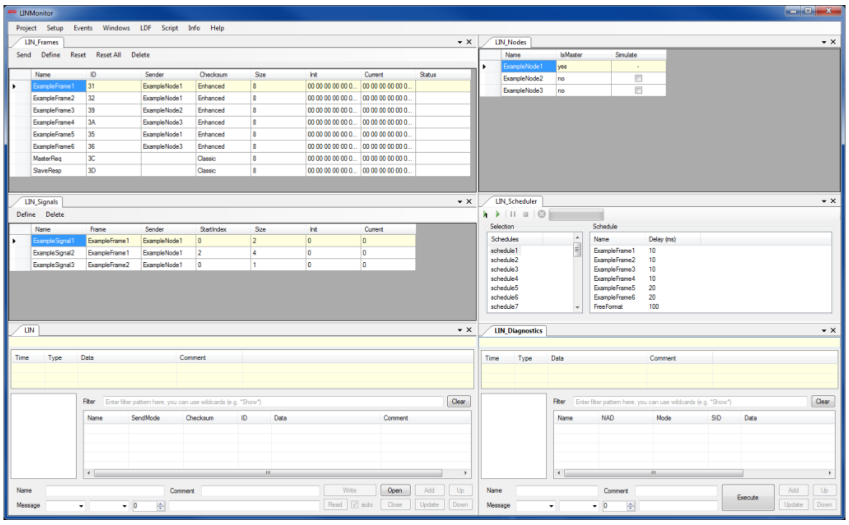
ELEKTRONIK GMBH

GET IN touch
WITH SENSITIVE TESTING

Manual

Toolmonitor LIN



Name	ID	Sender	Checksum	Size	Init	Current	Status
ExampleFrame1	31	ExampleNode1	Enhanced	8	00 00 00 00 00 0...	00 00 00 00 00 0...	
ExampleFrame2	32	ExampleNode1	Enhanced	8	00 00 00 00 00 0...	00 00 00 00 00 0...	
ExampleFrame3	39	ExampleNode2	Enhanced	8	00 00 00 00 00 0...	00 00 00 00 00 0...	
ExampleFrame4	3A	ExampleNode3	Enhanced	8	00 00 00 00 00 0...	00 00 00 00 00 0...	
ExampleFrame5	35	ExampleNode1	Enhanced	8	00 00 00 00 00 0...	00 00 00 00 00 0...	
ExampleFrame6	36	ExampleNode3	Enhanced	8	00 00 00 00 00 0...	00 00 00 00 00 0...	
MasterReq	3C		Classic	8	00 00 00 00 00 0...	00 00 00 00 00 0...	
SlaveResp	3D		Classic	8	00 00 00 00 00 0...	00 00 00 00 00 0...	

Name	IsMaster	Simulate
ExampleNode1	yes	-
ExampleNode2	no	☐
ExampleNode3	no	☐

Name	Frame	Sender	StartIndex	Size	Init	Current
ExampleSignal1	ExampleFrame1	ExampleNode1	0	2	0	0
ExampleSignal2	ExampleFrame1	ExampleNode1	2	4	0	0
ExampleSignal3	ExampleFrame2	ExampleNode1	0	1	0	0

Schedules
schedule1
schedule2
schedule3
schedule4
schedule5
schedule6
schedule7

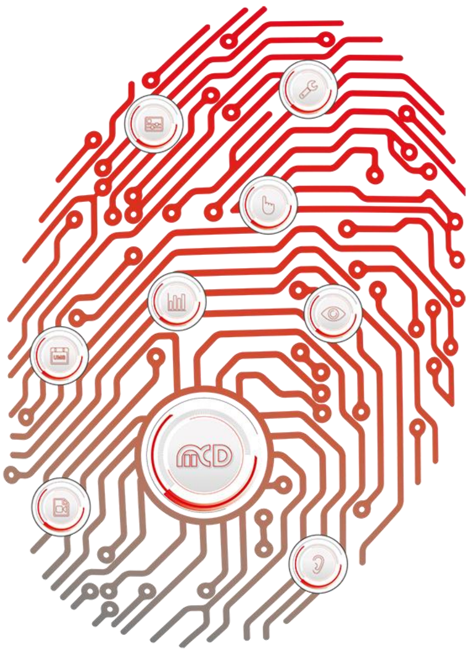
Name	Delay (ms)
ExampleFrame1	10
ExampleFrame2	10
ExampleFrame3	10
ExampleFrame4	10
ExampleFrame5	20
ExampleFrame6	20
FreeFormat	100

Time	Type	Data	Comment

Name	SendMode	Checksum	ID	Data	Comment

Time	Type	Data	Comment

Name	NAD	Mode	SID	Data



Softline

Modline

Conline

Boardline

Avidline

Pixline

Application

MCD Elektronik GmbH
Hoheneichstr. 52
75217 Birkenfeld
Tel. +49 (0) 72 31/78 405-0
Fax +49 (0) 72 31/78 405-10
info@mcd-elektronik.de
www.mcd-elektronik.com
HQ: Birkenfeld
Managing CEO: Bruno Hörter
Register Court Mannheim
HRB 505692

Template version: 5.4 / 2018-04-09
V1.4 2018-09-12 / BM (TS)

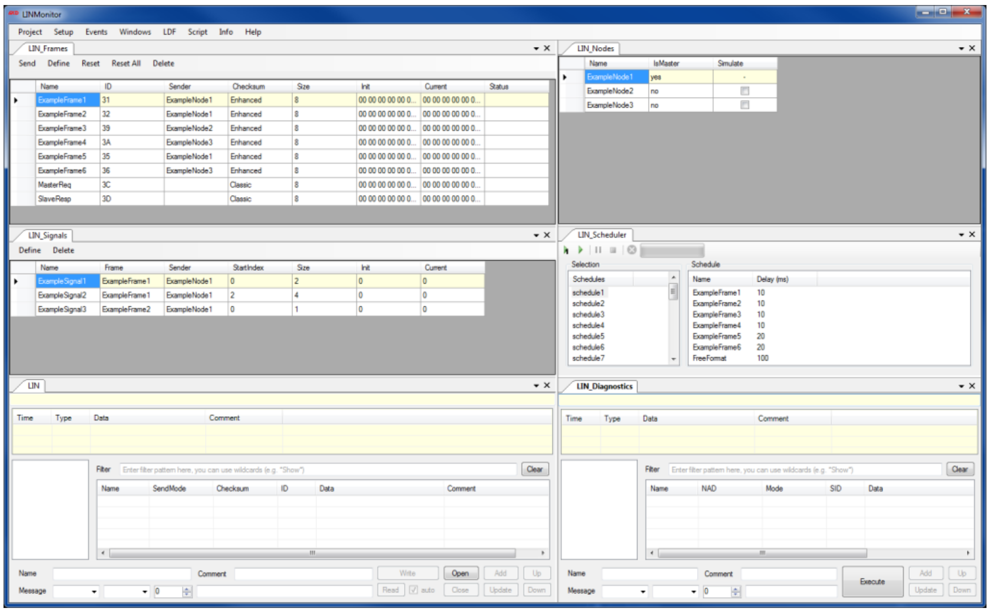
Table of Contents

1. General

The Toolmonitor LIN is used to send and receive LIN frames. These can be loaded from LDF files and sent either individually or in schedules.

The Toolmonitor provides the following functionality:

- Transmission and receipt of LIN frames
- Loading of frames, signals, schedules and notes from an LDF file
- Manual addition of additional frames and signals
- Execution of schedules
- Transmission of individual frames
- Transmission and evaluation of diagnostic frames
- Diagnostic frames can be transmitted or received while a schedule is being executed
- Toolmonitor can act either as master or as slave
- Simulation of slave nodes
- Editor to create and edit LDF files
- Control of multiple LIN interfaces in a single tool



Name	ID	Sender	Checksum	Size	Init	Current	Status
ExampleFrame1	31	ExampleNode1	Enhanced	8	00 00 00 00 00 0...	00 00 00 00 00 0...	
ExampleFrame2	32	ExampleNode1	Enhanced	8	00 00 00 00 00 0...	00 00 00 00 00 0...	
ExampleFrame3	39	ExampleNode2	Enhanced	8	00 00 00 00 00 0...	00 00 00 00 00 0...	
ExampleFrame4	3A	ExampleNode3	Enhanced	8	00 00 00 00 00 0...	00 00 00 00 00 0...	
ExampleFrame5	35	ExampleNode1	Enhanced	8	00 00 00 00 00 0...	00 00 00 00 00 0...	
ExampleFrame6	36	ExampleNode3	Enhanced	8	00 00 00 00 00 0...	00 00 00 00 00 0...	
MasterReq	3C		Classic	8	00 00 00 00 00 0...	00 00 00 00 00 0...	
SlaveResp	3D		Classic	8	00 00 00 00 00 0...	00 00 00 00 00 0...	

Name	IsMaster	Simulate
ExampleNode1	yes	-
ExampleNode2	no	☐
ExampleNode3	no	☐

Name	Frame	Sender	StartIndex	Size	Init	Current
ExampleSignal1	ExampleFrame1	ExampleNode1	0	2	0	0
ExampleSignal2	ExampleFrame1	ExampleNode1	2	4	0	0
ExampleSignal3	ExampleFrame2	ExampleNode1	0	1	0	0

Selection

Schedules
schedule1
schedule2
schedule3
schedule4
schedule5
schedule6
schedule7

Schedule

Name	Delay (ms)
ExampleFrame1	10
ExampleFrame2	10
ExampleFrame3	10
ExampleFrame4	10
ExampleFrame5	20
ExampleFrame6	20
FreeFormat	100

Time	Type	Data	Comment

Enter filter pattern here, you can use wildcards (e.g. "Show")

Name	SendMode	Checksum	ID	Data	Comment

Name	NAD	Mode	SID	Data

Figure 1: Toolmonitor LIN

Order number: # 153359

2. Safety Instructions

The Toolmonitor LIN is used to send and receive LIN frames. The product has been created and tested with the greatest possible care.

However, all liability for any damages resulting from use of the product is still explicitly excluded. All trademarks or service marks that appear in the documentation are subject to intellectual property law and are the property of their various owners.

3. Installation of Software

3.1. Software and Driver Installation

To be able to use the software, the “Vector Hardware” driver must be installed. To be able to control the Toolmonitor via COM, for example from TestManager CE, the COM server must first be registered.

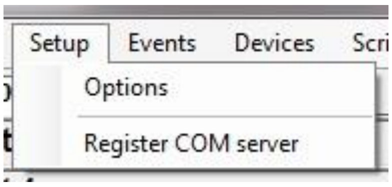


Figure 2: Registration Dialog

No additional installer is needed for the Toolmonitor.

3.2. Running the Installed Software

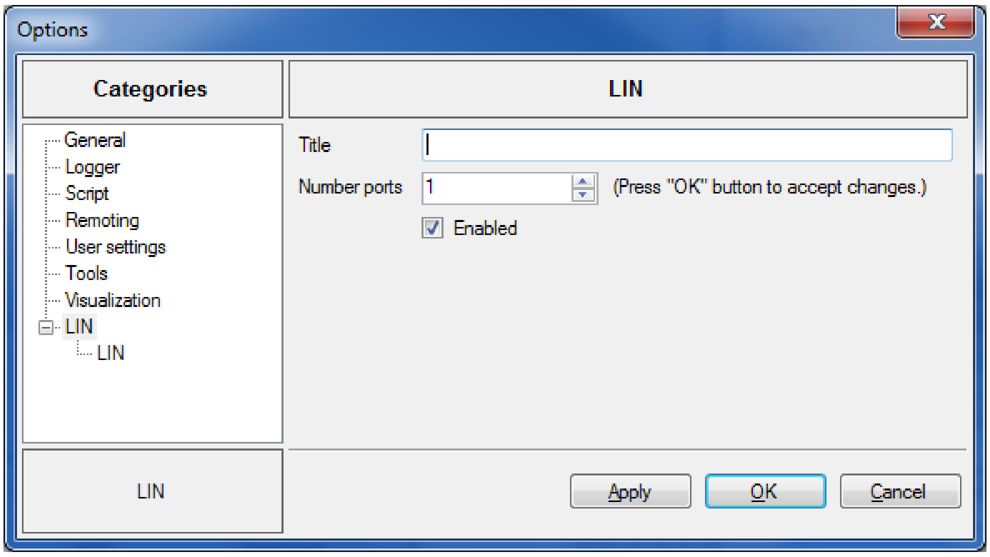
The software is started by opening “LINMonitor.exe”.

4. Introduction to the Operation

4.1. Quick Guide

The first time the software is started, the communication parameters must be defined. To do this, open the options under **Setup → Options**.

Multiple LIN modules can be created within the Toolmonitor. At least, one is needed to use the software.



Title: |

Number ports: 1 (Press "OK" button to accept changes.)

☑ Enabled

Figure 3: Creating a LIN Module

After a module has been created, the individual interfaces will already appear on the *Windows* tab.

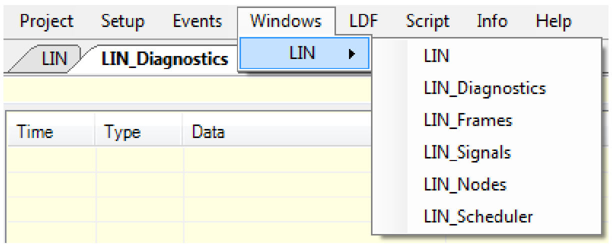
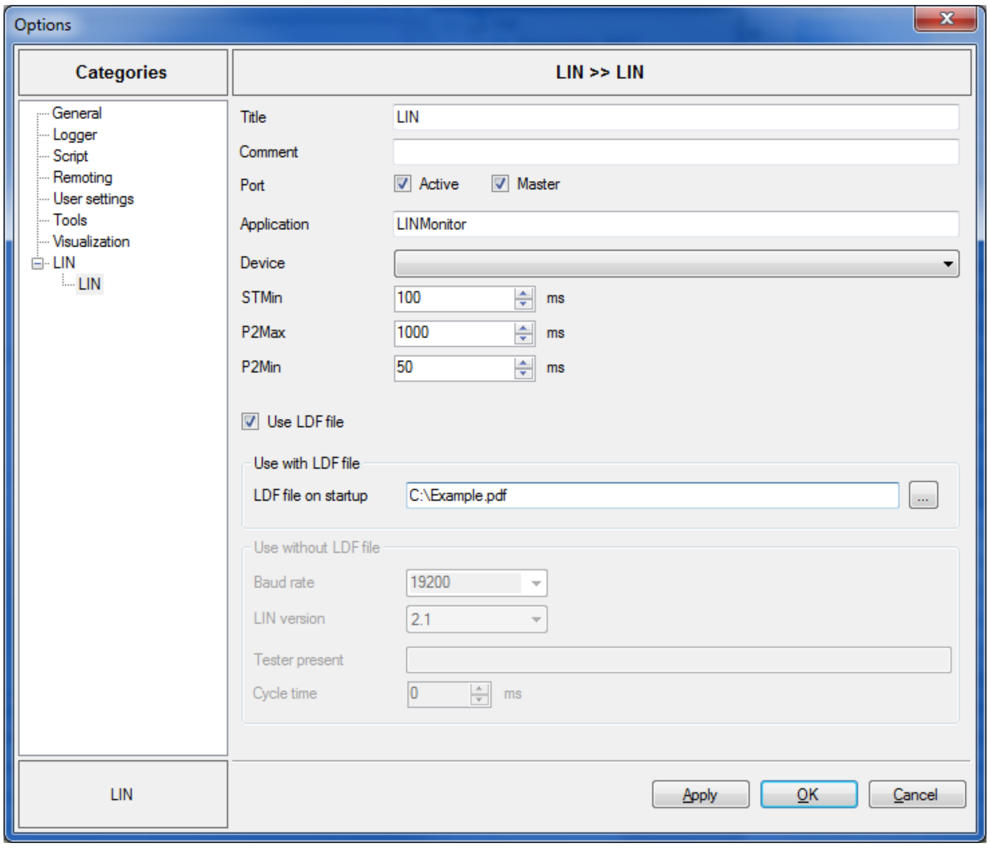


Figure 4: Module Interface Inside the Windows® Menu

In the next step, the parameters for the communication can be defined.

Here, you can assign a name to make it easier to distinguish between several interfaces. If the user wants to try out the Toolmonitor without hardware, the “Active” box should be unchecked. Otherwise, it must be checked. If the Toolmonitor should act as master, check the “Master” box.

For use with “Vector Hardware”, the device to be used must also be selected with the corresponding channel.


Options

Categories

- General
- Logger
- Script
- Remoting
- User settings
- Tools
- Visualization
- LIN
  - LIN

LIN >> LIN

Title	LIN
Comment	
Port	[✓ Active] [✓ Master]
Application	LINMonitor
Device	
STMin	100 ms
P2Max	1000 ms
P2Min	50 ms

[✓ Use LDF file]

Use with LDF file

LDF file on startup	C:\Example.pdf

Use without LDF file

Baud rate	19200
LIN version	2.1
Tester present	
Cycle time	0 ms

LIN

Figure 5: Specifying Parameters

The Toolmonitor can be used with or without an LDF file loaded. If operation with LDF is selected, a path can alternatively be created to an LDF file to be opened when the Toolmonitor starts.

If no LDF file should be used, more information will be needed for the connection, as this is normally stored in the LDF file. This includes the *baud rate* and *LIN version*. A *tester present string* and associated cycle time can optionally be defined.

When running without LDF, two nodes are automatically created. One is the master, the second the slave.

If a *tester present string* has been entered, a schedule is also created for the *tester present*. This schedule automatically starts when a response is received to a diagnostic message transmitted. If the “tester present” transmits using ID “0x3C”, a frame with ID “0x3D” is added to this schedule to evaluate the corresponding response.

The software is now prepared for the use. The individual interfaces and functions will be described in the next chapter.

5. Software Manual

This chapter describes the individual interfaces and functions of the Toolmonitor.

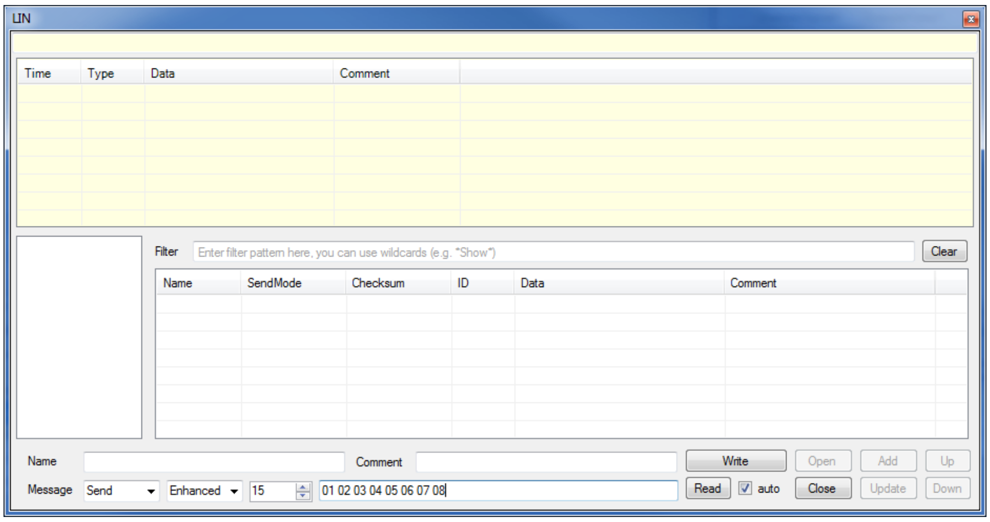
5.1. LIN Line

The *LIN Line Interface* can be used to transmit and receive LIN frames directly. It can also be used to open or close a connection. If the connection is closed, an automatic attempt to open it takes place when a message is sent.

If *Autoread* is enabled, the frames and signals created automatically update when a corresponding frame is received. If you want to read out frames manually, *Autoread* must be disabled.

To send a message, the following parameters must be specified:

- Mode: Transmission of data (master only), request of data (master only) or simulation of a slave
- Checksum: Enhanced or classic
- ID: Hex value
- Data: Up to 8 bytes, hex numbers, byte - wise separated by spaces



Filter: Enter filter pattern here, you can use wildcards (e.g. "Show")

Message: Send | Enhanced | 15 | 01 02 03 04 05 06 07 08

auto: ✓

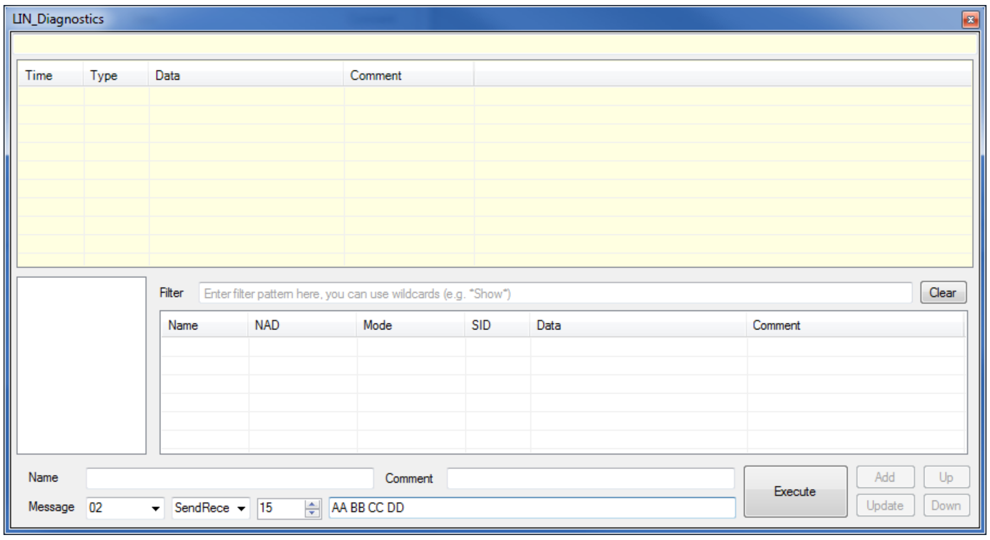
Figure 6: LIN Line Interface

5.2. Diagnostics

The *Diagnostics Interface* can be used to transmit diagnostic messages. The user does not need to worry about the division of the diagnostic message into its individual LIN frames. The software takes care of that.

To send a diagnostic message, the following parameters can be specified:

- NAD: Hex value or name of the node if an LDF file is loaded
- Mode: Data transmission only, data transmission and receipt, receive only
- Service ID: Hex value
- Daten: Hex value, byte - wise separated by spaces



Message: 02 | SendRece | 15 | AA BB CC DD

Figure 7: Diagnostics Interface

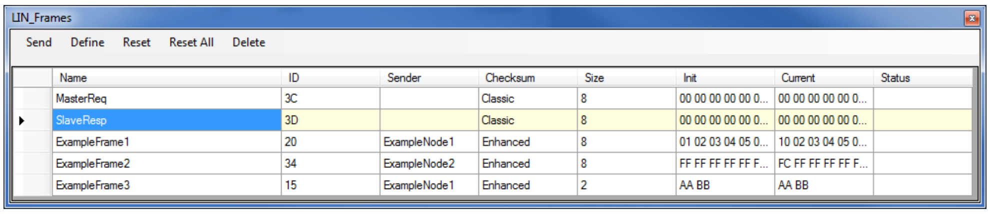
5.3. Frames

The *Frames Interface* displays the frames in a loaded LDF file.

A frame can be sent by selecting it and clicking the send button. If the sender of the frame is a master node, data will be transmitted; otherwise, they are requested. When a frame is clicked or selected, the *Node and Signal Interface* shows the records belonging to the frame in a different color.

A selected frame can be deleted via delete button. A reset overwrites the current value with the initial value.

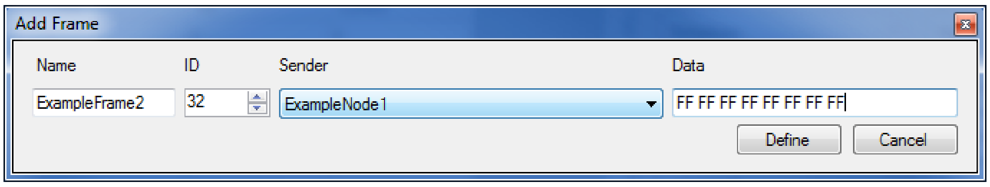
The user interface can also be used to define your own frames or edit existing ones by clicking the define button. This can be especially useful when no LDF file is loaded.



Name	ID	Sender	Checksum	Size	Init	Current	Status
MasterReq	3C		Classic	8	00 00 00 00 00 0...	00 00 00 00 00 0...	
SlaveResp	3D		Classic	8	00 00 00 00 00 0...	00 00 00 00 00 0...	
ExampleFrame1	20	ExampleNode1	Enhanced	8	01 02 03 04 05 0...	10 02 03 04 05 0...	
ExampleFrame2	34	ExampleNode2	Enhanced	8	FF FF FF FF FF F...	FC FF FF FF FF F...	
ExampleFrame3	15	ExampleNode1	Enhanced	2	AA BB	AA BB	

Figure 8: Frames Interface

Newly added or modified frames are not stored when the Toolmonitor is closed. They are used for temporary testing only. If the frames need to be kept for later, the LDF Editor can be used to create or edit an LDF file. You can find more information in the section LDF Editor.



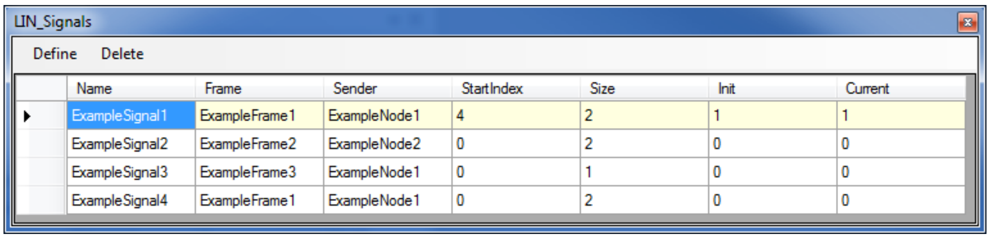
Name	ID	Sender	Data
ExampleFrame2	32	ExampleNode1	FF FF FF FF FF FF FF FF

Figure 9: Frame Processing Inside the LDF Editor

5.4. Signals

The *Signal Interface* displays the signals in a loaded LDF file. When a signal is clicked or selected, the *Node and Signal Interface* shows the records belonging to the signal in a different color. A selected signal can be deleted with the delete button.

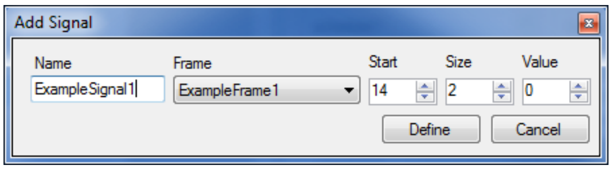
The user interface can also be used to define your own signal or edit existing ones by clicking the define button. This can be especially useful when no LDF file is loaded.



Name	Frame	Sender	StartIndex	Size	Init	Current
ExampleSignal1	ExampleFrame1	ExampleNode1	4	2	1	1
ExampleSignal2	ExampleFrame2	ExampleNode2	0	2	0	0
ExampleSignal3	ExampleFrame3	ExampleNode1	0	1	0	0
ExampleSignal4	ExampleFrame1	ExampleNode1	0	2	0	0

Figure 10: Signals Interface

Newly added or modified signals are not stored when the Toolmonitor is closed. They are used for temporary testing only. If the frames need to be kept for later, the LDF Editor can be used to create or edit an LDF file. You can find more information in the section LDF Editor.



Name	Frame	Start	Size	Value
ExampleSignal1	ExampleFrame1	14	2	0

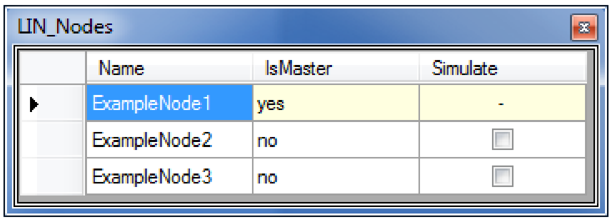
Figure 11: Signal Processing Inside the LDF Editor

5.5. Nodes

The *Nodes Interface* displays the nodes in a loaded LDF file. If no LDF file is loaded, then one master node and one slave are automatically created. The description of a node specifies whether it is a master or a slave node.

If communication is being tested with a slave that is not connected, it can be simulated using a simulation function.

If data is requested from the node, the Toolmonitor itself transmits the response with the current data in the frame table.

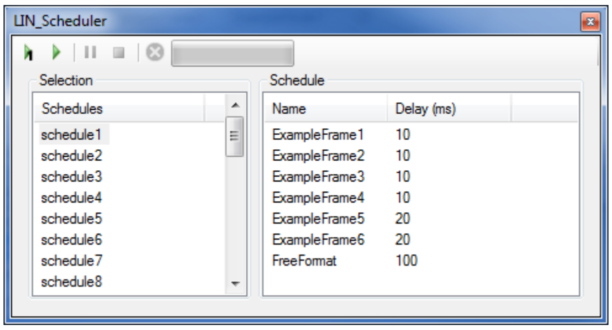

LIN_Nodes

Name	IsMaster	Simulate
ExampleNode1	yes	-
ExampleNode2	no	☐
ExampleNode3	no	☐

Figure 12: Nodes Interface

5.6. Schedules

The *Scheduler Interface* displays the schedules in a loaded LDF file. These can be started, paused, stopped and canceled from the user interface. A schedule can be executed either once or in a loop. When a schedule is canceled, it stops immediately, but when it is stopped it still runs until it ends. If no response can be received for a transmitted frame, the frame is displayed with a red background.


LIN_Scheduler

Selection

Schedules
schedule1
schedule2
schedule3
schedule4
schedule5
schedule6
schedule7
schedule8

Schedule

Name	Delay (ms)
ExampleFrame1	10
ExampleFrame2	10
ExampleFrame3	10
ExampleFrame4	10
ExampleFrame5	20
ExampleFrame6	20
FreeFormat	100

Figure 13: Scheduler Interface

5.7. LDF Files

Independently of the settings in the setup, the **LDF → Load file** menu tab can be used to load an LDF file. If operation without an LDF is selected in the setup, operation with an LDF file will now be specified.

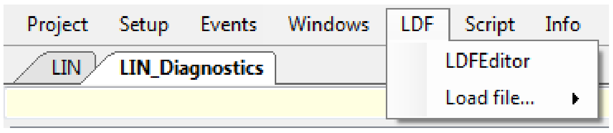
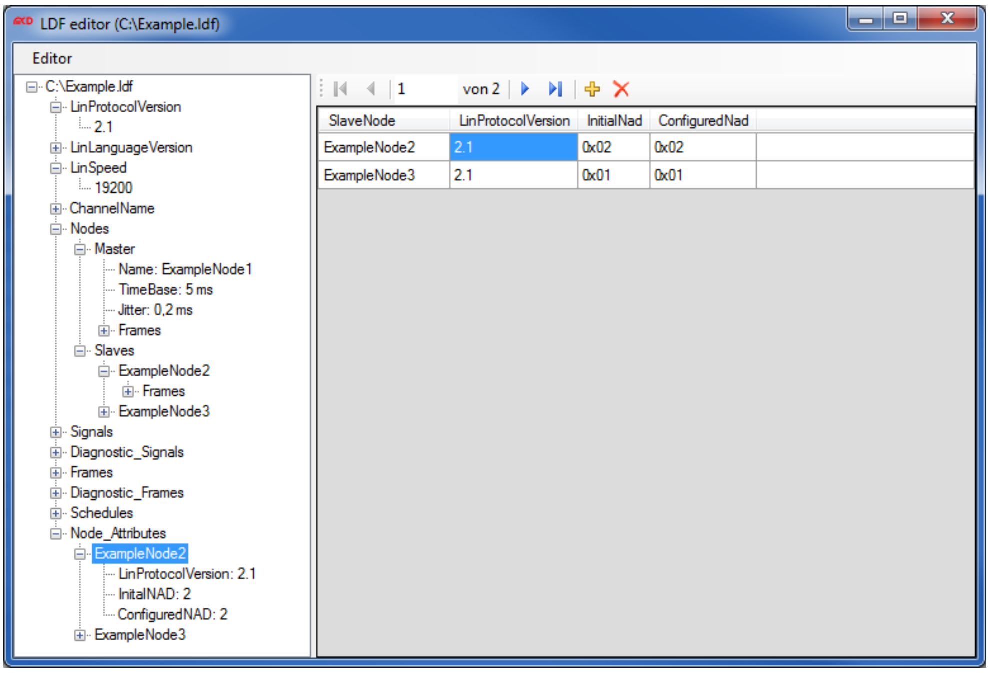


Figure 14: LDF Menu

New LDF files can be created or existing LDF files edited in the LDF Editor. It is possible to add, edit and delete frames, signals and nodes.

Note:
If a LDF file is opened and saved again, unparsed content such as comments will not be saved again.



```
C:\Example.ldf
  LinProtocolVersion
    2.1
  LinLanguageVersion
  LinSpeed
    19200
  ChannelName
  Nodes
    Master
      Name: ExampleNode1
      TimeBase: 5 ms
      Jitter: 0,2 ms
      Frames
    Slaves
      ExampleNode2
        Frames
      ExampleNode3
  Signals
  Diagnostic_Signals
  Frames
  Diagnostic_Frames
  Schedules
  Node_Attributes
    ExampleNode2
      LinProtocolVersion: 2.1
      InitalNAD: 2
      ConfiguredNAD: 2
    ExampleNode3
```

SlaveNode	LinProtocolVersion	InitialNad	ConfiguredNad
ExampleNode2	2.1	0x02	0x02
ExampleNode3	2.1	0x01	0x01

Figure 15: LDF Editor

5.8. Logging

The Toolmonitor allows several different methods to log information and communication data. The Toolmonitor LIN contains all of the standard modules of the MCD framework, like the general logging window and trace window (**Events → Logging** and **Events → Trace**). The *Virtual Interface* commands and their results are logged in the V*irtual Interface* window and communication data is logged on the LIN and diagnostics window.

In addition to these framework functionalities, a new functionality was added to the Toolmonitor to log communication data of the LIN and diagnostics interface. The logging can be started and stopped by *Virtual Interface* commands and the messages are saved in a user defined text file. Communication data can either be stored into a new file or be appended to an existing log file.

If "Appending" was selected, a new file is created when it does not yet exist. If the user wants to create a new file although there already exists a log file with the specified name, a copy of the existing file will be made and renamed, adding a timestamp to the filename.

Examples:

```
// Start logging into file C:/logfile.txt.
// If file already exists, a copy of the old file will be created.
SetValue("LIN.LogToFile", "C:/logfile.txt");
// Reads the current log file path.
string path = GetString("LIN.LogToFile");
// Start logging into file C:/logfile.txt.
// If the file already exists, new log messages will be appended.
SetValue("LIN.LogToFile.append", "C:/logfile.txt");
// Stops the logging
SetEvent("LIN.StopLogToFile");
```

The log file contains the following information:

```
Timestamp Direction (Write/Read) Layer (Frames/Diagnostics) Checksum Data Comment
```

5.9. Virtual Interface

The individual commands of the *Virtual Interface* are described in the CHM help file of the Toolmonitor.

This can be found under **Help → LIN Help**.