TrackAlert

A Short-Range Emergency Tracking System

Carlos G. Araujo, Melissa Grenier, and Gina McGhee, Sainjulien Senatus

Dept. of Electrical Engineering and Computer Science, University of Central Florida, Orlando, Florida, 32816-2450

***Abstract* — an amalgamation of three of the more prevalent communication technologies in the world today, TrackAlert is a practical electrical and computer engineering project which seeks to utilize RFID, RF, Ethernet, and GPS technologies to create a tracking system which allows its users to safely monitor those who are unable to supervise themselves. Intended for use in Daycares, Hospitals, and the like this system integrates a web application, GPS arm band, RFID system, and routing station to facilitate an alarm system which informs its users via the web application, visible on any internet accessible device, to the violation of a predetermined geo fence.**

***Index Terms* — TCPIP, Ethernet Network, Base Station, Transceiver, Communication Systems, Satellite Communication**

I. Introduction

Intended to enhance the safety of those who would otherwise be unable to care for themselves, this project is first and foremost a system designed to improve the care of others. Composed of four separate yet equal systems working in conjunction to deliver notifications regarding the location of the designated person, TrackAlert implements RFID, GPS, RF, and web application subsystems to deliver critical information to the end user who can in turn take the appropriate action to both find and help the distressed person. Consisting of three separate microcontrollers, the RFID uses Texas Instruments' MSP430F2370, the Base Station uses the Stellaris LM3S8962, while the GPS device implements the Atmel Atmega328 all of which use a 16-bit, 32 bit, and 8-bit CPU respectfully. To facilitate communication with the end user, the web application utilizes Google's App Engine to allow any internet enabled device to access not only the location of the tracking device but also the serial number of the device and name associated with it. In order to initiate the emergency tracking system the RFID must first alert the base station that the GPS device (which will be located on the arm) has been scanned, an indication that the device has breached the specified area. This signal, transmitted via an X-bee Pro 900 RF module, in turn directs the base station to turn on the GPS device while also alerting the end user though the web application that an emergency has occurred. As a result, the GPS coordinates of the arm band are transmitted to the base station via the same technology seen on the RFID subsystem, and then relayed to the web application thought the Ethernet capabilities of the station so the end user can be periodically updated on the whereabouts of the device and more importantly its wearer.

II. RFID Subsystem

The RFID subsystem, despite its simplistic name, employs a myriad of components to facilitate the demands of a system which is the catalyst to the TrackAlert emergency system. Comprised of an RFID reader, passive tag, stick antenna, and RF transmitter this system seeks to not only to act as an alarm monitoring system but also indicate to the base station which GPS device has left the designated area and at which time.

A. MSP430 Microcontroller

Selected for its lower power consumption and wide supply voltage, the MSP430F2370 was selected for its 32KB memory capability, 32 general purpose pins, variety of universal serial communication interfaces; the most important of which being UART. Featuring a 16-bit RISC CPU, it should be noted that this chip as a result of it being surface mount must be programmed using a JTAG interface with a standalone emulator as this is one feature the MSP430 lacks. Programmed in either C or C++ the MSP430 uses Code Composer, Texas Instruments IDE.

B. Reader and Tag

In order to determine when a person wearing the GPS device has left a designated a passive RFID tag will need to be used in conjunction with the corresponding reader. To facilitate this process the ID-20LA read was selected. This chip runs on a 2.8-5V supply voltage, uses a read frequency of 125 kHz, supports a baud rate of 9600bps, and has a range of 7 inches. EM4001, the tag to be attached to the GPS device, works with in the 125 kHz RF range supported by the reader and comes with a unique 32 bit ID. Unfortunately non reprogrammable, this tag and reader combination transmits data to the MCU over its serial pins.

C. RF Transmitter

To enable the RFID system to alert the base station, the X-bee Pro 900 was selected. This transmitter not only supports long range transmission of up to 6 mile but it also uses UART serial communication to interface with the MSP430.

D. Power System

Because a multitude of voltages are required to supply power to the various components within the RFID system, three LM2574 voltage regulators are used. Powered by a 12 volt DC source, the first regulator restricts the output voltage to 3.3 to support the microcontroller. The second regulator confines the output voltage to 5, to account for the needs of the RFID reader. The third and final regulator is implemented to not only limit the output voltage to 3.3 volts but to also ensure the output current is no more than 210mA. The capacitors, resistors, and inductors needed to implement this power supply moreover can be seen in the schematic representation of the system in Figure 1 below.

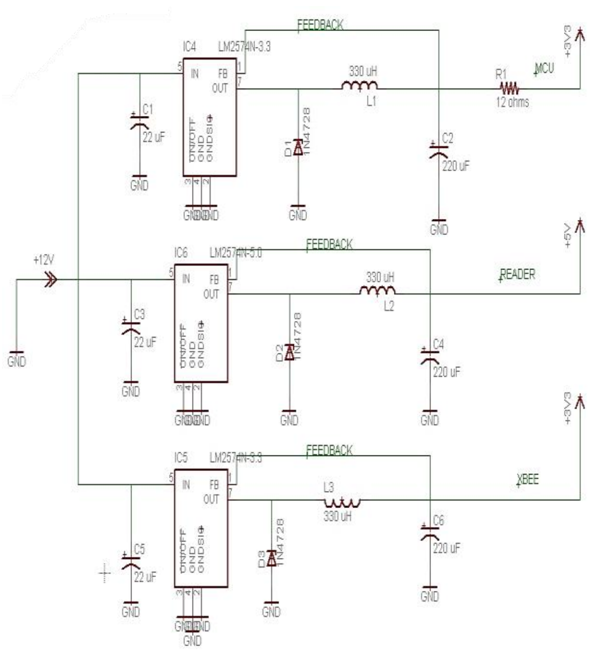

Figure 1: Schematic of RFID Subsystem

E. Logic

Over all, the RFID subsystem is very critical in ensure that the base station signals both the GPS and the web application. Based on the signals received from the RFID the base station will instruct the GPS device to being transmitting GPS coordinates while also alerting the end user via the web application that an emergency situation has occurred. This moreover can be more easily understood in the diagram in Figure 2 below.

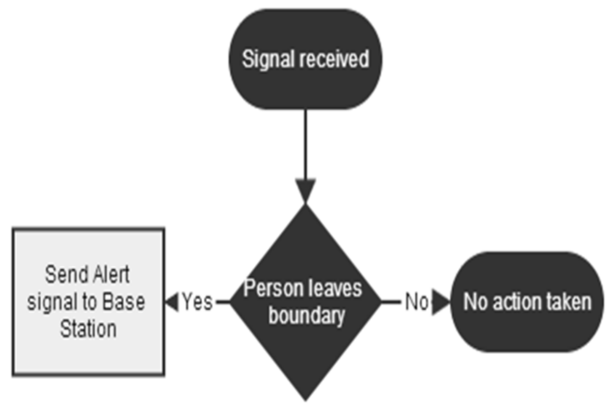

Figure 2: Flowchart of RFID Subsystem

III. GPS SUBSYSTEM

This device will contain a transmitter module, a rechargeable battery, a LCD character display, a microcontroller, an X-Bee, and the GPS chip. The GPS chip will calculate its actual location and outputs the pseudo range data for each satellite. This data will be processed on the microcontroller, and the X-Bee will transfer this information to the base station. Thus, the microcontroller main purpose on this device is to change the output of the GPS chip from NMEA protocol to SIRF binary protocol. By doing so, we will be able to get the pseudo range locations in an executable format.

A. LCD Display

The character display on the device will have many functions. Since each device will be assign to a specific person, the name, the date of birth, age and contact number of that person will be display on the device.

As far as character display components, there are many options in the market to choose from. One limitation on deciding on the right character display is the size. As part of the design, we are trying to make the GPS tracking device as small as possible, so each part add to the device has to be no bigger than 75x53 mm.

The LCD Shield w/ 16x2 character display allows us to control a 16x2 character LCD, up to 3 backlight pins and 5 keypad pins using only the two I2C pins. This shield goes well with stand - alone projects because the 4 directional buttons and the select button allow basic control without having connected a computer or laptop

B. GPS Module

Among all GPS Modules, The LS20031 GPS Module is one the commonly used in the market. Besides the reasonably price, it comes with its antenna and cables required for its respective connections. One of the advantages of using this GPS Module is its size. It is about 25.4mm square, which meet the size requirements for the tracking device being built. The price is about 60 dollar.

This GPS Module has 5 connections.

1) Enable: This signal operates under active high.
2) Transmit data and receive data: A standard serial protocol (UARTs)
3) Supply voltage
4) Ground
5) Ground

The LS20031 implements a subset of the NMEA 183. It defaults to binary mode at a high baud rate at 57600 baud, 1 stop bit and no parity. Its default outputs are GSA, GSV, and RMC. Depending of the processor used, a certain type of output will be better to implement than the others.

The GSA message lists the active satellites, those that are being used by the GPS receiver at the moment. It also includes dilution of precision information, which is an indication of how good or poor the current satellite geometry is. When the satellites are all in a line, or clumped together, then the dilution of precision is high (poor). When the satellites are spread throughout the sky then the dilution of precision is low (good). Figure 3 depicts the order of events.

From the GPS Satellites, the LS20031 calculates a valid position on the earth in a process that takes about 12.5 minutes when the module has not been used for the first time or when the module has been a long time since was last used. Otherwise, the module will save the data in memory throughout a capacitor so when power is reapplied the module can find its location in faster time.

The antenna being used by the module is a built in patch antenna. It is a type of antenna with a low profile and it is specially design to be mounted in a flat surface. This antenna has problem in getting signals from the satellites in the horizon. Special situations such as weather conditions or being indoors can affect the performance of the antenna and can take up to 5 minutes to get a lock. The most sensitive orientation for this antenna is pointing to the sky.

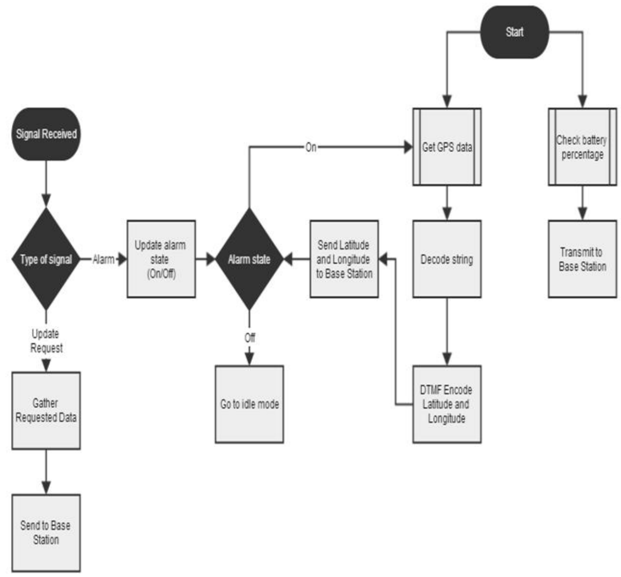

Figure 3: Flowchart of GPS Subsystem

C. Base Station Interaction

In the flow chart in Figure 3, we describe how the GPS device is communicating with the base station, and the information that is being sent to it. When the person carrying the GPS device is out of the safety range, an alarm state will be activated. At that moment, coordinates involving latitude and longitude will be transmitted to the base station. As safety method, the battery percentage is also being sent to the base station, so the caretaker will know for how long the device will be on.

D. Power Supply

The supply voltage tolerance for this GPS module change between ranges of 3.2 to 5 volts. According to its specifications, this module is very sensitive to values below 3.2 volts. Also, the normal operating current is 45mA, but it is important to keep in mind that the current jumps a lot during the startup process. It could go from 75mA to about 90mA peak. On the other hand, the current draw drops to about 0.4mA when the Module is not operating. Because of these characteristics, there are some terminal regulators that have to be considered. In addition, a rechargeable battery is required which required the implementation of a charge controller. According with the specifications of the project, size is an important requirement for the device. All components have to meet their specifications required in small sizes possible. The information from this section is gathered from. [1]

The Lithium ion (Li-Ion) battery seems to be best choice not just because it is recyclable but comes in different shapes. All the issues related with memory effect are not a problem anymore, so this battery can be recharged before it is completely discharged without affecting the energy capacity. In addition, Lithium ion batteries are smaller, lighter and provide more energy that the other ones describe above. The main disadvantage of this battery is that it is not response well with high temperatures. Li-Ion sometimes erupts or explodes in high heat, hot cars, or direct sunlight. That could be a problem because tracking a person over summer temperatures would cause the device to heat, but according with its specifications this battery will work without any problems here in Florida's temperatures [2].

D. Microcontroller

For the MCU in the GPS device, we have selected to work with an Arduino. The Arduino Uno is a development package that runs off a wiring-base language similar to C++, and is based on the 8-bit ATmega328 microchip. Arduino Uno functions are more accessible for beginners because it is meant as an electronic solution for those who might not be familiar with embedded design. Since its release, hobbyists have used Arduino extensively. There are variety libraries, and tons of information in terms of code for support.

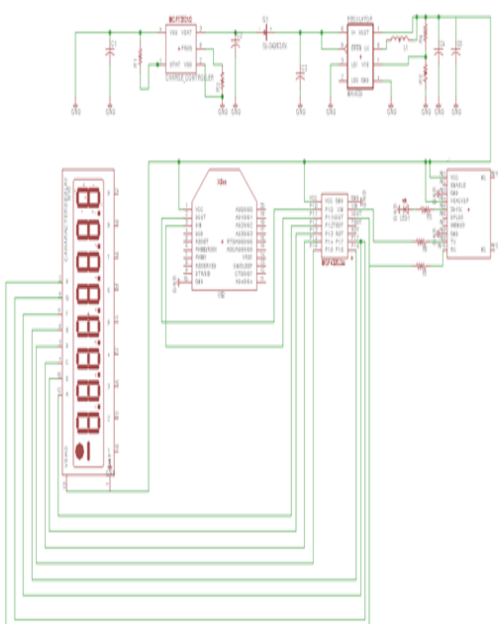

Figure 4: Schematic of GPS Subsystem

In addition, the ATmeg328 Series was implemented in the GPS project because this chip uses the RISC architecture and a low power draw level. This chip has a max clock speed of 20 MHz, an 8-bit AVR CPU, pin size of 32, 23 I/O pins, 10-bit ADC, an 8-channel, and an operating voltage of 1.8 to 5.5 programmable flash memory over 16 KB. The software used is AVR Studio 5, an IDE that allows the coding of assemble and C/C++ projects within any windows platform. The software used is free and can be found online. The schematic of the GPS device can be seen below in Figure 4.

V. BASE STATION SUBSYSTEM

Developed as a means through which information can be routed, the primary purpose of the base station is to use wireless technology to transmit data between the GPS device, RFID reader, and web application. With respect to the information exchanged between the base station and RFID reader only unidirectional communication is taking place. The base station simply waits for a flag sent from the RFID reader which will indicated that the GPS device has left the designated area. Once this signal is received the base station will then send a command to the GPS device signaling the continuous transmission of GPS coordinates.

Once the base station begins to receive GPS coordinates from the GPS device moreover, it can then begin communication with the web application by first alerting the application to the breach of perimeter indicated by the RFID reader. This will then allow the end user to safely track and locate the person in distress, quickly reducing any chances of injury.

A. Microcontroller

The microcontroller through which information will be routed and translated is the Stellaris LM3S8962. This chip was selected because it provides both the MAC and UART components needed to enable communication with both the transceiver and Ethernet controller. The Stellaris is in direct communication with the GPS device, RFID Reader, and smartphone via RF transmissions and internet connectivity. This MCU has 42 general purpose pins, 2 UART serial pins, operates at a 2.25 to 2.75 voltage, and supports a max baud rate of 3.125 Mpbs which is wide enough to support the demands of any RF transmitter.

Texas Instrument provides the LM Flash Programmer as a free programming method through which both the Stellaris microcontroller and evaluation board can be programmed. Programmable via the JTAG port see on the development board associated with this chip, Code

Composer is the IDE utilized by the Stellaris. Programmable in either C or C++, the code for the base station is a combination of the UART Echo and Ethernet sample codes provided with development board.

```
void
UARTSend(const unsigned char *pucBuffer, unsigned long ulCount)
{

    //
    // Loop while there are more characters to send.
    //
    while(ulCount--)
    {
        //
        // Write the next character to the UART.
        //
        UARTCharPutNonBlocking(UART0_BASE, *pucBuffer++);
    }
}
```

Figure 5: Sample Code for Transmitting Data

B. Ethernet

In order to facilitate communication with the web application, since the MCU selected to support the base station has both MAC and PHY layer, all that is needed to enable Ethernet communication is an RJ-45 jack with integrated magnetics and of curse a router. The Pulse J3011G21DNL RJ-45 Ethernet jack was selected because it eliminates the need to provide the 1:1 transformer, or magnetics, needed by the chip; and provides both a green and yellow LED so the flow of traffic over the Ethernet can be monitored. The MCU will host a small server that will be listening on port 80 for incoming HTTP requests. When the web application polls the MCU for data it will do so through an HTTP GET request, the server will parse the incoming request and route it to the appropriate CGI handler to process the request. Once the request is processed, which will be asking for updated data, it will put together the response and send it to the web application.

C. RF Transmitter

Touted for its ability to send information through long distances, the type of RF transmitter used for this project needed to not only transmit information but receive it as well. Because information will need to be transmitted over at least a mile, it is important to understand the differences, advantages, and short comings of both AM and FM signals. Although both are used in analog and digital communication, AM signals tend to have poor sound quality in comparison to its FM counterpart. Moreover despite the fact that FM is more resistant to interference, it is more heavily impacted by physical barriers which could prove to be an issue for a system which seeks to track a device in an urban setting.

In order to send and receive information to both the GPS wrist band and RFID reader, a wireless transmitter is required. While there are a plethora of methods with which to accomplish this transmission of data there a few constraints which decrease the selection pool. Among some of said restrictions include the need for a communication system which can transmit data up to a mile away. The transmitting device must also be fairly small in that not only will it need to be mounted on the base station but it will also need to be mountable and the GPS tracking device.

The X-bee Pro 900 was the RF module selected that best fits the needs of this project allowing for communication between the base station, RFID reader, and GPS device. Selected for its ability to transmit information at distances of a mile or more, its 3.3v power consummation and 210mA current draw are ideal. Requiring the use of a A09-HASM-675 high gain antenna to ensure the specified distance is achieved, the X-bee Pro 900 uses X-CTU to configure the devices for AT data transmission and supports a baud rate of up to 156 Kbps. Provided with the UART serial data interface, the X-bee seamlessly connect to the RX and TX pins seen on the Stellaris require little else to transmit data.

D. Board Design

Because all the components associated with the base station use a supply voltage of 3.3 V the design for the power supply was fairly simple and heavily based on the design used for the Stellaris development board. Supplied by a 5 volt input and a current of 1 amp, the LP8345ILD 3.3 voltage regulator was used to not only regulate the voltage to the aforementioned 3.3 volts but also to provide a 650 mA current.

As determining which resistors and capacitors would be best to support this system can be a daunting task, all of the same values and package sizes suggested in the bill of material for the Stellaris development board were used.

The actual board was designed using the PCB design software Eagle CadSoft. Populated with a variety of libraries for a wide range of components, libraries were found for the X-bee and MCU seen in the figure. The voltage regulator and RJ-45 jack however, did not have preexisting libraries and had to be handcrafted. Seemingly simple in nature the process of creating a new eagle library is very tedious. In order to complete the process a symbol must first be designed to represent the component. Once the symbol is complete, a device is created. During this process the dimensions of the part, be it a voltage regulator or RJ-45 jack, are used to accurately design the size and shape of the part in the CAD-like software. After the device has been completed both the symbol and device pins must be associated with each other, ensuring the traces to be drawn to the pin are accurate.

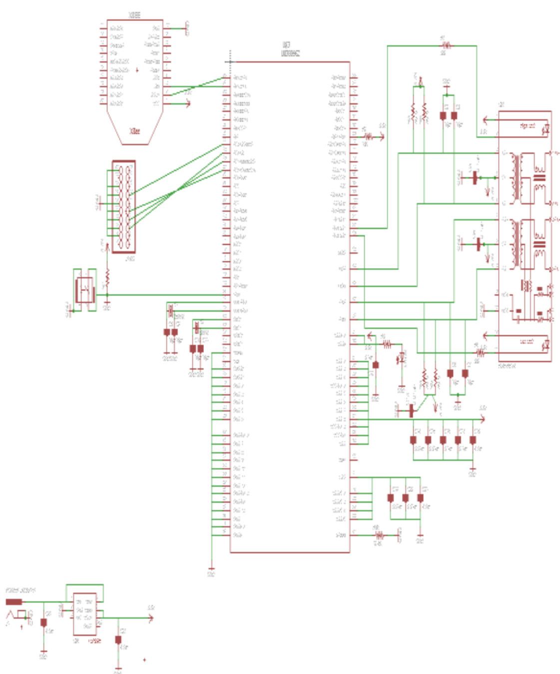

Figure 5: Schematic of Base Station

E. Logic

Despite the in-depth description of the hardware associated with the base station, to thoroughly understand the inner workings of the system a logic diagram is needed. The base station is designed to work off of interrupt handlers. When the base station receives an interrupt from the UART, it processes the data to see if it comes from the RFID system. A signal from the RFID indicates that a GPS device being monitored by the RFID has left the area. Once this signal has been received, the base station alerts the web application so that the emergency contact linked to that device is alerted. If the signal received comes from the GPS device, the data is processed and saved for when the web application next polls the MCU. And lastly, if it was the Ethernet interrupt handler that interrupted the MCU, the signal comes from the web application and is requesting data. In this case, the data received from the GPS device is packaged and sent to the web application as part of the HTTP response. This moreover can be better seen in the diagram in Figure 7 below.

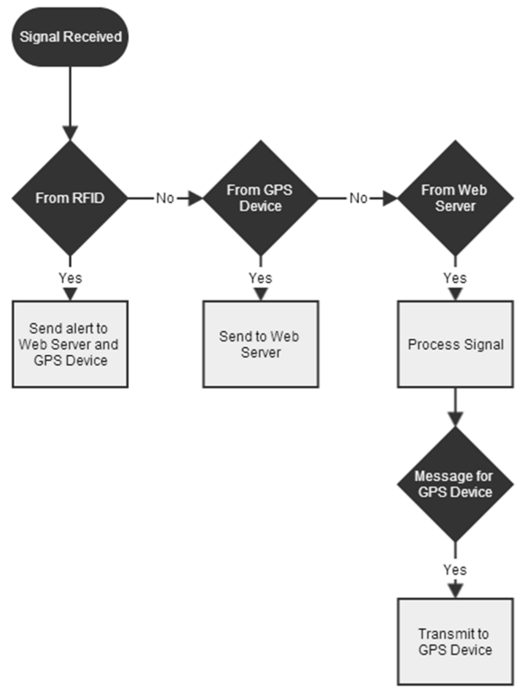

Figure 6: Flowchart of Events at Base Station

V. WEB APPLICATION

Using a web-based application gives the advantage of database availability and portability, since it is platform-independent and thus can be accessed from any hand-held device that has Ethernet connectivity, an objective for this project. The web application developed for the TrackAlert system consists of two main parts: the user interface and Google App Engine.

A. User Interface

One main aspect that was central to the implementation of the TrackAlert system was to provide an easy to use user interface. This was done using JQuery Mobile and HTML5 which allow for a clean looking interface on hand-held devices such as smartphones. This interface can be accessed either from a hand-held device or a regular computer as long as there is Ethernet connectivity.

For added security, the user interface requires the user to register/sign in to the application using a Google account username and password, the login page will be the first page seen on the application. Once signed in, the user will be taken to the main page where they will have the ability to view and modify the personal information regarding the person wearing the GPS tracking device, including their name, date of birth, emergency contact person and number. The user will also be able to view the information regarding the GPS device, including battery life and last known location's latitude and longitude. This last known location will also be available on a Google map embedded in the user interface.

There will also be two other features available to the user: to add a new device or delete an existing device from their account. A depiction of the user interface can be seen in the figure below.

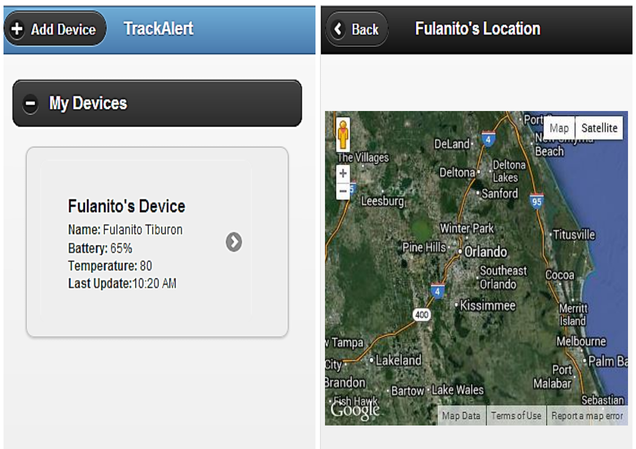

Figure 7: Web-App User Interface

B. Google App Engine

Google App Engine provides the server-side needs and services that allow the web application to function. In particular, it enables the creation of a web application on the same scalable system that power Google applications. App Engine applications are easy to build and there are no servers to sustain as Google hosts and scales the applications. It allows for development in Java, Python, Go, and recently, PHP. Another advantage to using Google App Engine is the accessibility of a database; each application has access to its own database. Google App Engine is being used for two main purposes: to host application and database, and to communicate with the MCU at the Base Station.

i. Application Back-End and Database

The TrackAlert application makes use of Google's BigTable High Replication Datastore, a noSQL database, and Python capabilities to power the application's back-end. The application was built using Web2Py, a free open source full-stack Python Model-View-Controller (MVC) framework that was also developed to easily integrate with Google App Engine. One of the best features Web2Py offers is its Database Abstraction Layer (DAL), an API that dynamically generates the objects and queries needed for a specified database backend from Python objects. This eliminates the need of writing the database tables, queries, and records or learning different database dialects and makes the application portable among different types of databases. In order to keep the database as simple as possible, there are two main data models used: User and Device. Those two models hold all the information needed that refers to either the account user or the GPS tracking device and the person wearing it.

Web2Py automatically addresses and prevents security issues such as SQL Injections and cross site scripting vulnerabilities [1]. Web2Py uses a Web Server Gateway Interface (WSGI) setup to handle requests. When a page is requested, it is routed to the appropriate controller function to be processed; the response file is formulated and then sent back to the browser. Web2Py templating engine allows for Python holders to be placed in the HTML file to indicate where the generated content should be place. An illustration of the server and client interaction is shown below.

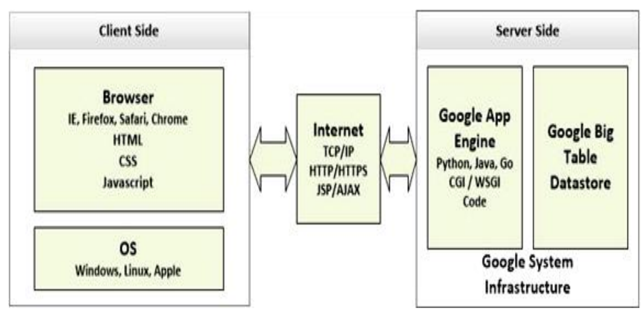

Figure 8: Client-Server Interaction

ii. MCU Communication

Google App Engine also offers two main services that will be used to communicate with the Stellaris at the Base Station: the App Engine Cron Service and the URL Fetch Service. The Cron Service is used to poll the MCU at regular intervals using an HTTP GET request from the URL Fetch Service to get updated information on the GPS device. This information will be processed and saved to the database. This is when the application will check to see if the signal flag was raised. If it was, an SMS message will be sent from the application to the number listed under emergency contact to alert them of the emergency state. To send the SMS message, a scheduled task will be invoked which will use the phone company unique e-mail address that is associated with the cell-phone number, so in reality, the SMS message will be sent as an e-mail. Web2py provides a module to help with the process.

VI. CONCLUSION

As the world becomes more and more dangerous for not only those who are unable to care for themselves by humanity in general the need for a system which will allow for peace of mind is paramount. To combat the fear associated with children being abducted, elderly people without sound judgment getting lost, or the negligence so often reported in long term care facilities TrackAlert hopes to have provided a solution. Striving to combine a number of technologies which include RFID, GPS, RF, and wireless communication, this system also has developed a web application making it accessible from any internet enable device. This combined with its ability to track up to a mile radius should allow TrackAlert to prevent an emergency situation before it has even truly begun.

VII. WORKS CITIED

As the tasking a designing, building integrating a plethora a different subsystems can be very challenge a number of papers, data sheets and technical forums were studied and referenced. Furthermore the help of countless professors and engineers were used as sounding boards to help facilitate the design process. Furthermore the knowledge gained in various programming, circuit, and electronic classes has also be applied to the implementation of this project.

ACKNOWLEDGEMENT

The authors wish to acknowledge the assistance and support of Dr. Samuel Richie and the University of Central Florida.

REFERENCES

[1] LOCOSYS Technology Inc. "Datasheet of GPS Smart Antenna Module." *Sparkfun.com*. N.p., 25 Oct. 2007. Web. 14 Nov. 2013. <https://www.sparkfun.com/datasheets/GPS/Modules/LS20030~3_datasheet_v1.2.pdf>.

[2] Bluejay, Michael. "Rechargeable Batteries — Compared and Explained." *Rechargeable Batteries Explained in Detail (NiMH, NiZn, NiCd,RAM)*. Michael Bluejay, 1999. Web. 14 Nov. 2013. <http://michaelbluejay.com/batteries/rechargeable.html>.

[3] Di Pierro, Massimo. "web2py - Introduction" Complete Reference Manual, 6th Edition. N.p., n.d.. Web. 14 Nov. 2013. <http://www.web2py.com/books/default/chapter/29/01/introduction

THE ENGINEERS

Carlos Gonzalez Araujo is receiving his BSEE from University of Central Florida. He is presently working at Lockheed Martin in Simulation & Testing, on The Medium Extended Air Defense System (MEADS). He will be taking an offer with the Raytheon Company and will start classes towards his Master's in the fall of 2014 at the University of Arizona. His current area of interest is in Microelectronics.

Melissa Grenier will be graduating with a Bachelor of Science in Computer Engineering. She is currently an intern at The Boeing Company and will start as a full time employee upon graduation. Melissa's educational goals include continuing on to her master's degree within the next two years.

Gina McGhee is receiving a BSEE from the University of Central Florida. She is presently working at Lighting Partners of Central Florida as an intern responsible for Cad drawings and lighting layout. She is working towards obtaining a commission in the Air Force as a development engineer.

Sainjulien Senatus will be graduating with a BSEE from the University of Central Florida.