INDOOR POSITIONING IN GPS DENIED AREAS USING ULTRA WIDE BAND TECHNOLOGY

PROJECT REPORT BY:

PRASHANT DWIVEDI

Contents

INTRODUCTION:

Positioning is the process of determining positions of people, equipment, and other objects.
It has recently been an active research area in which much of the research focuses on utilizing existing technologies to address the problem of positions' determination. Positioning can be classified into two types, depending on the environment in which the positioning is conducted: outdoor positioning and indoor positioning. Whereas outdoor positioning is performed outside buildings, indoor positioning is performed inside buildings (e.g., houses, hospitals, and malls). Different applications may require different types of positioning technologies that fit their needs and constraints. For example, Global Positioning System (GPS) is a technology that is suitable and efficient for outdoor spaces rather than indoor spaces because satellite radio signals cannot penetrate solid walls and obstacles.
Indoor positioning systems (IPSs) determine the position of an object in a physical space continuously and in real-time.

In this Project, a research activity aimed at developing an indoor positioning system, where we cannot rely on other positioning methods like GPS due to its unavailability or Cost Factor, is presented.

Problem Statement:

"To develop a Technology or System that can measure Position In Indoor Environment in GPS denied Areas with acceptable accuracy."

Motivation to Solve This Problem:

For this project we have used UWB (Ultra-Wideband) Technology to overcome this. Ultra-Wideband (UWB) is radio-based distance-measuring system which provides a measure of distance between two transceivers by measuring the time-of-flight of pulses.

Why UWB?

There can be an obvious question in your mind that only UWB technology, why not others like Wi-fi, Infrared and many others.

Actually, UWB has many distinct properties, which can not be found in other system, like:

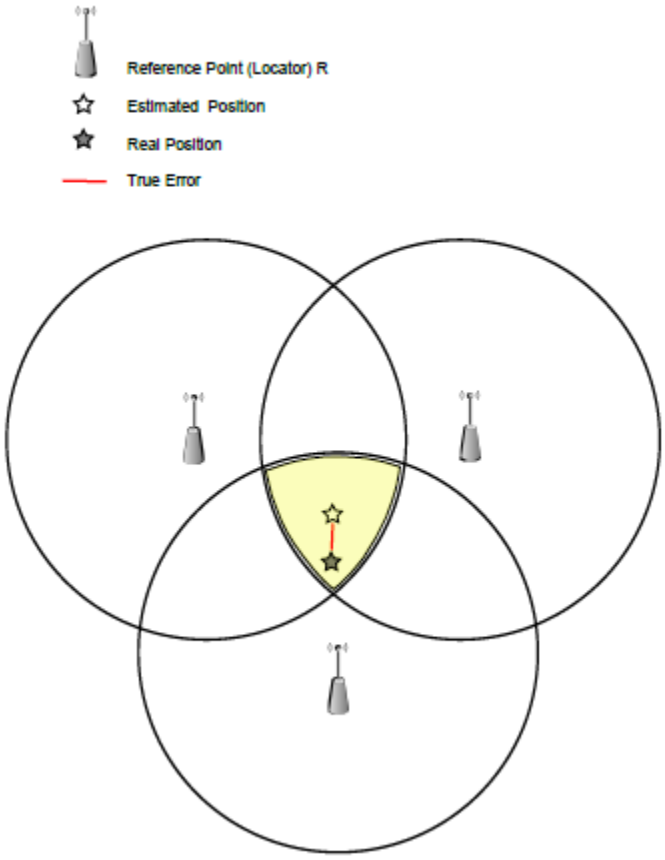
- UWB is an RF signal occupying a portion of the frequency spectrum that is greater than 20% of the center carrier frequency, or has a bandwidth greater than 500 MHz UWB is a communication channel that spreads information out over a wide portion of the frequency spectrum. This allows UWB transmitters to transmit large amounts of data while consuming little transmit energy. The high data rate of UWB can reach 100 Megabits per second (Mbps)

- UWB can be used for positioning by utilizing the time difference of arrival (TDOA) of the RF signals to obtain the distance between the reference point and the target (tag).

- High accuracy positioning, even in the presence of severe multipath, effectively passes through walls, equipment, and any other obstacles. UWB will not interfere with existing RF systems if properly designed.

- UWB positioning techniques can in fact give real-time indoor precision tracking for several applications such as mobile inventory and locator beacons for emergency services, indoor navigation for blind and visually impaired people, tracking of people or instruments, and military reconnaissance.

Approach to Solve the Problem:



We used Reference Point (anchors) fixed at walls and Estimator (tag) whose position is to be calculated.

Technology and Materials Used:

1. MDEK DWM1001 Kit:

The DWM1001 module comes pre-loaded with embedded firmware which provides two-way ranging (TWR) real time location system (RTLS) functionality and networking. The module can be configured and controlled via its API, which can be accessed through a number of different interfaces allowing flexibility to the product designer. The details of the API are described in document [2]. Additionally, Decawave also provides the module firmware in the form of libraries and source code along with a build environment so that user can customize the operation and/or add their own function.

Overview

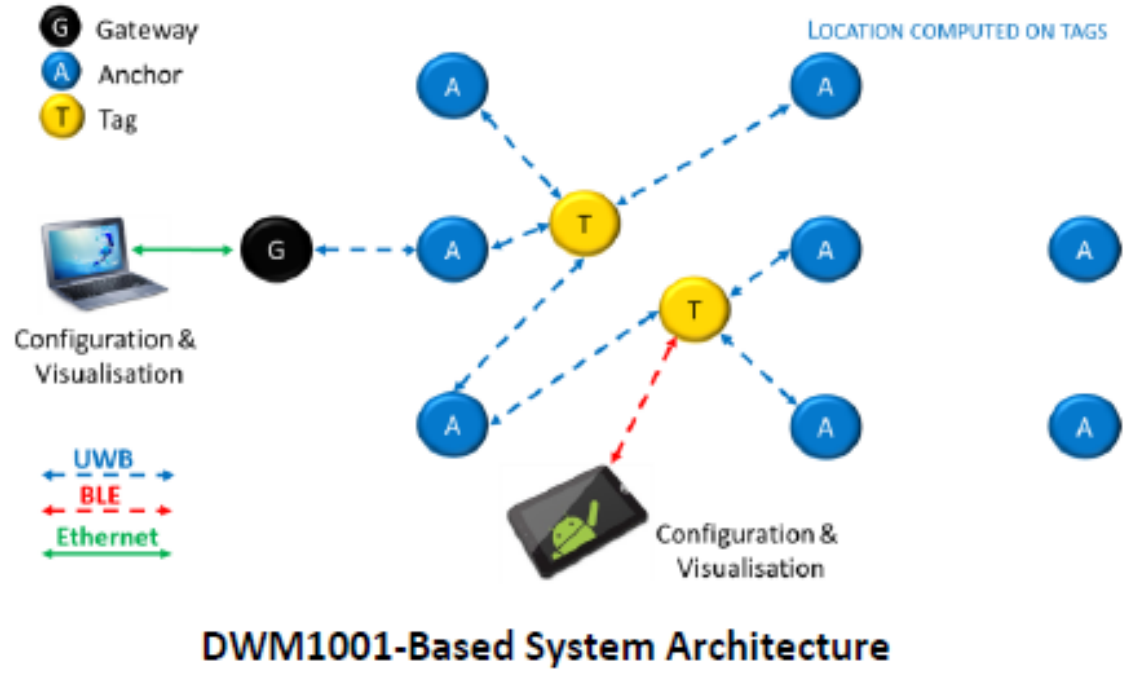
- UWB and *Bluetooth®* module based on Decawave's DW1000 IC and Nordic Semiconductor nRF52832 SoC.
- On board motion sensor and integrated antennas
- Embedded firmware available enabling: • Accurate UWB-based Real-Time Location Systems (RTLS)
- Data encrypted network connectivity.

Benefits

- Build scalable Two-Way-Ranging (TWR) RTLS systems with up to thousands of tags
- Accelerates product designs for faster time-to-market & reduced development costs
- Same module for anchor, tag and gateway designs
- No RF design required
- Embedded DRTLS firmware (DWM1001 SS-TWR RTLS) reduces software development Effort.
    - Firmware API to customise embedded user application
    - SPI, UART and *Bluetooth®* APIs to access DWM1001 from an external device
- On-board *Bluetooth®* for connectivity to phones/tablets
- Low-power hardware and software architecture for longer battery life

DWM1001 Module:

Connection View Of Network:



DWM1001-Based System Architecture

Document Link:

All Documents (zip file) about this DEV Board, android application can be downloaded from this link (or from Decawave.com)

DWM1001, DW10001-DEV and MDEK1001 Documents, Source Code, Android Application & Firmware Image

- Unzip the file (DWM1001_DWM1001DEV_MDEK1001_Sources_and_Docs)
- Install Android App on phone.(\Android Application)
- Read all documents given in \DWM1001\Product_and_Design_Documents.

Now , there are two ways to get positon,first on Android Phone,second on Laptop or PC.Brief Introduction is given for both methods.(for details follow Documents downlowded before).

Android App RTLS:

To perform RTLS on android phone

- Firstly, you need to install Android app.
- Now you need to update firmware of all the nodes and tags to latest version.For that install **Segger J-Flash Lite 6.22g** for re-Flashing DWM1001-DEV boards .(link below)(install the same version).
  https://www.segger.com/downloads/jlink/JLink_Windows_V622g.exe

  Tip: J-Flash Lite can be found in the folder where you have install the above file (mine is C:\Program Files (x86) \SEGGER\JLink_V622g).
- Follow the steps below(DWM1001 Gateway_Quick_Devlopment_Guide.pdf) "DWM1001_PANS_R2.hex" can be found in "\DWM1001_DWM1001-DEV_MDEK1001_Sources_and_Docs_v9\DWM1001\Factory_Firmware_Image" folder.

Connect the DWM1001-DEV to a computer through the USB interface and follow the steps below:

1. Launch J-Flash Lite
2. Select the NRF52832_XXAA target, SWD interface and 1000 kHz clock speed (see Figure 1)

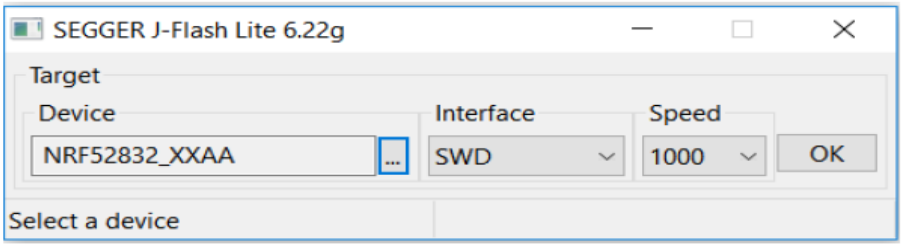


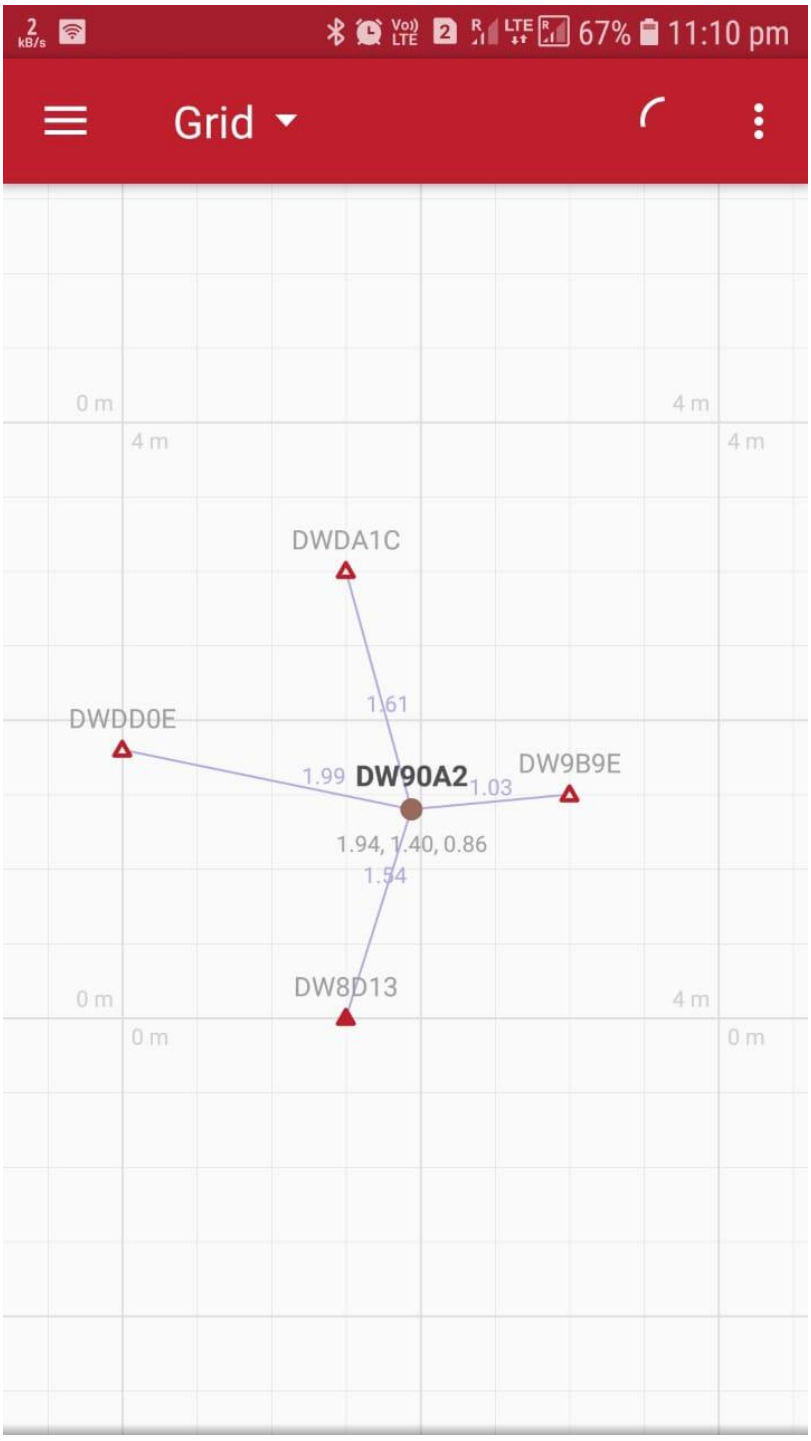
Figure 1 : Segger J-Flash configuration

3. Select "DWM1001_PANS_R2.hex" as data file and click on "Program Device"

Devices can be flashed successively without quitting J-Flash Lite. To quickly flash multiple DWM1001-DEV boards, it is only necessary to disconnect the flashed board, connect a new one over USB and click on "Program Device" again.

- Fix the anchors and power them. Power the tag.
- Watch this Video(or read **MDEK1001_Quick_Start_Guide.pdf**) to perform RTLS.
  https://www.youtube.com/watch?v=hI8EaU5nOmI

You can get this type of window.

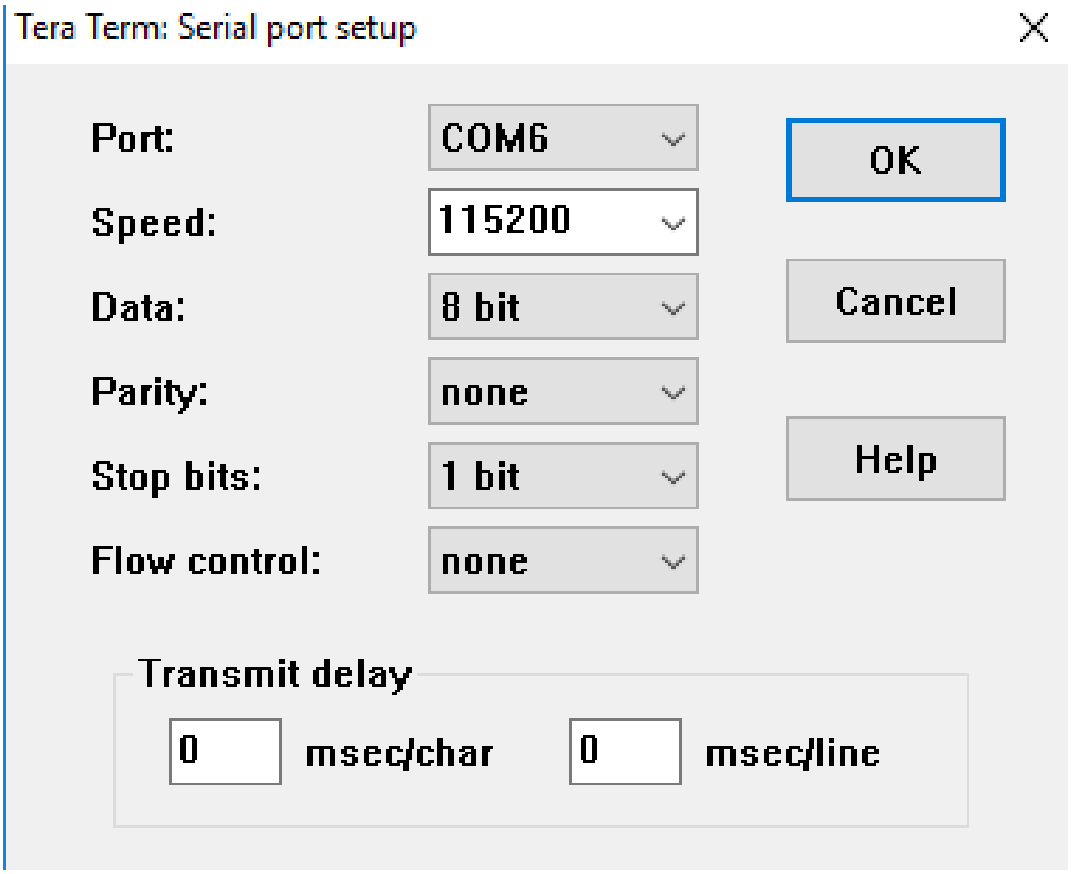


- Tip: Don’t rely on auto-positioning ,I would suggest to input your own measured position of anchors for better results.

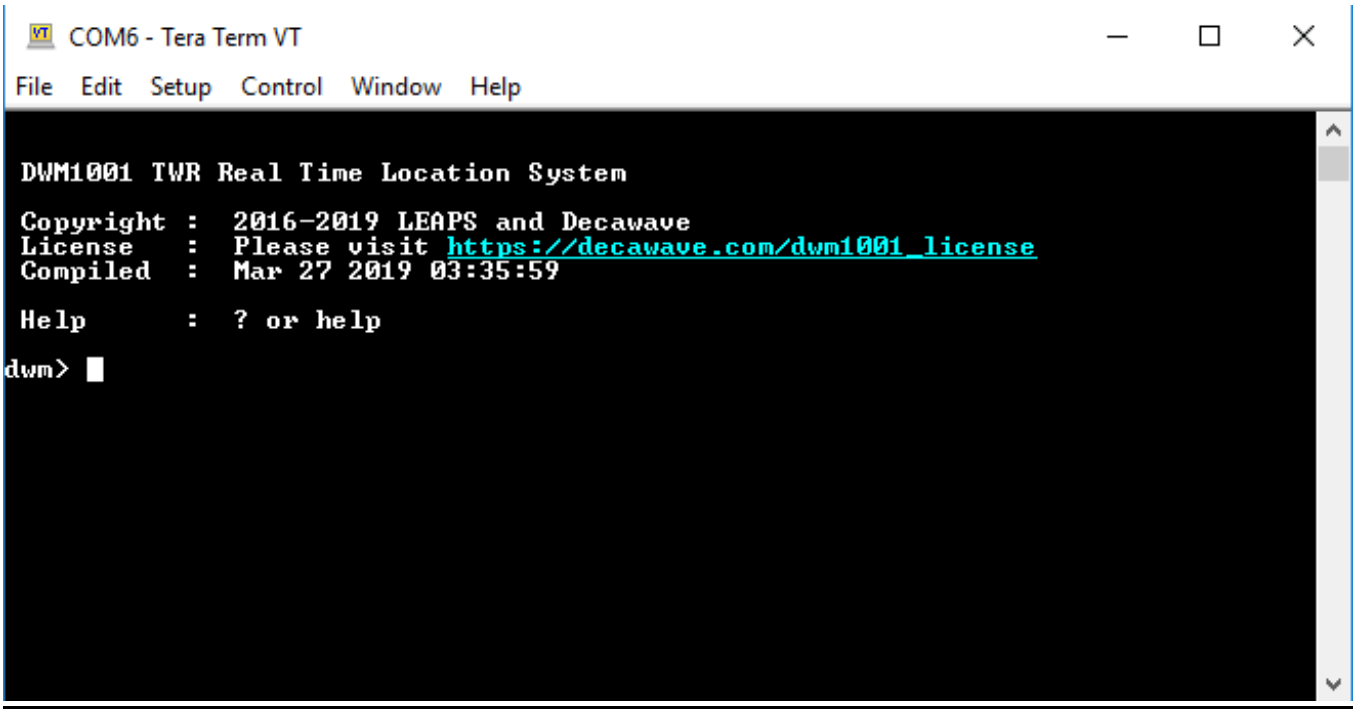
On PC Using the UART shell mode:

- Download and install Tera Term for Windows: (v 4.102)
  https://osdn.net/projects/ttssh2/releases/

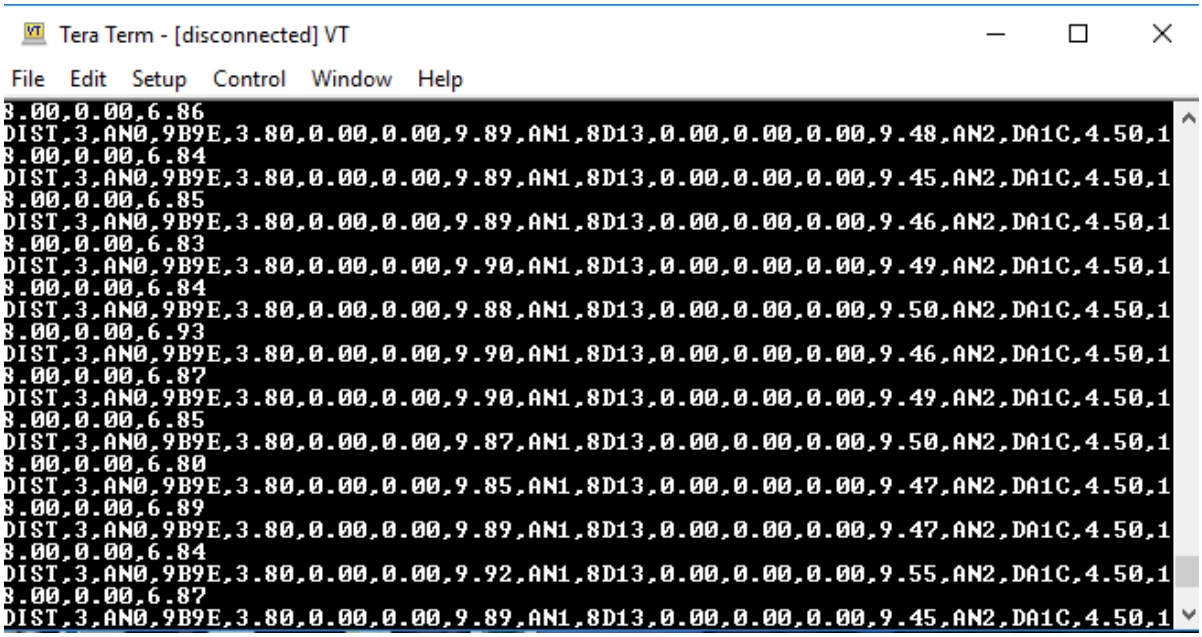
- Connect the DWM1001-DEV to the PC over USB and launch TeraTerm. Click on the “Setup Tab”, select “Serial Port”. The serial port configuration should be as defined as in Figure.(Port can different as in your case)



- In the shell, press “Enter” twice in order to start the DWM1001 UART shell mode. The Device should answer should match Figure 4:



- Type si to get system info and hit enter.
- Type lec to get Position(Make sure anchors are powered).These above steps will work if you have already setup your network with android app,if not you need to setup those steps here again in TeraTerm .For this you can go through DWM1001 Gateway_Quick_Devployment_Guide.pdf (section 3.2 in detail.).



We can see, printing of a line which gives information like anchors (AN0,AN1,AN2,AN03) and distances from the same.

This line is very useful as we can read this and can get these distances to get positions from our own algorithm like Least Square Method.

2. Raspberry Pi Kit:

Now, to get our Data and modification of same we need to setup a gateway, thus we will use **Raspberry Pi 3B** for that.
So, you will need:

Software:

- Raspberry Pi Model 3B Raspbian image: DRTLS_raspbian_R2.0.img
  {Included in zip file downloaded earlier at
  (DWM1001_DWM1001DEV_MDEK1001_Sources_and_Docs_v9\Raspbian_image\)}

Hardware:

- Raspberry Pi 3 model B (R3B) with a power supply and eventually ethernet cable (for Wi-fi Connection you must have a Smartphone)
- 16 GB micro-SD card and a micro-SD card reader
- 2x13 GPIO Header (long pin) for Raspberry Pi 3 model B
- A PC running Windows

Software Tools: (license free)

- Etcher – for writing the Raspbian image file to an SD card
  https://www.balena.io/etcher/
  Install it.
- Advanced IP Scanner – for finding the Raspberry Pi's network IP address .
  https://www.advanced-ip-scanner.com/
  Install it.
- SSH client such as MobaXterm to communicate with Raspberry Pi and display.
  https://mobaxterm.mobatek.net/download-home-edition.html

Connecting Raspberry to PC:

- Firstly, you have to flash OS to raspberry pi, follow the steps below:
  - Connect the micro-SD card to the computer and launch etcher. In "select image", chose "DRTLS_raspbian_R2.0.img" file. The sd-card should automatically be detected. Click on "flash".

    Note 1: Before Flashing the SD-Card ,make sure that SD card is formatted as FAT32.
    Note 2: If you want to flash SD Card again you have join the Partition made before, for that watch
    https://www.youtube.com/watch?v=8ae2aWYv_iI

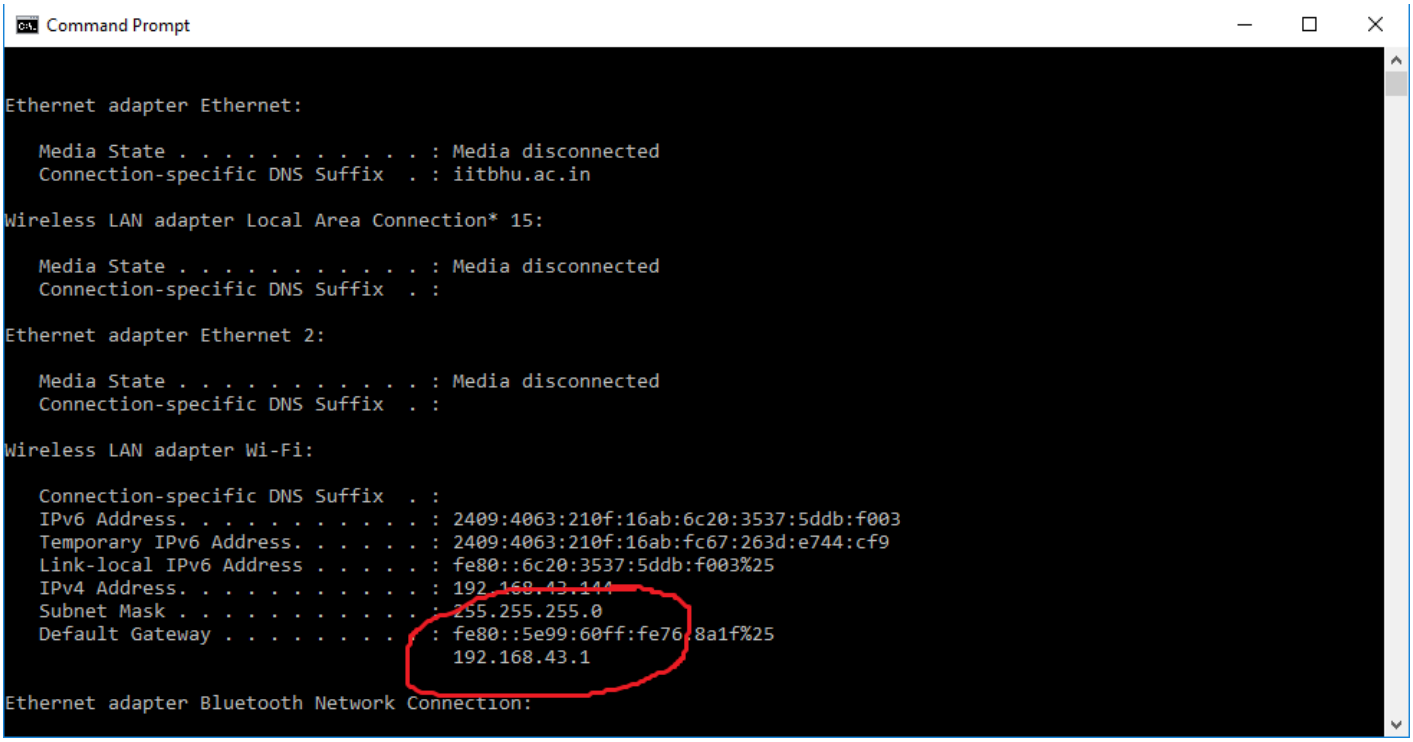
  - Create a file name "ssh" (having no extension)in the Boot Directory of SD Card.
- Now, to connect Raspberry Pi to PC there are two ways(prefer latter one):
  - Ethernet: For Connecting Through Ethernet you can watch below videos:
    https://www.youtube.com/watch?v=AJ7skYS5bjI

  - Wi-fi: For W-fi Connection follow the below Steps:

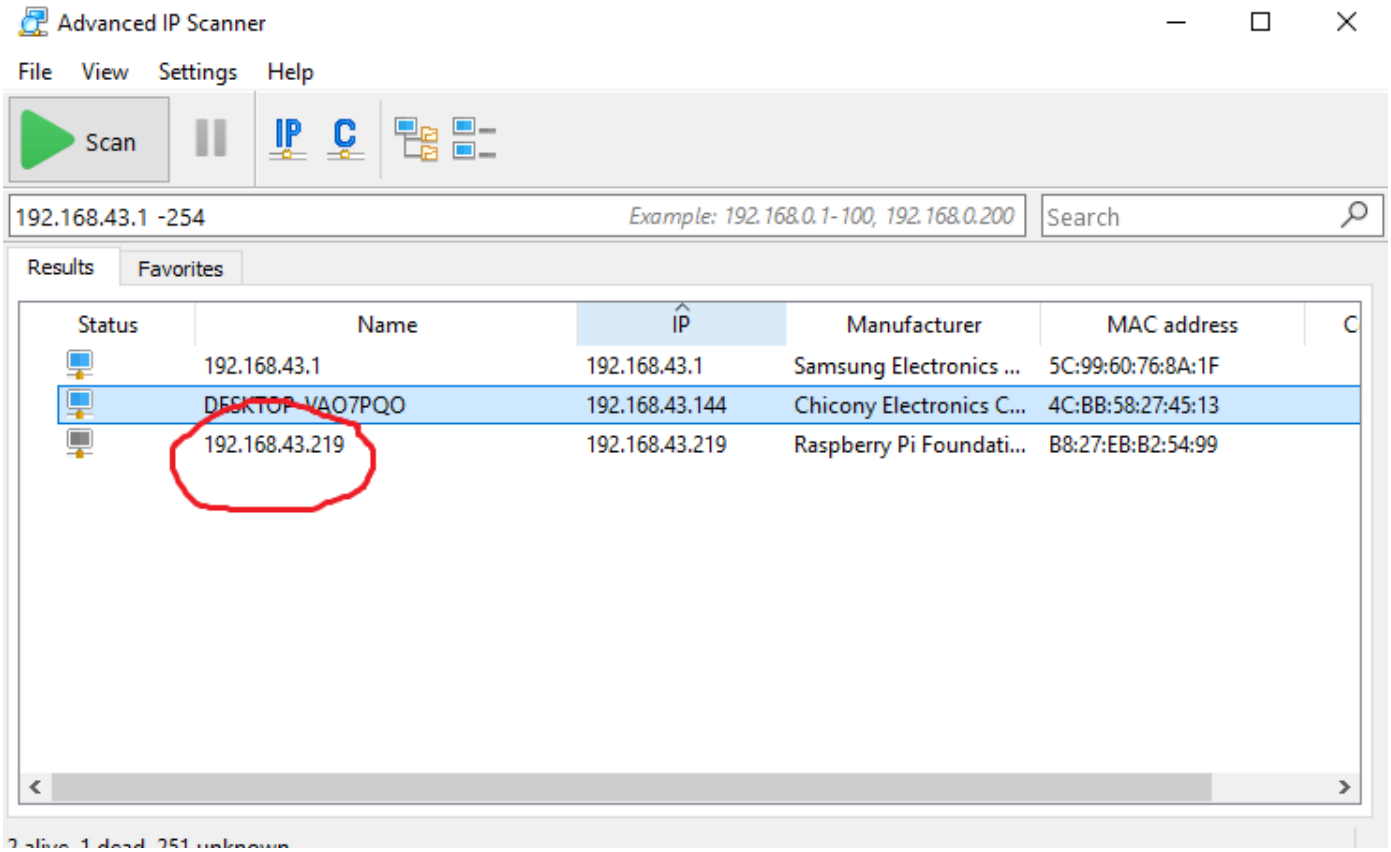
    - open the SD card in the windows explorer. By default, it will be called "boot". Create a file "wpa_supplicant.conf" in the boot directory using a text editor tool such as notepad. Note .conf must be the file extension and not only part of the name.
    - Copy the following in the wpa_supplicant.conf file previously created, altering the ssid and psk field with your network()credentials(make sure that "ssh" file is also there).

```
country=IN
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1
network={
      ssid="Your_Network_Name"
      psk="Password"
      key_mgmt=WPA-PSK
    }
```

- Now, you need to find the IP address of your Phone.For that,start the phone hotspot and connect to PC.
  Now,open command prompt(cmd) and type “ipconfig”,you will get this type of window.You can see under Wi-fi headin the IP address of your phone which has Default gateway as 192.168.143.1
  Note it down(in your case it may be different).



- Now,Power On the Raspberry Pi (with SD card).You can see that raspberry pi is connected to your phone in hotspot menu.(If not verify the above steps carefully).Connect your PC too with hotspot.
- Now, You need to get the IP address of Raspberry Pi,for that open Advance IP scanner.Type 192.168.43.1-254 in the IP column and hit Scan,it will show you the devices connected to your Mobile hotspot. From,there you will get your Raspi IP.See image at next page.



- Note Down the IP address of raspi.
- Now, open MobaXterm. Go to Session -> SSH -> Remote Host.
  Type the Raspberry Pi IP address there (mine is 192.168.43.219).
  Click Ok.
- Now, you will be asked for login ID and password
  Login : pi
  Password : raspberry
- Now, you have access to Raspberry Pi Linux Terminal.
- For, display access to Raspberry Pi ,type “startlxde” command in terminal ,you will et Raspi home screen.
  You can adjust the toolbar etc. see below video
  https://www.youtube.com/watch?v=AJ7skYS5bjI.

- Try programming in that,play with stuff to get to use of it,
  Also,you can update python library in that using pip.
  Explore the Raspi.

Now,after connecting Raspi to PC via Wi-fi,you can perform some basic python program in Raspberry Pi and can visualize your processing through MobaXterm.

If,you face any error while performing the above steps,try searching it on Internet.

Now,since connection is made,Let's try to attach DWM1001 with Raspi,there are two ways,first through USB(through Raspi USB port) second through GPIO header pins provided on Raspi.

I used the second one,(I would recommend first one tough).Follow the steps below to setup Gateway:

- The bridge node requires a DWM1001-DEV, which can be extracted from a MDEK1001 unit. Remove the DWM1001-Dev from the plastic enclosure by unscrewing it (three screws). The battery connector can be removed as the R3B provides power to the board.
- In order to connect the DWM1001-DEV pcb to the R3B, it is necessary to solder a 2x13 GPIO header (long pins) onto the pcb.(see Fig)

- Attach DWM board to Raspi to first 13 GPIO.See fig

Figure 10: Gateway: R3B plus bridge node (DWM1001-DEV)

- Now ,you are all set up. Connect PC to mobile hotspot and Power on the Raspi,
  You would see DWM led on.
- Now, power on all Anchors.

Now,you would have remember that we used TeraTerm to connect DWM1001 to PC via Serial Communication.
And we got output as a line printed with all anchor's position and distances to TAG.

Thus,we can use that line to read distances in Raspi and modify with our program to get Position.

I will attach my codes with this report.

Documents Links for this Setup:

- You can o through DWM1001_Gateway_Quick_Deployment_Guide.pdf for details.
- Try searching errors on youtube.

After setting up the Gateway, next part is to rum out Program to get distances.

Performance Evaluation:

Now, comes the time to apply codes. But before that I would like to remember you that we can access data from DWM1001 through Serial Communication.

Thus ,first you should read about Serial Communication and how to apply it.Below are the links:

https://pyserial.readthedocs.io/en/latest/shortintro.html

https://www.w3schools.com/python/ref_string_encode.asp

https://www.geeksforgeeks.org/python-strings-decode-method/

More you can google it out.

Also ,you should have basic Python knowledge to read line and parsing data.

https://docs.python.org/3/tutorial/

Since I have connectes DWM board through Header pins,I used below command to open Serial Port:

```
DWM=serial.Serial(port="/dev/serial0", baudrate=115200)
```

```
Baudrate is data Transmission speed.
```

```
If you are connecting through USB, change it to:
```

```
DWM=serial.Serial(port="/dev/ttyACM0", baudrate=115200)
```

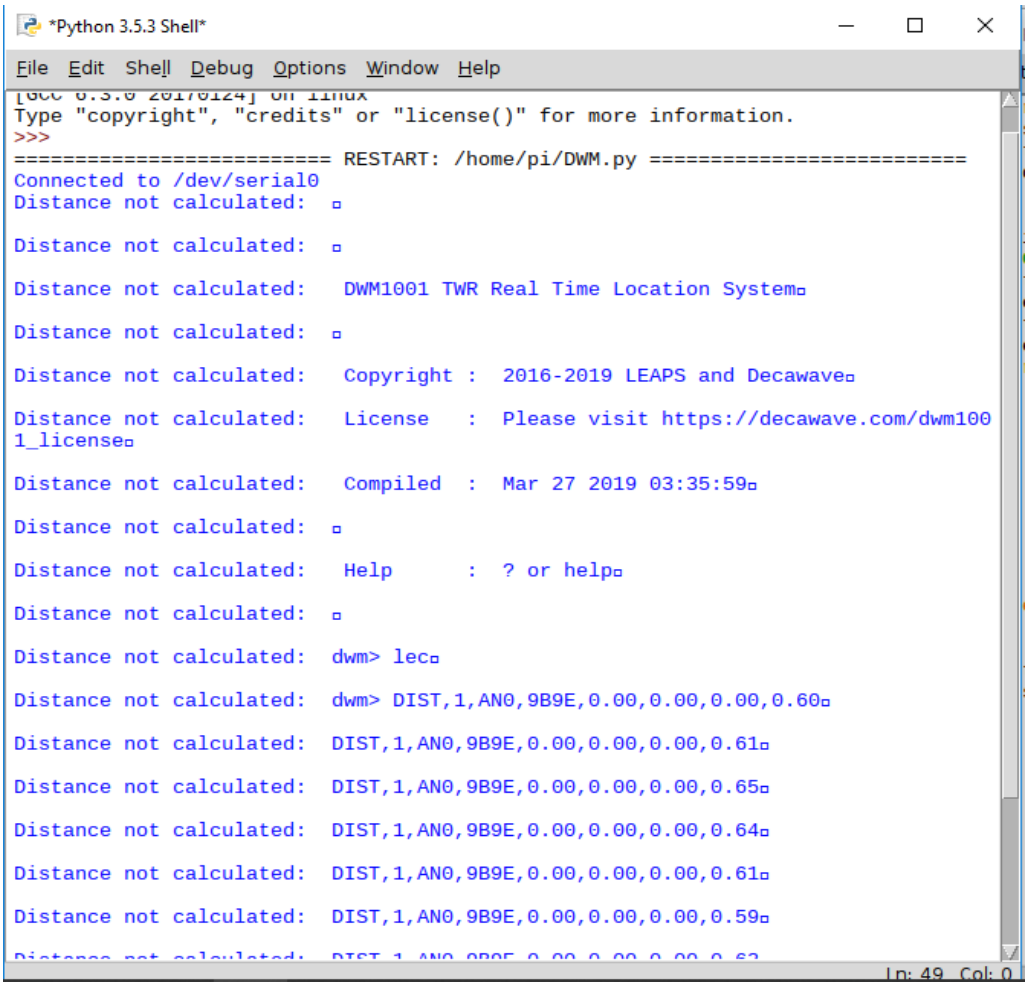
```
(Port may be different in your case.)
```

Now,you should start with basic codes like to get distances from all anchors,you can see mine below():

```
import serial
import time
import datetime
DWM=serial.Serial(port="/dev/serial0", baudrate=115200)
print("Connected to " +DWM.name)
DWM.write("\r\r".encode())
time.sleep(1)
DWM.write("lec\r".encode())
time.sleep(1)
while True:
    try:
        line=DWM.readline()
        if(line):
            parse=line.decode().split(",")
            if parse[0]=="DIST":

pos_AN0=(parse[parse.index("AN0")+2],parse[parse.index("AN0")+3],parse[parse.index("AN0")+4])
                dist_AN0=parse[parse.index("AN0")+5]

print(datetime.datetime.now().strftime("%H:%M:%S"),pos_AN0,":",dist_AN0)
            else:
```

```
            print("Distance not calculated: ",line.decode())
    except Exception as ex:
        print(ex)
        break
DWM.write("\r".encode())
DWM.close()
```

It will give you the distance with the first Anchor Connected.(see below)



Now, you can Modify the program to add functions etc.
I will provide you the Final code which give Position using Least Squares:

```
import numpy as np

import serial

import time

import datetime

import scipy

from scipy.optimize import least_squares



def
func(x1_AN0,y1_AN0,z1_AN0,x1_AN1,y1_AN1,z1_AN1,x1_AN2,y1_AN2,z1_AN2,x
1_AN3,y1_AN3,z1_AN3,a,b,c,d):

    x1, y1, z1, dist_1 = ( x1_AN0,y1_AN0,z1_AN0,a)

    x2, y2, z2, dist_2 = ( x1_AN1,y1_AN1,z1_AN1,b)

    x3, y3, z3, dist_3 = ( x1_AN2,y1_AN2,z1_AN2,c)

    x4, y4, z4, dist_4 = ( x1_AN3,y1_AN3,z1_AN3,d)



    def equations( guess ):

        x, y, z,r = guess

        return (

                (x - x1)**2 + (y - y1)**2 + (z - z1)**2 - (dist_1 -r)**2,

                (x - x2)**2 + (y - y2)**2 + (z - z2)**2 - (dist_2 -r)**2,

                (x - x3)**2 + (y - y3)**2 + (z - z3)**2 - (dist_3 -r)**2,
```

```
            (x - x4)**2 + (y - y4)**2 + (z - z4)**2 - (dist_4 -r)**2
            )

    initial_guess = (1, 1, 1,0)

    results = least_squares(equations, initial_guess)


print(datetime.datetime.now().strftime("%H:%M:%S"),",",results.x[0],",",results.x[
1],",", results.x[2])
DWM=serial.Serial(port="/dev/serial0", baudrate=115200)
print("Connected to " +DWM.name)
for j in range(1,100):
    for i in range(1,200):
        DWM.write("\r\r".encode())
        time.sleep(1)
        DWM.write("lec\r".encode())
        time.sleep(1)
        line=DWM.readline()
        if(line):
            parse=line.decode('latin-1').split(",")
            if parse[0]=="DIST":

                l=['AN0','AN1','AN2','AN3']
```

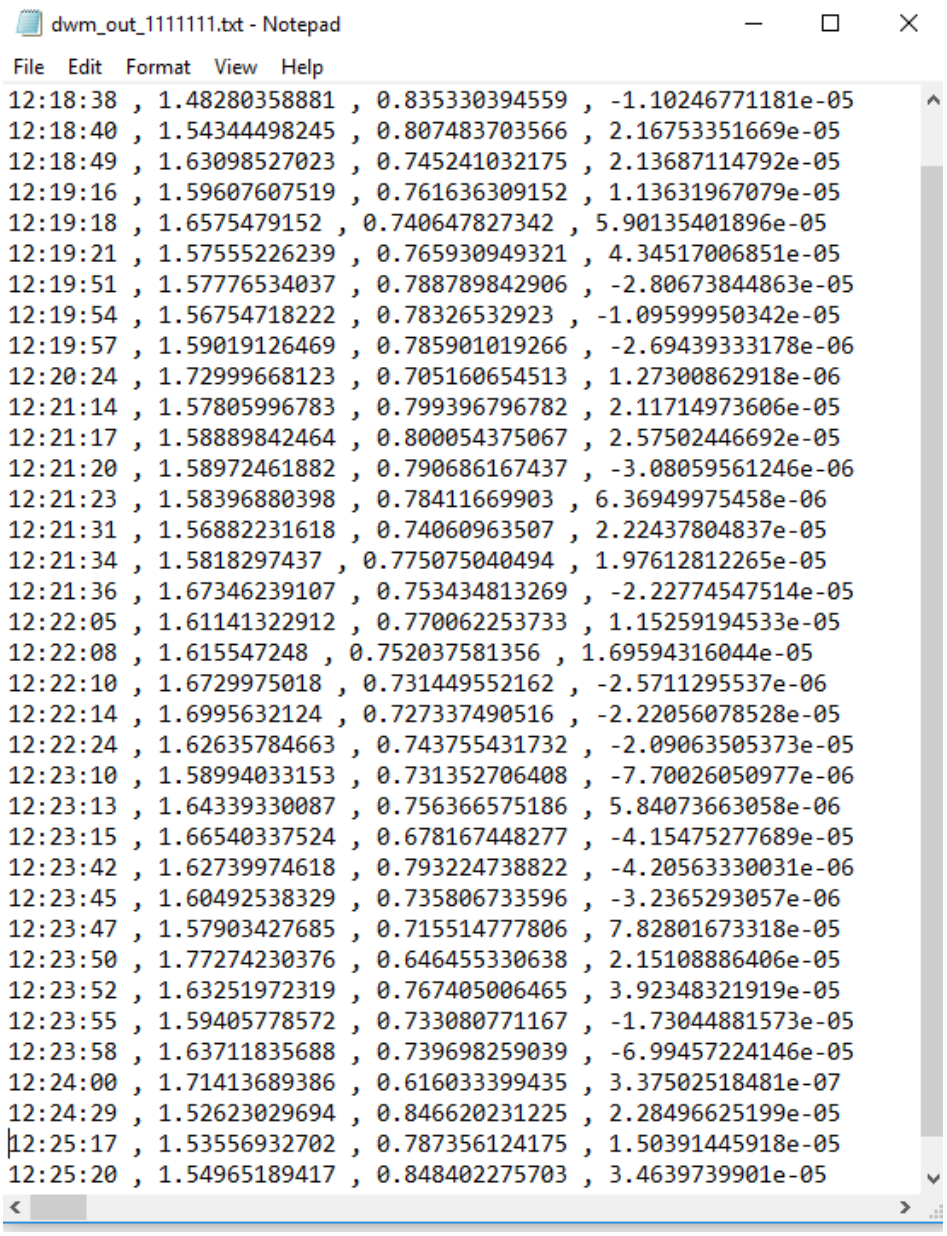
```
if(all(elem in parse for elem in l)):

  x1_pos_AN0=parse[parse.index("AN0")+2]
  y1_pos_AN0=parse[parse.index("AN0")+3]
  z1_pos_AN0=parse[parse.index("AN0")+4]
  dist_AN0=parse[parse.index("AN0")+5]

  x1_pos_AN1=parse[parse.index("AN1")+2]
  y1_pos_AN1=parse[parse.index("AN1")+3]
  z1_pos_AN1=parse[parse.index("AN1")+4]
  dist_AN1=parse[parse.index("AN1")+5]

  x1_pos_AN2=parse[parse.index("AN2")+2]
  y1_pos_AN2=parse[parse.index("AN2")+3]
  z1_pos_AN2=parse[parse.index("AN2")+4]
  dist_AN2=parse[parse.index("AN2")+5]

  x1_pos_AN3=parse[parse.index("AN3")+2]
  y1_pos_AN3=parse[parse.index("AN3")+3]
  z1_pos_AN3=parse[parse.index("AN3")+4]
  dist_AN3=parse[parse.index("AN3")+5]

  x1_AN0=float(x1_pos_AN0)
  y1_AN0=float(y1_pos_AN0)
```

```
z1_AN0=float(z1_pos_AN0)

x1_AN1=float(x1_pos_AN1)
y1_AN1=float(y1_pos_AN1)
z1_AN1=float(z1_pos_AN1)

x1_AN2=float(x1_pos_AN2)
y1_AN2=float(y1_pos_AN2)
z1_AN2=float(z1_pos_AN2)

x1_AN3=float(x1_pos_AN3)
y1_AN3=float(y1_pos_AN3)
z1_AN3=float(z1_pos_AN3)

b=float(dist_AN1)
c=float(dist_AN2)
a=float(dist_AN0)
d=float(dist_AN3)


y=func(x1_AN0,y1_AN0,z1_AN0,x1_AN1,y1_AN1,z1_AN1,x1_AN2,y1_AN2,z1_AN
2,x1_AN3,y1_AN3,z1_AN3,a,b,c,d)
```

DWM.write("\r".encode())

DWM.close()

It gives output as shown in fig.(dwm_out_1111111.txt)



You can save the output in a file using following command:

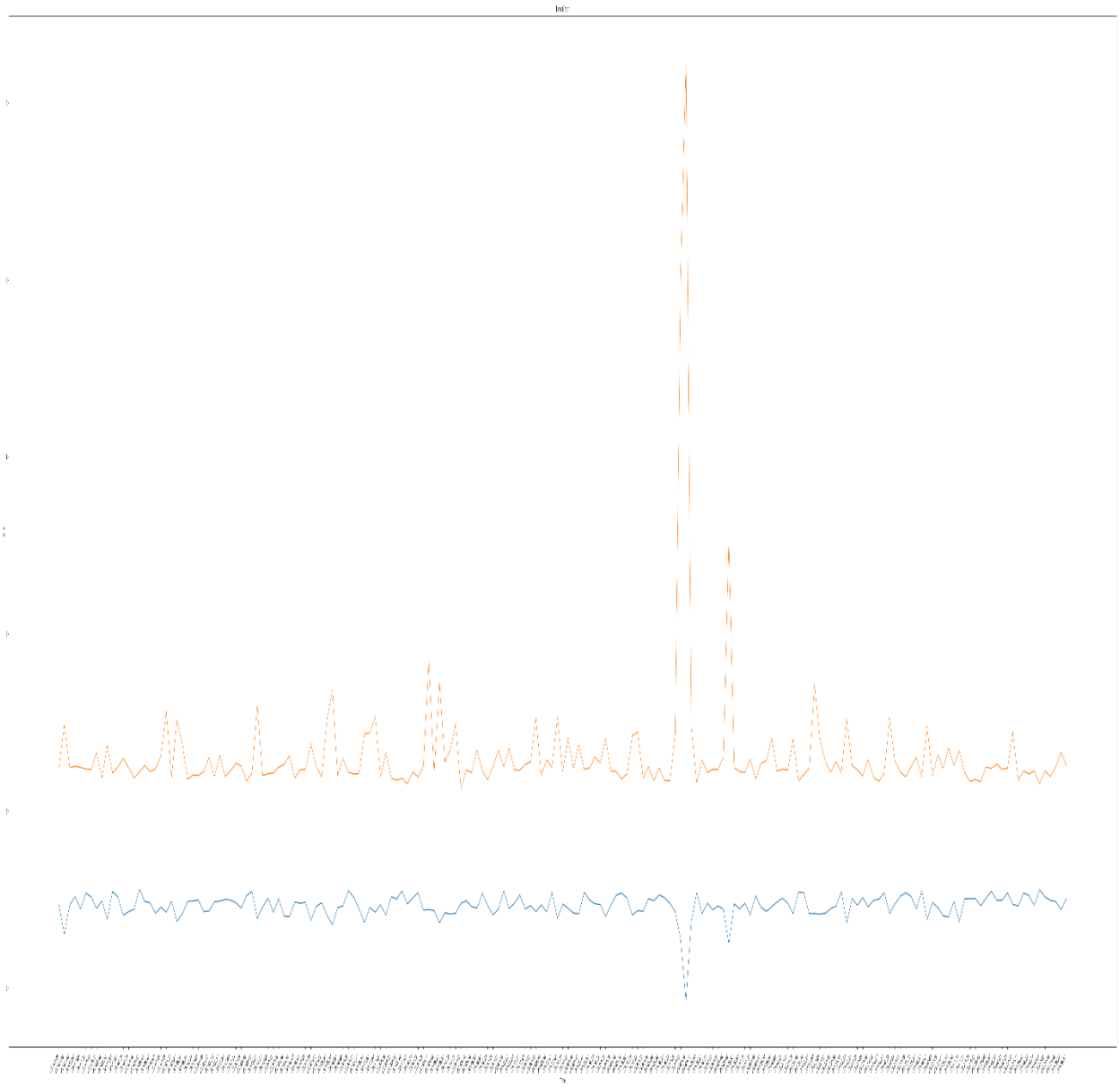
```
with open('example.txt',"a") as f:
    f.write("%f" % results.x[0])
    f.write("%s" % " ")
    f.write("%f\n" % results.x[1])
```

We will get different output depending on the anchors position and time at which we have perform the test.

Thus, we can analysis and visualize the data using Python Graphs,you can run the following program:

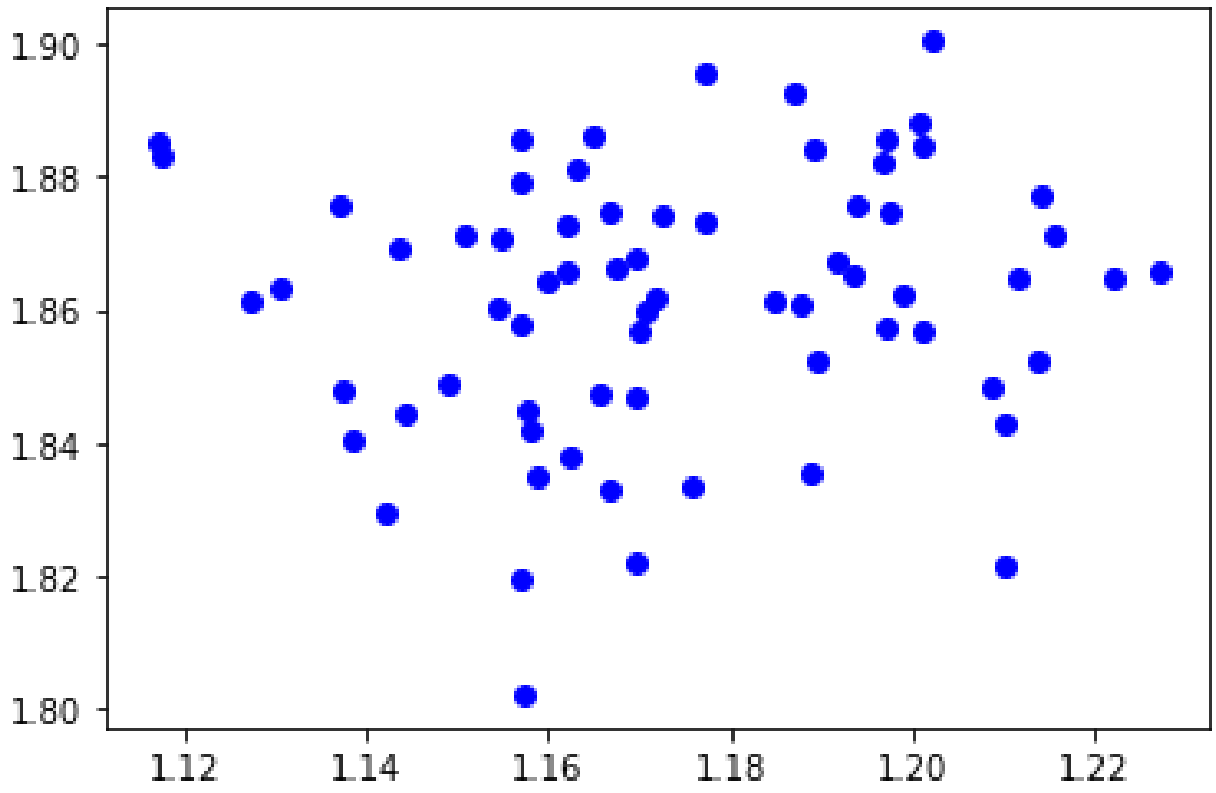
```
import math
import datetime
import matplotlib
import matplotlib.pyplot as plt
import pandas as pd #### import this library

df = pd.read_csv("d1_out1.txt", delimiter=',', encoding='latin-1')

t = df.iloc[:,0]
x = df.iloc[:,1]
y = df.iloc[:,2]


plt.xlabel('Time')

plt.ylabel('X Y')

plt.title('Time Graph')
plt.gcf().autofmt_xdate()
fig = plt.gcf()
fig.set_size_inches(100,30)
fig.set_dpi(100)
fig.savefig('d1_out1.png', dpi=100)
plt.plot(t,x,y)
plt.show()
```

one such graph Of X,Y with time I have shown below(d1_out1.png),you can view in PC for Details



It is large Data Graph, It has some Noises also.

Some more Graph of Position and Distances are below:



It is The 2-D Graph Of X-Y (X value at X-axis,Y value at Y-axis)

More Graphs can be Plotted with the Output Files Included in the Attached File

DOP:

As we know Dilution of Precision is the Quantity, which tells about the precision of our System, It depends on the Geometry Of the System. It can be calculated by Sum of Diagonals of Jacobin Matric formed while calculating the Position using Least squares Method. Less the no. more accurate the System is.

In my Case It was 4.92.

Conclusions and Future Work:

UAV Testing in Indoor:

Since, we are successful in getting Position, we can use this Position data to automate System like Drones, after interfacing it with the same. We can develop an Autonomous Drone which can move at a definite path, and further we can join more Drones to make a network or Drone Swarm. You can go through the following links for details:

- https://www.researchgate.net/publication/308813321_Design_of_an_UWB_indoor-positioning_system_for_UAV_navigation_in_GNSS-denied_environments
- https://robotics.stackexchange.com/questions/13883/quadcopter-controlled-by-raspberry-pi
- https://www.youtube.com/watch?v=6jWMuv6qaLs
- https://www.youtube.com/watch?v=kZW_eybFuF8
- https://www.instructables.com/id/The-Pi-Quadcopter/

We can also add Kalman-Filter and make it more reliable:

- https://dsp.stackexchange.com/questions/8860/kalman-filter-for-position-and-velocity-introducing-speed-estimates/8869
- https://github.com/rlabbe/Kalman-and-Bayesian-Filters-in-Python/blob/master/Appendix-G-Designing-Nonlinear-Kalman-Filters.ipynb

Local To Global Position:

Till Now, We have been Calculating Position with Local Co-ordinate System. But, we extend it to Global Position, only thing we would be needed is The position of any 3 Anchors in Global Positioning System.

https://gis.stackexchange.com/questions/84663/how-can-i-transform-real-time-local-position-system-coordinates-into-gps-wgs84

https://docs.unity3d.com/ScriptReference/Transform.InverseTransformPoint.html

https://www.researchgate.net/publication/27403081_Methods_to_convert_local_sampling_coordinates_into_geographic_information_SystemGlobal_positioning_systems_GISGPS-Compatible_coordinate_system

https://opencommons.uconn.edu/cgi/viewcontent.cgi?referer=https://www.google.com/&httpsredir=1&article=1005&context=nrme_articles

References:

- https://www.sciencedirect.com/science/article/pii/S2405959515300837
- https://pdfs.semanticscholar.org/5b65/ebca53ae85adf477a5bf17d3e6cf849fe62b.pdf
- http://www.diva-portal.org/smash/get/diva2:1251202/FULLTEXT01.pdf
- https://www.alanzucconi.com/2017/03/13/positioning-and-trilateration/
- https://ieeexplore.ieee.org/document/5355978
- https://github.com/lab11/polypoint
- https://decaforum.decawave.com/t/parsing-locations-from-dwm1001-dev-to-sql-database-using-python-on-raspberry-pi/4949
- https://www.raspberrypi.org/forums/viewtopic.php?f=63&t=173461
- https://decaforum.decawave.com/t/gateway-with-raspberry-pi-pin-header/4537
- https://decaforum.decawave.com/t/lot-of-interruption-and-breaks-in-rtls/1941/11

- https://github.com/etymologisk/Least-Squares-Trilateration/blob/master/Trilateration%20Using%20Limited%20Simulated%20Points%20With%20scipy.optimize.least_squares.ipynb

- https://github.com/lemmingapex/trilateration

- https://github.com/alduxvm/pyMultiWii

- https://github.com/zziz/kalman-filter

- https://arxiv.org/ftp/arxiv/papers/1204/1204.0375.pdf

- https://stackoverflow.com/questions/53814004/plotting-time-and-float-value-using-python-matplotlib-from-file
- https://www.youtube.com/watch?v=g5VVjEopxfw&list=LLis9oA9Ir8xmgCOWeEJRVhQ&index=19&t=0s