The Report Committee for Nathan Jered Hood
Certifies that this is the approved version of the following report:

Design of an Embedded System and Cloud Backend for Remote Monitoring of Smart Traps

APPROVED BY
SUPERVISING COMMITTEE:

Supervisor:

Adnan Aziz

Raj Shah

Design of an Embedded System and Cloud Backend for Remote Monitoring of Smart Traps

by

Nathan Jered Hood, B.S.

Report

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Master of Science in Engineering

The University of Texas at Austin

December 2014

Dedication

I would like to dedicate this report to my parents and to my partner. I appreciate their support and encouragement over the years.

I would also like to thank my parents, Fred and Nicole Hood and Rebecca Hood for each individually demonstrating to me that learning is a life-long process with the ability to open new doors throughout our careers.

Finally, I would like to thank my partner, Steve Jordan, for his patience and understanding over the years as I pursued this degree. I can finally vacate my residency at the Flightpath Coffeehouse and spend some evenings at home!

Acknowledgements

I would like to thank my supervisor, Dr. Adnan Aziz, for encouraging me to pursue a class project out into the world. I would also like to thank him for his support in then folding that work back into the academic setting to support completion of my graduate studies.

I would like to thank Raj Shah from Tech Ranch for agreeing to join my Master's Report Committee and for his ongoing advice and feedback regarding this project.

I would like to thank Kevin Koym from Tech Ranch for his support and for pushing me to leave the lab and talk to customers.

Finally, I would like to thank Alex Gabbi, Joe Forbes, Jackson Salling and Ilsun Sun for working with me on assessing this business venture. I hope to work with them again on future endeavors.

Abstract

Design of an Embedded System and Cloud Backend for Remote Monitoring of Smart Traps

Nathan Jered Hood, M.S.E.

The University of Texas at Austin, 2014

Supervisor: Adnan Aziz

The convergence of low cost cloud services, widespread Internet deployment and low cost SOCs gives rise to systems placing the Internet's vast compute power at the service of simple, everyday devices. Assisted by ubiquitous Wi-Fi deployment and smartphone ownership, a default infrastructure is emerging that supports rapid development of easy to use, low cost, Internet enabled devices. This nascent extension of the Internet into common, everyday devices has been termed the Internet of Things (IoT) and is attracting considerable commercial and academic interest. This paper evaluates the selection and application of IoT technologies to the operations of an existing industry that would benefit from a low cost, remote monitoring system by reducing the cost of delivering their services to their customers. The US pest control industry was selected for analysis as it has a healthy, growing revenue base ($7B in 2013) with a service delivery model that requires expensive manual monitoring (~$45 per inspection) of deployed traps and cages. A prototype system was built entailing a Wi-Fi connected smart rat trap, a

cloud based monitoring system and a smartphone app for associating the trap with a Wi-Fi access point.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

An Internet of Things (IoT) is emerging as Internet connectivity, on board computing resources and cloud-hosted functionality are designed into a broadening array of everyday devices. Market excitement is growing, and IoT is increasingly touted as the biggest technology revolution since the launch of the Internet. The following snapshot of recent forecasts and headlines captures the growing level of excitement in the technology and business press:

- The installed and connected base of IoT units will reach approximately 30 billion in 2020 [1].
- The wearables market will exceed $5 billion in 2014 [2].
- IoT and the resulting interconnectedness will generate $19T in economic value by 2024 [3].

On a daily basis, new IoT ventures are announced, new IoT products are launched and new industry experts join the discussion. Product innovation ranges widely from consumer goods such as the Nest, an Internet connected thermostat for the home, to products for agriculture and industry, such as the ingested, RF-enabled thermometer capsules for remotely monitoring the health of dairy cattle.

1.1 Problem Overview

This project involves the selection and application of IoT technologies to the business operations of the pest control industry. These businesses would benefit from a low cost, remote monitoring system by reducing the cost of delivering their services to their customers. In the pest control industry, the service delivery model consists of an initial installation followed by trap and cage checks conducted every one to two days. In general, this continues until the infestation is eliminated and the equipment is retrieved.

Toward the end of the engagement, it is common that field technicians will inspect an installation and find that no traps or cages required servicing. These unnecessary trips are wasteful of company resources with estimated truck roll expenses totaling approximately $45 per trip. Furthermore, these unnecessary trips needlessly interrupt customers' daytime schedules.

1.2 KEY TECHNOLOGIES

While different combinations of technology are suited to various IoT market segments and use cases, a specific set of technologies is increasing observed in products targeting the home environment. These key technologies were also employed in this project and are as follows:

- Cloud
- Wi-Fi
- Smartphones
- Low cost, advanced semiconductor components

Each of the above technologies offers unique advantages and contributes key functionality when building low cost, feature-rich IoT solutions.

Cloud-based services offer customers easy access to high-availability, redundant, scalable compute resources. These services typically support a wide range of customers and users from various organizations. Therefore, as a competitive differentiator, many companies have invested in building rich ecosystems (large example code base, expert moderated user groups, etc.) to assist customers' product development projects. These shared, cloud-based technologies are also typically offered via monthly subscriptions. By leveraging cloud-based platforms and services, new feature-rich IoT products can be

rapidly developed and launched while minimizing development costs and capital expenditure requirements.

According to a 2013 Gallup poll, 73% of American homes have a Wi-Fi WLAN AP deployed [4]. Although many of the emerging IoT devices do not require the high bandwidth capable via a Wi-Fi connection, its widespread availability makes it an attractive way to wireless connect devices to the Internet. Wi-Fi radio components can be more expensive and power hungry than other options such as Low Energy Bluetooth and ZigBee. However, for many wireless, home-centered use cases, leveraging the commonly available wireless Internet access offered by Wi-Fi yields a net cost savings versus the incremental cost of requiring an additional, customized wireless gateway to link the IoT device to the Internet.

The explosion of smartphone adoption continues to place advanced, mobile computing devices into the pockets of people around the world. The same 2013 Gallup poll cited above regarding Wi-Fi ownership also asked Americans about cellphone ownership and found that 62% of Americans owned a smartphone [4]. The common availability of smartphones allows designers of IoT solutions to consider specifying lower cost, "headless" designs in which keyboards, displays and related features have been stripped from the product. For these headless IoT solutions, including a custom smartphone app as part of the solution allows product designers to offer keyboard- and screen-like functionality without incurring the associated materials costs or footprint of actual keyboards and screens.

Finally, the availability of advanced functionality, low cost semiconductor components enables product designers to develop wirelessly connected IoT products at lower price points than previously possible. While some of these new IoT product might have been technically possible to design in the past, recently released components such as

Texas Instrument's CC3000 Wi-Fi SOC employed in this project allow products to be brought to market at steadily falling prices.

1.3 Summary of Contributions

Leveraging the key technologies outlined above, a prototype system was built comprised of a Wi-Fi connected smart rat trap, a cloud based monitoring system and a smartphone app for associating the trap with a Wi-Fi access point. Building the prototype entailed identifying the problem in the pest control industry, architecting a solution, developing the code across multiple architectural tiers, testing the system and conducting qualitative analysis on the system's performance to ensure it addressed the identified problem.

1.4 Report Overview

This report opens with a discussion in Chapter 2 of how the project topic was identified and refined. Next, Chapter 3 provides an overview of key protocols, tools and technologies used in creating the project. The project's architecture and implementation is then described in Chapter 4. Subsequently, Chapter 5 summarizes system performance across a variety of trials. Finally, the report concludes with Chapter 6 summarizing work completed and provides a plan for recommended next steps.

Chapter 2: Customer Requirements

The objective of this project was to develop a Wi-Fi connected smart rat trap, a cloud based monitoring system and a smartphone app for associating the trap with a Wi-Fi access point. From the beginning, it was a personal goal for the project to be worthy of future commercialization. The exact project topic was originally more general from a market perspective and then became more focused through the work of this project. Selecting a topic with commercialization promise was a step beyond the typical scope of an engineering Master's project. To improve the probability of commercialization success, local startup accelerators and incubators were reviewed and one was selected as a host organization for this project. Attending information sessions, arranging conversations with incubator leaders, interviewing incubator participants and conducting online research informed the decision process. Finally, the pest control industry was selected as the target customer base. To better understand the industry, market surveys were conducted to assess local firms' finances, processes and procedures.

2.1 Incubator Organization Selection

Five preeminent, Austin-based incubator/accelerator organizations participated in a May 2014 Forum hosted by the Rice Alliance titled "Weird and Profitable – A Discussion on Austin's Incredible Incubator and Accelerator Environment." The following organizations were represented in the discussion:

- Austin Technology Incubator (ATI)
- Capital Factory
- DreamIt Ventures
- Incubation Station
- Tech Ranch

All five organizations were assessed as a potential host for this project. During the course of the forum, two organizations were eliminated as hosts based on information presented by their representatives. First, Incubation Station was eliminated from further consideration as their primary focus is on supporting ventures with consumer products such as food, beverages, etc. Next, DreamIt Ventures was removed from further consideration based on their approach for working with ventures. Their methodology focuses on managing and driving their startups via a pre-defined, analytical, metrics-centered evaluation process. At the time of the forum, this project was at a very early stage and required coaching and advising to assist in defining the project's objectives. DreamIt Ventures seemed to be a better fit for ventures that were further along in product development. By the end of the Forum, the pool of candidate organizations had narrowed to ATI, Capital Factory and Tech Ranch.

In the weeks following the Rice Alliance forum, information-gathering conversations were held with representatives from ATI, Capital Factory and Tech Ranch. In the cases of ATI and Capital Factory, both organizations target teams with late stage prototypes or working MVP offerings. Consequently, both organizations were determined to not be viable options as this project was not yet at their targeted level of maturity. In parallel, it became clear that Tech Ranch had a different approach and was focused on building entrepreneurs and assisting them with identifying and developing concepts for new ventures. After participating in a daylong introductory program at Tech Ranch, it was selected as the incubator partner organization for this project.

2.2 TECH RANCH

Tech Ranch positions itself as an entrepreneur training organization. The group offers three different programs. Venture Start is their preliminary offering and offers a

daylong program focused on helping entrepreneurs select and refine a venture concept. Next, Venture Forth is an 8-week boot camp that teaches entrepreneurs about moving from a business concept to a profitable venture. Weekly modules include financial modeling, investor relations, presentation development and refinement, and team selection and recruitment. The final program offered by Tech Ranch is Venture Builder and targets ventures already possessing an initial product and initial customers. Venture Builder is a 26-week program to support entrepreneurs with scaling their product and company and involves allocating a portion of a Tech Ranch Exec-In-Residence's availability to mentor and work with the entrepreneur. While Venture Start and Venture Forth are both pay-to-participate programs, Venture Builder involves an equity assignment from participating ventures. With respect to this project, feedback received by participating in Venture Start and Venture Forth was instrumental in selecting the final topic for investigation.

2.3 PROJECT TOPIC IDENTIFICATION

Preliminary visions of this project involved developing a feature-rich hardware platform with multiple, solution-specific spinoff products. A product concept survey was created and submitted to ~80 friends and family as well as posted to Facebook. Feedback from this initial survey was then used as input to a brainstorming session with the Arrow Electronics Austin team. The resulting list of target use cases is summarized in Table 1.

TARGET USE CASES
Animal Trap and Cage Monitors
Garage Door Monitor
HVAC Filter Monitor
Swimming Pool Monitor
Tank Monitor (Water, Heating oil, propane, etc.)
Temperature/Drought monitor
Water Leak Detector

Table 1: Preliminary Set of Targeted Use Cases

To support the above use cases, a technology platform was specified to include Wi-Fi connectivity, a color/light/proximity sensor, a 3-axis accelerometer, a thermometer and capacitive sense functionality. However, shortly after this product vision was finalized, a competitor named Quirky was identified that made this initial project seem unrealistic as a viable product idea.

Quirky's stated intention was to produce a range of Internet connected, consumer devices for use around the home. By mid-2014, they were already selling an Internet connected device called the Wink Spotter that contained a similar array of sensors as envisioned for this project. The device was being sold via Amazon.com for ~$65 and was intended for detecting motion, sound, light, temperature and humidity. Finally, Quirky had recently accepted a $30M investment from GE and had also been given access to GE's patent portfolio [5]. Taken together, Quirky appeared to be too formidable of a direct competitor and the decision to pivot from a generic, consumer-targeted platform commenced.

In parallel to market developments regarding Quirky, coaching and feedback from Tech Ranch Mentors began to also influence the project trajectory. The first key piece of feedback was that a common source of failure for entrepreneurs is to attempt too many

things at one time. There was a risk that a generic platform attempting to support multiple end products was likely to poorly service multiple markets. Integrating the feedback from Tech Ranch with the decision to avoid direct competition with Quirky in the consumer market, this project narrowed it focus to one use case in the commercial space. The resulting project was TrapSense, a system of remotely monitored smart traps and cages for the pest control industry.

A final key piece of feedback from Tech Ranch mentors was that, in their experience, engineers rush to develop a full solution before engaging customers to validate the market and elicit product requirements. In response to this feedback, another customer requirements survey was conducted, this time with a narrow focus on Austin-based pest control firms.

2.4 Industry Survey Process

Using Yelp and sorting by "Highest Rated" in Austin, Texas, pest control firms were identified and contacted over a period of six weeks in late summer 2014. In total, 22 firms were contacted leading to nine in-depth interviews averaging approximately one hour in duration per interview.

The customer survey involved an in depth review of the firms' business operations. To preserve anonymity, survey results in this report forgo listing the name of the participating firm and each firm's feedback is normalized to be that of an equivalent firm with $1M in annual revenue. Interviews were conducted in participants' offices, in participants' home offices, in coffee shops and in restaurants. Survey participants were either sole proprietors, partial owners or general managers of the firm they represented. The firms included in the survey varied in size from a large, established firm to a small, recently started firm.

2.5 Pest Control Industry - Overview

The results from the survey are summarized in Appendix A. The key findings from the pest control industry are documented in the following sub-sections.

2.5.1 Proportionality of Animal Control Engagements

Among survey participants, the approximate proportional mix among animal control jobs was overwhelmingly skewed toward rat extermination jobs. The proportional mix among classes of animals is shown in Figure 1.

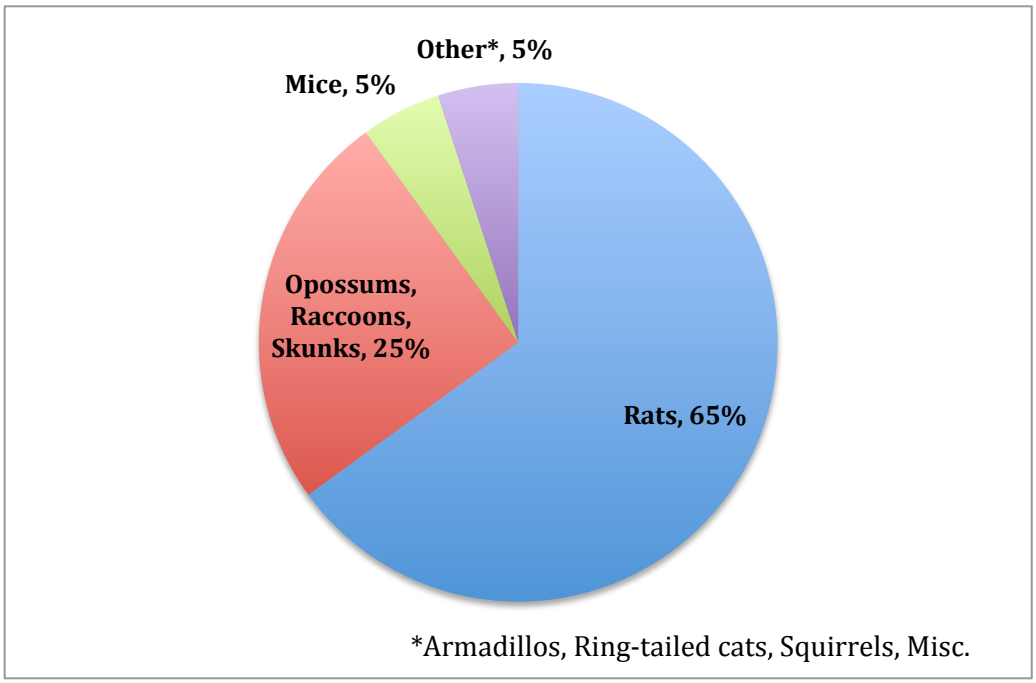

Figure 1: Pest Control Firms' Animal Control Job Mix

2.5.2 Operations Model for Representative Firm

Analyzing responses from participating firms, additional granularity emerged regarding the largest two categories of animal control jobs. Normalizing the results of each firm to that of a firm with $1M in annual revenue, it was determined that on average a pest control firm will execute as follows:

- Complete 31 rat control jobs per month
- Complete 14.7 mid-sized wildlife (raccoon, skunk, opossum) jobs per month
- Waste 1.2 truck rolls per rat control job on unnecessary trap checks
- Waste 0.8 truck rolls per mid-sized wildlife job on unnecessary trap checks
- Waste $42.61 per truck roll when conducting unnecessary trap checks

According to a survey conducted by industry publication, Pest Control Technology, the US pest control industry generated $7.2B in 2013 [6]. Extrapolation of the above results across the US pest control industry indicates an annual loss of $175M due to unnecessary truck rolls for trap and cage status checks.

2.5.3 Snap Trap Preference

Disposability is a key requirement for rat traps. Although a range of rat traps is available on the market, all surveyed firms employed disposable Victor snap traps as shown in Figure 2. Furthermore, seven of nine surveyed firms exclusively employed the disposable Victor snap trap.

Pest control firms typically pair rodent- and insect-control related services. Insecticides can possess a strong odor and are commonly stored in enclosed truck bed storage areas away from the passenger compartment. Rodents are deterred by strong chemical smells. Therefore, rat traps are commonly transported in the passenger cabin to avoid picking up the chemical smells present in the storage areas of trucks. By disposing of rodent traps after use, the pest management professionals avoid bringing the distasteful smells associated with dead rodents into the passenger compartment.

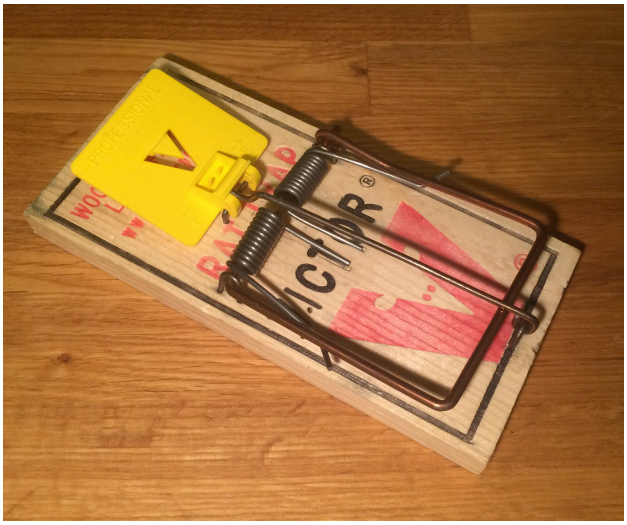

Figure 2: Victor Easy Set Rat Trap

Several firms also use Snap-E traps as they can be easily mounted to vertical surfaces. Shown in Figure 3, the trap is more expensive than the Victor snap trap and is typically washed and reused after a job was complete.

Figure 3: Big Snap-E Rat Trap

2.5.4 Snap Trap Deployments

The number of rat traps deployed per rat remediation job site varied based on surveyed firm's operations model and the nature of the infestation. In summary, reported trap deployments ranged in size from five traps per site to twenty-five traps per site. On average, approximately twelve traps are deployed per site.

2.5.5 Live Trap Preference

Although a range of live trap manufacturers exists, the most commonly employed cages among pest management professionals were the Havahart product line. However, two of the nine firms also reported occasionally employing cages manufactured by Tomahawk, and two additional firms reported exclusively employing Tomahawk cages. From either manufacturer, the large cage was employed for nuisance urban wildlife removal jobs including raccoons, opossums and skunks. The Havahart model shown below is representative of a commonly employed cage.

Figure 4: Havahart Large 1-Door Trap

2.5.6 Live Trap Deployments

Jobs involving the removal of nuisance urban wildlife (opossums, raccoons, skunks) typically involve the placement of multiple live traps. Most survey participants reported typically deploying two to three cages per work site. All responses taken together, the average number of cages deployed was slightly above 2 cages per site.

2.5.7 Environmental Conditions

Differing environmental conditions were reported when comparing rat trap deployments to live trap cage deployments. Rat traps are almost exclusively deployed inside the home. In some rare cases, rat traps are deployed within repurposed bait boxes placed along the exterior perimeter of buildings. However, the live trap cages are deployed both outdoors and indoors and must therefore be able to resist adverse environmental conditions. Finally, the live trap cages are commonly pressure washed after deployments, as captured animals will typically soil the cages in the time period between entrapment and relocation.

2.6 Pest Control Industry - Market Requirements

Starting with the initial vision of a remotely monitored system of smart traps and cages and then incorporating the feedback from the industry survey, the following Market Requirements were identified to guide future TrapSense requirements definition.

2.6.1 Monitoring System

- The smart trap and cage monitoring system should be accessible via smartphone app and web app.
- The smart trap and cage monitoring system should deliver alerts via smartphone app, web app, SMS messages and email messages.

- The smart trap and cage monitoring system should be simple to use with a flexible workflow regarding field technician assignment to customer site.

2.6.2 Smart Traps

- The smart snap traps must be inexpensive, effective and easy to use. Deployments typically involve at least twelve traps, so price is extremely important due to the large fleet size. Internet connection setup must be easy.
- Ideally, the smart snap trap system would support the current industry practice of single-use Victor snap trap deployments. Less optimally, the smart snap trap system could be built around reusable Snap-E traps.

2.6.3 Smart Cages

- The smart cages need to be moderately priced, effective, easy to use and water resistant. Internet connection setup must be easy.
- The smart cages should be developed based on an accepted, industry-leading cage such as the large, 1-door Havahart cage.
- The monitoring system should be able to track smart cage fleet status and ensure the traps are not lost or forgotten at a customer site.

Chapter 3: Review of Technology Standards

3.1 HARDWARE

Texas Instruments offers a range of modular development kits based on their popular microcontroller product families. Leveraging these boards as a foundation, TI has also assembled an array of application-specific BoosterPack boards that snap onto microcontroller development kits to quickly add functionality such as wireless connectivity, environmental sensors, display drivers and motor controllers. Supported by a range of IDE tools and code examples, TI's development boards and BoosterPacks provided an excellent ecosystem to leverage in building this project prototype.

3.1.1 MSP-EXP430FR5739 FRAM Experimenter Board

The MSP-EXP430FR5739 board is designed to allow for easy evaluation of the MSP430FR5739 microcontroller. For ease of use, the board supports a USB-based debugging and programming interface. The board also includes several sensors, including an accelerometer for measuring acceleration, inclination and shock and a thermistor for measuring temperature. Other sources of I/O include two user switches, a reset switch, accessible device pins, eight LEDs and two sockets for adding customized expansion boards or TI BoosterPacks [7].

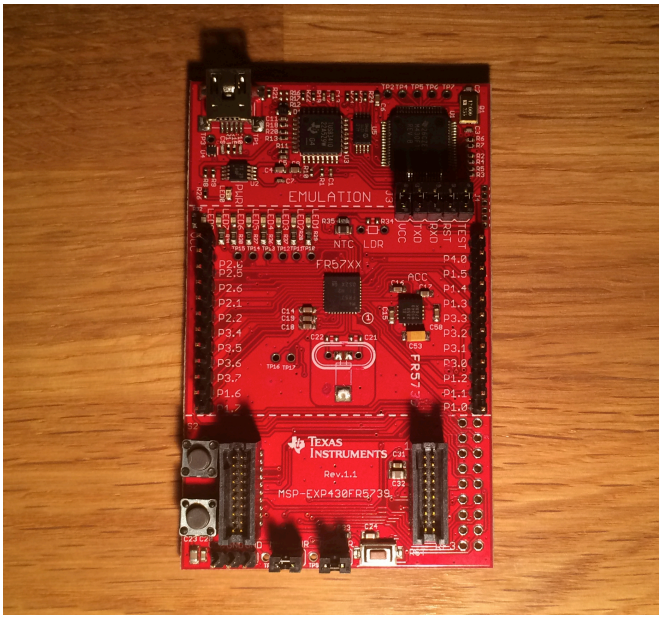

Figure 5: MSP-EXP430FR5739 Board

3.1.2 MSP430 Microcontroller

The MSP430FR5739 is an ultra-low power, 16-bit microcontroller featuring embedded Ferroelectric Random Access Memory (FRAM) nonvolatile memory and a range of peripherals. Key peripherals on the MSP430FR5739 include a 10-bit analog-to-digital converter, a 16-channel comparator with voltage reference generation, three enhanced serial channels capable of supporting I2C, SPI or UART protocols, direct memory access (DMA) controller, a real-time clock (RTC) with calendar and alarm, five 16-bit timers and 32 configurable general purpose input/output (GPIO) pins [8, 9].

The FRAM is a particularly interesting feature of the MSP430FR5739 vs. a more commonly implemented memory solution based on flash memory. According to TI documentation, the FRAM memory cell provides best-in-class memory endurance and can support an amazing 100-trillion read/write cycles (~100 billion times better than flash) rendering the FRAM solution a better fit for remote datalogging implementations. Furthermore, the FRAM offers lower current consumption characteristics than a flash-

based memory system, which makes the FRAM-based solution a better fit for battery-powered projects [8].

3.1.3 CC3000 BoosterPack

The CC3000 BoosterPack allows the functionality of a SimpleLink Wi-Fi CC3000 radio module to be quickly added to the MSP-EXP430FR5739 FRAM Experimenter Board. Together, these two boards create a low cost, turnkey Internet-connected development platform well suited for use as an IoT system endpoint. Beyond the CC3000 radio module, the BoosterPack contains power supply functionality and headers for connecting to the MSP-EXP430FR5739. Depending upon the board configuration, the onboard chip antenna can be employed or an external antenna can be attached using the onboard U.FL RF connector [11, 12].

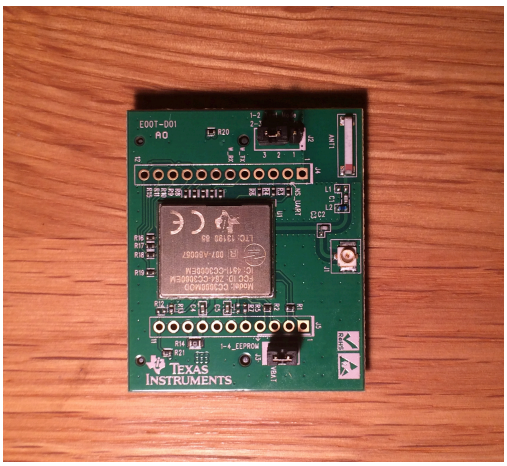

Figure 6: CC3000 BoosterPack Board

3.1.4 CC3000 Wireless Network Processor

The CC3000 is a wireless network processor that simplifies and offloads a majority of the workload related to implementing Internet connectivity. Coupled with a low cost microcontroller such as the MSP430 for handling the balance of protocol

management, a complete, low-cost Internet connectivity solution can be easily developed. The CC3000 incorporates an IEEE 802.11 b/g radio, modem and MAC. It supports WLAN communication in the 2.4-GHz ISM band and can store up to seven user-configurable connection profiles. Leveraging onboard security accelerators, the CC3000 supports all Wi-Fi security modes for personal networks including WEP, WPA and WPA2. The CC3000 integrates a network stack including IPv4 TCP/IP with BSD socket APIs allowing connected microcontrollers to easily create simple Internet connections. Up to four TCP or UDP sockets can be simultaneously supported by the CC3000. Onboard support for network protocols includes ARP, ICMP, DHCP-client and DNS-client to support connection to the local network and the Internet. An EEPROM can be used to store user information as well as firmware updates. Finally, the CC3000 supports firmware updates delivered both over the air and via wired connection [9].

3.2 ENABLING TECHNOLOGIES

3.2.1 Wi-Fi

Wi-Fi is a trademark developed to refer to wireless local area networks (WLAN) based on IEEE 802.11 standards. The Wi-Fi certification process ensures that products support the 802.11 specifications [10]. Of particular interest to the emerging IoT market, products that are newly Wi-Fi certified can be assured of interoperability via standards-based communications, of backwards compatibility with other Wi-Fi certified products and of supporting advanced security including WPA2. Interoperability, legacy support and security are key benefits as IoT solutions begin to enter markets with longer service life expectancies than those typically encountered in consumer electronics industries [11].

Wi-Fi connections are established between IEEE 802.11 compliant wireless network interface cards (WNIC) operating in one of the following modes: Master, Managed, Ad hoc, Mesh, Repeater and Monitor mode. Of interest to this project are Master, Managed and Monitor modes. A WNIC operating in Master mode provides an access point (AP) to the Internet for cards operating in the Managed mode and is commonly referred to as a Wireless Local Area Network Access Point (WLAN AP). This combination of cards running in Master and Managed modes can be commonly observed anytime a laptop (WNIC in Managed mode) wirelessly accesses the Internet via a WLAN AP (WNIC in Master mode). The third mode of interest, Monitor mode, warrants discussion as a WNIC in monitor mode captures packets from a channel without being associated with a broadcasting WLAN AP [16-18]. In effect, a WNIC in Monitor mode resembles a third party listening into a party-line telephone conversation – they might not understand what is being discussed, but they can hear the conversation.

Although the CC3000 is impressive, it does have some shortcomings. For example, it can support WPA-Personal and WPA2-Personal security, yet is not able to support WPA-Enterprise or WPA2-Enterprise security. To place this into context, note that the restricted.utexas.edu WLAN network requires hosts to connect with NICs supporting WPA-Enterprise or WPA2-Enterprise [12]. Accordingly, all UT-Austin campus demonstrations as well as some field demonstrations to pest control firm representatives occurred using an AT&T cellular network connected WLAN AP. The cellular connected WLAN AP used with this project was an AirCard 770S manufactured by Netgear and can be seen in Figure 7.

Figure 7: AT&T Cellular Wi-Fi Hotspot

3.2.2 SmartConfig

In many cases, IoT products are being developed as "headless products", a phrase that denotes products designed for use without attached monitors, displays, keyboard, computer mice or other standard input and output components. Product designers are opting for headless design for a range of reasons including lowering production costs, increasing products' environmental conditions resistance and allowing products to occupy smaller physical footprints. These headless products are typically accessed over an Internet connection using a web app or smartphone app. However, a key challenge that emerges for wireless, headless designs is how to initially program the device with the appropriate WLAN AP information required to establish a connection.

To address the problem of provisioning headless devices, Texas Instruments developed the SmartConfig technology to support provisioning CC3000-based, Wi-Fi enabled products. Using only a WLAN AP and a SmartConfig-enabled application running on a smartphone, tablet or computer, SmartConfig allows the CC3000 to be wirelessly programmed with the SSID and password information required to subsequently establish a WLAN connection. Furthermore, the SmartConfig process can

be used to simultaneously provision multiple CC3000 devices onto the same WLAN AP [20, 21].

SmartConfig combines two aspects of current networking protocols to communicate information to the CC3000 without the device being associated to the WLAN AP. First, when SmartConfig is started, the CC3000 is placed into the Wi-Fi Monitor mode previously described in this report. This mode allows the CC3000 to receive and monitor packets, even though the CC3000 is not associated to the WLAN AP. Second, SmartConfig leverages the fact that UDP packets are permitted to be of varying length. The SmartConfig application is installed on a smartphone or computer currently associated to the WLAN AP. The SmartConfig application encodes the WLAN SSID and password into UDP packets by modulating the packet length. Although TI has declined to formally publish the SmartConfig encoding algorithm, members of the technical community have taken issue with this attempt at "security through obscurity" and have reproduced and published the algorithm. In short, the encoding algorithm breaks SSID and password information into half bytes, combines each half byte with sequencing information, and then encodes that information into the lengths of UDP packets. Additional UPD packets of pre-determined, specific length are inserted into the data stream as ciphertext message delimiters [13].

SmartConfig does support AES encryption of the SSID and password prior to broadcasting the UDP packets. However, for this technology to be enabled, the CC3000 must be pre-programmed with an AES Security Key that is then also entered into the SmartConfig application. An example scenario would involve the device receiving a unique ID during its manufacture. The unique ID would also be printed on a sticker affixed to the device. Later, when the customer was using SmartConfig to provision the device onto its destination WLAN AP, the unique ID would need to be entered into the

SmartConfig application and its AES functionality would need to be activated before the provisioning process began.

The SmartConfig application is provided by TI as an iOS app, an Android App and a Java application for use on a network-connected computer. Developers can also implement their own customized application with SmartConfig functionality using 6 API calls into TI's code libraries [14]. For this project, the iOS SmartConfig application was downloaded from the iTunes App Store and installed onto an iPhone 5S. A screenshot of the iOS SmartConfig app can be seen in Figure 8.

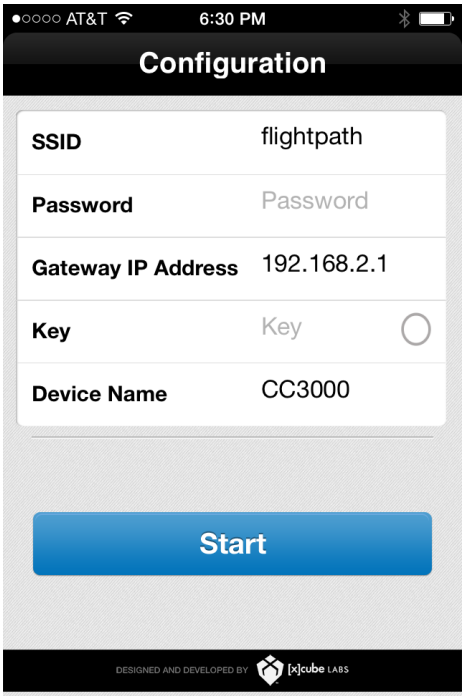

Figure 8: SmartConfig iOS App

3.3 PLATFORMS AND APIS

A key component of IoT's compelling value proposition lies in the ability to quickly and easily add system functionality by leveraging pre-existing cloud-based APIs

and Platforms. Three key cloud-based products were leveraged in the creation of this project.

3.3.1 Google App Engine Platform

Google App Engine (GAE) is a platform as a service that lets you create and run your applications using Google's proprietary infrastructure. GAE supports development in four languages: Java, Python, PHP and Go. Leveraging Google's infrastructure, applications running on GAE can automatically scale and load balance to support large computational loads running with large datasets. The platform supports persistent storage with a full set of supporting functionality to enable queries, sorting and transactional integrity. GAE also supports asynchronous task queues and scheduled task execution. Finally, GAE includes integration with other Google cloud services and APIs such as Google Compute Engine, Google Cloud Storage and Google BigQuery [23, 24].

3.3.2 Exosite Platform

Exosite is a cloud-based Internet of Things platform as a service providing a full range of services related to managing a device fleet, analyzing device fleet data, pushing device firmware upgrades and providing services related to end user account management. Exosite's platform allows for developers to write scripts in Lua to parse device data, extend platform functionality and create customized solutions. Finally, Exosite supports integration with third party business systems such as Oracle, SAP and Salesforce.com [25, 26].

3.3.3 Twilio SMS API

Twilio is a cloud-based API for enabling voice, multimedia and text based communications. Twilio exposes a RESTful API allowing for easy integration of its functionality into users' projects. This project leverages the SMS capabilities of Twilio's

API. For customers requiring SMS functionality, the key value provided by Twilio is that it works directly with 1,800 carriers around the world to ensure SMS message delivery and to achieve nearly global coverage [15].

3.4 IDEs, Tools and Demos

3.4.1 Code Composer Studio 5.5.0

Texas Instruments offers a variety of IDEs for working with their product lines. For this project, a free 90-day version of Code Composer Studio (CCS) version 5.5.0 was installed and used for working with firmware on the MSP430 and CC3000. CCS is a Texas Instruments proprietary product based on Eclipse and includes a suite of embedded processor specific tools used to develop and debug embedded applications [16].

3.4.2 Eclipse 4.3

Eclipse is a popular, open source IDE for developing a wide range of applications. This project used Eclipse 4.3.0.I20130605-2000 running Java SE 6 (version 1.6.0_51-b11-457) and supplemented with Google Plugin for Eclipse 3.4.2v201310081834-rel-43. The Google Plugin contained tools for working with Google APIs and Services and allowed for quick and easy deployment of web applications onto Google App Engine platform [17].

3.4.3 TeraTerm 4.75

TeraTerm is an open source, free, terminal emulator program for Microsoft Windows environment. The program supports serial port connections, TCP/IP connections and IPv6 communications and was to debug the root cause of observed communications breakdowns between the CC3000 and the Exosite cloud [18].

3.4.4 CC3000 Patch Programmer

Texas Instruments provides a patch programmer for updating the drivers and firmware on the CC3000 and instructs developers to perform these updates upon receiving newly purchased boards. The patch programmer first reads the CC3000 EEPROM and stores device specific information such as MAC address. Then, the programmer is used to overwrite the CC3000 EEPROM with the updated drivers, the updated firmware and then finally restores device specific information [19].

3.4.5 Exosite/TI Demonstration Code

Texas Instruments and Exosite have partnered to produce several TI/Exosite demonstrations. Beyond the CC3000/MSP430 demonstration leveraged in this project, the companies have also launched a demonstration project based on TI's next generation Wi-Fi radio, CC3200, as well as a third demonstration based on TI's TM4C1294, an Ethernet-connected microcontroller.

The Exosite demonstration code for the CC3000/MSP430 is comprised of two parts with one part consisting of an embedded application and the other part being a cloud based application. The embedded application is firmware for the CC3000/MSP430 development boards. This Exosite code leverages TI code libraries and provides a reference implementation for connecting to the Exosite cloud. The firmware reads data from various sensors and I/O ports on the development boards, manages the Wi-Fi connection and exchanges data with the Exosite cloud. The second key part of the Exosite demo is a cloud-based, device portal for reviewing graphical representations of data being uploaded from the CC3000/MSP430 development boards. The device portal also includes a scripting area for users to develop customized Lua scripts that extend the portal's default functionality.

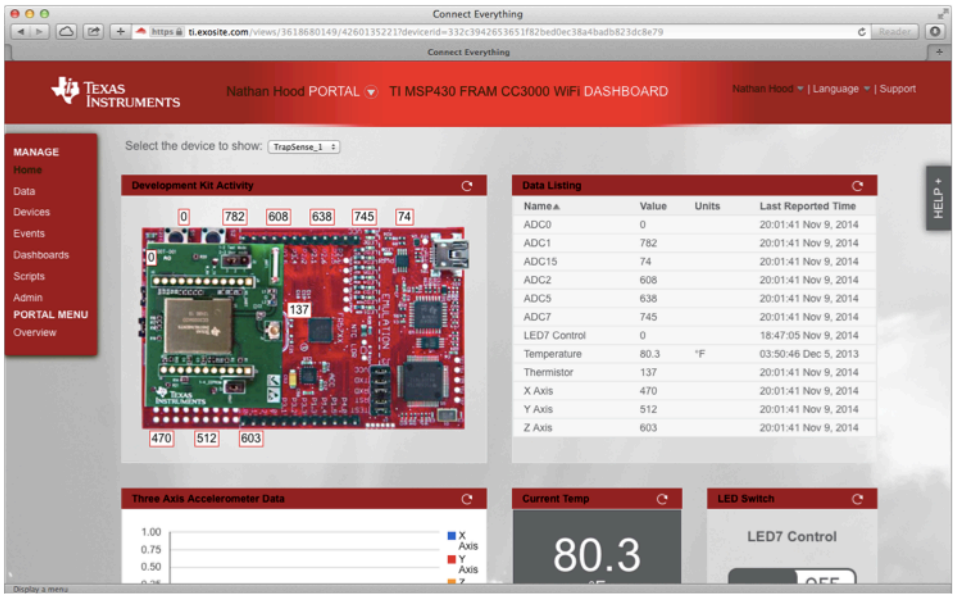

Figure 9: Screenshot of Exosite Platform

Chapter 4: System Architecture

The TrapSense system crosses multiple domains including electronics hardware, firmware, smartphone app and web app. A majority of the effort involved in developing the solution was in the area of technology integration. In the end, a working solution was developed which included a slice through all relevant technologies with opportunity for improvement at each tier of the solution.

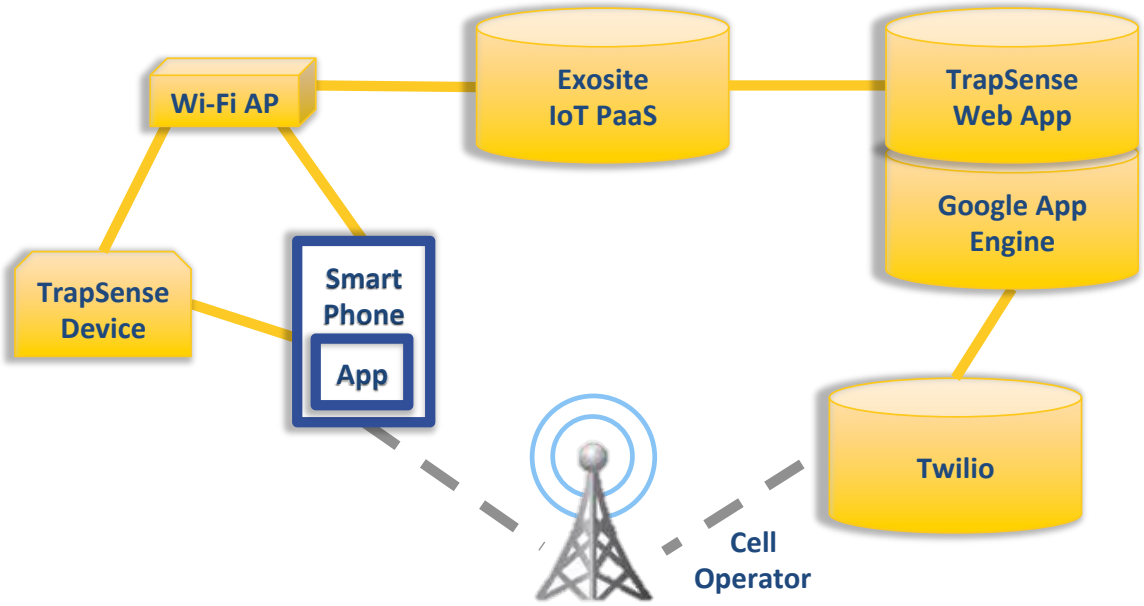

Figure 10: TrapSense Architecture

4.1 TrapSense Hardware

As a first order of business, significant effort was invested in component research and selection for developing the hardware portion of TrapSense. Work on previous embedded systems projects had highlighted complexities related to wireless connectivity based on IEEE 802.15.4 (specifically ZigBee and XBee). Accordingly, for this project, attention was directed toward Wi-Fi based solutions. After surveying available solutions,

the CC3000 was selected due to its innovative provisioning technology, SmartConfig. In support of the CC3000, the MSP430 microcontroller was next selected due to the large ecosystem surrounding this popular microcontroller as well as the extensive sample code base involving the pairing of CC3000 and MSP430. In the end, the hardware side of this project leverages development boards and code libraries from Texas Instruments, sample firmware from Exosite and a smartphone app from Texas Instruments.

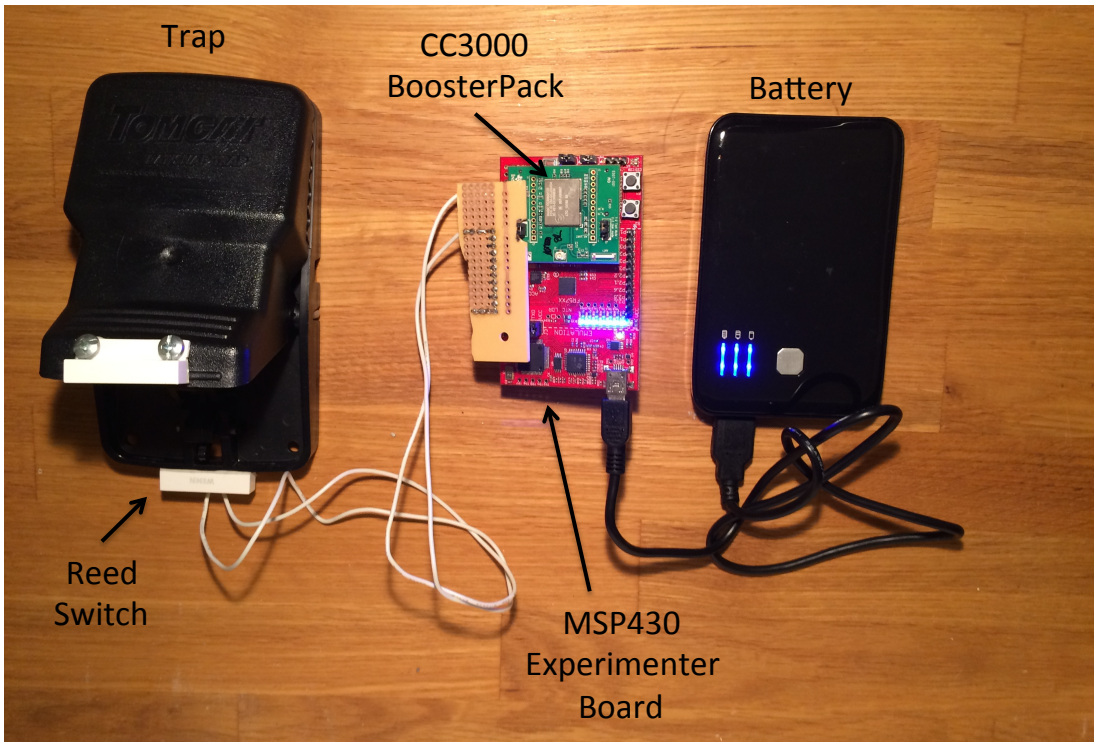

Figure 11: TrapSense Hardware

After updating the CC3000 drivers and firmware using the CC3000 Patch Programmer, the CC3000/MSP430 sample code was downloaded from Exosite's GitHub repository and then installed on the MSP430. In its default configuration, the Exosite sample code reads data from Analog-to-Digital Converter (ADC) pins, the tri-axial accelerometer and the thermistor on the MSP430 experimenter board. The example code

from Exosite uses HTTP POSTs to write data from the MSP430 experimenter board over a Wi-Fi connection into the Exosite cloud. As provided from Exosite, the code sends updates to the cloud approximately every 20 seconds. As part of the work on this project, the MSP430 example code was modified to decrease the delay between updates.

A reed switch is characterized by changing state in the presence of a magnetic field and is commonly used to instrument doors and windows as part of domestic alarm systems. As shown in Figure 11, a reed switch was attached to the rat trap with its contacts soldered between MSP430 ADC P1.0 and ground. When the rat trap was armed, the reed switch electrically closed the circuit. By closing the circuit, the P1.0 input pin was tied to ground and the ADC recorded 0 volts measurements. When the rat trap was triggered, the reed switch electrically opened the circuit and the input pin was allowed to electrically float.

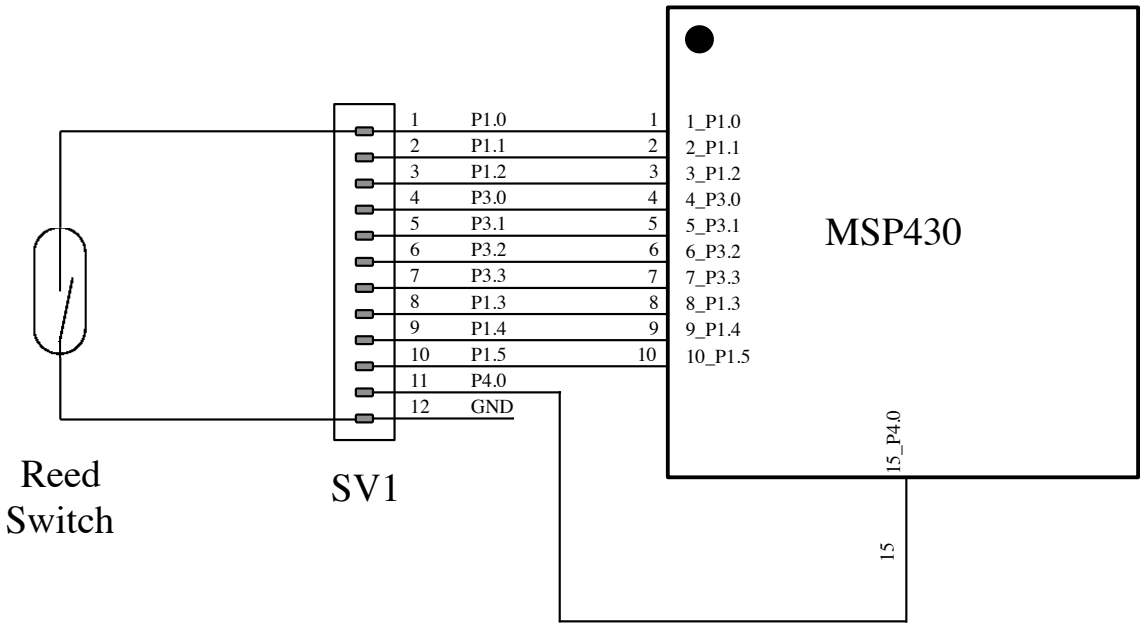

Figure 12: TrapSense Reed Switch Connection to MSP430

4.1.1 Costs

The materials used to build TrapSense were purchased from a range of sources including Texas Instruments, Fry's and Amazon.com. The materials costs are summarized in Table 2. Note: in the case of the reed switch and PCB, representative costs and manufacturers are recorded for these commodity parts.

Manufacturer	Part	Costs
TI	MSP-EXP430FR5739	$35.70
TI	CC3000 BoosterPack	$35.40
Golden Valley	5000 mAh External Rechargeable Battery	$17.99
Serco	ABS Enclosure	$11.99
Tomcat	Rat Snap Trap	$5.37
Philmore	PCB Protoboard*	$4.99
CES	Reed Switch and Magnet Assembly*	$3.50
	Total	$114.94

Table 2: TrapSense Project Materials Costs

4.2 TRAPSENSE SOFTWARE

The TrapSense software is comprised of four main software components. As shown in Figure 15, firmware running on the CC3000/MPS430 boards communicates with the TrapSense Exosite script. In turn, the TrapSense Exosite script communicates with the TrapSense GAE web app. Finally, not shown in Figure 15, the SmartConfig smartphone app communicates with the CC3000/MSP430 firmware during the initial step of connecting the trap to a WLAN AP.

4.2.1 CC3000/MSP430 Tier

Exosite sample code was used for CC3000 and MSP430 firmware without significant modifications. The code was written in C and was viewable, editable and recompilable using Code Composer Studio. The code consists of a looping main method controlled by a timer. When the timer expires, the code checks for a Wi-Fi connection

and attempts to reestablish a lost Wi-Fi connection if necessary. Once a Wi-Fi connection is established, the code gathers and posts sensor data from the development boards to the Exosite cloud. The demo code posts the board sensor data by building an HTTP POST request with name-value pairs storing the name and measured state of each relevant sensor and input pin. The only modification made for this project to the firmware was to increase the main loop update frequency. The firmware was initially configured to post updates to the Exosite cloud every 20 seconds. The firmware was modified to attempt an update every 5 seconds. However, as reviewed later in the report, the system was never quite able to post updates to the Exosite cloud at this aggressive frequency.

4.2.2 Exosite Tier

The Exosite demo code was chosen to provide reliable connectivity between the CC3000/MSP430 boards and the TrapSense GAE web app. Within the Exosite cloud, a script was written in Lua to analyze and respond to the data reported from MSP430 ADC1, the port connected to the reed switch on the rat trap. The Lua script was written as a looping main method that initiates a new execution loop upon receiving new board data or upon exceeding a two-minute timer. The script includes functionality to detect and respond to the following events:

- Triggered trap
- Losing communications between trap and Exosite
- Establishing communications between trap and Exosite

For each of the above events, an HTTP POST request is sent to the TrapSense web app hosted on GAE. The update type is encoded in a simple status name-value pair within the query portion of the URL. Finally, the Lua script includes basic functionality to debounce reported events and prevent unnecessary requests from being posted to the TrapSense

web app. This debounce functionality was required to provide a good user experience and to ensure that the Exosite account daily HTTP transaction limits are not exceeded.

4.2.3 Google App Engine Tier

The TrapSense GAE web app provides simple System On/Off functionality and stores basic user contact information. The web app consists of two simple screens and was implemented using Java, servlets, JSPs, Bootstrap and jQuery.

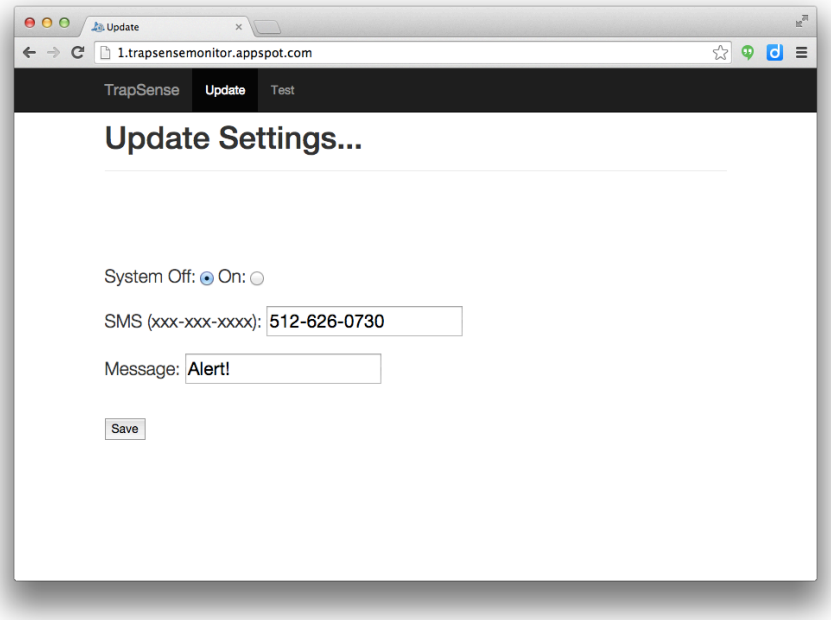

Figure 13: TrapSense Main Screen

Within the TrapSense GAE web app, a servlet receives and parses HTTP requests from the Exosite tier. The web app then interprets the status updates and takes appropriate action regarding communicating status changes to the user. Upon receipt of an alert from Exosite, if the TrapSense system is activated, then an appropriate SMS message is sent per user preferences via Twilio.

To support SMS communications, Twilio was incorporated into the TrapSense GAE web app. First, a Twilio account was setup. Next, the Twilio jar, twilio-java-sdk-3.3.16-with-dependencies.jar, was downloaded and added to the Eclipse project. Finally, the SMS functionality was added into TrapSense using fewer than a dozen lines of code.

A secondary TrapSense screen was implemented to host two basic system debug functions. The first function, "Test Twilio/TrapSense Connection" exercises the linkage between the TrapSense GAE web app and Twilio. The second function, "Emulate Triggered TrapSense Monitor", tests the full application by stimulating TrapSense as if it had received a triggered trap update.

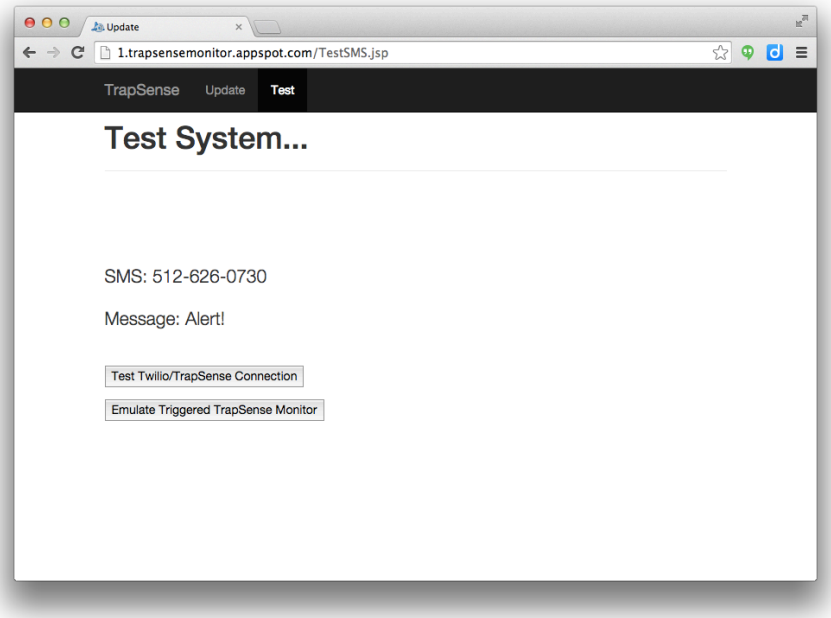

Figure 14: TrapSense System Test Page

4.2.4 Smartphone App

The Texas Instruments SmartConfig iPhone app was used to configure the TrapSense device with the WLAN AP SSID and password information. No changes were required for the smartphone app to support this project.

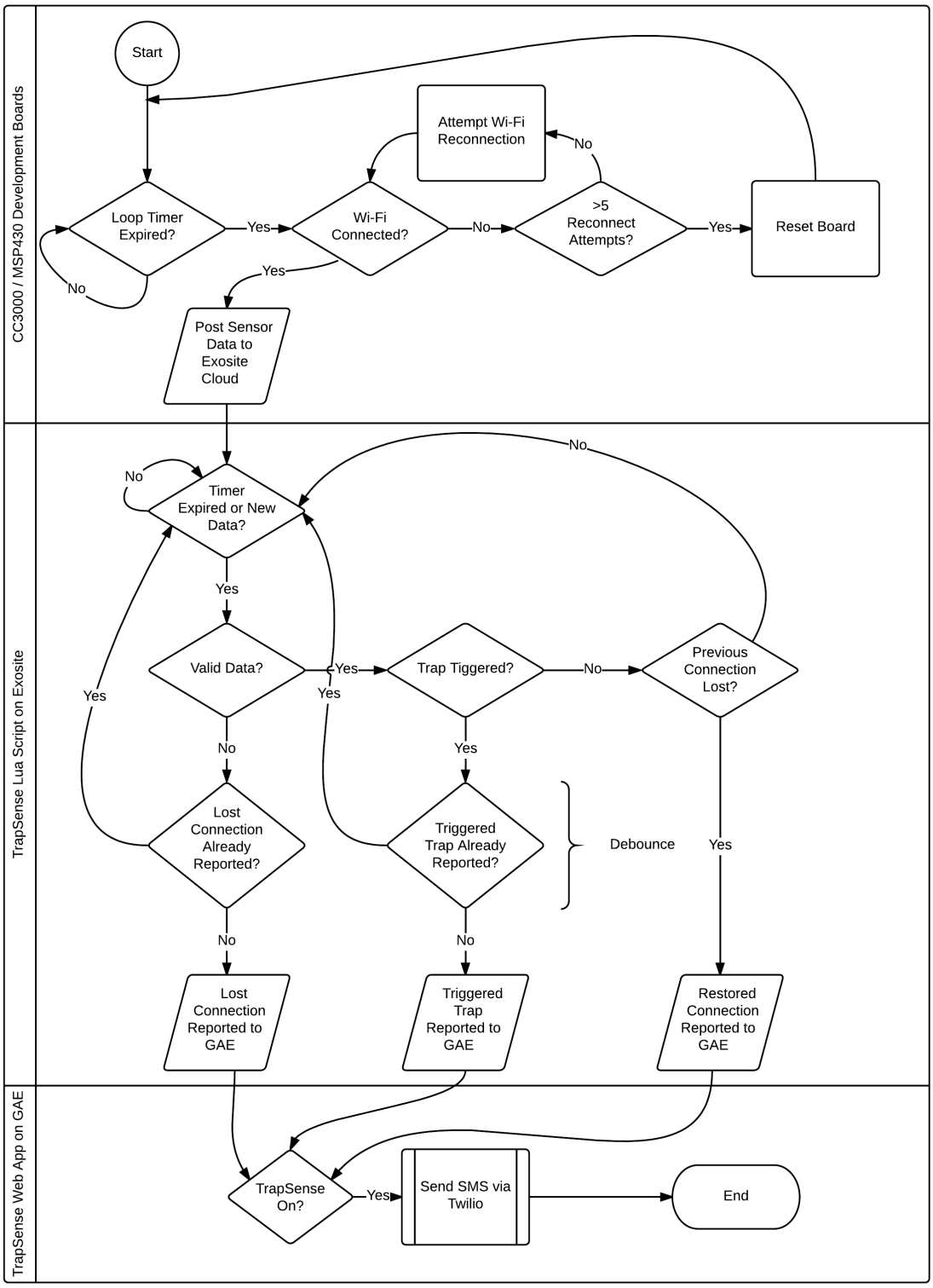

Figure 15: TrapSense System Flowchart

Chapter 5: Results

5.1 Quantitative Results

During demonstrations of the TrapSense system, varying response times were observed when measuring the time between trap discharge and SMS alert delivery. In an attempt to improve update system response time, firmware timing was modified on the CC3000/MSP430 boards to increase the update frequency from every twenty seconds to every five seconds. However, the system was not able to accomplish an update every five seconds, even while using a dedicated WLAN AP. Furthermore, a consistent update frequency remained elusive with the period between updates varying from ~15 seconds to ~3 minutes as measured from trap triggering event through SMS delivery.

As a controlled experiment, 10 trials were conducted in which an iPhone 5s timer app was used to measure the time elapsed between triggered trap and the receipt of an SMS alert. The results of these trials are seen in Table 3.

Trial	Time (sec)
1	9.83
2	12.08
3	12.93
4	17.03
5	11.68
6	7.86
7	10.34
8	10.18
9	11.08
10	57.51
Average	16.05
Std Dev	14.76

Table 3: TrapSense System Response Time

However, the results of these trials illustrate the variation in system responsiveness noted anecdotally during demonstrations. Before the last trial, results from this test were in the range of 9 to 12 seconds with outliers at 7 and 17 seconds. However, during the last trial, the system took nearly one minute to alert the user of the triggered trap. To investigate the source of variability, latency measurements were made on each of the major tiers of the TrapSense system.

5.1.1 Latency Study: Google App Engine to SMS Delivery

As previously noted, the TrapSense GAE web app has two system test functions. The most comprehensive system test spoofs the web app as if it had received an update from a triggered TrapSense device. Using this system test function and an iPhone5S timer app, the latency between stimulation of TrapSense GAE web app and SMS delivery was measured over 10 trials with results summarized in Table 4.

On average, it took 4.19 seconds from receiving an alert into the TrapSense GAE web app, through parsing the alert and sending an appropriate message from TrapSense to Twilio, and finally for Twilio to send an SMS through the cellular network to a mobile phone.

Trial	Time (sec)
1	4.14
2	4.54
3	4.69
4	3.44
5	3.54
6	3.96
7	5.04
8	4.49
9	4.39
10	3.66
Average	4.19
Std Dev	0.53

Table 4: Latency from TrapSense GAE Web App through SMS Delivery

5.1.2 Latency Study: Exosite to Google App Engine

The lack of good, cross platform logging capabilities complicated measuring the communications latency between the TrapSense Exosite script and the TrapSense GAE web app. To estimate the latency between the arrival of an event into Exosite and the arrival of an alert message into the TrapSense GAE web app, the study progressed over two steps.

First, a system test Lua script was written on the Exosite platform to dispatch an alert via HTTP POST in the same manner that the TrapSense Exosite script dispatches its triggered trap alert. The time from Exosite stimulation through TrapSense web app reaction and on through SMS delivery was then recorded. Ten trials were recorded in which the test script was executed on the Exosite tier and an iPhone 5S timer app was used to measure elapsed time between Exosite platform stimulation and SMS delivery. The results of the ten trials are summarized in Table 5.

Trial	Time (sec)
1	11.53
2	8.13
3	6.86
4	8.86
5	10.01
6	8.83
7	9.11
8	9.74
9	7.96
10	7.29
Average	8.83
Std Dev	1.38

Table 5: Latency from Exosite through SMS Delivery

Second, the latency between TrapSense Exosite script stimulation and TrapSense GAE web app stimulation was estimated. Starting with the average latency measured in this study (stimulated TrapSense Exosite script through SMS delivery; average of 8.83 sec per Table 5) and subtracting the average latency measured in the previous study (stimulated TrapSense GAE web app through SMS delivery; 4.19 sec per Table 4). The result is an estimated latency of 4.6 sec from stimulation of the TrapSense Exosite script to the stimulation of the TrapSense GAE web app.

5.1.3 Latency Study: Trap to Exosite (Dedicated WLAN)

A study was conducted over the course of sixty minutes to determine how often the TrapSense device (CC3000/MSP430 boards) was able to report trap status to the Exosite cloud. The test was conducted using a dedicated WLAN AP from AT&T, the Netgear AirCard 770S. The trap was turned on and allowed to operate for thirty minutes in the armed state. Next, the trap was triggered and allowed to operate for another thirty minutes in the discharged state. Throughout the sixty-minute trial, the CC3000/MSP430

boards were able to send an update over Wi-Fi to the Exosite platform 349 times, with an average period of 13.9 seconds between updates. However, the results included several outlier data points with periods as large as 183 seconds. As seen in Figure 16, the statistical mode of 8 seconds appears to be more representative of system performance.

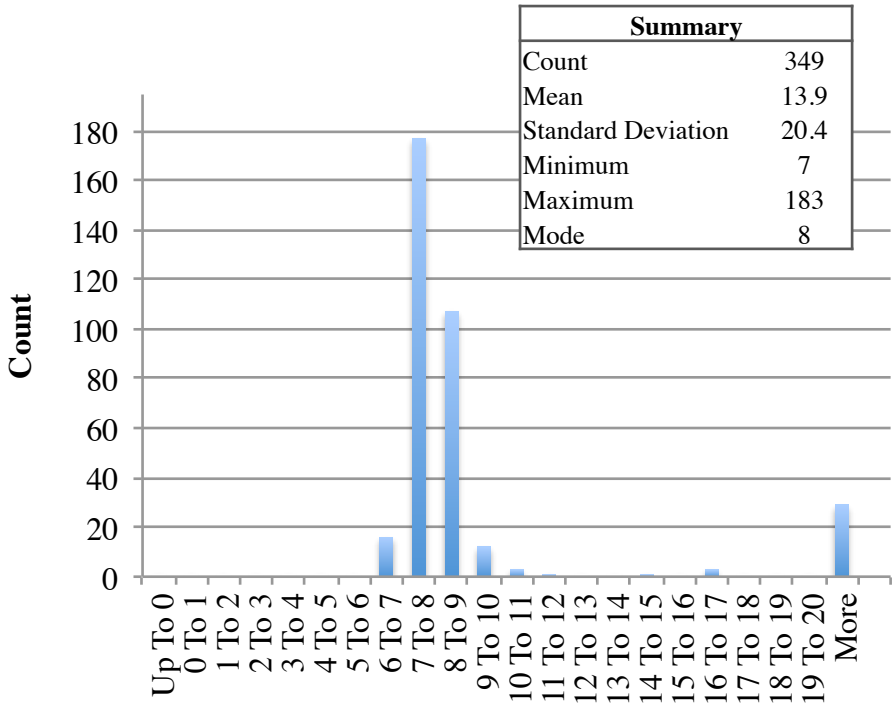

Figure 16: Update Period between Trap and Exosite (Dedicated WLAN AP)

5.1.4 Latency Study: Trap to Exosite (Coffee Shop WLAN)

To further explore latency in communications between the CC3000/MSP430 boards and Exosite, the TrapSense device was next associated to the WLAN AP in a busy coffee shop. For this test, there were approximately 50 patrons in the coffee shop working on their laptops. An unknown portion of these patrons also had smartphones connected to the WLAN AP.

The TrapSense device performance was considerably degraded when using the coffee shop's WLAN AP. It is not clear if this degradation was a result of the CC3000/MSP430 boards communicating via a heavily utilized WLAN AP or if there was something intrinsic to the coffee shop WLAN AP and network gear configuration that caused a conflict. While the trial reported in Figure 16 used a dedicated WLAN AP, the tests were conducted in the same coffee shop as data gathered in the trial using the coffee shop AP, therefore removing overall ambient wireless traffic as a root cause for the degraded performance. Again, the trap was allowed to operate for sixty minutes with the first half of the trial conducted with the trap armed and the second half of the trial conducted after the trap had been triggered. The average period of time between successful updates was 41.2 seconds and the trap was only able to connect 86 times to Exosite over the course of the 60-minute period.

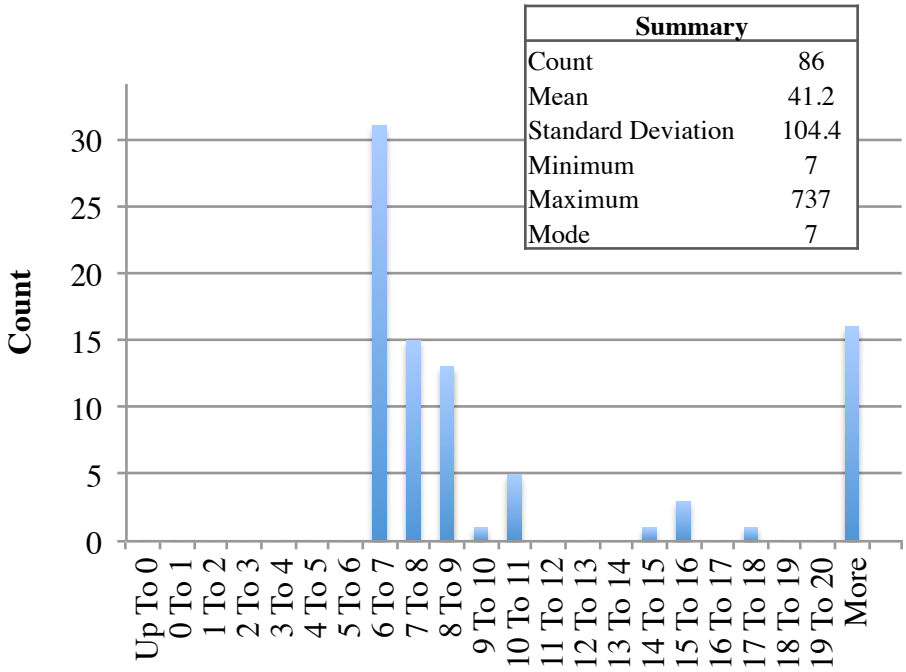

Figure 17: Update Period between Trap and Exosite (Coffee Shop WLAN AP)

5.1.5 Summarized Latency Stack

Taken together, the above TrapSense system studies begin to illustrate how latency stacks up across the various architectural tiers. As reported in Table 3, the overall average system response time is approximately 16 seconds between a trap being triggered and an alert SMS being delivered to a users phone. Trials recorded in Figure 16 revealed that on average the TrapSense traps were able to send an update to the Exosite platform approximately every 14 seconds when using a dedicated WLAN AP.

As illustrated in Figure 15, updates from the trap to the Exosite platform are sent based on expiration of a loop timer and Wi-Fi connection availability. To understand the overall system latency, it is important to emphasize that alerts are not sent as a result of the trap discharging but are sent as a result of a timer expiring. So, within the average update period of 14 seconds, the trap is equally likely to be triggered at anytime throughout the update period. Therefore, on average, it takes approximately 7 seconds for a periodically generated status report to report a triggered trap to the Exosite tier.

From the Exosite tier, data summarized in Table 5 indicates that it would take approximately 9 seconds for an alert to propagate from Exosite to a delivered SMS message. Within those 9 seconds of latency, data summarized in Table 4 indicates that it takes approximately 4 seconds for the alert to propagate from Google App Engine, through Twilio and then be delivered as an SMS to a cell phone. Therefore, a balance of approximately 5 seconds of latency remains to be assigned to the time taken for alerts to propagate from Exosite to Google App Engine.

Taken together, the latencies between architectural tiers are summarized into a stack up as shown in Figure 18.

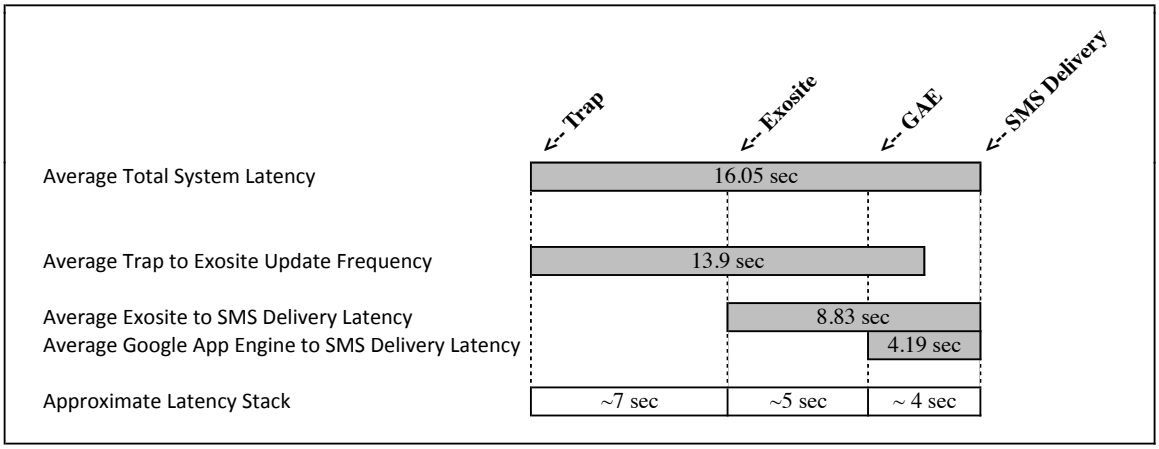

Figure 18: TrapSense Latency Stack

In the end, a typical system response time of ~16 seconds with outlier data points of several minutes is entirely satisfactory for reporting a triggered rat trap. The current manual process of checking traps does not typically manage to assess a trap status more frequently than once every 24 hours. So, this project's performance would offer a marked improvement in notification timeliness as well as offer the key benefit of not requiring truck rolls to check customer sites.

5.2 Qualitative Results

This following Qualitative Results section will summarize findings that would prove useful in converting this project into a commercial product.

5.2.1 System Observations

- Although the Texas Instruments CC3000 was innovative at the time this project was under active development, the next version of the part was recently launched with the name CC3200 and should be adopted as the platform for future work. The CC3200 offers more functionality than the CC3000 at a lower price point. For example, the CC3200 integrates a user programmable ARM-core

microcontroller, so the MSP430 used in this project could be cut out of a future, CC3200-based design yielding an additional cost reduction. The CC3200 also can operate in Master mode allowing it to be a WLAN AP for any other Wi-Fi enabled device operating in Managed mode. Finally, the CC3200 supports more advanced security such as WPA-enterprise, WPA2-enterprise and TLS/SSL.

- Currently, the firmware running on the CC3000/MSP430 boards sends periodic updates to the Exosite cloud. A script in the Exosite cloud interprets the data and determines what action is required. The advantage of this arrangement is that development and debug are much simpler in the cloud vs. in an embedded system. However, the disadvantage is that battery power is consumed rapidly by the system's ongoing use of the radio – by far the most current intensive function offered by the boards. Any implementation of this product would require that data analysis be pushed down to the CC3000/MSP430 boards. Under this arrangement, a timer would wake the MSP430 and periodically check the status of the trap. Only after a triggered trap is detected, would the radio be activated for sending an update. Regardless of triggered trap status, it would also send a periodic health update to the cloud, perhaps daily, to report that the trap is still operational and actively deployed.
- The trap for this project was selected because its design was readily instrumented with a reed switch. However, after completing the industry survey, it is clear that professionals more commonly use the Victor snap trap. A better design for this product would be in the form of a sleeve that fits around the non-baited end of a Victor Rat Trap as see in Figure 19. Ideally, this design would employ an accelerometer to detect the shock of the discharging trap. After the trap had killed the rat, the trap and rat could simply be slipped out of the sleeve and discarded by

a field technician. Then, the TrapSense sleeve would be reloaded with a fresh Victor Rat Trap and returned to service.

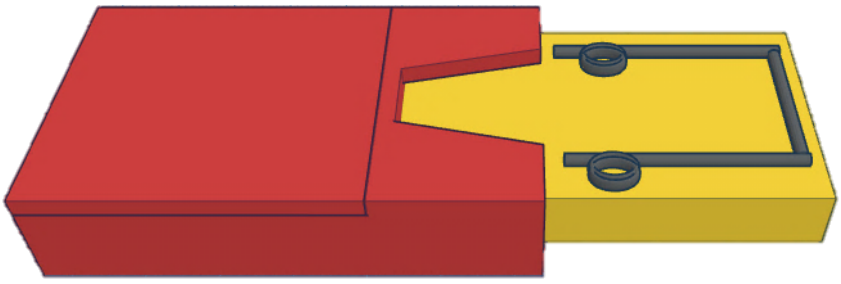

Figure 19: Instrumentation Sleeve (Red) for Disposable Rat Trap (Yellow)

- The TrapSense project currently employs the Exosite Tier as an IoT platform. For near term product development efforts, time-to-market would be reduced by using a third party IoT platform like Exosite to manage the fleet of deployed devices. Leveraging a preexisting, third party IoT platform would be especially valuable in executing more complicated functions such as over the air firmware upgrades. However, over the longer term, it might make sense to replace the third-party service with an internally developed service to allow for recovering the monthly service fees required by the third-party platform. For example, Exosite charges up to $2/device/month for IoT device management fees.
- Beyond the CC3000/MSP430 upgrade to CC3200, the overall hardware costs are too high due to their usage of feature and component rich development boards. To reduce system costs, a custom PCB offering only required functionality should be designed as part of converting this project into a commercially viable product.
- The smartphone application needs to have improved network issue debug capabilities. Currently, if a problem emerges during the SmartConfig process, the user is left with very little recourse but to retry the process. Furthermore, the

degraded performance observed in the coffee shop study (Figure 17) is very difficult to definitively understand without better troubleshooting tools. For TrapSense to be commercially viable, the process of pairing the traps to WLAN APs must be robust and simple.

5.2.2 Unexpected Challenges

By far, the most challenging issue with this project was to debug a problem that was originally observed as the system sporadically not being able to send messages between Exosite and Google App Engine. For several days in a row during development of the project, the system would work fine for hours in a row and then would stop working in late afternoon or early evening. Hours and hours of debugging would ensue. Then, the system would start working again as the debugging sessions stretched past midnight CST.

During these debugging sessions, various tiers of the system were swapped out in attempts to isolate the cause of the system failure. For example, the system at one point was configured to communicate via email messages between Exosite and Google App Engine versus the current configuration of HTTP POST requests.

Finally, swapping out hardware and conducting A-B-A testing began to narrow the focus of debugging. By swapping out hardware, radios with new MAC addresses were introduced to Exosite and were brought up on their own accounts. Eventually, it was determined that the daily Exosite HTTP request transaction quota was being exhausted before the end of the day. As debug work continued into the evening, the account quotas were being refreshed at midnight.

To reduce the number of transactions between Exosite and Google App Engine and avoid exhausting account quotas, the business logic regarding interpreting trap state

and checking for lost connection had to be moved. The business logic was pulled out of Google App Engine, re-written in Lua and hosted using the Exosite custom script functionality.

5.2.3 Top Fifteen User Stories

Incorporating Market Requirements from earlier in this report and brainstorming work from Joe Forbes and Ilsun Park, User Roles and User Stories were created to integrate and represent future product requirements.

User Roles and User Stories were chosen as the vehicle for representing requirements to align with Agile Software Development best practices [20]. A User Role is defined as a collection of defining attributes that characterize a population of users and how they are expected to interact with the system. User Stories simply describe system functionality that will be useful to a User Role of the system [20].

In total, seven User Roles and sixty-one User Stories were generated. The comprehensive set of User Roles and User Stories are documented in Appendix B and collectively define the system requirements for a commercial-intent version of this project. The Top Fifteen User Stories and related User Roles are excerpted below.

5.2.3.1 Key User Roles

Dispatcher - Works in office, routes work to field technicians.

Field Technician - Pest Management Professional working at the customer site.

Occupant - The person or business currently occupying the customer site.

5.2.3.2 Top 15 User Stories

- As a dispatcher, I want device fleet status sent to the web app so that all employees can access the information.

- As a dispatcher, I want device alerts sent to email so I can route those messages to field techs and integrate with our current business practices.
- As a dispatcher, I want the web app to support adding customers, then adding customer jobs (customer/job site/date) and then assigning customer jobs to a field tech so I can quickly organize and assign the team's work.
- As a dispatcher, I want the customer screen to include a field for "Customer Number" so I can link the TrapSense Customer record back to my existing CRM solution.
- As a dispatcher, I want to view device status by field tech, by customer and by stage (TrapSense Factory, In Transit, Inventory, Truck Inventory, Customer Assigned, Armed, Alert, Lost, Damaged, RMA Issued, RMA In Transit, TrapSense Received).
- As a dispatcher, I want tools to help identify lost devices to limit the expense of replacing lost devices.
- As a field tech, I want customer/address information included in any SMS alert so I know what unit is triggering the message.
- As a field tech, I want devices to be battery powered, so I can easily use them in areas like crawlspaces and attics that typically lack sufficient electrical outlets.
- As a field tech, I want devices to be individually addressable and capable of making an audible sound so that I can find a lost device at a job site.
- As a field tech, I want to access the TrapSense web app via browsers running on desktop computers, laptop computers, tablets and smartphones so I can check my traps' status while at home, at the office, or in the field.
- As a field tech, I want to be alerted if a TrapSense cage catches an animal so I can service the trap before the animal dies.

- As a field tech, I want to be alerted if a TrapSense snap trap kills an animal so I can service the trap before the occupant detects an odor.
- As a field tech, I want to view the device fleet sorted by customer, by customer job and by device status so I can plan my travel route for the day.
- As a field tech, I want TrapSense devices to be water resistant, so that I can deploy them outside and so that I can clean them off with sprayed water without the unit being damaged.
- As an occupant, I want all communications with TrapSense (device, smartphone app, web app) to be authenticated and encrypted so I can be protected from hackers injecting false alerts or stealing my personal information.

Chapter 6: Conclusion

6.1 Summary

A prototype was developed to demonstrate a remotely monitored smart rat trap for use by the pest control industry. The target application was selected after evaluating several potential markets with the input from advisors from Tech Ranch. The system was designed to minimize material costs while still providing an easy to use, wirelessly connected device to alert pest control professionals that a trap required servicing. After conducting interviews with nine pest control firms, a financial model was developed that estimates the US pest control industry could recover $175M annually in losses related to unnecessary truck rolls if deployed traps and cages were able to notify firms when servicing was required.

6.2 Lessons Learned

6.2.1 Top 5 Do:

- When considering a commercial venture, engage customers early in the process versus developing the product solely informed by your own experience.
- Invest time in component selection: features and ecosystem are both crucial.
- Set hard development deadlines and be willing to start compromising, defeaturing and altering course if deadlines are missed.
- Explore riskiest and most uncertain areas of a technology solution first.
- When conducting an extensive customer survey, send a copy ahead of time to allow customers to pull required data in advance of the meeting.

6.2.2 Top 5 Don't:

- Do NOT lose sight of account quotas and limits when using third-party services, platforms and APIs!

- Do not let pursuit of perfection inhibit the securing of sufficiency.
- Never start a serious hardware project with only one set of hardware. Always have at least two full sets of hardware on hand. Preferably three.
- Do not attempt to address too many markets when launching a new venture. Pick one market/approach/product and invest in accumulating deep domain expertise. Then, if necessary, pivot.
- As a student entrepreneur, do not miss out on participating in UT-specific programs from ATI. Had the program and application timelines been better publicized, this project would have benefited from participation in the Student Entrepreneurship Acceleration and Launch (SEAL) program - a two-month incubator program that meets during the summer term.

6.3 Related Work

The Woodstream Corporation owns both Victor and Havahart, leading manufacturers of equipment for the pest control industry. Woodstream had partnered with Exosite, to build a demonstration system in which an electronic rat trap, that works by electrocuting the rats, had been connected into the Exosite cloud. However their design has several disadvantages in comparison to this project. Among the disadvantages of the Woodstream project was that the wireless link to the trap required plugging a dongle into a PC that must be left powered on for the duration of the trap deployment. Another disadvantage of the Woodstream solution was their selection of the expensive electronic trap as their trap for instrumenting. Presumably, the product was developed with a focus on consumer markets versus the pest control professional market. The product is not commercially offered, but Woodstream has received patents for their work in this area (EP2710891 A1) [21].

Academic work related to this project can be found in areas relating to how TI's SmartConfig works – a key component of the TrapSense solution. Paul Martin, a Canadian M.A.Sc. candidate, wrote his 2007 thesis on using covert channels in secure wireless networks to communicate information. Although the premise of the paper was that a virus had been installed in a network and was sending out reports, this paper anticipates using frames of varying lengths to communicate information from within a network [22].

6.4 Future Work

Future work on this project involves three main paths of exploration. These three areas of exploration are business opportunity assessment, prototype development and beta customer engagement.

On the business front, an intellectual property lawyer needs to be engaged to assess the patentability of this device. Before extensive development occurs, the proposed solution would need to be reviewed to ensure that it has a reasonable chance of receiving a patent. In addition, a second round of customer surveys must be conducted to assess what the market can bear regarding smart trap and cage purchase prices and monthly monitoring fees.

Next, assuming the above business assessment yields promising results, a small set of beta units should be developed using the newly available CC3200 development boards, the 3-D printed enclosures based on the design seen in Figure 19, and the barest minimum viable subset of software features defined in Appendix B.

Finally, three pest control firms should be recruited as beta customers. Hardware would be lent to them, but they should be charged a modest monthly charge to

demonstrate viability to investors. Each firm would have a set of 10 traps and would be expected to provide ongoing feedback regarding device performance.

After the above three initiatives are complete, the feedback from customers and the performance of the devices will inform next steps on this project. In the best case, a limited amount of funding would be raised based on the initial customer traction with the beta customers. That funding would be used to drive design of a custom PCB, initial production runs, and building out the software part of TrapSense with a commercially feasible minimum set of features as outlined in Appendix B.

Appendix A: Pest Control Firm Survey Results

Firm	Monthly Rat Jobs/$M Annual Rev	Monthly Live Trap Jobs/$M Annual Rev	Est. Wasted Trips/Rat Job	Est. Wasted Trips/Live Trap Job	Est. Cost/ Truck Roll
A	21.6	11.8	0.0	0.0	$ 55.00
B	36.7	20.0	1.0	1.0	$ 55.00
C	40.3	15.5	2.5	0.0	$ 25.00
D	22.5	10.8	1.0	2.0	$ 40.00
E	-	-	0.0	1.5	$ 69.00
F	35.0	30.0	2.5	0.0	$ 9.50
G	33.3	1.7	1.5	1.0	$ 40.00
H	33.3	20.0	1.5	1.0	$ 50.00
I	25.0	7.5	0.6	1.0	$ 40.00
	31.0	14.7	1.2	0.8	$ 42.61

Figure 20: Pest Control Firm Survey Results

Appendix B: TrapSense User Roles and User Stories

Working with Ilsun Sun and Joe Forbes, the following User Roles have been identified for the TrapSense system. The User Roles are organized into "External User Roles" comprised of TrapSense system users not employed at a future TrapSense company and "Internal User Roles" comprised of TrapSense system users employed at a future TrapSense company.

External User Roles

- Dispatcher - Works in office, routes work to field technicians.
- Field Technician - Pest Management Professional working at the customer site.
- Occupant - The person or business currently occupying the customer site.

Internal User Roles

- Support Rep - Customer support representative provides remote assistance to customers, manages order fulfillment and coordinates customer returns.
- Developer - Software developer concerned with fixing bugs and adding new functionality to the system and then coordinating subsequent release migrations.
- Operations Admin - Monitors fleet of devices for usage and alert status.
- Marketer - Specifies new products and functionality and then supports their sale to customers.

Using the above User Roles, the following User Stories were developed with input from Joe Forbes and Ilsun Sun to represent the TrapSense system requirements and provide a starting point for implementing a commercially viable version of this project.

- As a developer, I want devices to only accept a firmware upgrade if their batteries have sufficient charge to complete the upgrade so that devices are not bricked during a partial upgrade.
- As a developer, I want the web and smartphone apps to be internationalized from the very beginning so that adding support later for additional languages will be easy.
- As a developer, I want to push web app upgrades without losing data from inbound device fleet status messages.
- As a developer, I want to track firmware upgrades within the device fleet to allow for escalation if devices not receiving required updates.
- As a developer, I want to track smartphone app upgrades to allow for escalation if users are not receiving required updates.
- As a dispatcher, I want device alerts sent to email so I can route those messages to field techs and integrate with our current business practices.
- As a dispatcher, I want device fleet status sent to the web app so that all employees can access the information.
- As a dispatcher, I want the customer screen to include a field for "Customer Number" so I can link the TrapSense Customer record back to my existing CRM solution.

- As a dispatcher, I want the web app to support adding customers, then adding customer jobs (customer/job site/date), and finally assigning customer jobs to a field tech so I can quickly organize and assign the team's work.
- As a dispatcher, I want to receive a new TrapSense device and easily associate it to our firm's account and move it into "Inventory" status so I can easily start using my new devices.
- As a dispatcher, I want to view contact information associated with each field technician.
- As a dispatcher, I want to view device status by field tech, by customer and by stage (TrapSense Factory, In Transit, Inventory, Truck Inventory, Customer Assigned, Armed, Alert, Lost, Damaged, RMA Issued, RMA In Transit, TrapSense Received).
- As a dispatcher, I want tools to help identify lost devices to limit the expense of replacing lost devices.
- As a dispatcher, I want tools to help identify lost devices to limit the legal liability resulting from lost company equipment being misused by other parties.
- As a dispatcher, I want TrapSense billing statements to capture what job sites were monitored by which traps.
- As a field tech, I want customer/address information included in any SMS alert so I know what unit is triggering the message.
- As a field tech, I want device alerts to be reported with localized time vs. UTC so I can easily understand the time of the event.
- As a field tech, I want devices to be battery powered, so I can easily use them in areas like crawlspaces and attics that typically lack sufficient electrical outlets.

- As a field tech, I want devices to be individually addressable and capable of making an audible sound so that I can find a lost device at a job site.
- As a field tech, I want devices to seamlessly receive firmware updates so the devices can be kept up-to-date without any intervention from me.
- As a field tech, I want a device activation to support adding an optional picture and optional text description to help me later find the device.
- As a field tech, I want the smartphone app to automatically add GPS-stamp or Address-stamp (if GPS signal blocked by building) information during a device activation to help me later find the device.
- As a field tech, I want the smartphone app to help troubleshoot TrapSense Wi-Fi/SmartConfig issues so I can easily fix Wi-Fi connection issues.
- As a field tech, I want to access the TrapSense web app via browsers running on desktop computers, laptop computers, tablets and smartphones so I can check my traps' status while at home, at the office, or in the field.
- As a field tech, I want to assign a TrapSense device to a given customer job either by scanning a QR/bar code on the device or by manually entering serial numbers so that I can later easily determine which traps are at which job site.
- As a field tech, I want to be alerted if a TrapSense cage catches an animal so I can service the trap before the animal dies.
- As a field tech, I want to be alerted if a TrapSense snap trap kills an animal so I can service the trap before the occupant detects an odor.
- As a field tech, I want to be alerted if TrapSense device loses communications with the TrapSense remote monitoring service so I can remedy the situation.
- As a field tech, I want to be alerted of a low battery so I can change the battery before communications are lost with the device.

- As a field tech, I want to check the network configurations and status of a specific device to assist with debugging possible network issues.
- As a field tech, I want to configure notification settings at my account level so I can set a default engagement model with TrapSense.
- As a field tech, I want to easily check the device battery level to avoid deploying a device that will shortly thereafter begin to issue low battery alerts.
- As a field tech, I want to easily replace the device batteries so I can service the devices in the field without having to invest very much time.
- As a field tech, I want to optionally have SMS messages spool up during non-work hours to allow me to sleep without interruption.
- As a field tech, I want to optionally override default account notification settings at the customer job level so that I can tailor my support to the specific customer's needs as necessary.
- As a field tech, I want to optionally record the kind of bait used with each TrapSense device and optionally set a reminder alarm to refresh the bait.
- As a field tech, I want to optionally reset a device's network settings to debug connection problems and to prepare the device for its next installation.
- As a field tech, I want to optionally view status of devices installed by other field techs in my firm in the event that I am asked to cover for my co-workers.
- As a field tech, I want to perform a hard reset that will wipe the device and pull down a new firmware image.
- As a field tech, I want to receive alerts as text messages so that I can know action is required without having to access a website or email account.
- As a field tech, I want to use a smartphone app to easily understand my device fleet status.

- As a field tech, I want to view the device fleet sorted by customer, by customer job and by device status so I can plan my travel route for the day.
- As a field tech, I want TrapSense devices to be water resistant, so that I can deploy them outside and so that I can clean them off with sprayed water without the unit being damaged.
- As a field tech, I want TrapSense devices to support batch provisioning, so I can quickly program SSID and password into all devices assigned to a specific job site.
- As a marketer, I want customer and device fleet data to be anonymized and aggregated so that I can pursue supplementary revenue streams.
- As a support rep, I want the TrapSense web app to manage generating monthly bills and collecting payments from credit cards and PayPal to reduce the overhead of billings and payments.
- As a support rep, I want to be alerted of bounced email messages so I can proactively address problems with customer profile email addresses.
- As a support rep, I want to be alerted of bounced SMS messages so I can proactively address problems with customer profile phone numbers.
- As a support rep, I want to issue a Return Material Authorization so customers can return non-functional units while within warranty period.
- As a support rep, I want to look up a dispatcher or field tech by email address so I can quickly access account details when the user calls.
- As a support rep, I want to look up a dispatcher or field tech by telephone number so I can quickly access account details when the user calls.
- As a support rep, I want to look up a pest control firm by device id so I can quickly access customer account info when the user calls.

- As a support rep, I want to receive a returned device and be able to look up the associated RMA by device number so I can determine next steps for the device.
- As a support rep, I want to see a history of events from a given device so I can identify when a problem may have started.
- As a support rep, I want to see the alerts sent to a field tech so I can see what that user is seeing in the event of a service call.
- As a support rep, I want to see the device configuration for a given device so I can troubleshoot customer problems.
- As an occupant, I want all communications with TrapSense (device, smartphone app, web app) to be authenticated and encrypted so I can be protected from hackers injecting false alerts or stealing my personal information.
- As an occupant, I want my personal info, SSID and password to be protected so I can avoid identity theft.
- As an occupant, I want to receive reports regarding my food service place of business that documents active monitoring services from TrapSense was in place so that I can prove to health inspectors that I have an active pest management plan.
- As an operations admin, I want to see device fleet data usage so that we do not incur unplanned overage fees.
- As an operations admin, I want TrapSense fleet analytics data to be anonymized so we do not raise privacy concerns.

WORKS CITED

[1] Andrew Burger. (2014, November) Telecompetitor. [Online]. http://www.telecompetitor.com/idc-internet-of-things-base-to-reach-30-billion-in-2020/

[2] Visiongain. (2014, June) Visiongain. [Online]. https://www.visiongain.com/Report/1283/Wearable-Technology-Market-Report-2014-2019

[3] Plamen Nedeltchev. (2014, January) Cisco. [Online]. http://www.cisco.com/c/en/us/solutions/collateral/enterprise/cisco-on-cisco/Cisco_IT_Trends_IoE_Is_the_New_Economy.pdf

[4] Andrew Dugan. (2014, January) Gallup. [Online]. http://www.gallup.com/poll/166745/americans-tech-tastes-change-times.aspx

[5] Joshua Brustein. (2013, November) Businessweek. [Online]. http://www.businessweek.com/articles/2013-11-13/why-ge-sees-big-things-in-quirkys-little-inventions

[6] Pest Control Technology, "2014 PCT Top 100 List," *PCT Magazine*, May 2014.

[7] Texas Instruments. (2012, February) Texas Instruments Incorporated. [Online]. http://www.ti.com/lit/ug/slau343b/slau343b.pdf

[8] Texas Instruments. (2011) Texas Instruments Incorporated. [Online]. http://www.ti.com/lit/ml/slau341/slau341.pdf

[9] Texas Instruments. (2012, November) Texas Instruments. [Online]. http://www.ti.com/lit/ds/symlink/cc3000.pdf

[10] Cory Doctorow. (2005, November) boingboing.net. [Online]. http://boingboing.net/2005/11/08/wifi-isnt-short-for.html

[11] Wi-Fi Alliance. (2014, January) Wi-Fi Alliance. [Online]. http://www.wi-fi.org/file/connect-your-life-wi-fi-and-the-internet-of-everything-2014

[12] University of Texas at Austin Information Technology Services. (2013, August) The University of Texas at Austin. [Online]. http://www.utexas.edu/its/help/network/1623

[13] George Hawkins. (2013, October) Depletion Region. [Online]. http://depletionregion.blogspot.ch/2013/10/cc3000-smart-config-transmitting-ssid.html

[14] Texas Instruments. (2014, April) Texas Instruments Incorporated. [Online]. http://processors.wiki.ti.com/index.php/CC3000_Smart_Config

[15] Twilio. Twilio. [Online]. https://www.twilio.com/sms

[16] Texas Instruments. Code Composer Studio (CCS) Integrated Development Environment (IDE). [Online]. http://www.ti.com/tool/CCSTUDIO

[17] Google. (2013, December) Google Developers. [Online].

http://code.google.com/eclipse/

[18] TeraTerm Project. (2014, June) Sourceforge.jp. [Online]. http://ttssh2.sourceforge.jp/manual/en/

[19] Texas Instruments. (2014, October) Texas Instruments Incorporated. [Online]. http://processors.wiki.ti.com/index.php/CC3000_Patch_Programmer

[20] Mike Cohn, *User Stories Applied for Agile Software Development*. Boston, MA, USA: Pearson Education, Inc., 2004.

[21] Christopher, Jr. Rich and Thomas Daly, "Wireless notification system and method for electronic rodent traps," Application EP2710891 A1, March 26, 2014.

[22] Paul Edwin Charles Martin, "Covert Channels in Secure Wireless Networks," Division of Graduate Studies, Royal Military College of Canada, Ottawa, Master's Thesis 2007.

[23] Leo Mirani. (2014, September) Quartz. [Online]. http://qz.com/263835

[24] Linux Wireless. [Online]. wireless.kernel.org/en/users/Documentation/modes

[25] Microsoft. [Online]. http://msdn.microsoft.com/en-us/library/windows/hardware/ff568369(v=vs.85).aspx

[26] Airsnort Homepage. [Online]. http://airsnort.shmoo.com/faq.html

[27] Texas Instruments. (2014, June) Texas Instruments Incorporated. [Online]. http://www.ti.com/lit/ds/symlink/msp430fr5738.pdf

[28] Texas Instruments. Texas Instruments Incorporated. [Online]. http://www.ti.com/product/MSP430FR5738

[29] Texas Instruments. Texas Instruments Incorporated. [Online]. http://www.ti.com/tool/cc3000boost

[30] Texas Instruments. (2014, August) Texas Instruments Incorporated. [Online]. http://www.ti.com/lit/ug/swru331a/swru331a.pdf

[31] Texas Instruments. (2013, September) Texas Instruments Incorporated. [Online]. http://processors.wiki.ti.com/index.php/CC3000_First_Time_Getting_Started_Guide

[32] Google. Google. [Online]. https://cloud.google.com/compute/docs/faq

[33] Google. Google Cloud Platform - What is Google App Engine? [Online]. https://cloud.google.com/appengine/docs/whatisgoogleappengine

[34] Exosite. Zero Barrier to Entry for IoT. [Online]. http://exosite.com/zero-barrier/

[35] Exosite. Exosite Support. [Online]. https://support.exosite.com/hc/en-us/articles/200513440-Scripting