İstanbul Bilgi University

UWB INDOOR LOCALIZATION

by

117202035 Hakan KARADENİZ
117202021 Akın Bora KILIÇ

Supervised by

Asst. Prof. Okan Zafer BATUR

A report submitted for EEEN 492 senior design project class in partial fulfillment of the requirements for the degree of Bachelor of Science
(Department of Electrical and Electronics Engineering) in
İstanbul Bilgi University

June 30th, 2022

ACKNOWLEDGMENTS

We thank Okan Zafer Batur for his technical advice and guidance during the project. We would also like to thank our college for financially supporting the development of this project.

Akın Bora Kılıç:

I want to thank my supervisor. I would also like to thank my parents and my girlfriend for the invaluable support and motivation they gave me during the undergraduate program and this thesis.

Hakan Karadeniz:

I want to thank my parents for their endless encouragement and enthusiasm, and my sister for her infectious positivity. Thanks to my colleague Akın, who stood by us throughout this process.

Istanbul, June, 2022

ABSTRACT

The increasing number of applications in location and tracking systems, user tracking and real-time location information acquisition systems are generally suitable for outdoor use. Thanks to high-bandwidth uwb systems, indoor location detection and tracking is less error-prone and provides secure data retrieval without interference between smart devices operating indoors. The work to be done in this thesis is a system that determines and locates the indoor location of the user with 1 transmitter and more than 2 receivers based on certain code algorithms to be made on UWB technology.

List of Contents

LIST OF FIGURES

LIST OF TABLES

LIST OF APPENDICES

Appendix

1. INTRODUCTION

1.1 Background and Literature Survey

With the development of technology, various workplaces began to reduce manpower and mechanize, and this situation created a new job opportunity for us engineers. The task of engineers is to produce solutions that will make human life easier in line with emerging needs. This localization project is designed for robots that will provide for online shopping and market needs that will increase in the future. The project, which is called UWB Indoor Localization, generally appeals to supermarkets. The customer profile that wants to buy this product is generally companies that want to reduce manpower and increase productivity in their markets and reduce tasks that make human life easier with various machines.vehicles designed for this purpose will be programmed to transfer various products to desired places in supermarkets.

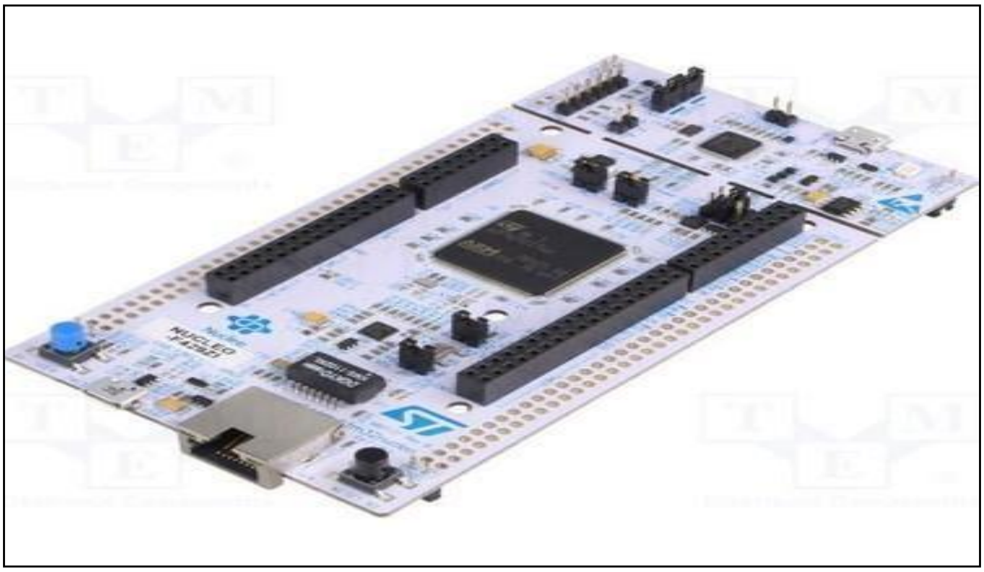


Figure 1. Nucleo-F429ZI

While creating the project, software will be made on the STM32 NucleoF429ZI (**Figure 1**) development board. However, our most important part will work with our DWM3000EVB board (**Figure 2**) and this thing is the chip on the DWM3000EVB that will do all the calculations. The DWM3000 UWB module is x50 faster than GPS technology, and with greater accuracy this provides real-time positioning of objects with an accuracy of up to 10 cm.

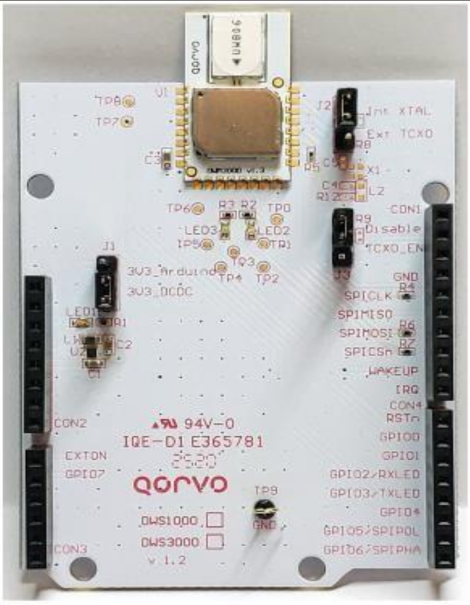


Figure 2. DWM3000evb(in future DWS3000)

The module supports data rates of 850 kilobits per seconds and 6.8 Mbps per seconds. In addition to these, it has worldwide Ultra Wide Band support with channel 5 **(6.5 GHz)** and channel 9 **(8GHz).**

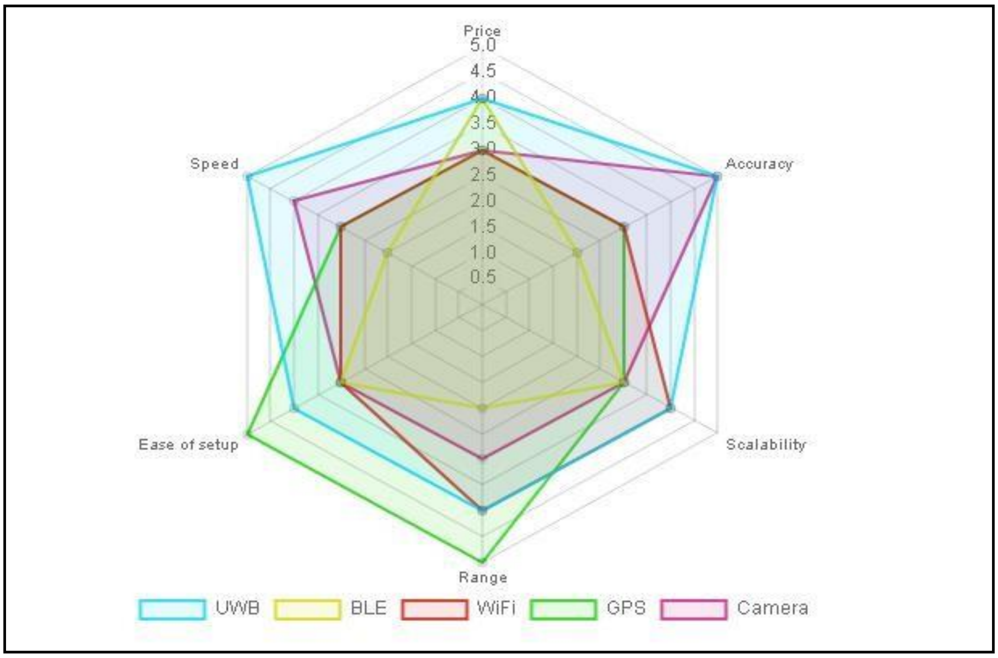


Figure 3. Comparison of other technologies with UWB

Ultra wideband, using **IEEE 802.15.4** standards, consists of many frequency channels in the 3-6 GHz range with a minimum width of 500 MHz. In contrast to narrowband RF technologies such as WiFi and Bluetooth, which signal at 2.4 GHz, distributing signals over a wide frequency range means using a narrow pulse in the time domain.

Due to its wide frequency range, UWB has incredible features and capabilities compared to other technologies **(Figure 3)**:

- UWB is not susceptible to narrowband interference.
- UWB can provide very precise range.
- UWB can provide a secure communication channel.

Huge growth is expected after UWB integration in mobile phones. On September 10, 2019, Apple announced the first use of UWB.

Resource	Web Adress
<u>**Search Enginees**</u>	
Google	**http://www.google.com**
Google Scholar	**http://www.scholar.google.com**
<u>**Electronic Search of Database**</u>	
Product Brief	**https://www.decawave.com**
DWS1000 & DWS3000 Schematics	**https://www.decawave.com**
DWS3000 API Software and API Guide	**https://www.decawave.com**
DWM3000EVB Quick Start Guide	**https://www.decawave.com**
UM 1974 User Manual (Stm32)	**https://www.st.com**
STM32CubeIDE	**https://www.st.com**
Ac6 Tools	**https://www.ac6-tools.com**
OpenOCD Debugger	**https://openocd.org/**
Forum	**https://community.st.com/**
Forum	**https://decaforum.decawave.com/**
Forum	**https://github.com/**

Table 1. Web Based Method of Literature Search

1.2 Objectives

The final project being worked on is aimed at determining the indoor location using ultra-broadband technology. Although there are many methods for location detection, the reason for using ultra-wideband is high bandwidth and low energy consumption compared to other methods. One of the advantages of high bandwidth is the ability to measure very precisely. Thus, it is aimed to minimize the error rate in the measurements made. As a first step, the communication time between the two modules and the elapsed time are shown as the distance between the two modules, which will ensure communication between the ultra-wideband devices used, one of the modules is kept fixed and functions as a transmitter, the other mobile module is used as a receiver, and a communication line is established between them.

After the stable operation of the first step, it is aimed to find the location of the user more clearly by increasing the number of modules, and it is possible to detect the location of more than one user with the increased modules.

1.3 Realistic Design Constraints:

Significant hardware adjustments are required to make the cards work together. The first of these is the soldering process in **Figure 4**.

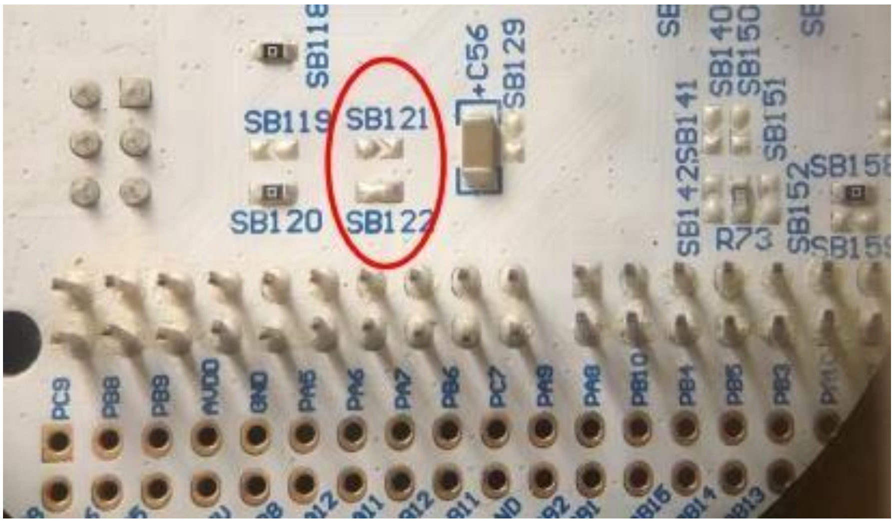


Figure 4. SB121/SB122 modification of the NUCLEO-F429ZI develoment board

On the NUCLEO-F429ZI development board, the MOSI pin of SPI1 must be modified from the default PA7 pin to the PB5 pin in order to use both Ethernet and SPI1. This is accomplished by shorting the SB122 solder bridge on the bottom of the development board and opening the SB121 solder bridge, as shown below.

The second hardware modification of the NUCLEO-F429ZI is an optional work in **Figure 5**. By attaching a wire between the "+" side of C54 and the "E5V" pin of CN11, the 5V from the "User USB" port may be linked to the "E5V" rail of the NUCLEO -F429ZI and used to power the development board.

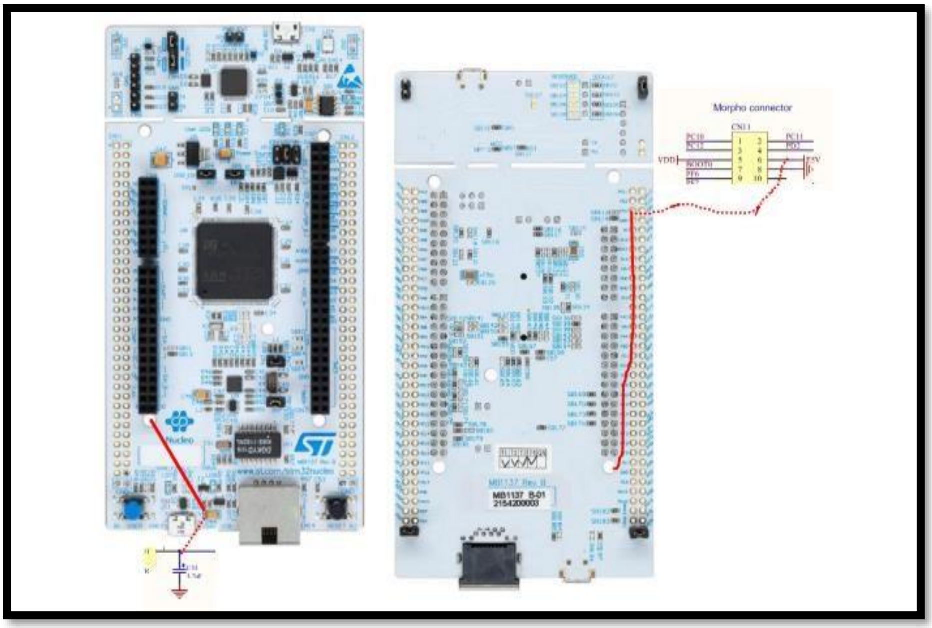


Figure 5. User USB E5V modification of the NUCLEO-F429ZI

1.4 Approach and Methodology

Today, people use many products that they call smart devices in their daily lives. Due to the prevalence of smart and business systems recently, it was decided to make product developments in this area. If we want to make an environment smarter with the technological tools used, it must be able to communicate and work in harmony with each other so that the system can be more efficient and smart.

First of all, high-bandwidth technologies should be used so that the devices are compatible with each other and in a wireless connection. Therefore, it was decided to continue the project with ultra-wideband technology. Among the UWB modules DWM1000, DWM1001c, and DWM3000, the DWM3000 module was chosen because it complies with various IEEE standards (IEEE 802.15.4-2015 and

802.15.4z2020(BPRF mode)) UWB channels support and has an easy interface. After selecting the module to be used, it remains to choose the card required to run it. These are nRF52832 and Nucleo-F429ZI are cards. The reason why we chose the Nucleo card was that ST products are generally used in the country, it was asked to work on it and experience it. After the selections were made in this way, many exemplary projects were made for a quick start and recognition of the card. UWB signals occur between the receiver and transmitter. It is planned to increase the number of buyers and make real-time and more consistent measurements. At this point, TDoA and 6GHZ-8GHz ranges give us accurate results, as the convenience provided by the UWB3000 module we have chosen, the frequency range in which it operates, and supports various location schemes.

1.5 Two-Way Ranging Process

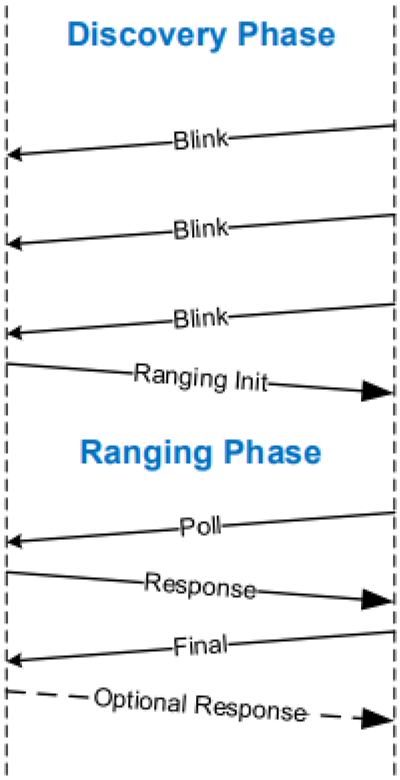


Figure 6. Discovery Phase

In order to connect 2-way DWM cards, it is necessary to put the unpaired anchor into listener mode first. In this state, the tag flashes and sends a message. It continues to flash periodically until it is paired, waiting for a response. The ranging init is sent after the other Anchor decides to match the unmatched tag. The ranging response seen by the tag is validated by the anchor. The distance is calculated by the anchor and a message is sent to the tag.

1.5.1 Discovery Part

Blink message is sent by the tag that is in discovery mode. The initialization response is expected by the anchor. If the message expected by the tag does not come, it goes into sleep mode. If the start message is received, the discovery phase is finished and the range is switched to the range phase.

1.5.2 Ranging Part

In the range stage, 2-way message sending and receiving is performed by the anchor and tag. These allow sending and receiving the poll message. Tag uses TOF to calculate distance

2. IMPLEMENTATION

During the execution of this project, many studies were carried out in the field of software. New "STM32CubeIde" learned. STM32 board has been researched in detail and studies have been done on it. In addition, example codes related to the DWM3000evb chip were examined.

The studies were carried out with the most widely used "C" language, one of the developer languages. In addition, work has been advanced with the "STM32CubeIde" based on "Eclipse".

The sample studies are as follows ;

2.1 Pin Project

This project is one of the first projects in the works. In "**Figure 6**" you can see a small part of the code written to main.c. Here, the previously identified pins are checked. Board LEDs were made functional using the library called HAL. Then, the selected green led flashed when data was sent with the button. In response to this sent data, the text "Hello World" was encountered. However, in every millisecond that we did not send data with the button, we saw the text "BYE". You can see the data in "**Figure 7**".

```c
while (1)
{
  /* USER CODE END WHILE */

  /* USER CODE BEGIN 3 */
    if(HAL_GPIO_ReadPin(BUTTON_GPIO_Port, BUTTON_Pin) && is_pressed){

        is_pressed=0;
        HAL_GPIO_WritePin(LD1_GPIO_Port, LD1_Pin, 1);
        printf("HELLO WORLD");

    } else if(HAL_GPIO_ReadPin(BUTTON_GPIO_Port, BUTTON_Pin) == 0 ){


        is_pressed=1;
        HAL_GPIO_WritePin(LD1_GPIO_Port, LD1_Pin, 0);
        printf("BYE BYE");


    }
}
/* USER CODE END 3 */
}
```

Figure 7. Small Part of Code in main.c

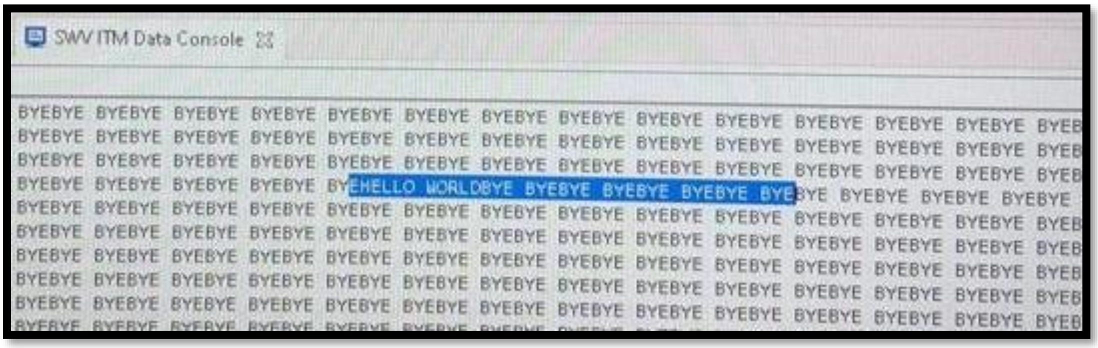


Figure 8. SWW ITM Data Console of Pin Project

2.2 Uart Polling (Usart Polling)

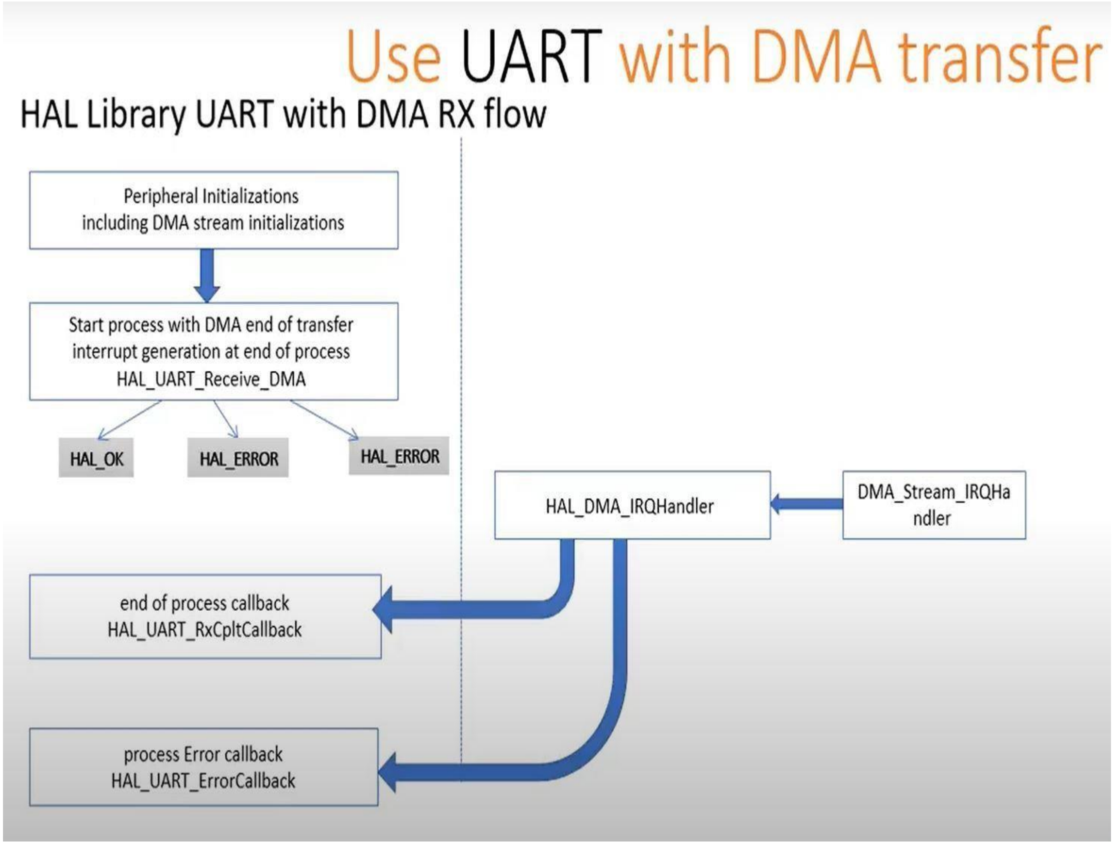


Table 2. Uart Data Transfer Chart

CubeMX generates some code after setting it up.

1- How its work in software uart and dma are initialized

2- Uart receive the buffer receiving the data from an external source

3- When the dma buffer is full you get an interrupt

4- Code process the interrupt

5- Buffer receive the call back

6- If the buffer receives the error callback the code is refactored

Uart communication method is used to communicate between receiver and transmitter. The sending device acts as a transmitter by serializing the parallel data and transferring it to the receiver. Data received in serial form is converted back to parallel data before being read. The pin that transmits data is called TX, and the pine that receives data is called RX.

The desired data is specified in **Figure 8**. A small part of the code written in this section is visible. Bit settings of the sent data have been made.

```c
uint8_t c=0x32;
char *buffer= "HAKAN AKIN\n\r";
uint8_t pData[5];

/* USER CODE END PV */

/* Private function prototypes ---------
void SystemClock_Config(void);
static void MX_GPIO_Init(void);
static void MX_USART3_UART_Init(void);
```

Figure 9. Settings of polling

Later, these bit settings were used to get correct information in the additionally downloaded "Tera Term" application. Tera Term application provides serial access to COMs on computers. It is connected serially to the port where the STM32 card is located and the desired data is seen. (**Figure 9**)

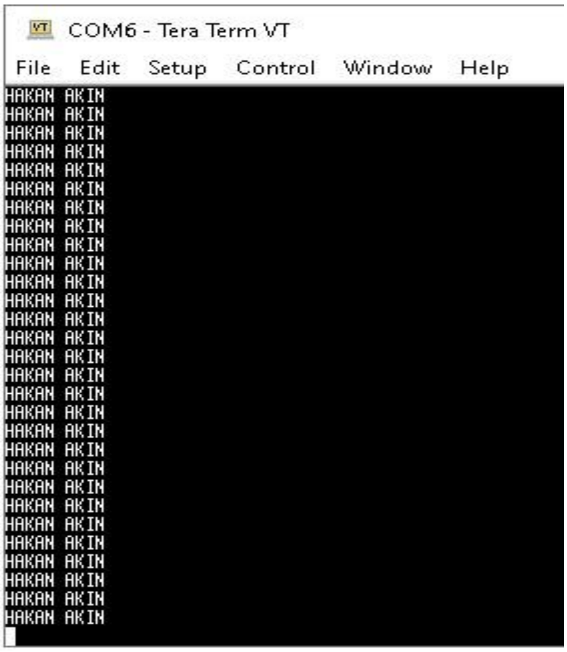


Figure 10. Tera-Term

2.3 Controller Area Network

In this project, two cards are required to communicate with each other. This is intended to be done with the buttons on the cards. When the button on the card is pressed, the leds on the other card will start to light. Actually, the coding part has been completed.
As seen in "**Figure 10**", special communication pins are defined. Once this is done, the HAL library kicks in and creates a file.
This file is the part where the settings related to the communication we will establish. You can see the library created in "**Figure 11**". Also, this library has many settings that we know nothing about. All necessary parts of the file were researched and used.

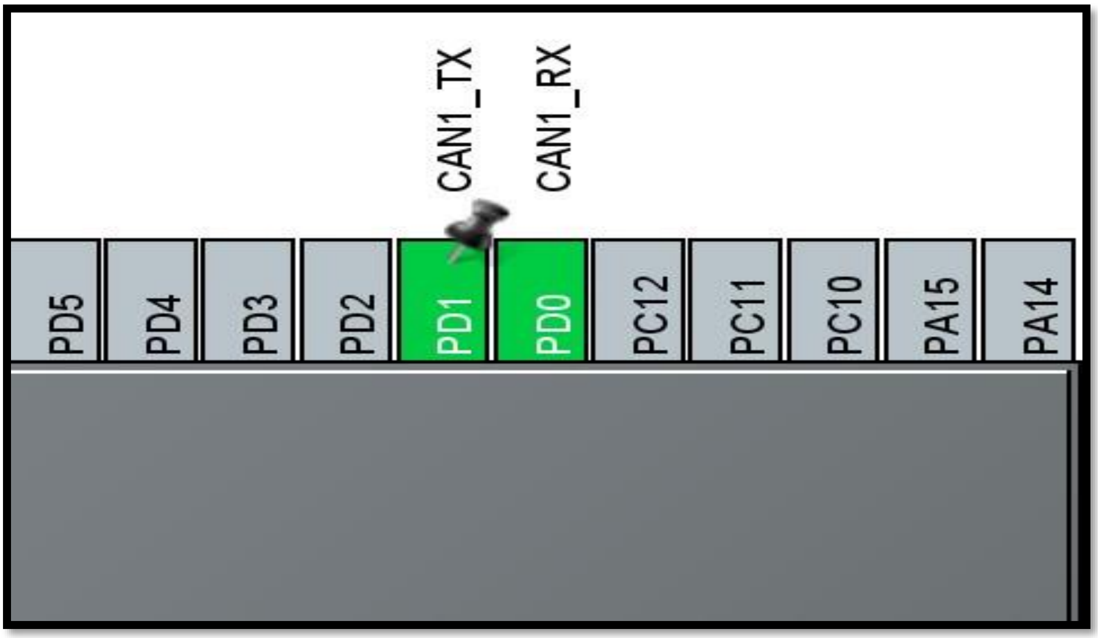


Figure 11. CAN pins defined in CubeMX

Figure 12. Outlines of stm32f4xx_hal_can.h All of the file's essential components have been investigated and utilised.

The controller area network's codes have been specified.(**Figure 12**)

- The CAN example we're using is hcan1.
- TxHeader is the message's header.
- The Data field is TxData.
- TxMailbox is the mailbox where the header and data messages will be sent.

STM microcontrollers, like most recent ones, include two hardware high-performance CAN controllers that require simply transmitters and software to interact over a bus.

```
/* USER CODE BEGIN PV */

CAN_HandleTypeDef hcan1;
CAN_TxHeaderTypeDef pHeader;
CAN_RxHeaderTypeDef pRxHeader;
CAN_FilterTypeDef sFilterConfig;
uint32_t TxMailbox;
uint8_t a,r;

/* USER CODE END PV */
```

Figure 13. Codes of data

3. LOCALIZATION

3.1 Procedure

3.1.1 System

In this section, it is the preliminary information of the interior location determination studies, which are the graduation project. Detailed information of the system is explained in the subsections.

3.1.2 Executive Summary

The aim of this thesis is on indoor positioning with Decawave products. Various applications have been used for location determination. Location information is obtained from applications, mapping and efficiency is provided by code sweaters.

3.1.3 Results

The received data has been processed on the accuracy of the DWM 3000 EVB. The measurements of the system designed as a result of error evaluations are at a level that can be used and improved in the laboratory.

- **PDoA.exe**
  Location data cannot be obtained without using this application. Data has to be collected using Windows.
- **Antenna**
  DWM 3000 EVB cards act as antenna. While the fixed ones serve to limit the determined location, the movable ones are calculated by distance calculations with the fixed ones and the location is processed into the map with various algorithms.
- **Antenna Locations**
  Antennas that are fixed in line with the algorithm should be in the designated places. It should work with different algorithms as it may give erroneous results in shape changes.

The UWB card, defined as a node, is connected to the application computer that provides the collection of distance data. In addition, the system, which has 2 anchors, is positioned across the room according to the size of the room. Thanks to the anchors placed in 2 corners and with the help of the designed algorithm, the image of the room is created virtually. Anchors in the corners meet the energy needs thanks to the power banks in this system. Data is collected in the region created by the antenna that acts as a node connected to the computer.

Figure 14. Located Cards

Calculations were made by considering the shape of the room as square in line with the algorithm created.

Standalone methods:

1) Counting every step taken as 1 meter and entering the lengths of the created virtual room accordingly.

2) Spacing using a tape measure.

3) Calibrating the UWB module used and getting the distance from the computer as data.

As a result of these methods used, verifying the calibration with the meter and getting the data from the computer gave the most accurate result.

After calibration, one-dimensional consistency and deviation tests were performed up to 2 meters at 5 cm intervals.

Afterwards, measurements in 2 dimensions were made at the determined locations and data were obtained.

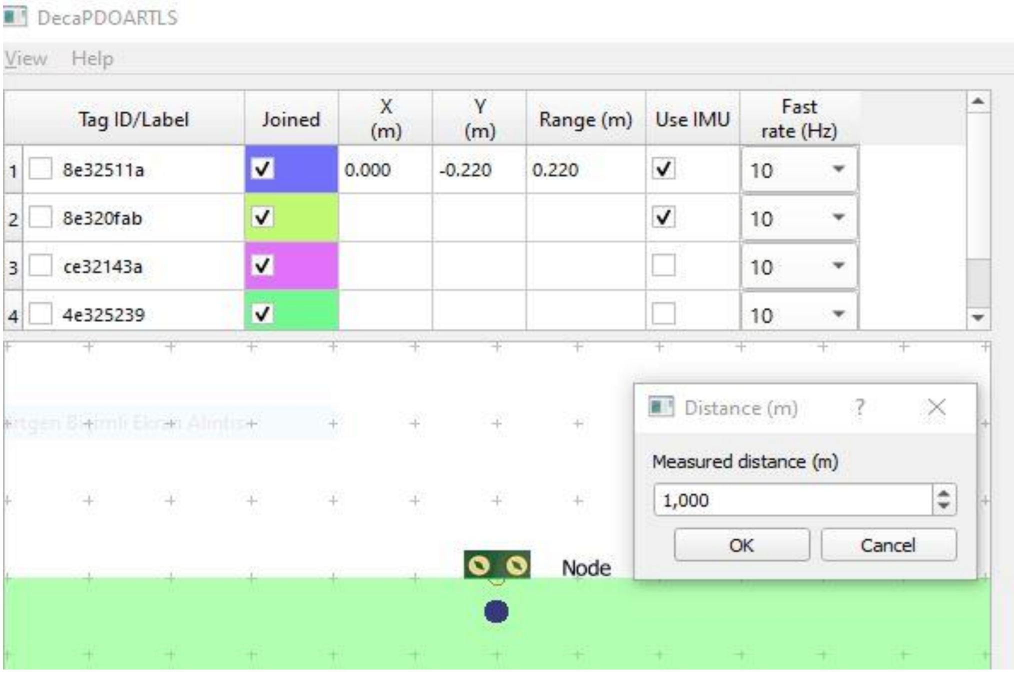


Figure 15. PDoA.exe

The node defined antenna is connected to the computer and the application opens. After the necessary calibrations are made, the anchors are placed face to face at the specified distances. The node communicates with the anchors and continuously receives and records distance data separately.

With the designed algorithm, the created records produce a visual that shows the path that the user has followed in a 2-dimensional plane.

The distance data generated during the movement were missing in the tests carried out during the design phase of the algorithm. However, various important additions ensured the complete output of the received data.

```
#Calculation of X coordinate / Change your j value for calculations
while True:
    if not j == 1000:
        a = float (deviceList2[j])*float (deviceList2[j])
        Ysqr = float(calculationListY[j])* float(calculationListY[j])
        valueOfX = math.sqrt(abs(a-Ysqr))
        calculationListX.append(valueOfX)
        j = j+1
    else:
        print("calculationListX")
        print(calculationListX)
        break

while True:
    if not k==1000:
      coordinateY = (float (c))- (calculationListY[k])
      calculationListD.append(coordinateY)
      k = k + 1
    else:
      print("calculationList D ")
      print(calculationListD)
      break
```

Figure 16. Calculation of X Coordinate

In the virtual space created with the designed algorithm, the data is collected by the computer and the recorded data is mapped using Python and its efficiency is calculated with distance measurements made in one dimension.

Thanks to the application used, the ID and distance information for each anchor is instantly extracted. This file, which is created sequentially, is accessed as a result of file opening operations using Python. The distance information listed instantly is listed separately within the algorithm. These lists are used to perform the operations of the designed algorithm in each iteration. Different distance information taken in a single plane, thanks to the designed algorithm, extracts the movements made in the 2D plane.

Apart from this, MAPE percentage ranges were taken as reference, and error rates were also written in Python and the results were obtained in graphics.

```
file = open("deneme.txt","r")

#Device Lists are here !
deviceList1 = []
deviceList2 = []

#Calculation Lists are here !
calculationListX = []
calculationListY = []
calculationListD = []

#Coordinate List are here !

#Reading data file line by line .
file.readline()
```

Figure 17. Reading file

The code throws a problem as long as the list size is entered correctly. The measurement and margin of error results are examined in more detail in the subsections.

3.2 Method Used

Position calculation is a mathematical phenomenon. During the development of the project, many mathematical calculations were used. The distances between "ANCHOR" and "TAG" are taken from files with the extension ".log" created by PDoA.exe. These distances are written to the file instantly. However, the program does not allow real-time use of this incoming data.

Since this was the case, it was decided to create a coordinate system with the calculations made after the incoming data. The coordinate system was designed to work actively on any square plane to be created. This algorithm was created by the students who have the project.

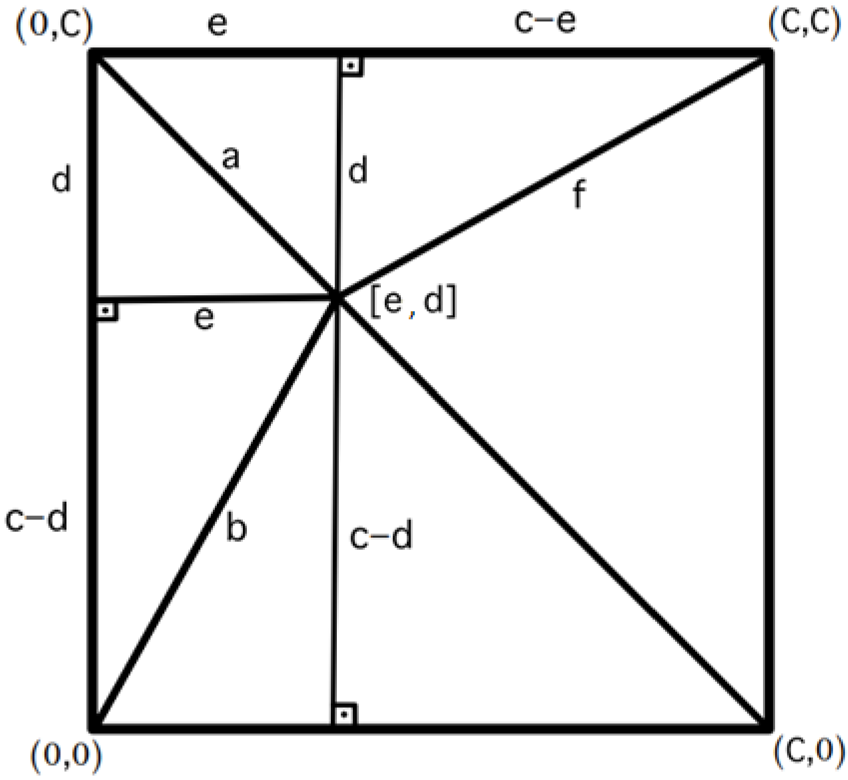


Figure 18. Coordinate System for Calculations

When viewed according to Figure X, there is one "Anchor" at (0,0) and (0,C) points. While making these calculations, the information from PDoA.exe and the side length of the square area measured should be known. There is known data when creating the project.
For example, these can be listed as follows:

- The value "a" is known.
- The value "b" is known.
- The value "c" is known (received from the user).

Mathematical calculations for the coordinate system:
Calculating y-axis

$$e^2 + d^2 = a^2$$
$$e^2 + (c - d)^2 = b^2$$

$$- e^2 - d^2 = - a^2$$
$$e^2 + c^2 - 2cd + d^2 = b^2$$

$$c^2 - 2cd = b^2 - a^2$$
$$d = b^2 - a^2 - c^2 / - 2c$$

Calculating x-axis

$$e^2 + d^2 = b^2$$
$$e^2 = b^2 - d^2$$
$$e = \sqrt{} b^2 - d^2$$

These calculations are combined with the code and designed as an algorithm. In this design process, "PYTHON" was used as the software language. We show the use of the values "a" and "b" as an example.

The shape of "X" and "Y" value specified on the code is given in lines :

```
print(" ")
print("CALCULATIONS")

#Calculation of Y coordinate / Change your i value for calculation
while True:
    if not i== 1000:
        asqr = float(deviceList1[i])* float(deviceList1[i])
        bsqr = float(deviceList2[i])* float(deviceList2[i])
        csqr = (float (c) * float (c))
        ust1 = float(asqr) - float (bsqr)
        ust2 = ust1 - float(csqr)
        alt = float (c) * -2
        valueOfY = ust2 / alt
      #  valueOfY = (float(asqr) - float(bsqr) - float (csqr)) / (
-2*float (c))
        calculationListY.append(round(valueOfY,4))
        i = i+1
    else:
        print("calculationListY")
        print(calculationListY)
        break
```

Figure 19. Codding Part I

```
#Calculation of X coordinate / Change your j value for calculation
while True:
    if not j == 1000:
        a = float (deviceList2[j])*float (deviceList2[j])
        Ysqr = float(calculationListY[j])* float(calculationListY[
j])
        valueOfX = math.sqrt(abs(a-Ysqr))
        calculationListX.append(valueOfX)
        j = j+1
    else:
        print("calculationListX")
        print(calculationListX)
        break
```

Figure 20. Codding Part II

```
while True:
    if not k==1000:
      coordinateY = (float (c))- (calculationListY[k])
      calculationListD.append(coordinateY)
      k = k + 1
    else:
      print("calculationList D ")
      print(calculationListD)
      break
```

Figure 21. Codding Part III

As a result, all these calculations serve to show the position visited within a certain area on the coordinate system. As a result of the calculations, the position of the device is determined at point (e, d) as shown in the figure.

3.3 Explanation of the Code

In this code, we have specified the necessary places. All of the code used in this section will be shared and the necessary parts will be explained respectively.

The code ,

```
from cmath import sqrt
import math
from matplotlib import pyplot as plt

print("Please Enter Your Actual Distance between the TAGs for Calc
ulation (meter)")
c = input()
i = 0 # i value will use in the While Loops
j = 0 # j value will use in the While Loops
k = 0 # k value will use in the While Loops
l = 0 # l value will use in the While Loops

file = open("deneme.txt","r")

#Device Lists are here !
deviceList1 = []
deviceList2 = []

#Calculation Lists are here !
calculationListX = []
calculationListY = []
calculationListD = []
Coordinate List are here !
```

Figure 22. Codding Part IV

In the first chapter, Python libraries and lists used to collect data are given. In addition, the "C" value received from the user, which is one of the most important parts of the calculation, is determined in this section. The value received is so important that if the user does not enter a correct value, there may be large deviations in the resulting graph.

```
#Reading data file line by line .
file.readline()

while True:
    str = file.readline()
    arr = str.split(":")    #Split the data like an array .
    if not str:
        break

    if arr[2] == "RR2": #If we have RR2 in the line, Use this line
!
        if arr[3] == "ce32143a":
            deviceList1.append(arr[5])
            deviceList1[len(deviceList1)-
1] = deviceList1[len(deviceList1)-
1][0:len(deviceList1[len(deviceList1)-1])-1]
        if arr[3] == "8e320fab":
            deviceList2.append(arr[5])
            deviceList2[len(deviceList2)-
1] = deviceList2[len(deviceList2)-
1][0:len(deviceList2[len(deviceList2)-1])-1]
        #We can add and drop any device if we know the device ID !

#The Device Lists can be seen like this .
print(deviceList1[0:10])
print(deviceList2[0:10])
```

Figure 23. Codding Part V

In this part of the code, it is seen that the desired data is selected from the 'text' file from which the data comes. First, the file is read. The names of the devices used are indicated. To mention this example, "ce32143a" device represents (0,0) point as position. Distance data from this device is added to the list. The split function is used for seperating the data. When the data seperate, the code can access the distance value here. The print functions are created for looking to top ten data.

```
print(" ")
print("CALCULATIONS")

#Calculation of Y coordinate / Change your i value for calculation
s
while True:
    if not i== 1000:
        asqr = float(deviceList1[i])* float(deviceList1[i])
        bsqr = float(deviceList2[i])* float(deviceList2[i])
        csqr = (float (c) * float (c))
        ust1 = float(asqr) - float (bsqr)
        ust2 = ust1 - float(csqr)
        alt = float (c) * -2
        valueOfY = ust2 / alt
      #  valueOfY = (float(asqr) - float(bsqr) - float (csqr)) / (
-2*float (c))
        calculationListY.append(round(valueOfY,4))
        i = i+1
    else:
        print("calculationListY")
        print(calculationListY)
        break

#Calculation of X coordinate / Change your j value for calculation
s
while True:
    if not j == 1000:
        a = float (deviceList2[j])*float (deviceList2[j])
        Ysqr = float(calculationListY[j])* float(calculationListY[
j])
        valueOfX = math.sqrt(abs(a-Ysqr))
        calculationListX.append(valueOfX)
        j = j+1
    else:
        print("calculationListX")
        print(calculationListX)
        break

while True:
    if not k==1000:
      coordinateY = (float (c))- (calculationListY[k])
      calculationListD.append(coordinateY)
      k = k + 1
    else:
      print("calculationList D ")
      print(calculationListD)
      break
```

Figure 24. Codding Part VI

In the calculation part of code is given in Method Used part.

```
#Calculation of coordinates
%matplotlib inline

plt.plot((calculationListX),(calculationListD), 'ro')

plt.xlim(-5,10)
plt.ylim(-5,10)

plt.xlabel("X")
plt.ylabel("Y")

plt.show()
```

Figure 25. Codding Part VII

The last part of the code is where the incoming data is graphed. Using the created lists, points on the coordinate system were determined.

Finally, an example is shared for the mentioned log files. This example is a "text" file obtained using PDoA.exe.

```
T:155551316:PDOARTLSGUI:LogFile:6.0.220217:version 3.7
T:155553993:RR2:ce32143a:1:6.56
T:155553993:LOC:ce32143a:1:0:-6.56
T:155553993:AA2:ce32143a:1:0:0
T:155554123:RR2:ce32143a:2:6.6
T:155554123:LOC:ce32143a:2:0:-6.6
T:155554123:AA2:ce32143a:2:0:0
T:155554223:RR2:ce32143a:3:6.65
T:155554223:LOC:ce32143a:3:0:-6.65
T:155554223:AA2:ce32143a:3:0:0
T:155554322:RR2:ce32143a:4:6.54
T:155554322:LOC:ce32143a:4:0:-6.54
T:155554322:AA2:ce32143a:4:0:0
T:155554632:RR2:ce32143a:7:6.66
T:155554632:LOC:ce32143a:7:0:-6.66
T:155554632:AA2:ce32143a:7:0:0
T:155554634:RR2:8e320fab:1:0.14
T:155554634:LOC:8e320fab:1:0:-0.14
T:155554634:AA2:8e320fab:1:0:0
T:155554722:RR2:8e320fab:2:0.15
T:155554722:LOC:8e320fab:2:0:-0.15
T:155554722:AA2:8e320fab:2:0:0
T:155554723:RR2:ce32143a:8:6.66
T:155554723:LOC:ce32143a:8:0:-6.66
T:155554723:AA2:ce32143a:8:0:0
```

Figure 26. Part of Log file

3.4 Geolocation Results

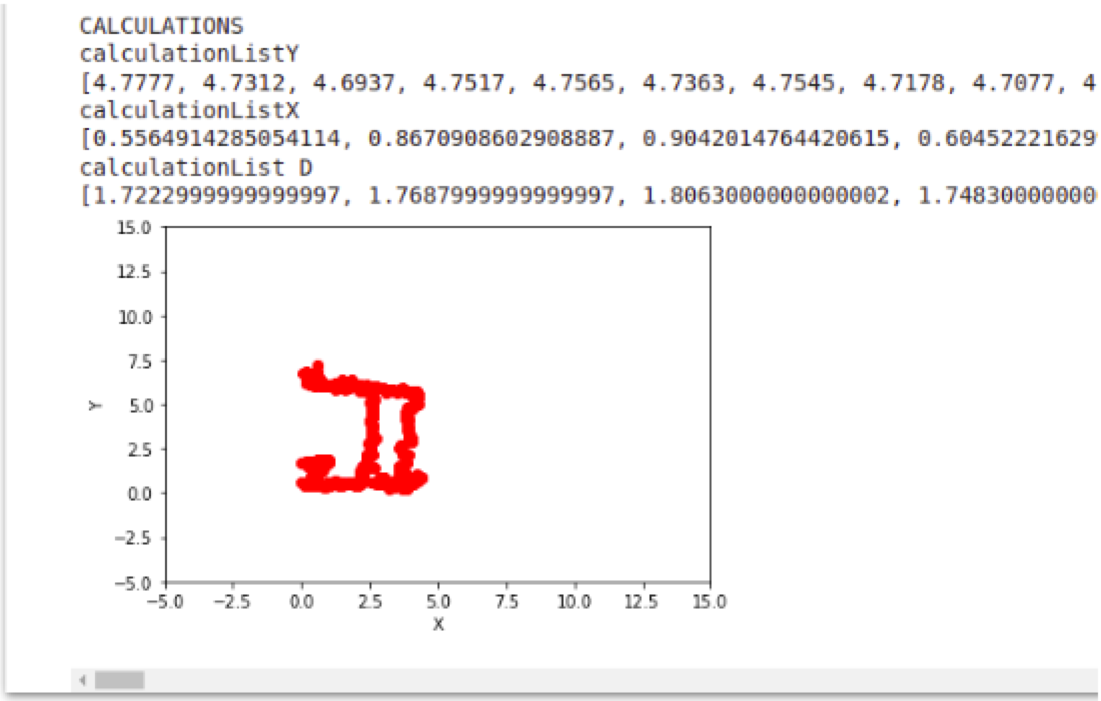


Figure 27. Geolocation Result I

As can be seen in the figure , a desired result was obtained in the test performed. This created shape was carefully followed and made on the same planes. The heights of the devices are set to the same. The test started after running PDoA.exe and the places the device traveled are shown on the coordinate system.

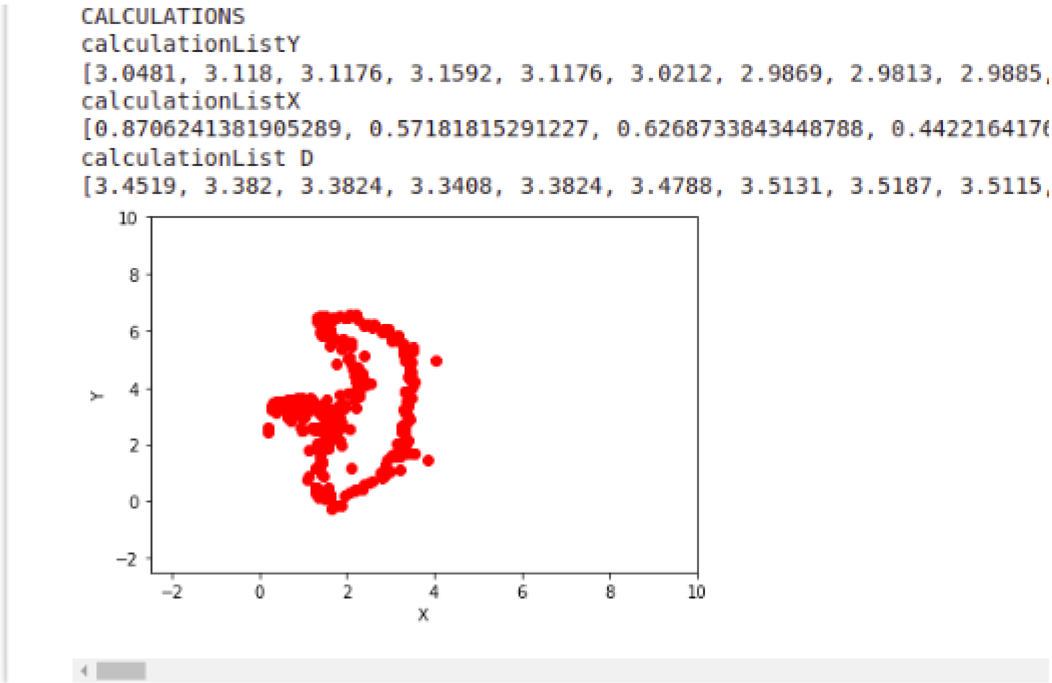


Figure 28. Geolocation Result II

According to the figure, the desired result was obtained in the test. But deviations do occur. This created shape was carefully followed and made on the same planes. The heights of the devices are set to the same. The test started after running PDoA.exe and the places the device traveled are shown on the coordinate system. One of the main causes of deviation is the environment. When a transition is made to a natural environment, deterioration occurs. For example, people passing in front of the device can be given.

3.5 Error Rates

MAPE was used when calculating error rates. The mean absolute percentage error (MAPE), also known as mean absolute percentage deviation (MAPD).

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\frac{|A_i - F_i|}{A_i}$$

A_i is the actual value

F_i is the forecast value

n is total number of observations

Figure 29. MAPE Calculation

MAPE	Interpretation
< 10 %	Very good
10 % - 20 %	Good
20 % - 50 %	OK
> 50 %	Not good

Figure 30. MAPE Scale

The calculations was commented with these MAPE values.

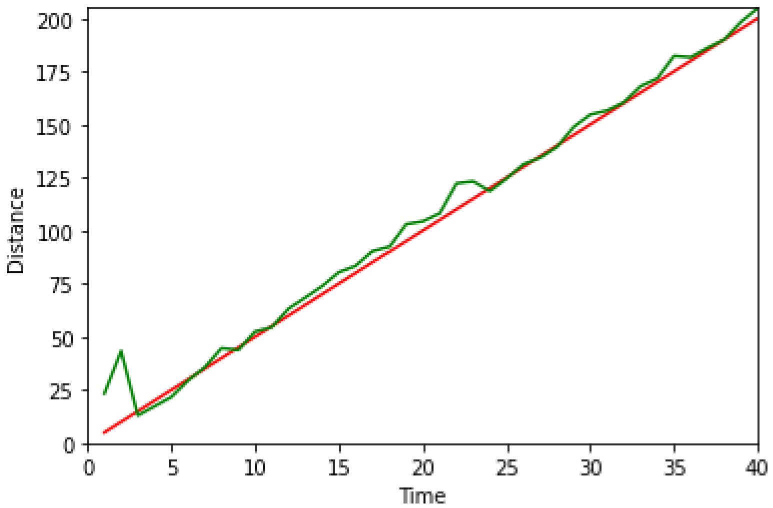


Table 3. Testing Looking Each Other

THE ANTENNAS OF THE DEVICES ARE LOOKING AT EACH OTHER WHEN THIS MEASUREMENT IS MADE !!!

Error rates are plotted on the graph. While making these calculations, the devices are in the same plane and stable position. The measurements made for each "5 centimeters" were averaged. The work continued until the length of "2 meters". As can be seen in the graph, the most deviation occurs on the first "15 centimeters".

MAPE = 21.421501615868394 OK!

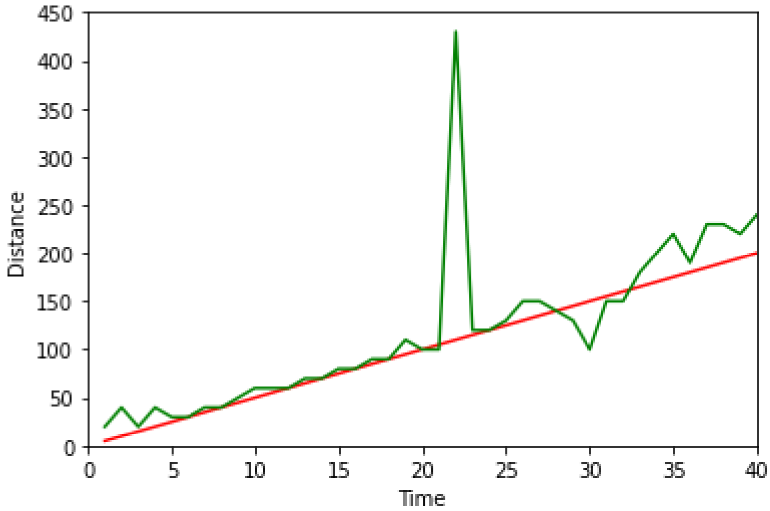


Table 4. Testing No Looking Each Other

WHEN THIS MEASUREMENT IS MADE, THE ANTENNAS OF THE DEVICES ARE LOOKING TO EACH ONE !!!

Error rates are plotted on the graph. While making these calculations, the devices are in the same plane and stable position. The measurements made for each "5 centimeters" were averaged. The work continued until the length of "2 meters". As can be seen in the graph, the most deviation occurs on the first "100 centimeters". In addition to this, when the distance is bigger than 1 meter, the measurements will continue with faults.

MAPE = 34.09314155845813 OK!

3.6 Future Reaserch

In this study, one-dimensional measurements were made due to the antenna used in the PDoA application. Instead, more precise solutions can be obtained with different algorithms by using different antennas. Apart from this, it is necessary to make various installations in the determined area in order to be able to measure with UWB. In the future, autonomous vehicles that enable these installations to be made automatically can be designed. Environmental conditions, obstacles caused by the installed environment can adversely affect the signal quality. In order to prevent this, it is important to work with more cards and to write appropriate algorithms for it.

Apart from this, it is possible to make UWB widespread in daily use by reducing the dependency on computers to collect data and bringing it to dimensions that can be easily carried in daily use, such as systems integrated into mobile phones or cards.

3.7 Applications

The algorithm created with the received one-dimensional position data provides position determination in the 2-dimensional plane. This system is used to provide various services in a properly positioned closed area, such as home, school, market, factory, hospital.

3.8 PDoA Shield

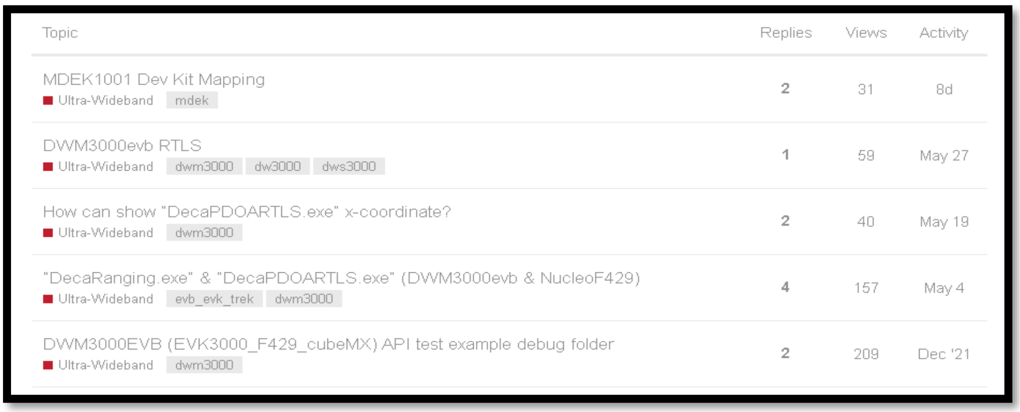
The current card, which has been used in PDoA studies, provides one-dimensional location information. Higher level cards that can be used instead can allow us to get 2 dimensions of data from a single anchor. Thus, 3D position or more precise results can be obtained thanks to the sheilds used.

3.9 Recommendation

Working in such a project is very productive for personal development. It should be noted that it is especially important that the project has progressed with research throughout the entire process.

Is it always good to use a new technology? Using a very new technology is good and enjoyable. However, this event can also lead to negative consequences. In an old and regulated technology, the margin of error is expected to be stable. This is a very important advantage to find more resources in the progress process.

On the other hand, new technologies can be faster and more efficient. But when starting this process, there may be problems to find resources. This event should not be ignored. The cards and technologies used in this project are very new. There were shortcomings on the sourcing side. In this process, we have received help from "FORUM" sites many times. This is shown in Figure 30.



Topic	Replies	Views	Activity
MDEK1001 Dev Kit Mapping Ultra-Wideband mdek	2	31	8d
DWM3000evb RTLS Ultra-Wideband dwm3000 dw3000 dws3000	1	59	May 27
How can show "DecaPDOARTLS.exe" x-coordinate? Ultra-Wideband dwm3000	2	40	May 19
"DecaRanging.exe" & "DecaPDOARTLS.exe" (DWM3000evb & NucleoF429) Ultra-Wideband evb_evk_trek dwm3000	4	157	May 4
DWM3000EVB (EVK3000_F429_cubeMX) API test example debug folder Ultra-Wideband dwm3000	2	209	Dec '21

Figure 31. Qorvo Forum Stats

4. CONCLUSION

4.1 Discussion and Future Work

Thanks to the dwm3000 evb arduino shield positioned on the Nucleo F429ZI board, it is desired to make our indoor positioning system workable. For this, receivers and transmitters working together with the necessary algorithms must be in continuous data exchange with each other and transmit this to the user with the help of a computer. In the continuation of the studies, tests will be carried out on the accuracy of the specified location.With more than 2 receivers requested in the project, the system will provide the opportunity to determine a 3-dimensional position. Finally, efforts will be made to further reduce power consumption costs and downsize the installed system.

4.2 Social Impact

The average person's time spent food shopping was surveyed on a social level. The average person spends 41 minutes food shopping. Each food shopping trip takes roughly 41 minutes, according to the Time Use Institute. When you consider that you go grocery shopping three times a week, this statistic adds up to more than two hours. People will be able to get the product they want without wasting time as a result of the project's overall goal.

4.3 Economical Impact

Markets, warehouses, and supermarkets that benefit financially from the project will be able to sell more than they do now. This is due to the fact that consumers can readily access the items they require. Because this convenience will be useful. The markets that use the UWB project will be chosen more since this convenience will suit the interests and wants of the people. Markets that use the project may also be able to employ fewer people.

4.4 Environmental Impact

From the standpoint of the environment and natural life, this initiative appears to have no beneficial or bad effects.

4.5.Standards

Authorities published a document outlining the UWB transmission and reception restrictions in February 2002. A UWB signal is defined as any signal with a fractional bandwidth larger than 20% or a bandwidth greater than 500 MHz, as defined in this specification. The published rule also outlines communication and measurement equipment' access to the 7.5 GHz unlicensed spectrum between 3.1 and 10.6 GHz.

When compared to the PSD of UWB signals observed by a UWB, narrowband signals present in the UWB range, such as IEEE 802.11a transmissions, can have high power spectral density (PSD) levels. A decrease in operating speed can be predicted based on the UWB bit error rate. The notched UWB antenna and filters are intended to allow UWB and narrowband devices to work together.

Also, the standards can seen with the codes of Harmonized National Standard No;

- TS EN 302500-2 V1.2.1
- TS EN 302500-2 V2.1.1
- TS EN 302498-2 V1.1.1
- TS EN 302065 V1.2.1
- TS EN 302065 V 1.1.1
- TS EN 301489-33 V1.1.1
- EN 301 489-33 V1.1.1
- EN 302 435-2 V1.3.1
- EN 302 065 V1.2.1

4.6 Cost Analysis

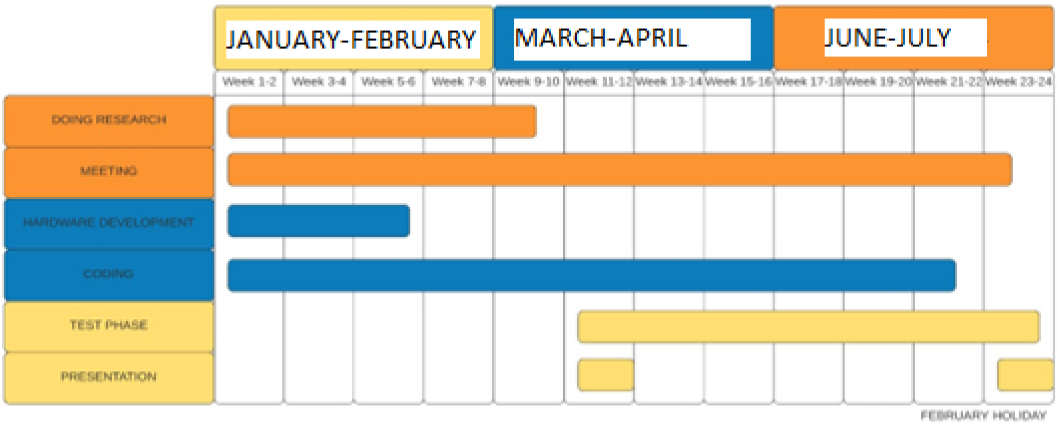
A consumption table **(Table I.)** was created as a result of the financial research carried out in line with the aims and means of the project.

As a result of this table created, the project expenses are stated in general.

ALL EXPENSES	
DWM3000EVB board	$19.50 for a product
NucleoF429ZI development board	$23.0 for a product
(10 of each development boards were purchased)	
USB Microcontroller Cable	$3.52 for a product
R&D Salary per hour	$17.0
(Each employee works 21 hours a week.)	
TOTAL	$24.000.11
(Total cost for 8 months)	

Table 5. Expense Table

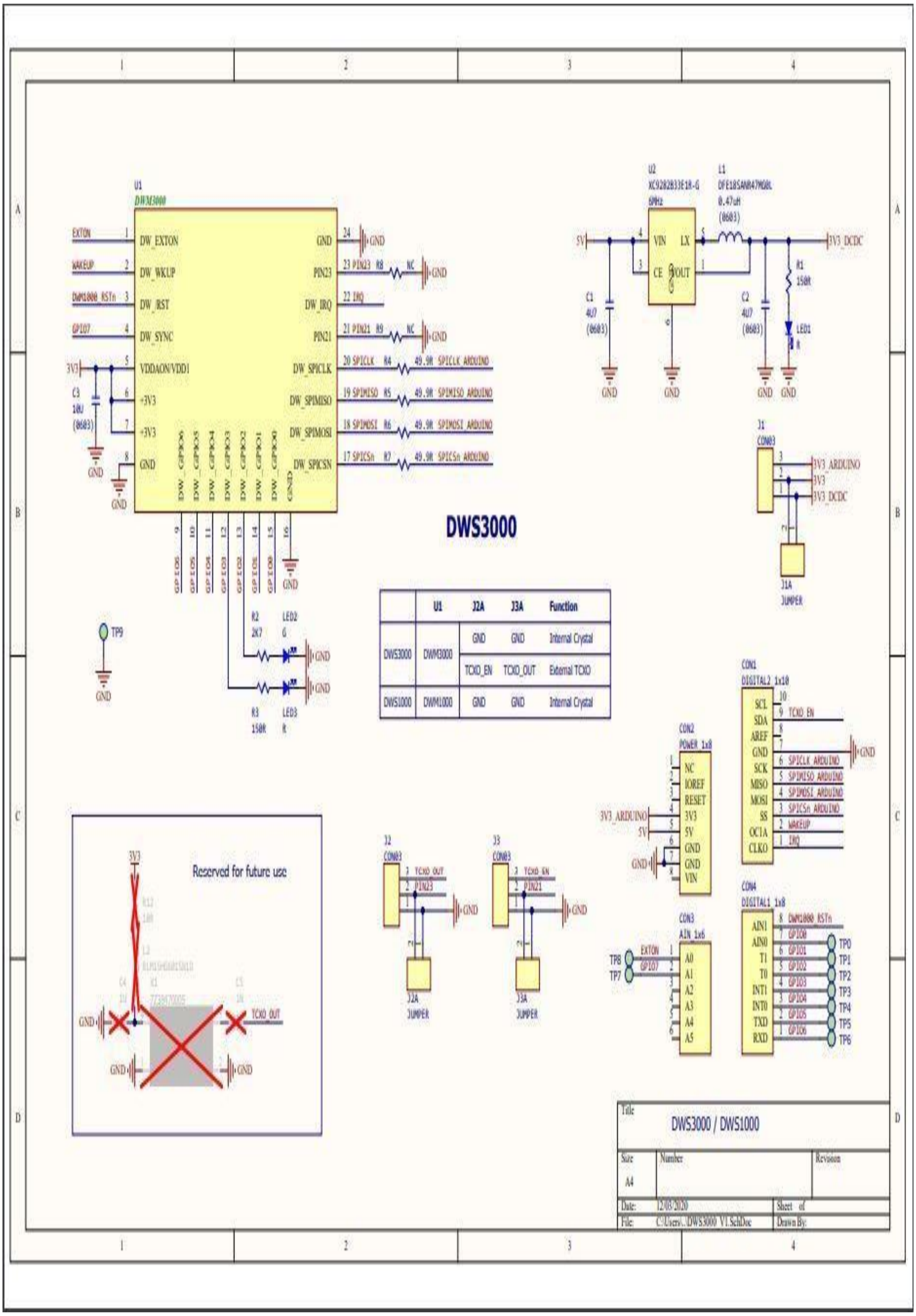
4.7 Time Table



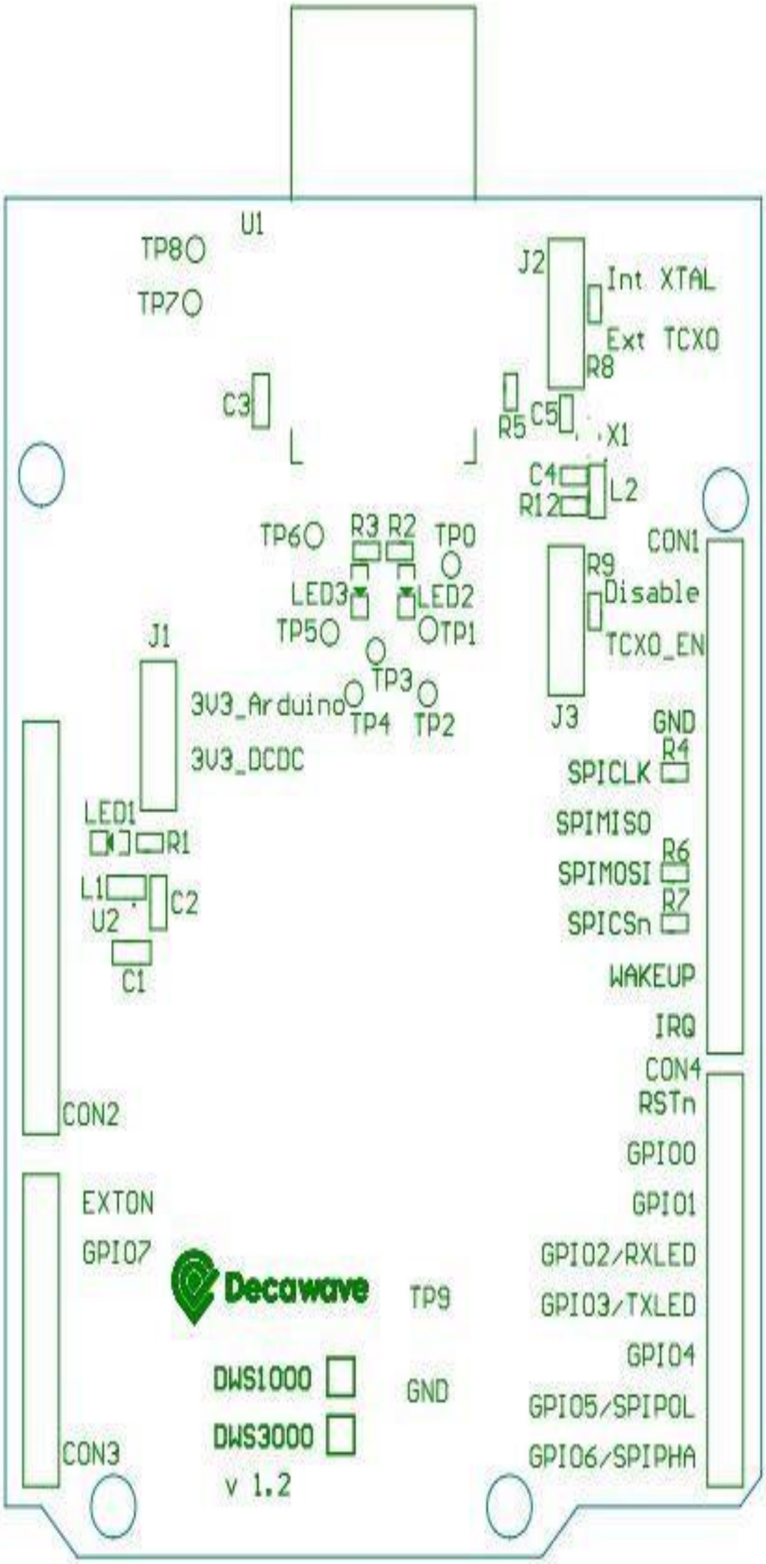
APPENDICES

APPENDIX A

BOARD SCHEMATICS



	U1	J2A	J3A	Function
DWS3000	DWM3000	GND	GND	Internal Crystal
		TCXO_EN	TCXO_OUT	External TCXO
DWS1000	DWM1000	GND	GND	Internal Crystal


U1
TP8
TP7
J2
Int XTAL
Ext TCXO
R8
C3
R5
C5
X1
C4
R12
L2
TP6
R3
R2
TP0
CON1
R9
Disable
LED3
LED2
TP5
TP1
TCXO_EN
J1
TP3
3V3_Arduino
TP4
TP2
J3
GND
R4
SPICLK
3V3_DCDC
LED1
R1
SPIMISO
R6
SPIMOSI
L1
C2
U2
R7
SPICSn
C1
WAKEUP
IRQ
CON4
RSTn
CON2
GPIO0
EXTON
GPIO1
GPIO7
Decawave
GPIO2/RXLED
TP9
GPIO3/TXLED
GPIO4
DWS1000
GND
GPIO5/SPIPOL
DWS3000
CON3
GPIO6/SPIPHA
v 1.2

REFERENCES

[1] https://www.qorvo.com/products/p/DWM3000EVB

[2] https://lucid.app/lucidchart/04751bfd-dae8-46c9-98c7-24f539c35bdf/edit?page=0_0&invitationId=inv_dc2e0e65-1b7b-4c73-850cb711b049d92d#

[3] https://www.st.com/en/evaluation-tools/nucleo-f429zi.html

[4] https://qz.com/1677747/americans-are-spending-way-less-time-shopping/

[5] https://www.qorvo.com/feature/uwb-solutions-compatible-with-apple-u1

[6] https://www.sciencedirect.com/topics/engineering/ultrawideband#:~:text=Ultra%2Dwideband%20(UWB)%20refers,lowpower%2C%20short%2Drange%20communications.

[7] https://en.wikipedia.org/wiki/Ultra-wideband

[8] https://rtloc.com/technology/

[9] https://www.resmigazete.gov.tr/eskiler/2012/07/20120703-20.htm

[10] https://intweb.tse.org.tr/

[11] https://forum.qorvo.com/t/mdek1001-dev-kit-mapping/12224/3

[12] https://forum.qorvo.com/t/dwm3000evb-rtls/12229/2

[13] https://forum.qorvo.com/t/how-can-show-decapdoartls-exe-x-coordinate/12192/3

[14] https://forum.qorvo.com/t/decaranging-exe-decapdoartls-exe-dwm3000evb-nucleof429/11913/5

[15] https://forum.qorvo.com/t/dwm3000evb-evk3000-f429-cubemx-api-test-example-debug-folder/11420/3

[16] https://github.com/espressif/arduino-esp32

[17] https://github.com/lijx10/uwb-localization

[18] https://github.com/KitSprout/UWB-Node