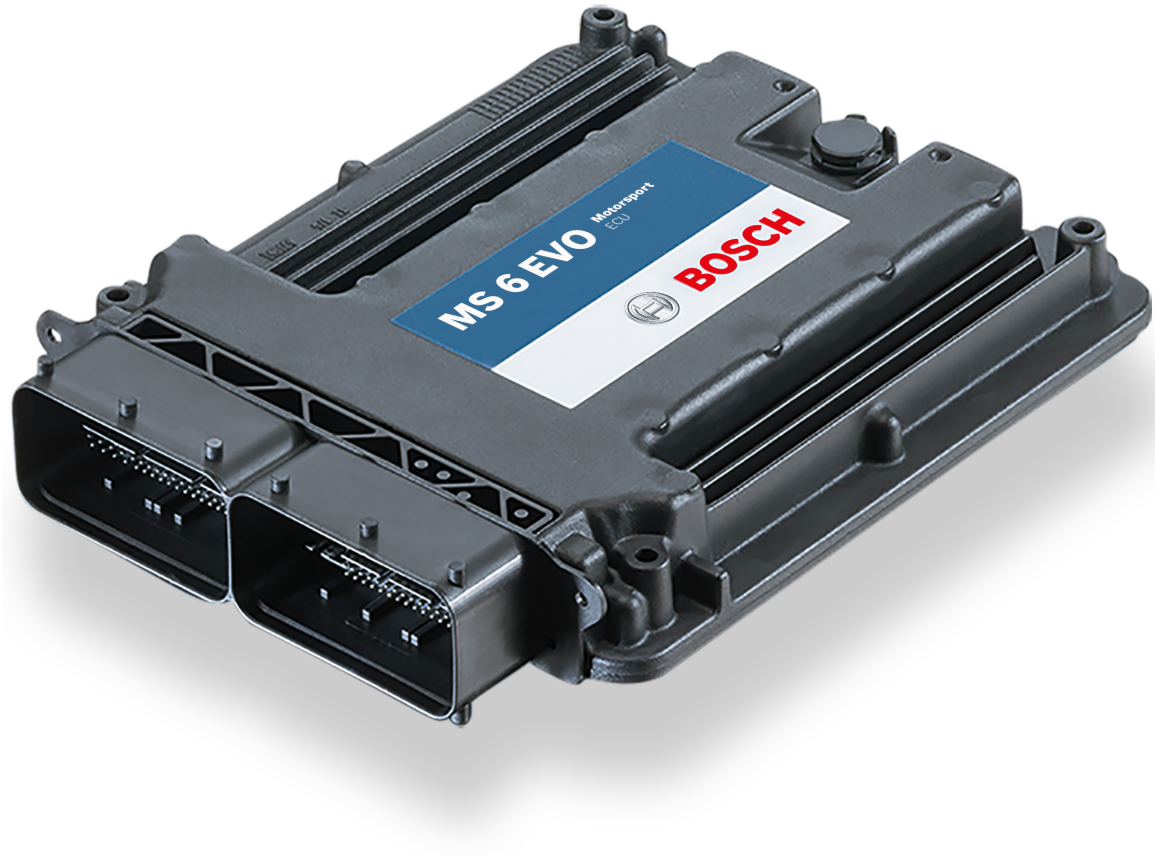
BOSCH



Engine Control Unit MS 6 EVO

Manual

Version 2.4 29/04/2025

Content

1 Getting Started

Disclaimer

Due to continuous enhancements we reserve the rights to change illustrations, photos or technical data within this manual. Please retain this manual for your records.

Before starting

Before starting your engine for the first time, install the complete software. Bosch Motorsport software is developed for Windows operation systems. Read the manual carefully and follow the application hints step by step. Don't hesitate to contact us. Contact data can be found on the backside of this document.

⚠ CAUTION

Risk of injury if using the MS 6 EVO inappropriately.

Use the MS 6 EVO only as intended in this manual. Any maintenance or repair must be performed by authorized and qualified personnel approved by Bosch Motorsport.

⚠ CAUTION

Risk of injury if using the MS 6 EVO with uncertified combinations and accessories

Operation of the MS 6 EVO 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 is only permitted when they have been determined to be compliant from a performance and safety standpoint by a representative from Bosch Motorsport.

NOTICE

For professionals only

The Bosch Motorsport MS 6 EVO was developed for use by professionals and requires in depth knowledge of automobile technology and experience in motorsport. Using the system does not come without its risks.

It is the duty of the customer to use the system for motor racing purposes only and not on public roads. We accept no responsibility for the reliability of the system on public roads. If the system is used on public roads, we shall not be held responsible or liable for damages.

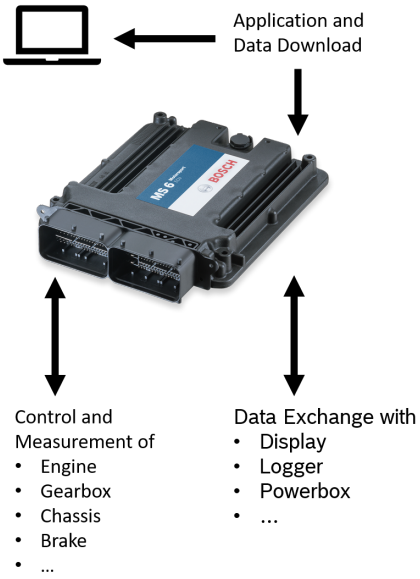
2 Technical Data

The MS 6 EVO engine control unit features a powerful digital processing dual-core with floating point arithmetic and a high-end field programmable gate array FPGA for ultimate performance and flexibility.

The software development process is based on MATLAB® & Simulink®. It significantly speeds algorithm development by using automatic code and documentation generation.

Custom functions can be generated quickly and easily. The flexible hardware design allows the MS 6 EVO to support complex or unusual engine or chassis configurations. Integrated logger control areas present a cost efficient and weight optimized all-in-one solution.

2.1 System Layout



Layout restrictions

CAN Network	Extended number of members and wiring leads extend the risk of error frames
RS232	Limited to one additional component
USB	Limited to additional Bosch Motorsport USB stick
LIN	Permitted for the use of Bosch Motorsport preconfigured configurations
SENT	Use with preconfigured configurations that are available from Bosch Motorsport on request

2.2 Mechanical Data

Aluminum housing	
2 automotive connectors, 196 pins in total	
Vibration suppression via multipoint fixed circuit boards	
Size without connectors	226 x 181 x 44 mm
Weight	1,086 g

Protection Classification	IP54
Temperature range	-20 to 80°C (0 to 80°C for P-Versions)
Inspection services recommended after 220 h or 2 years, no components to replace	

2.3 Electrical Data

Power supply	6 to 18 V
CPU	Dual Core 667 MHz; FPGA (866 MHz for P-Versions)

2.3.1 Inputs

The analogue inputs are divided in different hardware classes and qualities.

3.01 kOhm pull-ups are fixed or switchable designed to assist passive sensor elements like NTC temperature sensors or to change to active signal inputs.

Some of the inputs assist only active sensors and offer no pull-up.

To improve measurement tasks, angle related measurements are an option for some inputs, mainly used for engine related leading signals.

The connection between function and related input is free selectable, beside electronic throttle functionalities.

All linearization mappings are open to the customer, some signals offer online modes to calibrate gain and offset.

Digital inputs for speed measuring offer divers hardware options to connect inductive- or digital speed sensors.

Please respect: for camshaft- or wheel speed signals Hall-effect or DF11 sensors have to be used and for wide range Lambda measurement and control the Lambda sensor Bosch LSU 4.9 has to be used.

Standard number of Inputs; for additional channels see Structure of Devices and Licenses [▸ 8]

38 analog inputs (CUP: 26; 6.1, 6.3: 21)
6 x reserved for electronic throttle controls (Cup: 4)
10 x no integrated pull-up (Cup: 5; 6.1, 6.3: 3)
4 x option for angle synchronous measurement, no integrated pull-up (Cup, 6.1, 6.3: 3)
5 x fixed 3.01 kOhm pull-up (Cup, 6.1, 6.3: 4)
13 x switchable 3.01 kOhm pull-up (Cup: 10; 6.1, 6.3: 5)

8 analog/digital inputs (shared) (CUP, 6.1, 6.3: 0)
8 x option for angle synchronous measurement / digital (e.g. SENT)

10 digital inputs (CUP, 6.1, 6.3: 18)
1 x switchable Hall or inductive sensor for flywheel measurement
2 x Hall sensor for sync wheel detection
4 x switchable Hall or DF11 sensors for camshaft position or wheel speed
2 x switchable Hall or inductive sensors for turbo speed measurement

10 digital inputs (CUP, 6.1, 6.3: 18)
1 x digital switch for engine ON/OFF
8 x digital inputs, e.g. SENT (Only CUP, 6.1, 6.3)

6 internal measurements
1 x ambient pressure
1 x acceleration 6-axis
2 x ECU temperature
2 x ECU voltage

9 function related inputs (CUP: 3; 6.1, 6.3: 8)
2 x thermocouple exhaust gas temperature sensors (K-type) (CUP, 6.1, 6.3: 1)
2 x Lambda interfaces for LSU 4.9 sensor types (CUP: 1)
1 x lap trigger/beacon input (CUP: 0)
4 x knock sensors (CUP: 1)

2.3.2 Sensor supplies and screens

4 x sensor supplies 5 V / 50 mA
3 x sensor supplies 5 V / 150 mA
7 x sensor grounds
2 x sensor screens

2.3.3 Outputs

38 function related outputs (CUP: 15; 6.1, 6.2: 28)
High Pressure Injection (not 6.1, 6.2) – 8 x high pressure injection power stages for magnetic valves, e.g. HDEV 5 (CUP: 4) – 2 x outputs for high pressure pump with MSV controls (6.3: on request; CUP: 1)
Low Pressure Injection – 12 x low pressure injection power stages for high impedance valves (max. 2.2 amps and min. 6 Ohm internal resistance of the injectors) (CUP: 4)
Ignition – 12 x ignition controls, support of coils with integrated amplifier only (CUP: 4)
2 x 8.5 amp H-bridge for electronic throttle control (CUP: 1)
2 x 4 amp pwm lowside switch for Lambda heater (CUP: 1)

19 freely configurable outputs (CUP: 13)
1 x 8.5 amp H-bridge (CUP: 2)
2 x 4 amp pwm lowside switch (CUP: 1)
4 x 3 amp pwm lowside switch (CUP: 2)
8 x 2.2 amp pwm lowside switch (CUP: 5)

19 freely configurable outputs (CUP: 13)
4 x 1 amp pwm lowside switch (CUP: 3)

3 output signals
1 x engine rpm
1 x flywheel
1 x trigger wheel

2.4 Communication

3 x CAN	The MS 6 EVO has 3 CAN buses configurable as input and output. Different baud rates are selectable. Please note that the MS 6 EVO contain integrated switchable 120 Ohm CAN termination resistors.
1 x LIN	The Bus is not configurable by the customer, but Bosch Motorsport offers data selectable protocols to integrate LIN based devices into the system.
8 x SENT	The MS 6 EVO has 8 SENT interfaces for using SAEJ2716.
2 x Ethernet	Integrated are 100 Mbit full duplex Ethernet communication ports, internally connected with an Ethernet switch. The ports have “cable auto crossover” functionality
1 x USB	For data transfer to an USB-stick
1 x RS232	One serial port with programmable baud rate for online telemetry
1 x Timesync Co-ordination	For additional devices added via Ethernet

2.5 Structure of Devices and Licenses

To accommodate the wide range of different engine requirements and racetrack operating conditions, the MS 6 EVO Motronic system is classified into the main groups high- and low pressure injection support, subdivided into fully equipped and functional reduced versions.

Beside the change from low- to high-pressure systems, all limited functions may be activated later. The license concept is related to the individual device and the requested upgrading.

For MS 6.1 EVO	
Engine function package I	To activate electronic throttle, camshaft, and turbo control
Engine function package II	To activate traction and launch control

For MS 6.1 EVO and MS 6.3 EVO	
Measurement package	**17 Additional analog inputs**

For MS 6.1 EVO and MS 6.3 EVO	
	7 x no integrated pull-up 1 x option for angle synchronous measurement, no integrated pull-up 1 x fixed 3.01 kOhm pull-up 8 x switchable 3.01 kOhm pull-up
	Extension of the use of 8 digital channels as analogue / digital inputs (shared)
	1 x Additional Thermocouple K-type

For MS 6.3 EVO	
High pressure injection package	Enables the control of a 2nd high pressure pump

For MS 6.4P EVO	
PERF_LOG_1 (requires FULL_LOG_1)	Increase logging Partition 1 from 4 GB to 16 GB memory
Specific project SW for MS 6.4P EVO, based on MS 6.4 EVO SW, offered as engineering service	

For all MS 6 EVO Versions	
Hardware Upgrade for CCA per device	Provides the option to run customer developed software code on Bosch ECU
FULL_LOG_1	Extension for Recording 1 • 1,500 channels • fastest sampling 1,000 Hz or 1 syncro
FULL_LOG_2	Activation of Recording 2 • 1,500 channels • 4 GB memory (enabled at MS 6 EVO CUP) • fastest sampling 1,000 Hz or 1 syncro
DATA_USB	Data copy to USB flash drive
Gear control package I	Gear control MEGA-Line functionality, must be used with MEGA-Line components (License model via MEGA-Line)
-- Link to Mega Line Support Request --	
-- Link to Mega-Line License Request Form --	
Gear control package II	Gear control Bosch Motorsport functionality
SW Package MS 6 Drag 1 (not for CUP)	Launch Timer Launch Distance Torque Pre-Control Launch RPM Control Universal Outputs for Time/Distance Controls

For all MS 6 EVO Versions	
SW Package MS 6 Drag 2 (requires Drag 1 License) (not for CUP)	Acceleration Sensor MM5.10 included Time/Distance Boost Control Driveshaft Speed Control Driveshaft Gradient Control Acceleration Control Wheelie Control
Innovation License Device	Activation of a set of additional functions for a single device: – Crank rotation direction detection (using sensor DG23i) – Using a 2nd crank backup sensor – Crank-Pre-set, quick start based on previous crank stop position – Far-Bank, 2nd injector per cylinder possible – Cam-only-synchronisation, engine run without crank sensor signal (specific cam trigger wheel needed)
Innovation Package Project	Innovation Package Project has the same content as Innovation License Device, but license is valid for the whole project instead of a single device.

NOTICE

Verify the necessity of gearbox control licenses by checking the Features info window in RaceCon (see section Feature/License Activation [▶ 20]).

2.6 Installation

Mounting	Fastening with **Velcro®** / **3M Dual Lock®** or rubber band
Offer drawing	Available at Bosch Motorsport website on MS 6 EVO product page.
3D Data	Available at Bosch Motorsport website on MS 6 EVO product page.

Recommendation

Use rubber vibration absorbers for soft mounting in the vehicle. To assist the heat flow, especially if HP injection is active, the device must be mounted uncovered and air circulation must be guaranteed around the entire surface area.

Inside touring cars placement passenger side is favoured, open connectors should not be uncovered to vertical axe. It must be assured in mounting position that water cannot infiltrate through wiring harness into the ECU and that the pressure compensating element and the sealing in the revolving groove do not get submerged in water. Wiring harness needs to be fixed mechanically around the ECU in a way that excitation of ECU has the same sequence.

2.7 Supply System

Please ensure that you have a good ground installation with a solid, low resistance connection to the battery minus terminal. The connection should be free from dirt, grease, paint, anodizing, etc.

- MS 6 EVO power consumption at appr. 13 V (vary according to use cases)
  - ~ 25 - 30 amps (4 cyl. FDI at 8,500 1/min/200 bar single injection, 1 MSV, 1 electronic throttle, standard chassis equipment)
  - ~ 35 - 40 amps (8 cyl. FDI at 8,500 1/min/200 bar single injection, 2 MSV, 2 electronic throttle, standard chassis equipment)
- Power consumption of LP-injectors, actuators and coils are to calculate separately.
- The MS 6 EVO power supply is separated into the maintenance of controller and power stages.
- Ensure controller supply UBAT is activated before the power stages.
- The MS 6 EVO is able to control a main relay or even the power box itself via a low side output.
- As long as the controller is activated, data logging, telemetry and communication is also ongoing.
- The engine On/Off switch activates the ignition and injection outputs to enable engine start separately from power supply.

2.8 Harness

Harness connectors

Bosch automotive connectors are not available as complete set of components, so Bosch Motorsport itself offers such a package. For more technical details please check Bosch-connector homepage, 196 pins
http://www.bosch-connectors.com/bogscoca/category/142

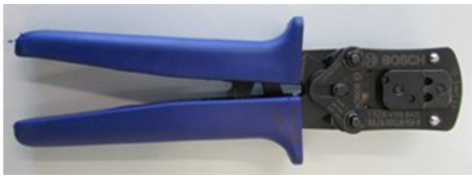
MS 6 harness connector type A (105 contacts), coding variant 1	F02U.B00.712-01
MS 6 harness connector type K (91 contacts), coding variant 1	F02U.B00.711-01
Protection Classification	IP X6K, X8, X9K
Temperature range	-40 to 120°C
Shakeproofed	Max. 3.4 g
Wiring diameter	0.35 to 2.5 mm^2
Pinsize	1.2 mm; 2.8 mm

Dummy Plug

Dummy plug 1928.405.459 for unused connections	Matrix 1.2 / CB / 0.75 to 1.0 mm^2
Dummy plug 1928.405.460 for unused connections	Matrix 1.2 / CB / 1.0 - 1.5 mm^2
Dummy plug 1928.301.207	BTL 2.8

Tools and Contacts

Tool	Matrix	Contact	Wire size
1928.498.212	Matrix 1.2	Clean Body 1928.498.991	0.35 to 0.5 mm^2
1928.498.213	Matrix 1.2	Clean Body 1928.498.992	0.75 to 1.0 mm^2
1928.498.837	1928.498.840	BTL 2.8 1928.498.651	1.5 to 2.5 mm^2



Wiring

Bosch Motorsport recommends using the specified cable material and harness layout for automotive connectors and wiring applications.

For Ethernet and USB connection CAT5 specified material is recommended and the pairs and shield connections have to be strictly respected as shown in the wiring diagram.

For USB, the maximum wiring length is limited to 3 m and it is not allowed to be included into a common harness and also there is no interruption allowed.

Due to installation condition, the length may have to be reduced.

Keep network wiring in distance to main sources of electrical noise like coils, coil- and HP-injector wirings and also in distance to any telemetry transmitter.

CAN-networks need a 120 Ohm termination at 2 ends of the wiring.

The MS 6 EVO is able to switch on an internal 120 Ohm termination, set CWCANx_TERM true to enable the termination.

For wiring layout, respect the common rules of failure reduction like separated sensor power supply between important system sensors (e.g. camshaft detection) and measure options (e.g. damper position).

Be ensure HP-injectors, electronic throttles and other high frequently switched actuators are connected within the wiring limits of 2.5 m and all wires are manufactured as twisted pairs.

If using a preinstalled production harness, first verify the way of sensor- and actuator controls.

Often production parts have to be connected to 12 V power supply and actuators are controlled in different ways. The production harness may need to be modified.

Office harness

Reduced layout to realize communication between PC, MS 6 EVO device and Display DDU, recommended for flash configuration, display configuration and installation tasks. Bosch Motorsport part number: F02U.V01.809

2.9 Ignition Trigger Wheel

To detect the engine position and to calculate the exact crankcase position, the system assumes toothed trigger wheels for proper operation. Recommended is to use 60 (-2) teeth for the flywheel and one teeth for the camshaft detection. Modifications of the mechanical designs are possible, such as using quick-start production designs for the camshaft or different number of teeth for the flywheel **(limited to 30 to 60 teeth)**.

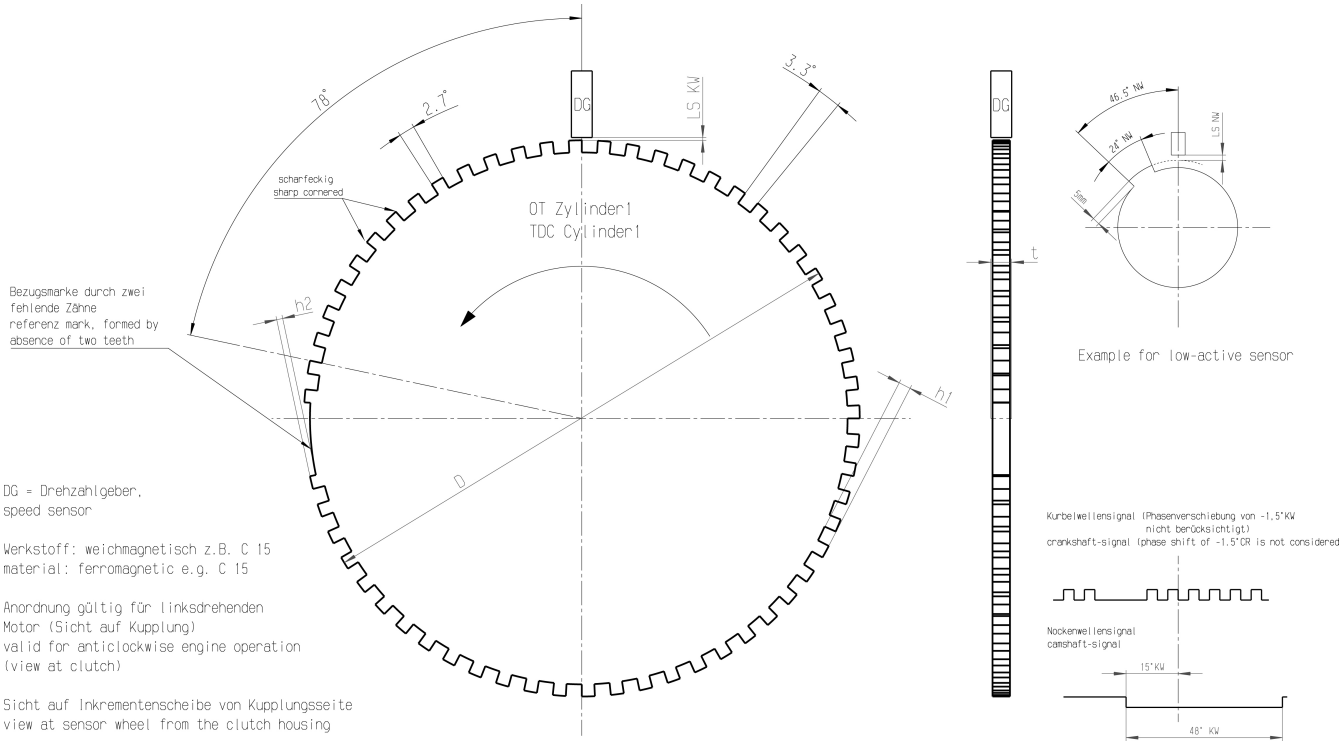
NOTICE

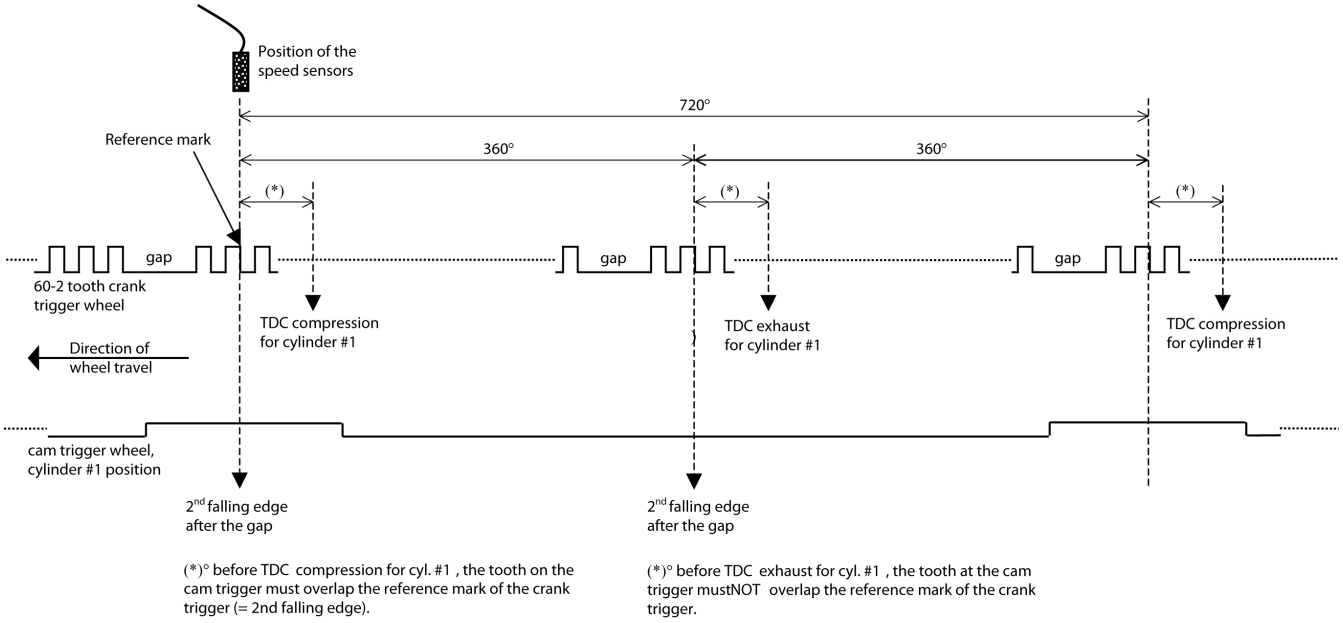
Less number of teeth reduces the accuracy of the system angle measurement.

Not usable are flywheels with 4-1 or 6-1 teeth. Please follow the description below as recommendation for the mechanical dimensions.

Recommended values:

- D = min. 160 mm
- h1 = 3.5 mm
- h2 = h1/2 (important for the use of inductive sensor)
- LSKW = 0.8 mm +/- 0.3 mm
- t = min. 5 mm
- LNSW = 1.0 mm +/- 0.5 mm





NOTICE

All angles are shown and indicated in crankshaft degrees.

The width of the cam trigger tooth is not important, however it is recommended to use at least 48 crankshaft degrees (24 cam degrees).

The Hall effect signal may be the inversion of its cam trigger: the tooth effects a “low” signal at the sensor and vice versa for other trigger wheel configurations the indicated values may vary.

3 Starting up

NOTICE

All following chapters (Starting up to Harness / Wiring) refer to the MS 6 base family. Some screenshots were taken from the MS 6 family.

3.1 Installation of Software Tools

PC tools and for the MS 6 EVO system are available at Bosch Motorsport homepage for free download, ECU programs and function description on request.

- **RaceCon** V2.7.0.9 or higher is the tool for system configuration, data application and online measurement.
- **WinDarab V7** is the analysis tool, **Light** version as shareware or **Expert** version if license available.

All tools are delivered as self-installing executable files.

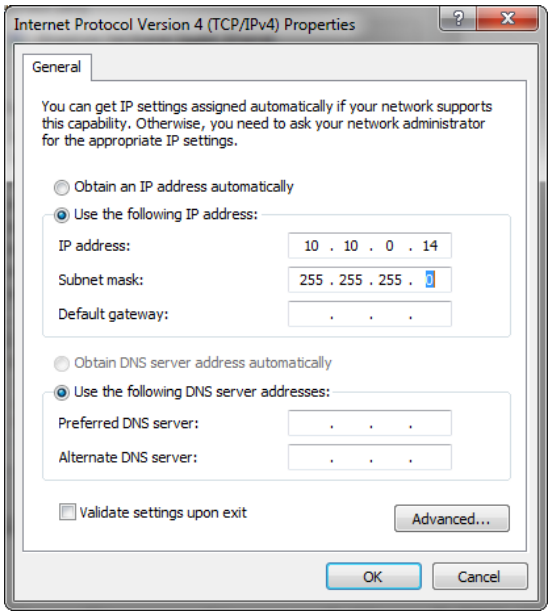
Select your personal installation folder.

3.1.1 Communication PC to device

Ethernet as used network may have some restrictions by firewall and IT protections. Be assure no firewall is active at the PC.

For assistance, Bosch Motorsport homepage explains the necessary PC installations.

The MS 6 EVO system requests a defined IP-adress at the PC, for example 10.10.0.14.



Middle of 2016, programs and basic systems were extended to handle automatic TCP/IP selection also. Former produced devices and program versions may be modified to customer request and -order.

MS 6 EVO devices are connectable via commercial CAT7 cables to the PC; also Bosch Motorsport offers diagnostic cable and programming harnesses as track- and office connections.

Successful connection between PC and MS 6 EVO is shown as green marked connection in the top left corner of RaceCon.

3.2 Configuration of the system

Bosch MS 6 EVO devices are delivered in a not engine executable mode. The customer must include the correct programs, data applications and licenses.

The MS 6 EVO offers two mainly different configuration areas, related to the two core areas of the controller.

MS 6 EVO ECU

1st core area for the functional part of the MS 6 EVO program. The available content is documented in the functional descriptions Bosch Motorsport adds to the customer deliveries. Application works will be done via opening the data labels in the edition windows of INCA or RaceCon.

MS 6 EVO Logger

2nd core area for the tool displayed parts like logger-, lap trigger, telemetry, and CAN-network configurations. Application work will be done in the predefined function windows of RaceCon.

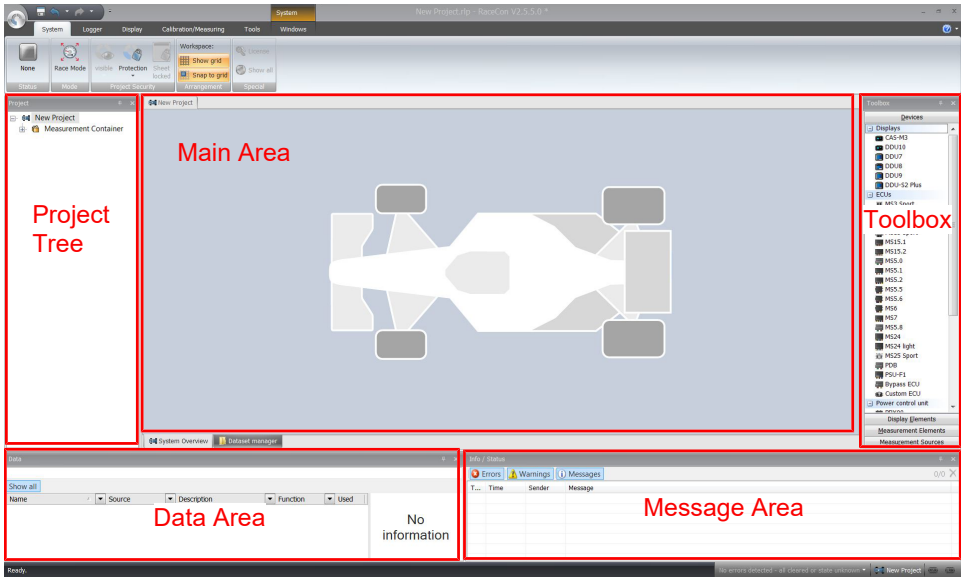
MS 6 EVO Programming

For system programming or flashing of the device we developed the system configuration tool RaceCon. After the start of the tool, RaceCon opens the screen “Welcome to RaceCon”.

With “Last Projects” former projects can be opened directly.

3.2.1 First Steps to create and configure a Project

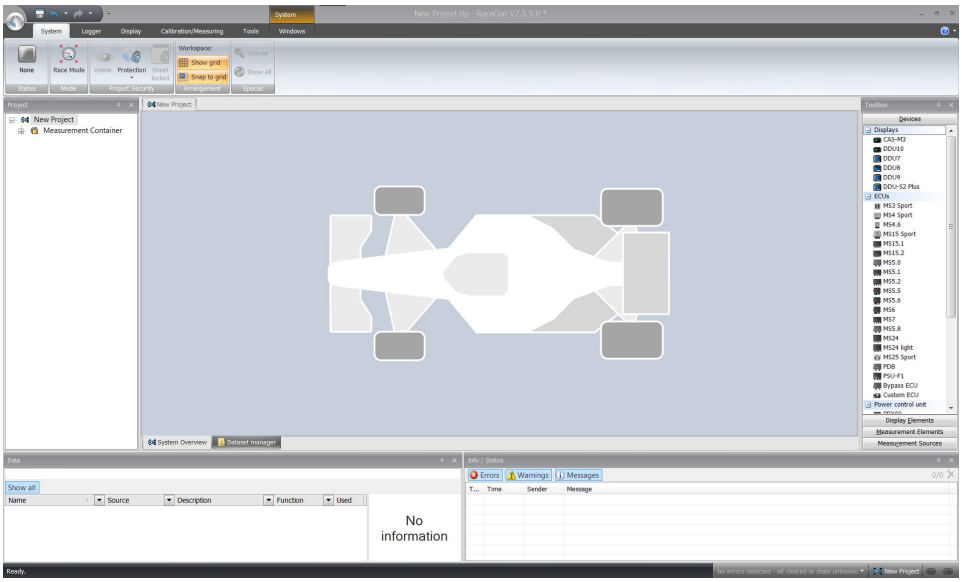
File / New / RaceCon Project opens a new project in RaceCon.



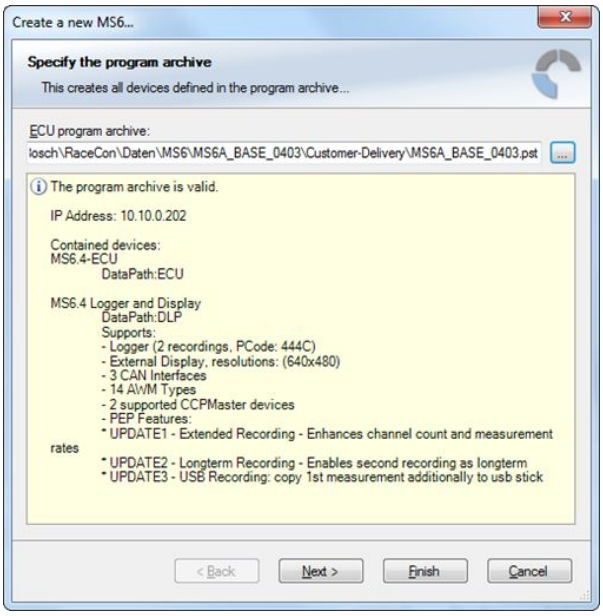
To create a new vehicle configuration, the devices can be pushed via drag & drop from the toolbox to the vehicle. Then they are part of the project and can be configured.

Select an ECU model MS 6 EVO from the Toolbox / Devices / ECUs.

Drag the ECU icon with pressed left mouse click on the vehicle view, then a dialog opens.

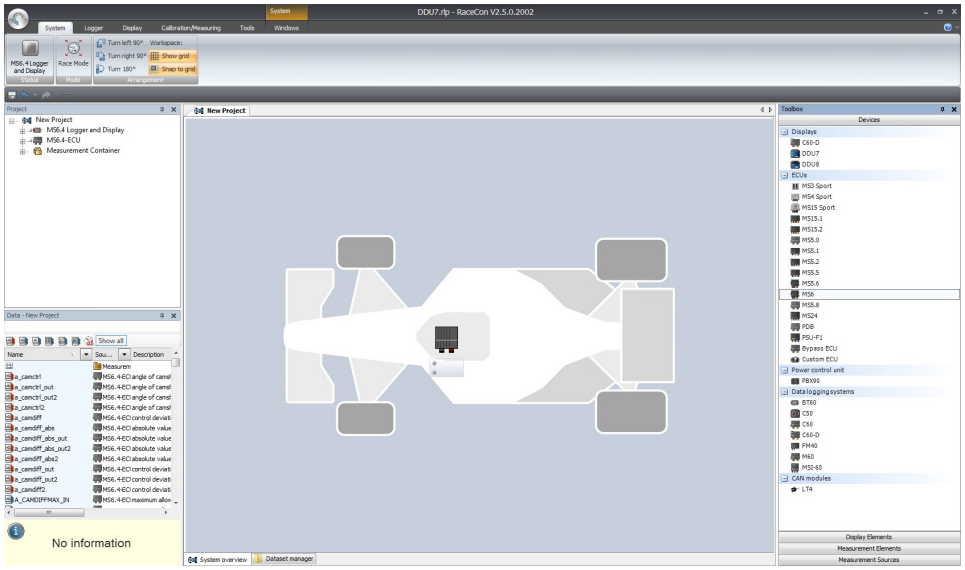


Now the ECU program archive PST files must be selected. These archives are delivered by Bosch or are available at Bosch Motorsport homepage. Specify the MS 6 EVO program archive: MS6B_XXX_xxx.pst.



Access to all configurable data is now available.

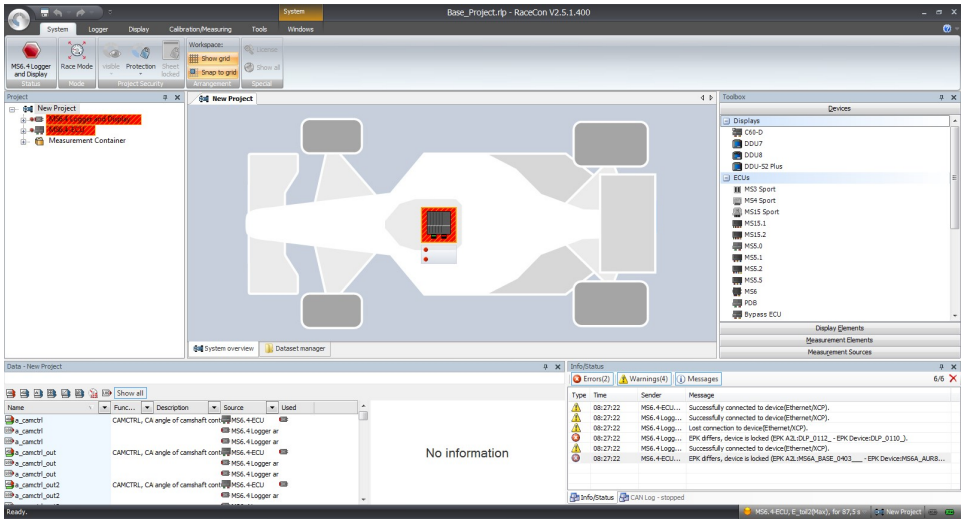
Installation may now be saved as customer project for further data application.



3.2.2 Programs Installation

Going Online for program and license configuration

In the project tree both parts of the MS 6 EVO core are shown as >red<, means MS 6 EVO device and RaceCon project differ in the used program version.

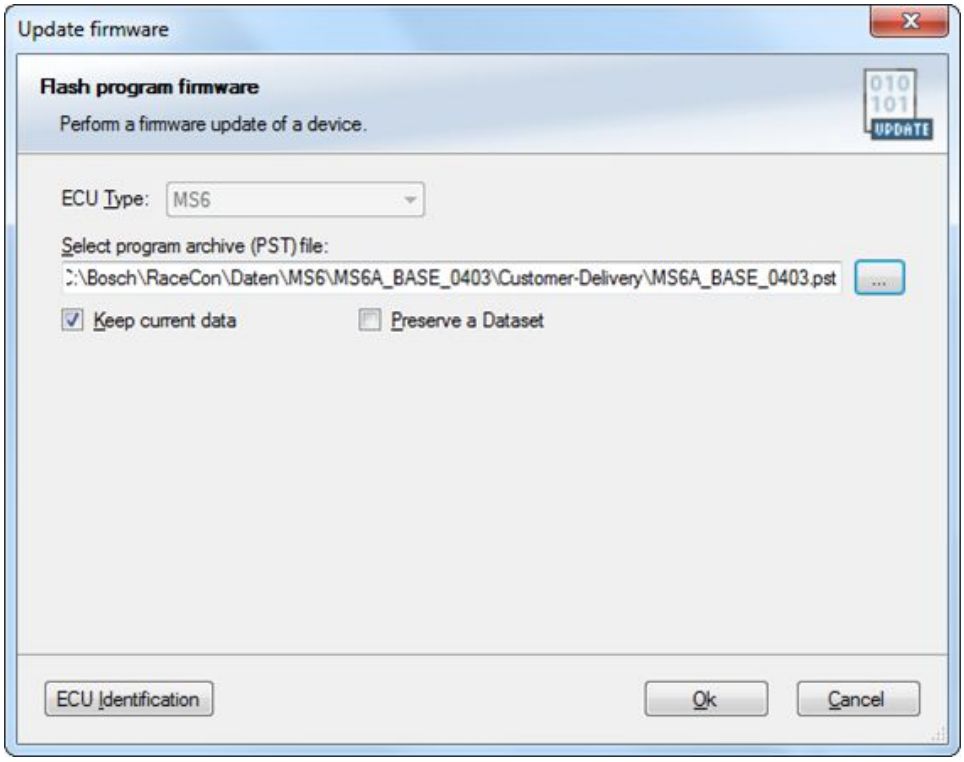


Synchronize MS 6 EVO and RaceCon program version / update the firmware of the device:

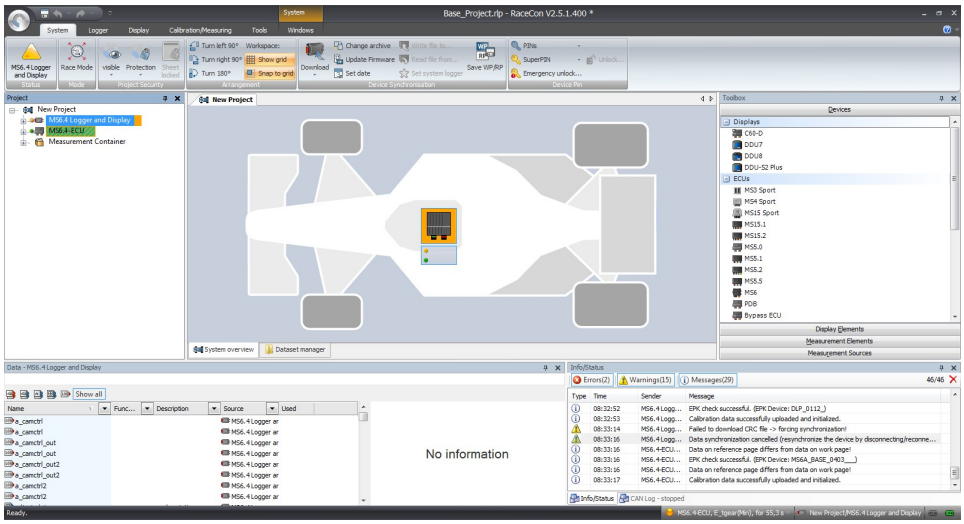
Project-tree / right mouse button to one of the red MS 6 EVO core / synchronize / update firmware >select customer software of the MS 6 EVO (file with extension: -.pst)

NOTICE

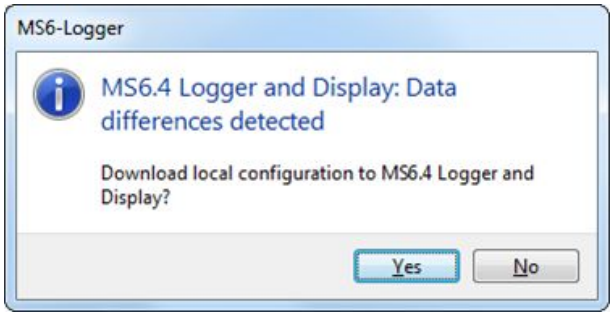
Do not interrupt flash process.



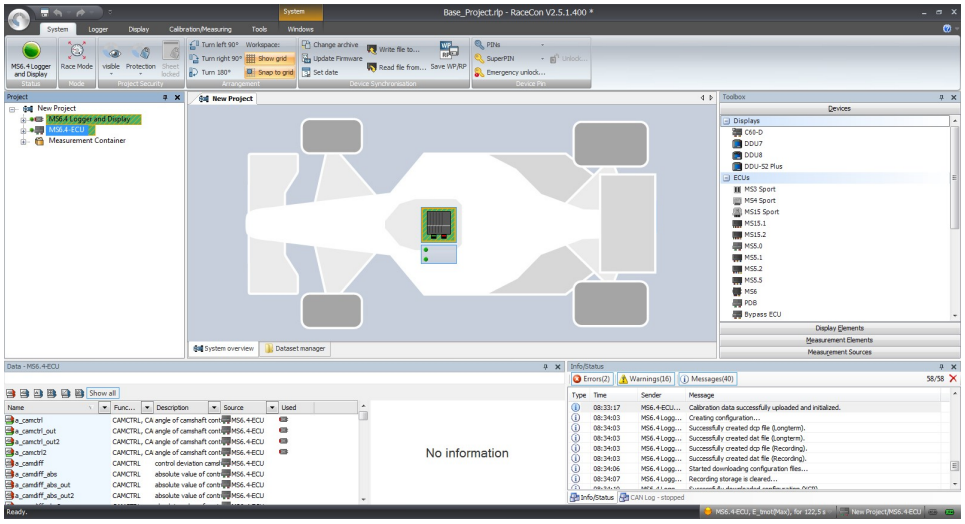
In the project tree, the MS 6 EVO logger core is shown as >yellow<, means the firmware of MS 6 EVO device and project are identical, but the data differs.



The offline preconfigured data have to be sent to the MS 6 EVO. Option one, select: Project tree / right mouse button to the yellow MS 6 EVO core / synchronize / or follow the RaceCon menu:

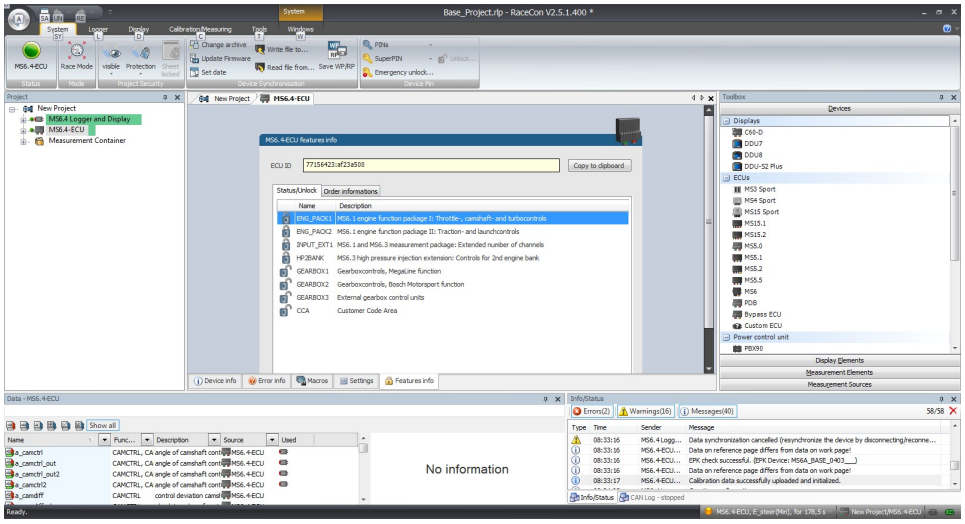


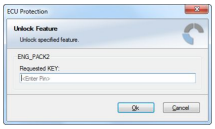
Both MS 6 EVO cores are shown as green, means firmware and data of device and project are now identical.



3.2.3 Feature/License Activation

For code area generation, additional functionalities and/or data logging licenses may be requested for activation. Generally all MS 6 EVO licenses are related to one specific device and the delivered code is only to activate for this ECU. Both cores, MS 6 EVO ECU and MS 6 EVO logger, content own license structures. Double-click to the core symbol at the project and choice features info. Select the license feature and activate the functionality using the related license code.





The licenses for gearbox and engine controls are to activate at the MS 6 EVO ECU core. The licenses for USB or logger packages are handled in the MS 6 EVO logger core. MS 6 EVO ECU is now ready for customer data and use.

4 Prepare Data Base

Using RaceCon, the data base is already generated and the modification may start immediately. For information, please see RaceCon manual.

4.1 Initial Data Application

The following chapter deals only with the main parameters which should be checked before a first engine startup. Several functions are recommended to be switched off; many software labels will not be explained in detail. To work on these functions and labels after the first startup, please refer the full-scope function description. The offline data application guide shall help to get the engine started the first time without problems.

NOTICE

Wrong engine setup data may lead to serious engine damages.

4.1.1 Basic Engine Data

The MS 6 EVO system can be used for engines up to 12 cylinders. Please ensure that the correct software variant is loaded in your ECU. Define the engine parameters like number of cylinders, firing order, injection system, and cam- and crankshaft designs in relation to TDC.

4.1.2 Crank- and Camshaft Wheel

The system initially supports wheels with 60-2 teeth. Other configurations **in** the limits between **30- and 60 teeth** may be possible to configure also. Please refer also to the chapter Ignition Trigger Wheel.

Main Data Labels to configure for crank- and camshaft wheel	
CRANK_TOOTH_CNT	Number of teeth of the flywheel (including the missing teeth) **(limited to 30-60 teeth)**
PIN_IN_CRANK	Selection of used crankshaft input pin
CWINTF_L43_L44	Selection of used crankshaft sensor type (Hall or inductive type), example for used pins L43/L44
CRANK_GAP_TOOTH_CNT	Number of missing teeth on the flywheel
PIN_IN_CAM_x	Selection of used camshaft input pin
CAM_MODE	Camshaft position detection mode
CAM_TOOTH_CNTx	Number of teeth on the camshaft
SYNC_CAM	Camshaft signal used for engine synchronization

4.1.3 Initial Steps

The following data must be set initially to start injection calibration for the first time.

Main Data Labels to configure for firing order and engine design	
DISPLACEMENT	Displacement of all cylinders
CYLBANK	Cylinder allocations bank 1 or bank 2 Example typ. 8 cyl. engine: Cylinder 1 2 3 4 5 6 7 8 9 10 11 12 CYLBANK 1 1 1 1 2 2 2 2 0 0 0 0 Engines with one Lambda sensor (e.g. 4-in-a-row) run as 1-bank-system. Set CYLBANK to 1.
CYLNUMBER	Number of cylinders
CYLANGLE	Angle of cylinder TDCs relative to reference mark (RM → TDC)
QSTAT	Static valve quantity for n-heptane in g/min (injectors are typically measured with n-heptane)
MP_TDTECORR	Injection valve delay correction map, low pressure
TECORPRAIL	Rail pressure correction for injection time

4.1.4 Basic Path of Injection Calculation

The ECU MS 6 EVO is a so called physically based system. This means that corrections are made according to their origin influence (e.g., air temperature, fuel pressure etc.). For it, the initial engine load signal (throttle angle ath) or the engine charge signal rl (relative load) is defined as 100 % if the cylinder is filled with air of 20°C and 1013 mbar ("standard condition"). Corrections related to the air path (air temperature, ambient pressure) are therefore performed to this value rl. Based on this central value most of the relevant ECU signals are calculated, first and foremost injection and ignition.

Due to this constellation changes in the air path are centrally considered for all following functions, independently whether they are caused by ambient influences, mechanical changes of the intake system or even a change from alpha/n-system to p/n-system.

Using this rl value, a relative fuel mass rfm is constructed. For an operating point of rl = 100 %, a fuel amount of 100 % is needed if the desired Lambda = 1. All corrections to the desired fuel quantity like start enrichment, warm up factor, transient compensation, but also the desired Lambda value and the correction factor of the Lambda control are considered as an adjustment of this relative fuel mass. I.e., all corrections are still made independently of the size and other specifications of the injectors.

Next step is the conversion of the relative fuel mass to a desired injection time te. Here the engine´s displacement, the fuel flow through the injector and influences of the fuel pressure are considered.

Finally, the actual duration of the control pulse ti is calculated, considering pick-up delays of the injectors, fuel cutoff (e.g., overrun cutoff, speed limiter, gear cut) and cylinder individual correction factors. Please refer also to the system overview in the Function Description ECOV.

4.1.5 Main Data Labels to configure for Engine Start up

Main Data Labels to configure for engine start up	
MP_MIXCORR	Mixture correction, set to 1.0 for startup
MIXCORR_APP	Global factor for mixture correction, set to 1.0 for the begin of startup
CWPRAILCOR	If a correction by fuel pressure is intended, set = 1. In this case please set PRAILREF according to the referenced fuel pressure. Also refer to MP_P22MOD. Usually the predefined values are suitable. If unsure, set CWPRAILCOR to 0 for first startup.
FINJ_WARMUP	Correction via engine coolant temperature. Usually the predefined values are suitable. Ensure, that for coolant temperatures driven on your dyno during calibration, no warm up factor applies (i.e. FINJ_WARMUP is 0.0 for this temperature).
MP_LAM_MP1	Desired Lambda value, valid for map position 1. According to your expectations, e.g. 0.9. For alternative positions of your map switch, the maps MP_LAM_MP2 (3) or (_PACE) apply, therefore ensure correct switch position

4.1.6 Main Data Labels for Load Calculation

Main Data Labels for Load Calculation	
CWLOAD	Decision between alpha/n or p/n related load calculation
CWLOADP1	Decision between P1 and ambient pressure
FRLTINT	Correction via ambient temperature. Usually, the predefined values are suitable. If unsure, set FRLTINT to 1.0 for first startup.
alpha/n system	
MP_RL	Relative load depending on throttle angle and engine speed. Set value until your desired Lambda is matched.
MP_FRLPLOAD	Correction via intake air pressure
p/n system	
FRLPTHR	Factor to throttle dependence. If unsure, set to 1.0 for startup.
MP_RLP1 ...P4	Relative load depending on throttle position 1-4
PALTCOR	Altitude correction for relative load. If unsure, set PALTCOR to 0.0.
MP_RL	Relative load depending on throttle angle and engine speed. Set value until your desired Lambda is matched.
Notice: For details, please refer to the Function Description LOADCALC.	

4.1.7 Main Data Labels for Injection

Main Data Labels for injection	
CWINJMODE	Choice of injection system:

Main Data Labels for injection	
	– Low Pressure – Low Pressure plus Far Bank – High Pressure – High Pressure plus Far Bank
CWINJANGMODE	Choice of angle of injection relation
MP_AOINJ	Map begin/end of injection
Notice: Before calibration starts, turn off Lambda closed loop control.	
CWLC	Codeword for enabling of the Lambda closed loop control. Set to FALSE during initial calibration, afterwards TRUE.

4.1.8 Main Data Labels for Ignition

The MS 6 EVO provides two alternatives to drive the ignition coils: For engines up to 8 cylinders the internal powerstages may be used. Alternatively, or for engines up to 12 cylinders external powerstages may be used.

IGNDRV_TYPE	For ignition coils with integrated powerstage set IGNDRV_TYPE to 0 („External PS (CK200)“). To use the ECU’s internal powerstages (for ignition coils without integrated powerstage), set IGNDRV_TYPE to 1 (“Internal PS”) The ECU must be restarted for changes to take effect.

Main Data Labels for ignition	
Notice: Positive values stand for ignition angles before TDC, negative values after TDC. Begin with moderate values to protect your engine from damages.	
MP_TDWELL	Coil dwell time. Consult the coil manufacturer for details. Most coils need dwell times about 1.5 to 2.5 ms at 12 to 14 V. For further background information please refer to the Function Description IGNITION.
DIGN_CYL1-…12	Cylinder individual corrections. Set to 0.0. Numbering refers to mechanical cylinders.
MP_IGN_START/DIGN_ST_TINT	Base spark advance during engine start. Set to 5 to 10 deg, according to the requirements of the engine.
MP_IGN(2/3)	Base ignition timing in deg crankshaft before TDC. Use modest values at the first time. Atmospheric engines may run safe at 20 to 25 deg in part load, turbo engines at high boosts may demand even less spark advance. These values are strongly dependant on compression ratio, fuel quality, temperature, and engine specifics. If you know you’re using “poor” fuel, run at high temperatures or your engine is very sensitive on spark advance, go to the safe side.
MP_DIGN_TEMP/MP_DIGN_TEMPW	Ignition angle temperature dependent

Main Data Labels for ignition	
DIGN_APPL	Delta value for spark advance, use for application work. Start at 0.0 for first startup.
IGN_IDLE_STAT	Ignition timing during idle. 10 deg are suitable for most applications
NIDLE_NOM / DIGN_IDLECTRL	Desired engine idle speed for idle stabilization. Set value to desired speed or deactivate stabilization by setting DIGN_IDLECTRL to 0.0.

4.1.9 Main Data Labels for Engine Speed Limitation

The rev limiter works in two steps:

- Soft limitation by ignition retardation or cylinder individual cutoff of injection and/or ignition
- Hard limitation by injection cut off and/or ignition cutoff of all cylinders

To achieve a good dynamic behaviour by advanced intervention, the engine speed is predicted by means of the speed gradient.

Main Data Labels for engine speed limitation	
CWNMAX_CUTOFF	Codeword for type of intervention during soft limiter: – no cut-off – injection cut-off – ignition cut-off – injection and ignition cut-off
CWNMAXH_CUTOFF	Codeword for type of intervention during hard limiter: – injection cut-off – ignition cut-off – injection and ignition cut-off
NMAX_GEAR	Engine speed limit, gear dependent
NMAX_P	Determines the slope of the soft limiter between soft limit and hard limit. Predefined. Vary according to your engine´s dynamic behaviour.
TC_GEARNMAXPR	Prediction time for rev limiter, depends on the inertial torque of the engine. If oscillations occur, reduce value, or turn off by setting = 0.0.

4.1.10 Main Data Labels for Cutoff Pattern

Cutoff Pattern	
MP_COPATTERN	Defines the appropriate cylinders for torque reduction by cylinder cutoff. At the beginning of an intervention the next possible cylinder for starting the cutoff pattern is determined. Based on this info the actual pattern is taken out of the map.

Cutoff Pattern	
	Pattern should be defined in view of minimized oscillations of the crankshaft. Usually, a regular distribution of firing and non-firing cylinders leads to the best result. However, investigations of the individual engine are recommendable. For it, cutoff pattern can be also turned on manually via CUTOFF_APP and CWCUTOFF_APP Example: 4-cylinder engine Start Cyl./Cutoff stage 1 2 3 4 1 1 (=0001b) 2 (=0010b) 4 (=0100b) 8 (=1000b) 2 9 (=1001b) 6 (=0110b) 6 (=0110b) 9 (=1001b) 3 11 (=1011b) 14 (=1110b) 7 (=0111b) 13 (=1101b) 4 15 (=1111b) 15 (=1111b) 15 (=1111b) 15 (=1111b) The cylinders are assigned bitwise, the lowest bit represents cylinder 1. Numbering refers to mechanical cylinders, e.g. pattern = 9: Mechanical cylinders 1 and 4 are fade out.
CUTOFF_APP	Cutoff pattern for test purposes. Bit representation as described at MP_COPATTERN
CWCUTOFF_APP	Set Codeword for type of intervention during test cutoff: – disabled – Injection – Ignition – Injection and Ignition
Notice: This option is also useful for searching a misfiring cylinder. Select one cylinder after the other during test cutoff and watch your engine.	

4.2 Peripherals

Sensors and peripherals can be checked when the system is powered up electrically.

Do not start the engine before all steps in this chapter are carried out.

NOTICE

Make sure the battery is connected properly, all sensors are connected, and ground wiring is fixed before powering up the system. Check all sensors for errors (E_...) and reliable measure values before starting the engine.

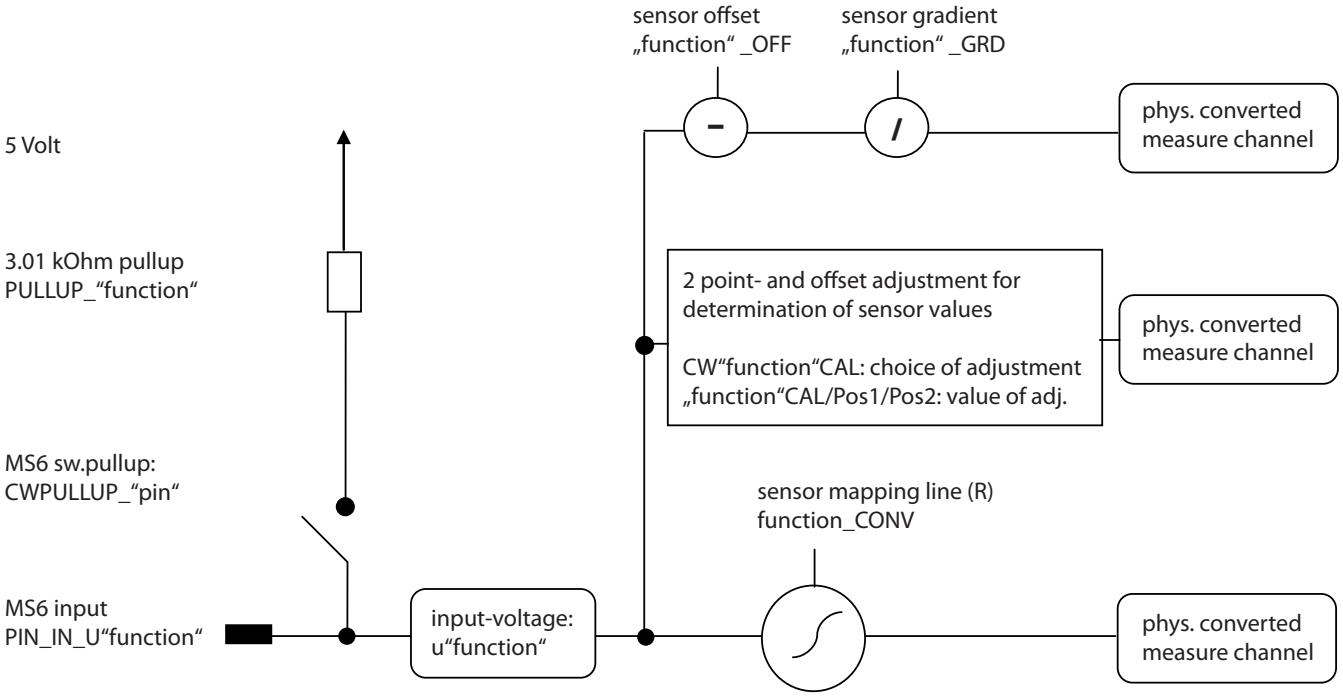
Sensor configuration

The MS 6 EVO has the option to link a lot of functionalities to a possible hardware input. The chapters “ECUPINS, SWITCHMATRIX and Input Signal Processing” of the functional description explains the details. All functions of Base MS 6 EVO programs are linked like described in the MS 6 EVO documents (e.g. function description ADC_ECU_MAP) or the wiring diagrams.

Analogue sensor inputs

The physical way of conversion from sensor signal voltage to physical values follow the same structures. The hardware input may be connected to different kinds of pull-up options. Inputs with fixed 1.47 kOhm or 3.01 kOhm pull-up resistors are prepared to handle passive sensor elements, for instance temperature sensors with integrated resistors (NTC- or PT100 sensors). Inputs without any pull-up resistors are prepared to handle active sensor elements, which deliver 0 to 5 V signals, for instance pressure-, potentiometer- or acceleration sensors. Inputs with switchable 1.47 kOhm pull-ups are designed to handle mainly active sensors with disabled pull-up, but are prepared for future measuring of digital signals. Inputs with switchable 3.01 kOhm pull-ups offer the most options and are recommended to link after the standard sensors are connected. The pull-up resistor itself is not modifiable and for better measure results may be, the version of sensor/mapping line has to be changed. To activate the Pin-Selection, first the label “PIN_IN_function” has to be enabled. Error detection of an analogue input signal detects short cuts to ground, U”function”_MIN recommended to be set to 0.2 V and short cuts to power supply U”function”_MAX recommended to be set to 4.8 V. Failure are activated after the adjustable debounce time of diagnosis TD”function”. If a sensor error is set, the output is switched to the default value “function”_DEF.

Principle structure of analogue signal conversion



Pressure measurements

The system offers many different pressure channels; please see function description input signal processing for details. For gradient and offset information contact sensor manufacturer.

Example: Ambient Pressure	
PAMB_DEF	Default value if an error occurred.
FCPAMB	Filter constant. For ambient pressure use 1 second, for other pressures choose appropriate values, ~ 100 to 200 milliseconds

All other variables are named by the same rule; replace “pamb” by e.g. “poil” to apply data for the oil pressure sensor.

Temperature measurements

The system offers many different temperature channels; please see function description input signal processing for details.

Example: Intake Air Temperature	
UTINT_MIN, UTINT_MAX	Minimum and maximum accepted sensor voltage. When violated, an error is set (E_tint = 1).
TINT_CONV	Sensor characteristic. Consult the sensor manufacturer.
PULLUP_TINT	Value of the used pull-up resistor. If only the ECU´s pull-up is used (standard case). Keep the predefined value of 3.01 kOhm.

Thermocouples

The exhaust gas temperatures are measured via thermocouple elements, using a special evaluation circuit. Predefined values should be suitable for NiCrNi or k-type elements. For further details and project specific variants, please refer to the function description.

Digital sensor inputs

MS 6 EVO digital sensor inputs used for frequency measurements can be configured for different sensor types.

CWINTF_A047_A048	Selection between Hall effect or inductive sensor for flywheel measurement, related to MS 6 EVO contact A047 (use ground A048 if inductive type is selected).
CWINTF_K045/K046	Selection between Hall effect or inductive sensors for frequency measurements, like turbo- or driveshaft speeds, related to MS 6 EVO contacts K045 or K046 (use ground K062 if inductive types are selected).
CWINTF_A049/A050/A051/A052	Selection between Hall effect or DF11 sensors for frequency measurement, like wheel speeds or cam position detection, related to MS 6 EVO contacts A49, A50, A51 or A52.

4.3 Throttle Control

The system supports mechanic and electronic throttle controls.

Using an MS 6 EVO device, respect the necessary license for electronic throttle is activated. Electronic Throttle Control is a safety-critical function. Depending on specific use and/or construction, the safety functions, fault detections and fault responses of the ETC system may differ in several points from ETC systems used in series production. Hence before each vehicle-commissioning the system must be checked for accuracy and faultlessness.

The customer is responsible for the activation of all ETC-relevant diagnosis and for their correct parameterization. By disregarding this information, the functionality of the ECU and the safety cannot be ensured.

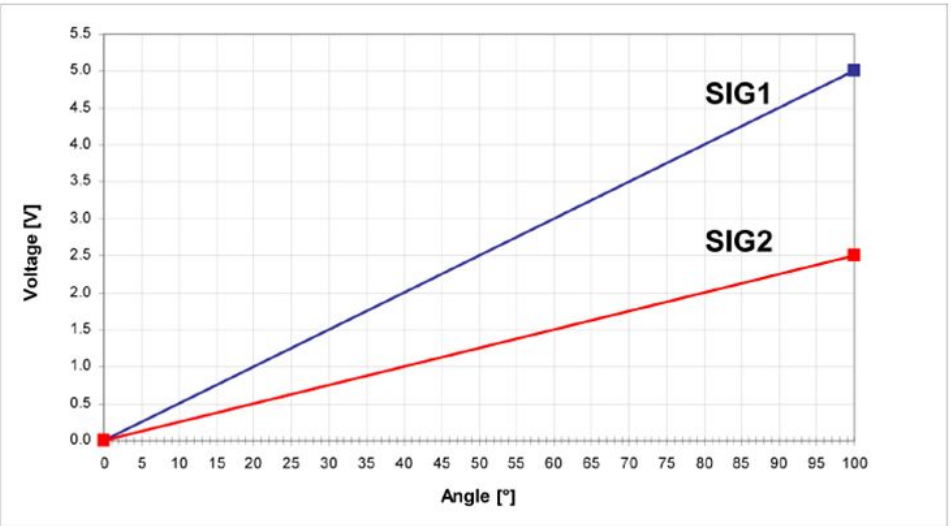
Notice: For detailed information see function description ETC

The usual route of ETC determines the drivers input measuring the pedal position and transferring this leading signal via functionality options into the control of an electrical throttle actuator. Pedal- and actuator positions are generally measured in a secondary redundant way to verify the reliability of the function. To activate the system, first verify the signal tolerances and error messages by moving acceleration pedal and throttle actuator manually. An inactive system usually is the result of inverted wired sensor signals or actuator controls. Calibrate the pedal- and throttle positions.

Verification of acceleration pedal signals:

The mathematic value of voltage pedal signal 1 - 2*voltage pedal signal 2 has to be below 0.5 V or below value of “UAPSCM_MAX”.

Signal principle of an acceleration pedal sensor:



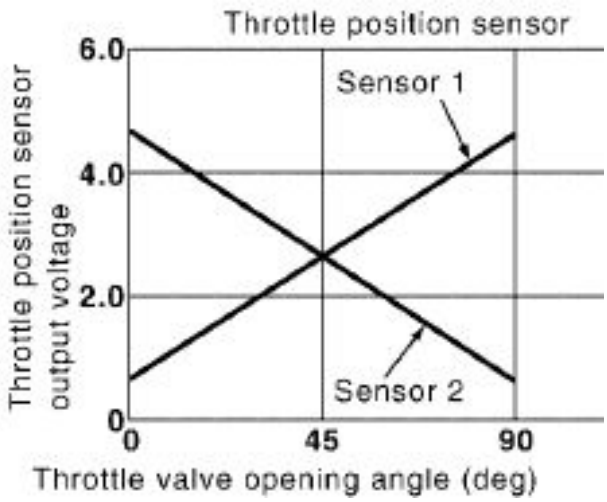
uaps_a	Voltage APS potentiometer a
uaps_b	Voltage APS potentiometer b
aps	Acceleration pedal position
UAPS_MIN, UAPS_MAX	Minimum and maximum accepted sensor voltage. Set to approx. 200 mV/4,800 mV. Check if the uaps(x) outputs are changing when the pedal is moved.

CWAPSADJ	Codeword to adjust acceleration pedal signal: – calibration inactive – calibrate release pedal – calibrate full-pressed pedal
E_aps	Detected error messages of acceleration pedal functionality. If errors are detected, the ETC functionality will become inactive.

Verification of throttle position signals:

The mathematic value of voltage throttle signal 1 + voltage throttle signal 2 - 5 V has to be below value of “UDTHRCM_MAX” (recommended 0.2 V)

The signal sequences of a throttle position sensor:



Throttle position main data labels:

CWTHR	Codeword for type of throttle controls: – manual throttle, without backup sensor – manual throttle, with backup sensor – electronic throttle, single bank – electronic throttle, dual bank

Throttle position signals:

UDTHR_MIN, UDTHR_MAX	Minimum and maximum accepted sensor voltage. When violated, an error is set (E_thr = 1). Set to approx. 200 mV/4800 mV. Check if the uthrottle(xx) outputs are changing when throttles are moved
uthrottle	2 sensor output values and their redundant signals (_b). The system expect a rising voltage for the main signals and a falling signal for the redundant one.
uthrottle_b	
uthrottle2	
uthrottle2_b	

UDTHRCM_MAX	max. allowed difference between sensor output and redundant signal abs (uthrottle(x)+uthrottle(x)_b)-5V < UD-THRCM_MAX

Calibration:

CWTHRADJ	Codeword for throttle adjust: – calibration inactive – automatical calibration – calibrate 1st mech. stop – calibrate 2nd mech. stop – calibrate limp home position – wiring check/recalculate

Manual Procedure:

- Close throttle and set CWTHRADJ to 2.
- Open throttle fully and set CWTHRADJ to 3.
- Adjust the throttle to idle point.
- Do not forget to set CWTHRADJ back to 0. Check calibration by moving throttle.

4.4 Vehicle Test

Before starting with your vehicle test, some initial data should be set:

Speed & distance measurements

The signals for speed calculation may be available from different sources, like MS 6 EVO own measurement, GPS data or via CAN received information from ABS calculation. For MS 6 EVO own calculation, mechanical influenced data like number of available sensors, front wheel drive, number of detected increments, wheel circumferences, and dynamic corrections like corner speed application, a lot of functional options assist the calculation of the effective vehicle speed. Distance measure channels may be derived from speed information. For detailed information, see function description >CARSPEED<

CWWHEELCAN	Selection for car speed from CAN signal
CWWHEEL	Connected number of wheel speed sensors or -signals
CWFWD	Selection of front driven vehicle
CWSPEEDDYN	Release of dynamic speed calculation
INC_FRONT	Number of pulses per revolution of the front speed signal
INC_REAR	Number of pulses per revolution of the rear speed signal
CIRCWHEEL_F	Wheel circumference of the front wheels
	Consider dynamic increase of the tire
CIRCWHEEL_R	Wheel circumference of the rear wheels. Consider dynamic increase of the tire.

vwheel_xx	Measure channel of the individual wheel speeds
speed	Result of calculated vehicle speed
accv	Result of speed based derivation of longit-udinal acceleration
ltdist	Lifetime distance as accumulated result of speed derivation

Lap information and -functions

The necessary data application is integrated in the system configuration tool RaceCon. The wizard leads to configure the beacon input, asks for trustable limits of lap- and signal detection. Additional options for track segmentation, additional on track beacons are also available. Drag and drop the subfolder lap trigger of the measurement sources into the project and follow the wizard.

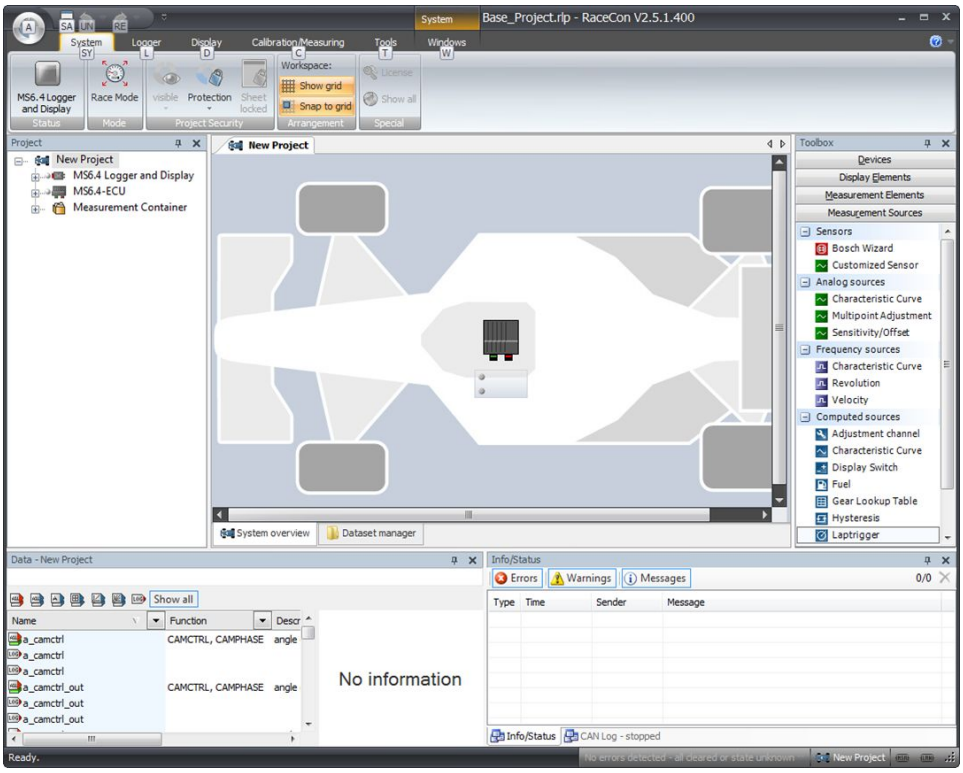


Illustration 1: Depending to the configuration, values for lap-and outing counter, lap time, segment times and differential lap- or segment times for data analysis and driver information will be created.

Laptrigger_xxxx_yy

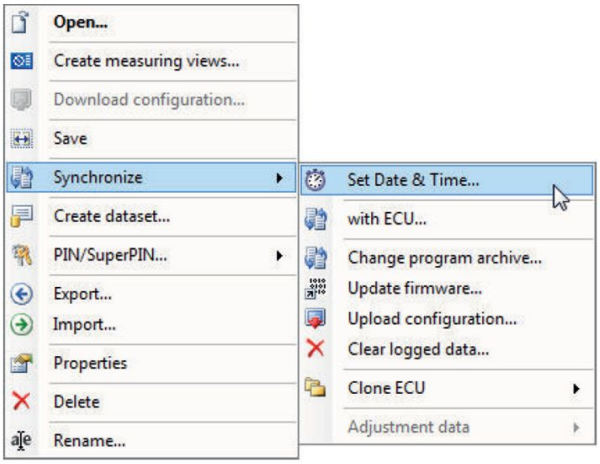
Results and measure channels of lap-functionalities

Consumption-calculation

Is designed in the same way as lap-information, drag and drop the subfolder to the pro-ject and follow the wizard.

Set Date & Time

MS 6 EVO device is equipped with a real time clock which is supplied for max. 14 days, if the ECU is disconnected from power supply. Please connect the ECU to the PC and click on “Set Date & Time” in the context menu of the MS 6 EVO



time_xx

The measure channels of the real time clock

5 ECU plus Data Logger

The MS 6 EVO combines ECU and data logger in one common housing for a cost efficient and weight optimized all-in-one solution.

5.1 Software Tools

RaceCon	Create and configure a project
	Configuration & management of recordings
	Create a new recording
	Add channels to a recording
	Create user-defined conditions for the recording
	Download recording configuration
WinDARAB	Upload recorded data
	Display and analyze the data

6 First Steps

Install the software required for the operation of the MS 6 EVO. It is developed for Windows system software. The following software versions are used in this manual:

- MS 6 EVO setup, configuration and calibration: RaceCon Version 2.9.0.7 or later.
- Measurement data analysis: WinDarab V7

Set up the 100 Mbit ethernet connection to the MS 6 EVO.

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

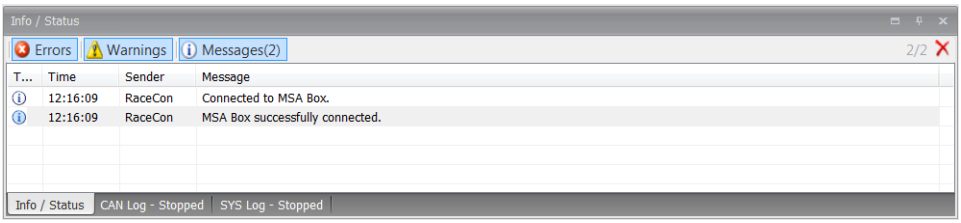
6.1 Connecting the unit to RaceCon

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 [▸ 37]”. For the status color, see chapter “Color indication [▸ 48]”.

Connection via Ethernet Cable

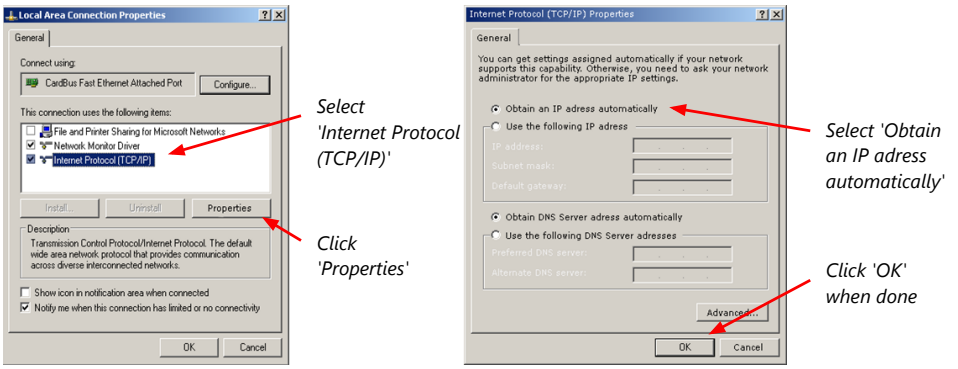
Instead of connecting the ethernet line to the MSA-Box, connect the ethernet directly to your computer.

Troubleshooting while setting up the network interface

The MS 6 EVO 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:

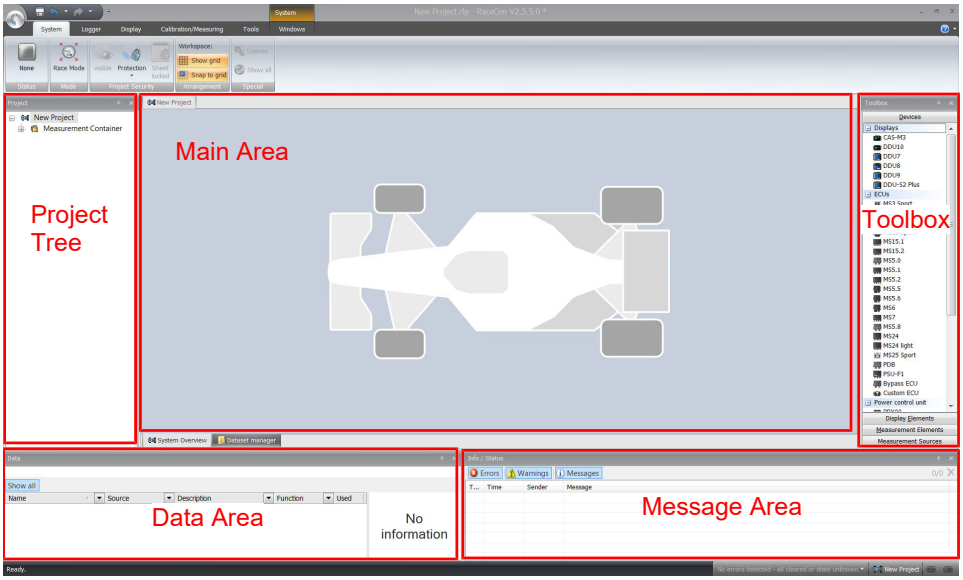
7. Switch off the PC’s firewall.

8. Reconfigure the PC or the MSA-Box network interface settings to obtain an IP address automatically as shown in the pictures below.

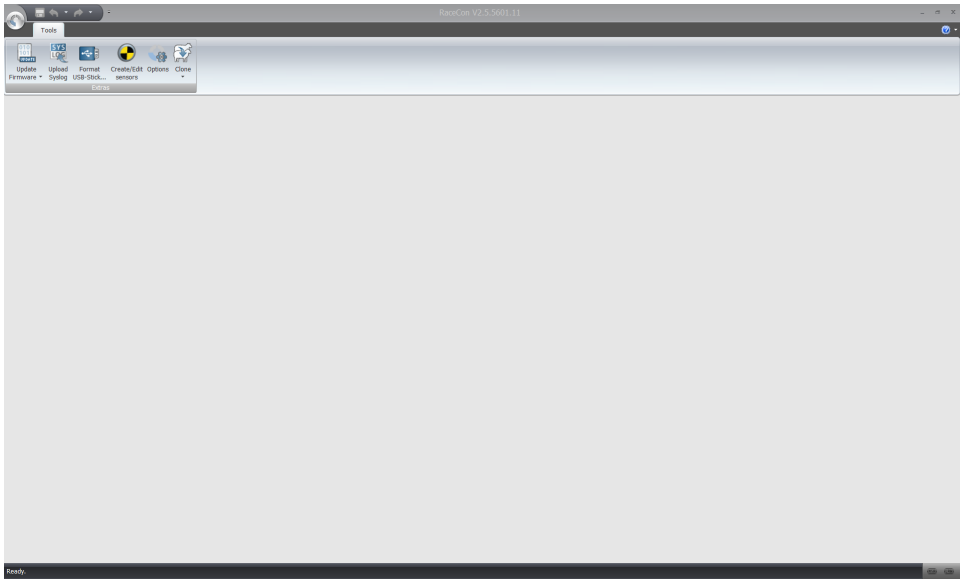


6.2 Setting up a new RaceCon Project

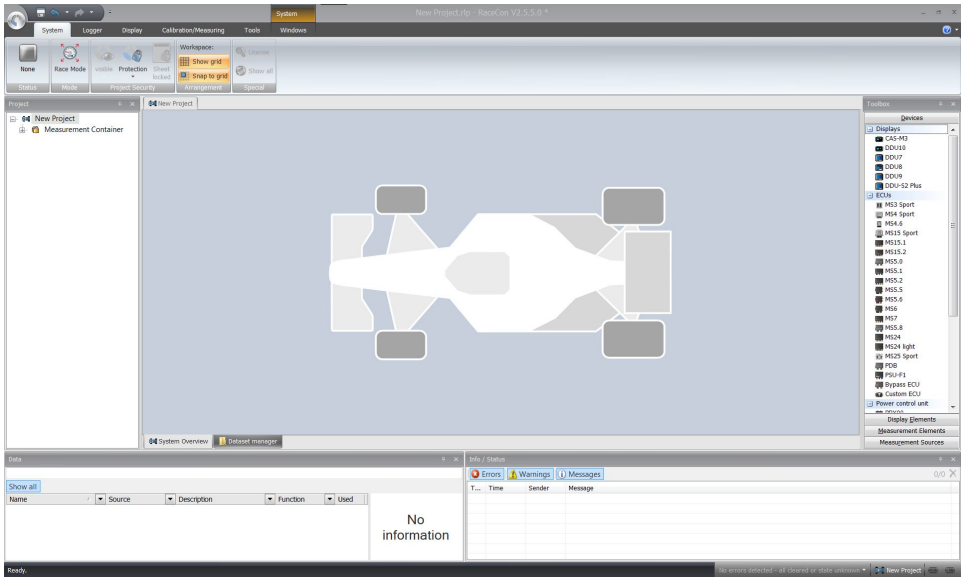
The following screenshot shows an overview of the RaceCon Main Screen with its areas. All (sub-) windows are resizable and dockable. You can find them under the 'Windows' tab.



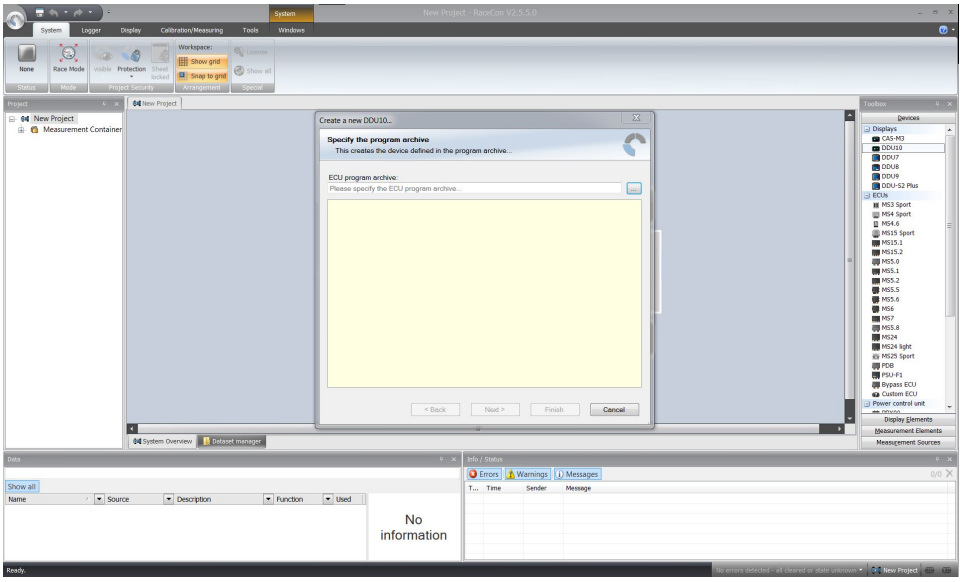
1. Start the RaceCon software.



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



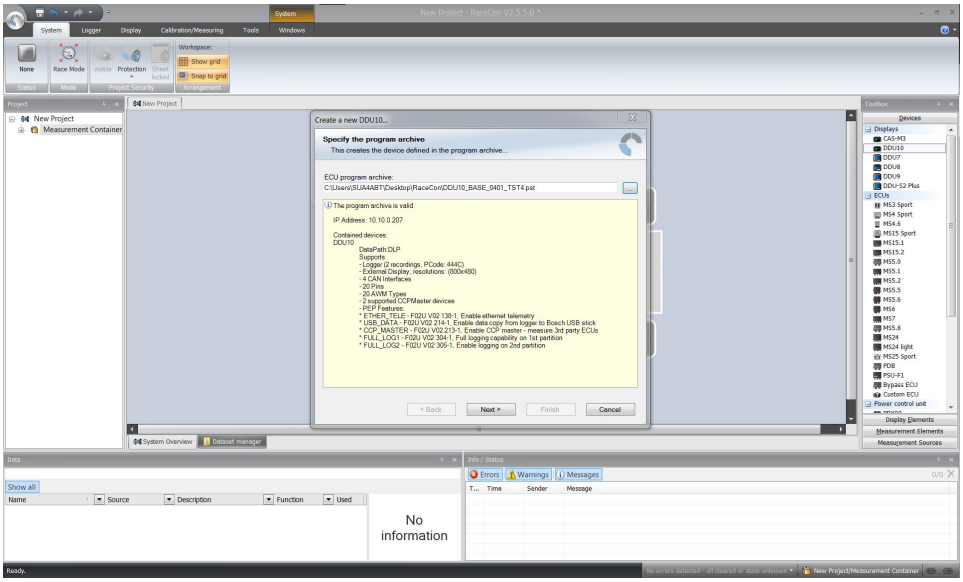
3. In the Toolbox, select the MS 6 EVO and drag it into the Main Area. A pop up window to specify the MS 6 EVO program archive appears.



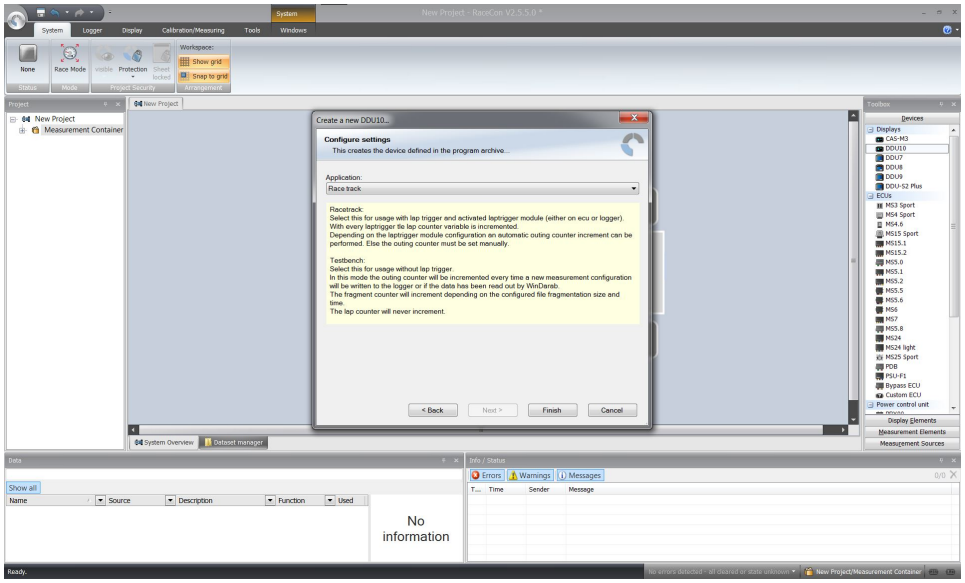
4. Download the firmware for your device:
   - from the RaceConnect project file share for PSU. This .pst file should be provided to RaceCon.
   - from www.bosch-motorsport.com for VCUs, DDUs, and Loggers.
   - You can get firmware for ECUs on request.

An information shows if the archive is valid or not.

5. Click ‘Next’ (for PSU: go on with step 7).

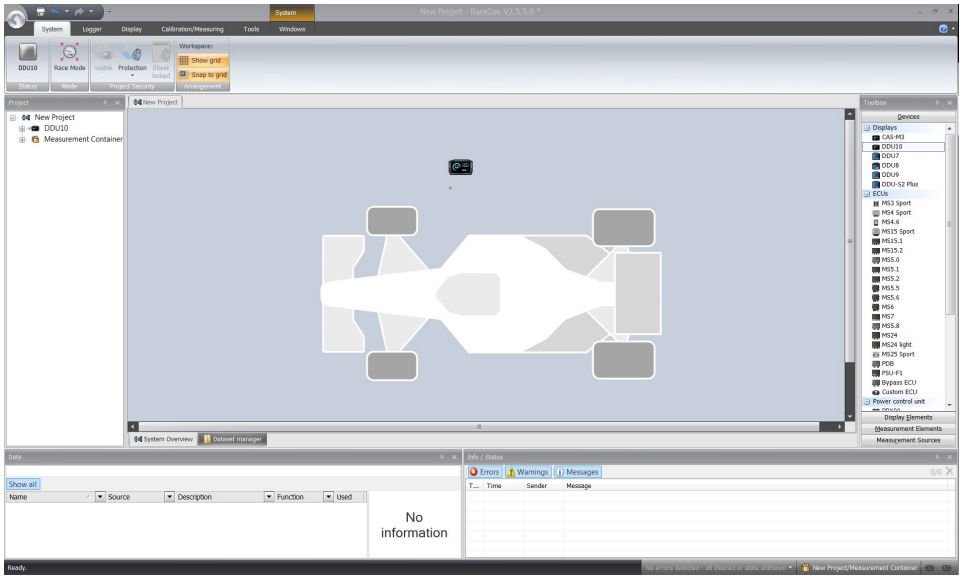


6. Select ‘Race track’ or ‘Testbench’ mode according to your application.



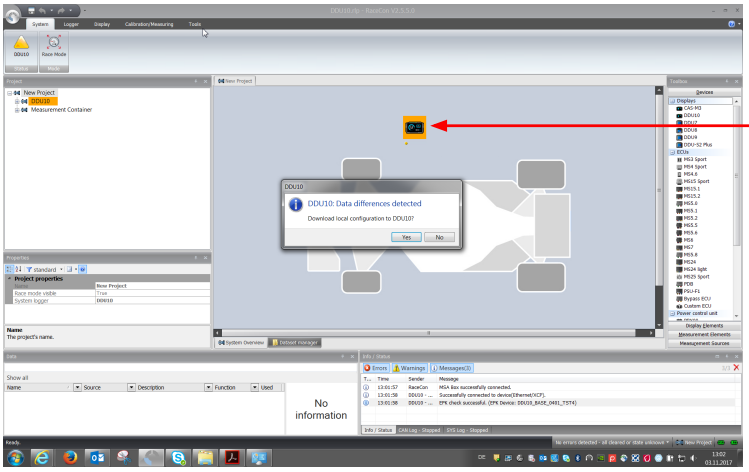
7. Click ‘Finish’.

The MS 6 EVO is inserted into the project and RaceCon tries to connect to the device.



RaceCon detects configuration differences between the MS 6 EVO and the RaceCon project and asks for permission for data download.

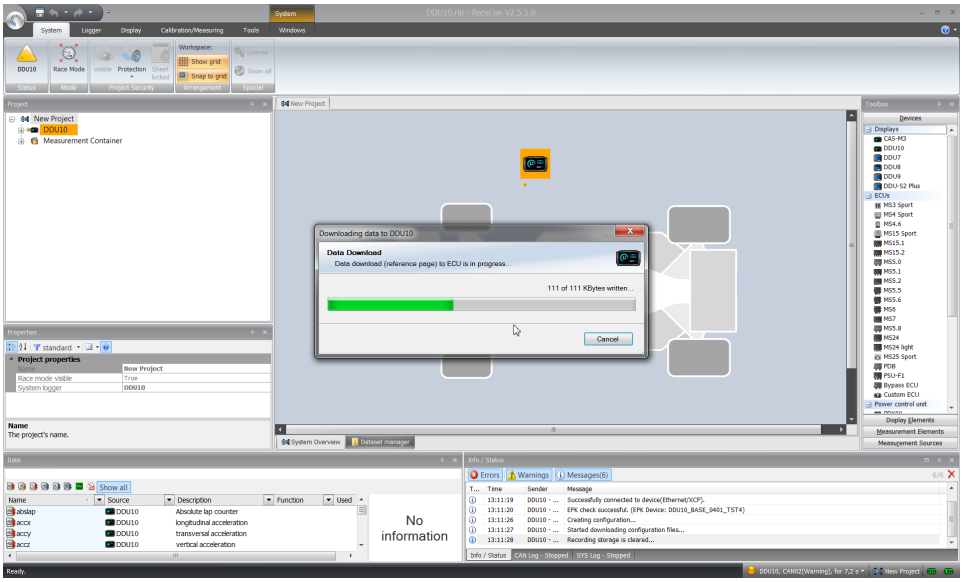
Click ‘Yes’ to download the configurations to the device or ‘No’ to continue without downloading the data.



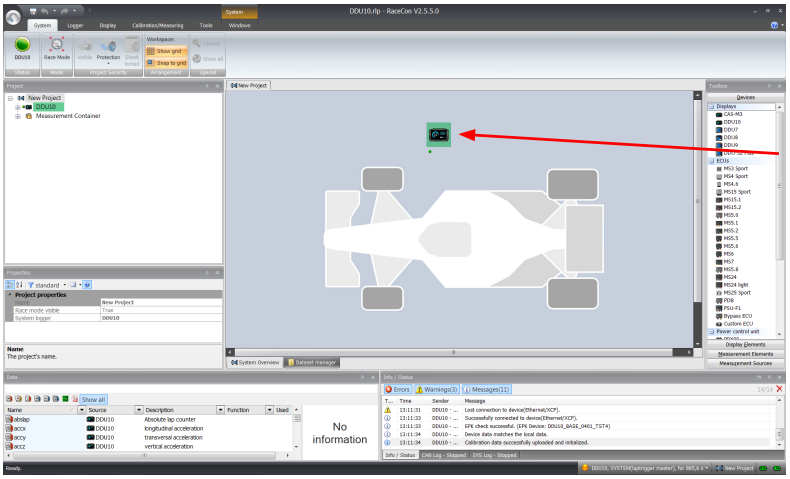
Successful connected, Device "talks" to PC.

If the device turns red, you might need to do a firmware update on the device. For more information see chapter “Firmware update [▷ 122]”.

The download starts and the MS 6 EVO carries out a reset.



After the reset, RaceCon reconnects to the MS 6 EVO. Local configuration on both the PC and MS 6 EVO match (indicated by green background and dot). The MS 6 EVO is now connected to RaceCon.



Green background and dot indicate matching configuration

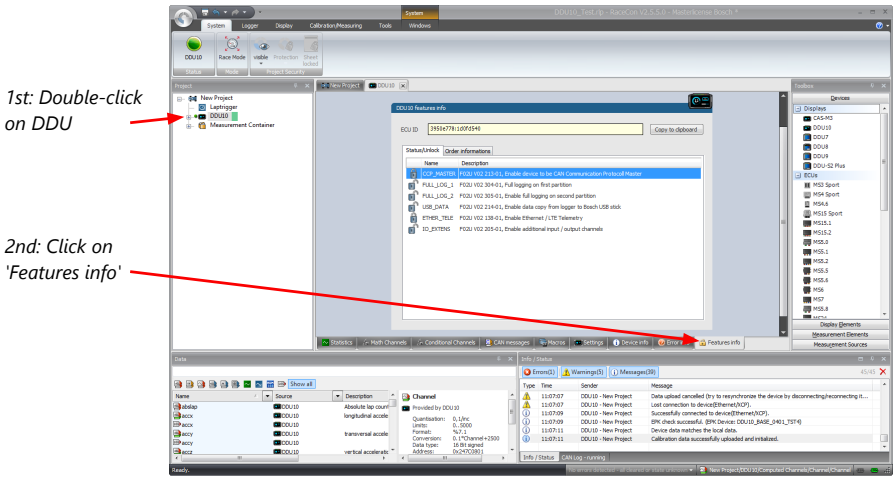
For further information on the color indication, see chapter “Color indication [▷ 48]”.

6.3 Feature activation

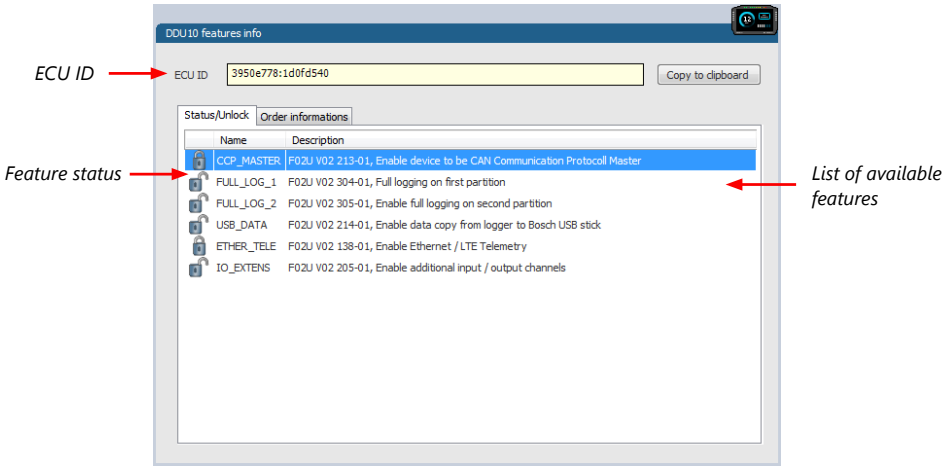
- Optional software feature packages are available for the MS 6 EVO
- All software feature packages can be purchased prior to delivery or after you have received your device.
- If you have purchased an optional software feature package, it must be activated before it becomes operational.
- The feature 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 MS 6 EVO does not work on any other MS 6 EVO.
- 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 'MS 6 EVO' in the Project Tree.
3. Click on the 'Features info' tab in the Main Area.

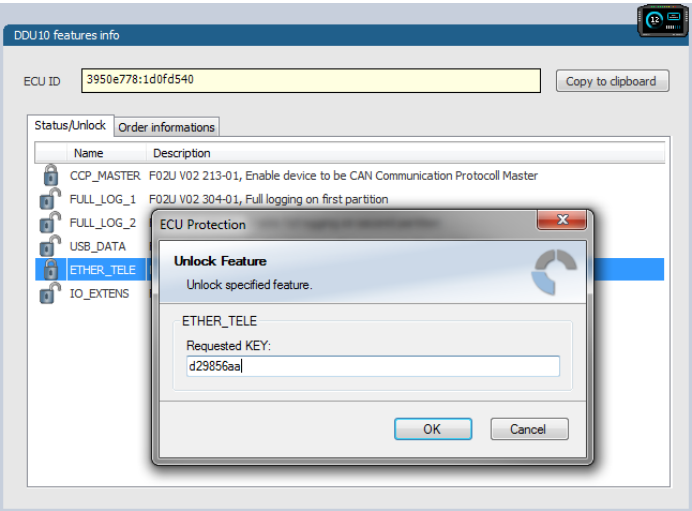


4. The 'MS 6 EVO features info' window appears.

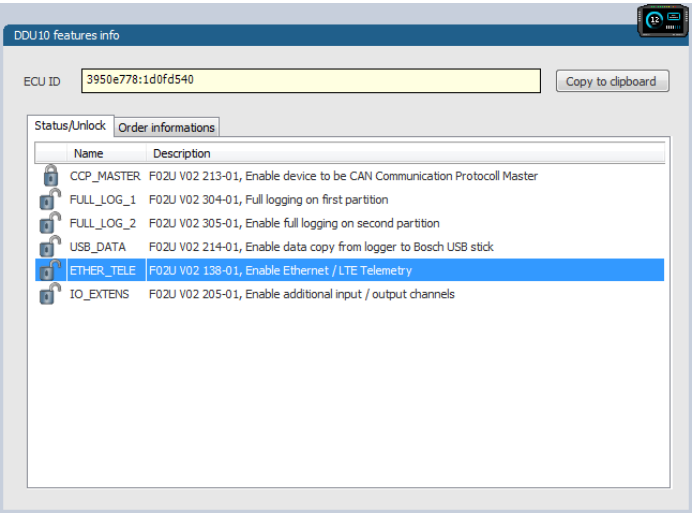


Locked (disabled) *Unlocked (activated)*

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'.

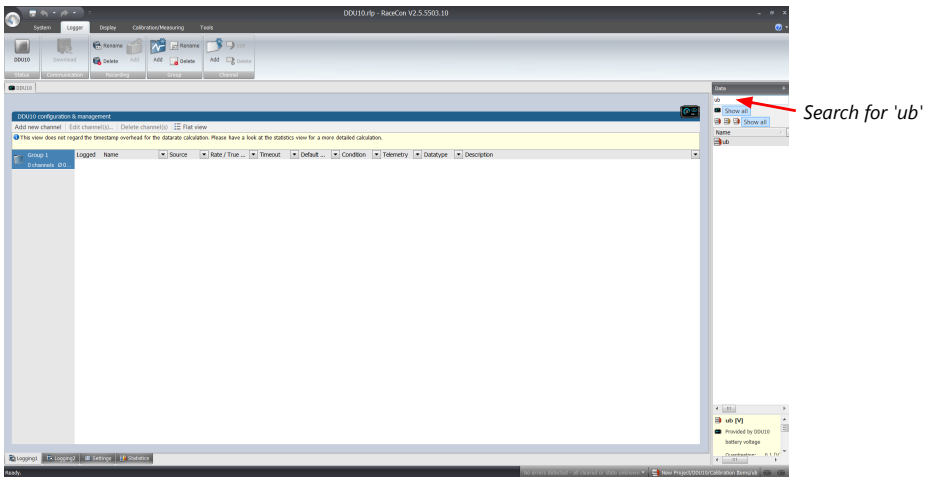


7. Perform these steps to activate other features you purchased.
8. Switch the car's ignition off and on again to cycle the power of MS 6 EVO.

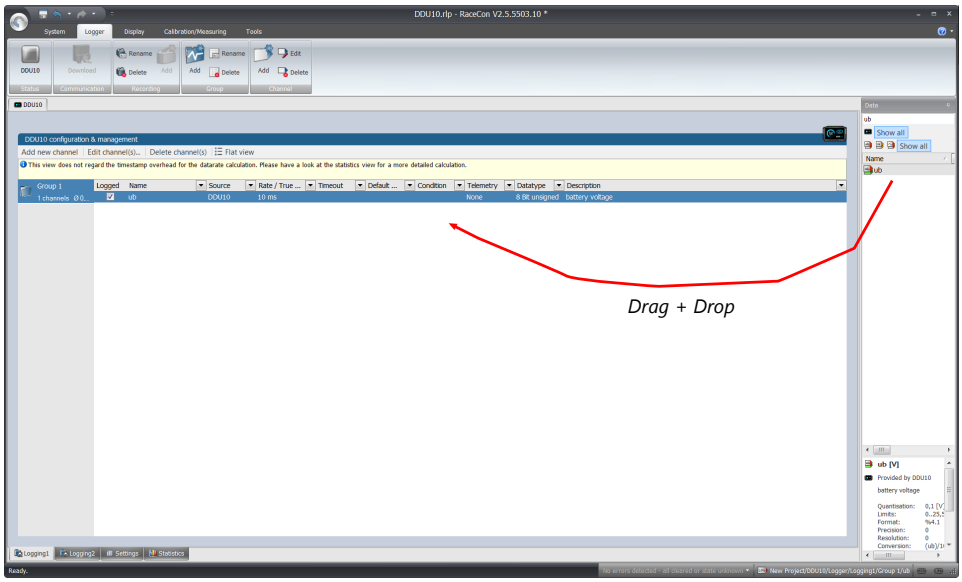
6.4 First recording (Quick Start)

This chapter explains the configuration of the recording of the battery voltage channel. See chapter 'Recording [▶ 87]' for a detailed instruction to configure recordings.

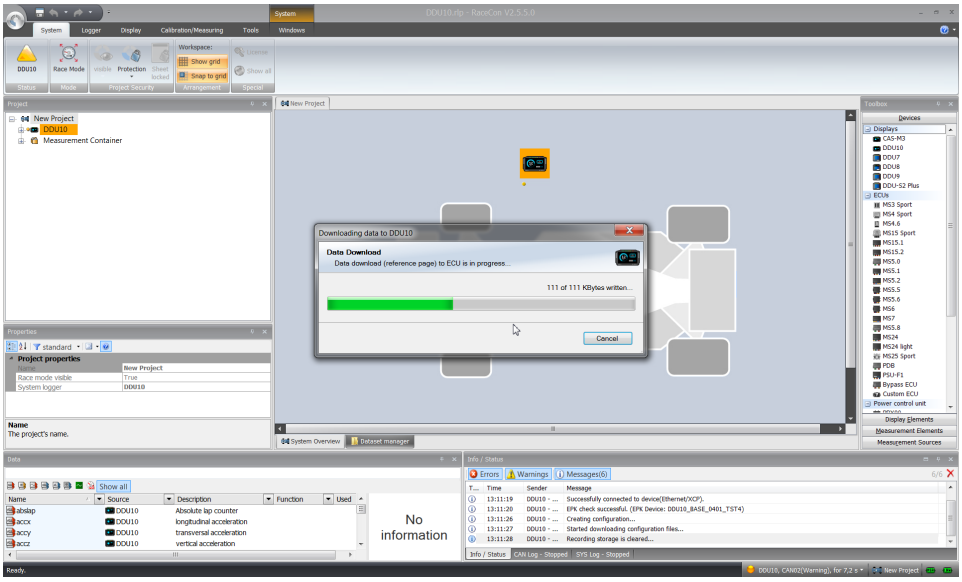
1. Click on the 'Logger' tab to go to the page 'Logger'.
2. Use the search bar in the 'Data' window, to search for 'ub' (measurement channel for battery voltage).



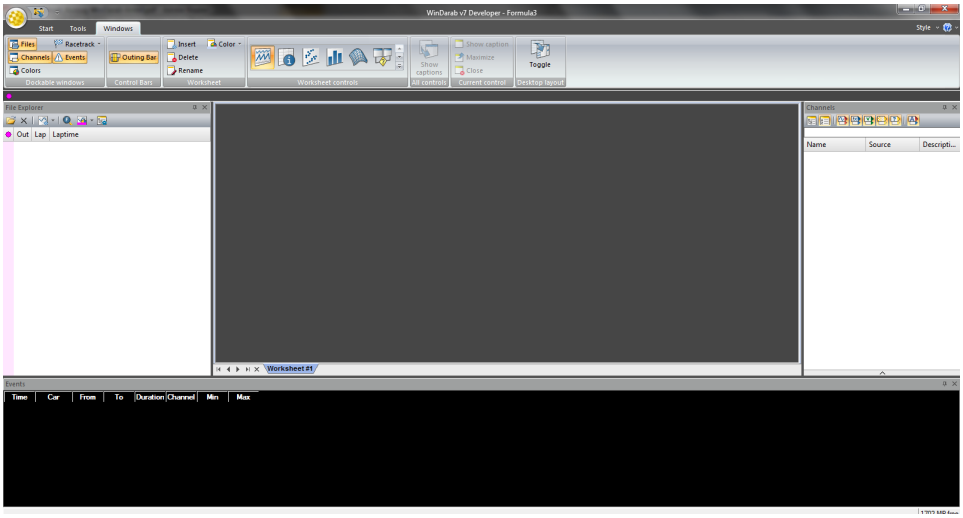
3. Drag and drop the 'ub' measurement channel into the recording area.



4. Click on the 'Download' button in the upper left corner. The configuration download starts and the MS 6 EVO 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.



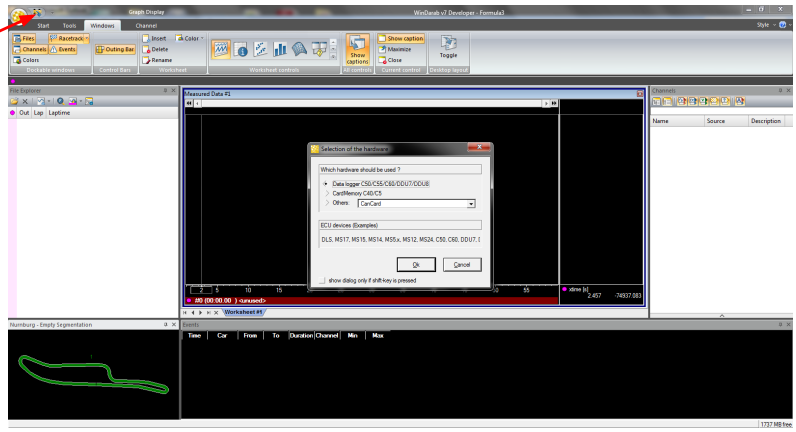
5. Start the WinDarab software.



6. Disconnect the MS 6 EVO network cable.
7. Click on the ‘Read Data from Logging Device’ icon.

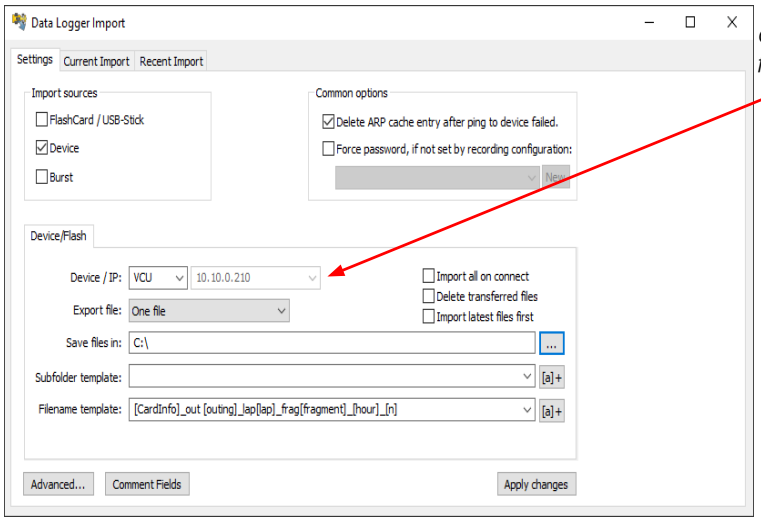
   Choose your logger and click ‘OK’ when done. The ‘Data Logger Import’ dialog opens. Refer to the WinDarab V7 manual for instructions on how to use the ‘Data Logger Import’ dialog and for more detailed descriptions and instructions.

Click 'read data from logging device'



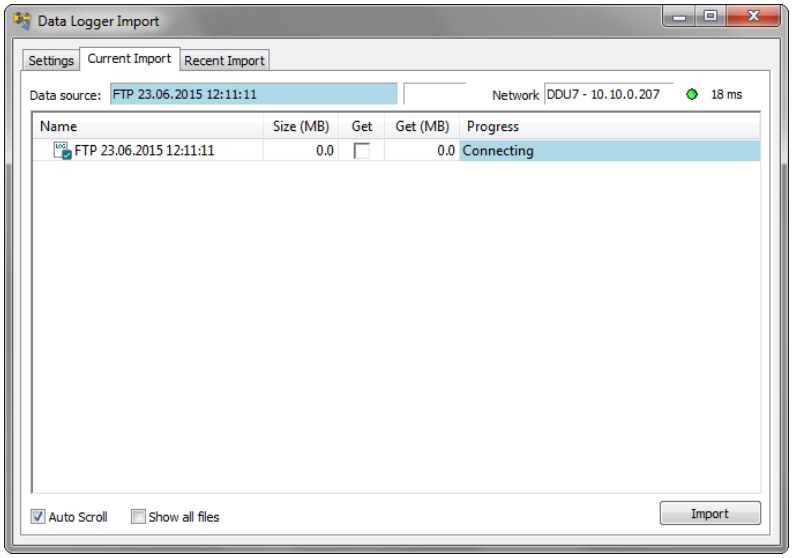
8. Choose the device and the IP address for the device.

   Click ‘Apply changes’ when done.

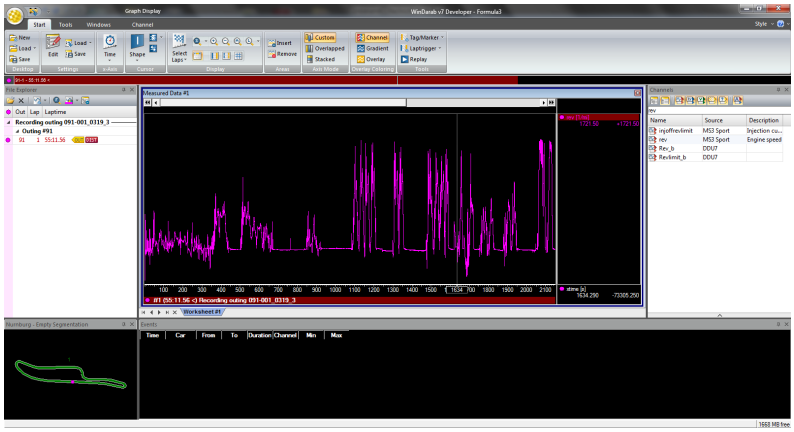


Choose your Device / IP from dropdown list

9. Connect the MS 6 EVO 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 MS 6 EVO 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.

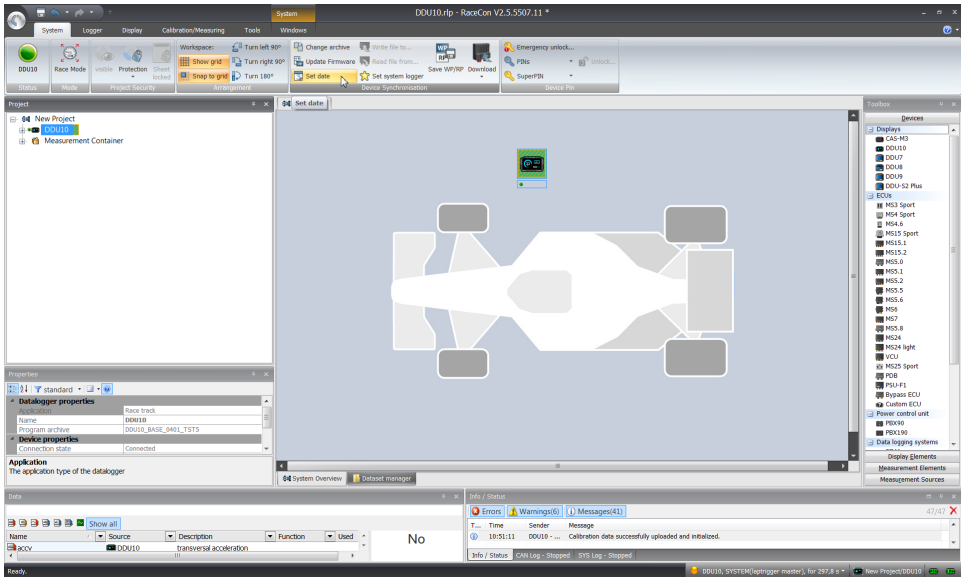


6.5 Set date and time

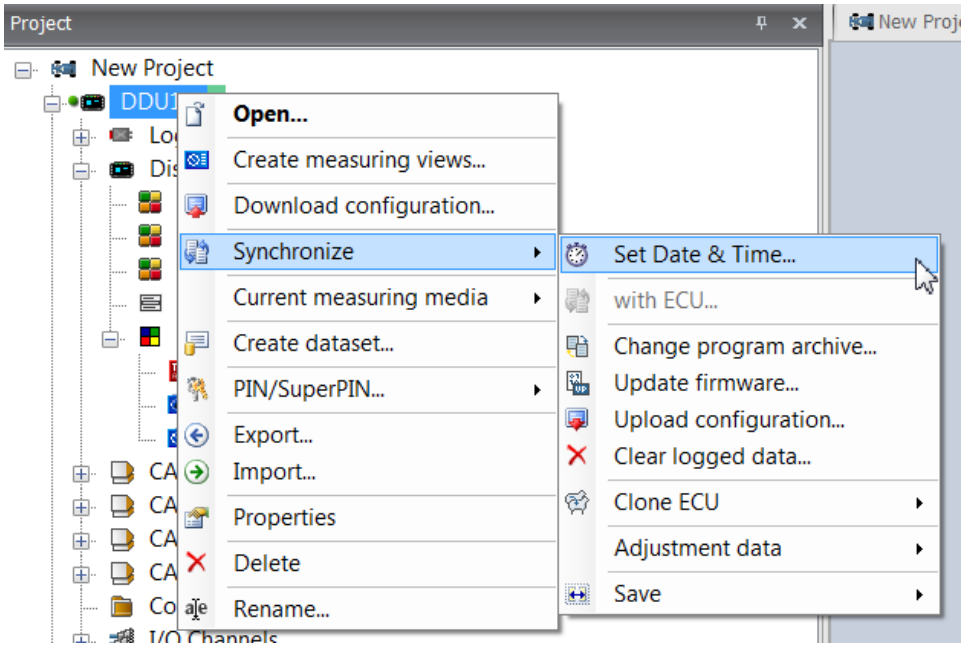
The MS 6 EVO is equipped with a real time clock which is supplied by an internal accumulator. Once this accumulator is charged correctly by 12 V supply of the display, ‘Date & Time’ can be programmed by RaceCon.

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

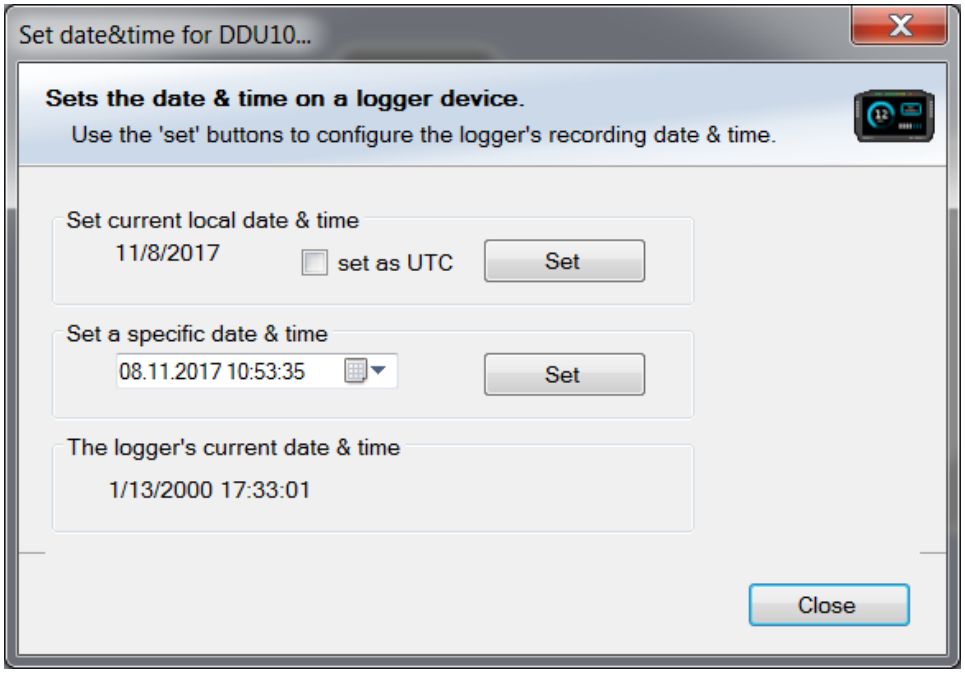
1. Connect the MS 6 EVO to the PC.
2. Click on the ‘Set date’ button in the ‘System’ tab menu.



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.

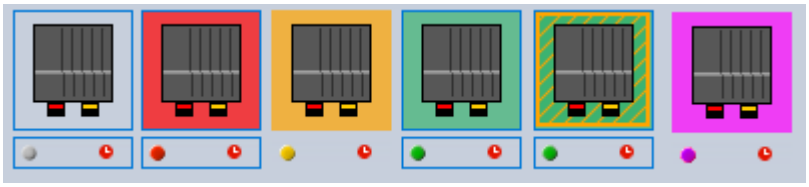


6.6 Color indication

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:

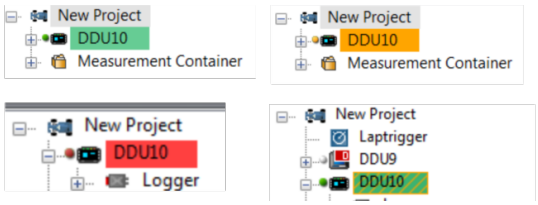
– In the status area in the upper left corner.



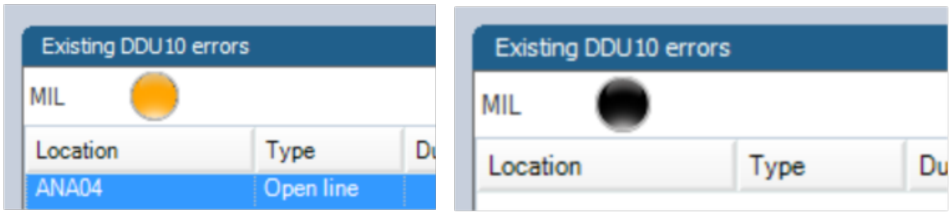
– As a background, as well as a little dot around the display icon in the ‘System window’.

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

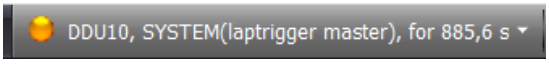
– As a colored background around the device name in the project tree.



– 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.

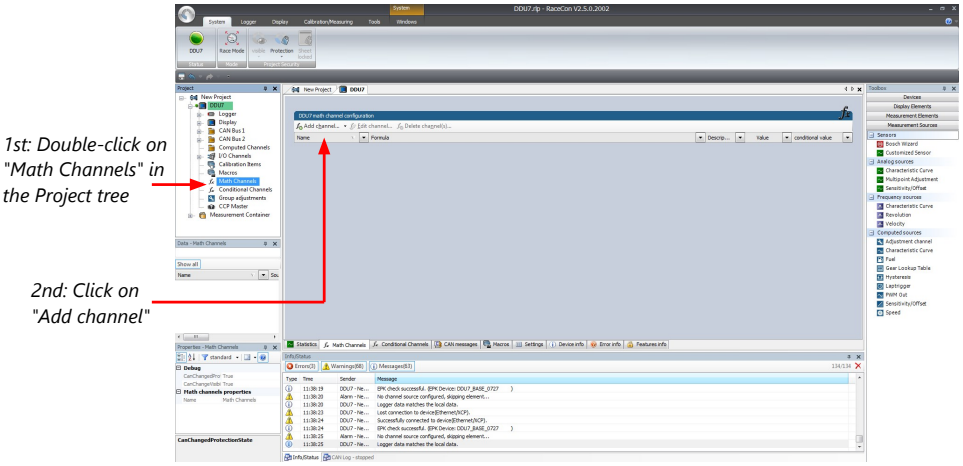
7 Project Configuration

7.1 Math Channels

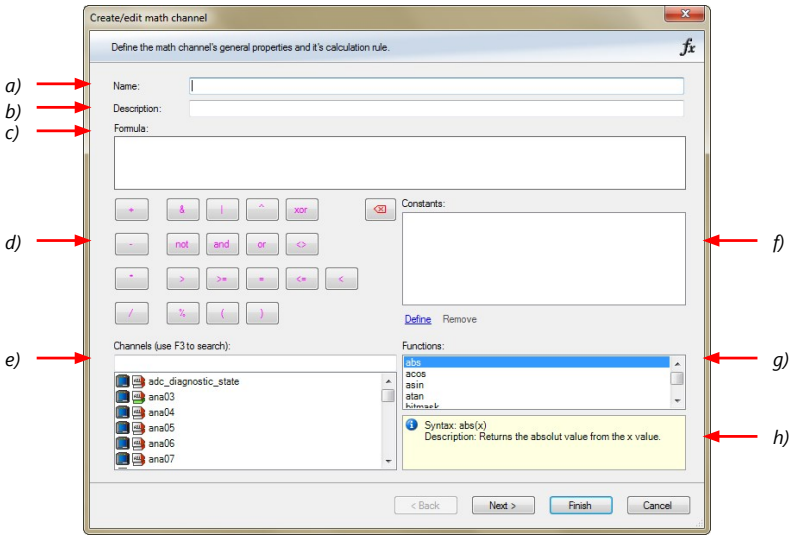
- Arithmetic and logical operations on up to 4 measurement channel(s)
- Numerical result
- Result can be used as input source for various display elements (numeric elements, alarms, bargraphs) and further calculations in the whole RaceCon project

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.

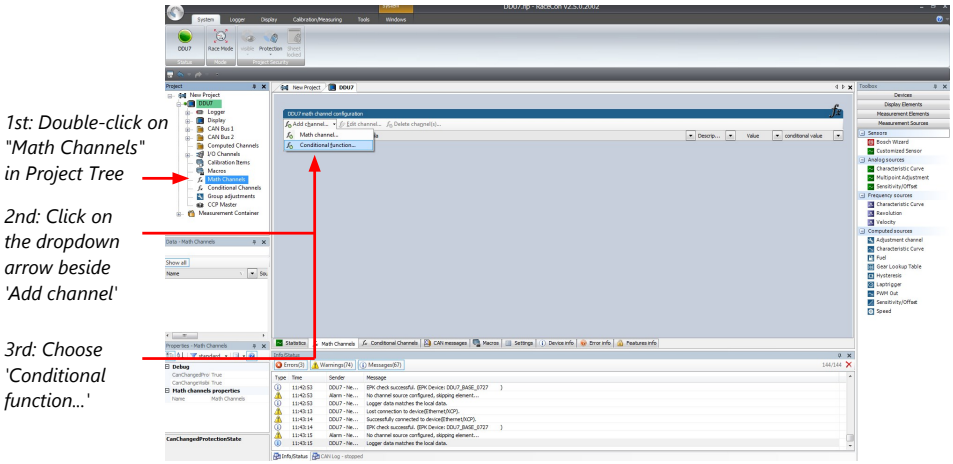
Click ‘Finish’ when done. The math channel is displayed in the math channel window.

7.2 Conditional Functions

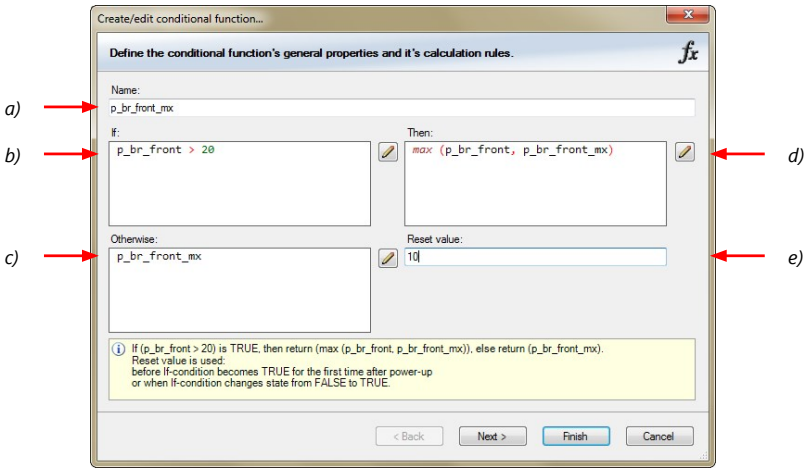
- Arithmetic and logical operations on one or more measurement channel(s)
- If-Else structure with reset
- Numerical result
- Result can be used as input source for various display elements (numeric elements, alarms, bargraphs) and further calculations in the whole RaceCon project.
  All math and conditional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

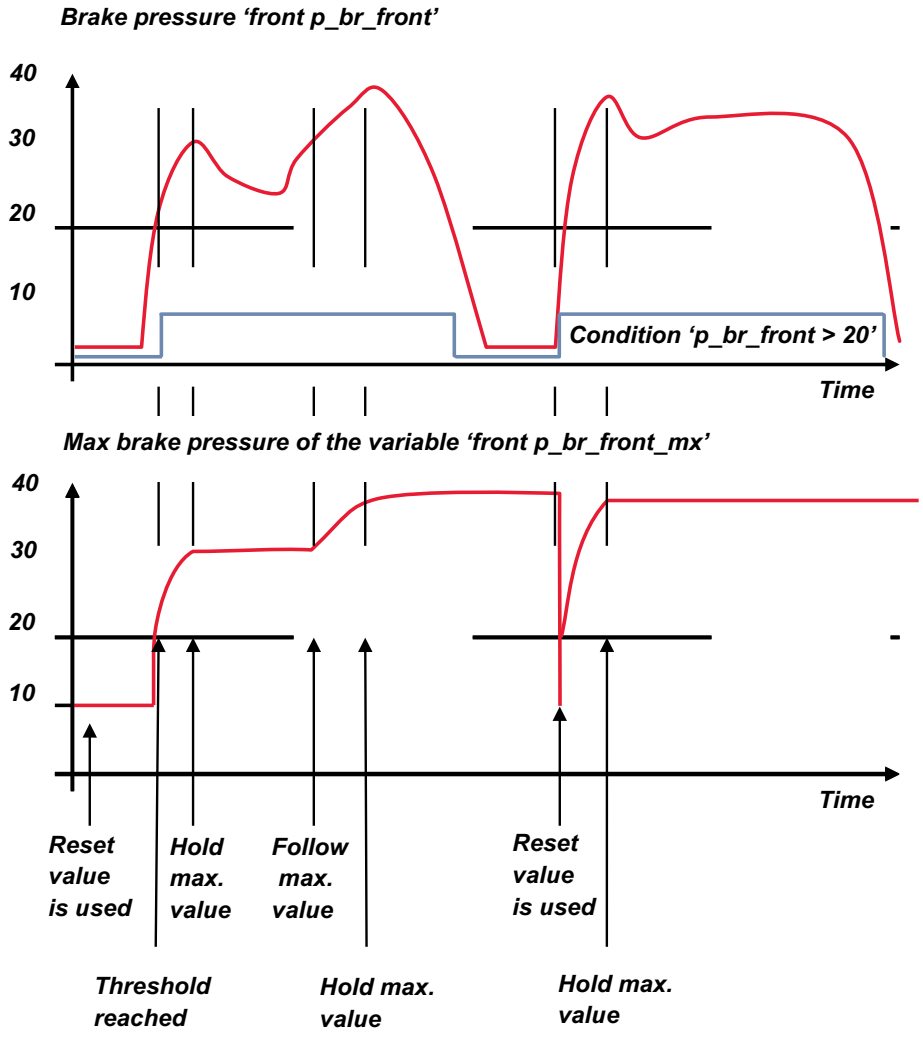
The reset value is always set for the new variable (entered in “Name”):

- 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 MS 6 EVO 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’.

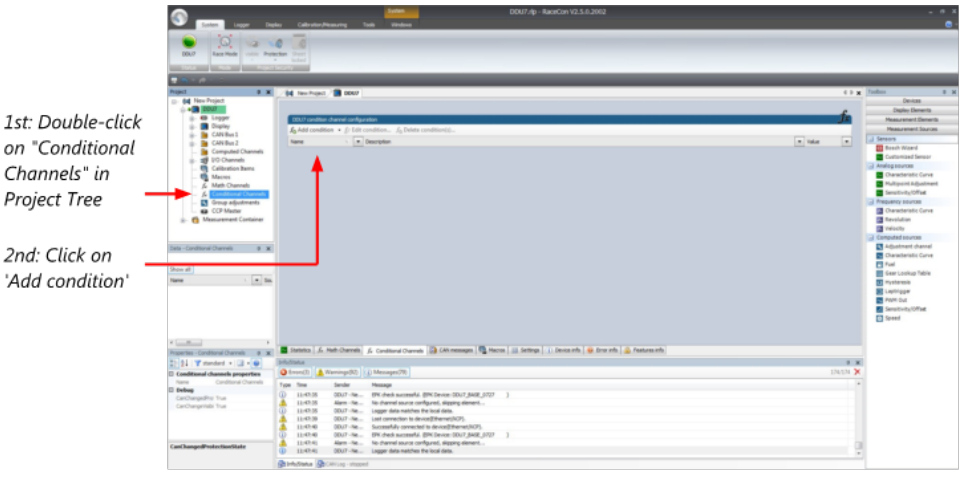
- ‘p_br_front’ rises to 30. As ‘p_br_front’ is > 20 (condition is TRUE), the condition ‘max (p_br_front, p_br_front_mx)’ in the THEN window is triggered. The condition sets the bigger value as new value for ‘p_br_front_mx’. As ‘p_br_front’ (30) is bigger than ‘p_br_front_mx’ (10), the new value for ‘p_br_front_mx’ 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

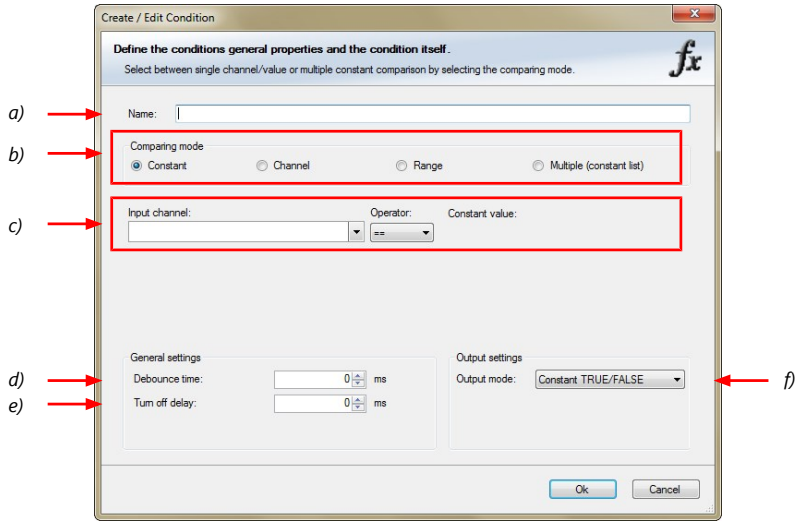
- Logical operations on measurement channel(s)
- If-Else structure with reset
- Logical result
- Result can be used as input source for alarm display elements and further calculations in the whole RaceCon project.

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.

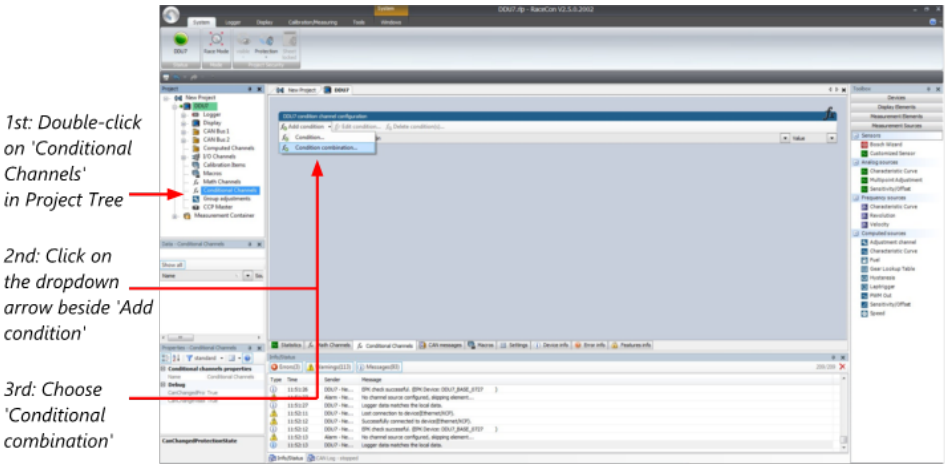
- Constant TRUE/FALSE: Result is as a constant with the value TRUE or FALSE.
- Blinking: Result is a blinking, if the condition is fulfilled.
- Pulse: Result is a short one-time pulse, if the condition is fulfilled.
- Toggling output: Result is a pulse that lasts until the next condition is fulfilled.
- Click ‘Ok’ when done. The conditional channel is displayed in the MS 6 EVO condition channel window.

7.4 Condition Combination

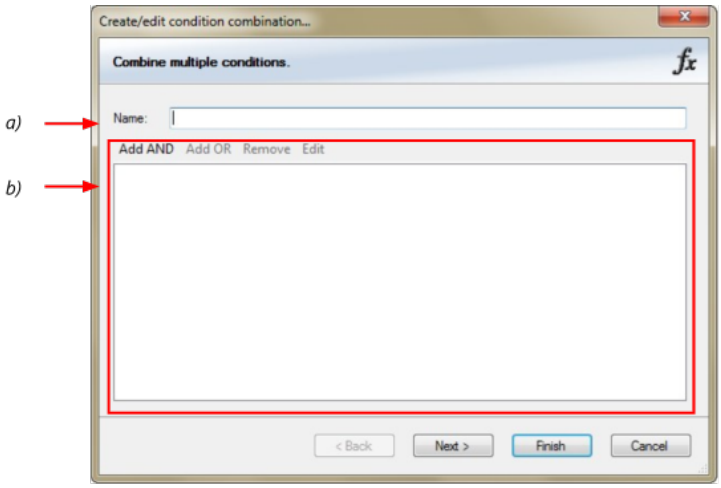
- Combination of several (up to 16) conditional channels for more complex calculations
- Logical results
- All conditions can be used globally in the whole MS 6 EVO project.

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:

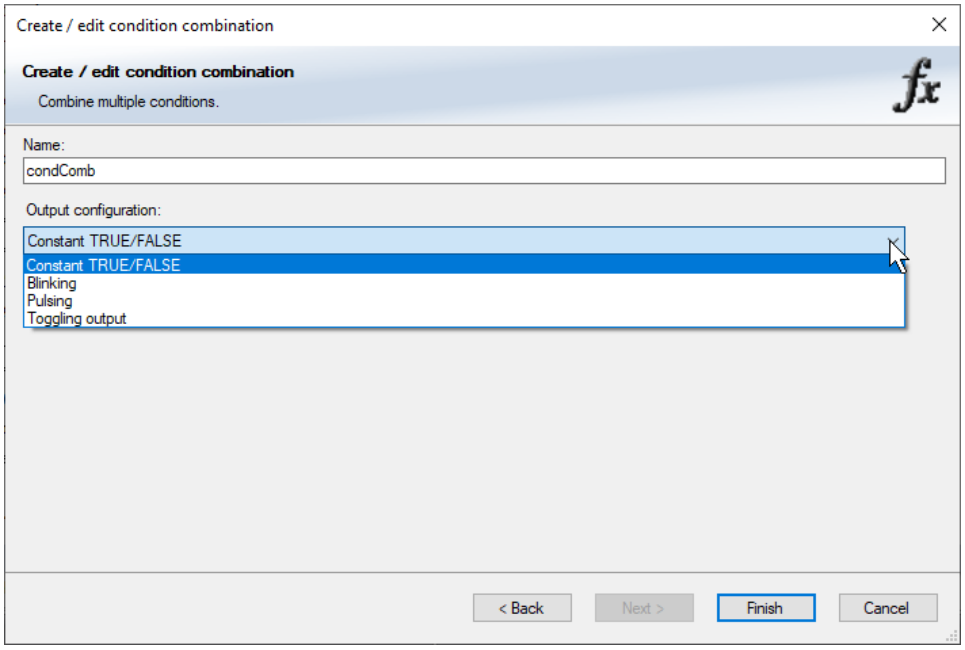


a) Enter the name of the condition combination.

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.

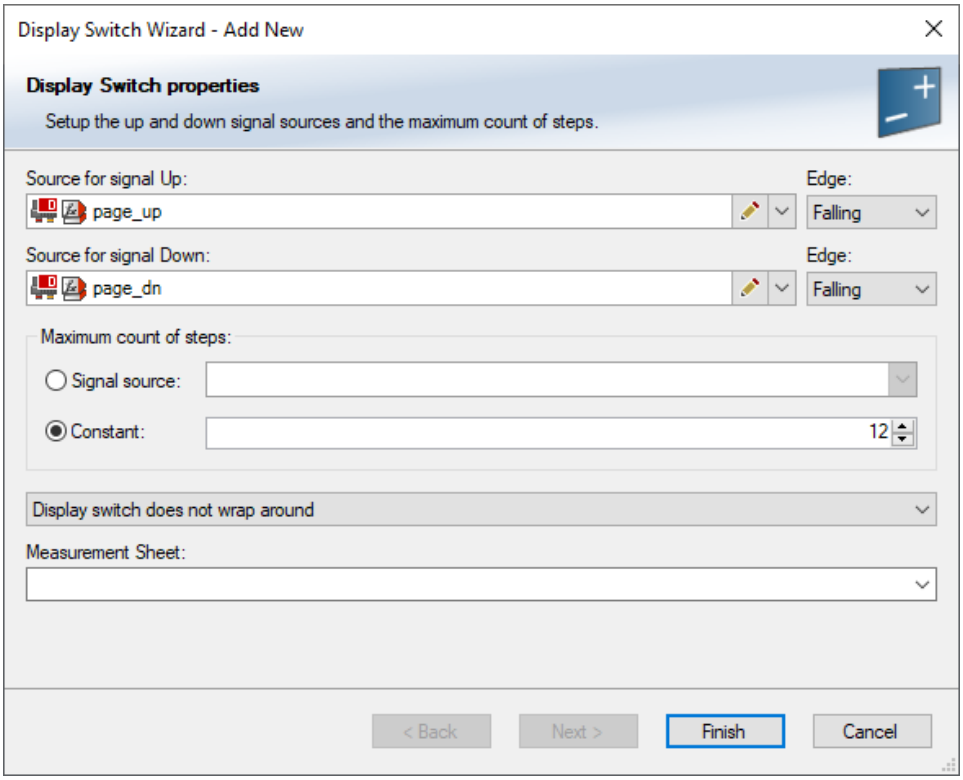
Click ‘Finish’ when done. The conditional combination is displayed in the MS 6 EVO condition channel window.

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.

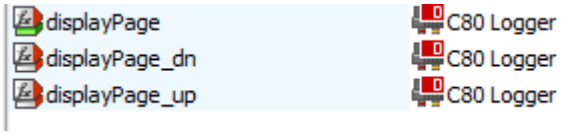
The page can be configured to wrap around. In this case, no page down condition is needed.



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

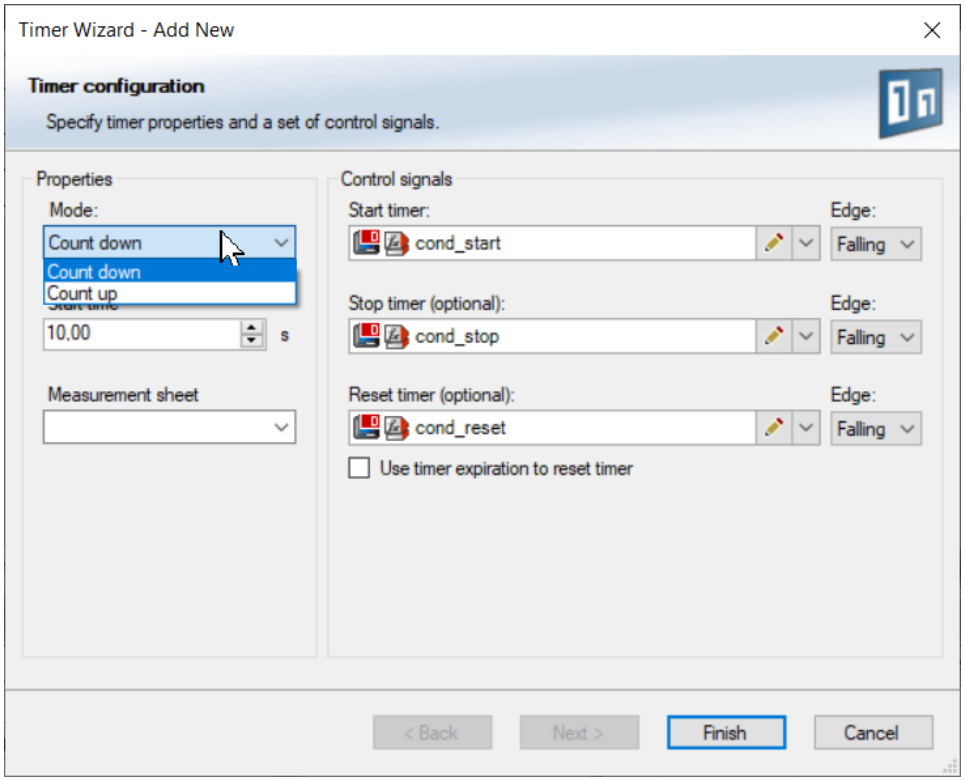
Example:



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.

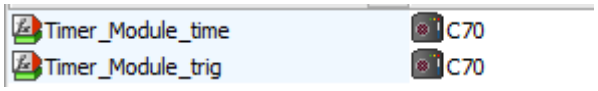
Up counting mode and down counting mode are available, triggers are fired at set time (up counting) or at zero (down counting). The running timer will keep its state over a power cycle.



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:

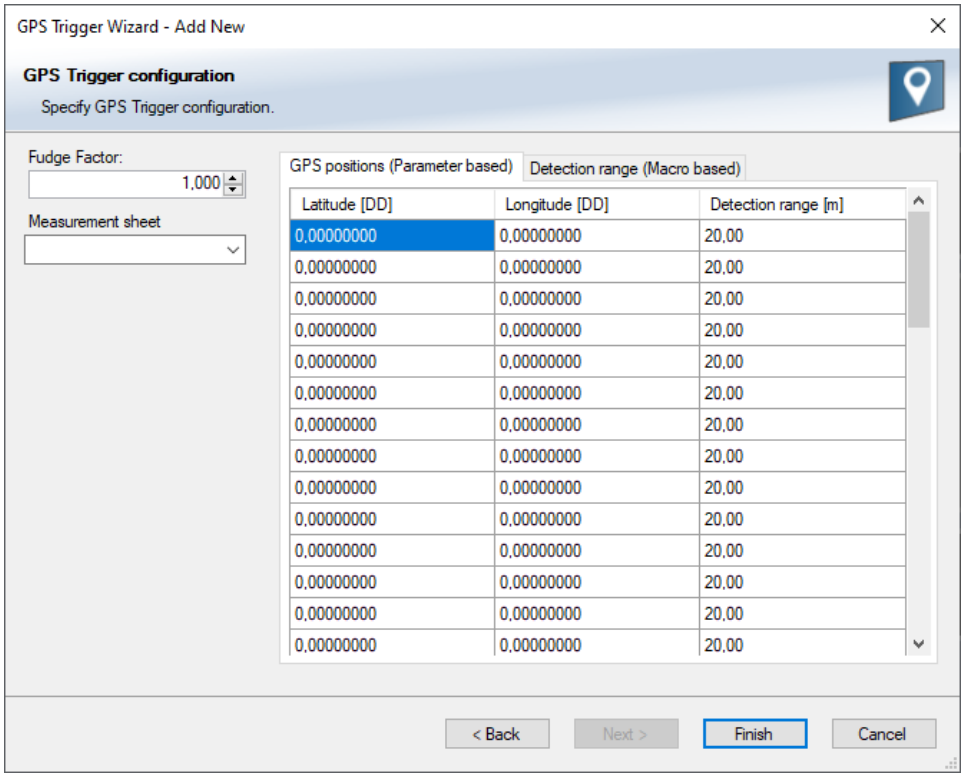


7.7 GPS Trigger Module

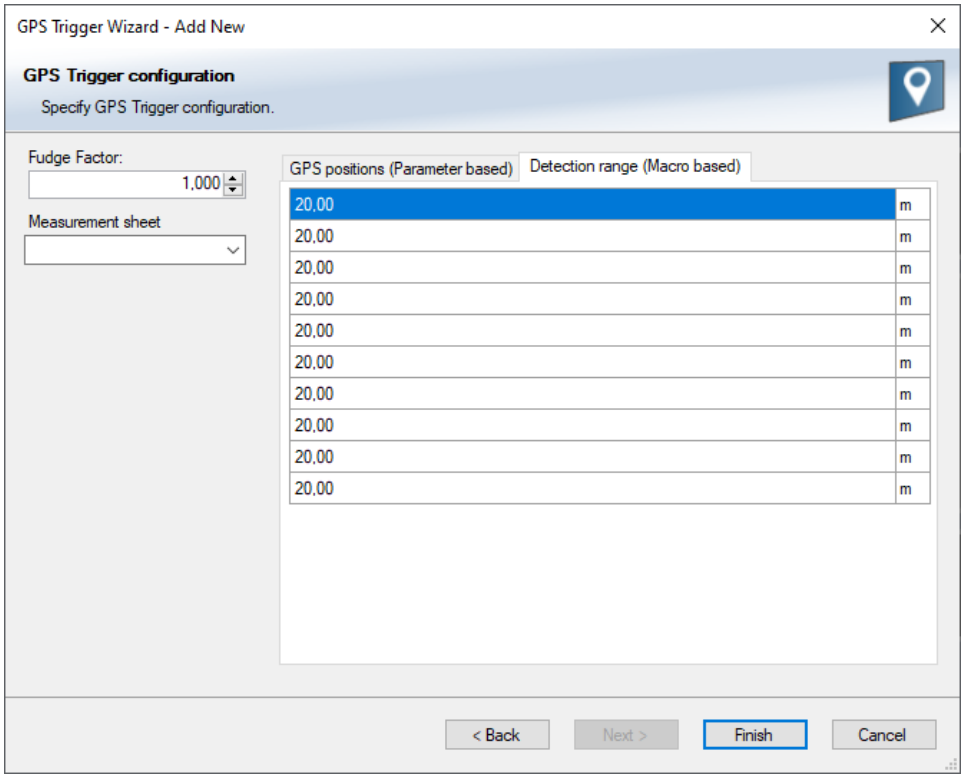
The GPS Trigger Module triggers depending on GPS-position, like the GPS-laptrigger.

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.

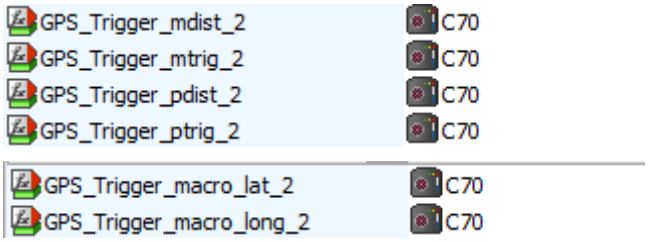


The GPS trigger points can also be used for segment triggering. If used as segment triggers and i.e., 3 trigger points are selected, the laptrigger module will use the first 3 trigger points on the list.

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 MS 6 EVO 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 MS 6 EVO 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
- Logger configuration (total logging rate [kB/s], conditional measurement rates)

To help respecting the limit of 85 % CPU load, the MS 6 EVO 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 MS 6 EVO resets due to complex configuration setups, please consider reducing the demands on the MS 6 EVO adapting the influencing factors mentioned above.

8 CAN Configuration

The MS 6 EVO has 3 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 MS 6 EVO 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).

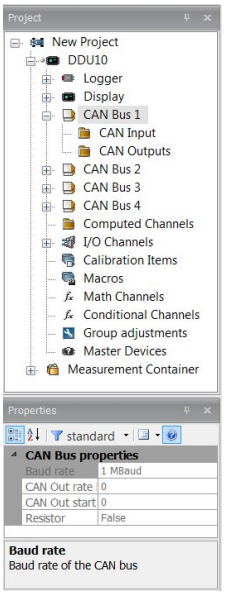
8.1 CAN Bus Trivia

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

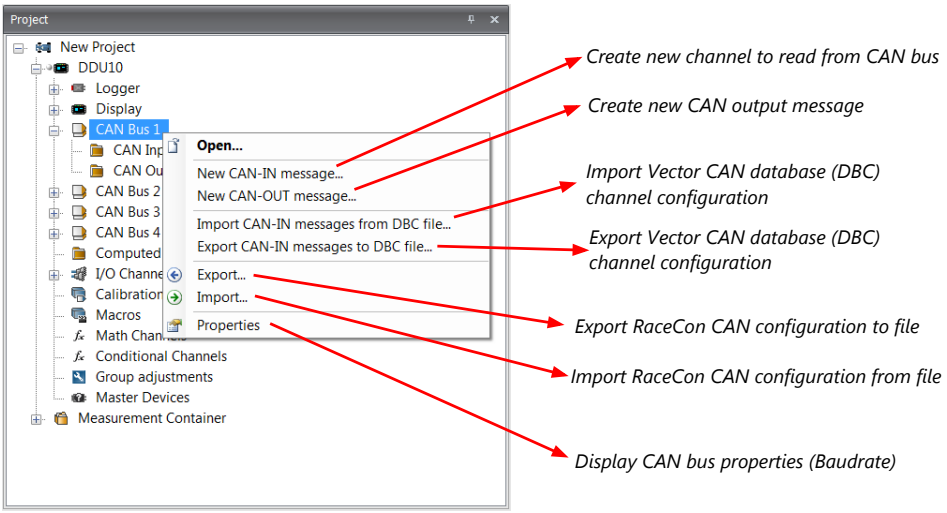
	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x100	0	p_oil		t_oil				
0x100	1	s_dam_fl		s_dam_fr				
0x100	2	s_dam_rl		s_dam_rr				

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

8.2 CAN input

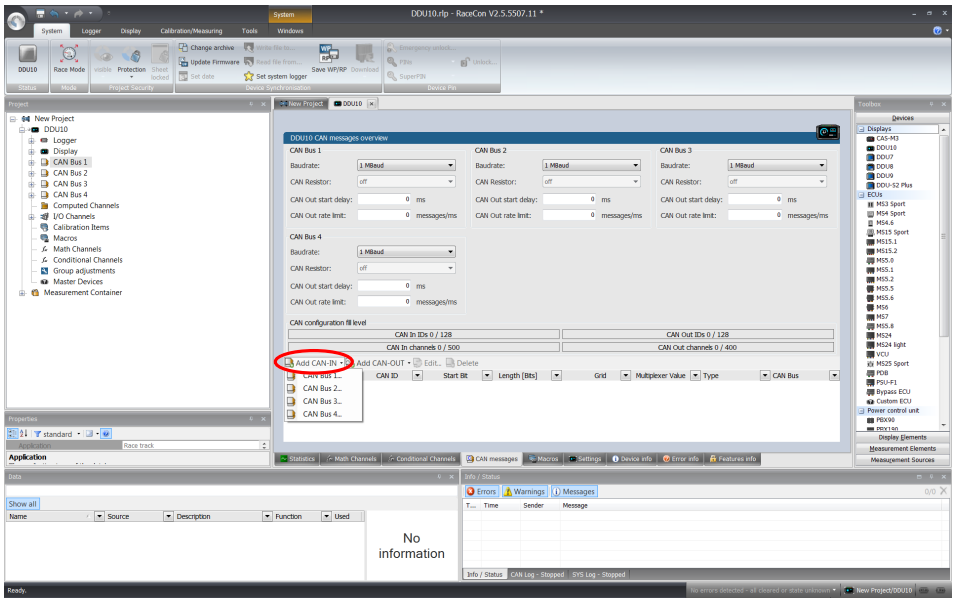
8.2.1 Input configuration

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



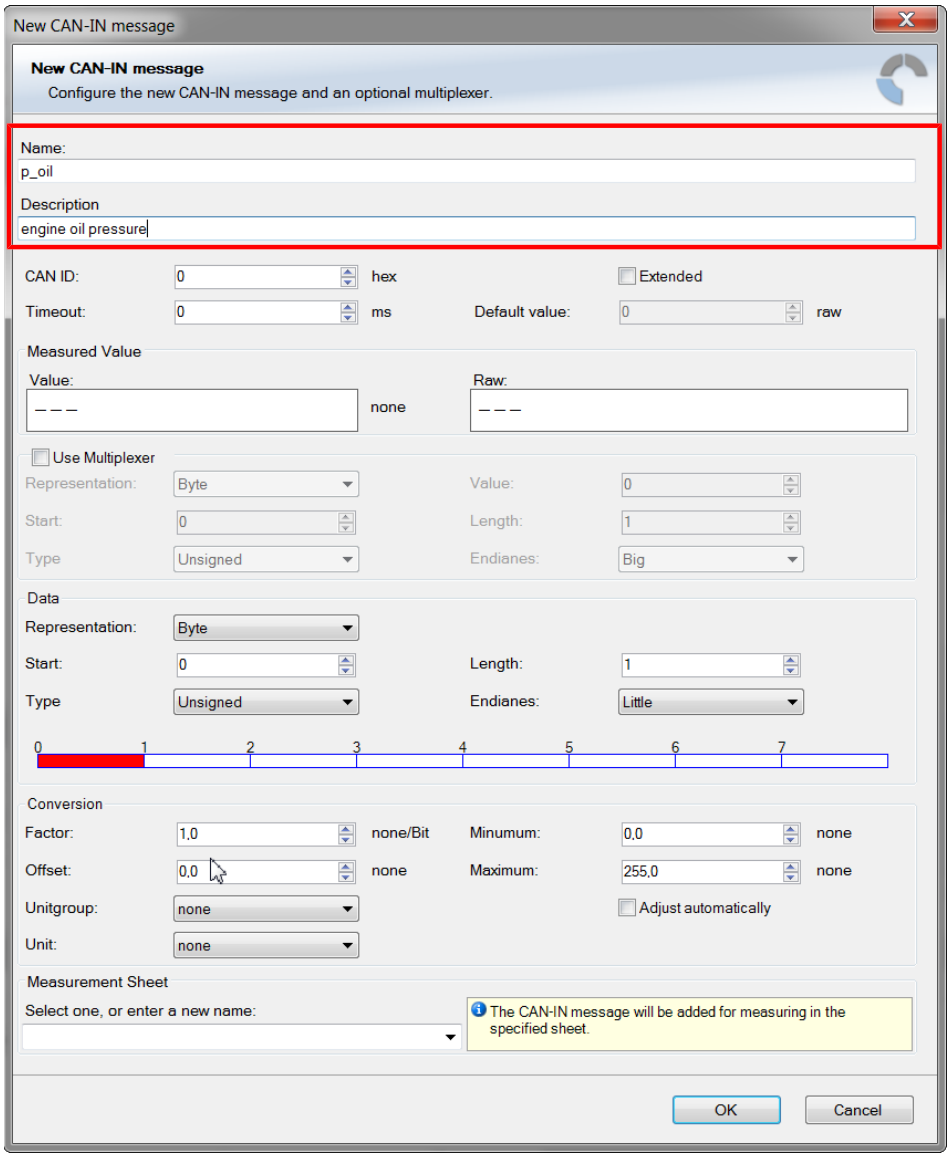
8.2.2 Create new CAN Input channel

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.

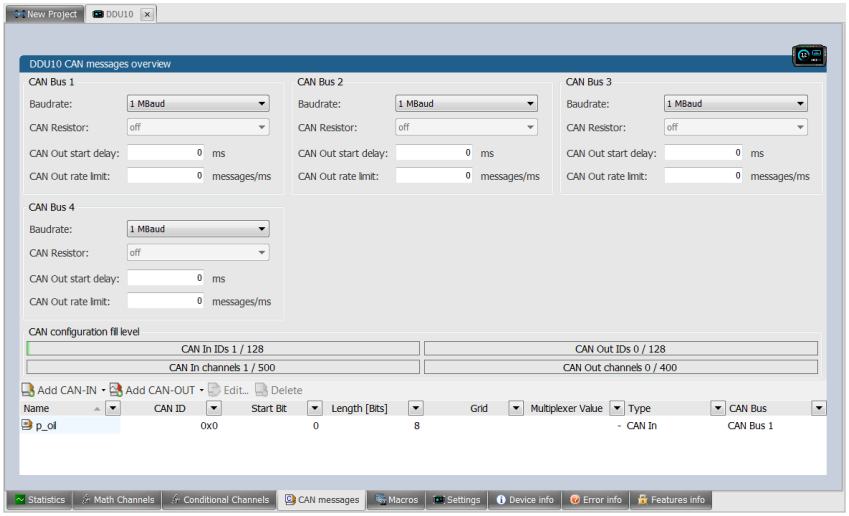


3. A CAN channel configuration window opens.

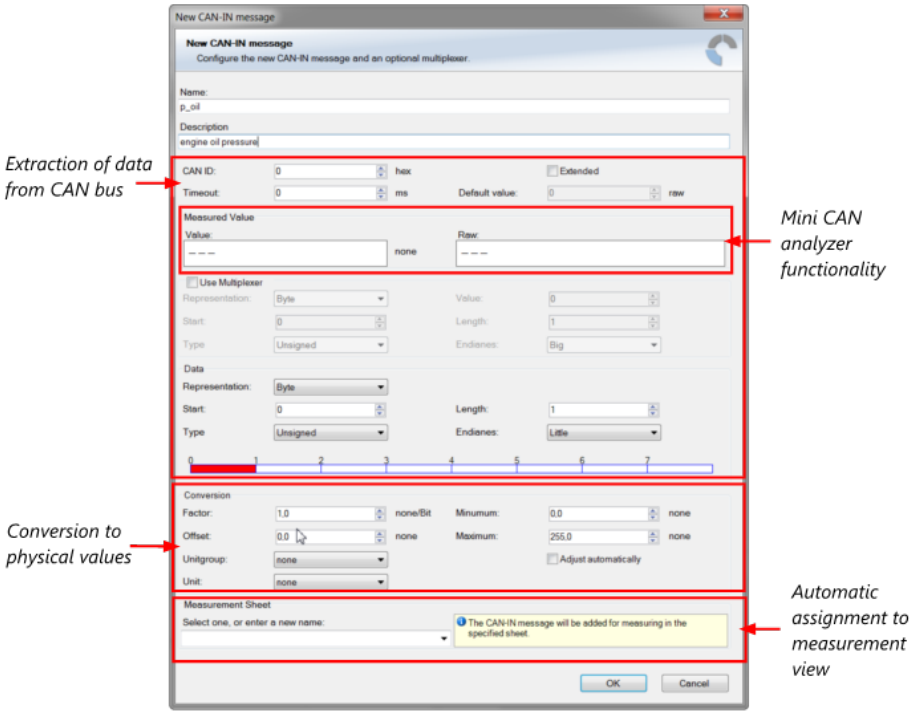
4. Insert the name and description of the channel.



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



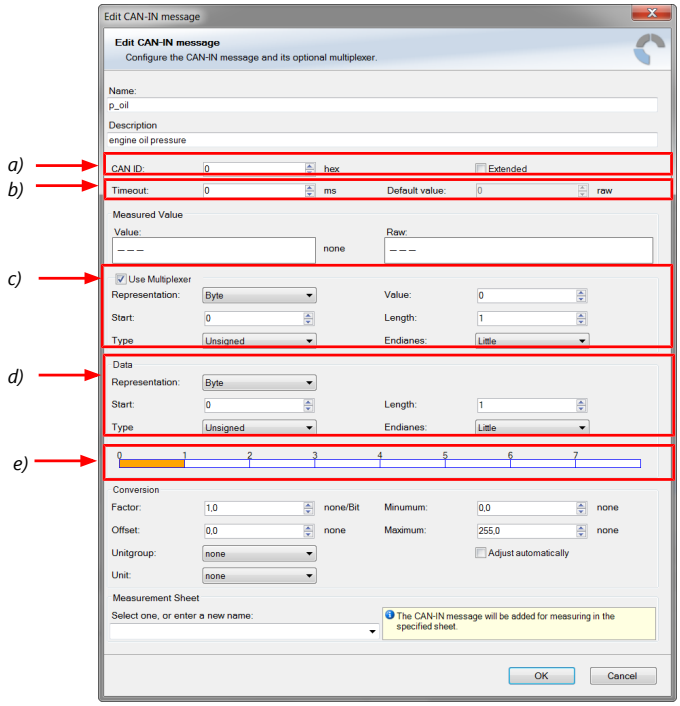
CAN channel configuration



8.2.3 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.

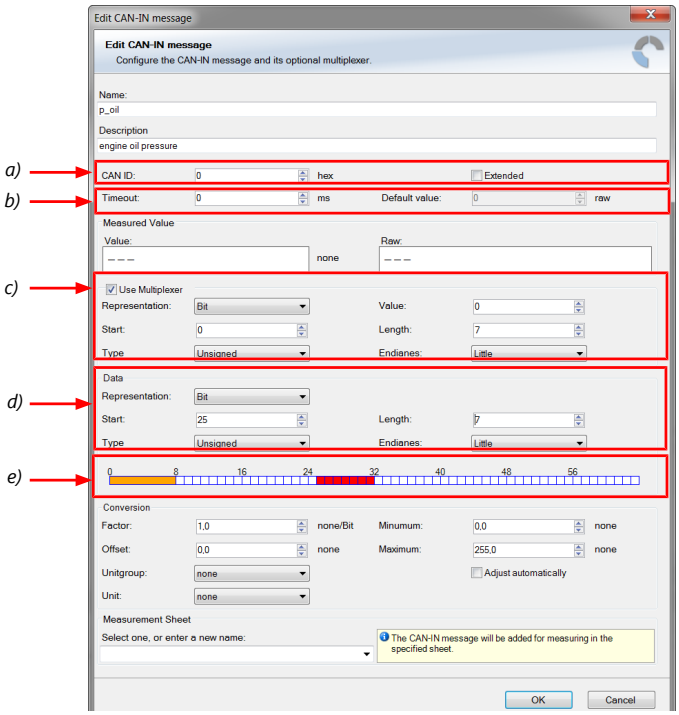
d) Enter data position, length and format.

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.

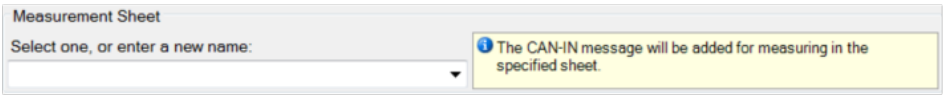
CAN analyzer functionality

This functionality is only available, if a MSA-Box (I or II) is used to connect the MS 6 EVO 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.

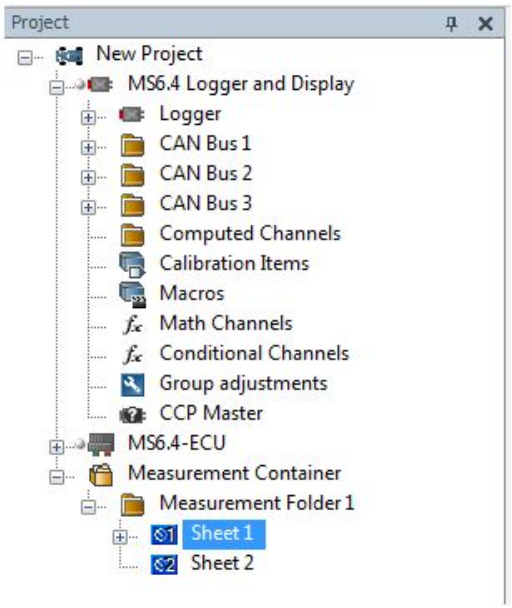
Automatic creation of online measurement sheets

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.

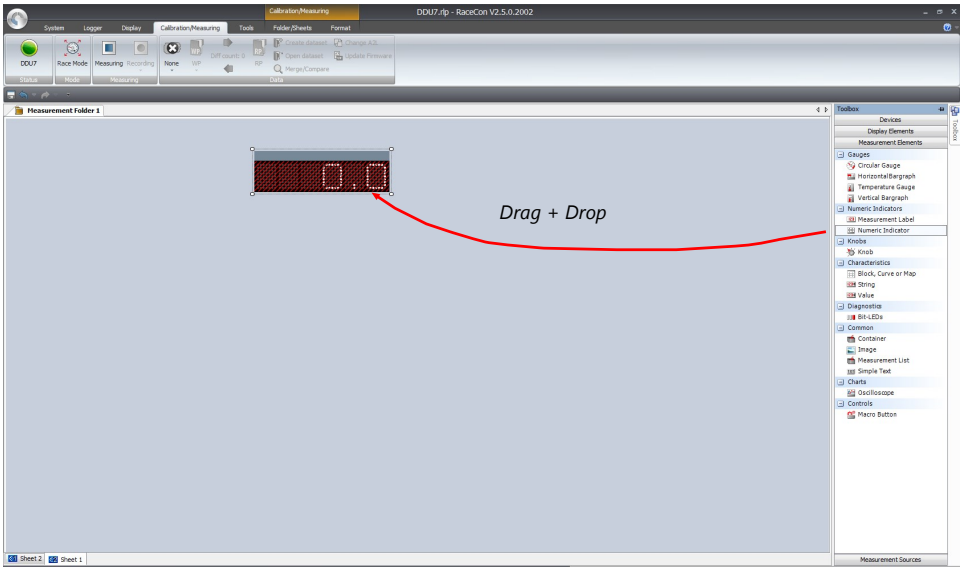
For an online view of the value measured by the MS 6 EVO, insert the channel in an online measurement sheet which is described in the chapter Setting up an online measurement [▷ 76].



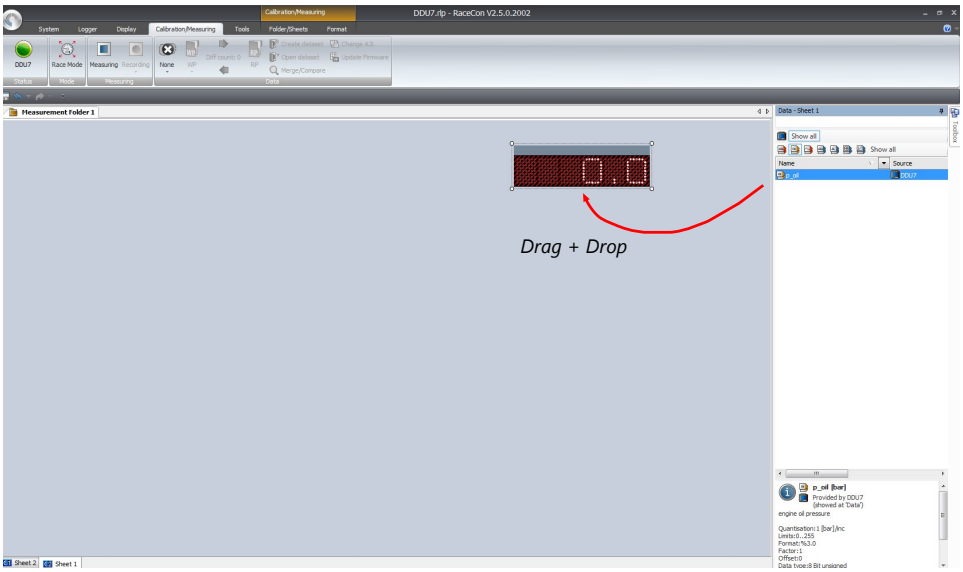
8.2.4 Online view of CAN channels in vehicle



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.



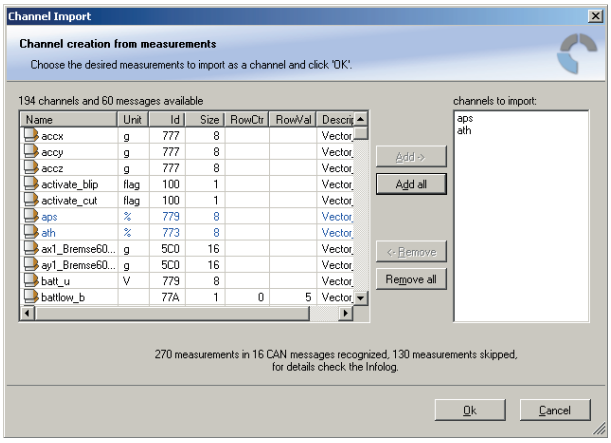
4. Click on folder ‘CAN Input’ of desired CAN bus to display available channels.
5. Drag desired Measurement channel and drop it on the Measurement element.



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.

8.2.5 Import a CAN database (DBC) file

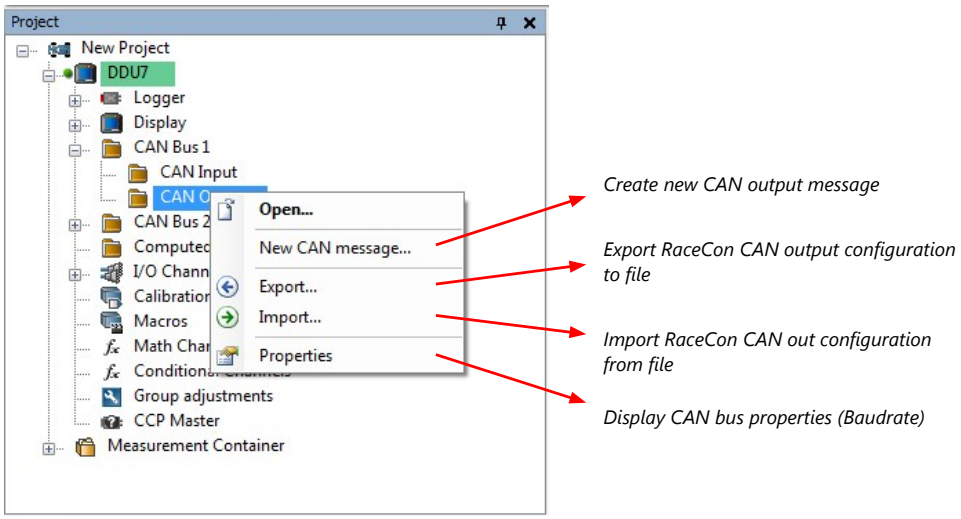
1. Right-click on CAN Input of desired bus (CAN1 or CAN2).
2. Select ‘Import DBC file’ from menu. A file browser opens.
3. Select DBC file to import and click ‘OK’ when done. A channel import window opens.



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.

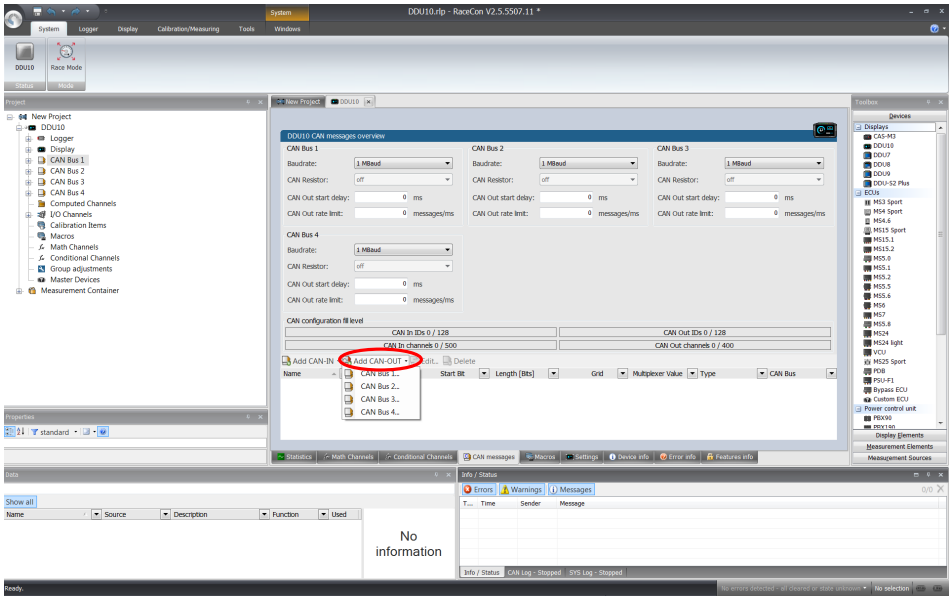
8.3 CAN output

8.3.1 Output configuration

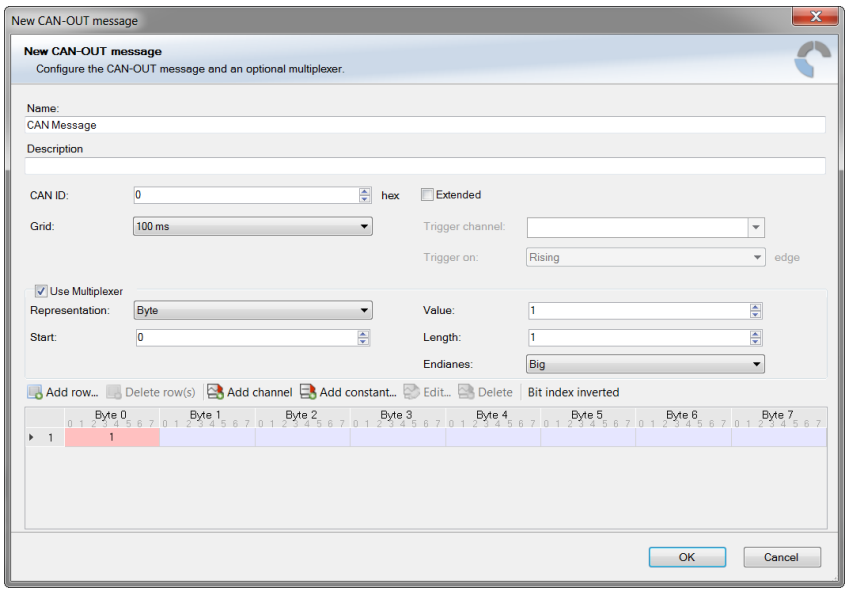


8.3.2 Create a new CAN output message channel

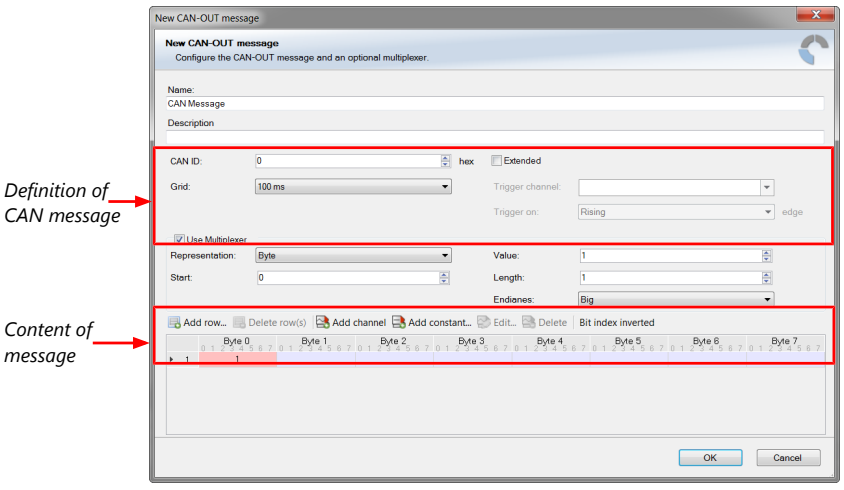
- Double-click on any CAN bus item to open the "CAN messages overview".
- Select ‘Add CAN-OUT’ and choose the desired CAN bus for the new output channel.



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

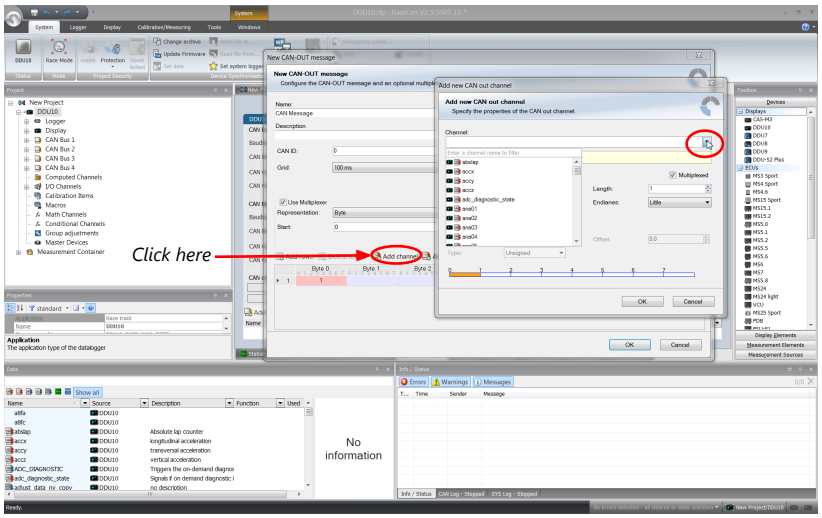


– Enter name of message, description, CAN-Id, and Grid (output interval). Optionally, specify a multiplexer.



– 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.

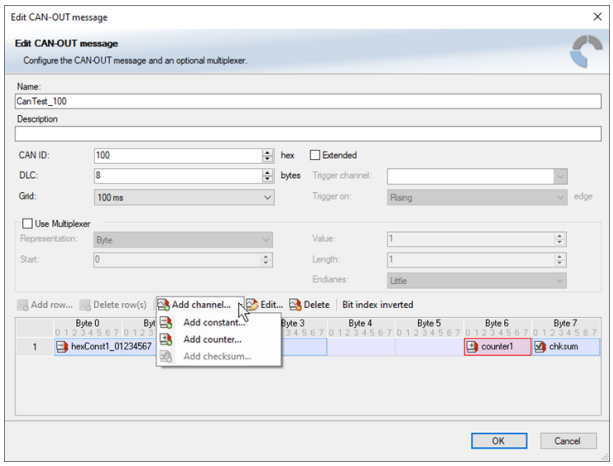


The measurement channel is now assigned to the CAN message.

8.3.2.1 Add CAN out constant

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

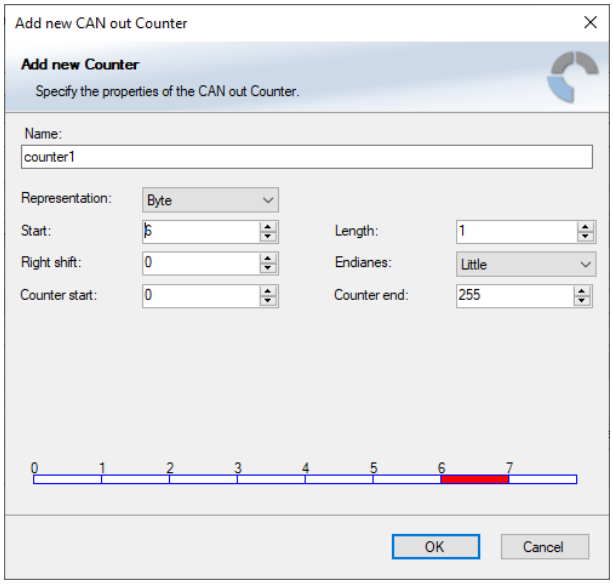
1. Create a new CAN output message or edit an existing message.
2. Click small arrow beside ‘Add channel...’ and select ‘Add constant...’. The ‘Add new CAN Out constant’ window appears.
3. Define the name of the constant, the required value in hex and define the CAN channel settings.
4. Click ‘OK’ when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



8.3.2.2 Adding CAN out counter

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

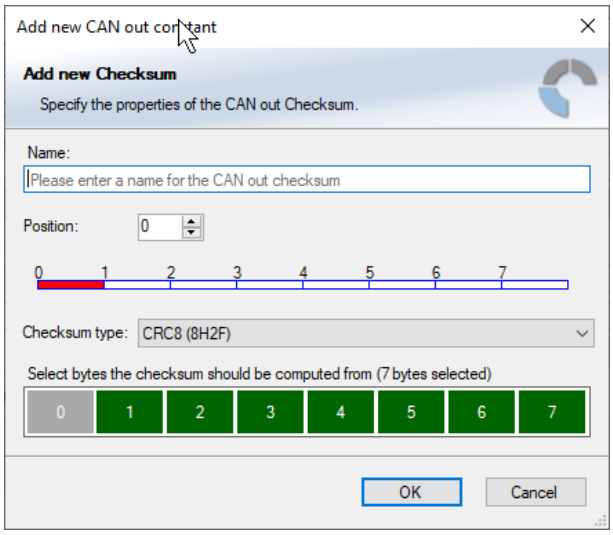
1. Create a new CAN output message or edit an existing message.
2. Click small arrow beside ‘Add channel...’ and select ‘Add counter...’. The ‘Add new CAN out counter’ window appears.
3. Define the name of the counter, define the CAN channel settings.
4. Click ‘OK’ when done.



8.3.2.3 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.



8.4 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.

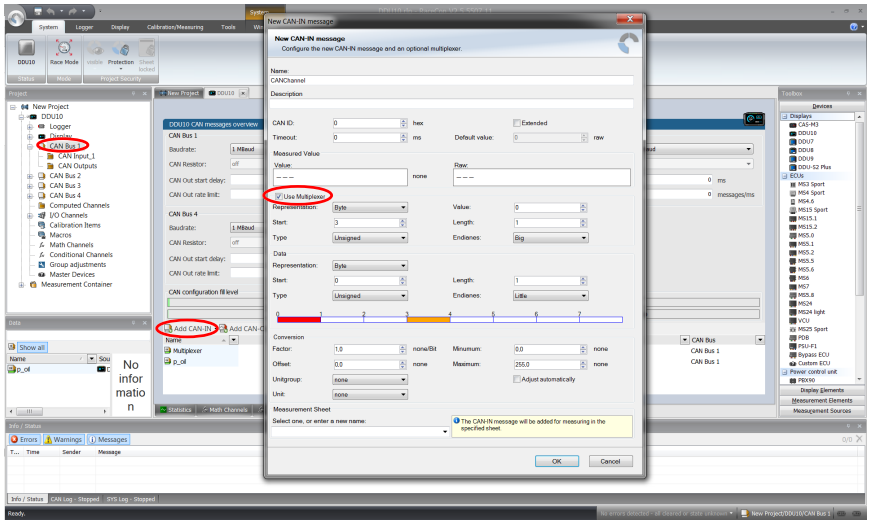
- Re-use (multiplex) of message identifiers by splitting it into several rows.
- Every row is assigned to a unique value of the multiplexer.
- One byte of message contains row counter.

– 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.

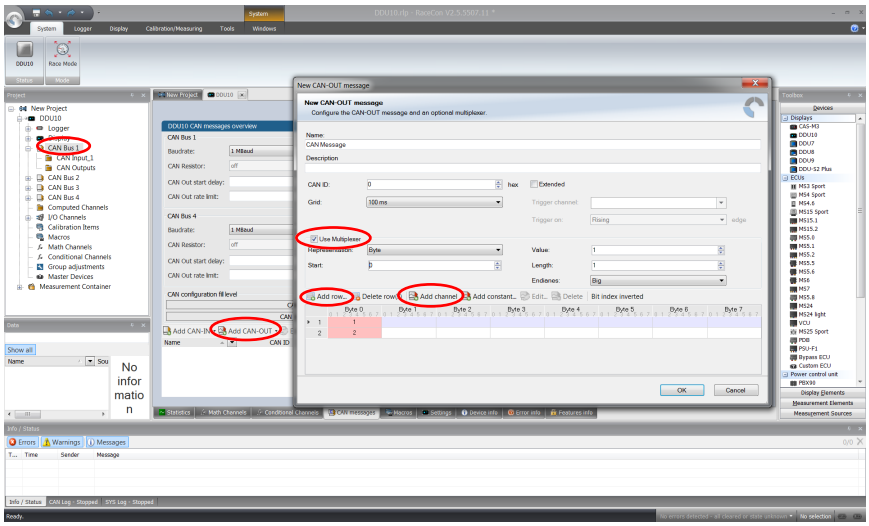
– Position of row counter is configurable.

To use a multiplexer perform the following steps:

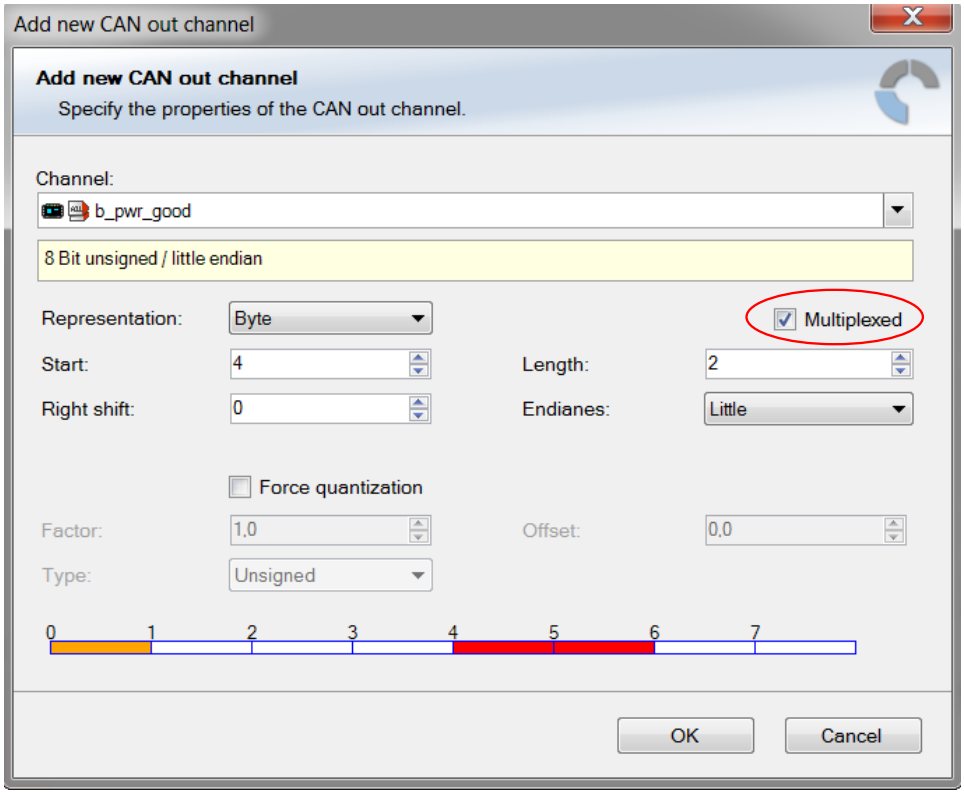
1. Double-click on any CAN bus item to open the “CAN messages overview”.
2. Select ‘Add CAN-IN’ and choose the desired CAN bus for the new input channel.
3. Check the box ‘Use Multiplexer’ and configure the multiplexer for the new CAN-IN channel.



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.

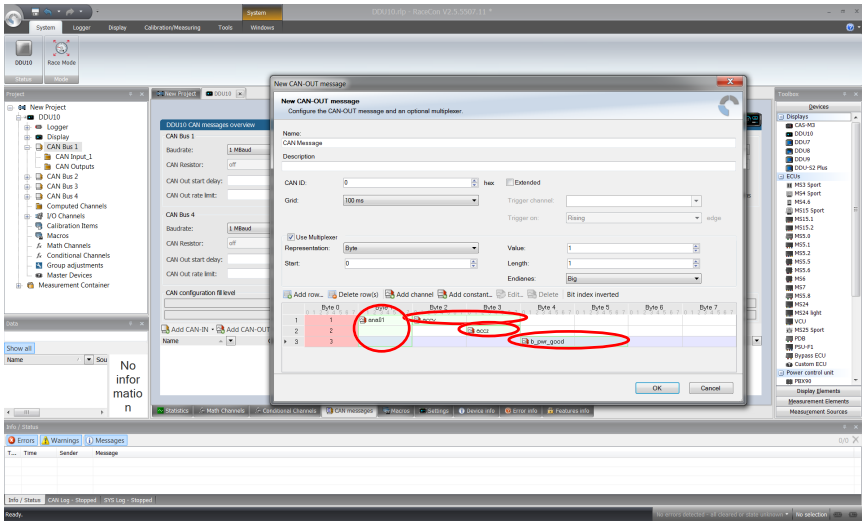


7. The ‘Add new CAN out channel’ dialog opens.
8. Select a channel and configure it. To assign it to the row selected before, check the box ‘Multiplexed’.
9. To move the channel message, change the “Start” value or click and hold the green field in the “Add new CAN out message” window.
10. Click ‘OK’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



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

12. Click ‘OK’ when done.



9 Export and Import in RaceCon

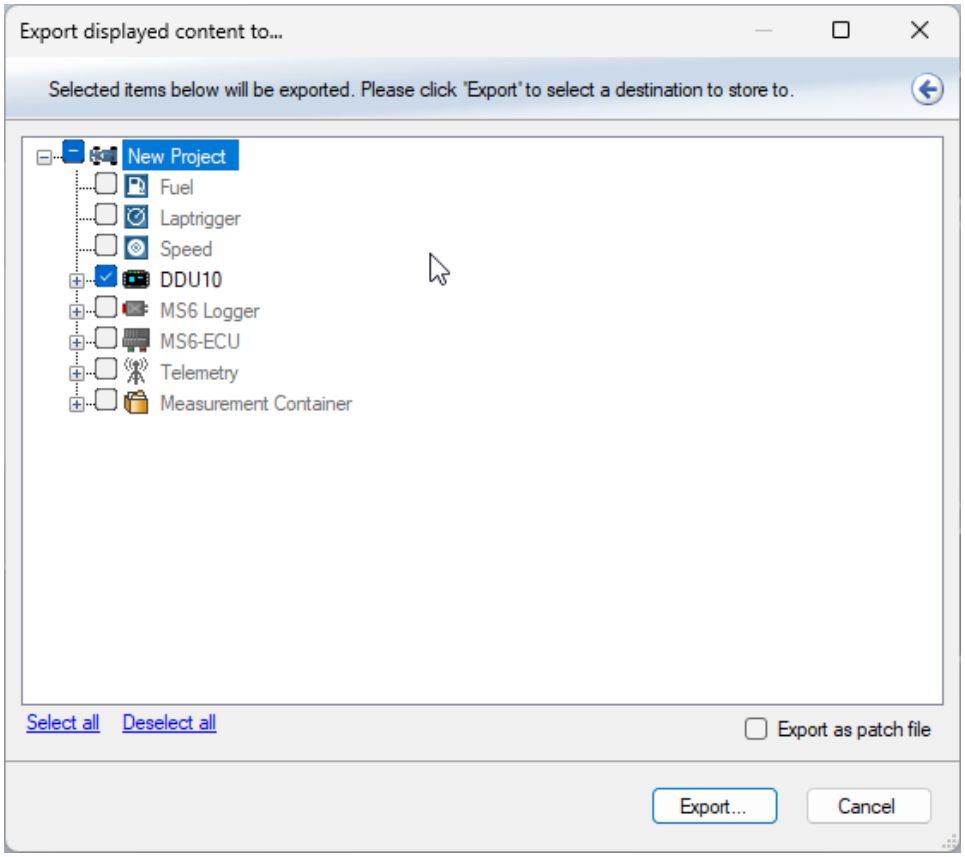
You can perform an export or an import on almost any level in the project tree.

9.1 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.



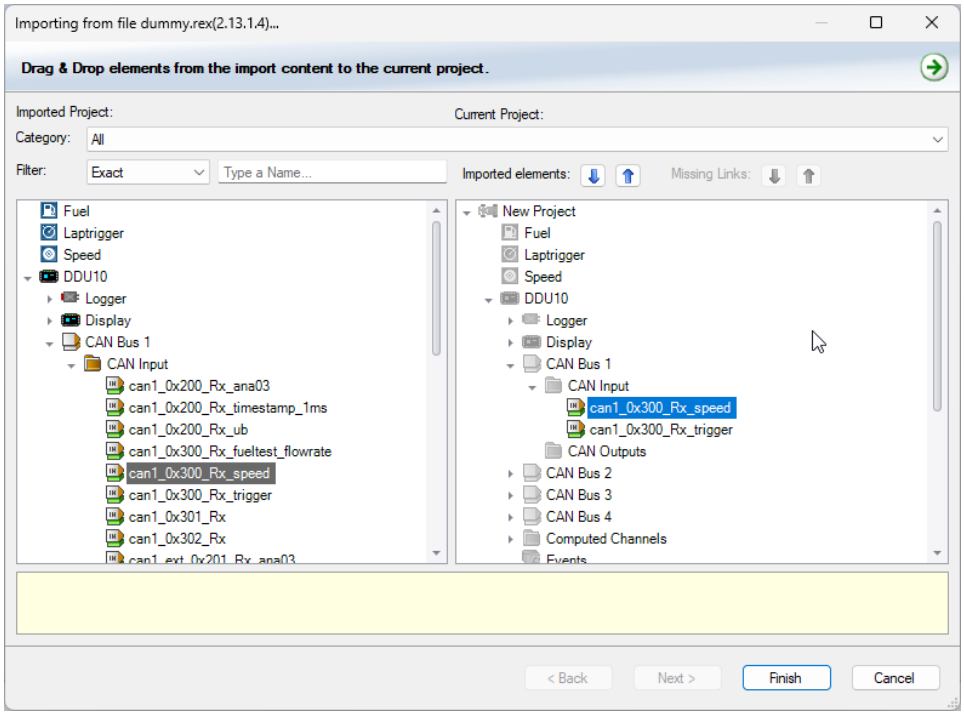
3. Click on 'Export' to select a destination to store.
4. Specify the filename.
5. Click 'Save' when done.

9.2 Import in RaceCon

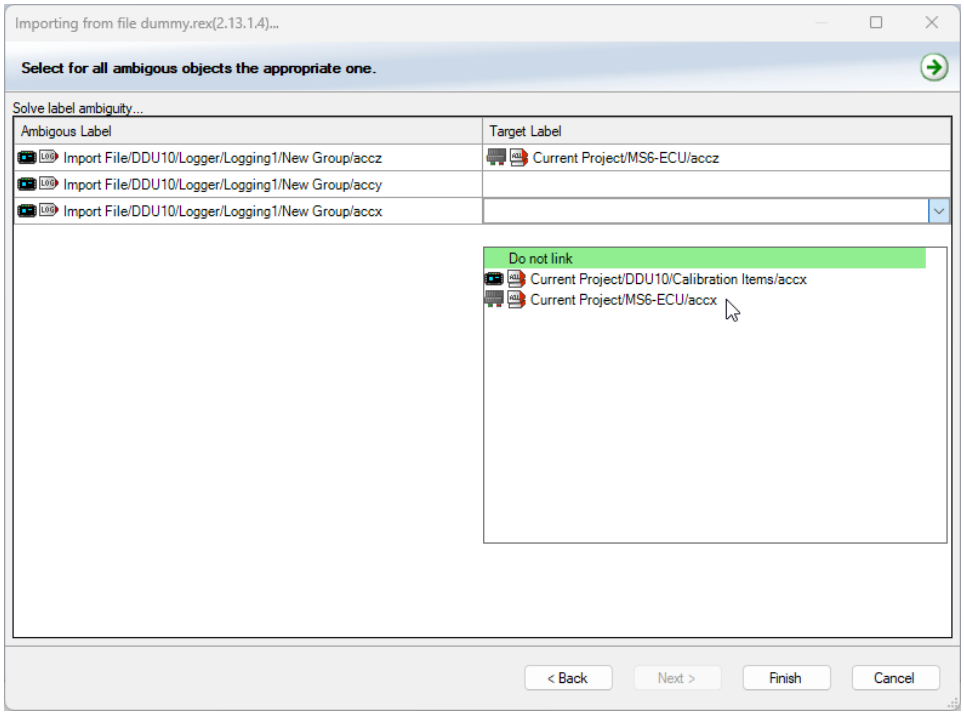
You can choose to import into the whole project or you can import into specific parts of the project.

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. MS 6 EVO and MS 6), the ‘Solve Label Ambiguity’ window opens.



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

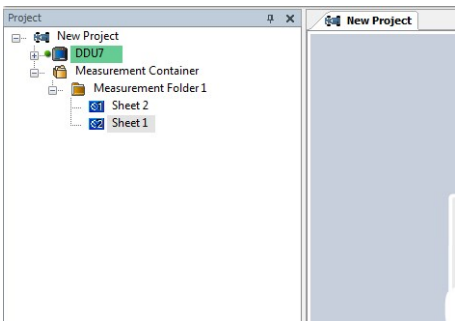
10 Online Measurement and Calibration

- 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

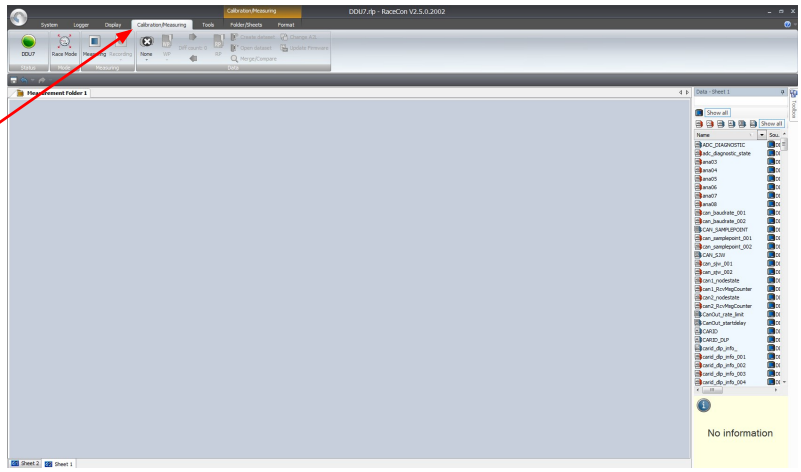
10.1 Setting up an online measurement

MS 6 EVO supports online measurement of sensor values and diagnostic variables.

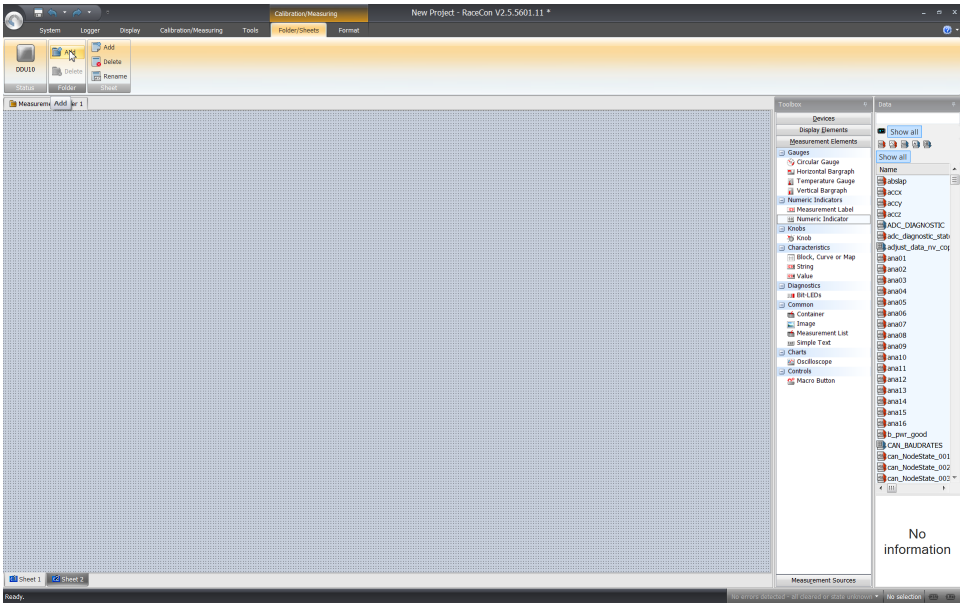
1. Expand ‘Measurement Container’ and ‘Measurement Folder 1’ in the Project Tree and double-click on ‘Sheet1’. Alternatively, click on the ‘Calibration/Measuring’ tab to open the window directly. ‘Sheet 1’ opens in a new ‘Calibration/Measuring’ window.



RaceCon changes automatically to "Calibration/ Measuring"

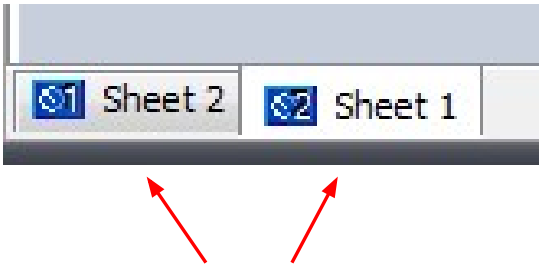


2. Click on the ‘Folder/Sheets’ tab, which appears when you are in the ‘Calibration/ Measurement’ window, to create a new measurement folder.
3. Click on the ‘Add’ button for folders in the upper left corner.



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

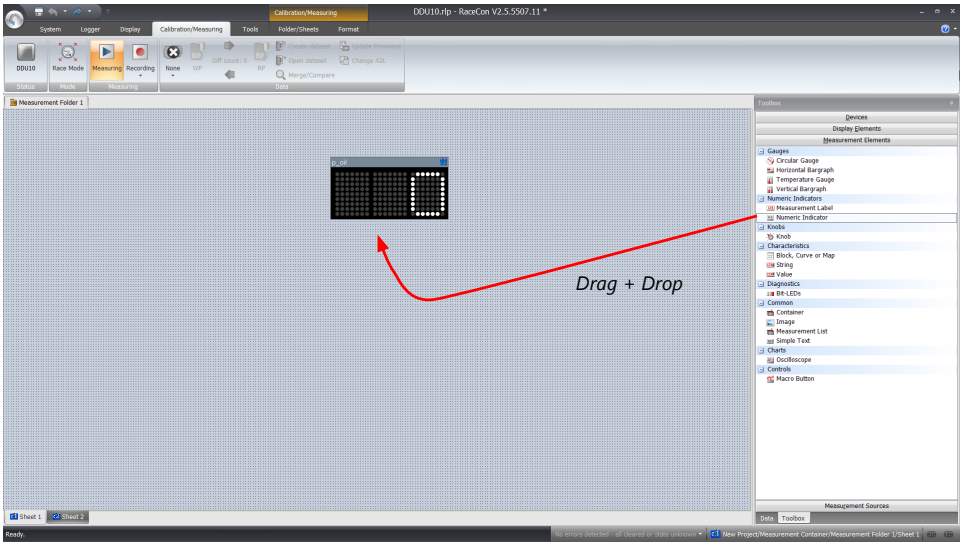
4. To change between different sheets, click on the tabs on the bottom of the 'Calibration/Measuring' window.



Tabs to switch between sheets

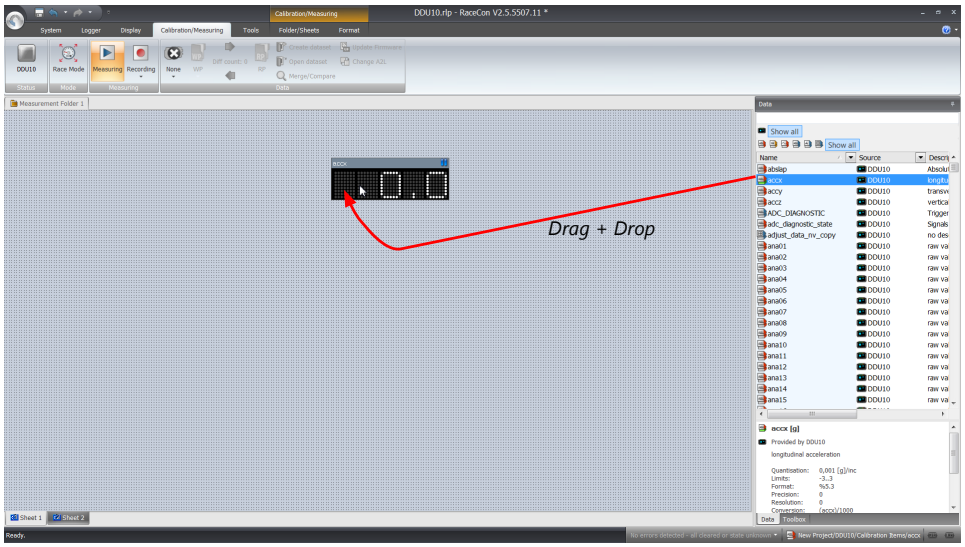
To add an element to a measurement sheet, perform the following steps:

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



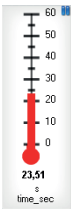
6. Select the desired measurement channel from the 'Data' area and drop it on the measurement element.

If the MS 6 EVO shows the green status, the value is displayed.

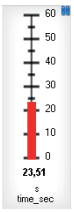


RaceCon offers different types of measurement elements:

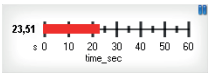
Circular gauge



Temperature gauge

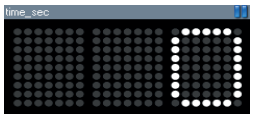


Vertical Bar graph style

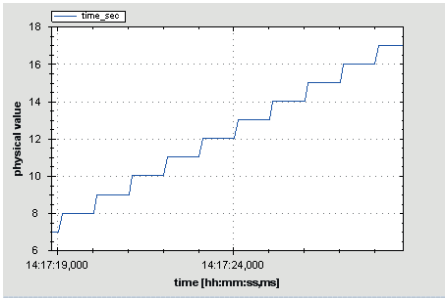


Horizontal Bar graph style

Measurement label



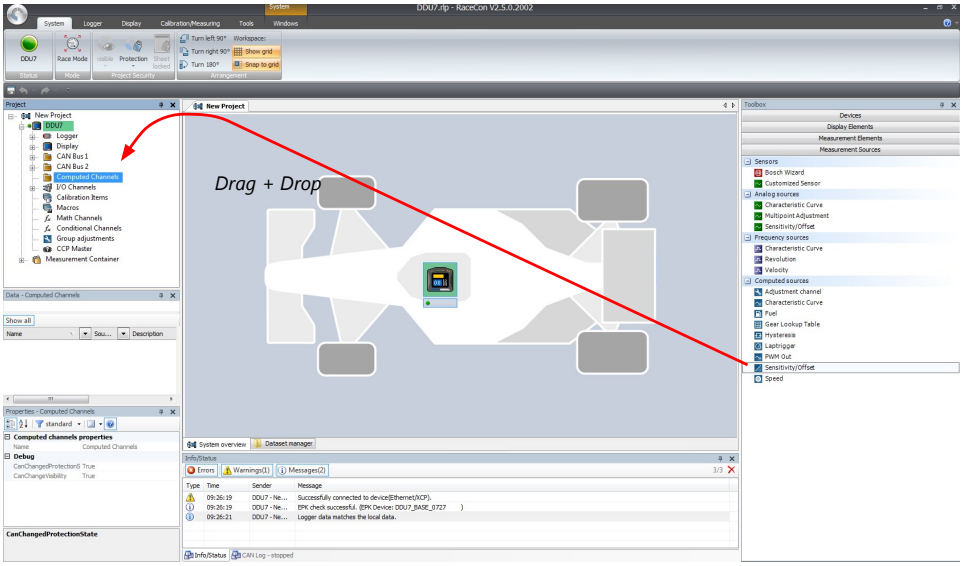
Numeric indicator



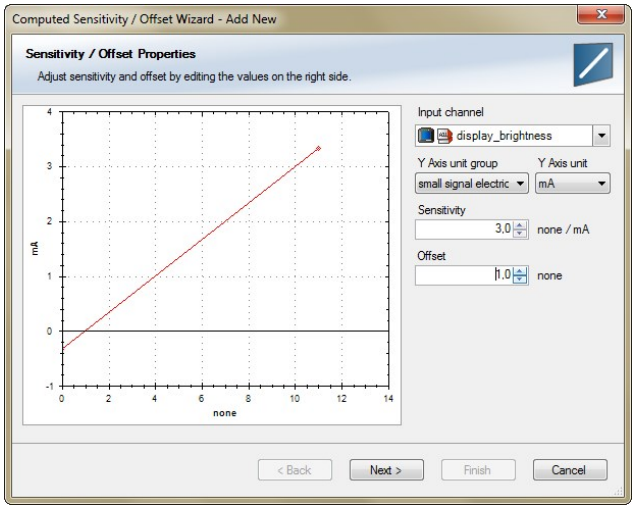
Oscilloscope (Chart)

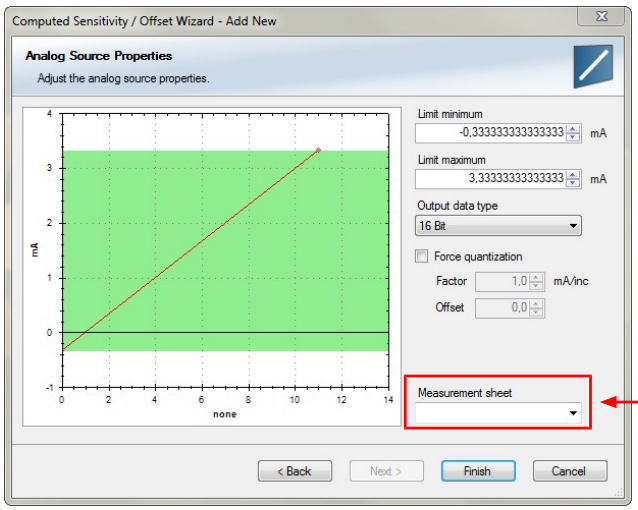
10.1.1 Automatic creation of measurement sheets

RaceCon can create measurement sheets automatically.
You can create and use measurement sheets with the MS 6 EVO as well as with all other devices connected to RaceCon.

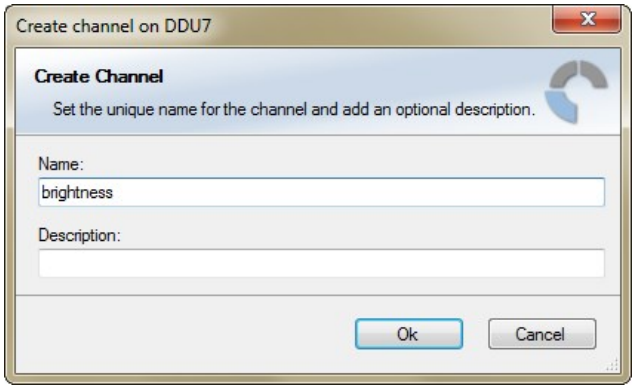


1. During the configuration of a measurement channel, select a measurement sheet from the list box or enter a name for a new measurement sheet.

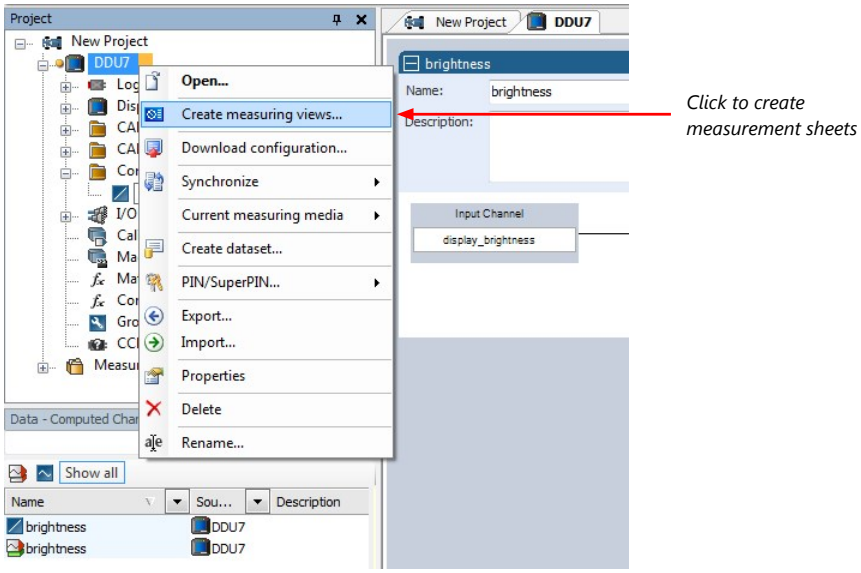




Select existing sheet from list or enter name of new sheet



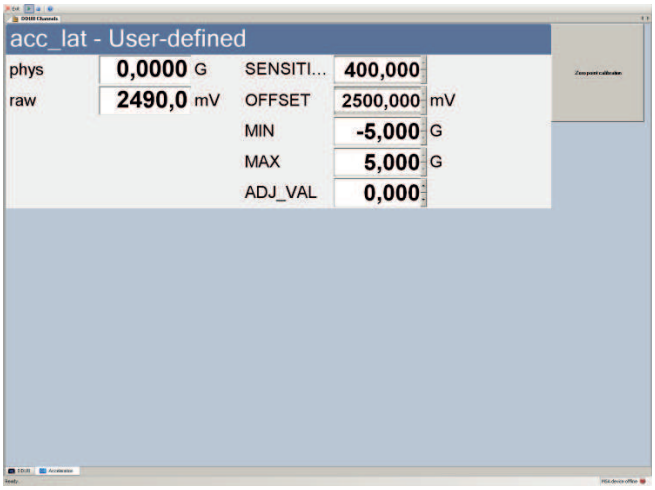
2. To create the sheets, right-click on MS 6 EVO and select ‘Create measurement views…’ from the MS 6 EVO context menu.



The automatically created sheet is inserted in the Project Tree under ‘Measurement Container’ and ‘Device Channels’. If the MS 6 EVO is connected to RaceCon and the status is green, live values of the channels are shown.

10.1.2 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’.

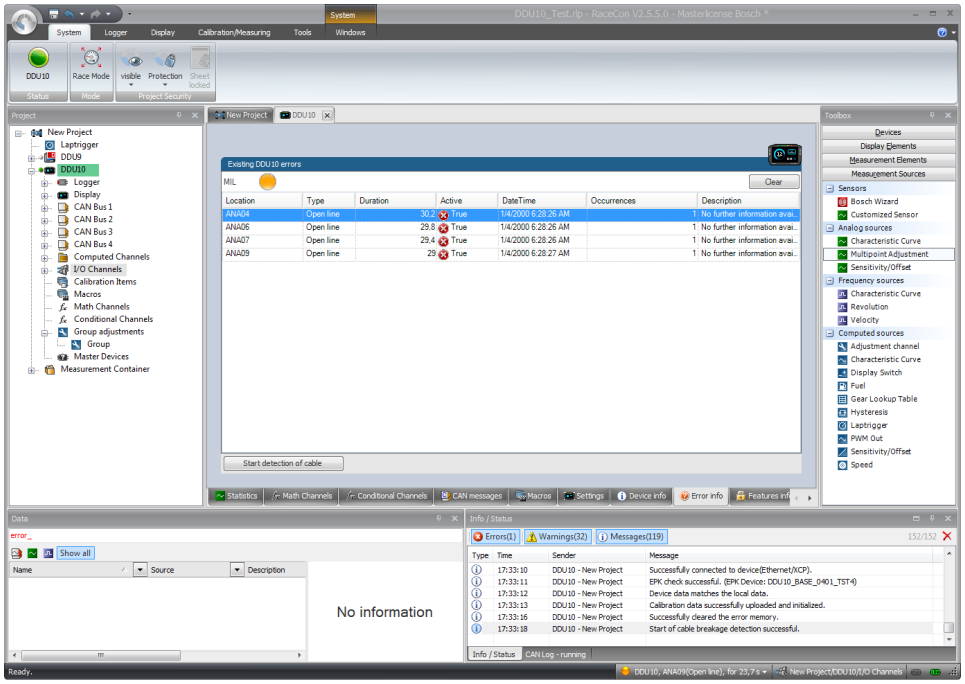

acc_lat - User-defined
phys
0,0000 G
raw
2490,0 mV
SENSITI...
400,000
OFFSET
2500,000 mV
MIN
-5,000 G
MAX
5,000 G
ADJ_VAL
0,000

11 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.

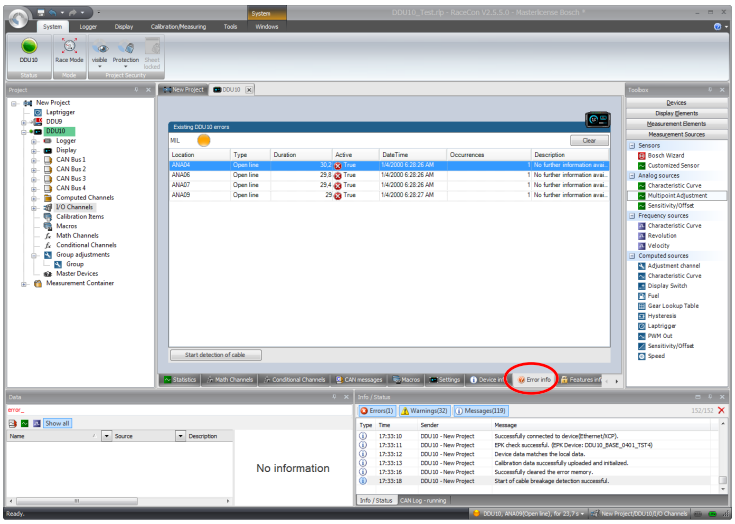
11.1 Error memory representation in RaceCon

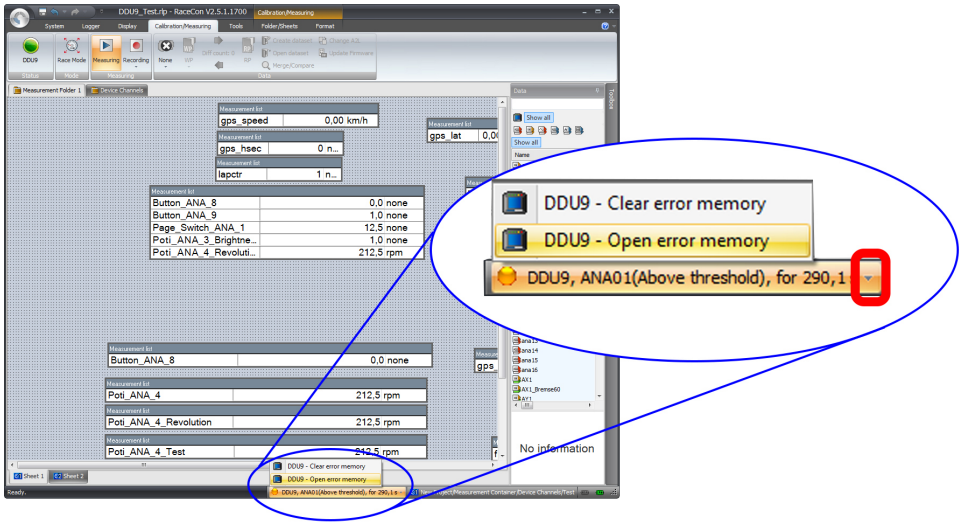
Bosch Motorsport devices feature an error memory. Information on errors can be visualized via RaceCon (online measurement) or can be transmitted via telemetry.



11.1.1 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.

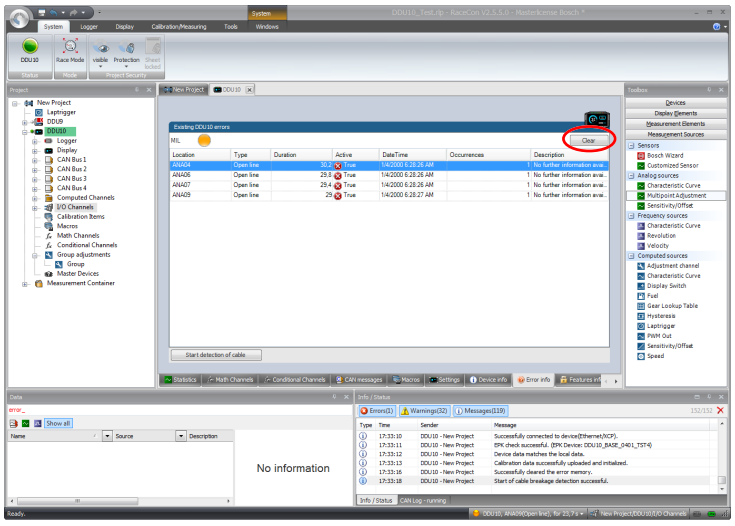
An error is deleted from the list when

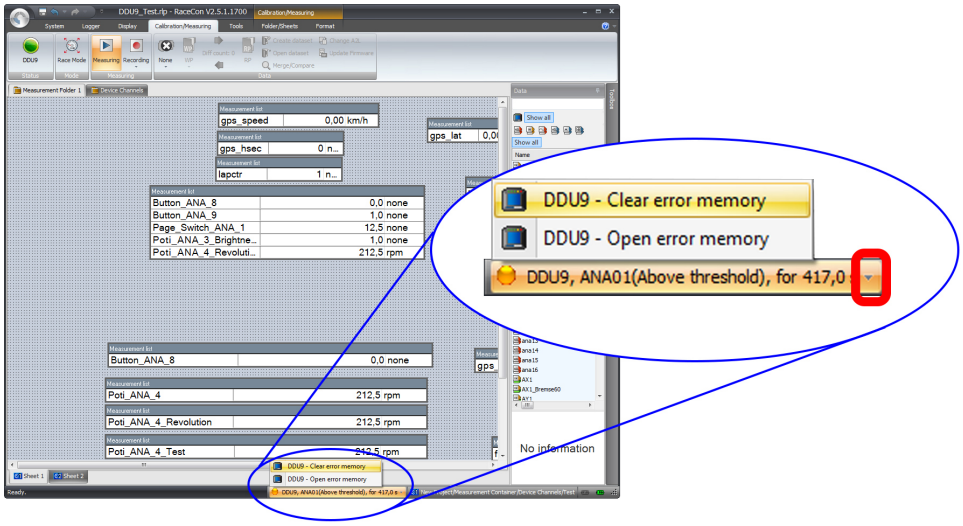
- the user actively clears the error memory
- the user updates the firmware

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

11.1.2 Clearing the error memory

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





11.2 Writing an Error

For the functional part of the MS 6 EVO system (MS 6 EVO -ECU) the error bits are related to the function and have to be distinguished if the function is activated. If an error is detected, the information may be shown as part of the error monitor in RaceCon, as display information and as measure channel. To support driver visibility, an activated error may activate also an output to enable the MIL-light (B_mildiag will be enabled).

CW_EM_xxx	Individual error related to a function
0	Error will not be stored in the monitor
1	Error is stored in the monitor
2	Not valid
3	Error is stored in the monitor and the MIL condition is switched on

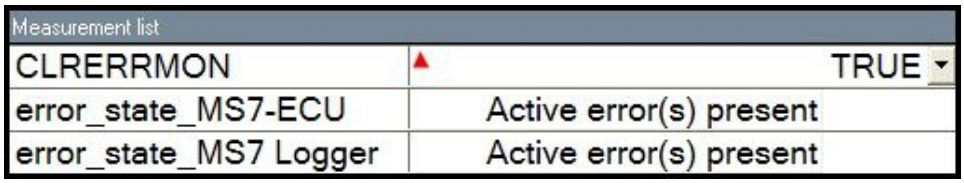
The single error bits may be collected in the error monitor.

11.3 Error Memory Properties

The following property is available for the error memory itself.

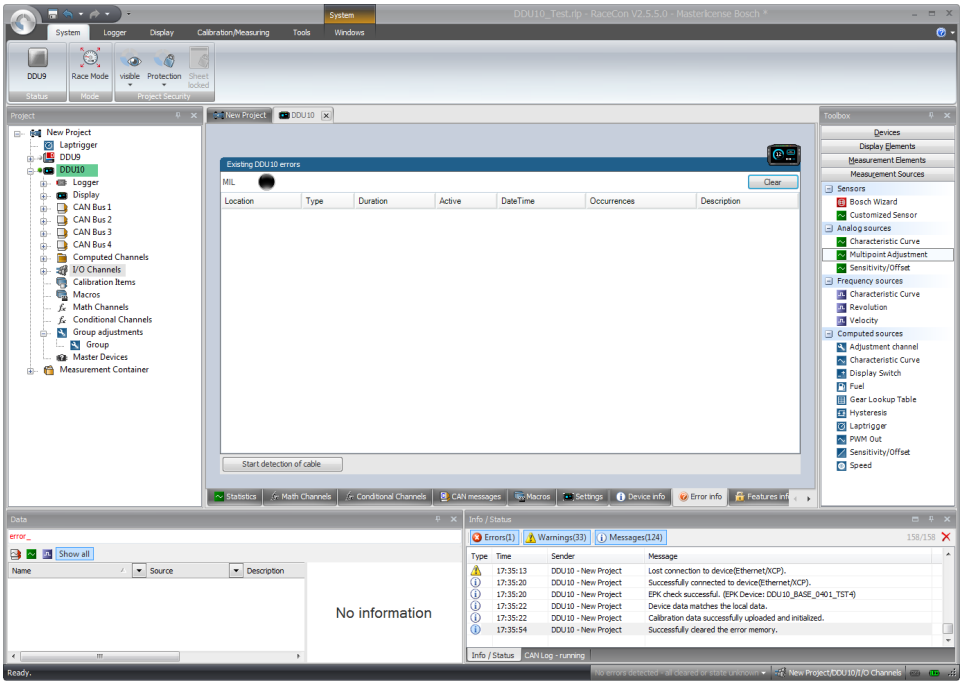
CLRERRMON	Reset of the error monitor
Error Status /device measurement label error_state	
0	No error present in the memory
1	At least one inactive error present in memory, no active errors
2	At least one active error present in memory

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

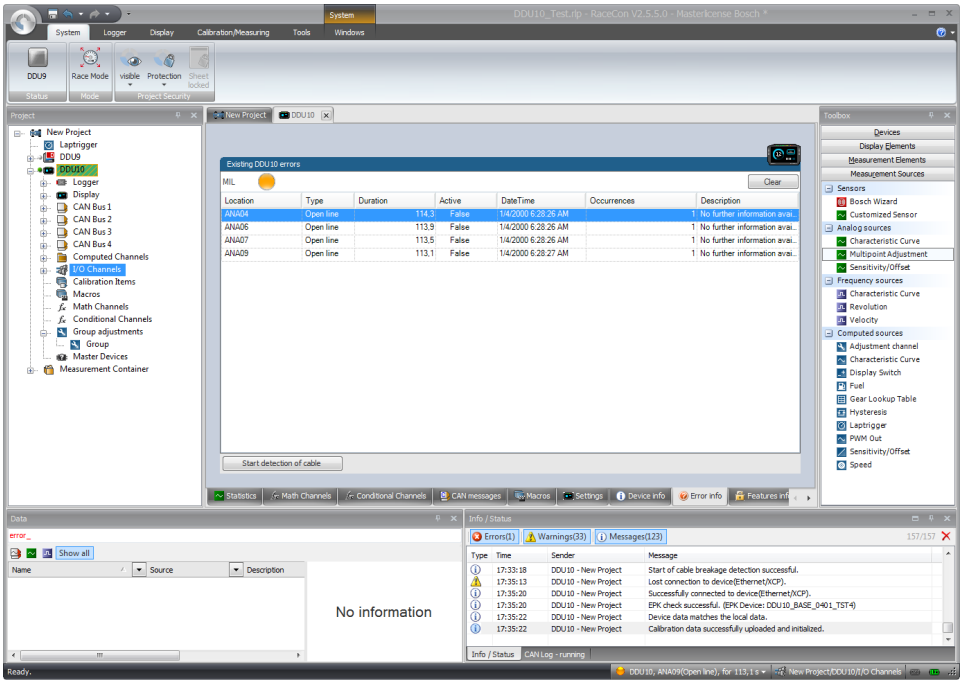


It is also represented by a color scheme within RaceCon (provided RaceCon is online with the system):

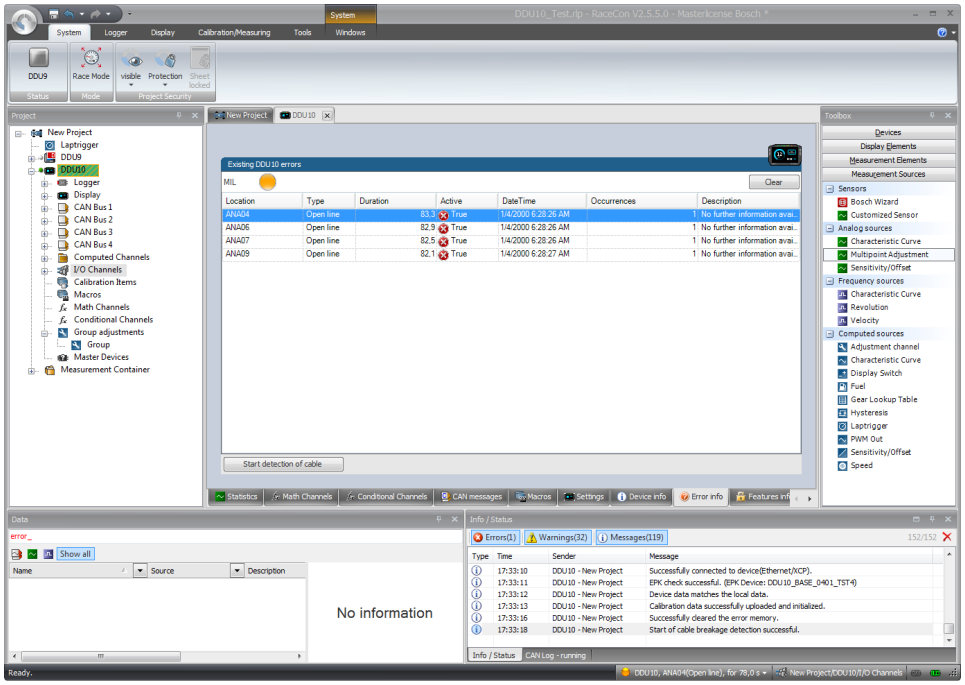
0 (no error present in memory)



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



2 (at least one active error present in memory)


DDU10_Test.rlp - RaceCon V2.5.5.0 - MasterLicense Bosch *
System
Logger
Display
Calibration/Measuring
Tools
Windows
DDU9
Race Mode
visible
Protection
Sheet locked
Status
Mode
Project Security
Project
New Project
Laptrigger
DDU9
DDU10
Logger
Display
CAN Bus 1
CAN Bus 2
CAN Bus 3
CAN Bus 4
Computed Channels
I/O Channels
Calibration Items
Macros
Math Channels
Conditional Channels
Group adjustments
Group
Master Devices
Measurement Container
New Project
DDU10
Existing DDU10 errors
MIL
Clear
Location
Type
Duration
Active
DateTime
Occurrences
Description
ANA04 Open line 83.3 True 1/4/2000 6:28:26 AM 1 No further information avai..
ANA06 Open line 82.9 True 1/4/2000 6:28:26 AM 1 No further information avai..
ANA07 Open line 82.5 True 1/4/2000 6:28:26 AM 1 No further information avai..
ANA09 Open line 82.1 True 1/4/2000 6:28:27 AM 1 No further information avai..
Start detection of cable
Statistics
Math Channels
Conditional Channels
CAN messages
Macros
Settings
Device info
Error info
Features info
Toolbox
Devices
Display Elements
Measurement Elements
Measurement Sources
Sensors
Bosch Wizard
Customized Sensor
Analog sources
Characteristic Curve
Multipoint Adjustment
Sensitivity/Offset
Frequency sources
Characteristic Curve
Revolution
Velocity
Computed sources
Adjustment channel
Characteristic Curve
Display Switch
Fuel
Gear Lookup Table
Hysteresis
Laptrigger
PWM Out
Sensitivity/Offset
Speed
Data
error_
Show all
Name
Source
Description
No information
Info / Status
Errors(1)
Warnings(32)
Messages(119)
152/152
Type
Time
Sender
Message
17:33:10 DDU10 - New Project Successfully connected to device(Ethernet/XCP).
17:33:11 DDU10 - New Project EPK check successful. (EPK Device: DDU10_BASE_0401_TST4)
17:33:12 DDU10 - New Project Device data matches the local data.
17:33:13 DDU10 - New Project Calibration data successfully uploaded and initialized.
17:33:16 DDU10 - New Project Successfully cleared the error memory.
17:33:18 DDU10 - New Project Start of cable breakage detection successful.
Info / Status
CAN Log - running
Ready.
DDU10, ANA04(Open line), for 78,0 s
New Project/DDU10/I/O Channels

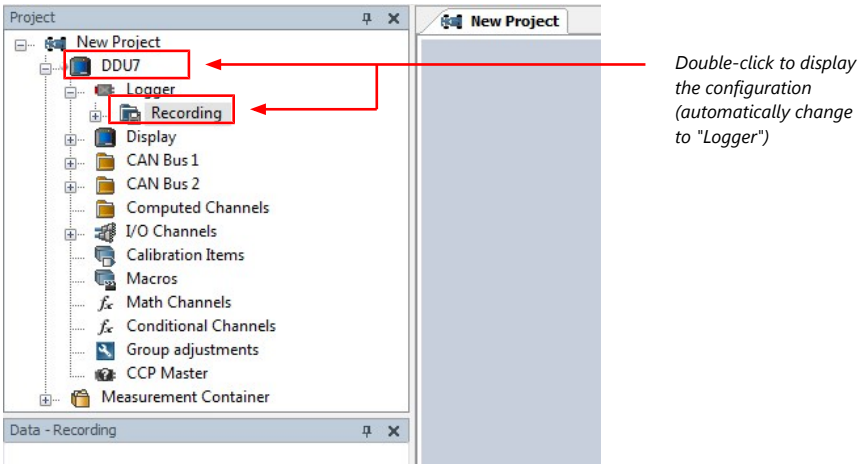
12 Recording

12.1 Features

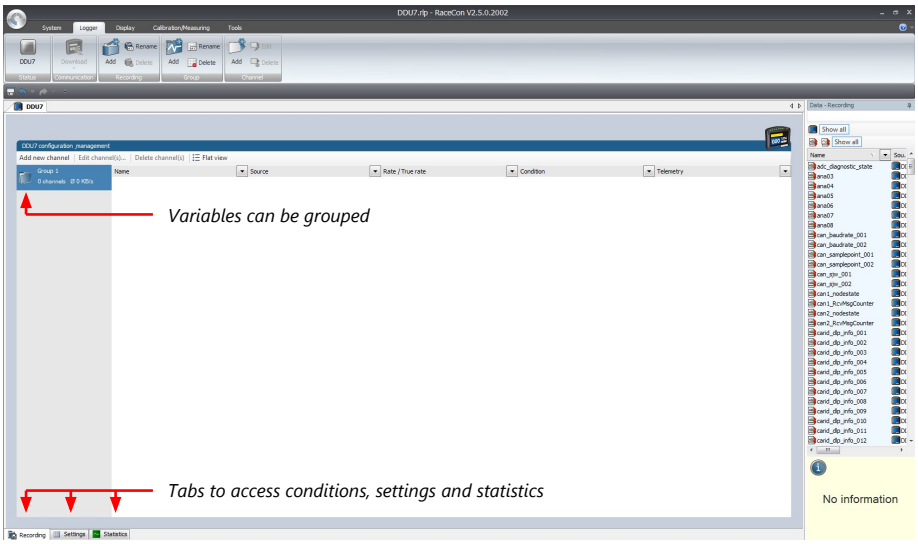
- Synchronized recording of MS 6 EVO analog and digital input channels, MS 6 EVO internal measurement channels, ECU data, Data from external sensor interfaces
- Up to two independent recordings
- Measurement rate 1 ms to 1 s
- Two global start conditions (thresholds)
- Up to 16 measurement conditions (fast-slow-switches)

12.2 Configuration of recordings

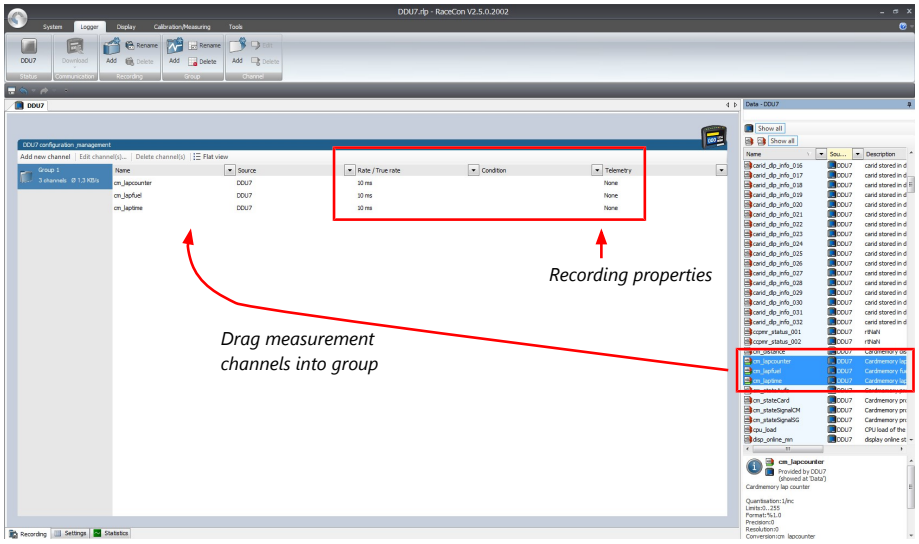
1. Expand the list of 'Loggers' by clicking on '+' in the MS 6 EVO Project Tree.



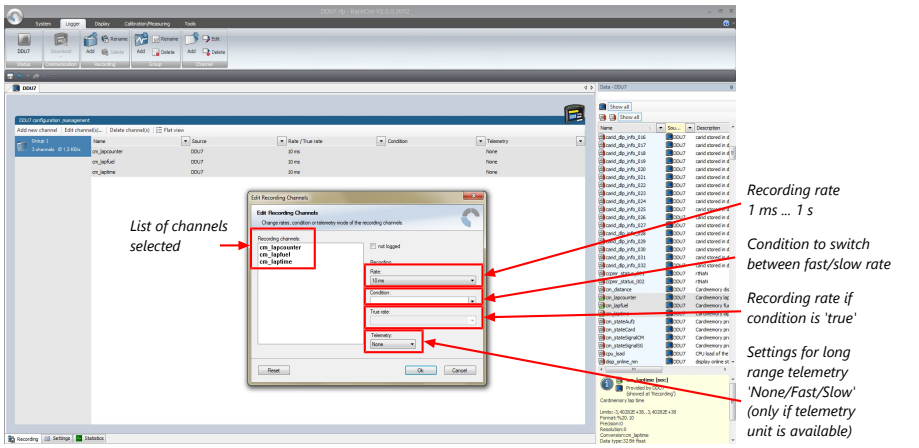
2. Double-click on 'Recording' in MS 6 EVO Project Tree. The recording configuration is displayed in the Main Area.



3. To add measurement channels to a recording, click 'MS 6 EVO' in the MS 6 EVO 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

MS 6 EVO 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).

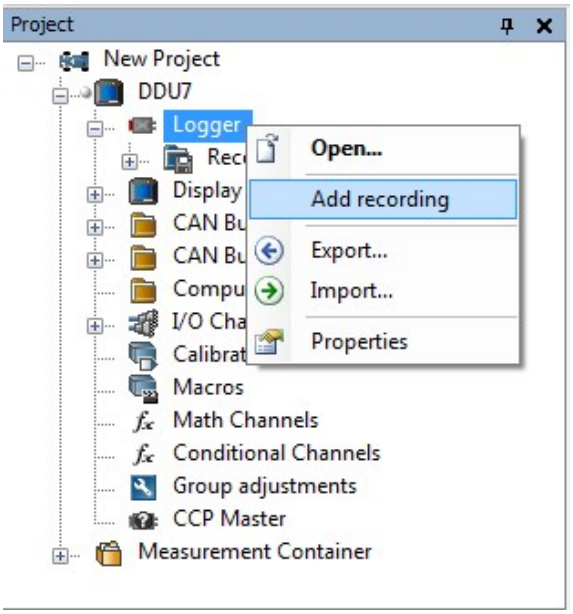
Transmission Scheme

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 [▶ 91]' for more information.

12.2.1 Adding a recording

MS 6 EVO supports up to two independent recordings.

To add a recording, select 'Add Recording' from the context menu of the Logger in the MS 6 EVO Project Tree.

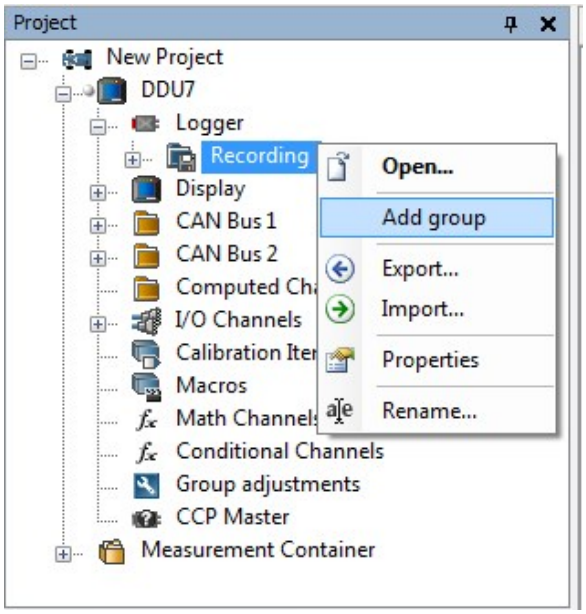


Maximum two recordings are possible. In the device software the 2nd recording is reserved for scruteneering data. This recording is invisible (protected).

12.2.2 Adding a recording group

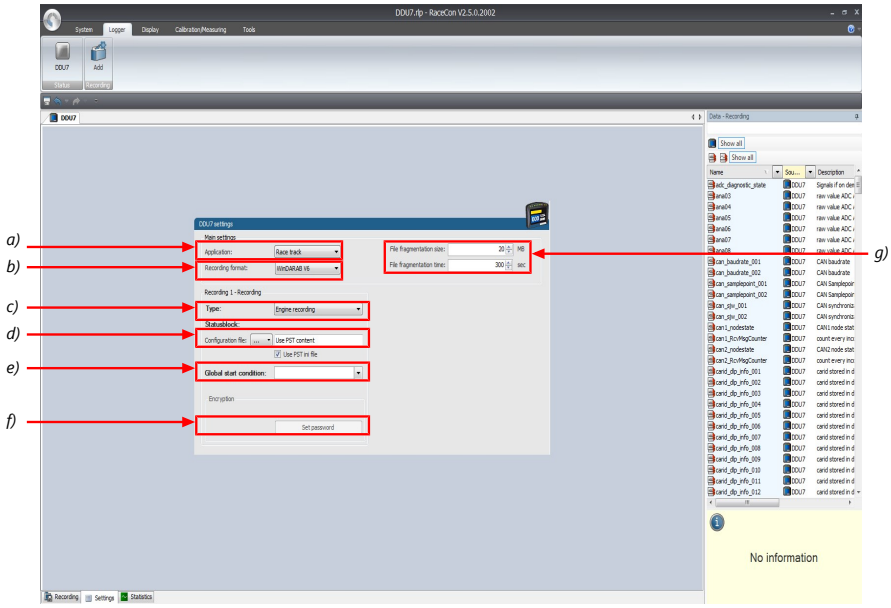
Recording channels can be grouped.

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.



12.2.3 Global settings

To display the global MS 6 EVO 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).

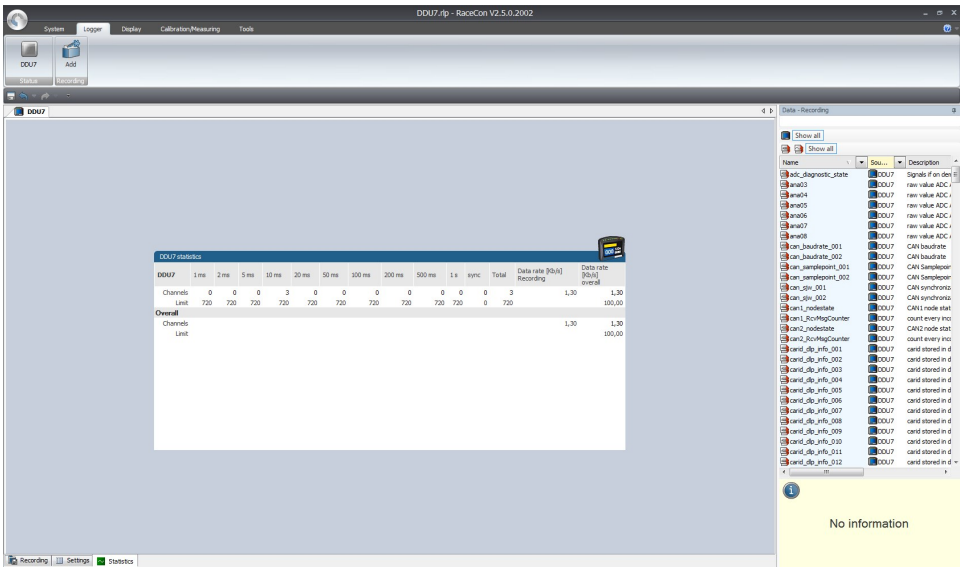
g) Setting for automatic fragmentation. Do not change!

12.2.4 Recording statistics

The tab ‘Statistics’ shows the channels’ allocation and their current data rate related to the transmission frequency of the MS 6 EVO 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 MS 6 EVO and limits the overall transmission speed.



12.2.5 Recording diagnosis

The channel ‘statectrl_ok’ of the MS 6 EVO 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	-

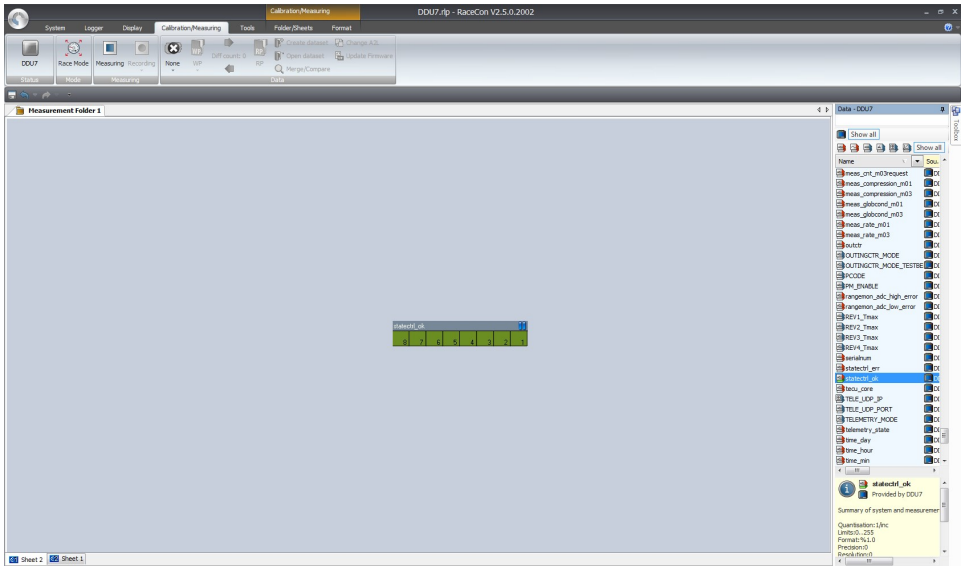
Content of status bits

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.

12.2.6 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, MS 6 EVO 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.

12.2.7 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'.

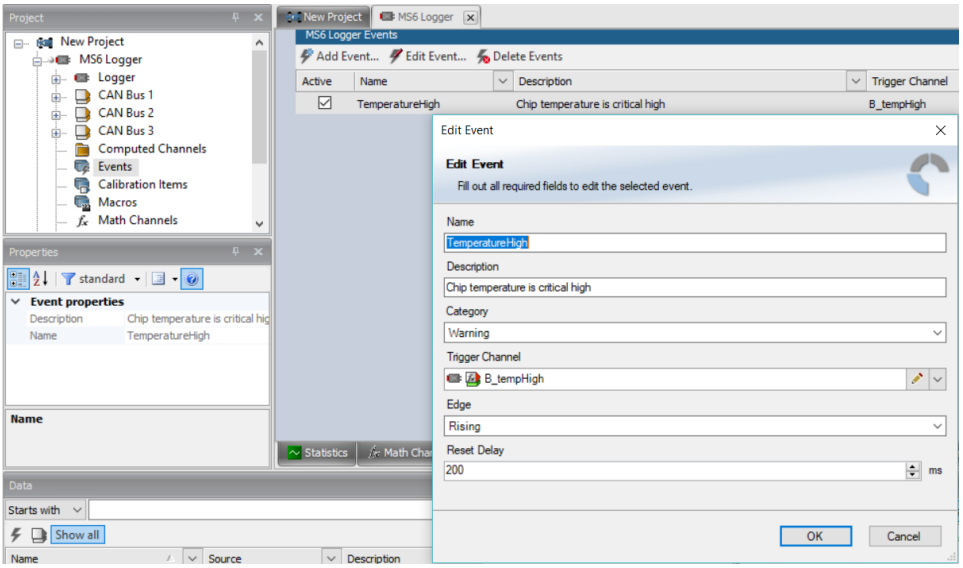
Measurement label	Function
card_part1_size	Size of the first logging data partition in MB.
card_part2_size	Size of the second logging data partition in MB. Attention, second logging can also be stored on first partition, depending on chosen settings (Logger -> Settings).
ftp_UserLoggedIn	This measurement 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.

12.3 Event logging

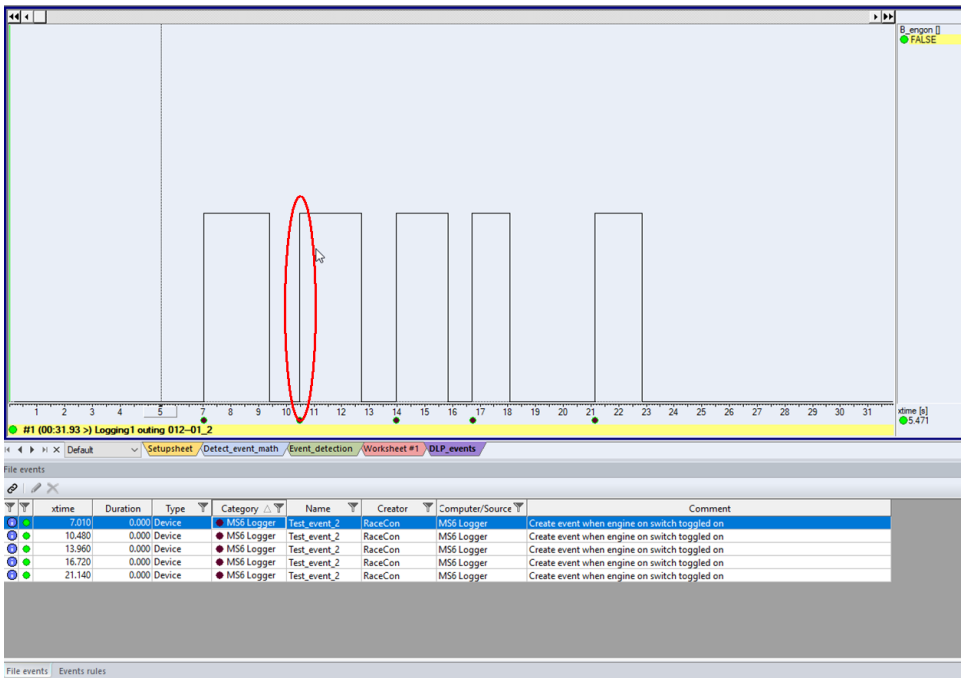
Event Logging implements the possibility to observe a channel if short spikes are expected. With Event Logging, every occurrence of a user defined threshold (more complex conditions are possible) leads to an event being raised. It is listed in a table along with its time stamp, its ID and even with a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

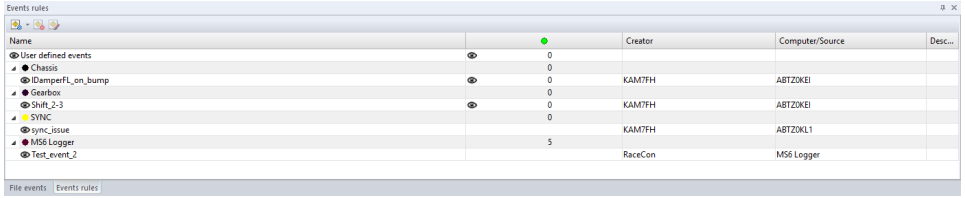
Events are stored as text in logging data and displayed in WinDarab like Darab-Events. Possible use cases are error entry, etc.

Configuration in RaceCon:



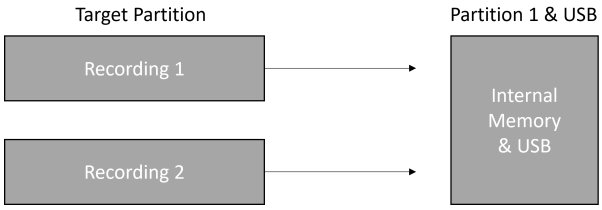
Display in WinDarab:





12.4 Data Logger and USB recording

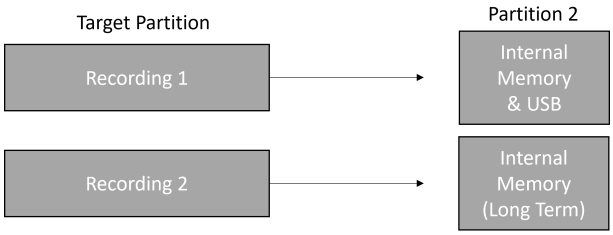
Default settings:



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.

12.5 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 DATA_USB enables USB recording. To activate Software Upgrade DATA_USB, enter the license key as described in the chapter 'Feature activation' [▶ 42].

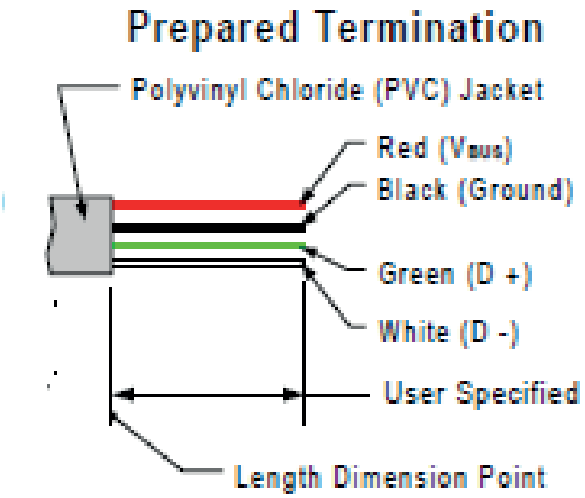
For USB recording, Software Upgrade FULL_LOG_1 should also be enabled.

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:

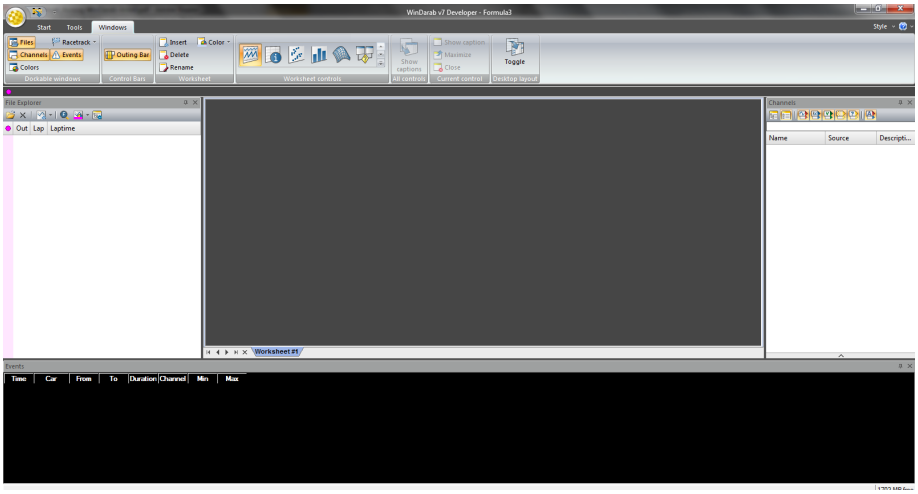
In RaceCon, select 'Tools' - 'Extras' and choose 'Format USB stick'.

Press 'Format'.

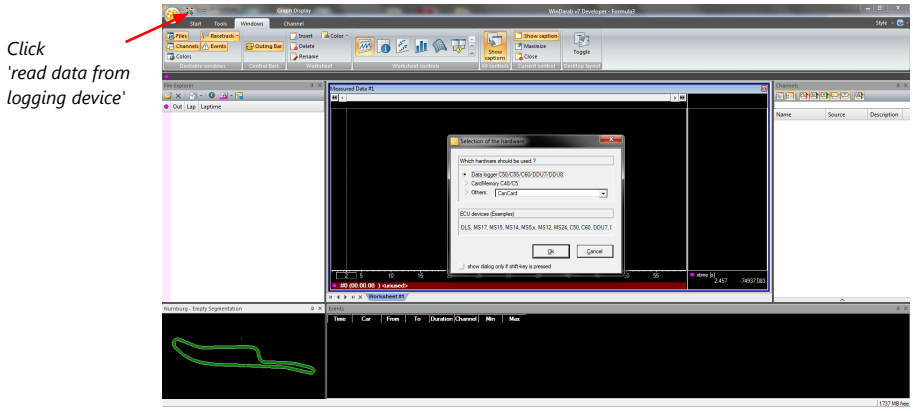
An USB device is recognized by Windows as a 'storage medium', but it can only be initialized with RaceCon and read with WinDarab.

12.5.1 Recording data on USB device

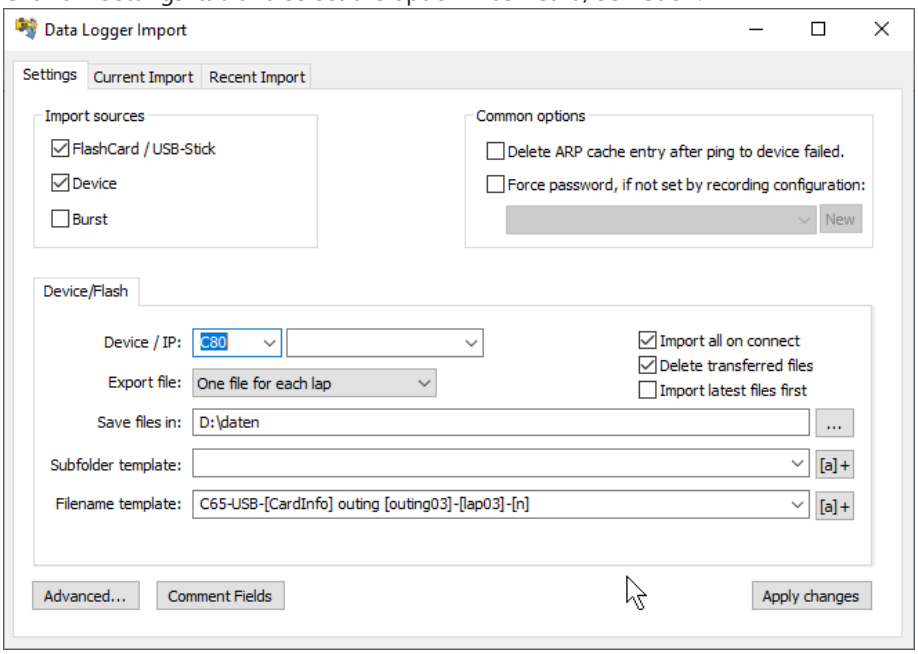
1. Plug an USB device to MS 6 EVO.
2. Prepare a recording configuration in RaceCon.
3. Power on the system and connect with RaceCon to the vehicle.
4. Download the configuration to the MS 6 EVO.
5. Record measurement data. If an USB device is present, the MS 6 EVO stores the data in parallel on the internal 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.

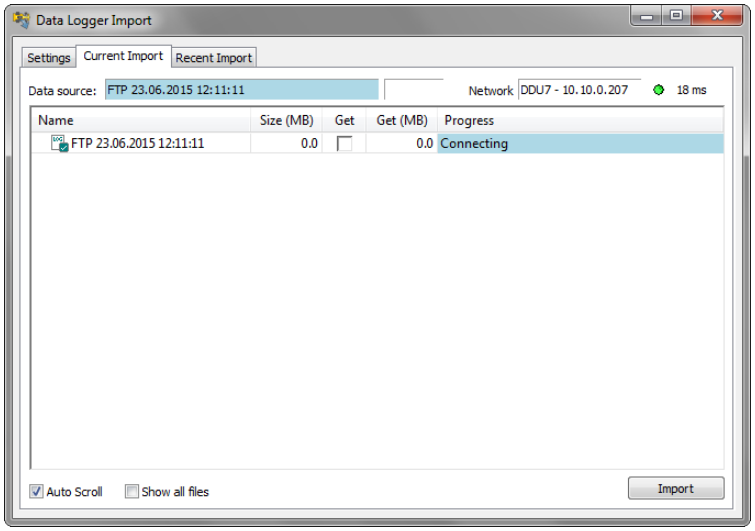


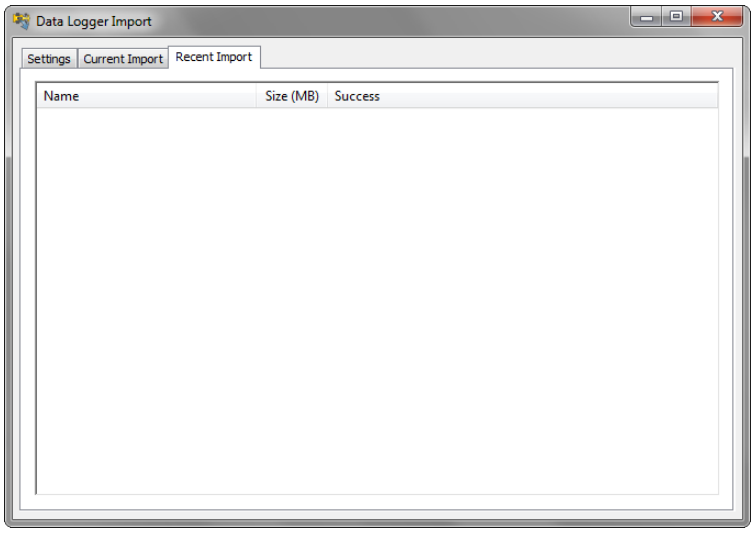
11. Click on ‘Settings’ tab and select the option ‘Flash Card/USB Stick’.



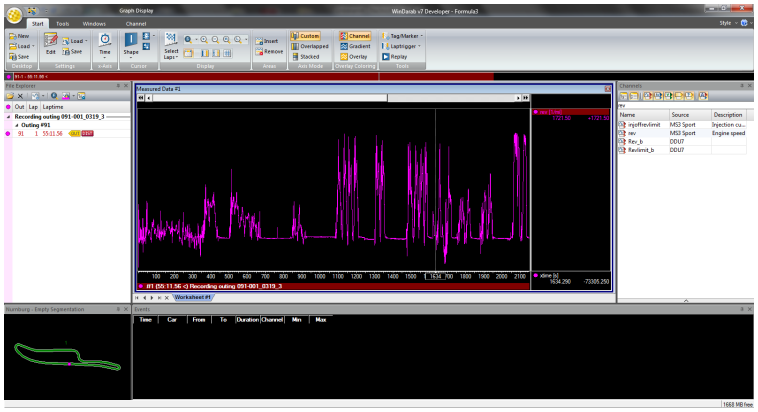
12. Activate ‘Apply changes’.

    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.

12.5.2 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 MS 6 EVO 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!

12.5.3 Troubleshooting

When no data on the USB device is recorded:

Configure the measurement label **usb_mediastate** on a RaceCon measurement view or on a MS 6 EVO 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 MS 6 EVO 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 MS 6 EVO.
6: Error: Media corrupt	The USB device is not in valid BFS format. (Hint: Re-format the USB device in RaceCon.)

13 Lap Trigger

13.1 Lap trigger (timing beacon)

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

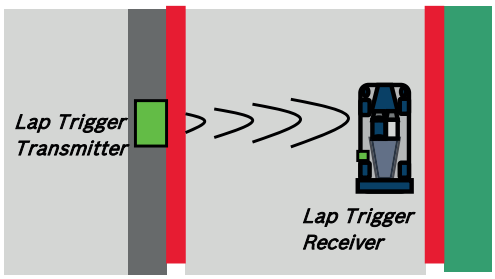
- Vehicle lap time measurement
- Calculation of lap-dependent functions (lap fuel consumption, min/max values)
- Calculation of lap distance dependent functions
- Control of data logging system

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)



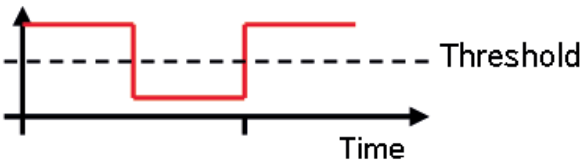
13.1.1 Electrical trigger signal

In MS 6 EVO all sources of measurement channels can be used as trigger signal.

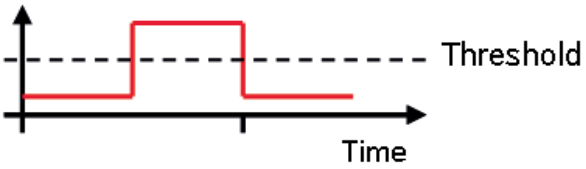
- Analog input
- Digital input
- CAN input

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)

13.1.2 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) 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.

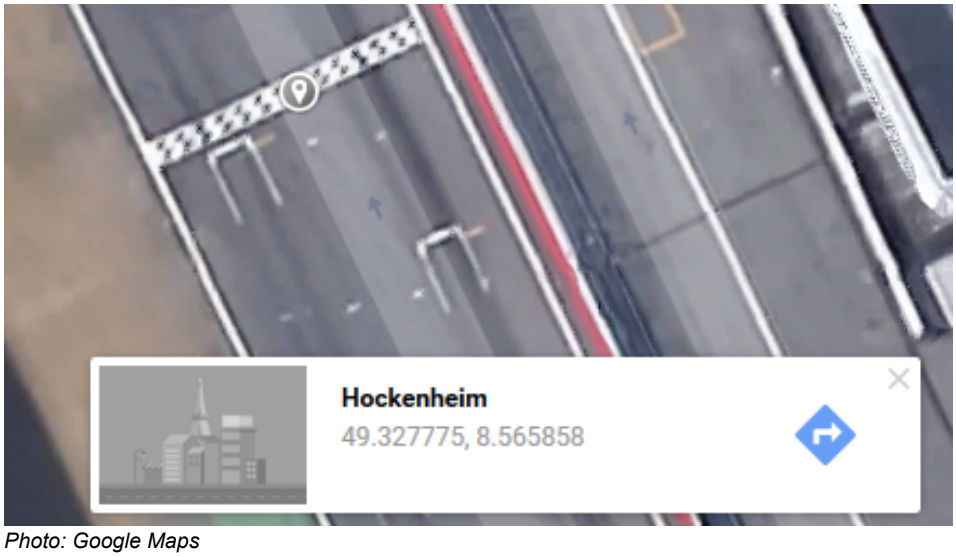


Photo: Google Maps

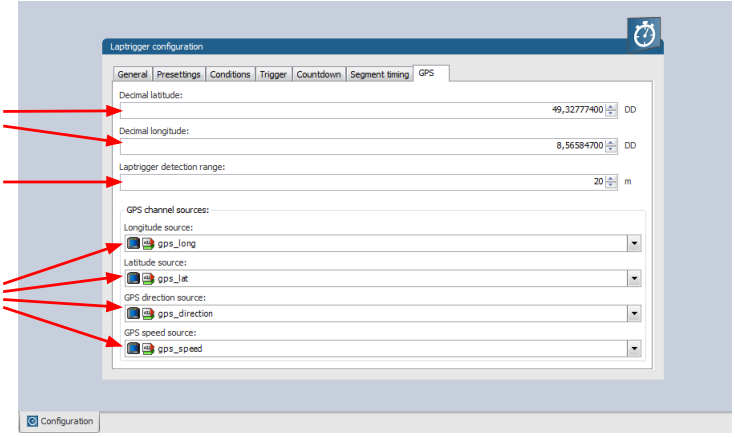
The detection range defines the radius of a circle around the detection point in which the lap trigger can be set. The lap trigger will be set as soon as the distance between the car and the detection point has reached its smallest peak. By this function an imaginary finishing line is calculated inside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

13.1.3 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.

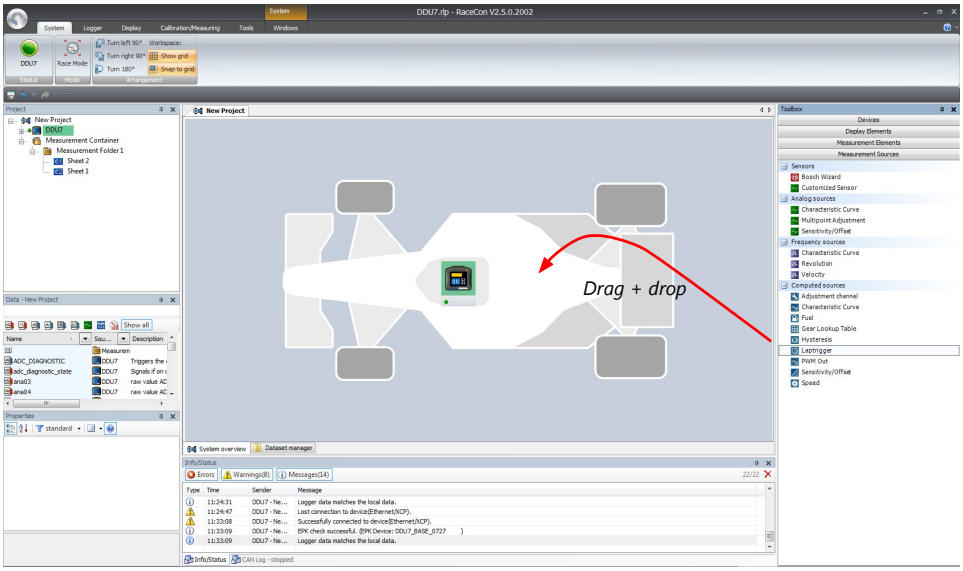
13.1.4 Forced triggers

Lap distance based insertion of ‘forced trigger’.

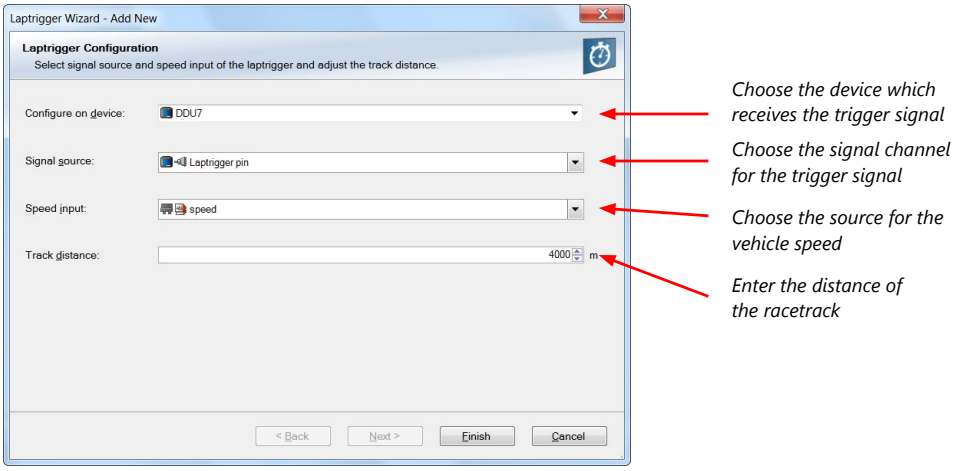
Under race conditions, trigger signals are sometimes missed. Software functionality introduces ‘forced trigger’.

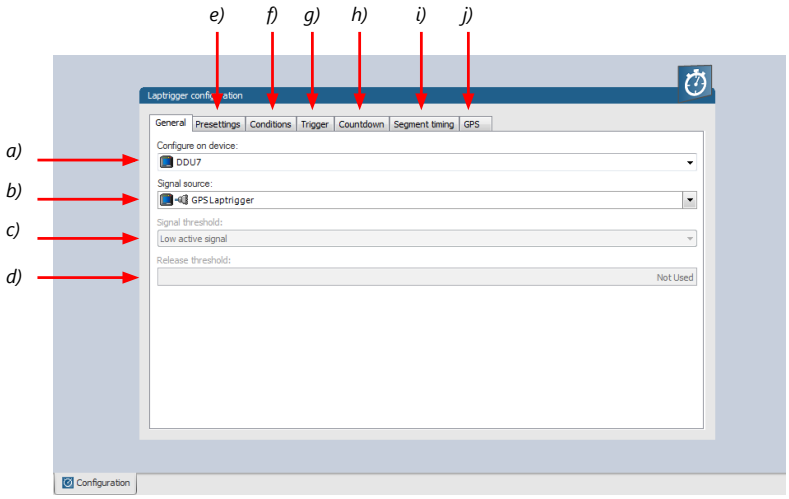
13.1.5 Setting up a lap trigger

1. Click ‘Measurement Sources’ in Toolbox.
2. Drag ‘Laptrigger’ into ‘System Overview’. Do not drop it on ‘MS 6 EVO’!



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.

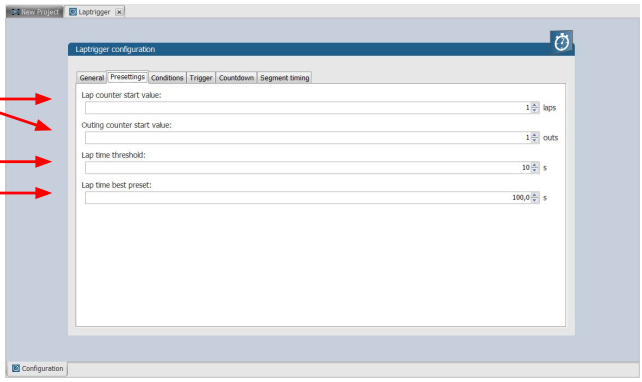
j) Define settings for a GPS lap trigger. See chapter 'GPS lap trigger' for details. Only applicable if the signal source is set to 'GPS lap trigger'.

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'

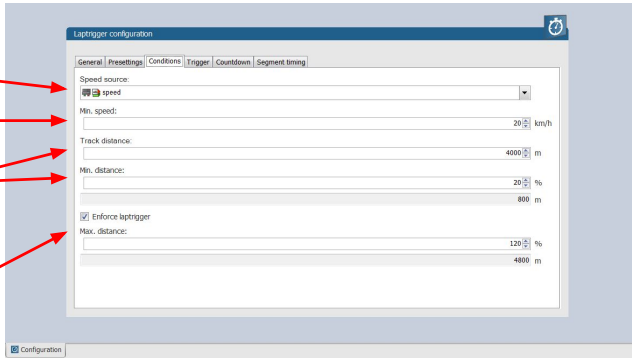


Change signal for vehicle speed, if desired.

Enter minimum speed for trigger release.

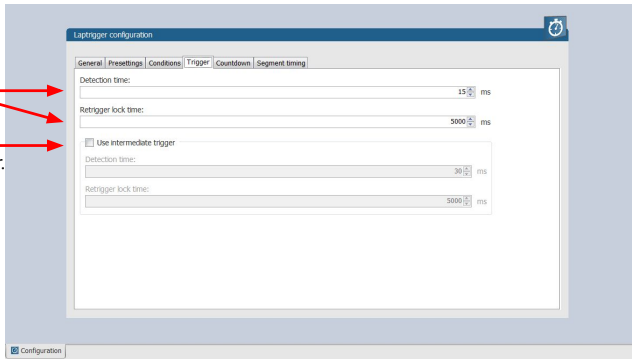
Define settings for distance based retrigger protection.

Define settings for distance based forced trigger.

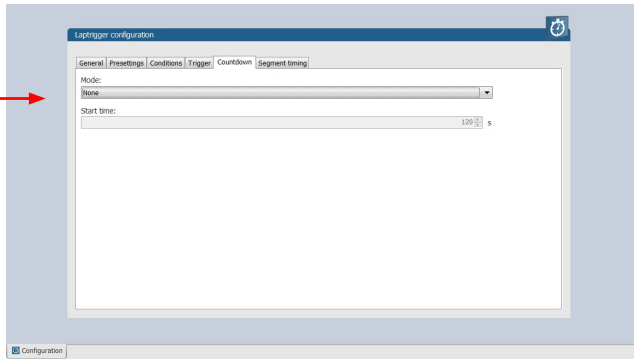


Define settings for lap timing (main trigger).

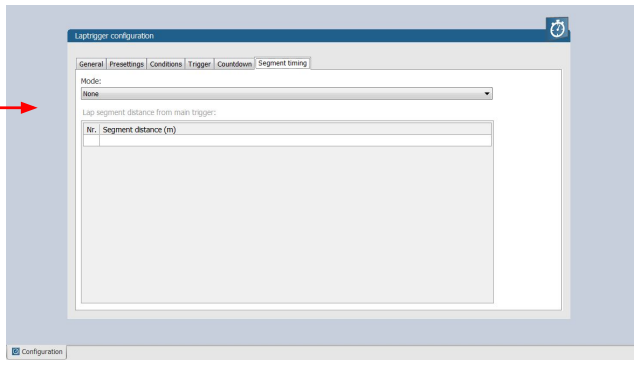
Define settings for sub trigger. Not applicable with a GPS lap trigger.



Define settings for countdown timer.



Define settings for segment timing.

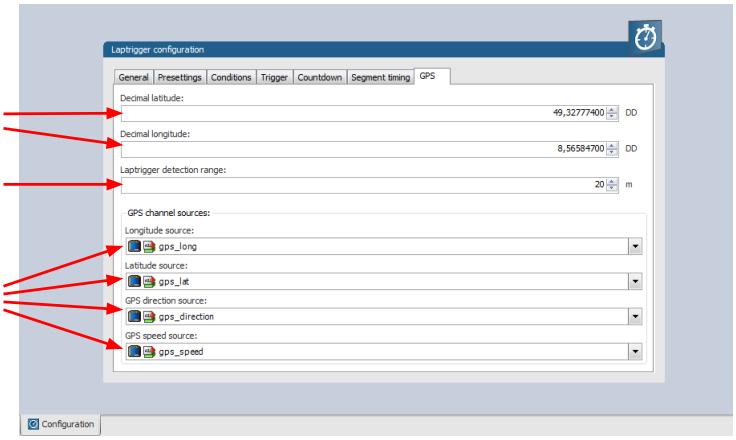


Only applicable for a GPS Laptrigger

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.

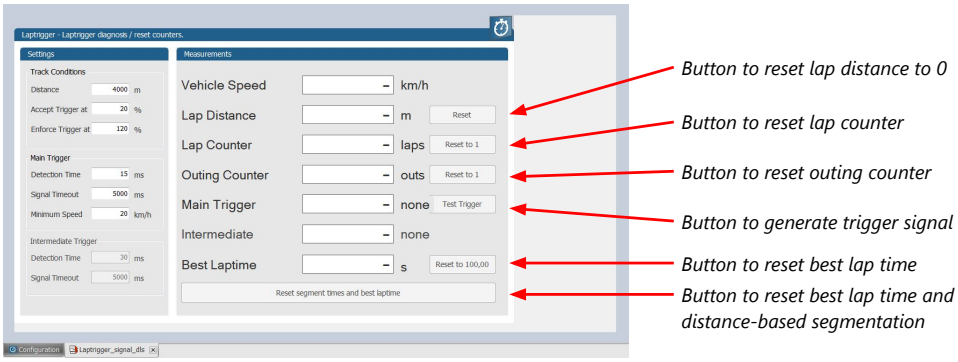


13.1.6 Lap trigger channel diagnosis/counter reset

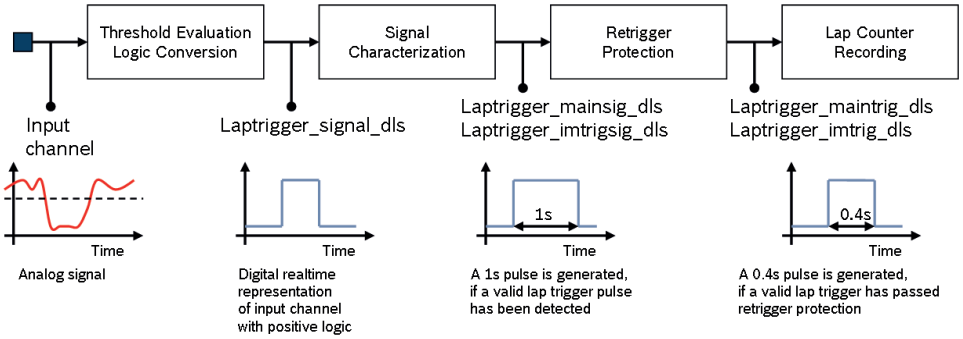
To display a quick lap trigger channel diagnosis and to reset counters use the diagnosis page in RaceCon. Any ‘Laptrigger_xxx’ channel can be displayed.

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

A diagnosis window opens in Main Area.

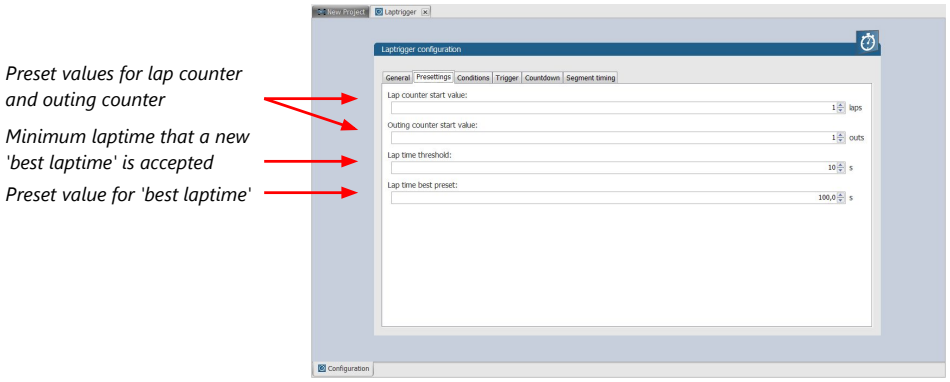


Lap trigger diagnosis scheme

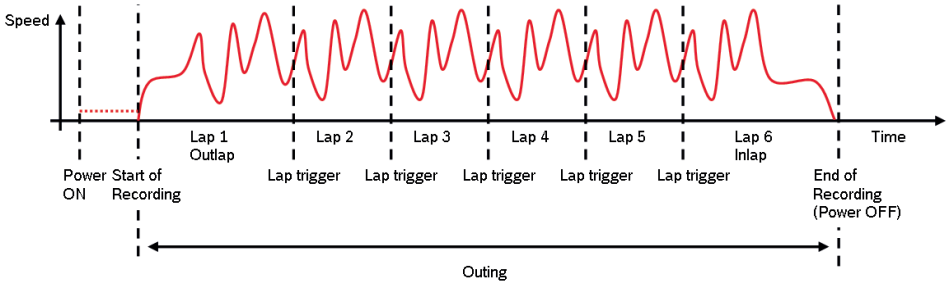


13.1.7 Lap trigger presettings

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



13.2 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 [▶ 109])

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’.

13.3 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

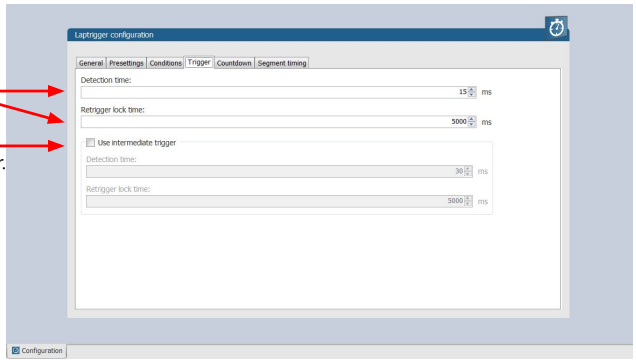
13.3.1 Time based retrigger protection

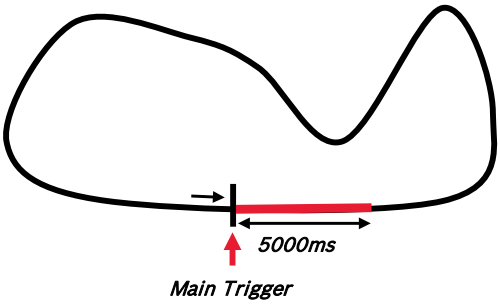
Trigger is locked for 5 s after main trigger was received.

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.





13.3.2 Distance based retrigger protection

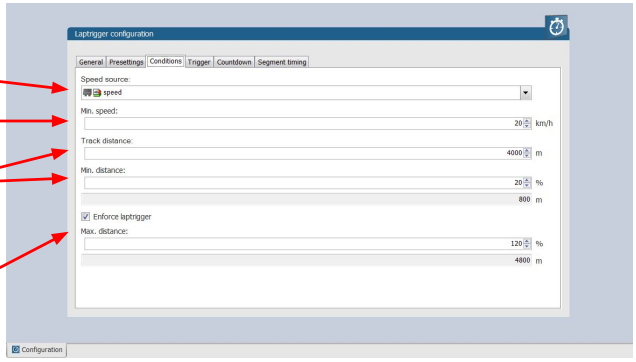
Trigger is locked until configured min distance (i.e. 80 % → 3200 m) of track distance (i.e. 4000 m) has been covered. To deactivate distance based retrigger protection, set min distance to 0 %.

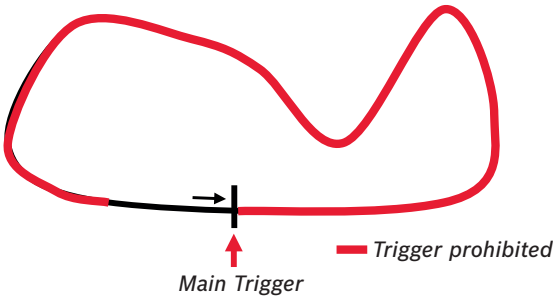
Change signal for vehicle speed, if desired.

Enter minimum speed for trigger release.

Define settings for distance based retrigger protection.

Define settings for distance based forced trigger.





13.3.3 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).

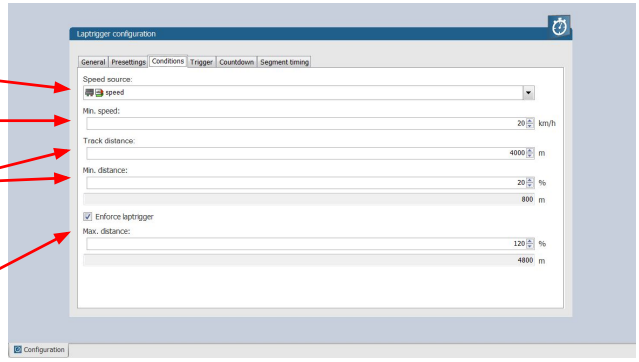
To deactivate distance based forced triggers, uncheck box.

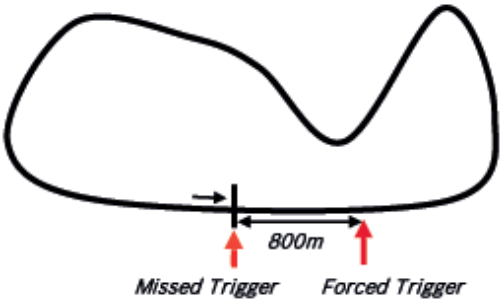
Change signal for vehicle speed, if desired.

Enter minimum speed for trigger release.

Define settings for distance based retrigger protection.

Define settings for distance based forced trigger.





13.4 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

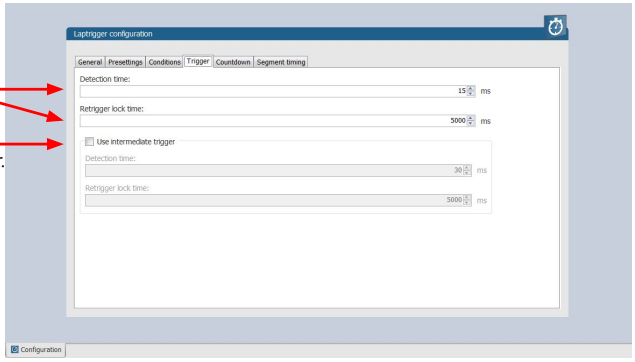
13.4.1 Sub trigger mode

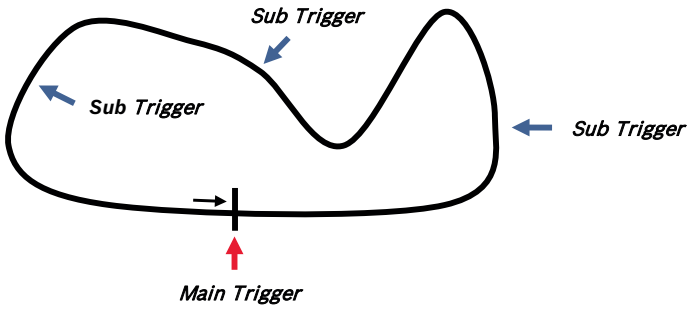
Using main trigger (20 ms pulse) at Start-Finish-Line. 3 sub triggers (40 ms pulse) positioned at 1,000 m, 2,000 m and 3,000 m.

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.

13.4.2 Distance mode

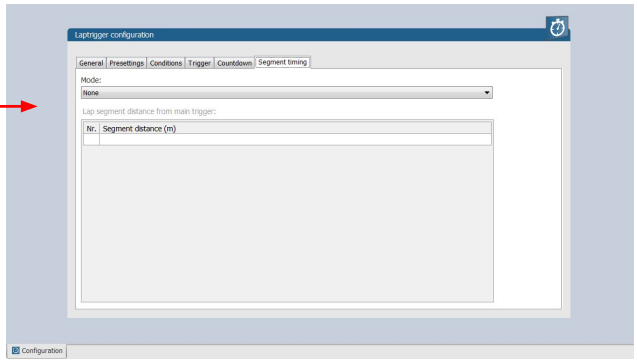
Using main trigger (20 ms pulse) at Start-Finish-Line.

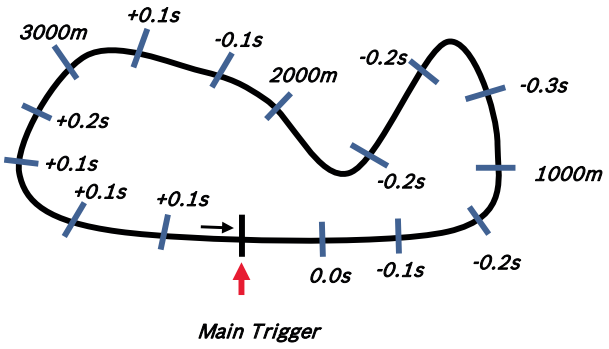
Set ‘Mode’ to ‘Distance’ and enter desired segment distances.

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.



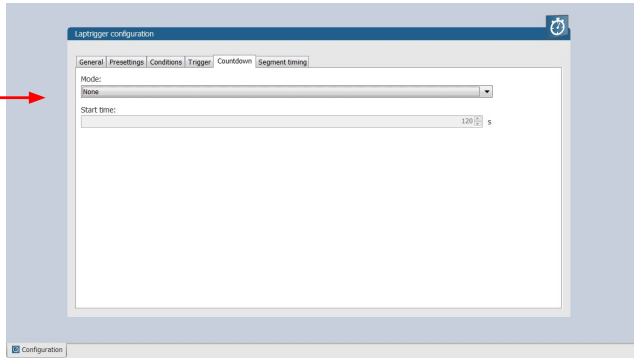


13.5 Countdown timer

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.

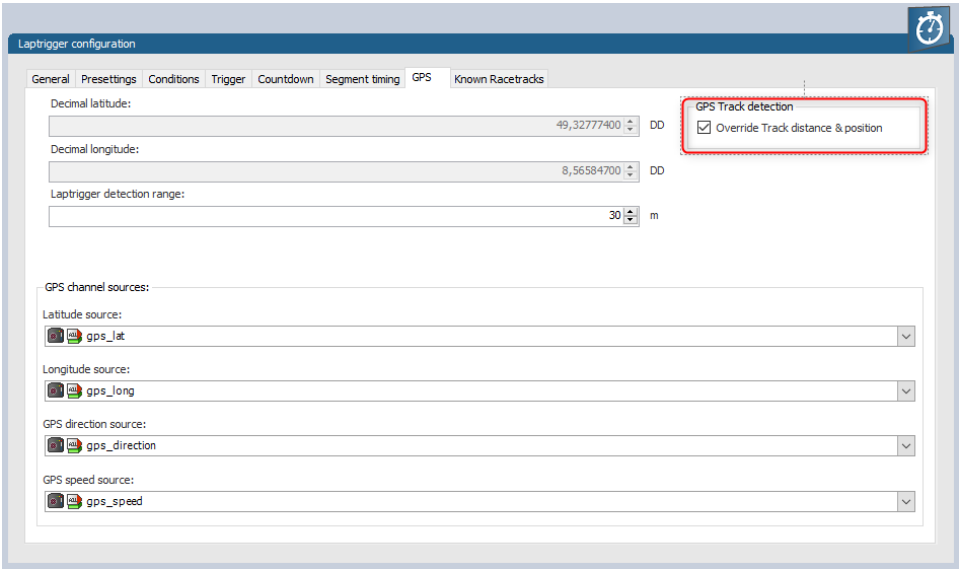


13.6 Automatic GPS Track Detection

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.

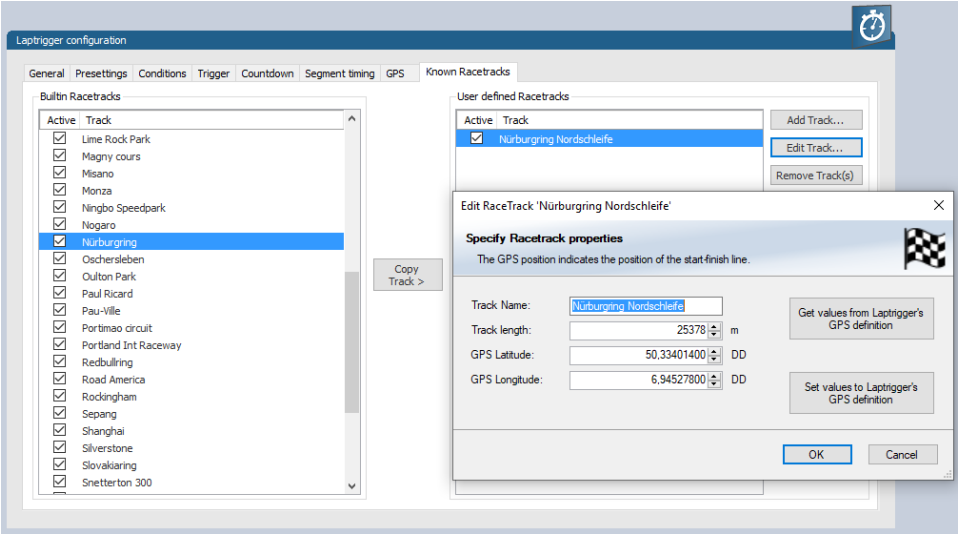


The known Racetrack list can be found in the lap trigger configuration menu within the Known Racetracks tab. It contains a built-in list and a user-defined list. Each track can be activated or deactivated with the checkbox to manually set the variant if needed.

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:

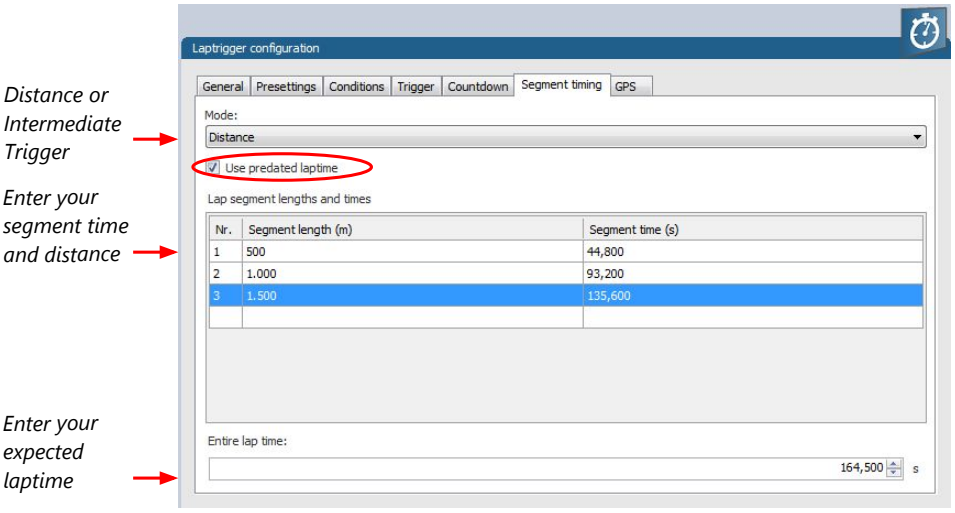
Laptrigger_trackdet_id	Signal will show the track name from the Racetracks list as an enumeration or can show the ID number as raw value. The enumerated name will also be visible in the logging.
Laptrigger_trackdet_laplen	Track length of the used track variant
Laptrigger_trackdet_lat	Latitude GPS coordinate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

13.7 Predicted Laptime

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

13.7.1 Setting up the predicted laptime

To use the predicted laptime function you need to set up a laptrigger as described in the chapter Lap Trigger [▶ 100]. 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.

13.7.2 Functionality and channel outputs

Following output channels are generated by the predicted laptime function.

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

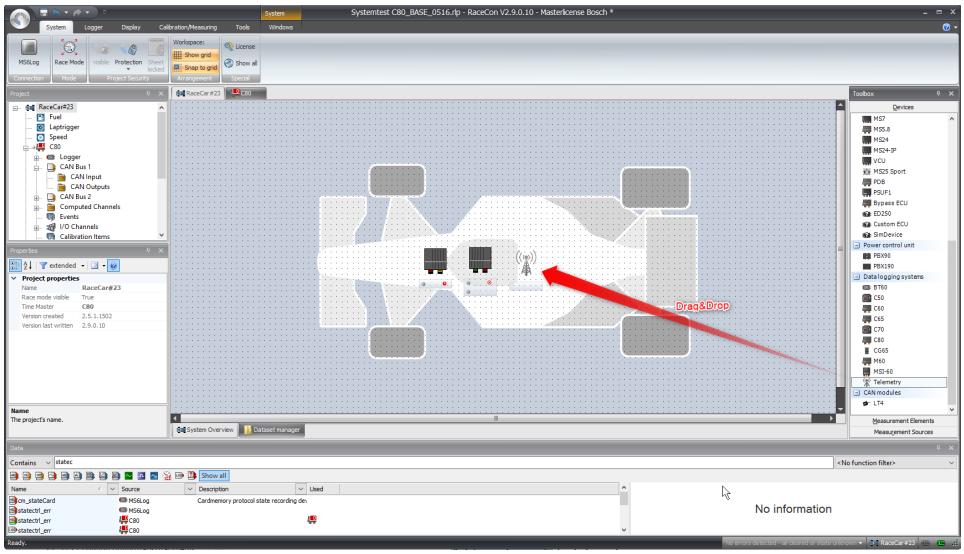
The channel Laptrigger_lapdiff_pred_dls is updated as soon as the main lap trigger is received. Both other channels are updated as soon as the next segment distance is travelled, or the next intermediate trigger is received.

14 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

14.1 Software setup

Drop Telemetry from Toolbox into system overview.



Adding channels to telemetry

1. Expand the list of ‘Loggers’ by clicking on ‘+’ in the MS 6 EVO Project Tree.
2. Double-click on ‘Recording’ in MS 6 EVO 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.

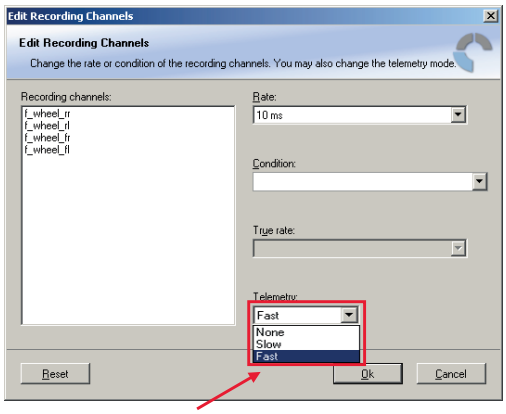
Using fast block/slow block transmission

MS 6 EVO 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

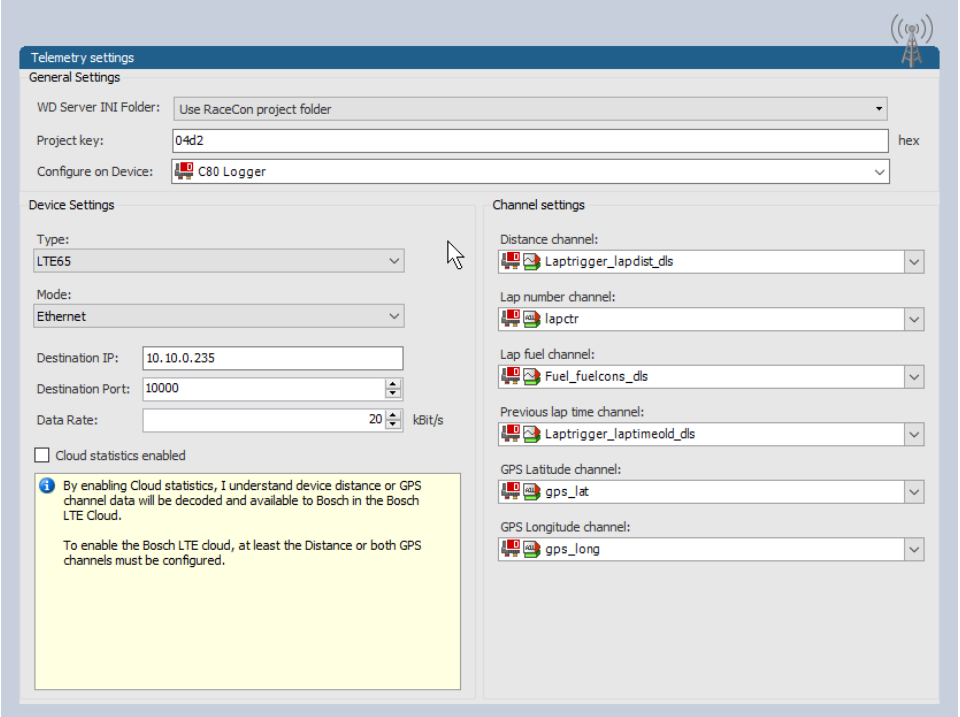
If the maximum bandwidth of a block is reached, a warning will be displayed. To fix this problem you can view the allocation of the channels and data rate in the 'Statistics' tab of the Main Area.

14.2 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).

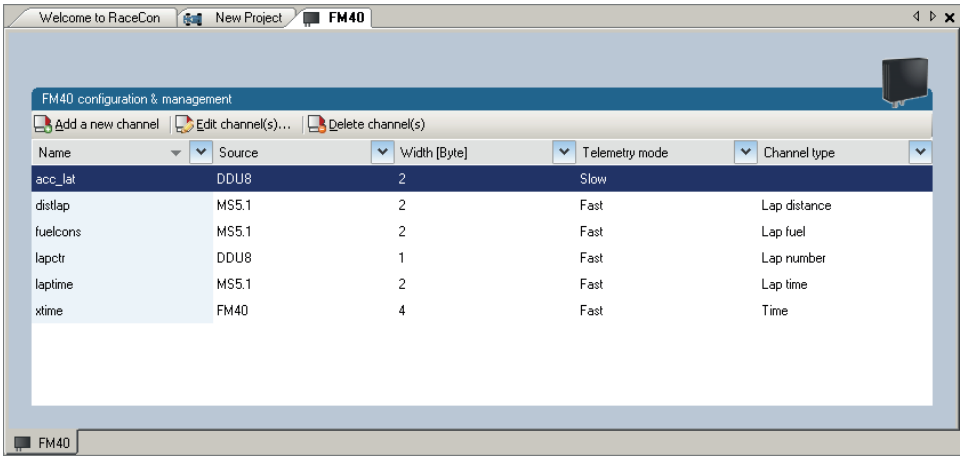
For displaying the position of the car in the cloud, additionally GPS-position and lapdist can be send to the cloud, this is activated with the checkbox "cloud statistics".



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.

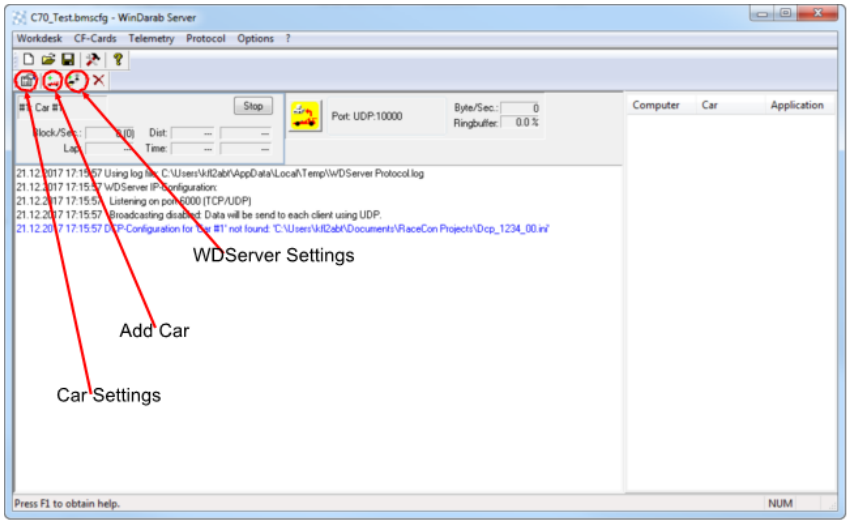
Measurement channel	Function
Distance	Running distance of current lap
Lap number	Lap number of current lap
Lap fuel	Fuel consumption of last lap completed
Lap time	Exact lap time of lap completed

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

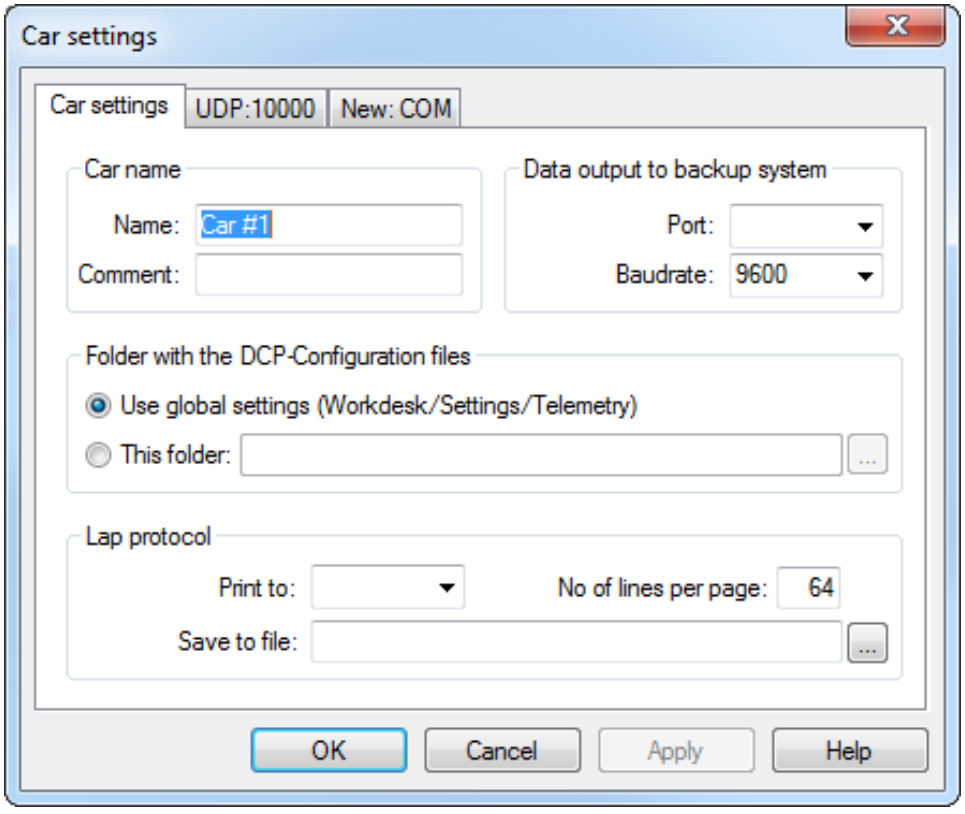


14.3 Setting up car in WDServer

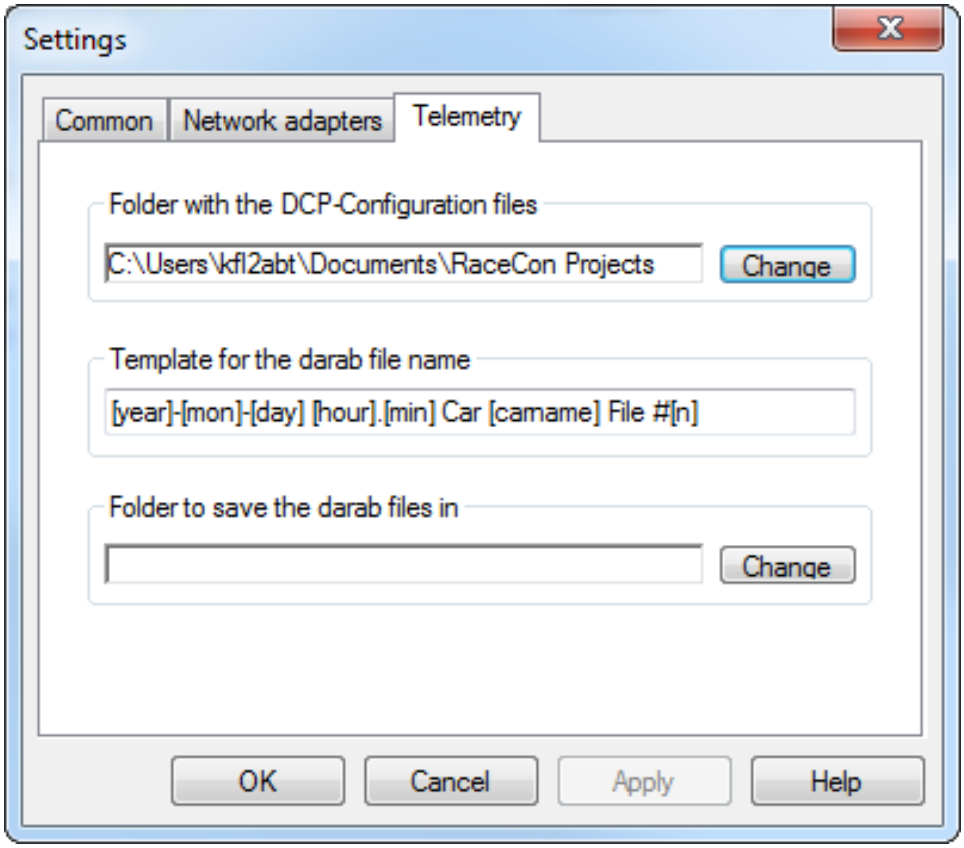
WDServer is a program used to capture data streaming from a transmitter and convert to WinDarab; WDServer also creates a log of the data received over telemetry.



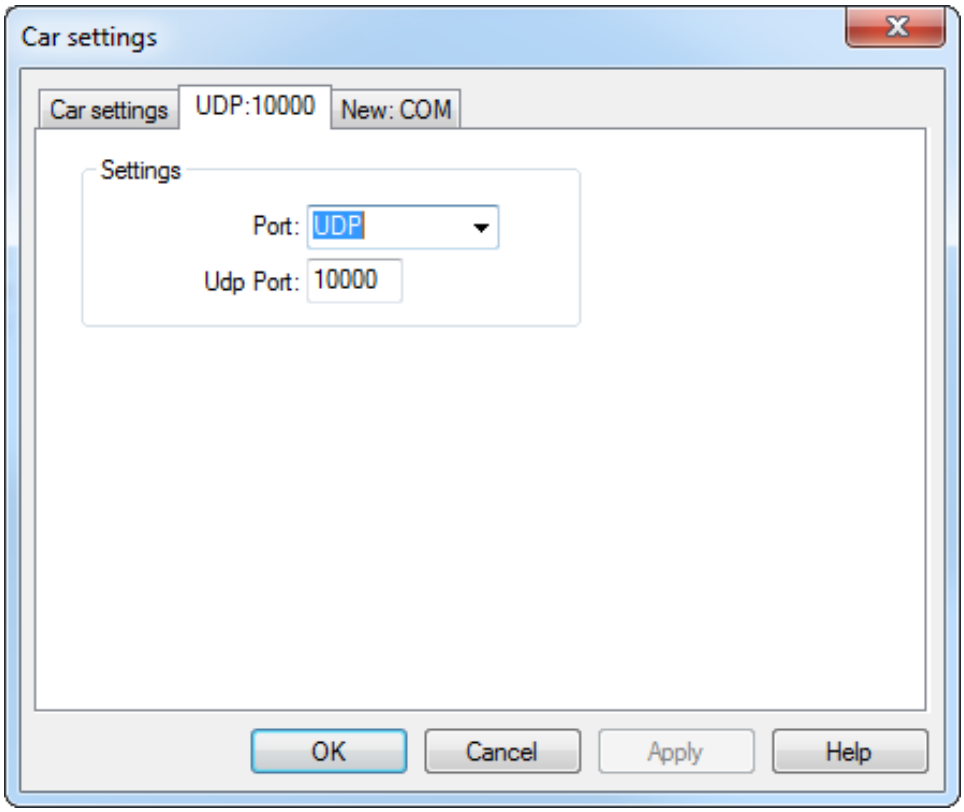
1. To set up a new car, select ‘Add Car’.
2. In the Car settings tab, enter a name for the new vehicle.
   This name will be used as a part of the file name for WDServer’s log of received telemetry data and will show up in WinDarab, when searching for the telemetry stream in the Network folder.



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.

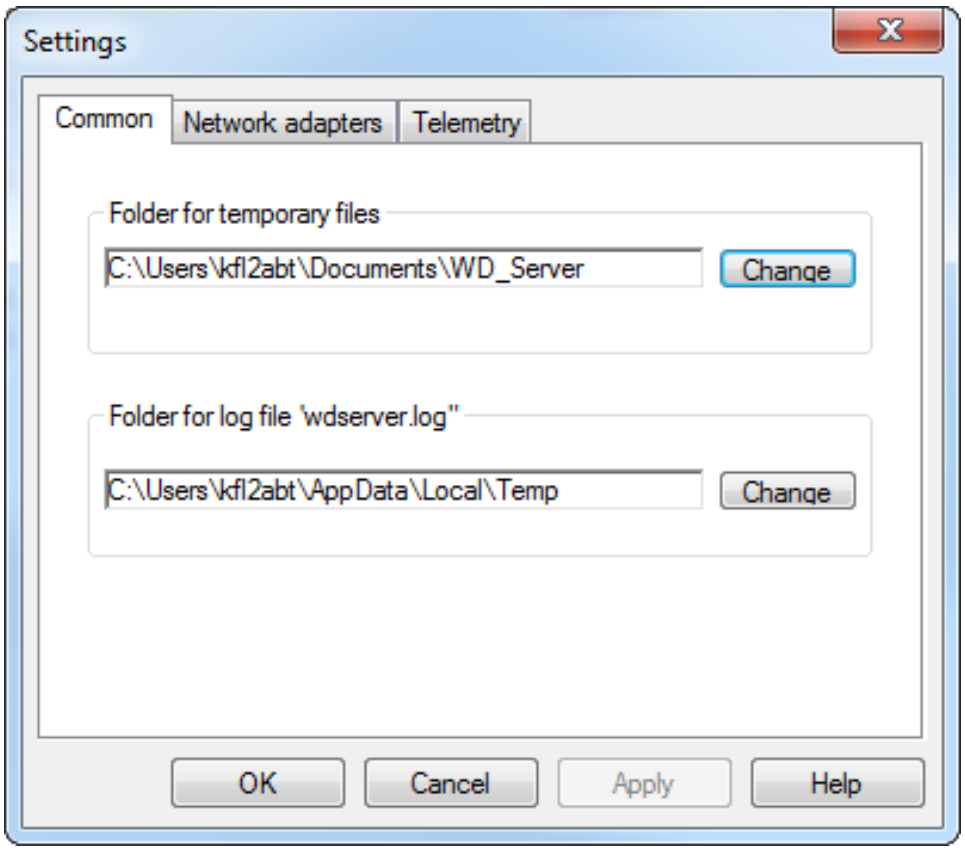


5. Under the 'UDP' tab, select the drop-down menu and type in "UDP".
6. For the UDP Port, type in the port number assigned to the device in RaceCon.
   Each vehicle being read by a single receiver device must have a unique port number.
   This information will be provided by Bosch upon delivery of the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



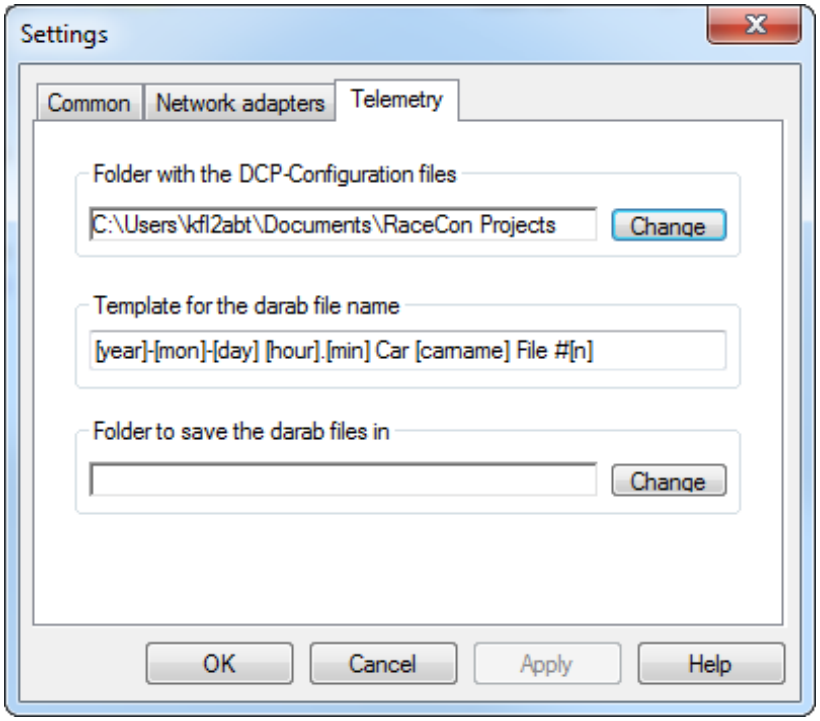
7. Click 'OK', to close the window.

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 WDServer.

10. 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.

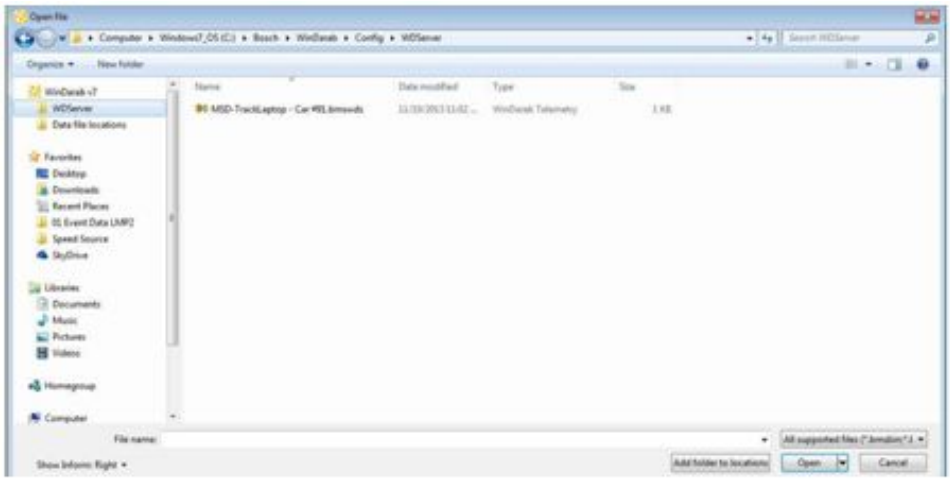


– To ensure proper communication between WDServer and the receiver, do not delete any old *.ini files from this folder path. As mentioned in section 5, RaceCon will generate a new *.ini file each time a project is synchronized; each new *.ini file instance has an incremented file name. Retaining all 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.

14.4 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.

15 Firmware

15.1 Firmware and configuration

MS 6 EVO holds 2 types of data:

Firmware: The software (PST program file) of the MS 6 EVO.

Configuration: The default parameters for controlling the output of the MS 6 EVO.

15.2 Firmware update

Firmware updates are performed via XCP. Following standards are supported:

- ASAM MCD-1 (XCP); Version 1.5.0; Date: 2017-11-30
- ASAM MCD-2 MC (ASAP2 / A2L); Version 1.7.1; Date: 2018-01-30

Additional files for firmware update (like ProF-Scripts for INCA) will be provided by BOSCH.

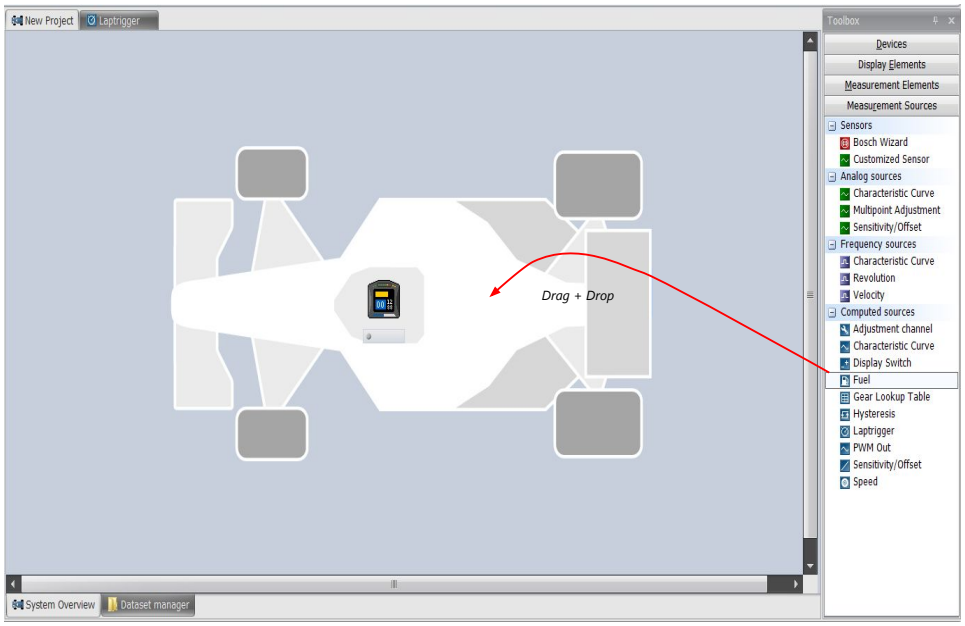
16 Cloning the Unit

Chapter left intentionally blank

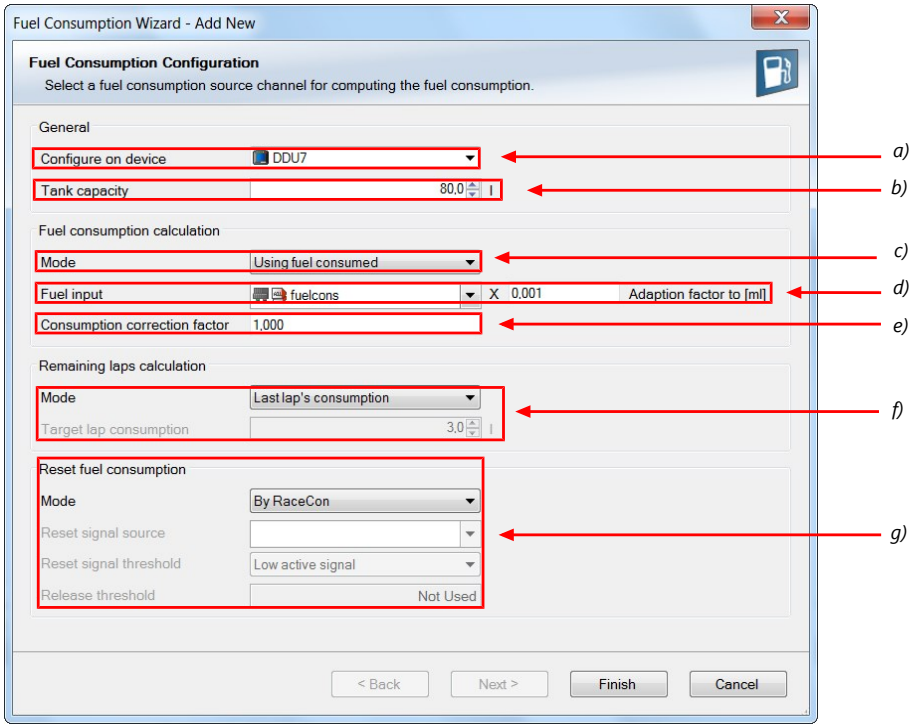
17 Fuel Consumption Calculation

17.1 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 MS 6 EVO!



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)

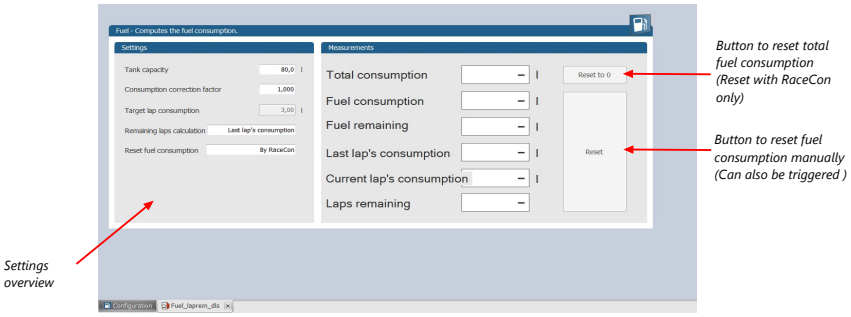
Press 'Finish' when done.

17.2 Fuel consumption diagnosis/counter reset

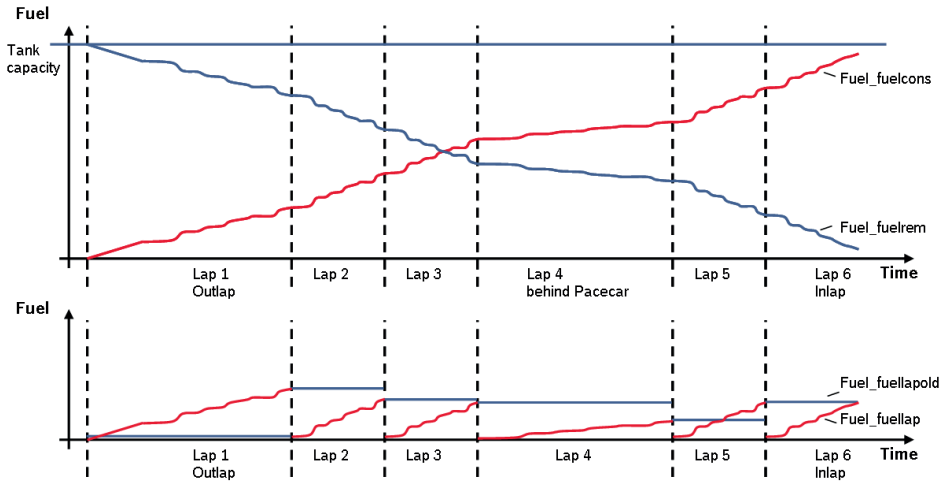
To display a fuel consumption diagnosis and to reset counters, use the diagnosis page in RaceCon.

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

18 RaceCon Shortcuts

The table shows important shortcuts simplify controlling the MS 6 EVO 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

19 Legal

19.1 Legal Restrictions

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.

19.2 Norms

Product Safety

EN IEC 62368-1:2020+A11:2020

Materials

REACH - Nr. 1907/2006

EMC

UNECE10:rev.6/AMD1:2020
KS-C9990:2017
ISO11452-2
ISO11452-4
ISO10605
ISO7637-2
ISO7367-3
ISO16750-2
US FCC: Title 47, Part 15 Subpart B
ICES-003

Testing

SAEJ1211

19.3 Open Source Software (OSS) declaration

19.3.1 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

19.3.2 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 PUR-

POSE 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.

19.3.3 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.

19.3.4 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/>.

/

19.3.5 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 direc-

tion 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 ne-

cessarily 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

19.3.6 Sensor Driver for BMI160 Sensor

BSD-3-Clause

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 copyright holder 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 HOLDER OR CONTRIBUT-

ORS 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.

19.3.7 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.

19.3.8 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.

19.3.9 Avalonia

The MIT License (MIT)

Copyright (c) .NET Foundation and Contributors All Rights Reserved

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.

19.4 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 monoxide (lead oxide)	1317-36-8
Lead	7439-92-1

20 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.

21 Pin Layout

The pin layout is available at Bosch Motorsport website on MS 6 EVO product page.

Most of MS 6 EVO functions to pin relations may be modified to project demands.

Please see details in the function description SWITCHMATRIX.

Bosch Motorsport tests check the defined connections of the pin layout.

Using a MS 6.1 EVO or MS 6.3 EVO version, ensure not using analogue inputs of the measurement package without enabled license.

For MS 6.1 EVO and MS 6.3 EVO, these hardware-options are only available if MS 6 EVO measurement package is in use.

Analogue Inputs

ECU Pin connector >A<	ECU Pin connector >K<	MS6.1 EVO MS6.3 EVO not available*)	MS6Cup EVO not available	I/O Type	hardware	pin related functions	ecu_name	rec.wire size AWG	MS6 function recommendation	function to pin coordination	related physical input measure channel
A032				analog input	pullup 3k01, 12bit		I_A_ANA_FIXPU[1]	24	engine temperature sensor	PIN_IN_UTMOT	utmot
A033				analog input	pullup 3k01, 12bit		I_A_ANA_FIXPU[2]	24	oil temperature sensor	PIN_IN_UTOIL	utoil
A034				analog input	pullup 3k01, 12bit		I_A_ANA_FIXPU[3]	24	intake air temperature sensor	PIN_IN_UTINT	utint
A035		not avl.	not avl.	analog input	pullup 3k01, 12bit		I_A_ANA_FIXPU[4]	24	fuel temperature sensor	PIN_IN_UTFUEL	utfuel
A079				analog input	no pullup, 12bit angle- or time related measurement		I_A_ANA[12]	24	rail pressure sensor	PIN_IN_UPRAIL	uprail
A080		not avl.	not avl.	analog input	no pullup, 12bit angle- or time related measurement		I_A_ANA[13]	24	rail pressure sensor, bank 2	PIN_IN_UPRAIL2	uprail2
A081				analog input	no pullup, 12bit		I_A_ANA[1]	24	fuel pressure sensor	PIN_IN_UPFUEL	upfuel
A082				analog input	switch.-pullup 3k01 12bit	CWPULLUP_A082	I_A_ANA_SWPU[13]	24	oil pressure sensor	PIN_IN_UPOIL	upoil
A058				analog input	no pullup, 12bit angle- or time related measurement		I_A_ANA[14]	24	pressure upstream throttle	PIN_IN_UP21	up21
A059		not avl.	not avl.	analog input	no pullup, 12bit, angle- or time related measurement, shared with digital and SENT inputs		I_A_ANA[15]	24	pressure upstream throttle, bank 2	PIN_IN_UP21_2	up21_2
A060				analog input	no pullup, 12bit angle- or time related measurement		I_A_ANA[16]	24	intake manifold pressure, mean value	PIN_IN_UP22M	up22m
A061		not avl.	not avl.	analog input	no pullup, 12bit, angle- or time related measurement, shared with digital and SENT inputs		I_A_ANA[11]	24	intake manifold pressure, mean value, bank 2	PIN_IN_UP22M_2	up22m_2

ECU Pin connector >A<	ECU Pin connector >K<	MS6.1 EVO MS6.3 EVO not available*)	MS6Cup EVO not available	I/O Type	hardware	pin related functions	ecu_name	rec.wire size AWG	MS6 function recommendation	function to pin coordination	related physical input measure channel
A056				analog input	no pullup, 12bit		I_A_APS1	24	APS potentiometer a	fixed function to pin coordination	uaps_a
A054				analog input	no pullup, 12bit		I_A_APS2	24	APS potentiometer b	fixed function to pin coordination	uaps_b
A041				analog input	no pullup, 12bit		I_A_UTH1	24	throttle potentiometer	fixed function to pin coordination	uthrottle
A053				analog input	no pullup, 12bit		I_A_UTH2	24	backup throttle poten-tiometer	fixed function to pin coordination	uthrottle_b
A036			not avl.	analog input	no pullup, 12bit		I_A_UTH3	24	throttle potentiometer. bank 2	fixed function to pin coordination	uthrottle2
A037			not avl.	analog input	no pullup, 12bit		I_A_UTH4	24	backup throttle poten-tiometer, bank 2	fixed function to pin coordination	uthrottle2_b
	K036			analog input	pullup 3k01, 12bit		I_A_ANA_FIXPU[5]	24	map switch	PIN_IN_UMAPSW	umapsw
	K031			analog input	switch.-pullup 3k01 12bit	CWPULLUP_K031	I_A_ANA_SWPU[1]	24	pitspeed switch	PIN_IN_UPITSPEEDSW	upitspeedsw
	K019	not avl.		analog input	switch.-pullup 3k01 12bit	CWPULLUP_K019	I_A_ANA_SWPU[2]	24	launch control switch	PIN_IN_ULAUNCHSW	ulaunchsw
	K015			analog input	switch.-pullup 3k01 12bit	CWPULLUP_K015	I_A_ANA_SWPU[3]	24	traction control switch	PIN_IN_UTCSW	utcsw
	K016	not avl.	not avl.	analog input	switch.-pullup 3k01 12bit	CWPULLUP_K016	I_A_ANA_SWPU[4]	24	reset chassis channels switch	PIN_IN_UCHRESSW	uchressw
	K017	not avl.	not avl.	analog input	switch.-pullup 3k01 12bit	CWPULLUP_K017	I_A_ANA_SWPU[5]	24	wet track switch	PIN_IN_UWETSW	uwetsw
A039				analog input	no pullup, 12bit		I_A_ANA[2]	24	gear poti	PIN_IN_UGEARP	ugearp
A055		not avl.		analog input	switch.-pullup 3k01 12bit	CWPULLUP_A055	I_A_ANA_SWPU[8]	24	reverse shift switch	PIN_IN_UREVSW	ushiftrevsw
A057		not avl.		analog input	switch.-pullup 3k01 12bit	CWPULLUP_A057	I_A_ANA_SWPU[9]	24	downshift switch	PIN_IN_USHIFTDNSW	ushiftdnsw
A076		not avl.		analog input	switch.-pullup 3k01 12bit	CWPULLUP_A076	I_A_ANA_SWPU[10]	24	up shift switch	PIN_IN_USHIFTUPSW	ushiftupsw
A077				analog input	switch.-pullup 3k01 12bit	CWPULLUP_A077	I_A_ANA_SWPU[11]	24	gearshift sensor	PIN_IN_UGS	ugs
A078		not avl.		analog input	switch.-pullup 3k01 12bit	CWPULLUP_A078	I_A_ANA_SWPU[12]	24	free measure channel A78		
A038		not avl.		analog input	no pullup, 12bit		I_A_ANA[5]	24	gearbox pneumatic pres-sure	PIN_IN_UPGEARAIR	upgearair
	K033	not avl.		analog input	no pullup, 12bit		I_A_ANA[4]	24	clutch pressure	PIN_IN_UPCLUTCH	upclutch
	K048	not avl.	not avl.	analog input	no pullup, 12bit		I_A_ANA[10]	24	free measure channel K48		

ECU Pin connector >A<	ECU Pin connector >K<	MS6.1 EVO MS6.3 EVO not available*)	MS6Cup EVO not available	I/O Type	hardware	pin related functions	ecu_name	rec.wire size AWG	MS6 function recommendation	function to pin coordination	related physical input measure channel
A040				analog input	no pullup, 12bit		I_A_ANA[6]	24	pressure brake rear	PIN_IN_UPBRAKE_R	upbrake_r
	K020			analog input	switch.-pullup 3k01 12bit	CWPULLUP_K020	I_A_ANA_SWPU[7]	24	pressure brake front	PIN_IN_UPBRAKE_F	upbrake_f
	K018	not avl.	not avl.	analog input	switch.-pullup 3k01 12bit	CWPULLUP_K018	I_A_ANA_SWPU[6]	24	damper sensor front/left	PIN_IN_UDAM_FL	udam_fl
	K032	not avl.	not avl.	analog input	no pullup, 12bit		I_A_ANA_[3]	24	damper sensor front/right	PIN_IN_UDAM_FR	udam_fr
	K034	not avl.	not avl.	analog input	no pullup, 12bit		I_A_ANA_[7]	24	damper sensor rear/left	PIN_IN_UDAM_RL	udam_rl
	K035	not avl.	not avl.	analog input	no pullup, 12bit		I_A_ANA_[8]	24	damper sensor rear/right	PIN_IN_UDAM_RR	udam_rr
	K050	not avl.	not avl.	analog input	no pullup, 12bit		I_A_ANA_[9]	24	steering angle sensor	PIN_IN_USTEER	usteer
	K066	not avl.	not avl.	analog input	no pullup, 12bit, angle- or time related measurement, shared with digital and SENT inputs		I_F_DIG_IN[5]	24	free measure channel K066		
	K067	not avl.	not avl.	analog input	no pullup, 12bit, angle- or time related measurement, shared with digital and SENT inputs		I_F_DIG_IN[6]	24	free measure channel K067		
	K083	not avl.	not avl.	analog input	no pullup, 12bit, angle- or time related measurement, shared with digital and SENT inputs		I_F_DIG_IN[3]	24	free measure channel K083		
	K084	not avl.	not avl.	analog input	no pullup, 12bit, angle- or time related measurement, shared with digital and SENT inputs		I_F_DIG_IN[4]	24	free measure channel K084		
	K049	not avl.	not avl.	analog input	no pullup, 12bit, angle- or time related measurement, shared with digital and SENT inputs		I_F_DIG_IN[7]	24	free measure channel K049		
A083		not avl.	not avl.	analog input	no pullup, 12bit, angle- or time related measurement, shared with digital and SENT inputs		I_F_DIG_IN[8]	24	free measure channel A083		

ECU Pin connector >A<	ECU Pin connector >K<	MS6.1 EVO MS6.3 EVO not available*)	MS6Cup EVO not available	I/O Type	hardware	pin related functions	ecu_name	rec.wire size AWG	MS6 function recommendation	function to pin coordination	related physical input measure channel
	K077			thermocouple 1+	k-type sensor		I_A_TC1A	24shield thermo	exhaust gas temperature sensor	fixed function to pin coordination	utexh
	K076			thermocouple 1-			I_A_TC1B				
	K079	not avl.	not avl.	thermocouple 2+	k-type sensor		I_A_TC2A	24shield thermo	exhaust gas temperature sensor, bank 2	fixed function to pin coordination	utexh2
	K078	not avl.	not avl.	thermocouple 2-			I_A_TC2B				

Digital Inputs

ECU Pin connector >A<	ECU Pin connector >K<	MS6.1 EVO MS6.3 EVO not available*)	MS6Cup EVO not available	I/O Type	hardware	pin related functions	ecu_name	rec.wire size AWG	MS6 function recommendation	function to pin coordination	related physical input measure channel
A047				crankshaft+ (Hall/Inductive)	switchable between halleffect- and inductive sensor	CWINTF_CRANK PIN_IN_CRANK CWINTF_CRANK_K CWINTF_CRANK_TH	I_P_CRANKA	24shield	engine speed	fixed function to pin coordination	nmot
A048				crankshaft - (inductive)			I_P_CRANKB	24shield			
A074				digital input	halleffect sensor only		I_P_CAM1	24shield	camshaft inlet	PIN_IN_CAM_IN	cam_pos_edges_001
A075				digital input	halleffect sensor only		I_P_CAM2	24shield	camshaft outlet	PIN_IN_CAM_OUT	cam_pos_edges_out_001
A049				digital input	switchable between halleffect- or DF11 sensors	CWINTF_A049	I_P_WHEEL1	24shield	camshaft inlet bank2 or wheelspeed front right	PIN_IN_CAM_IN2 or PIN_IN_FWEEL_FR	cam_pos_edges2_001 fwheel_fr
A050				digital input	switchable between halleffect- or DF11 sensors	CWINTF_A050	I_P_WHEEL2	24shield	camshaft outlet bank2 or wheelspeed front left	PIN_IN_CAM_OUT2 or PIN_IN_FWEEL_FL	cam_pos_edges_out2_001 fwheel_fl
A051				digital input	switchable between halleffect- or DF11 sensors	CWINTF_A051	I_P_WHEEL3	24shield	wheelspeed rear right	PIN_IN_FWHEEL_RR	fwheel_rr
A052				digital input	switchable between halleffect- or DF11 sensors	CWINTF_A052	I_P_WHEEL4	24shield	wheel speed rear left	PIN_IN_FWHEEL_RL	fwheel_rl
A059				digital input	switch.-pullup 14k7, shared with analogue and SENT inputs	CWPULLUP_A059	I_F_DIG_IN[1]	24shield	free digital channel A059		
A061				digital input	switch.-pullup 14k7, shared with analogue and SENT inputs	CWPULLUP_A061	I_F_DIG_IN[2]	24shield	free digital channel A061		

ECU Pin connector >A<	ECU Pin connector >K<	MS6.1 EVO MS6.3 EVO not available*)	MS6Cup EVO not available	I/O Type	hardware	pin related functions	ecu_name	rec.wire size AWG	MS6 function recommendation	function to pin coordination	related physical input measure channel
	K066			digital input	switch.-pullup 14k7, shared with analogue and SENT inputs	CWPULLUP_K066	I_F_DIG_IN[3]	24shield	free digital channel K066		
	K067			digital input	switch.-pullup 14k7, shared with analogue and SENT inputs	CWPULLUP_K067	I_F_DIG_IN[4]	24shield	free digital channel K067		
	K083			digital input	switch.-pullup 14k7, shared with analogue and SENT inputs	CWPULLUP_K083	I_F_DIG_IN[5]	24shield	free digital channel K083		
	K084			digital input	switch.-pullup 14k7, shared with analogue and SENT inputs	CWPULLUP_K084	I_F_DIG_IN[6]	24shield	free digital channel K084		
	K049			digital input	switch.-pullup 14k7, shared with analogue and SENT inputs	CWPULLUP_K049	I_F_DIG_IN[7]	24shield	free digital channel K049		
A083				digital input	switch.-pullup 14k7, shared with analogue and SENT inputs	CWPULLUP_A083	I_F_DIG_IN[8]	24shield	free digital channel A083		
	K045			digital input	switchable between halleffect- and inductive sensor	CWINTF_K045 CWINTF_K045_K CWINTF_K045_TH	I_P_CAM3	24shield	turbo speed	PIN_IN_FTURBO	fturbo
	K046			digital input	switchable between halleffect- and inductive sensor	CWINTF_K046 CWINTF_K046_K CWINTF_K046_TH	I_P_CAM4	24shield	turbo speed bank2	PIN_IN_FTURBO2	fturbo2
	K062			ground supply	if inductive sensos are connected to K045 or K046		G_R_GNDCAM	24shield	ground for turbo speed and -2		
	K054			digital input			I_S_ENGINE_ON	20	Engine On/Off switch		b_engon(_in)
	K047			digital input	fixed pullup to 5volts		I_S_LAPTRIG	24	laptrigger	fixed function to pin coordination	lapctr
A013				knock sensor in-put			I_A_KNOCK_IN[1]	24shield	knock sensor 1, bank1	KCSENCYL	ikcraw_n_..

ECU Pin connector >A<	ECU Pin connector >K<	MS6.1 EVO MS6.3 EVO not available*)	MS6Cup EVO not available	I/O Type	hardware	pin related functions	ecu_name	rec.wire size AWG	MS6 function recommendation	function to pin coordination	related physical input measure channel
A014				knock sensor input			I_A_KNOCK_IN[2]	24shield	knock sensor 2, bank1	KCSENCYL	ikcraw_n_..
A015			not avl.	knock sensor input			I_A_KNOCK_IN[3]	24shield	knock sensor 1, bank2	KCSENCYL	ikcraw_n_..
A016			not avl.	knock sensor input			I_A_KNOCK_IN[4]	24shield	knock sensor 2, bank2	KCSENCYL	ikcraw_n_..
A017				knock sensor ground			G_R_GNDKNOCK	24shield			
	K085			Lambda_IA	LSU4.9 probe only		I_A_LS1IA	24	Lambda	fixed function to pin coordination	lambda
	K086			Lambda_IP			I_A_LS1IP	24			
	K087			Lambda_UN			I_A_LS1UN	24			
	K088			Lambda_VM			I_A_LS1VM	24			
	K068		not avl.	Lambda_IA	LSU4.9 probe only		I_A_LS2IA	24	Lambda bank2	fixed function to pin coordination	lambda2
	K069		not avl.	Lambda_IP			I_A_LS2IP	24			
	K070		not avl.	Lambda_UN			I_A_LS2UN	24			
	K071		not avl.	Lambda_VM			I_A_LS2VM	24			

Ignition- & Injection Outputs

ECU Pin connector >A<	ECU Pin connector >K<	MS6.1 EVO MS6.3 EVO not available*)	MS6Cup EVO not available	I/O Type	hardware	pin related functions	ecu_name	rec.wire size AWG	MS6 function recommendation	function to pin coordination	related physical input measure channel
A026				ignition driver	output related to mechanical cylinder number; use of coil integrated power stages only	CWIGNDRV_MODE IGNDRV_CURRENT	O_P_IGNOUT[1]	24	Ignition cyl.1	CYLNUMBER CYLANGLE	ign_out_n_001
A027							O_P_IGNOUT[2]	24	Ignition cyl.2		ign_out_n_002
A028							O_P_IGNOUT[3]	24	Ignition cyl.3		ign_out_n_003
A029							O_P_IGNOUT[4]	24	Ignition cyl.4		ign_out_n_004
A030			not avl.				O_P_IGNOUT[5]	24	Ignition cyl.5		ign_out_n_005
A031			not avl.				O_P_IGNOUT[6]	24	Ignition cyl.6		ign_out_n_006
A068			not avl.				O_P_IGNOUT[7]	24	Ignition cyl.7		ign_out_n_007
A069			not avl.				O_P_IGNOUT[8]	24	Ignition cyl.8		ign_out_n_008
A070			not avl.				O_P_IGNOUT[9]	24	ignition cyl.9		ign_out_n_009
A071			not avl.				O_P_IGNOUT[10]	24	ignition cyl.10		ign_out_n_010
A072			not avl.				O_P_IGNOUT[11]	24	ignition cyl.11		ign_out_n_011
A073			not avl.				O_P_IGNOUT[12]	24	ignition cyl.12		ign_out_n_012

ECU Pin connector >A<	ECU Pin connector >K<	MS6.1 EVO MS6.3 EVO not available*)	MS6Cup EVO not available	I/O Type	hardware	pin related functions	ecu_name	rec.wire size AWG	MS6 function recommendation	function to pin coordination	related physical input measure channel
A098				injector output	output related to mechanical cylinder number; low pressure high impedance injector types		O_P_LSOUT_INJECTION[1]	24twist	Injection cyl.1	CYLNUMBER CYLANGLE or (PIN_OUT_LPINJ_A098 ... PIN_OUT_LPINJ_A084)	tinj_n_001
A100							O_P_LSOUT_INJECTION[2]	24twist	Injection cyl.2		tinj_n_002
A101							O_P_LSOUT_INJECTION[3]	24twist	Injection cyl.3		tinj_n_003
A096							O_P_LSOUT_INJECTION[4]	24twist	Injection cyl.4		tinj_n_004
A099			not avl.				O_P_LSOUT_INJECTION[5]	24twist	Injection cyl.5		tinj_n_005
A103			not avl.				O_P_LSOUT_INJECTION[6]	24twist	Injection cyl.6		tinj_n_006
A042			not avl.				O_P_LSOUT_INJECTION[7]	24twist	Injection cyl.7		tinj_n_007
A105			not avl.				O_P_LSOUT_INJECTION[8]	24twist	Injection cyl.8		tinj_n_008
A018			not avl.				O_P_LSOUT_INJECTION[9]	24twist	Injection cyl.9		tinj_n_009
A020			not avl.				O_P_LSOUT_INJECTION[10]	24twist	Injection cyl.10		tinj_n_010
A063			not avl.				O_P_LSOUT_INJECTION[11]	24twist	Injection cyl.11		tinj_n_011
A084			not avl.				O_P_LSOUT_INJECTION[12]	24twist	Injection cyl.12		tinj_n_012
A043				INJVH1	high pressure magnetic injectors		O_P_INJVH1	20twist	Injection cyl.A	PIN_OUT_HPINJ11A_A043_A064	tinj_n_(cyl.A)
A064				INJVL11			O_P_INJVL11	20twist			
A002			not avl.	INJVH3			O_P_INJVH3	20twist	Injection cyl.B	PIN_OUT_HPINJ32B_A002_A023	tinj_n_(cyl.B)
A023			not avl.	INJVL32			O_P_INJVL32	20twist			
A003				INJVH2			O_P_INJVH2	20twist	Injection cyl.C	PIN_OUT_HPINJ21C_A003_A024	tinj_n_(cyl.C)
A024				INJVL21			O_P_INJVL21	20twist			
A046			not avl.	INJVH4			O_P_INJVH4	20twist	Injection cyl.D	PIN_OUT_HPINJ42D_A046_A067	tinj_n_(cyl.D)
A067			not avl.	INJVL42			O_P_INJVL42	20twist			
A044			not avl.	INJVH1			O_P_INJVH1	20twist	Injection cyl.E	PIN_OUT_HPINJ12E_A044_A065	tinj_n_(cyl.E)
A065			not avl.	INJVL12			O_P_INJVL12	20twist			
A001				INJVH3			O_P_INJVH3	20twist	Injection cyl.F	PIN_OUT_HPINJ31F_A001_A022	tinj_n_(cyl.F)
A022				INJVL31			O_P_INJVL31	20twist			
A004			not avl.	INJVH2			O_P_INJVH2	20twist	Injection cyl.G	PIN_OUT_HPINJ22G_A004_A025	tinj_n_(cyl.G)
A025			not avl.	INJVL22			O_P_INJVL22	20twist			
A045				INJVH4			O_P_INJVH4	20twist	Injection cyl.H	PIN_OUT_HPINJ41H_A045_A066	tinj_n_(cyl.H)
A066				INJVL41			O_P_INJVL41	20twist			

Outputs

ECU Pin connector >A<	ECU Pin connector >K<	MS6.1 EVO MS6.3 EVO not available*)	MS6Cup EVO not available	I/O Type	hardware	pin related functions	ecu_name	rec.wire size AWG	MS6 function recommendation	function to pin coordination	related physical input measure channel
A095				lowside switch 4amps pwm			O_T_LSOUT_4A2[1]	24twist	camshaft inlet control	fixed pin to output control coordination	cam_pwm
A021			not avl.	lowside switch 4amps pwm			O_T_LSOUT_4A2[2]	24twist	camshaft inlet bank2 control	fixed pin to output control coordination	cam_pwm2
A102				lowside switch 3amps pwm			O_T_LSOUT_3A2[1]	24twist	camshaft outlet control	fixed pin to output control coordination	cam_pwm_out
A094			not avl.	lowside switch 3amps pwm			O_T_LSOUT_3A2[2]	24twist	camshaft outlet bank2 con-trol	fixed pin to output control coordination	cam_pwm_out2
A019				lowside switch 3amps pwm			O_T_LSOUT_3A2[4]	24twist		PIN_OUT_A019	
A104			not avl.	lowside switch 3amps pwm			O_T_LSOUT_3A2[3]	24twist		PIN_OUT_A104	
A097				lowside sw. 2,2amps pwm			O_T_LSOUT_2A2[1]	24twist	Wastegate 1inc	PIN_OUT_A097	wgc_inc_pwm
A093			not avl.	lowside sw. 2,2amps pwm			O_T_LSOUT_2A2[2]	24twist	Wastegate 2inc	PIN_OUT_A093	wgc_inc_pwm2
	K039		not avl.	lowside sw. 2,2amps pwm			O_T_LSOUT_2A2[5]	24twist		PIN_OUT_K039	
	K056			lowside sw. 2,2amps pwm			O_T_LSOUT_2A2[7]	24twist	air conditioning compressor	PIN_OUT_K056	comp_pwm
	K038			lowside sw. 2,2amps pwm			O_T_LSOUT_2A2[3]	24twist	gearshift actuator upshift	PIN_OUT_K038	shiftup_pwm
	K040		not avl.	lowside sw. 2,2amps pwm			O_T_LSOUT_2A2[6]	24twist		PIN_OUT_K040	
	K055			lowside sw. 2,2amps pwm			O_T_LSOUT_2A2[4]	24twist	gearshift actuator downshift	PIN_OUT_K055	shiftdn_pwm
	K074			lowside sw. 2,2amps pwm			O_T_LSOUT_2A2[8]	24twist		PIN_OUT_K074	
	K089			lowside switch 1amp pwm			O_T_LSOUT_1A[1]	24twist	fuel pump relay	PIN_OUT_K089	fpump_pwm
	K073		not avl.	lowside switch 1amp pwm			O_T_LSOUT_1A[2]	24twist		PIN_OUT_K073	
	K057			lowside switch 1amp pwm / reset < 3,5V			O_S_RELAY	24twist	control main relay	fixed pin to output control coordination	b_mainrelay
	K072			lowside switch 1amp pwm / reset < 3,5V			O_S_STARTER	24twist	Kl.50 / starter control	fixed pin to output control coordination	b_starter
	K022			lambda heater 4amp pwm			O_T_LSOUT_LSH[1]	24twist	heater lambda	fixed pin to output control coordination	lsuh_out

ECU Pin connector >A<	ECU Pin connector >K<	MS6.1 EVO MS6.3 EVO not available*)	MS6Cup EVO not available	I/O Type	hardware	pin related functions	ecu_name	rec.wire size AWG	MS6 function recommendation	function to pin coordination	related physical input measure channel
	K023		not avl.	lambda heater 4amp pwm			O_T_LSOUT_LSH[2]	24twist	heater lambda2	fixed pin to output control coordination	lsuh_out2

H-Bridges & Metering Unit

ECU Pin connector >A<	ECU Pin connector >K<	MS6.1 EVO MS6.3 EVO not available*)	MS6Cup EVO not available	I/O Type	hardware	pin related functions	ecu_name	rec.wire size AWG	MS6 function recommendation	function to pin coordination	related physical input measure channel
A089				H-Bridge 1 pos.	8,5 amps H-Bridge	CWHB1_EN	O_T_HB1_OUTA	24twist	electrical throttle 1	fixed pin to output control coordination	etc_pwm
A090				H-Bridge 1 neg.			O_T_HB1_OUTB				
A091				H-Bridge 2 pos.	8,5 amps H-Bridge	CWHB2_EN	O_T_HB2_OUTA	24twist	electrical throttle 2	fixed pin to output control coordination	etc_pwm2
A092				H-Bridge 2 neg.			O_T_HB2_OUTB				
K090				H-Bridge 3 pos.	8,5 amps H-Bridge	CWHB3_EN	O_T_HB3_OUTA	24twist		fixed pin to output control coordination	
K091				H-Bridge 3 neg.			O_T_HB3_OUTB				
A085				FCVH1			O_P_FCVH1	24twist	high press. pump MSV valve 1	fixed pin to output control coordination	msv_dlvy_angle
A086				FCVL1			O_P_FCVL1				
A087			not avl.	FCVH2			O_P_FCVH2	24twist	high press. pump MSV valve 2	fixed pin to output control coordination	msv_dlvy_angle2
A088			not avl.	FCVL2			O_P_FCVL2				

Network

ECU Pin connector >A<	ECU Pin connector >K<	MS6.1 EVO MS6.3 EVO not available*)	MS6Cup EVO not available	I/O Type	hardware	pin related functions	ecu_name	rec.wire size AWG	MS6 function recommendation	function to pin coordination	related physical input measure channel
	K029			CAN1_H	switchable CAN 120 Ohm resistor recommended for Motronic, Powerbox and ABS control functions	CWCAN1_TERM	B_D_CAN1_H	CAN	CAN1		E_can1
	K012			CAN1_L			B_D_CAN1_L				
	K028			CAN2_H	switchable CAN 120 Ohm resistor ~use for external ECU / gearbox control functions	CWCAN2_TERM	B_D_CAN2_H	CAN	CAN2		E_can1
	K011			CAN2_L			B_D_CAN2_L				
	K027			CAN3_H	switchable CAN 120 Ohm resistor ~use for measurement functions	CWCAN3_TERM	B_D_CAN3_H	CAN	CAN3		E_can1
	K010			CAN3_L			B_D_CAN3_L				
	K052			RS232_RX	used for telemetry		B_D_RS232_RX	24twist	RS232		

ECU Pin connector >A<	ECU Pin connector >K<	MS6.1 EVO MS6.3 EVO not available*)	MS6Cup EVO not available	I/O Type	hardware	pin related functions	ecu_name	rec.wire size AWG	MS6 function recommendation	function to pin coordination	related physical input measure channel
	K053			RS232_TX			B_D_RS232_TX				
	K044			ETH1RX+	ecu communication		B_D_ETH1RX+	CAT7	Ethernet 1		
	K043			ETH1RX-			B_D_ETH1RX-				
	K042			ETH1TX+			B_D_ETH1TX+				
	K041			ETH1TX-			B_D_ETH1TX-				
	K061			ETH2RX+	extended communica-tion to PBx90, DDU7, DDU8 or C60		B_D_ETH2RX+	CAT7	Ethernet 2		
	K060			ETH2RX-			B_D_ETH2RX-				
	K059			ETH2TX+			B_D_ETH2TX+				
	K058			ETH2TX-			B_D_ETH2TX-				
	K025			USB_DP	use for additional data stick		B_D_USB_DP	USB	USB		
	K024			USB_DN			B_D_USB_DN				
	K007			USB_GND			G_G_USB_GND				
	K008			USB_VBUS			O_V_USB_VBUS				
	K014			TIMESYNC	timeline to Ethernet extension modules		B_F_TIMESYNC	24	data time syncronising line		
	K066			not used							
	K067			not used							
	K083			not used							
	K084			not used							
	K051			LIN	LIN communication	CWLINMODE	B_D_LIN	24	LIN-Bus		
	K030			TN digital output	configurable rpm-output	TNSIG_PULSENUM TNSIG_PWM	O_F_DIGOUT[1]	24	rpm-signal	PIN_OUT_K030	
	K013			TN digital output	to check engine syn-cronisation,		O_F_DIGOUT[2]	24	flywheel-signal	PIN_OUT_K013	
	K037			TN digital output			O_F_DIGOUT[3]	24	triggerwheel-signal	PIN_OUT_K037	
A059				SENT1	shared with analog and digital inputs	CWPULLUP_A059	I_F_DIG_IN[1]	24	SENT-BUS		
A061				SENT2	shared with analog and digital inputs	CWPULLUP_A061	I_F_DIG_IN[2]	24	SENT-BUS		
	K066			SENT3	shared with analog and digital inputs	CWPULLUP_K066	I_F_DIG_IN[3]	24	SENT-BUS		

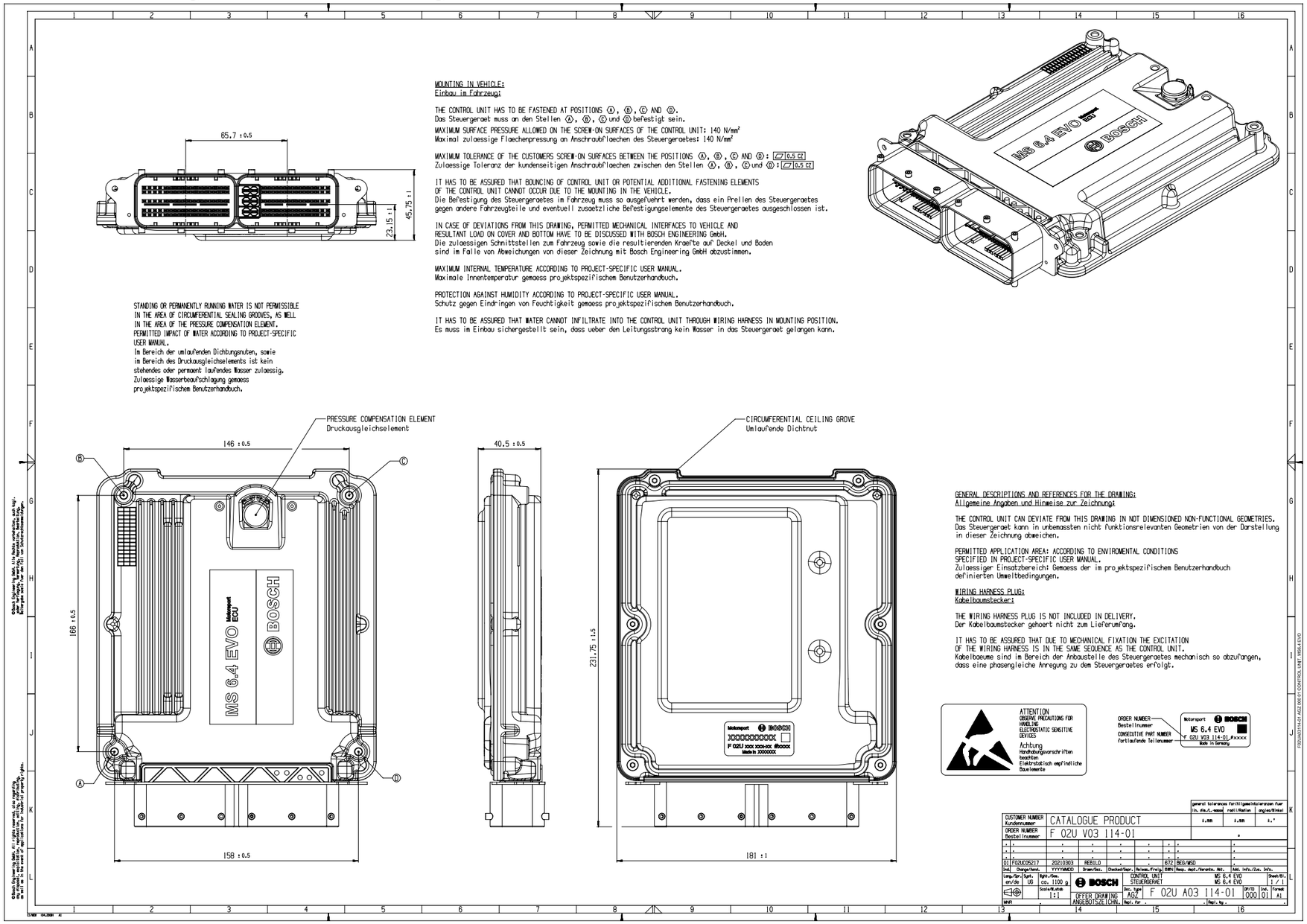
ECU Pin connector >A<	ECU Pin connector >K<	MS6.1 EVO MS6.3 EVO not available*)	MS6Cup EVO not available	I/O Type	hardware	pin related functions	ecu_name	rec.wire size AWG	MS6 function recommendation	function to pin coordination	related physical input measure channel
	K067			SENT4	shared with analog and digital inputs	CWPULLUP_K067	I_F_DIG_IN[4]	24	SENT-BUS		
	K083			SENT5	shared with analog and digital inputs	CWPULLUP_K083	I_F_DIG_IN[5]	24	SENT-BUS		
	K084			SENT6	shared with analog and digital inputs	CWPULLUP_K084	I_F_DIG_IN[6]	24	SENT-BUS		
	K049			SENT7	shared with analog and digital inputs	CWPULLUP_K049	I_F_DIG_IN[7]	24	SENT-BUS		
A083				SENT8	shared with analog and digital inputs	CWPULLUP_A083	I_F_DIG_IN[8]	24	SENT-BUS		

Power Supplies

ECU Pin connector >A<	ECU Pin connector >K<	MS6.1 EVO MS6.3 EVO not available*)	MS6Cup EVO not available	I/O Type	hardware	pin related functions	ecu_name	rec.wire size AWG	MS6 function recommendation	function to pin coordination	related physical input measure channel
	K003			battery plus			V_V_DYNPWR	14	dynamic power supply		
	K005			battery plus			V_V_DYNPWR	14	dynamic power supply		
	K006			battery plus			V_V_DYNPWR	14	dynamic power supply		
	K075			battery plus			V_V_VBAT	20	digital power supply		
	K001			battery minus			G_G_GND	14			
	K002			battery minus			G_G_GND	14			
	K004			battery minus			G_G_GND	14			
A009				sensor supply 5V/50mamp	recommended supply for: aps_a, etc		O_V_5VSNS4	24	ETC sensor supply 1		
A011				sensor supply 5V/50mamp	recommended supply for: aps_b, etc2		O_V_5VSNS5	24	ETC sensor supply 2		
	K065			sensor supply 5V/150mamp		CW5VOUT3_EN	O_V_5VSNS3	24	5 V sensor supply 4		
A007				sensor supply 5V/50mamp			O_V_5VSNS7	24	5 V sensor supply 5		
A005				sensor supply 5V/150mamp		CW5VOUT1_EN	O_V_5VSNS1	24	5 V sensor supply 1		
	K064			sensor supply 5V/150mamp		CW5VOUT2_EN	O_V_5VSNS2	24	5 V sensor supply 2		

ECU Pin connector >A<	ECU Pin connector >K<	MS6.1 EVO MS6.3 EVO not available*)	MS6Cup EVO not available	I/O Type	hardware	pin related functions	ecu_name	rec.wire size AWG	MS6 function recommendation	function to pin coordination	related physical input measure channel
	K063			sensor supply 5V/50mamp			O_V_5VSNS6	24	5 V sensor supply 3		
	K080			sensor ground 1			M_R_GNDSNS1	20	ground sensor supply		
	K081			sensor ground 2			M_R_GNDSNS6	20	ground sensor supply		
	K082			sensor ground 3			M_R_GNDSNS7	20	ground sensor supply		
A006				sensor ground 4			G_R_GNDSNS2	20	ground sensor supply		
A008				sensor ground 5			G_R_GNDSNS3	20	ground sensor supply		
A010				sensor ground 6	recommended ground for: aps_a, etc		G_R_GNDSNS4	20	ground sensor supply		
A012				sensor ground 7	recommended ground for: aps_b, etc2		G_R_GNDSNS5	20	ground sensor supply		
A062				screen ground			PCB_FUSE_2A	24	sensor screens		
	K021			screen ground			PCB_FUSE_2A	24	sensor screens		
	K026			screen ground			PCB_FUSE_2A	24	Ethernet and LIN screens		
	K049			not used							
	K009			screen ground			PCB_FUSE_2A	24	USB screen		
A083				not used							

22 Offer Drawing and Basic Wiring

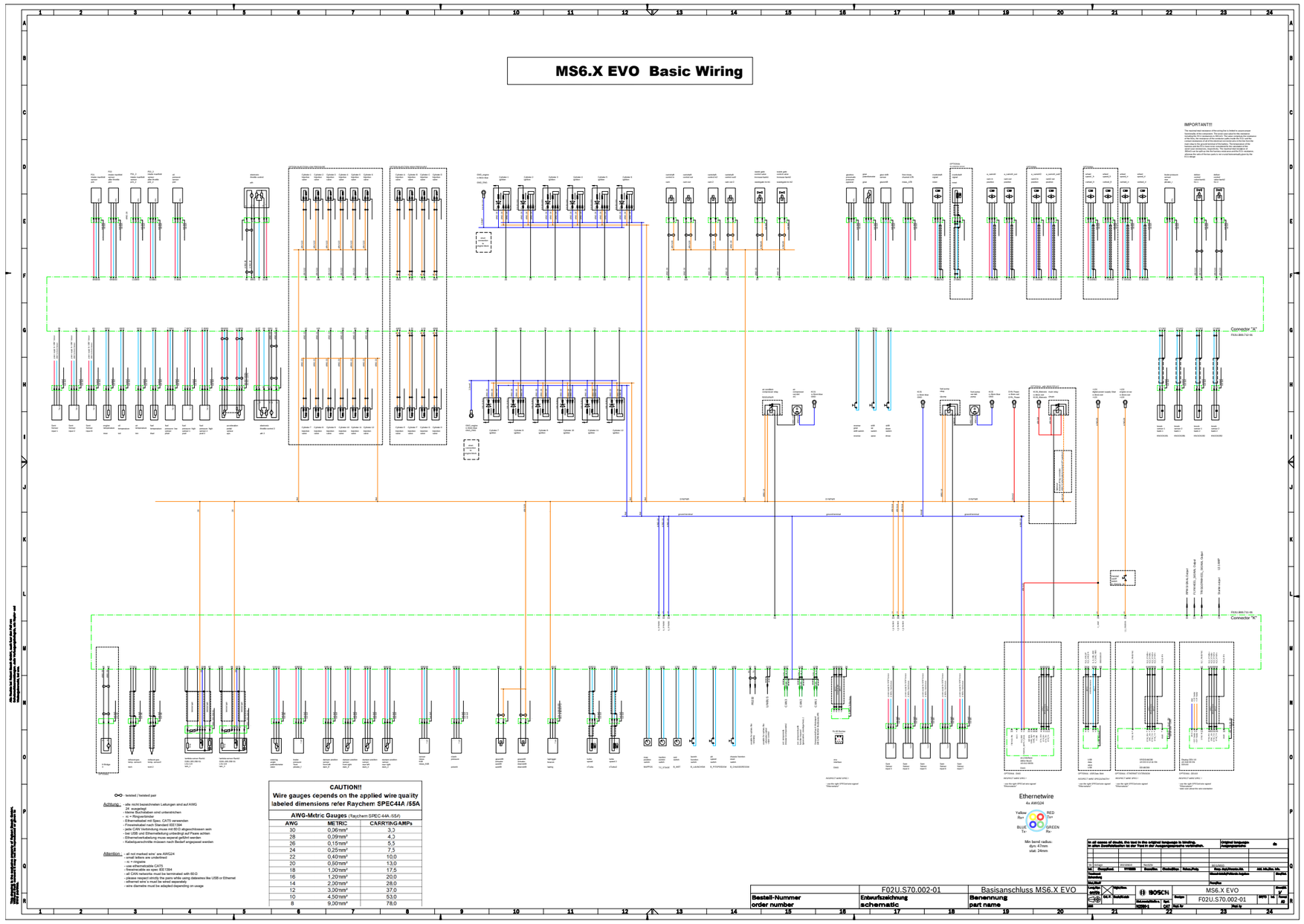


The wiring diagram is available at Bosch Motorsport website on the MS 6 EVO product page.

NOTICE

The wiring diagram shows a principle of wiring and connection options.

ECU pin relation may change to customer data application and program layout. Sensor-, actuator- and power supplies may also change to the request of the project.



Bosch Engineering GmbH
Motorsport
Robert-Bosch-Allee 1
74232 Abstatt

www.bosch-motorsport.com