# FULL DOCUMENTATION

## TRAFFIC MANAGEMENT SENSOR

TOPGRD (UMRR-A4 TYPE 171)

## CONTENTS

# 1 ABOUT THE RADAR SENSOR

In the following chapter, a general overview of the radar sensor is provided.

## 1.1 FIELDS OF APPLICATION

The smartmicro radar sensor can be used for many applications, such as:

- Intersection management
    - Stop bar detection
    - Advance detection
    - Combined stop bar and advance detection
    - Queue length estimation
- Arterial management
    - Traffic counting and classification
    - Wrong way driving detection
    - Incident detection
    - Ramp metering
    - Statistical analysis
- Enforcement
    - Red-light enforcement
    - Stationary speed enforcement
    - Portable speed enforcement
    - Mobile speed enforcement

Please note that the radar system – although being well optimized to be used for these applications – can neither achieve a detection probability of 100% nor a false alarm rate equal to zero.

## 1.2 MAIN FEATURES

The main features of the sensor are:

- Variety of applications
- Lane-specific detection
- Individual object tracking
- Flexible installation
- Statistics and event trigger modules
- Easy-to-use Traffic User Interface (TUI) software

## 1.3 PRINCIPLE OF OPERATION

Each radar sensor directly measures the following parameters of all moving objects simultaneously in the field of view, relative to the sensor:

- Direct unambiguous Doppler measurement (speed)
- Direct range measurement
- Direct azimuth angle measurement (horizontal angle)
- Direct elevation angle measurement (vertical angle)

Those data are stabilized by tracking algorithms. Stopped objects are kept in the tracking memory. The true vector of the relative speed is calculated and a data transformation into Cartesian coordinates is performed. Those tracked and transformed data are then transmitted via the chosen communication interface.

The interface to the superior system (detector card, PC, or similar) is a list of tracked objects transmitted via RS485, Ethernet, or another interface.

Each sensor comprises RS485 as standard interface for communication and an Ethernet interface.

### FLEXIBILITY OF THE RADAR WAVEFORM

The radar sensor supports a variety of stand-alone or combined radar waveforms. The operational mode can be determined via the firmware.

## 1.4 TRACKED OBJECT INTERFACE

The result of tracking is an object list with the following parameters:

- X-position
- Y-position
- Absolute velocity
- Heading angle
- Length
- Object ID and more

In any case, a visualization of objects is possible using the Traffic Management Configurator (TUI) software in any PC equipped with a RS485 or Ethernet interface. A visualization of targets can be enabled on request.[1]

For further information on the performance and communication options of the radar sensor, please refer to the respective datasheet on our website.

## 1.5 STATISTICS MODULE FEATURES

The Statistics Module is a software extension, which is based on the tracked object list. The tracked objects provide a high update rate of data per vehicle, which are interpreted and accumulated by the Statistics Module to reduce the amount of data transferred to the user. It also allows for the transfer of data at user-selectable time intervals. The user can select the statistics data separately.

Basic features are:

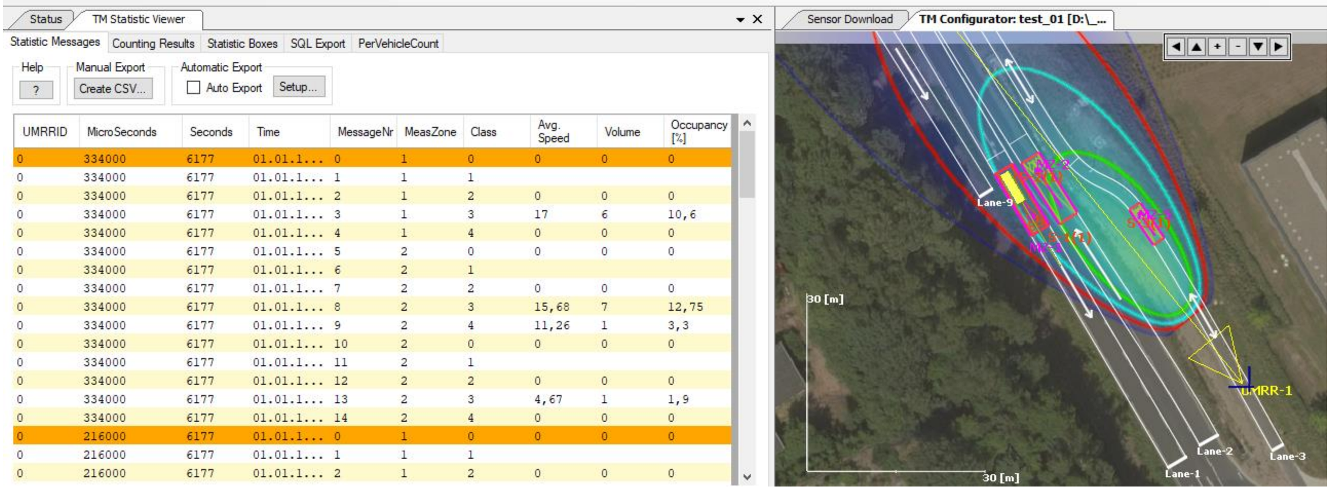
- Lane-specific reports
- Reports per measurement zone (max. 32 measurement zones)
- Report of statistical data
    - Volume
    - Occupancy
    - Average speed

1 Depending on the interfaces provided by the radar sensor; please check the system requirements of the TUI.

Extended features are:

- Reports per vehicle class
- Report of statistical data
    - 85 percentile speed
    - Headway
    - Gap
- Capability to either polling or reporting activities

Please contact us for a detailed documentation.




| UMRRID | MicroSeconds | Seconds | Time | MessageNr | MeasZone | Class | Avg. Speed | Volume | Occupancy [%] |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 334000 | 6177 | 01.01.1... | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 334000 | 6177 | 01.01.1... | 1 | 1 | 1 | | | |
| 0 | 334000 | 6177 | 01.01.1... | 2 | 1 | 2 | 0 | 0 | 0 |
| 0 | 334000 | 6177 | 01.01.1... | 3 | 1 | 3 | 17 | 6 | 10,6 |
| 0 | 334000 | 6177 | 01.01.1... | 4 | 1 | 4 | 0 | 0 | 0 |
| 0 | 334000 | 6177 | 01.01.1... | 5 | 2 | 0 | 0 | 0 | 0 |
| 0 | 334000 | 6177 | 01.01.1... | 6 | 2 | 1 | | | |
| 0 | 334000 | 6177 | 01.01.1... | 7 | 2 | 2 | 0 | 0 | 0 |
| 0 | 334000 | 6177 | 01.01.1... | 8 | 2 | 3 | 15,68 | 7 | 12,75 |
| 0 | 334000 | 6177 | 01.01.1... | 9 | 2 | 4 | 11,26 | 1 | 3,3 |
| 0 | 334000 | 6177 | 01.01.1... | 10 | 2 | 0 | 0 | 0 | 0 |
| 0 | 334000 | 6177 | 01.01.1... | 11 | 2 | 1 | | | |
| 0 | 334000 | 6177 | 01.01.1... | 12 | 2 | 2 | 0 | 0 | 0 |
| 0 | 334000 | 6177 | 01.01.1... | 13 | 2 | 3 | 4,67 | 1 | 1,9 |
| 0 | 334000 | 6177 | 01.01.1... | 14 | 2 | 4 | 0 | 0 | 0 |
| 0 | 216000 | 6177 | 01.01.1... | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 216000 | 6177 | 01.01.1... | 1 | 1 | 1 | | | |
| 0 | 216000 | 6177 | 01.01.1... | 2 | 1 | 2 | 0 | 0 | 0 |

*Figure 1-1: Example of the Statistics Module output in the TUI*

Please refer to chapter 5 which describes the Statistics Module features, the output ports, and the available commands.

## 1.6 EVENT TRIGGER MODULE FEATURES

The Event Trigger Module is a software extension, which is based on the tracked object list. With a focus on certain events, the user can define and select from a variety of trigger conditions for each zone. Those triggers can be assigned to virtual relays. Each sensor supports up to 64 virtual relays. The user can select the output from the virtual relay status conditions separately. The output rate is also user selectable.

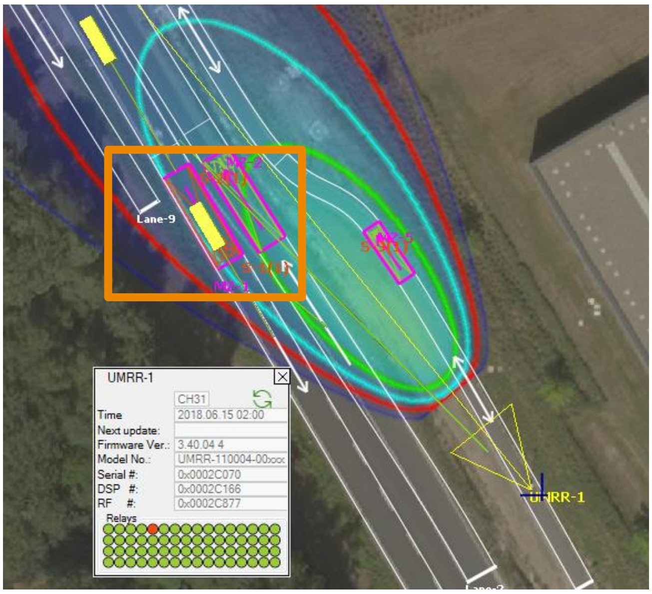



*Figure 1-2: Example of the Event Trigger Module setup and output in the TUI*

The following features are provided:

- Lane-specific reports
- Reports per zone (max. 32 zones)
- Up to 64 virtual relays are triggered
- Types of trigger
  - Presence
  - Speed
  - Estimated Time of Arrival (ETA)[2]
  - Vehicle class
  - Wrong way driving
  - Queue length estimation
  - Custom trigger
- Individual delay and extension of trigger signals

Please refer to chapter 5 which describes the Event Trigger module features, the output ports, and the available commands.

[2] For the United States, a one-time ETA is provided.

## 1.7 SENSOR INTERFACES

The sensor supports the following interfaces:

- RS485 full-duplex
    - 115kbps per default, higher baud rate is possible
- 100Mbps Ethernet per default

The interfaces can be accessed through the connector on the rear side of the sensor. Please refer to the sensor datasheet on our website for further details.

*Figure 1-3: Rear and front side of the sensor*

### WIRELESS COMMUNICATION

Several LTE modems have been tested and validated to work with the sensor. Please contact us for further details.

## 1.8 SENSOR AND HARDWARE IDENTIFICATION

The sensor housing is tagged with a type sticker containing the product description and the serial number. It also indicates which side of the sensor is the top side.

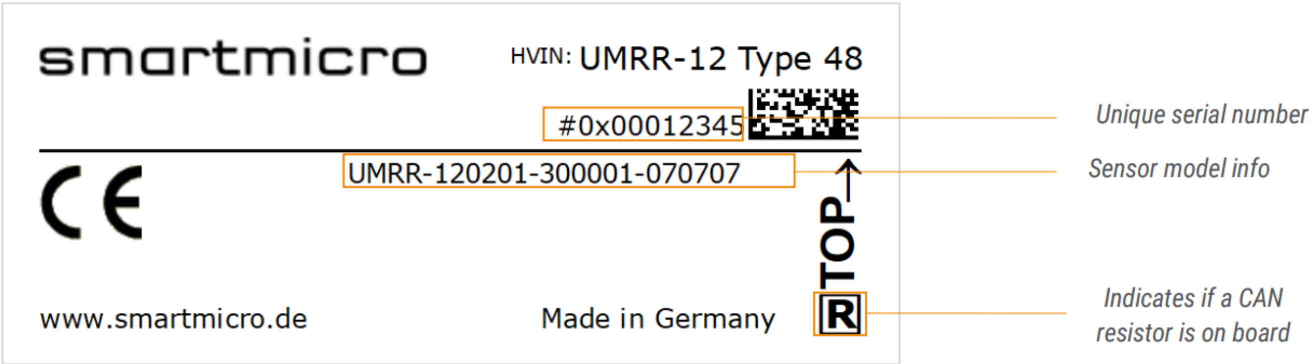



*Figure 1-4: Sticker example*

Each sensor is labelled with the respective sensor model information "UMRR-xxyyzz-aabbcc-ddeeff"

| | | |
|---|---|---|
| - | UMRR | Universal Medium Range Radar developed by smartmicro |
| - | -xx | DSP board generation |
| - | -yy | DSP board derivative/version |
| - | -zz | DSP board revision |
| - | -aa | RF board (antenna) |
| - | -bb | RF board derivative/version |
| - | -cc | RF board revision |
| - | -dd | Housing type |
| - | -ee | Housing version |
| - | -ff | Housing revision |

Additionally, the DSP board and the RF board have their own unique serial numbers.

## 1.9 COORDINATE SYSTEM

The angle measurement of the sensor always refers to the x-axis and is always between +180° and -180°, depending on the direction of the speed vector.

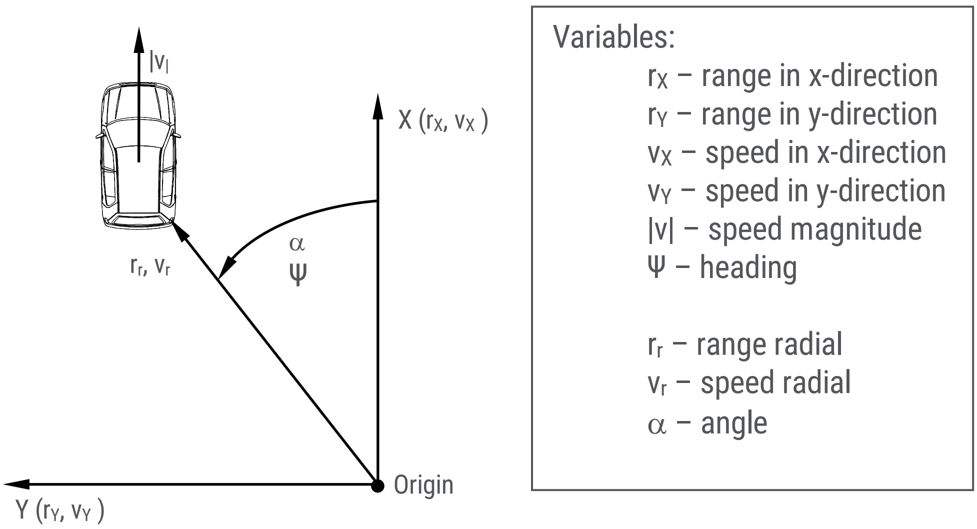



*Figure 1-5: Drawing of the coordinate system*

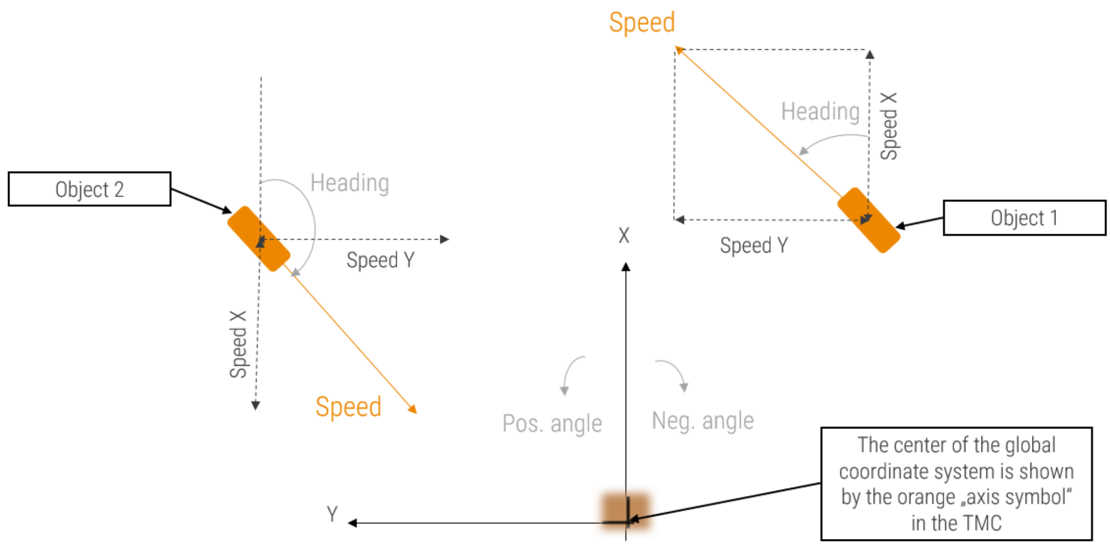



*Figure 1-6: Angle signs in the coordinate system*

## 1.10 GROUNDING REQUIREMENTS

The housing of the sensor is not grounded but connected to the negative supply voltage instead. To assure the correct operation of the sensor, please contact us for grounding instructions.

## 1.11 CABLE SPECIFICATIONS

The cable requirements for the installation of the sensor are:

- For the utilization with a smartmicro Junction Box (J-Box):
  - Outer diameter of 9mm to 13mm or 9.35in to 0.51in
  - Cable outlet also available for 6.5mm to 9.5mm or 0.26in to 0.37in
- Power wires:
  - AWG18, or rather cross section of 1mm² or larger recommended[3]
  - For all installations, the minimum voltage indicated in the datasheet needs to be granted for the operating unit
  - For NEMA cabinet installations, a voltage drop by less than 14V is required
- Data wires:
  - Twisted pair for RS485 full-duplex
  - Four wires, two wires each as twisted pair for RS485 full-duplex and 10/100 Mbps Ethernet
  - Z can vary from 100 Ohm to 120 Ohm at 100kHz/1MHz
  - AWG24, or rather cross section of 0.22mm² or larger
  - Loop resistance better than 190 Ohm/1000m

Please note that smartmicro can give no warranty on cable types other than those verified by smartmicro throughout testing and recommended as such. It is the customer's own responsibility to test and verify other cables for their particular purpose and installation variants, especially regarding communication capabilities.

### TESTED CABLES

The following field cables have been verified throughout testing by smartmicro, please contact us for further information.

- Lapp UNITRONIC BUS YV COMBI IBS 3x2x0.22, 3x1.0 ROHS
  - Manufacturer part no. 2170217
  - Rated for direct burial or natural UV resistance
  - Tested distance: 161m or 528ft
- Lapp UNITRONIC LAN 1000 s/FTP Cat. 7 (L) PE 4x2xAWG 23/1
  - Manufacturer part no. 2170198; Cat.7 cable
  - Rated for direct burial
  - Tested distance: 100m or 328ft
- Draka cable UC300 CAT5e AWG23
- Draka cable UC900 CAT7 AWG24

[3] For the utilization with a J-Box, a conductor reduction at the clamp by 0.5mm² is required.

# 2 ACCESSORIES

The sensor is enriched by a set of accessories to meet various customer needs. The following products are only a brief selection, for a complete list please visit our website.

## 2.1 BRACKET

To facilitate the mounting and adjusting of the sensor, we offer two types of brackets. Our standard brackets allow for adjusting the elevation angle of the sensor. They are designed to facilitate the mounting on a pole or wall, as well as the usage of the sensor on a tripod. Our advanced brackets allow for adjusting both azimuth and elevation angle. They are designed to further cover both horizontal and vertical pole mount, as well as straight or angled orientation. For further information, please refer to the datasheet on our website.

## 2.2 JUNCTION BOX

The smartmicro Junction Box (J-Box) offers a simple and reliable way to connect smartmicro radars to any home-run cable, connecting the sensor to the cabinet. It can accommodate a wide range of cables inside a water-tight sealing and connects the cables to a terminal block. The J-Box also protects the radar from voltage surges and overvoltage. It can be easily added to the back of the sensor. A simple J-Box is available for RS485 only and a full J-Box version is available for all interfaces. For further information, please refer to the datasheet on our website.

## 2.3 RS485 TO USB CONVERTER

By means of the RS485 to USB converter, you can easily connect the sensor to a PC at low cost. For 24/7 installations, an industrial solution like Moxa Uport 1130I or 1150 is suggested. Please visit our website for further information.

## 2.4 CABINET INTERFACES

### 2.4.1 TRAFFIC MANAGEMENT INTERFACE BOARD

Our Traffic Management Interface Board (TMIB) connects up to four sensors at an intersection and enables their connection to any type of traffic controller. The data lines from multiple connected radars are made available on one single Ethernet interface. For the US market, there are NEMA TS1/TS2 compliance and SDLC interfaces available. For further information, please refer to the datasheet on our website.

### 2.4.2 CABINET INTERFACE OPTION

The smartmicro Cabinet Interface Option (CIO) is an interface panel that provides power surge protection for up to four radar sensors and a Traffic Management Interface Board (TMIB). The typical installation of a CIO includes a TMIB that provides RS485 connectivity to the radar sensors.

The interface panel has individual power on/off switches and LED indicators for each radar sensor to simplify the installation. The CIO provides both electrical connectors for the cables coming in from the four sensors outside the cabinet and multiple stages of electrical surge suppression to protect the cabinet equipment from external surges and noise. This surge protection includes Gas Discharge Tubes (GDTs), Transient Voltage Suppression diodes (TVSs), high-speed resettable electronic fuses (TBUs) and a replaceable fuse. Power is 110V or 220V AC to the replaceable power supply on the interface panel, typically wired from the protected side of the cabinet power distribution. No supplemental surge suppression is required.

The CIO consists of a Rail, one CIO Module, power supply, circuit breaker, terminal blocks and end brackets. It comes with power signals pre-wired from the factory and has the following features:

- Inbuilt surge and power protection
- Four LEDs indicate the power status of each sensor/channel
- One LED indicates the status of the main power supply
- Two LEDs per sensor to indicate RS485 communication link and activity

For further information, please refer to the datasheet on our website.

## 2.5 COMMUNICATION INTERFACES

### 2.5.1 RELAY OUTPUT OPTION

The smartmicro Sensor Relay Option (SRO) can be attached to the back of the sensor and offers 8 hardware relays and surge protection in addition to the sensor's RS485 or Ethernet communication interface. It is available as a separate accessory for the sensor and offers surge protection for all signals.

For further details, please refer to the datasheet on our website.

### 2.5.2 CABINET RELAY OPTION

The smartmicro Cabinet Relay Option (CRO) is a hardware module that extends a smartmicro radar sensor by eight hardware relays via the standard RS485 communication interface.

It can be used to implement traffic applications without an additional controller unit being necessary and has the following features:

- Connector to a sensor cable
- Connector to power supply
- Connector for RS485 communication to a PC
- Eight configurable solid-state relays per sensor (normally-closed (NC) contacts)
- Surge protection on power, communication, and relay lines

The CRO Base consists of a rail, one CRO Module, power supply, circuit breaker, terminal blocks and end brackets. It comes with power signals pre-wired from the factory.

For the use of two, three or four sensors, the CRO Base can be extended with the respective number of CRO Modules. Please note that only the first CRO Module is pre-wired.

For further details, please refer to the datasheet on our website.

### 2.6 CIO AND JUNCTION BOX SURGE SUPRESSION

The surge protection of CIO and JBOX works as follows. Gas Discharge Tubes (GDTs) on all incoming power and data lines are used to protect against strong voltage spikes. At a second stage, Transient Voltage Suppression diodes (TVSs) eliminate any remaining unwanted voltage spikes. For all data lines, high-speed resettable electronic fuses (TBUs) offer an additional level of protection, which is not needed nor possible for power lines.

In case of a near-by lightning strike, all GDTs will ignite and divert the power surge away from the equipment. For this, they create an intentional short-circuit from the incoming wires to ground. Once the voltage on each wire has dropped below a threshold voltage of below 10 V (may vary slightly in different revisions), the protection mode is ended, and normal operation is resumed. For all data lines, this is instantly the case after the power surge has ended. On the positive supply line however, the voltage provided by the power supply may keep the protection mode running. As connected radars will not receive enough voltage while the GDT is still in protection mode, the power supply may need to be switched off and then on again to retain normal radar operation.

# 3 SELECTING THE SENSOR MODEL AND LOCATION

Please note that frequency regulations may vary from country to country and that the usage of the lower power mode of the sensor may be necessary.

## 3.1 SELECTING THE RIGHT SENSOR MODEL

This sensor model is the right choice if one or more of the following situations apply:

- Higher range resolution than achievable with UMRR-11 Type 44/45 or UMRR-12 T48 is desired
- Traffic density is very high
- Best possible detection and classification accuracy is required under challenging conditions
- Traffic density in a certain area
- than two radar sensors should be applied without any interference, since up to four frequency bands are available for UMRR-A4 Type 171

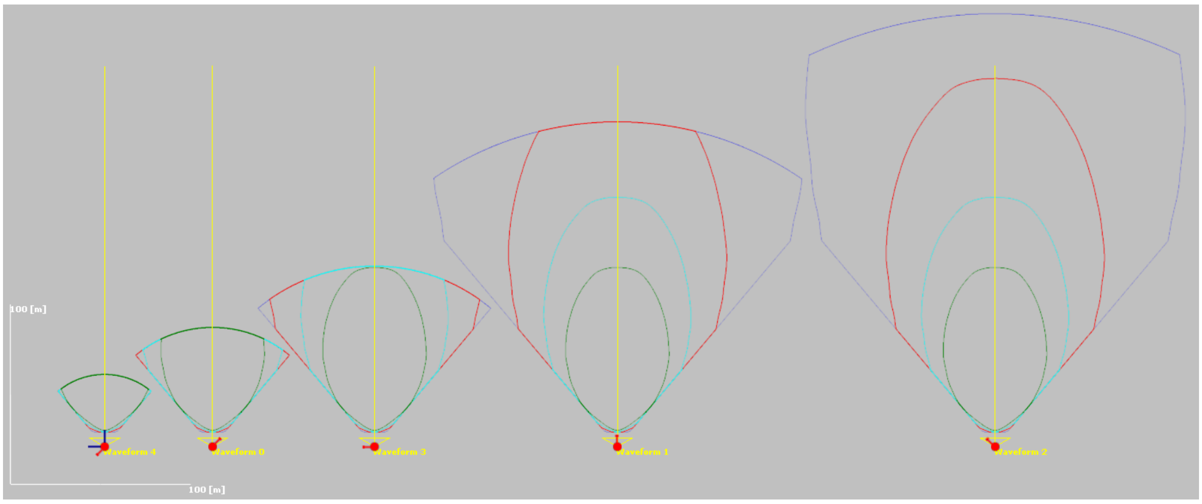



*Figure 3-1: General characteristics of the antenna (different waveforms)*

## 3.2 SELECTING THE RIGHT MOUNTING POSITION

To achieve the best radar performance, the adequate location needs to be chosen carefully. The responsible person for installing the sensor needs to go through the following steps:

- Verify that there are >20m or >65ft between the maximum range and the area of interest.
- Verify that the lanes are fully covered, including some allowance at the beam limits. The illustration below shows that the example (B) is selected too tight.
- Verify that the horizontal angle to the road is within -25° and +25°.
- Verify that the vertical angle to the ground is within -8° and +3°. Typically, -12° to +5° are plausible.[4]
- Mount the radar at suitable height, typically at 6m or 19.6ft.[5]

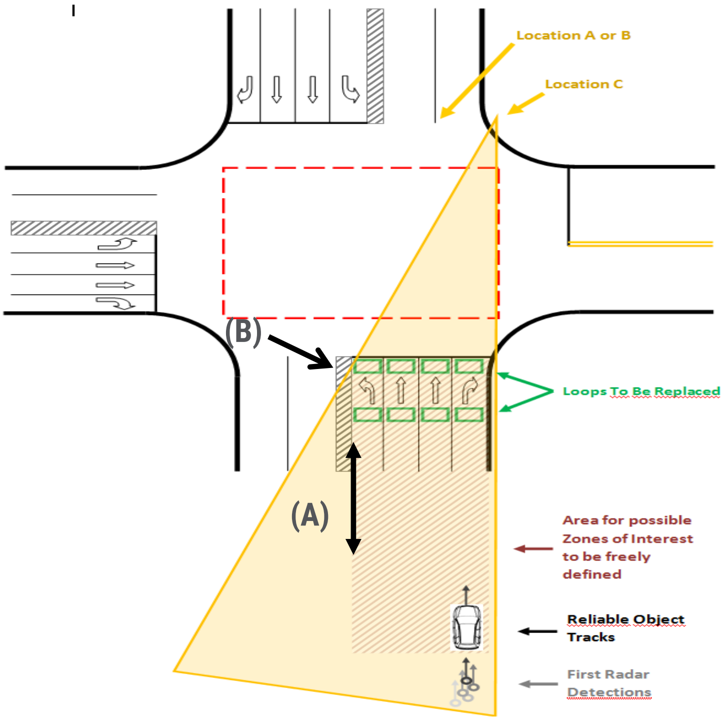



*Figure 3-2: Select Area of Interest*

Also, the sensor should be mounted on a stiff and solid mounting base. The smaller the angle towards the stop bar, the better. A smaller angle improves the utilization of the sensor beam and occlusion of vehicles in adjacent lanes becomes less likely. This means, that mounting the sensor on a mast arm is usually better than mounting it close to the ground. However, the mast arm must not move due to wind or other environmental influences. If it does, a position of the mast arm needs to be chosen, where the movement is kept to a minimum.

[4] A steeper elevation angle is possible but limiting the maximum range. A negative elevation angle means that the sensor is pointing towards the road.
[5] The mounting height may affect the maximum detection range. Occlusion needs to be considered.

Please note that vibration, oscillation, or any other kind of movement will reduce the sensor performance.

On existing infrastructure, the sensor can be mounted in the following positions:

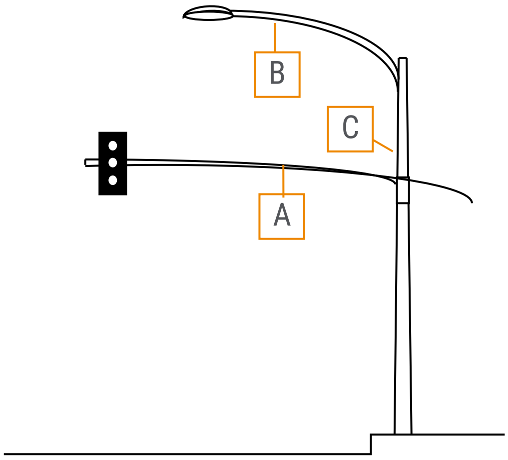



*Figure 3-3: Mounting positions on existing infrastructure*

A) On the mast arm close to the vertical pole (ideal position)
B) Adjacent to a luminaire
C) On a vertical pole

We recommend position (A) for best performance, as a stiff and motion-free mounting base is required while a small angle towards the detection zone is recommended. If the structural conditions of the mast arm do not allow for a stiff attachment of the sensor at position (A), or if no mast arm is present, positions (B) or (C) are potential alternatives.

Please refer to the respective sensor datasheet for a recommendation of the distance to the stop bar or the area of interest, as well as the azimuth and elevation angle that should be used to achieve the best performance.

### 3.3 SELECTING ALTERNATIVE MOUNTING POSITIONS

From time to time, it may not be possible to select the right mounting position and mast arm location that smartmicro recommends as indicated in section 4.2. There are other types of mounting positions commonly found can could accommodate a radar sensor but is generally not recommended by smartmicro as indicated below.

- Wooden Poles
- Span wires
- Strain poles

### 3.3.1 WOODEN POLE MOUNTING

A typical wooden pole installation at an intersection is illustrated below:

*Figure 3-4: Typical wooden pole installation*

Smartmicro does not recommend the mounting of radar sensors onto wooden poles for normal operation. Wooden poles, that are used for both temporary traffic signal locations and span wire locations, are not usually mounted onto a rigid base. Such poles are usually inserted into a hole that over time is subject to lateral movement. Additionally, wooden poles themselves are subject to bending or warping through hot or cold weather. All of these types of pole movement will affect the alignment of the sensor and impact the sensor's overall detection performance.

If mounting onto wooden poles, it is recommended that regular checking of sensor alignment is performed, especially during periods of adverse or changing weather conditions. Indications that a sensor may require re-alignment is when there are regular instances of missed or inaccurate detections as a result of the virtual detection zone(s) moving from their original position.

### 3.3.2 SPAN WIRE MOUNTING

A typical span wire installation at an intersection is illustrated below:

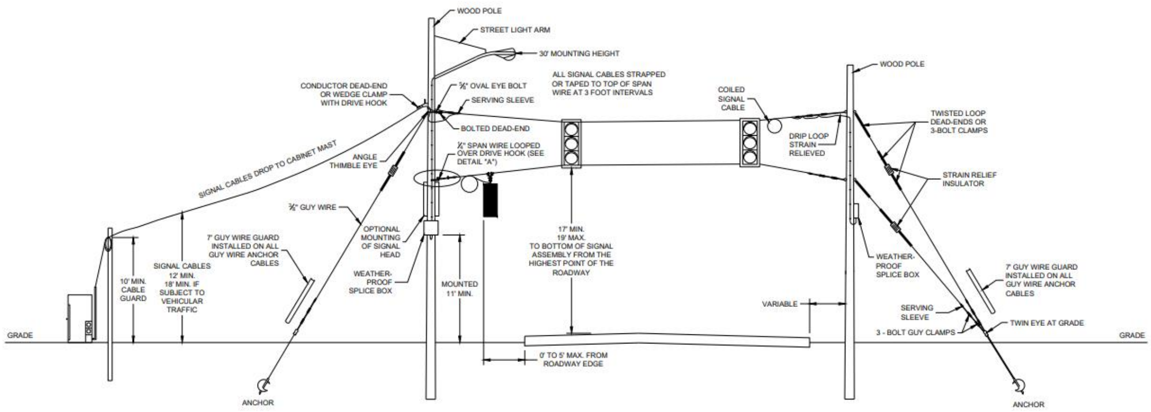



*Figure 3-5: Typical span wire installation*

Smartmicro does not recommend the mounting of radar sensors directly onto span wires whether single or dual wires. Generally, span wires do not offer sufficient firmness of radar position since the span wires are affected by wind (lateral movement), and heat or cold (vertical movement through contraction and expansion of the wires).

While there are instances of dual span wire installations, regular checking of the sensor's alignment to the stop bar detection area is required to ensure optimum detection performance.

Such an installation is illustrated below.

*Figure 3-6: Dual Span wire installation*

### 3.3.3 STRAIN POLE MOUNTING

A strain pole is defined as a single vertical steel tapered pole upright set at each four corners of an intersection. Cable and grounded wire are strung between the uprights to allow for traffic signal lights to be positioned above the appropriate travel lanes. A typical Strain pole installation at an intersection is illustrated below:

*Figure 3-7: Strain Pole installation*

Smartmicro recommends the mounting of radar sensors onto Strain poles for normal operation. Strain poles, although supporting span wire installations, are usually mounted onto a concrete base.

In all the mounting positions mentioned in this section, wooden or strain pole, it is recommended that the installer verifies the horizontal (azimuth) angle of the sensor to the stop bar to ensure the position is within the recommended settings. It is not uncommon to find that wooden and/or strain poles are set back from the road more than mast arms that leads to both a large angle looking towards the approach/stop bar, as well as limiting the field of view for advance detection.

### 3.4 COVERING THE AREA OF INTEREST

The radar beam covers an oval shaped area on the ground. The responsible person for the installation has to verify that the measurement point is located inside the area covered by the beam.

The following aspects need to be verified:

- (A) On all lanes there need to be >20m or >65ft covered by the beam between the area of interest and the maximum range of the sensor.
- (B) The lanes need to be fully covered by the beam. The stop bar zones should be fully covered by the cyan beam.
- (C) The green beam should not cover too much additional area on the right or left side of the stop bar. However, some allowance should be included at the outer lines of the beam on all lanes.

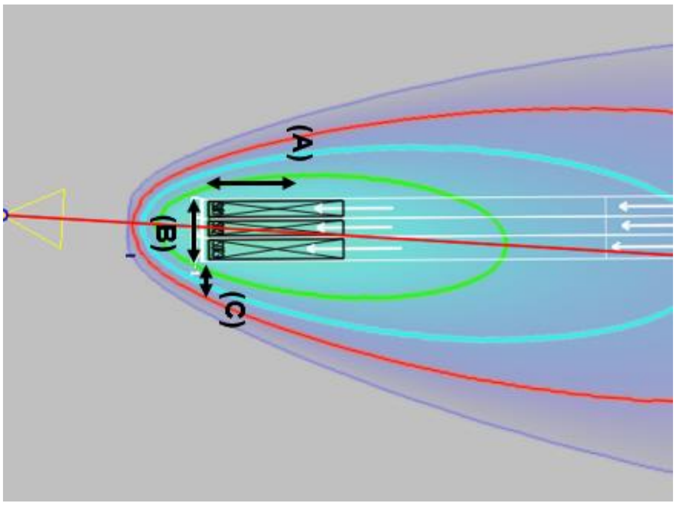



*Figure 3-8: Verify the beam coverage*

## 3.5 USING A VERTICAL INSTALLATION ANGLE

The radar beam has a limited vertical field of view, as shown in the figure below.

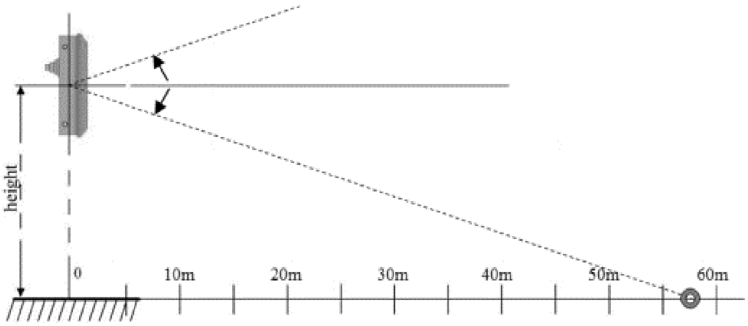



*Figure 3-9: Vertical opening angle of the radar beam*

The responsible person for installing the sensor needs to check the following steps:

- Possible mounting heights are 1m to 10m or 3ft to 33ft above the ground.
- The minimum detection range depends on the mounting height and the elevation angle.
- The beam simulation of the TUI software can be used to verify the coverage of your area of interest

The TUI software has been designed to simulate[6] the radar beam on the road according to the height, vertical and horizontal angle, as well as the selected antenna.

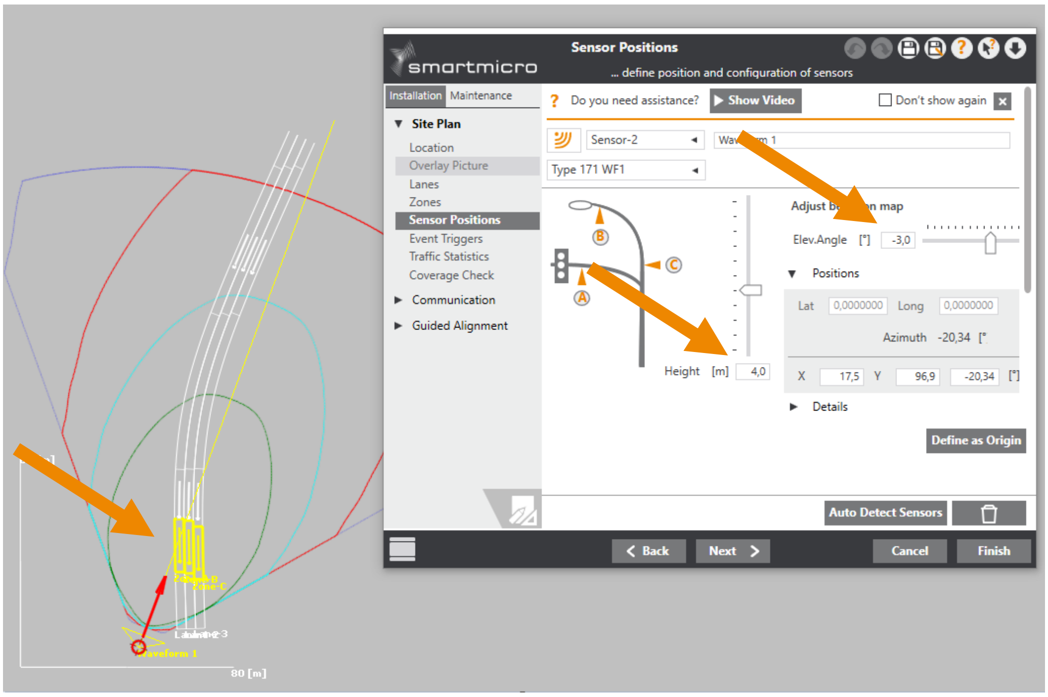



*Figure 3-10: TUI beam simulation*

[6] Please note that this is a simulation only aiming to provide guidance but not guaranteeing the detection within the area displayed.

### 3.6 TUI BEAM SIMULATION

The TUI (see section 6.7) provides a beam simulation for the responsible person to install the sensor which helps to predict the beam coverage of the road.

- Enter the sensor model information
- Select the right waveform for the application
    - Waveform 0 for medium range applications
    - Waveform 1 for long range applications
    - Waveform 2 for extra long range applications
    - Waveform 3 for extra medium range applications
    - Waveform 4 for short range applications (not yet available)
- Enter the mounting height and the elevation angle towards the road. The elevation angle towards the road should be within -12° and +3°
- The TUI will simulate the beam
-

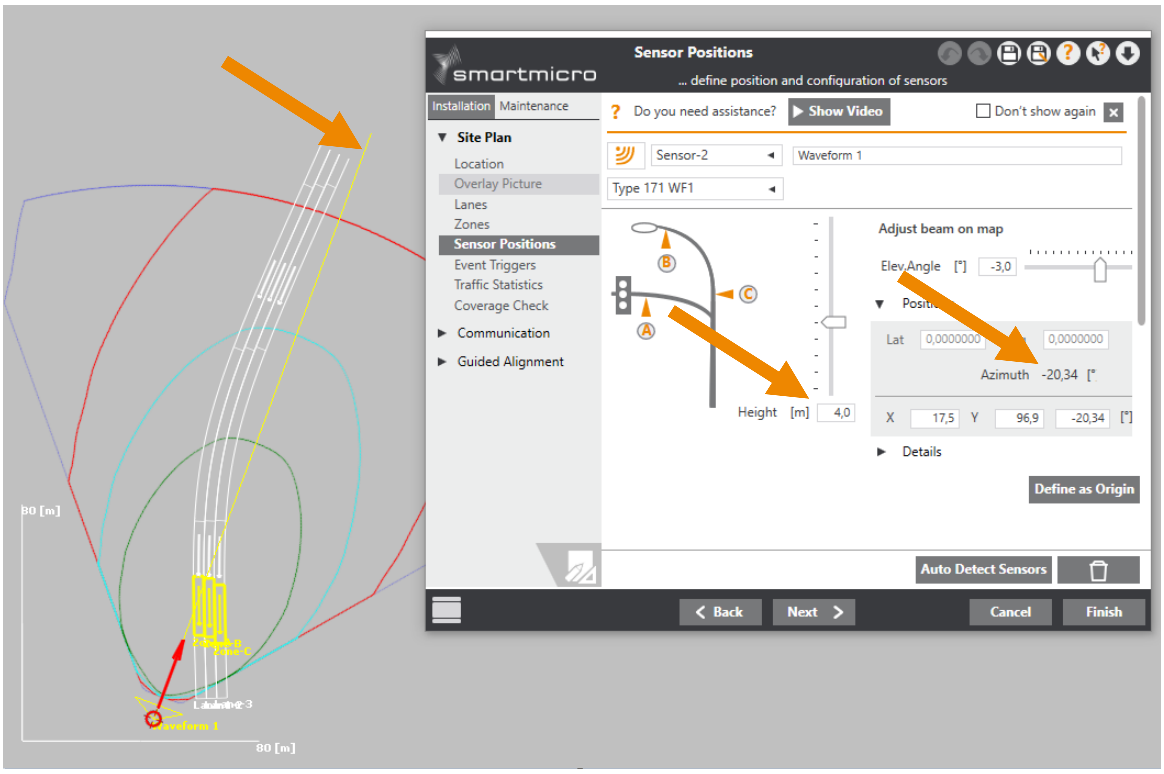



*Figure 3-11: TUI beam simulation*

# 4 DATA COMMUNICATION

The data communication protocol for the sensor is described separately, please contact us for further details and note that a Non-Disclosure Agreement (NDA) is required to obtain this document.

The user can parameterize the sensor by setting parameter instructions.

## 4.1 SENDING AN INSTRUCTION WITH THE TUI

Important instructions are provided within the sequence commander window. A list of important instructions can also be imported as .param, .status, or .command file. The TUI software is recommended for transmitting instruction messages to the sensor.

For sending instructions, please follow these steps:

1. Define a name for the instruction, for example “TestCommand”
2. Hit “+” button to add the instruction
3. Action (=Section), ParNr (=ID)., Value and UAT can be entered
4. Click “Save”

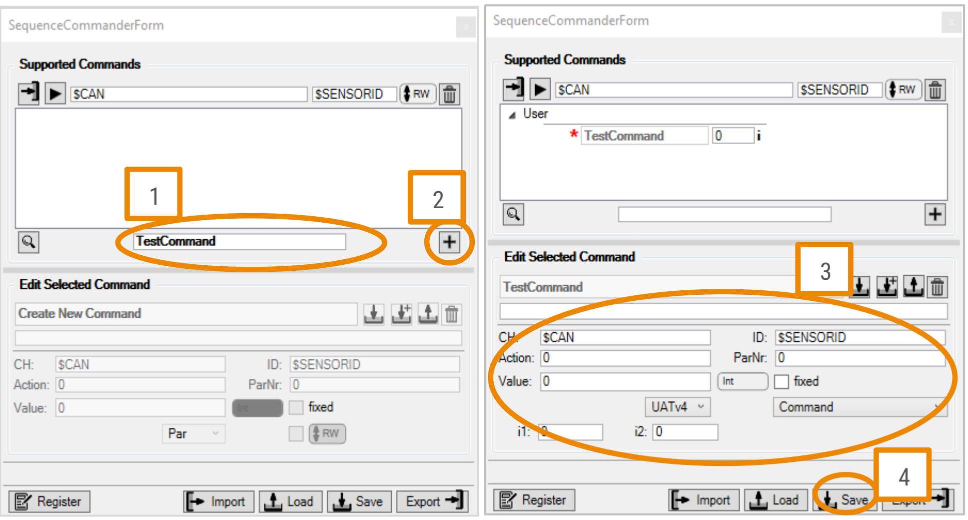



*Figure 4-1: Steps one to four*

5. The instruction can now be executed by clicking on the play button
6. The value can be changed directly in the upper part of the window
7. A predefined instruction can be loaded or exported

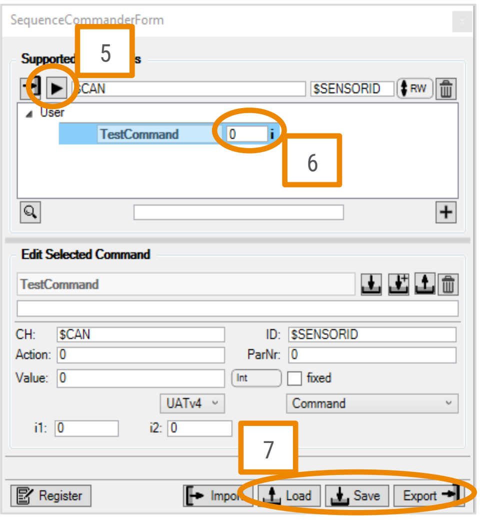



*Figure 4-2: Steps five to six*

## 4.2 BASIC INSTRUCTIONS

The following instructions are often used as user input, the TUI software (see also section 6.7) sets those instructions in alignment mode. Please note the indications that some instructions are required for managing installations with multiple radar sensors and some instructions are suggested to experienced users only.

Basic instructions (Table 4-1):

| Name | Description | ID | Default | Section | Types | Typical Values |
|---|---|---|---|---|---|---|
| tm_set_unix_time [7] [8] | Sets a new unix time as sensor time | 8 | 0 | 2020 | Command, Integer | Min.: 1. January 2018 00:00:01 → 1514764801 |
| rs485_active | Activates the RS485 interface | 11 | 1 | 2000 | Parameter, Integer | 0 = deactivated<br>1 = activated |
| eth_active | Activates the Ethernet interface | 12 | 1 | 2000 | Parameter, Integer | 0 = deactivated<br>1 = activated |
| output_control_target_list [7] | Activates the output of the detection target list | 200 | 0 | 2000 | Parameter, Integer | 0 = disabled<br>1 = enabled |
| output_control_object_list [7] | Activated the output of the object list | 201 | 1 | 2000 | Parameter, Integer | 0 = disabled<br>1 = enabled |
| output_control_triggers [7] | Activates the output of the trigger messages | 202 | 1 | 2000 | Parameter, Integer | 0 = disabled<br>1 = enabled |
| output_control_statistic [7] | Activates the output of the statistic messages | 203 | 1 | 2000 | Parameter, Integer | 0 = disabled<br>1 = enabled |

[7] Suggested to experienced users only.
[8] Not available yet.

| | | | | | | |
|---|---|---|---|---|---|---|
| output_control_pvr [7] | Activates the output of the PVR messages | 204 | 1 | 2000 | Parameter, Integer | 0 = disabled<br>1 = enabled |
| output_control_queue_length[7] | Activates the output of queue length messages | 205 | 1 | 2000 | Parameter, Integer | 0 = disabled<br>1 = enabled |
| Object Simulation | Activates the object simulation mode | 62 | 0 | 2004 | Parameter, Integer | 0 = deactivated<br>1 = simulation on straight lines<br>2 = simulation along splines/ lanes |
| Object Simulation Number of Lanes | Number of simulated lanes | 63 | 4 | 2004 | Parameter, Integer | Up to 64 lanes |
| Object Simulation Number of Objects per Lane | Number of simulated objects per lane | 64 | 5 | 2000 | Parameter, Integer | Up to 64 objects per lane |
| Relay Simulation | Trigger output simulation | 303 | 0 | 2000 | Parameter, Integer | 0: simulation off<br>1: submode 1 (individual trigger setting)<br>2: submode 2 (running trigger, in turn) |
| active_relays_part1 | Defines active relays (bit-coded) for relay simulation 1 (relay1…32) | 319 | 1 | 2000 | Parameter, Integer | Default: 1 |
| active_relays_part2 | Defines active relays (bit-coded) for relay simulation 1 (relay33…64) | 320 | 0 | 2000 | Parameter, Integer | Default: 0 |
| Number of Simulated Relays | Defines number of simulated relays for relay simulation 2 | 321 | 8 | 2000 | Parameter, Integer | 0 to 64 |

| | | | | | | |
|---|---|---|---|---|---|---|
| radar_detection_filter_active | Activates the RDF. The detection target with the highest amplitude within the programmed speed interval is copied into the object output list. **The sensor will not send tracked objects of real traffic with activated RDF!** | 450 | 0 | 2000 | Parameter, Integer | 0 = off<br>1 = on |
| rdf_speed_range_1_lower_limit | Lower speed limit [m/s] (Filter 1) | 451 | -11 | 2000 | IEEE Float | -100m/s … +100m/s |
| rdf_speed_range_1_upper_limit | Upper speed limit [m/s] (Filter 1) | 452 | -9 | 2000 | IEEE Float | -100m/s … +100m/s |
| rdf_speed_range_2_lower_limit | Lower speed limit [m/s] (Filter 2) | 453 | -21 | 2000 | IEEE Float | -100m/s … +100m/s |
| rdf_speed_range_2_upper_limit | Upper speed limit [m/s] (Filter 2) | 454 | -19 | 2000 | IEEE Float | -100m/s … +100m/s |
| rdf_speed_range_3_lower_limit | Lower speed limit [m/s] (Filter 3) | 455 | -31 | 2000 | IEEE Float | -100m/s … +100m/s |
| rdf_speed_range_3_upper_limit | Upper speed limit [m/s] (Filter 3) | 456 | -29 | 2000 | IEEE Float | -100m/s … +100m/s |
| rdf_speed_range_4_lower_limit | Lower speed limit [m/s] (Filter 4) | 457 | -41 | 2000 | IEEE Float | -100m/s … +100m/s |
| rdf_speed_range_4_upper_limit | Upper speed limit [m/s] (Filter 4) | 458 | -39 | 2000 | IEEE Float | -100m/s … +100m/s |
| rdf_speed_range_5_lower_limit | Lower speed limit [m/s] (Filter 5) | 459 | -51 | 2000 | IEEE Float | -100m/s … +100m/s |
| rdf_speed_range_5_upper_limit | Upper speed limit [m/s] (Filter 5) | 460 | -49 | 2000 | IEEE Float | -100m/s … +100m/s |
| orient_det_pitch_threshold | This is to set maximum value to detect fault in orientation pitch | 461 | 0 | 2000 | Parameter, Integer | 0 … 5000 |
| orient_det_roll_threshold | This is to set maximum value to detect fault in orientation roll | 462 | 10 | 2000 | Parameter, Integer | 0 … 5000 |

In order to set the network parameters for the sensor, it is necessary to set the sensor to "Configuration mode" via the following command instruction:

Network instructions (Table 4-2):

| Name | Description | ID | Default | Section | Type | Typical Values |
|---|---|---|---|---|---|---|
| Config Mode | switches sensor to Config Operational Mode | 12 | 1 | 2020 | Command, Integer | 1 |

| Name | Description | ID | Default | Section | Type | Typical Values |
|---|---|---|---|---|---|---|
| IP Adress byte 0 | First Byte of IP Address | 0 | 192 | 2021 | Parameter, Integer | 192 |
| IP Adress byte 1 | Second Byte of IP Address | 1 | 168 | 2021 | Parameter, Integer | 168 |
| IP Adress byte 2 | Third Byte of IP Address | 2 | 11 | 2021 | Parameter, Integer | 11 |
| IP Adress byte 3 | Fourth Byte of IP Address | 3 | 11 | 2021 | Parameter, Integer | 11 |
| Subnet Mask Byte 0 | First Byte of SubNet Mask | 4 | 255 | 2021 | Parameter, Integer | 255 |
| Subnet Mask Byte 1 | Second Byte of SubNet Mask | 5 | 255 | 2021 | Parameter, Integer | 255 |
| Subnet Mask Byte 2 | Third Byte of SubNet Mask | 6 | 255 | 2021 | Parameter, Integer | 255 |
| Subnet Mask Byte 3 | Fourth Byte of SubNet Mask | 7 | 0 | 2021 | Parameter, Integer | 0 |
| Broadcast Byte 0 | First Byte of Broadcast | 8 | 192 | 2021 | Parameter, Integer | 192 |
| Broadcast Byte 1 | Second Byte of Broadcast | 9 | 168 | 2021 | Parameter, Integer | 168 |
| Broadcast Byte 2 | Third Byte of Broadcast | 10 | 11 | 2021 | Parameter, Integer | 11 |

| | | | | | | |
|---|---|---|---|---|---|---|
| Broadcast Byte 3 | Fourth Byte of Broadcast | 11 | 255 | 2021 | Parameter, Integer | 255 |
| Default Gateway Byte 0 | First Byte of Default Gateway | 12 | 0 | 2021 | Parameter, Integer | |
| Default Gateway Byte 1 | Second Byte of Default Gateway | 13 | 0 | 2021 | Parameter, Integer | |
| Default Gateway Byte 2 | Third Byte of Default Gateway | 14 | 0 | 2021 | Parameter, Integer | |
| Default Gateway Byte 3 | Fourth Byte of Default Gateway | 15 | 0 | 2021 | Parameter, Integer | |
| Destination Client ID | Destination Client ID (target) | 16 | 0x1000001 | 2021 | Parameter, Integer | |

Please note, that it is important that all network parameters will be set with valid values.

Now, please switch back to the "normal mode" with the following command:

| Name | Description | ID | Default | Section | Type | Typical Values |
|---|---|---|---|---|---|---|
| Normal Mode | switches sensor to Normal Operational Mode | 11 | 1 | 2020 | Command, Integer | 1 |

The sensor will now use the new network setup parameters after the reboot.

Tracking parameters (Table 4-3):

| Name | Description | ID | Default | Section | Types | Typical Values |
|---|---|---|---|---|---|---|
| Execute Tracking[9] | Execute tracking | 0 | 1 | 2004 | Parameter, Integer | 0 = deactivated<br>1 = activated |
| Output Position on Spline | Object always moves on a spline | 127 | 1 | 2004 | Parameter, Integer | 0 = deactivated<br>1 = activated |
| Convert Class into Length | Class is coded in object length | 60 | 2 | 2004 | Parameter, Integer | 0 = only length estimation<br>1 = only class |

[9] Suggested to experienced users only.

| | | | | | | |
|---|---|---|---|---|---|---|
| | | | | | | 2 = class & length estimation |
| Max_hold_time | Maximum time to hold an object | 40 | 180 | 2004 | Parameter, IEEE Float | 0 to 65535s |
| Min_age_for_hold | Minimum age of cycles to hold an object | 41 | 35 | 2004 | Parameter, IEEE Float | 0 to 65535 cycles |

Mounting parameters (Table 4-4):

| Name | Description | UAT Version | ID | Default | Section | Type | Typial Valus |
|---|---|---|---|---|---|---|---|
| xPos | X-position of the sensor in the Cartesian coordinate system | 4 | 50 | 0.0 | 2000 | Parameter, IEEE Float | -300.0 to +300.0m |
| yPos | Y-position of the sensor in the Cartesian coordinate system | 4 | 51 | 0.0 | 2000 | Parameter, IEEE Float | -300.0 to +300.0m |
| zPos | Z-position of the sensor in the Cartesian coordinate system | 4 | 52 | 5.0 | 2000 | Parameter, IEEE Float | -20 to +20m |
| xy Orientation | Azimuth angle | 4 | 53 | 0.0 | 2000 | Parameter, IEEE Float | -180.0 to 180.0° |
| xz Orientation | Elevation angle | 4 | 54 | 0.0 | 2000 | Parameter, IEEE Float | -90.0 to 90.0° |
| yz Orientation | Turned upside down | 4 | 55 | 0.0 | 2000 | Parameter, IEEE Float | -180 to 180° 0°=unturned 180°=turned |

## 4.3 SPECIAL INSTRUCTIONS

The following command instructions are available for the sensor:

| Name | Description | ID | Section | Type | Typical Values |
|---|---|---|---|---|---|
| Factory Reset | Sensor reset to default configuration | 2 | 2020 | Command, Integer | 1 |
| Sensor Reset | Restart without delay | 3 | 2020 | Command, Integer | 1 |
| UNIX Time | set the unix time [s] | 8 | 2020 | Command, Integer | Unix time [s] |
| Reset TM Statistics | reset tm statistic module | 10 | 2020 | Command, Integer | 1 |
| tm_fsm_core 0_opmode | switches sensor to Normal Operational Mode or Configuration Mode | 11 | 2020 | Command, Integer | 1 = Normal,<br>2 = Configuration |

## 4.4 POLYGON FEATURE

This section describes the parameter setup for the polygon feature without using the TUI. Please also see the section describing how to use the polygon feature.

Polygon parameters are UAT V4 formatted arrays. The array dimensions are defined as follows:

- i1 = polygon = polygon number [0<=i1<= 15]
- i2 = corner = corner number [0<=i2<=7]

If the array has no dimensions (none), i1 and i2 are set to value 0.

General polygon settings (Table 4-5):

| Name | Description | ID | Dimen-sions | Default | Section | Types | Typical Values |
|---|---|---|---|---|---|---|---|
| Active Polygon | Bit field for active polygons | 90 | none | 1 | 2004 | Parameter, Integer | 0 to 16 |
| usage _default _track | Usage of settings for tracks outside of all polygons | 91 | none | 0 | 2004 | Parameter, Integer | bit field:<br>0x0000001: delete_tracks<br>0x0000002: dont_transmit_tracks<br>0x0000004: deactivate_tracks |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | | | | | | 0x0000008: slow_moving_track[10]<br>0x0000010: can_hold<br>0x0000020: not used<br>0x0000040: use_apriori_hypo _axes<br>0x0000080: use_apriori_spline<br>0x0000100:hypo_track_spacing_active |
| usage _default _target | Usage of settings for target outside of all polygons | 92 | none | 0 | 2004 | Parameter, Integer | bit field:<br>0x00001: ignore_targets<br>0x00002: dont_init_tracks<br>0x00004: dont_associate_with_track<br>0x00008: stat_tgt_can_init_track<br>0x00010: hypo_0_inactive<br>0x00020: hypo_1_inactive<br>0x00040: hypo_2_inactive<br>0x00080: hypo_3_inactive<br>0x00100: no_init_tracks_hypo_axes<br>0x00200: no_init_tracks_spline<br>0x00400: stat_tgt_can_update_track<br>0x00800: not used<br>0x01000: not used<br>0x02000: init_tracks_radial |

Polygon settings (Table 4-6):

| Name | Description | ID | Dimensions | Default | Section | Types | Typical Values |
|---|---|---|---|---|---|---|---|
| Number of lines | Number of lines or points | 0 | Polygon | 4 | 2009 | Parameter, IEEE Float | 0 to 8 |
| Priority | Priority of the polygon | 1 | Polygon | 1 | 2009 | Parameter, IEEE Float | 1 to 16 |
| vel_x_min | Minimum speed limit in x direction (relative or absolute, depending on control flags) | 2 | Polygon | -83.33 | 2009 | Parameter, IEEE Float | -88.0 to +88.0 m/s |
| vel_x _max | Maximum speed limit in x direction (relative or abso- | 3 | Polygon | 83.33 | 2009 | Parameter, IEEE Float | -88.0 to +88.0 m/s |

[10] Suggested to experienced users only.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | lute, depending on control flags) | | | | | | |
| Vel_y _min | Minimum speed limit in y direction (relative or absolute, depending on control flags) | 4 | Polygon | -50.00 | 2009 | Parameter, IEEE Float | -88.0 to +88.0 m/s |
| Vel_y _max | Maximum speed limit in y direction (relative or absolute, depending on control flags) | 5 | Polygon | 50.0 | 2009 | Parameter, IEEE Float | -88.0 to +88.0 m/s |
| Usage Track Inside Speed Limit | Bit-coded usage of track inside the speed limit | 7 | Polygon | 400 | 2009 | Parameter, IEEE Float | bit field:<br>0x0000001: delete_tracks<br>0x0000002: dont_transmit _tracks<br>0x0000004: deactivate_tracks<br>0x0000008: slow_moving _track[11]<br>0x0000010: can_hold<br>0x0000020: not used<br>0x0000040: use_apriori _hypo_axes<br>0x0000080: use_apriori_spline<br>0x0000100: hypo_track _spacing_active |
| Usage Track Outside Speed Limit | Bit-coded usage of track outside the speed limit | 8 | Polygon | 405 | 2009 | Parameter, IEEE Float | bit field:<br>0x0000001: delete_tracks<br>0x0000002: dont_transmit _tracks<br>0x0000004: deactivate_tracks<br>0x0000008: slow_moving _track[11]<br>0x0000010: can_hold<br>0x0000020: not used<br>0x0000040: use_apriori_hypo _axes<br>0x0000080: use_apriori_spline<br>0x0000100: hypo_track _spacing_active |
| Usage target | Bit-coded usage of settings for targets | 9 | Polygon | 480 | 2009 | Parameter, IEEE Float | bit field:<br>0x00001: ignore_targets<br>0x00002: dont_init_tracks |

[11] Suggested to experienced users only.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | | | | | | 0x00004: dont_associate<br>_with_track<br>0x00008: stat_tgt_can_init<br>_track<br>0x00010: hypo_0_inactive<br>0x00020: hypo_1_inactive<br>0x00040: hypo_2_inactive<br>0x00080: hypo_3_inactive<br>0x00100: no_init_tracks_hypo<br>_axes<br>0x00200: no_init_tracks_spline<br>0x00400: stat_tgt_can_update<br>_track<br>0x00800: not used<br>0x01000: not used<br>0x02000: init_tracks_radial |

Polygon parameters (Table 4-7):

| Name | Description | ID | Dimensions | Section | Types | Typical Values |
|---|---|---|---|---|---|---|
| Corner X-Position of Polygon | Unit[m] | 0 | Polygon, corner | 2007 | Parameter, IEEE Float | Interval [2046 to -2046] |
| Corner Y-Position of Polygon | Unit[m] | 1 | Polygon, corner | 2007 | Parameter, IEEE Float | Interval [2046 to -2046] |

Polygon status (Table 4-8):

| Name | Description | ID | Dimen-sions | Section | Types | Typical Values |
|---|---|---|---|---|---|---|
| Polygon Version | Version of the polygon feature | 11 | none | 2008 | Status, Integer | |
| max_nof _polygons | Maximum number of possible polygons | 12 | none | 2008 | Status, Integer | |

| | | | | | | |
|---|---|---|---|---|---|---|
| max_nof _lines | Maximum number of possible lines or points for each polygon | 13 | none | 2008 | Status, Integer | |
| Polygon_0_7_status | Polygon status of the polygons 0 to 7 | 14 | none | 2008 | Status, Integer | bit field (4 bits for each polygon):<br>0x0=inactive<br>0x1=invalid<br>0x2=temporarily inactive 0xF=active |
| Polygon_8_15_status | Polygon status of the polygons 8 to 15 | 15 | none | 2008 | Status, Integer | bit field (4 bits for each polygon):<br>0x0=inactive<br>0x1=invalid<br>0x2=temporarily inactive 0xF=active |

Reset hardware or software by command (Table 4-9):

| Name | ID | Default | Section | Types | Typical Values |
|---|---|---|---|---|---|
| Error Diag. Condition | 322 | 17 | 2000 | Parameter, Integer | 0..31 |

## 4.5 PARAMETERS FOR TRACK AND TARGET HANDLING

The settings parameters define the behavior of each polygon, or the area beyond all polygons, regarding raw targets[12] or tracked objects. Each parameter is bit coded.

Bit-coded settings for track handling (Table 4-10):

| Bit | Name | Description |
|---|---|---|
| 0 | Delete tracks | Tracks will be completely deleted |
| 1 | Don't transmit tracks | The object tracking will work normally but no tracked object are reported |
| 2 | Deactivate tracks | Track will be deactivated |
| 3 | [Irrelevant] | [Irrelevant] |
| 4 | Tracked objects can hold | Tracks will not be deleted after a full stop |
| 5 | [Irrelevant] | [Irrelevant] |
| 6 | Use apriori heading from hypo axes | Needed for tracking via hypo axes or main direction of movement |
| 7 | Use apriori heading from splines | Needed for tracing via splines |
| 8 | Hypo track spacing active | Keep the default setting |

[12] A raw target is an early pre-stage of a tracked object.

Bit-coded settings for target handling (Table 4-11):

| Bit | | Description |
|---|---|---|
| 0 | Ignore targets | The raw targets will not be used for the tracking |
| 1 | Don't initiate tracks | Targets cannot open new tracks |
| 2 | Don't associate with tracks | Keep the default setting |
| 3 | [Irrelevant] | [Irrelevant] |
| 4 | Hypo 0 inactive | Needed for tracking via hypo axes |
| .. | .. | .. |
| 8 | No init tracks hypo axes | Needed for tracking via splines |
| 9 | No init tracks splines | Needed for tracking via hypo axes |
| 10 | [Irrelevant] | [Irrelevant] |

The default values for a standard polygon vary depending on what kind of movement prediction method should be used for the tracking algorithm:

- For tracking via splines, the movement of an object is predicated using the course of the closest defined lane
- For tracking via hypo(thesis) axes, the movement is predicated using the sensor's zero-degree axis as the main direction of movement.

Default values (Table 4-12):

| Polygon | Tracking Method | Default Value | |
|---|---|---|---|
| | | Decimal | Bits = 1 |
| Track handling beyond all polygons | both | 0 | All bits are 0 |
| Target handling beyond all polygons | both | 1 | Bit 0 |
| Track handling inside defined speed range in a polygon | via splines | 400 | Bit 8, 7, 4 |
| | via hypo axes | 336 | Bit 8, 6, 4 |
| Track handling outside defined speed range in a polygon | via splines | 23 | Bit 4, 2, 1, 0 |
| | via hypo axes | 23 | Bit 4, 2, 1, 0 |
| Target handling in a polygon | via splines | 488 | Bit 8, 7, 6, 5, 3 |
| | via hypo axes | 744 | Bit 9, 7, 6, 5, 3 |

## 4.6 FAIL-SAFE CAPABILITIES [13]

The sensor offers diagnostics for different classes of failures. They are called error diagnostic conditions. Please note that not all possible failures can be detected by the onboard diagnostics, and that the diagnostic features neither have 100% detection rate nor 0% false alarm rate.

The event trigger relay module can be configured to activate all virtual relays of the sensor simultaneously under certain conditions detected by the sensor diagnostic and self-test module. The error diagnostic conditions can be configured by the parameter instruction:

Fail-safe mode parameter (Table 4-13):

| Name | ID | Default | Section | Types | Typical Values |
|---|---|---|---|---|---|
| Error Diag. Condition | 322 | 16 | 2000 | Parameter, Integer | 0..31 |

Possible parameter values (bitwise OR) (Table 4-14):

| Error Diagnostic Condition | Description | Value |
|---|---|---|
| disabled | Error Diagnostic output is not active | 0 |
| precipitation | Falling rain or snow (default) | 1 |
| interference | Interference by another sensor or other radar sensor[14] | 2 |
| reserve | - | 4 |
| reserve | - | 8 |
| sensor blind | Material detected in front of the sensor antenna, that significantly reduces sensor’s detection capabilities. | 16 |

The values are bitwise OR. For example, the default value 9 means that the error diagnostic output is activated for rain and critical errors only.

### RAIN DETECTION

The rain detection algorithm provides a set of parameters that can be adjusted to trigger the rain flag only at certain rain levels and optionally set the relay output to fail-safe state, which means that all relays are activated.

[13] Not available yet

[14] Please note that passing cars equipped with radar sensors may trigger the interference detection without decreasing the radar performance. Therefore, it is not recommended to use the interference feature as error diagnostic condition.

# 5 STATISTICS MODULE

In the following, the Statistics Module V2 is described as a top-level application running on the Traffic Management sensor.

## 5.1 OBJECTIVES OF THE STATISTICS MODULE V2 (SM2)

The Traffic Management sensor generates an output of tracked object lists. That output is versatile but unhandy for the end user. Furthermore, it requires comparatively high data communication bandwidth whereas the user may have low-performance microcontrollers on the backend.

The Statistics Module V2 is the 2nd generation module which is much more flexible and optimized for the customer's requests than the first generation. It is optimized for the latest and upcoming sensor generation structure.

Consequently, the Statistics Module V2 has the following capabilities:

- Executed on the sensor's processor
- Providing the sole communication interface to the user
- Significantly reducing the amount of information transmitted to the user
- Generating results directly exploitable for the user, without any further interpretation
- Supports up to 32 zones which can be freely placed or assigned to lanes
- Supports up to 16 vehicle classes[15]
- Providing statistics results on a per zone and per vehicle class basis with:
    - Volume output
    - Occupancy output
    - Average speed output
    - 85 percentile speed output
    - Headway output
    - Gap output
- Providing event trigger results on a per zone and per vehicle class basis with:
    - Support for multiple event trigger applications which can be set for each zone separately
    - 64 relay outputs which can be freely assigned to all 32 available zones

[15] Note: The firmware may support less classes than the Statistics Module can handle.

## 5.2 SPECIFICATIONS

In the following, the features and the capabilities of the Statistic Module are described.

### 5.2.1 TOP LEVEL CONSIDERATIONS

- The module supports monitoring of 32 zones in total
- The zones can be freely placed or assigned to predefined splines
- For each zone the statistics can be collected
- For each zone an event trigger can be defined
- The module keeps the last time stamp and calculates the time passed between last call and recent call
  - The module sends the following data though its communication interface:
    - Instantaneous or **Event Trigger output**:
      - Presence trigger
      - ETA trigger
      - Speed trigger
      - Queue Length Measurement (QLE) trigger
      - Wrong direction trigger
      - Custom trigger
      - Trigger extension & delay time can be set for each zone separately
      - Single pulse trigger can be set for each zone separately
      - 64 outputs which can be freely assigned to all 32 available zones
    - Every interval period or **Statistics output**:
      - The statistic interval for the module is selectable between 1s and 3600s.
      - It checks every sensor cycle if the configured interval time is reached
      - General statistics information (number of zones, synchronized time, …)
      - Volume counting, vehicle class specific
      - Occupancy
      - Average speed
      - 85 percentile speed
      - Headway
      - Gap
  - Support of up to 16 vehicle classes (depending on the used sensor hardware). Actual pre-defined classes are. The class number correspond to the number sent by the radar:
    - Undefined: class 0

- Pedestrian: 1.0m length, class 1
- Bike: 1.6m, class 2
- Motorbike: 2.6m length, class 3
- Passenger car: 4.6m…5.4m length, class 4
- Delivery/pickup: 5.6m…9.4m length, class 6
- Short truck: 9.0m…13.8m, class 7
- Long truck: 14.0m…25m, class 8

The TRUGRD and TRUGRD Stream sensor family also supports classification by object length which covers the ROSAVTODOR (EUR6-based) and GOST 32965-2014 (EUR7-based) standard.

*Table 15 ROSAVTODOR (EUR6-based)*

| class | Class number | Class length |
|---|---|---|
| Passenger car | 4 | Up to 4,5m |
| Minibus, delivery/pickup | 6 | 5m…8m |
| Short truck | 7 | 8m…11m |
| Bus | 10 | 11m…13.5m |
| Medium Truck | 9 | 13.5m…22m |
| Long truck | 8 | >22m |

*Table 16 GOST 32965-2014 (EUR7-based)*

| class | Class number | Class length |
|---|---|---|
| Motorbike | 3 | 2.5m |
| Passenger car | 4 | 2.6m… 4.5m |
| Minibus, delivery/pickup | 6 | 5m…8m |
| Short truck | 7 | 8m…11m |
| Bus | 10 | 11m…13.5m |
| Medium Truck | 9 | 13.5m…22m |

| | | |
|---|---|---|
| Long truck | 8 | >22m |

Further information for activating the  optinal classification can be found in the corresponding "HowTo_use_Russian_Highway_Classification" documentation.

### 5.2.2 ZONE SPECIFICATIONS

Statistics Module V2 uses up to 32 zones which can be placed freely within the coverage area from the used sensor. A zone can be optionally docked to one or multiple lanes.

A zone will be formed from multiple two-dimensional **segments** (x & y position).

A zone consists of at least two segments, the number of used segments for each zone can be dynamically increased. The heading of a zone will be defined by the used segment points, starting from the first segment to the next one and so on. See chapter 5.3.3 and 0 for the zone and segments parameters.

Examples:

- 32 zones used with 4 segments per zone
- One zone used with 128 segments
- Two zones used with 1x 96 segments and 1x 32 segments

The sensor supports up to 128 segments in total.

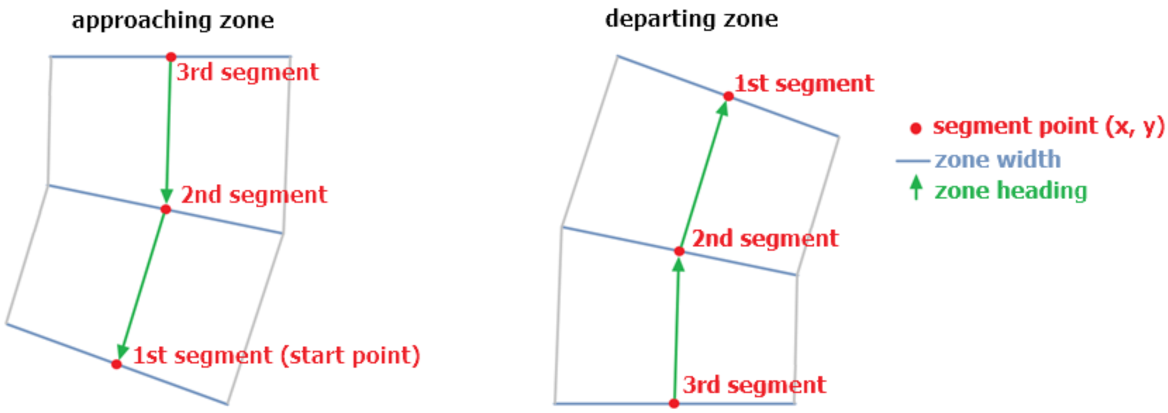



*Figure 5-1: A zone is composed of >1 segment*

A **zone** consists of the following components:

- Zone width
- Number of used segments

- Segments (X & Y positions)
- Used trigger application
- Output assignment
- Used statistics
- Used classes for statistics
- Used classes for trigger application
- Trigger type (presence or single pulse trigger), trigger delay & extension
- Trigger conditions (like ETA offsets, min. & max. speed)

For each zone, the number of used segments can be dynamically adjusted. The minimum number of segments to make a zone is 2. If, for example on a curved road, more than two segments are necessary, additional segments can be added. In total up to 128 segments are available for up to 32 zones.

### 5.2.3 CONFIGURATION DATA

Please make the following settings:

- Number of zones
- Definition of the zones
  - Width, number of segments, segment points
  - For what kind of application, the zone shall be used
- Operational Mode
  - **Statistics output** for a specific zone
    - Volume counting
    - Average speed
    - 85th percentile speed
    - Occupancy
    - Headway
    - Gap
- Used classes for statistics
- Statistics interval time
  - **Trigger application** for a specific zone
    - Presence trigger, speed trigger, ETA trigger, queue length trigger, and wrong direction trigger supported

- Trigger application parameters
- Used classes for event trigger

### 5.2.4 STATISTIC FEATURE SPECIFICATION

The Statistic Module sends its output data for every zone which is configured to be used for statistics gathering. The statistic interval time can be adjusted by a parameter. When the interval time is reached the statistical data will be transmitted by the sensor.

In order to reduce the amount of transmitted data, the module reports the statistics for each zone separately within a cycle (up to 32 cycles if 32 zones are used).

#### 5.2.4.1 Statistic Output: Volume

This application is a class-specific count per zone.

Whenever the zone is occupied by a tracked object at a function call, its track ID will be processed and recorded to an object-class dependent variable. When the track ID is unequal to the recorded track ID, a counter will be incremented. This solution prevents multiple counting of one object.

- Supports count by class
- The resolution of the counter is in second
- The counter will be reset when the statistics interval time is reached
- Data amount: 16 bits per zone

#### 5.2.4.2 Statistic Output: Occupancy

This feature is a counter per time that specifies the occupancy of the zone during the interval time. Whenever the measuring point is occupied by a tracked object at a function call, a counter will be incremented. When the interval time is reached the occupancy will be calculated. The result is a percentage.

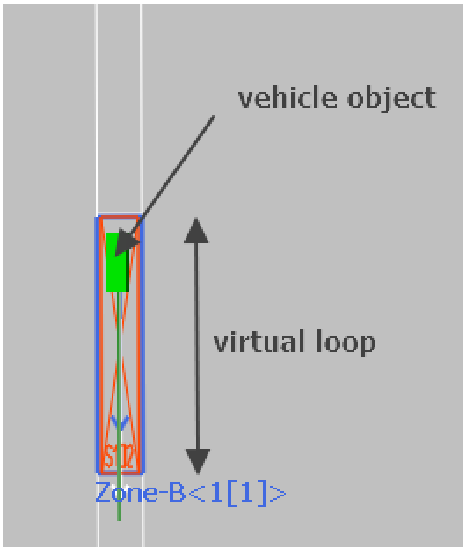
- Supports occupancy calculation by class
- The resolution is [0.05%].
- The output will be reset when the interval time is reached
- Data amount: 11 bits per zone

Calculation of the occupancy follows the formula:

$$Occupancy = \frac{\sum on - time}{\text{report interval}}$$

Explanation and details:

The on-time is calculated as the time that any portion of a vehicle object is inside the virtual loop. It is measured with the increment of the sensor cycle time, i.e., at approximately 56ms. The following example depicts a vehicle object that enters the virtual loop highlighted in red below. As long as there is an overlap of the red vehicle object and the virtual loop, the on-time is incremented.




The report-interval is the report time set to the sensor, e.g., 5 minutes by default.

#### 5.2.4.3 Statistic Output: Average Speed

This feature yields an average speed per zone.

Whenever the zone is occupied by a new tracked object having the same class at a function call, its speed will be added to an object-class dependent variable and a counter will be incremented. When the interval time is reached, the average speed for each zone will be calculated.

- Supports average speed calculation by class
- The resolution is [0.01 {m/s}].
- The output is valid from 0m/s to 89m/s absolute.
- The average speed will be reset when the interval time is reached
- Data amount: 16 bits per zone

#### 5.2.4.4 Statistic Output: 85th Percentile Speed

This feature yields the 85th percentile of speed distribution per zone. It means that 85% of all registered vehicles having the same class within a zone go at the reported speed or less.

Whenever the zone is occupied by a tracked object at a function call, its speed will be added to an object-class dependent history variable and a counter will be incremented.

When the interval time is reached the final 85th percentile speed for each zone will be calculated.

- Supports 85th percentile speed calculation by class
- The resolution is [0.01 {m/s}].
- The output is valid from 0m/s to 89m/s absolute.
- The 85th percentile speed will be reset when the general reset flag is set.
- Data amount: 16 bits per zone

#### 5.2.4.5 Statistic Output: Headway

This feature yields the average passing time between two vehicle's front-to-front in seconds per zone. Whenever the zone is overrun by a tracked object at a function call, a timer is started. When a new object crosses the zone, the elapsed time will be added to a sum time and a counter will be increased. When the interval time is reached the average Headway time for each zone will be calculated.

- The resolution is [0.055s].
- The output is valid from 0.00 s to 3600.00s.
- The average headway time and the previous timer will be reset when the general reset flag is set.
- Data amount: 16 bits per zone
- This output is without classification, it will be sent as class 0 (undefined)

#### 5.2.4.6 Statistic Output: Gap

This feature yields the average passing time between two vehicle's rear-to-front in seconds per zone. Whenever the zone is overrun by the back side of a tracked object at a function call, a timer is started. When a new object crosses the zone, the elapsed time will be added to a sum time and a counter will be increased. When the interval time is reached the average Gap time for each zone will be calculated.

- The resolution is [0.055s].
- The output is valid from 0.00s to 3600.00s.
- The average gap time and the previous timer will be reset when the general reset flag is set.
- Data amount: 16 bits per zone
- This output is without classification, it will be sent as class 0 (undefined)

#### 5.2.4.7 Statistic Output: Per Vehicle Record

If the Per Vehicle Record (PVR) is active, then each counted track will be transmitted with a PVR message immediately within the first cycle when the object enters the zone. This PVR message contains the following information for each registered object:

- Object ID (consistent during the lifetime of the object)
- Zone
- Object class
- Object speed [m/s]
- Optional "exclusive count" can be de/activated if PVR count is needed for each zone

### 5.2.5 EVENT TRIGGER FEATURE SPECIFICATION

The sensor can trigger (virtual or hardware) outputs for applications like

- Presence detection,
- Estimated Time of Arrival (ETA)
- Speed triggering
- Other applications

For each zone the user can select a desired application and change specific parameters which are:

- Trigger application
- Object classes to trigger
- Output assignment for up to 64 outputs
- Minimum/maximum time offset
- Minimum/maximum speed
- ETA: X, Y position
- Direction of traffic
- Presence or single pulse trigger type
- Pulse duration (if single pulse trigger is used)
- Trigger extension
- Trigger delay

#### 5.2.5.1 Event: Presence Trigger

This application triggers an assigned output whenever the predefined detection zone is occupied by one or multiple objects. For this application, the following parameters can be selected:

- Zone extension & width
- Object classes to trigger
- Output assignment

#### 5.2.5.2 Event: ETA Trigger

This application triggers an assigned output whenever the Estimated Time of Arrival (ETA) of one or multiple objects is within the predefined timeframe. For this application the following parameters can be configured:

- Zone extension & width
- Object classes to trigger
- Time offset for ETA calculation (min/max time of the triggering timeframe)
- Output assignment
- Option: X, Y position to which ETA shall be calculated; default X, Y position is the first segment of the zone

#### 5.2.5.3 Event: Speed Trigger

This application triggers an assigned output whenever one or multiple objects in the measuring zone are moving within a predefined speed interval. For this application the following parameters can be selected:

- Zone extension and width
- Object classes to trigger
- Speed interval (min/max speed of the triggering interval)
- Output assignment

#### 5.2.5.4 Event: Wrong Direction Trigger

This application triggers an assigned output whenever the predefined detection zone is occupied by an object that is driving in opposite direction. The used direction of movement will be set within the “zones” section of the Traffic Management Configurator (TUI) software.

Please note that the used lanes should be configures for bi-directional movement otherwise the sensor will only track objects for the configured lane-direction.

For this application the following parameters can be selected:

- Zone extension & width
- Object classes to trigger
- Speed Interval for triggering (min/max speed of the triggering interval)
- Minimum travelled distance for triggering
- Output assignment

#### 5.2.5.5 Event: Queue Application Trigger

This application triggers an assigned output whenever the minimum configured queue length is achieved. You can configure the minimum queue length in the Traffic Management Configurator (TUI). It is also possible to set the maximum allowed speed of the objects to be considered.

- Maximum speed of the objects considered
- Minimum queue length to trigger

#### 5.2.5.6 Event: Custom Trigger

This application is a logic AND-combination of the presence-, speed- and ETA-trigger. It triggers an assigned output whenever all preset requirements (minimum and maximum allowed speed; minimum and maximum allowed time) are fulfilled. You can configure all required parameters within the Traffic Management Configurator (TUI).

- Zone extension & width
- Object classes to trigger
- Speed interval (min/max speed of the triggering interval)
- Time offset for ETA calculation (min/max time of the triggering timeframe)
- Output assignment
- Option: x, y position to which ETA shall be calculated; default x, y position is the first segment of the zone

### 5.2.5.7 Fail-Safe Mode

Statistics Module V2 comes with an integrated failsafe trigger function which will activate all available output relays as long as the detection performance is reduced.

The sensor supports failsafe mode for the following diagnostic functions:

- Heavy rain detection
- Interference detection
- Blind detection

The diagnostic flags will be transmitted within the statistics port header (port-based communication) and within the 0x780 statistic output – Info Message 1 (CAN-based communication).

Each diagnostic function can be activated or deactivated for fail-safe detection. If at least one fail-safe criterion is fulfilled all available output relays will become active. It is also possible to select the desired relays for the fail-safe function.

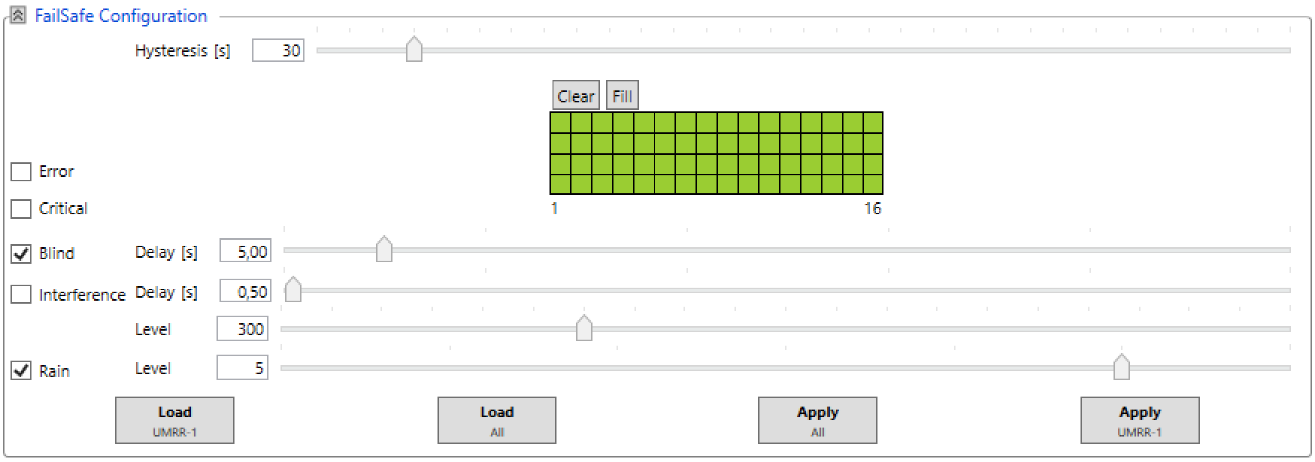



*Figure 3: Failsafe configuration within the TUIConfigurator*

**Important Notice:**

Fail-safe mode and fail-safe features of the sensor are designed to cover as many fail cases as possible during the operation of the sensor. However, not all fail cases are covered: It is important to understand that the detection rate of the sensor is not 100%, that the false alarm rate is not zero, and that not all fail cases will be detected and reported and that some fail cases may be reported in error. The manufacturer disclaims all liability for the operation of the fail-safe mode.

### 5.2.5.8 Fail-Safe Mode: Heavy Rain Detection

In heavy rain or snow conditions it is possible that the general sensor performance is reduced. The included rain detection algorithm can detect these conditions and activate a rain flag if a certain amount of rain targets have been detected. The rain detection function can be adjusted by parameters if desired.

#### 5.2.5.9 Fail-Safe Mode: Interference Detection

If another sensor or device works within the same frequency band of the sensor it is possible that it will interfere with the sensor and reduce the detection performance. The sensor recognizes if interference occurs and activates the interference flag if it occurs.

The Interference fail-safe function can be adjusted by parameters if desired.

#### 5.2.5.10 Fail-Safe Mode: Blind Detection

If the sensor radome is covered with snow or ice, or the field of view is disturbed (e.g., by a wall in front of the unit), the sensor becomes blind or at least the detection performance is reduced.

The sensor analyses stationary and moving targets by different criteria (number of detected targets, signal to noise ratio, etc.). If the quality level is too low, then the blind flag will become active.

The blind fail-safe function can be adjusted by parameters if desired.

## 5.3 STATISTIC MODULE COMMANDS

The Statistic Module contains various useful commands, which are described below.

### 5.3.1 STATISTIC MODUL – STATUS COMMANDS

The following commands must be sent as “status” commands. The sensor will send response messages which will deliver the requested status information.

| Name | Description | Section | ID | Format |
|---|---|---|---|---|
| TM Status Interface major version | Defines the major version of the TM Status Module | 2003 | 0 | Int |
| TM Status Interface minor version | Defines the minor version of the TM Status Module | 2003 | 1 | Int |
| SW generation | TM software version generation | 2003 | 2 | Int |
| SW version major | TM software version major | 2003 | 3 | Int |
| SW version minor | TM software version minor | 2003 | 4 | Int |
| SW version patch | TM software version patch | 2003 | 5 | Int |
| Customer ID | Customer identifier | 2003 | 6 | Int |
| Antenna Type | Antenna Type | 2003 | 7 | Int |
| Region code | Region_code | 2003 | 8 | Int |

| | | | | |
|---|---|---|---|---|
| Product serial | Sensor serial number | 2003 | 9 | Int |
| TM App major | Major version of the app_tm module | 2003 | 14 | Int |
| TM App minor | Minor version of the app_tm module | 2003 | 15 | Int |
| TM App patch | Patch version of the app_tm module | 2003 | 16 | Int |
| TM App Variant | Variant of the app_tm module | 2003 | 18 | Int |
| Roll Angle | Roll angle status | 2003 | 23 | Int |
| Pitch Angle | Pitch angle status | 2003 | 24 | Int |
| Calibration Status | accelerometer indication if calibration is complete | 2003 | 25 | Int |
| Nof Tracking Classes | number of tracking classes | 2003 | 30 | Int |
| Region Code | TM region code:<br>0=EU<br>1=CH<br>2=US | 2003 | 31 | Int |
| TM Application | Traffic Management Application:<br>0=Stopbar+<br>1=Forward+<br>2=Speed Enforcement<br>3=Red-Light Enforcement | 2003 | 32 | Int |

### 5.3.2 STATISTIC MODULE - BASIC PARAMETERS

Commands for basic parameters (Table 5-17):

| Name | description | ID | Section | Type | min | max | default |
|---|---|---|---|---|---|---|---|
| can_active | [1] 0 = Interface Listen (RX only), 1 = Interface Active | 10 | 2000 | u8 | 0 | 1 | 0 |
| rs485_active | [1] 0 = Interface Listen (RX only), 1 = Interface Active | 11 | 2000 | u8 | 0 | 1 | 1 |
| eth_active | [1] 0 = Interface Listen (RX only), 1 = Interface Active | 12 | 2000 | u8 | 0 | 1 | 1 |
| posX | [m] X position of the sensor in Cartesian coordinate system | 50 | 2000 | f32 | -3000 | 3000 | 0 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| posY | [m] Y position of the sensor in Cartesian coordinate system | 51 | 2000 | f32 | -3000 | 3000 | 0 |
| posZ | [m] Z position of the sensor in Cartesian coordinate system | 52 | 2000 | f32 | -200 | 200 | 50 |
| xyOrientation | [deg] Orientation of the sensor in Cartesian coordinate system in X-Y plane (azimuth angle), valid interval (-180.f, 180.f], 0.f for positions on the Z axis | 53 | 2000 | f32 | -1800 | 1800 | 0 |
| xzOrientation | [deg] Orientation of the sensor in Cartesian coordinate system in X-Z plane (elevation angle), valid interval [-90.f, 90.f], 0.f for the origin of the coordinate system | 54 | 2000 | f32 | -900 | 900 | 0 |
| yzOrientation | [deg] Orientation of the sensor in Cartesian coordinate system in Y-Z plane (roll angle) (0.f: not turned, 180.f: turned upside-down), valid interval (0.f, 180.f] | 55 | 2000 | f32 | -1800 | 1800 | 0 |
| output_control_target_list | [1] output data control, 0 = disabled, 1 = enabled | 200 | 2000 | u8 | 0 | 1 | 0 |
| output_control_object_list | [1] output data control, 0 = disabled, 1 = enabled | 201 | 2000 | u8 | 0 | 1 | 1 |
| output_control_triggers | [1] output data control, 0 = disabled, 1 = enabled | 202 | 2000 | u8 | 0 | 1 | 1 |
| output_control_statistics | [1] output data control, 0 = disabled, 1 = enabled | 203 | 2000 | u8 | 0 | 1 | 1 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| output_control_pvr | [1] output data control, 0 = disabled, 1 = enabled | 204 | 2000 | u8 | 0 | 1 | 1 |
| output_control_queue_length | [1] output data control, 0 = disabled, 1 = enabled | 205 | 2000 | u8 | 0 | 1 | 1 |
| nof_zones | number of used measuring zones | 300 | 2000 | u16 | 0 | 32 | 2 |
| use_statistics_features | use statistics features. bit field: 0x0000001: volume_counting, 0x0000002: occupancy, 0x0000004: av_speed | 301 | 2000 | u16 | 0 | 255 | 0 |
| use_trigger_features | activate or deactivate trigger module output | 302 | 2000 | u16 | 0 | 255 | 1 |
| simulation_mode | tm simulation mode. | 303 | 2000 | u16 | 0 | 2 | 0 |
| interval_time | used interval time [s] for calculating statistics | 304 | 2000 | f32 | 0 | 3600 | 300 |
| statistics_flags | used statistic flags. bit field: 0x0000001: speed_scale_unit | 305 | 2000 | u8 | 0 | 1 | 0 |
| exclusive_count | defines if an object should be counted only on the first zone (value 1) or on all zones where the object was detected (value 0) | 306 | 2000 | u8 | 0 | 1 | 0 |
| count_mode | defines counting mode. Value 0 == without classification, 1 == with classification | 307 | 2000 | u8 | 0 | 8 | 1 |
| occupancy_mode | defines occupancy mode. Value 0 == without classification, 1 == with classification | 308 | 2000 | u8 | 0 | 8 | 1 |
| av_speed_mode | defines av. speed mode. Value 0 == without classification, 1 == with classification | 309 | 2000 | u8 | 0 | 8 | 1 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 85th_perc_speed_mode | defines 85th perc. speed mode. Value 0 == without classification, 1 == with classification | 310 | 2000 | u8 | 0 | 8 | 1 |
| nof_simulated_objects_lane | defines number of simulated objects per lane | 311 | 2000 | u8 | 0 | 128 | 2 |
| nof_sim_lanes | defines number of simulated objects | 312 | 2000 | u8 | 0 | 8 | 2 |
| sim_lane_width | defines sim. lane width [m] | 313 | 2000 | f32 | 0 | 100 | 5 |
| sim_lane_length | defines sim. lane length [m] | 314 | 2000 | f32 | 0 | 500 | 100 |
| sim_speed | defines sim. object speed [m/s] | 315 | 2000 | f32 | 0 | 90 | 13889 |
| sim_heading | defines heading [deg] for simulated objects | 316 | 2000 | f32 | -360 | 360 | 360 |
| start_sim_x_pos | defines starting x pos [m] for simulation | 317 | 2000 | f32 | -500 | 500 | 100 |
| start_sim_y_pos | defines starting y pos for simulation | 318 | 2000 | f32 | -200 | 200 | 0 |
| active_relays_part1 | defines active relays (bit-coded) for relay simulation 1 | 319 | 2000 | u32 | 0 | 0xffffffff | 1 |
| active_relays_part2 | defines active relays (bit-coded) for relay simulation 1 | 320 | 2000 | u32 | 0 | 0xffffffff | 0 |
| nof_simulated_relays | defines number of used relays for simulation 2 | 321 | 2000 | u8 | 0 | 64 | 8 |
| failsafe_mode | activate failsafe mode. Bit0: rain failsafe, Bit1: interference, Bit2:general,Bit3:critical, bit4:blind | 322 | 2000 | u8 | 0 | 255 | 16 |
| reserve2 | defines number of used relays for simulation | 323 | 2000 | u16 | 0 | 255 | 0 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| failsafe_relays_pt1 | defines which relays (0...31) should be active in failsafe mode | 324 | 2000 | u32 | 0 | 0xffffffff | 0xffffffff |
| failsafe_relays_pt2 | defines which relays (32...63) should be active in failsafe mode | 325 | 2000 | u32 | 0 | 0xffffffff | 0xffffffff |
| trigger_output | defines trigger output mode, 0 = no output, 1= trigger header output, 2 = to be defined | 326 | 2000 | u8 | 0 | 255 | 1 |
| statistics_output | defines statistics output mode, 0 = no output, 1 = statistics header output, 2 = header + targets | 327 | 2000 | u8 | 0 | 255 | 2 |
| pvr_output | defines PVR output mode 0 = no output, 1 = PVR header output, 2 = PVR header + targets | 328 | 2000 | u8 | 0 | 255 | 2 |
| synchronize_statistics | synchronized statistics output | 329 | 2000 | u8 | 0 | 1 | 0 |
| interference_threshold | interference_threshold threshold | 330 | 2000 | u32 | 0 | 0xffffffff | 300 |
| failsafe_hysteresis | Failsafe hysteresis timer [s] | 331 | 2000 | u16 | 0 | 300 | 30 |
| interference_delay_cycles | interference delay cycles for Failsafe activation[s] | 332 | 2000 | u16 | 0 | 10000 | 10 |
| blind_delay_cycles | hysteresis timer for Failsafe activation[s] | 333 | 2000 | u16 | 0 | 10000 | 100 |
| queueDistTwoObjects | the maximum distance between two cars in a queue | 334 | 2000 | f32 | 0 | 300 | |
| queueDistFirstObject | the maximum distance for the first car in the queue from the beginning of the zone | 335 | 2000 | f32 | 0 | 200 | |
| polling_mode | statistics polling mode. Value 0 == off, 1== send | 336 | 2000 | u8 | 0 | 2 | 1 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | currently collected statistics. | | | | | | |
| 85th_perc_classes | number of used 85th perc speed classes | 337 | 2000 | u8 | 1 | 9 | 1 |

### 5.3.3 STATISTIC MODULE - ZONE COMMANDS

For each zone the following commands can be set separately.

In order to set a command for a specific zone it is necessary to set the zone number (0…31) within the 1st dimension ("i0"):

| name | description | ID | Section | dim0 | type | min | max | default |
|---|---|---|---|---|---|---|---|---|
| used_segments | number of used segments for the used zone | 0 | 2002 | 0..31 | u8 | 2 | 192 | 2 |
| trigger_application | used trigger application, bit-coded | 1 | 2002 | 0..31 | u8 | 0 | 255 | 1 |
| relay_assignment | defines which relay ## should be used for zone ## | 2 | 2002 | 0..31 | u8 | 0 | 255 | 0 |
| trigger_flags | defines additional trigger flags | 3 | 2002 | 0..31 | u8 | 0 | 255 | 0 |
| zone_width | width [m] for the used measuring zone. | 4 | 2002 | 0..31 | f32 | 0 | 1000 | 4 |
| eta_x_point | ETA range offset [m] for the used measuring zone. | 5 | 2002 | 0..31 | f32 | -1000 | 1000 | 0 |
| eta_y_point | ETA range offset [m] for the used measuring zone. | 6 | 2002 | 0..31 | f32 | -1000 | 1000 | 0 |
| time_min_offset | min. time [s] offset for ETA calculation. | 7 | 2002 | 0..31 | f32 | 0 | 1000 | 0 |
| time_max_offset | max. time [s] offset for ETA calculation. | 8 | 2002 | 0..31 | f32 | 0 | 1000 | 0 |
| speed_min_offset | min. speed [m/s] for speed trigger. | 9 | 2002 | 0..31 | f32 | 0 | 90 | 0 |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| speed_max_offset | max. speed [m/s] for speed trigger. | 10 | 2002 | 0..31 | f32 | 0 | 90 | 90 |
| used_trigger_classes | used classes for zone ## | 11 | 2002 | 0..31 | u16 | 0 | 479 | |
| used_statistics_classes | used classes for zone ## | 12 | 2002 | 0..31 | u16 | 0 | 479 | |
| statistics_features | active statistics features, bit-coded | 13 | 2002 | 0..31 | u8 | 0 | 255 | 255 |
| trigger_type | trigger type: 0 == default presence trigger; 1 == single pulse trigger | 14 | 2002 | 0..31 | u8 | 0 | 255 | 0 |
| pulse_duration | pulse duration in [ms] | 15 | 2002 | 0..31 | u16 | 0 | 32767 | 0 |
| trigger_extension | trigger extension in [ms] | 16 | 2002 | 0..31 | u16 | 0 | 32767 | 0 |
| trigger_delay | trigger delay in [ms] | 17 | 2002 | 0..31 | u16 | 0 | 32767 | 0 |
| min_mileage | minimum mileage [m] for wrong direction trigger | 18 | 2002 | 0..31 | f32 | 0 | 500 | 10 |
| max_heading | max. tolerable heading between zone position and object | 19 | 2002 | 0..31 | f32 | 0 | 360 | 30 |
| queueMinTriggerBorder | queue min trigger border | 20 | 2002 | 0..31 | f32 | 0 | 2550 | 300 |

### 5.3.4 STATISTIC MODULE - SEGMENTS COMMANDS

For each segment of a zone, the following commands can be set separately. In total 128 segments are available which can be used for up to 32 zones. At least two segments are necessary for a zone.

In order to set a command for a specific segment it is necessary to set the segment number (0…127) within the 1st first dimension ("i0"):

| name | description | ID | Section | dim0 | type | min | max | default |
|---|---|---|---|---|---|---|---|---|
| pos_x | x position [m] for the zone segment. | 0 | 2001 | 0…127 | f32 | -1000 | 1000 | 30 |
| pos_y | y start position [m] for the zone segment. | 1 | 2001 | 0…127 | f32 | -1000 | 1000 | 0 |

# 6 HOW TO'S

This chapter shows further capabilities and settings, which can be configured for the radar sensor.

## 6.1 CAPABILITIES OF SIMULATION MODES

The radar sensor has different simulation modes for sending synthetic simulated traffic objects or triggers. These modes can be activated and configured using the command view of the TUI software. In the following, the relevant simulation modes are described.

Simulation modes (Table 6-1):

| Simulation Modes | Function | Benefits | Availability |
|---|---|---|---|
| 1 | Generates targets and objects on lanes | Simulates targets, objects, speeds and driving directions; tests trigger zones and statistic zones | No |
| 2 | Generates objects on lanes | Simulates objects on lanes; tests trigger zones and statistic zones | Yes |
| 3 | Radar Cube simulator | Verifies the function of the entire signal processing chain | Yes |
| 4 | Trigger outputs | Verifies the reception of digital triggers at the intersection controller, PLC, etc. | Yes |

### SIMULATION MODE 2: OBJECT SIMULATION

Simulation Mode 2 generates objects with different speeds per lane and random classes depending on the number of configured lanes.

This simulation is independent from the statistics module and the trigger module. You can adjust it to simulate straight lanes or for using the defined tracking lanes/splines. It provides up to 64 lanes, where the speed, direction, and class for the first eight lanes can be chosen freely. The $9^{th}$ lane and above will copy the setup of the $8^{th}$ lane.

If the mode is set to “object simulation on lanes”, a number of objects will be simulated on “invisible” lanes. They appear as close as possible along the boresight axis of the radar sensor. With a rising number of simulated lanes, the $1^{st}$ lane will move to the right side, from the sensor’s point of view.

With “object simulation on splines” objects will be simulated on the lanes defined with the TUI Installation Wizard. The defined number of simulated lanes has no effect in this case.

Simulation mode 2 commands (Table 6-2):

| Name | Description | ID | Default | Section | Types | Typical Values |
|---|---|---|---|---|---|---|
| Object Simulation | Activates and deactivates the simulation | 62 | 0 | 2004 | Parameter, Integer | Object simulation:<br>0 = none<br><1 = on lanes<br>2 = on splines |
| sim_nof_lanes | Number of lanes for lane simulation mode | 63 | 4 | 2004 | Parameter, Integer | 0…64 |
| sim_objects_per_lane | Number of simulated objects per lane/spline | 64 | 5 | 2004 | Parameter, Integer | 0…64 |
| sim_obj_speed_laneX | Speed of simulated objects on lane/ spline X in m/s | 11= lane 0<br>12= lane 1<br>…<br>18= lane 7 | various | 2004 | Parameter, IEEE Float | >0 = in spline direction<br><0 = in opposite spline direction |
| sim_obj_class_laneX | class of simulated object on lane/ spline X | 19= lane 0<br>20= lane 1<br>…<br>26= lane 7 | various | 2004 | Parameter, Integer | 0…8 |

| | | | | | | |
|---|---|---|---|---|---|---|
| Convert_classes_into_vehicle_length | Object length into the length or length by class | 60 | 2 | 2004 | Parameter, Integer | 2 = default<br>3 = EUR6-based classification<br>4 = EUR7-based classification |

| Name | Description | ID | Default | Section | Types | Typical Values |
|---|---|---|---|---|---|---|
| tm_operation_mode | Activates and deactivates the signal simulation | 353 | 0 | 2000 | Parameter, Integer | 0 = deactivated<br>3 = Radar Cube Simulation |
| tm_rcsim_movingTarget | Selects between the simulation of a moving or a stationary target | 355 | 0 | 2000 | Parameter, Integer | 0 = stationary<br>1 = moving |
| tm_rcsim_statTgtPos | Position of the target in the stationary target mode | 356 | 30.0 | 2000 | Parameter, IEEE Float | 0.0 to 200.0 |
| tm_rcsim_movTgtBorder1 | First border for the moving target mode (target moves between border 1 and 2) | 357 | 5.0 | 2000 | Parameter, IEEE Float | 0.0 to 200.0 |
| tm_rcsim_movTgtBorder2 | Second border for the moving target mode (target moves between border 1 and 2) | 358 | 100.0 | 2000 | Parameter, IEEE Float | 0.0 to 200.0 |
| tm_rcsim_tgtSpeed | Speed of the simulated target | 359 | 5.0 | 2000 | Parameter, IEEE Float | -70.0 to 70.0 |
| tm_rcsim_tgtLevel[16] | Signal level of the simulated target | 360 | 30.0 | 2000 | Parameter, IEEE Float | 0.0 to 50.0 |
| tm_rcsim_noiseLevel[17] | Level of the simulated noise | 361 | -10.0 | 2000 | Parameter, IEEE Float | -30.0 to 30.0 |

## SIMULATION MODE 4: TRIGGER OUTPUT SIMULATION

[16] Suggested to experienced users only.
[17] Suggested to experienced users only.

The trigger output simulation mode, also called relay simulation, is only available with the statistics module and the event trigger module of the radar sensor.

Simulation Mode 4 generates triggered outputs, for example, digital binary signals to test the reception of the intersection controller, Programmable Logic Controller (PLC), or another connected host.

The trigger output simulation has two sub modes:

1) For setting individual triggers
2) For an automated scheme in turn, with, for example, triggers running from relay 1 to 32 and from 32 back to 1

Trigger simulation parameters (Table 6-3):

| Name | Description | ID | Section UAT V4 | Types | Typical Values |
|---|---|---|---|---|---|
| Relay Simulation | Trigger output simulation | 303 | 2000 | Parameter, Integer | 0: simulation off<br>1: sub mode 1 (individual trigger setting)<br>2: sub mode 2 (running trigger) |
| active_relays_part1 | Defines active relays (bit-coded) for relay simulation 1 (relay1…32) | 319 | 2000 | Parameter, Integer | Default: 1 |
| active_relays_part2 | Defines active relays (bit-coded) for relay simulation 1 (relay33…64) | 320 | 2000 | Parameter, Integer | Default: 0 |
| Number of simulated Relays | Defines the number of simulated relays for relay simulation 2 | 321 | 2000 | Parameter, Integer | Default: 8 |

The parameter, for example, for automated scheme of trigger output simulation, should be entered as follows:

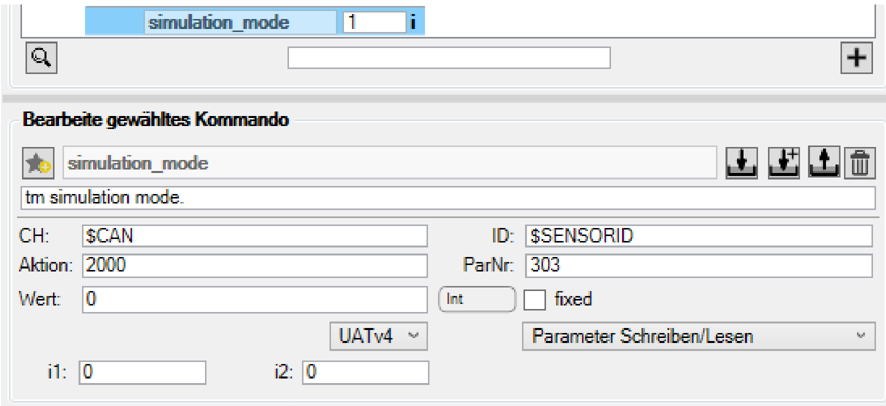



*Figure 6-1: Command example*

## 6.2 CHANGING THE FREQUENCY BAND

The frequency band can be defined via the firmware. Especially, if multiple radar sensors are used, special care is necessary to avoid influences of interference among the sensors. The available frequency bands listed below depend on the configuration. The allowances to use a respective transmit power level and frequency band depend on the region of application.

In the table below, frequency approvals for the regions are labelled as follows:

- A1: EU (incl. Belgium, Latvia), Norway, Iceland, Switzerland, Korea, Japan

Frequencies for 76GHz sensors (Table 6-4):

| Waveform Index | Frequency | Approval for Region |
|---|---|---|
| 0 | 76.032 - 76.249 GHz[18] | A1 |
| 1 | 76.282 - 76.499 GHz[18] | A1 |
| 2 | 76.532 - 76.748 GHz[18] | A1 |
| 3 | 76.782 - 76.999 GHz[18] | A1 |
| 4 | 76.074 - 76.657 GHz[19] | A1 |
| 5 | 76.314 - 76.897 GHz[19] | A1 |

To limit the effects of interference on two sensors facing each other, different frequency IDs should be

[18] Waveform configuration for maximum range
[19] Waveform configuration for high range resolution

used for every sensor on the same Pulse Repetition Frequency (PRF) set ID. For a second pair of sensors facing each other, the PRF is automatically incremented by the TUI.

Parameters for frequency changes (Table 6-5):

| Description | UAT V4 Section | ID | Value | Default | UAT | Types | Typical Values |
|---|---|---|---|---|---|---|---|
| Waveform Index[20] [21] | 2000 | 100 | 0 … 7 | 0 | UAT V4 | Parameter, Integer | 0[22], 1, 2, 3, 4, 5, 6, 7 |
| PRF Set ID | 2000 | 101 | 0 … 1 | 0 | UAT V4 | Parameter, Integer | 0 or 1 |

| Waveform Index | Waveform | Frequency | Modus | Center-Frequency/Bandwidth | Approval for Region |
|---|---|---|---|---|---|
| 0 | 0 | 76GHz | medium range | 76.365GHz/450MHz | A1 |
| 1 | 0 | 79GHz | medium range | 79.365GHz/450MHz | A1 |
| 2 | 1 | 76GHz | long range | 76.365GHz/166MHz | A1 |
| 3 | 1 | 79GHz | long range | 79.365GHz/166MHz | A1 |
| 4 | 2 | 76GHz | extra long range | 76.365GHz/125MHz | A1 |
| 5 | 2 | 79GHz | extra long range | 79.365GHz/125MHZ | A1 |
| 6 | 3 | 76GHz | extra medium range | 76.365GHz/300MHz | A1 |
| 7 | 3 | 79GHz | extra medium range | 79.365GHz/300MHz | A1 |

[20] Suggested to experienced users only.
[21] Please note that not all waveforms/frequency bands are available for all regions or software configurations.
[22] To be sure, please check the status parameter.

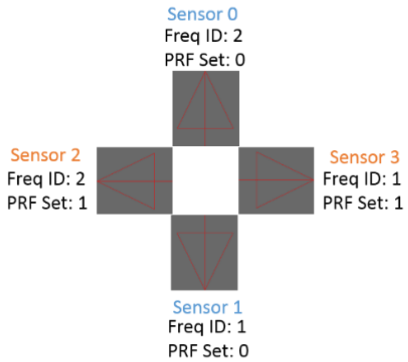



Figure 6-2 Parameter setup example for four sensors

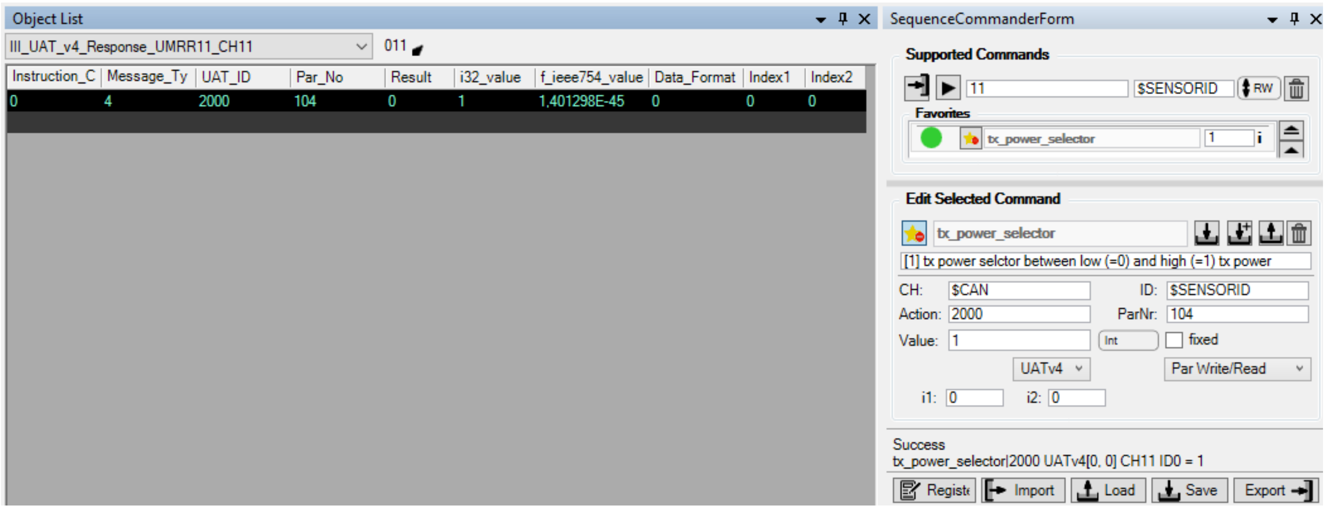



*Figure 6-3 Command example to check the acceptation of frequency and TX-power*

If the result value (i32_value) is 1, the set TX power seems valid. If the value is set to 0, the TX power and the frequency band combination is wrongful. In this example, the TX power was set to 0dBm by the sensor.

## 6.3 READING AND CHANGING SENSOR PARAMETERS

Please note that this section is for expert users only.

The sensor parameters can easily be read and changed with the sequence commander tool of the TUI software. To open the sequence commander window, please select "Views / CAN Data Views / sequence commander".

In order to read the current value of a parameter, please enter the parameter number (ParNr. = ID) and the action number (Section) into the command window. Please check if UAT V4 is selected, choose "read only" from the drop-down list, and enter the number format (Int or Float) of the parameter. By clicking "Send", the "ReadParameter" view in the "SensorTargetList" window will show the parameter as follows.

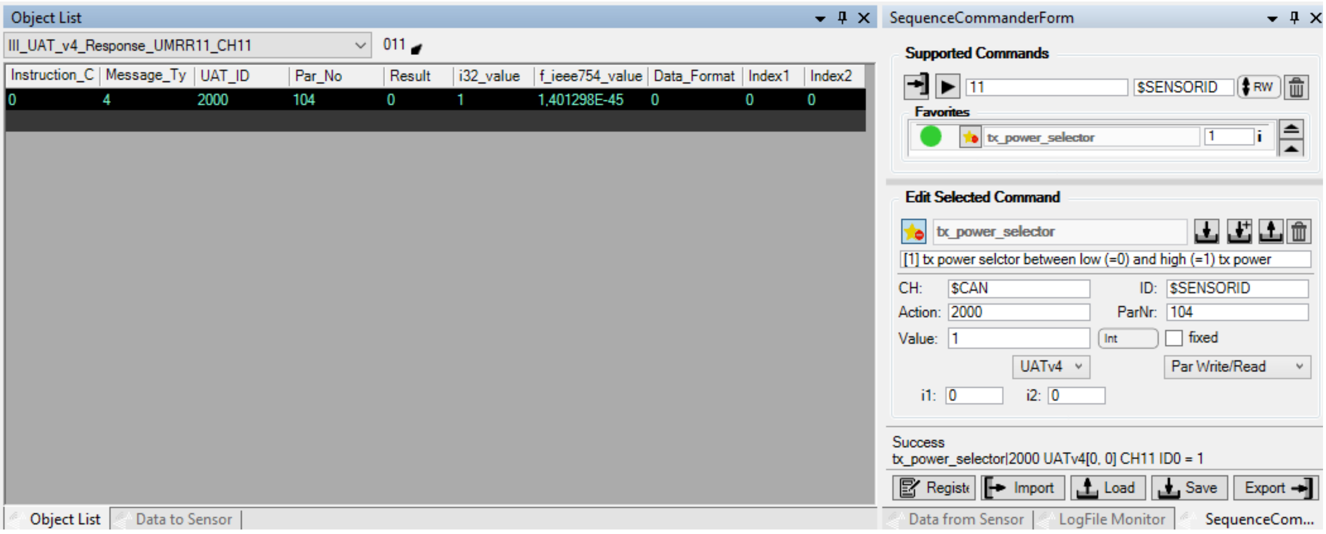



*Figure 6-4: Reading and changing a sensor parameter*

In order to change the value of a parameter, such as the orientation angle of the sensor, please enter the parameter (ID), the action number (Section), and the new value into the command window. Please check if UAT V4 is selected, choose "Par Write" from the drop-down list and send it to the sensor by clicking "Send". The parameter will be changed accordingly, as you can see below under f_value.

## REQUEST STATUS

To read the parameter status, please enter the parameter number (ParNr. = ID) and the action number (Section) into the command window. Please check if UAT V4 is selected, choose “read only” from the drop-down list and enter the number format (Int or Float) of the parameter. By clicking “Send” the “ReadParameter” view in the “SensorTargetList” window will show the parameter as below.

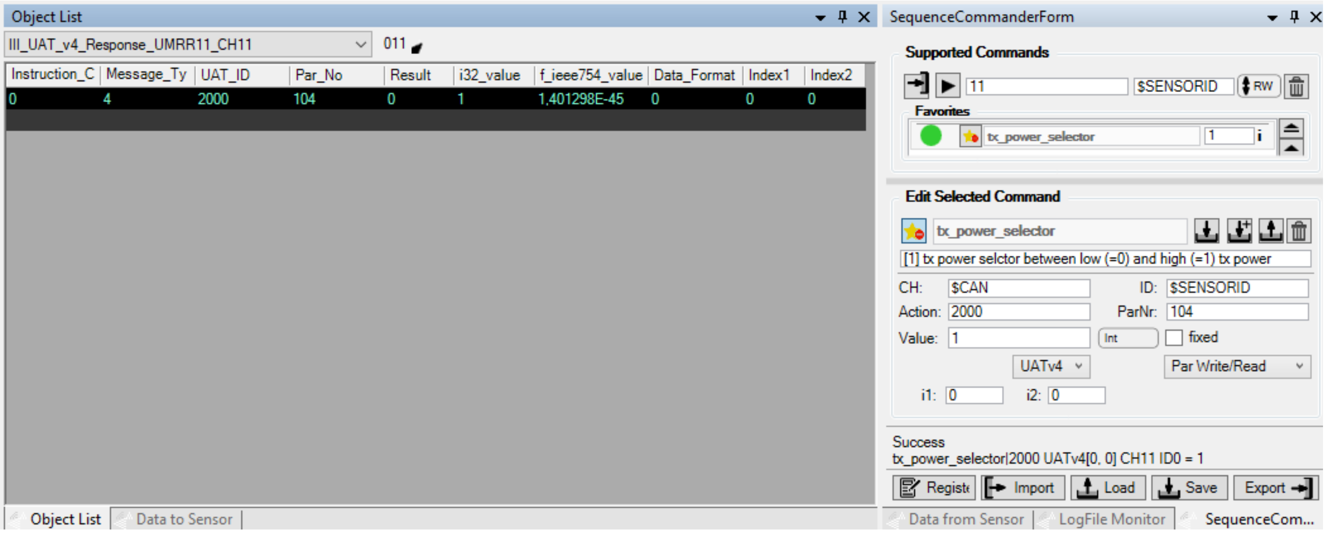


*Figure 6-5: Reading a sensor parameter*

## 6.4 EXPORTING DATA TO CSV FORMAT

To export data to the .csv format, please select “Views / Interpreted CAN Data Views / CSV Export”.

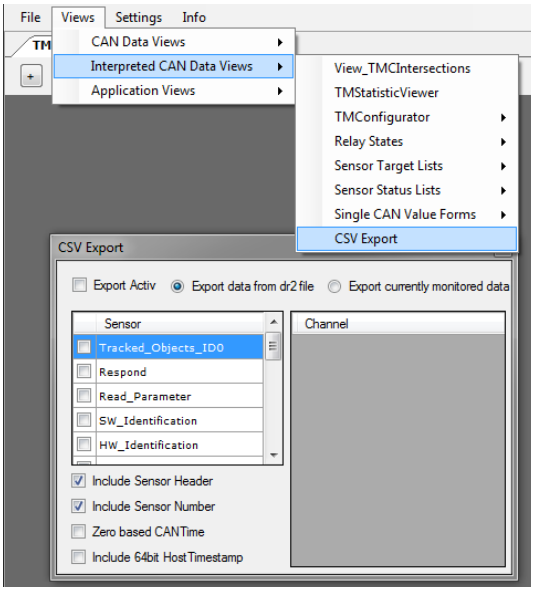



*Figure 6-6: CSV Export*

- To log tracked object data, select “Tracked Objects ID0”
- To log statistics data, select “TMStatistics ID0”
- To log Event Triggers, select “Relays ID0”

The export function can be activated by activating the “Export Active” checkbox in the upper left corner of the window. The .csv file will be named after the dr2-file and will be placed in the same location.

If the option “Include Sensor Header” is selected, there will be an additional .txt file with the column headlines.

The CSV export options are:

- Include Sensor Header: includes the header information (CAN ID 0x500, 0x501) in the .csv file
- Include Sensor No.: includes the sensor number in the .csv file
- Zero based CANTime: sets the CANTime to zero at the beginning of the .csv file

## EXAMPLE OF A CSV FILE

After the export, a .csv file and a .txt file are generated. The .txt file contains the description of the exported data and it indicates the meaning of values within the columns. When exporting into .csv format, the information taken over for the .csv file is shown.

Zwischenablage | Schriftart | Ausrichtung

A1 | fx 55,699

| | A | B | C | D | E | F | G | H | I | J |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 55,699 | 0 | 0 | 39,680002 | 6,784 | 1,5 | -11,8 | -0,3 | 3,4 | 7 |
| 2 | 55,699 | 0 | 1 | 45,440002 | 6,4 | 1,5 | -12,6 | -2,9 | 3,4 | 6 |
| 3 | 110,798 | 0 | 0 | 39,296002 | 6,592 | 1,5 | -11,8 | -0,5 | 3,4 | 7 |
| 4 | 110,798 | 0 | 1 | 44,928002 | 6,24 | 1,5 | -12,6 | -3 | 3,4 | 6 |
| 5 | 167,679 | 0 | 0 | 38,656002 | 6,4 | 1,5 | -11,8 | -0,5 | 3,4 | 7 |
| 6 | 167,679 | 0 | 1 | 44,416002 | 6,048 | 1,5 | -12,6 | -3 | 3,4 | 6 |
| 7 | 222,847 | 0 | 0 | 38,144002 | 6,176 | 1,5 | -11,8 | -0,8 | 3,4 | 7 |
| 8 | 222,847 | 0 | 1 | 43,776002 | 5,888 | 1,5 | -12,6 | -3 | 3,4 | 6 |
| 9 | 279,199 | 0 | 0 | 37,504002 | 5,952 | 1,5 | -11,7 | -1,6 | 3,4 | 7 |

*Figure 6-7: CSV export example in Excel*

dr2-File_00.txt - Editor

Datei Bearbeiten Format Ansicht ?

```
1. Time [ms], 2. Sensorno., 3. Object_Number, 4. x_Point1, 5. y_Point1, 6. Width, 7. Speed_x, 8. Speed_y, 9. Length, 10. Object_Class,
```

*Figure 6-8: CSV export description*

Exemplarily for line 1:

- Sensor Time: 55.699 ms
- Sensorno.: 0
- Object Number: 0
- X_Point: 39,680 m
- Y_Point: 6,784 m
- Width: 1.5 m
- Speed_x: -11,8 m/s
- Speed_y: -0,3 m/s
- Length: 3,4 m
- Object Class: 7

## 6.5 ADJUST STOPPING VEHICLE PERFORMANCE FOR FARTHER DISTANCE

In case that the user wants to track stopping vehicles in farther distance it is possible to adjust the preset default configuration following the required instructions:

| Name | ID | Comment | Section | Types | Recommended Values |
|---|---|---|---|---|---|
| max_range_stationary_targets | 387 | maximum range for stationary target detection | 2000 | Parameter, Integer | 80m default<br>50m…180m |
| max_dist_to_become_drone | 129 | max distance [m] for the track to become a drone. | 2004 | Parameter, Float | 100m default<br>50m …180m |
| max_dist_drone | 130 | max Distance [m] for a drone to become active | 2004 | Parameter, Float | 120m default<br>70m…200m |

Example Configuration:

max_range_stationary_targets: 160m

max_dist_to_become_drone: 160m

max_dist_drone: 180m

**Caution**: it is not recommended to increase the stationary target and drone distance range to a very value. Changing the default values can affect the overall sensor performance (increased cycle time may, additional ghost objects etc.)

## 6.6 USING THE POLYGON FEATURE

Please note that this section is for expert users only.

With the polygon feature of the radar sensor up to eight zones can be defined and configured separately regarding the kinds of objects that should be processed in the algorithm.

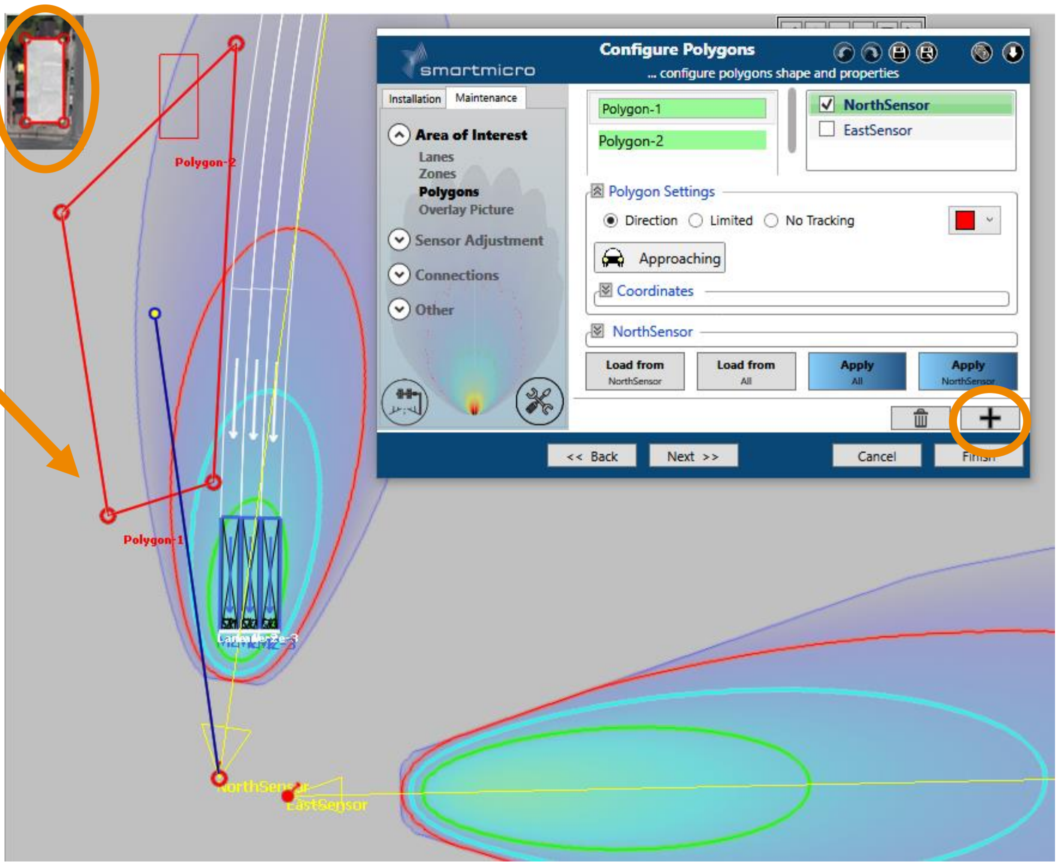



*Figure 6-9: Polygon feature*

### 6.6.1 CONFIGURATION OF POLYGONS

To configure polygons, please go to the “Maintenance / Area of Interest / Polygons”.

- Polygons can be added or deleted per drag-and-drop or by clicking the plus icon
- The corner points of a polygon can be changed by selecting the relevant polygon within the polygon list and either moving the points or adjusting them in the point list
- The behavior of the polygon can be determined by the polygon settings
- Polygons need to be assigned to sensors individually, for example, to “NorthSensor”
- In case two or more polygons are covering the same are, their behavior can be handled by defining their priority. In the are where polygons are overlaping, the polygon with the highest priority will define the behaviour and the others will be ignored.

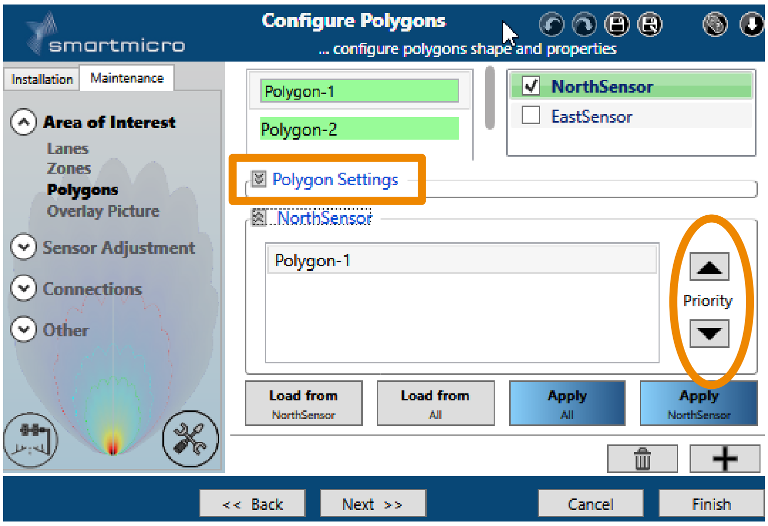



*Figure 6-10: Polygon priority*

### 6.6.2 POLYGON SETTINGS

There are three use cases for polygons:

1. Direction: The polygon will let the direction set as approaching or receding pass. The other direction will be blocked and not reported by the sensor.
2. Limited: The polygon will let all vehicle reports matching with the set speed limits pass. “Within limits”, “outside limits” and “target options” are for the experience user only.
3. No Tracking: No tracking is performed inside the polygon.

To make the polygons effective, please click “Apply All” for sending all polygons to all sensors or click “Apply <Name>” for sending the polygon settings to a particular sensor.

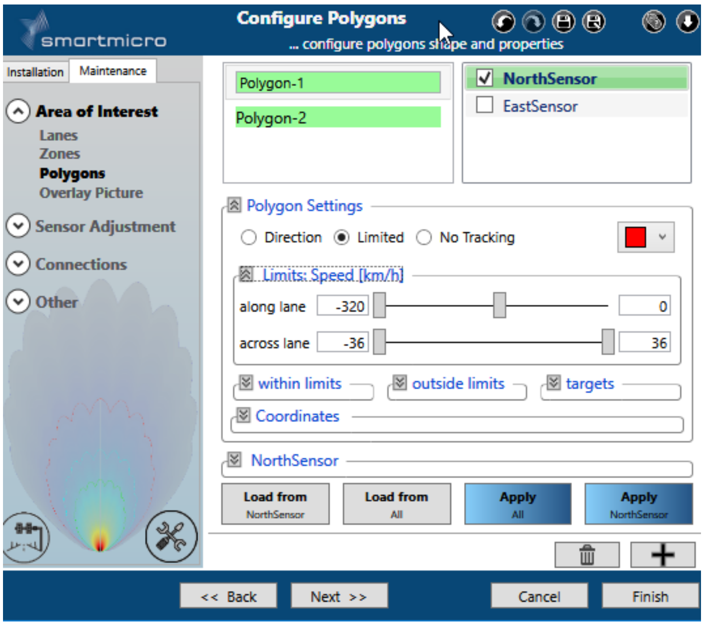



*Figure 6-11: Polygon settings*

### 6.6.3 LIMITED POLYGONS

The behavior of the polygon should be defined for the area within the limits and outside of the limits:

- Delete tracks: Object tracks will be deleted.
- Deactivate object tracking: The object tracking will be completely deactivated. As a result, vehicles will open up a new track after they leave the polygon area.
- Don’t transmit tracks: The object tracking will work normally but no tracked object will be reported.
- Tracked objects can hold: Tracks will not be deleted after a full stop.

The target handling can be defined as follows:

- Don’t initiate tracks: targets cannot open new tracks
- Don’t associate with tracks: keeps the default settings
- Stationary targets can init track: no effect

## 6.7 SETTING A SPLINE

It is possible to define up to 22 splines. A single spline can have four to six spline nodes.

The number of splines and the spline nodes cannot be determined directly. The end of the spline is indicated with a special value, the float32 value 2047.0. For every node *n* with this value, the number of nodes for this spline is n-1. If the first node of a spline *p* contains this value, only spline 1 to spline *p*-1 exist.

Each spline node has x- and y-coordinates. A spline can be used for approaching, receding, or bi-directional traffic in relation to sensor. The sequence of the spline nodes determines the direction of the spline in single direction mode, based on the assumption that a vehicle drives from the first node to last node. Bi-directionality is set by a separate parameter.

### 6.7.1 SPLINE PARAMETERS

The spline parameters are UAT V4 formatted arrays. The array dimensions are defined as follows:

- i1 = spline = spline number [0 ≤ i1 < max. number of splines]
- i2 = node = spline node number [0 ≤ i2 ≤ 6]

Spline parameter overview (Table 6-6):

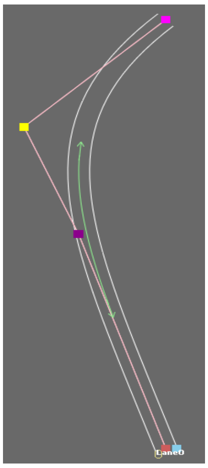
| Name | Description | ID | Dimensons | Default Value | Section | Type | Typical Values |
|---|---|---|---|---|---|---|---|
| Spline Node X-Coordinate | Position on x-axis of a node to calculate a spline | 0 | Spline, node | various | 2005 | Parameter, Integer | -300 to +300<br>2047≡ invalid |
| Spline Node Y-Coordinate | Position on y-axis of a node to calculate a spline | 1 | Spline, node | - | 2005 | Parameter, Integer | -20 to +20<br>2047≡ invalid |
| Bi-Direction | Allows for traffic in both directions | 4 | Spline | 1 | 2006 | Parameter, Integer | 0 or 1<br>0 = off, 1 = on |
| Bicycle_spline | Flag for bicycle spline | 5 | Spline | 0 | 2006 | Parameter, Integer | 0 or 1 |

### 6.7.2 SPLINE EXAMPLE

First, please create one lane in the TUI and send it to the sensor. This is useful to make the spline visible in the TUI. Secondly, please change the values of the lane as below.

Values and the shape of the exemplary spline (Table 6-7):

| Name | ID | i1 | i2 | Value | Type | Section |
|---|---|---|---|---|---|---|
| 1. Spline node x-coordinate | 0 | 0 | 0 | 0 | Parameter, Integer | 2005 |
| 1. Spline node y-coordinate | 1 | 0 | 0 | -2 | Parameter, Integer | 2005 |
| 2. Spline node x-coordinate | 0 | 0 | 1 | 50 | Parameter, Integer | 2005 |
| 2. Spline node y-coordinate | 1 | 0 | 1 | 14 | Parameter, Integer | 2005 |
| 3. Spline node x-coordinate | 0 | 0 | 2 | 75 | Parameter, Integer | 2005 |
| 3. Spline node y-coordinate | 1 | 0 | 2 | 24 | Parameter, Integer | 2005 |
| 4. Spline node x-coordinate | 0 | 0 | 3 | 100 | Parameter, Integer | 2005 |
| 4. Spline node y-coordinate | 1 | 0 | 3 | -2 | Parameter, Integer | 2005 |
| Bi-direction | 4 | 0 | 0 | 1 | Parameter, Integer | 2006 |
| Bicycle spline | 5 | 0 | 0 | 0 | Parameter, Integer | 2006 |



## 6.8 RECORD START STOP

The record start-stop feature offers help as soon as the customer has problems with the radar sensor performance. It allows the customer to easily make recordings, which are analyzed by us to find a solution. The easiest way to handle the feature is to use the Traffic Management Configurator. After starting the recording no more parameter changes are possible. Parameters and some static internal data will be saved first. After, the sensor streams the dynamic sensor information as long as the recording is running. After stopping the recording, you can take the data and send to your smartmicro correspondent.

### 6.8.1 COMMANDS/ PROCESS

You do not have to use the TUI tool. You have also all commands to handle the recording process of the sensor.

The first step is to mount the sensor in the position you want to make the recording. Afterwards configure the sensor for this situation. This is the normal installation process. When the normal installation and configuration process is done, you can start with the recording. Note, the sensor must run with a special record start stop software.

First the sensor must be set in a configuration mode. There you have the command tm_fsm_core0_opmode in table Record start stop commands (Table 6-8). Set the value to 2 to set the configuration mode. Afterwards, disable the parameter access with the command Tm_fsm_core0_param_wr_disable with the value 0. Now, the static data export can be started by the command tm_start_recording. The static data ports will be transmitted now. After there is no data left, thcome sensor can be set back into the normal mode by value 1 with command tm_fsm_core0_opmode. Now, you see again the dynamic ports, which must be recorded. Record the data and catch the scene, you want to show. After you stop the recording, you only must enable the parameter access again. In the end, are all data recorded, the parameter write access should change to active with sending the command Tm_fsm_core0_param_wr_enable.

Record start stop commands (Table 6-8):

| Name | ID | Default Value | Section | Type |
|---|---|---|---|---|
| tm_fsm_core0_opmode | 11 | - | 2020 | command |
| Tm_fsm_core0_param_wr_disable | 19 | - | 2020 | command |
| Tm_fsm_core0_param_wr_enable | 24 | - | 2020 | command |
| tm_start_recording | 20 | - | 2020 | command |
| tm_stop_recording | 22 | - | 2020 | command |

# 7 TUI SOFTWARE

The Traffic Management Configurator (TUI) software is the most convenient way to set up radar sensors, as well as smartmicro's Traffic Management Interface Board (TMIB), Serial Relay Option (SRO), and Cabinet Relay Option (CRO) controller cards using a Windows-based computer.

Please contact us for more information about the system requirements, the supported hardware interfaces, as well as a detailed description on how to connect the sensor and how to download and install the firmware.

## 7.1 OVERVIEW

When starting the Easy Mode for the first time, the graphical user interface will appear as follows:

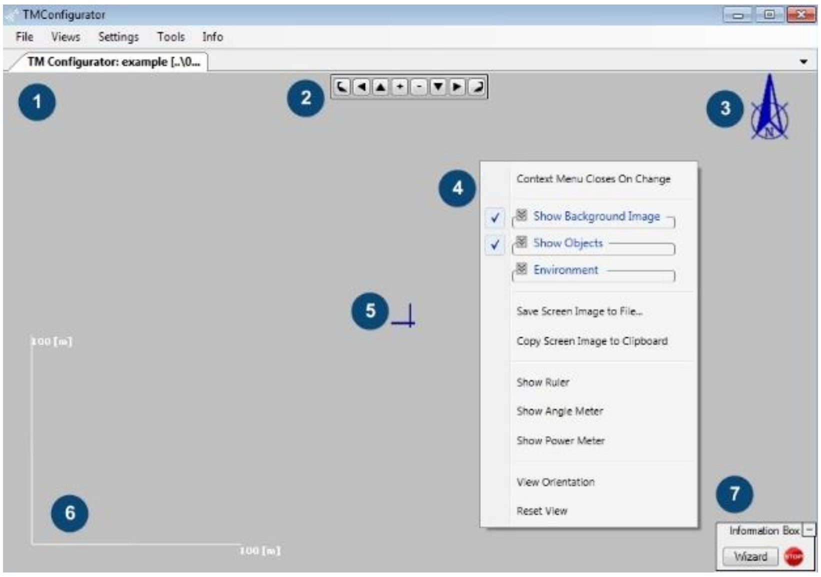



*Figure 7-1: TUI Interface*

The visualization and graphic user interfaces for the sensor setup and most traffic management applications is illustrated above. It contains the visualization of traffic data on a two-dimensional scale in which the objects, the vehicles, are represented by rectangles.

The key elements of the TUI are enumerated within the figure above and have the following functions:

1) With a click on the tab the views settings can be entered.
2) With the navigation bar the position, zoom and rotation of the area can be adjusted.
3) The compass indicates the alignment of the map in the background.
4) The context menu for highlighting or masking visualization settings can be opened by right click.
5) The crosshair indicates the origin of the area's coordinate system.
6) The scale on the lower left side shows the current zoom of the area.
7) To start the setup process, the wizard button on the information box can be used.

## 7.2 TUI INSTALLATION WIZARD

The TUI Installation Wizard simplifies the planning and field installation of smartmicro sensors with an intuitive step-by-step guide. The configuration includes site planning, sensor selection and configuration as well as physically installing radar sensors.

When the installation wizard opens, the following start screen is displayed:

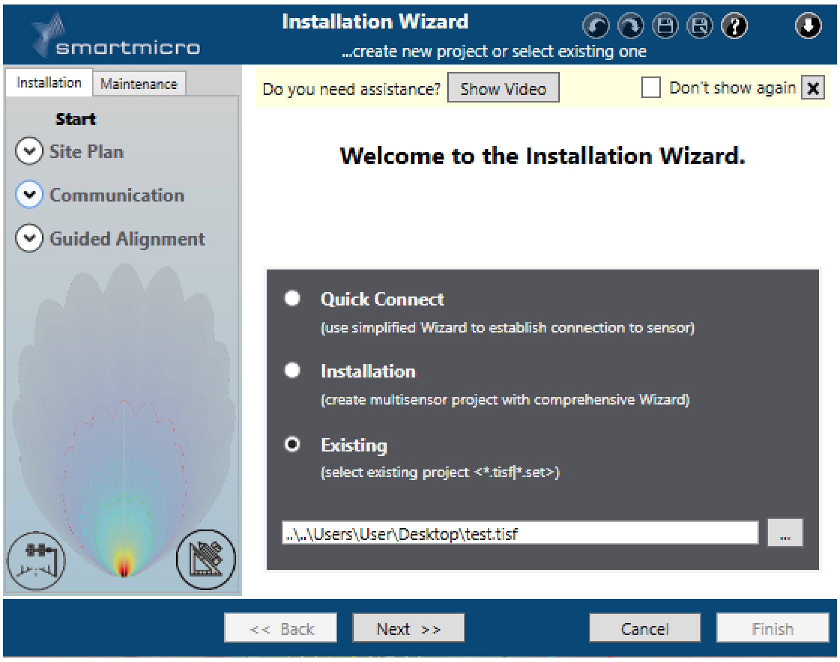



*Figure 7-2: The TUI Installation Wizard*

The menu on the left side indicates the current step of the setup process. While the installation tab includes all necessary settings and elements for the basic installation, the maintenance tab provides additional functions and allows for limited changes of settings.

On the top of the right side an explanatory video is provided for a better understanding of the functions of the current step. Below that, the display area shows the functions of adding elements and parameters to the installation.

## 7.3 NEW PROJECTS IN THE TUI

On the start screen of the TUI Installation Wizard there are for options of project types to choose from:

1. Quick Connect: intersection setup of one sensor and maximum one Traffic Management Interface Board (TMIB or TMIB2)
2. Installation: intersection setup using one or more sensors
3. New Speed Enforcement: setup of one enforcement sensor
4. Existing: selection of a project file (.tisf or .set) for editing

A project contains all setups of an installation and is saved as .tisf file. Projects are desktop independent, so every user can keep an own workspace, or rather desktop, and is able to switch between different projects.

## 7.4 STEP-BY-STEP SETUP WITH THE TUI

### STEP 1) FINDING AN INSTALLATION LOCATION

The desired intersection or highway location can be found by using the search function like on Google Maps. Please enter the address, type in the latitude and longitude coordinates directly or navigate to the desired location by using the mouse. The TUI will load a map from different providers of which some maps are of high definition, already including lanes and stop bar positions.

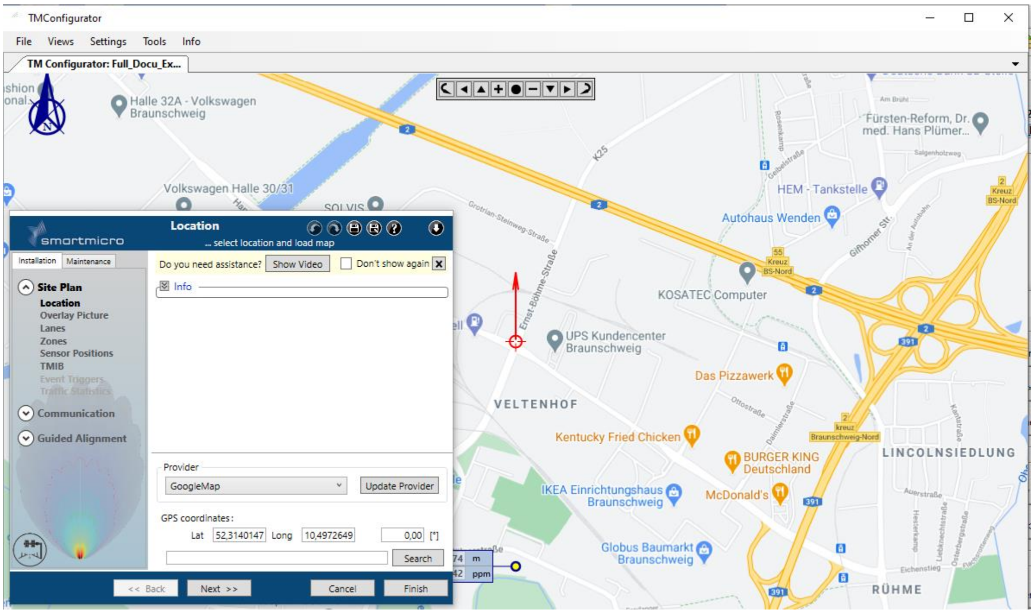



*Figure 7-3: Step 1) Finding an intersection or highway location*

### STEP 2) LOADING A SATELLITE MAP

Please load a satellite image or a cartographic image to overlay it with the map.

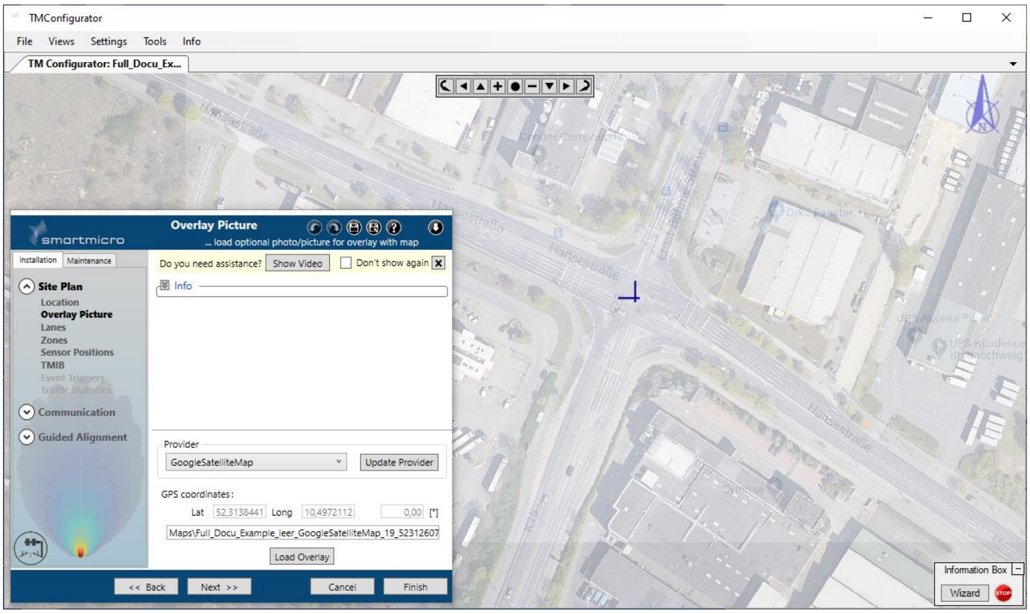



*Figure 7-4: Step 2) Loading a satellite image*

### STEP 3) CONFIGURING LANES

Based on the satellite image the lanes can now be created.

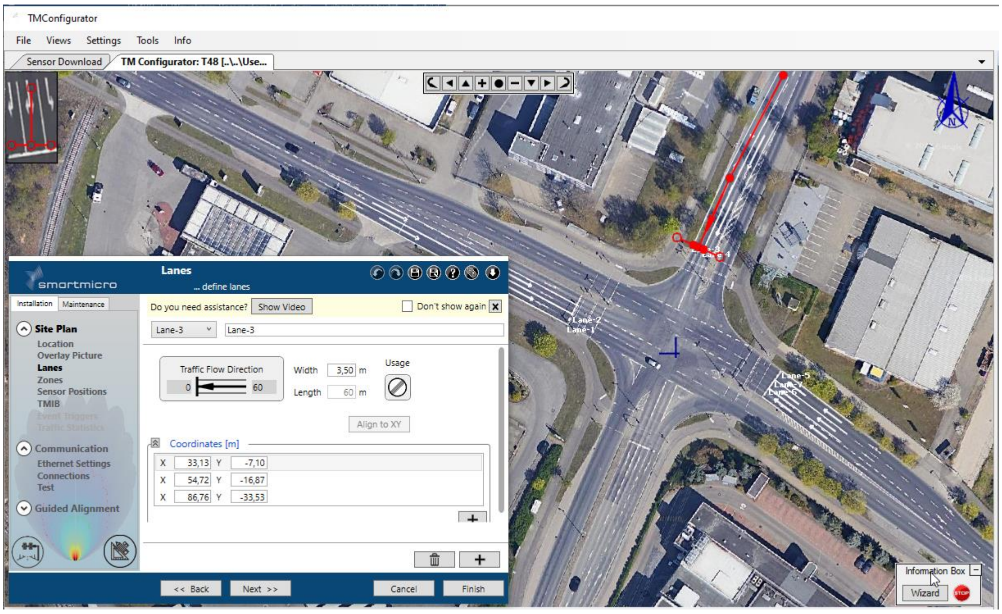



*Figure 7-5: Step 3) Configuring lanes*

### STEP 4) ADDING DETECTION ZONES

Detection zones can now be added to the location. To get default values for each application, please refer to the corresponding sensor datasheet.

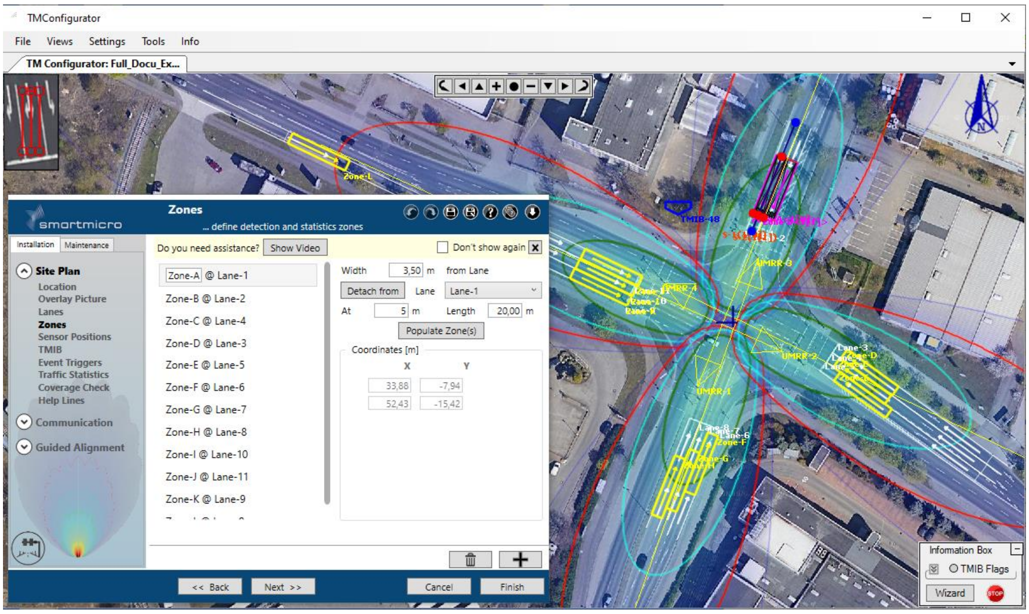



*Figure 7-6: Step 4) Adding detection zones*

### STEP 5) ADDING A SENSOR

Radar sensors can be added to the image via drag-and-drop from a list of automatically proposed sensor types. Each sensor is preconfigured, so their field of view and detection range are visible immediately. Please adjust each sensor, so that the sensor beam covers all lanes. For stop bar detection it is recommended to locate the stop bars within the green part of the sensor beam. For further information on the default values for each application, please refer to the corresponding sensor datasheet.

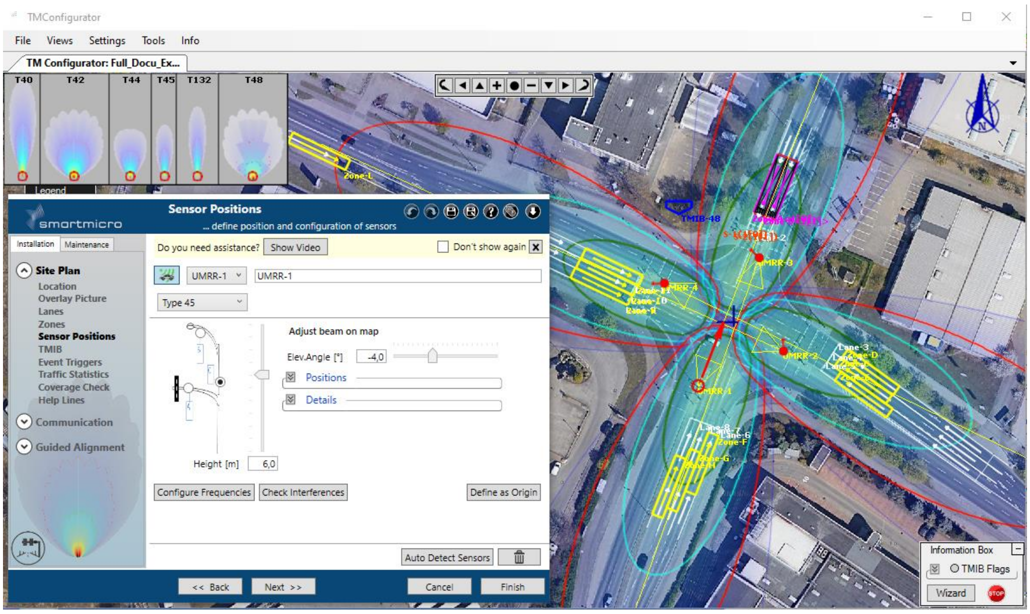


*Figure 7-7: Step 5) Adding a sensor*

## STEP 6A) ASSIGNING RELAYS

Please assign the detection zones to each sensor, set the desired trigger configurations and assign relays.

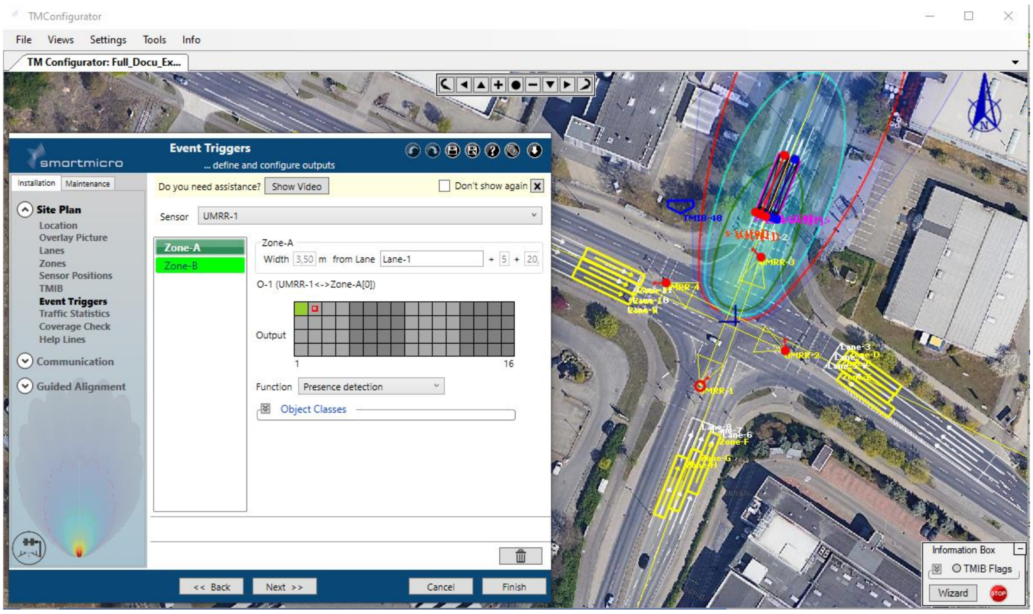


*Figure 7-8: Step 6a) Assigning relays*

## STEP 6B) ASSIGNING TRAFFIC STATISTICS

Please set the traffic statistics to the desired detection zone.

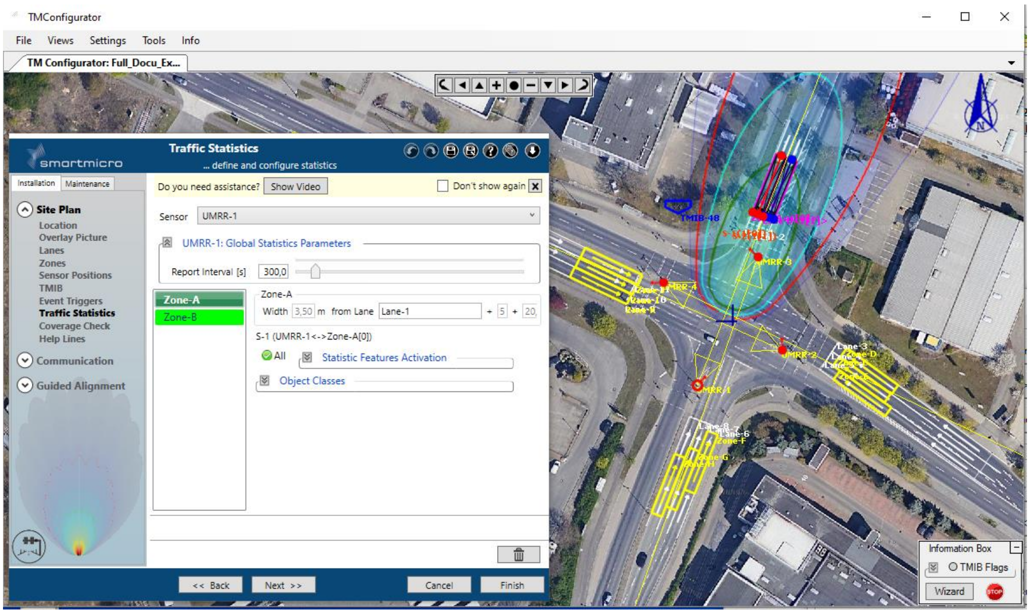



*Figure 7-9:Step 6b) Assigning traffic statistics*

## STEP 7) VERIFYING THE BEAM COVERAGE

After adding the lanes, sensors, and detection or counting zones to the project, the TUI will check the sensor beam coverage. In case of a misplacement, a remark will explain how to fix the beam coverage.

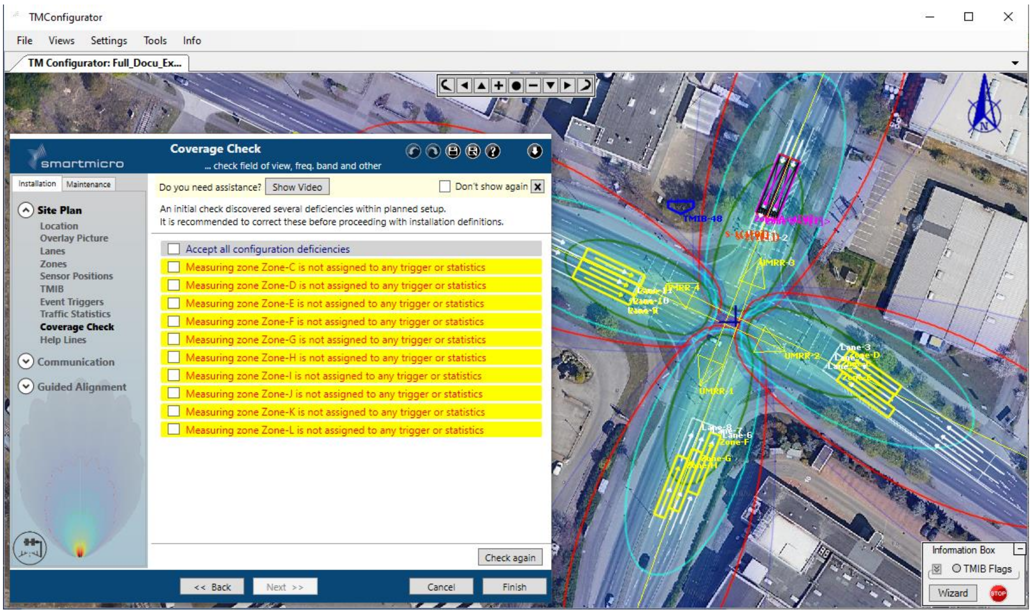



*Figure 7-10: Step 7) Verifying the beam coverage*

## STEP 8) VERIFYING THE COMMUNICATION

Please physically connect all sensors and the TMIB, for the communication of all radar devices to be established automatically. The TUI will verify the communication of the sensors. If necessary, the sensor ID will be changed automatically to avoid interference.

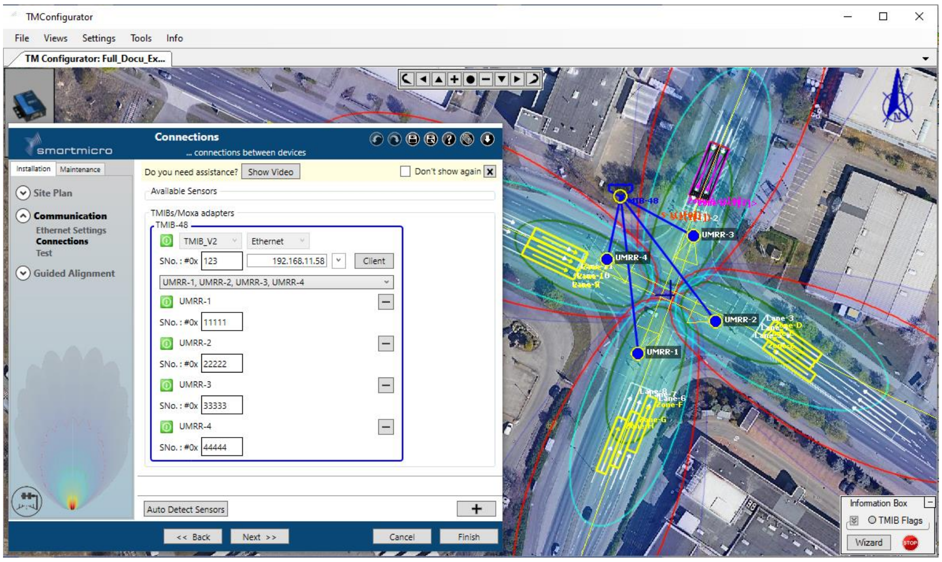



*Figure 7-11: Step 8) Verifying the communication*

### STEP 9) VERIFYING PITCH AND ROLL ANGLE

The TUI will also give feedback if the sensor has the correct pitch and roll angle. The sensor position can be physically adjusted, and the software will immediately show if the angle is correct.

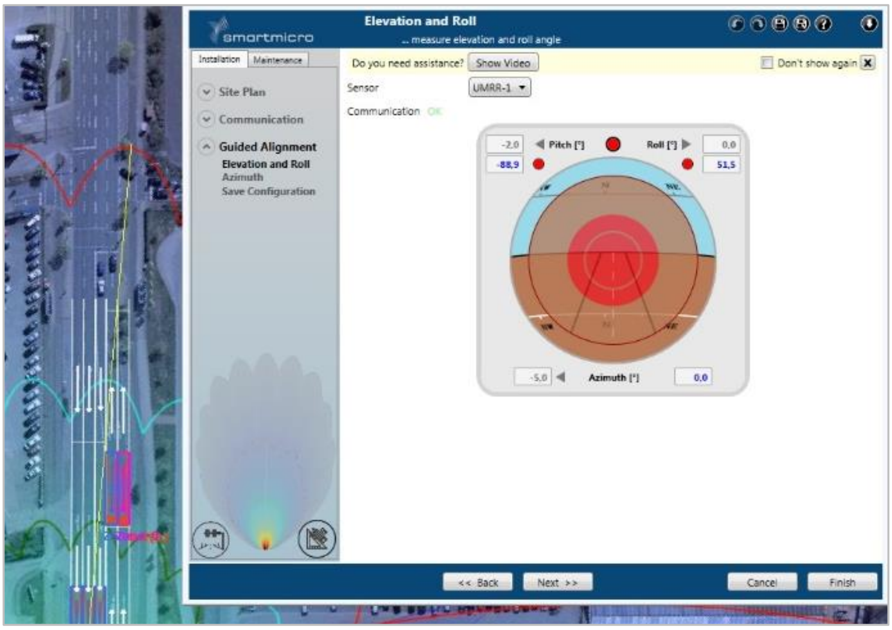



*Figure 7-12: Step 9) Verifying pitch and roll angle*

### STEP 10) ALIGNING THE SENSORS

The sensors and the TMIB can now be installed on-site at the selected location. After the installation, please start the “Check orientation” test for each sensor. The TUI will analyze the traffic flow, and, based on this, give a recommendation to adjust the azimuth and elevation angle.

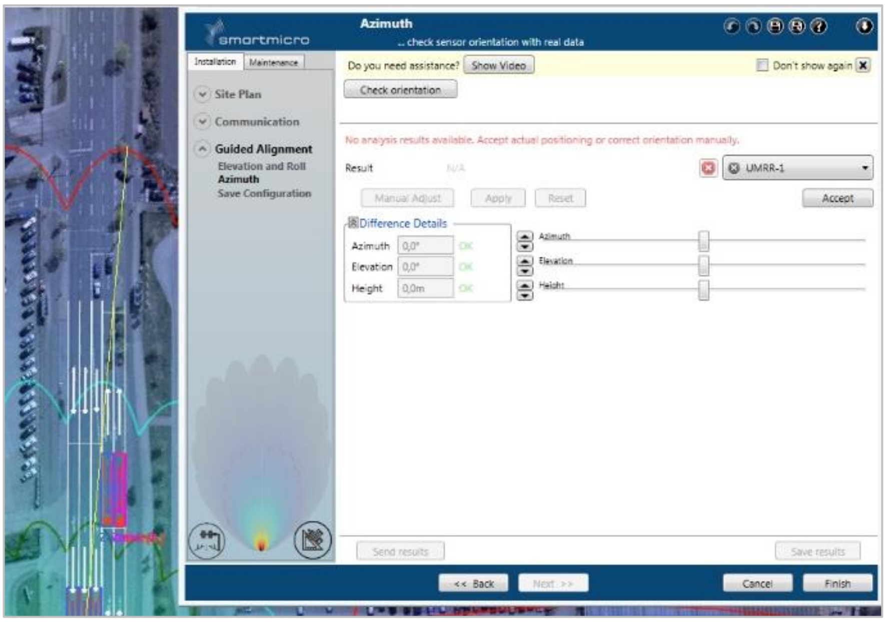



*Figure 7-13: Step 10) Aligning the sensors*

## STEP 11) SETTING UP THE TMIB

The setup for the TMIB can be changed and activated, depending on the traffic controller NEMA TS1/TS2. Also, the relay output from the TMIB can be adjusted.

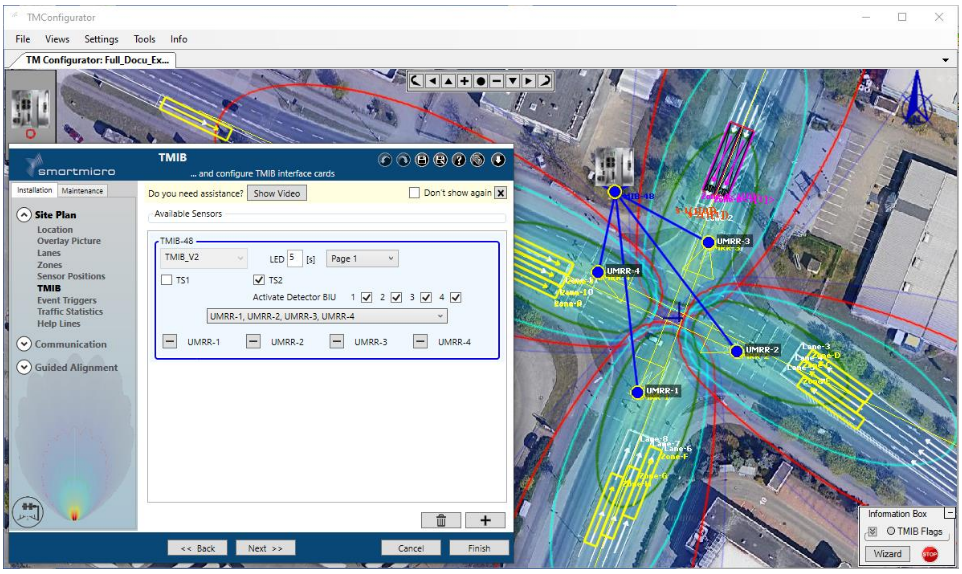



*Figure 7-14: Step 11) Setting up the TMIB*

## 7.5 FIRMWARE UPDATES FOR THE TUI

The download window of TUI can be used for firmware or software updates on smartmicro sensors. Firmware files can have as extension. cbx or .xml or .emc. To update the firmware, the connection to the sensor needs to be established for presetting the download feature. The sensor download window can be opened as follow

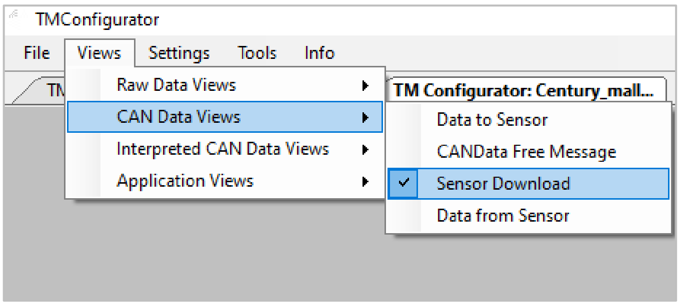



*Figure 7-15: Choose and load a desktop*

If the sensor is connected to the TUI, the Installation Wizard can be closed via the finish button. Now, please choose the file for the update.

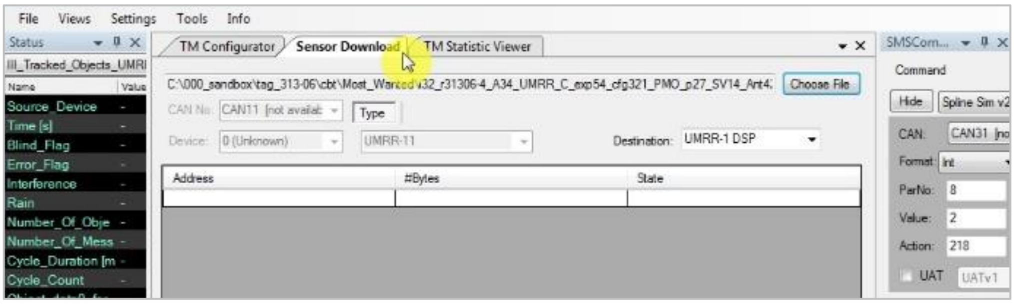



*Figure 7-16: Download window*

Please make sure that the automatic detection of the radar type and interface are selected correctly. If needed, "custom settings" can be chosen from the destination drop-down menu in order to change the sensor type and interface manually. Once the correct sensor type is selected, please click on "Start Download" to update the firmware.

## 7.6 SENSOR CONFIGURATION PARAMETERS

Sensor configuration parameters are stored within the sensor's non-volatile memory and available upon sensor reboot or restart without external intervention. To amend an existing sensor's configuration, it is recommended that the computer has the sensor's configuration file available to facilitate simple and effective re-programming of the sensor's parameters. This configuration file is typically named as the location and suffixed by the ".TSIF" file type.

Simply open the TUI Software, open the relevant TSIF file and connect to the target sensor. You are then able to optionally retrieve the sensor's configuration parameters and change as required prior to saving back to the sensor.

Please note that the TSIF file carries the background map data, which is not saved to the sensor.

If you do not have the TSIF file available, you are able to auto-connect to the sensor and retrieve the configuration parameters by navigating to the Maintenance Tab, then navigate to the Sensor Alignment Tab, and retrieve all parameters. In this instance, you would not have the background map to verify lane placement. You would be able to perform all functions to change zones, output assignment, and lane assignments. Saving the configuration parameters would be performed in the normal way.

### 7.6.1 SENSOR CONFIGURATION DOWNLOAD (NO TISF)

If you do not have the TSIF file available, follow the below steps:

1. Load TM Configurator (TUI) program
2. Create a New TISF file by selecting installation mode
3. TUI will open to Site Plan -> Location by default
    a. Navigate down to the Sensor Positions Tab
4. In the bottom right of the wizard select the Auto Detect Sensors button
    a. If connect Directly to the sensor or TMIB, go to step 5
        i. Note: Your computer should be set as server IP 192.168.11.1
    b. If connect via TMIB in server mode go to step 6
        i. Note: You will need to know your TMIB IP and make sure your connected to that network.
5. Wait for the sensor to be detected
    a. Once you see a green check mark, and Type and Antenna Return values
    b. Hit the OK button at the bottom
    c. The sensors will be placed on the map
    d. Go to 7
6. Open the double down arrow next to Connect to Server at the bottom
    a. Un-check Ethernet, SMS Ethernet, RS485, and CAN
    b. Put in the IP of the TMIB in the first IP address slot

c. Click Check button below

d. Check mark the Ethernet at the top of the wizard

e. Wait for all sensors to be found

f. Click OK from the bottom

g. The sensors will be placed on the map

h. Go to 7

7. Navigate to Communication

    a. Navigate to Test

    b. Wait for test to complete

8. Navigate to Maintenance Mode Tab

    a. Navigate to Sensor Adjustments

    b. Navigate to All Parameters

    c. Select Load (All) - This will download all parameters from the sensors.

In these instances, you would not have the background map to verify lane placement. But you would be able to perform all functions to change zones, output assignment, and lane assignments. Saving the configuration parameters would be performed in the normal way.

### 7.6.2 SETTING THE TIME IN THE SENSOR

TUI send by default the current UTC time stamp of the windows machine, where TUI is operating. This can be deactivated or activated in the TUI settings

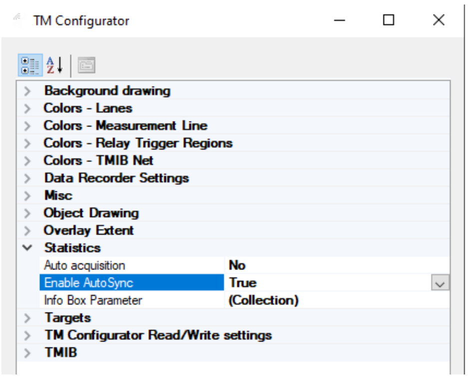



*Figure 7-17: TUI setting window*

In addition to this, the user can synchronize manually by button click in TUI. This can be done in the maintenance menu under section Others->time Synchronization

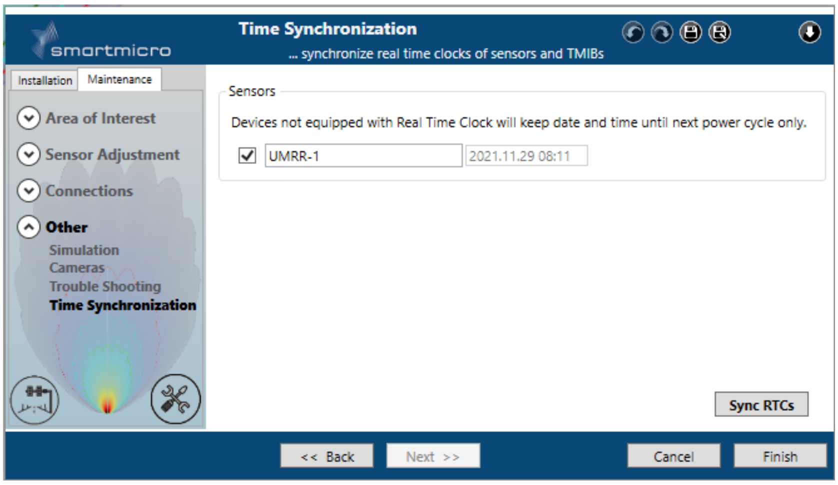



*Figure 7-18: TUI setting window*

Once the user clicks on the button 'Sync RTCs', The TUI will send the instruction for setting the unix time, which is described in the figure below

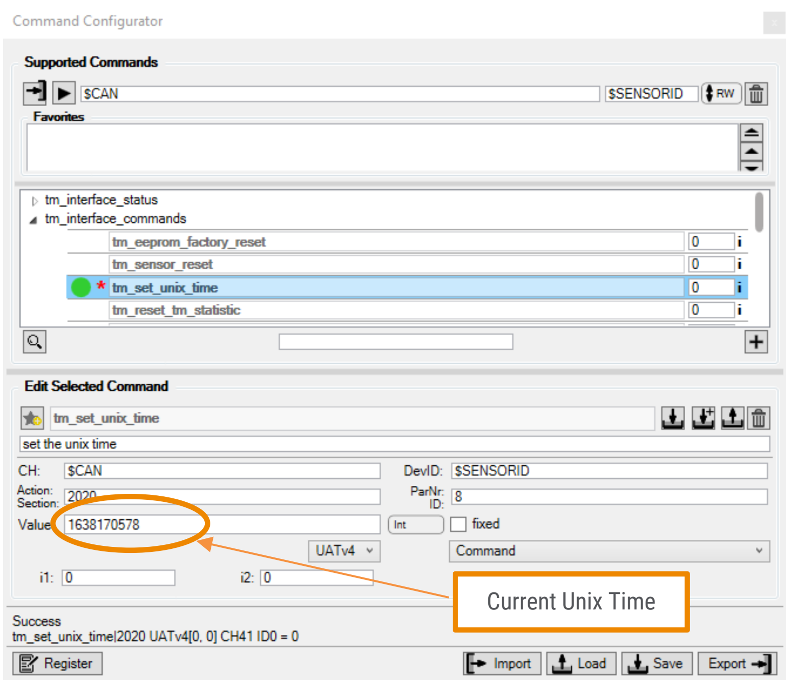



*Figure 7-19: Instruction for setting the unix time*

Note: The sensors are not equipped with a real time clock, which is able to store the date and time after a power cycle. After a power lost, the time clock will go back to 01.01.1970

### 7.7 RECORD START STOP WITH TUI

For an instruction into the radar start stop feature, please read the chapter 6.8. First, you must be sure, that the radar sensor is correctly installed and configured in the TUI. Attention, the umrr11 needs a special firmware to be able to work with the start stop feature. When you finished all configurations, you can start the wizard (Figure 7-17).


*Figure 7-20: Open Wizard*

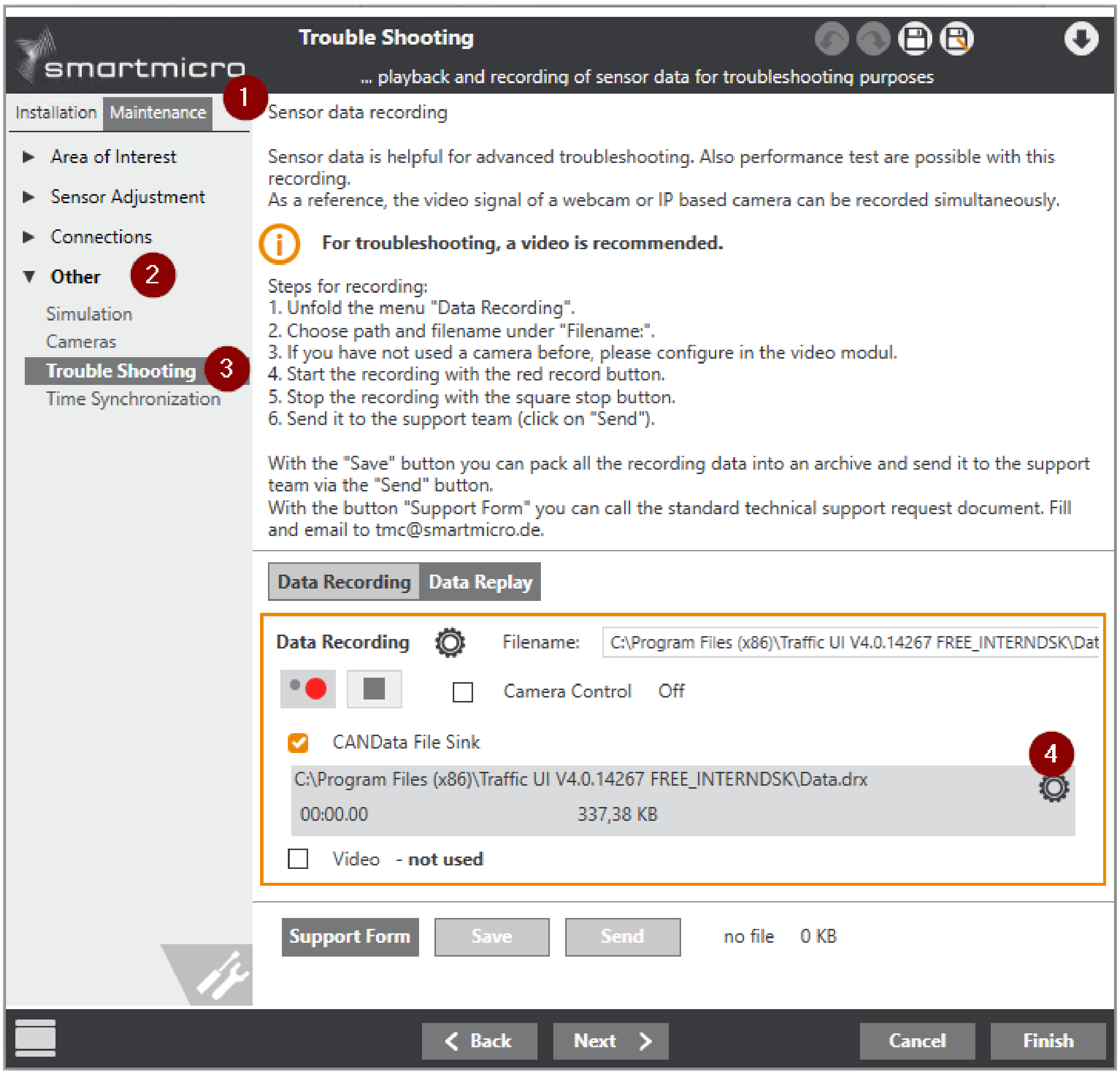



*Figure 7-21: Wizard Trouble Shooting*

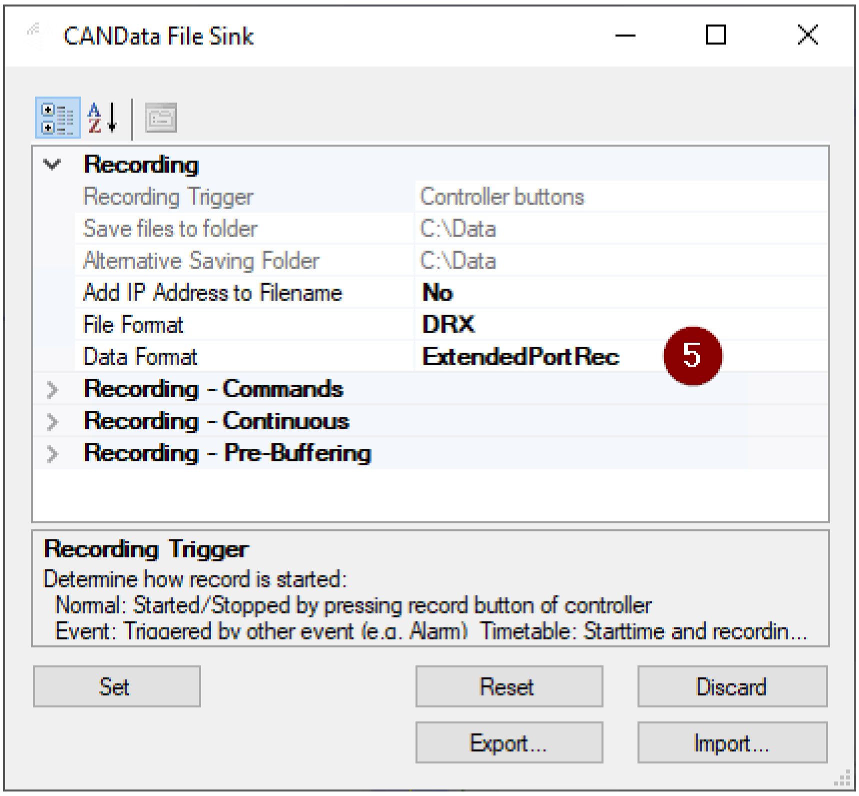



*Figure 7-22: CANData File Sink Window*

After you started the wizard (Figure 7-21), you can go to the “Maintenance” tab (1), “Other” (2) and “Trouble Shooting” (3). There you find the data recording window. Before you start the recording, you must go to the settings on (4) and change the data format to “ExtendedPortRec” (Figure 7-22). Now, you are ready for recording.

Choose the folder, where you want to save the recording and press the red recording button for starting the process.

# 8 FREQUENCY APPROVALS

smartmicro radar sensors are compliant with FCC, EU and other regulations and are notified in many countries. Please contact us to check the model-specific notification status at the time of the purchase.

## 8.1 DECLARATION OF CONFORMITY FOR EUROPE

The sensor has been marked with the CE mark. This mark indicates the compliance with the EC Directive 2014/53/EU. A full copy of the Declaration of Conformity can be obtained from s.m.s, smart microwave sensors GmbH, 38108 Braunschweig, Germany or via email at info@smartmicro.de.

# 9 LEGAL DISCLAIMER NOTICE

All products, product specifications and data in this document may be subject to change without notice to improve reliability, function or otherwise.

Not all products and/or product features may be available in all countries and regions. For legal reasons features may be deleted from products or smartmicro may refuse to offer products. Statements, technical information and recommendations contained herein are believed to be accurate as of the stated date. smartmicro disclaims any and all liability for any errors, inaccuracies or incompleteness contained in this document or in any other disclosure relating to the product.

To the extent permitted by applicable law, smartmicro disclaims (i) any and all liability arising out of the application or use of the product or the data contained herein, (ii) any and all liability of damages exceeding direct damages, including - without limitation - indirect, consequential or incidental damages, and (iii) any and all implied warranties, including warranties of the suitability of the product for particular purposes.

Statements regarding the suitability of products for certain types of applications are based on smartmicro’s knowledge of typical requirements that are often placed on smartmicro products in generic/general applications. Statements about the suitability of products for a particular/specific application, however, are not binding. It is the customer’s/user’s responsibility to validate that the product with the specifications described is suitable for use in the particular/specific application. Parameters and the performance of products may deviate from statements made herein due to particular/specific applications and/or surroundings. Therefore, it is important that the customer/user has thoroughly tested the products and has understood the performance and limitations of the products before installing them for final applications or before their commercialization. Although products are well optimized to be used for the intended applications stated, it must also be understood by the customer/user that the detection probability may not be 100% and that the false alarm rate may not be zero.

The information provided, relates only to the specifically designated product and may not be applicable when the product is used in combination with other materials or in any process not defined herein. All operating parameters, including typical parameters, must be validated for each application by the customer’s/user’s technical experts. Customers using or selling smartmicro products for use in an application which is not expressly indicated do so at their own risk.

This document does not expand or otherwise modify smartmicro’s terms and conditions of purchase, including but not being limited to the warranty. Except as expressly indicated in writing by smartmicro, the products are not designed for use in medical, life-saving or life-sustaining applications or for any other application in which the failure of the product could result in personal injury or death.

No license, expressed or implied, by estoppel or otherwise, to any intellectual property rights is granted by this document or by any conduct of smartmicro. Product names and markings noted herein may be trademarks of their respective owners.

Please note that the application of the product may be subject to standards or other regulations that may vary from country to country. smartmicro does not guarantee that the use of products in the applications described herein will comply with such regulations in any country. It is the customer’s/user’s responsibility to ensure that the use and incorporation of products comply with regulatory requirements of their markets.

smartmicro assumes no responsibility and cannot be held liable for the recording, processing, storage or other handling of video material and/or (personal) data captured or recorded with this product. It is the customer’s/user’s own responsibility to act in compliance with any applicable laws or other regulations. A clearly visible notice stating the usage of a video camera in the area of operation may be required.

If any provision of this disclaimer is, or is found to be, void or unenforceable under applicable law, it will not affect the validity or enforceability of the other provisions of this disclaimer.