Distributed System for **L**ocalization **a**nd **M**apping (DLAM)

Capstone Final Report – Team B2

Spencer Brennessel, Ivan Kartashov, Arthur Kautz,

Joe Yang, Joshua Zak

Advisor: Bahram Shafai

Table of Contents

Abstract

The DLAM team has designed and built a system that implements real time localization and mapping using radio wave time-of-flight principles and trilateration algorithms. The purpose of DLAM is to provide accurate and precise mapping and tracking capabilities in environments not suitable for conventional localization systems, such as the satellite-based Global Positioning System (GPS), where performance is degraded due to line-of-sight obstructions. Additionally, the DLAM is intended to be an improvement on other time-of-flight localization systems that require the placement of known anchor nodes to effectively track tags within the bounds of the network. While these anchor-tag systems are already prevalent in museums, galleries, and high-end boutiques, their practicality is limited by the need for a fixed, stable environment suitable for anchor placements. In use cases where the environment is constantly in flux, such as construction sites, it is not feasible to constantly reestablish the anchor network whenever mounting points are displaced. The DLAM project is intended as a proof of concept for a system that can fill this gap, with nodes acting as either anchors or tags depending on movement and a dynamic mapping system that adapts whenever nodes are added or removed from the network.

To use DLAM and implement effective trilateration positioning in 3D space, a minimum of five modules are required, with four acting as static reference points at any given time. Each module employs the Qorvo DW1000 ultra-wideband (UWB) transceiver to calculate distances of nearby neighbors using time-of-flight algorithm, and the onboard Nordic Semiconductor nRF52832 integrated circuits will facilitate two-way data packet communication to share distance and location information between modules using Nordic's proprietary Enhanced ShockBurst (ESB) protocol. As stationary reference points are crucial for precise trilateration positioning, anchor behavior will be dynamically set based on motion feedback from an onboard microelectromechanical system (MEMS) triaxial accelerometer. To implement visualization and mapping, any module can be connected to a personal computer running a MATLAB script that will track movement within an X-Y-Z plot referenced on stationary modules.

For testing and validation, commercially available DWM1001 modules were used. While the DLAM team has custom designed a board and enclosure that will fulfill the robust form factor requirements necessary for deployment in harsh environments such as construction sites, including a power regulated battery charger that can support up to a 28 V input, the DWM1001 offers all

required onboard circuit components within a convenient package that can be powered by readily available CR123A batteries. Five DWM1001 modules were set up in the ECE Capstone lab in the basement of Hayden Hall, with four modules placed as stationary anchor points. The fifth module was connected to a laptop with the MATLAB script, allowing for its movement to be tracked in real time on a 3D graph plot. The use of ESB for two-way data packet communication is not particularly efficient, and while the nRF52832 supports Bluetooth Low Energy (BLE) mesh protocol, there are payload size limitations. It is the belief of the DLAM team that Thread or Zigbee protocols will be more effective in supporting two-way communications, though these technologies are not available on the chosen hardware.

Introduction

The proliferation of location-aware mobile computing has led to an increased demand for real-time location systems that can accurately locate and track objects or devices within an area of interest. Trilateration algorithms are a popular localization method, calculating the location of a point in space using the distances from the point to a series of known geometrical entities [1].

While hyperbolic trilateration technologies such as the satellite based Global Positioning System (GPS) have been readily available for decades [2], they are only suitable for precision applications within open outdoor environments due to signal degradation from the attenuation and scattering of microwaves by roofs, walls, and other objects [3] [4].

For localization applications in indoor or urban environments, trilateration using the time of flight (ToF) method can be applied to provide accurate and precise tracking capabilities. The ToF method measures the time it takes for a signal to travel from a transmitter to a fixed receiver and back to the transmitter [5]. Trilateration algorithms require a minimum of three non-collinear receivers in fixed locations for two-dimensional localization, and four receivers are required to locate the transmitter within a three-dimensional space [6].

While hardware intensive, ToF systems can provide reliable localization in applications where environmental factors can hinder system performance, making them ideal for tracking resources in indoors or environmentally unpredictable locations.

Problem Formulation

The goal of Distributed System for Localization and Mapping (DLAM) is to design a real time location system that can operate independently within environmental conditions not suitable for conventional technologies such as GPS. The system will be built from microcontroller transceiver modules that can communicate with each other and, through trilateration algorithm (Fig. 1) using the ToF method, create a graph theory map of all operational modules within the system that are tracked in real time on a guided user interface (GUI).

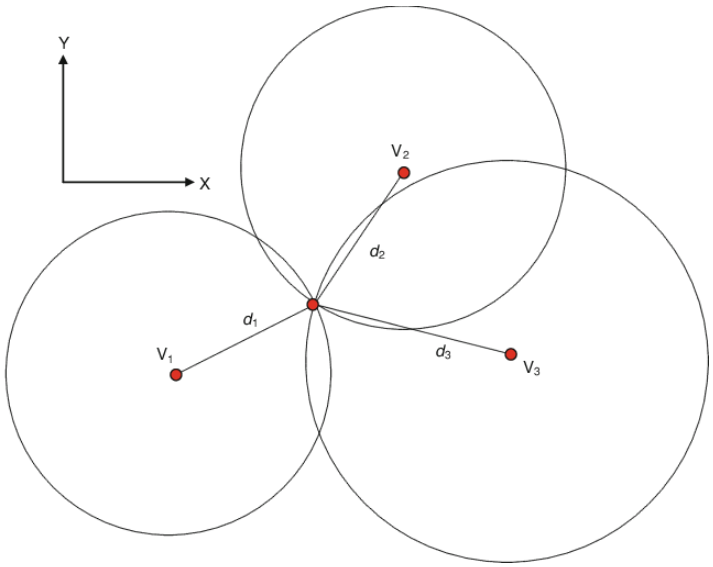

Fig. 1. Graphic representation of two-dimensional trilateration principle. [7]

Similar ToF systems have already been implemented in areas such as commercial asset tracking and self-guided tours, but these exist in controlled environments and rely heavily on a line-of-sight (LOS) approach where multiple stationary receivers placed at vantage points act as known anchors for the trilateration algorithm (Fig. 2) [8]. With DLAM, the system will be adaptive and capable of operating without fixed anchors. Each module can act as both transmitter and receiver for the ToF method depending on a current state set by feedback from an inertial measurement unit (IMU) within each module. While a set minimum number of modules will be required for the implementation of the trilateration algorithm, new modules can be introduced to the system on an ad hoc basis, allowing for dynamic scaling at the end user's discretion.

Fig. 2. Diagram of a LOS-based localization system for a museum. [9]

The intended purpose of DLAM is to provide a real time location system for industries operating in environmentally unpredictable locations where GPS localization is not reliable, one of such being the construction industry (Fig. 3). DLAM modules can be used to track tools and equipment to enhance material management at a site, allowing workers and managers to quickly locate assets with real-time localization. DLAM modules can also be used in conjunction with geofencing algorithms to ensure that equipment is not accidentally removed from the job site.

Fig. 3. Construction sites have numerous hindrances to accurate localization using GPS. [10]

Another industry that can benefit from DLAM is aviation maintenance (Fig. 4). Like construction sites, modern aircraft have numerous sources of interference that can reduce effectiveness of GPS systems, from complex avionics and electronics suites to metal hangar bays used for major maintenance tasks.

DLAM modules attached to test sets and tool kits can be used to provide quick accounting for all equipment utilized by maintenance crews at an aircraft, saving time and manpower needed to locate required assets. Additionally, the system will provide digital affirmation for both maintenance and flight crews that everything brought to the aircraft is removed before an aircraft is returned to service, reducing the likelihood for costly and potentially deadly foreign object debris (FOD) damage.

Fig. 4. Tool and equipment accountability is a top safety priority in aircraft maintenance. [11]

To provide real time localization system tracking capabilities for the complex environments using ToF method trilateration, DLAM will need to account for environmental challenges such as non-line-of-sight (NLOS) situations between nodes and the potential for multipath propagation effect due to obstacles and particulate matter. Additional considerations will need to be made to account for the magnetic fields generated by nearby electronic devices and the movement of objects and people. Since the system lacks fixed anchor points, robustness will be required for cases where there are fewer stationary modules than what is required for trilateration algorithm using the ToF method, resulting in a degradation of real time tracking accuracy.

The synchronization of DLAM ToF transceivers also poses an implementation challenge. While distance calculations using the ToF method is simple in theory, previous research on the topic showed that signal timestamps with nanosecond-precision are difficult to achieve and even slight clock offsets between the transmitter and receiver will result in noticeable errors in distance calculations [12].

Analysis of Major Systems

Module Overview

The base unit of the DLAM network is a node. Each node must be capable of communicating with neighbors in the mesh and performing ranging with other nodes. To support the deployment of the mesh the nodes should not require constant power from an external source.

For powering the nodes, an embedded battery provides the simplest user experience, though the size of the node limits the maximum capacity and recharging takes time and may involve removing the node from the mesh, external batteries can provide easy changes and arbitrarily large capacities. A common type of battery on construction sites are power tool batteries which are often have an output voltage around 20V and large capacities.

System Flowchart

To ensure system scalability, each DLAM module will be able to operate independent of other modules in the system (Fig. 5). After system startup and initialization, each module will listen for incoming data requests and upon trigger, ping neighboring modules to create and broadcast a list of known system nodes.

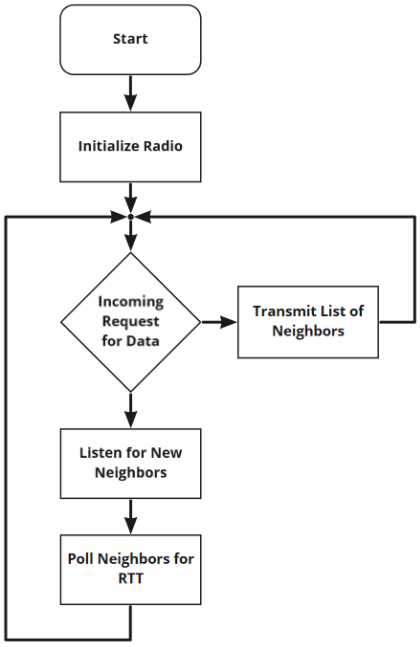

Fig. 5. General microcontroller flowchart.

Concurrently, each module will listen for responses from neighboring modules and poll them to establish the round-trip time (RTT) it takes for a signal to travel from the originating module to the target module and back. This RTT will allow for ToF method calculations to determine the distance between the two modules.

To establish a real time location system, DLAM will use the module distance data computed from the ToF method and apply hyperbolic trilateration like that used in GPS to determine the Cartesian coordinates of each module (Fig. 6).

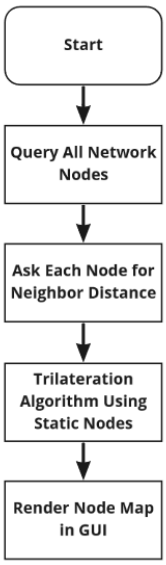

Fig. 6. General external map development flowchart.

The data is rendered as a node map on a MATLAB GUI using graph theory, with each module represented as a node and known distance measurements as edge weights, allowing end users to quickly identify and track each module in the system.

Time of Flight (ToF) Method

Once a DLAM module has established the presence of neighboring modules, the RTT between associated modules is measured (Fig. 7) and used to calculate the distance between modules using the ToF method.

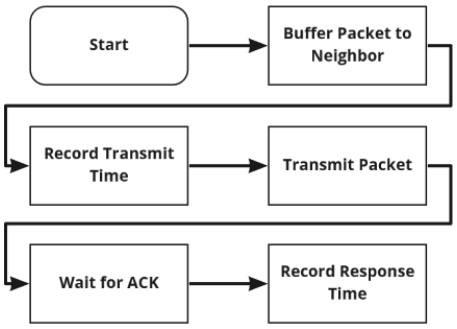

Fig. 7. RTT flowchart.

To measure the RTT required for the ToF algorithm (Fig. 8), the transmitting module will buffer packets, record the transmit time, and send the packet to a target module. Once the recipient module receives the packet, it will respond to the transmitting module with an acknowledgement message. The transmitting module will record the response time, which will allow for distance calculations using the ToF algorithm.

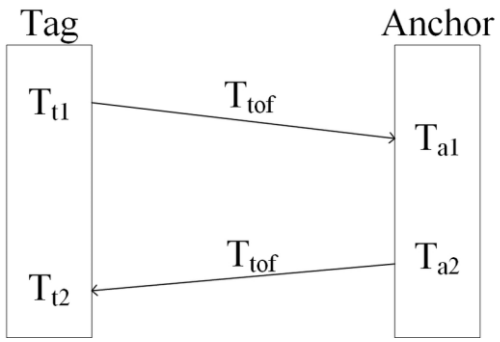

Fig. 8. ToF ranging algorithm principle. [13]

One potential obstacle to precise RTT measurements that will need to be accounted for are the delays from the enable time on the radio transceiver and the processing time required for the receiving module to acknowledge the transmission, though these should be constants. Additionally, considerations need to be made for instances when the radio on the receiver is in use at the time of transmission, and the RTT measurement will need to be timed out in the event of packet loss to avoid generating false distance data.

FreeRTOS

FreeRTOS is an open-source real-time operating system kernel for embedded systems microcontrollers that was originally developed by Richard Barry in 2003 [14]. It primarily uses the C programming language, making it easy to port across different platforms, and supports over thirty-five different microcontroller architectures. FreeRTOS features a real-time scheduler that supports multitasking, allowing for applications to be split into multiple independent tasks that share processor resources through rapid switching (Fig. 9) [15].

Fig. 9. FreeRTOS multitasking method. [15]

By allowing tasks to run in parallel, with time sensitive tasks taking precedence over lower priority ones, FreeRTOS makes an ideal operating system that can oversee the tight timing requirements needed to successfully implement the ToF method.

Ultra-Wideband (UWB)

Ultra-wideband (UWB) is a short-range wireless communication protocol that uses radio waves of short pulses to transmit information across a wide bandwidth greater than 500 MHz and is typically used in a spectrum of frequencies ranging from 3.1 to 10.5 GHz (Fig. 10) [16].

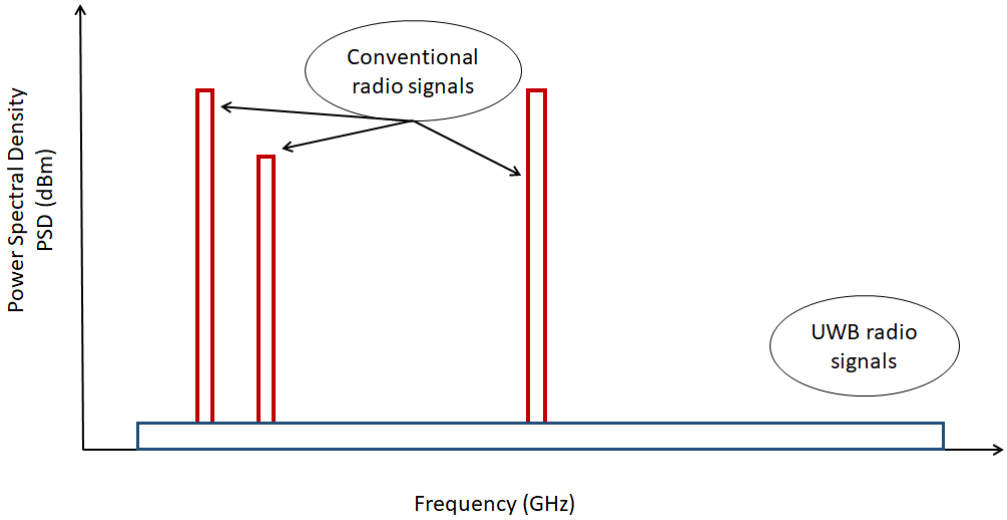

Fig. 10. Frequency comparison of UWB and conventional radio signals. [16]

Due to their wide bandwidths, UWB signals offer superior performance over traditional narrow-band systems in complex environments such as interior rooms, making it ideal for implementing the ToF method within the specified use cases of DLAM. Compared to other communication protocols such as GPS, Bluetooth, Wi-Fi, and RFID, UWB offers accuracy and precision at greater ranges and lower power consumption (Table 1) [17].

Table 1. Comparison of communication protocols. [17]

	UWB	**GPS**	**Bluetooth**	**Wi-Fi**	**RFID**
Battery	Low	Moderate	Low	Requires AC	None
Range	Up to 200 meters	No limit (outdoors)	Up to 70 meters	Up to 100 meters	Few centimeters
Accuracy	10 centimeters	5 meters	Up to one meter	Few meters	Few centimeters

Enhanced ShockBurst (ESB)

One of the many challenges that we faced when implementing our project design was how to send data between nodes in our network. Initially, we wanted to utilize the Bluetooth Mesh capabilities

of the nRF52832. This was because Bluetooth Mesh is an established protocol that offers a variety of network features such as retransmission and mesh flooding. Upon further investigation into Bluetooth Mesh, we realized during implementation that Bluetooth mesh would not fit our needs due to the small payload size of the data packets [18]. We then began looking into other possibilities for data transmission that were supported by our selected hardware. We finally selected a proprietary protocol called Enhanced ShockBurst(ESB), which was created by Nordic [19]. We selected ESB for a few reasons. The first was that our hardware already supported 2.4 GHz radio transmission using ESB, so implementation was straightforward within the hardware. The second was that ESB offered us a larger payload size of 31 bytes over Bluetooth Mesh advertising packets. We also selected ESB because we wanted to make sure that our data transmissions did not affect our UWB ranging. Thus, by using the 2.4 GHz radio with ESB we made sure that we would not deteriorate our UWB ranging signals.

DLAM Mesh Protocol

On top of the ESB protocol for packet broadcast we designed a packet structure to send the data (Fig. 11). Each packet fits within the 31-byte limit of ESB and is transmitted on a broadcast channel to all nearby nodes.

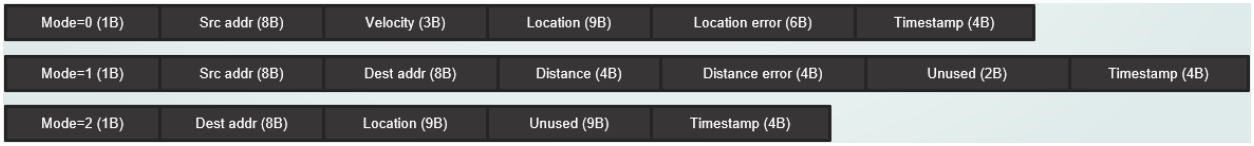

Fig. 11. DLAM data packet structures.

This data structure works in an intuitive way and allows us to send larger amounts of data while constrained by the payload size available to us. The first section of each packet type is the same, called Mode, it tells the receiving node what type of packet is to be decoded and read.

Mode 0 is used to update the status of a node on the network. It contains the address of the source node, the current velocity of the node, the location and error of the node if it is known, and the timestamp of the datapoint from the source node.

Mode 1 is used to share the distance between nodes. It contains the address of both nodes, the distance and its error, and a timestamp that matches the status update.

Mode 2 is used to set the location of a node to configure the initial state of the mesh to match an outside reference frame.

Each node keeps a fixed size list of nodes and the distances to their neighbors. Each packet can be parsed individually so the loss of a packet does not invalidate or corrupt the received packets. The timestamps are used to ensure that processed data is all from the same frame on the device. Retransmission is controlled by checking if the data from the received packet already exists in the list, so the devices do not waste bandwidth echoing packets multiple times. The data from the table is enough to reconstruct the geometry of the mesh.

Trilateration Algorithm

To enable decentralized localization, we must first find a way to locate and track a network of nodes in space while assigning them different coordinates. This is achieved by parsing distance data from each node the rest and generating a map. The most basic formulation of this problem can be stated as finding the coordinates of an unknown node given distance data to some other points with known coordinates.

The methods to achieve a solution of an unknown coordinate using distances to the relevant known points are called trilateration. In 2D space, the problem can be described as the intersection of three circles at a single point (Fig. 12) [20]. Three unique known coordinates and distances to the unknown point is the minimum requirement to get a unique solution. Given one distance the unknown point can be anywhere on a perimeter of a circle at a distance from the known point. Given two known points and distances narrows it down to the two points where two circles intersect, the third known point and distance creates a unique solution. Note that theoretically we assume that distance data is perfectly accurate as error in the data could either lead to infinite solutions or no solutions.

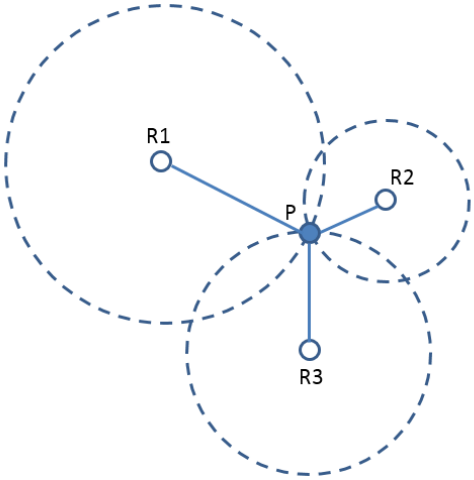

Fig. 12. Graphic representation of 2D trilateration.

The following equations describe the intersections of three circles at an unknown point (x, y), they are centered at coordinates (x_1, y_1), (x_2, y_2) and (x_3, y_3) with distance R_1, R_2 and R_3 to the unknown intersection point:

$$1.\ (x - x_1)^2 + (y - y_1)^2 = {R_1}^2$$

$$2.\ (x - x_2)^2 + (y - y_2)^2 = {R_2}^2$$

$$3.\ (x - x_3)^2 + (y - y_3)^2 = {R_3}^2$$

Expanding and subtract equation 2 from equation 1 and subtract equation 3 from equation 2.

$$(-2x_1 + 2x_2)x + (-2y_1 + 2y_2)y = {R_1}^2 - {R_2}^2 - {x_1}^2 + {x_2}^2 - {y_1}^2 + {y_2}^2$$

$$(-2x_2 + 2x_3)x + (-2y_2 + 2y_3)y = {R_2}^2 - {R_3}^2 - {x_2}^2 + {x_3}^2 - {y_2}^2 + {y_3}^2$$

The resulting equations may seem complicated but when we rename the constants, we get the following:

Let $Q = (-2x_1 + 2x_2)$

Let $R = (-2y_1 + 2y_2)$

Let $S = \left({R_1}^2 - {R_2}^2 - {x_1}^2 + {x_2}^2 - {y_1}^2 + {y_2}^2\right)$

Let $O = (-2x_2 + 2x_3)$

Let $P = (-2y_2 + 2y_3)$

Let $T = ({R_2}^2 - {R_3}^2 - {x_2}^2 + {x_3}^2 - {y_2}^2 + {y_3}^2)$

Which results in two linear equations with two unknowns.

$$x + Ry = S$$

$$Ox + Py = T$$

Therefore, the solution to this simple system of equations is:

$$x = \frac{SP - TR}{PQ - RO}$$

$$y = \frac{SO - QT}{RO - QP}$$

Following a similar methodology an analogous solution can be created for finding an unknown point in 3D space using four known coordinates and distances to the unknown point. Four unique coordinates and distances to the unknown point are required to get a unique solution in 3D space, where one point must not lie in the same plane as the first three (Fig. 13). This is a critical note if all known points lie on a plane, it creates ambiguity whether the unknown point some distance away is above or below the points in the plane. Effectively, the fourth known point and distance serve a similar purpose as the third known point and distance in the 2D trilateration algorithm where you eliminate doubt between two possible points and find a unique solution.

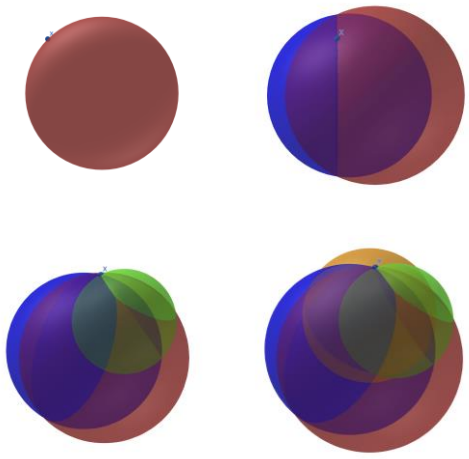

Fig. 13. Graphic representation of 3D trilateration.

The following equations describe the intersections of four spheres at an unknown point (x, y, z), they are centered at coordinates (x_1, y_1, z_1), (x_2, y_2, z_2), (x_3, y_3, z_3) and $(x_4, y_4, z_4$ with distance R_1, R_2, R_3 and R_4 to the unknown intersection point:

$$4.\ (x - x_1)^2 + (y - y_1)^2 + (z - z_1)^2 = {R_1}^2$$

$$5.\ (x - x_2)^2 + (y - y_2)^2 + (z - z_2)^2 = {R_2}^2$$

$$6.\ (x - x_3)^2 + (y - y_3)^2 + (z - z_3)^2 = {R_3}^2$$

$$7.\ (x - x_4)^2 + (y - y_4)^2 + (z - z_4)^2 = {R_4}^2$$

Expand and subtract equation 5 from equation 4, equation 6 from equation 5 and equation 7 from equation 6.

$$(-2x_1 + 2x_2)x + (-2y_1 + 2y_2)y + (-2z_1 + 2z_2)z = {R_1}^2 - {R_2}^2 - {x_1}^2 + {x_2}^2 - {y_1}^2 + {y_2}^2 - {z_1}^2 + {z_2}^2$$

$$(-2x_2 + 2x_3)x + (-2y_2 + 2y_3)y + (-2z_2 + 2z_3)z = {R_2}^2 - {R_3}^2 - {x_2}^2 + {x_3}^2 - {y_2}^2 + {y_3}^2 - {z_2}^2 + {z_3}^2$$

$$(-2x_3 + 2x_4)x + (-2y_3 + 2y_4)y + (-2z_3 + 2z_4)z = {R_3}^2 - {R_4}^2 - {x_3}^2 + {x_4}^2 - {y_3}^2 + {y_4}^2 - {z_3}^2 + {z_4}^2$$

The resulting equations may seem complicated but when we rename the constants, we get the following:

Let $A = (-2x_1 + 2x_2)$

Let $B = (-2y_1 + 2y_2)$

Let $C = (-2z_1 + 2z_2)$

Let $D = \left({R_1}^2 - {R_2}^2 - {x_1}^2 + {x_2}^2 - {y_1}^2 + {y_2}^2 - {z_1}^2 + {z_2}^2\right)$

Let $E = (-2x_2 + 2x_3)$

Let $F = (-2y_2 + 2y_3)$

Let $G = (-2z_2 + 2z_3)$

Let $H = \left({R_2}^2 - {R_3}^2 - {x_2}^2 + {x_3}^2 - {y_2}^2 + {y_3}^2 - {z_2}^2 + {z_3}^2\right)$

Let $J = (-2x_3 + 2x_4)$

Let $K = (-2y_3 + 2y_4)$

Let $L = (-2z_3 + 2z_4)$

Let $M = \left({R_3}^2 - {R_4}^2 - {x_3}^2 + {x_4}^2 - {y_3}^2 + {y_4}^2 - {z_3}^2 + {z_4}^2\right)$

Therefore, we get three systems of equations with three unknowns:

$$Ax + By + Cz = D$$

$$Ex + Fy + Gz = H$$

$$Jx + Ky + Lz = M$$

The solution to this system can be found using Cramer's Rule:

$$x = \frac{det\begin{vmatrix} D & B & C \\ H & F & G \\ M & K & L \end{vmatrix}}{det\begin{vmatrix} A & B & C \\ E & F & G \\ J & K & L \end{vmatrix}}, \qquad y = \frac{det\begin{vmatrix} A & D & C \\ E & H & G \\ J & M & L \end{vmatrix}}{det\begin{vmatrix} A & B & C \\ E & F & G \\ J & K & L \end{vmatrix}}, \qquad z = \frac{det\begin{vmatrix} A & B & D \\ E & F & H \\ J & K & M \end{vmatrix}}{det\begin{vmatrix} A & B & C \\ E & F & G \\ J & K & L \end{vmatrix}}$$

Note that if the bottom determinant is 0 then there are no solutions and if all determinants are 0 then there are infinite solutions.

Design Strategies and Approach

Nordic Semiconductor nRF52832 SoC

The node network that we are constructing needs to be able to handle substantial amounts of data, while simultaneously completing many operations per second. Thus, we needed to select a robust microprocessor that would be able to control these operations within each node in our network. The Nordic Semiconductor nRF52832 SoC (Fig. 14) was a great option to achieve this at a relative low unit cost. The Nordic SoC is a versatile Bluetooth 5.3 enabled chip that supports Bluetooth Low energy, Bluetooth mesh and NFC technologies for connection to peripheral devices [21].

Fig. 14. Nordic Semiconductor nRF52832. [21]

It has a 64 MHz Arm® Cortex-M4 with FPU, 512/256 KB Flash and 64/32 KB RAM. The chip supports many different digital interfaces including SPI, TWI, I2C, UART, PDM and QDEC. The chip also supports 2.4 GHz proprietary protocol which supports our project as the data sent between nodes was sent with 2.4 GHz Enhanced ShockBurst protocol. All these features enabled the nRF52832 to effectively work as the microprocessor for our project and handle multiple operations at one time, while simultaneously sending data between nodes in the network.

DW1000 Ultrawide Band Transceiver

Time of flight (TOF) ultra-wideband ranging is our selected method for determining the distances between nodes in our network. We needed to select an integrated circuit that had both ranging capabilities and a strong antenna to send and receive the ranging signals. To accomplish this, we have selected the DW1000 IC by Qorvo (Fig. 15). This chip is a 3.5-6.5 GHz Ultra-Wideband (UWB) Transceiver IC [22].

Fig. 15. The Qorvo DW1000 IC. [22]

The DW1000 is a powerful chip that offers dependable UWB ranging at a relative low price point and thus fits our needs well. It supports time of flight ranging to a precision of 10 cm, has low power consumption and an SPI interface to the host processor. The DW1000 is also an ideal choice since it is available in prebuilt modules that were used in the development of our project to enable simultaneous software and hardware development.

STM LIS2DH12TR Triaxial Accelerometer

A major component to our project being successful was the ability to know whether nodes were moving. To achieve this, we needed to select an accelerometer that would not draw too much power from the system. The accelerometer also needed to be accurate enough to determine the difference between vibration from environmental noise and actual movement of the node. The most cost-effective option to realize this in our project is the ST LIS2DH12TR. This chip is a digital 3-axis “femto” accelerometer that is ultra-low-power and high-performance [23].

The ST LIS2DH12TR offers motion detection, free-fall detection and an embedded temperature sensor. This is a nice added feature since this network was designed to work in all types of environments. Thus, a temperature safety feature is added to the product to prevent damage to nodes. The accelerometer also offers output data rates from 1 Hz to 5.3 kHz. Another reason we selected this accelerometer is because of the small form factor of this chip, it comes in at 2.0x2.0x1mm meaning that the chip does not utilize too much real estate on the PCBA.

Battery Power Source

Each DLAM module will require a portable power source in the form of a rechargeable battery along with a battery management system. Lithium-ion batteries are well suited for this application since they provide high energy density while maintaining a thin and light form factor [24]. For the DLAM form factor module, we chose a design that incorporates the commercially available 18650

battery due to their high performance, large capacities, and ruggedness. For ease of use in industrial environments, each DLAM module has a battery charging circuit that can accommodate up to the 28V common in commercial battery tool packs.

Outer Enclosure Design

As part of our design, we wanted to highlight what a final product might look like. This entails having our nodes geared towards being rugged enough to withstand the outdoor elements and any other type of contamination found at construction sites. To accomplish this, we designed an outer enclosure in Fusion 360 and then had it 3D printed to show case what the final product could look like (Fig. 16).

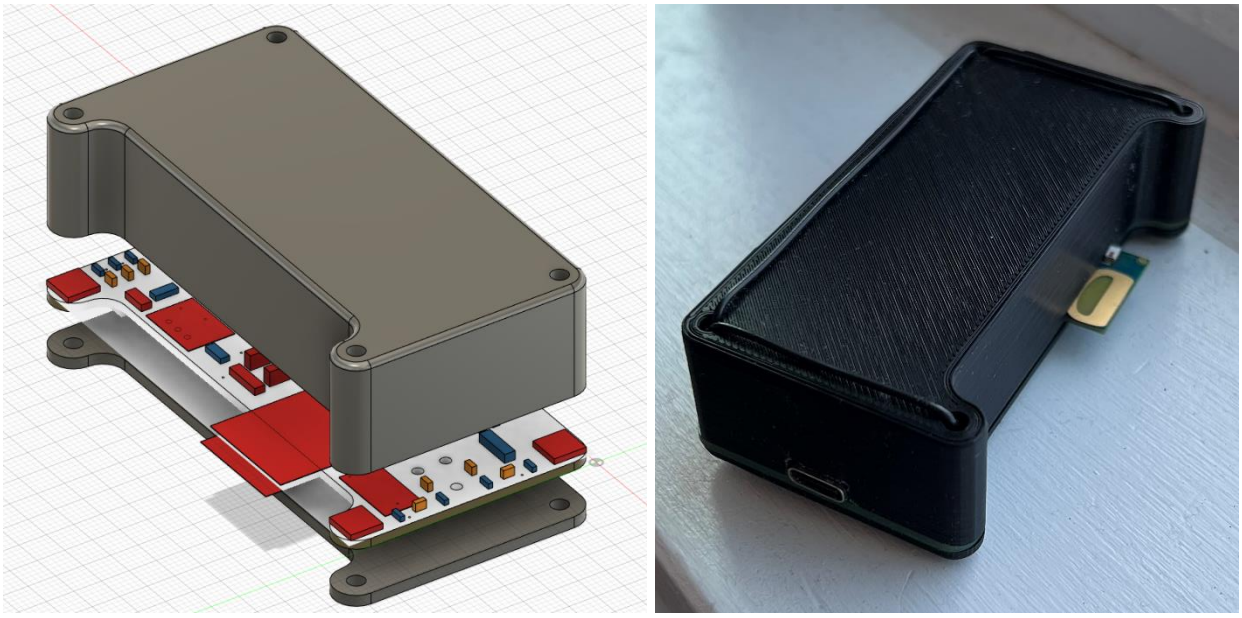

Fig. 16. 3D CAD model of enclosure and final 3D printed version.

The enclosure features a rugged construction that protects the internal PCBA and its components while also providing room for the antenna to protrude outside the enclosure for maximum signal strength. The case also allows for access to the J Link without needing to remove the enclosure. The final design would also likely have a rubber seal to prevent water or dust from getting into the enclosure. We would recommend the final case be design to IP65 rating.

Parts and Implementation

Form Factor Module Design

The node hardware is built around the DWM1001C module from Qorvo which incorporates the Qorvo DW1000 USB transceiver, Nordic nRF52832 Soc and a ST accelerometer and the necessary support components and antennas in a tested assembly [25]. Around this module our board provides a stable 3.3V supply, various inputs and outputs, and battery management (Fig. 17).

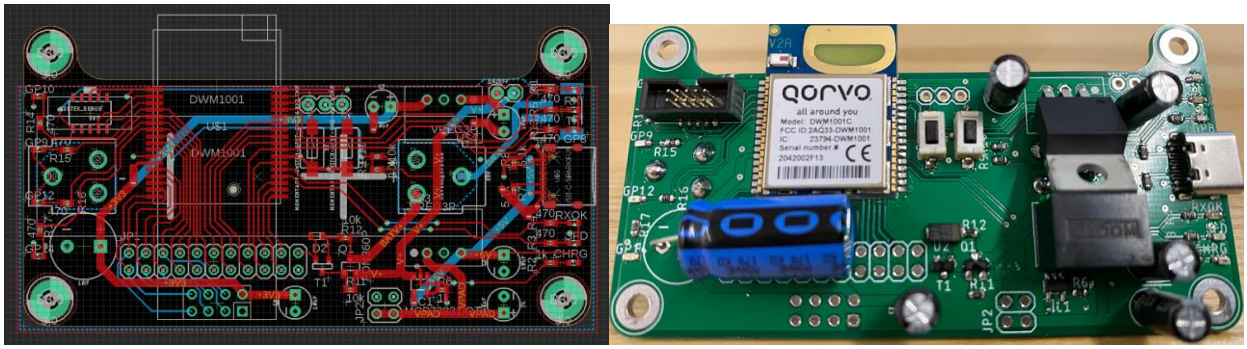

Fig. 17. Form factor board design.

The internal battery was managed by a MCP73831 charge controller. Two transistors, two resistors, and a diode were used to protect the module from reverse insertion of the battery. The 5V supply to charge the battery is supplied either from a USB-C port to connect to an AC power supply or from a switching regulator capable of handling an input of 28V connected to a commercial power tool battery pack.

Power for the module was take from the battery and regulated down the 3.3V by a linear voltage regulator and smoothed by two capacitors a small capacitor near the regulator and a larger bulk capacitor near the connection to the module.

The basic output of the module are 10 LEDs mounted on the two short ends of the board. Four of these LEDs are connected to the DW1000 IC and provide information about UWB activity. One is connected to the charge controller to show when the internal battery is fully charged. The remaining 5 LEDs are connected to the GPIO of the microcontroller for use by the device software. Default input to the board is provided by two pushbuttons, one connected to the hardware reset of the module and the other to the GPIO of the microcontroller (Fig. 18). For future expansion a 11x2

footprint for standard 0.1” header is connected to the remaining GPIO of the module. For programming the NRF52832 a header is provided to connect an external programmer.

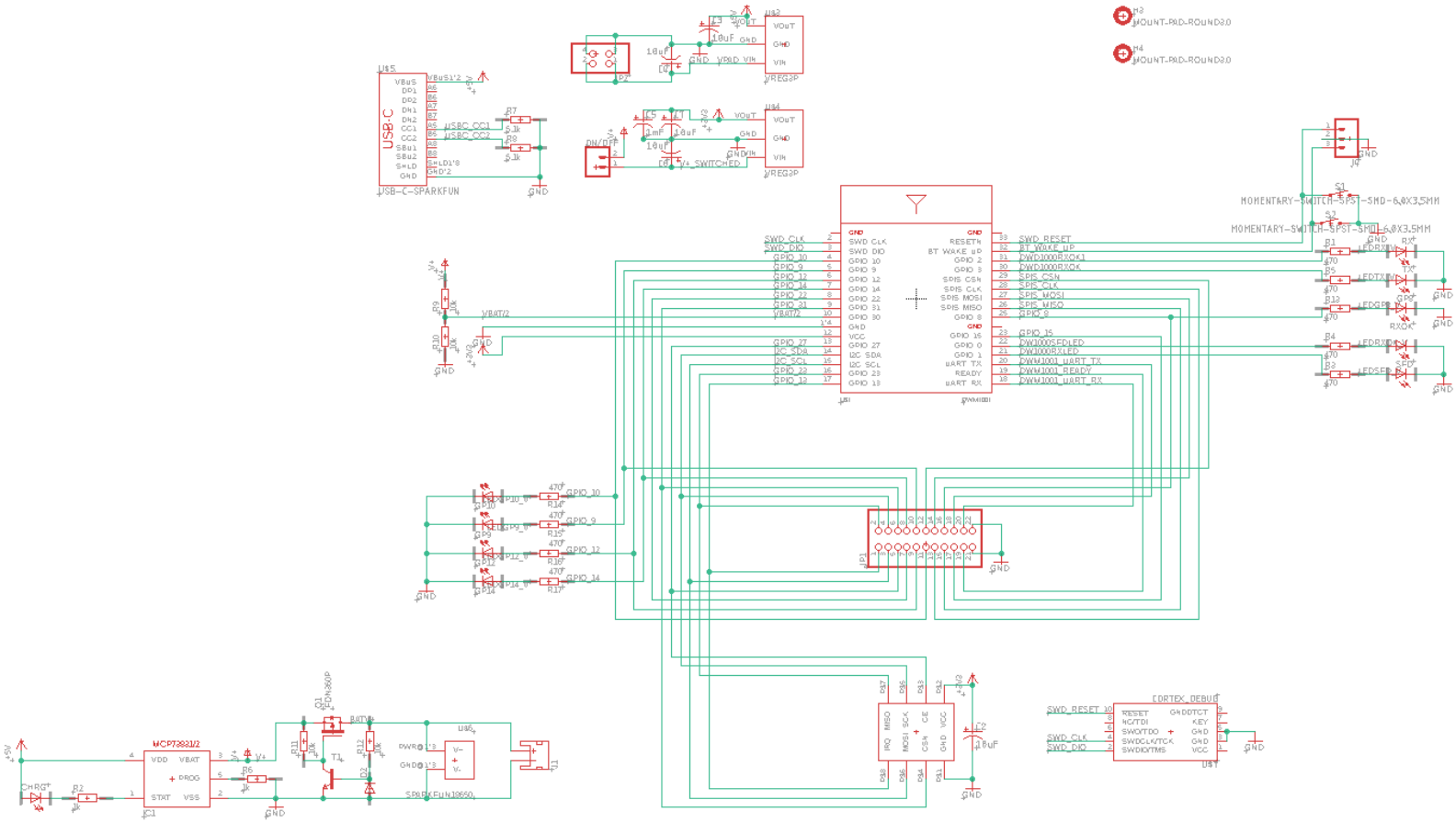

Fig. 18. Form factor board schematic.

MDEK1001 Module

To facilitate development, the DLAM team opted to purchase commercial off-the-shelf Qorvo DWM1001-DEV development boards. The DWM1001-DEV integrates the same DW1000 UWB transceiver and Nordic Semiconductor nRF52832 system on a chip (SoC) that was used in the form factor board (Fig. 19) [26], allowing for concurrent software development for both boards.

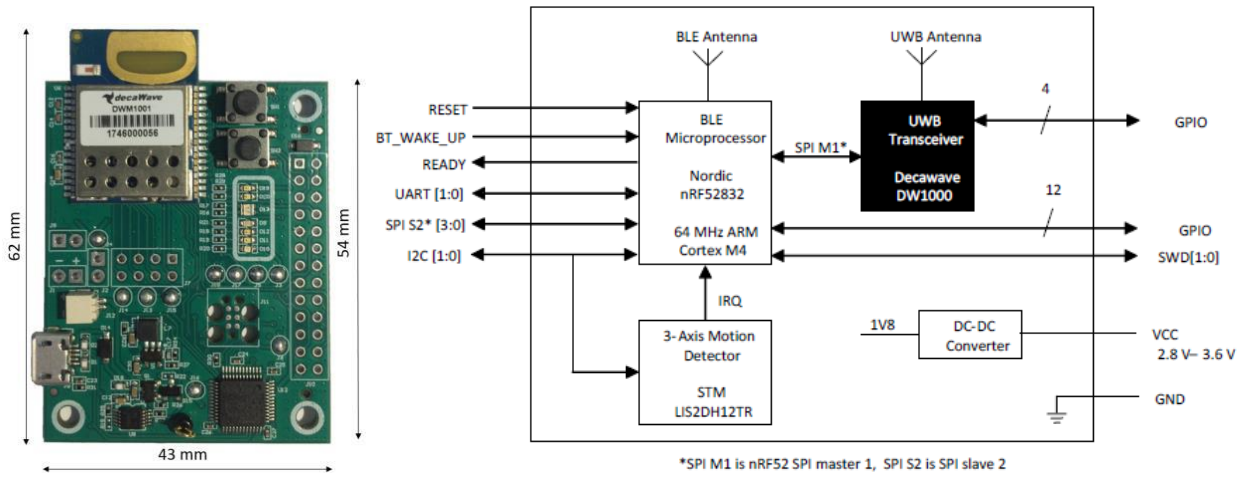

Fig. 19. DWM1001-DEV board and block diagram. [26]

While the DWM1001-DEV is not as robust as the form factor board, making it less suitable for deployment in unpredictable environments, it features an onboard J-Link debugger and can be powered by easily accessible CR123A batteries. The DWM1001-DEV also gave the DLAM team the convenience of concurrent asynchronous firmware development on the same hardware platform, negating a potential source of incompatibilities and permitting a rapid testing and development cycle.

Testing Procedures and Design Validation

Throughout the development of the system, we needed methods of testing the separate parts of our system, showing the design to others, and preparing demonstrations for the final presentation. To do this we began preparing many of these demos and testing gigs.

We used this method because we found it was better to develop a small part of a project as far as we can and then create a demo of these parts, before moving on to the rest of the system. For example, early in the project we got the Time-of-Flight system working well, and so we worked to get that system to run as smoothly as possible, then we created a demo using a python code which read out the distance measurements and displayed them (Fig. 20).

After this first demo, we wanted to demonstrate the ability for our distance measurements to be used to gather angle and other location information, so we created another demonstration. This demo used two of the initiator modules attached to a PC, where each initiator would repeatedly

send out range requests. Once each device had new distance data the data would be fed into a simple algorithm which would determine if the responding device were to the left, or to the right.

Fig. 20. Demo GUI with distance and orientation measurements taken with TOF.

The algorithm would then also display the average distance data between the two initiators to allow us to see not only the angle, but also the distance too. Using this we were able to get good assessment on how accurate we could depend on our system, and we could know how quickly the data would come in. We had concerns that as we were adding devices to the network, the time between successful ranges was dropping. With this system we were able to see how the network scaled and how likely it was that the system would work with all the devices connected.

After this demonstrator, we continued our development and managed to get larger numbers of devices connected, up to five devices at once. Again, we needed a way to be able to tell how fast the response time would be as we added more devices. This brought us to our next demo, the 3D location using known anchors demo (Fig. 21).

Fig. 21. Known anchors testing and validation setup.

This demo involved using four devices in locations we needed to know the precise location of, and then measuring the distance from the four devices to another one we were interested in knowing the location of. We did this by measuring the distance from the central device to all the others on the edge of the meter sticks, and then using the average of that distance value to act as the coordinates. This way the center device was at 0,0,0 and the others were at dist,0,0 : 0,dist,0 : 0,0,dist. Once we had the coordinate values for the devices, we were able to use a mathematical formula to turn the distance between each of these devices and the device we were looking for into an estimated location (Fig. 22).

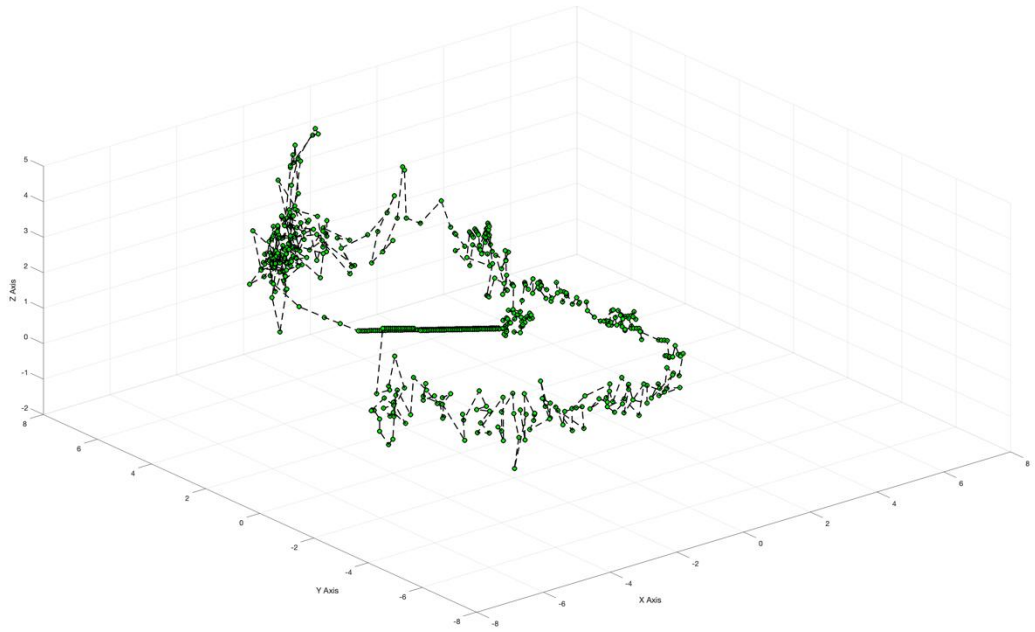

Fig. 22. GUI output from tracking one node in the network showing the path taken.

With this setup, we recorded a video of us walking around a backyard with an overhead shot of us walking, and then showed the data that we collected during this time to show what our estimated location was always. This demo was important because it ended up being the demo we showed at the presentation, but also it was a stress test for us to see how the system handled adding more devices to the network.

At all times in this process, we found that making demos and design validations were extremely helpful as it meant that we needed to test or tech as it stood, we needed to learn what went wrong with it in practice, and it meant we were constantly iterating our demonstrations so that when the time came for the final demo, we had a demo which worked well and looked clean.

External Map Development

The vision behind this project aimed to make tracking as simple and intuitive for the user as possible so the use of anchored known points for tracking was out of the question. However, given distance data between all nodes there should be a way to display a map of all of them if we assign them arbitrary coordinates. It should be noted that the map we define is only constrained by the distances between nodes and does not have an objective orientation as it is not grounded to a real-

life reference. Therefore, the user will be able to spin and rotate the map to fit the orientation they desire.

The resulting problem seems very challenging in principle as we have no foundation to base our coordinates on. However, we know that if we can get to three known or defined coordinates in 2D we can generate the rest of the points using trilateration algorithms.

Let us define the first node in the map as the origin and call it A (Fig. 23). Therefore, point A is located at $(0,0)$. Then we can define a second point C that is a distance AC from point in the x direction. Therefore, C is located at $(AC, 0)$. Now point B is some distance AB from point A and some distance BC from point C. This means that we have constructed a triangle with three sides AB, AC and BC.

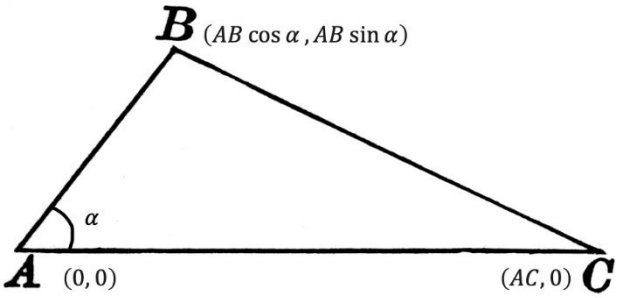

Fig. 23. Triangle used for defining three points in 2D space.

Using the cosine rule we can find the angle alpha which is the angle opposing side BC and find the x and y coordinates of B through simple trigonometry:

$$\cos\alpha = \frac{AB^2 + AC^2 - BC^2}{2AB * BC}$$

$$\alpha = \cos^{-1}\left(\frac{AB^2 + AC^2 - BC^2}{2AB * BC}\right)$$

Therefore, the x component of B is the base of the right triangle formed by drawing perpendicular line to the AC from point B. Furthermore, the y component of point B is simply the height of the same right triangle. From trivial trigonometry we get:

$$\sin\alpha = \frac{height}{AB}, \qquad AB\sin\alpha = height$$

$$\cos\alpha = \frac{base}{AB}, \qquad AB\cos\alpha = base$$

This results in point B having defined coordinates $(AB\cos\alpha\,, AB\sin\alpha)$. Using these three defined coordinates we can use trilaterate the remaining points in 2D space and create a 2D map of all points.

The 3D case of the same problem presents a new challenge as we can still define points A, B and C with ease as they all share a plane which we can assign to be $z = 0$. Therefore, point A is the origin at $(0, 0, 0)$, B is located at $(AC, 0, 0)$ and C is located at $(AB\cos\alpha\,, AB\sin\alpha\,, 0)$ (Fig. 24). However, point D must be out of the plane as described in Tracking and Trilateration Algorithms. This poses some real problems as D cannot be neatly pinned down above the triangle ABC it can be anywhere in 3D space barring the plane $z = 0$.

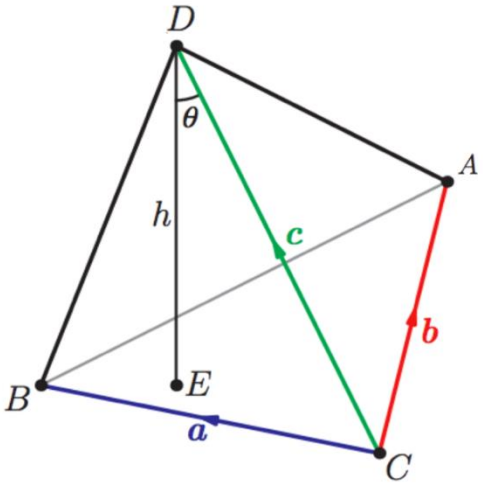

Fig. 24. Tetrahedron used for defining four points in 3D space.

However, we know that the four sides form a tetrahedron which has a volume that can be described by:

$$Volume\ of\ tetrahedron\ V = \frac{1}{3}AH$$

Where A is the Area of the base and h is the height of the apex point D. Therefore, we can rearrange this equation to solve for the height or the z coordinate of our point:

$$Height\ H = \frac{3V}{A}$$

Using Heron's Formula [27], we can find the area of the triangle ABC using just the side lengths:

$$s = \frac{AB + AC + BC}{2}$$

$$Area\ of\ triangle\ A = \sqrt{s(s - AB)(s - AC)(s - BC)}$$

Incredibly, there is an analog of Heron's Formula for the volume of a tetrahedron that uses all the side lengths of a tetrahedron to find its volume [28].

If we define the following constants, then we can find the volume by:

$$P = (AD - AC + CD) * (AC + CD + AD)$$

$$p = (AC - CD + AD) * (CD - AD + AC)$$

$$Q = (BD - AB + AD) * (AB + AD + BD)$$

$$q = (AB - BC + BD) * (AD - BD + AB)$$

$$R = (CD - BC + BD) * (BC + BD + CD)$$

$$r = (BC - BD + CD) * (BD - CD + BC)$$

$$a = \sqrt{p * Q * R}$$

$$b = \sqrt{q * R * P}$$

$$c = \sqrt{r * P * Q}$$

$$d = \sqrt{p * q * r}$$

$$V = \frac{\sqrt{(-\mathrm{a} + \mathrm{b} + \mathrm{c} + \mathrm{d}) * (\mathrm{a} - \mathrm{b} + \mathrm{c} + \mathrm{d}) * (\mathrm{a} + \mathrm{b} - \mathrm{c} + \mathrm{d}) * (\mathrm{a} + \mathrm{b} + \mathrm{c} - \mathrm{d})}}{192 * BD * CD * AD}$$

Once we found the volume, we can easily find the height and thus the z coordinate of point D. To find the x and y coordinate we can find point E directly below D through the right triangle that is formed with the base and the corners of the tetrahedron:

$$AE = \sqrt{AD^2 + H^2}$$

$$BE = \sqrt{BD^2 + H^2}$$

Now we have a triangle with three known sides and two known coordinates as we did in the original 2D case. Therefore, we can find point E through the same methodology as we found point B:

$$\cos\beta = \frac{AB^2 + AE^2 - BE^2}{2AB * BE}$$

$$\beta = \cos^{-1}\left(\frac{AB^2 + AC^2 - BE^2}{2AB * BE}\right)$$

This creates a familiar right triangle by creating a height from point E to base AB.

$$\sin\beta = \frac{height}{AB}, \qquad AE\sin\beta = height$$

$$\cos\beta = \frac{base}{AB}, \qquad AE\cos\beta = base$$

As a result, we can define point D by $(AE\cos\beta, AE\sin\beta, H)$. Using the four defined coordinates and distance data from them to unknown points we can use the 3D trilateration algorithms to create a map of all the remaining nodes in the system.

Cost Analysis

To implement our design, the DLAM team decided to purchase commercial off-the-shelf Qorvo DWM1001 development boards. Each DWM1001 came with the same transceiver and SoC components we originally selected for our form factor boards at a fraction of the cost, meaning we could undertake concurrent firmware development on proven hardware. We ultimately purchased fifteen DWM1001 boards to ensure that every member has enough hardware on-hand for testing and evaluation.

Since at this point, we still had more than half of our original budget left, we decided to build several form-factor boards so that we have hardware to show during the capstone competition. We also designed and 3D printed a prototype board enclosure. While this enclosure did not meet ingress protection standards and would therefore be unsuitable for deployment in our intended use cases, it nevertheless allowed us to showcase our vision for the final design of a DLAM module.

Table 2. DLAM project cost.

Quantity	Name	Description	Manufacturer	$ Part	$ Total
15	DWM1001	Development Board	Qorvo	$19.50	$292.50
5	DW1000	Radio Transceiver	Qorvo	$7.80	$39.00
5	nRF52832	Multiprotocol SoC	Nordic Semiconductor	$5.50	$27.50
5	LIS2DH12TR	ST accelerometer	ST Microelectronics	$2.11	$10.55
5	18650 Battery	Lithium-Ion Battery	Samsung	$5.99	$29.95
1	Miscellaneous Electronics	Capacitors, resistors, etc.	Multiple	$90.00	$90.00
1	3D Printing	Module Enclosure	N/A	$5.00	$5.00
		Total Project Cost	$494.50		

Conclusion

This report outlines the design and implementation for the Distributed System for Localization and Tracking, or DLAM. By decentralizing the real time localization system and negating the need for fixed anchor points, DLAM is meant to provide precise and accurate tracking and mapping in environments not suitable for satellite-based systems, such as GPS, or traditional anchor-tag networks. The DLAM project demonstrates effective implementation of the ToF method for accurate distance calculations between nodes, which are then used with trilateration algorithms to calculate node locations in 3D cartesian space. Data between nodes is shared using a mesh created from Nordic Semiconductor's proprietary Enhanced ShockBurst protocol. The promising results observed by the DLAM team through extensive testing validates the proof of concept and showcases the feasibility of real-world implementation for a distributed localization system.

Compliance and Final Remarks

Since the DLAM project uses ultra-wideband radio protocol to implement time of flight ranging, we ensured that the components used in each DLAM module, specifically the Qorvo DW1000 radio transceiver, are IEEE 802.15.4 compliant. As our target use cases are largely industrial, we also took guidance from the Ingress Protection Code defined in IEC 60529 in our design for the board enclosure.

While we were able to implement our design for DLAM on schedule, we did run into difficulties with hardware purchases due to ongoing global supply chain issues and the economic conflicts. This was particularly evident when we were sourcing the DWM1001 development boards, as multiple retailers had them on backorder or required exorbitant tariff fees to ship from overseas warehouses. We did eventually find a California-based retailer that had the boards in stock, though they imposed order limits for individual purchases, requiring our team to split the order amongst multiple members.

References

[1] P. Cotera, M. Velazquez, D. Cruz, L. Medina and M. Bandala, "Indoor Robot Positioning Using an Enhanced Trilateration Algorithm," *International Journal of Advanced Robotic Systems,* vol. 13, no. 3, 2016.

[2] A. Kleusberg, Geodesy - The Challenge of the 3rd Millennium, Berlin: Springer, 2003.

[3] National Coordination Office for Space-Based Positioning, Navigation, and Timing, "GPS Accuracy," United States Government, 22 April 2020. [Online]. Available: https://www.gps.gov/systems/gps/performance/accuracy/. [Accessed 13 June 2021].

[4] W. M. Y. W. Bejuri and M. M. Mohamad, "Performance Analysis of Grey-World-based Feature Detection and Matching for Mobile Positioning Systems," *Sensing and Imaging,* vol. 15, no. 95, 2014.

[5] C. Poellabauer and W. Dargie, Fundamentals of Wireless Sensor Networks: Theory and Practice, New York: Wiley, 2010.

[6] University of Tasmania, "How GPS Works," 2014. [Online]. Available: https://www.gtav.asn.au/documents/item/422. [Accessed 12 June 2021].

[7] F. Franceschini, M. Galetto, D. A. Maisano, L. Mastrogiacomo and B. Pralio, Distributed Large-Scale Dimensional Metrology, London: Springer-Verlag, 2011.

[8] P. Langer, "Localization system of an autonomous mobile device," 2010. [Online]. Available: https://is.muni.cz/th/euja8/LangerDiplomaPrintable.pdf. [Accessed 12 June 2021]

[9] R. Giuliano, G. C. Cardarilli, C. Cesarini, L. D. Nunzio, F. Fallucchi, R. Fazzolari, F. Mazzenga, M. Re and A. Vizzarri, "Indoor Localization System Based on Bluetooth Low Energy for Museum Applications," *Electronics,* vol. 9, no. 1055, 2020.

[10 R. J. Driscoll, "Site-Structural Engineering for the Urban Environment," 29 March 2018. [Online]. Available: https://www.richardjdriscoll.com/2018/03/site-structural-engineering-for-the-urban-environment/. [Accessed 19 June 2021].

[11 Dviation Blog, "The Dangers of Aircraft Maintenance," Dviation, 14 November 2018. [Online]. Available: https://blog.dviation.com/2018/11/14/the-dangers-of-aircraft-maintenance/. [Accessed 19 June 2021].

[12 R. Dobbins, S. Garcia and B. Shaw, "Software Defined Radio Localization Using 802.11-style Communications," 2011. [Online]. Available: https://web.wpi.edu/Pubs/E-

project/Available/E-project-042811-163711/unrestricted/NRL_MQP_Final_Report.pdf. [Accessed 12 June 2021].

[13 X. Hu, Z. Luo and W. Jiang, "AGV Localization System Based on Ultra-Wideband and Vision Guidance," *Electronics,* vol. 9, no. 448, 2020.

[14 FreeRTOS, "About FreeRTOS," OpenRTOS, [Online]. Available: http://www.openrtos.net/RTOS.html. [Accessed 23 April 2022].

[15 admin, "Using FreeRTOS kernel in AVR projects," ScienceProg, [Online]. Available: https://scienceprog.com/using-freertos-kernel-in-avr-projects/. [Accessed 23 April 2022].

[16 M. Bakr, "Introduction to Ultra-Wideband (UWB) Technology," All About Circuits, 19 March 2020. [Online]. Available: https://www.allaboutcircuits.com/technical-articles/introduction-to-ultra-wideband-uwb-technology/. [Accessed 23 April 2022].

[17 Bleesk, "Ultra-Wideband (UWB)," Bleesk, [Online]. Available: https://bleesk.com/uwb.html. [Accessed 23 April 2022].

[18 Bluetooth, "Bluetooth Mesh Networking," Bluetooth, [Online]. Available: https://www.bluetooth.com/learn-about-bluetooth/recent-enhancements/mesh/. [Accessed 23 April 2022].

[19 Nordic Semiconductor, "Enhanced ShockBurst User Guide," Nordic Semiconductor, 04 October 2016. [Online]. Available: https://infocenter.nordicsemi.com/index.jsp?topic=%2Fcom.nordic.infocenter.sdk5.v12.0.0%2Fesb_users_guide.html. [Accessed 23 April 2022].

[20 101 Computing, "Cell Phone Trilateration Algorithm," 101 Computing, 29 March 2019. [Online]. Available: https://www.101computing.net/cell-phone-trilateration-algorithm/. [Accessed 24 April 2022].

[21 Nordic Semiconductor, "nRF52832," Nordic Semiconductor, [Online]. Available: https://www.nordicsemi.com/products/nrf52832. [Accessed 22 April 2022].

[22 Qorvo, "Qorvo DW1000," Qorvo, [Online]. Available: https://www.qorvo.com/products/p/DW1000. [Accessed 22 April 2022].

[23 ST, "LIS2DH12," ST, [Online]. Available: https://www.st.com/en/mems-and-sensors/lis2dh12.html. [Accessed 22 April 2022].

[24 Adafruit, "Lithium Ion Polymer Battery - 3.7v 100mAh," Adafruit Industries, [Online]. Available: https://www.adafruit.com/product/1570. [Accessed 19 June 2021].

[25 Qorvo, "DWM1001C," Qorvo, [Online]. Available: https://www.qorvo.com/products/p/DWM1001C. [Accessed 23 April 2022].

[26 Qorvo, "DWM1001-DEV Ultra-Wideband (UWB) Transceiver Development Board," Qorvo [Online]. Available: https://www.qorvo.com/products/p/DWM1001-DEV. [Accessed 23 April 2022].

[27 K. Kendig, "Is a 2000-year-old formula still keeping some secrets?," *The American Mathematical Monthly,* vol. 107, no. 5, pp. 402-415, 2000.

[28 W. Kahan, "What has the Volume of a Tetrahedron to do with Computer Programming Languages," Mathematics Dept., and Elect. Eng. & Computer Science Dept., Berkley, California, 2001.

[29 DSM&T, "IP Rating Chart," DSM&T, [Online]. Available: https://www.dsmt.com/resources/ip-rating-chart/. [Accessed 19 June 2021].