

# User Manual

## CANinterpreter

### Version history

Version	Changes	Date	Editor
V1.0	First version	2013/07/03	ged
V1.0.3	Minor bugfixes	2013/09/20	ged
V1.1	Additional PlugIns added	2014/02/28	ged
V1.2	CAN message format changed	2016/04/01	ri
V 2.6	Added description for multi interfaces	2020/04/28	ri

## Disclaimer

Even if software from emotas embedded communication GmbH is produced on high reliability levels and tested on different environments there is no proof on absolute reliability. Therefore emotas embedded communication GmbH cannot give any liability on even software or documentation and cannot guarantee their functionality in use cases of our customers. Especially description and technical data are not granted properties of our products. As a consequence emotas embedded communication GmbH will not be responsible for any damages and losses in consequence of using our products.

emotas embedded communication GmbH may apply changes on products and documentation to improve reliability, stability and serve technical advantages. These changes may not be announced.

emotas embedded communication GmbH holds all rights on their products and documentation. Propagation to third parties even from parts of products or documentation may be permitted by emotas embedded communication GmbH. Copies of products or documentation may be established only on backup purposes. It is in the responsibility of our customers not to pass these copies to third parties. We appreciate very much all considerations of our customers on errors or other aspects of improvement.

## Copyright

© 2020 emotas embedded communication GmbH

Fritz-Haber-Str. 9

D-06217 Merseburg

Germany

Tel. +49 3461/79416-0

Fax. +49 3461/79416-10

[service@emotas.de](mailto:service@emotas.de)

<http://www.emotas.de>

## Table of Contents

1	Introduction.....	4
2	Installation.....	4
2.1	Windows.....	4
2.2	Linux.....	4
3	First steps.....	5
4	CAN Multiinterface handling.....	6
4.1	General.....	6
4.2	Configuration.....	7
4.3	Global usage.....	8
4.4	Usage in CAN Analyzer.....	8
5	Program components and Plugins.....	10
5.1	CAN View.....	10
5.2	CAN Object View.....	12
5.3	CANopen Interpretation (optional).....	14
5.4	CANopen Object View (optional).....	16
5.5	EnergyBus Object View (optional).....	16
5.6	User Interpretation.....	16
5.7	File Logger.....	16
5.8	CAN Replayer.....	16
5.9	CAN/CANopen Scripting (optional).....	17
5.10	CAN message handling.....	19
6	Menu.....	20
7	Settings.....	21
7.1	CAN Settings.....	21
7.2	Program Settings.....	22
8	Support & Contact.....	22

## 1 Introduction

Thank you for using the CANinterpreter. The CANinterpreter is a versatile tool to test, configure and monitor CANopen devices. The following manual explains the installation and use of the program.

## 2 Installation

### 2.1 Windows

To install the tool on Windows start the setup `setup_caninterpreter.exe` and follow the instructions of the setup. The setup creates a shortcut to start the program. At the first connection to CAN the program requires a license file, which is copied to the program and enables the licensed features.

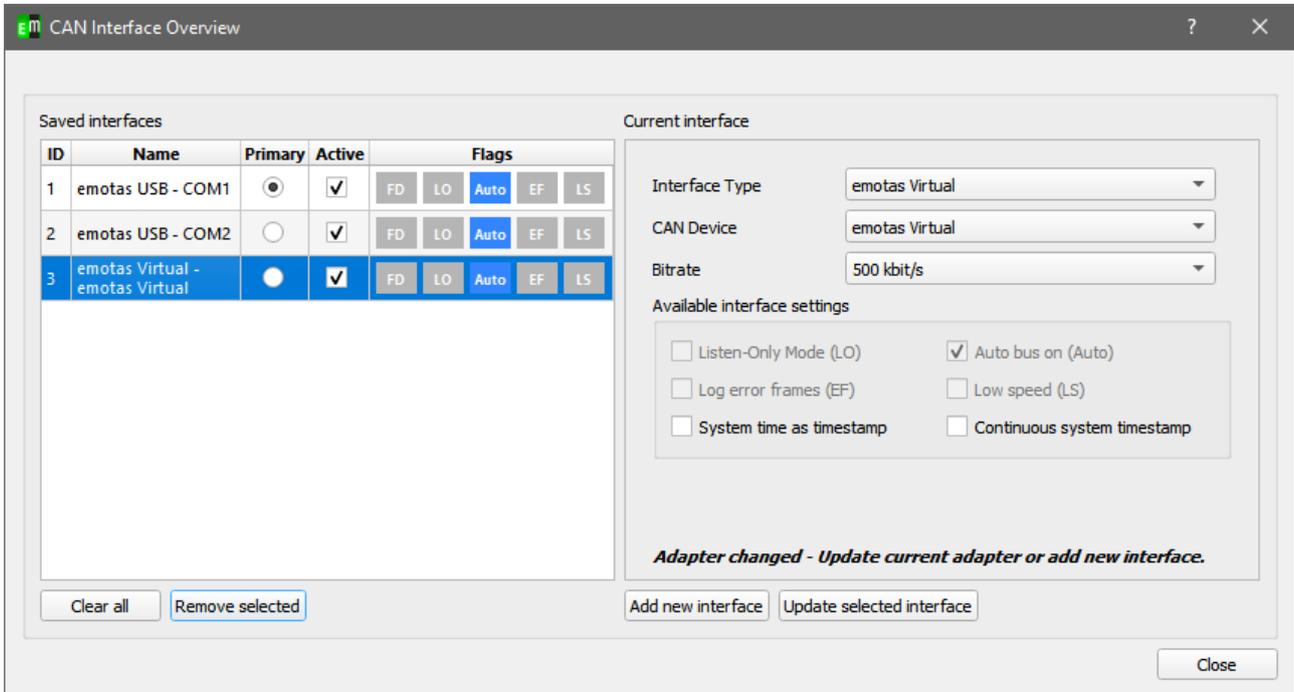
*It is possible to use the tool without license file for one hour with a fixed bit rate of 125 kbit/s for evaluation purposes.*

### 2.2 Linux

To install the tool in Linux just unzip the ZIP file `setup_caninterpreter.zip` into a directory. To start the program run the script `CANinterpreter.sh` in this directory. At the first connection to CAN the program requires a license file, which is copied to the program and enables the licensed features

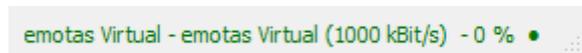
### 3 First steps

The first step at the very first start of the tool is the configuration of the CAN interface. Open CAN interface settings at the menu entry “Connection → CAN Interface Settings”. The following mask appears.



Choose the type of the CAN interface, the name of the CAN device and the bit rate in the CANopen network and confirm the settings with “Add new interface” before closing the dialog.

Connect now the CANinterpreter with the CAN interface via “Connection → Connect”. In the status bar you can see now “Connected to” and the name of the CAN device and the current bitrate.



## 4 CAN Multiinterface handling

### 4.1 General

With version 1.11.0 of this tool the possibility to connect to several CAN adapters at the same time was added, if your license supports this. There is always **one primary adapter** and the possibility to set other adapters active.

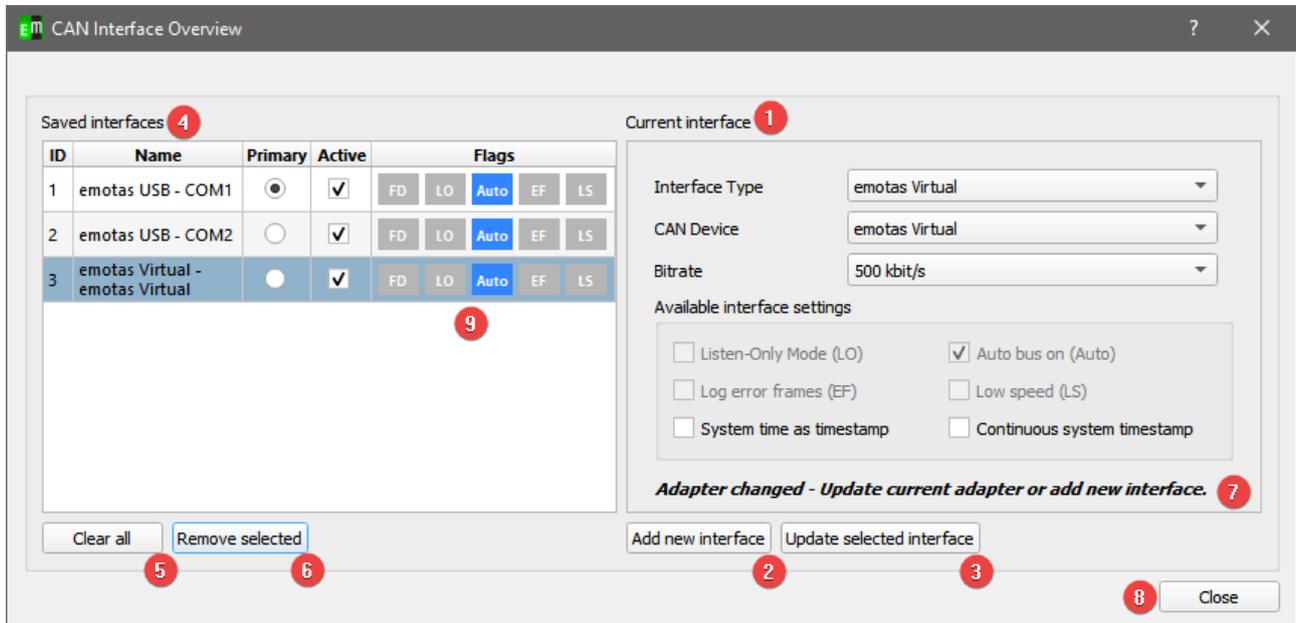
All write actions of the tool are handled by the primary adapter. This includes, for example, interaction with the Object Browser, Network Overview, PDO Configuration, PDO Tx & Rx, but also scripting, LSS Master etc.

The CAN messages of the active adapters are displayed in the supported CAN analyzers. The supported analyzers include CAN View, CAN Object View, CANopen Interpretation, CANopen Object View, EnergyBus Object View, User-defined Interpretation, J1939 Interpretation, J1939 Object View and ISO-TP Interpretation.

Each adapter has its unique ID. See the screenshot below. This ID can be found in the column "IF" in the individual CAN analyzers.

## 4.2 Configuration

The configuration of several CAN lines is done via the "CAN Interface Overview" dialog.

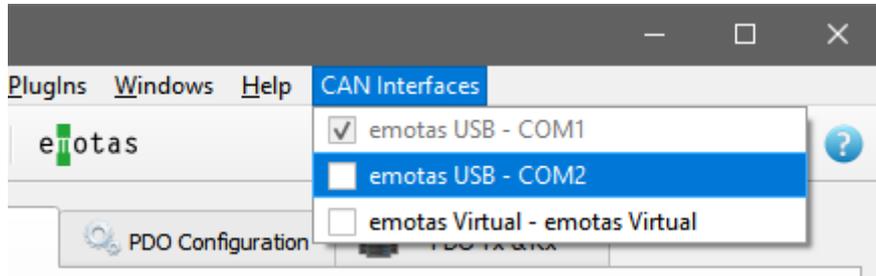


1. The "Current Interface" area contains all information about the currently selected adapter or possible new adapter.
2. "Add new interface" adds a new adapter with settings from "Current Interface" to the list.
3. "Update selected interface" updates the currently selected adapter in the left list to the new settings.
4. The "Saved Interfaces" overview shows the currently saved interfaces. In this list, one adapter is always the primary and therefore active. Additionally, several adapters can be set to active if the corresponding license is available.
5. "Clear all" deletes all set interfaces from the list.
6. „Remove selected“ deletes the currently selected interface.
7. The message "Adapter changed - Update current adapter or add new interface" appears as soon as an adapter is selected and the input mask has been changed by the user. It is then necessary to apply the changes to the current adapter or add it as a new adapter.
8. "Close" closes the dialog.
9. The Flags column highlights the flags set for the adapter. The following abbreviations are used:
 

FD - CAN FD	LO - Listen-Only Mode	Auto - Auto bus on
EF - Error frames	LS - Low Speed	

### 4.3 Global usage

After setting several active adapters and successfully connecting them, a new menu item "CAN Interfaces" is displayed in the main window. The primary adapter can be changed at any time using this menu.



### 4.4 Usage in CAN Analyzer

Each supported analyzer shows the ID of the adapter in the column "IF" and has the possibility to display, hide and color the CAN messages of the different adapters. The tooltip of the column shows the name of the currently set adapter of the ID.

	IF	Time	
19	2	41656.772000	156
20	2	emotas USB - COM2	156
23	2	41657.770000	156

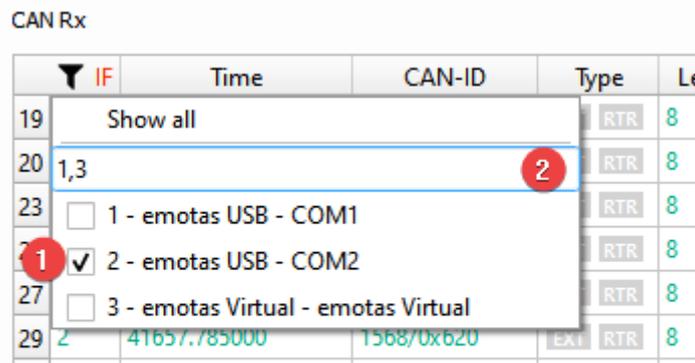
Note that only the ID and not the corresponding adapter is saved when importing and exporting data. So there could be a mismatch when importing datafiles.

#### Quick filter

In the individual views, each active adapter can be quickly hidden by right-clicking on the title of the "IF" column. At this point a menu with all active adapters opens, which can be deselected by a simple click. (1)

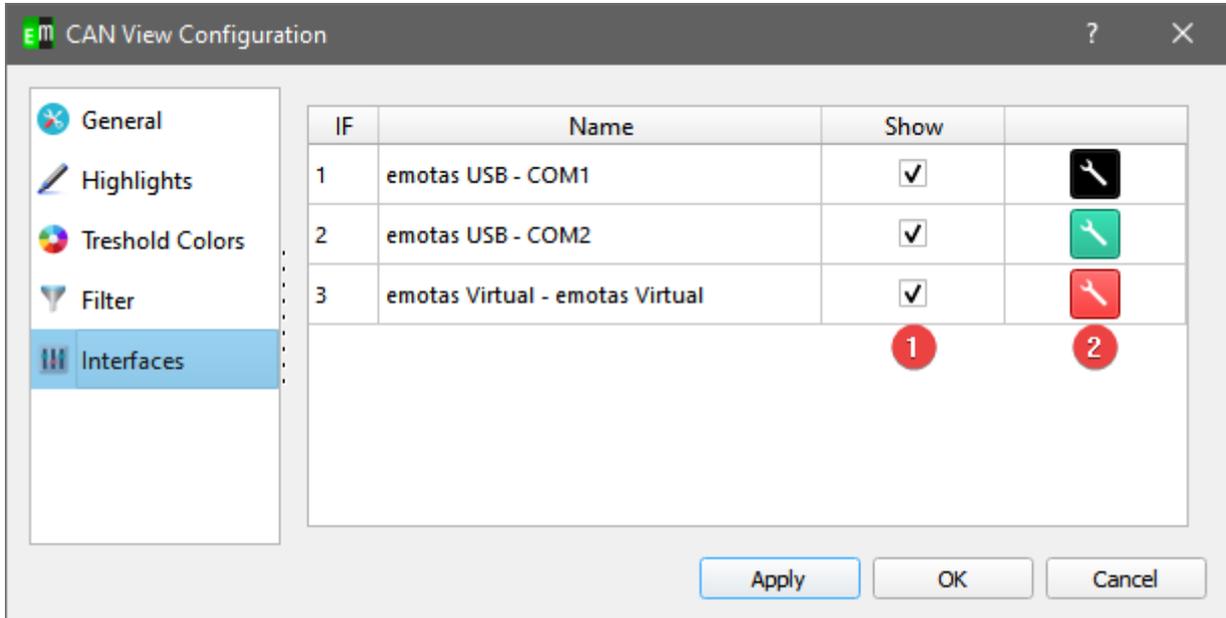
Since it can happen that the current list of active adapters does not match the displayed data, for example, after importing old data, each adapter can be filtered with its unique ID by entering this ID in the comma-separated list and confirming with Enter. (2)

The IDs of the adapters are shown in the column "IF".



## Configuration in the CAN Analyzer

In each view, an extended configuration of the adapters is possible via the menu item "Configuration". Here you can also show or hide the adapter and set the font color of the CAN messages for the adapter.



With the boxes in the "Show" column (1) the CAN messages of the adapters can be shown or hidden as with the quick filter. With the buttons of the last column (2) the font color can be changed. The buttons are colored according to the selected font color.

## 5 Program components and Plugins

### 5.1 CAN View

The CAN View shows received and transmitted CAN messages. To send a CAN message the Transmit table below can be used. The values for CAN-IDs, DLC and data can be specified as decimal values or as hexadecimal values with leading 0x.

Up to 16 transmit messages can be defined. The selected one is sent by clicking on the Transmit button. Cyclic messages can be sent automatically by the tool, if the value in the column 'Interval(ms)' is larger than 0.

In the menu of the CAN View windows a filter for distinct CAN IDs can be defined. A list of CAN IDs can be defined, which either can be displayed or ignored. The list can be a list of single values like (100,200,0x400,0x500) or a range (0x100-0x200) or a combination of both e.g. 1,2,0x300-0x400,720.

The filter type defines the behavior of the filter:

- PASS – only the CAN IDs in the filter list are displayed
- REJECT – the CAN IDs in the filter list are ignored, all other CAN IDs are displayed

The screenshot shows the CAN View interface with the following components:

- Toolbar:** Includes icons for Autoscroll, Relative time, Filter enabled, Refresh, and a dropdown menu set to HEX. A status bar shows '436/2000000' and a 'Clear view' button.
- Search:** A search bar with a magnifying glass icon.
- CAN Rx Table:**

IF	Time	CAN-ID	Type	Len	0	1	2	3	4	5	6	7	Notes	
299	1	0.002682	1535/0x5ff	EXT RTR	8	0x60	0x00	0x1a	0x02	0x00	0x00	0x00	0x00	
300	1	0.001257	1663/0x67f	EXT RTR	8	0x2f	0x00	0x1a	0x00	0x02	0x00	0x00	0x00	
301	1	0.002652	1535/0x5ff	EXT RTR	8	0x60	0x00	0x1a	0x00	0x00	0x00	0x00	0x00	
302	1	0.001213	1663/0x67f	EXT RTR	8	0x23	0x00	0x18	0x01	0xff	0x01	0x00	0x00	
303	1	0.002654	1535/0x5ff	EXT RTR	8	0x60	0x00	0x18	0x01	0x00	0x00	0x00	0x00	
- CAN Tx Table:**

∞	Interval (ms)	CAN-ID	Type	Len	0	1	2	3	4	5	6	7	Name
1	<input type="checkbox"/> 1000	0xc	EXT RTR	1	0x1								
2	<input type="checkbox"/> 1000	0xd	EXT RTR	2	0x1	0x2							
3	<input type="checkbox"/> 1	0xe	EXT RTR	3	0x1	0x2	0x3						
4	<input type="checkbox"/> 1	0xf	EXT RTR	4	0x1	0x2	0x3	0x4					
5	<input type="checkbox"/> 1	0x7ff	EXT RTR	8	0xff								
- Transmit Button:** A button labeled 'Transmit' is located at the bottom right of the CAN Tx table.

The time stamp of the CAN message can be absolute or relative values and the accuracy depends on the used CAN interfaces and the operation system. For most CAN interfaces no TX time stamp is available.

The recorded CAN messages can be exported into text files by CAN View → Export CAN-Logging.

The format of the save text is explained below:

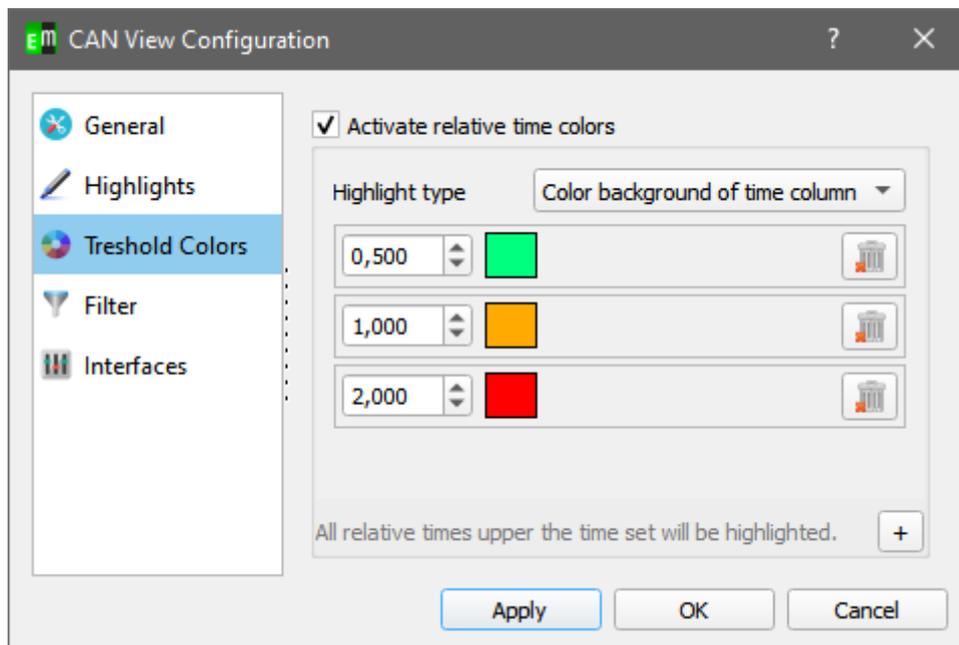
```
3.653302 0x5a0/1440 (8): 43 18 10 03 00 00 00 00
time stamp
          CAN-ID
                DLC
                    Data in hexadecimal notation
```

The CAN View PlugIn is part of the standard scope of delivery of the CANopen DeviceExplorer.

### Relative times

If “relative time” is set, each message could be used as time reference by right-clicking on it and select “Use messages as time reference”. The relative times of each other message is now calculated to this messages.

Furthermore if “relative time” is set, the messages could be highlighted. To setup the thresholds and colors select “Configuration” in the “CAN View” menu.



In this dialog you could add new thresholds and add a color. If the relative time of one message is higher than the threshold, the message is highlighted in the chosen color.

## 5.2 CAN Object View

The CAN Object View shows all received CAN messages in the so called 'Object View'. That means all received CAN IDs are shown in a table with the last data and the number of receptions.

CAN Object View

Autoscroll Relative time Toggle filter Refresh HEX Clear view

Count	IF	Time	CAN-ID	Type	Len	0	1	2	3	4	5	6	7
518	1	468.267...	511/0x1ff	EXT RTR	6	0x04	0x00	0x00	0x00	0x00	0x00		
297	1	468.866...	1919/0x...	EXT RTR	1	0x05							
16	1	197.385...	1663/0x...	EXT RTR	8	0x23	0x00	0x18	0x01	0xff	0x01	0x00	0x00
16	1	197.388...	1535/0x...	EXT RTR	8	0x60	0x00	0x18	0x01	0x00	0x00	0x00	0x00

The table can be sorted by the number, the time stamp of the CAN ID. A filter can be configured in the same way as in the CAN View PlugIn. The CAN Object View is included in the standard scope of delivery of the CANopen DeviceExplorer.

### CAN Object View Configuration

At the CAN Object View you could highlight changed bytes or highlight the whole row if a CAN ID was not received within a certain time. You will find those settings via the “CAN Object View”- menu at “Configuration”.

#### General

- Refresh interval

This time determine the refresh rate of the view. The lower this time, the more computing time is required.

- Highlight changed bytes

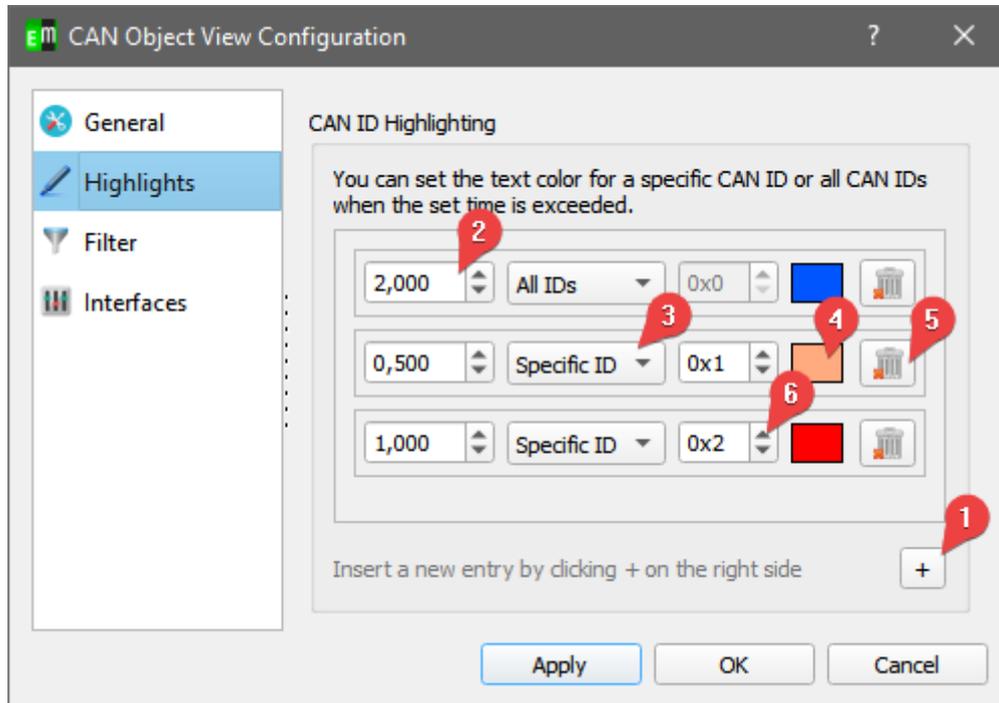
If this option is enabled changed bytes will be printed bold, if they changed from message to message.

#### Filter

Please refer to the description of the [CAN View filtering](#).

## Highlighting

With the Highlighting-Tab you could set a textcolor for the case a special CAN ID or all CAN IDs were not received within a certain time.



1. Add a new entry for the highlighting.
2. The time in seconds after which the CAN ID should be highlighted if they was not received.
3. Select "Specific ID" if you want to highlight only a specific CAN ID or select "All IDs" to highlight all CAN IDs.
4. Select the color for the text if the CAN ID was not received within the time.
5. Deletes the entry
6. Set the specific CAN ID

### 5.3 CANopen Interpretation (optional)

The CANopen Interpreter interprets all received CAN messages according to the CANopen protocol and displays the CANopen service of the message, the source or target node ID and a service-specific interpretation in a human-readable format. Additionally the CANopen Interpreter can interpret the content of PDO messages according to the PDO mapping as defined in EDS or DCF files.

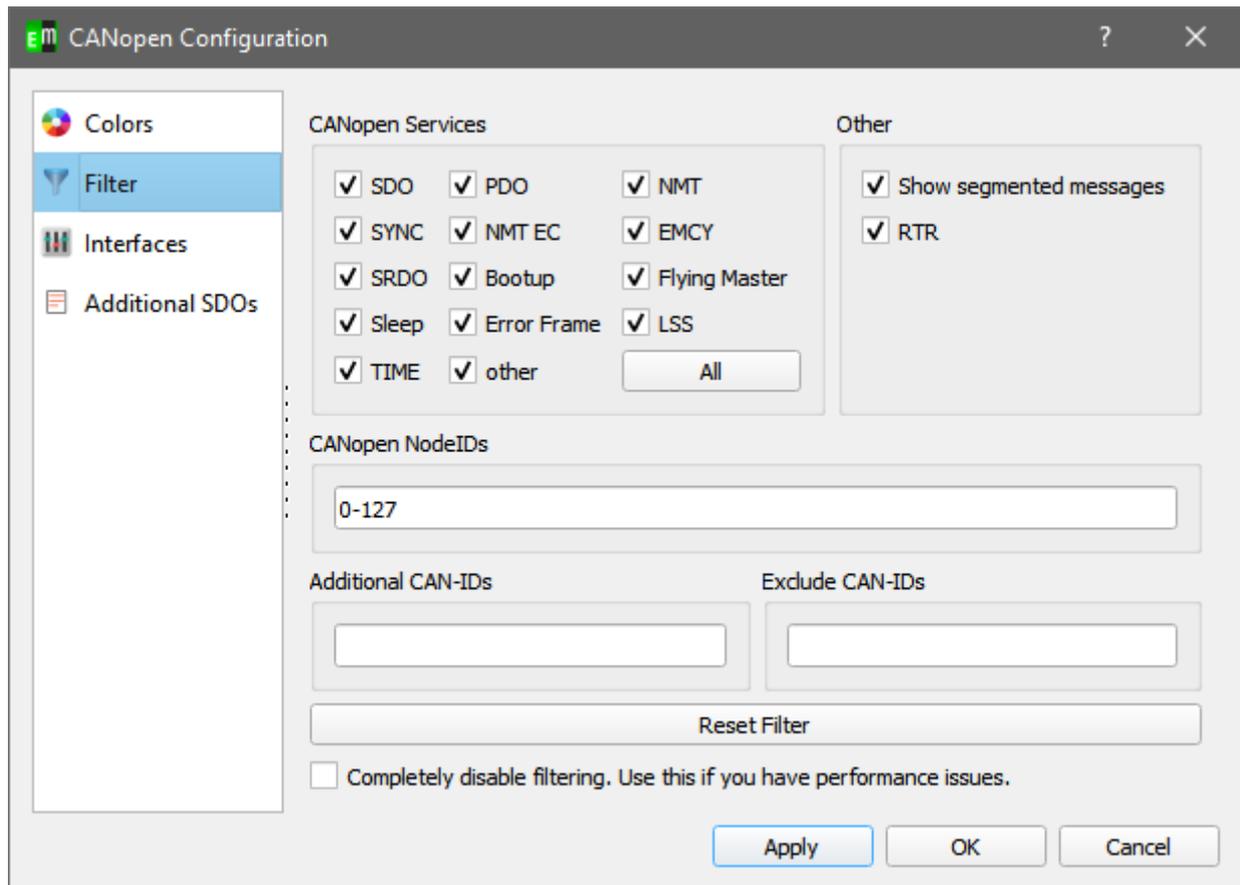
Time Stamp	CAN-ID	Type	Node-Id	Data	Interpretation
1491479795.991180	1409/0x581	SDO	1 - device according CiA 401	43 18 10 02 15 08 00 00	r Res exp 1018:2 Val 0x815
1491479795.995680	1409/0x581	SDO	1 - device according CiA 401	43 18 10 03 01 00 01 00	r Res exp 1018:3 Val 0x10001
1491479795.995680	1537/0x601	SDO	1 - device according CiA 401	40 18 10 03 00 00 00 00	r Req - 1018:3
1491479796.039180	1537/0x601	SDO	1 - device according CiA 401	40 18 10 04 00 00 00 00	r Req - 1018:4
1491479796.041180	1409/0x581	SDO	1 - device according CiA 401	43 18 10 04 01 00 00 00	r Res exp 1018:4 Val 0x1
1491479796.300180	1793/0x701	Error Control	1 - device according CiA 401	05	Operational toggle Bit 0 Node State Operational
1491479796.302680	385/0x181	PDO	1 - device according CiA 401	00 00 0a 43 50 4e 00 00	0 17162 20048 0 Read Analogue Input 16 Bit:Dial left (6401:1) = 0 Read Analogue Input 16 Bit:Dial right (6401:2) = 17162 Read Analogue Input 16 Bit:Slider left (6401:3) = 20048 Read Analogue Input 16 Bit:Slider right (6401:4) = 0
1491479796.918180	385/0x181	PDO	1 - device according CiA 401	00 00 0a 43 38 52 00 00	0 17162 21048 0
1491479797.262180	385/0x181	PDO	1 - device according CiA 401	00 00 0a 43 20 56 00 00	0 17162 22048 0
1491479797.340180	1793/0x701	Error Control	1 - device according CiA 401	05	Operational
1491479797.582180	385/0x181	PDO	1 - device according CiA 401	00 00 14 21 20 56 00 00	0 8468 22048 0
1491479797.670680	385/0x181	PDO	1 - device according CiA 401	00 00 aa 22 20 56 00 00	0 8874 22048 0
1491479797.725180	385/0x181	PDO	1 - device according CiA 401	00 00 5a 24 20 56 00 00	0 9306 22048 0
1491479797.965680	385/0x181	PDO	1 - device according CiA 401	00 00 ba 3b 20 56 00 00	0 15290 22048 0
1491479797.991680	1409/0x581	SDO	1 - device according CiA 401	4f 18 10 00 04 00 00 00	r Res exp 1018:0 Val 0x4
1491479797.991180	1537/0x601	SDO	1 - device according CiA 401	40 18 10 00 00 00 00 00	r Req - 1018:0

264 (264) messages

A filtering of the CAN message is possible by various criteria:

- CANopen Service – only the enabled services are displayed
- CANopen NodeID – only the selected node IDs are displayed. A definition of ranges like e.g (1,2-30,40) is possible.
- Additional CAN-IDs: CAN IDs, which are rejected by the previous filters can be enabled again.
- Exclude CAN-IDs: CAN IDs, which have passed the previous filters can be filtered out selectively. A definition of ranges like e.g. 100,0x710-0x730 is possible.

Interpreted CANopen messages can be exported as text files and raw CAN loggings can be imported and interpreted as well.



## 5.4 CANopen Object View (optional)

The CANopen ObjectView combines the CANopen interpretation with an object view of the CAN messages. All received CAN-IDs are shown with the last interpreted values.

## 5.5 EnergyBus Object View (optional)

The EnergyBus ObjectView combines the EnergyBus interpretation with an object view of the CAN messages. All received CAN-IDs are shown with the last interpreted values.

## 5.6 User Interpretation

The user interpretation allows to interpret the CAN message according to the configuration of the user.

## 5.7 File Logger

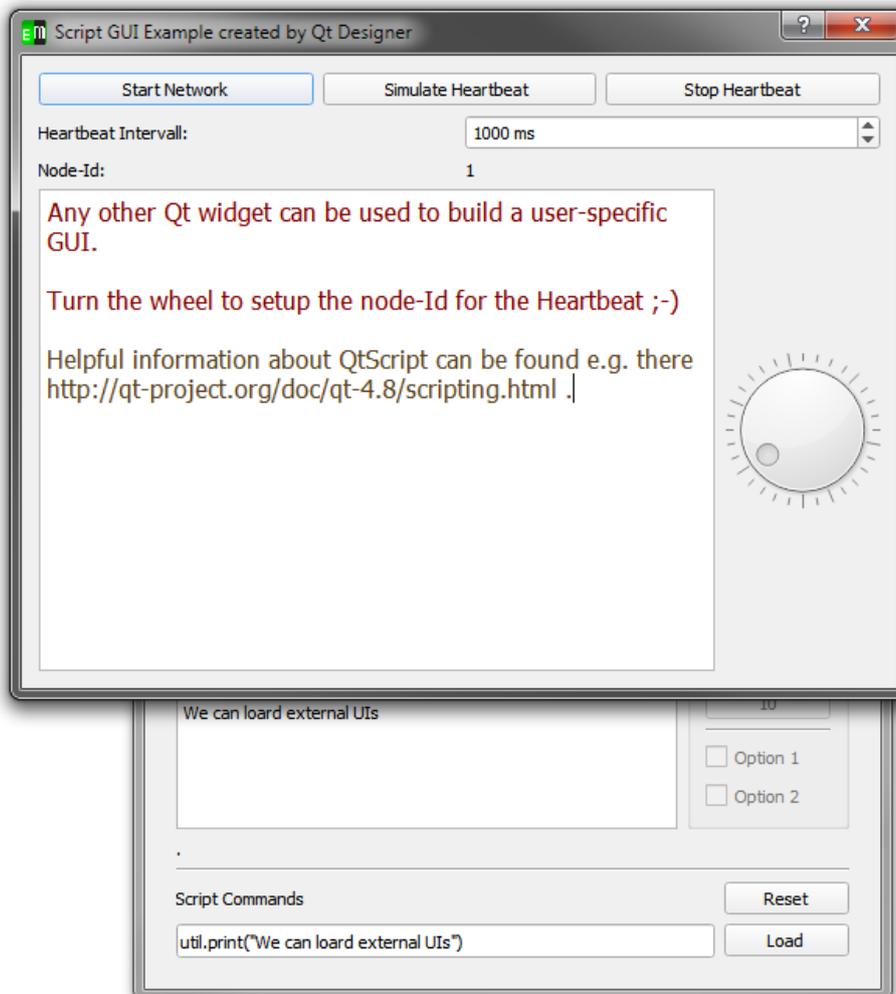
The File Logger is able to record CAN messages directly into log files according to certain trigger conditions.

## 5.8 CAN Replayer

The CAN Replay can be used to send CAN messages that has been logged in advance. Specific CAN-IDs may be skipped for transmission and the timings are reproduced as good as possible depending on the operating system.

## 5.9 CAN/CANopen Scripting (optional)

The Scripting Interpreter provides the possibility to run QtScript(JavaScript) programs with special extensions for CANopen.



Even own user interfaces can be generated by the QtDesigner and used in scripts.

**All additional CANopen-specific commands are explained in the separate document “CANopen Scripting Interpreter – API Reference” (cde\_script\_api.pdf).**

The following example shows the usage of a QtScript program:

```
// print something to console and set device to operational
util.print("Test of simple device");
nmt.preopNetwork();
nmt.startNode(32);
i = 0;
util.print("We are in " + util.pwd());

// set node id for SDO access
sdo.setNodeId(32);
```

```

// loop over objects 0x4000 to 0x04010
for (object = 0x4000; object < 0x4005; object ++) {
    str = "Test object ";
    str = str + object;
    util.print(str);

    // write value to object 0x4000..
    result = sdo.write(object, 0x0, UNSIGNED32, i);
    if (result == "SDO_OK") {
        util.print(" Write OK");
    } else {
        util.print(" Write NOT OK");
    }
    // wait a bit to allow device update its internal values
    util.msleep(10);

    // read from 0x4100.. and expect same value
    result = sdo.read(object+0x100, 0x0, 0x07);
    if (result == i) {
        util.print("Read OK");
    } else {
        util.print(" Read NOT OK");
    }
    i++;
}

// user defined function that can be called from Scripting tab
function urk(count) {
    for (i = 0; i < count; i++) {
        nmt.startNetwork();
        nmt.stopNetwork();
        var dlc = 4;
        var canId = 0x100 + i;
        can.sendBaseFrame(canId, dlc, 1 , 2 , 3 ,4 , 0, 0, 0 ,0);
    }
}
// call user defined function
urk(4);
// setup cyclic timer every 2 seconds
timerId = util.every(2000, "urk(10)" );

```

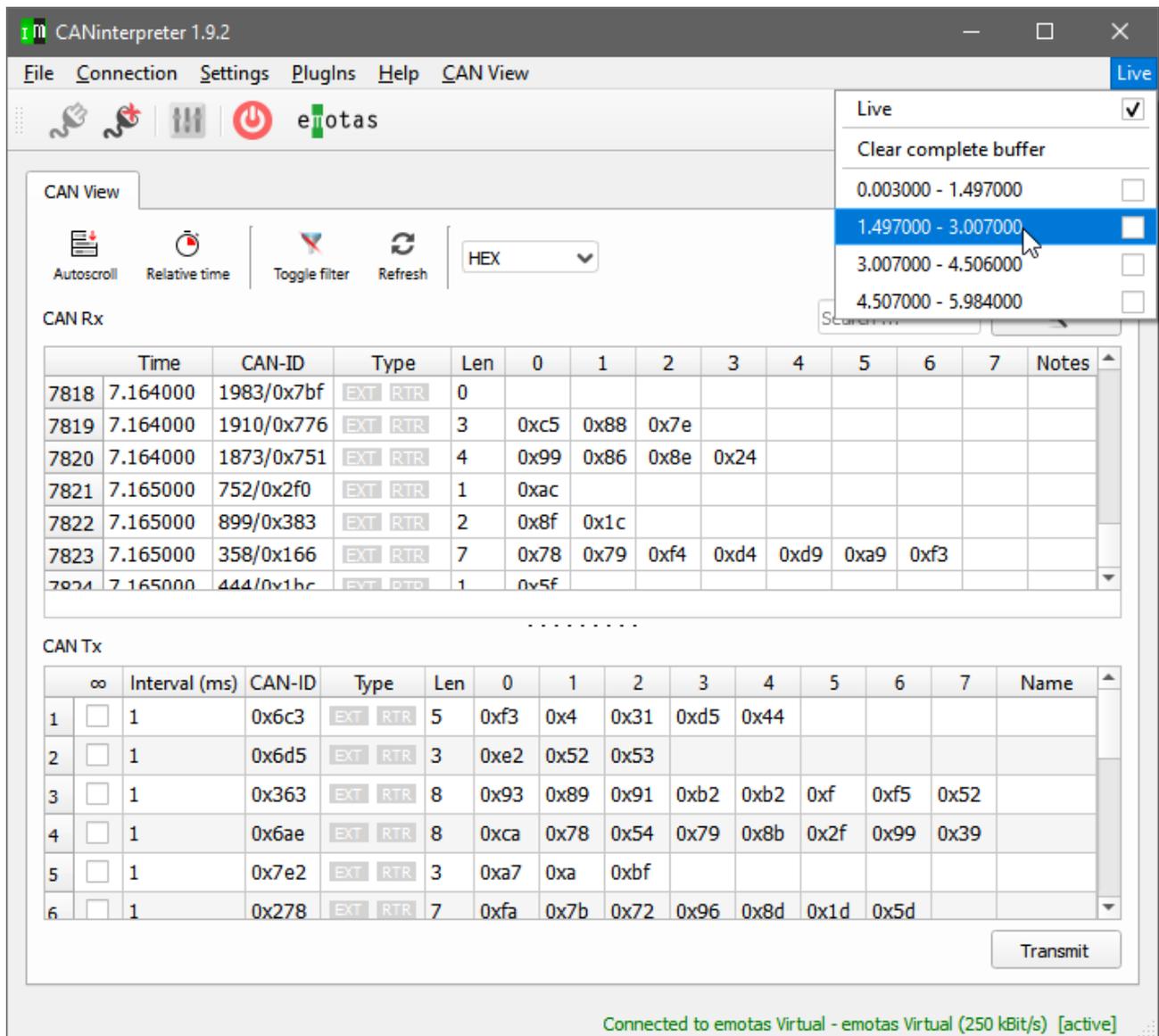
This example is installed with the CDE as example1.js.

Script commands can be entered into the command line at the bottom of the window. Complete script files can be loaded as well if a path to a script file is specified in the command line. The command line stores its history and using the cursor buttons up and down older commands can be selected again. The command line history is stored when the program is closed.

### 5.10 CAN message handling

The received messages are stored as message package to the hard disc drive if the “Maximal message count” (configurable in the options) is reached and removed from the internal buffer.

The saved messages could be viewed via the menu in the upper right corner of the window.



Each message package holds the timestamp of the first and of the last message as name.

If a message package is selected newly received message are still being processed in the background and could be viewed by switching back to “Live”.

## 6 Menu

The menu provides access to various functions and settings of the CANinterpreter.

### File

- **Export CAN Logging**  
Export of CAN messages as text file. Import and interpretation in the PlugIns at a later point of time is possible.
- **Quit**  
Quit the application.

### Connection

- **CAN Interface Settings**  
Dialog to configure CAN interface and bit rate
- **Connect**  
Connect to CAN using the configured interface
- **Disconnect**  
Disconnect from CAN

### Settings

- **Filter Settings**  
Filter configuration for CAN messages. Only valid for the main part of the program. There are separate filters for all PlugIns.
- **Options**  
Opens the options dialog to configure various settings of the program.
- **Save**  
Saves the current settings. Enabled “Settings → Options → Save settings automatically at exit” saves the settings automatically when the program is quit.
- **Export Settings**  
Export of the current settings into a configuration file. Can be used to store various settings of different use cases.
- **Import Settings**  
Import of settings from a configuration file.
- **Update Licence File**  
Dialog to select a new license file. The content of the license file can be viewed before the

import.

- **Check for Updates**

Query the web server for updates of the tool. Beside the IP address no additional data is transmitted.

## PlugIns

Menu to activate various extensions of the CANinterpreter. The availability of PlugIns depends on the license.

## Help

- **Manual**

Shows the complete manual as PDF file.

- **About**

Shows 'about' dialog including license information.

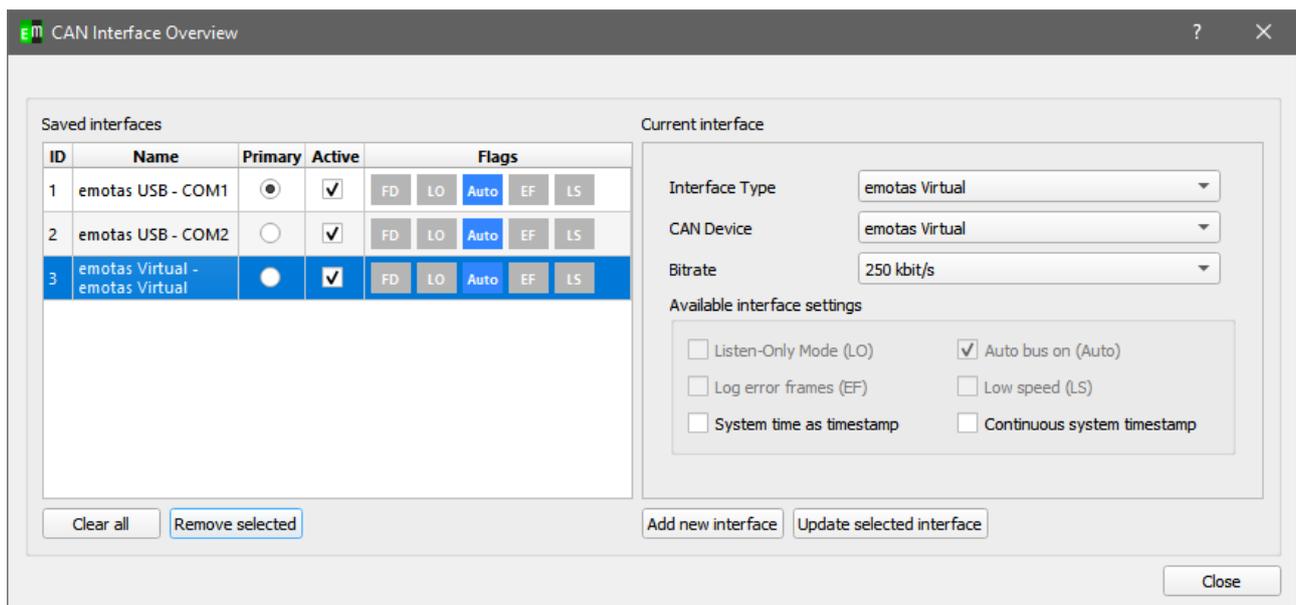
- **About Qt**

Information about the Qt framework and license information for the used Qt components.

## 7 Settings

### 7.1 CAN Settings

The CAN interface can be configured using the CAN settings dialog.



- **Interface Type**

Selection of the CAN interface. Currently different CAN interface manufacturers are

supported on Windows and on Linux.

- **CAN Device**

Depending on the Interface type the available CAN devices are listed here to choose from. E.g. can0, can1, ...van0 for SocketCAN or usb1...usb8 for PCANBasic.

- **Bit Rate**

Configuration of the bit rate of the CANinterpreter. The configured bit rate must match with the bit rate of the CAN network. For SocketCAN it is necessary to set the bit rate before starting the program

- **Advanced Settings**

Some CAN Interfaces support some additional settings that can be configured there. These might be:

- Listen-Only-Mode
- User specific bit rates
- Activation of terminator resistors

## 7.2 Program Settings

The option dialog provides access to various settings of the CANinterpreter.

### General Settings

- Warning when exit with active CAN connection
- Save Settings automatically at exit
- Auto Connect to CAN after startup

Automatic setup of a connection to CAN interface with program start

### Scripting

- Autostart Script at connect

Autostart Script starts automatically with active Scripting Interpreter and connection to CAN. In combination with the "Auto Connect to Can after startup" the script starts automatically after starting the program

## 8 Support & Contact

On all questions and upcoming problems on CANinterpreter you may contact us via email ([support@emotas.de](mailto:support@emotas.de)) or by phone +49(0)3461/794160. If a CANopen device does not react as expected, a logging of the CAN communication is useful for the analysis. Please send us your current CAN logging by email, ideally also before you contact us by phone.