Continental Engineering Services GmbH

ARS 404-21

Entry

ARS 408-21

Premium

Standardized ARS Interface

Technical Documentation

ARS 404-21 (Entry)
ARS 408-21 (Premium)

Version 1.91

January 24, 2018

Table of Contents

1. Introduction 7
2. Operating Conditions 7
3. Safety Information 7
4. Interface 8
5. Description 10
6. Configuration Messages 12
6.1. Radar configuration (0x200) 12
6.2. Cluster and Object filter configuration (0x202) 15
6.3. Collision detection configuration (0x400) 19
6.4. Collision detection region configuration (0x401) 20
7. State Output 23
7.1. Radar state (0x201) 23
7.2. Cluster and Object filter configuration state header (0x203) 27
7.3. Cluster and Object filter configuration state (0x204) 27
7.4. Collision detection state (0x408) 31
7.5. Collision detection region state (0x402) 32
7.6. Control message for relay (0x8) 34
7.7. Software Version Identification (0x700) 35
7.8. Practical ways to use the module along with the sensor 33

8. Motion Input Signals 40
8.1. SpeedInformation (0x300) 40
8.2. Yaw rate Information (0x301) 41
9. Cluster List 42
9.1. Cluster list status (0x600) 43
9.2. Cluster general information (0x701) 44
9.3. Cluster quality information (0x702) 45
10. Object List 51
10.1. Object list status (0x60A) 52
10.2. Object general information (0x60B) 53
10.3. Object quality information (0x60C) 54
10.4. Object extended information (0x60D) 58
10.5. Object collision detection warning (0x60E) 60

History

Version	Date	SW	Change description
V 1.0	14.10.2016	04.30.0.16 and 04.31.0.16	First released version
V 1.1	18.01.2017	04.30.0.16 and 04.31.0.16	Second released version - Table 36 and Table 43 – corrections
V 1.2	25.02.2017	04.30.0.16 and 04.31.0.16	Third released version – Table 3 page 12
V 1.3	23.03.2017	04.30.0.16 and 04.31.0.16	Fourth released version - Table 43 page 47 – corrections – Last service of SW 04.30.0.16 (Standard) and of SW 04.31.0.16 (Reduced Power)
V 1.4	31.05.2017	1st Update 04.30.1.16 Fusion of: 04.30.0.16 and 04.31.0.16	Fifth released version – Adaption for SW version 04.30.1.16: Radar state (0x201), Software Version Identification (0x700)
V 1.5	21.07.2017	1st Update 04.30.1.16	Sixth released version – No. of clusters/objects specified *CollDetRegState_WarningLevel* regarding 0x2 changed in Table 20 Change to 196 m and 260 m regarding max. distance Change to 196 m and 1200 m regarding extended max. distance
V 1.6	01.08.2017	1st Update 04.30.1.16	Seventh released version – declaration of the ambiguity (darkness, fuzziness) of the clusters on page 40, table 34
V 1.7	16.10.2017	1st Update 04.30.1.16 and 04.10.0.16	Eighth released version – Adaptions regarding ARS 404-21 with SW 04.10.0.16 – integration of Offsets in different tables and minor clarifications
V 1.8	18.10.2017	1st Update 04.30.1.16 and 04.10.0.16	Ninth released version – Adaptions regarding layout and CAN-Bus Structures
V 1.90	17.11.2017 06.12.2017 11.01.2018	1st Update 04.30.1.16 and 04.10.0.16	Tenth released version – Figure 1 extended – object orientation angle Center of vehicle coordinate system, Chapters 7.6.1 and 7.6.2 added

V 1.91	22.01.2018	1st Update 04.30.1.16	Eleventh released version – comment added to table 14 “filter state header” Voltage Error scale added to table 12
V 1.92	XX.YY.2018	1st Update 04.30.1.16	Twelfth released version –

Table of Figures

Figure 1: CAN network terminated on CAN-Bus side 11
Figure 2: CAN network terminated on CAN-Bus side and on Radar side 9
Figure 3: ARS 40X coordinate system for clusters and objects. 11
Figure 4: RadarCfg– message layout (0x200). 12
Figure 5: FilterCfg - message layout for FilterCfg_Index=0x1 (0x202) with 12 bit length for minimum and maximum value. 16
Figure 6: FilterCfg - message layout for FilterCfg_Index=0xA (0x202) with 13 bit length for minimum and maximum value. 17
Figure 7: CollDetCfg - message layout (0x400). 19
Figure 8: Coordinates for defining a collision detection region.....................................20
Figure 9: CollDetRegionCfg - message layout (0x401)...21
Figure 10: Overview of state messages that are sent cyclically by the radar..................23
Figure 11: RadarState - message layout (0x201). 24
Figure 12: FilterState_Header - message layout (0x203). 27
Figure 13: FilterState_Cfg - message layout for FilterState_Index=0x1 (0x204) with 12 bit length for minimum and maximum value...29
Figure 14: FilterState_Cfg - message layout for FilterState_Index=0xA (0x204) with 13 bit length for minimum and maximum value...29
Figure 15: CollDetState - message layout (0x408). 31
Figure 16: CollDetRegionState - message layout (0x402). 32
Figure 17: CAN-Bus I/O Module: 4 digital outputs (Article Code DV-CANFRAS4-01) 34
Figure 18: CollDetRelayCtrl - message layout (0x8). 34
Figure 19: Real connection between the sensor, PLC and the relay……………………...35
Figure 20: Block diagram shows connection between the sensor, PLC and the relay…..36
Figure 21: VersionID - message layout (0x700)..37
Figure 22: SpeedInformation - message layout (0x300). 40
Figure 23: YawRateInformation - message layout (0x301). 41
Figure 24: Overview of cluster list messages that are sent cyclically by the radar...........42

Figure 25: Cluster_0_Status - message layout (0x600). 43
Figure 26: Cluster_1_General - message layout (0x701). 44
Figure 27: Cluster_2_Quality - message layout (0x702). 46
Figure 28: Overview of object list messages that are sent cyclically by the radar. 51
Figure 29: Object_0_Status - message layout (0x60A). 52
Figure 30: Object_1_General - message layout (0x60B). 53
Figure 31: Object_2_Quality - message layout (0x60C). 54
Figure 32 Object_3_Extended - message layout (0x60D). 58
Figure 33: Object_4_Warning - message layout (0x60E). 59

Table of Tables

Table 1: Sensor CAN messages (exemplary for sensor ID 0) 8
Table 2: RadarCfg – message content (0x200) 13
Table 3: RadarCfg – signal description (0x200) 14
Table 4: Possible filter criteria selectable with FilterCfg_Index. The columns C and O specify if this filter can be applied to clusters or objects, respectively. 15
Table 5: FilterCfg - message content (0x202). 17
Table 6: FilterCfg - signal description (0x202) 18
Table 8: CollDetCfg - signal description (0x400). 19
Table 9: CollDetRegionCfg - message content (0x401) 21
Table 10: CollDetRegionCfg - signal description (0x401) 21
Table 11: RadarState - message content (0x201) 24
Table 12: RadarState - signal description (0x201) 25
Table 13: FilterState_Header - message content (0x203) 27
Table 14: FilterState_Header - signal description (0x203) 27
Table 15: Possible filter criteria selectable with FilterState_Index. The columns C and O specify if this filter can be applied to clusters or objects, respectively. 28
Table 16: FilterState_Cfg - message content (0x204). 29
Table 17: FilterState_Cfg - signal description (0x204) 31
Table 18: CollDetState - message content (0x408) 31
Table 19: CollDetState - signal description (0x408) 32
Table 20: CollDetRegionState - message content (0x402) 33
Table 21: CollDetRegionState - signal description (0x402). 33
Table 22: CollDetRelayCtrl - message content (0x8) 35

Table 23: CollDetRelayCtrl - signal description (0x8) 35
Table 24: VersionID - message content (0x700)..37
Table 25: VersionID – signal description (0x700) 38
Table 26: SpeedInformation - message content (0x300) 40
Table 27: SpeedInformation – signal description (0x300)...40
Table 28: YawRateInformation - message content (0x301)..41
Table 29: YawRateInformation – signal description (0x301)..41
Table 30: Cluster_0_Status - message content (0x600) ARS 404-21 43
Table 31: Cluster_0_Status - message content (0x600) ARS 408-21 43
Table 32: Cluster_0_Status – signal description (0x600)..43
Table 33: Cluster_1_General - message content (0x701) 44
Table 34: Cluster_1_General - signal description (0x701)..45
Table 35: Cluster_2_Quality - message content (0x702)..46
Table 36: Cluster_2_Quality - signal description (0x702)..48
Table 37: Signal value table for Cluster_DistLong_rms, Cluster_DistLat_rms, Cluster_VrelLong_rms and Cluster_VrelLat_rms (0x702)..49
Table 38: Object_0_Status - message content (0x60A) 52
Table 39: Object_0_Status - signal description (0x60A) 52
Table 40: Object_1_General - message content (0x60B)..53
Table 41: Object_1_General - signal description (0x60B)..54
Table 42: Object_2_Quality - message content (0x60C) 55
Table 43: Object_2_Quality – signal description (0x60C) 56
Table 44: Signal value table for Obj_Orientation_rms, Obj_DistLong_rms, Obj_DistLat_rms, Obj_VrelLong_rms, Obj_VrelLat_rms, Obj_ArelLat_rms, Obj_ArelLong_rms (0x60C). 57
Table 45: Object_3_Extended - message content (0x60D) 59
Table 46: Object_3_Extended - signal description (0x60D)..60
Table 47: Object_4_Warning - message content (0x60E) 61
Table 48: Object_4_Warning - signal description (0x60E)..61

1. Introduction

The ARS 400 is a Radar Sensor System developed by Continental for the Automotive Industry to realize advanced driver assistance functions.

For the ARS 408 and ARS 404, the software of the sensor was adapted to use it for general purposes applications. The adaption includes simplifications for a non-automotive usage. With the easy to use software interface it is possible to connect the sensor to a CAN network and to provide radar based environmental information to one or several evaluation units. The sensor can also be configured via CAN.

In the following description for the radars, there will be MIN and MAX values for certain information (e.g. clusters, objects…). However, there will be also an offset and you shall find an offset when the MIN of that certain information is below ZERO. The offset value has been integrated into several tables already.

2. Operating Conditions

Please refer to *ARS 404/408-Technical Description.*

3. Safety Information

Please refer to *ARS 404/408-Technical Description.*

4. Interface

The ARS 404-21 and ARS 408-21 sensor has one CAN interface. The communication network is a CAN bus as specified in ISO 11898-2 with a transmission rate of 500 KBits/s. The sensor is not equipped with a termination resistor and care has to be taken to properly terminate the CAN bus.

The CAN connection of the ARS is used for configuration, sensor state output, data input and data output. To be able to add up to eight sensors to one CAN bus, the sensor ID can be configured, which will change the message IDs. The following table gives the message IDs for sensor ID 0. For the sensor IDs 0 to 7 the message ID can be calculated by

MsgId = MsgId_0 + SensorId * 0x10.

For example, the configuration message 0x200 for sensor ID 0, will be 0x210 for sensor ID 1. After setting the sensor ID, the sensor will only react to this new message ID for configuration. The only exception is the relay control message (0x8), which has the same ID for all sensors.

Table 1: Sensor CAN messages (exemplary for sensor ID 0)

In/Out	ID	Message Name	Content	Section
In	0x200	RadarCfg	Radar sensor configuration	6.1
Out	0x201	RadarState	Radar status	7.1
In	0x202	FilterCfg	Cluster and Object filter configuration	6.2
Out	0x203	FilterState_Header	Filter status header	7.2
Out	0x204	FilterState_Cfg	Filter configuration status	7.3
In	0x400	CollDetCfg	Collision detection configuration	6.3
In	0x401	CollDetRegionCfg	Collision detection region configuration	6.4
Out	0x408	CollDetState	Collision detection status	7.4
Out	0x402	CollDetRegionState	Collision detection region status	7.5
In	0x300	SpeedInformation	Vehicle (sensor platform) speed	8.1
In	0x301	YawRateInformation	Vehicle (sensor platform) yaw rate	8.2
Out	0x600	Cluster_0_Status	Cluster status (list header)	9.1
Out	0x701	Cluster_1_General	Cluster general information	9.2
Out	0x702	Cluster_2_Quality	Cluster quality information	9.3
Out	0x60A	Obj_0_Status	Object status (list header)	10.1
Out	0x60B	Obj_1_General	Object general information	10.2
Out	0x60C	Obj_2_Quality	Object quality information	10.3
Out	0x60D	Obj_3_Extended	Object extended information	10.4
Out	0x60E	Obj_4_Warning	Object collision detection warnings	10.5
Out	0x700	VersionID	Software Version Identification	7.6.1
Out	0x8	CollDetRelayCtrl	Control message for relay	7.6

As an example, a possible CAN bus network is shown in figure 1. Since no termination resistors are included in the radar sensor ARS 404-21 and ARS 408-21, two 120 Ohm terminal resistors have to be connected to the network (separately or integrated into the CAN interface of the corresponding unit). The CAN connection of the ARS 4 is used for configuration, sensor state output, data input and data output. In case of missing termination the CAN error BUSHEAVY can be the result.

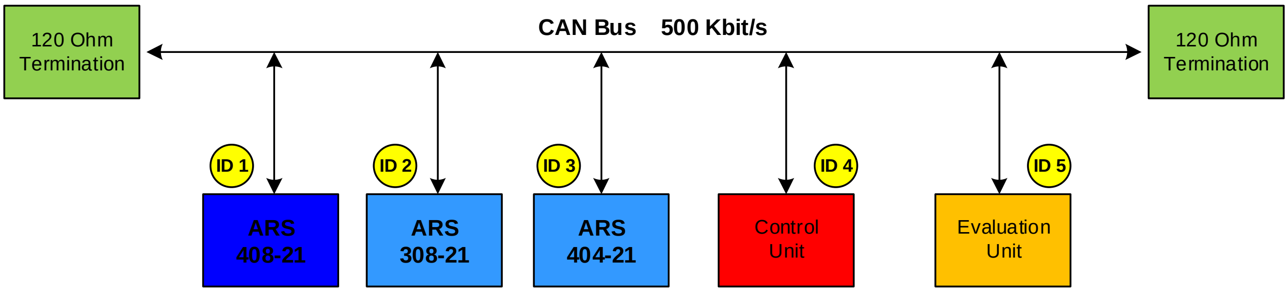


Figure 1: CAN network terminated on CAN-Bus side

As an example, a possible CAN bus network is shown in figure 2. One termination resistor is included in the radar sensor ARS 308-2T and one 120 Ohm terminal resistor has to be connected to the network (separately or integrated into the CAN interface of the corresponding unit). The CAN connection of the ARS is used for configuration, sensor state output, data input and data output.

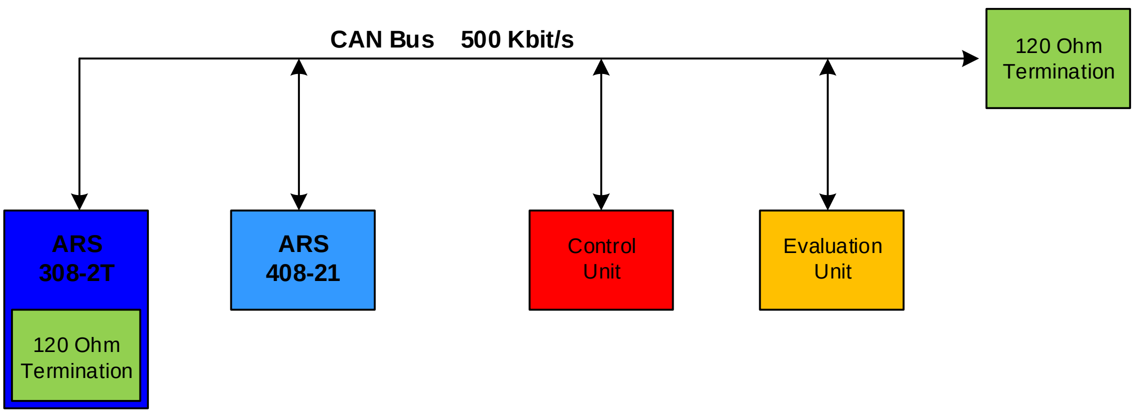


Figure 2: CAN network terminated on CAN-Bus side and on Radar side

5. Description

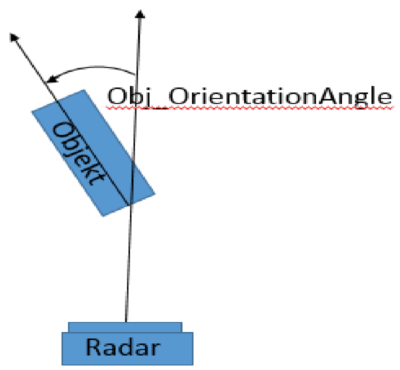
The ARS sensor uses radar radiation to analyze its surroundings. The reflected signals are processed and after multiple steps they are available in form of clusters and objects. Clusters are radar reflections with information like position, velocity and signal strength. They are newly evaluated every cycle. In contrast to this, objects have a history and dimension. They consist of tracked clusters.

The position is given in a Cartesian Coordinates System relative to the sensor as shown in figure 3.

The ARS 408-21 sensor shall be mounted with the plug pointing to the left, seen from behind (towards the positive y-axis) as shown in figure 3.
The velocity is calculated relative to an assumed vehicle course. The course is determined by using the speed and yaw rate information assuming that the radar sensor is mounted on the front and the movement is in longitudinal direction.
If speed and yaw rate information is missing, it will be set to default values:

Yaw Rate =	0 deg/s
Speed =	0 m/s
RadarDevice_SpeedDirection =	standstill

In the orientation of azimuth the Radar is made and prepared for the position of the center of a vehicle´s coordinate system.
In the orientation of longitudinal direction (+X-Axis) the center of rotation (center of gravity) is assumed to be 1,95 m behind the position of the center of a vehicle´s coordinate system which is the position of the radar.



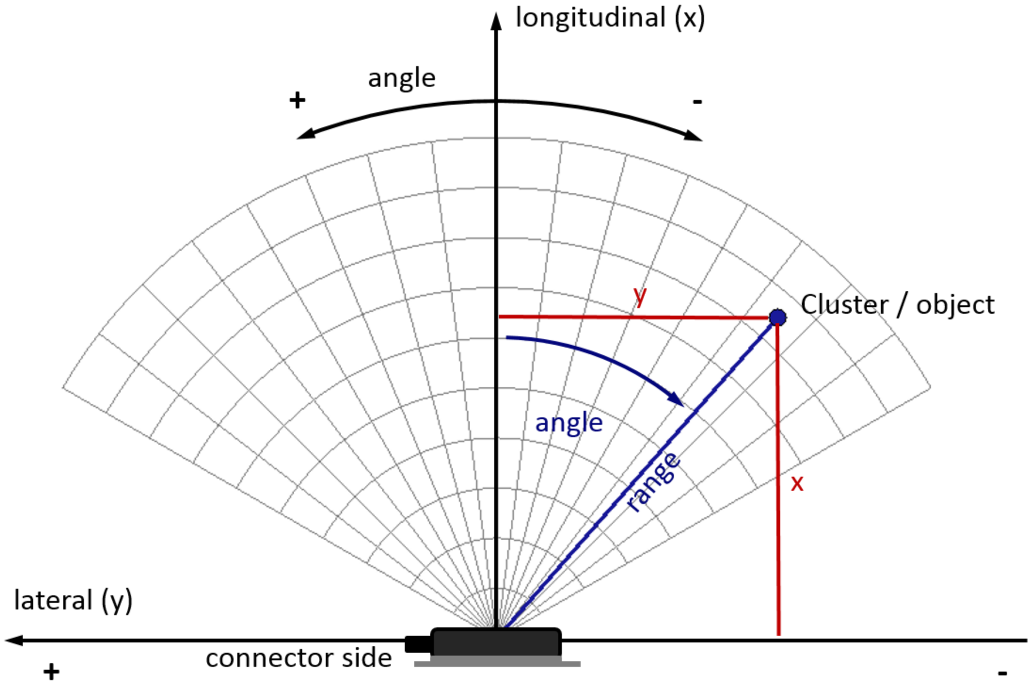


Figure 3: ARS 40X coordinate system for clusters and objects

Optionally, the output cluster and object lists can be filtered by setting filter criteria based on their attributes. Like this, the clusters or objects of interest that are sent on the CAN-bus can be selected. The filters for different attributes can be combined. More information on the filter configurations can be found in section 6.2.

Furthermore, a region-based collision detection is implemented on the sensor. If objects are detected within up to eight defined regions, the sensor will issue a warning message. More information on the collision detection configuration can be found in sections 6.3 and 6.4.

6. Configuration Messages

The basic configuration parameters of the radar sensor can be set with message ***RadarCfg*** (0x200). It is not necessary to cyclically repeat the configuration message. To store the configuration in the non-volatile memory (NVM) to be automatically set at startup on any subsequent power up, the signal ***RadarCfg_StoreInNVM*** has to be set to ***active*** (0x1). It is important to note that the number of transmissions to the NVM should be kept to a minimum as this could reduce the service life of the memory. Detailed descriptions of all configuration parameters can be found in section 6.1.

Additionally, filter criteria for the cluster and object list can bet set by message ***FilterCfg*** (0x202). The kind of filter to configure is set by multiplexor signals ***FilterCfg_Type*** (filter clusters or objects) and ***FilterCfg_Index*** (filter criterion). More details can be found in section 6.2.

The region-based collision detection can be configured with messages ***CollDetCfg*** (0x400) and ***CollDetRegCfg*** (0x401) for up to eight different regions. Details can be found in sections 6.3 and 6.4.

The filter and collision detection configurations are always stored in Nvm (when the configuration has been changed) and set automatically when the sensor restarts.

6.1. Radar configuration (0x200)

The ARS 404-21 and 408-21 can be configured via message ***RadarCfg*** (0x200). The parameters can be changed individually or in combinations. For each parameter, the message contains a validity bit (e.g. parameter ***RadarCfg_MaxDistance*** with ***RadarCfg_MaxDistance_valid***). If the validity bit is set to ***valid*** (0x1), the corresponding parameter will be updated in the ARS, otherwise it is ignored.

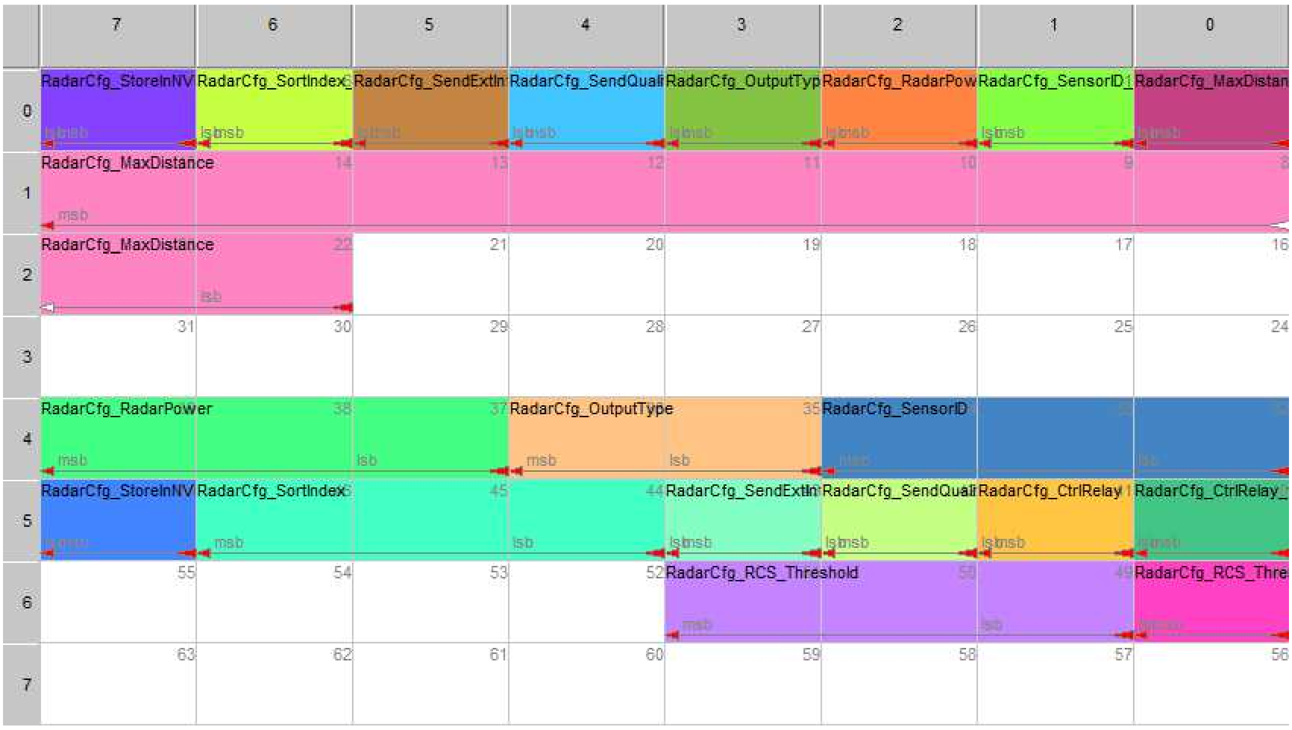


Figure 4: RadarCfg– message layout (0x200)

Table 2: RadarCfg – message content (0x200)

Signal	Start	Len	Min	Max	Res	Unit
RadarCfg_MaxDistance_valid	0	1	0	1	1	0x0: invalid 0x1: valid
RadarCfg_SensorID_valid	1	1	0	1	1	0x0: invalid 0x1: valid
RadarCfg_RadarPower_valid	2	1	0	1	1	0x0: invalid 0x1: valid
RadarCfg_OutputType_valid	3	1	0	1	1	0x0: invalid 0x1: valid
RadarCfg_SendQuality_valid	4	1	0	1	1	0x0: invalid 0x1: valid
RadarCfg_SendExtInfo_valid	5	1	0	1	1	0x0: invalid 0x1: valid
RadarCfg_SortIndex_valid	6	1	0	1	1	0x0: invalid 0x1: valid
RadarCfg_StoreInNVM_valid	7	1	0	1	1	0x0: invalid 0x1: valid
RadarCfg_MaxDistance	22	10	0	2046	2	m
RadarCfg_SensorID	32	3	0	7	1	Sensor ID 0x0 to 0x7
RadarCfg_OutputType	35	2	0	2	1	0x0: none 0x1: send objects 0x2: send clusters
RadarCfg_RadarPower	37	3	0	7	1	0x0: Standard 0x1: -3dB Tx gain 0x2: -6dB Tx gain 0x3: -9dB Tx gain
RadarCfg_CtrlRelay_valid	40	1	0	1	1	0x0: invalid 0x1: valid
RadarCfg_CtrlRelay	41	1	0	1	1	0x0: inactive 0x1: active
RadarCfg_SendQuality	42	1	0	1	1	0x0: inactive 0x1: active
RadarCfg_SendExtInfo	43	1	0	1	1	0x0: inactive 0x1: active
RadarCfg_SortIndex	44	3	0	7	1	0x0: no sorting 0x1: sorted by range 0x2: sorted by RCS
RadarCfg_StoreInNVM	47	1	0	1	1	0x0: inactive 0x1: active

RadarCfg_RCS_Threshold_valid	48	1	0	1	1	0x0: invalid 0x1: valid
RadarCfg_RCS_Threshold	49	3	0	7	1	0x0: Standard 0x1: High sensitivity

Table 3: RadarCfg – signal description (0x200)

Start	Signal	Description
32	RadarCfg_SensorID	Sensor ID 0 – 7 (see also section 4 for further explanation)
1	RadarCfg_SensorID_valid	Allow change of sensor ID if true
22	RadarCfg_MaxDistance	Maximum distance of far scan (near scan maximum distance is set proportionally to half of the far scan). The maximum distance will also change the range resolution proportionally. For example, 200 m maximum distance will lead to: - far scan 200 m with 1.79 m range resolution, - near scan 100 m with 0.42 m range resolution. **ARS408:** Standard Range: 196 m – 260 m Extended Range: 196 m – 1200 m **ARS404:** Standard Range: 150 m – 190 m Extended Range: 90 m – 1000 m
0	RadarCfg_MaxDistance_valid	Allow change of maximum distance if true
37	RadarCfg_RadarPower	Configures the transmitted radar power (Tx attenuation). The output RCS of cluster and objects will be compensated for this attenuation. Reducing the output power can improve detection in case of close range scenarios or inside rooms.
2	RadarCfg_RadarPower_valid	Allow change of radar output power if true
35	RadarCfg_OutputType	Configures the data output to clusters (0x2) or objects (0x1)
3	RadarCfg_OutputType_valid	Allow change of output type if true
42	RadarCfg_SendQuality	Cluster or object quality information (message 0x60C or 0x702) is sent if true
4	RadarCfg_SendQuality_valid	Allow change of quality option if true
43	RadarCfg_SendExtInfo	Extended information (message 0x60D) is sent for objects if true (if clusters are selected as output type this value is ignored)
5	RadarCfg_SendExtInfo_valid	Allow change of extended information option if true
44	RadarCfg_SortIndex	Selects the sorting index for the object list (ignored for clusters as they are always sorted by range)

6	RadarCfg_SortIndex_valid	Allow change of sorting index if true
41	RadarCfg_CtrlRelay	Relay control message (0x8) is sent if true and the collision detection is activated
40	RadarCfg_CtrlRelay_valid	Allow change of relay control output if true
49	RadarCfg_RCS_Threshold	Sets the sensitivity of the cluster detection to standard (0x0) or high sensitivity (0x1)
48	RadarCfg_RCS_Threshold_valid	Allow change of RCS threshold option if true
47	RadarCfg_StoreInNVM	Stores the current configuration to non-volatile memory to be read and set at sensor startup.
7	RadarCfg_StoreInNVM_valid	Allow storing to non-volatile memory if true

6.2. Cluster and Object filter configuration (0x202)

With multiplex message ***FilterCfg*** (0x202), output filters for the cluster or object list can be configured. The kind of filter to configure is set by multiplexor signals ***FilterCfg_Type*** (filter clusters or objects) and ***FilterCfg_Index*** (filter criteria). In Table 4, the possible filter indices are described.

The filters are designed as pass though filters, i.e. for each filter criterion, a valid range with minimum (***FilterCfg_Min_xxx***) and maximum (***FilterCfg_Max_xxx***) value can be specified. Exception is the filter index 0x0 (NofObj) for which only a maximum value can be set.

When a filter configuration message is sent, the sensor replies with the messages ***FilterState_Header*** (0x203) with the number of configured filters and one message ***FilterState_Cfg*** (0x204) for the filter that has been changed.

Table 4: Possible filter criteria selectable with FilterCfg_Index. The columns C and O specify if this filter can be applied to clusters or objects, respectively.

Index	Filter criterium	C	O	Description
0x0	NofObj	x	x	Limits the maximum number of cluster or objects to send out (minimum value is ignored)
0x1	Distance	x	x	Radial distance in m [r = sqrt(x^2 + y^2)]
0x2	Azimuth	x	x	Azimuth angle in degree [a = arc tan(y/x)]
0x3	VrelOncome	x	x	Radial velocity in sensor line-of-sight in m/sec of oncoming clusters or objects (all departing clusters and objects are ok)
0x4	VrelDepart	x	x	Radial velocity in sensor line-of-sight in m/sec of departing clusters or objects (all oncoming clusters and objects are ok)
0x5	RCS	x	x	RCS value (Radar cross section) in dBm^2
0x6	Lifetime		x	Life time (since first detection) in seconds
0x7	Size		x	Object size as area in m^2 (length x width)

0x8	ProbExists		x	Probability of existence, i.e. probability for being a real target and not a sensor artifact caused by multipath etc. Possible values are - 0x0: 0% - 0x1: 25% - 0x2: 50% - 0x3: 75% - 0x4: 90% - 0x5: 99% - 0x6: 99.9% - 0x7: 100%
0x9	Y		x	Y-position in m (lateral distance)
0xA	X		x	X-position in m (longitudinal distance)
0xB	VYRightLeft		x	Lateral velocity component in m/sec for right-left moving objects (all left-right moving objects are ok)
0xC	VXOncome		x	Longitudinal velocity component in m/sec for oncoming objects (all departing objects are ok)
0xD	VYLeftRight		x	Lateral velocity component in m/sec for left-right moving objects (all right-left moving objects are ok)
0xE	VXDepart		x	Longitudinal velocity component in m/sec for departing objects (all oncoming objects are ok)

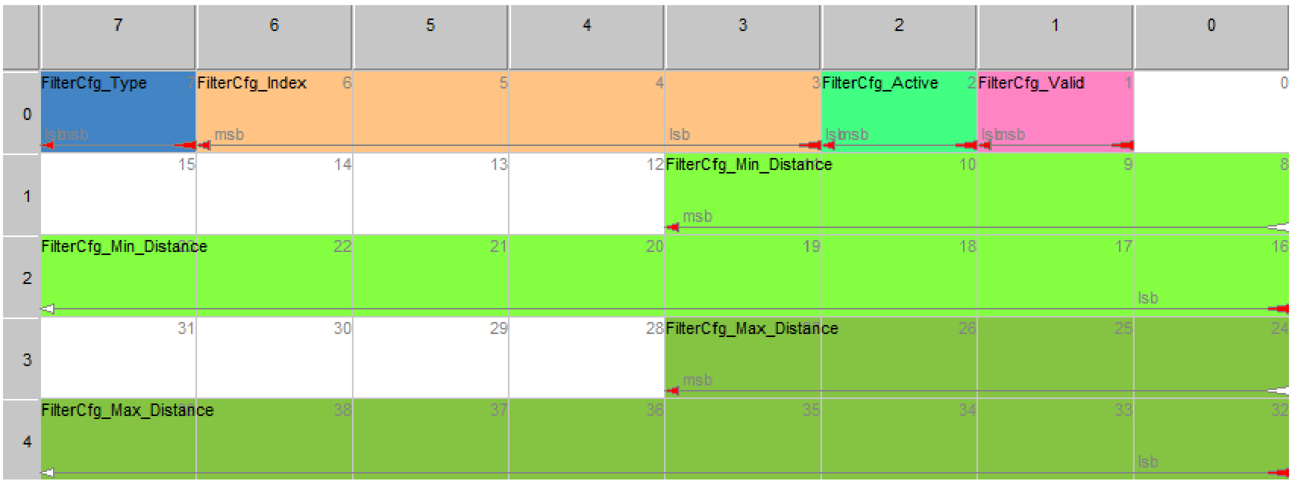


Figure 5: FilterCfg - message layout for FilterCfg_Index=0x1 (0x202) with 12 bit length for minimum and maximum value

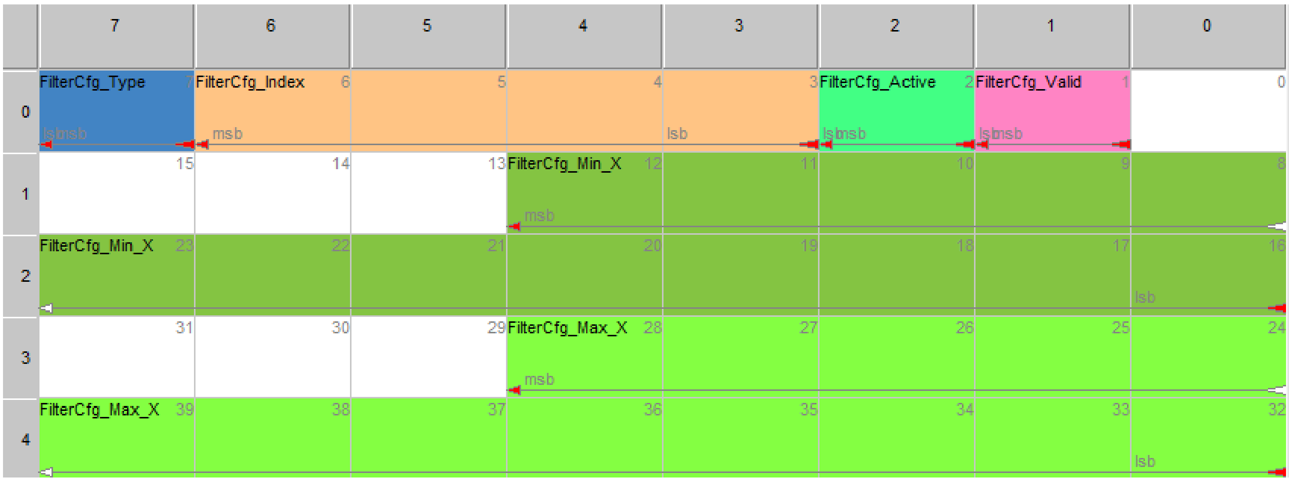


Figure 6: FilterCfg - message layout for FilterCfg_Index=0xA (0x202) with 13 bit length for minimum and maximum value

Table 5: FilterCfg - message content (0x202).

Signal	Start	Len	Offset	Min	Max	Res	Unit
FilterCfg_Valid	1	1		0	1	1	0x0: invalid 0x1: valid
FilterCfg_Active	2	1		0	1	1	0x0: inactive 0x1: active
FilterCfg_Index	3	4		0	15	1	Table 4 about the 15 different filter criteria: 0x0 –> 0xE
FilterCfg_Type	7	1		0	1	1	0x0: Cluster filter 0x1: Object filter
FilterCfg_Min_NofObj	16	12		0	4095	1	ignored
FilterCfg_Min_Distance	16	12		0	409.5	0.1	m
FilterCfg_Min_Azimuth	16	12	-50	-50	52.375	0.025	deg
FilterCfg_Min_VrelOncome	16	12		0	128.993	0.0315	m/s
FilterCfg_Min_VrelDepart	16	12		0	128.993	0.0315	m/s
FilterCfg_Min_RCS	16	12	-50	-50	52.375	0.025	dBm²
FilterCfg_Min_Lifetime	16	12		0	409.5	0.1	s
FilterCfg_Min_Size	16	12		0	102.375	0.025	m²
FilterCfg_Min_ProbExists	16	12		0	7	1	0x0: 0% 0x1: 25% 0x2: 50% 0x3: 75% 0x4: 90% 0x5: 99%

							0x6: 99.9% 0x7: 100%
FilterCfg_Min_Y	16	12	-409.5	-409.5	409.5	0.2	m
FilterCfg_Min_X	16	13	-500	-500	1138.2	0.2	m
FilterCfg_Min_VYRightLeft	16	12		0	128.993	0.0315	m/s
FilterCfg_Min_VXOncome	16	12		0	128.993	0.0315	m/s
FilterCfg_Min_VYLeftRight	16	12		0	128.993	0.0315	m/s
FilterCfg_Min_VXDepart	16	12		0	128.993	0.0315	m/s
FilterCfg_Max_NofObj	32	12		0	4095	1	-
FilterCfg_Max_Distance	32	12		0	409.5	0.1	m
FilterCfg_Max_Azimuth	32	12	-50	-50	52.375	0.025	deg
FilterCfg_Max_VrelOncome	32	12		0	128.993	0.0315	m/s
FilterCfg_Max_VrelDepart	32	12		0	128.993	0.0315	m/s
FilterCfg_Max_RCS	32	12	-50	-50	52.375	0.025	dBm²
FilterCfg_Max_Lifetime	32	12		0	409.5	0.1	s
FilterCfg_Max_Size	32	12		0	102.375	0.025	m²
FilterCfg_Max_ProbExists	32	12		0	7	1	0x0: 0% 0x1: 25% 0x2: 50% 0x3: 75% 0x4: 90% 0x5: 99% 0x6: 99.9% 0x7: 100%
FilterCfg_Max_Y	32	12	-409.5	-409.5	409.5	0.2	m
FilterCfg_Max_X	32	13	-500	-500	1138.2	0.2	m
FilterCfg_Max_VYRightLeft	32	12		0	128.993	0.0315	m/s
FilterCfg_Max_VXOncome	32	12		0	128.993	0.0315	m/s
FilterCfg_Max_VYLeftRight	32	12		0	128.993	0.0315	m/s
FilterCfg_Max_VXDepart	32	12		0	128.993	0.0315	m/s

Table 6: FilterCfg - signal description (0x202)

Start	Signal	Description
1	FilterCfg_Valid	Allow change of filter configuration if true
2	FilterCfg_Active	De-/activate filter configuration for specified filter criterion (FilterCfg_Index) and type (FilterCfg_Type)
3	FilterCfg_Index	Multiplexor to specify which of the 15 filter criteria to configure (0x0 → 0xE)

7	FilterCfg_Type	Multiplexor to choose between object or cluster filter configuration
16	FilterCfg_Min_xxx	Minimum value for filter criterion. The content depends on FilterCfg_Index (see Table 4 for more details)
32	FilterCfg_Max_xxx	Maximum value for filter criterion. The content depends on FilterCfg_Index (see Table 4 for more details)

6.3. Collision detection configuration (0x400)

Region-based collision detection can be activated with message ***CollDetCfg*** (0x400). With activated collision detection, the sensor sends cyclically (once per second) the message ***CollDetState*** (0x408) and messages ***CollDetRegionState*** (0x402) for all configured regions.

Collision detection is only possible with objects, if ***RadarCfg_OutputType*** is set to ***send Objects*** (0x2). Then, the message ***Obj_4_Warning*** (0x60E) is send for each object, indicating if the object is violating any regions.

Furthermore, the relay control message ***CollDetRelayCtrl*** (0x8) is sent, if it is activated with ***RadarCfg_CtrlRelay***. It contains a bit field with the current warning state of the regions, to control a relay.

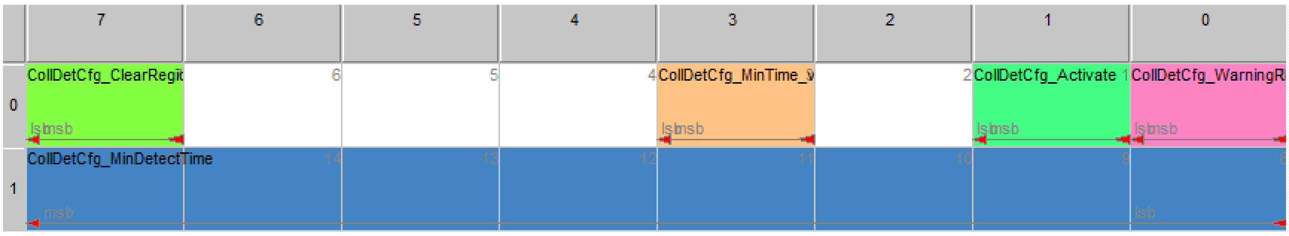


Figure 7: CollDetCfg - message layout (0x400)

Table 8: CollDetCfg - message content (0x400)

Signal	Start	Len	Min	Max	Res	Unit
CollDetCfg_WarningReset	0	1	0	1	1	0x0: idle 0x1: reset warnings
CollDetCfg_Activation	1	1	0	1	1	0x0: inactive 0x1: active
CollDetCfg_MinTime_valid	3	1	0	1	1	0x0: invalid 0x1: valid
CollDetCfg_ClearRegions	7	1	0	1	1	0x0: idle 0x1: clear regions
CollDetCfg_MinTime	8	8	0.0	25.5	0.1	sec

Table 7: CollDetCfg - signal description (0x400)

Start	Signal	Description

0	CollDetCfg_WarningReset	Reset currently active warnings of all regions (CollDetRegState_WarningLevel) if true
1	CollDetCfg_Activation	De-/activate collision detection function
3	CollDetCfg_MinTime_valid	Allow change of time parameter if true
7	CollDetCfg_ClearRegions	Clear all region configurations (set all to inactive) if true
8	CollDetCfg_MinTime	Minimum time an object needs to be detected inside the region before a warning is triggered

6.4. Collision detection region configuration (0x401)

Eight different rectangular regions can be configured with message ***CollDetRegCfg*** (0x401). Each region is defined by two points with x and y coordinates spanning a rectangle as shown in Figure . Collision regions can also overlap.

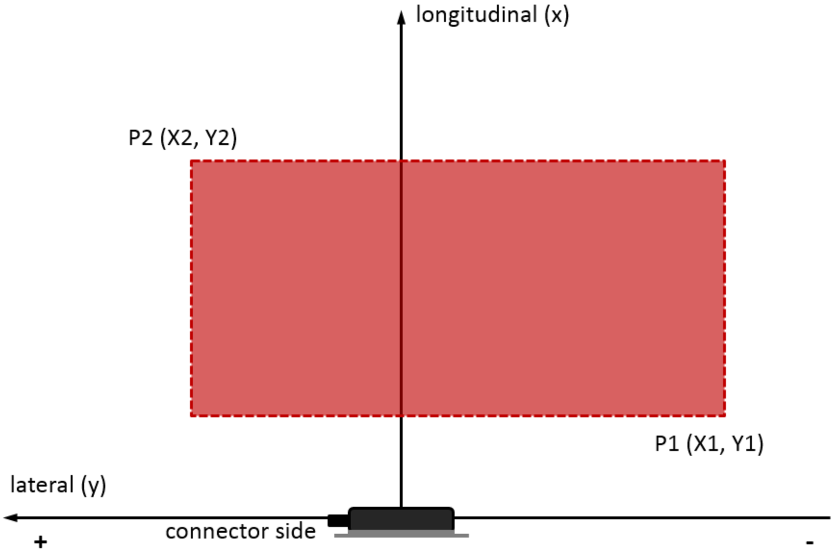


Figure 8: Coordinates for defining a collision detection region

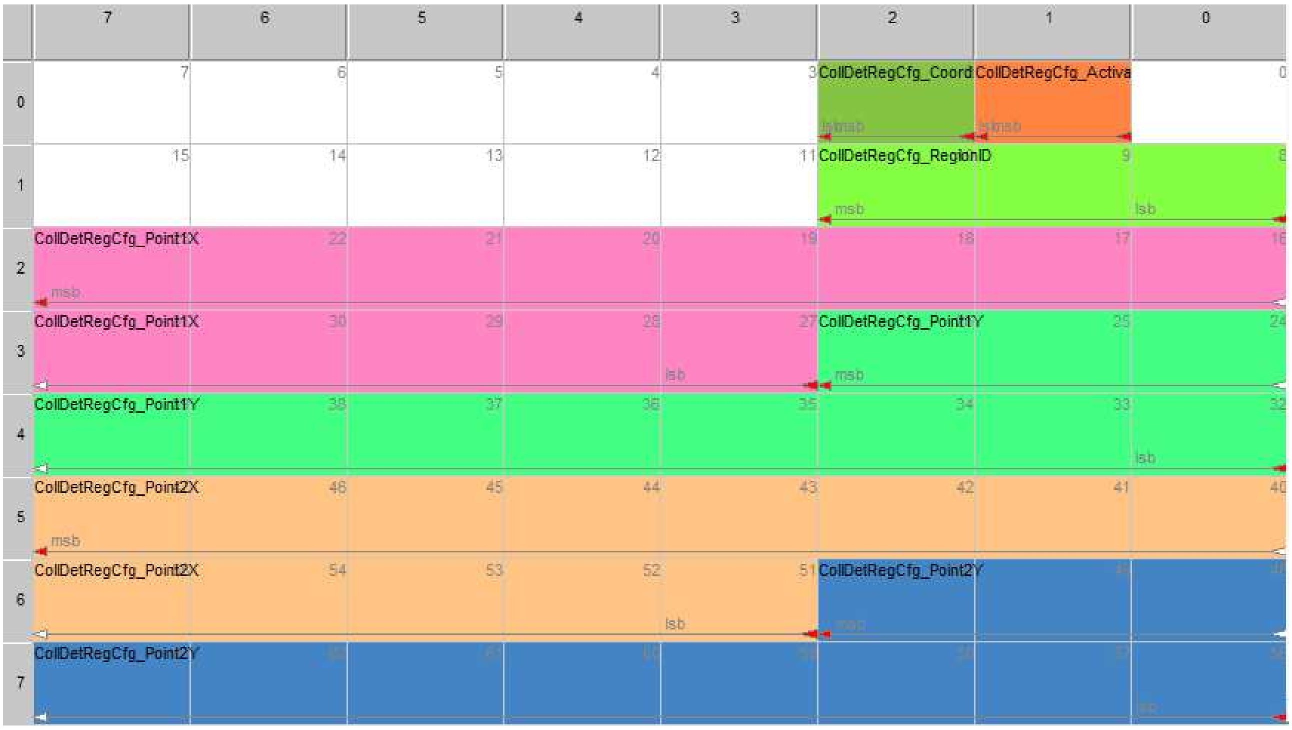


Figure 9: CollDetRegionCfg - message layout (0x401)

Table 8: CollDetRegionCfg - message content (0x401)

Signal	Start	Len	Offset	Min	Max	Res	Unit
CollDetRegCfg_Activation	1	1		0	1	1	0x0: inactive 0x1: active
CollDetRegCfg_Coordinates_valid	2	1		0	0	1	0x0: invalid 0x1: valid
CollDetRegCfg_RegionID	8	3		0	7	1	-
CollDetRegCfg_Point1X	27	13	-500	-500	1138.2	0.2	m
CollDetRegCfg_Point1Y	32	11	-204.6	-204.6	204.8	0.2	m
CollDetRegCfg_Point2X	51	13	-500	-500	1138.2	0.2	m
CollDetRegCfg_Point2Y	56	11	-204.6	-204.6	204.8	0.2	m

Table 9: CollDetRegionCfg - signal description (0x401)

Start	Signal	Description
1	CollDetRegCfg_Activation	De-/activate current region
2	CollDetRegCfg_Coordinates_valid	Allow change of current region's coordinates if true
5	CollDetRegCfg_RegionID	ID of current region to configure (0-7)
27	CollDetRegCfg_Point1X	1st X (longitudinal) coordinate of rectangular region
32	CollDetRegCfg_Point1Y	1st Y (lateral) coordinate of rectangular region

51	CollDetRegCfg_Point2X	2nd X (longitudinal) coordinate of rectangular region
56	CollDetRegCfg_Point2Y	2nd Y (lateral) coordinate of rectangular region

7. State Output

The sensor is always sending cyclically (once per second) the current configuration and sensor state in message ***RadarState*** (0x201) and current firmware version in message ***VersionID*** (0x700).

For the filter configuration, the sensor only sends the current configuration as a reply to the filter configuration message ***FilterCfg*** (0x202). The sensor replies with the message ***FilterState_Header*** (0x203) with the number of configured filters and one message ***FilterState_Cfg*** (0x204) for the filter that has been configured.

When collision detection is active, additionally, the sensor sends the current collision detection configuration and warning state in message ***CollDetState*** (0x408) and for the individual regions in messages ***CollDetRegionState*** (0x402). Furthermore, the relay control message ***CollDetRelayCtrl*** (0x8) is sent, if it is activated with ***RadarCfg_CtrlRelay***. They are also sent cyclically (once per second).

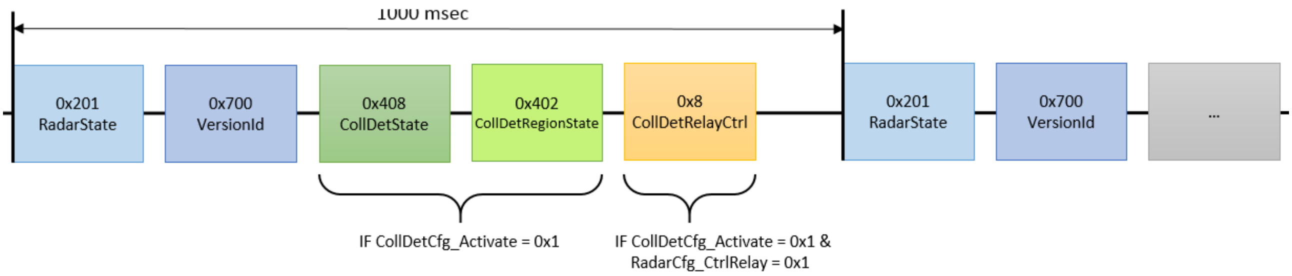


Figure 10: Overview of state messages that are sent cyclically by the radar

7.1. Radar state (0x201)

The message ***RadarState*** (0x201) is sent by the sensor in a regular interval (once per second). After configuring a radar configuration parameter, the corresponding signal in message 0x201 can be checked in order to verify that the configuration change has been accepted.

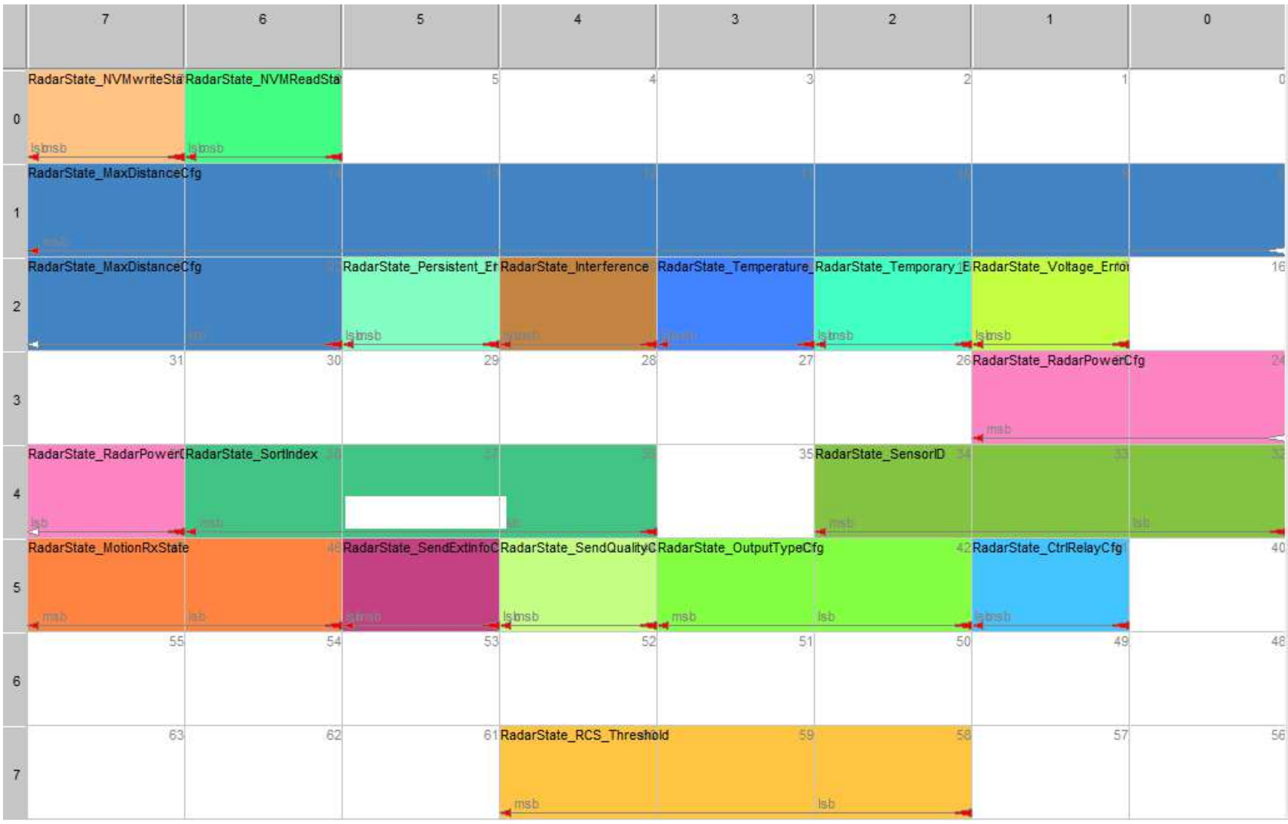


Figure 11: RadarState - message layout (0x201)

Table 10: RadarState - message content (0x201)

Signal	Start	Len	Min	Max	Res	Unit
RadarState_NVMReadStatus	6	1	0	1	1	0x0: failed 0x1: successful
RadarState_NVMwriteStatus	7	1	0	1	1	0x0: failed 0x1: successful
RadarState_MaxDistanceCfg	22	10	0	2046	2 2 2 2	**ARS408:** **-** Standard Range: 196 m – 260 m **-** Extended Range: 196 m – 1200 m **ARS404:** **-** Standard Range: 150 m – 190 m **-** Extended Range: 90 m – 1000 m
RadarState_Persistent_Error	21	1	0	1	1	0x0: No error 0x1: Persistent error active
RadarState_Interference	20	1	0	1	1	0x0: No interference 0x1: Interference detected

RadarState_Temperature_Error	19	1	0	1	1	0x0: No error 0x1: Temperature error active
RadarState_Temporary_Error	18	1	0	1	1	0x0: No error 0x1: Temporary error active
RadarState_Voltage_Error	17	1	0	1	1	0x0: No error 0x1: Voltage error active
RadarState_SensorID	32	3	0	7	1	Current sensor ID (0-7)
RadarState_SortIndex	36	3	0	7	1	0x0: no sorting 0x1: sorted by range 0x2: sorted by RCS
RadarState_RadarPowerCfg	39	3	0	7	1	0x0: Standard 0x1: -3dB Tx gain 0x2: -6dB Tx gain 0x3: -9dB Tx gain
RadarState_CtrlRelayCfg	41	1	0	1	1	0x0: inactive 0x1: active
RadarState_OutputTypeCfg	42	2	0	3	1	0x0: none 0x1: send objects 0x2: send clusters
RadarState_SendQualityCfg	44	1	0	1	1	0x0: inactive 0x1: active
RadarState_SendExtInfoCfg	45	1	0	1	1	0x0: inactive 0x1: active
RadarState_MotionRxState	46	2	0	3	1	0x0: input ok 0x1: speed missing 0x2: yaw rate missing 0x3: speed and yaw rate missing
RadarState_RCS_Threshold	58	3	0	7	1	0x0 Standard 0x1 High sensitivity

Table 11: RadarState - signal description (0x201)

Start	Signal	Description
6	RadarState_NVMReadStatus	State of reading the configuration parameters from non-volatile memory at startup
7	RadarState_NVMwriteStatus	State of storing a configuration parameter to non-volatile memory (initially this value is set to 0x0 and set to 0x1 after a configuration has been sent and successfully stored)
17	RadarState_Voltage_Error	Error will be active if the operating voltage is below or above the defined range for more than **5** seconds. U1 = < 7.1 VDC ---> Radar switches off U2 = 7.3-7.4 VDC ---> voltage error = 1

		U3 = 7.5-7.6 VDC ---> voltage error = 0 Radar works U4 = 7.6-32.8 VDC---> Radars sends telegram U5 = 32.8 VDC ---> voltage error = 0 U6 = 32.9 VDC ---> voltage error = 1 U7 = > 32,9 VDC ---> Radar switches off It does always automatically quit after it has gone.
18	RadarState_Temporary_Error	A temporary error which will most probably disappear after a sensor reset has been detected.
19	RadarState_Temperature_Error	Error will be active if the temperature is below or above the defined range. It does automatically quit after it has gone.
20	RadarState_Interference	Interference with another radar sensor has been detected. In case of Interference there is no change in any interface function of the radar. Internally the background noise increases which decreases the performance slowly depending on the strength of interference, but the radar will go on in sending Clusters or Objects to the interface as long as any obstacle is detected. It does always automatically quit after it has gone. It should be used as an indication that the measurement of the radar is influenced by another radar and the performance will probably decrease a little bit. Every radar is protected against the effect of interference by a randomly unbalanced length of interrupts between the measurements. This avoids a permanent interference with another radar and it only creates partly interference with the assurance to leave immediately after appearing. To create full radar performance any interference should be avoided / kept as less as possible !!
21	RadarState_Persistent_Error	An internal error which might not disappear after a reset has been detected.
22	RadarState_MaxDistanceCfg	Current configuration of maximum distance of far scan – different of ARS 404 and ARS 408
32	RadarState_SensorID	Sensor ID 0 - 7
36	RadarState_SortIndex	Current configuration of sorting index for object list
39	RadarState_RadarPowerCfg	Current configuration of transmitted radar power parameter
41	RadarState_CtrlRelayCfg	True if relay control message is sent
42	RadarState_OutputTypeCfg	Currently selected output type as either clusters or objects
44	RadarState_SendQualityCfg	True if quality information is sent for clusters or objects
45	RadarState_SendExtInfoCfg	True if extended information is sent for objects
46	RadarState_MotionRxState	Shows the state of the speed and yaw rate input signals
58	RadarState_RCS_Threshold	If true the sensors high sensitivity mode is active

7.2. Cluster and Object filter configuration state header (0x203)

After configuring a filter parameter, the sensor replies with messages ***FilterState_Header*** (0x203) and filter status message ***FilterState_Cfg*** (0x204) of the filter that was modified.

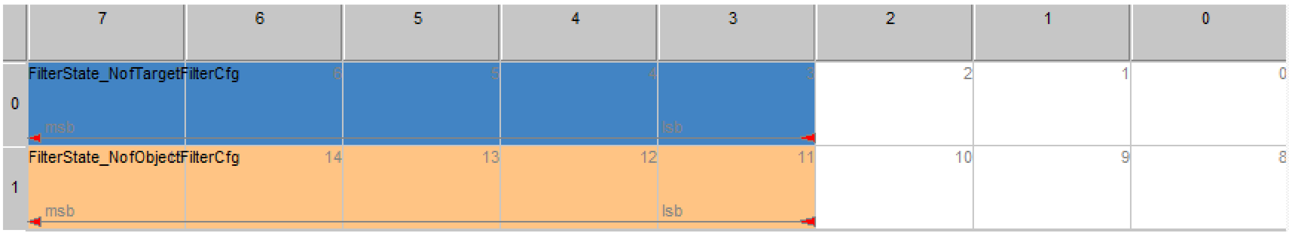


Figure 12: FilterState_Header - message layout (0x203)

Table 12: FilterState_Header - message content (0x203)

Signal	Start	Len	Min	Max	Res	Unit
FilterState_NofClusterFilterCfg	3	5	0	31	1	-
FilterState_NofObjectFilterCfg	11	5	0	31	1	-

Table 13: FilterState_Header - signal description (0x203)

Start	Signal	Description
3	FilterState_NofClusterFilterCfg	Number of filter status messages for configured Cluster filter Depending on the amount of massages it could happen that there are no clusters displayed although the radar is working with full performance.
11	FilterState_NofObjectFilterCfg	Number of filter status messages for configured Object filter Depending on the amount of massages it could happen that there are no objects displayed although the radar is working with full performance. Clusters are not influenced by this event.

7.3. Cluster and Object filter configuration state (0x204)

After configuring a filter parameter, the sensor replies with messages ***FilterState_Header*** (0x203) and filter status message ***FilterState_Cfg*** (0x204) of the filter that was modified.

The kind of filter is defined by multiplexor signals ***FilterState_Type*** and ***FilterState_Index***. In Table 14, the possible filter indices are described. The table is a copy of Table 4 and just repeated here for convenience.

Table 14: Possible filter criteria selectable with FilterState_Index. The columns C and O specify if this filter can be applied to clusters or objects, respectively.

Index	Filter criterion	C	O	Description
0x0	NofObj	x	x	Limits the maximum number of cluster or objects to send out (minimum value is ignored)
0x1	Distance	x	x	Radial distance in m [r = sqrt(x^2 + y^2)]
0x2	Azimuth	x	x	Azimuth angle in degree [a = arctan(y/x)]
0x3	VrelOncome	x	x	Radial velocity in sensor line-of-sight in m/sec of oncoming clusters or objects (all departing clusters and objects are ok)
0x4	VrelDepart	x	x	Radial velocity in sensor line-of-sight in m/sec of departing clusters or objects (all oncoming clusters and objects are ok)
0x5	RCS	x	x	RCS value (Radar cross section) in dBm^2
0x6	Lifetime		x	Life time (since first detection) in seconds
0x7	Size		x	Object size as area in m^2 (length x width)
0x8	ProbExists		x	Probability of existence, i.e. probability for being a real target and not a sensor artefact caused by multipath etc. Possible values can be - 0x0: 0% - 0x1: 25% - 0x2: 50% - 0x3: 75% - 0x4: 90% - 0x5: 99% - 0x6: 99.9% - 0x7: 100%
0x9	Y		x	Y-position in m (lateral distance)
0xA	X		x	X-position in m (longitudinal distance)
0xB	VYRightLeft		x	Lateral velocity component in m/sec for right-left moving objects (all left-right moving objects are ok)
0xC	VXOncome		x	Longitudinal velocity component in m/sec for oncoming objects (all departing objects are ok)
0xD	VYLeftRight		x	Lateral velocity component in m/sec for left-right moving objects (all right-left moving objects are ok)
0xE	VXDepart		x	Longitudinal velocity component in m/sec for departing objects (all oncoming objects are ok)

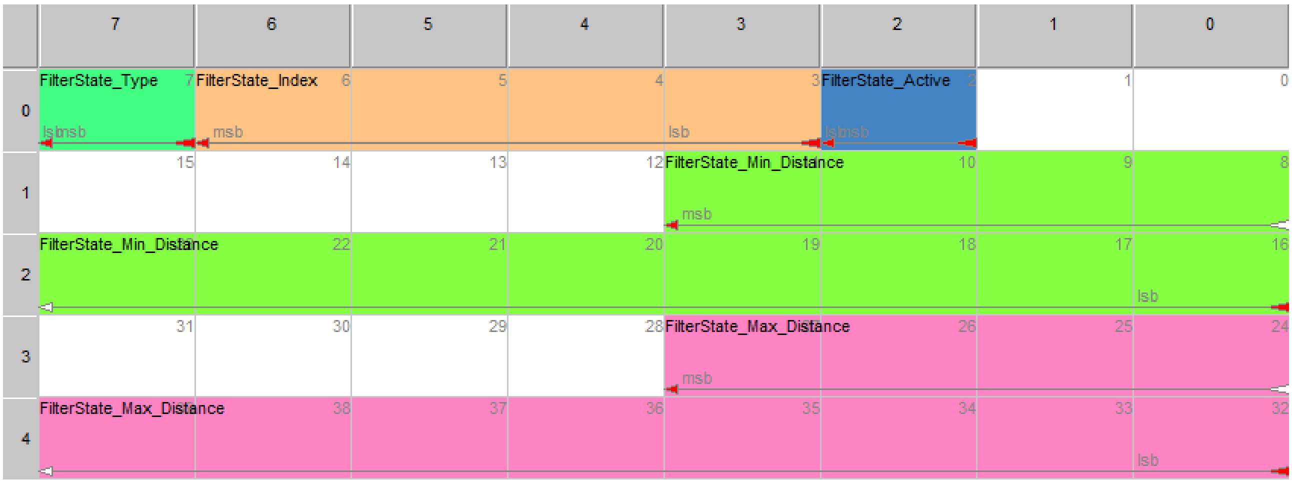


Figure 13: FilterState_Cfg - message layout for FilterState_Index=0x1 (0x204) with 12 bit length for minimum and maximum value.

Figure14: FilterState_Cfg - message layout for FilterState_Index=0xA (0x204) with 13 bit length for minimum and maximum value.

Table 15: FilterState_Cfg - message content (0x204).

Signal	Start	Len	Offset	Min	Max	Res	Unit
FilterState_Active	2	1		0	1	1	0x0: invalid 0x1: valid
FilterState_Index	3	4		0	15	1	see Table 4 about the 15 different filter criteria: 0x0 –> 0xE
FilterState_Type	7	1		0	1	1	0x0: Cluster filter 0x1: Object filter
FilterState_Min_NofObj	16	12		0	4095	1	not used

FilterState_Min_Distance	16	12		0	409.5	0.1	m
FilterState_Min_Azimuth	16	12	-50	-50	52.375	0.025	deg
FilterState_Min_VrelOncome	16	12		0	128.993	0.0315	m/s
FilterState_Min_VrelDepart	16	12		0	128.993	0.0315	m/s
FilterState_Min_RCS	16	12	-50	-50	52.375	0.025	dBm²
FilterState_Min_Lifetime	16	12		0	409.5	0.1	s
FilterState_Min_Size	16	12		0	102.375	0.025	m²
FilterState_Min_ProbExists	16	12		0	7	1	0x0: 0% 0x1: 25% 0x2: 50% 0x3: 75% 0x4: 90% 0x5: 99% 0x6: 99.9% 0x7: 100%
FilterState_Min_Y	16	12	-409.5	-409.5	409.5	0.2	m
FilterState_Min_X	16	13	-500	-500	1138.2	0.2	m
FilterState_Min_VYLeftRight	16	12		0	128.993	0.0315	m/s
FilterState_Min_VXOncome	16	12		0	128.993	0.0315	m/s
FilterState_Min_VYRightLeft	16	12		0	128.993	0.0315	m/s
FilterState_Min_VXDepart	16	12		0	128.993	0.0315	m/s
FilterState_Max_NofObj	32	12		0	4095	1	
FilterState_Max_Distance	32	12		0	409.5	0.1	m
FilterState_Max_Azimuth	32	12	-50	-50	52.375	0.025	deg
FilterState_Max_VrelOncome	32	12		0	128.993	0.0315	m/s
FilterState_Max_VrelDepart	32	12		0	128.993	0.0315	m/s
FilterState_Max_RCS	32	12	-50	-50	52.375	0.025	dBm²
FilterState_Max_Lifetime	32	12		0	409.5	0.1	s
FilterState_Max_Size	32	12		0	102.375	0.025	m²
FilterState_Max_ProbExists	32	12		0	7	1	0x0: 0% 0x1: 25% 0x2: 50% 0x3: 75% 0x4: 90% 0x5: 99% 0x6: 99.9% 0x7: 100%
FilterState_Max_Y	32	12	-409.5	-409.5	409.5	0.2	m
FilterState_Max_X	32	13	-500	-500	1138.2	0.2	m
FilterState_Max_VYLeftRight	32	12		0	128.993	0.0315	m/s

FilterState_Max_VXOncome	32	12		0	128.993	0.0315	m/s
FilterState_Max_VYRightLeft	32	12		0	128.993	0.0315	m/s
FilterState_Max_VXDepart	32	12		0	128.993	0.0315	m/s

Table 16: FilterState_Cfg - signal description (0x204)

Start	Signal	Description
2	FilterState_Active	Filter activation switch
3	FilterState_Index	Multiplexor to specify which filter criterion to configure (0x0 to 0xE)
7	FilterState_Type	Multiplexor to choose between object or cluster filter configuration
16	FilterState_Min_xxx	Minimum value for filter criterion. The content depends on FilterState_Index (see Table 14 for more details)
32	FilterState_Max_xxx	Maximum value for filter criterion. The content depends on FilterState_Index (see Table 14 for more details)

7.4. Collision detection state (0x408)

Region-based collision detection can be activated with message ***CollDetCfg*** (0x400). When collision detection is active, the sensor sends the current collision detection configuration and warning state in message ***CollDetState*** (0x408) and for the individual regions in messages ***CollDetRegionState*** (0x402) cyclically (once per second).

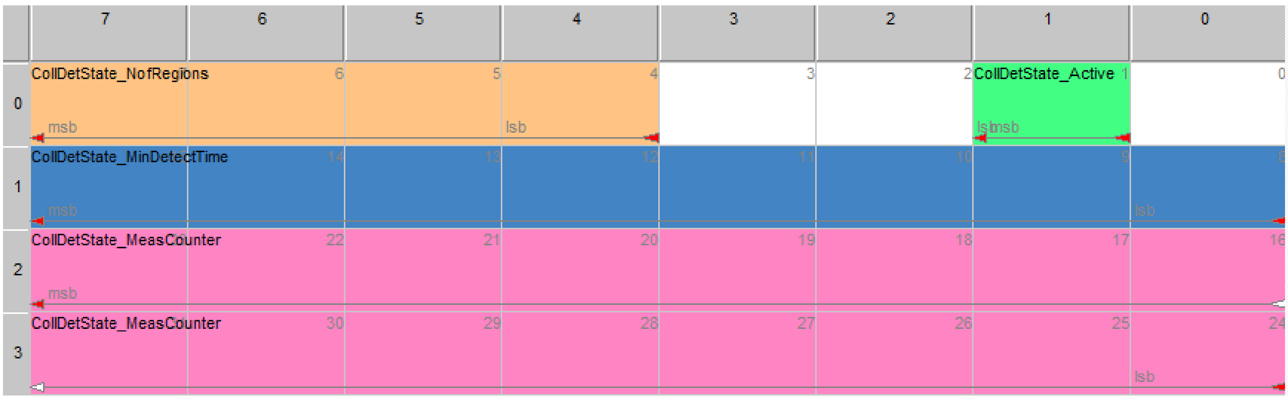


Figure 15: CollDetState - message layout (0x408)

Table 17: CollDetState - message content (0x408)

Signal	Start	Len	Min	Max	Res	Unit
CollDetState_Activation	1	1	0	1	1	0x0: inactive 0x1: active
CollDetState_NofRegions	4	4	0	8	1	-

CollDetState_MinDetectTime	8	8	0.0	25.5	0.1	sec
CollDetState_MeasCounter	24	16	0	65535	1	-

Table 18: CollDetState - signal description (0x408)

Start	Signal	Description
1	CollDetState_Activation	State of collision detection activation
4	CollDetState_NofRegions	Number of configured regions
8	CollDetState_MinDetectTime	Current configuration for minimum detection time of objects before a warning is triggered
24	CollDetState_MeasCounter	Measurement cycle counter (counting up since startup of sensor and restarting at 0 when > 65535)

7.5. Collision detection region state (0x402)

When collision detection is active, the sensor sends the current collision detection configuration and warning state in message ***CollDetState*** (0x408) and for the individual regions in messages ***CollDetRegionState*** (0x402) cyclically (once per second).

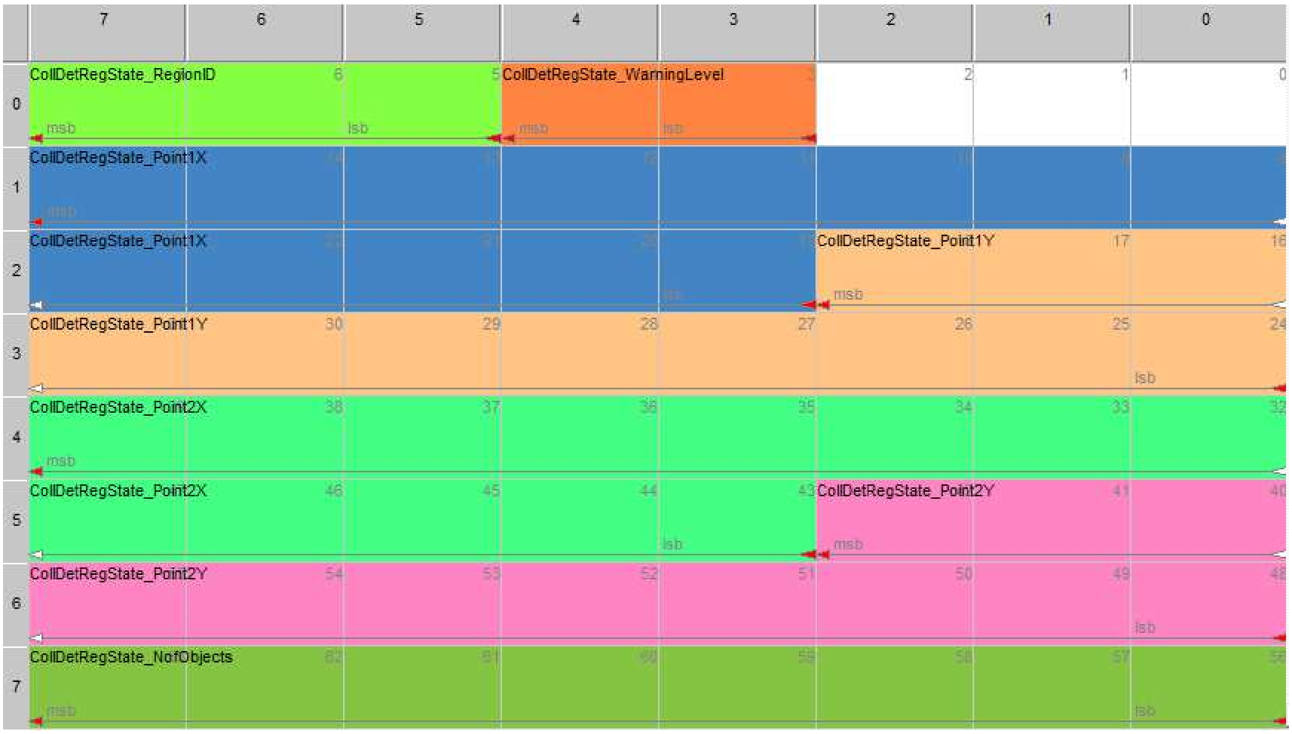


Figure 16: CollDetRegionState - message layout (0x402)

Table 19: CollDetRegionState - message content (0x402)

Signal	Start	Len	Offset	Min	Max	Res	Unit
CollDetRegState_WarningLevel	3	2		0	0	1	0x0: no warning 0x1: object warning 0x2: unused 0x3: warning deactivated
CollDetRegState_RegionID	5	3		0	7	1	-
CollDetRegState_Point1X	19	13	-500	-500	1138.2	0.2	m
CollDetRegState_Point1Y	24	11	-204.6	-204.6	204.8	0.2	m
CollDetRegState_Point2X	43	13	-500	-500	1138.2	0.2	m
CollDetRegState_Point2Y	48	11	-204.6	-204.6	204.8	0.2	m
CollDetRegState_NofObjects	56	8		0	255	1	-

Table 20: CollDetRegionState - signal description (0x402)

Start	Signal	Description
3	CollDetRegState_WarningLevel	Indicates that an object is inside the region or that it has ever been inside the region. If it is 0x0, no object has ever been detected inside the region. If it is 0x1, an object is currently inside the region and has exceeded the minimum detection time configured in CollDetCfg_MinTime. 0x2 keeps unused If it is 0x3, an object has been inside this region in the past, but has already left. To reset the warning level to 0x0, use CollDetCfg_WarningReset (see also section 6.3).
5	CollDetRegState_RegionID	ID of current region
19	CollDetRegState_Point1X	1st X (longitudinal) coordinate of rectangular region
24	CollDetRegState_Point1Y	1st Y (lateral) coordinate of rectangular region
43	CollDetRegState_Point2X	2nd X (longitudinal) coordinate of rectangular region
48	CollDetRegState_Point2Y	2nd Y (lateral) coordinate of rectangular region
56	CollDetRegState_NofObjects	Number of objects currently detected inside the region that fulfil the conditions for collision detection.

7.6. Control message for relay (0x8) (Configuration of ARS 40X, Relay, CRPLC)

The ARS 40X Collision detection supports a Wilke CAN Bus Relay for direct switching of external hardware. When collision detection is active and the relay control output is activated with ***RadarCfg_CtrlRelay***, the sensor sends the relay control message ***CollDetRelayCtrl*** (0x8) cyclically (once per second).

Four different relays (i.e. the first four collision regions) are supported. The module has to be terminated on the CAN bus on both ends with a 120 Ohm resistor.

Figure 1: CAN-Bus I/O Module: 4 digital outputs (Article Code DV-CANFRAS4-01)

To use this module it has to be configured to 500kBit/s. The module address has to be configured to 0x8 (Upper switch: 0, bottom switch: 1).

The sensors send the relay control message all with the same message ID (independently on the chosen sensor ID). Thus, only one sensor in a network of sensor should be sending this message. As response, the CAN Bus Relay sends out a cyclic status message as response with CAN ID 0x10 when it is working.

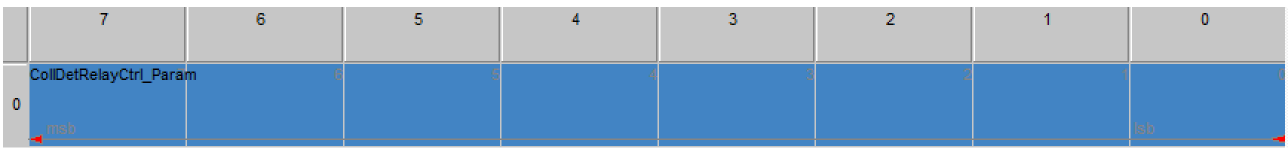


Figure 2: CollDetRelayCtrl - message layout (0x8)

Table 21: CollDetRelayCtrl - message content (0x8)

Signal	Start	Len	Min	Max	Res	Unit
CollDetRelayCtrl_Param	0	8	0	255	1	Bit 0: Relay 1 Bit 1: Relay 2 Bit 2: Relay 3 Bit 3: Relay 4 Bit 4: Reserved Bit 5: Reserved Bit 6: Reserved Bit 7: Reserved

Table 22: CollDetRelayCtrl - signal description (0x8)

Start	Signal	Description
0	CollDetRelayCtrl_Param	Control bit field for the 4 Relays on the module

7.6.1. Opportunity to have an ARS 40X connected with PLC & CAN Bus Relay:

Referring to the PLC manual, we can have very better view to the sensor visualization with the bird eye, the long region range and the near region range. However we can expand the opportunity for using the PLC by adding a relay. This relay simply will provide you with digital outputs. So through T-can-bus-connection as it is shown in the following schematic and picture (figures 17-18) can say that you will be able to increase the digital outputs of the PLC with full visualization on the screen.

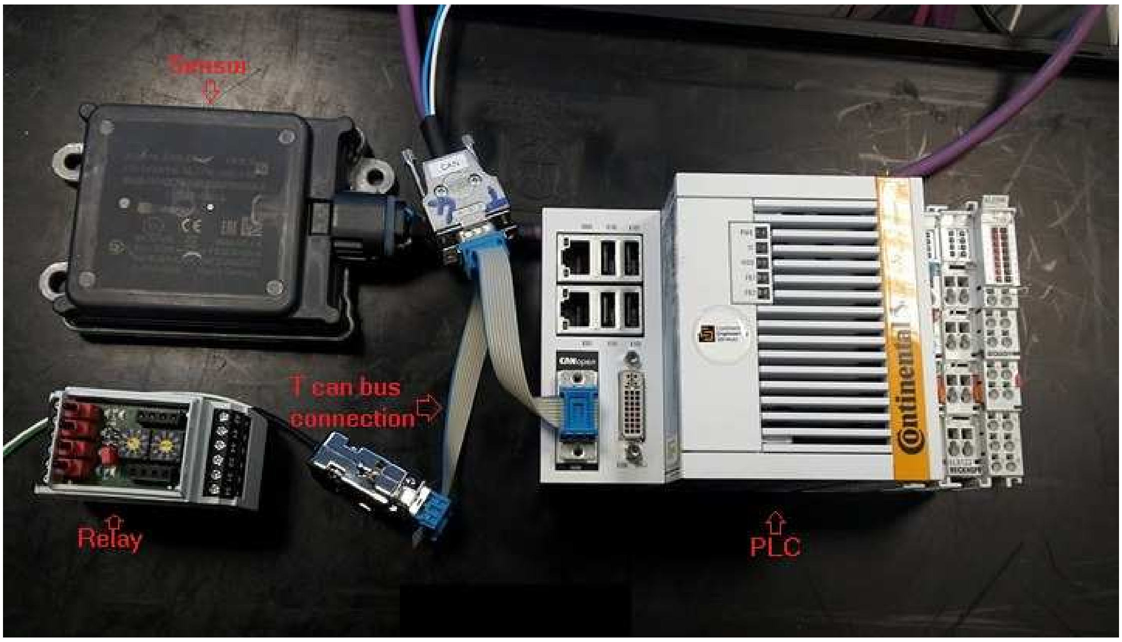


Figure 19: Real connection between the sensor, PLC and the relay.

7.6.2. Configuration of the connection between the PLC, ARS 40X-21 & Relay:

1. Make sure that you have connected the three devices through the T- can-bus-connection.
2. Make sure that you have connected the power for the ARS 40X & Relay.
3. Choose your sensor type through PLC interface.
4. Now through sensor configuration enable your relay control (Please follow PLC manual).
5. Now - this is only for example - activate the anti-collision configuration.
6. We have prepared for you a program.ini file. This is a test pattern file.
7. Program.ini has 4 overlapped regions that will detect any object and issue immediately and it will create an anti-collision warning.
8. You can find the Program.ini file in our documentation DVD and we are working on updates that will allow you in the future to prepare your own files with any regions you need and just choosing them with browsing.
9. Load your regions boundaries from Program.ini using PLC interface and send it to the sensor (same as 7.5 point in this manual but 7.5 is done with Canalyzer).
10. Now you can see through the screen (PLC) - in regions detected objects
11. You can also detect those objects through your relay trigger.
12. Every channel has one sensor and as much as relays with duplicate results.
13. Later you can add another CAN-Bus - channels and then more relays with more sensors

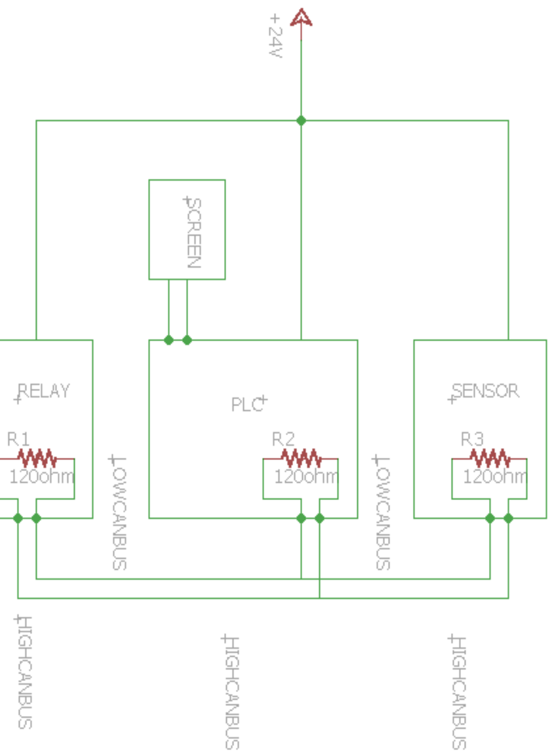


Figure 20: Block diagram shows connection between the sensor, PLC and the relay

7.7. Software Version Identification (0x700)

Figure 21: VersionID - message layout (0x700)

Table 23: VersionID - message content (0x700)

Signal	Start	Len	Min	Max	Res	Unit
Version_MajorRelease	0	8	0	256	1	
Version_MinorRelease	8	8	0	256	1	
Version_PatchLevel	16	8	0	256	1	

Version_ExtendedRange	25	1	0	1	1	0x0 – Standard Range Version 0x1 – Extended Range Version without any increase of the Power of Radiation
Version_CountryCode	24	1	0	1	1	0x0 – International Version 0x1 – South Korea, Japan – Power Reduction is demanded

Table 24: VersionID – signal description (0x700)

Start	Signal	Description
0	Version_MajorRelease	Software Main Version
8	Version_MinorRelease	Software Minor Version
16	Version_PatchLevel	Software Patch Level
24	Version_CountryCode	The version for South Korea / Japan has reduced power output of -3dB.
25	Version_ExtendedRange	The maximum range can be configured with the following limits. **ARS 408-21:** Standard Range: 96 m – 260 m Extended Range: 96 m – 1200 m **ARS 404-21:** Standard Range: 150 m – 190 m Extended Range: 90 m – 1000 m

7.8. Practical ways to use the module along with the sensor

Using the module can make your life easier. Having a digital signal from it can easily control another device, microcontroller or even feeding data to a C programming file.

Here, an example will be given on something we are wondering about all the time. An object is in the warning zones. You have the warning coming from your sensor but how can you handle this incoming data ? The answer is your module.

In section 6.4 “Collision detection region configuration (0x401)”, there are instructions how to activate that feature. However our example will make this better and better to understand.

Please follow the following instruction:

1. Open your communication software tool.
2. Choose the collDETregioncfg.
3. Activate and make it valid to send data.
4. Choose your first region (for example make your Y-axis constant y1 = -20, y2 = 20 and move your offset always in the positive X-axis with 2m).
5. Be careful: We are handling axis X and Y differently than well common known (please review figure 9).
6. You can choose many regions as much as the device allows.
7. Be careful you will find the minimum X-value you can feed is almost 0.6m.
8. Be careful: Once you set your first region → directly send it to the radar for memory addressing reasons→now set your second region and so.
9. Now read and understand figure 9 perfectly!
10. Figure 10 shows you how your settings will be sent and then shown in HEX.
11. Please make sure you understand what the conversions method are between hex, bin & dec.
12. If you do not understand this, do not worry the device is working perfectly. Due high testing restrictions we are applying.
13. If you do not believe us → connect your sensor now to the module and start trying your anti-collision regions that you have already set.
14. Your module relays will be activated according to the previous setup and you shall find the indication LEDs dancing around.
15. Now connect your module to whatever device you need from the respected output pins.

This previous example is only a demonstration how to use the module but you can simply use it in very wide areas than this.

8. Motion Input Signals

The sensor accepts input messages 0x300 and 0x301, but will still function without these two messages. After a 500 ms timeout, the sensor will default to the following states:

1. ***Speed*** 0 m/s and standstill
2. ***Yaw rate*** 0 deg/s

Each message has independent timeout monitoring. The timeout state is reflected in signal ***RadarState_MotionRxState*** (0x201).

The input signals are used to evaluate a vehicle course with is used to determine the movement of detected clusters and objects.

8.1. SpeedInformation (0x300)

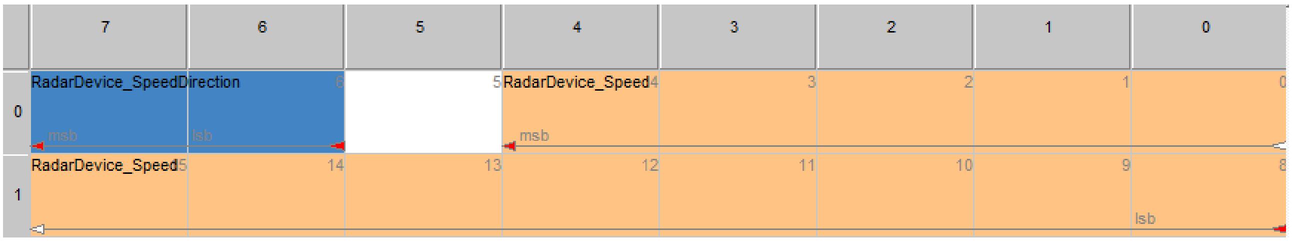


Figure 22: SpeedInformation - message layout (0x300)

Table 25: SpeedInformation - message content (0x300)

Signal	Start	Len	Min	Max	Res	Unit
RadarDevice_SpeedDirection	6	2	0	2	1	0x0: standstill 0x1: forward 0x2: backward
RadarDevice_Speed	8	13	0	163.8	0.02	m/s

Table 26: SpeedInformation – signal description (0x300)

Start	Signal	Description
6	RadarDevice_SpeedDirection	Indicates the direction of the radar movement while looking into positive straight ahead direction. Please, see also Chapter **5. Description**
8	RadarDevice_Speed	Absolute magnitude of speed in the direction the radar is moved while looking into positive straight ahead direction

8.2. Yaw rate Information (0x301)

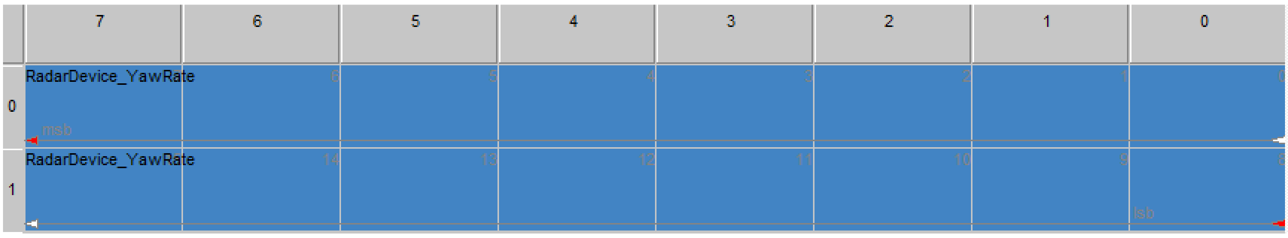


Figure 23: YawRateInformation - message layout (0x301)

Table 27: YawRateInformation - message content (0x301)

Signal	**Start**	**Len**	**Offset**	**Min**	**Max**	**Res**	**Unit**
RadarDevice_YawRate	8	16	+327.68	-327.68	327.68	0.01	deg/s

Table 28: YawRateInformation – signal description (0x301)

Start	**Signal**	**Description**
8	RadarDevice_YawRate	Rate of change of angular velocity looking into positive straight ahead direction. The center of rotation is assumed to be 1.95 m behind the sensor. Please, see also Chapter **5. Description**

9. Cluster List

The Cluster list output consists of up to three message definitions that are sent in a regular interval (about 70 to 80 ms) if clusters are selected in signal ***RadarCfg_OutputType*** (0x200).

1. ***Cluster_0_Status*** (0x600) – The first message contains list header information, i.e. the number of near scan clusters and number of far scan clusters that are sent afterwards.
2. ***Cluster_1_General*** (0x701) – This message contains the position and velocity of the clusters and is sent repeatedly for all the detected clusters (first near scan, then far scan). Each of the two cluster lists is range sorted. If there are more than 250 clusters, only the first 250 clusters are sent.
3. ***Cluster_2_Quality*** (0x702) – This message contains the quality information of the clusters and is only sent if it was activated in signal ***RadarCfg_SendQuality*** (0x200). It is sent repeatedly for all clusters in the same way as message ***Cluster_1_General*** (0x701).

If the quality information is sent, first all messages of type ***Cluster_1_General*** (0x701) are sent and afterwards all messages of type ***Cluster_2_Quality*** (0x702).

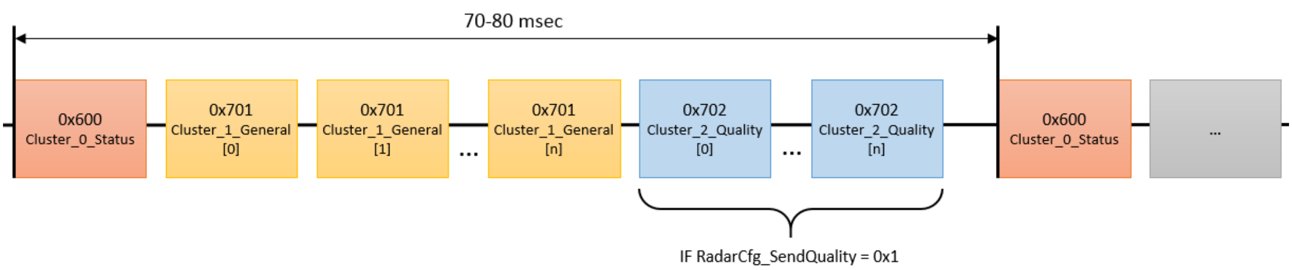


Figure 3: Overview of cluster list messages that are sent cyclically by the radar.

9.1. Cluster list status (0x600)

The message ***Cluster_0_Status*** (0x600) contains the cluster list header information and is sent as first message of the cluster list output and only once per measurement cycle.

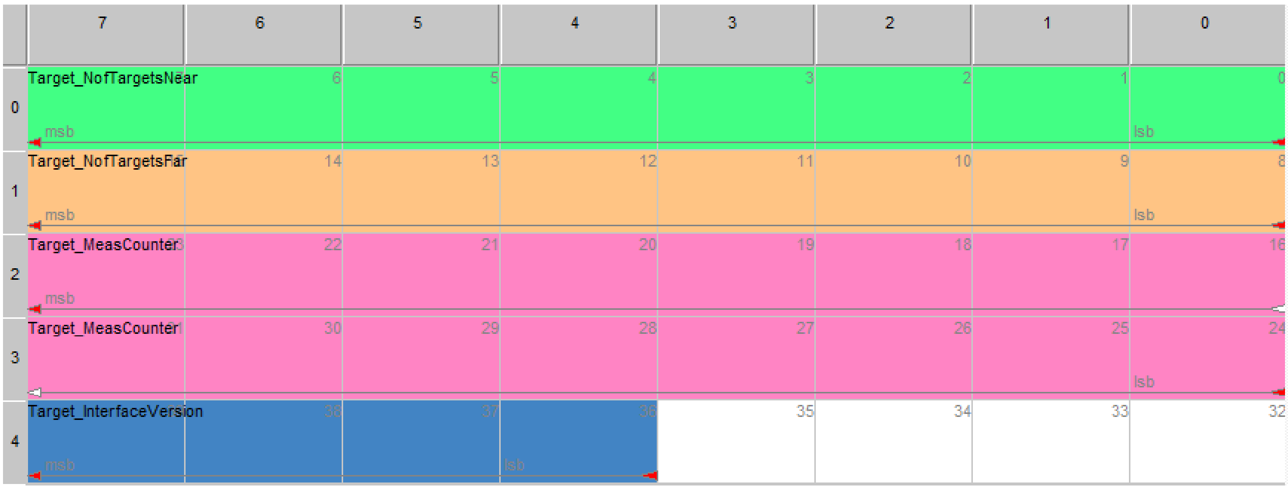


Figure 25: Cluster_0_Status - message layout (0x600)

Table 29: Cluster_0_Status - message content (0x600) ARS 404-21

Signal	Start	Len	Min	Max	Res	Unit
Cluster_NofClustersNear	0	8	0	255	1	-
Cluster_NofClustersFar	8	8	0	0		
Cluster_MeasCounter	24	16	0	65535	1	-
Cluster_InterfaceVersion	36	4	0	15	1	-

Table 30: Cluster_0_Status - message content (0x600) ARS 408-21

Signal	Start	Len	Min	Max	Res	Unit
Cluster_NofClustersNear	0	8	0	255	1	-
Cluster_NofClustersFar	8	8	0	255	1	-
Cluster_MeasCounter	24	16	0	65535	1	-
Cluster_InterfaceVersion	36	4	0	15	1	-

Table 31: Cluster_0_Status – signal description (0x600).

Start	Signal	Description
0	Cluster_NofClustersNear	Number of detected clusters in the near range scan (see below)
8	Cluster_NofClustersFar	Number of detected clusters in the far range scan

		(Near range and far range together max. 250 clusters; in case of activation of the quality mode app. 110 – 120 clusters before the CAN-Bus gets an overrun, which must be avoided) ARS 404-21: this value is not supported
24	Cluster_MeasCounter	Measurement cycle counter (counting up since startup of sensor and restarting at 0 when > 65535)
36	Cluster_InterfaceVersion	Cluster list CAN interface version

9.2. Cluster general information (0x701)

This message contains the position and velocity of the clusters and is sent repeatedly for all the detected clusters (first near scan, then far scan). Each of the two cluster lists is range sorted. If there are more than 250 clusters, only the first 250 clusters are sent.

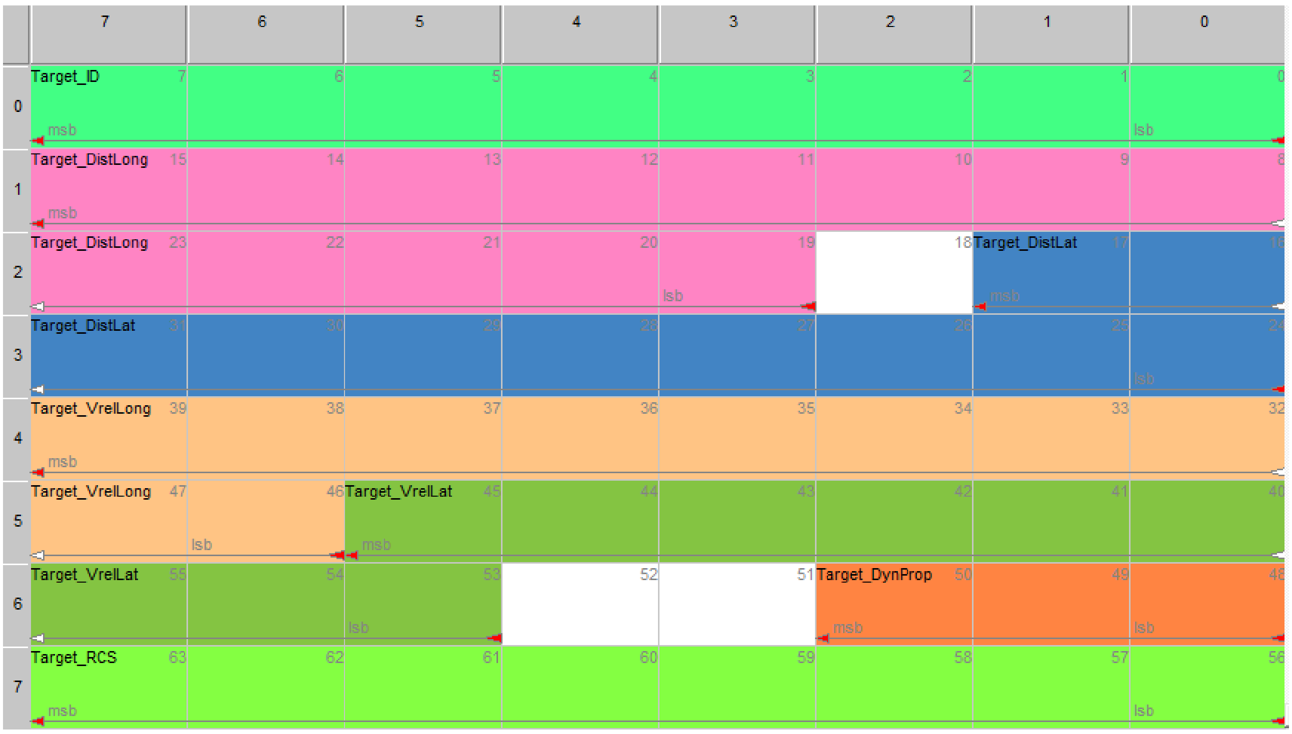


Figure 4: Cluster_1_General - message layout (0x701)

Table 32: Cluster_1_General - message content (0x701)

Signal	Start	Len	Offset	Min	Max	Res	Unit
Cluster_ID	0	8		0	255	1	
Cluster_DistLong	19	13	-500	-500	+1138.2	0.2	m
Cluster_DistLat	24	10	-102.3	-102.3	+102.3	0.2	m
Cluster_VrelLong	46	10	-128.00	-128.00	127.75	0.25	m/s

Cluster_DynProp	48	3		0	7	1	0x0: moving 0x1: stationary 0x2: oncoming 0x3: stationary candidate 0x4: unknown 0x5: crossing stationary 0x6: crossing moving 0x7: stopped
Cluster_VrelLat	53	9	-64.00	-64.00	63.75	0.25	m/s always “0”
Cluster_RCS	56	8	-64.0	-64.0	63.5	0.5	dBm²

Table 33: Cluster_1_General - signal description (0x701)

Start	Signal	Description
0	Cluster_ID	Cluster number
19	Cluster_DistLong	Longitudinal (x) coordinate
24	Cluster_DistLat	Lateral (y) coordinate
46	Cluster_VrelLong	Relative velocity in longitudinal direction (x)
53	Cluster_VrelLat	Relative velocity in lateral direction (y) This value is permanently set to “Zero”
48	Cluster_DynProp	Dynamic property of cluster to indicate if it is moving or not
56	Cluster_RCS	Radar cross section

9.3. Cluster quality information (0x702)

This message contains the quality information of the clusters and is only sent if it was activated in signal ***RadarCfg_SendQuality*** (0x200). It is sent repeatedly for all clusters in the same way as message ***Cluster_1_General*** (0x701). If this message is activated the maximum number of clusters has to be limited to 125 either by filtering or environment.

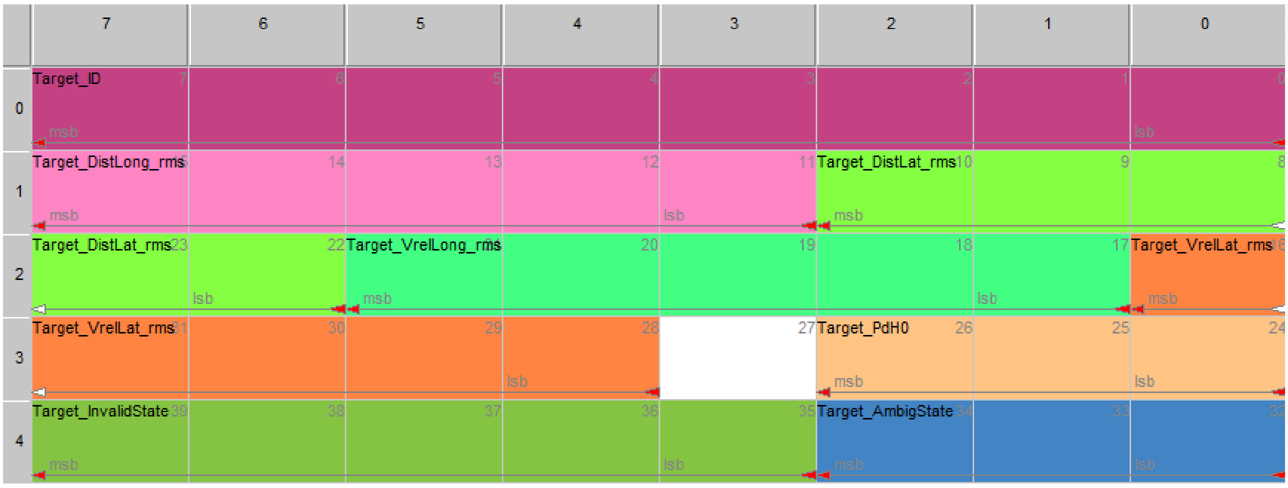


Figure 5: Cluster_2_Quality - message layout (0x702)

Table 34: Cluster_2_Quality - message content (0x702)

Signal	Start	Len	Min	Max	Res	Unit
Cluster_ID	0	8	0	255	1	-
Cluster_DistLong_rms	11	5	0	31	1	see Table 36
Cluster_VrelLong_rms	17	5	0	31	1	see Table 36
Cluster_DistLat_rms	22	5	0	31	1	see Table 36
Cluster_Pdh0	24	3	0	7	1	0x0: invalid 0x1: <25% 0x2: <50% 0x3: <75% 0x4: <90% 0x5: <99% 0x6: <99.9% 0x7: <=100%
Cluster_VrelLat_rms	28	5	0	31	1	see Table 36

Cluster_AmbigState	32	3	0	7	1	0x0: invalid (cannot be used) 0x1: ambiguous (cluster is not clear as there is fuzziness (darkness, ambiguity)). (Should not be used and not recommended anyway, because it could be there twice or even more times) 0x2: staggered ramp (the cluster is not clear somehow as the fuzziness (ambiguity) has been resolved partially). (Should not be used and not recommended anyway) 0x3: unambiguous (everything is clear, the fuzziness(ambiguity) has been resolved – it can be used and recommended for usage) 0x4: stationary candidates (everything is clear and probably there is something stationary (fixed) as the fuzziness(ambiguity) has been resolved – it can be used and recommended for usage)

Cluster_InvalidState	35	5	0	31	1	0x00: Valid 0x01: Invalid due to low RCS 0x02: Invalid due to near-field artefact 0x03: Invalid far range Cluster because not confirmed in near range 0x04: Valid Cluster with low RCS 0x05: reserved 0x06: Invalid Cluster due to high mirror probability 0x07: Invalid because outside sensor field of view 0x08: Valid Cluster with azimuth correction due to elevation 0x09: Valid Cluster with high child probability 0x0A: Valid Cluster with high probability of being a 50 deg artefact 0x0B: Valid Cluster but no local maximum 0x0C: Valid Cluster with high artefact probability 0x0D: reserved 0x0E: Invalid Cluster because it is a harmonics 0x0F: Valid Cluster above 95 m in near range 0x10: Valid Cluster with high multi-target probability 0x11: Valid Cluster with suspicious angle

Table 35: Cluster_2_Quality - signal description (0x702)

Start	Signal	Description
0	Cluster_ID	Cluster number
11	Cluster_DistLong_rms	Standard deviation of longitudinal distance
22	Cluster_DistLat_rms	Standard deviation of lateral distance
17	Cluster_VrelLong_rms	Standard deviation of longitudinal relative velocity
28	Cluster_VrelLat_rms	Standard deviation of lateral relative velocity

24	Cluster_Pdh0	False alarm probability of Cluster (i.e. probability for being an artefact caused by multipath or similar)
32	Cluster_AmbigState	State of Doppler (radial velocity) ambiguity solution (see the seventh raw in table 34 for more details)
35	Cluster_InvalidState	State of Cluster validity state (see Table 34 for more details)

Table 36: Signal value table for Cluster_DistLong_rms, Cluster_DistLat_rms, Cluster_VrelLong_rms and Cluster_VrelLat_rms (0x702).

Parameter	**Value for signals** **Cluster_DistLong_rms, Cluster_DistLat_rms [m]** **Cluster_VrelLong_rms, Cluster_VrelLat_rms [m/s]**
0x0	<0.005
0x1	<0.006
0x2	<0.008
0x3	<0.011
0x4	<0.014
0x5	<0.018
0x6	<0.023
0x7	<0.029
0x8	<0.038
0x9	<0.049
0xA	<0.063
0xB	<0.081
0xC	<0.105
0xD	<0.135
0xE	<0.174
0xF	<0.224
0x10	<0.288
0x11	<0.371
0x12	<0.478
0x13	<0.616
0x14	<0.794
0x15	<1.023
0x16	<1.317
0x17	<1.697
0x18	<2.187
0x19	<2.817
0x1A	<3.630

0x1B	<4.676
0x1C	<6.025
0x1D	<7.762
0x1E	<10.000
0x1F	invalid

10. Object List

The object list output consists of up to five message definitions that are sent in a regular interval if objects are selected in signal ***RadarCfg_OutputType*** (0x200).

1. ***Object_0_Status*** (0x60A) – The first message contains list header information, i.e. the number of objects that are sent afterwards
2. ***Object_1_General*** (0x60B) – This message contains the position and velocity of the objects and is sent repeatedly for all the tracked objects.
3. ***Object_2_Quality*** (0x60C) – This message contains the quality information of the objects and is only sent if it was activated in signal ***RadarCfg_SendQuality*** (0x200). It is sent repeatedly for all objects in the same way as message ***Object_1_General*** (0x60B).
4. ***Object_3_Extended*** (0x60D) – This message contains additional object properties and is only sent if it was activated in signal ***RadarCfg_SendExtInfo*** (0x200). It is sent repeatedly for all objects in the same way as message ***Object_1_General*** (0x60B).
5. ***Object_4_Warning*** (0x60E). This message contains the collision detection warning state and is only sent if collision detection was activated in message ***CollDetCfg*** (0x400). It is sent repeatedly for all objects in the same way as message ***Object_1_General*** (0x60B).

If the quality information, extended information and/or warning state is sent, first all messages of type ***Object_1_General*** (0x60B) are sent and afterwards all messages of type ***Object_2_Quality*** (0x60C), afterwards of type ***Object_3_Extended*** (0x60D) and/or afterwards of type ***Object_4_Warning*** (0x60E).

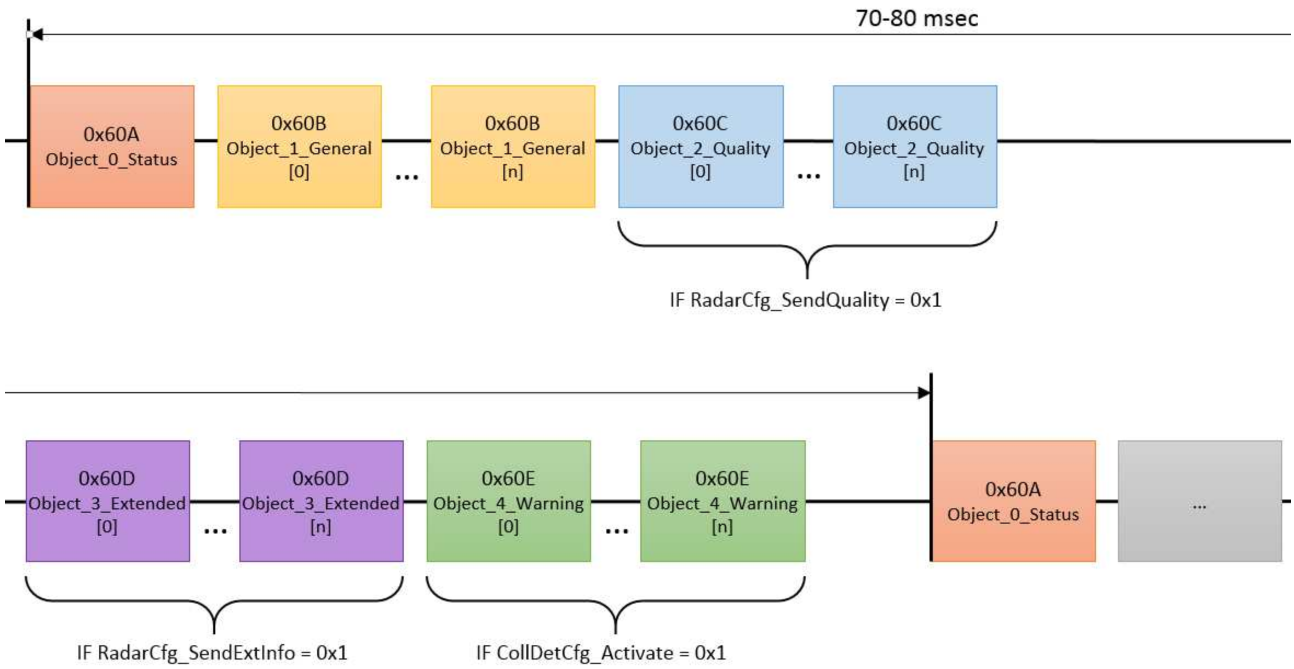


Figure 28: Overview of object list messages that are sent cyclically by the radar

10.1. Object list status (0x60A)

The message ***Object_0_Status*** (0x60A) contains the object list header information and is sent as first message of the cluster list output and only once per measurement cycle.

Figure 29: Object_0_Status - message layout (0x60A)

Table 37: Object_0_Status - message content (0x60A)

Signal	Start	Len	Min	Max	Res	Unit
Object_NofObjects	0	8	0	255	1	-
Object_MeasCounter	16	16	0	65535	1	-
Object_InterfaceVersion	28	4	0	15	1	-

Table 38: Object_0_Status - signal description (0x60A)

Start	Signal	Description
0	Object_NofObjects	Number of objects (max. 100 Objects)
16	Object_MeasCounter	Measurement cycle counter (counting up since startup of sensor and restarting at 0 when > 65535)
28	Object_InterfaceVersion	Object list CAN interface version

10.2. Object general information (0x60B)

This message contains the position and velocity of the objects and is sent repeatedly for all the tracked objects.

Figure 30: Object_1_General - message layout (0x60B)

Table 39: Object_1_General - message content (0x60B)

Signal	Start	Len	Offset	Min	Max	Res	Unit
Object_ID	0	8		0	255	1	
Object_DistLong	19	13	-500	-500	+1138.2	0.2	m
Object_DistLat	24	11	-204.6	-204.6	+204.8	0.2	m
Object_VrelLong	46	10	-128.00	-128.00	127.75	0.25	m/s
Object_DynProp	48	3		0	7	1	0x0: moving 0x1: stationary 0x2: oncoming 0x3: stationary candidate 0x4: unknown 0x5: crossing stationary 0x6: crossing moving 0x7: stopped
Object_VrelLat	53	9	-64.00	-64.00	63.75	0.25	m/s
Object_RCS	56	8	-64.0	-64.0	63.5	0.5	dBm²

Table 40: Object_1_General - signal description (0x60B)

Start	Signal	Description
0	Object_ID	Object ID (since objects are tracked, the ID is kept throughout measurement cycles and does not have to be consecutive)
19	Object_DistLong	Longitudinal (x) coordinate
24	Object_DistLat	Lateral (y) coordinate
46	Object_VrelLong	Relative velocity in longitudinal direction (x) stationary candidates => vrel (y) made by own speed and yaw rate => rest of speed projected to (x) Math.: f_VrelY = - (f_DistX + LongPosToRot) * YawRate f_VrelX = (f_VrelRad - sin_(f_AzAngle) * f_VrelY)/cos_(f_AzAngle) otherwise (moving candidates): => projected to (x) Math.: f_VrelY = 0.0 f_VrelX = f_VrelRad/COS_(f_AzAngle) if cos_(f_AzAngle) = 0, => f_VrelRad is used
53	Object_VrelLat	Relative velocity in lateral direction (y) (see (x) above)
48	Object_DynProp	Dynamic property of the object indicating if the object is moving or stationary (this value can only be determined correctly if the speed and yaw rate is given correctly)
56	Object_RCS	Radar cross section

10.3. Object quality information (0x60C)

This message contains the quality information of the objects and is only sent if it was activated in signal ***RadarCfg_SendQuality*** (0x200). It is sent repeatedly for all objects in the same way as message ***Object_1_General*** (0x60B).

Figure 31: Object_2_Quality - message layout (0x60C)

Table 41: Object_2_Quality - message content (0x60C)

Signal	Start	Len	Min	Max	Res	Unit
Obj_ID	0	8	0	255	1	
Obj_DistLong_rms	11	5	0	31	1	see Table 43
Obj_VrelLong_rms	17	5	0	31	1	see Table 43
Obj_DistLat_rms	22	5	0	31	1	see Table 43
Obj_VrelLat_rms	28	5	0	31	1	see Table 43
Obj_ArelLat_rms	34	5	0	31	1	see Table 43
Obj_ArelLong_rms	39	5	0	31	1	see Table 43
Obj_Orientation_rms	45	5	0	31	1	deg

Obj_MeasState	50	3	0	7	1	0x0: deleted 0x1: new 0x2: measured 0x3: predicted 0x4: deleted for merge 0x5: new from merge
Obj_ProbOfExist	53	3	0	7	1	0x0: invalid 0x1: <25% 0x2: <50% 0x3: <75% 0x4: <90% 0x5: <99% 0x6: <99.9% 0x7: <=100%

Table 42: Object_2_Quality – signal description (0x60C)

Start	Signal	Description
0	Obj_ID	Object ID (since objects are tracked, the ID is kept throughout measurement cycles and does not have to be consecutive)
11	Obj_DistLong_rms	Standard deviation of longitudinal distance
17	Obj_VrelLong_rms	Standard deviation of longitudinal relative velocity
22	Obj_DistLat_rms	Standard deviation of lateral distance
28	Obj_VrelLat_rms	Standard deviation of lateral relative velocity
34	Obj_ArelLat_rms	Standard deviation of lateral relative acceleration
39	Obj_ArelLong_rms	Standard deviation of longitudinal relative acceleration
45	Obj_Orientation_rms	Standard deviation of orientation angle (see: figure 3)

50	Obj_MeasState	Measurement state indicating if the object is valid and has been confirmed by clusters in the new measurement cycle: 0x0: Deleted Object – Object has been deleted – is displayed during the last cycles of transmission just before the object ID disappears. 0x1: New Object is created – is displayed during the first cycles of transmission just after the object ID is created. Can also be checked by the character *Object_MeasCounter*. 0x2: Measured – Object creation has been confirmed by the actual measurement. Cluster could be created. 0x3: Predicted - Object creation could not be confirmed by the actual measurement. Cluster could not be created. 0x4: Deleted for merge - Object became deleted in order to be merged with another Object. 0x5: New from merge – new Object became created after a merge.
53	Obj_ProbOfExist	Probability of existence

Table 43: Signal value table for Obj_Orientation_rms, Obj_DistLong_rms, Obj_DistLat_rms, Obj_VrelLong_rms, Obj_VrelLat_rms, Obj_ArelLat_rms, Obj_ArelLong_rms (0x60C).

Parameter	**Values for signal** **Obj_Orientation_rms [deg]**	**Value for signals** **Obj_DistLong_rms, Obj_DistLat_rms [m]** **Obj_VrelLong_rms, Obj_VrelLat_rms [m/s]** **Obj_ArelLat_rms, Obj_ArelLong_rms [m/s²]**
0x0	<0.005	<0.005
0x1	<0.007	<0.006
0x2	<0.010	<0.008
0x3	<0.014	<0.011
0x4	<0.020	<0.014
0x5	<0.029	<0.018
0x6	<0.041	<0.023
0x7	<0.058	<0.029
0x8	<0.082	<0.038
0x9	<0.116	<0.049
0xA	<0.165	<0.063
0xB	<0.234	<0.081
0xC	<0.332	<0.105

0xD	<0.471	<0.135
0xE	<0.669	<0.174
0xF	<0.949	<0.224
0x10	<1.346	<0.288
0x11	<1.909	<0.371
0x12	<2.709	<0.478
0x13	<3.843	<0.616
0x14	<5.451	<0.794
0x15	<7.734	<1.023
0x16	<10.971	<1.317
0x17	<15.565	<1.697
0x18	<22.081	<2.187
0x19	<31.325	<2.817
0x1A	<44.439	<3.630
0x1B	<63.044	<4.676
0x1C	<89.437	<6.025
0x1D	<126.881	<7.762
0x1E	<180.000	<10.000
0x1F	invalid	invalid

10.4. Object extended information (0x60D)

This message contains additional object properties and is only sent if it was activated in signal ***RadarCfg_SendExtInfo*** (0x200). It is sent repeatedly for all objects in the same way as message ***Object_1_General*** (0x60B).

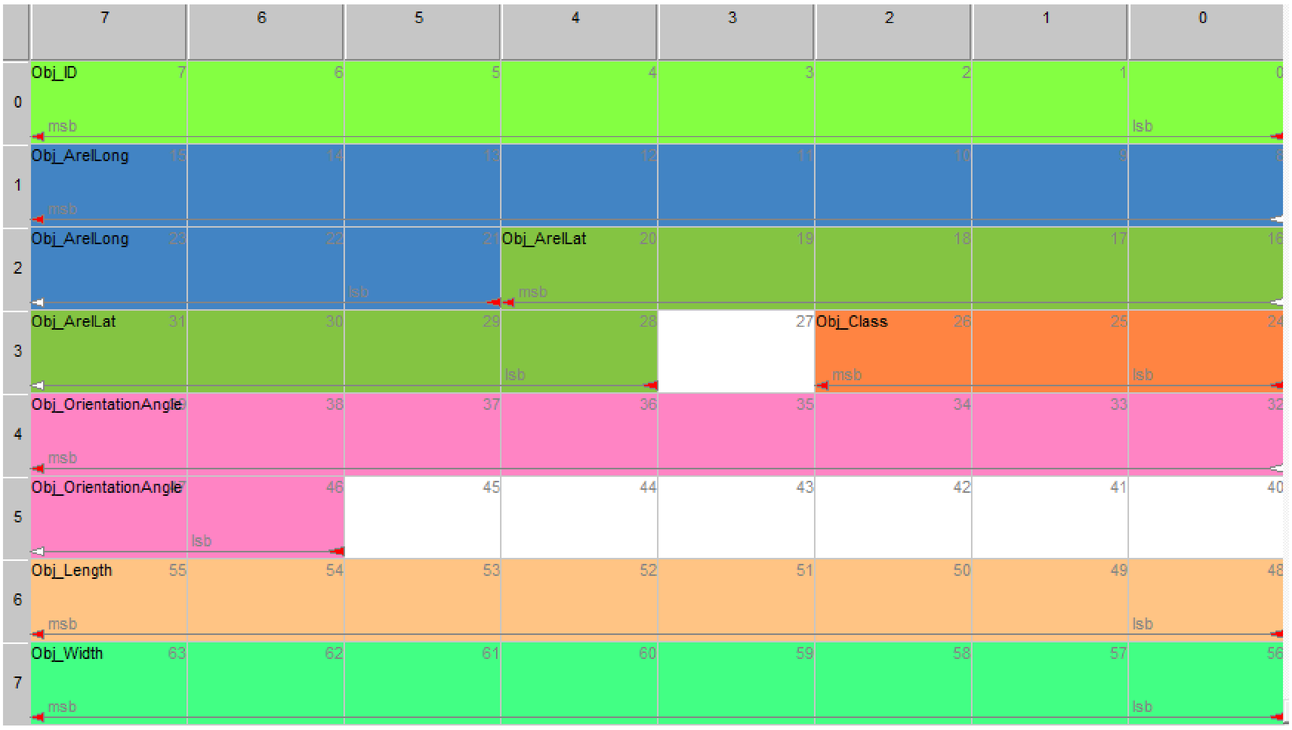


Figure 32: Object_3_Extended - message layout (0x60D)

Table 44: Object_3_Extended - message content (0x60D)

Signal	Start	Len	Offset	Min	Max	Res	Unit
Object_ID	0	8		0	255	1	
Object_ArelLong	21	11	-10.00	-10.00	10.47	0.01	m/s²
Object_Class	24	3		0	7	1	0x0: point 0x1: car 0x2: truck 0x3: not in use 0x4: motorcycle 0x5: bicycle 0x6: wide 0x7: reserved
Object_ArelLat	28	9	-2.50	-2.50	2.61	0.01	m/s²
Object_OrientationAngel	46	10	-180.00	-180.00	180.00	0.4	deg
Object_Length	48	8		0.0	51.0	0.2	m
Object_Width	56	8		0.0	51.0	0.2	m

Table 45: Object_3_Extended - signal description (0x60D)

Start	Signal	Description
0	Object_ID	Object ID (since objects are tracked, the ID is kept throughout measurement cycles and does not have to be consecutive)
21	Object_ArelLong	Relative acceleration in longitudinal direction
24	Object_Class	0x0: point 0x1: car 0x2: truck 0x3: not in use 0x4: motorcycle 0x5: bicycle 0x6: wide 0x7: reserved
28	Object_ArelLat	Relative acceleration in lateral direction
46	Object_OrientationAngel	Orientation angle of the object (see: Figure 1)
48	Object_Length	Length of the tracked object
56	Object_Width	Width of the tracked object

10.5. Object collision detection warning (0x60E)

This message contains the collision detection warning state and is only sent if collision detection was activated in message ***CollDetCfg*** (0x400). It is sent repeatedly for all objects in the same way as message ***Object_1_General*** (0x60B).

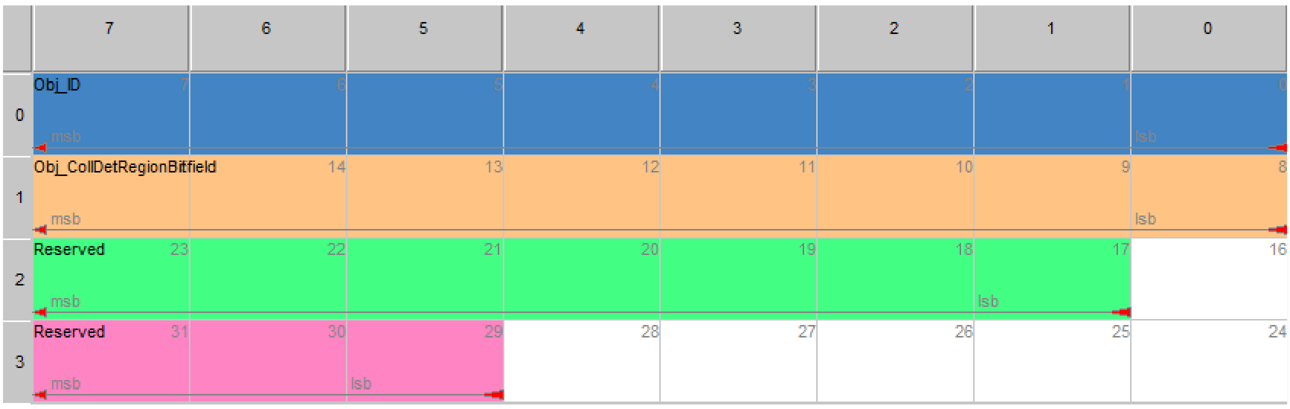


Figure 33: Object_4_Warning - message layout (0x60E)

Table 46: Object_4_Warning - message content (0x60E)

Signal	Start	Len	Min	Max	Res	Unit
Object_ID	0	8	0	255	1	
Object_CollDetRegionBitfield	8	8	0	255	1	

Table 47: Object_4_Warning - signal description (0x60E)

Start	Signal	Description
0	Object_ID	Object ID (since objects are tracked, the ID is kept throughout measurement cycles and does not have to be consecutive)
8	Object_CollDetRegionBitfield	Bit field of the regions, with bit set to 1 for regions that have a collision with this object