Semiconductor Fab Personnel Tracking

UTDesign Fall 2023 - Team 1682

Dorothy Bui
Bryce Canillo
Mitch Clark
Edy Gonzalez
Nahiyan Muzaffar
Remy Nguyen

Corporate Sponsor:
Dan Carothers

Faculty Advisors:
Dr. Neal Skinner
Dr. Marco Tacca

Instructor:
Dr. Neal Skinner

THE UNIVERSITY OF TEXAS AT DALLAS

Department of Electrical and Computer Engineering
University of Texas at Dallas
17 December 2023

Abstract

This abstract outlines the UTD Senior Design project for a tracking system to be implemented at STRIKE Photonics' semiconductor fab. This system will be critical for life safety and process management procedures during equipment failure, gas leaks, or other emergencies. STRIKE Photonics is initially seeking a method that is capable of tracking the location of people in designated zones of the fab, recording their position over time. This method should have the potential for future expansion to wafer tracking and other materials or resources.

To accomplish this, a tracking system was designed using ultra-wideband (UWB) technology. Data structures will be employed to identify the optimal paths and locations for the fabrication equipment and UWB solutions. The system will record the location of people in real-time, allowing emergency rescue personnel to identify if all zones have been evacuated in the event of an emergency. The UTDesign Studio was used for environment simulation, taking advantage of the existing objects and obstructions present to test and refine the solutions to ensure optimal functionality.

The success of this project will rely heavily on the accuracy and reliability of the tracking system. It is a critical aspect of life safety and process management procedures, making it essential to design and implement a system that meets the specific requirements of STRIKE Photonics. The proposed solution offers a practical and effective approach to track personnel in the semiconductor fab, ensuring optimal safety and process management procedures.

Acknowledgements

The project team extends appreciation to several individuals and organizations whose support significantly contributed to the success of our project. Foremost, we express gratitude to our professor and mentor, Dr. Neal Skinner, for providing invaluable information, practical advice, and insightful comments throughout the project's development. Special acknowledgment goes to our corporate mentor, Dan Carothers, for his supervision, insightful suggestions, and continuous support throughout the project's duration. The wealth of knowledge, guidance, and encouragement from our mentors played a pivotal role in making this project a reality. We extend thanks to the University of Texas at Dallas and Strike Photonics for organizing the opportunity to collaborate on this impactful project.

Table of Contents

List Of Images

List Of Tables

Chapter 1: Introduction

As semiconductor fabs become increasingly sophisticated, the need for robust safety measures and streamlined process management procedures becomes ever more crucial. Although thorough mandatory training is an important part of preparing individuals for safety purposes, unpredictable situations may arise in the event of actual emergencies. To prepare for these unpredictable situations, Strike Photonics is actively seeking a cutting-edge tracking solution that goes beyond conventional methods. The primary objective is to track the precise location of personnel within designated zones of the fab, recording their movements over time. In the event of an emergency, rescue personnel could use this system to quickly assess if employees are still present inside of the fab and deploy for rescue. This project addresses a critical need for an advanced tracking system within the semiconductor fab, with a primary focus on bolstering life safety and refining process management procedures during emergencies such as equipment failure, gas leaks, or other critical incidents.

Chapter 2: Review of Conceptual and Preliminary Design

2.1 Problem Analysis

2.1.1 Specifications

The main specifications for the design problem include that the system should accurately track personnel during emergencies. The system should also be operational during a full day of work, specifically at least 12 hours. Furthermore, the system should include a graphical user interface (GUI) that can access external servers and still function properly in the event of server disconnection, with a touch screen panel that is small enough to not interfere with other equipment but large enough to be visible with a mask on. Lastly, the system should possess a method to record and access time-stamped location history.

A notable feature in the design problem is a robust GUI that remains accessible at all times, with a particular focus on its usability during emergency situations. Along with the GUI, another pivotal aspect of the design focuses on providing easy access to location history. Strike Photonics aims to have comprehensive information regarding the whereabouts of personnel within a designated area. These dual features, the responsive GUI and efficient location history access, together form the cornerstone of addressing the specific needs and objectives outlined by Strike Photonics for their tracking system.

2.1.2 Technical Approach

The team approached the design problem by first addressing the hardware specifications by looking into products which utilize ultra-wideband (UWB), radio frequency (RF), Bluetooth, and other technologies. Research into these various technologies was done in order to compare performances, costs, ease of implementation between the different systems, and more. Ultimately, we decided that the best hardware options would be using UWB or RF technologies. For the software specifications, we decided that Python would be the best coding language to write our programs in due to its simplicity and its many resources available online. Additionally, we would look at any provided software programs that came with the hardware solutions and take that into account when selecting the hardware.

2.2 Decision Analysis

2.2.1 Alternative Solutions

The first solution that was considered included the use of an ultra high frequency RF (UHF RF) system consisting of an RFID reader, a RFID tag, and a UHF antenna. The team specifically used a SparkFun RFID Reader - M6E Nano, passive RFID tags, an Arduino Shield, and a UHF antenna. We proceeded to thoroughly test this solution, and the RF system did function. However, one of the main problems with the system was that the tracking was not reliable, as the receiver sensitivity was too low. This led to the tracking being very inconsistent and not practical for the design problem.A maximum nominal read range of 16ft was achieved with a 6 dBi linearly polarized patch antenna, however, the narrow beamwidth was not practical for creating a base station capable of scanning the entire bay within a reasonable amount of time.

Market research concluded that there exists a gap in the commercial space between low-cost, low-end RFID readers and large-scale integrated solutions for asset tracking which require quotations from third party companies. A design was planned to improve the RF front-end of the SparkFun Simultaneous RFID Reader by using a bidirectional amplifier with self-interference cancellation (SIC). More information about this design is outlined in Appendix D.

2.2.2 Effectiveness Criteria

In order to effectively compare all of our potential solutions against each other, we judged them based on a list of five categories and gave them the following weights:

1. Number of requirements met (25%)
2. Likelihood of success (20%)
3. Performance (20%)
4. Cost (15%)
5. Design simplicity (10%)
6. User accessibility (10%)

2.2.3 Decision Analysis

Since we were able to thoroughly test the RF system, we concluded that using RF was not the best solution, taking into account the cost and time needed to complete the design problem. With Dr. Skinner's help, we researched ultra-wideband and Qorvo's MDEK1001 development kits. We discovered that the DWM1001-DEV boards came with a software, Decawave, that displayed the tracking of a tag within an area of anchors. The cost for designing this system fit in our remaining budget, and we deduced that the duration of completing the system was sufficient relative to our remaining time.

ID	Criteria	Grading Scale (Max)	Weight	Score	Weighted Score
1	Number of requirements met	5	25%	5	25%
2	Likelihood of success	5	20%	4	16%
3	Performance	5	20%	5	20%
4	Cost	5	15%	3	9%
5	User accessibility	5	10%	4	8%
6	Design simplicity	5	10%	4	8%
Total			100%		86%

Table 1. MDEK1001 Kit Effectiveness Score.

ID	Criteria	Grading Scale (Max)	Weight	Score	Weighted Score
1	Number of requirements met	5	25%	5	25%
2	Likelihood of success	5	20%	4	16%
3	Performance	5	20%	3	12%
4	Cost	5	15%	3	9%
5	User accessibility	5	10%	5	10%
6	Design simplicity	5	10%	3	6%
Total			100%		78%

Table 2. SparkFun Simultaneous RFID Reader - M6E Nano Effectiveness Score.

ID	Criteria	Grading Scale (Max)	Weight	Score	Weighted Score
1	Number of requirements met	5	25%	5	25%
2	Likelihood of success	5	20%	3	12%
3	Performance	5	20%	5	20%
4	Cost	5	15%	2	6%
5	User accessibility	5	10%	4	8%
6	Design simplicity	5	10%	2	4%
Total			100%		75%

Table 3. ThingMagic Micro Embedded RFID Reader Module Effectiveness Score.

Chapter 3: Basic Solution Description

A fully wireless DWM1001-DEV Two-Way-Ranging Real Time Location System (DRTLS) was designed using Qorvo's MDEK1001 development kit, a commercial off-the-shelf (COTS) ultra-wideband solution. The system consists of several development boards, each fitted with an ultra-wideband transceiver. A given board is individually configured as an anchor or a tag using a preexisting application programming interface (API) via serial communication from a universal asynchronous receiver / transmitter (UART) terminal emulator. Each device is also given a matching Positioning and Networking Identifier (PAN-ID) which allows them to search for and connect to each other when in range and powered on. If a gateway, defined as a DWM1001-DEV connected to a Raspberry Pi 3B (RP3B) using the header interface, is in the network, it will host a web-based client on the Local Area Network (LAN) using its local Internet Protocol version 4 (IPv4) address. This web application, known as the DRTLS Gateway Web Manager, can generate a visualization of the connected devices and the position of tags relative to anchors with any PC or mobile device on the LAN. We will determine the optimal configuration for the network as well as the method(s) of communication to a central computer unit. In the software application, a map of the facility's floor plan can also be overlayed on the visualization so that each unit is accurately displayed on an intuitive map feature.

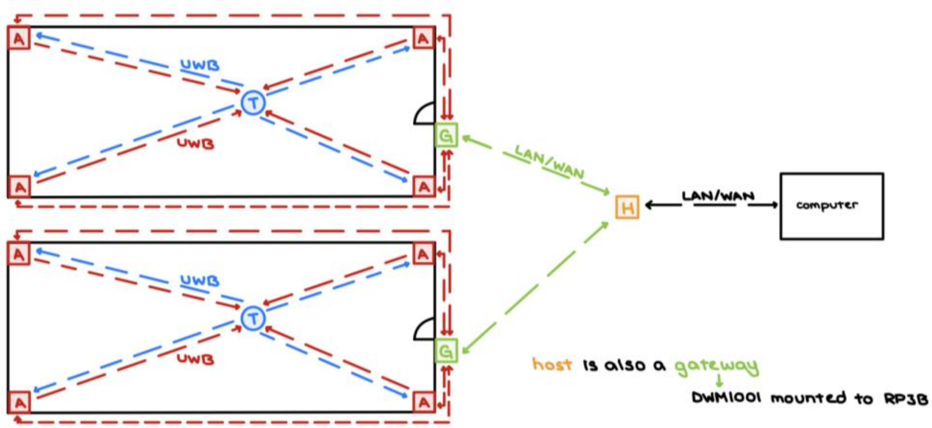


Figure 1. Real Time Location System communication architecture.

3.1 Tags, Anchors, and Gateways

Each board can be individually configured to assume a certain role in the system either as a gateway, an anchor, or a tag. Tags are the units which personnel in the fab will carry with them on their body. Anchors are stationary units which communicate with tags in an area to determine the position of such tags in an area. A minimum of 3 anchors must be able to communicate with a given tag in order to determine the tag's location in the area. The anchors then relay the information collection to a gateway. A gateway is a configured DWM1001-DEV board that is then mounted onto a Raspberry Pi 3B in order to extend the range of the system by communicating with other gateways on the LAN.

3.2 Batteries and Charging

The DWM1001-DEV boards can be powered via Micro-USB, but they do not hold a charge. Therefore we equipped units with lithium polymer (LiPo) batteries that could power a unit well throughout a full working shift (12 hours) as well as a switching circuit so that tags would not operate (and therefore drain power from the battery) while the battery is recharging.

3.3 User Interface

While connected to the same LAN that the system is set up on, a user can access a webpage using the IP address of the host gateway that can display the sample image below. Gateways will automatically scan for anchors and tags connected to the network and all units will be listed for view and additional configuration (if necessary). On the visual display, anchors and tags will be shown. Custom overlays can be uploaded for an accurate mapping of the environment.

3.4 Position Data Logging

Gateways continuously receive position data from tags on the network, and to access the position logs, they are sent from the gateway to the user's local machine for review. The Message Queuing Telemetry Transport (MQTT) connectivity protocol is used to interface the DRTLS to the cloud, enabling devices to publish or subscribe to various topics related to device localization, configuration, or other non-location IoT data. In this project, the MQTT protocol allows gateways to connect to a cloud-based broker, publish position logging data, and enable users to subscribe to topics and receive the data. To set up the MQTT broker, we wrote a Python program using the “paho-mqtt” library, specifying the IP address and port of the gateway. With the MQTT broker running, specific topics are subscribed to for each tag on the network, identified by the tags unique four-digit ID. The program will then output tag positions, quality, and superframe numbers in the terminal. And the final step involves formatting the output for user-friendliness, including date and time information, and saving it to a .txt or .csv file. The resulting output displays logged position data with relevant details.

3.5 Cases

Although there are a few open source models available for tag cases for this product, due to our novel solution, a new case would have to be created in order to properly house all the components that were used. This case is shown below in Figure 2. The anchors will utilize the plastic cases that they came with as part of the MDEK1001 kit, while the gateway will use an existing open source model for the case. An example image of the gateway and its case[1] is shown below in Figure 3.

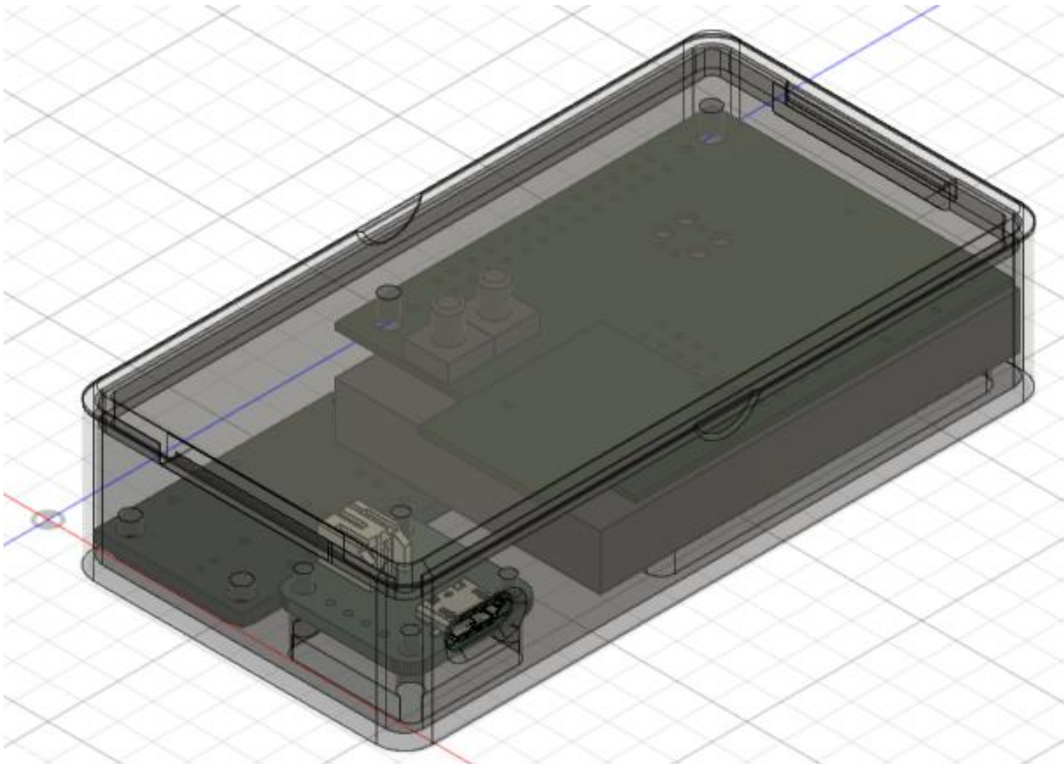


Figure 2. Tag case with simplified components.

[1] The image used to display the gateway and its case can be found at https://gitlab.com/Inria-Chile/Atelier-Inria/dwm-rtls/-/tree/master/enclosures?ref_type=heads.

Figure 3. Gateway case with components.

Chapter 4: System and Component Design

4.1 Overview

This section will outline the details and performance of the Decawave DRTLS as it pertains to the specifications of this project. The primary Decawave components are the DWM1001-DEV UWB module hardware and the Positioning and Network Stack (PANS) software. The DRTLS, along with our additions to it, are described in the following sections of this chapter.

4.1.1 Decawave DRTLS

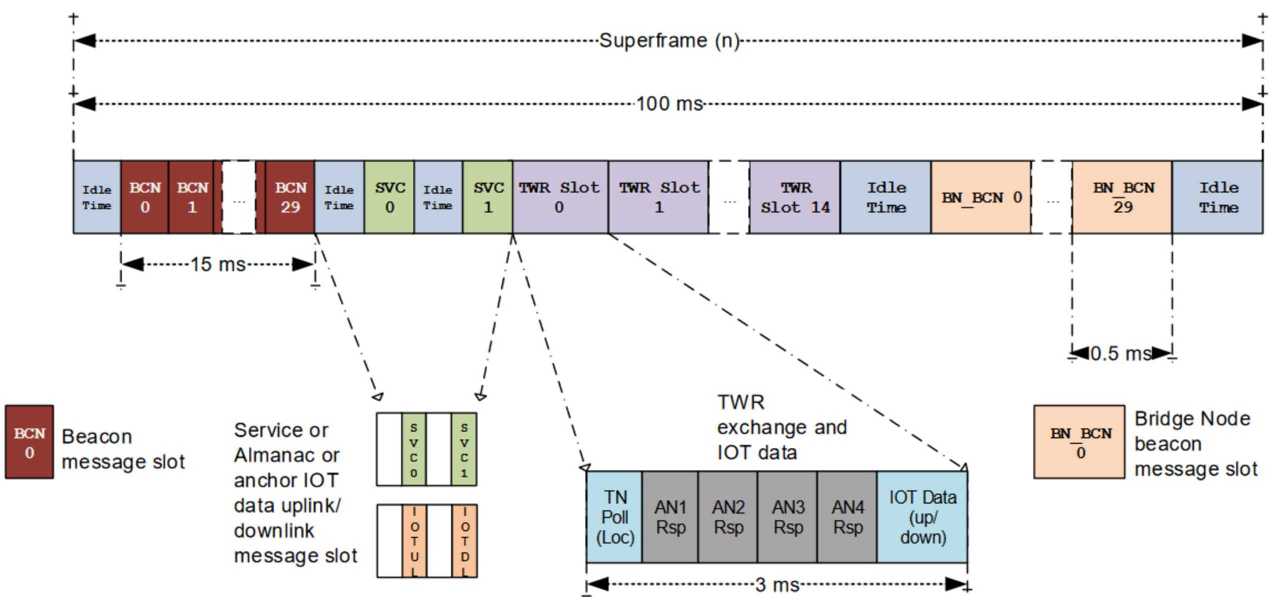


Figure 4. Superframe timing structure.

As described in Section 3.1, each DWM1001-DEV module can be configured as a tag, anchor, or gateway. Each bay in the fab must have a minimum of 4 anchors and 1 gateway, totalling 5 wall-mounted devices. One anchor will automatically be configured as an *Initiator anchor*; this is typically the first anchor that is powered on and it synchronizes the UWB clock timing. An accurized and synchronized clock is necessary because RF direction finding is performed via Time Difference of Arrival (TDoA) and because data transfer uses the Time-Division Multiple Access (TDMA) channel access scheme. The TDMA system utilizes repeating "*superframes*" of 100ms, each of which containing 30 *Beacon* (BCN) slots, 15 *Two-Way-Ranging* (TWR) slots, along with other service/utility slots and guard times, as shown in Figure 4. Each anchor in a communication cluster[2] will occupy a BCN slot while each tag will occupy a TWR slot. This implies if tags are configured to update every superframe (10Hz or

[2] A cluster is defined as the maximum area that allows for reliable UWB communications. If bay walls have sufficient attenuation, then each bay may be its own cluster. Note that smaller clusters allow for higher throughput per area.

100ms), then only 15 tags are allowed to communicate per cluster. Maximum and minimum update rate allows for 15 tags/cluster at 10 Hz or 9000 tags/cluster at 0.0167 Hz, however, this can be configured using IoT commands via the DRTLS Gateway Web Manager. For maximum efficiency, a dynamic update rate should be implemented that allows for maximum throughput depending on the system load by issuing IoT commands to the MQTT broker as done in Section 4.4.

In order to ensure reliability in all conditions, each bay should include its own gateway which will allow the DRTLS to function even if UWB communications cannot be established from adjacent bays due to signal attenuation. This is because the gateways communicate on the LAN using Ethernet or WiFi infrastructure. Note only one gateway will host the DRTLS Gateway Web Manager, however, it can be accessed by any web client on the same LAN. This gateway is defined as the proxy server host which should have high priority on the LAN as shown in Section 6.4; if this device fails, the DRTLS may still function although it will not be accessible on the LAN.

4.2 Switching Circuit

The DWM1001-DEV boards can be powered by Micro-USB but lack a means of holding charge. Anchors and gateways, being stationary, will use Micro-USB wall plugs to receive power. To enable mobility for tags, we connected LiPo batteries so that users are not tethered to a source of power. The MDEK1001 kit includes plastic cases for each board that are compatible with RCR123a or 16340 rechargeable batteries. We initially equipped our boards with 16340 rechargeable batteries with a capacity of 850mAh, but ultimately decided to use flat LiPo batteries with a capacity of 2000mAh for extended battery life and a slimmer form factor.

A previous version of DWM1001-DEV was equipped with a charging circuit so that the device could simply be plugged and charged via the existing Micro-USB port. Due to supply chain issues, the charging circuit is now omitted, necessitating the assembly of a separate charging circuit for the acquired version.

The circuit was designed such that the DWM1001-DEV is not drawing power the battery is charging. If the battery is powering a load, the measured voltage across the battery terminals will be lower than an open circuit measurement. The charging board stops sending current intermittently in order to read the battery voltage and estimate remaining charge, thus, if the battery is connected to the DWM1001-DEV at all times, the charging board will always read a lower battery charge than expected. This would cause overheating and premature deterioration of the battery. The switching circuit disconnects the battery from the DWM1001-DEV once connected to a charger via Micro-USB.

4.2.1 Components

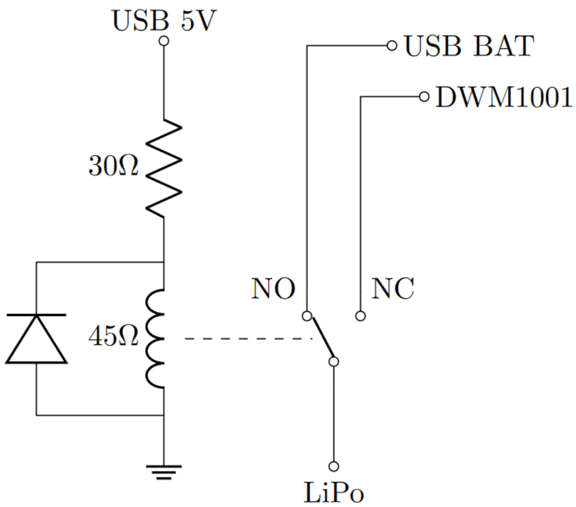


Figure 5. Relay-based switching circuit diagram.

Item	Quantity
J102K1CS33VDC.20 General Purpose Relay 3.3VDC Coil	1
30Ω Resistor ¼ Watt	1
1N4148 Fast Switching Diode	1
S2B-PH-K-S JST Connector	1

Table 4. PCB Component List.

4.2.2 Protoboard Testing

The switching circuit was initially built on a breadboard so that we could test our design for successful functionality. The DWM1001-DEV, a LiPo charging board, and the LiPo battery were connected to complete the wiring of a whole tag unit. When a Micro-USB cable is plugged into the LiPo charger board, a distinct *click* from the relay indicates proper function.

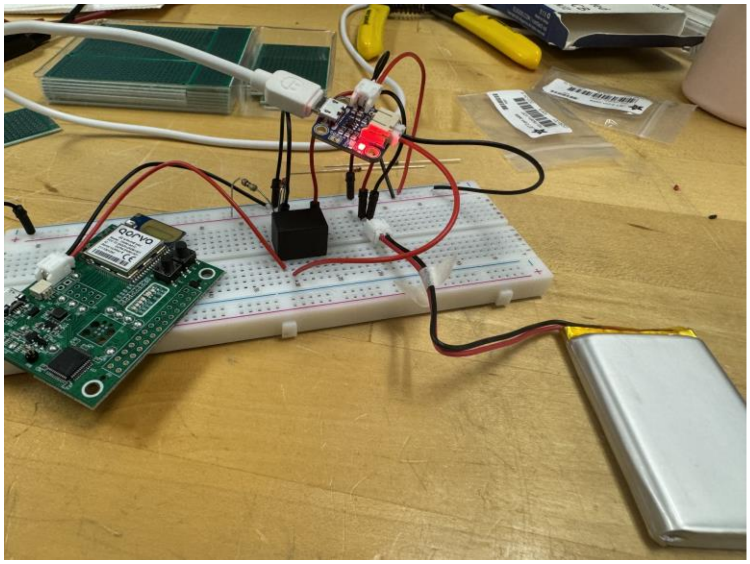


Figure 6. Assembly of switching circuit on a breadboard with connections to DWM1001, LiPo charger, and LiPo battery.

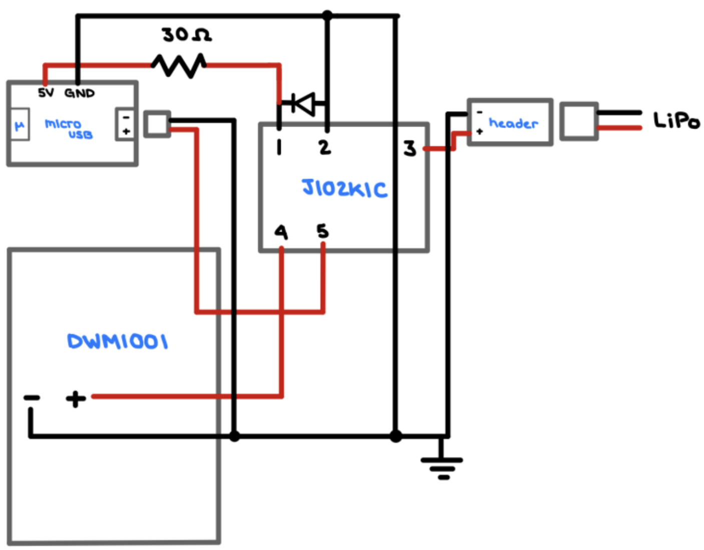


Figure 7. Charging circuit, switching circuit, DWM1001-DEV, and LiPo wiring diagram.

After building the initial circuit and confirming its functionality, the circuit was then replicated on a protoboard. The protoboard was trimmed of the excess so that all components could fit in a tag case. Initially, the plan was to use this implementation in our final product, but we encountered issues with our wires breaking as well as messy connections with solder dragged across the backside of the board.

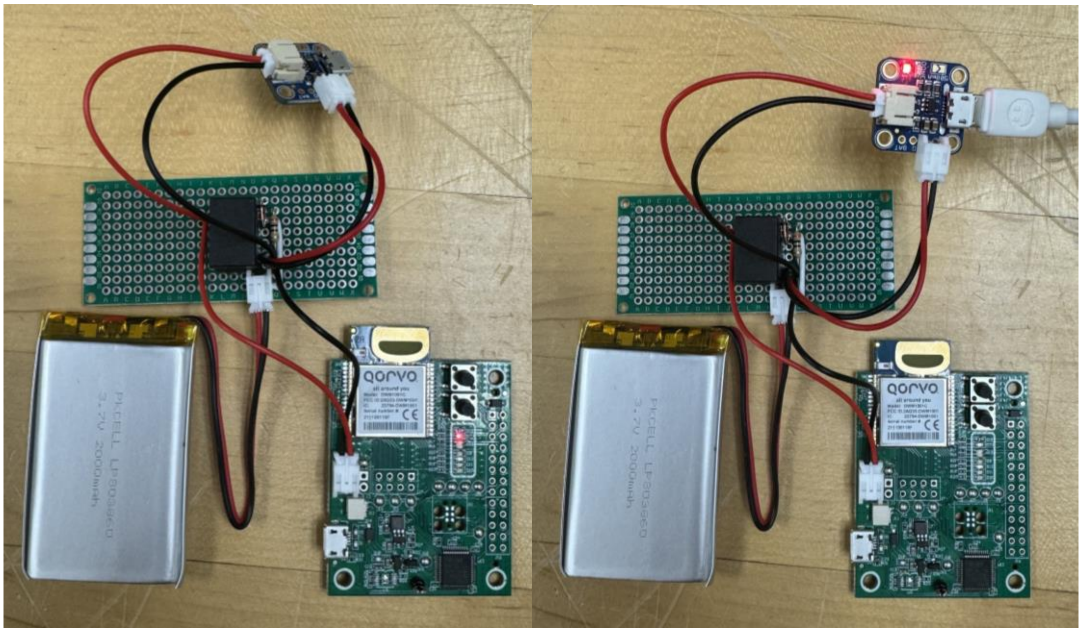


Figure 8. Assembly of switching circuit on a protoboard with connections to DWM1001, LiPo charger, and LiPo battery. DWM1001-DEV is on when not plugged in for charging (left) and is off when connected for charging (right).

4.2.3 Printed Circuit Board Design

In order to ensure uniformity and reliability in all conditions, a custom PCB was designed that would fit in the 3D printed case. With the circuit assembled on a PCB, more secure connections could be made between the components and drill holes allow for mounting hardware implementation in the case.

When picking components, the resistor and diode must be picked based on the relay coil voltage and current ratings. The purpose of the resistor is to clamp the voltage across the coil to its turn-on voltage based on the nominal 5V USB input voltage and the coil's DC resistance. The flyback diode parameters must have a reverse breakdown voltage greater than the coil switching voltage and a peak current rating greater than the coil's continuous on-state current. The circuit may function with a 0Ω resistor and no flyback diode, although it will be power inefficient and cause a back-EMF when the coil is disconnected.

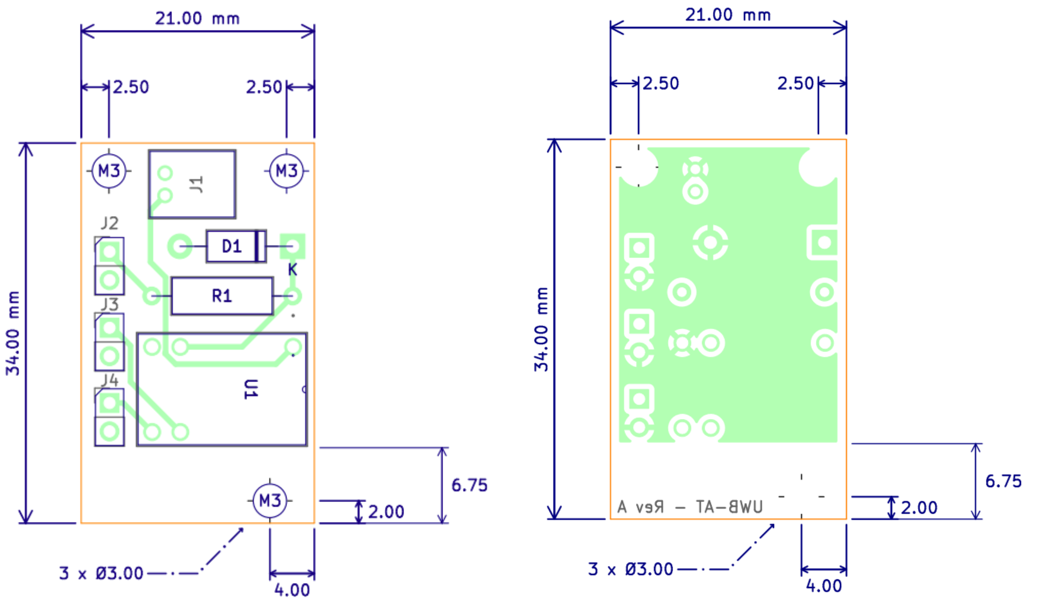


Figure 9. Assembly drawing of the PCB.

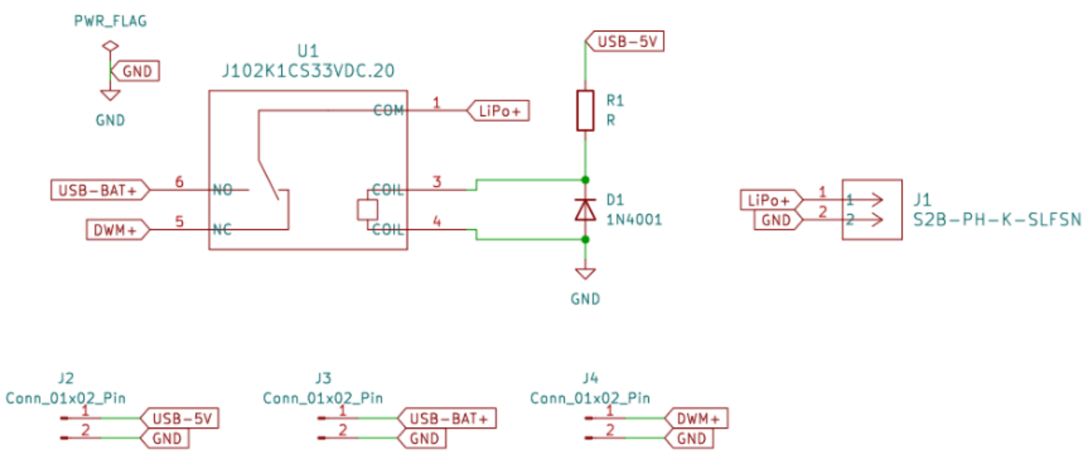


Figure 10. Schematic of the PCB.

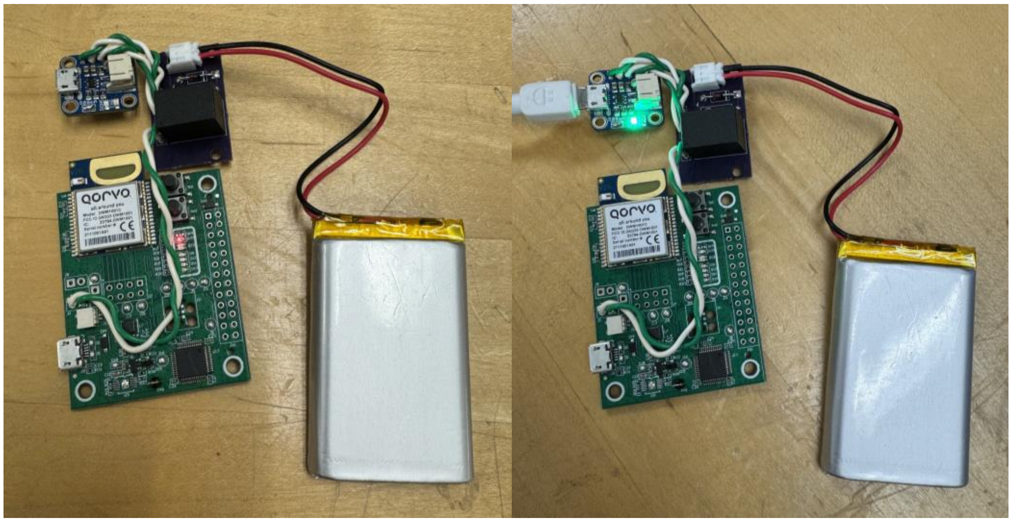


Figure 11. Assembly of switching circuit on a PCB with connections to DWM1001, LiPo charger, and LiPo battery. DWM1001-DEV is on when not plugged in for charging (left) and is off when connected for charging (right).

4.3 Cases

In order to properly contain all the components, a new case needed to be designed. The goals of this case were that there should be easy access for charging, it should be simple to open and close the case in order to access the components, and the case should be as compact as possible. The cases feature a snap-fit lid with areas to grip which makes it easy to open and close the case. To keep certain components in place, cylinders were created to fit in the mounting holes of the respective components.The dimensions of the case are 92.5 mm long x 48.5 mm wide x 20.5 mm tall. Seen below in Figure 12 are the 3D models of the two pieces of the case, while Figure 13 showcases all the components housed in the 3D printed case. The case for the gateway must be able to contain a DWM1001-DEV that is mounted on top of a Raspberry Pi 3B. We were able to find an open source model for a gateway case which met all of our needs. The models[3] for this case can be seen in Figure 14.

[3] The models for the gateway case can be found at https://gitlab.com/Inria-Chile/Atelier-Inria/dwm-rtls/-/tree/master/enclosures/STL_files/Gateway-RPI?ref_type=heads.

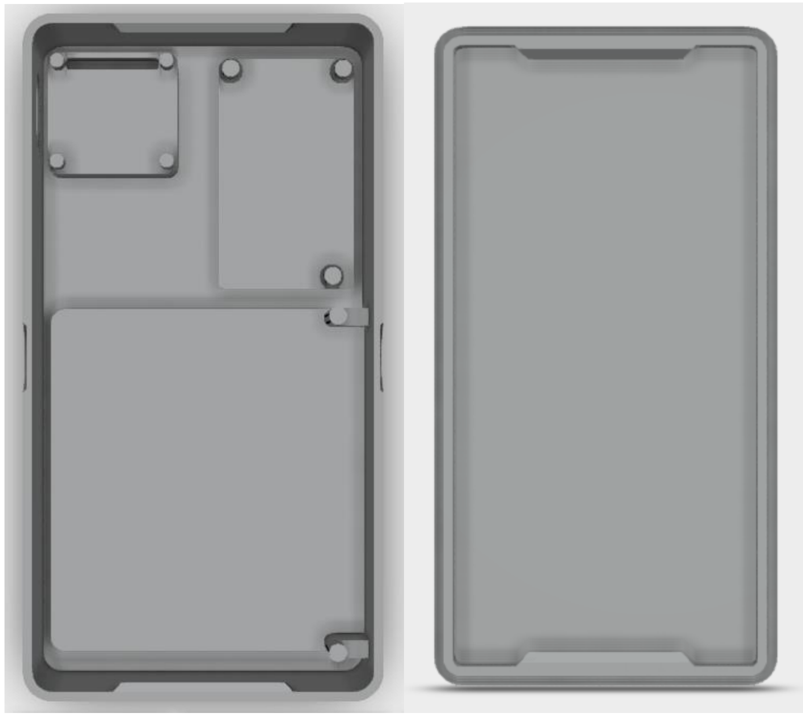


Figure 12. Bottom and top cases for a tag.

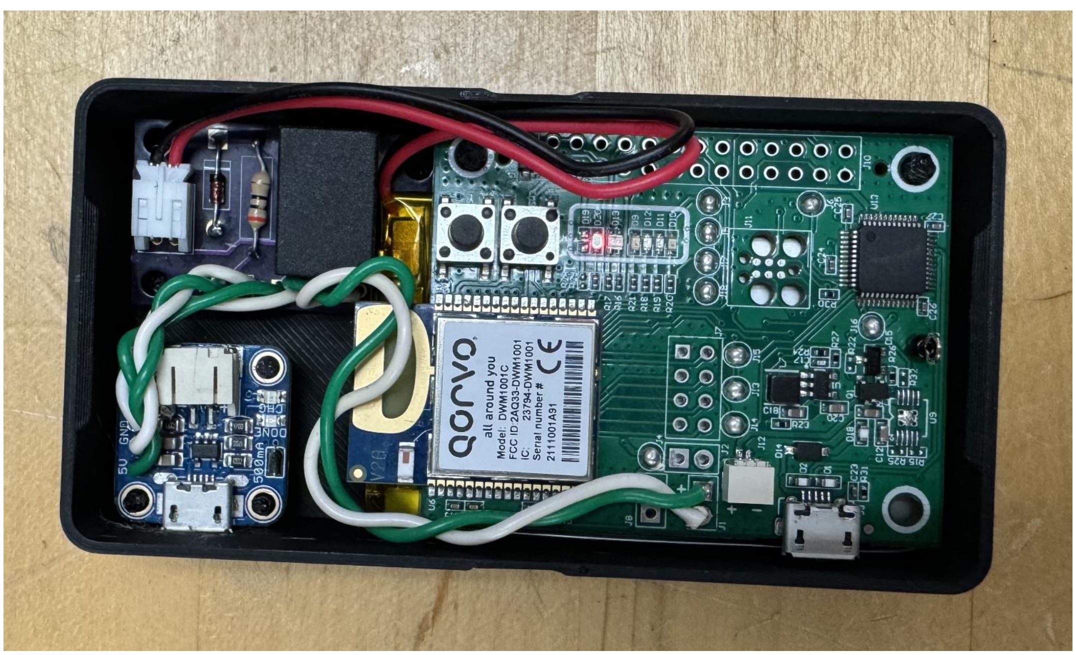


Figure 13. Top view of the final tag case.

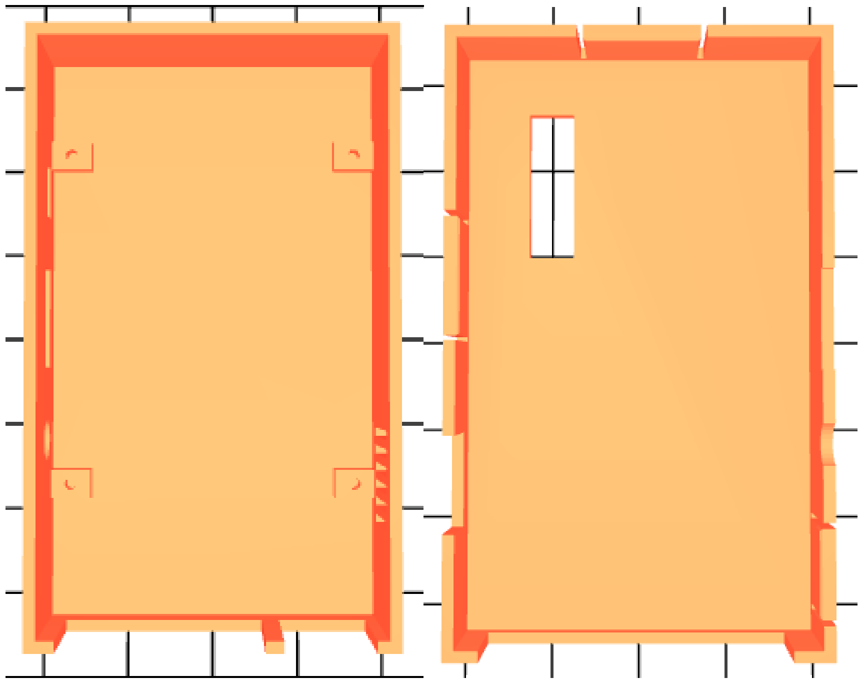


Figure 14. Bottom and top cases for a gateway.

4.4 Recording Location Data

4.4.1 Accessing Position Data

Gateways are constantly receiving position data from every tag that is on the network. For the user to access the position logs of each tag on the network, the logs need to be sent from the gateway to the user's local machine for when the logs need to be reviewed. To accomplish this, the gateways use the MQTT connectivity protocol to interface the DRTLS to the cloud. The MQTT protocol enables devices within the network to publish or subscribe to diverse topics, encompassing device localization, configuration, or non-location IoT data (e.g., sensor data from nodes). In the scope of this project, the MQTT protocol allows the gateways to connect to a broker and then publish the position logging data for another device to subscribe to the topic and receive the position logging data.

For the user to subscribe to a topic and receive the position logging data, a few prerequisites need to be established. First an MQTT broker needs to be set up in the cloud to act as a server, so topics can be subscribed and published to. To set up the MQTT broker we wrote a Python program, which can be seen in Appendix B, that uses the Paho-MQTT library. This library provides a client class that enables applications to establish connections with an MQTT broker for message publication, as well as subscribing to topics and receiving published messages. Additionally, it includes some helper functions to simplify the process of sending one-off messages to an MQTT server. With the broker up and running, we then specified the local IPv4 address and port of the Raspberry Pi 3B we are using as the gateway, in our Python code.

This allows the gateway to connect to the MQTT broker at all times, as long as both the Python program and the gateway are on the same local area network.

Once the gateway and the DRTLS network is active and publishing topics to the MQTT broker, the user machine then needs to subscribe to the specific topics. An example of a topic subscription is “dwm/node/5000/uplink/location”. For every tag on the network, the subscription is going to be the same, except for the third segment of the string, which is the Tag_Node_ID. The tag ID must be specified because every tag has its own unique four digit ID, composed of both numbers and letters.

4.4.2 Recording Position Data

At this point, when the program is run it will output to the terminal, as shown in Figure 15. The figure shows the x, y, and z coordinates, the quality number, and the superframe number. The quality number is a factor between 0 and 100, that is given based on the estimated error of the calculated position. The superframe number is the time slot that the tag is allotted, as shown in Figure 4. The last step in this process is to convert the output into a more user-friendly format and have it saved to either a .txt or .csv file. This is fairly simple, as all that must be done is to parse the incoming payload from the subscription topic to remove all of the unneeded information. Information may also be added to the output in this stage, such as timestamping as shown in Figure 16. This allows the user in the future to find a tag’s position data during any specified time frame. Once all of these steps are completed, our output can be seen in Figure 16, which shows the logged position data, with the date it was logged, the Tag_Node_ID, the x, y, and z coordinates, the quality, and the superframe number.

```
{"position":{"x":0.5443446,"y":8.3553429,"z":-0.63554037,"quality":77},"superFrameNumber":848}
{"position":{"x":0.5443446,"y":8.3553429,"z":-0.63554037,"quality":77},"superFrameNumber":848}
{"position":{"x":0.5443446,"y":8.3553429,"z":-0.63554037,"quality":77},"superFrameNumber":848}
{"position":{"x":0.5443446,"y":8.3553429,"z":-0.63554037,"quality":77},"superFrameNumber":848}
{"position":{"x":0.5443446,"y":8.3553429,"z":-0.63554037,"quality":77},"superFrameNumber":848}
{"position":{"x":0.5443446,"y":8.3553429,"z":-0.63554037,"quality":77},"superFrameNumber":848}
{"position":{"x":0.5443446,"y":8.3553429,"z":-0.63554037,"quality":77},"superFrameNumber":848}
{"position":{"x":0.5443446,"y":8.3553429,"z":-0.63554037,"quality":77},"superFrameNumber":848}
{"position":{"x":0.5443446,"y":8.3553429,"z":-0.63554037,"quality":77},"superFrameNumber":848}
{"position":{"x":0.5443446,"y":8.3553429,"z":-0.63554037,"quality":77},"superFrameNumber":848}
{"position":{"x":0.5443446,"y":8.3553429,"z":-0.63554037,"quality":77},"superFrameNumber":848}
{"position":{"x":0.5443446,"y":8.3553429,"z":-0.63554037,"quality":77},"superFrameNumber":848}
{"position":{"x":0.5443446,"y":8.3553429,"z":-0.63554037,"quality":77},"superFrameNumber":848}
{"position":{"x":0.5443446,"y":8.3553429,"z":-0.63554037,"quality":77},"superFrameNumber":848}
{"position":{"x":0.5443446,"y":8.3553429,"z":-0.63554037,"quality":77},"superFrameNumber":848}
{"position":{"x":0.5443446,"y":8.3553429,"z":-0.63554037,"quality":77},"superFrameNumber":848}
{"position":{"x":0.5443446,"y":8.3553429,"z":-0.63554037,"quality":77},"superFrameNumber":848}
{"position":{"x":0.5443446,"y":8.3553429,"z":-0.63554037,"quality":77},"superFrameNumber":848}
{"position":{"x":0.5443446,"y":8.3553429,"z":-0.63554037,"quality":77},"superFrameNumber":848}
```


Figure 15. The unparsed position data that is outputted to the terminal.

```

2023-11-21 23:14:23 5000,0.70871663,2.2359235,1.2992126,73,222

2023-11-21 23:15:23 5000,0.77162361,2.2385314,1.2717839,75,223

2023-11-21 23:25:56 5000,0.8282969,4.9245205,0.41912296,90,1389

2023-11-21 23:25:57 5000,1.1108797,4.9502883,0.71288455,90,1390

2023-11-21 23:25:58 5000,1.2992263,4.8889022,0.76694494,91,1391

2023-11-21 23:25:59 5000,1.5548162,4.8106985,0.39310578,90,1392

2023-11-21 23:26:00 5000,1.514511,4.63488,0.073635474,93,1393

2023-11-21 23:26:01 5000,1.4293947,4.4663281,-0.076884031,97,1394

2023-11-21 23:26:02 5000,1.4041138,4.2824235,-0.41604027,83,1395

2023-11-21 23:26:03 5000,1.1425792,4.2230678,-0.56340283,90,1396

2023-11-21 23:26:04 5000,1.0156319,4.29118,-0.1291157,87,1397

2023-11-21 23:26:05 5000,0.74947983,4.4061069,-0.3422133,89,1398
```

Figure 16. The parsed position data, outputted to a .txt or .csv file format.

Chapter 5: Testing Procedure

5.1 Test Case #1: Communication Range

Tested By:	UTDesign Team 1682
Test Type	Hardware and Software
Test Case Number	#1
Test Case Name	Communication Range
Test Case Description	Tests to evaluate maximum communication range between anchors and tags in an area, as well as how it performs in the presence of interferences. Additionally, to evaluate if there are limitations for communication from the units in the system to the central computer unit.
Item(s) to be tested	
1	Maximum distance for reliable communication between an anchor and a tag.
Specifications	
Input	**Expected Output/Result**
1. 4-6 anchors 2. 1 tag 3. Monitoring computer 4. Readings at different distances	1. Communication with the tag will function up to a certain distance away from the anchors.
Procedural Steps	
1	Set up anchors
2	Introduce a tag
3	Connect and adjust the anchors with a monitoring computer
4	Move away from setup with unit running the application
5	Introduce obstructions (walls and/or equipment of different materials) and evaluate if these obstructions affect the accuracy of the displayed position on the application. Determine a margin of error with obstructions.

Table 5. Test Case #1: Communication Range.

5.2 Test Case #2: Position Accuracy

<table>
<tr><td colspan="2">Tested By:</td><td colspan="2">UTDesign Team 1682</td></tr>
<tr><td colspan="2">Test Type</td><td colspan="2">Hardware, Software, Demonstration</td></tr>
<tr><td colspan="2">Test Case Number</td><td colspan="2">#2</td></tr>
<tr><td colspan="2">Test Case Name</td><td colspan="2">Position Accuracy</td></tr>
<tr><td colspan="2">Test Case Description</td><td colspan="2">Tests to ensure that a configuration of anchors can detect and keep track of several tags in an area simultaneously. Additionally, we are looking for how accurate the displayed location is, even when obstructions are introduced.</td></tr>
<tr><td colspan="4">Item(s) to be tested</td></tr>
<tr><td>1</td><td colspan="3">Accuracy of the position of a tag displayed on the software application at various positions within a 4 or 6 anchor test bay.</td></tr>
<tr><td>2</td><td colspan="3">Accuracy of the position of multiple tags displayed on the software application at various positions within a 4 or 6 anchor test bay.</td></tr>
<tr><td>3</td><td colspan="3">Accuracy of the potion of the tags displayed on the software application from other rooms, between multiple walls.</td></tr>
<tr><td colspan="4">Specifications</td></tr>
<tr><td colspan="3">Input</td><td>Expected Output/Result</td></tr>
<tr><td colspan="3">1. 4-6 anchors around the test environment
2. People holding tags
3. Display software on a laptop</td><td>All the tags are trackable within a margin of error, with obstacles in the way, such as walls, desks, and various electronics.</td></tr>
<tr><td colspan="4">Procedural Steps</td></tr>
<tr><td>1</td><td colspan="3">Set up 4-6 anchors at the limits of a bay’s dimensions.</td></tr>
<tr><td>2</td><td colspan="3">Introduce a tag at a set location and evaluate the accuracy of the displayed position on the application. Determine a margin of error.</td></tr>
<tr><td>3</td><td colspan="3">Evaluate the accuracy of a moving tag in the area.</td></tr>
<tr><td>4</td><td colspan="3">Introduce several other tags and repeat steps 1-3.</td></tr>
<tr><td>5</td><td colspan="3">Introduce obstructions (walls and/or equipment of different materials) and evaluate if these obstructions affect the accuracy of the displayed position on the application. Determine a margin of error with obstructions.</td></tr>
</table>

Table 6. Test Case #2: Position Accuracy.

5.3 Test Case #3: Loss of Signal

<table>
<tr><td>Tested By:</td><td colspan="2">UTDesign Team 1682</td></tr>
<tr><td>Test Type</td><td colspan="2">Hardware, Software, and Demonstration</td></tr>
<tr><td>Test Case Number</td><td colspan="2">#3</td></tr>
<tr><td>Test Case Name</td><td colspan="2">Loss of Signal</td></tr>
<tr><td>Test Case Description</td><td colspan="2">System should be able to determine if a tag has left the supposed detection area or if an unexpected loss-of-signal has occurred. In the event of the latter, the system will throw a warning.</td></tr>
<tr><td colspan="3">Item(s) to be tested</td></tr>
<tr><td>1</td><td colspan="2">Loss-of-signal detection rate and false positive/negative error rate.</td></tr>
<tr><td colspan="3">Specifications</td></tr>
<tr><td colspan="2">Input</td><td>Expected
Output/Result</td></tr>
<tr><td colspan="2">4. 4 or 6 mounted anchors
5. 1 or multiple tags
6. Monitoring computer</td><td>1. Distinguishable loss of signal detection on the monitoring computer when expected. No detection when an individual leaves the tracking area.</td></tr>
<tr><td colspan="3">Procedural Steps</td></tr>
<tr><td>1</td><td colspan="2">Set up 4-6 anchors at the limits of a bay’s dimensions.</td></tr>
<tr><td>2</td><td colspan="2">Connect and calibrate the anchors with a monitoring computer.</td></tr>
<tr><td>3</td><td colspan="2">Determine system response when a tag leaves the tracking area.</td></tr>
<tr><td>4</td><td colspan="2">Determine system response when a tag is unexpectedly disconnected in the tracking area.</td></tr>
</table>

Table 7. Test Case #3: Loss of Signal.

5.4 Test Case #4: Battery Performance

<table>
<tr><td colspan="2">Tested By:</td><td colspan="2">UTDesign Team 1682</td></tr>
<tr><td colspan="2">Test Type</td><td colspan="2">Hardware</td></tr>
<tr><td colspan="2">Test Case Number</td><td colspan="2">#4</td></tr>
<tr><td colspan="2">Test Case Name</td><td colspan="2">Battery Performance</td></tr>
<tr><td colspan="2">Test Case Description</td><td colspan="2">Tags operating off of a battery should be able to last at least a full shift (12 hours).</td></tr>
<tr><td colspan="4">Item(s) to be tested</td></tr>
<tr><td>1</td><td colspan="3">How long a tag can last operating off of a 2000mAh LiPo battery.</td></tr>
<tr><td>2</td><td colspan="3">The time it takes to recharge a battery from 0%.</td></tr>
<tr><td colspan="4">Specifications</td></tr>
<tr><td colspan="3">Input</td><td>Expected Output/Result</td></tr>
<tr><td colspan="3">1. Fully charged battery
2. Completely depleted battery</td><td>1. The tag should last at least 12 hours
2. A complete charge can be completed overnight</td></tr>
<tr><td colspan="4">Procedural Steps</td></tr>
<tr><td>1</td><td colspan="3">Plug in a DWM1001-DEV into a fully charged 2000mAh flat LiPo battery.</td></tr>
<tr><td>2</td><td colspan="3">Allow the tag to run until the battery has been completely depleted.</td></tr>
<tr><td>3</td><td colspan="3">While the tag runs, retrieve the tag’s time-stamped location data. The time stamps will be used to determine the length of operation.</td></tr>
<tr><td>4</td><td colspan="3">With a fully depleted battery, plug the battery into a charger and record the time it takes to reach full capacity.</td></tr>
</table>

Table 8. Test Case #4: Battery Performance.

Chapter 6: Project Implementation/Operation and Assessment

6.1 Overview of Final Implementation

We configured a gateway, four anchors, and several tags to demonstrate a functioning system on a smaller scale relative to the realistic application at Strike. The fabrication environment at Strike's facility is divided into smaller sub areas with rooms called bays, with each bay containing four zones.

The assembled charging circuit is an important and successful component in the project, transforming the tags into rechargeable devices. This development significantly enhances its practicality for day-to-day operations at Strike. The integration of the charging circuit has addressed the need for a reliable power solution by allowing the tags to be easily recharged. This feature is particularly advantageous for prolonged shifts, as it ensures that the tags can sustain their functionality throughout the entirety of a work shift at Strike. The battery's capacity, combined with the efficiency of the charging circuit, guarantees a dependable power source, minimizing downtime and ensuring continuous tracking and monitoring capabilities.

Figure 17 shows the current setup of the tag case. The case of the tag includes a DWM1001-DEV board, PCB board for the charging circuit, Micro-USB charging board, and a LiPo battery. The DWM1001-DEV board is stacked on top of the battery to help limit the length and width of the case.

Figure 17. Current setup of tag case with components.

6.2 Operational Test Results

In configuring our system, we strategically positioned anchors assuming the largest dimension of a bay at Strike's facility, a key assumption that allowed us to realistically assess the tracking capabilities of our system. Through our observations, we found that the maximum range between anchors reached 42 feet. The 42-foot anchor range implies that, in a typical bay setup, utilizing four anchors positioned at each corner, the system can seamlessly cover the entire bay area. This coverage ensures that every point within the bay falls within the range of at least one anchor, facilitating robust and accurate real-time location tracking for tags within that space.

In our testing environment, we configured tag units to communicate at a high refresh rate of 10 Hz, meaning they reported their data every 100 milliseconds. However, we recognize that the practical implementation of the system in Strike's environment may require adjustments to the refresh rate for optimal efficiency. In a realistic deployment, it is often beneficial to strike a balance between communication frequency and system capacity. While a 10 Hz refresh rate provides detailed and frequent updates, it may also impose a higher demand on system resources when there are many more tags. To enhance system capacity, the tags can be configured to communicate at a lower refresh rate. This adjustment does not compromise the overall functionality but rather optimizes the system for scalability and efficiency.

The successful confirmation of reliable communication between tags and anchors within the designated area, as defined by the anchor positions, is important for the functionality of our tracking system. This means that within the specified boundaries defined by the anchor locations, the tags consistently establish and maintain communication, allowing for accurate and real-time tracking of their positions. As we expected, performance of tags becomes less reliable when operating outside of this predefined area. Understanding the limitations of tag performance outside the designated area is essential for managing expectations and optimizing the system's deployment. It may also guide the strategic placement of anchors to ensure that the entire desired coverage area is encompassed by reliable communication zones.

In our testing, the tag units demonstrated an impressive operational duration of 33 hours on a single charge of a 2000mAh LiPo battery. This extended battery life is critical in ensuring sustained and uninterrupted functionality, especially over the long shift hours at Strike. Equally significant is the recharge time required for the battery. After being completely drained, the tag units needed 4.5 hours to recharge fully. These battery performance metrics not only meet but exceed expectations for a system deployed in a dynamic environment, providing a robust and sustainable power solution that aligns with the operational needs and demands of the intended application.

6.3 Evaluation of Results Relative to Design Criteria

In terms of cost, the total expenses of the UWB system were satisfactory relative to our overall budget. Comparing the cost of the UWB system and the RF system, UWB came out to be less expensive. In terms of performance, the active tracking of the system performs exceptionally, as the update rate of the tag on the display is almost perfect. With regard to user accessibility, the difficulty of understanding the software is not very high, as navigating through the GUI is easy and straightforward. In regards to the simplicity of our design, the final design of our system is not complex. All of the major requirements requested by Strike have been met, as the system tracking works very well, the battery life is more than 12 hours, the GUI is fairly accessible, and the recording of location history was completed.

6.4 Operating Instructions

To set up the UWB network, allowing you to view the precise location of the tags in real-time, we first need four anchors, at least one tag with the battery and charging circuit, a host gateway, and a computer. You first need to plug in the gateway, power it on, and connect a display. This will take you to the Raspberry Pi operating system desktop. You now need to find the IP address of the gateway, this can be done in two different ways. You can either use the terminal on the gateway to find the IP address, or you can download an IP scanner program onto your computer. To find the IP address on the gateway, first, make sure the gateway is connected to your LAN via an ethernet cable or using the wifi. Next, open the terminal and type in “hostname -I”, then press enter, and your IP address will be displayed. Alternatively, you can download an IP scanner program from Google. Once you have the program downloaded, just make sure that the computer is on the same LAN as the gateway. The program should populate a list of all devices and their IP addresses on the LAN. Now that you have the IP address of the host gateway, you can open the DRTLS gateway web manager by typing in the IP address of your gateway into the address bar of your web browser.

Next, you need to set up the one anchor for each corner of the bay. All you need to do is make sure the anchors are plugged in and have power. Now, to set up the anchors and make sure that they are set at the correct coordinates, we need to go to the web manager, where you should see that the anchors have populated onto the right side of the screen. You can now click on each anchor to make sure they are set up as anchors and not tags, while also specifying their x, y, and z coordinates. You will also need to choose one anchor as the initiator. This will be the anchor that communicates with the gateway, while all of the others communicate with it. That means you should choose the anchor that is closest to the gateway to be the initiator. From the initiator, which should have its coordinates set at x = 0, y = 0, and z = 0, you can set the coordinates for the other anchors in a counter clockwise order. Think of all your anchors being on a x-y coordinate graph, with the initiator set at 0, 0. You can then find the distance from the initiator in meters, and then set those distances as your x and y coordinates in the anchor setting menu. The anchor settings menu can be seen in Figure 18.

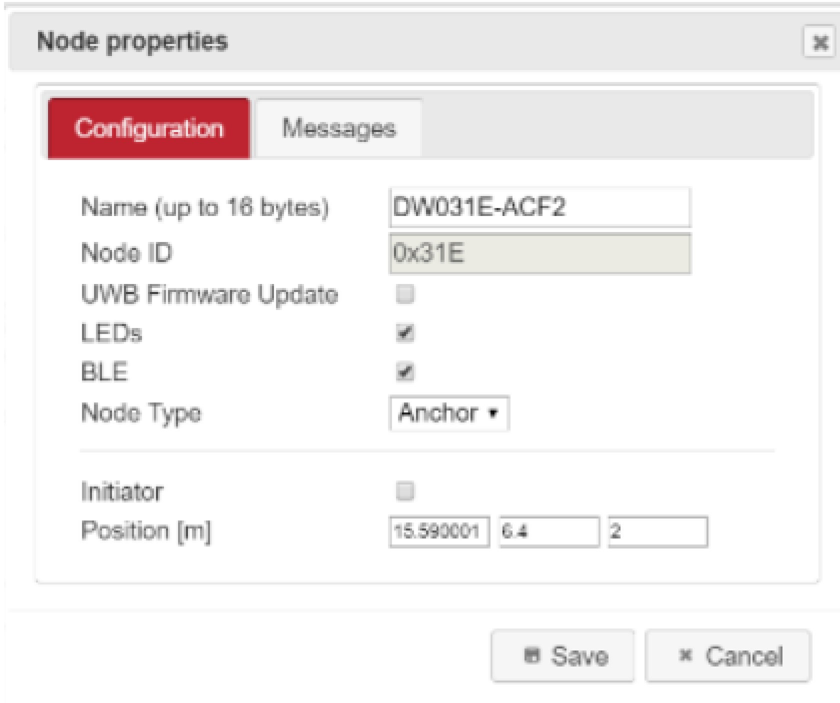


Figure 18. Anchor settings menu in the DRTLS gateway web manager.

Next, you need to set up the tags for your bay. Just like with the anchors, once you provide the tag with power, like from a battery, the tag should automatically be detected in the web manager on the right side of the screen. In the settings menu for the tag, first make sure it is set up as a tag, then you need to specify its name, so you can tell the difference between it and other tags. You can also change the refresh rate of the tag, this will determine how often the tag refreshes its locations and sends it to the gateway. An example of the tag settings menu can be seen in Figure 19.

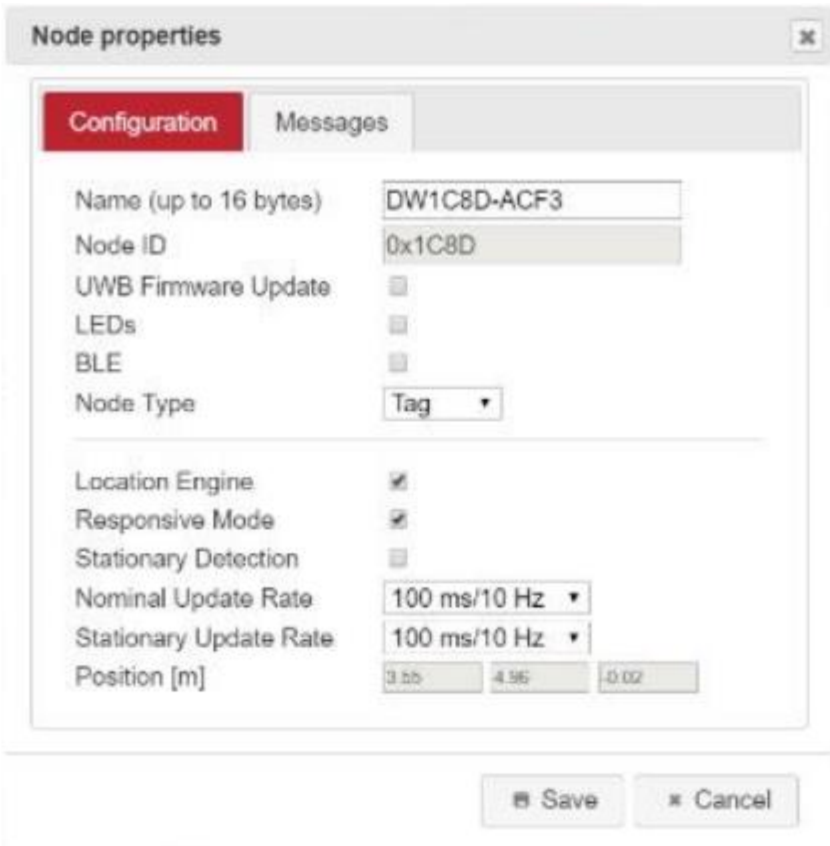


Figure 19. Tag settings menu in the DRTLS gateway web manager.

Now that all of the anchors and tags are set up, you just need to configure the floor plan of the bay in the web manager. You will first need to grab a schematic of the bay to use. Then in the web manager, at the top of the screen you will see the settings menu, which looks like a box with three perpendicular lines. In the settings menu, the first option is for the floorplan configuration window. An example of the floorplan configuration window can be seen in Figure 20. As can be seen in this figure, there is a place to upload your floor plan schematic, adjust the aspect ratio, and adjust the origin. Once you have set up all of the anchors, tags, and set up your floor plan, you should have something similar to Figure 21, but with your specific anchors and tags. This will complete the setup instruction for the UWB personnel tracking within a bay.

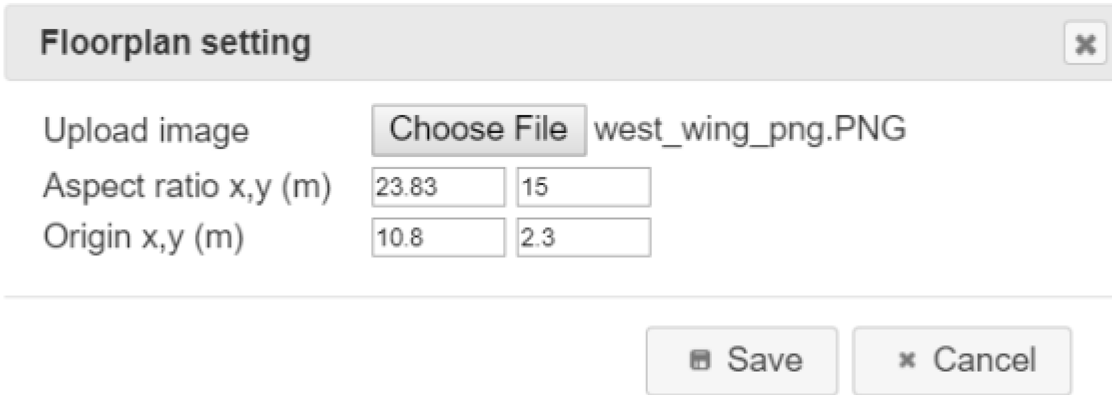


Figure 20. Floor plan configuration window in the DRTLS gateway web manager.

Figure 21. DRTLS gateway web manager.

Chapter 7: Standards Used

The following standards were used in development of this project:

- IEEE802.15.4-2011
- ANSI C63.4: 2004
- FCC CFR 47, Part 15, Subpart A
- FCC CFR 47, Part 15, Subpart B
- FCC CFR 47, Part 15, Subpart C
- FCC CFR 47 Part 15, Subpart F
- CISPR 16-1

Chapter 8: Conclusion

In conclusion, this project has culminated in the successful development and demonstration of a comprehensive system that effectively covers the largest bay size. The inclusion of a switching circuit in the design ensures the protection of the battery during charging, enhancing the overall durability and longevity of the system.

One of the notable achievements is the outstanding performance on batteries, with units consistently lasting over 30 hours. This remarkable battery life is crucial for sustaining operations through extended shifts, promoting efficiency and reliability in real-world applications.

Additionally, the project involved the thoughtful design of cases to enclose both tag and gateway units, ensuring their physical protection and facilitating their deployment in various environments. The incorporation of a Python script for accessing and recording location data further enhances the system's functionality, providing valuable insights and data management capabilities.

Looking ahead, the system developed in this project holds significant potential to become an integral component of life safety management at Strike Photonics. Furthermore, its adaptability to track non-personnel objects, such as wafers, underscores its versatility and applicability across diverse domains. The utilization of UWB technology and wireless communication underscores the efficacy of this solution in delivering real-time tracking capabilities, positioning it as a cutting-edge technology with broad implications for enhanced operational efficiency and safety.

Chapter 9: Future Work Recommendations

Exploring future enhancements for the system, one significant improvement entails consolidating all components onto a single board. Currently, the system utilizes a DWM1001-DEV development board. This board allows us to configure each unit with different roles in the system, but there are components on the development board that are not needed for operation of a tag. Designing a board to carry the module and an integrated charging circuit can reduce the complexity of the system as well as provide for a more compact solution.

In addition to consolidating components, future improvement involves augmenting the features of the casing. Specifically, integrating a clipping mechanism into the case design would add practicality and user convenience. The system was developed to be pocket-sized and we had considered designing a clip, but did not fully explore the design with such a mechanism. This feature would enable users to easily attach the device to various pieces of clothing, enhancing the system's versatility and adaptability in different environments. A clip-on feature would contribute to the system's user-friendliness, allowing for more flexible and convenient deployment scenarios.

Finally, with respect to the charging of the device, switching the Micro-USB board to a USB Type-C board would improve charging speeds and help to minimize the number of different cables needed due to the continued widespread adoption of the USB Type-C port.

Chapter 10: Other Issues

10.1 Ethics

10.1.1 Privacy

An important consideration in the system's design revolves around maintaining data confidentiality. To uphold user privacy and safeguard the integrity of local networks, the devices integrated into the system are not designed to collect any data pertaining to individual users or details about the local network in which they operate. This intentional exclusion of data collection ensures a high level of privacy and data security, aligning with contemporary standards and regulations governing the protection of sensitive information. By abstaining from gathering user-related data and local network specifics, the system prioritizes privacy, fostering trust and confidence in its use among individuals and organizations alike.

10.1.2 Intent

Another core principle guiding the implementation of the system is a commitment to non-malicious use. Its primary purpose is expressly not geared towards any form of intentional harm, malicious attacks, or theft of company assets and hardware. This commitment to non-malicious use is underpinned by a dedication to ethical practices and responsible deployment, ensuring that the technology serves its intended purpose of enhancing operations and security without posing any threat to the company's assets, hardware, or day-to-day business activities.

The design actively discourages any attempts to exploit vulnerabilities for malicious purposes, fostering a secure and trustworthy environment for the company. This commitment reflects not only a technical approach to security but also an ethical stance that underscores the responsible and constructive use of the system within the broader context of the company's objectives and values.

10.1.3 Reliability

Unlike some technologies designed with a predetermined lifespan or intentional decline in performance, this system has been crafted with a commitment to longevity and sustained functionality. The goal is to provide users with a reliable and enduring solution that resists becoming obsolete or experiencing deterioration over time.

Furthermore, this approach aligns with a customer-centric perspective, emphasizing the value of sustained usability and minimizing the total cost of ownership over the system's lifetime. It underscores a commitment to delivering enduring solutions, reinforcing the system's reliability and the trust placed in it by users over the long term.

10.2 Soft Skills

10.2.1 Lifelong Learning

The project served as a dynamic learning experience for all team members, emphasizing the mindset of perpetual learning within the team. We embraced the idea that learning is an ongoing process, team members proactively sought out new information and innovative approaches to achieve the project's overarching goals. Challenges encountered during the project were valuable opportunities for individual and collective growth. These obstacles were chances to expand our skill sets, problem-solving abilities, and domain expertise. This proactive and positive approach towards challenges contributed to the overall success of the project.

10.2.2 Time Management

The team embraced the practice of scheduling regular weekly meetings to foster open communication, address project-related concerns, and adapt to evolving circumstances. These meetings provided a space for team members to collaboratively discuss progress, share insights, and promptly address any emerging conflicts throughout the project's duration. Our meetings with Strike and Dr. Skinner were also important opportunities to receive feedback on our work and progress.

The designated meeting manager played a role in enhancing coordination and minimizing disruptions between scheduled meetings. This team member took on the responsibility of overseeing the adherence to meeting timelines, ensuring that critical discussions were not deferred, and conflicts were promptly addressed. By centralizing this responsibility, the team experienced a notable reduction in major issues arising during the intervals between meetings, creating a smoother workflow and reinforcing a sense of accountability among team members.

10.2.3 Multidisciplinary Teams

The diversity within the team, manifested through the unique skill sets and perspectives brought by each member, played a pivotal role in shaping the project's success. As challenges emerged throughout the project, the collective wealth of skills and ideas became an invaluable resource, providing a multifaceted approach to problem-solving.

The open-sharing of ideas was instrumental in enhancing communication within the team. Team members learned to leverage their diverse strengths, creating an environment that allowed for the smooth flow of ideas and information. This collaborative dynamic not only enriched the quality of solutions but also cultivated an inclusive and supportive atmosphere where everyone felt empowered to contribute.

Chapter 11: Costs Estimate

11.1 Materials and Construction

The figure below is a pie chart of our project expenses regarding the purchase of devices and components. Note that the section labeled as RFID approach encompasses all components (including tags, readers, antennas, testing equipment, etc.) that were purchased. Although these items were ultimately not used in the final solution, we must acknowledge these expenses as a part of our project in the engineering processes of research and design.

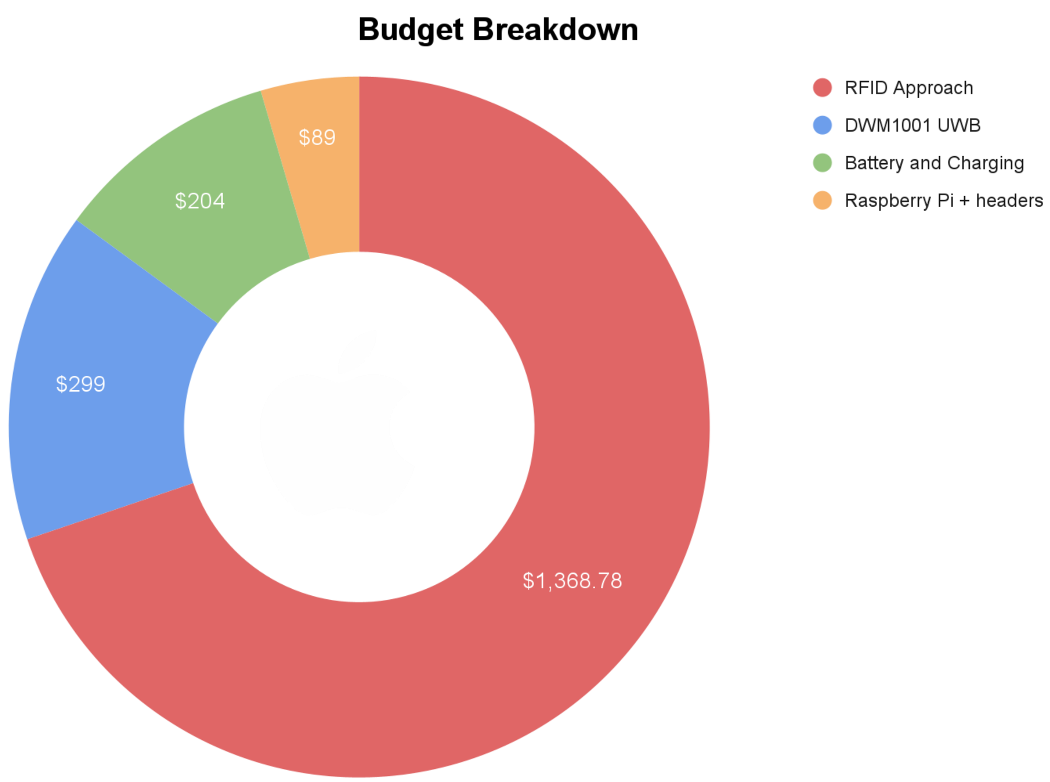


Figure 22. Estimated project expenses.

The project was given a budget of $2,000 for purchasing parts and an additional $1,000 was allocated for machine shop work. If machine shop work was not necessary, we were allowed to use that part of the budget to purchase components. Strike Photonics had also offered the services of their own machine shop to help reduce some costs. Expenses for our ultra-wide band solution totalled up to ~$592. Out of the total budget of $3,000, we consumed $1,960 throughout the project.

11.2 Estimate of design cost (man -hours, travel, etc)

The estimated man -hours for project completion from inception to delivery.

Semester 1: 940 Hours

Semester 2: 1020 Hours

Chapter 12: Project Management Summary

12.1 Tasks Completed

Overall, the team completed 40 tasks throughout the first and second semester. Below, the tasks will be divided into RF tasks and UWB tasks that were completed. The Gantt Chart for both semesters can be found in the appendices.

- Research
    - RF
        - Research Bluetooth, RF
        - Research for parts
        - Research Mercury API
        - Duplexers
        - Circulators
        - Vector Modulation
        - Power and low noise amplifiers
    - UWB
        - Qorvo MDEK1001
            - How to adjust each board to be an anchor, initiator, or tag
        - Gateways
            - The use of multiple gateways for multiple networks
        - Decawave (software)
            - How to incorporate a custom floor plan diagram
        - Issue with python 3
        - Relay circuit
- Hardware Tasks Completed:
    - RF

        - Test all ordered components
        - RFID reader, tags, and antenna
            - Solder headers onto RFID reader
        - Gather RSSI values for different positions at Strike
            - Analyze the max temperature the RFID can reach
        - Test system using fans
            - Analyze the max temperature of the RFID
        - Develop a location determining algorithm
        - Plan out PCBs for power amplifier and low noise amplifier
    - UWB
        - Test the system
            - Solder off the default charging board on the tags
            - Analyze the full duration of the battery life
            - Analyze the max temperature that the tag reaches
        - Build breadboard prototype
        - Test using the PCB and tag case
        - Troubleshoot the system
- Software Tasks Completed
    - RF
        - Develop the Arduino code for the RF system
        - Develop a code to convert RSSI values to distance in meters
    - UWB
        - Understand how to use MQTT.fx
        - Develop python script for recording location history
- UWB

- Design a custom 3D printed case for a tag
    - Adjust the original design for better fit
- Design a custom 3D printed case for a gateway
    - Adjust the original design for better fit
- Design a PCB board for the charging circuit

12.2 Tasks Pending Completion

There were a total of 2 pending tasks that were not completed

- Hardware
    - Maximize the network capacity
    - Using multiple gateways

12.3 Time allocated

Total project duration: February 2023 to December 2023.

12.4 Facilities used

Strike Photonics offered an empty space on the second floor of their facility for us to conduct our research regarding the range capabilities of our initial RFID approach. The UTDesign Studio open lab is where the majority of our work was completed. Final testing was also completed in the open lab as we took advantage of the existing obstructions in order to test the capabilities and robustness of our system.

Appendices

Appendix A: Design Drawings

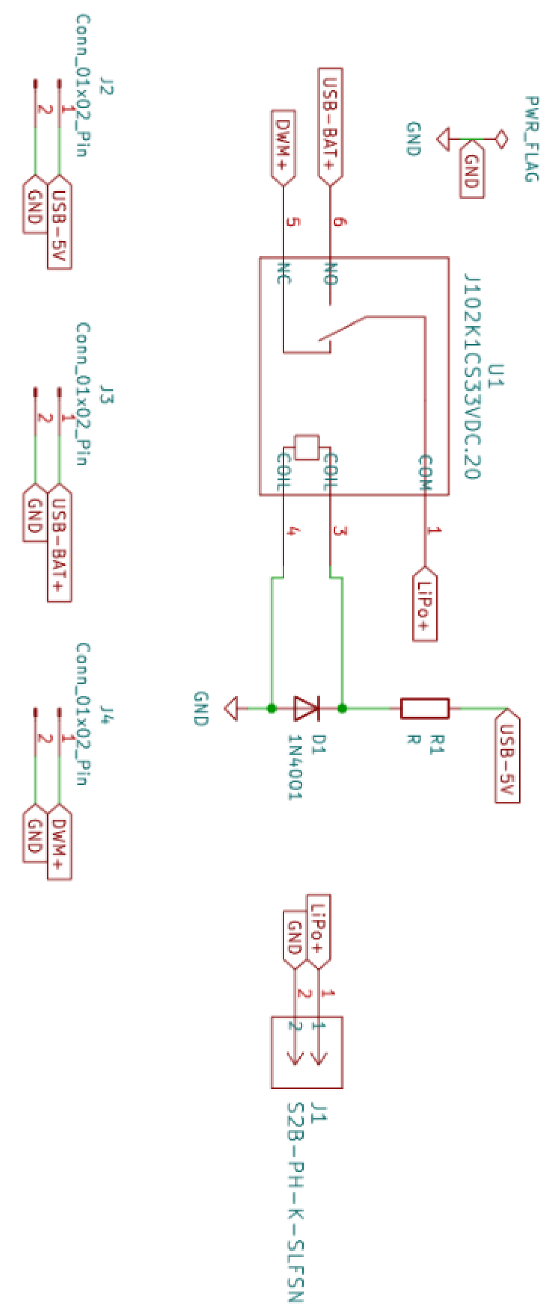


Figure 23. Full switching circuit schematic.

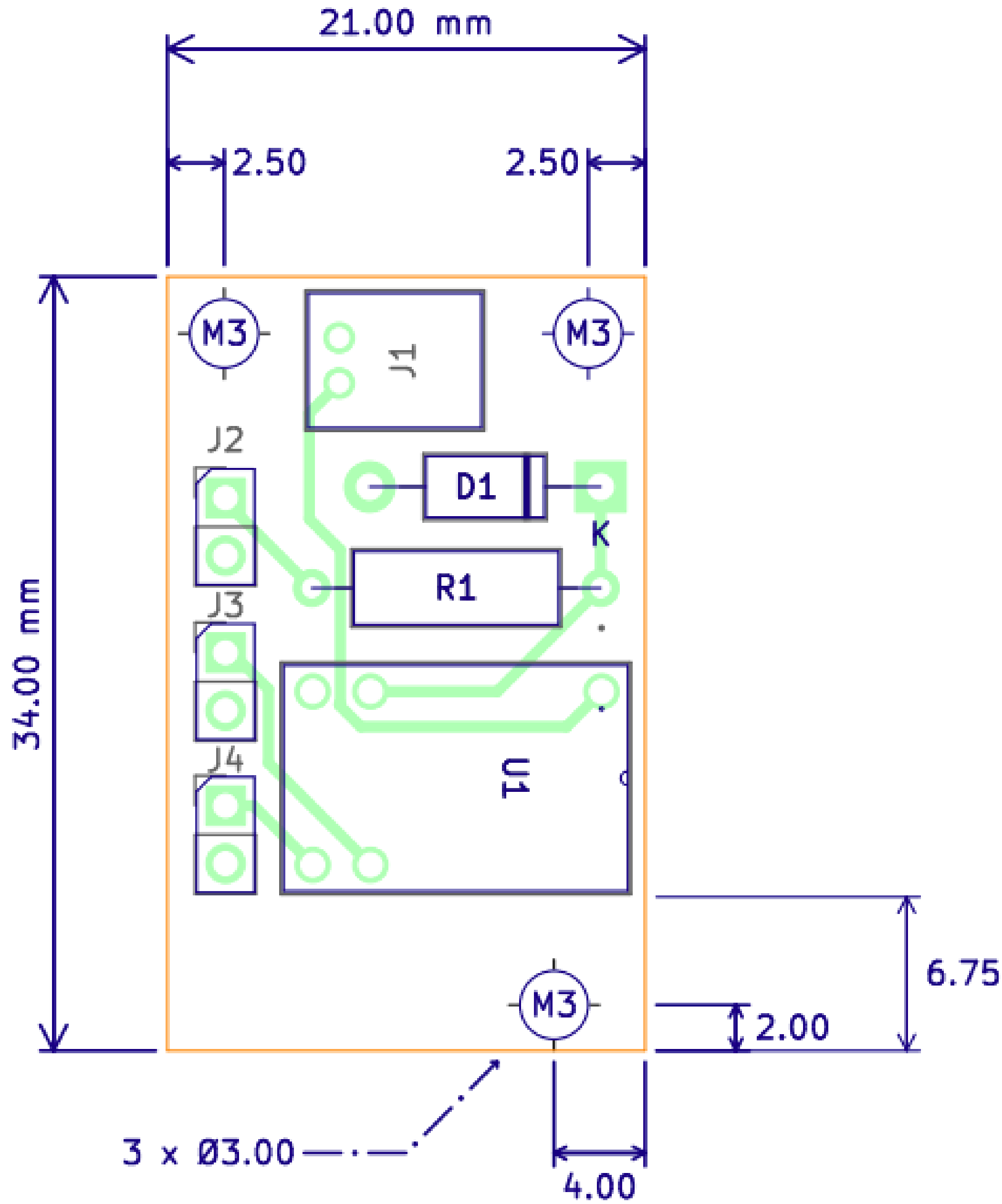


Figure 24. PCB assembly drawing (front).

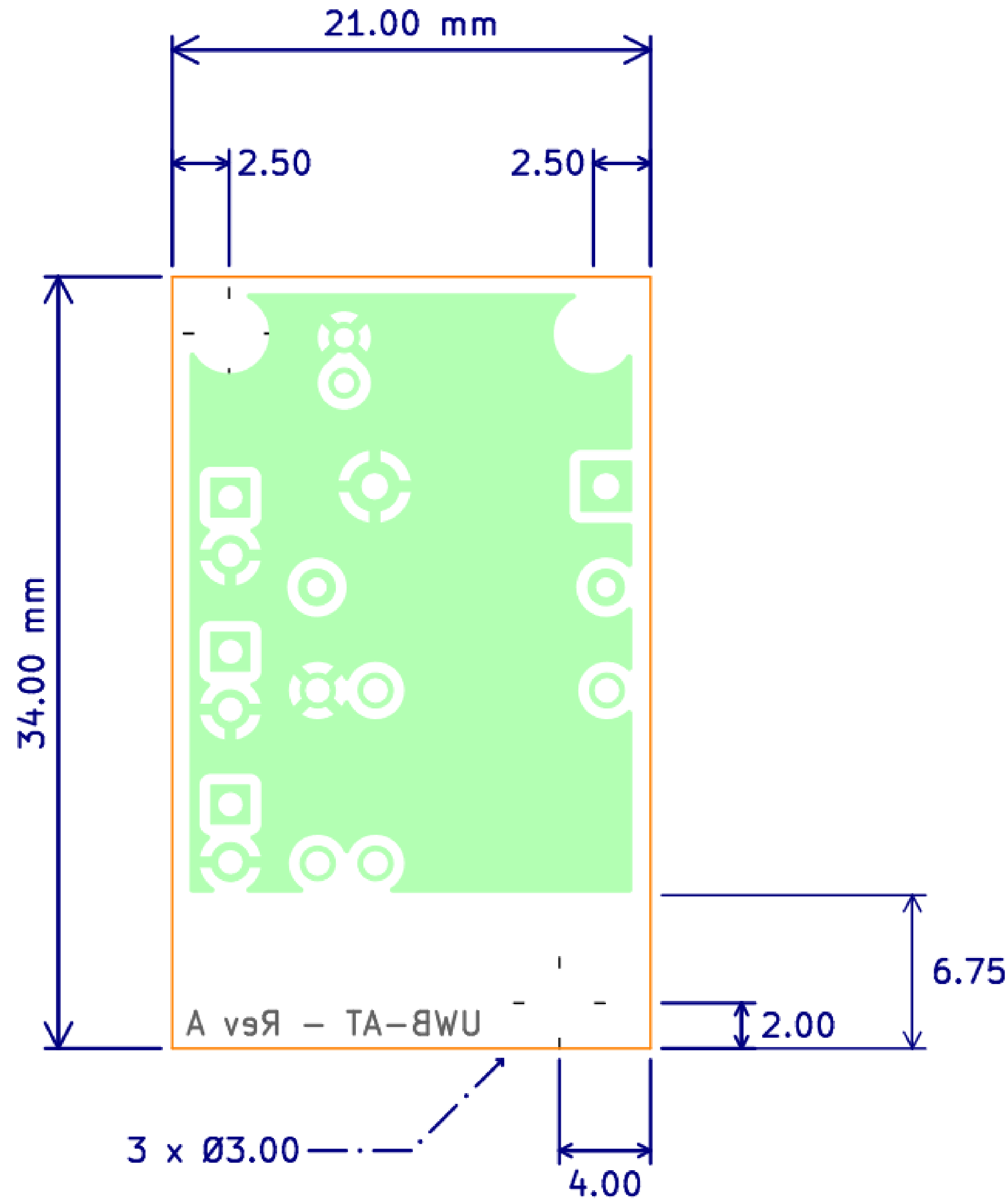


Figure 25. PCB assembly drawing (back, mirrored).

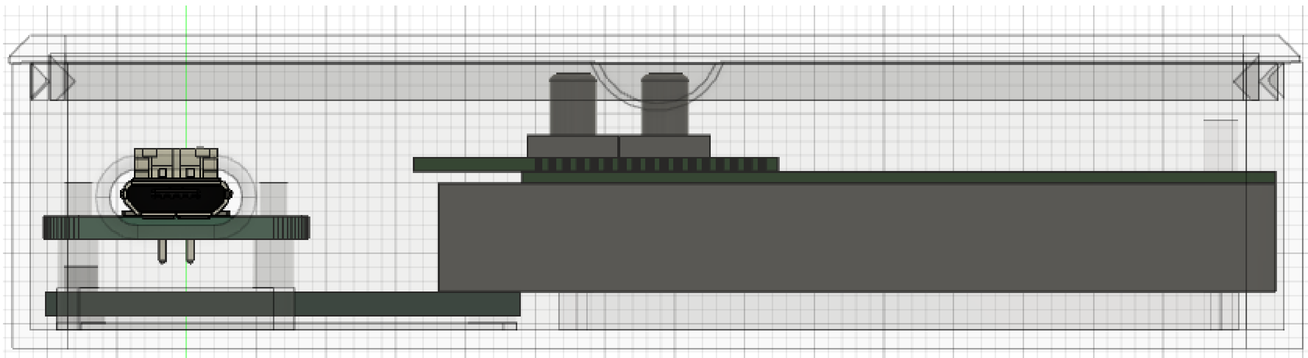


Figure 26. Full side view of tag case with simplified components.

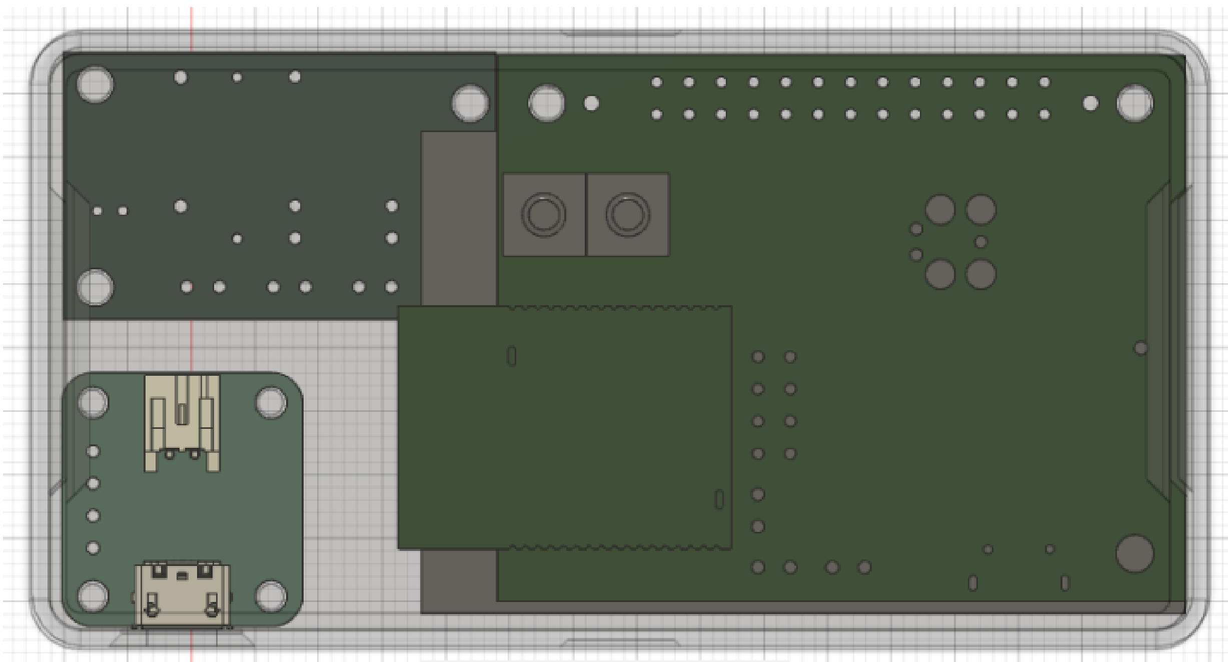


Figure 27. Full top view of tag case with simplified components.

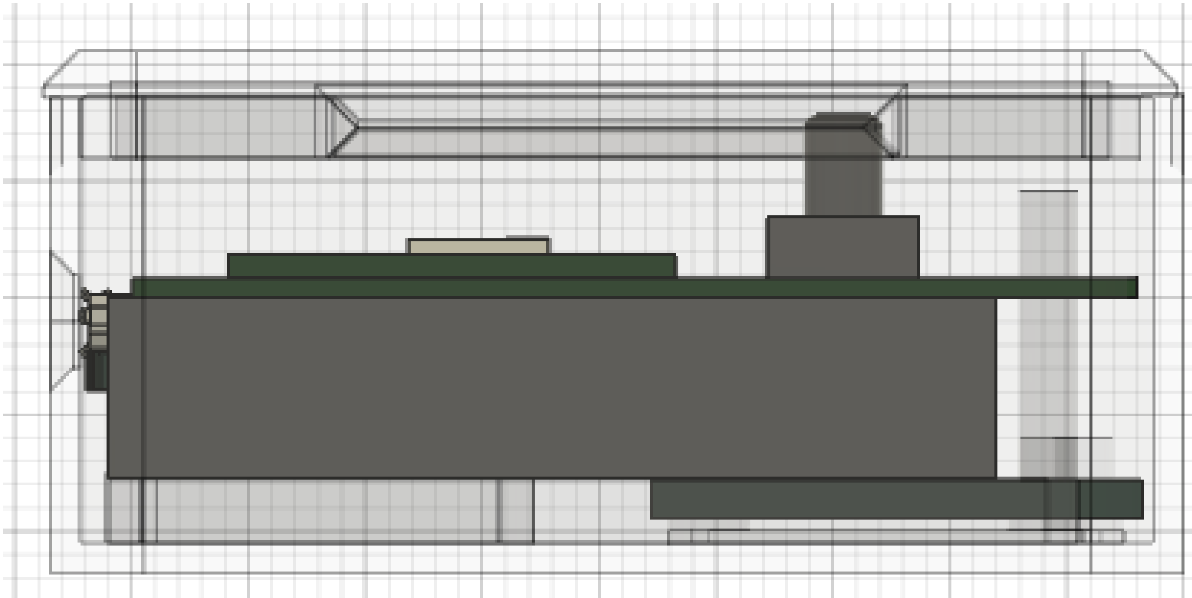


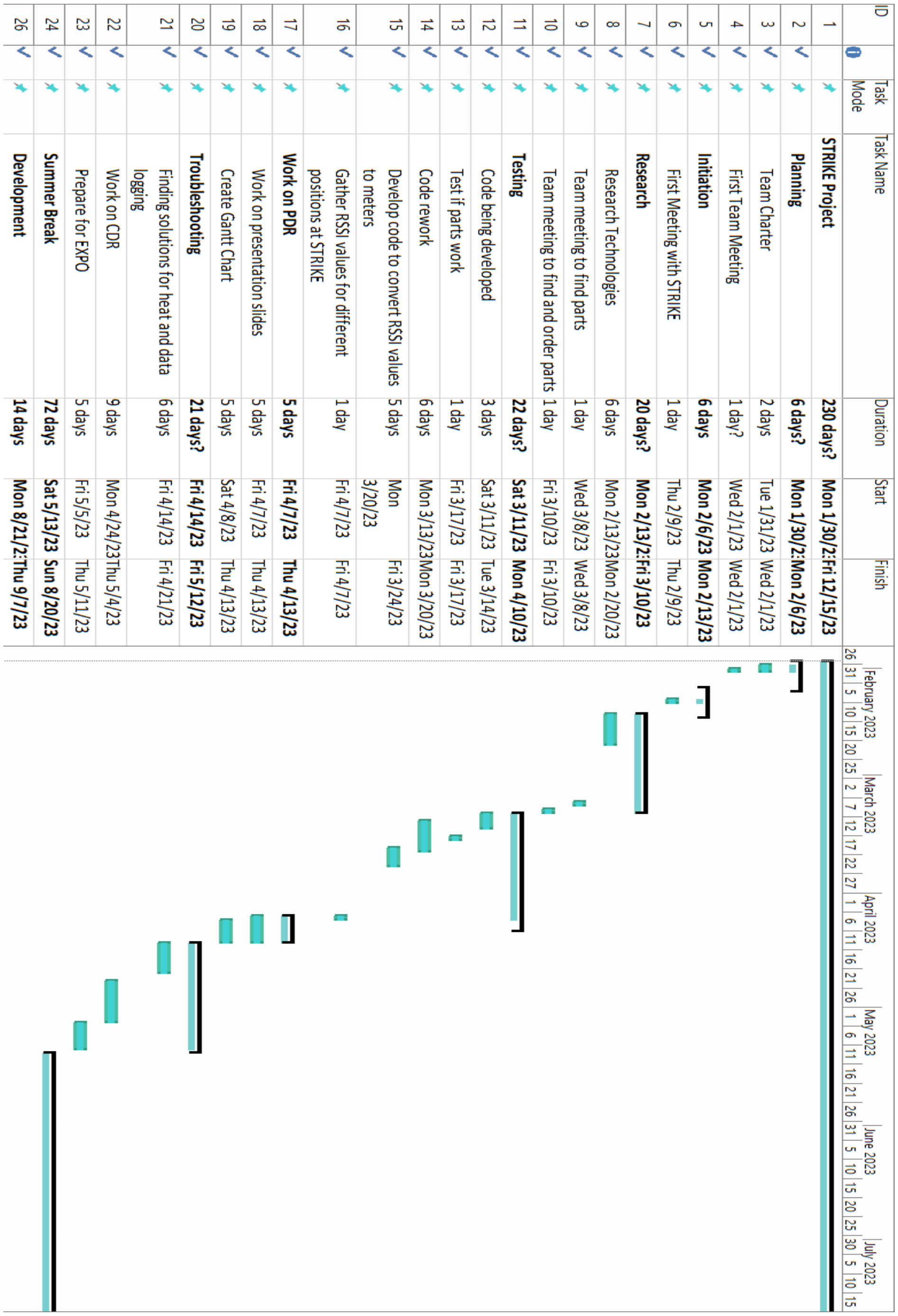
Figure 28. Full front view of tag case with simplified components.

Appendix B: Python Script

```
import paho.mqtt.client as mqttClient
import time
import datetime
import os
import csv
import json

# Callback function for when the client connects to the MQTT broker
def on_connect(client, userdata, flags, rc):
    if rc == 0:
        print("Connected to broker")
        # This is where you add the subscription topic and message for each tag on the
network, you just need to change the 4 digit Tag_Node_ID, to the one matching the tag.
        client.message_callback_add("dwm/node/5000/uplink/location", lambda client,
userdata, message: on_message(client, userdata, message, tag_node_id=5000))
        client.message_callback_add("dwm/node/1916/uplink/location", lambda client,
userdata, message: on_message(client, userdata, message, tag_node_id=1916))
        client.subscribe("dwm/node/5000/uplink/location")
        client.subscribe("dwm/node/1916/uplink/location")
    else:
        print(f"Connection failed with result code {rc}")
        client.reconnect()

# Callback function for when a message is received from the MQTT broker
def on_message(client, userdata, message, tag_node_id):
    # Decode the payload of the message
    payload_str = message.payload.decode('utf-8')

    # Parse JSON data
    payload_json = json.loads(payload_str)

    # Extract position information
    position = payload_json.get('position', {})
    x = position.get('x', '')
    y = position.get('y', '')
    z = position.get('z', '')
    quality = position.get('quality', '')
    superFrameNumber = payload_json.get('superFrameNumber', '')  # Adjusted this line

    # Define the tag node ID, add a variable for each tag on the network
    print(f"Tag Node ID: {tag_node_id}")

    # Print individual coordinates
    print(f"X: {x}, Y: {y}, Z: {z}, Quality: {quality}, SuperFrameNumber:
{superFrameNumber}")

    # If you would like the output to be printed to a .txt file, use this block of code:
    # Set the file path for the .txt file
    '''script_dir = os.path.dirname(os.path.realpath(__file__))
    file_path = os.path.join(script_dir, 'test.txt')

    # Write the formatted data to the .txt file
    with open(file_path, 'a+') as f:
        formatted_data = (
            f"{'{:%Y-%m-%d %H:%M:%S}'.format(datetime.datetime.now())} "
            f"{tag_node_id}      X:{x},      Y:{y},      Z:{z},      Quality:{quality},
SuperFrameNumber:{superFrameNumber}\n"
        )
```

```
        f.write(formatted_data)'''


    # If you would like the output to be printed to a .csv file, use this block of code:
    # Set the file path for the .csv file
    script_dir = os.path.dirname(os.path.realpath(__file__))
    file_path = os.path.join(script_dir, 'test.csv')

    # Write the formatted data to the .csv file
    with open(file_path, 'a+', newline='') as csvfile:
        csv_writer = csv.writer(csvfile)
        formatted_data = [
            f"{'{:%Y-%m-%d %H:%M:%S}'.format(datetime.datetime.now())}",
            tag_node_id, x, y, z, quality, superFrameNumber
        ]
        csv_writer.writerow(formatted_data)

    # Introduce a delay of 5 seconds (you can adjust this value)
    time.sleep(5)

# Initialize connection status variable
Connected = False

# MQTT broker configuration
broker_address = "172.20.10.2"
port = 1883
user = ""
password = ""

# Create an MQTT client instance
client = mqttClient.Client("UWB")
client.username_pw_set(user, password=password)
client.on_connect = on_connect
client.on_message = on_message

# Attempt to connect to the MQTT broker with retries
while not Connected:
    try:
        client.connect(broker_address, port, 60)  # the 60 corresponds to the number of
seconds before the broker has to reconnect. But during testing, I found it's double this
number before the broker must reconnect. I do not know why this is.
        Connected = True  # Assume connection is successful
        client.loop_start()  # Start the loop in the background
    except OSError:
        print("Connection failed, retrying...")
        time.sleep(5)

# Main loop to keep the program running
try:
    while True:
        time.sleep(1)
except KeyboardInterrupt:
    print("Disconnecting...")
    client.disconnect()
    client.loop_stop()
    print("Disconnected.")
```

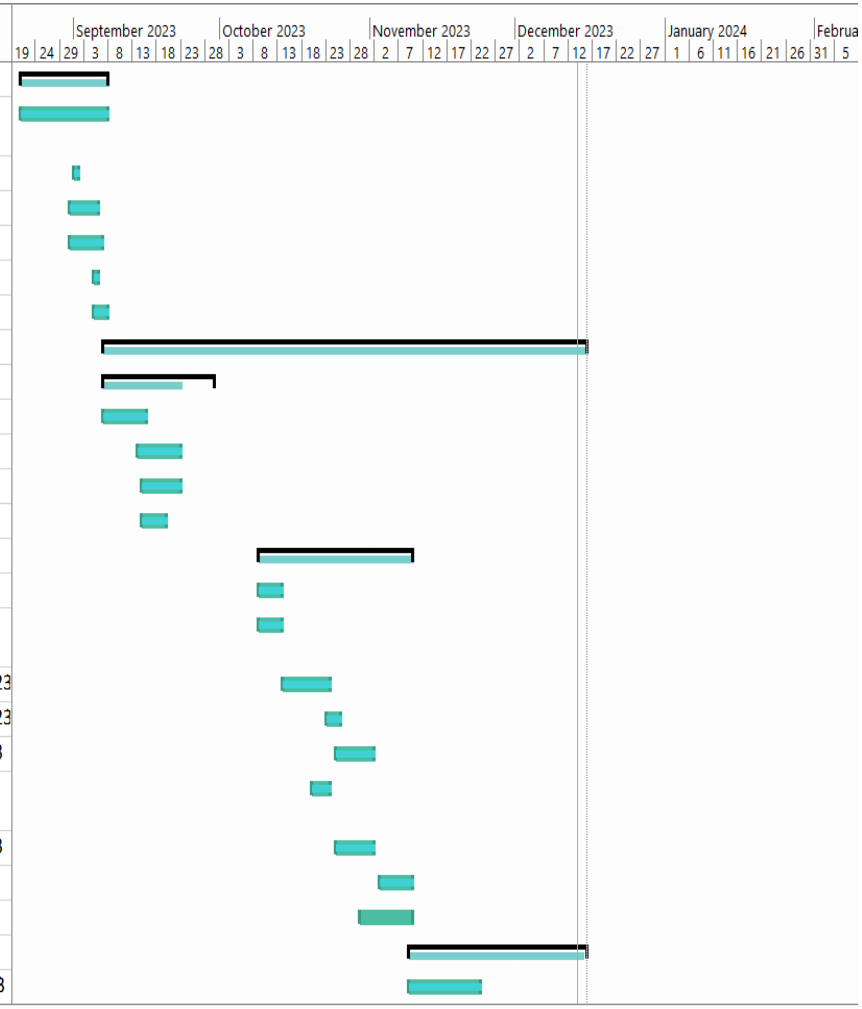
Appendix C: Gantt Charts



ID	Task Mode	Task Name	Duration	Start	Finish
1		**STRIKE Project**	**230 days?**	**Mon 1/30/2:**	**Fri 12/15/23**
2		**Planning**	**6 days?**	**Mon 1/30/2:**	**Mon 2/6/23**
3		Team Charter	2 days	Tue 1/31/23	Wed 2/1/23
4		First Team Meeting	1 day?	Wed 2/1/23	Wed 2/1/23
5		**Initiation**	**6 days**	**Mon 2/6/23**	**Mon 2/13/23**
6		First Meeting with STRIKE	1 day	Thu 2/9/23	Thu 2/9/23
7		**Research**	**20 days?**	**Mon 2/13/2:**	**Fri 3/10/23**
8		Research Technologies	6 days	Mon 2/13/23	Mon 2/20/23
9		Team meeting to find parts	1 day	Wed 3/8/23	Wed 3/8/23
10		Team meeting to find and order parts	1 day	Fri 3/10/23	Fri 3/10/23
11		**Testing**	**22 days?**	**Sat 3/11/23**	**Mon 4/10/23**
12		Code being developed	3 days	Sat 3/11/23	Tue 3/14/23
13		Test if parts work	1 day	Fri 3/17/23	Fri 3/17/23
14		Code rework	6 days	Mon 3/13/23	Mon 3/20/23
15		Develop code to convert RSSI values to meters	5 days	Mon 3/20/23	Fri 3/24/23
16		Gather RSSI values for different positions at STRIKE	1 day	Fri 4/7/23	Fri 4/7/23
17		**Work on PDR**	**5 days**	**Fri 4/7/23**	**Thu 4/13/23**
18		Work on presentation slides	5 days	Fri 4/7/23	Thu 4/13/23
19		Create Gantt Chart	5 days	Sat 4/8/23	Thu 4/13/23
20		**Troubleshooting**	**21 days?**	**Fri 4/14/23**	**Fri 5/12/23**
21		Finding solutions for heat and data logging	6 days	Fri 4/14/23	Fri 4/21/23
22		Work on CDR	9 days	Mon 4/24/23	Thu 5/4/23
23		Prepare for EXPO	5 days	Fri 5/5/23	Thu 5/11/23
24		**Summer Break**	**72 days**	**Sat 5/13/23**	**Sun 8/20/23**
26		**Development**	**14 days**	**Mon 8/21/2:**	**Thu 9/7/23**

Figure 29. First semester Gantt chart.

ID		Task Mode	Task Name	Duration	Start	Finish
26	✓		**Development**	**14 days**	**Mon 8/21/2:**	**Thu 9/7/23**
27	✓		Develop a location determing algorithm	14 days	Mon 8/21/23	Thu 9/7/23
28	✓		Test using fans	1 day	Fri 9/1/23	Fri 9/1/23
29	✓		Research Duplexers	4 days	Thu 8/31/23	Tue 9/5/23
30	✓		Research Mercury API	5 days	Thu 8/31/23	Wed 9/6/23
31	✓		Look for components (PA, LNA)	1 day	Tue 9/5/23	Tue 9/5/23
32	✓		Plan out PCBs for PA, LNA	3 days	Tue 9/5/23	Thu 9/7/23
33			**Shift to UWB**	**72 days?**	**Thu 9/7/23**	**Fri 12/15/23**
34	✓		**Research**	**17 days?**	**Thu 9/7/23**	**Fri 9/29/23**
35	✓		Research Qorvo MDEK1001	7 days	Thu 9/7/23	Fri 9/15/23
36	✓		Research the use of gateways	7 days	Thu 9/14/23	Fri 9/22/23
37	✓		Research Decawave (Software)	6 days	Fri 9/15/23	Fri 9/22/23
38	✓		Research the issue with Python 3	3 days	Fri 9/15/23	Tue 9/19/23
39			**Testing**	**24 days?**	**Mon 10/9/2:**	**Thu 11/9/23**
40	✓		Test the MDEK1001	5 days	Mon 10/9/23	Fri 10/13/23
41	✓		Figure out how to adjust each board to be either an anchor, tag,	5 days	Mon 10/9/23	Fri 10/13/23
42	✓		Look for parts for charging circuit	7 days	Sat 10/14/23	Mon 10/23/23
43	✓		Build breadboard prototype	3 days	Mon 10/23/2	Wed 10/25/23
44	✓		Design a new case	6 days	Wed 10/25/2	Wed 11/1/23
45	✓		Incorporate a custom floor plan into the Decawave software	2 days	Fri 10/20/23	Mon 10/23/23
46	✓		Design a PCB for charging circuit	6 days	Wed 10/25/2	Wed 11/1/23
47	✓		Test using PCB and New Case	5 days	Fri 11/3/23	Thu 11/9/23
48			Incorporate multiple gateways	9 days	Mon 10/30/2	Thu 11/9/23
49			**Troubleshooting**	**27 days?**	**Thu 11/9/23**	**Fri 12/15/23**
50	✓		Access history logs of tracked locati	11 days	Thu 11/9/23	Thu 11/23/23



ID	Task Mode	Task Name	Duration	Start	Finish
49		**Troubleshooting**	**27 days?**	**Thu 11/9/23**	**Fri 12/15/23**
50		Access history logs of tracked locati	11 days	Thu 11/9/23	Thu 11/23/23
51		Present demo to STRIKE	1 day	Thu 11/9/23	Thu 11/9/23
52		Begin working on Final Report	19 days	Fri 11/10/23	Wed 12/6/23
53		Troubleshoot any problems	11 days	Fri 11/10/23	Fri 11/24/23
54		Last touches on circuits/case	9 days	Tue 11/21/2:	Fri 12/1/23
55		Adjust the original case for a better	15 days	Mon 11/20/2	Fri 12/8/23
56		Maximize the network capacity	14 days	Mon 11/13/2	Thu 11/30/23
57		**Prep for EXPO**	**10 days**	**Fri 12/1/23**	**Thu 12/14/23**
61		Finish Final Report	10 days	Fri 12/1/23	Thu 12/14/23

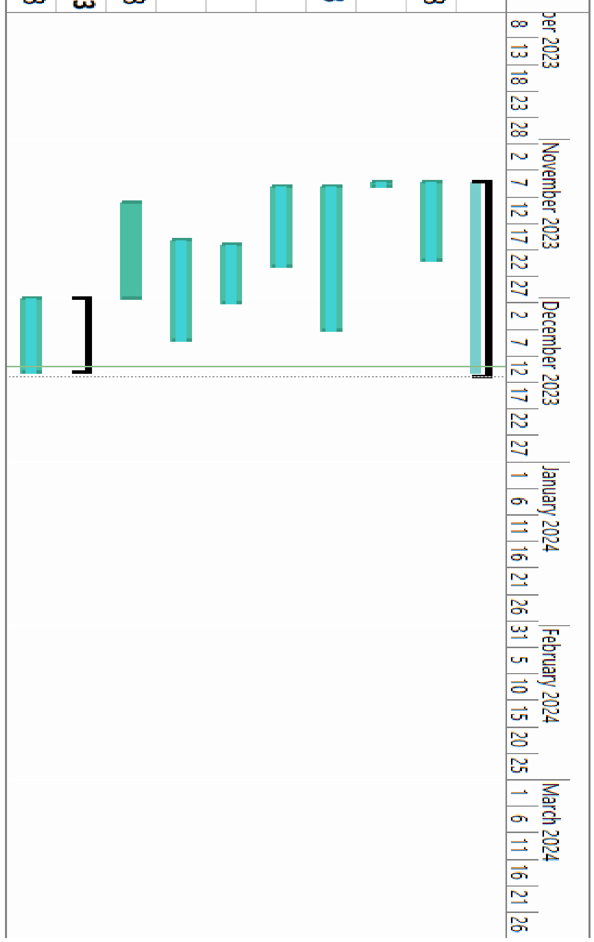


Figure 30. Second semester Gantt charts.

Appendix D: Improved RF Front End for RFID Readers

As described in Section 2.2.3, the commercial RFID readers available within this project's budget do not have the necessary sensitivity in order to be used for personnel tracking. The M6e Nano on the SparkFun Simultaneous RFID Reader has a modular output power from 0 dBm to +27 dBm, along with a minimum detectable signal level of -57 dBm[4] at maximum output power. However, sensitivity improves by roughly 1 dB for every dB decrease in transmit power. Thus, a sensitivity of -84 is achieved with an output power of 0 dBm. The premise of the design shown in Figure 31 is to decrease the output power at the RFID Transceiver Board to 0 dBm in order to improve sensitivity, then use a power amplifier (PA) to keep the transmit power at the antenna at +27 dBm or even increase it to +30 dBm for more performance.

The isolation of the onboard circulator was estimated to be 22 dB. This is because we believe the relationship between sensitivity and transmit power comes from saturation of the receiver ADC during full-duplex communications. If the receiver sensitivity is -57 dBm at a transmit power of 27 dBm, and the dynamic range of a 10-bit ADC is ~62 dB, then the onboard isolation must be the difference between the transmit power and the sum of the remaining quantities. This creates an important design goal to ensure that any changes to the RF front-end can improve on the preexisting architecture.

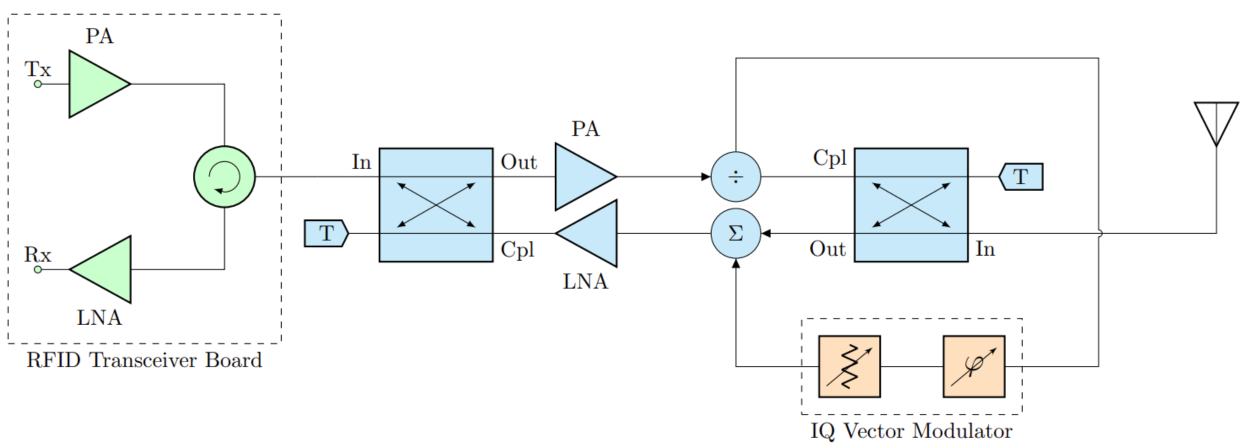


Figure 31. Improved RF front-end block diagram.

Additionally, a low-noise amplifier (LNA) is used on the receive channel for further increased performance. This design requires 2 high isolation directional couplers to be used as circulators that can split a single channel into an isolated receive and transmit path. Commercial couplers do not offer enough isolation, so the transmit signal must also be "*tapped*", phase-shifted by 180 degrees, and injected back into the receive channel at the

[4] Ref: ThingMagic Nano User Guide, Page 23. This is determined with 8 Miller-modulated subcarriers (M value).

same power level. This is a form of single tap, analog SIC and was tested using an Analog Devices AD8340 Vector Modulator. 22 dB of cancellation was achieved with a 3 dB bandwidth of ~30 MHz, however, it may be possible to achieve higher cancellation by matching the attenuation of each channel more accurately than by modulating the I and Q components alone.

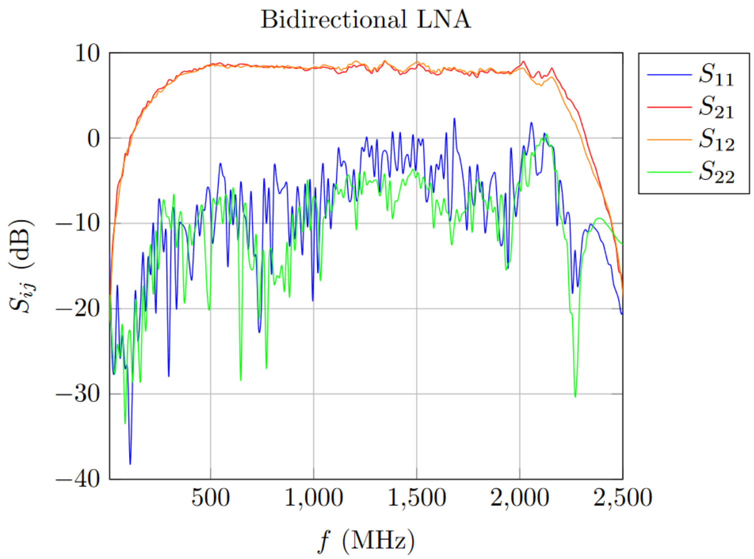


Figure 32. Bidirectional amplifier s-parameters without SIC tap.

The bidirectional amplifier was built and tested without SIC using 2 Watkins & Johnson WJ6203-12 LNAs and 2 Macom 2025-6002-10 Directional Couplers, as shown in Figure 32. The amplifiers provide gain in both directions but suffer from poor return loss on both ports, this is likely due to the fact that any incident signal on one port will be amplified on the channel in the reverse direction. Thus, in order to prevent creating an unstable oscillator, the gain within the loop must be less than the isolation of both couplers. Additionally, in order to achieve sub-zero dB return loss, the gain within the loop must be less than the isolation of the second coupler and the coupling factor of the first coupler, assuming ideal return loss and insertion loss of each component. If the transmit power at the RFID board is set to 0 dBm, then in order to achieve a +27 dBm transmit power at the antenna, the gain of the PA must be equal to the coupling factor of the second coupler plus 27 dB.

One of the primary difficulties of this design is that, according to the EPCglobal Gen2 Standard for UHF RFID, tags and readers must be capable of full-duplex communications. This means an RF switch cannot be used to simply turn off the inactive channel as is the case for half-duplex communications.