



Baby-LIN-3 Family

Getting started guide V1.0

Lipowsky Industrie-Elektronik GmbH Römerstraße 57 | 64291 Darmstadt | Germany Phone: +49 (0) 6151 / 93591 - 0 | Fax: +49 (0) 6151 / 93591 - 28 Website: www.lipowsky.com | E-Mail: info@lipowsky.de





1 Introduction	2
2 Feature Matrix	4
3 Getting started	4
3.1 Introduction	4
3.2 Installation	4
4 Session Description File (SDF)	5
4.1 How to create a LIN application	5
4.2 Introduction	5
4.3 Create a SDF	5
4.4 Common Setup	5
4.4.1 Emulation	5
4.4.2 GUI-Elements	6
4.4.3 Virtual signals	6
4.4.4 System signals	7
4.4.5 Macros	7
4.5 Example SDF	8
5 Start the bus communication	8
6 Updates	10
6.1 Update philosophy	10
6.2 Downloads	10
6.3 Installation	11
6.4 Check version	11
7 Support information	12







1 Introduction

This getting started guide will show you how to setup the Baby-LIN-3 Family to communicate with or monitor the LIN-Bus. Simply follow the next steps.



Information

Information This guide assumes you are using a Microsoft Windows operating system. If you use a Linux operating system please contact us to receive software for your distribution.

This guide is made for new Baby-LIN-3 Family users. If you already have experience with Baby-LIN products or you are an advanced

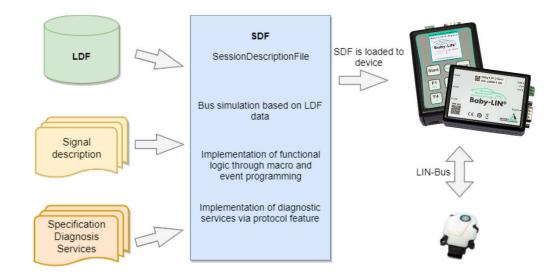
For this purpose, we will introduce the following components to you:

- LIN description file (LDF)
- Signal description
- Specification Diagnosis Services

From this information, the SessionDescriptionFile (SDF) can be created. The SDF is the linchpin in LINWorks-based applications.

The following graphic shows the typical workflow of a LIN-based application with our Baby-LIN-3 Family .

LIN-Bus user then this guide probably is not suited for you.

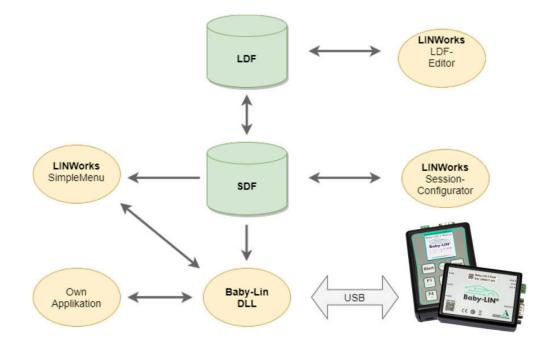


This diagram shows how the individual LINWorks software applications are linked to each other.















2 Feature Matrix

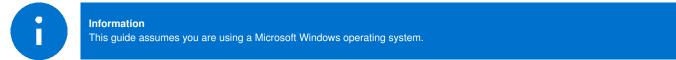
For an easier choice which device of the Baby-LIN-3 family may be suitable for you please check this matrix which list all devices and there features.

Device	1.LIN	2.LIN	CAN	IOs	Display and	RTC	microSD- Card	RS232
					Keyboard			
Baby-LIN-3-Single	Х							
Baby-LIN-3-Dual	Х		Х					
Baby-LIN-3-RC	Х				Х			
Baby-LIN-3-RCplus	Х	Х	Х	Х	Х	Х	Х	Х

3 Getting started

3.1 Introduction

This getting started guide will show you how to create your Lin application using the information from the LDF and the signal descriptions. In the following, you will learn how to create an LDF and integrate it into the SDF. Furthermore, the Unifeid Diagnostic Services will be introduced. After you have successfully created the SDF, the Baby-LIN-3 Family can be operated in standalone mode, LIN bus data can be logged, or macros can be defined for autostart.



3.2 Installation

Before you can start using the Baby-LIN-3 Family you have to install several components of the LINWorks software.

If you have not already downloaded the LINWorks software, please download it now from our website under following link: www.lipowsky.de

The following components are required for this getting started guide:

- Baby-LIN driver
- SessionConf
- SimpleMenu
- LDFEdit







4 Session Description File (SDF)

4.1 How to create a LIN application

1. Requirement



A LIN node (slave) and a suitable LDF file are available. An application is to be implemented in which a simulated LIN master allows the node to be operated in a certain way.





However, the information in the LDF is usually not sufficient. The LDF describes the access and interpretation of the signals, but the LDF does not describe the functional logic behind these signals. Therefore you need an additional signal description which describes the functional logic of the signals.

3. Requirement

Specification Diagnosis Services If the task also requires diagnostic communication, a specification of the diagnostic services supported by the nodes is also required. In the LDF, only the frames with the respective data bytes are defined, but not their meaning.

These requirements can then be defined and edited together in a Session Description file (SDF).

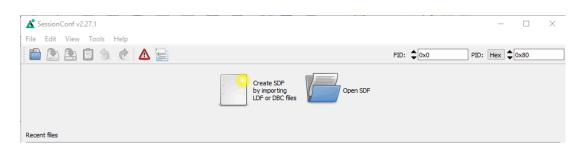
4.2 Introduction

The Session Description file (SDF) contains the bus simulation based on the LDF data. The logic of the individual frames and signals can be programmed by macros and events. In addition to the LDF LIN schedule, further diagnostic services can be implemented in the SDF via protocols.

This makes the SDF the central working point of all LINWorks applications.

4.3 Create a SDF

The SessionConf software application is used to create and edit the SDF. For this purpose, an existing LDF is imported.



4.4 Common Setup

4.4.1 Emulation

Select Emulation in the navigation menu on the left. Here you can select which nodes you want to be simulated by the Baby-LIN-3 Family . If you only want to monitor the LIN-Bus, select nothing.







🗎 🖄 🔔 🥱 🖉 🤷			FID: 0)x0	PID: 0x80	
PF Version 3 ▼	Name		FrameId	State	Set unused bits to 1	Comment
LIN: SimpleWiper 🔹	4 🔽	MasterECU [master]		Emulated		
		MasterCmd	0x10	Emulated		
+ - 2 🔁 😰		MasterReq	0x3c	Emulated		
	4 🗸	Slave1Motor		Emulated		
SDF Properties		MotorFrame	0x20	Emulated		l)
Emulation	4 🗸	Slave2Sensor		Emulated		
Virtual signals Signalfunctions Protocols	-	SensorFrame	0x30	Emulated		8
GUI-Elements (SimpleMenu/HARP etc)						
Macros						
Macroselection						
Events						
Device-specific options						

4.4.2 GUI-Elements

Select GUI-Elements in the navigation menu on the left. Here you can add signals you want to monitor.

ile Edit View Tools Help										
🗎 🕭 🔶 ዽ 🟝									FID: 0x0	PID: 0x80
SDF Version 3 🔹			Туре	Name	Target	Comment	Signals	Macros	Macroselecti	ons
L-LIN: SimpleWiper 🔻	0	1	' Edit signal	MessageCounter	MessageCounter		Drag and Dra	op to add		
	1	1	Edit signal	Ignition	Ignition		Filter:			8
4 - 8 2	2	1	' Edit signal	WiperSpeed	WiperSpeed		SignalNr	Sior	alname	Nodename
SDF Properties	3	1	' Edit signal	Temperature	Temperature		0	-		MasterECU (master)
Emulation Virtual signals							1	/ Ignit		MasterECU (master)
Signalfunctions							2	/ Wipe		MasterECU (master)
Protocols	4						3			MasterECU (master)
GUI-Elements (SimpleMenu/HARP etc) Macros	4						4	/ Wip		Slave1Motor
Macroselection							5			Slave1Motor
> Events							6			Slave1Motor
Device-specific options							7			
							8	🖉 Valu	eSensor	Slave2Sensor
							9	/ Mast	terReqB0	MasterECU (master)
							10	/ Mast	terReqB1	MasterECU (master)



Information

There are other ways to monitor frames and signals, but this is a good and configurable starting point.

4.4.3 Virtual signals

Virtual signals can store values just like bus signals, but they do not appear on the bus. They can be used for many different tasks like:

• Temporary values, like counters

· Operands and results from calculations

- Store constants
- The size of a virtual signal can be set to 1...64 bits. important for use in the protocol feature.

Each signal has a default value that is set when the SDF is loaded.

©2024 Lipowsky Industrie-Elektronik GmbH Römerstraße 57 | 64291 Darmstadt | Germany Phone: +49 (0) 6151 / 93591 - 0 Website: www.lipowsky.com



• etc.





	Local Date	expert settings 👻 Re	quired SDF	version: v3.14					
SDF Version 3	Г	Name	Length	Initial Value (decimal)	Initial Value (hexadecimal)	Initial Value (ASCII)	Reset on BUS start	Signed	
1-LIN: SimpleWiper 👻	26	©©SYSBUSSTATE	32	0	0x0				Gets the state of the LIN- or CAN-Bu
4 - 8 2	27	int8	32	0	0x0				
	28	int16	16	0	0x0				
Section properties	29	int32	32	0	0x0				
> Bus description Emulation	30	int64	64	0	0x0				
Tables	31	repetitions	32	0	0x0				
Virtual signals	32	runtime	32	0	0x0				
 Signalfunctions MessageCounter 	33	sync	1	0	0x0		Π	Π	
Protocols	34	failure	16	0	0x0		Π		

4.4.4 System signals

System signals are virtual signals with reserved names. When a system signal is applied, a virtual signal is created at the same time and linked to a specific behavior.

In this way, you can access timer, input and output resources and system information.

System Variable Wizard		?
Choose a system variable to add	I to the virtual signals list	
All Baby-LIN devices		▼ Name: @-@SYSBUSSTATE
iter:		Readonly: No
iter:		Reset policy: Default (as defined in virtuals signal table)
Name	Description	Description:
Timers		Gets the state of the LIN- or CAN-Bus.
Digital IO		
Analog		The following values are defined for the LIN-Bus:
 System 		Value Description
@@SYSINF01	System information	0 LIN-Bus voltage is missing
@@SYSINF02	System information	 LIN-Bus voltage is available, but no schedule is running
@@SYSINF03	System information	2 LIN-Bus voltage is available and a schedule is running
@@SYSINFO4	System information	The following values are defined for the CAN-Bus:
@@SYSINF05	System information	Value Description
@@SYSINFO6	System information	0 CAN-Bus has not been started or was stopped
@@SYSINF07	System information	 CAN-Bus was started, but no transmission was acknowledged and no frame was received
@@SYSBUSSTATE	Gets the state of the LIN- or CAN-Bus.	from another node 2 CAN-Bus was started and either a transmission was admowledged or a frame was received
@@SYSCFG1		2 Criveous was started and enter a transmission was acchievedged or a mame was received from another node.
@@SYSCFG2		
@@SYSCFG9		
@@SYSCFG30		Available on these devices:
@@SYSCFG31		All Baby-LIN devices
@@SYSCFG100		
@@SYSCFG101		
@@SYSCFG203		
@@SYSCFG204		
@@SYSINTERNAL @@SYSMACROS.CONCL		



Information

For more information and a list of all available system signals, please check the System Signal Wizard in SessionConf.

4.4.5 Macros

Macros are used to combine multiple operations into a sequence. Macros can be started by events or, can also be called from other macros in the sense of a Goto or Gosub. The DLL API calls a macro with the macro_execute command.

▲ SessionConf v2.30.12 - [C:/Users/jschafhausen/	/Desktop/	Software_Man	ual/SDF/Example.sdf*]	
File Edit View Tools Help				
🗎 🕭 🟝 🗊 🥱 🕐 🛕 🔛	Hide exp	ert settings 🔻	Required SDF version: v3.14	
SDF Version 3	Macro n	umber 1		
1-LIN: SimpleWiper	Name	Execu	te	
4 - 🕄 🙎	Paramet	ter count 0		
Section properties	Commer	nt 🗌		
 Bus description Emulation Tables 	La	ibel Conditio	n Command Print on Debug report: "Macro starts"	Comment
Virtual signals	1		Gosub macro "BusStart()"	Macro BusStart is being executed
✓ Signalfunctions	2		Gosub macro "Example(250, 1000)"	Macro Emxaple is executed and is passed the values 250 and 1000 as parameters
MessageCounter Protocols GUI-Elements (SimpleMenu/HARP etc) V Macros	3		Print on Debug report: "Execution was successful"	
BusStart				

All Macro Commands can use signals from the LDF and signals from the Virtual Signal section like the system signals.







Another important function of the macros is to control the bus. The bus can be started and stopped via macro. Furthermore, the schedule can be selected and the status of the bus can be checked with the help of the system signals.

🗎 🕭 🟝 🗊 🥱 🥐 🛆 🔚	Hide ex	pert settin	igs 👻 Required SDF version:	v3.14							
SDF Version 3	Macro	number	0								
1-LIN: SimpleWiper	Name		BusStart								
4 - 8 2	Param	eter count	0								
Section properties		10040									
 Bus description Emulation Tables 	0	abel	Condition	Command Print on Debug report: "Bus starts with schedule Table1"	Commer						
Virtual signals	1			Start BUS with schedule Table1							
✓ Signalfunctions	2			Delay 200ms							
MessageCounter Protocols GUI-Elements (SimpleMenu/HARP etc)	3	lf	Signal @@SYSBUSSTATE = 2	Print on Debug report: "The bus has been started and the schedule is running table1"							
GUI-Litements (simpleMenu/HARP etc) ✓ Macros BusStart Macroselection > Events > Device-specific options											

Each macro always provides 13 local signals:

_LocalVariable1, _LocalVariable2, ..., _LocalVarable10, _Failure, _ResultLastMacroCommand, _Return The last 3 provide a mechanism to return values to a callcontext _Return, _Failure) or to check the result of a previous macro command. The signals _LocalVariableX can be used e.g. as temporary variables in a macro.

File Edit View Tools Help	Hide expert setting	Required SDF version: v3.14	
SDF Version 3	Macro number	2	
1-LIN: SimpleWiper	Name	Example	
4 - 8 2	Parameter count	2	
	Parameter names	int16Value	int64Value
Section properties Bus description	Comment		
Emulation Tables Virtual signals	Label Co	ndition Command Set signal "speed_1" to value from signal "_LocalVariable1"	Comment
✓ Signalfunctions	1	Set signal "speed_1" to value from signal "LocalVariable2"	
MessageCounter	2	Set signal "WiperSpeed" to value from signal "speed_1"	
Protocols GUI-Elements (SimpleMenu/HARP etc)	3	Delay 500ms	
✓ Macros	4	Set signal "WiperSpeed" to value from signal "speed_2"	
BusStart Execute	5	Delay 500ms	
Example(int16Value, int64Value)			

A macro can receive up to 10 parameters when called. In the macro definition, you can give these parameters names, which are then displayed on the left in the menu tree in brackets after the macro name. The parameters end up in the signals _LocalVariable1...10 of the called. If no parameters or less than 10 parameters are are passed, the remaining _LocalVariableXsignals receive the value 0.

4.5 Example SDF

You can download the example SDF under the section "08 | Examples SDF 's" under the following link: GettingStarted_Example.sdf

5 Start the bus communication

Start the SimpleMenu. You should be able to find your Baby-LIN-3 in the device list on the left. Click the connect button and then load the SDF you created earlier.







Device View Toolbars Wind	lows Tools Help	▲ SimpleMenu v2.35.0 Device View Toolbars Windows Too □ <th>ols Help</th> <th></th> <th></th> <th></th> <th></th>	ols Help				
Device List	C 8 ×	Device List	ъх	Baby-LIN Simulation	Windo	w	
Baby-LIN-3 serial://COM5	¢	Baby-LIN-3-Single USB: COM5 Serial:2354929 SW-Version:6.24 rev0 test No SDF loaded Channels LIN Baudrate: N/A Section: None loaded	ی با م	H	C	Ê	⊉ (

Now you can see the variables you added to monitor. To start the simulation/monitoring click on the start button.

👗 SimpleMenu v2.35.0											_		×
Device View Toolbars Windows	Tools Help												
8008													
levice List	8×					Baby-LIN-3-S	ingle(2354929)	LIN					×
	C	Simulation Wi	indow										
Baby-LIN-3-Single USB: COM5 Serial:2354929	\$ • • •	▶ ■ C		0		Baudrate	: 19200 Bit/s	¢		~명 (3 ≤	E 4	м :
SW-Version:6.24 rev0 test		MessageCounter		1									
oaded SDF: Example.sdf		Ignition	٢		0								
DFVersion: 3.14 umber of Sections: 2	CD	WiperSpeed	/		Off								
Channels LIN	-	Temperature	۲	25	5 Signal not	available							
Baudrate: 19200 Bit/s 🖞	8	BusStart	Ru	n	Macro suc	ceeded, Result	= 0						
Section: SimpleWiper		Execute	Ru	n									
		BusStop	Ru	n									
		визэтор	KU	n									
		Report Monitor	KU	n								~	8
		Report Monitor	inuous Sta	î	Apply filter from se	ettings 🖂] Frames 🗌	Signals 🗹 Events	Errors	Debug	Macros		₽
		Report Monitor		tic	Apply filter from s	ettings 🖂 V1=0xFF		Signals 🗹 Events	Errors	✓ Debug	Macros		\$
		Report Monitor Conti < +0,020	inuous Sta	tic 🗆 0x00 03	t00 0x00 0x00		DL:4	Signals 🗹 Events	Errors	Debug	Macros		\$
		Continue +0,020 +0,020	inuous Sta s 0x20[0x20	tic [] 0x00 02 0x00 02	t00 0x00 0x00	V1=0xFF	DL:4	Signals 🗹 Events	Errors	☑ Debug	Macros	0	\$
		Report Monitor Cont Cont < +0,020 < +0,020 < +0,020	inuous Sta is 0x20[0x20 is 0x30[0xF0	tic 0x00 02 0x00 02 0x00 02	x00 0x00 0x00	V1=0xFF V1=0xFF	DL:4 DL:2 DL:4	Signals 🗹 Events	Errors	☑ Debug	Macros		\$
		Report Monitor Cont Cont Cont	inuous Sta Is 0x20[0x20 Is 0x30[0xF0 Is 0x10[0x50	bc	00 0x00 0x00 00 08 0x00 0xFF 00 0x00 0x00	V1=0xFF V1=0xFF V1=0xF7	DL:4 DL:2 DL:4 DL:4	Signals 🗹 Events	Errors	Z Debug	Macros		\$
		Report Monitor Cont Cont Cont	inuous Sta s 0x20 [0x20 s 0x30 [0xF0 s 0x10 [0x50 us 0x20 [0x20 s 0x30 [0xF0	ttc	00 0x00 0x00 00 08 0x00 0xFF 00 0x00 0x00	V1=0xFF V1=0xFF V1=0xF7 V1=0xFF	DL:4 DL:2 DL:4 DL:4 DL:2	Signals 🗹 Events	Errors	✓ Debug	Macros		\$
		Cont +0,020 < +0,020	inuous Sta s 0x20 (0x20 s 0x30 (0xF0 s 0x10 (0x50 s 0x20 (0xF0 s 0x30 (0xF0 s 0x10 (0x50	ttc 0x00 02 0x00 02 0x00 02 0x00 02 0x00 02 0x00 02 0x00 02 0x00 02	x00 0x00 0x00 x00 x08 0x00 0xFF x00 0x00 0x00 x00	V1=0xFF V1=0xFF V1=0xF7 V1=0xFF V1=0xFF V1=0xFF	DL:4 DL:2 DL:4 DL:4 DL:2 DL:2 DL:4	Signals 🗹 Events	Errors	☑ Debug	Macros	□	\$
		Cont +0,020 +0,020 +0,020 +0,020 +0,020 +0,020 +0,020 +0,020 <	inuous Sta s 0x20[0x20 s 0x30[0xF0 s 0x10[0x50 s 0x20[0x20 s 0x30[0xF0 s 0x10[0x50	bc 0x00 02 0x00 02 0x00 02 0x00 02 0x00 02 0x00 02 0x00 02 0x00 02 0x00 02	x00 0x00 0x00 x00 x00 0x00 0xFF x00 0x00 0	V1=0xFF V1=0xFF V1=0xF7 V1=0xFF V1=0xFF V1=0xFF	DL:4 DL:2 DL:4 DL:4 DL:2 DL:2 DL:4 DL:4	Signals 🗹 Events	Errors	☑ Debug	Macros		

Now you will see the changes of these signals.







6 Updates

6.1 Update philosophy

The functionality and features of the Baby-LIN-3 are defined by the installed firmware as well as the used versions of the LINWorks and Baby-LIN-DLL.

As we are permanently working on product improvements, the software and firmware are updated periodically. These updates make new features available and solve problems, which have been discovered by our internal tests or have been reported by customers with earlier versions.

All the firmware updates are done in a way, that the updated Baby-LIN-3 will continue to work with an already installed, older LINWorks installation. So updating the Baby-LIN-3 firmware does not mean, that you necessarily have to update your LINWorks installation as well.

Therefor it is highly recommended to always update your Baby-LIN-3 Family to the latest available firmware version.

We also recommend to also update your LINWorks software and Baby-LIN-DLL, if new updates get available. Since new versions of the SessionConf may introduce new features to the SDF format, it is possible that older firmware, SimpleMenu or Baby-LIN-DLL versions are not compatible. Therefor you should also update them.

> If you update your LINWorks it is highly recommended updating the firmware of your Baby-LIN-3 Family to the latest available firmware version as well as distributed the used versions of the Baby-LIN-DLL.

So the sole reason to stay with an older LINWorks version should be, that you use a Baby-LIN-3 with outdated firmware version, which you can't upgrade for whatever reason.

It is highly recommended updating the Baby-LIN driver to the latest version.

6.2 Downloads

The latest version of our software , firmware and documents can be found in the download area on our website www.lipowsky.de.



Information

The LINWorks archive contains not only the LINWorks software but also the manuals, datasheets, application notes and examples. Only the device firmware packages are not included. The firmware is available as separate package.

Documents such as the data sheets or introductions to LIN bus communication are freely available for download. For all other documents and our LINWokrs software you have to log in. If you do not have a customer account yet you can register on our website. After your account has been activated by us you will receive an e-mail and then you have full access to our download offer.

DOWNLOADS

HERE YOU CAN DOWNLOAD DOCUMENTS FREE OF CHARGE. FOR THE LOCKED CONTENT, PLEASE LOG IN WITH YOUR CUSTOMER ACCESS.

01 | Baby-LIN Software

LinWorks Software | Version 2.31.1 More A File name: LinWorks-PCSoftware-2X-CD zip Latest version of the LINWorks V2 software suite as zip archive. Contains current versions of LINWorks software, Baby-LIN DLL associated wrappers and Baby-LIN USB drivers as well as data sheets, manuals and program examples. (376.6MiB) 21.07.30 🗗 🕕







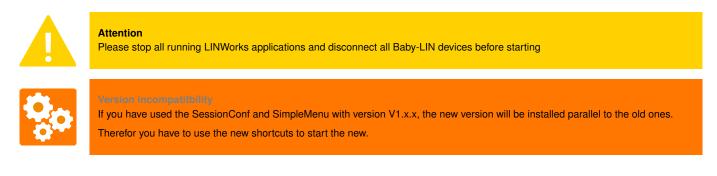
LOGIN	REGISTER
If you were previously registered in the customer portal, you must register again. All you need is your e-mail address with which you were registered on the portal and a new password. Your account will then be activated directly.	E-Mail: Password (minimum 6 characters):
E-Mait:	Repeat password:
assword: Password forgotten?	I have read and accept the privacy policy.* I would like to receive the newsletter. REGISTER
You do not have an account yet? Register	You already have an account? Log in

6.3 Installation

The LINWorks suite is delivered with a handy setup application. If you already have installed an older version you can simply install the newer versions. The setup application will take care of overwriting the required files.

Simply follow these steps:

- Start the "Setup.exe".
- Select the components you want to install.
- Follow the instructions.



6.4 Check version

If you want to check the current version of the Baby-LIN-3 firmware or a LINWorks component the following chapter shows you how it is done:

Baby-LIN-3 firmware

Start the SimpleMenu and connect to the Baby-LIN-3. Now the firmware version is visible in the device list.

SimpleMenu v2.31.2 Device View Toolbars Windows Tools	Help
80882	
Device List	Baby-LIN-RC-II(1822754) LIN Simulation Window
Baby-LIN-RC-II USB-COMS Serial:1822754 SW-Version:6.20 rev3 No SDF loaded	
▼ Channels LIN Baudrate: N/A ● Section: None loaded	

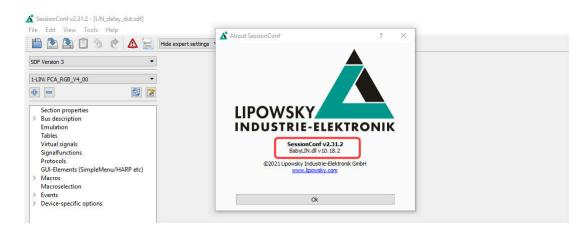






LINWorks [LDFEdit SessionConf SimpleMenu LogViewer]

Select the menu option "Help"/"About"/"Info". The info dialog will show the software version.



Baby-LIN-DLL

Call BLC_getVersionString(). The version is returned as string.

Baby-LIN-DLL .NET Wrapper

Call GetWrapperVersion(). The version is returned as string.

7 Support information

In case of any questions you can get technical support by email or phone. We can use TeamViewer to give you direct support and help on your own PC. This way we are able to sort out problems fast and direct. We have sample code and application notes available, which will help you to make your job.

Lipowsky Industrie-Elektronik GmbH realized many successful LIN and CAN related projects and therefor we can draw upon many years of experience in these fields. We also provide turn key solutions for specific applications like EOL (End of Line) testers or programming stations.

Lipowsky Industrie-Elektronik GmbH designs, produces and applies the Baby-LIN products, so you can always expect qualified and fast support.

Contact informations	Lipowsky Industrie-Elektronik GmbH, Römerstr. 57, 64291 Darmstadt		
Website:	www.lipowsky.com	Email:	info@lipowsky.de
Telephone:	+49 (0) 6151 / 93591 - 0		

