ModusToolbox™ LIN Configurator user guide

ModusToolbox™ tools package version 3.1.0

LIN Configurator version 1.21

About this document

Scope and purpose

The Local Interconnected Network (LIN) Configurator is used to generate build-time configuration for the LIN middleware.

Intended audience

This document helps application developers understand how to use the LIN Configurator as part of creating a ModusToolbox™ application.

Document conventions

Convention	Explanation
Bold	Emphasizes heading levels, column headings, menus and sub-menus
Italics	Denotes file names and paths.
`Courier New`	Denotes APIs, functions, interrupt handlers, events, data types, error handlers, file/folder names, directories, command line inputs, code snippets
File > New	Indicates that a cascading sub-menu opens when you select a menu item

Abbreviations and definitions

The following define the abbreviations and terms used in this document:

- API – application programming interface
- LIN – local interconnected network
- LDF – A LIN Description File describes a complete LIN system
- NCF – A Node Capability File describes a LIN slave node as seen from the LIN bus
- NAD – network address
- MRF – multiple reference frame
- SRF – single reference frame

Reference documents

Refer to the following documents for more information as needed:

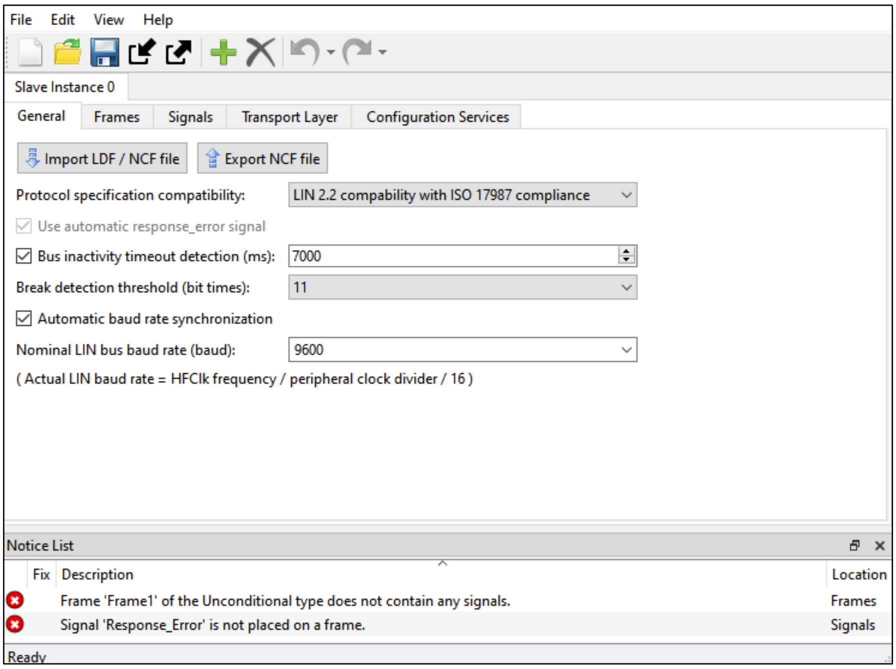
- Eclipse IDE for ModusToolbox™ user guide
- ModusToolbox™ user guide
- Device Configurator user guide
- Device datasheets

- Device technical reference manuals
- LIN Middleware API reference guide

Table of contents

1 Overview

The Local Interconnected Network (LIN) Configurator is a stand-alone tool included with the ModusToolbox™ software and used to generate build-time configuration for the LIN middleware. LIN is a low-cost serial communication standard used for various electronic components inside automobiles. The tool usage is an effective alternative to manual modification of the configuration structure. The tool is supported on Windows, Linux, and macOS.



Protocol specification compatibility: LIN 2.2 compability with ISO 17987 compliance

☑ Use automatic response_error signal

☑ Bus inactivity timeout detection (ms): 7000

Break detection threshold (bit times): 11

☑ Automatic baud rate synchronization

Nominal LIN bus baud rate (baud): 9600

(Actual LIN baud rate = HFClk frequency / peripheral clock divider / 16)

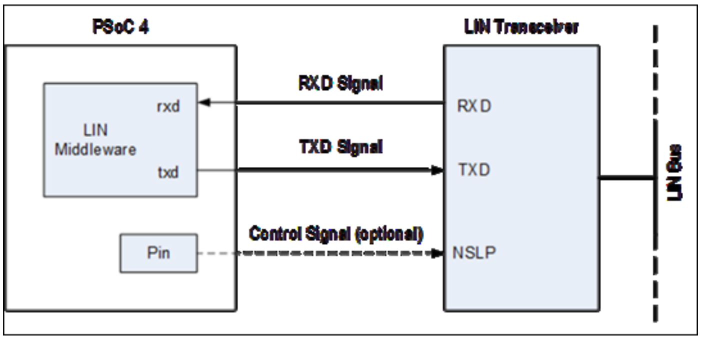
Fix	Description	Location
☒	Frame 'Frame1' of the Unconditional type does not contain any signals.	Frames
☒	Signal 'Response_Error' is not placed on a frame.	Signals

1.1 Supported middleware

The LIN Configurator middleware uses configuration setups developed in the LIN Configurator GUI or with external applications.

The LIN middleware implements a LIN 2.2 and ISO17987 compliance slave node on PSoC™ 4 devices. This middleware provides top-level APIs to allow the application code to easily interact with the LIN bus communication.

Hardware Interface between PSoC™ 4 and LIN Bus.



Name		Version	Link
LIN Middleware		1.0	https://infineon.github.io/lin/html/index.html

2 Launch the LIN Configurator

There are several ways to launch the LIN Configurator, and those ways depend on the application, and how you use the various tools.

2.1 make command

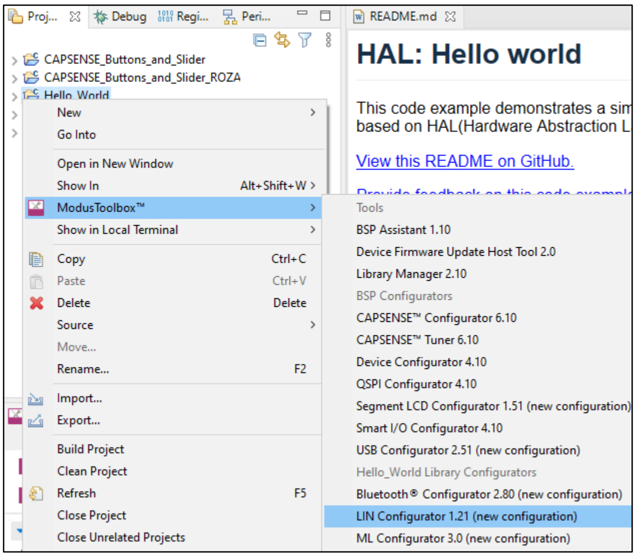
As described in the ModusToolbox™ user guide "Build System" chapter, you can run numerous `make` commands in the application directory, such as launching the LIN Configurator. After you have created a ModusToolbox™ application, navigate to the application directory and type the following command in the appropriate bash terminal window:

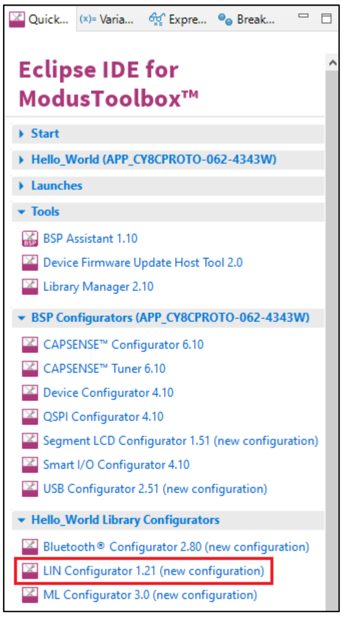
```
make lin-configurator
```

This command opens the LIN Configurator GUI for the specific application in which you are working.

2.2 Eclipse IDE

If you use the Eclipse IDE for ModusToolbox™, you can launch the LIN Configurator for the selected application. In the Project Explorer, right-click on the project and select **ModusToolbox™ > LIN Configurator <version>**. You can also click the LIN Configurator link in the IDE Quick Panel.


Proj...
Debug
Regi...
Peri...
README.md
CAPSENSE_Buttons_and_Slider
CAPSENSE_Buttons_and_Slider_ROZA
Hello_World
HAL: Hello world
This code example demonstrates a sim
based on HAL(Hardware Abstraction L
View this README on GitHub.
Provide feedback on this code example
New
Go Into
Open in New Window
Show In Alt+Shift+W
ModusToolbox™
Show in Local Terminal
Copy Ctrl+C
Paste Ctrl+V
Delete Delete
Source
Move...
Rename... F2
Import...
Export...
Build Project
Clean Project
Refresh F5
Close Project
Close Unrelated Projects
Tools
BSP Assistant 1.10
Device Firmware Update Host Tool 2.0
Library Manager 2.10
BSP Configurators
CAPSENSE™ Configurator 6.10
CAPSENSE™ Tuner 6.10
Device Configurator 4.10
QSPI Configurator 4.10
Segment LCD Configurator 1.51 (new configuration)
Smart I/O Configurator 4.10
USB Configurator 2.51 (new configuration)
Hello_World Library Configurators
Bluetooth® Configurator 2.80 (new configuration)
LIN Configurator 1.21 (new configuration)
ML Configurator 3.0 (new configuration)


Quick...
Varia...
Expre...
Break...
Eclipse IDE for ModusToolbox™
Start
Hello_World (APP_CY8CPROTO-062-4343W)
Launches
Tools
BSP Assistant 1.10
Device Firmware Update Host Tool 2.0
Library Manager 2.10
BSP Configurators (APP_CY8CPROTO-062-4343W)
CAPSENSE™ Configurator 6.10
CAPSENSE™ Tuner 6.10
Device Configurator 4.10
QSPI Configurator 4.10
Segment LCD Configurator 1.51 (new configuration)
Smart I/O Configurator 4.10
USB Configurator 2.51 (new configuration)
Hello_World Library Configurators
Bluetooth® Configurator 2.80 (new configuration)
LIN Configurator 1.21 (new configuration)
ML Configurator 3.0 (new configuration)

Similar to the `make` command method, launching a configurator using the Eclipse IDE opens the configuration file used by the selected application. If the application doesn't have an appropriate configuration file, the configurator opens with a new configuration. For details, refer to the Eclipse IDE for ModusToolbox™ user guide.

2.3 Executable (GUI)

You can launch the LIN Configurator GUI by running its executable as appropriate for your operating system (for example, double-click it or select it using the Windows **Start** menu). By default, the configurator is installed here:

<install_dir>/ModusToolbox/tools_<version>/lin-configurator<version>

When launched this way, the LIN Configurator opens without any frames or signals configured. You can either open a specific configuration file or create a new one. See Menus for more information.

Launching the configurator this way is not recommended as part of application development.

2.4 Executable (CLI)

The LIN Configurator executable can be run from the command line and it has a "cli" version of the executable as well. Running the executable from the command line can be useful as part of batch files or shell scripts to re-generate the source code based on the latest configuration settings. The exit code for the executable is zero if the operation is successful, or non-zero if the operation encounters an error. For more information about the command-line options, run the executable using the `-h` option.

3 Quick start

This section provides a simple workflow for how to use the LIN Configurator.

1. Launch the LIN Configurator.
2. Configure parameters as required.
3. Save the configuration. See Code generation.
4. Use the generated structures from *mtbcfg_lin.h* and *mtbcfg_lin.c* files as input parameters for functions in your application.

4 Code generation

The LIN Configurator generates header (*mtbcfg_lin.h*) and source (*mtbcfg_lin.c*) files that contain relevant firmware used by the LIN middleware configuration and operation. The generated .h and .c files are located in the *GeneratedSource* folder next to the **. mtblin* file which contains the user configuration.

Refer to the LIN Middleware Library for more information about this code.

5 GUI description

The LIN Configurator GUI contains menus, tabs and subtabs to configure the LIN settings and a Notice List to provide indications.

5.1 Menus

5.1.1 File

- **New** – Creates a new file with new configuration.
- **Open…** – Opens the configuration file.
- **Close** – Closes the configuration file.
- **Save** – Saves the existing configuration.
- **Save As…** – Saves the existing file under a different name.
- **Open in System Explorer** – Opens your computer's file explorer tool to the folder that contains the *.modus file.
- **Import…** – Imports a specified configuration file.
- **Export…** – Exports the current configuration file into a specified file.
- **Recent files** – Shows recent files that you can open directly.
- **Exit** – Closes the configurator.

5.1.2 Edit

- **Add new instance** – You can start a new instance by either clicking this button or the same button on the Multi-instance support.
- **Delete current instance** – You can remove the current instance by either clicking this button or the same button on the Multi-instance support.
- **Undo** – Undoes the last action or sequence of actions.
- **Redo** – Redoes the last undone action or sequence of undone actions.

5.1.3 View

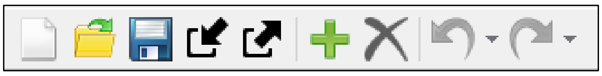
- **Notice List** – Shows/hides Notice List. See the Device Configurator user guide for more information about the Notice List.
- **Toolbar** – Shows/hides the toolbar.
- **Reset View** – Resets the view to the default.

5.1.4 Help

- **View Help** – Opens this document.
- **About LIN Configurator** – Opens the About box for version information, with links to open https://www.infineon.com and the current session log file.

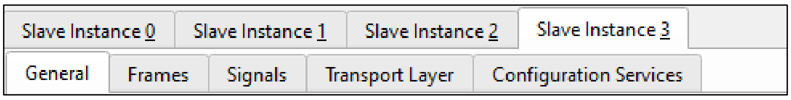
5.2 Toolbar

The toolbar contains common commands from the File and Edit menus. Use the check box under the View menu to show or hide the toolbar.



5.3 Multi-Instance support

The LIN Configurator supports multiple slave instances to configure different settings for each instance. By default, the configurator contains only one instance, which is Slave Instance 0. The configurator includes commands to add and remove instances.



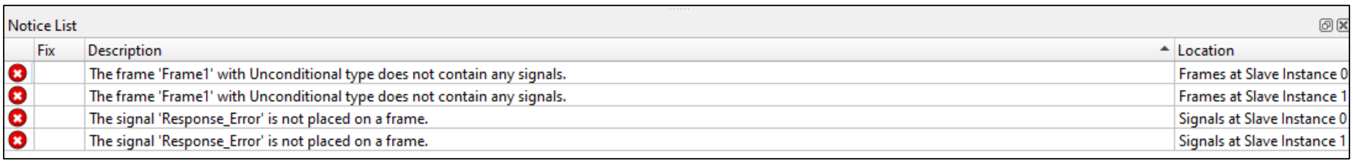
5.4 Parameter subtabs

Each slave instance contains subtabs to configure various settings. These include:

- General subtab
- Frames subtab
- Signals subtab
- Transport Layer subtab
- Configuration Services subtab

5.5 Notice List

The **Notice List** pane combines notices (errors, warnings, tasks, and notes) from many places in the configuration into a centralized list. If a notice shows a location, you can double-click the entry to show the error or warning.



	Fix	Description	Location
⊗		The frame 'Frame1' with Unconditional type does not contain any signals.	Frames at Slave Instance 0
⊗		The frame 'Frame1' with Unconditional type does not contain any signals.	Frames at Slave Instance 1
⊗		The signal 'Response_Error' is not placed on a frame.	Signals at Slave Instance 0
⊗		The signal 'Response_Error' is not placed on a frame.	Signals at Slave Instance 1

The **Notice List** pane contains the following columns:

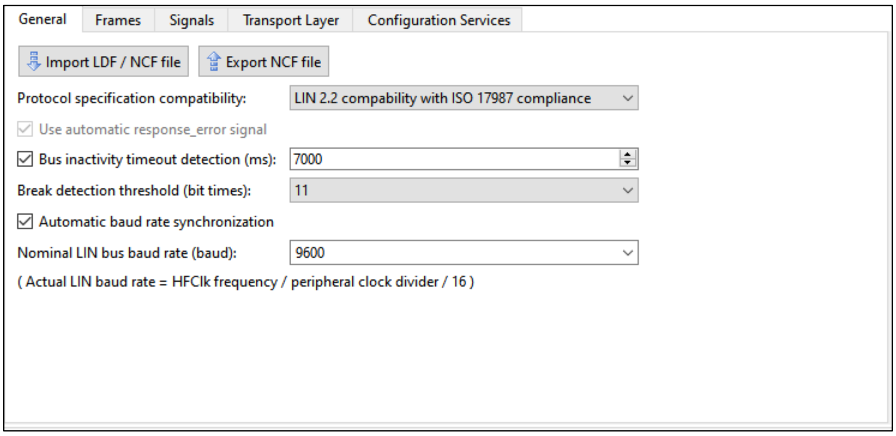
- **Icon** – Displays the icons for the error, warning, task, or note.
- **Fix** – This may display a wrench icon that can be used to automatically address the required notice.
- **Description** – Displays a brief description of the notice.
- **Location** – Displays the specific subtab of the message, when applicable.

For more information about the Notice List, refer to the Device Configurator user guide.

6 Parameters description

6.1 General subtab

The **General** subtab allows general configuration of the LIN resource. The subtab looks as follows:



Protocol specification compatibility: LIN 2.2 compability with ISO 17987 compliance

☑ Use automatic response_error signal

☑ Bus inactivity timeout detection (ms): 7000

Break detection threshold (bit times): 11

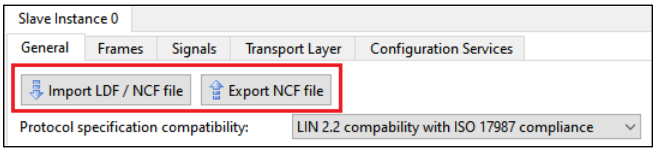
☑ Automatic baud rate synchronization

Nominal LIN bus baud rate (baud): 9600

(Actual LIN baud rate = HFClk frequency / peripheral clock divider / 16)

6.1.1 Slave instance properties

The LIN configurator supports importing NCF and LDF files to configure the LIN slave node and also to export the current configuration into an NCF file. Also, the configurator automatically configures signals based on LDF/NCF files supplied by the user.



Protocol specification compatibility: LIN 2.2 compability with ISO 17987 compliance

6.1.1.1 Import LDF / NCF file

Clicking this button allows you to import an LDF or NCF file. The imported file configures settings to match the configuration of the node selected from the list of the existing nodes of the NCF/LDF file. After the node to import is selected and the import is completed, a dialog displays to inform on the importing results. They contain the LIN slave configuration parameters that were not affected during importing.

6.1.1.2 Export NCF file

This option enables you to save information about the LIN slave configuration into an NCF file.

6.1.2 Protocol specification compatibility

This parameter supports the following functionality to select from the drop-down menu:

- LIN 2.2 compatibility with ISO 17897 compliance
- LIN 2.2 compatibility without ISO 17897 compliance

6.1.3 Use automatic response_error_signal

This option enables the response_error signal option (response error notification to the LIN master).

If this option is selected, the “RE” signal is added on the Unplaced signals pane on the Signals subtab.

6.1.4 Bus inactivity timeout detection (ms)

Select this option to automatically detect the LIN bus inactivity threshold with a free-running timer. The valid range is 4000 ms to 10000 ms.

6.1.5 Break detection threshold (bit times)

Select this option to configure a slave-node break detection threshold. The default value is 11 dominant local-slave bit times. It can be set to either 10 or 11 or 12.

Note: *For the P4000S, P4100S, P4100S Plus, and PSoC™ 4500S series, the actual threshold may be up to one bit less than the specified threshold. For example, if the value of the Break detection threshold option is 11, the actual threshold will be from 10 to 11 bits.*

6.1.6 Automatic baud rate synchronization

By default, this option is enabled. You can disable this option.

- Enabled – The middleware measures the exact baud rate of the bus from the sync byte field of each LIN frame header.
- Disabled – Instead of measuring, the middleware receives the sync byte field as a 0x55 data byte.

Note: *Select this option for successful communication within the cluster if a slave's clock variance is greater than ±1.5 percent.*

Note: *The baud rate synchronization is only used to match one nominal LIN bus baud rate chosen for any given LIN cluster. It is not used to synchronize to any valid baud rate.*

6.1.7 Nominal LIN bus baud rate (baud)

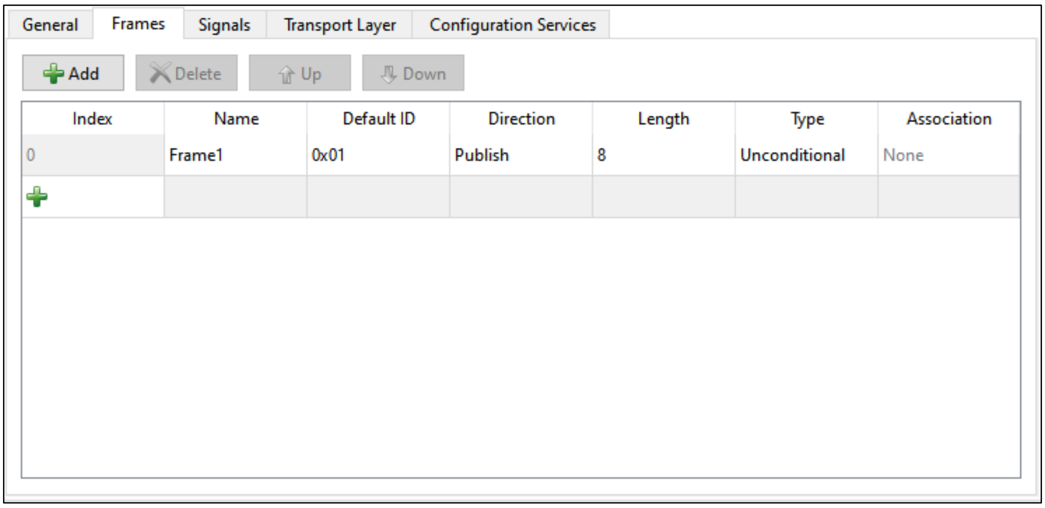
Enter the Nominal LIN Bus Baud Rate at which this slave node must operate. The valid range is 1000 to 20000 baud. The values in the drop-down list are 2400, 9600, 10417, and 19200. Also, you can enter any value between 1000 and 20000 here.

The configurator shows the formula to calculate the actual baud rate:

(Actual LIN baud rate = HFClk frequency / peripheral clock divider / 16)

6.2 Frames subtab

The **Frames** subtab is used to configure how the LIN slave responds to PID values sent by the master on the bus. The subtab contains a toolbar to navigate frames and frame configuration table.

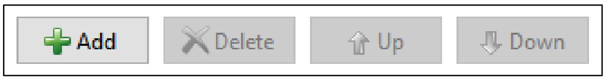


Index	Name	Default ID	Direction	Length	Type	Association
0	Frame1	0x01	Publish	8	Unconditional	None
+						

Note: The Frames subtab does not show diagnostic frames.

6.2.1 Frames Toolbar

The **Frames** subtab toolbar contains four buttons.



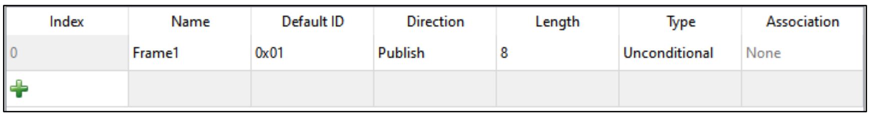
- **Add** – You can start a new frame on the table by either clicking the Add button or the + button at the end of the Index column.
- **Delete** – Removes the currently selected frame from the table. The index number fields are changed accordingly.

Note: If a frame is deleted on this subtab, any signals configured on the Signals subtab and placed into it are relocated into the Unplaced Signals section.

- **Up** /**Down** – Reorder the Index number values for each frame.

6.2.2 Frames configuration table

The table contains eight columns.



Index	Name	Default ID	Direction	Length	Type	Association
0	Frame1	0x01	Publish	8	Unconditional	None
+						

- **Index** – Shows the ordering number of each used frame. These numbers cannot be directly modified.

  You can start a new column clicking the **Add (+)** button at the end of the Index column.
- **Name** – Displays the unique name of each frame. To change the name, click on it.

Parameters description

- **Default ID** – Defines the frame ID that the frame will use before any configuration requests by the master. Note that the LIN master can reconfigure frame IDs at run time. Enter a value from 0x00 to 0x3B in the hex format into these cells.
- **Direction** – Provides two drop-down options for each frame to select the direction to send the data for the frame:
- **Publish** – Data transmission.
- **Subscribe** – Data reception.
- **Length** – Defines how many bytes are received or sent for each frame. The valid range is 1 - 8.
- **Type** – Provides two drop-down options for each frame to select the frame type:
    - **Unconditional**
    - Event-**triggered**

    For more details, refer to the next paragraph **Association**.

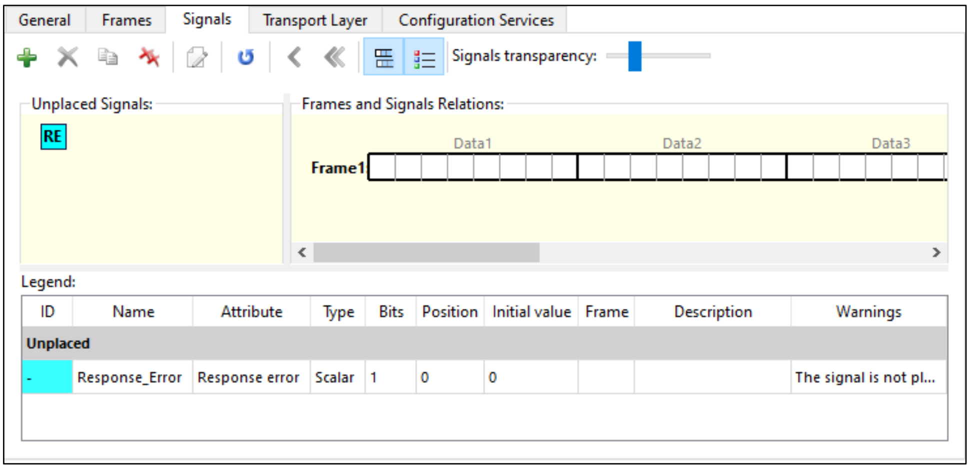
Note: *You cannot choose the Event-triggered type for a Subscribe frame. If you change this cell from Event-triggered to Unconditional, you must change the value of this frame to “None” in the Association column, if its name appears in any cells in that column.*

- **Association** – Used to relate unconditional frames with event-triggered frames. Allows the selection of the frame name of any unconditional frames that are not already associated with an event-triggered frame.

  An event-triggered frame must have at least one conditional frame associated with it. The valid values for this setting are the names of any existing unassociated unconditional frames. Only one unconditional frame can be associated with an event-triggered frame. As a result, when one of these cells has the name of an unconditional frame in it, this unconditional frame name cannot be available to any of the other rows.

6.3 Signals subtab

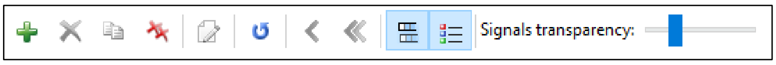
The **Signals** subtab is used to specify the parameters of signals added into the LIN frames. The subtab has the following sections: a toolbar to manage and navigate signals, the **Unplaced Signals** section to store newly added signals, the **Frames and Signals Relations** section to place configured signals, and the **Legend** pane to reflect.



ID	Name	Attribute	Type	Bits	Position	Initial value	Frame	Description	Warnings
Unplaced									
-	Response_Error	Response error	Scalar	1	0	0			The signal is not pl...

6.3.1 Signals toolbar

The **Signals** toolbar contains buttons to configure and navigate signals:



- **Add signal** – Puts a new signal on the Unplaced Signals section. The Signal Properties dialog displays after clicking this button.
- **Delete signal** – Removes the selected signal. A warning will display first.
- **Clone signal** – Duplicates the selected signal.

Note: *If the duplicated and original signals have the same name, a warning about the parameters and location of such signals will display.*

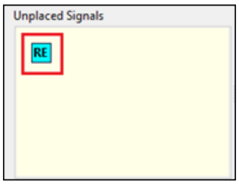
- **Delete all signals** – Removes all signals. A warning will display first.
- **Signal properties** – The Signal Properties dialog of the selected signal displays after clicking this button.
- **Renumber signals' IDs** – Reorders the signals index values.
- **Move signal to unplaced** – Relocates the selected signal to the Unplaced Signals section.
- **Move all signals to unplaced** – Relocates all signals to the Unplaced Signals section.
- **Show/hide event-triggered frames** – Shows or hides event-triggered frames.
- **Show/hide Legend** – Shows or hides the Legend pane.
- **Signals transparency** – This slider sets the transparency for signals graphics.

6.3.2 Unplaced Signals

The **Unplaced Signals** graphical pane serves as temporary to store added signals. Signals can be moved back and forth between the **Unplaced Signals** pane and the **Frames and Signals Relations** pane.

Note: If a frame is deleted on the ***Frames*** *subtab, any added to it signals (configured with the* ***Signals*** *subtab) are moved into the* ***Unplaced Signals*** *pane.*

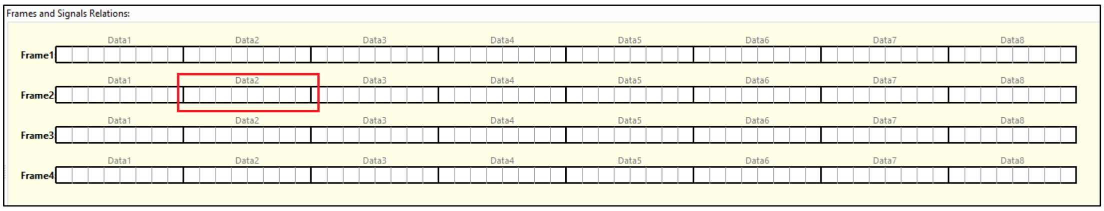
To display the "RE" symbol here, select the Use automatic response error signal option on the **General** subtab. Double-clicking this symbol opens the Signal Properties dialog.



6.3.3 Frames and Signal Relations

The Frames and Signal Relations graphical pane of the Signals subtab displays interactive graphics of the configured frames and signals.

Each frame graphic represents a frame added (configured) in the **Frames** subtab.

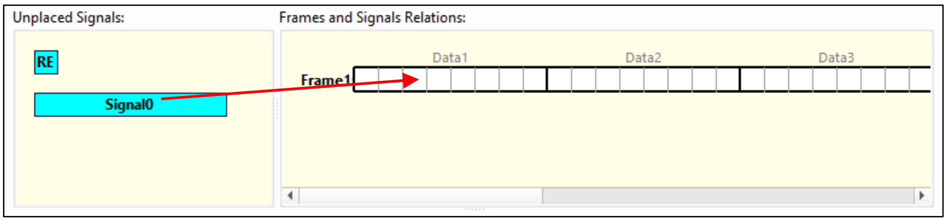


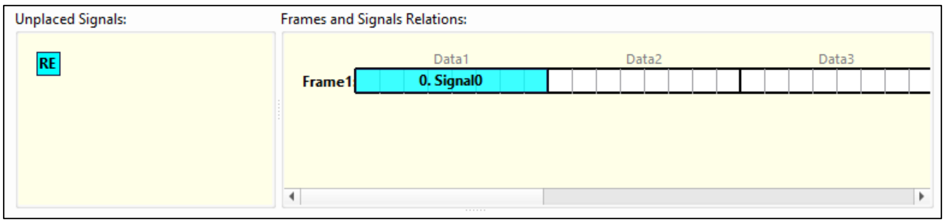
One frame length can be any from 1 to 8 data bytes. Each byte contains eight 1-bit cells: Data1 through Data8. The total number of frames cannot exceed 60. The total size of all frames is limited to 256 bytes.

Each signal graphic represents one signal defined for the LIN slave. The graphic for a signal appears as a solid bar. A signal is placed onto the frames using the drag-and-drop function. Placed signals occupy bits or bytes of the frames. Clicking a signal symbol selects that signal. Double-clicking a signal symbol opens the Signal Properties dialog.

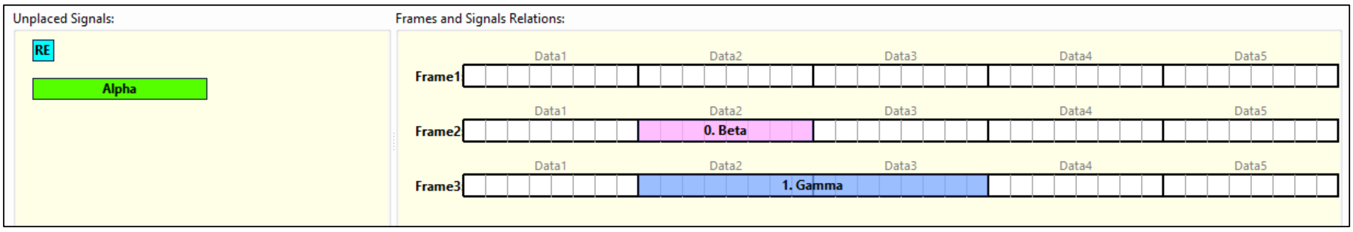
6.3.4 Graphical Interaction of Unplaced Signals and Frame and Signals panes

The interactive graphical panes of the **Signals** subtab allow selecting signals by clicking them, then, dragging and dropping them onto the frames:



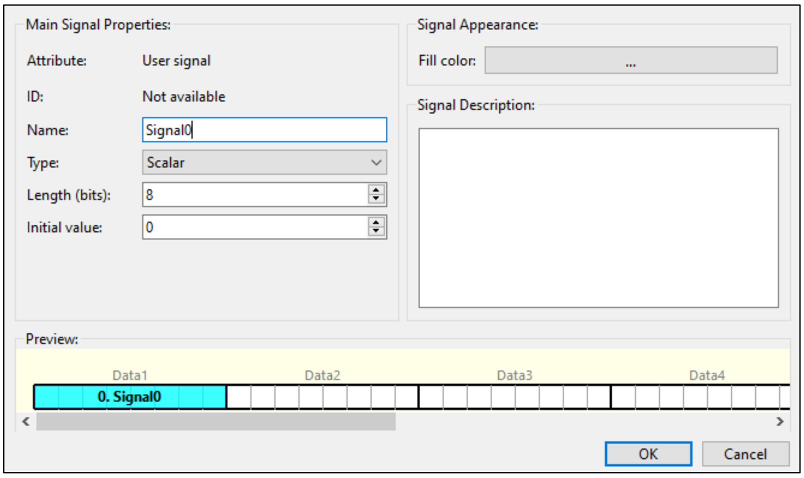


The graphical panes may look as follows depending on the selected signals’ parameters and graphic color:



6.3.5 Signal Properties

The Signal Properties dialog contains various parameters to configure signals.



Main Signal Properties:

Attribute: User signal

ID: Not available

Name: Signal0

Type: Scalar

Length (bits): 8

Initial value: 0

Signal Appearance:

Fill color: ...

Signal Description:

Preview:

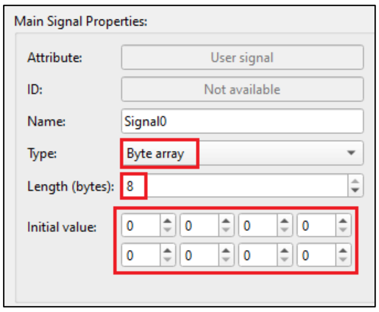
The Signal Properties dialog displays if you click the **Add** button on the toolbar to configure and add a new signal or double-click an already configured and added signal’s symbol to edit its configuration.

6.3.5.1 Main Signal Properties

- **Attribute** – The attribute value cannot be modified. The field displays the following values depending on the various signal configuration:
    - **User signal** – The default value.

  - **Response error** – Enable the Use automatic response_error_signal option on the General subtab. To display it on the Signal Properties dialog, click on the "RE" symbol and click the Signal properties button on the Signals subtab toolbar.
- **ID** – The ordering number of a signal placed on a frame. Cannot be directly modified.
- **Name** – The default signal name is Signal0, where "0" is equal to the signal index number. The signal name can be changed but it must contain only letters, digits, and underscores.
- **Type** – There are two types of signals:
  - **Scalar** – A signal of 1- to 16-bit length
  - **Byte array** – A signal of 1- to 8-byte length.

    If the Byte array type is selected, the Initial value parameter displays as many fields for entering values as there are bytes in the Length parameter:



Main Signal Properties:

Attribute: User signal

ID: Not available

Name: Signal0

Type: Byte array

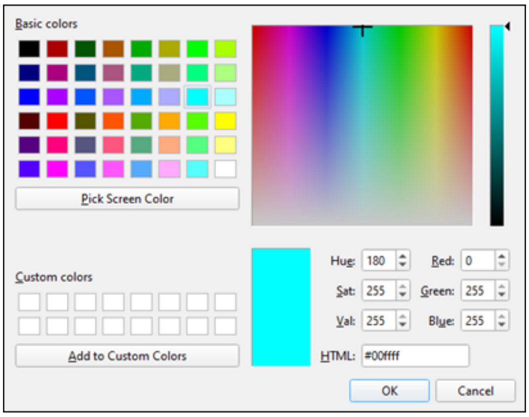
Length (bytes): 8

Initial value: 0 0 0 0 0 0 0 0

- **Length** (bits) – The length of a signal: 1-16 bits for Scalar signals; 1- 8 bytes for Byte array signals.
- **Initial value** – The initial value must be entered in the decimal format.

6.3.5.2 Signal Appearance

- **Fill color** – Clicking the color bar displays the dialog where to select the signal graphic color and its parameters.



Hue: 180 Red: 0

Sat: 255 Green: 255

Val: 255 Blue: 255

HTML: #00ffff

6.3.5.3 Signal Description

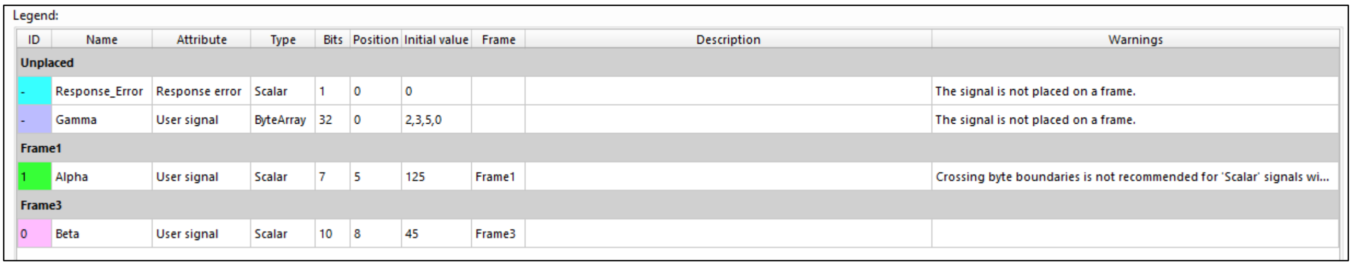
Any signal-related description or other information entered by the user.

Preview

The **Preview** graphical pane shows what the signal will look like when it is added.

6.3.5.4 Legend

The **Legend** pane describes the properties of configured signals.



Legend:

ID	Name	Attribute	Type	Bits	Position	Initial value	Frame	Description	Warnings
Unplaced									
-	Response_Error	Response error	Scalar	1	0	0			The signal is not placed on a frame.
-	Gamma	User signal	ByteArray	32	0	2,3,5,0			The signal is not placed on a frame.
Frame1									
1	Alpha	User signal	Scalar	7	5	125	Frame1		Crossing byte boundaries is not recommended for 'Scalar' signals wi...
Frame3									
0	Beta	User signal	Scalar	10	8	45	Frame3		

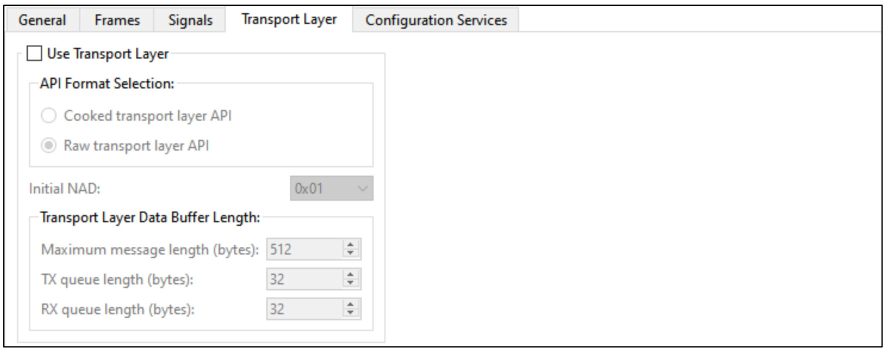
The descriptions of the Legend pane columns – ID, Name, Attribute, Type, Bits (Length (bits), Initial Value, Signal Description – are available on the Main Signal Properties pane of the Signal Properties dialog as they are properties of the same configured signals. Also, the Legend pane consists of the following columns:

- **Position** – Indicates the start bit where a signal is located on a frame. The range is 0-63.
- **Frame** – The name of the frame where a signal is located.
- **Warnings** – These are notices related to the configured signals.

6.4 Transport Layer subtab

The Transport Layer subtab contains parameters to configure the API and data buffer length.

Note: *The transport layer itself defines transportation of data contained in one or more frames. See the LIN 2.2 or ISO 17987 specification for detailed information on the Transport Layer.*



□ Use Transport Layer

API Format Selection:

○ Cooked transport layer API

● Raw transport layer API

Initial NAD: 0x01

Transport Layer Data Buffer Length:

Maximum message length (bytes): 512

TX queue length (bytes): 32

RX queue length (bytes): 32

6.4.1 Use Transport Layer

- If you select this option, a warning displays to ensure that the values entered on the Configuration Services subtab are correct.
- If you deselect this option, a warning displays that the Automatic Configuration Request Handling option will also be disabled.

6.4.2 API Format selection

This section is used to select the format for the Transport Layer API functions:

- **Cooked transport layer API** – The cooked format is used to send and receive Transport Layer messages. It is the user's responsibility to ensure that just one API function is used for each message.
- **RAW transport layer API** – The raw format is used to send or receive each frame that makes up a Transport Layer message using one API function call for each frame.

6.4.3 Initial NAD

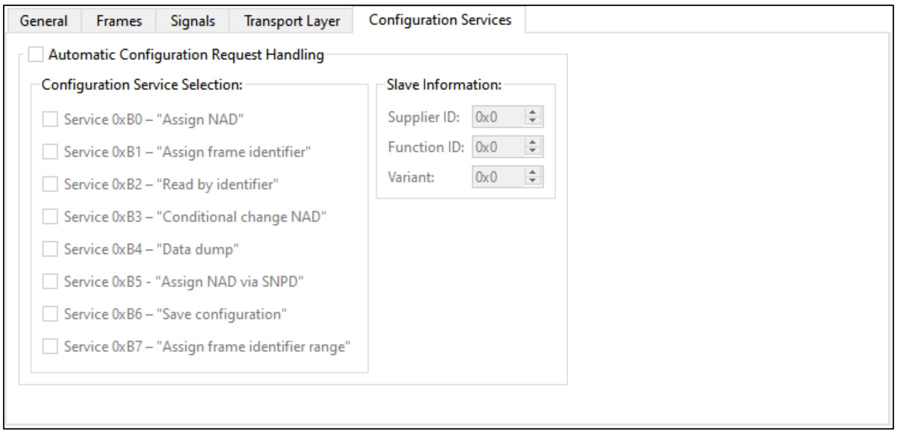
This field is used to select the Network Address (NAD) of the slave node. Each slave node has an initial NAD list. The NAD is used in MRF and SRF frames to address one particular slave node in a cluster. Note that this field is used to select the initial NAD for the node. The NAD of a slave node can change at run time.

6.4.4 Transport Layer Data Buffer Length:

- **Maximum message length (bytes)** – This property is used to select the maximum Transport Layer message length that this slave node supports. The minimum value is 6. This configurator only supports Transport Layer messages with lengths up to 4095 bytes.
- **TX queue length / RX queue length (bytes)** – These properties are only applicable when the Raw Transport Layer API format is selected. The configurator supports queue lengths from 8 to 2048 with 8-byte steps. The default size of each queue is 32 bytes.

6.5 Configuration Services subtab

The **Configuration Services** subtab contains a check box list of eight services and the slave parameters.



□ Automatic Configuration Request Handling

Configuration Service Selection:

□ Service 0xB0 – "Assign NAD"
□ Service 0xB1 – "Assign frame identifier"
□ Service 0xB2 – "Read by identifier"
□ Service 0xB3 – "Conditional change NAD"
□ Service 0xB4 – "Data dump"
□ Service 0xB5 - "Assign NAD via SNPD"
□ Service 0xB6 – "Save configuration"
□ Service 0xB7 – "Assign frame identifier range"

Slave Information:

Supplier ID: 0x0
Function ID: 0x0
Variant: 0x0

6.5.1 Automatic Configuration Request Handling

All the other options on the subtab are available only if the Automatic Configuration Request Handling option is selected. Deselect this option to handle these requests with your own custom application code. Whenever any automatically-handled request occurs during the LIN bus operation, the corresponding MRF and SRF frames will not be available to the application through the Transport Layer API. If this option is not selected on this subtab, then the corresponding MRF and SRF frames of the configuration service request are received or sent by the application using the Transport Layer API.

6.5.1.1 Configuration Service selection

The LIN Configurator supports eight configuration service requests (0xB0 to 0xB7) to select for automatic handling:

- **Service 0xB0** – "Assign NAD" – An optional service per LIN 2.2 and ISO17987 specifications. This is a service request where a new NAD value is assigned to the slave node.
- **Service 0xB1** – Assign frame identifier" – An obsolete service per LIN 2.2 and ISO 17987 specifications. It is available only if the LIN 2.0 Compatibility checkbox is selected on the **General** subtab.
- **Service 0xB2** – "Read by identifier" – This service is mandatory per LIN 2.2 and ISO 17987 specifications. This request is used to allow the LIN master to read the slave's identification information (Supplier ID,

Function ID, and Variant). This middleware only supports the LIN Product Identification version of this request.

- **Service 0xB3** – "Conditional change NAD" – An optional service per LIN 2.2 specifications. Also, this is an optional obsolete service in the ISO 17987-3 specification.
- **Service 0xB4** – "Data dump" – This service is optional per LIN 2.2 and ISO 17987 specifications and is not supported by this middleware.
- **Service 0xB5** – "Assign NAD via SNPD" – This service is not supported per LIN 2.2, and ISO 17987 specifications. However, when the Enable J2602-1 Compliance checkbox is selected on the **General** subtab, this service's meaning changes to "Targeted Reset", which is supported by this middleware.
- **Service 0xB6** – "Save configuration" – An optional service request per LIN 2.2 and 17987 specifications.
- **Service 0xB7** – "Assign frame identifier range" – A mandatory configuration service per LIN 2.2 and ISO 17987-3 specifications. This service allows the LIN master to change the volatile frame PID values for the slave's frames.

6.5.1.2 Slave information

If the user selects the Automatic Configuration Request Handling option, three fields become available to enter values used in the configuration service requests. These fields are as follows:

- **Supplier ID** – This is a 16-bit value, but its valid range is from 0x0000 to 0x7FFE.

Note: The supplier ID default value is 0x00C3.

- **Function ID** – This is 16-bit value, and its valid range is 0x0000 to 0xFFFE.
- **Variant** – This is 8 bit-value and its valid range is from 0x00 to 0xFF.

7 Version changes

This section lists and describes the changes for each version of this tool.

Version	Change descriptions
1.0	New tool.
1.10	Added Multi-Instance support.
	Added the Import LDF/NCF file and Export NCF file options.
1.20	Changed the device library file from xml to *props.json*.
1.21	Fixed GUI issues, which were truncated label text on the Transport Layer and Configuration Services tabs.

Revision history

Revision	Date	Description
**	2021-04-23	1.0 New tool.
*A	2021-09-22	Updated to version 1.10.
*B	2022-09-28	Updated to version 1.20.
*C	2023-05-16	Updated to version 1.21.

Trademarks
All referenced product or service names and trademarks are the property of their respective owners.

Edition 2023-05-16

Published by

Infineon Technologies AG
81726 Munich, Germany

Do you have a question about this document?

Email: erratum@infineon.com

Document reference

002-32984 Rev. *C

Important notice
The information given in this document shall in no event be regarded as a guarantee of conditions or characteristics ("Beschaffenheitsgarantie")

With respect to any examples, hints or any typical values stated herein and/or any information regarding the application of the product, Infineon Technologies hereby disclaims any and all warranties and liabilities of any kind, including without limitation warranties of non-infringement of intellectual property rights of any third party.

In addition, any information given in this document is subject to customer's compliance with its obligations stated in this document and any applicable legal requirements, norms and standards concerning customer's products and any use of the product of Infineon Technologies in customer's applications.

The data contained in this document is exclusively intended for technically trained staff. It is the responsibility of customer's technical departments to evaluate the suitability of the product for the intended application and the completeness of the product information given in this document with respect to such application.

Warnings
Due to technical requirements products may contain dangerous substances. For information on the types in question please contact your nearest Infineon Technologies office.

Except as otherwise explicitly approved by Infineon Technologies in a written document signed by authorized representatives of Infineon Technologies, Infineon Technologies' products may not be used in any applications where a failure of the product or any consequences of the use thereof can reasonably be expected to result in personal injury.