CAN Protocols Driver Builder

1 Introduction

CAN Protocols Driver Builder is a utility included in Race Studio 3, intended for developing a driver capable to read CAN data streams coming from ECUs or other CAN connected devices. The produced driver can be used on any device supported by Race Studio 3 software.

As a general recommendation, drivers designed to **handle** ECU data streams are assigned to CAN1, while other devices (i.e. ABS, Gearshift, TPMS modules) are managed by CAN2 of AiM devices featuring this extra data bus.

AiM Race Studio 3 offers three different ways to build a CUSTOM CAN protocol driver:

- Importing an XC1 file (AiM proprietary file).
- Importing a DBC file (a standard file used by manufacturers).
- Manually building a driver, based on datasheets or reverse engineering notes.

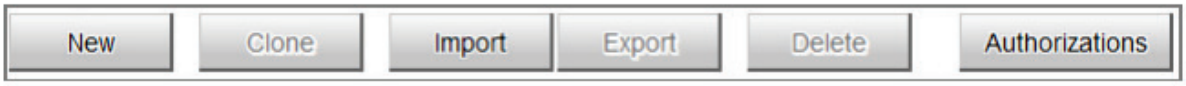
Note: AiM does NOT supply any kind of readable or editable files relevant to CAN protocols. Every development is released on RS3 official database and will not be disclosed in any way.

2
CAN Protocols Driver Builder - Main page

To access the CAN Protocols Driver Builder, press the button CAN Protocols (highlighted in the picture) from the top menu of Race Studio 3.

The Main page will open up with a blank list and the main commands placed in the upper bar and the collection filters in the left side of the view.

The main commands functions are briefly reported here below.



New - creates a new driver
Clone - generates a new project copying an existing driver.
Import - loads a driver from an XC1 or DBC file
Export - saves a driver into an XC1 file
Delete - clears a driver project
Authorizations - protect drivers from unwanted access

In the following chapters the main functions are described more in detail.

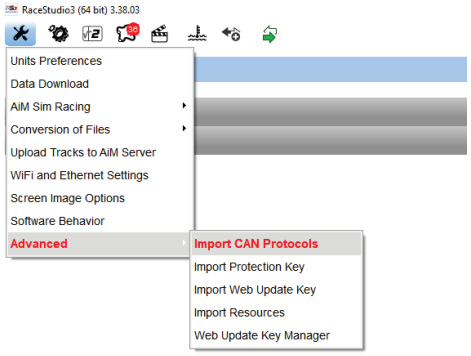
2.1

Import – XC1 files

An XC1 file is an AiM proprietary format, containing info relevant to the CAN protocol and its integration with AiM devices. This file can be exchanged between Race Studio 3 users to get a custom protocol ready to be used, for example between the developer and his customers.

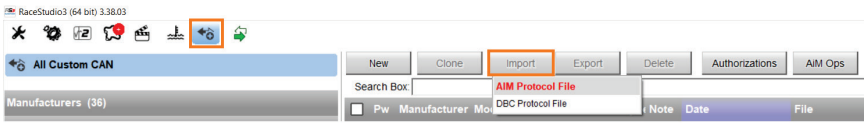
Once received the XC1 file, save on the PC and **Import** it into Race Studio 3, through the following steps:

Open Race Studio 3
Select Preferences (first icon)
Advanced
Import CAN protocols
Select the file to import

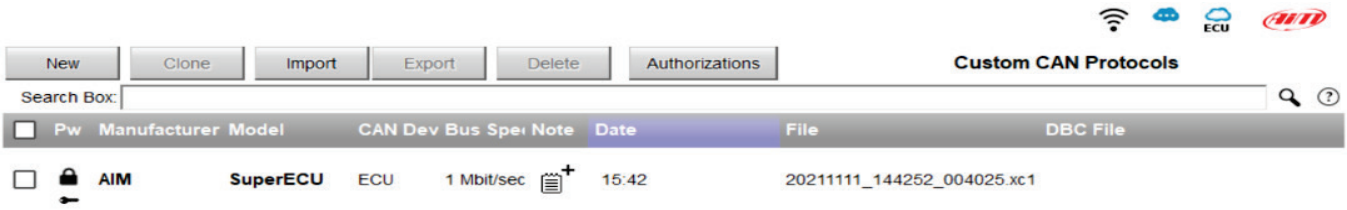


In alternative:

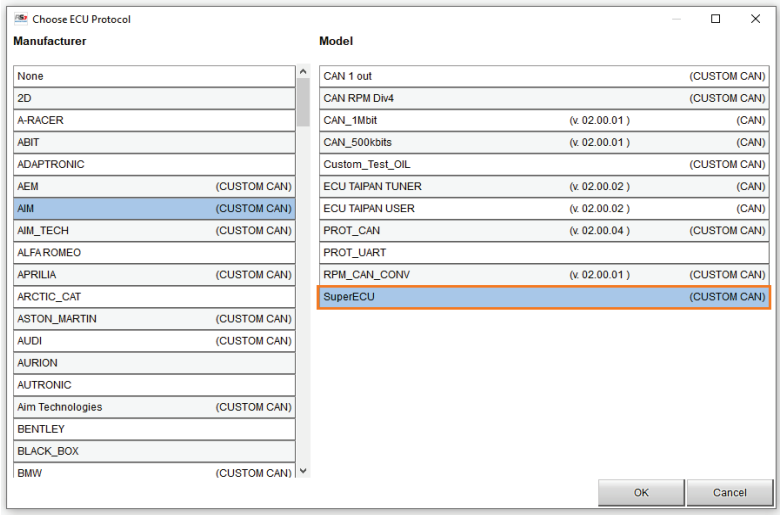
- Open Race Studio 3
- Select CAN Protocols
- Import
- **AiM Protocol File**
- Select the file to import



Once imported, the CAN driver is loaded in the CAN Protocols list



and in the ECU Stream Tab of Race Studio 3 Configuration.



No further editing is required, this is ready to be used.

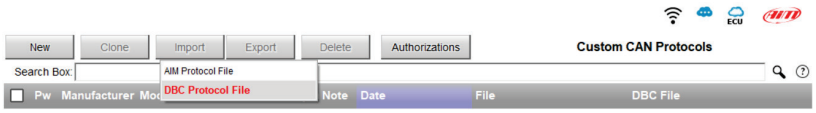
2.2
Import – DBC files

The DBC file containing CAN Bus information, is usually provided by the vehicle, or the ECU, manufacturer, moreover if these are aftermarket or racing products. This file contains most of the info, but requires a final editing by the user, in order to assign AiM functions and measure units, so to have the highest integration with AiM devices.

Starting from Race Studio 3 vers 3.38.03, the DBC Import function has been introduced.

The importing procedure is the following:

- Open Race Studio 3
- Select CAN Protocols
- **Import**
- **DBC Protocol File**
- Select the file to import



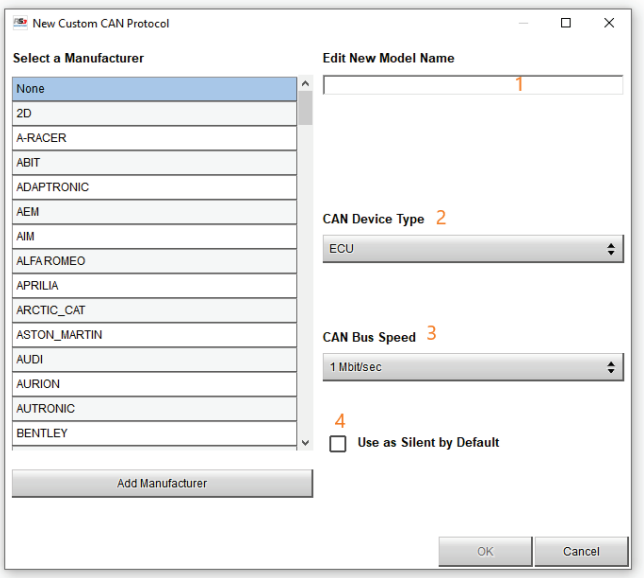
Once imported, the following window appears, where the driver properties must be set:

1. **Manufacturer** and **Model** names: these have to be carefully set, in order to be able to identify the proper driver among the hundreds available in the database.
   In case the manufacturer is not in the list, this can be added with Add Manufacturer.

2. **CAN device type**: ECU or Other CAN Device are the two options to list the protocol driver only in CAN1 under the ECU Streams or both in CAN1 and CAN2

3. **CAN Bus Speed**: the value must match with the vehicle CANbus rate. Wrong values will bring to CAN faults on the vehicle side.

4. **Use as Silent by Default**: if the transmitting node (i.e. ECU) doesn't expect any ack, or if acks are interfering with existing bus nodes, select this option. Silent Mode can also be changed when setting up the ECU Stream configuration.



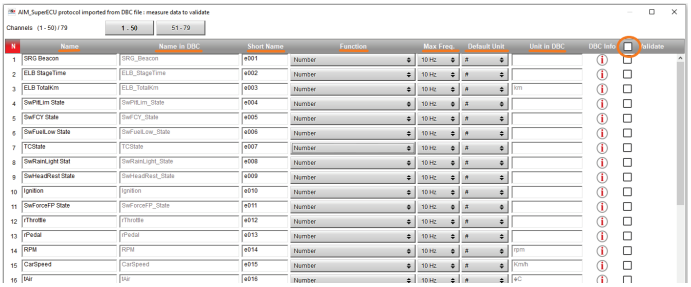
Once properties set up completed, DBC imported CAN protocols are shown in the ECU driver list with a specific icon which can be RED or GREEN. If RED, the driver still needs to be completed in some of the fields, press on the icon to open the complete list of channels requiring an editing.

Per every channel the fields are:

Name: proposed for AiM devices management

Name in DBC: the original name (it can be truncated if longer than permitted length)

Short Name: the label for LCD dashes with a maximum of 4 characters



Function: the most important field, it allows to link the CAN channel with AiM device functions, so to correctly handle the information both for live data and logged data.

Max Frequency: the maximum frequency allowed to save specific channels. This is set to limit the load on the data logger, in particular for channels not requiring high rates. End users will then be able to save data also at lower frequencies in the configuration.

Default Unit: the function already narrows the type of measure units, this field is to select the most appropriate.

Unit in DBC: since DBC files are based on strings, there isn't an automatic selection for measure units. This field reports what was set in the original DBC file for the user reference.

The flag helps taking note of validated channels. When all flags are ticked, the icon in the protocol list turns into GREEN.

2.3
New – Build a new CAN driver

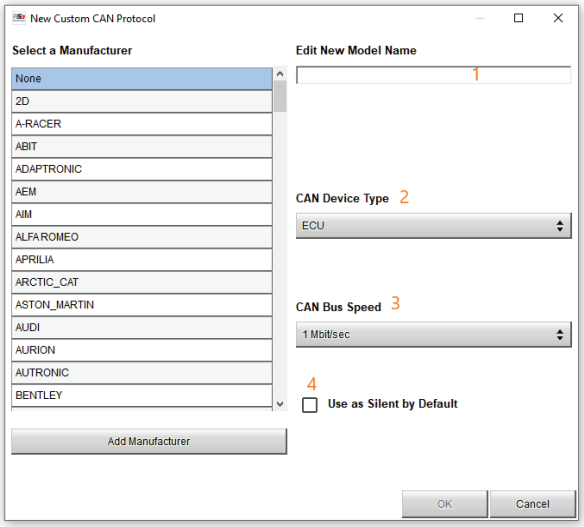
In the main page of the CAN Protocols ECU Driver Builder, press **New** for creating a new custom CAN protocol driver and set the main driver properties:

1. **Manufacturer** and **Model** names: these have to be carefully set, in order to be able to identify the proper driver among the hundreds available in the database.
   In case the manufacturer is not in the list, this can be added with Add Manufacturer.

2. **CAN device type**: ECU or Other CAN Device are the two options to list the protocol driver only in CAN1 under the ECU Streams or both in CAN1 and CAN2

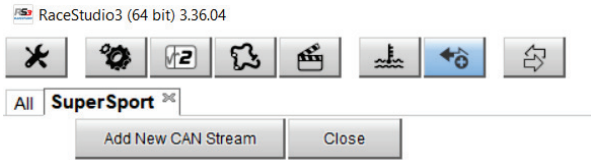
3. **CAN Bus Speed**: the value must match with the vehicle CANbus rate. Wrong values will bring to CAN faults on the vehicle side.

4. **Use as Silent by Default**: if the transmitting node (i.e. ECU) doesn't expect any ack, or if acks are interfering with existing bus nodes, select this option. Silent Mode can also be changed when setting up the ECU Stream configuration.



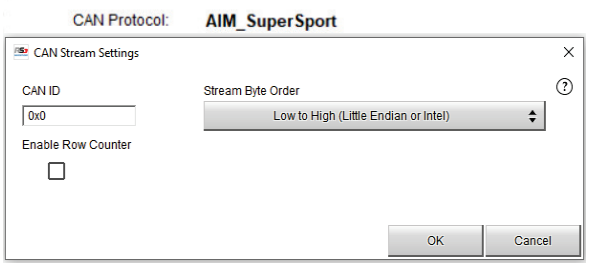
After having set the proper fields and confirmed with the OK button, the CAN driver page appears as a white board and two options are shown:

- Add New CAN Stream
- Close



Add New CAN Stream enables the addition of a new CAN frame with the following fields:

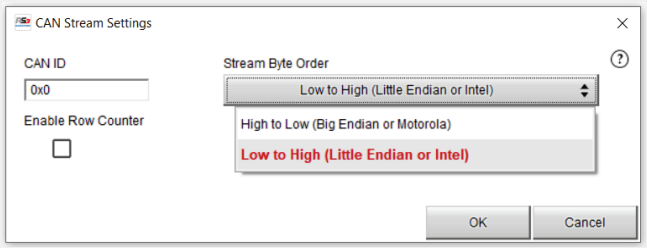
- **CAN ID**: the arbitration ID indicated in the CAN protocol documentation. This can be either 11 or 29bit, the input format is hexadecimal.

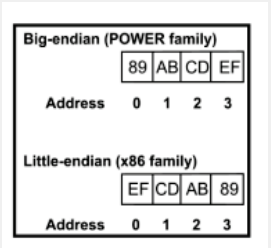


- **Stream Byte Order**: toggles between the Little (Intel) or Big (Motorola) endianness, depending upon the ECU or CAN module processor. This is a parameter that must be reported in the CAN protocol documentation.

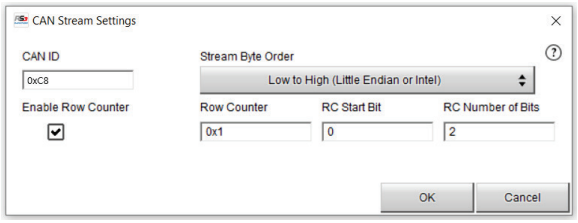
Tip:

Here an example of the two byte orders. Big Endian has a byte order where the Most Significant Byte is stored first, oppositely to Little Endian, where the Least Significant Byte is stored first.



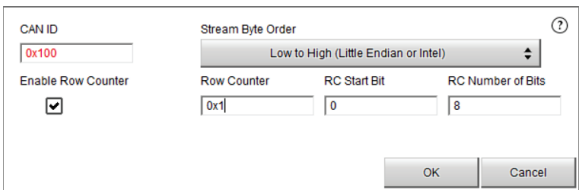


Enable Row Counter: this is used to manage a multiplexed stream, a particular CAN protocol where a single arbitration ID is carrying multiple frames, and each frame is determined by the value of an index, here called Row Counter.

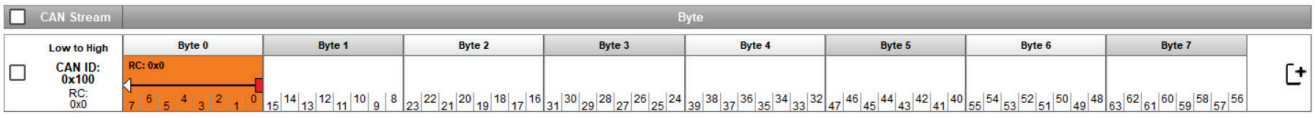


Start Bit, **Number of bits** (length) and **Counter Value** of every single row, have to be defined.

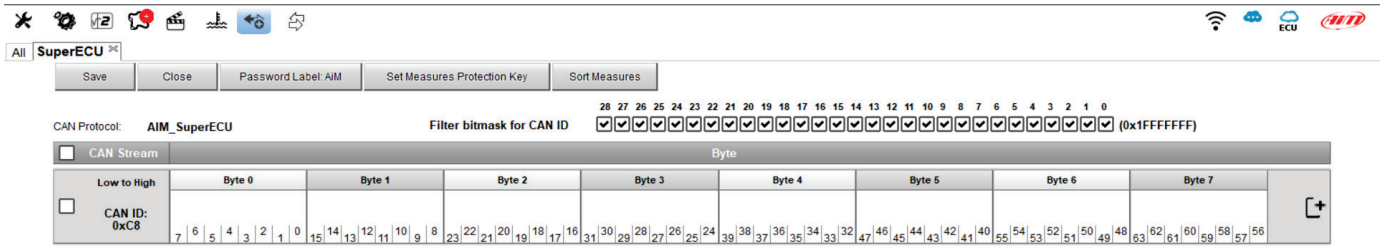
i.e. If the index is located on the first byte, set:
RC Start Bit = 0
RC Number of Bits = 8



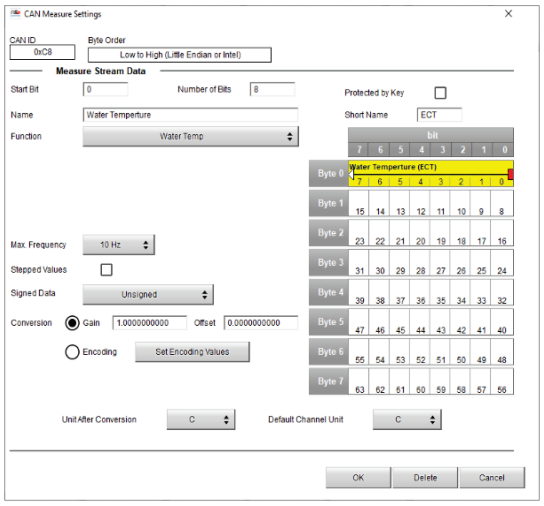
The result will be an orange field that marks the changing index on the first byte and the next seven bytes available for the CAN payload.

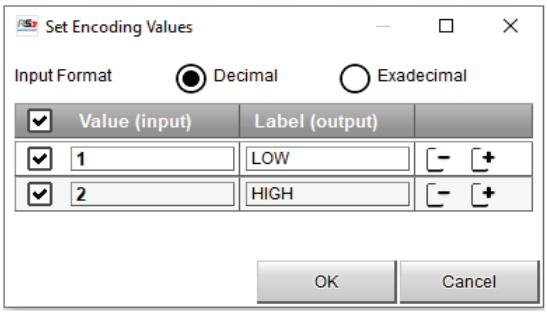


After having set all parameters, press OK and the first frame of the data stream is ready to be filled with channels and the following definitions:



Start Bit: for the channel positioning
Number of Bits: the channel bit length
Protected by Key: a protection that hides the channel to users without the software key code.
Name: to be used for AiM devices management
Short Name: label used on AiM LCD displays
Function: links the CAN channel with the AiM device functions, so to correctly handle the information
Max Frequency: sets the higher limit of the channels saving rate (not the sampling rate)
Stepped Value: deactivates interpolation in the Analysis software, for stepped measures (i.e gear)
Data Format: handles the sign set up
Conversion: defines a Gain and an Offset (Output = Input x Gain + Offset) or an Encoding. In this case, a conversion table is to be defined per every value of the channel (see second picture).
Unit After Conversion: this matches with the measure unit described in the CAN protocol definition
Default Channel Unit: RS3 Configuration default measure unit

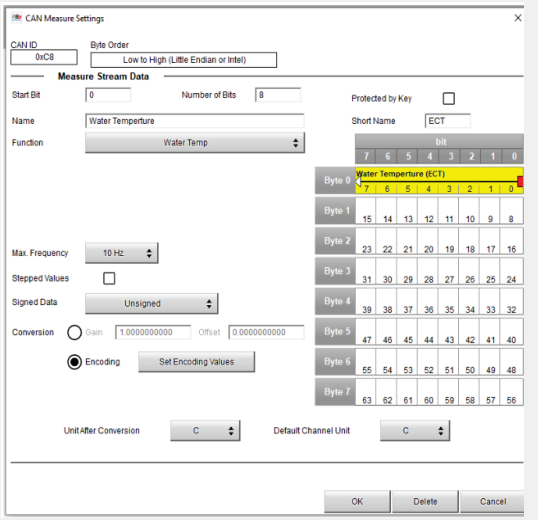




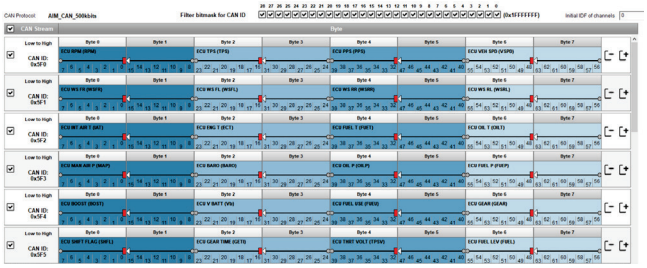
Tip:

The window CAN Measure Settings shows on the right the entire CAN ID payload and – in yellow – the corresponding measure being set.

To quickly set the Start Bit and the Number of Bits of each measure, it is possible to drag the center of the arrow, or pull and push the endpoints of it, to place the channel in the appropriate bits.



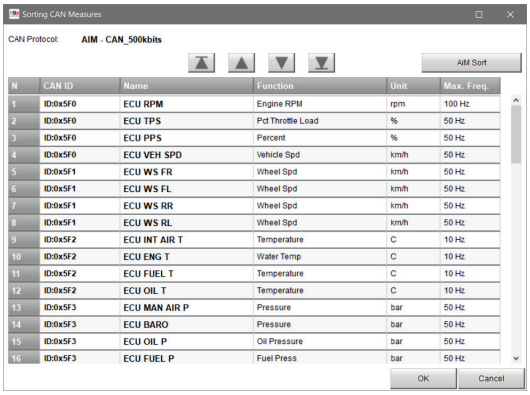
Once the driver has been completed all, the frames will appear as shown in the picture here.



+ and **–** are used to add or delete frames.
The **flag on the left**, enables/disables the stream.
The **29 flags on top**, are used to mask CAN IDs, carrying additional info (i.e. source, priority, etc.).
Initial ID of channels sets the Base ID of protocols working with offset IDs.

The button menu also features the **Sort Measures**.

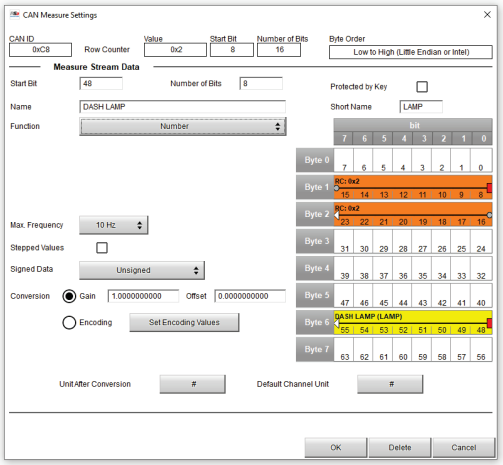
Through this, a custom sorting of the channels can be set, dragging channels up and down, or using the arrow keys to scroll up/down or top/bottom of the list.
Automatic operations can also be achieved pressing on the fields CAN ID, Name Function, Unit or Max Frequency.
There is also an option to automatically sort, following the AiM rules.



N	CAN ID	Name	Function	Unit	Max. Freq.
1	ID:0x5F0	ECU RPM	Engine RPM	rpm	100 Hz
2	ID:0x5F0	ECU TPS	Pct Throttle Load	%	50 Hz
3	ID:0x5F0	ECU PPS	Percent	%	50 Hz
4	ID:0x5F0	ECU VEH SPD	Vehicle Spd	km/h	50 Hz
5	ID:0x5F1	ECU WS FR	Wheel Spd	km/h	50 Hz
6	ID:0x5F1	ECU WS FL	Wheel Spd	km/h	50 Hz
7	ID:0x5F1	ECU WS RR	Wheel Spd	km/h	50 Hz
8	ID:0x5F1	ECU WS RL	Wheel Spd	km/h	50 Hz
9	ID:0x5F2	ECU INT AIR T	Temperature	C	10 Hz
10	ID:0x5F2	ECU ENG T	Water Temp	C	10 Hz
11	ID:0x5F2	ECU FUEL T	Temperature	C	10 Hz
12	ID:0x5F2	ECU OIL T	Temperature	C	10 Hz
13	ID:0x5F3	ECU MAN AIR P	Pressure	bar	50 Hz
14	ID:0x5F3	ECU BARO	Pressure	bar	50 Hz
15	ID:0x5F3	ECU OIL P	Oil Pressure	bar	50 Hz
16	ID:0x5F3	ECU FUEL P	Fuel Press	bar	50 Hz

2.3.1
How to set the Bit Fields

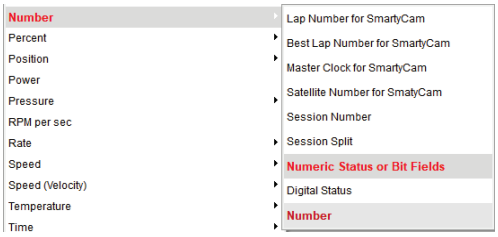
Among the available Functions, the **Number - Bit Field** needs a further explanation.



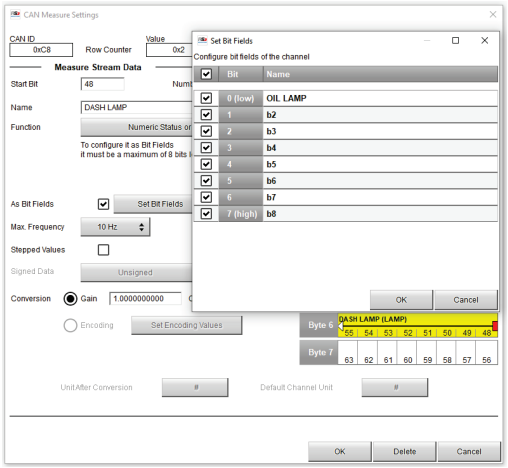
If a CAN stream includes a sequence of status bits, indicating if parameters are ON/OFF, True/False, High/Low, Fault/noFault, all these bits can be managed together with the **Bit Field** function. The constraint is that they can only be grouped together in groups of 8 bits (1 Byte).

Select the position of the bits within the CAN frame, like in the Dash Lamp example here shown.

Select the Function **Number - Numeric Status or Bit Fields**



Flag the box **As Bit Fields** and strike the **Set Bit Fields** button. The table **Configure bit fields of the channel** shows the bits to be monitored; the label aside can be customized to have a quick reference of the specific alarm, warning, status that it is turning on.



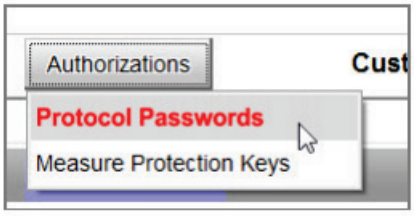
2.4
Authorizations – Password and Protection Key

Race Studio 3 allows to protect CAN protocol drivers at different levels by mean of a **Protocol Password** and a **Measure Protection Key**

Protocol Password: protects the opening and editing of a driver. This is generally applied not to disclose sensible CAN details, property of their respective owners.
This also means that an XC1 file can be protected with a Protocol Password and shared with other users without the risk of having it opened, because this can only be imported and used to log data.

Measure Protection Key: protects some specific channels that have been flagged. These locked channels remain hidden and encrypted to other users, while they can be read in Live Measures, downloaded and analyzed by those users owning the specific Measure Protection Key file.
Downloaded data always contain also the protected channel, but these can be disclosed only by those software where the Key has been previously imported.

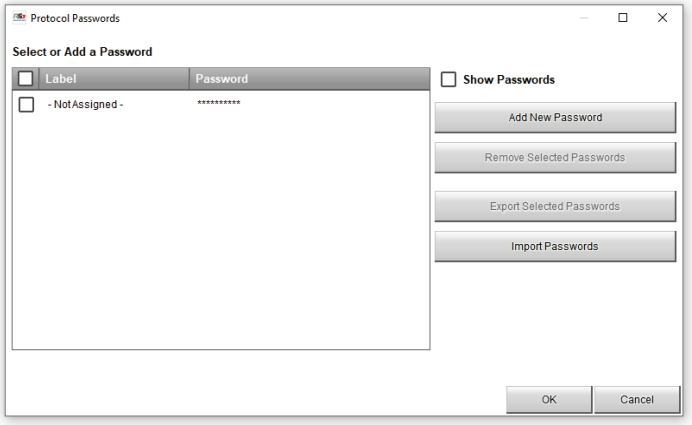
Both the Protocol Password and the Measure Protection Keys are accessible from different point of the ECU Driver Builder section.
The first access is through the main page of CAN Protocols, under the Authorizations button. Here the aim is to manage the different passwords and keys that a power user has to deal with, when more are assigned to different projects, without opening every single project.



The other access is when a specific protocol is opened, the **Set Protocol Password** and **Set Measures Protection Key** are displayed on the top button menu.

2.4.1 Protocol Password

Selecting **Protocol Password**, the dedicated window opens up. This differs if the access is from the management page (top image) or from the specific protocol (bottom image).



Add new password creates a new entry

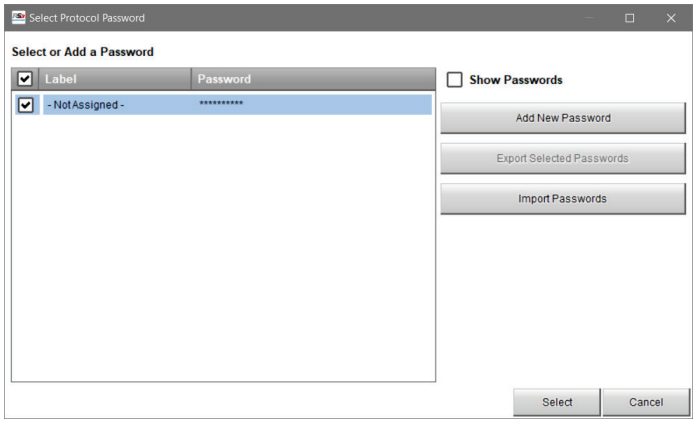
Remove Selected Passwords deletes one or more entries. This option is only available when opening from the managing page.

Export Selected passwords exports one or more passwords into a file to be shared or moved onto another laptop.

Import Passwords updates one or more passwords sourced by an AiM passwords file.

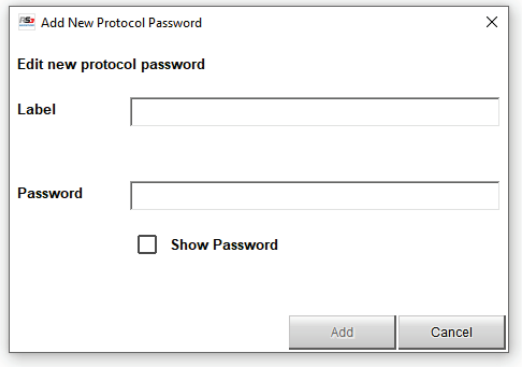
OK saves the changes in the management page.

Select assigns the flagged password to the opened protocol.



Tip:
Double-clicking on the password selects or de-selects it.
Ticking the flag box allows multiple selections, in particular for the export operations.

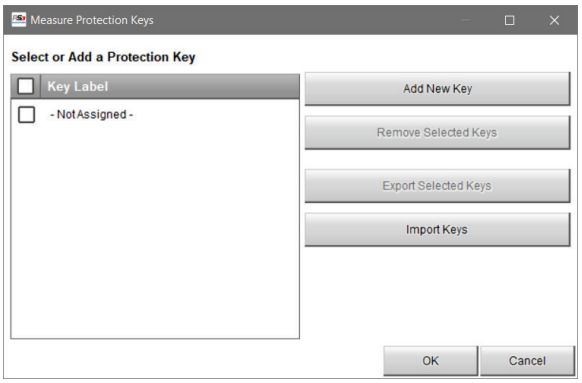
Add New Password



Set a **label** for the password, this will be displayed as a reference once the driver is opened.
Set the protection **password**, this won't be displayed in the protocol pages, and even in this window it is hidden and can be displayed only ticking Show Password.

2.4.2
Measure Protection Key

Selecting **Measure Protection Key**, the dedicated window opens up. This differs if the access is from the management page (top image) or from the specific protocol (bottom image).



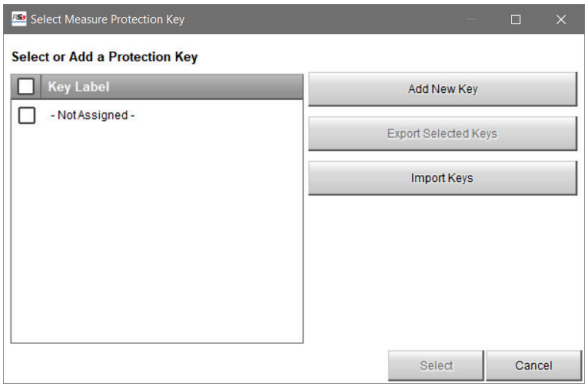
Add New Key creates a new entry

Remove Selected Keys deletes one or more entries. This option is only available when opening from the managing page.

Export Selected Keys exports one or more keys into a file to be shared or moved onto another laptop.

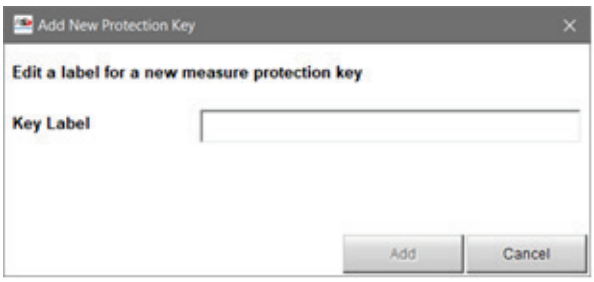
Import Keys updates one or more keys sourced by an AiM protection key file.

OK saves the changes in the management page.

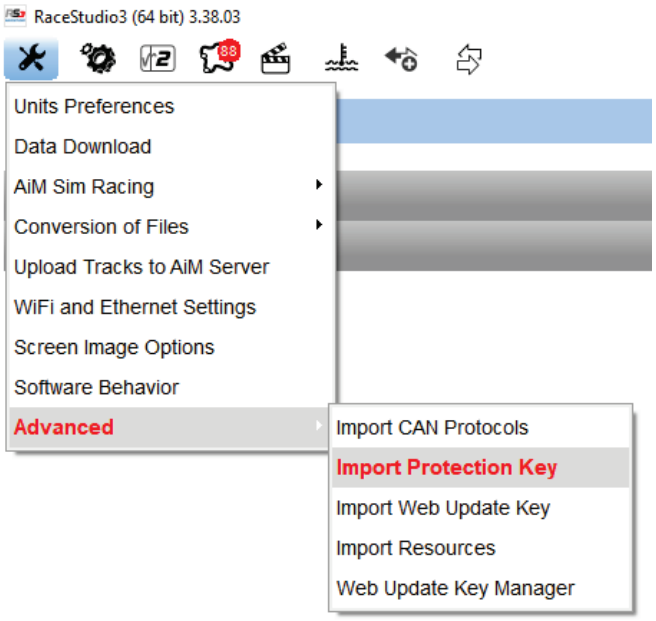


Select assigns the flagged key to the opened protocol.

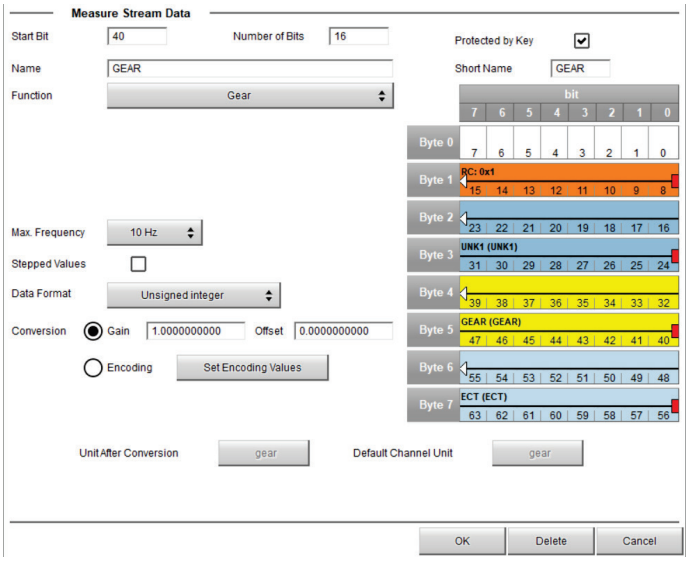
Add New Key
Choose a **label** for the Protection Key, this will be displayed as a reference once the driver is opened.
Differently from the Protection Password, there is no Protection Key code that can be typed in, because in this case the protection is guaranteed by the software generated file, so this file has to be present into the laptop in use, there is no way to manually type the code and unlock hidden channels.



To unlock measures protected by key it is necessary to import the key file previously exported by the generating software.
To import it click **"Advanced"** and **"Import Protection Key"** in the same drop-down menu.



When building the protocol, some of the channels can be hidden to other users, both in real time reading and in analysis. To do this, tick the flag **Protected by Key** for every channel to be protected.



In the final resume of the protocol, protected channels shall be underlined like in the example here reported for the Gear channel.