École supérieure d'ingénieurs en génie électrique

Development of test tool for Automotive Radars

Submitted in fulfilment for the requirements of awarding the Master's in science and Technology Degree in Electronics Embedded Systems

Joshiga santh SIGAMANI

16th March 2020 – 28th August 2020

ZF Autocruise
Brest, France 29200.

CONTENTS

LIST OF FIGURES:

LIST OF TABLES:

GLOSSAIRE:

ACC	: Adaptive Cruise Control
ADAS	: Advanced Driver Assistance System
RCS	: Radar Cross section
ADC	: Analogue to digital converter
MMIC	: Monolithic microwave integrated circuit
TX	: Transmission
Rx	: Reception
FFT	: Fast Fourier Transform
MR	: Medium range radar
FR	: Full range radar
OEM	: Original equipment manufacture
CAN	: Control area network
MIMO	: Multiple input, multiple output
IDE	: Integrated development environment
RTS	: Radar target simulator
UDP	: User datagram protocol

1.INTRODUCTION AND OBJECTIVE:

In today's automotive technologies, RADAR's are becoming an unavoidable sensor. Vehicles which have adaptive cruise control and autonomous driving assistant systems (ADAS) would not reach high success rate without radars. Radar is considered as an essential part when it comes to road safety due to the detailed image of surrounding.

Applications of Automotive Radar systems are *Adaptive cruise control, Automated Emergency Break, crossing Traffic alert, blind spot detection and corner detection, Parking assist etc.*

Automotive Radars are usually classified into three categories:

- *Short Range Radar (80m Range detection),*
- *Medium Range Radar (180m Range detection),*
- *Full range radar (300m Range detection)*

Main objective of this internship is to understand radar functionalities and develop a testing tool to plot live data which will be available in specified industrial standard UDP Protocol.

Followed by Introduction of ZF Group and ZF Autocruise, this report will give detailed explanation of different types of radars and its working.

2.INTRODUCTION ABOUT ORGANIZATION

2.1. About ZF groups:

This Company was founded by Luftschiffbau Zeppelin GmbH in 1915 at Zepernick, Germany. Primarily they started by producing gears for Zeppelins and other airships. Founder Ferdinand von Zeppelin changed company name to '*Zahnradfabrik*' called "Gear Factory" in English. And its abbreviated to ZF.

In 1919, ZF stepped into automobile market. Moving to Friedrichshafen a city in South Germany which is near to the border of Switzerland and Austria. Eventually, ZF became ZF Friedrichshafen.

ZF is known for research, design and supplies systems for automotive industries. Domains like Vehicle Motion control, Integrated safety, Autonomous driving and Electric Mobility, ZF offers comprehensive solutions to established vehicle manufacturers, newly emerging transports and mobility service providers. ZF had not stopped from keeping his leg in Marine, avionics, defence, Rail and general industrial applications. Census taken on 29th May 2020 in ZF Groups says 160,000 employees working around the world with approximately 260 locations in 41 countries.

2.2. Products of ZF:

2.2.1. Automotive Products:

Active safety : Anti-lock Braking System, Electric Park Break, Electronic Stability Control, Front Calliper, Integrated Break Control, Rear Calliper and Integrated Park, Single and Tandem.

Passive safety : Airbag Systems, Seat Belt Systems, Steering Wheel Systems

Driveline : 1 – Stage Power Take-off Unit, 2-Stage Power Take-off Unit,4th generation of 8 speed Automatic Transmission, 7-speed Dual Clutch Transmission, 7-speed Manual Transmission, 8-speed Automatic Transmisson,8 speed Dual Clutch Transmission, 8 speed Hybrid Transmission, 8-speed Plug in hybrid Transmission, 8 speed Automatic Transmission, Automated Manual Transmission, Clutch Disc with Torsional Damper, Clutch by Wire, Converter Hybrid Transmission, Driven Solid Beam Axle, Dual Mass Flywheel, Dual wet Clutch, Electric Axle Drive, Engine Bound Front Axle Drive, Frame Bound Front Axle Drive, HAG Rear Axle Drive, HAG Rear Axle Drive Standard , HAG Rear Axle Drive EM, Hybrid Clutch, Hybrid Module, Hydrodynamically Cooled Clutch HCC, Motor-Pump units for transmissions, Rear axle drive with integrated all-wheel drive Clutch, Torque Converter, VECTOR DRIVE, Tend.

Electronics : Adaptive Cruise Control, Automatic Emergency Breaking, Electronic Control Units, Emergency Steering Control, Forward Facing Monocular Cameras, Forward Looking Full

Range Radar, Forward Looking Medium Range Radar, Gearshift Systems, Highway Driving Assist, Lane Keeping Assist, Power Electronics, Safety Domain ECU (Generation 2) , Safety domain ECU (SDE), Sensor Solutions, Switch/Component, Traffic Jam assist, Tri Cam Front Camera

Steering : Dual Pinion Drive, Electrically Powered Steering Belt Drive, Electrically Powered Steering Column Drive, Mechanical Steering Gear.

Chassis : Active Kinematics control, Axle concept Easy Turn, Axle Concept starts, Axle Systems, CDC – Continuous Damping Control, CDC1XL, Control arm, Corner Modules, cross axis joints, Damping Modules, ERC, Knuckles and hubs, Modular Valve System Predictive Damping, Sensitive Damping Control, motion Active Suspension Concept, Stabilizer Link, Twin tube Damper.

2.2.2. Non-Automotive products:

Construction machinery, Agriculture Machinery, Marine, Wind Power, Test Systems, Rail Vehicles, Special Driveline Solutions, Material Handling, Aviation Technology, Industrial Gearboxes, Machinery & plants.

2.3. Sales of ZF Groups on 2017 in millions:

Car Powertrain Technology	€ 8,725
Car Chases Technology	€6,484
Commercial Vehicle Technology	€ 3,172
Industrial Technology	€ 2,530
E-Mobility	€ 924
Active and Passive Safety Technology	€ 13,970
ZF After market	€ 3,007
Corporate R&D, Corporate Headquarters and Service Companies	€ 468
Consolidation	*€ -2,836*
ZF Group	**€ 36,444**

Table 1 Sales of ZF Group

2.4. Autocruise Presentation:

In 1998, Autocruise was found as a joint venture of Thales and the American automotive supplier TRW. Autocruise is located at the city of Plouzané near city of Brest, France. Their specializations are the development of collision warning radars and distance control radars for driver assistant. Autocruise got inspired by Thales expertise and knowledge in avionics technology particularly in military radars and developed advance driver assistant system which works on radio signals.

The 1st generation Radar AC10 was developed in 2002. In 2003, Autocruise became 100% a TRW company, after TRW acquired Thales stakes. Then Autocruise released different radars by updating and upgrading previous versions.

The German automotive Supplier ZF Friedrichshafen officially bought the American TRW in 2015. Thus, TRW become one of ZF groups with autocruise.

The expansion of autocruise has enlarged its premises and now it has two plants. The first building, Plant 1 is the production site, it brings together the logistics, production quality management, human resources and finance teams.
The second building, Tech 2 was opened in 2018 close to the head office, which brings together the Research & Development and Testing teams. Spacious working area and good exposure provides comfortable work environment. In 20 years, the population of autocruise has been risen from 30 to 140 staff.

2.4.1. Products of Autocruise:

Model	Year	Purpose
AC10	2002	• Adaptive cruise control
AC20	2005	• Adaptive Cruise Control • Forward Collision Warning
AC100	2012	• Adaptive cruise control • Forward Collision warning • Automatic Emergency Breaking
AC1000	2017	• AC100 Functionalities • Following Distance Indicator • Automatic Emergency Breaking (EuroNCAP Pedestrian & Urban)
AC2000	2019	• AC1000 Functionalities • HIGH Speed ACC • EURO NCAP Bicycle • Corner Radar
Imaging Radar	2021	• 3D • AD level 3

Table 2 Products of Autocruise

2.4.2. AC100 – 24GHz Radar 3rd generation:

Released year	: 2002
Operating frequency	: 24.150 – 24.250 GHz
Detection Range	: 2 to 150 m
Field of View	: 12°/8° (Azimuth, close and medium Range)
	3° (Elevation)
Angular Accuracy	: 0.5°
Operating temperature	: -40°c to 85°
Object Discrimination	: 2.5m or 1.4kmph
Velocity Operating Range	: up to 250kmph
Update Rate	: 40ms

Figure 1 AC100

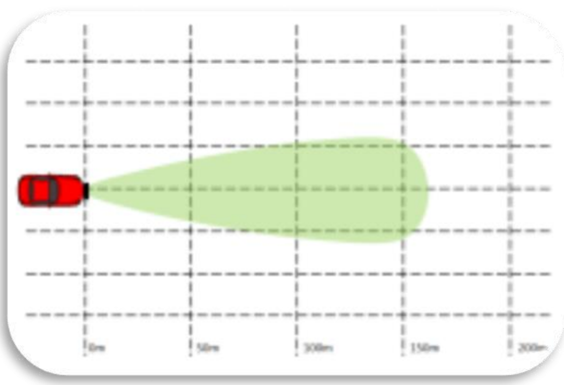


Figure 2 AC100 Coverage

2.4.3. AC1000 – 77GHz Radar – 4th Generation

Released year	: 2002
Operating frequency	: 76.0 - 77 GHz
Detection Range	: 1 to 180 m for cars
	: 1 to 70m for pedestrians
Field of View	: Azimuth: Low speed mode >70° & High speed mode >20°
Angular Accuracy	: <0.25° in centre area (+/-9°) <1° in outer area (+/-30°)
Object Discrimination	: 1m in low speed mode, 2m in high speed mode or 1.4kmph
Velocity Operating Range	: up to 250kmph
Update Rate	: 50ms

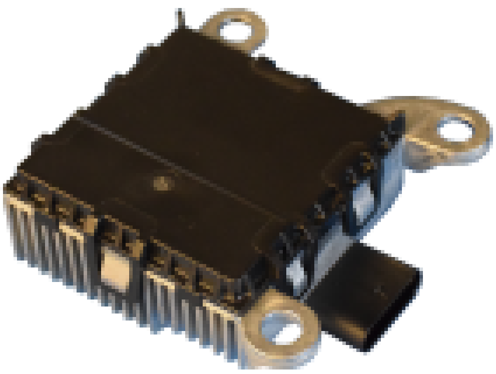


Figure 3 AC1000

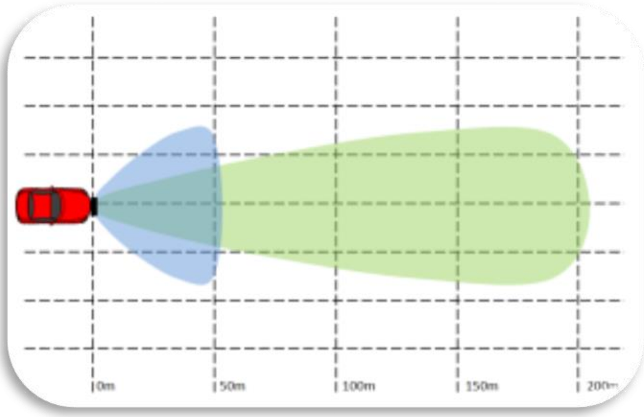


Figure 4 AC1000 Coverage

3. MY TEAM:

- I worked under Test team which composed of 5 members including team head.
- *Julien & Tristan* working on Full Range Radar and *Thierry & Steffen* working on Medium Range Radar

3.1. Team chart:

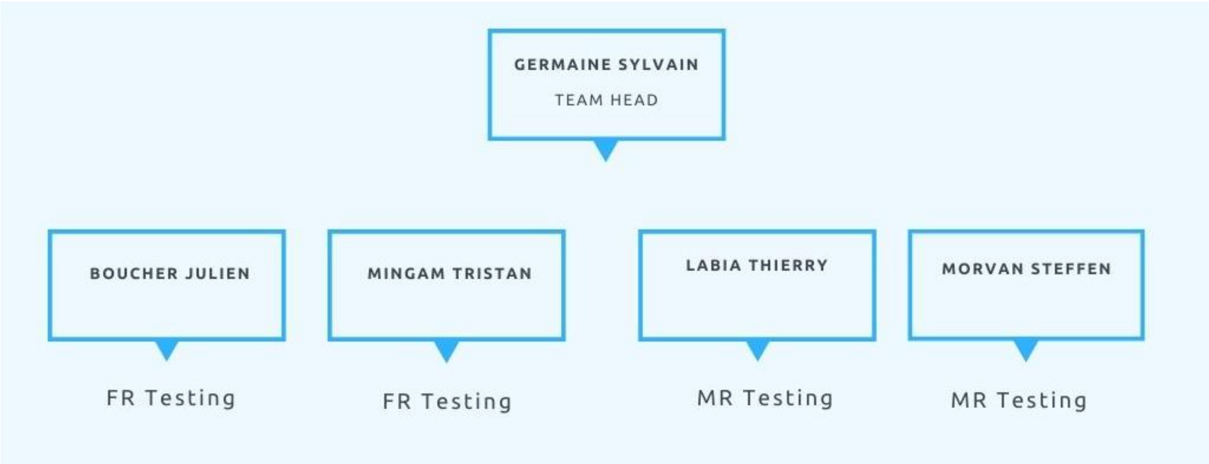


Figure 5 Team Hierarchy

Team comprised of 5 members.

- Two for Medium range radar, Two for Full range radar and one manager.
- Myself is an intern in software development for testing team.
- Apart from ty team, there are several other teams working on, hardware design, RF design, Signal processing team and management team.

4. ABOUT INTERNSHIP

4.1. Introduction to internship:

Test engineers apply their analytical skills to the field of Design engineering's. They produce test plans that can be used on products, such as electronic components or systems. They help evaluate these products to make sure that they are ready for the market.

Once test engineers have developed a testing plan for a specific system or component, they perform the required tests and then use the test results to assess the product and produce a report outlining their findings and recommendations.

Important aspect of test engineers in automotive industry is to test and validate every electronical and mechanical parts (included embedded SW such as signal processing) in real time environment. Checking whether the products follow industrial standard rules and conditions. That consumes lot of time for analysing.

In autocruise, the test department has several testing software developed by previous software engineers, which are used for testing and analysing the sensors. But analysing is currently done by post-processing recorded data.

My mission during this internship was to develop a graphical tool which integrated both testing and analyzation.

The objective of this internship is to Develop a Human machine interface, which should plot live data from respective Radar and save it for further use. Saved data can be used for further analyzation

4.2. Technical Details:

4.2.1. What is radar?

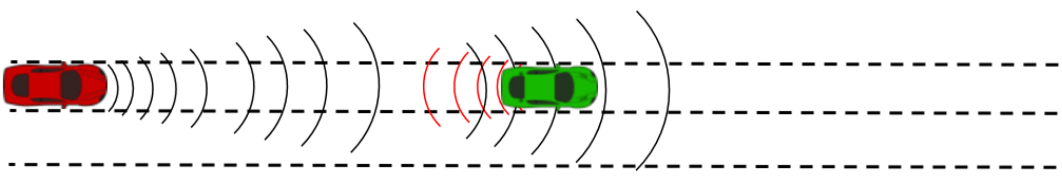


Figure 6 RADAR WORKING

- Radar transmit the radio wave and receives back
- Parameters which are measured by radar for every target:
  - ✓ Range
  - ✓ Speed
  - ✓ Angles (in Azimuth and Elevation)
  - ✓ RCS (radar cross section)

4.2.2. Typical Block diagram of Radar:

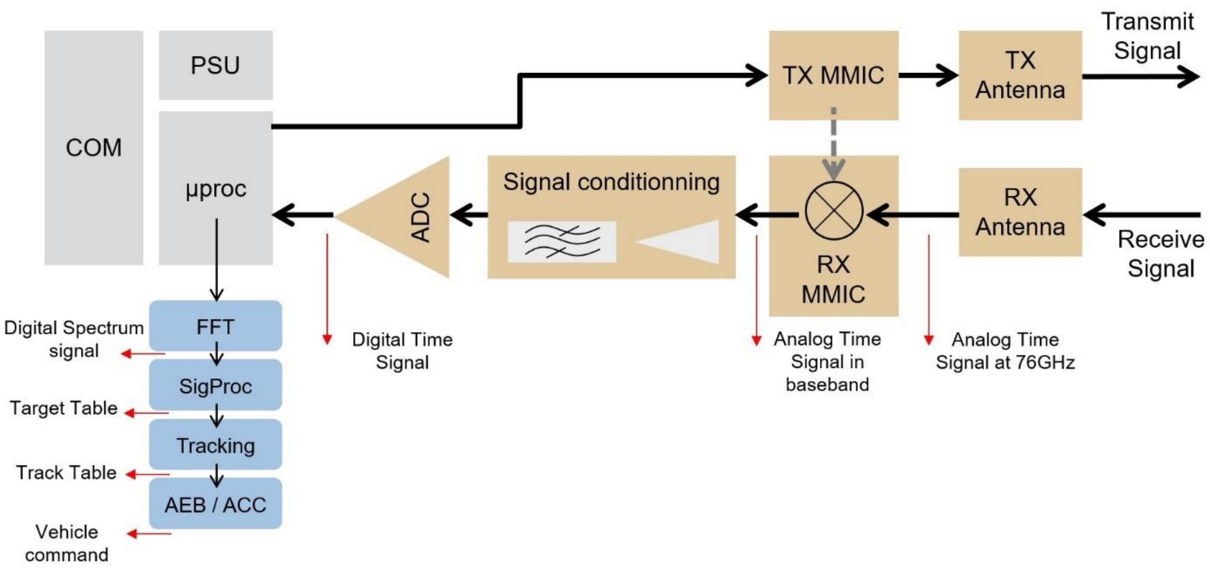


Figure 7 BLOCK DIAGRAM OF RADAR

Above diagram explains the working flow of radar.

4.2.3. Classification of Radars:

- Short Range Radar (NOTE: this is not developed by ZF Autocruise)
- Medium Range Radar
- Full Range radar

4.2.3.1. Short Range Radar:

This radar is not developed by ZF autocruise. It has the working frequency of 77GHz. Provides features like Object List, Blind Spot Warning, Lane Change Assist, Rear Cross Traffic Alert (with Braking), Front Cross Traffic Alert , Rear Pre Crash Sensing etc.

4.2.3.2. Medium Range Radar:

Current Medium Range Radar developed by ZF Autocruise is called MRGen21 and has working frequency of 77GHz. Provides features like adaptive cruise control includes emergency stopping. It is also capable of detecting jamming or blind status, misalignment and so on.

4.2.3.3. Full Range Radar:

Current Full Range Radar developed by ZF Autocruise is called FRGen21 and operates at the same frequency as MR. Features available in MR are also available in FR.

MR radars are usually cheap and small radars used for mass market to offer ACC and AEB functions and answer the EuroNcap safety requirements. On the other hand, Automotive industries prefer FR for autonomous driving, because FR can provide high resolution 4-d data. This enables detection of static environment processing, which are free space detection and road topology estimation.

4.3. Currently working Modules in Autocruise:

4.3.1. MRGen21:

- ✓ MRGen21 radar is 77Ghz forward facing automotive radar.
- ✓ Complies with cyber security standards and OEM specific requirement
- ✓ Operating temperature range from -40°C to +80°C
- ✓ AUTOSAR 4.2.2
- ✓ Datalogger for Development purpose
- ✓ OEM specific communication and diagnostic protocols

4.3.1.1. Radar Signal Processing Architecture:

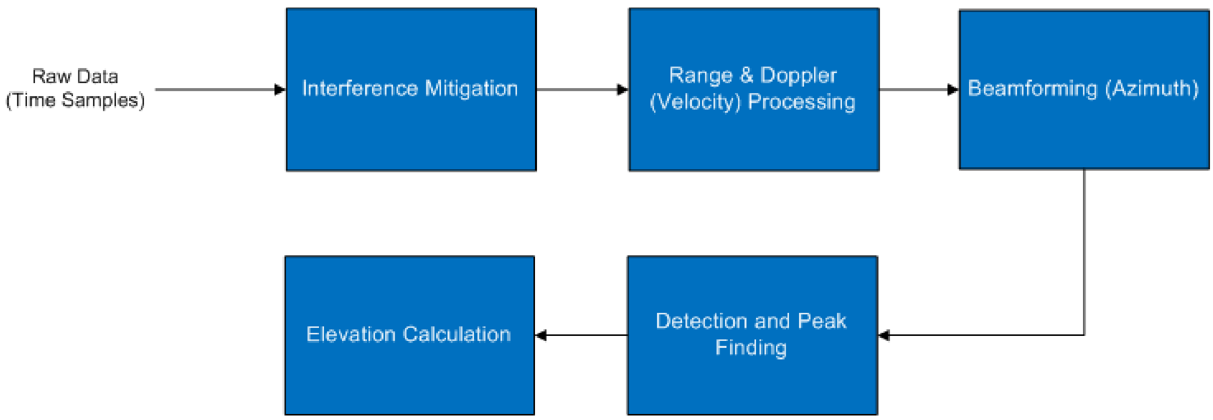


Figure 8 MR signal processing architecture

The processing sequence comprises *Fourier transformation, MIMO demodulation, beamforming, detection, and peak finding.*

4.3.1.2. Modulation:

Main radar modulation parameters:

Bandwidth Min	150MHz (eff) => 1 m
Bandwidth Max	375MHz (eff) => 0.4 m
Cycle time	60 ms
RF Frequency	77 GHz
Number of Rx channels	4
Number of Tx	3
Sampling rate	5 Msamples/s IQ
Number of chirps	256
Number of samples per chirp	256 IQ
Chirp repetition time	70 – 90 us
Speed ambiguity (+/-)	51.5 km/h
Speed resolution	0.4 km/h (0.4 km/h with HR)

Table 3 Radar Modulation parameters

The proposed parameters contain 3 variants for the Bandwidth, i.e., range resolution of 40, 70, or 100 cm. Depends on the host vehicle speed, radar gives more resolution at lower speed and maximum detection range at high speed.

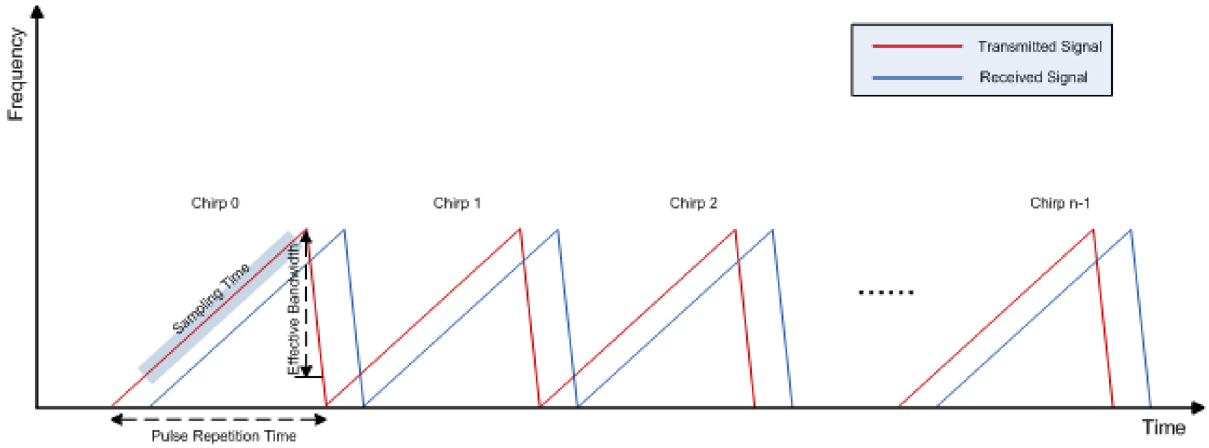


Figure 9 MR Single Chirp Timing

Modulation scheme based on a coherent train of linearly frequency modulated pulses. which is also known as fast chirp-sequence modulation as shown in Figure 9.

4.3.1.3. Range mode definition:

- There is just one single antenna, but the range resolution is dynamically adjusted to the ego speed
  - Low Speed: < 65 km/h
  - Mid. Speed: >65 km/h and < 115 km/h
  - High Speed: >115 km/h

4.3.1.4. Product Design:

Below table explains all variant of MRGen21

Vehicle Type	HW Platform	System functionality	Bus Assumptions	Description
Passenges Car	RRU2 small	Remote2	Private CAN-FD to Camera / SDE (2Mb/s) Or Automotive Ethernet 100Base-T1 Vehicle CAN HS (1Mb/s)	Fusion and functions on external ECU, but not on radar MCU; for passenger car Object output to external ECU via CAN (with Tracking, no functions)
		Standalone	Vehicle CAN HS (1Mb/s)	Radar only; for passenger car
Commercia l Vehicle (Truck)	RRU2 big	Remote2	Private CAN-FD to Camera / SDE (2Mb/s) Or Automotive Ethernet 100Base-T1 Vehicle CAN HS (1Mb/s)	Fusion and functions on external ECU, but not on radar MCU; for commercial vehicle Object output to external ECU via CAN (with Tracking, no functions)
		Standalone	Vehicle CAN HS (1Mb/s)	Radar only; for commercial vehicle

Table 4 Variants of MRGen21

4.3.1.5. Detection capability:

- At no Misalignment

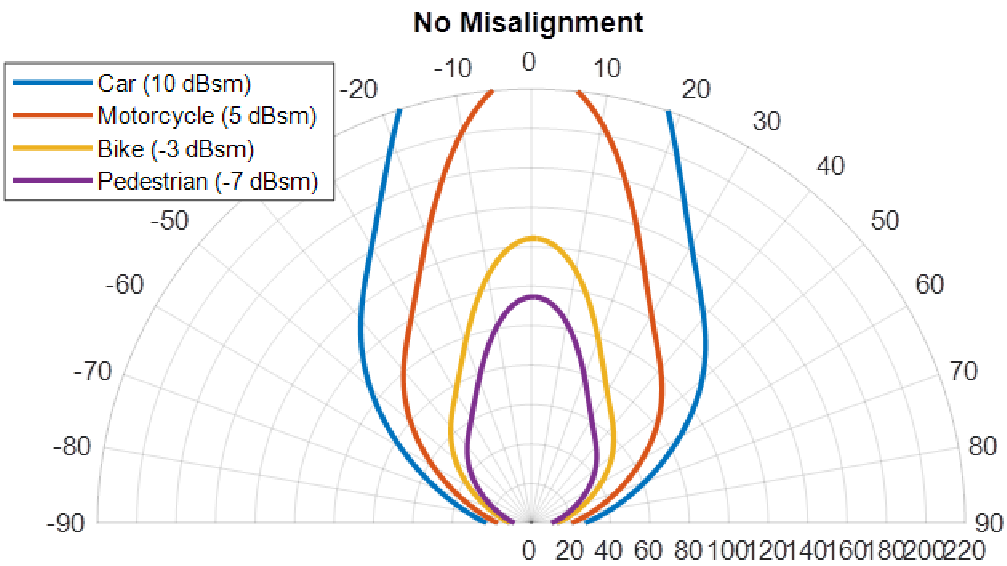


Figure 10Detection capability at zero misalignment

- At 6° Misalignment

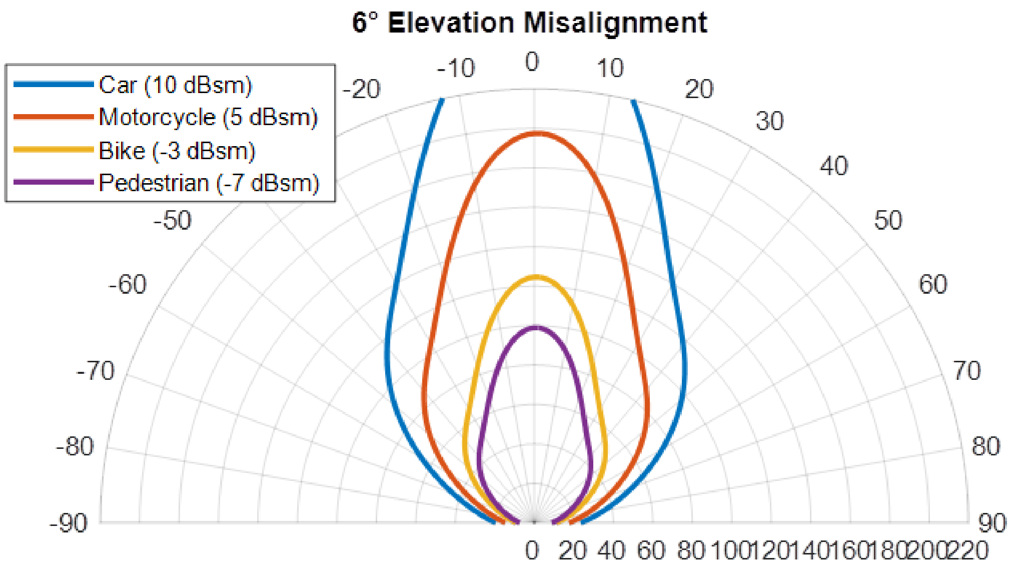


Figure 11 Detection capability at 6° misalignment

The above image explains, how detection rate reduced because of misalignment

4.3.2. FRgen21:

- ✓ FRgen21 is 76GHz forward facing automotive radar
- ✓ It is high resolution 4-D radar that is apt for autonomous driving
- ✓ Static environment processing (free space detection, road topology estimation, …)
- ✓ Operating temperature range from -40°C to +85°C
- ✓ AUTOSAR 4.2.2 operating system
- ✓ Data logger for development purposes
- ✓ OEM specific communication & diagnostics protocols
- ✓ Ethernet interface (Typically for ADAS Fusion)

4.3.2.1. Radar Architecture:

The processing sequence comprises Fourier transformation, MIMO demodulation, beamforming, detection, and peak finding. An overview is given below.

Figure 12 signal processing chain

4.3.2.2. Modulation:

We use a modulation scheme based on a coherent train of linearly frequency modulated pulses, also known as fast chirp-sequence modulation, which is illustrated in Figure below.

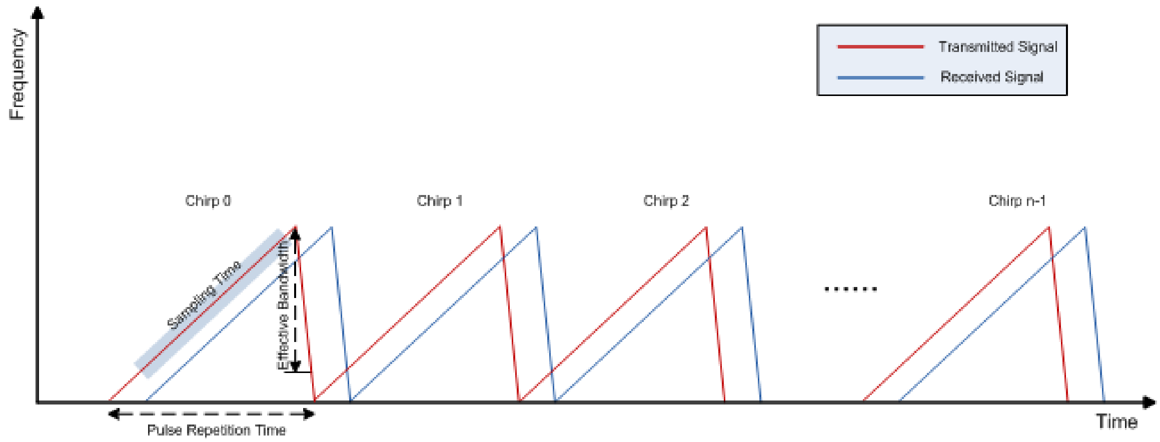


Figure 13 Chirp timing

4.3.2.3. Detection performance:

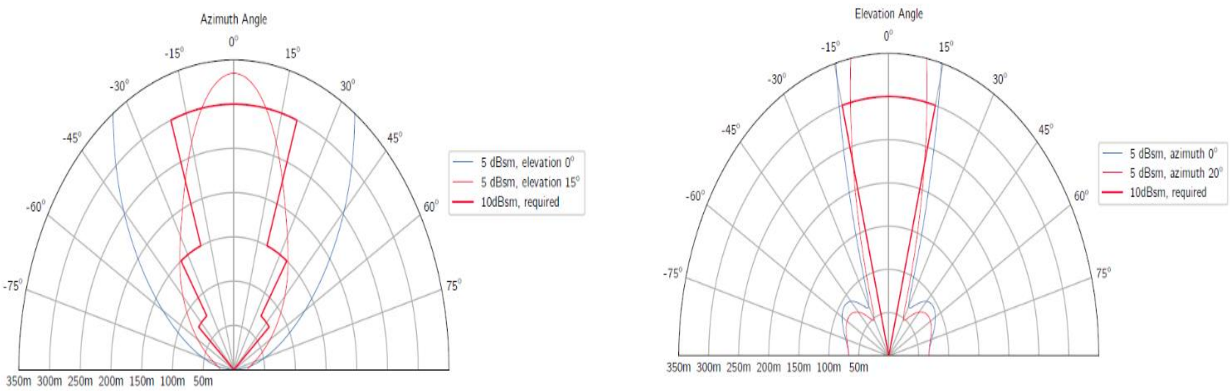


Figure 14 Detection capability of car (10dBsm)

The FRGen21 radar sensor features three modes of operation - long range, mid-range and short range. These range modes are defined as:

- Short range: < 15km/h
- Mid-range: > 15km/h and < 65km/h
- Long range: > 65km/h

There is just one single antenna, but the range resolution is dynamically adjusted to the own speed

4.3.2.4. Detection Capability:

The following figures depict the detection capability of the sensor for different target types:
Car (10 dBm²)
Motorcycle (5 dBm²)

Pedestrian (-7 dBm²)

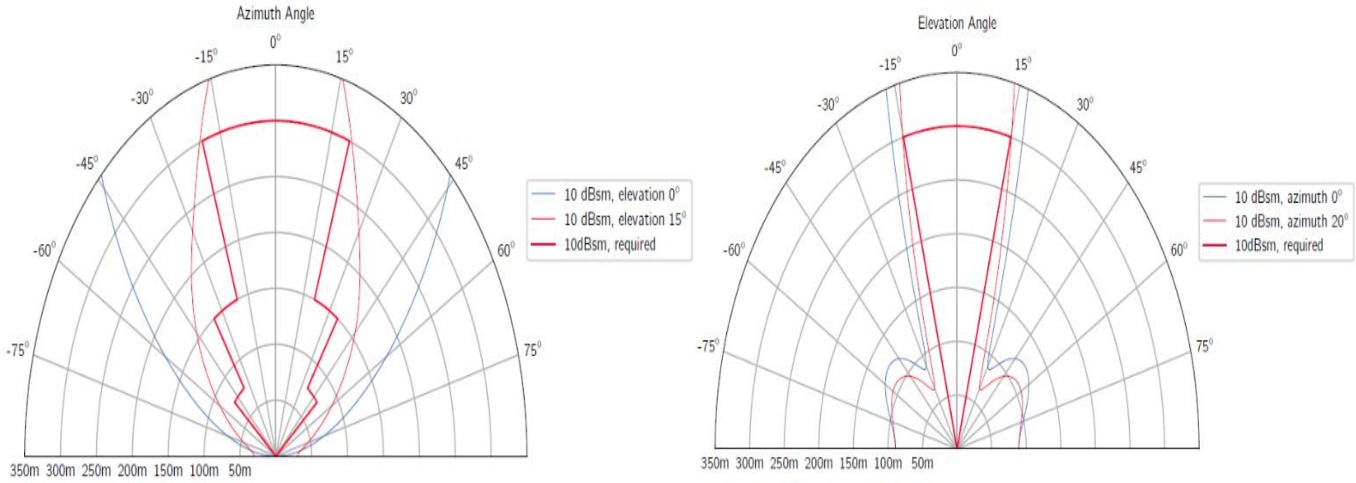


Figure 15 Detection capability of motorcycle (5dBsm)

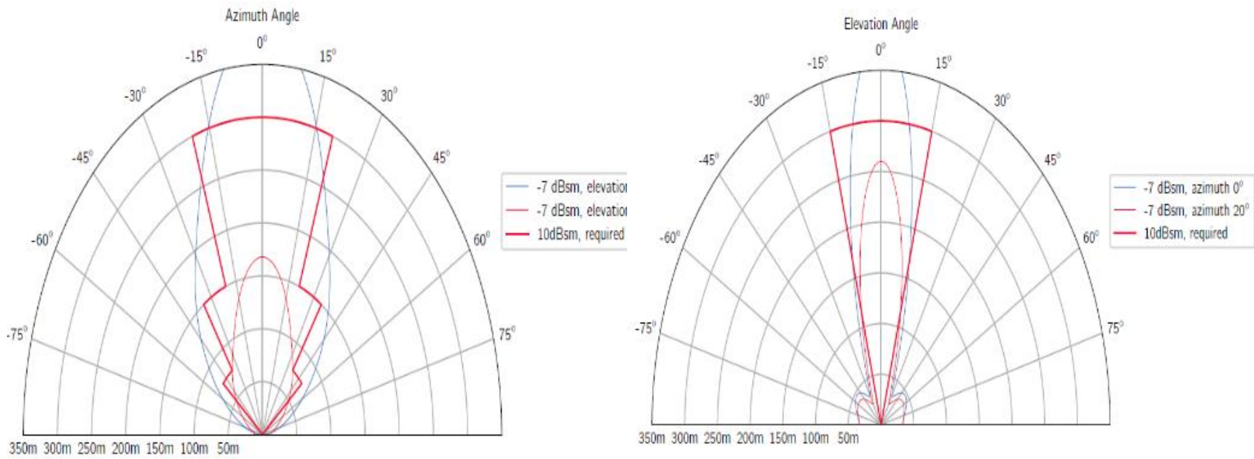


Figure 16 Detection capability of pedestrian (-7dBsm)

You can find lot of similarities between FR and MR. The major differences are detection level, and accuracy.

Detection level and accuracy are higher in FR compare to MR. These are achieved by adding more antennas compare to MR.

4.4. Mounting of MR and FR

The radar sensor includes mounting features for attachment to the vehicle.

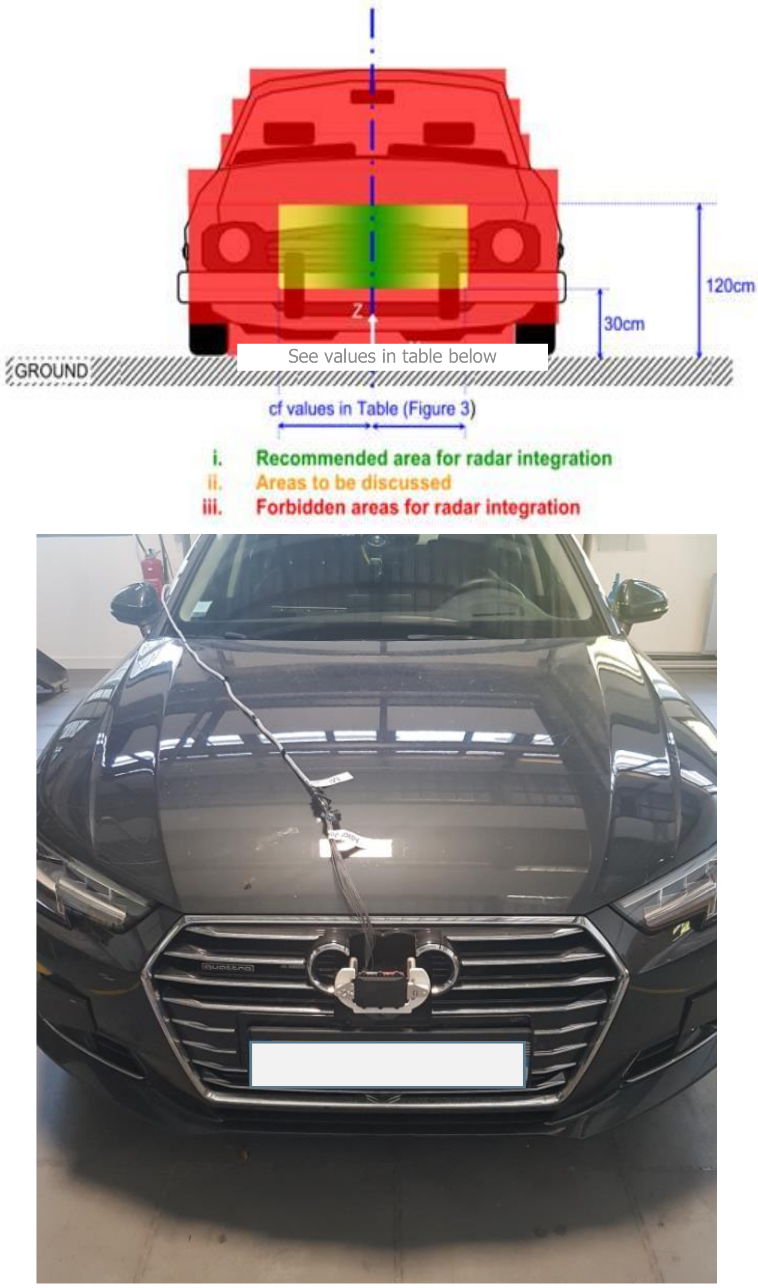


Figure 17 Positional limits for radar sensor mounting

4.5. Hardware Used:

- FRgen21 Radar
- MRgen21 Radar
- Radmon (BroadR-Reach® - automotive ethernet – to 10/100 Ethernet (100Base-TX)
- FRgen21 connecting cables
- MRgen21 Connecting cables
- Radmon connecting cables

4.5.1. FRgen21 :

Technical details of this sensor are explained at 4.3.2

Photos of radar which I used for testing purpose are shown below:

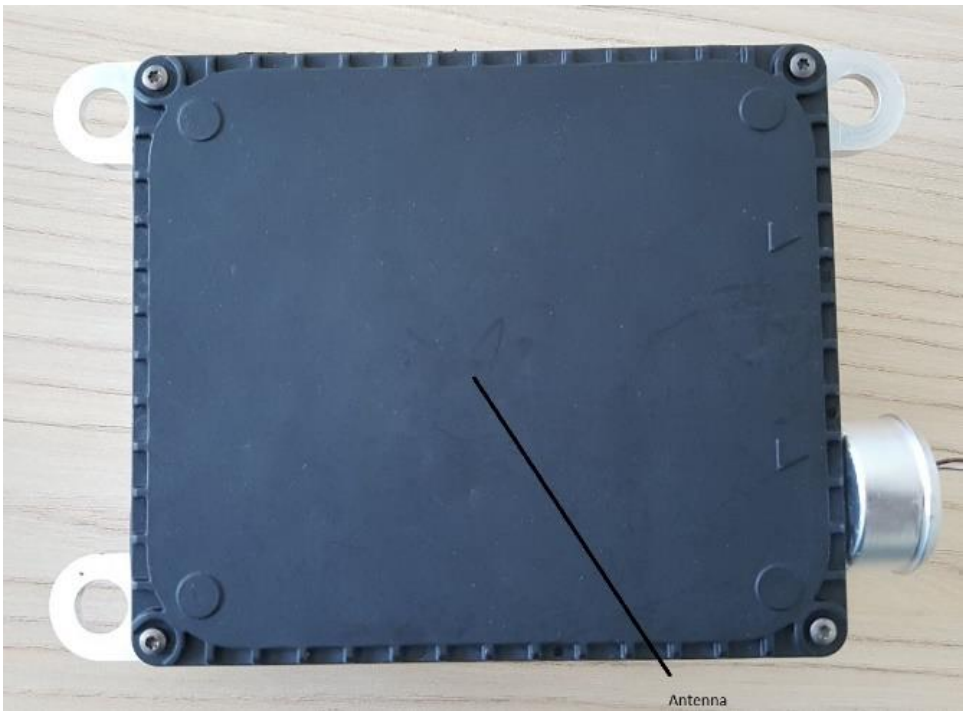


Figure 18 : FRGen21 Front View

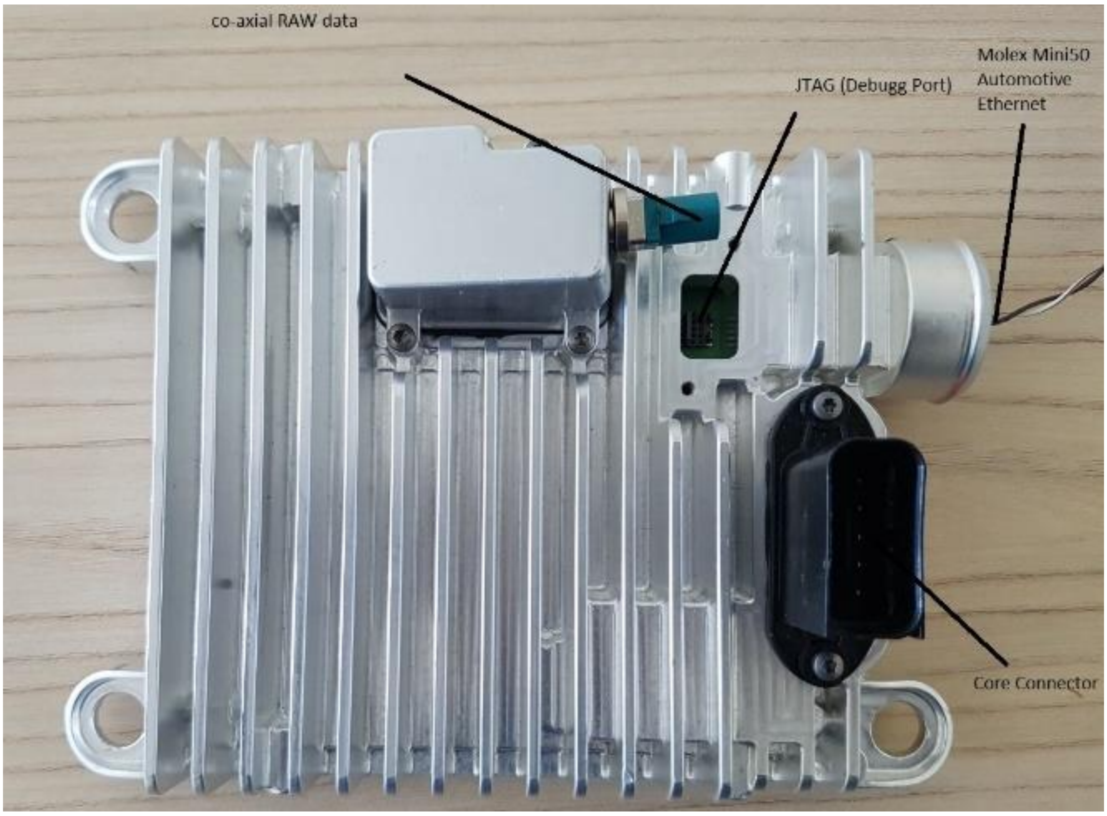


Figure 19 FRGen21 Back View

Antenna:

- Antenna is arranged at front side of the radar
- Number of reception antennas (16) is higher than that number of transmission antennas (12).
- Compare to MRgen, number of transmitter and receiver antennas is 4 times higher.

Co-axial Raw data:

- This port is used for co-axial connection.
- Pure Raw data (before applying signal processing) can be retrieved from this port.

JTAG:

- It is an industrial standard port.
- This pin is used for debugging purpose and/or software flashing

Molex Mini50:

- This is an automotive industrial standard ethernet port
- In future CAN protocol will most likely be replaced by Ethernet.

Core Connector:

- Only used for powering up the Radar.

4.5.2. MRGen21:

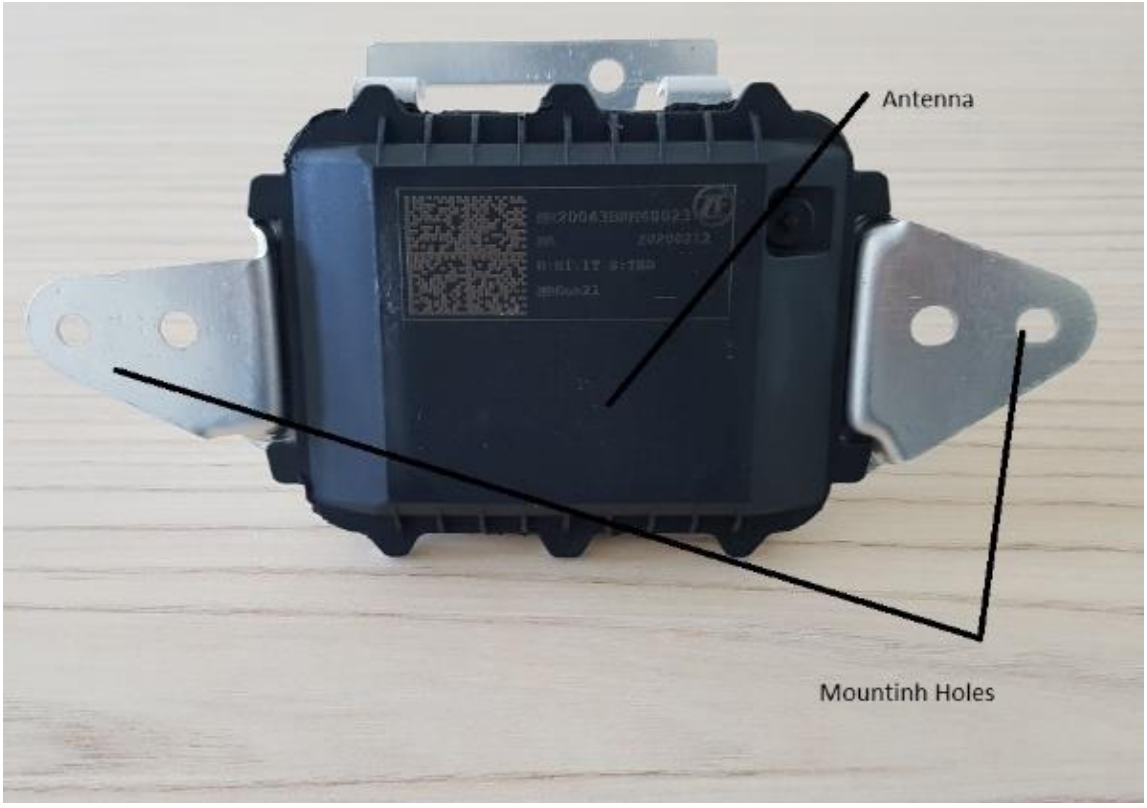


Figure 20 MRGen21 Front View

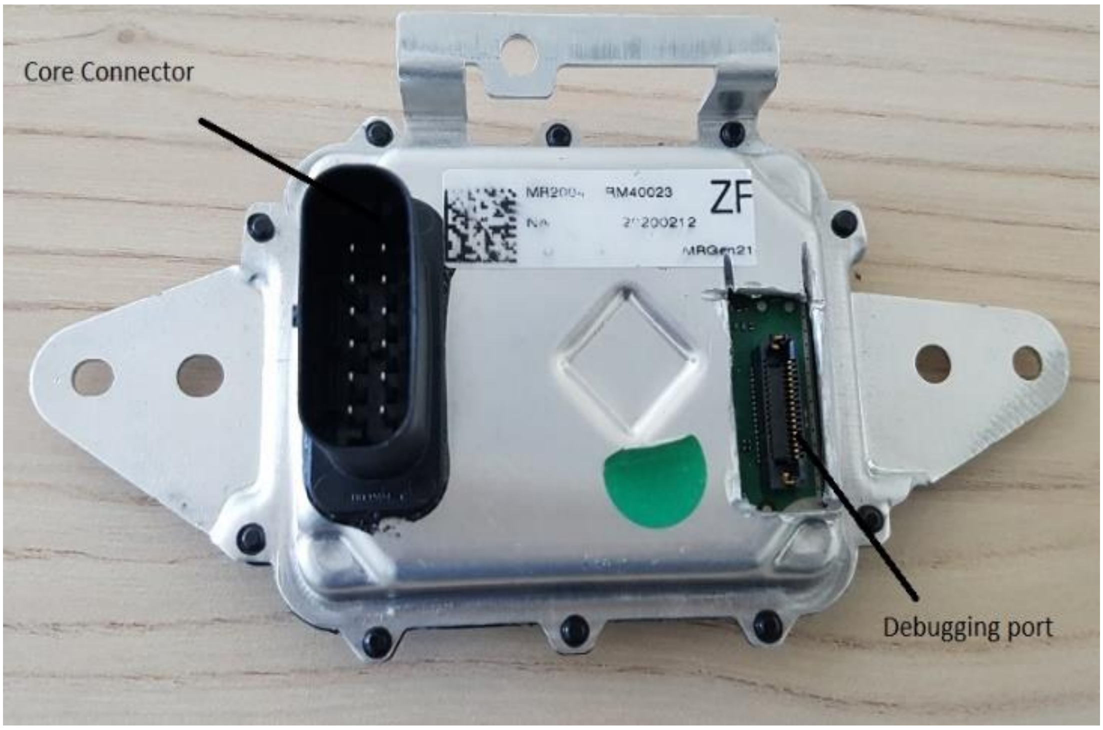


Figure 21 MRGen21 Back View

Antenna:

- Antenna are arranged at front side of the radar
- It has 3 transmitter and 4 receiving antennas. Reason for having 4 receivers are to get higher angular accuracy.

Debugging port:

- This port is used for debugging purpose.
- Only for Industry use

Core Connecter:

It is a 12-pin connected for power supply and data communication to other devices (test PC, vehicle, camera, etc).

4.5.3. Core Pin Connector:

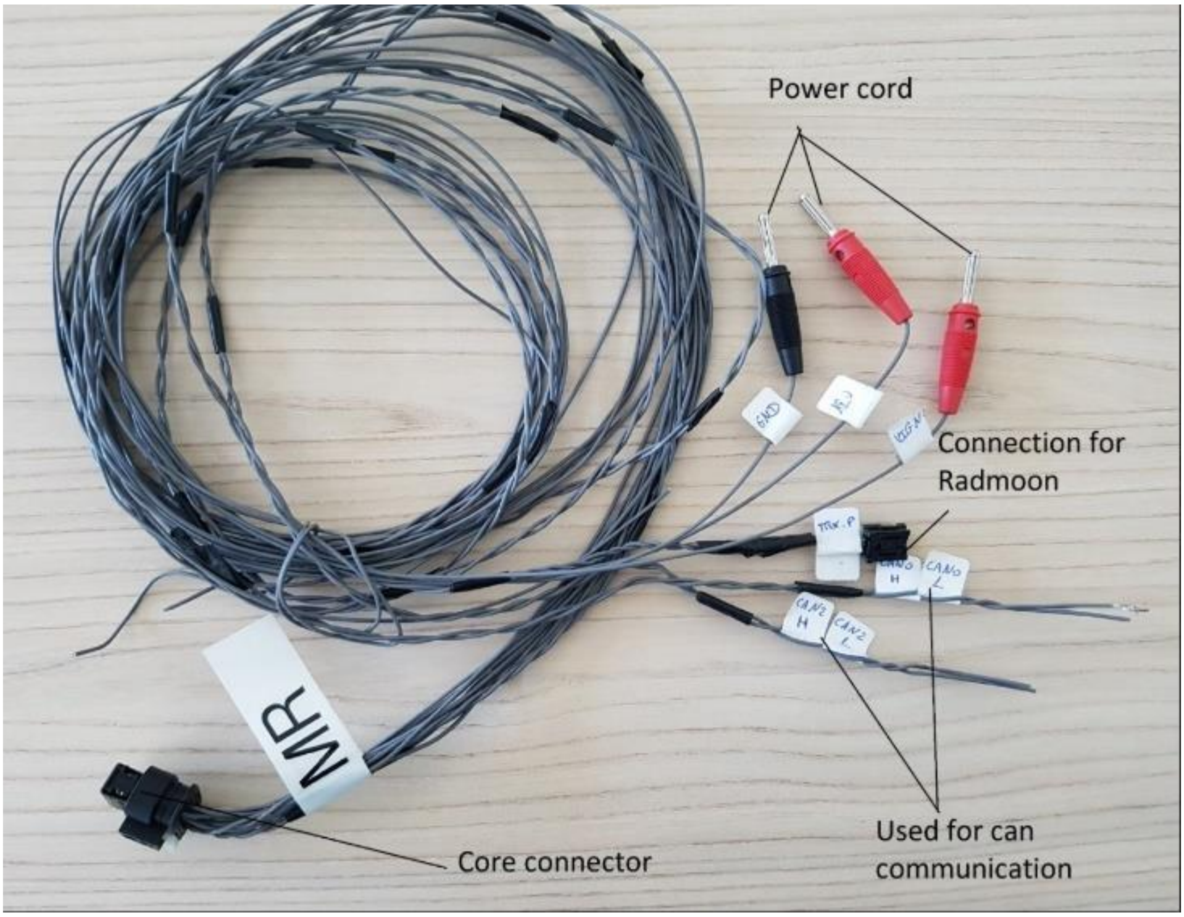


Figure 22 MRGen21 Connecting Wire

Power cord:

- Used for powering up Radar.
- Radar starts working at 12v.

CAN bus:

CAN bus is a vehicle protocol which allow to communicate within all the slave controllers which are connected same bus without help of master.

MRGen reads data (vehicle speed) from other sensors or controlling units to change working mode.

Connection for Radmoon:

- Port available in this module names as *Molex Mini50*. Same port can be found in FR.
- An industrial standard ethernet protocol connector
- Use to simulate radars in testbench.

4.5.4. Radmoon:

Figure 23 Radmoon

Media converter which allow to connect BroadR-Reach® / 100BASE-T1 Automotive Ethernet device to a PC with a conventional 4-wire 10/100 Ethernet port.

4.6. Current tool:

4.6.1. Currently using software for Testing purpose:

Name of the software: Python can

- Advantage
  - Analyse both MR and FR
  - Compactable for multiple radars
  - Selective filtering
- Disadvantage
  - live streaming is not possible

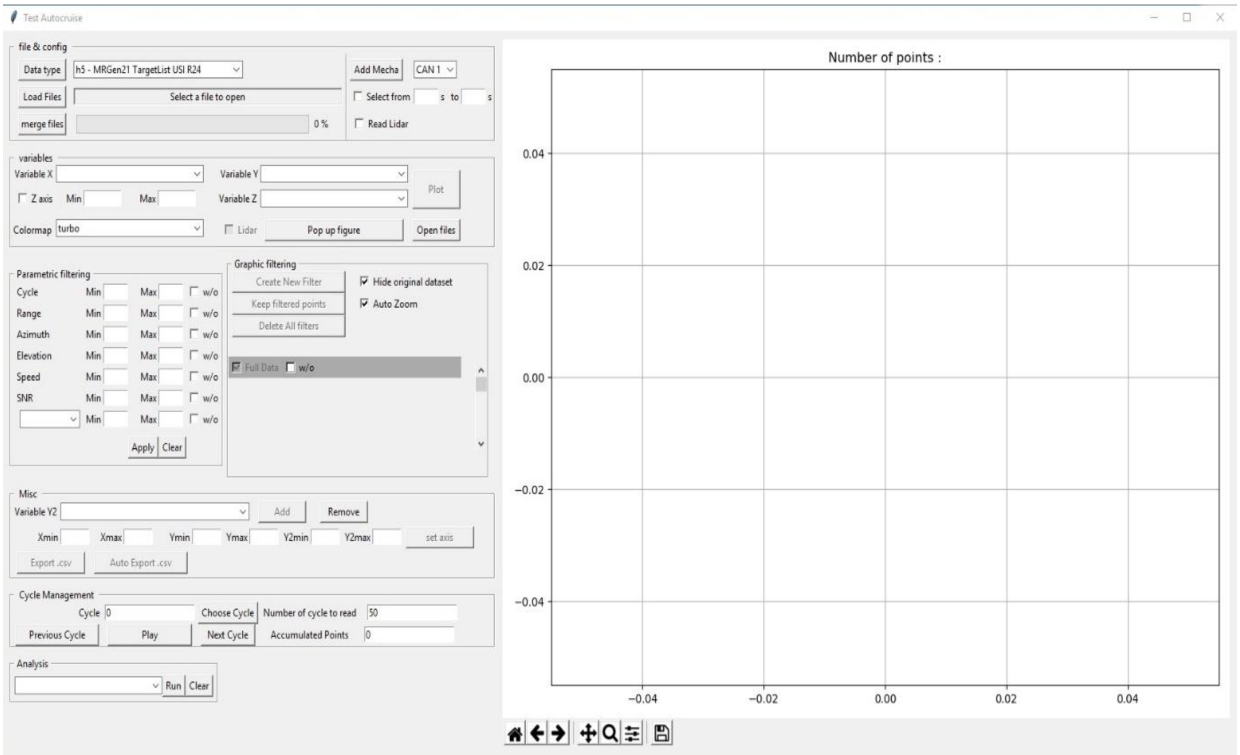


Figure 24 Python Can software

4.7. Tools used to develop Interface:

- Anaconda Spyder python v7.8.0
- QT designer
- PYQT graph
- MATplotlib
- QT designer file to .py code
- Visual Studio

4.7.1. Anaconda:

Anaconda is a Free open source distribution for most popular Programming language like Python and R. Mostly used for data science project and machine learning. It has inbuilt environment manager, that helps to manage all the packets and install new packets.

By default, you can find some following applications in anaconda Navigator,

- *JupyterLab*
- *Jupyter Notebook*
- *Spyder*
- *PyCharm*
- *VSCode*
- *Glueviz*
- *Orange 3 App*
- *RStudio*

In above all, I used Spyder to develop my tool.

***Spyder**:*
A powerful environment tool use to write python codes. It is specifically designed for Scientist, engineers and data analysts. Moreover, it also has PYQT5 extension library which I used it to develop Interactive console.

What is Python:

Python is an interpreted, object-oriented high-level programming language. It has high level programming data structure. Simple, easy to remember syntax, readability.

Comparing python with other popular programming languages like C++ and JAVA

- Python is slower than C++ and JAVA.
- Python code size is 2 – 3 times shorter than java
- Python code size is 5 – 10 times shorter than C++

4.7.2. QT designer:

It is a tool use to design and develop Graphical user interface using QT Widgets. The main advantage is that, customize windows or dialogs and test using different styles and resolution.

QT designer file to .py code:
pyuic5 is used to convert .ui into .py file
Syntax :
pyuic5 -x source.ui -0 destination.py
.ui = File extinction of QT Designer
.py = File extinction of Python

4.7.3. Visual Studio code:

- Visual Studio code is a code editor software developed by Microsoft. It also called integrated development environment (IDE).
- Use to develop computer programs, websites, web applications etc.

4.7.4. Matplotlib:

Matplotlib is a collection of function that could preform MATLAB functions by python programming language. It has a comprehensive library which helps to do signal processing, 2D, 3D, math works, interactive visualizations etc.

4.7.5. PYQT graph:

It is a python library, which is like matplotlib used for scientific engineering applications. This library is completely written in python NumPy. It is very fast compare to Matplotlib

4.8. TESTBENCH

It is an environment, which generate real time environment. By using test bench, testing devices, error alignment, manipulation of the tools has been made.

4.8.1. Testbench Block Diagram:

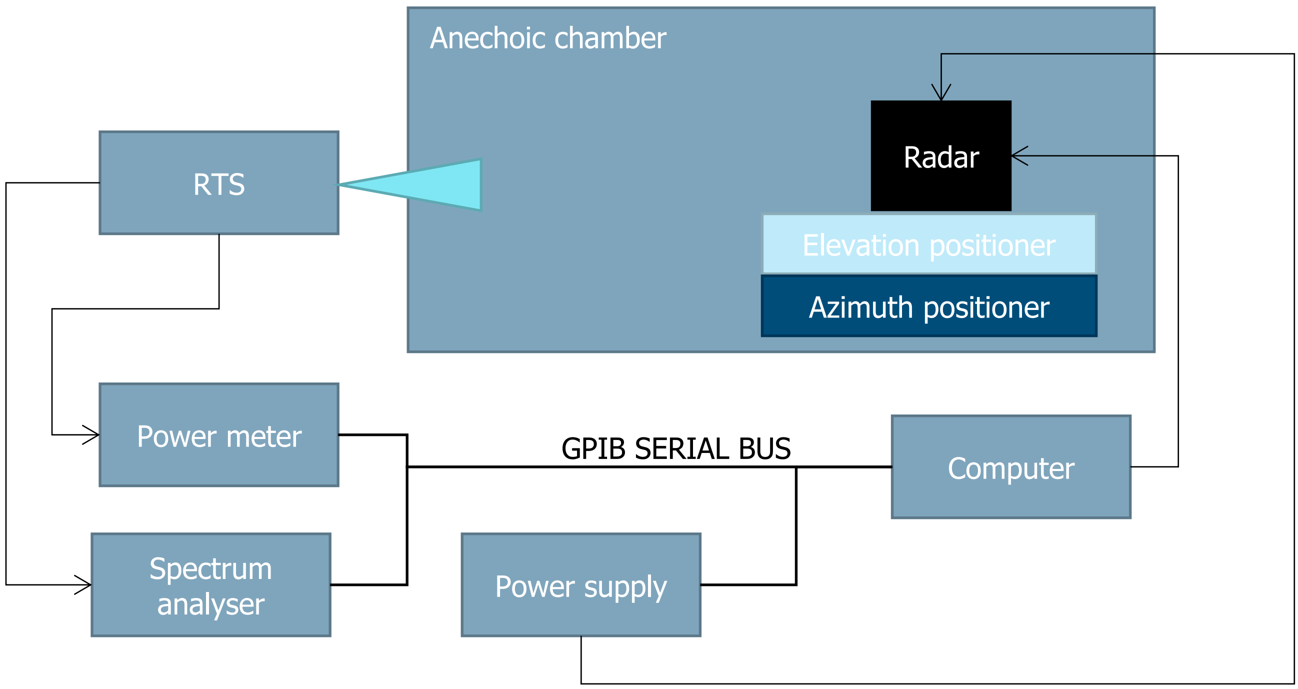


Figure 25 Test Bench Block Diagram

- Autocruise has specific software, which use to control the Azimuth and elevation of the positioner.
- The azimuthal or elevational position is moved according to standardized sequences in front of the radar target simulator

4.8.1.1. Anechoic chamber:

- This chamber is covered by cone shape radio waves absorber, to avoid reflections between the radar and the target (suppression of multi-path errors)
- Radar positioner and RTS are placed inside this chamber.
- With the help of RTS, the radar can read back the radiated signal. All other will be absorbed by the absorbers.

4.8.1.2. RTS (Radar target simulator):

This device receives the radar signal and apply delay and/or doppler shift on it. When it sends back the signal, the radar can identify that there is one object at a distance with a certain velocity. Distance and speed of the object are user controllable by specified software.

4.8.1.3. Spectrum analyser:

It can observe the spectra of electrical signals, dominant frequency, power, distortion, harmonics, bandwidth, and other spectral components of a signal which are not easily detectable in tile domain waveforms. These parameters are useful in the characterization of electronic devices such as wireless transmitters.

Spectrum analyser is connected to RTS which helps to check the signal which are emitted by the radar.

4.8.1.4. Power meter:

- Used to measure power of the received signal from the radar. (In addition to the spectrum analyser)

4.8.2. Image of TestBench:

Figure 26 Anechoic chamber

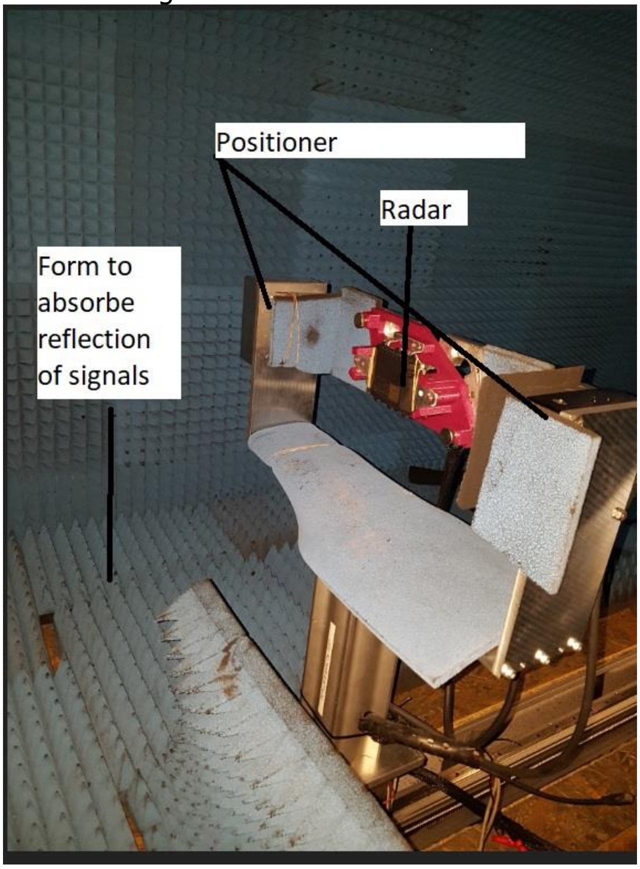


Figure 27 Positioner

5. WORK DONE DURING INTERNSHIP

My mission was to develop a Human machine interface. The main purpose of this tool is to visualize the targets which are captured from respective radars (MR and FR). Autocruise has given me a Laptop with preinstalled software on it, so that I can start developing the tool without facing any software installation problems. My teammates provided me some documentations and explained what they really need from this tool. They also suggested some APIs that would help to improving processing speed of the tool.
I started by developing prototype of the tool. Image of prototypes are given below.

5.1. Prototype

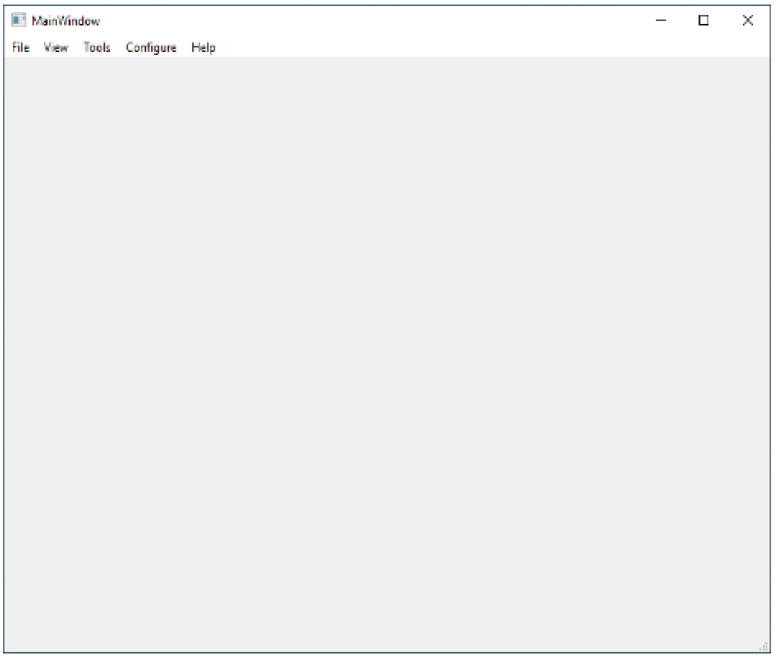


Figure 28 Prototype MainWindow

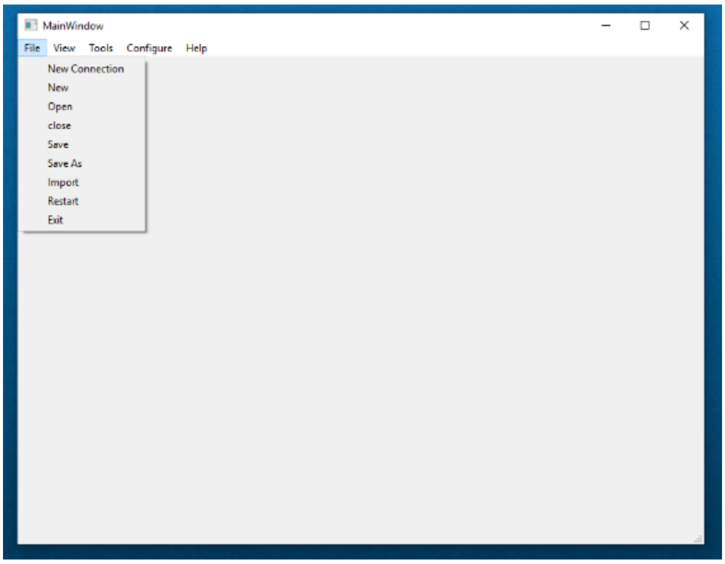


Figure 29 Prototype FileMenu

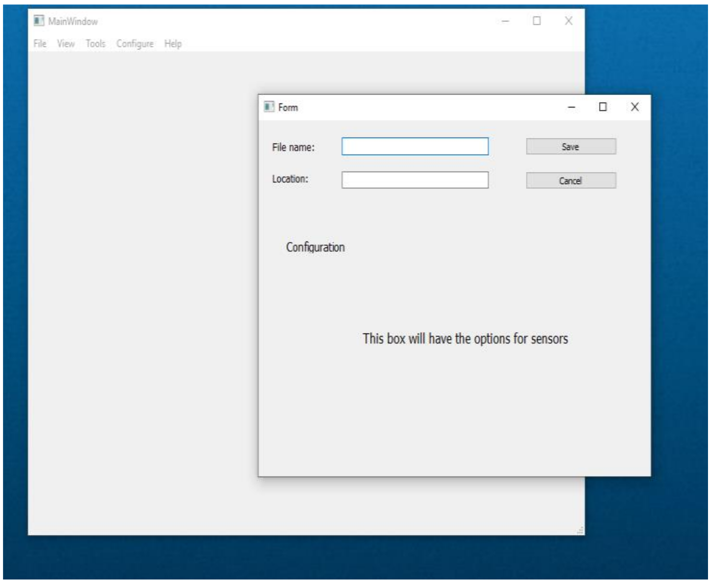


Figure 30 Prototype New Connection

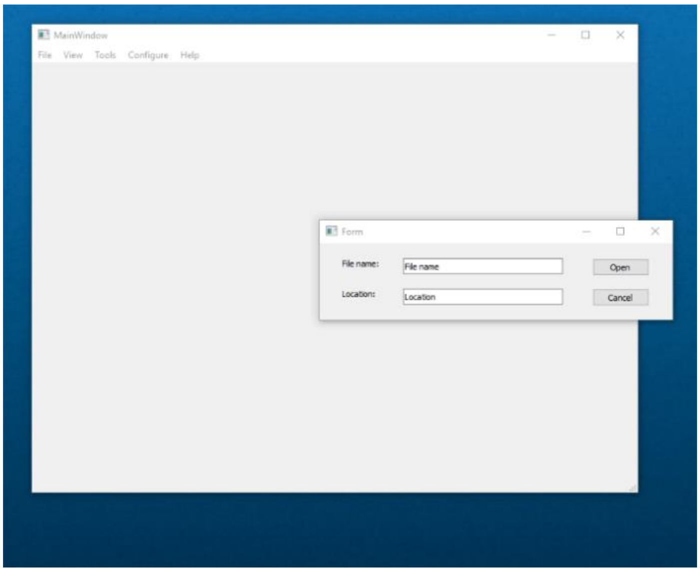


Figure 31 Prototype Open Window

5.2. Work Flow:

5.2.1. Data's from RADAR:

Radar provides huge amount of data for every ~60ms.

MRgen21:

This radar uses 6 different UDP port and provide data's at every ~60ms. Previous software named UDPMAN , access all the ports simultaneously and provide those datas in binary (.bin) formate.

They are RAWDATA, Target list frequency, Dynamic parameter, Static parameter, RF Monitoring, Target List USI, TxMIMO. Upon this list, Target list USI contains the parameters use to do visualize of targets. 54 parameters in that gives details description about one cycle. From this Range, azimuth, elevation, SNR, RCS, Level, no of targets, are considered as important parameters.

FRgen21:

It is advanced. It uses single port and gives 44 parameters at single cycle. But it gives more precised data compare to MRGen21. Same as MRgen21, it also has Range, velocity, Azimuth, SNR etc for single target.

Note:

Target: An object which are detected by the radar. Range, azimuth, elevation etc are respect to targets.

Cycle: It has the data for multiple targets.

Read all the parameters from respective radar and plot it with respect to time. Every individual parameter must be plotted with respect to time.

5.2.2. Processing .bin File:

Started my Working on visualization by using .bin file, which has list of targets for MRgen 21. Autocruise has specific binary reader developed by a engineer in autocruise. That helps to decode .bin file in microseconds. This makes my job easier.

With 60ms timer and data visualization process was successful using matplotlib.

Using Matplotlib, data visualization was successful with 60ms timer. Matplotlib is huge library and it need more processing power. Required refreshing rate during plotting was not enough. PYQT graph has fast refreshing rate compare to Matplotlib. Data visualization from .bin file using PYQT graph was succeed.

5.2.3. Connection Diagram:

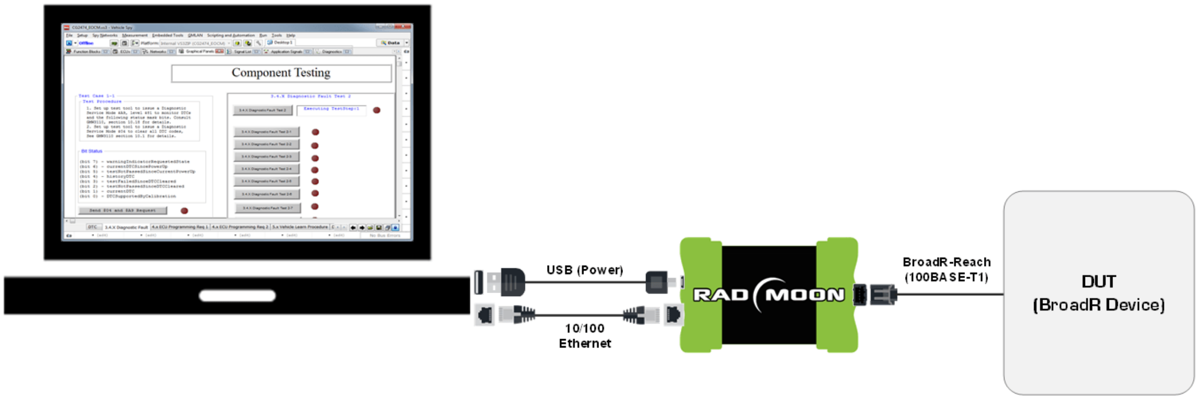


Figure 32 Connection Diagram from Radar to Computer

Above figure show how the connection has been made between computer to radar using radmoon. DUT is replaced with radar in the connection.

5.2.4. Software Flow chart

Having flowchart for the work before starting will make the job perfect. You can find the flowchart of the software below.

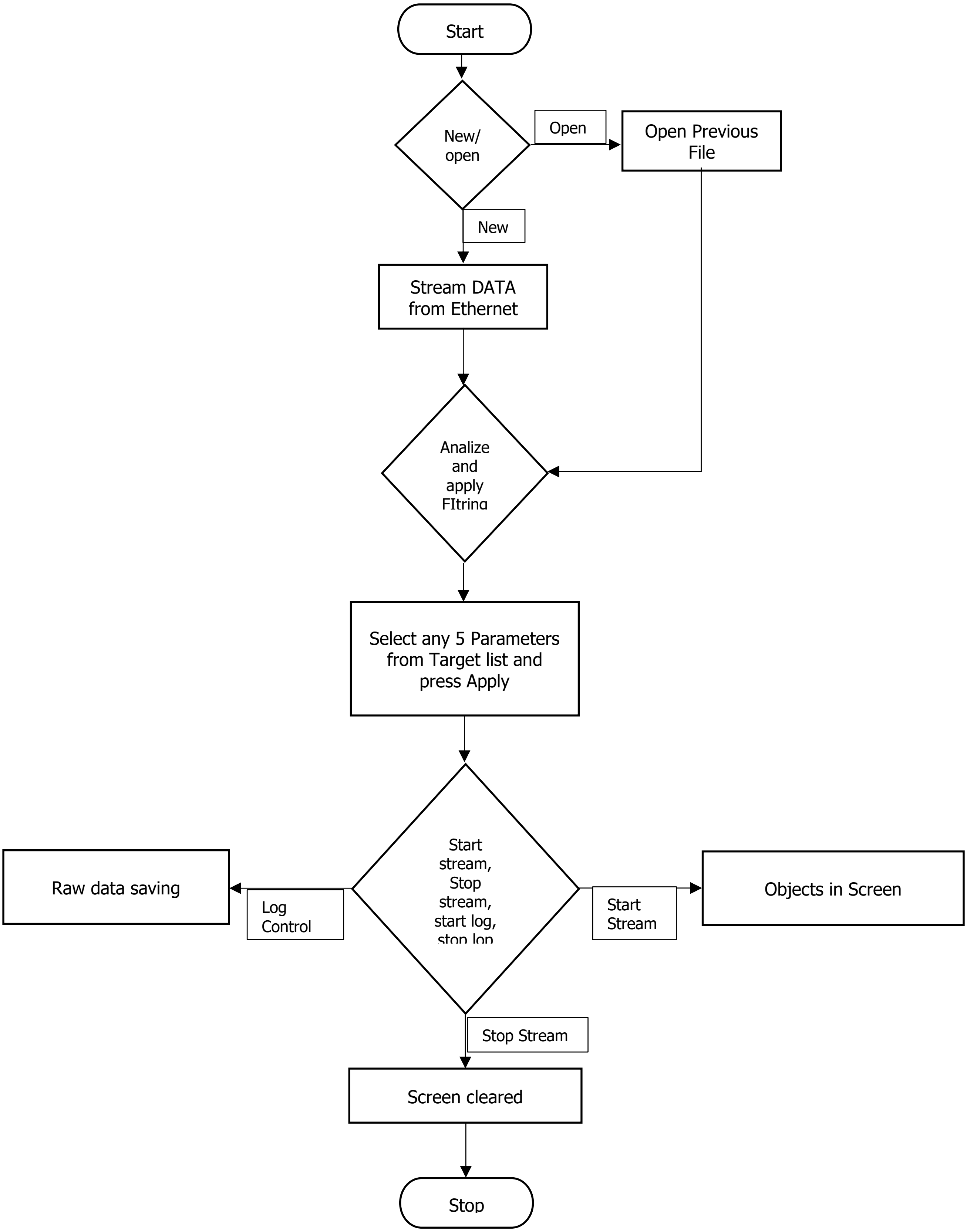


Figure 33 Software Flowchart

5.4. Finalized Tool:

Below images explains the software structure.

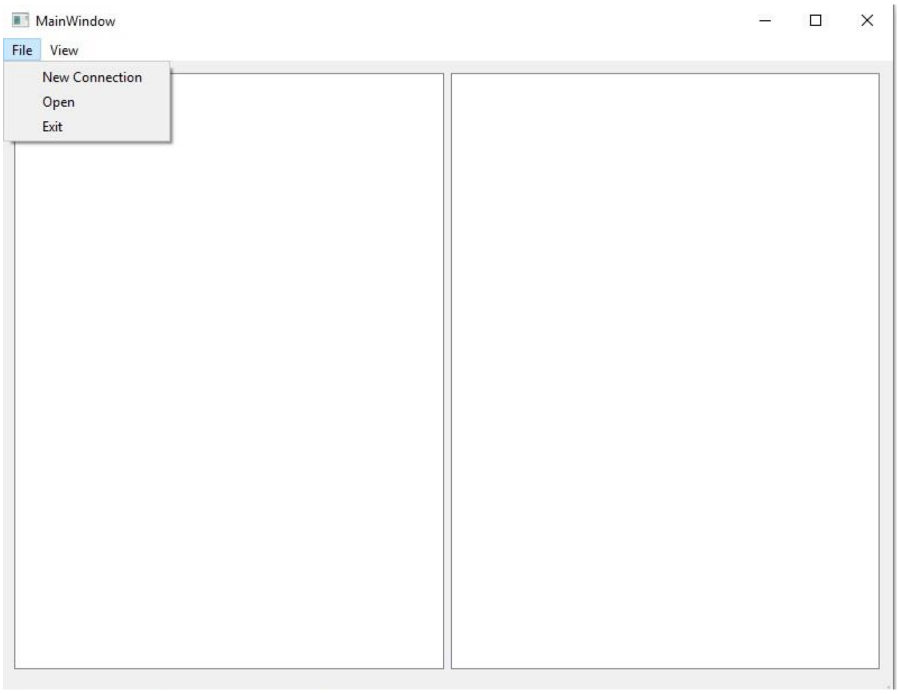


Figure 34 Main Window

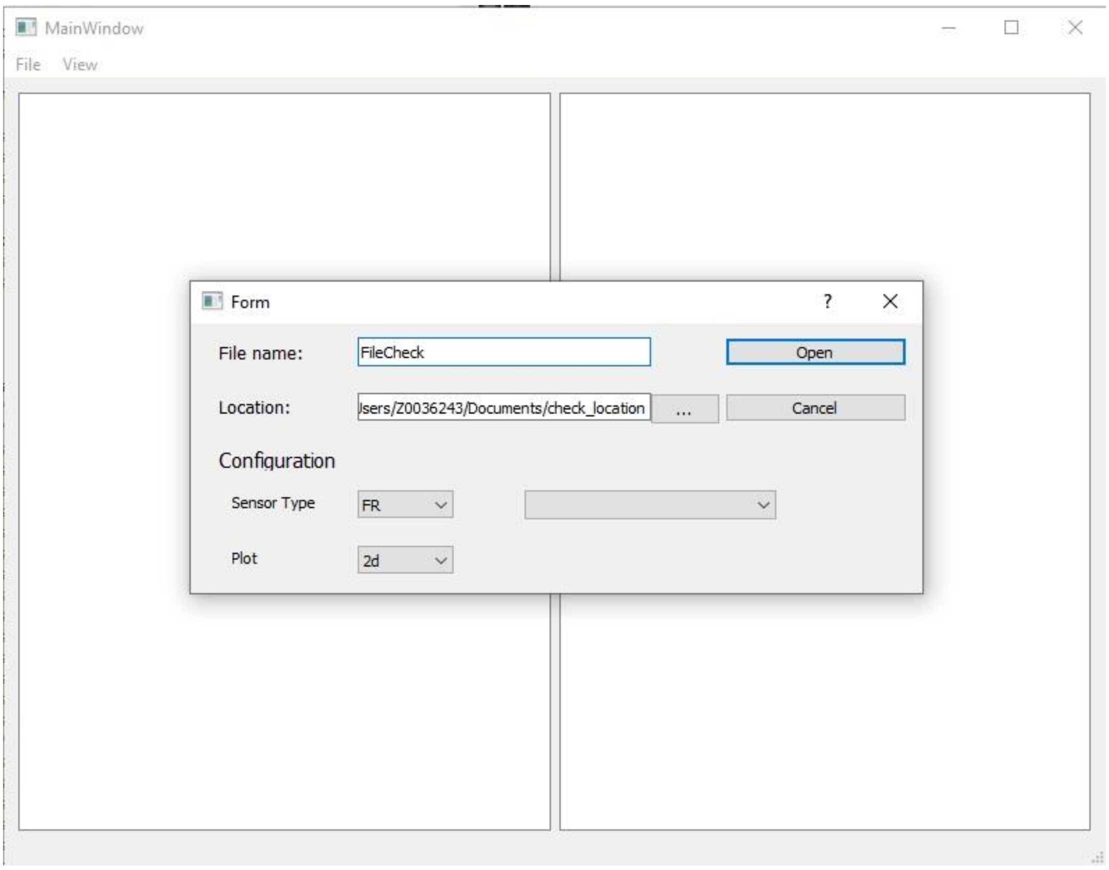


Figure 35 New Connection

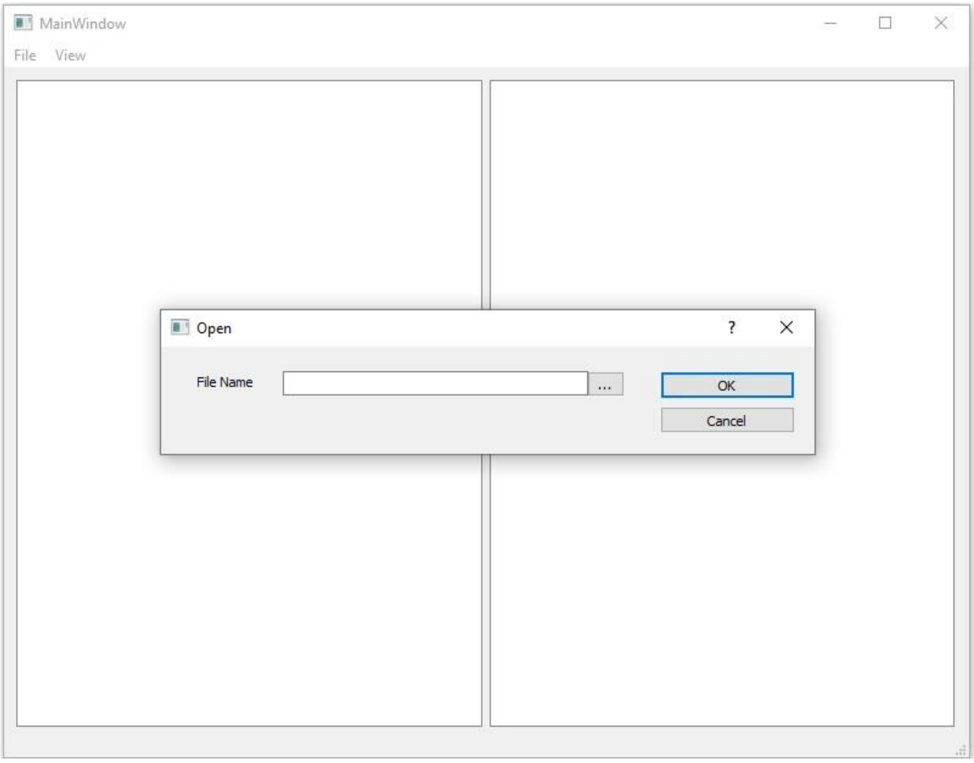


Figure 36 Open File

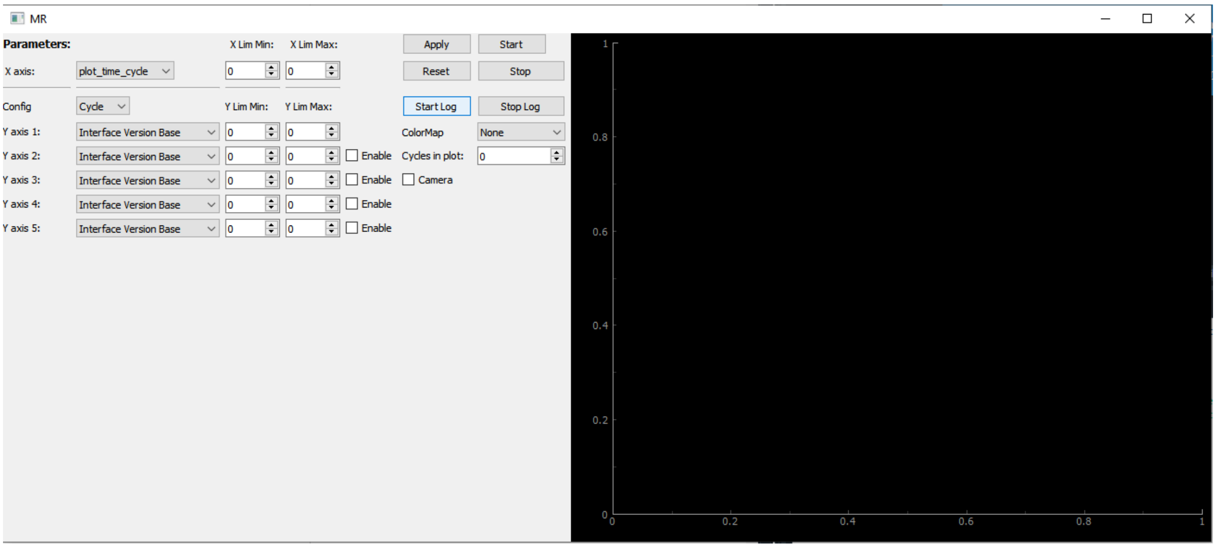


Figure 37 FR Plotting

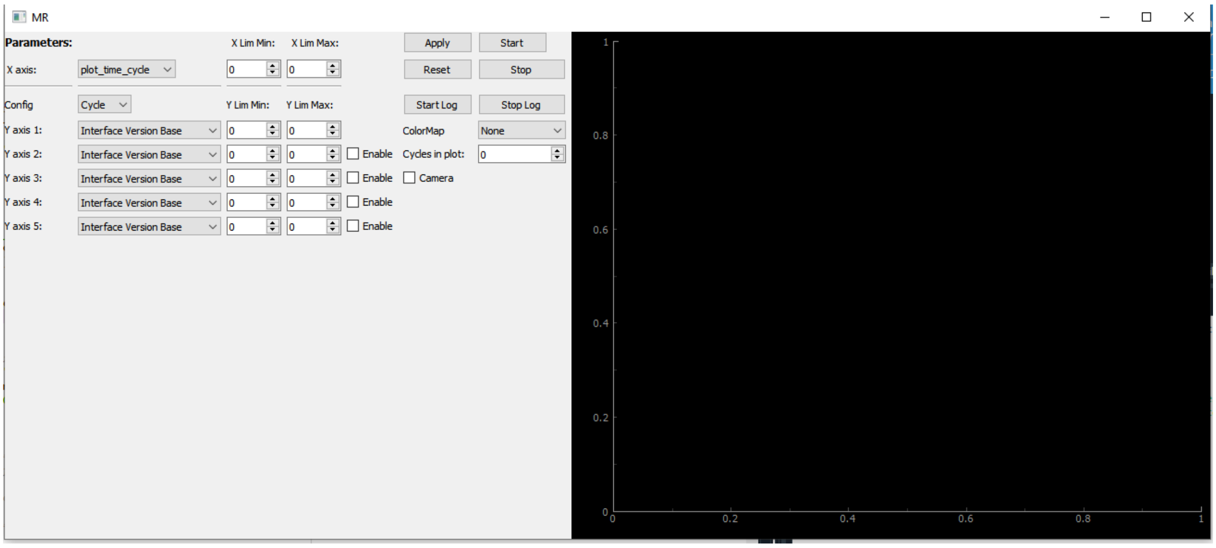


Figure 38 MR Plotting

- This tool can stream live data and plot it on the screen
- Compatible for MR and FR
- Section of MR and FR can be made in new connection window.
- File name and location for new connection are available in single window.

5.4.1. Plotting window:

- Simultaneously multiple Y axis can be enables with specific parameters.
- Selection of X axis is depending on the parameter in Y axis.
- X lim and Y lim are customized and also be changed during run time just by scrolling cursor
- Change values from 1 – 500 for MR and 1 – 40 for FR in “Cycle in plot” can help to keep previous cycles in screen

Apply button:

Press apply after finishing all configuration for plotting.

Start button:

Start the threads to stream and plot data. This enabled, after applying the configuration settings.

Stop button:

Stop the streaming. This enabled only, when streaming starts.

Reset button:

Resets all the configuration to default values.

Start log :

To start recording data

Stop log:

To stop recording data

ColorMap:

Adding selected parameter by colourmap in bird view gives more detailed visualization of the radar. Mostly this function is used in Bird View.

Cycles in plot:

Number of cycles to keep in the plot while live visualization is on progress.

Camera:

Enabling or disabling this option will open separate window , that shows live video, if USB camera connected to the respective laptop or a computer.

Camera :

Enable this option will open camera in separate window when during live acquisition.

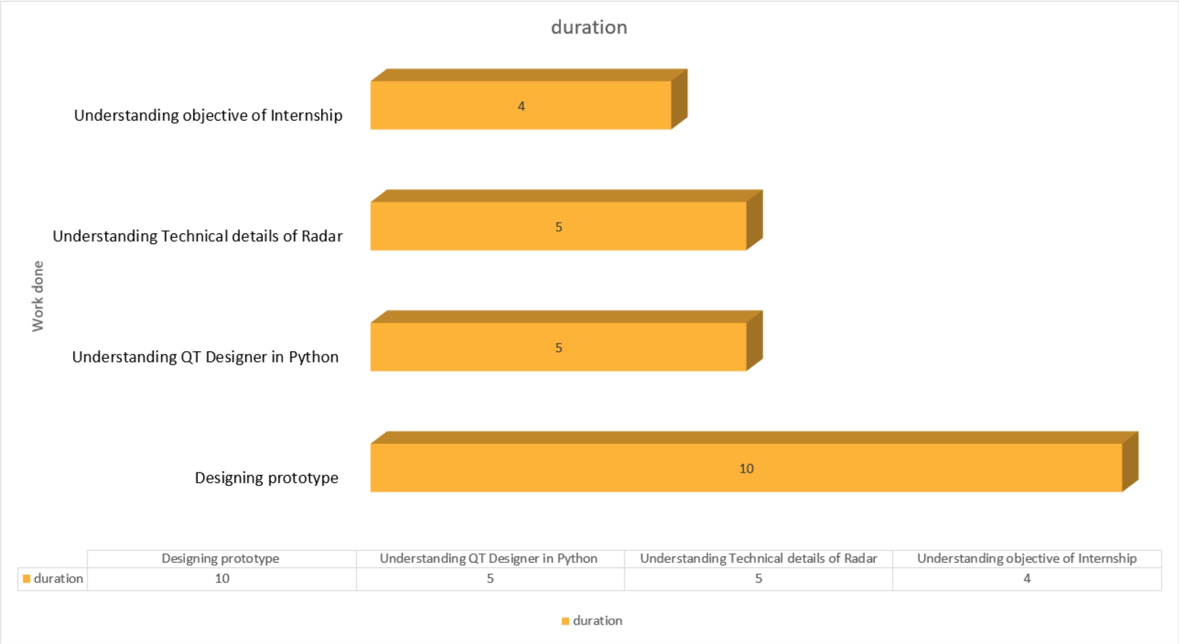
6. PROJECT SCHEDULE:

Project Tracker

Start date:	3/16/2020
End date:	8/28/2020

Column1	Start Date	End Date	Milestone/Activity	Start on Day	Task Duration
1	3/17/2020	3/20/2020	Understanding objective of Internship	0	4
2	3/23/2020	3/27/2020	Understanding Technical details of Radar	6	5
3	3/30/2020	4/3/2020	Understanding QT Designer in Python	13	5
4	4/6/2020	4/15/2020	Designing prototype	20	10
5	4/16/2020	5/20/2020	Decode File and plot with 60ms	30	35
6	5/22/2020	6/17/2020	Working on FR	66	27
7	6/18/2020	7/31/2020	Working on MR	93	44
8	8/3/2020	8/14/2020	Working on Report and Presentation	139	12
9	8/17/2020	8/28/2020	Testing and Finilizing Tool	153	12

Figure 39 Timing schedules



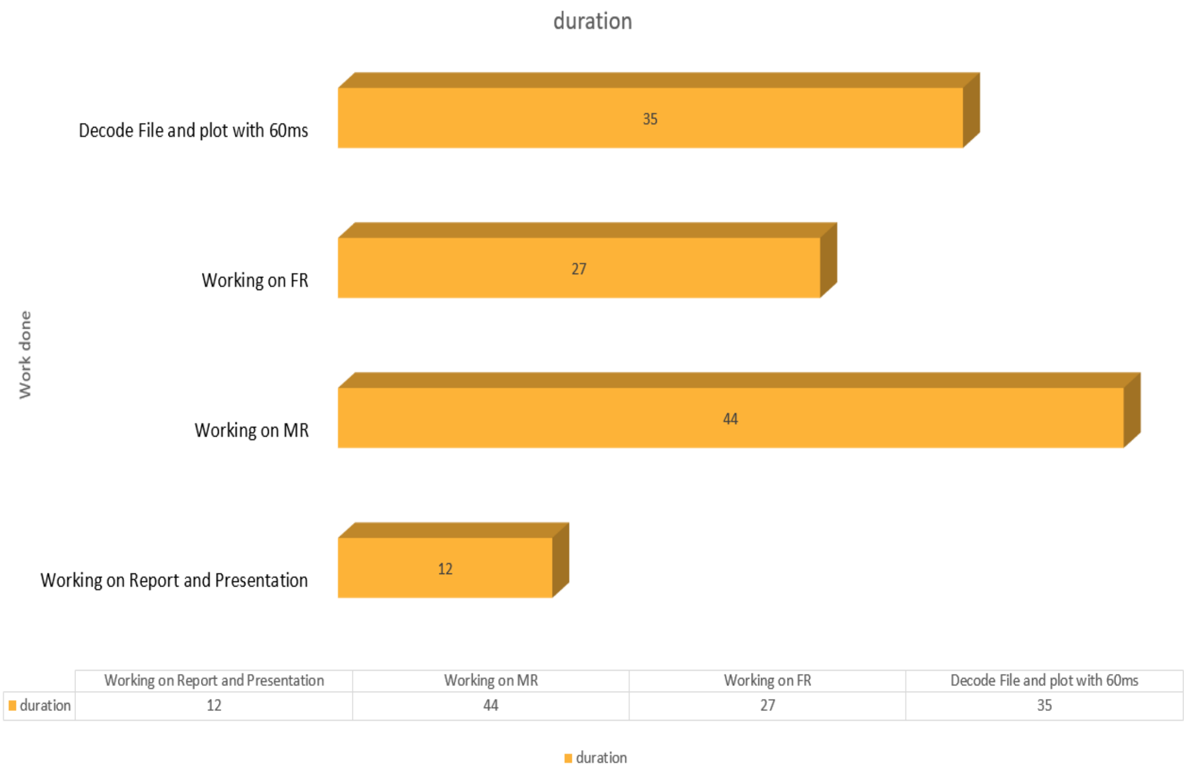


Figure 40 Time scheduling Diagram

I have started this internship with no much knowledge of Radars and COVID – 19 confinement. Figure 41 explains, Working on MR took 44 working days because I was told to translate previous software written in C programming into Python code. Reading and understanding of another person concept was challenging for me. I got some guidance from the colleague understand the concept of that.

Second longest time consumed task was to Decode previously saved file and to plot at every 60ms. Because of Lockdown I didn't have an opportunity to check radar in real time to understand how it works. Thanks to Steffen, he made me understand the concept of radar working during confinement.

The above two are the tasks that consumed more time than expected.

My manager didn't force me or gave me fixed time to finish the work. All planning and schedules are done by me.

7. DIFFICULTIES FACED:

7.1. Working from Home:

Because of COVID-19, the whole country "France" where I am living, had decided to go on Full lockdown to reduce spreading of virus. The starting day of lockdown and my starting day of my internship were lay on same date. ZF Autocruise asked everyone to work from home.
Thank for my manager, he came to my apartment to provide me the laptop. So that, I could work from home.
I felt very difficult to understand what my colleague was trying to explain me. With the help of skype calls, eventually, I understood what my work would be for the next 5 months.

7.2. Working culture:

During my previous internships, I was told to finish a project on time limit. But in ZF autocruise, it wasn't like that, they are treating people to be autonomous and to schedule their work by themselves. When I exposed into this working culture, because of some introvert character in me, I felt more difficult and hesitated to speak with people. Skype was the connecting medium between me and the company members. I realize that, I have to give some time for my brain to adapt for this.

It took some months to get adapted to this working style and culture. I learn, how to be autonomous, responsible on my work and finish job on time.

7.3. Technical difficulties:

Main problem in live streaming of radars is reading and plot data within 60ms. During my first step, I tried using simple concept by introducing loop between reading and plotting. It was failed, because sequential process executes line by line, it won't suitable to achieve plotting in same 60ms. Searching for another way gave me pipelining concept. Applying pipeline helped me to plot the data at same 60ms.

List of problems faced while trying to achieve plotting with in 60ms are listed below:

Decoding Binary data:

I have explained in previous topic how the data are sent and received. Speaking about reception using UDP port, Radar sending data in Binary format. Regardless of radar, data has identical parameters for both radars (MR and FR). MRGen21 and FRgen21 have their unique data frame when it comes to sending packes in UDP. Both radars are not working in same data Frame. Finding the starting packet and merge it was the problem. Converting the received packet from binary into decimal to find out the what data sent with encoding method and the packet number . With the help of packet number, first packet was found and and combined to get a proper cycle.

Introducing threading:

As I explained above data's availability at UDP port. As soon as data arrived at the port, it must be plotted. Cycle from radar will be available approximately at every 60ms. Initially, I tried using matplotlib, it needs high processing power and tends to be slow in plotting. Total time taken to plot one cycle requires approximately 65ms. Within of 5ms, there are high probability of losing data. To solve this 5ms, threading concept had been introduced in the code. Reading UDP port in separate thread and plotting data in separate thread. Try to run two process concurrently gave better results than before. Because of heavy processing of Matplotlib, plotting time was not good enough.

Replacing Matplotlib with PYQTgraph:

After getting successful result by threading, and drawbacks of Matplotlib, I decided to move from matplotlib to PYQT graph. PYQT graph is similar to matplotlib, which has low processing power and high plotting speed.

Translation from C code to Python code:

The previous software used to read MR had written in C programming language. I had to integrate this previous software into this tool . The problem was, I could not run it. In order to translate and get understood the logic, I must run and check it. To solve it, I asked my teammates what all the important things to be considers from this software and what exactly needed from this tool, so that, I could understand the logic. Eventually, I understood and integrated into the tool.

Decoding Byte array with respective parameter:

While Working on MRgen, data received in UDP port was in binary format. After successful completion of finding cycles from binary data, all the data's have to be decoded into human understandable parameters and be stored in specific variable or in dictionary. I saw couple of failure while trying to decode the binary data, finally I decided to store data in binary file and read it back to decode with respective encoding method. That worked. Reading and writing binary file took some time. Using one cycle in temporary file was fast enough to have live visualization. It didn't give any delay while plotting.

Saving Multiple cycles in single file for further use:

Data read from the radar must be saved for further use. Using temporary variable inside the code and save all the data at same time was reducing threading speed. Using multiple external file helped to maintain the same threading speed. When stop button pressed, all files will read back and save it in a single file.

8. RESULT & FURTHER DEVELOPMENT:

8.1. Tool tested for MR in Testbench:

Detection of target depends on speed of the vehicle.
During Testbench testing, Radars are Configured directly by changing non-volatile memory using CAN communication.
In Realtime, radar internal embedded software communicates with other control units to get vehicle speed and depending on it, will change its mode.

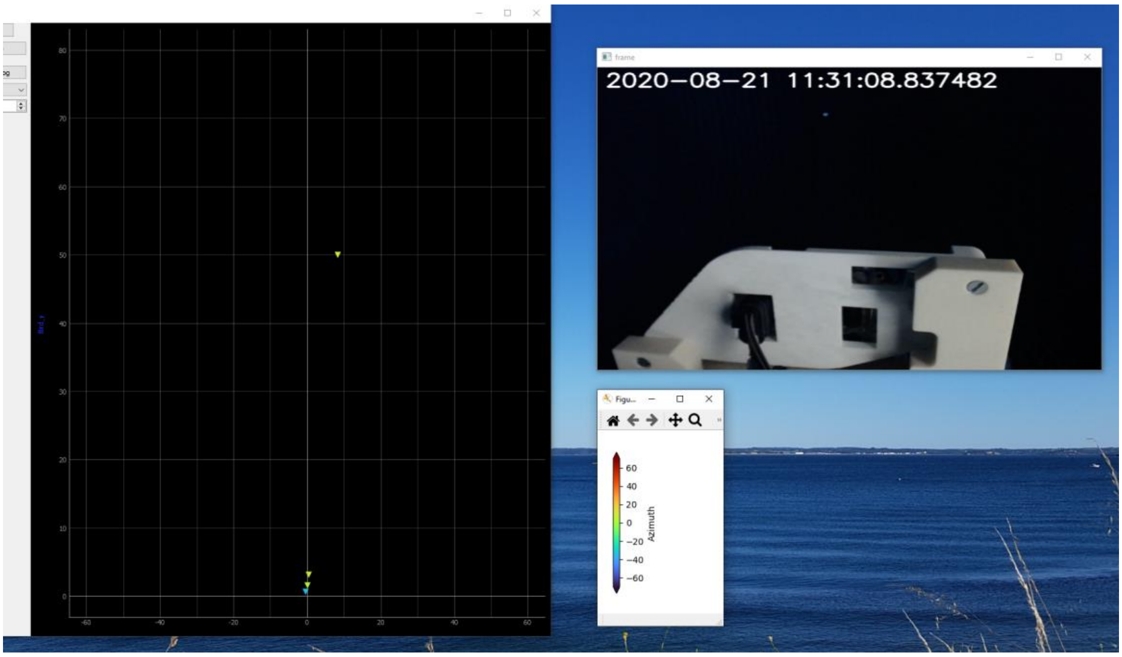


Figure 41 RTS set to 50m and positioner set to +10°.

Above figure captured during testbench working on low speed mode. You can find the target in green triangle . X axis is Azimuth and Y axis is Distance. This is because , Tool configured for birdView. You can see then target lies at 50m at 10°.The x axis is in azimuth and the colour map is in azimuth. Azimuth in 10°, the colourmap and the triangle colour are same. Checking with x axis, as target is at 10° and colour map gives the value for 10. By changing the colourmap parameter, example power of the signal, will show the signal strength by colour .

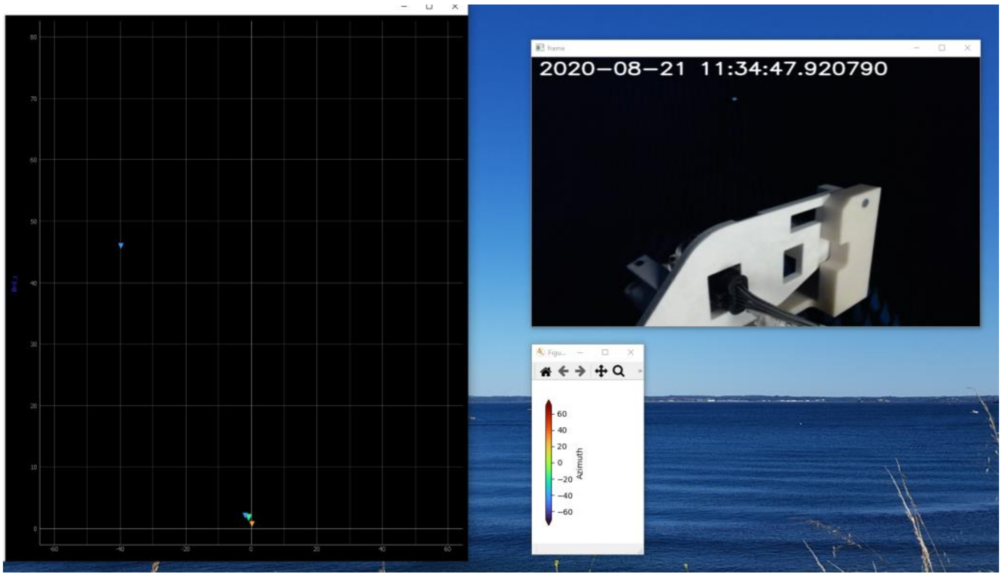


Figure 42 RTS set for 60meter and positioner set at -40°

Above figure captured when RTS set for 60 meter. Here target is blue Triangle shows azimuth at -40. Also it verified with X axis.

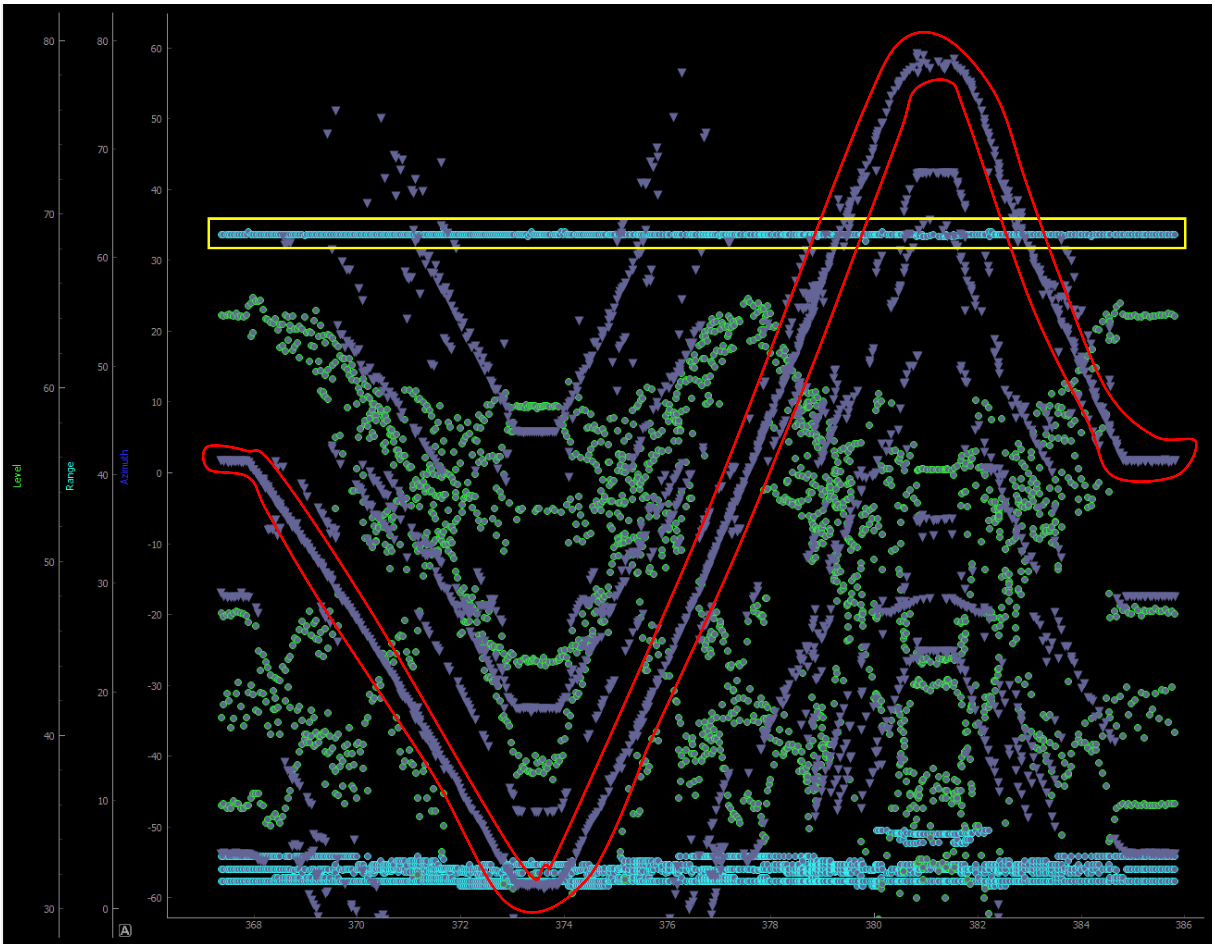


Figure 43 Multiple data's at single scan

Above figure is taken while testing the tool with multiple parameters in testbench at low speed mode. This tool configured to keep previous 500 cycles.

Points shown using red free hand drawing are the targets. This image was took during positioner movement from 0° to -60 , -60° to +60°, +60° to 0. This sweep showing the testing of Azimuth level in Radar. Other points out of red zone are reflections from small corners inside test bench which are not cover by absorber.

Points shown inside yellow box shows the target at 61 meters during whole sweep. Here RTS is static object. Which do not move. Distance are configured inside RTS, it creates need delay in the received signal and sent it back. So, the radar can identify the Target. This is the reason target is stable at 61m until the sweep finished. Points lies between 0 to 10 meters are small reflections from the test bech.

The green dots are showing the power of the target. i.e signal strength of the target. When positioner is at 0°, radar and RTS is at 90° i.e radar receive the reflected signal in high power. Comparing Pink triangle target with green bubble explains, when the target at -60° and in +60, power of the target is low, when targets moves to 0° , you can see the gradual change in power level.

8.2. Tool tested for FR in real time :

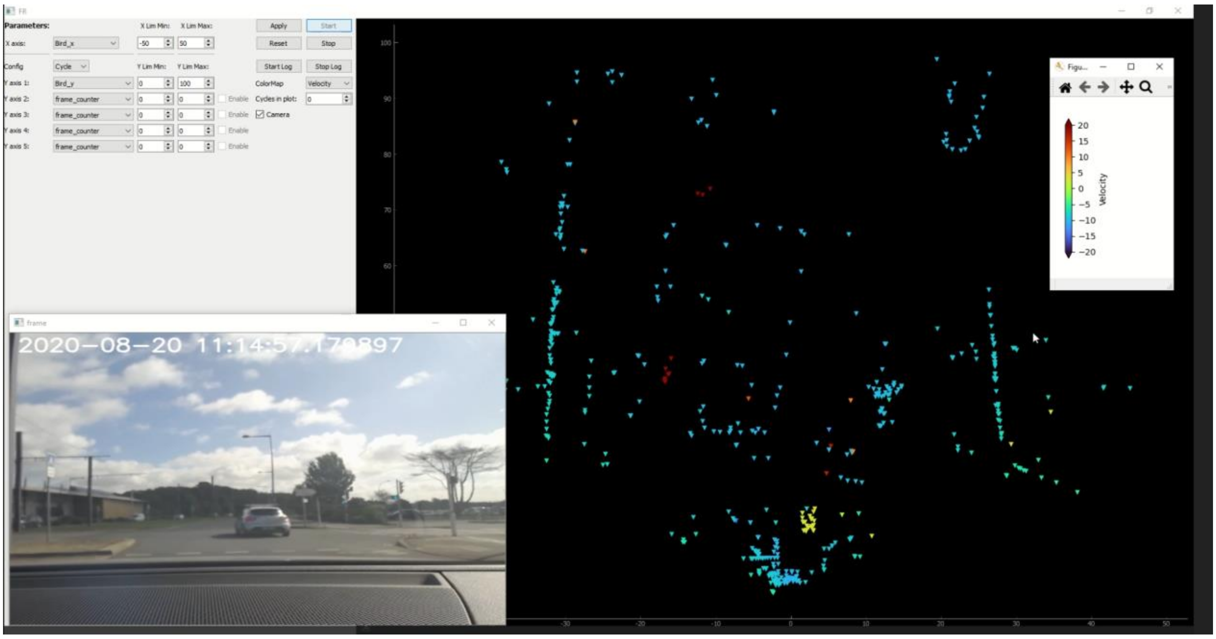


Figure 44 Real time capturing of FR (1)

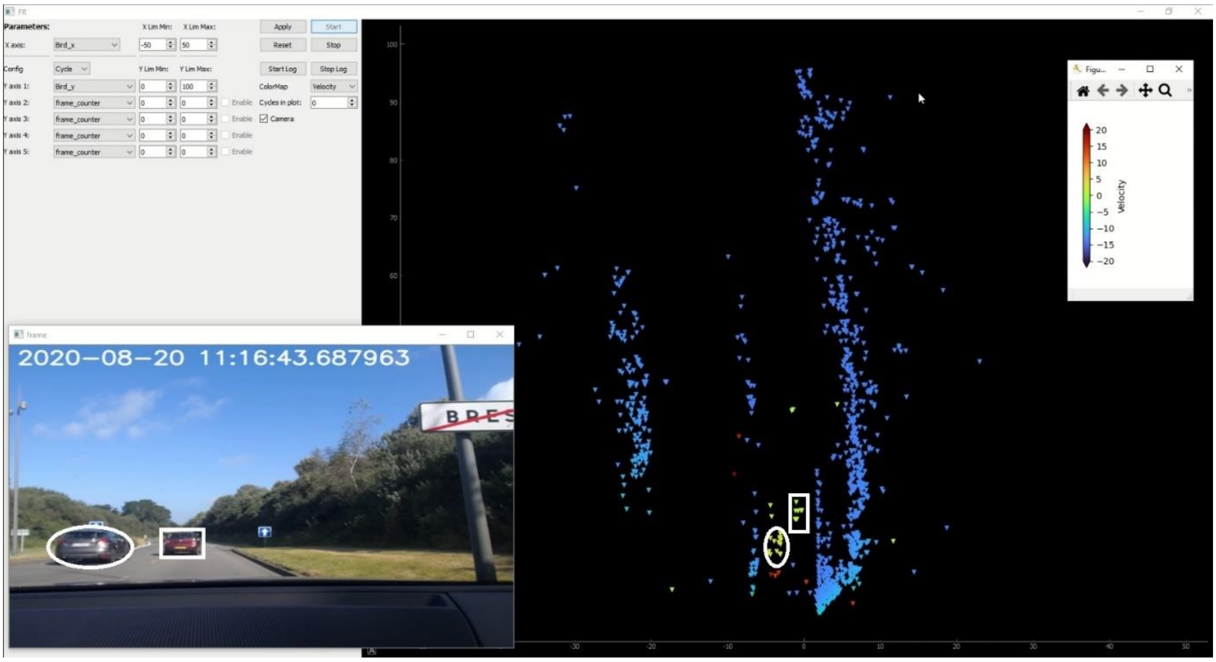


Figure 45 Real time capturing of FR (2)

Above figures explains how the radar captures the objects.

Figure 45 and 46 was taken during real time testing of the software.

By enabling the camera option, camera window will automatically opened.
In the Figure 45, plotting was configured for Birdview with velocity in colourmap.
This differentiate, static object(trees, grills, notice boards , etc) and moving objects (Car, truck, pedestrians etc) .In the plotting you can see, the Green dotted points in the plot with colourmap gives 0 velocity. 0 velocity is the difference of the speed from radar to another object.

Figure 46 shows, multiple objects are detected in different distance and in same velocity. Radar was configured for low speed mode. It cannot capture long distance. Since, you can see in the figure that, radar captured for 100m .

8.3. Further development:

- 3D plotting to be added
    - This is possible as target data have spherical coordinates (Range, Azimuth, Elevation)
- Real-time filtering
- Communication with sensors using CAN
- Interface IP Camera replacing USB camera.

9. Feedback:

9.1. Workflow:

Autocruise is my first company in automotive sector and in Europe. This company named for radar more than a decade. With minimal knowledge of radars, I have started my internship. Thanks to my Team, and team manager for helped me to understand about Radars and its works. Especially during confinement.

9.2. Unplanned problems and solutions:

I almost finished half of my internship by working from home. After easing lockdown, company opened for important works, which could not do from home. I used this opportunity to work on test bench. After releasing the lockdown, it made my job to done regularly from company.

9.3. Language:

With some basic knowledge of French, I arrived in France. Thanks to the French courses from the school. At the end of my course, I could converse with people in French. Because of my hesitation in French, I started applying for English speaking internships.

I am quit lucky person, I got internship where everyone speaks English. I was thinking to use this opportunity to improve my French speaking skills. Here comes the problem COVID-19. I started speaking with my neighbours. I realize that, Brittany people have separate accent, it took me some time to adapt for this accent. After started working from company, slowly I started speaking with people in French. I believe that, I will speak French properly in future.

9.4. Cross Culture environment:

I never had the chance to feel the new culture and language before. Thanks to everyone who accepted to hire me as an intern. Autocruise has mixture of international people. Most of the people are from local regions. Interacting with Brittany accent and culture felt entirely different from the place I used to live previously (north of France, a city named Rouen). After I used to culture and accent, i felt comfortable to work with them and in that city. It feels great and interesting as this is my first professional experience in my carrier.

9.5. Work environment:

This is my turning point of my life. Company environment, relationship with manager, with my teammates and with other colleague changed my life. Apart from technical work, they have some parties and exercises that even helps to interact with all other people working in the company.

Due to pandemic, company kept hand sanitizers, tissues, face masks in all the corners and in Rest rooms. Also, they gave me a small course on COVID-19 safety procedures which must be followed until further notification arrived from company management.

Speaking about environment, Autocruise is situated near to sea. If there are overwhelming stress, fresh air and beautiful view from the company help to reduce stress.

10. LESSONS LEARNT FROM INTERNSHIP:

Familiarized myself in automotive sectors. Technologies and products gave me a chance to see the world in field of technologies.

Weekly Interaction with the manager and teammates leads me to get aware of technologies and felt comfortable to work with them. My experience in previous internships were task oriented. Liberty given by the company had improved my autonomous thinking.

Using proper documentation and commenting on the work was essential. It showed me how that will make my work easy to understand and upgrade my work in future if needed.

From the day one, interaction with my manager was informative in technical and in personal. I spoke about PHD and full-time job, though he could not offer me any job or courses. He provided me reference letter for my career search.

With the help of team members, I learnt about working culture, behaviour inside team members and in meeting.

Speaking about technical knowledge, what I learnt during this internship fulfilled my expectation and enriched my experience.

This internship showed me a new way for my career. Knowledge about programming, radar, culture, languages, that shows me how I want to mould myself in personal and professional life.

11. CONCLUSION:

Software is an important tool in embedded market. It has many advantages like problem solving, speed, accuracy etc. Possibility to upgrade in future if needed.

ZF autocruise gave me an opportunity and allowed me to get experienced in automotive sector. Besides experience, it also gave me a way to compare my previous experience and my education with reality.

The discussions which I had with my teammates and manager have shaped my thinking and made me think like a professional. The motivation and gratitude which they showed me gave more confident on my work. Working under this dynamic and enthusiastic team will take to me to next level in my carrier. I realize what is engineering profession from them. I learnt to be rigorous and meticulous, develop my capacity for adaptation, integration and to be autonomous.

I figured out myself in this 6 month of internship which I never seen before. I don't know how I worked during in lockdown, time went like a Rocket. I learnt lot of things from this internship. Thanks for ESIGELEC and Autocruise who gave me an opportunity for starting my career as a developer in France.

With grateful regards, I want to thank ZF autocruise and ESIGELEC for helping me to get an industrial experience from a reputed company. Great thanks to my Manager who trust my skills and gave me an opportunity to get experienced. Special thanks to Stephen, Thierry, Tristan and Julien for guiding me throughout my internship period and also during pandemic.

12. SUMMARY:

Autocruise works on the automotive radars. Here, my job was to develop a tool which give live visualization for the radars. As everyone knows about covid-19, due to this lock down, I started my internship via home office. With the help of skype meetings, expectation from the tool and the internship goal became clear for me.

I started my work by developing prototypes for the tool, then with radars. Introduce threading helped me to get the need processing time to plot the data in real time. Moreover, to improve the software efficiency, the plotting library was changed from, matplotlib to pyqtGraph. Addition to that, colourmap gave additional detailed information for the visualization. This tool was finalized by adding the camera using open cv library into the code and gave recording option into that without any lag in live visualization. The main difficulty during my internship was the pandemic, working from home without clear information took me lot of time to understand the whole concept. As soon as lockdown became easy, I started my work from the office.

To conclude, the dynamic team, and the company, gave me liberty, to work and finalize the tool before the deadline. This also gave me knowledge on how to be responsible, autonomous, team management etc.

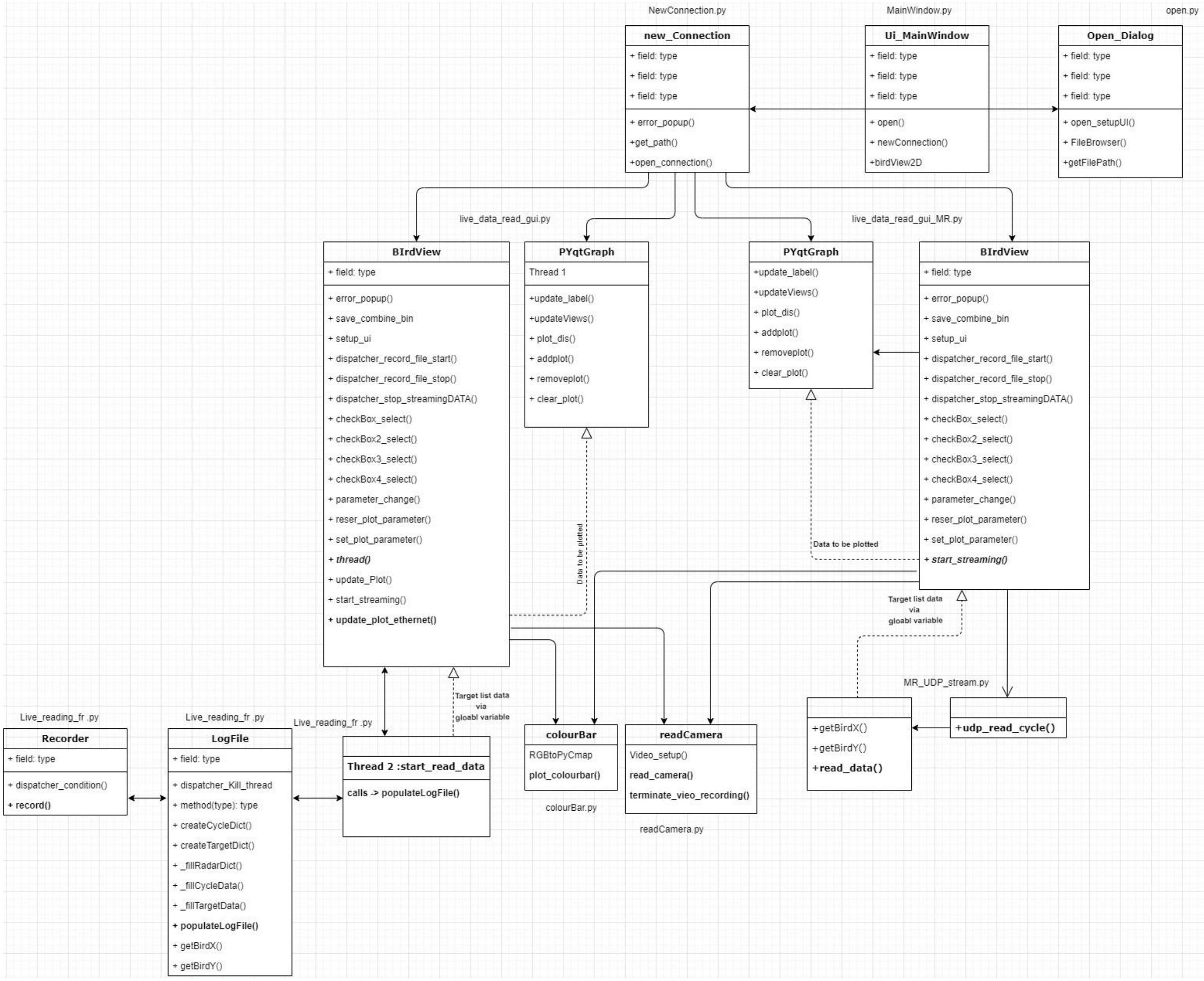
Keywords: Radar, python, Threading, pyqtgraph, colourmap, openCV

13. REFRENCES :

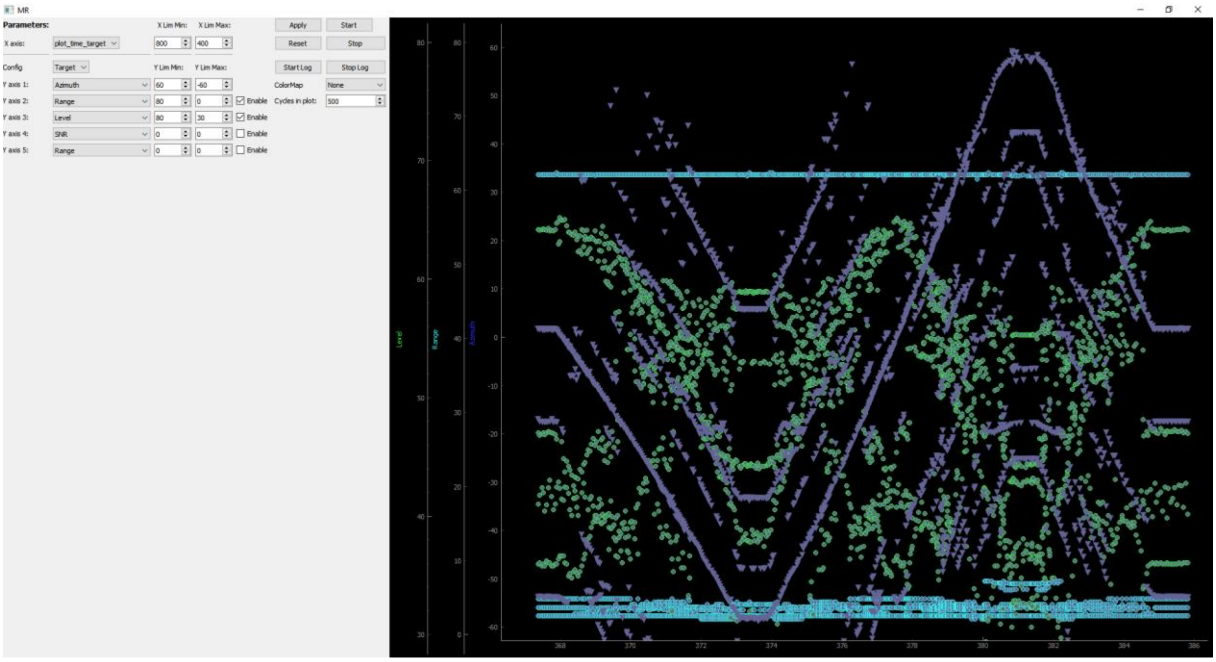
1. https://www.autocruise.com/produits/
2. https://www.zf.com/mobile/en/homepage/homepage.html
3. https://en.wikipedia.org/wiki/ZF_Friedrichshafen
4. https://www.anaconda.com/products/individual
5. https://pyqtgraph.readthedocs.io/en/latest/
6. https://matplotlib.org/contents.html
7. https://docs.python.org/fr/3/
8. https://doc.qt.io/qt-5/qtdesigner-manual.html
9. Datasheets and presentations about RADAR from Autocruise

14. Appendix:

14.1: Software architecture

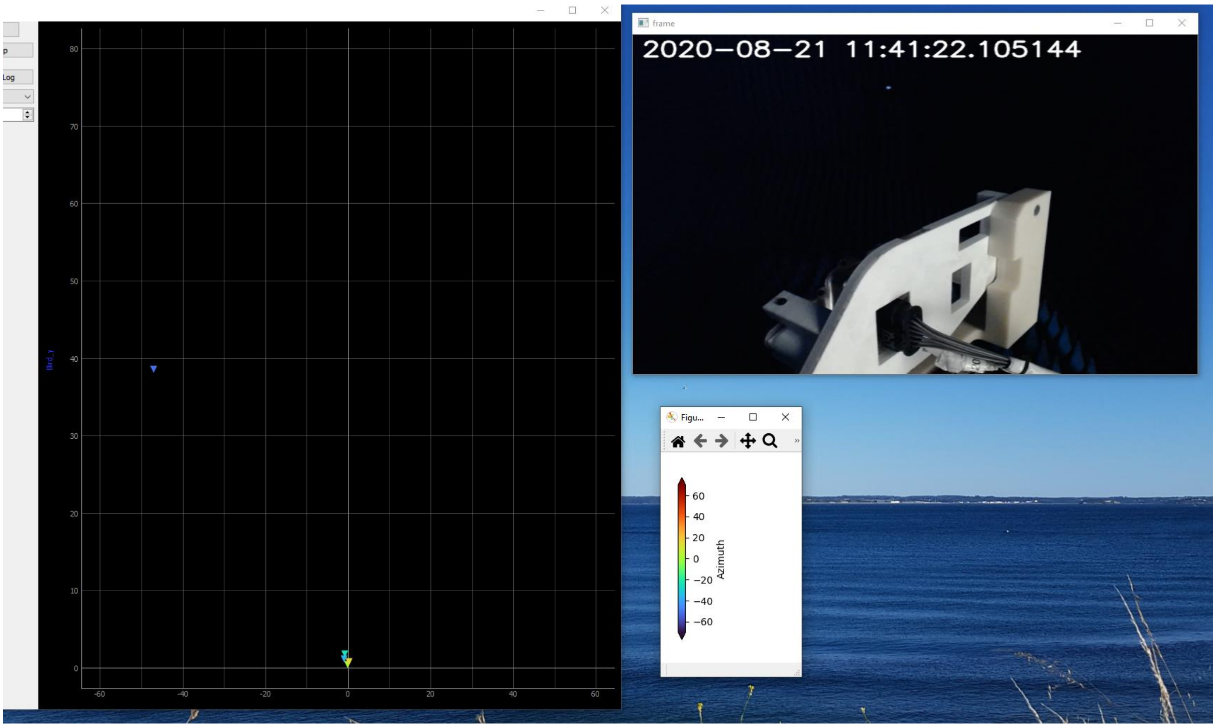


14.2 : Checking with multiple parameters with 500 previous cycles in screen

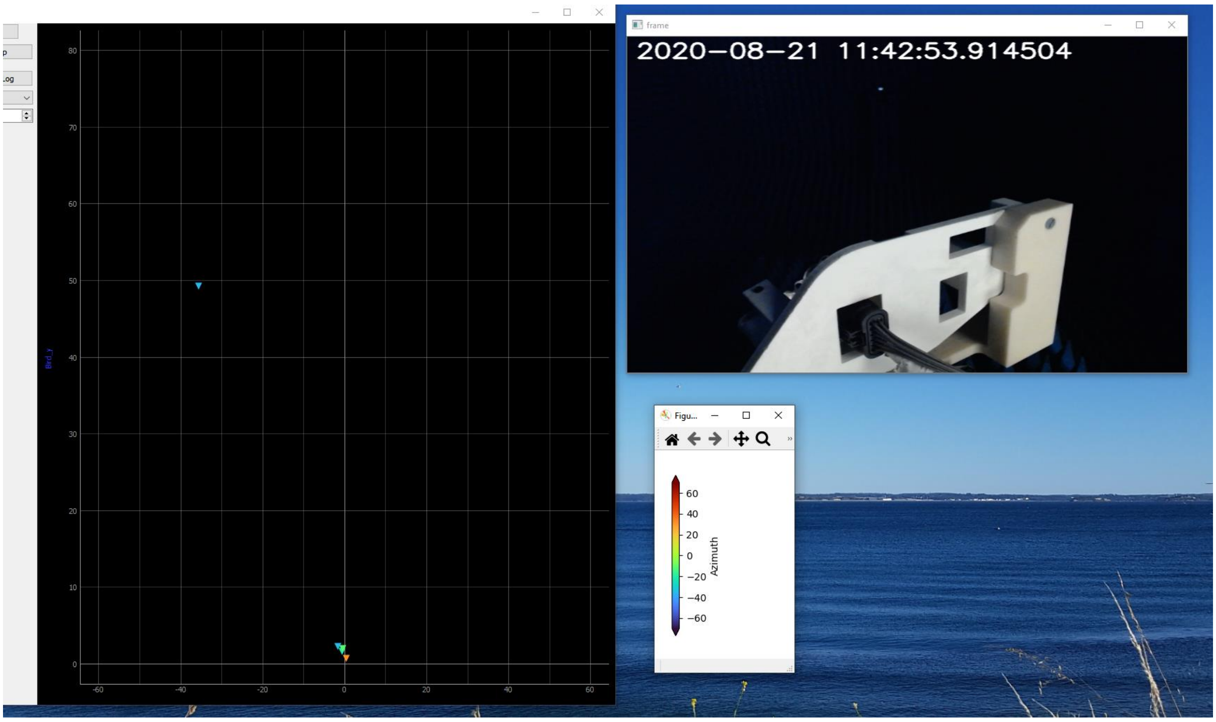


14.3. Bird view with colourmap

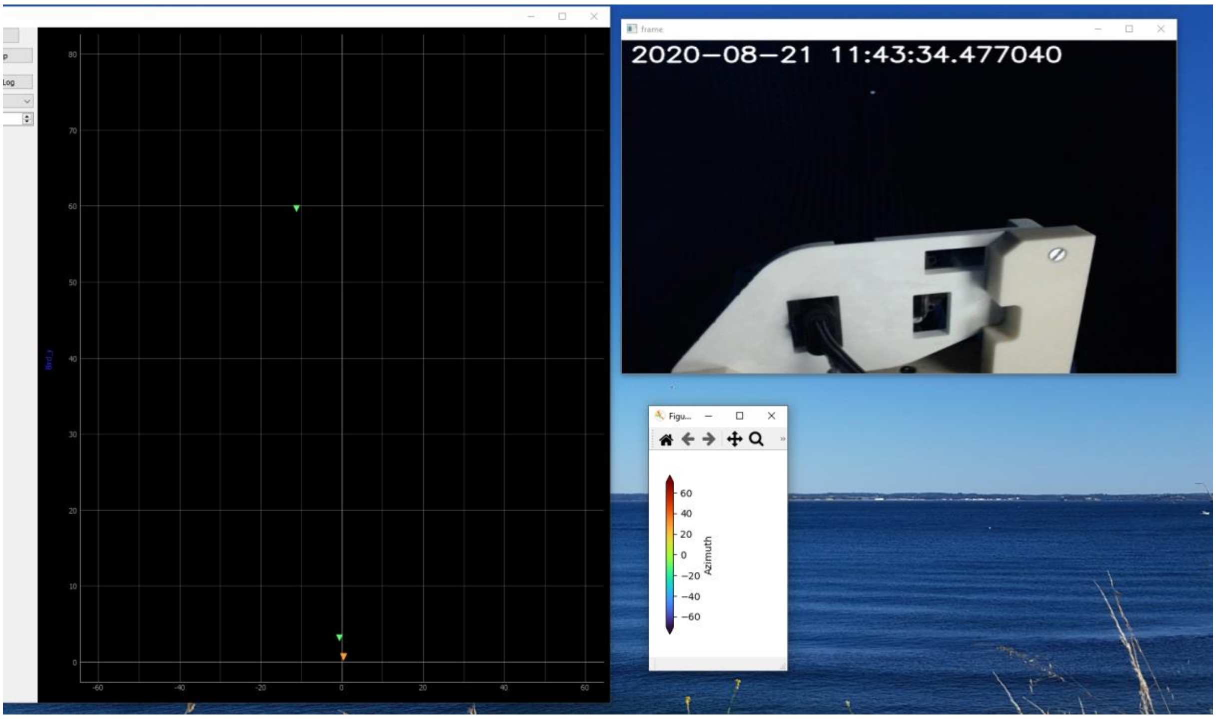
14.3.1 : Target = 61 meter, azimuth = -50°



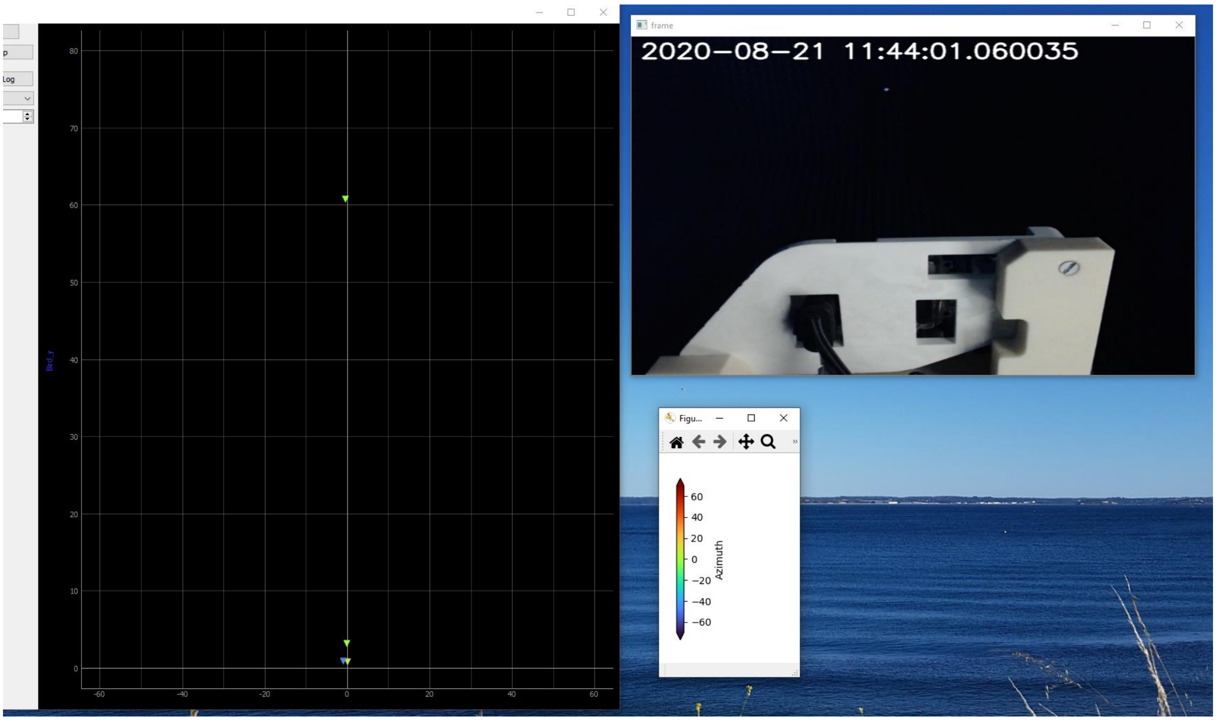
14.3.2 : Target = 61 meter, Azimuth = -35°



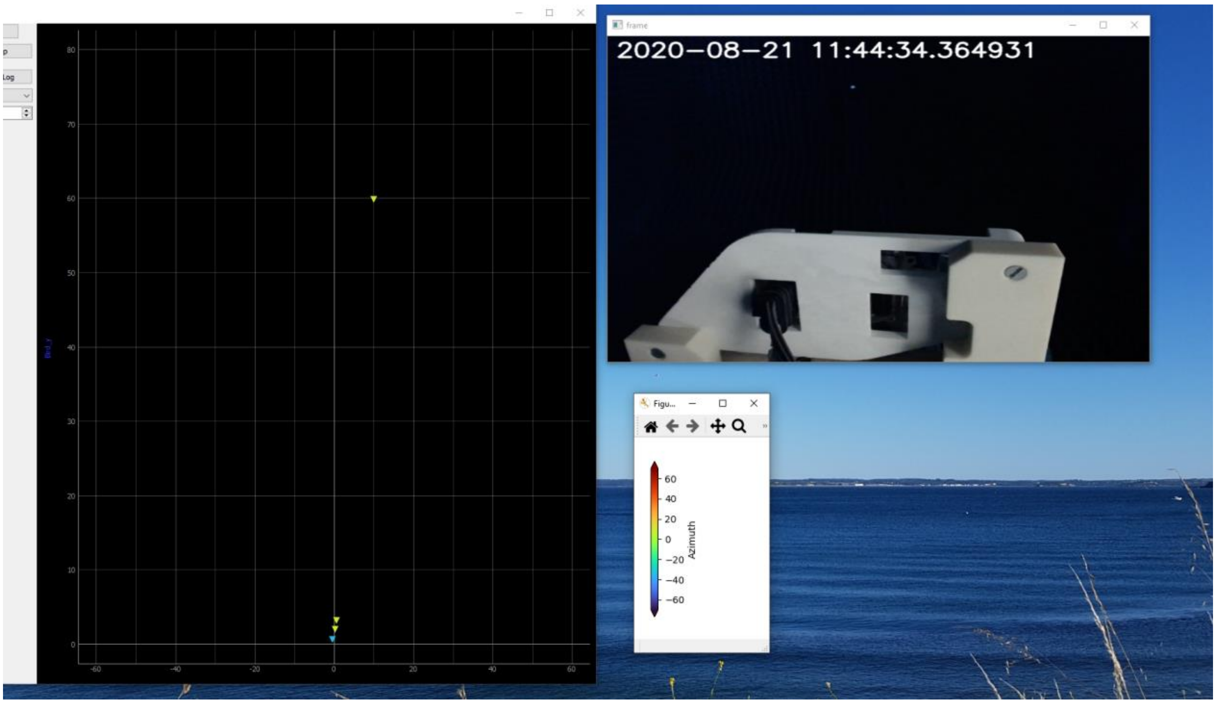
14.3.3 : Target = 61 meter, Azimuth = -10°



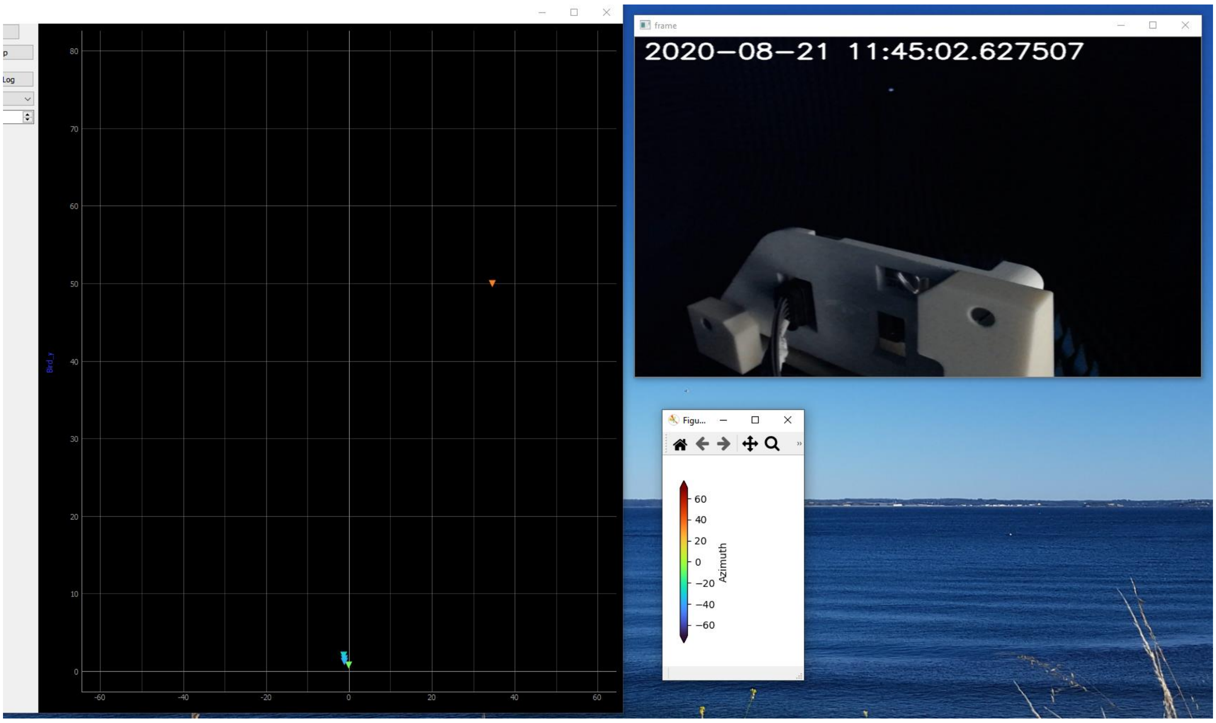
14.3.4 : Target = 61 meter, Azimuth = 0°



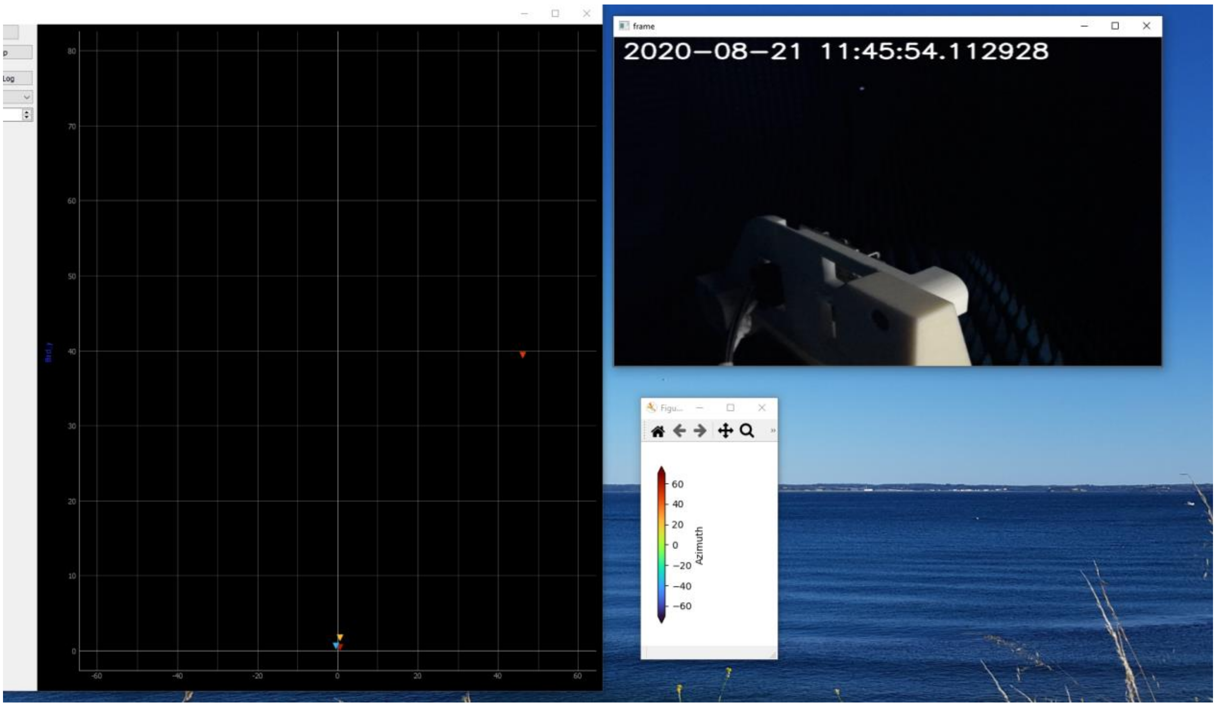
14.3.5 : Target = 61 meter, Azimuth = 10°



14.3.6 : Target = 61 meter, Azimuth = 35°



14.3.7 : Target = 61 meter, Azimuth = 50°



14.4: FR Real time checking in BirdVIEW