**EN:** This Datasheet/Document is presented by the manufacturer.

Please visit our website for pricing and availability at www.hestore.hu.




# Baby-LIN-II

*User Manual V4.0*

Phone: 010-2601-9622
E-Mail: info@haehong.com
Website: *https://haehongtec.com/*

**1 Imprint** **3**

**2 Glossary** **4**

**3 Safety instruciotns** **4**

3.1 Warning signs....4

3.2 Safety precautions....5

**4 Preface** **6**

4.1 Updates....6

4.1.1 Update philosophy....6

4.1.2 Downloads....6

4.1.3 Installation....7

4.1.4 Check version....7

4.2 Support information....8

**5 Hardware** **9**

5.1 Overview....9

5.2 Connectors....9

5.2.1 X1 - PC connector....9

5.2.2 X2 - LIN-Bus connector....9

5.2.3 Equivalent circuit of the LIN-Bus interface:....10

5.3 Power supply....11

5.4 LEDs....11

5.4.1 LD1 - Run LED....11

5.4.2 LD2 - Error LED....11

**6 Firmware** **12**

6.1 Introduction....12

6.2 Required software....12

6.3 Update the firmware....12

6.4 Stand-alone mode and autostart....13

6.4.1 Enable the stand-alone mode....13

6.4.2 Configure the autostart macro....13

6.4.3 Store a SDF persistently....13

6.4.4 Configure the device to automatically load a SDF and start a macro....14

6.5 Logging....15

6.5.1 Introduction....15

6.5.2 Log data targets ..... 15
6.5.3 Log data formats ..... 17

**7 Workflow** **18**
7.1 Overview ..... 18
7.2 Getting started ..... 19
7.2.1 Introduction ..... 19
7.2.2 Installation ..... 19
7.3 LDF ..... 20
7.3.1 LDF Example ..... 20
7.3.2 LIN application frames ..... 21
7.3.3 LIN Scheduling ..... 22
7.3.4 LIN Diagnostic frames ..... 22
7.4 Session Description File (SDF) ..... 24
7.4.1 How to create a LIN application ..... 24
7.4.2 Introduction ..... 24
7.4.3 Create a SDF ..... 24
7.4.4 Common Setup ..... 24
7.4.5 Start the bus communication ..... 27

**8 LINWorks Software - Overview** **28**

**9 Migration information** **31**
9.1 Migration from Baby-LIN-II to Baby-LIN ..... 31
9.2 Performance ..... 31
9.3 LIN-Bus transceiver ..... 31

# 1 Imprint

| | |
|---|---|
| Author | Lipowsky Industrie-Elektronik GmbH Römerstraße 57 64291 Darmstadt |
| Phone | +49 (0) 6151 / 93591 - 0 |
| Fax | +49 (0) 6151 / 93591 - 28 |
| E-Mail | *info@lipowsky.de* |
| Website | *www.lipowsky.com* |
| CEO | Andreas Lipowsky |
| Commercial register | Darmstadt HRB 5139 |
| VAT-ID | DE 111647423 |
| Quality Management | DIN EN ISO 9001:2015 |

| | |
|---|---|
| Title | Baby-LIN-II user manual |
| Version | V4.0 |
| Date | 2022-02-25 |
| Valid for | Baby-LIN-II |
| Copyright | © 2021, Lipowsky Industrie-Elektronik GmbH, Darmstadt |

This publication is copyright protected. All rights reserved, including those to translation, performance, use of illustrations and tables, broadcasting, microfilming or reproduction by other means, or electronic storage of all material contained herein.

All other brand names and trademarks used within this manual are unlimited subject to the applicable trademark laws and the ownership rights of their registered owners.

The hardware, firmware, software and documents of the Baby-LIN-II are subject to change without prior notice. Lipowsky Industrie-Elektronik GmbH thereby has no obligations.

# 2 Glossary

| | |
|---|---|
| **ADC** | Ampere Direct Current. This is the unit of DC current values. |
| **CAN** | Controller Area Network |
| **CAN-HS** | CAN high speed. Theses are CAN interfaces with high data rates according to ISO-11898. |
| **CAN-LS** | CAN low speed. Theses are CAN interfaces with fault tolerant low data rates according to ISO-11519. |
| **CD** | Compact Disk |
| **DBC** | Database CAN: A file that contains the description of a CAN bus. It contains nearly the same information as a ARXML file. |
| **DLL** | Dynamic Link Library. It can be used to execute the DLL functions in custom applications. |
| **ECU** | Electronic control unit |
| **EOL** | End of line |
| **ESD** | Electro static discharge. The sudden flow of electricity between two electrically charged objects caused by e.g. contact. |
| **EU** | European Union. The Lipowsky Industrie-Elektronik GmbH resides inside the EU. Therefor shipping within the EU can be done without customs duties. You should definitely check out our worldwide distributors. Check chapter Distributors for more information. |
| **LIN** | Local Interconnect Network |
| **LINWorks** | Application software suite to configure the Baby-LIN devices. |
| **PC** | Personal Computer |
| **PLC** | Programmable Logic Controller |
| **PWM** | The pulse-width modulation is a modulation technique used to encode a value into a pulsing signal. |
| **RTC** | Real-time clock. |
| **SD** | Secure Digital Memory Card. This is a type of non-volatile memory cards. |
| **SDF** | Session Description File |
| **SID** | Service identifier. This number identifies a protocol service. |
| **USB** | Universal Serial Bus |
| **VDC** | Voltage Direct Current. This is the unit of DC voltage values. |

# 3 Safety instruciotns

## 3.1 Warning signs

The following warning signs are used for safety precautions:



DANGER indicates a hazardous situation which, if not avoided, will result in death or serious injury.



WARNING indicates a hazardous situation which, if not avoided, could result in death or serious injury.



CAUTION indicates a hazardous situation which, if not avoided, could result in minor or moderate injury.

**NOTICE** NOTICE is used to address practices not related to physical injury.

**SAFETY INSTRUCTIONS** Safety instructions signs indicate specific safety related instructions or procedures.

The following notice types are used to give you non safety precaution related information, e.g. software or configuration related problems:



**Attention**
This notice type signals possible problems, you should definitely pay attention to. Ignoring them probably lead to unexpected behaviour or data loss.



Version Incompatibility
This notice type signals possible version incompatibilities and may lead to unexpected behaviour. These incompatibilities can be caused by old or incompatible software or firmware versions as well as missing activation codes.



**Warning**
This notice type signals possible problems, you should pay attention to. Ignoring them may lead to unexpected behaviour or data loss.

**Attention**
This notice type should inform you about useful information, that help you understand the Baby- LIN-RM-III better.



**Attention**
This notice type should give you tips, that help to reduce your expense and time to implement.

## 3.2 Safety precautions

Despite compliance with the relevant laws and regulations, residual risks can not be excluded. The following safety precautions define the hazards that can occur when operating the Baby-LIN-II

| | |
|---|---|
| **DANGER**  | Mortal danger by automatic start of connected devices.<br>• Prepare for actions from connected devices.<br>• Study safety precautions of connected devices. |
| **WARNING**  | Mortal danger by electric shock.<br>• Operate the Baby-LIN-II only within dry conditions.<br>• Do not touch the Baby-LIN-II if powered.<br>• Do not touch the Baby-LIN-II if damaged. |
| **CAUTION**  | • Do not touch the Baby-LIN-II if wet.<br>Injury by damaged battery.<br>• Operate the device only within the defined operating temperature.<br>• Observe the correct polarity when inserting the battery.<br>• Do not touch the battery if damaged.<br>• Do not touch the battery if wet. |
| **NOTICE** | Please recycle or dispose the battery safely and properly according to local laws and regulations. |

# 4 Preface

## 4.1 Updates

### 4.1.1 Update philosophy

The functionality and features of the Baby-LIN-II 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-II will continue to work with an already installed, older LINWorks installation. So updating the Baby-LIN-II 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-II 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-II 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-II 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.**

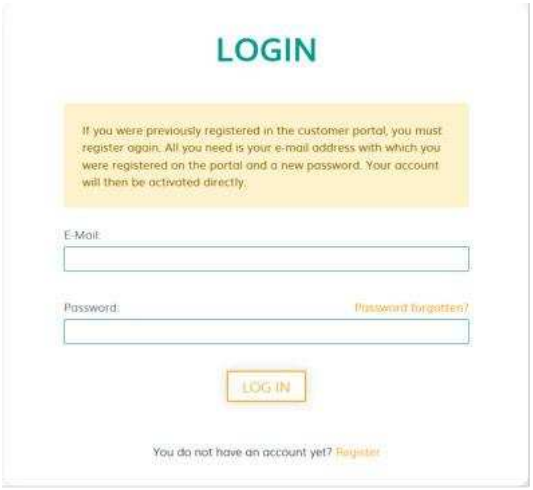
### 4.1.2 Downloads

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

**Advice**
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.









### 4.1.3 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.



**Warning**
Please stop all running LINWorks applications and disconnect all Baby-LIN devices before starting the setup.



**Version incompatitbility**
If you have used the SessionConf and SimpleMenu with version V1.x.x, the new version will be installed parallel to the old ones. Therefor you have to use the new shortcuts to start the new versions.

### 4.1.4 Check version

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

| Component | How to check the version |
|---|---|
| Baby-LIN-II firmnware | Start the SimpleMenu and connect to the Baby-LIN-II . Now the firmware version is visible in the device list. |
| LINWorks:<br>• LDFEdit<br>• SessionConf<br>• SimpleMenu<br>• 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. |



**Advice**
If you need support please always tell us the firmware and software versions you are using.

## 4.2 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 | | |

# 5 Hardware

## 5.1 Overview

The following images show you what features the Baby-LIN-II has to offer. The following features will be shown:

| Abbreviation | Description |
|---|---|
| X | Connectors to access the different interfaces. |
| LD | LEDs that symbol certain states. |




| Abbreviation | Type | Description |
|---|---|---|
| X1 | USB 2.0 type B-Mini | PC connector. Can be used as logic power supply. |
| X2 | Socket for MC 1,5/ 3-ST-3,81 and MCVR 1,5/3-ST-3,81 | LIN-Bus connector. Can be used as logic power supply. |
| LD1 | Green LED | LIN-Bus voltage indicator. |
| LD2 | Red LED | LIN-Bus communication error. |

## 5.2 Connectors

### 5.2.1 X1 - PC connector

This connector is a USB type B-Mini. It is used to connect the Baby-LIN-II to a PC. To use this interface the Baby-LIN USB driver has to be installed on the PC.



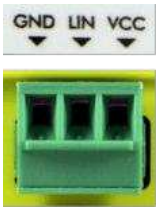
| Description |
|---|
| The connector uses the default pin assignment of USB type BMini. |

**Advice**
The USB interface is galvanically isolated from all other connectors.

### 5.2.2 X2 - LIN-Bus connector

The LIN-Bus interface is available via a MC 1,5/ 3-ST-3,81 connector.

The LIN interface of the Baby-LIN-II can be operated between voltages of 8-26 VDC.




| Pin | Signal | Description |
|---|---|---|
| X2-1 | GND | LIN-Bus ground |
| X2-2 | LIN | LIN-Bus signal |
| X2-3 | VCC | |




Keep the LIN-Bus voltage within the following range: 8-26 VDC.

- Injury by damaged Baby-LIN-II .
- The Baby-LIN-II may get damaged.




Check LIN-Bus node specifications before using voltages above 18 VDC.

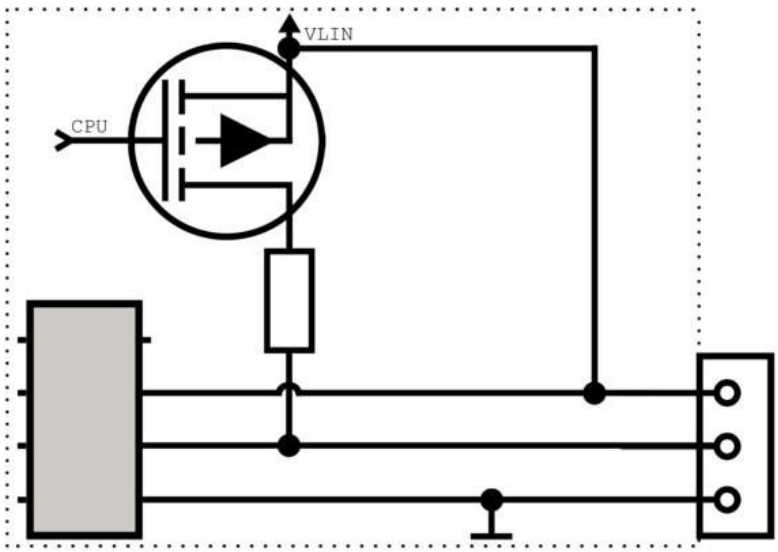
If voltages in excess of 18 VDC are used as LIN-Bus supply voltage, it must be ensured that all connected nodes can cope with this voltage level. It is possible, that some nodes will function incorrectly in voltages exceeding 18 VDC, since the LIN specification states a maximum voltage of 18 VDC.

- Injury by damaged Baby-LIN-II .
- The Baby-LIN-II may get damaged.

**Advice**
The LIN-Bus interface is galvanically isolated from the logic supply, the USB interface.

**Attention**
The LIN-Bus supply must be provided by an external power supply and must not be interrupted during the LIN communication.

### 5.2.3 Equivalent circuit of the LIN-Bus interface:



| Electrical characteristics | Value | Unit |
|---|---|---|
| **Maximum operational voltage** | 26 | V |
| **Maximum LIN voltage** | 32 | V |
| **Maximum LIN current** | 450 | mA |
| **Maximum LIN supply voltage** | 51 | V |
| **Maximum LIN supply current** | 550 | mA |

| Electrical characteristics | Voltage threshold for LIN detection | Voltage threshold for LIN transceiver |
|---|---|---|
| **Minimum** | With USB connection: 3.9 V<br>Without USB connection: 5.58 V | 6.8 V |

The pull-up resistor of the LIN-Bus driver is switched to 1 kOhm, if the master node is emulated and to 30 kOhm, if only slave nodes are emulated.

## 5.3 Power supply

The Baby-LIN-II can be powered by one of the following sources:

- Power supply over the LIN-Bus supply: "X2 - LIN-Bus connector" Supported power supply voltage: 5V (Typical USB specifications)
- Power supply over USB: USB 2.0 type B-Mini Supported power supply voltage: 8-26 VDC

The Baby-LIN-II has a typical power consumption of 70 mA @ 12 VDCmA.

## 5.4 LEDs

### 5.4.1 LD1 - Run LED

This green LED shows the state of the LIN-Bus voltage. It will blink once per second. The duration the LED is on gives you information about the LIN-Bus voltage:

| The duration the LED is on | Information |
|---|---|
| 50 ms | No LIN-Bus voltage available. |
| 500 ms | LIN-Bus voltage is available. |

This LED can also be freely programmed by setting the value of the system variable `@@SYSLED11`:

| System variable value | Description |
|---|---|
| 0 | LED is off |
| 1 | LED is on |
| 2 | LED is controlled by the firmware. The default behaviour is active. |
| 3 or higher | LED is on for the assigned value in milliseconds. |

### 5.4.2 LD2 - Error LED

This red LED shows errors on the LIN-Bus. If it blinks an error like the following has happened on the LIN-Bus:

- Checksum error
- Missing responses

If the Baby-LIN-II enters the transparent mode while the ASCII mode is active, the LED will blink alternatively with "LD1 - Run LED". Check chapter

"ASCII mode" and "TRANSPARENT" for more information.

This LED can also be freely programmed by setting the value of the system variable `@@SYSLED12:`

| System variable value | Description |
|---|---|
| 0 | LED is off |
| 1 | LED is on |
| 2 | LED is controlled by the firmware. The default behaviour is active. |
| 3 or higher | LED is on for the assigned value in milliseconds. |

# 6 Firmware

## 6.1 Introduction

The firmware is stored in the flash memory of the Baby-LIN-II and can easily be programmed from a PC. The update process uses an ISP (In-system programming) operation and can be executed in the field.

As we are permanently working on product improvements, the firmware is 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-II will continue to interwork with an already installed, older LINWorks installation. So updating the Baby-LIN 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-II to the latest available firmware version.

## 6.2 Required software

| Download archive | Description |
|---|---|
| BabyLinDriverSetup.exe | This setup contains the Baby-LIN driver. It is required to update the firmware. |
| FirmwarePackage-Baby-LIN-Devices.zip | This package contains the firmware for the Baby-LIN-II as well as the update tool "blprog". |

## 6.3 Update the firmware

To update the firmware of the Baby-LIN-II you have to follow these steps:

- Install the drivers for the Baby-LIN-II, if you have not already installed them.
- Connect the Baby-LIN-II with your PC.
- Unpack the firmware archive.
- Start the "blprog.exe". A command prompt will be opened and guide you through the process.
- If you have other Baby-LIN products aside the Baby-LIN-II connected with the PC, you will be asked which Baby-LIN product you want to update.
- Press the "y" key to confirm the correct firmware and start the flash process.
- Wait until the flashing finished.
- After the flashing has finished press the "Enter" key to exit the tool.

## 6.4 Stand-alone mode and autostart

### 6.4.1 Enable the stand-alone mode

The Baby-LIN-II is able to operate stand-alone without a PC, PLC or operator. To enable this mode several requirements need to be met:

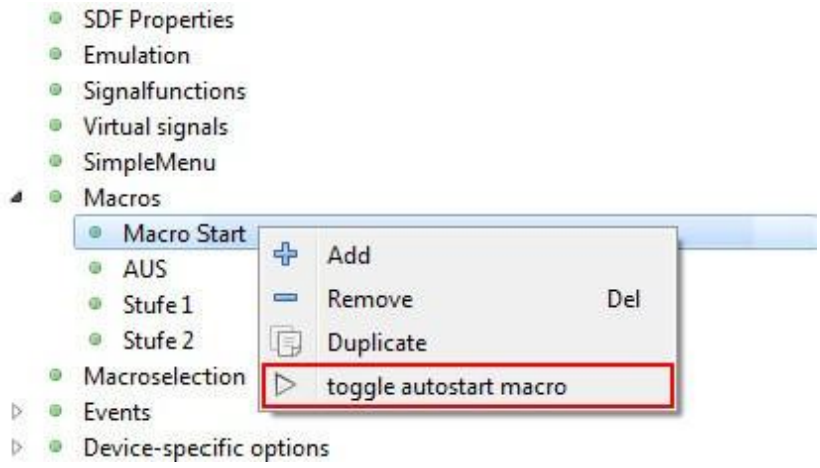
- You need a SDF with a macro that is marked as "autostart" macro.
- The SDF needs to be stored persistently on the Baby-LIN-II.
- The device needs to be configured to automatically start the autostart macro of a persistently stored SDF when powered up.

### 6.4.2 Configure the autostart macro

You can mark a macro in a SDF as autostart macro. This means that this macro is started automatically when the SDF is loaded. Usually this macro will start the LIN-Bus communication and perform necessary initialisations.

To mark a macro as autostart macro the following steps are necessary.

- Open your SDF using the SessionConf.
- If this SDF does not already have a macro that you want to use as autostart macro please create one.
- Right click on the macro and select "toggle autostart macro". The macro will now have the "[autostart]" marker.



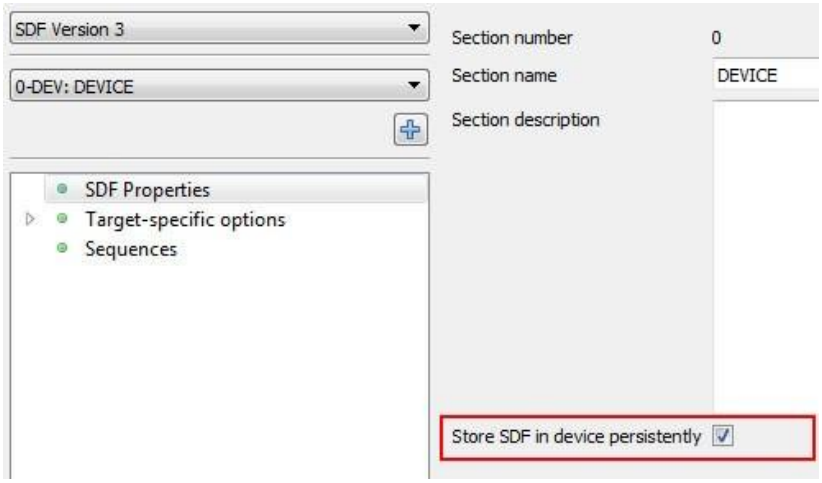

**Advice**
Please note that only one macro can be marked as autostart macro.

**Tip**
You probably want to start the LIN-Bus within the macro since it is not automatically started by loading the SDF.

### 6.4.3 Store a SDF persistently

It is possible to store a SDF persistently on the Baby-LIN-II. Therefor a special option in the SDF must be configured:

- Start the SessionConf and load your SDF.
- In the left menu click on "SDF Properties".
- Check the "Store SDF in device persistently" option.




- Save your SDF and reload it to a device using the SimpleMenu.

**Advice**
The option "Store SDF in device persistently" can always be found in the "SDF Properties". It is independent from the SDF version (SDF v2, SDF v3) or type of section currently selected. This option is global for the whole SDF even though it can be found in every section.

### 6.4.4 Configure the device to automatically load a SDF and start a macro

The Baby-LIN-II can be configured to automatically start a autostart macro. The Autostart can be triggered if the device is powered on or if the LIN-Bus is powered on. These target-specific options (also known as target configuration) are persistently stored on the Baby-LIN-II.

There are 2 important options that affect the autostart feature:

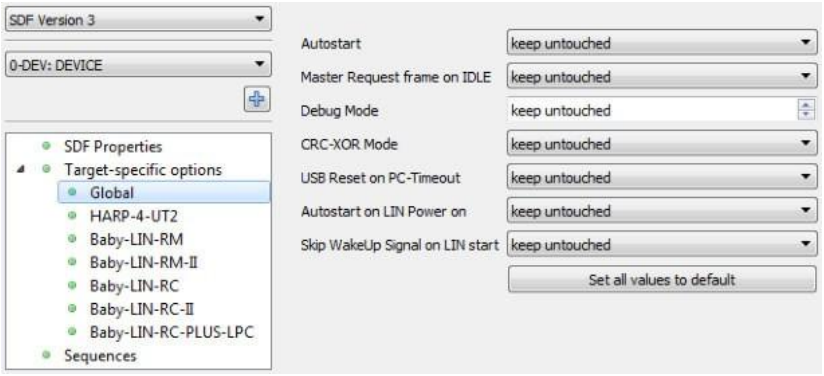
| Target-specific option | Possible values | Description |
|---|---|---|
| Autostart | Off | No schedule or macro is started when the autostart is triggered. |
| | Schedule | The first schedule but not the autostart macro is started when the autostart is triggered. |
| | Schedule + Macro | The first schedule and the autostart macro are started when the autostart is triggered. |
| | Macro only | No schedule but the autostart macro is started when the autostart is triggered. |
| Autostart on LIN Power | Off | An autostart is triggered only when the device is powered on. |
| | On | An autostart is triggered each time the LIN-Bus is powered on. |

**Advice**
If you want to start another but the first schedule simply select the macro only option and start another than the first schedule from that macro.

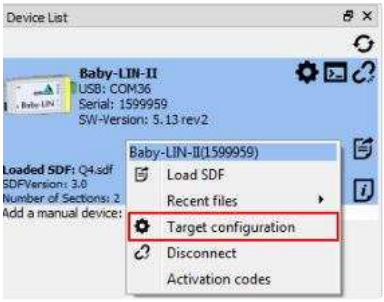
**Tip**
Triggering the autostart each time the LIN-Bus is powered on is especially helpful for applications in which the node connected to the LIN-Bus is changed periodically, e.g. EOL applications. In such a case, the simulation may be started automatically every time a new node is connected and the LIN-Bus voltage is turned on.

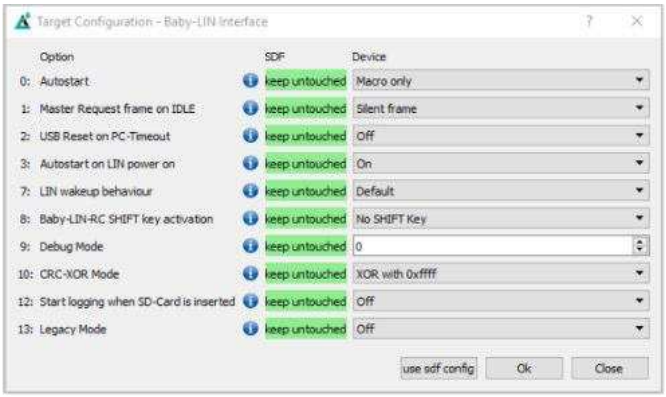
**Warning**
The option "Autostart on LIN Power on" is not required for the Baby-LIN-II if the LIN voltage is also used as the logic supply.

SDF can store preset values for the target-specific options. These preset values can be found in the device section. These values overwrite the values in the device each time the SDF is uploaded. If a value is set to "keep untouched" the value in the device will not be changed.




The target-specific options can be set using the SimpleMenu. Right click on the connected device and click on "Target configuration". In the following dialog you can change the values in the device. For comparison reasons the preset values of the SDF are shown.




## 6.5 Logging

### 6.5.1 Introduction

The Baby-LIN-II supports the logging of the frames on the LIN-Bus. The following possibilities to log the bus data are available:

- The bus data can be logged using the SimpleMenu if the Baby-LIN-II is connected to a PC. The logging using the SimpleMenu is described here: Check chapter **"SimpleMenu"** for more information..

Version compatitbility
This feature requires an up to date firmware. Check chapter **"Updates"** for more information.

### 6.5.2 Log data targets

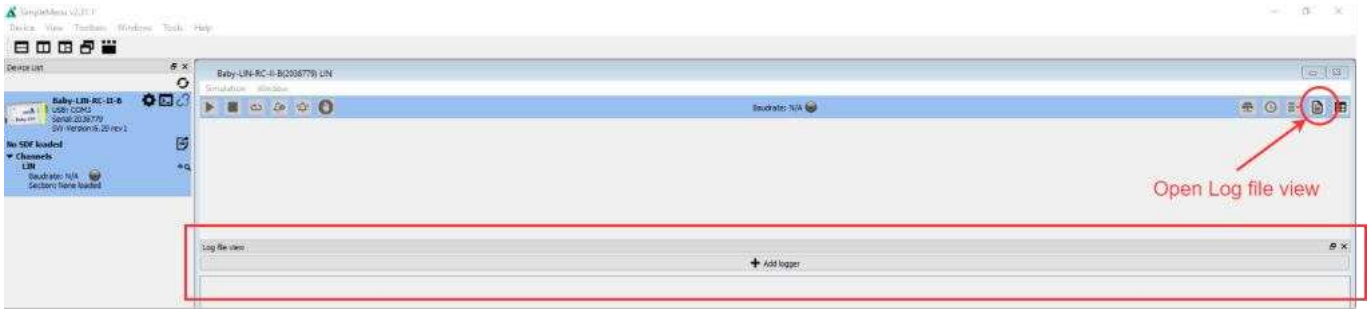
### 6.5.2.1 Overview

The Baby-LIN-II can write log data to the following targets:

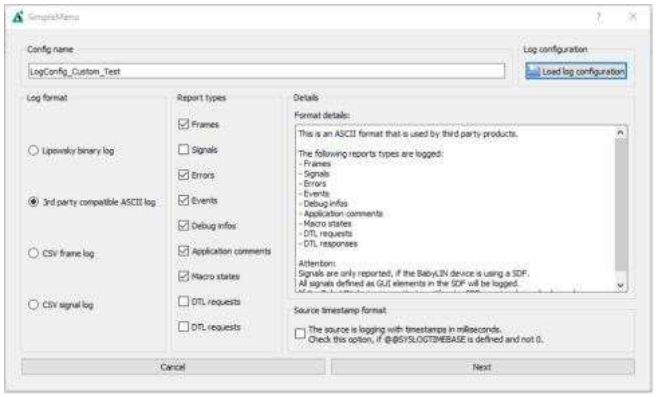
| Target | Description |
|---|---|
| **"SimpleMenu"** | The logging data can be viewed and written into a file by using the "SimpleMenu". |

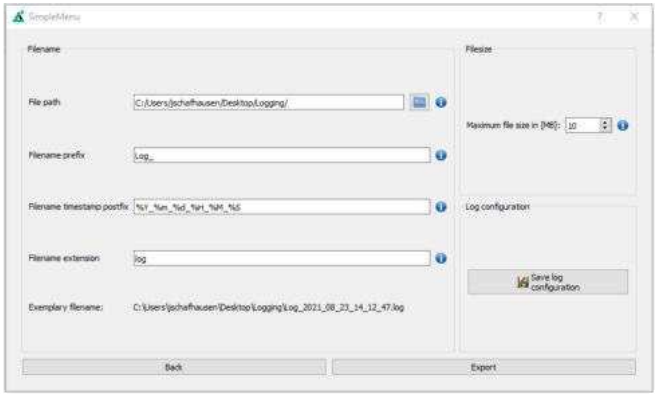
### 6.5.2.2 SimpleMenu

The SimpleMenu offers the possibility to record the complete bus communication of the Baby-LIN-Decvice and to save it locally in a file on the computer. To do this, open the integrated log viewer in the SimpleMenu.




After opening the Log file View, you can now add a logger. Adding the logger opens the Logging Configuration. Existing LOG configurations can be loaded or new ones can be created.





You can customise the logging of bus communication according to your requirements. You can select which data should be tracked and how this data should be saved in the next step.

### 6.5.3 Log data formats

#### 6.5.3.1 Format overview

The Baby-LIN-II is writing log files in the following formats:

| Target | Description | Frame size |
|---|---|---|
| Binary format | This is the default format of the current firmware of the Baby-LIN-II . This format uses a proprietary binary data format to store the data. It is optimized for speed and a low file size. Files in this format can be converted into other formats using the "LogViewer". | 24 Bytes |
| ASCII format | This format is a human readable ASCII format, that can be processed by some third party products. | 88 Bytes |
| Debug format | This format is a human readable ASCII format, that is mainly used for debugging purposes. It does not contain any bus information. | |

#### 6.5.3.2 Binary format

This format uses a proprietary binary data format to store the log data. It is optimized for speed and a low file size. This file can be viewed, edited and converted into other formats using the "LogViewer".

#### 6.5.3.3 ASCII format

This format is a human readable ASCII format. It consists of a header, the logged frame data and comments. It can be processed by many third party products. The ASCII format has the following structure:

```
date Fri May 12 13:38:13 2017
base hex timestamps absolute
internal events logged
// version HARP-4 V.1.43 Build1
  2.788508 Li           11 Tx 1 00                      checksum = ff         CSM = classic
  2.788508 Li           11 Tx 1 00                      checksum = ff         CSM = classic
  3.288498 Li           12 Rx 8 43 a1 16 d0 a7 53 29 00 checksum = 10         CSM = classic
  3.538493 Li           13 Rx 0                         NodeResponseMissing
  3.788488 Li           11 Tx 1 01                      checksum = fe         CSM = classic
  4.288531 Li           12 Rx 8 4d a6 19 d3 a9 54 2a 00 checksum = f6         CSM = classic
  4.538525 Li           13 Rx 0                          NodeResponseMissing
  4.788519 Li           11 Tx 1 02                      checksum = fd         CSM = classic
  5.288509 Li           12 Rx 8 56 ab 1d d5 ab 55 2a 00 checksum = df         CSM = classic
  5.538504 Li           13 Rx 0                          NodeResponseMissing
  5.788499 Li           11 Tx 1 03                      checksum = fc         CSM = classic
  6.288489 Li           12 Rx 8 60 b0 20 d8 ad 56 2b 00 checksum = c6         CSM = classic
  6.538536 Li           13 Rx 0                          NodeResponseMissing
  6.788531 Li           11 Tx 1 04                      checksum = fb         CSM = classic
  7.288521 Li           12 Rx 8 6a b5 23 da af 57 2b 00 checksum = af
  7.538515 Li           13 Rx 0                          NodeResponseMissing
  7.788510 Li           11 Tx 1 05                      checksum = fa         CSM = classic
  8.288500 Li           12 Rx 8 74 ba 27 dd b1 58 2c 00 checksum = 95         CSM = classic
  8.538495 Li           13 Rx 0                          NodeResponseMissing
  8.788490 Li           11 Tx 1 06                      checksum = f9         CSM = classic
  9.288532 Li           12 Rx 8 7e bf 2a df b3 59 2c 00 checksum = 7e         CSM = classic
  9.538527 Li           13 Rx 0                          NodeResponseMissing
  9.788522 Li           11 Tx 1 07                      checksum = f8         CSM = classic
  10.288511 Li          12 Rx 8 88 c4 2d e2 b5 5a 2d 00 checksum = 65         CSM = classic
  10.538506 Li          13 Rx 0                          NodeResponseMissing
  10.788501 Li          11 Tx 1 08                      checksum = f7         CSM = classic
```

The frame line components are explained using the first frame:

| TarComponentget | Description |
|---|---|
| 2.788508 | This is a timestamp. It represents the seconds since the Baby-LIN-II was powered on. |
| Li | A bus identifier |
| 11 | The ID of the frame |
| Tx | The direction of the frame |
| 1 | The lenght of the frame |
| 00 | All data bytes of the frame |
| checksum = ff | The checksum of the frame. |
| CSM = classic | The checkszum type used by the frame |

# 7 Workflow

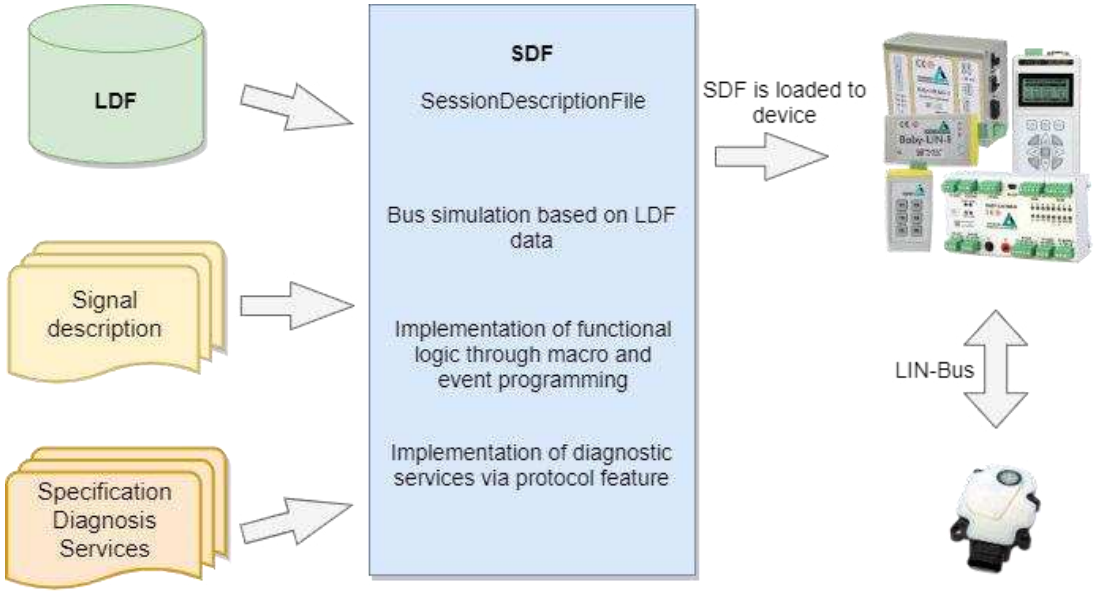
In this chapter we will show you how the workflow looks like in a typical LIN use case. For this purpose, we will introduce the following components to you:

- LDF
- Signal description
- Specification Diagnosis Services

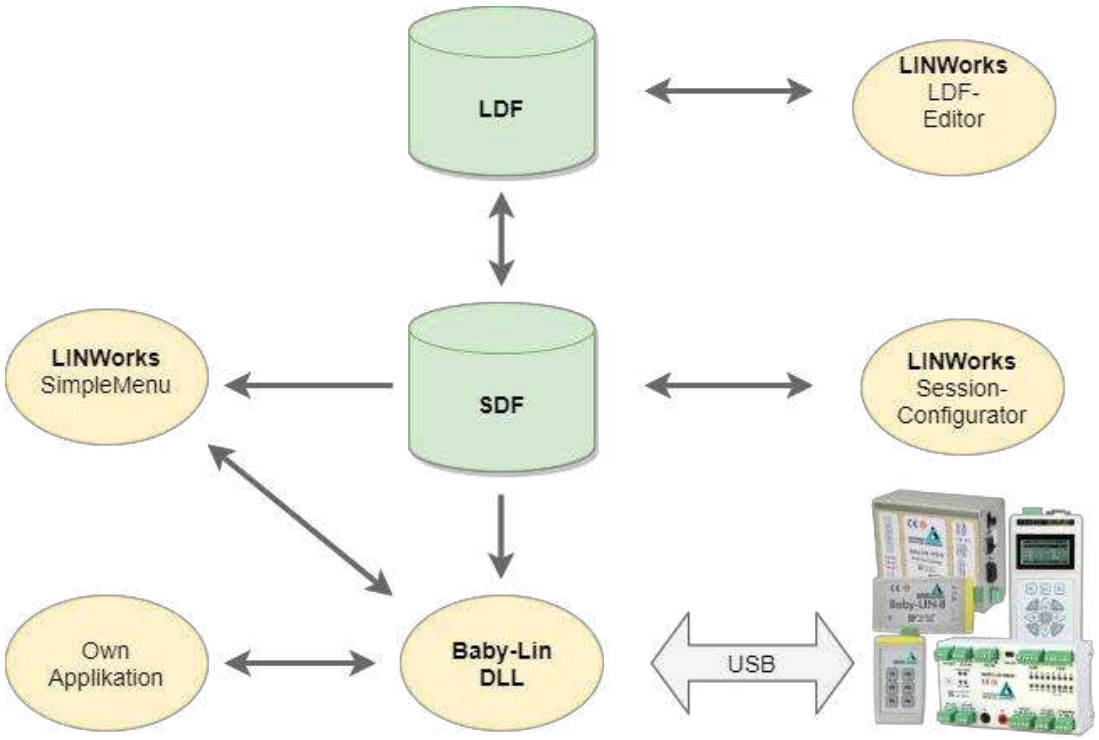
From this information, the SessionDescriptionFile (SDF) can be created. The SDF is the linchpin in LINWorks-based applications.

## 7.1 Overview

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




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




## 7.2 Getting started

### 7.2.1 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-II can be operated in standalone mode, LIN bus data can be logged, or macros can be defined for autostart.

**Advice**
This guide assumes you are using a Microsoft Windows operating system.

### 7.2.2 Installation

Before you can start using the Baby-LIN-II 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 customer portal: *portal.lipowsky.de*

**Advice**
Check chapter "Downloads" for more information.

The following components are required for this getting started guide:

- Baby-LIN driver
- SessionConf
- SimpleMenu
- LDFEdit

## 7.3 LDF

LDF (LinDescriptionFile) has been developed by the LIN Consortium, in which various parties such as car manufacturers, suppliers and tool suppliers were involved. This means that the LDF specification is not dependent on a single manufacturer and can be used universally.The Format and syntax of the LDF are described in the LIN specification.

Each LIN bus has its own LDF, which collects all the properties of this specific bus in one document. This includes which nodes are present on the bus, which frames are defined and according to which scheme they are to be emulated.

### 7.3.1 LDF Example

The following example shows the LDF of a windscreen wiper motor.

**LDF header**

```
LIN_description_file ;
LIN_language_version = "1.3" ;
LIN_speed = 19.200 kbps ;
```

**Node section**

```
Nodes {
Master:MasterECU,1.0000 ms,0.1000 ms ;
Slaves:Slave1Motor,Slave2Sensor;
}
```

**Frame section**

```
{ MessageCounter:8,0x00,MasterECU,Slave1Motor,Slave2Sensor;

Ignition:1,0x0,MasterECU,Slave1Motor,Slave2Sensor;
WiperSpeed:3,0x0,MasterECU,Slave1Motor;
Temperature:8,0xFF,MasterECU,Slave1Motor,Slave2Sensor;
WiperActive:1,0x0,Slave1Motor,MasterECU;
ParkPosition:1,0x0,Slave1Motor,MasterECU;
CycleCounter:16,0x00,Slave1Motor,MasterECU;
StatusSensor:8,0x00,Slave2Sensor,MasterECU;
ValueSensor:8,0x00,Slave2Sensor,MasterECU;
}

Frames {
MasterCmd:0x10,MasterECU,4{MessageCounter,0;
Ignition,8; WiperSpeed,9; Temperature,16; }
MotorFrame:0x20,Slave1Motor,4{ WiperActive,0;
ParkPosition,1; CycleCounter,16; }
SensorFrame:0x30,Slave2Sensor,2StatusSensor,0; ValueSensor,8;
}
```

**Schedule table**

```
Schedule_tables {
Table1 { MasterCmd delay 20.0000 ms ;
MotorFrame delay 20.0000 ms ;
SensorFrame delay 20.0000 ms ;}
}
```

**Signal section**

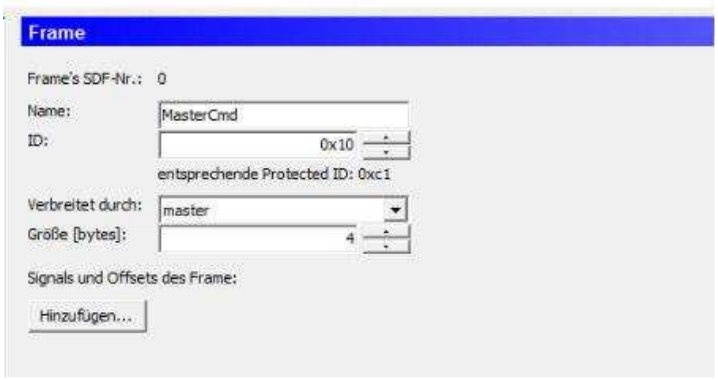
**Signal encoding section**

```
Signals
Signal_encoding_types {
EncodingSpeed { logical_value,0x00,"Off" ;
logical_value,0x01,"Speed1" ;
logical_value,0x02,"Speed2" ;
logical_value,0x03,"Interval" ;}
EncodingTemp { physical_value,0,253,0.8,- 35,"degrees C" ;
logical_value,0xFE,"Signal not supported" ;
logical_value,0xFF,"Signal not available" ;}
}

Signal_representation
{ EncodingSpeed:WiperSpee
d;
EncodingTemp:Temperature;
}
```

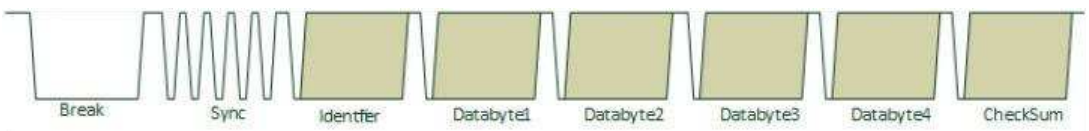
### 7.3.2 LIN application frames

With the information from an LDF, you can assign all frames that appear on the bus to your publisher using the PID. You can also interpret the data regarding the signals it contains.




```
Frames{
    MasterCmd:0x10,MasterECU,4 {
        MessageCounter,0;
        Ignition,8;
        WiperSpeed,9;
        Temperature,16; }
        }
```

The frame is structured as shown in the following graphic. The frame defined in the LDF is recognised with the identifier with ID = 0x10 and the signals can be mapped from the 4 databytes.




#### 7.3.2.1 Protected LIN identifier

The Frame Id is 8 Bit in size, where by the upper 2 bits are used as parity bits. So only 6 bits remains to represent the effective frame identifier. This

makes a range of 64 different frame id's.

| Paritybit P1 (ID.7) | Paritybit P0 (ID.6) | Identifier Bits ID.5 - ID.0 |
|---|---|---|
| ID.1^ID.3^ID.4^ID.5 | !(ID.0^ID.1^ID.2^ID.4) | 0...63 |

| Id dec | Id hex | PID | Id dec | Id Hex | PID | Id dec | Id hex | PID | Id dec | Id hex | PID |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0x00 | 0x80 | 16 | 0x10 | 0x50 | 32 | 0x20 | 0x20 | 48 | 0x30 | 0xF0 |
| 1 | 0x01 | 0xc1 | 17 | 0x11 | 0x11 | 33 | 0x21 | 0x61 | 49 | 0x31 | 0xB1 |
| 2 | 0x02 | 0x42 | 18 | 0x12 | 0x92 | 34 | 0x22 | 0xE2 | 50 | 0x32 | 0x32 |
| 3 | 0x03 | 0x03 | 19 | 0x13 | 0xD3 | 35 | 0x23 | 0xA3 | 51 | 0x33 | 0x73 |
| 4 | 0x04 | 0xc4 | 20 | 0x14 | 0x14 | 36 | 0x24 | 0x64 | 52 | 0x34 | 0xB4 |
| 5 | 0x05 | 0x85 | 21 | 0x15 | 0x55 | 37 | 0x25 | 0x25 | 53 | 0x35 | 0xF5 |
| 6 | 0x06 | 0x06 | 22 | 0x16 | 0xD6 | 38 | 0x26 | 0xA6 | 54 | 0x36 | 0x76 |
| 7 | 0x07 | 0x47 | 23 | 0x17 | 0x97 | 39 | 0x27 | 0xE7 | 55 | 0x37 | 0x37 |
| 8 | 0x08 | 0x08 | 24 | 0x18 | 0xD8 | 40 | 0x28 | 0xA8 | 56 | 0x38 | 0x78 |
| 9 | 0x09 | 0x49 | 25 | 0x19 | 0x99 | 41 | 0x29 | 0xE9 | 57 | 0x39 | 0x39 |
| 10 | 0x0A | 0xCA | 26 | 0x1A | 0x1A | 42 | 0x2A | 0x6A | 58 | 0x3A | 0xBA |
| 11 | 0x0B | 0x8B | 27 | 0x1B | 0x5B | 43 | 0x2B | 0x2B | 59 | 0x3B | 0xFB |
| 12 | 0x0C | 0x4C | 28 | 0x1C | 0x9C | 44 | 0x2C | 0xEC | 60 | 0x3C | 0x3C |
| 13 | 0x0D | 0x0D | 29 | 0x1D | 0xDD | 45 | 0x2D | 0xAD | 61 | 0x3D | 0x7D |
| 14 | 0x0E | 0x8E | 30 | 0x1E | 0x5E | 46 | 0x2E | 0x2E | 62 | 0x3E | 0xFE |
| 15 | 0x0F | 0xCF | 31 | 0x1F | 0x1F | 47 | 0x2F | 0x6F | 63 | 0x3F | 0xBF |

**Advice**
Note that the following IDs are reserved for protocol extensions and diagnostic and configuration data:
• 60 (0x3C) and 61 (0x3D) are used to carry diagnostic and configuration data.
• 62 (0x3E) and 63 (0x3F) are reserved for future protocol enhancements.

### 7.3.3 LIN Scheduling

The order in which the frames are sent to the LIN bus is defined in a so-called Schedule Table. One or more Schedule Table(s) are defined in each LDF.

Each table entry describes a frame by its LDF name and a delay time, which is the time that is made available to the frame on the bus.



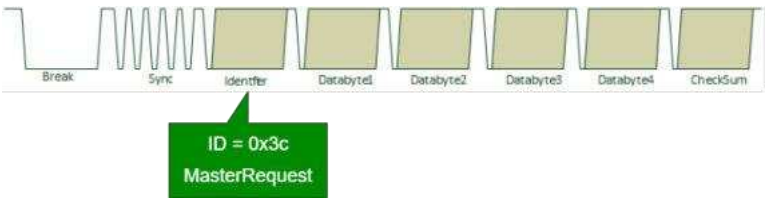

A Schedule Table is always selected as active and is executed by the master.The master places the corresponding frame headers on the bus and the publisher assigned to this frame places the corresponding data section + checksum on the bus.

Only the master can switch the Schedule Table. Thus the master application determines which frames appear on the bus in which time sequence.

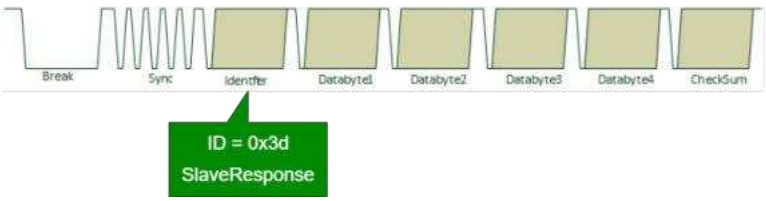
### 7.3.4 LIN Diagnostic frames

Diagnostic frames are a pair of MasterRequest (0x3c) and SlaveResponse (0x3D) frames. Used to send information that is not described in the LDF.

0x3C MasterRequest:
Request Data define the node and the requested action.



0x3D SlaveResponse:
Data generated by the addressed slave; content depends on request




The Master Request and Slave Response have special properties:

- They are always 8 bytes long and always use the Classic Checksum.
- No static mapping of frame data to signals; frame(s) are containers for transporting generic data.
- Request and response data can consist of more than 8 data bytes.

The MasterRequest - SlaveResponse mechanism can be used to transmit a wide variety of data because it is a universal transport mechanism. A main application is the diagnosis and End of Line (EOL) configuration of nodes.

In the field there is a whole range of different protocols, depending on the vehicle and ECU manufacturer:

- A lot of proprietary diagnostics or EOL protocols
- DTL based protocols (Diagnostic Transport Layer)
- Keyword 2000 Protocol (ISO 14230 -1 to 4)
- UDS (Unified Diagnostic Services) (ISO 14229-1:2013)

**Advice**
A more detailed explanation of all the possibilities of using Unified Diagnosis Services can be found in the chapter "Diagnostic Modes".

## 7.4 Session Description File (SDF)

### 7.4.1 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.

**2. Requirement**




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**




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).**

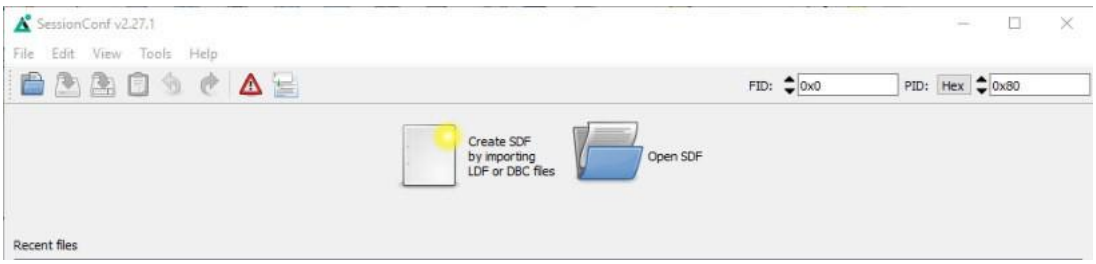
### 7.4.2 Introduction

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.

### 7.4.3 Create a SDF

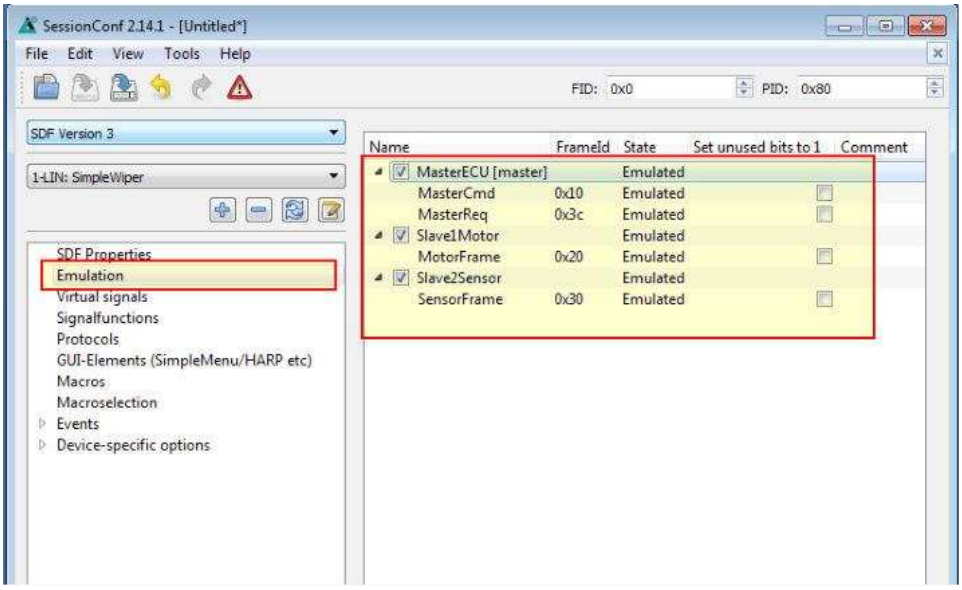
The SessionConf software application is used to create and edit the SDF. For this purpose, an existing LDF is imported.




### 7.4.4 Common Setup

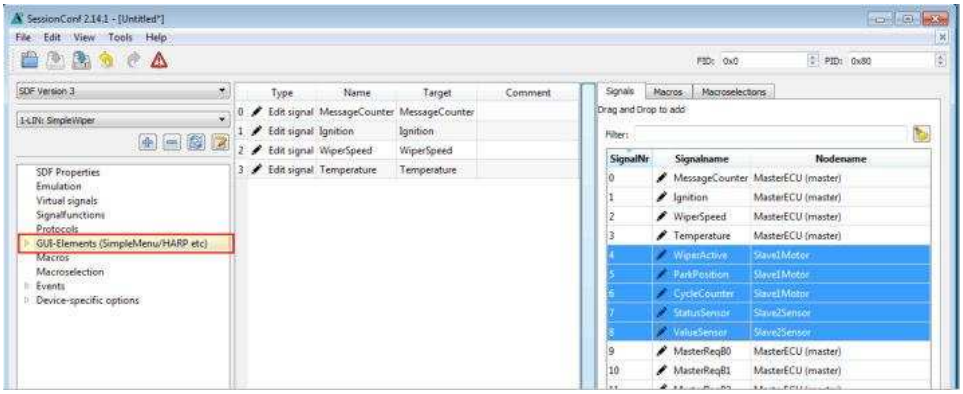
#### 7.4.4.1 Emulation

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




### 7.4.4.2 GUI-Elements

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



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

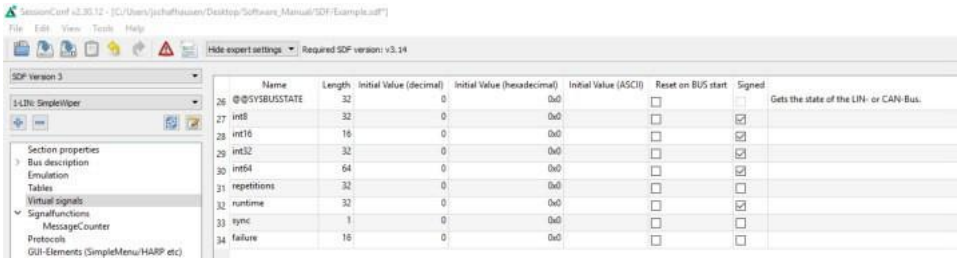
### 7.4.4.3 Virtual signals

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
- Store constants
- Operands and results from calculations
- etc.

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

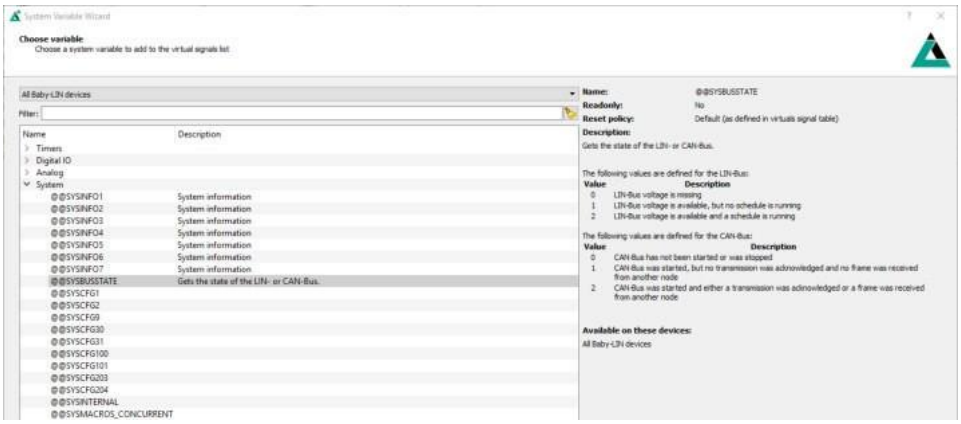
Each signal has a default value that is set when the SDF is loaded.



### 7.4.4.4 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 behaviour.

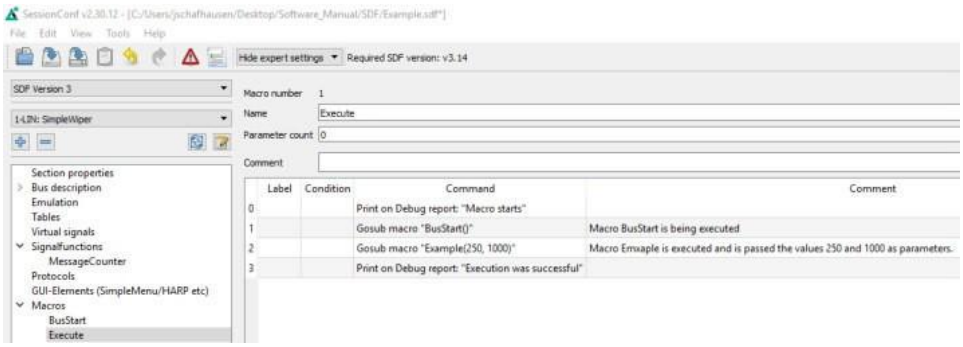
In this way, you can access timer, input and output resources and system information.



**Advice**
For more information and a list of all available system signals, please check the chapter "System variables".

### 7.4.4.5 Macros

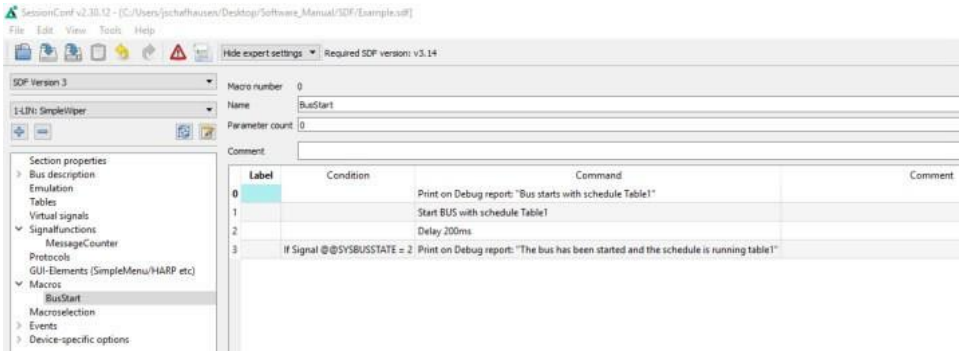
Macros are used to combine multiple operations into a sequence. Macros can be started by events or, can also be called from other macros in the sense of a Goto or Gosub. The DLL API calls a macro with the macro_execute command.



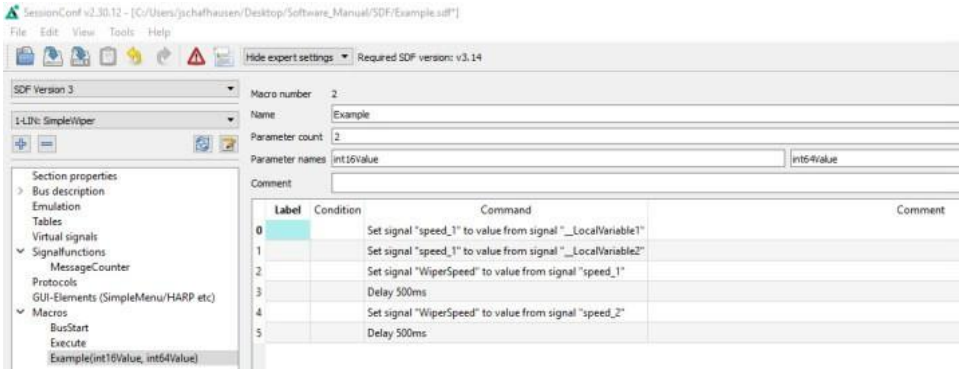
All Macro Commands can use signals from the LDF and signals from the Virtual Signal section like the system signals.

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.

**Advice**
For more information or if you want to implement an autostart macro, please read the chapter "Configuring an autostart macro".




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.




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 `_LocalVariableX` signals receive the value 0.

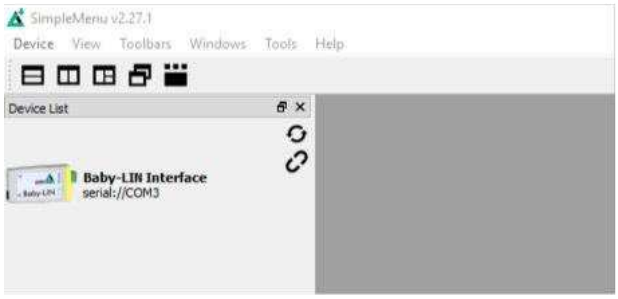
#### 7.4.4.6 Embbeded SDF

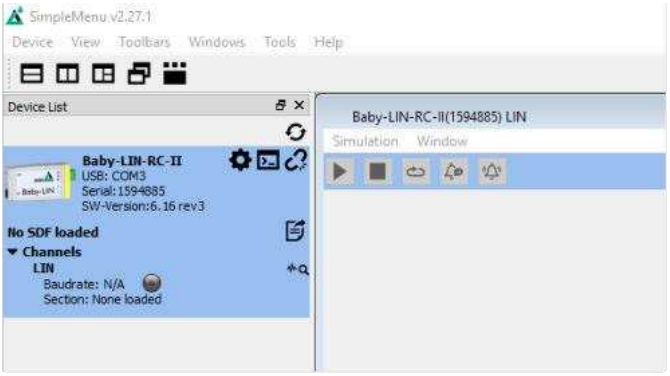
You can download the sample SDFs in the download area on our website under the following link.

Link: *https://www.lipowsky.de/downloads/*

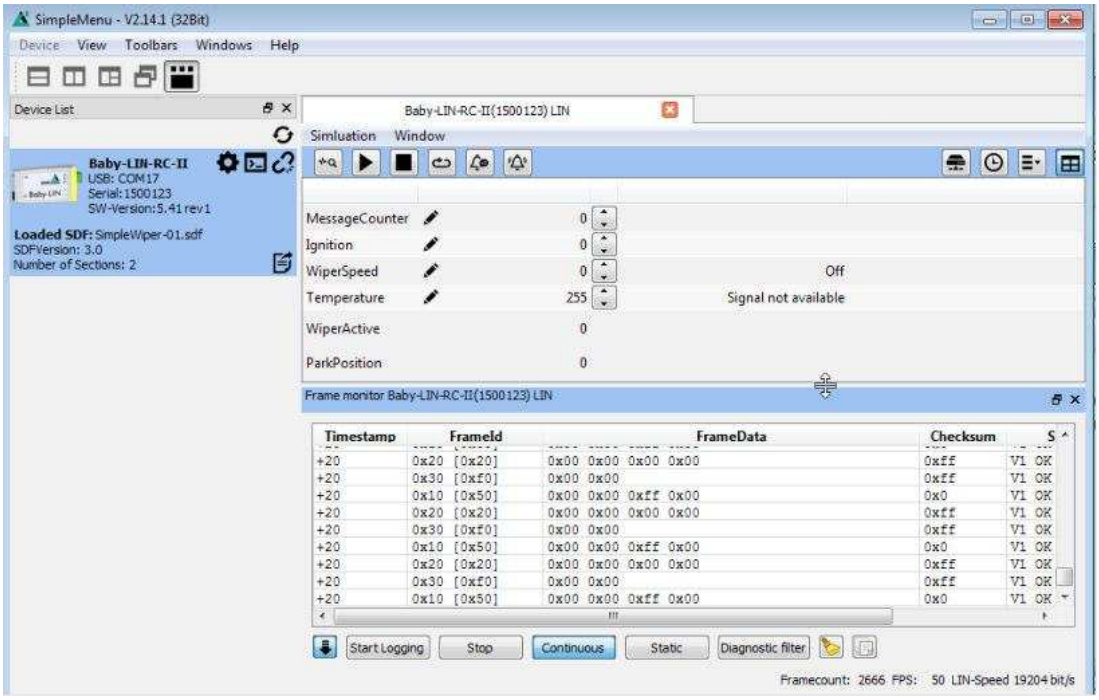
### 7.4.5 Start the bus communication

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




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




Now you will see the changes of these signals.

**Tip**
The Baby-LIN-II features a lot more features and possibilities and can be used for a wide selection of applications. Keep on reading this manual to learn more about the Baby-LIN-II .

# 8 LINWorks Software - Overview

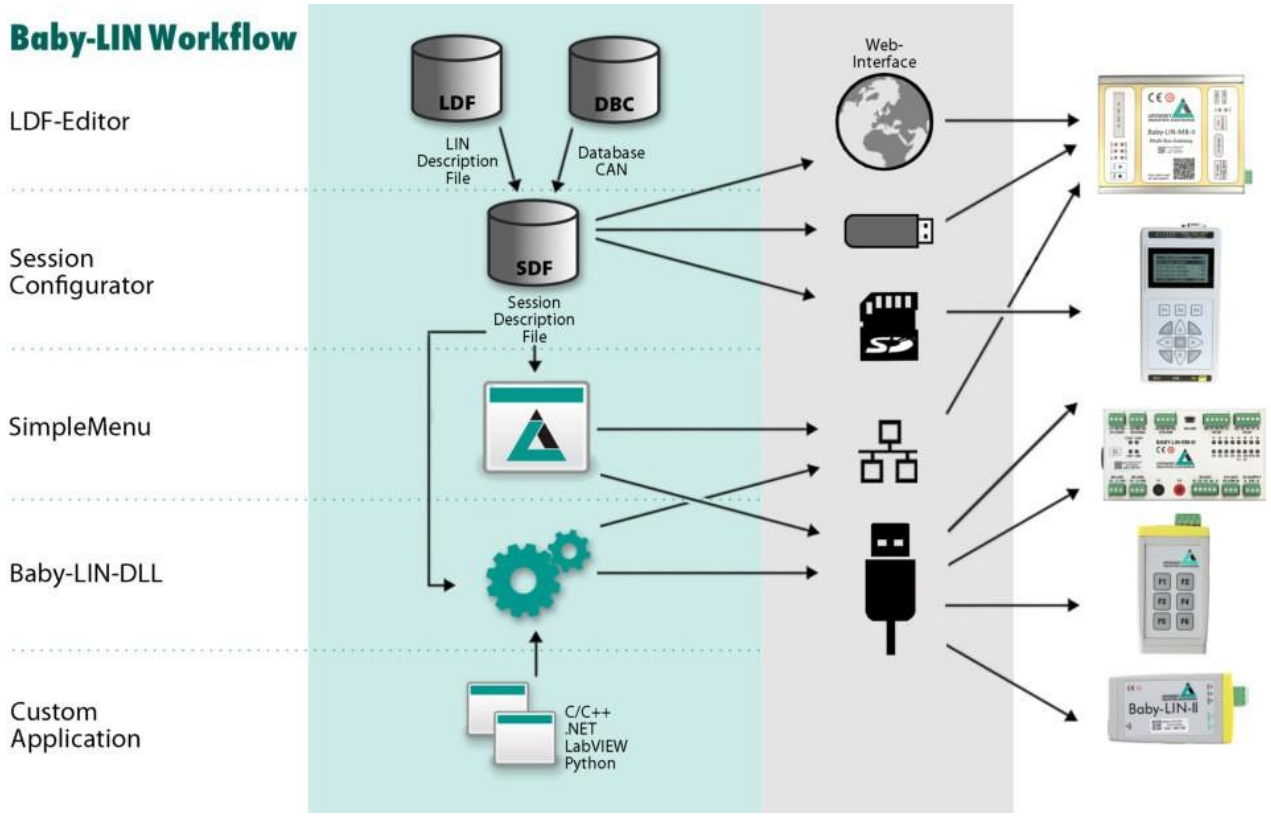
The LINWorks is a collection of software to operate the Baby-LIN-II.The complete LINWorks software package is available for download on our website. There you will also find the LINWorks Software Manual, which gives a detailed overview of the individual program and how to work with and create Session Description Files.

You can download both from the following link: *https://www.lipowsky.de/downloads/*



| Component | Archive subfolder | Description |
|---|---|---|
| Datasheets User manuals Application notes | Documentation | The datasheets show a quick overview about a Baby-LIN product and its features. The user manuals contain the main documentation. The application notes contain some older information, that have not been added to the user manuals yet. |
| **"Baby-LIN driver"** | | The **Baby-LIN driver** is necessary to connect a Baby-LIN-II to a windows PC via USB. The Baby-LIN-II will be available as **virtual COM port**. |
| **"LDFEdit"** | LINWokrs | The **LDFEdit** allows the inspection, creation and edit of a LDFile (LIN Description File). |
| **"SessionConf"** | LINWokrs | The **SessionConf** allows the inspection, creation and edit of a SDFile (Session Description File) and features a file import for LDFiles (for LIN-Bus simulation). It defines everything needed for a complete simulation of each available bus, e.g. which nodes on each bus are available and which nodes should be simulated by the Baby- LIN-RC-II. Moreover it allows defining an application logic. This programming ability is available for each device out of the box. |
| **"SimpleMenu"** | LINWokrs | The **SimpleMenu** is used to establish a connection to the Baby-LIN-II and upload SDFiles, change the device target configuration, control the bus and monitor the frames and signals on the bus. Even without a LDFile/ SDFile the bus can be monitored and the frames can be logged. |
| **"LogViewer"** | LINWorks | The **LogViewer** can show and convert the log files of the Baby-LIN-II as well as the SimpleMenu. |
| **"Baby-LIN-DLL"** | Development | The **Baby-LIN-DLL** allows customers to create their own application and use all features of the Baby-LIN-II like controlling and monitoring the LIN-Bus interfaces. The **Baby-LIN-DLL** is a native **C/C++** DLL. It is available for **Windows**, **Linux** and **RaspberryPi**. Wrapper for **.NET**, **Python**, **VB6** and **LabView** are available. Of course we provide examples for all supported languages. |
| **"Serial writer"** | Tools | The **serial writer** is used to change the **serial number**, that is stored within the persistent memory of a Baby-LIN-II . This serial number influences the allocation of the **virtual COM port** number, the Baby-LIN-II is available under. |
| **"BLProg"** | Tools | The **BLProg** is used to update the **firmware** of a Baby- LIN-RC-II. If you download a firmware package from our customer portal (*portal.lipowsky.de*) a current version of the **BLProg** will always be included. |

The following graphic shows how you can use our LINWorks software in connection with our the Baby-LIN-Devices.



# 9 Migration information

## 9.1 Migration from Baby-LIN-II to Baby-LIN

All Baby-LIN products of the second generation were designed to be compatible with the first generation. Due to hardware and software changes, the compatibility may be affected in certain areas.

If you have used a Baby-LIN in your environments and applications and now want to replace it with a Baby-LINII, the following chapters give you an overview of the topics, you have to consider.



Version incompatitbility
Each of the following chapters may decrease the compatibility depending on your application and the way, you use the Baby-LIN-II. Therefor you should check all these chapters very carefully.

## 9.2 Performance

The Baby-LIN products of the second generation are in common more powerful.

The faster and more powerful CPU executes operations faster and therefor allows more operations in the same time interval. The higher memory allows to download bigger SDFiles into the Baby-LIN-II.

The SDF-V3 format allows to use new powerful features within the SDFile.

| | **Baby-LIN** | **Baby-LIN-II** |
|---|---|---|
| CPU | ARM-7, 60 Mhz | ARM Cortex-M4, 168 MHz |
| Memory | 32 kB RAM | 196 kB RAM |
| SDF-Version | SDF-V2 | SDF-V3 |

## 9.3 LIN-Bus transceiver



Version incompatitbility
If you want to replace a Baby-LIN with a Baby-LIN-II you should check the following chapter, since this issue reduce the compatibility depending on your application and the way, you use the Baby-LIN-II.

The used LIN-Bus transceiver has changed. The following table shows you, what properties have changed:

| Baby-LIN product | **Baby-LIN** | **Baby-LIN-II** |
|---|---|---|
| LIN-Bus transceiver | Si9241A | MC33662 |
| Maximum LIN-Bus baudrate | 200 kBaud | 125 kBaud |
| Minimum LIN-Bus voltage | 3,8 V | 6,9 V |
| | We recommend a minimum LIN-Bus voltage of 8 V. | |
| Maximum LIN-Bus voltage | 36 V | 26 V |

