Continental Engineering Services GmbH

Standardized SRR Interface

Technical Documentation

SRR 208-21

SS / HS

"Standard Sensitivity / High Sensitivity"

TC / CL

***"Tracks and Collision Avoidance / Cluster** (or Tracks without collision avoidance)"*

Version: 1.7

This document Vers.: 1.7 is related to
SRR 208 Software Vers.: 1.0.3.0.

Table of Contents

History

Version	Date	Chapter	Change description
V1.0	21st Nov. 2012		Initial revision after takeover from ARS308
V1.1	23rd Nov. 2012		Adaptations for SRR completed
V1.2	29th Nov. 2012		Changed signal resolution and offset of various CAN signals according to the latest dbc file
V1.3	25th Dec. 2012		Alignment of objects in figure 1
V1.4	17th Dec. 2012		Different corrections
V1.5	7th May 2013		Change of Type names, Drawing 1
V1.6	16th July 2014		Change of diff. Rolling Counter values and CD Warn Track Index
V1.7	15th Feb. 2018		Track speed description added

1. Introduction

The SRR 2 is a **S**hort **R**ange **R**adar Sensor System developed by Continental for the Automotive Industry to realize advanced driver assistance functions. The usual interface of this system is mainly based on indication requests to the vehicle network. For example, in order to indicate, that a vehicle is entering the blind spot of the ego vehicle.

The software of the sensor was adapted to use it also for general purposes. With the simple 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. The SRR sensor with this general purpose software interface is referred to as SRR 208.

There are **four** different types of sensors:

the **SRR 208-21 SS/CL** and the **SRR 208-21 HS/CL** (Standard Sensitivity > SS or High Sensitivity > HS transmitting Clusters or Tracks > CL (no collision avoidance))

and

the **SRR 208-21 SS/TC** and the **SRR 208-21 HS/TC** (Standard Sensitivity > SS or High Sensitivity > HS transmitting all Tracks which have been responsible for a collision warning (including Collision Avoidance) > TC).

2. Operating Conditions

Please refer to *SRR 20X -Technical Description = Short Description.*

3. Safety Information

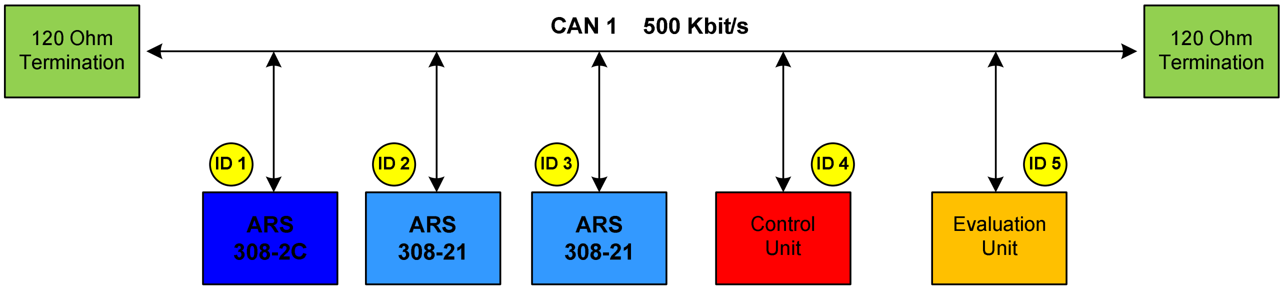
Please refer to *SRR 20X -Technical Description = Short Description.*

4. Interface

The SRR sensor has one CAN interface (called CAN1). The communication network is a CAN bus as specified in ISO 11898-2 with a transmission rate of 500 KBits/s. The messages on the CAN-Bus are defined as described below.

As an example, a possible CAN bus network is shown in drawing 1 below. Since no termination resistors are included in the SRR 208 sensor, two 120 Ohm terminal resistors have to be connected to the network (separately or integrated in the CAN interface of the corresponding unit). The CAN1 connection of the SRR 208 is used for configuration, sensor state output and data output.

Drawing 1: CAN network (here: taken from the Radar ARS)



5. Description

The SRR sensor uses radar radiation to analyze its surrounding. The reflected signals are processed and after multiple steps they are available in form of clusters and tracks. A cluster consists of multiple reflections which have a similar position and movement and therefore can be combined. The information about a cluster like relative velocity and position is then transmitted on CAN1. The position is represented in an angular coordinate system, thus by distance and azimuth angle relative to the sensor. The clusters are newly evaluated every cycle. In contrast to this, tracks have a history and consist of clusters tracked over time. The position of the object is calculated relative to the sensor and output in a Cartesian coordinate system.

here: Radar seen from the front side

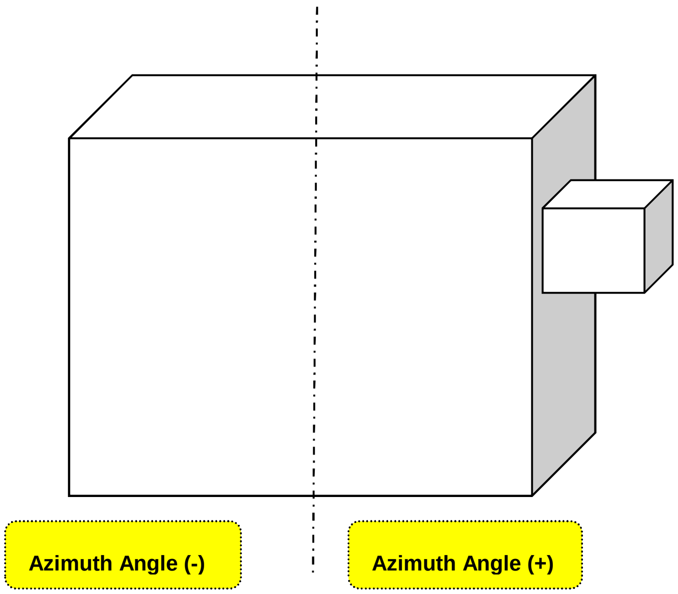


Figure 1: Sign Convention for the Azimuth Angle

CAN	Frame Format	Message ID[1]	Message Name	Content	Section
1	CAN 2.0A (11 Bit)	0x200	RadarConfiguration	Sensor configuration	6
1	CAN 2.0A (11 Bit)	0x60A	RadarStatus	Sensor Status output	7
1	CAN 2.0A (11 Bit)	0x70B	CAN1_ClusterStatus	Cluster status	8.1
1	CAN 2.0A (11 Bit)	0x70C	CAN1_Cluster_1	Cluster information 1	8.1
1	CAN 2.0A (11 Bit)	0x60B	CAN1_TrackStatus	Track status	8.2
1	CAN 2.0A (11 Bit)	0x60C	CAN1_Track_1	Track information 1	8.2
1	CAN 2.0A (11 Bit)	0x60D	CAN1_Track_2	Track information 2	8.2
1	CAN 2.0A (11 Bit)	0x400	CollDetConfig	Collision detection configuration	9.1
1	CAN 2.0A (11 Bit)	0x408	CollDetState	Collision detection state	9.2
1	CAN 2.0A (11 Bit)	0x409	CollDetWarn	Collision detection: warning tracks	9.3

Table 1: Sensor CAN messages (depending on the sensor variant not all messages are available)

6. Sensor Configuration

The SRR 208 can be configured via message 0x200 on CAN1 (see: Figure 2 and Table 2). This message should only be transmitted to SRR 208 when a parameter change is desired. It is not necessary to transmit this message cyclically. It is necessary to transmit the DLC of each identifier all 8 Bytes, even if there are unused Bits and Bytes.

The following parameters can be changed:

1. The type of the output list
   a. ***SendTracks*** – The track list will be transmitted on CAN1
   b. ***SendCluster*** – The cluster list will be transmitted on CAN1

[1] This is the base message ID for sensor ID 0. For other sensor IDs this is:
(*base message ID* | (0x10 * *sensor ID*))

2. Unique Sensor ID – For each sensor on the same CAN-Bus an ID between 0 and 7 can be assigned. The sensor then sends and receives its messages in a CAN message ID range, that depends on its Sensor ID. The Sensor ID is coded in the second digit of the message ID: (*base message ID* | (0x10 * *sensor ID*)). *Example:*
   Sensor ID **0**: RadarStatus is sent on 0x6**0**A, Configuration is received on 0x2**0**0 etc.
   Sensor ID **1**: RadarStatus is sent via 0x6**1**A, Configuration is received on 0x2**1**0 etc.

3. When the sensor ID is invalid only the *RadarStatus* message is transmitted. In case the user tries to reconfigure a sensor ID, which is already in use, the sensor will keep its old ID and the signal value of Radar_Cfg_Status will change to 0x02 (Desired ID already used).

   Within this documentation it is assumed that the sensor is configured with the Sensor ID 0.

These parameters can be changed individually or both at the same time. For each of the above parameters, the message contains a validity bit. If the validity bit is set to ***Valid/True***, the corresponding parameter will be updated in the SRR, otherwise it is ignored. If a parameter is updated, it will also be stored in non-volatile memory so that it is automatically set at startup on any subsequent power up.

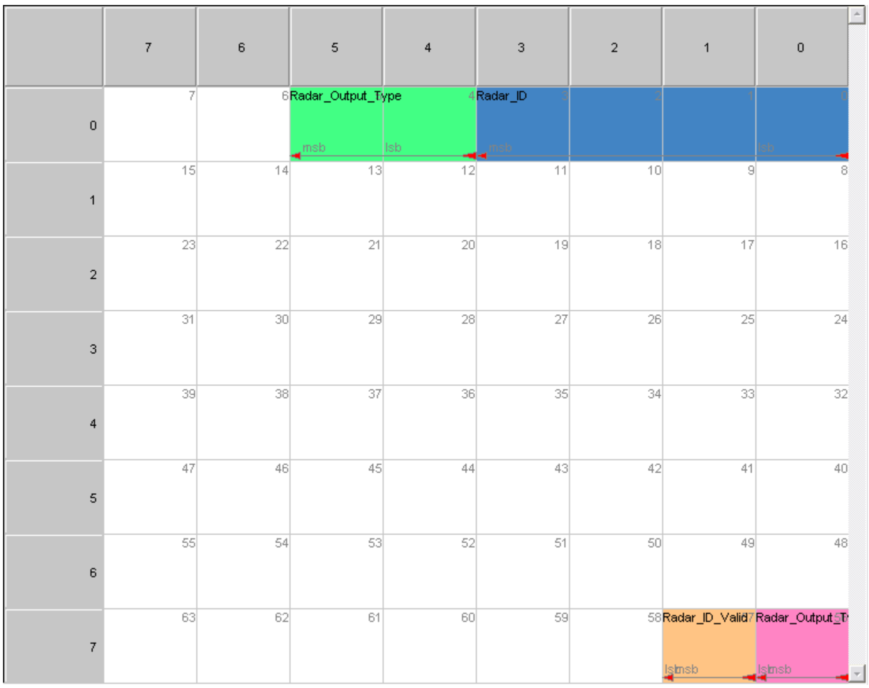


Figure 2: Radar Configuration message structure (0x200)   (Bytes 2-6 and Bits in white are unused)

Signal	Start	Len	Byte Order	Value Type	Res	Value Range
Radar_Output_Type_Valid	56	1	Motorola	Unsigned	1	▪ 0 -> Invalid ▪ 1 -> Valid
Radar_Output_Type	4	2	Motorola	Unsigned	1	▪ 0 -> (0 x 0) SendTracks ▪ 1 -> (0 x 1) SendCluster
Radar_ID_Valid	57	1	Motorola	Unsigned	1	▪ 0 -> False ▪ 1 -> True
Radar_ID	0	4	Motorola	Unsigned	1	0 ... 7

Table 2: Radar Configuration - message description (0x200)

Signal	Start	Len	Byte Order	Value Type	Res	Default Factory Settings
Radar_Output_Type_Valid	56	1	Motorola	Unsigned	1	▪ 0 -> Invalid
Radar_Output_Type	4	2	Motorola	Unsigned	1	▪ 1 -> Send Cluster
Radar_ID_Valid	57	1	Motorola	Unsigned	1	▪ 0 -> Invalid
Radar_ID	0	4	Motorola	Unsigned	1	▪ 0

Table 3: Radar Configuration message - Default factory settings (0x200)

7. Sensor Status Output

Message 0x60A contains signals which report the state of the sensor.

After configuring a parameter, (except for the track/cluster selection) by sending message 0x200 to the SRR 208-21_ _/_ _, the signal Radar_Cfg_Status in message 0x60A can be checked in order to verify that the configuration change was accepted.

Figure shows the layout of the 0x60A message, while Table 4 shows the details of each signal.

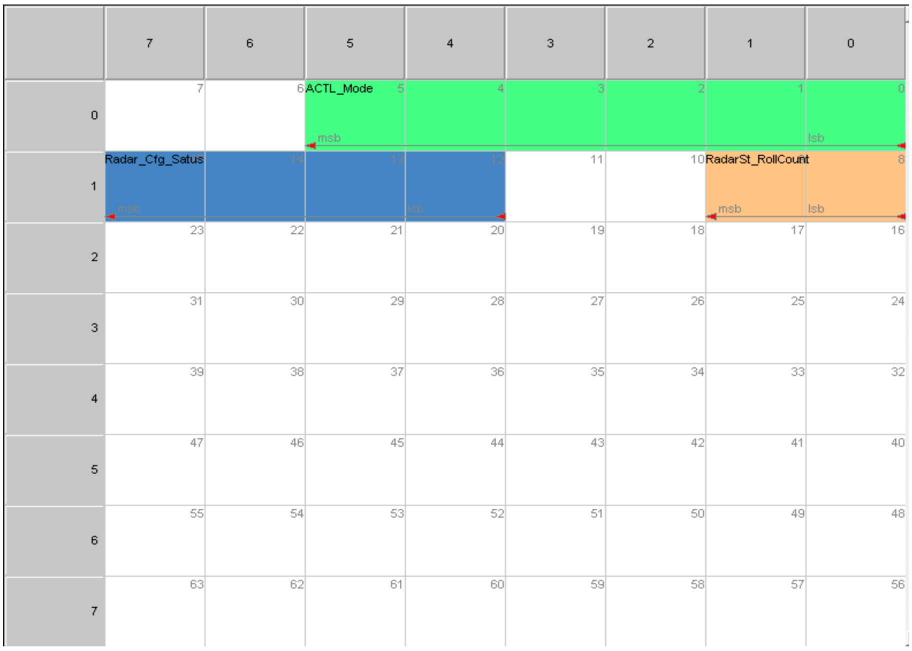


Figure 3: RadarStatus message Structure (0x60A) (Bits in white are unused)

Signal	Start	Len	Byte Order	Value Type	Res	Value Range
ACTL_Mode	0	6	Motorola	Unsigned	1	▪ 0 -> Unused ▪ 1 -> Normal operation ▪ 2 -> Hardware defect ▪ 3 -> Unused ▪ 4 -> Appl. Initialisation ▪ 5 -> Over temperature
Radar_Cfg_Status	12	4	Motorola	Unsigned	1	▪ 0 -> Last learned cfg used ▪ 1 -> New config is used ▪ 2 -> Desired ID already used ▪ 3 -> Nvm reading failed ▪ 4 -> Nvm writing failed
RadarSt_RollCount	8	2	Motorola	Unsigned	1	0 -> 3 tpcl.: 0->3

Table 4: RadarStatus message description (0x60A)

The following is a detailed description of each of the signals in message 0x60A:

1. ***ACTL_Mode* -** Shows the internal sensor state
   a. ***Normal operation*** – sensor is operational
   b. ***Hardware defect*** – sensor hardware is defective
   c. ***Overtemperature*** – sensor temperature is too high
2. ***Radar_Cfg_Status*** – Represents the status of the sensor configuration. Indicates whether the configuration parameters could be read out or written into the NVM successfully. In case the sensor ID is reconfigured, the signal indicates whether a sensor with the desired sensor ID is already connected to the CAN network.
   a. ***Last learned cfg used*** – the configuration parameter stored in NVM are used (default value after sensor startup)
   b. ***New cfg is used*** – the configuration parameters were successfully changed in the current power cycle
   c. ***Desired ID already used*** – the signal value indicates that the user tried to configure a sensor ID, which is already in use.
   d. ***Nvm reading failed –*** the last learned configuration parameter could not be read out of the NVM.
   e. ***Nvm writing failed –*** the last attempt to write a configuration parameter into the NVM was not successful.

8. Data Output

The data about detected clusters or tracks is transmitted on CAN1. Clusters are newly determined by the processing of the detected radar reflections every cycle. The tracks in turn are evaluated by tracking the clusters over time, so that they include a history over multiple cycles.

8.1. Cluster List

The cluster list output consists of two messages: CAN1_Cluster_Status (0x70B) and CAN1_Cluster_1 (0x70C). The message CAN1_Cluster_Status contains general information about the cluster list itself, for example the number of clusters it currently contains. CAN_Cluster1 contains cluster specific attributes like position and relative velocity. While the message CAN1_Cluster_Status is sent only once during one cluster transmission cycle, CAN_Cluster1 is sent multiple times, once for each cluster in the cluster list. Since the number of CAN_Cluster1 messages can reach a maximum value of 128, an updated cluster list is sent only once in every second radar cycle. Two radar cycles correspond to approx. 66 milliseconds. Otherwise the cycle time is 33 ms.

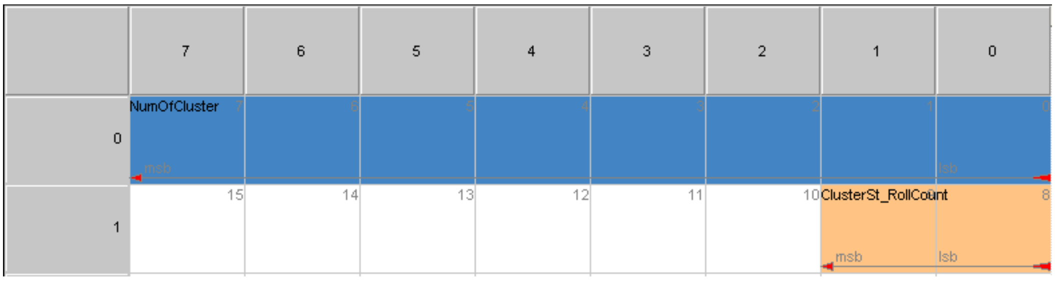


Figure 4: CAN1_Cluster_Status message structure (0x70B) (Bytes 2-7 are unused)

Signal	Start	Len	Byte Order	Value Type	Res	Offset	Value Range
NoOfCluster	0	8	Motorola	unsigned	1	0	0 -> 255 tpcl.: 1->128
ClusterSt_RollCount	8	3	Motorola	unsigned	1	0	0 -> 3 tpcl.: 0->3

Table 5 CAN1_Cluster_Status message description (0x70B)

The following is a detailed description of each of the signals in message 0x70B:

1. ***NumOfCluster*** – Number of clusters in the current cluster list
2. ***ClusterStRollCount*** – The rolling counter is incremented with each valid message

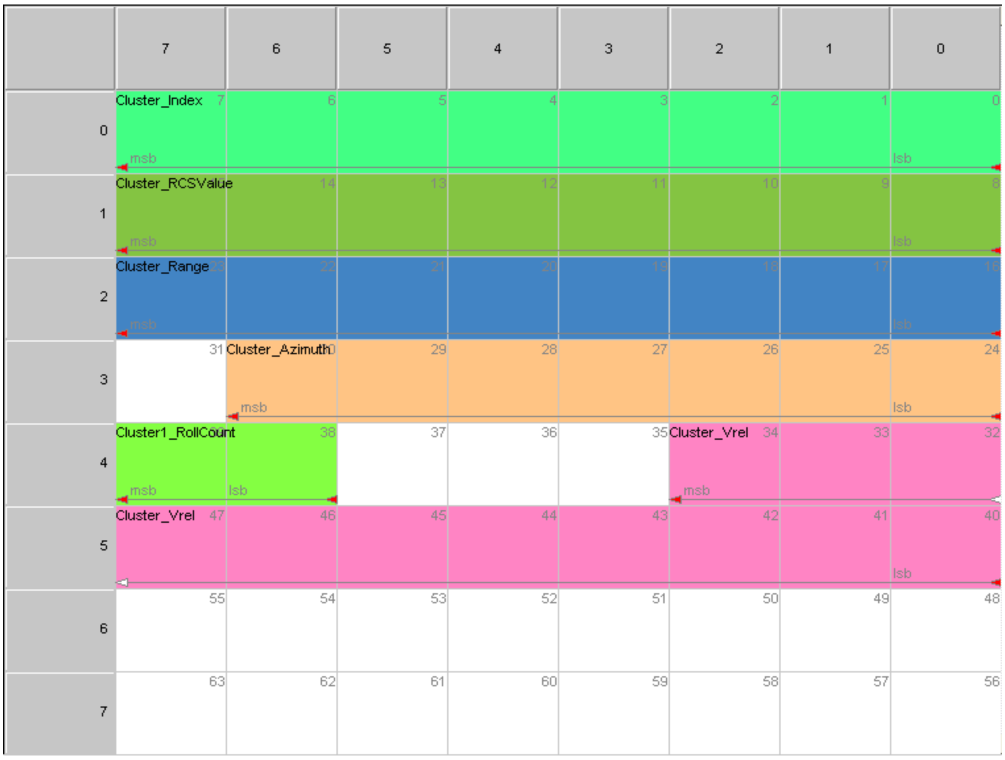


Figure 5: CAN1_Cluster_1 message structure (0x70C)

Signal	Start	Len	Byte Order	Value Type	Res	Offset	Value Range
Cluster_Index	0	8	Motorola	Unsigned	1	0	0 -> 127
Cluster1_RollCount	38	2	Motorola	Unsigned	1		0 -> 3 tpcl.: 0->3
Cluster_RCSValue	8	8	Motorola	Unsigned	0.5 dBm^2	-50	-50 -> 30
Cluster_Range	16	8	Motorola	Unsigned	0.2 m	0	0 -> + 51
Cluster_Azimuth	24	7	Motorola	Unsigned	2 deg	-90	-90 -> + 90
Cluster_Vrel	40	11	Motorola	Unsigned	0.05 m/s	-35	-35 -> +35

Table 6: CAN1_Cluster_1 message description (0x70C)

The following is a detailed description of each of the signals in message 0x70C:

1. ***Cluster_Index*** – Current index of the cluster in the cluster list
2. ***Cluster1_RollCount*** – The rolling counter is incremented with each valid message
3. ***Cluster_RCSValue*** – Radar cross section
4. ***Cluster_Range*** – Radial distance of the cluster
5. ***Cluster_Azimuth*** – Cluster angle
6. ***Cluster_Vrel*** – Relative velocity of the cluster

8.2. Track List

The track list output consists of three messages: CAN1_VersionID (0x700), CAN1_Track_Status (0x60B), CAN1_Track_1 (0x60C) and CAN1_Track_2 (0x60D). The message CAN1_Track_Status contains general information about the track list itself, for example the number of tracks it currently contains. The messages CAN_Track1 and CAN1_Track2 contain track specific attributes like position and relative velocity. While the message CAN1_Track_Status is sent only once during one transmission cycle, CAN1_Track1 and CAN1_Track2 are sent multiple times as message pairs for each track in the track list. The maximum number of tracks in the track list cannot exceed 25, therefore (in contrast to the cluster lists) it is possible to send an updated track list each radar cycle. One radar cycle corresponds to approx. 33 milliseconds.

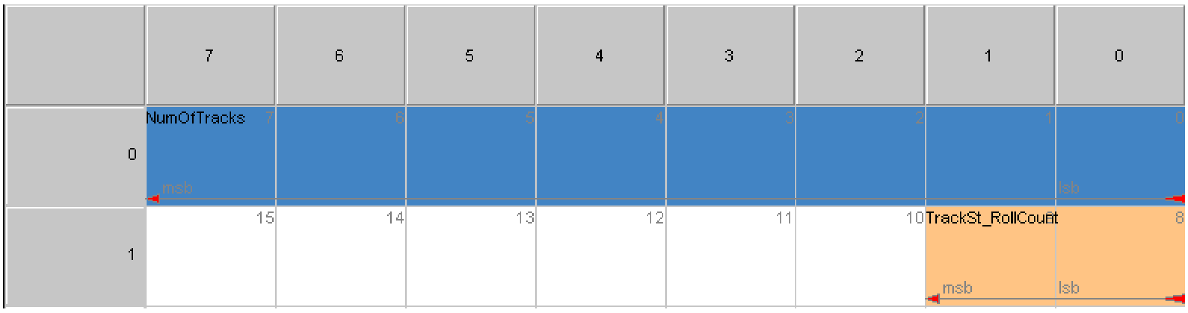


Figure 6: CAN1_Track_Status message structure (0x60B) (Bytes 2 to 7 are not used)

Signal	Start	Len	Byte Order	Value Type	Res	Offset	Value Range
NumOfTracks	0	8	Motorola	Unsigned	1	0	0 -> 255 tpcl.: 0 -> 25
TrackSt_RollCount	8	2	Motorola	Unsigned	1	0	0 -> 65535 Recount starts after 65535

Table 7: CAN1_Track_Status message description (0x60B)

The following is a detailed description of each of the signals in message 0x60B:

1. ***NumOfTracks* –** Number of measured tracks
2. ***TrackStRollCount* –** The rolling counter is incremented with each valid message

Figure 7: CAN1_Track_1 message structure (0x60C)

Signal	Start	Len	Byte Order	Value Type	Res	Offset	Value Range
Track_ID	8	16	Motorola	Unsigned	1	0	0 -> 65535
Track_LongDispl	29	9	Motorola	Unsigned	0.1 m	0	0 -> 51.1
Track_LatDispl	46	10	Motorola	Unsigned	0.1 m	-51.1	-51.1 -> 51.2
Track_Index	24	5	Motorola	Unsigned	1	0	0 -> + 24
Track_VrelLong	50	12	Motorola	Unsigned	0.02 m/s	-35	-35 -> + 35
Track_VrelLat	56	8	Motorola	Unsigned	0.25 m/s	-32	-32 -> 31.75
Track1_RollCount	48	2	Motorola	Unsigned	1	0	0 -> 3 tpcl.: 0->3

Table 8: CAN1_Track_1 message description (0x60C)

The following is a detailed description of each of the signals in message 0x60B:

1. ***Track_ID*** – Unique track ID
2. ***Track_LongDispl*** – Longitudinal displacement
3. ***Track_LatDispl*** – Lateral displacement
4. ***Track_Index*** – Current index of the track in the track list
5. ***Track_VrelLong*** – Relative longitudinal velocity
   (Obstacles which oncoming have got a negative prefix and leaving obstacles are positive.)
6. ***Track_VrelLat*** – Relative lateral velocity
   (Obstacles which move from left to right have got a negative prefix and from right to left, they are positive.)
7. ***Track1_RollCount*** – The rolling counter is incremented with each valid message

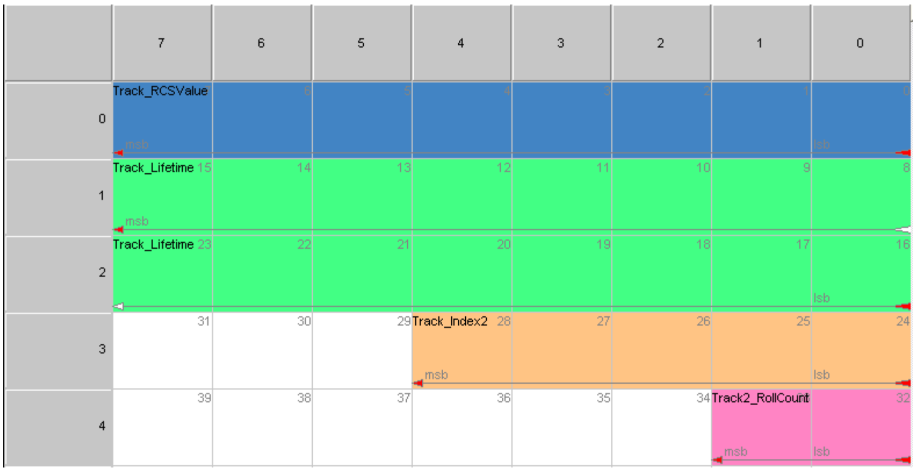


Figure 1: CAN1_Track_2 message structure (0x60D) (Bytes 5-7 are unused)

Signal	Start	Len	Byte Order	Value Type	Res	Offset	Value Range
Track_RCSValue	0	8	Motorola	Unsigned	0.5 dBm^2	-50	- 50 -> + 30
Track_Lifetime	16	16	Motorola	Unsigned	0.1 s	0	0 -> + 6553.5
Track_Index2	24	5	Motorola	Unsigned	1	0	0 -> 24
Track2_RollCount	32	2	Motorola	Unsigned	1	0	0 -> 3 tpcl.: 0->3

Table 9: CAN1_Track_2 message description (0x60D)

The following is a detailed description of each of the signals in message 0x60D:

1. ***Track_RCSValue*** – Radar cross section (RCS) is the measure of the reflective strength of a cluster or in other words it is the measure of power scattered in a given spatial direction when a cluster is illuminated by an incident wave.
2. ***Track_Lifetime*** – The current lifetime of the track. In case the lifetime exceeds 6553.5, the value remains at this maximum value. No overflow takes place.
3. ***Track_Index2*** – Current track list index of this track
4. ***Track2_RollCount*** – The rolling counter is incremented with each valid message

9. Collision Detection

For the two sensor variants (SRR 208-21 SS/HS / TC) (TC = Tracks and/or Collision Avoidance) a collision detection functionality is available. The additional CAN messages are described in the following sections.

9.1. Collision Detection Configuration Input

This message is used for configuration of the collision detection warning function.

Figure 9: CollDetConfig message structure (0x400)

Signal	Start	Len	Byte Order	Value Type	Res	Offset	Value Range
ResetAllWarnings	38	1	Motorola	Unsigned	1	0	0 -> Idle 1 -> Reset warning
CollisionDetectionActivation	11	1	Motorola	Unsigned	1	0	0 -> deactivated 1 -> activated
ObjDetectionTimeSetFlag	9	1	Motorola	Unsigned	1	0	0 -> Idle 1 -> Change time

CfgObjMinDetectionTime	0	8	Motorola	Unsigned	0.1	0	0 -> 25.5 s
ClearAllCollDetSettings	12	1	Motorola	Unsigned	1	0	0 -> Idle 1 -> Clear all regions
RegionActivation	8	1	Motorola	Unsigned	1	0	0 -> deactived 1 -> activated
CfgRegionID	13	1	Motorola	Unsigned	1	0	0 -> 7
CoordinatesSetFlag	10	1	Motorola	Unsigned	1	0	0 -> Idle 1 -> Change coordinates
CfgRegionPoint1X	16	8	Motorola	Unsigned	0.2 m	0	0 -> 51
CfgRegionPoint1Y	39	9	Motorola	Unsigned	0.2 m	-51	-51 -> +51.2
CfgRegionPoint2X	40	8	Motorola	Unsigned	0.2 m	0	0 -> 51
CfgRegionPoint2Y	63	9	Motorola	Unsigned	0.2 m	-51	-51 -> 51.2

Table 10: CollDetConfig message description (0x400)

The following is a detailed description of each of the signals in message 0x400:

1. ***ResetAllWarnings* –** Reset currently active warnings[2]
2. ***CollisionDetectionActivation*** – De-/activate collision detection function[2]
3. ***ObjDetectionTimeSetFlag*** – Allow change of time parameter[2]
4. ***CfgObjMinDetectionTime*** – Minimum time an object needs to be detected before a warning is triggered[2]
5. ***ClearAllCollDetSettings* –** Clear all collision detection settings[2]
6. ***RegionActivation*** – De-/activate current region[3]
7. ***CfgRegionID* –** ID of current region to configure[3]
8. ***CoordinatesSetFlag*** – Allow change of current regions coordinates[3]
9. ***CfgRegionPoint1X*** – 1st X coordinate of region[3]
10. ***CfgRegionPoint1Y*** – 1st Y coordinate of region[3]

2 This is a global, region independent setting, which is valid for all warning regions.
3 This is a local resp. region dependent setting, which is valid only for the warning region, specified by the signal CfgRegionID.

11. ***CfgRegionPoint2X*** – 2nd X coordinate of region[3]
12. ***CfgRegionPoint2Y*** – 2nd Y coordinate of region[3]

9.2. State Output

This message shows region specific information for the respective region, specified by the signal RegionIDState. If the collision detection functionality is active, this message is sent 8 times (once for each region). The message is sent even if the corresponding region is deactivated.

Figure 10: CollDetState message structure (0x408)

Signal	Start	Len	Byte Order	Value Type	Res	Offset	Value Range
RegionStatus	2	2	Motorola	Unsigned	1	0	0 -> No warning 1 -> Track warning 2 -> Reserved 3 -> Warning deactivated
CollSt_RollCount	0	2	Motorola	Unsigned	1	0	0 -> 3 tpcl.: 0->3
NumOfWarn Tracks	8	5	Motorola	Unsigned	1	0	0 ... 25
ObjMinDetection Time	16	8	Motorola	Unsigned	0.1 s	0	0 ... 25.5
RegionIDState	42	3	Motorola	Unsigned	1	0	0 ... 7
NumberOf Regions	45	3	Motorola	Unsigned	1	0	0 ... 7
RegionPoint1X	24	8	Motorola	Unsigned	0.2 m	0	0 ... 51
RegionPoint1Y	49	9	Motorola	Unsigned	0.2 m	-51	- 51 ... 51.2
RegionPoint2X	32	8	Motorola	Unsigned	0.2 m	0	0 ... 51
RegionPoint2Y	56	9	Motorola	Unsigned	0.2 m	-51	- 51 ... 51.2

Table 11: CollDetState message description (0x408)

The following is a detailed description of each of the signals in message 0x408:

1. ***RegionStatus* –** The status of the respective region. See Table 11 for details.
2. ***CollState_RollCount*** – The rolling counter is incremented with each valid message
3. ***NumOfWarnTracks*** – Number of tracks in an active warning region
4. ***ObjMinDetectionTime* –** Value of configured minimum object detection time
5. ***RegionIDState*** – Region ID
6. ***NumberOfRegions*** – Number of active regions
7. ***RegionPoint1X*** – 1st X coordinate of region
8. ***RegionPoint1Y*** – 1st Y coordinate of region

9. ***RegionPoint2X*** – 2nd X coordinate of region
10. ***RegionPoint2Y*** – 2nd Y coordinate of region

9.3. Warning Output

This type of message shows information about the tracks in an active warning region. It is only sent if a warning for an active region was triggered. If none of the regions are in warning state no CollDetWarn message is sent.

The collision detection function is implemented and activated in the sensortype SRR 208-21 SS/HS / TC (TC = Tracks and/or Collision Avoidance). It cannot be switched on or off. It is not possible to transmit clusters on the CAN 1, because each track in each region which is relevant in the warning mode is transmitted to the CAN 1. This could be all detected tracks if the region is determined by the max. geometrical dimensions by the parameters “Region Point X” and Region “Point Y”

(that is why: (SRR 208-21 SS/HS / **TC**)).

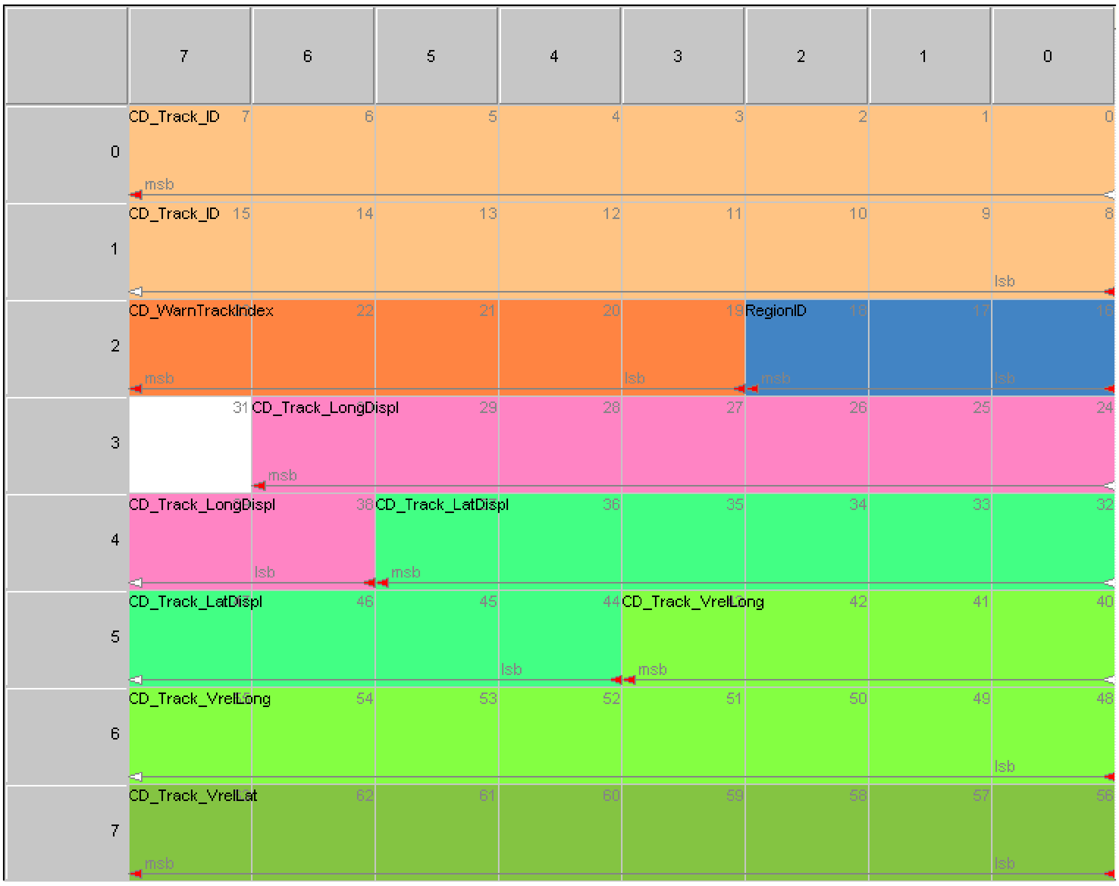


Figure 11: CollDetWarn message structure (0x409)

Signal	Start	Len	Byte Order	Value Type	Res	Offset	Value Range
CD_TrackID	8	16	Motorola	Unsigned	1	0	0 ... 65535
CD_WarnTrackIndex	19	5	Motorola	Unsigned	1	0	0 ... 24
RegionID	16	3	Motorola	Unsigned	1	0	0 ... 7
CD_Track_LongDispl	38	9	Motorola	Unsigned	0.1 m	0	0 ... +51.1
CD_Track_LatDispl	44	10	Motorola	Unsigned	0.1 m	-51.1	-51.1 ... +51.2
CD_Track_VrelLong	48	12	Motorola	Unsigned	0.02 m/s	-35	-35 ... +35
CD_Track_VrelLat	56	8	Motorola	Unsigned	0.25 m/s	-32	-32 ... +31.75

Table 12: CollDetWarn message description (0x409)

The following is a detailed description of each of the signals in message 0x409:

1. ***CD_TrackID* –** Unique track ID
2. ***CD_WarnTrackIndex*** – Index of the warning
3. ***RegionID*** – Region ID, to which the message belongs.
4. ***CD_TrackLongDispl*** – Longitudinal displacement of the warning track
5. ***CD_TrackLatDispl*** – Lateral displacement of the warning track
6. ***CD_Track_VrelLong*** – Longitudinal velocity of the warning track
7. ***CD_Track_VrelLat*** – Lateral velocity of the warning track