

---

# Continental Engineering Services GmbH



***Standardized ARS Interface***

***Technical Documentation***

***ARS 308-2***

V1.12

## Table of Contents

<b>1. Introduction</b> .....	3
<b>2. Operating Conditions</b> .....	3
<b>3. Safety Information</b> .....	3
<b>4. Interface</b> .....	3
<b>5. Description</b> .....	4
<b>6. Sensor Configuration</b> .....	6
<b>7. State Output</b> .....	9
<b>8. Input Signals</b> .....	12
8.1. Speed Information.....	12
8.2. Yaw Rate Information .....	13
<b>9. Data Output</b> .....	14
9.1. Target List.....	14
9.2. Object List.....	18

## History

Version	Date	Chapter	Change description
V1.4	31 Jul 2009	9	Added data output description
V1.5	31 Jul 2009	9	Minor changes
V1.6	31 Jul 2009		<ul style="list-style-type: none"> <li>▪ Regenerated Table Of Contents</li> <li>▪ Renamed CAN1 to CAN2 in order to match message names</li> <li>▪ Renamed CAN0 to CAN1</li> <li>▪ Updated messages 0x200 and 0x201 descriptions</li> </ul>
V1.7	31 Jul 2009		Minor changes
V1.8	31 Jul 2009	9.1	Added values for angle status
V1.9	8 Dec 2010		Update for single CAN hardware
V1.10	13 Dec 2010		Version update
V1.11	06 Sep 2011	6	<ul style="list-style-type: none"> <li>• radar_elevation signal description</li> <li>• SensTempErr signal description</li> <li>• 0x600 and 0x701 signal descriptions</li> <li>• 0x60A, 0x60B and 0x700 signal descriptions</li> </ul>
	08 Sep 2011	7	
	27 Sep 2011	9.1	
	04 Oct 2011	9.2	
V1.12	11 Nov 2011	all	<ul style="list-style-type: none"> <li>• Layout</li> </ul>

## 1. Introduction

The ARS 300 is a Radar 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 deceleration requests to the vehicle network.

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.

## 2. Operating Conditions

Please refer to *ARS 300-Technical Description*.

## 3. Safety Information

Please refer to *ARS 300-Technical Description*.

## 4. Interface

The ARS 300 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 by corresponding DBC files.

As an example, a possible CAN bus network is shown in Figure 1. Since no termination resistors are included in the radar sensor ARS 300, 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 ARS 300 is used for configuration, sensor state output, data input and data output.

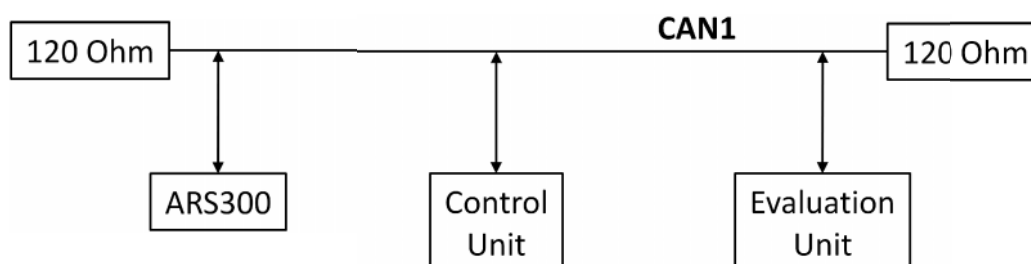


Figure 1: CAN network

## 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 targets and objects. One target consists of multiple reflections which have a similar position and movement and therefore can be combined. The information about a target like size, relative velocity and position is then transmitted on CAN1. The position is calculated in an angular coordinate system, i.e. distance and angle relative to the sensor. The targets are newly evaluated every cycle. In contrast to this, objects have a history and consist of tracked targets. The position of the object is calculated relative to an assumed vehicle course. The course is determined by using the speed and yaw rate information. If information is missing, it will be set to default values: yaw rate = 0 deg/s, speed = 0 m/s and standstill.

The elevation of the radar radiation can be changed in a certain range via configuration of the radar sensor. The angles used for the position of targets and for the elevation of the radar beam are shown in Figure 2.

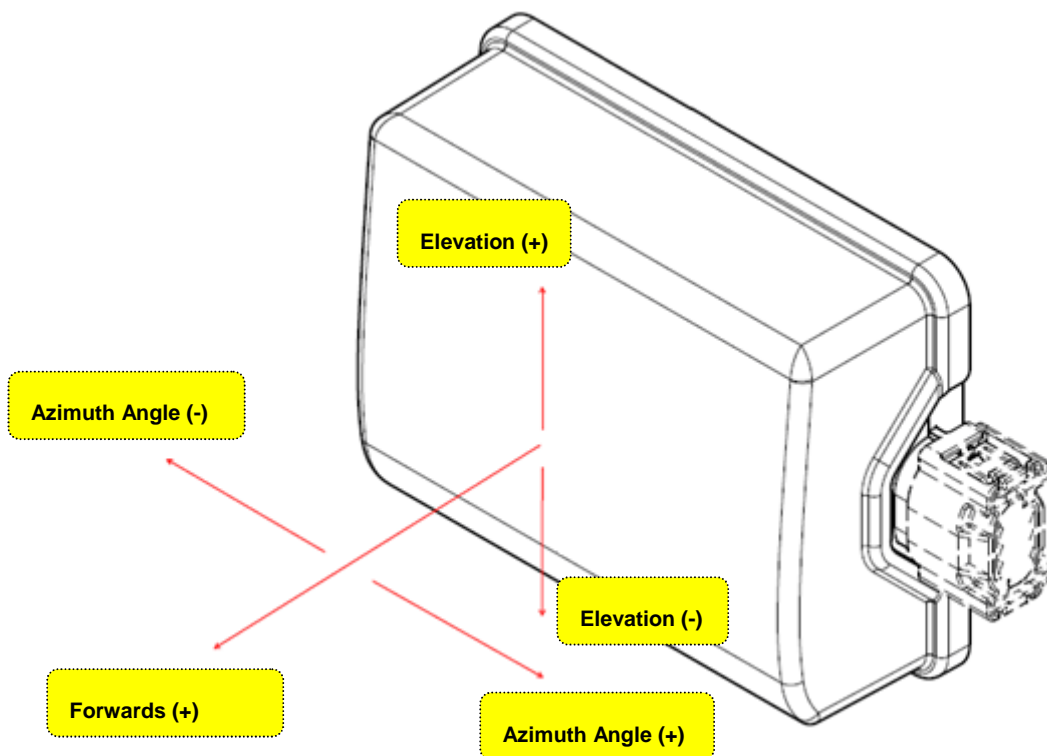


Figure 2: Sign Conventions - Targets Azimuth Angle and Beam elevation angle

CAN	Frame Format	Message ID	Message Name	Content	Section
1	CAN 2.0A (11 Bit)	0x200	RadarConfiguration	Sensor configuration	6
1	CAN 2.0A (11 Bit)	0x201	RadarState	State output	7
1	CAN 2.0A (11 Bit)	0x300	SpeedInformation	Vehicle speed	8.1
1	CAN 2.0A (11 Bit)	0x301	YawRateInformation	Vehicle yaw rate	8.2
1	CAN 2.0A (11 Bit)	0x600	CAN1_Target_Status	Target status	9.1
1	CAN 2.0A (11 Bit)	0x60A	CAN1_Obj_Status	Object status	9.2
1	CAN 2.0A (11 Bit)	0x60B	CAN1_Obj_1	Object information 1	9.2
1	CAN 2.0A (11 Bit)	0x60C	CAN1_Obj_2	Object information 2	9.2
1	CAN 2.0A (11 Bit)	0x700	CAN1_VersionID	Object List Interface Version	9.2
1	CAN 2.0A (11 Bit)	0x701	CAN1_Target_1	Target information 1	9.1
1	CAN 2.0A (11 Bit)	0x702	CAN1_Target_2	Target information 2	9.1

Table 1: Sensor CAN messages

## 6. Sensor Configuration

The ARS 300 sensor can be configured via message 0x200 on CAN1 (cf. Figure 3 and Table 2). This message should only be transmitted to the ARS 300 when a parameter change is desired. It is not necessary to transmit this message cyclically.

It should also be noted that in order to guarantee that a parameter change is accepted, the signal **NVMreadStatus** in message 0x201 (RadarState) must have a value of **successful** as defined in the DBC file.

The following parameters can be changed

1. Radar range length – **Maximum distance** at which objects will be detected !
  - a. **Radar range length: = 50 m** ->  
Near Range = 25 m, Far Range = 50 m
  - b. **Radar range length: = 60 m** ->  
Near Range = 30 m, Far Range = 60 m
  - c. **Radar range length: = 70 m** ->  
Near Range = 35 m, Far Range = 70 m
  - d. **Radar range length: = 80 m** ->  
Near Range = 40 m, Far Range = 80 m
  - e. **Radar range length: = 90 m** ->  
Near Range = 45 m, Far Range = 90 m
  - f. **Radar range length: = 100 m** ->  
Near Range = 50 m, Far Range = 100 m
  - g. **Radar range length: = 100 m -> 200 m** ->  
Near Range = 50 m constant ! Far Range = 100 m -> 200 m
2. Elevation of the sensor's plate – The elevation angle of the radar beam. Please note that Elevation angle of 0 deg corresponds to the sensor plate inclined towards the road and 32 deg inclined towards the sky.
3. Power reduction mode
  - a. **True** – The ARS300 will shift to low power output mode at standstill.
  - b. **False** – The ARS300 will not shift to low output mode.
4. The type of the output list
  - a. **SendObjects** – The object list will be transmitted on CAN1
  - b. **SendTargets** – The target list will be transmitted on CAN1

These parameters can be changed individually or in combinations. 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 ARS, 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.

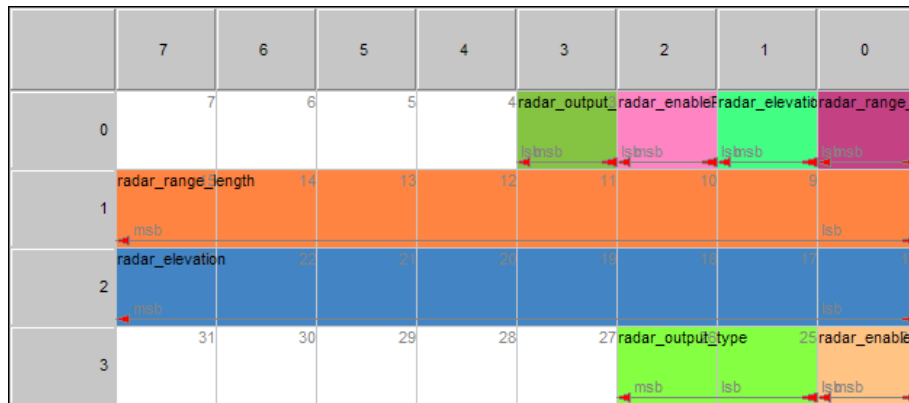


Figure 3: Radar Configuration message structure (0x200) (Bytes 4-7 and Bits in White are unused)

Signal	Start	Len	Byte Order	Value Type	Res	Value Range
radar_range_lengthV	0	1	Motorola	Unsigned	1	<ul style="list-style-type: none"> <li>▪ 0 -&gt; Invalid</li> <li>▪ 0 x 0</li> <li>▪ 1 -&gt; Valid</li> <li>▪ 0 x 1</li> </ul>
radar_range_length	15	8	Motorola	Unsigned	1 m	50 m, 51 m, 52 m ..... 200 m 0 x 32 (= 50) 0 x C8 (= 200)
radar_output_typeV	3	1	Motorola	Unsigned	1	<ul style="list-style-type: none"> <li>▪ 0 -&gt; Invalid</li> <li>▪ 1 -&gt; Valid</li> </ul>
radar_output_type	26	2	Motorola	Unsigned	1	<ul style="list-style-type: none"> <li>▪ 0 -&gt; (0 x 0) inapplicable</li> <li>▪ 1 -&gt; (0 x 1) SendObjects</li> <li>▪ 2 -&gt; (0 x 2) SendTargets</li> <li>▪ 3 -&gt; (0 x 3) inapplicable</li> </ul>
radar_enablePowerReductionV	2	1	Motorola	Unsigned	1	<ul style="list-style-type: none"> <li>▪ 0 -&gt; Invalid</li> <li>▪ 1 -&gt; Valid</li> </ul>
radar_elevationV	1	1	Motorola	Unsigned	1	<ul style="list-style-type: none"> <li>▪ 0 -&gt; Invalid</li> <li>▪ 1 -&gt; Valid</li> </ul>
radar_enablePowerReduction	24	1	Motorola	Unsigned	1	<ul style="list-style-type: none"> <li>▪ 0 -&gt; False</li> <li>▪ 1 -&gt; True</li> </ul>
radar_elevation	23	8	Motorola	Unsigned	0.25 deg	0 <sup>0</sup> , 0.25 <sup>0</sup> , 0.50 <sup>0</sup> ..... 32 <sup>0</sup> (Road -> Sky)

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

Signal	Start	Len	Byte Order	Value Type	Res	Default Factory Settings
radar_range_lengthV	0	1	Motorola	Unsigned	1	▪ 0 -> Invalid
radar_range_length	15	8	Motorola	Unsigned	1 m	200 m
radar_output_typeV	3	1	Motorola	Unsigned	1	▪ 0 -> Invalid
radar_output_type	26	2	Motorola	Unsigned	1	▪ 2 -> Send Targets
radar_enablePowerReductionV	2	1	Motorola	Unsigned	1	▪ 0 -> Invalid
radar_elevationV	1	1	Motorola	Unsigned	1	▪ 0 -> Invalid
radar_enablePowerReduction	24	1	Motorola	Unsigned	1	▪ 0 -> False
radar_elevation	23	8	Motorola	Unsigned	0.25 deg	tpcl.: $16^0$ ( $\pm 1^0$ ) 0 x 40 (= 16) possible > $16^0$ after calibration possible < $16^0$ after calibration

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

## 7. State Output

Message 0x201 contains signals which report the state of the sensor.

After configuring a parameter, (except for the object/target selection) by sending message 0x200 to the ARS300, the corresponding signal in message 0x201 can be checked in order to verify that the configuration change was accepted. The following list shows the correspondence between the configuration parameters and the signals in message 0x201 which can be used to verify that the configuration change has been accepted.

1. Radar range length -> ***currRangeLengthCal***
2. Elevation of the sensor's plate -> ***currElevationCal***
3. Power reduction mode -> ***RadarPowerReduction***

Figure 4 shows the layout of the 0x201 message, while Table 4 shows the details of each signal.

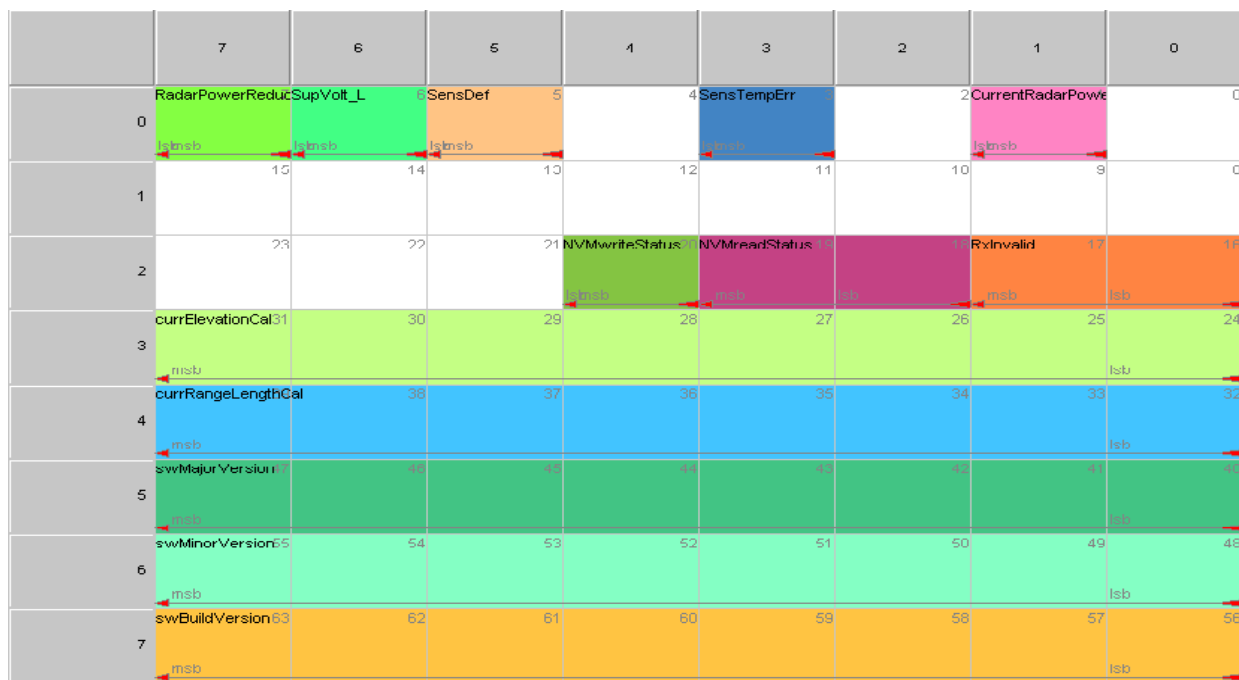


Figure 4: Radar State message Structure (0x201) (Bits in White are unused)

Signal	Start	Len	Byte Order	Value Type	Res	Value Range
NVMwriteStatus	20	1	Motorola	Unsigned	1	<ul style="list-style-type: none"> <li>▪ 0 -&gt; failed</li> <li>▪ 1 -&gt; successful</li> </ul>
NVMreadStatus	19	2	Motorola	Unsigned	1	<ul style="list-style-type: none"> <li>▪ 0 -&gt; pending</li> <li>▪ 1 -&gt; failed</li> <li>▪ 2 -&gt; successful</li> <li>▪ 3 -&gt; irrelevant/unused</li> </ul>
currRangeLengthCal	39	8	Motorola	Unsigned	1 m	50m, 51m, 52m ... 199m, 200m
currElevationCal	31	8	Motorola	Unsigned	0.25 deg	0 <sup>0</sup> , 0.25 <sup>0</sup> , 0.50 <sup>0</sup> ..... 31.75 <sup>0</sup> , 32 <sup>0</sup>
RadarPowerReduction	7	1	Motorola	Unsigned	1	<ul style="list-style-type: none"> <li>▪ 0 -&gt; disabled</li> <li>▪ 1 -&gt; enabled</li> </ul>
CurrentRadarPower	1	1	Motorola	Unsigned	1	<ul style="list-style-type: none"> <li>▪ 0 -&gt; radarLowPower</li> <li>▪ 1 -&gt; radarHighPower</li> </ul>
SupVolt_L	6	1	Motorola	Unsigned	1	<ul style="list-style-type: none"> <li>▪ 0 -&gt; False</li> <li>▪ 1 -&gt; True</li> </ul>
RxInvalid	17	2	Motorola	Unsigned	1	<ul style="list-style-type: none"> <li>▪ 0 -&gt; noneMissing</li> <li>▪ 1 -&gt; speedMissing</li> <li>▪ 2 -&gt; yawRateMissing</li> <li>▪ 3 -&gt; speedAndYawRate Missing</li> </ul>
SensDef	5	1	Motorola	Unsigned	1	<ul style="list-style-type: none"> <li>▪ 0 -&gt; False</li> <li>▪ 1 -&gt; True</li> </ul>
SensTempErr	3	1	Motorola	Unsigned	1	<ul style="list-style-type: none"> <li>▪ 0 -&gt; False</li> <li>▪ 1 -&gt; True</li> </ul>
# swMajorVersion	47	8	Motorola	Unsigned	1	0 -> 255
# swMinorVersion	55	8	Motorola	Unsigned	1	0 -> 255
# swBuildVersion	63	8	Motorola	Unsigned	1	0 -> 255

Table 4: Radar State message description (0x201)

# → *Valid only for the Company's internal use*

The following is a detailed description of each of the signals in message 0x201:

1. **NVMwriteStatus** - Shows the state of storing a configuration parameter to non-volatile memory. It can be equal to the following values
  - a. **failed** – a problem occurred during the last attempt to write a parameter.
  - b. **successful** – the last configuration parameter received was successfully written to non-volatile memory
2. **NVMreadStatus** - Shows the state of reading the configuration parameters from non-volatile memory at startup. It can be equal to the following values
  - a. **pending** – configuration not read yet.
  - b. **successful** – the configuration parameters were successfully read from non-volatile memory
3. **currRangeLengthCal** – The current value of the *Radar range length* configuration parameter
4. **currElevationCal** – The current value of the elevation angle of the reflector plate.
5. **RadarPowerReduction** - The current value of the *Power reduction mode* configuration parameter
6. **CurrentRadarPower** – The actual current radar power level (**radarLowPower** or **radarHighPower**)
7. **SupVolt\_L** – True if the supply voltage of the ARS300 is too low.
8. **RxInvalid** – Shows the state of the speed and yaw rate CAN input signals. Can be one of the following states:
  - a. **noneMissing** – Both the speed and yaw rate are being received.
  - b. **speedMissing** – Only the yaw rate is being received
  - c. **yawRateMissing** – Only the speed is being received
  - d. **speedAndYawRateMissing** – Both the speed and yaw rate are not being received.
9. **SensDef** – Sensor defective if 1
10. **SensTempErr** – 1 indicates that the Sensor is switched off because the operating temperature is too high.
11. **#swMajorVersion** – Major version of the radar firmware
12. **#swMinorVersion** – Minor version of the radar firmware
13. **#swBuildVersion** – Build version of the radar firmware

## 8. Input Signals

Input messages 0x300 and 0x301 are accepted by the ARS300. The radar sensor will still function without these two messages. After a 500msec 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 **RxInvalid** in message 0x201.

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

### 8.1. Speed Information

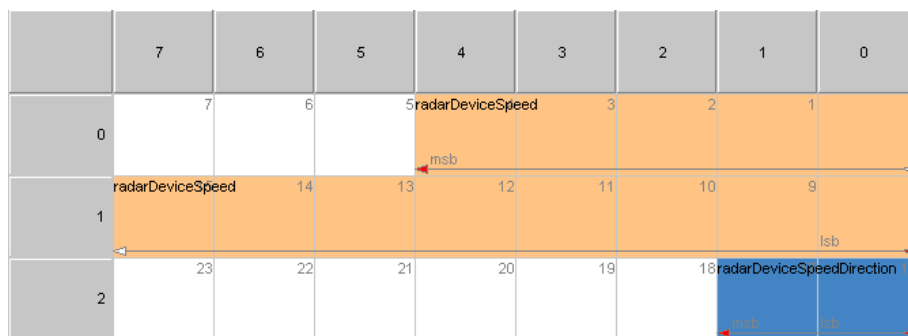


Figure 5: Speed Information message structure (0x300) (Bytes 3-7 and Bits in White are unused)

Signal	Start	Len	Byte Order	Value Type	Res	Value Range
radarDeviceSpeed	4	13	Motorola	Unsigned	0.02m/s	0 m/s -> 163.8 m/s
radarDeviceSpeedDirection	17	2	Motorola	Unsigned	1	<ul style="list-style-type: none"> <li>▪ 0 -&gt; Standstill</li> <li>▪ 1 -&gt; Forward</li> <li>▪ 2 -&gt; Reverse</li> <li>▪ 3 -&gt; irrelevant / unused</li> </ul>

Table 5: Speed Information message description (0x300)

1. **Radar Device Speed** - Shows the absolute magnitude of speed in the direction the Radar is moved while looking into positive straight ahead direction (see Figure 2)
  - a. **Input unit: m/s**
  - b. **Range: 0 m/s – 163.8 m/s = 0 kph – 589.68 kph**

2. **Radar Device Speed Direction** - Indicates the direction of the Radar while looking into positive straight ahead direction (see Figure 2)
  - a. **Standstill** – no movement in any direction
  - b. **Forward** – Radar is moved into positive straight ahead direction
  - c. **Reverse** – Radar is moved against positive forward direction = backwards
  - d. **No use** – no matter if high or low

## 8.2. Yaw Rate Information

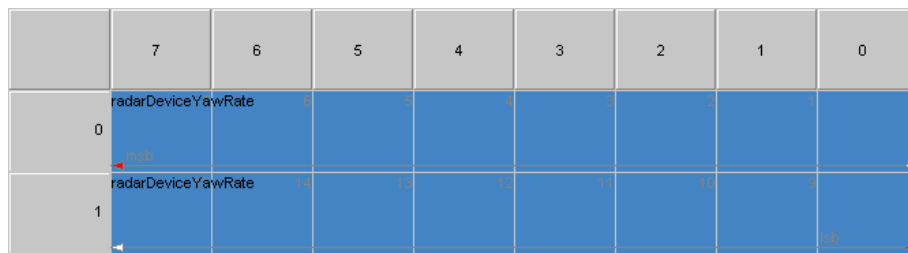


Figure 6: Yaw Rate Information message structure (0x301) (Bytes 2-7 are unused)

Signal	Start	Len	Byte Order	Value Type	Res	Offset	Value Range
radarDeviceYaw Rate	7	16	Motorola	unsigned	0.01 deg/s (-) to Right (+) to Left	-327.68 deg/s	-327.68 deg/s ..... ...+327.66 deg/s

Table 6: Yaw Rate Information message description (0x301)

1. **Radar Device Yaw Rate** – Rate of change of angular velocity
  - a. **Input unit: deg/s**
  - b. **Range: - 327.66 deg/s to +327.68 deg/s**

## 9. Data Output

The data about detected targets or objects are transmitted on CAN1. Targets are newly determined by the processing of the detected radar reflections every cycle. However, the objects are evaluated form by tracking the targets, so they inhibit a history about multiple cycles.

### 9.1. Target List

The target list output consists of three messages: CAN1\_Target\_Status (0x600), CAN1\_Target\_1 (0x701) and CAN1\_Target\_2 (0x702). For each cycle the status message and 96 message pairs (CAN1\_Target\_1, CAN1\_Target\_2) for identified targets are sent. The first 32 targets are from near range, the following 64 targets are from far range.

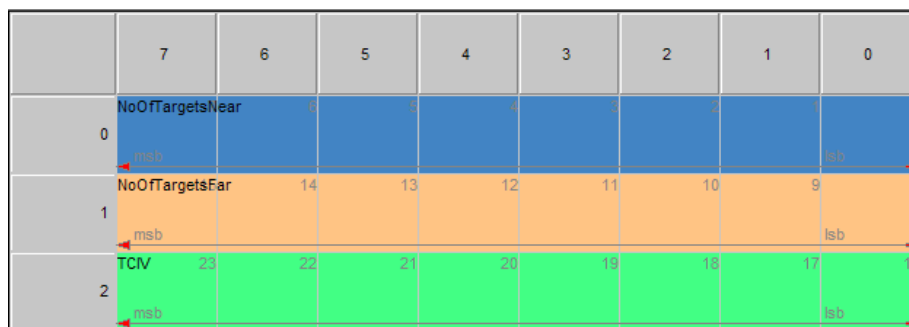


Figure 7: CAN1\_Target\_Status message structure (0x600) (Bytes 3-7 are unused)

Signal	Start	Len	Byte Order	Value Type	Res	Offset	Value Range
NoOfTargetsNear	0	8	Motorola	unsigned	1	0	0 -> 255 tpcl.: 1->32
NoOfTargetsFar	8	8	Motorola	unsigned	1	0	0 -> 255 tpcl.: 1->64
# TCIV	16	8	Motorola	unsigned	0.1	0	0 -> 25.5

Table 7 CAN1\_Target\_Status message description (0x600)

# → *Valid only for the Company's internal use*

The following is a detailed description of each of the signals in message 0x600:

1. **NoOfTargetsNear** – Number of detected targets in the near range
2. **NoOfTargetsFar** – Number of detected targets in the far range
3. **#TCIV** – Target list Can Interface Version

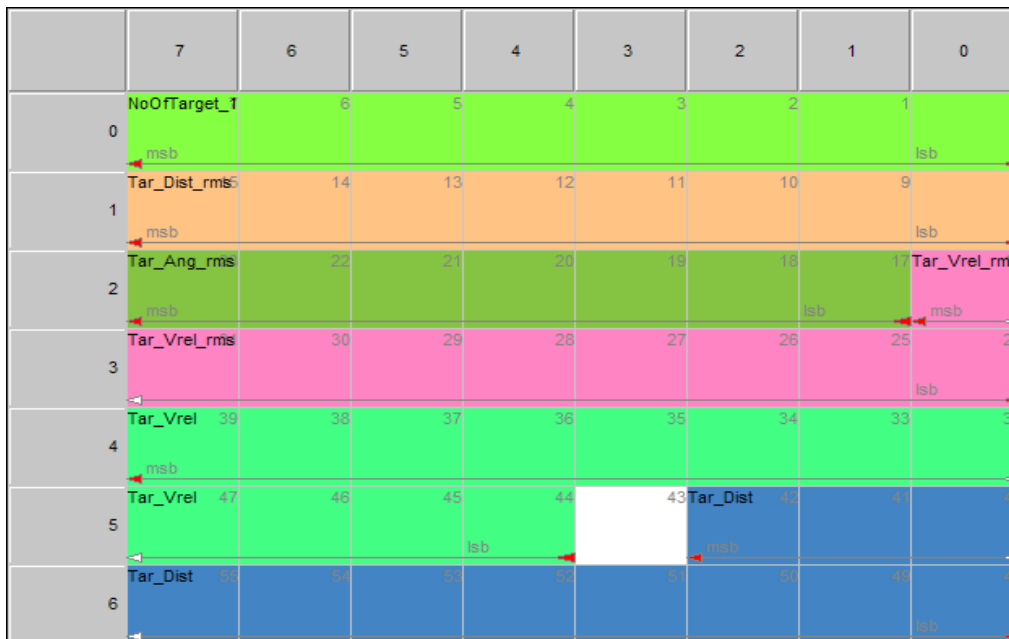


Figure 8: CAN1\_Target\_1 message structure (0x701) (Byte 7 and the Bit in White are unused)

Signal	Start	Len	Byte Order	Value Type	Res	Offset	Value Range
NoOfTarget_1	0	8	Motorola	Unsigned	1	0	0 -> 255 tpcl.: 1 -> 96
Tar_Dist_rms	8	8	Motorola	Unsigned	0.1 m	-10 m	-10 m -> +15.5 m
Tar_Ang_rms	17	7	Motorola	Unsigned	0.1 deg	0 <sup>0</sup>	0 <sup>0</sup> -> 12.7 <sup>0</sup>
Tar_Vrel_rms	24	9	Motorola	Unsigned	0.03 m/s	-5 m/s	- 5 m/s -> + 10.33 m/s
Tar_Vrel	44	12	Motorola	Unsigned	0.03 m/s	-25 m/s	-25 m/s -> + 97.85 m/s
Tar_Dist	48	11	Motorola	Unsigned	0.1 m	0 m	0 m -> 204.7 m

Table 8: CAN1\_Target\_1 message description (0x701)

The following is a detailed description of each of the signals in message 0x701:

1. **NoOfTarget\_1** – Target number
2. **Tar\_Dist\_rms** – Target range standard deviation
3. **Tar\_Ang\_rms** – Target angle standard deviation
4. **Tar\_Vrel\_rms** – Target relative velocity standard deviation
5. **Tar\_Vrel** – Target relative velocity
6. **Tar\_Dist** – Target range

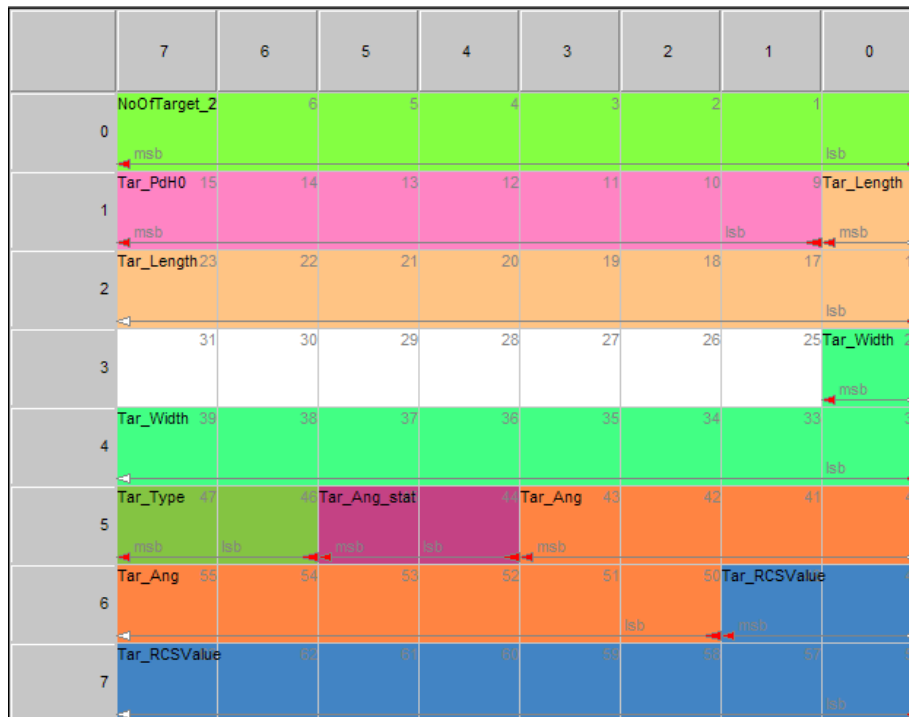


Figure 9: CAN1\_Target\_2 message structure (0x702) (Bits in White are unused)

Signal	Start	Len	Byte Order	Value Type	Res	Offset	Value Range
NoOfTarget_2	0	8	Motorola	Unsigned	1	0	0 -> 255 tpcl.: 1 -> 96
Tar_PdH0	9	7	Motorola	Unsigned	1 %	0 %	0 % ->127 %
Tar_Length	16	9	Motorola	Unsigned	0.1 m	0 m	0 m ->51.1 m
Tar_Width	32	9	Motorola	Unsigned	0.1 m	0 m	0 m ->51.1 m
Tar_Ang_stat	44	2	Motorola	Unsigned	1	0	0 -> 3
Tar_Type	46	2	Motorola	Unsigned	1	0	0 -> 3
Tar_Ang	50	10	Motorola	Unsigned	0.1 deg	-30 <sup>0</sup>	-30 <sup>0</sup> -> 72.3 <sup>0</sup>
Tar_RCSValue	56	10	Motorola	Unsigned	0.1 dBm <sup>2</sup>	-50dBm <sup>2</sup>	- 50 dBm <sup>2</sup> -> + 52.3 dBm <sup>2</sup>

Table 9: CAN1\_Target\_2 message description (0x702)

The following is a detailed description of each of the signals in message 0x702:

1. **NoOfTarget\_2** – Target number (same as **NoOfTarget\_1** in 0x701)
2. **Tar\_PdH0** – Target false alarm probability
3. **Tar\_Length** – Target length
4. **Tar\_Width** – Target width
5. **Tar\_Ang\_stat** – Target angle status
  - 0 -> Expanded target
  - 1 -> Point target
  - 2 -> Digital
  - 3 -> invalid
6. **Tar\_Type** – Target type
  - 0 -> No target
  - 1 -> Oncoming
  - 2 -> Stationary
  - 3 -> Invalid data
7. **Tar\_Ang** – Target angle
8. **Tar\_RCSValue** – Radar cross section

### 9.2. Object List

The object list output consists of four messages: CAN1\_VersionID (0x700), CAN1\_Obj\_Status (0x60A), CAN1\_Obj\_1 (0x60B) and CAN1\_Obj\_2 (0x60C). For each cycle the status message and 40 message pairs (CAN1\_Obj\_1, CAN1\_Obj\_2) for identified objects are sent.



Figure 10: CAN1\_Obj\_Status message structure (0x60A) (Byte 7 is not used)

Signal	Start	Len	Byte Order	Value Type	Res	Offset	Value Range
NoOfObjects	0	8	Motorola	Unsigned	1	0	0 -> 255 tpcl.: 1 -> 40
MeasCounter	16	16	Motorola	Unsigned	1	0	0 -> 65535 Recount starts after 65535
(§) SensorOutput Reduced	24	1	Motorola	Unsigned	1	0	0->1
(§) Sensor External Disturbed	25	1	Motorola	Unsigned	1	0	0->1
(§) Sensor SwitchedOff	26	1	Motorola	Unsigned	1	0	0->1
(§) Sensor Misaligned	27	1	Motorola	Unsigned	1	0	0->1
(§) Sensor Defective	28	1	Motorola	Unsigned	1	0	0->1
(§) SensorSupply VoltageLow	29	1	Motorola	Unsigned	1	0	0->1
(§) Sensor RxInvalid	30	1	Motorola	Unsigned	1	0	0->1
(§) Reserved	31	1	Motorola	Unsigned	1	0	0->1
InterfaceVersion Number	32	4	Motorola	Unsigned	0	0	0 -> 16
(§) NumOfLanes Right	36	2	Motorola	Unsigned	1	0	0->3
(§) NumOfLanes Left	38	2	Motorola	Unsigned	1	0	0->3
(§) LatDistTo Border Left	40	8	Motorola	Unsigned	0.1 m	0 m	0 m ->25.4 m
(§) LatDistTo Border Right	48	8	Motorola	Unsigned	0.1 m	0 m	0 m ->25.4 m

Table 10: CAN1\_Obj\_Status message description (0x60A)

**(§) → Signal is exclusive ONLY for the automotive applications and NOT for industrial functions.**

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

1. **NoOfObjects** – Number of Objects
2. **MeasCounter** – Counts the number of measurements made
3. (§) **SensorOutputReduced** – Value 1 indicates output of sensor is reduced
4. (§) **SensorExternalDisturbed** - Value 1 indicates Sensor has external disturbance
5. (§) **SensorSwitchedOff** - Value 1 indicates Sensor is switched off due to High Temperature
6. (§) **Sensor Misaligned** - Value 1 indicates Sensor is misaligned
7. (§) **SensorDefective** - Value 1 indicates Sensor is defective
8. (§) **SensorSupplyVoltageLow** - Value 1 indicates Low Voltage
9. (§) **SensorRxInvalid** – Value 1 indicates data received is invalid
10. **InterfaceVersionNumber** – Unique number that identifies the version of the CAN interface
11. (§) **Reserved** – this bit is reserved for automotive applications
12. (§) **NumOfLanesRight** – Position of Object expressed as Number of adjacent lanes to the Right
13. (§) **NumOfLanesLeft** – Position of Object expressed as Number of adjacent lanes to the Left

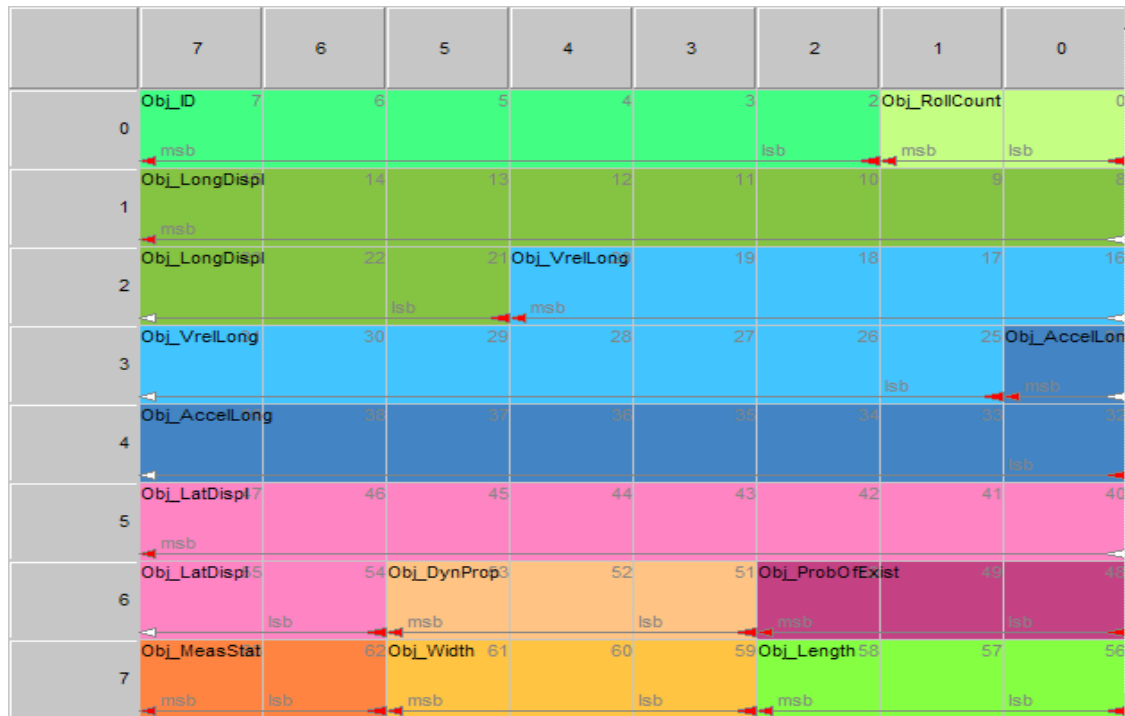


Figure 11: CAN1\_Obj\_1 message structure (0x60B)

Signal	Start	Len	Byte Order	Value Type	Res	Offset	Value Range
Obj_RolCount	0	2	Motorola	Unsigned	1	0	0 -> 3
Obj_ID	2	6	Motorola	Unsigned	1	0	0 -> 63 tpcl.: 1 -> 40
Obj_LongDispl	21	11	Motorola	Unsigned	0.1 m	0 m	0 m -> 204 m
Obj_VrelLong	25	12	Motorola	Unsigned	0.0625 m/s	-128 m/s	- 128 m/s -> + 127.9375 m/s
Obj_AccelLong	32	9	Motorola	Unsigned	0.0625 m/s <sup>2</sup>	-16 m/s <sup>2</sup>	-16 m/s <sup>2</sup> -> + 15,9375 m/s <sup>2</sup>
Obj_ProbOfExist	48	3	Motorola	Unsigned	1	0	0 -> 7
Obj_DynProp	51	3	Motorola	Unsigned	1	0	0 -> 7 (only 0->4 used)
Obj_LatDispl	54	10	Motorola	Unsigned	0.1 m	-52 m	-51.9 m -> 52 m
Obj_Length	56	3	Motorola	Unsigned	1	0	0 -> 7
Obj_Width	59	3	Motorola	Unsigned	1	0	0 -> 7
Obj_MeasStat	62	2	Motorola	Unsigned	1	0	0 -> 3

Table 11: CAN1\_Obj\_1 message description (0x60B)

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

1. **Obj\_RolCount** – Rolling counter
2. **Obj\_ID** – Object ID
3. **Obj\_LongDispl** – Longitudinal displacement
4. **Obj\_VrelLong** – Relative longitudinal speed
5. **Obj\_AccellLong** – Relative longitudinal acceleration
6. **Obj\_ProbOfExist** – Probability of existence calculated for an Object
  - 0: invalid
  - 1: < 25 %
  - 2: < 50 %
  - 3: < 75 %
  - 4: < 90 %
  - 5: < 99 %
  - 6: < 99.9 %
  - 7: < 99.99%
7. **Obj\_DynProp** – Dynamic property:
  - 0: unclassified
  - 1: standing
  - 2: stopped (has never been moving before)
  - 3: moving
  - 4: oncoming
8. **Obj\_LatDispl** – Lateral displacement
9. **Obj\_Length** – Object length
  - 0: unknown
  - 1: < 0.5 m
  - 2: < 2 m
  - 3: < 4 m
  - 4: < 6 m
  - 5: < 10 m
  - 6: < 20 m
  - 7: exceeds

10. **Obj\_Width** – Object width

- 0: unknown
- 1: < 0.5 m (pedestrian)
- 2: < 1 m (bike)
- 3: < 2 m (car)
- 4: < 3 m (truck)
- 5: < 4 m
- 6: < 6 m
- 7: exceeds

11. **Obj\_MeasStat** – Object measurement status

- 0: no object
- 1: new object
- 2: object not measured
- 3: object measured



Figure 12: CAN1\_Obj\_2 message structure (0x60C) (Bytes 3-7 and the Bit in White are unused)

Signal	Start	Len	Byte Order	Value Type	Res	Offset	Value Range
Obj_RCSValue	0	8	Motorola	Unsigned	0.5 dBm <sup>2</sup>	-64 dBm <sup>2</sup>	- 64 dBm <sup>2</sup> -> + 63.5 dBm <sup>2</sup>
Obj_LatSpeed	8	8	Motorola	Unsigned	0.25 m/s	-32 m/s	- 32 m/s -> + 31.75 m/s
Obj_Obstacle Probability	16	7	Motorola	Unsigned	1 %	0 %	0 % ->100 %

Table 12: CAN1\_Obj\_2 message description (0x60C)

The following is a detailed description of each of the signals in message 0x60C:

1. **Obj\_RCSValue** – Radar cross section (RCS) is the measure of the reflective strength of a target or in other words it is the measure of power scattered in a given spatial direction when a target is illuminated by an incident wave.
2. **Obj\_LatSpeed** – Object lateral velocity
  - negative value: object moves left to right
  - positive value: object moves right to left
3. **Obj\_ObstacleProbability** – Probability that the Object is an Obstacle

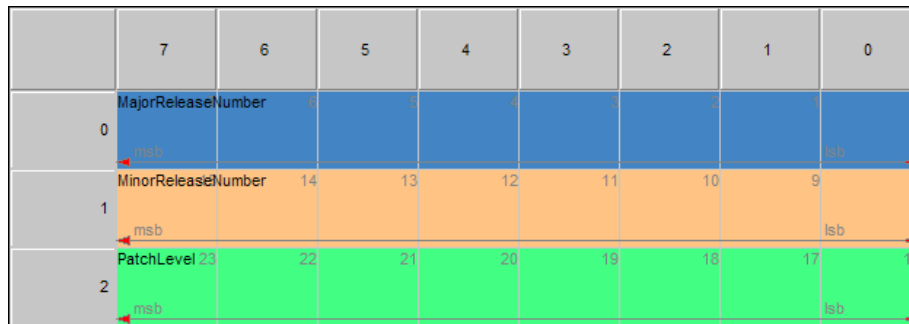


Figure 13: CAN1\_VersionID message structure (0x700) (Bytes 3-7 are unused)

Signal	Start	Len	Byte Order	Value Type	Res	Offset	Value Range
#MajorRelease Number	0	8	Motorola	Unsigned	1	0	0 -> 256
#MinorRelease Number	8	8	Motorola	Unsigned	1	0	0 -> 256
#PatchLevel	16	8	Motorola	Unsigned	1	0	0 -> 256

Table 13: CAN1\_VersionID message description (0x700)

The following is a detailed description of each of the signals in message 0x700:

Each of the following three numbers is Binary Coded Numbers of the MCU

1. **#MajorReleaseNumber** – Major release number
2. **#MinorReleaseNumber** – Minor release number
3. **#PatchLevel** – Patch level

**# → Valid only for the Company's internal use**