

# RADAR

## RADAR CONFIGURATION

### REQUIREMENTS

Name	Quantity	Description	Part Number
PCAN-USB-x6	1	Short-range RADAR interface board	IPEH-004064-0 02940
Cable Socket 8 pin Automotive Connector	24	Female Reverse Sensor connector for LR Radar	<u>1-1534229-1</u>
Yazaki HS025 WPC 8P	24	Female Reverse Sensor connector for SR Radar	<u>7298-2755-30</u>
Connector pins (0.25-0.35mm <sup>2</sup> )	500	Terminal	7116-4415-02
Single wire seal for above pin, blue	250	Single wire seal	7158-3165-90
Cavity seal, gray	250	Cavity seal	7158-3169-40
Connector pins (0.25-0.35mm <sup>2</sup> )	500	TE MQS contact	5-962885-1
Single wire seal for above pin, yellow	125	TE single wire seal	967067-2
Single wire seal for above pin, green	125	TE single wire seal	967067-1
Cavity seal, blue	250	For cavity diameter 3.45mm	967056-1
Technica gateway	1	Long range RADAR interface board	TE-1100

## SYSTEM SETUP

The PDK is built, developed and tested for Linux Ubuntu starting from version 18.04. Generally, it supports x86\_64 and arm64 architecture. It is recommended that the customer uses the same system setup to ensure a smooth operation of the PDK. For general compatibility matrices, refer to the page [page\\_system\\_overview](#).

## INSTALL COMMANDS

# Install Base PDK

\$ cd downlods/files/ (pdk-document file)

\$ tar -xavf pdk\_base\_0.7.5-4773fac-bionic\_amd64.tar.gz

\$ sudo apt install ./eCAL-5.9.1-Linux.deb

for ubuntu 20 find eCAL [https://eclipse-ecal.github.io/ecal/getting\\_started/setup.html](https://eclipse-ecal.github.io/ecal/getting_started/setup.html)

\$ sudo apt install ./codemeter-lite\_6.90.3691.500\_amd64.deb

\$ sudo apt install ./pdk\_base\_0.7.5-4773fac-bionic\_amd64.deb

## LONG RANGE RADAR

### TECHNICA GATEWAY PRECONFIGURATION (for new)

1. Assign the IP address 192.168.0.10 to your ethernet interface.
2. Open a web browser and access the media gateways config page on IP address 192.168.0.49.
3. Use the web interface to upload the media gateway config file  
/opt/pdk/etc/mediaGatewayConfig.cfg
4. Save and reset the media gateway using the web interface
5. To confirm radar required installation
  - /opt/pdk/bin/pdk\_monitoring\_tool

## CONNECTIONS

Note: every BroadR-Reach port on the MediaGateway supports up to 4 BroadR-Reach sensors. This means that sensor 1-4 has to be connected to BroadR-Reach ethernet port 1 whereas sensor 5-8 has to be connected to BroadR-Reach ethernet port 2, etc.

The BR+ port of the sensor has to be connected to the respective BR+ port on the MediaGateway.

The BR- port of the sensor has to be connected to the respective BR- port on the MediaGateway.

Blue connector on MediaGateway (blue cap with black insert), repeat for every required BroadR-Reach data port on the MediaGateway:

1: +12V

2: GND

3: BR Port 1 Negative (BR-)

4: BR Port 1 Positive (BR+)

6: BR Port 2 Positive (BR+)

7: BR Port 2 Negative (BR-)

10: BR Port 3 Positive (BR+)

11: BR Port 3 Negative (BR-)

17: BR Port 4 Negative (BR-)

18: BR Port 4 Positive (BR+)

## RADAR PINNING

1: BR Negative

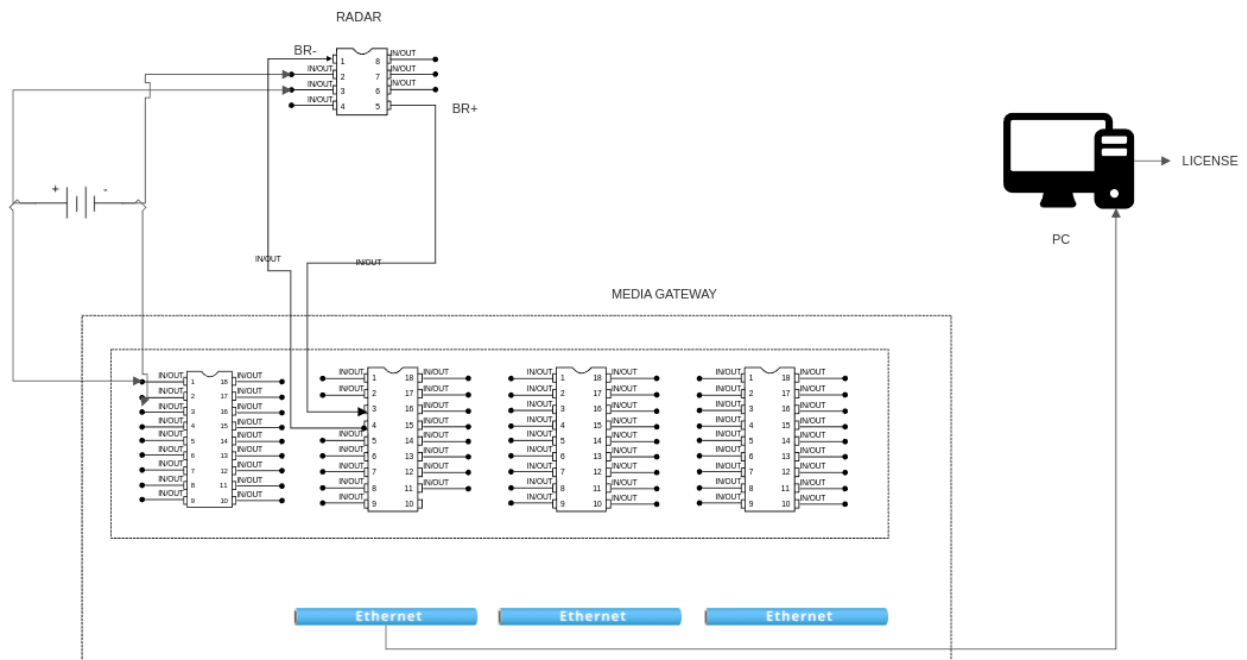
2: Ground

3: +12V

5: BR Positive



ARS430DI Radar Connector (Exemplary Illustration)



LINE DIAGRAM

### To see the radar output

/opt/pdk/bin/pdk\_start.sh

### mention radar id

/opt/pdk/etc/pdk\_config.json

### To edit pdk-config.json file

gedit /opt/pdk/etc/pdk\_config.json

example

```
{  
  "name": "ars_3",  
  "type": "ars430di",  
  "id": 22,  
  "mounting_parameters": {  
    "xyz": [1, 0, 0.5],  
    "pitch": 0,  
    "yaw": 0,  
    "orientation": "PLUG_RIGHT"  
  }  
}
```

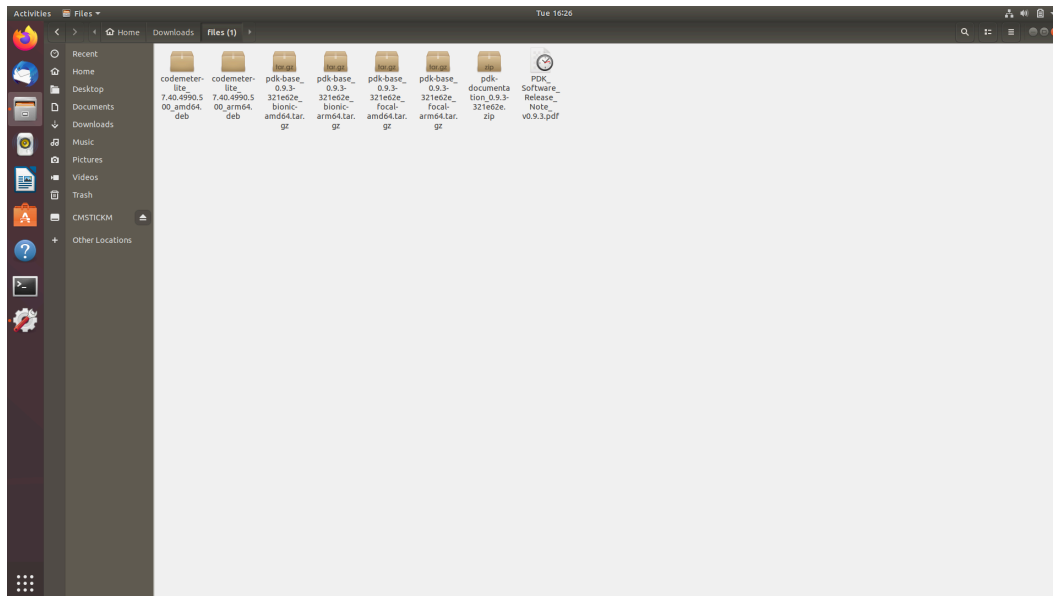
## Requirements

- ❖ For long
  - Technica media gateway
  - Long range radar
  - 12v power
  - Connectors
  - Ethernet cable
  - Pc
- ❖ For short
  - Pcan usb

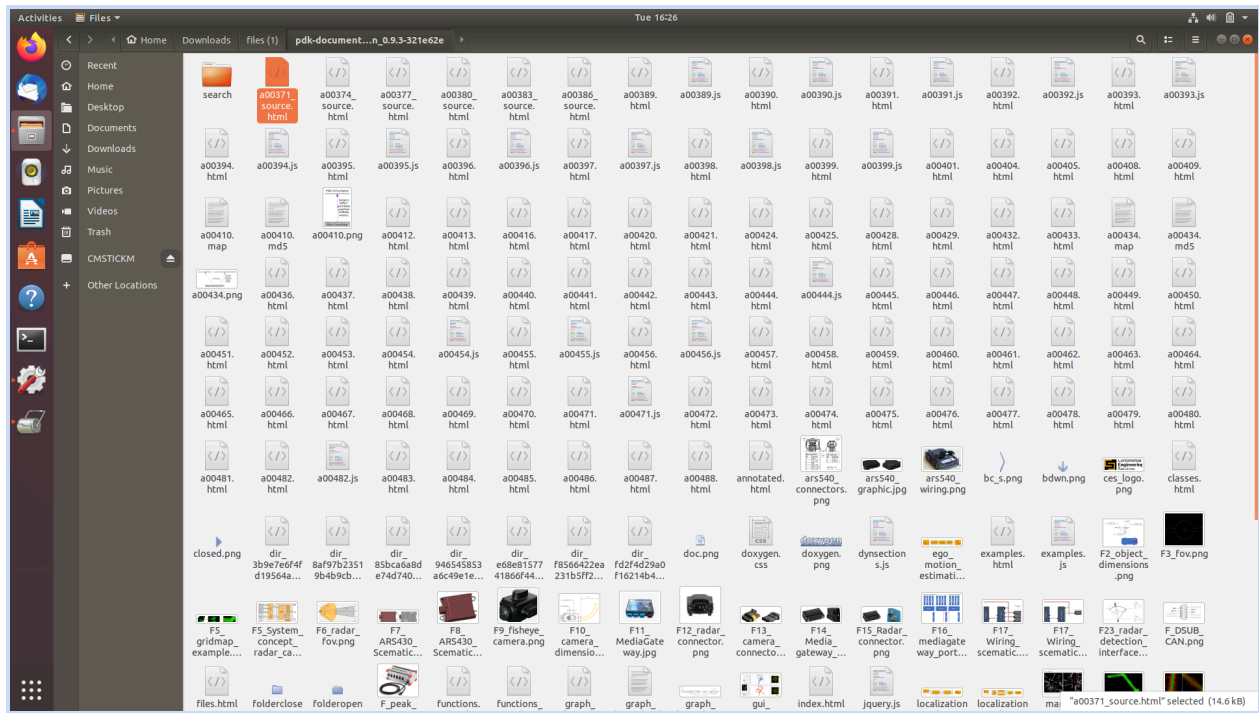
- Short range radar
- 12v power for both radar and pcan usb device
- Connectors
- Db9 connector
- Internet connection

## The following steps to be followed

1. Download and extract the pdk file
2. You'll be seen some zip files after extracting the pdk file



3. From the extracted the  
`/home/s186/Downloads/files/pdk-documentation_0.9.3-321e62e.zip`
4. pdk-documentation file in that go to Doc  
`/home/s186/Downloads/files/doc/a00371_source.html`



5. It will open a browse window it should contain entire information

Activities Firefox Web Browser Tue 16:27

Perception Development Kit 0.9.3-321e62e

file:///home/s186/Downloads/files (1)/pdk-documentation\_0.9.3-321e62e/a00445.html

**Perception Development Kit** 0.9.3-321e62e

In addition to raw RDI data the AR5430DI radar sensor provides tracked objects. For this purpose mounting parameters and vehicle dynamics data are mandatory to be provided to the sensor via the PDK. Sensor objects are tracked within the field of view of a single sensor only. Hence, in areas with overlapping sensors, multiple objects will be tracked per target object. These objects need to be fused in a further processing step that is not in scope of the PDK. For proper sensor fusion and 360 degree object tracking the PDK offers the Dynamic Object Tracking (DOT) software module.

### Time Synchronization

The AR5430DI sensor transmits timestamped data. The provided timestamps are either the local sensor MCU clock or the synchronized time via the PTP protocol, if available. As a time synchronization master, the ptpd and ptp4l implementations under Linux are reported to provide good results. For an example host-side PTP configuration see the corresponding section in the system setup chapter.

### Connectors and Pinning

#### Connectors

The PDK with the AR5430DI is delivered with a set of connectors with the following part numbers:

Name	Quantity	Description	Ordering Number
AR5430DI connector	1 per sensor	TE MQS Connector housing, 8 positions	1-1534229-1
Connector pins (0.25-0.35mm <sup>2</sup> )	2 per sensor	TE MQS contact	5-962885-1
Single wire seal for above pin, yellow	2 per sensor	TE single wire seal	967067-2
Connector pins (0.5-0.75mm <sup>2</sup> )	2 per sensor	TE MQS contact	5-965906-1
Single wire seal for above pin, green	2 per sensor	TE single wire seal	967067-1
Cavity seal, blue	4 per sensor	For cavity diameter 3.45mm	967056-1

#### Pinning

1: BR Negative  
2: Ground  
3: +12V  
5: BR Positive

**AR5430DI Radar Connector (Exemplary Illustration)**

© 2022 Continental Engineering Services

## 6. Find installation commands

Continental Engineering Services

# Perception Development Kit 0.9.3-321e62e

## Install commands

The PDK debian installers come in multiple archives. Copy all provided archives into a directory of your choice and open the directory in a terminal. Extract the delivered software installers and install the PDK packages:

```
# Install Base PDK
$ tar -xavf pdk_base_0.7.5-4773fac-bionic_amd64.tar.gz
$ apt install ./eCAL-5.9.1-Linux.deb ./codemeter-lite-6.90.3691.500_amd64.deb
$ apt install ./pdk_base_0.7.5-4773fac-bionic_amd64.deb

# Install optional PDK modules
$ tar -xavf pdk_dot_0.7.5-4773fac-bionic_amd64.tar.gz && apt install ./pdk_dot_0.7.5-4773fac-bionic_amd64.deb
$ tar -xavf pdk_ogg_0.7.5-4773fac-bionic_amd64.tar.gz && apt install ./pdk_ogg_0.7.5-4773fac-bionic_amd64.deb
$ tar -xavf pdk_eme_0.7.5-4773fac-bionic_amd64.tar.gz && apt install ./pdk_eme_0.7.5-4773fac-bionic_amd64.deb
$ tar -xavf pdk_mal_0.7.5-4773fac-bionic_amd64.tar.gz && apt install ./pdk_mal_0.7.5-4773fac-bionic_amd64.deb
```

Please note: The exact package names in the install commands above may vary with different PDK versions and need to be adapted to the packages you received.

The `apt install` method is preferred over `dpkg -i` because it does resolve package dependencies and also installs recommended packages necessary for the PDK GUI components. If you prefer a minimal headless installation, please add the `--no-install-recommends` flag to the above commands.

The installation creates the folder `/opt/pdk/` in the file system. The individual executables are located at `/opt/pdk/bin`.

## Setup Sensor HW

Depending on which sensors were purchased with the PDK different setups are used.

## BR Reach Ethernet

The sensors as well as the gateway should be powered from a 12V source. The current draw is approximately 0.5A for the gateway and a maximum of 1A per sensor. All sensors should be connected via a twisted-pair copper cable as required by the BroadR-Reach specification.

Please refer to the wiring information on the following pages:

- [ARS430DI Radar Sensor](#)
- [ARS548 Radar Sensor](#)
- [Fisheye Camera](#)
- [BroadR-Reach / Ethernet Gateway](#)

Up to 4 sensors per connector

© 2022 Continental Engineering Services

## 7. Use following commands from file location

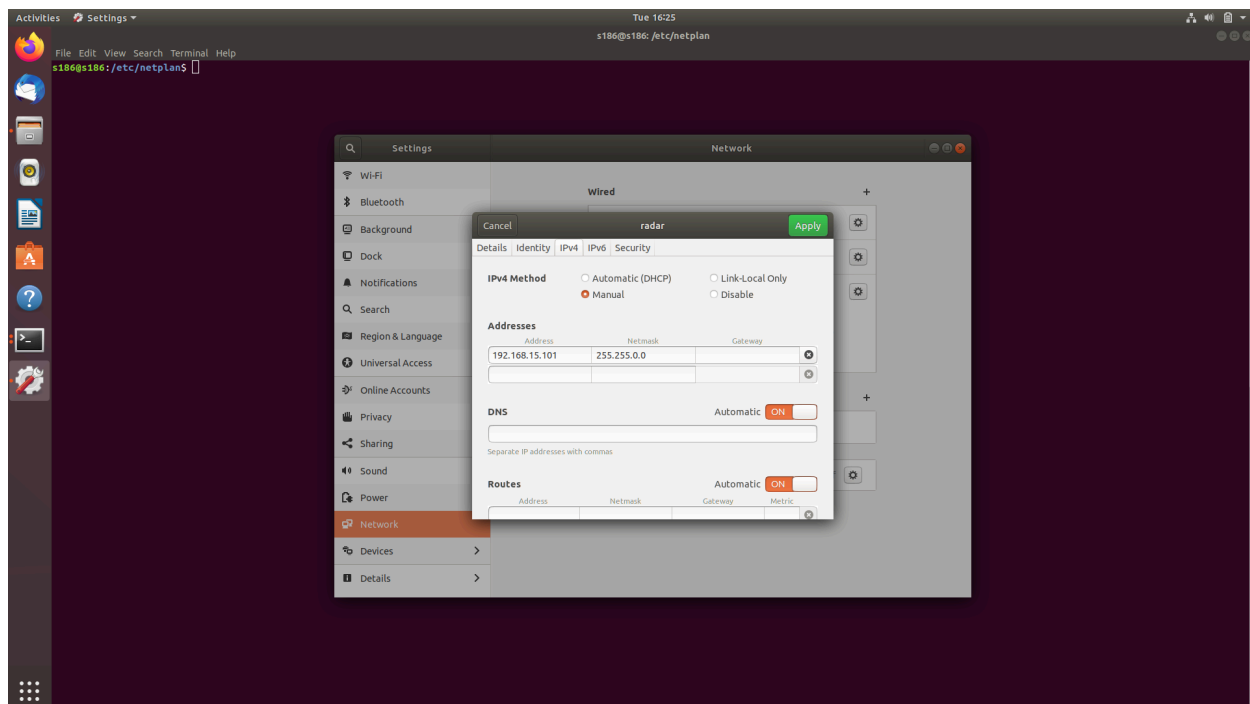
```
s186@s186:~/Downloads/files$ cd Downloads/files/
s186@s186:~/Downloads/files$
```



8. `tar -xvf pdk_base_0.7.5-4773fac-bionic_amd64.tar.gz`
9. `apt install ./eCAL-5.9.1-Linux.deb`
  - a. [https://eclipse-ecal.github.io/ecal/getting\\_started/setup.html](https://eclipse-ecal.github.io/ecal/getting_started/setup.html)
  - b. Enter below commands to install the required ecal requirements
  - c. `sudo apt update`
  - d. `sudo apt install libc6 libcurl4 libgcc1 libhdf5-100 libprotobuf10 libqt5core5a libqt5gui5 libqt5widgets5 libqt5svg5 libstdc++6 sysstat ifstat libqwt-qt5-6 libyaml-cpp0.5v5`

Note: According to the ubuntu version please select the ecal file.

10. `apt install ./codemeter-lite_6.90.3691.500_amd64.deb`
11. `apt install ./pdk_base_0.7.5-4773fac-bionic_amd64.deb`
12. After the installation change your system ip to
  - a. IP address: 192.168.15.101
  - b. Network mask: 255.255.0.0

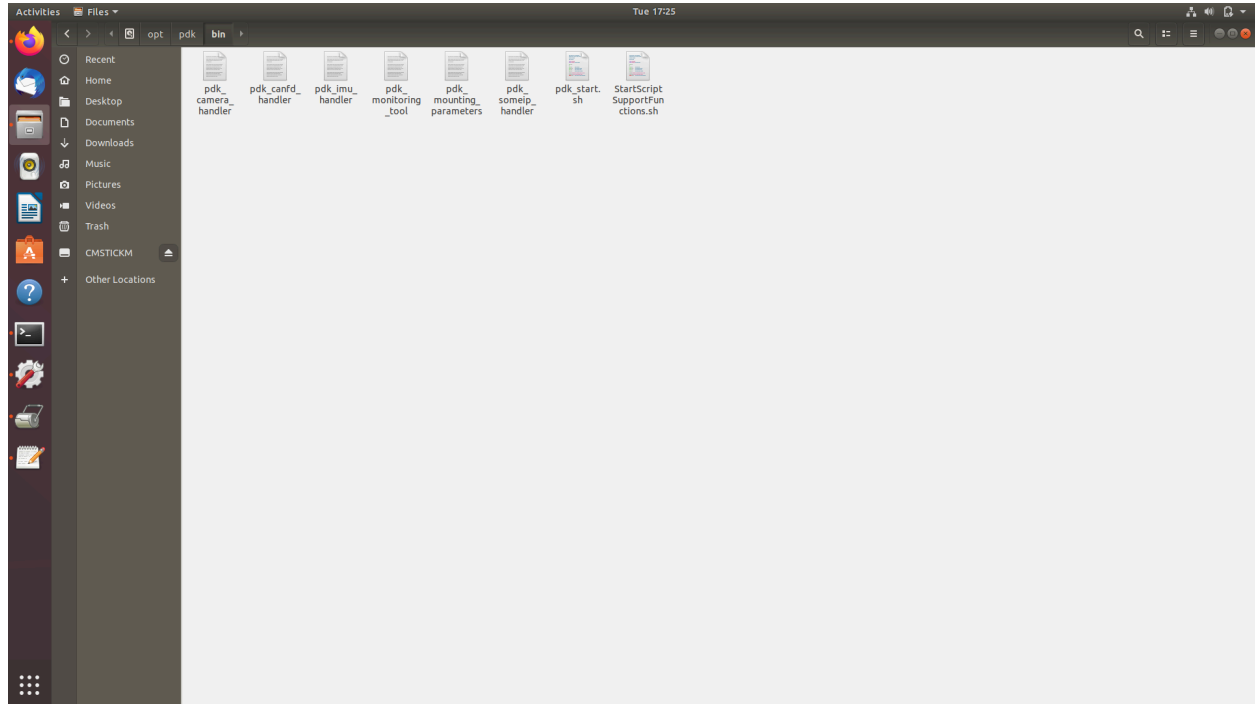


13. Also change the netplan yaml file
  - a. `Cd /etc/netplan`
  - b. `s186@s186:/etc/netplan$ sudo gedit 01-network-manager-all.yaml`
  - c. Now you're able to edit the file copy the below

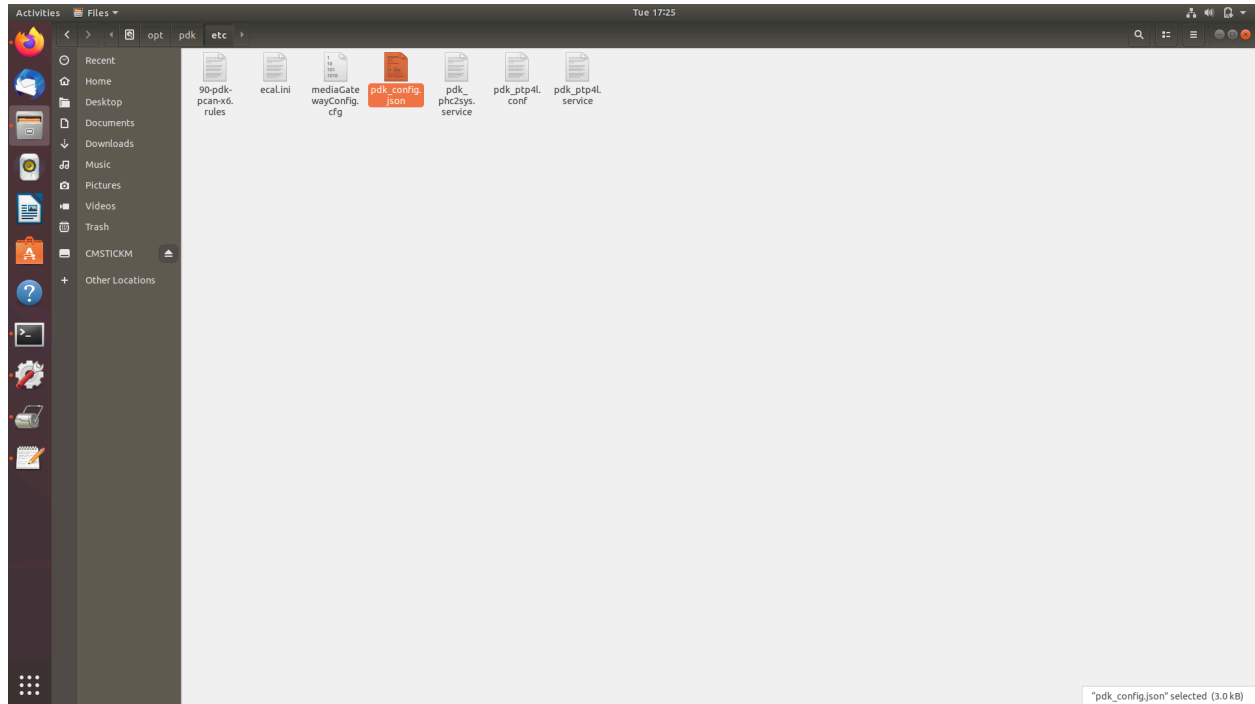
```
network:
  version: 2
  renderer: NetworkManager
  ethernets:
    eno1:
      dhcp4: no
      addresses: [192.168.15.101/16]
```

14. Ensure internet connection should be required

15. After the installation completed the pdk file will be created in opt/pdk inside pdk find the bin



16. opt/pdk/etc



- a. We need to place the sensor ID in the shown json file for that use below command

- i. Sudo gedit pdk\_config.json
- ii. And place the ID like below

```
// -----
//
// PDK Configuration sample file to be adapted by customer
//
// -----
{
  "radar_sensors": [

    {
      "name": "srr_front_left",
      "type": "srr520di",
      "id": 0,
      "can_channel": "can0",
      "mounting_parameters": {
        "xyz": [0.8, 0.8, 1.2],
        "yaw": 1.37,
        "x_cog": 2.8,
        "orientation": "PLUG_DOWN"
      }
    },
    {
      "name": "srr_front_left",
```

```

    "type": "srr520di",
    "id": 1,
    "can_channel": "can1",
    "mounting_parameters": {
        "xyz": [0.8, 0.8, 1.2],
        "yaw": 1.37,
        "x_cog": 2.8,
        "orientation": "PLUG_DOWN"
    }
},
{ "name": "srr_front_left",
  "type": "srr520di",
  "id": 2,
  "can_channel": "can2",
  "mounting_parameters": {
      "xyz": [0.8, 0.8, 1.2],
      "yaw": 1.37,
      "x_cog": 2.8,
      "orientation": "PLUG_DOWN"
  }
},
{
  "name": "ars_1",
  "type": "ars430di",
  "id": 20,
  "mounting_parameters": {
      "xyz": [1, 0, 0.5],
      "pitch": 0,
      "yaw": 0,
      "orientation": "PLUG_RIGHT"
  }
},
{
  "name": "ars_2",
  "type": "ars430di",
  "id": 21,
  "mounting_parameters": {
      "xyz": [1, 0, 0.5],
      "pitch": 0,
      "yaw": 0,
      "orientation": "PLUG_RIGHT"
  }
},
{

```

```

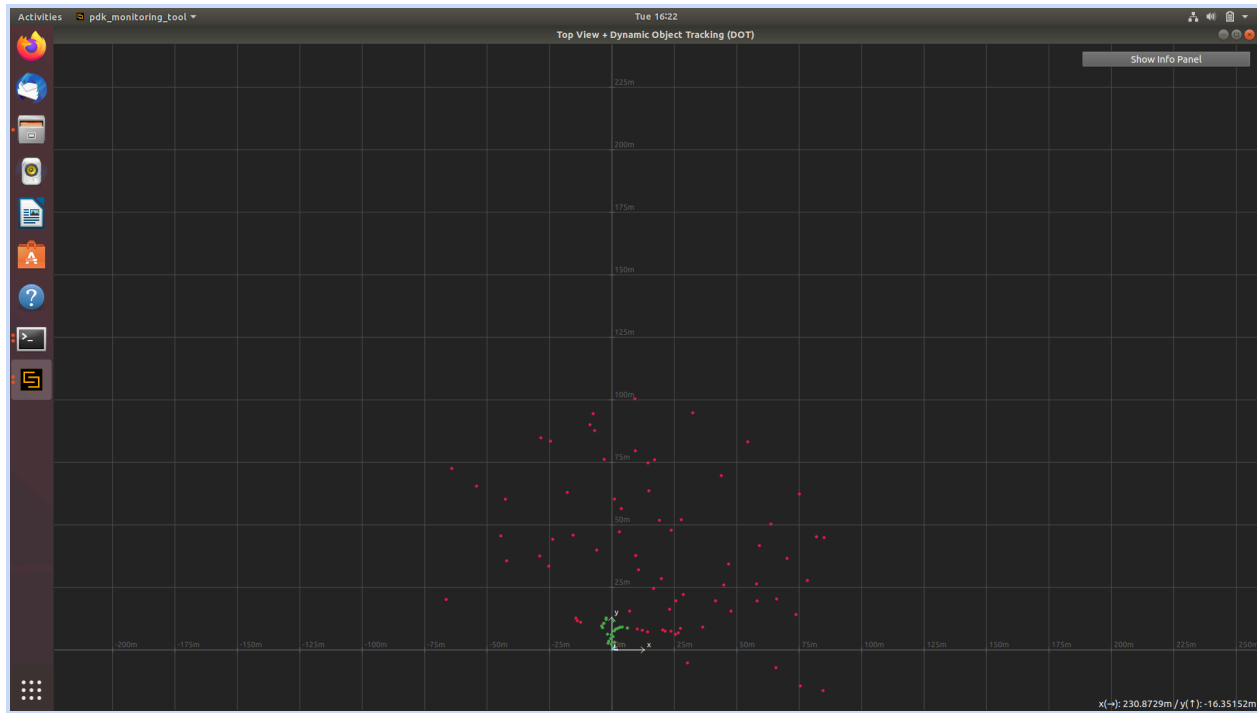
        "name": "ars_3",
        "type": "ars430di",
        "id": 22,
        "mounting_parameters": {
            "xyz": [1, 0, 0.5],
            "pitch": 0,
            "yaw": 0,
            "orientation": "PLUG_RIGHT"
        }
    },
    ],
    "camera_sensors": [
        {
            // Optional: sensor description
            "name": "bcam_front",
            // Camera type [bcam, svc215]
            "type": "bcam",
            // Sensor id to be published with camera images
            "id": 0,
            // Stream id for respective camera, given by hardware
            // [bcam]: "192.168.100.1:<port>" where <port> is in range [5000 ..
5003]
            // [svc215]: e.g. "001a371f80011001", "001a371f80021002",
            "001a371f80031003",...
            "stream_id": "192.168.100.1:5000",
            "topic_name_id": "svc_front"
        },
        {
            "name": "bcam_rear",
            "type": "bcam",
            "id": 1,
            "stream_id": "192.168.100.1:5001",
            "topic_name_id": "svc_rear"
        }
    ],
    "localization": {
        "map_directory": "/path/to/map"
    }
}

```

17. Open `opt/pdk/bin/` in terminal and enter `./pdk_mounting` for confirming successful installation
18. For checking the long range radar data
  - a. `./pdk_start.sh`

19. For short range radar

- a. `sudo ip link set can0 up type can bitrate 500000 dbitrade 2000000 fd on`
- b. Then run `./pdk_start.sh`
- c. The data will be shown like below



## Experiment

Radar sensor output data Comparison according to place of radar fixed

## Requirements

- ❖ For long
  - Technica media gateway
  - Long range radar
  - 12v power
  - Connectors
  - Ethernet cable
  - Pc
- ❖ For short
  - Pcan usb
  - Short range radar
  - 12v power for both radar and pcan usb device
  - Connectors

- Db9 connector
- Internet connection

## Theory

RADAR technology in autonomous vehicles operates with millimeter waves and offers millimeter precision. The utilization of millimeter waves in autonomous vehicular RADAR **ensures high resolution in obstacle detection and centimeter accuracy in position and movement determination.**

For utilizing complete radar data the place of the radar should be the key factor. this experiment will focus on behavior of the radar sensor while placing it in enclosed (inside the bonnet) and free space (outside of the body).

## Procedure

### 1. Sensor placed outside the car

As shown in the image(1), we fixed a long range sensor outside of the car with a double side plaster. and connections are given as per the document provided by the radar manufacturer.



Image : 1

## 2. Sensor placed inside the car bonnet

As shown in the image (2) we have placed the sensor inside the car bonnet with double side plaster.and connections are given as per the document provided by the radar manufacturer.



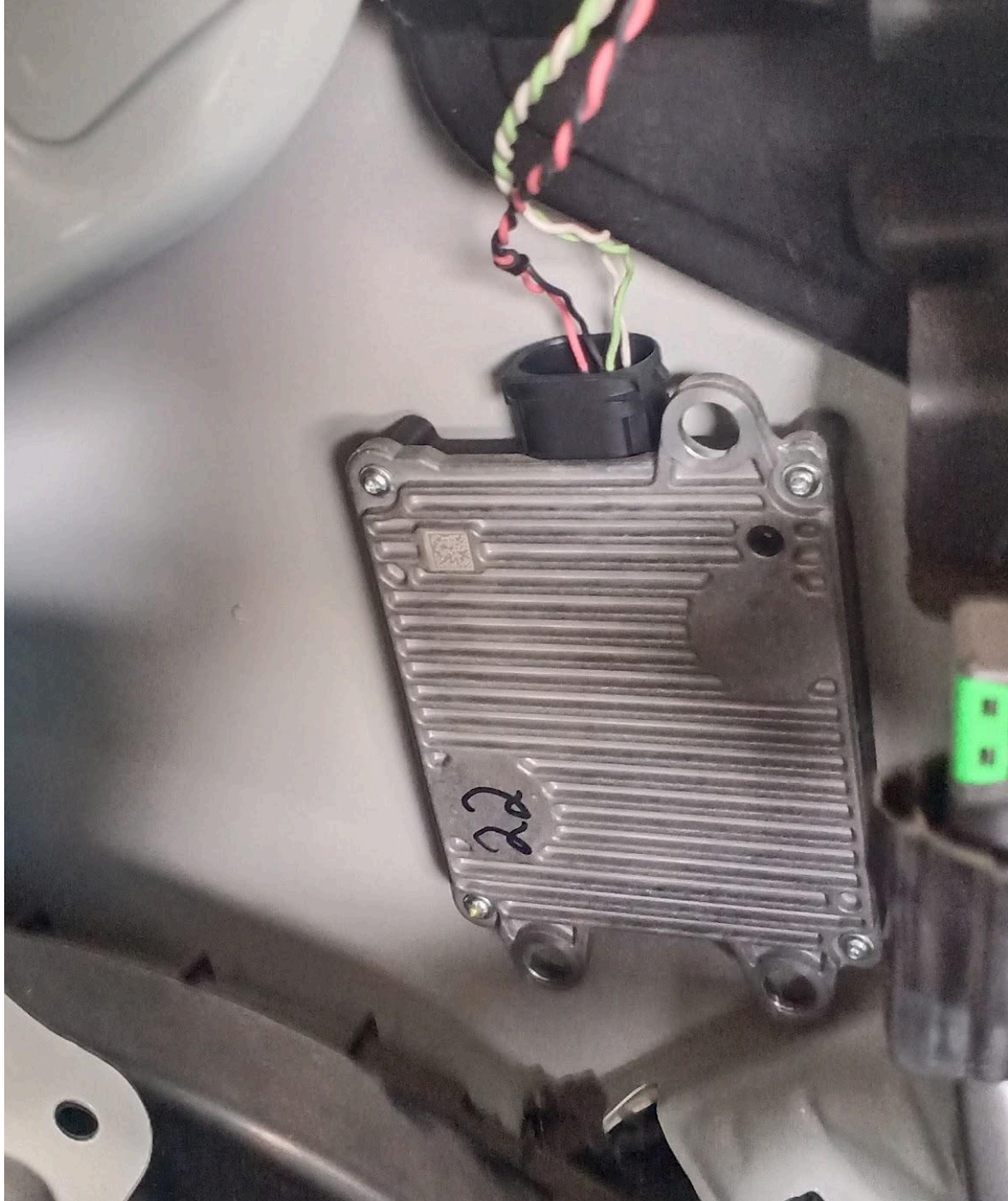


Image : 2

## Results

- In the first case we can observe from image(3), radar can able to detect objects up to its mentioned range 200 meters with efficient cloud points

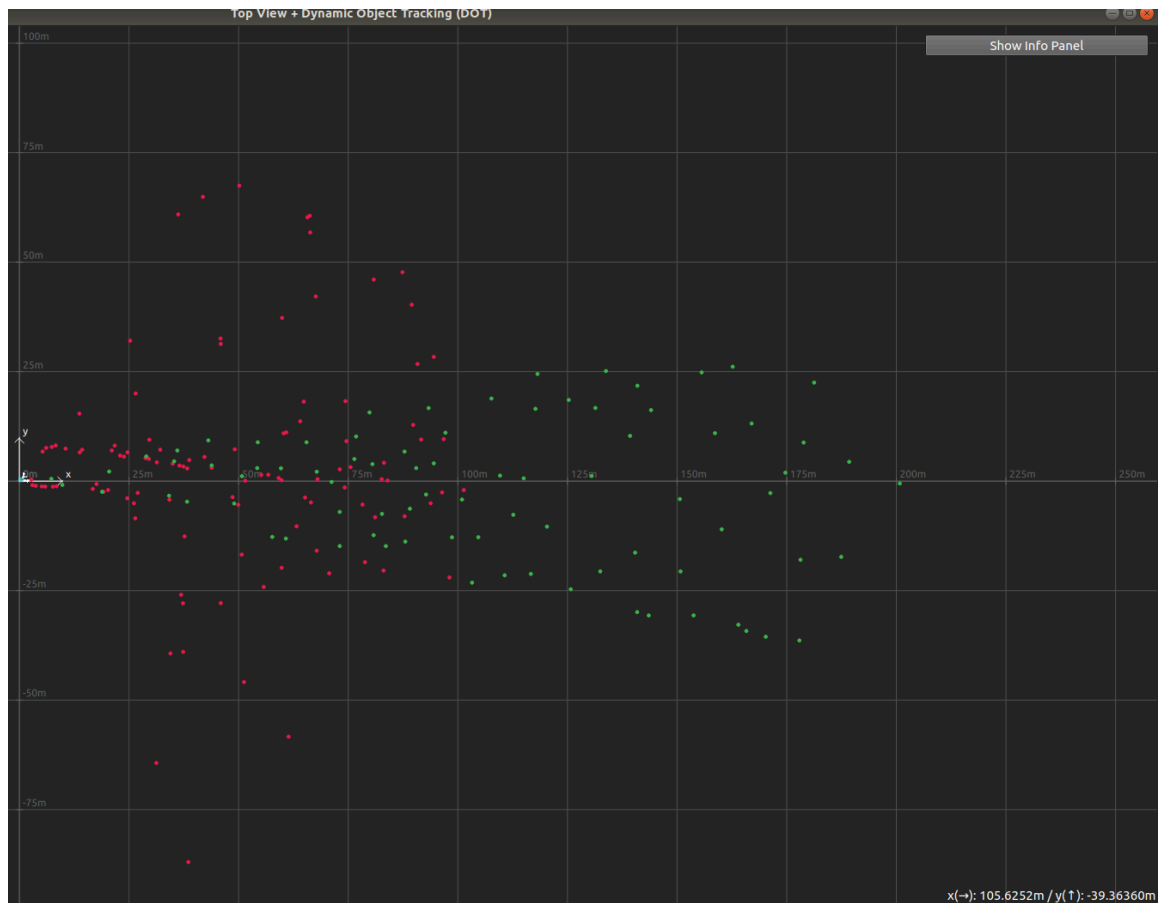


image:3

- In the second case from the image (4), we can see that the radar wont detected any objects because of its closed surface area

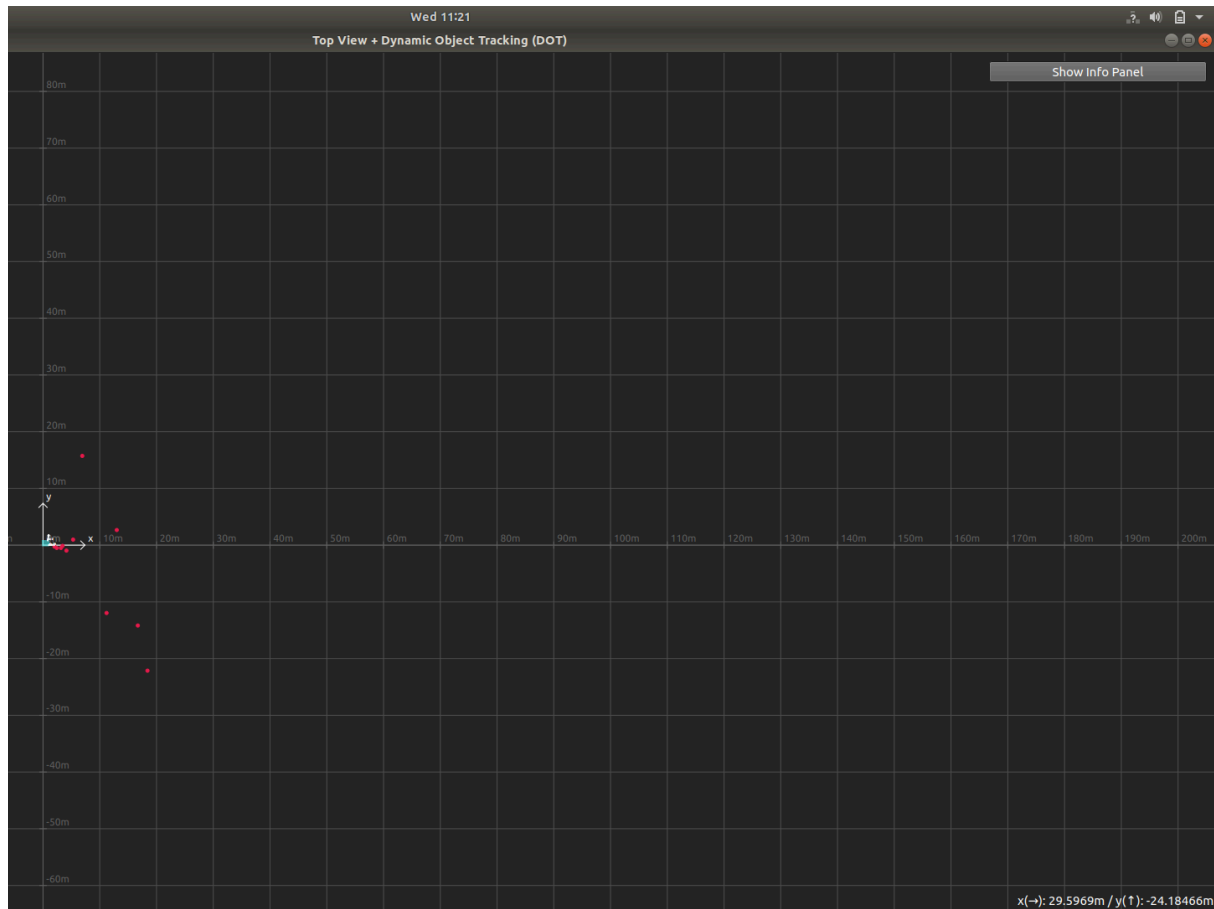


image:4

## Precautions

1. Ensure that connections should be proper. avoid loose connections
2. Use proper connectors while connecting the sensor with power and data connections
3. Take the power 12v. If you're using the battery, check the power continuously. Because of power issues you may not get results.
4. Internet connection should be required.

## Conclusion

1. The radar sensor surface should be placed in an open place.

2. If we place the sensor inside a closed place (inside car bonnet) sensor considers it like a obstacle.it can't project behind that closed obstacle

## Sensor fusion

Environment creation:

1. python3 -m venv fusion
2. source fusion/bin/activate

Pypylon library installation:

1. git clone <https://github.com/basler/pypylon.git>
2. cd pypylon
3. pip install .

Yolov8 requirements:

kiwisolver, cyclor, pillow, packaging, contourpy, zipp, importlib-resources, fonttools, matplotlib, pytz, tzdata, pandas, seaborn, scipy, filelock, mpmath, sympy, networkx, MarkupSafe, jinja2, typing-extensions, fsspec, torch, thop, idna, urllib3, charset-normalizer, certifi, requests, torchvision, py-cpuinfo, tqdm, psutil, ultralytics

Successfully installed MarkupSafe-2.1.5 certifi-2024.2.2 charset-normalizer-3.3.2 contourpy-1.1.1 cyclor-0.12.1 filelock-3.13.3 fonttools-4.51.0 fsspec-2024.3.1 idna-3.6 importlib-resources-6.4.0 jinja2-3.1.3 kiwisolver-1.4.5 matplotlib-3.7.5 mpmath-1.3.0 networkx-3.1 packaging-24.0 pandas-2.0.3 pillow-10.3.0 psutil-5.9.8 py-cpuinfo-9.0.0 pytz-2024.1 requests-2.31.0 scipy-1.10.1 seaborn-0.13.2 sympy-1.12 thop-0.1.1.post2209072238 torch-2.2.2 torchvision-0.17.2 tqdm-4.66.2 typing-extensions-4.11.0 tzdata-2024.1 ultralytics-8.1.43 urllib3-2.2.1 zipp-3.18.1

1.\$ roscore

2./opt/pdk/bin\$ ./pdk\_start.sh

**PDK Monitoring Tool**

Add Visualization Widget   Select Color Theme   Open Documentation   Show as List

### Sensors

ID: 0   Type: SRR520DI	ID: 1   Type: SRR520DI	ID: 2   Type: SRR520DI	ID: 20   Type: ARS430DI
ID: 21   Type: ARS430DI	ID: 23   Type: ARS430DI	ID: 25   Type: ARS430DI	ID: 0   Type: BCAM
ID: 1   Type: BCAM	Type: IMU SC23   6DOF		

### Input Signals

Vehicle Dynamics	Mounting Parameters	GNSS	DGNSS
------------------	---------------------	------	-------

### Algorithm Data

Object Tracking	Occupancy Grid	Freespace	Localization
-----------------	----------------	-----------	--------------

Sensor ID

Timestamp [s]

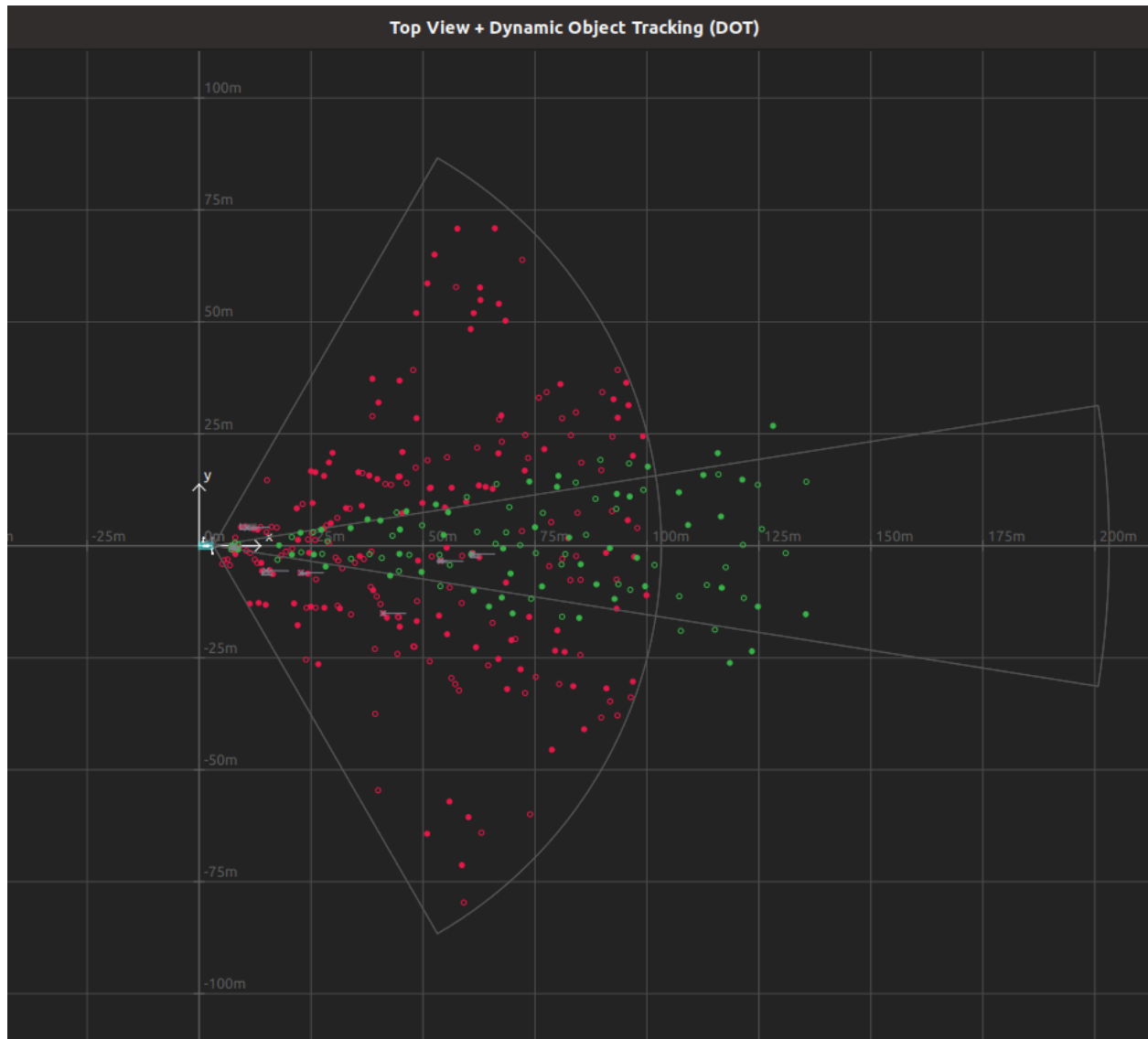
Timestamp [ns]

Sync State

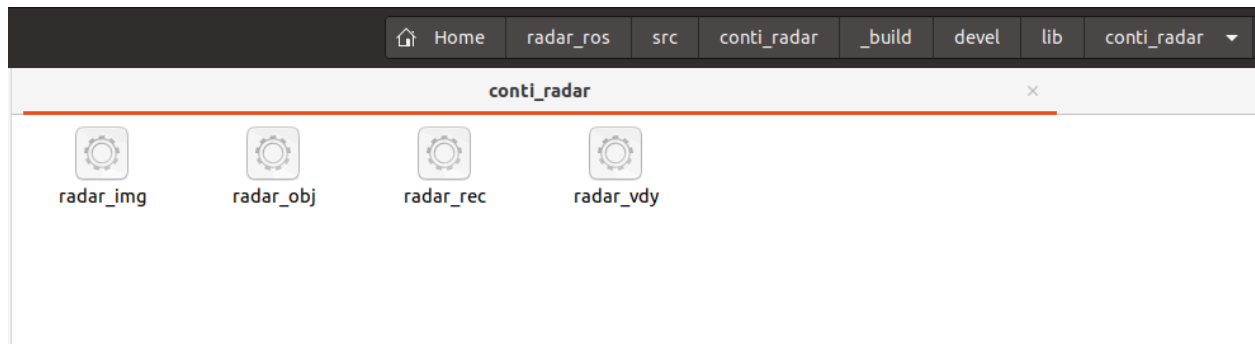
Software Version

Mounting Parameters

Longitudinal Position [m]	2.8
Lateral Position [m]	0.8
Vertical Position [m]	0.43
Pitch Angle [rad]	0
Yaw Angle [rad]	1.37



3.~/radar\_ros/src/conti\_radar/\_build/devel/lib/conti\_radar\$ ./radar\_obj



[illegible]

```
4.~/radar_ros/src/conti_radar/_build/devel/lib/conti_radar$ ./radar_vdy
```

[illegible]

```
5.(fusion) orin@ubuntu:~/basler_v8$ python3 bas-pub.py
```



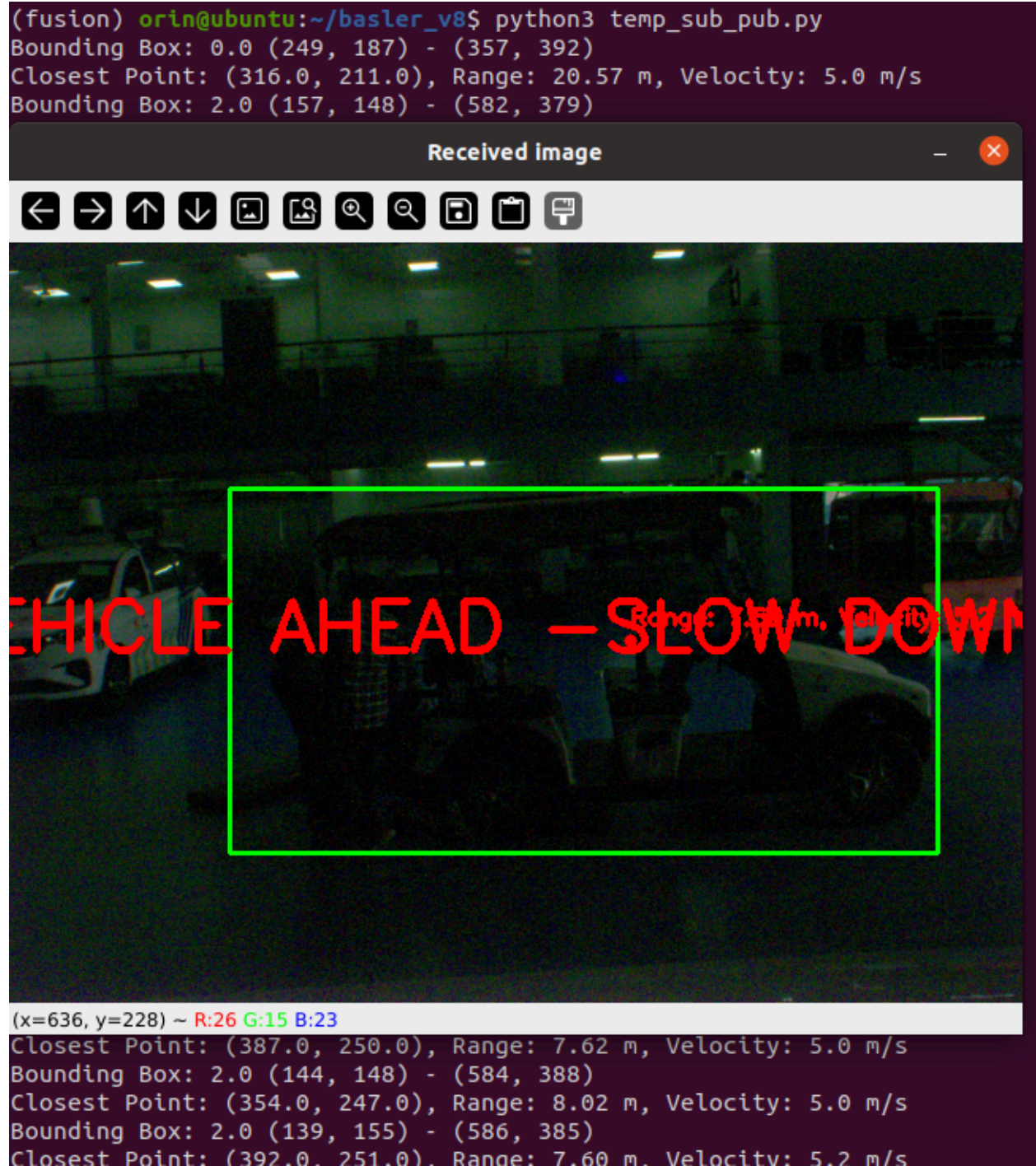
```
img_msg.data = cv_image.tostring()

9: 480x640 3 persons, 1 car, 1 bus, 1 traffic light, 1179.9ms
Speed: 5.3ms preprocess, 1179.9ms inference, 3.4ms postprocess per image at shape (1, 3, 480, 640)
bas-pub.py:59: DeprecationWarning: tostring() is deprecated. Use tobytes() instead.
img_msg.data = cv_image.tostring()

Image
← → ↑ ↓ 🖼️ 🔍 📄 🗑️ 🖨️
ge at shape (1, 3, 480, 640)
e at shape (1, 3, 480, 640)
e at shape (1, 3, 480, 640)
ge at shape (1, 3, 480, 640)
ge at shape (1, 3, 480, 640)
e at shape (1, 3, 480, 640)
e at shape (1, 3, 480, 640)
e at shape (1, 3, 480, 640)
ge at shape (1, 3, 480, 640)
ge at shape (1, 3, 480, 640)
(x=636, y=74) ~ R:14 G:40 B:22
9: 480x640 1 person, 1 car, 1 bus, 1 traffic light, 1325.0ms
Speed: 9.7ms preprocess, 1325.0ms inference, 2.3ms postprocess per image at shape (1, 3, 480, 640)
```

- 6.(fusion) orin@ubuntu:~/basler\_v8\$
- source/home/orin/radar\_ros/src/conti\_radar/\_build/devel/setup.bash
- 7.(fusion) orin@ubuntu:~/basler\_v8\$ python3 temp\_sub\_pub.py





For cuda pytorch and torchvision installations

[https://docs.google.com/document/d/1R56QsxFw\\_WxeJUKjWA\\_8POHq8MICPZeUY6lguWTg8io/edit?usp=sharing](https://docs.google.com/document/d/1R56QsxFw_WxeJUKjWA_8POHq8MICPZeUY6lguWTg8io/edit?usp=sharing)