, will be passed to the analog to digital converter to be converted to a value which can be interpreted and linked to a specific light intensity value. Effectively with this a light to electrical signal transducer can also be created.

This configuration is more power efficient than that of the LDR. The reason for this is the fact that the current that is going through the light diode is quite small when compared to that of the LDR. Since an argument can be made that the signal might be too small with the use of the light diode an operational amplifier can be placed at the output to amplify the signal and enable the ADC converter to pick up the signal.

After doing research of this type, our group decided to go with a light sensing solution that employed a light diode network for sensing light, because this would help in optimizing the system's overall power consumption.

8.2 Parts and Integrated Circuits Used to Employ Light Sensing

After doing the initial research, came the time to scavenger the market and look for the most optimal parts that we can use in order to obtain the most optimal results.

Our search led to the doors of many reference designs and manufactures. At first our group was keen on finding a solution using reference designs from Texas Instruments' Webench Tool. This tool would take all this parameters that the user was interested in and would look up parts and go through its data base to find an ideal circuit with which the parts can be placed. Below is the reference design circuit that Texas Instruments was able to generate for us (figure 5):

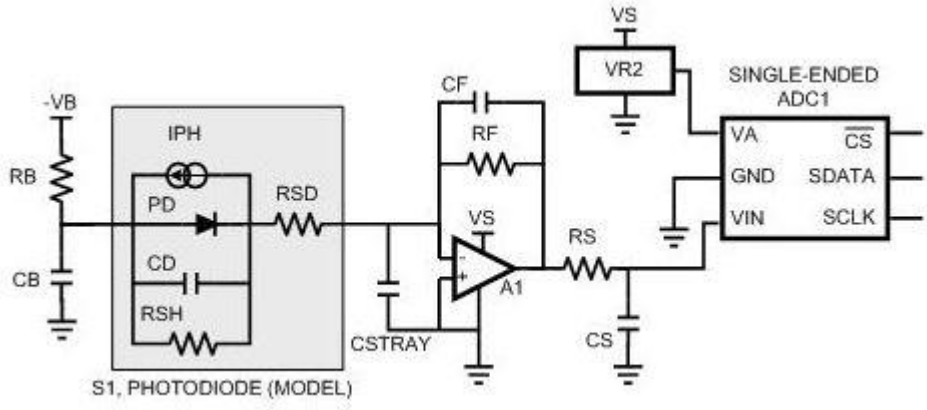

FIGURE 5: WEBENCH PHOTODIODE REFERENCE CIRCUIT

(REPRINTED WITH PERMISSION FROM TEXAS INSTRUMENTS)

As show in the circuit, the network employs some passive elements (resistors and capacitors) an operational amplifier, and a single ended analog to digital converter IC. This reference circuit uses all of these components to generate a voltage value for the different light intensities in a low power design means. This also provided a very viable solution especially due to the fact that it suggests the use of an ADC converter that has the ability to communicate with an MCU with the used of I2C communication.

The only general down side of the use of reference circuits is the room for error it creates, this room of error can only be brought up when reference designs are compared to integrated circuits that have the entire system or network on one chip. These ICs are especially reliable and easy to use, in addition they also sometimes come with added features that can be quite beneficial and quite resourceful, hence saving time, effort and money. From this we decided to go for a discrete chip for sensing light in the system.

8.2.1 TAOS TSL2561 LIGHT-TO-DIGITAL CONVERTER:-

As the name suggests the TSL2561 is an all in one light sensing circuit that employs a light diode network in order to sense light. It is easy to use and employs the usage of internal registers in order to configure different parts of it. The IC contains within it the light diode with the passive and active (namely MOSFETs for amplification) elements surrounding it, a small scale voltage regulator, and an analog to digital converter.

The IC has a total of 6 usable pins [1] as seen in the table 9 below:

TABLE 9: TSL2561 PIN USAGE

Name	Pin Number	Type	Description
ADDR SEL	3	I	SMBus device select – three state
GND	2		Power supply ground. All voltages are referenced to GND.
INT	6	O	Level or SMB Alert interrupt — open drain
SCL	4	I	SMBus serial clock input terminal — clock signal for SMBus serial data.
SDA	5	I/O	SMBus serial data I/O terminal — serial data I/O for SMBus.
VDD	1		Supply voltage.

From the table of pin distribution above it can be noted that as far as the hardware is concerned all that it takes to make this IC an operational part of the system is correctly connecting the pins into their correct location of input or output, and the rest of the processes in programing correctly in order to sense the light.

Below is a list summarizing the benefit of adding such a part to our system:-

- Ability to communicate in I2C helps for better system integration, because this makes it similar to the other peripherals in the system when it comes to communication, meaning that it can share the bus with them.
- The ability to generate interrupts on the user's configuration helps the battery consumption of the system by a great amount. The reason for this is that the MCU (the greatest source battery consumption in the system) can be placed in low power mode and be woken up by the interrupt rather that the use of constant polling.
- The inclusion of its own ADC converter saves the use of the MCU's internal ADC converter and saves it to be used by other components in the system.
- As stated earlier the use of the light diode helps with the overall battery life of the system.
- The inclusion of the registers in the IC assists in the configurability of the IC, with that settings and the interrupt threshold can be constantly adjusted.

Below is a schematic of how the IC will be connected in the system (figure 6):-

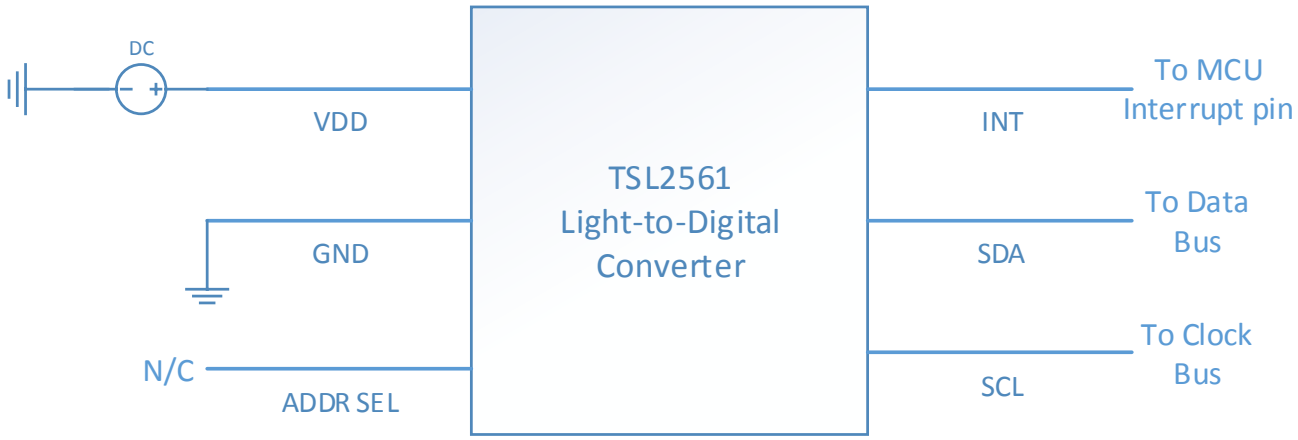

FIGURE 6: LIGHT SENSOR IC CONNECTION

9 Motion Sensing From a Hardware Perspective

The detection system comes with the ability to sense whether the object it is placed on has been moved. It does so through the use of motion sensors. As with the light sensor this is considered a transducer, converting one source of energy to electrical energy, in this instance motion (mechanical energy) to an electrical signal. That is one way to think of another way is to use a motion sensitive switch.

9.1 Research on Hardware Means of Sensing

In its simplest form light detection can be accomplished with the use of a spring that has the flexibility to move and touch another wire in order to close the circuit. This is what is called a vibration sensor which consists of a spring with a wire in the middle. The vibration sensor has two leads, the end of the metal spring is connected to one lead and the central wire is connected to the other lead. With this if the sensor were to be connected in a circuit then the circuit would close every time the vibration sensor would vibrate, because the

vibration would cause the coil to touch the wire, hence the connection would be made. The circuit shown below in figure 7 is a great illustration of how the vibration sensor can be connected in the circuit.

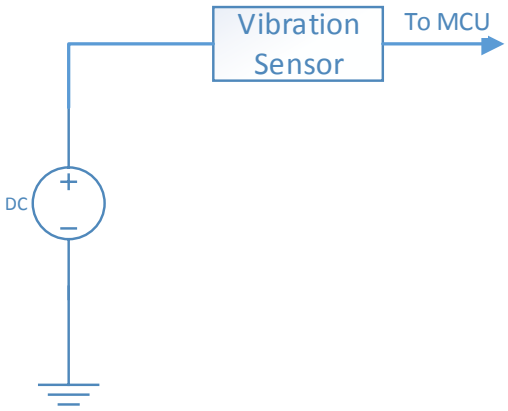

FIGURE 7: VIBRATION SENSOR CONNECTION TO MCU

With this the output can be connected to an interrupt pin on the MCU therefore generating an interrupt upon the vibration of the vibration switch.

The down side of using the vibration switch is the lack of sensitivity settings. Each discrete vibration sensor can either be very sensitive or be very stiff and need a hard knock in order to be triggered, a solution to this would be to use different vibration switched with varying sensitivity and connect them to different pins on the MCU, hence each interrupt pin would be each specific sensitivity setting. This still does not provide a viable solution, because the MCU has but a limited number of interrupt pins and we would run out of pins before we could incorporate any good number of sensitivity settings. As such a different solution would be needed.

In a system such as the one we have you need a Micro Electrical Mechanical system (MEMS) that would employ a better technique for sensing. A MEMS would provide the flexibility that we need with regards to the number of sensitivity settings. The reason for this is because they mainly come in ICs that are programmable and includes modes of operation that are quite useful to use.

A good example of a MEMS that would be quite useful for sensing is an accelerometer. With nowadays' technology accelerometers can sense motion in all 3 planer axes, some the latest and most interesting accelerometers operate thermally. These thermally operated accelerometers have a wire right in the middle of two heat sensors. When at a standstill the temperature sensors read the same temperature, because they are at an equal distance from the heated wire. As the accelerometer movers the heated wire moves as well this causes it to sway in the direction of either of the two sensors, causing a difference in the temperature read. When this is applied in all three planar axes then with a function relating the difference in temperature with the actual acceleration, motion can be sensed.

For the purposes mentioned above, we decided to go with an accelerometer IC for implementing the system's motion sensing application. The accelerometer we decided to use was the Bosch BMA222 3-AXIS ACCELEROMETER DIGITAL SMD.

9.1.1 Bosch BMA222 3-AXIS ACCELEROMETER DIGITAL SMD.

This is accelerometer is a 12-pin IC [2] (all the pin usage can be seen in table 10) which is able to detect acceleration in all three planar axes. It is loaded with registers which are used for programing the different configuration settings. The accelerometer also has the ability to generate interrupts, which can be very useful in maintaining the system's low power design by allowing the MCU (the greatest source of power consumption in the system) to go into sleep power mode.

TABLE 10: PIN USAGE OF BMA222 ACCELEROMETER

Pin Number	Name	Type	Description
1	SDO	O	Serial data output in SPI Address select in I2C mode
2	SDx	I/O	SDA serial data I/O in I2C SDI serial data input in SPI 4W SDA serial data I/O in SPI 3W
3	VDDIO	Supply	Digital I/O supply voltage (1.2V … 3.6V)
4	NC	--	
5	INT1	O	Interrupt output 1
6	INT2	O	Interrupt output 2
7	VDD	Supply	Power supply for analog and Digital domain (1.62V … 3.6V)
8	GNDIO	Ground	Ground for I/O
9	GND	Ground	Ground for digital and analog
10	CSB	I	Chip select for SPI mode
11	PS	I	Protocol select (GND = SPI, VDDIO = I2C, float = uC-less). Pin must not float unless dedicated mode is used
12	SCx	I	SCK for SPI serial clock SCL for I2C serial clock

This accelerometer works using variable capacitors. The capacitance is varied with the aid of springs that change the distance between the plates of the capacitors. When these springs

experience acceleration they immediately contract or expand, at the occurrence of this the values of the capacitance will change based on this formula:-

$$\Delta C = C_2 - C_1 = \epsilon_0 * A * \left(\frac{1}{d_2} - \frac{1}{d_1}\right)$$

Where:-

ΔC *is the change in capacitance*
C_2 *is the final capacitance*
C_1*is the initial capacitance*
ϵ_0 *in the permitivity of free space*
A *is the surace area of the capacitor plate*
d_2 *is the final distance between the capacitor plates*
d_2 *is the final distance between the capacitor plates*

The change in capacitance will also cause a change in the charge stored in the capacitor. This change in charge will lead to the change in the voltage across the capacitor, and will represent the signal analyzed and covered in order to calculate the acceleration that the IC or system is experiencing. And with the programing of the thresholds of acceleration for an interrupt, interrupts can be generated upon the value of the acceleration exceeding the defined threshold. That describes the advantage of using an accelerometer, when taking all of its features into consideration and comparing it to the vibration switch, there comes to mind no reason what so ever to why not to take the accelerometer over something like the vibration switch.

Below is a list of the advantages of using an accelerometer:-

- Ability to communicate in I2C helps for better system integration, because this makes it similar to the other peripherals in the system when it comes to communication, meaning that it can share the bus with them. Saving hardware cost and design when compared to proprietary communication methods.
- The ability to generate interrupts on the user's configuration helps the battery consumption of the system by a great amount. The reason for this is that the MCU (the greatest source battery consumption in the system) can be placed in low power mode and be woken up by the interrupt rather that the use of constant polling methods.
- The Micro Electronic Mechanical System that is present within the accelerometer IC provides a clever and powerful way to measure and sense motion and acceleration of the system.
- The inclusion of the registers in the IC assists in the configurability of the IC, with that settings and the interrupt threshold can be constantly adjusted.

10 Microcontroller from a Hardware Perspective

This section of the document will discuss how the MCU as a hardware component will fit into the system, and how it will be networked to make the system functional and

operational. It is worth pointing out the MCU is the most pivotal and central component of the entire system, it represents the central hub that takes all of the information and processes it. From a system perspective it can be considered as a giant black box that can take in many inputs and can generate many outputs as well. Within that black box is a system within itself, this microsystem consist mainly of silicon and layers of metal. All of these connections and subsystems within the MCU are all managed by the programmer, the code used configures and creates all the necessary connections for the MCU to operate as described by the programmer.

10.1 Initial Research Finding an Optimal MCU

There are is a countless number MCUs that are manufactured by a wide variety of manufactures. Each MCU having its own parameters (strengths and weaknesses), with a clear idea of our need in mind we can easily form a decision chart with all the MCU listed with how they can be ranked and how we grade them, but this was not necessary for our project. With how our project is meant to be applied and accomplished (IOT application, with Wi-Fi connectivity) our search led us one possibility to make this happen most optimally, Texas Instruments CC3200 SimpleLink™ Wi-Fi® and Internet-of-Things solution, a Single-Chip Wireless MCU. Just from a simple reading of the MCU's features we were able to tell that it will be the best option.

Our group didn't stop there we tried to look for other manufactures that could possibly make an MCU that competes with this on, with hopes to find a better price point or better features. The group was not able to find any other MCU with the shear ability and the shear amount of features this MCU can handle.

10.2 CC3200 SimpleLink™ Wi-Fi® and Internet-of-Things solution, a Single-Chip Wireless MCU

Below is a portion of the description that we read from the Texas Instruments web site that talks about the CC3200 MCU:-

> *Start your design with the industry's first Wi-Fi CERTIFIED single-chip microcontroller unit (MCU) with built-in Wi-Fi connectivity. Created for the Internet of Things (IoT), the SimpleLink CC3200 device is a wireless MCU that integrates a high-performance ARM Cortex-M4 MCU, allowing customers to develop an entire application with a single IC. With on-chip Wi-Fi, Internet, and robust security protocols, no prior Wi-Fi experience is required for faster development. The CC3200 device is a complete platform solution including software, sample applications, tools, user and programming guides, reference designs, and the TI E2E™ support community. The device is available in a QFN package that is easy to layout.*

The description above detailed and went beyond our requirements for an MCU. With the inclusion of an embedded Wi-Fi module within the MCU we were able to avoid the cost and effort of obtaining an external module to connect to the internet.

What makes the CC3200 MCU very impressive is the fact that it pushes the ability of a microcontroller to the absolute limit. A more accurate description for the CC3200 would be a system on a chip, for it within itself is a system that is acting on its own accord. It alone hosts the following within the chip:-

- ARM Cortex M4 Core 80Mhz Processor
- Dedicated Wi-Fi Network Processor
    - Featuring Wi-Fi Internet-On-a-Chip™
    - Dedicated ARM MCU Completely Offloads Wi-Fi and Internet Protocols from the Application Microcontroller
    - Wi-Fi and Internet Protocols in ROM
    - SimpleLink Connection Manager for Autonomous and Fast Wi-Fi Connections
-
- Embedded Memory
    - RAM (Up to 256KB)
    - External Serial Flash Bootloader, and
    - Peripheral Drivers in ROM
- 8-Bit Parallel Camera Interface
- Dedicated External SPI Interface for Serial Flash

This and many more other features that make this MCU truly a system on a chip.

Despite how useful this MCU is to our system, there was but only one down side to the story, which was the lack of extensive support from the development community. We were able to find some software support, but the hardware support was quite scarce indeed. For this we took up some extra research and we found out that Texas Instruments also provided what is called SimpleLink Wi-Fi CC3200 Internet-on-a-chip Wireless MCU module. The key word of difference here being the word MODULE, the MCU being more known as CC3200MOD.

10.3 CC3200MOD over the CC3200

An interesting fact to note with the original CC3200 MCU was the fact that it is not FCC (Federal Communications Commissions), CE (Conformité Européenne, CE marking), nor IC (Industry Canada) approved for emissions when without all of the passives and antenna circuitry that are referenced in the LaunchPad reference design. This is understandable since Texas Instruments are the first manufacturer to come up with an MCU that works in such a way. The implication of this is that when not used that as in the reference design it would limit legally releasing and selling products with this MCU in United States, Europe, Australia, New Zealand and do not have high volumes [3]. Although the main reason we shifted from the regular CC3200 was because of the lack of extensive hardware support, this factor acted also as a major catalyst to the transition.

The CC3200MOD was Texas Instruments' solution for this predicament, even though it come with a higher price tag due to FCC, CE, and IC certifications, and Wi-Fi Certified modules with ability to request transfer certificate with Wi-Fi Alliance members [3], it still serves as a viable solution. It solves the problem by integrating all of the passives and the

antenna circuity into one discreetly sold module, this packaging causes the MCU to have a different or shifted pin location and overall different hardware arrangement. As such the Texas Instruments E2E took more of a liking to the CC3200MOD over the original CC3200, developing more reference designs and hardware support. One item to notice is that as far as the programmability is concerned there is not difference what sever when it comes to the code used, the code used to develop the CC3200 can be directly imposed on the CC3200MOD. This explains why there was more software support than hardware support for the original CC3200 MCU.

10.4 Hardware for Development Environment

To increase productivity and to not lag the software development (not having to wait for hardware to develop) it is necessary for our group to purchase ready-made development kits. These development kits contain the MCU soldered into a printed circuit board in a development environment. Such development environments contain everything as far as the hardware is concerned, hence giving the software development less hindrance.

In the early proof of concept, development, and testing phases of the project we decided to use Texas Instruments CC3200 Module LaunchPad. TI LaunchPads are the product line that place some of the most popular Texas Instruments microcontrollers in a development kit fitted with many sensors, buttons, LEDs, and other peripherals. These LaunchPads also come with all the necessary passives and oscillators making ready-to-go upon taking it out of the box.

An important factor is that the CC3200 LaunchPad comes coupled with on-board emulation, which means we can program and debug the microcontroller for our project without the need for additional tools. Also from a hardware point of view it comes packaged in a LGA (land grid array) package for easy assembly and low cost printed circuit board design. This hardware packaging will help enhance the hardware development and make it copasetic from all angles. The reason for this is that the CC3200 LauchPad’s hardware design is quite optimal and provides a very exemplary way to accomplish the hardware system, hence it would be most beneficial to use the LaunchPad’s design and to modify it to the liking of our system (more of this will be discussed later in the document).

Below is an example of one portion of the LaunchPad's reference design which can be viewed and modified on and PCB and schematic editing tool, figure 8:-

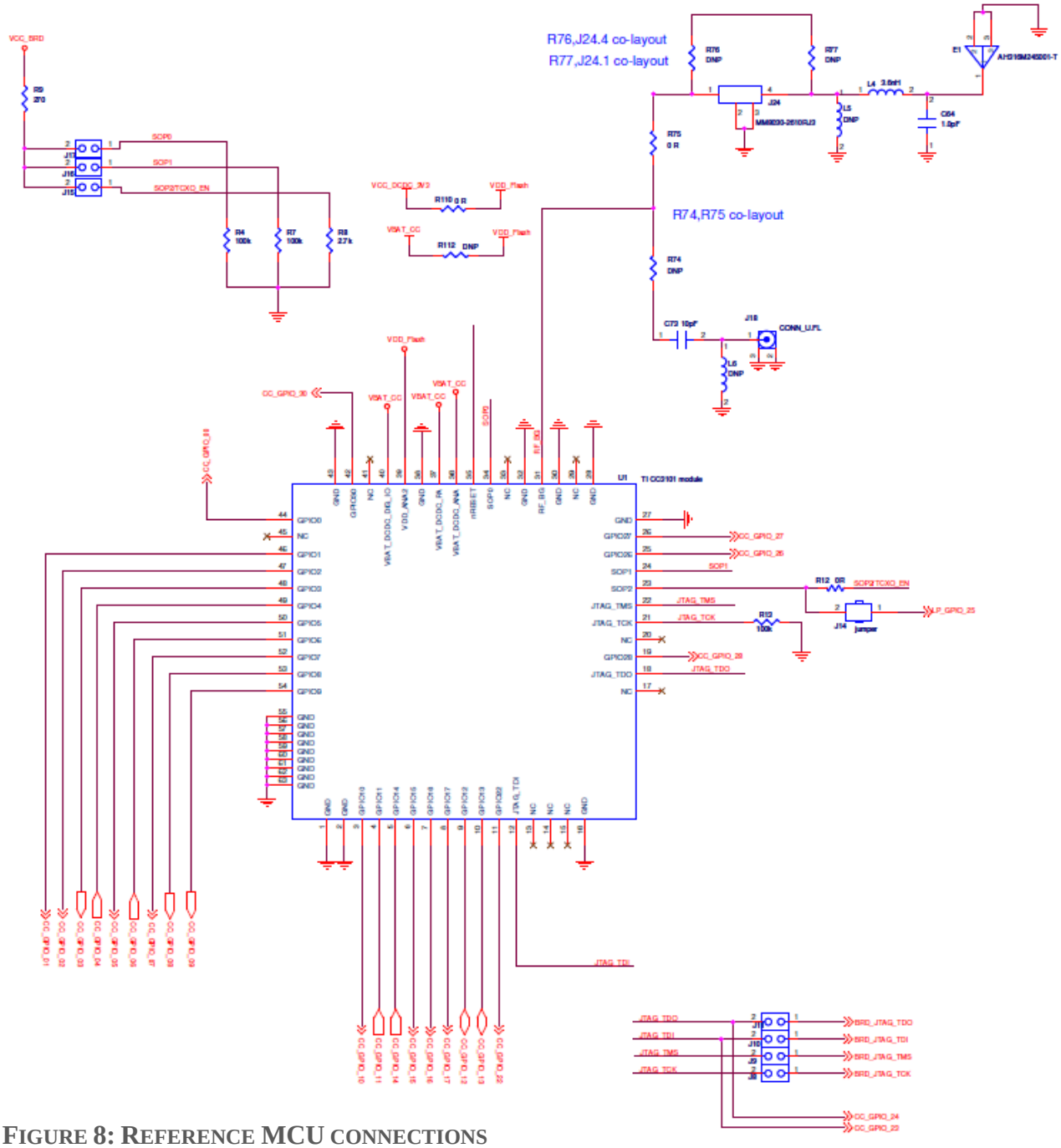

FIGURE 8: REFERENCE MCU CONNECTIONS

(REPRINTED WITH PERMISSION FROM TEXAS INSTRUMENTS)

The LanchPad schematic is quite dense, figure above only represents 1 out of the 4 other schematics. The schematic above represents the most important and central one the rest of

the schematics only show how the sensors, the flashing and debugging, antenna, etc. circuits are connected, but the schematic with the MCU represents the most vital and critical one.

By editing the hardware schematic files such as the one above, we can ensure that the final prototype will have a minimal amount of flaws possible.

Another point worthy of note is the re-usage of the development LaunchPad even in the final prototype. The final prototype which will be discussed in more detail later on in this document will consist only and primarily of part and circuity to make them work as a system with any extra unneeded components or subsystems. With that being said, a programing circuit will not be included in the final prototype, since it is not needed for the perpetuation of the system's operation, it is only needed at one point in the systems' life, not mention that fact that including it increases the device's unit cost and make the device's size much bigger than it needs to be. Hence to rectify that, in the final prototype we shall include simple headers that connect to the MCU's programing pins.

Simple metal headers do not cost much at all nor do they occupy much space on the printed circuit board, which made them the go to solution for programming the final prototype. Most importantly the above mentioned statements place emphasis on the fact that we still will make use of the LaunchPad even after the full hardware development phase is complete.

10.5 MCU Pin Connections

There still remains a discussion to be made about the MCU's pins and where each and every single one of them is going to be connected to. It is quite vital to understand and know where each pin on the MCU must connect to this. As mentioned earlier in this document, a proper understanding of pin termination locations (VCC, Ground, or Non-connect) must be in place in order to avoid running into power dissipation issues and other unexpected issues. For this the table below includes all of the pin of the CC3200MOD microcontroller and all the pin termination locations that we used in the design of the final prototype, table 11:

TABLE 11: MCU PIN CONNECTIONS USED IN THE PROJECT

Module Pin No.	Module Pin Name	Type	Device Pin No.	Module Pin Description	Use (in final prototype)
1	GND	-		Ground	
2	GND	-		Ground	
3	GPIO10	I/O	1	GPIO	SCL of I2C bus
4	GPIO11	I/O	2	GPIO	SDA of I2C bus

Module Pin No.	Module Pin Name	Type	Device Pin No.	Module Pin Description	Use (in final prototype)
8	GPIO17	I/O	8	GPIO	Battery Gas Gauge Interrupt pin
10	GPIO13	I/O	4	GPIO	Motion Sensing Interrupt Pin
12	JTAG_TDI	I/O	16	GPIO	Mapped to JTAG header
16	GND	-		Ground	
18	JTAG_TDO	I/O	17	GPIO	Mapped to JTAG header
21	JTAG_TCK	I/O	19	JTAG TCK input. Needs 100-kΩ pulldown resistor to ground.	Mapped to JTAG header
22	JTAG_TMS	I/O	20	JTAG TMS input. Leave unconnected if not used on product.	Mapped to JTAG header
23	SOP2	-	21	Add 2.7-kΩ pulldown resistor to ground needed for functional mode. Add option to pullup required for UART.	Mapped to SOP header

Module Pin No.	Module Pin Name	Type	Device Pin No.	Module Pin Description	Use (in final prototype)
24	SOP1	-	34	Reserved. Do not connect.	Mapped to SOP header
25	ANTSEL1	I/O	29	Antenna selection control	Mapped to RFID chip
26	ANTSEL2	I/O	30	Antenna selection control	Mapped to RFID chip
27	GND	-		Ground	
28	GND	-		Ground	
30	GND	-		Ground	
31	RF_BG	I/O	31	2.4-GHz RF input/output	Mapped to RFID chip
32	GND	-		Ground	
34	SOP0	-	35	Optional 10-kΩ pullup if user chooses to use SWD debug mode instead of 4-wire JTAG	Mapped to SOP header
35	nRESET	I	32	Power on reset. Does not require external RC circuit	Mapped to nRESET header
36	VBAT_DCDC_ANA	-	37	Power supply for the device, can be connected to battery (2.3 V to 3.6 V)	Connected to output of voltage regulator

Module Pin No.	Module Pin Name	Type	Device Pin No.	Module Pin Description	Use (in final prototype)
37	VBAT_DCDC_PA	-	39	Power supply for the device, can be connected to battery (2.3 V to 3.6 V)	Connected to output of voltage regulator
38	GND	-		Ground	
40	VBAT_DCDC_DIG_IO	-	10, 44, 54	Power supply for the device, can be connected to battery (2.3 V to 3.6 V)	Connected to output of voltage regulator
43	GND	-		Ground	
46	GPIO1	I/O	55	GPIO	RX for flashing
47	GPIO2	I/O	57	GPIO	AUX button/TX for flashing
48	GPIO3	I/O	58	GPIO	
49	GPIO4	I/O	59	GPIO	Light Sensor Interrupt Pin
50	GPIO5	I/O	60	GPIO	Green TEST LED
54	GPIO9	I/O	64	GPIO	Red TEST LED
55-63	GND	-		Thermal Ground	

An important fact to note is the difference between “Module Pin No.” and “Device Pin No.”. Module Pin No. is referring to the pin that is enumerated on the CC3200MOD, which is the microcontroller with all the necessary passives and antenna circuit in one discrete module. On the other hand, Device Pin No. refers to the pin location on the CC3200 MCU itself.

11 Power Monitoring System

Not only can a battery fail and cause a system to fail but we must be able to keep track of the power that the system is putting out. As well as how much power each component should be receiving to ensure a properly working system. Typical things we want to be aware of is battery failures which include things such as over- charging, state of charge mismatch, over and under voltage, overheating and exceeding the lifetime of the charge cycle.

The monitoring system is a key element to producing a low maintenance device. This will be helpful when you have multiple devices connected to a network, and instead of going to each individual device and checking every single battery to make sure it has been charged or its output or input into the device is fine. It would be beneficial to the user if a monitoring system was in place in order to check for you charge, life and error in the system. This would of course relay the information back to the user in any way we seem fit.

After much thought I have decided that the monitoring, reporting, switching and charging systems will be designed individually and then integrated into the main system to form some sort of managing system. A battery management IC could be used in order to reduce space, because our project is limited with board space. So we would reduce the amount of components we would have to use and use less board space. This type of device will be a key part to our project design because of its capability to monitor the battery functionality and report back any issues back to the user without the user ever having to take apart the device and physically check. Ensuring that the battery is working properly and charged to its specified specifications to maximize the battery life of the device. This battery management IC chip meets all of our standards, requirements and would definitely save us time in having to design and build the entire system ourselves [4].

11.1 System-Side Impedance-Track Fuel Gauge with Direct Battery Connection-BQ27510-G3

Below is some information on the IC:

- The single series cell Li-Ion battery fuel gauge can be found on our board:
  - It is an integrated 2.5 VDC LDO IC chip
  - Contains an external low-value 10 mΩ resistor
- Patented Impedance Track technology:
  - Adjusts for battery aging, self-discharge, temperature, and rate changes
  - Reports Remaining Capacity, State of Charge (SOC), and Time-to-Empty

- Optional Smoothing Filter
    - Battery State of Condition (aging) approximation
    - Supports certain embedded or detachable packs with up to 32Ahr capacity
    - Accommodates pack swapping with 2 separate battery profiles
- Micro-controller peripheral provisions:
    - 400-kHz I2C serial interface to communicate
    - 32 Bytes of Scratch-Pad FLASH NVM
    - Battery Low digital output warning
    - Configurable SOC Interrupts
    - External thermistor, internal sensor, or host reported temperature options
- The board has small 12-pin 2,5 mm × 4 mm SON Package available for use [4]

12 Battery Types

One of the most important components to our project is our power system. It must be able to handle a large current draw from the system for a long period of time. This was a big change because the size restrictions of our system. We need a power supply with a larger voltage and current but must be extremely small in size. As we know this very difficult to accomplish simply because small power supplies cannot supply a large amount of power for a long period of time.

The Battery management system from above would have to compatible with the different types of technologies available. Lucky we have chosen a design that is compatible with technologies such as Li-ion/Polymer, NiMH/NiCd and Coin cells. Not only will it have to stand up to the requirement above but there are many requirements that we are looking at in order to find what works for our project. The parameters we are working with include size, discharge rate, shelf life, continuous current draw, energy density, heat, discharge rate, operating temperature, safety and of course maintenance requirements will be taken into consideration for choosing between the different technologies available.

It is also important to mention which configuration we used with battery after we have chosen because there are many different configurations in which to utilize the battery in to deliver power to our device. The battery may be placed in series, parallel or a combination of both of these. We will of course choose the most efficient configuration after the entire system has been designed. We will need to calculate the current draw by each component in order to determine the best possible technology with its best configuration. For example having the formulas to calculate the capacity of our battery pack used in the safety monitoring system to determine how much charge is left.

12.1 Coin Cell

The first technology I began my research in was Coin batteries. While looking and researching what they were. I came to the conclusion that a standard coin cell would be great for small low power projects like ours. They were cheap to purchase, and could be purchased in large quantities. This would work perfect for our project as a possible means to power our device [5].

Going more into specifics I was looking into the CR3032 for our project because it is a lithium battery and a standard coin battery. With an overwhelming amount to choose from I looked for performance, application, size and availability as my parameters for choosing a power supply. This standard Cr3032 lithium coin battery could be bought from any store that sells battery because it is a standard battery used in many different applications. Further research was put in a table which can found down below, to make comparing the different technologies easier [5], table 12.

TABLE 12: COIN CELL BATTERY CHARACTERISTICS

Model	Electrical Characteristics (20°C)			Standard Load	Dimensions		
	Nominal Voltage (V)	Nominal Capacity (mAh)	Operating Temperature (°C)	Continuous Drain (mA)	Diameter (mm)	Height (mm)	Weight (g)
Cr3032	3	500	-30~+60	0.20	30.0	3.20	6.28

Key feature of the Cr3032 Lithium coin Battery listed down below [5]:

- Voltage of 3 volts - twice that of conventional dry batteries
- Extremely small self-discharge for long service
- Self-discharge for longer shelf life
- An extensive operational temperature range
- Compact and lightweight
- Exceptionally high energy density per unit weight
- Very safe
- Extremely strong load pulse characteristics
- Good operating temperature limits as specified in the graph above

12.2 Nickel Metal Hydride (NiMH)

The second option to be discussed is the nickel metal hydride (NiMH). The size of the battery is nearly the same when compared to your standard alkaline battery, with a nominal voltage of 1.25V. These batteries are typically lower in cost when comparing the different rechargeable batteries available on the market but suffer from lower densities when compared to the Lithium polymer. A positive aspect about the NiMH battery is the fact that is requires a stringent charging curve when compared to the others, which in turn makes for a lower costing unit of choice when compared to rechargeable technology. Important detail that must be considered when choosing between the technologies available is the charging characteristic [6].

The first characteristic for this technology to be considered is the self-discharge rate which compared to the other technologies is relatively high, at a rate of approximately 20%-30% per month. The rate must be taken into account when designing the charging system to make sure that we have an efficient system that is capable of providing the power need to

the system in an efficient manner. With this comes talk of overcharging and the effects that it may have on the battery and ultimately the system. Overcharging should definitely be avoided at all cost. When it comes to overcharging this technology lacks tremendously because there will be a decrease in the cycle life of the battery and it could cause many safety issues to arise if not monitored carefully [6].

The second characteristic being considered is what kind of memory effect the battery would have if the battery was fully discharged before any recharging could be done. One of the key benefits was that the NiMH battery does not require the battery to fully discharge is capacity in order to remain functioning properly. This would have to be taken into consideration when designing the charging system for our product [6].

12.3 Alkaline

Everyone may be familiar with this battery, and is probably the most common type of disposable battery. These types of batteries have been around for decades, so they can be found nearly anywhere. This also means that finding holders and accessories shouldn't be as difficult as some of the not so common battery types. This is why we felt like this would a solid choice for our project.

These batteries are relatively cheap, safe to use and are available everywhere, but unfortunately they are not rechargeable. These facts are simply obvious and common knowledge to most, but what maybe most people may not know is that AAs and AAAs are the most common alkaline battery and have an output nominal voltage of 1.2V (around 1.5V at first use). For our purposes, we would have to combine this in packs of 3 or 4 in order to run feed our MCU with about 3.3V. Alkaline batteries provide the longest service life for high drain devices and they have one of the best shelf life for storage, which would important for a project like ours where the device may not constantly being used. Further details can be seen by table13 down below of the alkaline characteristics [7].

TABLE 13 ALKALINE CHARCTERISTICS

Model	Alkaline Battery
Nominal Voltage/Cell	1.5 Volts
Maximum continuous current	High
Maximum pulse current	High
Operating Temperature Range	-30°C to 55°C (-20°F to 130°F)
Capacity Retention	> than 97% after 12 months @ 21°C(70°F) >than 85% after 5 years
Shelf life	10 years @ 21°C (70°F)

12.4 Lithium Polymer:

The last option of battery technologies we looked into was Lithium Polymer (LiPo) battery. They are very useful in embedded electronics and small projects such as ours. They also are said to have the highest density available on the market. This technology can be found predominately in cell phones, therefore is easy to find in the market and can be found at a reasonable price. They also require special charging units and not having the correct one could be bad for your battery, as well as your system if your battery isn't performing like it supposed to [8].

12.4.1 Nominal Voltage

Going into specifics, a single LiPo cell has a nominal voltage of 3.7V. When it is fully charged it will read 4.7V, but will quickly return down to a range within the norm. When the cell has been depleted, the voltage will drop down to 3V. This means that we will have to do more research into how our device will handle the different voltages and what actions it will take to assure no harm to the system. It is possible to connect two of these batteries either in series or in parallel depending on our needs. The actually configuration will be determined later after further research and once the battery is chosen [8].

12.4.2 Connectors

Most LiPo batteries come with a standard 2-pin connector. One is for positive terminal (typically red) and the other negative (typically black). A great approach we found was to connect a JST connector in order to prevent the battery from being plugged in the wrong way. This would also prevent any damages to the system which is an added bonus. The JST connector is a friction based connector and is it common to use pliers to remove the battery gently [8].

12.4.3 Charging and Discharging

When it comes to charging our system, there are a number of different low-cost chargers that are created specifically for LiPo batteries. The most common chargers use USB to charge the battery. There is no way of charging the battery without a special charger and is important to take note of this. It is also very important to not overcharge the battery because it could be harmed by doing so and damage our product. This is one of the reasons I chose to focus on battery protection designs for our product.

We must not forget that not only overcharging the battery would cause damage but that discharging the cell past a certain threshold would also cause damage. So in order to prevent this there is a built in circuit in most cells that come standard built into the top of the cell that would shut down the battery if it senses that the threshold voltage drops (normally 3V) [9] [10].

The LiPo battery has a low internal discharge rate. This means that this makes them a good choice for low power requirement projects such as our project that has to run for many days consecutively. These batteries also can source multiple amps continuously giving it major points as a choice for our project. The short circuit protection feature that it has built in detects when there is a short in the system and will automatically shut off the system [8] [9] [10].

13 Battery Protection

After investing hours on designs for our project, we wouldn't want to have our battery, which is going to be powering our system to suddenly cause harm to itself or our system. This is why I feel like I needed to talk about how the protection of the battery would be an important component in our product.

13.1 Li-Ion/Li Polymer Battery protection IC-BQ29700

The BQ29700 battery cell protection device provides an accurate monitor and trigger threshold for overcurrent protection for the duration of high discharge/charge current operation or battery overcharge conditions [4] [11].

The BQ29700 device provides the protection functions for Li-Ion/Li-Polymer cells, and monitors across the external power FETs for protection due to high charge or discharge currents. In addition, there is overcharge and depleted battery monitoring and protection. These features are implemented with low current consumption in NORMAL mode operation [4] [11].

There is also a timer delay for the recovery period once the threshold for recovery condition is satisfied. These parameters are fixed once they are programmed. There is also a feature called zero voltage charging that enables depleted cells to be charged to an acceptable level before the battery pack can be used for normal operation. Zero voltage charging is allowed if the charger voltage is above 1.7 V. Features and further information on the IC-BQ29700 are listed below [4] [11]:

- Input Voltage Range Pack+: VSS – 0.3 V to 12 V
- FET Drive:
    - CHG and DSG FET Drive Output
- Voltage Sensing Across External FETs for Overcurrent Protection (OCP) Is Within ± 5 mV (Typical)
- Fault Detection system
    - Overcharge Detection (OVP)
    - Over-Discharge Detection (UVP)
    - Charge Overcurrent Detection (OCC)
    - Discharge Overcurrent Detection (OCD)
    - Load Short-Circuit Detection (SCP)
- Zero Voltage Charging for Depleted Battery
- Factory Programmed Fault Protection Thresholds
    - Fault Detection Voltage Thresholds
    - Fault Trigger Timers
    - Fault Recovery Timers
- Modes of Operation Without Battery Charger Enabled
    - NORMAL Mode ICC = 4 µA
    - Shutdown Iq = 100 nA
- Operating Temperature Range TA = –40°C to 85°C
- Package layout: 6-Pin DSE (1.50 mm × 1.50 mm × 0.75 mm)

14 Battery Comparison & Conclusion

After doing much analysis and comparison of different battery technologies we came up with table 14 [11] which compares the different batteries and ultimately choosing one of these batteries for our project based on the table and research below. Table15 was used in order to further decrease the options down and compare them even further.

TABLE 14 BATTERY TECHNOLOGIES COMPARISON CHART 1

	Lithium ion Polymer	NiMH	Alkaline	Coin Cell
Energy Density(Wh/kg)	90-190	60-120	≈110-156	≈220
Cycle life (approx.)	1000	300-500	N/A	N/A
Charging (If applicable)	1 hr. or less	1-1.5 hrs.	N/A	N/A
Overcharging (If applicable)	Low	Low	N/A	N/A
Self-discharge	5% in 24h, then 1-2% per month (plus 3% for safety circuit)	10-15% in 24h, then 10 15% per month	2-3% per year	3% per year
Nominal Voltage	3.7V	1.25V	1.5V	3.0V
Rechargeable	Yes	Yes	No	No
Size(Height)		43.0mm	44.50mm	3.2mm

TABLE 15 BATTERY COMPARISON CHART 2

Typical Features	NiMH vs. Lithium ion	NiMH vs. Alkaline
Rated Voltage	1.25V vs. 1.5V	1.25V vs. 1.5V
Discharge Capacity	NiMH will not last as long as primary lithium (single cycle)	NiMH lasts longer in high drain, less in light drain devices than alkaline

Typical Features	NiMH vs. Lithium ion	NiMH vs. Alkaline
Recharge Capability	Several hundred cycles for NiMH, N/A for lithium primary	Several hundred cycles for NiMH, N/A for alkaline primary
Discharge Voltage Profile	Both relatively flat discharge	NiMH is flat vs. sloped for alkaline
Self-Discharge Rate	NiMH retains 50-80% @ 6 months Lithium retains 80% @ 15 years	NiMH retains 50-80% @ 6 months Alkaline retains 80% @ 7 years
Low Temperature Performance	Lithium better than NiMH	NiMH better than alkaline
Battery Weight	Lithium is lighter	Alkaline is lighter
Environmental Issues	Recycling options available for NiMH and lithium	Recycling options available for NiMH and some alkaline

After countless hours of researching the ideal battery for our project we are looking at using the lithium ion polymer rechargeable battery. Table 14 [11] & table 15 [11] demonstrated all the characteristics of each technology for us and using these tables, with our factors and restraints be came up with what we thought would be our best choice.

14.1 USB Connector

The USB connection is an important part of the system because it connect the outside world to our device. A brief understanding of the cabling of USB, ports, and transmission rates were all considered in choosing the right connectors for our project. Many different factors went into choosing the right connection for this project. Those factors will be introduced and explained down below.

14.2 Cabling

We first start with what is USB because it is important in understanding why we chose what we did. A USB twisted pair is when the "Data +" and "Data −" conductors are twisted together in a double helix formation. The wires are enclosed in a further layer of shielding so there won't be interference between wiring. The reason we have data cables USB 1.x and USB 2.x being a twisted pair is to reduce the amount of noise and crosstalk that could occur in our project. We wanted to make sure that the board wouldn't interfere with any surrounding cables. USB 3.0 cables contain twice as many wires as the previous USB 2.x

which supports SuperSpeed data transmission, making it larger in diameter but a better overall choice for our project [12] [13].

Then the next important parameter is how the standards comes into effect each USB type. The USB 1.1 standard stipulates that a standard cable can have a max length of 5 meters with devices operating at a Full Speed of approximately (12 Mbit/s), and a max length of 3 meters with devices operating at a Lower Speed of approximately (1.5 Mbit/s) [12] [13].

When it comes down to it the USB 2.0 provides for a maximum cable length of 5 meters for devices running at Hi Speed (480 Mbit/s). When adding USB device response time, delays from the maximum number of hubs added to the delays from connecting cables, the maximum tolerable delay per cable amounts to 26 ns. The USB 2.0 specification calls for that cable delay be less than 5.2 ns per meter (192 000 km/s, which is close or nearly the maximum achievable transmission speed for standard copper wire) [12] [13].

The USB 3.0 standard does not directly stipulate a max cable length, requiring only that all cables meet an electrical specification: for copper cabling with AWG 26 wires the max real-world length is 3 meters or (9.8 ft.) [12] [13].

14.3 Sleep-and-charge ports

The next important factor is a USB port representing when it is in sleeping or when it is in charging. Sleep-and-charge USB ports can be used to charge your everyday electronic devices even after the computer is switched off. Typically, when a computer is powered off the USB ports are also powered down, stopping phones and other devices from charging from its port. But when it comes to the sleep-and-charge USB ports, they continue to remain powered on even when the computer is turned off. On normal laptops, charging devices from the USB port when it is not being powered from AC drains the laptop battery quicker; although most laptops now have a capability to stop charging if their own battery charge level gets lower than a set threshold. This is an important factor for considering how power could be delivered to our device when an outlet may not be available for recharging the battery in the device [12] [13].

Essentially what we want is most laptop models to be able to charge our device properly when a wall plug is not an option. We also would like to make it possible to plug in our device to any workstation model with a USB port and have it charge it, this comes with USB standards discussed above. Another cool feature would be having our device charge while having the laptop put into sleep mode and continues to charge our device. All of these very much possible with today’s technology [12].

14.4 Transmission Rates

Another factor when considering the many options available to us in the market would have to be the speed, in which things can be transmitted. Further details into the different USB will be further discussed.

The hypothetical max data rate achievable in USB 2.0 is 480 Mbit/s (60 MB/s) and is shared between all attached devices. Some chipset manufacturers overcome this bottleneck by providing multiple USB 2.0 controllers within the Southbridge [13].

According to routine testing performed by CNet, write operations to typical Hi-Speed (USB 2.0) hard drives can sustain rates of 25–30 MB/s, while read operations are at 30–42 MB/s; this is 70% of the total available bus bandwidth. For us this means many great things. This means when we plug our device into a hub sort of center data can flow a quicker rate the better the USB transmission rate is. This is important to us when it deals with uploading information or debugging your device on a computer or laptop [12] [13].

For USB 3.0, a normal write speed is around 70–90 MB/s, while read speed is 90 110 MB/s. Mask Tests, also known as Eye Diagram Tests, are used to determine the quality of a signal in the time domain. They are defined in the referenced document as part of the electrical test description for the high-speed (HS) mode at 480 Mbit/s. which is important for us because we want to have quality signal at all times [12] [13].

According to a USB-IF chairman, "at least 10 to 15 percent of the stated peak 60 MB/s (480 Mbit/s) of Hi-Speed USB goes to directly above—the communication protocol between the card and the peripheral. Overhead is a component of all connectivity standards". The Table illustrating the USB data rate can be found down below (Table 16).

TABLE 16: USB DATA RATES

Mode	Gross data rate	Introduced in
Low Speed	1.5 Mbit/s	USB 1.0
Full Speed	12 Mbit/s	USB 1.0
High Speed	480 Mbit/s	USB 2.0
SuperSpeed	5 Gbit/s	USB 3.0
SuperSpeed+	10 Gbit/s	USB 3.1

The Table 16 located above clearly illustrates the best USB standard to incorporate into our device. Using the SuperSpeed+ for our project would make sense for us to use because we want the fast possible transmission for our device.

14.5 Mini-USB &Micro-USB

The mini USB is one of two technologies we are considering for our product. The mini-USB was designed to have a life cycle of approximately 10,000 insertion cycles and removal of about 5,000. Compared to the mini-USB the micro-USB was designed to have an insertion and removal of about 10,000 cycles. Not only was the total insertion and removal cycles increased but also designs of each standard USB had to be taken into consideration. We wanted to choose the technology that had backwards compatibility and extended life so replacement wouldn't be occurring sooner. We are leaning towards the

micro-USB since has become more of a standardized method for connecting devices to a power supply [14].

15 Charging Circuit

The battery that we will have in place in the system will contain a battery that is rechargeable. The rechargeable battery needs to have circuitry in place. The circuitry in consideration contains an IC that has that has the ability to monitor the battery and charge it from the 5V voltage that it receives from the USB connector. This section will describe briefly some of the ICs and reference circuits that we considered during the design of the system.

15.1.1 Texas Instruments BQ24210

This integrated circuit comes equipped with input voltage dynamic power management. It can also provide thermal regulation protection for output current protection. It also comes with battery short protection circuit. The figure 9 located below gives a reference schematic for this IC in the system [15].

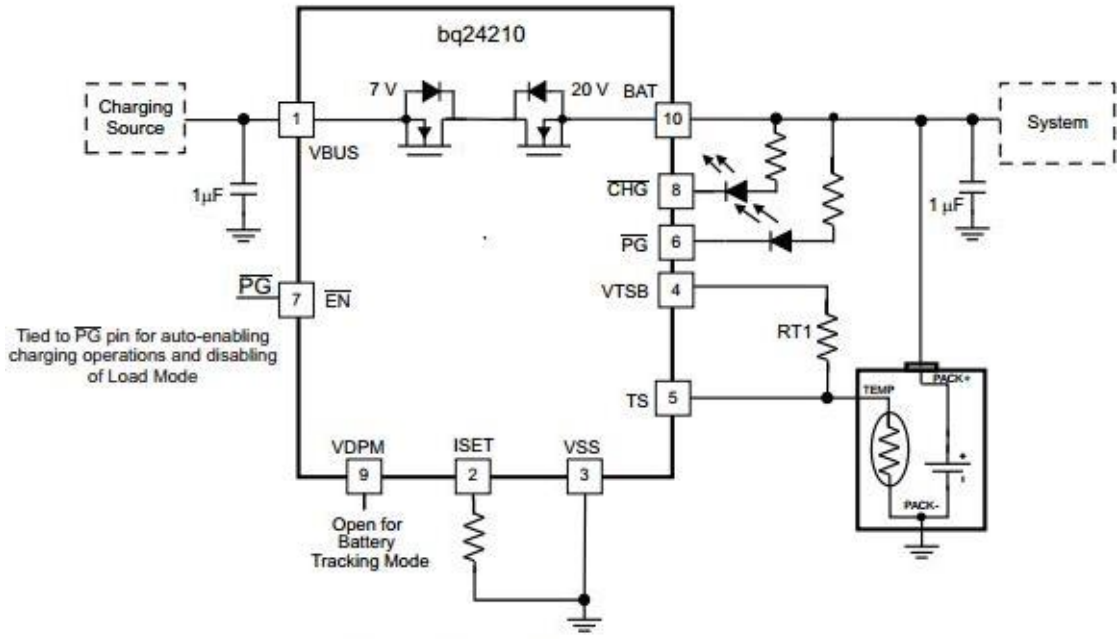

FIGURE 9: BATTERY CHARGER REFERENCE CIRCUIT

(REPRINTED WITH PERMISSION FROM TEXAS INSTRUMENTS)

16 Voltage Regulator

The switching regulator is an important component for our project because its ability to take and convert power efficiently. As well as increase our design flexibility because we don't have to worry about choosing components that use the same voltage to operate. Basically this means that we can get multiple output voltages from a single input voltage.

We will be looking at the four different switching converter types which include Buck, Boost, Buck-boost, flyback.

16.1 Switching Regulators

The first type of switching converter is the Buck and it is primarily used to reduce a DC voltage to a lower DC voltage. The Boost switch provides an output voltage that is much higher than your original input. The Buck-boost inverting switch as its name suggests generates an output voltage opposite in polarity from the original input it received. The last switch is called the flyback switch which is essential when you want an output voltage that is less than or greater than the input that was being generated, as well as generating multiple outputs.

16.2 Fundamentals

Before going into more details a quick explanation into the theory of this element is needed to understand why we decide to go with technology.

The law of inductance is an important concept. One must understand what and how it works. Simply put, if we force a voltage across an inductor then the current passing the inductor will vary with time. This is a fundamental concept which is also true for a time varying current forced across an inductor will give you a constant voltage.

So why is this important? This concept of inductors is needed to fully understand a technology known as Pulse Width modulation (PWM). Pulse width modulation is a standard form of output for voltage regulators in this section. The basic understanding that the feedback loop adjust or corrects the output voltage by changing switching element.

In the figures provided below we are assuming that we have a square wave pulse input and is then passed through the filter which give you a DC output voltage that is equal in amplitude to the pulse multiplied by the duty cycle. The formula is provided above. The duty cycle as defined above is simply the switch Ton divided by the total period Tp. This relationship helps explain how we can control the output by changing the ON time of the switch. Which is an essential part and fundamental in understanding which switch to pick for which part of the project.

16.3 Topologies

Since we knew our battery voltage was going to be higher than that of the microcontroller and other IC devices our regulator type of choice had to be buck and buck-boost regulators. Our goal was to optimize the circuit in order to achieve the best performance. The three factors we took into consideration for designing our circuits for which topology to use was space, cost and efficiency.

16.3.1Topology 1

The figure10 below depicts the topology involving a possible buck converter design:

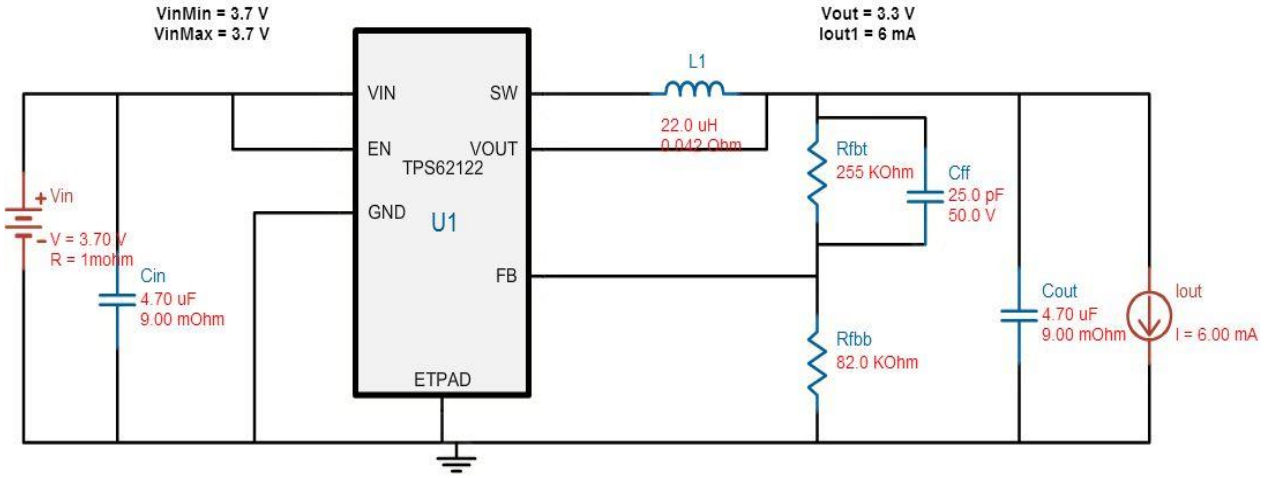

FIGURE 10: FIRST REGULATOR REFERENCE CIRCUIT

(REPRINTED WITH PERMISSION FROM TEXAS INSTRUMENTS)

The data for the circuit (figure 10) above was collected from TI's webench:

- Efficiency: 95%
- BOM cost: $0.96
- Footprint: 179

16.3.2 Topology 2

The figure11 located below shows a possible buck voltage regulator design:

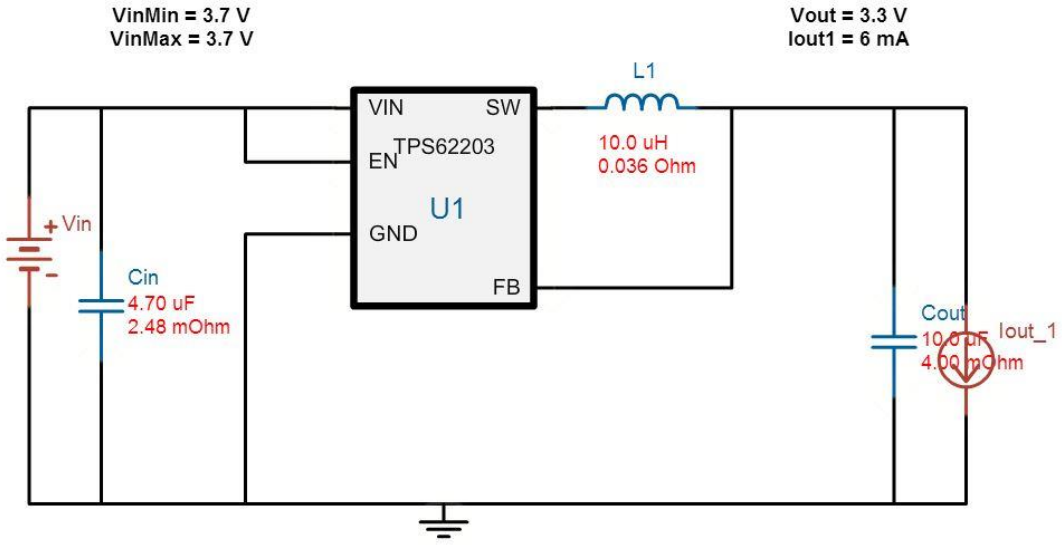

FIGURE 11: SECOND REGULATOR REFERENCE CIRCUIT

(REPRINTED WITH PERMISSION FROM TEXAS INSTRUMENTS)

The data for the circuit above (figure 11) was collected from TI's webench:

- Efficiency: 85%
- BOM cost: $1.79
- Footprint: 78

16.3.3Topology 3

The figure12 below depicts a Buck-Boost (Inverting) regulator:

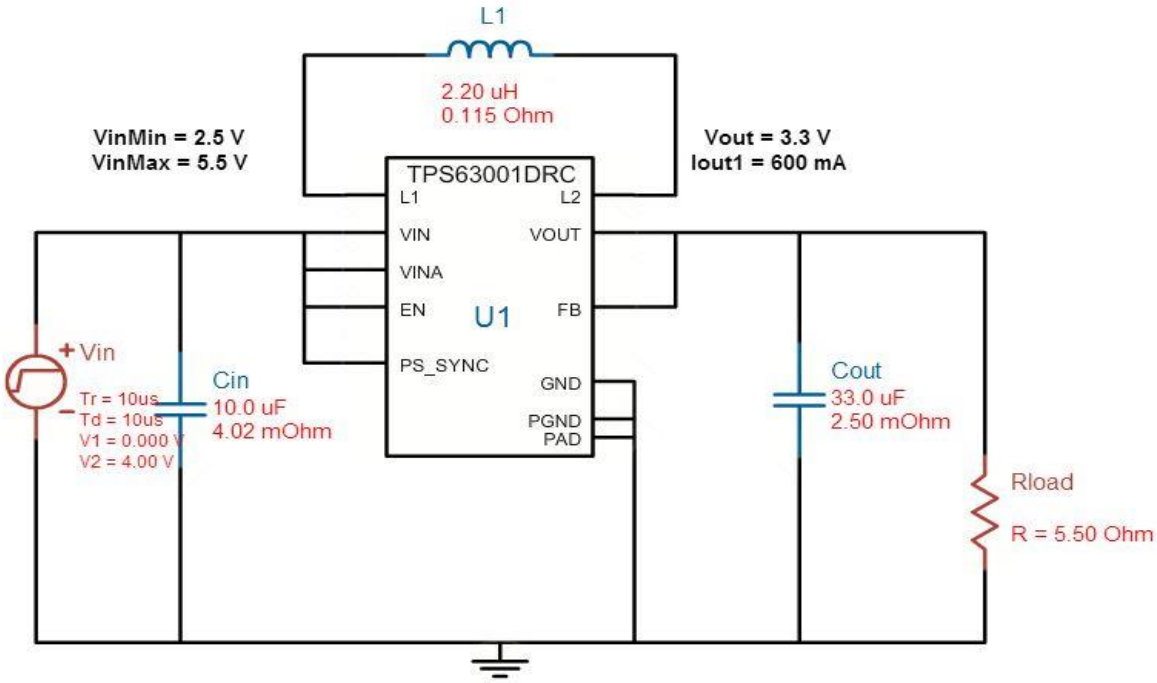

FIGURE 12: THIRD REGULATOR REFERENCE CIRCUIT

(REPRINTED WITH PERMISSION FROM TEXAS INSTRUMENTS)

The data for the circuit above was collected from TI's webench:

- Efficiency: 85%
- BOM cost: $1.79
- Footprint: 78

In the figure12 above we see that this circuit will be taking a DC input voltage and producing a DC output voltage with an inverted polarity. The output of the regulator may be smaller or larger than the original input voltage which works for our design because we can use this regulator in multiple parts of the circuit.

16.3.4Topology conclusion

Based on the data collected and after much research into each regulator our main characteristics that we need to focus on for this project was board space. We need every

component to fit neatly into a very tiny space provided. This restraint out weighted efficiency and cost because there is was point in having a cost efficient part with no place to put it. For our project TPS63001DRC voltage regulator topology 3 is the best choice for the final prototype design because of the huge restriction of board space due to the size of the project, going with topology 3 was the smartest decision for our project design.

17 Testing

In order to make sure our designs and system is functioning properly, as well working like it's supposed to be different parts of the system must be tested.

17.1 Electrical System Testing

The purpose for testing the electrical system is so that we may verify that our overall design is working properly with this system. The testing will be broken down into a few different sections.

1. Charging Circuit
2. Voltage Regulator
3. Battery
4. Battery Protection

17.1.1 Charging and Battery

As its name would suggest the charging circuit test will be to test to make sure that our circuit is properly providing the proper amount of current to our device and to assure that once our device is fully charged no overcharging should take place, which would cause harm to our device. The test will include attaching a load to the circuit and measuring the load. To fully charge in a reasonable time the circuit must output a certain amount of current, which we plan on measuring.

17.1.2 Voltage Regulator

Using a multi-meter we can test that the output under a load is what we are expecting. This condition will be tested for varying voltage inputs to the regulator, and with different loads at the outputs.

18 Overview of the Software System

The T.A.G.G. software system has three major components, the embedded software system running on the CC3200, the mobile application running on the android device, and the web service running on the cloud. The interactions between these systems is summarized diagrammatically below in figure 13.

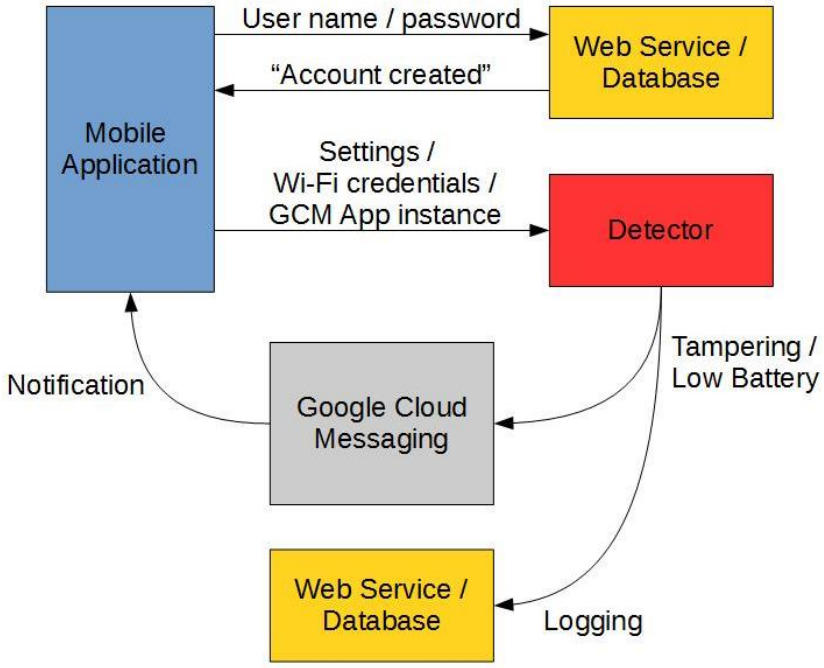

FIGURE 13: OVERALL SOFTWARE SYSTEM BLOCK DIAGRAM

The user's experience begins with the user creating an account through the mobile application. The web service receives this request with the new users name and password and checks to see if the user name is available. If it is, then the web service creates a record of the new user account and responds to the mobile application with a message indicating that it can continue. If the username is not available, then the user is instructed to try a different user name.

Next, the provisioning and settings transfer step begins. From the mobile application the user enters sensor settings for their detector, a name for the detector, and the network SSID and password. Holding down the button on the detector makes it so the detector can receive the information from the mobile application. The data is transmitted to it through TI's smart link protocol. The detectors notification information and settings are transmitted at this time as well. If successful, the user is sent a notification indicating that the detector is now ready to be placed. The detector sends notifications to the mobile application through Google Cloud Messaging Service, or GCM. These notifications are also logged in a database by the web service.

There are three types of future interactions between the mobile application and the detector. The first is in the case of some form of tampering being detected. The detector will send a notification through GCM to the mobile application. Following this, the detector will wait for 59 seconds before reactivating, going back into detecting mode. The second possible

future interaction occurs when the detector is running out of battery, in which case the user will receive a notification as well. The third interaction occurs when the user would like to modify the settings stored on the detector or when the detector battery has completely run out of power. In these cases, the user will need to press the button on the detector and use the mobile application to reload settings onto the detector before it is used again.

The user at some point may wish to remove there user account. This feature is supported through the mobile application. It interacts with the web service to remove the users account and group ID from GCM to prevent the mobile application from receiving future notifications from the detector.

Aside from the provisioning step, which utilizes TI smart link protocol and multicast domain name service, or mDNS, during the settings transfer, the systems above communicate using http POST requests. Data payloads utilize JSON data-interchange format. Using this common application layer protocol allows the use of many premade libraries and simplifies the interactions between T.A.G.G's software components.

19 Microcontroller Software

19.1 Overview

The microcontroller software system has several responsibilities. A few of the major ones are to successfully connect to the user's Wi-Fi network, detect tampering through the use of the attached sensors, communicate the tampering event through Google Cloud Messaging, detect battery status and communicate low battery status through Google Cloud Messaging, and to minimize the power consumption of the battery by placing the microcontroller in hibernation mode whenever possible. A program flow to accomplish these tasks is shown in figure 14.

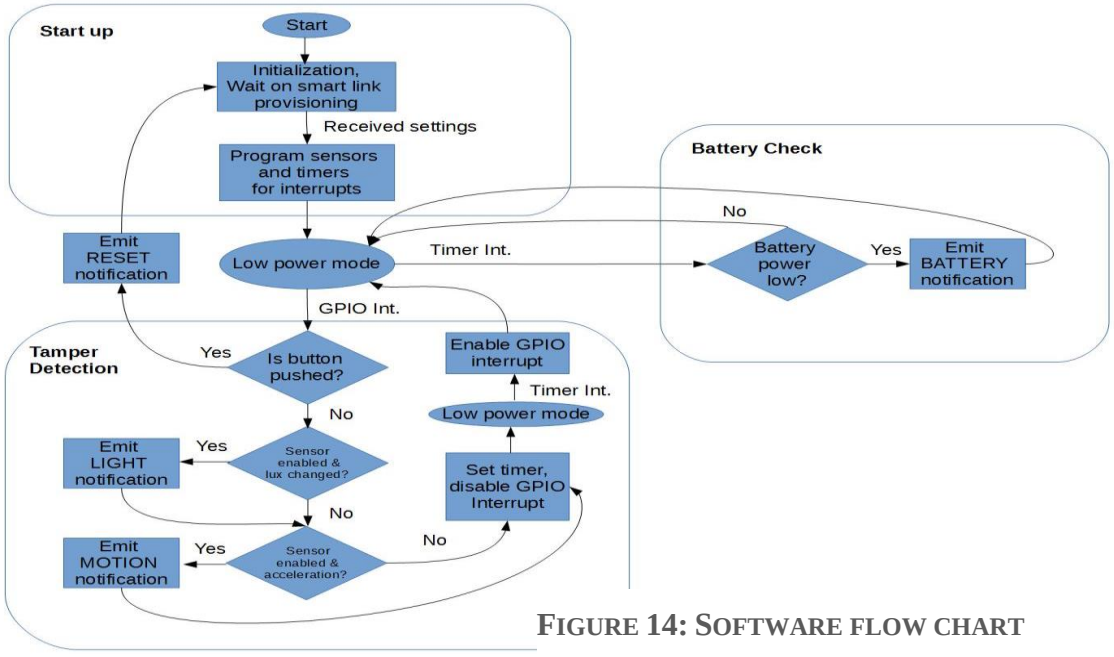

FIGURE 14: SOFTWARE FLOW CHART

The following sections dive into each of the parts in more detail beginning with the startup section. The startup section is the first interaction the user will have with the detector.

19.2 Provisioning and Transmitting Additional Data

The first thing that the software running on the microcontroller must do is communicate with the mobile application and receive the information it needs to continue its execution, including the information needed to join the user's Wi-Fi network. Considering that the detector provides no way for the user to enter the network's SSID or password directly through any sort of keypad or touch screen, some other solution has to be found. Texas Instruments suggest several possible ways to accomplish this. The viable options in the T.A.G.G's case are putting the detector into access point mode and creating its own WI-FI network temporarily or using TI's SmartConfig Technology. [16]

It was determined that amongst these options, TI's SmartConfig provisioning would be the simplest and easiest to integrate into the rest of the T.A.G.G's software design. The other option, of placing the detector into access point mode, would place an undue burden on the user by forcing them to change WI-FI networks on their device to join the detector's network. Additionally, using the access point method would complicate the program flow on the microcontroller by forcing it to act as a web server for the user's phone. TI's SmartConfig technology eliminates these additional steps and provides a ready-made solution for connecting the CC3200 to the user's Wi-Fi network. Using the smart link library provides a set of functions to set the CC3200 into a default state and wait for an application such as the Android application to transmit the network information. An open source Android application provides the libraries and an example of how to write code to allow the user to input the SSID and password and transmit this information. On start up the CC3200 is programmed to wait for this information. The user initiates the connection using the mobile application. Following the provisioning step, the detector is connected to the Wi-Fi network.

One unfortunate consequence of using the SmartConfig technology is that the provisioning step does not directly accommodate an extra data field which could be used to communicate the additional sensor settings, data vital to application. Also complicating this issue, is that neither the Android application nor the CC3200 know each other's IP address following the provisioning step. A solution to this problem is using the mDNS (multicast domain name system) to communicate this information. The CC3200, immediately following the provisioning step, becomes a mDNS listener. At the same time, the Android application becomes a mDNS advertiser. The Android application can then advertise the additional information to the listener. The CC3200 responds by registering for the service and then moving on to its next task of setting its sensors. The Android Application when it sees that the CC3200 has registered for its mDNS service can assume that the CC3200 has successfully received the information it needs and can stop advertising the service.

The necessary information for the CC3200 is encoded in the mDNS advertiser's text sent by the Android application. The fields in the text are delimited by a special character. The encoding is as follows

<user-name><user-password><detector-name><GCM-Application-Instance-ID><light-sensor-settings> <motion-sensor-settings>

It is the microcontroller's responsibility to parse this string, convert the string representation to values, and store the values in the variables it needs to set the sensors.

19.3 Network Communication

The network communication on the microcontroller is handled by the CC32XX SimpleLink Library. There are three types of communication used, not including i2c used to communicate to the sensors. The first is TI's proprietary SmartConfig used in the provisioning step. The second is the multicast Domain Name Service, or mDNS, communication that takes place immediately following the provisioning step to transfer needed data to the CC3200. The third type of communication is http POST requests made by the detector to Google Cloud Messaging.

The majority of functions needed for TI's SmartConfig communication are declared in the simplelink library header file wlan.h. The steps required to successfully complete the provisioning step are outlined below.

1. Delete existing network profiles. This assures that the CC3200 connects to the newly found WI-FI network.

2. Set the Wlan policy to SL_POLICY_CONNECTION. In this policy the CC3200 attempts to automatically connect to one of its stored profiles. This will happen each time the connection fails.

3. Call the function which begins the Wlan SmartConfig connection attempt. This function will attempt to gain the SSID and Password of the network.

4. Call a function which waits for an asynchronous event which handles Wlan events. From inside this event handler the SSID can be copied into memory if needed upon connection.

Following this procedure, the CC3200 is connected to the WLAN network. If this procedure is unsuccessful in steps 3 and 4 the CC3200 will continue waiting indefinitely to connect to the WLAN. This will consume more battery then the proper operation of the detector. [17]

The next step is the mDNS step. It is needed for transmitting the sensor and user information and application instance ID for notifications. Again, the functions needed to accomplish this are provided by the SimpleLink Library. The function call that is necessary to accomplish this is *sl_NetAppDnsGetHostByService()*. The programmer must provide this function with the name of the desired service, and some other network details. Returned is the IP address, port, and a piece of text provided by the service. In our case we

are exclusively interested in the text, as it contains the information needed to continue the program. The underlying protocol for this step is UDP, but this is certainly abstracted away from this application with the SimpleLink Library. [18]

19.4 Notifications

For sending notifications to the user application, http posts to GCM are used. This application layer protocol is provided for us by TI's HTTP Client Library. The first step in sending the http post is connecting to the HTTP Server, in this case the CC3200 is the client and GCM is the server. A function, provided by the smart link library, is first used to resolve the server name to an IP address. It is used to resolve the host name android.googleapis.com to an IP address. Following this, a function provided by the HTTP Client Library, HTTPCli_connect(), is used to connect the CC3200 to this IP address.

We can now connect to the server and make the post request. After filling out the appropriate fields in a library defined data structure, the following sequence of function calls is used to perform the request. The somewhat self-explanatory functions are HTTPCli_sendRequest(), HTTPCli_sendField(), HTTPCli_sendRequestBody(). The information needed for the request fields is found in the previous connection step, with the exception of the api key and the application instance. The api key is predefined in the CC3200 code. The application instance was found in the mDNS step. [19]

The data used in the request body (the notification message) is one of several predefined options, each corresponding to a different event. Which option is chosen depends on the circumstances under which the http request is made. In all cases, the data body is in JSON format and contains the notification message for the mobile application. The response from the server can be checked for an error in both the connection step and the post step. If either step fails, it is repeated until there is success.

19.5 Power Management

One of the main goals of the T.A.G.G system is to have the detectors operate for as long as possible without recharging. The main way of accomplishing this is to have the CC3200 and its attached sensors spend as much time in hibernate, or low power, mode as possible.

The only way to accomplish putting the CC3200 into low power mode if using the CC32xx Power Management Framework is with the function cc_idle_task_pm(). It is a function provided by the cc3200-SDK. This function will automatically attempt to push the CC3200 into the lowest power consuming state possible. Which power modes are valid choices is a decision made by the programmer. There are four possible power saving modes for the CC3200 to be in. They are listed below with some of their more relevant features.

S1: Sleep – Disables peripheral clocks by default, wakes from any interrupt, CPU context and RAM retained, 80 MHz clock.

S2: Deep Sleep - Disables peripheral clocks by default, wakes from any interrupt, CPU context retained, but RAM retention optional, and 40 MHz clock.

S3: LPDS (Low Power Deep Sleep) – CPU context not retained, wakes from GPIO (1 pin only), timer, or network interrupt, connection to AP is retained, .25 mA current draw, and 32,768 kHz clock.

S4: Hibernate – CPU context and RAM not retained, wakes from GPIO or Slow Clock Counter, connection to AP is lost, wake up source cannot be identified, 4 uA current draw, and 32,768 kHz clock. [17]

It was tempting when first developing the system to consider the use of LPDS mode. It would greatly simplify the design. Using LPDS would prevent the need to use the on board flash on the CC3200 to write to a file to save all of the information needed between hibernations, namely the networking, notification, and sensor data. Also, using LPDS prevents the need to reconnect to the access point after every hibernation, although admittedly this point may be moot if the detector has not communicated with the access point for a long enough time. Finally, another advantage would be that LPDS mode would allow for the detector to be woken up by a network request, allowing the user to choose when to update the data on the device.

These are major advantages, however there is a major drawback to LPDS mode. Considering that we are designing this detector with low power consumption as a major goal, it is unreasonable to take a 250/4 = 62.5 times increase in current consumption to make the programming of the device easier or to obtain the additional features given above. It was decided therefore, that hibernate mode would have to be used, along with the additional burden of using a file system to maintain enough state between hibernations to make the system operational.

There is an additional complication to the program flow created by using hibernation mode as well. Coming out of hibernation mode technically requires that the program flow begin back at the entry point of the program, which for us is the start of main. This is resolved by adding the following line of conditional code to the top of main and then branching to the appropriate code as though our program was following the flow as shown in the diagram above. [17]

```
if(MAP_PRCMSysResetCauseGet() == PRCM_HIB_EXIT)
```

Power consumption can be additionally reduced by putting the accelerometer into low power mode. In the case of our accelerometer, the Bosch BMA222, the power mode desired is appropriately named low-power mode. In this mode the accelerometer will be periodically awakened (the period being programmatically selected) and automatically made to take a sample. This sample can trigger an interrupt based on some programmable conditions. This will cause a change in the value on a pin on the BMA222 which in turn will cause a GPIO interrupt on the CC3200. With a sleep duration of 500 ms the average current consumption is .9 uA. The details of implementing this involve writing to specific registers in the BMA222 and are dealt with in depth in the section devoted to communicating with the accelerometer. [2]

Unfortunately, the TSL2561 light sensor offers no such low power mode which will still provide interrupts. It has a typical supply current of 240 uA based on sampling time. This

is for the most part a fixed value and will have to be factored into the battery life should light detection be desired. If light detection is disabled on the detector, the light sensor can be power down, in which case it only draws 3.2 uA of current. [1]

19.6 To RTOS or to not RTOS?

The major reasons why one would use a real time operating system, or rtos, include the need for a multi-threaded program execution and its associated synchronization primitives, a need to use a driver only supported by the rtos, or the desire to use some other feature like system logging, a file system, etc… provided by the operating system. The first reason is in fact the most important. In the case of the detector's software system, the program flow is simple and sequential. Although there is networking, which may have a high latency relative to most of the other parts of the program, the overall program flow must wait for these networking portions of the program to complete before continuing in their execution. If the detector involved more user interaction this would not be the case, so the system design would be forced to use an rtos. But for this system, an operating system is not needed.

In addition to this, many of the features offered by TI-RTOS are actually built on top of the middleware libraries used directly within our application. Examples of this are the power management system and the GPIO libraries. Using TI-RTOS to access them would have made them thread safe, a necessity if separate threads of execution were accessing them concurrently. Considering that the application is single threaded, using this rtos would have provided little extra.

19.7 Long Term Storage through Files

As was noted in the power management section, when the CC3200 goes into hibernation mode it does not retain its RAM. In order to allow the CC3200 to come out of hibernation mode and perform all the networking and sensor operations some form of persistent storage is needed. This is where files come into play.

TI's SimpleLink API provides the functions we need to write to files. These files are written to SFlash by default and will be persistent between hibernation periods. The functions that will be required for this application are sl_FsOpen(), sl_FsWrite(), sl_FsRead(), and sl_FsClose(). What these functions do should be self-evident by their names. [19]

To understand where in the program flow these file operations will need to take place, it is easiest to look at an expanded version of the program flow diagram as seen in figure 15 below.

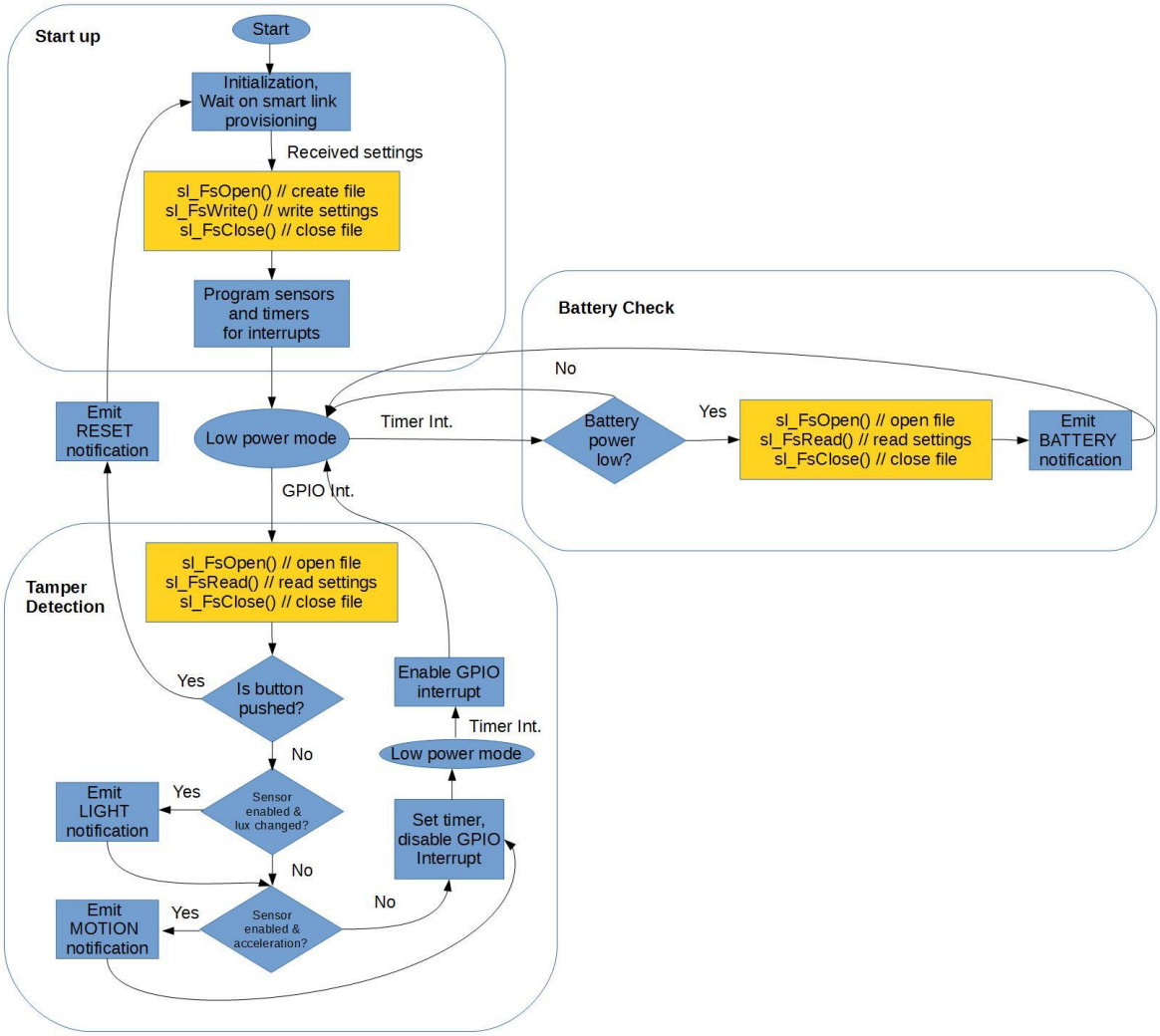

FIGURE 15: SOFTWARE FLOW CHART WITH FILE STORAGE

A default file name is used every time and the values stored in it are completely overwritten in the start-up phase of the program flow. Specifically, what needs to be written into the file and read upon waking up is the following data (the "< >" indicate some encoding of the semantic content between them).

<user-name> <user-password><detector-name> <GCM-Application-Instance-ID> <light-sensor-settings> <motion-sensor-settings>

Also of concern is the question of how does having this file relate to the possibility of running out of battery charge? A design decision had to be made as to whether or not in the case of running out of battery, if the user would have to reset the device using the mobile application, or if the detector could instead look in this file to recover the information it needs. To simplify the design, it was decided that it would be the case that

the user can reset the device rather than using the file. If the device runs out of battery the user must repeat the initial steps with the mobile application.

19.8 Interrupt Handling

Upon waking up from hibernation the software running on the CC3200 must determine what the cause of the waking interrupt was. In this design there is not an interrupt vector for each possible source. Instead, determining the source of the interrupt is done through a series of conditional statements which read the values on certain GPIO pins. For each sensor or the button, if it was the source of the interrupt, the GPIO pin will be a certain value as outlined below.

19.9 GPIO

Much of what needs to be done with the sensors is done through general purpose input output pins, or the GPIO. A falling/rising edge on a GPIO pin awakens the CC3200 from hibernation mode. A GPIO pin's value is checked to test whether or not the light sensor as potentially triggered the interrupt. Similarly, a GPIO pin's value is check to test whether or not the accelerometer as potentially triggered the interrupt. The following table indicates which pins are used for what external components and what their values indicate in terms of the operation of the CC3200's software.

TABLE 17: SOFTWARE GPIO USE

GPIO Number	Pin Number	Function	Values
2	57	Awakens the CC3200 from hibernation, checking value confirms that button is currently pressed	Falling edge causes interrupt, LOW indicates button is pushed
4	59	Awakens the CC3200 from hibernation, checking value confirms light sensor interrupt	Falling edge causes interrupt, LOW indicates light sensor interrupt
13	4	Awakens the CC3200 from hibernation, checking value confirms motion sensor interrupt	Rising edge causes interrupt, HIGH indicates motion sensor interrupt
8	63	Awakens the CC3200 from hibernation, checking value confirms state of charge interrupt from the power system	Falling edge causes interrupt, LOW indicates battery interrupt

Configuration of these pins is largely automated by using the source files generated by TI's Pin Mux tool. Within the tool you select the desired pins and their function. The tool generates a source file and a header file. The header file declares a single function PinMuxConfig() defined within the source file. It begins the necessary peripheral clocks and sets pin types and directions for each pin. Using these files, and a driver GPIO library

included with the CC32xx-SDK, largely abstracts away dealing with memory mapped IO, with certain exceptions. An example of an exception is that it is necessary for the programmer to be aware of GPIO's pin number relative to its base address. For instance, to read a pin the function MAP_GPIOPinRead(<Base>, <Pin>) is used. Examining the automatically generated file created by the pin mux tool can assist in finding these pin numbers as they are used to set the direction within it. [20]

To use the GPIO pins to come out of hibernation mode we must enable each GPIO pin as a wake up source and select whether the interrupt occurs on the rising or falling edge. For example, with the light sensor the interrupt pin is active low and it makes sense to select the interrupt as occurring with the falling edge. Before the interrupt is triggered, the GPIO pin attached to the sensor's interrupt pin will be HIGH. When the interrupt occurs the sensor's interrupt pin will become LOW causing the interrupt on the CC3200. At this point we read the GPIO 4 on the CC3200 and determine the light sensor caused an interrupt. [17]

19.10 Sensors

The detector software system has to utilize two different sensors. One is a light sensor, the TSL2561, which the CC3200 must set to trigger an interrupt given an appropriate change in ambient light. The other is an accelerometer, the BMA222, which the CC3200 must set to trigger an interrupt given that the detector begins moving or tapped hard enough. In addition to being able to set these devices to trigger the appropriate interrupts, the detector software should be able to turn off either sensor if desired to conserve power. Finally, after the interrupt is triggered the detector software must be able to determine which detector triggered the interrupt and then reset the detectors to allow for another interrupt in the case of there being a future tampering event. The registers and values that need to be written to configure these devices are included in the steps below.

19.10.1 The Light Sensor

The light sensor chosen for this design is the TSL2561. The CC3200 communicates with the TSL2561 through the i2c protocol. Communication over i2c is made relatively simple using the C32xx SDK where a few API calls provide most of the needed communication functionality. The address of the TSL2561 is assumed to 0x39 below which is the default address if the address pin is left floating. [1]

In the start-up phase of the program flow the detector obtains the sensor's settings and it must set up the sensor to provide interrupts before going into hibernation mode. The light sensor's settings consist of a factor by which the light amount can change from the pre-hibernation amount before causing an interrupt, or alternatively, a flag indicating that the sensor is disabled. If disabled, then the sensor is put to sleep, otherwise the steps to programming the interrupt function on the TSL2561 are given below. [1]

1. Turn the sensor on. Here we desire to minimize power consumption so the integration time of 402 ms is chosen. To do this write to register 0xa0 the value 0x02.

2. Disable the interrupt by writing into register 0xa6 the byte 0x00.

3. Obtain the real time light sensing data. To do this read from register 0xac 2 bytes. This returns two bytes, the first of which is the lower order 8 bits of the reading and the second of which is the higher order 8 bits of the reading. These two bytes can be stored in a single variable.

4. Compute the threshold values. The user gets to adjust the sensitivity to light and does so by specifying the factor by which the reading is changed from the current value (in absolute value). Initially, factors less than 1 were considered. However experimentation with the sensor revealed that the values produced by the samples varied drastically more than expected. In the typical application where the light sensor would be used, the detector is placed someplace dark such as a safe or a box. The light sensor should cause interrupt when the box is open. Below is the output of values from the sensor on one such experiment. The experiment was performed by placing the sensor inside a shoebox and slowly opening the box while a program continuously prints the sample values.

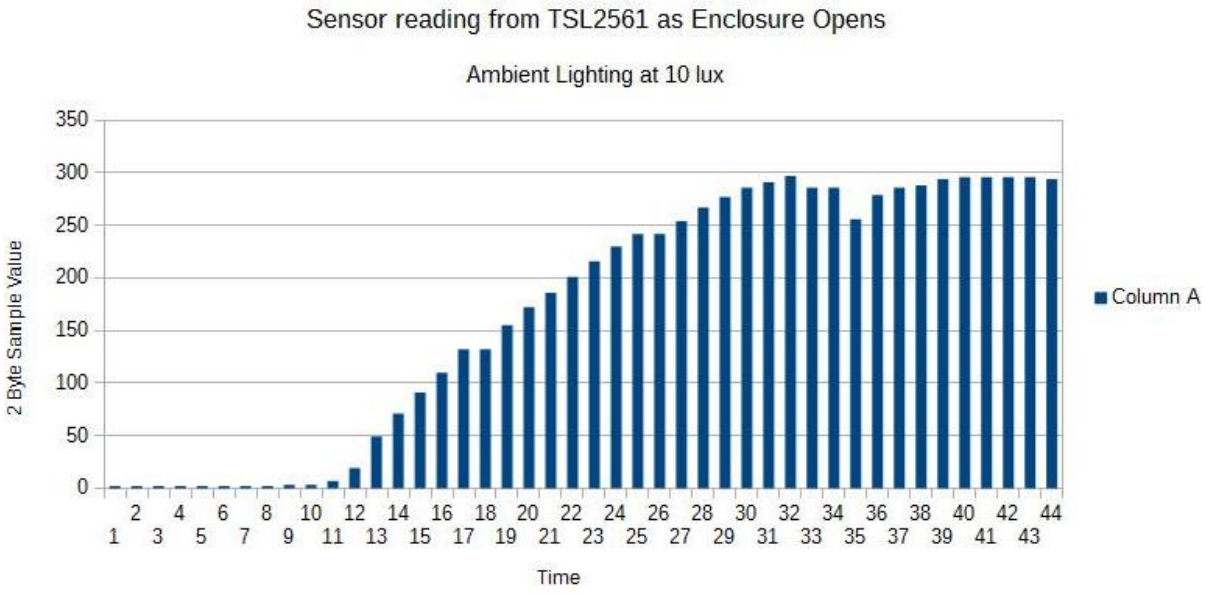

FIGURE 16: LIGHT SENSOR READING CHART

As can be seen, factors less than one are not appropriate. Instead, factors which are powers of 2 are used. The user can select between a factor of 2, 4, or 16. The following C code finds the threshold values accordingly.

1. int reading = (higherByte << 8) | lowerByte);

2. int difference = reading << factor;

3. int lowerThresh = reading – difference;
4. if(lowerThresh < 0) lowerThresh = 0;
5. int upperThresh = reading + difference;
6. if(upperThresh > 0xffff) upperThresh = 0xffff;

7. int lowerThreshLowerByte = lowerThresh & 0xff;
8. int lowerThreshUpperByte = (lowerThresh & 0xff00) >> 8;
9. int upperThreshLowerByte = upperThresh & 0xff;
10. int upperThreshUpperByte = (upperThresh & 0xff00) >> 8;

The conditionals which check if the lower threshold is negative or if the upper threshold too large can be expressed in to bytes work by saturating the result if this happens. The upper threshold is forced to its maximum value if it's over (line 6), and the lower threshold is forced to 0 if it's under (line 4).

5. Set the threshold values in the TSL2561 so that it can trigger an interrupt. This is done by writing the calculated upper and lower bytes of the two thresholds above into the appropriate registers. In particular, this is done by writing to register 0xa2 two bytes <lowerThreshLowerByte> and <lowerThreshUpperByte>. Similarly, for the upper threshold two bytes need to be written into register 0xa4.

6. Set the interrupt control to enable the interrupt and choose how the interrupt will occur. We have selected to use level interrupts for this project, as it corresponds to the process of checking the GPIO pins after the interrupt has occurred. Also in this step we set how many integration cycles are required before triggering the interrupt. To make is so the detector senses tampering after about a second of light over the threshold, 3 integration cycles are used giving us .402 * 3 = 1.206 seconds. Write to register 0xa6 the byte 0x13.

The light sensor is then set to trigger an interrupt to wake the CC32xx up from hibernation mode. Alternatively, the user may have selected to turn light detection off on the detector. This is done as follows

1. Set the TS2561 to sleep by writing to register 0xa0 the value 0x00

In the tamper detection phase of the program flow the detector must check to see if the TSL2561 has triggered an interrupt, causing the CC3200 to wake up. It does this by checking GPIO pin 4, but only if it has that sensor enabled. The CC3200 must then wait an appropriate amount of time so that duplicate tamper notifications are not sent. Finally, it must reset the TSL2561 to allow for future interrupts to be triggered. The details of how this are implemented are given below.

Upon awakening from hibernation

1. If light sensor enabled, check GPIO pin 4 to see if the TSL2561 has triggered an interrupt. If so, then send notification, else move on to motion sensor.
2. Perform actions related to other sensor interrupts.

3. After waiting a period of time, reset interrupts on sensor and re-enable GPIO interrupts on the CC3200. To reset the interrupt of the TSL2561, write to register 0xf0 the byte 0x03. What is proposed below is that upon detecting a light interrupt, slightly more than this should be done.

It makes sense to repeat the initial light sensor setup steps in the event of a light interrupt. The reason is that if the detector as moved into a different environment with different ambient lighting, the behavior of the detector should be to adjust its self to this new lighting. If the lighting changes from the new range, then it's appropriate for the detector to send a new tamper notification. This behavior makes the most intuitive sense from the user's perspective versus sending repeated notifications while the ambient lighting has not changed since the last notification. Implementing it involves making one additional functional call.

19.10.2 The Motion Sensor

The accelerometer, the BMA222, allows for two different modes of operation, general mode for control by a microcontroller, and dedicated mode for standalone operation. The first is appropriate for our purposes because it is being controlled by a CC3200. This allows for the CC3200 to perform the necessary set up through the i2c bus.

In low power mode the BMA222 switches between sleeping and waking-up. During wake up, the needed samples are acquired and this triggers an interrupt if required. It stays awake while the interrupt endures, the duration of which can be programmed. How often the BMA222 wakes up directly effects the amount of current that the device is consuming. Waking up one time a second consumes an average current of .7 uA while waking up every 10 ms pulls 16.4 uA, and waking up every 4 ms 34.5 uA.

The default acceleration range of +-2g is appropriate, since we are interested in low levels of acceleration. Slope detection detects any changes in motion. If the slope between successive acceleration samples is greater than the threshold, an interrupt is triggered. We use this on all three axis. The lower threshold of this value is selected by the user in the initial set up phase. Although by the BMA222's design, the sensor is supposed to have the program select this sensitivity value with the threshold value, it was discovered through experimentation that the sleep duration has more of an impact on the sensitivity than the threshold value. As a result the user's selection of sensitivity adjusts the wakeup time of the detector, which in turn effects the amount of current consumed. This makes some intuitive sense. A less sensitive tamper detection system can run without charging for longer.

Orientation interrupt is also enabled to detect if the detector has changed orientation with respect to the gravitational field. When the motion detection is enable for the detector this is always on. This interrupt is triggered whenever the value of the orientation has changed for the device. It does not have a user selectable threshold.

Setting up the motion sensor for operation consists of writing to the registers of the device using i2c. Inspection of the manual reveals which registers must be written to obtain the desired functionality. Set up steps are listed below.

--leave default values for electrical characteristics open-drain and active level 0 for INT1 pin

1. Route all interrupts to INT1 pin. This is done by writing to register 0x19 the byte 0xf7.

2. Reset interrupts and set interrupt latch to 2 seconds. Write to register 0x21 0x84.

3. Put BMA222 into low power mode with a sleep duration. Write to register 0x11
   a. Choose for low sensitivity the value 0x5e (sleep 1 s between samples).
   b. Choose for medium sensitivity the value 0x54 (sleep 10 ms between samples)
   c. Choose for high sensitivity 0x50 (sleep 4ms between samples)

4. Set the sensitivity for slope sets sensitivity to motion of the sensor. Write to register 0x28 0x03. This value was found through experimentation.

5. Enable the orientation interrupt and the slope interrupts for the three axis. Write to register 0x16 0x47. [2]

After these steps have been completed the BMA222 is ready to trigger an interrupt on the CC3200 in the event of motion. Note that this interrupt pin value resets itself after its programmed duration as opposed to the light sensor which needs to be written to be reset.

The settings for the motion sensitivity were found empirically. More specifically, we tried to pick it up and move it while adjusting the values until we found some which seem to work well for our needs.

19.11 Battery Monitoring

The IC chosen for battery monitoring was the bq27510-G3 Fuel Gauge. It can be programmed at startup to provide an interrupt in the case of a low battery, but it was decided it would be a more flexible scheme to have the CC3200 wake up periodically from hibernation and poll the Fuel Gauge. There was one major motivating factor behind this decision. Choosing to poll the Fuel Gauge periodically gives the option to notify the user of the battery status at various points before it needs recharging rather than just one. These points in the charge, could potentially be made user selectable, although not currently implemented as such.

Reading the value from the Fuel Gauge is done through i2c. We read the values from registers 0x20 and 0x21 which provides the remaining battery capacity as a percentage, from 0 to 100. Upon reading the capacity we have the choice of whether to send a notification to the mobile application or not. As a default configuration, it was decided that we will only send a notification in the advent of the charge being less than 10 percent and that the notification will include the current charge value. [4]

How often to wake up the CC3200 to perform this check was another interesting issue. The CC3200 allows you to set the number of 32.768 KHz clock ticks that pass before coming

out of hibernation. It seems appropriate to check the battery twice an hour or so. That gives us:

(2 check / 1 hour) x (1 hour / 60^2 sec) x (1 sec / 32,768 cycles) = 1 check / 58982400 cycles

This value easily fits inside the long long data type that the corresponding function to set the interval takes.

When we are done reading the value from the Fuel Gauge we can put it into HIBERNATE mode. This puts the Fuel Gauge in a low power state to conserve battery. To put the Fuel Gauge into HIBERNATE mode we must write the appropriate bits into the command register. To bring the Fuel Gauge back into normal mode we need only begin communicating with it through the i2c bus and it will automatically return to normal mode.

A design question was whether or not we should send a notification every half hour after hitting the critical percentage of battery remaining. It was decided that this would be annoying to users, and that only one notification would be sent out. Since hibernation fails to retain RAM, one option to accommodate only sending one such notification additional data has to be added to the file which is used to store the other persistent data. The additional data indicates whether or not we should send any future battery notifications. The other option is simply to put the CC3200 into hibernation indefinitely whilst alerting the user to this fact.

20 Web Service

20.1 Overview

The role of the web service is to ensure that the user has a unique user name and a password. In addition to this, the web service can log the tampering notifications sent out by the detector into a database. Although these records are not directly used in the current design, it is easy to imagine an expansion to the design were these records would be of value.

There are a number of options available for where to run this application. Amongst the plethora of options are Microsoft's Azure, Amazon Web Services, and Google Cloud Services. It was decided that Google Cloud Services would be the easiest. In fact, the web service itself can be further broken down into two components, a SQL database and a python application which acts as an http server and connector to the database. Both of these run on Google Cloud Services. The database runs in a separate virtual machine than the python web service.

20.2 Data Base

The database consists of two tables. The first table, called user, is a table of usernames and hashed passwords. The other table, tampering, is a database of tampering notifications sent out by detectors. It stores inside of it which user the tamper notification was sent out for, which detector sent it out, what the message content of the notification was, and the time and date of the notification. The two schema are given below.

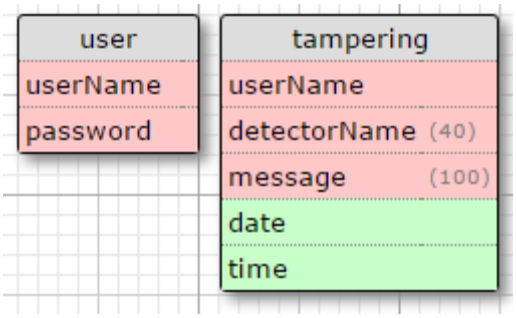

FIGURE 17: DATABASE SCHEMA

20.3 Account Creation

Account creation works as follows. The user name and password are chosen by the user in the mobile application. The user name and password get sent in the form of an http POST request to the python application running on Google Cloud Services. Note, as described later, this is not nearly as insecure as it seems because we use http over TSL (https). The encoding is done using JSON. The JSON encoding is

{

“username”: “<user-name>”,

“password”: ”<password>”

}

The decoding is done in Python using request.get_json() method, provided by the flask library, which parses and returns incoming JSON data from the POST request. [21]

The python application queries the database to see if a record exists of this username in the username table. The SQL for this query is

SELECT EXISTS(SELECT 1 FROM user WHERE username = ‘<user-selected-name>’ LIMIT 1);

Using the Python library MySQLdb we can connect to the database and execute our prepared SQL command. This is done by first executing the MySQLdb.connect() function that returns a database object. We get a cursor to the data base with db.cursor(). The SQL command above is stored in a variable, let’s call it sql, as a string and using the cursor we try to execute the command with cursor.execute(sql). We roll back if there is an exception. This is the general procedure used by the web service for using the database.

If the user name already exists in the table the response to the http POST is a message indicating the user must select a different username. The mobile application is programmed to respond accordingly. If the query indicates that this user name is still available, then the web service hashes the user provided password, and stores the username and hashed password as a new record in the table. It then reports the success to the mobile application.

The hashing is done in Python using the Passlib Library. This is done very easily with the following function call [22]

hash = pwd_context.encrypt(<user-password>);

It's this hashed value which gets stored in the user table, not the password itself; storing the password itself is bad practice. The hashed output can be compared with the user entered password at any later time very easily as well.

ok = pwd_context.verify(<user-password>,hash);

After hashing, we store the user name and password into the user table. This is done using the MySQLdb library, once again using the same general procedure outline above. The SQL passed to the execute command of the cursor is

INSERT INTO users (username, password) VALUES ('<user-name>', '<hash>');

20.4 Logging

The detector, in the event of tampering, wakes up and sends an http POST to Google Cloud Messaging. It can also be made to send an http POST to the web service, which can then insert this into the tampering table of the database. Like the rest of the http POSTs made in the operation of the detector the information is encoded in the JSON format. Asample body of one of these http POSTs is the following.

```
{
  "name" : "JohnSmith",
  "password" : "1234567",
  "detector" : "BedRoom",
  "message" : "light detected"
}
```

The request is made to URL corresponding to the web service by the detector. When received, the web service hashes the password and compares it to the one stored in the users table to see whether it is valid. If it is valid, then it logged into the tampering table using a SQL command similar to the one used in the account creation section.

21 Mobile Application

21.1 Overview

For this project it was decided that we would only develop and application for the android platform. The mobile application has two main functions. The first is provisioning. The mobile application allows the detector to join the wireless network. The second is as a Google Cloud Messaging Client. The mobile application must allow the user to receive

notifications from the detector. In fact, the order of these steps is reversed in terms of the initial setup. The application should know its instance ID for Google Cloud Messaging before performing the provisioning and settings transfer step.

The goal of the application is that, following the initial setup, the user interacts with the application very little. Rather, the application runs in the background and receives notifications from the detector. Note that after the initial setup, there is no way for the user to transmit additional information to the detector. It can only receive information in the form of notifications. Any changes that the user may want to make to the system involve physically pushing the button on the detector and restarting the initial setup. This feature, although ostensibly inconvenient, provides a layer of security since no person could remotely modify the settings on the detector.

21.2 Notifications

The Google Cloud Messaging, GCM, functionality of the application is based off of examples provided by Google. The steps performed upon first opening the application are to set up the application to receive notifications. We first get an instance ID from Google Cloud Messaging. This only needs to happen the first time the app starts. Google makes this available with the object/method instanceID.getToken(). This token is what gets sent over to the detector through mDNS. [23]

To detect that this is the first time the app is launched, the shared preferences for the app are read every time on startup, with a Boolean variable written to only on the first occasion. After the application receives its instance ID it must store this in a file. It must also communicate it to the detector in mDNS step, since the detector acts as the app server for GCM issuing downstream messages.

Receiving the notifications is handled by an object extending GcmListenerService. The method onMessageReceived() is overwritten to provide the desired functionality upon receiving the notification.

21.3 Provisioning and Settings Transfer

The mobile application must get the detector to connect to the wireless network and then transmit the information the detector needs to function. The first step is provisioning, the second step is settings transfer.

The provisioning portion of the mobile application is largely inspired by the open source code from TI which provides with this functionality. The major steps taken for this part of the program are the following. [24]

1. Get SSID and password from text fields of the user interface
2. Get gateway from NetworkUtil object (defined in TI provided smartconfig.utils package)
3. Create a new SmartConfig object (define in TI provided smartconfig.utils package)
4. Run method smartConfig.transmitSettings();
5. Use mDNS to identify the new device

Using TI's helper objects greatly simplifies the implementation of this part of the application.

The detector at this point is connected to the network. As mentioned earlier, before provisioning, the application receives it unique instance ID from GCM. We must also send the instance ID for the messaging along with the user selection of sensor settings which can be obtained from the user interfaces' text fields. This is done by putting the application into mDNS advertiser mode with a predefined name for the service which the detector knows. Upon registering for the service the data can be set over as the text of the advertiser's service to the detector which is listening for the service.

This service name must not already exist on the network to be successfully registered. This point is subtle, but important. If the mobile application failed to unregister this service with the network upon completing this step, it would be impossible for more detectors to be added to the system with different sensor settings. In fact, they would find the service from a previous initiation step, most likely with the correct instance ID but with the wrong settings. Therefore, care must be taken to avoid this by having the application unregister this service once the setup is complete. [25]

Once these steps are completed the user can simply background or exit out of the application. They will receive notifications if their detector is tampered with.

21.4 User Interface

The goal of the user interface is to be easy and simple. Two text fields let the user enter the network information and two sliders let the user select their sensor settings. To connect to the detector the user hits the connect button. A progress bar indicates how much longer the application will search for a detector before a message box pops up indicating "No Detectors Found!" If the application finds the detector before the allotted time a message box pops up indicating the "Detector is Ready!"

Below is an example of what the settings page could look like for the mobile application.

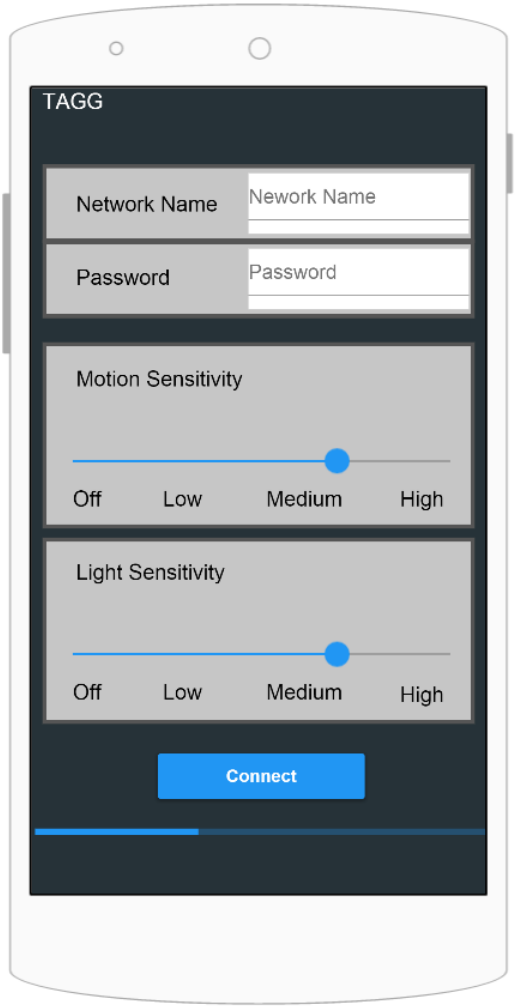

FIGURE 18: UI ON MOBILE DEVICE

22 Software System Testing and Mile Stones

The following tests indicate which steps should be taken in order to ensure the software system is fully functional. They indicate both the development sequence and which tests can be performed along the way to ensure the proper functionality of that part of the system.

22.1 Detector

This table indicates which mile stones should be achieved for the detector and how to test that they are accomplished.

TABLE 18: DETECTOR MILE STONES AND TESTS

Test	Description	Expected Results
CC3200 dev-board Provisioning	Use TI SmartConfig to connect to WI-FI network using TI test app.	Able to connect to at least two different wireless networks, results seen terminal.
CC3200 dev-board mDNS	Use laptop as mDNS advertiser to transfer setting to CC3200 dev-board.	CC3200 dev-board receives settings, parse settings string, results seen on terminal.
CC3200 dev-board HTTP POST to GCM	Post JSON format data in HTTP POST to GCM. Includes group notification information.	Notification received on GCM android test app.
CC3200 dev-board vibration (tap) interrupt	Use vibration causing GPIO pin interrupt to bring CC3200 out of sleep to send notification.	Notification received on GCM android test app indicating GPIO interrupt. Should work for any vibration.
CC3200 dev-board lux meter interrupt	The lux board sends interrupt to GPIO pin bring CC3200 out sleep to send notification.	Notification received on GCM android test app indicating GPIO interrupt. Should work for hard coded sensor values.
CC3200 dev-board lux meter settings transfer and interrupt	Modify the mDNS code to transfer lux meter settings. The lux board will send interrupt to GPIO.	Notification received. Should work for values specifically sent during mDNS.
CC3200 dev-board lux meter reading	Record a reading of the lux meter value before going to sleep, read the lux meter on wake up.	Upon wake up, print to terminal values from lux meter recorded before and after wake up.
CC3200 dev-board accelerometer reading	Check for accelerometer communication.	Print accelerometer value to the terminal.
CC3200 dev-board battery sensor reading	Read battery sensor.	Print battery sensor value to the terminal
CC3200 dev-board timer interrupts	Test that CC3200 awakes from timer interrupt prints to terminal and goes back to sleep.	Terminal output confirms CC3200 awakes as expected. Sleep should be programmable.

Test	Description	Expected Results
CC3200 complete program flow	Integrate elements above into program flow given in microcontroller overview. Test all transitions (interrupts and programmed) given in program flow.	Using printing to terminal to indicate program state. Should match flow chart.

22.2 Mobile App

This table indicates which mile stones should be achieved for the mobile application and how to test that they are accomplished.

TABLE 19: MOBILE APPLICATION MILE STONES AND TESTS

Test	Description	Expected Results
Android App GCM	Android App receives notifications using the proper notification group. Can be hard coded into CC3200 for testing.	Notifications appear only when proper notification group is used.
Android App SmartConfig provisioning	Integrate TI SmartConfig app code into T.A.G.G. android app.	CC3200 dev-board is able to connect to WI-FI through android application.
Android App mDNS	Android App works as mDNS advertiser and transmits user input sensor settings and notification group.	CC3200 dev-board is able to receive information and settings and print to the terminal.
Android App connect to web service	Android App connects to web service sending fake user name and password.	Response from web service indicating username and password are available.

22.3 Web Service

This table indicates which mile stones should be achieved for the web service and how to test that they are accomplished.

TABLE 20: WEB SERVICE MILE STONES AND TESTS

Test	Description	Expected Results
Web service responses to request for user name.	Android app or desktop computer sends request for user name.	Web service says name is available only if it has not been stored in DB yet.
Web service adds user name and password to DB	Android app or desktop computer sends request for user name. Name gets added to DB.	Web service responds first time saying name is has been added. Web service responds to duplicate request saying name is already used.
Web service deletes user name and password from DB	Android app or desktop computer sends request to delete user name. Name gets deleted from DB.	Web service responds to request to delete non-existent name with error, request to delete existent name with deletion success message.
Logging Tampering	Through HTTP post, the detector sends tampering information to web service which then logs it in DB.	After executing, or simulating several tampering events

23 Programming the CC3200

The program running on the CC3200 is stored on the serial flash. Uniflash is a software to accomplish writing the program into flash using UART. On a hardware level this involves using pin 32 to reset the device when prompted, pins 55 (UART1 TX) and 57 (UART1) are used to transmit the data, and pin 21(SOP2) is pulled up during the reset. On a software level this involves building the project which generates a binary, then setting that bin as the system file /sys/mcuimg.bin in Uniflash. Programming is then done by selecting the options erase, update, and verify in Uniflash. [26]

The file system on the CC3200 is proprietary and cannot be directly manipulated by the programmer. During boot the M4 processor performs its own initialization and then begins executing the user code. These features cannot be changed, so flashing the SFlash such that the user's code is in the file /sys/mcuimg.bin is the only correct option for programming the T.A.G.G software system onto the CC3200. [26]

This, by itself, is not secure. Potentially, a malicious user could reprogram the CC3200 with the appropriate hardware. The only insurance that the TAGG system can offer is that once the detector is set, if the sensor settings are configured properly and network connectivity is present, the user will receive a tamper notification before the detector could be reprogrammed.

24 Encryption and Security

The plaintext used for the notification can be encrypted on the detector and decrypted on the mobile application and web service upon receiving the notification. This would make it so that any intermediary party receiving the POST or the notification would be unable to receive information about which detector or type of tampering was detected. This is most easily done for use by using http with TSL, also known as https.

In order to use this the detector must have the current date in order to validate the certificate used in https. A good solution to this problem is to use a Simple Network Time Protocol Server to obtain the current time. This is a better solution than having the detector maintain its own time with the underlying assumption that the detector will live somewhere where a network connection is available. It performs the task of acquiring the current date and time information as part of performing the http POST requests.

There are still vulnerabilities to the T.A.G.G. system. For one thing, we have decided to not use encryption during the provisioning step. This could be done to prevent someone from intercepting the passkey for the wireless network, assuming they didn't possess an encryption key used. However, encrypting this seems a little meaningless if the encryption key is stored statically inside the mobile application and inside the CC3200. This would make it so the first person to obtain the encryption key from the software would make it completely insecure. It's difficult to envision a different solution to this problem that doesn't involve a physical connection, or placing the detector in access point mode, because prior to the provisioning step, the detector has no network connectivity.

An even more egregious vulnerability is in the settings transfer step. Using the mDNS text to advertise the username and password makes those available to all currently on the network. Although the mobile application deregisters this service as soon the detector responds notifying it has obtained the information, there is certainly a window in which this information is open to anyone on the network. Any more complicated solution might obfuscate this information, but it would face fundamentally the same issue. This is that the detector at this point has no way of certifying that it is the only entity that should receive this information. Other solutions might force an attacker to buy their own detector, and force them to perform their own provisioning step within a short time window, close to that of their victim, but ultimately we discovered no good way discern the detectors at this point in the process.

As such, we are forced to admit that during the initial setup the current design is locally insecure. An attacker currently on the Wi-Fi network could potentially receive the user's user name and password during the startup phase of the programs execution. Following this step in the process the T.A.G.G. will protect the user's information and message content using https and will not reveal any other sensitive information either through mDNS or TI's provisioning technology.

24.1 Limitations

It is important to consider what the T.A.A.G. system can do and can't do in terms of security for the user. One goal of the CC3200 is to send a notification out before any person

could disable the detector. It is impossible to ensure that following this initial tampering notification that the detector can continue to function. However, even with the sensor settings selected properly it is still possible for the T.A.A.G system to fail to send this initial notification. The most obvious way that this can happen is if there is no network connectivity for the detector. A person looking to get around the T.A.A.G system could do so by performing the following steps.

1. Turn off the network
2. Perform the tampering
3. Destroy, or worst yet (subtly disable), the detector
4. Turn back on the network

There is a solution to an attacker performing steps 1, 2, and 4, but not 3. The CC3200 could potentially store in a file that there was a tampering and periodically wake up, attempt to reconnect to the network, and attempt to send the notification as soon as possible. Presumably, the wireless network will eventually be restored and this notification will be sent out. This is not implemented in this project, but easily could be.

We saw no solution to detecting an attacker performing all four of these steps. It is important that, at very least, the detectors cannot be physically modified in a way that will disable them but not be noticeable. This is outside of the scope of the software design but is important when considering how the detector is encased.

The system works best when the would-be attacker is unaware of the TAGG system being in place. If the attacker lacks the foresight to disable the network, then the user will receive a notification and be aware of the tampering.

25 Battery Life

Three scenarios are presented below along with their expected amounts of time before the battery runs out of charge. These times are used in the initial prototype testing section to determine the best way to test the battery life in a given scenario, i.e., whether to actually test the battery life by letting run out or not.

The time that the CC3200 spends in active mode while sending a notification is dominated by time spent waiting on a network. It is 42 ms DNS lookup time for a typically well cached url, such as Google's. We add to that the time to do two http POST requests, which are typically slightly under 500 ms each. That is roughly 1 second in active mode to send a notifications and logging. This can vary of course, depending on what is cached and network conditions. [27]

We simplify the calculations by saying that the CC3200 spends that entire one second in transmit mode which pulls 272 mA according to TI's documentation. This is an overestimate since only a small portion of this time will actually be spent transmitting. [18]

The first of the three scenarios is what we expect to be the most common one.

Normal operation: This circumstance is described by saying there is a network connection, and notifications are sent off infrequently. We can define infrequently by saying notifications are set out twice a day. The expected battery life is calculated by using the 1 second in active mode to send a notification. There are 60^2 * 24 seconds every day, giving us an active time ratio of (2 / (60^2 * 24)).

Average MCU current [18]

= (2 / (60^2 * 24)) * 272 mA + ((60^2 * 24 – 2) / (60^2 * 24)) *.004 mA

= .01 mA

Average Gas Gauge current [4]

= (2 / (60^2 * 24)) * .103 mA + ((60^2 * 24 – 2) / (60^2 * 24)) *.004 mA

= .004 mA

BMA222 current (upper bound based on designed usage described in motion sensor section) [2]

= .0345 mA

TSL2562 current [1] = .24 mA

Total average current = .289 mA

1200 mAh / .289 = 4152 hours = 173 days

This is a reasonable amount of time. It is well within our design goals.

Stuck at startup: The downfall of making the code simple on the detector is that there are certain circumstance which the program will not handle gracefully. This is one of them. If the detector is turned on, and the user does not then use the mobile application to perform the setup step, the detector will not go into hibernation mode. Instead, it stays in active mode waiting for TI's provisioning to start. The expected battery life is significantly less if this is the case. The gas gauge and the sensor are assumed to start in active mode. The documentation for these values is the same sources as above.

MCU current in RX = 53 mA

Gas gauge current = .103 mA

Sensor Current = .2745 mA

Total current = 53.38 mA

1200mAh / 53.27 mA = 22.48 hours

Too many notifications: Provisioning is successful. Now we are interested in the possibility that the sensor is sending notifications as much as possible. Because the program running

on the detector forces a wait time between notifications, there is maximum number of notifications that can be sent in any hour given by (1 hour) / (forced wait time between notifications). Assume that we demand the detector wait at least 59 seconds between notifications. The time it spends active for the notification is 1 second as calculated above. This gives us a maximum notification rate of 1 per minute or 60 per hour. Also, we will spend roughly 1 minute every hour in active mode.

Average MCU current = (1 / 60) * 272 + (59 / 60) *.004

= 4.537 mA

Average Gas Gauge current = (1/60) * .103 mA + (59/60) *.004 mA = .00565 mA

Sensor Currents = .2745 mA

Total current = 4.8115 mA

1200mAh / 4.8115 mA = 249.1 hours = 10.4 days

The disparity between how long the battery will last isn't all that surprising considering how different the program flow will be between these three cases. A clever marketer would certainly advertise the 173 day battery life.

26 Initial Project Prototyping and Testing

The initial prototype must be tested to ensure that the design concept works as expected before proceeding to create the final design. The initial prototype will consist of a development board powered by a 1200 mAh battery with fully functioning software, a combination of mobile applications which can simulate the final mobile application, and fully functional web service and data bases running in Google Cloud Services.

Account Creation and Deletion Test: The mobile application should be able to create and delete user accounts. The web service also needs to be able to identify when a user name is already in use and respond to the mobile application by letting it know about this fact. The mobile application should also be able to delete an entry from the user table given the password and user name. The status of the database can be checked by viewing the database through the mysql client application. Importantly, notifications in the future tests should not be received by the user unless their detector possesses a valid user name and password.

The test is to create a username and check that it is in the user table. Then ensure notifications can be received. Delete that user name and check that it is no longer in the table. Then ensure notifications are no longer received.

Provisioning Test: Using at least two different Wi-Fi networks (using WPA2 security) confirm that the detector is able to connect to the local access point and receive an IP address. This can be done using TI's Simple Link mobile application if that functionality is not added to the T.A.G.G. mobile application at the time of the testing. Confirmation of the connectivity can be done through a notification or an http post to the web service.

Notification Test: The text included in all of the notifications needs to be checked by viewing it on the mobile device. This can be done creating the user account, selecting the settings on the detector, and then causing each type of notification to be sent. The settings on the detector can be set to high light sensitivity and high motion sensitivity. The connection notification will be sent out after the provisioning and settings transfer step. Then the device can be moved abruptly and quickly brought from the dark into the slight. The expected behavior is the motion notification and the light notification for their respective behaviors. For the battery notification, we could wait until the battery is drained to ensure the notification is working properly. Unfortunately, this may be rather time consuming as the detector is designed to preserve the battery's life for as long as possible. We will cheat here and force the notification to be sent by modifying the detectors program.

Logging Test: Every notification should cause a log to be entered in the tampering table of the database. The same procedure used in the notification test can be used, but instead of checking for the notifications on the mobile device we can use mysql client to check the contents of the table and ensure that the entries are in fact being inserted into the table.

Motion Sensitivity Test: This test gets at the heart of the desired functionality of the T.A.G.G. system. To test that the detector can successfully notice when some moves the object is to be tested by actually placing the detector on various objects and checking that notifications are sent when the object is moved. To do this, some sort of adhesive pad will need to be placed on the development board to allow it to be attached onto various objects.

The motion sensitivity tests outlined below should be performed by at least three people, with each test performed at least three times. The reason that these tests should be performed by more than one person is to achieve some sense of what the average person's effect on the detector while moving one of these objects would be, rather than an individual person's unique way of handling the object.

Slightly different results are expected for the various motion sensitivity settings. Below are several items to place the detector on and the results we expect for the different settings.

A front door: The test will be performed by attaching the detector to corner of a domestic front door. In particular, it should be attached to the inside, opposite the hinges. With high sensitivity, a door knock or heavy banging on door should be sufficient to cause a notification. For medium sensitivity banging should not be sufficient to cause a notification, but opening the door should. With low sensitivity, very carefully opening the door may not cause the notification to be sent, but carelessly opening the door quickly will.

A safe door: Same behavior as above. Making sure the behavior matches on both will be important since the actual motion of the objects may vary significantly.

Inside of a dresser drawer: Attaching the detector to the inside of the drawer we expect following behavior. With high sensitivity any sort of jostling of the dresser will cause a motion notification. For medium sensitivity, opening the drawer even slowly will cause a motion notification. For low sensitivity opening the drawer deliberately slowly may not cause a motion notification, but opening it in a normal fashion will.

Side of a cardboard box: With high sensitivity, tapping on the box should be enough to trigger a motion notification. With medium sensitivity, picking up the box, no matter how slow, should cause a notification. With low sensitivity, picking up the box deliberately slowly will not cause a notification, but picking up the way it in the way it normally would be will cause a motion notification.

Detector laying on surface of a table: With high sensitivity, when someone uses the table for writing there should be motion notification. With medium sensitivity, someone will actually have to move the detector to cause a motion notification. With low sensitivity, someone should be capable of very carefully moving the detector and not causing a motion notification, but careless motion will cause a notification.

Light Sensitivity Test: These tests can be perform by one person as they are not dependent on the way in which the tester behaves, but rather the detectors surrounding environment.

A dark closet: Place the detector inside a dark closet and open the door, while outside there is normal indoor ambient lighting. The expected behavior is that for all sensitivity settings (except off) the detector should send out a light notification.

The inside of a safe: Similar to the test above. For all sensitivity settings we should expect a light notification when the safe door is opened into normal indoor ambient lighting.

In front of a television: The idea of this test is that the user would like to be notified if someone turns on their television. The expected behavior of the detector is as follows. For the high and medium sensitivity settings there may be false notifications due to changes in ambient lighting. For the low sensitivity setting it should only send a notification when the television is turned on. To actually perform the test, the detector should be placed directly in front of the TV's screen all day with the lowest sensitivity setting. Leave it there for 24 hours and check that no light notifications get sent out. Following this, turn on the TV and confirm that indeed a notification does get sent out when the TV is turned on.

Underneath a pill bottle: With this test we place a pill bottle directly on top the light sensor on the detector. With high sensitivity changes in the ambient light not involving the removal of the pill bottle may cause a light notification. With low sensitivity the detector may not send a notification when the pill case is removed. It is with medium sensitivity we expect the detector to successfully determine when the pill bottle has been removed and send a light notification.

Battery Life: These test are designed to see how long the detector can run off of its battery in various circumstances. It is assumed that all of the tests below start with a fully charged battery. It should be expected that the battery performance using the development board will differ slightly from the battery performance in the final design.

Normal operation (2 notifications per day): This can be tested by inserting into the detector's program a timer which will awaken the detector twice a day, poll the gas gauge, and send the gas gauges reading in the form a notification to the mobile application. From this we can extrapolate the overall battery life without necessarily allowing the battery to complete drain. This is advantageous, since the expected battery life of 173 days.

Stuck at startup: In this scenario the detector never goes into hibernation mode causing the battery to die relatively quickly. To test this we can't use notifications, so instead we actually run it and time how long it takes for the battery to die. We expect the battery to be out of charge in 22.48 hours

Too many notifications (1 notification every minute): The calculation is the battery life section shows that this will take 10.4 days. This is will take too long to test by allowing the battery drain. We can modify the software on the detector to send a notification every minute with the message as the current battery reading from the gas gauge. From this data we can extrapolate the battery life after a brief period of time.

Off to On / Dead Battery Test: We would like to ensure that if the detector is powered off and then turned back on that it starts back up as though it was its first time being turned on. This means that the Wi-Fi profiles on the detector are cleared and it begins the provisioning step again. This is easy enough to test, the detector can be setup once and then restarted. The expected result is that the detector behaves in the way that it did when powered on for the first time. A dead battery is equivalent to cutting the power completely so this should adequately test what will happen when the battery dies as well.

27 Final Prototype Assembly

This section discusses the process and methodology of piecing the final prototype together. The system in general contains numerous sub-systems, these subsystems can be looked at and implemented physically as an electrical circuit. All of these electrical circuits must at the end be strategically placed together in one all-encompassing circuit. The way this must be done is by using a printed circuit board (PCB), this PCB will be the housing unit containing all of the needed circuitry and having all the necessary components soldered on to it.

27.1 PCB Design Software

When putting all of the system hardware circuits together on the discrete PCB there must be careful design and consideration in place, many design rules and regulations, and avoid any mistakes in the routing. The reason for this that this PCB design directly translates to the physical real-life implementation of the system, hence it takes us far away from the idealness of any type of simulation and unit testing. On a PCB we may have interference of different parts on each other, and this may cause noise and distortion. For this reason a PCB designer must be very careful indeed when it comes to this matter.

The design of the PCB in our system first started off from the circuit diagram level. After the actual circuit was known and all of the connections were made on the schematic level then the actual physical PCB layout and routing was then designed. Going through this process we needed the assistance of some PCB software below are the list of the software that was used:

27.1.1 OrCAD PCB Editor and Allegro

Allegro PCB in a well-known tool for PCB routing and development. The tool is quite powerful and enables detailed editing with a very powerful auto-routing algorithm. This tool is known to be a more industrial one meaning that the freeware that they offer, which they like to call “demo-mode” has very limited capabilities and does not allow you to do much editing at all.

During the initial stages of deployment the senior design team needed to take a look at not only the schematics of the development CC3200 LaunchPad board, which were accessible in .sch format and PDF, but we needed to open the actual PCB board file which was only available in OrCAD Allegro format. For this reason we had to look into using this software at some point in order to understand how the schematic files are being implemented into the PCB board file.

Due to budgetary constraints we were not able to purchase the full license for use in order to the LaunchPad file. Although we were able to get the free version by virtue of the fact that we are students, I did not do us much help. The reason why the demo version was not useful was because the LaunchPad files contained too many components, and the large number of components exceeded the maximum allocated by the demo version. For this we had to pursue an alternate solution.

27.1.2 Altium Designer

After our failed attempt to have full access of the CC3200 LaunchPad files using OrCAD Allegro we needed an alternate solution. In our pursuit for a solution Altium Designer came up, Altium Designer is also a very powerful PCB editing tool that is quite popular in the industry. The most important and useful feature that it held and feature that was of interest to us was the fact that it can take PCB schematic and board files from a large myriad of other PCB software, files of many different formats, and converts them to its own native Altium format.

Altium Designer gave us the solution that we needed, it too is not a free ware, but the fact that they offered a limited one month trial was all what we needed to convert the files and study them. Using the one month free trial we converted the OrCAD Allegro CC3200 LaunchPad files to Allegro files and we were able to successfully study it and obtain what we needed. Below some of the implementation methods that we needed to extract from the PCB board files:-

- Recommended drill and via diameter size.
- Recommended connection thickness, for communication wires and for power wires.
- The antenna matching circuitry. From looking at the PCB circuitry we learned that an antenna RFID chip cannot be simply connected to the CC3200MOD’s RF chip it needed to be matched using a connecting wire that that has an impedance of 50Ω.
- Placing specific parts in specific places to avoid interference and crosstalk between different parts.

The points mentioned above and many others assisted us in creating the final PCB prototype.

27.1.3EAGLE CAD 7.4.0

EAGLE CAD 7.4.0 was the major software that we used in completing the PCB that was used in the final prototype. The major benefit of using it was the fact that the free version, freeware had so many features, so much more that what OrCAD Allegro's demo version had too offer. EAGLE's free version had enough features and allowance for us to do most of the design for PCB board.

For added precaution and to be especially sure that the PCB circuit that we design works, with added emphasis to the antenna circuitry since it needs to be impedance matched, we consulted the Texas Instruments Engineer to Engineer community (aka e2e community) form. Over there we asked questions about the PCB board implementation of the CC3200 microcontroller for applications and other questions with regards to circuitry. We found the members there to be quite useful and they gave us quite the insight on what we needed to do. They also pointed out to us some very interesting open source projects and references that were outputted by some the engineers.

27.2 EAGLE Schematics of Final Prototype

After careful study of the data sheet and taking many other consideration into account (all of which were mentioned earlier in this document), the senior design group was able to successfully complete a design that is very close to the final design of the PCB. Below is a figure that shows the EAGLE CAD main schematic of the final design. Note changes will be made for the completely finalized design that will be sent out to the manufacturer to actually print out.

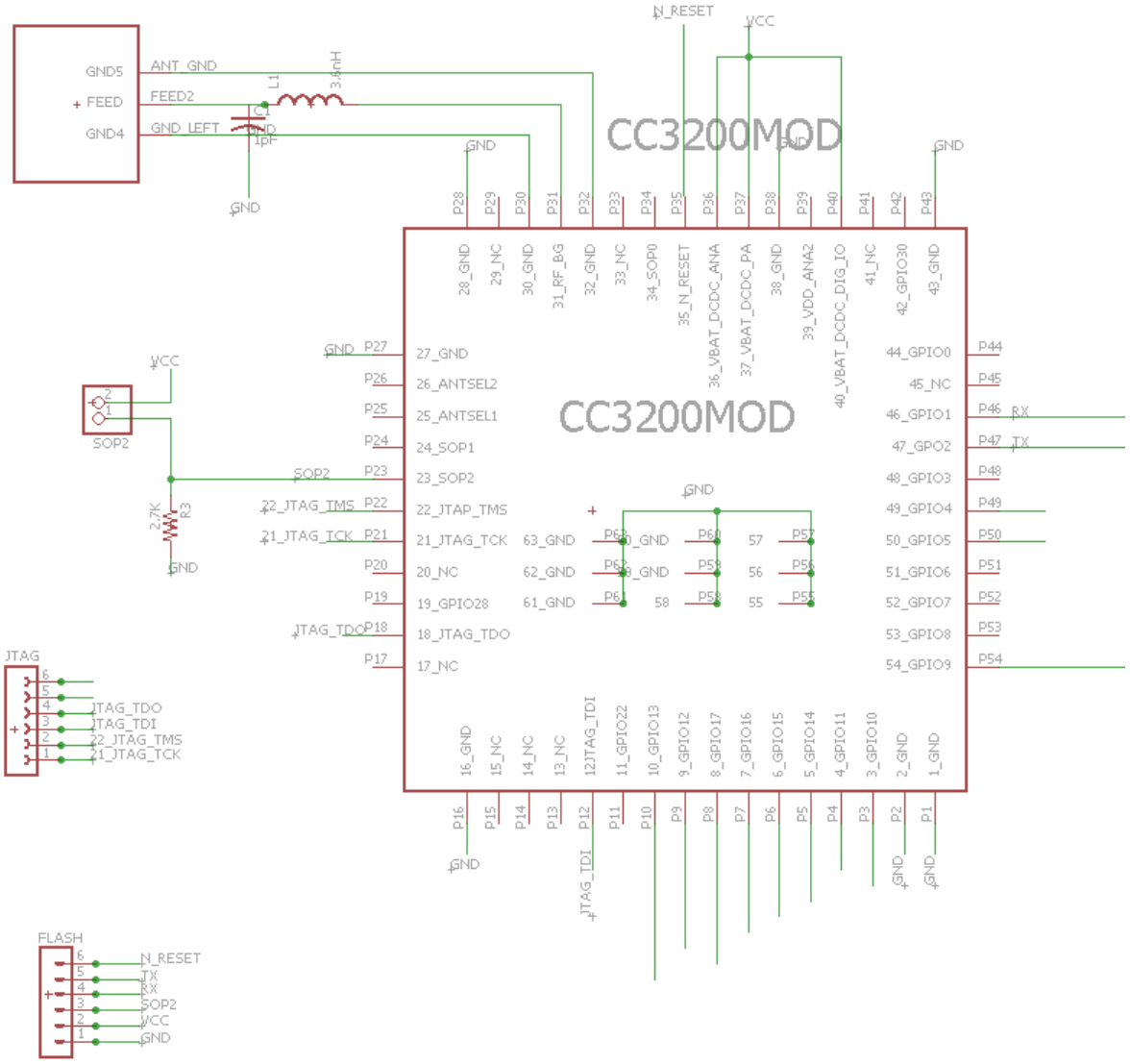

FIGURE 19: MCU AND HEADERS EAGLE SCHEMATIC

Here is some briefing of the schematic: All of the pins' default layout have been used as mentioned earlier in this document namely table 11, this is especially true for all of the pins we did not end up using in the final design. As for the pins that we used the schematic shows the final implementations of them. All the way from the top left of the schematic we can see the antenna chip connect in a way that is commentary to the reference that Texas Instruments provide to its customers.

A feature that we added on to this schematic is the use of jumper headers, this also considered as good practice especially when designing small project based prototype PCBs is the inclusion of headers. These header which we can use as jumpers later on come as quite useful indeed, the reason for this is because they are very beneficial in debugging the circuitry that is on the PCB. Another added benefit of including these jumpers is the fact that we can connect the flashing and the JTAG headers of the final prototype to that of the development LaunchPad in order to be able to program, this brings with it great savings in board space used (no need to include the programing circuitry) and in the overall cost and size of the final product. Below are more of the EAGLE CAD schematics that were used in the final schematic figures 20 – 27.

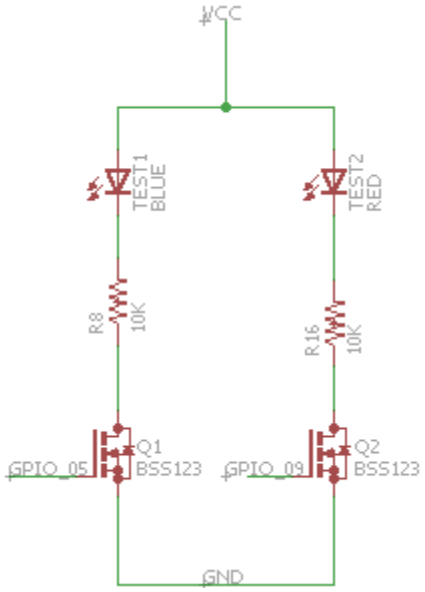

FIGURE 20: AUX LEDS EAGLE CAD SCHEMATIC

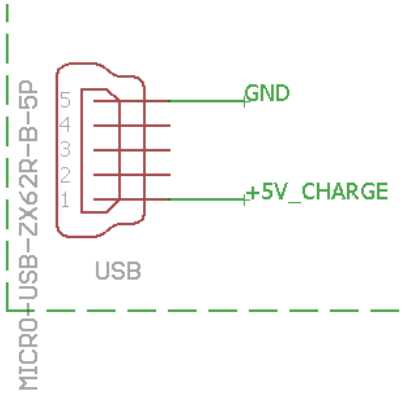

FIGURE 21: USB HEADER EALGE CAD SCHEMATIC

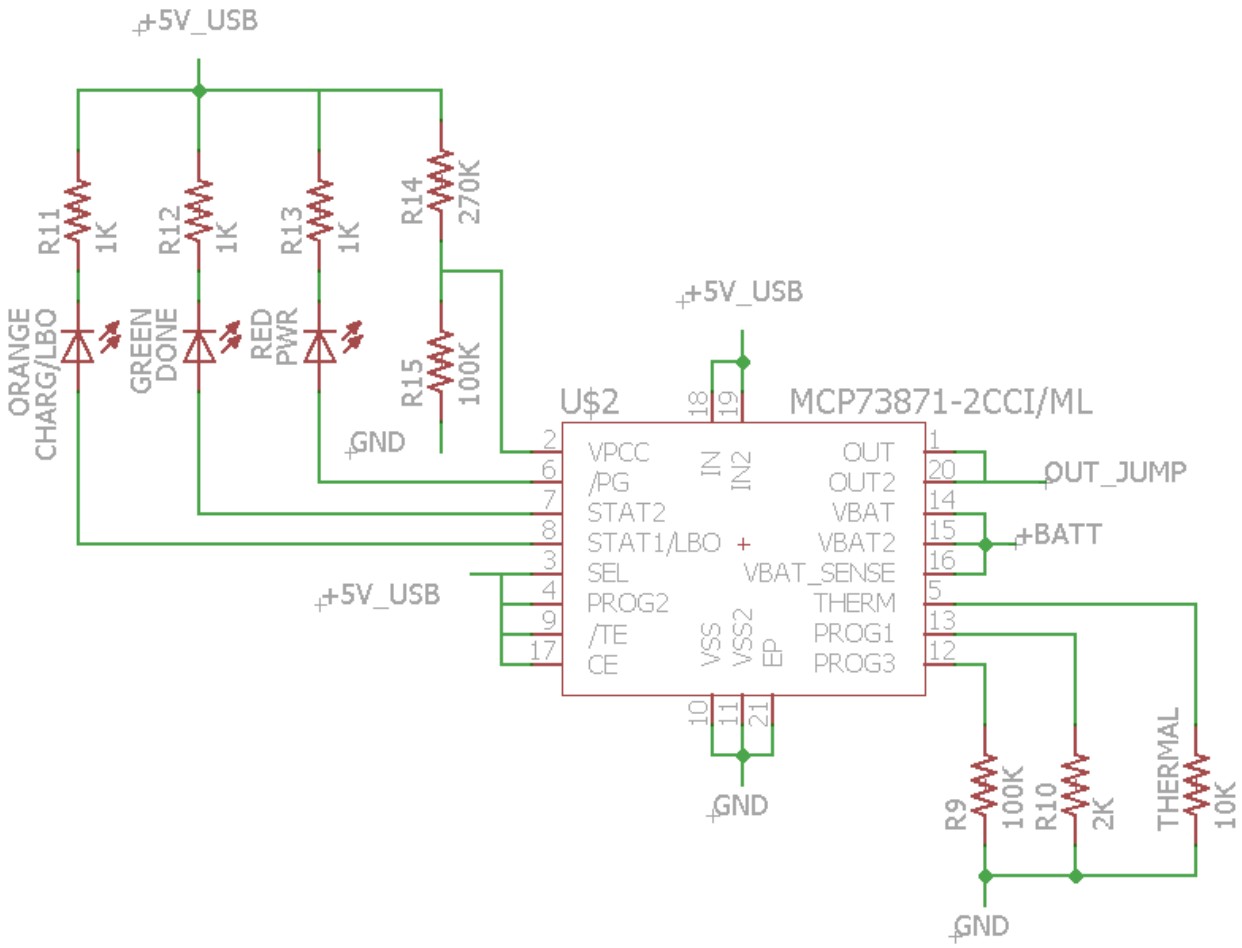

FIGURE 23: BATTERY CHARGER EAGLE CAD SCHEMATIC

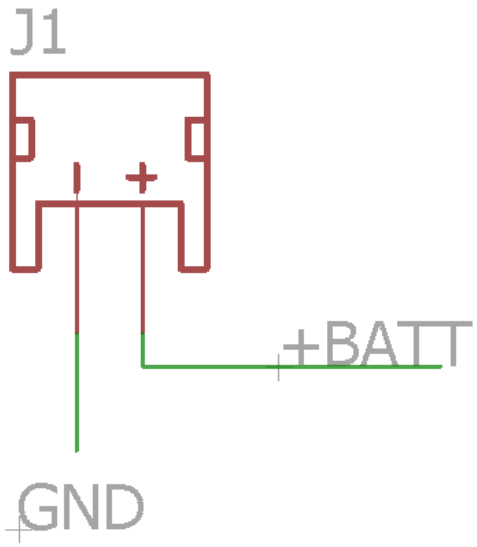

FIGURE 22: BATTERY HEADER EAGLE CAD SCHEMATIC

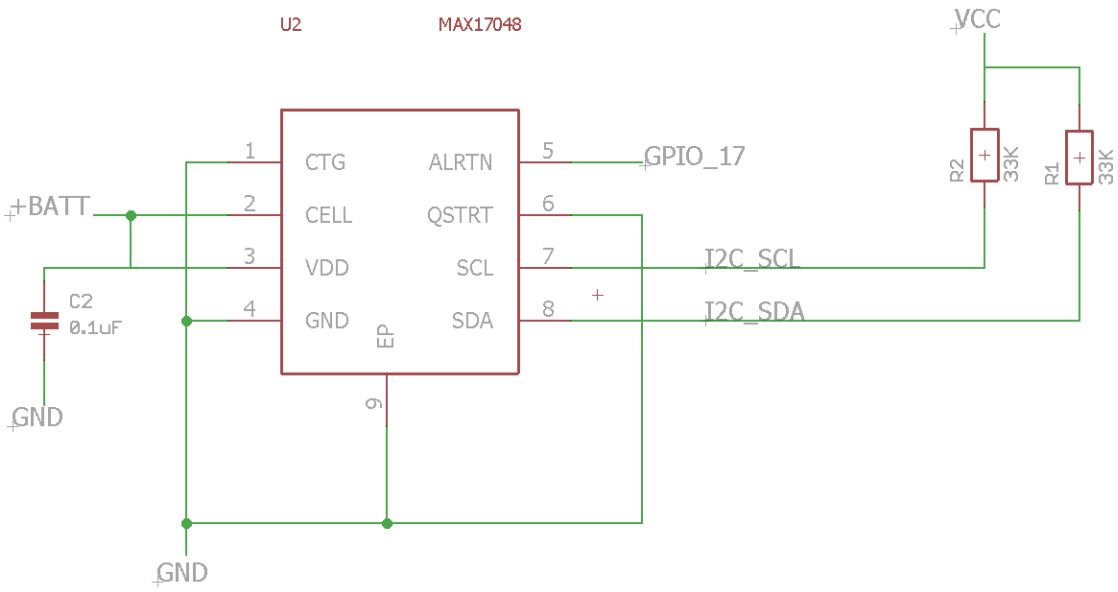

FIGURE 24: BATTERY GAUGE EAGLE CAD SCHEMATIC

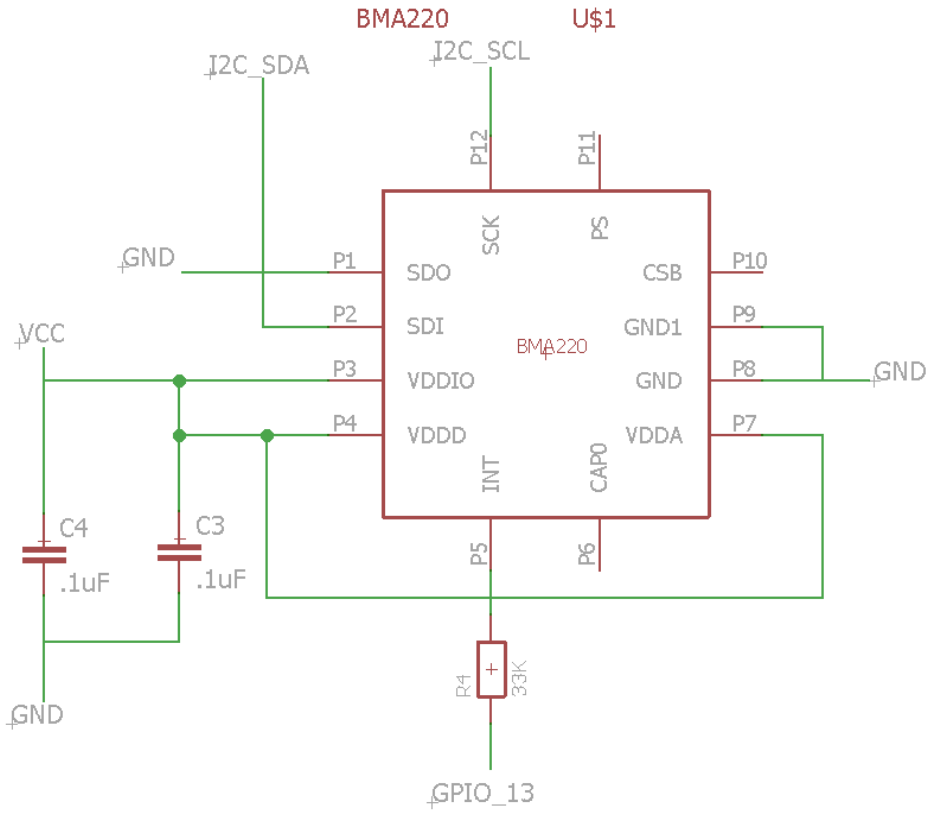

FIGURE 25: ACCELLEROMETER EAGLE CAD SCHEMATIC

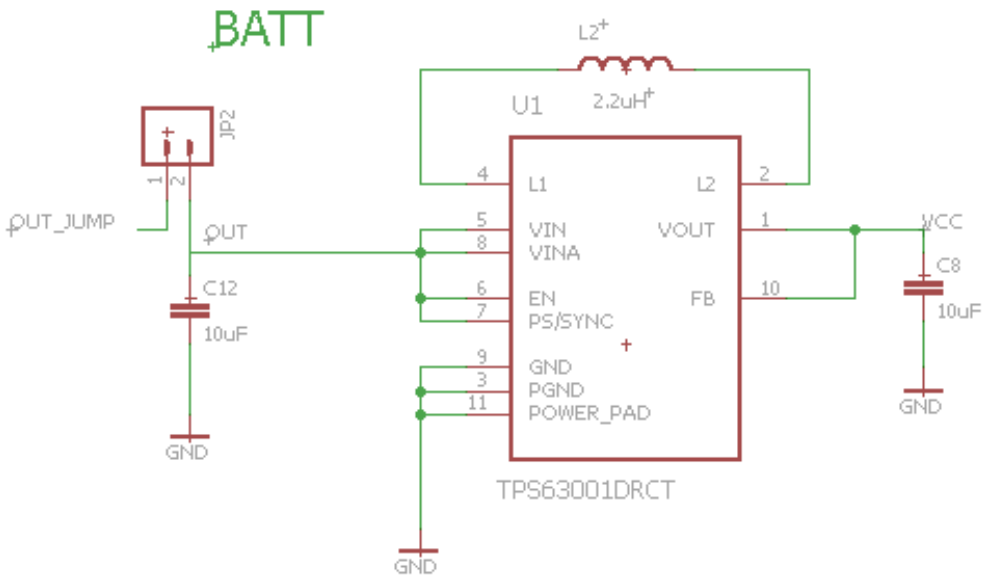

FIGURE 26: VOLTAGE REGULATOR EAGLE CAD SCHEMATIC

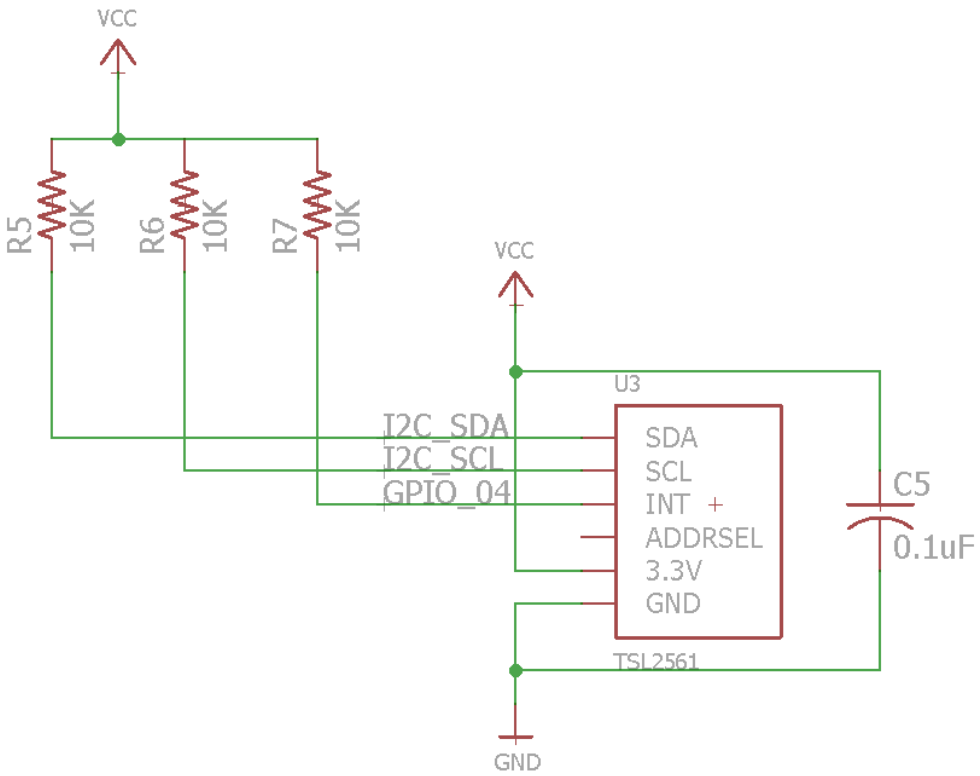

FIGURE 27: LIGHT SENSOR EAGLE CAD SCHEMATIC

27.3 EAGLE Board of Final Prototype

After completing all of the EAGLE CAD schematics, the next step that was needed to be taken was to actually implement the schematic designs on a PCB. A great feature that is offered by EAGLE is the ability to generate a board file from the schematic file. When generating the .brd from the .sch file the software automatically generates all of the footprints of all the components that were introduced in the schematic. After generating all of the footprints the software displays all of the components in the board space with air-wires connecting all of the networks or nets that have the same name.

The greatest benefit of using EAGLE CAD to design the PCB board along with the schematic is the fact that it allows you to cross caption between the two files. What occurs with the cross captioning is that any and every modification that is done on the schematic file it is automatically implemented on the board file. This helps in a great way, for it helps

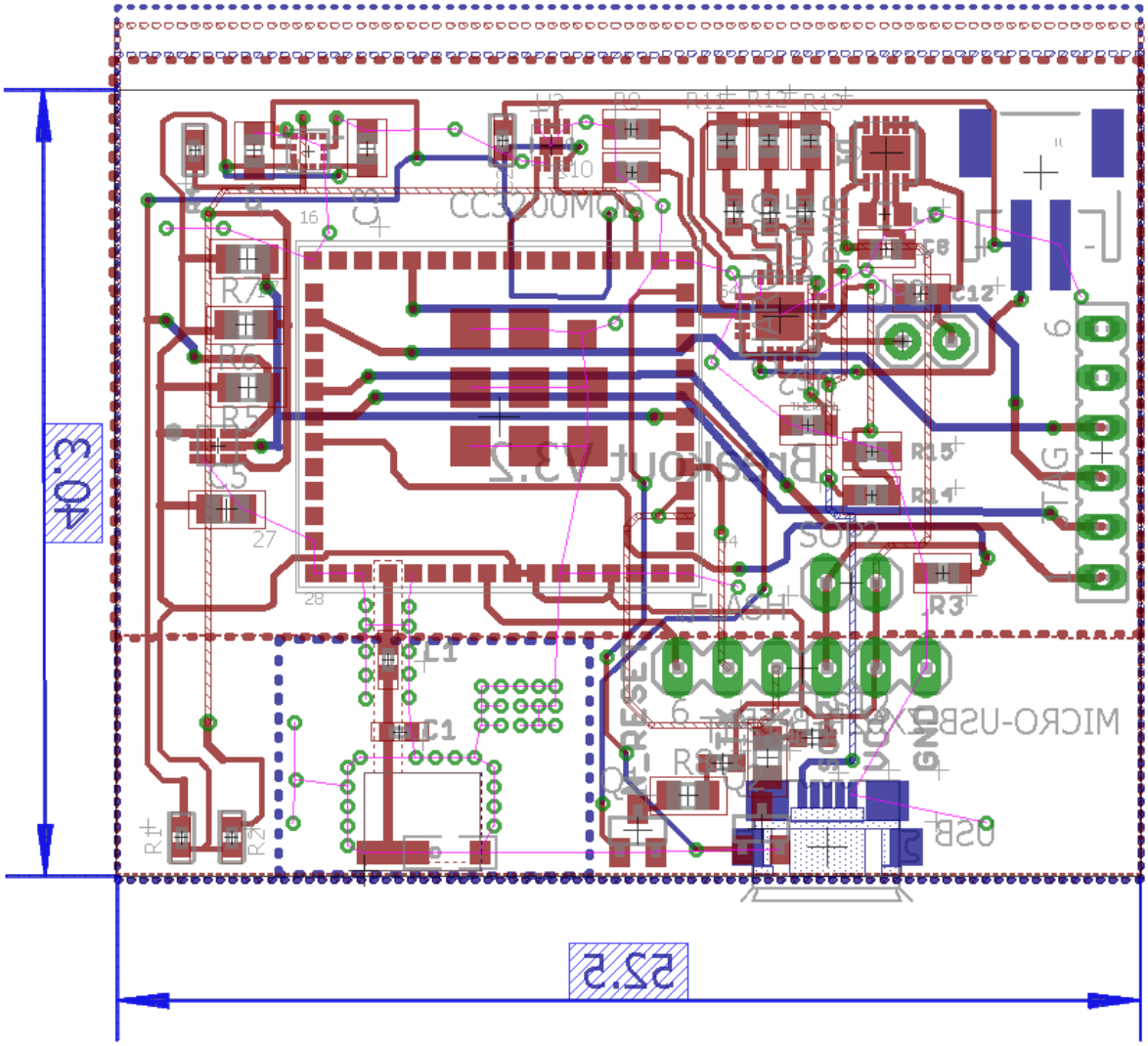

FIGURE 28: OVERALL SYSTEM PCB BOARD ON EAGLE CAD

keep all of the work that is done organized and synced up. Below is a screen capture of what the final prototype board currently looks like (at the time of this documentation):-

The figure above give the general final prototype PCB with all of the wiring shown. It is quite noticeable that there are quite a few air-wires that can be seen, all of those air-wires are of one net and that net is the ground. The reason why all of these ground nets are not connected with one another with actual wires is because there are a few polygons all over the PCB that are designated or mapped to ground. What happens is that by using ground polygons one can effectively make a ground pour.

The use of a ground pour also transcends a single layer, it goes beyond the top layer into the bottom layer by the use of vias that are placed all over the PCB. Ground pours can be very beneficial in that they simplify the routing process and by decreasing the number of discrete wires needed. Ground pours can also be quite useful in fact that it helps to dissipate heat in a more effective manner. In order to view the ground pour that has been placed on the PCB, the “ratsnest” view is needed to be enabled, this can be seen in the figure below.

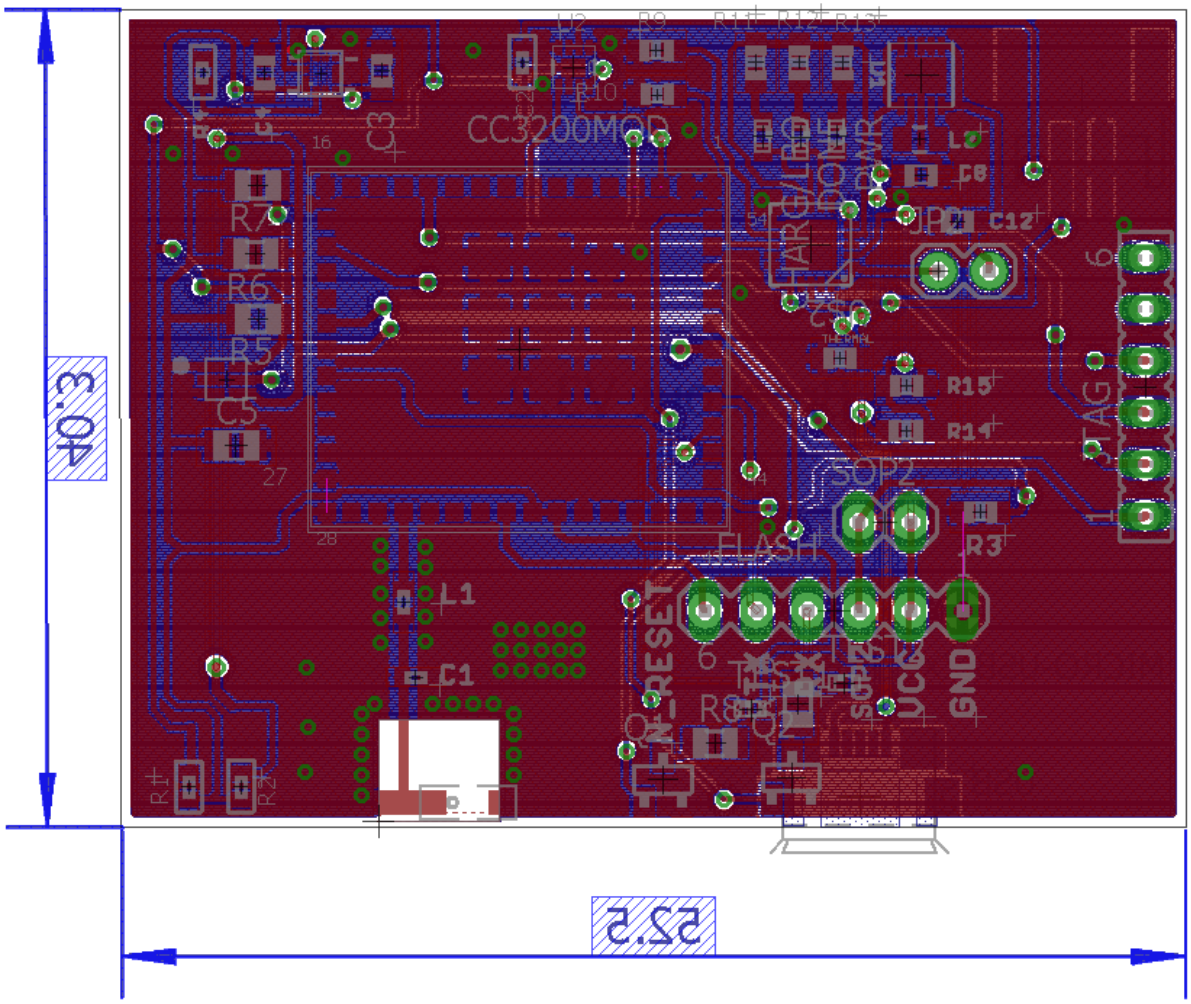

FIGURE 29: RATSNEST VIEW OF PCB BOARD

Now we can see that everything that was mapped to the ground is now connected to the ground pour and everything that is isolated from the ground (non-connect or connected to a different net) has been isolated from the ground pour. This can even be realized when looking at the wires that are connecting the different ends with each other.

One last point worthy of mentioning is the dimensions of the board and the layers. The as can be seen the dimensions of the board in millimeters is 52.5x40.3 this allows use to have an overall device size that is no larger than ~+10mm which is well under our projected 70x70mm that our initial system specification. One complication was the number of layers that were on this board, because of the design of the requirements of the CC3200 MCU we needed to have a board that was 4 layers, but this was not much of a hindrance for most of the work was accomplishable using only 2 layers (the top layer and the bottom layer), while for the few steps that actually needed us to exceed the free version allocation (which was 2 layers) we were able to access the university senior design lab where the university has available a computer that has the full version of EAGLE CAD installed.

27.4 Bill of Materials (BOM) of the system

The table below lists all of the materials that were used in that were used in the system:-

TABLE 21: BILL OF MATERIALS

Part	Value	Device	Package	Description
ANT_PAD	ANT	ANT	ANT	
C1	1pF	C-USC0402	C0402	CAPACITOR
C2	0.1uF	CAP_CERAMIC	C0402	Ceramic Capacitors
C3	.1uF	C-EUC0603	C0603	CAPACITOR
C4	.1uF	C-EUC0603	C0603	CAPACITOR, European
C5	0.1uF	C-USC0805	C0805	CAPACITOR, American
C8	10uF	C-EUC0603	C0603	CAPACITOR, European
C12	10uF	C-EUC0603	C0603	CAPACITOR, European
CC3200MOD	CC3200	CC3200	CC3200MOD	
CHARG/LBO	ORANGE	LEDSML0603	SML0603	LED
DONE	GREEN	LEDSML0603	SML0603	LED
FLASH		MA06-1	MA06-1	PIN HEADER
J1		JST_2MM_MALE	JST-2-SMD	Mates to single-cell LiPo batteries.
JP2		M02PTH	1X02	Header 2
JTAG		FE06-1	FE06	FEMALE HEADER
L1	3.6nH	INDUCTOR-SPARKFUN0402	C0402	Inductors
L2	2.2uH	INDUCTOR	L0805	Inductors
PWR	RED	LEDSML0603	SML0603	LED

Part	Value	Device	Package	Description
Q1	BSS123	BSS123	SOT23	N-CHANNEL MOS FET
Q2	BSS123	BSS123	SOT23	N-CHANNEL MOS FET
R1	33K	RESISTOR0402	R0402	Resistors
R2	33K	RESISTOR0402	R0402	Resistors
R3	2.7K	RESISTOR0603	0603-RES	Resistor
R4	33K	RESISTOR0402	R0402	Resistors
R5	10K	R-US_R0805	R0805	RESISTOR, American
R6	10K	R-US_R0805	R0805	RESISTOR, American
R7	10K	R-US_R0805	R0805	RESISTOR, American
R8	10K	R-US_R0805	R0805	RESISTOR, American
R9	100K	RESISTOR0603-RES	0603-RES	Resistor
R10	2K	RESISTOR0603-RES	0603-RES	Resistor
R11	1K	RESISTOR0603-RES	0603-RES	Resistor
R12	1K	RESISTOR0603-RES	0603-RES	Resistor
R13	1K	RESISTOR0603-RES	0603-RES	Resistor
R14	270K	RESISTOR0603-RES	0603-RES	Resistor
R15	100K	RESISTOR0603-RES	0603-RES	Resistor
R16	10K	R-US_R0805	R0805	RESISTOR, American
SOP2		PINHD-1X2	1X02	PIN HEADER
TEST1	BLUE	LEDSML0603	SML0603	LED
TEST2	RED	LEDSML0603	SML0603	LED
THERMAL	10K	RESISTOR0603-RES	0603-RES	Resistor
U$1	BMA220	BMA220	BMA222	The BMA220 is an ultra small triaxial, low-g acceleration sensor with digital interfaces, aiming for low-power consumer market applications.
U$2	MCP73871-2CCI/ML	MCP73871-2CCI/ML	QFN20-4X4	Stand-Alone System Load Sharing and Lithium-Ion / Li-Polymer Battery Charge Management Integrated circuit.

Part	Value	Device	Package	Description
U1	TPS63001DRCT	TPS63001DRCT	SON50P300X300X100-11N	HIGH EFFICIENT SINGLE INDUCTOR BUCK-BOOST CONVERTER WITH 1.8-A SWITCHES
U2	MAX17048	MAX17048	DFN200X200X80-9N	
U3	TSL2561	TSL2561	TSL2561_FN	TSL2561 Light-To-Digital Sensor
USB	MICRO-USB-ZX62R-B-5P	MICRO-USB-ZX62R-B-5P	ZX62R-B-5P	Hirose Micro USB Connector

27.5 Plastic 3-D Printed Encasing

As mentioned in the very beginning of the document in the system requirements section (section 3.2), the final product will be placed in a plastic encasing that will house all of the electronic components. Figure 30 shows how we envision the final system plastic encasing to look like.

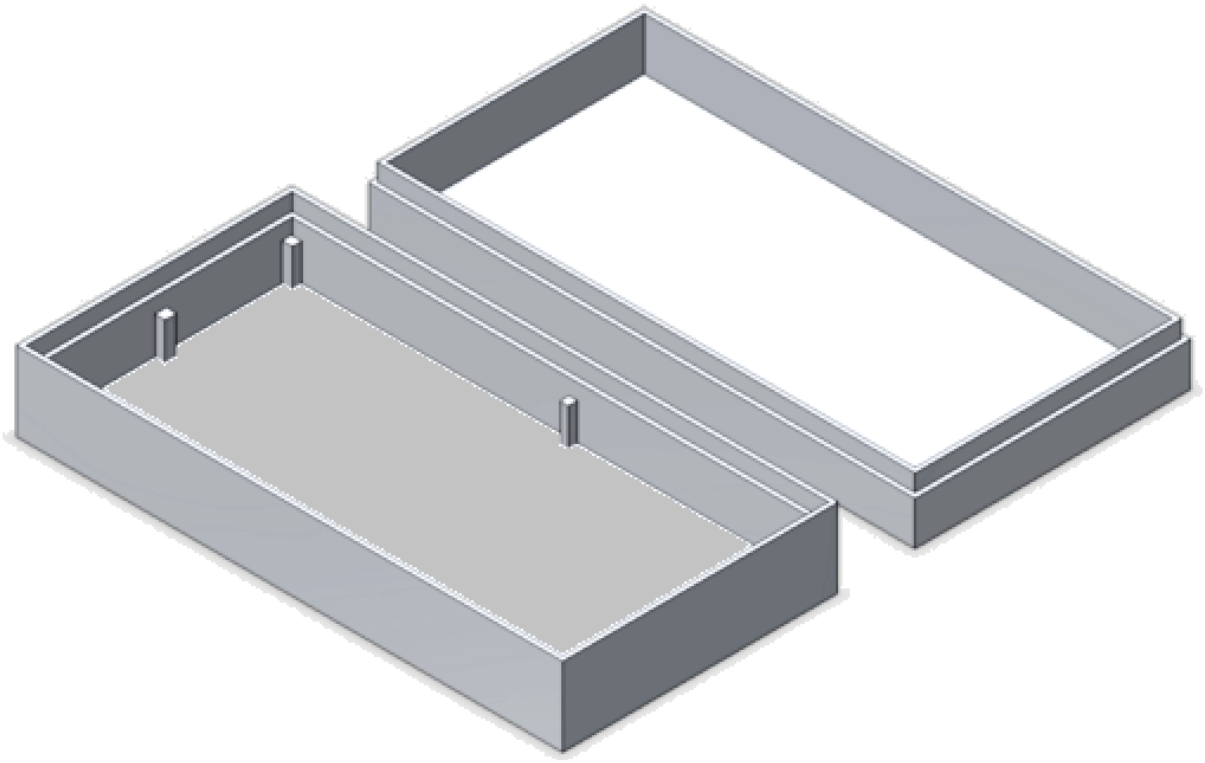

FIGURE 30: PLASTIC ENCASING OF FINAL SYSTEM PROTOTYPE

The encasing which was designed using AutoCAD drafting tool has a width of about 6 cm and a length of about 4.2 cm this encasing is more than sufficient to house the PCB which

contains all of the electronics. Also another feature the team decided to add to the encasing is to have a clear to that can be made out of clear plastic, this can be helpful as we can simply look and see the status of all of the LEDs without needed to make complicated light channeling plastic pieces.

The encasing shall be made in house by the use of 3-D printing. The material that we plan to use for 3-D printing will be ABS plastic which is quite strong and will be able to protect the system from many possible external damage. When 3-D printing the encasing we shall ensure that top and bottom portions of the encasing simply pop into place together rather than being screwed together, the reason for this is because of the constant need to access to the PCB for debugging purposes.

27.6 Mass of the Overall System

With any system and how the complexity of the hardware goes up as the number of final features increases, a factor that cannot be neglected is the overall mass of the system. Being how the device is supposes to operate it is in our favor to have the final prototype of the system to have as low of a mass as possible. The reason for this is to have the T.A.A.G be almost unnoticeable or un-impeding to the environment that it is placed in. below are the calculations that we have made for the masses of all of the comments.

Before calculating the projected mass of the system, an idea of the materials that some of the parts of the system consisted of needed to be known. The plastic encasing, as mentioned above, will be made out of ABS plastic this plastic has a density of around 1.42 grams per cubic centimeter. Another part of the system that also contributes great percentage of the systems mass is the PCB [28]. The PCB material will be a material called FR-4 which is a type of glass epoxy it has a density of about 1.84 gr/cc [29]. Below are the actual calculations that were made for the mass of the plastic encasing and the PCB.

Mass of the encasing = Length x Width x Thickness x 2 sides x density

= 6 x 4.5 x 0.2 x 2 x 1.42 = 15.34 grams

Mass of the PCB = same formula as the one for the plastic encasing

=5.25 x 4.03 x 0.3 x 1.85 = 11.74 grams

~= scale up the value to in include all of the components = 11.80 grams.

Another component of the system that also has a major mass contribution to the system is the battery. The battery has a mass of about 22 grams. With the major mass contributors put into consideration, now it is possible to find the estimated final mass of the overall system, which is the sum of all the masses i.e. 15.34 grams + 11.80 grams + 22 grams = 49.14 grams. This value is considered to be acceptable for it is below and close to the value, 50 grams that was originally in the requirements that was mentioned earlier in the document (section 3.2.7).

28 Final Prototype Testing

All of the tests used for the initial prototype can be reapplied to the final prototype. They capture many aspects of the desired functionality of system, such as making sure the startup phase of the program works, sensitivity settings have the desired effect, and that battery life is as expected. Performing these tests again ensures that we haven't lost any of the systems features, and so they are to be performed once again on the final prototype.

In addition to features that the initial prototype possessed, there are certain new features that need to be addressed on the final prototype. These range from the physical considerations, like weight and durability, to the functional, like the battery system, now integrated into the design. The tests below attempt to quantify that these parts of the design of working properly.

The Battery Charges: In the initial prototype the battery was part of an external kit. Now, the battery and all of surrounding components are integrated into the design. It is important that the battery can be fully charged through the USB port on the detector. To test this, we will allow the battery to charge on the detector and then use the gas gauge to confirm that indeed the battery is fully charged, or very close to it.

Wi-Fi Range: We must confirm that the antenna is working properly on the final prototype. Repeating the tests from the initial prototype will test this to a certain degree, but a better test involves comparing the range of the initial prototype, which has the development board's antenna, to the final prototype. As such, the test is to first find the distance from an access point at which the initial prototype ceases being connected to the network. Then, under the same conditions, find the distance at which the final prototype ceases to be connected to the access point. The two distances should hopefully not differ within more than 25% of each other.

Does it stick?: Remember, one of the main objectives of this design is to have the ability to stick the detector to most surfaces a person would want to, like doors and other flat wood and plastic surfaces. To test this, we will attach the detector to the following objects and check that it remains in place for at least an hour: a vertical finished wood surface, a painted wall, and an upside plastic surface. Although far from comprehensive, this will at least give us some assurance that the detector can remain in place. One possible bright spot on failing this test is that it might provide an equally valuable, although unscheduled, drop test.

29 Concluding Remarks

Bringing the T.A.G.G. system to fruition will involve a multidiscipline design effort. Already we have seen in depth hardware design in the form of component selection, circuit design, and PCB layout, as well as software design in terms of program flow and system interaction design. Undoubtedly, the implementation of the final prototype will involve even more research and problem solving.

Appendix A: References

[1] Texas Advanced Optoelectonic Solutions, "TSL2560, TSL2561 LIGHT-TO-DIGITAL CONVERTER," 2009.

[2] Bosch Sensortec GmbH, "BMA222 Digital, triaxial acceleration sensor Data Sheet," Reutlingen, 2012.

[3] A. Meland, "[Review]: CC3200MOD Vs CC3200 Launchpadxl," 13 01 2015. [Online]. Available: http://iottech.club/review-cc3200mod-vs-cc3200-launchpadxl/. [Accessed 07 12 2015].

[4] Texas Instruments, "bq27510-G3 System-Side Impedance Track™ Fuel Gauge With Direct Battery Connection," Dallas, 2013.

[5] Energizer, "Nickel Metal Hydride (NiMH)," 2010. [Online]. Available: http://data.energizer.com/PDFs/nickelmetalhydride_appman.pdf. [Accessed 05 12 2015].

[6] Cadex, "BU-203: Nickel-based Batteries," 2015. [Online]. Available: http://batteryuniversity.com/learn/article/nickel_based_batteries. [Accessed 06 12 2015].

[7] Energizer, "PRODUCT DATA SHEET ENERGIZER E91," [Online]. Available: http://data.energizer.com/PDFs/E91.pdf. [Accessed 05 12 2015].

[8] SparkFun, "Battery Technologies," [Online]. Available: https://learn.sparkfun.com/tutorials/battery-technologies. [Accessed 06 12 2015].

[9] B. University, "BU-206: Lithium-polymer: Substance or Hype?," 2015. [Online]. Available: http://batteryuniversity.com/learn/article/the_li_polymer_battery_substance_or_hype. [Accessed 06 12 2015].

[10] B. University, "Is Lithium-ion the Ideal Battery?," 2015. [Online]. Available: http://batteryuniversity.com/learn/article/is_lithium_ion_the_ideal_battery. [Accessed 06 12 2015].

[11] L. SHENZHEN PKCELL BATTERY CO., "Li-Polymer Battery Technology Specification," 03 06 2014. [Online]. Available: http://www.adafruit.com/datasheets/LiIon2000mAh37V.pdf. [Accessed 06 12 2015].

[12] C. C. t. Go), "USB CONNECTOR GUIDE — GUIDE TO USB CABLES," 2015. [Online]. Available: http://www.cablestogo.com/learning/connector-guides/usb. [Accessed 06 12 2015].

[13] Wikipedia, "USB," 04 12 2015. [Online]. Available: https://en.wikipedia.org/wiki/USB. [Accessed 06 12 2015].

[14] Visually, "Micro USB vs Mini USB," 2008. [Online]. Available: http://visual.ly/micro-usb-vs-mini-usb. [Accessed 05 12 2015].

[15] Texas Instruments, "bq24210 800-mA, Single-Input, Single-Cell Li-Ion Battery Solar Charger," 2015. [Online]. Available: http://www.ti.com.cn/cn/lit/ds/symlink/bq24210.pdf. [Accessed 05 12 2015].

[16] Gil Reiter, "A primer to Wi-Fi provisioning for IoT applications," Dallas, 2014.

[17] Texas Instruments, "CC32xx Power Management Framework," Dallas, 2015.

[18] Texas Instruments, "SimpleLink™ CC3100/CC3200 Wi-Fi Internet-on-a-chip™ Networking Sub-system Power Management," Dallas, 2014.

[19] Texas Instruments, "CC32XX SimpleLink Host Driver 1.0.0.10 (API)," Dallas, 2014.

[20] Texas Instruments, "CC32xx Blinky Application," Dallas, 2014.

[21] A. Ronacher, "Flask documentation," 2013.

[22] Assurance Technologies, LLC., "Passlib 1.6.5 documentation," 2015.

[23] Google, "Google Cloud Messaging - client libraries and sample implementations," 2015.

[24] Texas Instruments, "CC32xx SmartConfig Provisioning," Texas, Dallas, 2015.

[25] S. Cheshire and M. Krochmal, "Multicast DNS, Internet Engineering Task Force (IETF), PROPOSED STANDARD," 2013.

[26] Texas Instruments, "CC31xx & CC32xx UniFlash," Texas, Dallas, 2015.

[27] O. Saunders, "How long does an HTTP request take?," 2014.

[28] Stelray Plastic Products, Inc., "Reference Tables," Stelray Plastic Products, 2014. [Online]. Available: www.stelray.com/reference-tables.html. [Accessed 4 12 2015].

[29] ITS Engineering, "SSD Materials : G10-FR4," ITS Engineering, 2004. [Online]. Available: http://personalpages.to.infn.it/~tosello/EngMeet/ITSmat/SDD/SDD_G10FR4.html. [Accessed 4 12 2015].

[30] Texas Instuments, "Characteristics of Rechargeable Batteries," 2011. [Online]. Available: http://www.ti.com/lit/an/snva533/snva533.pdf. [Accessed 05 12 2015].

[31] Electropedia, "International Standards and Testing Applicable to Batteries," 2005. [Online]. Available: http://www.mpoweruk.com/standards.htm. [Accessed 05 12 2015].

[32] ETSI, "ETSI - Why we need standards," 2015. [Online]. Available: http://www.etsi.org/standards/why-we-need-standards. [Accessed 05 12 2015].

[33] T. Instrument, "bq27510-G3 System-Side Impedance Track™ Fuel Gauge With Direct Battery Connection," 11 2015. [Online]. Available: http://www.ti.com/product/bq27510-g3. [Accessed 06 12 2015].

[34] ENERGIZER, "Product Datasheet ENERGIZER CR2032," [Online]. Available: www.energizer.com. [Accessed 06 12 2015].

[35] T. Instrument, "bq27510-G3 System-Side Impedance Track™ Fuel Gauge With Direct Battery Connection," 2015. [Online]. Available: http://www.ti.com/lit/ds/symlink/bq27510-g3.pdf. [Accessed 06 12 2015].

[36] Visually, "MICRO USB VS MINI USB," 04 11 2014. [Online]. Available: http://visual.ly/micro-usb-vs-mini-usb. [Accessed 06 12 2015].

Appendix B: Standards Reference Name

Abbreviation	Name
AENOR	Asociación Española de Normalización y Certificación (Spain)
ANSI	American National Standards Institute sponsored by NEMA
AS	Australian Standard
ASE	Association Suisse des Electriciens (Swiss)
ASQC	American Society for Quality Control
ASTM	American Society for Testing and Materials
ATEX	Explosive Atmospheres (Safety directive)
BCI	Battery Council International (Publishes Automotive Battery Standards)
BS	British Standards
CARB	California Air Resources Board (Automotive Emission Standards)
CE	Conformance with EU directives
CEN	European Committee for Normalisation (Standards Committee)
CENELEC	European Committee for Electrotechnical Standardisation
CISPA	International Special Committee on Radio Interference
CODATA	Committee on Data for Science and Technology (Committee of ICSU)
CSA	Canadian Standards Association
DEF	Defence Standards (UK)
DEMKO	Danmarks Electriske Materielkontrol (Denmark)
DIN	Deutsches Institut für Normung (German Institute for Standardisation)
ECE	Economic Commission for Europe regulations.
EIA	Electronics Industry Association (USA)
EN	European Norms (Standards)
FCC	Federal Communications Commission (USA)
FIMKO	Finnish Electrical Inspectorate
FIPA	Foundation for Intelligent Physical Agents (Interoperability standards)
GB	Guo Biao = National Standard (People's Republic of China)
HSE	Health & Safety Executive (UK)
ICSU	International Council for Science
IEC	International Electrotechnical Commission
IEE	Institution of Electrical Engineers (UK)
IEEE	Institute of Electrical and Electronics Engineers (USA)

IMQ	Instituto Italiano del Marchio de Qualitá
IP	Ingress Protection
ISO	International Standards Organisation
ITU	International Telecommunications Union
JIS	Japanese Industrial Standard
KEMA	Keuring van Elektrotechnishe Materialen (Netherlands)
KIST	Korean Institute of Standards and Technology
MIL	Military Standards (USA)
MISRA	Motor Industry Software Reliability Association (UK)
MVEG	Motor Vehicle Emission Group (EU Emission standards)
NAMAS	National Measurement Accreditation Service (UK Calibration)
NEMA	National Electric Manufacturers Association (USA)
NEMKO	Norges Electriske Materiellkontroll (Norway)
NF	Norme Française (France)
NFPA	National Fire Protection Association (USA)
NIJ	National Institute of Justice (USA)
OSHA	US Department of Labor - Occupational Safety & Health Administration
OVE	Osterreichischer Verband für Elektrotechnik (Austria)
PowerNet	Automotive 42 Volt Battery Standard
RESNA	Rehabilitation Engineering & Assistive Technology Society of North America
SAE	Society of Automotive Engineers (USA)
SEMKO	Svenska Elektriska Materielcontrollanstalten (Sweden)
SEV	Schweitzerischer Elektrotechnische Verein (Swiss)
STANAG	NATO Standards Agreements
STRD	DTI Standards and Technical Regulations Directorate (UK)
TIA	Telecommunications Industry Association (USA)
TR	Technical Report (Used by IEC)
TÜV	TÜV Rheinland Group (TUV - Technical Inspection Asssociation)
UKAS	UK Accreditation Service (Assessment of test services)/(Calibration)
UL	Underwriters Laboratories Requirements (USA)
USABC	United States Advanced Battery Consortium
USNEC	United States National Electrical Code
UTE	Union Technique de l'Electriciteé (France)
VDE	Verband Deutscher Elektrotechniker (Germany)

Appendix C: Permission Emails

The following is the received Email after calling Texas Instruments' customer service help line (512-434-1560). Below is the Email response after explaining to Ed Watts, TI customer service rep that our senior design group would like to take Texas Instruments' permission to use some of their printed diagrams and information on our senior design report:-

Aiman Salih

From: support@ti.com
Sent: Wednesday, December 2, 2015 3:06 PM
To: Aiman Salih
Subject: RE: Service Request # 1-1982781610
Categories: Senior Desin Purchase

Hello Aimen,

Thank you for contacting TI support.

The information on our website www.TI.com is public information and can be used to reference details in your report.

Please let me know if you have any further questions.

Best Regards,
Ed Watts