Initial Project Document

NextGen Asset Tracking (NAT) Device

Group 1 – NAT

Team Members

Brianna Thomason	**Computer Engineer**
Brittney Fry	**Computer Engineer, Project Manager**
Lucas Dickinson	**Electrical Engineer, Computer Engineer**
Ralph Baird	**Computer Engineer**
Wayne Marshall	**Electrical Engineer**

Sponsor(s): ***Young Engineering Services, LLC***

Under the direction of Michael F. Young, ***UCF CS Adjunct Professor***

and

Michael McAlpin, ***UCF CS and IT Adjunct Professor***

Project Narrative

This project is about the designing, developing, and testing of a low cost, low-power, next-generation asset tracking device.

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 employee "making it one of the most costly and widespread challenges faced in today's business world" (InCorp, 2017). Why continue to take the risk associated with manual tracking of your assets or equipment. In today's advanced world where Internet access is at everyone's fingertips, it should be just as easy to locate your equipment by tracking it in real time. A 24 hour real time tracking device for equipment comes with many benefits. One will have peace of mind knowing where their equipment is at all times. With our NextGen Asset Tracking (NAT) device, you will have the ability to track your equipment whenever you need to. If your equipment gets lost or stolen, you could easily locate it. This would save you not just the money you would need to replace it but also the time you would spend looking.

With the NAT device, if something get lost or stolen, one will have the ability to deploy objects that can be located easily. 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. The device will also be useful in tracking packages that are being shipped, not just the vehicle it travels in, and in tracking people for use in law enforcement and private investigations. Obviously, GPS is not new technology, and neither are GPS tracking devices. Current trackers exclusively use GPS, very expensive hardware, and use LTE to send messages. This is very expensive and has a very negative effect on battery life. Some cell phones have GPS tracking built in but we are not competing with cell phones. Cell phones can cost anywhere from $700 to $800 and therefore would not be a very cost effective tracking device. The table below highlights some of these devices and their features:

Device	Real-Time	Battery life	Price	Monthly Fee
STI_GL300 Real-Time GPS Tracker	✓	2 weeks	$70	--
GX350 Real-Time GPS Tracker	✓	2.5 weeks	$100	--
XT-2000 OBD Real-Time GPS Vehicle Tracker	✓	Powered by vehicle battery	$120	$25
STI_Bolt Asset Tracker	X	9 months	$130	$14.95
Tracking Key GPS Logger 2	X	20 hours	$170	N/A
STI_GL300 Real-Time Tracker w/ 6 Month Battery & Case	✓	6 months	$230	--

The objective of our project is to design and build a low-cost IoT module that can be attached to anything and report back its location to the appropriate party. It shall primarily do this with GPS, however it will have a secondary position based on Inertial Navigation System (INS) when GPS signal is not available, or is suddenly lost. The secondary goal, potentially for future revisions past the scope of this project, is to have the hardware inputs allow for a variety of additional sensors that can monitor and report back on a myriad of conditions.

One way the NAT device differs from other devices is that it will have INS as a backup when GPS signal is lost. While others only use cellular networks to report data, we will also use a new wireless technology from Ingenu called Random Phase Multiple Access (RPMA). RPMA is a technology communication system that uses direct-sequence spread spectrum (DSSS) with multiple access. It utilizes the globally available and free 2.4 GHz band. In addition to RPMA, the NAT device will use a newer leading edge cellular network technology designed specifically for machine to machine communication called Narrowband-IoT (NB-IoT). Both RPMA and NB-IoT are Low Power Wide Area (LPWA) technologies which will contribute to our device having low power consumption, giving us longer a battery life. The module we will be designing will also be able to take analog or digital inputs, or be able to produce either analog or digital outputs.

Requirements Specifications

General Requirements

- Shall produce 1 PCB as well as supplemental software
- Shall produce a product of size smaller than 5.370 in ×4.125 in and less than ½ in thick
- Shall operate on a single 3.7V battery
- Option to operate on a 10V to 30V external DC power source
- Utilizes RPMA Wireless data link
- Utilizes NB-IoT wireless data link
- Input taken as motion, GPS, power
- Output data to the radio modules
- Interface to the Ingenu wireless module
- Interface to a licensed NB-IoT radio device
- TTL serial data interface for design and debugging
- USB interface to configuring the device
- Interface to the PIC processor to program it and for debugging
- Optional I/O support inputs from a variety of external sensors using an I2C, SPI, contact closures and/or analog inputs/outputs

Device Hardware Components

- Low power 16-bit MCU
- GPS receiver
- 9-Axis motion sensing device
- RPMA radio module
- NB-IoT radio module
- Micro-SD module
- Ability to record data
- Implement an antenna
- Interface to external devices using contact closures, I2C, and SPI

- Able to switch DC power to external devices using software.
- Able to run for at minimum some minimum time frame based on the client's specifications

Software Modules and Programs

- Able to configure the device through HID interface
- Option to be able to configure client specific settings through a web browser for NB-IoT
- Option to be able to configure client specific settings through a windows based GUI
- Windows based GUI to configure the device
- Able to read and decode the motion sensing hardware data
- I2C interface with GPS
- SPI interface with RPMA Radio module
- Interface to NB-IoT radio module
- Software to create the data packet to be sent on the Ingenu network
- Software to encrypt and decrypt the packets to and from the Ingenu network
- Windows GUI for retrieving data from the Ingenu network servers
- Windows based positioning software that tells user where selected devices are located
- Optional INS system algorithm that takes over when GPS system is lost
- Optional Unique INS system software
- Software interface from NB-IoT module to cellular carrier

Constraints

- Current INS algorithm does not exist for embedded microprocessor
- INS has not been implemented in such a small scale device as of yet
- Time to implement all requirements with all of the members schedules and other time constraints
- 16 bit bus may not be powerful enough with 32GHz
- Battery life may not work with power draw from applications of our product
- Sampling rate may not be fast enough, current draw of the sampling may be too much for power

Standards

- All wireless devices used will be FCC certified
- When using the Ingenu wireless link: the licensee-free 2.4 GHz band as per FCC, 47 C.F.R, Part 15
- When using cell systems: licensed radio band held by the cell provider (Verizon, AT&T, ect) FCC, 47 C.F.R, Part 22
- Certified unintentional radiator

House of Quality

House of Quality - NextGen Asset Tracker									
			Technical Requirements						
Battery Life				↓	↓	↓	↓	↓	↓
Cost					↓	↓	↓	↓	↓
Accuracy						↑	↑	↑	↑
Reliable & Error Free Data link RPMA									
Reliable & Error Free Data link Cell Network									
Rotational Vertical & horizontal Position									
Compass									
			Battery life	Cost	Accuracy	Reliable & Error Free Data link RPMA	Reliable & Error Free Data link Cell Network	Rotational Vertical & Horizontal Position	Compass
			+	-	+	+	+	+	+
Marketing Requirement	Low Cost	-	↓	↑	↓	↓↓	↓↓	↓↓	↓↓
	Long Battery Life	+	↑	↓	↓↓	↓↓	↓↓	↓↓	↓↓
	Small Size	-	↓	↓					
	Acceptable Accuracy	+	↓↓	↓	↑	↑	↑	↑	↑
	Real Time	+	↓↓		↓	↓	↓	↓	↓
			> 3 Months	<$40	<3 Meters	Pass/Fall	Pass/Fail	<10 degrees	<10 degrees

Project Block Diagrams

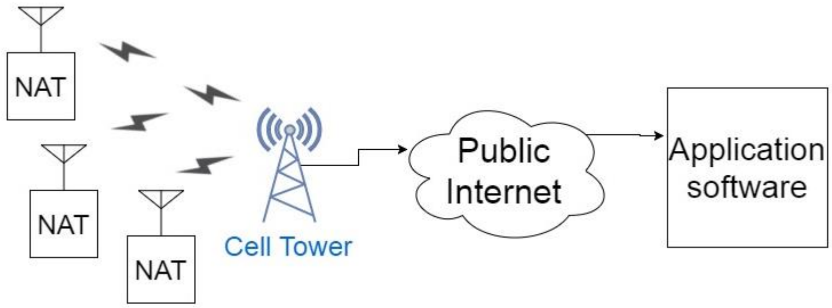


Figure 1 – Network System Diagram using NB-IoT

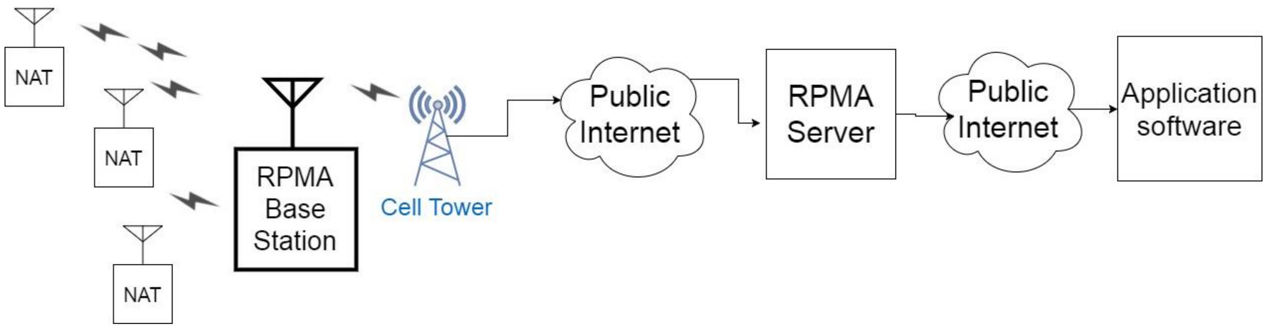


Figure 2 – Network System Diagram using RPMA

Figure 1 and Figure 2 show the system communication diagram, which depicts how our system will connect to other systems via global networks. Two options we have posed are communicating through NB-IoT (Narrowband Internet of Things), a low power wide area network that uses cellular communication bands, existing with current 2G, 3G, and 4G mobile networks, or through RPMA (Random phase multiple access), which is also a low power wide area network that was made to connect machines to machines through the IoT, shown in figure 1 and 2 respectively. Figure 1 depicts that with NB-IoT, the instances of our device (NAT) will communicate directly with the cell towers. This option is simpler, and yet still a plausible implementation. We plan to use NB-IoT technology since our data throughput requirement is so low, and this system handles that feature well. Figure 2 shows multiple instances of our system (NAT) which have their own built in antenna, speaking to the RPMA base station which then connects further to the cell towers. Research is currently being done into both systems for our application, and additionally we would like to research and test the option of implementing both options.

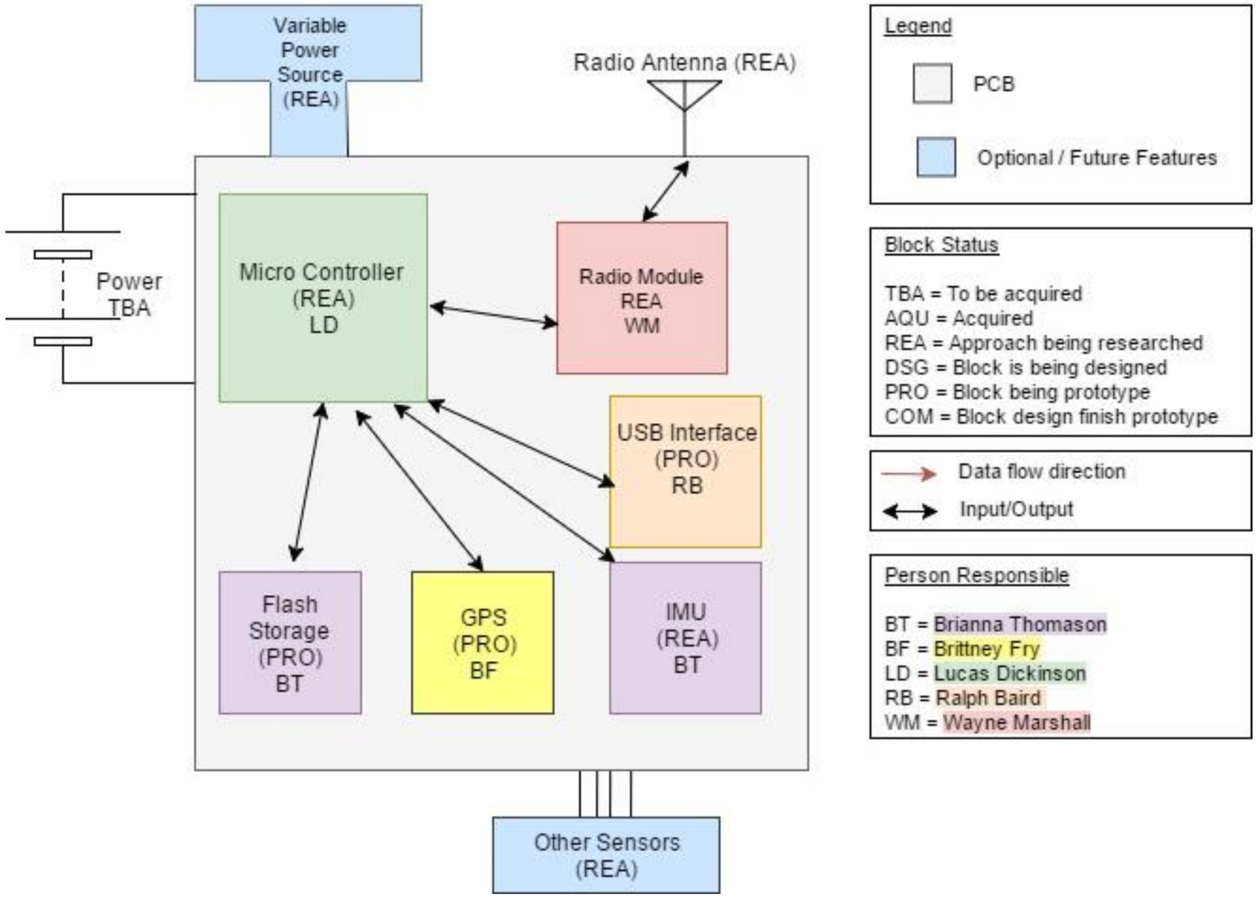


Figure 3 - NAT System Block Diagram

Figure 3 shows the NAT system block diagram, which depicts the hardware modules that will be implemented. We will be utilizing some microcontroller, which will be able to support the different communication interfaces as well as consume the lowest amount of power. This microcontroller will take input and give output to all of the components in the system. It also will be where the software that the CpE (Computer Engineering) team writes will be stored and ran. The radio module, which may provide RPMA communication, NB-IoT communication, or both. The IMU (Inertial Measurement Unit) that will output values of or related to acceleration, magnetic field, and orientation. These values will be sent to the microcontroller, so that the INS (inertial navigation system) algorithm can be ran on the values. The algorithm is the responsibility of the CpE team. The GPS module will read the current location of the device, using its internal hardware. It takes only input from the satellites that its antenna reads. The flash storage will store the value of every location the GPS module reads. All modules only input and output internal to the PCB via the microcontroller. All other data that needs to be exchanged to modules other than the microcontroller due so via the microcontroller. Such as the IMU needs the values that the flash storage has stored, and gets such via a request to the microcontroller. The other sensors module will be an optional expansion bus, which may read any variety of values that the user could want and output those alongside the location values.

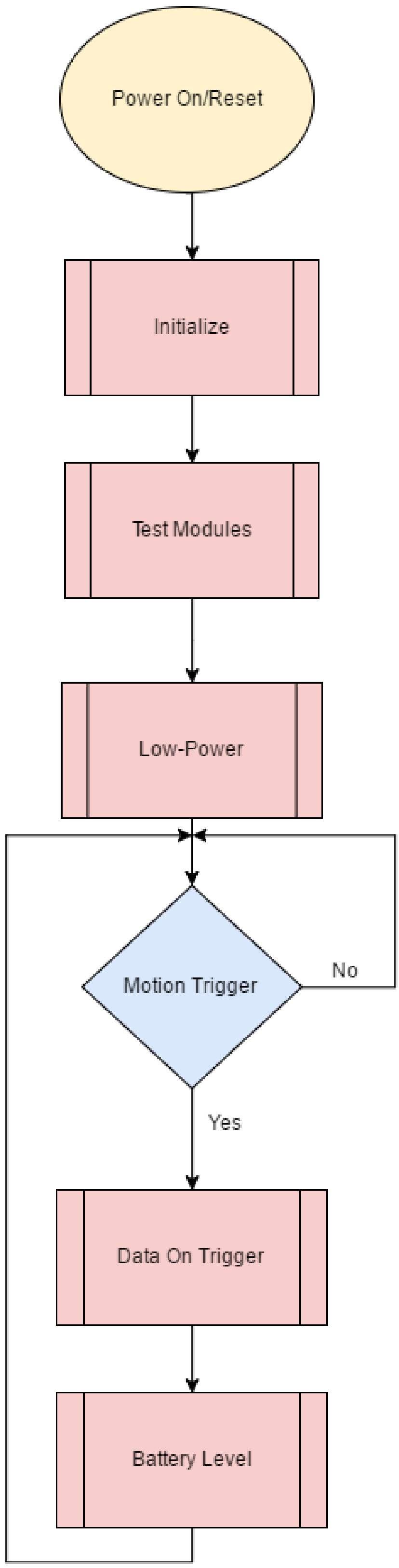


Figure 4 - High Level Software Flow Chart

In relation to figure 3, the CpE team (Ralph Baird, Brianna Thomason, and Brittney Fry) will be responsible for the software inside of the device, stored in the microcontroller. The CpE team will write all of the interfaces that the hardware needs to function as well as the applications that the user can use to see the output (devices location and other values such as battery power) nicely in the form of a Graphical User interface (GUI).

The main software that the NAT will run will be the Power On/Reset module. This is shown to the left in figure 4. Upon power on the module will initialize the variables that the device needs throughout operation. Then it moves down and tests every module so to show that the system is completely up and running, no broken or faulty modules. The system then moves into low power mode, conserving as much energy as it can as to prolong the battery life as long as possible. The only thing it does while in this stage is look for a motion trigger. The motion trigger is given by the IMU module, which was shown in figure 3 above. Until it recognizes this trigger, the device sits and waits in low power mode. However, upon the trigger it moves into the function Data on Trigger. This function will utilize the other modules (mainly the GPS, flash memory, and the radio modules if the GPS is working and has a signal otherwise it will also utilize the IMU and the microcontroller to calculate the current location) to output to the user the important data (such as the devices location and power status). While in the function it will continue to loop and send data until a timeout shows the object has stopped moving. Then, since the trigger has stopped, the focus returns to the loop shown in figure 4, leaving the data on trigger function block. The device then checks the battery level to determine if it should move to an emergency battery conservation effort. If not, it returns to the infinite motion trigger check loop and waits for another trigger.

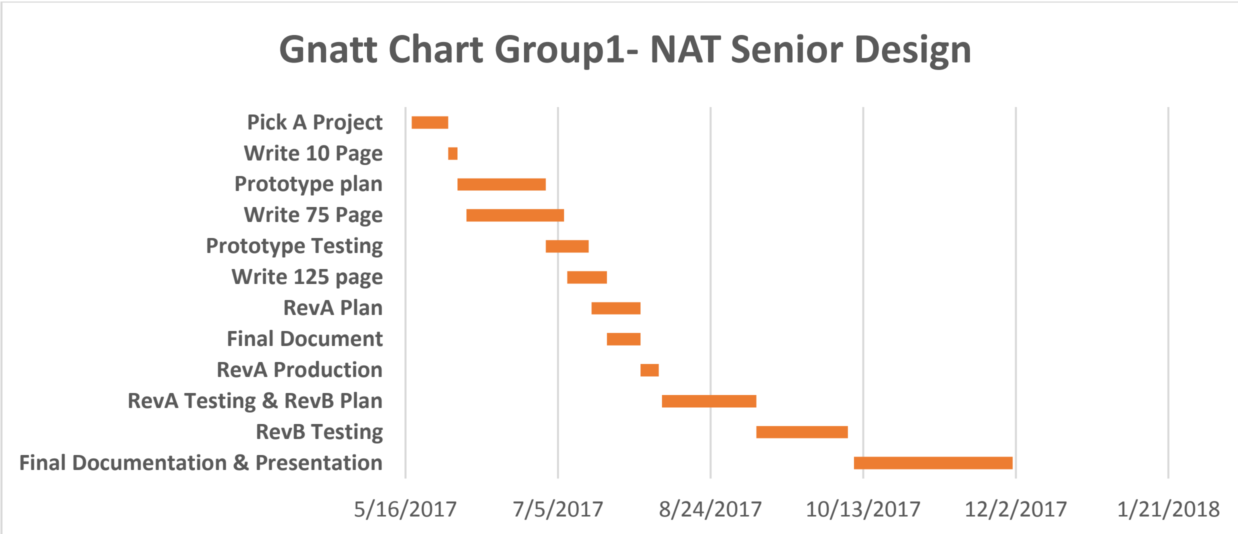
This is the main functionality. The device will output the its location after a motion trigger wakens it from low power mode, then output that location as well as other user designated relevant data (such as level of battery) over the radio modules continuously until it stops moving or the battery dies. However, decisions like whether to use GPS or INS, or transmit data through RPMA or NB-IoT are determined elsewhere.

Estimated Project Budget and Financing

Description	Estimated Price / Unit
Resistors/Capacitors/Inductors	$0.10
Pin Connectors	$0.75
Reed Switch	$1.00
GPS Modules	$25.00
Volt/Switch Regulator	$1.50
MICROSD Socket/Card Reader	$15.00
Voltage Reference	$1.50
IC MCU	$5.00
Motion Sensor IMU	$10.00
Crystals	$1.00
NB-IoT	$15.00
ESTIMATED TOTAL $95 / board	

The above is the estimated budget for one board to be designed and built. This project is sponsored by Young Engineering Services, LLC. The design group was given a budget of $1200 total. This budget includes the final design of 10 boards. Any part that is to be purchased by one of the group members has to first be signed and approved by Young Engineering Services, LLC, which is agreed upon in writing by each member. This will ensure that the group stays on budget and all the parts can be purchased.

Initial Project Milestone for Both Semesters



The milestones for the end of Senior Design 1

- GUI with basic functionality working
- Record on SD card and display 9 – axis Motion on GUI
- Read preliminary IMU data
- Send data through Ingenu network AND/OR NB-IoT network
- Understand INS and work has begun on algorithm
- Wiring diagram and schematic finished
- Tested with external wiring to produce rev0 board early next semester

Milestones for Senior Design 2

- Complete RevA PCB board early (1-2 weeks in semester)
- RevB PCB board tested past the point of confidence in functionality and have multiple deployed to demonstrate the covered range of device
- Develop a web browser to give real time location of device
- Device has basic INS functionality
    - Biggest issue with this will be power draw from processing power of IMU
- Get price point of device down to ~$40/unit when producing 1000units