# Installation and Testing of a Positioning System in the Context of Autonomous Drone Flight

Petter Nordin,
Independent Study
Course: T-749-INDS
*Reykjavik University*
pettern21@ru.is

***Abstract*—This paper presents a study made into implementing a reliable non-GPS navigation system for drones that uses a Real-Time Localization System made of DWM1001 modules from Qorvo/Decawave. This navigation system will be implemented in a courtyard at Reykjavik University and this study presents the steps taken to implement this system.**

## I. Introduction

Developing solutions to address drone navigation in GPS-denied environments is a hot research area. The paper [1] presents a solution for drone navigation in a GPS-denied environment using stereo vision to build a 3D representation of the surroundings and navigate the drone based on the observations from the stereo vision. In [2], they present an approach for drone navigation that uses visual input and CNN to follow a predetermined path without using GPS. Several solutions based on RF localization systems have been used to develop UAV navigation systems such as Wi-Fi, Bluetooth, UWB and more. In [3], they present a precise UAV position estimation and collision avoidance system that is Wi-Fi based. The paper [4] uses Bluetooth beacons and RSS measurement for UAV navigation in a GPS-denied environment. The survey made in [5] presents different types of RF-based localization systems that have been used to develop UAV navigation systems in GPS-denied environments.

In this study, the aim was to implement a reliable non-GPS navigation system for autonomous drone flight that will give access to the possibility to fly an autonomous drone inside a closed area where GPS is not reliant for drone navigation, whereas this navigation system will later be used as a way to test and deploy different autonomous drone navigation algorithms.

But the scope of this study will be to only implement a non-GPS navigation/position system, which will be implemented in an outdoor courtyard at Reykjavik University with a set of DWM1001 modules from Qorvo/Decawave [6] that implements a Real-Time Location System (RTLS) using UWB and Bluetooth.

This study will address the hypothesis, is it possible to transform the DWM1001 modules' Real Time Location System (RTLS) [7] into autonomous drone navigation system and can it be implemented in the outdoor courtyard.

## II. Related Work

The DWM1001 RTLS network uses the radio frequency technology, Ultra Wide Band (UWB) for localization, whereas using this technology in localization system have been successful. The survey [5] present an overall view of the different drone localization systems using different radio frequency technologies including UWB, such as presented in [8], [9], [10], [11] and [12].

## III. Methodology

This section will address the methods used to investigate the possibility to implement the non-GPS navigation system for drones in the courtyard at Reykjavik University.

### A. DWM1001 RTLS

A set of DWM1001 modules can be used to implement a two-way ranging real-time location system (RTLS) that uses Ultra Wide Band and Bluetooth for communicating between the modules. A module can be configured as either an anchor or a tag. An anchor behaves as a fixed node in the system with an x,y,z coordinate, where one of the anchors used in a system must be set as the initiator anchor. The tag behaves as a mobile node in the system of anchors and calculates its location in the system based on the anchors locations. An example of how the DWM1001 RTLS (DRTLS) network communicates can be seen in fig 1. The RTLS network that these modules implement will be used to develop the Non-GPS navigation system, where a tag will be mounted on a drone that will use GPS coordinates that are calculated from the tag position within the RTLS network. For further information on the RTLS, it is recommended that the reader read the system overview of DWM1001 [7].

### B. Research Questions

A set of research questions were constructed for the purpose of evaluating and analyzing potential factors that might affect the implementation of the non-GPS navigation/position system and evaluating what precision in GPS location the RTLS could provide for the navigation system. The identified factors that could affect the implementation of the navigation system that was to be investigated were the surrounding environment of the courtyard and how the DWM1001 module is affected by

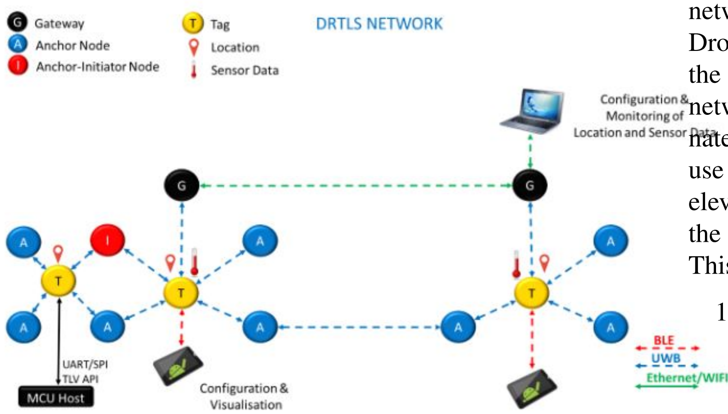


Fig. 1: DW1001 RTLS [source]

the courtyard. The courtyard consists of four walls made of metal coated class windows and metal plate windows in a rectangular shape, as shown in fig 2. These metal coated windows are made for the purpose of decreasing/disabling signal throughput, so it wouldn't interfere with the signals of the nearby airport. For this reason, these metal coated windows needed to be investigated to see how they affected the signal strength (RSSI) of the anchors when placed on the windows either indoors facing the courtyard or outdoors in the courtyard. The metal coated windows were identified as the main source that could affect the signal strength of the anchors in an undesirable way, which would therefore lead to challenges in implementing the navigation system in the courtyard. But nothing could be said about how the general construction of the courtyard might affect the signal of the anchors. How could the quality and accuracy of the anchors location data be affected by the courtyard? And if that were the case, how would it affect the implementation of the navigation system? As this was more of an open question to identify factors during the duration of the work, it could be the case that multiple factors could be found, but also the opposite, no factors found.

In order to evaluate what precision in GPS location the RTLS could provide for the navigation system, the tag (x,y,z) location estimation needed to be investigated first. For the tag location, it is possible to choose among the sampling frequencies of 10 / 5 / 2 / 1 / 0.5 / 0.2 / 0.1 / 0.03 / 0.017 Hz. The higher the frequency chosen, the more tag location estimations are generated per second, and the smaller the frequency, the fewer tag location estimations are generated per time unit (sec or min). When choosing 10Hz as the sampling frequency, 10 locations are estimated per second, leading to the tag location changing frequently. An approach to addressing this is through the use of filtering methods to potentially get more reliable tag location estimations. But how about lowering the sampling frequency of the tag location? Would that lead to a more reliable location and does it need more filtering than what is already provided for the tag location estimation in the RTLS network?

Drones navigate with the use of GPS, and for implementing the navigation system with the use of the DWM1001 RTLS network, the tag locations must be converted to GPS coordinates. But how precise is this GPS coordinate? Can the drone use it to navigate within the courtyard? Does the variation in elevation or altitude affect the GPS coordinate calculated from the RTLS network?

This led to the following research questions,

1) How reliable is the anchor signal strength when placed on metal coded windows indoors versus placing them on windows outdoors? Do we get too much signal interference from the metal coded windows if the anchors are placed on the windows indoor facing the outdoor courtyard?
2) Does the general construction of the outdoor grid map interfere with the anchor signal? Does this lead to inaccurate positioning data from the anchors?
3) Is it possible to get more reliable position estimates when using lower frequencies for the anchors? And is it possible to use filtering algorithms to get even better position estimates?
4) How precise are the GPS coordinates that the drone gets from the position system in the grid map? Has the deviation in elevation any impact on the position system leading to the GPS coordinates?
5) Can the position system be used as a autonomous drone navigation system? What about the mean absolute error? Gives it good enough precision for autonomous drone flight?

### C. Tasks

To be able to answer the research questions, a set of tasks were constructed to setup the navigation system. These tasks were divided into two phases, named Installation and Drone Flight Adaption.

The tasks of the installation phase consisted of preparing the courtyard for the DWM1001 RTLS that would be integrated into the drone navigation/positioning system. The tasks were to measure the courtyard walls, test the anchors and the tag in the courtyard, and test and evaluate the signal strength of the anchor when placed on the metal coated windows indoors and outdoors facing the courtyard.

The tasks of the Drone Flight Adaption consisted of converting the tag location of the DWM1001 RTLS network to GPS coordinates that could be used by the drone, integrating the GPS coordinates into the autopilot and ground control station used, testing the navigation system by hovering the drone, and then evaluating the precision of the navigation system and the drone.

The following sections will describe the steps involved in completing the set tasks. Due to time limitations, the task of testing and evaluating the navigation system with the drone hovering wasn't completed.

### D. Installation: Courtyard Measuring

The courtyard, located parallel to Malid Restaurant at Reykjavik University, was used to setup the navigation system. This courtyard consisted of four walls forming a rectangular shape with metal coated class windows and metal windows, where the walls were named Eating Area, Piano, Hallway, and Malid Wall. The walls can be seen in figure 2. The steel measure tape,

Fig. 2: Eating Area (up-left), Piano (up-right), Hallway (down-left), Malid (down-right)

500 Spring-Loaded Return Tape Measure 5m from Kapro, was used for measuring the walls [13]. The measuring tape is class 2 accuracy tape, which means that for this 5m tape, the accuracy is +/- 1.3mm when the temperature is 20° and a pull force of 50N is used [14]. As a result, if the tape is used under conditions other than this, the tape's accuracy cannot be guaranteed. But to resolve this, tape corrections can be used to correct measurement when it deviates from preferred conditions [15]. Unfortunately, the Reykjavik climate deviates from the preferred conditions of the tape, and this wasn't considered during the wall measurements. So, the resulting measurements shouldn't be considered correct, but the method of measuring the walls can be considered as a way to follow or find a better solution to measure the walls of the courtyard. The walls consisted of a set of windows (both class and metal), window frames (WF) between the windows, and areas that were designated as extra walls (EW) which were placed at the start and end of each wall. All of these wall elements were measured one by one, to get the measure of each one and also to add them up to a total length. The measurement of each wall can be seen in Table I. Due to the fact that the anchors can only take x,y,z positions in meters with 2 decimals, each wall element measurement was converted from centimeters to meters, and for the 2 decimal places, the measurement was rounded to the closest whole decimal meter, e.g 147.67 cm to 1.48 m. Because of this unit conversion and because the measurements were conducted by a human, measurement error will surface, therefore lowering the precision of the navigation system. As this conversion was done to follow the unit system in the anchors, a potential solution to counteract the lost precision might be to add a justified error value to some of the wall elements or to the anchors locations. But if the error is very low, then the lost precision might be accepted. Then a better solution would be to measure the walls more precisely, considering the tape corrections, keeping the centimeter measurement and doing the necessary conversion for the anchors. The RTLS's accuracy/error and precision of position have been reported to be approximately 10 cm and 1 cm, respectively, which should be taken into account when placing the anchor positions [7], [16]. A challenge that arose when doing the measurements was that the courtyard longsides, Malid and Hallway are not straight. They have a bit of an arc, which can be seen in figure 3. Because of this construction of the longsides, getting an accurate location, especially in the y-axis (fig 3), to place the anchors would be hard. An investigation to use Apriltag markers tag36h11 [17] as an approach to solve this was implemented. The feature of these Apriltags is that it is possible to get a 3D transformation matrix of the markers in the world coordinate frame from

| Wall | Length(m) | Windows | EW | WF |
|---|---|---|---|---|
| Eating Area | 8.39 | 6 | 2 | 7 |
| Piano | 8.39 | 6 | 2 | 6 |
| Hallway | 17.82 | 12 | 2 | 13 |
| Malid | 16.39 | 12 | 2 | 13 |

TABLE I: Wall measurements & Elements.

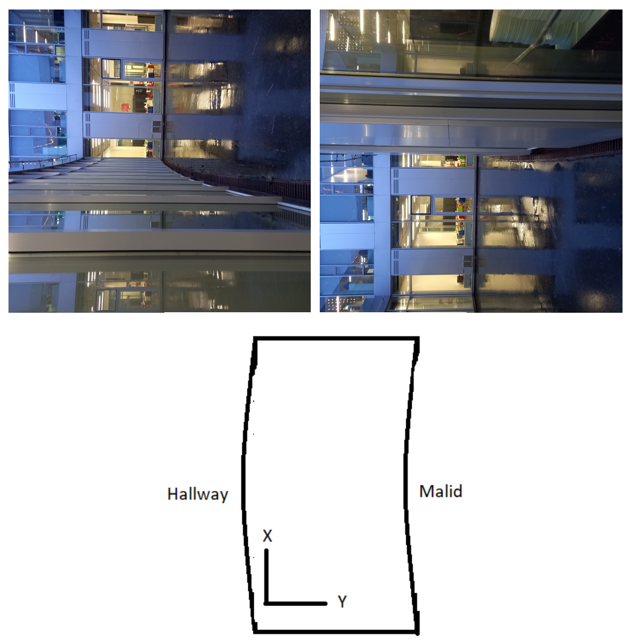


Fig. 3: arc-curved longsides, Hallway(Left), Malid(Right)

images. So the idea was to use these markers to map the courtyard with coordinates for easier placement of the anchors in the courtyard. The idea was implemented in the courtyard by putting markers along the four walls in a straight line. They were set at the beginning of each window, at the corners, and some reference markers were set close to the corners and middle. These reference markers were put there to validate distances between a reference marker and a marker. This validation was done and compared by calculating the distance between the markers' poses using the distance formula [18] and measuring the distance by hand with a tape measure [13]. Images were taken of the courtyard, containing the Apriltag markers, from above by a camera with auto-focus turned off, as seen in fig 4. The cameras used for this were a hand camera, a Canon Rebel T5 standard lens, and the camera on the drone, a DJI Spark [19]. When a bunch of images had been taken of the courtyard(see fig 4 and 5), camera calibration [20] was done by taking a bunch of images of a chessboard at different orientations to get the intrinsic parameters of the camera, the focal lengths $f_x, f_y$ and the optical centers $c_x, c_y$. Then the images of the courtyard containing the Apriltag markers were inserted into a python script to detect the markers (see fig 4), to extract the pose and rotation matrix of the markers [21], [22]. When the poses were extracted, they were to be used to construct a flat plane, where a new origin would be chosen in this plane and polar coordinates would be constructed from the data points relative to the new origin. Then these polar coordinates would be converted back to local Cartesian coordinates that model a coordinate system with one quadrant. If this were successful, these coordinates would be used as reference to set the locations of the anchors in the courtyard, but also to get the difference in the y-axis of the courtyard's long sides, as seen in fig 3.

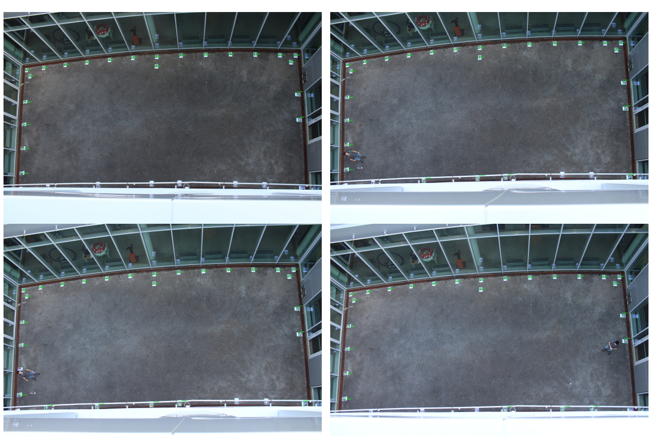

Fig. 4: Courtyard with apriltags, hand camera

Fig. 5: Courtyard with apriltags, drone camera

### *E. Installation: Test Anchors/Tag in Courtyard*

A small RTLS network of 6 anchors and a tag was setup in a section of the courtyard, following the quide [23]. The anchors were placed on the metal coated class windows of the Malid, Piano, and Hallway Wall with some measured x,y,z locations, whereas one of the anchors was set as the initiator anchor. This experiment was done for the purpose of evaluating the location of the tag inside the network and what tag setting should be used to get a tag location that is more stable and transitions smoothly. The tool used to evaluate the tag location and settings was the Android application Decawave RTLS Manager [23]. The experiment was conducted by testing the location of the tag by enabling and disabling the tag option stationary detection and placing the anchors at different heights, i.e., same height and different heights.

### *F. Installation: Anchor RSSI outdoor vs indoor comparison*

For the implementation of the navigation system, an experiment (see fig 6) was conducted to evaluate the signal strength, the received signal strength indicator (RSSI) [24], of the anchors and to compare which side of the metal coated class windows the anchors should be placed, either indoors facing the courtyard or outdoors in the courtyard. The reason

Fig. 6: RSSI experiment

for the experiment was to see how the windows would affect the signal strength of the anchors and if the signal could go through the window so that the tag could hear it. The experiment was setup in two phases. In phase 1, 6 anchors were placed on the windows in a straight line indoors on the same side as the receiving tag to collect RSSI values of the anchors. For phase 2, the anchors were placed outdoors in the same way on the windows on the opposite side of the receiving tag. The setup can be seen in fig 7, where the red rectangles (anchors) model phase 1 (indoors) and the green rectangles (anchors) model phase 2 (outdoors).
In each phase, 75 RSSI values were collected for each anchor using serial communication (UART) by connecting the tag to the laptop via USB. The decawave_ble python library was

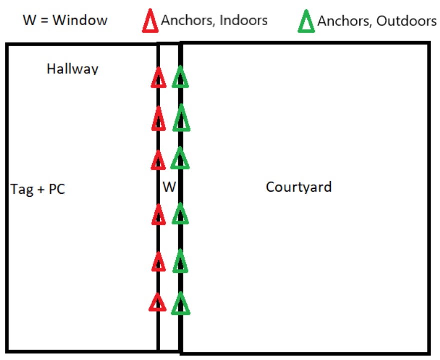


Fig. 7: RSSI collecting, indoors vs outdoors

used to get the RSSI values from the anchors [25].
When conducting the experiment of phase 2, it was detected that the tag had trouble establishing any communication with the anchors placed outdoors. When this happened, an RSSI value of -100 was set for those anchors.

### *G. Drone Flight: Tag 3D position to GPS*

Drones use the Global Positioning System (GPS) to navigate their environment, where they use a reference system, e.g., the World Geodetic System 1984 (WSG-84), to locate objects on the surface of the Earth by a coordinate made of latitude, longitude, and altitude [26]. The aim of the study was to implement a navigation system that does not use conventional GPS fixes from satellites to navigate drones. Instead, it would use a mobile DWM1001 module set as a tag, to move inside a local Cartesian coordinate system constructed by a Real-Time Location System (RTLS) made of DWM1001 modules to get a GPS fix by converting the tag's 3D position to a WSG84 GPS coordinate. To perform the conversion between local Cartesian coordinates and geographic WSG84 GPS coordinates, the Python library named geographiclib was used [27]. To use this library, a set of steps had to be performed. The steps performed consisted of addressing the flickered tag position, finding the bearing angle of the tag relative to geographic north inside the RTLS network, and getting the distance between the initiator anchor of the RTLS network and the tag. Then use geographiclib to convert tag position to GPS with the use of GPS coordinates that represent the initiator-anchor, the distance, and the bearing angle.
The sampling frequency of 10Hz was used to sample the position of the tag, whereas the tag position could be retrieved using the DWM1001 API Guide [16] by sending commands to UART and then listening for data. Retrieving the tag position with 10Hz led to a position that continuously changed and therefore needed to be filtered. To decide which method to use, variance was calculated on 10 x,y,z positions continuously to see the amount of spread they had. The result of the variance calculation led to a very small spread, no more than a 10mm difference. Due to the small spread, a simple running average was used by collecting the 5 latest tag positions to compute the average x,y,z position. Then, with every new position collected, the oldest position was removed, and a new average position was calculated. To find the bearing angle of the tag relative to the geographic north, two reference GPS coordinates were chosen that would represent the initiator anchor and the opposite anchor from the initiator with the same x coordinate in the RTLS network, or two coordinates that would be parallel to the initiator and the opposite anchor. Figure 8 shows these reference coordinates alternatives for the RTLS network that is setup in V207 at Reykjavik University.


Fig. 8: Reference GPS coordinates in V207

The geographiclib method Direct was used to get the GPS coordinate of the geographic north, by using the start GPS coordinate of either initiator or the parallel reference GPS coordinate, the $0°$ bearing and the distance between the initiator and the opposite anchor or between the 2 reference GPS coordinates. The distance between the coordinates could be calculated with the Inverse method in geographiclib [27]. With the geographic north GPS coordinate calculated, a blunt triangle could be modelled with the initiator/parallel reference GPS coordinate and the opposite anchor/other parallel reference GPS coordinate shown in fig 9. Then a bearing angle could be calculated in this blunt triangle by inserting the latitude and longitude of the geographic north and the opposite anchor/other reference GPS coordinate into a formula for finding the bearing between two GPS coordinates, shown in 1,2,3 where term A stands for opposite anchor from initiator, the term GN stands for Geographic North, $\lambda$ stand for latitude and $\phi$ stands for longitude.

$$X = \cos\lambda_A * \sin(\phi_A - \phi_{GN}) \tag{1}$$

$$Y = \cos\lambda_{GN} * \sin\lambda_A - \sin\lambda_{GN} * \cos\lambda_A * \cos(\phi_A - \phi_{GN}) \tag{2}$$

$$Bearing = atan2(X, Y) \tag{3}$$

Then, to get the bearing of the tag inside the RTLS network, the equation 4 was used.

$$TotBearing = Bearing + 90^\circ - atan2(tag_y, tag_x) \quad (4)$$

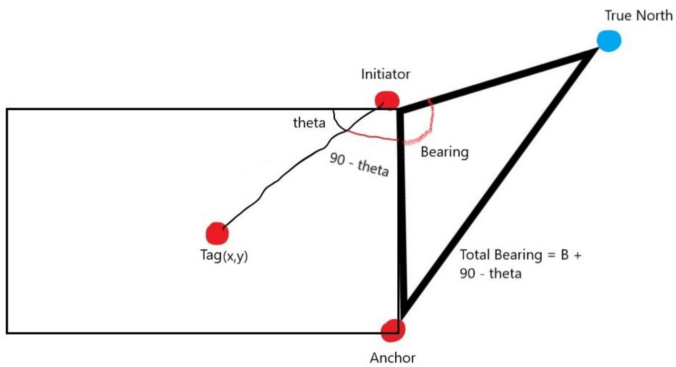

Fig. 9: Bearing

The distance between the initiator anchor and the tag was calculated with the Euclidean Distance formula for 2D [28]. To calculate the GPS coordinate of the tag, the following parameters: the GPS coordinate (lat/lon) of the initiator anchor, the total bearing, and the distance between the tag and the initiator, were used as input for the Direct method of geographiclib [27]. The lab of V207 at Reykjavik University was used to evaluate and test the GPS coordinates of the tag from the RTLS network. The tests were conducted by collecting GPS coordinates at fixed places in the lab and also by moving around in the lab. For each conducted test, the GPS coordinate was evaluated by inserting it into Google Maps to verify its correctness. The collection of GPS coordinates was done using a Raspberry Pi running the GPS calculation software and a tag connected to it via USB.

### *H. Drone Flight: Integrate GPS to autopilot & GCS*

To use the calculated GPS coordinates of the tag for the drone, it had to be integrated into the autopilot and a Ground Control Station(GCS) to control the drone. The autopilot Navio2 [29] was used, which is an autopilot hat for the Raspberry Pi that can be mounted onto the Pi, and the GCS used was Mission Planner [30]. The directions in [30] were used to install Mission Planner and [29] for Navio2. An important installation step for Navio2 was to use two telemetry connection options, one for the address of the device the GCS was installed on and one for a local host connection. The telemetry options used were an Ethernet connection between Navio2 and the device running GCS, "-A udp:192.168.XXX.XXX:14550" and the local host connection, "-C udp 127.0.0.1:14555". After they were installed, MAVLINK messages could be used to send GPS via autopilot to GCS using pymavlink [31]. For it to be possible to send a GPS coordinate, it was crucial to identify the vehicle and the component used in GCS. This could be identified by using the MavLink Inspector Tool in Mission Planner, see fig 10.

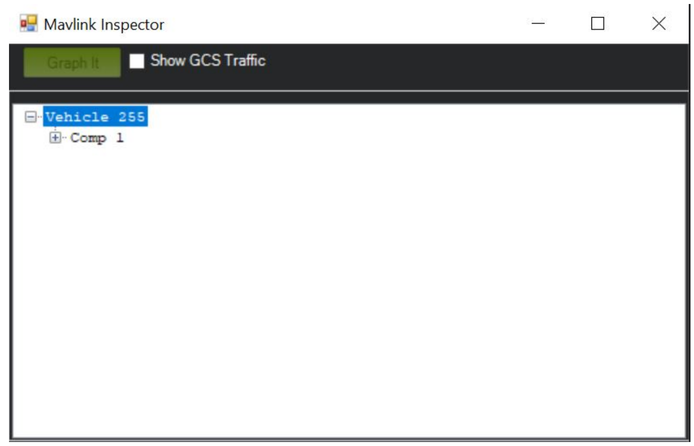

Fig. 10: Mavlink Inspector Tool

When the vehicle, source_system, and the component, source_component, had been identified, a udpin connection could be established using the pymavlink method mavlink_connection [32] seen below.

```
master = mavutil.mavlink_connection(
        'udpin:0.0.0.0:14555',
        source_system = 255,
        source_component=1 )
```

With the established connection, GPS coordinates could be sent to GCS by setting the type of the vehicle, the mav, to a value of 14 and using the pymavlink method gps_input_send() [33] seen below.

```
master.mav_type = 14
master.mav.gps_input_send(...)
```

Then the following parameters of the gps_input_send() was set.

```
fix_type = 3,# GPS fix type 3: 3D fix
lat = tag_latitude * 1e7,
lon = tag_longitude * 1e7,
```

The last step in getting the GPS into GCS was to change some parameters in the GCS Mission Planner, which was that the GPS_AUTO_CONFIG had to be set to 0 (Disabled), GPS_TYPE to 14 (MAV) and EK2_GPS_TYPE to 3 (No GPS), see fig 11.

## IV. Results and Discussion

The results achieved in this study will be presented for each conducted task that would led to answering the set research questions (RQ). Due to uncompleted experiments and time limitations, the RQ that could be answered were RQ1, followed by some notes for RQ2 and RQ3.

### *A. Installation: Courtyard Measuring*

The results from the task of measuring the courtyard could be considered unsuccessful. As mentioned in the methodology

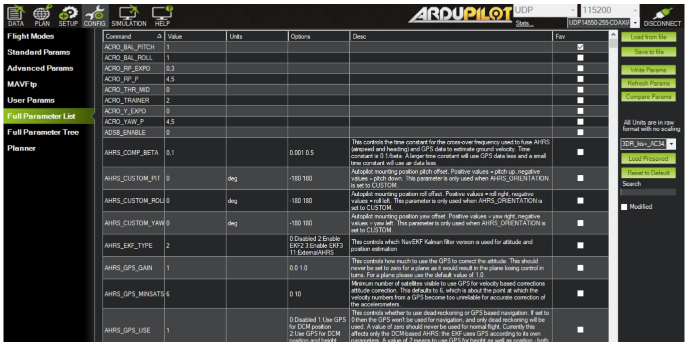


Fig. 11: Mission Planner Parameters

section for "Installation: Courtyard Measuring", Reykjavik weather climate has a big impact on the measuring tape accuracy leading to that the measuring of the courtyard is not correct and has to be redone with tape corrections [15].

The approach to use Apriltag markers to calculate the arc lengths of the long sides were not successful. Due to the camera angle from the images taken, the poses extracted from the markers created a tilted plane, where the desired one were to be flat plane with the normal $n_x, n_y, n_z = (0, 0, 1)$. To transform the tilted plane to a flat plane, affine transformations [34] was used, but due to data distortion [35] in the images providing the data, led to difficulties in solving it. The same approach were also conducted on the images taken by the drone camera [19], to test if the drone could provide with better quality images and a better angle to minimize the tilted plane. But it was also unsuccessful. The drawn conclusion of these results, is that the measuring the courtyard will be time consuming task if the desired outcome is to get an accurate measurement of the courtyard. The approach of using Apriltag markers to map the courtyard can be a promising solution if the data distortion in the images can be minimized. A way to do this is to fuse a bunch of images together with undistorted data to get a mapping of the courtyard that can be used to calculate the arc lengths of the courtyard.

Another solution that might be promising to try, is to use a chalk line [36] to construct a straight grid map in the middle of the courtyard and then from that grid measure specific points on the courtyard to get the difference in y for the anchor locations.

### *B. Installation: Test Anchors/Tag in Courtyard*

Setting the stationary detection option to enabled, it could be seen in the RTLS Manager app that the that location sometimes disappeared and then re-emerged after some time, depending on what update rate was used. But the issue was that when it reemerged, the location of the tag was sometimes way off from the correct location. The same situation happened no matter what height was used for the anchors. The same kind of test of anchor placement was conducted with the stationary detection option disabled. This led to a smooth transition of tag location inside the network, but when the anchors were placed at different heights, the tag could disappear a bit from the RTLS Manager app and reemerge but at the correct location. So the conclusion of the experiment on how to get a stable and smooth transition of the tag location is to disable the stationary detection option and place the anchors at the same height.

Some important findings were made by working with the DWM1001 and reading the system overview of the DWM1001 module [7]. The modules must be set with a name that begins with "DW" and to get the best performance from the DWM1001 RTLS, it is recommended to place the anchors at a distance of 20m apart. But from the measurements made of the courtyard walls, it can be seen that the distance is less than 20m, so using a low number of anchors in the courtyard to get better location performance should be considered as the way to setup the RTLS network in the courtyard.

### *C. Installation: Anchor RSSI outdoor vs indoor comparison*

Received signal strength indictor (RSSI) are measured in negative values, where the 0 mark is the strongest and further the RSSI value goes from the 0 mark, the weaker the signal gets. So, when an RSSI is between -60 and less than -80, it is considered as good connection. But considered as low or even unusable, when it is between -80 and -100 [37].

The result of the experiment indoors can be seen in the figures 14, 15 and the result from phase 2 can be seen in the figures 16,17, whereas the figures shows 75 RSSI values for each anchor.

In interpreting the results from the experiments, it can be seen in phase 1, that when the anchors are on the same side as the tag, an connection can be established between them. This can be clarified, as the RSSI interval is between -62 to -85 and be considered as a good connection. Even though some RSSI values are in the -80 range, it can be assumed that these values arose due to interference from pedestrians walking in the hallway and other networking devices. From phase 2, it can be seen that most of the RSSI values are -100, which where an arbitrary number that was set for the anchors when they couldn't establish a connection to the tag. So the conclusion that can be drawn about the metal coated windows and anchor placement is that the anchors should be placed on the same side of the windows where the tag is. This conclusion leads to that research question 1 can be answered, which is that the metal coated windows disables the signal when the tag and anchors are on the opposite sides of the windows. This also concludes for the navigation system that the anchors should be placed outdoors where the tag and drone will operate. The conducted experiment had to be setup as it was, due to cold weather conditions. But it models the same situations as if it would be reversed and the collection with the laptop+tag would be conducted outside.

### *D. Drone Flight: Tag 3D position to GPS*

The task for converting tag 3D position into WSG84 GPS coordinate using geographiclib [27] with the calculated total bearing and distance between initiator and tag explained in the methodology section "Drone Flight: Tag 3D position to GPS", was successful. In verify the GPS coordinates correctness, they were inserted into Google Maps in the area modelling V207 at Reykjavik University by comparing the GPS location with the location that was used to collect it. The verification's led to the result that the GPS coordinate was indeed correctly placed in V207. The figure 12 shows some of the coordinates that was collected and verified in V207.

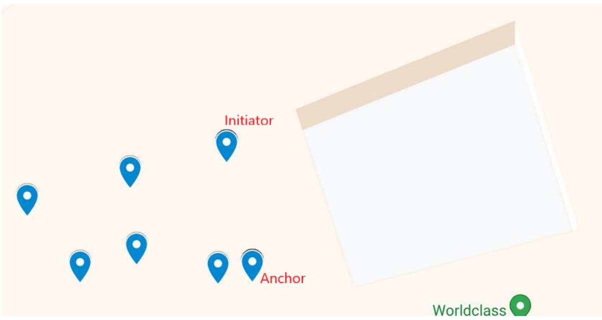


Fig. 12: GPS coordinates in Google Map

The conclusion that can be drawn, is that it is possible to convert the tag location in the RTLS network to a GPS coordinate which has led to promising result in implementing the No-GPS navigation system in the courtyard at Reykjavik University.

*E. Drone Flight: Integrate GPS to autopilot & GCS*

In integrating the tag GPS coordinate from the DWM1001 RTLS by sending the GPS coordinate via the autopilot to Mission Planner was successful. The results made it possible to test the navigation system in the RTLS of V207, seen in fig 13. The tests were conducted by moving around in V207 with the autopilot and tag connected and verifying that the GPS coordinates that were sent to Mission Planner were correct. The result of the testing led to the conclusion that it was hard to verify the correct position on the Mission Planner map. This was due to the scaling of the map and that the map showed outdoors instead of indoors, where the tests were conducted. But it could be verified that the movement of the vehicle was correct.
The conclusion is that it is possible to send GPS coordinates to the GCS, but to verify the correctness of the coordinate in the environment used is hard and it should be tested outdoors to see the correctness of GPS coordinate.

*F. Limitations*

The limitations of the system, is that it has not been tested on a real drone so the precision of the GPS coordinate can be questioned. The verification of the GPS coordinate has been done by comparing the locations by inserting it into Google Maps and Mission Planner, and verify where it should be in the map of the real environment. Verifying it in Mission Planner was very hard, so the method verifying the correctness of the GPS coordinate would need a better approach.
New reference GPS coordinate must be chosen when calculation the bearing of tag, as the GPS coordinates will change over time. Using a GPS receiver as a tool can be of help in


Fig. 13: Mission Planner GPS coordinate test

choosing new reference coordinates.
The altitude used when sending GPS coordinates(see reference [33]) to GCS via autopilot, the tag z coordinate has been used. But it should be fused with Navio2 barometer to get an accurate altitude for the drone.
Another limitation of sending the GPS coordinate is that the current system has no way to calculate the vertical(VDOP) and horizontal(HDOP) dilution of GPS position in meter. To calculate this, the following sources should be of use [38], [39] and take a look at [33] how it shall be inserted to send GPS coordinate.

*G. Research Questions*

Due to incomplete experiments, RQ2 to RQ5 couldn't be answered, but some notes on RQ2 and RQ3 could be made.
In RQ2, the question was to investigate if the general construction of the outdoor grid map/courtyard would interfere with the anchor signal strength and, if so, lead to inaccurate positioning data from the anchors. The first part of the question about the general construction of the outdoor grid map/courtyard couldn't be answered, but the other part about inaccurate positioning data could to some extent be answered. The anchors implement an RTLS system that uses Ultra Wide Band to communicate between anchors and uses the signal strength to just say that the anchors are present in the RTLS network. So the signal strength shouldn't affect the accuracy of the positioning from the anchors. Instead, the accuracy depends on the anchor placement. But this research question needs more experiments to prove it.
For RQ3, the idea was to investigate if the position estimations could become more reliable by using lower sampling frequencies for the tag and whether it is possible to use filtering algorithms to get more reliable position estimates. The

investigation into using lower frequencies can't be answered due to incomplete experiments. But it was possible to get more reliable position estimates of the tag using filtering algorithms such as the running average plus adding the RTLS network filtering method of fixed moving average [7].

## V. Conclusion

The conclusion of the study is that the measuring of the courtyard needs to be redone with tape corrections [15] to get more accurate measurements due to the fact that the weather climate of Reykjavik deviates from the conditions of the tape used [13] and the class 2 accuracy of the tape [14]. An approach to address the arc curves of the long sides of the courtyard needs to be developed, as the approach to using Aptiltag markers didn't succeed due to image data distortion. The tag option, stationary detection, should be disabled to get a smooth transition of the tag inside the RTLS network. The anchors should be placed outdoors in the courtyard on the same side where the tag will operate, as the signal strength of the anchors will not go through the metal coated windows.

Converting the tag's 3D position into a GPS coordinate is possible, but the accuracy of the GPS coordinates is dependent on choosing the correct reference GPS coordinates to calculate the total bearing angle. Sending the GPS to Mission Planner is possible, but depends on changing the settings mentioned in the methodology section "Drone Flight: Integrate GPS to autopilot & GCS".

The non-GPS navigation system still needs to be implemented in the courtyard, where the conclusion mentioned about the courtyard needs to be considered, but also the limitations mentioned in the result, such as the tag z coordinate needs to be fused with the Navio2 barometer for the altitude, the VDOP and HDOP need to be calculated and used for sending the GPS to GCS via the autopilot.

## References


[1] F. Valenti, D. Giaquinto, L. Musto, A. Zinelli, M. Bertozzi, and A. Broggi, "Enabling computer vision-based autonomous navigation for unmanned aerial vehicles in cluttered gps-denied environments," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, 2018, pp. 3886–3891.

[2] M. Samy, K. Amer, M. Shaker, and M. ElHelw, "Drone path-following in gps-denied environments using convolutional networks," 2019.

[3] C. Luo, S. I. McClean, G. Parr, L. Teacy, and R. De Nardi, "Uav position estimation and collision avoidance using the extended kalman filter," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 6, pp. 2749–2762, 2013.

[4] M. Zhou, J. Lin, S. Liang, W. Du, and L. Cheng, "A uav patrol system based on bluetooth localization," in *2017 2nd Asia-Pacific Conference on Intelligent Robot Systems (ACIRS)*, 2017, pp. 205–209.

[5] B. Yang and E. Yang, "A survey on radio frequency based precise localisation technology for uav in gps-denied environment," *Journal of Intelligent & Robotic Systems*, vol. 103, no. 3, pp. 1–30, 2021.

[6] Decawave, "Dwm1001 development board," last accessed 20 december 2021. [Online]. Available: https://www.decawave.com/product/dwm1001-development-board/

[7] ——, "Dwm1001 system overview and performance," 2018, last accessed 20 december 2021. [Online]. Available: https://www.decawave.com/wp-content/uploads/2019/05/DWM1001_System_Overview.pdf

[8] M. Strohmeier, T. Walter, J. Rothe, and S. Montenegro, "Ultra-wideband based pose estimation for small unmanned aerial vehicles," *IEEE Access*, vol. 6, pp. 57 526–57 535, 2018.

[9] T.-M. Nguyen, A. Zaini, K. Guo, and L. Xie, "An ultra-wideband-based multi-uav localization system in gps-denied environments," 10 2016.

[10] K. Guo, Z. Qiu, C. Miao, A. Zaini, C.-L. Chen, W. Meng, and L. Xie, "Ultra-wideband-based localization for quadcopter navigation," *Unmanned Systems*, vol. 04, pp. 1–12, 02 2016.

[11] J. Li, Y. Bi, K. Li, K. Wang, F. Lin, and B. M. Chen, "Accurate 3d localization for mav swarms by uwb and imu fusion," in *2018 IEEE 14th International Conference on Control and Automation (ICCA)*. IEEE, 2018, pp. 100–105.

[12] J. Tiemann and C. Wietfeld, "Scalable and precise multi-uav indoor navigation using tdoa-based uwb localization," in *2017 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2017, pp. 1–7.

[13] Kapro, "500 spring-loaded return tape measure," last accessed 20 december 2021. [Online]. Available: https://kapro.com/product/500-spring-loaded-return-tape-measure/

[14] I. Johnson, “The advantages of a class 1 tape measure,” last accessed 20 december 2021. [Online]. Available: https://www.thetapestore.co.uk/knowledgebase/Articles/The_Advantages_of_a_Class_1_Tape_Measure.html

[15] Wiki, “Tape correction (surveying),” last accessed 20 december 2021. [Online]. Available: https://en.wikipedia.org/wiki/Tape_correction_(surveying)#Correction_due_to_temperature

[16] Decawave, “Dwm1001 firmware application programming interface (api) guide,” 2019, last accessed 20 december 2021. [Online]. Available: https://www.decawave.com/wp-content/uploads/2019/01/DWM1001-API-Guide-2.2.pdf

[17] E. Olson, “AprilTag: A robust and flexible visual fiducial system,” in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, May 2011, pp. 3400–3407.

[18] Purplemath, “The distance formula,” last accessed 20 december 2021. [Online]. Available: https://www.purplemath.com/modules/distform.htm

[19] DJI, “Spark specs,” last accessed 20 december 2021. [Online]. Available: https://www.dji.com/spark

[20] OpenCV, “Camera calibration,” last accessed 20 december 2021. [Online]. Available: https://docs.opencv.org/4.x/dc/dbb/tutorial_py_calibration.html

[21] pupil labs, “apriltags,,” last accessed 20 december 2021. [Online]. Available: https://github.com/pupil-labs/apriltags

[22] A. Rosebrock, “Apriltag with python,” last accessed 20 december 2021. [Online]. Available: https://www.pyimagesearch.com/2020/11/02/apriltag-with-python/

[23] Decawave, “Mdek1001 quick start guide,” last accessed 21 december 2021. [Online]. Available: https://www.decawave.com/mdek1001/quickstart/

[24] Wikipedia, “Received signal strength indication,” last accessed 21 december 2021. [Online]. Available: https://en.wikipedia.org/wiki/Received_signal_strength_indication

[25] WildflowerSchools, “decawave_ble,” last accessed 21 december 2021. [Online]. Available: https://github.com/WildflowerSchools/decawave_ble/blob/master/decawave_ble/__init__.py

[26] Wikipedia, “World geodetic system,” last accessed 21 december 2021. [Online]. Available: https://en.wikipedia.org/wiki/World_Geodetic_System

[27] C. F. Karney, “Geographiclib api,” last accessed 21 december 2021. [Online]. Available: https://geographiclib.sourceforge.io/html/python/#

[28] Wikipedia, “Euclidean distance,” last accessed 21 december 2021. [Online]. Available: https://en.wikipedia.org/wiki/Euclidean_distance

[29] Emlid, “Installation and running,” last accessed 21 december 2021. [Online]. Available: https://docs.emlid.com/navio2/ardupilot/installation-and-running/

[30] Ardupilot, “Installing mission planner,” last accessed 21 december 2021. [Online]. Available: https://ardupilot.org/planner/docs/mission-planner-installation.html

[31] ArduSub, “Pymavlink,” last accessed 21 december 2021. [Online]. Available: https://www.ardusub.com/developers/pymavlink.html

[32] ArduPilot, “Pymavlink mavutil.py,” last accessed 21 december 2021. [Online]. Available: https://github.com/ArduPilot/pymavlink/blob/master/mavutil.py

[33] ArduSub, “Send gps position,” last accessed 21 december 2021. [Online]. Available: https://www.ardusub.com/developers/pymavlink.html#send-gps-position

[34] Wikipedia, “Affine transformation,” last accessed 21 december 2021. [Online]. Available: https://en.wikipedia.org/wiki/Affine_transformation

[35] ——, “Distortion,” last accessed 20 december 2021. [Online]. Available: https://en.wikipedia.org/wiki/Distortion_(optics)

[36] ——, “Chalkline,” last accessed 20 december 2021. [Online]. Available: https://en.wikipedia.org/wiki/Chalk_line

[37] metageek, “Understanding rssi,” last accessed 21 december 2021. [Online]. Available: https://www.metageek.com/training/resources/understanding-rssi/

[38] J. Chaffee and J. Abel, “Gdop and the cramer-rao bound,” in *Proceedings of 1994 IEEE Position, Location and Navigation Symposium - PLANS’94*, 1994, pp. 663–668.

[39] R. B. Langley *et al.*, “Dilution of precision,” *GPS world*, vol. 10, no. 5, pp. 52–59, 1999.

## VI. FIGURES

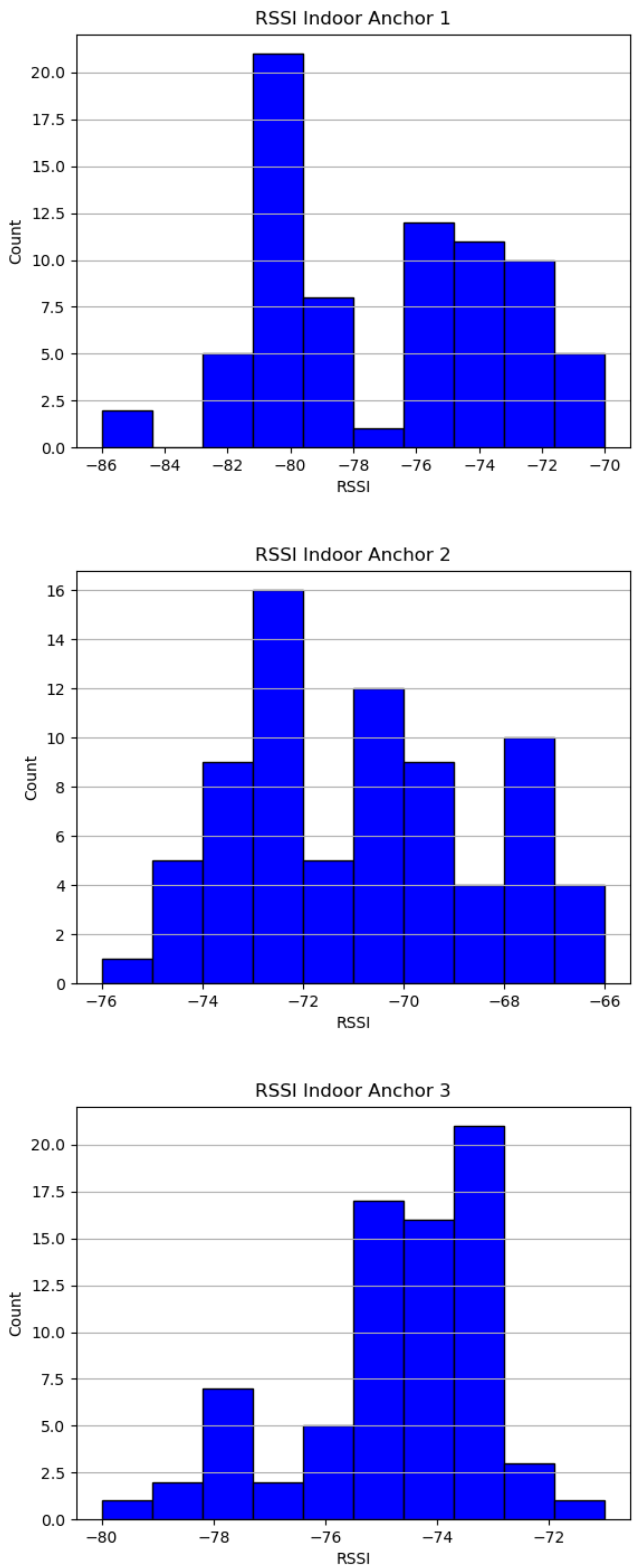


Fig. 14: RSSI Anchors Indoor

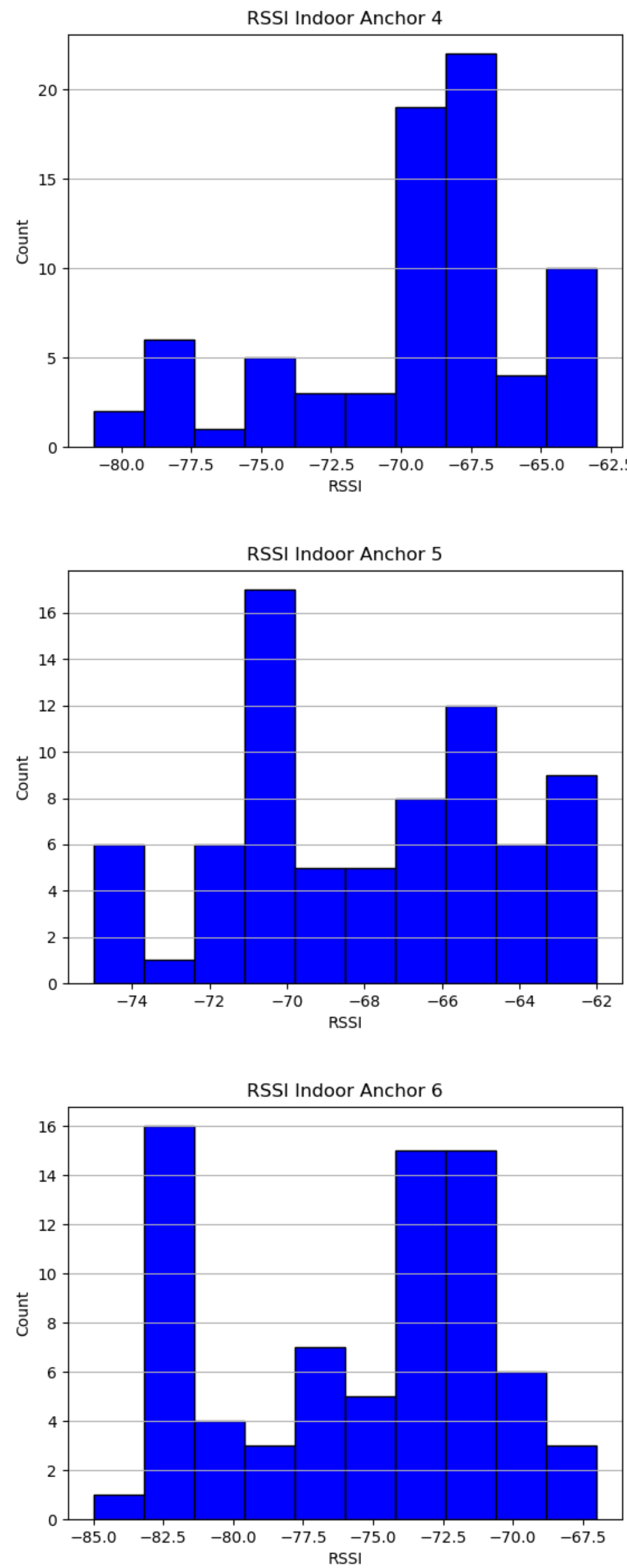


Fig. 15: RSSI Anchors Indoor

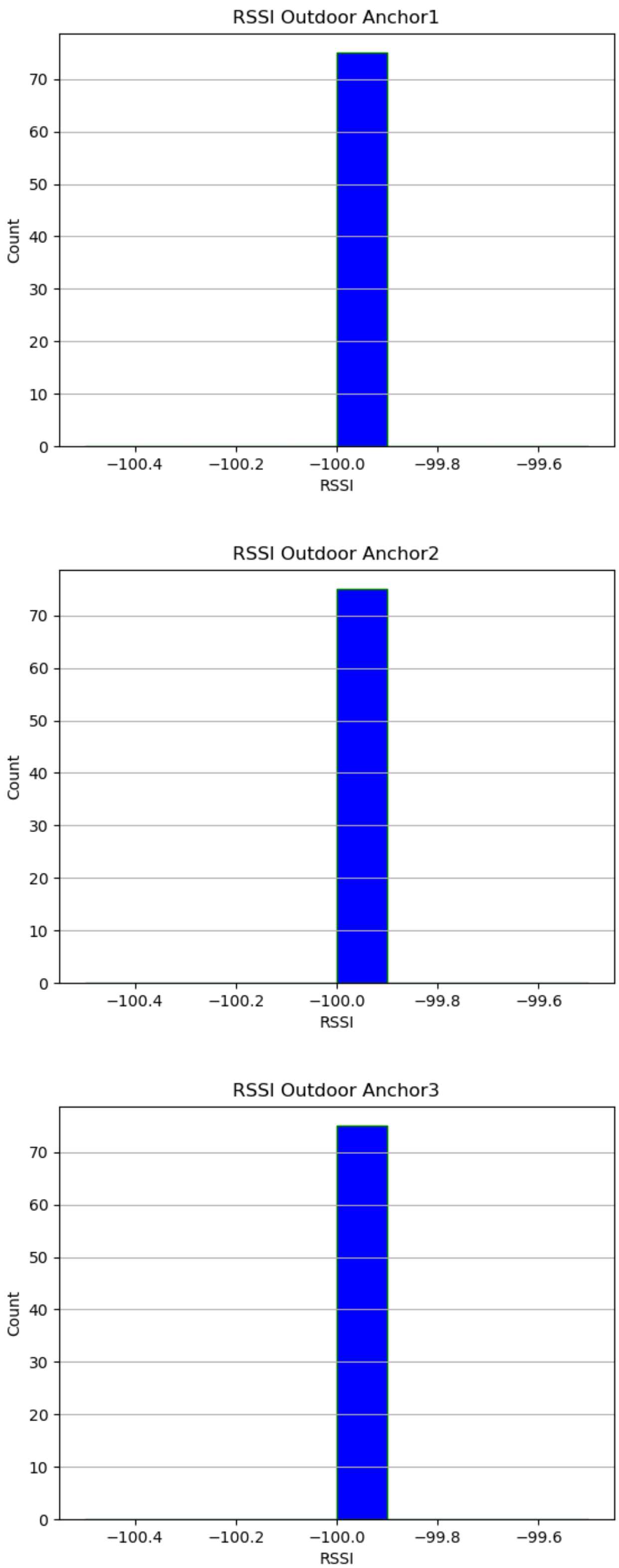


Fig. 16: RSSI Anchors Outdoors

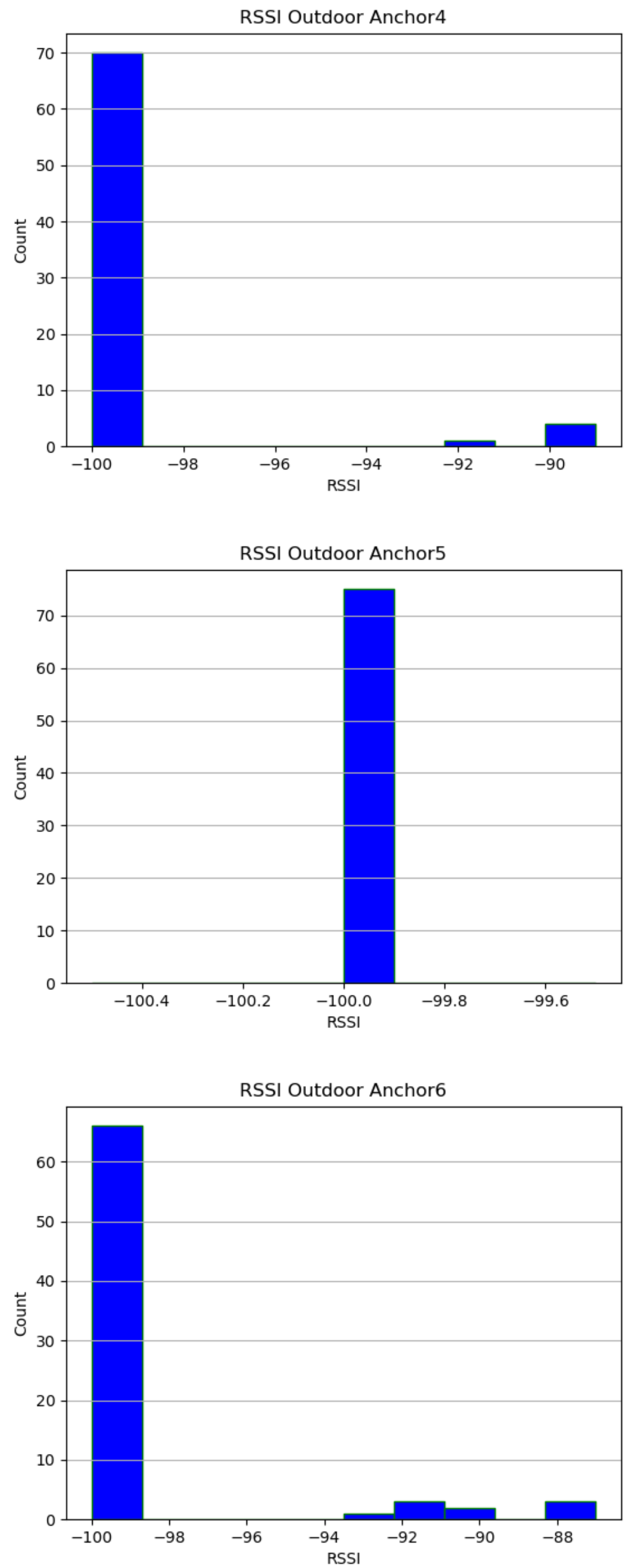


Fig. 17: RSSI Anchors Outdoors