Diagnostic SW PP2CAN

(CAN bus communication diagnostics)

User Guide
Document Version 0.71 EN

Ing. David Španěl
pp2can@seznam.cz
www.pp2can.wz.cz, www.usb2can.wz.cz

Verze dokumentu:

Verze:	**Datum:**	**Změny:**
0.50	2.12.2005	The first part of english version.
0.70	12.1.2006	Added more informations.

Table of contents:

1. Introduction

The purpose of the PP2CAN diagnostic SW is the diagnostics of CAN bus communication (Controller Area Network), as well as development of applications capable to communicate using this bus. It allows to analyze received data, generate bus data, save communication logs and play back these records. To connect PC to the CAN bus, you can use:

- PP2CAN: low cost LPT adapter, ideal for introduction to CAN bus.
- USB2CAN: professional USB adapter for bus diagnostics and process control both laboratory and industrial.
- ETH2CAN: Ethernet adapter (under construction)
- PCI CAN interface from IXXAT (experimental)

2. Installation

You can extract the contents of the pp2can.zip archive saved on the companion CD to any folder on your hard drive. It is only necessary to set the parallel port address for the PP2CAN adapter or to select right USB device for the USB2CAN adapter. You can carry out this in the Options dialog.

For PP2CAN adapter you must select the right adapter variant and parallel port address. You can find it in computer’s BIOS. However, Windows allows for mapping to a different address. Thus, you must use the port address stated in the Device Manager → LPTn → Resources. If an “ERROR: Device PP2CAN not connect” message appears on program startup, make the right address setting and save it. Don’t change the Windows parallel port address setting while PP2CAN is running. Otherwise the program could “freeze”. Sometimes it takes a bit of experimenting with BIOS parallel port mode (Normal, SSP, EPP, ECP). On most computers the Normal or SSP setting works well. In the previous versions (1.x) of the PP2CAN diagnostic SW the setting of the parallel port address was in the main window.

For the HW PP2CAN to work well, you have to plug the power cable in the USB socket with the high-speed variant. With low-speed or single wire variant, connect the external 12V power source to the CANON 9-pin connector. Low-speed USB2CAN adapter requires external power source as well.

Only one instance of PP2CAN or other PP2CAN API-based SW is allowed for one computer at a time. Go to Task Manager and check the number of running instances on the Processes tab if problems occur. However, it is possible to run several instances of PP2CAN with USB2CAN adapters, or one SW with PP2CAN adapter and several PP2CANs with USB2CAN adapters. The number of simultaneous instances of SW with virtual CAN port V2CAN is unlimited.

USB2CAN adapter is based in FTDI's USB chip. You can obtain latest driver version on the company site as well. The basic driver set is on the companion CD. FTDI provides two kinds of drivers: D2XX and VCP. The USB2CAN interface requires to use D2XX drivers. You can specify the USB adapter you want to work with using: Device description, Serial Number or Device number in Options dialog. The CAN must be restarted in order the changes to take effect - use the Reset button or restart the program after the new setting is saved (use the Save options button).

In the USB2CAN_driver_mod directory on the companion CD you can find modified D2XX drivers that allow to simultaneously run USB2CAN adapter and other company's devices that use the VCP drivers. You can find the description of the driver modification at www.pp2can.wz.cz as well.

If there is no adapter properly set at the start of the PP2CAN diagnostic SW version 2.000 and later, the V2CAN virtual port is automatically opened.

3. Main window

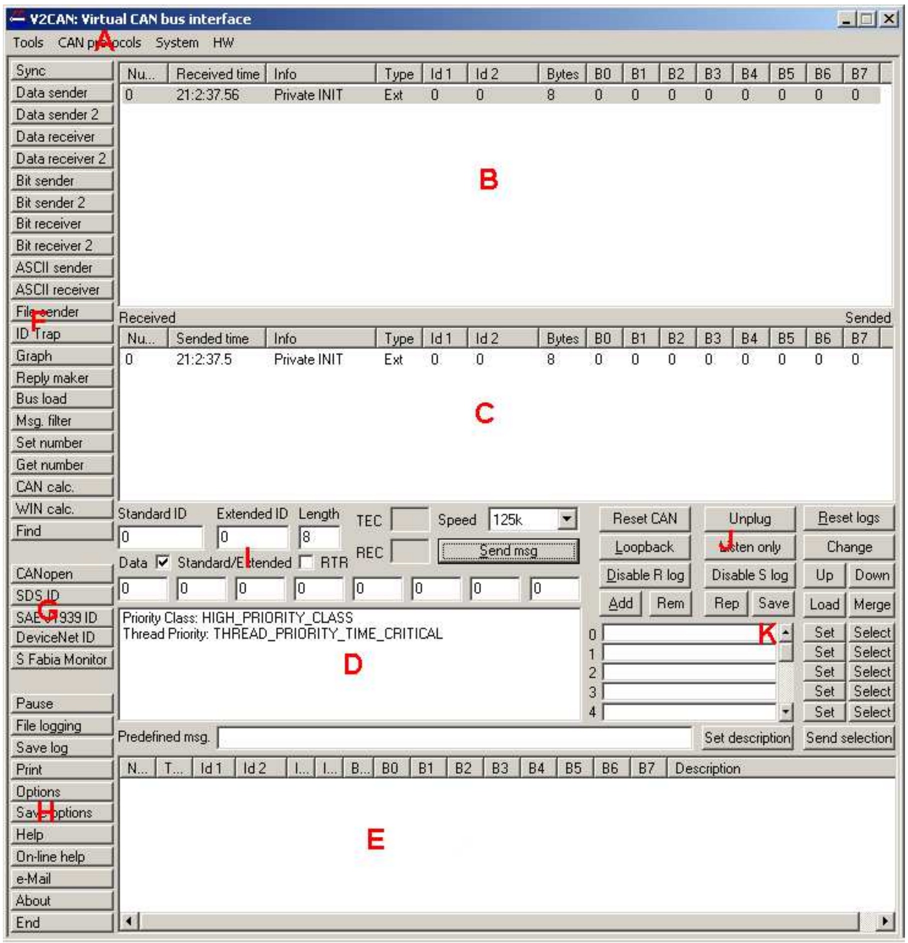
Picture 1 shows the window you will see first after the PP2CAN startup. There are several parts of the window. The red letters indicate these particular parts.

A) Program menu. This menu gives you access to the tools for communication analysis, data generation, system tools and others.
B) Received messages log window. Contains history of received messages. If not otherwise specified in filter configuration, all messages are received by default. You can sort the data by clicking the caption of the column. Click the ***Disable R log*** button to disable (or hold up) logging into this dialog.
C) Manually sent messages log window. If not otherwise specified in filter configuration, all messages are received by default. You can sort the data by clicking the caption of the column. Click the ***Disable S log*** button to disable logging into this dialog. Contains history of sent messages.
D) Information window for displaying the program messages and CAN bus API interface.
E) Predefined messages database window. To simplify your work, you can create frequently used message databases. The list of these messages is than displayed in this window.
F) Bus communication analysis and data generation toolbar.
G) High-level CAN protocol toolbar.
H) Auxiliary toolbar (Print, Help, etc.)
I) Setting and generating panel for manually generated message. After the message is sent, you must fill in the identifier, select the type and select whether it is data frame or RTR-type message (data request). Then, in case of data frame, select its length (0-8) and set the data bytes. The lowest data byte (B0) is on the left, the highest is on

the right. Press the **Send msg** button to send the message. Standard identifier ranges from 0-2047 (11 bits). If a message with extended identifier is being sent, you must fill in the extended part as well (18 bits). The range is 0-262143.

J) Control panel of the program.

K) Predefined messages database panel. It is useful to store frequently used messages in the database of predefined messages. By doing this, you can avoid constant overwriting of identifiers and data when sending manually. Press the **Add** button to add the filled message into the database. You can than save the database (by pressing the **Save** button) or load it back (by pressing the **Load** or **Merge** button). You can add a comment to a message as well. Use the **Set description** button to add the comment. Double-click the database item to copy it into the editing field for manual message sending. If you use the mouse together with the **Ctrl** (**Shift**) button to select more items, you can send them together by pressing the **Send selection** button. You can save these message selections into the message selection lists. You can define up to 10 particular selections. In addition, compared to message logs, the database shows the identifier in binary format.



Received

Nu...	Received time	Info	Type	Id 1	Id 2	Bytes	B0	B1	B2	B3	B4	B5	B6	B7
0	21:2:37.56	Private INIT	Ext	0	0	8	0	0	0	0	0	0	0	0

Sended

Nu...	Sended time	Info	Type	Id 1	Id 2	Bytes	B0	B1	B2	B3	B4	B5	B6	B7
0	21:2:37.5	Private INIT	Ext	0	0	8	0	0	0	0	0	0	0	0

Standard ID: 0 Extended ID: 0 Length: 8
Data ☑ Standard/Extended ☐ RTR
0 0 0 0 0 0 0 0
Speed: 125k

Priority Class: HIGH_PRIORITY_CLASS
Thread Priority: THREAD_PRIORITY_TIME_CRITICAL

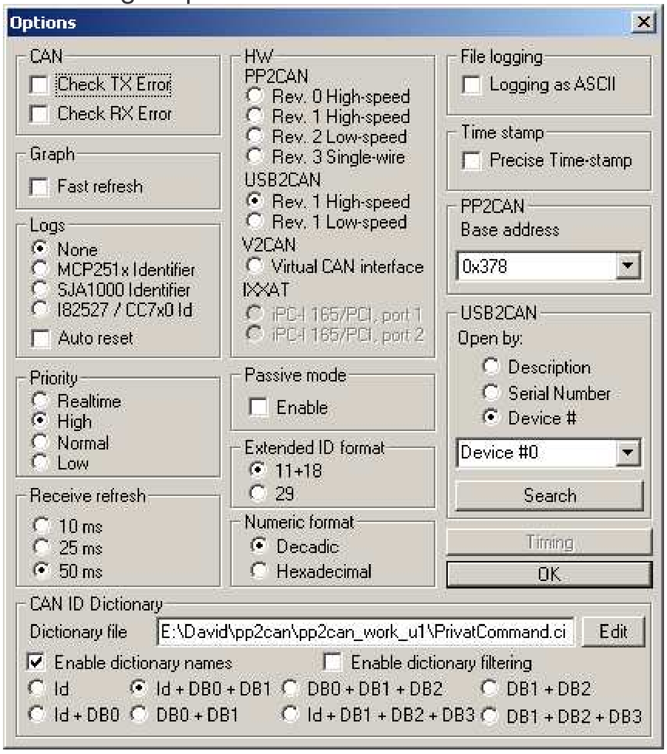
N...	T...	Id 1	Id 2	I...	I...	B...	B0	B1	B2	B3	B4	B5	B6	B7	Description

Picture 1: Main window of the PP2CAN diagnostic SW.

4. Settings

The Options dialog enables you to carry out all of the settings. Press the Options button on the Auxiliary toolbar in the bottom-left corner of the main window to open this dialog. Press the Save button on the same toolbar to save changes and to use new settings next time the program is started. If you don’t save the settings, it is valid only before closing the program.

You can see the Options dialog on picture 2.



CAN
□ Check TX Error
□ Check RX Error

Graph
□ Fast refresh

Logs
● None
○ MCP251x Identifier
○ SJA1000 Identifier
○ I82527 / CC7x0 Id
□ Auto reset

Priority
○ Realtime
● High
○ Normal
○ Low

Receive refresh
○ 10 ms
○ 25 ms
● 50 ms

HW
PP2CAN
○ Rev. 0 High-speed
○ Rev. 1 High-speed
○ Rev. 2 Low-speed
○ Rev. 3 Single-wire
USB2CAN
● Rev. 1 High-speed
○ Rev. 1 Low-speed
V2CAN
○ Virtual CAN interface
IXXAT
○ iPC-I 165/PCI, port 1
○ iPC-I 165/PCI, port 2

Passive mode
□ Enable

Extended ID format
● 11+18
○ 29

Numeric format
● Decadic
○ Hexadecimal

File logging
□ Logging as ASCII

Time stamp
□ Precise Time-stamp

PP2CAN
Base address 0x378

USB2CAN
Open by:
○ Description
○ Serial Number
● Device #
Device #0

CAN ID Dictionary
Dictionary file E:\David\pp2can\pp2can_work_u1\PrivatCommand.ci
☑ Enable dictionary names □ Enable dictionary filtering
○ Id ● Id + DB0 + DB1 ○ DB0 + DB1 + DB2 ○ DB1 + DB2
○ Id + DB0 ○ DB0 + DB1 ○ Id + DB1 + DB2 + DB3 ○ DB1 + DB2 + DB3

Picture 2: Options dialog.

.

CAN	***Check TX Error***	If you select this option, the program will read out the TEC register (Transit Error Counter). With PP2CAN adapter this can decrease the number of sent messages per second and increase the number of uncaptured messages (OVR, RCV_OVERFLOW). It is recommended to disable this option if you use this adapter.
	Check Rx Error	If you select this option, the program will read out the REC register (Receive Error Counter). With PP2CAN adapter this can decrease the number of sent messages per second and increase the number of uncaptured messages (OVR, RCV_OVERFLOW). It is recommended to disable this option if you use this adapter.
Graph	***Fast refresh***	Enables smooth rendering of graphs. However, it will increase the CPU usage. It is recommended to disable this option.
Logs	***None***	No additional identifier type is displayed.
	MCP251x Identifier	The identifier is displayed in binary format in the sent/received messages log; it has a format of MCP2510/2515/PIC CAN/PIC ECAN CAN controller registry.
	SJA1000 Identifier	The identifier is displayed in binary format in the sent/received messages log; it has a format of SJA 1000 CAN controller registry.
	I82527/CC7x0 Id	The identifier is displayed in binary format in the

		sent/received messages log; it has a format of I82527/CC7×0 CAN controller registry.
	Auto reset	Only certain number of messages is being kept in the R log (receive log) and S log (send log) if this option is checked. You can set the number in the pp2can.cfg file for version 1.xx and in the file x2can.cfg for version 2.xx respectively. For version 2.012 and higher this option doesn't affect the CPU usage significantly. Log file can contain tens of thousands of messages without rising the usage. However, enabling this option for earlier versions cuts down the CPU usage.
Priority	***Realtime***	Realtime priority setting for PP2CAN SW.
	High	High priority setting for PP2CAN SW
	Normal	Normal priority setting for PP2CAN SW.
	Low	Low priority setting for PP2CAN SW.
Receive refresh	***10/25/50 ms***	Interval setting for selection and processing messages with PP2CAN SW.
HW	***PP2CAN Rev. 0 High speed***	Compatibility mode with original development version.
	PP2CAN Rev. 1 High speed	Standard PP2CAN adapter for high speed CAN bus (ISO 11898-2).
	PP2CAN Rev. 2 Low speed	Standard PP2CAN adapter for low speed CAN bus (ISO 11519, ISO 11898-3).
	PP2CAN Rev. 3 Single wire	Standard PP2CAN adapter for single wire CAN bus (SAE J2411).
	USB2CAN Rev. 1 High speed	Standard USB2CAN adapter for high speed CAN bus (ISO 11898-2).
	USB2CAN Rev. 1 Low speed	Standard USB2CAN adapter for low speed CAN bus (ISO 11519, ISO 11898-3).
	V2CAN Virtual CAN interface	Virtual CAN interface for offline data analysis.
	IXXAT iPC-I 165/PCI port 1	Experimental support.
	IXXAT iPC-I 165/PCI port 2	Experimental support.
Passive mode	***Enable***	Setting only for receiving, prevents from unwanted sending of a message to the bus. Useful when analyzing unknown CAN protocol.
Extended ID format	***11+18***	Extended 29-bit identifier is divided into two parts: standard 11-bit part and extended 18-bit part. PP2CAN SW prefers to use this format. .
	29	Extended 29-bit identifier is not divided.
Numeric format	***Decimal***	Identifier is shown in a decimal format in the log.
	Hexadecimal	Identifier is shown in a hexadecimal format in the log. Note: To put the hexadecimal numbers in most of the fields, just enter the prefix "0x". You can enter the number 255 as a 0xFF. You can enter binary numbers as well using the "b" prefix with 8 bits of maximum length. Enter b1010 as a binary representation of 10.
File logging	***Logging as ASCII***	If File logging is active, data bytes are written in the log as ASCII characters.
Time stamp	***Precise Time-stamp***	More precise identifying of message receiving time, increases CPU usage. However, this option is recommended if you later intend to perform an offline log analysis using File sender and its Real-time mode.
PP2CAN	***Base address***	Parallel port address for PP2CAN adapter connection.

USB2CAN	***Open by Description***	Device description-driven selection variant of USB2CAN adapter.
	Open by Seriál Numer	Serial number-driven selection variant of USB2CAN adapter.
	Open by Device #	Device number-driven selection variant of USB2CAN adapter.
	Search	Start searching for FTDI circuit-based device, i.e. USB2CAN adapter as well.
CAN ID Dictionary	***Dictionary file***	CAN message dictionary file name and path.
	Edit	Display the dialog for selection of CAN message dictionary file.
	Enable dictionary names	Allows to display CAN message names in R and S log in terms of CAN message dictionary.
	Enable dictionary filtering	Allows to filter messages depending on CAN message dictionary
	Id	The dictionary is searched for the message name by identifier.
	Id + DB0 + DB1	The dictionary is searched for the message name by identifier, the first and the second data byte.
	DB0 + DB1 + DB2	The dictionary is searched for the message name by the first, second and third data byte.
	DB1 + DB2	The dictionary is searched for the message name by second and third data byte.
	Id + DB0	The dictionary is searched for the message name by identifier and first data byte.
	DB0 + DB1	The dictionary is searched for the message name by the first and the second data byte.
	Id + DB1 + DB2 + DB3	The dictionary is searched for the message name by the identifier, the first, second and third data byte.
	DB1 + DB2 + DB3	The dictionary is searched for the message name by the second, third and fourth data byte.
Timing	***Timing***	User setting of communication speed and sampling point for PP2CAN adapter. This adapter uses MCP2515 CAN bus controller and 20 MHz time crystal with 20 ppm precision. Note: You can adjust the default timing register settings for particular communication speeds in the X2CAN file both for PP2CAN and USB2CAN adapter. This option is present in version 2.00 and later. USB2CAN adapter uses the SJA 1000 circuit and 16 MHz crystal.

When you press the **OK** button to close the dialog, the parameters are set and valid only before the application is closed. When using CAN adapter, you must reset the CAN by pressing the **Reset CAN** button. Press the **Save options** button in the main application window to save the parameters for use next time the application is started. The selected communication speed is saved as well.

4.1 CAN ID Dictionary

The CAN message numeric identifier exactly identifies transferred data, however the text message description is more suitable for quick orientation in data. Version 2.003 and higher allows for automatic data identification using identifier dictionary (the ID Dictionary box in the Options dialog). A new Info column appears in the log that contains the message description. Text descriptions are defined in files with extension *.cid (CAN ID Dictionary).

User can edit these text files or create new ones. You can enter the currently used dictionary in the Dictionary file edit box either manually or using the file selection dialog. Press the Edit button in the Options dialog to open the file selection dialog. You can use these files for filtering the messages that are about to be written into the receive log (R log). The predefined messages database file can be used as a message dictionary too (.msg file extension). The structure of CID files is designed so that the simple manual editing and creating would be possible. You can use Notepad to do the editing.

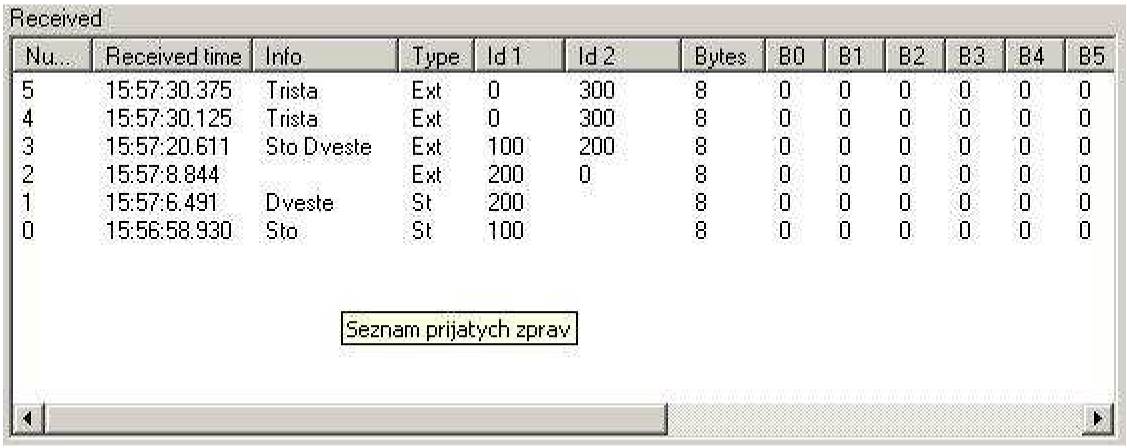
Here is example of the elementary CID file:

```
Comment, normally ignored
>Ext29-100-{1,2}-Extended29,One and two
>Ext29-100-{3,4}-Extended29,Three and four
>Ext29-100-{5}-Extended29,Five

>St-100-Hundred
>St-200-Two hundred

>Ext-100-200-Hundred Two hundred
>Ext29-300-Three hundred

>Ext-100-200-{1,2}-Extended,One and two
>Ext-100-200-{3,4}-Extended,Three and four
>Ext-100-{5}-Extended,Five

>St-100-{1,2}-Standard, One and two
>St-100-{3,4}-Standard, Three and four
>St-100-{5}-Standard, Five
```

The starting character of the line must be a “>“. Any other lines are ignored and can provide for example comments. The standard (11-bit) message is entered using the “St” keyword, extended (11-18-bit) message is entered using the “Ext” keyword. Extended message in 29-bit format is entered using the “Ext29” keyword. The keyword is followed by identifiers (one for St and Ext29, two for Ext) followed by text description. As a separator of the key frames you can use comma, dash or space. You can enter the message identifiers either as a decimal or hexadecimal number. Hexadecimal numbers must have the “0x” prefix. Example: 64 decimal can be entered as 0x40 or 0X40 hexadecimal. Data bytes should be entered in vinculums.

As a further data source for identification you can use predefined message database files (extension *.msg, *.MSG).

When receiving messages with these identifiers, their description will appear in the Info column of the received message log. Extended message with ID 200-0 has no description, because it is not defined in dictionary.



Received

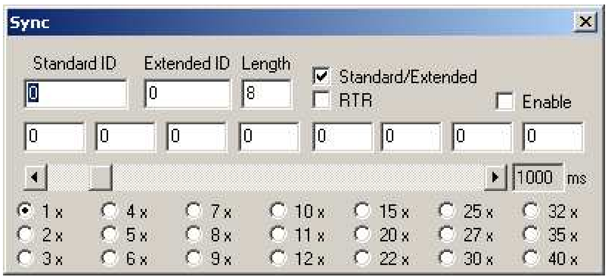
Nu...	Received time	Info	Type	Id 1	Id 2	Bytes	B0	B1	B2	B3	B4	B5
5	15:57:30.375	Trista	Ext	0	300	8	0	0	0	0	0	0
4	15:57:30.125	Trista	Ext	0	300	8	0	0	0	0	0	0
3	15:57:20.611	Sto Dveste	Ext	100	200	8	0	0	0	0	0	0
2	15:57:8.844		Ext	200	0	8	0	0	0	0	0	0
1	15:57:6.491	Dveste	St	200		8	0	0	0	0	0	0
0	15:56:58.930	Sto	St	100		8	0	0	0	0	0	0

Seznam prijatych zprav

Picture 3: Received message log with dictionary identification function set on.

5. Diagnostic tools

5.1 Sync



Sync

Standard ID: 0
Extended ID: 0
Length: 8
☑ Standard/Extended
□ RTR
□ Enable

0 0 0 0 0 0 0 0

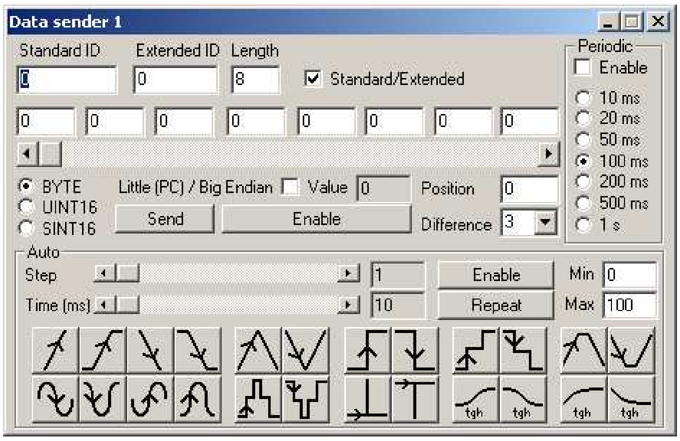
1000 ms

● 1 x ○ 4 x ○ 7 x ○ 10 x ○ 15 x ○ 25 x ○ 32 x
○ 2 x ○ 5 x ○ 8 x ○ 11 x ○ 20 x ○ 27 x ○ 35 x
○ 3 x ○ 6 x ○ 9 x ○ 12 x ○ 22 x ○ 30 x ○ 40 x

Picture 4: Sync window

This tool allows to periodically generate defined CAN message on CAN bus. Use the scrollbar to set the generation period. The Enable checkbox must be set in order to generate a message. The period can range from 10 ms to 10 s. For times shorter than 30 ms the period cannot be guaranteed, since it depends a lot on the actual CPU usage, PC performance and Windows version. It is recommended to use Windows 2000 or XP. The 1, 3, 5 … 40 × radio buttons provide multiple sending after the period has elapsed. They are useful especially for communication usage testing to maximize the bus traffic, as well as for performance tune-up of the application.

5.2 Data sender



Standard ID: 0
Extended ID: 0
Length: 8
☑ Standard/Extended

0 | 0 | 0 | 0 | 0 | 0 | 0 | 0

● BYTE
○ UINT16
○ SINT16

Little (PC) / Big Endian □
Value □ 0
Position: 0
Difference: 3

Periodic
□ Enable
○ 10 ms
○ 20 ms
○ 50 ms
● 100 ms
○ 200 ms
○ 500 ms
○ 1 s

Auto
Step: 1
Time (ms): 10
Min: 0
Max: 100

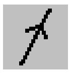
Picture 5: Data sender window.

Data sender is designed to generate data sequence in CAN message and send it on the bus. You can quickly change the data value using mouse, scrollbar or automatically based on selected curve. The data is generated when dragging the scrollbar in manual mode. Set the minimum difference between sent data in Difference combo box. Alternatively, you can make the setting to generate the message periodically. It is then generated periodically within selected period regardless the change occurred or not. You can pick one of three generated data types: BYTE, SIGNED INT16 and UNSIGNED INT16. The data is placed starting data byte specified in the Position box. You can change the order of lower-upper byte (Endian) by checking the Little / Big Endian box.

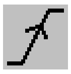
In "automatic sequence" mode the data is generated based on the selected curve and you can specify the increment or the duration of one of possible statuses. Furthermore, you can set the data generation period and select to automatically repeat the curve.

Automatic sequences

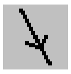
In order to specify the behavior of automatic sequence, you must select the curve and specify the Step, Time, Min and Max parameters. Min and Max parameters specify the minimal and maximal sequence value. It depends on selected data type (BYTE, UINT16, SINT16). Set the Parameter Time within the range 10 - 10000 ms to specify message generating interval and new value calculation. Depending on selected sequence format you can specify the Step parameter to set the value and data incrementing type. In the following list the incrementing calculation is stated for each sequence.



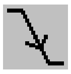
Progressive sequence. Step specifies the value of data sequence increment.



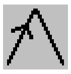
First, Step number of messages is generated with Min value. Next, each step increases the value by (Max-Min)/Step, so that after Step steps the sequence value reaches from starting Min value the Max value. Once the Max value is reached, Step number of messages is being generated with Max value.



Descending sequence. Step specifies the value of data sequence decrement.



First, Step number of messages is generated with Max value. Next, each step increases the value by (Max-Min)/Step. Once the Min value is reached, Step number of messages is being generated with Min value.



Triangular sequence. Data generation is started by incrementing the sequence from the Min value by Step for each step; once the Max value is reached, the sequence is decreased by a Step value until it reaches the Min value.

Triangular sequence. Data generation is started by incrementing the sequence from the Max value by Step for each step; once the Min value is reached, the sequence is increased by a Step value until it reaches the Max value.

First, Step number of messages is generated with Min value and Time interval between messages. Next, the Step number of messages is generated with Max value.

First, Step number of messages is generated with Max value and Time interval between messages. Next, the Step number of messages is generated with Min value.

First, Step number of messages is generated with Min value and Time interval between messages. Next, a Step number of messages is generated with (Max+Min)/2 value (mean value between Min and Max). Finally, the Step number of messages is generated with Max value.

First, Step number of messages is generated with Max value and Time interval between messages. Next, Step number of messages is generated with (Max+Min)/2 value (mean value between Min and Max). Finally, the Step number of messages is generated with Min value.

This sequence has four stages. First, increasing from Min to Max is performed by a (Max-Min)/Step. Next, Step number of messages is being generated with Max value. Next, the decreasing from Max to Min is performed. Finally, the Step number of messages is being generated with Min value.

This sequence has four stages. First, decreasing from Max to Min is performed by a (Max-Min)/Step. Next, Step number of messages is being generated with Min value. Next, the decreasing from Min to Max is performed. Finally, the Step number of messages is being generated with Max value.

The step value represents the angle increment in hundredths of degree. If Step = 100, the angle increment corresponds to one degree. The output value is then calculated this way:
value(t) = ((Max-Min)/2) * (sin(t*Step/100)+1.0)

The step value represents the angle increment in hundredths of degree. If Step = 100, the angle increment corresponds to one degree. The output value is then calculated this way:
value(t) = ((Max-Min)/2) * (cos(t*Step/100)+1.0)

The step value represents the angle increment in hundredths of degree. If Step = 100, the angle increment corresponds to one degree. The output value is then calculated this way:
value(t) = ((Max-Min)/2) * (-sin(t*Step/100)+1.0)

The step value represents the angle increment in hundredths of degree. If Step = 100, the angle increment corresponds to one degree. The output value is then calculated this way:
value(t) = ((Max-Min)/2) * (-cos(t*Step/100)+1.0)

This sequence has five stages. The first and the fifth stage generates Step number of Min values. The second and fourth stage generates Step number of (Max-Min)/2 values. The third stage generates Max value.

This sequence has five stages. The first and the fifth stage generates Step number of Max values. The second and fourth stage generates Step number of (Max-Min)/2 values. The third stage generates Max value.

First, Step values of Min is generated, then one Max value, and finally Step values of Min again.

First, Step values of Max is generated, then one Min value, and finally Step values of Max again.

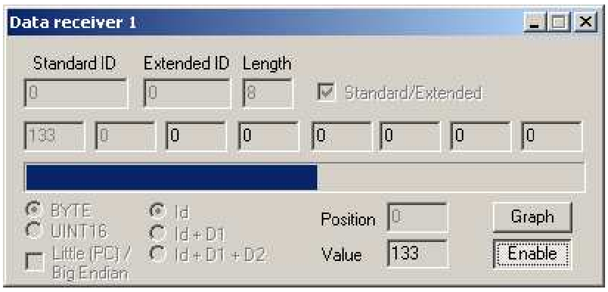
For each step the value is defined this way:
value(t)=((Max-Min)/2) + Max + tanh(t*0.01-(Step/100)) * ((Max-Min)/2);

For each step the value is defined this way:
value(t)=Max - ((Max-Min)/2) - tanh(t*0.01-(Step/100)) * ((Max-Min)/2);

For each step the value is defined this way:
value(t)=Min + tanh(t*0.01-(Step/100)) * ((Max-Min));

For each step the value is defined this way:
value(t)=Max - tanh(t*0.01-(Step/100)) * ((Max-Min));

5.3 Data receiver


Data receiver 1

Standard ID: 0 | Extended ID: 0 | Length: 8 | ☑ Standard/Extended

133 | 0 | 0 | 0 | 0 | 0 | 0 | 0

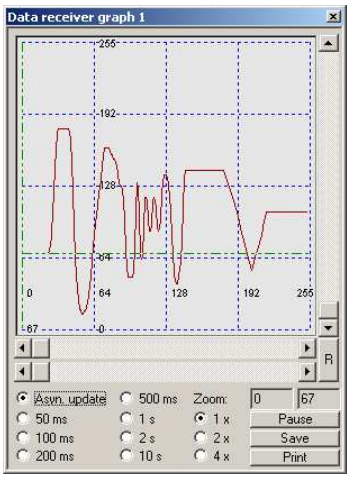
● BYTE ○ UINT16 □ Little (PC) / Big Endian

● Id ○ Id + D1 ○ Id + D1 + D2

Position: 0

Value: 133

Picture 6: Data receiver window.


Data receiver graph 1

● Asyn. update ○ 50 ms ○ 100 ms ○ 200 ms ○ 500 ms ○ 1 s ○ 2 s ○ 10 s

Zoom: ● 1 x ○ 2 x ○ 4 x

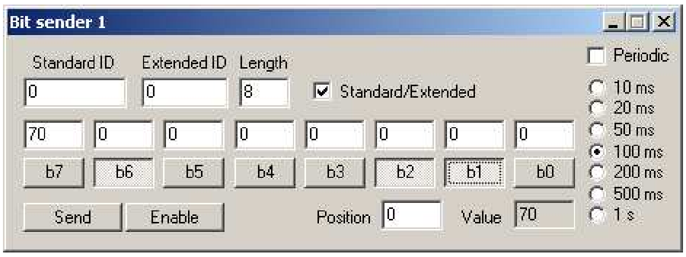
0 | 67

Picture 7: Graph window for values received by Data receiver.

Data receiver allows to picture data of selected message. You can select the message using identifier, identifier + first data byte, or identifier + first and second data byte. The Value box displays a Position (&Position+1) value that is of BYTE or UINT16 type. Press the Graph button to display the graph window that will provide you a synoptic data view.

The graph window displays data received by data receiver. The graph is updated in an asynchronous manner (Asyn. update) with each received message or based on last known data after interval of 50, 100, 200, 500 ms or 1.2 and 10 s. You can control the graph with the scrollbars or display the data values using the cursor that follows the data curve when moving with bottom horizontal scrollbar. You can select between 1.2 and 4 times of the graph scale zoom. Press R to reset the graph control to default settings. Press the Save button to save the graph values as a text file or press the Print button to print the graph. You can disable the graph data receiving using the Pause button.

5.4 Bit sender


Bit sender 1

Standard ID: 0 | Extended ID: 0 | Length: 8 | ☑ Standard/Extended

□ Periodic

○ 10 ms ○ 20 ms ○ 50 ms ● 100 ms ○ 200 ms ○ 500 ms ○ 1 s

70 | 0 | 0 | 0 | 0 | 0 | 0 | 0

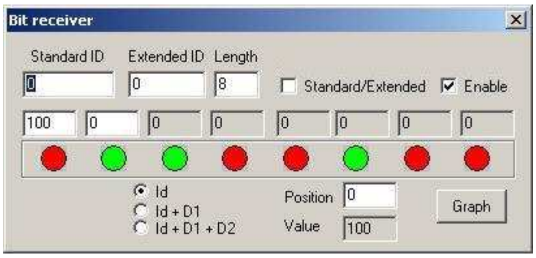
Position: 0

Value: 70

Picture 8: Bit sender window.

Use the Bit sender dialog to generate the messages which require to control the bit statuses in bytes of the message data part. This is useful when controlling the status of digital inputs or in case of keyword sending. Use the b7-b0 buttons to set the bits in data byte selected by a value in the Position box. You can send the message by pressing the Send button or by pressing any of the b0-b7 buttons when the bit status change occured, provided that the Enable box is checked.

5.5 Bit receiver



Bit receiver

Standard ID 0 | Extended ID 0 | Length 8 | ☐ Standard/Extended | ☑ Enable

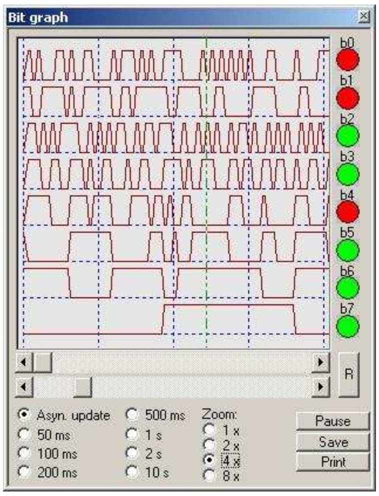
100 | 0 | 0 | 0 | 0 | 0 | 0 | 0

🟥 🟩 🟩 🟥 🟥 🟩 🟥 🟥

● Id
○ Id + D1
○ Id + D1 + D2

Position 0
Value 100

Picture 9: Bit receiver window.



Bit graph

● Asyn. update ○ 50 ms ○ 100 ms ○ 200 ms ○ 500 ms ○ 1 s ○ 2 s ○ 10 s

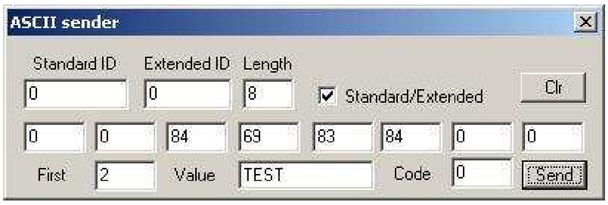
Zoom: ○ 1 x ○ 2 x ● 4 x ○ 8 x

Picture 10: Graph window for values received by Bit receiver.

The Bit receiver dialog allows to receive the data which require to control the bit statuses of the message data part. The data byte bit status is indicated by visual indicating objects; its position is selected in the Position box. The green spot indicates TRUE status of the data bit, red spot represents FALSE status. The b0 bit is displayed on the right. This dialog is useful when analyzing the behavior of CAN peripheral digital inputs and status words from these devices. You can display the graphical representation of the data as well as with the Data receiver.

The function of the Bit sender graph is the same as in case of Data receiver graph. However, it indicates the status of eight bits of the selected byte. Use the bottom scrollbar to move through the graph history; the bit statuses are visually indicated on the right in the place of the cursor. You can print the graph or save the values to a text file.

5.6 ASCII sender



ASCII sender

Standard ID 0 | Extended ID 0 | Length 8 | ☑ Standard/Extended

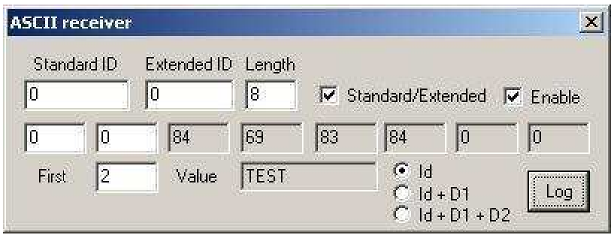
0 | 0 | 84 | 69 | 83 | 84 | 0 | 0

First 2 | Value TEST | Code 0

Picture 11: ASCII sender window.

Use this tool to obtain CAN messages with saved string in its data part. This is particularly useful for devices driven by so called ASCII codes. The string entered in the Value box is being written from the data byte entered in the First box. Unused data bytes after the last text character are filled with the value entered in the Code box.

5.7 ASCII receiver



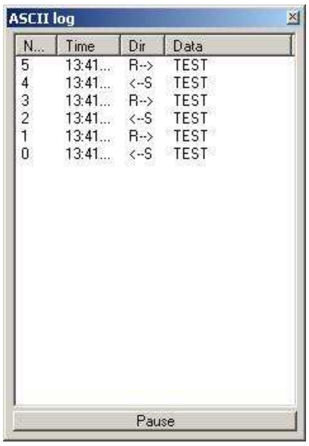
ASCII receiver

Standard ID: 0 | Extended ID: 0 | Length: 8 | ☑ Standard/Extended | ☑ Enable

0 | 0 | 84 | 69 | 83 | 84 | 0 | 0

First: 2 | Value: TEST | ● Id ○ Id + D1 ○ Id + D1 + D2

Picture 12: ASCII receiver window.



ASCII log

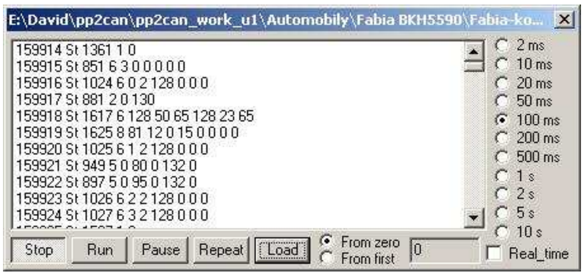
N...	Time	Dir	Data
5	13:41...	R-->	TEST
4	13:41...	<--S	TEST
3	13:41...	R-->	TEST
2	13:41...	<--S	TEST
1	13:41...	R-->	TEST
0	13:41...	<--S	TEST

Picture 13: ASCII log window.

Use the ASCII receiver dialog to monitor the text strings saved in the data part of CAN message. Enter the position of the first byte (character) of the text in the First box.

You can open the ASCII log to view the history and time sequence of sent and received messages. Press the Log button in the ACSII receiver dialog to open the log. The Dir column shows whether the message is received (R-->) or sent (<--S). Press the Pause button to hold the listing.

5.8 File sender



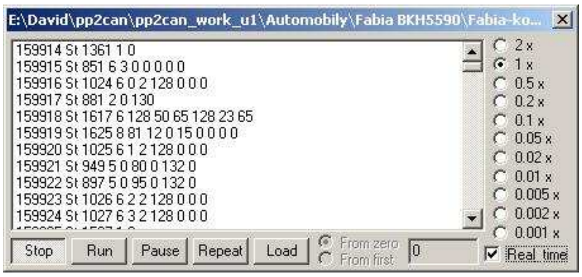
E:\David\pp2can\pp2can_work_u1\Automobily\Fabia BKH5590\Fabia-ko...

```
159914 St 1361 1 0
159915 St 851 6 3 0 0 0 0 0
159916 St 1024 6 0 2 128 0 0 0
159917 St 881 2 0 130
159918 St 1617 6 128 50 65 128 23 65
159919 St 1625 8 81 12 0 15 0 0 0 0
159920 St 1025 6 1 2 128 0 0 0
159921 St 949 5 0 80 0 132 0
159922 St 897 5 0 95 0 132 0
159923 St 1026 6 2 2 128 0 0 0
159924 St 1027 6 3 2 128 0 0 0
```

○ 2 ms ○ 10 ms ○ 20 ms ○ 50 ms ● 100 ms ○ 200 ms ○ 500 ms ○ 1 s ○ 2 s ○ 5 s ○ 10 s

● From zero ○ From first | 0 | □ Real_time

Picture 14: File sender in Fixed period mode.



E:\David\pp2can\pp2can_work_u1\Automobily\Fabia BKH5590\Fabia-ko...

```
159914 St 1361 1 0
159915 St 851 6 3 0 0 0 0 0
159916 St 1024 6 0 2 128 0 0 0
159917 St 881 2 0 130
159918 St 1617 6 128 50 65 128 23 65
159919 St 1625 8 81 12 0 15 0 0 0 0
159920 St 1025 6 1 2 128 0 0 0
159921 St 949 5 0 80 0 132 0
159922 St 897 5 0 95 0 132 0
159923 St 1026 6 2 2 128 0 0 0
159924 St 1027 6 3 2 128 0 0 0
```

○ 2 x ● 1 x ○ 0.5 x ○ 0.2 x ○ 0.1 x ○ 0.05 x ○ 0.02 x ○ 0.01 x ○ 0.005 x ○ 0.002 x ○ 0.001 x

● From zero ○ From first | 0 | ☑ Real_time

Picture 15: File sender in Real time mode.

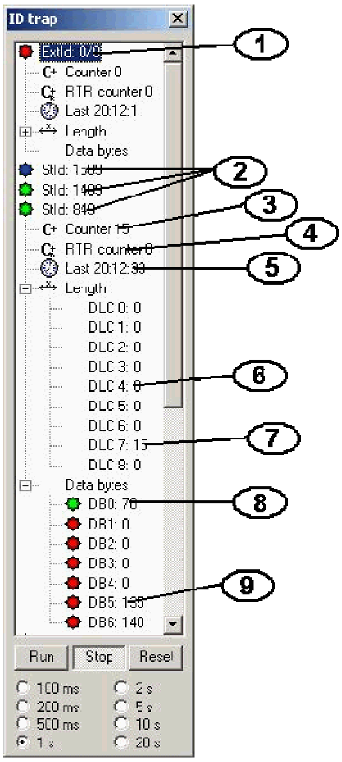
Use this tool to read saved communication logs of the CAN bus and to send it back. You can save these logs by saving the sent/received messages window or using the File logging. You can adjust the logs for example in Excel and export it back to text file. Keep the current file format when exporting the file, i.e. the message is stored on one line and begins with the > character. Other lines are ignored, so you can use them for example for comments. The items are separated with space or tab within the line.

There are two modes of the File sender – Fixed period and Real time. The first is used to consequent message generation based on the index with selected period. If there is more CAN messages with the same index, they are sent in the same period. If an index is not present, there is no message sent in corresponding period.

If you check the Real time box, the messages are being generated depending on the time saved in the log. Because Windows is not a real-time operating system, exact times of sending to CAN bus as with the original communication cannot be guaranteed, especially the same delays between messages. However, the message generation process generally corresponds with the original communication.

If the Repeat button is down, the program jumps to start and begins with sending messages again, after the file is completely sent. Click the From zero to generate the index from zero or click the From first to generate the index from the first log message. The actual index is displayed in bottom right corner of the dialog.

5.9 ID Trap



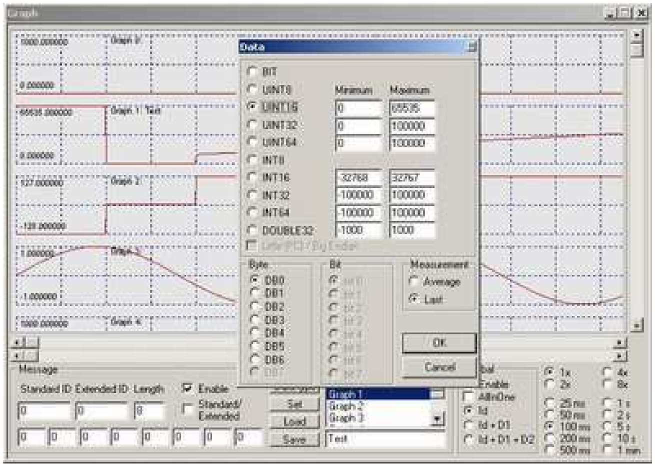
Picture 16: ID trap window.

The ID trap is particularly useful for two purposes. First, you can use it to read out the identifiers of messages present on the bus and count their frequency within set time interval. Second, it can help you to identify asynchronously generated message due to an event or status change.

Press Run to start the measuring; the data in the tree is refreshed based on selected time interval. Once a message is captured that is not present in the list, the identifier of that message is put in the list and the counter is set to one. Next time the message occurs, the counter is increased by one. After the specified time interval passed, all the counters are reset to zero. Messages not present in the just passed interval, i.e. with zero-value of the counter, however present on the bus in previous intervals, are displayed with a red spot. Messages with a non-zero counter value have a green spot. The blue spot stands for an identifier, that has been captured for the first time on the bus. Press the Stop button to stop measuring and evaluate the data. Press the Reset button to clear the message tree; the tree starts to create again next time the measuring starts.

1	Message with extended identifier 0-0. Already captured, however not present in last interval.
2	Messages with standard identifier. The message with blue spot is new – it has been added to the list in the just passed interval. The rest of the messages with green spots have already been captured in previous intervals.
3	The Counter represents the total number of messages with this ID in previous interval. The message with standard ID has been captured 15 times.
4	The RTR Counter represents, how many of them was a RTR type.
5	The Last represents time the message was last captured.
6	The Length subtree represents the number and the length of the messages. Length = the number of data bytes.
7	The message with length of 7 was captured 15 times within last second.
8	Data bytes subtree. DB0 changed the value in the last interval at least once (green spot). Last message contained data byte 0 with a value of 70.
9	DB1-DB6 didn't change the value in the last interval (red spot). DB1-DB4 contained a 0 value in all of the messages. DB5 had a value of 135 and DB6 had a value of 140.

5.10 Graph



Picture 17: Graph window, opened window for data format data putting.

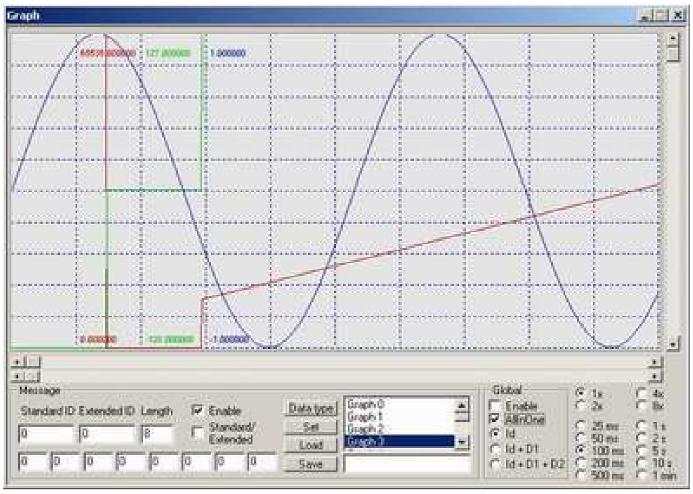
The tool Graph is meant for well arranged data monitoring, which are transmitted by the CAN bus. The tool makes possible on-line monitoring up to 10 quantities of various data types in various CAN messages. For each displayed quantity, there is defined message, which the quantity has occurred in, and furthermore data type and data positions in the message are included. Currently these data types are supported:

- bool
- signed / unsigned int 8
- signed / unsigned int 16
- signed / unsigned int 32
- signed / unsigned int 64
- double 32

Because of some large extent data types it is possible to set limiting interval of the display. It signifies if the quantity type DOUBLE32 and real value of the quantity is within the extent 0.0 – 1000.0, we will set the limiting interval up to this value.

There are stored 1024 samples of each quantity. The sample contains either last registered value in a given time interval or average value during the interval. It is possible to set time interval of the values between 50ms and 1 minute.

Display is carried out in two modes, there has each quantity its own graph in basic mode (see picture 17) and all quantities are displayed on one graph in the mode AllInOne (see picture 18). Configuration of the data setting can be saved in files.

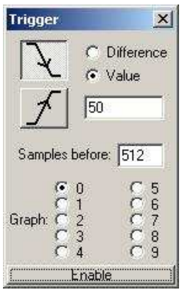


Standard ID: 0
Extended ID: 0
Length: 8
☑ Enable
☐ Standard/Extended
0 0 0 0 0 0 0 0
☐ Enable
☑ AllInOne
● Id
○ Id + D1
○ Id + D1 + D2
● 1x ○ 2x ○ 4x ○ 8x
○ 25 ms ○ 50 ms ● 100 ms ○ 200 ms ○ 500 ms ○ 1 s ○ 2 s ○ 5 s ○ 10 s ○ 1 min

Picture 18: Window of the tool Graph in the mode AllInOne.

There is added export of the graph values to a text file within the version 2.001 (it is possible to import to the program MS Office Excel).

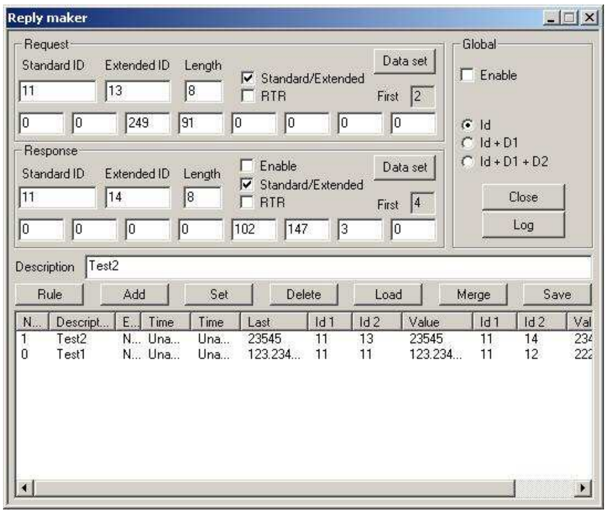
Following the version 2.002 the tool Graph has been supplemented with the function Trigger (see picture 19). Its principle is uniform with similar functions at digital oscilloscopes. The function Trigger permits to stop sampling as the response to an ascending or descending curve of the chosen graph. It is possible to choose curve type, decrease/increase in values between the samples or under/above the level and number of the displayed samples, before the event appearance.



○ Difference
● Value
50
Samples before: 512
Graph: ● 0 ○ 1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6 ○ 7 ○ 8 ○ 9

Picture 19: Tool Graph, Trigger function setting.

5.11 Reply maker



Reply maker

Request
Standard ID 11 Extended ID 13 Length 8
☑ Standard/Extended
□ RTR
First 2
0 0 249 91 0 0 0 0

Response
Standard ID 11 Extended ID 14 Length 8
□ Enable
☑ Standard/Extended
□ RTR
First 4
0 0 0 0 102 147 3 0

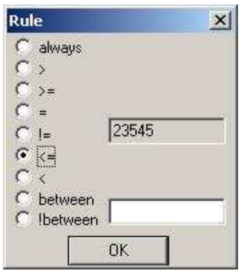
Global
□ Enable
● Id
○ Id + D1
○ Id + D1 + D2

Description Test2

N...	Descript...	E...	Time	Time	Last	Id 1	Id 2	Value	Id 1	Id 2	Val
1	Test2	N...	Una...	Una...	23545	11	13	23545	11	14	234
0	Test1	N...	Una...	Una...	123.234...	11	11	123.234...	11	12	222

Picture 20: Window of the tool Reply maker.

Purpose of the Reply maker function is to process and answer the messages, which are defined with a definite answer on the list. The answer may be determined or it may be chosen on basis of a definite value in the data section of the received message and the rules >,>=,<,<=, = interval. The function is possible to utilize either as simple directing script or for appliance configuration using the system question – answer.
Thus we define message REQUEST, if it is not the RTR type, it is possible to insert value of the types UINT8,16,32,64, INT8,16,32,64 and/or DOUBLE32 into the data section (DATA SET). That value is used as deciding value for assigned rule (RULE) at the same time. Further we define the answer RESPONSE. We also can use data setting by means of the dialog DATA SET. However that value does not affect the rules. Finally we define rule when the answer is generated. The rules are these: always >,>=,<,<=, = in interval and outside it. In the end we add the message into database using the key Add.



Rule

○ always
○ >
○ >=
○ =
○ != 23545
● <=
○ <
○ between
○ !between

Picture 21: Setting of the tool Reply maker rules.

In the event when the rule interval (the rule is fulfilled if there is value lying within interval in the received message) or the rule outside of interval has been used, it is necessary to add the second value of the interval extent. It is carried out together with the rule setting.

We can carry out elementary test so that we start up the program PP2CAN, then we switch on the mode Loopback if we also want to test operation together with adapter or we switch over to the mode V2CAN. We will open Reply maker. We will retain identifiers and data setting on zero values, we only activate Enable at the message Response. Using the key Add we add the message into database. Consequently we allow (Enable) processing in the section Global. In the main window we will generate manually the message together with the identifiers and data on zero level. There is continuously received this message in the log of received messages. In case of opening the graph Bus load, we can see that the same number of messages is sent off and received. What is actually going on? There has been defined the same message during receiving and the same message is defined as an action for sending off, by default the rule is defined always, in the Reply maker database. The sent off message is received back in the mode Loopback (or V2CAN). So the cycle, which receives message and consequently generates the same message as an answer that is returned to it back in the mode Loopback, is created. That is why it generates messages and so it does continuously round. Number of the cycles per second is affected by setting of the item “Receive refresh” in Options.

It is possible to save the database into the file with suffix *.rmk. That file is a text one and it is possible to edit it manually, for example in the program Notepad. The file can be loaded again by means of the command Load. If we have several separate rule files and want them to be loaded together, we can join those files to sooner loaded ones by means of command Merge.

Example:

To test this example, first withdraw an illustrative database for Reply maker from pages www.pp2can.wz.cz or www.usb2can.wz.cz and exemplary database of predefined messages at the same place. Then load that database ReplyMakerExample01.RMK using Load to Reply maker. That database contains 7messages, by means of them you can test function and principles of Reply Maker operation.

Message No. 1 (index 0) has an extended identifier 10 – 10. It contains data of the type DOUBLE32 which are lying from DB0 to DB3 and include value 123.456. The rule (RULE) has been set so that message defined in Response (identifier 10-0) is generated if incoming message has identifier 10-10 and implies on positions DB0 up to DB3 such values which after their conversion to the type DOUBLE32 will do higher value than 123.456. If the value is lower than 123.456, answer Response will not be generated.

Message No. 2 has identifier 20 – 20. Its rule has been set so that the answer Response with ID 20-0 is generated if data on DB0 – DB3 gain value 1000 after their conversion to the type UINT32. In the other cases answer will not be generated.

Message No. 3 has identifier 30 – 30. The answer Response with ID 30-0 will always be generated after receiving of the message with ID 30-0.

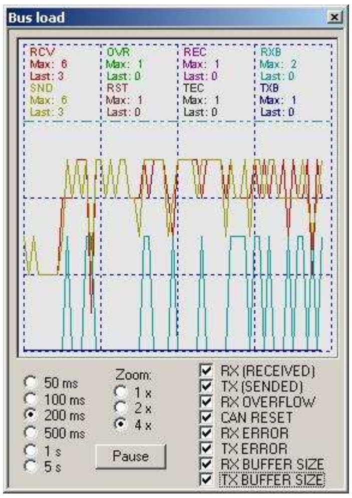
Message No. 4 is RTR type and has standard ID 50. The answer Response is also RTR type with standard ID 51. The answer is always generated. Because RTR messages do not bring any data, it is not possible to define any rule, prospectively any rule is not applied and it is ignored.

Message No. 5 is the same as No. 4. Only the answer has ID 52. The messages No. 4 and No. 5 demonstrate possibility of generation of several messages responding to the same message.

Message No. 6 is RTR type with extended ID 60-60. At its receiving the answer with data and ID 61-61 is generated.

Message No. 7 has standard ID 70. The answer with extended ID 70-70 and data length is generated if incoming message has on DB2 type INT8 with value –1.

5.12 Bus load



RCV
Max: 6
Last: 3
SND
Max: 6
Last: 3

OVR
Max: 1
Last: 0
RST
Max: 1
Last: 0

REC
Max: 1
Last: 0
TEC
Max: 1
Last: 0

RXB
Max: 2
Last: 0
TXB
Max: 1
Last: 0

○ 50 ms
○ 100 ms
● 200 ms
○ 500 ms
○ 1 s
○ 5 s

Zoom:
○ 1 x
○ 2 x
● 4 x

☑ RX (RECEIVED)
☑ TX (SENDED)
☑ RX OVERFLOW
☑ CAN RESET
☑ RX ERROR
☑ TX ERROR
☑ RX BUFFER SIZE
☑ TX BUFFER SIZE

Picture 22: Window of Bus load.

The graph Bus-load displays the following quantities:

- number of received messages during time interval (RX – RCV)

- number of sent off messages during time interval (TX – SND)

- number of messages as for the symptoms RCV OVERFLOW – loss of the message which has not been read pursuant to its rewriting in RX buffer during time interval. That could occur during high loading of the bus.

- RST total number of CAN processor resets during its conversion to BUSoff under the influence of incorrect synchronization or incorrect impedance termination of the bus.

- REC – current value of Receive Error Counter, its reading has to be permitted in Options offer.

- TEC – current value of Transmit Error Counter, its reading has to be permitted in Options offer.

- RXB (RX BUFFER) – extent of software buffer messages, which wait for sending off.

- TXB (TX BUFFER) - software buffer extent of received messages, which wait for processing.

Time axis zoom can be set to extent 1.2 and 4x. The other axis zoom for individual quantities is changed pursuant to maximum reached value, which has been reached within displayed interval. Its value is indicated next to symbol Max. Along with it the value has been highlighted with double dash-and-dot line of relevant colour. The interval, which number of messages (symptoms) has been measured in, is possible to set on values 50, 100, 200, 500 ms a 1.5 s. The measuring may be suspended by means of the key Pause.

5.13 Message filter

The message filter is used for HW filtration of messages, which are received by the adapter. Because the PP2CAN and USB2CAN adapters use different CAN bus control units, setting of the adapter filters is different. It is not possible to set filtration in the mode V2CAN. In addition to HW filtration, there is possible to use SW filtration by means of messages dictionary (CAN ID dictionary). The filtration is activated in Options and is also accessible in the mode V2CAN. That filtration carries out filtration for log of received messages. It does not filter messages for other tools.

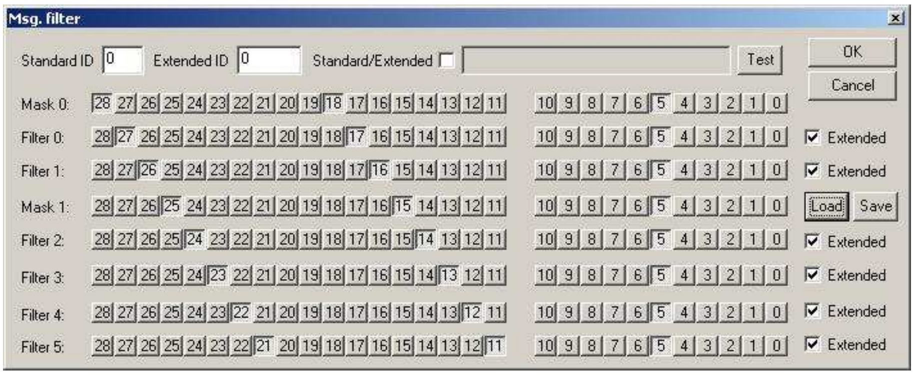
Message filter of the adapter PP2CAN

That dialog is intended for messages filtering according to identifier on HW level by means of filters and message masks. The appliance contains 2 receiving buffers. For the first one it is possible to set a mask and two filters, for the second one a mask and 4 filters. Filtration principle is depicted on the following table:

MaskBit n	FilterBit n	IDbit n	Akcept or reject
0	X	X	Akcept
1	0	0	Akcept
1	0	1	Reject
1	1	0	Reject
1	1	1	Akcept

If the MaskBit has been set on 0, the bit is always accepted. If its value is 1, the identifier message bit is accepted if its value is same as some filter's one. The message is accepted if all the identifier bits have been accepted.

Sense of the filters in this program is to facilitate orientation in volume of messages on bus. If the filters are set so that they accept only needed messages, we decrease load of computer during message processing and particularly reduce possible message volume which may escape. Further we will make then monitoring and data orientation more transparent.



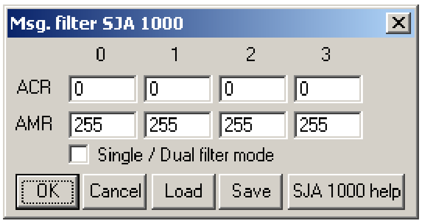
Standard ID 0
Extended ID 0
Standard/Extended □

Filter 0: ☑ Extended
Filter 1: ☑ Extended
Filter 2: ☑ Extended
Filter 3: ☑ Extended
Filter 4: ☑ Extended
Filter 5: ☑ Extended

Picture 23: Setting of message filters for the adapter PP2CAN.

There, on upper window part, is possible to fill out message identifier and to test whether and which of the filters responds to it. The test is activated by depressing of homonymous key, result is displayed on the window left to that key Test.

Message filter of the adapter USB2CAN



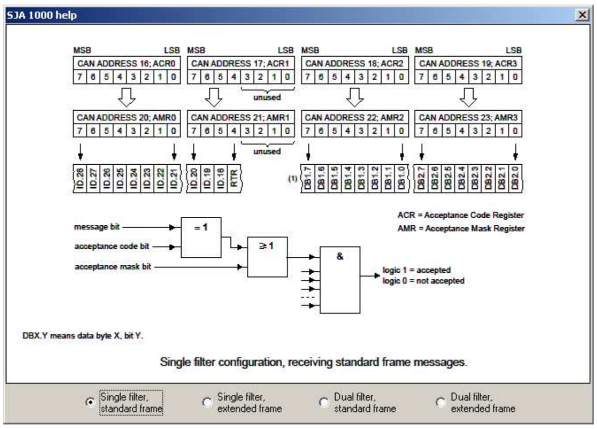
Msg. filter SJA 1000

	0	1	2	3
ACR	0	0	0	0
AMR	255	255	255	255

□ Single / Dual filter mode

Picture 24: Setting of message filters for the adapter USB2CAN.

The adapter USB2CAN use the circuit SJA 1000 as CAN control unit. It is possible to set the message filter registers ACR0-3 and AMR0-3 in that dialog. Filter setting is possible to save into a file and it may be repeatedly loaded. To set filters in individual modes there is image help at your disposal. To acquaint more with filter principles it is convenient to extract the datasheet of the circuit SJA1000. That may be found on pages of the manufacturer, which is Philips, or in section Download on the pages www.usb2can.wz.cz.



SJA 1000 help

ACR = Acceptance Code Register
AMR = Acceptance Mask Register

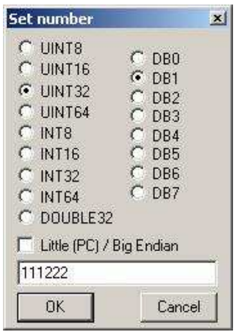
DBX.Y means data byte X, bit Y.

Single filter configuration, receiving standard frame messages.

● Single filter, standard frame
○ Single filter, extended frame
○ Dual filter, standard frame
○ Dual filter, extended frame

Picture 25: Image of help for the filter type “Single filter, standard frame” of the adapter USB2CAN.

5.14 Set numer



Set number

○ UINT8
○ UINT16
● UINT32
○ UINT64
○ INT8
○ INT16
○ INT32
○ INT64
○ DOUBLE32

○ DB0
● DB1
○ DB2
○ DB3
○ DB4
○ DB5
○ DB6
○ DB7

□ Little (PC) / Big Endian

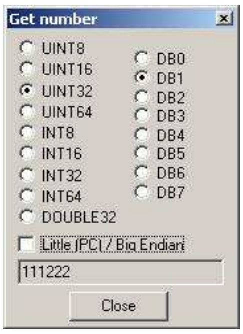
111222

Picture 26: Window of the tool Set number.

The dialog Set number is intended to set data of entered data type into manually sent off message or setting values into Reply maker. The data are set from selected data byte. In default there is used variant Little Endian, that is also used on the platform Intel. Data may be entered even contrariwise (Big Endian) after marking of this option. Difference between Little and Big Endian can be seen at the following example for UINT32. The Byte0 is the lowest byte and the Byte3 is the highest one.

	Little Endian (PC)	Big Endian
Memory (RAM)	Base Address+0 Byte0 Base Address+1 Byte1 Base Address+2 Byte2 Base Address+3 Byte3	Base Address+0 Byte3 Base Address+1 Byte2 Base Address+2 Byte1 Base Address+3 Byte0
CAN message	DBx+0 Byte0 DBx+1 Byte1 DBx+2 Byte2 DBx+3 Byte3	DBx+0 Byte3 DBx+1 Byte2 DBx+2 Byte1 DBx+3 Byte0

5.15 Get numer



Get number

○ UINT8
○ UINT16
● UINT32
○ UINT64
○ INT8
○ INT16
○ INT32
○ INT64
○ DOUBLE32

○ DB0
● DB1
○ DB2
○ DB3
○ DB4
○ DB5
○ DB6
○ DB7

□ Little (PC) / Big Endian

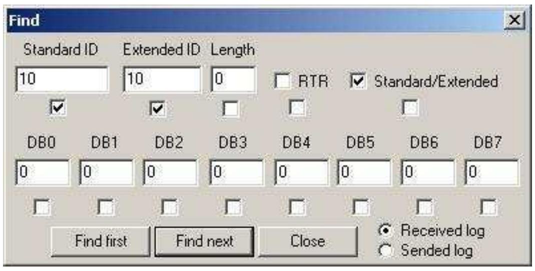
111222

Close

Picture 27: Window of the tool Get number.

Get number is intended to decode numerical values of received logs, sent off messages and predefined messages database. After opening of this window and clicking on the relevant log (list) line the data according to selected types and positions are decoded.

5.16 Find



Find

Standard ID: 10 ☑
Extended ID: 10 ☑
Length: 0 □
□ RTR □
☑ Standard/Extended □

DB0: 0 □
DB1: 0 □
DB2: 0 □
DB3: 0 □
DB4: 0 □
DB5: 0 □
DB6: 0 □
DB7: 0 □

Find first | Find next | Close

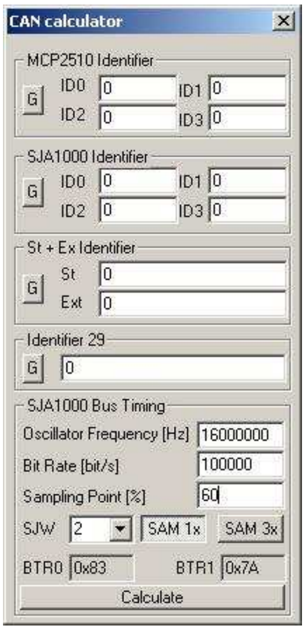
● Received log
○ Sended log

Picture 28: Window of the tool Find for message scanning.

Following the version 2.012 the diagnostic SW PP2CAN contains reworked core of log. It holds good particularly for log of received messages (Receive log) that it may contain tens even hundreds thousands of messages. To facilitate orientation in data registered to this log there, after version 2.013, the function Find has been implemented. That one is intended to seek messages according to entered data. Check box under each one of the items states whether that item-value has been used in the seeking criterion. It is possible to seek on basis of arbitrary valid items combination which creates CAN message. It is possible to carry out seeking in log received or sent off messages.. The log window is set for found item, which is also highlighted.

If seeking has been set according to the shown picture, the messages with extended identifier 10-10 or standard identifier 10 are successively sought after, in the process it does not matter whether message RTR or data message is concerned and the same holds good for prospective data.

5.17 CAN calculator



MCP2510 Identifier: ID0 0, ID1 0, ID2 0, ID3 0
SJA1000 Identifier: ID0 0, ID1 0, ID2 0, ID3 0
St + Ex Identifier: St 0, Ext 0
Identifier 29: 0
Oscillator Frequency [Hz] 16000000
Bit Rate [bit/s] 100000
Sampling Point [%] 60
SJW 2
BTR0 0x83
BTR1 0x7A

Picture 29: The tool CAN calculator.

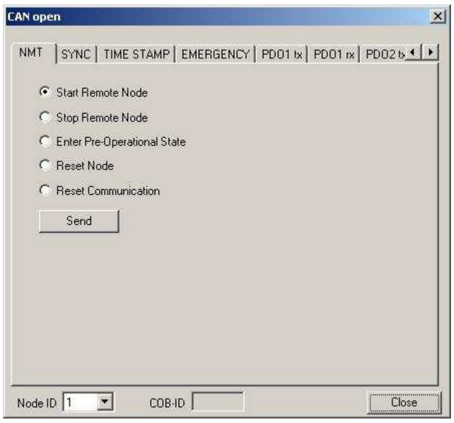
The CAN calculator is an instrument meant for conversion particularly for extended CAN identifiers. It makes possible to carry out conversion among identifiers of the formats:

- 29 bit identifier
- identifier of standard format (11 bit) + extended (18 bit) section
- saved in 8-bit registers of the circuit SJA1000

Further it contains the tool for computation of optimal Timing registers setting (baud rate & point of sampling) BTR0 and BTR1 for CAN bus control units SJA 1000 and 82C200. After assignment of oscillator frequency, desired communication velocity, point of sampling that will compute optimal setting of the registers BTR0 and BTR1.

5.18 CANopen

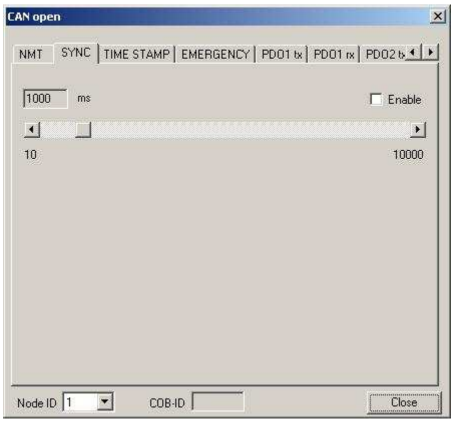
This dialog contains tools for facilitation of working with high-level protocol CANopen. The tool has been under development, for the present it contains several basic functions.



● Start Remote Node
○ Stop Remote Node
○ Enter Pre-Operational State
○ Reset Node
○ Reset Communication

Node ID 1
COB-ID

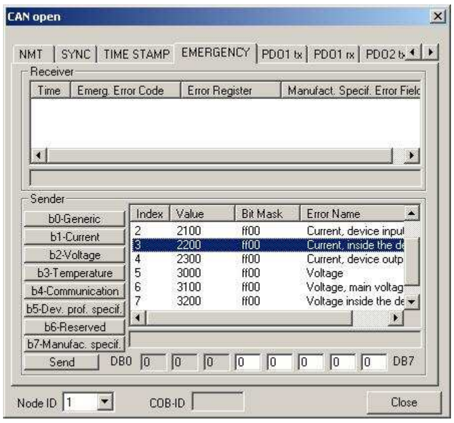
Picture 30: Tool CANopen, bookmark NMT.



1000 ms
□ Enable

Node ID 1
COB-ID

Picture 31: Tool CANopen, bookmark SYNC.



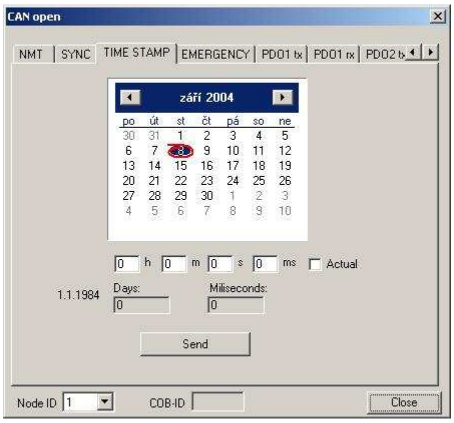
Time	Emerg. Error Code	Error Register	Manufact. Specif. Error Field

Index	Value	Bit Mask	Error Name
2	2100	ff00	Current, device input
3	2200	ff00	Current, inside the de
4	2300	ff00	Current, device outp
5	3000	ff00	Voltage
6	3100	ff00	Voltage, main voltag
7	3200	ff00	Voltage inside the de

DB0 0 0 0 0 0 0 0 0 DB7

Node ID 1 COB-ID

Picture 32: Tool CANopen, bookmark EMERGENCY.



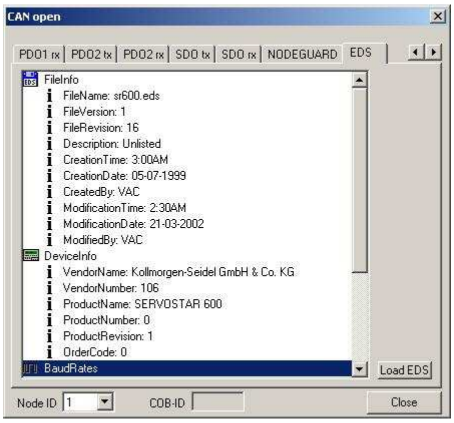
po	út	st	čt	pá	so	ne
30	31	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	1	2	3
4	5	6	7	8	9	10

0 h 0 m 0 s 0 ms ☐ Actual

1.1.1984 Days: 0 Miliseconds: 0

Node ID 1 COB-ID

Picture 33: Tool CANopen, bookmark TIME STAMP.



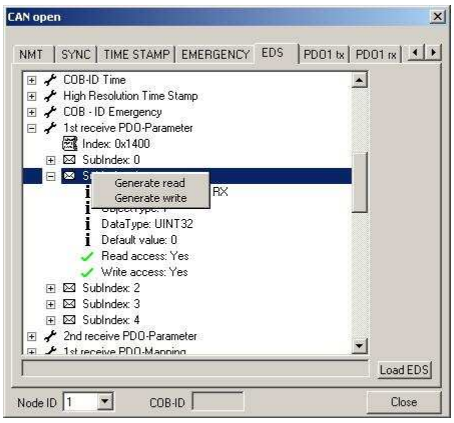
FileInfo
- FileName: sr600.eds
- FileVersion: 1
- FileRevision: 16
- Description: Unlisted
- CreationTime: 3:00AM
- CreationDate: 05-07-1999
- CreatedBy: VAC
- ModificationTime: 2:30AM
- ModificationDate: 21-03-2002
- ModifiedBy: VAC

DeviceInfo
- VendorName: Kollmorgen-Seidel GmbH & Co. KG
- VendorNumber: 106
- ProductName: SERVOSTAR 600
- ProductNumber: 0
- ProductRevision: 1
- OrderCode: 0

BaudRates

Node ID 1 COB-ID

Picture 34: Tool CANopen, bookmark EDS.



- COB-ID Time
- High Resolution Time Stamp
- COB - ID Emergency
- 1st receive PDO-Parameter
  - Index: 0x1400
  - SubIndex: 0
  - S
    - RX
    - DataType: UINT32
    - Default value: 0
    - Read access: Yes
    - Write access: Yes
  - SubIndex: 2
  - SubIndex: 3
  - SubIndex: 4
- 2nd receive PDO-Parameter
- 1st receive PDO-Mapping

Node ID 1 COB-ID

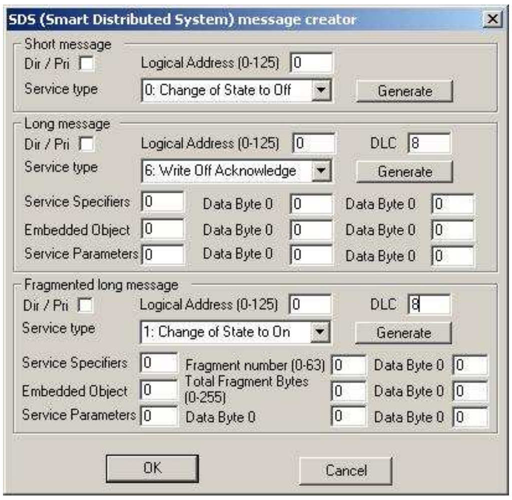
Picture 35: Tool CANopen, bookmark EDS, tree of objects.

The file EDS (Electronic Data Sheet) is configuration file for parameters and communication profiles setting of the CANopen unit. Those files are delivered by manufacturers of these CANopen units.

There is possible to load the file EDS for CANopen units at the bookmark EDS. In addition to information and supported objects image, it is able to generate CAN message into the field for message manual sending off according to selected object (index, subindex). It is possible to generate message for data setting (entry) even for data reading in the unit.

Some files are very extended, for example the file for servo amplifier Servostar 600 contains more than 300kB. That file contains over 20000 text lines. Parsing of that text file may take even several tens seconds.

5.19 SDS ID



SDS (Smart Distributed System) message creator

Short message
Dir / Pri □
Logical Address (0-125): 0
Service type: 0: Change of State to Off

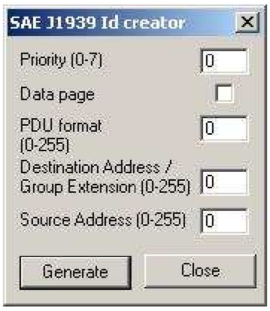
Long message
Dir / Pri □
Logical Address (0-125): 0
DLC: 8
Service type: 6: Write Off Acknowledge
Service Specifiers: 0
Embedded Object: 0
Service Parameters: 0
Data Byte 0: 0 (×6)

Fragmented long message
Dir / Pri □
Logical Address (0-125): 0
DLC: 8
Service type: 1: Change of State to On
Service Specifiers: 0
Embedded Object: 0
Service Parameters: 0
Fragment number (0-63): 0
Total Fragment Bytes (0-255): 0
Data Byte 0: 0 (×4)

Picture 36: Tool SDS ID creator.

The tool is intended for generation of high-level protocol SDS identifier. The identifier and some data bytes are set pursuant to entered data into the field for manual generation of message.

5.20 SAE J1939 ID



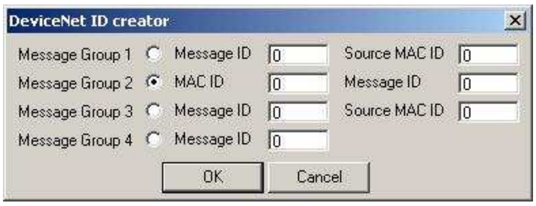
SAE J1939 Id creator

Priority (0-7): 0
Data page □
PDU format (0-255): 0
Destination Address / Group Extension (0-255): 0
Source Address (0-255): 0

Picture 37: Tool SAE J1939 ID creator.

The tool is intended for generation of high-level protocol SAE J1939 identifier. The identifier is set pursuant to entered data into the field for manual generation of message.

5.21 DeviceNet ID



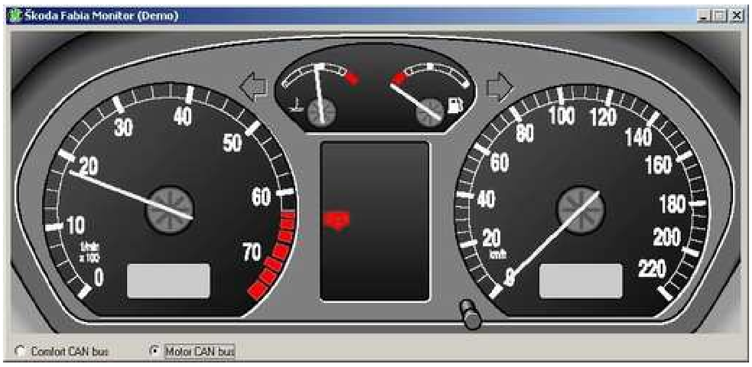
DeviceNet ID creator

Message Group 1 ○ Message ID: 0 Source MAC ID: 0
Message Group 2 ● MAC ID: 0 Message ID: 0
Message Group 3 ○ Message ID: 0 Source MAC ID: 0
Message Group 4 ○ Message ID: 0

Picture 38: Tool DeviceNet ID creator.

The tool is intended for generation of high-level protocol DeviceNet identifier. The identifier is set pursuant to entered data into the field for manual generation of message.

5.22 Škoda Fabia Monitor

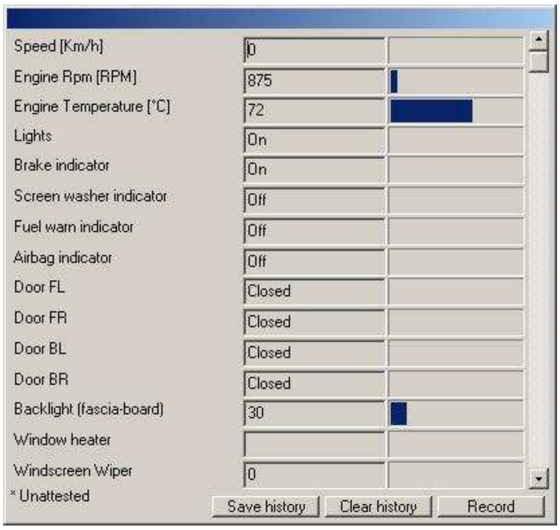


Škoda Fabia Monitor (Demo)

○ Comfort CAN bus ● Motor CAN bus

Picture 39: Main window of the tool Škoda Fabia Monitor.

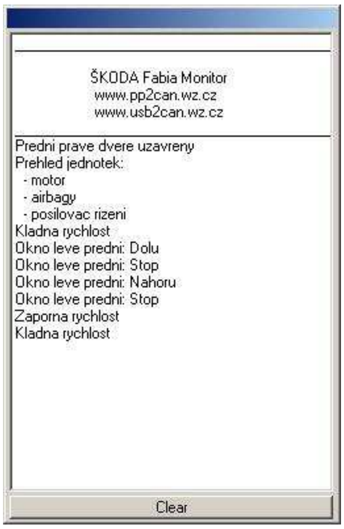
The tool is intended for data monitoring on CAN bus of the car Škoda Fabia. It is designed for data monitoring on CAN drive and CAN comfort of that car. It enables to store data time history and generates a file for their display by means of the program GNUplot.



Speed [Km/h]	0
Engine Rpm [RPM]	875
Engine Temperature [°C]	72
Lights	On
Brake indicator	On
Screen washer indicator	Off
Fuel warn indicator	Off
Airbag indicator	Off
Door FL	Closed
Door FR	Closed
Door BL	Closed
Door BR	Closed
Backlight (fascia-board)	30
Window heater	
Windscreen Wiper	0

* Unattested

Picture 40: Window display of Škoda Fabia monitored quantities.



ŠKODA Fabia Monitor
www.pp2can.wz.cz
www.usb2can.wz.cz

```
Predni prave dvere uzavreny
Prehled jednotek:
 - motor
 - airbagy
 - posilovac rizeni
Kladna rychlost
Okno leve predni: Dolu
Okno leve predni: Stop
Okno leve predni: Nahoru
Okno leve predni: Stop
Zaporna rychlost
Kladna rychlost
```

Picture 41: Window of the Škoda Fabia Monitor log.

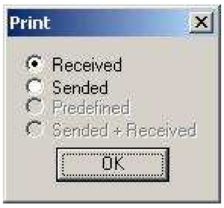
5.23 File logging

This function enables to carry out logging of communication on bus CAN into file. After its switching on there does not occur any further processing of messages. So load of processor is decreasing and, at simple adapter PP2CAN, quantity of uncaptured messages is lowered. That simple adapter is able to send off and receive at the same time no more than circa 650 messages per second in both directions. Quantity of lost messages depends on communication character particularly on messages spacing. The adapter USB2CAN is able to process severalfold higher data quantity!

5.24 Save log

By clicking on this key it is possible to save log contents of received or sent off messages into file. File data format is congruent with the log which has been provided with the function File logging. Those files may be loaded into the tool File sender and consecutively off-line data analysis may be done by means of diagnostic SW PP2CAN. Since text files are concerned, it is possible to import data even, for example, into the program EXCEL which provide a lot of other tools for analysis of data sequences.

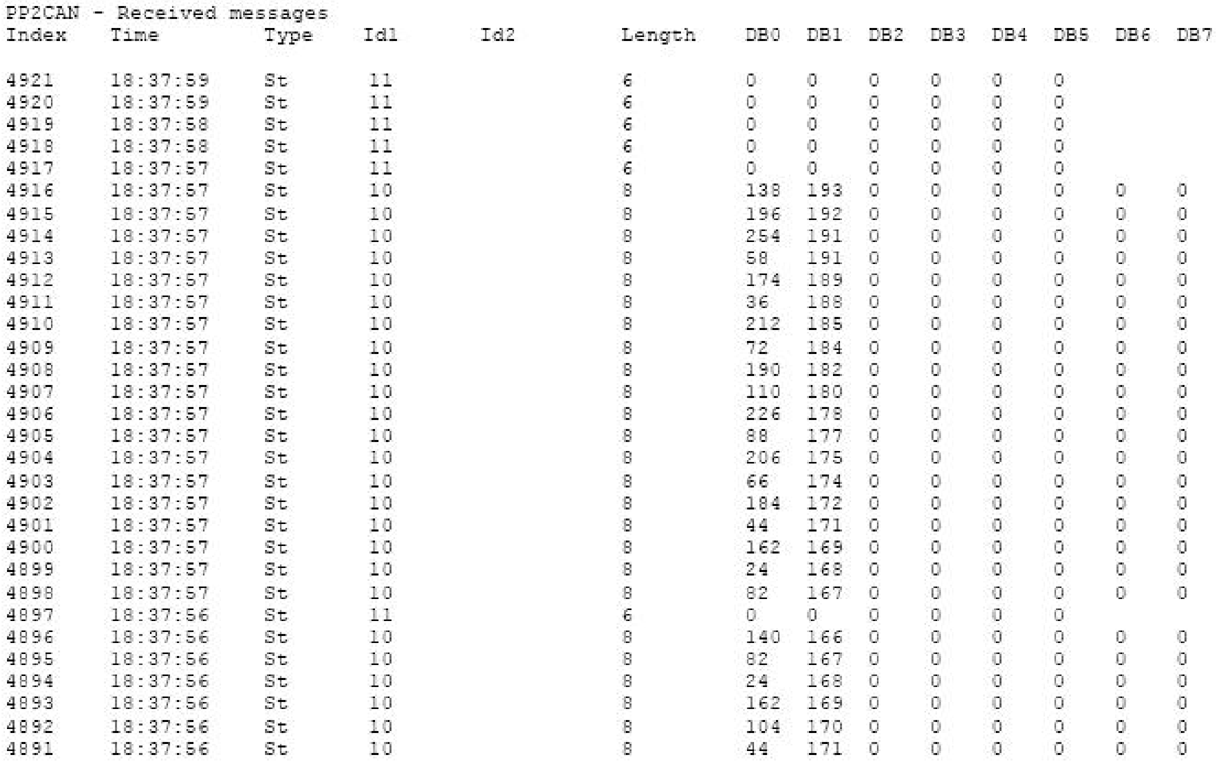
5.25 Print



● Received
○ Sended
○ Predefined
○ Sended + Received

Picture 42: Dialog of data selection for printing.

Nowadays it is possible to carry out printing of received and sent off messages log. Data format after printing is on the following picture.



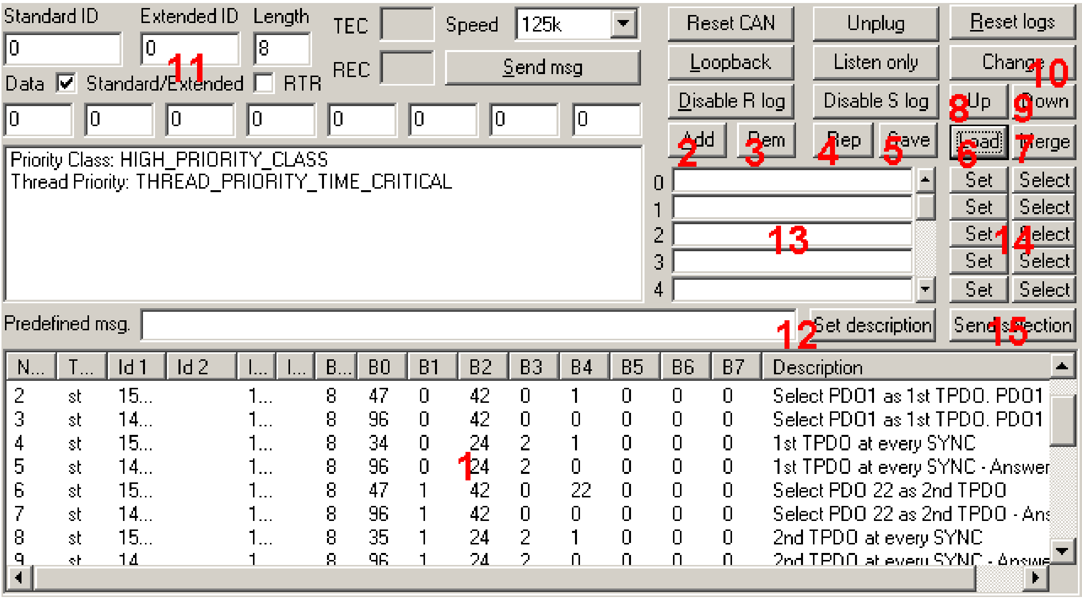
```
PP2CAN - Received messages
Index    Time         Type    Id1       Id2         Length    DB0  DB1  DB2  DB3  DB4  DB5  DB6  DB7

4921     18:37:59     St      11                    6         0    0    0    0    0    0
4920     18:37:59     St      11                    6         0    0    0    0    0    0
4919     18:37:58     St      11                    6         0    0    0    0    0    0
4918     18:37:58     St      11                    6         0    0    0    0    0    0
4917     18:37:57     St      11                    6         0    0    0    0    0    0
4916     18:37:57     St      10                    8         138  193  0    0    0    0    0    0
4915     18:37:57     St      10                    8         196  192  0    0    0    0    0    0
4914     18:37:57     St      10                    8         254  191  0    0    0    0    0    0
4913     18:37:57     St      10                    8         58   191  0    0    0    0    0    0
4912     18:37:57     St      10                    8         174  189  0    0    0    0    0    0
4911     18:37:57     St      10                    8         36   188  0    0    0    0    0    0
4910     18:37:57     St      10                    8         212  185  0    0    0    0    0    0
4909     18:37:57     St      10                    8         72   184  0    0    0    0    0    0
4908     18:37:57     St      10                    8         190  182  0    0    0    0    0    0
4907     18:37:57     St      10                    8         110  180  0    0    0    0    0    0
4906     18:37:57     St      10                    8         226  178  0    0    0    0    0    0
4905     18:37:57     St      10                    8         88   177  0    0    0    0    0    0
4904     18:37:57     St      10                    8         206  175  0    0    0    0    0    0
4903     18:37:57     St      10                    8         66   174  0    0    0    0    0    0
4902     18:37:57     St      10                    8         184  172  0    0    0    0    0    0
4901     18:37:57     St      10                    8         44   171  0    0    0    0    0    0
4900     18:37:57     St      10                    8         162  169  0    0    0    0    0    0
4899     18:37:57     St      10                    8         24   168  0    0    0    0    0    0
4898     18:37:57     St      10                    8         82   167  0    0    0    0    0    0
4897     18:37:56     St      11                    6         0    0    0    0    0    0
4896     18:37:56     St      10                    8         140  166  0    0    0    0    0    0
4895     18:37:56     St      10                    8         82   167  0    0    0    0    0    0
4894     18:37:56     St      10                    8         24   168  0    0    0    0    0    0
4893     18:37:56     St      10                    8         162  169  0    0    0    0    0    0
4892     18:37:56     St      10                    8         104  170  0    0    0    0    0    0
4891     18:37:56     St      10                    8         44   171  0    0    0    0    0    0
```

Picture 43: Example of data format at printing.

6. The other options

6.1. Database of predefined messages

Frequently used messages may be stored into the database of predefined messages. Further we may use that database as source for automatic identification of messages (CAN ID Dictionary) which is set in Options.



Standard ID 0 Extended ID 0 Length 8

Speed 125k

Data ☑ Standard/Extended ☐ RTR

0 0 0 0 0 0 0 0

Priority Class: HIGH_PRIORITY_CLASS
Thread Priority: THREAD_PRIORITY_TIME_CRITICAL

0
1
2
3
4

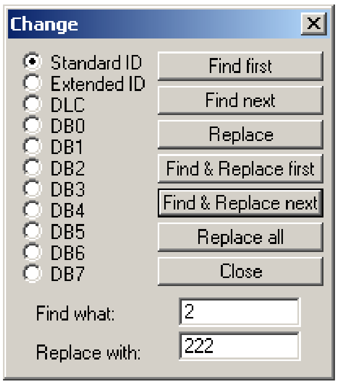
Predefined msg.

N...	T...	Id 1	Id 2	I...	I...	B...	B0	B1	B2	B3	B4	B5	B6	B7	Description
2	st	15...		1...		8	47	0	42	0	1	0	0	0	Select PDO1 as 1st TPDO. PDO1
3	st	14...		1...		8	96	0	42	0	0	0	0	0	Select PDO1 as 1st TPDO. PDO1
4	st	15...		1...		8	34	0	24	2	1	0	0	0	1st TPDO at every SYNC
5	st	14...		1...		8	96	0	24	2	0	0	0	0	1st TPDO at every SYNC - Answer
6	st	15...		1...		8	47	1	42	0	22	0	0	0	Select PDO 22 as 2nd TPDO
7	st	14...		1...		8	96	1	42	0	0	0	0	0	Select PDO 22 as 2nd TPDO - Ans
8	st	15...		1...		8	35	1	24	2	1	0	0	0	2nd TPDO at every SYNC
9	st	14...		1...		8	96	1	24	2	0	0	0	0	2nd TPDO at every SYNC - Answe

Picture 44: Working with the database of predefined messages.

1	Window with list of predefined messages. Double-click of the left mouse key will overwrite selected message into the field for manual entry of message (11).
2	Subjoining of the message, which has been filled in the field 11, into database. Description is added subsequently by double-click on the subjoined message, further its entry into the field 12 is carried out by depressing Set description.
3	Deletion of database selected message.
4	Replace, message from field for manual message entry will replace selected message in the window of predefined messages database
5	Saving of predefined message database.
6	Loading of predefined message database.
7	Selected database is loaded and joined to currently loaded database.
8	Sliding of selected message by a position above.
9	Sliding of selected message by a position down.
10	Tool for mass data changes in a database. For example it is possible to change value of identifier in predefined message window for another one.
11	Field for message manual entry and sending off.
12	Editing field for description entry and key of its setting for selected database message.
13	List of message selection. There we may define message cluster in a database that cluster is sent off by depressing of the key Send selection (15). That characteristic will find practical use in cases when we need to send en masse more messages onto bus, which will carry out device initialisation, for example.
14	Keys for selection setting and their selection. Using the key Set the messages, which are marked in the window of predefined message database, are saved into selection. There is carried out selection of predefined messages using Ctrl + mouse clicking for single message selection and Shift + mouse clicking for message block selection in the window of predefined messages. Text on relevant line in selection list will be used for marking of selection. The key Select will carry out marking of message selection.
15	Sending off of selected messages to CAN. Marked messages sending off will occur, i.e. we may carry out message cluster selection and before their sending off to carry out additional amendments, i.e. marking/unmarking.

Function Change



● Standard ID
○ Extended ID
○ DLC
○ DB0
○ DB1
○ DB2
○ DB3
○ DB4
○ DB5
○ DB6
○ DB7

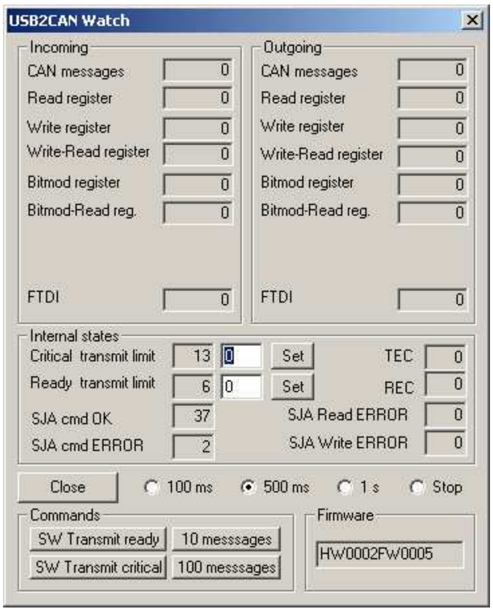
Find what: 2

Replace with: 222

Picture 45: Function Change.

Use of predefined messages database is very practical, however provided that we put together such a database for some device we will also want to use that database even for same device of different address. Together with this it does not matter where in that message that address lies. Just we need change in messages for example Id1 = 2 on 222. That will do only to open the window Change and click on homonymous key in left central part of main window. It will do then in this window, which value will be replaced, which value will be replaced with and where it lies. There are functions not only for retrieval itself but also for retrieval provided with automatic substitution.

6.2. USB2CAN Watch



Incoming		Outgoing	
CAN messages	0	CAN messages	0
Read register	0	Read register	0
Write register	0	Write register	0
Write-Read register	0	Write-Read register	0
Bitmod register	0	Bitmod register	0
Bitmod-Read reg.	0	Bitmod-Read reg.	0
FTDI	0	FTDI	0

Internal states

Critical transmit limit	13	0	TEC	0
Ready transmit limit	6	0	REC	0
SJA cmd OK	37		SJA Read ERROR	0
SJA cmd ERROR	2		SJA Write ERROR	0

○ 100 ms ● 500 ms ○ 1 s ○ Stop

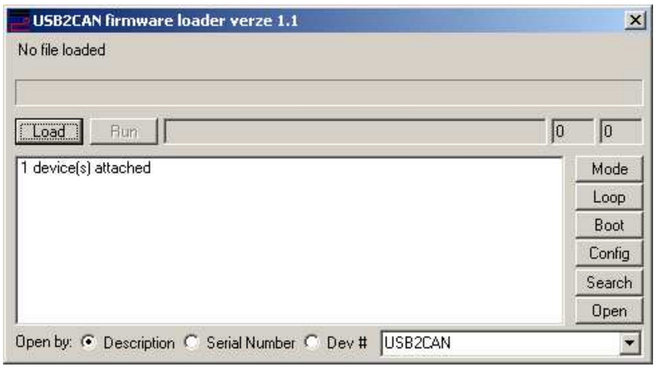
Firmware: HW0002FW0005

Picture 46: Window of the tool USB2CAN Watch.

The tool USB2CAN makes possible to monitor inner states of the USB2CAN adapter and API interface of this adapter.

6.3. USB2CAN firmware loader

In order to ensure possibility of easy firmware converter updating with new prospectively user application versions, USB2CAN contains bootloader which ensures this possibility. Firmware change takes place by force of specialized application in PC, using USB.



USB2CAN firmware loader verze 1.1

No file loaded

1 device(s) attached

Open by: ● Description ○ Serial Number ○ Dev # USB2CAN

Picture 47: Program window for firmware CAN bus adapter USB2CAN updating.

USB2CAN has 3 basic operating modes available, BOOT MODE, CONFIG MODE NORMAL MODE, furthermore there exists supplementary mode LOOPBACK MODE. After connecting to USB the unit appears in the mode BOOT, there is possible to carry out firmware change in this mode. It is only possible to switch over from this mode to the mode CONFIG. That mode has already been just as further modes under direction of firmware itself. There are identically entitled keys meant for switching over between modes BOOT and CONFIG on upper right program window part. The key MODE will send off inquiry about current mode. That is successively written in logging window of the program. There can be seen that device was in mode CONFIG then it was switched over to mode BOOT and successively switched over to mode CONFIG back, on this window.

In case that only one adapter USB2CAN has been connected, automatically communication is established with this adapter. If there are connected several adapters at the same time, it is possible to carry out their seeking pursuant to chosen criterion (Device description, Serial number, Device number) by means of the key Search. Opening of selected adapter is carried out by the key Open. Potential errors are written out on the program logging window.

The key Load is meant for loading of firmware file. Path to opened file is displayed on upper window part. Starting of firmware update process will take place by means of the key Run. Termination message or possible errors are again displayed on the logging window.

7. Tipy

7.1. Klávesové zkratky

Help	F1
On-line help	Alt + F1
Sync	F2
Data sender	F3
Data sender 2	Alt + F3
Data receiver	F4
Data receiver 2	Alt + F4
Bit sender	F5
Bit sender 2	Alt + F5
Bit receiver	F6
Bit receiver 2	Alt + F6
ASCII sender	F7
ASCII receiver	F8
CAN open	F9
ID Trap	F10
Reply maker	F11
File sender	F12
Saving of predefined message database	Ctrl +Alt + S
Loading of predefined message database	Ctrl +Alt + L
Addition of next predefined message database	Ctrl +Alt + M
Message manual sending off	S
Send selection	Alt + M
Addition of message into a database	A
Get number	Alt + G
Loopback mode	L
Disable R log	D
Bus load	Alt + B
Program closing	ESC

List of pictures: