NextGen Asset Tracking Device

Ralph Baird, Lucas Dickinson, Brittney Fry, Wayne Marshall, Brianna Thomason

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

***Abstract* — Companies, not just people, lose objects. However, when a company loses objects it can add up to significant costs and losses for the company. This paper describes the design methodology to create a device that will be utilized in the industrial market to locate items that get lost, misplaced, or stolen from companies. The NextGen Asset Tracking (NAT) device is a device that will utilize location technologies to aid companies in reducing the loses from these lost items. What specifically will set this device apart from other tracking devices is its utilization of an Inertial Measurement Unit (IMU), as well as its specification for use in the industrial market.**

Index Terms — DC-DC power converters, Global Positioning System (GPS), Inertial Navigation, Internet of Things, Wide Area Networks

I. INTRODUCTION

An article in InCorp indicated that in the U.S. alone there is an average estimated range of $20 to $50 billion dollars in stolen assets or equipment by employees "making it one of the most costly and widespread challenges faced in today's business world" [1]. With the NAT device, if something gets lost or stolen, one will have the ability to utilize the deployed objects' associated NAT device, which can be located easily through a website application. The NAT device user will be able to create an account on the client website, as well as add and track the devices they have purchased.

This paper will discuss the designing, developing, and testing of a small sized, low cost, low-power device. This is ideal for things like construction equipment, shipping containers, airport equipment, military equipment, oil field equipment, computer communication equipment, and especially objects with no power themselves. Individuals and companies can easily lose these items, or take many hours diligently documenting their locations. This device would save companies money, time, and labor.

The NAT device has applications in many different markets, however it is designed specifically for the industrial market. The device is configurable, allowing a user to designate some features that may be beneficial to their specific application. While it is essentially a GPS tracker, there are many features that distinguishes the NAT from its competitors. These are described below.

II. NAT UNIQUENESS

A GPS tracker is not cutting-edge technology. There are many GPS trackers that come in many sizes with a wide range of features. What sets our device apart from other devices, such as the relatively famous Tile device, is that our device is developed for the industrial market whereas the Tile and similar products are developed for the consumer market. Our device integrates more components, such as the IMU and the radio module, which expands its functionality greatly. Also, our device is not necessarily going to remain close to the person who wishes to track it. Remote, real-time tracking over a long period of time is a necessary application. The biggest functionality that distinguishes the NAT from others is the INS technology that will be implemented in the device. While not implemented in this revision of the device, the Computer Science team that is taking control of the NAT development will design and write the INS algorithm that the NAT device will use as a backup to the GPS location. The INS technology is described later in the paper, but the current engineering team developed the device to a point where the Computer Science team just implements values that the firmware outputs from the IMU. INS technology on such a small application has not, to the knowledge of the engineers, been accomplished as of yet.

III. BASIC USE CASE

A company has an issue with its expensive or important items getting lost or misplaced by employees. It decides to investigate its options and finds the NAT device, a low cost, long life, dependable asset tracker. The company decides to move forward with a purchase order and purchases a thousand NAT devices, intending to connect them to the items that are frequently lost. In the purchase order, the company will designate what configuration settings of the options available to them they wish the devices to have.

The purchase order is sent to the manufacturer/distributer. They print the PCB and stuff the parts onto it. The manufacturer/distributer would also handle the placing of the device in its encasement, if necessary. Encasement was not a requirement of this portion of the development of the NAT device, but it is intended to be a part of the device. They would also be configuring the device through the developed configuration GUI. This GUI is a windows form that will be described in further detail later in the paper, but basically this software will configure the device to the various configuration values that the device allows, which were designated upon purchase of the device by the company. The manufacturer/distributer will manipulate the settings via the user-friendly form and then send the settings change to the device. The device will then update its configuration settings accordingly via the firmware.

Once all the devices have been configured via the purchase order, it is shipped to the user. The device is shipped with a reed switch, taking the device to the lowest power state possible.

The purchase order comes in, and with each item there is a serial number on the device as well as a 1-time use access code.

The user goes onto the associated website and, if this is their first order, creates an account. From their account they will begin to add their devices. To add a device the user will enter the serial number and 1-time access code of the device. This access code is intended to keep someone from stealing the device or guessing the serial number, and gaining access to information of a device that is not theirs. In addition to this, the user will designate a human-friendly name to the device, such as shippingContainer_12, NursesCart_level2_3, or even just brittneyFry_car. By adding this device, the device is now permanently associated with this client. The device is not intended to be able to be transferrable to other people, because while this may be preferable in the future, we do not see this as currently necessary. After the device has been added, the user can look on the tracking page of the website and see where their devices are most currently located at, along with some features to manipulate which devices are shown on the screen.

The company deploys the items with their associated NATs. The configuration that the company has set is motion trigger location reporting. Therefore, when they log into their account to view one of their items, they simply find the device through either its serial number or the human friendly name they designated, and can view its current location as well as every previous location that was documented every time the item moved with motion strong enough to set off the motion trigger. If they need to go collect that item, they can simply go to that location and pick it up!

IV. PROJECT GOALS

The project goals, in addition to the basic use case described above, were dictated by the sponsor/client to the engineers. The goals were to develop a small device that could read its location and report it back to a software component, which would display the location data as a pin on a map. The NextGen Tracker was to be an Internet of Things (IoT) device that is small, specifically around the size of a credit card, low-power, and utilizes both GPS location and location based on IMU (Inertial Measurement Unit) data calculated in an INS (Inertial Navigation System) algorithm. The goals and the overall use case were considered by the engineers and developed into the device requirements. These were then presented and approved by the sponsor/client. These requirements are shown in the section below.

V. PROJECT REQUIREMENTS

The engineers developed and documented detailed technical requirements, as well as deliverables needed to develop this device. In addition to the goals that were stated above and simply replicated into requirements, the team developed the functionality and requirements for the following software, a configuration GUI (Graphical User Interface), a client GUI, as well as the firmware that resides within the device. They were to integrate a radio communication device to communicate with the aforementioned software, it would operate on a 3.7V Lithium battery, it would provide GPS accuracy to 3m, and IMU motion accurate to 10 degrees. The scope of this project does not include the INS algorithm, but it provides the hardware and software needed for the motion values necessary to an INS algorithm that come from the IMU to be returned by the device and then easily implemented by the current Computer Scientists working on developing the algorithm.

VI. ARCHITECTURE DESIGN

Analyzing the goals and the now specified requirements, the engineers developed the architecture style that would be implemented for the hardware components. There are many ways that the system could be designed while meeting the project requirements, however the engineers decided that the best way to meet the size requirement was to develop an architecture that integrates all of the components onto a single PCB (printed circuit board), with everything soldered directly onto the device. During development of the device, this design had to be updated to include a last-minute design change due to radio module restrictions and updates. The final hardware design is still a single PCB, with a header for the LTE CAT 1M cellular radio module that was chosen for radio communication to be attached as a daughterboard into soldered on headers on the backside of the PCB. This is what was necessary for the selected component, but benefits the project as well because it allows for an easy update into better technology in the future, which is discussed later in this paper.

VII. HARDWARE SYSTEM DESIGN

The electrical design will be discussed in detail in the upcoming subsections, which include the various components that were interfaced together through the central digital logic to create the final NextGen Asset Tracking device. The components each subsection utilizes will be discussed, especially in regard to why they are necessary to meet the project goals and requirements, and which component was ultimately chosen to meet the functionality necessary for each subsection and why.

In Fig. 1 below, the electrical block diagram is shown, which is a simplified version of our device schematic and includes all major components as well as the interface (SPI, I2C, UART) they communicate on.

An important note on the block diagram is that the RPMA module shown in the design has been removed (as discussed in section VI) and a charging circuit has been added. These changes are reflected in the appropriate subsection discussions.

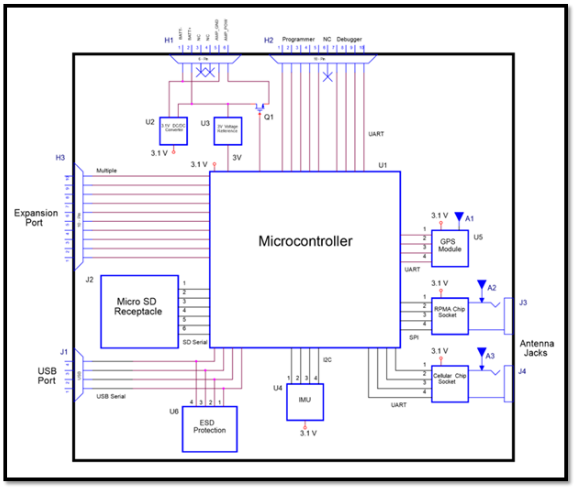


Fig. 1 Electrical Block Diagram with Connection Types

A. Central Digital Logic

Coordination and control of the data passed between the GPS receiver, IMU, and IoT communication modules requires central digital logic. Embedded systems may make use of specially-designed Application Specific Integrated Circuits, more flexible Field Programmable Gate Arrays, or more generalized microprocessor designs. The NAT device will instead utilize a microcontroller as the central digital logic, as it is best suited for the application as a low power IoT device.

The PIC24FJ128GC006 (PIC) was chosen as the main digital logic because it is a high performance, low power, low cost 16-bit microprocessor with very flexible IO pins. The 64 flexible IO pins allow for communications over UART, SPI, and I2C which are necessary for the components that are integrated into the NAT design, shown in Fig.1 [2]. Specifically to the NAT design engineering team, the PIC allows for In-circuit debugging (ICD) which the hardware necessary for was already available to the engineers through the sponsors lab.

The microcontroller is the heart of the device. It will function as the base for every component in our design. It is connected to all the components, and everything is, essentially, only connected to it. Its specific pins are configured to be connected to each component over the appropriate interface (SPI, I2C, UART). If a component needs to communicate with another component that is not the PIC, (such as sending GPS data over LTE), the PIC manages and handles this communication. The PIC will also manage low power modes, as well as USB (Universal Serial Bus) communication and data storage on the SD (Storage Device) card.

A key reason for choosing the PIC is that is has been designed for use in IoT products. It draws a minimal amount of power, which is great for an application such as the NAT device because our device ultimately hopes to be in the field for long periods of time.

All of the firmware will reside in the PIC, and the PIC will manage the configuration of every component as well, as it will parse the nat.conf file that stores its desired configuration settings. This file is discussed further in section VII subsection E Data Storage, as well as in section IX subsection B The Configuration GUI.

B. Location Data

The primary function of the NAT device is to track location via GPS or, in absence of GPS, using INS. Therefore, the main location component is the GPS. Analyzing the GPS in regard to the project's goals and requirements, the GPS is required to provide a location that is accurate within 3 meters, has a small form factor (as the overall device has a size limitation requirement) and is relatively low power (the device is meant to be in the field for a long period of time).

The direct functionality of the GPS component in the NAT device is that it will power on as appropriate (designated by the PICs power modes), it will read GPS location data, and it will send it to the PIC, which will then move the data along as appropriate (either to other components, the SD card, or onto the radio communication).

The device that was chosen was the OriginGPS NanoHornet (ORG1411). This component meets all of the needs the NAT device requires, as well as some beneficial properties. It has a small size with an embedded antenna, which aided in the overall design because an external antenna did not need to be researched, integrated, and tested with the GPS module. It has an accuracy within 2.5 meters, well within the required range. It has a quick time-to-first-fix (TTFF), which will be beneficial to users who are waiting on their device's location. It's power range is well within the rest of the components chosen, has autonomous operation, and has self-managed low power modes, all beneficial to the NAT device's operation. [3]

The NextGen Tracker will also utilize the integrated IMU to implement INS tracking in a later revision. The NAT device will not be market ready on the completion of the initial project scope. A team specializing in Computer Science will then take what has been accomplished with the NAT device and develop an INS algorithm. What this algorithm will do is utilize the data being read from the IMU (force, angular rate, and magnetic field) and the most recent location data (from the GPS, stored on the SD card), calculate where the devices new location is, and report that to the user just as it would report the GPS value directly. The specifics to the IMU will be discussed further in the document, as it is utilized for another feature as well.

C. Communication

The NAT device specifically needs communication technology to remotely send the user software data in order for the software to report the devices location to the user via a pin on a map as well as the raw latitude and longitude data. This was initially going to be accomplished through two separate components utilizing two distinct communication technologies, NB IoT (Narrowband Internet of Things) and RPMA (Random Phase Multiple Access). These technologies are cutting edge and were ideal for the NAT device application. RPMA is an LPWA (Low Power Wide Area) technology, perfect for embedded devices on the Internet of Things. It would run on the unlicensed 2.4GHz band, meaning it would be very cheap to communicate on. However, this technology and its current developers are showing signs of difficulties that may be fatal to the technology and the company itself, and therefore the engineers researched and developed an alternative design. The other technology that was cutting edge and very interesting to our application was NB IoT. NB IoT is also an LPWA technology that was designed to work virtually anywhere. It connects devices and operates independently or on existing cellular base stations that can allocate a resource block to NB-IoT. NB-IoT was also designed to handles small amounts of 2-way data transmitted infrequently, securely, and reliably. This would have been ideal for our application, however NB IoT has not been deployed substantially in the United States region yet, therefore a reliable communication technology for the region had to be chosen instead.

When the engineers discovered Digi Xbee radio modules, a possibility for an easily updated design was significant. The Digi modules have a 20-pin form factor across all of their modules. Therefore, when NB IoT arrives and is implemented significantly and reliably in the region, Digi will develop a module that communicates on NB IoT technology, and updating the device in a future revision is as simple as swapping the module and updating the firmware (if even necessary).

The final decision on which communication technology to integrate and implement was LTE (Long Term Evolution). It is supported in the region that the development took place, and the technology is reliable and significant. However, the downside is that it is not optimized for an IoT type of device, specifically in that it was not designed to be low power. The future upgrade to NB IoT will be better suited for the device overall, specifically benefitting a long battery life.

The specific module that was chosen was Digi Xbee LTE CAT 1M. In addition to availability, Digi assures that changing the cellular module from LTE Cat-1 to Cat-M or NB-Iot will have little to no impact on the design, which leaves room for future improvements and exploration of technologies that the engineers were initially interested in.

D. Motion Sensing

The IMU allows location to be calculated when GPS-signals are inaccessible. However, IMU data is applied elsewhere as well. The NAT intends to use the motion readings to indicate to the device that the object it is attached to is moving, and the location readings should begin. Such a case was described in Section III Basic Use Case. The IMU itself is an electronic device that calculates a body's exact force, angular rate and occasionally the magnetic field surrounding the body. It does so by using a combination of accelerometers, gyroscopes and magnetometers.

Specifically, the device has two reporting types. It can either report every time it is moved and creates a motion value greater than the one designated by its configuration, or it can report on a frequency basis, where the location will be reported either daily, weekly, monthly, or yearly. The IMU is necessary for those devices set to report location on a motion trigger, the specific values that prompt the location to be read are set in the configuration GUI software, discussed in section IX Subsection B Configuration GUI. This, in addition to the eventual INS software, will be what the IMU is utilized for.

The IMU component that was chosen had to read values with an accuracy of 10 degrees on angular motion, as well as attempt to be as low power as possible while maintaining a small size due to the restrictions set in the requirements at the start of the project. The IMU component that was ultimately selected was the MPU-9250. The MPU-9250 is a 9-axis device that tracks motion because it combines a 3-axis gyroscope, 3-axis accelerometer, 3-axis magnetometer and a Digital Motion Processor™ (DMP) all in a small 3x3x1mm package. The contents and size of this IMU is ideal for this project. [4]

The engineers have designed and encoded the IMU values to be read and outputted easily so that the future Computer Science teams can develop the INS algorithm, integrating the IMU values with ease.

E. Data Storage

The NAT device will have a mount for an external SD card for a few reasons. Initially, the device can be configured to store all GPS locations and IMU motion value readings in the SD card. Once the INS algorithm gets developed, those location values will also get stored in the SD card. The data being stored on the SD card is important because the INS algorithm needs the previous location value to complete its algorithm. The values will be read from the SD card by the PIC microcontroller, and then distributed where necessary, mostly in use by firmware or sent to the LTE module to then be sent to a software component.

```
[Date and Time]
Day: 5 ;1-31
Month: 2 ;1-12
Year: 2017 ;2017-3000
Time: 1535 ;0000-2400
[Interrupt]
Wake-Up Interval: True ;true/false
Month: 2 ;0-12
Day: 5 ;0-28
Weekday: UMTWRFS ;UMTWRFS/U/M/T/W/R/F/S/*
strt_Time: 0100 ;0000-2400
stp_Time: 0200 ;0000-2400
[Motion Trigger]
Wake on Motion: True ;true/false
Magnitude: 20 ;0-9999
Duration: 15 ;0-9999 milliseconds
Dwell: 15 ;0-9999 seconds
[GPS]
GPS Record: True ;true/false
GPS Poll: 2 ;2-30 minutes
[IMU]
IMU record: True ;true/false
INS record: True ;true/false
Record: MotionTrigger ;Continuous/MotionTrigger
Sample Rate: 10 ;1-100 samples/second
[Server]
Location: warthog.eastus.cloudapp.azure.com ;
END
```

Fig. 2 The nat.conf file the device uses to set configuration settings

The nat.conf file, shown in Fig. 2 above, is stored in the SD card as well. This file stores the configuration settings of the device, and there will always be a copy residing in the device's memory via the SD card. The firmware will parse this file and setup the device and its components accordingly. The file is only ever updated by the configuration GUI. The GUI will allow the distributer/manufacturer to set the device's configuration settings as designated by what the user requested. Once the distributer/manufacturer sets the values, they will indicate to the configuration GUI to send the values to the device. The GUI will update the nat.conf file that it had previously downloaded from the device's memory with the new designated values, and then return the file to the SD card to overwrite the existing one. More functionality on this will be discussed in the section IX subsection B Configuration GUI.

The way the configuration GUI reads files from the SD card is that the device is to be set so that when it is plugged in, via USB, to a host computer the computer will recognize it as a Mass Storage Device (MSD) and allow the host computer to see its files. Therefore, since this is possible, the SD card on the device is also available to the user as an MSD, so they can view the files stored on the SD card directly with ease.

The engineers chose the Position Card Connector Secure Digital - microSD™ Surface Mount, Right Angle Gold for our micro SD card mount. This is due to availability and ease of use and testing.

F. Computer Interface

The NAT device has a microUSB port in order to connect the device to a host computer for configuration, as well as allowing for charging capabilities via the charging circuit embedded in the device.

The NAT device utilizes the component USB - micro AB USB 2.0 Receptacle Connector 5 Position Surface Mount, Right Angle for the same reasons that the SD card mount was chosen, availability and ease of use and testing.

G. Power Circuit

The DC/DC converter is the main component of the power system for the entire device. The NAT device utilizes a Texas Instruments TPS62200x switching voltage regulator that steps the 3.7 volt input battery voltage down to the necessary 3.1 volt operating voltage. The 3.7 volts is a standard lithium ion battery voltage, but it is not the desired operating voltage of the components that are integrated in the NAT device, and the battery voltage can also vary widely, which creates noise in the system. For these reasons a DC/DC converter is needed to create a stable operating voltage.

The TPS62200x was chosen in particular because of its high efficiency, high output power and low power operation under light loads. In addition, the Texas Instruments label brings reliability, support and lower costs than competitor devices.

The other significant piece of the power system is the lithium ion battery charging circuit. Since the engineers hope that the device lasts as long as the user needs it to, occasionally the battery of the device needs to be charged. The NAT device will allow the user to access this charging circuit via the embedded micro-USB port the device already integrated for other purposes.

The device makes use of Microchip MCP73831. This chip allows the device to operate while being charged and extends the operating lifetime of the device without adding maintenance costs.

H. Other

There are a few more components that are interesting to the electrical design, that do not constitute any subsection of the device specifically.

The NAT device implements a reed switch, which when a magnet is placed onto the switch, the connection to the power is removed until the magnet is removed from the switch. This is beneficial in situations such as transporting or shipping the device, intending to conserve the battery that it is shipped with.

There is an ESD (Electrostatic Discharge) protection unit integrated to protect from any issues a sudden surge in electricity would cause our device.

IX. SOFTWARE SYSTEM DESIGN

There are 3 software components that have been developed in addition to the corresponding hardware discussed above. These are the firmware (software on the device), the Configuration GUI (software to configure various device and component settings), and the Client GUI (software for the user directly). The engineers deemed these 3 components necessary for full use of the NAT device, and developed them according to the functionality the sponsor/client described was necessary. These will be analyzed in detail below.

A. Firmware

The MCU software will be a structured program, with execution parameters defined in a configuration file on the mass storage device. In operation, a loop will execute, reading the values from the GPS and IMU, logging the sensor data to the mass storage device and transmitting the data over the cellular links, logging success or failure. Mature peripheral drivers provided by the MCU vendor will be used wherever possible to reduce the likelihood of faults in the bulk of the code. The operational loop will be controlled by the power-saving profile defined in the configuration file, either waking up based on a timing interval, or via motion trigger.

The NAT device firmware components are defined as a sequence of modules, each servicing a separate peripheral according to its specific interface. The USB library being used requires a cooperative multitasking environment, and so each module will execute under the expectation that it will cede control to the next module when it has nothing to do at the present time in an infinite loop, broken only by transitions to low power mode and returned to by configured timer or peripheral interrupts. Care will be taken in implantation that no long-term blocking code is used, deferring to the next loop as often as necessary for the smooth operation of all firmware modules.

The main NAT device firmware runs when power is first applied and whenever the device is awoken by motion or real-time-clock interrupts. The microcontroller hardware is initialized, with each peripheral power enable pin configured for output and asserted. Each piece of hardware is queried, and any errors or failed responses are indicated using up to three independent debugging indicators: lines appended to the debug log on the SD card, descriptive text transmitted over the debugging UART, and predetermined blink sequences on an LED. The device configuration is then read from the SD card and applied. After configuration, data is read from the GPS and IMU and logged to the SD card. The current firmware can only obtain position data form the GPS, but later expansion is expected to use an inertial navigation system (INS) to deduce position from IMU data. Once the position is known, a packet of data is sent to the cellular data module, which then transmits the device ID and position to a centralized server. Once this is completed, the firmware switches to low-power mode, de-asserting the external peripheral power pins. If voltage is sensed on the USB power line at any time, the device switches to USB Mass Storage Device (MSD) operation, allowing a new configuration file to be written, and device logs read on the configuration workstation. Normal operation is then resumed once the USB power line is disconnected.

B. Configuration GUI

As previously discussed, the configuration GUI is intended to simply set the various configuration settings within the NAT device which are able to be designated by the user. The GUI is a windows form, accessible only to the person initially turning it on, meaning the manufacturer/distributer. It is not intended to ever be distributed directly to the user, as once the user’s configuration is set, it never has to be changed or reloaded onto the device.

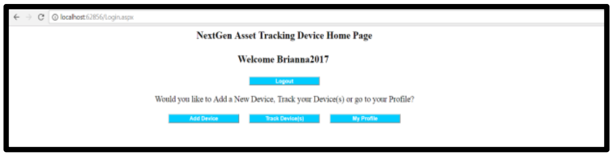
The first thing the GUI will do is attempt to find the device through its connection as a MSD, which is described in Section VII Subsection E Data Storage, as well as Section IX subsection A Firmware. If it is able to locate it from the removable devices on the host computer, the GUI will open and show that the device has been connected. However, until the GUI can find the device the GUI form will not open. This is set because without the device, the GUI is not usable. Until the device is connected and found the GUI will continuously cycle looking for the device, until either it finds the device, or the user chooses to exit the GUI. Once it’s connected, the user will see all of the settings the NAT device currently holds according to the nat.conf file it downloaded from the NAT device once it was successfully connected. They will be able to change a variety of settings. Once they change the values they want to, they will hit a button, “Program Device”, and the file will be written and sent to the device so the device can overwrite the file it has in storage and then the device’s settings can be parsed by the it's firmware and the appropriate settings set.

This is the essential functionality of the configuration GUI, and other functionality can be added easily as the device is revised because of the file passing structure and the software structure of the GUI. Just simply add the new settings to the file, add them to the parser in the configuration GUI and firmware, and add the appropriate objects in the GUI form.

The Configuration GUI is necessary to our device because it was designed to allow a few configuration settings to be set by the user to make the device as ideal as possible to their application. The possible settings able to be changed are as follows. One of the main configuration settings designates what triggers the GPS location to be transmitted. The possibilities are either through location frequency, meaning that the user will designate what time frame the device will turn on and send the GPS location to the appropriate location. This type of option is good for an application that does not move often, but is always in reliable GPS coverage. The user will designate the daily, weekly, monthly, or yearly frequency and the values associated with that choice. The other option is motion trigger location retrieval. This option will have the embedded IMU listen for motion of the device, and if a strong enough motion is detected, the location is read and returned to the user. This is better for applications that move more frequently, and may need to utilize the backup INS location service. This is because since the device needs the IMU to have at least low power at all times since it is necessary to detect the motion trigger, it will also have it available to log the motion values necessary to calculate the location via INS. The user will designate the threshold for the values that the IMU can read to activate the motion trigger. Other more minor configuration options are setting sample rates of some of the components, as well as which values are to be saved to the SD card during operation. The file that stores all of this is shown in Fig. 2. As described previously, the nat.conf file is stored in the memory of the NAT device on it's associated SD card, and downloaded into the GUI upon opening of the form if the device has been successfully connected.

C. Client GUI

The client GUI (graphical user interface) is currently in the form of a website, with a future scope of being developed into a mobile application. The client will be able to create a personal account which will contain the current location, along with the previous location history of any device the user registers. Once logged in, the user will be prompted to select one of three options, which are to go to "My Profile," "Add a Device" or "Track a Device," shown in Fig. 3 below.



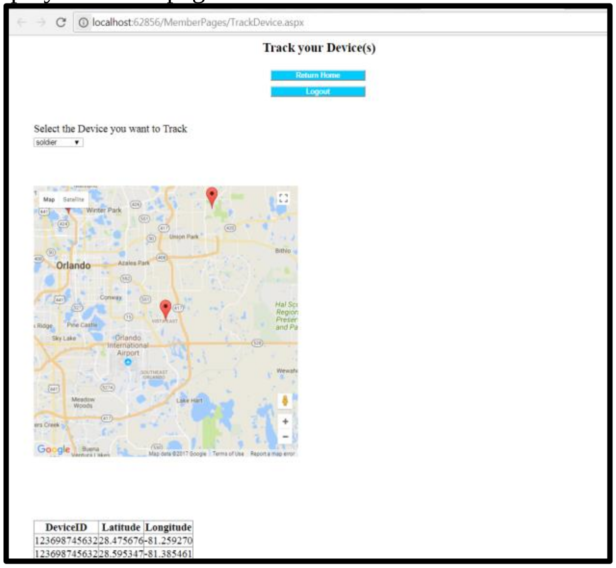
NextGen Asset Tracking Device Home Page

Welcome Brianna2017

Would you like to Add a New Device, Track your Device(s) or go to your Profile?

Fig. 3 The User page seen after logging into the NAT website

Under the profile page, the options to edit username or password are available and a list of the user's current devices will be displayed. The option to add a device will prompt the user to enter the new device's serial number, the one-time access code and to nickname the device (previously described as a human-friendly name in Section III Basic Use Case). The list of current devices will be displayed on this page as well.



Track your Device(s)

Select the Device you want to Track

DeviceID	Latitude	Longitude
123698745632	28.475676	-81.259270
123698745632	28.595347	-81.385461

Fig. 4 The Tracking Devices page of the Website

Under the track a device page, as seen in Fig. 4 above, the client will see the list of current devices with the option to choose one to track. Once selected, that specified device will be shown on the map, along with the latitude and longitude coordinates of its current location. In addition, there will be a list of all the previous locations of that device that can be viewed.

There is the main database for the client GUI which will hold the account information for the users, along with their registered devices. It will also have a separate table used for holding the latitude and longitude coordinates of each device. Once a location is found, the coordinates are added to the list of location history for that device that is viewable to the user on the tracking page when that device is selected.

X. FINAL BOARD DESIGN

The device is integrated on a four-layer PCB designed using the Cadence suite of software: OrCad Schematic Capture and Allegro PCB Designer. The top and bottom layers of the board are data and signaling traces, the second layer is the ground plane which is placed below the top layer for shielding and the third layer is the power plane which is split into three sections: The battery voltage (3.7 V), the operating voltage (3.1 V) and the USB power input. All of the passive and discrete components are placed on the top layer for ease of application by a pick-n-place machine. However, the three external connectors to the board are place on the bottom: the power connector where the battery and signaling pins connect, the debugging/programming header and the connection to the XBee daughter board.

The Client

The NAT device was a project idea presented to the engineers by Michael Young with his company Young Engineering Services LLC. The client presented the project as a set of goals and a basic use case, intending for the engineers to develop many of the requirements and specifications for the device, with his final approval. Young Engineering services handled the financing of the project, as well as provided a space for the engineers to design and develop the project. Young Engineering Services and the associated lab the NAT engineers worked in is located in Winter Park, Florida.

The NAT Engineers

Ralph Baird's undergraduate career has taken many turns since enrolling in Brevard Community College (now Eastern Florida State College) After several changes in focus from information technology to programming to engineering, he completed two college credit certificates and two Associate's degrees before being accepted to the University of Central Florida and majoring in Computer Engineering, for which he will earn a Bachelor's degree in December 2017. He plans to apply high engineering standards to software development with interests across the board.

Lucas Dickinson is Computer and Electrical Engineering student graduating in the Fall of 2018. He is currently interning at Leidos, a technology company with a facility located in the research park. Lucas has plans to continue into the engineering workforce before returning to UCF for his Masters Degree in Electrical Engineering. His interests are in signal processing, embedded systems, computer architecture, and system design.

Brittney Fry is a 22 year old Computer Engineering student graduating in Fall of 2017. She has interned for Cummins and Intel, working in mostly software development. She has accepted a full-time offer starting in 2018 from Cummins as a Vehicle Electronic Systems Integration Engineer, working on developing automated testing software for the integration team.

Wayne Marshall is an Electrical Engineering student graduating in the Fall of 2017. He has an interest in embedded systems. He currently works at World Class Installations, and has been offered a full-time position here with the Research and Development team upon graduation.

Brianna Thomason is a Computer Engineering student graduating in the Spring of 2018. She intends to pursue a career in the field of government or defense with a focus in software or hardware. She spends her time volunteering in her community and serving at her church.

Acknowledgements

Thanks to Michael Young for sponsoring our project through his company Young Engineering Services, LLC. We appreciate all of the guidance and assistance you have given, as well as access to countless resources. We also acknowledge the assistance, support, and guidance of Dr. Samuel Richie and Dr. Lei Wei. They have aided greatly in answering all questions for both the actual engineering as well as assistance in the management of the NAT team. We finally acknowledge the assistance of QMS services, specifically Sam Hanna, who aided in placing the components with a very small pitch on our device with their Pick-n-place machine. They did this without compensation, and we appreciate their services.

References

[1]"How Big of a Problem is Employee Theft and Fraud?", Incorp.com, 2017. [Online]. Available: https://www.incorp.com/help-center/business-articles/employee-theft-and-fraud-part1. [Accessed: 15-Jun- 2017].

[2]“16-Bit Flash Microcontrollers with ... - Microchip Technology.” [Online]. Available: http://ww1.microchip.com/downloads/en/DeviceDoc/30009312d.pdf. [Accessed: 16-Nov-2017].

[3]2017. [Online]. Available: https://www.origingps.com/wp-content/uploads/2017/06/Nano-Hornet-ORG1411-Datasheet-Rev-3.4.pdf. [Accessed: 05- Jun- 2017]

[4] S. MPU-6050, S. MPU-9250, S. ITG3200/ADXL345, 3. MPU-6050, A. R3, R. 3, B. Straight, S. Bi-Directional, L. v3 and A. R3, "SparkFun Triple Axis Accelerometer and Gyro Breakout - MPU-6050 - SEN-11028 - SparkFun Electronics", Sparkfun.com, 2017. [Online]. Available: https://www.sparkfun.com/products/11028. [Accessed: 29- Jul- 2017].