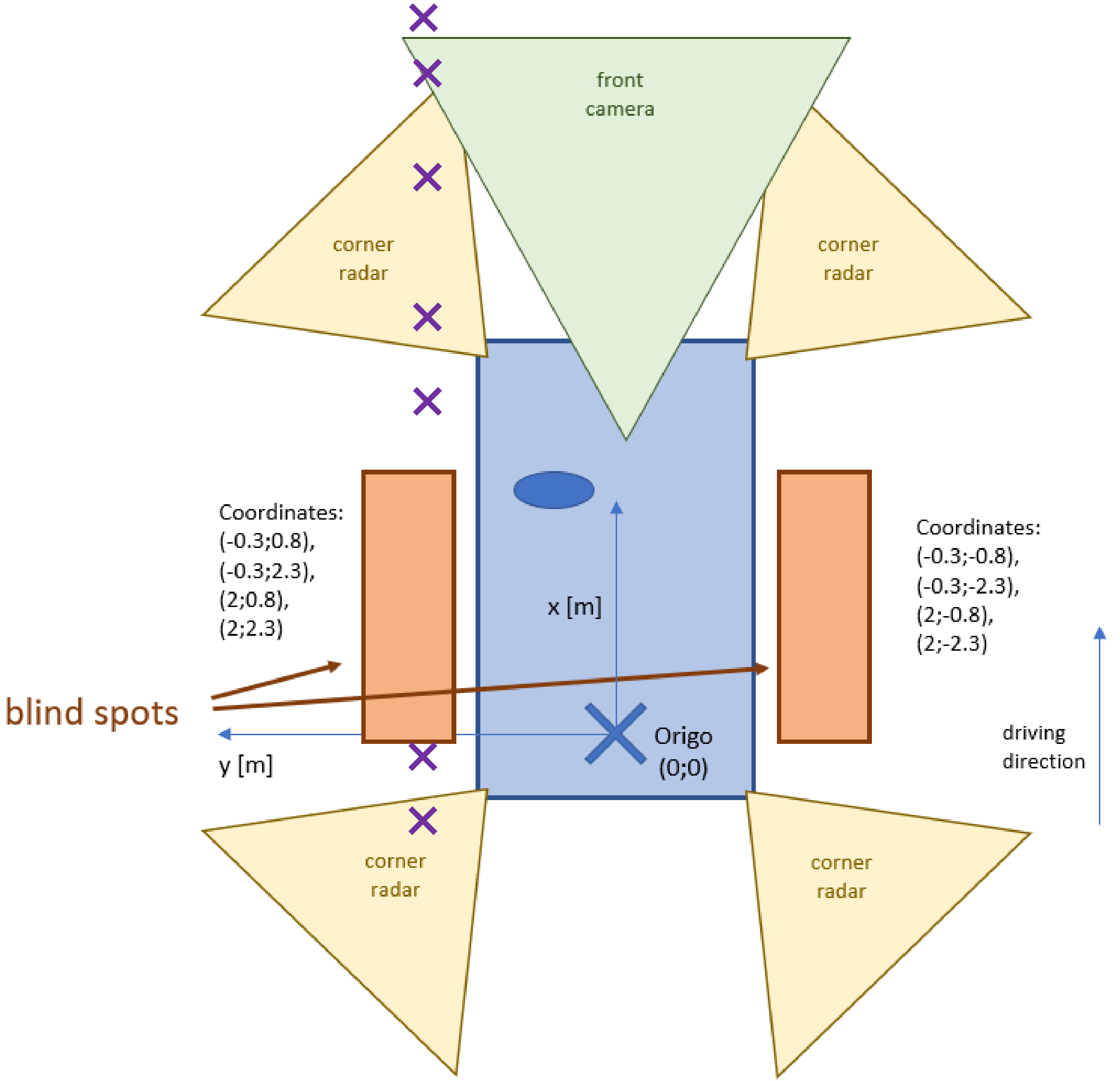
CODE #LIKEABOSCH

SW CHALLENGE 2022

ÁDÁM WITTMANN, 30.09.2022

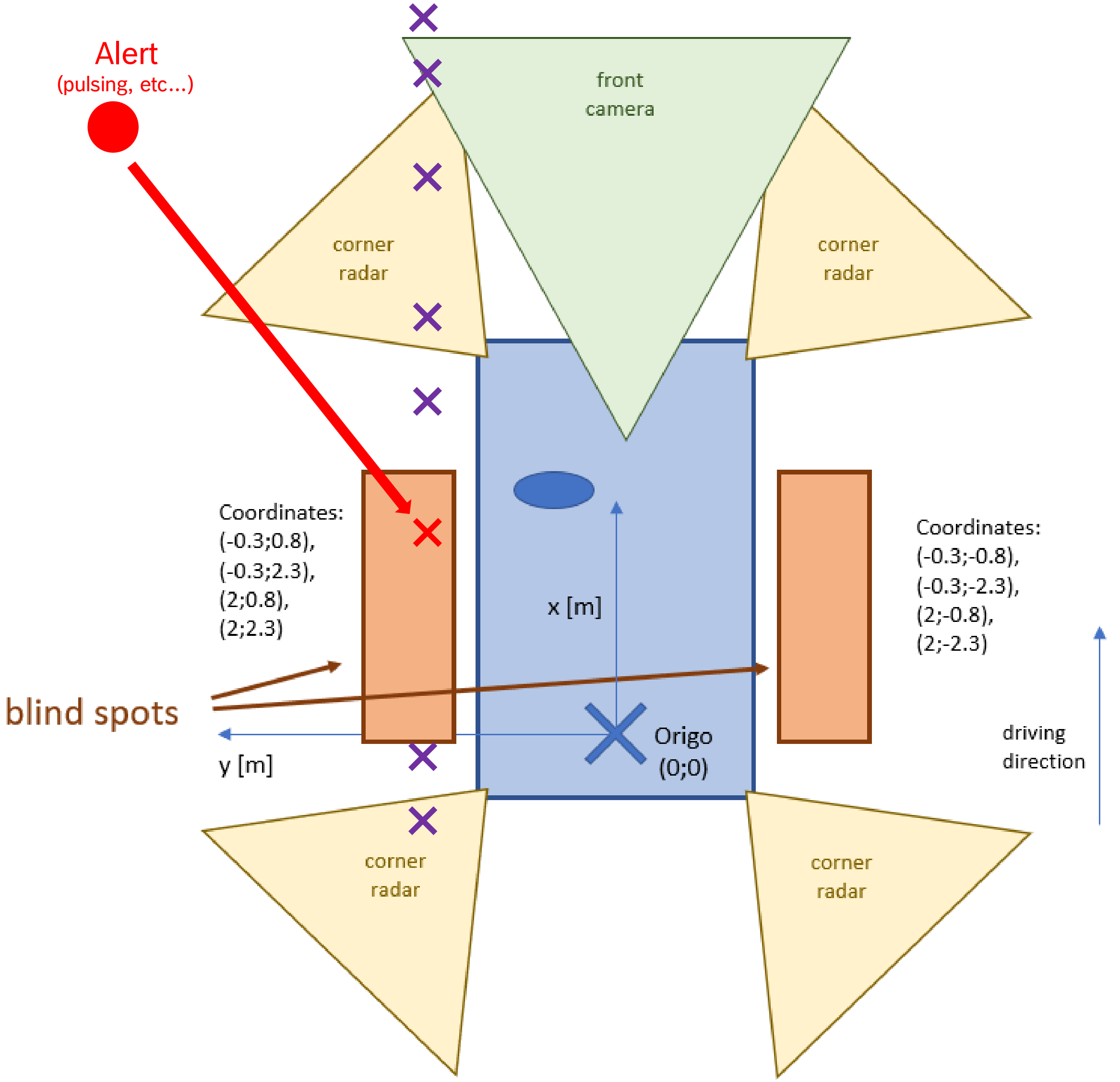
Code #LikeABosch
The Challenge – Part 1

- Part 1: Object tracking
  - Create your own estimated objects around the vehicle
  - Track these objects over time, update the estimated object information from timestamp to timestamp
  - If an object is out of the sensors' field of view, track it further based on your estimations
  - If an object is seen by multiple sensors, you could „merge" the two sensor inputs to one common estimated object
    - Absolute distance based
    - Kalman-filter based (more advanced)
    - Etc…



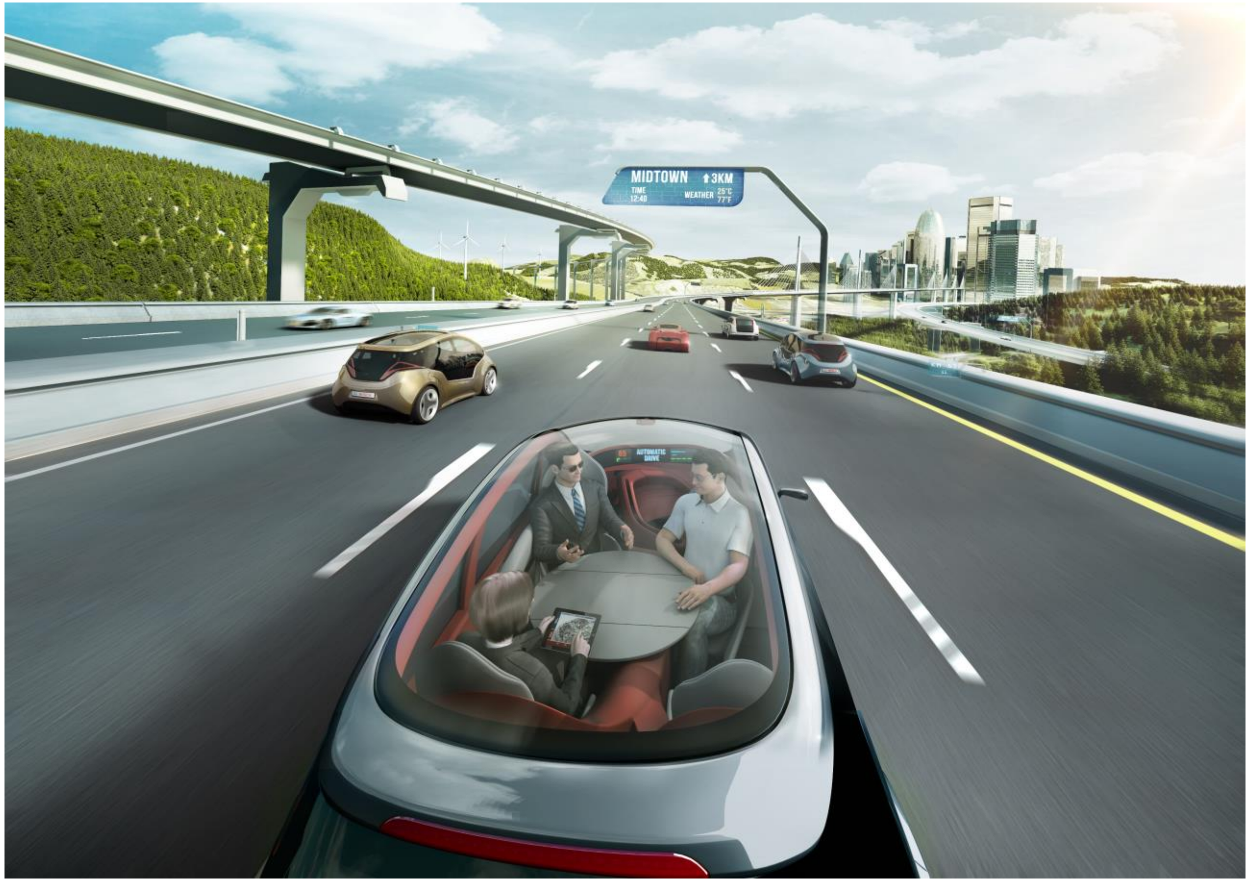
Code #LikeABosch
The Challenge – Part 2

- Part 2: Blind spot detection
  - Blind spot is defined by coordinates (orange/brown rectangles)
  - Make an alert / warning (optic, acoustic or whatever) if an object is in the blind spot area
  - Extra fun ☺
    - Only show warning if the object classification is ‚car (value 2U)’
    - Object classification information (whether a detected object is a vehicle, pedestrian, etc...) is only available from the front camera
    - For this part, you’ll need to carry on the classification info from the front camera to the side of the vehicle



Code #LikeABosch
The Challenge – Part 3

- Part 3: Innovation and creativity
  - Invent / find new ideas, features, that could be realized based on these currently available sensors (4 corner radar, 1 video)
  - The new features shall currently not be on the market
  - We’re curious about your
    - Ideas
    - Technical details
    - Business case (why the market needs it)
    - If you have, prototype implementation



Code #LikeABosch
Evaluation guideline

- It’s not mandatory to complete or start all parts of the challenge
  - Deal with those parts which gives you the most joy and fun
  - In each part you have the chance to hand in a ‚basic’ and more advanced solutions as well, up to you what you choose
  - Having only highlevel solutions for all 3 parts can be as good as having more advanced and sophisticated solutions for less parts

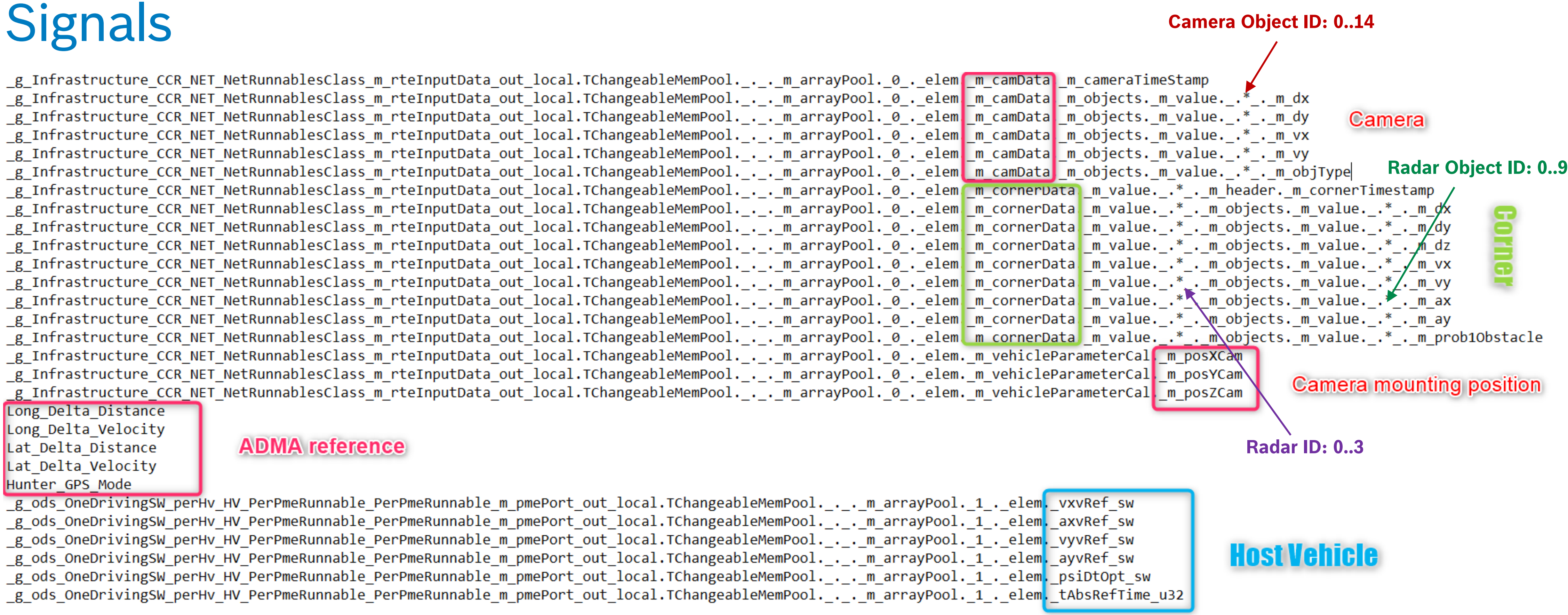
- Part 1:
  - Accuracy of the object tracking even in blindspots (using the validation dataset)
  - Technical realization of the solution
  - Presentation (method, visualization, presentation style)

- Part 2:
  - How precisely your solution gives the warnings (e.g. only when the objects are in the designated area, etc…)
  - If the warning comes only on pedestrians and cyclists or on all objects
  - Technical realization of the solution
  - Presentation (method, visualization, presentation style)

- Part 3:
  - Level of elaboration of the new idea(s)
  - Innovativeness
  - impact/value, business case
  - sustainability
  - feasibility
  - prototype implementation
  - presentation

Code #LikeABosch

Signals



```
_g_Infrastructure_CCR_NET_NetRunnablesClass_m_rteInputData_out_local.TChangeableMemPool._._._m_arrayPool._0_._elem._m_camData._m_cameraTimeStamp
_g_Infrastructure_CCR_NET_NetRunnablesClass_m_rteInputData_out_local.TChangeableMemPool._._._m_arrayPool._0_._elem._m_camData._m_objects._m_value._.*_._m_dx
_g_Infrastructure_CCR_NET_NetRunnablesClass_m_rteInputData_out_local.TChangeableMemPool._._._m_arrayPool._0_._elem._m_camData._m_objects._m_value._.*_._m_dy
_g_Infrastructure_CCR_NET_NetRunnablesClass_m_rteInputData_out_local.TChangeableMemPool._._._m_arrayPool._0_._elem._m_camData._m_objects._m_value._.*_._m_vx
_g_Infrastructure_CCR_NET_NetRunnablesClass_m_rteInputData_out_local.TChangeableMemPool._._._m_arrayPool._0_._elem._m_camData._m_objects._m_value._.*_._m_vy
_g_Infrastructure_CCR_NET_NetRunnablesClass_m_rteInputData_out_local.TChangeableMemPool._._._m_arrayPool._0_._elem._m_camData._m_objects._m_value._.*_._m_objType
_g_Infrastructure_CCR_NET_NetRunnablesClass_m_rteInputData_out_local.TChangeableMemPool._._._m_arrayPool._0_._elem._m_cornerData._m_value._.*_._m_header._m_cornerTimestamp
_g_Infrastructure_CCR_NET_NetRunnablesClass_m_rteInputData_out_local.TChangeableMemPool._._._m_arrayPool._0_._elem._m_cornerData._m_value._.*_._m_objects._m_value._.*_._m_dx
_g_Infrastructure_CCR_NET_NetRunnablesClass_m_rteInputData_out_local.TChangeableMemPool._._._m_arrayPool._0_._elem._m_cornerData._m_value._.*_._m_objects._m_value._.*_._m_dy
_g_Infrastructure_CCR_NET_NetRunnablesClass_m_rteInputData_out_local.TChangeableMemPool._._._m_arrayPool._0_._elem._m_cornerData._m_value._.*_._m_objects._m_value._.*_._m_dz
_g_Infrastructure_CCR_NET_NetRunnablesClass_m_rteInputData_out_local.TChangeableMemPool._._._m_arrayPool._0_._elem._m_cornerData._m_value._.*_._m_objects._m_value._.*_._m_vx
_g_Infrastructure_CCR_NET_NetRunnablesClass_m_rteInputData_out_local.TChangeableMemPool._._._m_arrayPool._0_._elem._m_cornerData._m_value._.*_._m_objects._m_value._.*_._m_vy
_g_Infrastructure_CCR_NET_NetRunnablesClass_m_rteInputData_out_local.TChangeableMemPool._._._m_arrayPool._0_._elem._m_cornerData._m_value._.*_._m_objects._m_value._.*_._m_ax
_g_Infrastructure_CCR_NET_NetRunnablesClass_m_rteInputData_out_local.TChangeableMemPool._._._m_arrayPool._0_._elem._m_cornerData._m_value._.*_._m_objects._m_value._.*_._m_ay
_g_Infrastructure_CCR_NET_NetRunnablesClass_m_rteInputData_out_local.TChangeableMemPool._._._m_arrayPool._0_._elem._m_cornerData._m_value._.*_._m_objects._m_value._.*_._m_prob1Obstacle
_g_Infrastructure_CCR_NET_NetRunnablesClass_m_rteInputData_out_local.TChangeableMemPool._._._m_arrayPool._0_._elem._m_vehicleParameterCal._m_posXCam
_g_Infrastructure_CCR_NET_NetRunnablesClass_m_rteInputData_out_local.TChangeableMemPool._._._m_arrayPool._0_._elem._m_vehicleParameterCal._m_posYCam
_g_Infrastructure_CCR_NET_NetRunnablesClass_m_rteInputData_out_local.TChangeableMemPool._._._m_arrayPool._0_._elem._m_vehicleParameterCal._m_posZCam
Long_Delta_Distance
Long_Delta_Velocity
Lat_Delta_Distance
Lat_Delta_Velocity
Hunter_GPS_Mode
_g_ods_OneDrivingSW_perHv_HV_PerPmeRunnable_PerPmeRunnable_m_pmePort_out_local.TChangeableMemPool._._._m_arrayPool._1_._elem._vxvRef_sw
_g_ods_OneDrivingSW_perHv_HV_PerPmeRunnable_PerPmeRunnable_m_pmePort_out_local.TChangeableMemPool._._._m_arrayPool._1_._elem._axvRef_sw
_g_ods_OneDrivingSW_perHv_HV_PerPmeRunnable_PerPmeRunnable_m_pmePort_out_local.TChangeableMemPool._._._m_arrayPool._1_._elem._vyvRef_sw
_g_ods_OneDrivingSW_perHv_HV_PerPmeRunnable_PerPmeRunnable_m_pmePort_out_local.TChangeableMemPool._._._m_arrayPool._1_._elem._ayvRef_sw
_g_ods_OneDrivingSW_perHv_HV_PerPmeRunnable_PerPmeRunnable_m_pmePort_out_local.TChangeableMemPool._._._m_arrayPool._1_._elem._psiDtOpt_sw
_g_ods_OneDrivingSW_perHv_HV_PerPmeRunnable_PerPmeRunnable_m_pmePort_out_local.TChangeableMemPool._._._m_arrayPool._1_._elem._tAbsRefTime_u32
```

Camera Object ID: 0..14

Camera

Radar Object ID: 0..9

Corner

Camera mounting position

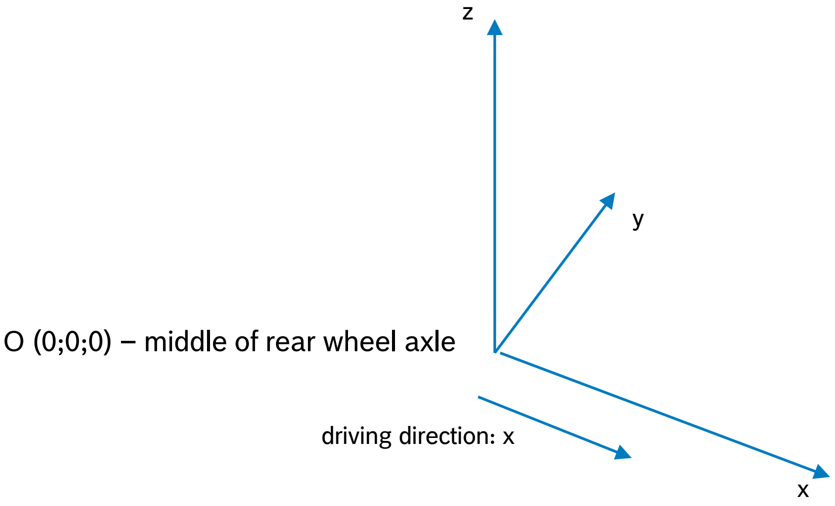
Radar ID: 0..3

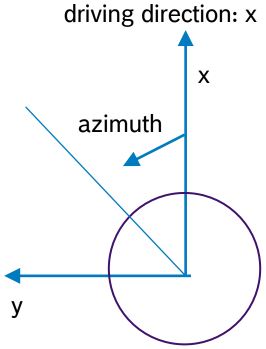
ADMA reference

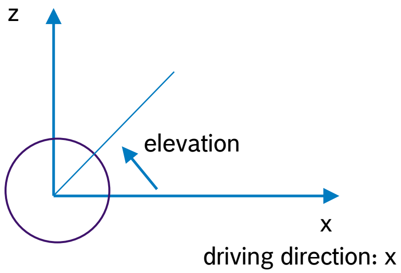
Host Vehicle

Code #LikeABosch
Sensor positions

	Value in degrees or mm
ANGLE_AZIMUTH_CORNER_RADAR_LEFT_FRONT	42
ANGLE_AZIMUTH_CORNER_RADAR_LEFT_REAR	135
ANGLE_AZIMUTH_CORNER_RADAR_RIGHT_FRONT	-42
ANGLE_AZIMUTH_CORNER_RADAR_RIGHT_REAR	-135
ANGLE_ELEVATION_CORNER_RADAR_LEFT_FRONT	0
ANGLE_ELEVATION_CORNER_RADAR_LEFT_REAR	0,48
ANGLE_ELEVATION_CORNER_RADAR_RIGHT_FRONT	0
ANGLE_ELEVATION_CORNER_RADAR_RIGHT_REAR	0,48
X_POSITION_CORNER_RADAR_LEFT_FRONT	3473,8
X_POSITION_CORNER_RADAR_LEFT_REAR	-766,4
X_POSITION_CORNER_RADAR_RIGHT_FRONT	3473,8
X_POSITION_CORNER_RADAR_RIGHT_REAR	-766,4
Y_POSITION_CORNER_RADAR_LEFT_FRONT	628,6
Y_POSITION_CORNER_RADAR_LEFT_REAR	738
Y_POSITION_CORNER_RADAR_RIGHT_FRONT	-628,6
Y_POSITION_CORNER_RADAR_RIGHT_REAR	-738
Z_POSITION_CORNER_RADAR_LEFT_FRONT	515,6
Z_POSITION_CORNER_RADAR_LEFT_REAR	735,9
Z_POSITION_CORNER_RADAR_RIGHT_FRONT	515,6
Z_POSITION_CORNER_RADAR_RIGHT_REAR	735,9


O (0;0;0) – middle of rear wheel axle





Code #LikeABosch
Details on signals

- Video object type enum key (m_ObjType):

```
{
  noDetection = 0U,
  truck       = 1U,
  car         = 2U,
  motorbike   = 3U,
  bicycle     = 4U,
  pedestrian  = 5U,
  carOrTruck  = 6U
};
```

- Host vehicle info:
  - vxvRef: longi velocity
  - axvRef: longi acceleration
  - vxvRef: lateral velocity
  - ayvRef: lateral acceleration
  - psiDtOpt: angular speed around axis z (yaw rate)
- Sensor input info:
  - dx: longi object distance
  - dy: lateral object distance
  - vx: longi object velocity
  - vy: lateral object velocity
  - ax: longi object acceleration
  - az: lateral object acceleration
- Units: meter, m/s and m/s^2

- Normalization:
- You need to denorm the sensor (both radar and video) inputs to get them in m, m/s, m/s^2 :
  - distances: devide the input by 128
  - velcoties: divide the input by 256
  - acceleration: divide the input by 2048
  - yaw arte: divide the input by 16384
  - probability: divide the input by 128

- ADMA signals
  - everything is in meter, m/s
  - no need for denorming
  - Hunger_GPS_Mode: gives the mode of the ADMA syste – it can be used when it’s 8 or 9