Galahad

real-time location ranging tag that makes use of UWB and BLE technology to find items

UNIVERSITY OF
CENTRAL FLORIDA

Senior Design I
Group 23

Anthony Venkersammy, **CPE**

Brandon Powell, **EE**

Ernst Dormevil, **EE**

Reinaldo Romero, **EE**

Table of Contents

1. Executive Summary **1**

2. Project Details **2**

2.1 Project Motivation *2*

2.2 Goals & Objectives *2*

2.3 Market Analysis *3*

2.4 Engineering Requirements *6*

2.5 Project Features *8*

2.6 House of Quality *9*

2.7 Hardware Block Diagram *10*

2.8 Microcontroller Software Diagram *12*

2.9 Mobile Application Block Diagram *13*

3. Technology Investigation & Part Selection **14**

3.1 Ultra-Wideband (UWB) *14*
3.1.1 UWB communication 14
3.1.2 Time Difference of Arrival (TDoA) 15
3.1.3 Two-Way Ranging (TWR) 15
3.1.4 Phase Difference of Arrival (PDoA) 16

3.2 Ultra-Wideband IC Considerations *16*
3.2.1 Qorvo DW3110 17
3.2.2 NXP Trimension SR040 18
3.2.3 NXP Trimension SR150 18

3.3 Ultra-Wideband IC Selection *19*

3.4 Bluetooth Low Energy (BLE) *20*
3.4.1 Bluetooth Low Energy 20
3.4.2 Location 20
3.4.2.1 BLE Positioning with Sensors 20
3.4.2.2 BLE Positioning with Beacons 21

3.3 Near Field Communication (NFC) *22*

3.4 Microcontroller *24*
3.4.1 Types of Microcontrollers 24

3.5 Microcontroller Considerations *26*
3.5.1 Nordic Semiconductor nRF52840 26
3.5.2 Nordic Semiconductor nRF52833 27
3.5.3 NXP QN9090 28

3.6 Microcontroller Selection *29*

3.7 DC/DC Converter *31*
3.7.1 Boost Converter 31
3.7.2 Buck Converter 31
3.7.3 Buck-Boost Converter 31
3.7.4 Cuk converter 32

3.8 UWB DC-DC Converter Considerations ... *32*
3.8.1 Torex Semiconductor XC9282 ... 33
3.8.2 Texas Instruments TPS62746 ... 34
3.8.3 Texas Instruments TPS62840 ... 35
3.9 Ultra-wideband DC/DC IC Selection ... *36*
3.10 Button Cell Battery ... *37*
3.11 Button Cell Battery Considerations ... *40*
3.11.1 CR2016 Battery ... 40
3.11.2 CR2025 Battery ... 41
3.11.3 CR2032 Battery ... 42
3.11.4 CR2032H Battery ... 43
3.12 Battery Selection ... *44*
3.13 3D Printing ... *46*
3.14 3D Printer Filament Considerations ... *47*
3.14.1 ABS ... 48
3.14.2 PETG ... 49
3.14.3 Nylon ... 49
3.14.4 Carbon Fiber ... 50
13.14.5 PLA ... 51
13.14.6 ASA ... 51
3.14.7 Polypropylene ... 52
3.14.8 Wood Filled ... 52
3.14.9 Metal Filled ... 53
3.14.10 PVA ... 53
3.14.11 HIPS ... 54
3.14.12 Flexible ... 54
3.14.13 Polycarbonate ... 55
3.15 3D Printer Filament Selection ... *56*
3.16 Coin Cell Battery Holder Selection ... *57*
3.16.1 Keystone Electronics 1057TR ... 57
3.16.2 Memory Protection Devices BU2032SM-JJ-MINI-GTR ... 58
4. Standards, Certifications, & Constraints ... **59**
4.1 IEEE 802.15.4 Ultra-Wideband ... *59*
4.2 IEEE 802.15.1 Bluetooth ... *60*
4.3 FiRa Certification ... *62*
4.4 MFi Certification ... *62*
4.5 Software Constraints ... *63*
4.6 Smartphone Constraints ... *63*
4.7 Size Constraints ... *64*
4.8 Environmental Constraints ... *64*
4.9 Economic Constraints ... *65*
4.10 Ethical Constraints ... *66*
5. Hardware & Software Design ... **67**
5.1 Hardware Development Tools ... *67*

5.1.1 - Qorvo DWM3001CDK ... 67
5.1.2 - Nordic nRF52833 DK ... 69
5.1.3 - Qorvo DWM3000EVB ... 71
5.1.4 - Nordic Power Profiler II ... 73
5.1.5 - Segger J-Link ... 76
5.2 Initial Development Board Testing ... 77
5.3 PCB Design Software Considerations ... 80
5.3.1 Autodesk EAGLE ... 80
5.3.2 KiCad ... 81
5.4 PCB Design Software Selection ... 81
5.5 Hardware Design ... 82
5.5.1 MCU ... 82
5.5.2 UWB ... 83
5.5.3 DWM3001C ... 84
5.5.4 DW3110TR13 ... 84
5.5.5 Buck Boost Converter ... 85
5.5.6 Decoupling Capacitors ... 85
5.5.7 Battery Holder ... 86
5.5.8 Overall Schematic ... 87
5.5.9 BOM ... 88
5.6 Software Development Tools ... 88
5.6.1 Microsoft VScode ... 88
5.6.2 Nordic nRF Connect for VScode ... 90
5.6.3 Nordic nRF Connect for Desktop ... 91
5.6.4 SEGGER Embedded Studio ... 92
5.6.4.1 SEGGER Embedded Studio Price ... 93
5.6.5 SEGGER JFlashLite ... 94
5.6.6 Xcode ... 95
5.6.7 Discord ... 96
5.6.8 GitHub ... 98
5.7 Programming Languages ... 99
5.7.1 C ... 99
5.7.2 C++ ... 100
5.7.3 Swift ... 101
5.8 Microcontroller Software Design ... 103
5.8.1 nRF Connect SDK ... 103
5.8.2 Zephyr RTOS ... 104
5.8.3 DWM3000 API ... 106
5.8.4 OpenHayStack ... 106
5.9 Mobile Application Design ... 107
5.9.1 Chosen Operation System ... 107
5.9.2 Nearby Interaction Framework ... 107
5.9.3 ARkit Framework ... 108
5.9.4 SwiftUI Framework ... 109
6. System Testing ... 110
6.1 Prototype Testing ... 110
6.2 Testing Environment ... 110
6.3 Indoor Testing ... 111
6.4 Outdoor Testing ... 112

6.5 Testing Antenna Delay *113*
6.6 Operational Testing *114*
6.7 Mobile App Testing *116*
6.8 Software Specific Testing *116*

7. Administration **118**
7.1 Budget *118*
7.2 Financing *118*
7.3 Milestone Timeline *119*

8. Conclusion **120**

Appendix ..... **i**
Permissions ..... *i*
References ..... *v*

Tables & Figures

Figure 2-1: Stock Image of Apple AirTag **3**

Figure 2-2: Stock Image of Samsung SmartTag **4**

Figure 2-3: Stock Image of Chipolo ONE Spot **5**

Table 2-1: Summary of Top Three Finder Devices **6**

Table 2-2: Engineering Requirements **7**

Figure 2-4: House of Quality **9**

Figure 2-5: Hardware Block Diagram **11**

Table 2-3: States of UWB Tag Microcontroller **12**

Figure 2-6: Microcontroller Software Block Diagram **12**

Figure 2-7: Mobile Software Block Diagram **13**

Table 3-1: DW3110 Parameters **19**

Table 3-2: Summary of Bluetooth Generations **21**

Table 3-3: Summary of NFC Types **23**

Table 3-4: Summary of the 5 Different Microcontrollers **25**

Table 3-5: nRF52840 Key Features **27**

Table 3-6: nRF52833 Key Features **28**

Table 3-7: QN9090 Key Features **29**

Table 3-8: Microcontroller Selection Reasoning **30**

Table 3-9: Summary of The Different DC/DC Converters **32**

Figure 3-1: DC/DC Converter for UWB Block Diagram **33**

Table 3-10: XC9282 Key Features **34**

Table 3-11: TPS62746 Key Features **35**

Table 3-12: TPS62840 Key Features **35**

Table 3-13: Summary of DC/DC Buck Converters **36**

Table 3-14: Summary of Button/Coin Cell Battery Types **39**

Figure 3-2: Stock Photo of CR2016 Lithium battery **40**

Table 3-15: CR2016 Lithium Battery Specifications **41**

Figure 3-3: Stock Photo of CR2025 Lithium Battery **41**

Table 3-16: CR2025 Lithium Battery Specifications **42**

Figure 3-4: Stock Photo of CR2032 Lithium Battery **42**

Table 3-17: CR2032 Lithium Battery Specifications **43**

Figure 3-5: Stock Photo of CR2032H Battery **43**

Table 3-18: CR2032H Lithium Battery Specifications **44**

Table 3-19: Summary of Battery Selection **45**

Table 3-20: Cost of ABS For Different Spool Size **49**

Table 3-21: Price Per Ounce for Nylon Filament **50**

Table 3-22: Price Per Ounce for Carbon Fiber Filament **50**

Table 3-23: Price Per Ounce for ASA Filament **52**

Table 3-24: Price Per Ounce for Metal Filled Filament **53**

Table 3-25: Price Per Ounce for Flexible Filament **55**

Table 3-26: Price Per Ounce for Polycarbonate Filament **56**

Figure 3-6: Through Hole PCB Mount View **57**

Figure 3-7: Stock Photo of Keystone 1057 ..... **58**

Figure 3-8: Stock Photo of MPD BU2032SM-JJ-MINI-GTR 58

Table 3-27: Summary of Coin Cell Battery Holders 58

Table 4-1: Amount of Frequency Drift in A Packet 62

Table 4-2: Fee Associated with Device Type 62

Figure 5-1: Schematic of DWM3001CDK 68

Figure 5-2: Schematic of DWM3001 MCU 68

Figure 5-3: Schematic of Micro USB 69

Figure 5-4: Schematic of nRF52833 70

Figure 5-5: Buck Regulator for VDD 71

Figure 5-6: 5V Boost Regulator 71

Figure 5-7: Qorvo Arduino Compatible Shield 72

Table 5-1: Key Features of Qorvo DWM3000EVB 73

Figure 5-8: Stock Photo of Nordic Profiler II 73

Table 5-2: DC Power Profiling Devices 74

Table 5-3: Key Features of Nordic Power Profiler II 75

Table 5-4: J-Link Version and Comparisons 76

Figure 5-9: SEGGER J-Flash Lite to Flash Binary File 77

Figure 5-10: Personal Photo of DWM3001CDK Successful Flash 78

Figure 5-11: Virtual COM Port Selection 78

Figure 5-12: Ranging Statistics 79

Table 5-5: Guide for Test Data 79

Table 5-6: Key Features of EAGLE Software 80

Table 5-7: Key Features of KiCad Software 81

Figure 5-13: Schematic of nRF52833 Microcontroller 82

Figure 5-14: Schematic of DWM3000TR13 (Integrated UWB module) 83

Figure 5-15: Schematic of DWM3001C (Fully Integrated UWB Chip) 84

Figure 5-16: Schematic of DWM3110TR13 (Base UWB Chip) 84

Figure 5-17: Schematic of Buck Boost Controller 85

Figure 5-18: Schematic of Decoupling Capacitors 85

Figure 5-19: Schematic of BU2032SM Battery Holder 86

Figure 5-20: Schematic of Overall Project 87

Table 5-8: Bill of Material of All Major Components 88

Table 5-9: Key Features of Nordic nRF Connect for Desktop 92

Table 5-10: SEGGER Embedded Studio Key Features 93

Table 5-11: SEGGER Embedded Studio Fees 94

Figure 5-21: Discord Logo 96

Table 5-12: Discord Key Features 98

Figure 5-22: Group 23's GitHub Page 98

Table 5-13: Differences Between C And C++ 100

Figure 5-23: nRF SDK Usage Among MCU and App Interaction 103

Figure 5-24: Connect SDK Components 105

Figure 5-25: NI Framework Visual 108

Figure 5-26: Schematic of DWM3001CDK 109

Table 6-1: Test Setup – General Conditions 110

Figure 6-1: Qorvo Chip Antenna Delay Process 113

Table 6-2: Production Test Prior Checking from Qorvo 115

Table 7-1: Project Budget List 118

Table 7-2: Milestone Timeline 119

1. Executive Summary

This project's team consists of four team members, Anthony Venkersammy, Brandon Powell, Ernst Dormevil, and Reinaldo Romero. Anthony is a computer engineering major in the comprehensive track that is interested in embedded systems, software, and hardware development. Brandon is an electrical engineering major on the comprehensive track with interests in embedded systems and PCB design. Ernst is an electrical engineer in the power and renewable energy track and is interested in power distribution. He is currently interning for a company with a similar background. Reinaldo is an electrical engineering major in the comprehensive track and is interested in RF technology and PCB design.

Coming from similar backgrounds and interests as a group we wanted to find a suitable project that would be engaging for all of us. Not only appealing but something we can take with us to potential employers in related career tracks after we graduate with our degrees. After many project ideas and brainstorming sessions, we decided on a project that could be used by consumers. Specifically, a consumer electronic that would help people find any item they attach this device to, in their given surroundings. More commonly this little device that is used to locate your misplaced, forgotten, or stolen items is called a finder or tag. This tag is meant to be placed on keys, bags, wallets, equipment, and even pets.

Throughout the years these finders have gone from a simple receiver with a transmitter to full communication systems using Bluetooth low energy. While the market has several excellent examples of finder products that consumers can buy, we think our group can expand on this area by utilizing the newest technologies, having a usable design, and features not yet found in other tags.

Our team plans to execute this project by relying on ultra-wideband technology and smartphones. Consumer UWB (ultra-wideband) modules have capabilities of precision location allowing for tags make with these modules to be tracked with an accuracy of 10 centimeters. On the other end of this system will be the smartphone to guide the user to the location of the module which is what we will be integrating into our tag. With our project making use of the consumer's smartphone, we hope to leverage the functions smartphones give us in our daily lives with the app.

To make the project as realistic as possible we want to focus on the main ideas of what makes a tag work. These ideas can be more thoroughly explained in the project details. Without these concrete functions, we will not be able to make a finder that has similar utilities. This includes the ability to use these technologies that other devices are not currently. In a nutshell, our project aims to create an ultra-wideband tag at first. Not only that but an ultra-wideband tag that will generalize many different concepts of how people will use want to use it and create new notions of how people will use finders to come.

2. Project Details

This section will discuss the basics of what our project will stem from. What and how do we intend to reach the goals through objectives and engineering requirements. We also plan to give insight into some of the project's development through hardware and software block diagrams.

2.1 Project Motivation

People have a lot of stuff, many of most could be misplaced, left behind, or stolen. There are some items you can't afford to lose. What if you could make any personal object trackable? It would cut out the need for time-consuming searches and not quite being able to retrace your steps. Galahad helps people find their possessions.

2.2 Goals & Objectives

The goal of this senior design project is to create a device that will be able to find any object the user attaches to. We want this finder to be simple in design and have a small footprint for the most user applicability. This small form factor is possible by eliminating a speaker in the device to audibly help users find their item and solely relying on the real-time location system. Another way to cut down on device dimensions is utilizing ultrawideband devices with the integration of a radio frequency antenna on the module and SoCs supporting built-in Bluetooth Low Energy. We want to use the smallest components as much as possible in our design to achieve an overall small dimension.

This device would be capable of being located on a smartphone application map. It would also be capable of providing real-time distance. Acting as an anchor for the finder, these finder features would only work with a smartphone application as a receiver. Real-time direction and distance will be possible with the Nearby Interaction Framework on Apple iPhones equipped with the U1 chip. Utilizing the user's smartphone app, we plan to create a platform that can expand upon the equipment of our ultra-wideband tag.

We will try incorporating this device into the Apple Find My network to achieve mobile map location. Although this will be a far-off goal for our group due to the stipulations of the Find My network. Apple only allows manufacturers to access the necessary software and tools needed for our device to be capable of global location. Simply, the hardware we will use has the technology to be on the Find My network, but we will not be able to access any software to enable this. We will however attempt to access this network's features through the means of open-source software.

That being said, the main purpose of this device will be to lead you to your tagged item. We want to focus on the main mechanism of what an ultra-wideband tag consists of. A tag ultimately needs to accurately plus constantly give the distance

between the object and the user. The tag will need to have an interface, in this case, connect to an interface that will be easy for the user to follow. Lastly, this UWB tag will need to update in response to the user's location i.e., connected device. Our main goal will be to achieve these concepts first, secondary we want to build new features that will use the main notions as the foundation.

2.3 Market Analysis

There are similar tracking products on the market, Apple has its AirTag, and Samsung has its SmartTag. Both devices are similar in functionality, using Bluetooth and UWB technology to locate the device. We decided it was best to mimic these with our device as UWB has very precise location pinning. We decided to use Samsung, Apple, and a third-party device to help identify product features. Below are some of the features of each:

Figure 2-1: Stock Image of Apple AirTag

- Nordic nRF52832
- Single frame with 3 antennas
- AirTag measures 1.26 inches in diameter, and it has a height of 0.31 inches
- 11 grams in weight
- Managed in Find My app
- AirTag uses a replaceable CR2032 battery
- IP67 water and dust resistance rating
- Only compatible with iPhones
- Speaker for locating
- U1 UWB IC

The Apple AirTag is a small circular-shaped tracking device that allows the user to pinpoint the location of non-Apple devices. It uses ultra-wideband technology to get precision location when paired with an iPhone 11 or newer. Apple AirTag works with the Find My app and is shown there was the initial pairing is successful. It also includes a speaker for when it is lost in a tight space such as a couch where sound will benefit the user in locating the device. The Apple AirTag is conveniently small and uniquely shaped. This design choice forces the consumer to invest in their 3rd party accessories to use it more generally. We will consider removing this in our

design. The AirTag has a metal surface that would tend to scratch up considering it is on a keychain or something similar.

Figure 2-2: Stock Image of Samsung SmartTag

- 1.8 inches in length and width with 0.4 inches thick
- 13 grams in weight
- Speaker for locating
- Square in shape
- IP53 rating for water resistance
- Replaceable CR2032 battery
- Only compatible with Samsung devices
- NXP SR040 UWB IC

The Samsung SmartTag is a square shape tracking device that uses UWB using the NXP SR040 chip. The Samsung SmartTag uses other Samsung devices in the area to bounce off a signal if the user is at a distance too big for a connection. The device connects remotely via Bluetooth to a Samsung device and is managed on the SmartThings app. Samsung SmartTag has a more user-friendly design made from hard plastic with an included keyhole. It also has a way for the user to replace the battery if needed. We will be closely following this device as far as squareness and durability go.

Another feature of the Samsung smart tag is its customizable button. By using the SmartThings app, a user can communicate with IoT devices. For example, you can turn on a smart light with the button just as long as the smart light can connect to the SmartThings app. The button has two different customizable options, one short press or one long press. To find the tag, you can use the search nearby feature on the SmartThings app. This uses a Bluetooth signal that will give the location distance based on the strength of the signal. There is also a speaker capable of 100dB that comes customizable with 10 different ringtones, so if the tag is nearby activating the speaker will be the better option.

Figure 2-3: Stock Image of Chipolo ONE Spot

- Speaker for locating
- Only uses Bluetooth technology
- Round
- Works with Apple's Find My app
- 1.49 inches in diameter and 0.25 inches thick
- 8 grams in weight
- IPx5 rating splashproof
- Replaceable CR2032 battery

Chipolo ONE Spot was the first Chipolo device that could use the Apple Find My app. When compared to the Apple AirTag is a little bit bigger, but it comes with a loophole, so no accessory is needed. The Chipolo ONE Spot also has a speaker rated at 120dB which is quite loud. It uses Bluetooth technology instead of UWB so there might be some downside in battery life, but the battery is easily replaceable. The Chipolo ONE Spot was chosen to help identify product features because it is a third-party device that can be used in Apple's Find My app. What we will do differently is to integrate ultra-wideband technology into our device. The key features of the top three finder devices on the market are summarized in table 2-1.

	Apple AirTag	Samsung SmartTag	Chipolo ONE Spot
Length/Diameter	1.26 inch	1.8 inch	1.49 inch
Width	N/A	1.8 inch	N/A
Height	0.31 inch	0.4 inch	0.25 inch
Weight	11 grams	13 grams	8 grams
Price	$29	$29.99	$28
Technology	U1 ultra-wideband chip	NXP SR040 UWB	Bluetooth
Operating System	iOS	Android	iOS/android

Table 2-1: Summary of Top Three Finder Devices

- Device Similar to Our Project

2.4 Engineering Requirements

To meet our goals and objectives on this project we have carefully picked our engineering requirements. These can be found in table 2-2 with their justifications and how we plan to quantify these terms. The requirements were chosen to develop a device that will have our specified functionality for this project but also a platform that will enable us to build upon it.

The engineering requirements featuring technologies such as ultra-wideband, Bluetooth low energy, and Apple iPhone operability will enable us to bring key finder features that will be the most useful to users. Ultra-wideband will enable us to make a tag that can precisely find a stationary object down to centimeters. Bluetooth low energy will allow for extended tracker range and smartphone background processes. While iPhone combability will help us build an app to increase usability and features.

Of course, device features are important to this project we also want a package that will make sense from a consumer standpoint. Therefore, our engineering requirements focus on the size and power of this finder. We want our tag to be

comparable to other products on the market. Most tags on the market have a small footprint and low battery often lasting months to a year on a single coin cell battery.

In summary, our project aims to create an ultra-wideband tag. We want to generalize many different concepts of how people will use a tag. We can't achieve the ability to use these technologies if we don't establish solid engineering requirements before getting further into this project. To complete our goals and finish in a realistic time frame we broke down the engineering specifications that are the basic requirements needed for a tag device to be useful, practical, and intuitive. While these requirements are simple, they are essential not only for the reasons previously mentioned but for a tag device we can continue to expand the uses of.

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 2-2: Engineering Requirements

- Requirements That Will Be Demonstrated in SD2

2.5 Project Features

The function of this project is to give the user a simple tracking device that will work at a two-dimensional distance. On set up the device will first be paired using ultra-wideband technology, that comes with most modern smartphones. We decided to use ultra-wideband instead of Bluetooth low energy mainly because it is more precise in pinpointing a location. Ultra-wideband also can connect to more devices at a time when compared to Bluetooth. This will benefit us because we plan on using a protocol that allows our UWB device to connect to nearby devices this will allow greater accuracy when locating a device that is out of range from the user. Ultra-wideband is also a newer technology that is beginning to take over where Bluetooth was already implemented, so we wanted to have the newest technology in our device.

Another important thing we considered was the actual OS (operating system) it would be configured to. The two popular OS we have today, are Google with Android and Apple with its iOS. We decided to go with Apple because we feel Apple has the nicest ecosystem that will benefit us in the long run. With Apple, there is a Find my app that organizes all the user's device locations. Our device will attempt to work as a third-party device using the Find My app as a far-off goal.

Finder devices on the market are typically small. We decided this is an important feature because a user should be able to just attach it to a keychain or something similar, and not notice a huge change in weight. We will achieve this small weight feature by having minimum circuity to create a simple and small PCB design. To shell our PCB design, we will 3D print a minimalist shape design using lightweight plastic. Our 3D-printed design will include a built-in loophole that will do away with the need for third-party accessories. We need to consider having the battery replaceable, so our design will be two square shelling’s that clip together.

One of the most important features of any tracking device is the longevity of its battery life. Most devices on the market use a CR2032 battery. This battery is circular hence why most of the designs are based on this shape. With the combination of UWB and this long-lasting battery, we will strive to have one of the most long-lasting devices compared to the ones on the market. Most tracking devices come with an integrated speaker that will play a loud noise when the user activates it, usually by pushing a button on the app. We see the speaker as unnecessary and won’t be implementing it in our design. The speaker will also drain the battery much faster than without. We also want the battery to be replaceable because we understand the battery life will only diminish over time.

2.6 House of Quality

Figure 2-4 shows features and qualities consumers would like to have in a tag device such as ours and competitors. Our engineering requirements are shown in the House of Quality to clearly illustrate our areas of maximizing and minimizing. Relationships between the most valuable needs of consumers and the product were our focus when defining this HOQ (house of quality).

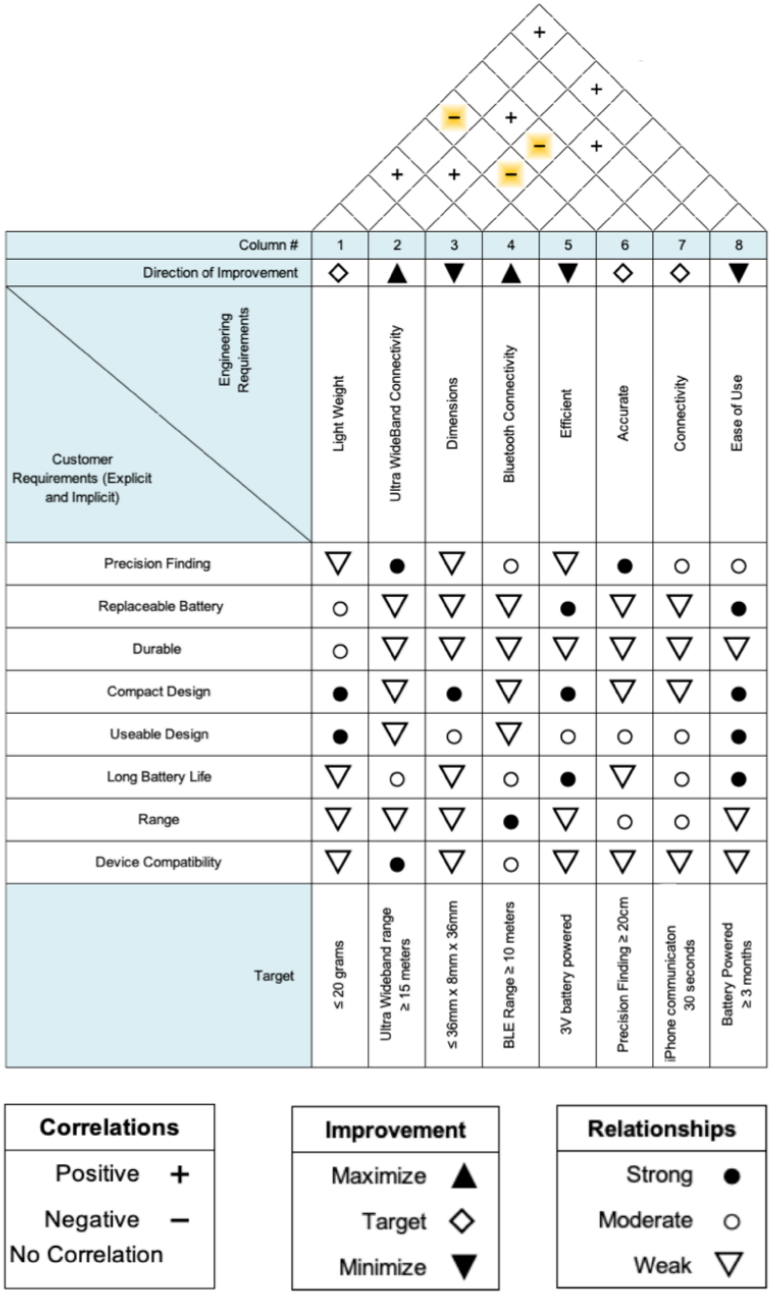


Figure 2-4: House of Quality

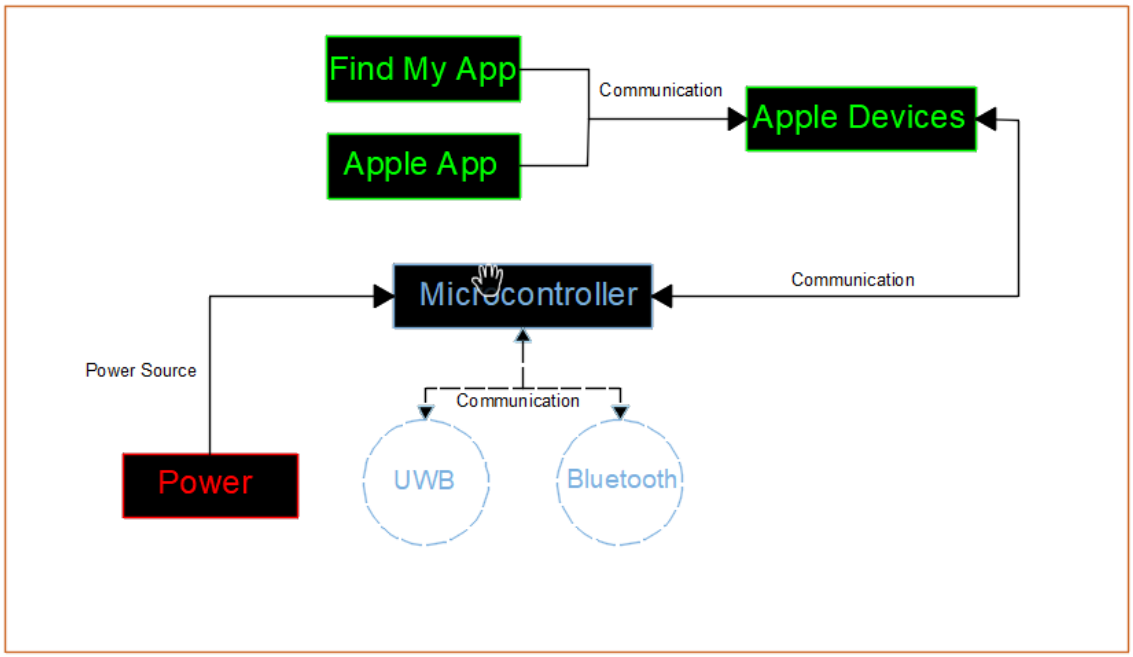
2.7 Hardware Block Diagram

Figure 2-5 located below, which is the block diagram for the project, is all of the layers and specifications that the group will have to follow to accomplish the project. Moreover, in the figure, it could be seen that there are several distinct colors representing each part that each of our group members will be focused on and their names and their names are associated based on the color that each of the members has chosen to work on. However, even though the members specified the field that they would like to focus on the most, their area of work does not only limit to the specified field chosen since some of the areas require more work and research to be done.

To make this device work, our group, which is composed of three EE major students and one CPE, has decided to go over a series of important steps. First, as seen in the figure, the Find My App is intended to work as a third-party software app that will communicate to Apple directly and that will also allow our distance tracking device to get connected to any Apple devices that use a U1 chip because of the microcontroller constraints that can only support that type of technology, and also because we encountered software constraints related to Apple because we were not able to get access to their app since we are not a manufacturing company. Moreover, as was mentioned in section 6.3 the list of Apple devices that we can use for our project are the series of Apple phones that were built from 2019 up to 2021, which are the iPhone 11, 12, and 13 because they use ultra-wideband technology. Even though the iPhone 14 is also capable of using ultra-wideband signals, our microcontroller will not be able to communicate with it because it does not support iOS 16.

Furthermore, the microcontroller can connect to Apple via Bluetooth, which is the primary way of connecting to the Find My network, which will have the capabilities of that app. Using an ultra-wideband integrated circuit as a signal tracker is far better than what Bluetooth can do because the device could be able to catch up to its signal up to 100 meters, but more accurately at 10 meters. Both Bluetooth and ultra-wideband will communicate to the device via the microcontroller. Also, the communication between the microcontroller and the Apple device will be two-way, which implies that the Apple Device could directly communicate with the microcontroller, or it could be the other way around.

Finally, there will be the power source which will be a 3V DC powered by a coin battery, CR2032 lithium battery, that will power the microcontroller and the different components that will be part of our design. Since our device is a low energy consumption device, we are expecting it to last if it lasts in tracking devices that use similar technology and battery as our device.



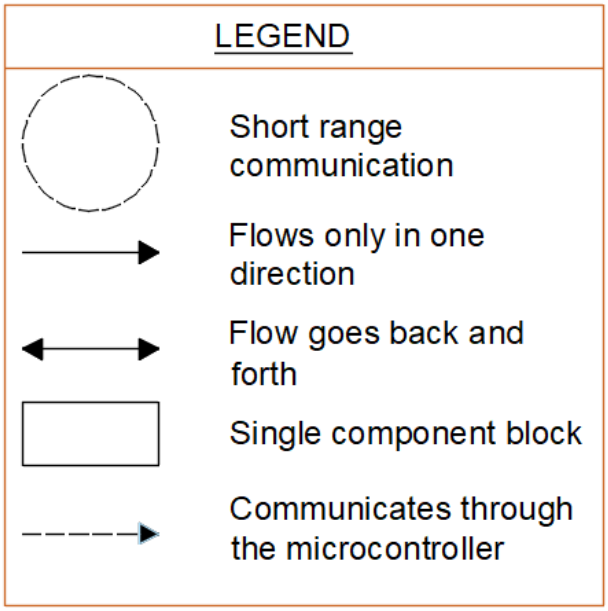


Figure 2-5: Hardware Block Diagram

2.8 Microcontroller Software Diagram

Figure 2-6 gives us an overview of how our group plans to code the microcontroller inside our tag device. Table 2-3 describes the states the microcontroller will enter depending on the situation. Although not described in this overview it will be a priority to utilize deep sleep states and interrupts throughout the software design of the microcontroller.

States	Description
Not Registered	The device has not been connected to the user's iPhone; the tag will advertise a signal periodically.
Initialize	The device has registered and public keys have been generated and shared between the Device and iPhone
Disconnected	The device is out of range of users' iPhone
Connected	The device is in range of the iPhone and stops all signal broadcasting

Table 2-3: States of UWB Tag Microcontroller

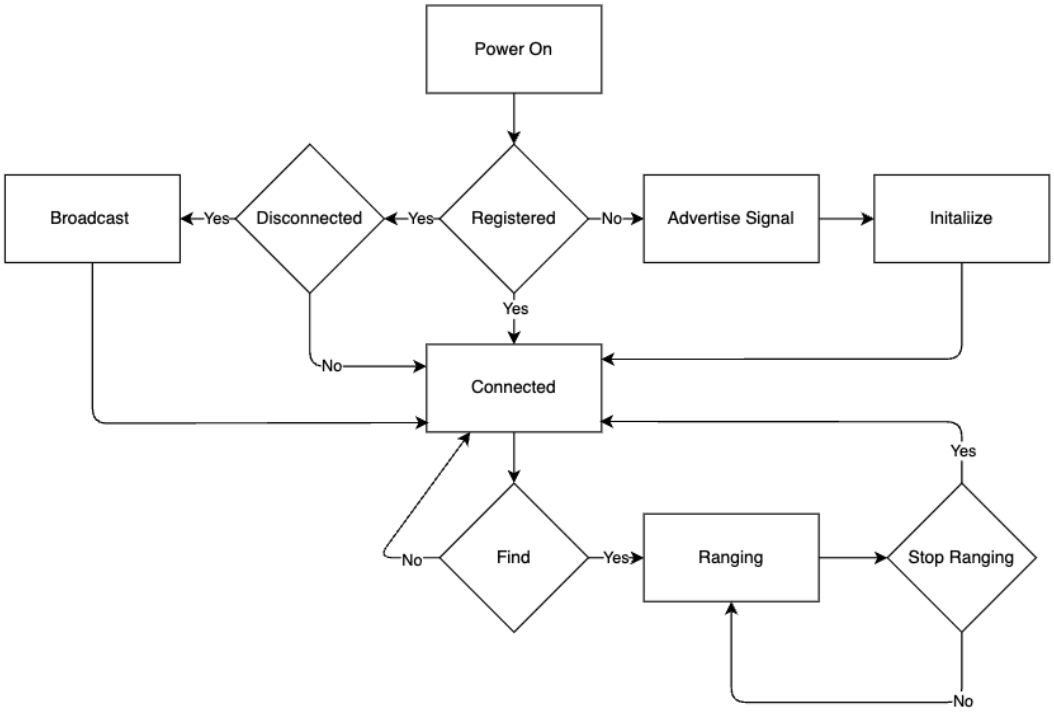


Figure 2-6: Microcontroller Software Block Diagram

2.9 Mobile Application Block Diagram

This section will be an overview of the mobile app that will interface with our device, a overview can be found in figure 2-7. This app will be on an Apple iPhone 13 Pro for this project but will work as standard when used on an Apple iPhone equipped with a U1 chip and using the operating system iOS 16. Since iOS will be a requirement the app will show a not supported page if the user is not using the correct operating system. Upon opening the app, the user must grant permission to the app to let it use the Bluetooth capabilities of the phone. BLE (Bluetooth Low Energy) will be equipped on the tag device we will create for this project and will be how the phone and our device connect.

The smartphone's Bluetooth will search for a Bluetooth signal from the tag device and connect if found. There will be no need to ask the app to search, if the application is open it will search for a device. When connected to our tag the home screen will be displayed for the user to interact with. While we will cover the main goals of this project in this section, we plan to add more features that will interact with the finder on this home page inside the app. For now, the main purpose of the applications is to establish an ultra-wideband connection with the two devices to begin ranging. To activate this feature, we will need to have permission from the user to use the phone's ultra-wideband capabilities. From there the ranging will begin and there will be a distance interface that will update in real-time for the user to look at. To stop the ranging the user will need to interact with the interface button. The user will then be sent back to the home screen inside the mobile application.

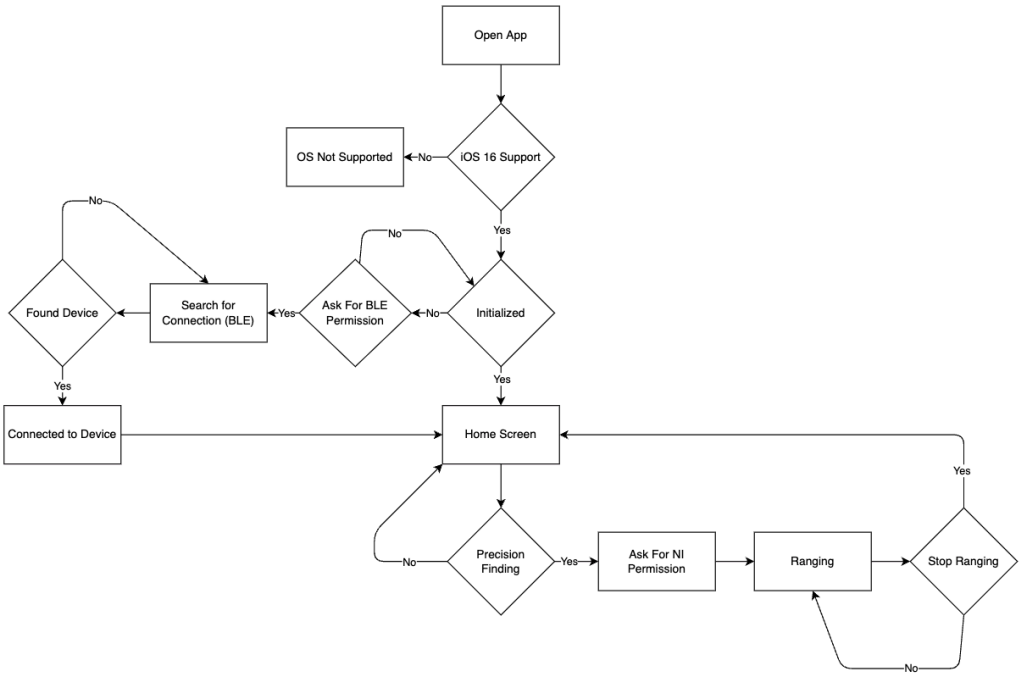


Figure 2-7: Mobile Software Block Diagram

3. Technology Investigation & Part Selection

This section will show the extensive consideration of part selection along with technology investigation related to this project.

3.1 Ultra-Wideband (UWB)

Ultra-wideband is a short-range wireless communication that is different from the traditional communications widely used today. Unlike Bluetooth technology or WIFI, it uses a wide range of frequencies with minimal power. Ultra-wideband is a fairly new technology with many companies beginning to implement it into their business model.

Some potential UWB applications:

- Wireless communication
- High-speed local networks
- Military/airport communication
- Short range radios
- Radar and sensing
- Surveillance cameras
- Through Wall sensing for fire, police officials
- Ground penetrating radar
- Location detection
- Precision location
- Airport travel guide
- Lost device finder

3.1.1 UWB communication

Ultra-wideband uses a wide range of frequencies which are from 3.1GHz to 10.6GHz. This allows radio channels to have a bandwidth of around 500MHz if the center frequency isn't too close to the frequency ends. This wide range of frequencies allows for short communications with speeds in the Gigahertz range. This is exceptionally fast for wireless communication. Now because ultra-wideband can have low frequencies, this allows data to not attenuate from buildings and walls, thus having packets not get lost in communication. Lower frequencies also allow for minimal power consumption which in all achieves better performance.

UWB works by sending out pulses and using the time in between those pulses to identify the signal. If two signals need to share the same channels one can modulate based on pulse polarity, pulse modulation, or pulse amplitude. With it only sending out short pulses this also adds to the minimal power consumption.

Time of flight (ToF) is used in wireless communications to decipher the distance of one object to the receiver. How it works is the receiver will send a signal to the

object for it to bounce back to the receiver. Ultra-wideband leverages this technique in three different ways depending on the application's needs.

3.1.2 Time Difference of Arrival (TDoA)

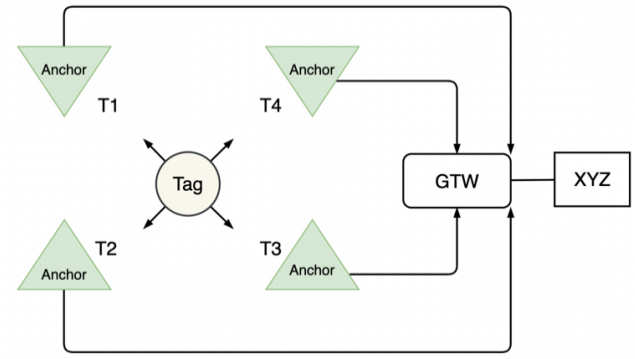


Figure 3-1: TDoA Diagram

The time difference of arrival uses multiple anchors that are stationary in position time synchronized with each other. A phone will send out a beacon that the anchors will receive and the time it takes for the beacon to be received will be time-stamped and then sent to the central unit which will run algorithms that will calculate the position depending on the time difference of each path. This will allow thousands of devices to simultaneously work at the same time. The downfall is that the anchors need to be time synchronized and if any one anchor is out of sync, then the whole operation will fail.

3.1.3 Two-Way Ranging (TWR)

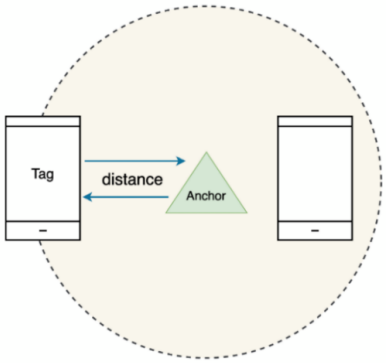


Figure 3-2: TWR Diagram

The two-way ranging is the simplest of the three, it works by having two devices with two-way communication sending a signal that measures the time of flight. The signal then gets multiplied by the speed of light and then divided by two to get the 2-dimensional distance. A 3-dimensional distance can be achieved by

implementing a third device, this is known as triangulation. Unlike the TDoA, TWR has a higher power consumption and is limited to a few hundred devices at a time.

3.1.4 Phase Difference of Arrival (PDoA)

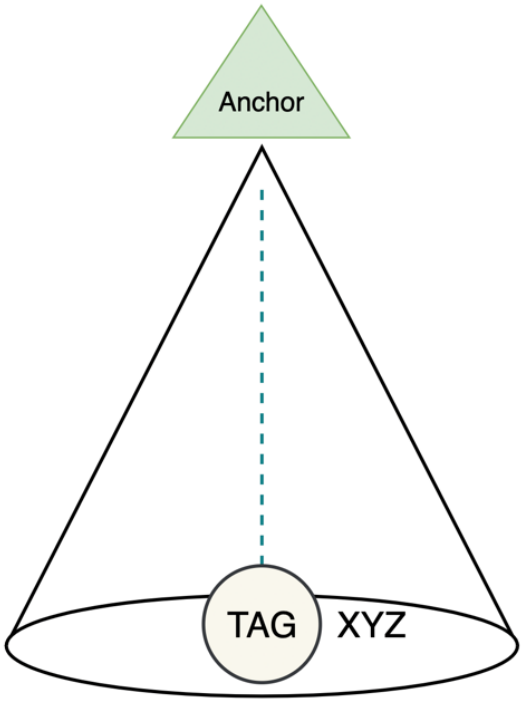


Figure 3-3: PDoA Diagram

The phase difference of arrival consists of the two-way ranging for distance but also implements the angle between the two devices. The combination of distance plus angle difference allows accurate positions of the two devices without complex infrastructure.

After doing some research, we decided UWB was the best fit for our project goals. With it being low power consumption, we know that we will have a long-lasting device paired with a replaceable battery when necessary. UWB also has a great range of up to 300 meters theoretically and with precision down to 50 cm. With ultra-wide bands, we don't have to worry about lost connections as we do with classical Bluetooth. Also, UWB having better immunity to narrowband interferences, allows it to penetrate materials. This is important because in our application we plan on using it in an indoor setting.

3.2 Ultra-Wideband IC Considerations

Our group made it a priority to find ultra-wideband ICs that were interoperable with Apple’s U1 chip in their iPhones. Given this compatibility would mean only needing one responder to accomplish the goals of our project. It would give a consumer an intuitive way to interact with our device and features that our group can use

towards development given Apple's frameworks. We also confirmed from our market analysis and our house of quality that it was a must-feature for our finder to operate using precise positioning. Which is something ultra-wideband will give our tag device the means of doing.

From our UWB (ultra-wideband) investigation our group has concluded we want an IC that supports two-way ranging and time difference of arrival and or phase difference of arrival. Two-way ranging is a method of two units with radio frequency trading the time of transmission and reception. In this case, one of those units would be an Apple iPhone equipped with a U1 chip and the other unit an Ultra-wideband IC. TDoA (time difference of arrival) is a unique method of determining the position of the transmitter. Location is determined by calculating the difference in time of arrival on the transmission of two units. While PDoA (phase difference of arrival) is using the phase difference between the signal received to determine the direction of transmission. These are all beneficial to our project goals.

Though Ultra-wideband is a relatively new technology in the consumer world our group considers it will be the most useful wireless communication option to include in our tag. Giving the consumer of the device a real-time positioning of the item, they have attached the tracker. Furthermore, gives engineers the to expand on this technology's use cases.

3.2.1 Qorvo DW3110

This integrated chip is part of Qorvo's second generation of ultra-wideband transceivers. It is capable of asset location to what Qorvo states is 10 centimeters. While real-world applications might give us worse results, at best the accuracy is much higher than what we originally set our engineering requirement for. Qorvo has made this ultra-wideband transceiver with real-time location systems in mind. Offering key features such as low power consumption. The DW3110 is intended for coin cell applications, perfect for our size constraints and engineering requirement of a device that will be battery-powered. Some of the modules containing the DW3110 come with RF solutions to further simplify the integration with optimized antennae design. Most importantly the DW3110 is interoperable with Apple's U1 chip.

Although very new to the market, releasing in sometime in 2021. Qorvo has documentation of the DW3110 and general DW3000 family of chips. Presenting data sheets, API software guide, family user manual, and application notes for developers to access from their website. Product availability is abundant with most online distributors having their ICs along with the DW3110 in stock and ordering more. Critical to our ultra-wideband development Qorvo offers several development kits with the DWM3110.

All the offers from the DW3110 seem advantageous to our finder goals. Though a few disadvantages stemmed from the provided information and examples still

being in beta or what Qorvo says is production. This posed a problem in finding the correct information as some of the information was out of date. Along with providing examples for the DW3110 using an SDK that was obsolete.

3.2.2 NXP Trimension SR040

The NXP Trimension SR040 is another ultra-wideband IC that is created as an IoT solution. This IC specializes in precision location, ranging, and Apple U1 chip interoperability. Which makes it a perfect candidate to consider for UWB incorporation to our finder. Like the DW3110, NXP has designed the SR040 with optimization for CR2032 coin battery usage, which is one of the batteries we will contemplate using for this project. Similarly, to the Qorvo offering, the SR040 features a location accuracy of 10 centimeters. Some highlights of the NXP SR040 are being based on the ARM Cortex M33 and ARM TrustZone. ARM TrustZone technology provides security for components and the operating system of the CPU.

While the Trimension SR040 has the advantage of being built specifically to be used as an ultra-wideband tag, it doesn't benefit from the lack of support in the development space. NXP's only assistances to the SR040 are a data sheet and high-level explanations of ultra-wideband pairing. In addition to this, there are very few development kits to be bought or are completely out of the budget for our undertaking. Amongst the vendors we rely on for this project the SR040 is not stocked in any, making it a worse candidate in comparison to the Qorvo DW3110.

3.2.3 NXP Trimension SR150

SR150 is a similar integrated chip that can interoperate with Apple's U1 chip. The NXP Trimension SR150 is very alike to the SR040. It includes an ARM Cortex-based chip with TrustZone and It can achieve 10 centimeters of location accuracy. The SR150 is more of a robust design intended for large ultra-wideband infrastructures compared to the SR150. The SR150 can measure the AoA (angle of arrival) with an accuracy of 3 degrees. It also supports 2-way ranging, time difference of arrival (TDoA), and phase difference of arrival (PDoA). All these methods of ultra-wideband operation offer the most flexibility in the applications of our ultrawide-band finder.

Unlike the NXP SR040 and Qorvo DW3110, the TrimensionSR150 is not intended for ultra-wideband tackers and tags. This UWB IC is also not meant for coin cell battery usage, which will be a serious strain on one of the engineering requirements and goals of our device. This IC also suffers from the same disadvantages as the NXP SR040. NXP has sparse documentation on the SR150 only providing a product-short data sheet. An expensive development kit created for manufacturing companies. Unfortunately, market obtainability is nonexistent for this chip, but we will continue to check vendors for the initial start of this project.

3.3 Ultra-Wideband IC Selection

After much consideration on ultra-wideband integrated chips compatible with Apple's U1 chip, we have decided on Qorvo's DW3110. Though UWB components for consumer devices are very new to the market, especially ones compatible with the U1 chip in Apple iPhones, we believe the DW3110 is robust enough to meet our finder requirements. The parameters that were most sought can be found in table 14-3. Our project mostly relies on ultra-wideband technology, so we wanted to find a suitable IC for our needs and more. Parameters such as interface, package type, and ranging accuracy are especially important to project constraints and engineering requirements.

Description	Value
Voltage	2.4V to 3.6V
Antenna Ports	1
Data Rate	6.8 Mbps
Ranging Accuracy	< 10 centimeters
TDoA Support	Yes
Interface	SPI
Package Type	WLCSP

Table 3-1: DW3110 Parameters

Just as important as it is to have the features to use, we need to be able to understand how to use them. Qorvo has detailed application notes on the DW3110 for us to use as guidelines. These application notes include antenna delay calibration, production tests, and maximizing range. Collimation of application notes, user manuals, API guides, and datasheets is another reason why we picked the DW3110, these give our group many resources to work from.

We can also make use of several development kits Qorvo has on the market, considering multiple team members will be working on the embedded proportions of the project. Some of these include the exact nRF52833 microcontroller we have selected, further unifying the process of creating this project. Through the course of this design, we will use the DWM3000EVB Shield which is supported by Nordic's nRF52833 development kit, and the DWM3001CDK which is more of an all-in-one solution.

3.4 Bluetooth Low Energy (BLE)

Bluetooth technology is a popular short-range wireless communication that is standard in almost all devices. Its physical layer operates on the same frequency range as WIFI centered at 2.4GHz. Bluetooth devices are managed in a star topology with one being the master and the others being the slaves, there is a limit to the number of devices connected but no limit to the number of cascades in the star topology. This means that for the one-star topology the master can act as a slave for another star topology. The wireless link is divided into time units known as slots which store the packets of information sent from Bluetooth devices.

3.4.1 Bluetooth Low Energy

Bluetooth low energy (BLE) was introduced with Bluetooth V4.0 back in 2011. BLE is well suited for applications that need to periodically send out small amounts of data. This makes it very well suited with sensors, or low-powered devices, like a tracking device. BLE keeps its power to a minimum by having the device in sleep mode at about 90% of its life, and only active when it's scanning for a connection. For our tracking device, this is a must, we don't want to have a device that needs a battery replacement once a week. One of our goals in this project is to have a long-lasting device that you can just place in one spot and just forget about it until needed. If our device doesn't last that long, then the user would place it somewhere and it would die, and the user is left without knowing where the device is and it would be useless.

BLE was considered for this project because we need a way for our microcontroller to connect to our smartphones. While Bluetooth can be used for location finding, it doesn’t meet our goal for precision. BLE has a range of up to 100 meters but only precise down to 5 meters. For our project we need to be more precise than 5 meters, so therefore we will only implement it in our design as a way of connection.

3.4.2 Location

The Bluetooth indoor location uses either sensors or beacons that communicate with Bluetooth-enabled devices.

3.4.2.1 BLE Positioning with Sensors

BLE Positioning with sensors utilizes sensors in a fixed indoor location with BLE enabled. It will communicate with passing BLE smartphones and gather information depending on the signal strength. This data is then sent to the central indoor positioning system. This computer then analyzes this data and runs it in a multilateration algorithm to determine the location of the device.

3.4.2.2 BLE Positioning with Beacons

BLE positioning with Beacons uses beacons to locate a device. The beacons or in a fixed indoor position will send out pulses of data that will be received by the smartphone or other BLE devices. The signal being sent out contains a unique identification number that only that beacon will have. When in range and the signal is received, the signal will be sent to a server that will then be run in an RSSI multilateration algorithm to determine the location.

Bluetooth over the last decade has drastically increased in performance. The very first Bluetooth generation, Bluetooth 1, had a range of only 10 meters, with a slow speed of 732.2 kb/s. The reason for this comparison is that we expect the same type of growth with UWB technology. Below in table 3-2 shows the different Bluetooth generations with their features.

Factors	Bluetooth 1	Bluetooth 2	Bluetooth 3	Bluetooth 4	Bluetooth 5
Speed	732.2 kb/s	2.1 Mb/s Enhanced Data Rate (EDR)	24 Mb/s (via Wi-Fi)	24 Mb/s (EDR)	50 Mb/s (EDR)
Range	10 meters	30 meters	30 meters	60 meters	240 meters
Compatibility	N/A	OK for any phone but expect some possible sound sync issues	OK for any phone	OK for any phone, but works best if the Bluetooth versions match	Works with any phone, but is best suited for newer phone models
Power Requirement	High	High	High	Mid-High	Low
Reliability	Low	Low	Low	Mid-High	High

Table 3-2: Summary of Bluetooth Generations

- Selected Bluetooth Generation for Project

3.3 Near Field Communication (NFC)

We decided to research near-field communication due to its contactless communication, and proximity. NFC is a new short-range wireless technology that was based on the existing RFID. With NFC two devices can be brought together for a transfer of services. It allows users to integrate their personal credit/debit cards, etc. into their phones, to provide a seamless transaction.

There are three modes of operation for NFC, which include passive mode, active mode, and peer-to-peer mode. In passive mode, the NFC mobile device acts as a standard credit card and then gets read by the NFC reader. In active mode, the NFC mobile device can read or modify an NFC tag. Peer-to-peer mode is a wireless communication between two NFC mobile devices, one acts as a listener while the other acts as the speaker.

While a payment transaction is one of the most popular ways NFC is getting implemented into our society, there are still many other applications of NFC technology such as unlocking car doors, PC access, tickets, building access, etc. it is easy to see that NFC is usually based on a key or an individual transaction. The way NFC is safe and secure is because it uses an integrated circuit with a secure element (SE).

While NFC is a great way to have secure transactions, it would not be a good choice in implementing it in our project as a way of locating our device. For our project, the device needs to be able to send out a signal more than 10 meters away. NFC lacks range and frequency bandwidth. Additionally, NFC can only have a max of 2 devices communicating with each other. Many of the trackers on the market use Bluetooth and when another Bluetooth device is in proximity it will ping out a signal automatically. This isn't possible with NFC because it is based on a secure connection and would need the passing-by device to authenticate a connection, which is not likely.

This technology could be implemented with existing technology such as Bluetooth or UWB, and we might consider it as an additional feature as we progress. We could authenticate that the device was found using NFC. For example, when the user finds the device they can tap on it with their smartphone, and NFC can prove authentication. This can also work as a security measure having one user per device, and not allowing a connection for an unauthenticated device. Below in table 3-3, shows the features and specs of NFC.

	Type 1	Type 2	Type 3	Type 4	Type 5
Standard	ISO/IEC 14443A (Type A)	ISO/IEC 14443A (Type A)	ISO/IEC 18092 JIS X 6139-4 FELICA (Type F)	ISO/IEC 14443A ISO/IEC 14443B (Type A + Type B)	ISO/IEC 15693
Memory	96 bytes to 2Kb	48 bytes to 2Kb	2Kbytes	32Kb	64Kb
Data Rate	106 kbit/s	106 kbit/s	212 kbits/s 424 kbit/s	106 kbits/s 212 kbit/s 424 kbit/s	26.48 kbit/s
Capability	Read Re-write Read-only	Read Re-write Read-only	Read Re-write Read-only	Read Re-write Read-only Factory-configured	Read Read-write Read-only
Anti-collision	No	Yes	Yes	Yes	Yes
Notable	Simple, cost-effective	None	Higher cost, complex applications	None	Vicinity area

Table 3-3: Summary of NFC Types

- Selected NFC Type

3.4 Microcontroller

A Microcontroller is a small, lightly powered computer on an integrated chip consisting of a CPU, RAM, ROM, and I/O ports. They are designed to perform a single task of an embedded system and are the heart of most household appliances. Microcontroller hardware costs are low and easy to replace, which makes it suitable in design. For our design, it is necessary to implement a microcontroller. We need a way for our UWB chip to communicate with the software in our phones by either SPI or I2C. We need to research different microcontrollers and see which one will fit best our needs based on our hardware and software specifications. Not every microcontroller is built the same and most are built to perform a specific instruction, so it's important to understand the microcontroller's specifications. We will need a microcontroller that runs most of its life in low-power mode and we also need it to be relatively fast.

Microcontrollers are in a wide range of applications, some applications include:

- Microcontroller applications
    - Dryer/washer
    - Sensors
    - Automobiles
    - Mobile phones
    - Fire alarms
    - Electronic measuring devices

3.4.1 Types of Microcontrollers

Microcontrollers are divided into different categories depending on the bits, memory, architecture, and instruction set:

- Bit
    - 8-bit
    - 16-bit
    - 32-bit
- Memory
    - External memory
    - Embedded memory
- Instruction set
    - RISC
    - CISC

PIC microcontrollers are the smallest microcontrollers in the world, they are often found in phones, tablets, and laptops. Even though they are the smallest in size, they can be programmed to perform a wide variety of tasks. The architecture of PIC microcontrollers consists of stacks and registers. The stacks save the return address, and the registers act as random-access memory (RAM).

The ARM Microcontroller is a 32-bit microcontroller that has its instructions based on the 32-bit architecture. Each instruction set is performed on a Boolean algebra expression and executed conditionally. The ARM microcontroller is one of the most advanced microcontrollers with many devices having them integrated inside. The ARM microcontroller has different versions like ARMv1, ARMv2, etc. with each having different advantages over the other. The ARM processor is different from the other microcontrollers because it has components like program status registers. The program status register contains the processor flags that will show interrupts within itself. The 8051 microcontroller is a 40-pin, 8-bit microcontroller with a required power supply of 5 volts. The memory of the 8051 consists of two parts, program memory and data memory. The program memory stores the program that is being executed while data memory stores the data that is concluded from the results.

The AVR microcontroller is an 8-bit microcontroller with an RISC architecture. Because of this architecture, the AVR is fast, with the ability to execute an instruction set in one cycle. AVR spends most of its life in power savings mode and thus resulting in low power consumption. The MSP microcontroller, marketed by Texas Instruments, is a 16-bit ultra-low power based on the RISC architecture. This microcontroller has an analog, digital, and microprocessor all integrated on a single chip. In terms of speed, this microcontroller can run there a cycle in 125ns, this is due to its 8MHz crystal. Table 3-4 shows a comparison of the 5 different types of microcontrollers.

Microcontroller	Bits	RAM	Speed	Instruction set	Power
PIC	8-Bit	16 X 256 Bytes	Moderate	RISC	Low
ARM	32-Bit	96k Bytes	Fast	RISC	Mid-Low
8051	8-Bit	128 Or 256 Bytes On-Chip	Slow	CISC	Mid-Low
AVR	8-Bit	32 Bytes Up to Several KB	Fast	RISC	Low
MSP	16-Bit	512k Bytes	Fast	RISC	Low

Table 3-4: Summary of the 5 Different Microcontrollers

- Selected Microcontroller Architecture

3.5 Microcontroller Considerations

When considering a microcontroller for the project our group considered many technical aspects ahead. Such as a microcontroller unit that supported at least two interfacing protocols such as SPI and or I2C. A controller that has enough flash memory and RAM to meet the most features we would see possible for this project scope. Lastly, a microcontroller that could meet our power consumption constraints. These properties add flexibility to our design in case of unforeseen details and make it possible for us to complete the goals of our device.

Another important factor was to use a microcontroller with ARM architecture. This was important for many reasons. Microcontroller ARM Architectures are used extensively by some of the biggest corporations in a wide range of products. From our investigation early in the project, we have found support for this architecture to be broad, increasing the parts available to us. We can choose a microcontroller to fit our needs without having to worry about sparse documentation, tools, and development kits. We also want interoperability between the ICs we plan to incorporate, such as the ultra-wideband IC into our design. Bearing in mind all these aspects will eliminate the need to switch parts and find a microcontroller suited for our group's tag device.

3.5.1 Nordic Semiconductor nRF52840

The nRF52840 is Nordic's most advanced SoC (system on a chip) in their nRF52 series. It features the same 64MHz ARM Cortex-M4 processor found in our other considerations but offers the most flash memory and RAM. It also features Matter, an open-source application layer that is trying to be a standard across smart homes. Matter aims to create a unification of smart home devices and mobile applications through many technologies such as Bluetooth, Wi-Fi, and Thread. These standout features of the nRF52840 will be great to have, especially the more flash and memory. While Matter although still in the experimental phase and outside the scope of our project, would be important to consider in terms of our project being able to evolve.

nRF52840 was built with an IoT solution in mind. So, it has many of the relevant technologies we need for our project and more. Features such as Bluetooth LE, SPI, and 1.7 to 5.5 supply voltage range. It also includes additional features we may want to use in our devices such as NFC and a security subsystem. Most importantly, Nordic offers affordable hardware development kits that have many GPIOs and an onboard SEGGER J-Link for debugging. These dev kits also feature a lot of information on Nordic's Dev Zone. The Nordic Dev Zone is their website where people can post problems they are experiencing, ask a question, or find solutions to problems other users have posted in the past. On top of this development kit and website, this microcontroller also has an SDK (software development kit) that makes it easy to start developing and integrating the technologies included on the nR52840. The key specifications of this

microcontroller for our development can be found in table 3-5 below. The biggest advantage of the nRF52840 will be the flash memory and RAM.

	Key Features
Nordic Semiconductor nRF52840	64MHz ARM Cortex-M4 Processor
	1MB Flash
	256KB RAM
	Bluetooth 5.3 LE (long range)
	IEEE 802.15.4 Radio Support (Thread, Zigbee)
	1.7 – 5.5 Supply Voltage Range
	32MHz SPI
	NFC Type A
	UART/QSPI/SPI/TWI/I2S/PDM/QDEC

Table 3-5: nRF52840 Key Features

3.5.2 Nordic Semiconductor nRF52833

Similarly, to the nRF52840 the Nordic nRF52833 has many of the same key features. Some of these offerings include a 64 MHz Arm Cortex M4 processor, Bluetooth 5.3, and a 1.7 to 5.5 voltage supply range. Fortunately, like the nRF5240, the nRF52833 has huge availability of affordable development kits. Nordic also offers the same SDK for the nRF52833 with abundant documentation to begin development faster.

Like the Nordic nRF2840 supports Bluetooth 5.3, Bluetooth long-range, and Bluetooth mesh. The nRF52833 is capable of all those features and Bluetooth direction finding. Bluetooth usually relies on RSSI (received signal strength indicator) to consider the position of an item. Bluetooth direction find can get the actual direction of the sign by using the AoA (angle of arrival) and the AoD (angle of departure) of the transmitted signal. Although our project will heavily rely on ultra-wideband capabilities, having Bluetooth direction find could extend the abilities involving Bluetooth features in our tag device.

While the nRF52833 has the advantage of Bluetooth direction and more mature hardware in comparison to the nRf52840. The nRF52840 features almost twice as much flash and RAM. In this regard picking the nRF52833 over the nRf52840 could

mean lower headroom for our project. This will be something to consider for our project along with key features of this microcontroller that can be found in table 3-6 below.

	Key Features
Nordic Semiconductor nRF52833	64MHz ARM Cortex-M4 Processor
	512KB Flash
	128KB RAM
	Bluetooth 5.3 LE (long range, direction finding)
	IEEE 802.15.4 Radio Support (Thread, Zigbee)
	1.7 – 5.5 Supply Voltage Range
	32MHz SPI
	NFC Type A
	UART/SPI/TWI/I2S/PDM/QDEC

Table 3-6: nRF52833 Key Features

3.5.3 NXP QN9090

The QN9090 is the most recent microcontroller to be considered for this project, it was released sometime in 2020. It offers an ARM Cortex M4 processor like the other microcontrollers but at a lower 48MHz clock rate. Which might prove beneficial to the overall power consumption of our finder. The QN9090 features 640KB of flash and 152KB of RAM, providing a little more headroom than our second microcontroller choice, the Nordic nRF52833. This microcontroller has Bluetooth LE like the other considerations but only BLE 5.0. Compared to Bluetooth Low Energy 5.3, BLE 5.0 is not as polished. It does not offer less interference, lower power consumption, and better connectivity than Bluetooth 5.3 can provide like in the other microcontroller considerations.

The NXP QN9090 offers similar specs to the Nordic microcontroller considerations such as processor, flash, RAM, DC/DC converter, embedded Bluetooth Low Energy, and NFC type A. NXP also has an SDK and tools to get started. Although one shortcoming of this microcontroller is the lack of availability of parts. Development kits have either very low or no stock at the time of this writing. The Obtainability of NXP QN9090 seems to be very low too with distributors having little to no inventory. On the positive side NXP, QN9090 may suggest easy integration

with their ultra-wideband modules. Given that NXP offers a few options to choose from including dev kits pairing their microcontrollers and ultra-wideband integrated chips we might simplify the number of software, SDKs, and tools needed for our project. The key specifications of this microcontroller for our development can be found in table 3-7 below.

	Key Features
NXP **QN9090**	48MHz ARM Cortex-M4 Processor
	640KB Flash
	152KB RAM
	Bluetooth 5.0 LE
	1.9 – 3.6 Supply Voltage Range
	NFC Tag
	UART/SPI/I2C/PDM

Table 3-7: QN9090 Key Features

3.6 Microcontroller Selection

For our ultra-wideband tag device, we have decided to select Nordic's nRF52833 as the microcontroller. Many of the sought-after specifications needed for our project can be found in the nRF52833. The Cortex-M4 inside the nRF52833 is powerful enough for our application. It features 512KB of flash and 128KB of RAM, which should be adequate for the scope of this project. Some of the biggest deciding factors of our microcontroller selection come down to the power management of the microcontroller, Bluetooth capabilities, development tools/hardware, and obtainability. An overview of selection reasoning can be found in table 14-2. The WLCSP (3.2mm x 3mm x 2mm) package option will be useful for our size constraints too.

The nRF52833 is purposed for engineers to make ultra-low powered wireless devices with. There are many Bluetooth features such as having BLE 5.3, long-range, and direction finding. While most will directly fit our needs for this device, some technologies will give us the freedom to expand on other areas in our group's development. Additionally, Nordic offers many tools to complement the functions of the nRF52833. These tools include nRF Connect SDK, which will be key to our integration of the nRF52833's technologies by using Zephry RTOS, protocol stacks, and examples, included inside this SDK.

Nordic's Bluetooth protocol stack, SoftDevice, will also be convenient for us for the same reasons. Another factor in choosing the nRF52833 is the plentiful documentation that can be found on their website. Not only do we have access to schematics, SDK guides, datasheets, and family user guides, but there will also be Nordic's technical support on their DevZone website to help us with any technical questions or problems we might have.

	Nordic nRF52833	Nordic nRF52840	NXP QN9090
ARM Cortex-M4 Processor	✓	✓	✓
Bluetooth 5.3 (2 Mpbs)	✓	✓	
Bluetooth Long Range	✓	✓	
Bluetooth Direction Finding	✓		
SPI Digital Interface	✓	✓	✓
NFC	✓	✓	✓
Intended for Asset Tracking	✓		✓

Table 3-8: Microcontroller Selection Reasoning

- Selected Microcontroller

3.7 DC/DC Converter

A DC-DC converter is an electronic circuit or device that converts one DC voltage to another DC voltage. It can convert a high voltage to a low voltage, or a low voltage to a high voltage. DC-DC converters usually consist of some semiconductor device (diode or transistor) and one passive element (capacitor or inductor). It was invented because many integrated devices need to have different voltages for their sub-circuits to operate. So instead of having one battery per sub-circuit, a DC-DC converter was implemented so one battery can supply the voltage to each sub-circuit. Devices like smartphones or laptops have DC-DC converters and they must have them.

With our project, there is going to be a microcontroller with some sub-circuits, so research on a DC-DC converter is needed. We plan on using a Replaceable CR2032 battery as our main power supply which produces 3 volts. Depending on the remaining circuit we may need to increase the voltage or decrease the voltage for our microcontroller. There are four different types of DC-DC converters, and each is necessary depending on the type of application needed below is a brief description of them.

3.7.1 Boost Converter

A boost controller is a type of DC-DC converter that will increase the voltage while decreasing the current with respect to the input power supply. So, this converter is great for applications that need a higher voltage to operate but not a huge load of current. Usually works best with microcontrollers that don't need a large amount of current to operate.

3.7.2 Buck Converter

Buck converters are mainly used when the input voltage is high like 12V, and a low output voltage is desired. For example, a PC will need a huge power supply to operate but the USB only needs about 1.8V so a buck converter lowers the voltage for the USB port to operate in spec. The circuitry in a buck controller is simple, usually containing two semiconductors and one energy-storing element.

3.7.3 Buck-Boost Converter

A buck-boost converter is a combination of both buck and boost converters. It is made up of the same elements as both so it's capable of increasing the voltage or decreasing the voltage. Because of the more elements, one downside is that it has a larger footprint than the others.

3.7.4 Cuk converter

A Cuk converter is like a buck-boost converter but with a switch implemented in the middle. On the input side, you will see the boost converter and on the output side, you will see the buck converter. One difference between the Cuk converter vs the buck-boost converter is it produces a low ripple current due to the inductors. It's also an inverting converter so the output voltage will be negative with a positive input voltage. The four different types of converters, as well as the spec, are shown below in table 3-9.

Converter	Optimal Power Demand	Efficiency	Number of Components	Voltage Stress	Cost
Boost Converter	Low <500W	High	4	High	Low
Buck Converter	Low <500W	High	5	High	Low
Buck-Boost Converter	Low <1kW	Medium	7	Medium	Medium
Cuk Converter	Low <1kW	Medium	6	High	Medium

Table 3-9: Summary of The Different DC/DC Converters

- Selected DC/DC Converter

3.8 UWB DC-DC Converter Considerations

For the integration of the ultra-wideband transceiver to work, our group needs to think about options for buck converters. The ultra-wideband ICs that were considered all need a buck converter to set down the DC voltage it will be receiving from an internal battery. By design, only the ultra-wideband ICs will need this DC-to-DC converter because all our considerations for microcontrollers feature an onboard buck convertor. Figure 3-1 represents a visual aid for this.

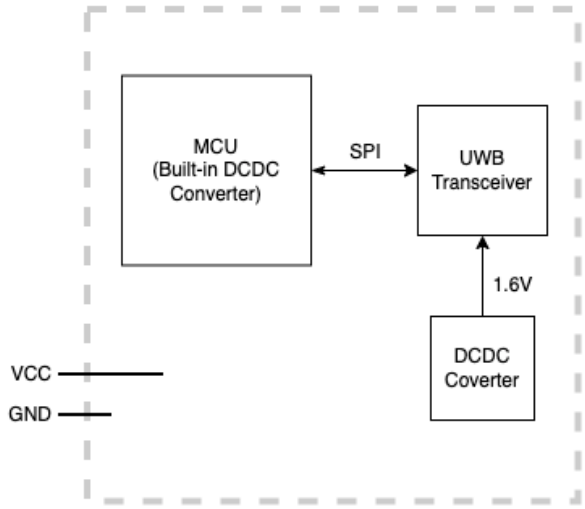


Figure 3-1: DC/DC Converter for UWB Block Diagram

When determining a buck converter for the UWB IC we pulled information from several different areas. First by using WEBENCH power designer. With Ti's WEBENCH we could specify min and max input voltage along with the output for voltage and current. By focusing on designs that were high efficiency we were able to find converters suitable for our finder. Another area of component investigation involved device teardowns of similar products on the market. Additionally, we were able to find information for suitable DC-to-DC buck converters by following design guidelines recommended by manufacturers of the UWB ICs.

For our deliberation of switching regulators, we considered certain specifications that will match our engineering requirements. Parameters for the step-down converter of the UWB IC include input voltage range, output voltage range, output current, and most significantly quiescent current. It is imperative to accommodate input power fluctuations a switching regulator will provide the flexibility needed for our tag. Most importantly we want to minimize the current drawn by the system in standby mode. Seeing as our ultra-wideband IC will be in a state of sleep most of the time until the user chooses to use the functionality. Low quiescent current will be necessary to boost efficiency.

3.8.1 Torex Semiconductor XC9282

The XC9282 is the first switching regulator to be inspected for DC-to-DC step-down for the ultra-wideband integrated circuit. Torex has created this converter for applications that are small, wireless, and sourced with low power. The XC9282 was first found in some of Qorvo's DW3000 family of ultra-wideband modules. These modules employed several components that our group has considered for this project including the XC9282. While not having as low of quiescent current as some other considerations in this section, the Torex XC9282 has desirable

characteristics. A notable feature of this converter is Torex's high-speed transient response technology that they call HiSAT-COT. Furthermore, the XC9282 is available in two packages, the LGA-6B01 (1.2mm x 1.2mm x 0.3mm) and the WLP-5-06 (0.88mm x .96mm x .0.33mm). These key features can be easily readable in table 3-10.

Although the XC9282 is applied to a few of Qorvo's designs that have their ultra-wideband IC, Torex does not recommend this series of DC-to-DC step-down converters for new designs. Torex has recently at the time of writing made this regulator component obsolete. The XC9282 can still be purchased from major part distributors, but once global stock is sold out vendors will not restock this series of switching regulators.

This may prove problematic in terms of sourcing this component in the following months. From a manufacturing standpoint, it is also not wise to create a device with obsolete products. However, our team thinks utilizing modules with this step-down regulator will streamline the development of this project which will outweigh this con. Moreover, without sacrificing the engineering requirements and goals of this tag device. We will still consider the Torex XC9282 because of the solution it provides in off-the-shelf modules existing from Qorvo.

Input Voltage Range	2.5V to 5.5V
Output Voltage Range	1.2V to 3.6V
Output Current	600 mA
Quiescent Current	11 µA (11000 nA)
Price	$0.84

Table 3-10: XC9282 Key Features

3.8.2 Texas Instruments TPS62746

The TPS62746 is another buck converter devised for ultra-low power applications including Bluetooth low-energy products in mind. Texas Instrument's TPS62746 is a very efficient step-down converter posing a low quiescent current of 360 nanoamps. This converter only comes in one package, WCSP (1.6mm x 0.9mm x 0.9mm). Making it a challenge if we cannot work in the package size to our final PCB. Texas Instrument's TPS62746 also features DCS-Control, a regulation topology technology that offers a smooth transition into power-save mode. The crucial characteristics of this switching regulator can be found in table 3-11. A significant benefit of the TPS62746 out of all the buck converters is it is the cheapest. Quantity available for the global market from wholesalers is plentiful. Including Texas Instruments supplying direct orders and samples.

This component was first sourced from Apple AirTag teardowns. Investigation into some of the components used in the AirTag the TPS62746 can be found in Apple's design being used to regulate the DC power going to their U1 chip. Notably, the TPS62746 was found again in Texas Instruments WEBENCH Power Designer after entering our sought-after DC-to-DC input and output. Though it won't necessarily mean our finder will be successful, it is positive to know will be using a component that has a proven track record in a highly developed product that is already on the market.

Input Voltage Range	2V to 5.5V
Output Voltage Range	1.2V to 3.3V
Output Current	300 mA
Quiescent Current	360 nA
Price	$0.46

Table 3-11: TPS62746 Key Features

3.8.3 Texas Instruments TPS62840

This buck converter is another part made by Texas Instruments. The TPS62840 is a highly efficient step-down converter with a very low quiescent current of 60 nanoamps. It was amongst the highest-ranked design found on WEBENCH, having an 88% efficiency. This switching regulator also comes in several packages that will be convenient for prototyping and the final PCB. These packages included SON-9 (1.5mm,1.5mm, 2mm), WCSP-8 (0.97mm x 0.97mm x 1.47mm), and HVSSOP-8 (3mm x 3mm x 5mm). It also has Texas Instrument's DCS-Control and abundant samples plus inventory to be purchased from their website.

Input Voltage Range	1.8V to 6.5V
Output Voltage Range	.8V to 3.6
Output Current	750 mA
Quiescent Current	60 nA
Price	$0.65

Table 3-12: TPS62840 Key Features

	Torex XC9282	TI TPS62746	TI TPS62840
Input Voltage Range	2.5V to 5.5V	2V to 5.5V	1.8V to 6.5V
Output Voltage Range	1.2V to 3.6V	1.2V to 3.3V	.8V to 3.6
Output Current	600 mA	300 mA	750 mA
Quiescent Current	11 µA (11000 nA)	360nA	60nA
Price	$0.84	$0.46	$0.65

Table 3-13: Summary of DC/DC Buck Converters

- Selected DC/DC Converter

3.9 Ultra-wideband DC/DC IC Selection

Taking into consideration the several options we reviewed for DC-to-DC buck converters. We have decided to use the Torex Semiconductor XC9282 as a regulator to step the DC voltage down to the ultra-wideband IC. Despite this part being EOL (end of life) we still think the specifications will be useful for us within the realms of this project's time. The XC9282 has a low quiescent current, an input voltage range, and the output we need.

More so the XC9282 is already integrated with modules that have been developed to work together with parts we have decided on using. These modules contain the Torex XC9282, Nordic nRF52833, and Qorvo DW3110 working together. Using the Torex XC9282 will mean less testing in terms of trying a different buck converter.

A shorter BOM (bill of material) since we will be ordering a module with all the necessary components to develop. Shorter development time due to operating with kits that already have our components together. Also, shorten assembly time since the components will be soldered all together on a convenient segment, we can then augment our PCB. As a group, we think there are too many advantages to using this DC-to-DC step-down and only a few disadvantages to not.

3.10 Button Cell Battery

Button cell battery is one of the most used batteries worldwide. It is commonly used to power small devices such as wristwatches, calculators, computer motherboards, security devices, toys, fitness appliances, hearing aids, automobile key fobs, tracking or location devices like Samsung and Apple tags, and low-power consumption gadgets. These batteries are called button cell batteries or coin cell batteries because they resemble a button. There are many kinds of button cell batteries on the market and each one of them is built from different types of materials and has its chemistry. Some of them have cylinder shapes and are considerably long, but on the other hand, some are very small in size.

Button cell battery comes in different sizes, and it is intended to be used once and replaced once the battery life has been drained. However, today's technology has permitted scientists and engineers to create button cell batteries that are rechargeable. The only downside with this new technology is that it is a little bit pricey and requires the user to also buy a charger, which will cost the user approximately $25 to $30 in total. This has a huge economic impact on the consumer's end because a regular battery price ranges from $1.99 to $2.99, which is about 10 to 13-fold higher than what that person could have spent. Even though that technology will be profitable in many aspects in the long run, both economically and environmentally, consumers will always tend to be attracted to the lowest price possible and what seems to be more convenient for their current ongoing situation.

As we can see in table 3-14, there are a variety of batteries such as Lithium, Alkaline, Silver, Zinc-air, Nickel Oxyhydroxide, and Mercury Batteries. It could also be noted that each one of these batteries has a specific letter code, which identifies their chemical compositions according to the IEC standard system and it also refers to the nominal voltage that these batteries could hold. The way that these batteries keep their energy is by the electrolyte that they have. The electrolyte is what is going to keep the flow of energy going and it is the heart, in e terms, of the battery. Its role is to serve as a catalyst, so the battery can conduct energy by stimulating the movement of ions from the cathode to the anode on charge and in reverse on discharge. The electrolyte of a battery could consist of many things such as soluble salts, acids, gelled and dry formats, liquid base, and more.

Furthermore, these electrolytes could also be organic or alkali, which is going to be the ones that we are considering for our project. Also, depending on which

battery technology one is using one should make sure to keep regularly maintained by adding the necessary amount of electrolyte needed to their battery or by charging them if their energy is renewable.

The users should make sure that the weather conditions that the battery they are using should be are respected because that could cause the battery to become defective or explode if they are in hotter conditions. From an economic perspective as well, users should ensure that the electrolyte inside of these batteries does not drain quickly because that would imply that the battery life will be shorter, and they will most likely have to buy a new one if needed.

As we can see in the table below, the technology that these batteries use vary. Some of them have common negative electrode components, Lithium and Zinc, but different types of positive electrodes will form chemical reactions. Electrodes are an essential part of any battery because they oversee its potential energy. Electrodes are divided into two categories; one is Anodes and Cathodes. An anode is an electrode where an oxidation reaction takes place which causes the loss of electrons, and a cathode is an electrode where a reduction takes place which means a gain of an electron.

It is to be noted that the positive electrode has a higher potential than the negative electrode. When referring to positive and negative electrodes, they can be named cathode or anode depending on the state of the battery. If the battery is discharging the positive electrode becomes a cathode and the negative electrode becomes an anode. However, if the inverse state is happening, meaning that the battery is charging, the positive electron then now becomes the anode and the negative one becomes the cathode.

Mercury used to be the most popular battery that was used for some decades, but because of how toxic it was to the environment and that they were not fully recyclable, it had to be banned and had its standards taken away by, both, ANSI and IEC. These batteries were a reference and were used in many electronics. However, due to the concerns that many entities all around the world, they were first banned in Europe in 1992 by the European Commission directive, and then in the US in 1992, in New Jersey. However, they were officially banned in the whole United States in 1996 when the Mercury-Containing and Rechargeable Battery Management Act of 1996 was voted.

Name	Code (Letter)	Electrolyte	(+) Electrode	(-) Electrode	Nominal Voltage	End-Point Voltage
Lithium Battery	C	Organic	Manganese Dioxide	Lithium	3	2
	B	Organic	Carbon monofluoride	Lithium	3	2
	G	Organic	Copper Oxide	Lithium	1.5	1.2
Alkaline Battery	L	Alkali	Manganese Dioxide	Zinc	1.5	1
Silver Battery	S		Silver Oxide	Zinc	1.55	1.2
Zinc-air Battery	P		Oxygen	Zinc	1.4	1.2
Nickel Oxyhydroxide Battery	Z		Manganese Dioxide, Nickel Oxyhydroxide	Zinc	1.7	1.5
Mercury Battery (Banned)	M		Mercury Oxide	Zinc	1.35	1.1
	N		Mercury Oxide	Zinc	1.40	1.1

Table 3-14: Summary of Button/Coin Cell Battery Types

- Selected Battery Type

3.11 Button Cell Battery Considerations

In section 4.8, we talked about many relevant button cell technologies that are used around the world. We also talked about their specifications, banned types of batteries, and building materials, which provided insights into what their voltages are and what their limits could be. We have decided to consider only one of them, which is the lithium battery that has the letter code C. The reason why we have chosen to take this one as part of our considerations is purely market and energy-oriented since they are now the most used and most popular battery for the type of device that we intend to develop, and also The code letter C battery has all the power specs that our device will need such as the nominal voltage, required amperage, and the right operating temperature range that will restrain our device from malfunctioning. Moreover, most of our future competitors, such as Samsung, Tile, and Apple, are currently using this kind of battery technology, and it will only make sense to go in the same direction that they are going because these batteries have proven to be efficient. The batteries that we have considered are CR2016, CR2025, CR2032, and CR2032.

3.11.1 CR2016 Battery

A CR2016 battery, in figure 3-2, is a lithium cell battery technology that is not rechargeable and that has a total length of 20mm, and this battery is 1.6mm tall. This battery has a nominal voltage of 3v and can be from a range of 70 to 100 mAh, depending on what the brand is. Also, the CR1016 battery can operate in extreme weather conditions because its operating range goes from -20°C to +70°C. Furthermore, this battery is the lightest among the four batteries that we have considered, which makes it a superb battery that can be used in small and optimized electronic gadgets. The CR2016 Lithium cell battery weight goes from 1.7 up to 2 grams. Moreover, that type of lithium battery has a life expectancy that is about 1 to 5 years, depending on many environmental aspects and storage conditions. It is also noted in table 3-15, that the CR2016 lithium cell battery is about 6 to 10 years when it is stored in the right temperature conditions, which goes from 20°C to 60°C.

Figure 3-2: Stock Photo of CR2016 Lithium battery

Description	Value
Nominal Voltage of CR2016 (V)	3V
Nominal Capacity (mAh)	80-100
Temperature Range	-20°C to +70°C
Height (mm)	1.6
Length(mm)	20
Weight (g)	1.7 - 2
Battery Type	Lithium Coin
Battery Life (Years)	1-5
Shelf Life (Years)	6-10

Table 3-15: CR2016 Lithium Battery Specifications

3.11.2 CR2025 Battery

A CR2025 battery, which is shown in figure 3-3, is a non-rechargeable lithium cell battery that has a diameter of 20mm and a total height or thickness of 2.5mm. Its nominal voltage is 3V and has a total amperage range of 150 to 175mAh, depending on which company manufactured it. The CR2025 lithium battery, as shown in table 10-2, can operate in temperatures that range from -30°C to +60°C. The CR2025 weight also depends on the manufacturer it comes from since the typical weight is 2.3 to 2.6 grams. Also, their life expectancy depends on how they are used, which conditions they faced, and what gadget is being used. They typically last from 4 to 5 years in car key fobs and calculators, but in a device such as Apple Air, Samsung Tag and Tag+, Tile, and ours, its life expectancy will be around a year. However, when not used these devices could stay in storage for about 10 years before their energy gets drained.

Figure 3-3: Stock Photo of CR2025 Lithium Battery

Description	Value
Nominal Voltage of CR2025 (V)	3V
Nominal Capacity (mAh)	150-175
Temperature Range	-30°C to +60°C
Height (mm)	2.5
Length(mm)	20
Weight (g)	2.3-2.6
Battery Type	Lithium Coin
Battery Life (Years)	1-5
Shelf Life (Years)	10

Table 3-16: CR2025 Lithium Battery Specifications

3.11.3 CR2032 Battery

A CR2032 battery, figure 3-4, just like a CR2025 battery, is also a non-rechargeable lithium coin battery that has a diameter of 20mm, but, unlike the CR2025 battery, as it is noted in table 10-3, its thickness is 3.2mm. Its nominal voltage is also 3V but has a higher amperage hour range that typically goes from 210 to 235mAh. The CR2032 battery has an operating temperature range from -20°C to +60°C, which implies that it can perform in harsh temperatures, but harsh temperature conditions will certainly diminish its life expectancy, which is about 1 to 5 years, depending on what they were used for. Just like the last battery, they are used in many small devices such as car key fobs, motherboards, tracking devices, and more. Its total weight can weigh from 3 to 3.2 grams. Even though they have a life expectancy of 5 years, in a less energy-consuming device, their shelf life is impressive since they can be stored for up to 10 years under the right conditions, which is most likely related to weather conditions.

Figure 3-4: Stock Photo of CR2032 Lithium Battery

Description	Value
Nominal Voltage of CR2032 (V)	3
Nominal Capacity (mAh)	210-235
Temperature Range	-20°C to +60°C
Height (mm)	3.2
Length(mm)	20
Weight (g)	3.0-3.2
Battery Type	Lithium Coin
Battery Life (Years)	1-5
Shelf Life (Years)	10

Table 3-17: CR2032 Lithium Battery Specifications

3.11.4 CR2032H Battery

The CR2032H battery can operate in weather conditions that are from -20°C to +80°C, and in comparison, with the CR2032, which can only operate from -20°C to +60°C, we can denote from table 10-4 below, that CR2032H battery can operate in temperature that is 1.33-fold hotter. The CR2032H battery weight is somewhat like the weight of the CR2032 battery, but it has a higher range that goes from 3 to 3.3 grams. Furthermore, the CR2032H has a larger nominal capacity range, which is beneficial, in terms of how long some devices could use them for, to better life expectancy. Even though the CR2032H battery has all these qualities and specs, they are not even close to as popular as the regular CR2032. One main reason is the price that they are sold for, which is around 10 dollars for one unit. On the other hand, the price for a pack of two CR2032 battery cells is about 3.29 dollars, which means one battery costs 1.65 dollars. Based on these numbers, the unit price of the CR2032H battery costs 6.1-fold higher than it, which explains perfectly its popularity.

Figure 3-5: Stock Photo of CR2032H Battery

Description	Value
Nominal Voltage of CR2032H	3
Nominal Capacity (mAh)	220-240
Temperature Range	-20°C to +80°C
Height (mm)	3.2
Length(mm)	20
Weight (g)	3.0-3.3
Battery Type	Lithium Coin
Battery Life (Years)	1-5
Shelf Life (Years)	10

Table 3-18: CR2032H Lithium Battery Specifications

3.12 Battery Selection

Based on the current competitive devices on the market, we realized that the most common coin battery used for tracking devices is the coin that has a 20mm diameter. There are up to three main types of 20mm diameter coin batteries that we can select from, and they are lithium-based technology. We are referring to CR2016, CR2025, and CR2032 that is divided into two categories: CR2032 and CR2032H. Since we have insights on each of these batteries and we know what their specifications are, referring to section 10, we have decided to go along with the CR2032 Duracell lithium cell battery.

The reason why we have decided to use this battery is that on how well it performs, and also because we wanted to make sure that we get the best battery possible, by implying the performance, size and price. If it is true that all of these batteries could fit in our device, their life duration will not be as same because they carry less amperage hour that a CR2032 battery, which will have a direct impact on how well the device will perform and how quickly the battery will be drawn. If drawn too quickly that will most likely upset the quality of experience that the users will expect, and that could affect how well our device could do on the market. The other battery that could replace it, since they have almost the same type of specifications, is the CR2032H, but since it is more expensive than the CR2032, it will be an economic constraint. However, the user will have the opportunity, if desired, to buy the battery that they see fit.

In table 14-1, the Duracell battery is shown, and its specifications are also noted. It is a 3V lithium coin battery with a 265 mAh capacity that has a maximum discharge and maximum pulse current rate, respectively, 0.3 and 3 mA. Its temperature goes from -20°C to +60°C and it weighs 2.92 grams. Furthermore, the battery type used

is lithium, and it is 20mm long and 3.2mm thick. Its battery life is usually about 4 to 5 years in less power-consuming gadgets, but in our device, which requires more power, it will be about a year. This battery also has a long shelf-life span since it can last up to 10 years if the right storage conditions are considered.

	CR2016	CR2025	CR2032	CR2032H
Nominal Voltage	3V	3V	3V	3V
Nominal Capacity	80mAh – 100mAh	150mAh – 175mAh	210mAh – 235mAh	220mAh – 240mAh
Temperature Range	-20°C to +70°C	-30°C to +60°C	-20°C to +80°C	-20°C to +80°C
Height	1.6mm	2.5mm	3.2mm	3.2mm
Length	20mm	20mm	20mm	20mm
Weight	1.7g – 2g	2.3g – 2.6g	3g – 3.2g	3g – 3.3g
Battery Type	Lithium Coin	Lithium Coin	Lithium Coin	Lithium Coin
Battery Life	1 to 5 years	1 to 5 years	1 to 5 years	1 to 5 years

Table 3-19: Summary of Battery Selection

- Selected Battery

3.13 3D Printing

The first documented trials of 3D printing can be traced back to the country of Japan in the early 1980s. A developer by the name of Hideo Kodama was trying to figure out a way to develop a rapid prototyping system. A layer-by-layer approach is what he came up with for manufacturing that uses a photosensitive resin that is polymerized by UV light. Kodama is often credited as being the inventor of this manufacturing system even though he was unable to file the patent requirement for the technology.

Charles Hull was an American furniture builder who developed a system for creating 3D models by curing photosensitive resin layer by layer the same year Kodama was working on his approach. In 1988 he founded the 3D Systems Corporation which released the first commercial SLA 3D printer. Also in 1988, Carl Deckard designed a system that used a laser to fuse powders instead of liquids.

Around this same time, Scott Crump patented Fused Deposition Modeling which differs from SLA because the filament is directly extruded from a heated nozzle rather than using light. This technology is the most common form of 3D printing we use today. These technologies are some of the most used for 3D printing. These 3 forms laid the groundwork for technology to advance and create some of the designs we see today.

3D printing technology has grown plenty over the last few decades. Many companies began experimenting with different additive manufacturing technologies in the 1990s. The first commercially available SLS printer was released in 2006 which created on-demand manufacturing of industrial parts. SLS uses laser technology to sinter the particles of a polymer powder to fuse them to build part one layer at a time. Around this time, CAD tools also started to become more available which would allow the development of 3D models through the use of computers. This was a crucial tool in the early stages of 3D printing. Machines around this time were more difficult to use compared to the ones we see today. These machines were also much more expensive and required a lot of post-processing.

In 2005, Open Source gave people more access to this type of technology which changed the use of 3D printing. Adrian Bowyer created an open-source initiative called the RepRapProject that would design a 3D printer that could build another 3D printer along with the printed objects. The first prosthetic leg was designed in 2008 using 3D printing which brought a lot of attention to the technology. The FDM patents that were filed in the 1980s became public information which paved the way for innovating 3D printing technology.

The price for 3D printers started to decline in the last decade which made them much more available to the general public. The process of 3D printing was also modified to make it more accessible to the public. The materials used for 3D

printing have evolved so much since the 1980s. As seen in the previous section, there are over 10 different filaments that can be used for 3D printing. Today we even see creative projects that use chocolate and pasta for their 3D printing. Each one has its features and qualities that can help you design a 3D model as desired.

Today 3D printing is used in almost any field. Industrial companies use 3D printing to create buildings. The world's largest 3D-printed building was constructed in 2019. 3D printing is also widely used in the medical field. Technology like hearing aids and prosthetic limbs like arms and legs. 3D printing has also helped with facial reconstruction prosthetics and other items to help animals. With 3D printing technology continuing to grow, it is expected that sometime soon, the technology will help design organs for humans in need. Researchers have been able to produce living tissue with the 3D printing technology available today. There have been some cases where the tissue has been safely implanted in animals. 3D printing has become a large business as well as a very important technology to help design objects that we use every day in life. As 3D printing evolves, it may someday create solutions for some of the problems we face today.

3.14 3D Printer Filament Considerations

In today’s market, 3D printing has become one of the most useful technologies. When using 3D printing, the user must also consider the filament that will be used. It is important to note that our goal is to make ranging devices while minimizing costs where we can. This includes choosing the most effective 3D printer filament for the best price.

Some of the different types of filaments are the following:

- ABS
- PETG
- Nylon
- Carbon Fiber
- PLA
- ASA
- Polypropylene
- Wood filled
- Meta filled
- PVA
- HIPS
- Flexible
- Polycarbonate

After listing out many of the possibilities for 3D printer filament, it is important to consider which 3D printer you will be using. Our group decided to use two 3D printers. We will use the Ender-3 3D printer and the Dimension Family 3D printer. The Ender-3 3D printer will be provided by one of our group members. The

Dimension Family 3D printer is provided by the University of Central Florida. The university provides a 3D printer for senior design students in the TI lab. Later in the document, we will go into the features of the 3D printers and why we decided on these two printers. Making this decision gave us a better understanding of 3D printing and which 3D filaments we should consider using.

The group will consider which filament is the best for our design based on 5 categories. The 5 categories are printability, durability, stiffness, strength, and cost. Printability determines the difficulty of printing the design when factoring in things like clogging, warping, and stringing. Clogging occurs when something is blocking the hole at the end of the nozzle which will prevent the plastic from flowing. Warping is caused by the shrinkage of the material while 3D printing. This will cause the corners of the print to lift and separate from the building plate. Stringing occurs when small strings are attached to the 3D model. This is due to the extruder moving on to a new location while plastic is still coming out of the nozzle. Strength is determined by calculating the amount of stress the material can resist before it collapses. Durability is based on how resistant the material is to different elements such as temperature and water. Stiffness is measured by how well the material reacts to bending. The cost will be measured in U.S dollars.

3.14.1 ABS

ABS is known as acrylonitrile butadiene styrene, and it is one of the most popular filaments to use for 3D printing. This filament is well known for its durability and strength. This filament has a very high resistance to impact and thus it will do better in different environments and will have a long lifetime. ABS is also cost-effective as it is one of the more affordable types of filaments. ABS is recyclable and has a very minimal impact on the environment. Although ABS has very good attributes, it does also have some weaknesses. For example, a large amount of warping is associated with using ABS. Some parts do usually end up shrinking which in turn causes inaccuracy in the 3D model. With ABS, the 3D printer can print at an extruder temperature between 225 and 245 degrees Celsius. The bed temperature that the 3D printer can print at with ABS filament is anywhere between 45 to 60 degrees Celsius. To summarize, ABS filament performs well in printability, strength, stiffness, durability, and price. Shown below is table 3-20 which lists some pricing options for ABS filament.

Spool Size	Price
220 g	$9.00
650 g	$15.00
1 kg	$21.00
2.25kg	$47.25

Table 3-20: Cost of ABS For Different Spool Size

3.14.2 PETG

PET stands for Polyethylene Terephthalate. The “G” in PETG means that it is a glycol-modified version of Polyethylene Terephthalate. This type of 3D printer filament is most used for designing liquid containers such as water bottles and soda bottles. The filament is stiff but flexible which is good for impact resistance. PETG does have a softer surface than other filaments which makes it susceptible to wear over time. When printing, PETG does not give off a smell. This filament has great thermal characteristics, allowing the plastic to cool rapidly and decrease the probability of any major warping. This material prints at an extruder temperature between 230 to 250 degrees Celsius and a bed temperature between 75 to 90 degrees Celsius. When using PETG, the 3D printer in use must have a build surface with a glue stick and painter’s tape. The price for 1 kg of PETG filament is $24 U.S dollars. In summary, PETG is another filament that performs well in printability, strength, stiffness, durability, and cost.

3.14.3 Nylon

Nylon is also known as Polyamide, and it is another one of the most popular 3D printer filaments on the market. This filament is famous for the high flexibility and strength of the material. This material is highly resistant to impact and wear. Unfortunately, the required printing temperatures for Nylon filament is usually around 250 or greater. Some 3D printers cannot print effectively at this temperature and stringing will begin to occur. One of the main issues with the Nylon material is that the filament is hydroscopic which means that it can absorb moisture well and easily from its surroundings. This means that Nylon does not do well in some environments and will require airtight storage to prevent humidity and moisture to be present in the vicinity. Overall, Nylon performs well in printability, strength, and durability. For Nylon to work well, the environment that it is working in needs meets specific conditions. Shown below is table 3-21 which represents some of the different companies in the market that currently provide Nylon filament.

Company Name	Price Per Ounce
Taulman3D	$1.75
Overture	$0.94
Polymaker	$1.70
SainSmart	$1.73

Table 3-21: Price Per Ounce for Nylon Filament

3.14.4 Carbon Fiber

Carbon Fiber has small fibers in the base material to improve the properties of the filament. carbon fiber can be infused in any other type of 3D filament. The fibers injected into the filament improve on qualities like strength and stiffness. This will cause the model to be much lighter and more stable. There is one major downfall that comes with using carbon fiber as the 3D printer filament. The fibrous component of the filament has a higher probability of clogging which will require special hardware to prevent damaging the 3D printer. carbon fiber does have some requirements for it to be useful. For example, it is important to use a hardened steel nozzle because carbon fiber can be very abrasive. Adjusting the retraction settings will help reduce the chances of clogging. Using a guided filament path can prove to be beneficial because carbon fiber tends to be more brittle compared to the base material. Carbon fiber is a good filament due to its printability, strength, and stiffness. This filament is not very durable and is some of the more expensive filaments. If this filament was to be used for our project, there will need to be a lot of modifications to ensure the result of our model. Table 11-3, shown below, lists some of the companies in the market for carbon fiber filament along with the price.

Company Name	Price Per Ounce
3DF	$0.99
SainSmart	$1.73
IEMAI	$0.79
Priline	$1.39
Polymaker	$0.99

Table 3-22: Price Per Ounce for Carbon Fiber Filament

13.14.5 PLA

PLA stands for Polylactic acid, and it is one of the most common filaments for home desktop prints. For PLA, the extruder temperature prints between 190 to 220 degrees Celsius. The bed temperature is between 45 to 60 degrees Celsius for PLA filament. The filament is known for being strong along with being easy to print since not much preparation is required. Although it's easy to print, the model does not last too long and degrades over time. Also, the texture of the model will end up being rough when the design has been printed. This filament does not perform well in high temperatures. This means that using a design that is made up of PLA would not do well outside. The PLA filament does well in printability, strength, stiffness, and price. The issue with PLA is that it is not durable and will not last long. PLA costs $9 for 220 grams and $24 for 900 grams.

13.14.6 ASA

ASA stands for acrylonitrile styrene acrylate and is similar to the ABS filament. The UV resistance of ASA increases by changing the rubber type used in the formulation. ASA is one of the filaments that are known for its resistive elements that help with impact and temperature. This filament does have a couple of unwanted qualities. ASA is one of the more expensive types of filaments. The extruder temperature required to print with ASA filament is 220 to 245 degrees Celsius. The bed temperature for the ASA filament is 90 to 110 degrees Celsius. This filament is one of the safest ones to design a model out of since it is anti-static and water-resistant. ASA also consumes a good deal of energy to be able to print. Some of the applications where the ASA filament is applied are things such as automotive exterior parts, housing components, sporting goods, and gardening equipment. According to these examples, it is evident that the ASA filament works well in an outdoor environment. When using ASA, it is recommended to use a bed adhesive such as ABS glue, Kapton tape, glue, or hairspray. It is also recommended to adjust the temperature accordingly to prevent overheating. ASA does well with printability, strength, durability, and price. One of the downfalls of using ASA as the filament is that it is not as stiff as some of the other filaments. Table 3-23, shown below, lists some of the companies that are in the market for ASA filament production.

Company Name	Price per Ounce
Amazon	$0.72
Polymaker	$0.85
Flashforge	$0.74
Overture	$0.91
Ataraxia	$0.61

Table 3-23: Price Per Ounce for ASA Filament

3.14.7 Polypropylene

Polypropylene is a semi-rigid and very light filament that is often used in spaces like storage containers. The filament is thermoplastic and is made from a combination of propylene monomers. This material has good impact resistance and does not diminish while under high-temperature conditions. Polypropylene has a good printing surface which is smooth and makes it easy to work with. The flaws of polypropylene are that the filament is not very strong, so it is susceptible to lots of warping. Polypropylene is also known to be a very expensive type of 3D printer filament. The extruder temperature that polypropylene will print at is anywhere between 85 to 120 degrees Celsius. The price for polypropylene in today's market is anywhere between 60 to 120 U.S dollars. To summarize, polypropylene only performs well in the durability category. Polypropylene does not fare well in other categories like printability, strength, stiffness, and price.

3.14.8 Wood Filled

Wood-filled filament mostly consists of 60% to 80% polymers and 20% to 40% wood fibers that are recycled of different kinds. If the wood fiber filament has a high content of wood, it will be more difficult to print than other filaments. Printing with wood-filled filament does give off a slight smell of wood which is not bad. The wood-filled filament is highly used in artistic applications. Most people like to use wood-filled filaments to design character models from cartoon shows and movies. When the wood filament is created from certain types of wood like bamboo, can result in sturdy objects that look like blocks of wood. The design will have a wooden-style look to it when it is done printing and the printer does not require a wear-resistant nozzle. When using wood-filled filament, the printer does require a larger nozzle size because small nozzles can lead to clogging. Clogging is one of the major issues that manufacturers face when using wood-filled filaments. The wood-filled filament is one of the cheapest kinds of filament with the price per kilogram usually being around 25 to 55 U.S dollars. The extruder temperature for wood-filled

filament is anywhere between 190 to 220 degrees Celsius. The bed temperature is in the range of 45 to 60 degrees Celsius for wood-filled filament. Wood-filled does well in categories like printability, stiffness, and price. The flaw of wood-filled filament is that it is not very strong or durable which means you cannot rely on the 3D model to last very long.

3.14.9 Metal Filled

Metal-filled filaments consist of metal particles that come from materials such as copper, bronze, brass, and stainless steel. Due to the material being made up of metal, it is one of the heavier types of 3D printer filaments. The pros of metal-filled filament are that it does not require a high extruder temperature and it gives a nice smooth metallic look to the 3D printed model. Some of the cons that come with metal-filled filament are things like fragile parts, clogging, and nozzle requirements. When printing with metal-filled filament, the nozzle required would have to be wear resistant. Some parts of the design when using metal-filled filament may come out hard and fragile. Clogging is a big issue that is associated with using metal-filled filaments. The metal-filled filament is one of the most expensive types of 3D printer filament. The extruder temperature required to print metal-filled filament is in the range of 190 to 220 degrees Celsius. The bed temperature required for metal-filled filament is between 45 to 60 degrees Celsius. Compared to other filaments, the metal-filled filament is only good in printability and stiffness. The metal-filled filament is expensive along with not very strong or durable. Shown below is table 3-24 which lists some of the companies in the market for metal-filled filament along with their prices.

Company Name	Price per Ounce
MIKA 3D	$0.76
AMOLEN	$0.99
BBLIFE	$0.68
GIZMO	$0.71

Table 3-24: Price Per Ounce for Metal Filled Filament

3.14.10 PVA

PVA is also known as PolyVinyl alcohol, it is an environmentally safe polymer that is highly sensitive to moisture. This means that PVA can be easily dissolved by water. PVA is created by polymerizing vinyl acetate so that it can form polyvinyl acetate. After this process is over, the material is then hydrolyzed to create the PVA filament. This type of filament is mostly used as s support material to print highly complex designs. When creating designs, manually removing the base that

is supporting the model can be extremely challenging and sometimes impossible. With PVA material, one can just simply leave the design in a water bath to dissolve the material. Since PVA filament is highly sensitive to moisture it will require storage in an airtight container. PVA also has a higher probability of clogging the nozzle in comparison to other types of filaments. This is another type of filament that is considered to be expensive. The price per kilogram for PVA filament is around 40 to 110 U.S dollars. The extruder temperature required when using PVA is between 185 to 200 degrees Celsius. The bed temperature required is in the range of 45 to 60 degrees Celsius. PVA has good strength and durability, however, it is not that good in printability, stiffness, or price.

3.14.11 HIPS

HIPS stands for High Impact Polystyrene. HIPS is another filament that is used as support material since it is dissolvable. HIPS is most commonly used with other filaments similar to ABS. The material that HIPS is made out of is a blend of polystyrene plastic and polybutadiene rubber. By mixing these two materials, the result is a filament that is both tough and flexible. Using d-Limonene to dissolve HIPS is beneficial for leaving the printed models looking clean and without any unwanted marks. Performing this procedure to dissolve the filament does require the 3D printer to have attachments such as a heated chamber. As the name suggests, HIPS is as strong as any other filament on the market. It is one of the most affordable filaments coming in at around 24 to 32 U.S dollars per kilogram. The extruder temperature required to print with this filament is between 230 to 245 degrees Celsius. The bed temperature required to print with HIPS is around 110 to 115 degrees Celsius as long as the heating chamber is used. HIPS has good attributes like stiffness, durability, and price. Printability and strength are some of the qualities that High Impact Polystyrene lacks.

3.14.12 Flexible

Flexible filaments are composed of TPE or also known as thermoplastic elastomers. Flexible filaments are commonly used for their flexibility and high-impact resistance. This filament has nice features such as flame resistance and anti-bacterial properties. The flexible filament can also eliminate electrostatic discharge and resist fading in UV light. These features make flexible filament a good choice to use for industrial applications. In today's market, flexible filament is often approved for long-term skin contact. Some examples of recent technologies that include flexible filaments are prosthetics and wearable electronics. Although flexible filaments are very beneficial, it does require a good amount of knowledge about the 3D printer being used. Printing with this type of filament requires experience and skill from the user. Someone who tries to print with flexible filament without any experience will find it difficult and could result in an undesired model. Some objects cannot be printed using flexible filaments. For example, 3D models that include joint parts work well with other filaments like PLA or ABS but need

special modification to be suitable for flexible filament. This filament is cost-effective with the price per kilogram being around 30 to 70 U.S dollars. The extruder temperature required to print with flexible filament is around 225 to 245 degrees Celsius. The bed temperature required to print with this filament is in the range of 45 to 60 degrees Celsius. The flexible filament has good durability for an affordable price, but it does not do good in the categories of printability, strength, and stiffness. Shown in the table below 3-25 are the companies that are in the market for flexible filament along with their prices.

Company Name	Price Per Ounce
DURAMIC	$0.68
ATARAXIA	$0.99
Flashforge	$0.77
SUNLU	$1.13
GIANTARM	$0.74

Table 3-25: Price Per Ounce for Flexible Filament

3.14.13 Polycarbonate

Polycarbonate also known as PC, is a strong material that is often used in very rough environments. Polycarbonate is also a good material to use due to its high resistance to heat and impact. This type of material is effective for technology like electronic cases, automobile parts, robot bumpers, and other parts that require durability. The glass transition temperature for polycarbonate is very high. Having a high glass transition temperature means that the design can stay intact while withstanding high temperatures past 150 degrees Celsius. The downside to using polycarbonate filament is that it requires high temperatures to operate. Warping is another big issue associated with using polycarbonate to 3D print. Lastly, polycarbonate absorbs moisture very easily which can create a problem with storage. The price per kilogram of polycarbonate filament is around 40 to 75 U.S dollars. The extruder temperature required for printing with polycarbonate is between 260 to 310 degrees Celsius. This is one of the highest temperatures we have seen from comparing all the filaments mentioned before. The bed temperature required to use polycarbonate is in the range of 85 to 100 degrees Celsius. Polycarbonate has good rankings in printability, strength, stiffness, durability, and costs. Shown in table 3-26 below is a list of companies that are in the market for polycarbonate along with the price.

Company Name	Price Per Ounce
RANKI	$0.85
3D BEST	$1.39
CC3D	$0.71
Gizmo Dorks	$1.55

Table 3-26: Price Per Ounce for Polycarbonate Filament

3.15 3D Printer Filament Selection

After careful consideration, our group has decided to use the ABS filament to complete our 3D model. This section will describe the many benefits that ABS has to offer and what led us to this decision. To start, ABS filament is one of the most used types of filament. This means that the success rate of getting the desired result is very high. It also means that it'll be easier to find solutions to our problems when it comes to 3D printing since so many people have experience with this type of filament. For example, if our group runs into the problem of warping, we know there are many places we can look for help. When looking for resources on how to deal with problems when 3D printing with ABS, it'll be easy since most people share their experiences about working with the filament.

Our group listed out categories that need to be considered to decide on which filament will be used for our project. These categories are printability, strength, durability, stiffness, and price. Besides warping, ABS flows well through the 3D printer and has a nice result. ABS is made from strong material which makes it highly resistant to impact. Strong material will give you more opportunities to design things that are a little more complex since the material can be trusted to not break. Stiffness is another attribute that ABS offers. Stiffness will add to strength which means your design will react well in different environments and conditions. Since ABS reacts well to different environments means that it is very durable and can have a long lifetime. The price for ABS is some of the most affordable compared to other types of filaments. Most companies like Flashforge and Polymaker market their ABS filament at around $0.62 U.S dollars per ounce. According to the table, 650 grams of ABS filament is $15 U.S dollars. This price is affordable for our group since the 15 can be divided between the four of us.

To summarize, these are the factors that our group considered when deciding which type of material, we should use for 3D printing. Even though there are other types of filaments that are better in some categories, ABS has a nice balance between each category. ABS is not the strongest material or the easiest to print with, but it is not the worst. Choosing ABS as our 3D printer filament is also a safe decision as it is one of the most used filaments in 3D printing.

3.16 Coin Cell Battery Holder Selection

While there are many coin battery holders to consider, we have opted for a few that have key aspects we will need for this project's PCB. We want a battery holder that will have an ultra-low-profile mount while retaining access to the user in case of battery replacement. Our group has set engineering requirements for our device to have a very small footprint compared to products on the market currently or smaller than them. To achieve this, we plan to use surface mount holders that sit partially through a hole in the PCB. Making use of this design we help us achieve a very slim electronic device that will have a compact height. Figure 3-6 shows a simplistic view of the through-hole PCB mount for the battery cell holder.

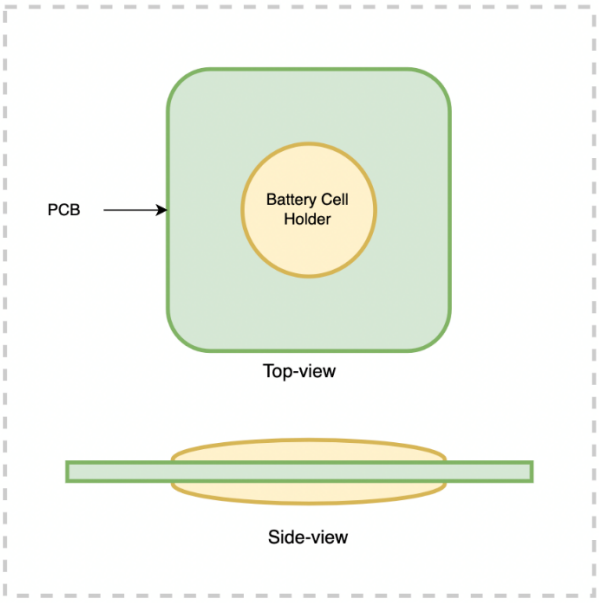


Figure 3-6: Through Hole PCB Mount View

3.16.1 Keystone Electronics 1057TR

The Keystone 1057TR is an ideal candidate for our project because it allows for an ultra-low-profile design. The 1057TR battery cell holder has a PCB and surface mounting type. It also has an SMD (SMT) tab termination style and is made for the 2032 series of batteries making it a perfect fit for 20mm coin batteries. With the sit-through PCB mounting style Keystone is estimated that the holder will only rise 2mm from the PCB board. Besides the compact design of the Keystone 1057TR battery cell holder, this part has plentiful stock available to purchase at a reasonable price. We are also able to find a datasheet regarding the dimensions of not only the part but the cutout hole needed to fit it in our PCB design. Websites like Ultra librarian and SnapEDA have symbols, footprints, and 3D models obtainable for our project too. Figure 3-7 shows what the part will look like.

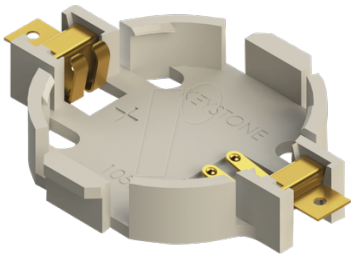


Figure 3-7: Stock Photo of Keystone 1057

3.16.2 Memory Protection Devices BU2032SM-JJ-MINI-GTR

Our second choice for a coin cell battery holder will be from Memory Protection Devices. This coin cell battery holder is the same design as the Keystone 1057TR. The Keystone 1057TR is the equivalent design to the MPD BU2032SM-JJ-MINI-GTR. Our group has considered this part too because of increased purchasing stock and part information. Some materials not available on Ultra librarian and SnapEDA can be found for the MPD component. Which will be convenient for our PCB design if we run into issues that require these data. Unlike the Keystone 1057TR we could find specifications about the coin cell battery holder not found in the equal design. Such as shock testing, vibration testing, and drop testing. The BU2032SM-JJ-MINI-GTR can be dropped from 1 meter and a 2032 battery won't dislodge. Summary of these coin holders can be found in table 3-27.

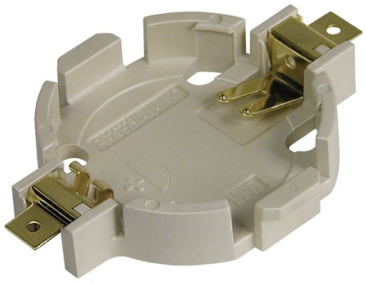


Figure 3-8: Stock Photo of MPD BU2032SM-JJ-MINI-GTR

	Keystone 1057TR	MPD BU2032SM-JJ-MINI-GTR
Price	$2.12	$1.36
Stock (Nov. 2022)	3,009	15,984

Table 3-27: Summary of Coin Cell Battery Holders

- Selected Battery Holder

4. Standards, Certifications, & Constraints

This section will cover standards, certifications, and constraints related to our project. We will also look at relevant device certifications in the scope of this project and consortiums related to ultra-wideband technology.

4.1 IEEE 802.15.4 Ultra-Wideband

IEEE 802.15.4 is a standard that was approved on February 6, 2012. IEEE 802.15.4z is an amendment that was put forth in 2019 and was approved on June 2, 2020, by the IEEE SA Standards board. IEEE 802.15.4 focuses on radio frequency, and physical and MAC layers. IEEE 802.15.4z emphasizes the Ultra-Wideband physical layers. The IEEE 802.15.4z amendment also provides coding options to increase the range measurements. This section will focus on some specifications that are emphasized in IEEE 802.15.4 standard and IEEE 802.15.4z amendment.

The role of physical layers is to transmit data through network nodes. Ultra-Wideband technology has two physical layers called high-rate pulse frequency (HRP) and low-rate pulse (LRP). The low-rate pulse physical layer sends pulses at a lower repetition rate than the high-rate pulse physical layer. The purpose of lower repetition pulses is to allow high instantaneous peak power. This means that the high-rate pulse layer sends more pulses, but they are weaker than the low-rate pulse layer. The high-rate pulse mode was the first to be introduced and therefore is the more widely used out of the two for UWB devices.

There are conditions that need to be met in order for two UWB radio chips to be compatible. The most important condition is that both devices need to have the same center frequency. Without this condition being met, the reception would not be possible. The IEEE 802.15.4 standard introduces 16 channels, and each channel has its own center frequency and maximum bandwidth. Most channels have the same bandwidth of 499.2 MHz which is the minimum bandwidth used. Some channels use the same center frequency but have different bandwidths. An example of this would be channels 2 and 4 which have a center frequency of 3993.6 MHz. Some UWB radio chips require specific channels to function.

IEEE 802.15.4 also emphasizes the requirements for the pulse shape in the high-rate pulse mode for UWB. The pulse shape that is transmitted will be restricted by two functions. The two functions are the cross-correlation function and the standard reference pulse function. The reference pulse function used is a root-raised-cosine pulse that has a roll-off factor of 0.5. The pulse being transmitted will need to have a magnitude of the cross-correlation function where the main lobe is equal to 0.8 or greater for at least a time equal to TW. TW is defined as the main lobe width time of the pulse. Another requirement that IEEE 802.15.4 focuses on is the time domain mask for the pulse. In order to follow the regulations mentioned in the standard, the pulse cannot surpass the set bounds. Meeting this requirement will help with compatibility for other UWB radio chips. Pulses can still be different

whilst meeting the requirements of this standard. With the possibility of different timing for pulses,

In the IEEE 802.15.4, The frame structure of the UWB high-rate pulse mode includes 4 different fields. The IEEE 802.15.4z amendment offers four frames that are included in five different fields. The SYNC field works on synchronizing the sender and the receiver. In this process, the receiver will detect the preamble and align the sender with the preamble. The SFD field will mark the end of the preamble as well as the start of the switch to the PHR. The SFD field is also critical for timestamping which makes it important for ranging performance. The SFD fields must be the same in each UWB radio ship in order for them to communicate. If the SFD field is not configured correctly in both UWB radio chips, both the sender and receiver cannot properly determine the time of arrival. The PHR field is responsible for providing information about the payload that will be received by the receiver. The IEEE 802.15.4z amendment provides information that allows more data to be transmitted in one packet. The PHR field can be modified so that the PHY payload length field may increase up to 12 bits.

The high-rate pulse mode is most likely the mode we will be focused on since low-rate pulse mode is only compatible when the base mode is used. Since high-rate pulse mode is the most used mode it will be less difficult to apply for this project.

4.2 IEEE 802.15.1 Bluetooth

The IEEE 802.15.1 standard was approved on April 15th, 2002, by the IEEE-SA Standards Board. A couple of months on July 26, 2002, it was approved by the American National Standards Institute. This standard focuses on the medium access control and physical layers of Bluetooth technology.

In the seven-layer OSI model, the physical layer is the first layer. The physical layer oversees the transfer of bits between systems in the nearby area through the air. This layer is bound to two functions. The physical layer can receive a bitstream from the media access control sublayer and transmit the bitstream through radio waves to an associated station. Also, the physical layer can receive radio waves from an associated station and convert the waves to a bitstream that is transferred to the media access control sublayer.

The Bluetooth transceiver runs in the 2.4 GHz band. In most countries, the specific range of this frequency band is 2400 MHz to 2483.5 MHz. The requirement for the Bluetooth transceiver is designed to provide compatibility between radios used in the system and to define the quality of the system.

The requirements for the transmitter characteristics are given as power levels at the antenna connector. For systems that have an integral antenna, it is preferred that they provide a temporary antenna connector during type approval. There are three power classes that the equipment is divided into. The power class 1 equipment requires power control. Power control will limit the transmitted power

to over 0 dBm. The power steps will form a monotonic sequence with the maximum step size being 8dB and the minimum being 2dB. Class 1 equipment is able to control its transmit power to 4dBm or less with a maximum transmit power of +20dBm. If the receiving side of a connection does not support the messaging for power control of the sending side, then power class 1 will not be able to send packets from one device to another.

Another problem is when a class 1 device is paging another device in the nearby area and has a larger input power than the stated requirement. This could cause the device that is listening to not respond. A device that is performing a page should use power class 2 or power class 3.

The next requirements that are talked about in the IEEE 802.15.1 standard are modulation characteristics. The modulation used in Bluetooth is Gaussian frequency shift keying with a bandwidth time of 0.5. The modulation index must be between 0.28 and 0.35. To represent a positive frequency deviation, a binary 1 will be used. The opposite will occur to represent a negative frequency deviation. The symbol timing will be better than 20 ppm. For each transmit channel, the minimum frequency deviation that corresponds to the 1010 sequence can be no smaller than ±80% of the frequency deviation that corresponds to a 00001111 sequence.

Moreover, 115kHz is the minimum deviation allowed. The transmitted data will have a symbol rate of 1Msymbol/s. The time difference between the ideal symbol period and the measured crossing time is considered the zero-crossing error. The zero-crossing error must be less than ±0.125 of a symbol period. The maximum range for frequency deviation is 140 kHz to 175 kHz.

There are specifications for radio frequency so that the transmitter is able to send data. The initial center frequency accuracy of the transmitter must be ±75 kHz maximum from the center frequency. The frequency accuracy before any data is transmitted is considered to be the initial frequency accuracy. The ±75 kHz does not include the frequency drift requirement. Table 4-1, shown below, includes the different types of packets along with the frequency drifts associated with the packet.

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

Table 4-1: Amount of Frequency Drift in A Packet

4.3 FiRa Certification

FiRa is a company that specializes in UWB technology. The name FiRa stands for fine ranging. This company highlights UWB's ability to accurately measure the distance to a target and determine position. FiRa certifies devices by testing the equipment and making sure it meets their specifications. The FiRa Compliance & Certification Working Group completes test specifications to ensure that the device complies with the FiRa MAC and PHY technical requirements. After the device passes conformance testing, there will be some fees applied for different devices. Table 4-2, shown below, lists the fees.

Type of Device	Fee
Initial Tested Device	$2,500
Rebranded Device	$1,000
Variant Device	$1,000
Device Using a Certified Module	$1,000

Table 4-2: Fee Associated with Device Type

4.4 MFi Certification

MFi offers technology that can be used with devices that companies will develop. MFi provides access to technical specifications and resources needed to create accessories that communicate with Apple devices using technology Mfi technology. Some of the technologies provided by MFi are Find My Network, CarPlay, HomeKit, and AirPlay audio. This would be beneficial to our project since the Find My network would do a good job of determining the position of our device. The one downfall is that after reaching out to MFi, we were informed that we would

only be allowed to use this technology if we were going to get our device manufactured.

4.5 Software Constraints

Software planning is the most complex objective for an engineer. It takes a huge amount of time, money, and resources. As engineers, we need to consider this and be efficient in our software planning. With proper planning and consideration, we can ensure that our project would be delivered on time. When planning for our projects it is important to consider three major constraints, time, money, and scope.

For our project, we plan on using Apples operating system for our user's face. With Apple, there is the Find My app that integrates all Apple-compatible devices into one user-friendly app. Apple regularly updates its software, and a completely new version once every year. The newest version as of writing this paper is iOS 16. With iOS 16 Apple included a nearby interaction accessory protocol. The nearby interaction accessory protocol facilitates configuring an apple accessory to an Apple device. For example, the Apple AirTag with an iPhone. We must implement this protocol in our design, so it is a requirement for our iOS version to be iOS 16 or higher.

Apple is very strict with its developers, meaning to get a development license one must go through a lot of hoops. To program an Apple app the developer must use an Apple device such as a MacBook. With this requirement, we will use a MacBook pro to program our app. As of today, the newest version available for the MacBook Pro is the Ventura. Also, for app development, Apple only supports a small selection of programming languages, one of which is swift. This will be the programing language we are required to use for our project’s app development. For our hardware, we will be using a microcontroller as the heart of our device. A microcontroller uses C/C++ as its programing language. This is standard and should cause any issues with our embedded design.

4.6 Smartphone Constraints

Being that the project will incorporate Ultra-Wideband technology, there will be some constraints on which smartphone we will use to find the device we are interested in. The phone we will use must have an Ultra-Wideband chip so that the smartphone may find the desired device.

There are currently 4 companies in the market that sell smartphones which include Ultra-Wideband technology. The four companies are Apple, Google, Samsung, and Xiaomi. Xiaomi is the only company out of the four which only has one smartphone on the market which is called MIX4. Each of the other three companies offers at least two phones that have Ultra-Wideband technology. Google offers the Pixel 6 and the Pixel 7, but the phones must be the pro model only. The UWB controller that the Pixel 6 and Pixel 7 have is the Qorvo DW3720. Out of the four companies that offer these smartphones, Samsung offers the most. Samsung offers a total of

6 smartphones that contain Ultra-Wideband technology. The Galaxy Note20, Galaxy S21, and Galaxy S22 must be the Plus or Ultra models only. There is also the Galaxy Z Fold2, Galaxy Z Fold3, and Galaxy Z Fold4. The NXP SR100T is the UWB controller that the phones from Samsung use. In September 2019, Apple was the first company that offered Ultra-Wideband technology in smartphones. Apple offers iPhone 11, 12, 13, and 14 with Ultra-Wideband technology. The UWB controller embedded in the Apple iPhone is the Apple U1 chip. Our group will choose to work with the Apple iPhone 13 since the group is the most comfortable and familiar with Apple technology. Also, Apple has worked with UWB technology the longest so it should be the most reliable.

4.7 Size Constraints

The desired function of this device is the capability to tag most if not all objects to which it is attached. One issue that will need to be addressed is the size of the finder device. Tracking large objects such as motor vehicles is not too much of a challenge since some of the largest tracking devices are small enough to be handheld. The issue arises when the user would like to track objects that are handhelds such as phones or keys. The device should be designed in such a way that the user should not experience a change in the weight of the object that is being tracked. The tracking device should not be easily visible and should be small enough to blend in with most objects. Overall, the device should be compact and hardly noticeable when attached to objects.

Within the last few decades, technology has advanced to where most electronic devices are more powerful but also smaller in size. According to Moore's Law, the number of transistors that can be included in a chip will double every two years. Due to the increasing number of transistors included in chips, the chips became more powerful. In recent years, companies have managed to design transistors to be in the nanometer range. This type of modern technology will help design a tracking device that can be powerful and easily sit on the palm of a human hand as well. The one downfall of the device is so small is that it can be easily misplaced or lost. In those unfortunate circumstances, the device can be simply tracked and recovered by using an app that we will develop. This of course will be the primary function of the device.

4.8 Environmental Constraints

Constraints in the engineering field are any limitations that a design or project could have on various aspects such as social, legal, governmental, cultural, technical requirements, and more. In today's society, environmental constraints are among the most predominant requirements that any company needs to follow to accomplish and launch a project because it is important to design products that would not be a liability to its consumers and also that will not cause harm to the environment. There are many environmental constraints to be considered in the realization of any kind of project, such as making sure that the product will not

pollute the atmosphere, that it will be recyclable, that it will not use hazardous materials, that it will be energy efficient, and that it will be in compliance with the codes.

Pollution has been for years the number one issue that the world is currently facing because it contributes to climate change and limits resources. That is why engineers, lawmakers, politicians, and scientists around the world are putting their efforts together, so they could fight this problem and aim for the use of cleaner and better energy that will replace some of the current ones that harm the environment, such as coal and fossil fuels. Our project is intended to be eco-friendly and will not constitute a problem for the environment. It will not emit any type of greenhouse gases such as carbon dioxide, methane, or nitrous oxide because it will be DC-powered.

Furthermore, our device will be designed to be reusable once the battery has been drained. As mentioned in the lines above, the device will be DC powered by using a coin battery. The device will also be designed to use low power mode when no actions have been taken, so its battery life lasts longer and so the users do not have to repeatedly buy batteries, which would harm their pockets and especially the enrolment since that will create more waste. To make contribute to climate change and to limit the number of items that are being thrown out every day in nature, our group has intended to design our casing to be made from plastic material, and most of the components that will be used in the realization of the distance tracking device will be recyclable.

4.9 Economic Constraints

This design project will run into a few economic constraints. The costs that this project will demand will be covered by each member of the group. This means that the project costs will be constrained by the budget of each team member. Since the goal of this project is to create a small device that measures range, the costs of the individual parts needed should not be expensive. The total cost should be able to be divided evenly between group members since this is a small-scale project.

Monitoring the market for the prices of similar products is another economic constraint that is important to address. The main priority for our group is to create a range-finding device that works as it was intended to, but if the opportunity arises, it would be a good decision to put it on the market. We need to make sure that the device works well enough that we can compete with other vendors already in the industry. This means we need to study the market and see where the price should be set for our device. Again, we are not trying to create a product for the market, but we will keep this constraint in mind just in case this is a possibility. The range-finding device will need to be user-friendly if it makes it onto the market. There will probably need to be a slight adjustment to the design so that the device will be easier to use. There also needs to be some sort of maintenance service for when

the device is not working properly. We also need to monitor the market for the pieces needed for this device so that they may be replaced. We need to make sure that the pieces are available to the consumer when they need replacing.

4.10 Ethical Constraints

Ethical constraint plays a significant role in any project there could be because it makes sure that one’s work is authentic, does not offend anyone by any means, respects other people’s intellectual property, is not biased towards a certain group, is inclusive to all, and is target audience is being identified, has all the permissions needed, is legal, and is not subject to plagiarism. In sum, ethical constraints ensure that every individual, company, manufacturer, or enterprise, when realizing their project, works within norms that have been long established and accepted in society, so their work is not offensive to a certain group of people.

Ethical constraints are the type of constraints that we take seriously because we want to ensure that anyone that will use our product will feel comfortable while doing so. It is our mission to keep users at ease and to make sure they enjoy the product as much as possible and for what is what to do, which is to keep track of objects that users see fit. Our UWB tag should be user-friendly and easy to program. However, since our tag is designed to be small and weigh less or equal to 20 grams, its users must ensure they keep it away from children, their ages are typically ranged from 0 to 9 years old because there is a possibility that they could swallow the device and harm themselves. If it is true that we own the brand and any intellectual property related to our work, we are not responsible for matters that happen outside of our project development scope, which are issues related to its functioning, malfunction, or defect. That is why, per our ethical constraints, it is our responsibility to inform the users that will buy and use our product about how they should use the product, any safety concerns they should know about and follow, and what steps they should take to address a concern.

Every technology has its pros and cons, and, as a company, manufacturers, or inventors, we do not have much control over how people will use a specific product. Most people, when they buy a product, will use it as intended, but there are always some mal-intentioned ones who like to use it in their wrongdoing. For instance, thieves are using tracking devices like Samsung SmartTag or Apple AirTag to steal other people's belongings, and this is a big concern for most tracking device users. Even though those companies are not responsible for these people's actions, they sure have ethical obligations to internally address these issues and build new software that will aid to overcome, restrain, or prevent these bad actions from happening. Furthermore, it will surely help to protect the users against inconveniences such as kidnapping, illegal tracking, and stealing.

5. Hardware & Software Design

This section will involve all the details of hardware and software of our ultra-wideband tag device. For hardware we will consider what tools are needed to accomplish our engineering requirements, how we plan to design the components on the PCB in a software we have chosen. Mostly importantly schematics for our design and selected components.

As for software in this section we plan to cover similar topics such as development programs needed to accomplish the features. We will also look at very important frameworks, software development tools, and APIs that will be used in the project. There will be an overview of how we plan to use the Apple iPhone mobile application to operate our ultrawideband tag finder.

5.1 Hardware Development Tools

For this project we had many uses of various development tools. In this sub-section we will cover these tools we used for the hardware portion of development.

5.1.1 - Qorvo DWM3001CDK

The Qorvo DWM30001CDK is a fully integrated (UWB) module based on the Qorvo DW3110IC. It has a micro-USB port for direct contact which is also connected to a J-Link that can be used for debugging. The module has two antennas one for Bluetooth 5.2 and one for UWB. The UWB RF bands 5 (6.5GHz) and 9 (8GHz) are supported by the antenna.

Using this kit will allow us to build a real time location system using UWB as a Two-Way Ranging (TWR) or Time Difference of Arrival (TDoA) tag. The UWB technology on the chip allows us to reach precision within 10 cm, which will achieve our goal of 20cm or less. This module is also interoperable with Apples Nearby interaction beta evaluation software. With a U1 equipped iPhone we will be able to use Apples features based on location, distance, and direction using the module. This module is a perfect solution by integrating UWB that can use Apples Nearby interaction as well as a low power consumption.

Below in figure 5-1 shows a whole schematic which includes the DWM3001 microcontroller with its pin outs. Testing points are included in the MCU to help aid down potential problems in our project.

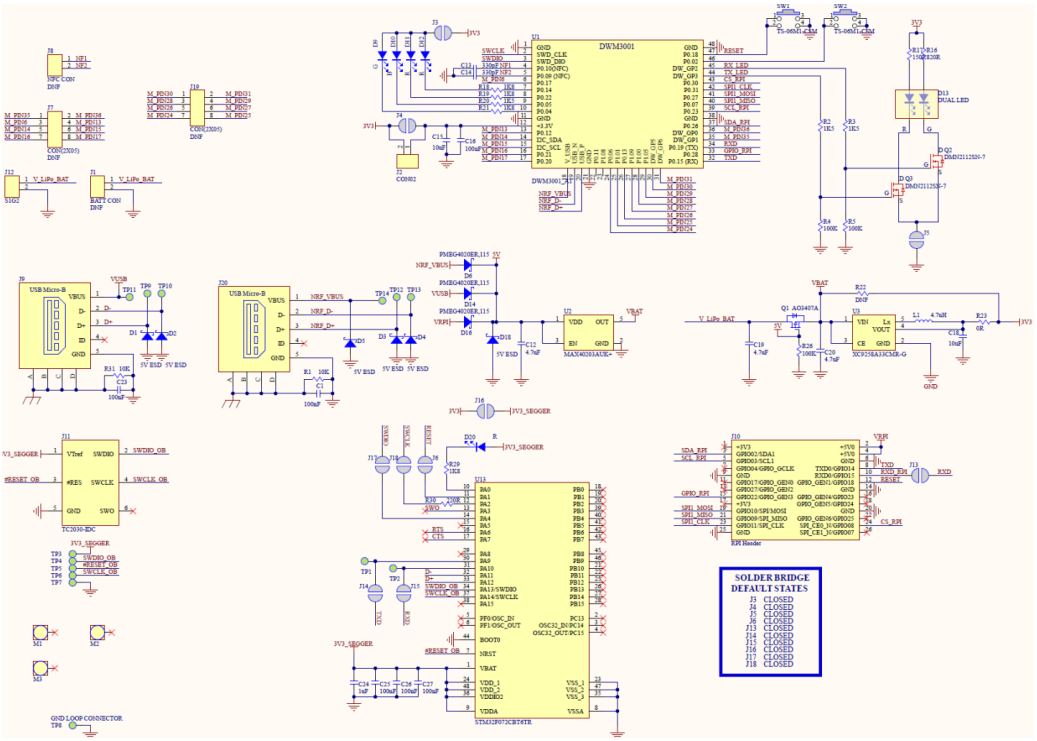


Figure 5-1: Schematic of DWM3001CDK

Usage Approved by Qorvo (see appendix)

The DWM3001 has a total of 48 pins, pin functionality is still under research. For debugging the DWM3001 uses a J-Link OB. It functions as a USB to a serial wire debug. This allows debugging and programing to the module.

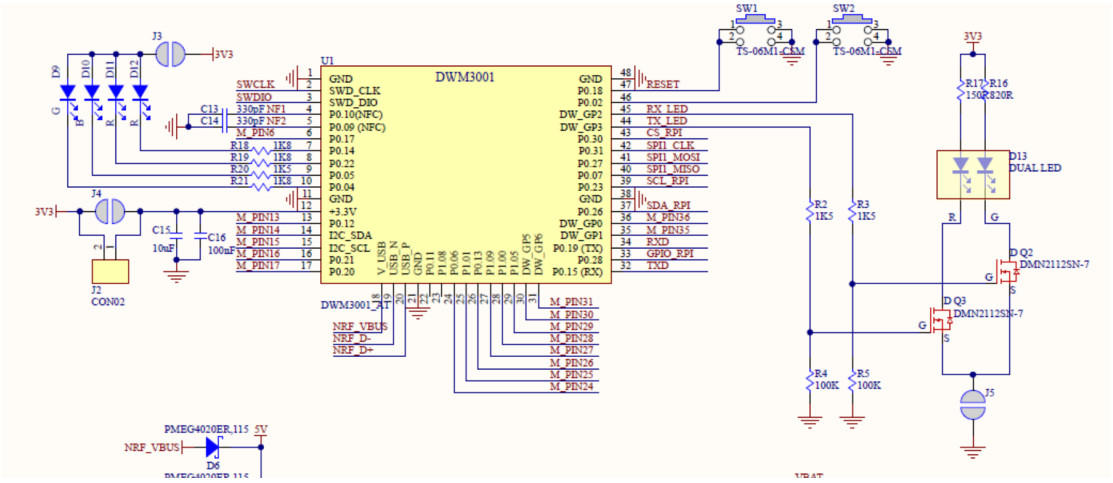


Figure 5-2: Schematic of DWM3001 MCU

Usage Approved by Qorvo (see appendix)

The Micro USB has five pins in which the data and power is transferred. Pin 5 is used as a ground and pin 1 is used as a VCC. Pin 4 is needed to indicate which mode the USB is in, either power or data transfer mode. Pin 2 and pin 3 are used to transfer the data.

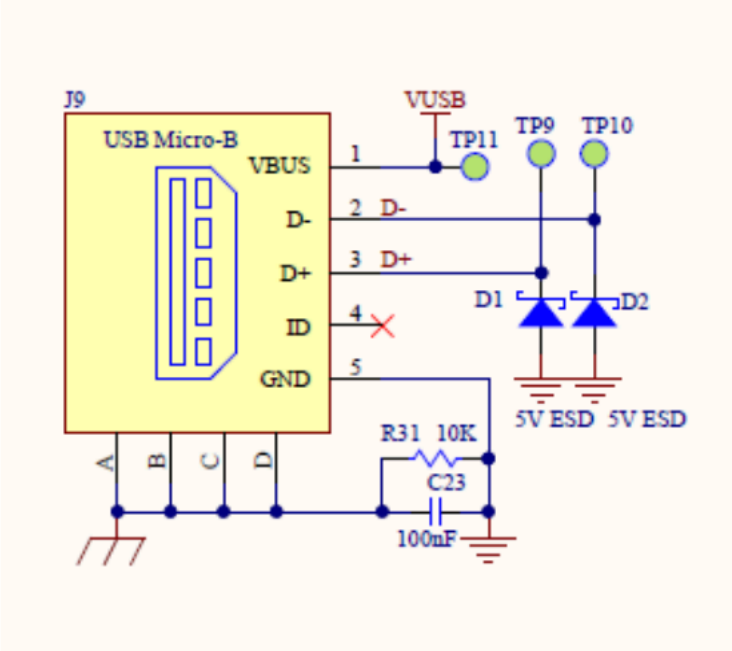


Figure 5-3: Schematic of Micro USB

Usage Approved by Qorvo (see appendix)

5.1.2 - Nordic nRF52833 DK

The Nordic nRF52833 DK is a development kit that we would use for the development of Bluetooth low energy as well as NFC technology. The board has a 2.4GHz antenna for Bluetooth as well as multiple NFC antennas that communicate using the on-chip NFC-A tag. It includes a on board SEGGER J-Link for debugging that can be communicated via micro-USB. The Nordic nRF52833 DK has a wide range of ambient temperate ranging from minus 40 degrees to 105 degrees Celsius making it a suitable component for a device that could possibly be left in a car on a hot summer day. Some features of the Nordic nRF52833 DK include:

- 4 LEDs and 4 buttons with user programmability
- 2.4 GHz and NFC antennas
- SWF RF connector for direct RF measurements
- On-board SEGGER J-Link for debugging or programming
- Arduino Uno Rev3 form factor capability
- Dedicated Pins for measuring power consumption
- 1.7-5.0 V supply from USB, with the capability of using a CR2032 coin cell battery

Below in figure 5-4 is a schematic of the nRF52833 controller as well as the pin layouts

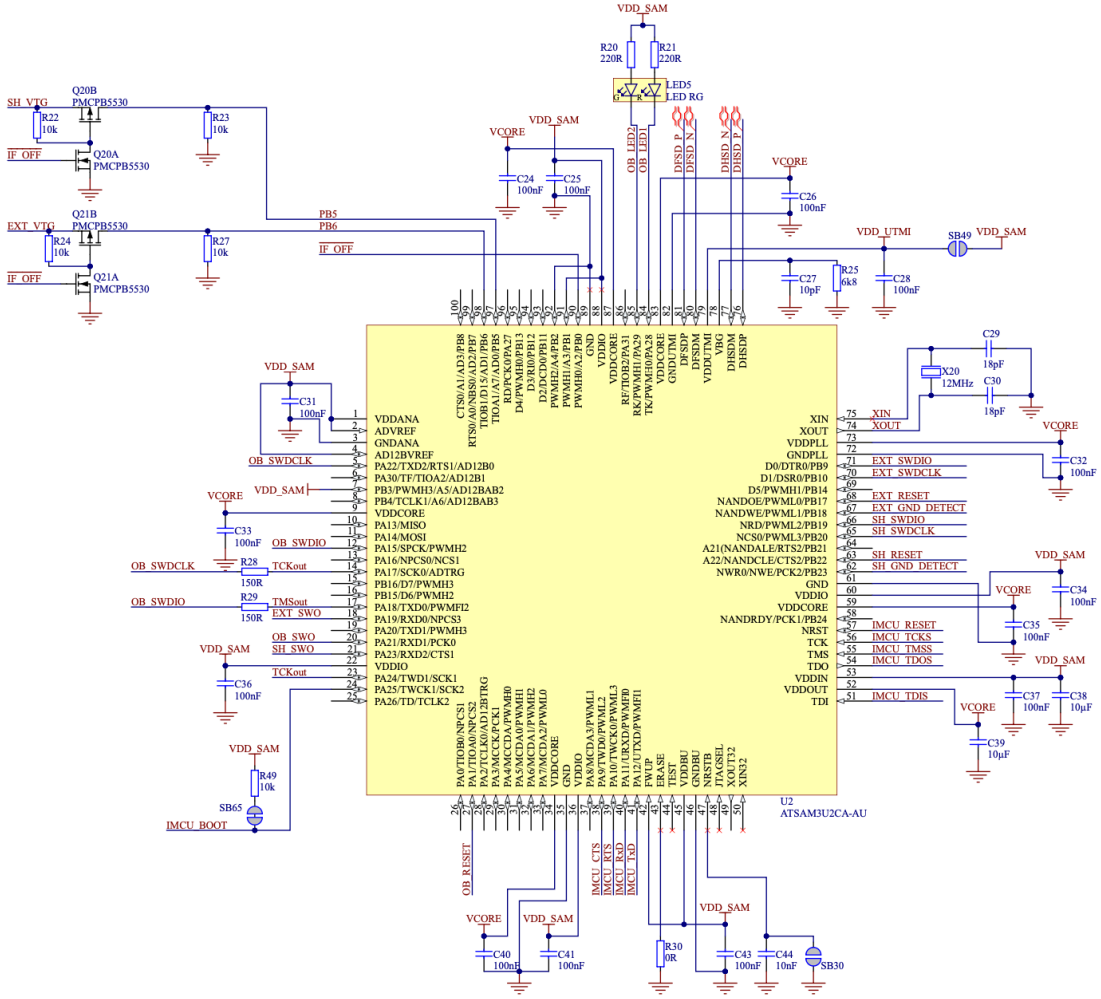


Figure 5-4: Schematic of nRF52833

Usage Approved by Qorvo (see appendix)

This buck converter is used to convert the high input voltage to the desired low output voltage. In this buck converter there are 2 resistors, 3 capacitors, 1 inductor, and 1 diode, totaling in 7 components.

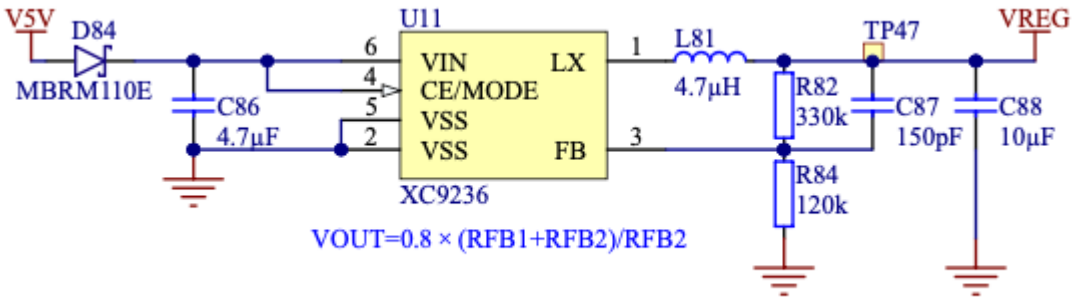


Figure 5-5: Buck Regulator for VDD

Usage Approved by Qorvo (see appendix)

This boost regulator is used to regulate the input current that is produced with the higher voltage. In this boost converter there are 2 resistors, 5 capacitors, and 1 inductor totaling in 8 components.

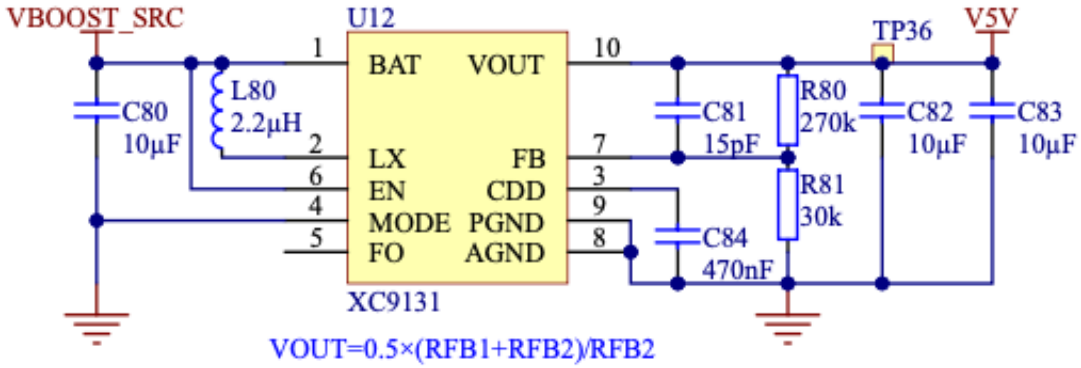


Figure 5-6: 5V Boost Regulator

Usage Approved by Qorvo (see appendix)

5.1.3 - Qorvo DWM3000EVB

Qorvo DWM3000EVB is part of the Qorvo DW3000 family of products, launched in 2021, that is interoperable with the Apple U1 chip that is used in many Apple watches, tablets, and cellphones. Qorvo DW3000 is considered as the company next-gen chipset family equipped with the latest Ultra-wideband technology that will allow developers to have a better app-based experience while using them since the new chipset is compatible with any Apple devices equipped with U1 chip. Some of the experiences that will be enhance, both for the developer and user, is directly related to the direction, location, and direction because the upgrades that have been made throughout the entire chipset. It must be noted that the new chipset, DW3000, is in compliance with the IEEE 802.15.4-2015 and IEEE 802.15.4z and in accordance with the specifications of FiRa consortium PHY and MAC.

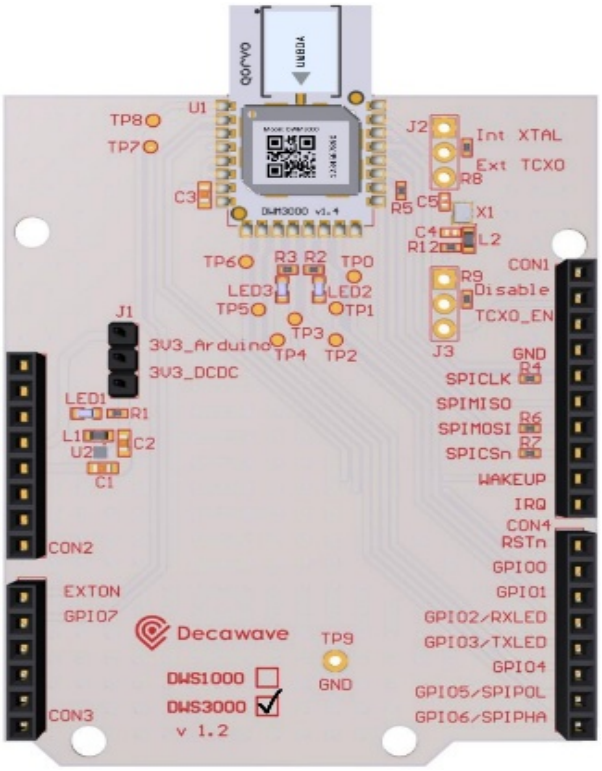


Figure 5-7: Qorvo Arduino Compatible Shield

Usage Approved by Qorvo (see appendix)

Qorvo DWM3000EVB that was formerly known as DWS3000, where EVB stands for evaluation board, is basically based on Qorvo DWM3000 module. It is commonly named Arduino Shield module kit and is compatible with every DW3000 2nd generation chipset. As it can be seen in figure 5-7, the DWM3000 chip pins are soldered to the evaluation board and is fully linked to it. The main goal of this board is to accurately capture location data and measure distance by using the UWB signal emitted by the chip which is within a certain scale. This type of operation is often called RTLS that stands for real time location systems.

The DWM3000EVB evaluation Board kit, as it can be seen in table 5-1, is completely interoperable with Apple U1 chip that is compatible with Apple devices equipped with UWB technology was described in section 4.6. This board also support UWB channels 5 and 9 that range, respectively from 6.5 GHz to 8GHz. Also, it is compatible with Arduino Shield form factor and its module pins are accessible via Arduino connectors which are as seen in figure 5-7. The evaluation board also features a DC-DC convertor that converts and stabilizes the voltage. Also, this board is compliant to the standards and specifications that were mentioned earlier.

	Key Features
Qorvo **DWM3000EVB**	Interoperable with Apple U1 chip
	UWB channels 5(6.5 GHz) and 9(8GHz)
	On-board 3V3 DC-DC convertor
	Compatible with Arduino Shield form factor
	Module Pins accessible via Arduino connectors
	Jumper allows selection of the power supply and current consumption measurements
	Incorporates the DWM3000 module containing DW3110 UWB IC and ceramic UWB antenna

Table 5-1: Key Features of Qorvo DWM3000EVB

5.1.4 - Nordic Power Profiler II

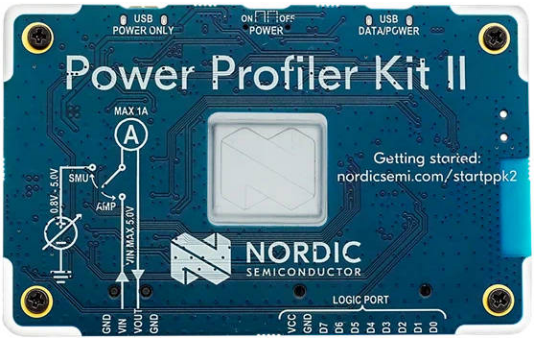


Figure 5-8: Stock Photo of Nordic Profiler II

The Power Profiler Kit II, as shown in figure 5-8 below, is a power debugging of embedded applications and is used for current measurement and power consumption optimization. This kit was launched in the year 2020, more precisely in December, and has been a highly competitive product ever since it was launched. This kit is mainly help developers to understand how their protypes work, and it also help them to know how much power that has been drawn by each and every part of their circuit.

By knowing all this data, developers could easily choose to validate and optimize the circuit by making the appropriate changes such as in deciding if the current

battery in the prototypes needs to get upgraded, validated, or decreased, also it will help determine if more RLC components will be needed, replaced, or removed. We have considered using this semiconductor because it will allow us to get accurate measurements of the current and power consumption of our protype. Furthermore, its price is outmatched compared to the ones in the market that has the same features. For example, in table 5-2, the prices, range, and Brand of each device is shown, and based on the prices, the Power Profiler Kit II makes more sense for us to get because of the economic constraints, battery selection, and PCB design. Even though the uCurrent Gold seems to be the least expensive, it is not because it is a product located in Australia, and it will be a lot more expensive when it is ordered. It can also be seen that beside Nordic Power Profiler II, uCurrent Gold, and Battlab that stay below 100 us dollars, the Otii and Joulescope are the most expensive ones, and they cost, respectively, $699 and $799.

Device name	Cost	Voltage Range	Current Range
Nordic Power Profiler II	$89	0.8 – 5V	0-1A
uCurrent Gold (AUS)	$59.22 (+ shipping fee)	2.7 – 5.5V	0 – 1.25A
BattLab	$99	1.2 – 4.5V	0 – 450mA
Otii	$699	0.55 – 4.55V	0 – 3A
Joulescope	$799	-1 – 15V	-1 – 3A

Table 5-2: DC Power Profiling Devices

Using the Nordic Power Profiler II, provides many benefits because of the key features that it possesses. Based on the table 5-3, we can determine that it can accurately measure current that ranges from 200nA to 1A. Moreover, it can operate in two modes, ampere meter and source modes. When it is in ampere meter mode, it serves as an ammeter and measure the current drawn throughout the system and does not energize the module to be tested, which implies that the module will require an external source or power supply to be powered to. On the other hand, unlike the ampere meter mode, the source meter mode energizes the module and works like any power supply, but under the range that is stated in the table. In this mode it is capable of supplying a maximum of 1A current and has voltage that varies from 0.8 to 5V DC. When it is in this mode, it measures the power consumption that exist throughout the entire module.

The kit is also able to support all of the Nordic DKs, which are the different development kit that they have such as nRF52 DK, nRF52840 DK, nRF9160 DK, nRF52833 DK, and more. The Power Profiler II is also faster than the previous generation that they had. It is 10-fold quicker sampling compared to the last one. Furthermore, the kit can connect to the any computer that support the Power Profiler app in nRF connect. Also, it has eight digital inputs that enable low-end logic analyzer support. For all these reasons, that is why the Nordic Power Profiler kit II is considered as a standalone unit because it can be a power source, an ammeter, and a device that measures power.

	Key Features
Nordic Power Profiler II	200nA to 1A measurement range
	Support for all Nordic DKs, in addition to custom HW
	Ampere meter and source modes
	10x quicker sampling (100ksps) versus previous generation
	Standalone unit
	0.8 – 5V VCC output range and up to 1A current supply
	8 digital inputs enable low-end logic analyzer support
	Power Profiler app in nRF Connect for Desktop

Table 5-3: Key Features of Nordic Power Profiler II

5.1.5 - Segger J-Link

SEGGER J-Link is the most widely used debugger probe that's been improves throughout the years. It is USB powered and is supporting many CPU cores. It can communicate at high speeds because it is based on a 32-bit RISC CPU. Shown below are the specifications of the different J-Links available.

Feature	J-Link EDU	J-Link EDU Mini	J-Link BASE Classic	J-Link PLUS Classic	J-Link PRO
Download speed into RAM	1.0 MB/s	200 KB/s	1.0 MB/s	1.0 MB/s	4.0 MB/s
Max. target interface speed	15 MHz	4 MHz	15 MHz	15 MHz	50 MHz
Max. SPI interface speed	12 MHz	4 MHz	12 MHz	12 MHz	50 MHz
Max. SWO speed	30 MHz	4 MHz	30 MHz	30 MHz	100 MHz
High Speed Sampling Bandwidth	1 kHz	1 kHz	1 kHz	1 kHz	Unlimited
Supported target voltage	1.2V - 5V	3.3V	1.2V - 5V	1.2V - 5V	1.2V - 5V
USB	Yes	Yes	Yes	Yes	Yes
Ethernet	No	No	No	No	Yes
JTAG interface	Yes	Yes	Yes	Yes	Yes

Table 5-4: J-Link Version and Comparisons

5.2 Initial Development Board Testing

To showcase the capabilities and test the DWM3001C module we plan to use for this project Qorvo has supplied a PC application. With this Windows application, we can test line-of-sight range, non-line-of-sight range, and time-of-flight ranging. We can also observe the device's ultrawideband channels, modes, data rates, and preamble codes. The initial test of our development board will give us insight into implementing the ultra-wideband and BLE technology into our own ultrawide-band tag.

The evaluation board needs to be connected to a PC via USB. For the development board to connect to the program and be tested, it first needs code flashed to its nRF52 series microcontroller. To program the microcontroller for Bluetooth detection and ultrawide-band ranging we used the SEGGER J-Flash lite to flash the provided binary file pre-built by Qorvo. Figure 5-9 shows the program and its specifications needed. Flashing is relativity quick and after it is finished there isn't anything else to do but begin testing with the computer program.

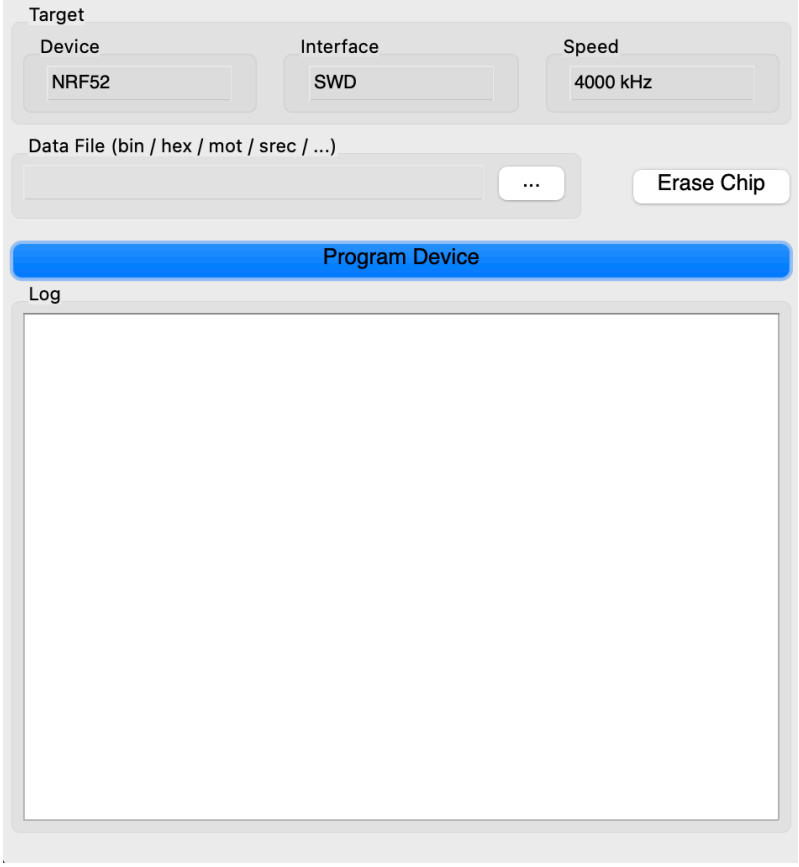


Figure 5-9: SEGGER J-Flash Lite to Flash Binary File

J-Link Lite will compare, erase, and verify the flash. After the software is done programming, we can verify that the binary code is successful on the board from the LEDs. The LEDs specified D20, D13, D9, D12, D11, and D10 will light up to show the development board has begun Bluetooth broadcasting for the PC program to connect to. Figure 5-10 displays what was discussed in writing and a picture of the development board.

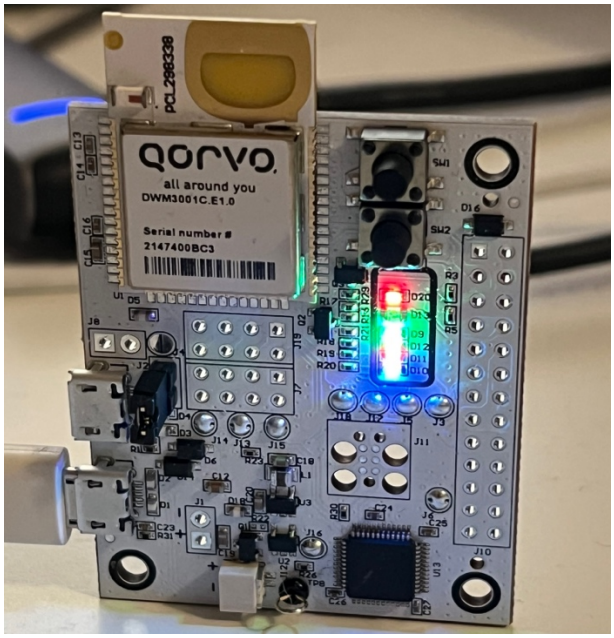


Figure 5-10: Personal Photo of DWM3001CDK Successful Flash

The preliminary setup of the computer program begins with a COM port driver installation. From there the executable file can be installed onto a system for us to run. When first running the application, we must specify the COM port that we have connected to the development board via USB. This selection screen can be shown in figure 5-11.

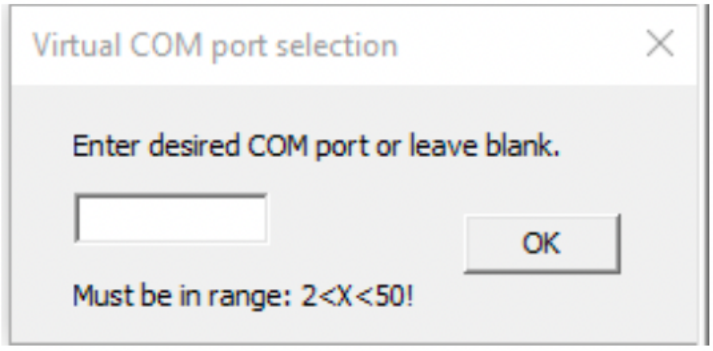


Figure 5-11: Virtual COM Port Selection

Once the valid COM port has been selected and connected to the unpaired tag can found in discovery phase. Discovery phase listens for a connect from the program. When connected we can initiate ranging and look at the statistics. To enable the ranging function, we set the role first. In this case we specify tag mode, in this mode the devices are waiting for a ranging initiation. When ranging is activated, we can see the status and statics of this ranging inside the application. This can be seen in figure 5-12 below.

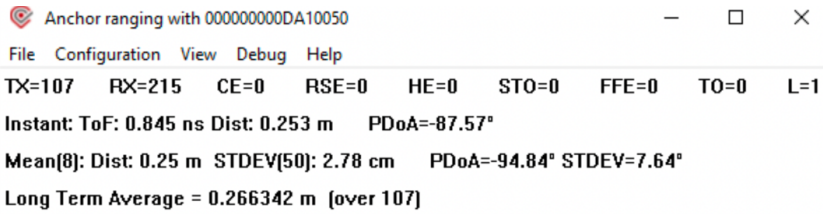


Figure 5-12: Ranging Statistics

This data has been important to our ultra-wideband tag project regarding setup, configuration, and usability. It was a significant part of this project because it gave us a solid basis to understand how to use the ultrawide-band technology. It also gave us insight into Bluetooth connectivity between two ultrawide-band devices. Our group has gained a much better understanding of the measurement and reporting that is involved with the microcontroller. Additionally, what we hope to implement inside our mobile app. To decipher some of the more important data in figure xx we overserved when testing table 5-5 can be used.

Value	Description
TX	Number of frames transmitted
RX	Number of frames received
CE	Count of errors
RSE	Unrecoverable errors
STO	Timeout events
ToF	Measured time of flight
Dist	Distance in meters
PDoA	Measured degrees

Table 5-5: Guide for Test Data

5.3 PCB Design Software Considerations

This sub sections will discuss why we chose EAGLE as the primary program for PCB Design.

5.3.1 Autodesk EAGLE

When choosing a PCB design software, there are many things that one should take into consideration such as price, how malleable is the software, what features it grants, its limitations, and its performance. For our project, we were interested in software that will fulfill our economical constraints related to our set budget, and that is we have decided to consider two design software that is commonly used by students because they are either free to them or they have been granted a yearly pass. These two designs are EAGLE and KiCad. EAGLE PCB design software is one the best-paid software on the market, but it can be a little bit pricey because the annual subscription costs $495. However, if someone only needs it for a month, EAGLE has an option where a user could get a monthly subscription only for $60, which can be convenient for the user if he or she wants to use it for a small period. Eagle is the property of Autodesk, and it was acquired in 2016. There are many key features to this app, and they are mentioned in table 5-6.

Feature	Description
New Routing Engine	Avoids obstacles by using the routing technique
Advanced ERC And DRC	A new rule-checking engine that covers electrical and design
Design Of High-Speed Features	Help related to physical constraints of the design
3D Viewing Capabilities	Being able to view the design in 3-Dimension
Reusable Design Blocks	Blocks are saved and will be ready to go when needed
BGA Fanout Router	A solution to high-density interconnect which shrinks the PCB
Supply Chain Management	Number of libraries with parts that the software contains

Table 5-6: Key Features of EAGLE Software

EAGLE is supported only on windows 7, 8, and 10 (64-bit), Mac OS (64-bit), and Linux Ubuntu (SSE, 8bpp). It also requires 3GB RAM, a DSL internet connection, and 650MB of available storage space.

5.3.2 KiCad

As for the KiCad design software, it is one of the best free design software programs on the market, and it also offers some functionality that is even better than some of the programs that are paid for. KiCad EDA has been around since 1992, and it is now many times better than how it was 30 years ago. Also, KiCad is an open-source program and has many great features such as schematic capture, layout, and routing. Some of the key features of this software have been noted in table 5-7 below.

Feature	Description
Online Library Import and Export	Used to import and export online components
3D Viewing Capability	Components and boards in 3-dimensions
Intuitive UI	User-friendly interface
Open-Source	Does not require a subscription or pay fees
Design And Electrical Rule Checking	Hassle-free design verification and component checking
Open Import/migrate	Supports schematics from EAGLE, Altium, CADSTAR
Integrated Simulation	Various built-in SPICE models

Table 5-7: Key Features of KiCad Software

Unlike EAGLE, KiCad is supported by more platforms and supports ARM-based machines, ARM 32 and 64-bit. Some of its supported platforms are Windows 7, 8, 8.1, and 10, and Apple macOS 10.12, 10.13, and 10.14. KiCad only requires 1GB of RAM and up to 10GB of storage space, and it requires a screen resolution with at least 16K colors to run perfectly.

5.4 PCB Design Software Selection

We have talked about two different types of design software that we were considering and about the key features that they both have, and based on these key features, we have decided to select the EAGLE design software as the software that we will be using to create our PCB design.

Even though KiCad seems to be a very interesting program since it is free, it has some performance limitations that could cost us time and effort. For example, Kicad does offer auto-routing or any tools that will help the users to use high-speed features, has slower performance, and its library is not quite impressive as the EAGLE library because there is limited part availability. As students, we can get the educational license software at no cost and we will be using better design software that is both efficient and powerful for taking care of PCB designing jobs and for advanced circuits and boards.

5.5 Hardware Design

This sub section involves all things related to the hardware design of this project.

5.5.1 MCU

The Design for the MCU is shown below in figure 5-13. The MCU includes test points to help debug potential problems that could arise in the developing stage. The data sheets and similar design by Qorvo were followed heavily in the design and layout of the MCU. The naming convention of the nets are followed by the datasheets for simplicity use. In the datasheet there are I2C devices incorporated in the design, we decided to leave these out as it wouldn't be necessary for our requirements. There is numerous input power supply labeled VDD. Each of which is connected to a capacitor to create a coupling capacitor. There is one VSS port which is connected directly to ground. A total of 6 connections were made to the UWB module for communication.

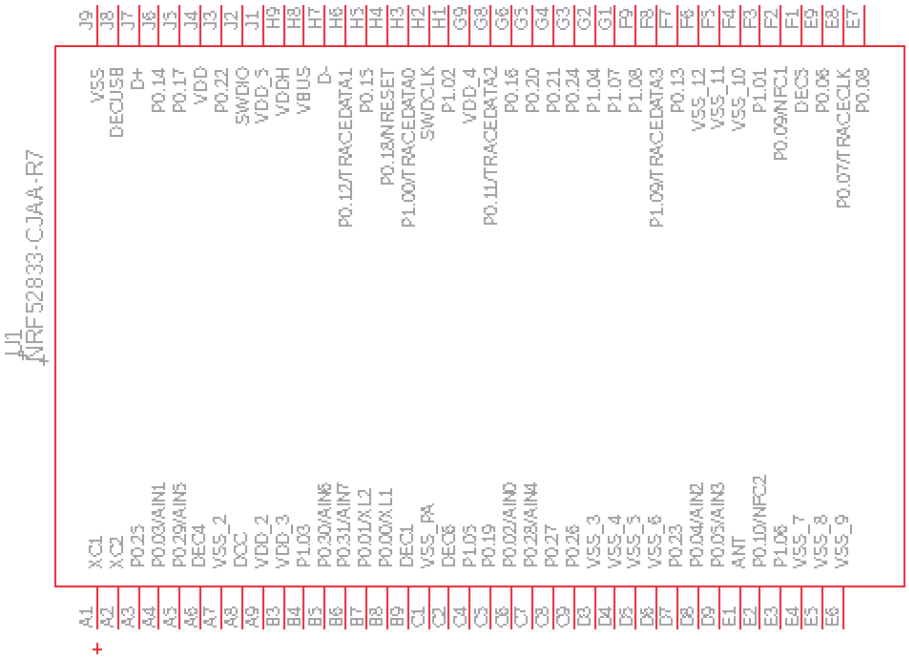


Figure 5-13: Schematic of nRF52833 Microcontroller

5.5.2 UWB

The design for the UWB module is shown below in figure 5-14. This chip includes a DC/DC converter as well as a Ceramic UWB antenna combined with the DW3110 UWB chip. With this integration they created a 24-pin chip that already has a UWB included. In our PCB design we need to take this antenna into consideration and not have any metal interference so we can have a greater accuracy. This module only needs to have power connected, which is 3.3 volts, and a communication connection to the host microcontroller.

All the GPIO pins include a software controllable pull up/down resistors to ensure a safe operation. Each GPIO pin has a different operation which can be easily found in the datasheet. We use this information to help connect to the pins of the microcontroller. The SPIMISO, SPIMOSI, IRQ, SPICLK, SPICSn, and RSTn are the 6 pins that are connected to the microcontroller for proper communication. The SPICLK, SPIMISO, SPIMOSI is the SPI clock, data output, and data input respectively. The RSTn is the reset pin, which is active low by default, and could be pulled low to open drain and reset the module. The IRQ pin acts as an interrupt request output from the DWM3000 module that will be heard from the microcontroller.

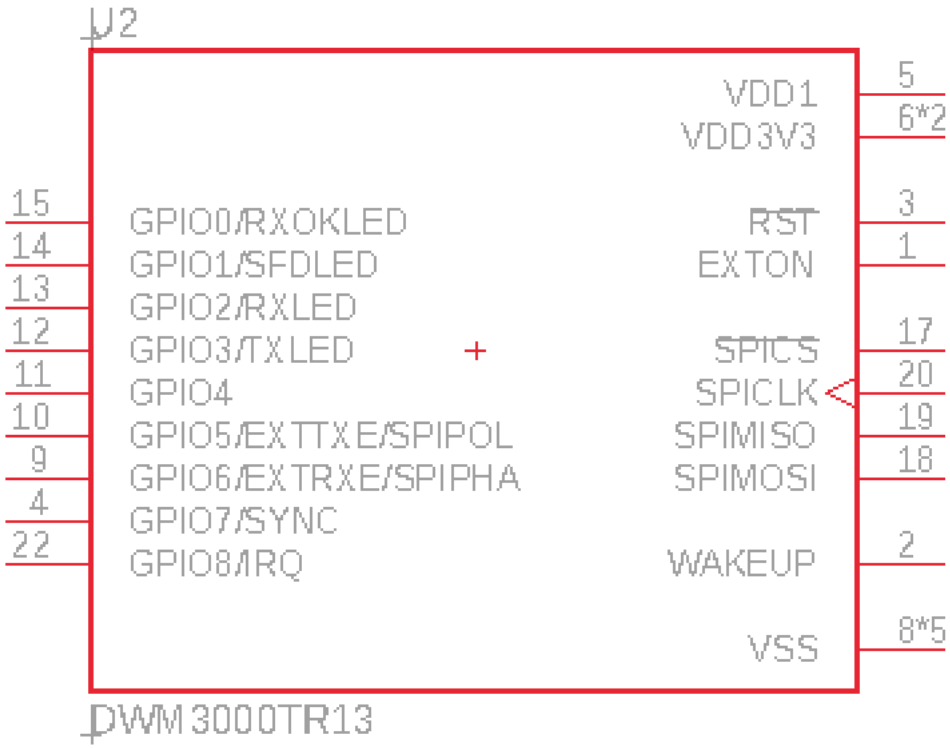


Figure 5-14: Schematic of DWM3000TR13 (Integrated UWB module)

5.5.3 DWM3001C

The design schematic of the fully integrated UWB chip is shown below in figure 5-15. this module has the microcontroller with Bluetooth capabilities, the UWB with antenna, as well as some other I2C protocols. What we plan on doing is to use this as our minimal design and use it mainly for software testing and for creating a benchmark in our actual design.

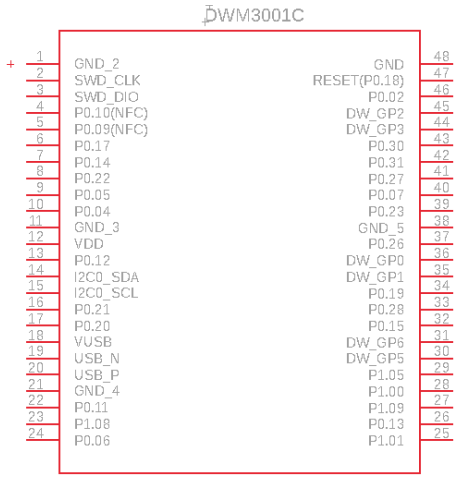


Figure 5-15: Schematic of DWM3001C (Fully Integrated UWB Chip)

5.5.4 DW3110TR13

Below in figure 5-16 shows the schematic of the bare UWB chip. This is the chip that is included in the UWB module we will be using in our design.

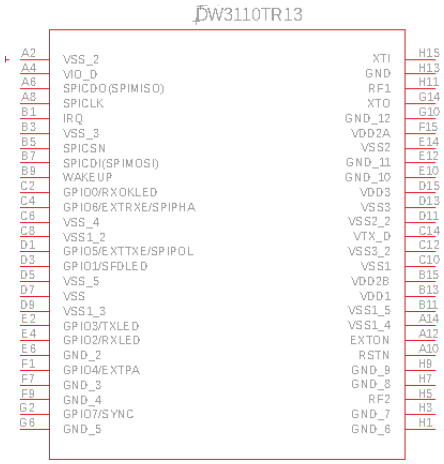


Figure 5-16: Schematic of DWM3110TR13 (Base UWB Chip)

5.5.5 Buck Boost Converter

The design for our BUCK BOOST converter is shown below in figure 5-17. it uses the TPS61098 Boost converter that is high in efficiency and Ultra-Low Quiescent Current. For our design using the DWM3000 which is the integrated UWB chip this Buck Boost converter could possibly be left out. The chip includes a DC/DC converter and thus has a wider range of supported input supply voltages. The same can be said for the nrf58233 microcontroller. There is a chance that we don't use the integrated chip and use the specifically UWB only chip (DW3110), we would then need to implement this into our design. The DW3110 doesn't have a range of supported input supply voltages so it's necessary to have our DC/DC converter.

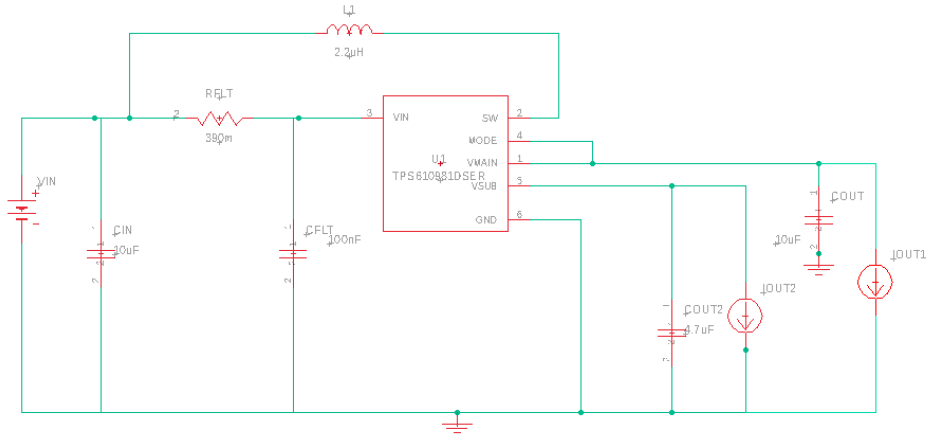


Figure 5-17: Schematic of Buck Boost Controller

5.5.6 Decoupling Capacitors

Many decoupling capacitors are needed for this microcontroller. It's necessary for the input voltage to be a constant DC value and the capacitors help eliminate the noise that could be present in the circuit. As far as the values for each capacitor, the datasheet was closely followed.

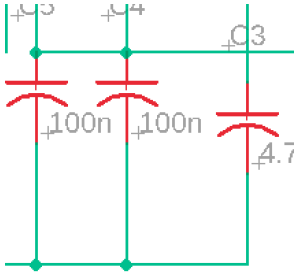


Figure 5-18: Schematic of Decoupling Capacitors

5.5.7 Battery Holder

The schematic of the battery holder is shown below in figure 5-19. there are hundreds of battery holders to choose from on the market but considering that we are going to use the CR2032 battery we needed to have a battery holder that is small and fits this type of battery. The BU2032SM-BT-GTR was chosen because it is a Surface mount holder and was one of the smallest in the marker with a length from terminal to terminal of 31.86mm, and width of 16mm. our design. Needs to be minimal in weight so with this holder we would roughly increase our overall weight by 1.12 grams, which is light when considering this is going to be the heaviest of the PCB elements.

For our design we need to keep in mind that the device will be in harsh environments and will need to pass rigorous test. According to the BU2032SM-BT-GTR datasheet this battery holder will be suffice in passing the environmental tests. The specification of this holder is shown in a list below:

- Holder material- White Zenite, 30% glass reinforced Liquid Crystal Polymer resin
- Contacts: 0.020" thick
- Temperature range: -40°C to +280°C
- Vibration Test: 10-50-10 Hz swept in 60 seconds X, Y, Z direction: 1 hour each direction
    - * Result: No dislodgement of the cell or damage No discontinuity over 10 micro-seconds
- ROHS COMPLIANT

Figure 5-19: Schematic of BU2032SM Battery Holder

5.5.8 Overall Schematic

Below in figure 5-20 shows are overall schematic thus far, as we might still add components as we see necessary. We were able to follow both datasheets of the DWM3000 UWB module and the nRF52833 microcontroller to successfully connect all necessary pins and have all the necessary components.

In this schematic the UWB module and microcontroller communicate by SPI communication. There needs to be a 32k crystal connected to the microcontroller which we are still under the research of how to implement it on our design.

We will be soon implementing a header to have software communication. We plan on doing this with the fully integrated module first to ensure we have a correct running code, to help with our debugging later.

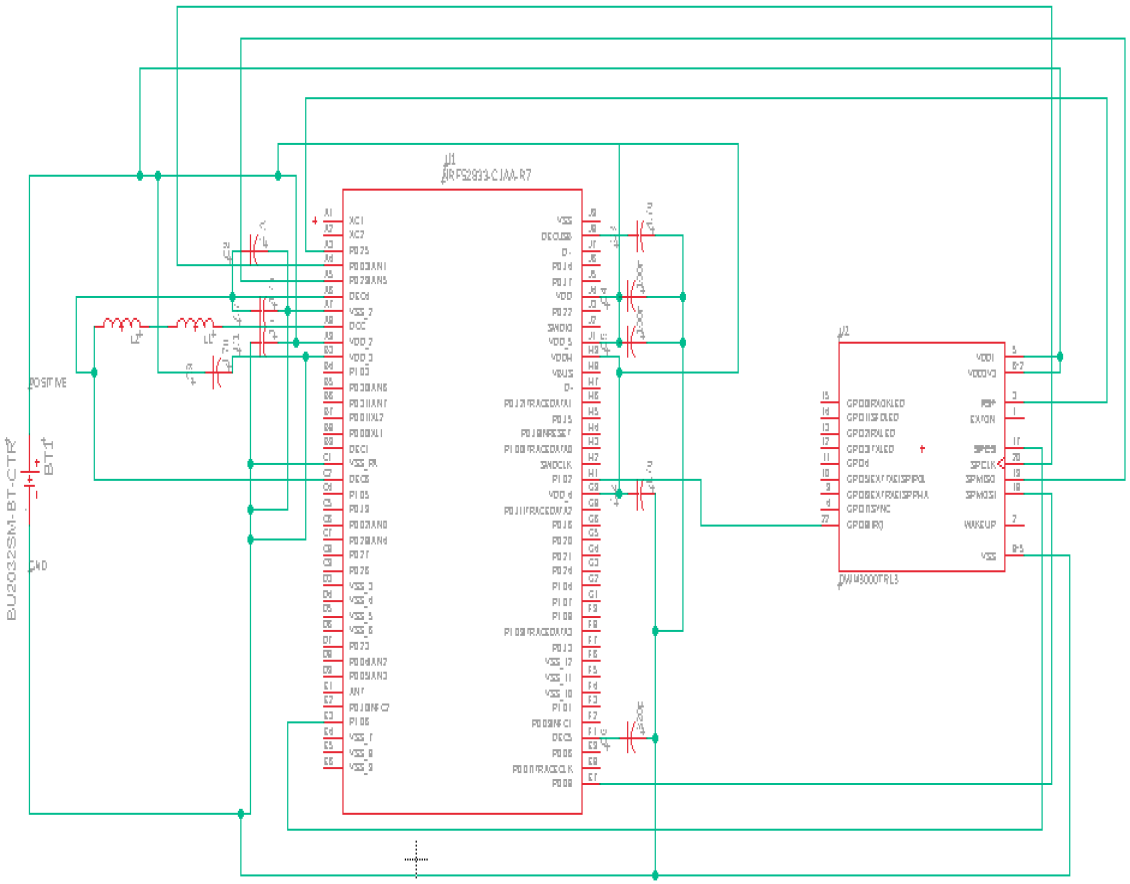


Figure 5-20: Schematic of Overall Project

5.5.9 BOM

The major components selected for the final PCB design are listed in table 5-8 below. Capacitors, inductors, and resistors are not included as it would be a large list, for there are many different values of each. There are plenty of these components in stock, so ordering them should not be an issue. The only component not in stock is the fully integrated UWB module which we already have on hand.

Component	Model	Manufacturer	Source	Price
MCU	NRF52833-CJAA-R7	Nordic Semiconductor ASA	Digi-Key	$3.30
UWB module	DWM3000TR13	Qorvo	Mouser	$20.02
Integrated UWB	DWM3001C	Qorvo	Mouser	$19.51
UWB chip	DW3110TR13	Qorvo	Mouser	$8.27
Battery Holder	BU2032SM-BT-GTR	MPD (Memory Protection Devices)	Digi-Key	$1.20

Table 5-8: Bill of Material of All Major Components

5.6 Software Development Tools

For this project we had many uses of various development tools. In this sub-section we will cover these tools we used for the software portion of development.

5.6.1 Microsoft VScode

VScode or Visual Studio Code is a source-code editing program created by Microsoft with the Electron framework. Electron is an open-source software framework that was designed by GitHub to create desktop applications using web technologies. Visual Studio Code is available for Windows, macOS, Linux, and Rasberry Pi OS. Microsoft fist announced the Visual Studio Code project on April 29, 2015, at the 2015 Build conference. Under MIT license, the source of Visual Studio Code was released on November 18, 2015 and made available on GitHub.

Visual Studio Code was initially designed for developers to create web apps using JavaScript and TypeScript. This source code editor not only comes with built-in

support for JavaScript and TypeScript but for Node.js as well. Visual Studio Code includes a variety of extensions for other programming languages like C++, C#, and Python. VScode also includes extensions for runtimes and clouds. Some examples of these runtimes are .NET and Unity. Examples of clouds that are supported by VScode are Amazon Web Services, Microsoft Azure, and Google Cloud Platform. Visual Studio Code offers IntelliSense code completion for things like variables, methods, and imported modules. IntelliSense is a very helpful feature to include in a source-code editor program. IntelliSense speeds up the process of coding applications by reducing typos and correcting common coding mistakes. IntelliSense helps the developer by requiring less keyboard input and necessity for name memorization. IntelliSense offers the software developer the convenience to refer less frequently to external documentation.

VScode provides syntax highlighting to help developers keep track of the code and be more productive. Visual Studio Code offers bracket matching to help with having the correct number of brackets. Having an equal number of brackets can be one of the most tedious parts of programming. When debugging code, developers can spend a long period of time looking for errors and most times the problem would be a bracket that is needed in a line of code. The intuitive keyboard shortcuts offered by Visual Studio Code can help with minimizing errors as well as the time spent on writing the code. Visual Studio Code offers an interactive debugger that helps software developers correct mistakes and run the code more efficiently. Visual Studio Code has built-in support for the runtime from Node.js. The debugger has the ability to debug code written in any language that gets transpiled to JavaScript. To debug other languages and runtimes, debugger extensions can be found in the Visual Studio Marketplace. Selecting “Install Additional Debuggers” from the top-level Run menu is another way to get access to debugger extensions.

In summary, Visual Studio Code is a source-code editing program created by Microsoft with the help of the Electron framework. Visual Studio Code provides many features that can help with editing and optimizing a program. VScode includes a variety of extensions for programming languages like C, C++, and Python. This software program also includes extensions for clouds like Amazon Web services, Microsoft Azure, and Google Cloud platform. Visual Studio Code also provides extensions for runtimes such as .NET and Unity. Another feature that is offered by Visual Studio Code is IntelliSense. IntelliSense is a context aware code-completion feature. This feature helps software developers by reducing the time it takes to complete the code. The feature reduces time by automatically correcting typos and other common coding mistakes. Our group believes that all of the features that Visual Studio Code provides will help us in writing safe and efficient code for our project. The group will be using Visual Studio Code to edit the codes that are written in C and C++.

5.6.2 Nordic nRF Connect for VScode

For our project, we will use nRF Connect to link the Nordic nRF52833 DK for Visual Studio Code. Using the Visual Studio Code Integrated Development Environment (VS Code IDE), nRF Connect for VS Code will allow the user to develop and debug applications based on the nRF Connect SDK. Information on nRF Connect SDK can be found in another section further down in the document.

As mentioned in a previous section of the document, Visual Studio Code is a fast, cross-platform and popular integrated development environment. The Nordic nRF Connect extension pack converts Visual Studio Code into a complete IDE for developing applications for the nRF52, nRF91, and nRF53 series devices. nRF Connect for VS Code is supported on Windows, macOS, and Linux. nRF Connect for VS Code includes an interface to the compiler and linker. nRF Connect for VS Code also offers features like an RTOS-aware debugger, an interface to the nRF Connect SDK, and a serial terminal. A DeviceTree viewer is another feature offered by nRF Connect for VS Code. DeviceTree is a data structure that describes the hardware components of a computer. This way the operating system's kernel can use and manage those components. The components that the operating system's kernel manages are the following:

- CPU
- Memory
- Buses
- Integrated Peripherals

DeviceTree can hold any form of data due to it being an internal tree of named nodes and property. nRF Connect for VS Code helps visualize the hardware components and makes it easy to navigate the DeviceTree source. The Create New Board wizard included in nRF Connect for VS Code makes it significantly easier to add custom boards. Custom boards are useful for laying out the blueprint for the hardware components that are going to be configured in the project. nRF Connect for VS Code offers additional extensions from Microsoft for C, C++, and many more to complete the firmware development life cycle. This development environment gives the user access to a command line interface through a built-in terminal and an intuitive graphical user interface. The project production can be increased by the native support of scripting, commands, and shortcuts.

There are a couple of things that will be needed in order to install nRF Connect for VS Code. The nRF Command Line Tools will need to be installed to develop, program, and debug the Nordic nRF 52833 DK. Next, the nRF Connect for Desktop must be installed to gain access to the Toolchain Manager. Once this step is completed, the Toolchain Manager must be installed and opened. Once VS Code is installed on the desktop, VS Code must be opened from the Toolchain Manager to install the nRF Connect Extension pack. A list of missing extensions will pop up that need to be installed. These missing extensions can be installed by clicking on

install missing extensions. When this process is complete, it is important to relaunch both the Toolchain Manager and nRF Connect for Desktop to check for any remaining tools that need to be installed.

In conclusion, nRF Connect for VS Code will be used to convert Visual Studio Code into a complete IDE to develop applications for the Nordic nRF 52833. nRF Connect for VS Code has many features that can help with optimizing and editing codes. Some of these features included in nRF Connect for VS Code are RTOS-aware debugger, interface to the nRF Connect SDK, serial terminal, and an interface to the compiler and linker. It also includes the DeviceTree feature which helps describe the hardware components of a computer to the operating system's kernel so that the components may be used and managed.

5.6.3 Nordic nRF Connect for Desktop

Nordic nRF Connect for Desktop is a cross-platform development software or tool framework designed for Nordic products. It was created to help the developers to test, monitor, measure, optimize and program the applications they will be working on. It was specifically designed to work along with the development kits such as the Nordic Profiler II or dongles which are small pieces of computer hardware that connect to a port on another device, so there are additional functionalities that could be added, or enable a pass-through to the device that is capable of adding the functionalities. Some of the nRF Connect for Desktop Apps are:

- Bluetooth Low Energy
- Direct Test Mode
- Getting Started Assistant
- LTE Link Monitor
- Power Profiler
- Programmer
- RSSI Viewer
- Toolchain Manager

In table 5-9, are some of the key features that the Nordic nRF Connect for Desktop has. There is no such thing as perfect software, and that is why it is important for developers to keep their mindset open for changes. Most of these changes will come overtime, when they have certain feedbacks from consumers or when they have new ideas that they could include in the next updates. That is why automatic updates is very important because it creates an undirect connection between the consumers and developer since they have the possibility to share their feedback based on their need and how the software works, and the developers will be able to address the issues accordingly. The software is also compatible and will be able

to connect to all nRF Connect for Desktop apps. Furthermore, the nRF Connect software can test Bluetooth low energy connectivity and supports its security features, and it can monitor cellular modem/link status and activity using AT command which its documentation can be found in Nordic Semiconductor Infocenter. The software is also able, just like mentioned in the table, to measure the power consumed by the device.

	Key Features
Nordic nRF Connect for Desktop	Automatic Updates
	All nRF Connect for Desktop apps
	Bluetooth LE
	Cellular
	Power Measurement
	Toolchain Management

Table 5-9: Key Features of Nordic nRF Connect for Desktop

5.6.4 SEGGER Embedded Studio

SEGGER Embedded Studio is a complete C/C++ development system or an all-in-one solution for microcontrollers and microprocessors that provides stability and continuous workflow. It is, as for today, one of the most powerful and efficient IDEs (Integrated Development Environment) for ARM controller that exist on the market. It was mainly designed for professional that deal with stability issues when realizing their projects, but that does not mean it can only be used by only them since it is also used by non-professional. What makes it different to the other IDEs is because was created specifically for embedded developers, because it has an intuitive user interface that is easy for the users to navigate through, works on all PC such as Linux, MAC, and Windows, and it all also support the newly ARM based processor machine. Also, there are no code length limitation nor feature restrictions because it is a full featured IDE for all users, either for commercial or non-commercial use. The embedded studio works with the J-Link kit that connects it to the PC via a J-Link Driver, which allows the embedded program to debug the targeted system or device.

As it can be seen in table 5-10, SEGGER Embedded Studio is an all-in-one solution just like it was described in the lines prior. Also, it has a great visual which is great for any user because it enhances their navigation experience when using it.

Furthermore, it is free for both commercial and non-commercial use, but they must be licensed with partner devices such as Nordic. Moreover, developers can choose the preferred coding language like Clang/LLV, C/C++, and more. At last, it is has multi-threaded build that minimizes the amount of built time that is required, and it be used in multi-platform like Windows, Linux, and Mac.

	Key Features
SEGGER Embedded Studio	All-in-one solution
	Visual Studio-like appearance
	Free for commercial use with licensed partner devices
	Free for non-commercial use
	Clang/LLVM & GCCC & SEGGER C/C++ toolchains included
	Multi-threaded build minimizes built time
	Multi-platform: Windows, Linux, or Mac

Table 5-10: SEGGER Embedded Studio Key Features

5.6.4.1 SEGGER Embedded Studio Price

If it is true that SEGGER Embedded Studio is one of the most user friendly, time efficient, and interactive embedded programs in the market, its prices speak volume because it is not a cheap product. SEGGER Embedded Studio has three main license types, such as Cortex-M, ARM, and RISC-V. By referring to table 5-11, we can denote that the least expensive license type is the Cortex-M, which costs $1,848, and two others are $2,498 both. However, if one is a Nordic user, it will be free because of the partnership that exist between SEGGER and Nordic. Since it is free for Nordic users, using Nordic product will grant us access to this technology that will be very beneficial to our project development. Being able to utilize an embedded tool as such will put us in better condition to code and debug our tracking devices. Also, because of all the features that it has, it would benefit our group since it is able to minimize build time and easily use it because of its user-friendly interface.

License Type	Fee
Nordic Embedded Studio	Free for Nordic Users
Embedded Studio Cortex-M (20.00.23)	$1,848
Embedded Studio ARM (20.00.00)	$2,498
Ebedded Studio RISC-V (20.00.55)	$2,489

Table 5-11: SEGGER Embedded Studio Fees

5.6.5 SEGGER JFlashLite

J-Flash Lite is a flash programming application that programs data images to the flash of a microcontroller. when comparison to J-Flash, J-Flash Lite has a reduced feature set hence the Lite version. J-flash Lite does not require a J-Link PLUS or higher to operate. J-Flash Lite is very simple to use and is user friendly. It consists of only two components: The configuration dialog which allows the user to configure the programing or debugging, and the main window which is the user interface of the program. Some of the limitations of J-Flash Lite when compared to J-Flash include:

- No support for external CFI NOR flash
- No support for custom initial steps
- No support for automation from the command prompt
- No project management support

Some features of the J-Flash Lite are shown below:

- Any ARM7/ARM9/ARM11, Cortex-M0/M1/M3/M4/M7, Cortex-A5/A8/A9/R4/R5 and Renesas RX600 core supported
- Microcontroller support.
- Support for most external flash chips.
- High speed programming: up to 550 Kbytes/s
- Smart read back
- Verbose logging of all communication.
- .hex, .mot, .srec, .bin and .elf support.
- simple user interface.

5.6.6 Xcode

Xcode is an integrated development environment created by Apple to develop software for macOS, iOS, watchOS, and tvOS. Xcode is the only tool officially supported for developing and publishing apps to Apple's app store. This application is designed for use by beginners and experienced developers.

The first version of Xcode was released back in 1992 by NeXT and it was called Project Builder. At that time, this integrated development environment was revolutionary because it used the Objective-C programming language which was new back then. Project Builder was modified and rebranded as Xcode in 2003 with Mac OS X Jaguar.

Apple has made many changes throughout the years to the Xcode program. In 2005, Apple released Xcode 2 that included the Quartz Composer and the Apple Reference Library. Apple released Xcode 2.1 which included an intel processor instead of PowerPC. This allowed developers to build universal binary apps that could run on both processor types and provided Mac developers the ability to write truly processor-independent code. During this time, Apple added new features like unit testing for targets, conditional breakpoints, and watchpoints. With the release of Xcode 3.1 in 2008, developers were now allowed to target both Mac OS X and iPhone OS 2.0 using the iPhone SDK. This was a groundbreaking achievement because back then no other company had offered an IDE that would allow developers to create apps for both their main platform and mobile platform. The 5th version of Xcode came with support for 64-bit ARM architectures for iOS devices and removal of garbage collection binaries for Mac targets in version 5.1. The 6th version of Xcode introduced the use of Swift programming language and Playgrounds support for quickly iterating over Swift code. Xcode 7 came withs support for the Swift 2.0 language updates and the introduction to Metal. The next few versions of Xcode came with support for new updates on the Swift programming language. Versions like 8 and 9 included new features like new SDKs for Mac, iOS device updates, wireless debugging of iOS, and tvOS devices.

The 10th version of Xcode came with many new key features. This version included support for Bitbucket, Gitlab, and Github for source management. This version also offered the ability to train CoreML models. The changes made to the editor and build systems in this version allowed for projects to build at a faster rate. These changes to the editor allowed room for new autocompletion features. Xcode 11 added support for Swift 5.1 and the Swift UI framework that allowed developers to lay out their Swift apps for all Apple platforms in a new way. The two latest versions of Xcode remain relatively unchanged with only minor changes.

Xcode includes many features to help experienced developers create powerful apps. It also has great features to help beginners face less confusion and barriers to create a good application. Xcode supports a larger variety of programming languages like C, C++, Objective-C, Objective-C++, Java, AppleScript, Python,

Ruby, ResEdit, and Swift. Xcode uses the Cocoa, Carbon, and Java programming models. Xcode makes it easier for developers to write source code by offering a source code checker and autocomplete feature. Xcode also offers templates to give the developer a basic framework to expand on. These features are helpful for new developers to rely on as they learn to code. More advanced developers use these features to streamline their workflow and make the development process much faster.

To conclude, Xcode is an IDE created by Apple for developing software for macOS, iOS, watchOS, and tvOS. The program has gone through many changes throughout the years to provide a great experience for new and experienced developers. Today, the features offered by Xcode allow developers to create fast and powerful applications for Apple. Some of the features included in Xcode are source code checker, autocomplete feature, and templates to give a basic framework to expand on. Xcode also supports a larger variety of programming languages like C, C++, Objective-C, Objective-C++, Java, AppleScript, Python, Ruby, ResEdit, and Swift. These features will be helpful in creating a strong application used for the range finding process. As engineers with minimal experience, we believe that Xcode is the perfect program to help us design code for our project.

5.6.7 Discord

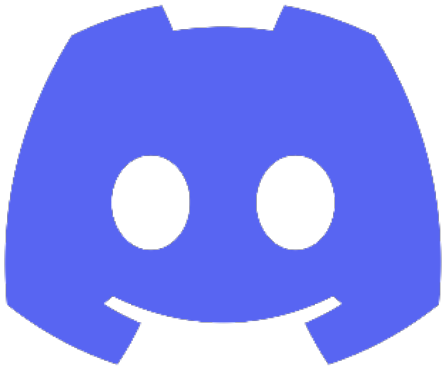


Figure 5-21: Discord Logo

If one pays attention to the shape that the Discord Logo has in figure 5-21, it can be deducted that it has a controller shape. At first, 7 years ago, when Discord was created, it was meant to be used by gamers, so they could feel more connected, play together, and chat. It was a dream and passion that, both, Jason Citron and

Stan Vishnevskiy shared, so they could create more connection among gamers. In 2015, they finally launched the app that was called discordapp.com. From 2015 to 2019, Discord was considered as the biggest gaming playing platform and attracted investment from many big companies, such as WarnerMedia, Twitch, Microsoft, and AT&T. By the end of 2019 Discord was evaluated at $2 billion, and its motto was “Chat for the gamers”.

However, in 2020, when the COVID-19 hit and restrained people from going through their daily routines, Discord became an essential platform where business owner could trade crypto, where student could regroup and work on their projects, and where people could communicate. Hence, Discord quickly changed its motto and decided to target a bigger audience than only gamers.

The new Discord’s motto is “Chat for Communities and Friends”. That audience shifting and the pandemic benefited them so much that in 2021 its valuation went 5-fold from what they previously estimated. Hence, they continue to improve their app and change its name on every platform to only Discord instead of discordapp.com.

We, as a group really benefited from the shifting of audience and all the updates that have been made to that app for the last two years. We used Discord in many ways, such as share our thoughts, discuss about the projects, group meetings, files and documents sharing, and weekly huddles. The way it is designed made it very useful to us for organizing our work and to keep track of what is important.

We usually communicate through the main text channel, which is called general, but for our meeting we use the voice channel that allows use to talk and hold our briefings. Also, in the voice channel, we sometimes use the screen sharing tools when we want to illustrate, demonstrate, or visually explained some part of our project. Since it is a free app, there are some size constraints when sending files that are more than 10 megabytes. That is why, in order to stay within the size constraints, we either zip the files or share a link that contains the information needed.

Discord is a very elaborated app and simple to use. It has given us the flexibility to work in group no matter where we are. Also, it also helped top stay within our budget constraints because we do not have to meet in person at any given location to work on our project. Also, this app is key to all the goals that we have met and to our project success because it helps us stay informed about changes related to the project that must be made or planned for the next steps. Some of the key features of this app that were useful to us are noted in the table 5-12 below.

	Key Features
Discord	User Friendly
	Text channel and Voice Channel
	Screen Sharing and Video Streaming
	Video Background
	Supported by PCs and Cellphones
	Server Folders

Table 5-12: Discord Key Features

5.6.8 GitHub

GitHub will be used as a version control and collaboration tool for the member working on the software design portion of the project. With a repository, everyone cand have access to the code written for the microcontroller and mobile app. This version control system will provide a simple way to make any changes while working on the same code at the same time. With the microcontroller and smartphone app code, we plan to work on it simultaneously and GitHub will be a powerful tool for code review. We plan to break down the coding into sections to merge any changes to separate branches for review. Figure 5-22 shows our GitHub page.

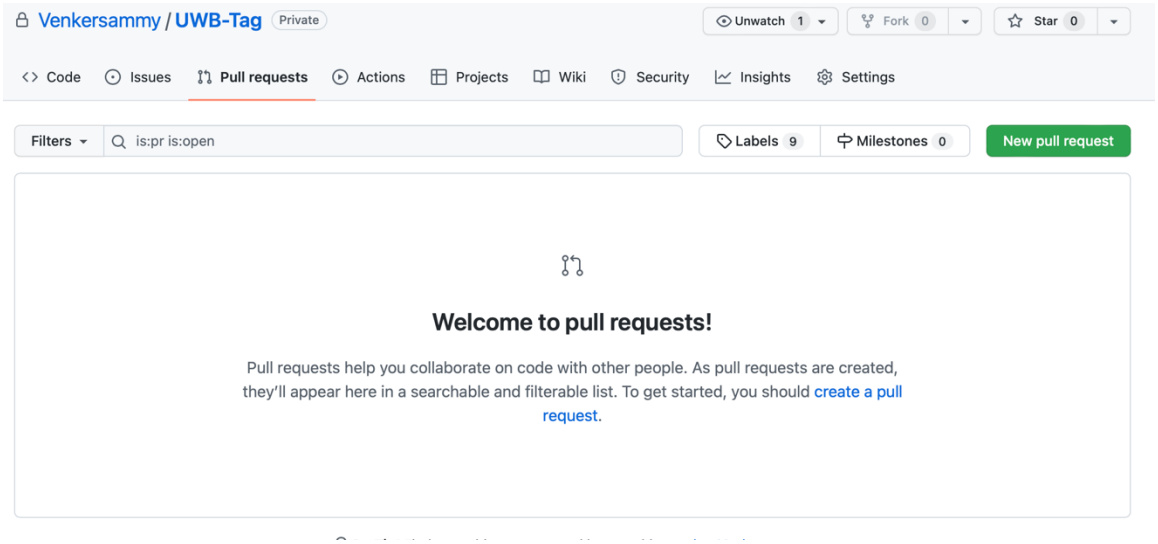


Figure 5-22: Group 23's GitHub Page

5.7 Programming Languages

This sub section will discuss the required programming languages for this project.

5.7.1 C

The C programming language is one of the most popular programming languages used today. It is a general-purpose programming language that is simple and flexible to use. The programming language is machine-independent which means that the program is portable and can be used on any machine. C programming language is used for a variety of applications, operating systems, and other types of complex programs. C programming language is known to be the base of programming which means that once you understand C, it will make it relatively easy to learn other programming languages that use the same concept. Having a background in computer memory mechanisms is essential when using the C programming language.

The base of most programming languages is ‘ALGOL’, and it was introduced in 1960. The ALGOL program introduced the concept of structured programming to the development community. The next programming language to be developed was called ‘BCPL’ which was designed for writing system software. The programming language called ‘B’ was developed after BCPL which contains some of the same features. In 1972, Dennis Ritchie created the C programming language. The C programming language was created from the programming languages ALGOL, BCPL, and B. C contains features from all these programming languages as well as many additional ones.

C is associated with the UNIX operating system. Most of the UNIX operating system is programmed with the C programming language. Today, C is used in a variety of operating systems and hardware components unlike before when it was only used in the UNIX operating system. As the language evolved, many developers found it difficult to keep up with the latest versions as the systems that were using the language were running on the older versions. The American National Standards Institute introduced a commercial standard for the C programming language in 1989. Today, many other programming languages are developed from C.

C is used in a large selection of applications. Coding for embedded systems is one of the applications where C is most applied. There are many reasons why C is the preferred language for this concept. C is not specific to any system and can be used to work on various hardware configurations. Portability is another reason why C is used for embedded systems. A program can be written on one machine and run on another machine with minimal modifications. Code written in C is also efficient and easy to maintain. Code written in C is compiled into raw binary executable which can be loaded right into the memory and begin execution. Machine instructions for inputs are optimized in C which increases the performance of the system. Other high-level languages rely on libraries which

means they require more memory and can be a challenge for embedded systems. C does not provide support for things like garbage collection or dynamic typing which can decrease the performance of the embedded system. C provides low level bit wise data manipulation which is very helpful when designing code for embedded systems.

In summary, C is one of the most popular programming languages in the development industry. It is very common to use C for designing programs for embedded systems which is one of the main reasons our group decided to use this language for our project. Our group will be designing code that will be used for programming the nRF52833 microcontroller. All members in our group have previous experience working with C language to program microcontrollers. The many features that the C programming language offers will help us in creating a code to optimize the performance of the microcontroller we will use.

5.7.2 C++

C++ was developed from C and the two are very similar. This programming language was created by Bjarne Stroustrup in 1980. It is named C++ to suggest “C incremented” and ++ is an increment operator in the C programming language. C++ can compile most C programs without changing a single line of code a large percentage of the time. C++ is an object-oriented programming language which makes it a more well-structured and safer language than C. An object-oriented program organizes software design around data or objects instead of functions and logic. C++ and C share the same syntax. Most of the operators and keywords in C can be found in C++ with the same functions. The basic memory model for C and C++ is very close to the hardware. Table 5-13 shows the key differences between the two programming languages.

C	C++
Does not support overloading	Supports overloading
Structures do not have access modifiers	Structures do have access modifiers
Does not support exception handling	Does support exception handling
Focuses on method or process	Focuses on data
Does not support inheritance	Does support inheritance.
Function-driven language	Object-driven language

Table 5-13: Differences Between C And C++

Just like C, C++ is used for a variety of applications. Some examples of systems that are developed with C++ are operating systems, IoT devices, and machine learning tools. Operating systems substantially benefit from C++ due to the low-level capabilities that this language offers. This can help developers make small changes to increase the speed and energy efficiency of an operating system. IoT devices include embedded systems which give the device limited computing resources and strict power requirements to work with. C++ helps developers manipulate the way the resources are used in order for the program to run efficiently. C++ acts as a back-end programming language when used for machine learning tools. The C++ code is what enables the core machine learning calculations to be performed. The large collection of libraries contained in C++ is what powers the complex calculations to train machine learning models.

5.7.3 Swift

Chris Lattner began development for Swift in 2010 along with other programmers like Doug Gregor, John McCall, Ted Kremenek, and Joe Groff at Apple. Swift is based on many other programming languages like Objective-C, Rust, Haskell, Ruby, Python, C#, CLU, and many more. Swift 2 was released in 2015 and it was a major upgrade from the first version. On December 3, 2015, Swift 2.2 was made an open-source software for Apple's software and Linux. The syntax of Swift was modified in its third upgrade which was a significant change for the programming language. Swift rapidly became one of the most popular programming languages on the market and overcame the popularity of Objective-C. Swift is now up to 2.6 times faster than Objective-C and 8.4 times faster than Python 2.7. The fourth version of Swift also went through some changes like updating code that was written in previous versions of Swift.

Apple has used Swift to create some of the most popular pieces of technology seen today. Apple has used Swift to design technology like the mac, Apple TV, and Apple watch. Today, much more companies have begun using Swift to develop their applications. Some examples of these companies are Lyft, Linkedin, and Evenbrite. Academic institutions are also adding Swift to their curriculum. Universities like Plymouth University, RMIT University, and Stanford University have begun to offer classes centered around the Swift programming language.

The Swift programming language has features that catch the attention of many software developers. Swift offers Automatic Reference Counting otherwise known as ARC. This feature determines which instances are no longer in use and disposes of them automatically. Closures contain function pointers that signify particular behaviors by calling on the function rather than writing code to perform the actions. Closures are unified with function pointers when using Swift's nested functions which allows values to be taken from the functions enclosed within the closure.

Swift allows the developer to return multiple values from a function instead of one. There are other C based languages that allow you to return multiple values by using

pointers or arrays. The difference is that Swift allows you to return the variables by using tuples which return a single entity from a function. Tuples are sets of elements arranged in an ordered sequence that are unchangeable. Swift offers some of the simplest ways to use loops in a program. Swift can efficiently run blocks of code by assigning a constant to the elements of an array. Swift offers extensions that add additional functionality when using structs. Swift also includes the option to extend types even if the developer does not have access to the original code source. Generics offered by Swift allow the developer to write functions and types that are reusable and can work with the requirements that are defined. Functional programming patterns are another feature offered by Swift. This function will take an array and repeatedly compute a value which in turn creates a pattern that is returned to the code that called for the action. Swift offers functions to better control the flow of the program when running. Some of the control flow statements are transfer statements, loop statements, and branch statements. Swift's do statement accesses the more advanced control flow features that allow the developer to introduce scope, guard to manage errors, defer statement to run clean-up actions, and repeat to allow for a block of code to run multiple times.

Swift programming comes with many advantages. Swift is designed to be safer than other languages that are based on C. Swift eliminates unsafe code to clean up the program and prevent more runtime clashes. The variables are initialized before they are used, arrays and integers are checked for overflow, and the memory is managed automatically. Code errors are easily detectable which reduces the time spent on debugging and lowers the risk of low-quality code. The compiler used by Swift is called LLVM. LLVM translates the assembly language to the machine code and optimizes it. This way, less code will be used in comparison to a language like Objective-C and can make development faster. Swift is an ideal programming language for mobile developers because it can be used across major platforms. Using cross-platform programs, Swift can run on platforms like Windows and Linux. Swift is considered a user-friendly programming language due to its use of concise syntax and inferred type.

To summarize, Swift is an open-source programming language that was meant to replace languages based on C. Compared to other programming languages, Swift is safe, fast, and user-friendly. Swift has features like Generics, Functional programming patterns, and powerful error handling to help developers create an efficient and safe program. Some advantages that come with using Swift are speed, cross-platform, and accessibility which helped the program grow in popularity over the last couple of years. Swift is mostly known for developing programs for apps. Our group has decided to use Swift to develop the phone application that will be used for range finding. We have come to this decision because Swift is fairly easy to use and has also created some of the most impressive technologies in today's market.

5.8 Microcontroller Software Design

This section will consist of major design choices for the microcontroller portion of this ultrawide-band tag device.

5.8.1 nRF Connect SDK

For our tag to operate the Bluetooth and ultra-wideband capabilities to our requirements we will be using a software development kit from Nordic. The nRF Connect SDK will give us the building blocks we need to create the software on our Nordic nRF52833 microcontroller. This SDK will integrate samples, protocols, stacks, libraries, and hardware drivers that will help us create optimized software for this portion of the project. Figure 5-23 gives an overview of the many portions related to the mobile application and microcontroller interaction to which the nRF Connect SDK will be applied to.

The software design of our ultra-wideband tag will revolve around Bluetooth connectivity with an Apple iPhone. It will also involve the user-triggered command inside the app of when to begin ranging between the finder and smartphone. For this to be possible an exchange of information using BLE will be needed. Without the tools inside nRF Connect SDK, it would be more time-consuming in terms of our development.

The most important part of any software development kit is the documentation. This SDK has documentation that is excellent including a lot of information on libraries and drivers even including examples for certain applications. Outside of the documentation, we can take advantage of Nordic's getting started tutorials for this SDK they offer.

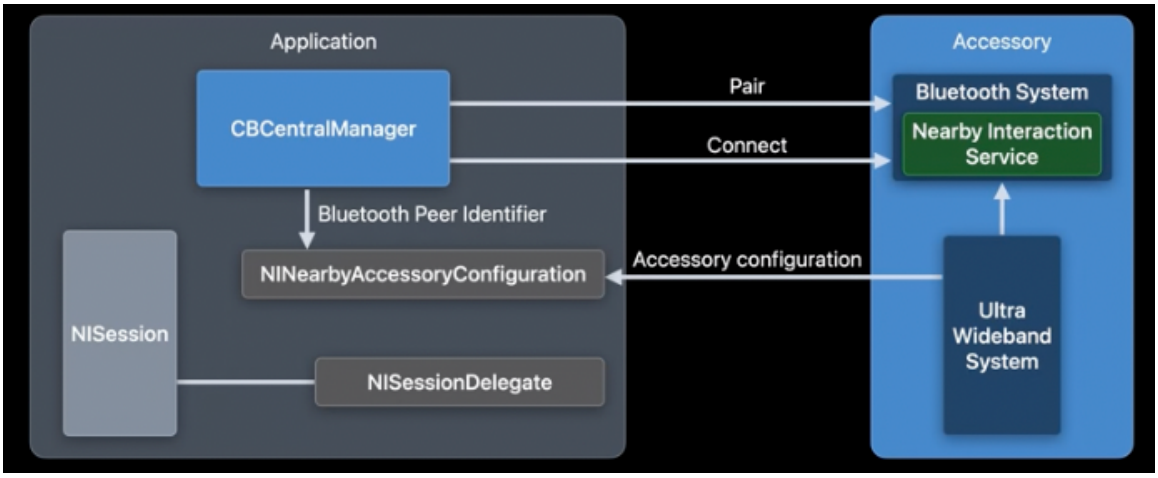


Figure 5-23: nRF SDK Usage Among MCU and App Interaction

Usage Approved by Apple Guidelines (see appendix)

5.8.2 Zephyr RTOS

RTOS (real-time operating system) is meant for embedded systems. We plan to use a real-time operating system for our microcontroller to respond to real-time events. Tasks are given different priority levels and the ones with the highest priority execute first, creating a responsive system that will switch between different tasks. The main purpose of using an RTOS in our device microcontroller is to have a scheduling service for ultra-responsiveness.

Zephyr RTOS is a type of real-time operating system we will utilize for this project. Zephyr supports multiple hardware architectures such as ARM which is being used by our project's microcontroller. It contains low-level drivers along with libraries that we will apply to our project. More importantly, nRF Connect SDK has been developed with a fork of Zephyr for its software development for connectivity such as Bluetooth low energy and ultrawide-band technology.

Zephyr RTOS is highly integrated within many tools we will use for this tag device. Such as nRF Connect SDK and nRF Connect for VS Code. It is open source allowing for easy distribution and readily available documentation. Figure xx is a diagram of all the components of nRF connect we plan to leverage for this project including the Zephyr RTOS kernel.

Our nRF device, the nRF 52833 will be integrated with the nRF Connect SDK and as a part of that will have a Bluetooth low energy stack. To initiate a session within our mobile app we will need the device configuration data provided by the microcontroller of our finder. We Zephyr RTOS will execute the correct tasks at the right time such as first-time connectivity, Bluetooth broadcasting, or Bluetooth connectivity. Zephyr will also provide the correct tasks when ultra-wideband ranging will be triggered by the user from the mobile application. Zephyr will provide much more than just the primary engineering requirements. As we expand the feature lists of our ultrawide-band tag we can use Zephyr to perform communication, configuration, and additional user-related functions later.

This real-time system will also be a solution to our project's power requirements. Zephyr features power timer assimilation. Offering scheduling for certain tasks will help with power saving. These tasks can include deep sleep when the tag is not being used. Intermediate signal broadcasting with the switching of sleep states will be beneficial to our power constraints. In comparison to the standard bare metal programming style, we think the application of Zephyr RTOS will be more beneficial to our project for the reasons mentioned above. The only downside our group foresees is the complexity of programming the controller in comparison to the normal super loop. Figure 5-24 shows the components Zephyr is grouped with.

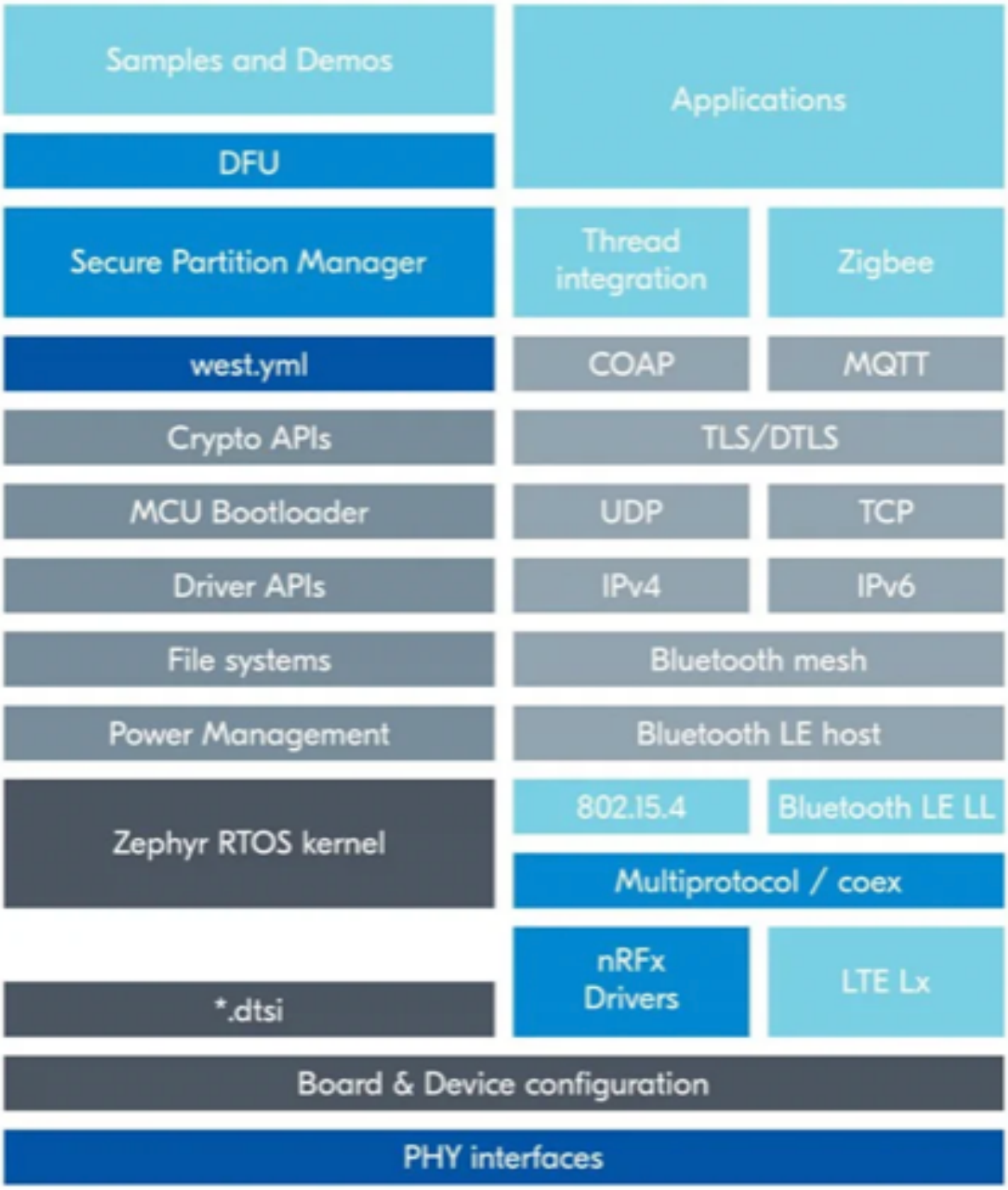


Figure 5-24: Connect SDK Components

Usage Approved by Qorvo (see appendix)

5.8.3 DWM3000 API

The DWM3000 API is a low-level functioning driver for the DWM 3000-series ultrawideband ICs. With the integration of a microcontroller host, we can access the features of our ultrawide-band device directly through SPI. It offers a simplistic utility for the development of ultrawide-band ranging in this project without having to code from scratch.

That being said, there are some discrepancies between the use of Qorvo's DWM3000 API and Nordic's nRF Connect SDK. The DWM3000 API was originally intended for bare-metal and FreeRTOS systems. Since the release of Nordic's nRF Connect SDK, they have moved to the Zephyr SDK previously discussed. Qorvo has no plans of implementing this API to the nRF Connect SDK and has stopped updating it.

Our team has instead decided to focus on porting the driver. Most functions and examples have already been ported to the Zephyr-based solution we plan on using for this project. The driver has been mostly ported by users on GitHub. From these open projects, we will use their repositories to solve the issue of legacy code.

5.8.4 OpenHayStack

For the groundwork in deciding the ultrawide-band tag engineering requirements our group originally wanted the finder to be a part of Apple's Find My Network. The Find My Network offers increased functionality for the user. Unfortunately, while Nordic does offer an SDK solution for implementing the Find my Network features to our chosen nRF 52833 microcontrollers, they only allow access to this SDK to those with an MFi license. This license is only available from Apple and only given to manufacturers that are already in the process of creating a device.

Given that our group has a financial budget that would not be able to cover the licensing, we have opted for an open-source solution provided by Seemoo-lab. OpenHaystack allows any Bluetooth accessories to be tracked by Apple's Find My network. It's a framework we can use to integrate our ultrawide-band tag onto the Find My Network.

Due to the complexities of combining this open-source framework into our existing code, we are pursuing this feature with caution. Additionally, OpenHaystack is too experimental to consider a primary engineering requirement of this project. Successfully porting the firm would still mean a fixed public key that our device will broadcast and allow other devices to track it. We see this as a far-off goal to show that our tag device could a part of the Find My Network given the financial income to afford an MFi license.

5.9 Mobile Application Design

This sub section involves details of mobile application design.

5.9.1 Chosen Operation System

As for our chosen operation system, we have chosen to use a framework that will be compatible with the new IOS 16 Apple operating system. This operating system is the latest that operating system that they have developed and published, and it uses a more advance programming that allows the UWB signal from the U1 chip to be captured more accurately. The IOS 16 offers a better stability and has some enhanced features. Since it is now the main operating system that Apple devices, from Apple 8 to Apple 14, our framework should align with the software constraints presented. In order do that, we will follow the interaction accessory protocol and MFi-certified UWB solution.

The new set of chips U1 are so accurate that they give the impression that they are GPS-like. They can also be used in many applications such as to turn on or off a TV, to turn on a light, radio, or more. However, we intend to use it only for localizing things that one's find necessary to attach a tag to, so if lost, they can find them. With the new IOS 16, the UWB is better and more accurate. These are all the reasons why we decided to be in compliance with latest operation system that our tag will run on because if the right framework is not used, our device will not be able to work and connect the Apple devices that support UWB and has U1 chipset. The framework that will be used to realize this project is called ARkit Framework and is described in section 5.8.3.

5.9.2 Nearby Interaction Framework

Nearby Interaction is a framework will be basing our most of our mobile application on. It lets our app get the position of Apple iPhone devices with the U1 chip. The framework is built upon an interaction between our ultra-wideband finder and the U1 chip equipped inside an Apple iPhone. When the app is running with this framework the user's smartphone shares their position and a device token that will be needed to identify the device. When triggered by the user it will report the direction and distance of our ultra-wideband tag in meters. Without the Nearby Interaction Framework, the ability to share the position of the U1 chip within our mobile application would not be possible. We will also be able to access other features in conjunction of the Nearby Interaction framework. Other frameworks such as the ARKit we will talk about later.

Having this framework is one of the reasons our group decided on using the Apple iPhone to implement into our project. Although this framework from Apple is still very new, they have been updating and putting out information of this framework consistently. Apple has a in depth documentation on their developer's site we plan to use extensively. Not only this we also will take note of Apple's design cues as

they are mentioned on what should and shouldn't be done using this framework. Figure 5-11 gives a visual representation of some of the data we will be able to use.

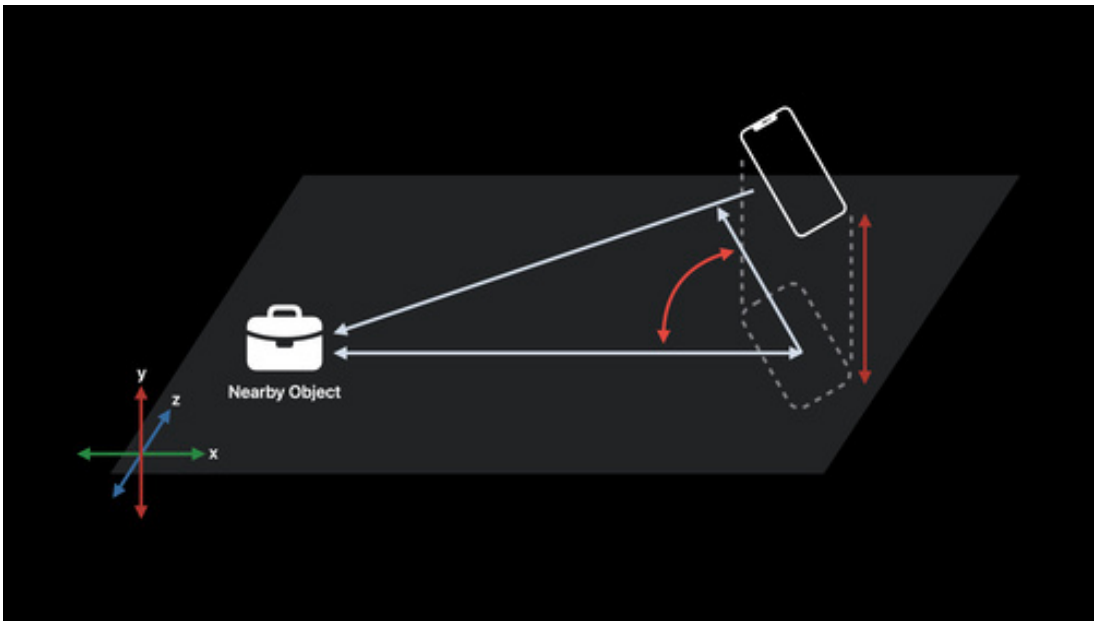


Figure 5-25: NI Framework Visual

Usage Approved by Apple Guidelines (see appendix)

5.9.3 ARkit Framework

ARkit is a platform for building augmented reality apps in 3 dimensional. It was first introduced with the release of IOS 11 and for development with XCode 9. To add to that the features of AR are only available starting with A9 chips or earlier versions of A10 and A11. Since the SDK is an Apple product the ARkit framework will be used to build games or apps for the iPhone or iPad. The way it works is by using the sensors or cameras built into the iPhone to analyze the world around the user. The framework provides many features, such as:

- Face tracking
- Location anchors,
- Plane detection,
- Scene geometry
- People Occlusion

The ARKit framework uses a Visual inertial odometry (VIO), this helps with the stabilization and accuracy. How the VIO works is by using a powerful framework called Core motion as well as collecting data from the camera. By using these two frameworks will result in a well detailed tracking of your position. What we can do

is integrate our location tag with the ARkit framework to create a whole new experience.

This is a very advanced milestone of ours, that would need more than a semester worth of work. What we like about is that it shows that our device can be implemented into different technologies. In the figure 5-12 shows a model of how the levels of what's needed for a whole AR application to be running smoothly. The ARKit is used as the processing level and that level uses capturing for the data

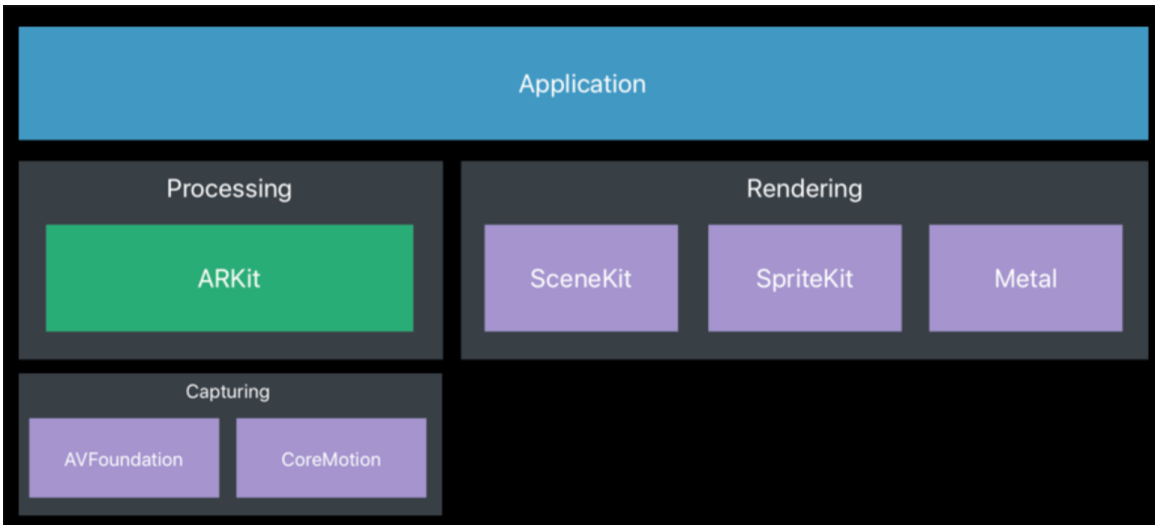


Figure 5-26: Schematic of DWM3001CDK

Usage Approved by Apple Guidelines (see appendix)

5.9.4 SwiftUI Framework

SwiftUI gives the user controls, layout structure and views in the app's user interface. This framework also gives input from the user, like event handlers for when the device is taped, or gestured, as well as tools to manage the flow of data gathered from the app. SwiftUI can be integrated with objects from the UIkit, APPkit frameworks to expand the functionality of the platform. Apple's main goal when creating SwiftUI is to have the developer build the user interfaces easily and rapidly. SwiftUI provides all the tools to allow this to happen

XCode being integrated into SwiftUI allows the developer to build the user interface at a rapid pace. You can edit the user interface in visual editor and in real time will reflect what your code translates to.

6. System Testing

This sub section entails device and mobile application testing.

6.1 Prototype Testing

Prototype testing of a product is an important step that is imperative to every developer or engineer to do. It will allow the developer to test the product and verify its specifications. Once these data have been collected, the group of engineers of developers that have been working on the project could verify if the product specification and requirement goals have been met. Prototype testing also helps to determine the product's working conditions based on its environment. It also helps determine if the product is defective or does not work as intended, or if everything works as it was anticipated.

Hence, we have determined that testing our device is important because it will allow us to know its potential and its flaws. Our testing will be conducted on many fronts, such as indoor testing, outdoor testing, antenna delay testing, and operational testing.

6.2 Testing Environment

Testing our device in various aspects will ensure that we deliver a quality product to our consumers, and also it will ensure that they have a quality experience while using the device. The testing environment test will be intended to evaluate the device's performance when it is used outside and inside. By operating these tests, we will be able to measure at what specific distance that either the Bluetooth or UWB signal will cut off or will not be visible by the receiver, which will be the Apple cellphone that will use.

Item	Conditions	Value	Units
1	Voltage testing	2.4 – 3.6	VDC
2	Voltage for programming calibration data into on-chip non-volatile memory	2.75 – 3.6	VDC
3	Test temperature	-40 - +85	°C
4	Inside testing	Range	meter
5	Outside testing	Range	meter

Table 6-1: Test Setup – General Conditions

Also, there are many testing that needs to be made on the Qorvo DW3000 chip to verify certain conditions, such as testing its nominal voltage that is intended to be used when the product is being used during normal operation. Once the voltage has been recorded, we will make sure we secure and store the data in the One-time programmable (OTP), which is a non-volatile memory. The nominal voltage generally must be between the value seen in table x-x above, from 2.4 to 3.6 VDC. Also, from the same table, we will test the voltage for programming the calibration data into on-chip non-volatile memory, which has its voltage ranging from 2.75 to 3.6 VDC. This step is required because it allows the DW3000 to be programmed on-chip OTP memory. In the case that our test reveals to us that our product design does not support that kind of technology, it will be imperative to us to have the calibration data stored in some other non-volatile memory in the product.

The temperature should also be tested and recorded because the product is said to be only operational when it is being used in temperatures ranging from -40°C (40°F) to +85°C (185°F). This step will be important to measure because some calibration parameters may vary depending on how hot or cold the device is. Once recorded, the results will also be recorded in the OTP or any non-volatile memory, so it knows what to do in these types of situations.

6.3 Indoor Testing

As it was mentioned in the product description sections, this device is was designed and created to help people to locate and find objects that they feel like could easily be lost or objects that could be important to them and that they do not want to forget when leaving their house or any location they could be in. A car key or house key, for example, are among the easiest things that one could lose and spend time looking for. This is an annoying and frustrating situation to be in, especially when that person has some type of urgency to go to work, hospital, meeting, or simply entering his or her home. However, having and using one of our devices will make sure that does not happen since it will be able to track and lead the user to the lost item location.

That is the reason why our group made sure to test how well and easy it would be to find an item, both in indoor and outdoor conditions. Even though indoor testing seemed to be the easiest and less challenging test that our group will be doing because of the short distance there will be between the device and the receiver, it sure has its level of difficulties to find the object when it was in a confine space, such as behind a concrete wall, that blocks the signal or makes it hard to find. It was totally understandable because signal to not travel through walls wirelessly, unless it is hard wired to an anchor that will catch any signal past that wall. Since the device does not have a speaker integrated and relies solemnly on the Apple UWB and the tracking software that it uses to be found, it might take the user a relatively lower amount of time to locate the device, but level is only in optimal situation since house are not completely enclosed.

In normal indoor conditions, which users will be in 99% of the time, we were able to connect to the device faster by using the Bluetooth Low Energy signal and we were able to measure how far the object was by using the UWB signal as a reference point. Once the device UWB signal was found, it numerically showed the distance on the Apple phone, powered by the find my app software, from where the user was to the object that had the tracking device attached to it. We did were not only able to measure the distance, but we were also able to see it in real time, which means that the value, noted in meter, changes accordingly when the user moves. That real time functionality helps the user to find the object more accurately since its distance will decrease when the user is going towards it and increase if otherwise. Also, it will be able to let the user know if the object is under by showing negative value once the user is close enough to the object, or a positive value if the object is up.

6.4 Outdoor Testing

Once we were done with the indoor testing, our next step was to test the device's performance in an outdoor situation to see how it would respond as good as it did for the indoor testing. In a state like Florida, the temperature can sometime change drastically since it could go from being all shinny to rainy. These weather conditions could sometime be challenging because we did not know how the device would react to certain changes and how well it would function. The device is expected to work in extreme weather conditions because of its operating temperature ranges, which can go as low as -40 degrees Celsius and as high as 85 degrees Celsius. Even though the chip could support such temperatures, it has to be noted that a good functioning of the entire device does not only rely on the chip itself, but also all the components that are in the assembly, specifically the battery, which powers the device and that can only support temperature from -20 degrees Celsius to 80 degrees Celsius. In extreme heat, the battery could inflate and explode, and in extreme cold temperatures, the battery will most likely die out due to a fast discharge rate.

Testing the device in a open space was a little bit more challenging because the are so many factors that are being considered, such as the speed of the wind, if it is raining, the humidity level, and also when it is cloudy. It was a little bit trickier to test it outdoors because it was when we tested it the first time, it was about to rain and the Bluetooth had issue connecting to the device once it was at a certain distance, about 6 meters to be exact. The signal was going a little bit on a off because of the wind since it was taking away a portion of the signal. However, we were able to address that issue because we later realized that our casing was not seal sealed properly, which also contributed to the device signal issues. Since we want our users to have the best experience while using the device, we made some minor adjustments to the casing, which slightly improved the quality of the signal.

On the other hand, we had a much better results testing the device when the weather conditions were optimum, meaning that the test was done when it was

sunny, because we were to connect to the device and locate it every time the test was done. The application software was able to successfully connect to the device by using Bluetooth Low Energy signal, and it accurately showed how far the device was from us and lead us to the it by capturing the UWB signal.

6.5 Testing Antenna Delay

Testing the antenna delay is one of our testing priorities because it will allow us to know how long it will take for the application software to connect to the antenna of the DW3000 chip and connect to it. This type of testing is also important because it helps us, developers, to calibrate the antenna as needed, in case the end results is not as projected. It will also help our group to know the exact time that it will take for the anchor to capture the calibrated distance and send us results that will help us, either user or developer, to locate the specific location that the tagged object will be at. There are a total of two steps that need to be done in order to know in the distance is well calibrated or if some arrangements should be made.

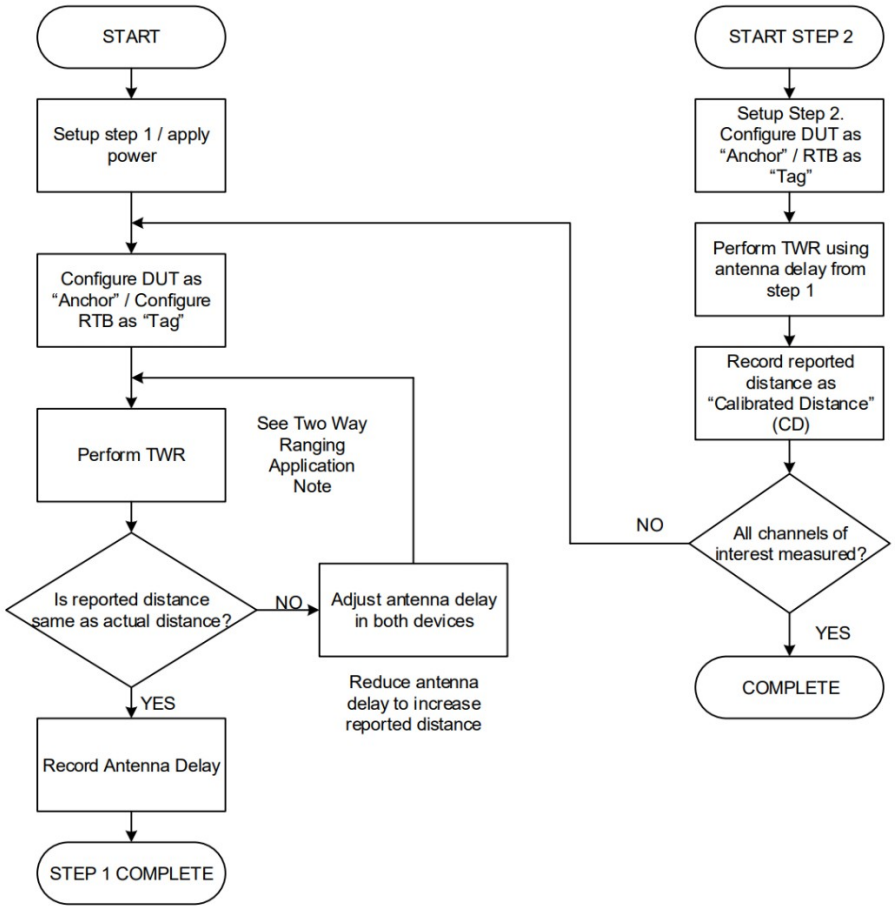


Figure 6-1: Qorvo Chip Antenna Delay Process

Usage Approved by Qorvo (see appendix)

In the figure 6-1 above, we can see a logical diagram that indicates the two levels that were mentioned in the sentences before, and it can also be seen all the processes that will take place before, both, the first step and second step, could be complete. For the first step, we will test the antenna using a free space measurement, which is a known distance, to calibrate the antenna. If the known distance is measured accurately, then it will proceed to step two, which ensure to insert the DUT and RTB data into the setup and repeat the process. The second step distance measurement will usually be larger than the one tested in the first step; however, that difference is a reflection of the additional delay that took place in the second step. That is the reason why it is called calibration distance. If that calibration distance is not sent, the antenna shall be calibrated.

To start the first step, a DC voltage will need to be applied to the chip, which will power all the components within the chip, including the antenna. Then, as it can be seen in the figure above, the DUT will be configured as anchor and RTB as tab, and after that process, a two-way ranging (TWR) test will be performed. Once performed, the distance should be reported. In the scenario that the distance was actually not reported, it will go back to the two-way test, and once it successfully measured the distance, the next stage is to record the antenna delay, which will complete step 1 and go to step 2.

In step 2, the process is kind of redundant, but it is needed for the accuracy of the device and also to measure all the channels of interest. Just like in step 1, the DUT (device under test) as an anchor, and the RTB (reference test board) as tag, then the TWR using antenna delay from step 1 will be performed, and just like in step 1 it will record a new distance referred as “calibrated distance (CD)”. If all the channels have been measured, step two will be complete, but in the case that some other channels of interest are still needed, it will go back to step 1 and do the processes all over again.

6.6 Operational Testing

The operational testing of our product is a very important part of all the testing that we must do. It is there to test many functions of our product and make us aware beforehand if there might be a malfunction in our product or else. According to the Qorvo production tests prior to checking, that is mentioned in table 6-2, there are a total of seven tests that are important to the development team to operate. These tests are to be done to ensure that the product works as intended and that the right values have been captured or measured.

The first test is the idle mode current consumption, and its values should range from 10mA to 13 mA. The second test is the SPI test, which can only have two outcomes, it would either fail or succeed. This test, just like the first test, is required per DUT basis. Moreover, for the third test, we will measure the current consumption of the device that must be from the range of 30mA to 90mA. Then we will test the GPIO (general-purpose input/output) strobe using SPI (serial

interface) commands to verify that all GPIO, IRQ, and SYNC line are asserted. Then, we will perform the reset test to verify if the reset line goes high after power is applied, and we will also test the WAKEUP function, which usually help the device to bring the device out of sleep, this process will be useful when the user will try to locate their item because, when not used, our device will go to sleep, and testing to make sure that this function works is a great priority, for the users could enjoy the product. Sleep mode current is generally less than 1μA, but once wake up the value should be from 10mA to 13mA. Finally, the EXON test, but it will only be done if necessary.

Test	Test Name	Test Description	Limit	Test Application
1	Idle mode current consumption	Apply supply voltage to product and measure current after 10ms	Min 10mA Max 13mA	Recommended on a per-DUT basis
2	SPI Test	Run SPI Test	Pass/Fail	Recommended on a per-DUT basis
3	Current Consumption	Configure DUT as transmit or receive	Min 30mA Max 90mA	Recommended on a per-DUT basis
4	GPIO Strobe Test	Use SPI commands to run GPIO strobe routing	GPIO,IRG, and SYNC	Only necessary if used in the product
5	RSTn Test	Check that the reset line goes high after power has been applied	Reset line goes high	Recommended on a per-DUT basis
6	WAKEUP test	Using SPI commands to set GPIO device into sleep mode. Assert WAKEUP pin to bring device out of sleep	Same as test 1 limit, but should be <1μA when in SLEEP mode	Recommended on for WAKEUP functionality
7	EXTON Test	Check that EXTON line goes high after power is applied	EXTON line goes high	Only necessary if used in the product

Table 6-2: Production Test Prior Checking from Qorvo

Usage Approved by Qorvo (see appendix)

6.7 Mobile App Testing

As soon as the mobile application prototype has been completed, the design will need to be tested. Since the mobile application is an essential part of our project, there will need to be many trials to see if the app is functioning properly. This section will specify the methods that will be used to test the mobile application after it is fully completed. The mobile application has a series of requirements that it must execute. In order for the system to succeed, it is crucial for the requirements to be properly specified and tested.

Included in the prototype design of the mobile application, there are development environments that supplement the procedure for testing. The development environment includes ways to determine that the integrated parts of the system are working properly and efficiently. This development environment will help decrease the time spent on the testing and debugging process. When testing the requirements, it is important to note that each requirement must be tested individually so that we can be certain that part of the application is running correctly. This will speed up the process of identifying any malfunctions and inform us of where the code needs to be modified. Testing multiple requirements at once can lead to many malfunctions and leave many doubts on where to start fixing the code. The debugging code cannot interfere with the completed application code. The debugging code should be able to be removed without having a negative effect on the system or stop the completed application code from functioning properly.

For the system to run successfully, the mobile application must correctly search for the Bluetooth signal so that the device can connect to the smartphone. Once it is connected, the next task to be tested is the connection of the ultra-wideband between the device and the smartphone. As soon as the user grants permission, the mobile application should begin the ranging process. If our application design can complete these two major requirements, the system can function properly in trying to detect the range between the user and the device.

6.8 Software Specific Testing

One of the most important aspects of this project is software. The software that will be implemented in both the device and mobile application is crucial in allowing the ranging process to be successful. To make sure that this process is running efficiently, the software component of this system must be tested properly and accordingly. The most crucial part of the ranging process is connecting the smartphone and tag device via Bluetooth technology. Device testing for measuring Bluetooth connectivity is key in order to ensure communication with no noise or interference between the two devices. The environment is very important to consider when testing the communication between the two devices. The environment must be controlled and isolated from any unwanted interference. For any IoT application to successfully run the communication requirements mentioned below must be met.

The maximum throughput must be measured. The maximum throughput is measured in bits per second, and it is responsible for defining the rate of data successfully transmitted over a channel. The formula that can be used to determine the maximum throughput is R=D/T. R represents the throughput rate, D represents the data inventory, and T represents the time of completion. The value of D can be retrieved by multiplying the number of packets that are captured. This will determine the quantity of data captured on the receiver end. T will be the time taken for the receiver to capture the packets. RTT stands for Round-Trip Time. Once the entire communication cycle is completed, the RTT displays the time allotted for the packet to be transferred from the instant transmitter to immediate receiver after receiving the acknowledgement.

The One-Way Delay shows the amount of time the packet traveled to get to the receiver end. The One-Way Delay can be found by calculating the difference in the time the packet took to depart from the transmitter to the arrival at the receiver. When measuring the One-Way Delay, it is important to synchronize the clocks from the transmitter and receiver in order to yield an accurate result.

The connection establishment latency is another important requirement that should be tested. The connection establishment latency displays the amount of time it took for the connection to be established so that the Bluetooth link is set to transfer data.

The uncontrolled factors must be taken into consideration when testing Bluetooth communication. Although these factors are difficult to deal with, it is still important to understand how these factors will impact the outcome of the system. One of these factors is the environment noise. The signal noise that comes from the environment is natural and can cause severe packet losses. This environmental noise can break down the link capacity or increase the packet delay.

Another factor to consider is system load. The testing results can be heavily affected if the system load is high when performing tests. This occurs because factors like speed, stability, reliability, and scalability are dependent of the system load.

To summarize, Bluetooth technology will be an integral part of our design. The testing for this technology will be intensive since it is one of the most crucial parts for range finding. There are many factors that need to be considered when testing the Bluetooth technology. The maximum throughput will need to be measured properly and it is measured in bits per second. The Round-Trip Time is another factor that we will need to be aware of since it will show the amount it took to complete one communication cycle. One-Way delay will display the amount of time it takes for the packet to reach the receiver end. The uncontrolled variables will be very important to consider when testing this technology. Understanding the uncontrolled variables will help make necessary adjustments to design a successful system.

7. Administration

The following section will discuss budgeting, financing, and scheduling.

7.1 Budget

Creating a budget is imperative for any project. The budget for a project dictates what kind of resources will be used or what type of procedures will be applied. Poor estimation, miscalculation, or judgement can cause problems for a project. The results of poor estimation are often negative consequences. Miscalculations can lead to poor results and cause the project to not meet the deliverables that were agreed upon. This can lead to negative effects on credibility and future projects. Table 7-1 is shown below which will list the parts that will be needed for this project along with the costs.

Part	Cost	Quantity
iPhone 13	Previously Owned	1
3D Printing Filament	$21.99	1
Wires	Previously Owned	N/A
Qorvo DWM3001CDK	$19.51	1
Nordic nRF52833DK	$49.00	1
Qorvo DWM3000EVB	$19.50	1
Nordic Power Profiler II	$92.50	1
Segger J-Link EDU Mini	$125.80	1
Supplies/Tools	$30.00	N/A
Total Amount	$358.30	

Table 7-1: Project Budget List

7.2 Financing

The financing aspect of the project will be handled by the group members. The budget will consist of the total amount spent divided equally between the four members of the group. The total amount for this project is currently $358.3 which will be $89.58 once the amount is divided between the four group members. Most of the parts that will be used for this project are not too expensive and some parts are already obtained by some of the group members. Since the budget for this project is not a large one, there will not be a need to receive any external funding.

7.3 Milestone Timeline

The milestone timeline was created to show the events that took place in order to create the project. The timeline shows a description of the work that was done and the week that the work was completed. Table 7-2 demonstrates the milestone timeline below.

Senior Design I	**Week**
Group Formation	Week 1
Project Brainstorming	Weeks 1 - 2
Project Idea Proposals	Week 3
UWB Investigation	Week 3
Microcontroller Investigation	Week 3
Divide & Conquer V1	Week 3 – 4
Divide & Conquer V1 Meeting	Week 5
Project Revisions	Week 5
Divide & Conquer V2	Weeks 6 – 8
Order Prototyping Parts	Week 7
Continue Project Documentation	Weeks 9 – 12
Finalize Paper	Weeks 13 – 16
Fall Break	N/A
Senior Design II	**Weeks**
Group Regathering and Project Coordination	Week 1
Begin Prototyping	Weeks 2 - 4
Test and Debug	Weeks 5 – 10
Final Revisions and Document Submission	Weeks 10 – 13
Final Project Presentation	Weeks 13 - 16

Table 7-2: Milestone Timeline

8. Conclusion

In closing, the plan for this project is to design a small device such as a finder or tag that can be used to locate an object that was misplaced, forgotten, or stolen. The device that we are trying to create is meant to be placed on objects like keys, bags, wallets, or equipment. Some users can even use this device to place on their pets so that they can be located if they were ever misplaced. There will be a smartphone that is going to be used to guide the user to the desired object which will be tagged by our device. There were many different tools, parts, and strategies that the team came up with to create a good system that will help us build our device.

Consumer Ultra-Wideband modules are capable of precision tracking which allows tags to be located at an accuracy of 10 centimeters. Ultra-Wideband communication was introduced a while back but was not utilized until recent years. Apple was one of the first companies to use Ultra-Wideband technology to track objects like phones, tags, laptops, and earphones. Other companies followed in the footsteps of Apple and used Ultra-Wideband technology for the same concept. From Apple, the phones that contain the Ultra-Wideband are the iPhone 11 and any phone after that. The smartphone that our group decided to use to test out our device is the iPhone 13. This decision was made to save time and to make our project a little more cost effective. Another technology that will be utilized is Bluetooth Low Energy. The plan is to use this technology to tell the device that the user is in range. Once the tag device is connected to the user's smartphone, the device may start the ranging communication process using Ultra-Wideband.

In order to create our design, our group made use of many software components. To build the printed circuit board, our group decided to use Autodesk EAGLE. Since members in the group have educational access to EAGLE from previous projects, the group thought it was a good idea to reduce costs. If not for the educational access, the group would have to pay $60 to get full access to EAGLE. Without full access to EAGLE, the program will not allow the programmer to use a larger plane to work with. When it comes to programming languages, our group decided to use C, C++, and Swift. The C and C++ languages will be used to program the embedded systems in our device such as a microcontroller. The Swift programming language will be used to design code for the mobile application. The programs the group will use that contain C and C++ are Xcode and SEGGER Embedded Studio. Xcode will also be used to write the code in Swift for the mobile application. Programs like GitHub and Microsoft VScode will be used for fixing and debugging the codes written for the embedded systems and mobile application.

After discovering that in order to use the Find My Network, one must have an MFi license, the group came up with an alternative software program to use called OpenHayStack. OpenHayStack is an open-source software that allows any Bluetooth accessory to be located using Apple's Find My Network. The final

software program that will be used for this project is the DWM3000 API. The DWM3000 API is a low-level driver for the DWM 3000-series Ultra-Wideband integrated circuits. This program will allow us to access the Ultra-Wideband features of our device via SPI. This program will save us time in having to write a completely new code for the Ultra-Wideband module.

The design for the tag device incorporates many intriguing hardware components as well. As mentioned above, the tag device will make use of a microcontroller. The microcontroller that our group decided to go with for our design is the nRF52833 by Nordic. The specifications of Nordic's nRF52833 aligned with the features we were looking for in a microcontroller. The nRF52833 microcontroller offers 512KB of flash and 128KB of RAM. Another hardware component that we will make use of is a DC/DC converter. The DC/DC converter will be used to increase or decrease the voltage coming from our power supply so that it may be suitable for the microcontroller. The DC-to-DC converter that the group decided to use is the Torex Semiconductor XC9282. This DC/DC converter will be used to decrease the voltage so that it may be suitable for the Ultra-Wideband integrated circuit. For our power source, the group decided to use a button cell battery because it is cost effective and consumes less energy compared to other batteries. The button cell battery that the group chose for our design is the CR2032 Duracell lithium cell battery. To build the external case of the device, the group wanted to use a 3D printer and chose ABS as the 3D printing filament. The group chose this filament because it is one of the more popular kinds of filament which means that it is effective and easy to use. The Qorvo DWM3001CDK is another hardware component that the group decided to incorporate in our design. The Qorvo DWM3001CDK is a fully integrated Ultra-Wideband module. This module will allow us to design a real time location system via UWB as a Two-Way Ranging or Time Difference of Arrival tag.

The group used many different methods and ideas to make sure that the design would be completed by the deadline and the requirements would be delivered. The group had weekly meetings to see where each group member was at in their contributions for the project. Sometimes these meetings were used to check up on each other and see if a member needed a little more help with their work for the project. When one of our group members could not make it to the scheduled meetings, the rest of the group would work on providing the information that was shared during that meeting. We used the Discord app to communicate and help each other out when needed. The group also used the online Word application to write the document. This way we were able to see each other's work and make adjustments where they were necessary. The milestone timeline was created by the group to make sure that we were on top of the work that needed to be done and so that we met every deadline. With all of these steps that were taken to create an effective tagging device, we hope everything goes well and have a successful project.

Appendix

Permissions

Hello,

I am currently a senior computer engineering student at the University of Central Florida. My team and I are working on a capstone project that involves a finder device that uses ultra-wideband. We have been utilizing the DWM3000, DWM3001C, DWM3001CDK for our project along with all the documents provided.

We have been using schematics, diagrams, and pictures from these documents and would like to ask for permission to use it in our paper. I would just need a reply to this email confirming the use of your company's material if Qorvo will allow it.

Thank you,
Anthony Venkersammy

Hi Anthony.

It should be OK, but I am checking with one of our Technical Marketing Managers to be sure.

I should hear back tomorrow.

David Oliver
Manager, America's Field Applications Engineering
Qorvo – Power, Networks & Defense
Broomfield, CO
E david.oliver@qorvo.com
O +1 303.464.2422
M +1 303.204.5210

DESIGN TOOLS Qorvo Design Tools

Core technologies for everything that connects the world at qorvo.com

Hi Anthony.

Could you attach the documents that you'd like to take the schematics, diagrams, and pictures from?

I'd like to come to faster resolution internally at Qorvo.

Thanks.

David Oliver
Manager, America's Field Applications Engineering
Qorvo – Power, Networks & Defense
Broomfield, CO
E david.oliver@qorvo.com
O +1 303.464.2422
M +1 303.204.5210

DESIGN TOOLS Qorvo Design Tools

Core technologies for everything that connects the world at qorvo.com

Hi David,

sure no problem.

Thank you,
Anthony Venkersammy

Hi Anthony. Where did you find the quick start guide?

I see that the others can be downloaded from our website.

David Oliver
Manager, America's Field Applications Engineering
Qorvo – Power, Networks & Defense
Broomfield, CO
E david.oliver@qorvo.com
O +1 303.464.2422
M +1 303.204.5210

DESIGN TOOLS Qorvo Design Tools

Core technologies for everything that connects the world at qorvo.com

all around you

David,

Here's the link from the Qorvo website.

DWM3001CDK
qorvo.com

It will be under the documents tab, then user manual category.

Go for it! Not sure how I missed that.

David Oliver
Manager, America's Field Applications Engineering
Qorvo – Power, Networks & Defense
Broomfield, CO
E david.oliver@qorvo.com
O +1 303.464.2422
M +1 303.204.5210

DESIGN TOOLS Qorvo Design Tools

Core technologies for everything that connects the world at qorvo.com

all around you

Product Name : Apple Developer Support

Support Category : Development and Technical
Support Topic : Other Development or Technical Questions

Additional Details :

Message:

Hello,

I am currently a senior computer engineering student at the University of Central Florida. My team and I are working on a capstone project that involves a finder device that uses ultra-wideband. We have been utilizing the Nearby Interaction framework, ARKit framework, and SwiftUI framework for our project, and all information provided from Apple's website.

I would like to ask permission to use the diagrams, pictures, and documentation in our paper. I would need a reply to this email confirming the use of your company's material if Apple will allow it.

Thank you,
Anthony Venkersammy

Hello Anthony,

Thank you for contacting Apple Developer Program Support. My name is Tanekia, and I will be happy to assist you today.

I understand that you would like to use WWDC material and Apple documentation. You are free to use these based on our guidelines.

See Guidelines for Using Apple Trademarks and Copyrights.

Contact Apple Legal with any questions you have about the use of Apple trademarks.

Please let me know if you have any follow up questions. You can also contact us using the link:

https://developer.apple.com/contact/

When contacting us, please reference your Case ID: 101874053031

Best regards,

Tanekia
Apple Inc.

References

[1] “Apple AirTag Teardown | TechInsights,” *www.techinsights.com*. https://www.techinsights.com/blog/apple-airtag-teardown

[2] “OpenHaystack,” *GitHub*, Jun. 24, 2022. https://github.com/seemoo-lab/openhaystack

[3] “Apple AirTag Reverse Engineering - Adam Catley,” *adamcatley.com*. https://adamcatley.com/AirTag.html#background (accessed, 2022).

[4]“Zigbee - Nordic Semiconductor,” *www.nordicsemi.com*. https://www.nordicsemi.com/Products/Bluetooth-Direction-Finding

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

[6] “Nordic Semiconductor Infocenter,” *infocenter.nordicsemi.com*. https://infocenter.nordicsemi.com/index.jsp?topic=%2Fps_nrf52840%2Fkeyfeatures_html5.html (accessed, 2022).

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

[8] “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).

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

[10] “Brief Comparison of BLE5.1 nRF52833 and nRF52840 Parameters,” *www.linkedin.com*. https://www.linkedin.com/pulse/brief-comparison-ble51-nrf52833-nrf52840-parameters-%E6%83%85-%E4%BD%95 (accessed, 2022).

[11] “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).

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

[13] “Nrf-connect sdk or nrf5 SDK for DWM3001C,” *Qorvo Tech Forum*, Sep. 13, 2022. https://forum.qorvo.com/t/nrf-connect-sdk-or-nrf5-sdk-for-dwm3001c/12473/11 (accessed, 2022).

[14] “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).

[15] “BATTERYHOLDERS.COM an ISO 9001:2015 company.” Accessed, 2022. [Online]. Available: https://www.memoryprotectiondevices.com/documents/catalogs/MPD-Battery-Holders.pdf

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

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

[18] “The Ultimate PCB Design Software Comparison | SFCircuits,” *www.sfcircuits.com*. https://www.sfcircuits.com/pcb-school/pcb-design-software-comparison-guide (accessed, 2022).

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

[20] “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).