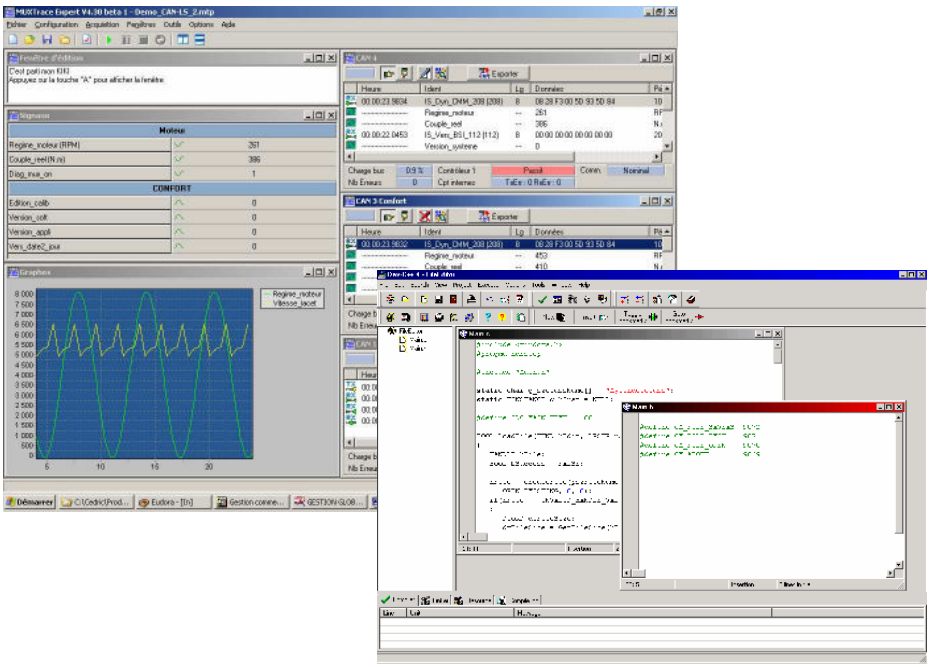
MUX TRACE
MUX TRACE / Expert



USER GUIDE

Document n. 054578-10 *Published 7/8/2005*

ANNECY ELECTRONIQUE Z.A. Les Marais 74410 St JORIOZ
Phone : +33 (0) 450 68 90 65 Fax: **+33 (0) 450** 68 58 93

INDEX

1 Aim of this document

1.1 Aim of this document

The aim of this document is to give the required information for using the MuxTrace programme, which allows the user to control communication channels CAN, CAN fault tolerant, LIN, VAN, ISO9141, Diagnostics On CAN and NMEA0183 and to visualize the state of bus errors thanks to a user friendly graphic interface.

The available functions are:

- Simultaneous management of multi channel and multi protocols.
- Independent channel configuration and triggering.
- Setting *Transmission* function (Period, transmission conditions with key)
- Setting *Reception* function (Acceptance filter, spy mode)
- Permanent indication of frame characteristics and their content in hexadecimal
- Setting acquisition modes
- Recording of measurements configuration

This program is available with all EXXOTEST's network access boards.

1.2 Reference documents

ISO11898 : Road vehicles -- Interchange of digital information -- Controller area network (CAN) for high-speed communication

ISO 11519-2 : Road vehicles -- Low-speed serial data communication -- Part 2: Low-speed controller area network (CAN)

LIN V1.2 : Specifications package

ISO 11519-3 Road vehicles -- Low-speed serial data communication -- Part 3: Vehicle area network (VAN)

ISO 9141 : Road vehicles - Diagnostic System – Characteristics of the numerical data exchange

ISO 9141-2 : Road vehicles - Diagnostic System – Characteristics CARB of the numerical data exchange

ISO 14230 - 1 Physical layer (Keyword Protocol 2000 - Part 1).

ISO 14230 - 2 Data Link layer (Keyword Protocol 2000 - Part 2).

ISO 14230 - 3 Application layer (Keyword Protocol 2000 - Part 3).

ISO 15765-1 Road vehicles – diagnostics on CAN – Part 1 : General information

ISO 15765-2 Road vehicles – diagnostics on CAN – Part 2 : Network layer services

ISO 15765-3 Road vehicles – diagnostics on CAN – Part 2 : Application layer

ISO 15765-4 Road vehicles – diagnostics on CAN – Part 4 : Requirements for emission related systems

OSEK/VDX Network management V 2.5

2 Software installation

2.1 Minimum configuration requirements

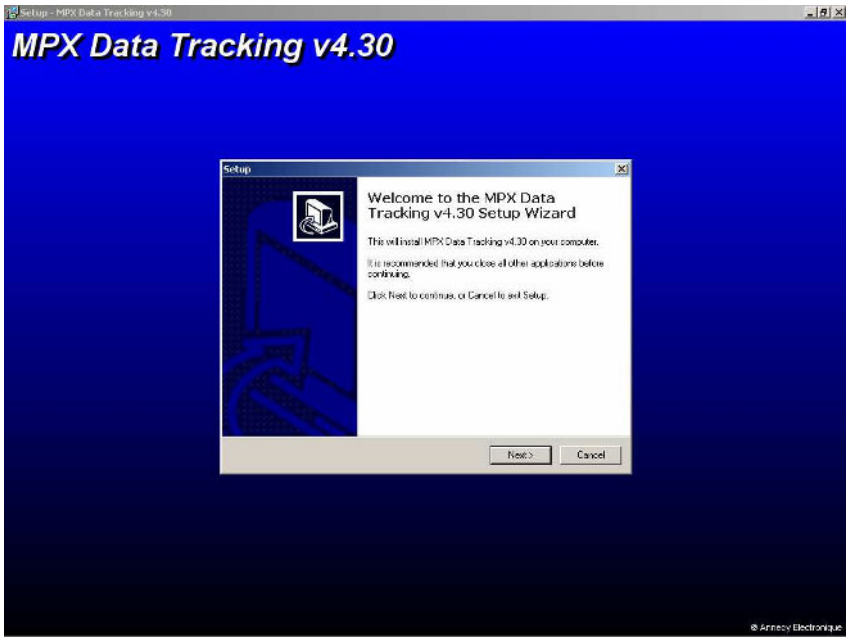
Operating system
- Windows 95
- Windows 98
- Windows NT
- Windows 2000/Me/XP

The minimum recommended configuration is the following:
- PC type computer equipped with a Pentium microprocessor (PIII 600 or over recommended) with CD ROM reader

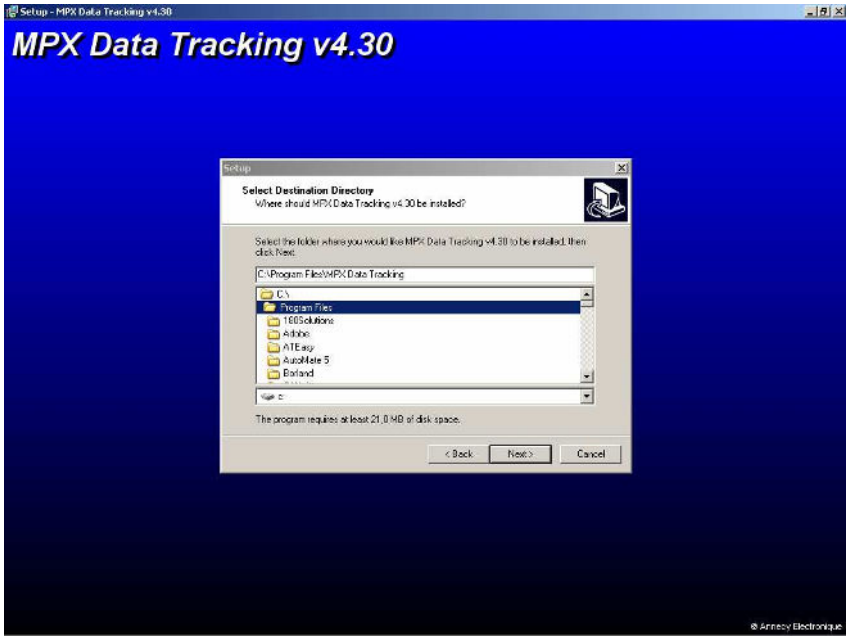
The performance of the MuxTrace program depends on type of PC used and it might be altered depending on the computer configuration
- Another application working simultaneously with the MuxTrace program
- Screen saver
- Antivirus program
- ….

2.2 Installation

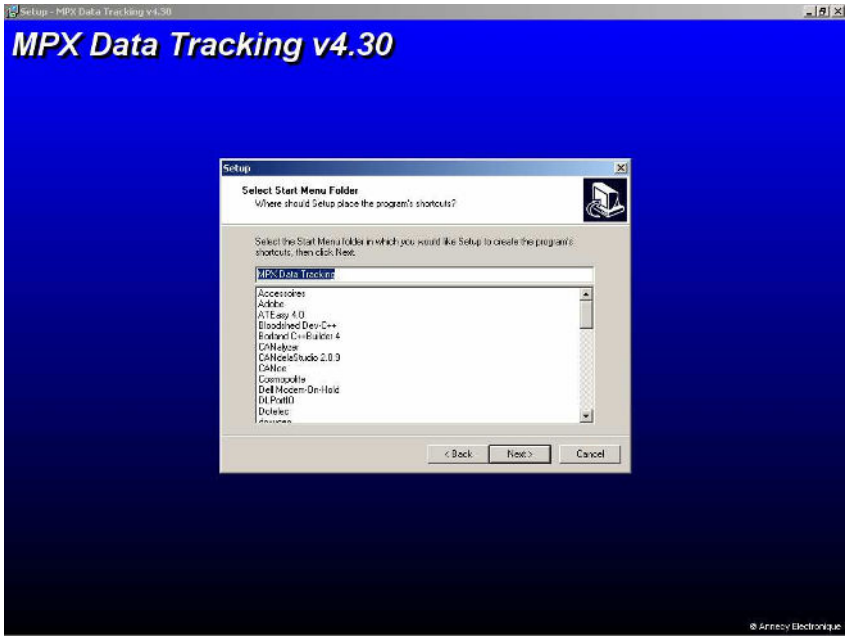
Go to the MuxTrace directory in the CDROM and double click on SETUP.EXE



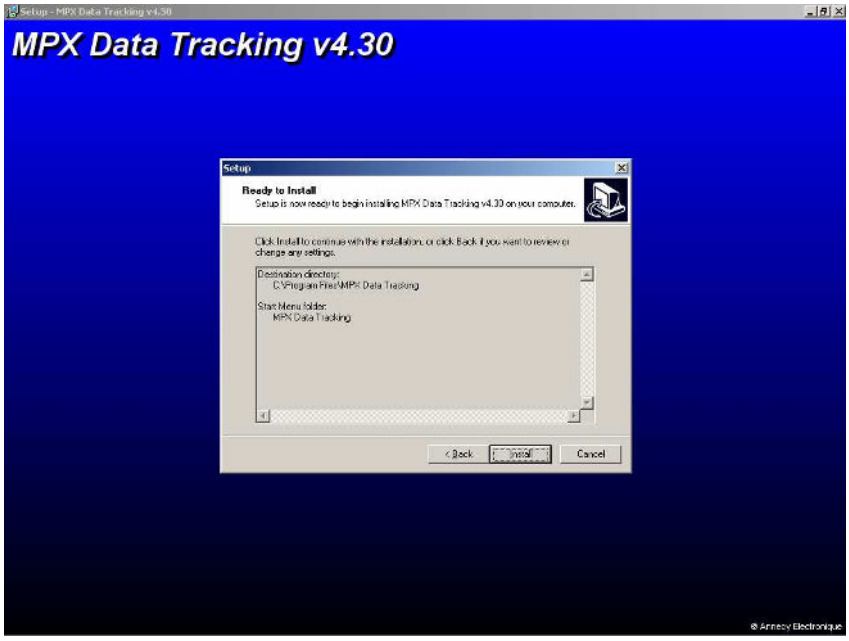
Choose directory for files to be installed



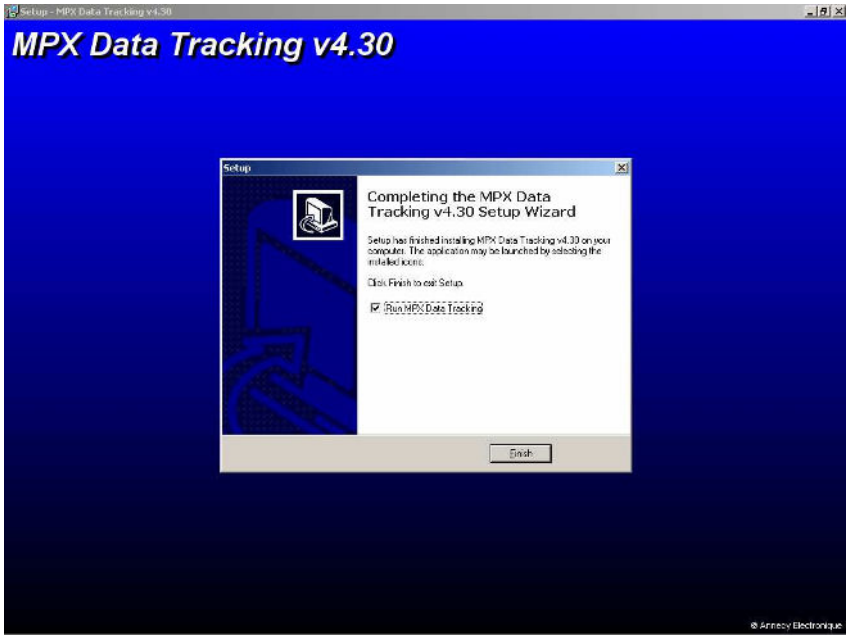
Click *Next*



Click *Finish*



Installation is completed



3 MuxTrace Software

3.1 Presentation

The MuxTrace program is organized according to a project which consists of :

- General configuration of the project,
- Configuration of each network,
- Possibility to add an user's program,
- Configuration of signals to be visualized,
- Configuration of digital inputs

- The project configuration allows the user to define both the networks that shall be displayed and messages to send.
- The configuration of the networks allows the user to define the characteristics of each network CAN, LIN, ISO, J1587, VAN or NWC, baud rate, spy mode, sample points and other settings specific to each network, as well as the list of messages sent.
- The programming makes it possible in the form of a DLL associated with MuxTrace to carry out for example complex scenarios or to start recordings on particular conditions.
- The configuration of the signals allows the user to define a classification and a list of the signals found in the data bases. The signals coming from the networks will have to be decoded.
- The configuration of digital inputs allows the user to define the surveillance of these inputs.
- The configuration of digital outputs allows the user to define their remote triggering.

Each project can be logged to a file (*.MTP) and like that they can be used later.

MuxTrace features advanced functions:

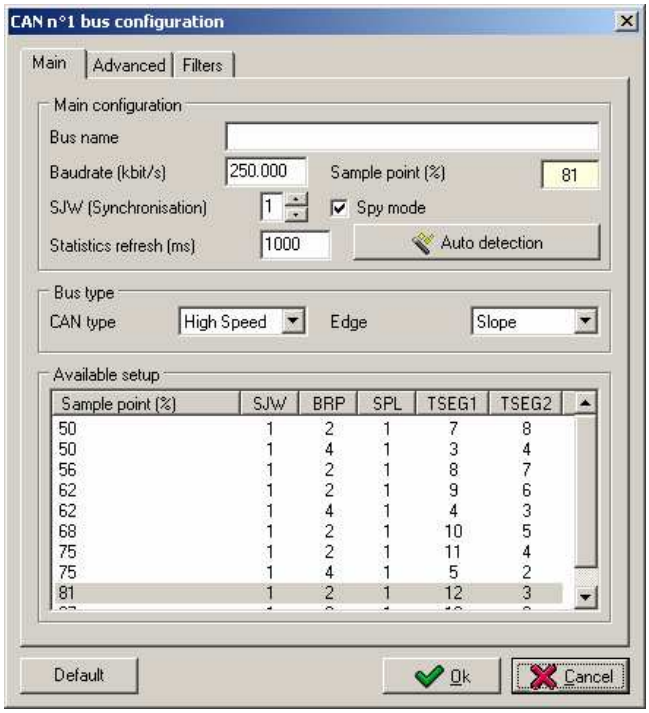
- managing data bases,
- displaying signals present in the data bases,
- surveillance of digital inputs,
- managing the communication layer Diag On Can (*Iso15765-2*),
- saving messages into text files

All these functions are available on the Expert Mode of the MuxTrace (cf *Expert mode*).

3.2 Configuration of the CAN network

This configuration depends on the type of board installed in the PC. Up to 4 CAN networks can be set up. This configuration will allow the user to choose the configuration of the different parameters related to the CAN bus.

3.2.1 General configuration of the CAN network



Bus name Logical name given to the network. It is shown while this network is in use.

Baud rate Network baud rate expressed in Kbit/sec

SJW Synchronization jump width

Spy mode Not selected: the board behaves like a CAN station, active on the network. It can send messages as well as generate acknowledgement and frame errors.
Selected: the board is totally inactive on the network. It is impossible to either send messages or generate acknowledgement or frame errors.

Auto detection Seek automatically the bus bit rate.

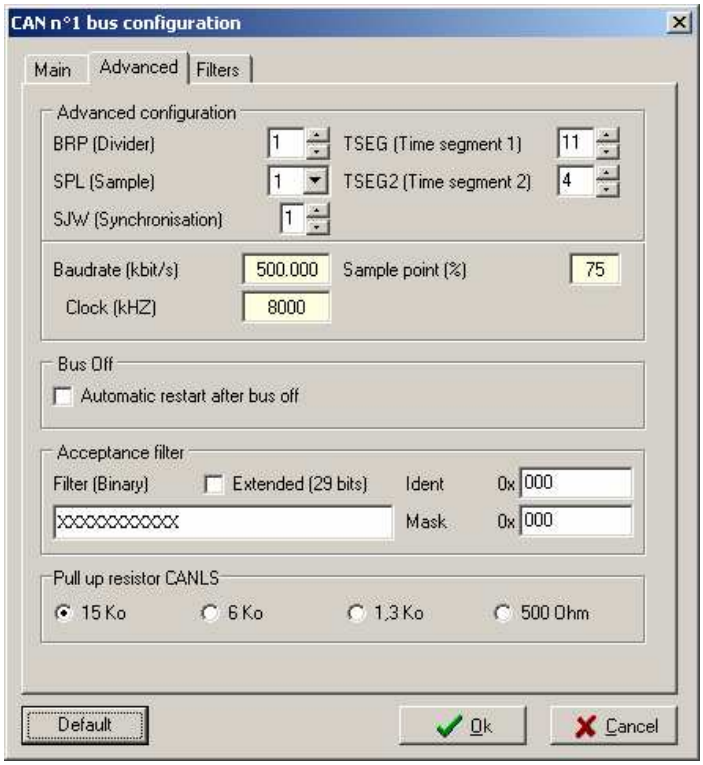
Statistics refresh Shows the period of statistics refresh in the bus. Value 0 deactivates the statistics.

Bus type Choice between bus CAN high speed and CAN low speed – fault tolerant.
This choice depends on the type of board used. It is carried out either by the programme or by a staple on the board.

Edge It selects for the bus CAN high speed the transition slope on the CANH and CANL lines.

Available setup The sample point is introduced so as to select, depending on the baud rate, the different possible configurations of the TSEG1, TSEG2 and BRP parameters.

3.2.2 Advanced configuration of the CAN network



BRP Clock divider. The divider allows the user to define the time basis of the CAN protocol controller from its clock. This time basis is expressed in quantum and it is used as a reference for the TSEG1, TSEG2 and SJW settings.

SJW Synchronization jump width (in quantum)

TSEG1 Time previous to sample point (in quantum)

TSEG2 Time post sample point (in quantum)

SPL Number of sample points

Bus off This parameter enables the beginning of communication after the CAN bus controller goes into disconnected mode « bus

off ».

Acceptance filter The acceptance filter allows the user to decrease the number of messages received by the PC by placing a reception filter on the messages the user does not want to deal with.

Extended Filter of identifiers, standard (11 bits) or extended (29 bits)

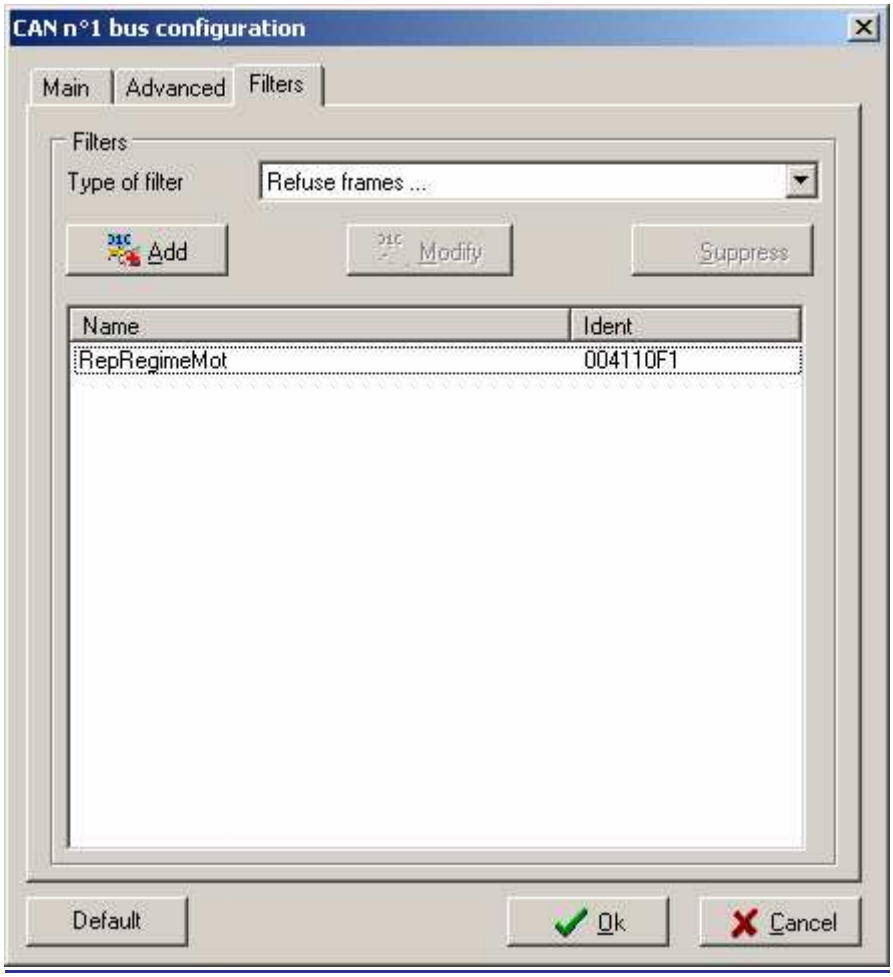
Binary filter Bit-to-bit filter of identifiers that the user wishes to filter.
0 : Filter which lets through identifiers with bit 0
1 : Filter which lets through identifiers with bit 1
X : No filter

The filter can also be obtained with Ident and Mask parameters.

Pull up resistor This setting allows the user to dynamically configure the value of pull-up and pull-down resistors in the CAN low speed. The global impedance of the CAN low speed network depends on the number of ECUs present in it.
Note: This setting is only used with certain type of material.

3.2.3 CAN network filters

The filters allow displaying only the messages to analyze.
Filtering is defined by a standard or wide identifier.



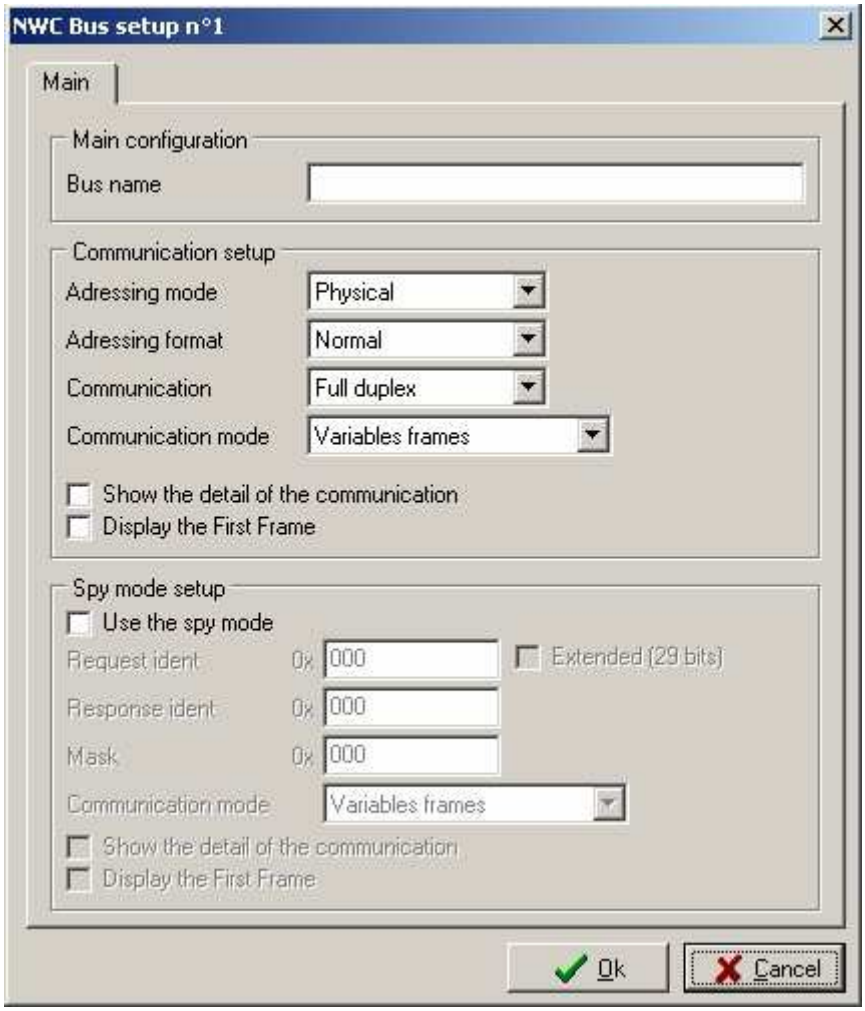
Type of filter

Filter definitions:
Accept all frames: No active filters
Refuse frames : Only the specified frame are not displayed
Accept only frames : Only the specified frame are displayed

3.3 Configuration of the NWC network (Diag On Can)

This configuration depends on the type of board installed in the PC, the number of NWC networks depends on the number of CAN networks found on the board, which will allow the user to choose the configuration of the different parameters related to the CAN bus.

3.3.1 General configuration of the NWC network (Diag On Can)



Bus name — Logical name given to the network. Shown during execution.

Addressing mode — Physical or functional addressing

Addressing format — Address normal, normal fixed, extended or mixed

Communication — Half Duplex or Full Duplex

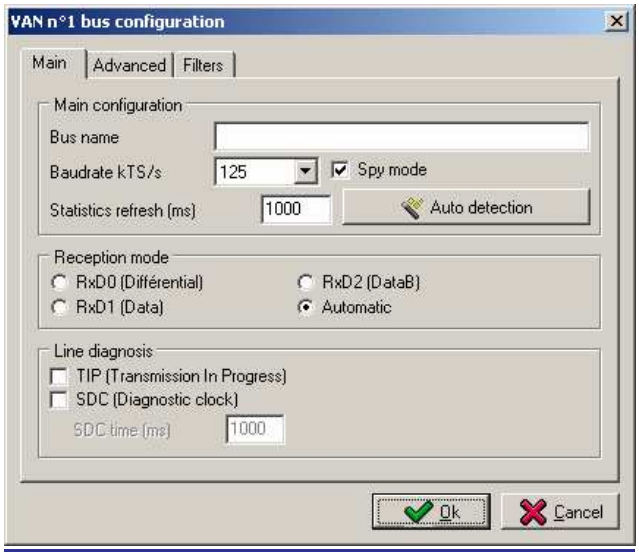
Communication mode — Transmission of variable frames or of fixed frames of 8 bytes.

Show the detail of the communication	Shows the detail of the segmentation in the CAN network visualization window associated to the NWC network.
Display the First Frame	Displays *First Frame* frames on the network in the visualization window.
Use the spy mode	Activates the bus's spy mode.
Request ident	Request identifier or FC response
Response ident	Response identifier or FC request
Mask	Mask that allows user to define 2 groups of identifiers to be interpreted.
Communication mode	Transmission of variable frames or of fixed frames of 8 bytes.
Show the detail of the communication	Displays details of the segmentation in the CAN network visualization window associated to the NWC network.
Display the First Frame	Displays *First Frame* frames on the network in the visualization window.

3.4 Configuration of the VAN network

This configuration depends on the type of board installed in the PC. Up to 4 VAN networks can be set up. It allows the user to choose the configuration of the different parameters related to the VAN bus.

3.4.1 General configuration of the VAN network



Bus name Logical name given to the network. Shown during execution.

Baud rate Network baud rate expressed in kilotime slot/sec

Auto detection Seek automatically the bus bit rate.

Spy mode Not selected: the board accepts all messages transiting the network.
Selected: the board is totally inactive in the network. No acknowledgement when the frame is received. However, it is possible to send messages and to respond in the frame.

Statistics refresh Shows the period of statistics refresh in the bus. The value 0 deactivates the statistics.

Reception mode Reception line of protocol controller

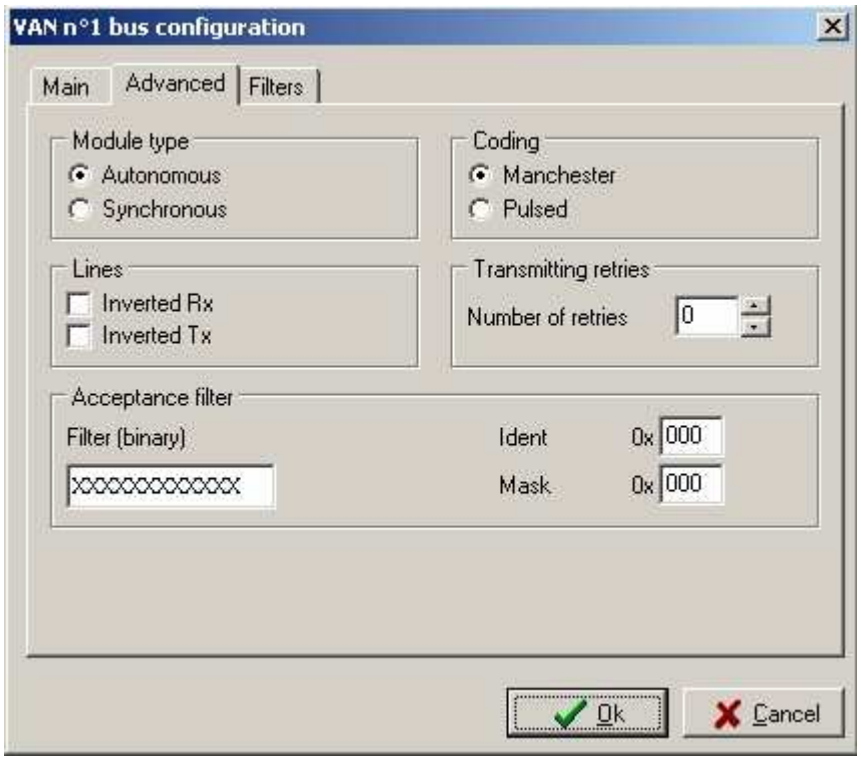
- RXD0 : Forced reception in differential mode
- RXD1 : Forced reception on the data line
- RXD2 : Forced reception on the datab line
- Automatic: the choice of reception line is carried out according to a protocol controller's internal algorithm. Going from one line to the other is done automatically.

TIP Diagnosis being sent. This parameter depends on the desired application. It is generally used to detect a line opening.

SDC Validation of the diagnostic clock. When an error, on the line, is detected, the SDC clock allows the user to monitor communication in both lines, so as to indicate the fact of being back in differential mode if the error disappears.
Caution: This parameter depends on the charge and on the bus baud rate.

Clock Value of the diagnostic clock

3.4.2 Advanced configuration of the VAN network



Module type Autonomous: When sending out a message, the board can generate a SOF (Start of frame). The message is sent out immediately.
Synchronous: When sending out a message, the VAN protocol controller cannot generate a SOF and monitoring the network. When a SOF of a message from another station transits onto the network, the message is transmitted and clashed with the current message.

Coding Manchester : By default, coding used by the VAN interface line found on the boards.

Pulsed : Coding that might be used by an external line interface (optical fiber for example).

Inverted Rx line
Inverted Tx line
Possibility to invert the logical status of recessive and dominant levels (for external line interface only)

Number of retries Number of retries during a transmission in the event of error.

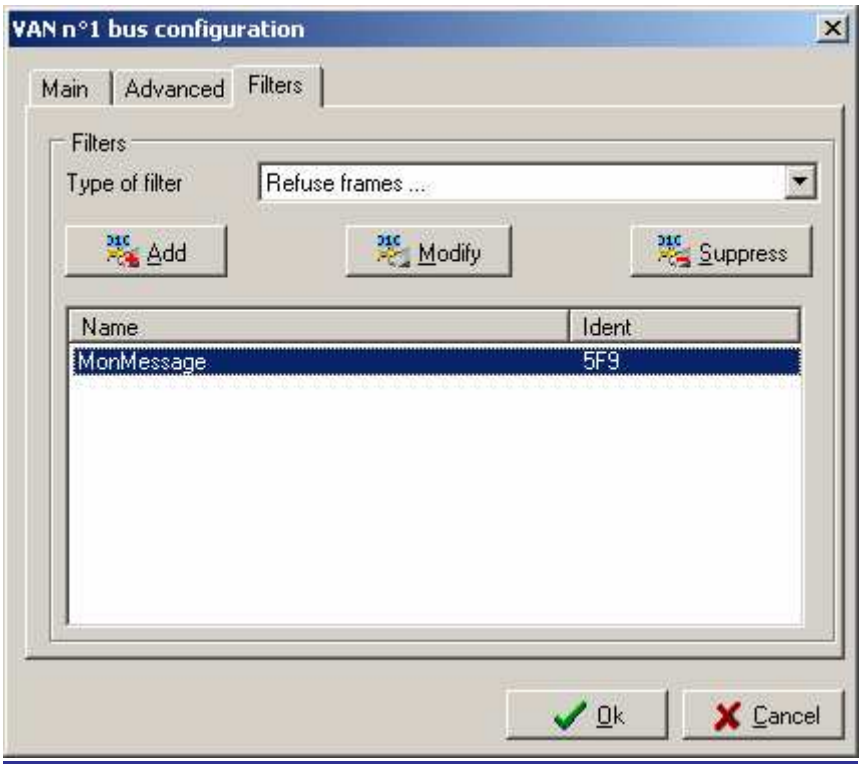
Acceptance filter The acceptance filter allows the user to decrease the number of messages received by the PC by placing a reception filter on the messages the user does not want to deal with.

Binary filter Bit-to-bit filter of identifiers that the user wishes to filter.
0 : Filter which lets through identifiers with bit 0
1 : Filter which lets through identifiers with bit 1
X : No filter

The filter can also be obtained with Ident and Mask parameters.

3.4.3 VAN network filters

The filters allow displaying only the messages to analyze.
Filtering is defined by a standard or wide identifier.



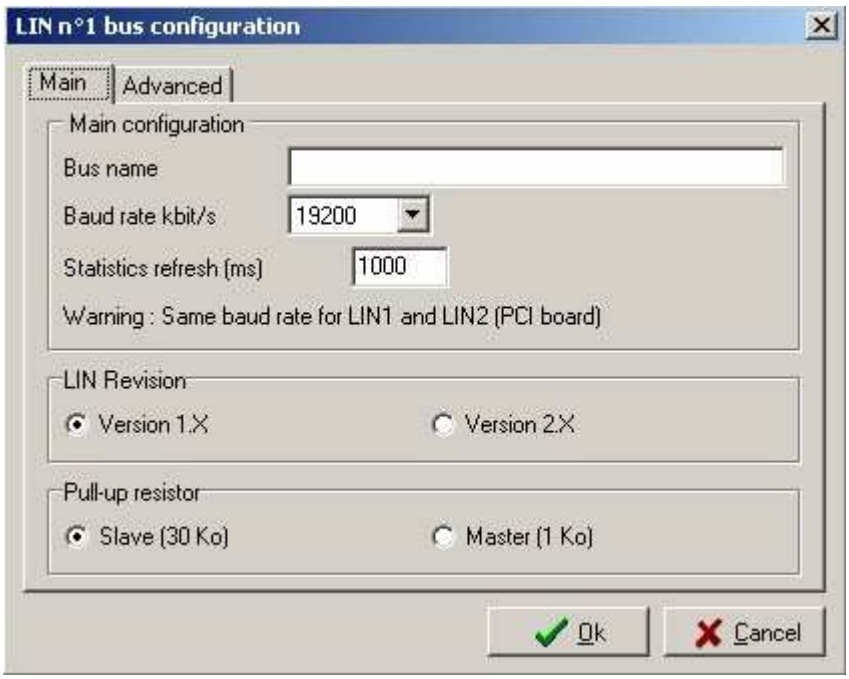
Type of filter Filter definitions:
Accept all frames: No active filters
Refuse frames : Only the specified frame are not displayed

Accept only frames : Only the specified frame are displayed

3.5 Configuration of the LIN network

This configuration depends on the type of board installed in the PC. Up to 2 LIN networks can be set up. It will allow the user to choose the configuration of the different parameters related to the LIN bus.

3.5.1 General configuration of the LIN network



Bus name Logical name given to the network. Shown during execution.

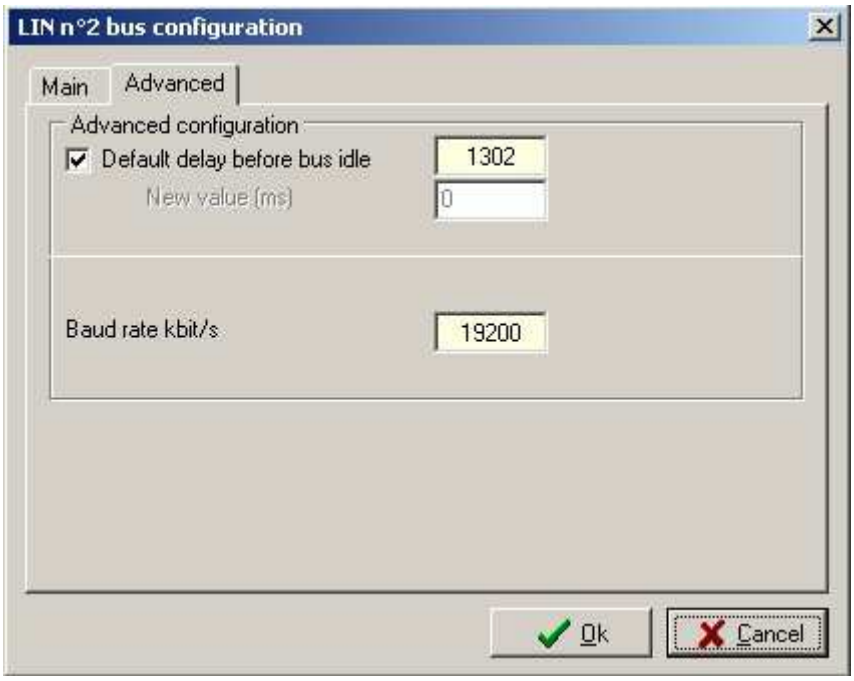
Baud rate Network baud rate expressed in kbit/sec

Statistics refresh Shows the period of statistics refresh in the bus. Value 0 deactivates the statistics.

Revision LIN Version 1.X : The calculation of the CRC is in conformity with the revision LIN 1.0, 1.2 and 1.3
Version 2.X : The calculation of the CRC is in conformity with the revision LIN 2.0.

Pull up resistor This setting allows the user to dynamically configure the value of pull-up resistors.

3.5.2 Advanced configuration of the LIN network



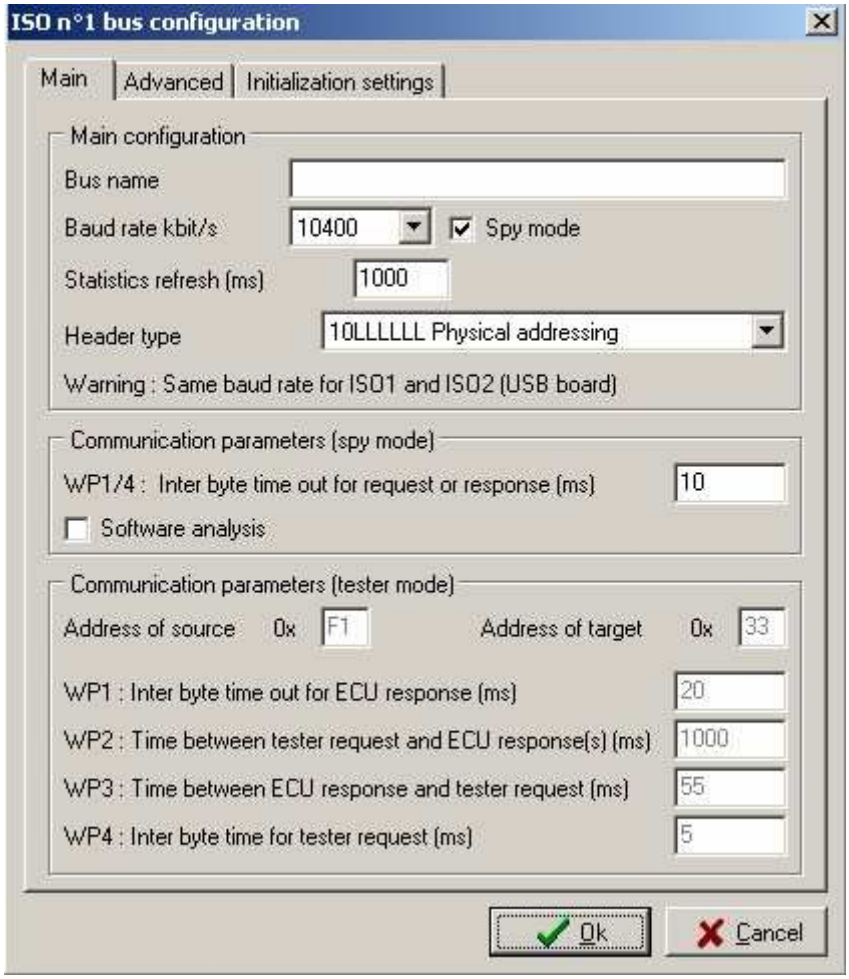
Default delay before bus idle

By default, the LIN standard specifies that the delay before the detection of communication loss equals the duration of 25000 bits. For example, for a baud rate of 19200 kBit/sec (1 bit=52µSec) the duration of the detection of communication loss is 52µSec*25000, i.e. 1302 ms.
For a unitary test, not included in the context of total integration, it is possible to adjust this setting according to the configuration.
Note : 0 means an endless time out.

3.6 Configuration of the ISO9141 network

This configuration depends on the type of board installed in the PC. Up to 2 ISO9141 networks may be set up. It will allow the user to choose the configuration of the different parameters related to the ISO9141 (K & L) bus.

3.6.1 General configuration of the ISO9141 network



General configuration

Bus name Logical name given to the network. Shown during execution.

Baud rate Network baud rate expressed in kbit/sec

Spy mode **Selected** : Spy mode. **Not selected** : Tester mode
Spy mode : The MuxTrace program analyses the communication between a tester tool and an ECU.
Tester mode: The MuxTrace program simulates the presence of a Tester tool and allows the user to send an ECU diagnosis request.

Statistics refresh Shows the period of statistics refresh in the bus. Value 0 deactivates the statistics.

Header type This parameter selects the coding type of a header byte of an ISO9141 or ISO 14230 message.
10LLLLLL Physical addressing
11LLLLLL Functional addressing
01LLLLLL Exception mode CARB
00LLLLLL No address information

Communication parameters (spy mode)

WP1/4 Maximum value of inter-byte delay of a request or a response. This parameter is used for detecting the end of a request or a response.

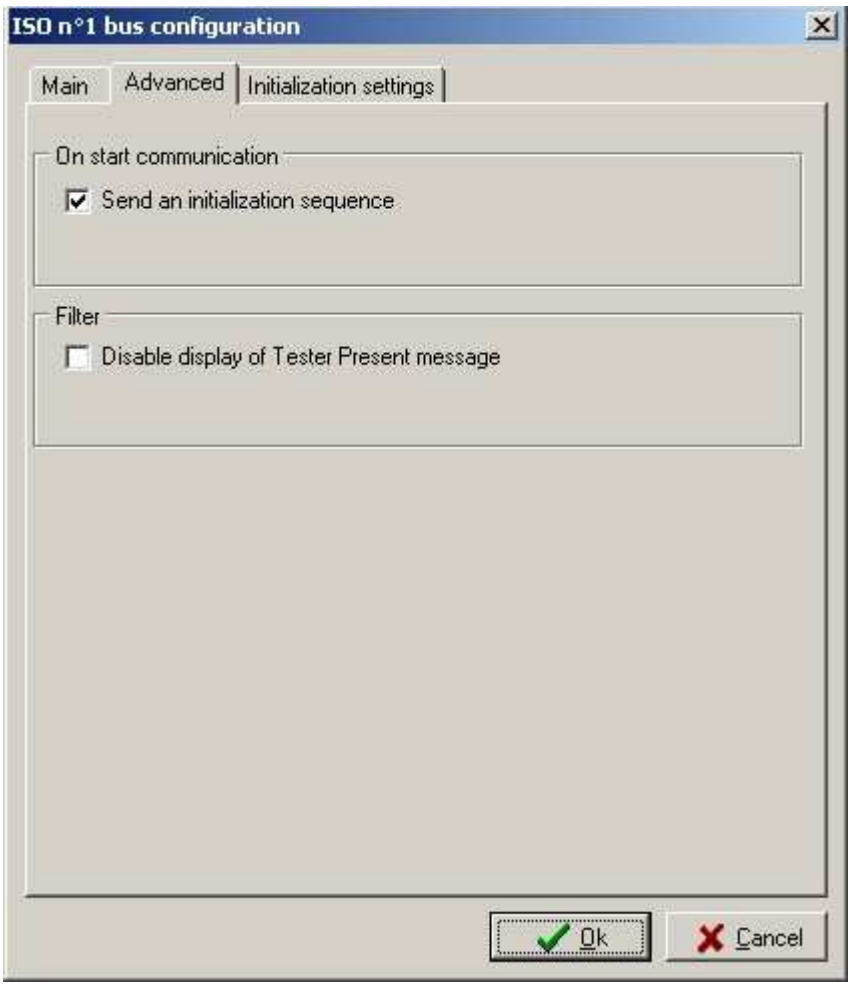
Software analysis If this option is not ticked, the end of frame detection is done through WP1/4 timeout.
If this option is ticked, the end of frame detection is done through WP1/4 timeout but also by analyzing the first header bytes that comprise the frame's length

Note : The end of a request or a response in the bus is detected when the WP1/4 times are exceeded. In order to ensure the proper functioning of the program, it is important that the times between a request and a response (WP2) or between a response and a new request (WP3) are above the inter byte times WP1/4.

Communication parameters (tester mode)

Address of source Source address in hexadecimal (tester's address)
Address of target Target ECU's address in hexadecimal.
WP1 Inter-byte time-out of the ECU's response in ms
WP2 Time-out between a tester request and the ECU's response in ms
WP3 Delay between a response from the ECU and a new request from the tester in ms
WP4 Inter-byte delay of the tester's response.

3.6.2 ISO9141 network advanced configuration.



On start communication

On start communication

Selected : Sends an initialization sequence
Not selected : No sending
When the analysis starts, validating this parameter allows the user to send an initialization sequence by means of the information Source address and Target address. The start of request is of the « init fast » or « 5 bauds init » type, followed by a Start Communication request (code 0x81).

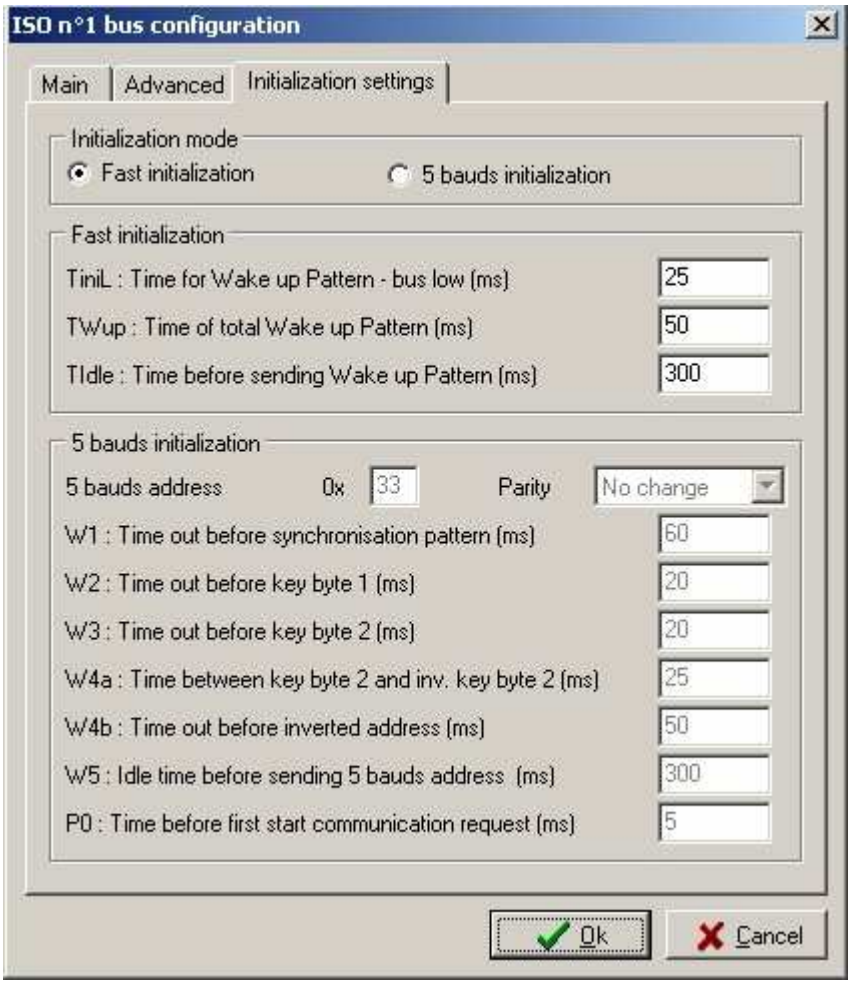
Filter

Tester Present

Selected : The request or response Tester Present is not displayed
Not selected : The request or response Tester Present is displayed
The request Tester Present is used for keeping communication but has no other applicable functions. It can be filtered so that the user does not have too much

information.

3.6.3 Initialization settings of the ISO9141 network



Initialization type

Initialization mode Fast initialization or
5 bauds initialization

Fast initialization

TiniL Time for Wake up pattern in ms

Twup Duration in ms before sending the first request. This duration consists of the low and high levels of the fast initialization sequence.

Tidle Inactivity time of the bus before sending the « wake up pattern »

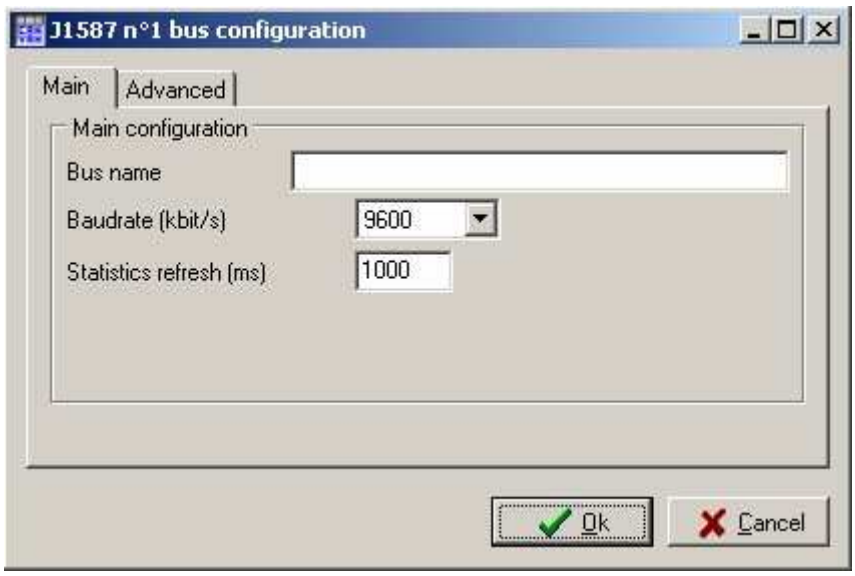
5 bauds initialization

5 bauds address	Target address in hexadecimal sent at 5 bauds
Parity	**None** : 5 bauds address bauds sent without changes **Even** : 5 bauds address sent with an even parity **Odd** : 5 bauds address sent with an odd parity
W1	Time between the end of the address byte and the beginning of the synchronization pattern
W2	Time between the end of the synchronization pattern and the beginning of key byte 1
W3	Time between key byte 1 and key byte 2
W4a	Time between key byte 2 (coming from the UCE) and its inversion, carried out by the MuxTrace
W4b	Time between inverted key byte 2 and the inverted address coming from the UCE
P0	Time between reception of the inverted address and the beginning of the emission of the StartCommunication request

3.7 Configuration of the J1587 network

This configuration depends on the type of board installed in the PC. Up to 4 J1587 networks can be set up. It allows the user to choose the configuration of the different parameters related to the J1587 bus.

3.7.1 General configuration of the J1587 network

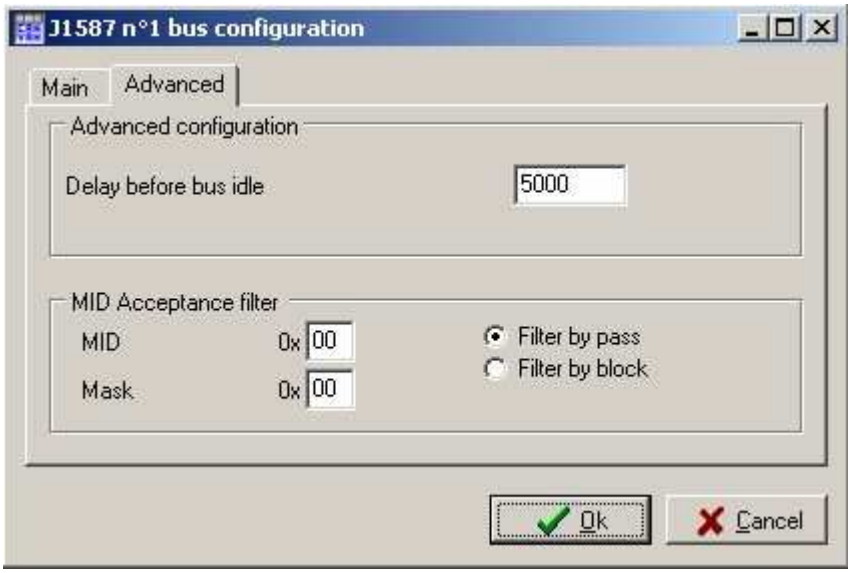


Bus name Logical name given to the network. Shown during execution.

Baud rate Network baud rate expressed in kbit/sec

Statistics refresh Shows the period of statistics refresh in the bus. Values 0 deactivate the statistics.

3.7.2 Advanced configuration of the J1587 network



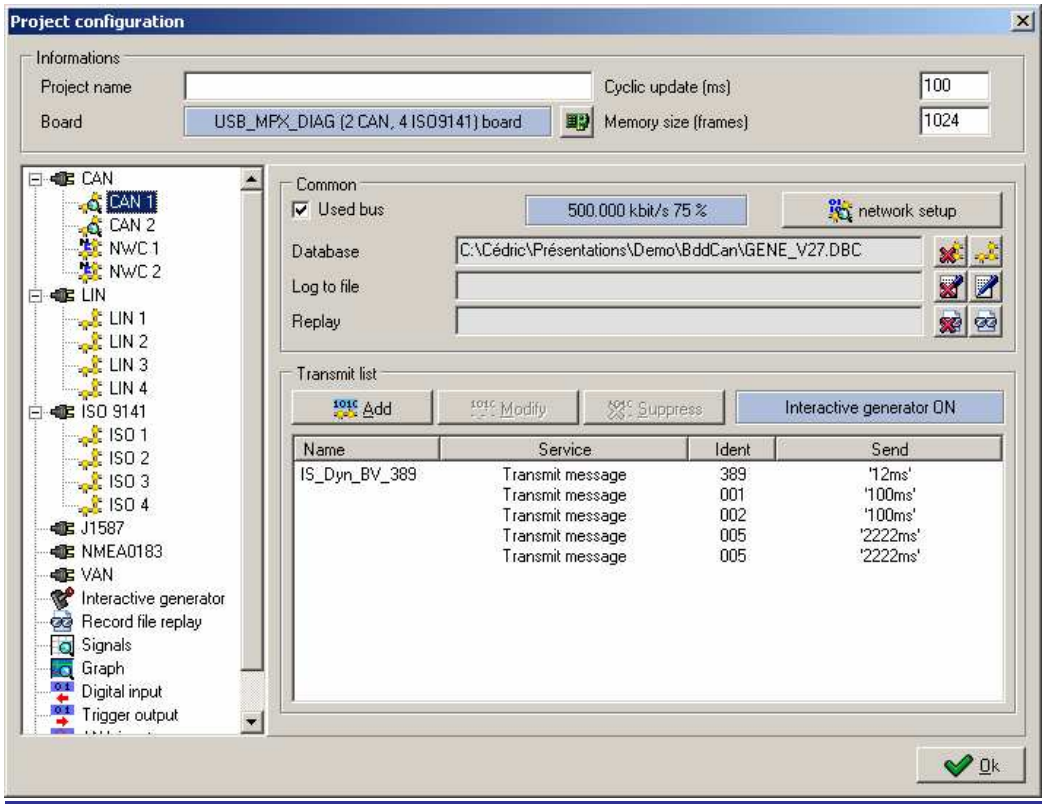
Delay before bus idle Delay before detection of loss of communication.
Note : value 0 means an endless time out.

Acceptance filter Parameter which allows the user to add acceptance filter (filter by pass / filter by block) to a message identifier or a family of message identifiers so as not to overcharge the display coming from the network.

3.8 Project configuration

The project configuration establishes how the Muxtrace will work :

- Visualization parameters
- List of used buses,
- Sending messages
- Choice of data bases
- Logging to a file
- Replay a logging file



Project name Logical name given to the project. Shown when executed.

Cyclic update In the event of a fixed visualization, this parameter allows the user to periodically update the visualization.

Memory size In the event of a sequential visualization, this parameter indicates the name of the messages saved in the memory to be visualized.

Used bus Selection of buses used during execution. A display window per bus is created.

Database Selection of a base in .DBC or .DBV format.

Log to file Selection of a file and its format in order to save the

messages transiting the network.

Replay Selection of file in order to replay messages contained.

Transmit list Creation, suppression and modification of messages to be sent by the MuxTrace

Performance

The performance of the MuxTrace program depends on the type of PC used, as well as on the frequency in which the information coming from the bus is displayed. In order to optimize this performance, we recommend to :

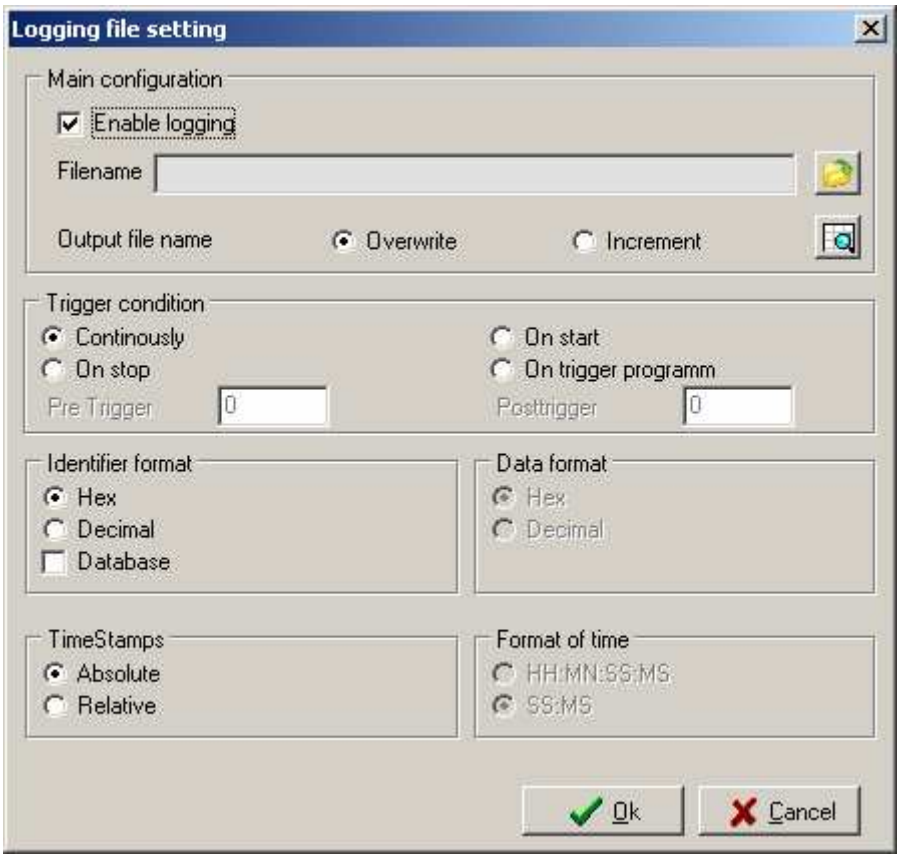
- Increase cyclic update to a maximum level
- Decrease memory size to a minimum level

3.8.1 Adding a data base

A data base associated to a network will allow the user to work with physical values of signals. The data bases supported by the MuxTrace program are in .DBC, .DBV , DBL or DBx format.

3.8.2 Creating a logging file

The messages received can be logged onto a text file for subsequent analysis.



Enable logging Enables logging messages transiting the network to a file

Filename Name of file out

Trigger conditions Continually : The recording is carried out between the starting and the stop of measurement.
On start : The recording is carried out between the starting of measurement and stops n messages after (defined by post trigger)
On stop : The recording is carried out between n messages (defined by pre trigger) and the stop of measurement and stops.
Trigger : The recording is carried out between N messages (defined by pre trigger) and N messages (defined by post trigger) around the trigger of release defined by programming (see chapter programming)

Format of identifier Hex : Identifier in hexadecimal
Decimal : identifier codified in decimal (EXCEL…)
Database : If the data base exists, the logical name of the message is registered.

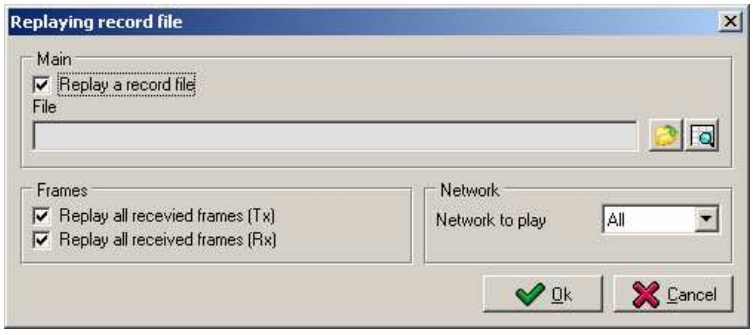
Format of data Hex : Data in hexadecimal
Decimal : Data in decimal (EXCEL…)

Timestamps Absolute : Each event is dated in reference to the start of communication
Relative : Each event is dated in reference to the previous event

Format of time HH :MN :SS :MS (Hours, minutes, seconds, milliseconds)
SS MS (EXCEL type usage…):

3.8.3 - Replay a logging file

The messages recorded in a asc file, can be replayed in order to carry out an analysis.



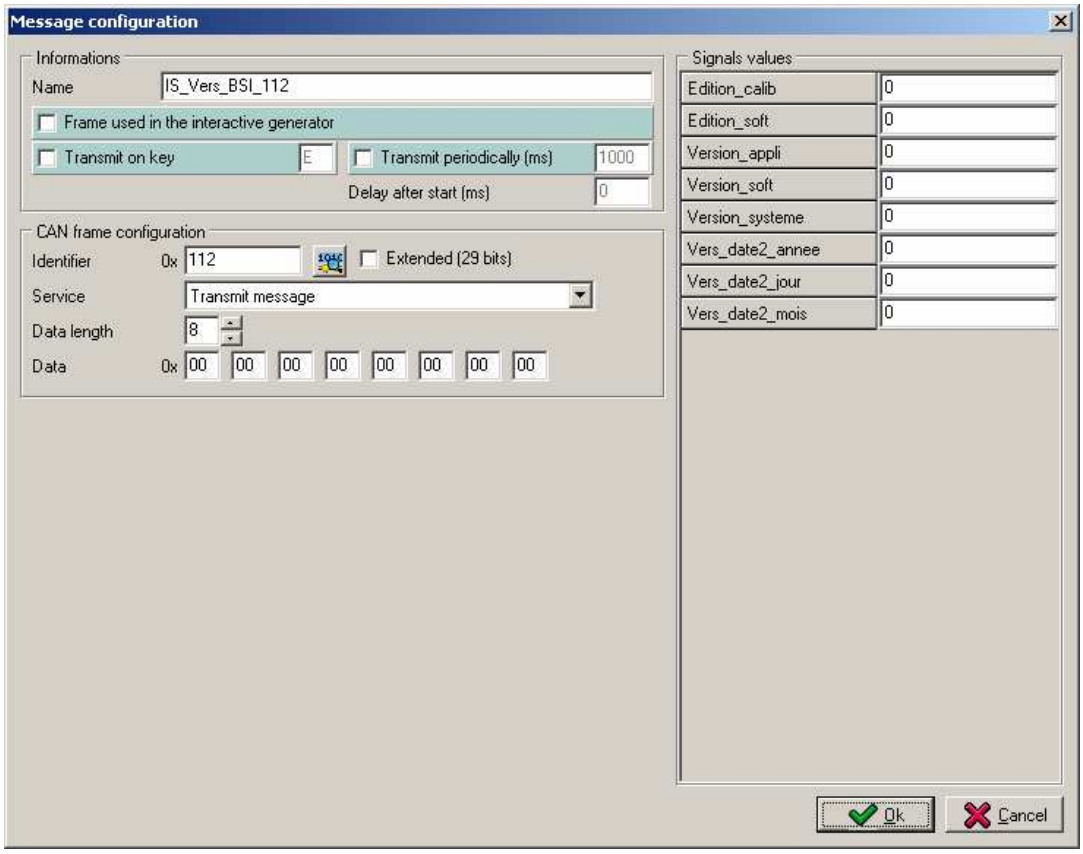
Enable Replay Enables to replay recorded messages

Replay Tx Frames Replay all the messages marked like Tx in this file

Replay Rx Frames Replay all the messages marked like Rx in this file

Replay Selected bus Select a bus to replay from the file.

3.8.4 Creating a CAN message



Name — Logical name given to the message. Shown when this list is displayed.

Frame used in the interactive generator — This message will be able to be modified in the interactive generator.

Transmit on key — Transmits every time the selected key of the message is pressed.

Transmit periodically — Periodical transmission of the message in milliseconds.

Frame configuration — Frame selection in a data base.
Frame configuration is done automatically.

Identifier — Value of message identifier

Extended — Selection of type of identifier : Standard (11 bits) or Extended (29 bits)

Service — CAN service of the message :

- Transmit message

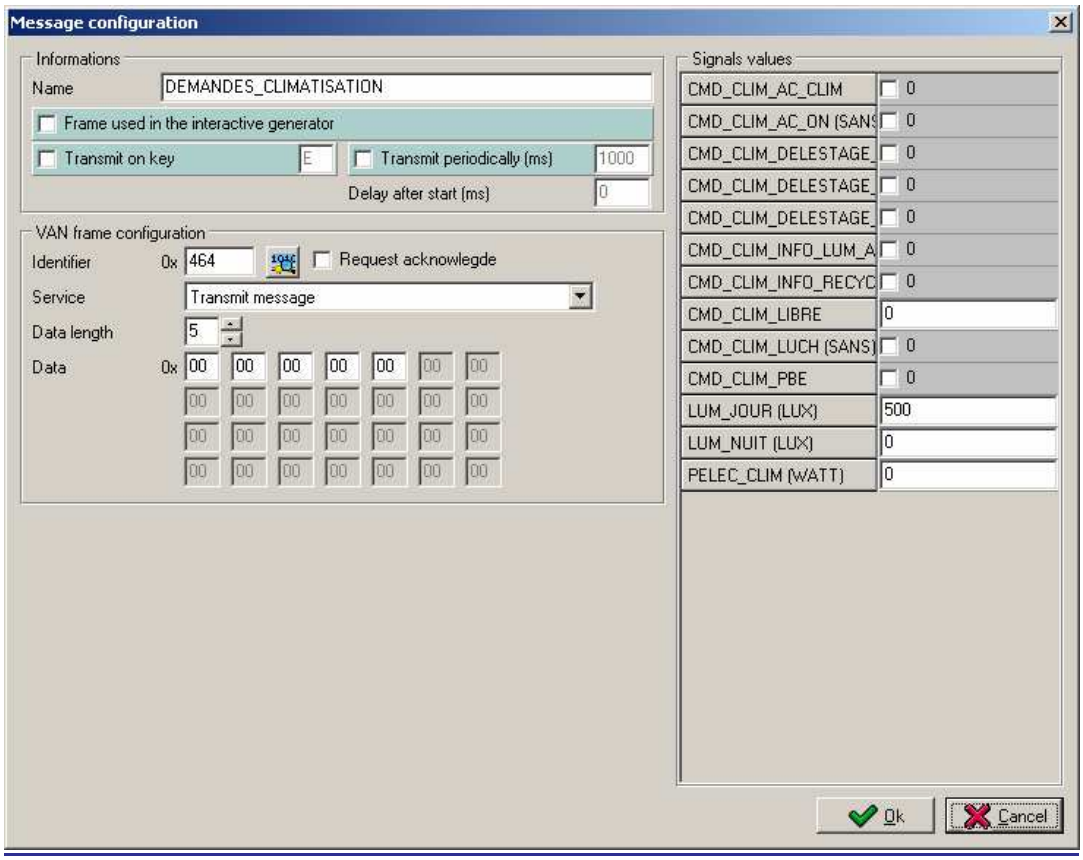
- Remote frame

Data length Length of the data included in the message

Data Value of the data in hexadecimal

Signals values Enter signals values to be codified in the present frame.

3.8.5 Creating a VAN message



Name Logical name given to the message. Shown when this list is displayed.

Frame used in the interactive generator This message will be able to be modified in the interactive generator.

Transmit on key Transmission every time the selected key of the message is pressed.

Transmit periodically Periodical transmission of the message in milliseconds.

Frame configuration Frame selection in a data base.
Frame configuration is done automatically.

Identifier Value of he message identifier

Request acknowledge Indicates if the message requires request acknowledgement.

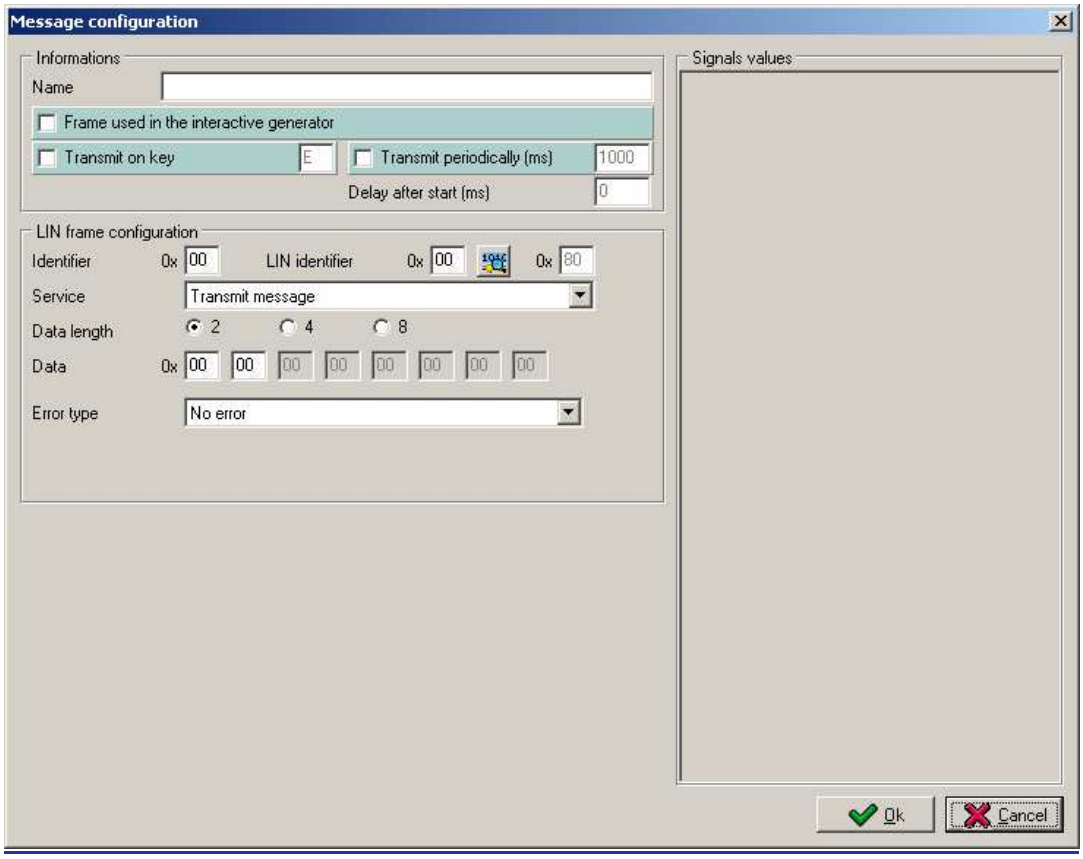
Service VAN service of the message
- Transmit message
- Request an in frame response
- In frame response
- Differed reply message

Data length Length of data included in the message

Data Value of the data in hexadecimal

Signals values Enter signals values to be codified in the current frame.

3.8.6 Creating a LIN message



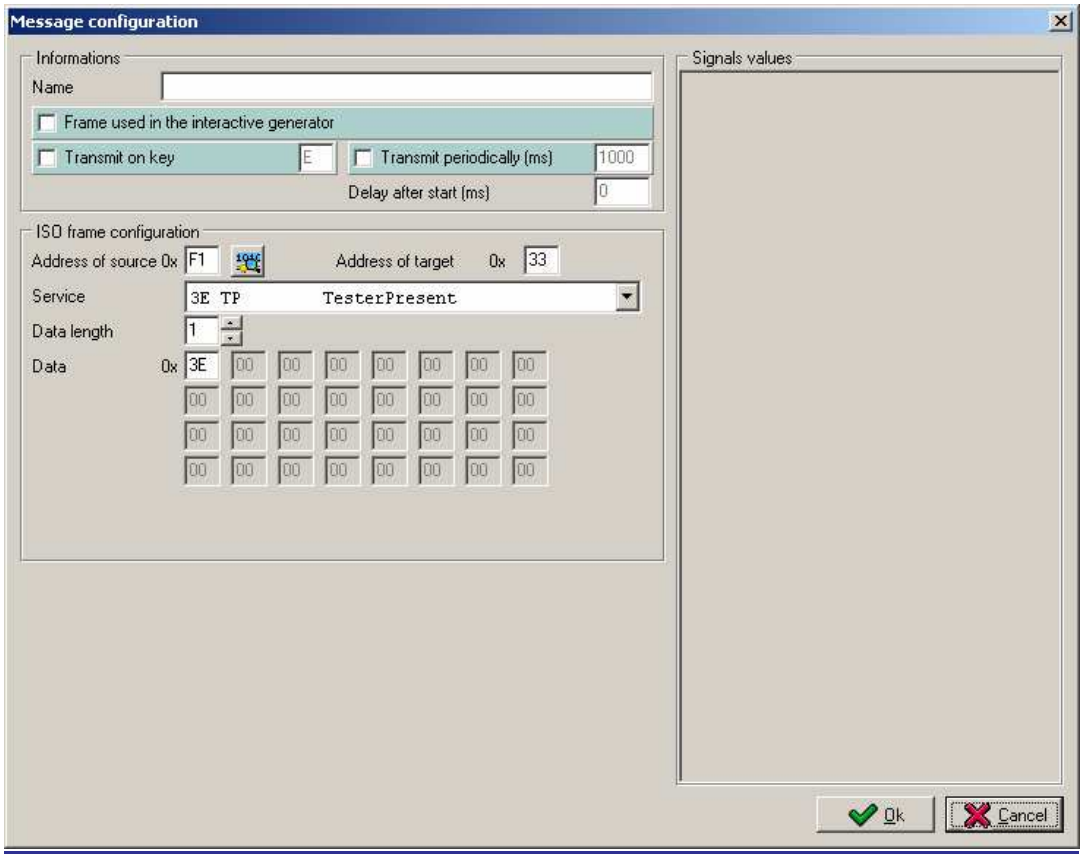
Name Logical name given to the message. Shown when this list is displayed.

Frame used in the interactive generator This message will be able to be modified in the interactive generator.

Transmit on key Transmission every time the selected key of the message is pressed.

Transmit periodically	Periodical transmission of the message in milliseconds.
Frame configuration 	Frame selection in a data base. Frame configuration is done automatically.
Identifier	Value of the message identifier
LIN Identifier	Value of the message identifier which includes a field with the length of the message (6 bits)
Service	VAN service of the message - Transmit message - Request an in frame response - In frame response
Data length	Length of data included in the message
Data	Value of the data in hexadecimal
Signals values	Enter signals values to be codified in the present frame.
Error type	In order to send a protocol test, different frame types with protocol errors can be sent. - No errors - P0 Parity bit error - P1 Parity bit error - CRC error - Synchro byte error - Data length +1 error - Data length +2 error - Data length -1 error - Data length -2 error

3.8.7 Creating an ISO message



Name Logical name given to the message. Shown when this list is displayed.

Frame used in the interactive generator This message will be able to be modified in the interactive generator.

Transmit on key Transmission every time the selected key of the message is pressed.

Transmit periodically Periodical transmission of the message in milliseconds.

Frame configuration Frame selection in a data base.
Frame configuration is done automatically.

Address of source Address of source transmitted in the message. By default, the address of source corresponds to that programmed in the configuration parameters.

Address of target Address of target transmitted in the message.

Service KWP service of the message corresponding to the first byte of data. The proposed services are those ones described by the ISO14230 standard.

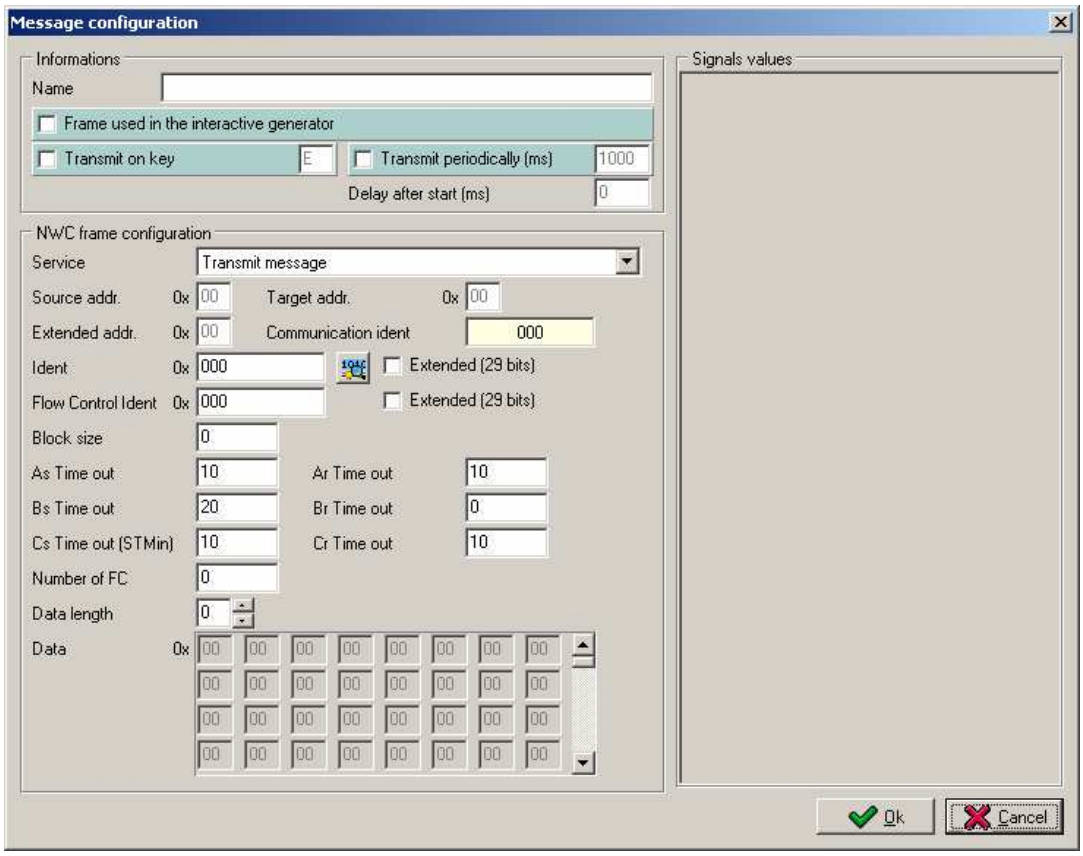
Data length Length of data included in the message (excluding header byte,

address of source, address of target and CRC).

Data Value of data in hexadecimal. The values will be given according to the selected service.

Signals values Enter signals values to be codified in the present frame.

3.8.8 Creating an NWC message



Name Logical name given to the message. Shown when this list is displayed.

Frame used in the interactive generator This message will be able to be modified in the interactive generator.

Transmit on key Transmission every time the selected key of the message is pressed.

Transmit periodically Periodical transmission of the message in milliseconds.

Service Type of service: Transmit message or Data reception

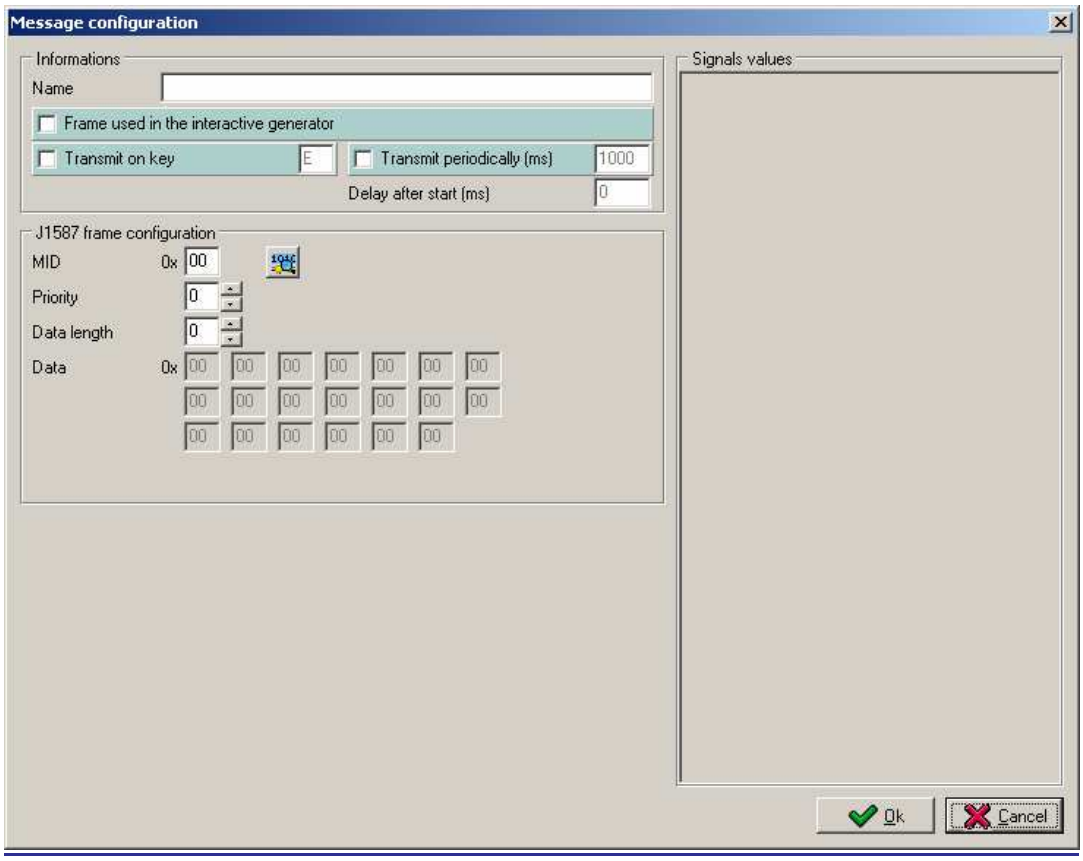
Source addr. Address of source

Target addr. Address of target

Extended addr. Extended address

Communication ident	Communication identifier that will transit the network.
Extended	Selection of type of identifier. Standard (11 bits) or Extended (29 bits).
Ident	Frame identifier.
Flow Control Ident	Identifier of the flow control frame.
Frame configuration 	Frame selection in a data base. Frame configuration is done automatically.
Block size	Number of consecutive blocks after reception of a flow control frame.
As Time out	Maximum transmission time of the transmitter.
Ar Time out	Maximum transmission time of the receiver.
Bs Time out	Time until reception of flow control.
Br Time out	Time until transmission of flow control.
Cs Time out (STMin)	Time between 2 blocks.
Cr Time out	Time until transmission of Consecutive Frame.
Number of FC	Number of maximum flow control expected.
Data length	Length of data included in the message.
Data	Value of data in hexadecimal. The values will be given according to the selected service.
Signals values	Enter signals values to be codified in the current frame.

3.8.9 Creating a J1587 message



Name	Logical name given to the message. Shown when this list is displayed.
Frame used in the interactive generator	This message will be able to be modified in the interactive generator.
Transmit on key	Transmission every time the selected key of the message is pressed.
Transmit periodically	Periodical transmission of the message in milliseconds.
Frame configuration 	Frame selection in a data base. Frame configuration is done automatically.
MID	Value of the message identifier
Priority	Transmission priority of the message [0-7]
Data length	Length of data included in the message.
Data	Value of the data in hexadecimal

Signals values Enter signals values to be codified in the present frame.

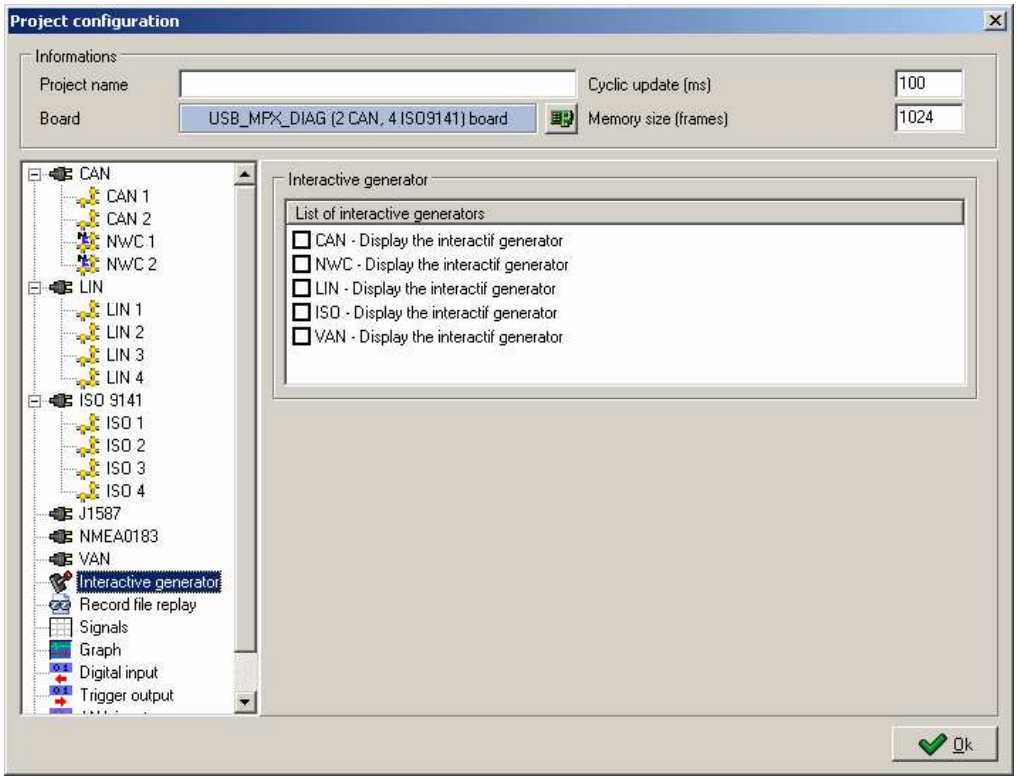
3.9 Interactive generator

Interactive generator allows user to modify transmitted messages during the muxtrace run.

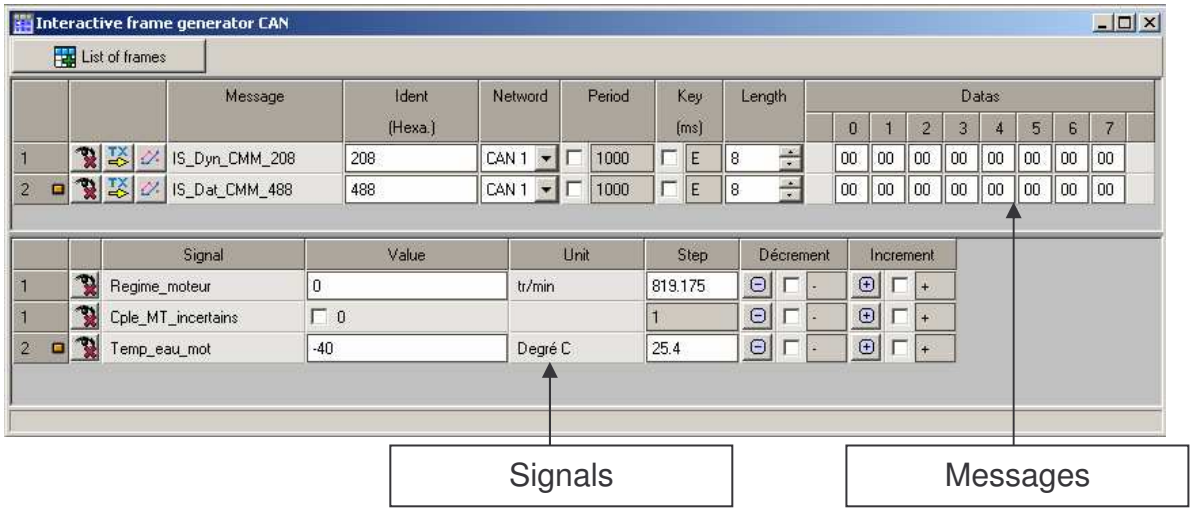
3.9.1 Posting of the interactive generators windows

The interactive generators can be used with the messages transmitted on networks CAN, NWC, FLAX, ISO and VAN.

You can activate the interactive generators from this window:



3.9.2 Interactive generator window



3.9.2.1 Messages

The list of the messages makes it possible to modify during acquisition:

- Identifiers value,
- period of the message,
- the key of emission,
- Message size and message data.

Selection of Frames  Allows user to choose messages having to appear in the interactive generator.

Message index  Indicate message index allowing user to make the correspondence with signals.

Delete  Allows user to delete a message and all associated signals

Sending a message  Sending this message.

Selection of signals  Allows user to choose signals having to appear in the interactive generator.

3.9.2.2 Messages

The list of the signals makes it possible to modify during acquisition:

- signal value,
- the step for increase and decrease the signal,
- To increase or to decrease the signal value with buttons or keys.

Message index  Indicate the index of the message to which the signal belongs.

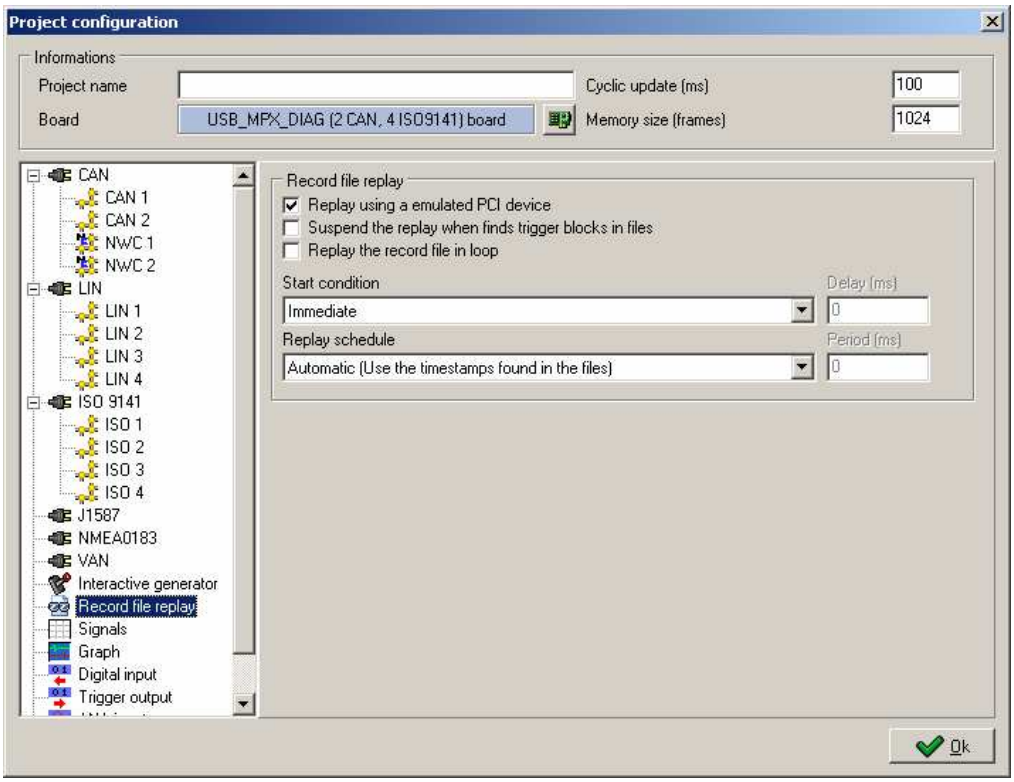
Delete  Remove this signal of the interactive generator.

3.10 Replay asc files

MuxTrace allows user to replay messages contained in an ASC file.

3.10.1 Configuration of replay functionality

Once an asc file is associated with at least one bus, the replay functionality is activated. It is however necessary to configure the general.



3.10.1.1 General options

The general options allow:

- To use the demonstration mode in order to not emit physically on a real bus, thus avoiding disturbing the network.
- To suspend the replay functionality when a trigger is found in the asc files.
- Repetitive output sending mode

3.10.1.2 Start conditions

- **Immediately**:

The first message is transmitted at the start of measurement.

- **With the first event**:

The replay starts according to the dating of the first message present in the file. Thus, if the first message is dated at 15 seconds, the replay will begin 15 seconds after the startup of acquisition.

- **After timeout**: The first message will be transmitted after a time chosen by the user.
- **With an user event**:

The first message will be transmitted when the user presses on the button of startup of replay.

3.10.1.3 To schedule replay

This functionality allows user to change periodicity of the messages found in the asc files with several manners:

- **Automatically**
The messages are transmitted according to the dating defined in the asc file.

- **According a period of x ms between each frame**
All the messages will be transmitted according to the same period chosen by user.

- **Frame by frame**
The messages are transmitted one by one. The user chose using a button when the next message will be transmitted.

- With an user program
He messages will be transmitted according to their respective dating, but the associated program will be able to choose the messages being able to be emitted and when the replay must be suspended.

3.10.2 Tool bar for the replay

3.10.2.1 State

Disabled	No asc file is associated to the replay functionality.
Enabled	Fixed picture : The replay is activated but acquisition is not started yet. Animate picture : The replay is started.
Stopped	The replay is finished.

3.10.2.2 Commandes

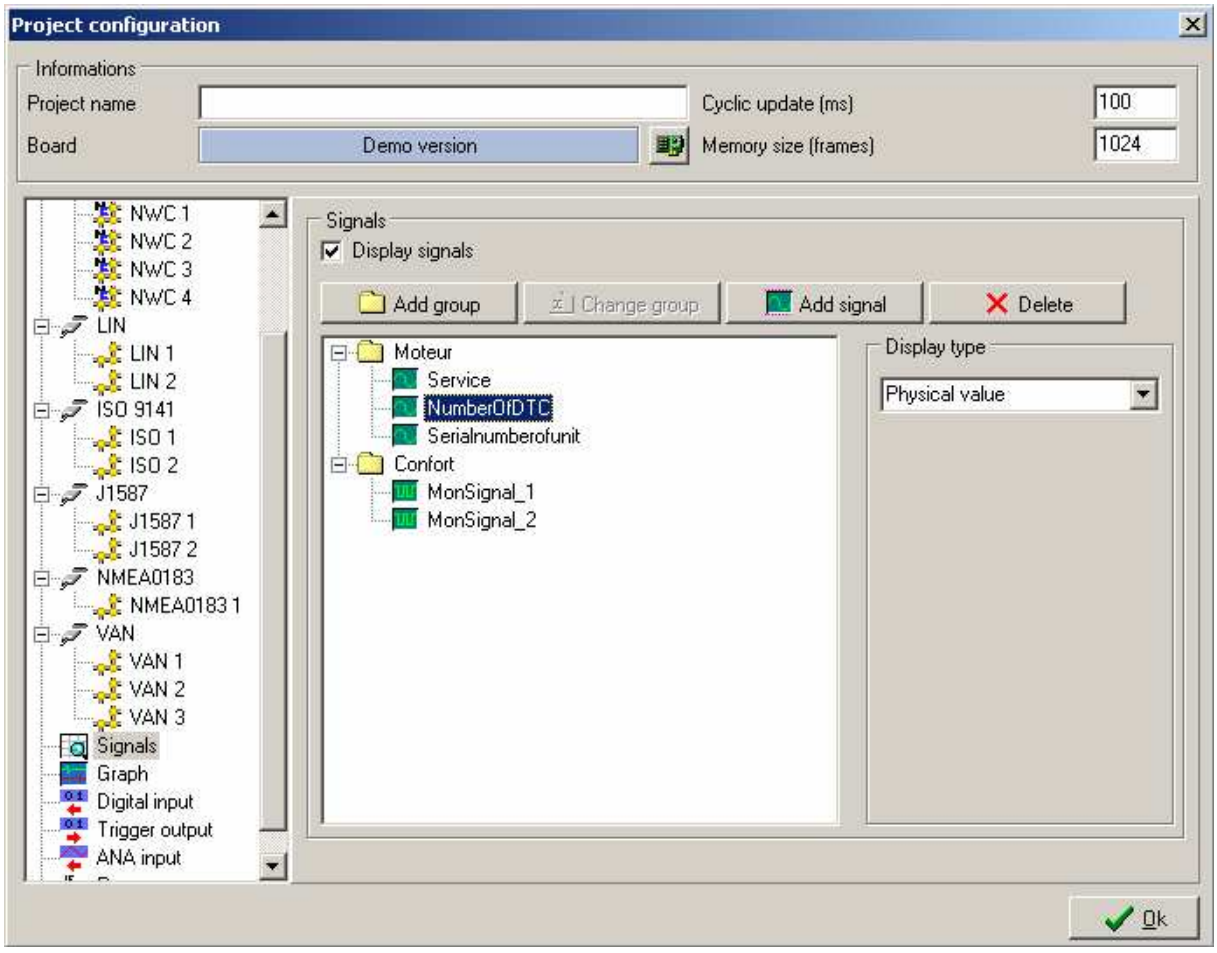
Start/Resume Starts the relay or resumes if this one were suspended.

Suspend Suspends the replay.

3.11 Display of signals

When data bases are associated to the buses, it is possible to visualize the value of the codified signals in these bases.

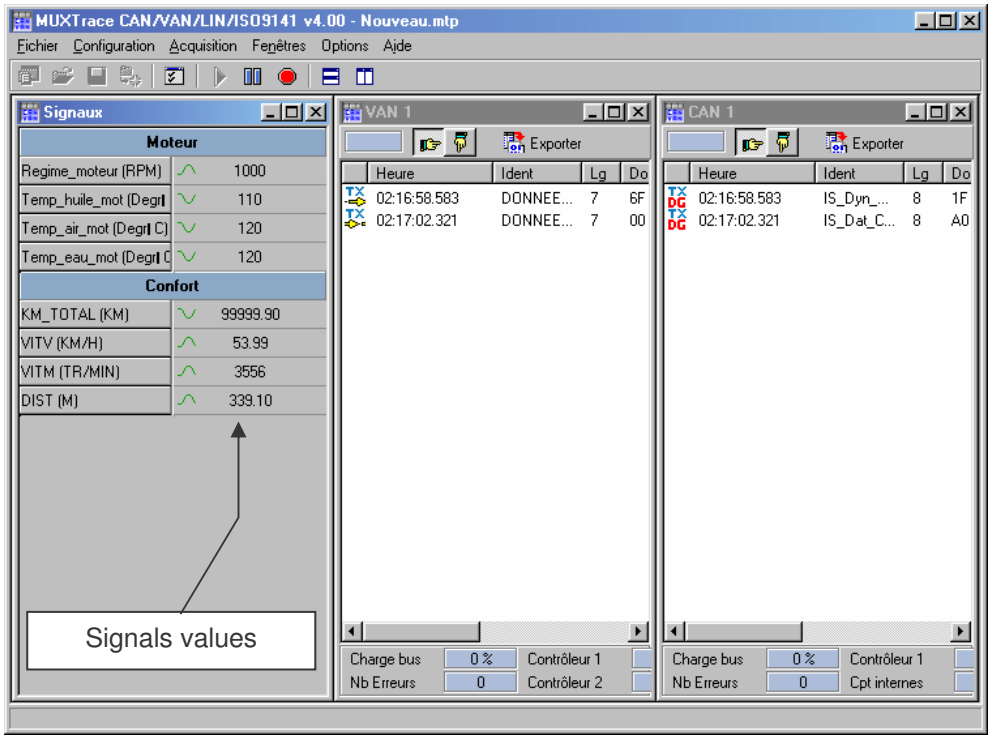
3.11.1 Creating a list of signals



The signals are ordered by group. This first selection allows the user to organize the signals according to their environment. For example, create the group *Engine* where the signals *Engine Speed, Water Temperature, …* are classified. Create the group *Comfort* where the signals *Distance Covered, Vehicle Speed, …* are classified.

3.11.2 Visualizing the signals

When the program is executed, a window opens to display the signals.



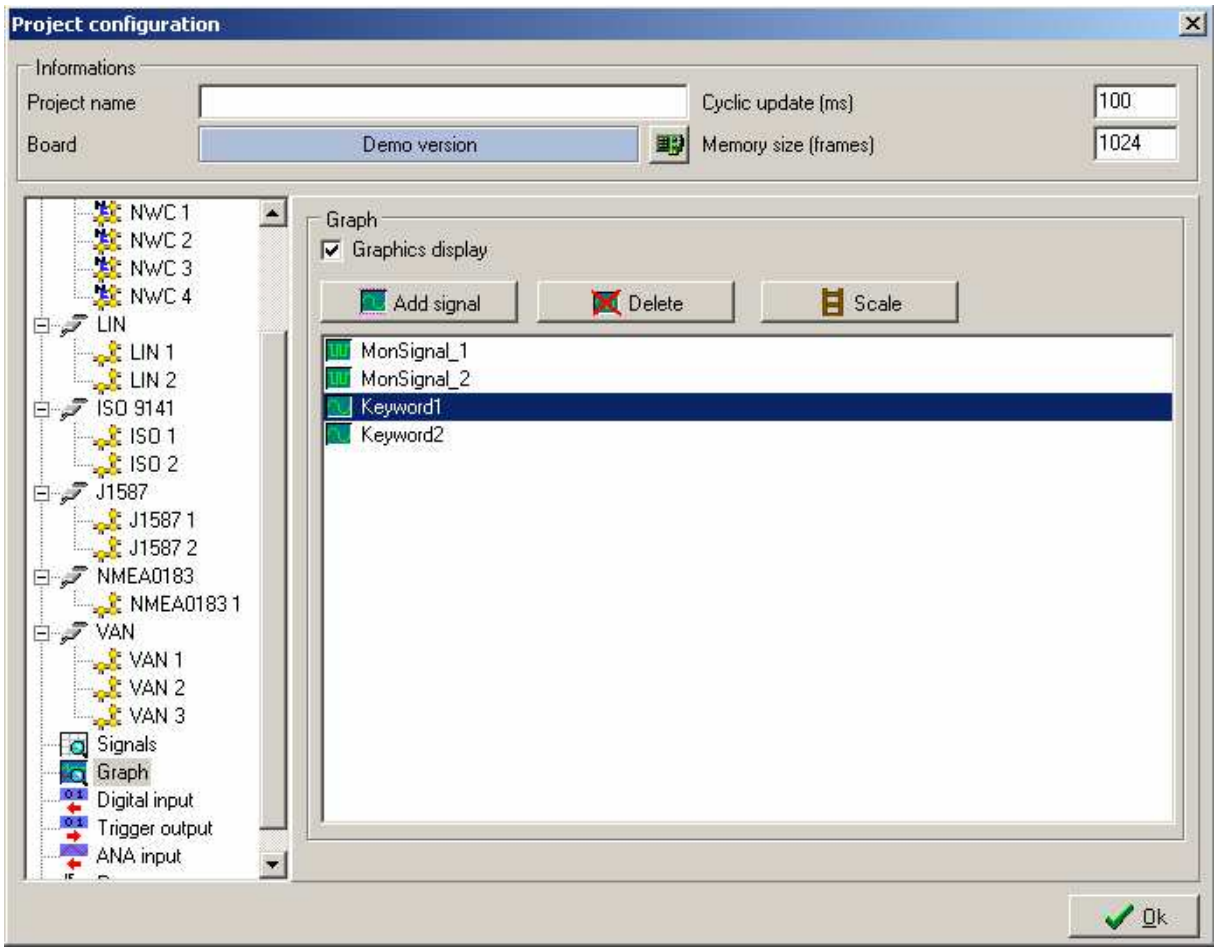
Cyclic update

The logo on the left of the value of the signal changes every time a new value of the signal is displayed.
If the logo does not change, it means that the frame containing the signal information is not being received.

3.12 Graph display of signals

When the data bases are associated to the buses, it is possible to visualize graphically the evolution of the value of the signals codified in these bases.

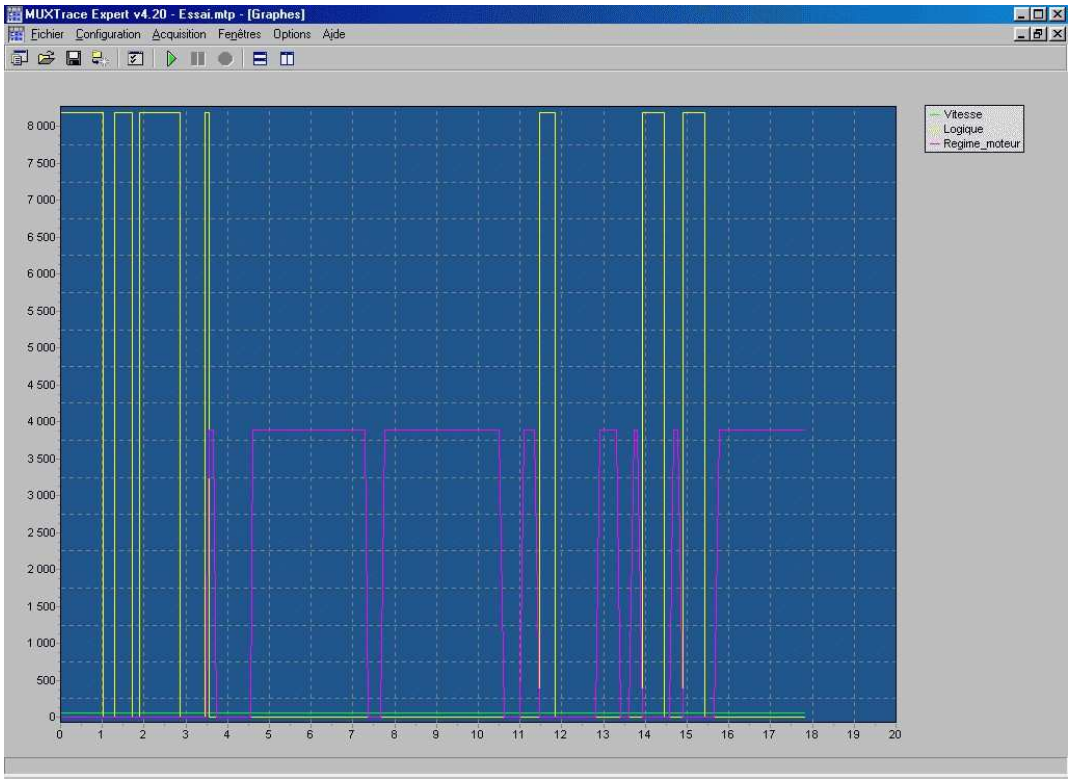
3.12.1 Creating a list of signals



It is possible to visualize up to 16 signals simultaneously. The user can also customize the scale used during visualization. By default, the scale is that stated in the data base.

3.12.2 Visualizing the signals

When the program is executed, a graph window opens to display the signals.



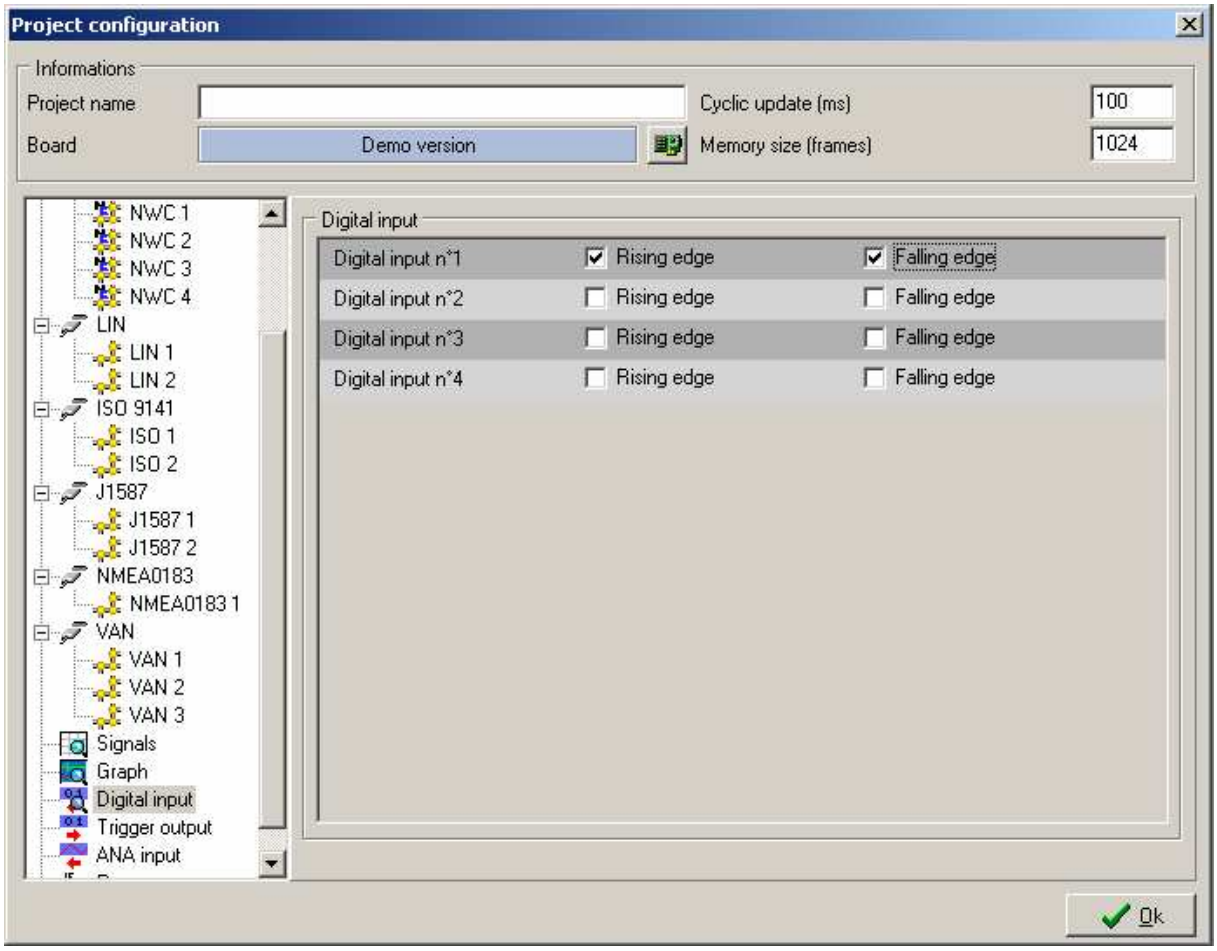
Cyclic update The update frequency of the graph is defined in the project parameters.

Axis X : Time in seconds

Axis Y : Black : Minimum and maximum scale of all signals (double click on the legend)
Green, yellow, red or blue : scale that adapts to the signal (one click on the legend)

3.13 Digital Inputs

MuxTrace allows the surveillance of digital inputs present on all network access boards of the range of EXXOTEST products.



Rising edge Only those inputs with a rising edge will be displayed in the visualization windows.

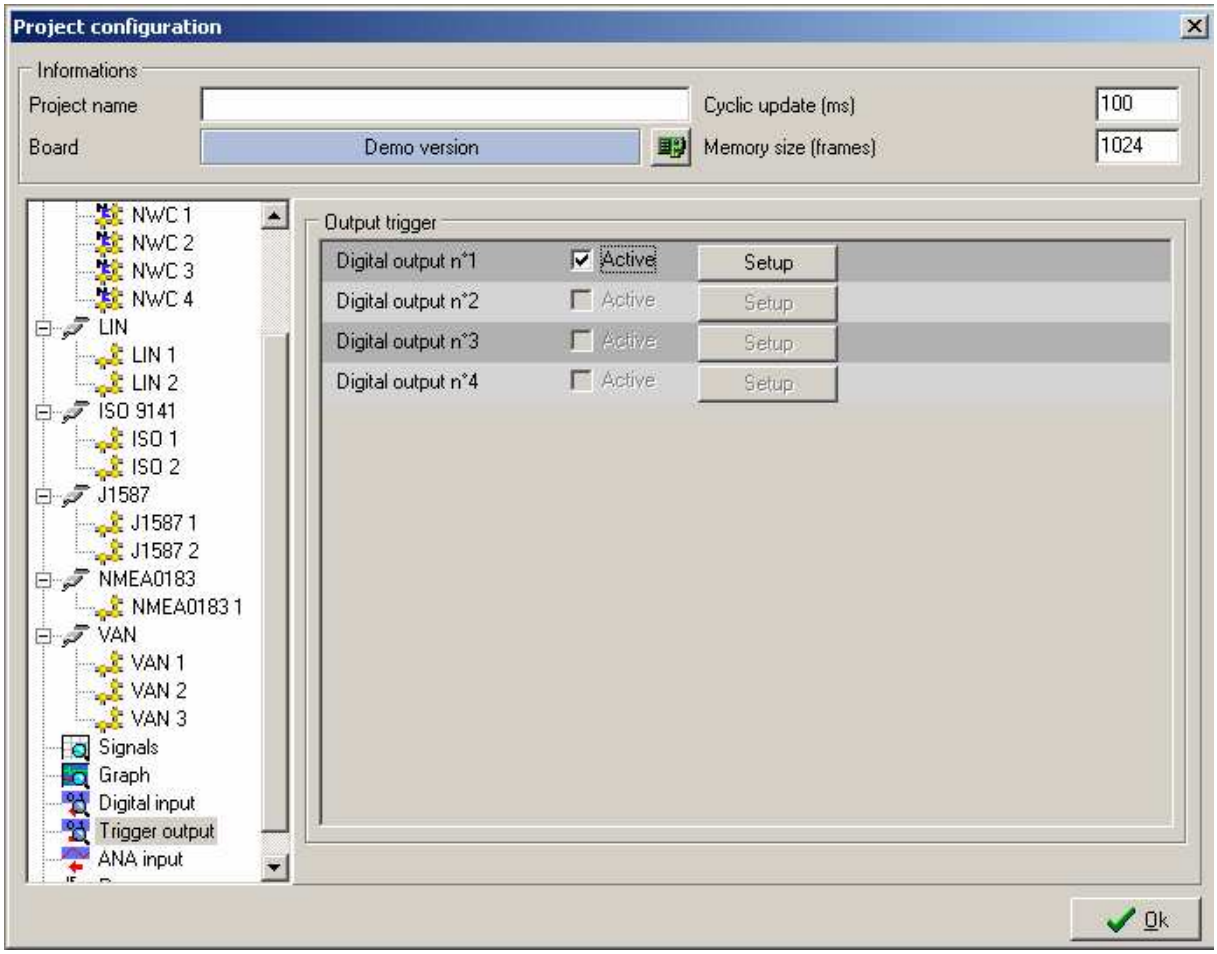
Falling edge Only those inputs with a falling edge will be displayed in the visualization windows.

When a rising or falling edge has been detected, all visualization windows receive a list of digital inputs whose status has changed.

3.14 Digital Outputs

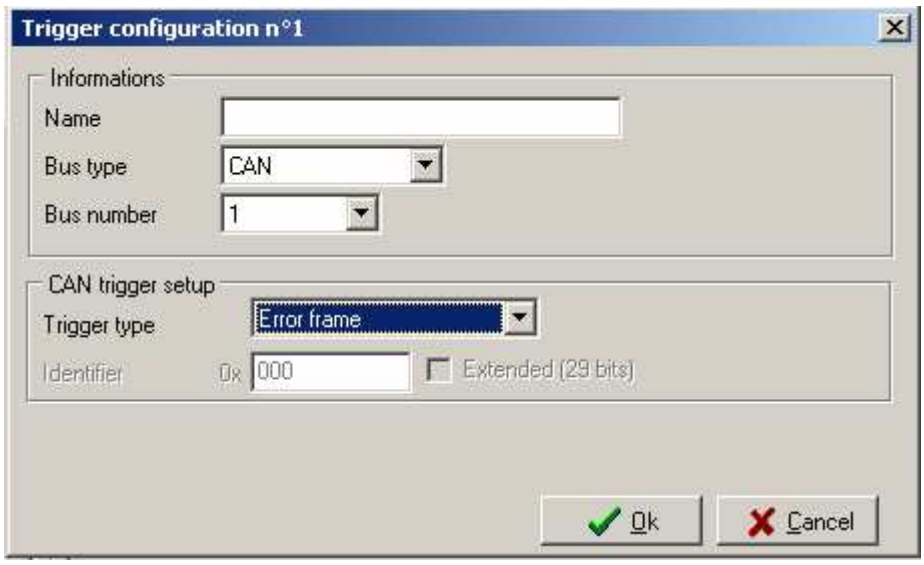
3.14.1 Output triggering

MuxTrace allows the user to activate a digital output present on all network access boards of the range of EXXOTEST products. The aim of this triggering (positive impulse of a few microseconds) is to synchronize an external tool with an event that takes place in the network.



Active An impulse is generated in the output when the configured event is detected (see material's installation guide to connect the output)

3.14.2 Configuring the triggering condition



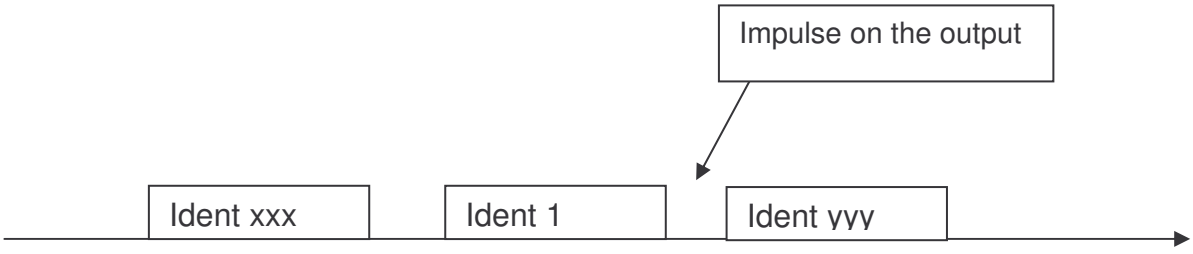
Frame mode Selection of network type

Bus Number of bus on which the event is detected

Type of event CAN identifier

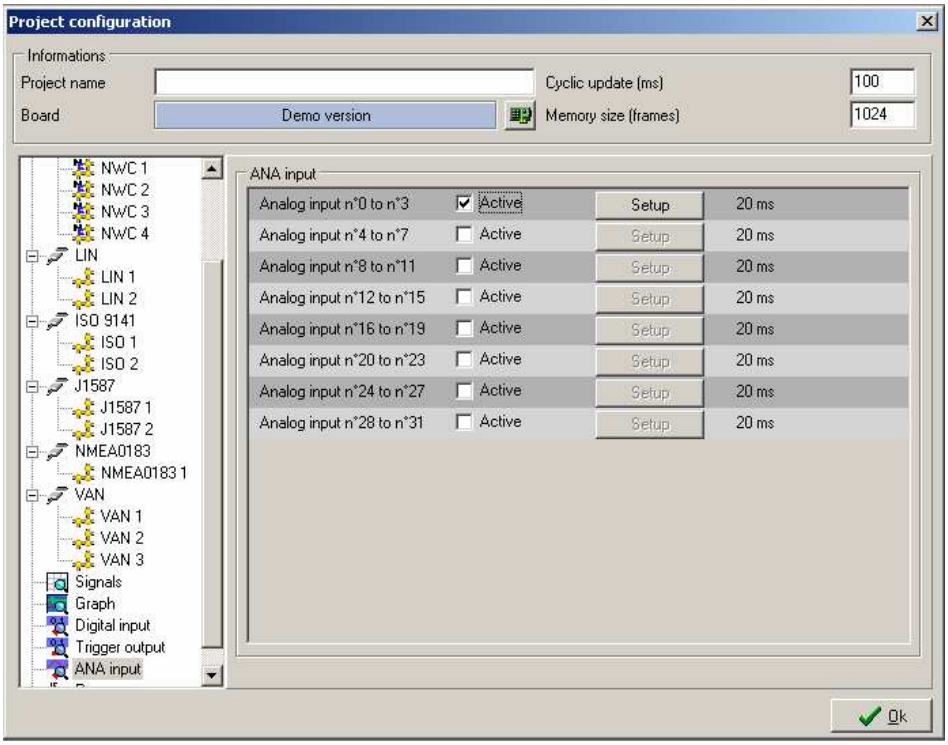
Example

Identifier 1 is chosen as a triggering condition



3.15 Analog inputs (ANA)

MuxTrace allows the surveillance of analog inputs on all network access boards of the range of EXXOTEST products. The value of these inputs can be correlated with the numerical information transiting the networks.



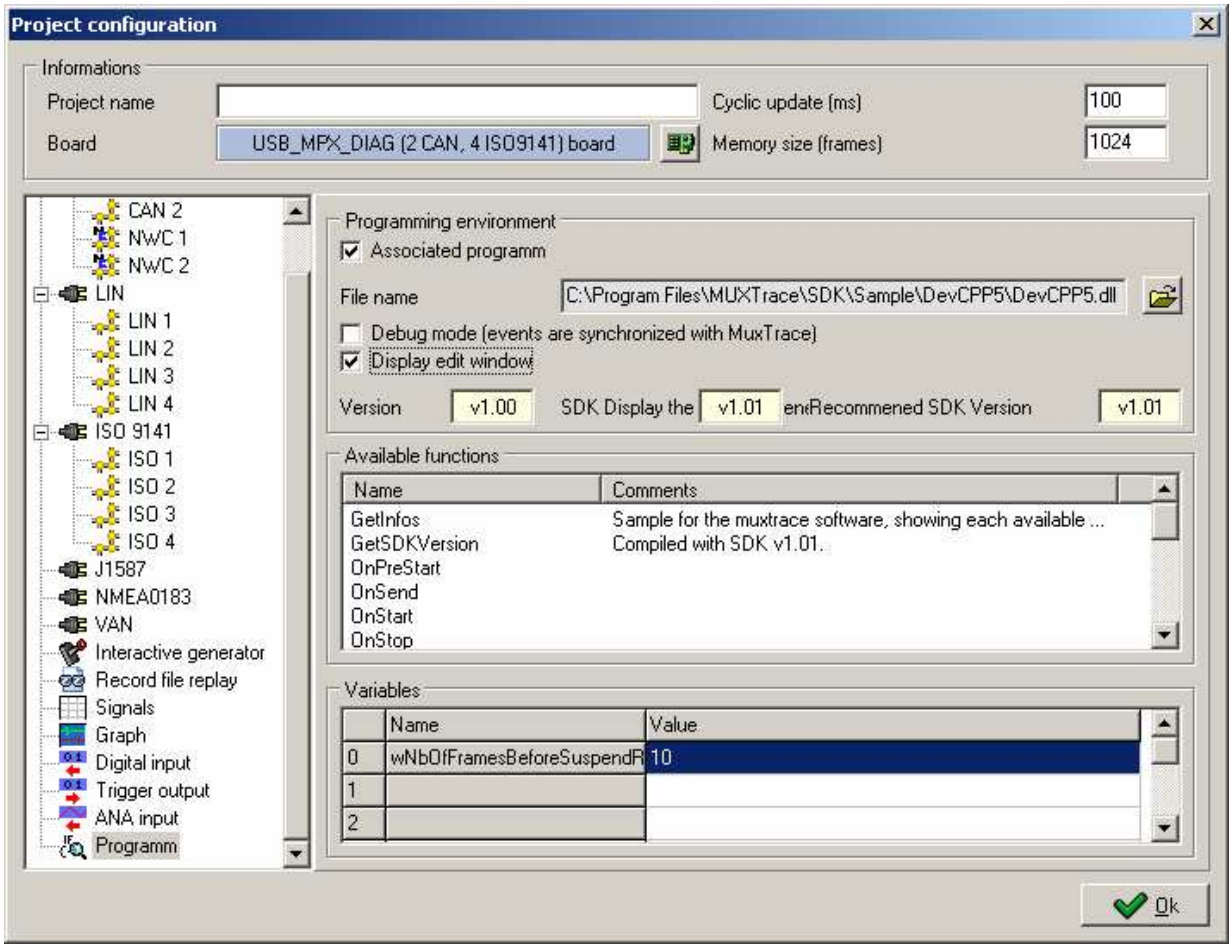
Input ANA n°x to n°y Select the list of analog inputs to go back to in the trace window.

The configuration parameters are an identifier and a periodicity. Thanks to their information, analog data are received in a way similar to a network message. It is necessary to use a data base according to the identifiers chosen (see example of ADC.DBC file)

3.16 Programming module

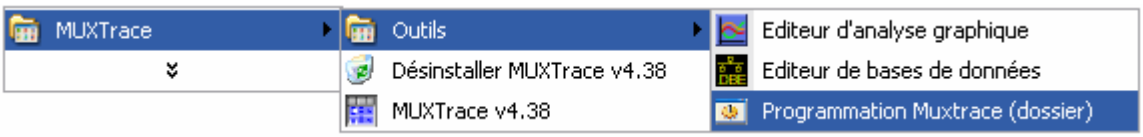
The module of programming allows the user to create his own program inside the MuxTrace environment. This program (DLL : dynamic library link), will permit to the user to personalize the MuxTrace with for example:

- create a sending scenario
- display user's text and values into the edit window
- start a recording on a particular condition
- to schedule asc file replay
- …



3.16.1 Starter Development Kit

The MuxTrace repertory contains a sub repertory with the SDK for develop a MuxTrace DLL.



The SDK contains the following files and repertories:

Sample	Repertory containing samples of MuxTrace DLL.
Prgmux.cpp	File containing the definition of the functions and the reserved entrance points. Do not modify.
Prgmux.h	File containing the declarations of the functions and the entrance points reserved. Do not modify.
PrgMux_Skeleton.cpp	File containing the skeleton and the list of the points entered user. To recopy before modifying.

3.16.2 Library entry points

Muxtrace calls the DLL at various entrance points, these calls can be synchronized or not:

- If debug mode is checked off, the entrance points are synchronized to the display functions of MuxTrace. The performances are lesser.
- If debug mode is not checked off, the entrance points are not synchronized. The performances are optimized.

User's List of entrance points

3.16.2.1 GetInfos : DLL Information

Prototype : int GetInfos(char *szInfos, int *iVersion)

Parameters: szInfos = Character strings receiving information relating to the DLL.
iVersion = DLL version : BCD coding (ex : 0x120 = v1.20).

Description Muxtrace calls this function when the DLL is loaded, then posts this information in the window of the programming module.

3.16.2.2 OnStart : Starting of acquisition

Prototype : int OnStart(void)

Description : Muxtrace calls this function when the user starts acquisition

3.16.2.3 OnStop : Stop of acquisition

Prototype : int OnStop(void)

Description : MuxTrace calls this function when the user stops acquisition

3.16.2.4 OnGetVariableName : Definitions of the user variables

Prototype : int OnGetVariableName(unsigned short wIndex, char *szVariable, char *szDefaultValue)

Parameters: wIndex = index of the variable to be defined. From 0 to 15.
szVariable = Variable name
szDefaultValue = Variable value.

Codes retour : Return value : 0 if succeed.

Description : Muxtrace calls this function when the DLL is loaded, then displays the list of variables in the window of programing module. This function is called as much as it returns a value different from 0.

3.16.2.5 OnSetVariableValue : Initialization of the user variables

Prototype : int OnSetVariableValue(const unsigned short wIndex, const char *szVariable, const char *szValue)

Parameters: wIndex = Variable index. From 0 to 15.
szVariable = Variable name.
szValue = Value to be assigned to this variable.

Description : MuxTrace calls this function at the time of the initialization of acquisition.

3.16.2.6 OnKey : When the user press a key

Prototype : int OnKey(int lKey)

parameter : lKey = Key code (symbol VK_xxx)

Description : MuxTrace calls this function when the user presses a key

3.16.2.7 OnTimer : every milliseconde

Prototype : OnTimer(DWORD dwTimer)

parameter: dwtimer = Time stamp in ms since the starting of acquisition

Description : MuxTrace calls periodically this function : every millisecond

3.16.2.8 OnCanEvent: On a reception of a CAN event

Prototype : OnCanEvent (tCanEvent *hCanEvent)

parameter: hCanEvent = Pointer on a structure containing the type of CAN event (see file REFMUX.H)
Description : MuxTrace calls this function when a CAN event happens. This event corresponds to :

- On receipt of a message
- On End of message transmission
- On Error detection

3.16.2.9 OnVanEvent: On a reception of a VAN event

Prototype : OnVanEvent (tVanEvent *hVanEvent)

parameter: hVanEvent = Pointer on a structure containing the type of VAN event (see file REFMUX.H)

Description : MuxTrace calls this function when a CAN event happens. This event corresponds to :

- On receipt of a message
- On End of message transmission
- On Error detection
- On End of transmission in error

3.16.2.10 OnLinEvent: On a reception of a LIN event

Prototype : OnLinEvent (tLinEvent *hLinEvent)

parameter: hLinEvent = Pointer on a structure containing the type of LIN event (see file REFMUX.H)

Description : MuxTrace calls this function when a LIN event happens. This event corresponds to :

- On receipt of a message
- On End of message transmission
- On Error detection
- On End of transmission in error

3.16.2.11 OnIsoEvent: On a reception of a ISO9141 event

Prototype : OnIsoEvent (tIsoEvent *hIsoEvent)

parameter: hIsoEvent = Pointer on a structure containing the type of ISO event (see file REFMUX.H)

Description : MuxTrace calls this function when an ISO event happens. This event corresponds to :

- On receipt of a message
- On End of message transmission
- On Error detection
- On End of transmission in error

3.16.2.12 OnReplayFrame : On a replay frame

Prototype : int OnReplayFrame(ST_PRGMUX_ONREPLAYFRAME *hOnReplay, unsigned short &wReplayThisFrame, unsigned short &wSuspendReplay)

Parameters: hOnReplay = Replay message (see file PRGMUX.H)
wReplayThisFrame = if value = 0 this message will not be send.
wSuspendReplay = if value = 0 replay will be suspended.

Description : Muxtrace calls this function just before to send a replay message. It is thus possible to schedule the replay and to filter messages. You must activated "by program" in the replay configuration window.

3.16.2.13 OnTriggerReplay : Trigger present in a asc file

Prototype : int OnTriggerReplay(ST_PRGMUX_ONTRIGGERREPLAY *hOnTrigger, unsigned short &wSuspendReplay)

Arguments : hOnTrigger = Trigger definition (see file PRGMUX.H)
wSuspendReplay = if value = 0 replay is suspended.

Description : Muxtrace calls this function when a trigger event is read from asc file. You must activated "by program" in the replay configuration window.

Reserved List of entrance points

3.16.2.14 GetSDKVersion : SDK Version

Prototype : int GetSDKVersion(void)

Codes retour : Return SDK version used for create the DLL. BCD coding (ex : 0x230 = v2.30).

Description : Calls this function when the DLL is loaded, thus allowing MuxTrace to compare its kit of development with that used by the DLL, in order to prevent possible conflicts being able to occur when that the kit used by the DLL is more recent than that of MuxTrace.

3.16.2.15 OnPreStart: Initialization before starting of acquisition

Prototype : OnPreStart (void)

Description : MuxTrace calls this function on start of acquisition but before starting of the communication with the networks. This event is useful to carry out initializations of shared information between MuxTrace and the DLL.

3.16.2.16 OnSend: management of the queue between the DLL and MuxTrace

Prototype : OnSend(tExxoFifoMsg *hCurExxoFifoMsg)

Parameter: hCurExxoFifoMsg = Pointer on a structure containing the actions to be transmitted towards MuxTrace
Description : MuxTrace calls this function periodically to detect the actions to be carried out. All the actions (sending, display, trigger...) are stored in a queue, which is unqueued at each call of this function.

3.16.3 Accessible functions since the library

3.16.3.1 DisplayMsg: Printing a text in the window of edition

Prototype : DisplayMsg (char *szText)

Parameter : szText = String to display (max : 1024 bytes)

Return: STATUS_OK if succeed
Other value if failed

Description : This function allows to print the text in the window of edition.

3.16.3.2 CanSendMsg: Sending a CAN message

Prototype : tMuxStatus CanSendMsg(unsigned short wCard, unsigned short wBus, tCanMsg *hCanMsg)

Parameter: see file REFMUX.H (document DLL-MUX-CAN)

Return: STATUS_OK if succeed
Other value If failed

Description : This function allows to send a data message or a request of distant transmission.

3.16.3.3 LinSendMsg: Sending LIN message

Prototype : tMuxStatus LinSendMsg(unsigned short wCard, unsigned short wBus, tLinMsg *hLinMsg)

Parameter: see file REFMUX.H (document DLL-MUX-LIN)

Return: STATUS_OK if succeed
Other value if failed

Description : This function allows to transmit a message on the bus CAN.

3.16.3.4 IsoSendMsg: Sending a ISO9141 message

Prototype : tMuxStatus IsoSendMsg(unsigned short wCard, unsigned short wBus, tIsoMsg *hIsoMsg)

Parameter: see file REFMUX.H (document DLL-MUX-ISO)

Return : STATUS_OK if succeed
Other value if failed

Description : This function allows to transmit a message on the bus ISO9141.

3.16.3.5 Iso14230SendMsg : Sending a ISO14230 message

Prototype : tMuxStatus Iso14230SendMsg(unsigned short wCard, unsigned short wBus, tIso14230Msg *hIso14230Msg)

Parameters: see file REFMUX.H (document DLL-MUX-ISO)

Codes retour : STATUS_OK if succeed
Other value if failed

Description : This function allows to transmit a message on the bus ISO14230.

3.16.3.6 VanSendMsg: Emission d'un message sur le bus VAN

Prototype : tMuxStatus VanSendMsg(unsigned short wCard, unsigned short wBus, tVanMsg *hVanMsg)

Parameter: See file REFMUX.H (document DLL-MUX-VAN)

Return: STATUS_OK if succeed
Other value if failed

Description : This function allows to transmit a message on the bus VAN.

3.16.3.7 IOSetOutput: Activation of logic outputs

Prototype : tMuxStatus IOSetOutput(unsigned short wCard, unsigned short wOutputValue, unsigned short wOutputMask)

Parameter: See file REFMUX.H (document DLL-MUX-CAN)

Return: STATUS_OK if succeed
Other value if failed

Description : This function allows the user to activate the logic outputs on the card.

3.16.3.8 LinSetSleepMode : Send a master request frame for force all slaves into sleep mode

Prototype : tMuxStatus LinSetSleepMode(unsigned short wCard, unsigned short wBus)

Parameters: See file REFMUX.H (document DLL-MUX-CAN)

Codes retour : STATUS_OK if succeed
Other value if failed

Description : Send a master request frame for force all slaves into sleep mode.

3.16.3.9 LinSetWakeUpMode : Request a wake up

Prototype : tMuxStatus LinSetWakeUpMode(unsigned short wCard, unsigned short wBus)

Parameters: See file REFMUX.H (document DLL-MUX-LIN)

Codes retour : STATUS_OK if succeed
Other value if failed.

Description : This function allows a slave to request a wake-up.

3.16.3.10 Trigger : trigger for file recording

Prototype : tMuxStatus Trigger(void)

Parameter: None

Return: STATUS_OK if succeed
Other value if failed

Description : This function allows to start a recording when the window "logging file setting" is set with the trigger condition "On trigger programm".

3.16.3.11 Stop : stop acquisition

Prototype : tMuxStatus Stop(void)

Parameter: None

Return: STATUS_OK if succeed
Other value if failed

Description : This function stops the acquisition.

3.16.3.12 OpenProject : Open a MuxTrace project file

Prototype : tMuxStatus OpenProject(char *szProjectPath, unsigned short wAutoRun, unsigned short wNoSave)

Parameter: szProjectPath = Project path.
wAutoRun = if value = true : start automatically muxtrace acquisition.
wNoSave = if value = true : don't save the current project.

Codes retour : STATUS_OK if succeed
Other value if failed

Description : Stop and close the current project to open another project and to start again acquisition automatically.

3.16.3.13 SuspendReplay : Suspend replay

Prototype : tMuxStatus SuspendReplay(void)

Codes retour : STATUS_OK if succeed
Other value if failed

Description : Suspend replay. Call *ResumeReplay function to resume replay*..

3.16.3.14 ResumeReplay : Resume replay

Prototype : tMuxStatus ResumeReplay(void)

Codes retour : STATUS_OK if succeed
Other value if failed

Description : Resume a suspended replay.

3.17 Access mode to the cards

MuxTrace makes it possible to work according to two access modes:

3.17.1 Exclusive mode

The exclusive access mode makes it possible to call directly the charts and to prevent other applications from using the chart in the course of use, thus making it possible MuxTrace to obtain optimal performances.

3.17.2 Shared mode

The shared mode allows several applications to work with a same card. In this mode, you can spy, with Muxtrace, others applications working with the same card.

MuxServer setup is available on the MuxTrace CDROM.

3.18 Expert mode

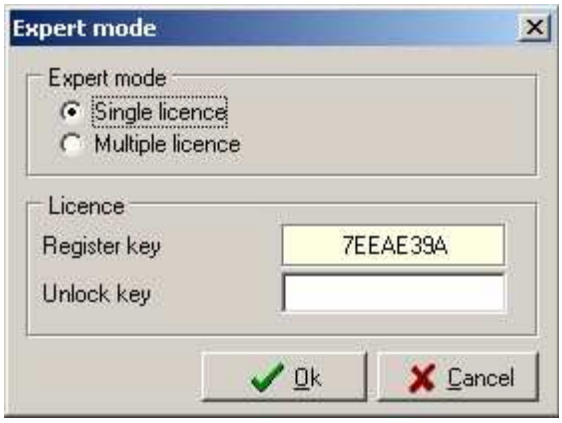
MuxTrace has an expert mode with which the following advanced functions can be used :

- managing data bases,
- display of signals present on the data bases,
- surveillance of digital inputs, triggering of outputs
- logging to a text file
- Managing the communication layer DIAG ON CAN (*ISO 15765-2*).

The expert mode can be activated in two different ways, Single license or Multiple license.

3.18.1 Expert mode Single license

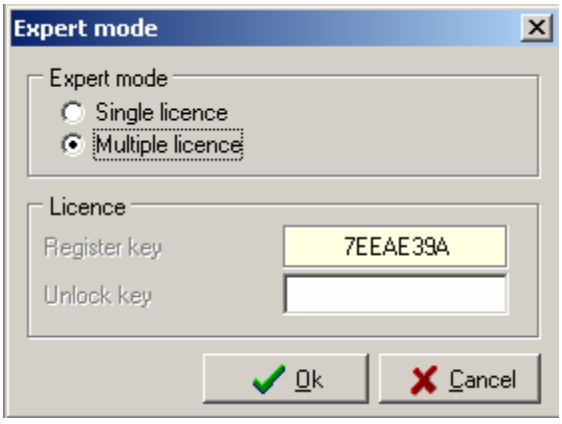
Expert mode Single license activates all advanced functions for one computer only. It works with all network access boards of the range of EXXOTEST products.



The expert mode Single license is protected by an authorization key. Contact your retailer to obtain it.

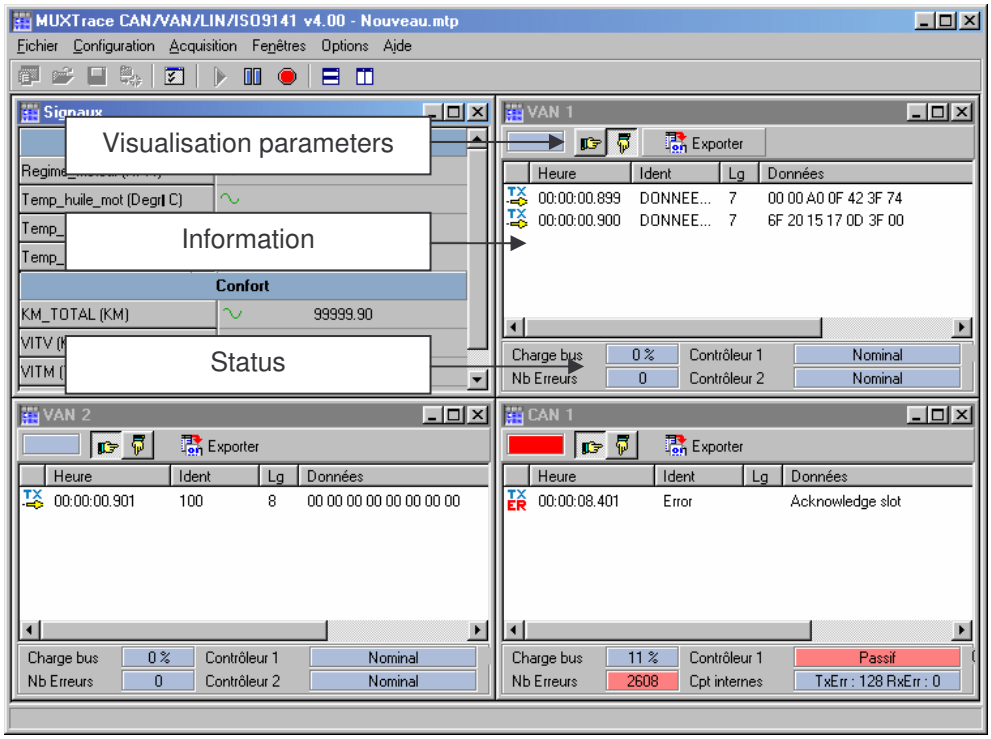
3.18.2 Expert mode Multiple license

The expert mode Multiple license activates all advanced functions for any computer providing the network access board being used has the Multiple license option.



3.19 Execution

When the programme is executed, a window opens for each different network visualized. After starting the programme, the communication board is active on the network and it can acknowledge and receive the messages coming from the networks.



3.19.1 Visualization parameters

Status


Correct operation of the display.


Correct operation of the display but PC finds it difficult to deal in time with all the messages coming from the network.


Loss of messages coming from the network. The PC's capacity to display events is too lower. Increase refreshes frequency or decrease memory size.

Display mode


Display in fixed position :each identifier has a display line
Sequential display : unroll messages on display

Exporting

Logs displayed messages on a text file for later use (printing, EXCEL type table …)

LIN only :

- Sending a wake-up call
- Sending a stand-by message

3.19.2 Information window

The information window displays in real time the messages transiting the network.

Dir

End of message transmission
Reception of a message
End of transmission with error
Reception with error
Transmission in degraded mode
Reception in degraded mode

Clock Time when message has been received (absolute, after execution has started)

Ident Value of the network identifier in hexadecimal (CAN, VAN, LIN) or address of source -> address of target for ISO9141

Lg Length of message data

Data Content of data in hexadecimal

Period Display in fixed position: time difference in ms, in relation to the last identical identifier displayed.
Display in sequential position : time difference in ms, in relation to the previous message (no matter the value of the identifier is).

Svc Message service
For the CAN bus:
- **DA** : Frame of data
- **DR** : Frame of remote transmission (RTR=1)

For the NWC bus:
- **DA** : Frame of data

For the LIN bus:
- **DA** : Frame of data or transmission of a writing frame
- **RR** : Transmission of a reading request
- **IFR** : Transmission of an in frame response
- **WK** : Reception of a wake-up call

For the VAN bus:
- **DA** : Frame of data
- **DAA** : Frame of data with acknowledgement
- **DRA** : Differed response with acknowledgement or reception of a demand for a response in the frame with response.
- **RRA** : Transmission of a demand for a response in the frame
- **IFR** : Transmission of a response in the frame

For the ISO9141 bus, interpretation of command code or response interpretation. For example :
- **STCOM** : Start communication
- **STCOMPR** : Positive response to the start communication

request
- **NAK ServiceNotSupported** : Negative response

Sender Name of transmitting ECU in the frame

3.19.3 Single click of the mouse

In pause or stop mode, one click of the mouse on a message allows the user to set all windows of different networks at the same time as the event on which the click has been made. This allows the user to visualize the status of the data of other networks at that moment.

3.19.4 Double click of the mouse

If a data base is associated, a double click on the message allows the user to visualize all the data carried by this message in real time (in fixed display position only) or while the trace is seen in pause mode or when it stops.

3.19.5 Drag&Drop
Muxtrace allows user to drag signals from the trace window to the data window or to the graphical window.

3.19.5 Sorting out messages

With display in fixed position, the selection of the top part of the column, allows the user to sort out the identifying columns, the period and the senders of the message.

3.19.6 Status

Charge Occupation charge of the network. The charge is calculated every second

No errors Counter of number of errors observed after execution has started

Controller 1 For the CAN bus:
- Controller status : ACTIVE, PASSIVE or BUS OFF

For the LIN bus:
- Communication status : NOMINAL, DEGRADED or BUS IDLE

For the VAN network:
- Communication status of controller 1 : Nominal, communication in data, communication in datab, serious error.

Controller 2 For VAN network only :
- Communication status of controller 2 : Nominal, communication in data, communication in datab, serious error.

Internal cntrs For CAN and LIN networks :
- Value of internal counters of protocol controller, which manages the status of the controller (ACTIVE, PASSIVE and BUS OFF)

Comm. For bus CAN low speed – fault tolerant :
- Bus status NOMINAL or DEGRADED

List of versions

Version	Date	Author	Modifications
01	22/06/2001	P. CHAZOT	Creating a document
02	05/10/2001	P. CHAZOT	Adding a LIN network
03	14/11/2001	P. CHAZOT	Adding a LIN identifier
04	05/09/2002	P. CHAZOT	Adding ISO9141 functions
05	30/01/2002	A.GAMBIER	Adding data bases, signals, digital inputs, Diag On Can communication layers (*ISO 15765-2*)
06	15/05/2003	P. CHAZOT	Adding logging on a text file, output triggering, re-starting after bus off, graph of signals
07	06/11/2003	P. CHAZOT	Adding analog inputs Adding decoding of a message in real time
08	09/12/2003	P.CHAZOT	Adding transmitting ECUs
09	08/06/2004	P.CHAZOT	MuxTrace V 4.30 - Adding filters (CAN, VAN) - Adding file recording functions - Adding programming functions
10	10/05/2005	A.GAMBIER	MuxTrace v4.40 - Adding bite rate detection - Adding interactive generator - Adding replay module - Adding new functionalities to the programming module