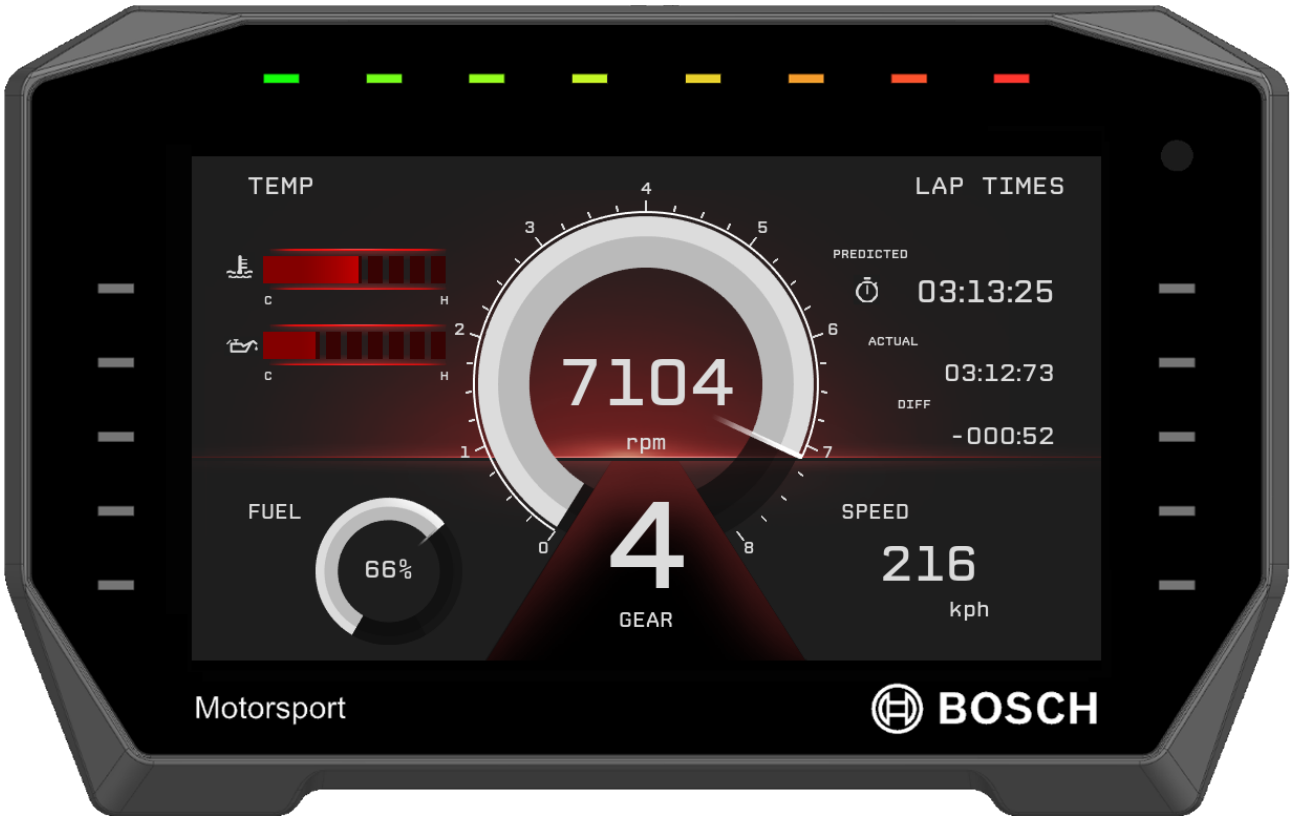
BOSCH



Display DDU 11

Manual

Version 0.1 18/01/2023

Content

1 Preparation 5

2 Warnings and Safety Instructions 6

3 Onboard Network Concept 7

4 Technical Data 8
4.1 Configurable Pins 10
4.2 Upgrades 11

5 System Configuration Tool RaceCon 12

6 First Steps 13
6.1 Starting the unit 13
6.2 Connecting the unit to RaceCon 13
6.3 Setting up a new RaceCon Project 14
6.4 Feature activation 19
6.5 First display configuration (Quick Start) 21
6.6 First recording (Quick Start) 23
6.7 Set date and time 26
6.8 Color indication 27
6.9 Assign the Mounting Location 29

7 Project Configuration 31
7.1 Math Channels 31
7.2 Conditional Functions 32
7.3 Conditional Channels 34
7.4 Condition Combination 36
7.5 Display Switch Module 38
7.6 Timer Module 39
7.7 GPS Trigger Module 39
7.8 CPU Load Limits 41

8 Display Configuration 42
8.1 Display page setup 42
8.2 Element Configuration 43
8.3 Types of display elements 48
8.4 Context menu 53
8.5 LEDs 54
8.6 Page select / Display brightness / LED brightness 58
8.7 Alarms 69
8.8 User Interface Menu UIM 76

9 CAN Configuration 79
9.1 CAN Bus Trivia 79
9.2 CAN input 80
9.3 CAN output 86
9.4 Multiplexer 89

10 Export and Import in RaceCon 92
10.1 Export in RaceCon 92
10.2 Import in RaceCon 92

11 Analog and Frequency Inputs 94
11.1 Analog inputs 94
11.2 Configuring inputs 95
11.3 Configuring computed sources 108
11.4 Hysteresis 109

12 Online Measurement and Calibration 113
12.1 Setting up an online measurement 113
12.2 Online calibration of measurement channels 118
12.3 Online calibration of multipoint adjustment channels 120
12.4 Using the Measurement Sheets 121

13 Error Memory 122
13.1 Error memory representation in RaceCon 122
13.2 Information on errors available from the error memory 124
13.3 Analog Input Diagnosis 128

14 Recording 130
14.1 Features 130
14.2 Configuration of recordings 130
14.3 Event logging 137
14.4 Data Logger and USB recording 138
14.5 USB recording 138

15 Ambient Light Sensor 143

16 Lap Trigger 145
16.1 Lap trigger (timing beacon) 145
16.2 Counting outing/laps/fragments 151
16.3 Lap timing 152
16.4 Segment timing 154
16.5 Countdown timer 156
16.6 Automatic GPS Track Detection 157
16.7 Predated Laptime 158

17 GPS Sensor 160
17.1 GPS (Global Positioning System) 160
17.2 Protocol 160
17.3 Sensor recommendation 161
17.4 Measurement labels 163
17.5 GPS troubleshooting 164

18 Telemetry System LTE 65 166
18.1 Software setup 166
18.2 Telemetry channels with special functionality 167
18.3 Setting up car in WDServer 168
18.4 Loading the telemetry data 172

19 Firmware 173
19.1 Firmware and configuration 173
19.2 Firmware update 173

20 Cloning the Unit 176

21 Fuel Consumption Calculation 178
21.1 Setting up fuel consumption calculation and tank management 178

21.2 Fuel consumption diagnosis/counter reset 179

22 RaceCon Shortcuts 181

23 Legal 182
23.1 Legal Restrictions of Sale 182
23.2 Open Source Software (OSS) Declaration for the Display 182
23.3 REACH Statement 197

24 Disposal 198

25 Offer Drawing 199

1 Preparation

Use the DDU 11 only as intended in this manual. Any maintenance or repair must be performed by authorized and qualified personnel approved by Bosch Motorsport.

Operation of the DDU 11 is only certified with the combinations and accessories that are specified in this manual. The use of variant combinations, accessories and other devices outside the scope of this manual are only permitted when they have been determined to be compliant from a performance and safety standpoint by a representative from Bosch Motorsport.

Read the manual carefully and follow the application hints step by step. Do not hesitate to contact us, contact data can be found on the last page of this document.

Important information on Electromagnetic Conformity

To avoid unwanted interference with the environment (people, animals, electronic devices) or unwanted harm to the environment, it is mandatory that the user of the DDU 11 carries out an appropriate analysis to determine the electromagnetic interaction the DDU 11 may have with its individual installation environment.

Disclaimer

Due to continuous enhancements, we reserve the rights to change any illustrations, photos, and technical data within this manual.

Please retain this manual for your records.

Note

In this document, many screenshots are created by way of example for a display. Please consider this and replace the product names with the name of your device.

2 Warnings and Safety Instructions

The classification of the warnings and safety instructions is carried out by the respective signal word (Danger, Warning, Caution) next to the warning symbol.

Danger

⚠ DANGER

Nature and source of danger

Consequences

Warning of death or serious physical injury, which are sure to occur if ignored.

Warning

⚠ WARNING

Nature and source of danger

Consequences

Warning of death or serious injury, which can occur if this is not observed.

Caution

⚠ CAUTION

Nature and source of danger

Consequences

Warning of slight bodily injury in case of Disregard.

Notice

NOTICE

Nature and source of danger

Consequences

Warning of damage to equipment in case of ignoring.

3 Onboard Network Concept

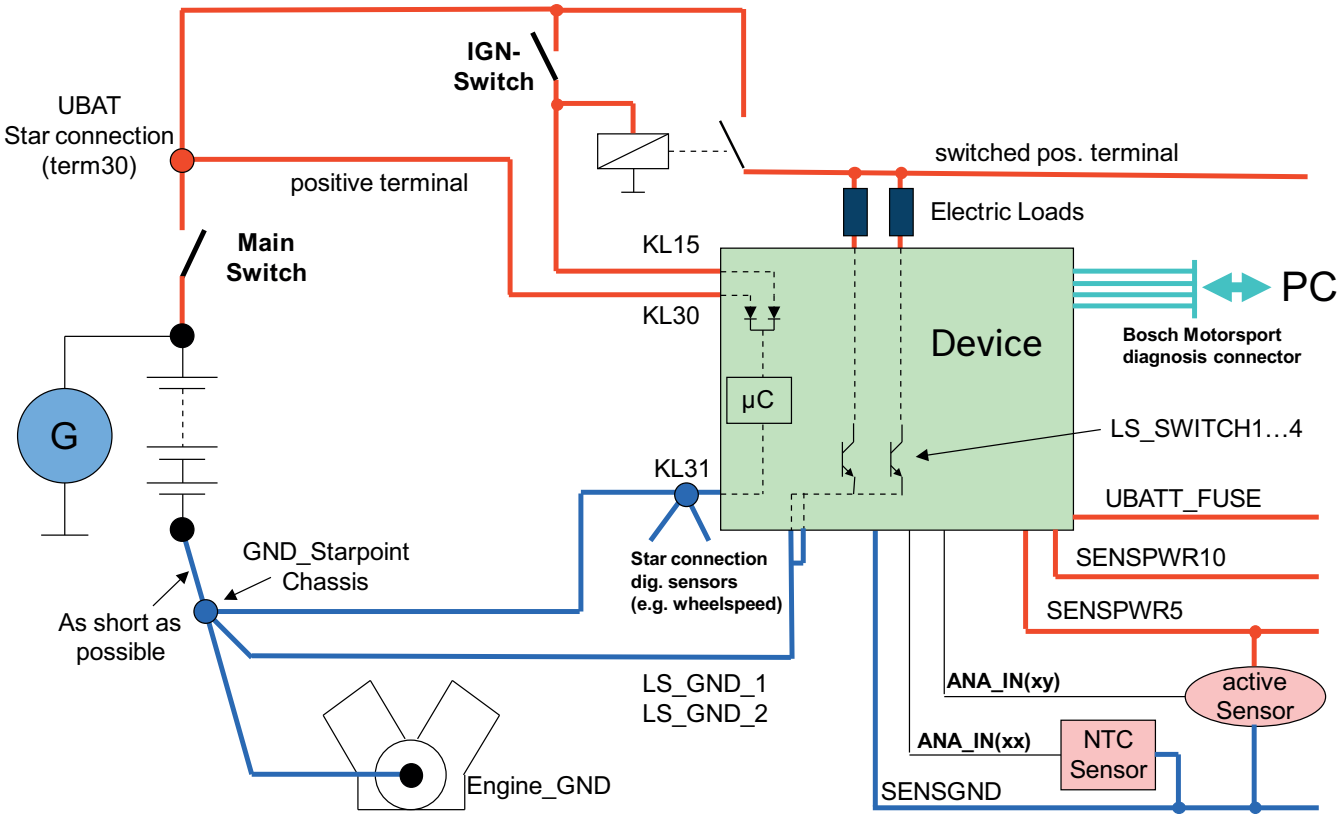
Please ensure that you have a good ground installation. That means:

- A ground that has a solid, low resistance connection to the negative battery terminal
- Connection should be free from dirt, grease, paint, anodizing, etc.
- Use large diameter wire
- More metal-to-metal contact is better!

The following notations for power signals are used:

- KL 15 is a switched battery rail controlled by the IGN-switch
- KL 30 is an unswitched battery positive rail (same as battery positive terminal)
- KL 31 is an unswitched ground rail (same as battery negative terminal)

Be careful to observe current limits of wires and connector pins!



Note

This schematic is not device specific. Please see the section Technical Data for the specifications of your device.

4 Technical Data

The display DDU 11 integrates a programmable full colour dashboard display with a data logging system for motorsport applications. Additional input devices can be connected via Ethernet, CAN buses, and RS 232.

A basic logging function of 100 channels with recording rate of 20 Hz (50 ms) (4 GB) is always included. The logger can be upgraded to full logging performance (max. 1 kHz (1 ms)). In addition, a 2nd logging partition of 4 GB can be activated. An optional external GPS Sensor keeps track of position on the racetrack.

With the DDU 11, a library of graphical elements for the individual design of display pages and a state-of-the-art user interface menu is available. For quick data transfer from the car, e.g., during pit stop, data logging to a USB stick is available as an option.

Application

Display	
6.5" graphic color display	
48 user configurable pages (12 pages organized in 4 switchable page sets)	
Resolution	800 x 480 pixel
Supported image file formats	PNG, BMP, JPG, GIF
Processor	667 MHz Dual Core
AD-Converters	10 kHz sampling rate with digital down-sampling filter
Real time clock supply	Ultracap™, sustains realtime for 2 weeks
Configurable math channels	
User configurable CAN in/out messages	Up to 256 IDs (128 in and out)
Sampling rate	20 Hz (50 ms) standard, max. 1 kHz (1 ms) as optional upgrade (FULL_LOG_1/FULL_LOG_2)
Online data compression	
Internal logger	
– 100 channels (standard)	
– Up to 1,500 channels (optional)	
– 4 GB memory (standard)	
– Plus 4 GB memory (optional)	
Logging rates	
– Usage of all features: 600 kB/s	
– Primary logging use case: >800 kB/s	
– Logging data download rate: up to 6.2 MB/s	
Ambient light sensor	

Mechanical Data

Size	203 x 128 x 38.5 mm
Weight	649 g
Protection classification	IP67

Operating temperature internal	-20 to 85°C
Max. vibration	Vibration profile 1 (see bosch-motorsport.com)

Electrical Data

Supply voltage	6 to 18 V
Current consumption (w/o sensor supply)	<2 A (at 12 V)

Outputs

1 x Sensor supply 5 V +/-1 % or UBAT switched (max. 250 mA)
1 x Sensor ground

Inputs (option)

4 x Analog In (5 V/12 bit), Digital In, Wheelspeed HALL, or Wheelspeed DF11
1 x Analog In (5 V/12 bit), Digital In or Laptrigger In
Switchable 3.01 kOhm pull-up resistors for all analog inputs

Connectors and Wires

Mating Connector: TE Connectivity 3-1437290-7
Connector Pins: TE Connectivity 3-1447221-4

Pin Configuration

Pin	Name	Status
1	KL30_KL15/UBAT	Incl.
2	SENSGND	Incl.
3	SENSPWR (5 V or UBAT switched)	Incl.
4	SIG2_IN (Analog In, Digital In, Wheelspeed HALL, Wheelspeed DF11, switchable Pull-Up 3.01 k)	Opt.
5	SIG4_IN (Analog In, Digital In, Wheelspeed HALL, Wheelspeed DF11, switchable Pull-Up 3.01 k)	Opt.
6	USB_GND	Opt.
7	USB_PWR	Opt.
8	KL31/GND	Incl.
9	SIG5_IN_LAPTRG (Analog In, Digital In, switchable Pull-Up 3.01 k)	Opt.
10	SIG1_IN (Analog In, Digital In, Wheelspeed HALL, Wheelspeed DF11, switchable Pull-Up 3.01 k)	Opt.
11	SIG3_IN (Analog In, Digital In, Wheelspeed HALL, Wheelspeed DF11, switchable Pull-Up 3.01 k)	Opt.
12	USB_DN	Opt.
13	USB_DP	Opt.
14	CAN_2_H_RS232_TX (switchable CAN or RS232)	Opt.
15	CAN_1_H	Incl.
16	ETH_CHANNEL0_TXN (Ethernet 100BaseTX)	Incl.

17	ETH_CHANNEL0_TXP	Incl.
18	ETH_CHANNEL0_RXN	Incl.
19	ETH_CHANNEL0_RXP	Incl.
20	TIMESTMP_INOUT (Timesync)	Incl.
21	CAN_2_L_RS232_RX (switchable CAN or RS232)	Opt.
22	CAN_1_L	Incl.
23	ETH_CHANNEL1_TXN (Ethernet 100BaseTX)	Incl.
24	ETH_CHANNEL1_TXP	Incl.
25	ETH_CHANNEL1_RXN	Incl.
26	ETH_CHANNEL1_RXP	Incl.

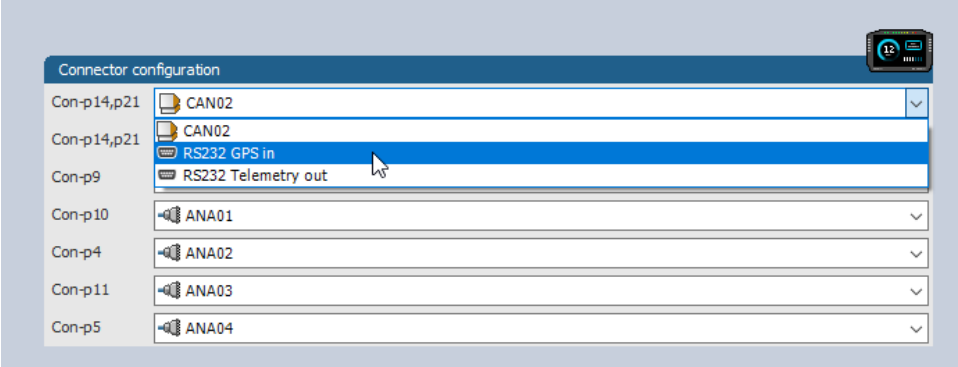
Mating Connector Kit AMP Superseal	F02U.B01.201-01

Communication

CAN interfaces	1
2nd CAN interface or RS232 with IO_EXTENS	e.g., Telemetry or GPS
Ethernet 100BaseTX	2
Laptrigger input	1
Configuration via RaceCon	Over Ethernet or MSA-Box II
CCP-Master, data acquisition from ECU that support CAN Calibration Protocol (CCP_MASTER option)	
XCP-Master, data acquisition from ECU that support eXtended Calibration Protocol over UDP or CAN (CCP_MASTER option)	

4.1 Configurable Pins

The DDU 11 has a set of configurable pins, that can provide different functionality. To select the different options, double click on the DDU 11 in project tree and select the sheet 'Configurable pins'.



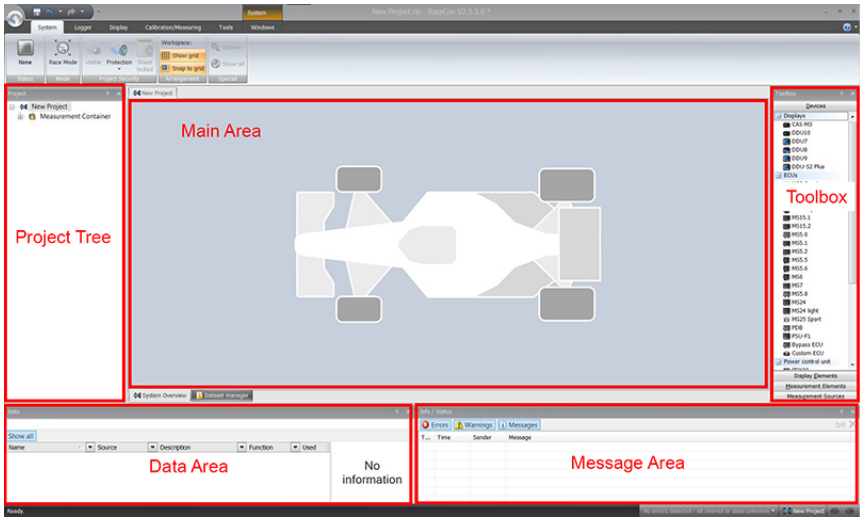
The figure above shows an example pin configuration. The labels to the left of the dropdowns refer to the pin number of the physical connector. Labels to the right add additional information on the functionality. Each dropdown entry will configure the functionality provided by this specific pin. Some functionalities require multiple pins, like switching between CAN or RS232 interface. A change to such an entry automatically affects all relevant pins.

The settings applied in ‘Configurable pins’ also change visibility in the project tree. For example, if ANA04 is changed to REV04, then ANA04 will get invisible in the project tree and REV04 will become visible for further configuration.

4.2 Upgrades

CCP/XCP_MASTER
Enables CCP/XCP master functionality to request data from foreign devices via CAN/CCP protocol, XCP over Ethernet (UDP) or XCP over CAN. (ASAP2 file from ECU manufacturer required)
FULL_LOG_1
Enable full logging performance with sampling rate of max. 1 kHz (1 ms) and 1,500 channels for partition 1
FULL_LOG_2
Enable logging partition 2 with 4 GB
IO_EXTENS
4 x Analog In (5 V/12 bit), Digital In, Wheelspeed HALL or Wheelspeed DF11
1 x Analog In (5 V/12 bit), Digital In or Laptrigger In
Switchable 3.01 kOhm pull-up resistors for all analog inputs
1 x CAN or RS232 (switchable via RaceCon)
USB_DATA includes
Rugged USB flash drive Bosch File System (BFS) format included, works with Bosch File System (BFS) preformatted USB Flash drive only
Adapter cable to PC USB-Port
Mating connector for USB flash drive on car loom side
Software license for USB-Logging
External switch for page selection, 12 steps B261.209.658-01
External switch for brightness adjustment or page selection, 6 steps B261.209.659-01

5 System Configuration Tool RaceCon



RaceCon is an all integrated software tool for configuration and calibration of Bosch Motorsport hardware products, such as ECUs, displays, loggers. The communication is based on Bosch Motorsport MSA-Box interface.

Calibration of ECU maps and curves
ECU data file up- and download
Parameter file up- and download
Diagnostic functionality for Bosch Motorsport ECUs
Data file / Work base management
Integrated flash functionality
Integrated Bosch sensor database
Configuration of Bosch Motorsport displays
Configuration of Bosch Motorsport data loggers
Configuration of Bosch Motorsport CAN modules
Communication via K-Line/CAN/Ethernet (KWP/CCP/XCP)
CAN communication log functionality (baud rate changeable)
Quick data access over Race Mode
Intuitive design, easy to use

Environment

PC
IBM PC Pentium/AMD Athlon compatible, min. 1.6 GHz
Min. 2 GB RAM
Min. 1 GB free hard disc space
VGA/WGA monitor (min. 1,024 x 768)
Recommended Operating System: Windows 10

Optional Accessories

MSA-Box II	F02U.V00.327-03

6 First Steps

Install the software required for the operation of the DDU 11. It is developed for Windows system software. The following software versions are used in this manual:

- DDU 11 setup, configuration and calibration: RaceCon Version 2.11.
- Measurement data analysis: WinDarab V7

Set up the 100 Mbit ethernet connection to the DDU 11.

- The ethernet port has “cable auto crossover” functionality.

6.1 Starting the unit

The DDU 11 powers up by turning on the ignition of the car. At startup the DDU 11 will display a Bosch logo.

After a moment, the DDU 11 shows a display element screen.

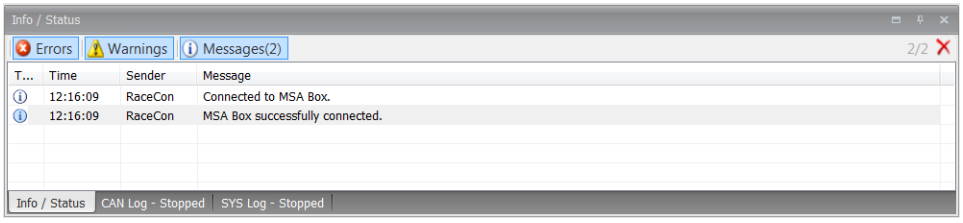
6.2 Connecting the unit to RaceCon

For testing new device configurations, you can connect the device to your computer via MSA-Box or ethernet cable.

Connection via MSA-Box

1. Reassure that the MSA-Box driver is installed properly on your computer. If needed, download the MSA-Box driver from www.bosch-motorsport.com.
2. Connect an ethernet line of the device to the ethernet line of the MSA-Box.

   Please note, that the MSA-Box also requires power supply on the MSA-Box connector of your wiring loom.
3. Open RaceCon and connect the MSA-Box to the computer.
4. In the ‘Info / Status’ Box of RaceCon, you will receive messages that the connection was successful.



5. Reassure that the device is switched on.
6. ‘Link LED’ at the computer’s network adapter will illuminate.

   If the LED is off, check the wiring harness.

   After you created a RaceCon project with the device, the status icon of the device will switch from grey to one of the following colors: red, orange, green. For further information on how to set up a project, see the chapter “Setting up a new RaceCon Project [▶ 14]”. For the status color, see chapter “Color indication [▶ 27]”.

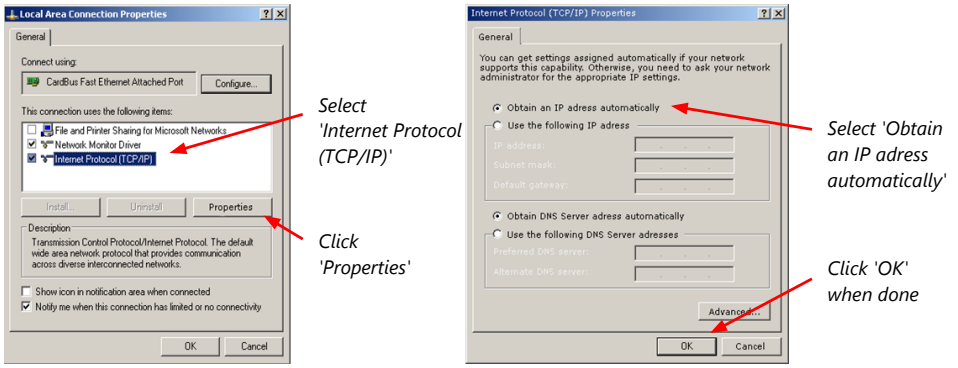
Connection via Ethernet Cable

Instead of connecting the ethernet line to the MSA-Box, connect the ethernet directly to your computer.

Troubleshooting while setting up the network interface

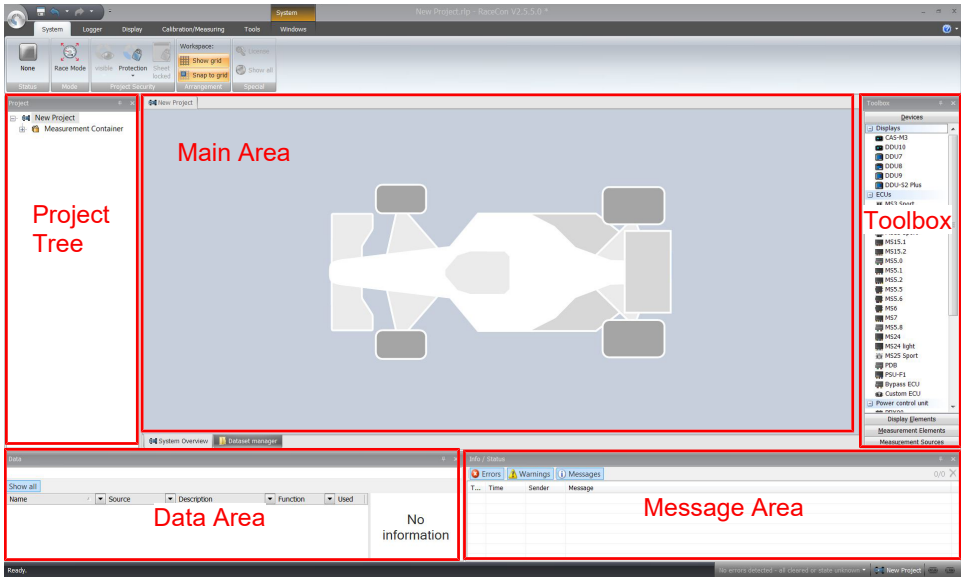
The DDU 11 contains a DHCP server, network addresses can be assigned automatically to the configuration PC. In case of problems during the network connection, please try the following steps:

1. Switch off the PC's firewall.
2. Reconfigure the PC or the MSA-Box network interface settings to obtain an IP address automatically as shown in the pictures below.

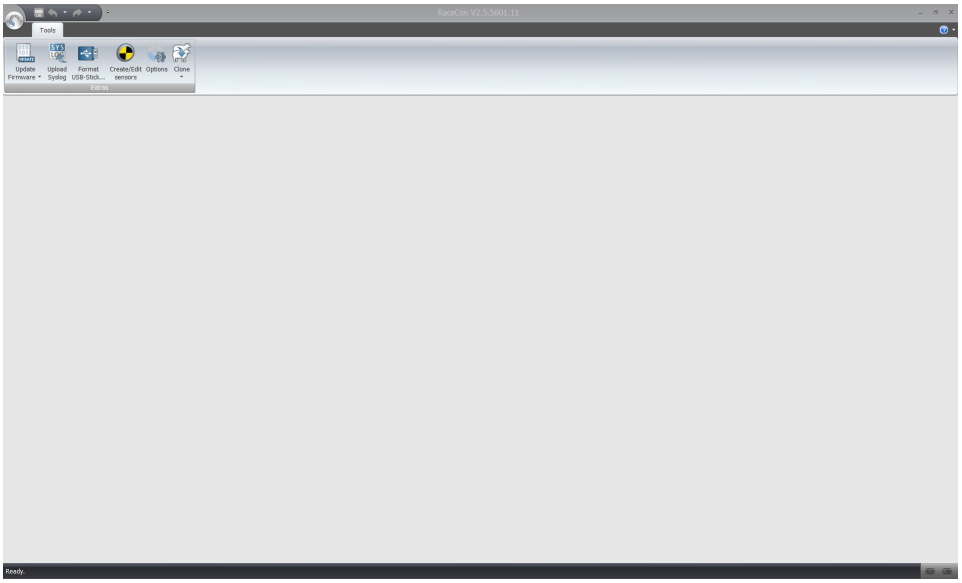


6.3 Setting up a new RaceCon Project

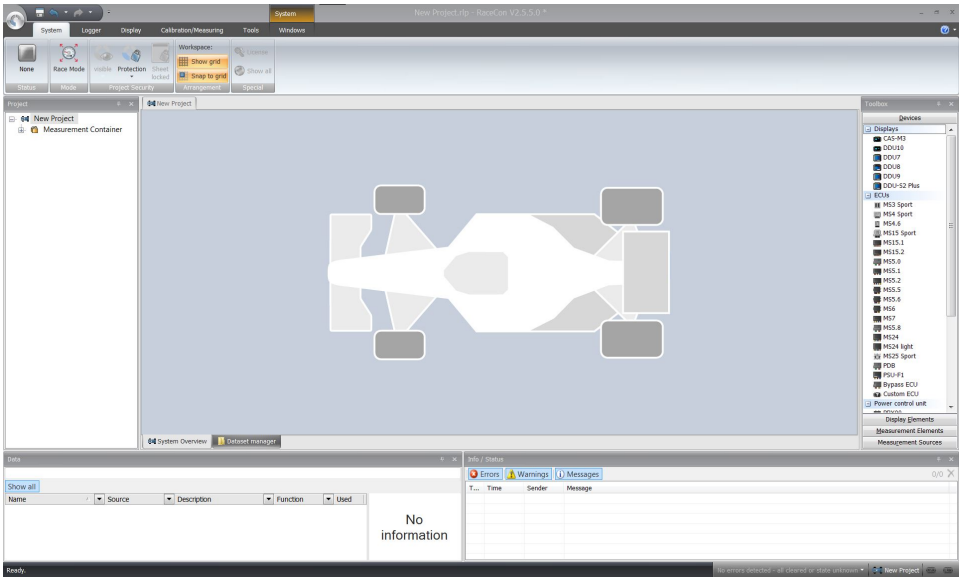
The following screenshot shows an overview of the RaceCon Main Screen with its areas. All (sub-) windows are resizable and dockable. You can find them under the 'Windows' tab.



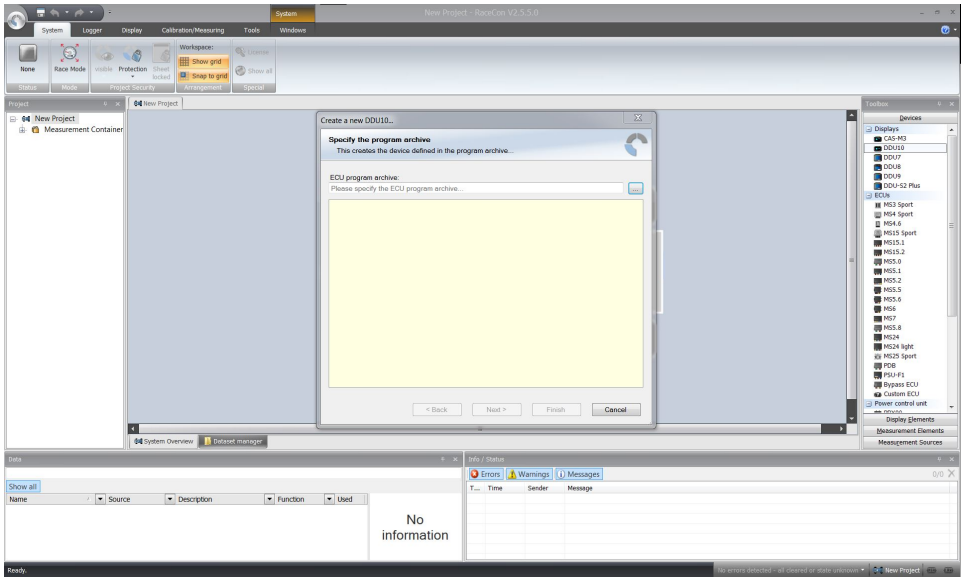
1. Start the RaceCon software.



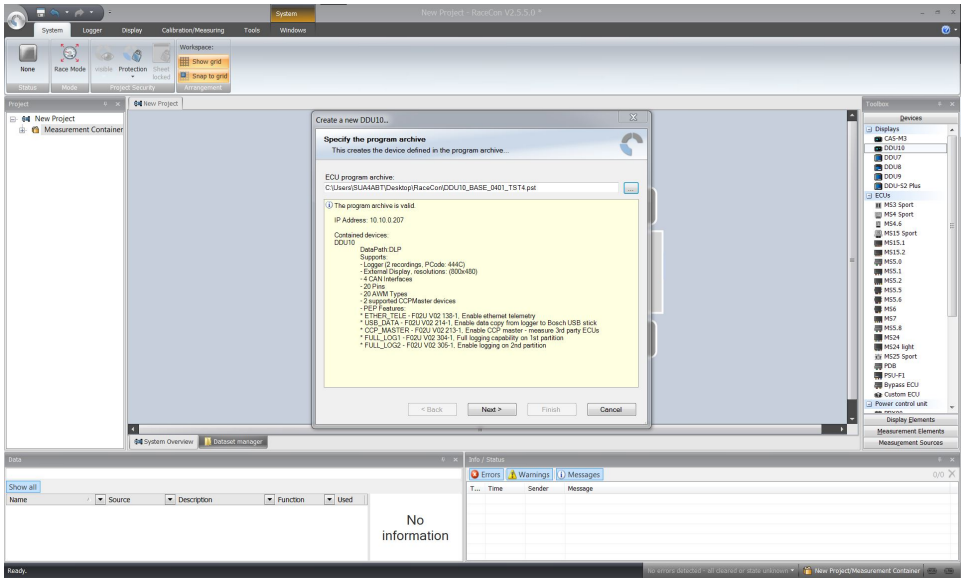
2. In the ‘File’ menu, select ‘New project’ to create a new project.



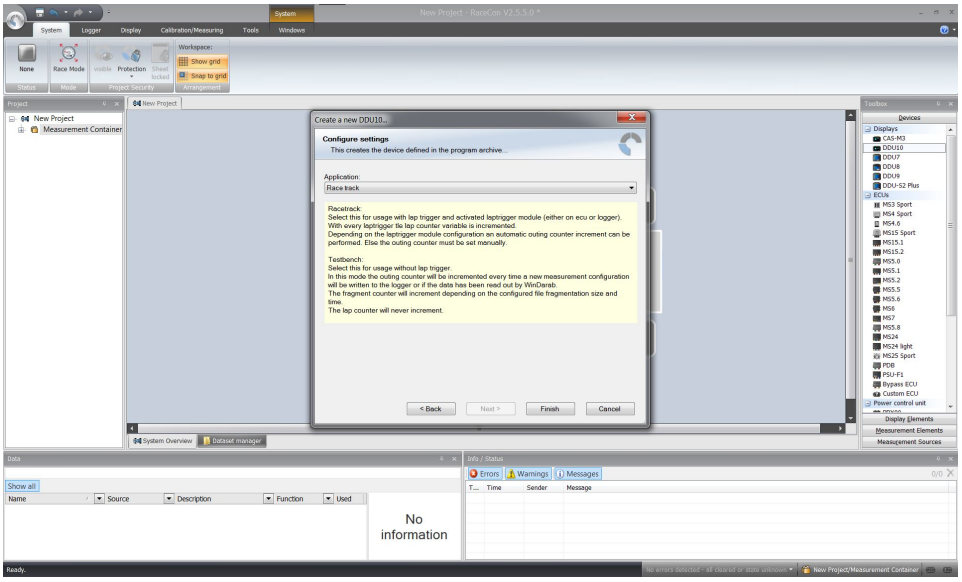
3. In the Toolbox, select the DDU11 and drag it into the Main Area. A pop up window to specify the DDU11 program archive appears.



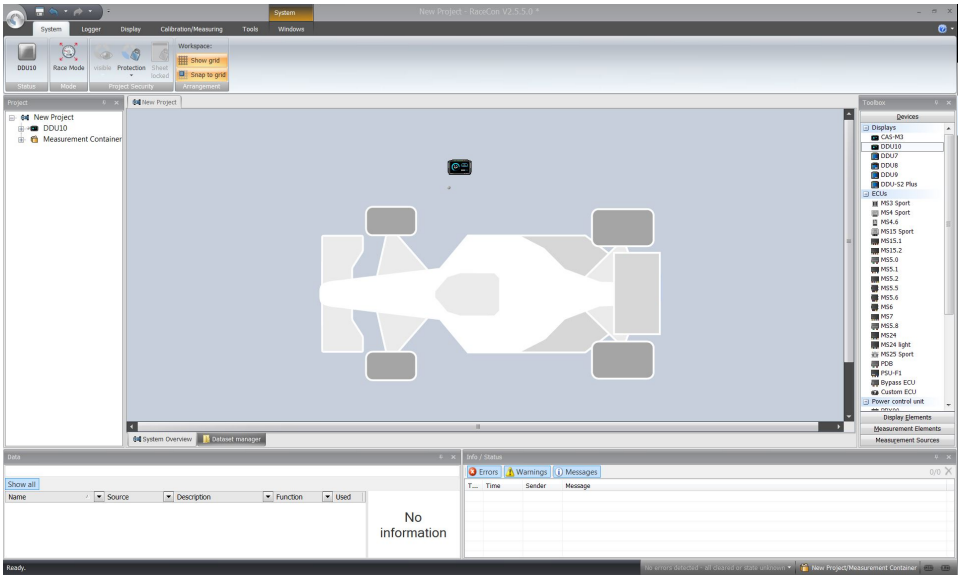
4. Download the firmware for the DDU 11 from www.bosch-motorsport.com. An information shows if the archive is valid or not.
   Click ‘Next’.



5. Select ‘Race track’ or ‘Testbench’ mode according to your application.

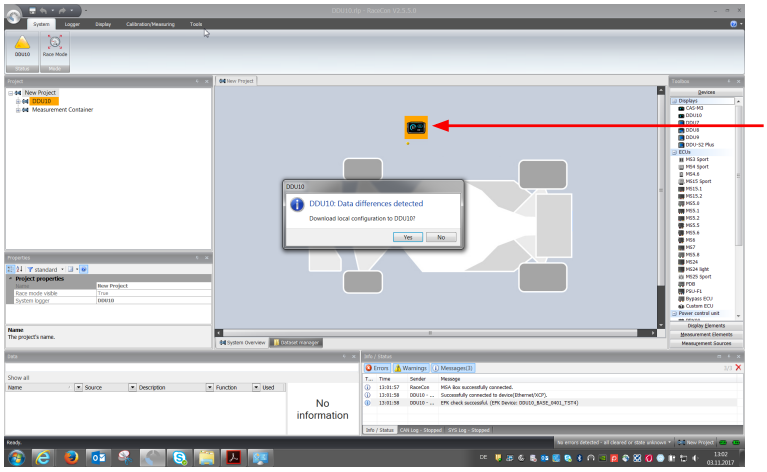


6. Click ‘Finish’. The DDU 11 is inserted into the project and RaceCon tries to connect to the device.



RaceCon detects configuration differences between the DDU 11 and the RaceCon project and asks for permission for data download.

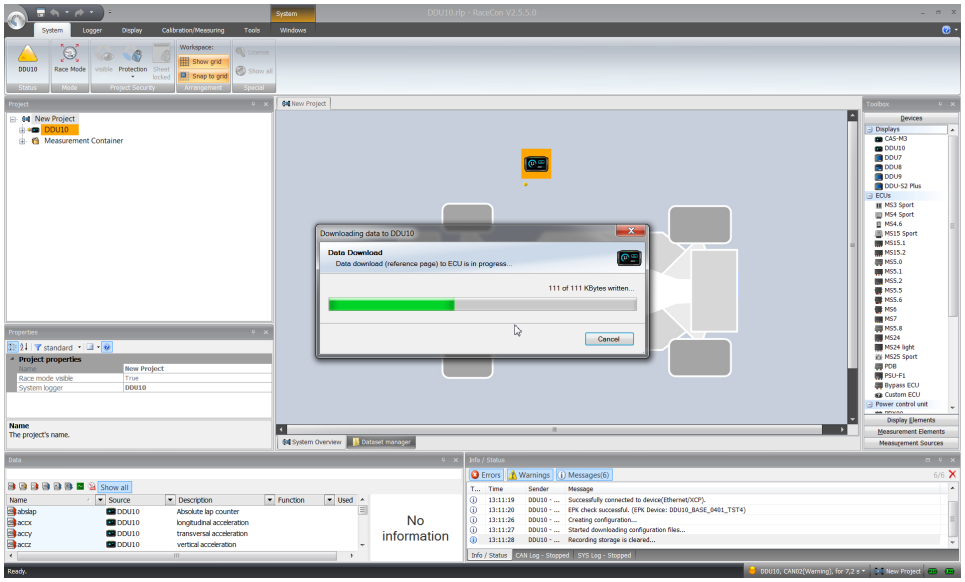
Click ‘Yes’ to download the configurations to the device or ‘No’ to continue without downloading the data.



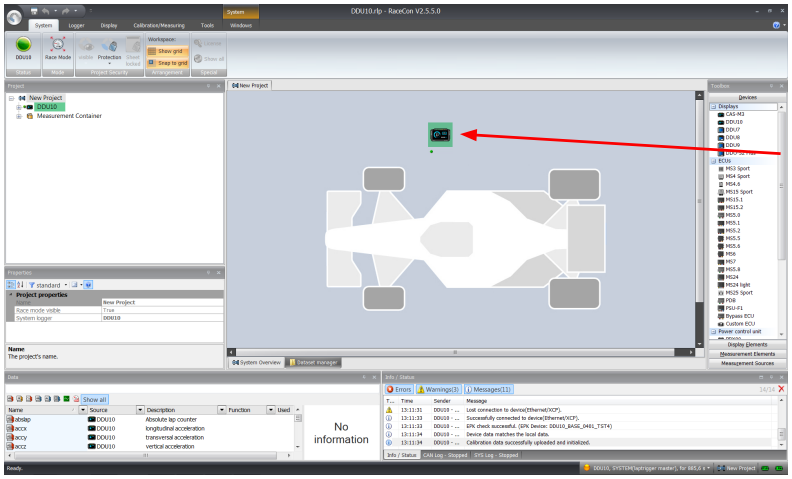
Successful ethernet connection, Device "talks" to PC.

If the device turns red, you might need to do a firmware update on the device. For more information see chapter “Firmware update [▶ 173]”.

The download starts and the DDU 11 carries out a reset.



After the reset, RaceCon reconnects to the DDU 11. Local configuration on both the PC and DDU 11 match (indicated by green background and dot). The DDU 11 is now connected to RaceCon.



Green background and dot indicate matching configuration

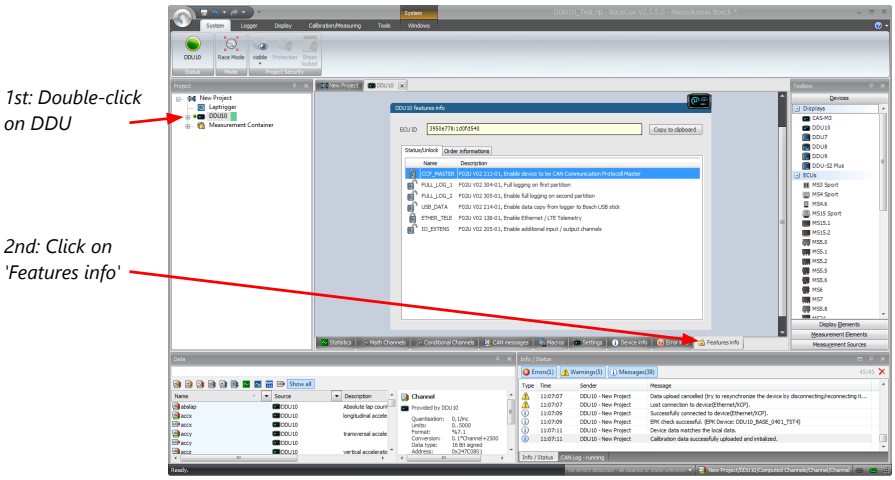
For further information on the color indication, see chapter “Color indication [▶ 27]”.

6.4 Feature activation

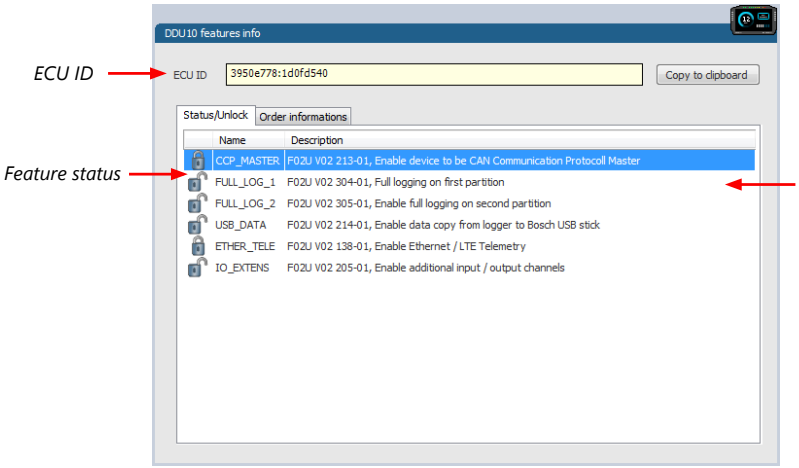
- Optional software feature packages are available for the DDU 11
- All software feature packages can be purchased prior to delivery or after you have received your device.
- If you have purchased an optional software feature package, it must be activated before it becomes operational.
- The feature activation status is stored permanently in the device and requires activating once only.
- As the activation key is device specific, a key delivered with one DDU 11 does not work on any other DDU 11.
- When purchasing a software feature package, you have to tell Bosch the ECU ID code.

  The ECU ID code is device specific and can be found in the 'features info' window, shown in the screenshots below.
- If you have not purchased an optional software feature package, the next steps can be skipped.

1. Ensure a connection to the device.
2. To activate a feature, double-click on 'DDU 11' in the Project Tree.
3. Click on the 'Features info' tab in the Main Area.

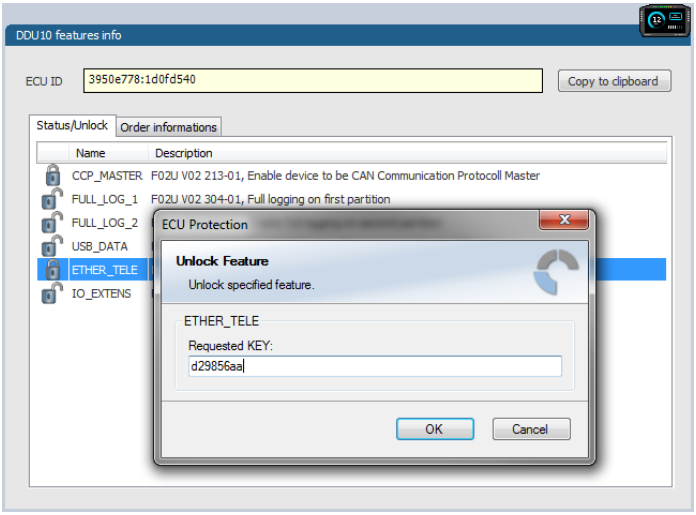


4. The 'DDU 11 features info' window appears.

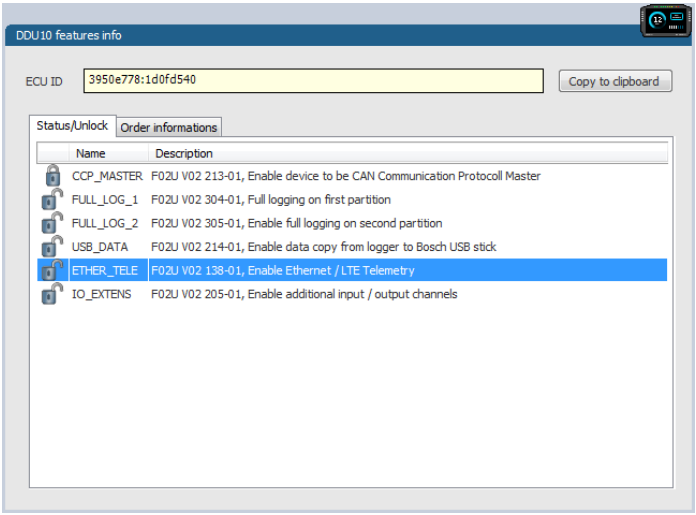


Locked (disabled) *Unlocked (activated)*

5. Double-click on the feature you want to activate. A feature unlock window appears.



6. Enter the activation key you received for this feature on this device and click 'OK' when done. The feature's status changes to 'unlocked'.



7. Perform these steps to activate other features you purchased.
8. Switch the car's ignition off and on again to cycle the power of DDU 11.

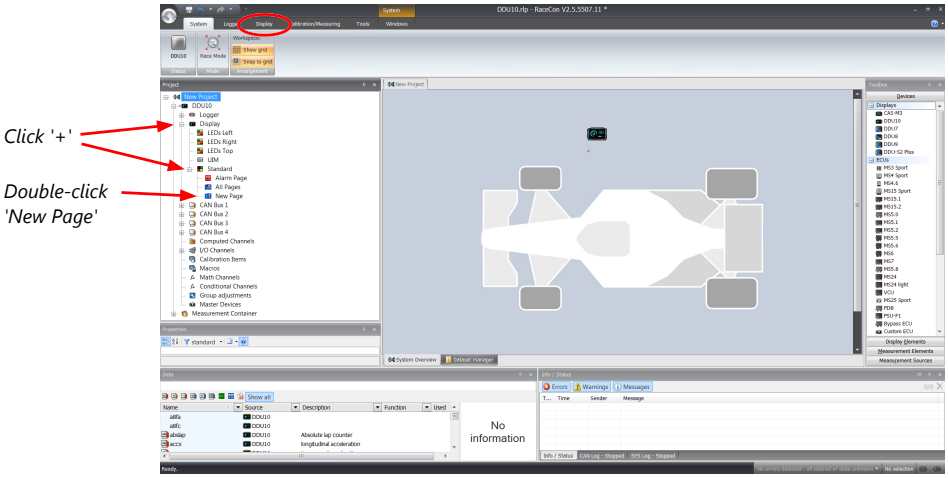
6.5 First display configuration (Quick Start)

NOTICE:

The screenshots shown in this document may show a different device. However, the procedure is identical for all devices.

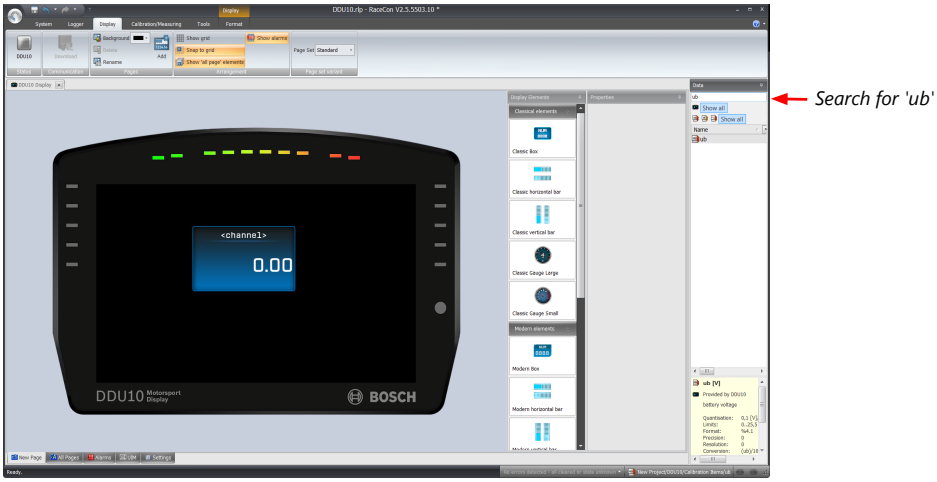
This chapter explains the configuration of a display element showing the battery voltage. See chapter Element Configuration [▶ 43] for a detailed instruction to configure display elements.

1. Click on '+' to expand the DDU 11 project tree.
2. Click on '+' to expand 'Display'.
3. Double-click on 'New Page', or click on the 'Display' tab. RaceCon changes to the page 'Display' to open the DDU 11 display configuration area.

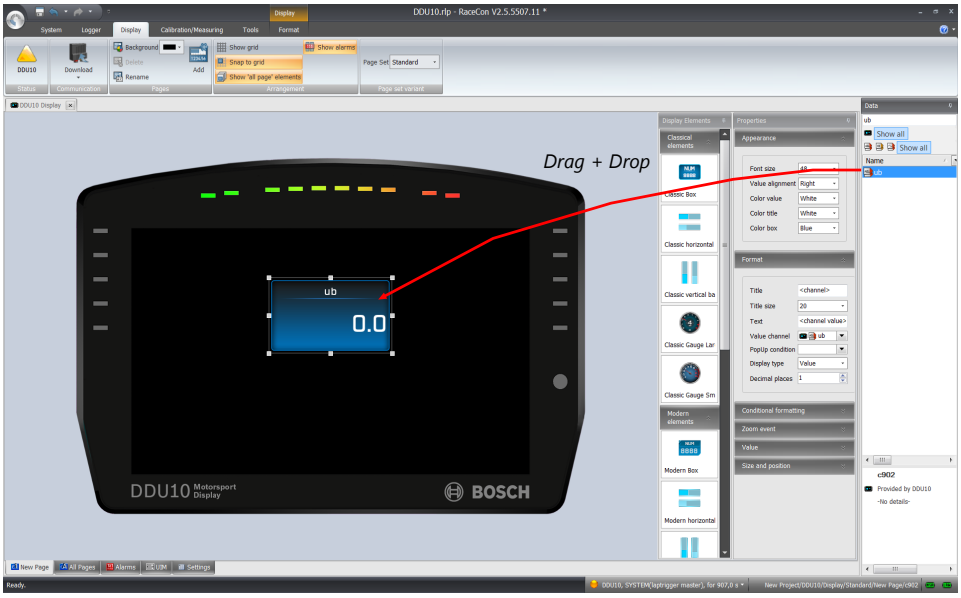


4. Drag any display element from the Toolbox and drop it on the display page. The status signal in the upper left corner switches from green to orange, because the configuration in the tool differs from the configuration of the device.

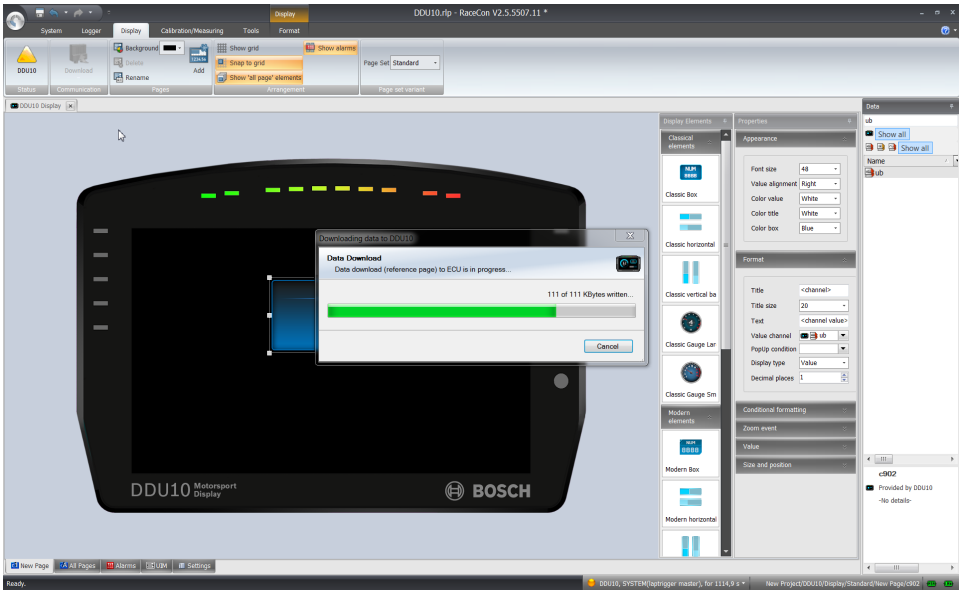
5. Use the search bar in the 'Data' window, to search for 'ub' (measurement channel for battery voltage).



6. Drag the ‘ub’ measurement channel from the ‘Data’ window and drop it on the display element.



7. Click on the ‘Download’ button in the upper left corner. The configuration download starts and the DDU 11 carries out a reset. The status signal in the upper left corner switches to green.

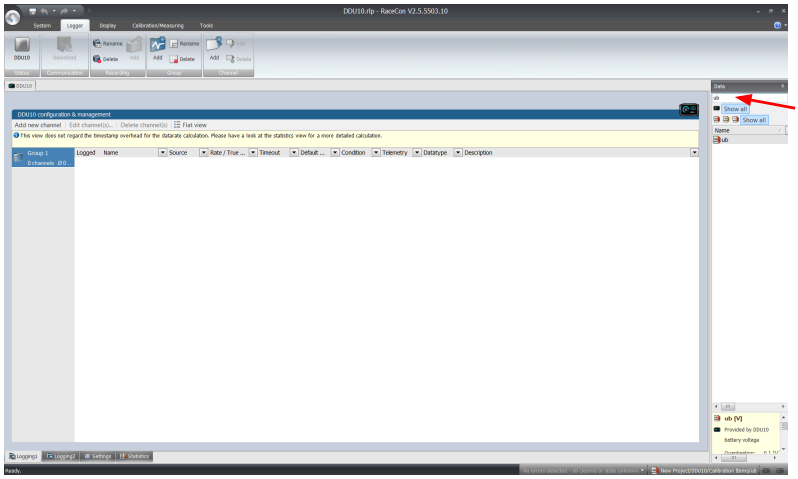


The value of the battery voltage is displayed on the DDU 11.

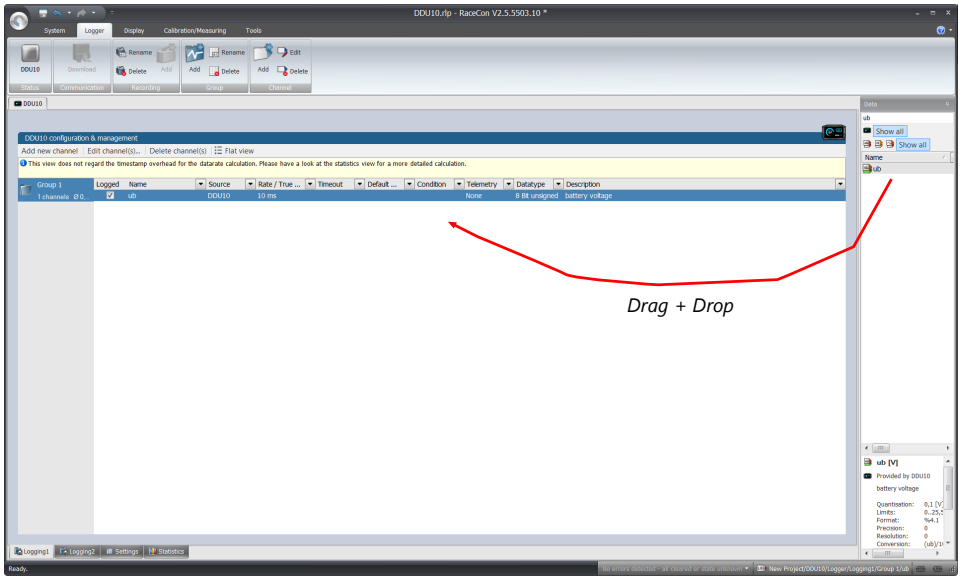
6.6 First recording (Quick Start)

This chapter explains the configuration of the recording of the battery voltage channel. See chapter 'Recording [▶ 130]' for a detailed instruction to configure recordings.

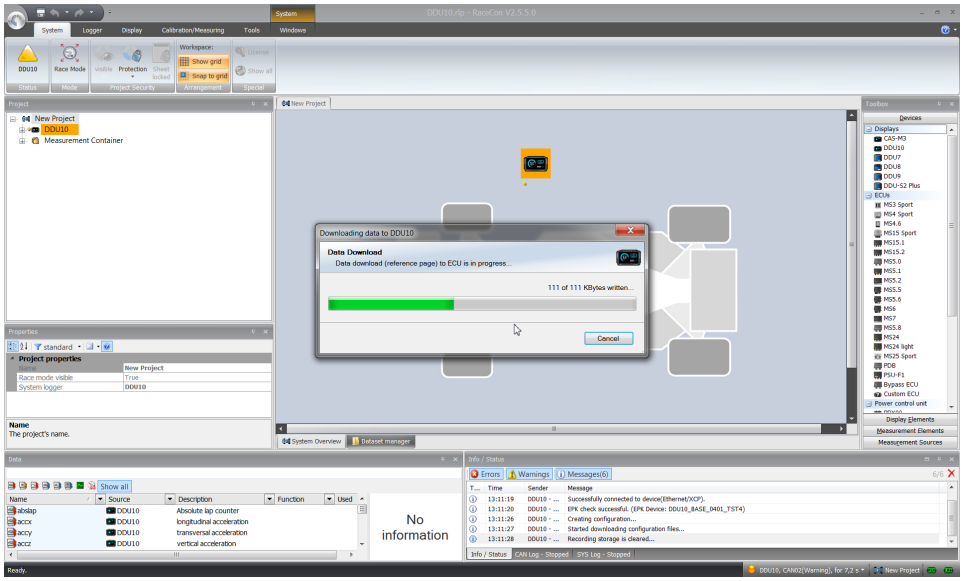
1. Click on the 'Logger' tab to go to the page 'Logger'.
2. Use the search bar in the 'Data' window, to search for 'ub' (measurement channel for battery voltage).



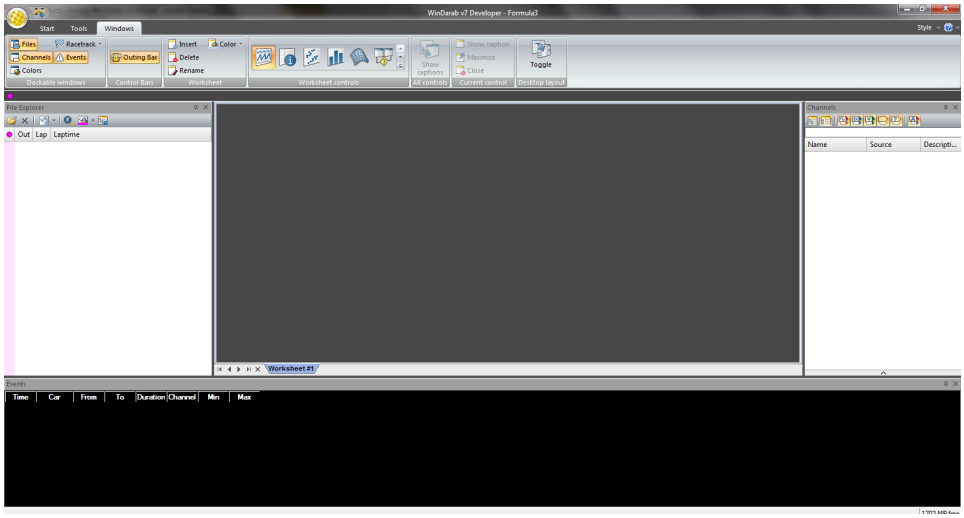
3. Drag and drop the 'ub' measurement channel into the recording area.



4. Click on the 'Download' button in the upper left corner. The configuration download starts and the DDU 11 carries out a reset. Now you can find the 'ub' measurement channel in the 'Data Area'. As we did not define global start conditions, recording starts immediately.



5. Start the WinDarab software.

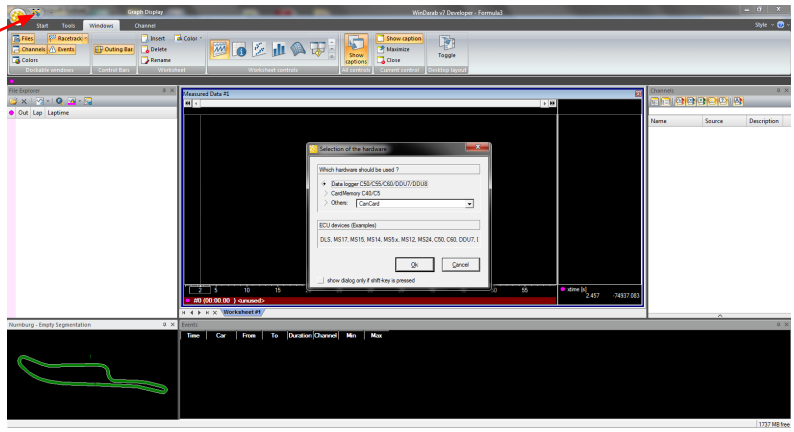


6. Disconnect the DDU 11 network cable.

7. Click on the ‘Read Data from Logging Device’ icon.

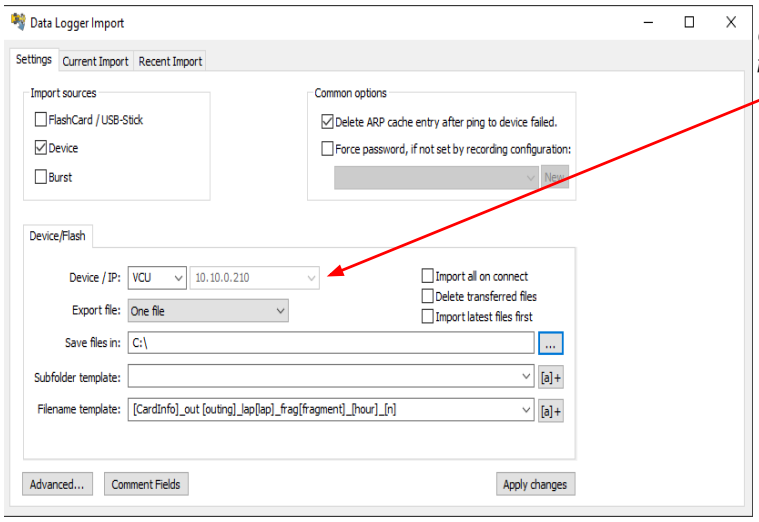
   Choose your logger and click ‘OK’ when done. The ‘Data Logger Import’ dialog opens. Refer to the WinDarab V7 manual for instructions on how to use the ‘Data Logger Import’ dialog and for more detailed descriptions and instructions.

Click 'read data from logging device'



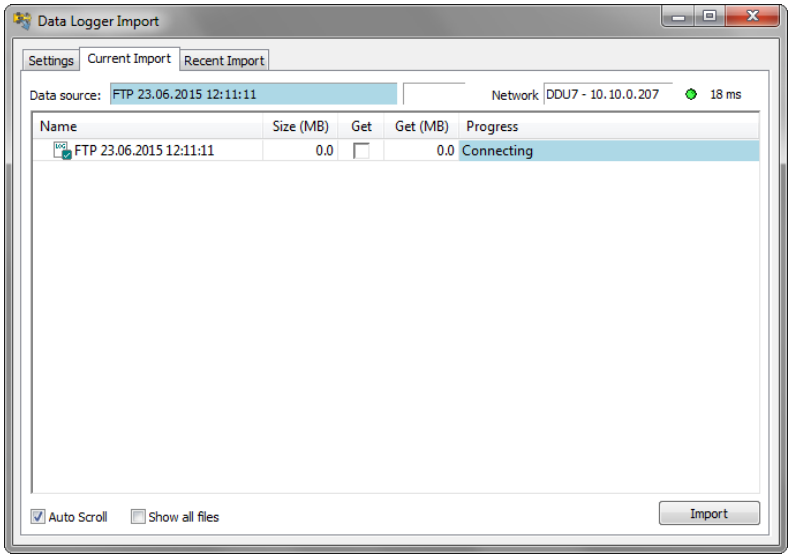
8. Choose the device and the IP address for the device.

   Click ‘Apply changes’ when done.

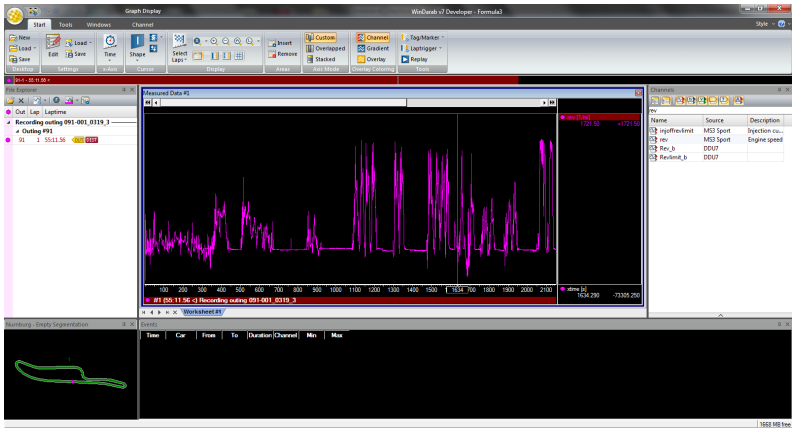


Choose your Device / IP from dropdown list

9. Connect the DDU 11 network cable.
10. Click on the ‘Current Import’ tab.
11. Click on ‘Import’ in the lower right corner. If the ‘Import all on connect’ box is checked, the data transmission from the DDU 11 starts automatically. Measurement files are stored automatically in the folder defined under ‘Settings’.



12. Click on ‘Close’ when the transmission has finished.
13. Click on the Start button and choose ‘Open measurement file’.
14. Select the measurement files from the storage folder.
15. Click on ‘Open’.
16. Click on ‘New Desktop’ to open a new measurement data window.
17. Drag the ‘ub’ measurement channel from the channel list and drop it into the measurement data window. The ‘ub’ measurement channel‘s graph is displayed.

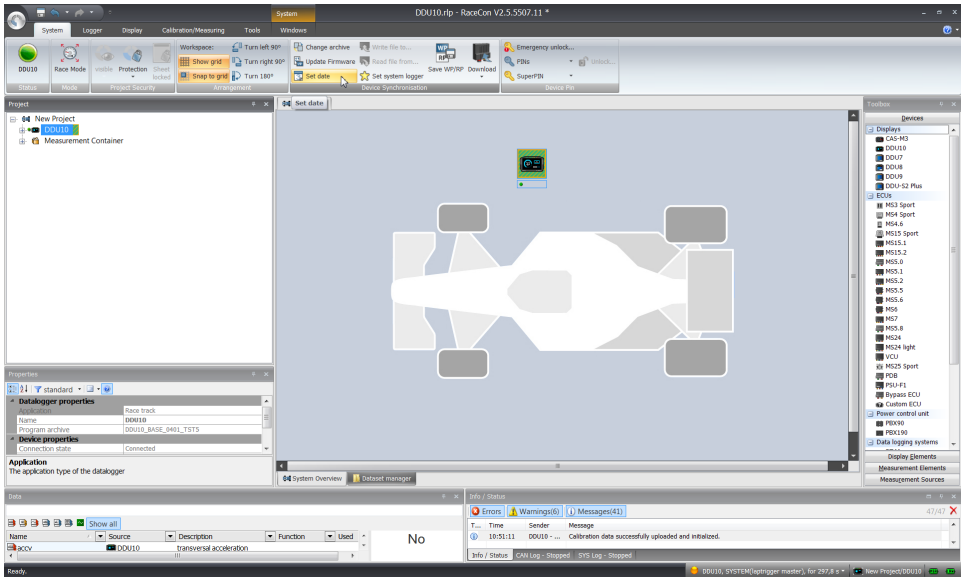


6.7 Set date and time

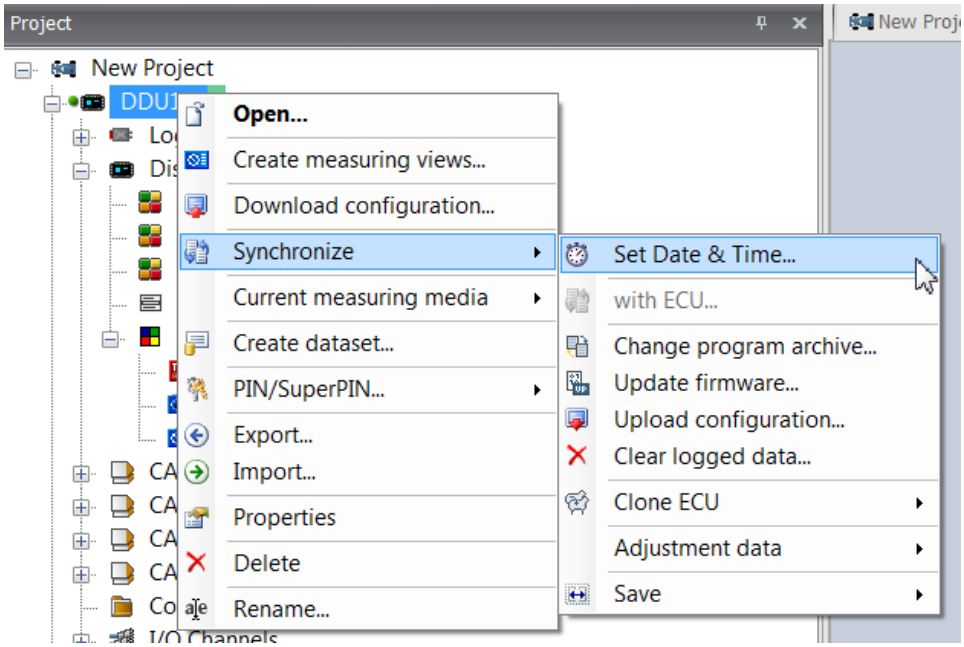
The DDU 11 is equipped with a real time clock which is supplied by an internal accumulator. Once this accumulator is charged correctly by 12 V supply of the display, ‘Date & Time’ can be programmed by RaceCon.

Reassure that the time is set correctly, if the device has not been used for more than two weeks.

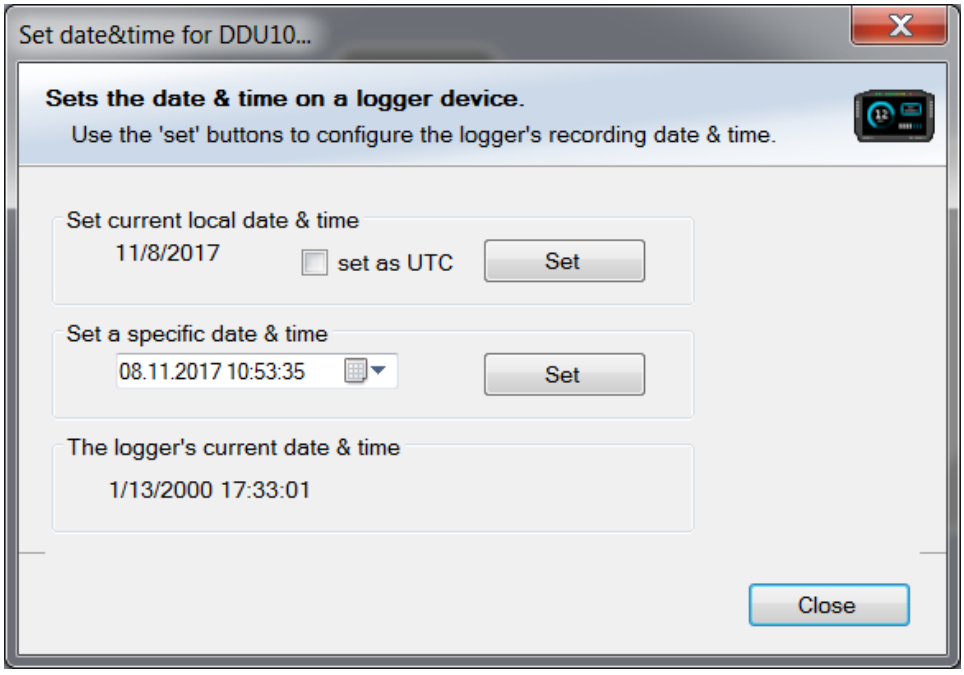
1. Connect the DDU 11 to the PC.
2. Click on the ‘Set date’ button in the ‘System’ tab menu.



3. Alternatively, click on ‘Set Date & Time’ in the context menu of the device. A ‘Set Date & Time’ menu opens



4. Set the current local date and time as coordinated universal time.
5. At 'Set a specific date & time' click and type on the value you want to change or choose from the dropdown menu.

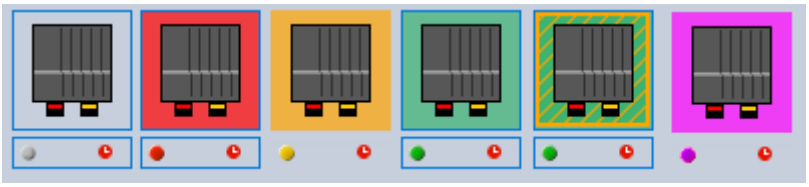


6.8 Color indication

The color indication in RaceCon visualizes different messages, such as differences between tool and device, status of the device configuration or the accrual of errors.

Visible color indications:

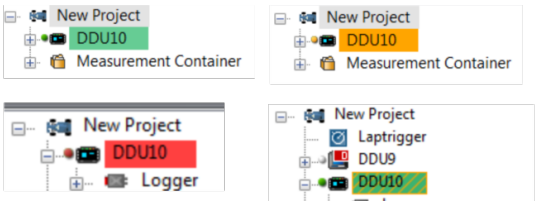
– In the status area in the upper left corner.



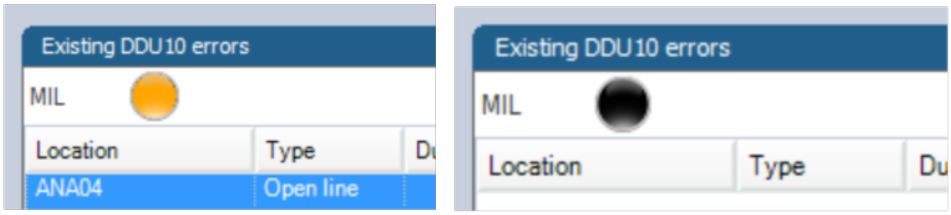
– As a background, as well as a little dot around the display icon in the ‘System window’.

– As a colored stripe beside the device name in the project tree.

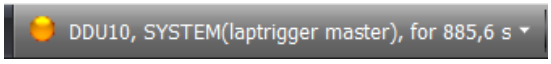
– As a colored background around the device name in the project tree.



– As a colored MIL in the “Error Info” window.



– As a colored dot in the error memory at the bottom.



The colors and their meaning:

– Grey: No connection with the device.

– Green: Matching configuration and firmware between device and project.

– Orange: A different configuration between device and project.

– Red: A different firmware between device and project.

– Purple: Device is bricked, too many resets. Reflash the device, reconsider last changes.

– Colored background with orange stripes: Matching configuration with stored (inactive) errors in the device.

– Blinking colored background with orange stripes: Matching configuration with active errors in the device.

– Black MIL: No errors.

– Orange MIL: Inactive Errors (Error entries existing, but no longer active).

– Blinking MIL (orange/black): Active Errors.

For further information, see chapter Error Memory Properties [▶ 124].

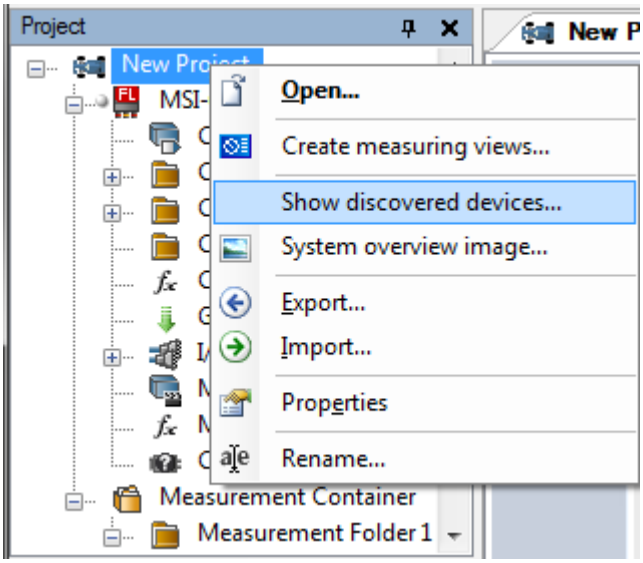
6.9 Assign the Mounting Location

At delivery the default role of the devices is set. There is no need to change this, until you want more than one device of this kind. Up to eight DDU 11 can be used additionally in one network for I/O expansion and or Multilogging, the mounting location is used for determination between the different DDU 11.

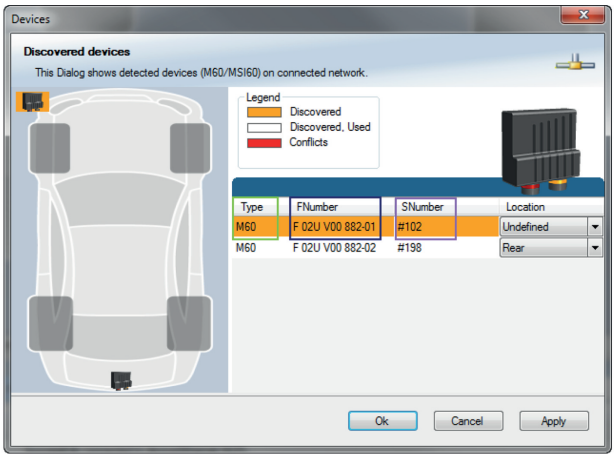
The mounting location is permanently saved in the DDU 11. If necessary, you can reassign a different mounting location at any time following the same procedure.

A mounting location must not be used several times in one network, this would disturb the functionality of the respective DDU 11.

1. In the Project Tree right click on the project name e.g. 'New Project' and then select 'Show discovered devices...'.

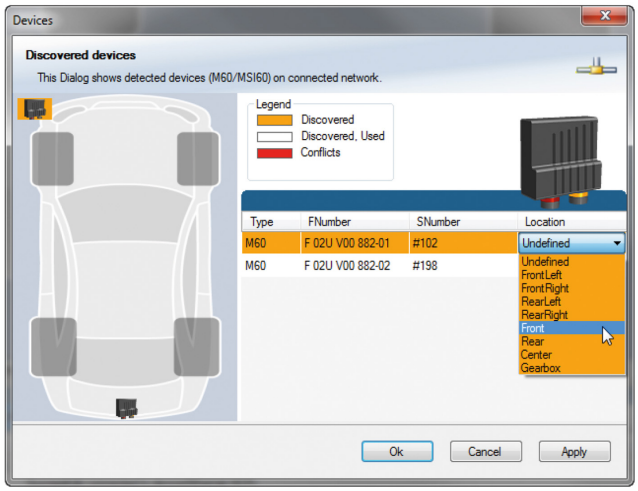


All connected DDU 11 are listed.

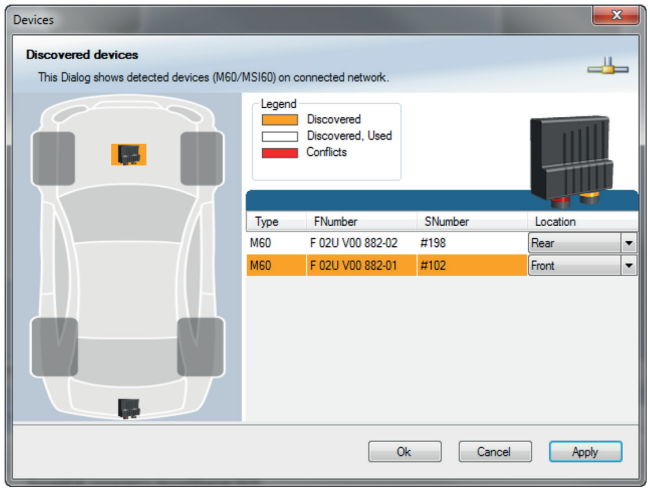


2. Compare the listed device Type, FNumber and SNumber to the identification plate to identify the device you want to make changes to:

3. Assign the desired mounting location (e.g. 'Front') and confirm by clicking 'Apply'.



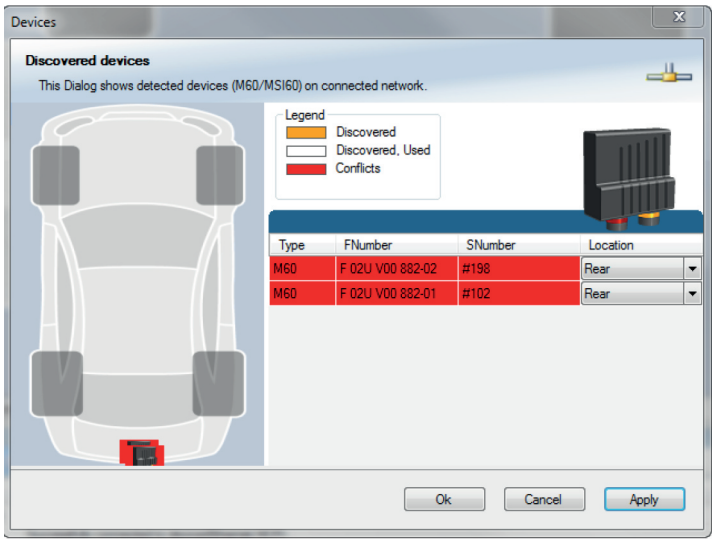
The mounting location is now stored in the device. The device will do a reset and the 'RUN' LED will change to green. The list will show the new mounting location assignment.



It is good practice to physically label the DDU 11 with its mounting location. Now the device is ready to be used.

A different coloring of the DDU 11 is used to indicate that the device is already configured in the currently loaded RaceCon project or not (white/orange).

A conflict of several connected DDU 11 using the same location is indicated by red coloring the involved devices:



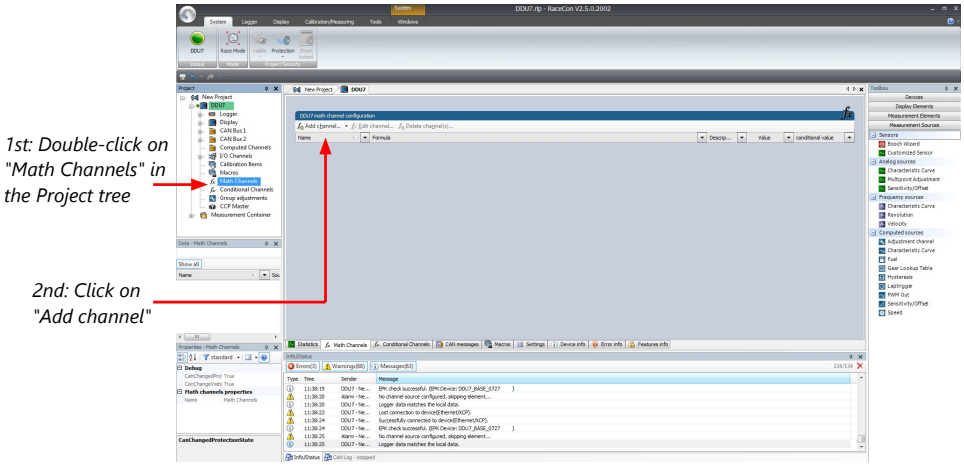
7 Project Configuration

7.1 Math Channels

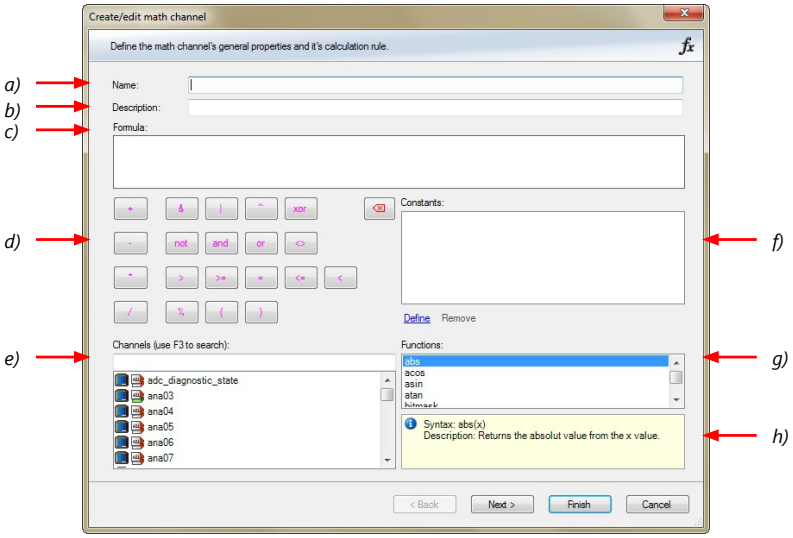
- Arithmetic and logical operations on up to 4 measurement channel(s)
- Numerical result
- Result can be used as input source for various display elements (numeric elements, alarms, bargraphs) and further calculations in the whole RaceCon project

Creating a new Math Channel

1. Follow the steps shown in the screenshot. The “Create/edit math channel” window appears.



2. Define the math channel using the following configuration possibilities:



a) Enter the name of the math channel.
b) Enter a description of the math channel.
c) Enter the formula.
d) Select the logical operator.
e) Choose a measurement channel.
f) Define a value that can be used as a constant in the formula.
g) Choose a function.
h) Describes the function selected above.

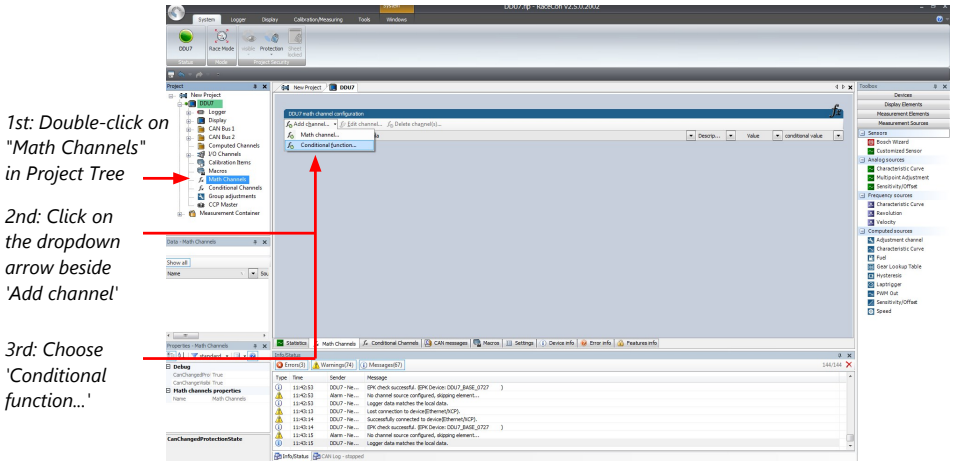
Click ‘Finish’ when done. The math channel is displayed in the math channel window.

7.2 Conditional Functions

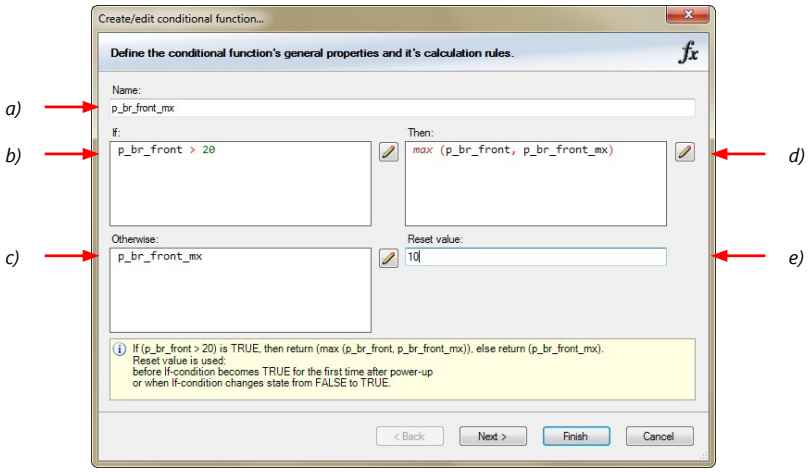
- Arithmetic and logical operations on one or more measurement channel(s)
- If-Else structure with reset
- Numerical result
- Result can be used as input source for various display elements (numeric elements, alarms, bargraphs) and further calculations in the whole RaceCon project.
  All math and conditional channels can be used globally in the whole RaceCon project.

Creating a new Conditional Function

1. Follow the steps shown in the screenshot. The “create/edit math channel” window appears.



2. Define the math channel using the following configuration possibilities:



a) Enter the name of the conditional function.
b) Enter the If-condition. Click pencil symbol to open an editor to enter expressions.
c) Enter the Then-condition. Click pencil symbol to open an editor to enter expressions.
d) Enter the Otherwise-condition. Click pencil symbol to open an editor to enter expressions.
e) Enter the reset value (must be a number).

Click ‘Finish’ when done.

The conditional function works the following way:

The program always calculates the condition entered in the IF window and checks if the condition is TRUE or FALSE.

If the condition entered in the IF window is TRUE, the program calculates the condition entered in the THEN window. The returned value is the content of the new variable (entered in “Name”).

If the condition entered in the IF window is FALSE, the program calculates the condition entered in the OTHERWISE window. The returned value is the content of the new variable (entered in “Name”).

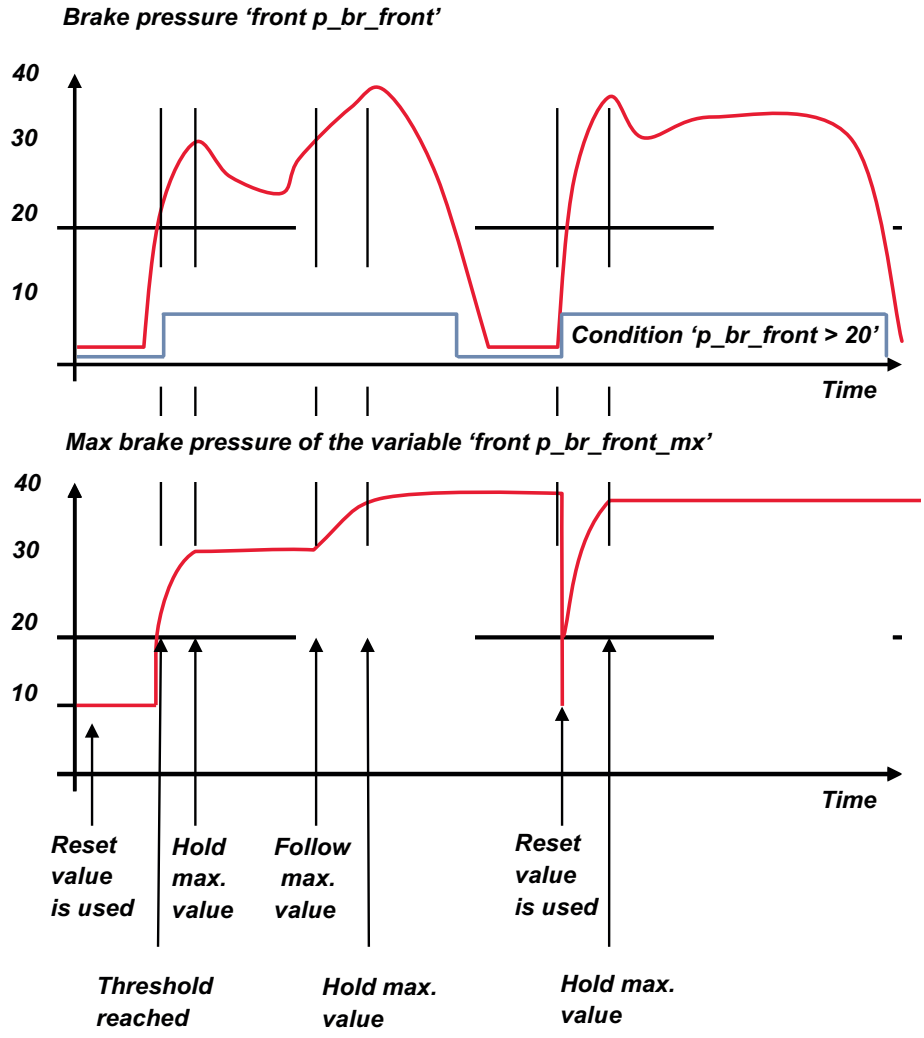
The reset value is always set for the new variable (entered in “Name”):

- before the If-condition becomes TRUE for the first time after power-up
- when the If-condition changes state from FALSE to TRUE.

An example of a condition to set up the maximum front brake pressure is given on the next page.

The conditional function is displayed in the DDU 11 math channel window.

Example: Setting up a condition for maximum front brake pressure



- At power-up, the reset value (10) is used for ‘p_br_front_mx’.

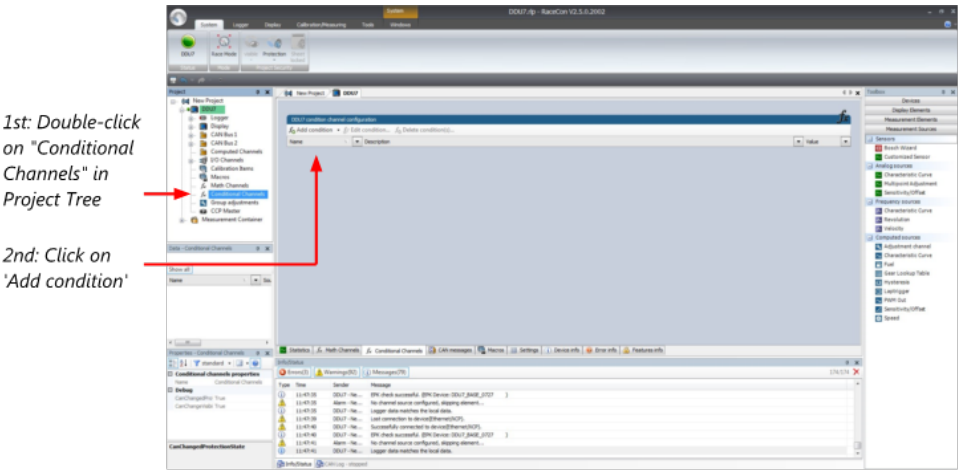
- ‘p_br_front’ rises to 30. As ‘p_br_front’ is > 20 (condition is TRUE), the condition ‘max (p_br_front, p_br_front_mx)’ in the THEN window is triggered. The condition sets the bigger value as new value for ‘p_br_front_mx’. As ‘p_br_front’ (30) is bigger than ‘p_br_front_mx’ (10), the new value for ‘p_br_front_mx’ is set to 30.
- Although ‘p_br_front’ falls to 25, the value of ‘p_br_front_mx’ stays 30. This is caused by the THEN-condition, because p_br_front_mx’ (30) is still bigger than p_br_front’ (25).
- ‘p_br_front’ rises to 40. As ‘p_br_front’ (40) is bigger than ‘p_br_front_mx’ (30), the new value for ‘p_br_front_mx’ is set to 40.
- As ‘p_br_front’ falls below 20, the IF-condition turns to FALSE. Now the OTHERWISE-condition is triggered. Because the condition ‘p_br_front_mx’ sets the value of ‘p_br_front_mx’ and the value is already set to 40, nothing changes.
- When ‘p_br_front’ rises to 40, the IF-condition changes to TRUE again and triggers the THEN-condition. Now the reset value (10) is used for ‘p_br_front_mx’ in the THEN-condition.
- The new value of ‘p_br_front_mx’ is 40 because 40 is bigger than 10.

7.3 Conditional Channels

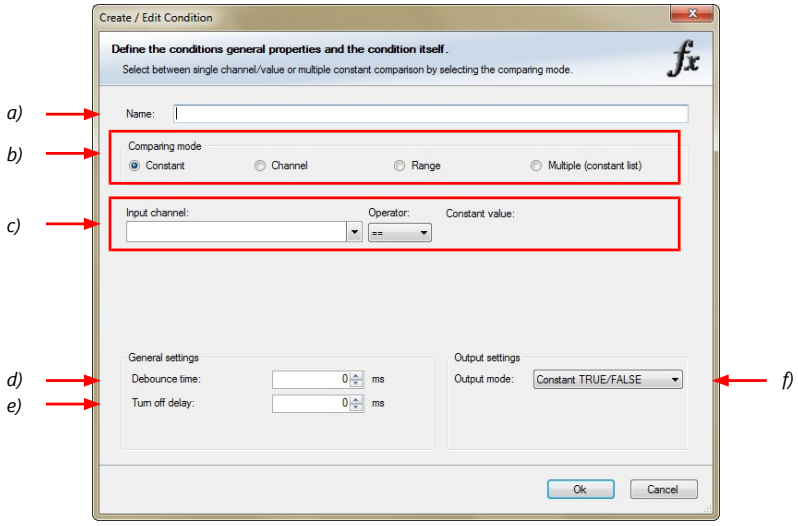
- Logical operations on measurement channel(s)
- If-Else structure with reset
- Logical result
- Result can be used as input source for alarm display elements and further calculations in the whole RaceCon project.

Creating a new Conditional Channel

1. Follow the steps shown in the screenshot. The “Create/edit condition” window appears.



2. Define the condition channel, using the following configuration possibilities:



a) Enter the name of the conditional channel.

b) Select the comparing mode:

- Constant: Compare a measurement channel with a constant value.
- Channel: Compare a measurement channel with a measurement channel.
- Range: Compare a measurement channel with a defined value range.
- Multiple: Compare a measurement channel with up to 5 constant values.

c) Depending on the chosen comparing mode, you can enter the following values:

- Constant: Choose the measurement channel or condition, the operator and enter the value of the channel.
- Channel: Choose the measurement channel or condition, the operator and the measurement channel or condition to be compared.
- Range: Choose the measurement channel or condition, the operator and define the minium and maximum value.
- Multiple: Choose the measurement channel or condition, the operator and enter the value of up to 5 constants.

d) Enter the minimal time to detect the signal of the measurement channel, to avoid high-frequent switchovers.

e) Enter the time by which the signal of the measuring channel is delayed after its end.

f) Choose the output setting of the result.

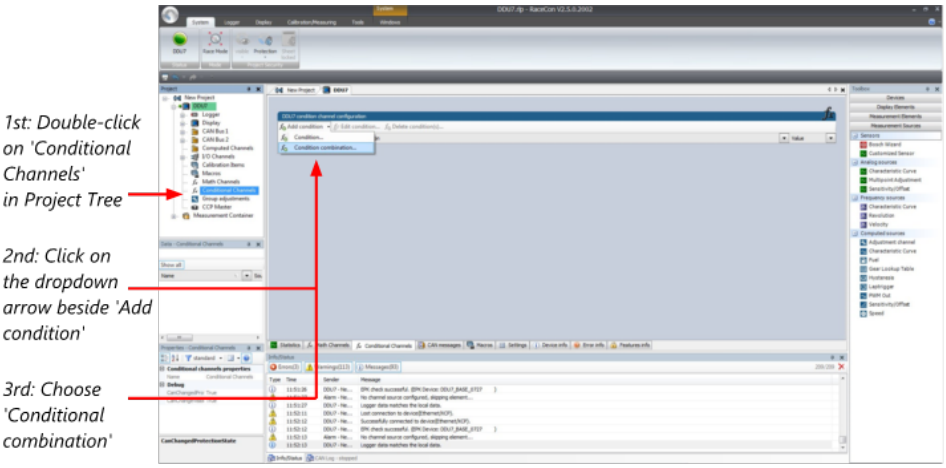
- Constant TRUE/FALSE: Result is as a constant with the value TRUE or FALSE.
- Blinking: Result is a blinking, if the condition is fulfilled.
- Pulse: Result is a short one-time pulse, if the condition is fulfilled.
- Toggling output: Result is a pulse that lasts until the next condition is fulfilled.
- Click ‘Ok’ when done. The conditional channel is displayed in the DDU 11 condition channel window.

7.4 Condition Combination

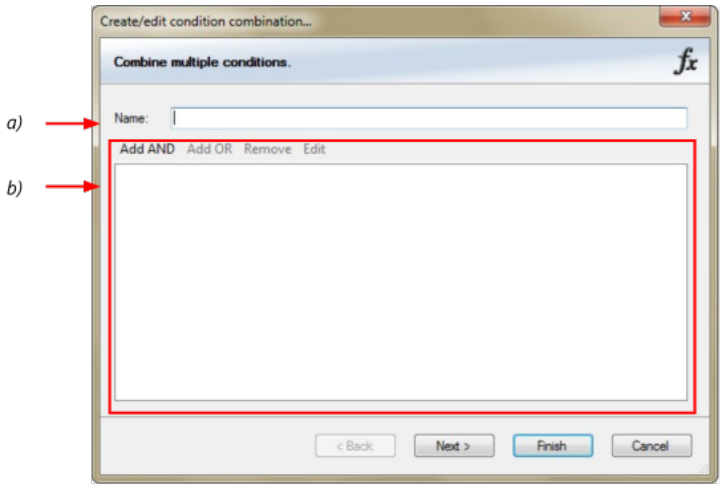
- Combination of several (up to 16) conditional channels for more complex calculations
- Logical results
- All conditions can be used globally in the whole DDU 11 project.

Creating a new Condition Combination

Follow the steps shown in the screenshot.



The "Create/edit condition combination" window appears. Define the condition combination, using the following configuration possibilities:

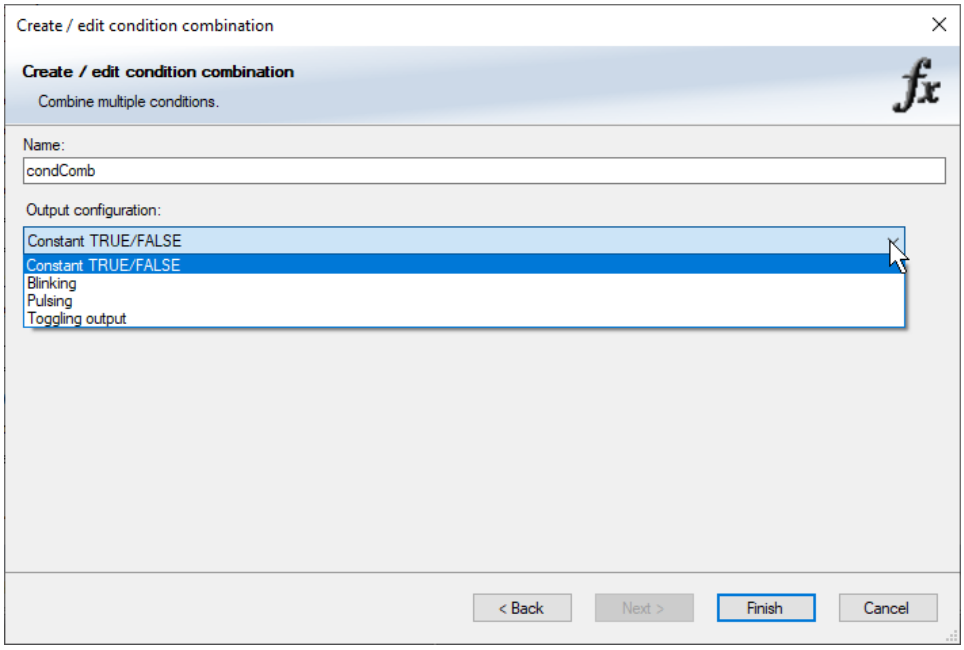


a) Enter the name of the condition combination.

b) Create the condition combination in the window.

- Choose a channel (condition, conditional function, math, measurement channel with binary values) to be compared.
- Combine multiple conditions, by adding 'AND' or 'OR' relations.
- To negate a condition, click with the right mouse-button on the condition and select 'Negation (!)'.
- Combine several (up to 16) conditions.

Click 'Next' to go to the next page. Choose the output setting of the result:



- Constant TRUE/FALSE: Result is as a constant with the value TRUE or FALSE.
- Blinking: Result is a blinking, if the condition is fulfilled.
- Pulsing: Result is a short one-time pulse, if the condition is fulfilled.
- Toggling output: Result is a pulse that lasts until the next condition is fulfilled.

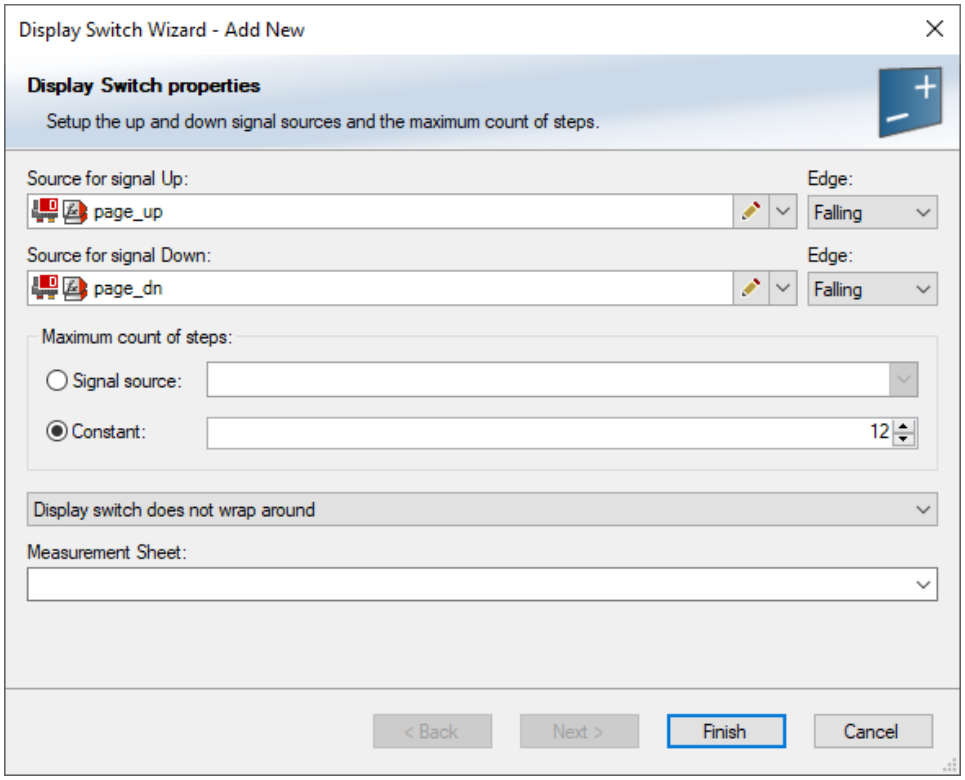
Click ‘Finish’ when done. The conditional combination is displayed in the DDU 11 condition channel window.

7.5 Display Switch Module

You can use the Display Switch Module to switch display pages and brightness. The output is a display page or brightness output that can be used in display configurations. The value sustains over a power cycle.

The conditions for incrementing/decrementing the value can be set freely. The maximum value can be set as constant or read from a measurement.

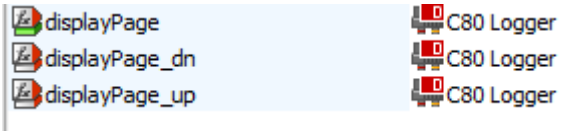
The page can be configured to wrap around. In this case, no page down condition is needed.



The resulting outputs are the display switch value and the input conditions.

Measurement label	Function
name	page or brightness value
name_**dn**	input condition for decrement
name_**up**	input condition for increment

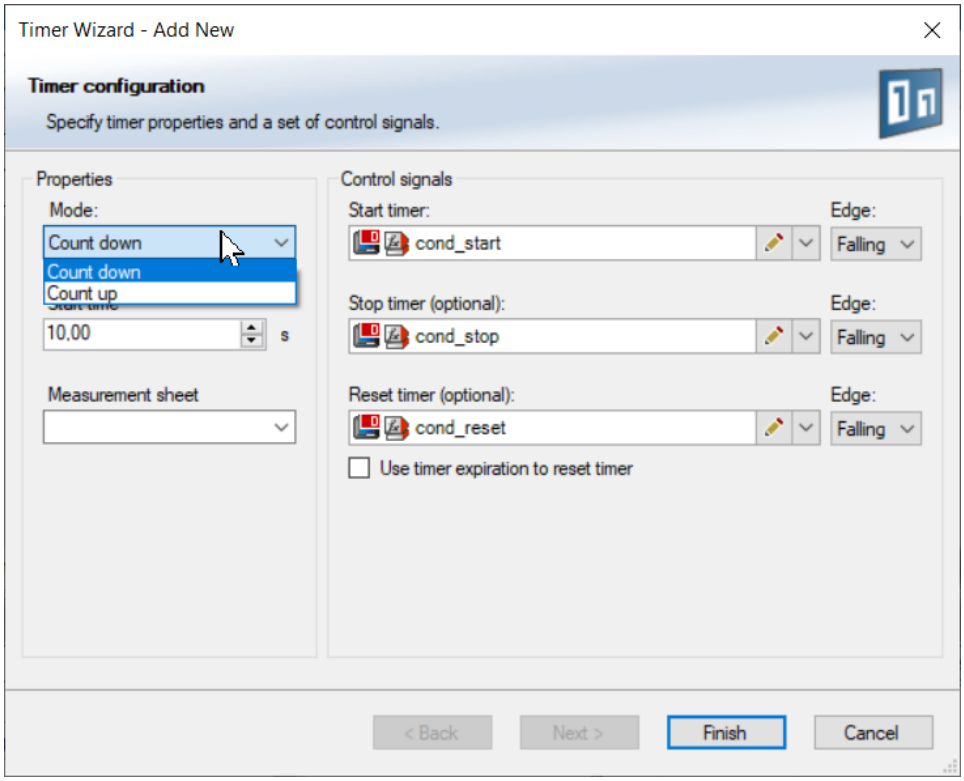
Example:



7.6 Timer Module

The Timer Module is designed to implement timing triggers, i.e. for rallye stage timing or minimum pit time calculations. Any event in the system can be used for starting, stopping and resetting the timer.

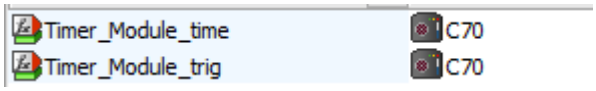
Up counting mode and down counting mode are available, triggers are fired at set time (up counting) or at zero (down counting). The running timer will keep its state over a power cycle.



The output channels for this module depend on the name used for the module and are called ..._time and ..._trig.

Measurement label	Function
name_**time**	actual timer value
name_**trig**	trigger set by timer alarm

In this example, the module is named “Timer_Module”. Resulting channels are:

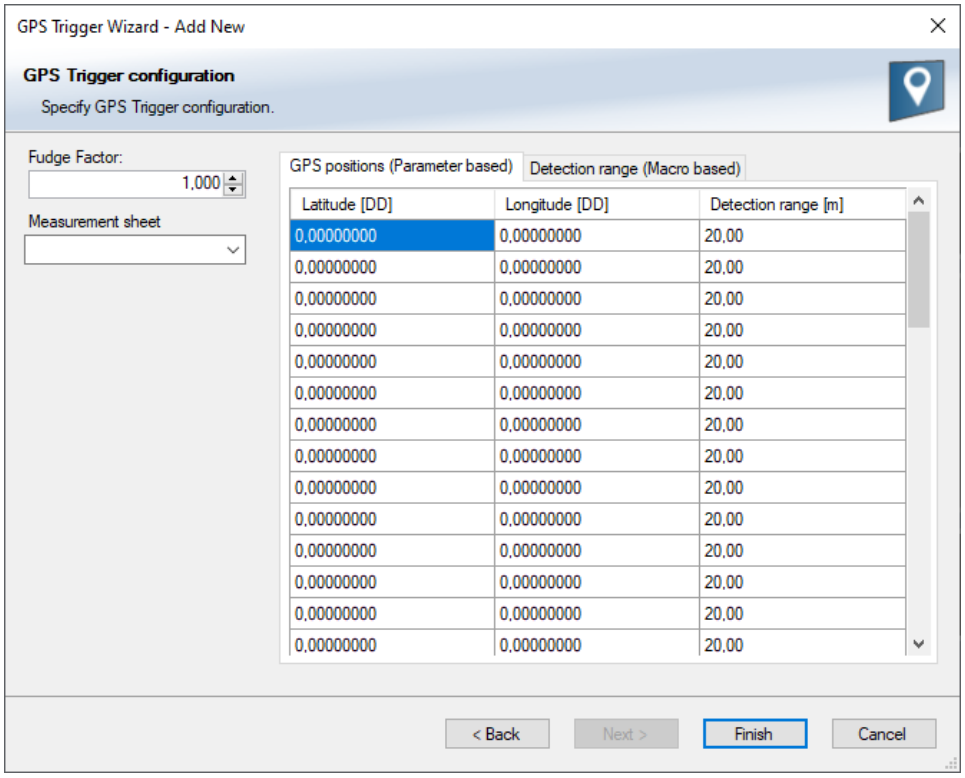


7.7 GPS Trigger Module

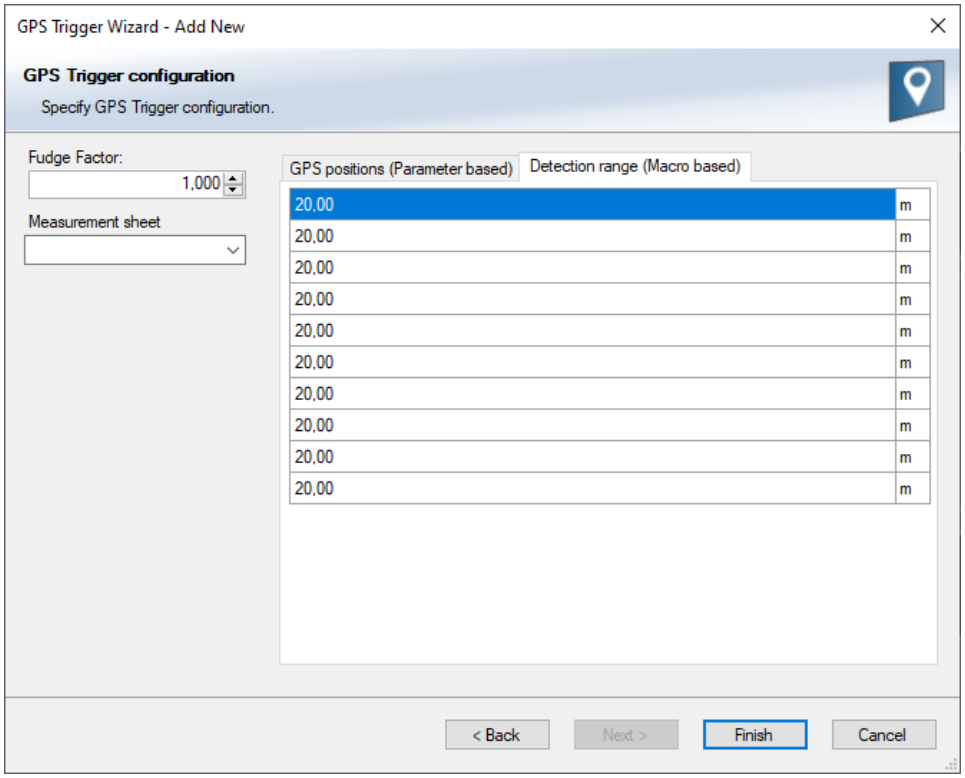
The GPS Trigger Module triggers depending on GPS-position, like the GPS-laptrigger.

There are 50 GPS trigger points for parameter application of latitude/longitude coordinates, as well as 10 macro-based coordinates.

If the car passes one of the trigger points, an output signal is set to 1 shortly. Each trigger requires a defined latitude, longitude, and detection range.



The parameter-based trigger points need to be set manually in RaceCon, the macro-based trigger points will store latitude and longitude values when the configurable trigger condition comes true (i.e., steering wheel button). This trigger condition and the detection range need to be configured in RaceCon.

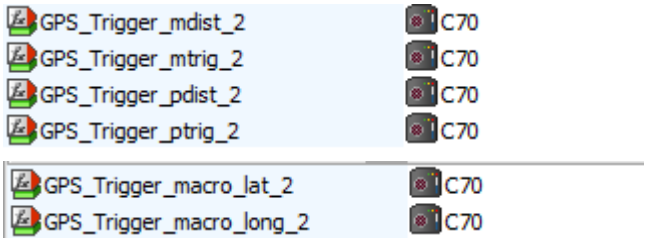


The GPS trigger points can also be used for segment triggering. If used as segment triggers and i.e., 3 trigger points are selected, the laptrigger module will use the first 3 trigger points on the list.

The channel names depend on the name used for the module, in this example GPS_Trigger. Each trigger has a distance and a trigger channel with the abbreviation m for macro or p for parameter based. The trigger channel will be set to 1, when the lowest distance to the trigger point is detected. For the macro-based trigger, the stored latitude and longitude values can be seen with the channels.

Measurement label	Function
name_**lat**	interpolated gps latitude
name_**long**	interpolated gps longitude
name_**ptrig_1..50**	trigger output of parameter based trigger (n)
name_**pdist_1..50**	distance to trigger of parameter based trigger (n)
name_**mtrig_1..10**	trigger output of macro based trigger (n)
name_**mdist_1..10**	distance to trigger of parameter based trigger (n)
name_**macro_lat_1..10**	stored latitude for macro based trigger (n)
name_**macro_long_1..10**	stored longitude for macro based trigger (n)

Example:



7.8 CPU Load Limits

As all microprocessors, the two processors of the DDU 11 have limited capacities. The current load of the processors can be monitored using the channel “cpu_load_001” or “cpu_load_002”. When configuring your device, please make sure the used CPU load is in a save range below 100 %.

Bosch recommends a maximum CPU load of 85 % (averaged). Exceeding this limit might result in the DDU 11 not being able to fulfill its required measuring/logging/display tasks or even in crashing and rebooting.

Main factors influencing the CPU load are:

- Number and complexity of math channels
- Number and complexity of conditions
- CAN traffic on both CAN lines
- Display configuration, especially displaying pictures
- Logger configuration (total logging rate [kB/s], conditional measurement rates)

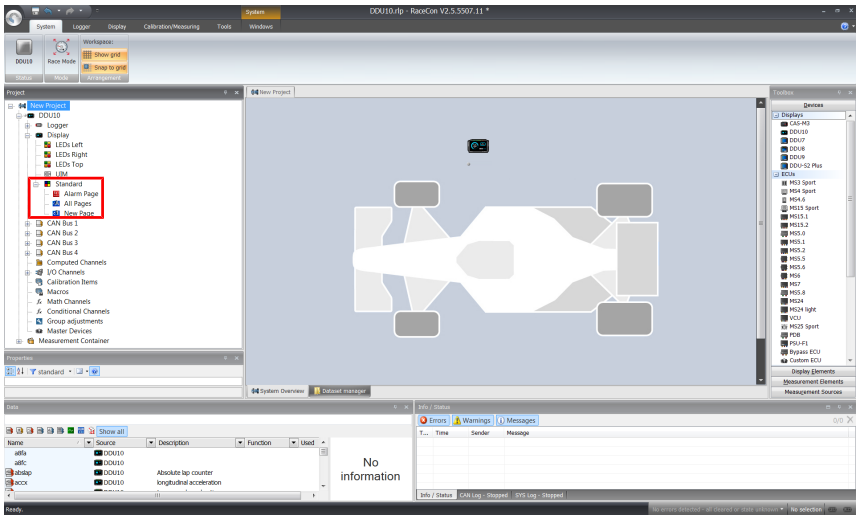
To help respecting the limit of 85 % CPU load, the DDU 11 creates an error memory entry. To trigger this error entry, the CPU load must exceed the limit for 5 minutes without interruption.

When being confronted with this error memory entry (see ‘Error info’ in RaceCon) or when being confronted with DDU 11 resets due to complex configuration setups, please consider reducing the demands on the DDU 11 adapting the influencing factors mentioned above.

8 Display Configuration

8.1 Display page setup

8.1.1 Organizing display pages



- All Pages: Display elements placed on this page are displayed on all pages.
- Single Page: Display elements placed on this page are displayed only on this page.
- Under a Page set you can create different display configurations, like style or language based. You can choose which page set should be used on the device (via UIM), or via RaceCon, by right-clicking on the Page set in the project tree overview.

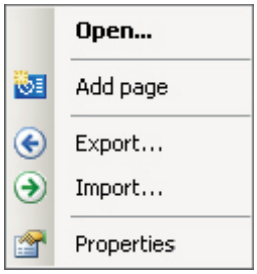
The priority of display elements placed on 'All Pages' is higher than the priority of display elements placed on single pages.

Alarm Elements have the highest priority and are displayed in front of all other elements. You can find alarm pages above "All pages" in the project tree.

Example: A bar graph placed on 'All Pages' is displayed on all display pages and is always in front of other display elements on single pages.

8.1.2 Adding a new display page

Click with the right mouse button on 'Display' in the project tree and click on 'Add page' in the menu or on the 'Add' button at the top.

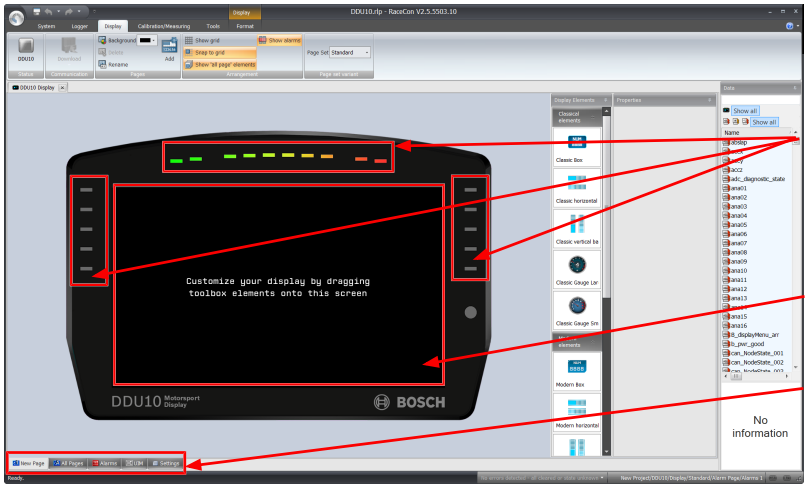


A new empty page opens.

8.1.3 Selecting display pages

Click on the 'Display' tab and then on one of the page tabs at the bottom (example: 'New Page').

In the Main Area, a representation of the selected page opens and can be configured.



8.1.4 Non-symmetrical arrangement of pixels

The display has a non-symmetrical arrangement of pixels in the display panel. Therefore, elements with e.g. 40*40 pixels do not appear square, but as a rectangle, stretched a bit along the horizontal.

To keep WYSIWIG working, we have chosen a technical trick: RaceCon offers a resolution of 867 x 480 pixels for the display, instead of the 800 x 480 pixels in the controller, to compensate for the asymmetry. So, the pages are designed for 867 pixels width in RaceCon but scaled down to 800 pixels width in the control unit, so that the dimensions in RaceCon fit to the dimensions in the control unit.

8.2 Element Configuration

You can insert different display elements and configure these in the 'Properties' toolbox. For an overview of the display elements, see chapter 'Types of display elements [▶ 48]'.

1. Drag a display element from the toolbox and drop it on the display page.

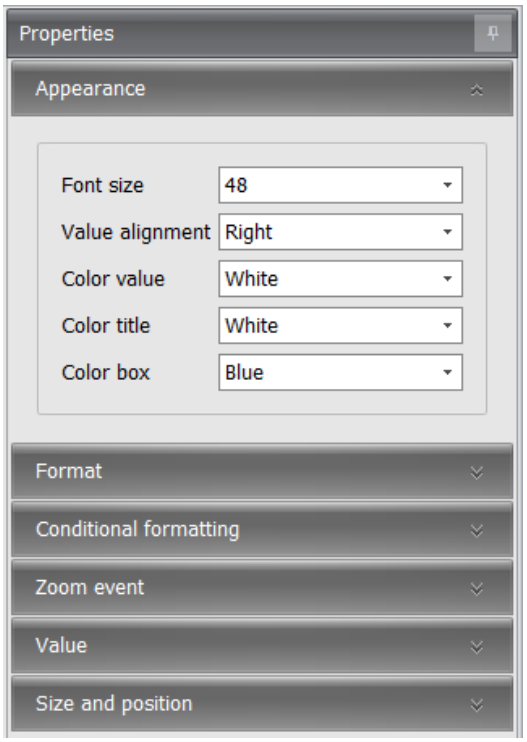
   The 'Properties' toolbox opens and you can configure the active element.

2. Use the search bar in the 'Data' window, to search for a measurement channel.
3. Drag the desired measurement channel from the 'Data' window and drop it on the display element.
4. Alternatively, use 'Value channel' in the 'Properties' toolbox to define a measurement.
5. To swap the active channel element, click on the needed element.
6. To perform a multi selection, click and hold the mouse button and swipe over the wanted elements. Alternatively, you can add or remove elements from the selection with CTRL + mouse click.

   (Only overlapping properties will be displayed.)

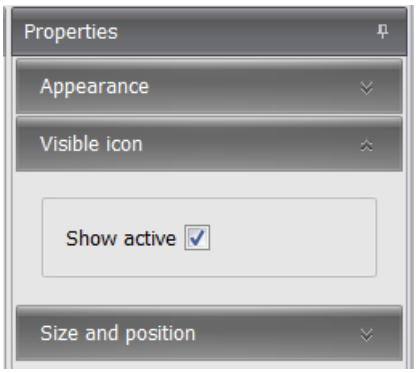
You can find the following configuration options in the 'Properties' toolbox. The following screenshots are created by way of example for available options. Which options will be available, depends on which element is active:

Appearance



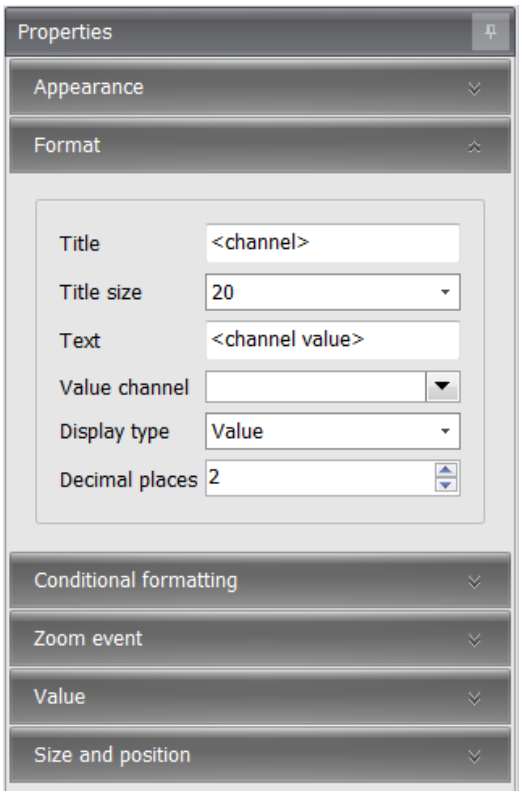
- Font size: Defines the size of the font.
- Alignment: Defines the alignment of the element content.
- Value alignment: Defines the alignment of the value.
- Orientation: Defines the orientation of the element content.
- Color value: Defines the color of the value.
- Color title: Defines the color of the title.
- Color box: Defines the color of the box.
- Color bar: Defines the color of the bar.
- Color gauge: Defines the color of the gauge.
- Color ticks: Defines the color of the ticks.
- Color border: Defines the color of the border.
- Background: Defines the color of the background.
- Image file: Here you can select the image you want to display.
- Scale image: Enables the fitting of the image according to the box.
- Keep aspect ratio: Keeps the original ratio of the image.
- State channel: Defines the channel to which an alarm icon should react.
- Active icon: Defines an icon, which appears when the channel is active.
- Use inactive icon: Enables the usage of an inactive icon.
- Inactive icon: Defines an icon, which appears when the channel is inactive.

Visible icon



– Show active: Enables testing of the active icon (only for alarm icon elements).

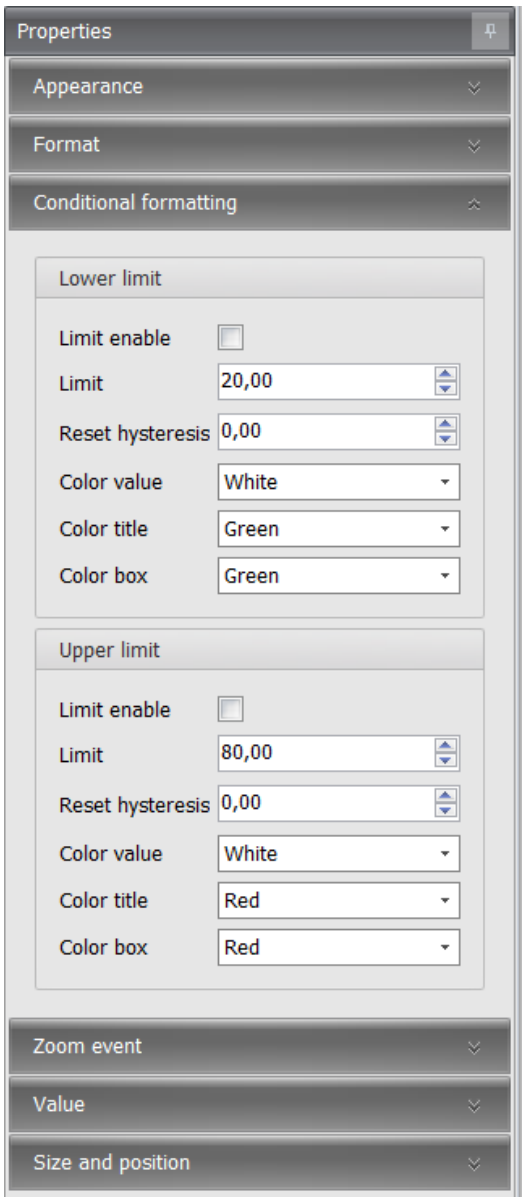
Format



– Title: Defines a specific title.
– Title size: Defines the font size of the title.
– Text: Defines a specific text and/or a value, which will always be displayed in the box.
– Value channel: Defines the value channel, which will be monitored.
– PopUp condition: Defines the condition when the element will be visible on the display.
– Display type: Defines the type of information, which will be displayed. You can differentiate between 'Value', 'Gear' and a type of time display.
– Decimal places: Defines the decimal places, which will be displayed.
– Fill style: Defines the style of the bar filling. You can choose between blocks and solid.
– Tick visibility: Defines the visibility of the ticks.
– Minimum value: Defines the minimum value at which the counting starts.

- Maximum value: Defines the maximum value at which the counting ends.
- Start angle: Defines the angle at which the ticks start.
- Angle range +/-: Defines the angle range of the ticks.
- Tick mark list: Here you can edit the tick mark position.
- Sub divisions: Defines the number of sub divisions between the main ticks.
- Conversion table: Here you can edit the conversion table.
- Fallback: Defines the displayed information when the value channel is invalid for the enumeration box.

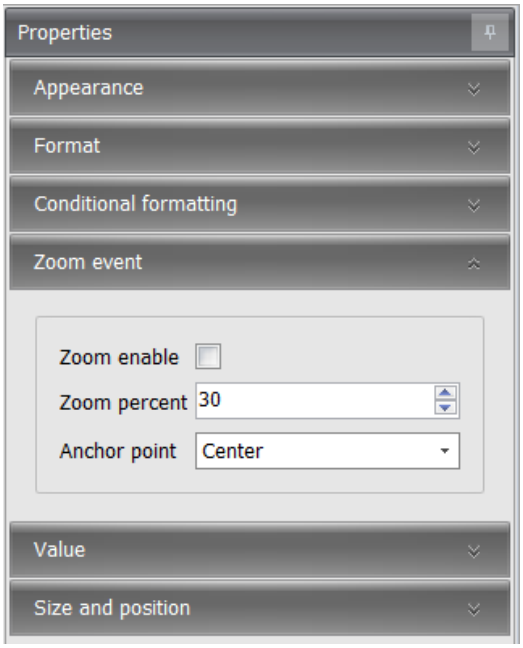
Conditional formatting (lower and upper limit respectively)



- Limit enable: Enables a lower limit and an upper limit respectively.
- Limit: Defines the threshold limit.
- Reset hysteresis: Defines the value as off when the hysteresis will be reset. You can find more information about the hysteresis function in general in the chapter Hysteresis [▷ 109].
- Color value: Defines the color when the value is above/below the set threshold limit.

- Color title: Defines the new color of the title when the value is above/below the set threshold limit.
- Color box: Defines the new color of the element when the value is above/below the set threshold limit.
- Color gauge: Defines the new color of the gauge when the value is above/below the set threshold limit.
- Color ticks: Defines the color of the ticks when the value is above/below the set threshold limit.
- Color bar: Defines the color of the bar when the value is above/below the set threshold limit.
- Color border: Defines the new color of the border when the value is above/below the set threshold limit.
- Background: Defines the color of the background when the value is above/below the set threshold limit.

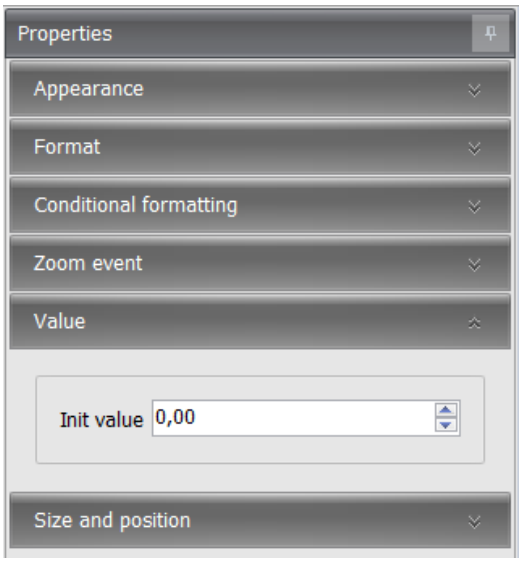
Zoom event



- Zoom enable: Enables the zoom event.

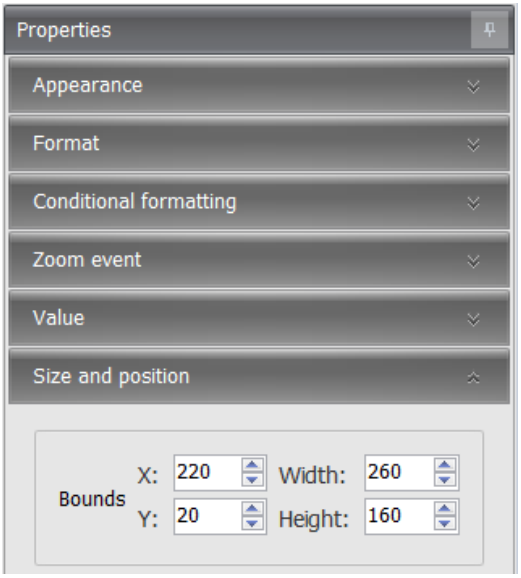
  If checked, the box will increase its size for a short time, after the value changed.
- Zoom percent: Defines the percentage by which the box will zoom out.
- Anchor point: Defines the anchor point, which will be fixed during the zoom event.

Value



- Init value: Defines the value, which will be shown in RaceCon during configuration or on the device if the channel has no link. Allows testing the conditional formatting by entering a representative value.

Size and position



- X-Pos: Defines the x-position of the top left corner of the box within the page.
- Y-Pos: Defines the y-position of the top left corner of the box within the page.
- Width: Defines the width of the element.
- Height: Defines the height of the element.

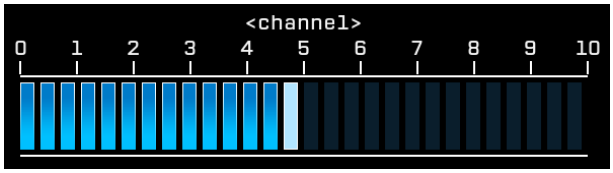
8.3 Types of display elements

RaceCon has two types of element styles, the ‘Classical Elements’ and the ‘Modern Elements’. Both styles contain the same type of elements; the difference is solely in appearance. Additionally, there is the basic element collection, which consists of elements without any style. Some of these are also transparent and can be used in combination with other elements by layering them.

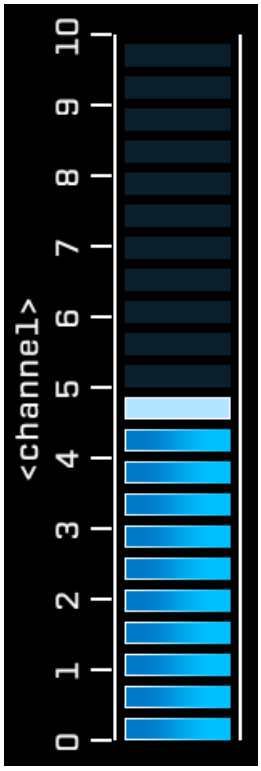
You can find the following display elements in the ‘Display Elements’ catalog:

Classic Box

Classic horizontal bar



Classic vertical bar

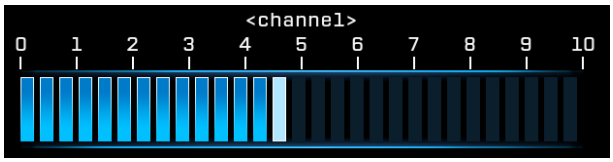


Classic Gauge Large

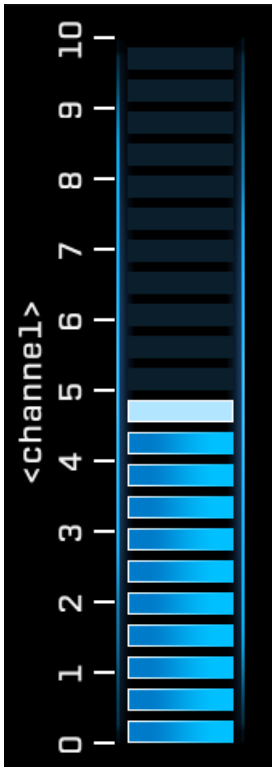
Classic Gauge Small

Modern Box

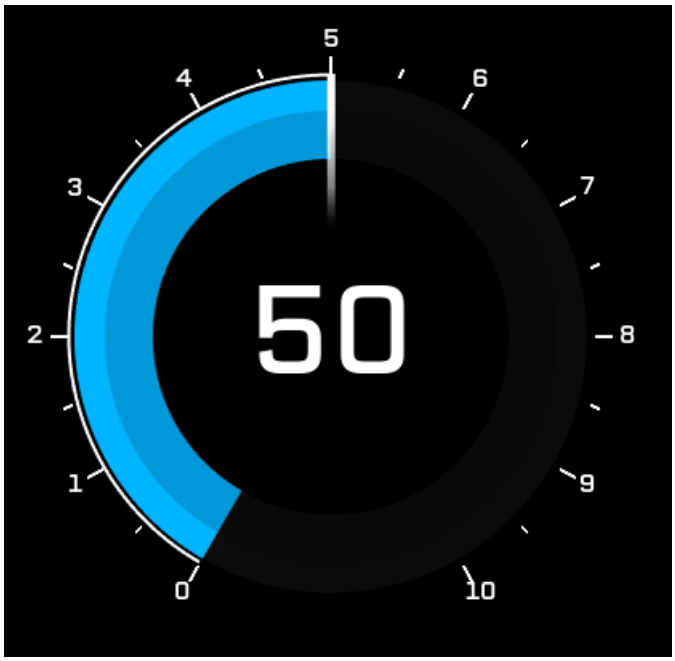
Modern horizontal Bar



Modern vertical Bar



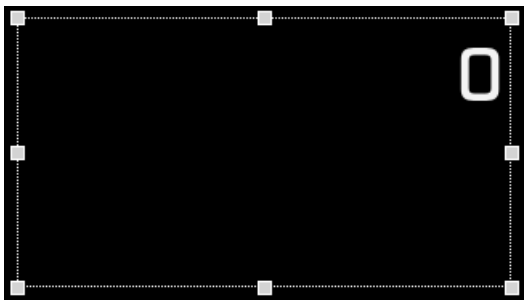
Modern Gauge Large



Modern Gauge Small

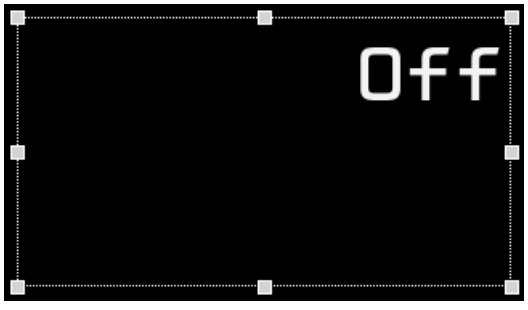
Basic Box

Value / Text



Can be used as a simple value or text element, which might be placed on top of another element, like the classic gauge large.

Enumeration

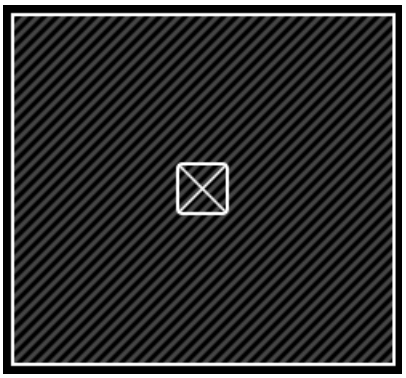


Can be used to translate a value into a text. For example, for a wiper setting 0 - Off, 1 - Slow, 2 - fast

Image

Can be used to insert images, for example as background images.

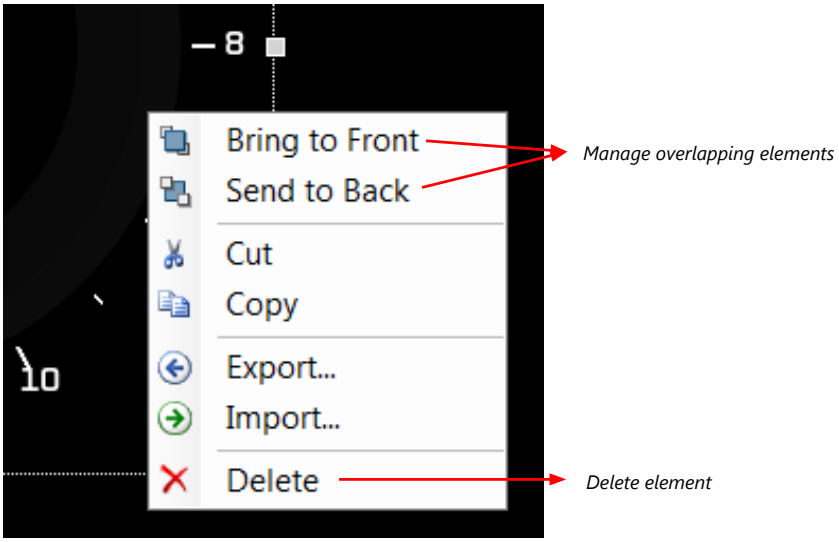
Alarm Icon



Icon or Icons, which can be linked to a conditional channel.

8.4 Context menu

The context menu appears when you click with the right mouse button on a display element.



8.5 LEDs

The LEDs are fully configurable to show for example the optimal shifting point. They can also be configured to flash in case of a customized condition becoming 'true'.

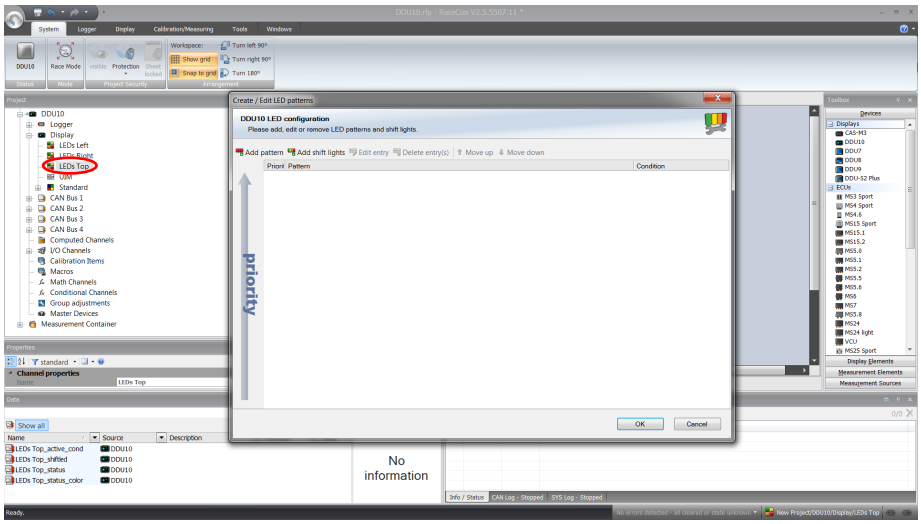
8.5.1 Configuring shift LEDs

To use shift LEDs, you need a channel with the unit RPM. It is also possible to use gear depending shift light.

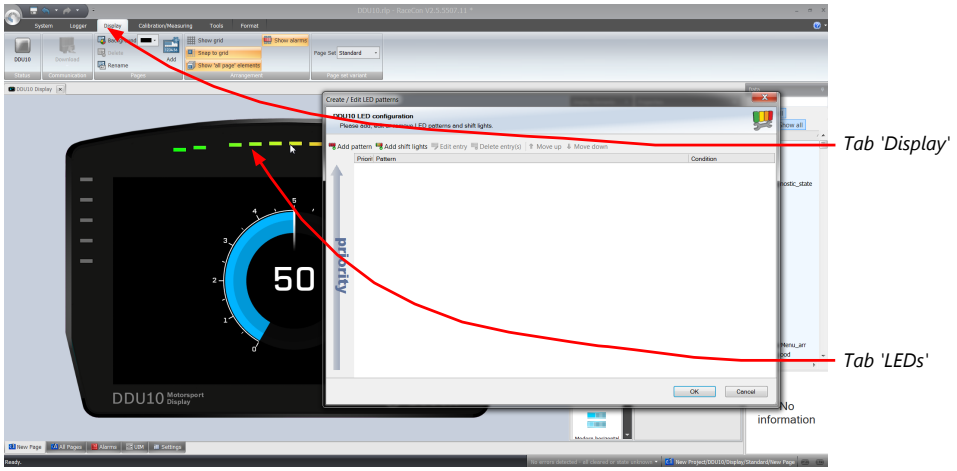
You can configure the shift LEDs only in the 'LEDs Top'.

1. Double-click on 'LEDs Top' in the project tree.

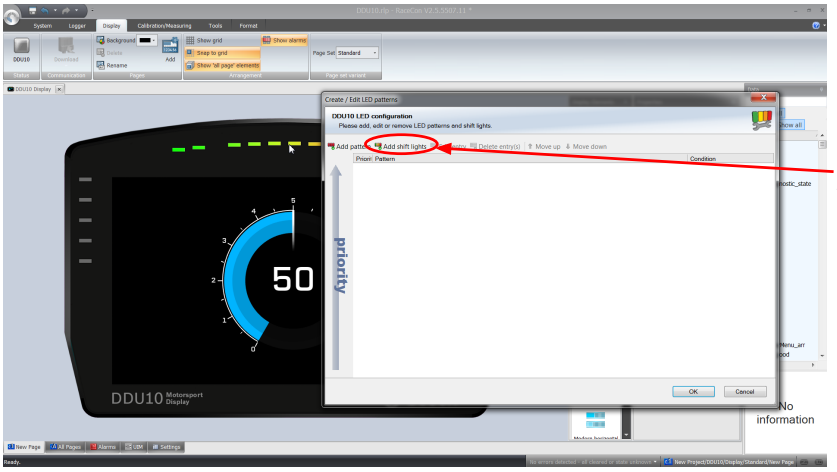
   You will find them under 'Display'.



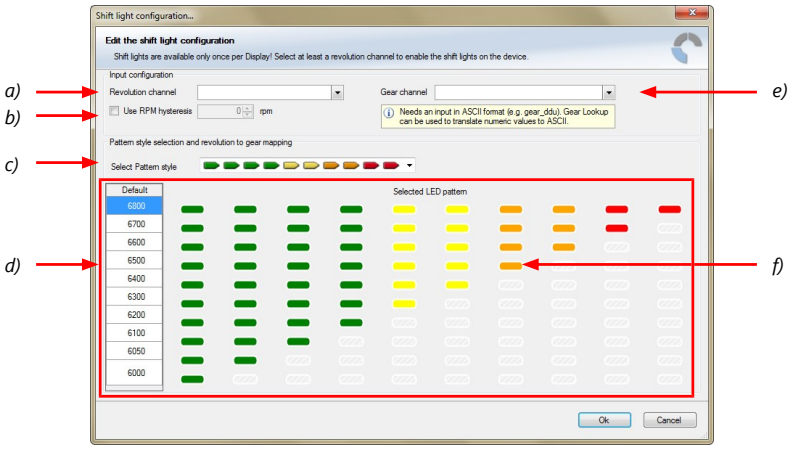
2. Alternatively, click on the 'Display' tab and then click on the colored LEDs at the top of the display image.



3. Click on the button 'Add shift lights'.



4. The ‘shift light configuration’ window appears. Set up the shifting lights using the following configuration possibilities:



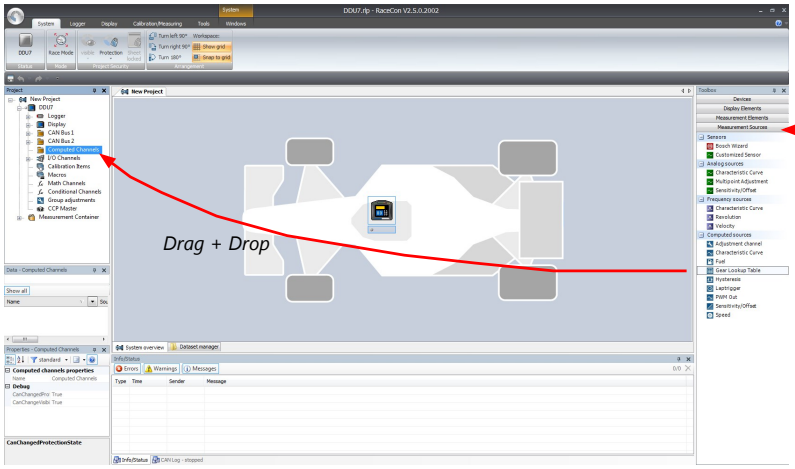
a) Choose the measurement channel for 'Revolution'. Revolution must have 1/min quantization.
b) Enter the limit value when the RPM hysteresis function is active. The RPM hysteresis function avoids the high-frequent switchover of the measurement channel value.
c) Choose a predefined Pattern style.
d) Define the rpm thresholds to show the LED pattern.
e) Choose the measurement channel for 'Gear'. Gear must have an ASCII quantization (1st gear='1' = 49, 2nd gear='2' = 50, ...). (ASCII quantization is standard for the 'gear' channel of Bosch ECUs.
If you get the gear information of a different control unit as the Bosch ECU (e.g. a gearbox control unit), use the Gear Lookup Table to translate numeric values to ASCII format.
For more information see chapter "Converting a gear channel to ASCII representation".)
f) Click with the right mouse button on the LEDs to reconfigure the LED patterns.

5. Click ‘OK’ when done. The configuration is displayed in the DDU 11 LED Configuration window.

8.5.2 Converting a gear channel to ASCII representation

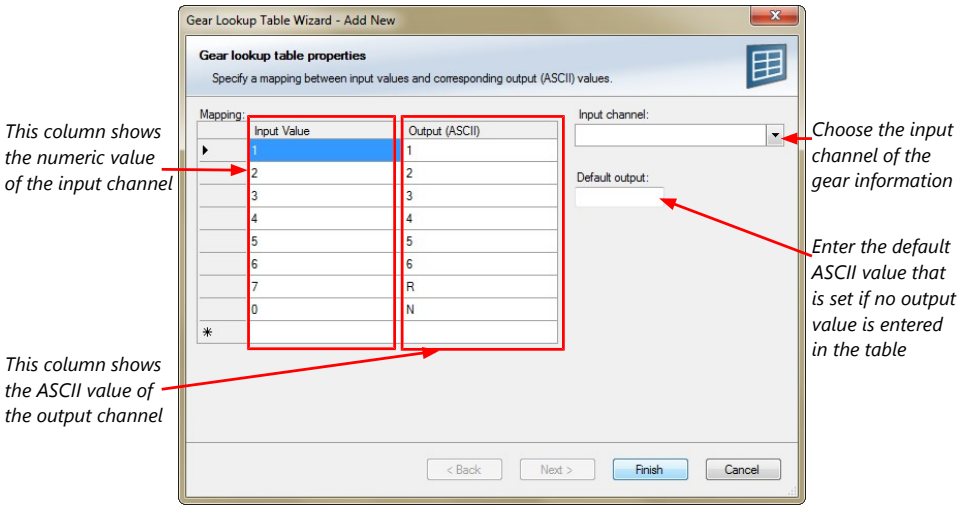
If you get the gear information of a different control unit as the Bosch ECU (e.g. a gearbox control unit), use the Gear Lookup Table to translate numeric values to ASCII format.

1. Click on the ‘Measurement Sources’ button in the Toolbox.
2. Drag the ‘Gear Lookup Table’ symbol and drop it on the ‘Computed Channels’ folder. The Gear Lookup Table Wizard appears.



Gear Lookup Table Wizard

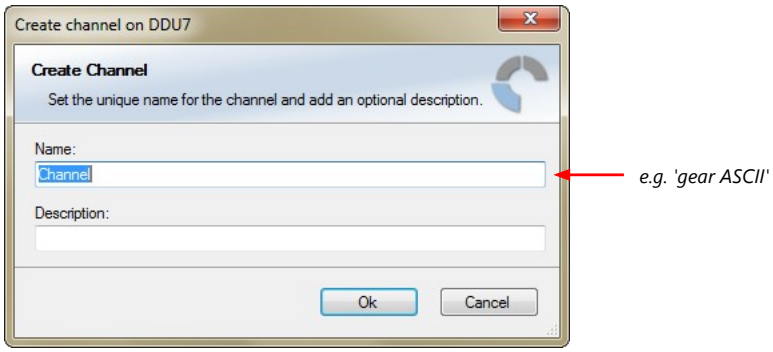
1. Set up the settings as shown in the screenshot.



2. Click ‘OK’ when done.

   The ‘Create channel on DDU 11’ window appears.

3. Enter the name and an optional description of the translated ASCII measurement channel.



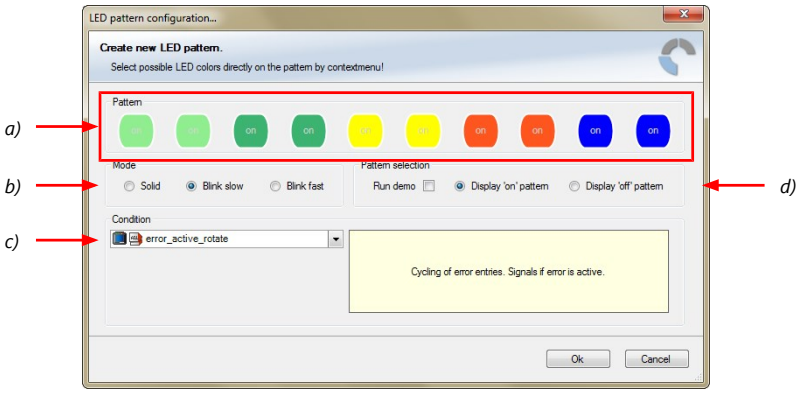
4. Click ‘Ok’ when done.

   A graphic shows the connection between the input and output channels. You can now use the measurement channel in the shift LED configuration.

8.5.3 Creating customized LED pattern

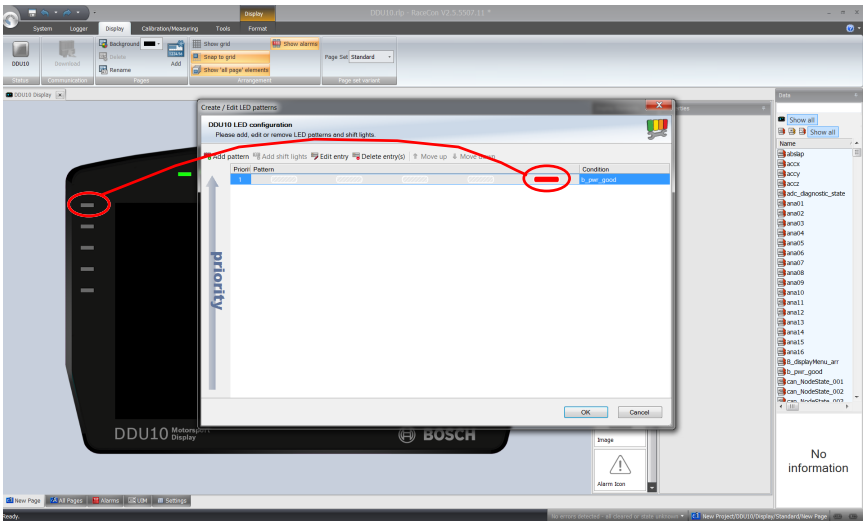
You can create your own LED pattern with an individually created condition, using the top LEDs and the ones on the sides. The LEDs illuminate or flash if the condition becomes true.

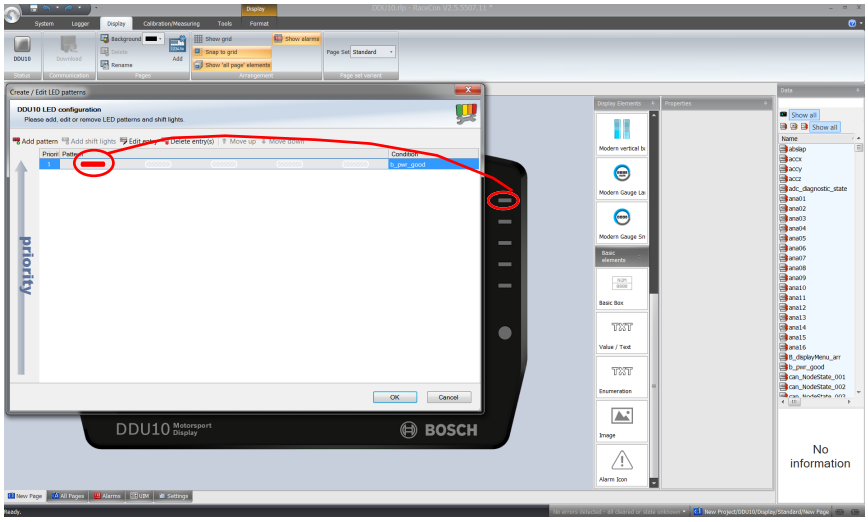
Click on the button ‘Add pattern’ in the display view. The LED pattern configuration window appears.



a) Choose the needed LEDs by clicking or via multiselect with ctrl + clicking and define the color of the LEDs by right-clicking on one of the selected LEDs
b) Select if the LEDs blink or do not blink.
c) Choose the condition when the LEDs will flash.
- Create a condition using the Condition Creator. For more information see chapter "Creating a new condition channel".
- Choose an existing condition.
d) Check the box to show a demo of the LEDs. (Important to check blinking)
To create a LED that alternately blinks in two different colors, choose 'Display "on" pattern' and define the LEDs in the first color. Then choose 'Display "off" pattern' and define the LEDs in the second color.

The direction of the pattern changes for each side. For the LEDs on the left side the pattern starts at the bottom LEDs (**right** side of pattern is for the top LEDs), and for the LEDs on the right side the pattern starts at the top LEDs (**left** side of the pattern is for the top LEDs).





Click ‘OK’ when done.

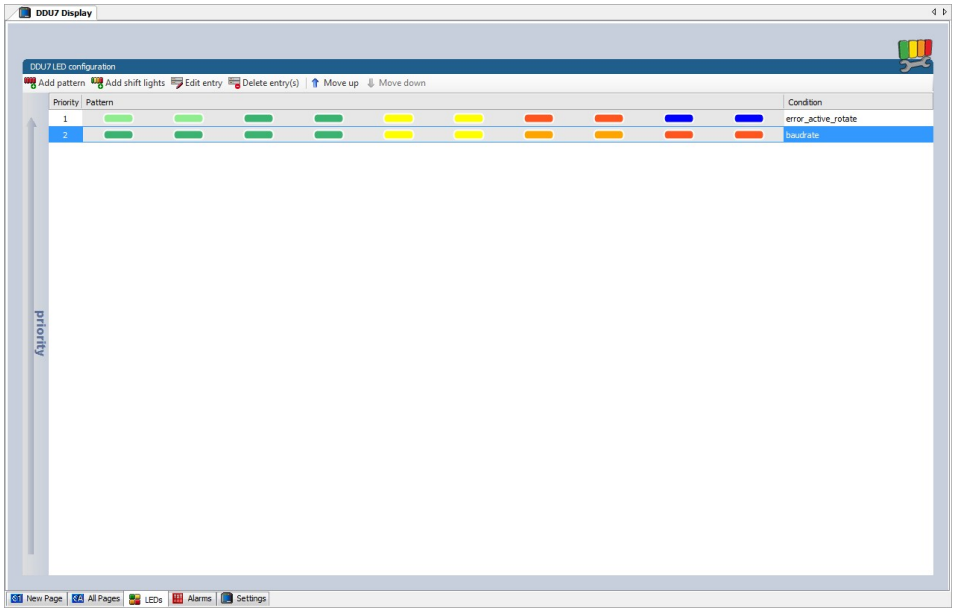
The configuration is displayed in the DDU 11 LED Configuration window.

8.5.4 Assigning display pattern priority

You can assign the priority of the created display pattern and shift lights.

The pattern with a higher priority will always cover patterns with a lower priority, when it becomes active. If a transparent (grey) LED is used in a pattern, the LEDs of lower patterns will be visible. Please ensure that for example shift lights do not cover important warnings.

Click the ‘Move up’ or ‘Move down’ button to change the priority. The pattern with the lowest number will have the highest priority.



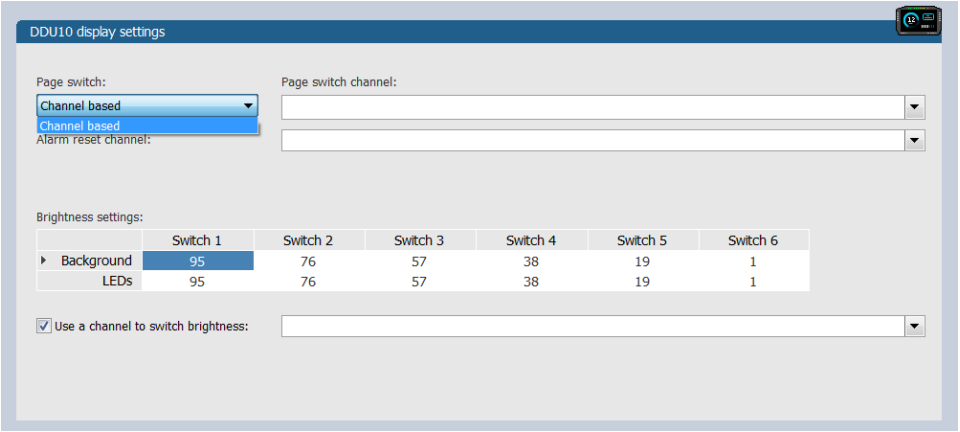
8.6 Page select / Display brightness / LED brightness

Any “event” can be used to change the display and LED brightness or the display page. Those events can be any input channel or an internal calculated channel. In the following chapters, you will find some examples on how to set up such a configuration.

8.6.1 General information

To use a channel as a page switch, select “Channel based” as page switch and select the channel you configure for a switch in the ‘display settings’ dialog (as described in the following chapters).

To use a channel as a brightness switch, check the box “Use a channel to switch brightness” and select the channel you configure for a switch.



Page switch

The pages can be switched from page one to twelve. If not all twelve pages are defined, the device switches up to the highest defined page number, and ignores higher numbers, which are not defined.

If the channel value does not only consist of integers, the pages will be switched as follows:

Page 1 is shown with the value < 1.5

Page 2 is shown with 1.5 <= the value < 2.5

Page 3 is shown with 2.5 <= the value < 3.5

...

Page 12 is shown with 11.5 <= the value

Brightness switch

The brightness can be switched with 6 positions. In the display settings dialog, you will find a chart for the 6 switch positions over the display brightness and the LED brightness (the values are in percent).

If the channel value does not only consist of integers, the pages will be switched as follows:

Switch 1 is shown with the value < 1.5

Switch 2 is shown with 1.5 <= the value < 2.5

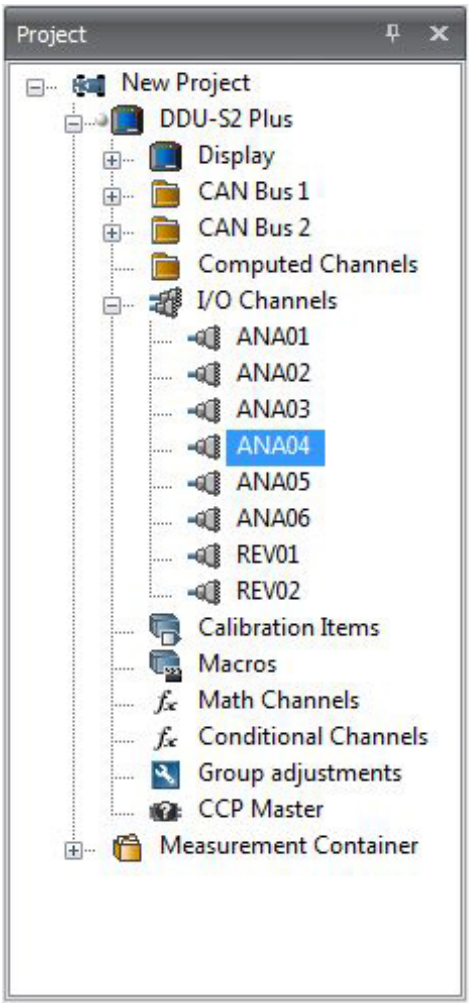
Switch 3 is shown with 2.5 <= the value < 3.5

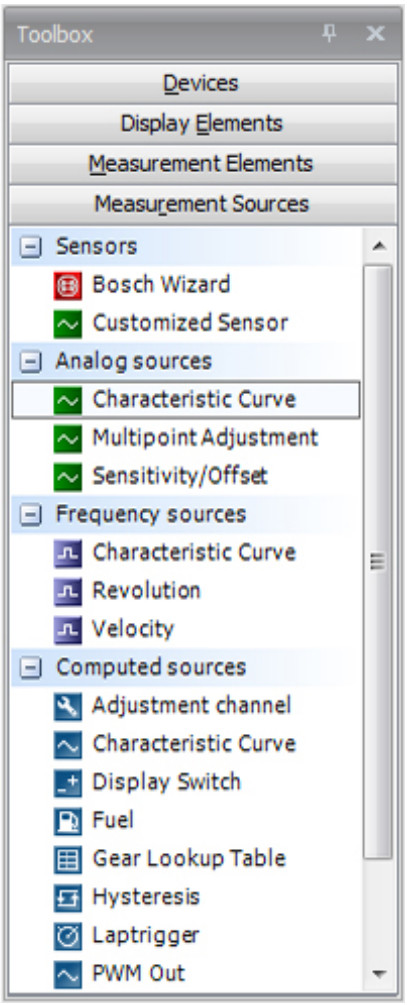
...

Switch 6 is shown with 5.5 <= the value

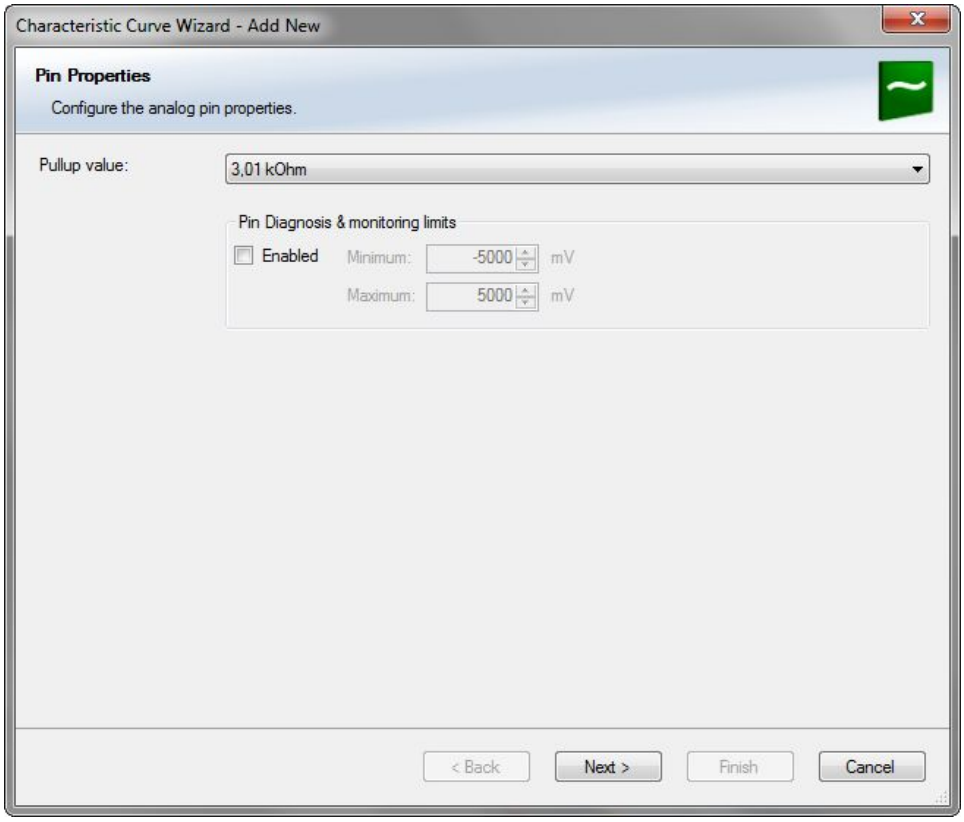
8.6.2 Option 1: Hardwired positions switch

1. Connect a 6- or 12-position switch to one of the analog input pins ANAxx and to the sensor ground. For recommended position switches, please see chapter Upgrades [▶ 11].
2. Select one of the analog inputs in the project tree.
3. Go to 'Measurement Sources' in the toolbox and select the 'Characteristic Curve' under 'Analog sources'. Drag and drop it on the selected analog input channel in the project tree.



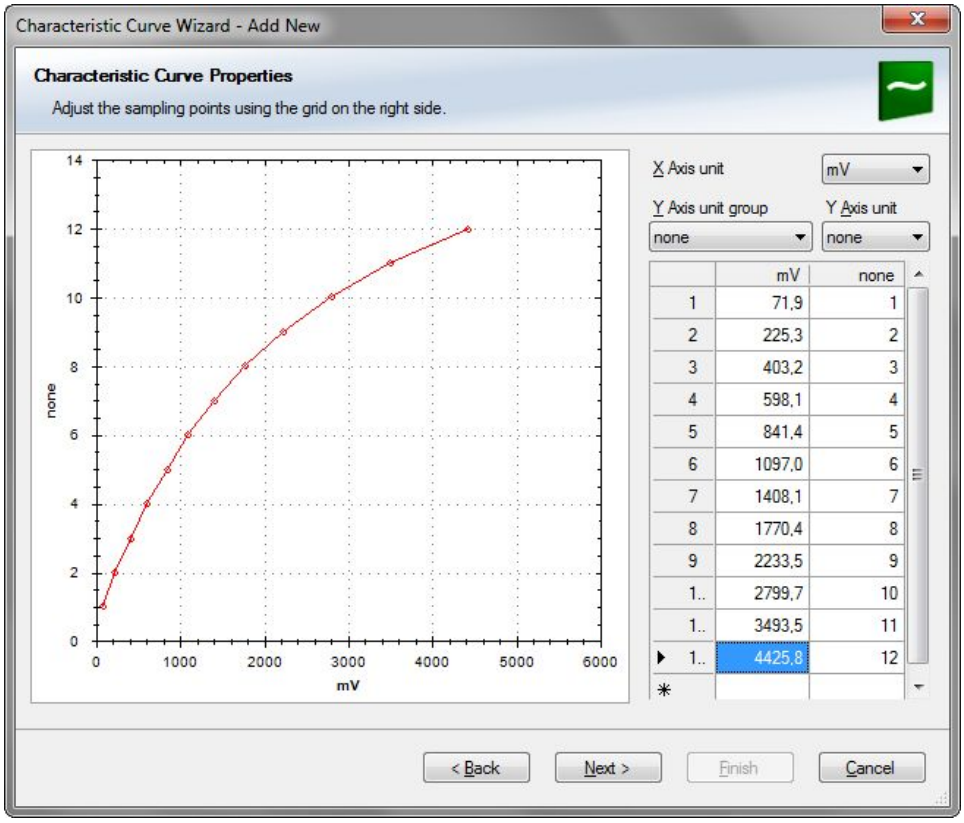


4. Select "Pullup value:" 3.01 kOhm and click on 'Next'.

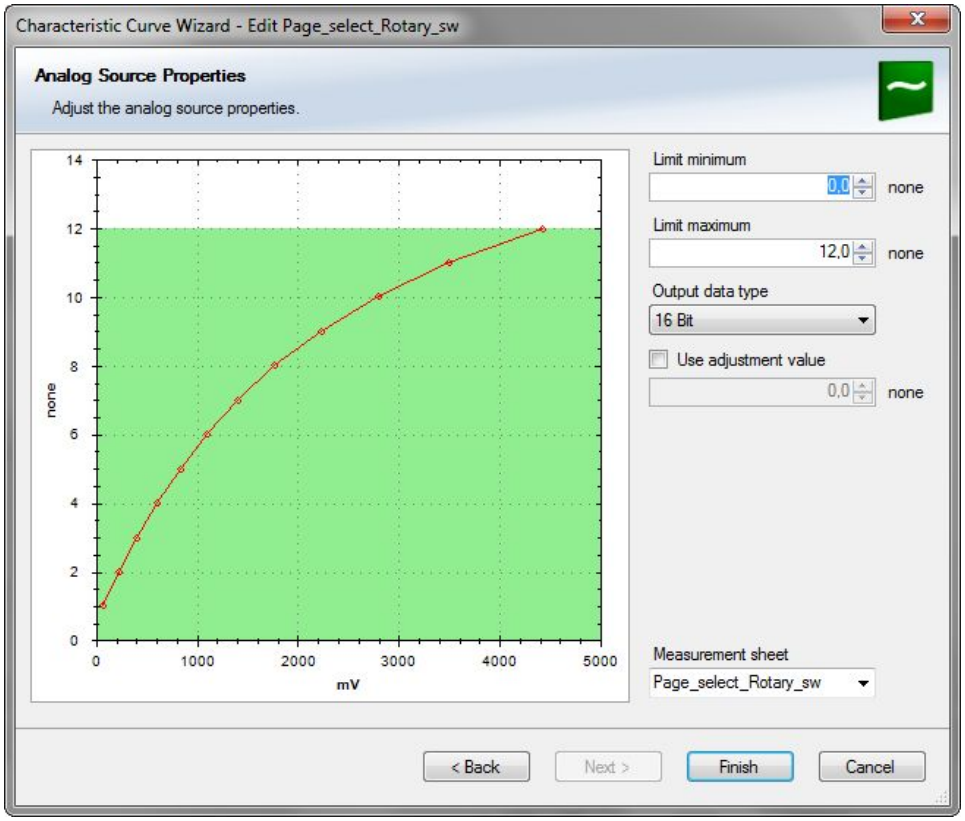


5. Define the relation between voltage and switch position and click on 'Next'.
   Voltage = 5000 x R/(R + 3010)
   5000: Sensor supply (mV)
   R: Resistor for each Rotary switch position (Ohm)
   3010: Pull-up resistor (Ohm)

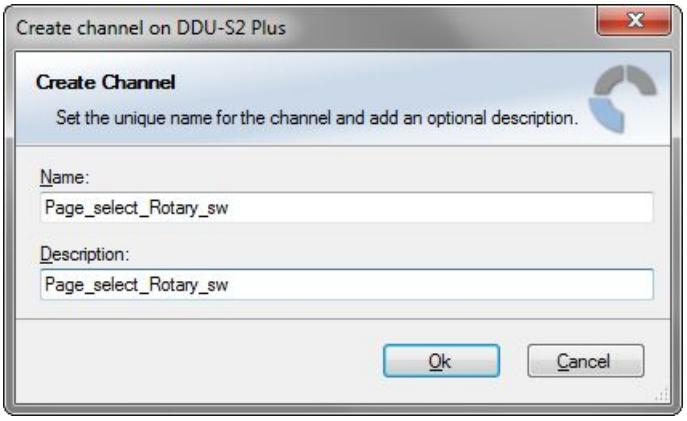
   The following screenshot and the data are an example for a Bosch switch.

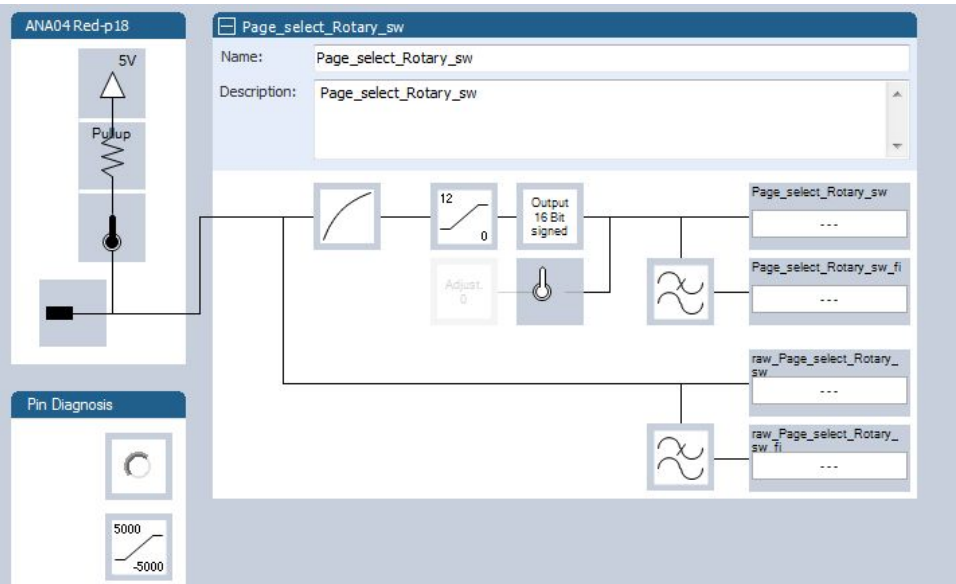


6. Define minimum and maximum Limit.
   Select “Output data type” from 8, 16 or 32 Bit.
   Do not check “Use adjustment value”.
   Choose the Measurement sheet and click on ‘Finish’.

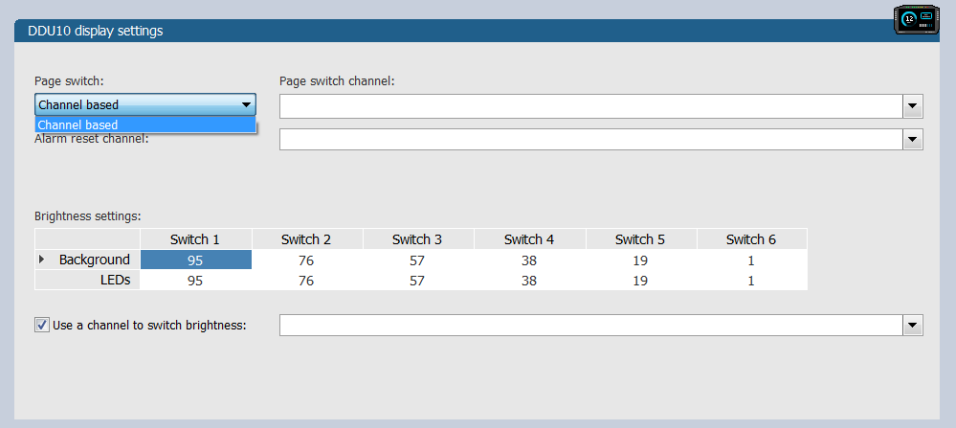


7. Define Name and Description and click on ‘Ok’.

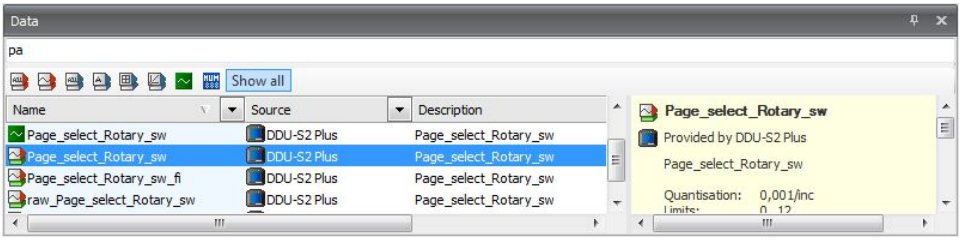




8. Click on the ‘Display’ tab and select the ‘Settings’ tab at the bottom.
9. To use the channel as a Page Switch, Select "Channel based" as page switch and select the channel configured above.
10. To use the channel as a brightness switch, check the box "Use a channel to switch brightness" and select the channel configured above.

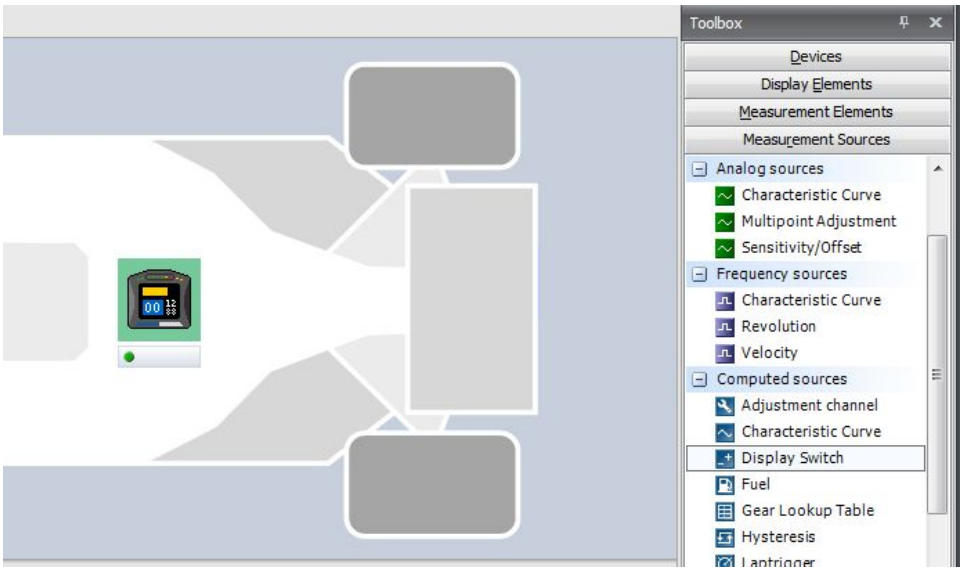


11. Click the ‘Download’ button at the top, to download the configuration.
12. If you want to check your configured channel, ensure that the device status is green, search for the configured channel in the Data window and double-click on it.
    You will see a graphical display with the raw and the physical value of the channel.

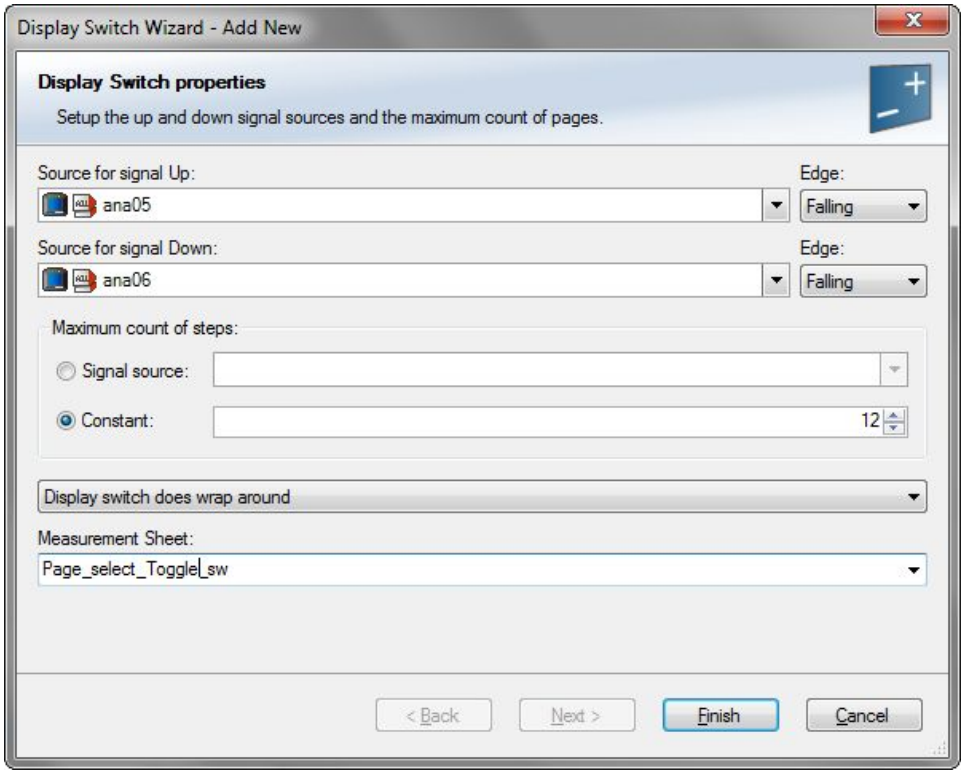


8.6.3 Option 2: Up/Down switches

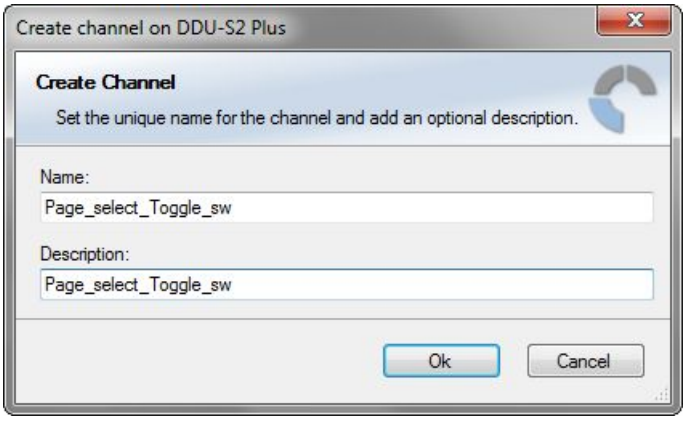
1. Define either one signal for a wrap around switch or two signals for an up/down switch.
2. Select the 'Display Switch' and drag it into the DDU 11.

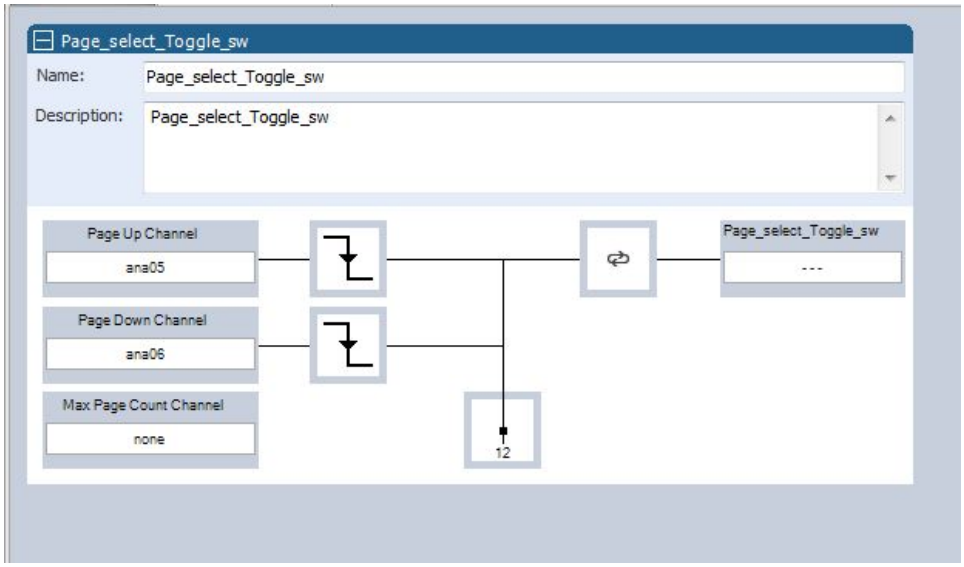


3. Select the Source for signal Up/Down and Edge Falling or Rising.
   Select the Maximum count of steps from signal source or from constant.
   Select "Display switch does wrap around" or "Display switch does not wrap around".
   "Display switch does wrap around" goes from maximum position to minimum position or the other way around (by switching) in a loop, after the last page it starts again with the first page.
   If you choose "Display switch does not wrap around", you need two switches to turn the pages in both directions.

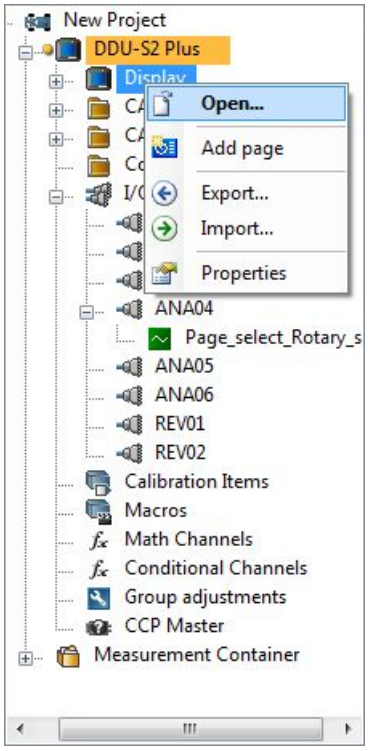


4. Define Name and Description and click on 'Ok'.

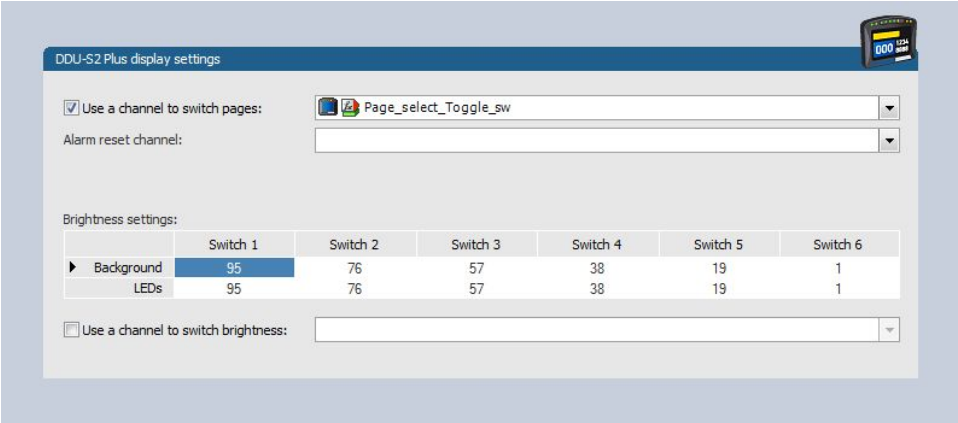




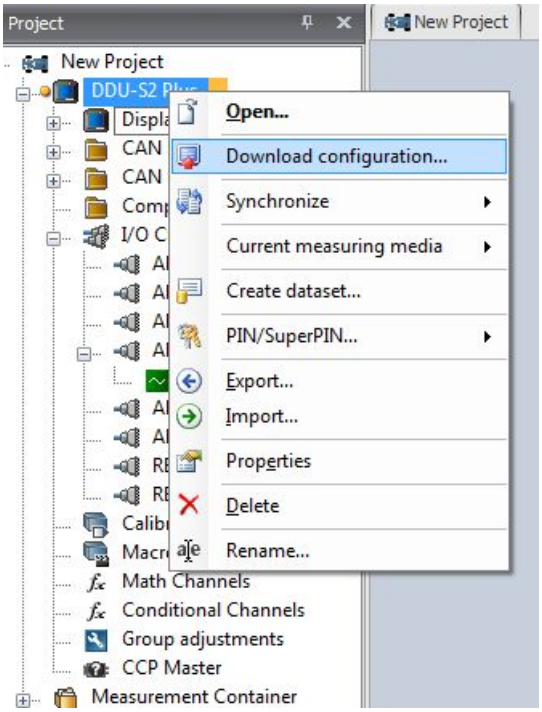
5. In the project tree, select "Display" and then "Open".



6. To use the channel as a Page Switch, check the box "Use a channel to switch pages" and select the channel configured above.
7. To use the channel as a brightness switch, check the box "Use a channel to switch brightness" and select the channel configured above.

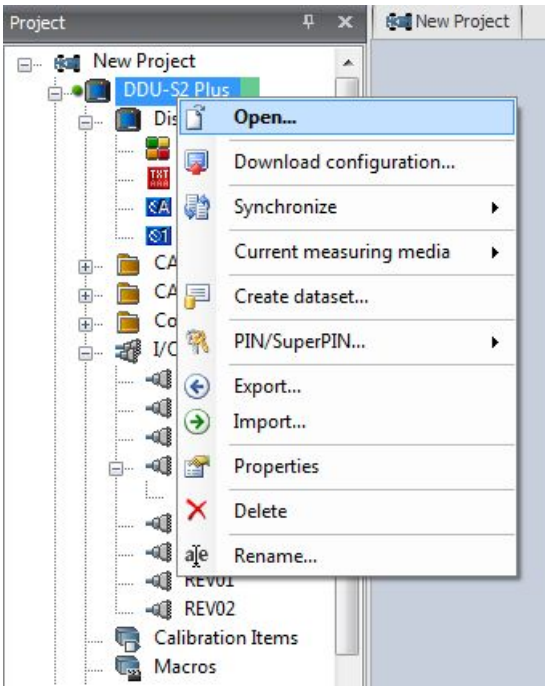


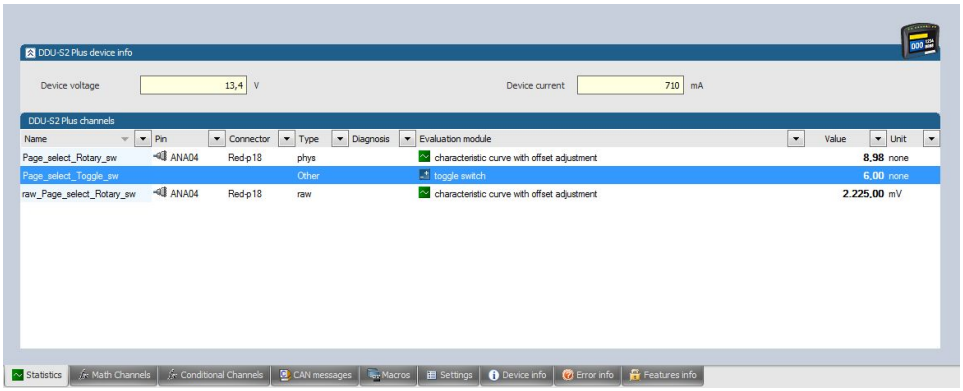
8. In the project tree, select “Download configuration”.



9. If you want to check your configured channel, reassure that the device status is green, double-click on the DDU 10 and select the "Statistics" tab.

   The configured channel position opens.

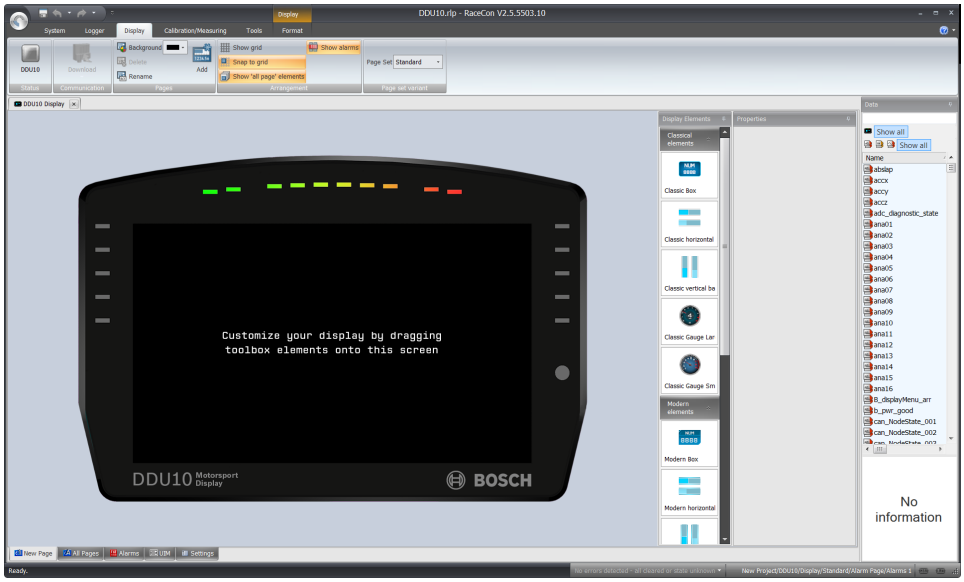




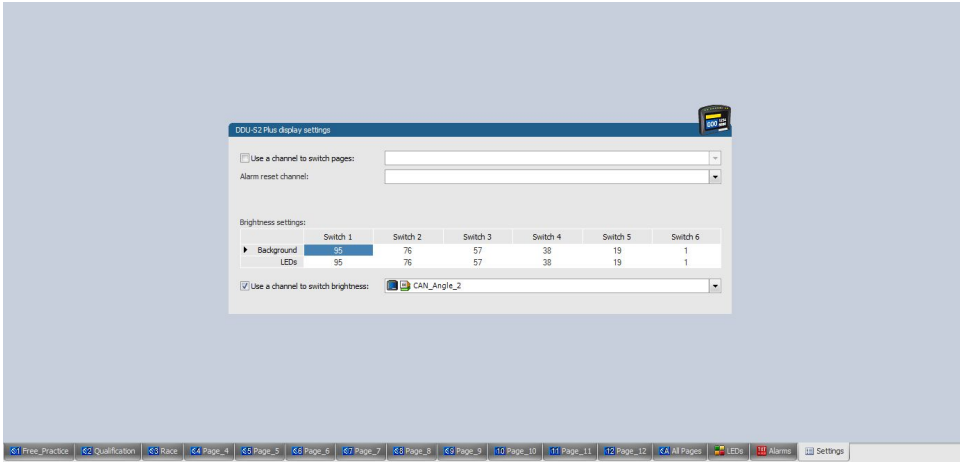
8.6.4 Option 3: CAN input signal, math channel or ANA_IN channel

Define your CAN, math or analog input channel.

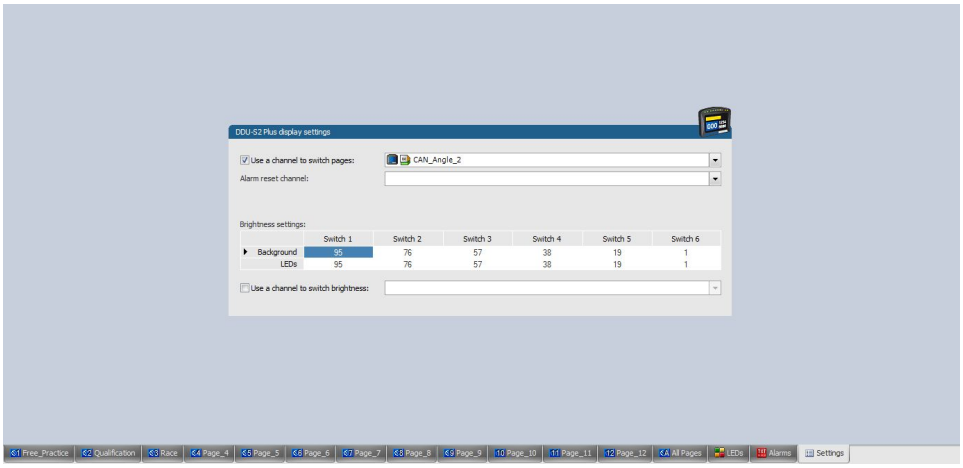
1. Select the 'Display' tab and then the 'Settings' tab at the bottom.



2. A ‘display settings’ window opens.



3. To use the channel as a brightness switch, check the box "Use a channel to switch brightness" and select the channel configured above.



4. To use a channel as a Page Switch, check the box" Use a channel to switch page" and select the channel configured above.
5. Click on the ‘Download’ button in the upper left corner.

8.7 Alarms

The ‘Alarm’ display element displays a warning message, when a defined condition becomes ‘true’. When a condition becomes ‘false’, the ‘Alarm’ display element disappears.

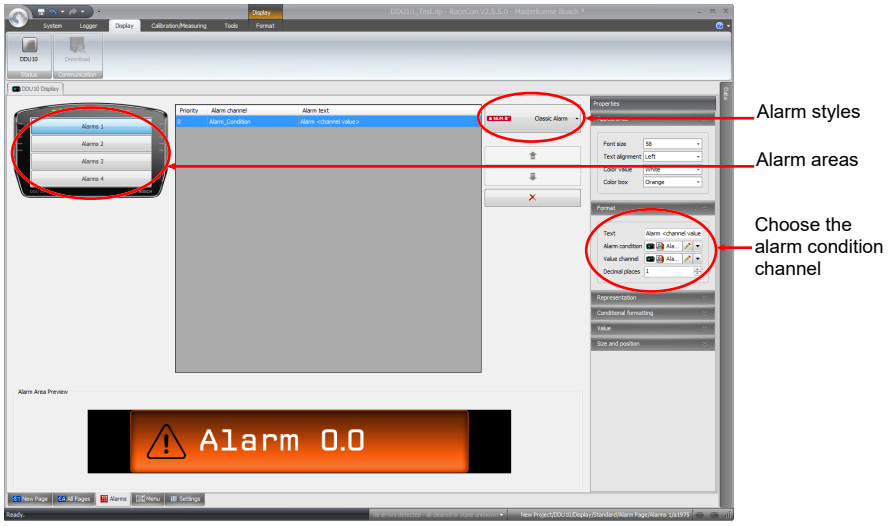
8.7.1 General information on the Alarm Representation

- Since an alarm is generally used for an urgent information, the ‘Alarm’ element has the highest priority and will always be displayed on top of other elements.
- The ‘Alarm’ elements can be defined in four separate areas. In each area, only one alarm message can be visible at the same time. If more than one alarm message is active in one area, each one will be visible for 1.5 seconds.
- The size and position within the alarm area is freely configurable.
- The ‘Alarm’ elements match the design styles of the general display elements and are available in the design styles “Classic” and “Modern”. For better readability, both styles have a non-transparent background.
- An ‘Alarm’ element can display either a text, icon or a value, or any combination of those separately.

8.7.2 Alarm configuration

To create an alarm, perform the following steps:

1. Click on the ‘Display’ tab at the top.
2. Click on the ‘Alarm’ tab at the bottom. The alarm configuration window opens.



3. Choose one of the four alarm areas at the top left.

   This defines the position on the screen, where the alarm will be displayed.

4. Select the design of the ‘Alarm’ element from the drop-down list on the top right.

   You can choose between the “Classic” and the “Modern” style. By clicking on the alarm style, an alarm will be added to the alarm list in the middle of the screen.

5. Select the “Alarm condition” channel, which will trigger the alarm when the channel becomes true.

   The “Alarm condition” channel can be any Boolean channel, like any other conditional channel.

6. If desired, select a “Value Channel”.

   The value can be displayed or used for a conditional formatting.

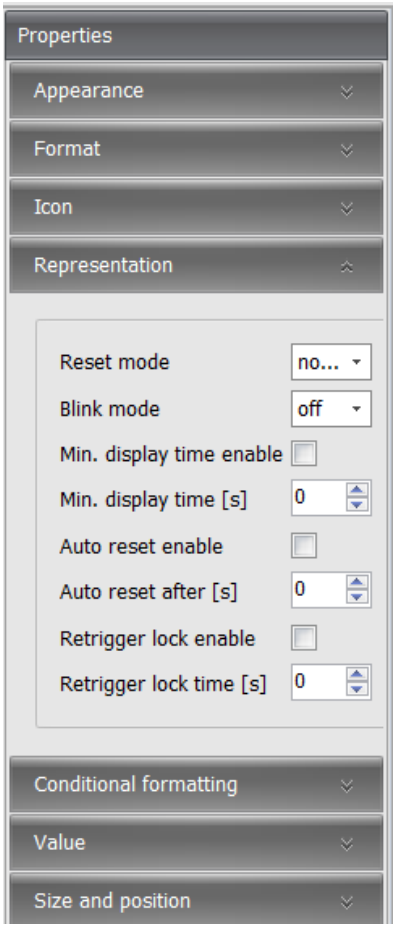
7. Define the text, which will be displayed in the alarm icon.

   If you use the phrase <*channel value*>in the text, it will be substituted with the value of the chosen channel value.

8. Check the size, position and appearance of the alarm in the “Alarm Area Preview”. Change the appearance according to your needs with the properties described below.

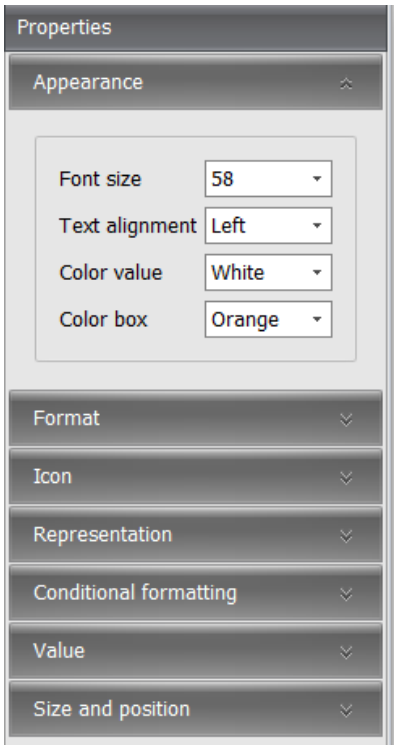
8.7.3 Further Alarm Properties

Representation



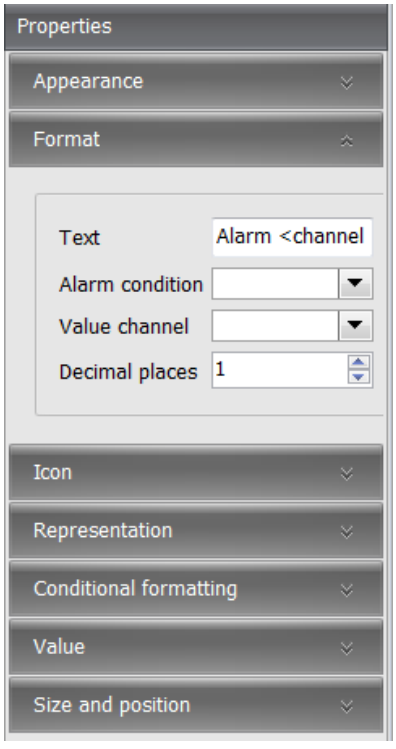
- Reset Mode: Defines if the alarm is resettable. For more details, see chapter Resettable Alarm [▶ 76].
- Blink mode: Defines the blinking frequency of the alarm when displayed.
- Min. display time enable: Enables the function min. display time.
- Min. display time: Defines the minimum time an alarm will be displayed.
- Auto reset enable: Enables the function “Auto reset”.
- Auto reset after: Defines the time after an alarm will be hidden, even when the condition is still true, only active for resettable alarms.
- Retrigger lock enable: Enables the function “Retrigger lock”. For more details, see chapter Resettable Alarm [▶ 76].
- Retrigger lock time: Defines the time in which a new rising edge of the alarm condition will not be accepted anymore, after an alarm was reset.

Appearance



- Font size: Defines the size of the font.
- Text alignment: Defines the alignment of the text.
- Color value: Defines the color of the value inside the alarm box.
- Color box: Defines the color of the alarm box.

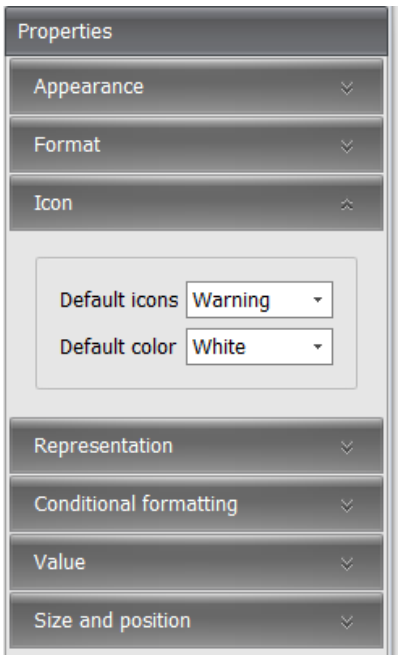
Format



- Text: Defines the Text and/or the value of the alarm.

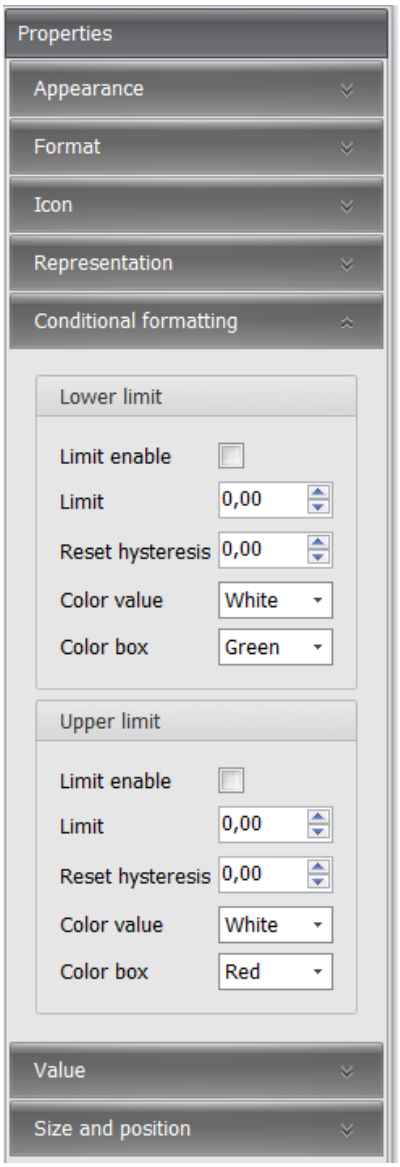
- Alarm condition: Defines the alarm condition.
- Value channel: Defines the value channel, which may be displayed and is relevant for the conditional formatting.
- Decimal places: Defines the decimal places, which should be displayed.

Icon



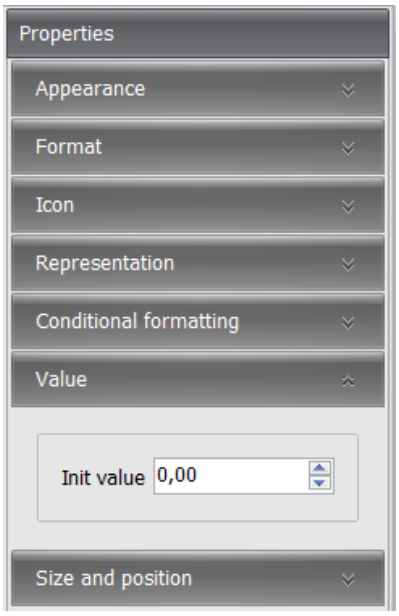
- Default icons: Defines the alarm icon, which should be displayed.
- Default color: Defines the color of the icon.

Conditional formatting (lower and upper limit respectively)



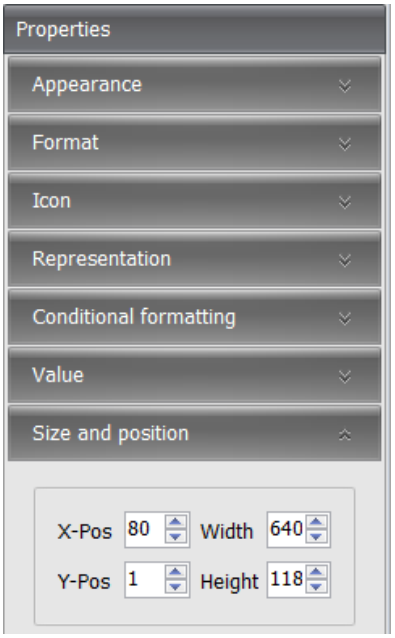
- Limit enable: Enables a lower limit and an upper limit respectively.
- Limit: Defines the threshold limit.
- Reset hysteresis: Defines the value as off when the hysteresis should be reset. You can find more information about the hysteresis function in general in the chapter Hysteresis [▷ 109].
- Color value: Defines the color when the value is above/below the set threshold limit.
- Color box: Defines the new color of the box when the value is above/below the set threshold limit.

Value



- Init value: Defines the value, which will be shown in RaceCon during configuration or on the device if the channel has no link. Allows testing the conditional formatting by entering a representative value.

Size and position



- X-Pos: Defines the x-position of the top left corner of the alarm message within the alarm area.
- Y-Pos: Defines the y-position of the top left corner of the alarm message within the alarm area.
- Width: Defines the width of the alarm message.
- Height: Defines the height of the alarm message.

8.7.4 Edit/Delete/Change priority of an Alarm

Perform the following steps:

1. Click on the 'Display' tab and then on the 'Alarm' tab to go to the alarm configuration window.
2. Select the alarm area of the desired alarm.

   The center of the alarm configuration window only shows the alarms of the selected alarm area.
3. Click on the alarm, which needs to be edited or deleted.
4. To delete the alarm, click on the red cross.
5. Change the properties of the alarm in the properties window on the right.
6. To change the priority of the alarm, click the up/down arrow to rearrange the priority list.

   If the option "Show alarms" is ticked, the alarm with the highest priority will be visible on the display page configurator. The visibility of the alarms will not be influenced by the priority; the alarms will always rotate, if several conditions become true.

8.7.5 Resettable Alarm

An alarm can be configured as a resettable alarm. This means that the alarm can be linked to a second conditional channel, which will hide the alarm, when the condition becomes true.

To configure and use a resettable alarm, perform the following steps:

1. Click on the 'Display' tab and then on the 'Settings' tab to go to the "DDU10 display settings"
2. Select your "Alarm reset channel", which can be any boolean channel, like any other conditional channel.
3. Click on the 'Alarms' tab, to change to the alarm configuration window.
4. Create an alarm or edit an existing alarm as described in the previous chapters, to access the properties "Representation".
5. Change the "Reset mode" to resettable. The alarm will be displayed with a little cross at the top right corner.

Important information on resettable channels:

- Only visible alarms can be reset. If several alarms are active, only the alarm which is visible during the reset condition becoming true, will react to the reset.
- A reset alarm will not be shown again, if the alarm condition becomes true again, until the next power cycle. Exceptions are, a change of the display page or if the retrigger lock is enabled and the retrigger lock time is over.

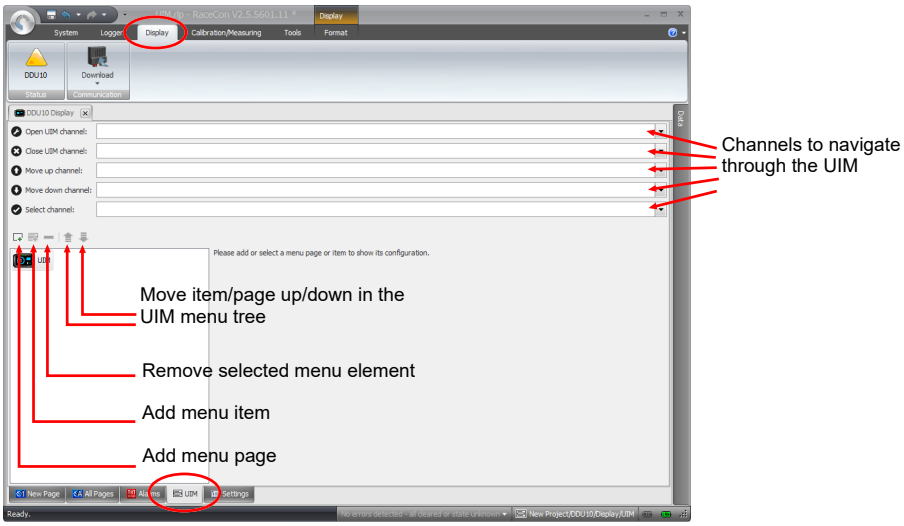
8.8 User Interface Menu UIM

You can use the User Interface Menu (UIM) to conduct minor setting changes on the DDU, like lap time or fuel reset.

How to configure the UIM

Perform the following steps, to set up an UIM on your device:

1. Click on the 'Display' tab on the top or double-click on "Display" in the project tree.
2. Click on the 'UIM' tab at the bottom. This will open the UIM configurator.



3. Define the channels at the top, to navigate through the UIM. The selected channels can be any Boolean channels, such as conditional channels. You can also use condition combinations to open the UIM, to avoid an unintentional start of the UIM.

NOTICE

When the close condition is true, you cannot open the UIM. This function can be used to avoid an unwanted UIM while driving, i.e with a condition RPM>300

4. Click "Add menu page" to create a new link to a new page on the selected page.
5. Click "Add menu item" to create a new item on the selected page.

   On a new page, you can create new items or pages as needed.
6. Define a name for your items and pages and define the settings for your items.

   With a new item, you can define either labels, special functions or menu triggered events.

Labels

– With a label you can show any channel of the current project or just a text.

Special Functions

– Page set variation

  This menu entry enables you to switch between all available page sets. Activating a different page set automatically reboots the device.

– Outing counter increment

  Increments the outing counter by 1.

– Lap counter increment

  Increments the lap counter by 1.

– Lap distance reset

  Resets the traveled lap distance to zero.

– Lap counter reset

Resets the lap counter to the reset value configured in the Laptrigger options.

– Outing counter reset

  Resets the outing counter to the reset value configured in the Laptrigger options.

– Best laptime reset

  Resets the best laptime.

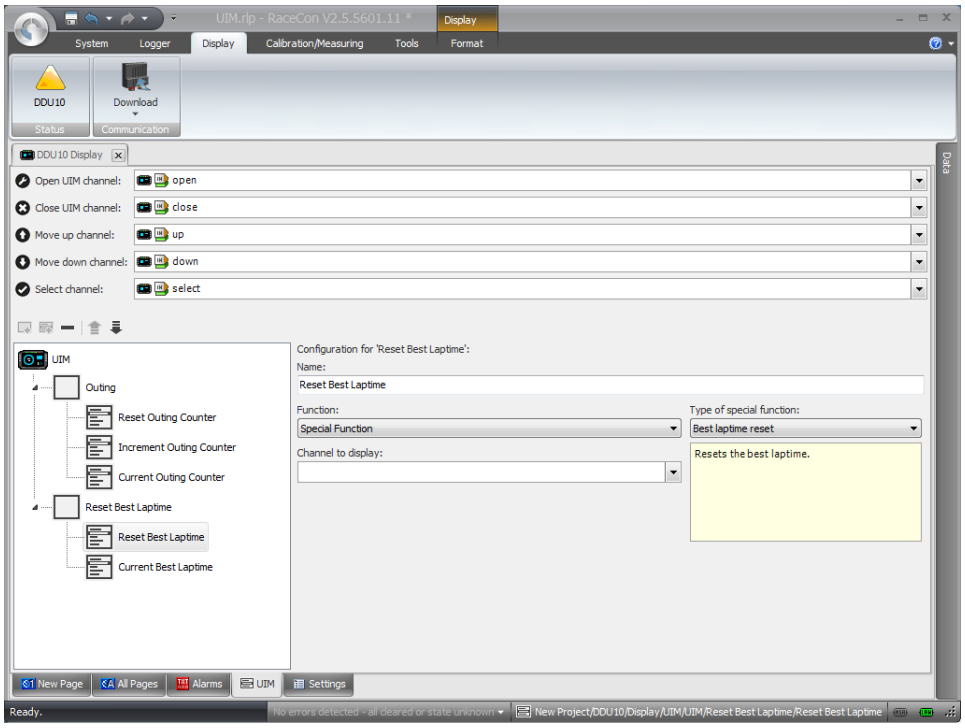
– Segment times reset

  Resets the best laptime segment times.

Menu Triggered Events

– With a menu triggered event you can define a new virtual Boolean channel, which will jump to the condition true for 100 ms, when selected through the UIM. You can use such a channel for sensor calibration or any further custom calculation.

  A UIM configuration to change the current outing counter and to reset the best lap time could look like this:



9 CAN Configuration

The DDU 11 has 2 (plus 2 on request) fully configurable CAN bus(es).

- Baudrate 125 kbaud to 1 Mbaud
- 11 Bit or 29 Bit identifiers
- Input configuration: Read messages from CAN bus and convert to DDU 11 measurement/display variables. CAN bus supports row counter configuration.
- Output configuration: Write RaceCon measurement variables to CAN messages; output frequency and row counter are configurable, CAN gateway functionality (transfer from one bus to another).

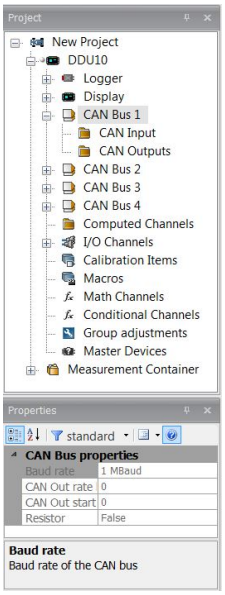
9.1 CAN Bus Trivia

CAN Message

- 11 Bit (standard) or 29 Bit (extended) identifier
- Up to 8 bytes of data payload

CAN Bus

- Needs termination resistors in wiring harness
- All devices connected to the bus must use identical data rate
- Configuration of bus data rate in the 'CAN messages overview' menu. To access the menu, double-click on one of the CAN bus items of the project tree



Row Counter Concept

- Re-use (multiplex) of message identifiers
- One byte of message contains row counter
- 7 bytes payload remaining
- Position of row counter is configurable

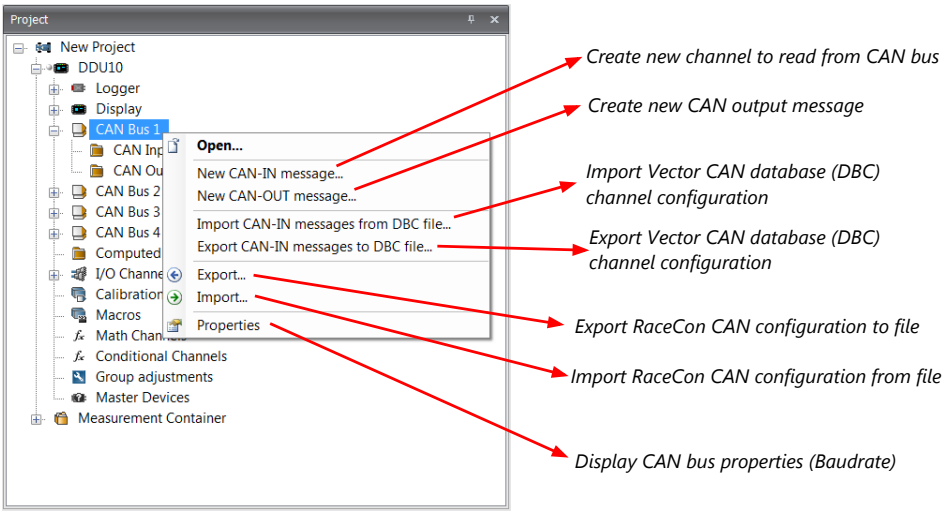
	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x100	0	p_oil		t_oil				
0x100	1	s_dam_fl		s_dam_fr				
0x100	2	s_dam_rl		s_dam_rr				

Message Id ***Row Counter*** ***Payload Area***

9.2 CAN input

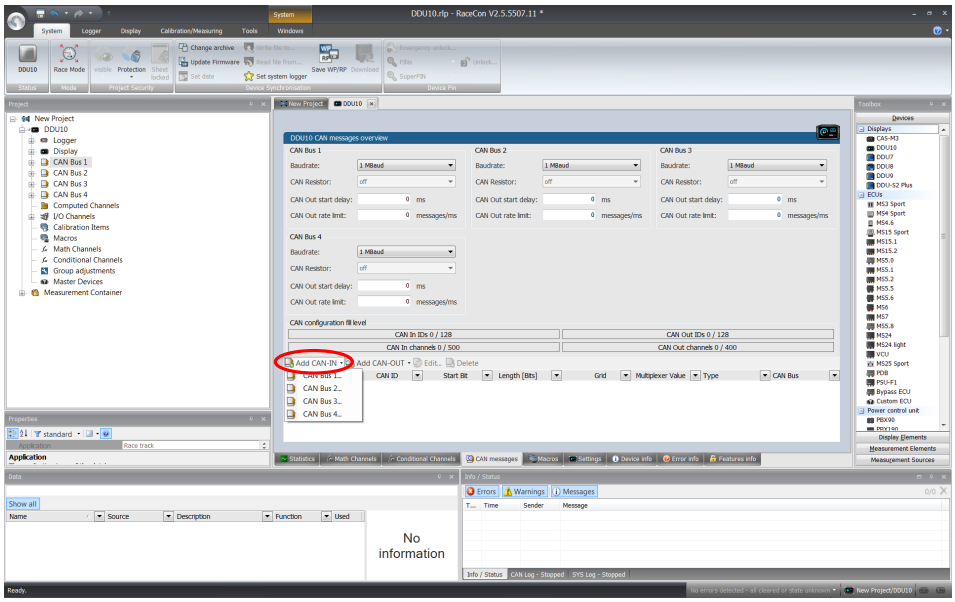
9.2.1 Input configuration

Click with the right mouse button on the desired CAN bus to open the CAN bus drop-down menu.



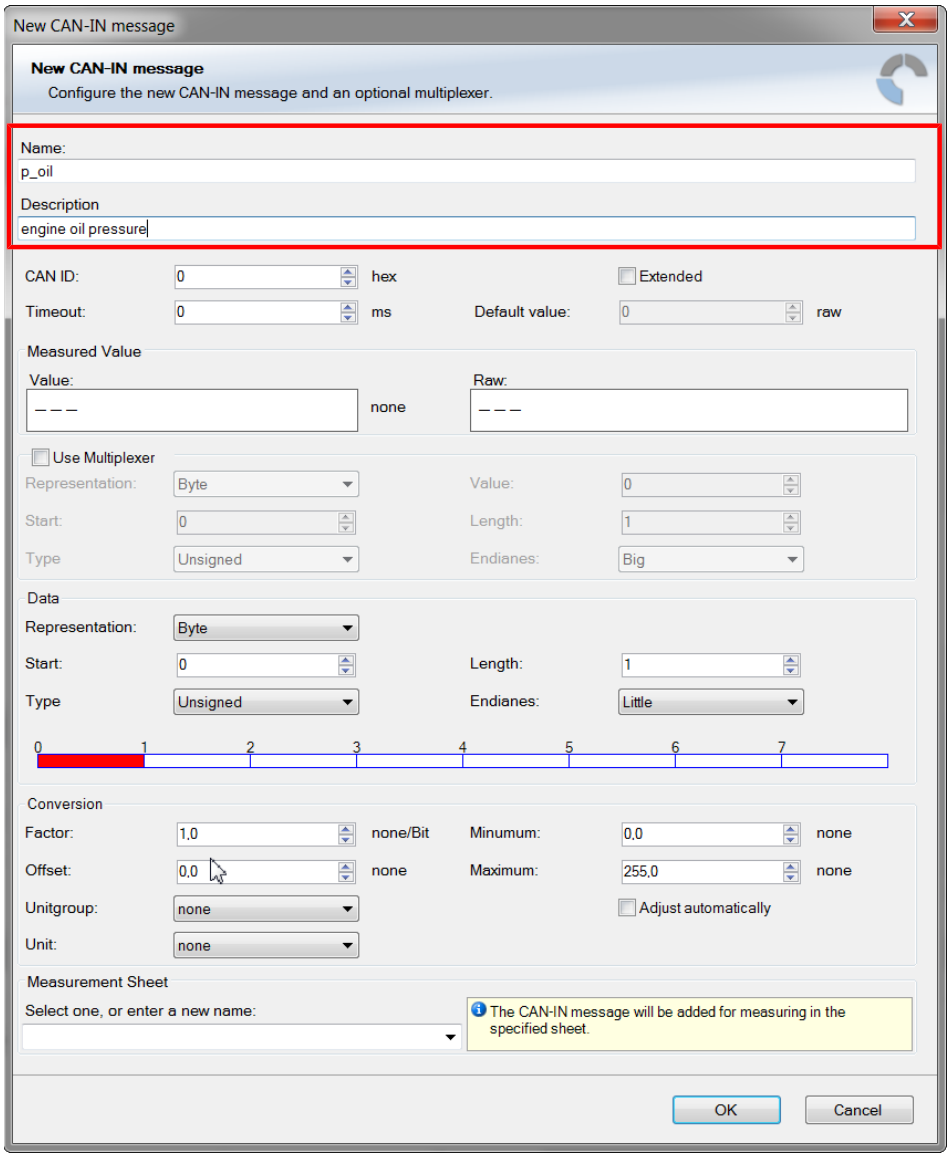
9.2.2 Create new CAN Input channel

1. Double-click on any CAN bus item, to open the "CAN messages overview".
2. Select ‘Add CAN-IN’ and choose the desired CAN bus for the new input channel.

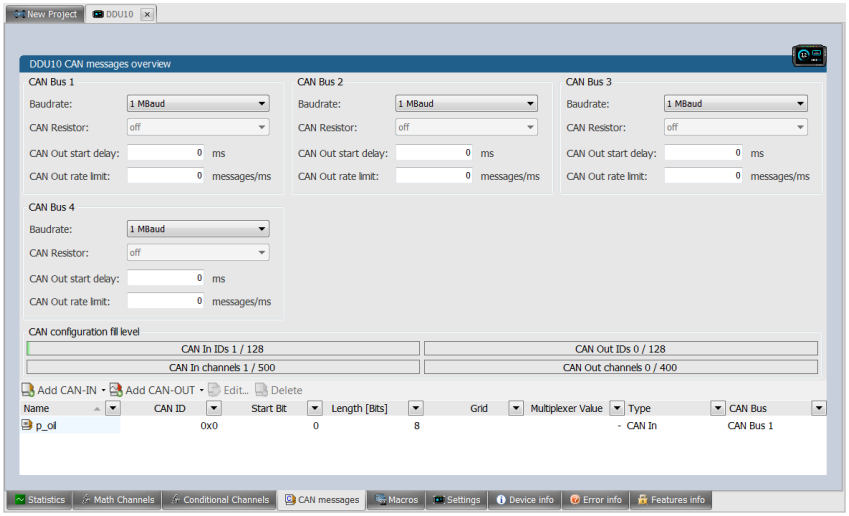


3. A CAN channel configuration window opens.

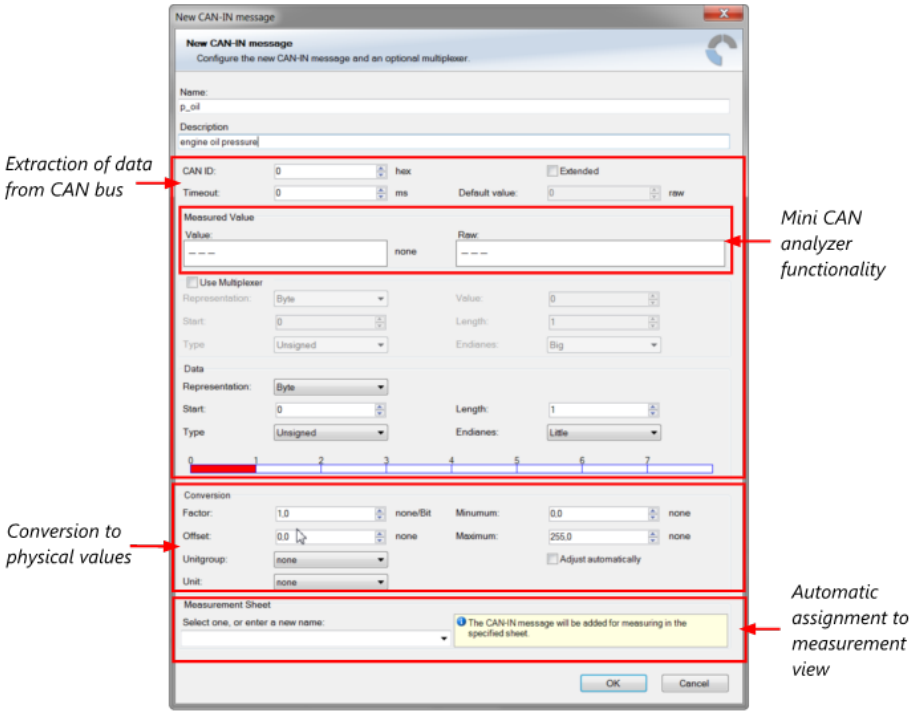
4. Insert the name and description of the channel.



5. Click ‘OK’ when done.
   The channel is listed in the Data window.



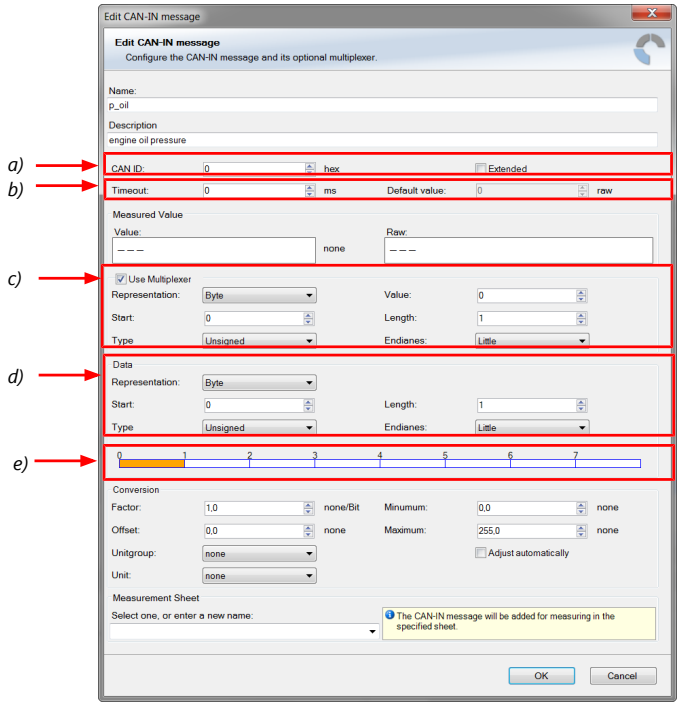
CAN channel configuration



9.2.3 Extracting data from CAN bus

Representation: Byte

Some CAN devices need to be addressed by a byte represented CAN channel. The address can be assigned in this window and is illustrated by a bargraph.



a) Enter CAN message ID. If extended IDs (29 bit) are used, check the box.

b) If replacement values are used, specify time-out period and raw value.

c) If a multiplexer (row counter) is used, check the box.

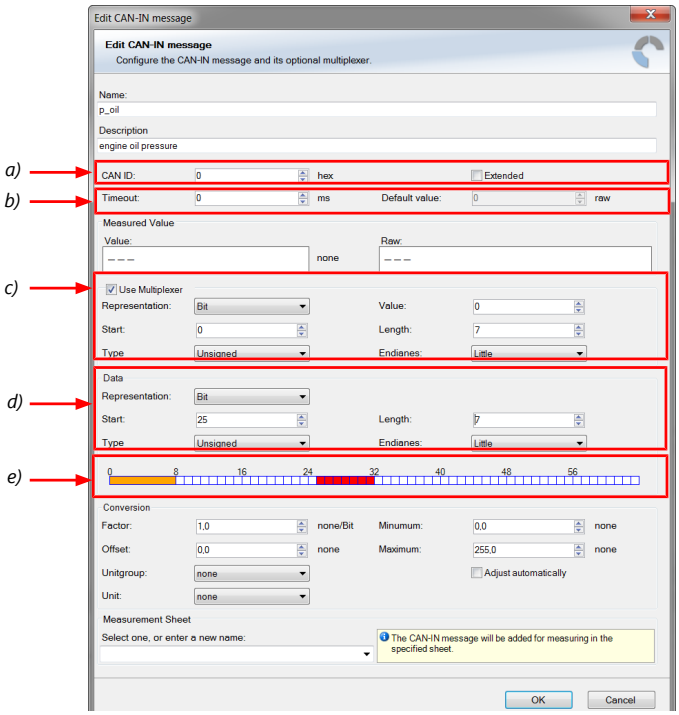
d) Enter data position, length and format.

e) The bargraph shows assignment of the bytes.

- Red colored fields show the assignment of the data bytes.
- Orange colored fields show the assignment of the multiplexer bytes.

Representation: Bit

Some CAN devices need to be addressed by a bit represented CAN channel. The address can be assigned in this window and is illustrated by a matrix table.



a) Enter CAN message ID. If extended IDs (29 bit) are used, check the box.

b) If replacement values are used, specify time-out period and raw value.

c) If a multiplexer (row counter) is used, check the box.

d) Enter data position, length and format.

e) The bargraph shows assignment of the bytes.

- Red colored fields show the assignment of the data bytes.
- Orange colored fields show the assignment of the multiplexer bytes.

Conversion to physical value

a) Enter factor (gain) for conversion to physical value.

b) Enter offset for conversion to physical value.

c) Select type of physical value.

d) Select unit of physical value.

e) Enter minimum physical limit of the channel. (for manual setup)

f) Enter maximum physical limit of the channel. (for manual setup)

g) Check the box to automatically adjust the limits of the channel.

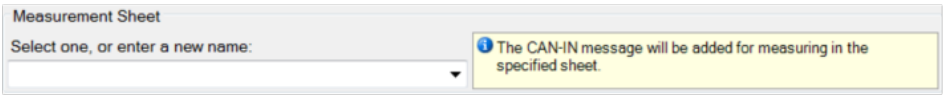
CAN analyzer functionality

This functionality is only available, if a MSA-Box (I or II) is used to connect the DDU 11 to the PC. Choose the CAN bus that is connected to the MSA-Box to display the raw value and the converted physical value here.

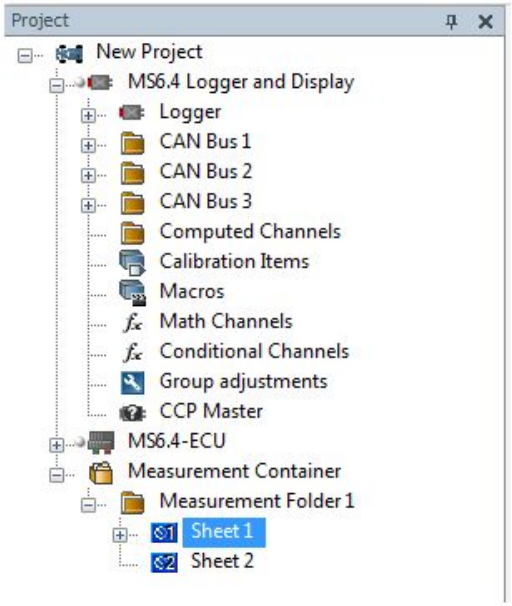
Automatic creation of online measurement sheets

The CAN channel can be automatically inserted into a measurement sheet. Insert a name for a new sheet or select an existing sheet from the list box.

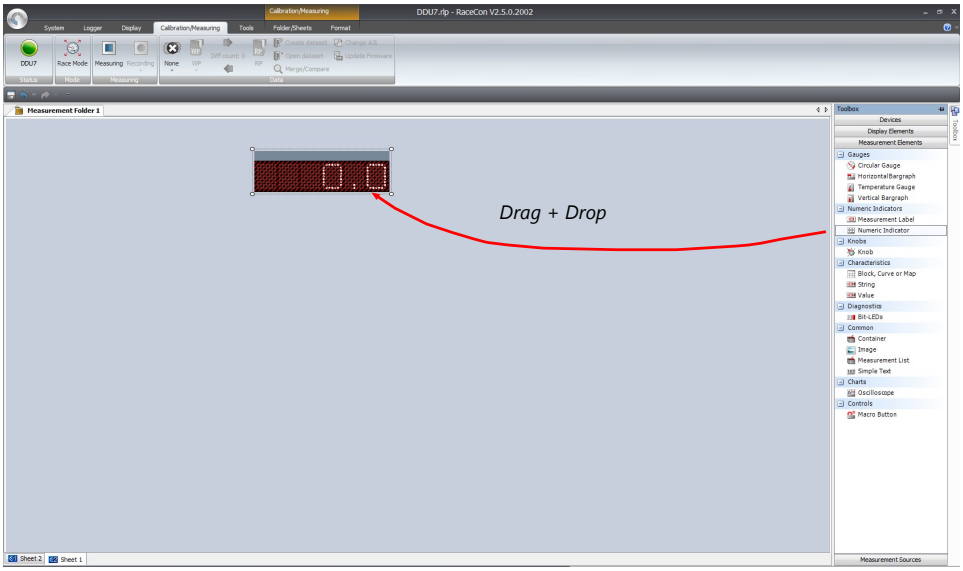
For an online view of the value measured by the DDU 11, insert the channel in an online measurement sheet which is described in the chapter Setting up an online measurement [▷ 113].



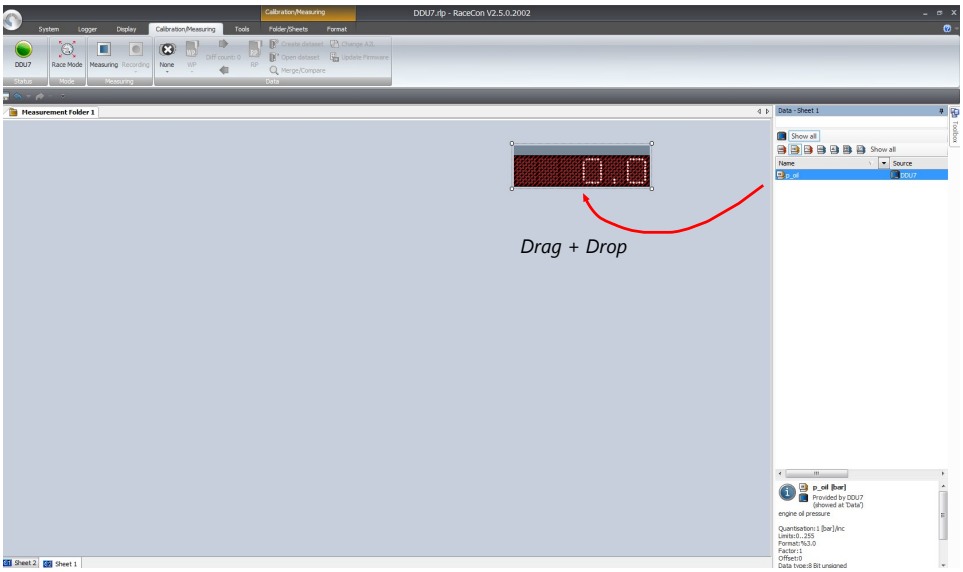
9.2.4 Online view of CAN channels in vehicle



1. Double-click on ‘Sheet 1’ in Project Tree. Measurement Sheet 1 is displayed in Main Area.
2. Click on ‘Measurement elements’ in the Toolbox.
3. Drag the desired Measurement element (e.g. Numeric Indicator) and drop it on the Measurement Sheet.



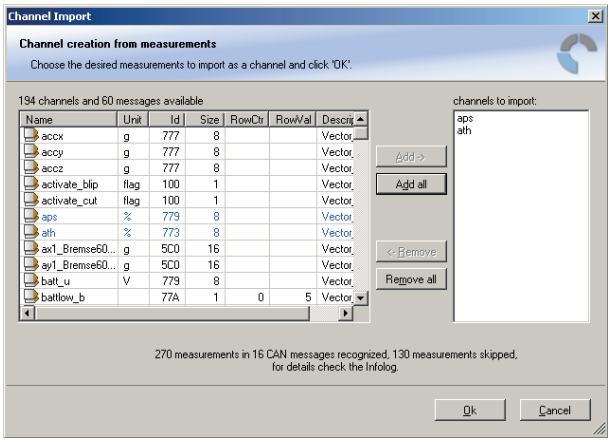
4. Click on folder ‘CAN Input’ of desired CAN bus to display available channels.
5. Drag desired Measurement channel and drop it on the Measurement element.



6. The measurement element displays the values of the assigned channel.
7. Connect PC to the vehicle and switch to ‘Race Mode’ by clicking ‘F11’ on the keyboard to display online data.

9.2.5 Import a CAN database (DBC) file

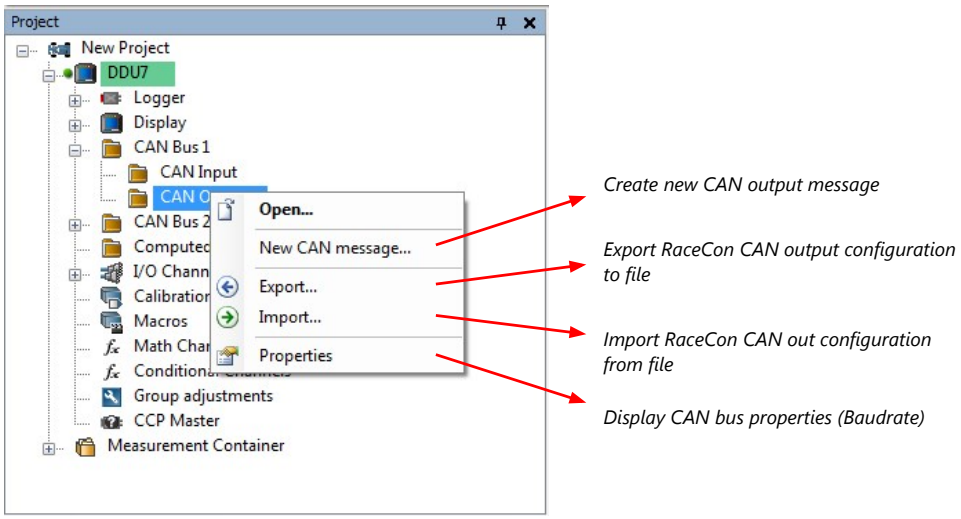
1. Right-click on CAN Input of desired bus (CAN1 or CAN2).
2. Select ‘Import DBC file’ from menu. A file browser opens.
3. Select DBC file to import and click ‘OK’ when done. A channel import window opens.



4. Select desired channels on the left and use the ‘Add’ button to add them to import list.
5. Click ‘OK’ when complete. The channels are inserted in the Data window.

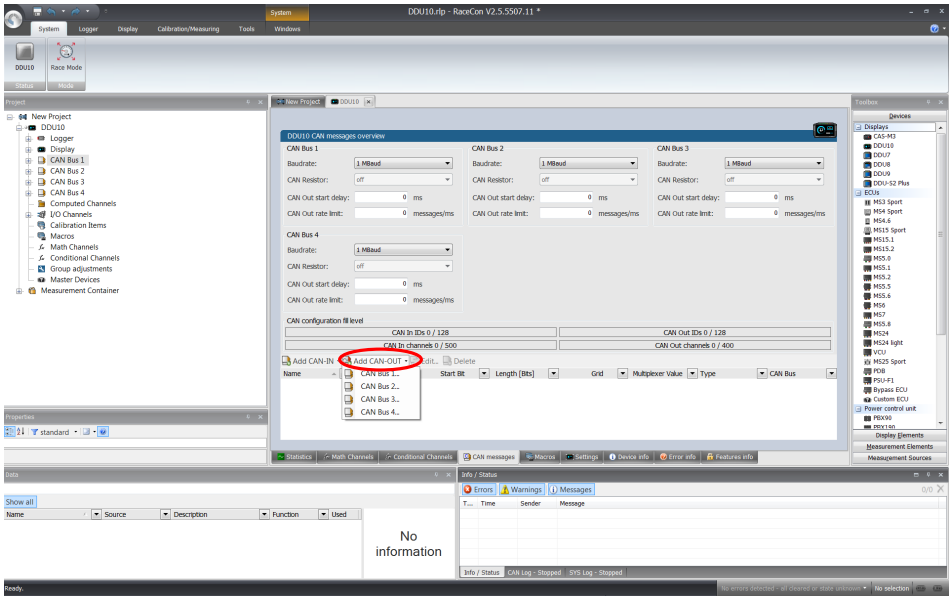
9.3 CAN output

9.3.1 Output configuration

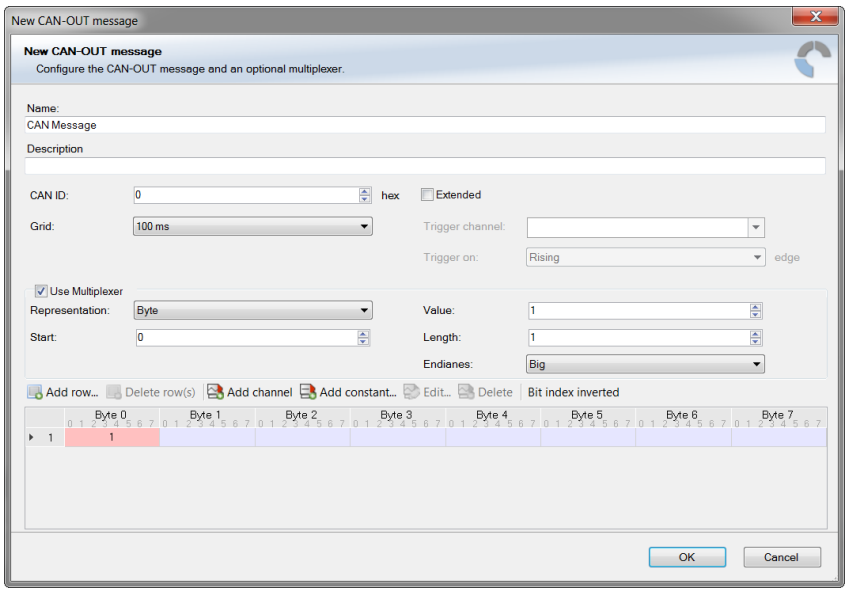


9.3.2 Create a new CAN output message channel

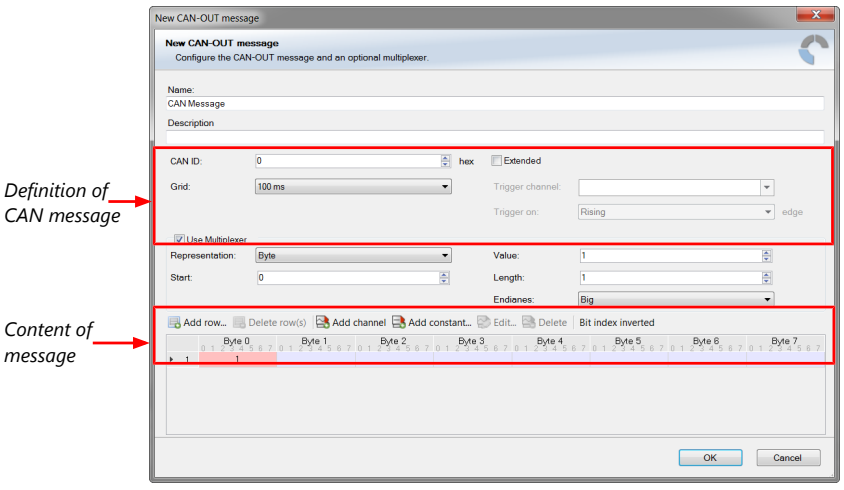
- Double-click on any CAN bus item to open the "CAN messages overview".
- Select ‘Add CAN-OUT’ and choose the desired CAN bus for the new output channel.



– The ‘New CAN-OUT message’ window opens.

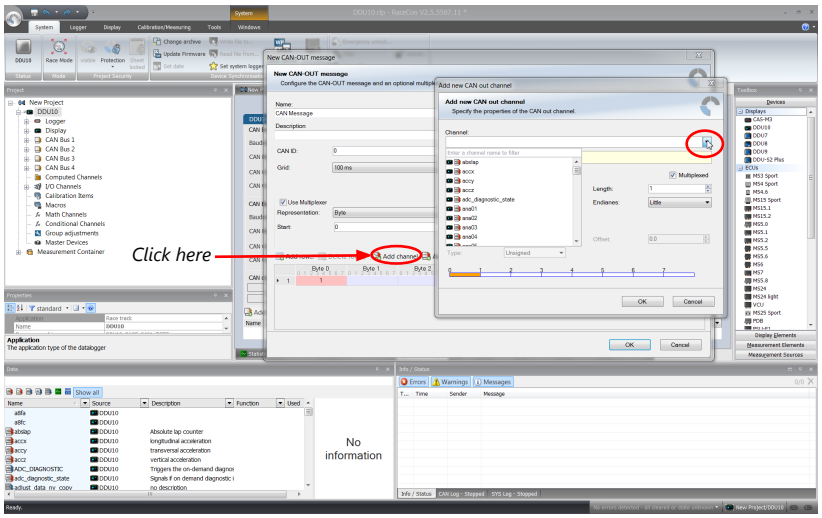


– Enter name of message, description, CAN-Id, and Grid (output interval). Optionally, specify a multiplexer.



– Click on ‘Add channel...’ or ‘Add constant...’, this opens the ‘Add new CAN out channel’ window.

– Select the desired measurement channel and specify the message settings.

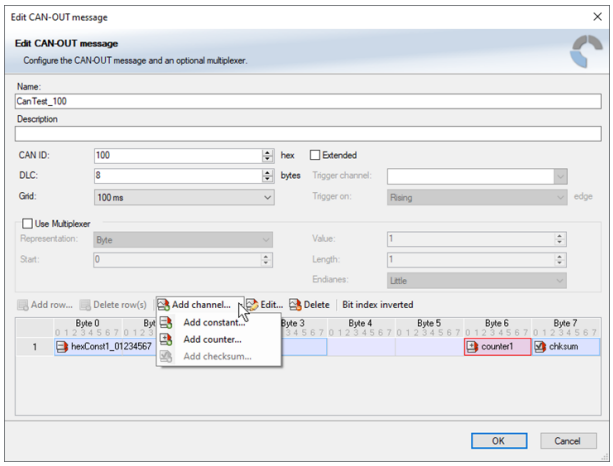


The measurement channel is now assigned to the CAN message.

9.3.2.1 Add CAN out constant

To send a constant value on the CAN, perform the following steps:

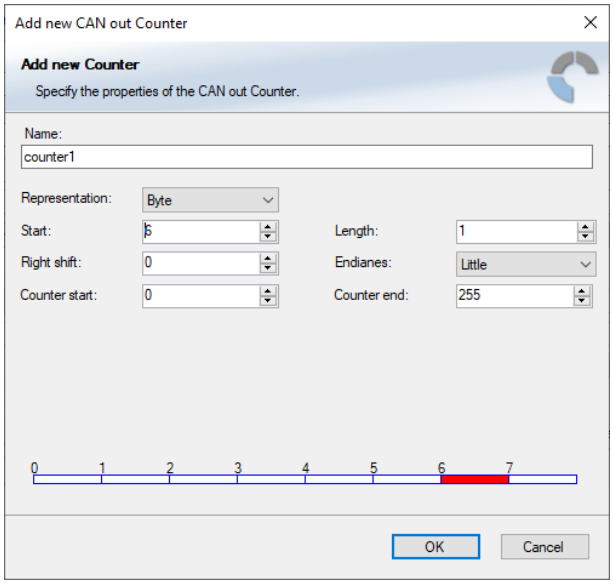
1. Create a new CAN output message or edit an existing message.
2. Click small arrow beside ‘Add channel...’ and select ‘Add constant...’. The ‘Add new CAN Out constant’ window appears.
3. Define the name of the constant, the required value in hex and define the CAN channel settings.
4. Click ‘OK’ when done.



9.3.2.2 Adding CAN out counter

To send a counter value on the CAN, perform the following steps:

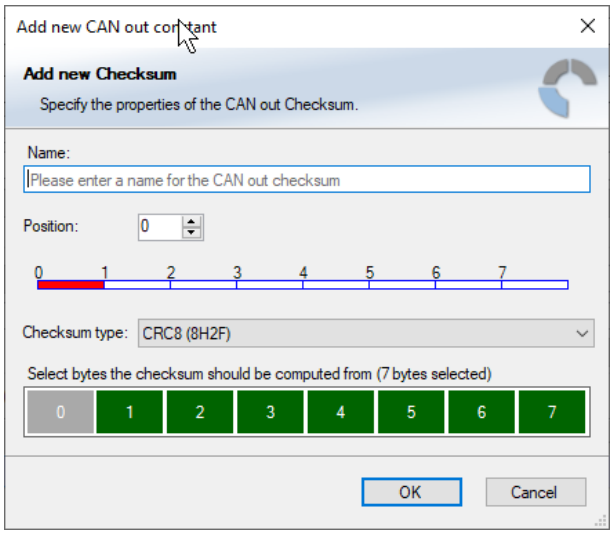
1. Create a new CAN output message or edit an existing message.
2. Click small arrow beside ‘Add channel...’ and select ‘Add counter...’. The ‘Add new CAN out counter’ window appears.
3. Define the name of the counter, define the CAN channel settings.
4. Click ‘OK’ when done.



9.3.2.3 Adding CAN out checksum

To send a checksum on the CAN, perform the following steps:

1. Create a new CAN output message or edit an existing message.
2. Click small arrow beside 'Add channel...' and select 'Add checksum...'. The 'Add new CAN out checksum' window appears.
3. Define the name of the checksum, the algorithm, the byte which should be covered by the checksum and define the CAN channel settings.
4. Click 'OK' when done.



9.4 Multiplexer

Row counter concept

If certain channel messages are not time-critical and can be imported or exported slowly, you can use a multiplexer to put several channel messages on one message identifier.

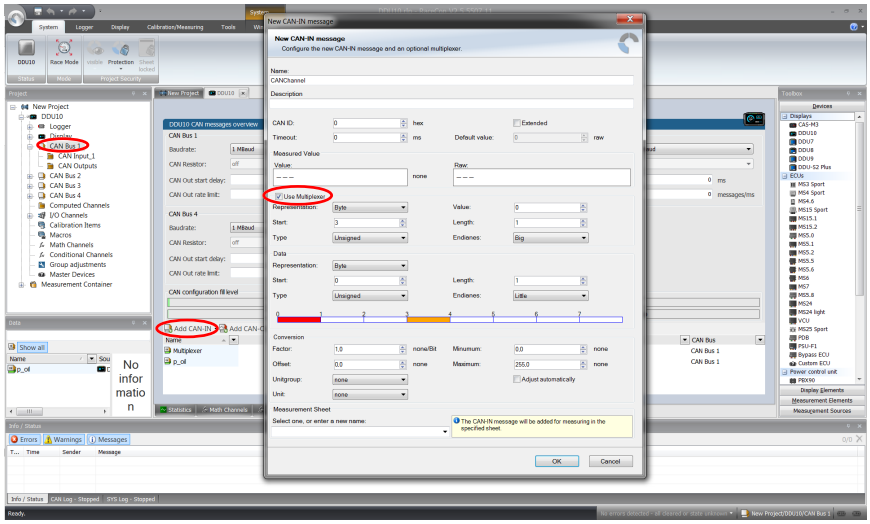
- Re-use (multiplex) of message identifiers by splitting it into several rows.
- Every row is assigned to a unique value of the multiplexer.
- One byte of message contains row counter.

– 7 bytes payload remaining. A multiplexer does not have to consist of one byte only, it can consist of several bytes as well as single bits.

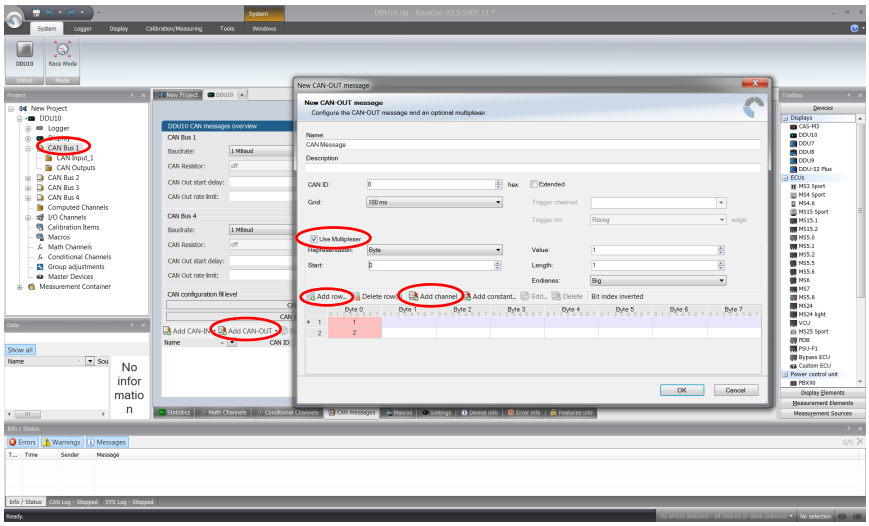
– Position of row counter is configurable.

To use a multiplexer perform the following steps:

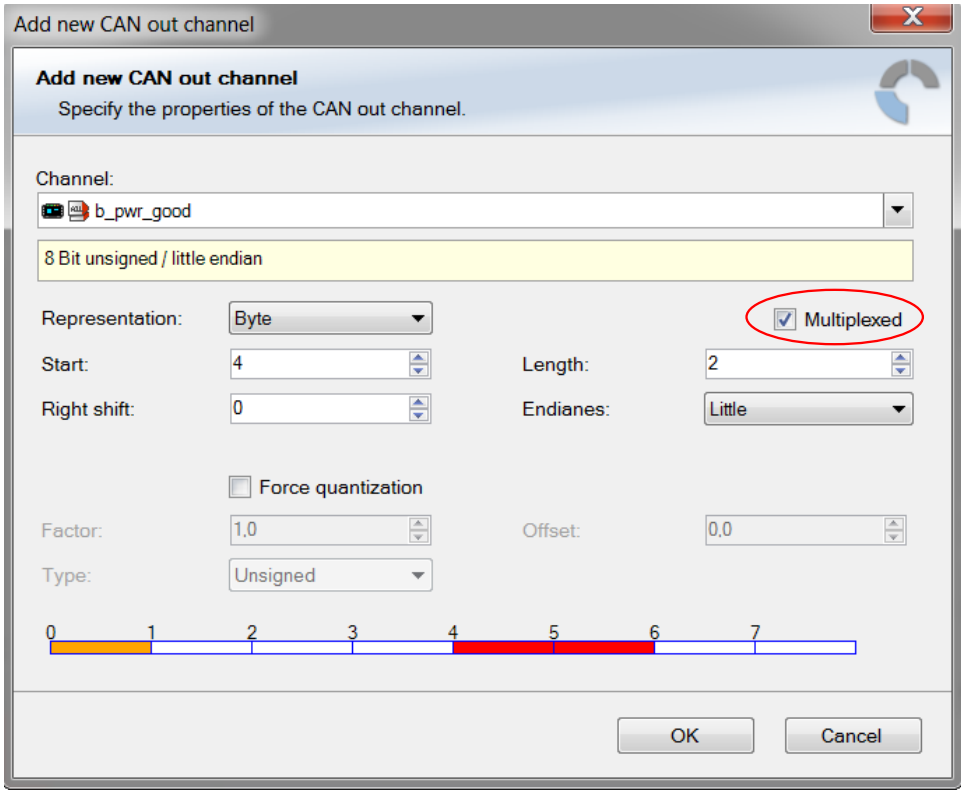
1. Double-click on any CAN bus item to open the “CAN messages overview”.
2. Select ‘Add CAN-IN’ and choose the desired CAN bus for the new input channel.
3. Check the box ‘Use Multiplexer’ and configure the multiplexer for the new CAN-IN channel.



4. To configure the multiplexer for a CAN-OUT channel, select ‘Add CAN-OUT’.
5. Check the box ‘Use Multiplexer’ and click on the button ‘Add row…’ to split the message identifiers into several rows.
6. Click on one row and select ‘Add channel’ to assign a channel to the row.

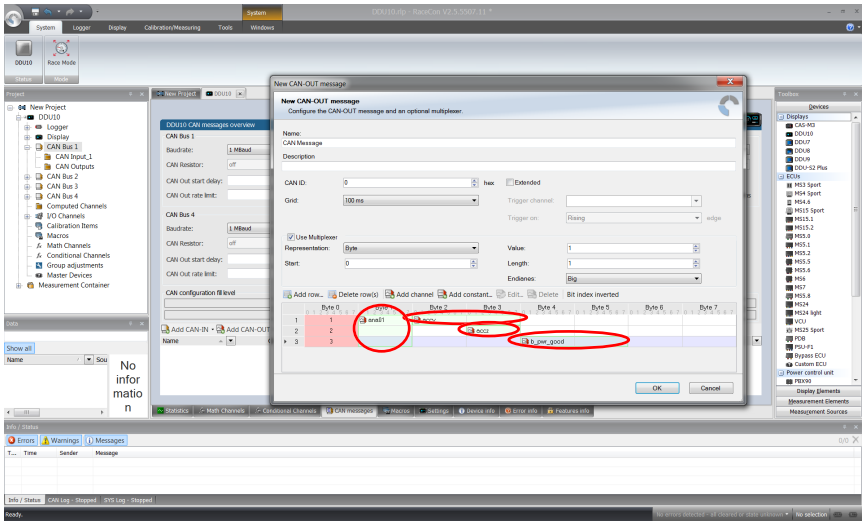


7. The ‘Add new CAN out channel’ dialog opens.
8. Select a channel and configure it. To assign it to the row selected before, check the box ‘Multiplexed’.
9. To move the channel message, change the “Start” value or click and hold the green field in the “Add new CAN out message” window.
10. Click ‘OK’ when done.



11. The channel message is assigned to the selected fields.

12. Click ‘OK’ when done.



10 Export and Import in RaceCon

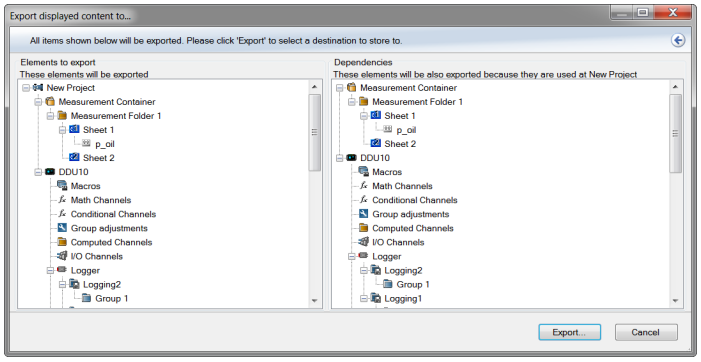
You can perform an export or an import on almost any level in the project tree.

10.1 Export in RaceCon

You can choose to export the whole project or you can export specific parts of the project.

Proceed with the following steps to perform an export:

1. Click with the right mouse button on an item in the project tree.
2. Select ‘Export...’ from menu. An ‘Export Selection’ window opens.



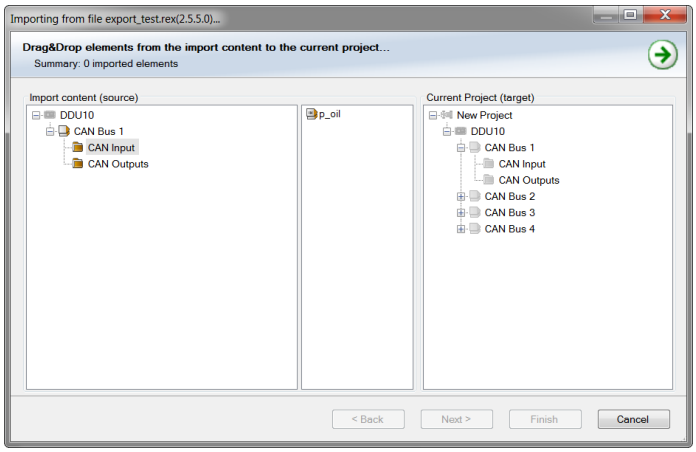
3. Click on ‘Export’ to select a destination to store.
4. Specify the filename.
5. Click ‘Save’ when done.

10.2 Import in RaceCon

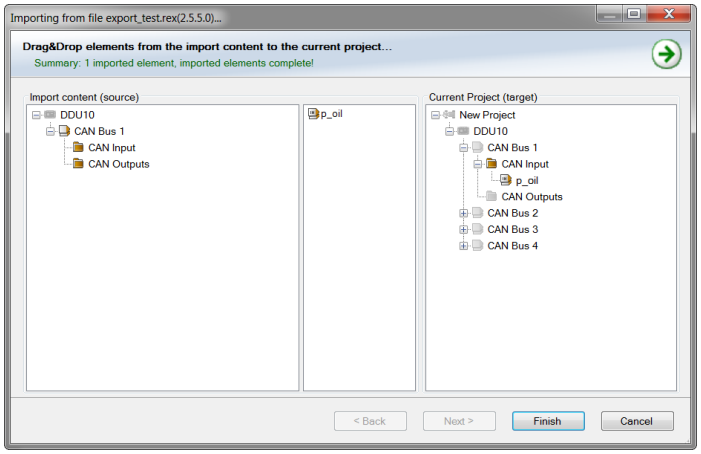
You can choose to import into the whole project or you can import into specific parts of the project.

Proceed with the following steps to perform an import:

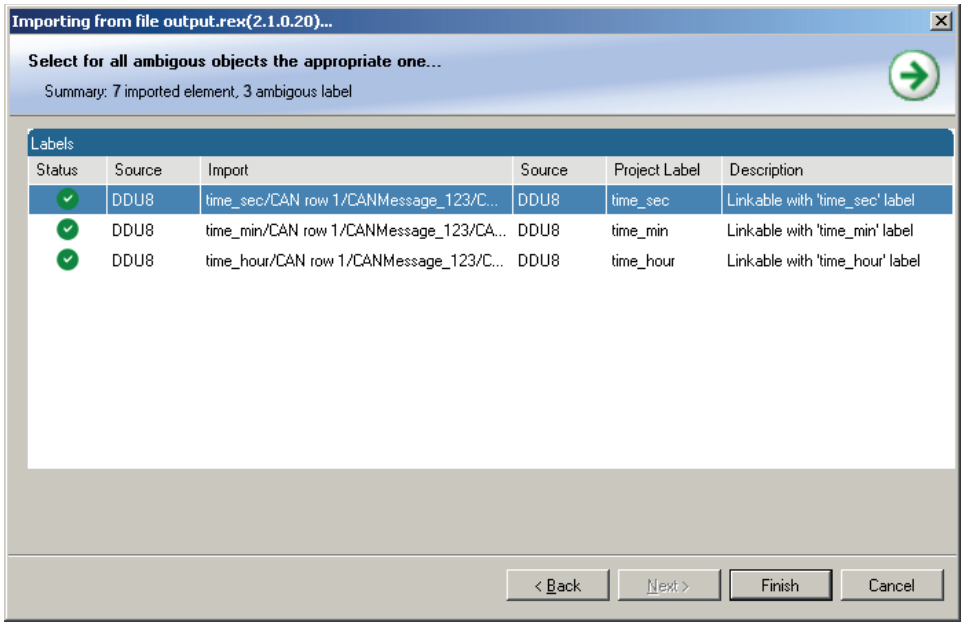
1. Click with the right mouse button on any item in the project tree.
2. Select ‘Import...’ from menu. A file browser opens.
3. Select the input file and click ‘Open’. An ‘Import Selection’ window opens.



4. Select channels to import.
5. Drag and drop the channel to ‘CAN Input’ of desired CAN bus on right hand side.



6. Click ‘Finish’. If a measurement channel belongs to more than one source (e.g. DDU 11 and MS 6), the ‘Solve Label Ambiguity’ window opens.



7. Assign the ambiguous channels to the desired source.
8. Click ‘Finish’.

11 Analog and Frequency Inputs

Analog inputs

- 0 to 5 V
- 12 bit A/D converter
- Switchable 3.01 kOhm pull-up resistor
- 10 kHz acquisition rate, up to 1 kHz recording rate
- Linear phase digital filter

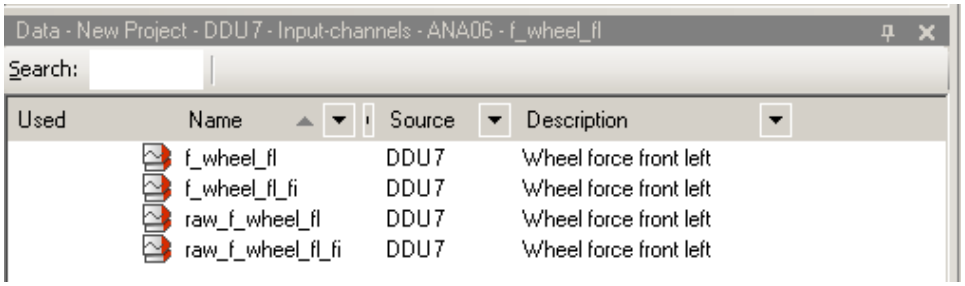
Frequency inputs

- 5 V Hall-effect type, 2.5 V trigger level (DF11 input with current interface or 5 V Hall-effect input with 2.5 V trigger level)
- 20 kHz max. frequency
- 10 ms measurement window

11.1 Analog inputs

11.1.1 Measurements channels

For each analog channel, several ‘subchannels’ are available.



Measurement labels with the characters ‘raw’ show the exact values in mV.

Measurement labels with the characters ‘_fi’ show filtered values.

The word ‘name’ in the table is a placeholder for the channel’s name.

Measurement label	Function
raw_name	mV value of sensor
raw_name_fi	Filtered mV value of sensor
name	Physical value of sensor
name_fi	Filtered physical value

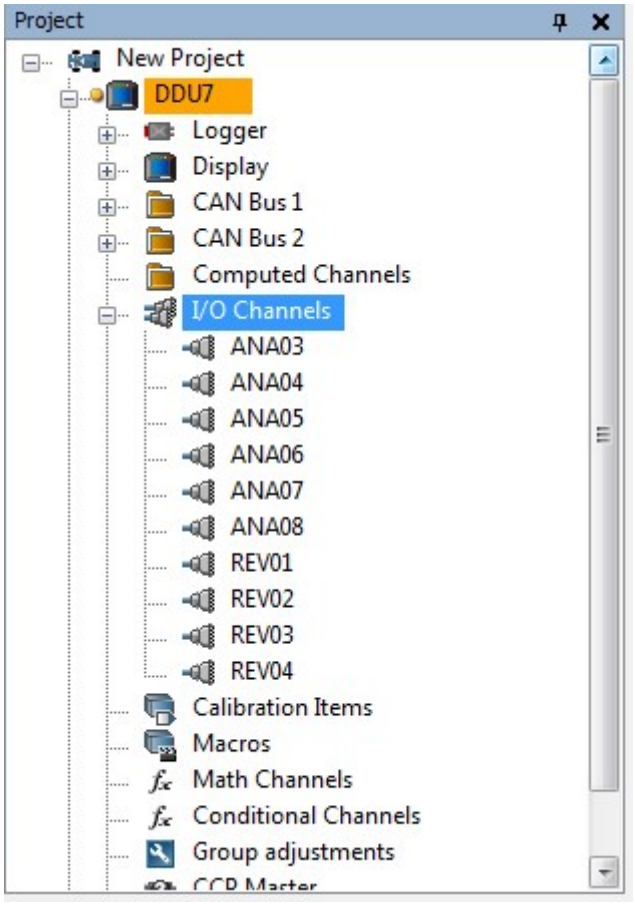
Filtered channels are routed through digital low pass filters:

- DDU 11 uses A/D converter oversampling and digital filtering to recording rate
- Digital filters eliminate ‘out-of-band’ noise
- Cut-off frequency automatically adjusted to recording rate
- Linear phase – no signal distortion
- Latency compensation – no filter delay in recorded data

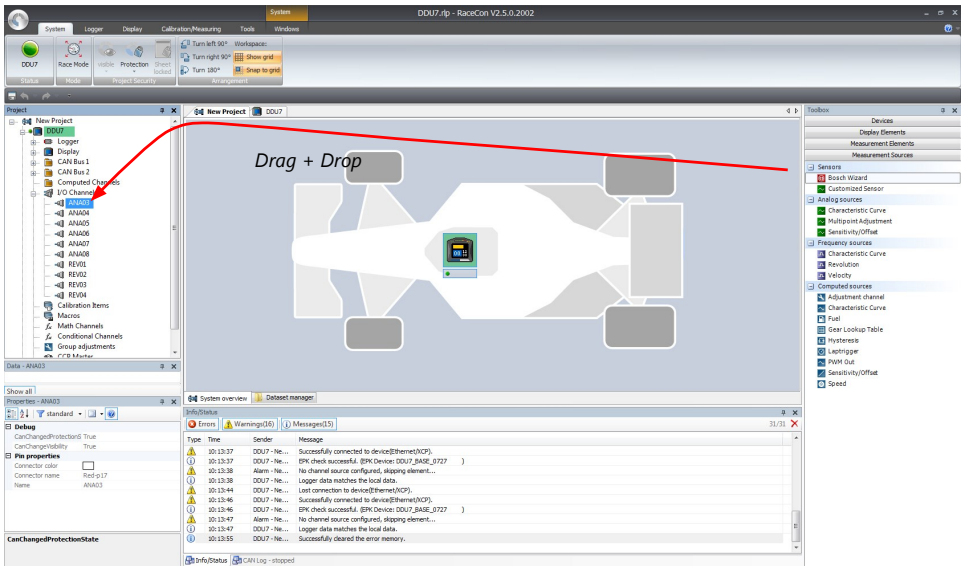
11.2 Configuring inputs

11.2.1 Configuring a predefined Bosch sensor with the 'Bosch Sensor Wizard'

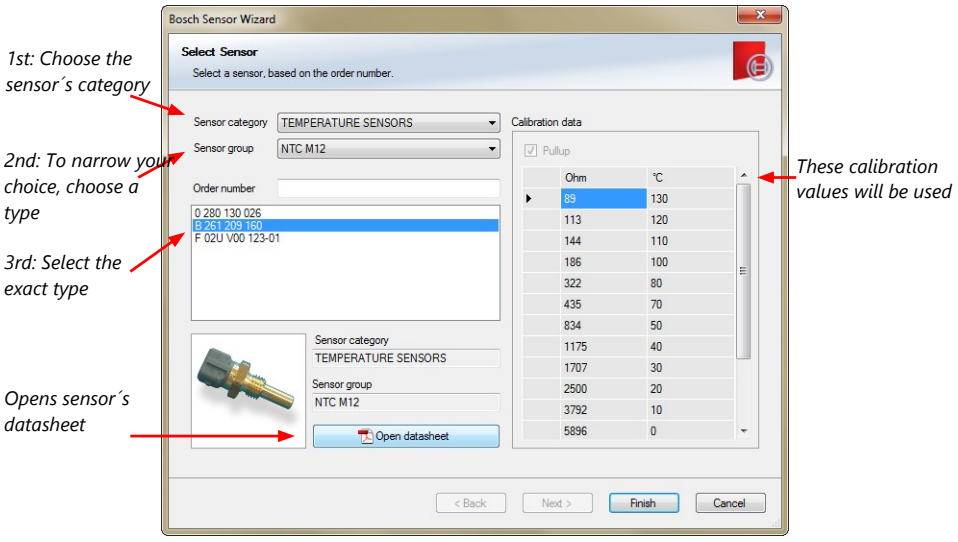
1. Click on ‘Measurement Sources’ in the Toolbox.
2. To expand the list of ‘I/O Channels’, click on ‘+’ in the DDU 11 Project Tree.



3. Drag the “Bosch Sensor Wizard” from the Toolbox and drop it on the desired analog input channel in the DDU 11 Project Tree.

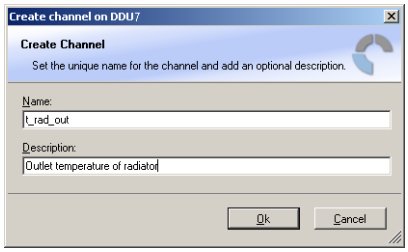


4. The “Bosch Sensor Wizard” opens.

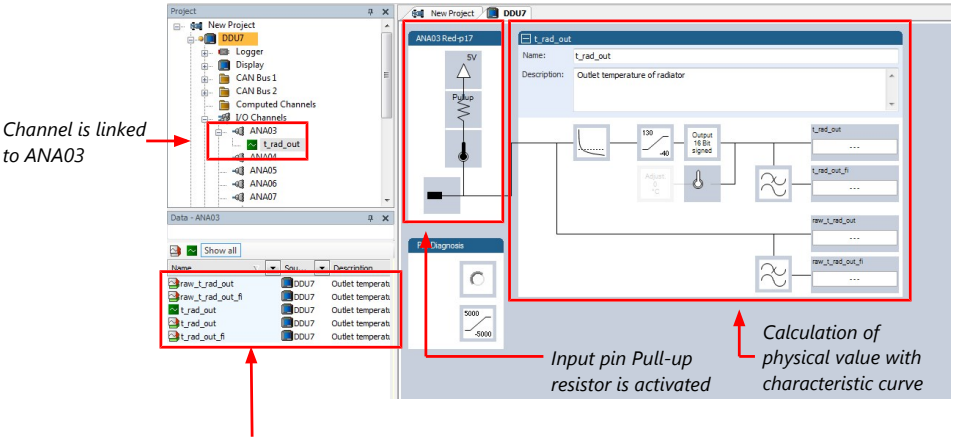


5. Click ‘Finish’ when done.
   The “Create channel” window opens.

6. Enter the channel name and description.



7. Click ‘Ok’ when done.
   The channel is inserted into the DDU 11 Project Tree.



Available measurements for channel:

Measurement label	Function
raw_name	mV value of sensor
raw_name_**fi**	Filtered mV value of sensor
name	Physical value of sensor
name_**fi**	Filtered physical value

11.2.2 Configuring a generic linear sensor

Example: Acceleration sensor 5 g

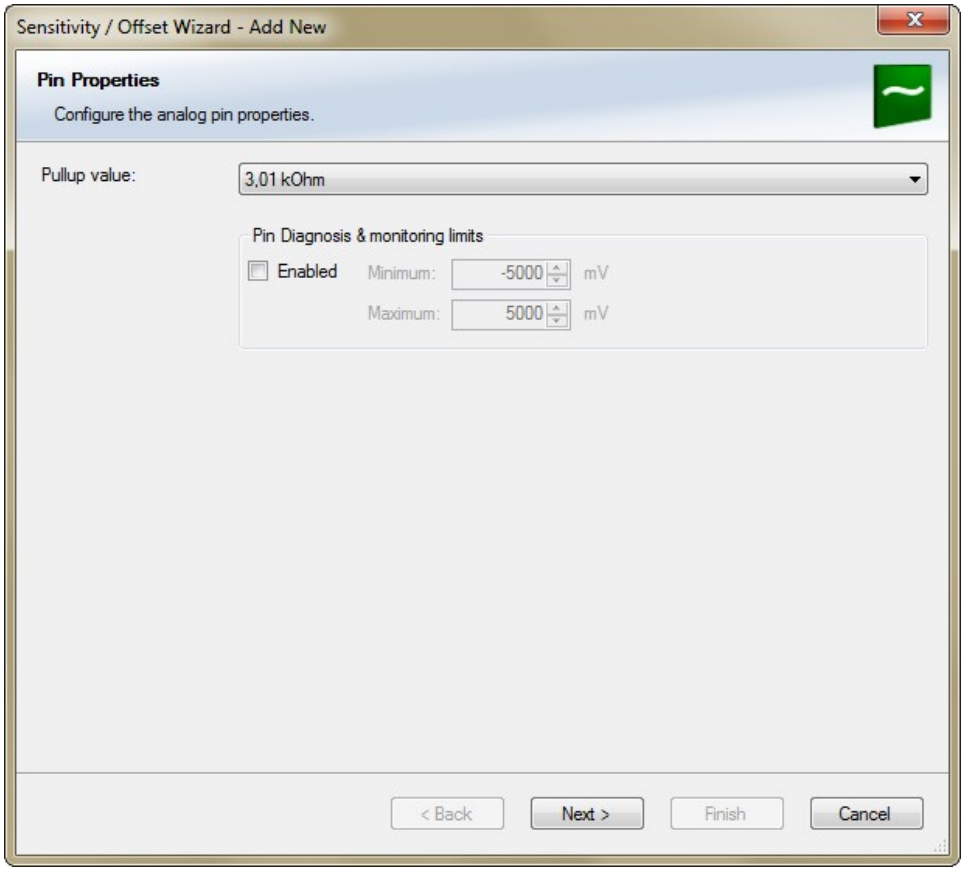
– From sensor data sheet - operating characteristics:

Output Signal					
Zero g (T_A = 25°C, V_{DD} = 5.0 V)(4)	V_{OFF}	2.25	2.5	2.75	V
Zero g (V_{DD} = 5.0 V)	V_{OFF}	2.0	2.5	3.0	V
Sensitivity (T_A = 25°C, V_{DD} = 5.0 V)(5)	S	380	400	420	mV/g
Sensitivity (V_{DD} = 5.0 V)	S	370	400	430.1	mV/g
Bandwidth Response	f_{-3dB}	42.5	50	57.5	Hz
Nonlinearity	NL_{OUT}	–1.0	—	+1.0	% FSO

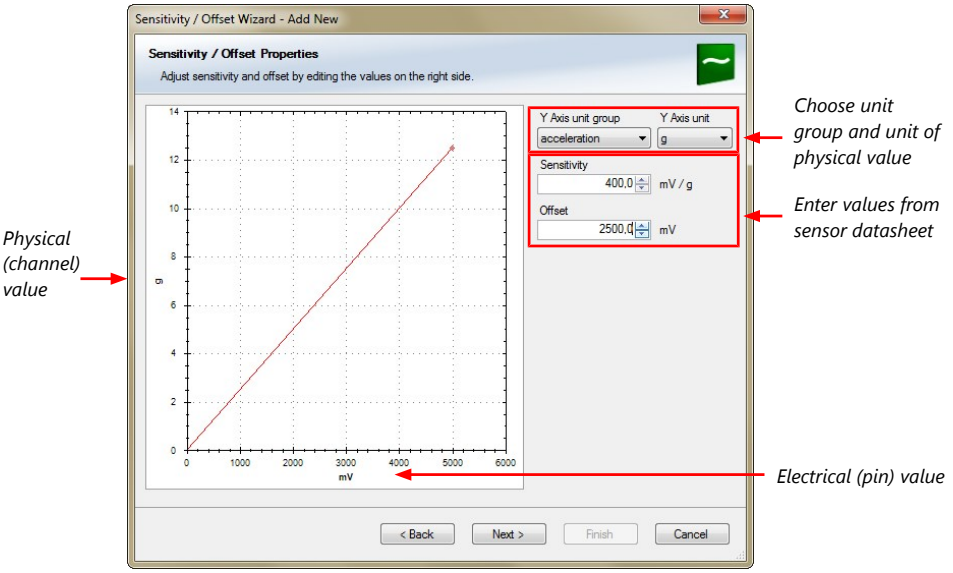
– Sensitivity 400 mV/g, Offset 2,500 mV

– The sensor has a linear output signal with sensitivity and offset

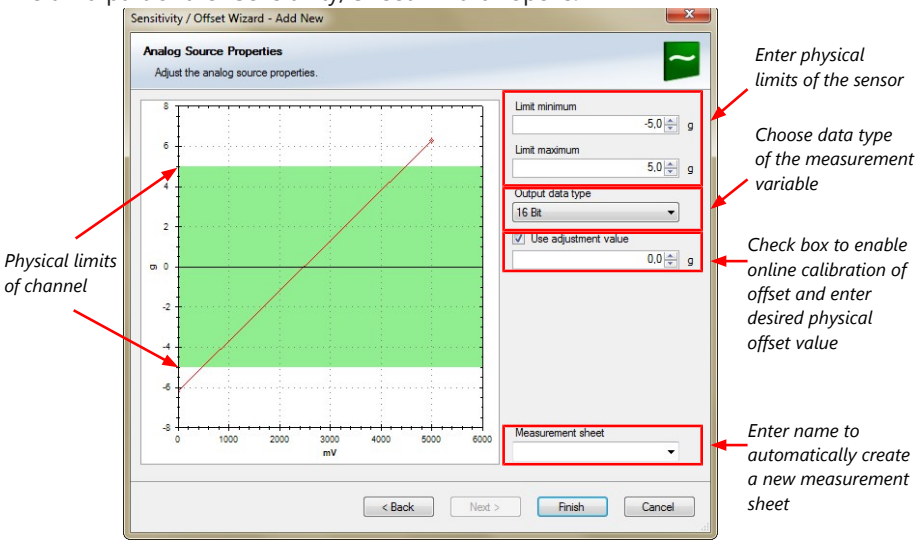
1. Click on ‘Measurement Sources’ in the Toolbox.
2. To expand the list of ‘I/O Channels’, click on ‘+’ in the DDU 11 Project Tree.
3. Drag the “Sensitivity/Offset” analog signal source from the Toolbox and drop it on the desired analog input channel in the DDU 11 Project Tree. A “Sensitivity/Offset Wizard” opens.
4. To activate the internal pullup-resistor, check the box. The internal pullup-resistor is used to get a 5 V signal at the analog channel of the DDU 11. It allows you to use a push-button. The fixed value of the internal pullup-resistor is 3,010 Ohm. If using an additional external pullup-resistor, set up the overall resistance.



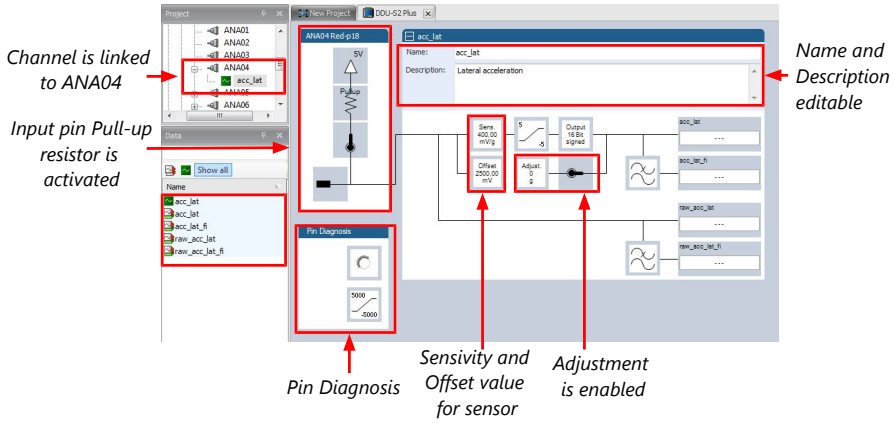
5. Click ‘Next’ when done.
   The second part of the “Sensitivity/Offset Wizard” opens.



6. Click ‘Next’ when done.
   The third part of the “Sensitivity/Offset Wizard” opens.



7. Click ‘Finish’ when done.
8. Enter a channel name and a description.
9. Click ‘OK’ when done.
   The channel is inserted into the DDU 11 Project Tree.



Available measurements for channel:

Measurement label	Function
raw_name	mV value of sensor
raw_name_**fi**	filtered mV value of sensor
name	physical value of sensor
name_**fi**	filtered physical value

Note

Working with automatically created measurement sheets is explained in chapter 'Setting up an online measurement [▶ 113]'.

11.2.3 Configuring a generic nonlinear sensor

Example: Thermistor 5 kOhm

– From sensor data sheet - resistance values over temperature:

PART NR.: 2381 640 502
HTCLE100E3502

T_{oper} [°C]	R_T [Ω]
-40	166 047
-35	119 950
-30	87 600
-25	64 643
-20	48 179
-15	36 250
-10	27 523
-5	21 078

T_{oper} [°C]	R_T [Ω]
0	16 277
5	12 669
10	9 936
15	7 849
20	8 244
25	5 000
30	4 030
35	3 267

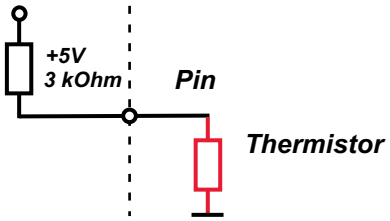
T_{oper} [°C]	R_T [Ω]
40	2 685
45	2 166
50	1 903
55	1 494
60	1 245
65	1 024
70	876
75	740

T_{oper} [°C]	R_T [Ω]
80	628
85	535
90	457
95	399
100	338
105	292
110	251
115	221

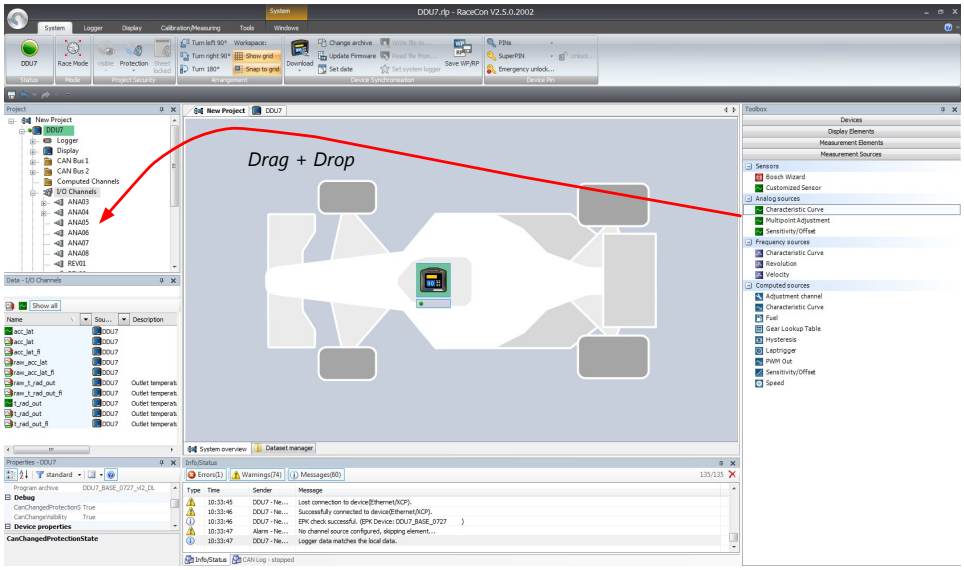
– The sensor has a nonlinear behavior

– Use characteristic curve for linearization

– Input voltage is the ratio between pull-up resistor and thermistor

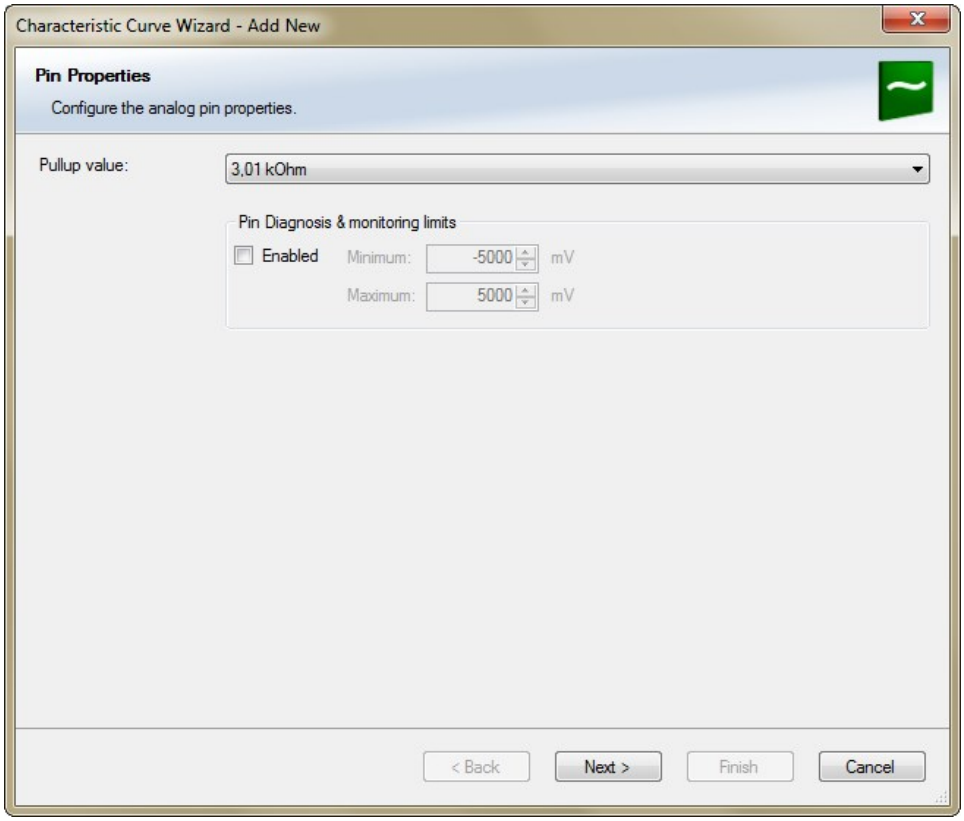


1. Click 'Measurement Sources' in the Toolbox.
2. To expand the list of 'I/O Channels', click on '+' in the DDU 11 Project Tree.
3. Drag the "Characteristic Curve" analogue signal source from the Toolbox and drop it on the desired analogue input channel in the DDU 11 Project Tree.

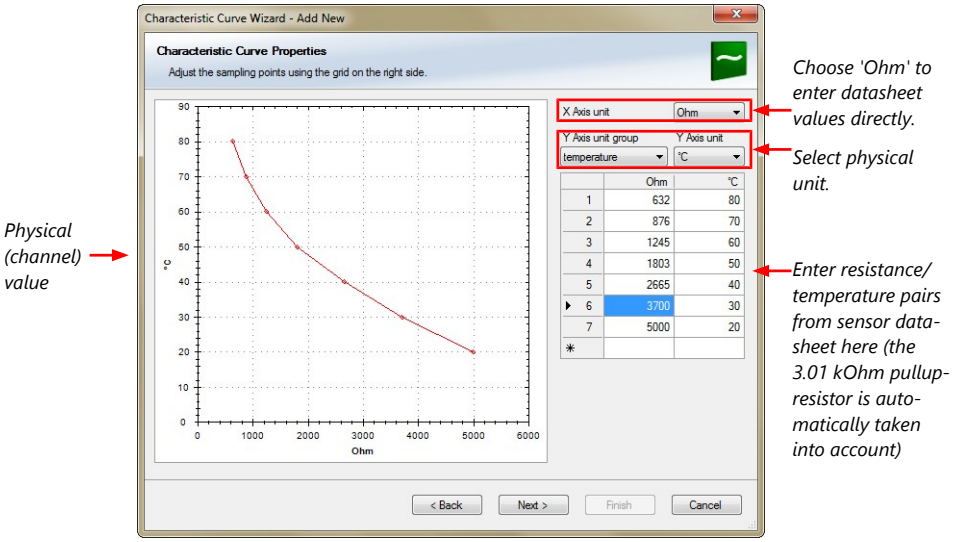


4. A “Characteristic Curve Wizard” opens.

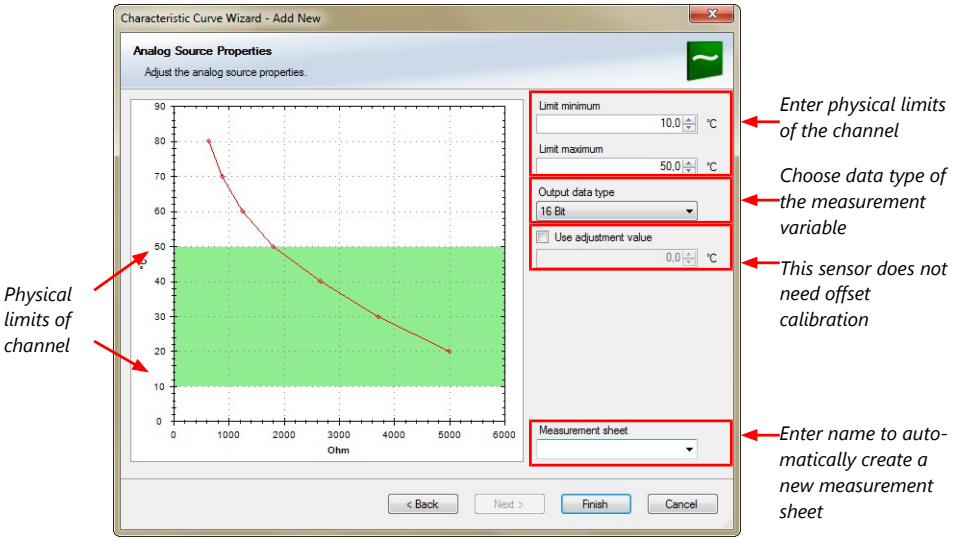
5. To activate the internal pull up-resistor, check the box. The DDU 11 pull up-resistor is used to get a 5 V signal at the analogue channel of the DDU 11. It allows you, to use a push-button. The fixed value of the internal pull up-resistor is 3,010 Ohm. If using an additional external pull up-resistor, set up the overall resistance.



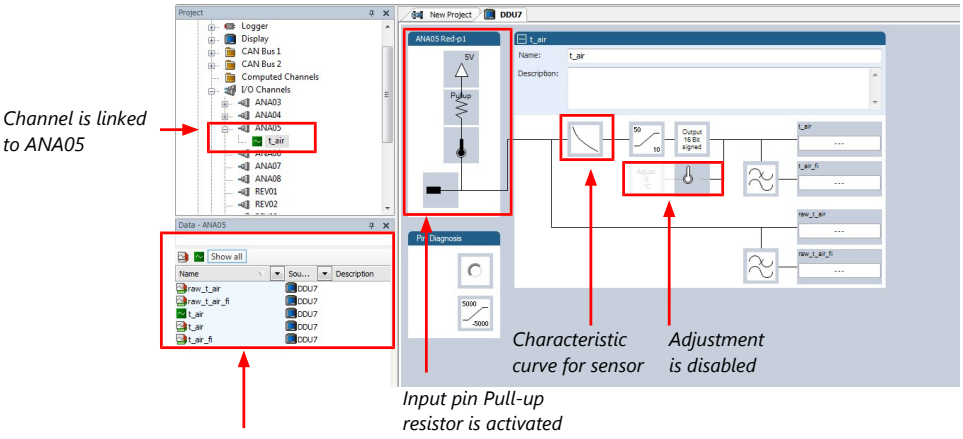
6. Click ‘Next’ when done.
   The second part of the “Sensitivity/Offset Wizard” opens.



7. Click ‘Next’ when done.
   The third part of the “Characteristic Curve Wizard” opens.



8. Click ‘Finish’ when done.
9. Enter channel name and description.
10. Click ‘OK’ when done.
    The channel is inserted into the DDU 11 Project Tree.



Available measurements for channel:

Measurement label	Function
raw_name	mV value of sensor
raw_name_**fi**	filtered mV value of sensor
name	physical value of sensor
name_**fi**	filtered physical value

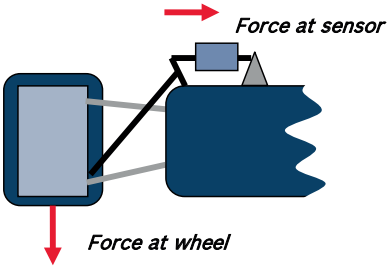
Note

Working with automatically created measurement sheets is explained in chapter 'Setting up an online measurement [▶ 113]'.

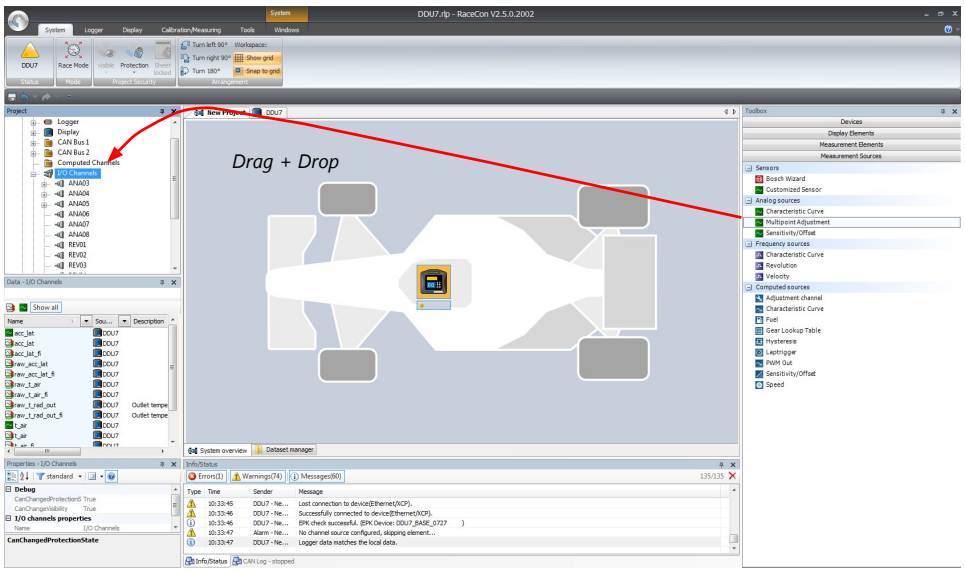
11.2.4 Configuring a multipoint adjustment

Example: Measurement of wheel force

- Physical property 'wheel force' not directly measurable
- Load transfer through suspension kinematics
- Physical value at sensor position defined by vehicle
- Curve definition by online adjustment at vehicle

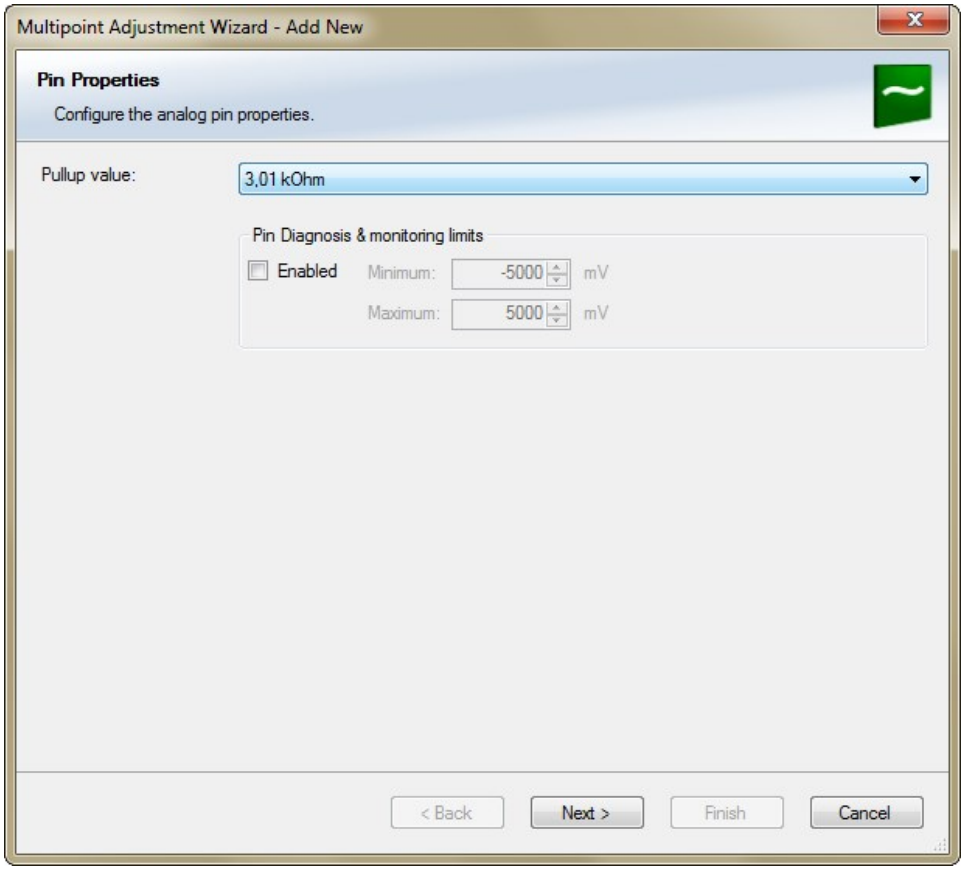


1. Click on 'Measurement Sources' in the Toolbox.
2. Expand the list of 'I/O Channels' by clicking on '+' in the DDU 11 Project Tree.
3. Drag the 'Multipoint Adjustment' analog signal source from the Toolbox and drop it on the desired analog input channel in DDU 11 Project Tree.

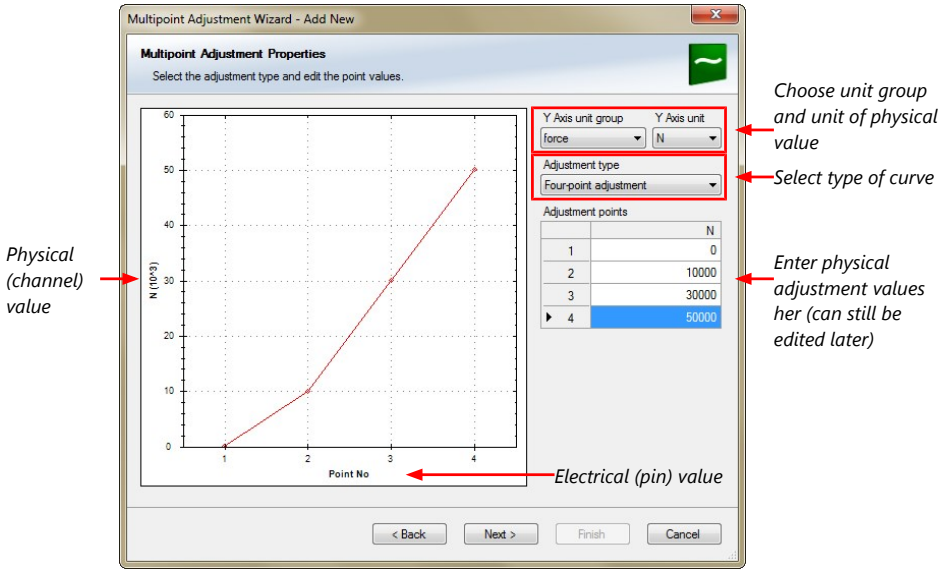


4. A 'Multipoint Adjustment Wizard' opens.

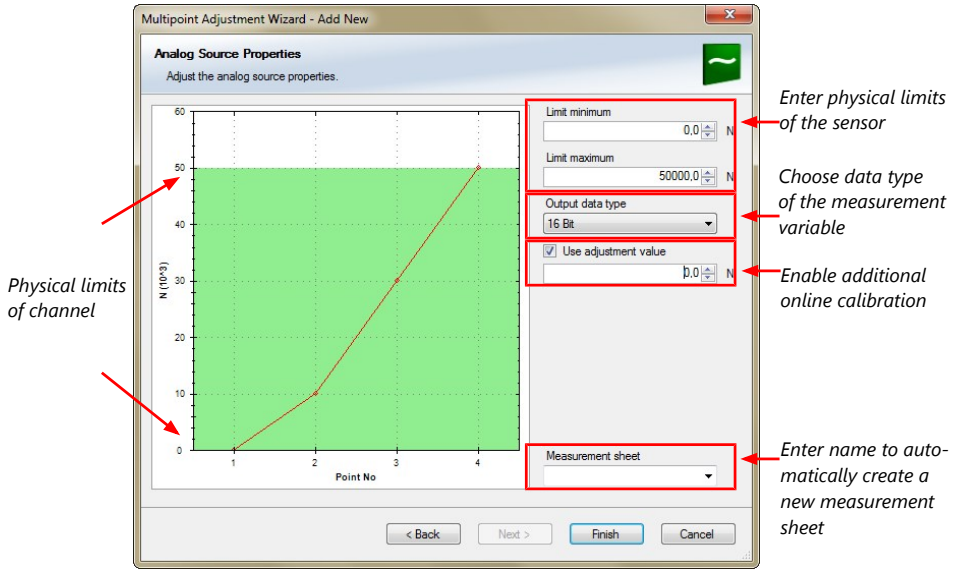
5. To activate the internal pullup-resistor, check the box. The internal pullup-resistor is used to get a 5 V signal at the analog channel of the DDU 11. It allows you to use a push-button. The fixed value of the internal pullup-resistor is 3.01 kOhm. If using an additional external pullup-resistor, set up the overall resistance.



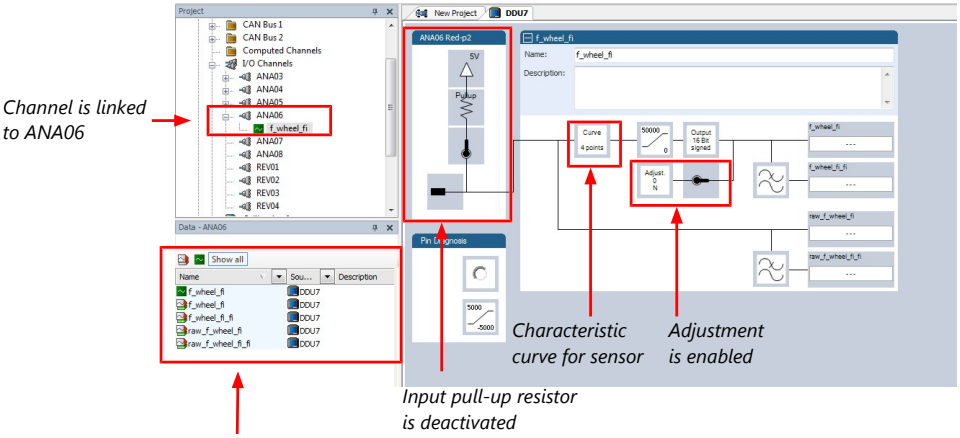
6. Click ‘Next‘ when done.
   The second part of the ‘Multipoint Adjustment Wizard‘ opens.



7. Click ‘Next‘ when done.
   The third part of the ‘Multipoint Adjustment Wizard‘ opens.



8. Click ‘Finish’ when done.
9. Enter channel name and description.
10. Click ‘OK’ when done.
    The channel is inserted into the DDU 11 Project Tree.



Available measurements for channel:

Measurement label	Function
raw_name	mV value of sensor
raw_name_**fi**	filtered mV value of sensor
name	physical value of sensor
name_**fi**	filtered physical value

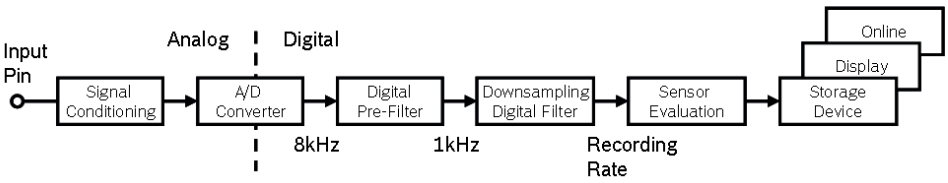
Online definition of the curve is covered in chapter ‘Online calibration of measurement channels [▶ 118]’.

Note

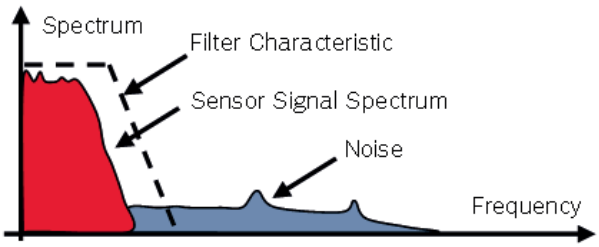
Working with automatically created measurement sheets is explained in chapter ‘Setting up an online measurement [▶ 113]’.

11.2.5 Digital filter details

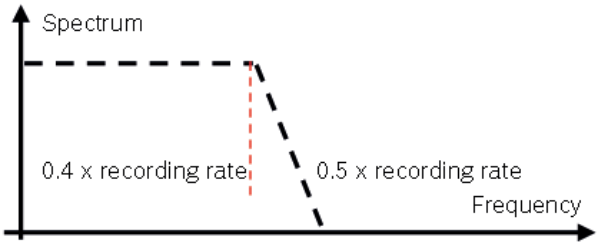
DDU 11 uses A/D converter oversampling and digital filtering to recording rate.



Digital filters eliminate 'out-of-band' noise



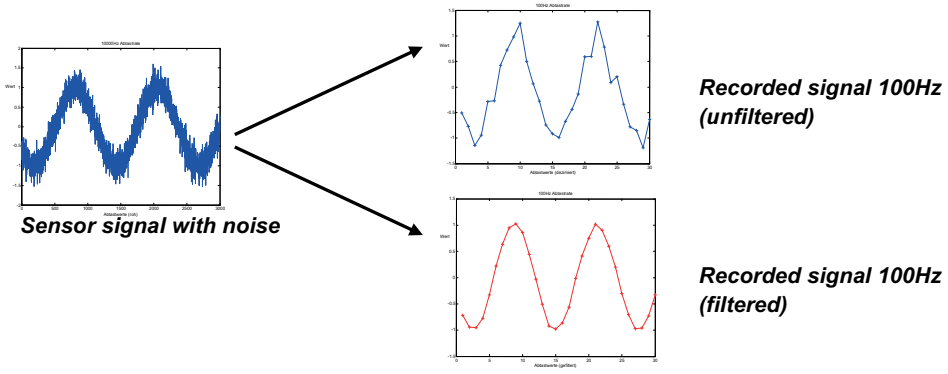
Cut-off frequency automatically adjusted to recording rate



Example:

- 100 Hz recording rate (10 ms)
- <40 Hz pass band (>99 %)
- >50 Hz stop band (<1 %)

Linear phase – no signal distortion



Latency compensation – no filter delay in recorded data

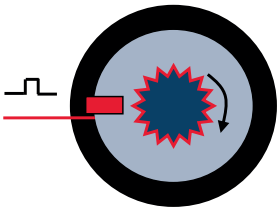
- Filtering is (smart) averaging over several samples
- Filtered signal is delayed with respect to real time signal

- DDU 11 filters have constant, frequency independent delay
- Delay (e.g. 22 samples at 10 ms) is corrected during recording
- No delay filtered vs. unfiltered in recorded data
- Correction is (of course) not possible for real time data (display, online, PWM out)
- Use filtered data for recording, use unfiltered data for real-time

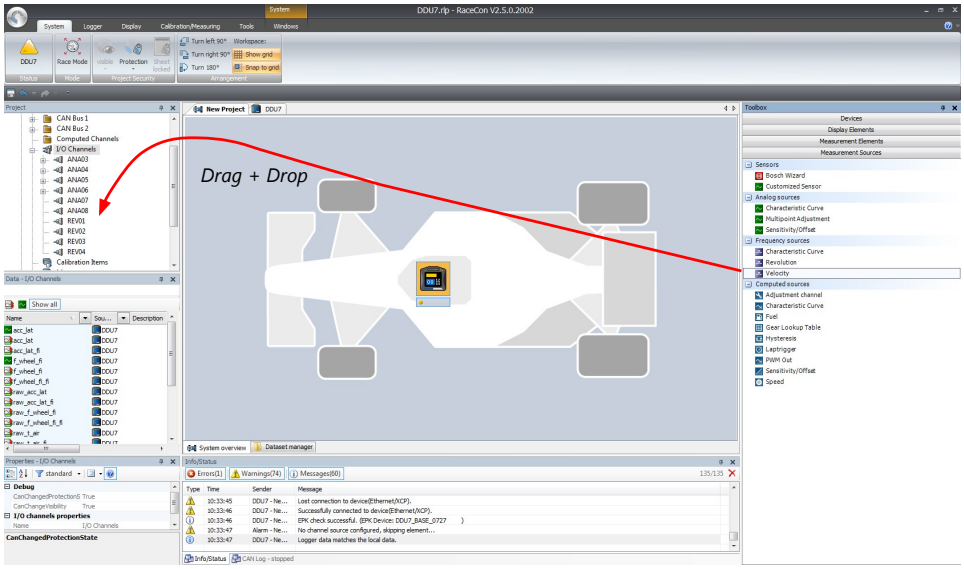
11.2.6 Configuring a frequency input

Example: measurement of wheel speed

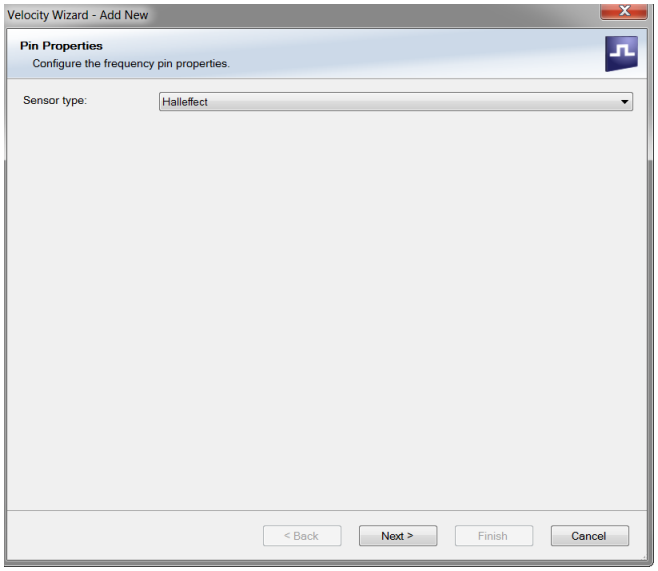
- Pulse wheel attached to wheel
- Each passing tooth of pulse wheel triggers hall sensor
- Calculation of wheel speed with wheel circumference



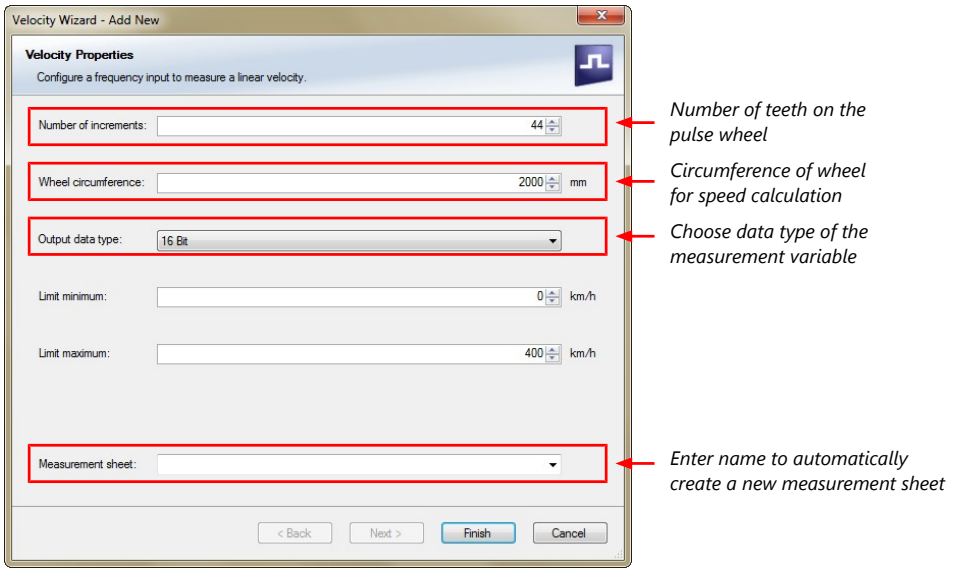
1. Click on 'Measurement Sources' in the Toolbox.
2. To expand the list of 'I/O Channels', click on the '+' in the DDU 11 Project Tree.
3. Drag the 'Velocity' digital signal source from the Toolbox and drop it on the desired 'REV' input channel in the DDU 11 Project Tree.



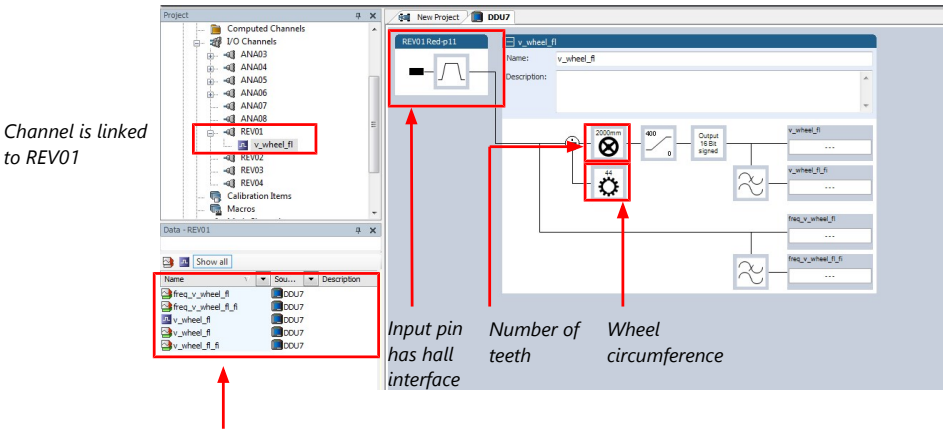
4. The 'Velocity Wizard' opens.
5. Select the sensor type. The DDU 11 works with Halleffect and DF11 sensors.



6. Click ‘Next’.
7. Define the settings for the sensor.



8. Click ‘Finish’ when done.
9. Enter the channel name and description.
10. Click ‘OK’ when done.
    The channel is inserted into the DDU 11 Project Tree.



Available measurements for channel:

Measurement label	Function
raw_name	mV value of sensor
raw_name_**fi**	filtered mV value of sensor
name	physical value of sensor
name_**fi**	filtered physical value

Note

Measurement of ‘Revolution’ is similar.

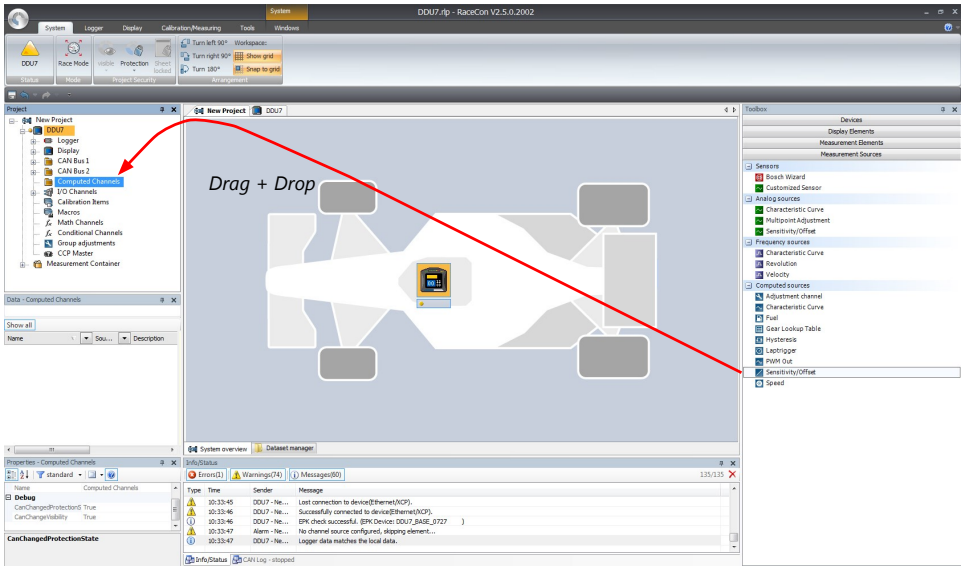
11.3 Configuring computed sources

Computed sources receive data from a measurement channel rather than an input pin.

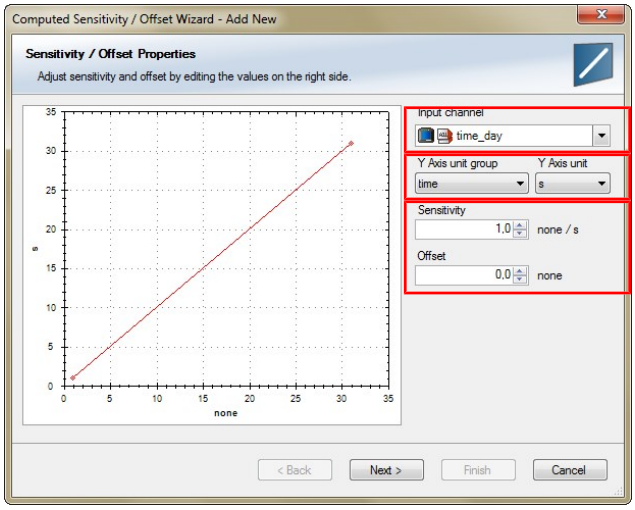
- Sensitivity/Offset calculation on input channel
- Characteristic curve calculation on input channel
- Computed vehicle speed
- Lap trigger (covered in a special separate section)

Example: Sensitivity/offset calculation on input channel

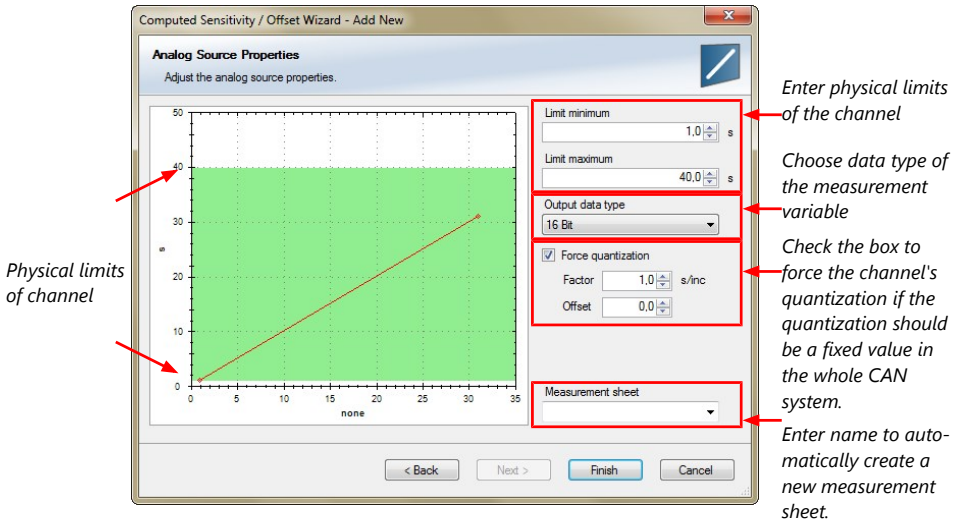
1. Click ‘Measurement Sources’ in the Toolbox.
2. Drag the ‘Sensitivity/Offset’ computed source from the Toolbox and drop it on ‘Computed Channels’ in the DDU 11 Project Tree.



3. A ‘Computed Sensitivity/Offset Wizard’ opens.



4. Click ‘Next’ when done.
   The second part of the ‘Computed Sensitivity/Offset Wizard’ opens.



5. Click ‘Finish’ when done.
6. Enter channel name and description.
7. Click ‘OK’ when done.

   The channel is inserted into the DDU 11 Project Tree.

Note

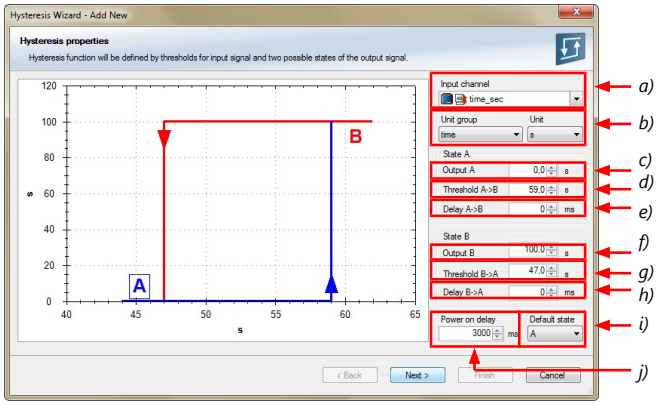
Working with automatically created measurement sheets is explained in chapter ‘Setting up an online measurement [▶ 113]’.

11.4 Hysteresis

The hysteresis function avoids the high-frequent switchover of the measurement channel value. The hysteresis can be adjusted for each input measurement channel individually and can be used for further processing.

1. Click ‘Measurement Sources’ in the Toolbox.
2. Drag the ‘Hysteresis’ computed source from the Toolbox and drop it on ‘Computed Channels’ in the DDU 11 Project Tree.

   A ‘Hysteresis Wizard’ opens.



a) Choose input measurement channel.

b) Choose unit group and unit of output.

c) Enter output value of state A in the unit selected in b).

d) Enter threshold value when state changes from A to B.

e) Enter delay time when state changes from A to B.

f) Enter output value of state B in the unit selected in b).

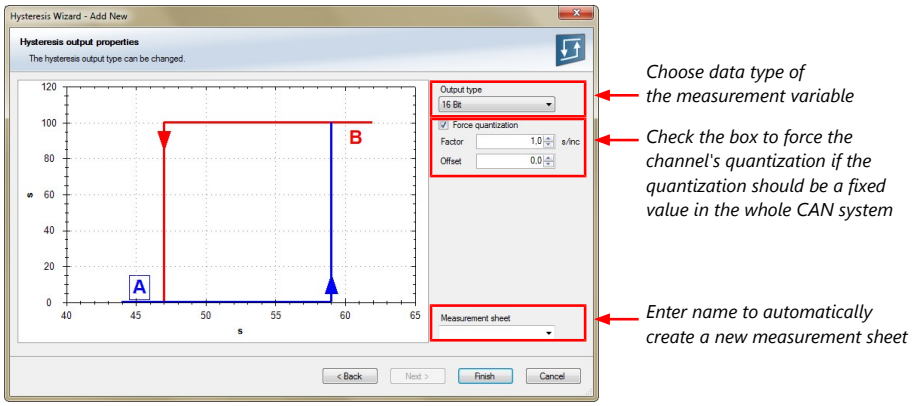
g) Enter threshold value when state changes from B to A.

h) Enter delay time when state changes from B to A.

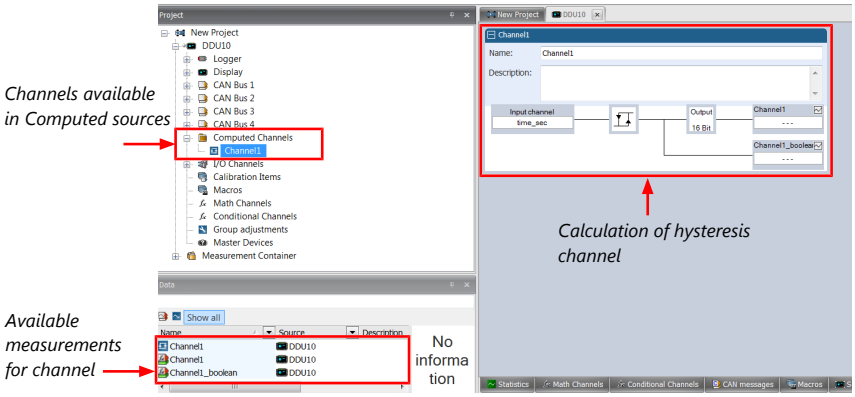
i) Enter time when the hysteresis function is activated after vehicle´s startup.

j) Enter the channel´s state (A or B) at startup.

3. Click ‘Next’ when done.
   The second part of the ‘Hysteresis Wizard’ opens.



4. Click ‘Finish’ when done.
5. Enter channel name and description.
6. Click ‘OK’ when done.
   The channel is inserted into the DDU 11 Project Tree.



11.4.1 Special functionality: Vehicle speed

This functionality allows:

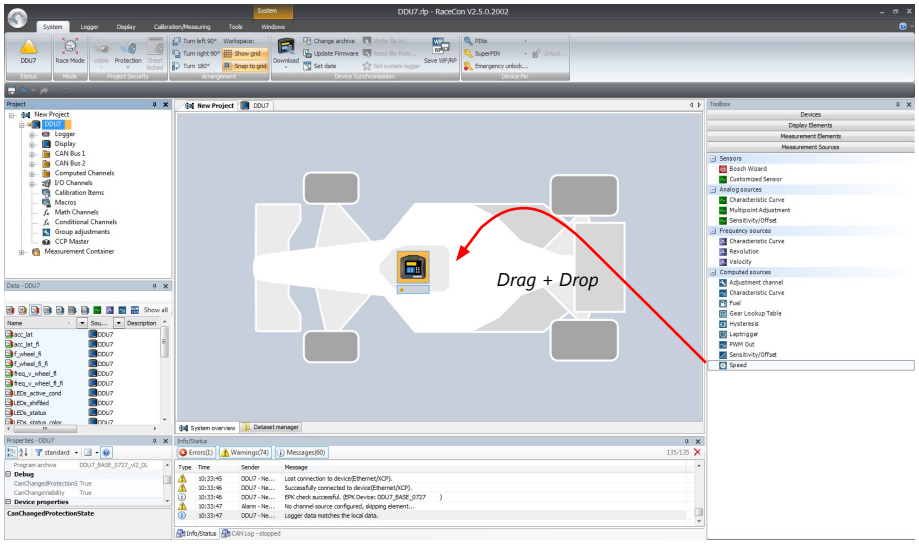
- high performance vehicle owners to measure wheel spin under acceleration and wheel slip/lock under braking.
- calculating vehicle 'speed over ground'.

Vehicle speed calculation function

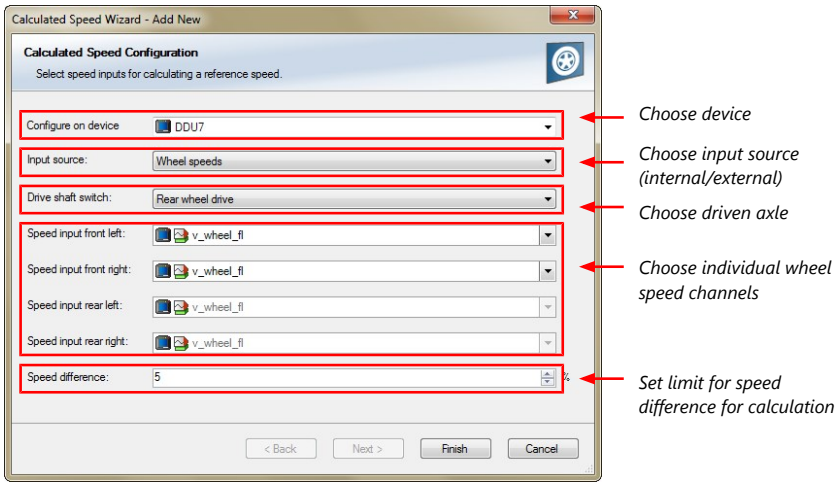
- Calculating vehicle speed of 2 wheel drive: (Wheel speeds of non-driven axle as input)
  - Calculated speed is average of both speeds if speed difference between wheels < limit.
  - Calculated speed is maximum of both speeds if speed difference between wheels > limit.
- Calculating vehicle speed of 4 wheel drive: (Wheel speeds of all wheels as input)
  - Calculated speed is speed of 2nd fastest wheel.

11.4.2 Setting up calculated speed

1. Click on tab 'System Overview'.
2. Click on 'Measurement Sources' in the Toolbox.
3. Drag the 'Speed' computed source from the Toolbox and drop it on the project name in the DDU 11 Project Tree. Do not drop it on 'DDU 11'!

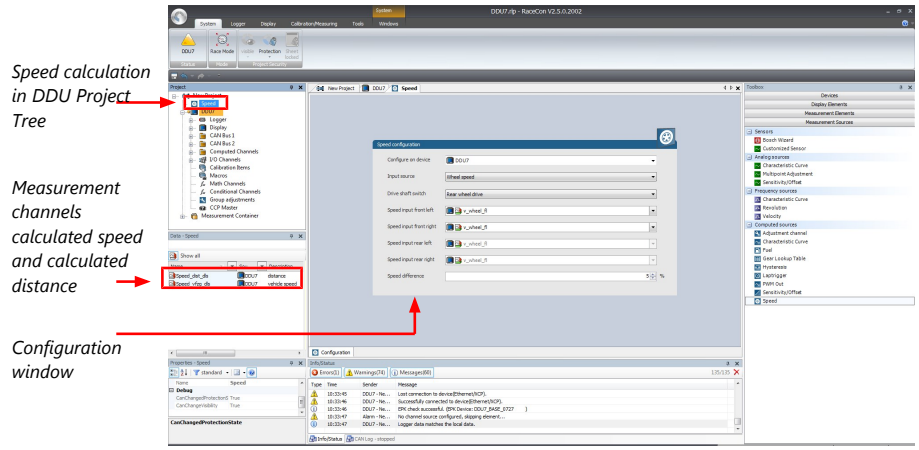


A ‘Calculated Speed Wizard’ opens.



4. Click ‘Finish’ when done.

The speed calculation is inserted into the DDU 11 Project Tree.



12 Online Measurement and Calibration

DDU 11 configuration

- System configuration (channel + display configuration, CAN I/O, etc.) is stored in the DDU 11
- Use RaceCon to create and download configuration from the PC to DDU 11 Communication interface: Ethernet
- Communication protocol: XCP

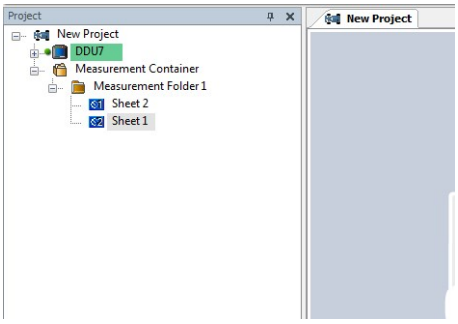
Online Measurement and Calibration

- System status and diagnosis
- Check and calibrate sensors in the vehicle
- Live display of sensor values on the PC
- Use RaceCon for diagnosis, online measurement and calibration
- Communication interface: Ethernet
- Communication protocol: XCP

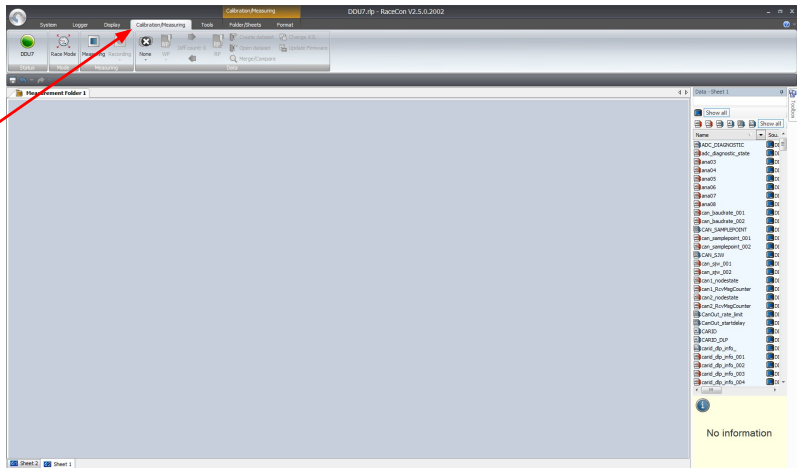
12.1 Setting up an online measurement

DDU 11 supports online measurement of sensor values and diagnostic variables.

1. Expand 'Measurement Container' and 'Measurement Folder 1' in the Project Tree and double-click on 'Sheet1'. Alternatively, click on the 'Calibration/Measuring' tab to open the window directly. 'Sheet 1' opens in a new 'Calibration/Measuring' window.

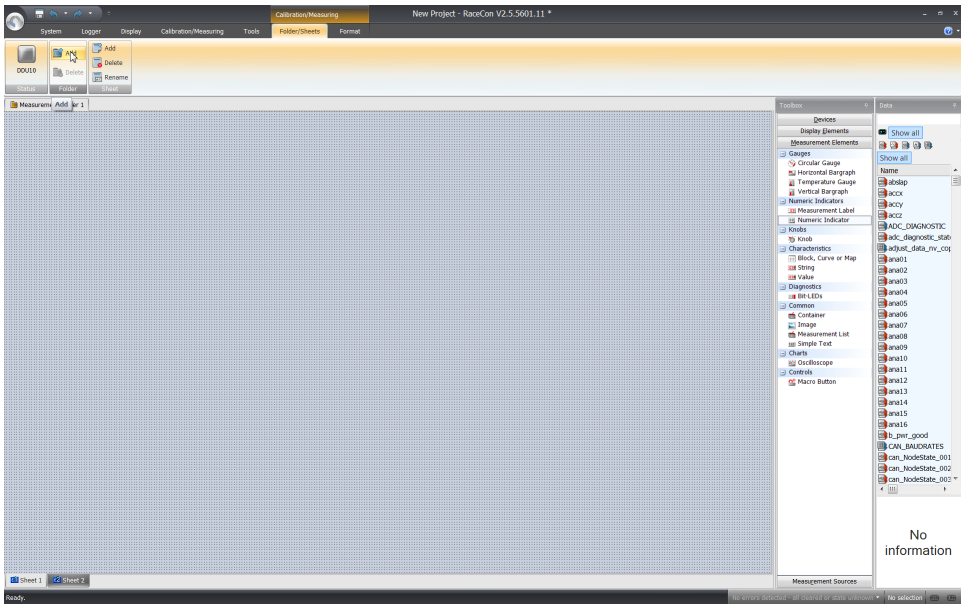


RaceCon changes automatically to "Calibration/ Measuring"



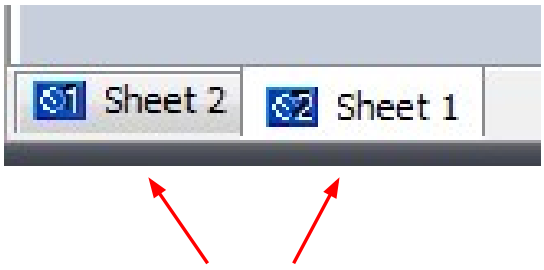
2. Click on the 'Folder/Sheets' tab, which appears when you are in the 'Calibration/ Measurement' window, to create a new measurement folder.

3. Click on the ‘Add’ button for folders in the upper left corner.



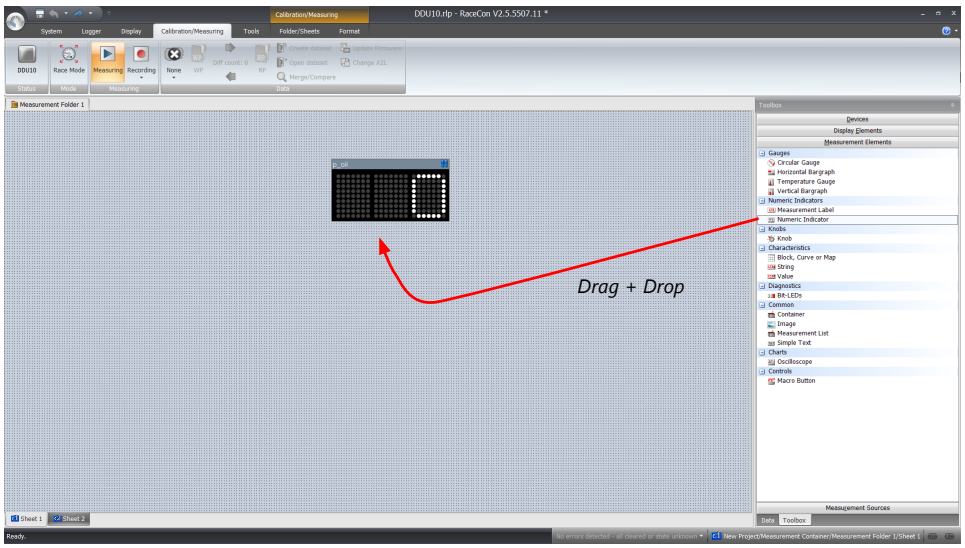
In the menu for sheets, you will find buttons to add, delete and rename new sheets

1. To change between different sheets, click on the tabs on the bottom of the ‘Calibration/Measuring’ window.



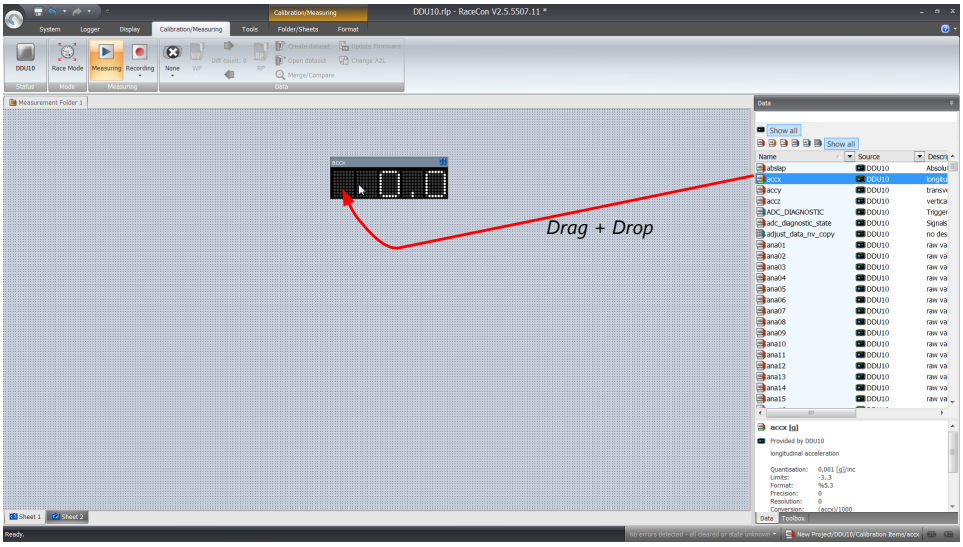
To add an element to a measurement sheet, perform the following steps:

1. Drag a measurement element from the Toolbox and drop it on the measurement sheet.



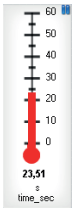
2. Select the desired measurement channel from the ‘Data’ area and drop it on the measurement element.

If the DDU 11 shows the green status, the value is displayed.

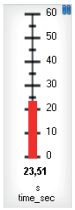


RaceCon offers different types of measurement elements:

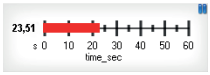
Circular gauge



Temperature gauge



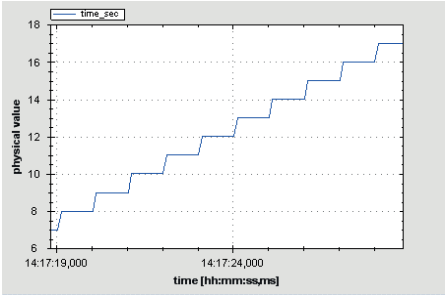
Vertical Bar graph style



Horizontal Bar graph style

Measurement label

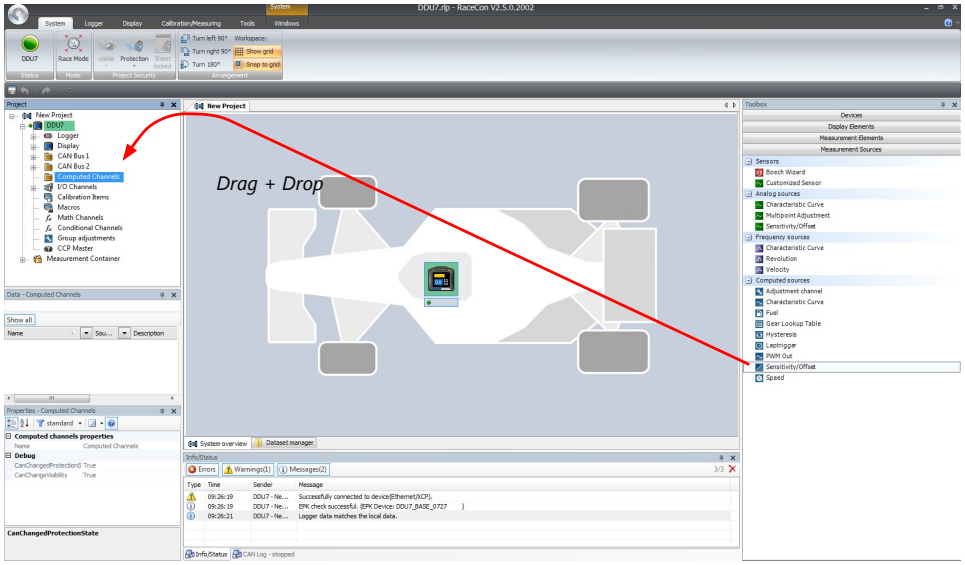
Numeric indicator



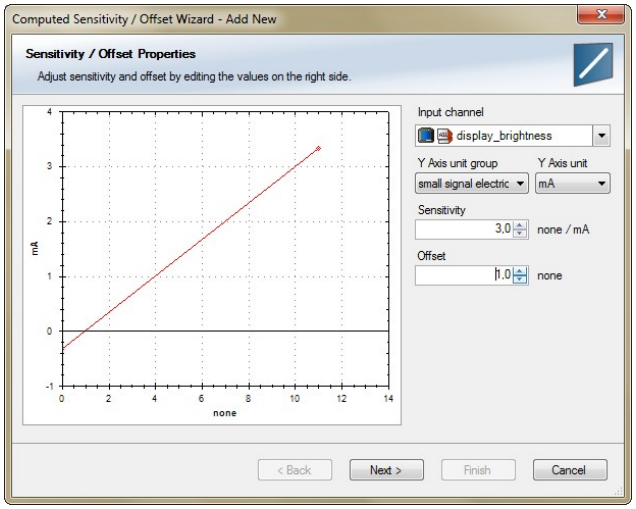
Oscilloscope (Chart)

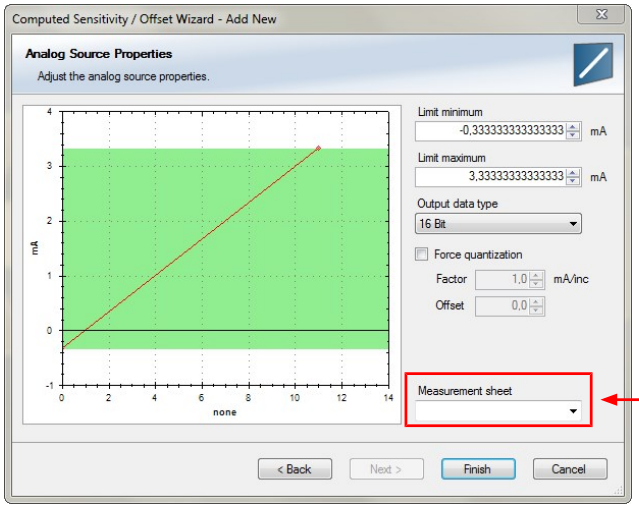
12.1.1 Automatic creation of measurement sheets

RaceCon can create measurement sheets automatically.
You can create and use measurement sheets with the DDU 11 as well as with all other devices connected to RaceCon.

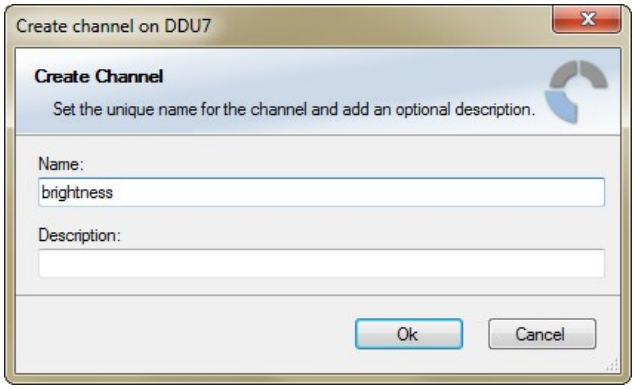


1. During the configuration of a measurement channel, select a measurement sheet from the list box or enter a name for a new measurement sheet.

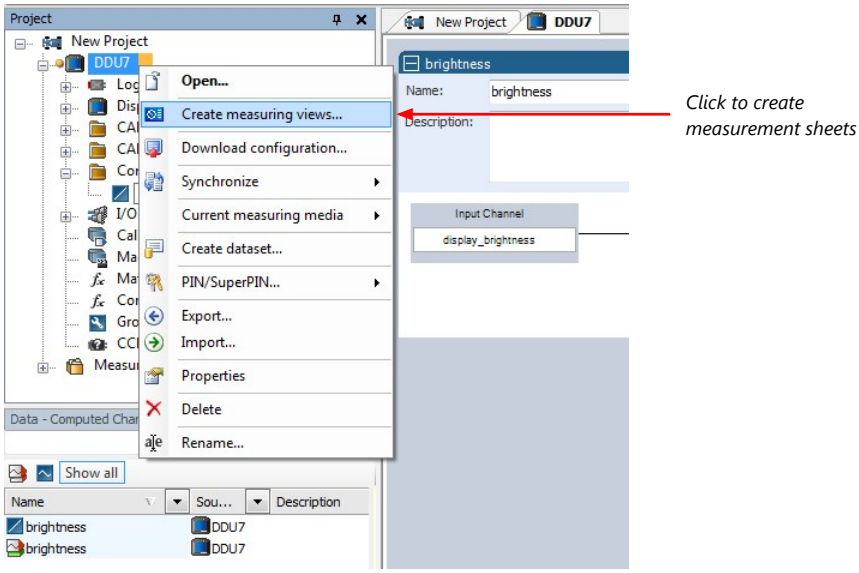




Select existing sheet from list or enter name of new sheet



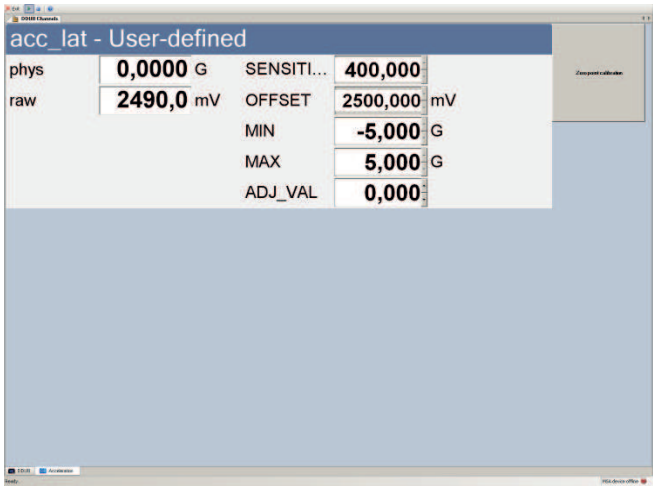
2. To create the sheets, right-click on DDU 11 and select ‘Create measurement views…’ from the DDU 11 context menu.



The automatically created sheet is inserted in the Project Tree under ‘Measurement Container’ and ‘Device Channels’. If the DDU 11 is connected to RaceCon and the status is green, live values of the channels are shown.

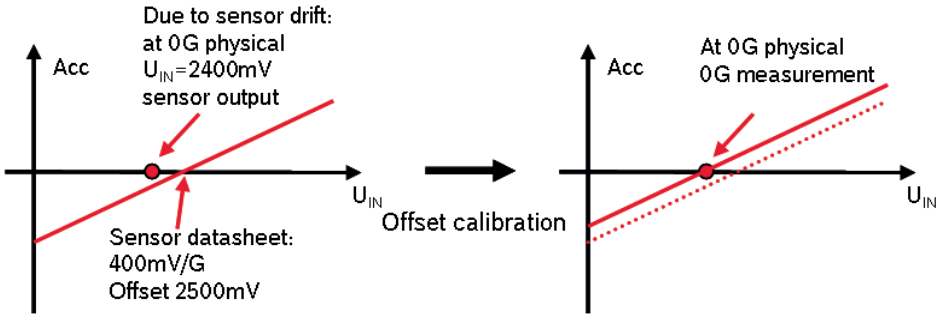
12.1.2 Using the measurement sheets

1. When RaceCon is online, press the ‘F11’ key to switch from ‘Design Mode’ into ‘Race Mode’. The measurement sheet is extended to full screen. The button for offset calibration is active.
2. Switch between different sheets using the tabs at the bottom of the window.
3. Press the ‘Esc’ key to return to ‘Design Mode’.



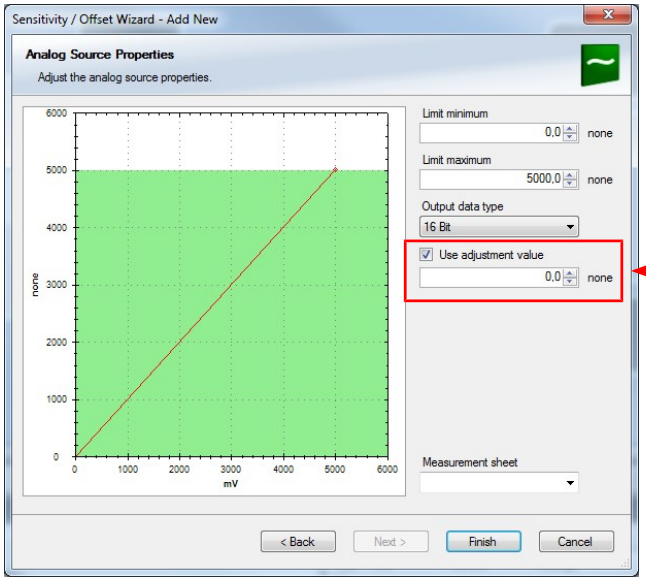
12.2 Online calibration of measurement channels

- Analog sensors drift with age, temperature, etc.
- Manual calibration is necessary
- Solution: online offset calibration
- Example: acceleration sensor



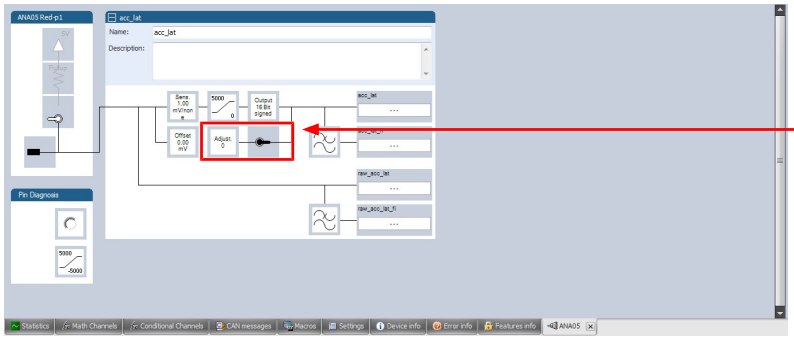
12.2.1 Enable online offset calibration for measurement channel

During creation of the measurement channel



Check box to enable online offset calibration and enter desired physical target value

In the channel view



Activate switch to enable online calibration

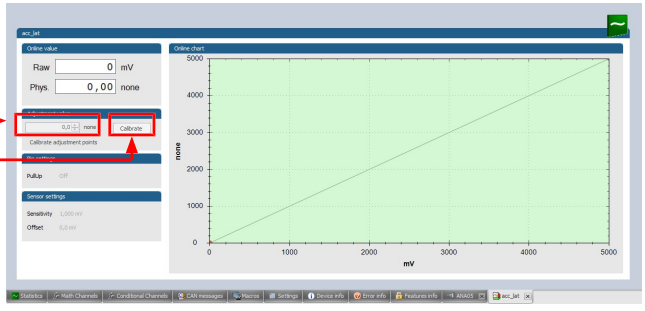
12.2.2 Performing the online offset calibration

DDU 11 has to be connected to RaceCon to calibrate the sensor's offset.

1. Apply the desired physical condition to the sensor (e.g. 1 G to an acceleration sensor).
2. Open the measurement channel's online page by double-clicking on the measurement channel name in the Data Area.
3. Enter the physical target value (e.g. 1 G) and press the 'Calibrate' button.

Calibration target value

Initiate calibration

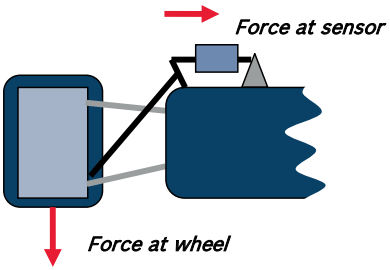


The sensor's offset is now calibrated.

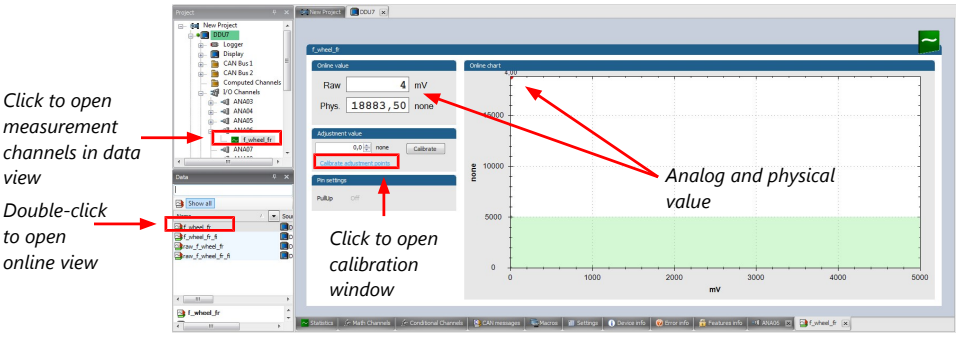
12.3 Online calibration of multipoint adjustment channels

Example: measurement of wheel force

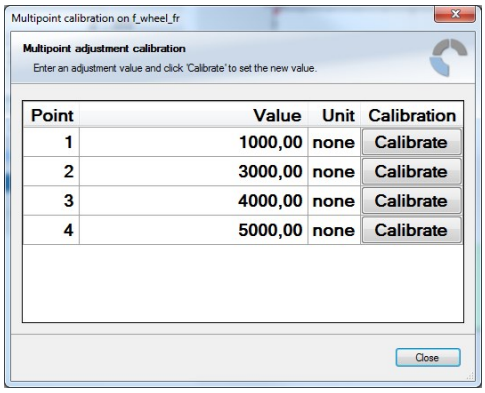
- Physical property ‘wheel force’ not directly measureable
- Load transfer through suspension kinematics
- Physical value at sensor position defined by vehicle
- Curve definition by online adjustment at vehicle



1. Create a multipoint adjustment measurement channel. To create a multipoint channel, see chapter ‘Configuring a multipoint adjustment [▶ 102]’.
2. Download the configuration on the DDU 11. To connect the DDU 11 to RaceCon, see chapter ‘Setting up a new RaceCon Project [▶ 14]’.
3. Click on the desired channel in the DDU 11 Project Tree.
4. Double-click on a measurement channel in the Data Area to open the online view.



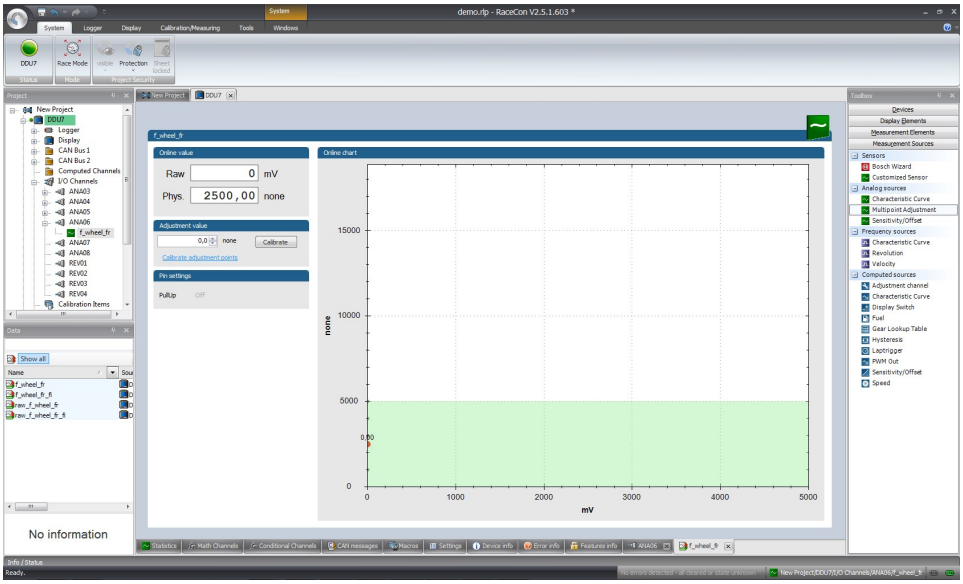
5. Click on ‘Calibrate adjustment points’ to open calibration window.



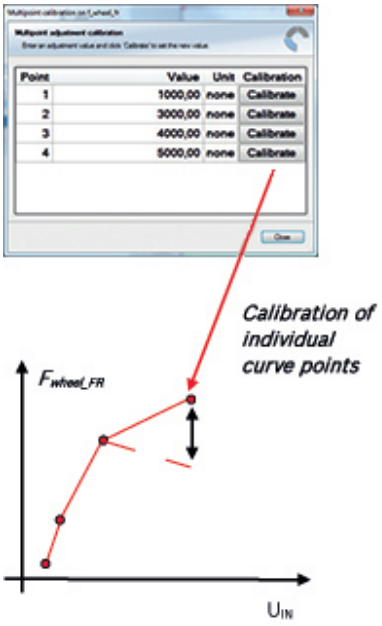
6. Apply the desired physical condition to the sensor (e.g. by applying a force on the wheel).
7. Enter the physical value in the value column of the desired calibration point (e.g. 745 N).

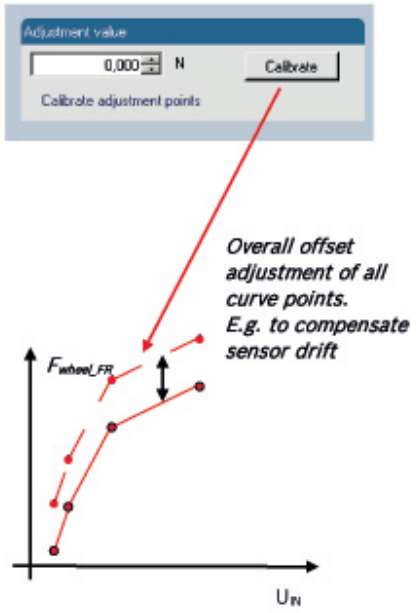
8. Press the ‘Calibrate’ button of the desired calibration point.
9. Repeat for all curve points.
10. Click ‘Close’ when done.

The calibration curve is displayed in the online view.



Adjustment points vs. offset adjustment





12.4 Using the Measurement Sheets

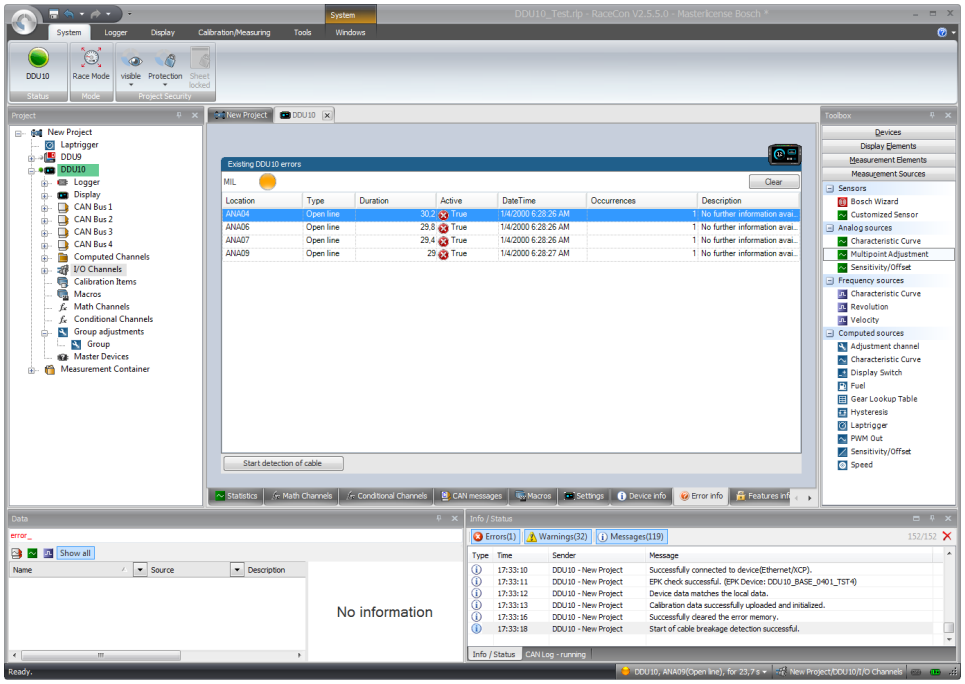
- When RaceCon is online, press “F11” key to switch from Design Mode into Race Mode.
- The measurement sheet is extended to full screen.
- Switch between different sheets using the tabs at the bottom of the page or the keyboard shortcuts associated with the sheets.
- Press ESC key to return to Design Mode.

13 Error Memory

In this chapter “Error Memory”, a lot of screenshots are created by way of example for DDU 8. Please consider this and replace the product name ‘DDU 8’ in this case with the name of your product.

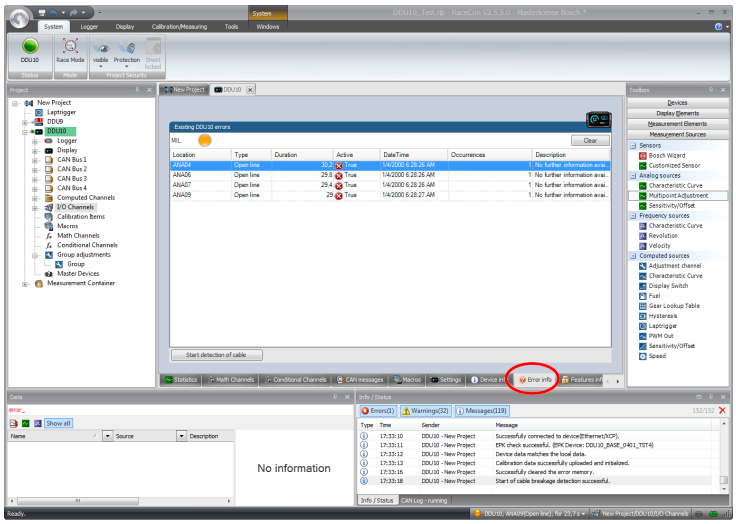
13.1 Error memory representation in RaceCon

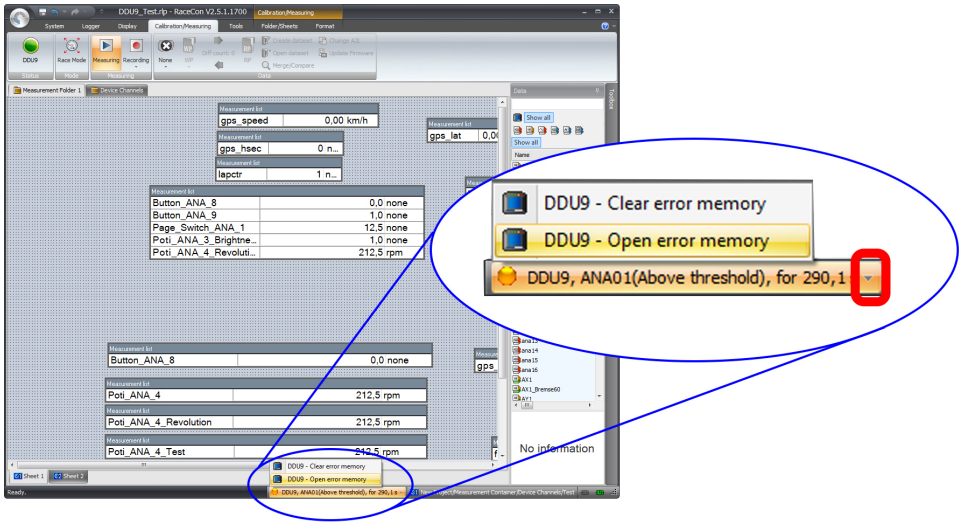
Bosch Motorsport devices feature an error memory. Information on errors can be visualized via RaceCon (online measurement) or can be transmitted via telemetry.



13.1.1 Accessing the memory

The error memory can be accessed as shown in the illustration:





The memory is situated inside the device and is non-volatile. As a consequence, an error which has occurred and has not been cleared by the user will remain in the error memory even after a power cycle. The error state will then reflect if the error is still active or not.

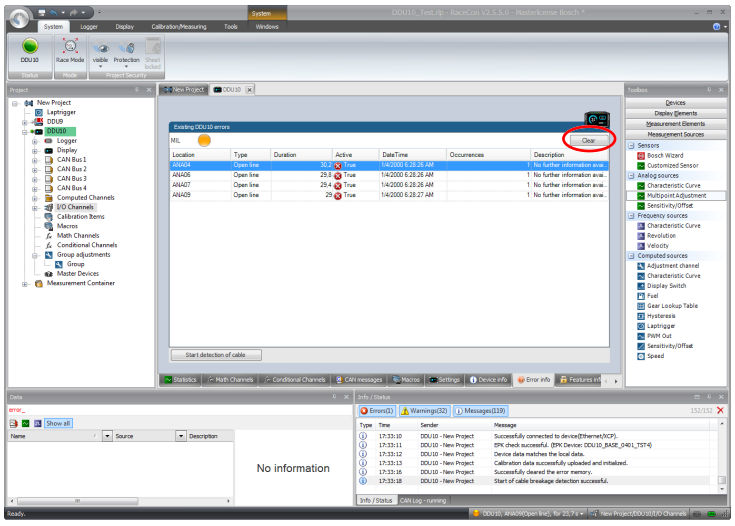
An error is deleted from the list when

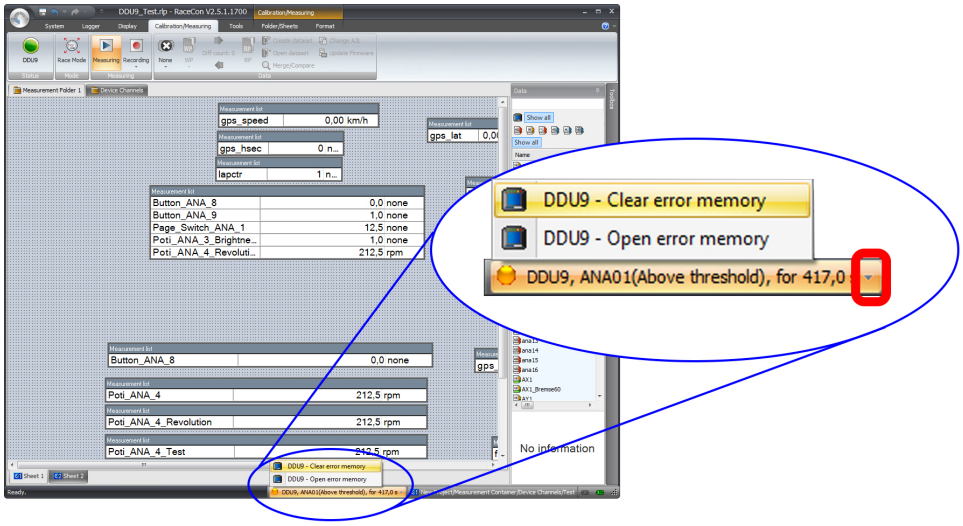
- the user actively clears the error memory
- the user updates the firmware

The error memory is not cleared by a configuration download and is not cleared by a power cycle.

13.1.2 Clearing the error memory

There are two ways of clearing the error memory, both are shown in the following illustration:





13.2 Information on errors available from the error memory

In general, properties of the error memory and properties of an individual error need to be distinguished.

13.2.1 Error Memory Properties

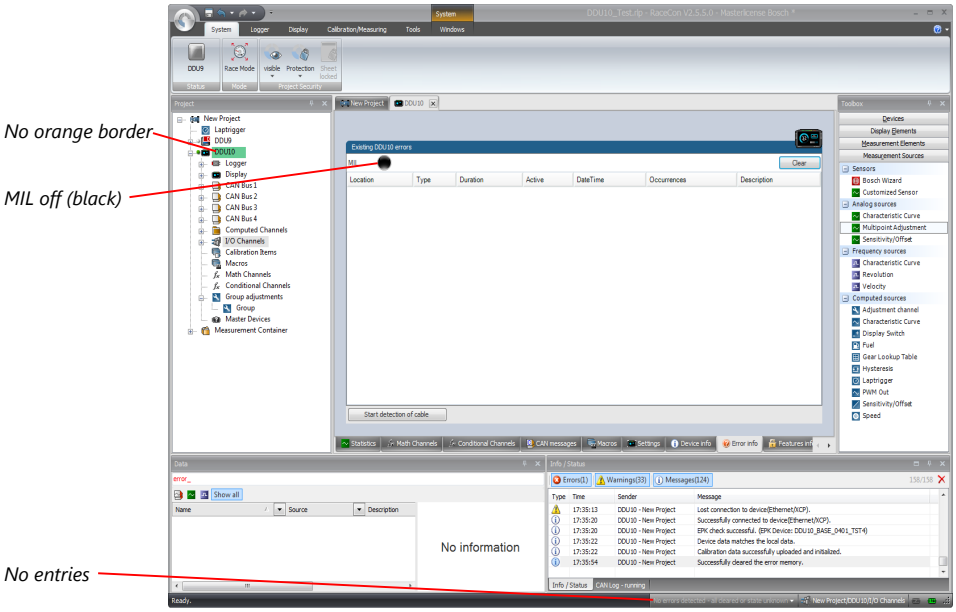
The following property is available for the error memory itself:

- Error Status (device measurement label “error_state”)
  0: no error present in memory
  1: at least one inactive error present in memory, no active errors
  2: at least one active error present in memory

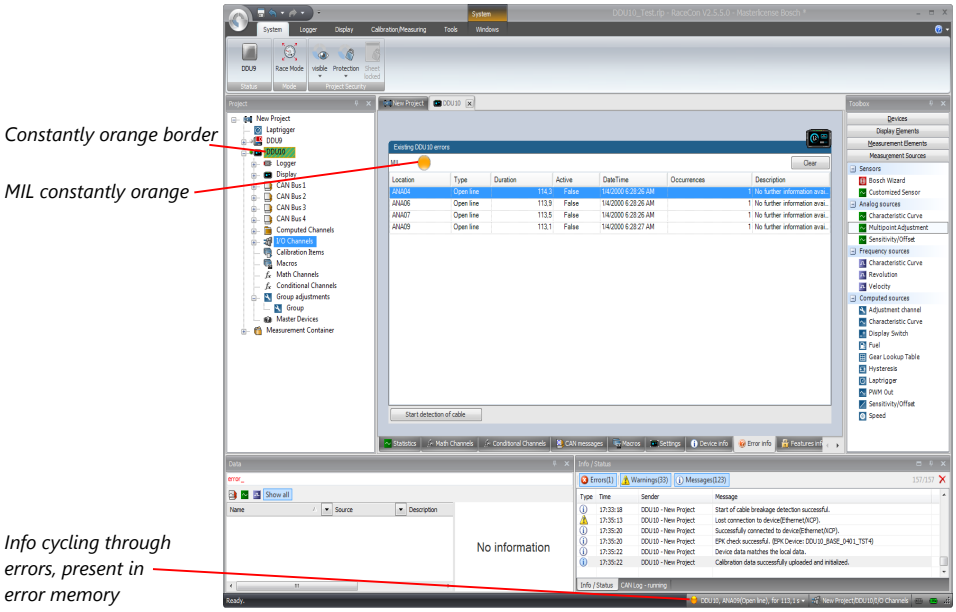
If displayed in a measurement sheet, this property’s value (0, 1 or 2) is translated into a verbal description:

It is also represented by a color scheme within RaceCon (provided RaceCon is online with the system):

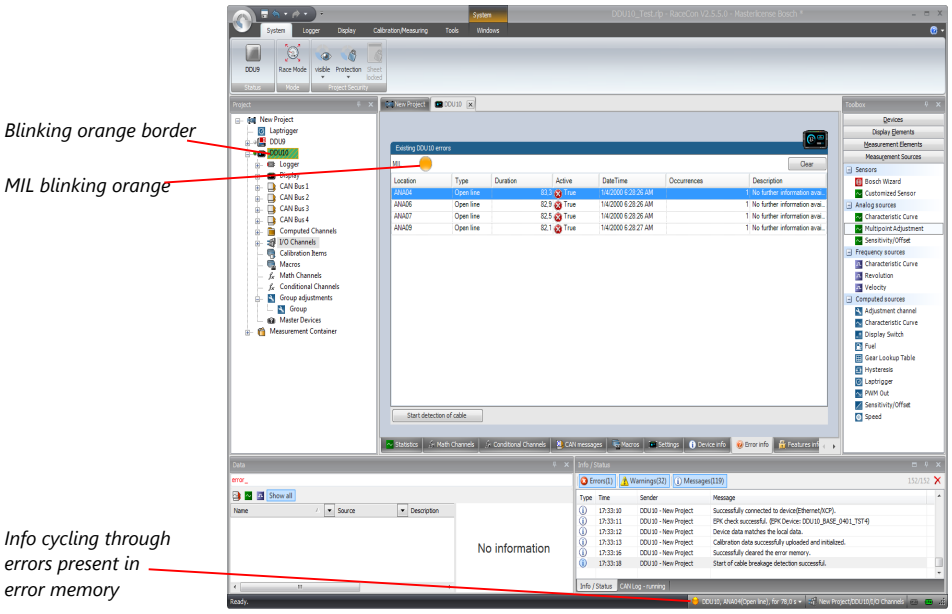
0 (no error present in memory):



1 (at least one inactive error present in memory, no active errors):

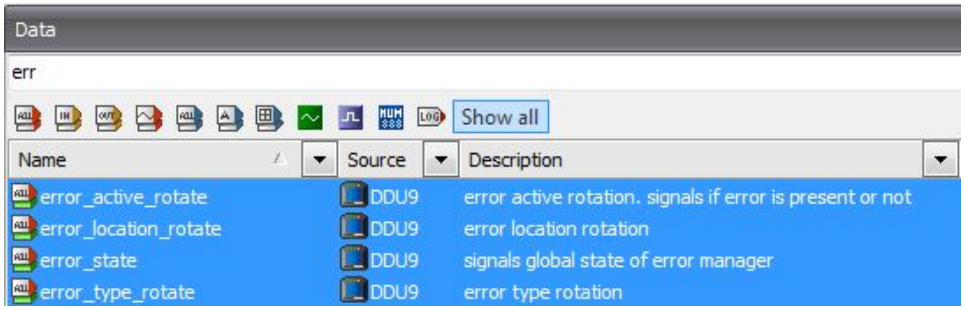


2 (at least one active error present in memory):



13.2.2 Error Properties

The following channels are recognized and memorized inside the devices:

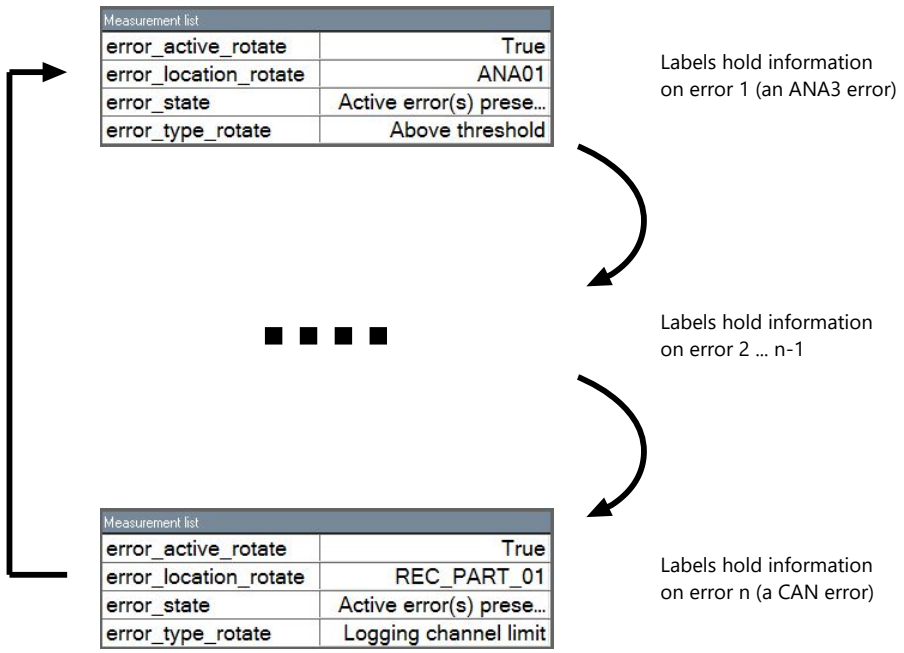


- Error type (device label “error_type_rotate”):
  e.g. “below_threshold” for a violation of the minimum voltage range defined in the configuration, “shortcut_Batt“ for a shortcut to battery voltage etc.
- Error locations (device label “error_location_rotate”):
  e.g. “ANA01” for an error concerning the first ANA channel
- Error durations
  How long has the error been active? If an error encounters a non-active period before being cleared from the memory and is then detected again, the error duration keeps on accumulating. The number of active periods can be seen from the “number of occurrences”.
- Number of occurrences
  How many times has the error been detected since the last time the error memory was cleared.
- Error active state (device label “error_active_rotate”)
  All failure modes are continuously diagnosed; any error detected will be written to the error memory. Once an error is detected, it is qualified as “active”.
  - 1 (TRUE) Error was detected in most recent diagnose run (active)

– 0 (FALSE) Error is inactive: error was not detected in most recent diagnostic run, however the error has not been cleared from the memory by the user and remains in the non-volatile memory

The aforementioned channels (error_active_rotate, error_location_rotate, error_type_rotate) are device specific properties (e.g., C 60) and are not related to the complete RaceCon project (e.g., “error no. 3 from the error memory”). Therefore, only one property label is available in each device. The errors from the error memory (possibly more than one error possible per device) share these three labels. The labels cycle through the errors currently present in the memory and represent the respective property of each error periodically.

The following screenshot shows error properties, which can be displayed or logged:

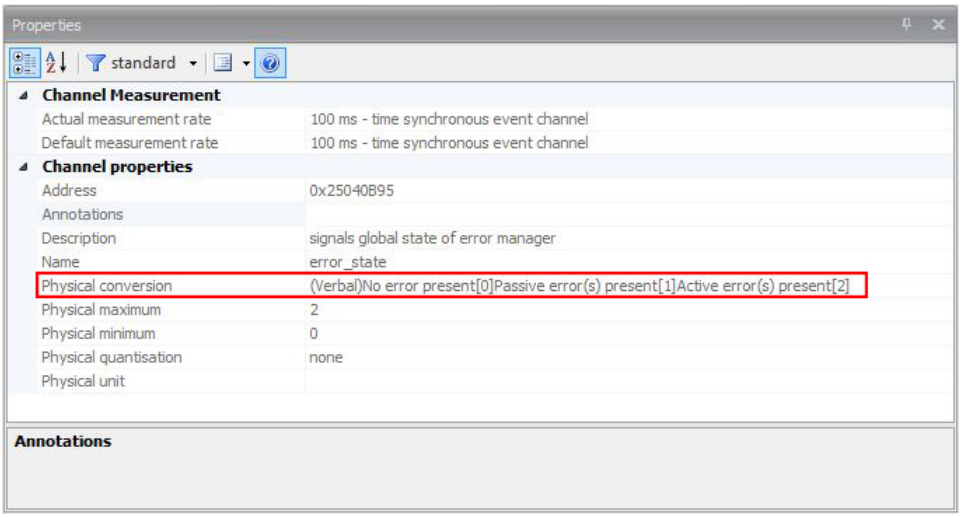


After the last error and its error properties have been displayed, the labels will start again with the first error in the error memory stack and its error properties will be displayed again. Therefore, monitoring these labels over a sufficiently long period provides the information on all individual errors in the error memory.

To understand this behavior, it is recommended to observe the three labels in a measurement sheet (while more than one error is active) and watch the values change periodically:

Measurement list	
error_active_rotate	True
error_location_rotate	REC_PART_01
error_state	Active error(s) prese...
error_type_rotate	Logging channel limit

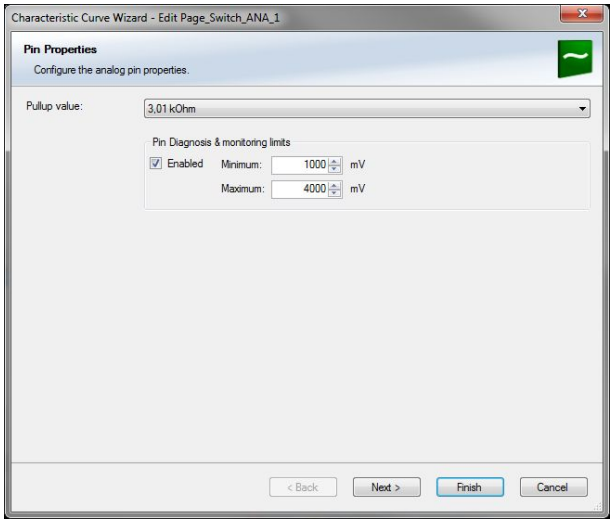
The verbal representation of the numerical codes of these labels can be visualized in the properties window of the measurement page:

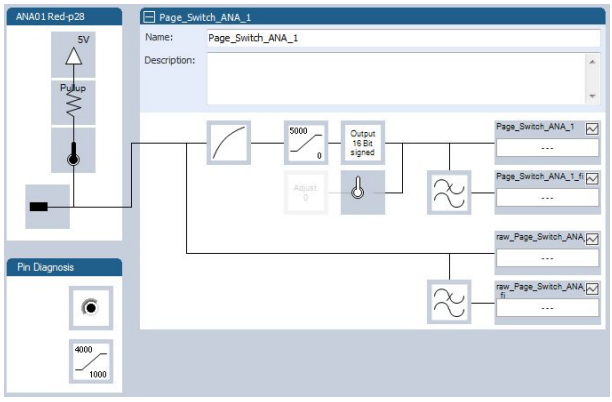


13.3 Analog Input Diagnosis

13.3.1 Monitoring limits / Shortcut Detection / Cable Breakage

The pin diagnosis functionality (check whether measurement is within the desired range) can be activated in the ANA pin setup wizard; to allow for a diagnosis regarding shortcut to ground, shortcut to battery voltage and cable breakage, a minimum / maximum has to be defined.

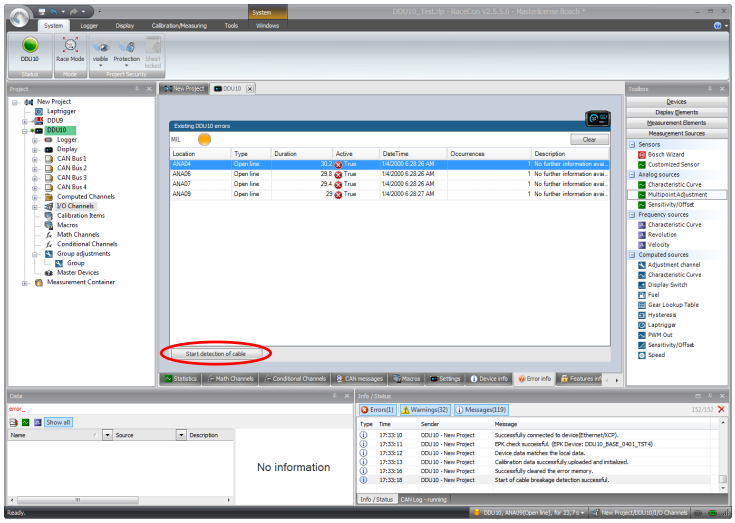




13.3.2 Open Line Detection

The implementation of open line detection consists of pull up resistors being activated and deactivated; evaluating the behavior of the measured value detects cable breakage, regardless of the pull up resistor being activated by the user.

1. Open the Error Memory of the Device.
2. Click "start detection of cable".
3. Check the Error Memory for new fault entries, regarding "Open line errors".



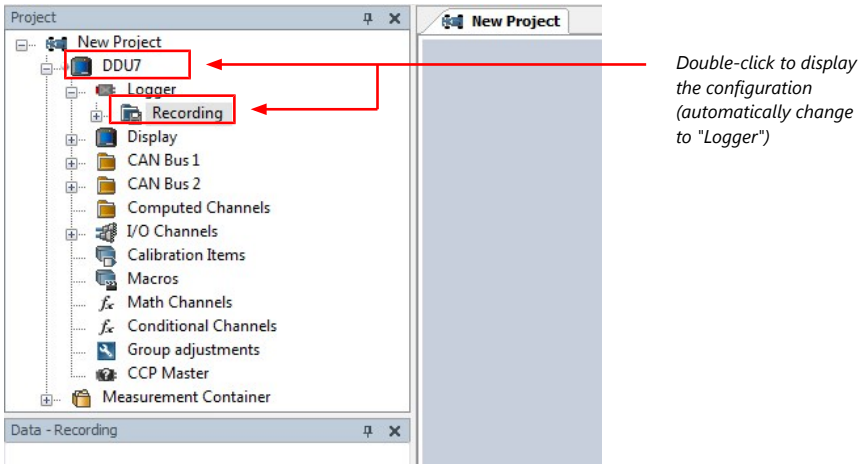
14 Recording

14.1 Features

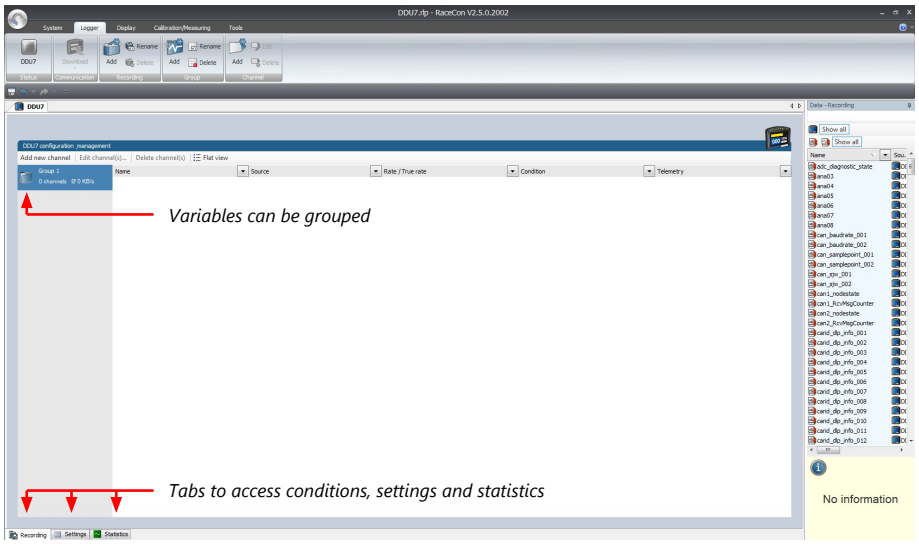
- Synchronized recording of DDU 11 analog and digital input channels, DDU 11 internal measurement channels, ECU data, Data from external sensor interfaces
- Up to two independent recordings
- Measurement rate 1 ms to 1 s
- Two global start conditions (thresholds)
- Up to 16 measurement conditions (fast-slow-switches)

14.2 Configuration of recordings

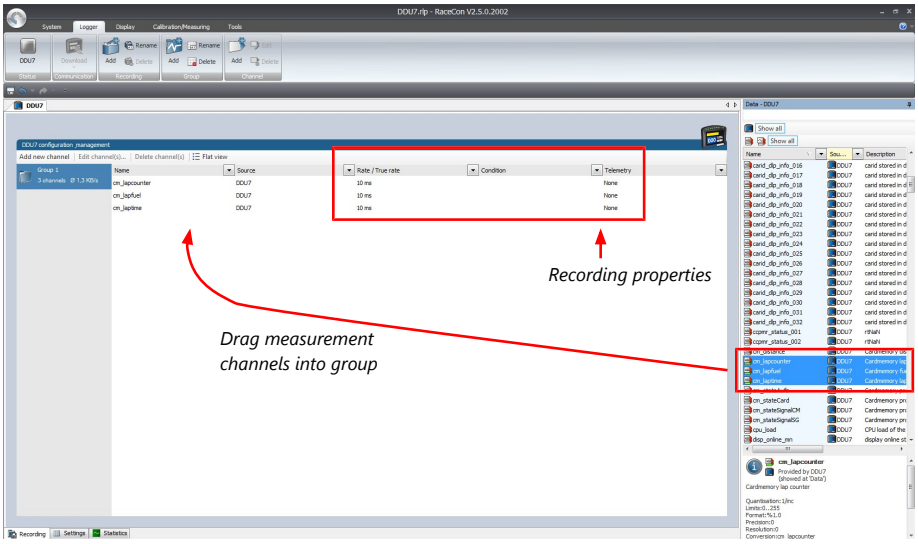
1. Expand the list of 'Loggers' by clicking on '+' in the DDU 11 Project Tree.



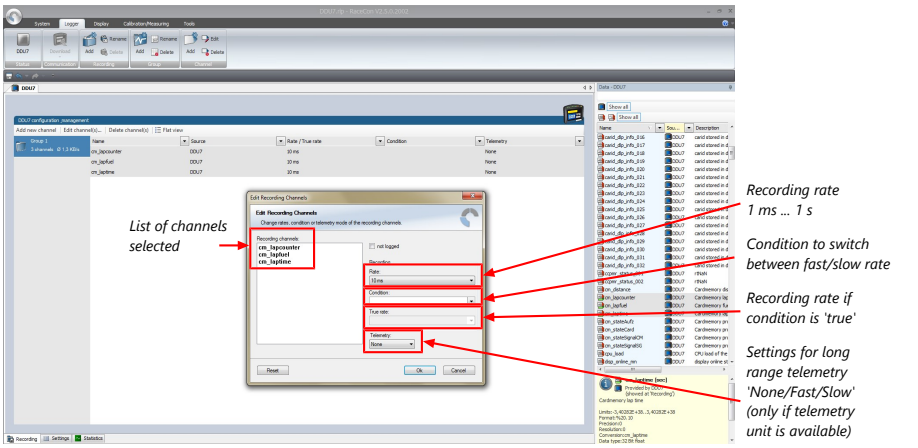
2. Double-click on 'Recording' in DDU 11 Project Tree. The recording configuration is displayed in the Main Area.



3. To add measurement channels to a recording, click 'DDU 11' in the DDU 11 Project Tree. In the Data Area, the measurement channels are displayed.
4. Drag and drop desired measurement channels into recording group.



5. To edit channel's settings, mark the channel(s) and click 'Edit Channel'. An 'Edit Recording Channels' window opens.



6. Click 'OK' when done.

NOTICE

If no condition is defined or condition is 'false', measurement channels are recorded at the value chosen in 'Rate'.

If the condition is 'true', measurement channels are recorded at the value chosen in 'True rate'.

Using fast block/slow block transmission

DDU 11 telemetry uses available bandwidth of Telemetry Unit FM 40 (19,200 baud -> approx. 1,700 bytes/s). The bandwidth has to be divided into channel information to be transmitted high-frequently and low-frequently using the 'fast/ slow block' setting.

Channels are grouped into 8 blocks which are transferred each cycle:

- Fast block (Block 1) is transferred every cycle and used for a high-frequent transmission of channel information (e.g. speed, rpm).
- Slow blocks (Block 2...n) are transferred every n-th cycle and used for a low-frequent transmission of channel information (e.g. tire pressure, oil temperature).

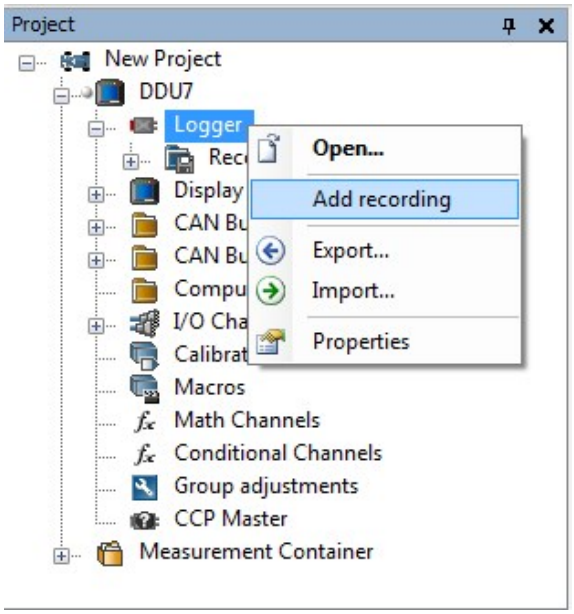
Transmission Scheme

If the maximum bandwidth of a block is reached, a warning will be displayed. To fix this problem you can view the allocation of the channels and data rate in the 'Statistics' tab of the Main Area. See chapter 'Recording statistics [▶ 134]' for more information.

14.2.1 Adding a recording

DDU 11 supports up to two independent recordings.

To add a recording, select 'Add Recording' from the context menu of the Logger in the DDU 11 Project Tree.

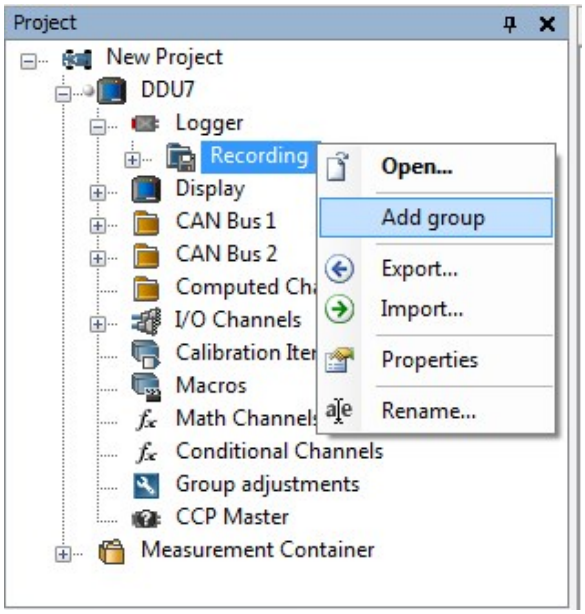


Maximum two recordings are possible. In the device software the 2nd recording is reserved for scruteneering data. This recording is invisible (protected).

14.2.2 Adding a recording group

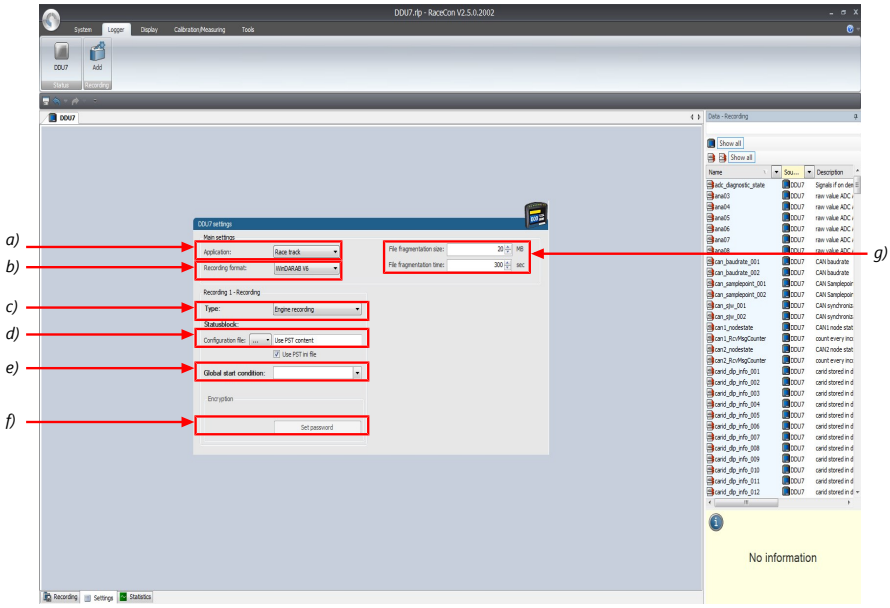
Recording channels can be grouped.

To add a new group, select 'Add group' in the context menu of the recording. The groups can be renamed to 'Gearbox', 'Aero', 'Engine', etc.



14.2.3 Global settings

To display the global DDU 11 settings, select the ‘Settings’ Tab.



a) Choose setting for outing counter mode:

- For testbench (without lap trigger) select ‘Testbench’.
- For racetrack (with lap trigger) select ‘Racetrack’.

b) Choose your WinDarab version. In V6 the file is encrypted by WinDarab. In V7 you can enter an optional self created password in the ‘Encryption’ field shown in f).

c) Recording Type (Engine or Chassis).

d) Statusblock configuration file for custom Statusblock definition.

e) Choose or create the condition to start recording.

f) If selecting WinDarab V7 in b), enter a password hint and a password (optional).

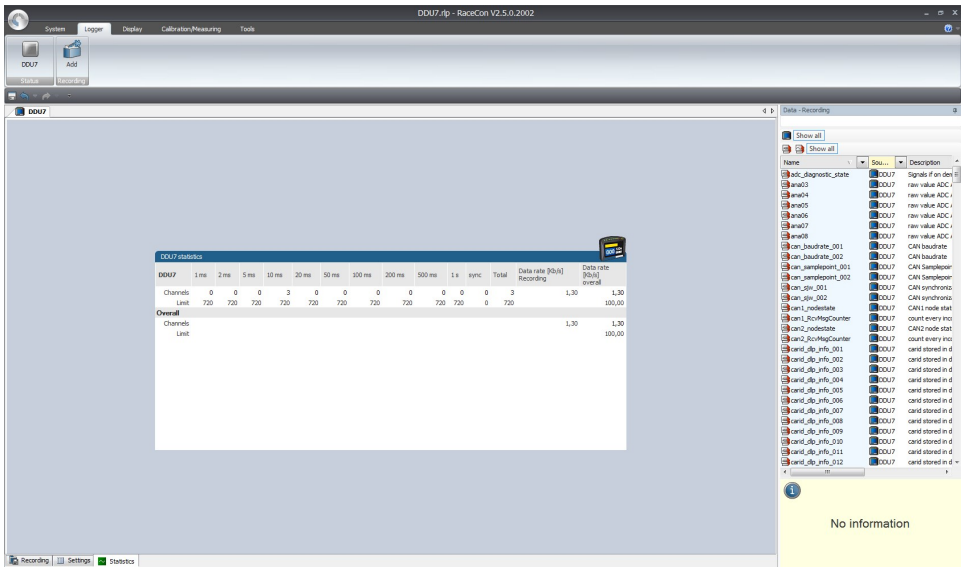
g) Setting for automatic fragmentation. Do not change!

14.2.4 Recording statistics

The tab ‘Statistics’ shows the channels’ allocation and their current data rate related to the transmission frequency of the DDU 11 and the whole transmission system.

The overview helps to detect bandwidth bottlenecks of channels. Bandwidth bottlenecks can be solved by changing the ‘fast/slow block’ setting for each channel.

The data rate of the whole system is often less than the data rate of the DDU 11 and limits the overall transmission speed.



14.2.5 Recording diagnosis

The channel ‘statectrl_ok’ of the DDU 11 can be used for online monitoring of recording status.

Bit	Value	Name
0	1	RECORD
1	2	DATAOK
2	4	BLKOK
3	8	-
4	16	-
5	32	-
6	64	STARTED
7	128	-

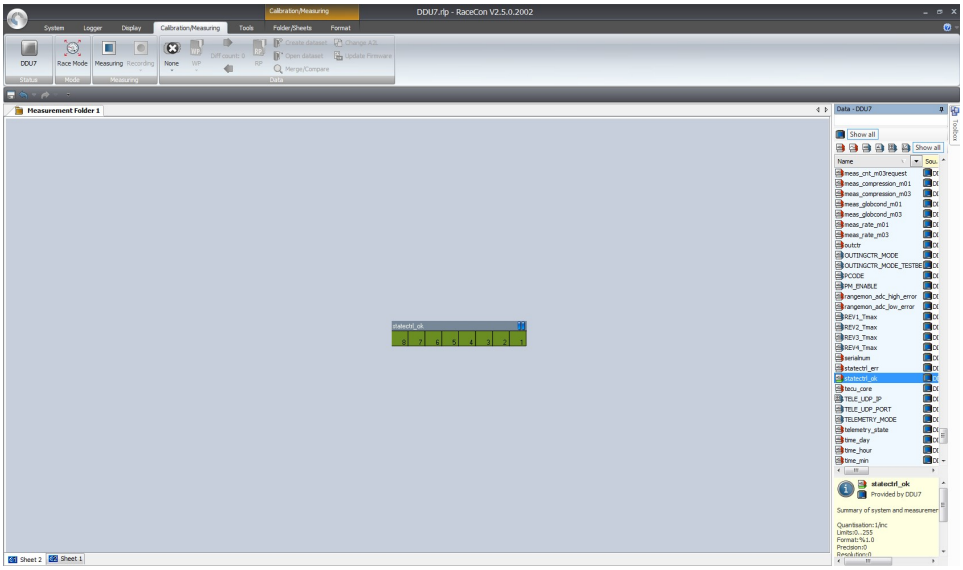
Content of status bits

Name	Bitset	Bit cleared
RECORD	Measurement data is recorded.	No measurement data will be stored because measurement thresholds are not reached.
DATAOK	Received data without error.	Discarding received data because of wrong timestamps. Check wiring of SYNC signal.
BLKOK	All measurement blocks have been set up correctly.	Some measurement blocks have not been set up correctly.

Name	Bitset	Bit cleared
STARTED	A measurement has been set up.	A measurement is not set up. Either no recording configuration has been found or logger software upgrade is not activated.

14.2.6 Displaying online recording diagnosis ('statectrl_ok')

1. To add a Recording Diagnosis element to a measurement sheet, change to page "Calibration/Measuring" and drag a 'Bit-LED' element from the Toolbox and drop it on measurement sheet.
2. Drag channel 'statectrl_ok' from the Data Area and drop it on the 'Bit-LED' element.



The 'Bit-LED' element shows the state of received channel data in bit-representation. A green highlighted channel means 0, a red highlighted channel means 1.

- Measurement correctly initialized, but recording threshold(s) not reached: 254
- Measurement correctly initialized, DDU 11 is recording data: 255
- Values less than 254 indicate an error state
- 'statectrl_ok' can be linked to an alarm on the display. See chapter ''Alarm' display element' for details.

14.2.7 Further measurement labels

These additional measurement labels may help you diagnosing the state and operation of the data logging in more detail. There are a few more, but these are usually enough. Please refer to statectrl_ok, mentioned in more detail in chapter 'Recording diagnosis'.

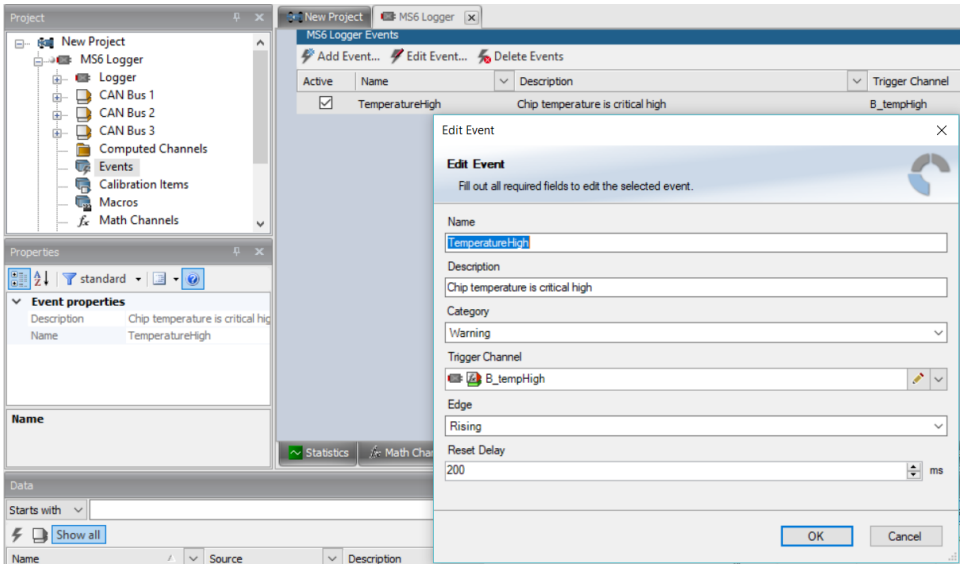
Measurement label	Function
card_part1_size	Size of the first logging data partition in MB.
card_part2_size	Size of the second logging data partition in MB. Attention, second logging can also be stored on first partition, depending on chosen settings (Logger -> Settings).
ftp_UserLoggedIn	This measurement allows to monitor for active FTP connections. RaceCon (WinDCP) and WinDarab may not connect in parallel.
meas_globcond_m01 / _m03	State of the global logging start condition for first / second logging. TRUE means data is actively recorded.
meas_rate_m01 / _m03	Incoming measurement data rate (first / second logging) for further processing. Does not include compression. Active when meas_globcond_m0x is TRUE but may also be active while meas_globcond_m0x is FALSE, if a pretrigger time is configured. In that case data is transferred to the pretrigger buffer, but not necessarily written to storage medias.
meas_cnt_ecu / _fde	Processed data blocks for first / second logging. This does not ensure writing the data to a storage media, e.g., if pretrigger is configured and meas_globcond is FALSE.
meas_cnt_int / _forked	Processed data blocks per media (internal / USB).
meas_compression_m01 / _m03	Compression factor for first / second logging. For example, factor 2.0 means incoming data can be reduced to half the size, before data is written to storage medias.
meas_pretrig_buf_size_ecu / _fde	Size of data buffered in pretrigger, e.g., while global logging condition is FALSE. Data will be forwarded to storage medias when logging condition becomes TRUE.
meas_backend_buf_size_ecu / _fde	Size of data buffered (for first / second logging) for processing by different storage medias (intern / USB). It is possible, that e.g., internal storage has processed the data already, while USB is still busy writing the data blocks. Data is removed from the buffer as soon as all medias have processed it.
meas_write_rate_intern_001 / _002	Effective data write rate to internal storage media, after compression, for first / second logging.
meas_write_rate_usb_001 / _002	Effective data write rate to USB storage media, after compression, for first / second logging.

14.3 Event logging

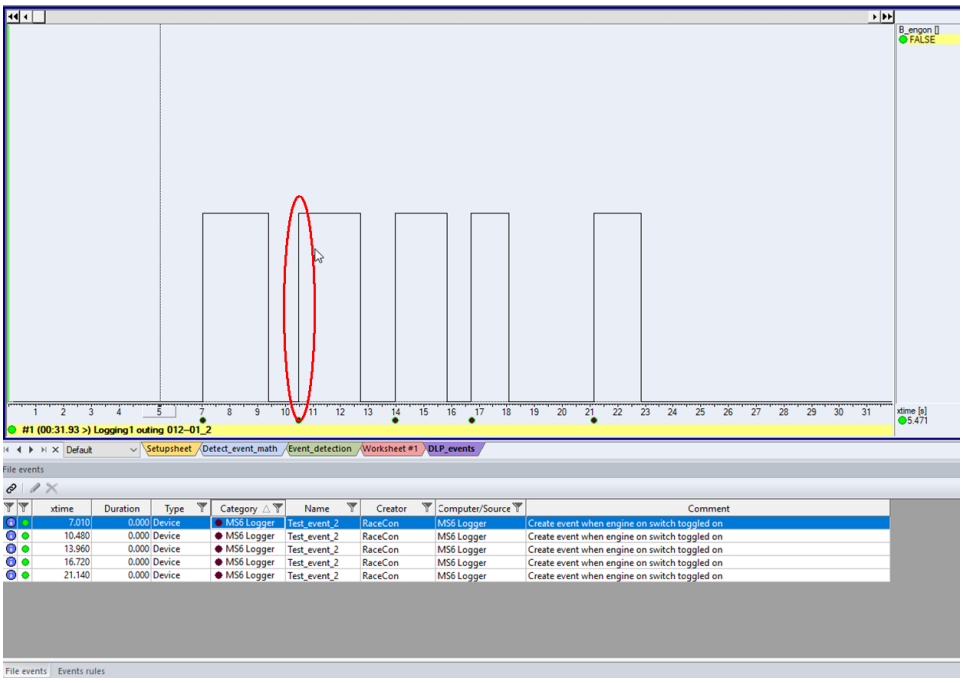
Event Logging implements the possibility to observe a channel if short spikes are expected. With Event Logging, every occurrence of a user defined threshold (more complex conditions are possible) leads to an event being raised. It is listed in a table along with its time stamp, its ID and even with a text string freely definable in RaceCon.

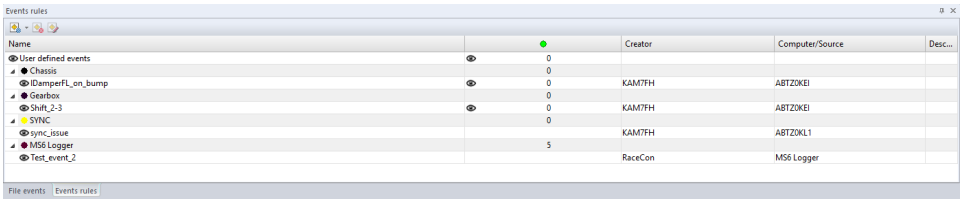
Events are stored as text in logging data and displayed in WinDarab like Darab-Events. Possible use cases are error entry, etc.

Configuration in RaceCon:



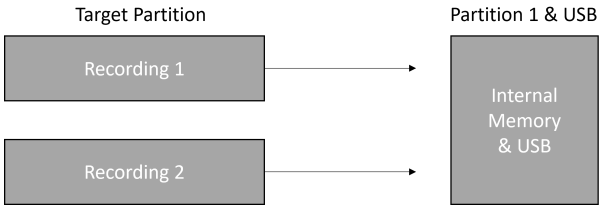
Display in WinDarab:





14.4 Data Logger and USB recording

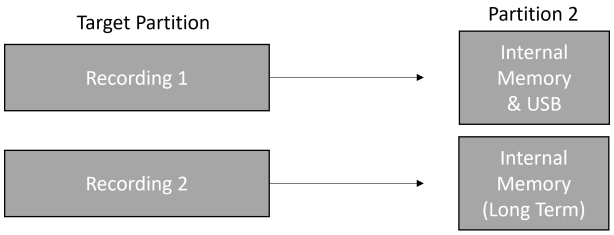
Default settings:



Data from **Recording 1** and **Recording 2** are stored both into the Internal Memory and additionally on the USB stick in copy.

To download the data from the Internal Memory of the logger, the Data Logger must be selected in WinDarab and the data will be downloaded in parallel.

Alternative setting:



Recording 1 is stored on the Internal Memory and additionally on the USB stick in copy. To download this data, the Data Logger must be selected in WinDarab.

Recording 2 is stored on only the Internal Memory. To download this data, the Long Term logger must be selected in WinDarab.

14.5 USB recording

This function requires the installation of Software Upgrades. Look into the datasheet of your device, to see which upgrades are available for your device.

Software Upgrade USB_DATA enables USB recording. To activate Software Upgrade USB_DATA, enter the license key as described in the chapter 'Feature activation' [▷ 19].

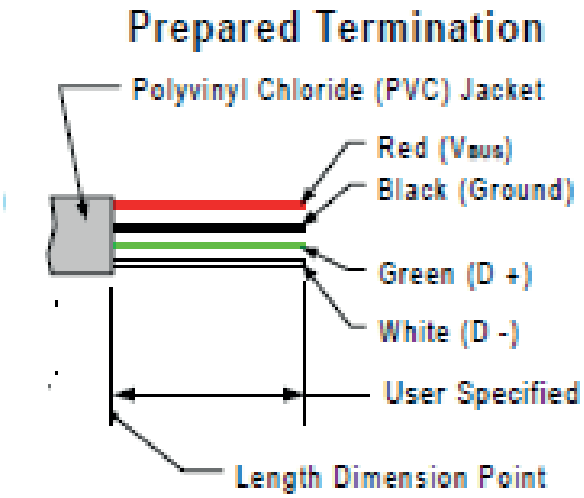
For USB recording, Software Upgrade FULL_LOG_1 should also be enabled.

Wiring harness

Bit	Value
USB_Device_Power	Power (red)
USB_Device_DP	D+ (green)
USB_Device_DN	D- (white)
USB_Device_Gnd	GND (black)

For further information, see the pinlayout of the device.

Colors matching a standard USB cable



Storage device

The recording function can be used with a dedicated Bosch Motorsport USB device. The USB device must be preformatted with the Bosch File System (BFS) in RaceCon before first use.

To format the USB device with the Bosch File System (BFS), do the following steps:

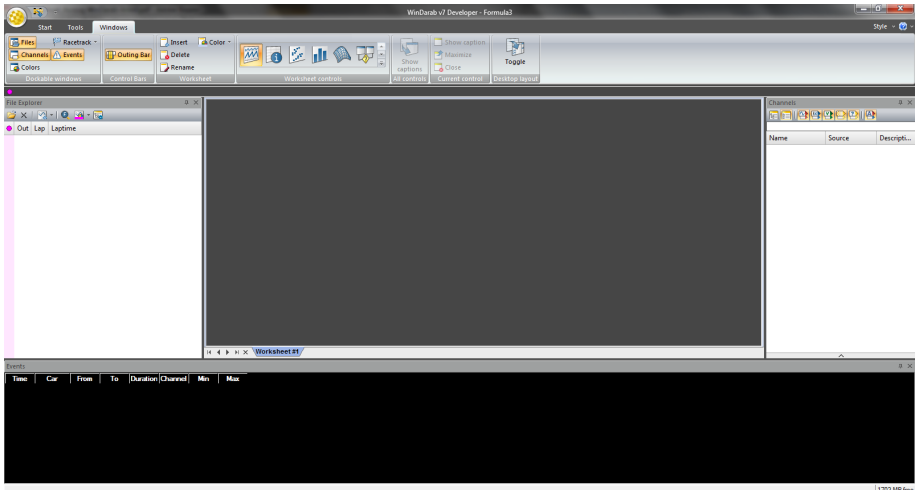
In RaceCon, select 'Tools' - 'Extras' and choose 'Format USB stick'.

Press 'Format'.

An USB device is recognized by Windows as a 'storage medium', but it can only be initialized with RaceCon and read with WinDarab.

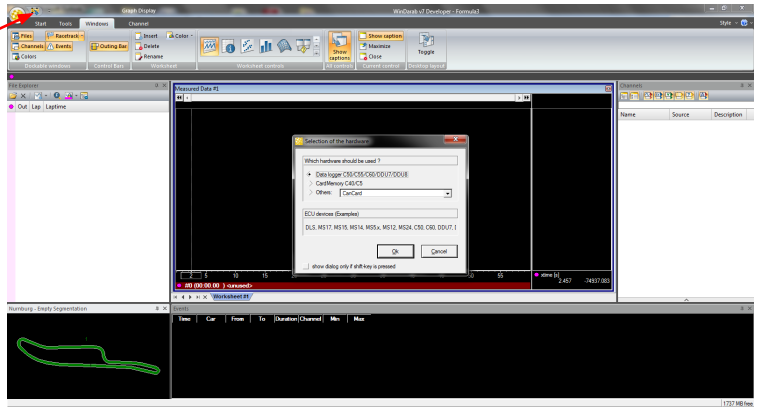
14.5.1 Recording data on USB device

1. Plug an USB device to DDU 11.
2. Prepare a recording configuration in RaceCon.
3. Power on the system and connect with RaceCon to the vehicle.
4. Download the configuration to the DDU 11.
5. Record measurement data. If an USB device is present, the DDU 11 stores the data in parallel on the internal memory and the USB device.
6. Power off the system.
7. Remove USB device from the vehicle.
8. Start the WinDarab software.

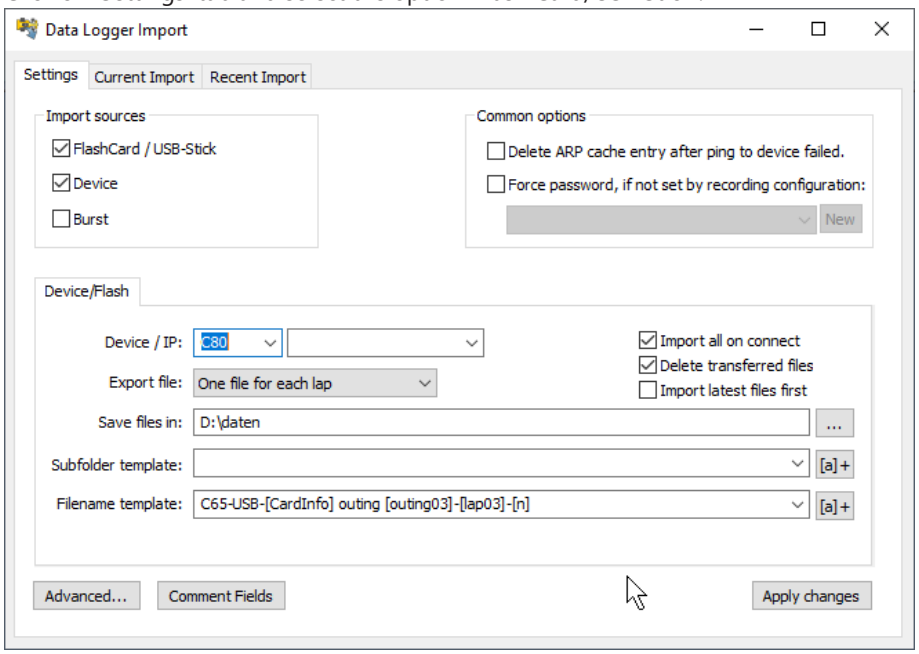


9. Click on the ‘Import/Export’ icon.
10. Select ‘Data logger CXX/DDUX/MSX and click ‘OK’ when done. The ‘Read measurement data’ dialog opens.

Click 'read data from logging device'

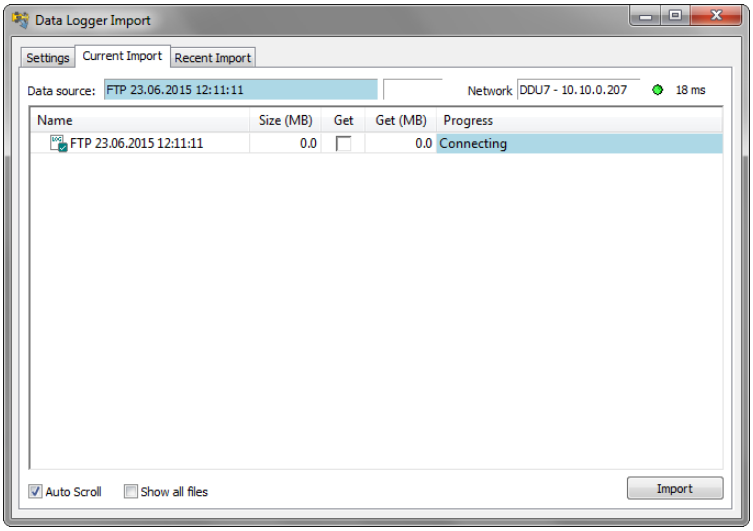


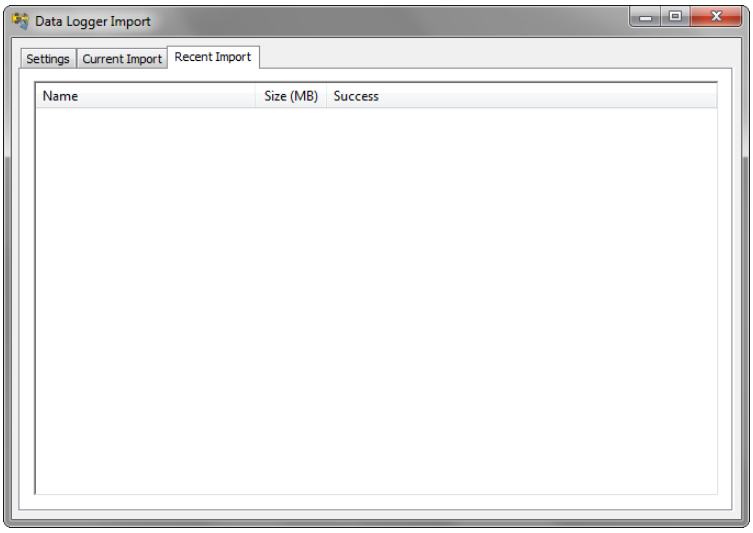
11. Click on ‘Settings’ tab and select the option ‘Flash Card/USB Stick’.



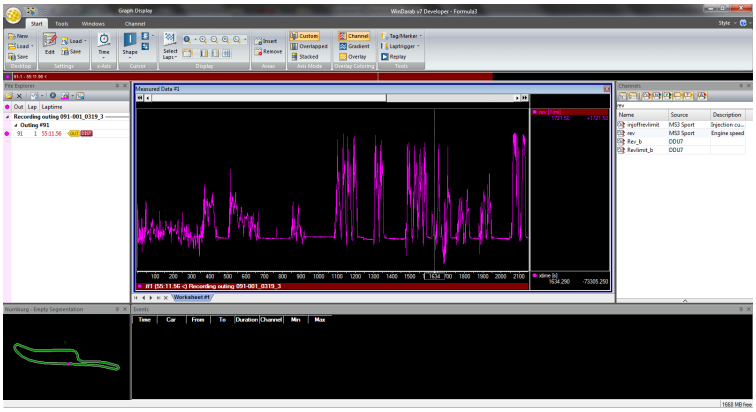
12. Activate ‘Apply changes’.

    Insert the USB device into the PC. Data transmission from device starts automatically. Measurement files are stored automatically in the base folder.





13. Click ‘Close’ when transmission has finished.
14. Click on the Start button and choose ‘Open measurement file’.
15. Select the measurement files from the storage folder.
16. Click on ‘Open’.
17. Click in ‘New Desktop‘ to open a new measurement data window.
18. Drag the desired measurement channel from the Channel list and drop it into the measurement data window. The measurement channel‘s graph is displayed



For more detailed descriptions and instructions, refer to the WinDarab V7 manual.

14.5.2 USB device handling hints

Using the USB device

Always plug the USB device into vehicle before power up to ensure that all measurement data is stored on the USB device.

If the USB device is plugged in after recording has started, only the current data is saved.

Data recorded on the DDU 11 before the USB device is plugged in will not be saved.

Removing the USB device

Always power off the system before unplugging the USB device!

14.5.3 Troubleshooting

When no data on the USB device is recorded:

Configure the measurement label **usb_mediastate** on a RaceCon measurement view or on a DDU 11 display page.

The value of **usb_mediastate** reflects the operating condition of the USB bus:

State	Description
0: Wait: Device not found	The USB device is not found (also: waiting for re-plug stick). No USB device inserted. USB device is defect. No electrical connection or wiring harness problem. USB software upgrade not activated (Purchase of unlock code needed).
1: Wait: Device detected	An USB device is found, but not yet installed.
2: Ok: Media installed	The USB device is found and is operational (idle). This does not imply that recording data is written!
4: Stop: Device unplugged	The USB device has been removed. The DDU 11 performs a restart when an USB device is re-plugged in.
5: Error: Media error	The communication to the USB device broke down. The USB device is defect. The USB device is not supported by DDU 11.
6: Error: Media corrupt	The USB device is not in valid BFS format. (Hint: Re-format the USB device in RaceCon.)

15 Ambient Light Sensor

The DDU 10 is equipped with an ambient light sensor. The sensor is located at the front of the DDU 10 below the LEDs on the right side. The sensor is linked to the channel “eamb” with the unit lux.

The ambient light sensor can be used to calculate an automatic brightness switch. To use this function, please follow the steps below.

1. Measure the range of light intensity in your vehicle during all relevant conditions.

   The easiest option to evaluate the needed range is to log the “eamb” channel and expose the vehicle to the brightest and darkest conditions it will face during usage.

   In this example, the expected value will be 10 lx as the lowest and 300 lx as the highest measured light intensity in the cockpit.

2. Drag and drop a “Characteristic Curve” from the “Computed Sources” onto the “Computed Channels”.

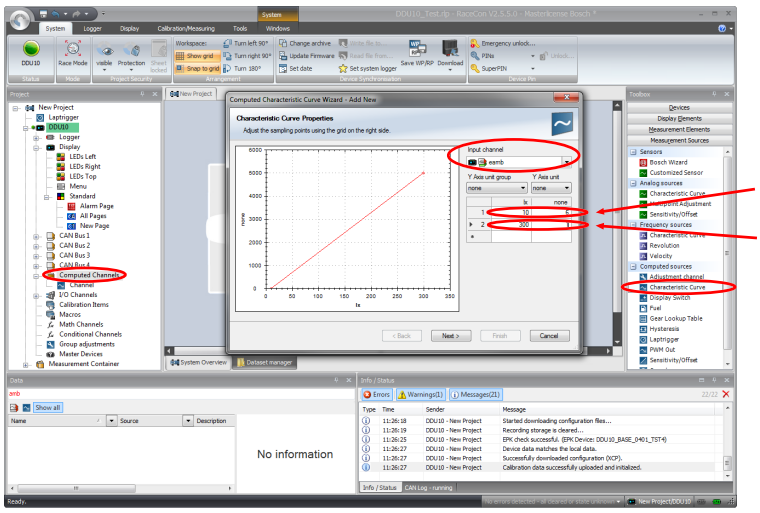
   This will open the configuration wizard.

3. Select “eamb” as input channel.

4. Assign the lowest lux value to the highest possible brightness switch setting (6) and the highest lux value to the lowest switch position (1).

   This will generate a linear change of the switch. If needed you can include additional points to change the linear behavior.

5. Click “Next” when all points are defined as needed.



6. The “Analog Source Properties” window opens in the wizard. The maximum and minimum value of the channel is automatically assigned according to the previous settings. In general, it should be from 1 to 6.

7. Click “Finish”.

8. Assign a suiting name to the new channel and click “OK”.

   In this example, the channel will be named “Ambient_Light_Switch_Computed”

   To avoid a toggling brightness setting, a math channel with a kind of hysteresis has to be created.

9. Double click on “Math Channels” and click on “Add channel”.

10. Copy the Formula below into the new math channel and replace the example name with your name of the computed channel.

latch(#DDU10#Ambient_Light_Switch_Computed+0.5, *abs*(#DDU10#Ambient_Light_Switch_Computed-#DDU10#display_brightness) > 0.6)

The formula compares the brightness setting of the device with the computed channel. If the deviation is within a certain range, it allows sending of the new value to the device. This formula only works correctly if it is used as the brightness switch channel.

11. Assign a suiting name to the new channel and click “OK”.
12. Go to “Display” - “Settings” and define the created math channel as the brightness switch channel.
13. Define the percentagewise brightness of the display and the LEDs according to each switch position.

16 Lap Trigger

16.1 Lap trigger (timing beacon)

Why do we need a lap trigger (timing beacon)?

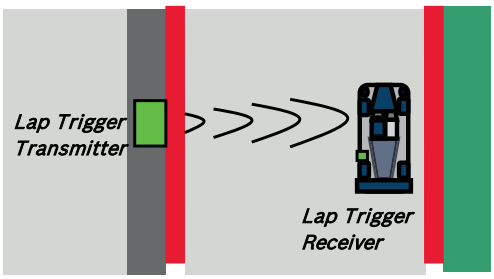
- Vehicle lap time measurement
- Calculation of lap-dependent functions (lap fuel consumption, min/max values)
- Calculation of lap distance dependent functions
- Control of data logging system

Types of Systems

- GPS based (low cost, low precision)
- IR based (low cost, high precision, limited reliability)
- RF (microwave) based (high precision, high reliability)

IR and RF based Systems consists of

- Transmitter (trackside unit)
- Receiver (in-vehicle unit)



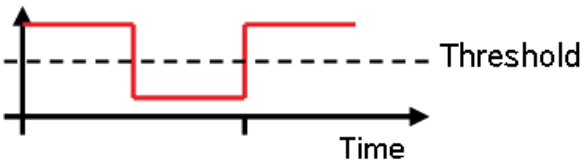
16.1.1 Electrical trigger signal

In DDU 11 all sources of measurement channels can be used as trigger signal.

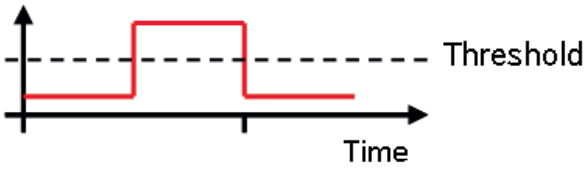
- Analog input
- Digital input
- CAN input

Signal (measurement channel) properties

Low active signal (Bosch triggers): Trigger releases if signal is below the threshold.



High active signal (other manufacturer's triggers): Trigger releases if signal is above the threshold.



Two types of trigger signal:

- Main trigger (end-of-lap at start/finish line)
- Sub-trigger (segment time, optional, not applicable with GPS lap trigger)

Bosch standard:

- Main trigger 20 ms, low active (Recommendation for RaceCon “Detecion Time” setting: 15 ms, Setting must be a slightly shorter period than the signal length of the trigger to avoid a missed trigger due to the update rate)
- Sub trigger 40 ms, low active (Recommendation for RaceCon “Detecion Time” setting: 30 ms)

16.1.2 GPS Lap trigger

The GPS lap trigger uses a GPS signal to trigger the lap timer. To function this timer an external GPS sensor (see GPS Sensor [▶ 160]) has to be connected to the device and a detection point with a detection range has to be defined in RaceCon.

The GPS detection point is defined by the latitude and longitude. The easiest way to get the latitude and longitude of a finishing line is due to a web mapping program such as google maps. With google maps, simply left-klick on the spot where you want to set the detection point. The information about the latitude and longitude will show up, in general the latitude is given at first. You should insert at least five decimal places for sufficient precision.

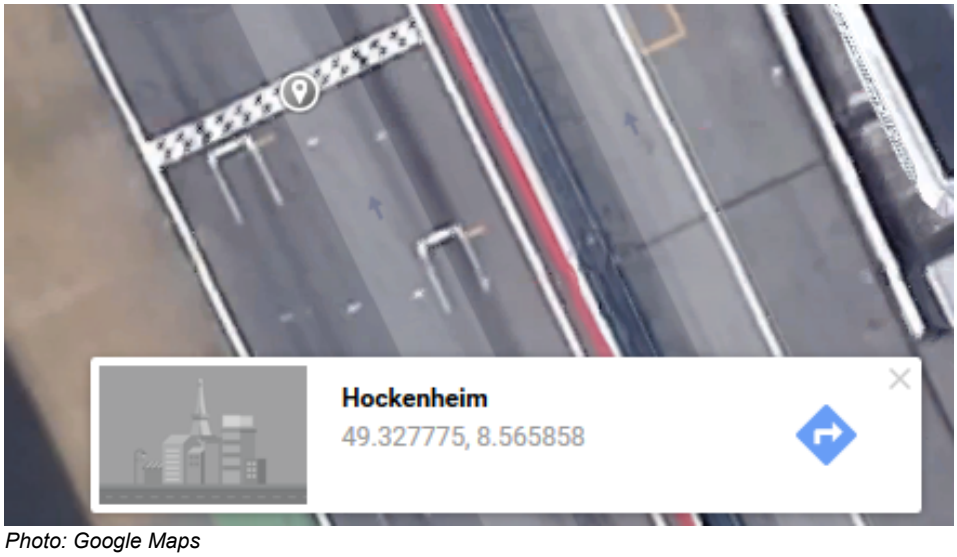


Photo: Google Maps

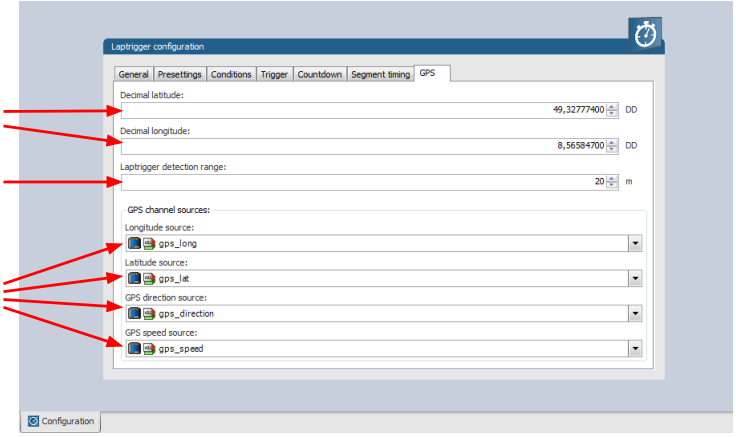
The detection range defines the radius of a circle around the detection point in which the lap trigger can be set. The lap trigger will be set as soon as the distance between the car and the detection point has reached its smallest peak. By this function an imaginary finishing line is calculated inside of the detection circle.

The imaginary finishing line can only be calculated if all channel sources are defined correctly. The latitude and longitude channel sources are mandatory for the functionality. Missing direction or speed source lowers the precision of the system.

Define the latitude and longitude of the GPS detection point.

Define the detection range around the detection point.

Define the channel sources for Longitude, Latitude, Direction and Speed.



Note

The configuration of the sensor update rate and the detection range must insure to receive a valid GPS point in the detection range, despite the occurring vehicle speed near the detection point.

16.1.3 Prevention of false triggers

- Race track topology and transmitter location frequently cause false triggers.
- Software functionality prevents acceptance of false triggers.
- Minimum vehicle speed for acceptance of trigger prevents false triggers while vehicle is stationary in the pits.
- Time based re-trigger protection prevents false triggers due to signal reflections on Home Straight.
- Lap distance based retrigger protection prevents false triggers due to track topology.

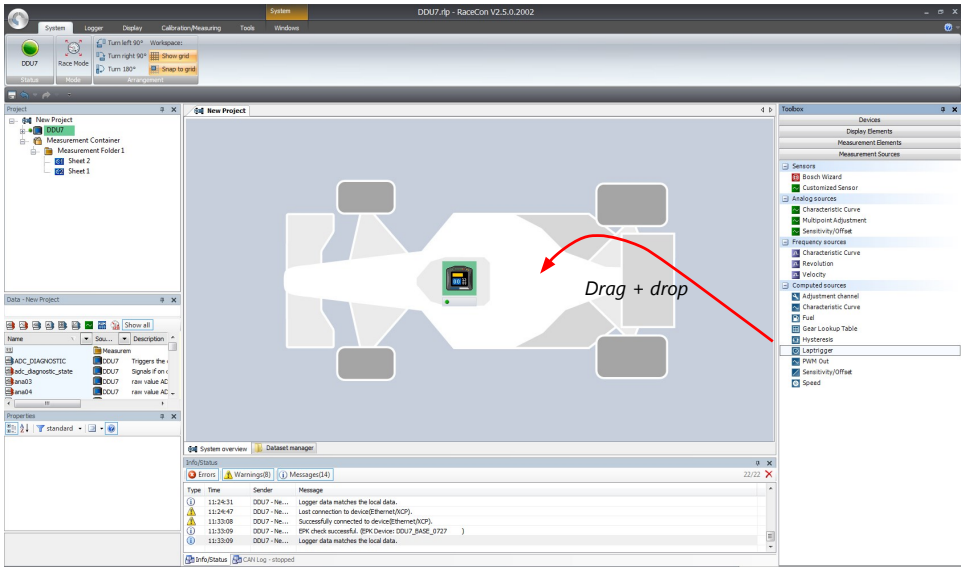
16.1.4 Forced triggers

Lap distance based insertion of ‘forced trigger‘.

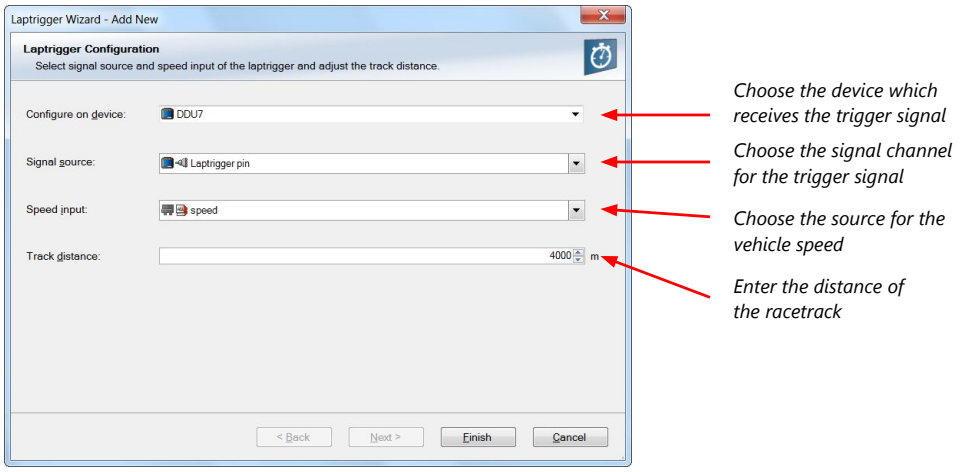
Under race conditions, trigger signals are sometimes missed. Software functionality introduces ‘forced trigger‘.

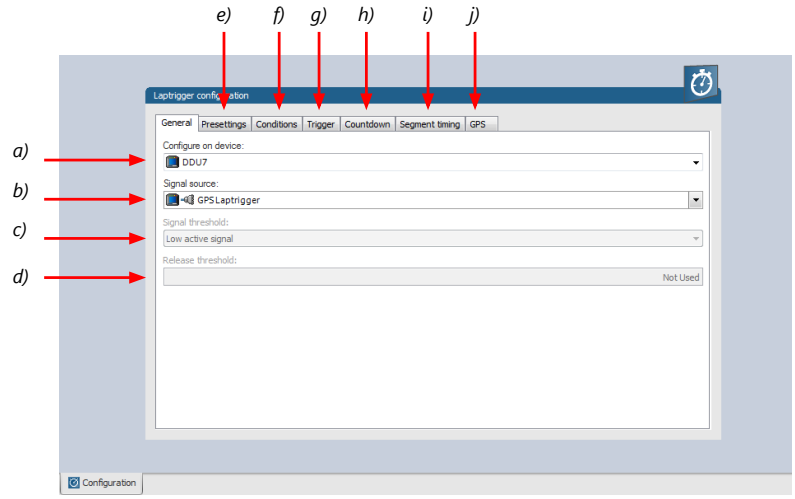
16.1.5 Setting up a lap trigger

1. Click ‘Measurement Sources‘ in Toolbox.
2. Drag ‘Laptrigger‘ into ‘System Overview‘. Do not drop it on ‘DDU 11‘!



A ‘Laptrigger Wizard‘ window opens.





a) Change signal device, if desired.

b) Change signal channel, if desired.

c) Choose signal threshold. See chapter 'Electrical trigger signal' for details.

d) Define threshold of input channel signal when trigger is released. Only possible, if no digital source is selected as signal source.

e) Define presettings for trigger. See chapter 'Lap trigger presettings' for details.

f) Define condition settings; change signal for vehicle speed, define speed settings. See chapter 'Distance based retrigger protection' and 'Distance based forced trigger' for details.

g) Define settings for main trigger. See chapter 'Lap timing' for details.

h) Define settings for counddown timer. See chapter 'Countdown timer' for details.

i) Define settings for sub trigger. See chapter 'Segment timing' for details.

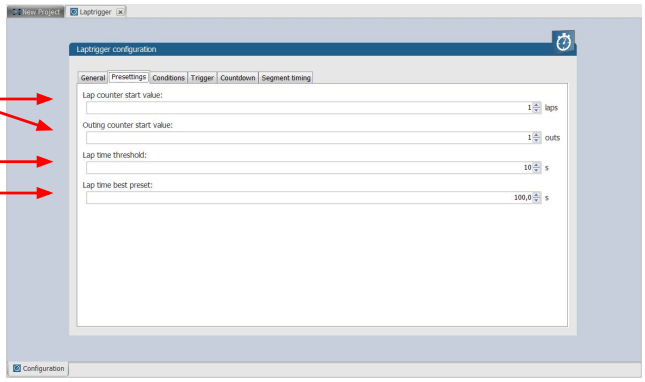
j) Define settings for a GPS lap trigger. See chapter 'GPS lap trigger' for details. Only applicable if the signal source is set to 'GPS lap trigger'.

Click ‘Finish’ to complete the operation. A pre-configured lap trigger window opens.

Preset values for lap counter and outing counter

Minimum laptime that a new 'best laptime' is accepted

Preset value for 'best laptime'

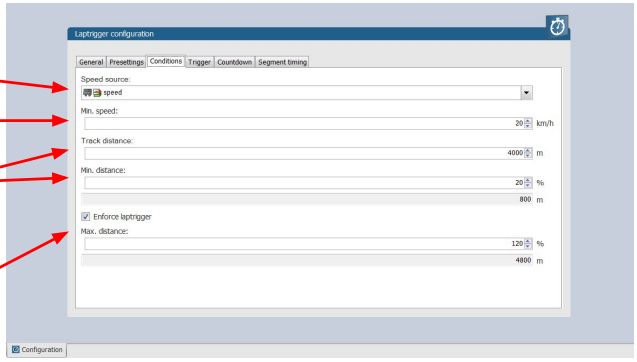


Change signal for vehicle speed, if desired.

Enter minimum speed for trigger release.

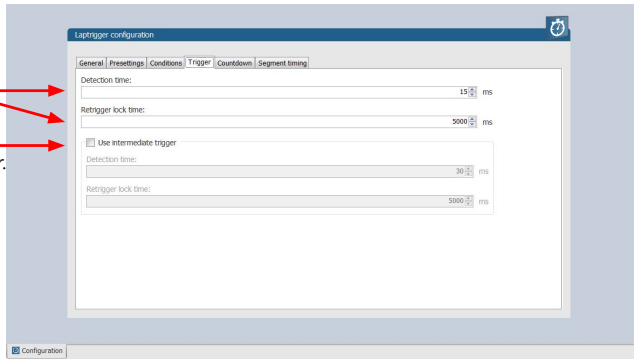
Define settings for distance based retrigger protection.

Define settings for distance based forced trigger.

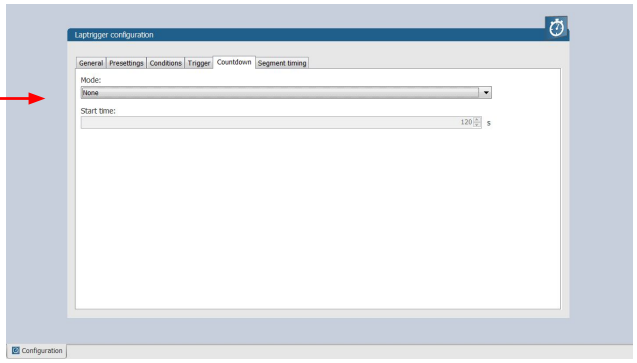


Define settings for lap timing (main trigger).

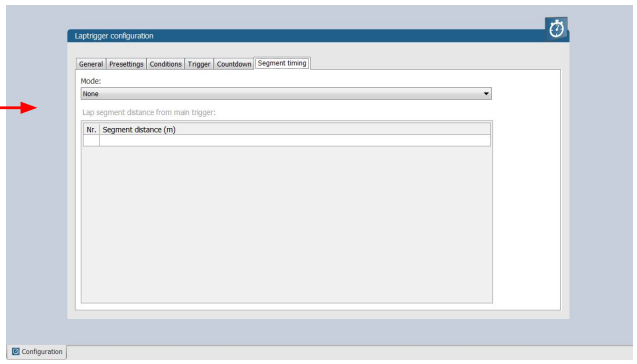
Define settings for sub trigger. Not applicable with a GPS lap trigger.



Define settings for countdown timer.



Define settings for segment timing.

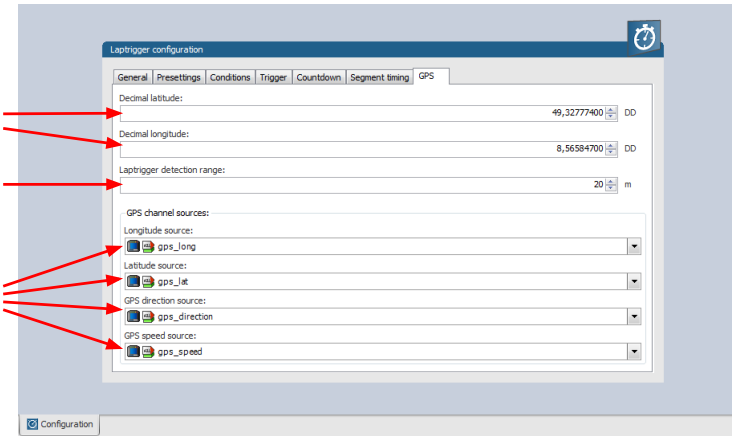


Only applicable for a GPS Laptrigger

Define the latitude and longitude of the GPS detection point.

Define the detection range around the detection point.

Define the channel sources for Longitude, Latitude, Direction and Speed.

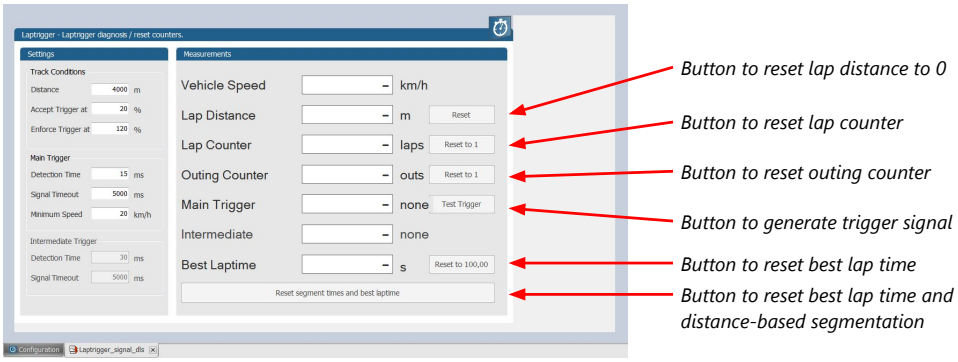


16.1.6 Lap trigger channel diagnosis/counter reset

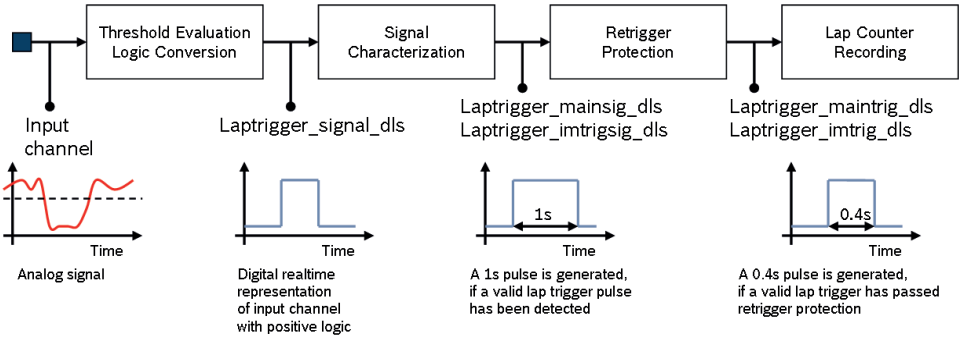
To display a quick lap trigger channel diagnosis and to reset counters use the diagnosis page in RaceCon. Any ‘Laptrigger_xxx’ channel can be displayed.

Double-click on any ‘Laptrigger_xxx’ channel in the Data Area. Example: ‘laptrigger_lap-dist_dls’

A diagnosis window opens in Main Area.

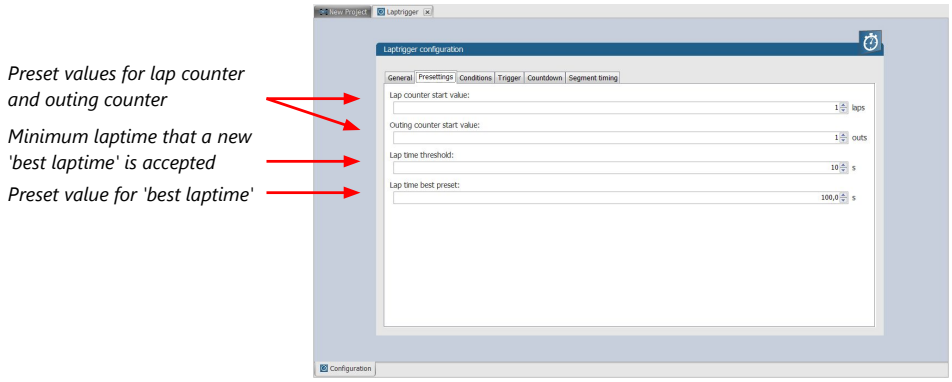


Lap trigger diagnosis scheme

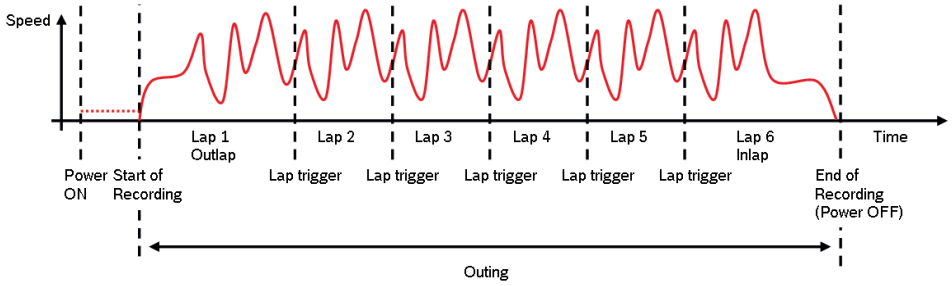


16.1.7 Lap trigger presettings

When the reset buttons on the diagnosis page are activated, these values are used.



16.2 Counting outing/laps/fragments



Functionality

– Power ON: system + measurement is initialized but not yet started

- Global start condition fulfilled: recording starts
- Reception of valid lap trigger: recording of lap completed, new lap starts
- Power OFF or Global start condition not fulfilled: recording of lap completed, system shutdown

The system is counting:

Outing:

- The outing counter is incremented with each power cycle when at least one valid lap (not by forced lap trigger) was recorded

Lap:

- Leaving the pits to lap trigger
- Lap trigger to lap trigger
- Enforced lap trigger (see Distance based forced trigger [▶ 154])

Fragment:

- Test bench operation
- Power cycle on track or box (e.g. engine stalled)
- File fragmentation size [MB], time [sec]

Channels for display

To display counters use the following channels:

Channel	Function
Laptrigger_outcnt_dls	Outing counter
Laptrigger_lapctr_dls	Lap counter
Fractr	Fragment counter

Counting in WinDarab

To automatically name recorded files use filename templates in WinDarab dialog:

Filename template	Function
[outing]	Value of outing counter
[lap]	Value of lap counter
[fragment]	Value of fragment counter

[###03] indicates: 'always use 3 digits with leading zeros'.

16.3 Lap timing

There are different possibilities to adjust the lap trigger to the timing situation.

The detection time defines the minimum time the input signal changes its state. E.g. a low active signal needs to be below the threshold for min. 15 ms to release the trigger.

Channels for display

To display lap times use the following channels:

Channel	Function
Laptrigger_lapctr_dls	Number of completed laps

Channel	Function
Laptrigger_laptime_dls	Running laptime
Laptrigger_laptime_best_dls	Laptime of best lap
Laptrigger_laptimeold_dls	Laptime of last lap completed
Laptrigger_laptimeseg_dls	Segment time of last segment
Laptrigger_lapctr_dls	Number of completed laps

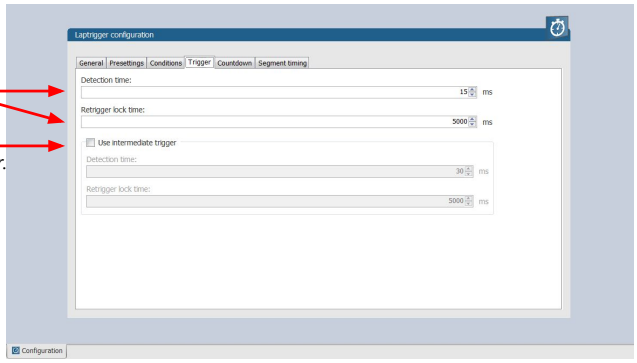
16.3.1 Time based retrigger protection

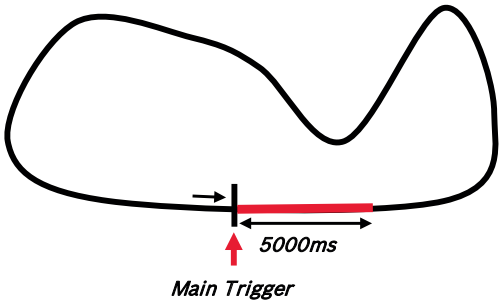
Trigger is locked for 5 s after main trigger was received.

To deactivate time based retrigger protection, set 'Retrigger lock time' to 0 ms.

Define settings for lap timing (main trigger).

Define settings for sub trigger. Not applicable with a GPS lap trigger.





16.3.2 Distance based retrigger protection

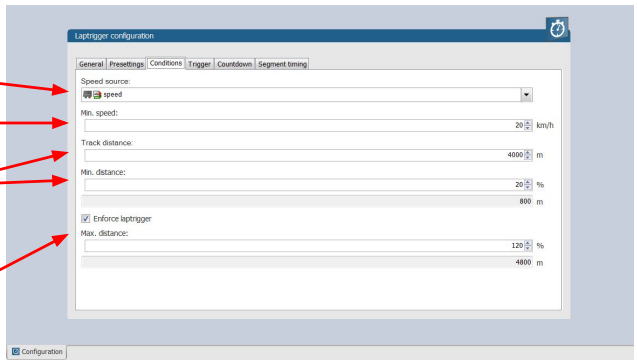
Trigger is locked until configured min distance (i.e. 80 % → 3200 m) of track distance (i.e. 4000 m) has been covered. To deactivate distance based retrigger protection, set min distance to 0 %.

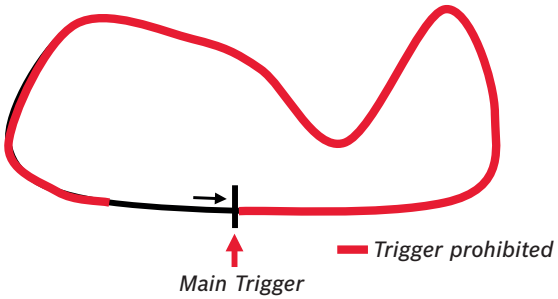
Change signal for vehicle speed, if desired.

Enter minimum speed for trigger release.

Define settings for distance based retrigger protection.

Define settings for distance based forced trigger.





16.3.3 Distance based forced trigger

After a missed main trigger, a forced trigger is inserted, if the configured max. distance (i.e. 120 % → 4800 m) of the track distance (i.e. 4000 m) has been reached. In this case, the channel 'Laptrigger_distlap_dls' starts at the delta between the max. distance and the track distance (i.e. 800 m).

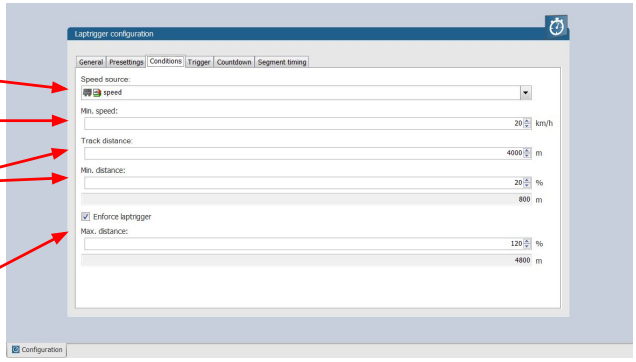
To deactivate distance based forced triggers, uncheck box.

Change signal for vehicle speed, if desired.

Enter minimum speed for trigger release.

Define settings for distance based retrigger protection.

Define settings for distance based forced trigger.



16.4 Segment timing

Segment timing is the calculation of elapsed time for parts of laps (segments).

Segments are defined:

- based on sub-trigger signals (additional transmitters)
- based on distance travelled

Times for segments are compared to:

- Last lap completed
- Fastest lap

Channels for display

To display segment times use the following channels:

Channel	Function
Laptrigger_lapdiff	Time difference between finished lap and last lap
Laptrigger_lapdiffb	Time difference between finished lap and best lap
Laptrigger_lapseg_dlast	Difference of lap segment time compared to last lap
Laptrigger_lapseg_dbest	Difference of lap segment time compared to best lap

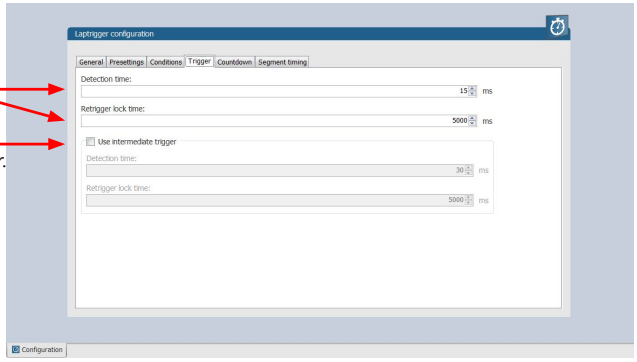
16.4.1 Sub trigger mode

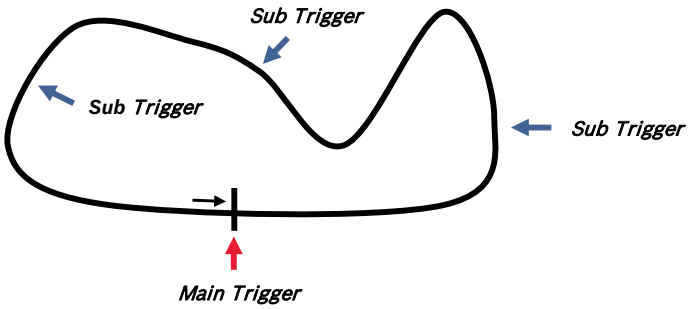
Using main trigger (20 ms pulse) at Start-Finish-Line. 3 sub triggers (40 ms pulse) positioned at 1,000 m, 2,000 m and 3,000 m.

To deactivate sub trigger mode uncheck box.

Define settings for lap timing (main trigger).

Define settings for sub trigger. Not applicable with a GPS lap trigger.





The sub trigger mode cannot be used with the GPS lap trigger.

16.4.2 Distance mode

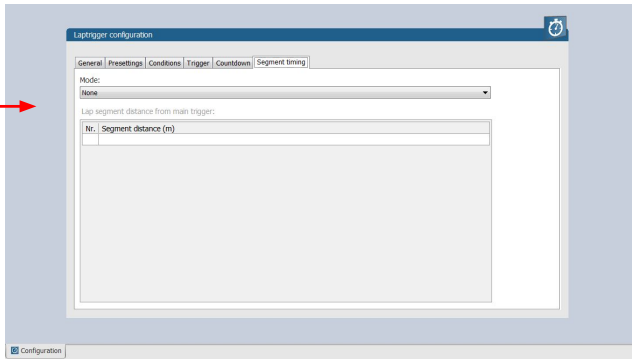
Using main trigger (20 ms pulse) at Start-Finish-Line.

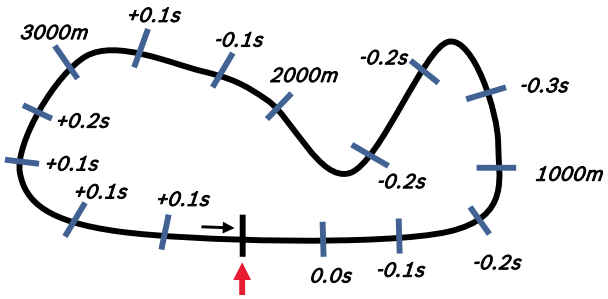
Set 'Mode' to 'Distance' and enter desired segment distances.

Segment time is automatically calculated at each segment. Time difference to last lap and fastest lap is automatically calculated at each segment.

To deactivate distance mode set 'Mode' to 'None'.

Define settings for segment timing.



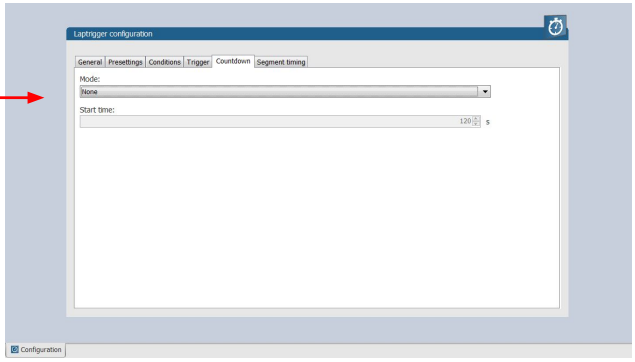


16.5 Countdown timer

Some race classes require a minimum time spent in the pits. An additional lap trigger Tx is configured as a segment trigger positioned at pit entry. The trigger signal starts a timer countdown.

The current value of the timer is stored in the variable **Laptrigger_cntdown_dls** which can be displayed.

Define settings for countdown timer.

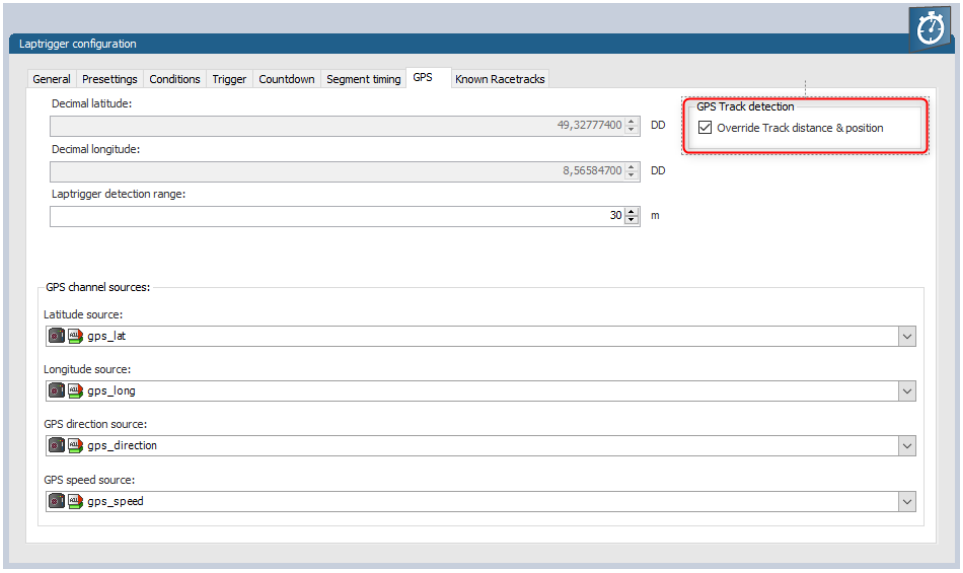


16.6 Automatic GPS Track Detection

With the GPS lap trigger, an automatic track detection can be activated by checking the highlighted box in the lap trigger configuration at the GPS tab.

Activating this functionality will overwrite the GPS trigger point coordinates and the track length with the values of a detected racetrack. The coordinates and the track length will be grayed out in the tool.

The function will compare the current GPS position with the coordinates of the known Racetracks list and use the closest GPS trigger point. If there are track variants with different track length, the system will adapt itself to the correct variant, if it is in the known racetrack list, after ~three detected laps. Please note that the track length needs to be quite accurate, within +/- 100 m, to adapt itself to another variant.

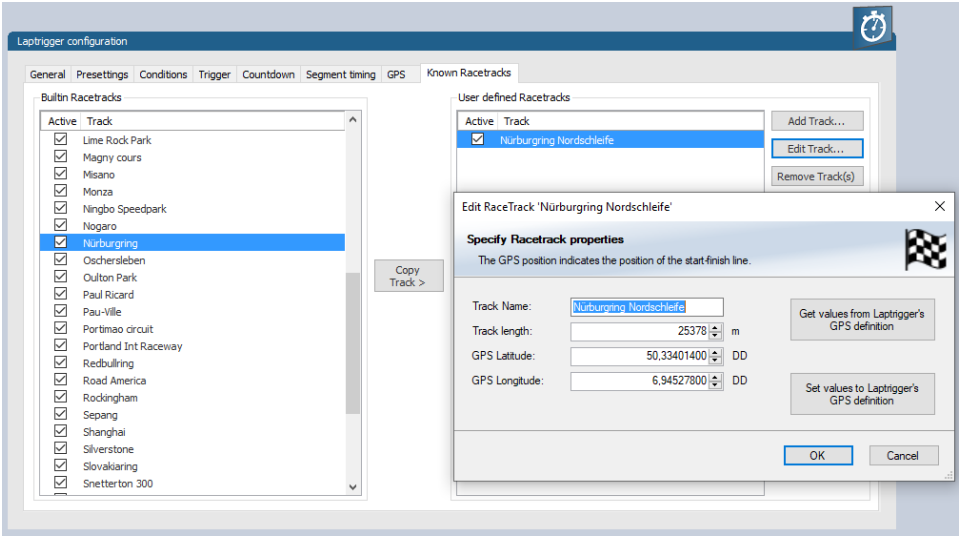


The known Racetrack list can be found in the lap trigger configuration menu within the Known Racetracks tab. It contains a built-in list and a user-defined list. Each track can be activated or deactivated with the checkbox to manually set the variant if needed.

User defined tracks can either be added from scratch with the Add Track button or with the Copy Track button as a modified version of a built in track. Both buttons will open the same Edit Race Track menu.

In the menu a track name, length and the coordinates of the detection point is required. The coordinates can be pulled from the GPS tab with the button “Get values …” or sent to the GPS tab with the button “Set values …”. This allows an easy interaction with the manual GPS lap trigger mode.

The user-defined tracks will be part of the project. If the tracks are required in another project, the lap trigger module can be ex-/imported into another project.



Following signals are assigned to the function:

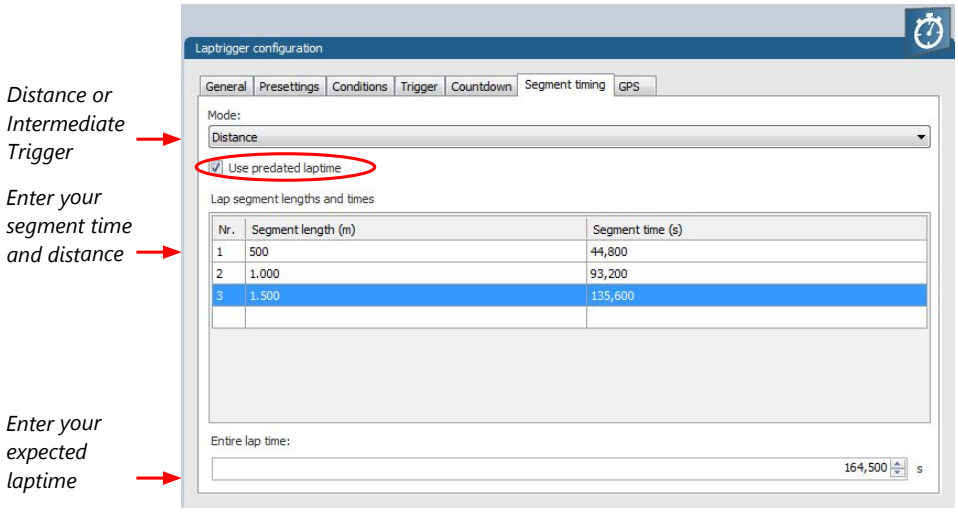
Laptrigger_trackdet_id	Signal will show the track name from the Racetracks list as an enumeration or can show the ID number as raw value. The enumerated name will also be visible in the logging.
Laptrigger_trackdet_laplen	Track length of the used track variant
Laptrigger_trackdet_lat	Latitude GPS coordinate of the used GPS trigger point
Laptrigger_trackdet_long	Longitude GPS coordinate of the used GPS trigger point

16.7 Predated Laptime

The predated laptime function allows to compare the current lap- and segment time with the predated time of an expected lap. Additional the function can estimate the laptime of the current lap. This functionality is integrated in the laptrigger module in RaceCon.

16.7.1 Setting up the predated laptime

To use the predated laptime function you need to set up a laptrigger as described in the chapter Lap Trigger [▶ 145]. Under the ribbon “Segment timing” you need to choose your segmentation mode which can either be distance or intermediate trigger based.



For the distance mode, you need to check on an old lap or estimate how long it takes to travel the segment distance. Please enter those values into input field. The values can also be copied and pasted to the input field from an excel sheet as a normal text. In the intermediate trigger, you just need to set the expected time the driver takes to reach the segment trigger.

Note

Please note that the segment time and length is always measured from the start line or where the main lap trigger is set.

16.7.2 Functionality and channel outputs

Following output channels are generated by the predated laptime function.

Laptrigger_lapdiff_pred_dls	Laptime difference between the predated and the last laptime
Laptrigger_lapsegdiffpred_dls	lagseg difference between the last segment and the predated segment.
Laptrigger_Lapcurrpred_dls	Estimated laptime of the current lap, based on the predated laptime and the predated segment deviations

The channel Laptrigger_lapdiff_pred_dls is updated as soon as the main lap trigger is received. Both other channels are updated as soon as the next segment distance is travelled or the next intermediate trigger is received.

17 GPS Sensor

17.1 GPS (Global Positioning System)

Space-based global navigation satellite system.

GPS provides positioning, navigation, and timing services to worldwide users.

GPS receiver (sensor) gives digital information about position (longitude, latitude, height), ground speed, course, and status.

Two types of GPS receivers:

CAN output -> Read in messages via CAN Input of DDU 11 (not covered here)

Serial output -> Read in messages via RS232 Interface of DDU 11

Serial Interface is characterized by:

Voltage levels: RS232 is standard (+/-12 V), UART (0 V/ 5 V) needs level shifter

Baud rate: 9,600 is standard for GPS, DDU 11 supports 1,200 to 115,200 baud. GPS Rx interface baud rate must match DDU 11 interface baud rate. DDU 11 Baud rate can be set with the 'GPS_BAUDRATE' characteristic Data format: DDU 11 expects 8 data bits, no parity bit, 1 stop bit (8N1)

17.1.1 Serial interface characterization

Serial Interface is characterized by:

Voltage levels: RS232 is standard (+/-12 V), UART (0 V/ 5 V) needs level shifter

Baud rate: 9,600 is standard for GPS, DDU 11 supports 1,200 to 115,200 baud. GPS Rx interface baud rate must match DDU 11 interface baud rate. DDU 11 Baud rate can be set with the 'GPS_BAUDRATE' characteristic Data format: DDU 11 expects 8 data bits, no parity bit, 1 stop bit (8N1)

17.2 Protocol

DDU 11 expects NMEA Protocol (ASCII).

The following messages are decoded:

Message	Function
GGA	GPS fix information
GSA	Overall satellite data
GSV	Detailed satellite data
RMC	Recommended minimum data for GPS
VTG	Vector track and speed over the ground

On most GPS sensors, these messages are activated in the default configuration.

17.3 Sensor recommendation

The system has been tested with the Navilock NL-8004P MD6 Serial PPS Multi GNSS Receiver. This sensor is based on a U-Blox 8 chipset and is fully configurable with the Navilock "U-Center" software. To use this sensor with Bosch Motorsport components the transfer rate, the satellite system and the update rate need to be reconfigured. More information about the configuration can be found in the Appendix.

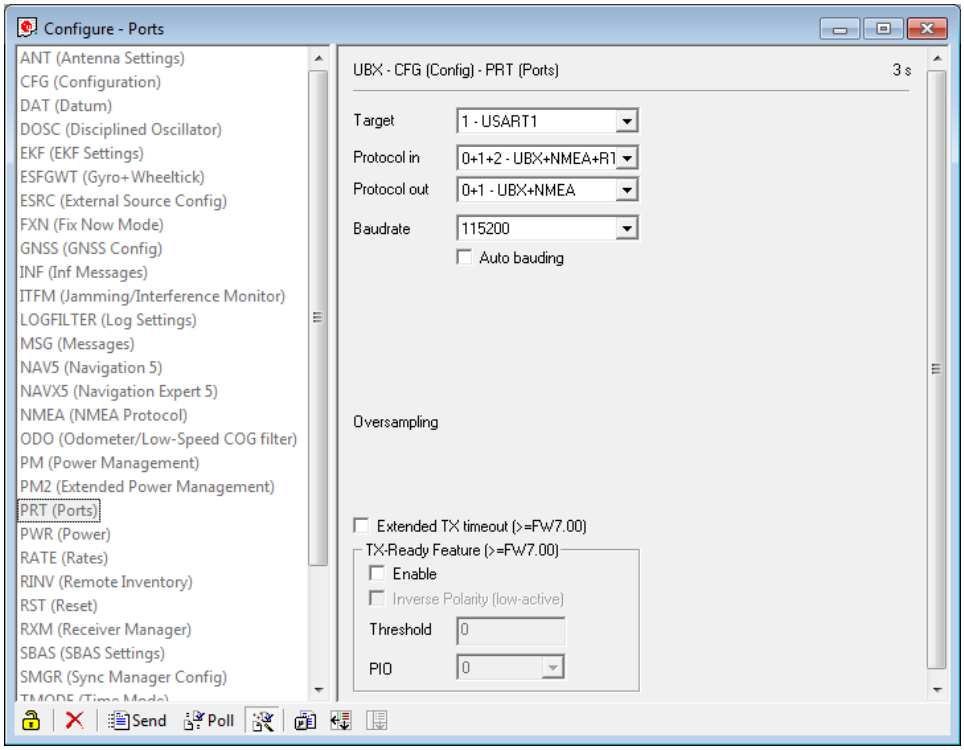
17.3.1 Configuration of the recommended Navilock NL-8004P MD6 Serial PPS Multi GNSS Receiver

For the sensor configuration, the sensor needs to be connected to the Navilock software "U-Center" which is available from Navilock free of charge. Navilock offers a USB connection cable for the sensor.

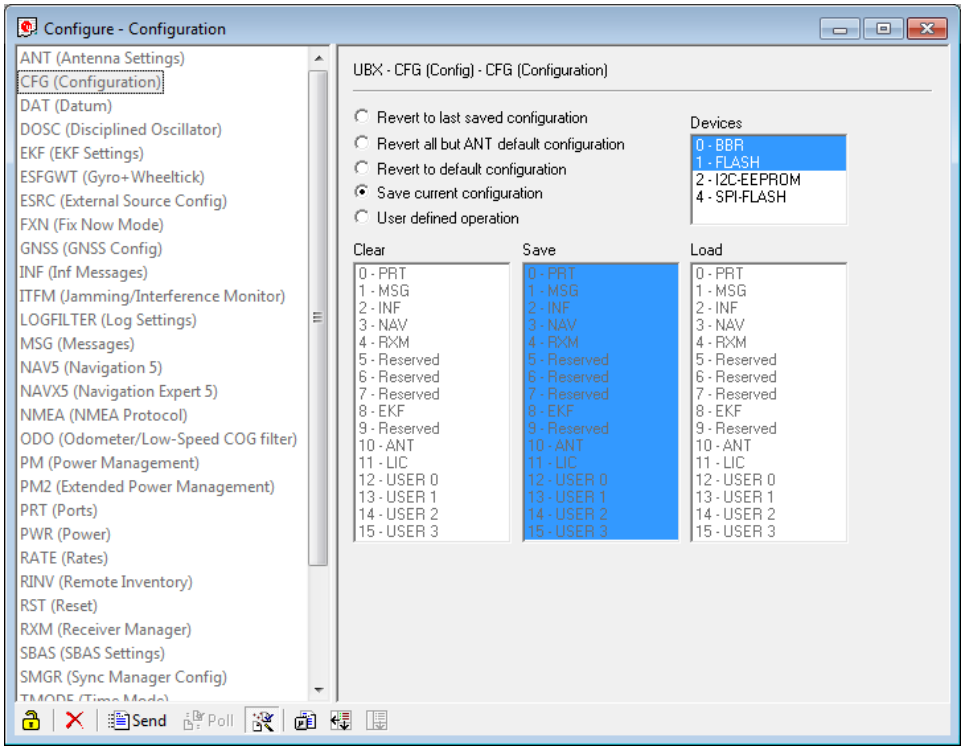
In "U-Center" click **"View"** – **"Configuration View"** to start the configuration. The following 3 points have to be changed:

Transfer Rate

- Click on "**PRT (Ports)".**
- Change the baud rate to a fixed value, this value needs to meet the setting of RaceCon. For a good signal quality we recommend 115,200 baud.
- Click on **"Send"** to store the new setting in "U-Center".

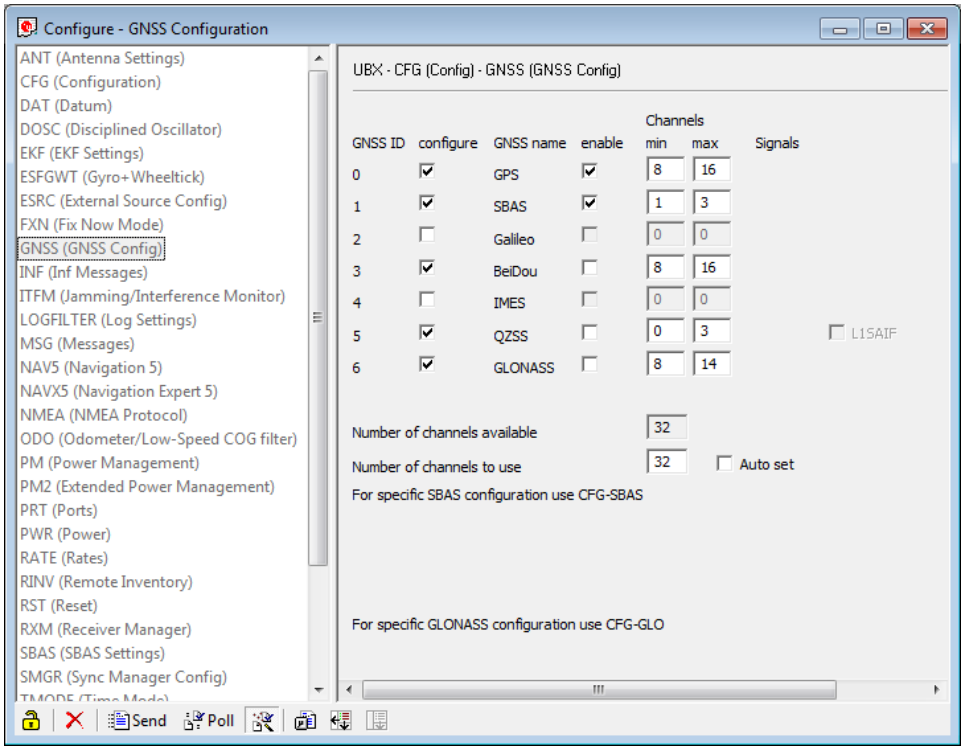


- Click on **"CFG (Configuration)".**
- Click on **"Send"** to save the new setting on the sensor.



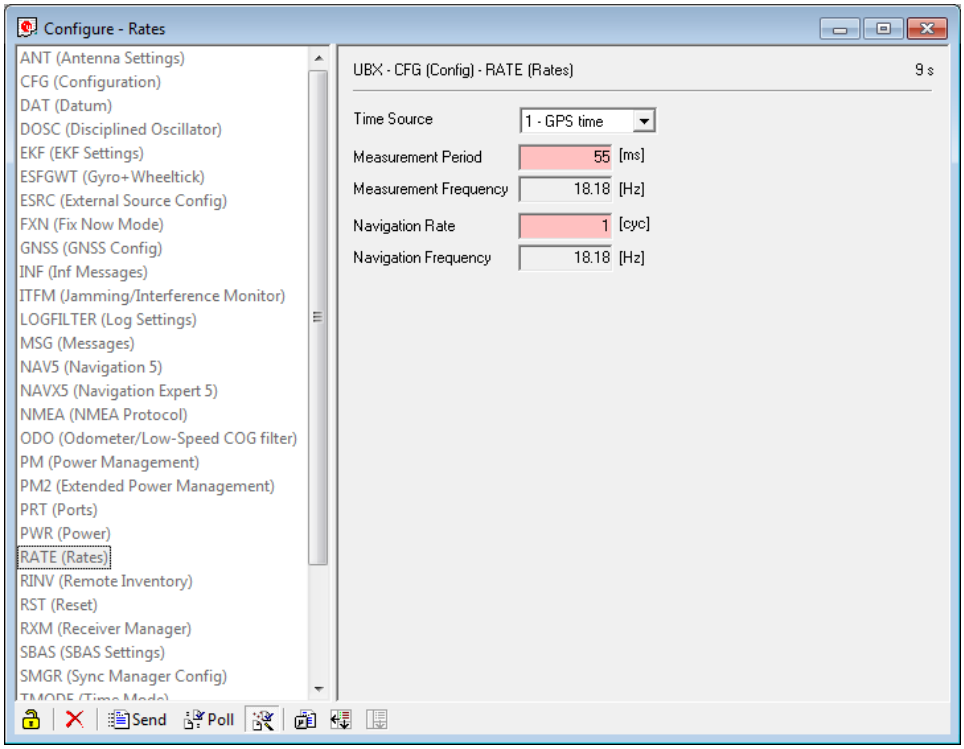
Satellite System

- Click on **“GNSS (GNSS Config)”.**
- Set the ticks as shown in the following picture.
- Click on “**Send**” to store the new setting in “U-Center”.
- As during configuration step 1, click on “**CFG (Configuration)**”.
- Click on“**Send**” to save the new setting on the sensor.



Update Rate

- Click on **“RATE (Rates)”.**
- Change the “Measurement Period” to 55 ms.
- Change the “Navigation Rate” to 1 cyc.
- Values which lead to a lower frequency will lower the precision of the sensor, we recommend the mentioned values.
- Click on “**Send**” to store the new setting in “U-Center”.
- As during configuration step 1, click on “**CFG (Configuration)**”.
- Click on“**Send**” to save the new setting on the sensor.



Note

Sensor needs reception for visible signal.
It takes time to start the sensor.

17.4 Measurement labels

The decoded NMEA messages are copied to these DDU 11 measurement labels.

Measurement label	Function
gps_PDOP	Position Dilution Of Precision
gps_HDOP	Horizontal Dilution Of Precision
gps_VDOP	Vertical Dilution Of Precision
gps_lat	Latitude +/- [degree]
gps_long	Longitude +/- [degree]
gps_elv	Antenna altitude above/below mean sea level (geoid) in meters
gps_speed	Speed over the ground in kilometers/hour
gps_direction	Track angle in degrees

Measurement label	Function
gps_declination	Magnetic variation degrees (Easterly var. subtracts from true course)
gps_year	Years since 1900
gps_mon	Months since January - [0,11]
gps_day	Day of the month - [1,31]
gps_hour	Hours since midnight - [0,23]
gps_min	Minutes after the hour - [0,59]
gps_sec	Seconds after the minute - [0,59]
gps_hsec	Hundredth part of second - [0,99]
gps_smask	Bit mask over received NMEA sentences (Bit 0 = GGA, Bit 1 = GSA, Bit 2 =GSV, Bit 3 = RMC, Bit 4 =VTG) within last second.
gps_sig	GPS quality indicator (0 = Invalid; 1 = Fix; 2 = Differential, 3 = Sensitive)
gps_fix	Operating mode, used for navigation (1 = Fix not available; 2 = 2D; 3 = 3D)

These measurement labels are arrays, where the indexed element points to the same satellite.

(E.g. gps_info_satsigstrength[3] tells the receiving signal strength of satellite 3. Satellite 3 has the SAT-ID given in gps_info_satid[3])

Measurement label	Function
gps_info_satid[ ]	Satellite PRN number
gps_info_satinuse[ ]	Used in position fix
gps_info_satelevation[ ]	Elevation in degrees, 90 maximum
gps_info_satazimuth[ ]	Azimuth, degrees from true north, 000 to 359
gps_info_satsigstrength[ ]	Signal, 00-99 dB

17.5 GPS troubleshooting

Electrical

Is the transmitter signal of the GPS sensor connected to the receiver pin of serial interface of the DDU 11?

Is the GPS sensor powered up?

Does the GPS sensor deliver RS232 signal levels?

Is the sensor connected to the „sensor ground“ of the device?

Interface

Do the baud rates of the GPS sensor and the DDU 11 match?

Is the GPS sensor set up for 8N1 transmission parameters?

Is the GPS sensor set up for NMEA messages?

Are the GGA, VTG, RMC messages activated?

GPS sensor start-up

Does the GPS sensor ‘view’ the sky?

Did the GPS sensor complete its initial start-up procedure? This may take up to 20 min.

A correct reception is indicated when ‘gps_fix’ is showing ‘3D Fix’.

GPS sensor values are frozen

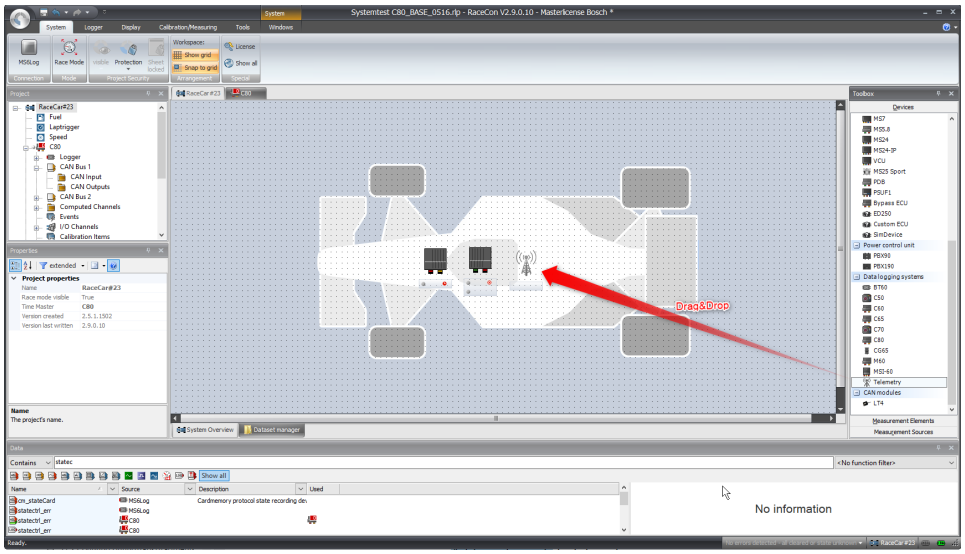
Does the sensor has lost its reception? The old values will be kept if the reception is lost. The gps_smask channel shows which NMEA sentence is received.

18 Telemetry System LTE 65

- Support for long-range online telemetry
- Individual programmable team code
- Fast block – slow block mechanism
- Programmable data rate
- Ethernet or RS232 interface
- Full online track coverage on almost all tracks

18.1 Software setup

Drop Telemetry from Toolbox into system overview.



Adding channels to telemetry

1. Expand the list of 'Loggers' by clicking on '+' in the DDU 11 Project Tree.
2. Double-click on 'Recording' in DDU 11 Project Tree.

   The recording configuration is displayed in the Main Area.
3. Click 'Edit channel(s)'.

   The 'Edit Recording Channels' window appears.
4. Choose between 'Fast/Slow block' transmission.

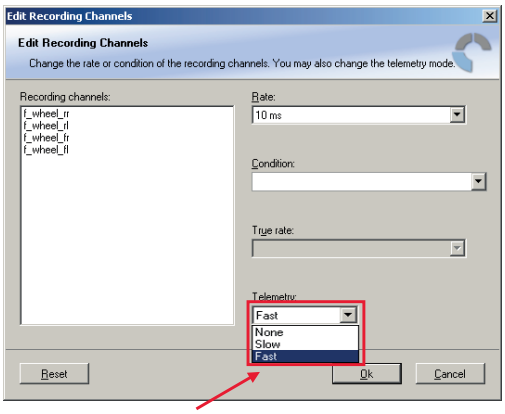
Using fast block/slow block transmission

DDU 11 telemetry has a bandwidth 200 kBit/s, the used bandwith can be adjusted to cope with the transmitting system. The bandwidth has to be divided into channel information to be transmitted high-frequently and low-frequently using the 'fast/ slow block' setting.

Channels are grouped into 8 blocks which are transferred each cycle:

- Fast block (Block 1) is transferred every cycle and used for a high-frequent transmission of channel information (e.g. speed, rpm).
- Slow blocks (Block 2...n) are transferred every n-th cycle and used for a low-frequent transmission of channel information (e.g. tire pressure, oil temperature).

Transmission Scheme



None – channel(s) are not transferred
Slow – channel(s) are transferred in the slow telemetry block
Fast – channel(s) are transferred in the fast telemetry block

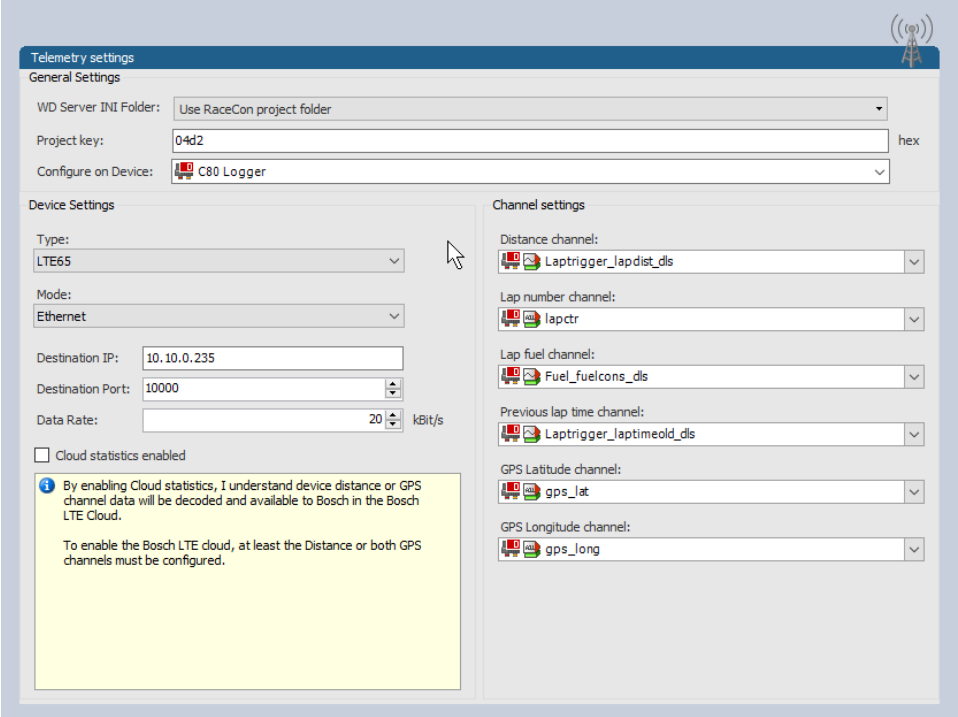
If the maximum bandwidth of a block is reached, a warning will be displayed. To fix this problem you can view the allocation of the channels and data rate in the 'Statistics' tab of the Main Area.

18.2 Telemetry channels with special functionality

The Telemetry system allows the transmission of special information such as running distance of current lap, lap number of current lap and lap time, fuel consumption of last lap completed. You have to assign the channel type to the telemetry channel so that it is recognized accurately by RaceCon.

Channel's names are e.g.: Laptrigger_lapdist_dls, fuelcons, lapctr, Laptrigger_lapdist_dls. Different channel names are possible between different devices (e.g. ECU MS6, laptrigger module used in RaceCon).

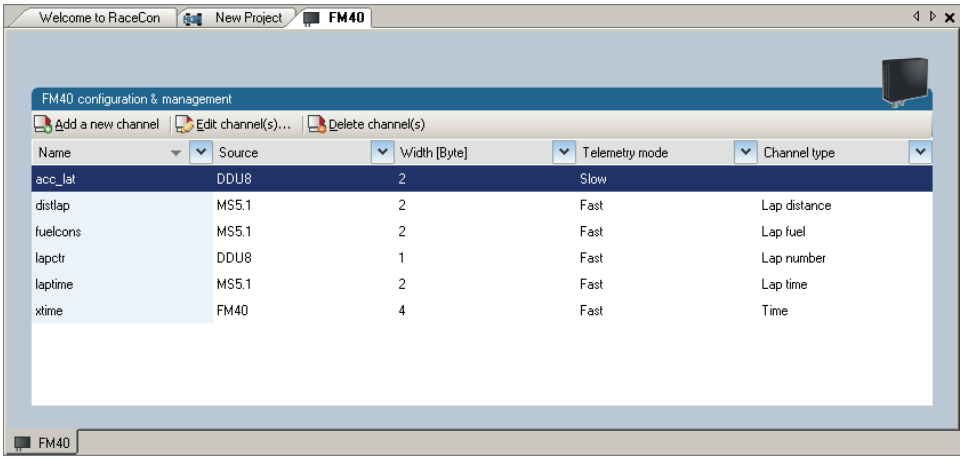
For displaying the position of the car in the cloud, additionally GPS-position and lapdist can be send to the cloud, this is activated with the checkbox "cloud statistics".



1. Assign the desired channels to the channel types. The table below shows the function of the available channel types.
2. Click 'Ok' when done.

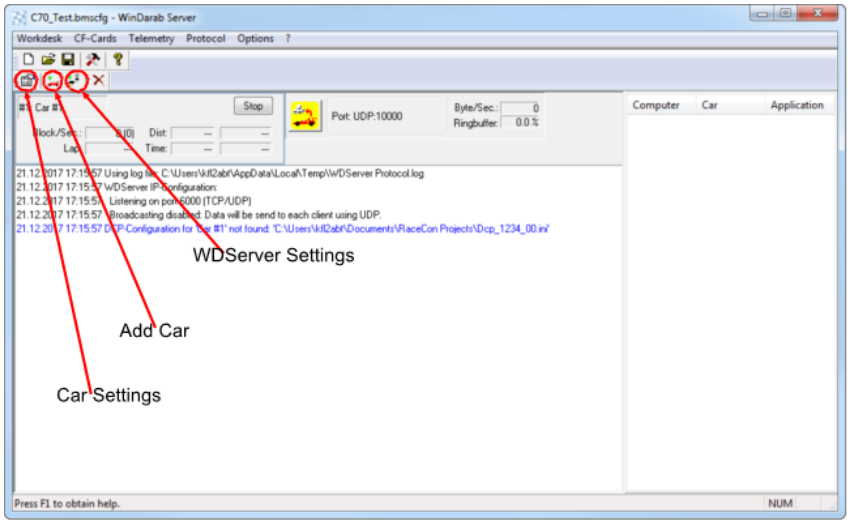
Measurement channel	Function
Distance	Running distance of current lap
Lap number	Lap number of current lap
Lap fuel	Fuel consumption of last lap completed
Lap time	Exact lap time of lap completed

The telemetry channels and their assigned channel types are displayed in the overview list.

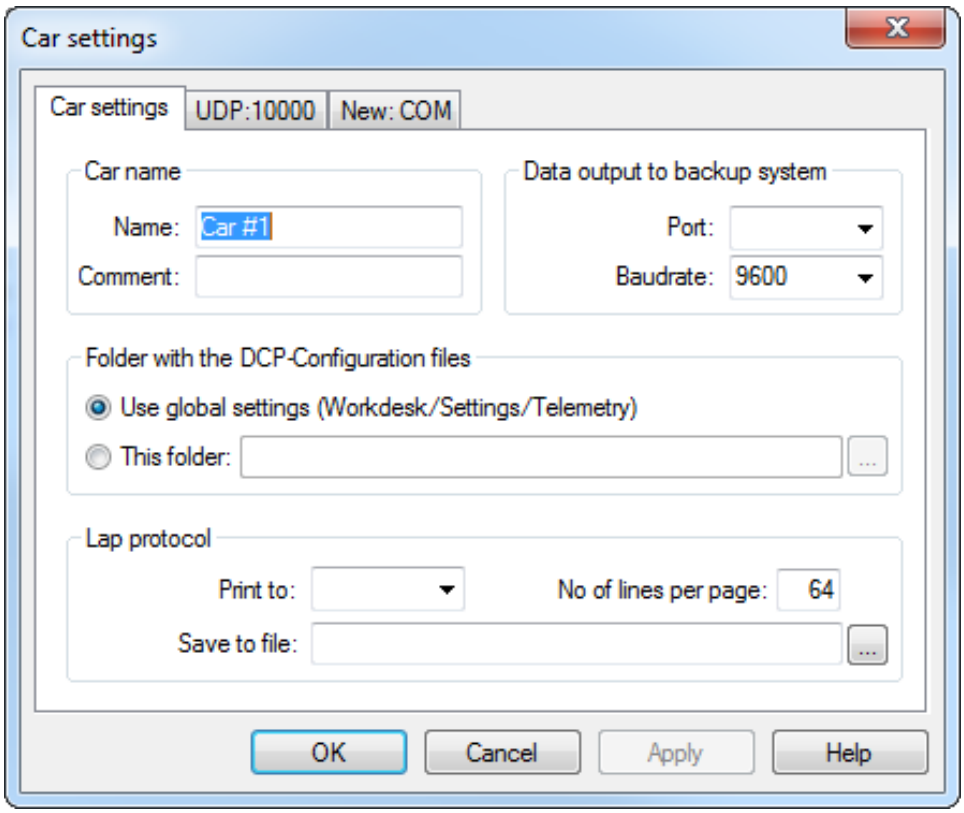


18.3 Setting up car in WDServer

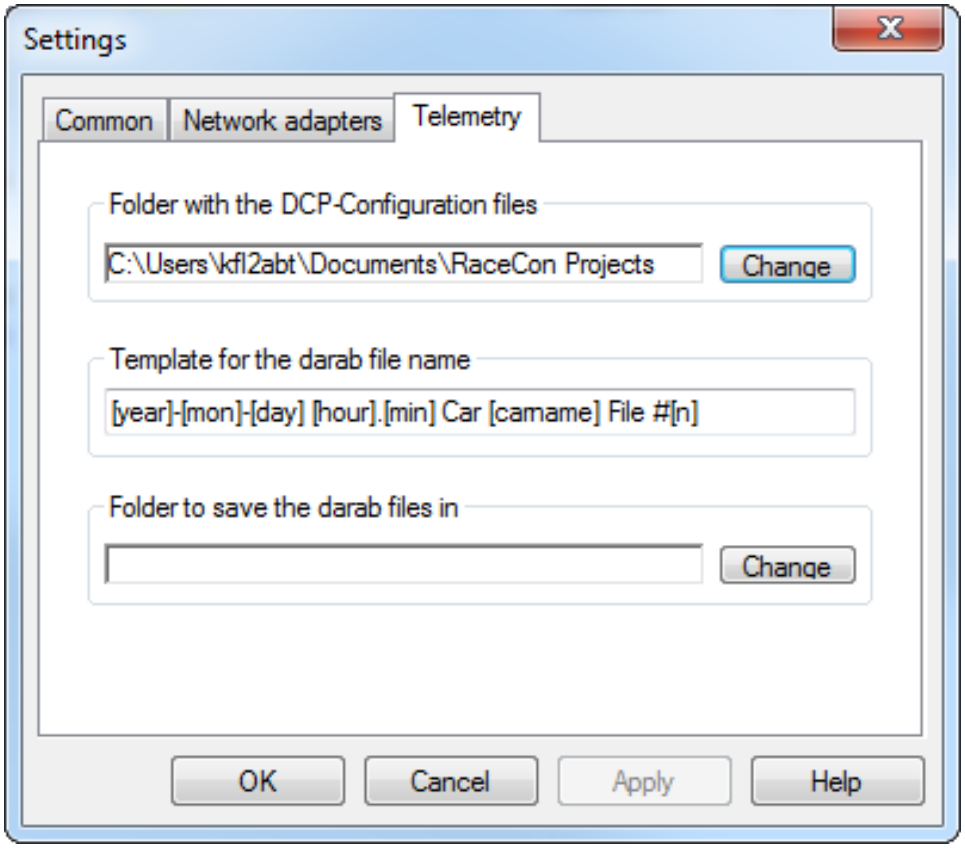
WDServer is a program used to capture data streaming from a transmitter and convert to WinDarab; WDServer also creates a log of the data received over telemetry.



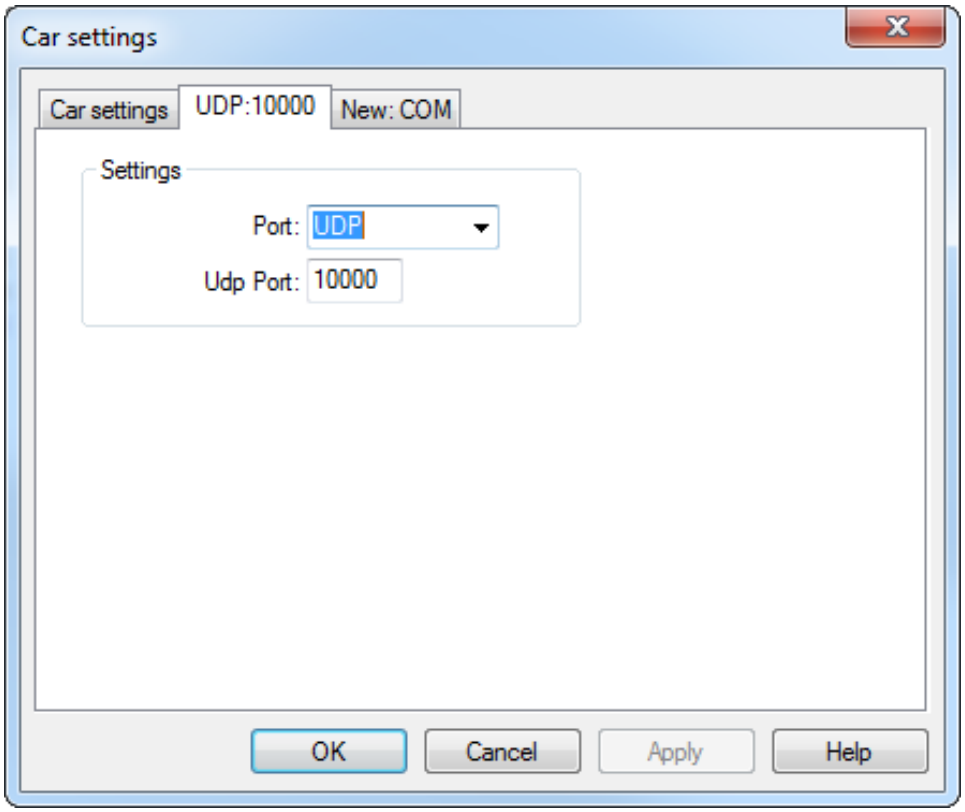
1. To set up a new car, select ‘Add Car’.
2. In the Car settings tab, enter a name for the new vehicle.
   This name will be used as a part of the file name for WDServer’s log of received telemetry data and will show up in WinDarab, when searching for the telemetry stream in the Network folder.



3. You are now at the final step of configuring the telemetry stream. In order for the data to be decrypted by WDServer, two *.ini files must be referenced by WDServer. After the configuration is sent to the logger, these two different *.ini files will be created in the base folder. You can find the base WDServer folder, if you right-click the Telemetry and select ‘Properties’. You can change this folder location for easier access if desired.
4. Define the link to the folder of the *.ini files for each car or define it in the general WDServer settings, under the ‘Telemetry’ tab.

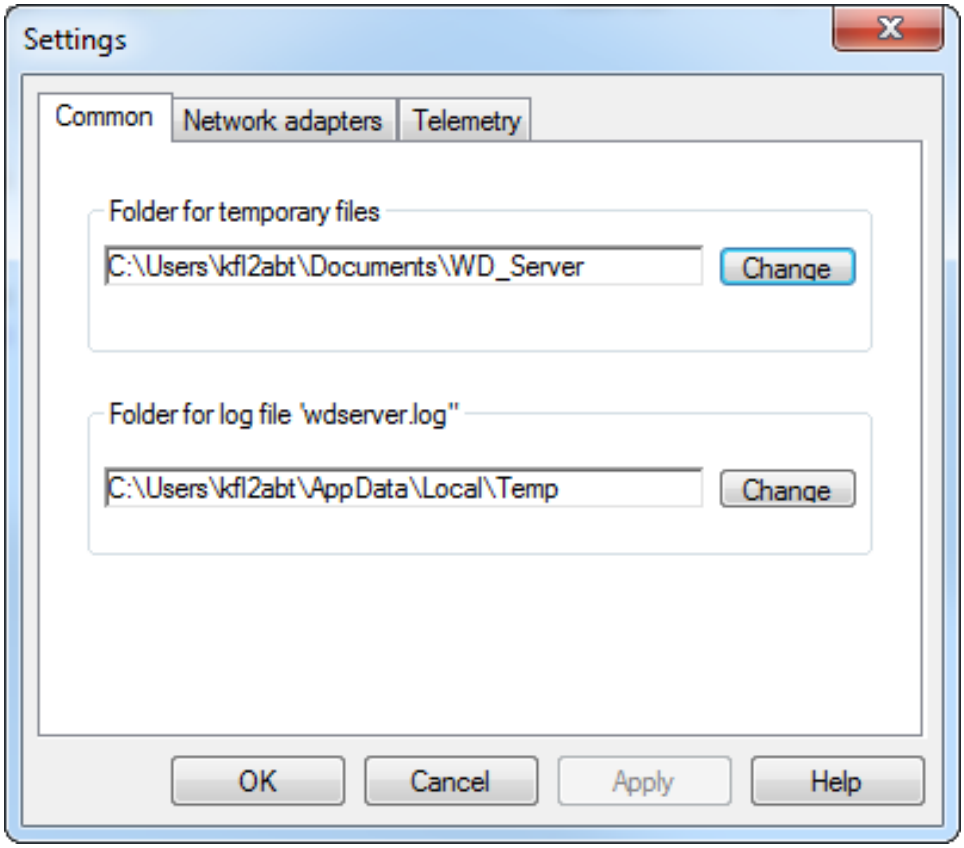


5. Under the 'UDP' tab, select the drop-down menu and type in "UDP".
6. For the UDP Port, type in the port number assigned to the device in RaceCon.
   Each vehicle being read by a single receiver device must have a unique port number.
   This information will be provided by Bosch upon delivery of the devices.



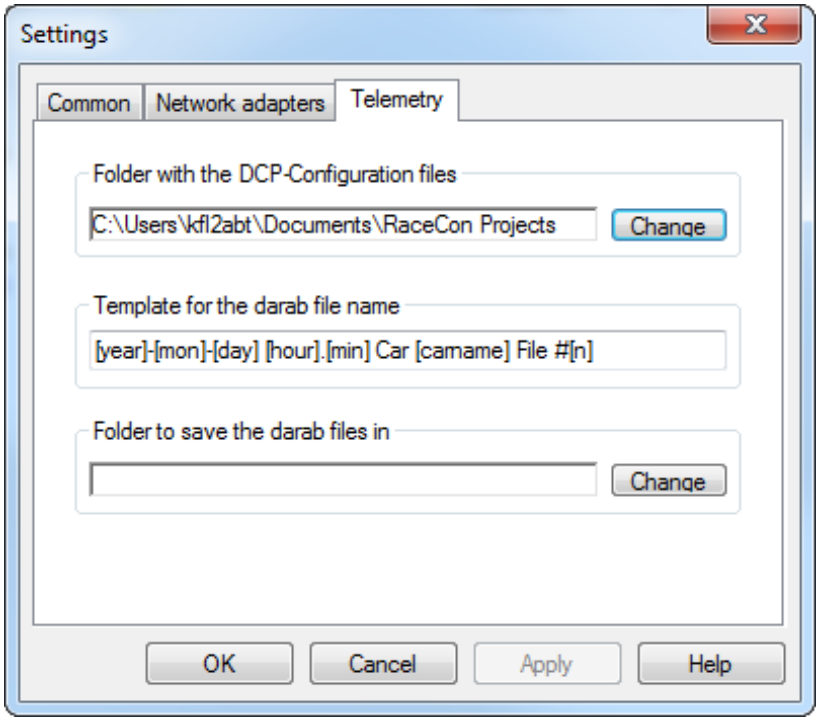
7. Click 'OK', to close the window.

8. Select the button ‘WDServer Settings’.
9. Under the ‘Common’ tab, choose directories where WDServer can store its temporary files and log files. These are created during telemetry reception and can be used to help diagnose issues.



Under the ‘Telemetry’ tab, the first section requests a folder path for the DCP- Configuration files. This is the folder path where RaceCon stored the *.ini files required by WD-Server.

1. Click on the “Change” button next to this section and navigate to this folder.
   A template can also be specified for the file nomenclature for logged telemetry as well as a save location.

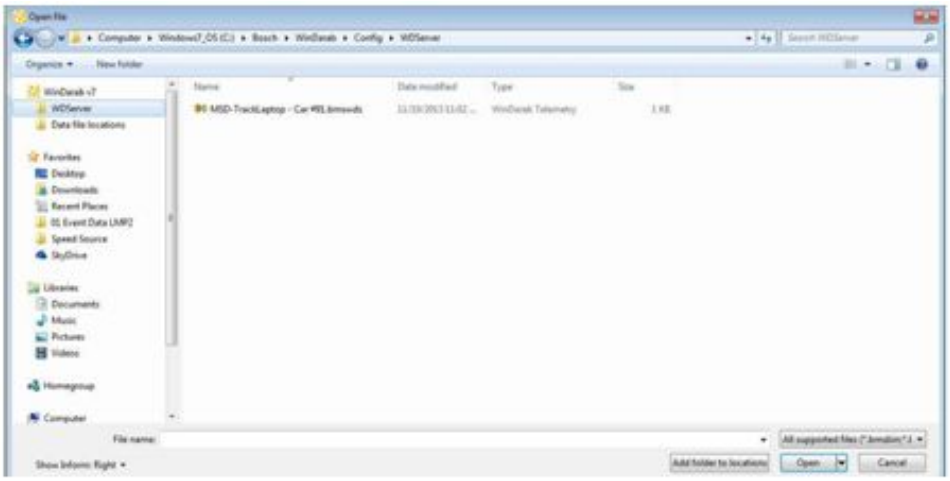


– To ensure proper communication between WDServer and the receiver, do not delete any old *.ini files from this folder path. As mentioned in section 5, RaceCon will generate a new *.ini file each time a project is synchronized; each new *.ini file instance has an incremented file name. Retaining all of these *.ini file iterations will ensure that WDServer always has a reference to whichever configuration is programmed into your vehicle's logger system.

– If the RaceCon project for the vehicle resides on a different computer, than that which is used for telemetry, then all *.ini files for a given project should be transferred to the telemetry computer after every data synchronization in RaceCon. WDServer may have trouble recognizing *.ini files stored on removable media, so best practice is to copy these files to the telemetry computer's hard drive.

18.4 Loading the telemetry data

The following is an example of a file name and data format for Car #91. File is typically located in WinDarab/Config/WDServer:



In the File Explorer, click 'Open' and navigate to the data set. Under 'Network', search for the car or cars that are required for viewing.

19 Firmware

19.1 Firmware and configuration

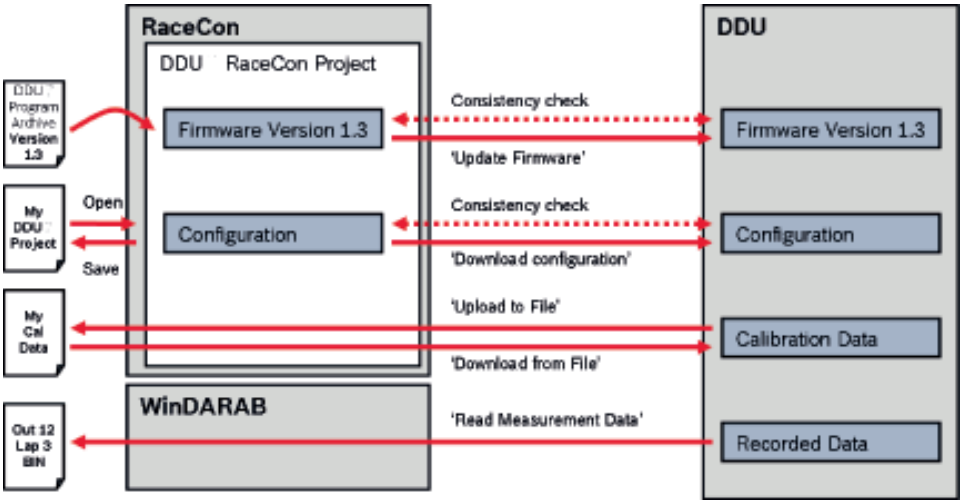
DDU 11 holds 4 types of data:

Firmware: the software (PST program file) of the DDU 11.

Configuration: the configuration of Input channels, CAN I/O, PWM, display configuration, recording + telemetry configuration.

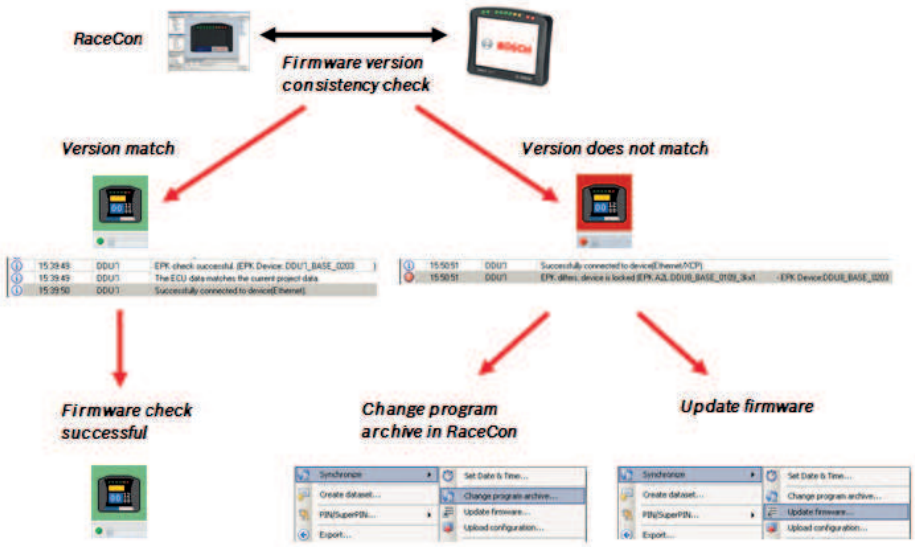
Calibration data: Characteristic curves and offsets created by online calibration at the vehicle.

Recorded data: Measurement data recorded during vehicle operation.



19.2 Firmware update

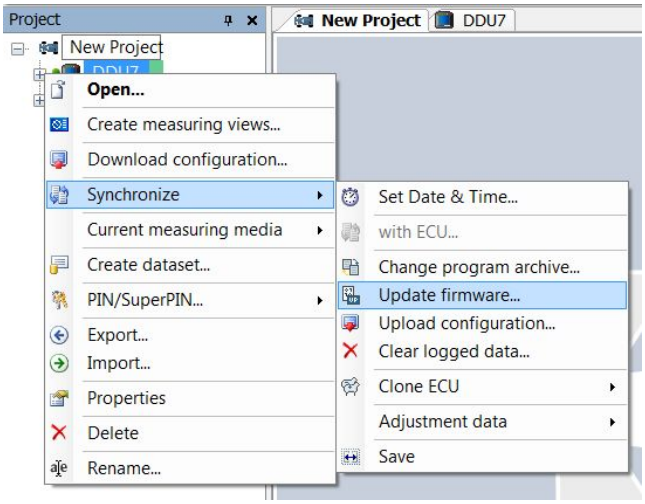
The scheme shows the process during each connection between RaceCon and DDU 11.



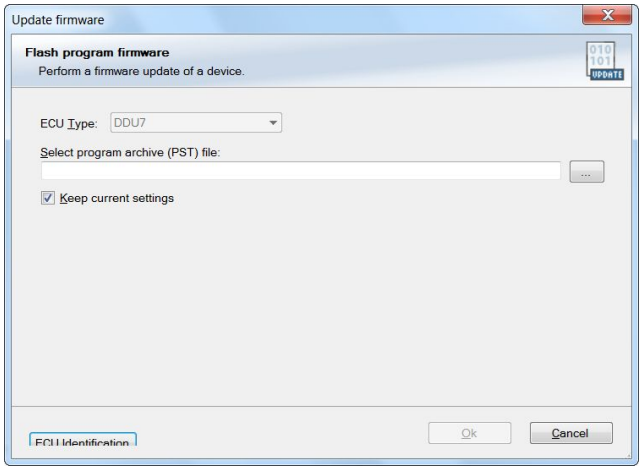
19.2.1 Performing the firmware update

Firmware update is only possible if the DDU 11 is connected to RaceCon. The configuration of Input channels, CAN I/O, display, recording + telemetry will not be changed.

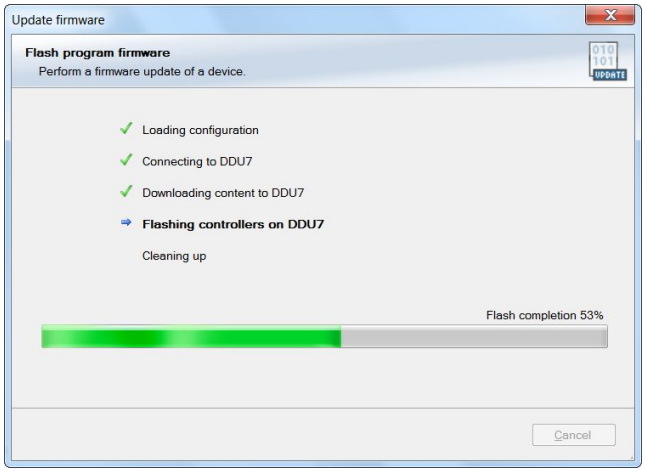
1. In the DDU 11 Project Tree, right-click on 'DDU 11' and choose 'Synchronize' then 'Update firmware'. A pop-up menu opens.



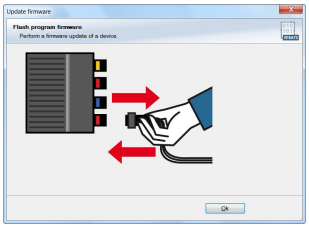
2. Select the destination of the firmware archive (PST). You can find the latest firmware for the device at the Bosch Motorsport homepage.



3. Click 'OK' when done. The firmware update starts. The DDU 11 displays the message 'Updating firmware'. Do not switch off the car's ignition or interrupt the power supply of the DDU 11!



When the firmware update is complete, the DDU 11 displays the message ‘Updating firmware finished. Do a powercycle.’ Switch the car’s ignition off and on again to cycle the power of the DDU 11.

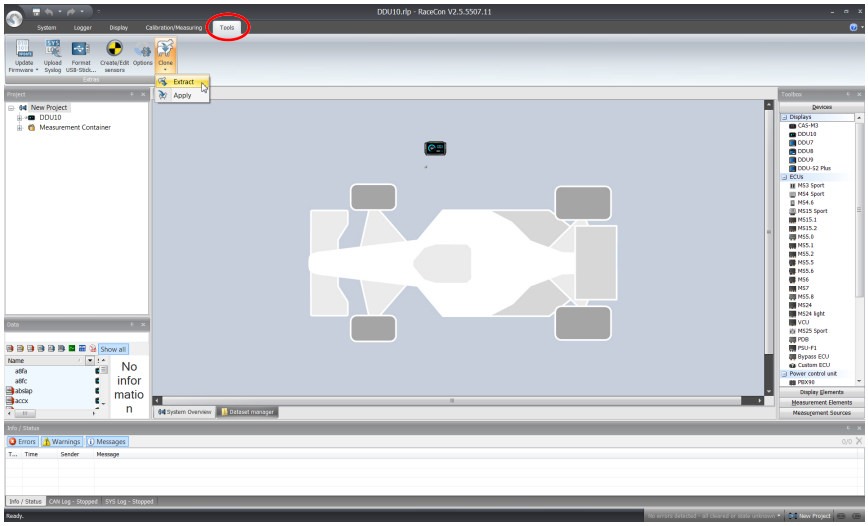


20 Cloning the Unit

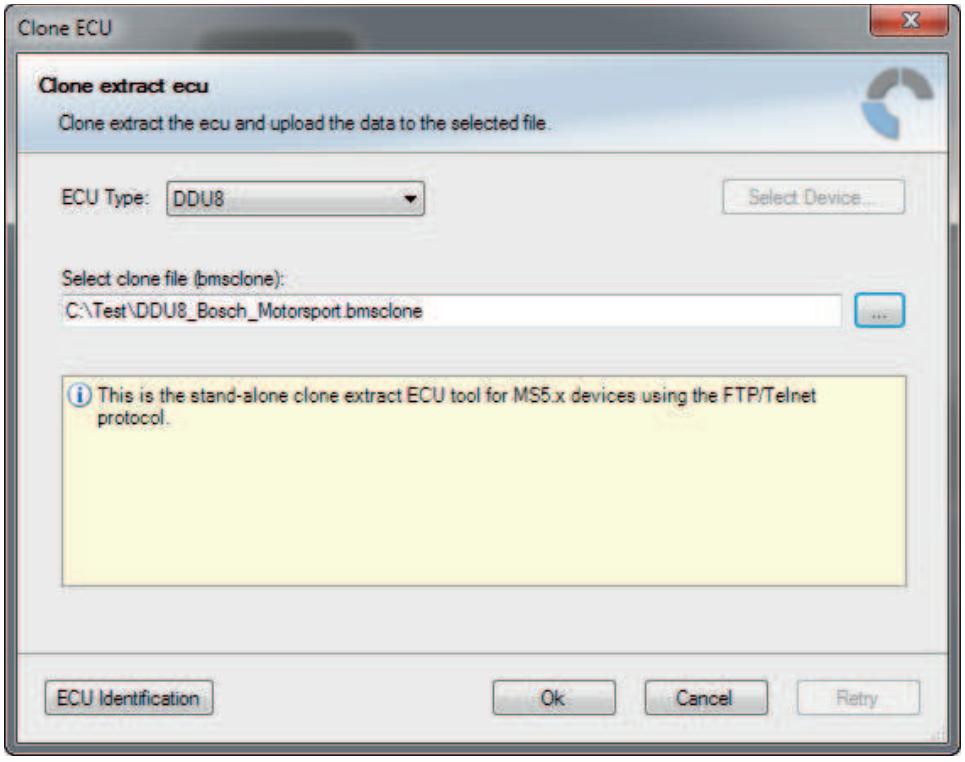
To replace a DDU 11 by another device, it is possible to clone it. A clone is a 1:1 copy of a device. This can be useful for copying specific data, like sensor-offset calibration to a spare unit for a specific car.

Creating a clone file

1. Open the 'Tools' window and click on the 'Clone' button in the 'Extras' menu. Select "Extract" from the dropdown menu.



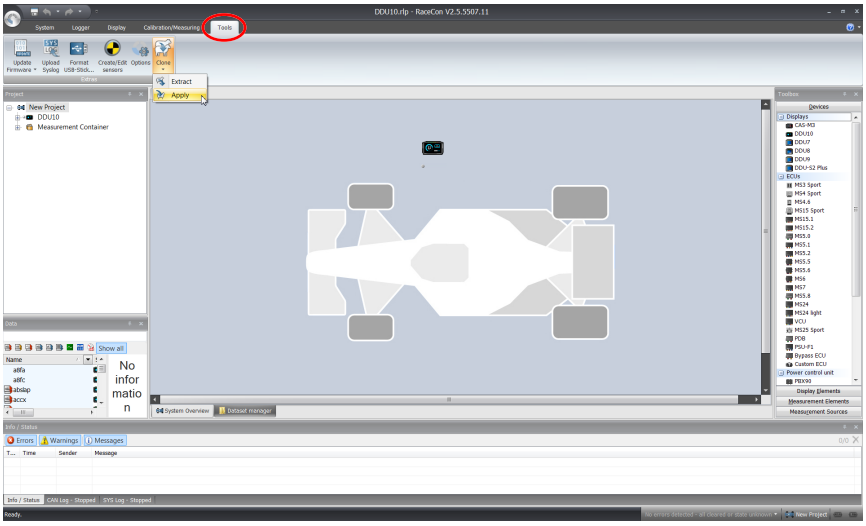
2. Choose the hardware device, which should be cloned.
3. Define destination and filename.



4. Click 'OK' to start procedure.

Applying a clone file to a device

1. Click 'Clone apply' in Extras menu.



2. Choose clone file.
3. Click ‘Ok‘.

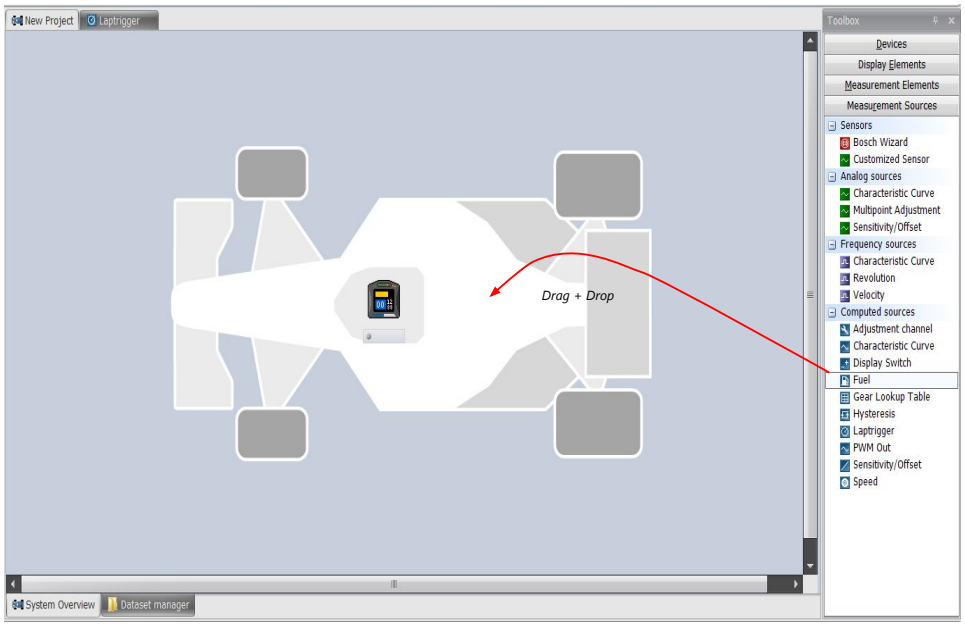
Please remember that following properties are not stored into the clone:

- Lifetime of device
- Serial number
- Upgrade features

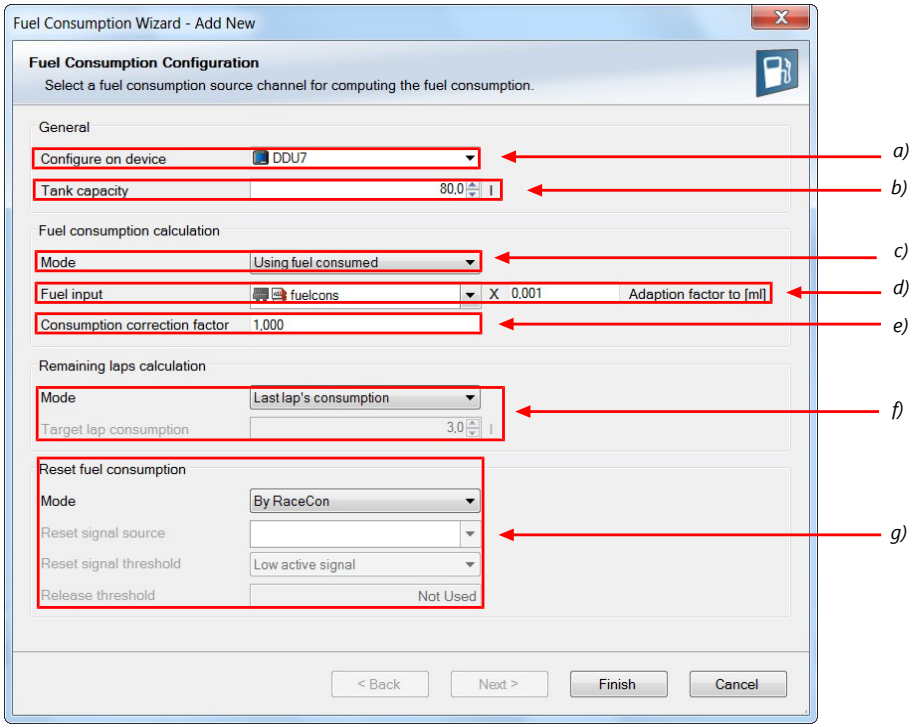
21 Fuel Consumption Calculation

21.1 Setting up fuel consumption calculation and tank management

1. Select ‘Measurement Sources’ in Toolbox.
2. Drag ‘Fuel’ element and drop it on the vehicle in System Overview. Do not drop it on the DDU 11!



A ‘fuel consumption wizard’ opens.



a) Change device for fuel calculation, if desired.

b) Enter tank capacity of vehicle.

c) Choose calculation mode:

- using fuel consumed (summed-up fuel consumption)
- using fuel flow rate (momentary fuel consumption)

d) Choose input channel and enter adaption factor. Use adaption factor to adapt value of input channel to:

- 1ml per inc for summed-up fuel consumption
- 1ml/s per inc for momentary consumption

e) Enter factor to correct calculated consumption in device vs. 'real' consumption of vehicle, if required.

f) Choose method to calculate remaining laps with fuel in tank, if desired:

- using fuel consumption of last lap completed
- using target lap consumption (entered in the field 'Target lap consumption')

g) Choose values to initiate a reset of fuel consumption, if desired:

- Manually using RaceCon
- On 'power down' (assuming that the tank is filled each time the ignition is turned off)
- By signal source as input channel (e.g. a switch connected to input pin)

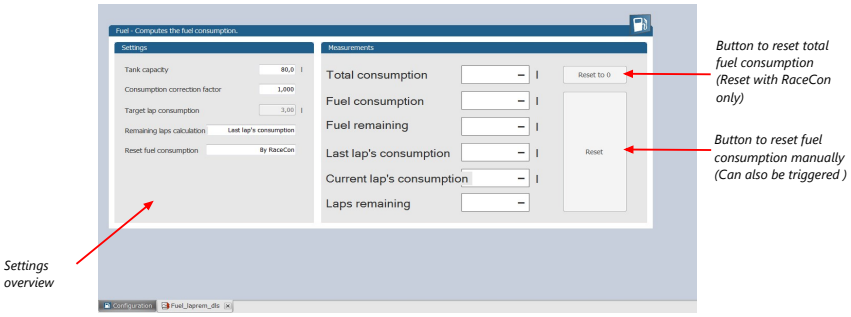
Press 'Finish' when done.

21.2 Fuel consumption diagnosis/counter reset

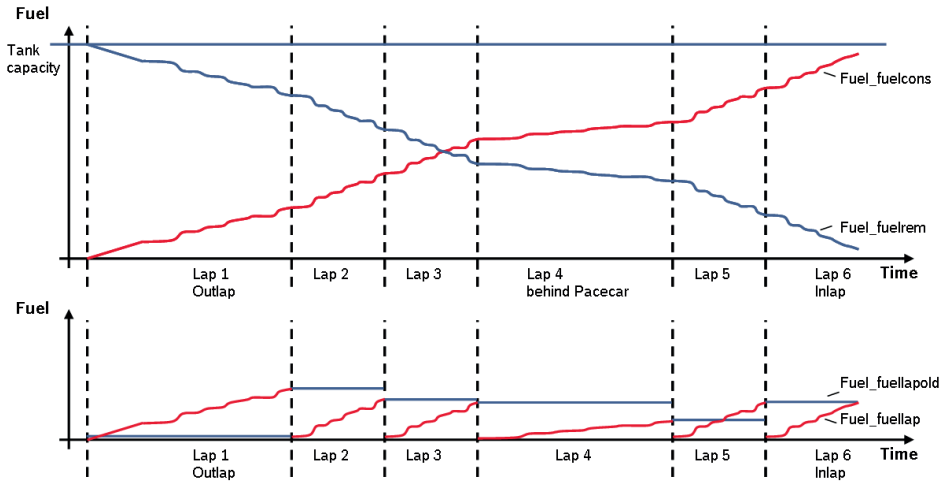
To display a fuel consumption diagnosis and to reset counters, use the diagnosis page in RaceCon.

Double-click on any 'fuel_xxx' channel in channel list.

A diagnosis window opens in Main Area.



Example



Measurement label	Function
Fuel_fuelcons_dls	Running fuel consumption, starting at '0'
Fuel_fuelrem_dls	Remaining fuel in tank, starting at tank capacity
Fuel_fuellap_dls	Fuel consumption for current lap, starting at '0'
Fuel_fuellapold_dls	Fuel consumption of last lap completed
Fuel_laprem_dls	Remaining laps with fuel in tank

22 RaceCon Shortcuts

The table shows important shortcuts simplify controlling the DDU 11 in RaceCon.

Shortcut	Function
General navigation	
F1	Open RaceCon help
F2	Rename selected object
F3	Select Data Area
F4	Select Project Tree
F5	-
F6	Start the data comparison
F7	Start dataset manager
F8	Toggle WP/RP
F9	Start measurement
CTRL + F9	Start recording
F10 or Alt	Go to menu bar
F11	Toggle display to fullscreen 'Race Mode'
F12	Enlarge main screen
CTRL + Tab	Switch between opened windows
Project Tree	
Plus (+) at numeric pad or right cursor	Expand selected node
Minus (-) at numeric pad or left cursor	Close selected node
Star (*) at numeric pad	Open all nodes
DEL	Delete seleted object
Display page, measurement page	
Cursor	Move selected display element one grid unit in chosen direction
SHIFT + cursor	Enlarge/reduce selected display element one grid unit
Tab	Switch between display elements

23 Legal

23.1 Legal Restrictions of Sale

The sale of this product in Mexico is prohibited.

Due to embargo restrictions, sale of this product in Russia, Belarus, Iran, Syria, and North Korea is prohibited.

23.2 Open Source Software (OSS) Declaration for the Display

These third party software components are used within the Graphic Rendering Engine.

23.2.1 antlr-2.7.7.jar License

ANTLR-2.7.7

SOFTWARE RIGHTS

ANTLR 1989-2006 Developed by Terence Parr

Partially supported by University of San Francisco & jGuru.com

We reserve no legal rights to the ANTLR--it is fully in the public domain. An individual or company may do whatever they wish with source code distributed with ANTLR or the code generated by ANTLR, including the incorporation of ANTLR, or its output, into commercial software.

We encourage users to develop software with ANTLR. However, we do ask that credit is given to us for developing ANTLR. By "credit", we mean that if you use ANTLR or incorporate any source code into one of your programs (commercial product, research project, or otherwise) that you acknowledge this fact somewhere in the documentation, research report, etc... If you like ANTLR and have developed a nice tool with the output, please mention that you developed it using ANTLR. In addition, we ask that the headers remain intact in our source code. As long as these guidelines are kept, we expect to continue enhancing this system and expect to make other tools available as they are completed.

The primary ANTLR guy:

Terence Parr

parrt@cs.usfca.edu

parrt@antlr.org

23.2.2 antlr311runtime.jar License

ANTLR-3.1.1

ANTLR 3 License

[The BSD License]

Copyright (c) 2010 Terence Parr

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of the author nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

23.2.3 BSD 3-Clause License (getsubopt)

The NetBSD getsubopt argument parsing function is used by Graphic Engine plugins to parse plugin options.

Applies To:

libgre.dll, libgre.a

The Graphic Runtime engines used on most operating systems, including all Windows platforms (win32).

Copyright (c) 1990, 1993

The Regents of the University of California. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

23.2.4 BSD 4-Clause License (getopt)

The NetBSD getopt argument parsing function is used to parse command line arguments.

The Graphic Runtime engines used on all Windows platforms (win32, wince, wincompact7, wec2013) includes the getopt source from the NetBSD operating system distribution.

Copyright (c) 1987, 1993, 1994

The Regents of the University of California. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgement:

   This product includes software developed by the University of California, Berkeley and its contributors.
4. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER

CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

23.2.5 crc32 License

Copyright (c) 2003 Markus Friedl. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

23.2.6 EZXML License

The ezxml library (ezxml.sourceforge.net) provides XML model parsing support for the Graphic Engine.

Applies To:

libgre.dll, libgre.a

The ezxml xml parsing library is used by all Graphic Engine runtime configurations

Copyright 2004-2006 Aaron Voisine aaron@voisine.org

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL

THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

23.2.7 FreeType License

The FreeType Project's www.freetype.org) library is used by the Graphic Engine for non-bitmap font rendering.

Applies To:

All Graphic Runtime Engines using sbfreetype libraries

Applies To:

librender-plugin-*.dll, librender-plugin-*.a

Unless explicitly configured, all Graphic Engine render plugins link against the FreeType libraries.

Portions of this software are copyright © <2011> The FreeType Project (www.freetype.org). All rights reserved.

Legal Terms

0. Definitions

Throughout this license, the terms `package', `FreeType Project', and `FreeType archive' refer to the set of files originally distributed by the authors (David Turner, Robert Wilhelm, and Werner Lemberg) as the `FreeType Project', be they named as alpha, beta or final release.

`You' refers to the licensee, or person using the project, where `using' is a generic term including compiling the project's source code as well as linking it to form a `program' or `executable'. This program is referred to as `a program using the FreeType engine'.

This license applies to all files distributed in the original FreeType Project, including all source code, binaries and documentation, unless otherwise stated in the file in its original, unmodified form as distributed in the original archive. If you are unsure whether or not a particular file is covered by this license, you must contact us to verify this.

The FreeType Project is copyright (C) 1996-2000 by David Turner, Robert Wilhelm, and Werner Lemberg. All rights reserved except as specified below.

1. No Warranty

   THE FREETYPE PROJECT IS PROVIDED `AS IS' WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT WILL ANY OF THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY DAMAGES CAUSED BY THE USE OR THE INABILITY TO USE, OF THE FREETYPE PROJECT.

2. Redistribution

   This license grants a worldwide, royalty-free, perpetual and irrevocable right and license to use, execute, perform, compile, display, copy, create derivative works of, distribute and sublicense the FreeType Project (in both source and object code forms) and derivative works thereof for any purpose; and to authorize others to exercise some or all of the rights granted herein, subject to the following conditions:

- Redistribution of source code must retain this license file (`FTL.TXT') unaltered; any additions, deletions or changes to the original files must be clearly indicated in accompanying documentation. The copyright notices of the unaltered, original files must be preserved in all copies of source files.
- Redistribution in binary form must provide a disclaimer that states that the software is based in part of the work of the FreeType Team, in the distribution documentation. We also encourage you to put an URL to the FreeType web page in your documentation, though this isn't mandatory.

These conditions apply to any software derived from or based on the FreeType Project, not just the unmodified files. If you use our work, you must acknowledge us. However, no fee need be paid to us.

3. Advertising

   Neither the FreeType authors and contributors nor you shall use the name of the other for commercial, advertising, or promotional purposes without specific prior written permission.

   We suggest, but do not require, that you use one or more of the following phrases to refer to this software in your documentation or advertising materials: `FreeType Project', `FreeType Engine', `FreeType library', or `FreeType Distribution'.

   As you have not signed this license, you are not required to accept it. However, as the FreeType Project is copyrighted material, only this license, or another one contracted with the authors, grants you the right to use, distribute, and modify it. Therefore, by using, distributing, or modifying the FreeType Project, you indicate that you understand and accept all the terms of this license.

23.2.8 GNU LESSER GENERAL PUBLIC LICENSE (pthread-win32)

Applies To:

All Graphic Runtime Engines using Windows operating systems

Applies To:

pthreadVC2.dll, pthreadVC2.lib

This provides POSIX pthread API functionality in a Windows environment (win32, wince, wincompact7, wec2013).

Project Page: https://www.sourceware.org/pthreads-win32/

pthreads-win32 - a POSIX threads library for Microsoft Windows

This file is Copyrighted

This file is covered under the following Copyright:

Copyright (C) 2001,2006 Ross P. Johnson

All rights reserved.

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Pthreads-win32 is covered by the GNU Lesser General Public License

Pthreads-win32 is open software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation version 2.1 of the License.

Pthreads-win32 is several binary link libraries, several modules, associated interface definition files and scripts used to control its compilation and installation.

Pthreads-win32 is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

See the GNU Lesser General Public License for more details.

A copy of the GNU Lesser General Public License is distributed with pthreads-win32 under the filename:

COPYING.LIB

You should have received a copy of the version 2.1 GNU Lesser General Public License with pthreads-win32; if not, write to:

Free Software Foundation, Inc.

59 Temple Place

Suite 330

Boston, MA 02111-1307

USA

The contact addresses for pthreads-win32 is as follows:

Web: http://sources.redhat.com/pthreads-win32

Email: Ross Johnson

Please use: Firstname.Lastname@homemail.com.au

Pthreads-win32 copyrights and exception files

With the exception of the files listed below, Pthreads-win32 is covered under the following GNU Lesser General Public License Copyrights:

Pthreads-win32 - POSIX Threads Library for Win32

Copyright(C) 1998 John E. Bossom

Copyright(C) 1999,2006 Pthreads-win32 contributors

The current list of contributors is contained in the file CONTRIBUTORS included with the source code distribution. The current list of CONTRIBUTORS can also be seen at the following WWW location:

http://sources.redhat.com/pthreads-win32/contributors.html

Contact Email: Ross Johnson

Please use: Firstname.Lastname@homemail.com.au

These files are not covered under one of the Copyrights listed above:

COPYING

COPYING.LIB

tests/rwlock7.c

This file, COPYING, is distributed under the Copyright found at the top of this file. It is important to note that you may distribute verbatim copies of this file but you may not modify this file.

The file COPYING.LIB, which contains a copy of the version 2.1 GNU Lesser General Public License, is itself copyrighted by the Free Software Foundation, Inc. Please note that the Free Software Foundation, Inc. does NOT have a copyright over Pthreads-win32, only the COPYING.LIB that is supplied with pthreads-win32.

The file tests/rwlock7.c is derived from code written by Dave Butenhof for his book 'Programming With POSIX(R) Threads'. The original code was obtained by free download from his website http://home.earthlink.net/~anneart/family/Threads/source.html and did not contain a copyright or author notice. It is assumed to be freely distributable.

In all cases one may use and distribute these exception files freely. And because one may freely distribute the LGPL covered files, the entire pthreads-win32 source may be freely used and distributed.

General Copyleft and License info

For general information on Copylefts, see:

http://www.gnu.org/copyleft/

For information on GNU Lesser General Public Licenses, see:

http://www.gnu.org/copyleft/lesser.html

http://www.gnu.org/copyleft/lesser.txt

Why pthreads-win32 did not use the GNU General Public License

The goal of the pthreads-win32 project has been to provide a quality and complete implementation of the POSIX threads API for Microsoft Windows within the limits imposed by virtue of it being a stand-alone library and not linked directly to other POSIX compliant libraries. For example, some functions and features, such as those based on POSIX signals, are missing.

Pthreads-win32 is a library, available in several different versions depending on supported compilers, and may be used as a dynamically linked module or a statically linked set of binary modules. It is not an application on it's own.

It was fully intended that pthreads-win32 be usable with commercial software not covered by either the GPL or the LGPL licenses. Pthreads-win32 has many contributors to it's code base, many of whom have done so because they have used the library in commercial or proprietry software projects.

Releasing pthreads-win32 under the LGPL ensures that the library can be used widely, while at the same time ensures that bug fixes and improvements to the pthreads-win32 code itself is returned to benefit all current and future users of the library.

Although pthreads-win32 makes it possible for applications that use POSIX threads to be ported to Win32 platforms, the broader goal of the project is to encourage the use of open standards, and in particular, to make it just a little easier for developers writing Win32 applications to consider widening the potential market for their products.

23.2.9 log4j.jar License

The Apache Software License, Version 1.1

Copyright (C) 1999 The Apache Software Foundation. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The end-user documentation included with the redistribution, if any, must include the following acknowledgment: "This product includes software developed by the Apache Software Foundation (http://www.apache.org/)." Alternately, this acknowledgment may appear in the software itself, if and wherever such third-party acknowledgments normally appear.
4. The names "log4j" and "Apache Software Foundation" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact apache@apache.org.
5. Products derived from this software may not be called "Apache", nor may "Apache" appear in their name, without prior written permission of the Apache Software Foundation.

THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This software consists of voluntary contributions made by many individuals on behalf of the Apache Software Foundation. For more information on the Apache Software Foundation, please see <http://www.apache.org/>.

/

23.2.10 Lua License

The Lua engine (http://www.lua.org) provides a script interface for manipulating content in a Graphic application.

Applies To:

libgre-plugin-lua.dll, libgre-plugin-lua.a

The Lua Graphic Engine plugin

License for Lua 5.0 and later versions

Copyright © 1994-2008 Lua.org, PUC-Rio. Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the

Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions: The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

23.2.11 opencsv License

opencsv http://opencsv.sourceforge.net sconway@users.sourceforge.net arjones@t-online.de

Apache License

Version 2.0, January 2004

http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

   "License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

   "Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

   "Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

   "You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

   "Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

   "Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

   "Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or

attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition,

"submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

   (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

23.2.12 Sensor Driver for BMI160 Sensor

Applies to BMI160

Copyright (C) 2014 Bosch Sensortec GmbH

License:

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of the copyright holder nor the names of the contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE

The information provided is believed to be accurate and reliable.

The copyright holder assumes no responsibility for the consequences of use of such information nor for any infringement of patents or other rights of third parties which may result from its use.

No license is granted by implication or otherwise under any patent or patent rights of the copyright holder.

23.2.13 SharpZipLib License

SharpZipLib 0.86.0 © Mike Krueger and John Reilly https://github.com/icsharpcode/SharpZipLib

MIT License

==========

Copyright © 2000-2016 SharpZipLib Contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

23.2.14 Simple OpenGL Library License (SOIL)

The SOIL library (http://www.lonesock.net/soil.html) is used to load images in a Graphic application.

Applies To:

libgre.dll, libgre.a

The SOIL image library is used by all Graphic Engine runtime configurations

Jonathan Dummer

2007-07-26-10.36

Simple OpenGL Image Library

Public Domain

using Sean Barret's stb_image as a base

Thanks to:

* Sean Barret - for the awesome stb_image

* Dan Venkitachalam - for finding some non-compliant DDS files, and patching some explicit casts

* everybody at gamedev.net

23.2.15 stringtemplate License

[The "BSD licence"]

Copyright (c) 2003-2008 Terence Parr

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

23.2.16 xml_io_tools License

xml_io_tools

Copyright (c) 2007, Jaroslaw Tuszynski

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,

STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

23.3 REACH Statement

According to the REACH regulations, any supplier of an article containing a substance of very high concern (SVHC) in a concentration above 0.1 % (w/w) has the duty to provide the recipient of the article with sufficient information to allow safe use of the article. Our product contains:

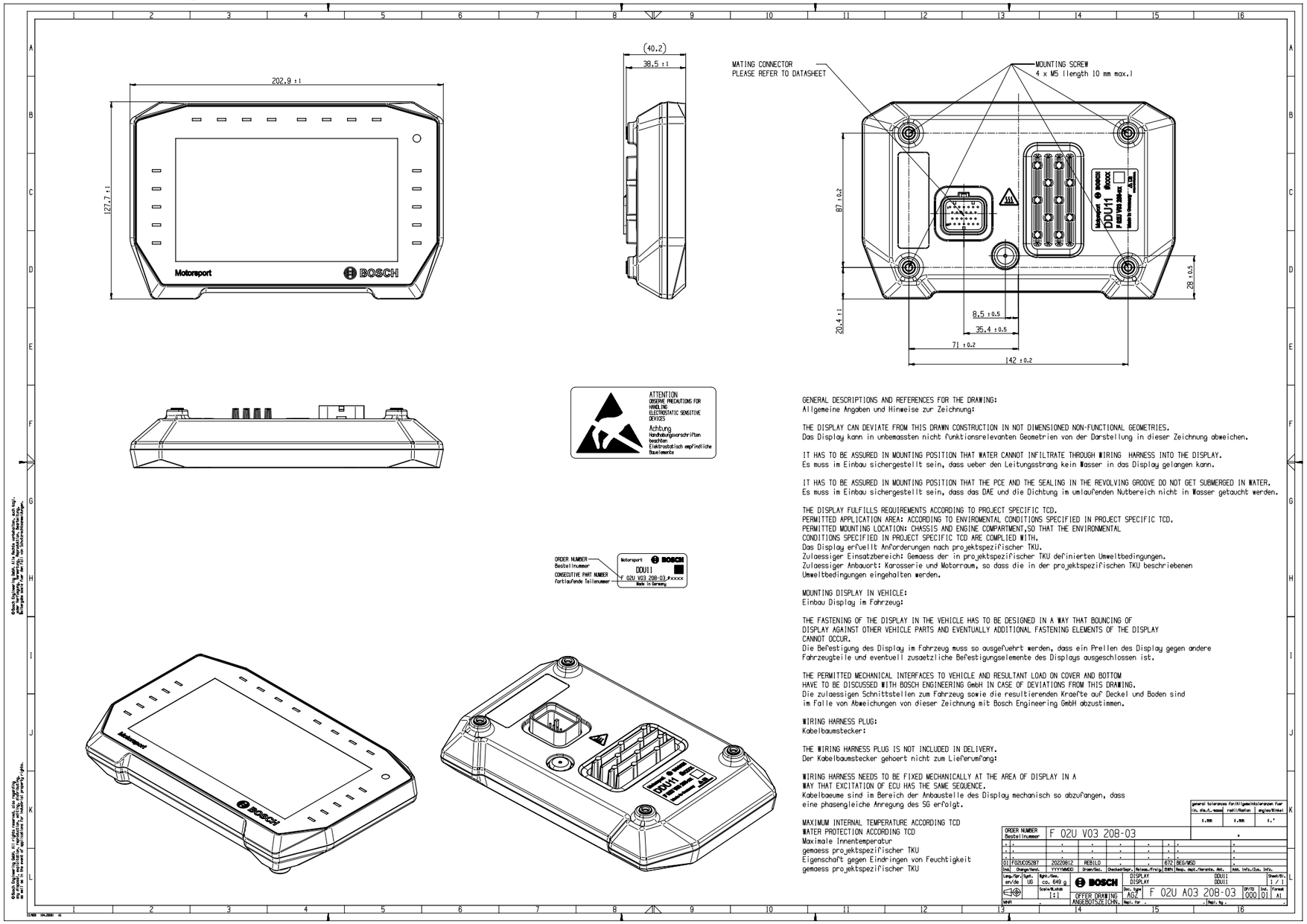
SVHC Substance	CAS Number
Lead	7439-92-1

24 Disposal

Hardware, accessories and packaging should be sorted for recycling in an environment-friendly manner.

Do not dispose of this electronic device in your household waste.

25 Offer Drawing



Bosch Engineering GmbH
Motorsport
Robert-Bosch-Allee 1
74232 Abstatt

www.bosch-motorsport.com