Galahad: Real-Time Location Ranging Tag

Reinaldo Romero, Brandon Powell, Anthony Venkersammy, and Ernst Dormevil

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

***Abstract* — This conference paper will explain the different methods and concepts used for designing the real-time location ranging tag. Galahad is a real-time location tag that can be attached to most objects so that they may be located when misplaced. The device will be using ultra-wideband and Bluetooth low-energy technology to perform the ranging capabilities. Using a smartphone, the user will be able to locate the misplaced object by detecting the signal pulses sent by the tag and determining the range between the tag and the smartphone. The range-finding device will be able to make updates based on the real-time location of the smartphone.**

***Index Terms* — Ultra Wideband, Bluetooth Low Energy, Real-Time Systems, Microcontroller.**

I. Introduction

People have constantly experienced feelings of frustration, sadness, and disappointment after having misplaced an item. This was what motivated our group to create a project like Galahad. Galahad will help the user by significantly cutting down the time it will take to find the lost object. One very important goal that the group heavily considered was the size of the tag. In order to be attached to most objects, the tag will need to have a small footprint. The range-finding device makes use of popular technologies such as ultra-wideband and Bluetooth low energy. Galahad will be communicating with a smartphone capable of ultra-wideband ranging. The smartphone will receive the signal pulses provided by the range-finding tag. The range between the user and the device will then be displayed on the smartphone's screen. The smartphone that is used for testing the device is an Apple iPhone 13 since this phone has the U1 chip capable of doing ultra-wideband ranging. The Theice will be constantly updating the range from the phone since the user will need to know in which direction to head in order to decrease the distance to and arrive at the location of the lost item. Bluetooth technology will be used to link the smartphone and tag so that the ranging process may begin. The group has done market analysis to understand other devices similar to our range-finding tag. Products like the Apple AirTag, Samsung SmartTag, and Chipolo ONE Spot are what we want to model our device after. These devices make great use of range-finding technology to be able to track lost items.

II. System Components

Explaining the components used for the range-finding tag is an efficient method to better understand how the Galahad functions. This section will describe the different components along with what role they play in the device.

A. Microcontroller

The microcontroller that the group decided on using was the nRF52833 by Nordic. One key feature that this microcontroller offers is Bluetooth 5.3 LE. The nRF52833 can support Bluetooth long-range and Bluetooth mesh. The nRF52833 also offers 1MB flash and 256 KB RAM. This microcontroller has a supply voltage range of 1.7 volts to 5.5 volts. The QFN-40 package was chosen for the nRF52833. This package contains no leads which reduces the coupling and noise between the pins.

B. Ultra-Wideband IC

It was very important to consider an IC that would be compatible with Apple's U1 chip. This is a requirement because the phone we will use for the ranging is an Apple iPhone 13. The integrated chip that the group decided on using was the DW3110 from Qorvo. The DW3110 was intended for coin cell applications which is perfect for meeting our size constraint and the engineering requirement of using a battery as a power source.

C. Bluetooth Low Energy Antenna

Bluetooth technology is what the microcontroller in our device will use to connect to the smartphone. The group has decided to go with the 2450AT18A100E Bluetooth antenna. The frequency range of operation for this antenna is 2400 MHz to 2500 MHz. The peak gain for the antenna is 0.5 dB. The average gain of the antenna is -0.5 dB. The trace impedance on the PCB will need to be 50 ohms. The maximum input power for the antenna is 2 Watts.

D. Coin Cell Battery

One of the most commonly used battery types for tags is coin cell batteries. This is because it provides a sufficient amount of voltage and is relatively small in size. The most common coin cell battery used in the market today is the

CR2032. This coin cell battery is the one we will be using in our design. The CR2032 is one of the better-performing coin cell batteries on the market and is also one of the most affordable ones. The battery has a nominal voltage of 3 volts. It also has a nominal capacity of 210 mAh to 235 mAh. The height of the battery is 3.2mm and the length is 20 mm. The life of the CR2032 is 1 to 5 years. The battery weighs 3 grams to 3.2 grams.

E. Battery Holder

For the battery holder, the group considered an ultra-low-profile mount while still being easily accessible for the user. The group wants the battery holder to be accessible so that the user may replace the battery when necessary. The battery holder will sit on a hole cutout on the PCB. This will help the device be as slim as possible making it convenient to attach to most objects. The group has decided to use the BU2032SM-JJ-MINI-GTR. This battery holder is in stock in many places and is also one of the economic.

F. Capacitor Bank

In order to increase the pulse current so that it may be suitable for the ultra-wideband module, the group created a capacitor bank. The capacitor bank includes 5 capacitors connected in parallel with the coin cell battery. The value for the capacitors is 100 microfarads. The capacitors used for this capacitor bank are tantalum capacitors since they allow for a thin dielectric layer which results in a higher capacitance value per volume.

G. 3D Printed Case

The PCB will sit on a 3D printed case so that it may be protected, and the device can look stylish. The group wanted to make sure that PCB will not be damaged by normal wear and tear. This is why it was very important to consider which type of 3D printing filament was going to be used for the case. The filament we decided to use for our design is ABS. Economically, ABS is one of the most affordable options. Companies like Flashforge and Polymaker offer this filament at $0.62. It is also fairly easy to print which will make it easier for us to handle. ABS is one of the more durable filaments which can help our device handle most environments.

III. IEEE Standards

IEEE standards are developed so that developers can create technology that can be updated and innovated. The standards also provide guidelines to protect public safety, health, and well-being. The two major standards that were used for creating the range-finding tag were IEEE 802.15.4 and IEEE 802.15.1.

The IEEE 802.15.1 focuses on ultra-wideband technology and the two physical layers that are used. These two layers are the high-rate pulse and the low-rate pulse. For our design, the high-rate pulse is the one we will be focusing on since this physical layer sends pulses at higher repetition compared to the low-rate pulse physical layer. The high-rate pulse physical layer includes modes that provide reduced on-air time for higher density and lower power operation. The frames have a ciphered sequence known as scrambled timestamp sequence (STS) which increases the integrity and accuracy of the ranging signal. A device that offers these modes is referred to as an HRP-ERDEV. The HRP-ERDEV supports the transmission and reception of packets based on the position of STS in the PPDU. Table 1 shows the STS configuration in the PPDU based on the number of packets.

STS Packet	**Position of STS in PPDU**	**Support**
0	No STS field included	Mandatory
1	Placed immediately after SFD field, before PHR field	Mandatory
2	Placed after PHY payload	Optional
3	Placed after SFD field, no PHR or Data field	Mandatory

Table 1: Configuration of STS packets in PPDU

The other IEEE standard that was used for the design of the range-finding tag is the IEEE 802.15.1. This standard focuses on Bluetooth Low Energy technology.

According to this standard, the physical layer is the first layer of the OSI model. The physical layer will oversee the transfer of bits between systems in the surrounding area through free space. The physical layer can receive a bitstream from the media access control sublayer. The layer can then transmit the data through radio waves to the associated station. The physical layer can also receive the radio waves from the station and convert them to a bitstream that will then be transferred to the media access control sublayer.

The Bluetooth transceiver operates in the 2.4 GHz range. This requirement for the Bluetooth transceiver is meant to provide compatibility between radios used in the system and define the quality of the system.

For our design, we will use a Bluetooth antenna. For an antenna, it is preferred that a temporary antenna connector is provided during type approval. The equipment is divided into three power classes. Power control is required by the power class 1. The power control will limit the power that is transmitted to over 0 dBm. The equipment classified as class 1 is able to control the transmitted power to 4dBm or less with a maximum transmitted power of +20dBm. The power class 1 will not be able to send packets to another device if the receiving side does not support the messaging for power control of the sending side. Another problem that class 1 faces is when the device pages another device in the nearby area and the input power is larger than the stated requirement.

Modulation characteristics is also mentioned in IEEE 802.15.1. The type of modulation that is used in Bluetooth technology is Gaussian frequency shift keying with a bandwidth time of 0.5. The modulation index for Gaussian frequency shift keying must be anywhere between 0.28 and 0.35.

To represent a positive frequency deviation, a binary 1 will be used. To represent a negative frequency deviation a binary 0 will be used. The symbol timing is less than +20 ppm and -20 ppm.

There are a couple of radio frequency specifications so that the transmitter can send data. The initial center of frequency accuracy of the transmitter must be + 75 kHz and -75 kHz maximum from the center frequency. This does not include the frequency drift requirement. Table 2 shown below lists the number of packets associated with the frequency drift.

Packet Type	Frequency Drift
One-slot packet	±25kHz
Three-slot packet	±40kHz
Five-slot packet	±40kHz
Maximum drift rate	400Hz/µs

Table 2: Packet size associated with frequency drift.

IV. Software Design Tools

There will be many different software programs that will be used to design the software structure of the range-finding tag. Some software programs will be used to program the hardware components in the range-finding tag and other software programs will be used to design the code for the mobile application. The following sections will go into detail about what each software program will be specifically used for in our system.

A. Microsoft VScode

Visual Studio Code is a source-code editing program designed by Microsoft. It was designed using the Electron framework which was created for designing desktop applications using web technologies. Visual Studio Code includes extensions for programming languages like C++, C#, and Python. We will be using Microsoft VScode to write code that will program the microcontroller. This software has many features to help developers write efficient programs. Some of these features are keyboard shortcuts, bracket matching, and syntax highlighting. The VScode debugger has the ability to debug code written in languages that get transpiled to JavaScript. For other languages, debugger extensions can be found in Visual Studio Marketplace.

B. Nordic nRF Connect

Our project will make use of the Nordic nRF Connect to link the Nordic nRF52833 development kit for Visual Studio Code. Using the VScode IDE, nRF Connect will allow us to develop and debug code for our application. Nordic nRF Connect offers features like RTOS-aware debugger, an interface to the nRF Connect SDK, a serial terminal, and an interface to the compiler and linker. It also offers a DeviceTree which helps identify the hardware components of a computer to the operating system's kernel so that the system components can be managed properly.

C. Nordic nRF Connect for Desktop

The Nordic nRF Connect for Desktop is a development software designed for Nordic products. It is designed to help developers test, modify, and optimize their applications. We will use this alongside the Nordic Profiler II to monitor the results of the device. Some of the key features that the Nordic nRF Connect for Desktop are Automatic Updates, Bluetooth LE, and power management. Power management is a feature we will use quite often to test the power of the device and make sure we are getting good readings.

D. SEGGER Embedded Studio

SEGGER Embedded Studio is a development system designed to provide stability and continuous workflow for microcontrollers and microprocessors. This programming software is free for non-commercial use and for commercial use with licensed partner devices. This software will help us deal with the stability issues of our device.

E. SEGGER JFlashLite

SEGGER JFlashLite is a programming application that programs data images to the flash of a microcontroller. This is the program we will use to flash the nRF52833 microcontroller embedded in our device. The two components that JFlashLite is composed of are the configuration dialog and the main window. The configuration dialog allows the user to configure the program or debug it. The main window is the user interface of the program. JFlashLite supports most external flash chips and has high-speed programming up to 550 Kbytes/s. It also offers Smart read back and verbose logging of all communication. Since this program offers a simple-to-use user interface, it will be a convenient program for us to utilize when flashing the microcontroller.

F. Xcode

Xcode is another integrated development environment we will use for project. Xcode will be used to create the program for our mobile application. The software is created by Apple and is mainly used to develop software for macOS, iOS, watchOS, and tvOS.

The software offers many features for developers to design efficient and powerful applications. Some of these features are source code checker, autocomplete feature, and templates to give a basic framework to expand on. Xcode supports a variety of programming languages such as C, C++, Objective-C, Java, Python, and many more.

G. GitHub

GitHub is used as a version control and collaboration tool for group when working on the software portion of the project. Everyone in the group has access to the code written for the microcontroller and the mobile application. This will make it simple for group members to work on the same code and make the necessary changes to the code at the same time. The group will work on the code for the microcontroller and the mobile application simultaneously and GitHub will be an important tool to review the code.

V. Mobile Application Design

The group decided to use a framework that is compatible with the iOS 16 Apple operating system. This operating system uses advanced programming that allows the UWB signal to be captured more accurately.

The nearby interaction framework is where we will be basing most of our mobile application on. This will allow our mobile application to identify the position of the Apple iPhone devices which contain the U1 chip. When the application is running with the framework, the smartphone will share its position along with a device token that will be needed to identify the device. The user will then trigger the action and it will display the direction and distance of the range-finding tag in units of meters. These functions will not be possible without the nearby interaction framework.

The SwiftUI framework is what will display the user controls and layout structure in the interface. This framework provides inputs from the user and also tools to manage the flow of data gathered from the mobile application. The SwiftUI framework was developed to help the developer create user interfaces easily and efficiently. Xcode being integrated into SwiftUI will help the group create the user interface at a rapid pace.

VI. Hardware Design

A. Block Diagram

For the realization of the Galahad project, there were many layers that had to be followed. These layers were important because they demonstrated the logical path that excited between the devices, technology, and electronic components that are needed.

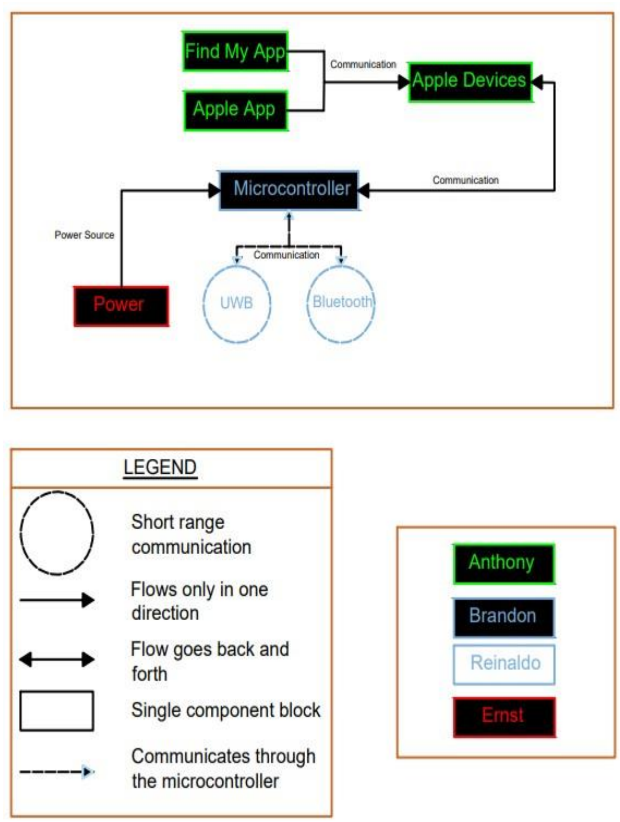

Figure 1: Test Setup – General Conditions

As it can be denoted in figure 1 below, there are four different types of colors that were taken into account, and each of these colors represents the area of focus of the engineers that are working on the project.

In order for the device to work, our developers have decided to elaborately go over the steps. The Find My App will be used as a third-party software app that will allow any Apple device, ranging from iPhone 11 to the most recent version or any Apple device that uses the A1 chip, so they could directly be connected to the Apple network and locate our device, which is the Galahad.

As for connecting to the Apple devices, the microcontroller will be in charge of finding the right output once a command has been passed. The primary technology that will be in charge of a steady connection will be Bluetooth, and once connected, the UWB could be activated when the user has made such command. The UWB, once connected, will be in charge of accurately measuring the distance between the Apple device and the distance tracking device.

Finally, we will be using a lithium coin cell battery of 3 volts DC to power the device, meaning the microcontroller and the different parts that will be involved in our design. The battery used to power the device is the CR2032, and the reason why we have decided to go with it is because of its performance on other devices on the market that are intended to work just like ours, for example, Apple tag or Samsung tag, and also because it meets our engineering requirements. Since our device is a low power consumption device, depending on usage and environment, it could last from as low as 2 months to a full year.

B. Hardware Overall Schematic

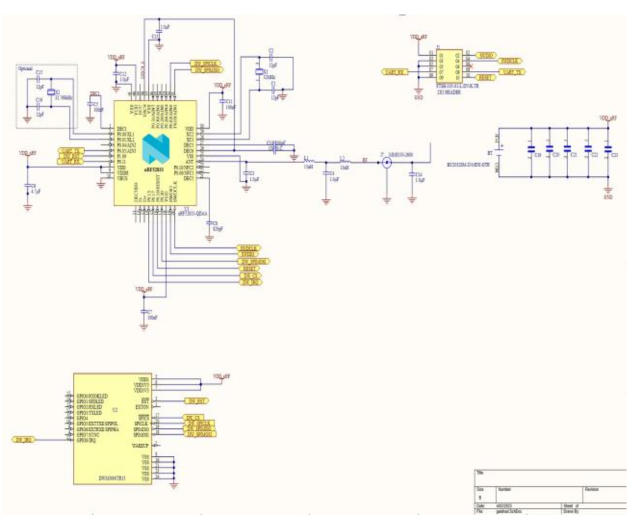

Figure 2: Overall Design Schematic

In the figure 2 above, we are demonstrating all the electronic components that are being used within our design, and all of them play a specific role in our device. In the schematic, we have decided to add a capacitor bank, so it could be used as a short-term energy reservoir, which implicitly means that it will be able to store electrical energy and then be released as quickly as possible when needed because we wanted to make sure that the battery does not get drained in a fast time period. There are some other capacitors used in the design, but their role is to serve as decoupling capacitors, meaning they will help reduce noise and voltage fluctuation throughout the entire circuit.

Some of the other components that we used in this design are inductors L1 and L2, which are used to filter out the unwanted noises in the circuit by allowing lower-frequency signals of the Bluetooth antenna to pass through.

Also, we used two crystals, 32 MHz and 32.768 kHz, to generate a precise frequency signal that will help with the synchronization process of the Bluetooth antenna for better reliable operations. This step is required because the specific connections, once established, need to be timed in a certain way to avoid any type of noise in the system.

At last, some of the other components represented in the circuit are the microcontroller, the UWB chip, the NFC module, the J-Link pins, and the battery holder, which were all mentioned in the previous lines above, more precisely, in the introductory chapter.

VII. System Testing

A. Testing Goals

As for how the system testing goes, there were many parameters that were established in order to meet our end goals, and they can be read as follows in table 3 below. One of our main goals for this project is to make it lightweight, and that is why we decided to constrain our design to 20 grams or less. Also, the required voltage throughout the entire circuit, as mentioned many times, will need to be 3V or at least close to it.

Furthermore, the precision finding must be from a 20 centimeters distance or less, and the device shall communicate with the end-user Apple device in a timeframe that no more than 30 or less. Also, the Bluetooth Low Energy and the Ultrawideband module should be able to connect with the device from a distance, respectively, more or equal to 8 meters and more or equal to 9 meters.

Engineering Requirement	**Justification**	**Unit**
Battery powered	Increase user operability and reduce device size	≤ 3 Volts

Lightweight	The device will need to be unnoticeable while carried by the user/object	≤ 20 Grams
Precision Finding	Users will be led to their missing item	≤ 20 Centimeters
The device Will Communicate with Apple iPhone	Implementation of Apple's robust framework	≤30 Seconds
Ultrawideband Range	Helpful for users using the precision-finding feature	≥ 9 Meters
Bluetooth Low Energy	Used for smartphone background tasks	≥8 Meters

Table 3: Testing Objectives

B. Testing Results

As mentioned in section A of this chapter, there were several areas of focus in what we wanted our device to perform well because our ultimate goal is that our device is as competitive as the similar distance devices in the market and because we want our users to have the best experience possible. That is why we have carried out all the necessary tests mentioned in table 3, so we ensure that our goal will be met.

In table 4 below here are the results that got from the tests that we have performed. We were able to test the three volts across the entire circuit, and the voltage often got, varied from 2.85 to 2.97 VDC, which met our voltage goal. Also, our precision finding was about a little bit more than 10 centimeters, which makes us happy because better accuracy enhances users' ability to precisely find their lost items.

As for the UWB and Bluetooth connections, we focused our research on two types of connections, outdoor and indoor connections, because these are the two different conditions that users will be facing. Regarding the Bluetooth connection, we were able to fast connect to the apple device, on average, every 2.89 seconds, and in the best outdoor conditions, meaning optimal weather conditions, clear view, and no barriers, we were able to connect at a measured distance of 27 meters. Furthermore, when tested in indoor conditions, where all the doors are opened, we were able to get a connection at a distance that is as far as 12.3 meters, but once all of these doors were closed, we were able to get a connection in a distance that was at most 9.8 meters, which is 0.8 higher than our engineering distance requirement related to the Bluetooth antenna.

Item	**Conditions**	**Value**	**Units**
1	Voltage testing (coin battery)	2.85 – 2.97	VDC
2	Voltage for programming calibration data into on-chip non-volatile memory (DC source)	2.75 – 3.6	VDC
3	Precision finding	10	cm
4	Bluetooth connect	2.89	sec
5	UWB connect	1.84	sec
6	UWB disconnect	.52	sec
7	Bluetooth connect (outdoor best condition)	27	meters
8	Bluetooth connect indoors (opened doors)	12.3	meters
9	Bluetooth connect indoors (closed doors)	9.8	meters
10	UWB connect (outdoor)	25	meters
11	UWB connect (indoor)	9.3	meters

Table 4: Test Setup – General Conditions Tested

Moreover, we tested the integrity of the Ultra-wideband connection, just like the Bluetooth connection, in indoor and outdoor situations. We also measured how much time it took on average to connect and disconnect to the device.

Once Bluetooth connectivity is established, the UWB module is now ready to get connected to the device when asked. The average time that it took the UWB to connect was 1.84 seconds and took only about half a second to disconnect. When the UWB signal was tested outdoors, the furthest distance that we could get a connection from was 25 meters. On the other hand, when we tested it indoors, we were able to connect at a distance, at most, 9.3 meters, which entirely meets all of our engineering testing criteria.

C. Bluetooth and UWB signals

In figures 2 and 3 below, we are showing the broadcast signal of both, the Bluetooth and the Ultra-wideband current outburst once they are connected to the device. In the first figure, we could see a current outburst happening every 3.584 seconds, and they occur from a range that is as low as 2.15mA and that goes as high as 21.55mA.

Furthermore, in the next figure, figure 4, when the UWB gets to be connected, these outbursts happen within the same consistent timeframe of 3.584 seconds seen earlier, however, more current is being drawn. The current range goes, on average, from 7.52mA to 99.65mA.

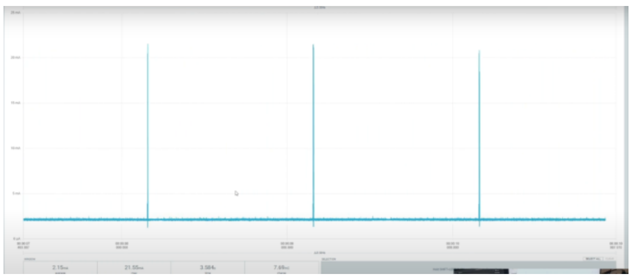

Figure 3: Bluetooth Signal Upon Connection

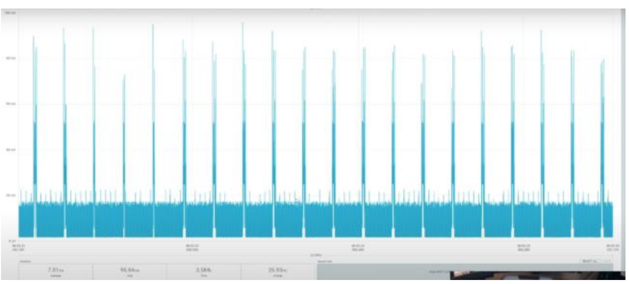

Figure 4: UWB Signal Upon Connection

VIII. PCB Design

In order to meet our weight constraint, which was less or equal to 20 mg, we have decided to go with a PCB design that is as compact as we possibly can. The board has a total of four layers, and these layers are a top layer, two middle layers, and a bottom layer. The top layer, as shown in figure 5, has all of our components and the power traces, and the bottom layer, from figure 6, however, is only used to route certain traces by the use of Vias. It has to be noted that both of these layers have also a ground plane.

As for the middle layers, they were used for dialectic and signal integrity since we have a lot of high-frequency and low-frequency components within our PCB board. Hence, we wanted to ensure that we strengthen our signals by limiting the noise that would occur if we were to with only a two-layer design, which would have most likely caused inadequate and inaccurate readings from our UWB module and Bluetooth antenna.

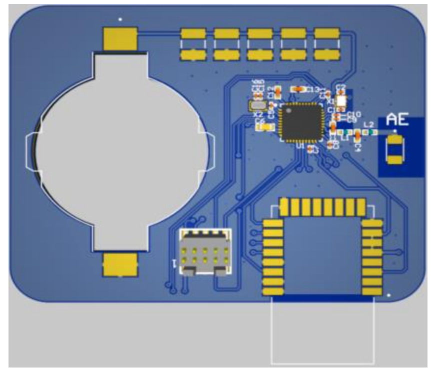

Figure 5: Top Layer 3D render

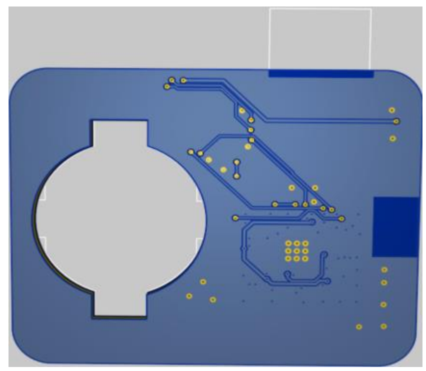

Figure 6: Bottom Layer 3D render

IX. Conclusion

In conclusion, as young engineers, this project is one of the most challenging that we have encountered so far because it was a very time-consuming project and required a ton of effort, dedication, and problem-solving research. Nevertheless, this must be one of the most exciting projects that we have also done because of all the excitement we got from it once we were able to build our PCB, test it, and make sure it worked.

There were a lot of gains doing this project as well since many people were involved in the realization of the device. It sure helped us have a win-win mentality, prepared us for what our work environment would be post-graduation, learn how to build trust among the group, and most importantly, accountability and responsibility.

We are also excited to know that we have a created device that will help many people to accurately find what is dear or indispensable to them, which has always been our goal from the best because we wanted our users to have the best experience possible while using our product, and also that will help them be carefree if they were to lose any object trackable by Galahad because of its precision and accuracy.

THE ENGINEERS

Reinaldo Romero is a 24-year-old Electrical Engineering student. Reinaldo hopes to one day work for a large corporation as a systems engineer who specializes in aircraft design.

Brandon Powell is a 26-year-old Electrical Engineering student, who hopes to one day work on hardware design.

Anthony Venkersammy is a 27-year-old Computer Engineering student, who is interested in embedded systems and will be working with a flight simulation and training company after graduating.

Ernst Dormevil is a 28-year-old graduating Electrical Engineering student, who is presently working as an intern at **Florida Municipal Power Agency** and wishes to pursue his career in the power industry as a generation and distribution engineer.

ACKNOWLEDGMENT

The authors would like to express their gratitude and acknowledgment to all the faculty members who helped us to sharpen our knowledge throughout our academic years at UCF. We would also like to address special thanks to Dr. Lei Wei for his genuine feedback regarding our Senior Design 1 project proposal paper and also for assisting us in the realization of our project in Senior Design 2.

We would also like to express our gratitude and special thanks to **Best Global Source**, more precisely to Janak Bhakta, for his contribution to helping us assemble some of the components from our PCB at no cost.

REFERENCES

[1] "OpenHaystack," *GitHub*, Jun. 24, 2022. https://github.com/seemoo-lab/openhaystack

[2]"Zigbee - Nordic Semiconductor," *www.nordicsemi.com*. https://www.nordicsemi.com/Products/Bluetooth-Direction-Finding

[3] "Find My network," *www.nordicsemi.com*. https://www.nordicsemi.com/Products/Find-My-network (accessed, 2022).

[4] "DWM3001CDK - Qorvo," *www.qorvo.com*. https://www.qorvo.com/products/p/DWM3001CDK#evaluation-tools (accessed, 2022).

[5] "Type2BP UWB Development Kit," *www.nxp.com*. https://www.nxp.com/products/wireless/secure-ultra-wideband-uwb/murata-type2bp-evk-trimension-sr150-uwb-development-kit:UWB-DEV-TYPE2BP-EVK (accessed, 2022).

[6] "Bluetooth Low Energy," *www.nordicsemi.com*. https://www.nordicsemi.com/Products/Bluetooth-Low-Energy (accessed, 2022).

[7] "Board support — nRF Connect SDK 2.1.99 documentation," *developer.nordicsemi.com*. https://developer.nordicsemi.com/nRF_Connect_SDK/doc/latest/nrf/app_boards.html (accessed, 2022).

[8] "GitHub - br101/zephyr-dw3000-decadriver at bin," *GitHub*. https://github.com/br101/zephyr-dw3000-decadriver/tree/bin (accessed, 2022).

[9] "Introduction to Nordic nRF5 SDK and Softdevice - Getting Started - nRF5 SDK guides - Nordic DevZone," *devzone.nordicsemi.com*. https://devzone.nordicsemi.com/guides/short-range-guides/b/getting-started/posts/introduction-to-nordic-nrf5-sdk-and-softdevice (accessed, 2022).

[10] "About Swift — The Swift Programming Language (Swift 5.2)," *docs.swift.org*. https://docs.swift.org/swift-book/

[11] "Apple Developer Documentation," *developer.apple.com*. https://developer.apple.com/documentation/nearbyinteraction (accessed, 2022).

[12] "The Ultimate PCB Design Software Comparison | SFCircuits," *www.sfcircuits.com*.

https://www.sfcircuits.com/pcb-school/pcb-design-software-comparison-guide (accessed, 2022).

[13] “What is Ultra-Wideband (UWB)? Here’s everything you need to know | Bleesk,” *bleesk.com*. https://bleesk.com/uwb.html

[14] “UWB Solutions Compatible with Apple’s U1 Chip - Qorvo,” *www.qorvo.com*. https://www.qorvo.com/innovation/ultra-wideband/products/uwb-solutions-compatible-with-apple-u1 (accessed, 2022).