Tamper Automated Alert Gadget (T.A.A.G)

Aiman Salih, Daniel Gibney and Leaphar Castro

Dept. of Electrical Engineering and Computer Science, University of Central Florida, Orlando, Florida, 32816

Abstract — **An implementation of a security system for household items which demonstrates the use of modern Internet Of Things technologies is presented. The system works by providing a small standalone device which, after being set up through the user's mobile device, can be unobtrusively attached to a household item. Upon either motion of the item or change of light around the item, the device provides a notification to the user's mobile device via a web server in the cloud. The architecture of this system is described in detail as well as the underlying motivation for the design decisions made. The design presented here focuses heavily on security, not only with regards to the systems stated purpose of providing physical security, but also in the security of information communicated by elements of the system. Secure ways of transmitting information from the mobile device to the detector device during the setup phase are explored.**

Index Terms — **Tampering device, Detector, Tampering event, IOT.**

I. Introduction

As a standalone product, the Tamper Automated Alert Gadget can provide the user a novel and convenient way of protecting some of their everyday objects found around the household. 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. The ability to know that your child has just picked up or moved something they're not supposed to would seem to be 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.

The data that are being sent in this system are analogous to the data that can be transmitted in an IOT system that would be used in a hospital. For instance, the EKG information of a very important social or political figure may be needed by their physician to constantly monitor their condition especially if they have a tendency for suddenly falling ill. As such, the information that is being received by the physician in the hospital's computer database must be communicated securely to protect it from potential hackers that may want to steal some of this important person's medical information. This shows the importance of software security and encryption in IOT systems and the need to protect them from vulnerability. The system in this senior design project is designed to give an example of such a system.

The remainder of this paper will give insight on the systems design and operation, including the systems hardware components and its software components. Additionally, some of the methodologies that were used to complete and test the project are described.

II. Overall System Design

At a system level, the project can be broken down into the following major modules:

- Detector: This is the device placed on the user's item of interest. Small and battery powered, this is a device the user sets up by syncing it with their mobile phone or tablet. This is also where all of the hardware integration for this project was done.
- Web Service/Database: This is an application running in the cloud. It is what the detector communicates with in order to have notifications sent to the user. The web service also logs internally all of the notifications sent out by any detector and information regarding which user the notification was for.
- Mobile application: The mobile application serves as the user interface for the whole system. It is the tool for the user to set up the detector, allowing the detector to connect to the Wi-Fi network. It allows the user to program the detector with their desired sensitivity settings for detecting tampering. Notifications are received through this application as well.

Figure 1 shows a block diagram of the system architecture.

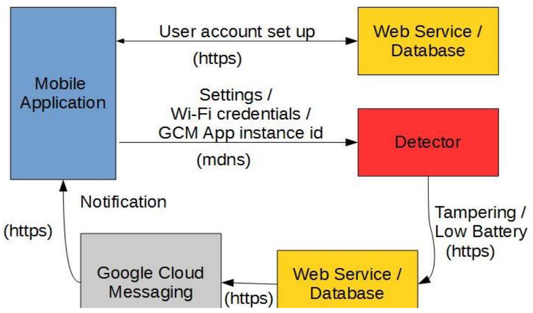


Figure 1: Overall System Block Diagram

III. Hardware System Components

The entirety of the components that were used in this system were circuited together. A single PCB was designed making one concise circuit board for the entire detector system.

Figure 2 shows a block diagram of the hardware.

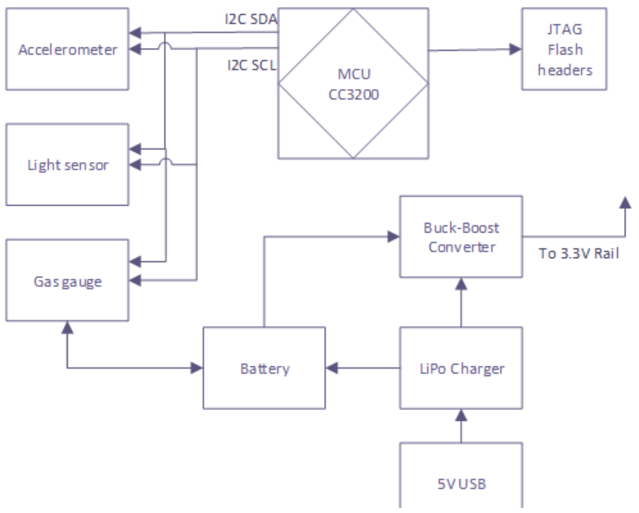


Figure 2: Overall Hardware Block Diagram

A. *Microcontroller- CC3200*

With how our system is meant to be applied and accomplished (IOT application, with Wi-Fi connectivity) our search led us to an ideal candidate microcontroller, the Texas Instruments' CC3200 SimpleLink™ Wi-Fi® and Internet-of-Things solution, a Single-Chip Wireless MCU. This MCU comes with its own Wi-Fi module embedded, which prevented us from needing a dedicated module for that purpose. This contributed to a compact system design. Below are some of the MCU's key features most relevant for this project [5]:

- A dedicated Wi-Fi Network Processor: This was useful for the reasons that are given above. And, besides preventing an additional hardware module from being needed, it offloads Wi-Fi and Internet Protocols from the application microcontroller leaving it free to do work for our application. The software library support for networking the CC3200, called SimpleLink, also provides for easy, autonomous Wi-Fi connections.
- Embedded Memory:
  - RAM (Up to 256KB)
  - External Serial Flash Bootloader, and
  - Peripheral Drivers in ROM

  In particular, the ROM is required to store information when the microcontroller is placed in hibernation mode and information stored in RAM is no longer retained, this is explained more in the microcontroller software section.

B. *Light and Motion sensing*

Light sensing: The system employs a TAOS TSL2561, which 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.

Motion sensing: The system employs the Bosch BMA222 3-axis digital accelerometer, 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.

C. *Battery*

The best option among the battery technologies we looked into was a 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 in 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 harmful for the battery, as well as the system if your battery isn't performing like it supposed to. 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

Power Monitoring system: The battery monitoring system is a key element to producing a low maintenance device with low power design. This will be helpful when you have multiple devices connected to a network; instead of going to each individual device and checking each battery to make sure it has been charged one can use the battery monitoring system. 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 could relay the information back to the user through their mobile device. This type of system will be a key part of 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] The single series cell Li-Ion battery fuel gauge can be found on our board. With Patented Impedance Track technology with the possibility to send reports of remaining capacity, the state of charge (SOC), and time-to-empty.

D. 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. 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. The 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 the normal mode of operation. 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.

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 the ability to monitor the battery and charge it from the 5 volts that it receives from the USB connector. This section will describe briefly the IC and reference circuits that we considered during the design of the system. The figure 3 located below gives a reference schematic of the IC being implemented into the system

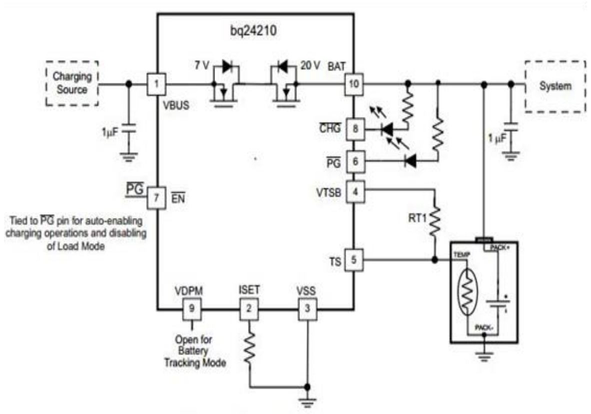

bq24210
Charging Source
7 V
20 V
BAT
System
VBUS
1µF
CHG
PG
1 µF
VTSB
PG
EN
RT1
Tied to PG pin for auto-enabling charging operations and disabling of Load Mode
TS
TEMP
PACK+
PACK-
VDPM
ISET
VSS
Open for Battery Tracking Mode

Figure 3: Battery Charging Reference Circuit

This integrated circuit comes equipped with input voltage dynamic power management. It can also provide thermal regulation protection, output current protection, and battery short protection. The figure 3 located above gives a reference schematic for how to connect the IC to a system.

E. Voltage Regulator

The switching regulator is an important component because our power efficient design requires it to convert voltage efficiently. It also increases our design flexibility because we do not have to worry about choosing components that use the same voltage to operate. Basically, this means that we can get a single output voltage from a varying input voltage.

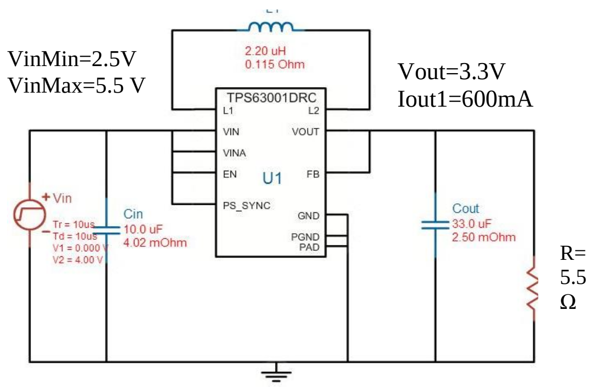


Figure 4: Buck Boost Voltage regulator from TI's Webench

In the figure 4 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 is 3.3V which is the operating voltage of all of the IC chips in the system. The buck-boot topology is beneficial due to the fact that the battery's voltage rages from 4V-1.8V depending on how full or depleted the battery is, as such the converter can adapt to this voltage range and provide a consist output. With this reference design we should be operating with an

Since one of the goals of the product was to be inconspicuous when placed on one of the user's objects of interest, the PCB had to be compact. Figure 5 shows an Eagle CAD board diagram of the printed circuit board. The

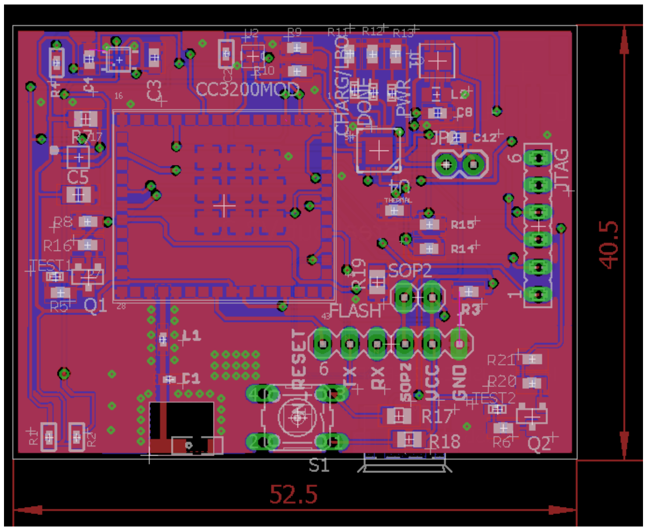


Figure 5: PCB on EAGLE CAD

board dimensions are 52.5x40.3mm which is quite comparable to the size of the Samsung SmartSence multipurpose detector device which is 48.3x34.3mm (magnet not included) [6].

IV. Software System

A. Microcontroller Software

The microcontroller software system has several responsibilities. It must successfully connect to the user's Wi-Fi network, detect tampering through the use of the attached sensors, communicate the tampering event to the web service, and to minimize the power consumption of the battery by placing the microcontroller in hibernation mode whenever possible. A program flow to accomplish this is shown in Figure 6.

The first thing that the software running on the microcontroller must do is to 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. TI's SmartConfig technology provides a 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 allows it to wait for an application such as the Android application to transmit the

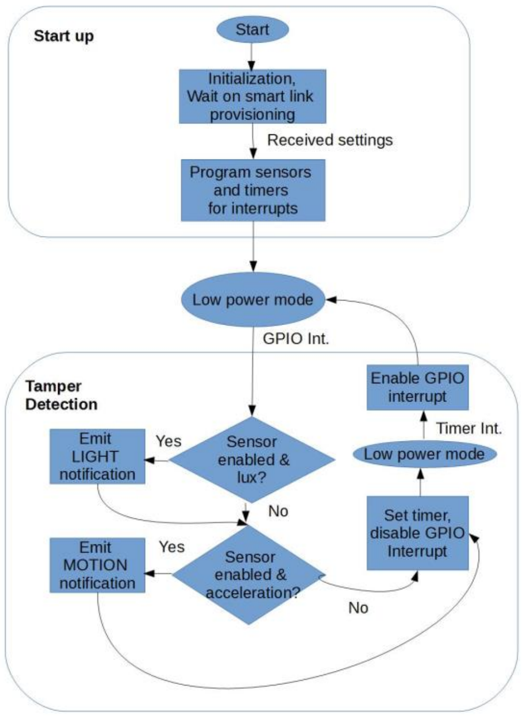


Figure 6: MCU Software Flow

network 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.

Next, additional user information needs to be communicated to the detector. Multicast domain name system, or mDNS, is used to communicate this information. The CC3200, immediately following the provisioning step, becomes an mDNS listener. At the same time, the Android application becomes an 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.

For sending notifications to the user application, HTTP posts to Google Cloud Messaging, or GCM , are used. These are HTTP post requests. This application layer protocol is provided for us by TI's HTTP Client Library. 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.

Power consumption is reduced by placing the CC3200 into hibernation mode. In hibernation mode CPU context and RAM are not retained, and the wake up source cannot be identified. Because of the ram being lost, needed user values must be stored in a file in flash. Because of the wake-up source being undetectable, the gpio pins are checked to detect which sensor caused the interrupt. The reason for selecting this power mode was the 4 uA current draw of the microcontroller [7]. Power consumption can be additionally reduced by putting the accelerometer into low power mode.

Threshold values programmed into the motion sensor for triggering an interrupt are from a fixed set of values. The threshold values used are based on which setting the user as selected in the syncing process. The threshold values for the light sensor are based on the lux values read from the sensor during the syncing process. The upper value is found by considering the current lux value multiplied by some factor while the lower bound is found by considering the current lux divided by some factor. This factor depends on the user-selected sensitivity. These values were found through experimentation.

B. Web service

The role of the web service is to ensure that the user has a unique username and password, to forward the notification to the appropriate user, and to log the tampering notifications sent out by the detector into a database. The web service can be broken down into two major components. The first of these components is the API which the web service provides to the detector for sending notifications. The second of these is the database system.

The API written for the system runs in the cloud using a suite of services offered by Google, called Google Cloud Services. In particular, it is a python application running on top of a deployment application provided by Google called Cloud App Engine. A framework called Flask is used to make handling HTTP request easier. Using this framework each route provided by the API becomes a function. Each function does what is required of it in terms of database access.

Notifications are sent by the web service when a particular route is used. Inside the body of the request to these routes are the username, password, and message body. The correctness of the username and password is verified in order to avoid unwanted parties from causing notifications to be sent. After the web service checks these credentials, it finds any Google Cloud messaging tokens associated with that username and pushes notifications to Google Cloud Messaging.

The database also runs on the services provided by Google Cloud Services, Google Cloud SQL. Three tables are used within this database. One is to store user information, another is to store tampering logs, and the last is to store the google cloud messaging tokens. These tokens identify specific mobile devices belonging to the user and are needed by the web service to push notifications to the user's device.

C. Mobile Application

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 goal of the application is that following the initial provisioning step 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 at first appearing inconvenient, does provide a layer of security since no person could remotely modify the settings on the detector.

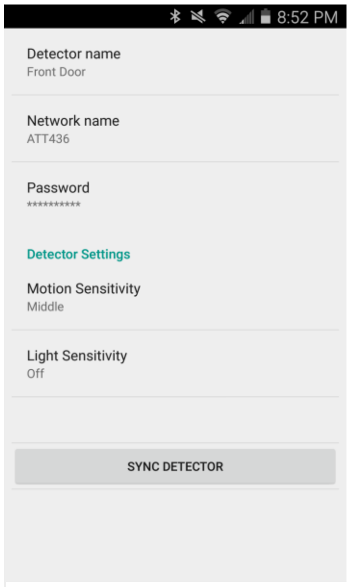
In the detector startup phase, the mobile application must get the detector to connect to the wireless network and then transmit the information the detector needs to function properly. The major steps taken for the provisioning part of the program are the following.

1. Get the SSID and the password from text fields of the user interface
2. Get the gateway by looking at the network which the mobile device is currently connected to.
3. Put this information is information into a new SmartConfig object, a class provided by Texas Instruments for using the SmartConfig protocol.
4. Run method smartConfig.transmitSettings().
5. Use mDNS to identify the new device

At this point, assuming all of the above went well, the detector is now connected to the network and through the 5th step, the mobile application is aware of this. The mobile application is now ready to transfer the additional settings to the detector.

As mentioned earlier, the sensor settings and user information transfer to the detector is done through mDNS. This is an additional mDNS step, different from the one done in the steps above. Specifically, after the mobile application identifies the detector, it starts broadcasting a service through the mDNS. The name of this service is predetermined and known by the detector. In this mDNS advertisement, the text field includes the additional information needed by the detector. This includes the username, the password, and the sensor settings. This broadcast lasts for 10 seconds before it is de-registered. It was found that this is an adequate amount of time for the detector to receive the information.

Figure 7. Mobile device interface for syncing detector


8:52 PM

Detector name
Front Door

Network name
ATT436

Password
****

Detector Settings

Motion Sensitivity
Middle

Light Sensitivity
Off

SYNC DETECTOR

Figure 5: Mobile App UI

D. Encryption and security

There are two main vulnerabilities in the parts of the system under our control (this excludes, for instance, securing Google's Services beyond proper configuration). The first is the mDNS step in which sensitive information such as the username and password are advertised over the network. The second is the transmission of the username and password through HTTP requests to the cloud.

To abate the first issue, the plain text used in the text field of the mDNS broadcast can be encrypted. A key contained on the detector can be used to decrypt the plain text after it is received. Assuming that symmetric key encryption is used, keys on both the detector and the mobile application can be periodically updated. The detector can update over-the-air with some slight modification to its current code, and the mobile application can be easily updated as well. This is not currently implemented in the prototype, partly because the original purpose of the system was as a platform on top of which future security measures can be implemented.

The second issue of how to securely transmit usernames and passwords inside of HTTP requests has been solved long ago. The CC3200 simple link library supports HTTP over TLS. The TLS protocols supported by the Texas Instruments CC3200 software library include 6 different ciphers one can use, including RSA with AES 256 and CBC_SHA.[8] Using this sort of solution for the common operation of an HTTP request to a cloud API is standard practice. This is not to say, that researchers cannot add an additional layer of end-to-end encryption on top of this.

Inherent Security Limitations in the Design

It is important to consider what the T.A.A.G. system can do and cannot 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. Additionally, even with the sensor settings selected properly, it is still possible for the 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, of subtly disable, the detector
4. Turn back on the network

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 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.

V. System Testing

A. Tamper Detection

How successful is the detector at detecting when someone is tampering with an item? To test this, a prototype was attached to several likely items that the system would be used for. The first is the front door, the second is the inside of a safe, and the third is the backside of a cardboard box. For all of the experiments, settings were picked which seemed intuitive for the given situation. Specifically, for the door the motion sensitivity was set at high while the light sensitivity was disabled; for the safe the light sensitivity was set to high while the motion sensitivity was disabled; and for the cardboard box both sensitivities were left at low(in an effort to avoid false triggering from slight shaking or changes in ambient light.) Each experiment was repeated five or more times, and each of these times the nature of movement applied to the object or the change in exposure to light was varied slightly.

For the door, it was discovered that on all the experiments the detector successfully detected when the door opened. However, the detector appeared slightly too sensitive. Notifications were sometimes sent due to unforeseeable movement, perhaps due to outdoor breezes or nearby footsteps. Efforts to change the acceleration thresholds on the accelerometer resulted in the detector being unable to detect a very gradual opening of the door. Improvement to acceptable levels of sensitivity was finally reached by increasing the number of samples used in calculating whether the threshold was crossed.

The experiments conducted with the safe were more successful. For every run, a notification was sent when the safe was opened and no notifications were sent while the safe was closed. This is most likely due to a large difference in the lux readings found by the light sensor in total darkness and ambient room light. As we see with the next experiment slight changes in ambient room light are far more difficult for the detector to perceive.

The last experiment with the cardboard box attempted to test both the functionality of the light detection and the motion detection. The box was placed in a somewhat darker room with its door closed. The goal was that a notification should be sent if either someone in the room moves the box, or if someone opens the door to the room. In terms of motion, it was found that the detector behaves well. No false motion detection notifications were sent and whenever the box was moved a motion detection notification was sent. However, we were unable to cause a light detection event by subtly changing the ambient lighting in the room by opening the door or window blinds. Shining a light directly at the detector was sufficient, but this was not the desired result.

It appears as though it is indeed a very slight range of thresholds, both in terms of light and motion which are well suited for a particular purpose. One proposed extension is the creation of a catalog of such accelerations and changes in lux suited for a particular use. This could be built into the product and would allow to user to select the appropriate settings based on their needs. Such a catalog might be useful in general since it measures physical quantities and is not tied to any particular device.

B. Battery Life

Three scenarios are presented below along with their expected amounts of time before the battery runs out of charge. These expected times are then compared with some experimental results.

The time that the CC3200 spends in active mode while sending a notification is dominated by the time spent waiting for Google Cloud to respond. This can be up to 30 seconds. This can vary greatly and be as short as 5 seconds in some cases. We simplify the calculations by saying that the CC3200 spends that entire 30 seconds in transmit mode which pulls 272 mA according to TI's documentation. When the CC3200 is in receive mode it only draws 53 mA [7].

Stuck at startup: 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 CC3200 current draw is 53 mA. The other components, even in active mode, draw comparably insignificant amounts and so are ignored here. The battery life is therefore 1200 / 53 = 22.48 hours.

It was found through three 24 hour runs that average time before the battery charge is too low to maintain operation is between 23 and 24 hours, agreeing with this prediction.

No interrupts: At the other extreme, the detector might never have an event to detect. In this case, it can remain in hibernation mode. The CC3200 current draw here is a meager .004 mA. At this point is useful to include the other components in the calculation. The light sensor current draw is .24 mA and the motion sensor .0345 mA. The total battery life should be 1200 / .2785 = 4309 hours or 179.5 days.

Experimentally, this was clearly too long to wait for a test to complete. Instead, data from after 24 hours was extrapolated to give the expected battery life. This data was collected using the gas gauge monitor built into the design. It was found over 3, 24 hour runs with one of the early development board prototypes that without interrupts the battery life had dropped 2.7% within the day. This suggests a battery life of 37 days instead of the expected 179.5 days, a significant difference.

One possible explanation for this discrepancy comes from periodic waking up of the CC3200, the gas gauge monitor, and the power needed for the UART communication to print the result read from the gas gauge to the terminal. Also possible, is that further configuration steps are required to tune the gas gauge properly to get a truly accurate reading from the battery.

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 sent out twice a day, which gives us 60 seconds. Considering that there are 86,400 seconds in a day and that we have a 1200 mAh battery. The expected CC3200 current draw in a day is

$$(60/86{,}400) * 272 + (85{,}340/86{,}400) * .2785 = 0.4640 \text{ mA}$$

The battery life should be roughly 1200 / 0.4640 = 2586 hours or 107.75 days.

This was checked experimentally on a prototype board by programming the CC3200 to wake twice a day and send a notification containing the percentage of battery remaining. This was done three times, each time for one day. Again the average drop in battery life was roughly three percent.

These results suggest that either the way in which we are estimating current consumption for the detector is inaccurate, or that the gas gauge is not providing accurate readings. Further work is still being done to explain these results.

Casing: Figure 8 shows the prototype casing.

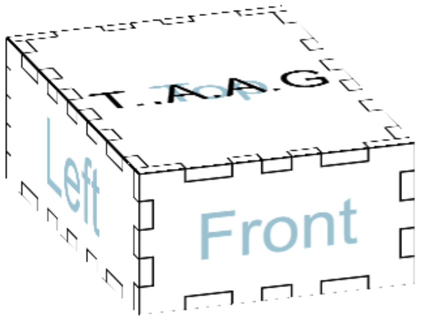


Figure 6: Prototype 3D Model Casing

The dimensions for the 3D casing in the figure above for our device are 50mm x 25mm x 65mm. We have decided to go with a dark Plexiglas plastic that is durable, and strong enough that it can withstand being placed on items that may be pressed against hard surfaces. It is light and small enough to be adhered to many different common objects.

VI. Conclusion

The Tamper Automated Alert Gadget (T.A.A.G) is a product that can be used to protect the user's household objects. It also serves as an example of a modern IOT system. The T.A.A.G detector, which was designed and implemented by the members of the senior design group, acts as the heart of the alert system and is one of the main pieces of hardware involved in the system's operation.

It is important to consider security in terms of the information transferred between elements of an IOT system. The T.A.A.G system, as well as being useful, hopefully, provides a platform on which researchers can try to improve security even further, through the development of new encryption methods well suited for an application such as the T.A.G.G system.

.

Acknowledgement

The authors wish to acknowledge the assistance and support of Dr. Jiann S. Yuan, the Co-Director of the MIST Research Center at the Universal of Central Florida. We would also like to recognize and acknowledge Dr. Richie for his continued advice and student mentoring. The authors would also like to congratulate Dr. Riche for his retirement and would also like to congratulate him on achieving the title of Professor Emeritus.

References

[1] 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.
[2] Texas Instrument, "bq27510-G3 System-Side Impedance Track™ Fuel Gauge with Direct Battery Connection,"112015. [Online].Available: http://www.ti.com/product/bq27510-g3.
[3] Texas Instruments, "bq27510-G3 System-Side Impedance Track™ Fuel Gauge with Direct Battery Connection," Dallas, 2013.
[4] Texas Instruments, "Characteristics of Rechargeable Batteries," 2011. [Online]. Available: http://www.ti.com/lit/an/snva533/snva533.pdf.
[5] Texas Instruments, "CC3100/CC3200 SimpleLink™ Wi-Fi® Interneton-a-Chip," 2014. [Online]. Available: http://www.ti.com/lit/ug/swru368a/swru368a.pdf.
[6] Samsung, "Samsung SmartThings Multipurpose Sensor _ Things _ SmartThings Shop," 2016. [Online]. Available: https://shop.smartthings.com/#!/products/samsung-smartthings-multipurpose-sensor.
[7] Texas Instruments, "CC32xx Power Management Framework," Dallas, 2015.
[8] Texas Instruments, " CC32xx SSL Demo ApplicationDallas, 2014. http://processors.wiki.ti.com/index.php/CC32xx_SSL_Demo_Application

Aiman Salih will graduate from the University of Central Florida in May 2016 with his Bachelors in Electrical Engineering. He currently works as an undergraduate research assistant under the supervision of Jiann S. Yuan where he investigates novel structures for super-junction power MOSFETs. He plans to come back to Central Florida to complete his Master's degree in Electrical Engineering with a specialization in Micro-Systems and Nano-Systems.

Dan Gibney will graduate from the University of Central Florida in May 2016 with his Bachelors in Computer Engineering. He works as a live sound engineer. After graduation, he plans on obtaining his Masters and Ph.D. in mathematics.

Leaphar Castro will graduate from the University of Central Florida and receive his Bachelors of Science in Electrical Engineering in May of 2016. He worked as an intern at OUC in the EMS/ Transmission planning team. After Graduating Leaphar Castro plans to continue studying in a graduate School to specialize in Power.