

Go Configure™ Software Hub

User Guide

v.2.0.1

Contents

1	Introduction	3
1.1	Quick Start	4
1.1.1	System Requirements	4
1.1.2	Support	4
2	Software overview	5
2.1	Design tools	6
2.1.1	Interface overview	7
2.1.2	Working with project files	12
2.1.3	NVM viewer	17
2.1.4	Components	18
2.1.5	Connections	21
2.1.6	Keyboard commands	23
2.1.7	Settings	25
2.1.8	Legend box	29
2.1.9	Help window	29
2.1.10	Snipping tool	31
2.1.11	Rules checker	31
2.1.12	Comparison tool	32
2.1.13	Power Dissipation Calculator	36
2.2	Debug tools	41
2.2.1	Hardware sources	42
2.2.2	Generators	47
2.2.3	Signal Wizard	59
2.2.4	Custom Signal Wizard	62
2.2.5	Hardware configurations	63
2.2.6	Debugging Controls	64
2.2.7	Demo Board and Demo Mode	71
2.2.8	I2C Tools	72
2.2.9	OTP Programmer	80
2.2.10	Project data window	83
2.2.11	I2C I/O Tool	85
2.2.12	Logic Analyzer	87

2.2.13	UART Terminal	95
2.3	Configuration tools	97
2.3.1	Asynchronous State Machine Editor	97
2.3.2	EPG Waveform Editor	101
2.3.3	Timing Diagram	105
2.4	Software simulation tool	108
2.4.1	Schematic Library	108
2.4.2	External components	110
2.4.3	Working with models and subcircuits	111
2.4.4	Source Setup	114
2.4.5	Simulation configurations	117
2.4.6	Probes	117
2.4.7	Debugging Controls	119
2.4.8	Simulation progress	121
2.4.9	Simulation Results window	123
3	Devices	127
3.1	GreenPAK devices	128
3.1.1	GreenPAK Advanced Development Platform	128
3.1.2	GreenPAK DIP Development Platform	130
3.1.3	GreenPAK Lite Development Platform	132
3.1.4	GreenPAK Serial Debugger	134
3.1.5	Connecting external GreenPAK	136
3.2	ForgeFPGA devices	138
3.2.1	ForgeFPGA Advanced Development Platform	138
3.2.2	ForgeFPGA Evaluation board	141
3.3	PowerPAK Devices	143
3.3.1	PowerPAK Development Platform	143
3.3.2	PowerPAK SLG51002C Demo board	145
3.3.3	Connecting external PowerPAK	147
4	How to	149
4.1	Software update	150
4.2	Printing	151
4.2.1	Print	151
4.2.2	Print Editor (obsolete)	152
5	Appendices	154
5.1	Main menu commands	155
5.2	Keyboard and mouse controls	157
5.3	Debugging Controls feature availability	160
5.4	Abbreviations	161
5.5	Changelog	162

1 Introduction

Renesas Go Configure™ Software Hub is a full-featured Integrated Development Environment (IDE) that enables a completely graphical design process, allowing you to configure, program and simulate custom circuits in minutes.

This guide provides the information for the usage of Go Configure Software Hub, describing the features for Renesas products, such as GreenPAK™, AnalogPAK, HVPAK™, Power GreenPAK, AutomotivePAK, and ForgeFPGA™ Workshop.

1.1 Quick Start

1.1.1 System Requirements

Minimum system requirements for *Go Configure Software Hub*:

- CPU: 2.5 GHz
- RAM: 4 GB
- GPU: 512 MB
- Free disk space: 2 GB
- OS: Windows 7/8.1/10 (x64), macOS (v10.15+), Ubuntu 18.04 (x64), Debian 11 (x64)

1.1.2 Support

In case you need more information about *Go Configure Software Hub*, online support is available at www.renesas.com.

You can also find *Go Configure Software Hub* on Renesas social media. To visit the page, go to *Help* → *Social* in the main menu.

The latest version of the software application is available on the [Renesas website](#), on [Software](#) page.

Go Configure Software Hub notifies about pending software updates. You can learn more in section [4.1 Software update](#).

2 Software overview

Go Configure Software Hub is a product, created to design a specific device configuration. The software gives a direct access to all GreenPAK, SLG51000/1, and ForgeFPGA device features and complete control over the routing and configuration options.

The software contains the tools that allow you to:

- Program a chip with the created design
- Read a programmed part and import its data to the software
- Simulate with external components

Getting Started:

1. [Download](#) and install *Go Configure Software Hub*
2. Open the particular chip Part Number
3. Select the components for your project
4. Specify the pinout
5. Configure and interconnect the components
6. Test the design with the *Debug Tool*, using *Simulation* feature or any of the supported hardware development platforms

2.1 Design tools

Start your project from the *Hub* window with the following sections:

- *Welcome* — useful info and tips for new users
- *Recent files* — the list of the recently opened [project files](#)
- *Develop* — the chip Part Number selection. See the *Details* section to learn more about the selected chip

At the bottom of the window, you can find the *New*, *Open*, and *Close* buttons, which allow you to start a new project for a selected Part Number, open an existing project or close the *Go Configure Software Hub*. The *Datasheets* and *User Guides* buttons redirect you to the *Renesas* website, where you can download the corresponding files.

- *Demo* — the list of *Demo* projects. You can use the specific *Demo Board* for the project debugging (read more in section [2.2.7 Demo Board and Demo Mode](#))
- *Application notes* — design examples for different purposes. An application note includes a design description with various Integrated Circuits (ICs) and a preconfigured circuit project, where you can make customized changes
- *Recovery files* — restored project files after a crash or freeze. To read more refer to section [2.1.7 Settings](#)

The screenshot shows the Go Configure Software Hub interface. On the left is a vertical sidebar with buttons for Welcome, Recent files, Develop, Demo, Application notes, Recovery files, Datasheets, and User Guides. The Welcome button is currently selected. The main area has tabs for Software Tool (set to All) and Part Family (set to All). A large image of the SLG51001C chip is displayed. Below it is a table with columns: Part Number, DS, VDD (V), VDD2 (V), GPIO|GPO|GPIO, AEC-Q100, and Notes. The table lists several parts under GreenPAK Designer, including SLG51003C, SLG47003V, SLG47011V, SLG51000C, SLG51001C, SLG47910V, SLG51002C, SLG47004V, SLG47115V, and SLG51001CTR. The SLG51001C row is highlighted. At the bottom of the main area, there are buttons for New, Open, and Close, and links for Datasheet, Product page, Application notes, Resources, Get samples, and Contact us. The Details section at the bottom provides information about the package (WLCSP-16), supported development platforms (GreenPAK Serial Debugger and PowerPAK Development Board + SLG51001CTR Evaluation Socket), and a description of the SLG51001's performance characteristics.

Part Number	DS	VDD (V)	VDD2 (V)	GPIO GPO GPIO	AEC-Q100	Notes
SLG51003C	Contact us	2.80 to 5.00	1.20 to 5.00	1 0 4	-	3x LDO (1x HP 475 mA)
SLG47003V	Contact us	2.30 to 5.50	-	5 0 10	-	
SLG47011V	Contact us	1.71 to 3.60	-	1 0 13	-	ADC (14-bit; SAR), D
SLG51000C	PDF	2.80 to 5.00	-	1 0 5	-	7x LDO; 2x Lo
SLG51001C	PDF	2.80 to 5.00	-	1 0 3	-	6x LDO; 1x Lo
SLG47910V	Contact us	0.99 to 1.21	1.71 to 3.60	0 0 19	-	
SLG51002C	PDF	2.80 to 5.00	1.20 to 5.00	1 0 5	-	8x LDO; 5x Load Switch
SLG47004V	PDF	2.40 to 5.50	-	1 0 7	-	2x Op Amp or 1x I
SLG47115V	PDF	2.30 to 5.50	4.50 to 26.40	1 2 7	-	1x H
SLG51001CTR	PDF	2.80 to 5.50	2.20 to 12.20	1 1 7	-	

Go Configure Software Hub User Interface

2.1.1 Interface overview

The basic interface elements are *main menu*, *toolbar*, *work area*, *work area controls*, *properties panel*, and *components list*.

Main menu

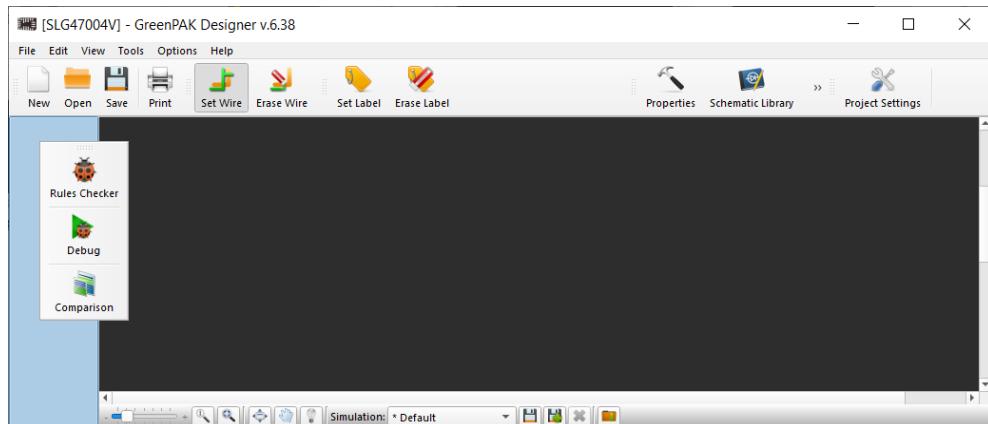
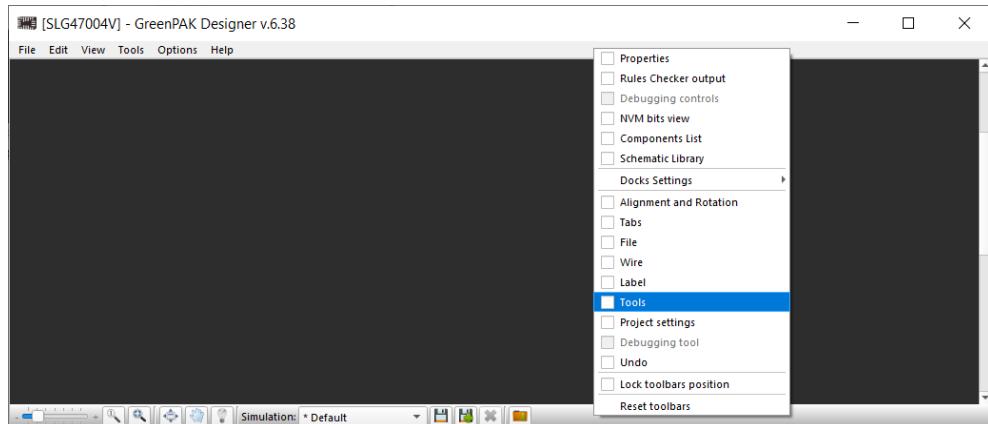
After you select the Part Number and start the project, you can see the menu bar with the following items: *File*, *Edit*, *View*, *Tools*, *Options* and *Help*. See the full description of all the menu commands in section 5.1 Main menu commands.



Main menu

Toolbar

The toolbar items are located under the menu bar by default. Move, show, and hide toolbar items to customize the environment. Use *Settings* to reset the *Appearance*.

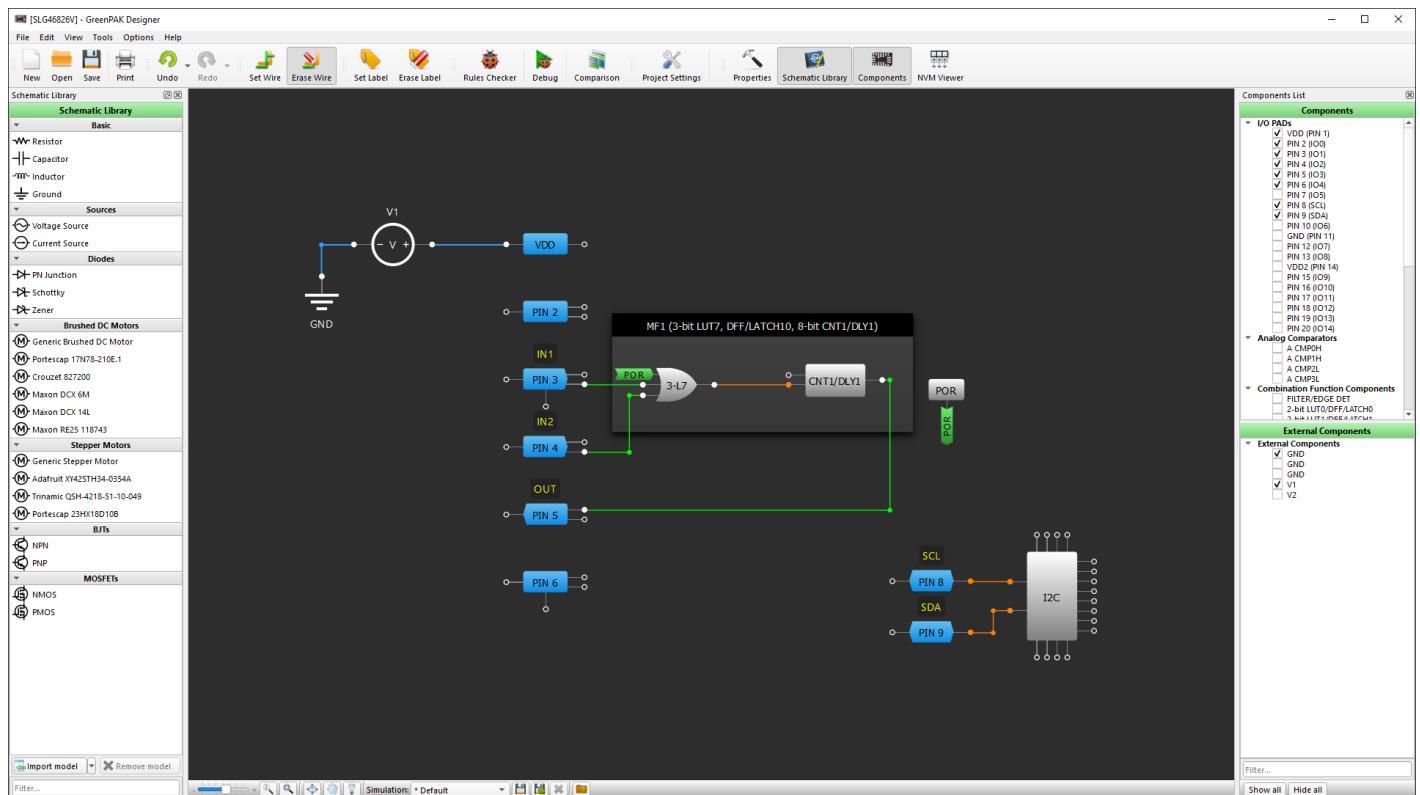


Toolbar customization

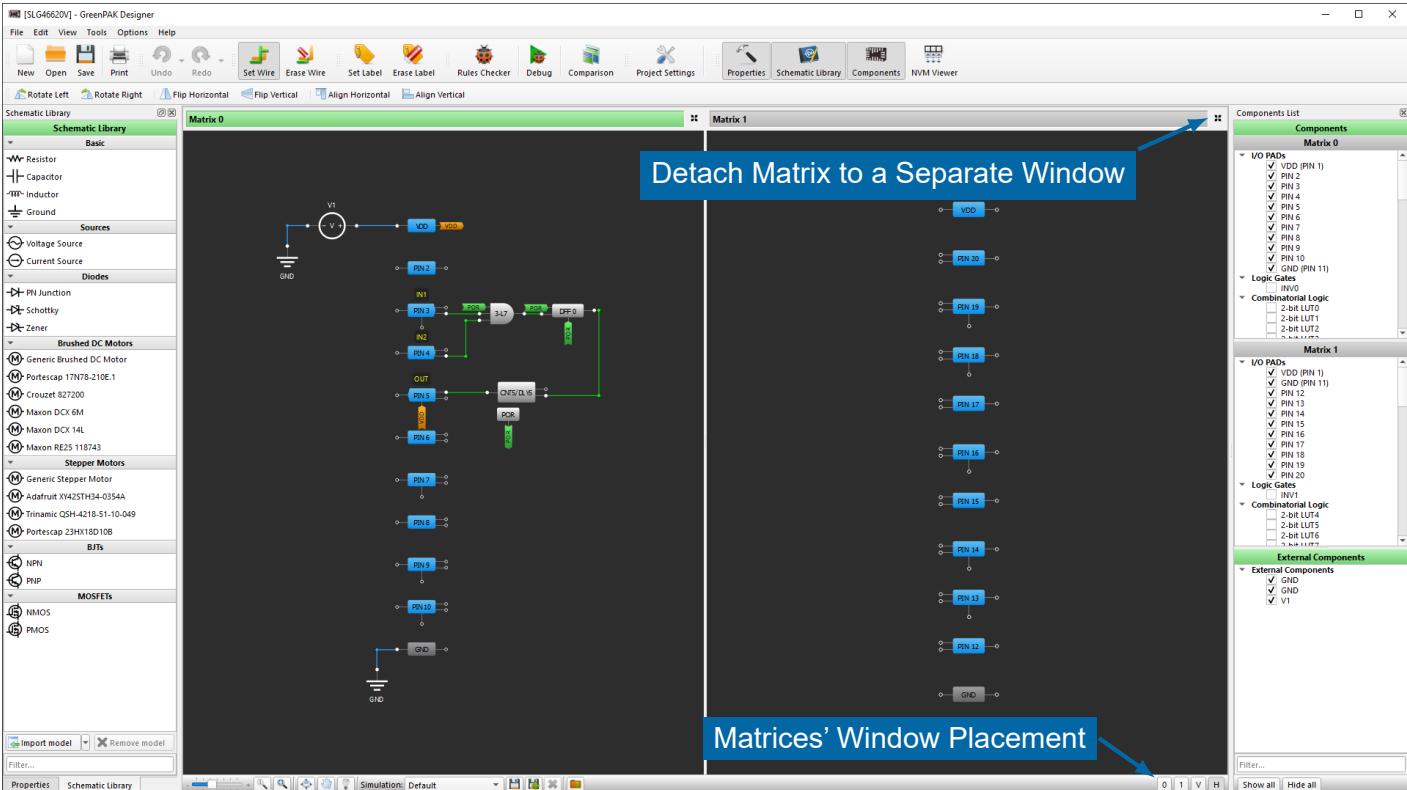
Work area

The work area displays the selected components (macrocells) from the components list and the connections between them. The Part Numbers contain different sets of macrocells, therefore the IDE's interface varies.

The IDE may contain one or two work areas (an example of Part Number with two work areas is SLG46620V).



Part Number with one work area



Part Number with Two Work Areas

The Part Numbers with two work areas have several distinctive buttons, which define the matrix window placement:



- 0 — only *Matrix 0* is displayed
- 1 — only *Matrix 1* is displayed
- V — both matrices are displayed vertically (on top of each other)
- H — both matrices are displayed horizontally (beside each other)
- ┏ ┐ — makes one of the matrices a separate window

The *Matrix* title bar turns green when the work area is in focus.

Work area controls

You can find the additional work area controls on the bottom toolbar.



Work area controls



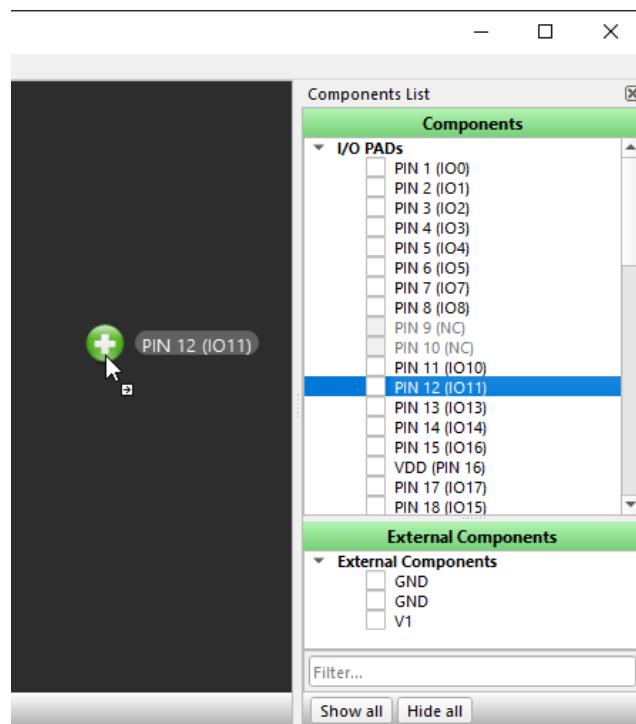
- Adjust the work area (zoom, fit the work area or make it full-screen)
- Enable the *Pan mode* to move the work area (click and hold the middle mouse button as an alternative)
- Activate the possibility to see the hint box with the block's property info upon hovering the mouse over a component

Components list

This panel contains a list of available macrocells in a chip. You can show, hide, or search for the components via the list.

Show/hide a component by using the checkbox next to its name.

Drag and drop a block from the list to the work area.



Checkbox / Drag and Drop

You can use filter to easily search for the required component. Show/hide all the components by using *Show all* and *Hide all* buttons.

Note 1: A hidden component retains the configurations set on its properties panel (see more info about the [Properties](#) panel).

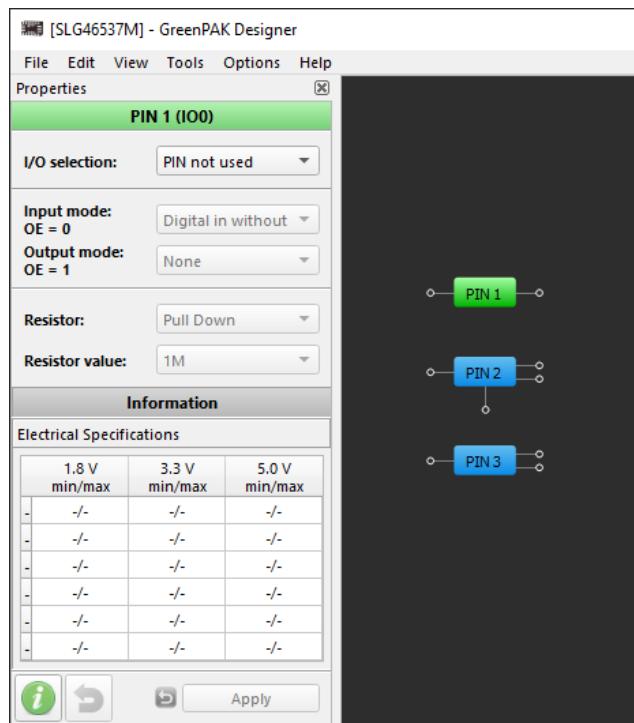
Note 2: It is not possible to hide connected blocks (see sections [2.1.4 Components](#) and [2.1.5 Connections](#) to find out more).

Properties panel

The panel contains all settings available for a selected component. The set of properties may vary depending on the macrocell.

The *Properties* panel may contain:

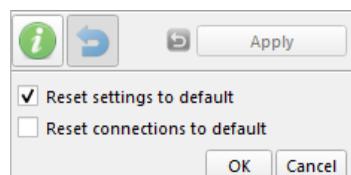
- Settings and parameters that can be specified for a selected block
- Settings that control the predefined connections to a selected block



Properties panel

Once you change the property values, you can *Apply* or *Revert* the modifications.

After changes are applied, it is possible to *Reset* the changes. The following options may be available: *Reset connections to default*, *Reset settings to default* or *Reset configurations to default* (the latter is available only for the components with more than one mode, e.g. LUT).

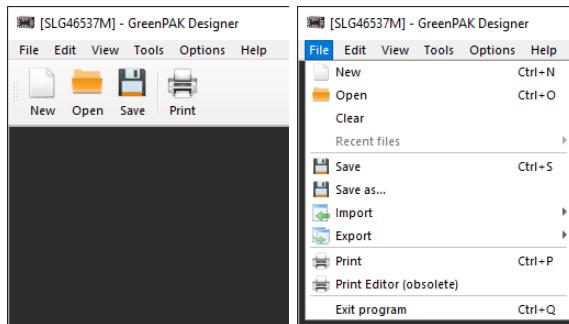


Property reset feature

You can also *Reset* the block via the component's context menu.

2.1.2 Working with project files

A project file is the latest saved state of the designed project. The file contains component configurations, connections between blocks, component location on the work area, etc.



File operations

Create/Save/Open a project

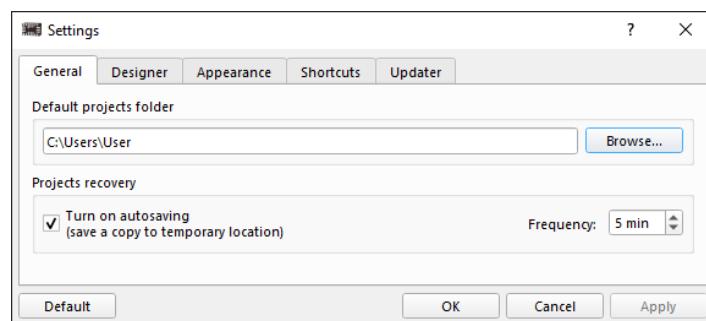
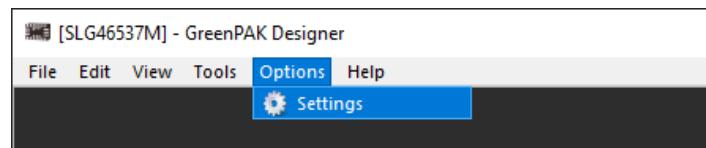
You can start a new project from the *Hub* window (click the Part Number → *New*) or from an opened instance (click the *New* icon on the toolbar, or *File* → *New*)

Once you start a new project, the *Project settings* window appears. You can skip this step, but once the data is needed, you'll be asked to fill it in again. Read more about *Project settings* window later in this section.

The default project (state of the work area right after you open a new project) is usually configured for minimal power consumption. That is why some components are disabled.

To save your project click the Save icon on the toolbar, or click *File* → *Save/Save as*.

Change the folder to which the project is saved in *Settings*.



Project file folder

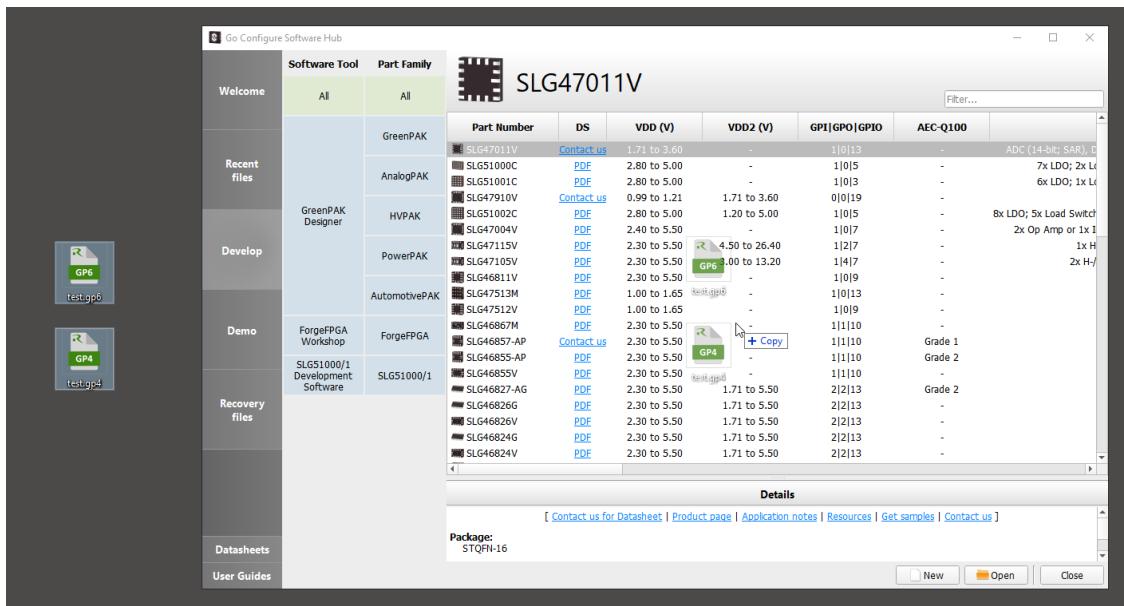
The supported project file extensions are: .aap, .can, .gp3, .gp4, .gp5, .gp6, .hvp, and .ppak (.gp1, .gp2,

.gpp and .ldo are the obsolete file extensions).

There are several ways to open a project file:

- Open button on the *Hub* window
- Open icon on the toolbar (or *File* → *Open*)
- Drag and drop the project file to the *Hub* window
- Drag and drop the project file to the work area
- Double-click the project file

It is also possible to open multiple project files at once using drag and drop feature.



Open a project file

Note: It is highly recommended not to make manual changes in the file to prevent from its corruption.

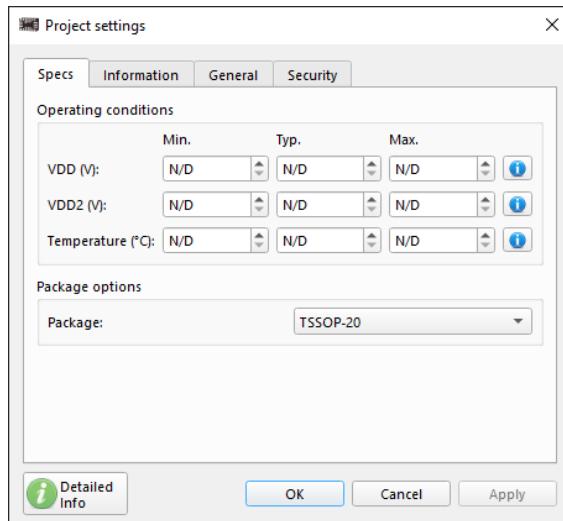
Project settings window

As mentioned earlier in this section, the *Project settings* window appears right after you start a new project. It contains four tabs: *Specs*, *Information*, *General*, and *Security*.

In *Specs*, you can see the fields to fill in the chip operating conditions, such as VDD and Temperature (Min., Typ. and Max). The Part Numbers may have one or multiple VDDs. Click the blue *information* button to see the recommended range within which the chip can operate safely.

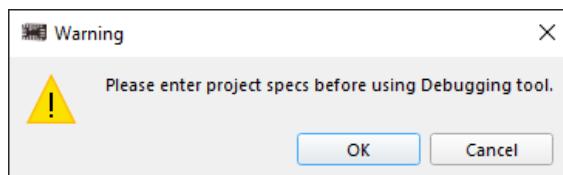
The Part Numbers may support different packaging. The package can also be selected in the *Specs*

tab.



Project settings window examples

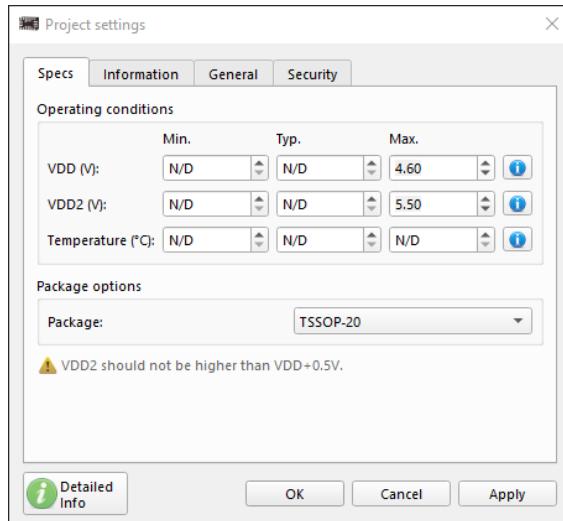
Note: You can close the *Project settings* window without entering the operating conditions. Working with *Simulation* or hardware development platforms in the *Debug tool* requires entering the specs. Once you click *Debug* on the toolbar while specs are not specified, a warning notification to add specs appears. Click the *OK* button on the warning window or the *Project Settings* icon on the toolbar to enter specs.



Warning to add specs to use Debug tool

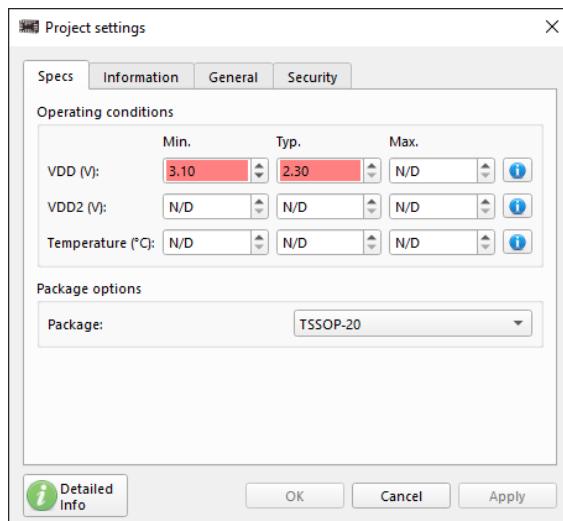
In case the improper operating conditions are filled in, a warning with the corresponding text message will appear. Some warning notifications do not stop you from applying the specs, while

others require to fix them before the application. Here are the warning examples:



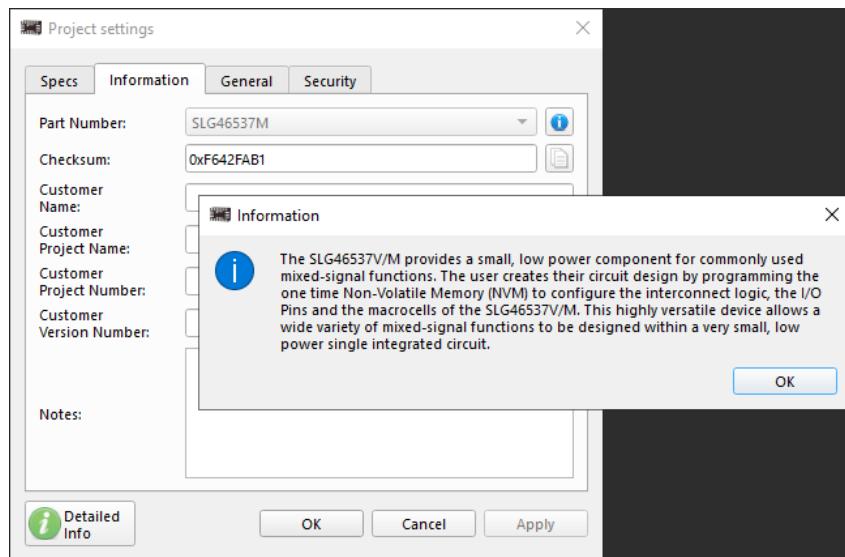
Specs-related warning examples

If e.g. Min. value is higher than Typ., and in other similar cases, the affected fields become highlighted in red. The specs cannot be applied until the issue is fixed.



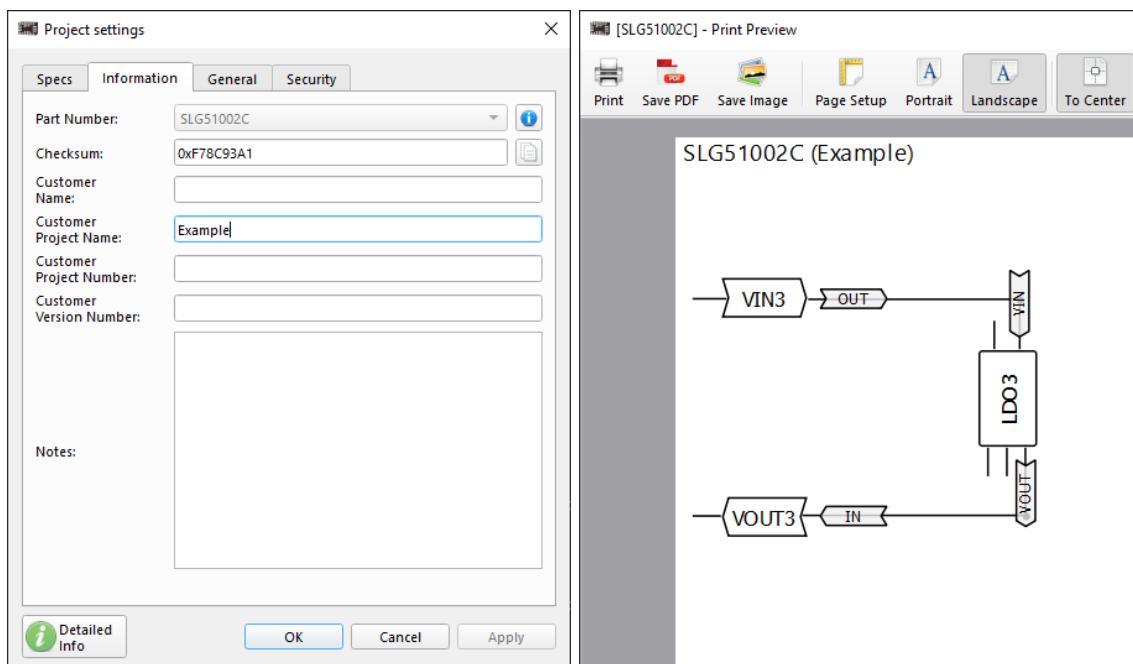
Improper specs values

The *Information* tab contains the short piece of information about the Part Number. To see the info, click the blue *info* button. You can also see the input fields to fill in the information about the project.



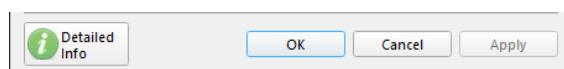
Information tab

The entry in *Customer project name* field is also displayed in *Print Preview*.



Customer project name in Print Preview

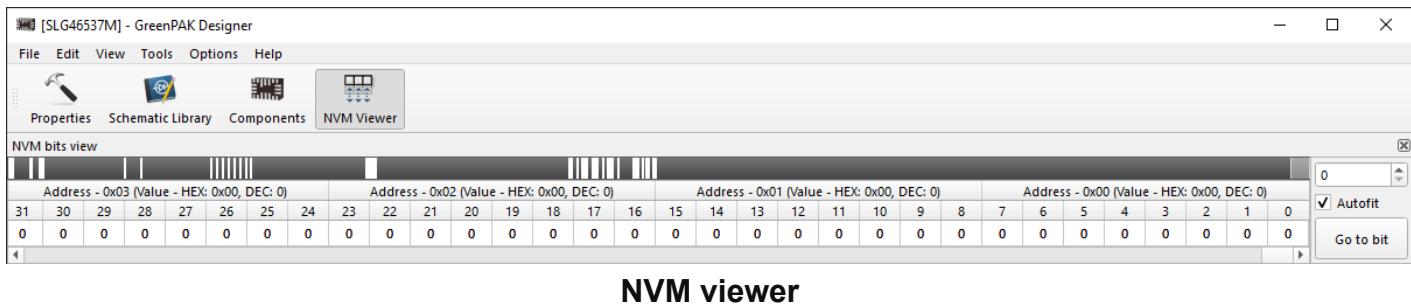
The *General* and *Security* tabs contain the global chip configurations. The set of settings varies depending on the Part Number. You can find the information about the particular chip configurations by clicking the *Detailed info* button on the *Project settings* window or referring to the Datasheet (you can reach the Datasheets from the *Hub* window).



Detailed info button

2.1.3 NVM viewer

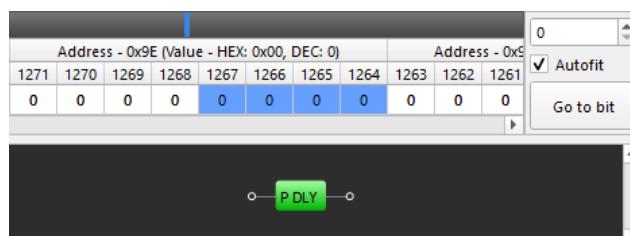
The non-volatile memory (NVM) tool represents the permanent memory of a chip. The NVM is grouped into bytes. Each byte is an 8-bit sequence. Above each byte, you can see the information about the byte's address along with its value in hexadecimal and decimal formats. NVM changes occur after the macrocell's property modifications or setting a connection between the [blocks](#).



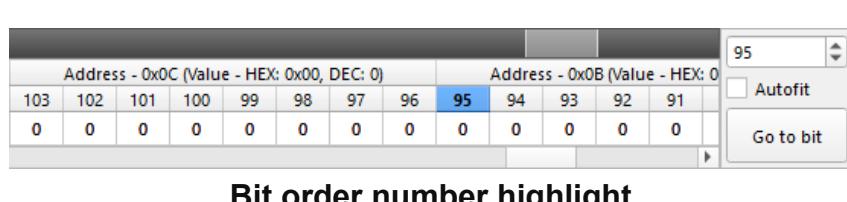
NVM viewer cells can be in the following states:

- 0 bit range of a selected block
- 0 bits with 0 value
- 1 bits with 1 value
- 0 latest bit changed to 0 value
- 1 latest bit changed to 1 value

You can also use the NVM viewer controls for quick navigation through the bit table. Use the *Autofit* feature to jump directly to the bit range of the selected component.



Click *Go to bit* once you enter a bit number (in the decimal format) to find the bit you need. The bit order number becomes highlighted after clicking a bit or using the *Go to bit* button.



The NVM viewer navigation bar can also orient you to the location of selected or changed bits in the table (the color on the bar corresponds the cells color described above). In addition, black color represents 0 and white shows 1 values.

Address - 0xA9 (Value - HEX: 0x04, DEC: 4)																Address - 0xA8 (Value - HEX: 0x07, DEC: 7)							Address - 0xA7 (Value - HEX: 0x00, DEC: 0)						
1359	1358	1357	1356	1355	1354	1353	1352	1351	1350	1349	1348	1347	1346	1345	1344	1343	1342	1341	1340	1339	1338	1337	1336						
0	0	0	0	0	1	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0							

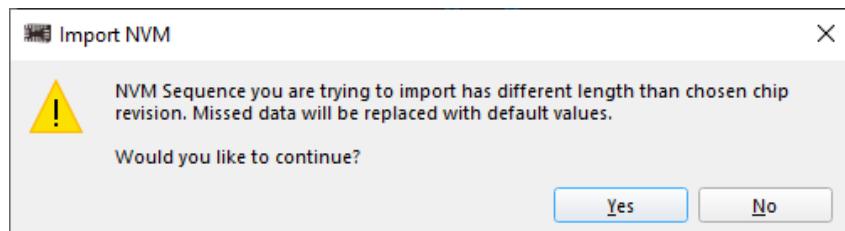
NVM viewer top bar

Hover the mouse cursor over the bit to see which macrocell this bit belongs to.

Address - 0x92 (Value - HEX: 0x00, DEC: 0)								Address - 0x91 (Value - HEX: 0x18, DEC: 24)							
1175	1174	1173	1172	1171	1170	1169	1168	1167	1166	1165	1164	1163	1162	1161	1160
0	0	0	0	0	0	0	0	A CMP1	0	0	1	1	0	0	0

NVM hint

You can save or load the NVM sequence by clicking *File* → *Import/Export* in the main menu (see the description of all main menu items in section [5.1 Main menu commands](#)). If the imported file contains a different bit sequence length than the selected Part Number, the corresponding pop-up appears and it is possible to proceed with loading the data.

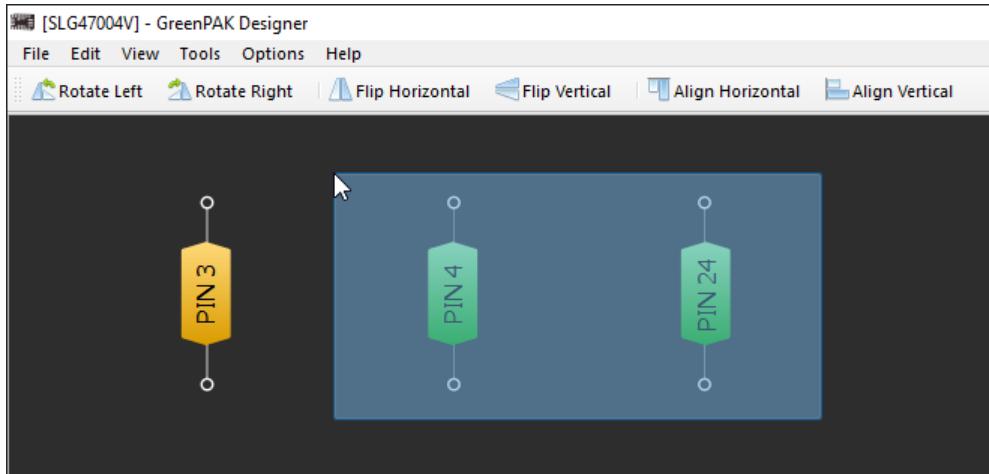


Import different NVM sequence length

2.1.4 Components

After you add a macrocell from the [components list](#) to the work area, you can choose how to interact with it. Move the component(s) use a mouse or a keyboard (see section [5.2 Keyboard and mouse controls](#)). You can also use the toolbar widgets to rotate, flip, and align the selected block(s). To select more than one block click, hold, and drag over the components you want to

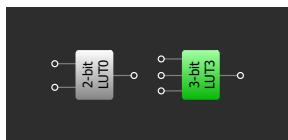
select.



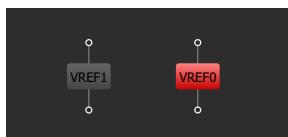
Selected block adjustment

GreenPAK chips components highlight

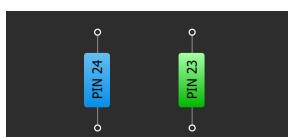
The macrocell color gives you information about its mode (enabled/disabled) and state (deselected/selected). Input/Output (I/O) PIN block's color informs about its relation to a particular VDD.



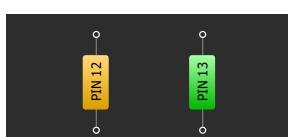
enabled, deselected/selected



disabled, deselected/selected



PIN, deselected/selected

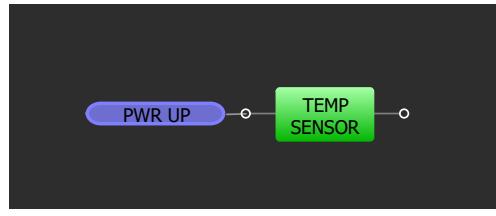


PIN for connection with VDD2 (analog), deselected/selected

You can connect macrocells via the pins.

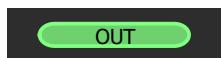
Hover the cursor over the block to see the pin hints. For more info about managing pin hints

behavior, refer to section 2.1.7 Settings.



Pin hint

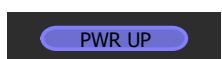
Macrocell pin hint color indicates the connection possibility between the blocks: whether a wire can or cannot be added, or which connection type can be set.



open for connection



connection limit is reached



temporarily closed for connection (you can change macrocells properties to make it available for connection)



used for hard-wired connections



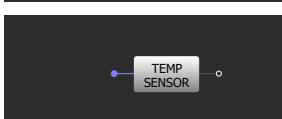
external IN/OUT

Read about connection types in section 2.1.5 Connections.

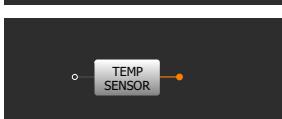
When you are in the process of setting a connection, you can see the pin highlight. Just like pin hint color, the pin color indicates the connection possibility.



open for connection



temporarily closed for connection (you can change macrocell settings on its *Properties* panel to make it available for connection)



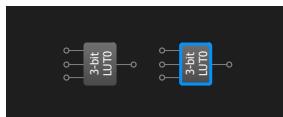
used for hard-wired connection



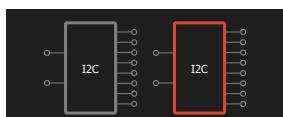
connection is not allowed

SLG51000/1 chip components highlight

Component highlight for SLG51000/1 chips in different modes or states is as follows:



enabled, deselected/selected



disabled,deselected/selected



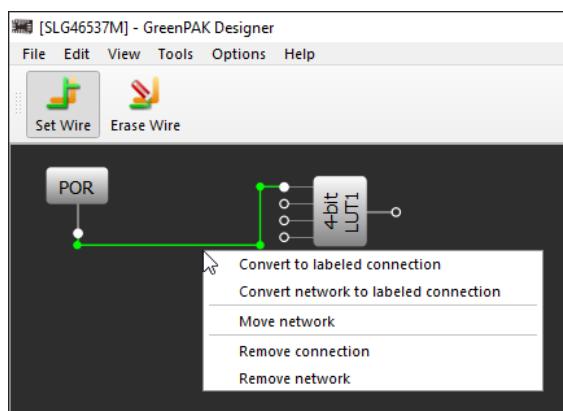
I/O PIN, deselected/selected

Pin and pin tip colors are the same as for chips described above.

2.1.5 Connections

The *Set Wire* widget helps you to set a connection between the blocks. To add a wire, click two pins between which you are setting a connection. More than one connection set from one OUT pin is a network. You can dismiss connection creation by clicking the right mouse button.

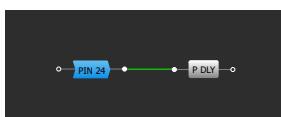
To remove the connection, enable the *Erase Wire* tool and click the wire (you can also delete the connection or network via the context menu, triggered by right-clicking the wire).



Set/Erase wire/network

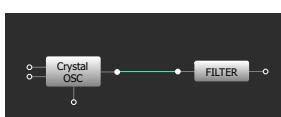
GreenPAK chips connections highlight

While working with different Part Numbers you may encounter the following component connection types:



Matrix

the most common connection which can be set between green pins



Shared

matrix connection which is physically shared with more than one component

	<i>Hard-wired</i>	connections which are predefined and depend on block settings
	<i>Bus</i>	same as hard-wired
	<i>State dependent</i>	connections set to state dependent components (e.g. <i>Dynamic Memory</i> , <i>F</i> or <i>State</i> blocks within ASM subsystem in SLG46880V chip)
	<i>External</i>	connection set from/to external components (read more in section 2.4.2 External components)

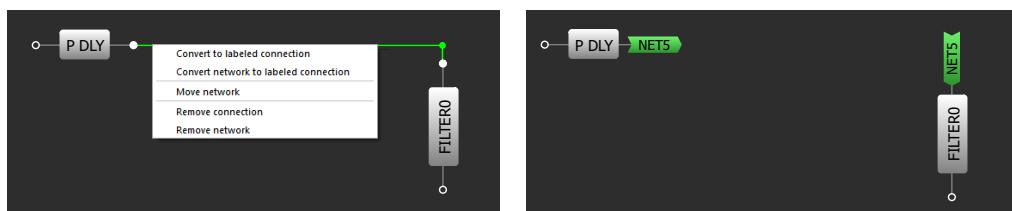
SLG51000/1 chip connection highlight

In SLG51000/1 Part Number there are two connection types:

	<i>Matrix</i>	the most common connection which can be set between green pins
	<i>Hard-wired</i>	connections which are predefined and depend on block settings

Common connection-related behavior for all Part Numbers

A wired connection/network can be converted to labeled one. Trigger the context menu upon right-clicking the wire to see such option.



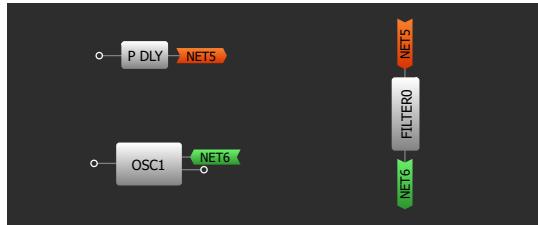
Convert a wire to label

After conversion, the label color remains the same as its wired version (Exception: different connection types set from one OUT. In this case, label is yellow. Applicable only to GreenPAK Part Numbers). Labels and pin colors during connection are also the same.

The connections can be labeled by default, but you can change them to wired upon need. You can

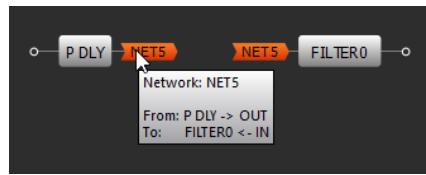
also change the labeled connection's name. Double-click the label and enter a new name in a pop-up window.

After hovering the mouse over a label, it becomes highlighted along with all other parts of the same connection/network.



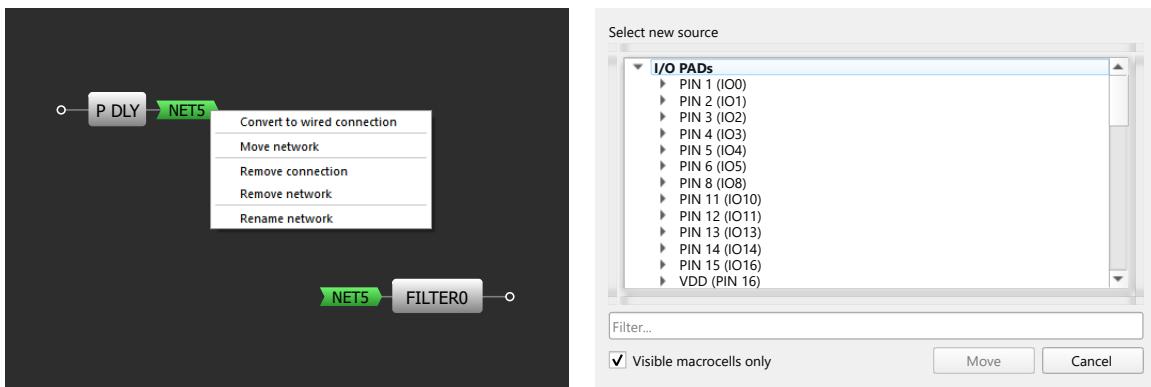
Highlighted label

You can also see the hint with the connection or network info upon hovering the cursor over the wire or label.



Connection/Network hint

You can move the connection/network to another OUT. Select *Move network* from the context menu. Choose the new component in a *Move network* window and click *Move*.



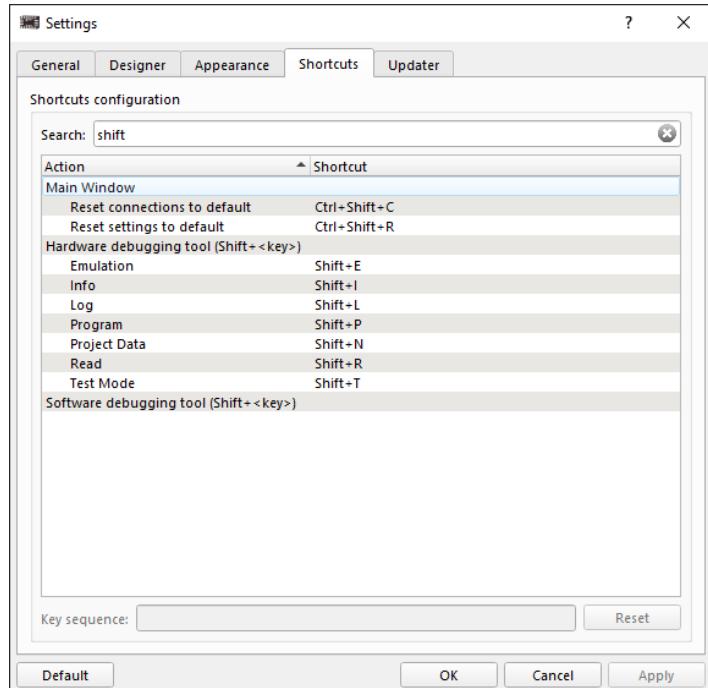
Move connection/network

The information about the components, pins, connections colors and their description is also present in *Go Configure Software Hub*, in *Legend* box. You can read more is section [2.1.8 Legend box](#).

2.1.6 Keyboard commands

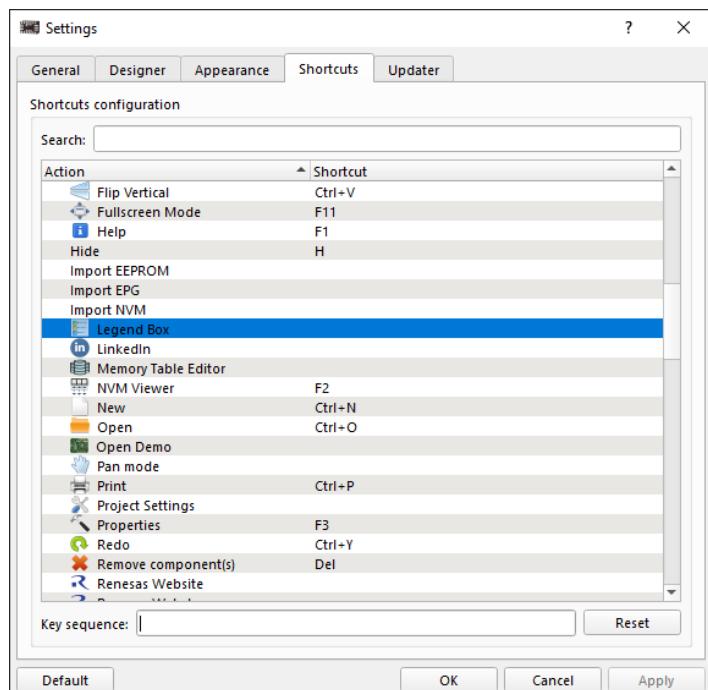
There are configurable and non-configurable keyboard commands. The configurable commands can be managed in *Settings* (for more info, see section [2.1.7 Settings](#)). To find the command easily,

use the search field.



Shortcuts table

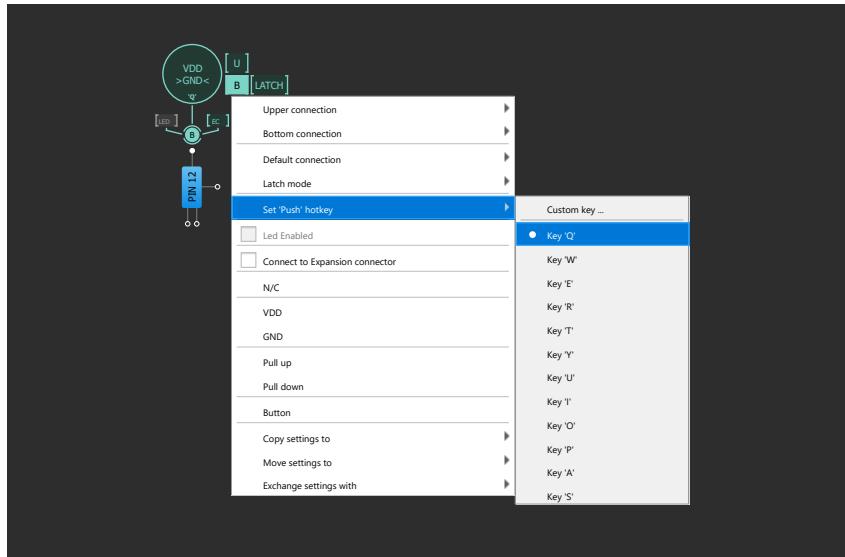
To change or add a shortcut, double-click the action and press the key sequence on your keyboard to add it to the corresponding field. To discard changes, use the *Reset* button.



Assign/Change shortcut

In *Debug tools*, you can assign a hotkey to some hardware sources via the context menu. Use a hotkey to change the source state (read more in section [2.2 Debug tools](#)). Assign a hotkey from

the context menu or create a custom one.



See the list of configurable and non-configurable hotkeys in section [5.2 Keyboard and mouse controls](#).

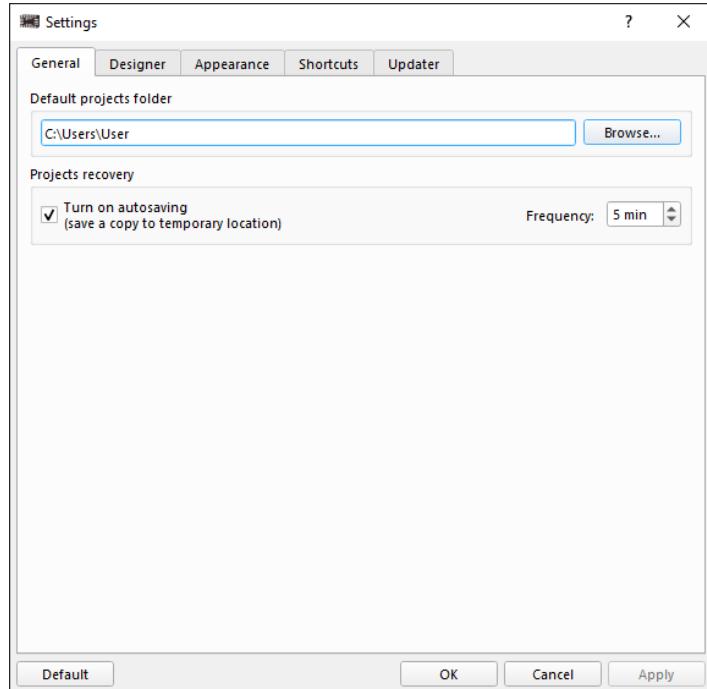
2.1.7 Settings

To reach the settings window open *Options* → *Settings* (on macOS open *App menu* → *Preferences*). The window contains the following tabs: *General*, *Designer*, *Appearance*, *Shortcuts*, and *Updater*.

General

- *Default projects folder* — define the path to your project files
- *Projects recovery* — activate project file autosaving

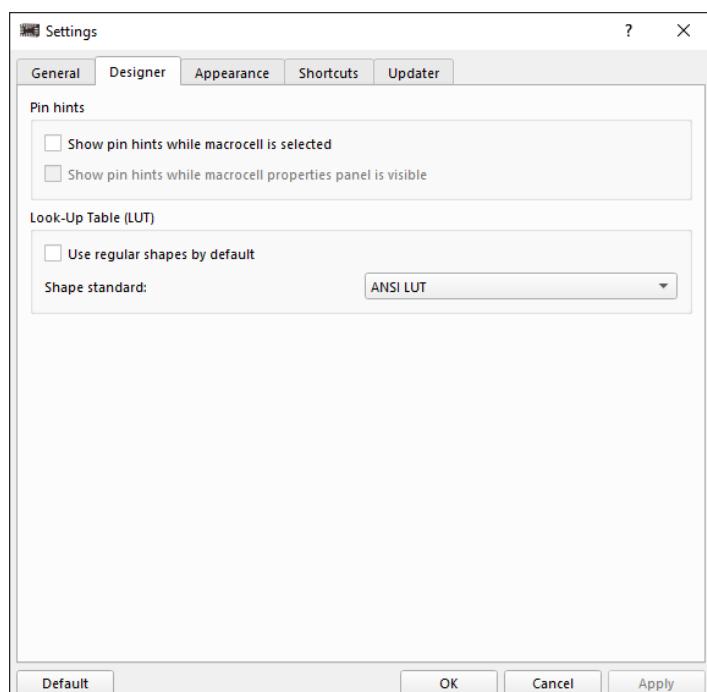
This feature reduces the risk or impact of data loss in case of a crash or freeze. The project file copy is saved to a temporary location at a predefined interval. If a critical issue occurs, the file appears in the *Recovery Files* tab in the *Hub* window (see the location of the *Recovery files* tab in section [2.1 Design tools](#))



General tab

Designer

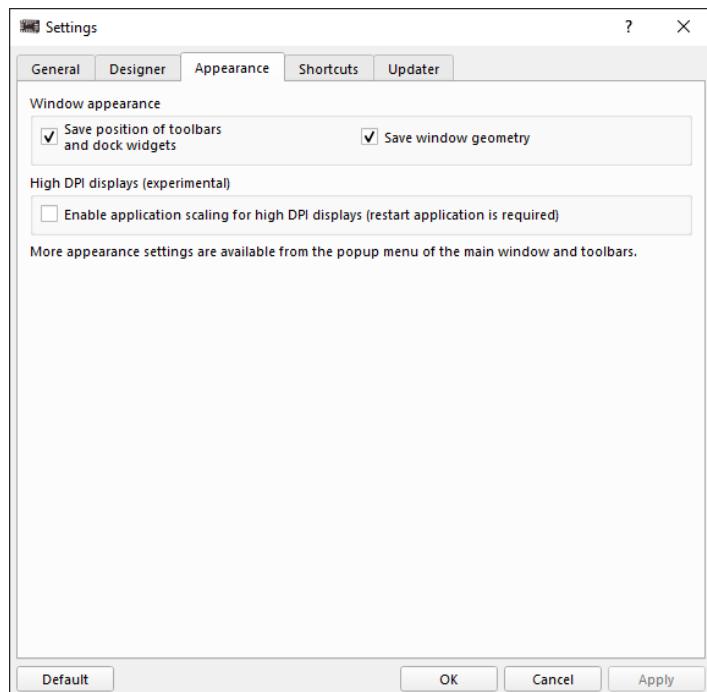
- *Pin hints* — show pin hints while a block is selected or the properties panel of a component is visible (find out more about pin hints in section [2.1.4 Components](#))
- *Look-Up Table (LUT)* — select the preferred LUT shape (regular, ANSI or IEC) using different standard gates



Designer tab

Appearance

- *Window appearance* — save position of the toolbars/dock widgets and window geometry of the workspace
- *High DPI displays* — enable Go Configure Software Hub scaling on high DPI displays



Appearance tab

Shortcuts

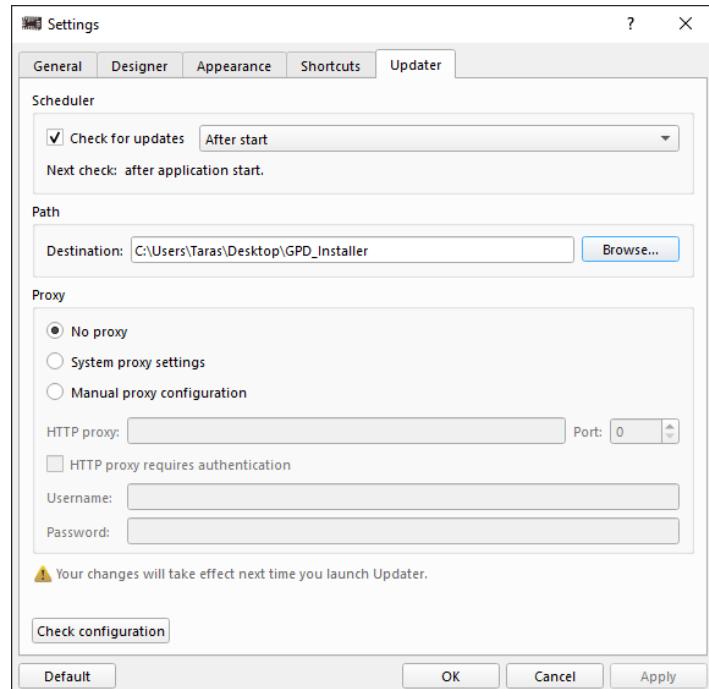
- *Shortcuts configuration* — assign/change the shortcut for the available actions

Read more about shortcuts in section [2.1.6 Keyboard commands](#).

Updater

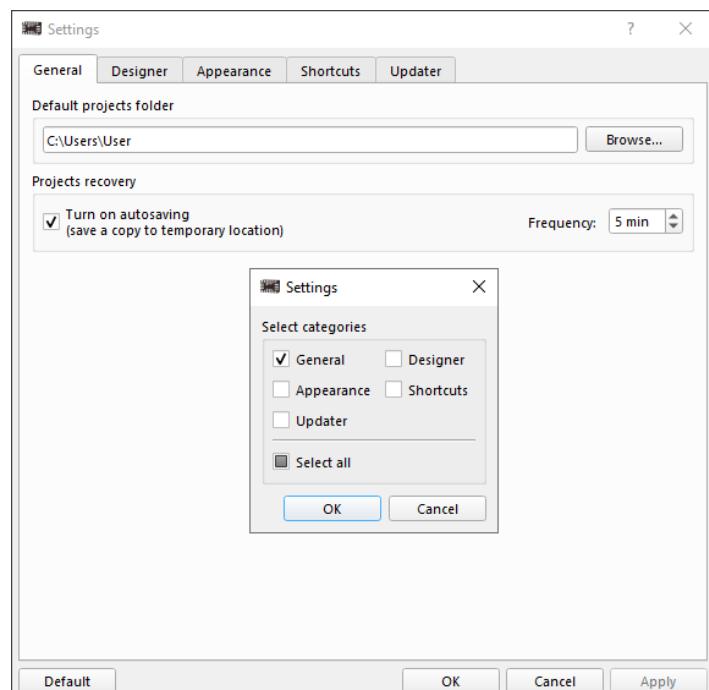
- *Scheduler* — set frequency of the updates check
- *Path* — define a location to download updates to
- *Proxy* — configure a proxy for updates

- Check configuration button — test the connection to the server



Updater tab

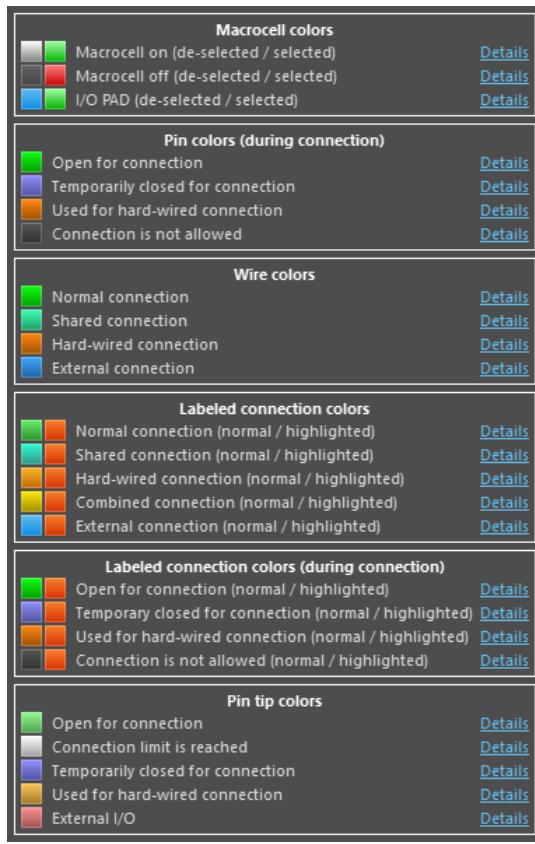
In order to reset the settings, click *Default* button at the bottom left corner of the *Settings* window. You can reset settings for a particular category or all categories at once.



Default button

2.1.8 Legend box

The *Legend box* window shows the color scheme of the components and connections-related features in *Go Configure Software Hub*. The *Legend box* view may vary depending on the Part Number. You can find it in the main menu, *Help* → *Legend box*.



Legend box

Find out more in sections [2.1.4 Components](#) and [2.1.5 Connections](#).

2.1.9 Help window

Help materials provide information about the IC's parameters, components, and tools. There are several ways to reach the *Help* window.

If you prefer to open the unified *Help* materials for a particular Part Number, go to the main menu

→ Help → Help (F1). Walk through the categories to find the information you need.

GreenPAK6 Reference (SLG46811V)

The Renesas SLG46811 GreenPAK is a one-time programmable Mixed signal array available in a small 1.6 mm x 1.6 mm STQFN-12 package.

Features

- Programmable Logic;
- Programmable Mixed Signal Macrocells;
- Read Back Protection (Read Lock);
- 2.3 V to 5.5 V Supply;
- Operating Temperature Range: -40°C to 85°C;
- RoHS Compliant / Halogen-Free;
- 12 Pad STQFN: 1.6 x 1.6 x 0.55 mm, 0.4 mm pitch;

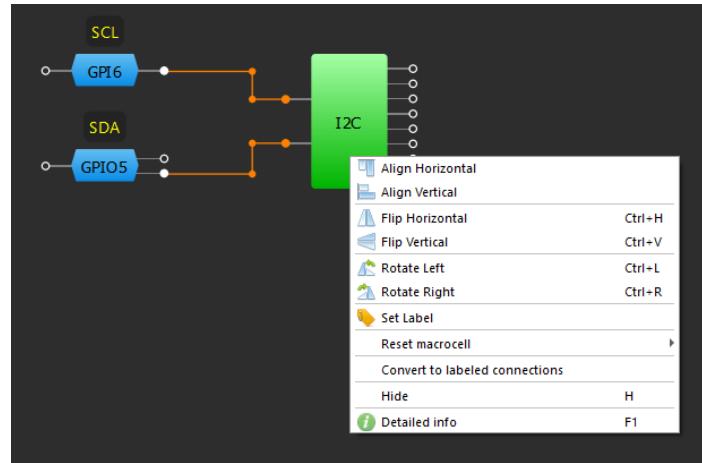
Description

The SLG46811 provides a small, low power component for commonly used Mixed-Signal functions. The user creates their circuit design by programming the one time programmable (OTP) Non-Volatile Memory (NVM) to configure the interconnect logic, the IO Pins, and the macrocells of the SLG46811. This highly versatile device allows a wide variety of Mixed-Signal functions to be designed within a very small, low power single integrated circuit. The macrocells in the device include the following:

- Multichannel Sampling Analog Comparator (MS ACMP):
 - Sampling up to Four Analog Channels;
 - Selectable Voltage Reference for Each Channel;
 - Different Sampling Scenarios;
 - Synchronous or Asynchronous Result Appearance;
- Integrated Voltage References (VREF);
- Twelve Combination Function Macrocells:
 - Two 2-Bit LUT or (DFF/LATCH) Macrocells;
 - One Selectable Programmable Pattern Generator or 2bit LUT;
 - Four 3-Bit LUT or (DFF/LATCH) with Set/Reset;
 - Four Selectable DFF/LATCH or 3-bit LUTs or Shift Registers;
 - One 4-Bit LUT or (DFF/LATCH) with Set/Reset Macrocell;
- Six Multi-Function Macrocells:

Help window

To open the *Help* window for a selected block, right-click the component and select the *Detailed info* (i) option (F1) in the context menu.



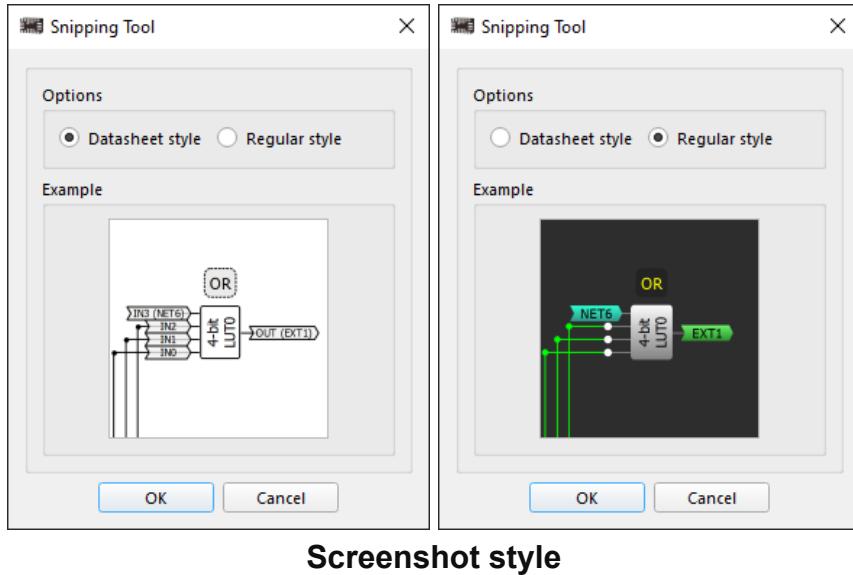
Detailed info

You may also see the information buttons (i) in different workspace locations, e.g., on a component's *Properties* panel or *Project settings* window. Clicking the button also opens the *Help* window.

2.1.10 Snipping tool

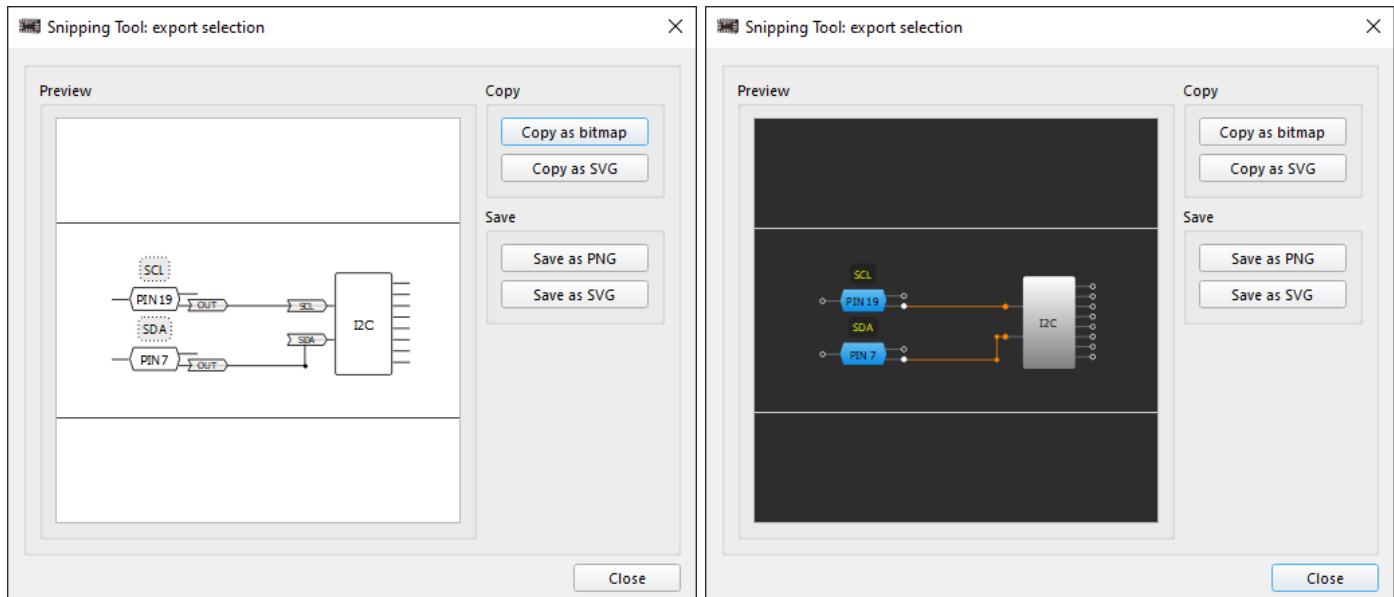
Snipping tool is a work area screenshot-maker. You can find the tool in the main menu, *Tools* → *Snipping tool*.

Choose between the *Datasheet* and *Regular* screenshot style (the *Datasheet* style view is the same as in *Print Preview*)



Screenshot style

Once you select the area, copy/save the file in the supported format.



Supported formats

2.1.11 Rules checker

The *Rules Checker* tool scans a project for errors related to incorrect block connections or settings. To check the design, click *Rules Checker* on the toolbar (also, find the tool in the main menu → *View/Tools*).

The *Rules Checker* output window consists of three main columns:

- *Event* — shows the message type (Fail, Warning, Note)
- *Rule* — explains the essence of the issue
- *Note* — suggests the way to correct the error or provides more details about the issue

Rules Checker output			
Time	Event	Rule	Note
23:37:22	Fail	3-bit LUT2/DFF/LATCH2: The truth table is configured incorrectly.	The truth table is configured so that all combinations of the inputs that are connected to the macrocells, do not cause changes on the output.
23:37:22	Fail	3-bit LUT3/DFF/LATCH3: The truth table is configured incorrectly.	The truth table is configured so that all combinations of the inputs that are connected to the macrocells, do not cause changes on the output.
23:37:22	Fail	VDD and temperature parameters must be filled in.	Please check project specs in Project Settings.
23:37:22	Warning	3-bit LUT2/DFF/LATCH2: No input connected.	3-bit LUT2/DFF/LATCH2's input is not connected.
23:37:22	Note	3-bit LUT3/DFF/LATCH3: macrocell is placed in the schematic but not used.	3-bit LUT3/DFF/LATCH3 is placed in the schematic but has no connection. Please hide the macrocell or make some connection.

Rules checker window

You can use controls to sort the messages by their type. Click the *Refresh* button to see the latest events.

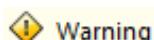


Rules checker controls

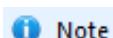
Rules Checker window shows three message types:



A critical error that may cause the project to fail is present in the design. In some cases, you can ignore this message type (the program may consider the design solution to be erroneous, but it is the correct solution for you).



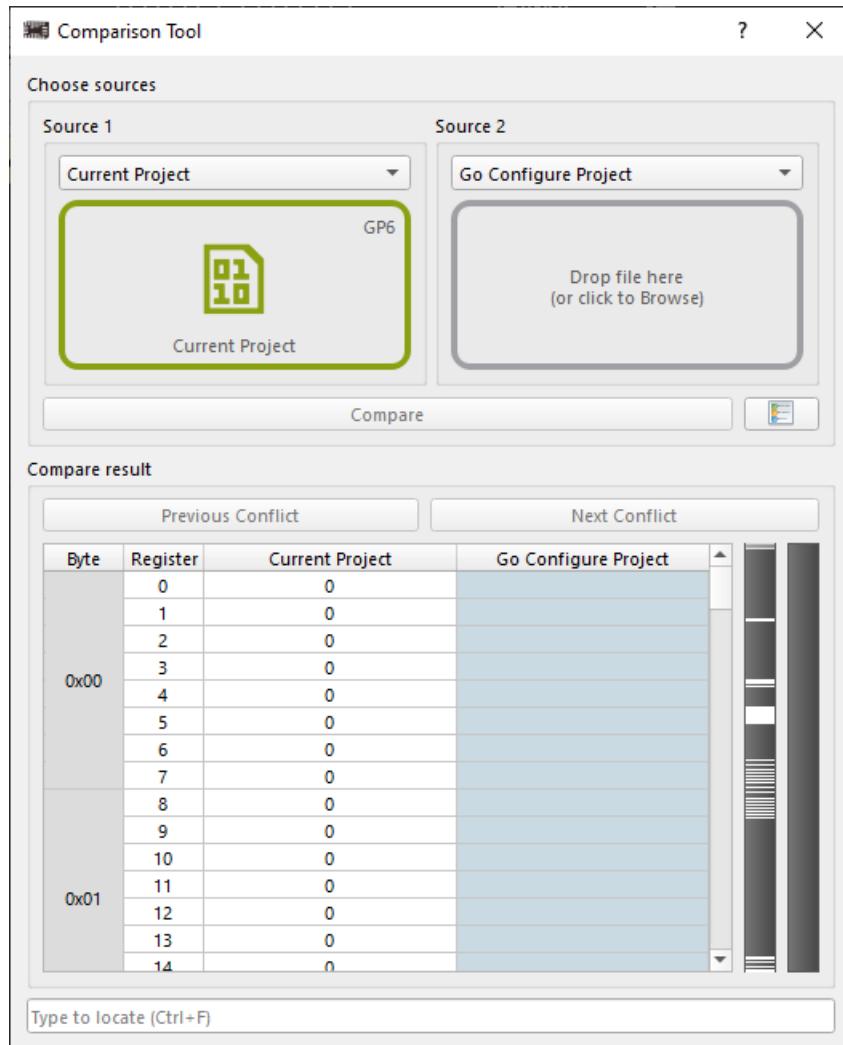
The project contains improper block(s) connections or settings. This message does not necessarily imply an error, but it notifies about a potential problem and urges checking the block(s) connections and settings.



This message type suggests minor improvements in the design. The suggestions are optional and can be ignored.

2.1.12 Comparison tool

The *Comparison tool* allows you to compare the NVM state of two projects. You can open the tool by clicking the *Comparison tool* widget on the toolbar or by reaching the main menu → *Tools* → *Comparison tool*.



Comparison tool

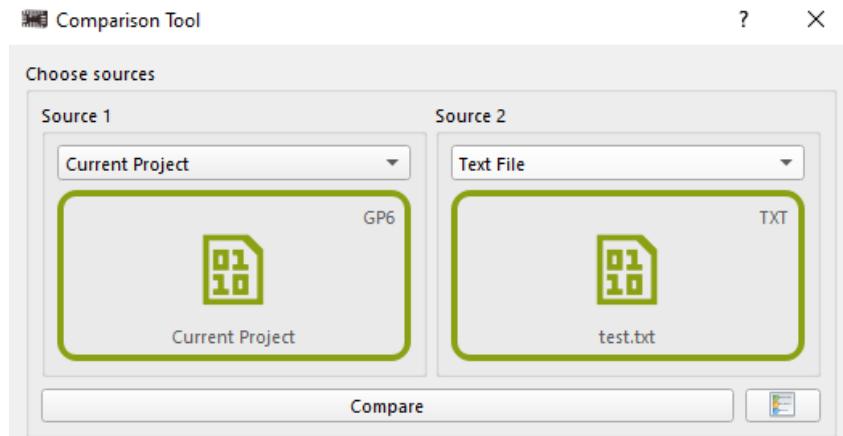
The tool consists of two main parts: *Choose sources* and *Compare result*.

Choose sources

Here you can select the sources for the NVM comparison. Select the source types from the dropdowns. Then click the designated area to choose the file or drag and drop it. The available sources are as follows:

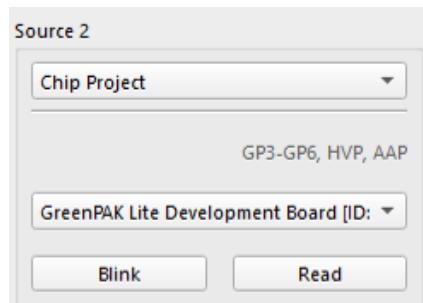
- *Current Project* — current state of your design
- *Go Configure Project* — GCSH project file with the following extensions: .aap, .can, .gp3, .gp4, .gp5, .gp6, .hvp, and .ppak
- *Text File* — exported NVM file in .txt format

➤ *Chip Project* — configurations programmed on a chip



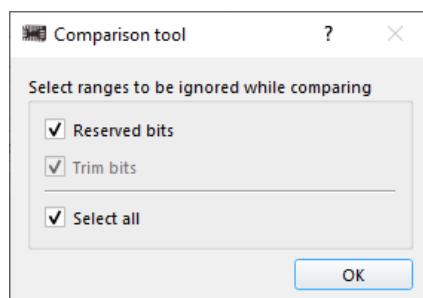
Choose sources

If you select *Chip Project* as a source, make sure you connect the supported platform with the inserted IC to your computer. The board info appears in the designated area. Click *Read* to add the programmed data to the tool. Clicking the *Blink* button triggers the blue LED on the connected platform. This feature is useful to check which board is currently detected by the *Go Configure Software Hub* instance. If multiple platforms are connected, select the desired one in the dropdown. Note that *Debug tool* should be disabled to use *Chip Project* as a source.



Chip source

The *Compare* button becomes active once two sources are selected (for *Chip Project*, after the source is selected, click *Read*). Upon clicking *Compare*, you can see a pop-up, where you can select the ranges to ignore while counting the conflicts.



Range selection

Once the pop-up is closed, the *Comparing* window with the results appears.

Compare results

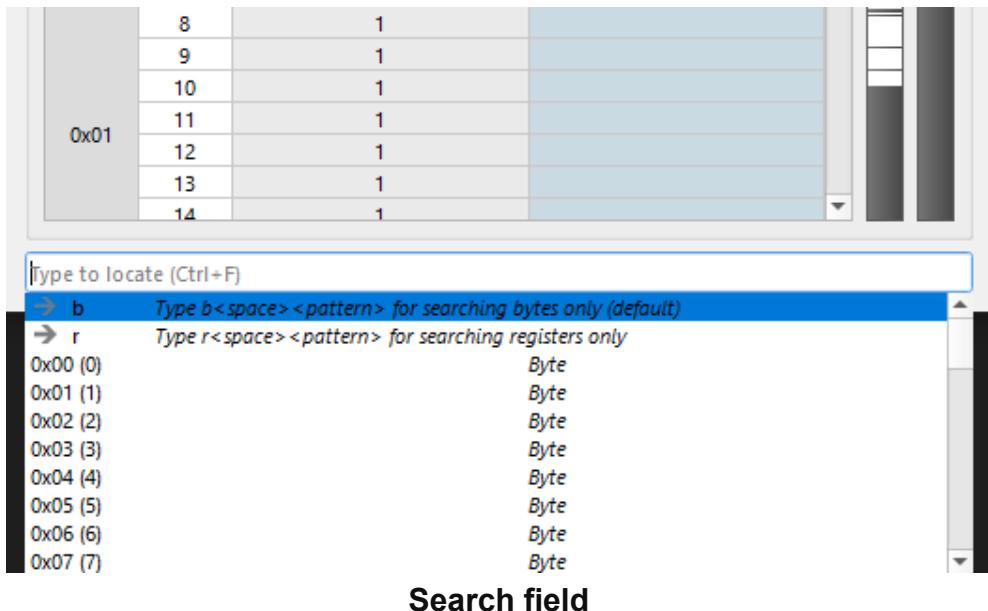
After the sources are compared, you can see the table with the results. Navigate through the unmatched bits using *Previous Conflict* and *Next Conflict* buttons.

The screenshot shows the 'Comparison Tool' window. In the 'Source 1' section, 'Current Project' is selected, and its icon (GP6) is shown. In the 'Source 2' section, 'Go Configure Project' is selected, and its icon (GP6) is shown. Below these, a 'Compare' button is visible. The main area is titled 'Compare result' and contains a table and two vertical bar charts. A 'Color scheme' dialog box is overlaid on the table, explaining the meaning of colors: white for 'Register set to 1', red for 'Mismatches', and yellow for 'Ignored mismatches'. The table has columns for 'Byte', 'Register', and 'NVM'. Rows are grouped by register number (0x00, 0x01). The first byte of each group is '0'. The second byte of the first group is '1'. The third byte of the first group is '2'. The fourth byte of the first group is '3'. The fifth byte of the first group is '4'. The sixth byte of the first group is '5'. The seventh byte of the first group is '6'. The eighth byte of the first group is '7'. The ninth byte of the first group is '8'. The tenth byte of the first group is '9'. The eleventh byte of the first group is '10'. The twelfth byte of the first group is '11'. The thirteenth byte of the first group is '12'. The fourteenth byte of the first group is '13'. The fifteenth byte of the first group is '14'. The first byte of the second group is '0'. The second byte of the second group is '1'. The third byte of the second group is '2'. The fourth byte of the second group is '3'. The fifth byte of the second group is '4'. The sixth byte of the second group is '5'. The seventh byte of the second group is '6'. The eighth byte of the second group is '7'. The ninth byte of the second group is '8'. The tenth byte of the second group is '9'. The eleventh byte of the second group is '10'. The twelfth byte of the second group is '11'. The thirteenth byte of the second group is '12'. The fourteenth byte of the second group is '13'. The fifteenth byte of the second group is '14'. The vertical navigation bar on the right indicates the location of 0 and 1 register values, along with conflicts and ignored bits. At the bottom, there is a search bar with the placeholder 'Type to locate (Ctrl+F)'.

Compare results and color scheme

The table contains the information about a byte, register, and NVM of the selected sources. The vertical navigation bar indicates the location of 0 and 1 register values, along with conflicts and ignored bits. See the color explanations by clicking the *Color scheme* button.

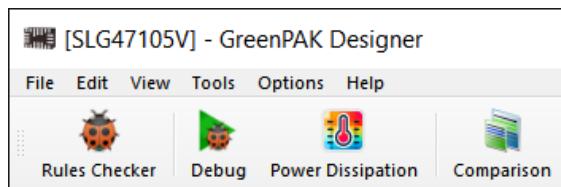
You can also use the search field to find the particular byte or register.



2.1.13 Power Dissipation Calculator

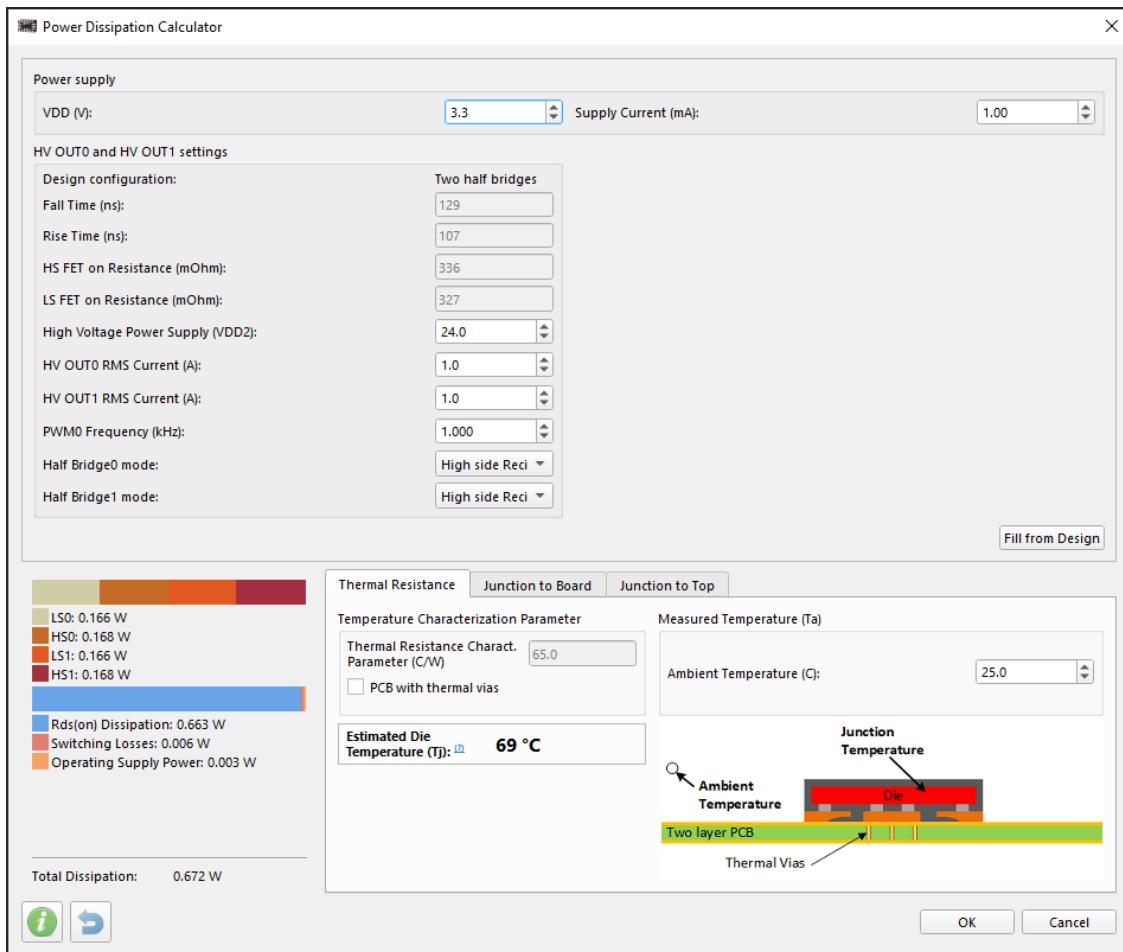
A crucial step in developing an effective thermal management solution is to determine the amount of heat dissipation of the device. The tool allows you to automatically calculate this parameter.

The *Power Dissipation Calculator* is available only in the HVPAK and mainly applies to the HV OUT CTRL macrocell.



Power Dissipation Calculator on the toolbar

Click the *Power Dissipation* to launch the tool.



Power Dissipation Calculator preview window

This window consists of four main parts:

- *Power supply* — adjust the supply voltage for the chip and set the current that the design draws when the HV OUT CTRL output pins are in the *Hi-Z* state.



Power supply

- *HV OUT 0 and HV OUT 1 settings* — the main parameters of the power pins used in the design.

The first parameter shows the *Design configuration* corresponding to the HV OUT CTRL.

The *Fall Time*, *Rise Time*, *HS FET on resistance (mOhm)*, and *LS FET on resistance (mOhm)* parameters are fixed and cannot be adjusted. They are automatically adjusted based on the conditions and configuration of HV OUT CTRL.

The *High Voltage Power Supply* allows you to set the supply voltage for HV OUT CTRL.

The HV OUT RMS current is responsible for the output current and cannot exceed 1.5 A per HV OUT

PWM 0 Frequency (kHz) determines the frequency at which HV OUT CTRL operates. You can set the frequency value using the PWM 0 macrocell. Additionally, you can automatically transfer the frequency value using the *Fill from Design* button in the lower right corner of the HV OUT 0 and HV OUT 1 settings section.

The screenshot shows the 'HV OUT0 and HV OUT1 settings' window. It contains the following fields:

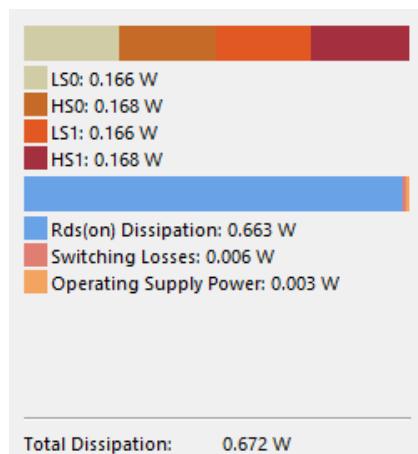
- Design configuration: Two half bridges
- Fall Time (ns): 129
- Rise Time (ns): 107
- HS FET on Resistance (mOhm): 336
- LS FET on Resistance (mOhm): 327
- High Voltage Power Supply (VDD2): 24.0
- HV OUT0 RMS Current (A): 1.0
- HV OUT1 RMS Current (A): 1.0
- PWMO Frequency (kHz): 1.000
- Half Bridge0 mode: High side Reci
- Half Bridge1 mode: High side Reci

A 'Fill from Design' button is located in the bottom right corner.

HV OUT 0 and HV OUT 1 settings

Note: When the output pins connected to the HV OUT CTRL macrocell are in the *Hi-Z* state, you cannot change the settings of HV OUT0 and HV OUT1 in the Power Dissipation Calculator.

- Power dissipation bar — within this window segment, you can review the power dissipation of each transistor: *LSx (low side)* and *HSx (high side)*. Below is the total power dissipation, including open transistor resistance *Rds(on)*, transistor switching losses, and the power dissipated by the chip when the output pins of the HV OUT CTRL are in *Hi-Z*.

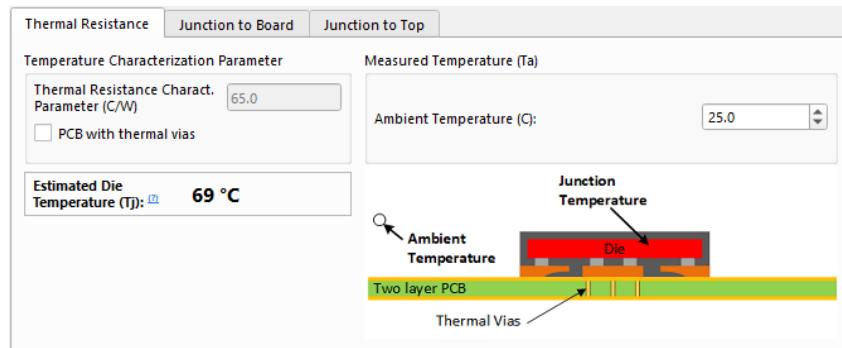


Power dissipation bar

- Temperature Characterization parameter — provides three options for calculating the device *Die* temperature:

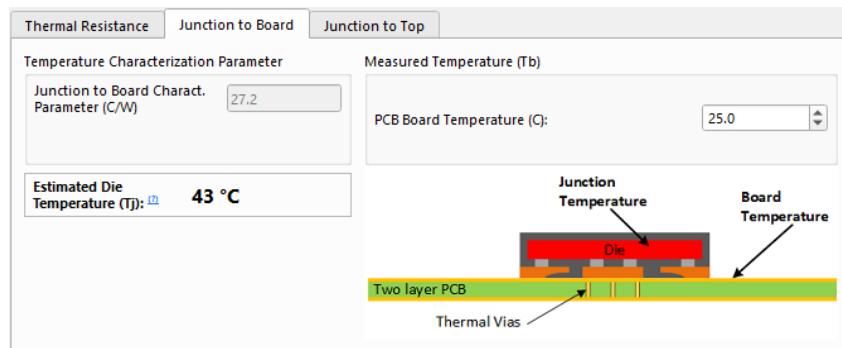
- Thermal Resistance — you need only the ambient temperature of the device for calculations. You can set this in the *Ambient Temperature* section. The Ambient Temperature point shows where the temperature measuring device is located under these conditions. This method is the least accurate.

This section has an additional setting: PCB with thermal vias. Enable this if you use a PCB with thermal vias, which provides better heat dissipation.



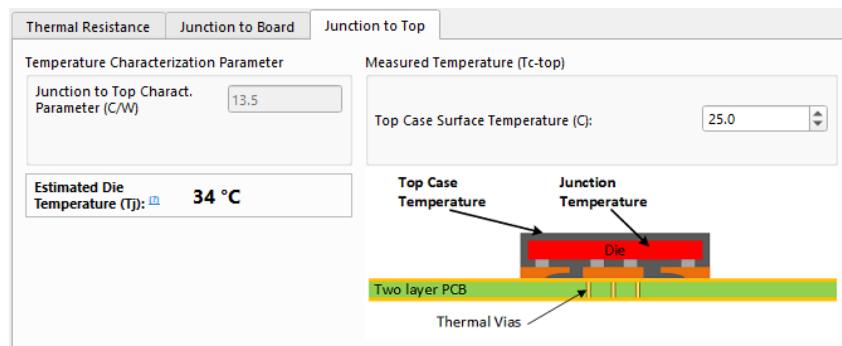
Thermal resistance window

- Junction to Board — the PCB temperature is used to calculate the *Die* temperature, which you can set in the *PCB Board Temperature (C)* window. The PCB board temperature should be measured 1 mm from the device. This method has a higher accuracy compared to Ambient Temperature.



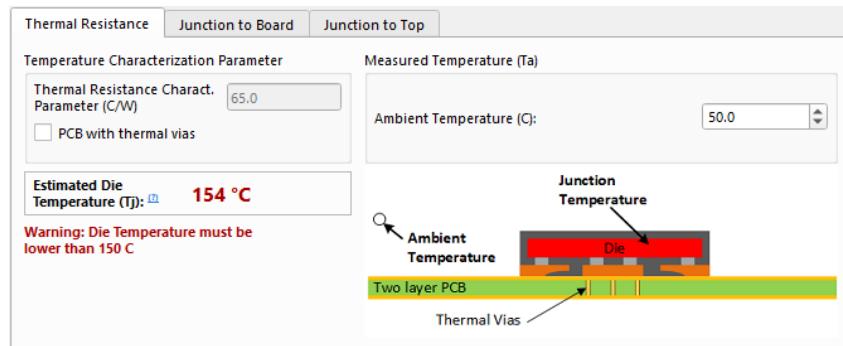
Junction to Board window

- Junction to Top — the temperature of the *Die* is determined based on the upper surface temperature of the chip. This method has the highest accuracy.



Using Junction to Top window

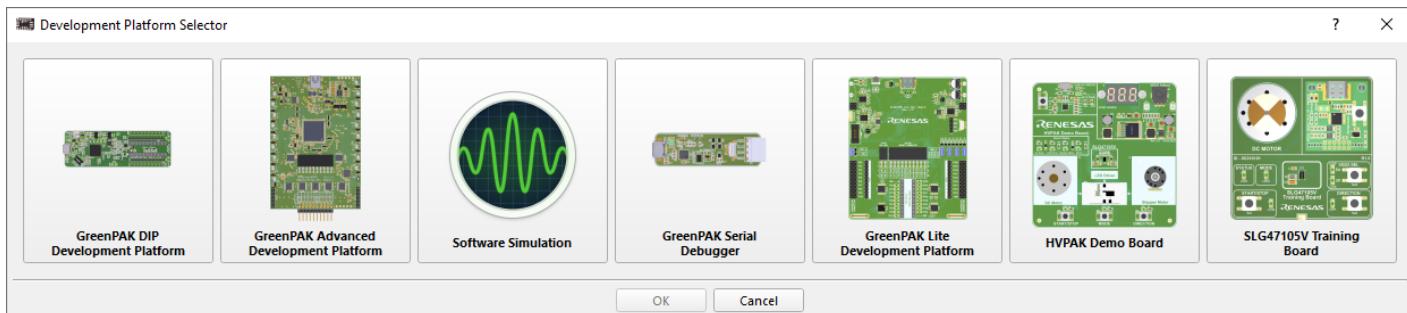
Keep the *Die* temperature below 150°C to ensure the chip operates correctly. An indication of overheating appears if the temperature exceeds the limit.



Overheating warning

2.2 Debug tools

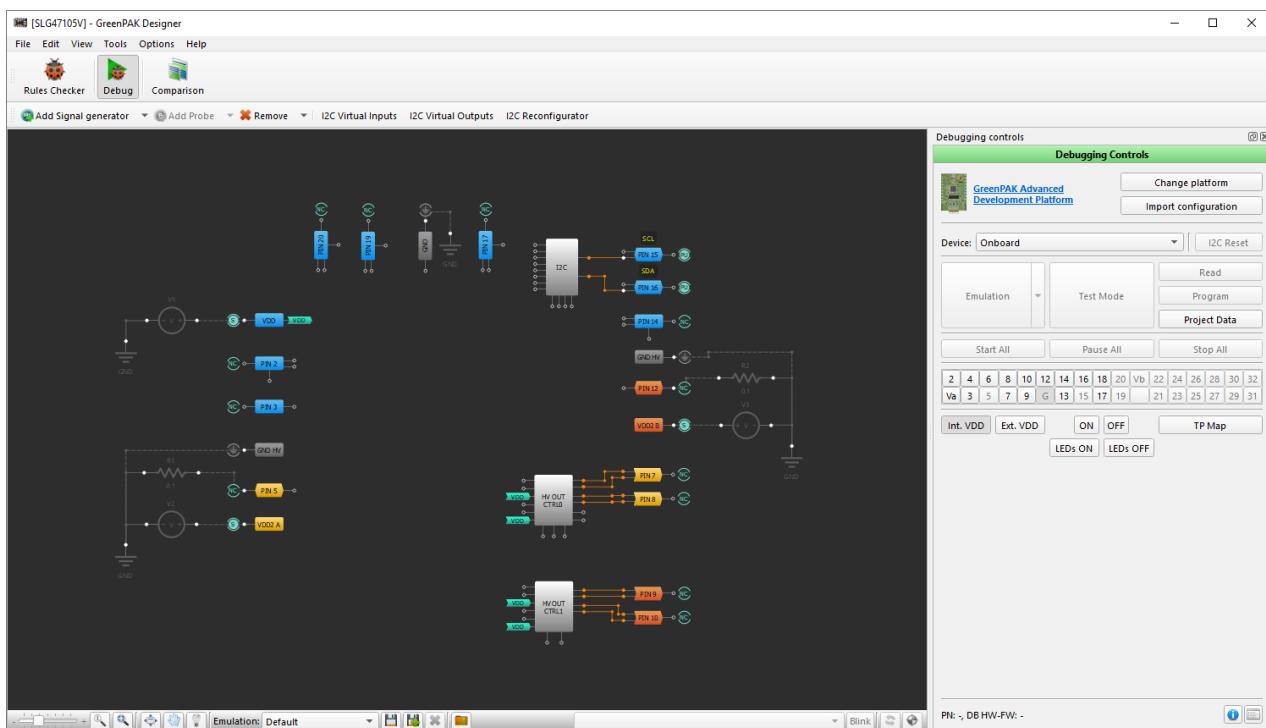
Debug tools are a set of instruments, that allow you to test and debug your design. To access *Debug tools*, click the *Debug* button on the toolbar. Since there are different hardware platforms available for a specific Part Number, select the platform most suitable for your project.



Platform selector window for SLG47105V

Note: You can only use the hardware in one application instance. If you wish to transfer control from one instance to another, disable *Debug tools* on the controlling instance.

After you select a platform, the software activates a toolbar and a panel with controls for main procedures, including emulation and chip programming.



Software UI after Debug is enabled

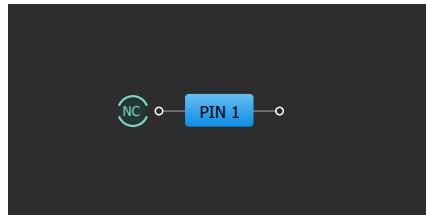
2.2.1 Hardware sources

Go Configure Software Hub provides software tools to configure varied hardware sources that manage or generate input and test signals for a chip. Each signal source is connected to an external pin on a chip.

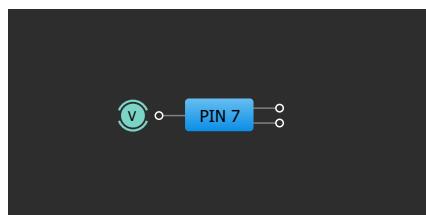


Below, you can find the complete list of all the hardware sources.

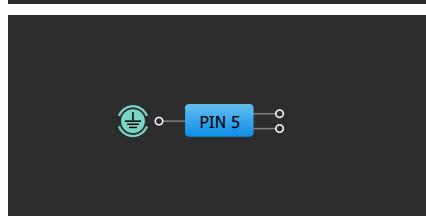
No source (Not Connected)



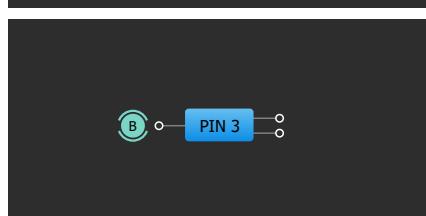
VDD



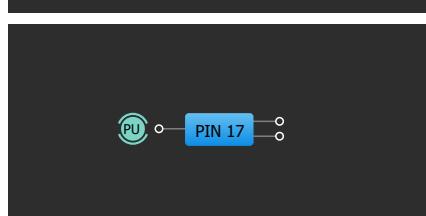
GND

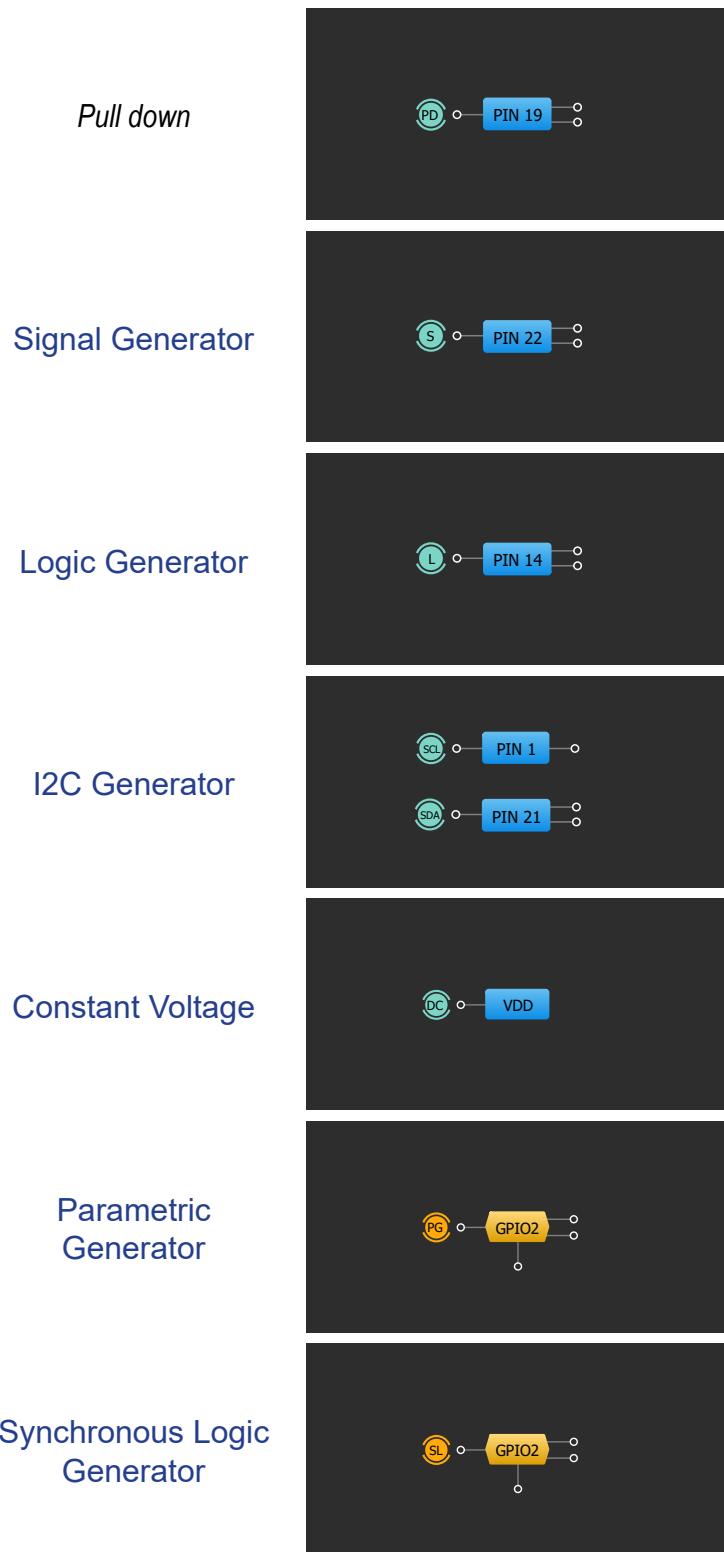


Button



Pull up

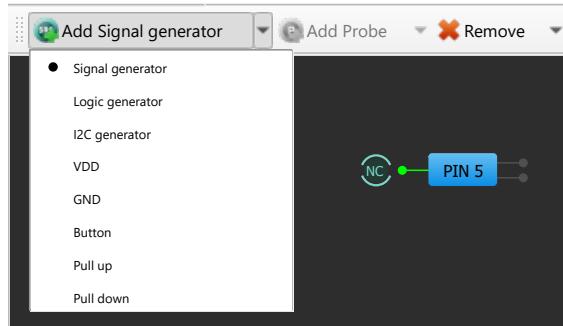




Note: The choice of hardware sources depends on the development platform features and chip restrictions.

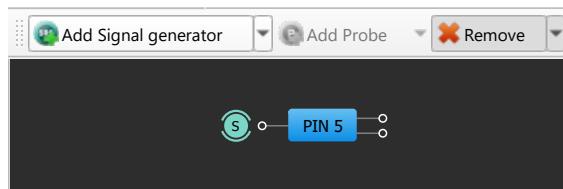
You can switch the controls on/off via the context menu or from the toolbar via the *Add Signal*

Generator or Add VDD button (the button view depends on the available sources).



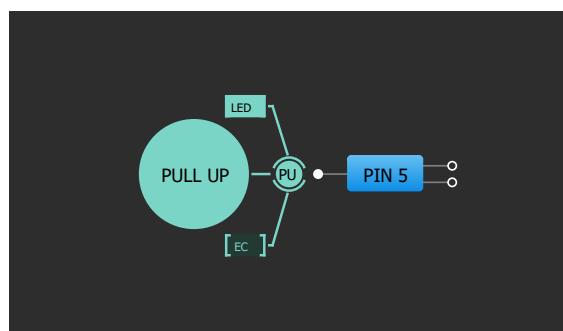
Debug toolbar

To remove the source click the Remove button or select N/C (Not Connected) in the context menu.



Remove button on Debug toolbar

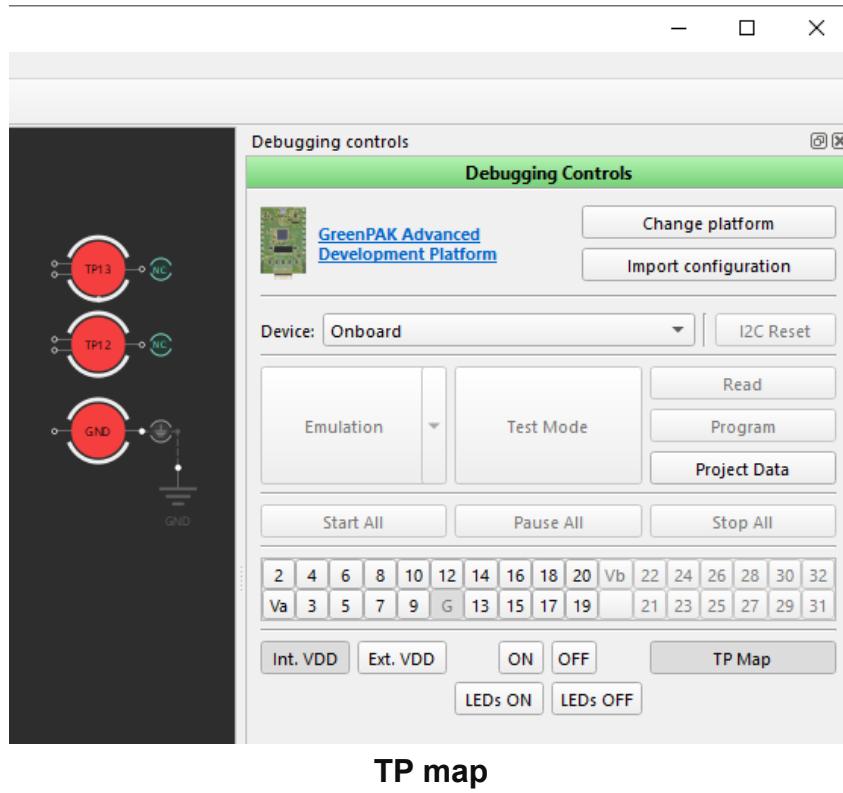
Some of the chip pins may have additional controls, such as *LED indicators* or *Expansion Connectors (ECs)*, which are available even if a development platform is not connected yet. You can enable/disable the controls by hovering over the source and clicking the control you need.



Buffered LED

If you need to identify the chip pin connected to a specific test point on a board, use the *Test Point*

(TP) Map tool. Note that the pins mapping to test points varies based on the chip type.



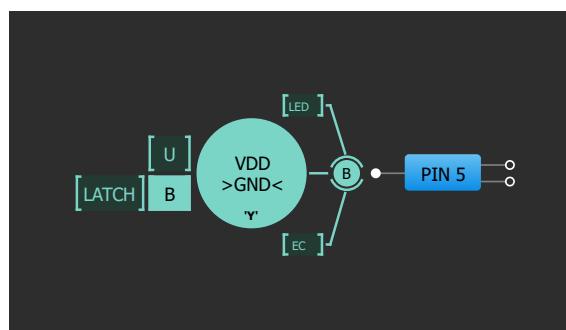
TP map

Hardware sources can be divided into two categories: *basic Hardware Sources* and *Generators*. Generators provide a comprehensive solution for creating analog or digital signals with configurable settings and are discussed in more detail in the section [2.2.2 Generators](#). Later in this chapter you can read about the *basic Hardware Sources*.

Basic Hardware Sources

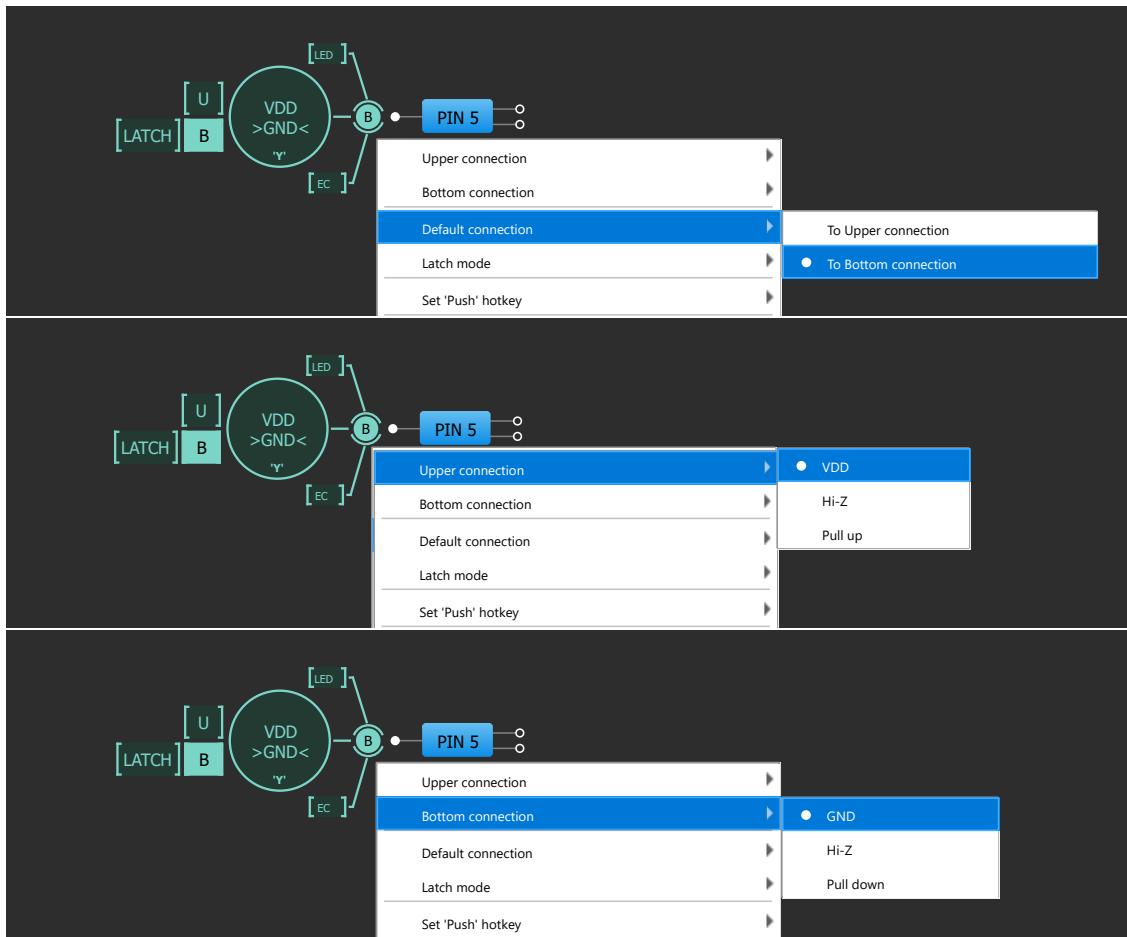
The available *basic Hardware Sources* are: *VDD*, *GND*, *Pull up*, *Pull down*, and *Button*. Unlike others, *Button* has more configuration possibilities (for the rest of the basic sources, all necessary information is already described above).

The *Button* hardware source allows quick switching between two predefined states: *Upper connection (U)* and *Bottom connection (B)*. These predefined states can be set to *VDD/GND*, *High-Z*, or *Pull Up/Down*. Hover the mouse cursor over the *Button* control to see its configuration.



Configurable Button

The default connection can be set to either the *Upper connection (U)* or the *Bottom connection (B)*.

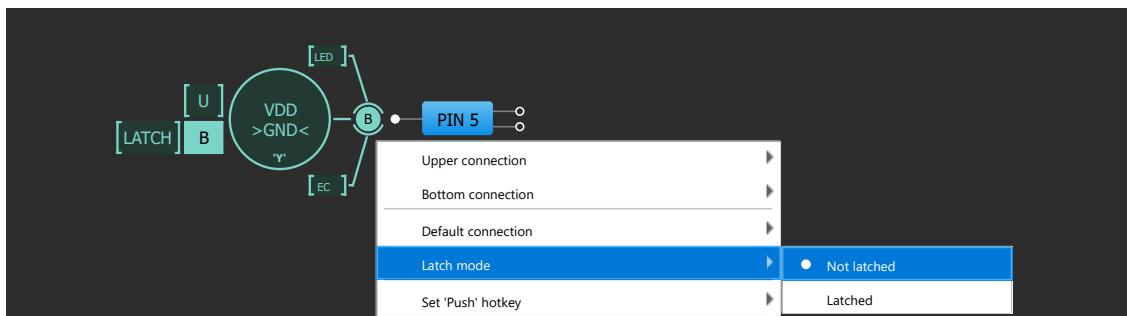


Default Key Connection

Latch control has two modes: *Latched* and *Not latched*. You can configure these modes from the context menu or by clicking the *LATCH* button to change the value.

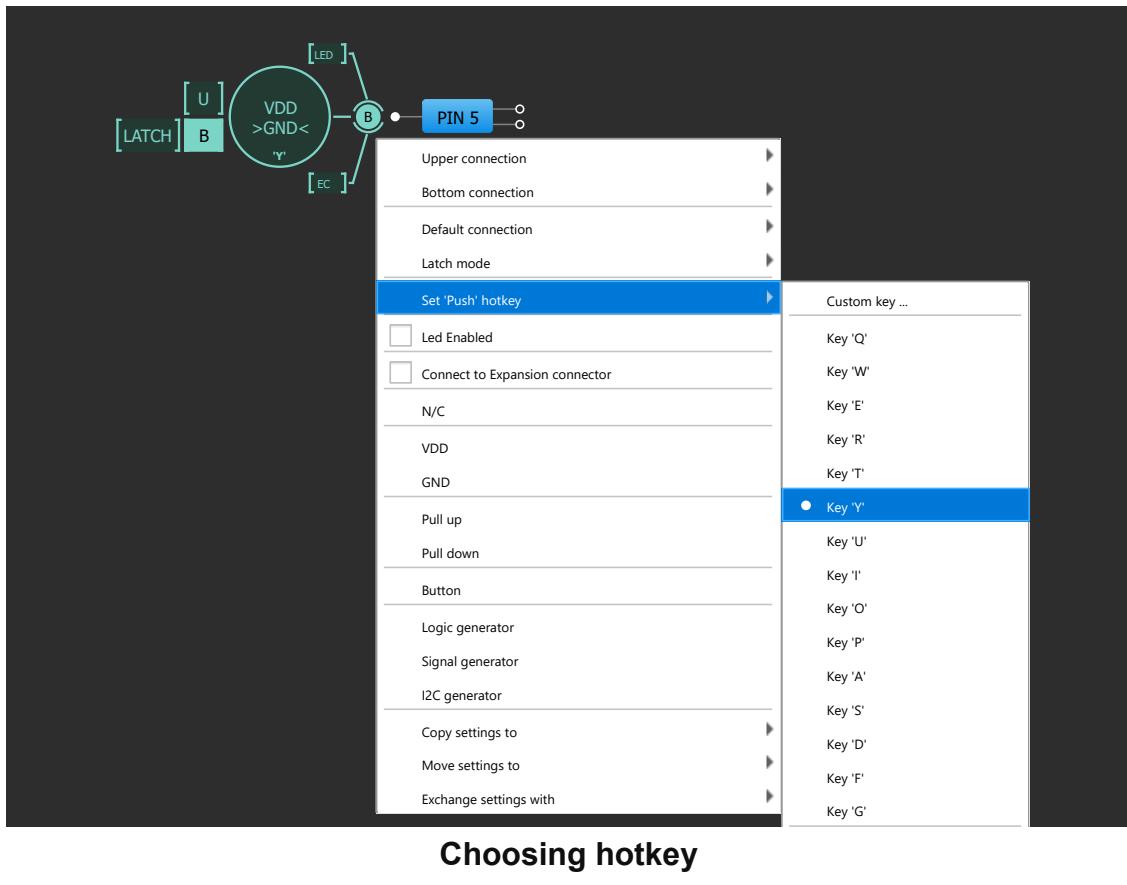
In *Latched* mode, the *Button* toggles its state on click and remains in the new state until the next click.

In *Not Latched* mode, the *Button* changes its state upon the left-click and returns to its previous state after the mouse button released.



Key Mode

You can assign a hotkey for the *Push* action. Pressing the hotkey is equal to a mouse click.



You can assign the same hotkey to multiple *Buttons*, which allows changing the states of all the *Buttons* with the same hotkey at once.

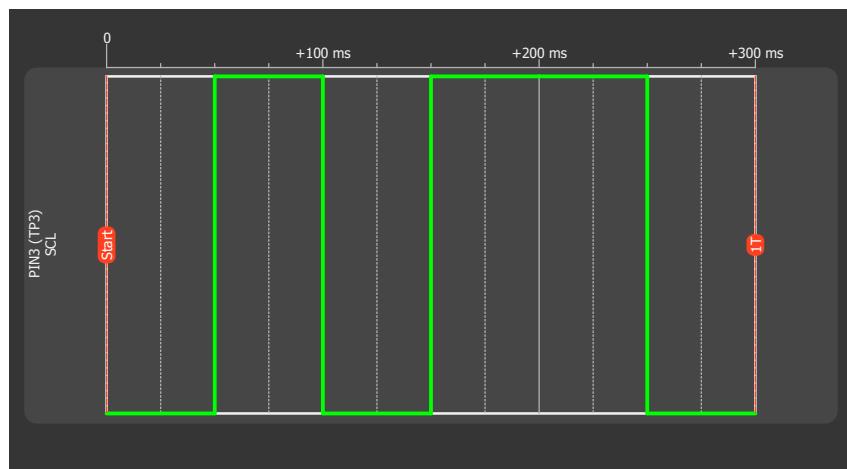
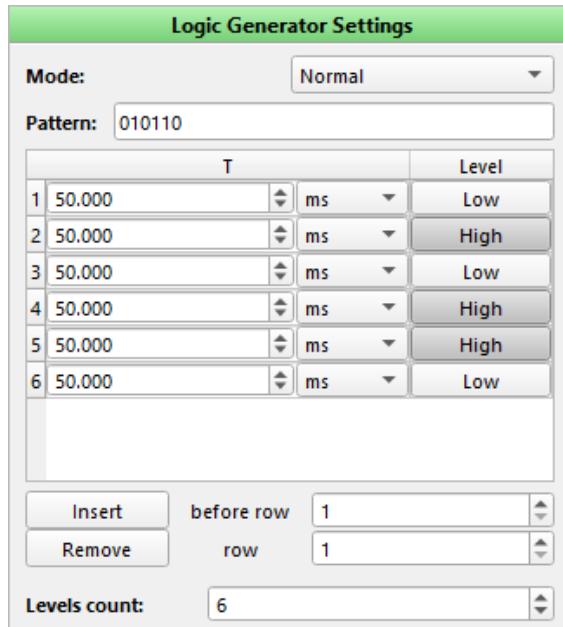
2.2.2 Generators

A *Generator* is a hardware source type, that produces analog or digital electronic signals onto test points on a board. Go *Configure Software Hub* allows setting up the generators in an easy and convenient way with visual control of their settings and states. You can control the *Generator* using the sticker, which appears upon hovering over the respective icon. For more advanced configurations use the *Signal Wizard* tool. To find out more refer to section 2.2.3 *Signal Wizard*.

Logic Generator

The *Logic Generator* generates logic pulses. A logic pulse is one of two voltages that correspond to

two logic states (*low state* and *high state*, 0 and 1).



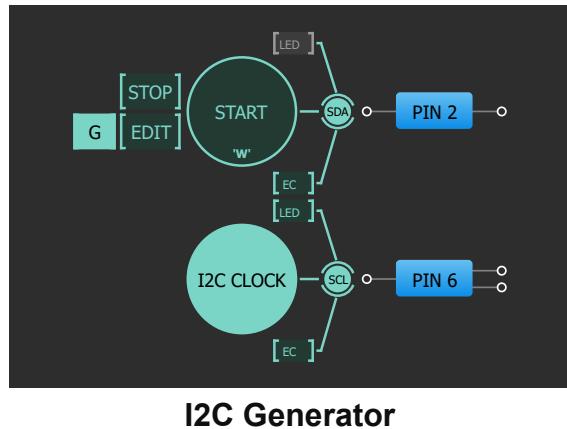
Logic Generator in Signal Wizard

You can export/import all *Logic Generator* settings. To do this, copy the *Pattern* field content and paste it to a text editor. The content will automatically convert to XML format text. You can use this feature to save your custom generators or load them from an external file.

I²C Generator

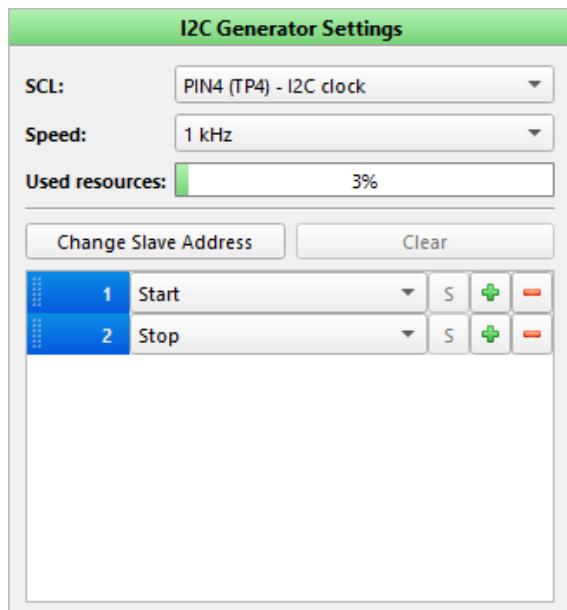
The *I²C Generator*'s purpose is to create *I²C* (Inter-Integrated Circuit) communication patterns based on *Logic Generators*. There are two *Logic Generators* combined as *SDA* (Serial Data) and *SCL* (Serial Clock) lines. You can combine predefined *I²C* primitives to generate a required waveform

appropriately and choose the *SCL* frequency.

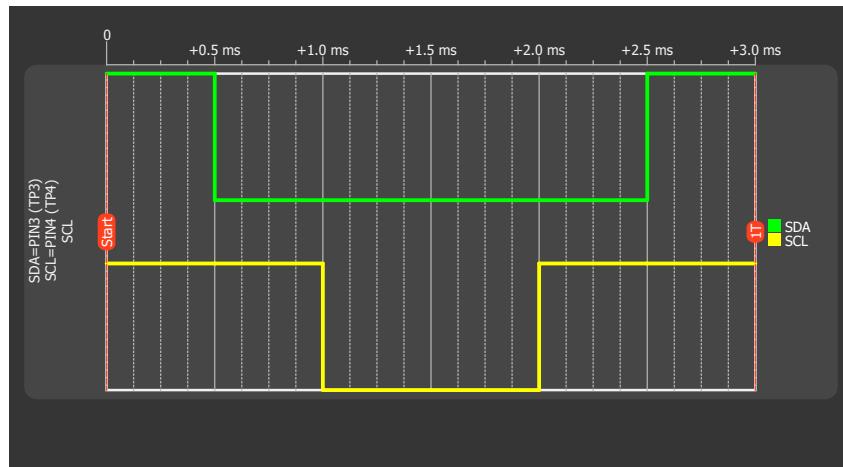


SDA signal is a special case of *Logic Generator* used for sending data via *I2C*. The *Signal Wizard* editor shows the sequence of commands.

SCL signal is a particular *Logic Generator* that can be used for board configuration only. The *SCL* is configured by choosing a predefined frequency. The set of these frequencies depends on the development platform.

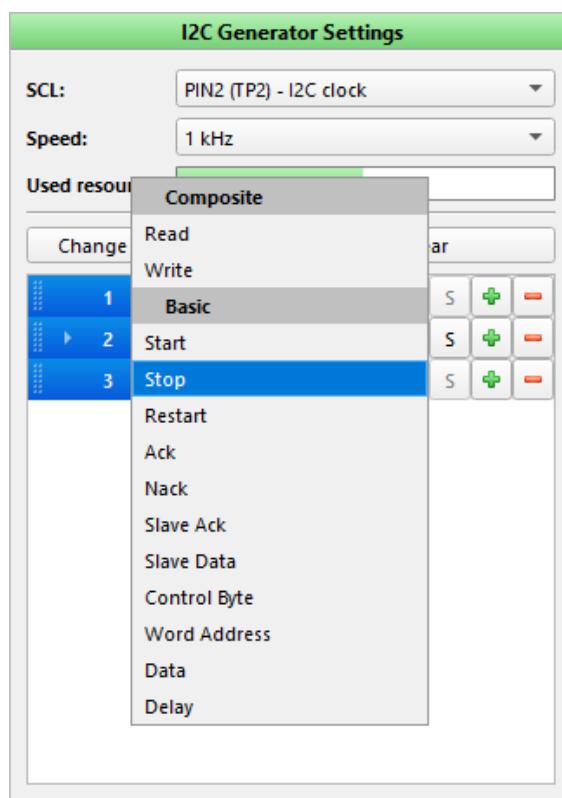


I2C Generator command editor



I2C Generator Signal Wizard

If you decide to change a type of a command, click an arrow ▾ and the drop-down menu will show the available commands.



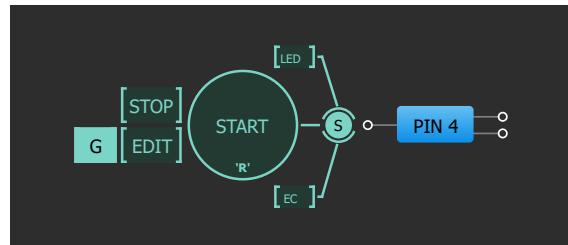
I2C command list

Composite commands *Read* and *Write* may be split into a sequence of basic commands (all commands are listed in the drop-down menu above).

Signal (Analog) Generator

The *Signal Generator* produces the analog signal, and can be configured using one of the following

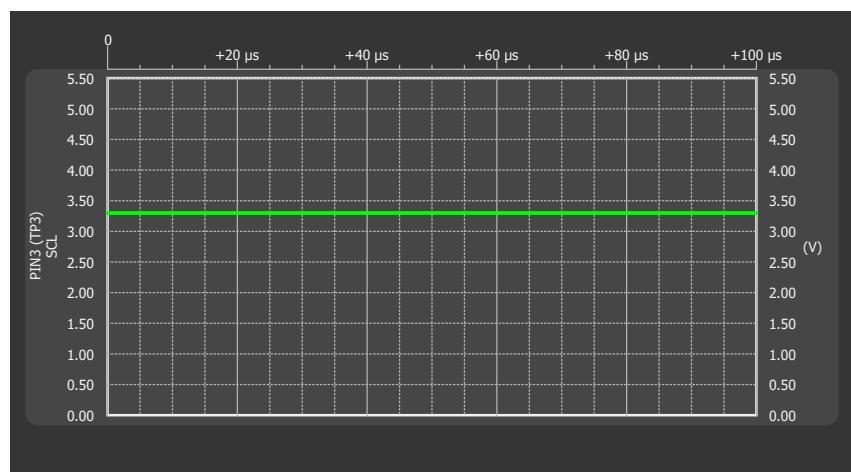
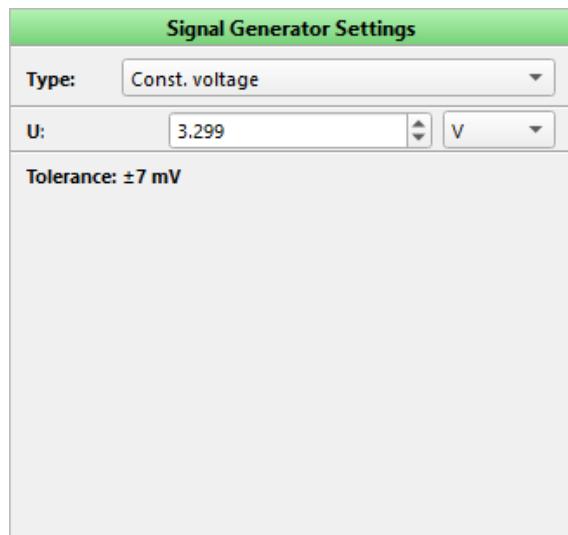
pre-defined types: constant voltage, sine, trapeze, logic pattern, or custom signal.



Signal Generator

Below, you can see the *Signal Wizard* view for different signal types.

- Constant voltage waveform type

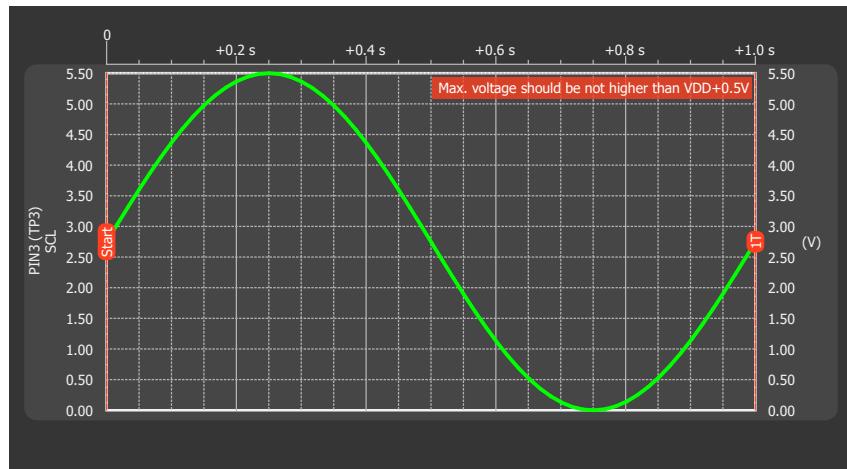


Configuration options of Signal Generator, type Constant Voltage

► Sine waveform type

Signal Generator Settings

Type:	Sine
Phase:	0
Custom phase:	0.00 rad
Amplitude:	2.751
Zero offset:	2.751
Period:	1000.000
Frequency:	1.00 Hz
Data:	Modify
Tolerance: ±7 mV	



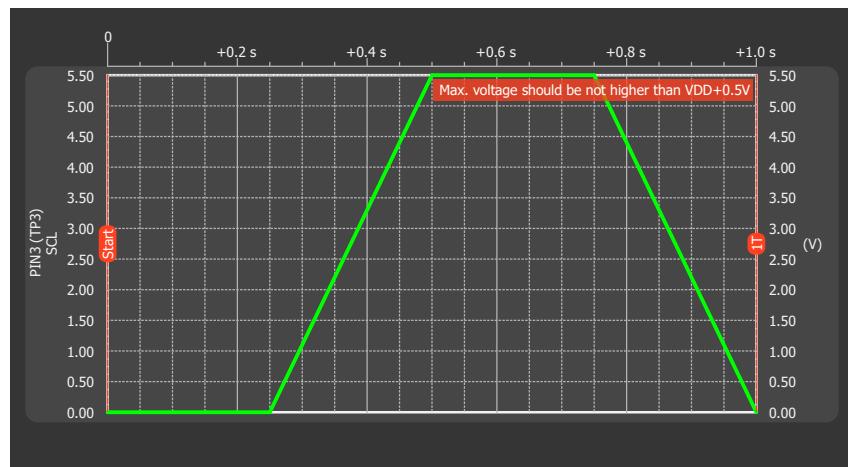
Configuration options of Signal Generator, type Sine

► Trapeze waveform type

Signal Generator Settings

Type:	Trapeze (Triangle, Saw)	
Mode:	Normal	
Umax:	5.501	V
Umin:	0.000	V
T low:	250.000	ms
T rising:	250.000	ms
T high:	250.000	ms
T falling:	250.000	ms

Tolerance: $\pm 7 \text{ mV}$

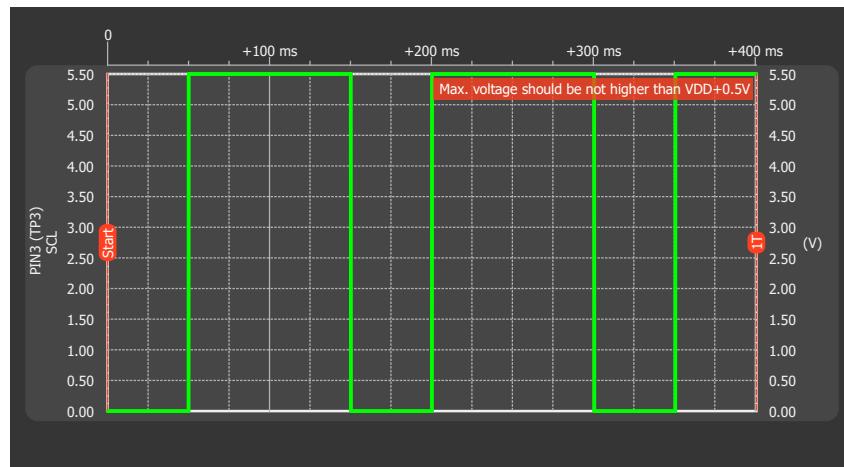


Configuration options of Signal Generator, type Trapezoid (Triangle, Sawtooth)

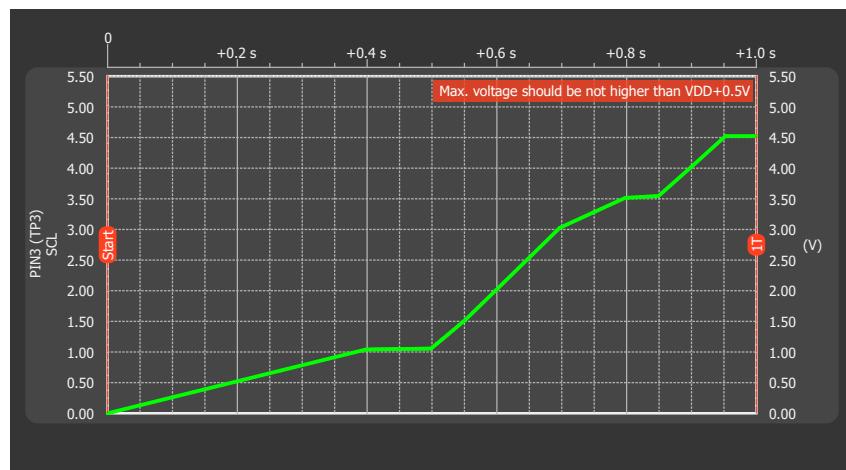
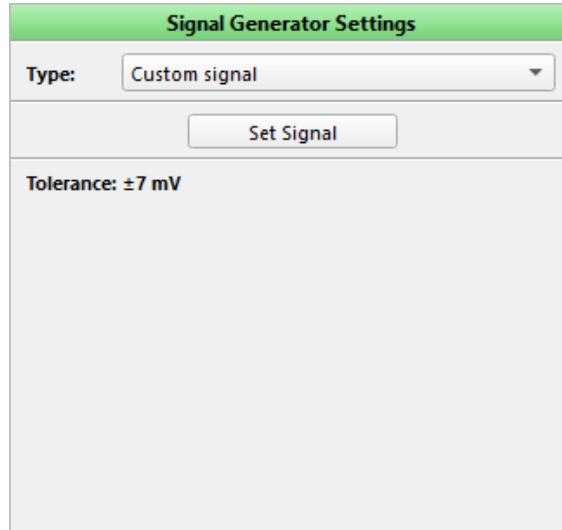
► Logic pattern waveform type

Signal Generator Settings

Type:	Logic pattern																											
Mode:	Normal																											
Levels adjustment:	Standard																											
Umax:	5.501 V																											
Umin:	0.000 V																											
Pattern:	01101101																											
<table border="1"> <thead> <tr> <th></th> <th>T</th> <th>U</th> </tr> </thead> <tbody> <tr><td>1</td><td>50.000 ms</td><td>Umin</td></tr> <tr><td>2</td><td>50.000 ms</td><td>Umax</td></tr> <tr><td>3</td><td>50.000 ms</td><td>Umax</td></tr> <tr><td>4</td><td>50.000 ms</td><td>Umin</td></tr> <tr><td>5</td><td>50.000 ms</td><td>Umax</td></tr> <tr><td>6</td><td>50.000 ms</td><td>Umax</td></tr> <tr><td>7</td><td>50.000 ms</td><td>Umin</td></tr> <tr><td>8</td><td>50.000 ms</td><td>Umax</td></tr> </tbody> </table>			T	U	1	50.000 ms	Umin	2	50.000 ms	Umax	3	50.000 ms	Umax	4	50.000 ms	Umin	5	50.000 ms	Umax	6	50.000 ms	Umax	7	50.000 ms	Umin	8	50.000 ms	Umax
	T	U																										
1	50.000 ms	Umin																										
2	50.000 ms	Umax																										
3	50.000 ms	Umax																										
4	50.000 ms	Umin																										
5	50.000 ms	Umax																										
6	50.000 ms	Umax																										
7	50.000 ms	Umin																										
8	50.000 ms	Umax																										
Insert	before row	1																										
Remove	row	1																										
Levels count:	8																											
Data:	Modify																											
Tolerance: ±7 mV																												



Configuration options for Signal Generator, type Logic pattern

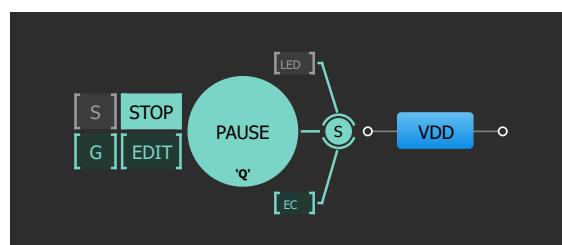


Configuration options of Custom Signal (arbitrary waveform) Generator

The custom signal can be configured in a separate window. Find out more in section [2.2.4 Custom Signal Wizard](#).

VDD/VDD2 Power Signal Generator

Some Part Numbers have an additional power supply (*VDD2*); if present, it allows to interface two independent voltage domains within the same design. You can configure the pins dedicated to each power supply as inputs, outputs, or both (controlled dynamically by the internal logic) to *VDD* and *VDD2* voltage domains. Using the available macrocell, you can implement mixed-signal functions bridging both domains or pass through level translation in *HIGH* to *LOW* and *LOW* to *HIGH* directions.

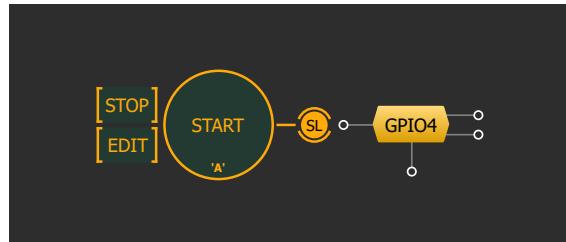


VDD/VDD2 Power Signal Generator

If the *Sync Power Rails [S]* mode is enabled in *Signal Wizard*, VDD and VDD2 share the same power settings. The option is available only for VDD / VDD2 power generators.

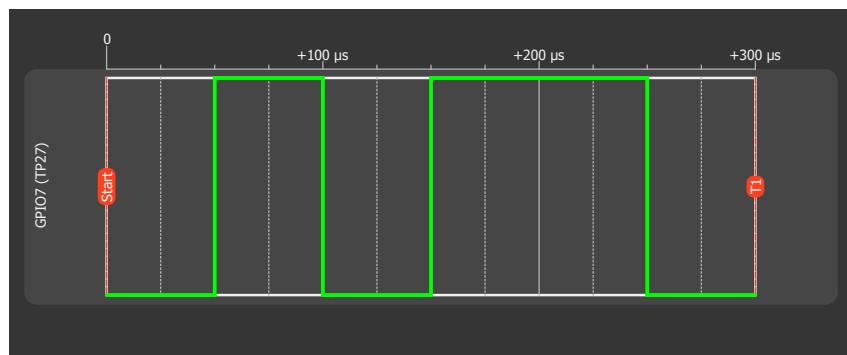
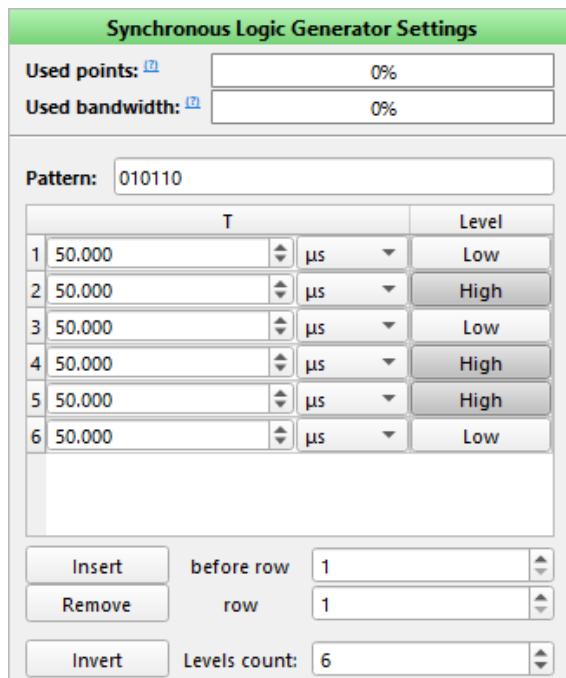
Synchronous Logic Generator

The *Synchronous Logic Generator* is used for generating the logic pulses and waveforms on GPIO pins. It is a 64-channel digital pattern Generator, provided only by *ForgeFPGA Advanced Development Platform*.



Synchronous Logic Generator

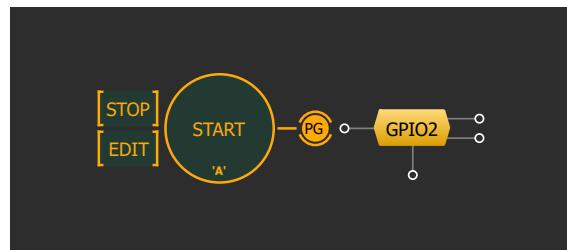
The settings of this Generator is also tuned by *Signal Wizard*:



Signal Wizard for Synchronous Logic Generator

Parametric Generator

The *Parametric Generator* generates logic pulses that follow different protocol sequences. This type is also available for *ForgeFPGA Advanced Development Platform*.

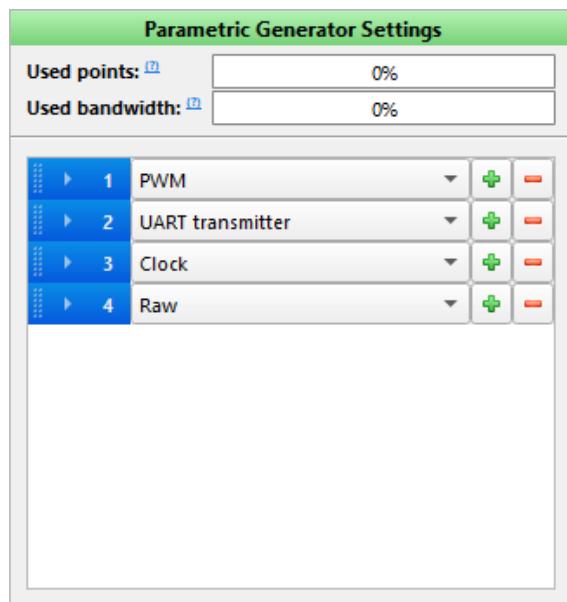


Parametric Generator

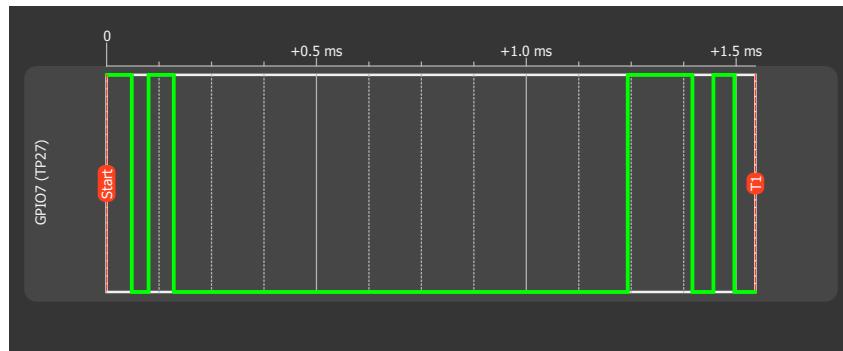
In *Signal Wizard*, a special editor shows the sequence of commands.

Actions available in the command editor:

- Add or remove commands by clicking and
- Change command parameters
- Change the order of commands by dragging the command to another position



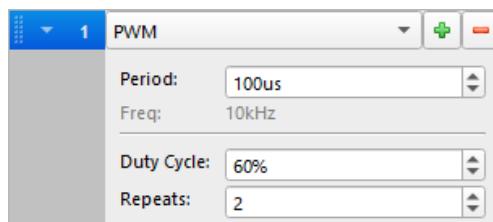
Parametric Generator command editor



Signal Wizard for the Parametric Generator

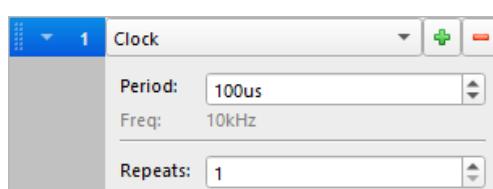
The list of available commands:

- *PWM* (Pulse Width Modulation) — the *PWM command editor* has three input fields:
 - *Period* — a united duration of high and low states per repeat
 - *Duty cycle* — the percentage of the total duration in the high state
 - *Repeats* — pattern repeat count



PWM command editor

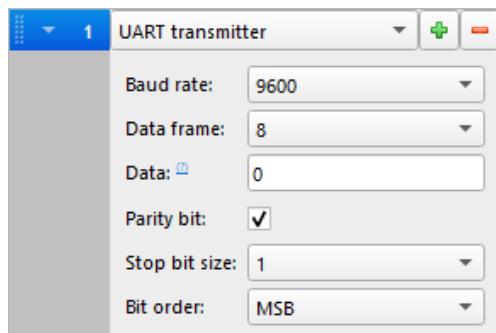
- *Clock* — the command generates a signal oscillating between a high and a low state. The editor has two input fields:
 - *Period* — the united duration of high and low states per repeat
 - *Repeats* — pattern repeat count



Clock command editor

- *UART Transmitter* — generates signal according to the *UART* standard:
 - *Baud rate* — signal's frequency
 - *Data frame* — number of bits allocated for user input

- *Data* — user input in hex format. In case the data frame is lower than the bits required to represent data, more significant bits are ignored
- *Parity bit* — create parity bit for error detection
- *Stop bit size* — duration of the stop bit
- *Bit order* — serial data transfer format

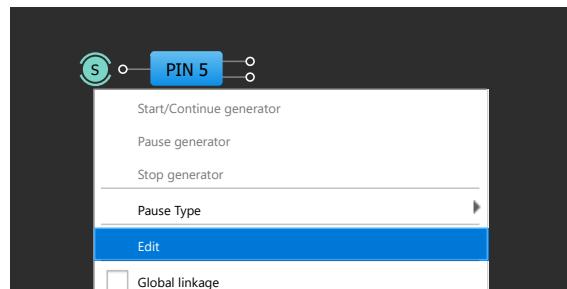


UART transmitter command editor

- *Raw* — this pattern works as a typical *Logic Generator*

2.2.3 Signal Wizard

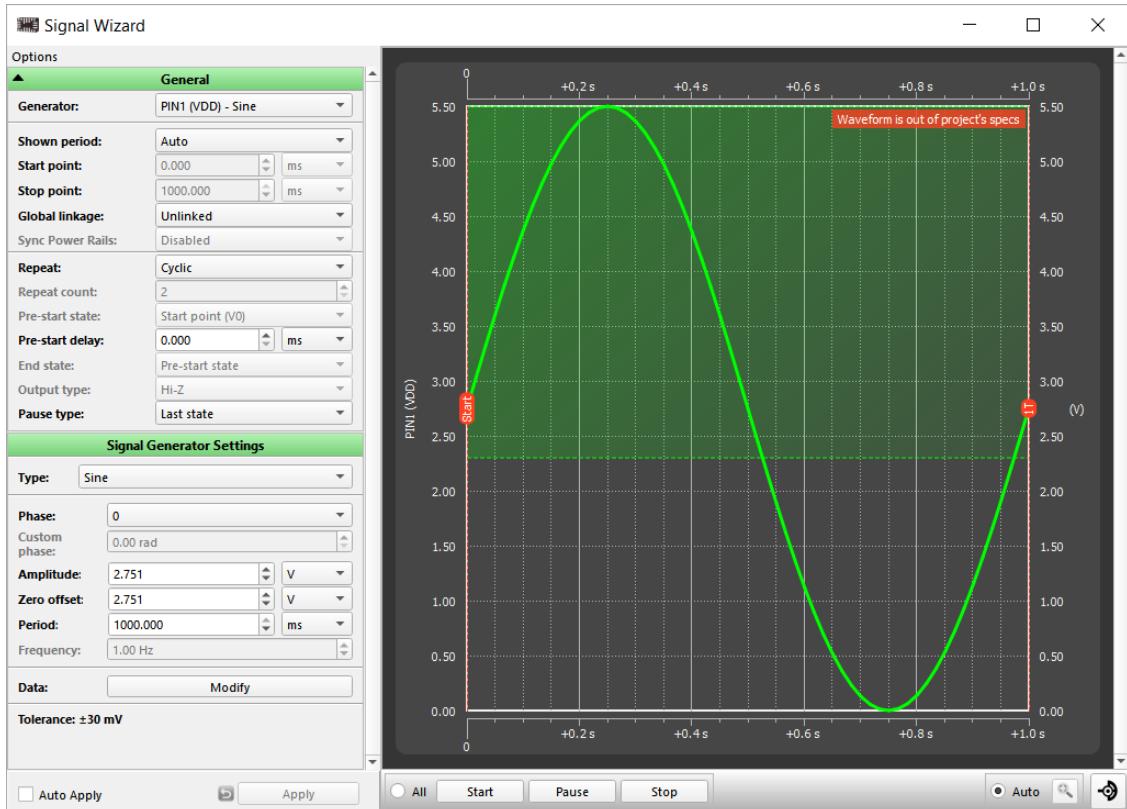
To start configuring a *Generator*, double-click its icon next to the pin or find *Edit* in its context menu.



Opening Signal Wizard

The *Signal Wizard* window contains a *Generator* settings panel and the plot (waveform preview). The window is common for all *Generators*, though different set of settings may be activated, depending on

the selected hardware source.



Signal Wizard window

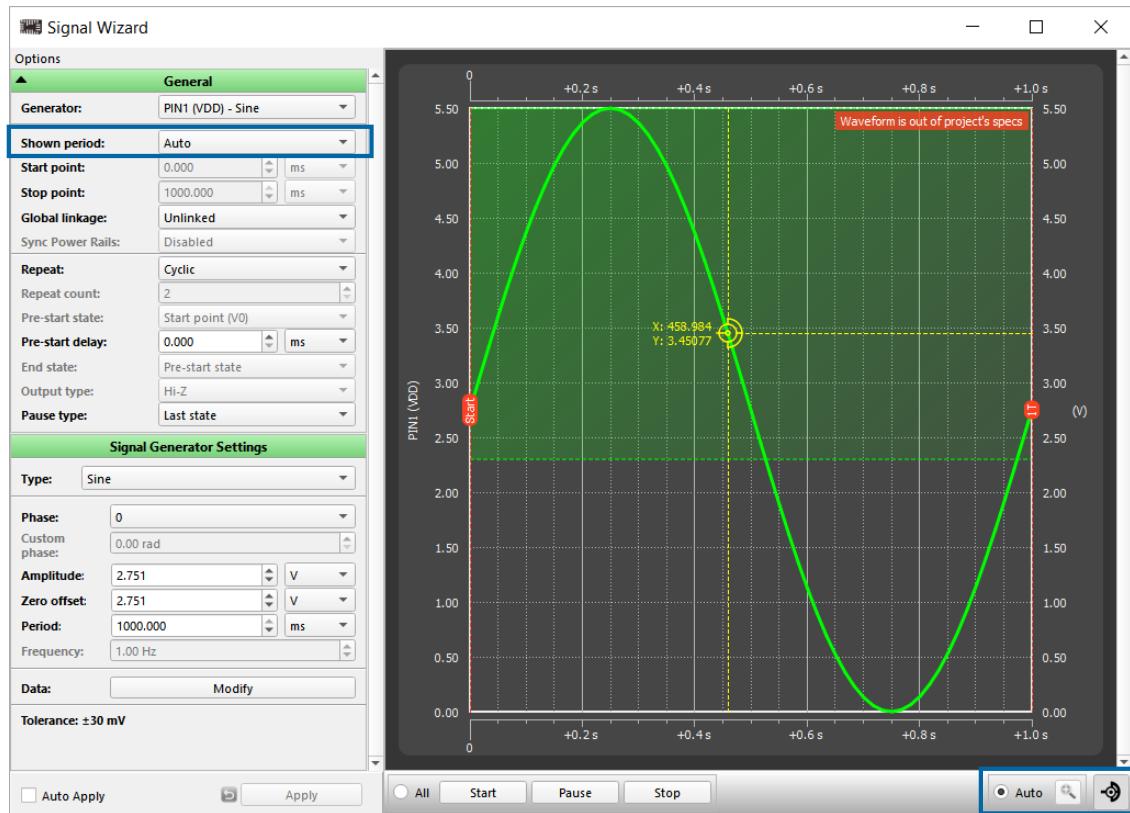
General settings category mostly provides signal configurations related to time, period, and state. Below you can see the settings which may require more detailed description.

- **Global Linkage** — when enabled, provides possibility to control Generator using Start/Stop/Pause buttons on the *Debugging Controls* panel
- **Sync Power Rails** — enable for VDD and VDD2 to share the same power settings. The option is available only for *VDD / VDD2 Power Generators*

The lower settings part is generator-specific. Read more in section [2.2.2 Generators](#).

Scaling controls

Scaling controls are located at the bottom right corner of the *Signal Wizard*. The controls let you adjust the number of periods shown in the waveform preview.



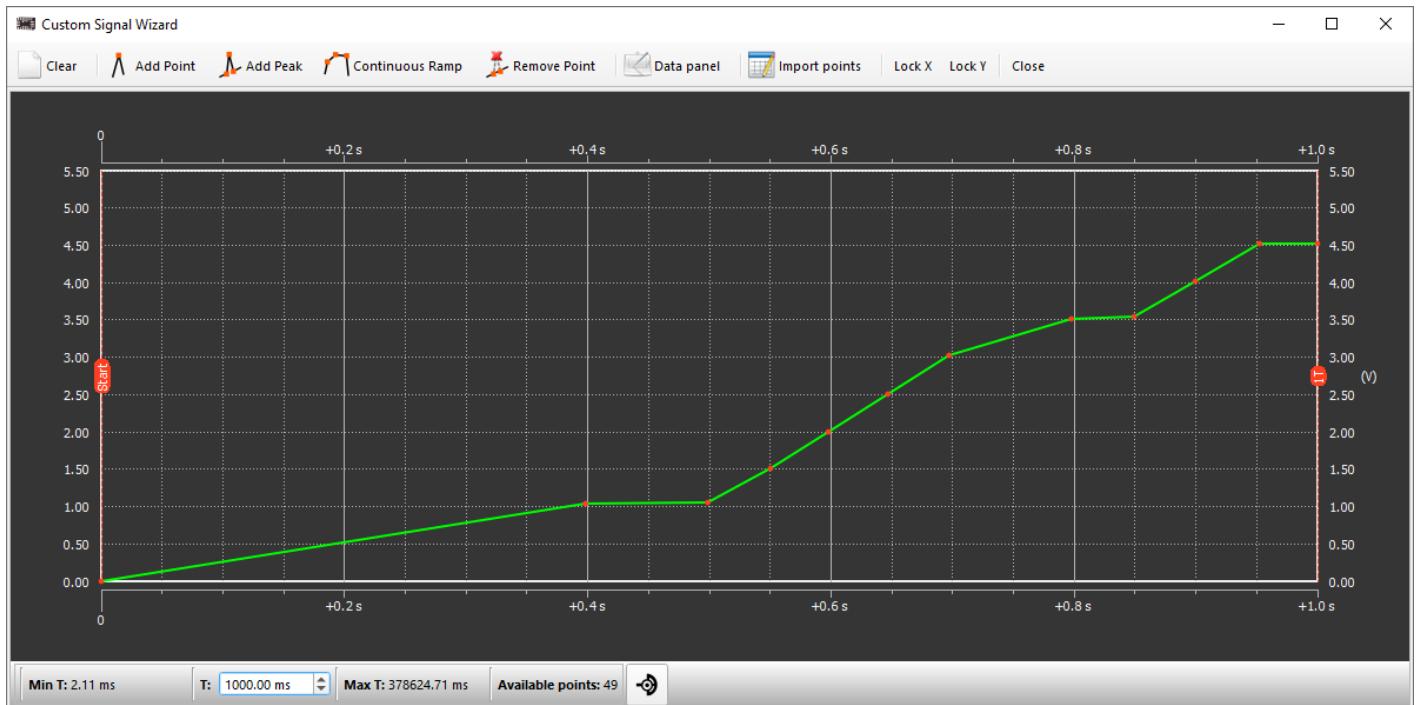
Scaling in Signal Wizard

You can use the *Cursor* button to turn the mouse coordinates on/off in the timing diagrams.

- **Auto** — scale all generators to fit the biggest period among them. Only generators with *Shown period* set to *Auto* are affected.
- **Custom** — waveforms can be re-scaled manually. To do that, ensure that the *Auto* mode is OFF and then scale by *Ctrl* + mouse wheel.

2.2.4 Custom Signal Wizard

Custom Signal Wizard allows creating, importing, and editing the signal waveforms (applicable only for [Signal \(Analog\) Generators](#)). The signal can be created by manually adding points or importing a custom list of points from an external source.

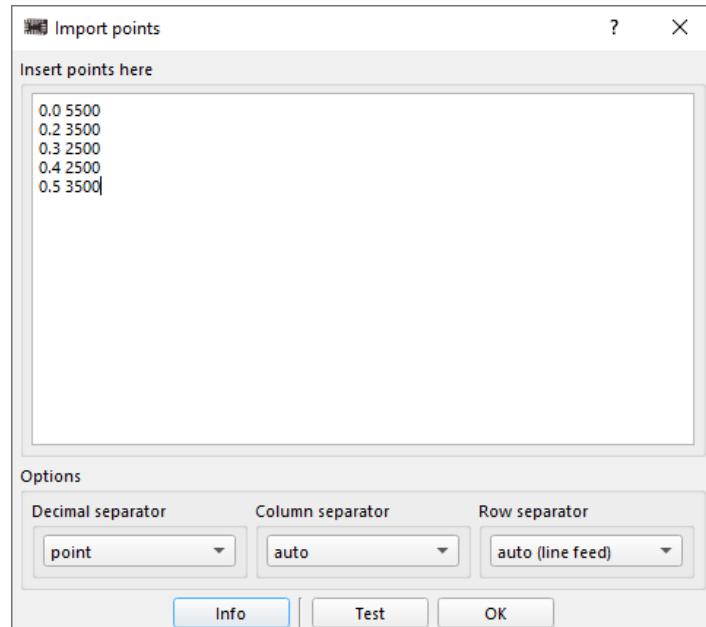


Drawing Signal (Arbitrary waveform)

Use the controls on the toolbar to manipulate the waveform.

- *Add Point/Peak/Continuous Ramp* — use the buttons to start creating the signal waveform
- *Remove Point* — click the button to remove a selected point (or use a right-click)
- *Data Panel* — show or hide the data table. You can remove/change values for a selected point or add a point in between

► *Import Points* — click the widget to open the *Import* window and insert the points' coordinates



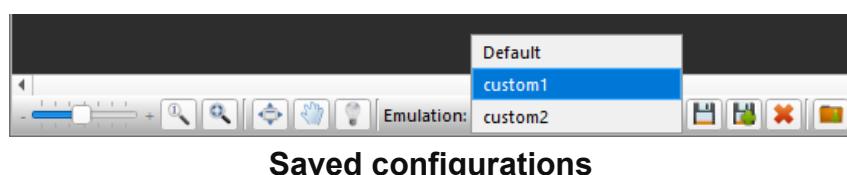
Import Points

See the coordinates formatting options:

- *Decimal separator*: point/comma
- *Column separator*: auto/tab/other
- *Row separator*: auto(line feed)/tab/other

2.2.5 Hardware configurations

You can create configuration state snapshots to restore a previous configuration when needed. The snapshots include your hardware sources' configuration and are saved and loaded with your project file. This option is available for both *Generators* and *basic Hardware Sources*.



Saved configurations



Save the current configuration state



Save a new configuration state



Delete the selected configuration



Import new configuration

Default
custom1
custom2

The list of saved configurations

2.2.6 Debugging Controls

To access the required development platform through the software, click *Debug* and select the board from the list of supported platforms.

The *Debugging Controls* panel appears. It contains the UI controls for chip communication and programming, brief information about the connected devices, and other features described below.

Later in this section, you can read about all *Debugging Controls* panel UI elements you may encounter while working with different platforms. In chapter [3 Devices](#), you can see the supported development platforms description and the *Debugging Controls* panel for each board.

The table listing *Debugging Controls* elements availability for all boards is given in the appendix, in section [5.3 Debugging Controls feature availability](#).

Click the reference below to quickly find the *Debugging Controls* panel interface for the certain platform.

[GreenPAK Advanced Dev. Platform](#)

[ForgeFPGA Evaluation Board](#)

[GreenPAK DIP Dev. Platform](#)

[PowerPAK Dev. Platform](#)

[GreenPAK Lite Dev. Platform](#)

[PowerPAK Demo Board](#)

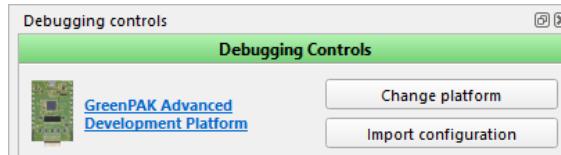
[GreenPAK Serial Debugger](#)

[GreenPAK Serial Debugger \(with SLG5100x\)](#)

[ForgeFPGA Advanced Dev. Platform](#)

Debug configuration

- Recommended platform configuration — find information about the supported adapter, development board and the devices' ordering information. Click on the platform name link to open the *Recommended platform configuration*



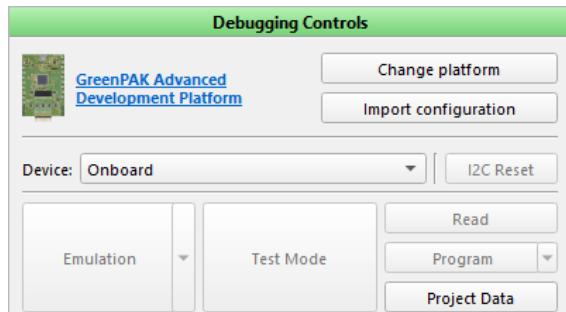
Recommended platform configuration

- Change platform — open the list of the development platforms that are supported by a Part Number
- Import configuration — upload configurations from a different development platform supported by a Part Number

Device selector

- Onboard — search the device placed on the board

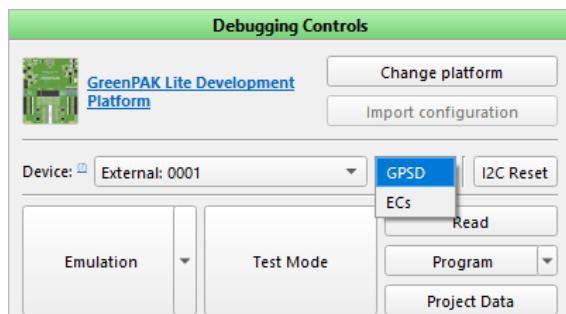
- *External* — search the device on the selected I2C slave address



Device selector

External device modes

- *GPSD* — use a separate 4-pin connector (PWR, SCL, SDA, GND) for I2C communication
- *ECs* — use the expansion connector for I2C communication:
 - *VDD EC* — power
 - *SCL/SDA* — EC according to the chip TP map



Lite board device selector

Chip procedures

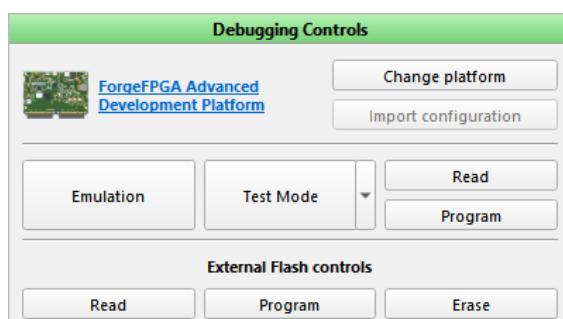
- *Test Mode* — debug the programmed project. Enable power and load the configured hardware sources
- *Emulation* — debug the current project. Enable power and load the design with configured hardware sources
- *Emulation(sync)* — the project changes are automatically loaded to the chip when the control is active
- *Sync* — load the current project to the chip once
- *Read* — read the programmed chip and open the project in the new software instance or in the *Project Data* window of the current instance

- *Program* — program the chip with the current project. For some Part Numbers, e.g. SLG47004, EEPROM programming is available
- *I2C Reset* — change the I2C reset bit (from 1 to 0). This causes the device to re-enable the Power-On Reset (POR) sequence, including the reload of all registers data from NVM

Flash procedures

Flash memory is located on the FPGA sockets. See the available controls to work with it.

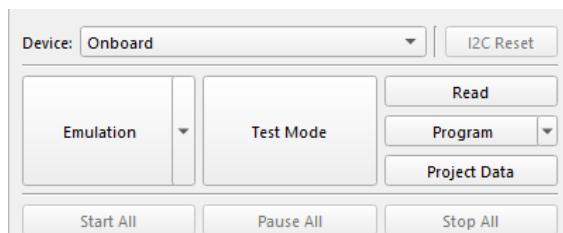
- *Test Mode(*)* — load data from flash to the chip
- *Read* — read the chip project from the flash memory
- *Program* — program a flash memory with the current project
- *Erase* — clean flash memory (erased flash memory address values will be read as 0xFF)



Flash procedures

Project data window

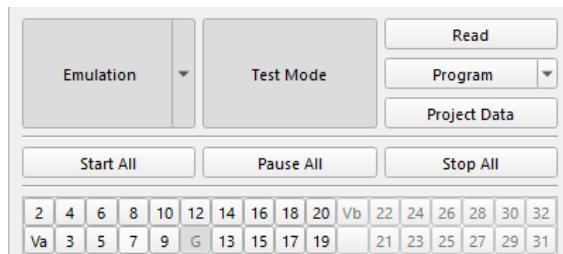
- *Project data* — a table with NVM/EEPROM bit sequences. You can change the bit values, import/export the sequences and use data to program the chip (for more info see section 2.2.10 Project data window)



Project data control

Generator controls

- *Start/Pause/Stop All* — control the generator state while *Global linkage* generator setting is enabled



Generator controls

Expansion Connectors (EC)

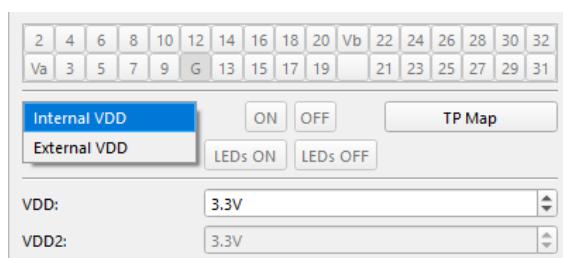
- *Expansion Connectors* — this port was designed to connect the platform to the external circuits and apply external power, signal sources, and loads. There are several ways to connect/disconnect the expansion connectors (also, control the ECs on the work area via a hardware source):

- Connect/Disconnect all ECs by clicking ON/OFF buttons
- Connect/Disconnect a specific EC by clicking the ECs sequence number button
- Connect/Disconnect *Va* EC by clicking the *Ext. VDD* button



Power source selector

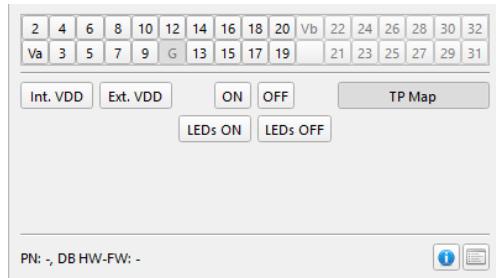
- *Internal VDD* — the power is provided by the GreenPAK board
- *External VDD* — the board measures voltage on the active power connector, and provides the same voltage level



Power source selector example

TP map

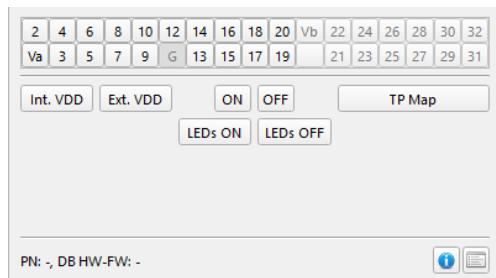
- *TP map* — show the test point map on the work area to reflect the physical test points on the development platform



Test point control

LEDs ON/LEDs OFF

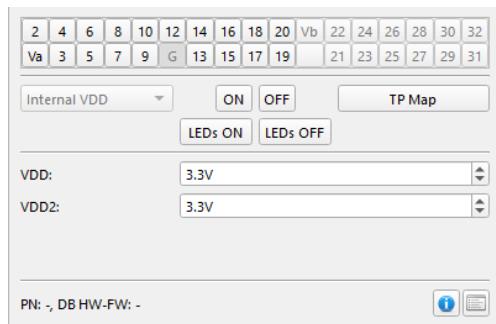
- *LEDs ON/LEDs OFF* — enable/disable all LEDs on the current platform (applicable only for development platforms with LED support). You can control a particular LED on the work area via a [hardware source](#)



LEDs ON/OFF controls

Voltage level controls

- *VDD* — set the voltage level on the corresponding test points. If a selected value exceeds any of the board limitations, the dependent controls are blocked, and a warning is shown.

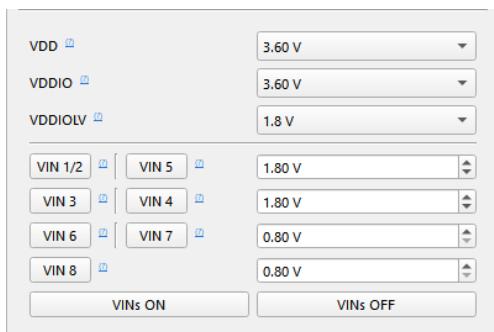


Voltage level controls

PowerPAK voltage controls

- *VDD/VDDIO/VDDLV* — power supply voltage for overall chip and GPIOs

- *VIN* — power supply voltage on the corresponding LDO component's VINs
- *VINs ON/OFF* — enable/disable all VINs

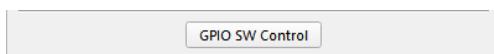


PowerPAK voltage controls

GPIO SW control

- *GPIO SW Control* — add buttons for software control of GPIOs, GPIO(SDA), GPI(SCL) and CS. Read more in section [2.2.1 Hardware sources](#)

Note: the *GPIO SW Control* feature is permanently enabled in the *PowerPAK Development Platform*. For the *SLG51002CTR Demo Board*, you can activate the feature using the corresponding button in the *Debugging Controls*.



GPIO software control

Info details

Brief information about the connected chip and development platform (e.g., Part Number, firmware ID, etc.).



Info details

- ⓘ Information about the connected Part Number, development platform, software version, and operating system
- ⌂ Show operation log

Board selector

Once the board is connected, the platform info appears on the bottom toolbar. To switch to a different platform, use the Board selector.

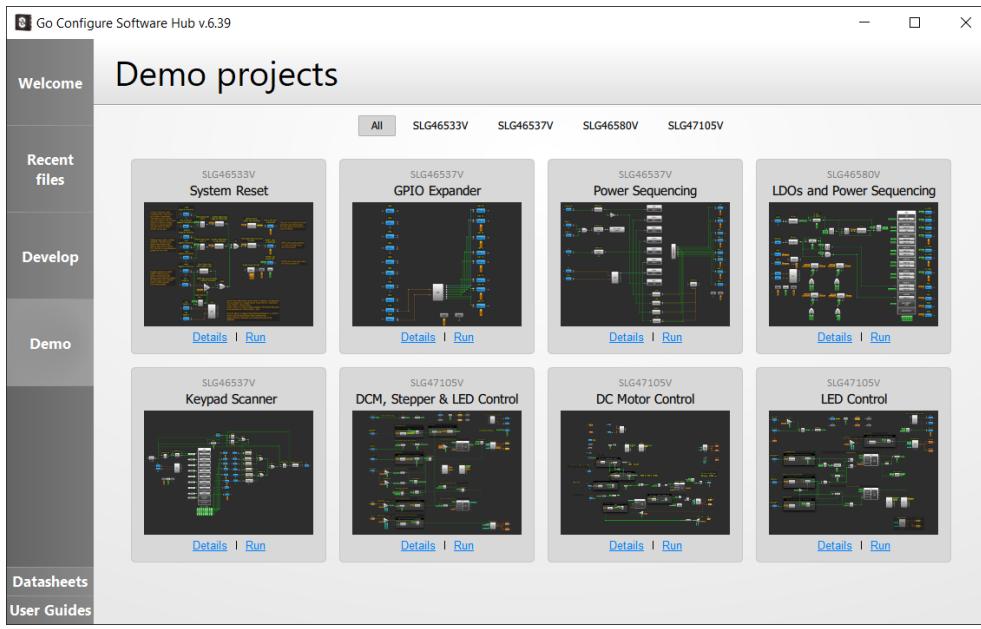


Board selector

- Blink* Blink with an LED of the board in use
-  Update the chip and board *Info details*
-  Test if the socket is successfully connected to the platform

2.2.7 Demo Board and Demo Mode

The *Demo Mode* allows exploring possible applications for a specific Part Number. You can connect a *Demo Board*, open a demo project for it, and experiment with a design. To start the *Demo Mode* click the *Demo* tab in the *Hub* window, select the design you are interested in and click *Run*:



Select a specific project

- *Details* — technical information about the selected project and its configurations
- *Run* — open the desired demo project

Once you open the project, the workspace UI depends on whether the *Demo Board* is connected.



Waiting for the corresponding Demo Board connection

A *Demo Board* is a piece of hardware designed to demonstrate and debug the selected project. It has an IC with a pre-programmed project soldered onto the board. After the board detection is successful, it is possible to manipulate the chip's NVM and use *I2C Tools*.



Demo Board detected

- *Write* — load the current project NVM sequence to the device

➤ *I2C Tools* — the following *I2C Tools* are available for *Demo Mode*:

- *I2C Virtual Inputs*
- *I2C Virtual Outputs*
- *I2C Reconfigurator*

For more information about *I2C Tools* see section [2.2.8 I2C Tools](#)

➤  — information about *Part Number*, development platform and operating system

➤ *Close* — exit demo mode

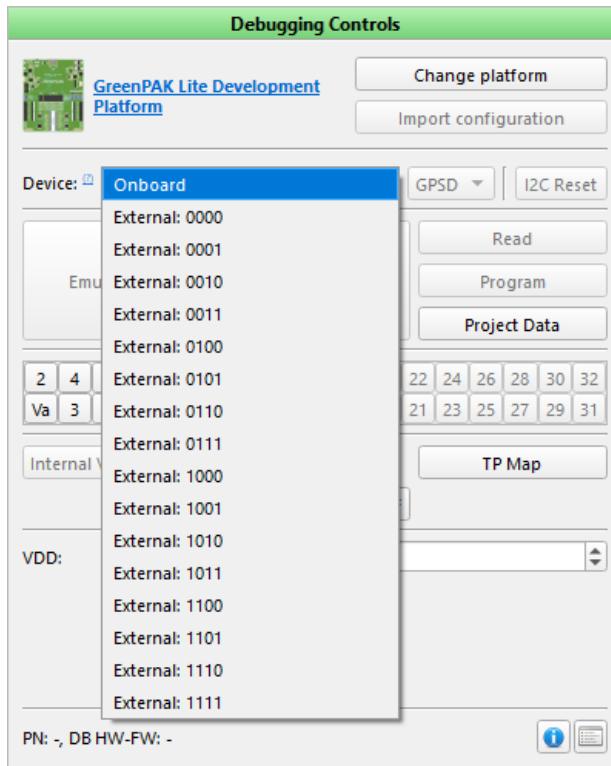
Note: *Demo Mode* applies some limitations on specific features (operations with project files, *Debug Tool*, *Simulation*). Exiting the *Demo Mode* will remove all limitations, yet keep the current project open.

2.2.8 I2C Tools

The *I2C Tools* are the instruments that help to examine the project configuration by reading or writing the register data on the chip.

A chip's *I2C Serial Communication Macrocell* (*I2C*) allows an *I2C bus master* to *read* and *write* information at any moment via a serial channel directly to the registers via *I2C protocol*. The tool allows configuring the macrocell data on the fly.

You can choose any external *device address* and work with an external chip using the *Device selector* in the [Debugging Controls](#) section. You can select the *Onboard* chip, or any of 16 *GreenPAK I2C slave devices*.



Selecting device

The *I2C tools* are available when the *Debug* utility for a hardware board is enabled. To start *I2C Tools*, *Emulation* or *Test Mode* is required.

I2C Tools are comprised of three sections, *I2C Virtual Inputs*, *I2C Virtual Outputs*, and *I2C Reconfigurator*. The tools can be launched from the *I2C toolbar*.



I2C Tools at top toolbar

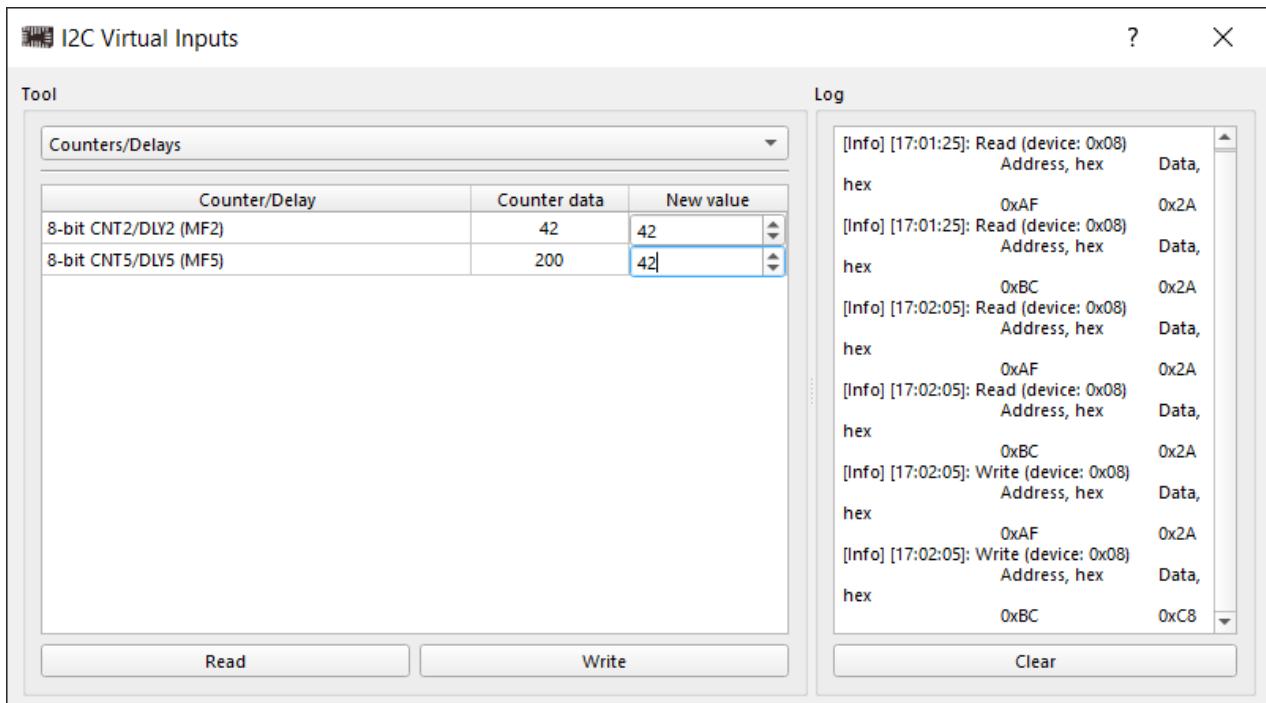
I2C Virtual Inputs

By definition, the *I2C Virtual Inputs* tool provides the possibility to interact with the I2C input data, that is the configuration data of a component.

Counters/Delays

The *Counters/Delays* option shows the *Counter/Delay* data. Click *Read* to actualize the data. A *Counter/Delay* is included in the current list only if the component is in the CNT/DLY mode on the work area. If you need to change the data, input your value in the corresponding field and click *Write*.

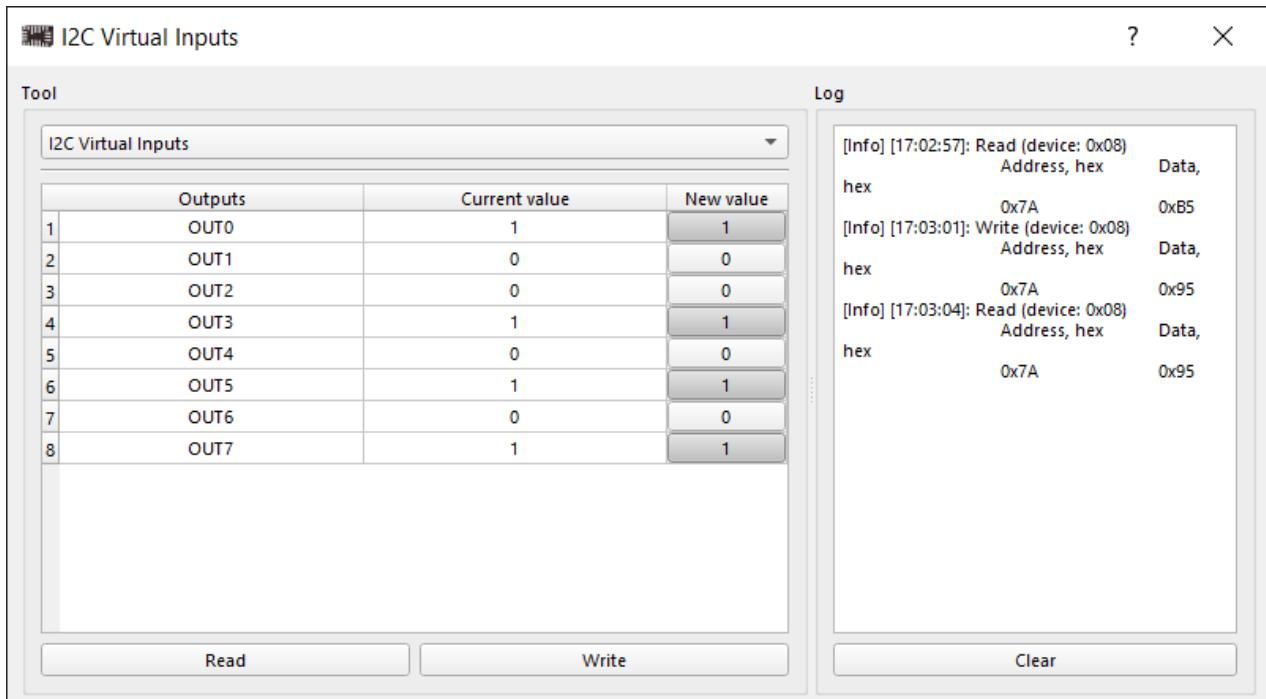
This will substitute the current value on the chip with a new one.



Managing Counters/Delays

I2C Virtual Inputs

The *I2C Virtual Inputs* option allows you to *read* and *write* the value of all available virtual inputs via *I2C*.



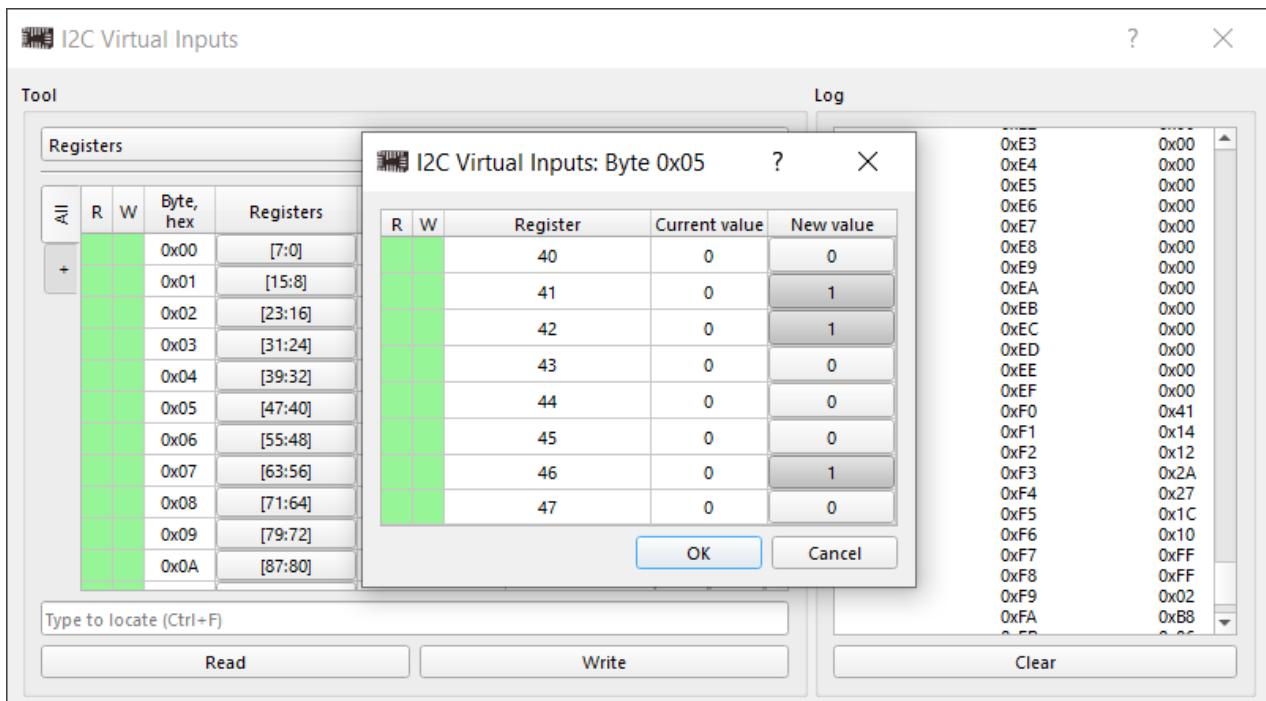
I2C Virtual Inputs

Registers

The *Registers* page shows entire chip NVM. The columns at the left, *R* and *W* show if a register is readable and writable, respectively.

- I2C operations are supported
- I2C operations are not supported
- I2C operations are unsupported for some bits of the register

You can edit each bit selectively, or input the whole register value as a number. To see the bits of a register, click a button in the *Registers* column for the required register.



Changing registers by bits

The context menu of a register allows switching the current value to hex or decimal format.

You can read/write the entire chip data by using the bottom *Read/Write* buttons. To update a particular register click the *Read/Write* buttons in the respective row.

You can find a required byte or register by typing its number in the search field. To filter out bytes or registers from the search, use the following markers:

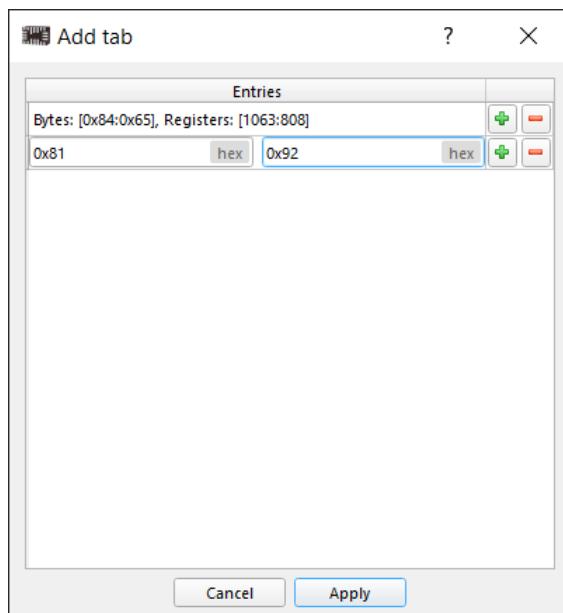
- b — searching for bytes only

➤ r — searching for registers only

All	R	W	Byte, hex	Registers	Current value, hex	New value, hex	Read	Write
			0xBD	[1519:1512]	0x00	0x00	Read	Write
			0xBE	[1527:1520]	0x00	0x00	Read	Write
			0xBF	[1535:1528]	0x02	0x02	Read	Write
			0xC0	[1543:1536]	0x00	0x00	Read	Write
			0xC1	[1551:1544]	0x01	0x01	Read	Write
			0xC2	[1559:1552]	0x00	0x00	Read	Write
			0xC3	[1567:1560]	0x00	0x00	Read	Write
			0xC4	[1575:1568]	0x02	0x02	Read	Write
			0xC5	[1583:1576]	0x00	0x00	Read	Write
			0xC6	[1591:1584]	0x01	0x01	Read	Write
			0xC7	[1599:1592]	0x00	0x00	Read	Write

Searching for a register, filtering out bytes

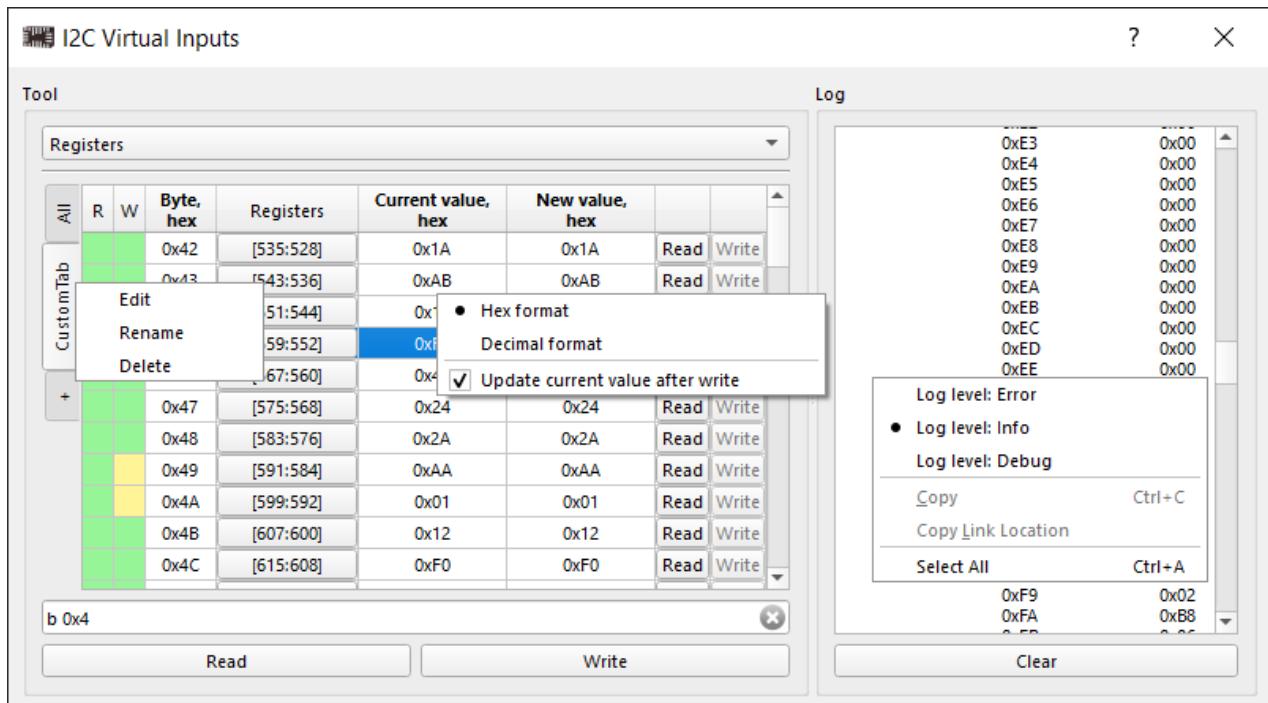
The *Registers* page allows making tabs with specific register ranges. Click the and add entries with register ranges:



Add tab

Tabs can be edited, renamed, or removed. To do that, right-click a register range tab and then click

Edit, Rename or Delete in the context menu. The custom tabs are saved to the project file.



Context menus

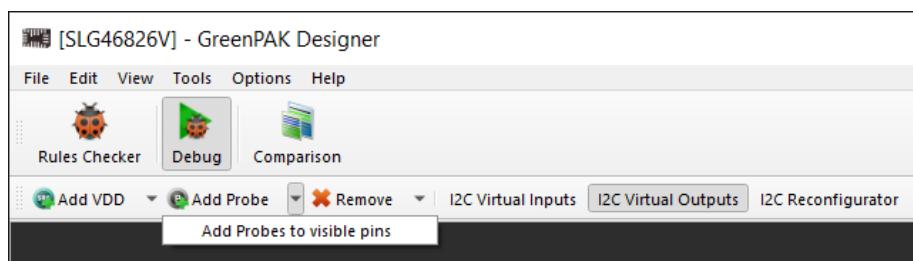
Log

The *Log* section lists the results of the recent *read* and *write* operations. The *Log level* has three severity settings, with *Debug* being the most detailed, and *Error* being the most permissive, showing critical information only. You can change the severity in the context menu. The actual log message type is indicated in a log record itself.

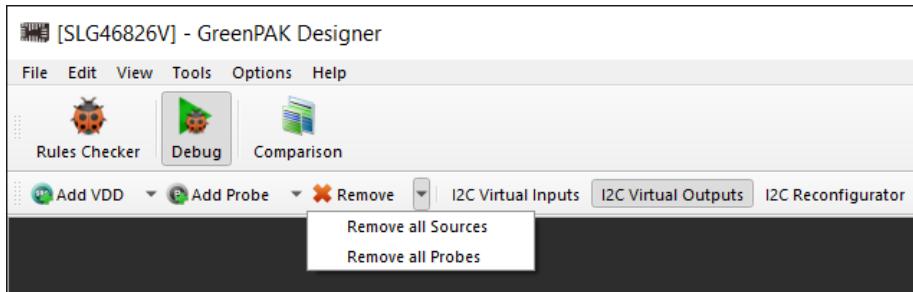
I2C Virtual Outputs

I2C Virtual Outputs is a built-in feature that enables reading the state of the chip components at any given moment. The tool reflects data from the outputs of the chip macrocells. The *I2C Virtual Outputs* tool also includes probes, ASM state, and the counted data table.

Probes (*Matrix inputs*) reflect the current level on the component output pins. You can add probes to pins by clicking *Add Probe* and then manually clicking the required pins. Internal pins that support adding a probe are highlighted in green. Also, you can add probes to all visible pins at once by clicking *Add Probes to visible pins*.



You can remove probes one by one, remove all sources, or remove all probes at once as well.



Counted Data is the current value of the CNT/DLY components read from the chip project (see the Datasheet for the selected Part Number). To refresh the data click the respective button on the tool panel.

I2C Virtual Outputs		
I2C Virtual Outputs		
Refresh Last Update: 14:03:53		
Macrocell	Property	Value
3-bit LUT10/8-bit CNT4/DLY4	Counted Data	5
3-bit LUT8/8-bit CNT2/DLY2	Counted Data	0
RTC	Counted Data	0

I2C Virtual Outputs tool

I2C Reconfigurator

I2C Reconfigurator allows changing chip data dynamically by sending NVM snapshots to the chip. A snapshot is a current state of the project, which includes macrocell configurations and connections. You can configure a list of snapshots and send them one by one or all at once. Also, you can add delays between snapshots. It is also possible to edit the snapshot name and change the delay time.

I2C Reconfigurator		
I2C Reconfigurator		
On	Name	Value
<input checked="" type="radio"/>	Snapshot @16:45:35	
<input checked="" type="radio"/>	Delay	1000 ms
<input checked="" type="radio"/>	Snapshot @16:52:01	
<input checked="" type="radio"/>	Snapshot @16:58:15	
<input checked="" type="radio"/>	Delay	5000 ms
<input checked="" type="radio"/>	Snapshot @17:03:25	
<input checked="" type="radio"/>	Snapshot @17:12:31	<input type="button" value="Load"/> <input type="button" value="Overwrite"/>

Snapshot configuration

Scenario can be edited from the toolbar:

-  add a snapshot of the work area to the list
-  add a delay between snapshots to the list
-  move the selected list item one level up
-  move the selected list item one level down
-  remove the selected list item

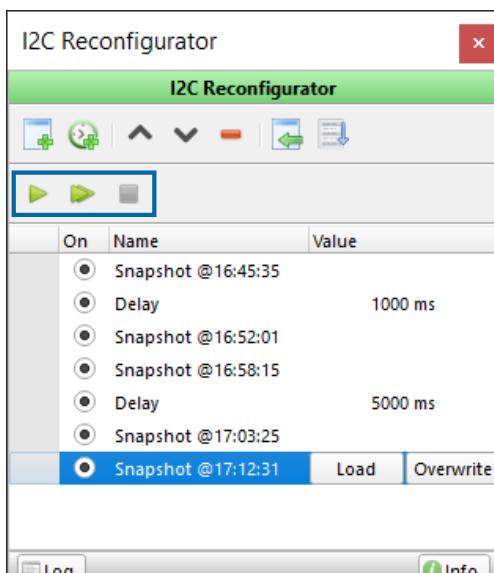
The *Reconfiguration scenario* contains the exact list of settings applied. If you want to see the list of actual changes as they are applied, refer to *Reconfiguration scenario*.

-  open a dialog with the list of changes applied; delays are mentioned. You can also export the whole list to a file.
-  Import presets from a previously configured project.

Snapshots are diff-based lists of changes. The first snapshot substitutes any configuration currently present on the chip; further snapshots are processed compared to the preceding snapshot.

You can load any snapshot to the list apart from walking through the whole scenario. To do that, click *Load* at the corresponding snapshot row. If you need to apply changes to an existing snapshot, prepare the state of the chip and click *Overwrite*. It will not, however, respect the former state of a snapshot but substitute the snapshot with the current state.

Use snapshot sending controls to get the required debug flow:



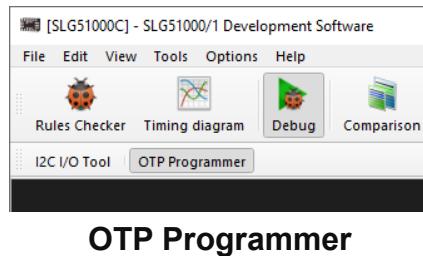
Snapshot operation

- ▶ send the next snapshot to the chip and pause list execution; if the list doesn't have an active pointer, the first snapshot will be sent
- ▶ send snapshots one by one continuously; delays are respected
- stop sending snapshots, reset the list pointer, and restart the list

There is a diff-based *Log* utility of actual changes applied to the chip. As delays don't imply any changes to the chip, they do not appear in the *Log*. The *Log* can be rolled up, down, and cleaned with the corresponding buttons.

2.2.9 OTP Programmer

OTP Programmer allows you to read *OTP* (*One-Time Programmable*) memory from a chip, make changes, and program the new data. The tool shows differences and conflicts between the programmed chip project and current project data.



OTP Programmer

To reach the *OTP Programmer* tool, click on *Debug* button and select suitable development platform.

The screenshot shows the OTP Programmer window with a table of OTP data. The columns are Name, OTP / Register address, Chip OTP Data, hex, and Data to Program, hex. The data table contains the following rows:

Name	OTP / Register address	Chip OTP Data, hex	Data to Program, hex
PWRSEQ_RESOURCE_EN_3	0x70 / 0x1903	0x00	0x00
PWRSEQ_RESOURCE_EN_4	0x71 / 0x1904	0x00	0x00
PWRSEQ_RESOURCE_EN_5	0x72 / 0x1905	0x00	0x00
PWRSEQ_SLOT_TIME_MIN_UP0	0x73 / 0x1906	0x00	0x00
PWRSEQ_SLOT_TIME_MIN_DOWN0	0x74 / 0x1907	0x00	0x00
PWRSEQ_SLOT_TIME_MIN_UP1	0x75 / 0x1908	0x00	0x00
PWRSEQ_SLOT_TIME_MIN_DOWN1	0x76 / 0x1909	0x00	0x00

Below the table are buttons for Next diff, Next conflict, and a Legend box. At the bottom are three tabs: Load Table, Transfer Table, and Program Device, each with its own set of buttons: Read OTP, Load from project, Open in new software instance, Save to file, and Program OTP.

OTP Programmer

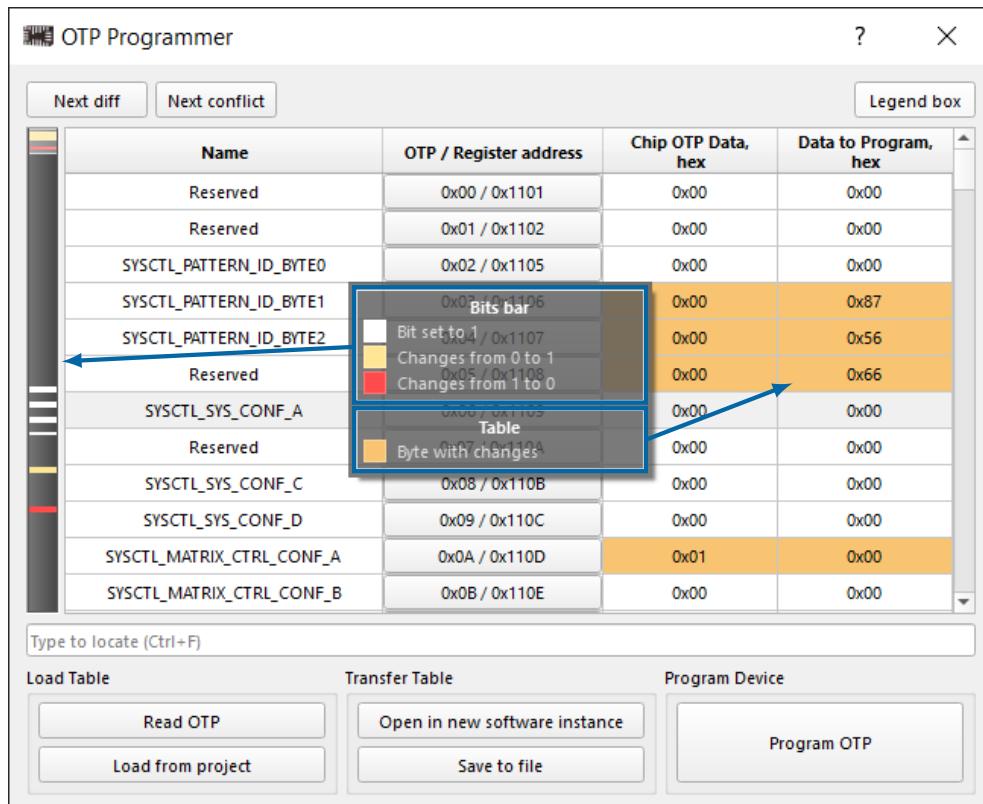
Before you start working with *OTP Programmer*, check whether the chip is detected (platform and chip info should appear in the *Info details* on the *Debugging Controls* panel).

The tool contains the following controls:

- *Read OTP* — read the data from the chip and load/set it into the *Chip OTP Data* column
- *Load from project* — load the data from the current project into the *Data to Program* column
- *Open in new software instance* — open the *Data to Program* in the new software instance

- Save to file — save the *Data to Program* into a text file
- Program OTP — program the chip with the *Data to Program* (**Note:** It is possible to program bits from 0 to 1, but not vice versa)

Open in new software instance, Save to file, and Program OTP buttons become available after using Read OTP or Load from project buttons.



Legend Box

- Legend Box — shows the color scheme of Bits changes in Bits bar
- Next diff — switch focus to the next Byte with changes
- Next Conflict — switch focus to the next Byte with changes in which the value was changed from 1 to 0

- Search field — find a required byte or register

The screenshot shows the OTP Programmer software interface. A modal dialog box titled "Byte view: 0x25 / 0x1507" is displayed in the center. This dialog contains a table with columns: Bit, Chip OTP Data, and Data to Program. The table has 8 rows, each corresponding to a bit from 0 to 7. The "Data to Program" column for bit 0 is currently selected, indicated by a blue border around the cell containing the value "0". Below the table are "OK" and "Cancel" buttons. In the background, the main program window shows a table with four columns: Name, OTP / Register address, Chip OTP Data, hex, and Data to Program, hex. The row for bit 0 (0x25 / 0x1507) is also highlighted with a blue border. The main window also includes sections for Load Table, Transfer Table, and Program Device.

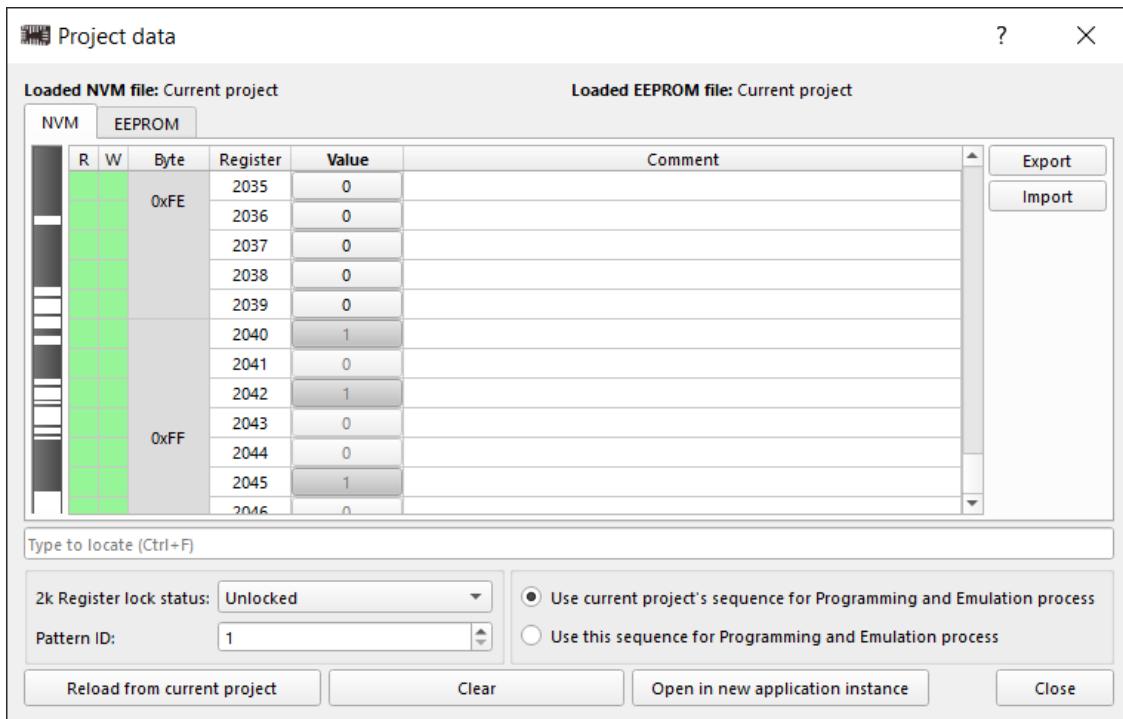
Bit	Chip OTP Data	Data to Program
0 GPIO1_IN_MODE<1:0>	0	0
1 GPIO1_IN_MODE<1:0>	0	0
2 GPIO1_OUT_MODE<1:0>	0	0
3 GPIO1_OUT_MODE<1:0>	0	0
4 GPIO1_RES_MODE<1:0>	0	0
5 GPIO1_RES_MODE<1:0>	0	0
6 GPIO1_PULLUP	0	0
7 Reserved	0	0

Byte view

You can edit *Data to Program* by double-clicking a cell in the *Data to Program* column or open the *Byte view* window by clicking the required cell in the *OTP/Register address* column.

2.2.10 Project data window

This section contains the description of all *Project data window* controls.



Project data window with I2C Read/Write operations

Project data window table:

- *R* and *W* — show if a register is readable and writable

- █ I2C operations allowed
- █ I2C operations is not allowed

- *Value* — allow to change the bit value of a register

- *Comment* — add the notes

Note: The comments are stored neither in chip memory nor in the project file. However, you can *Export NVM* that includes the comments.

Project data window controls:

- *Lock status* — lock NVM Reading/Writing. Use this control to determine the possibility of *Read /Write* operations
- *Pattern ID* — assign an *ID* to the current design
- *Use current project's sequence for Programming and Emulation process* — choose the bit sequence from NVM Viewer for the programming and emulation processes
- *Use this sequence for Programming and Emulation process* — choose the bit sequence from the *Project Data* table for the programming and emulation processes

- *Reload from the current project* — load bit sequence from the NVM Viewer to the *Project Data* table
- *Clear* — set the whole *Project Data* table's bit range to 0
- *Open in a new application instance* — open the bit sequence from the *Project Data* table in a new software instance
- *Export* — save the bit sequence to a text file
- *Import* — load the bit sequence from a text file

You can find the required byte or register by typing its number in the search field. To filter out bytes or registers from the search, use the following markers:

- b — searching for bytes only
- r — searching for registers only

The screenshot shows the 'Project data' window with the 'NVM' tab selected. The main table displays memory locations from 2035 to 2046. A search bar at the bottom left contains the text 'r 42'. An overlay window titled 'Searching for registers' lists register numbers 42 through 425, all categorized as 'Register'. The table columns are labeled R, W, Byte, Register, Value, and Comment.

R	W	Byte	Register	Value	Comment
		0xFE	2035	0	
		0xFE	2036	0	
		0xFE	2037	0	
		0xFE	2038	0	
		0xFE	2039	0	
		0xFF	2040	1	
		0xFF	2041	0	
		0xFF	2042	1	
		0xFF	2043	0	
		0xFF	2044	0	
		0xFF	2045	1	
		0xFF	2046	0	

Searching for registers

Register
42
142
242
342
420
421
422
423
424
425

2.2.11 I2C I/O Tool

I2C I/O Tool allow examination of the project configuration by reading or writing the register data on the chip. *Emulation* or *Test Mode* is required to transfer the data.

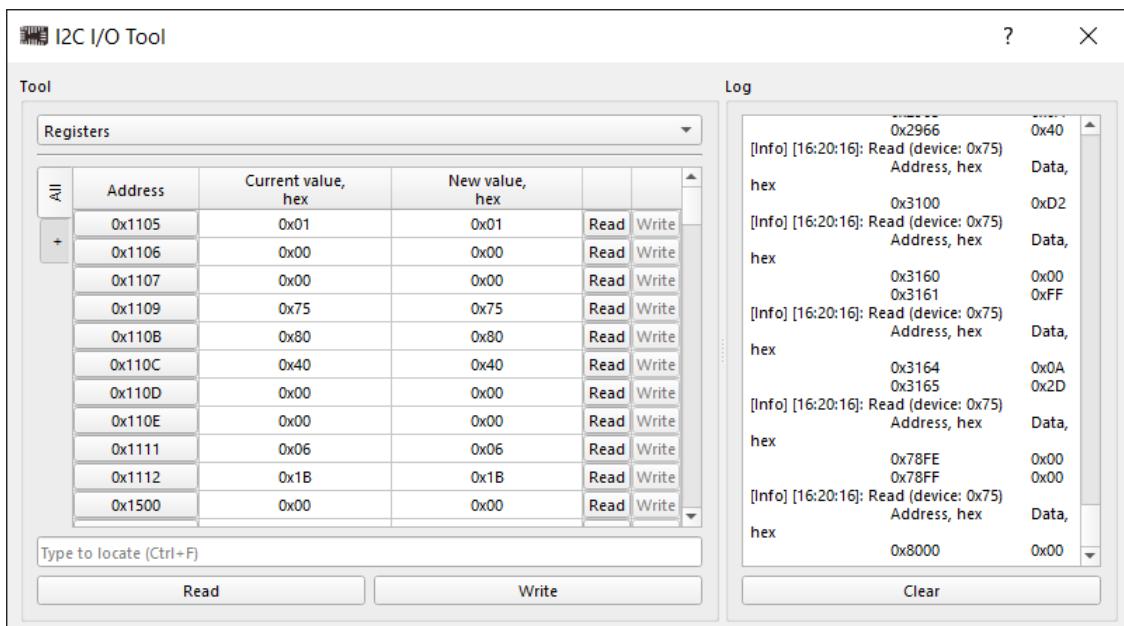
Make sure the correct development platform is selected and click *I2C I/O Tool* at the top toolbar to open the *I2C I/O Tool* window.



Debugging Tool toolbar

Registers

The *Registers* page shows chip NVM. This tool corresponds to the *Registers* tool in section 2.2.8 I2C Tools.

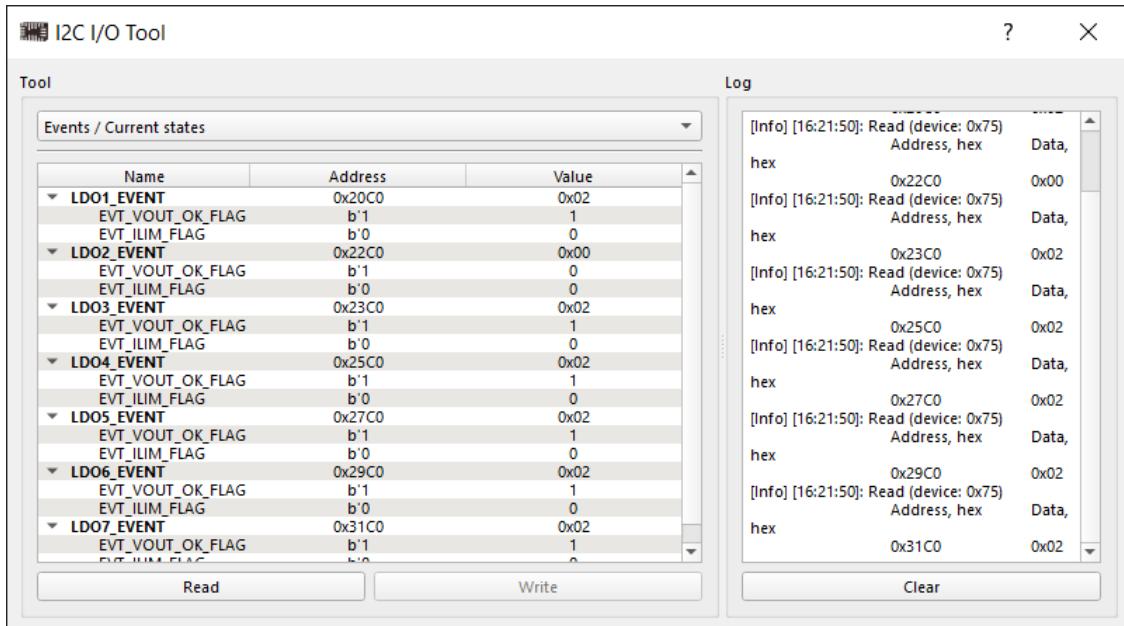


I2C I/O Tool window

Events / Current states

The *Events / Current states* page shows the read-only list of the LDO component events: name,

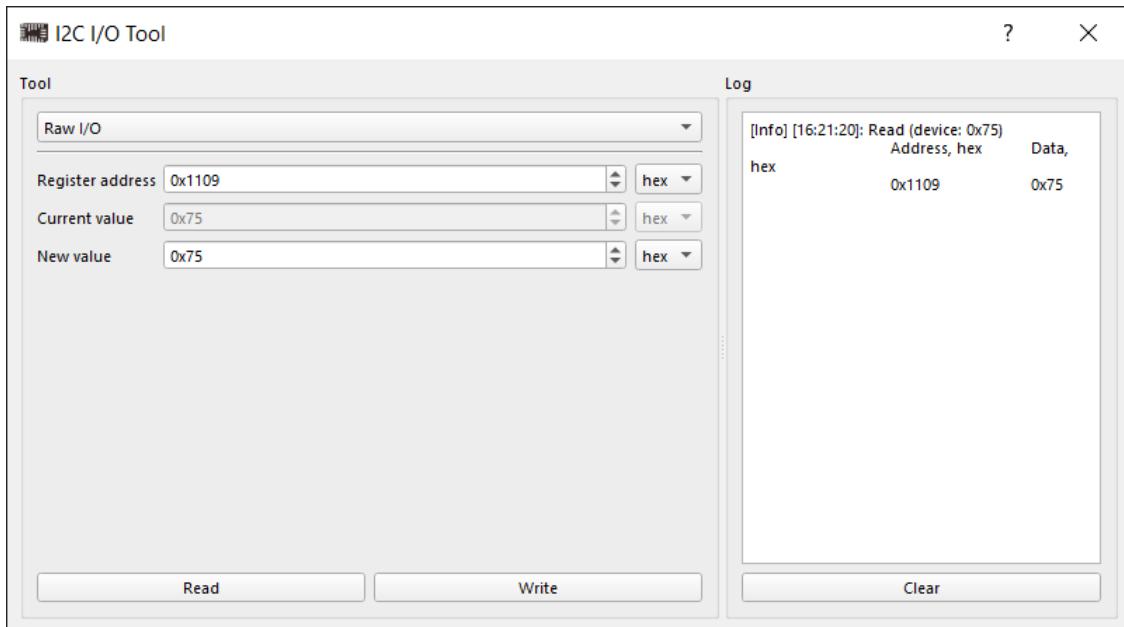
address, and values.



I2C I/O Tool window with events

Raw I/O

The Raw I/O page provides extended access via I2C to the chip registers. It allows reaching the registers that may be inaccessible from the *Registers* page.



I2C I/O Tool window with Raw I/O

Log

The *Log* section lists the results of the recent *read* and *write* operations. This tool corresponds to the *Log* tool in Section 2.2.8 I2C Virtual Inputs.

2.2.12 Logic Analyzer

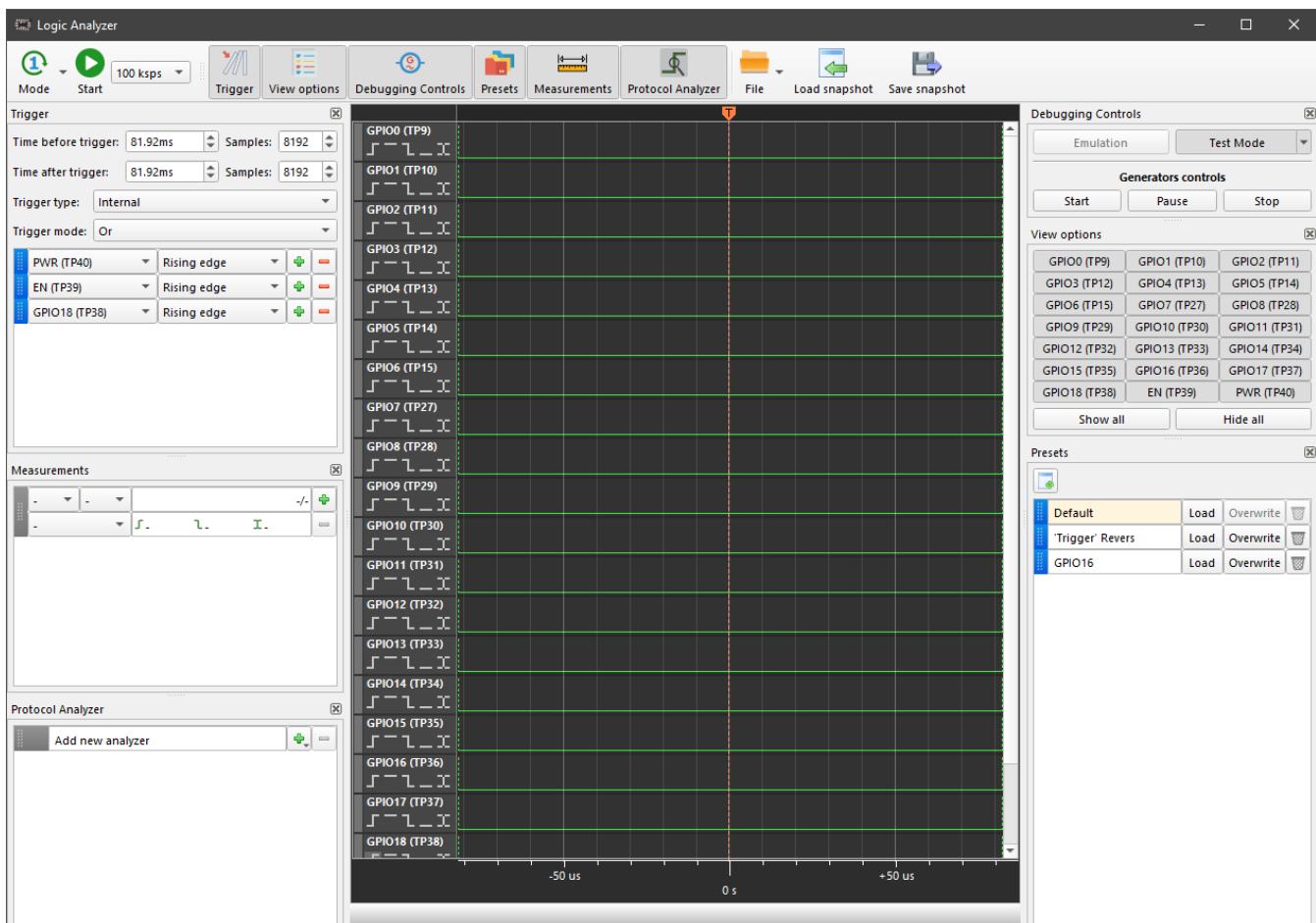
Logic Analyzer tool allows you to capture multiple signals from a digital circuit. It has advanced triggering capabilities and a protocol decoder that helps to see the timing relationships between multiple signals and decode them.

The characteristics of the *Logic Analyzer* are the following:

- Frequency range — 500 Hz - 200 MHz
- Buffer size — 16384 samples
- SPI, I2C, and UART protocol support

Operational controls

- *Start* — launch *Logic Analyzer*. The *Start* button becomes active after *Emulation* or *Test Mode* is started
- *Sampling rate* — drop down selector of the signal sampling frequency



Logic Analyzer window

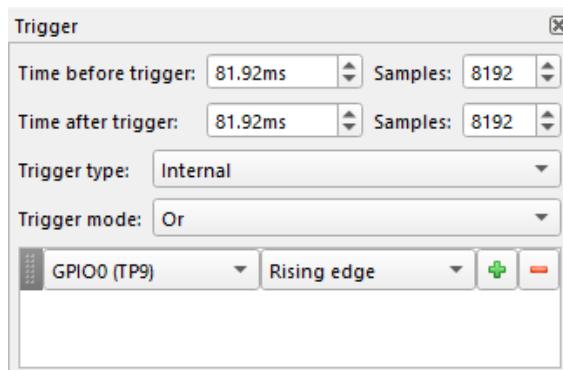
Mode

There are three operating modes:

- *Auto mode* — trigger events are ignored. The signal on the pins is shown as a continuous waveform
- *Single shot* — refreshes the waveform pattern once a trigger event is detected
- *Normal mode* — refreshes the waveform pattern each time when a trigger conditions are met

Triggers

Set time parameters to choose the trigger time position or a specific sample.



Trigger parameters

You can also change the trigger type:

- *Hardware button* — the trigger is activated by pressing a physical button on the board
- *Internal* — the trigger is activated when the trigger condition, which is set in the trigger list, is met

You can set the trigger mode for the internal trigger:

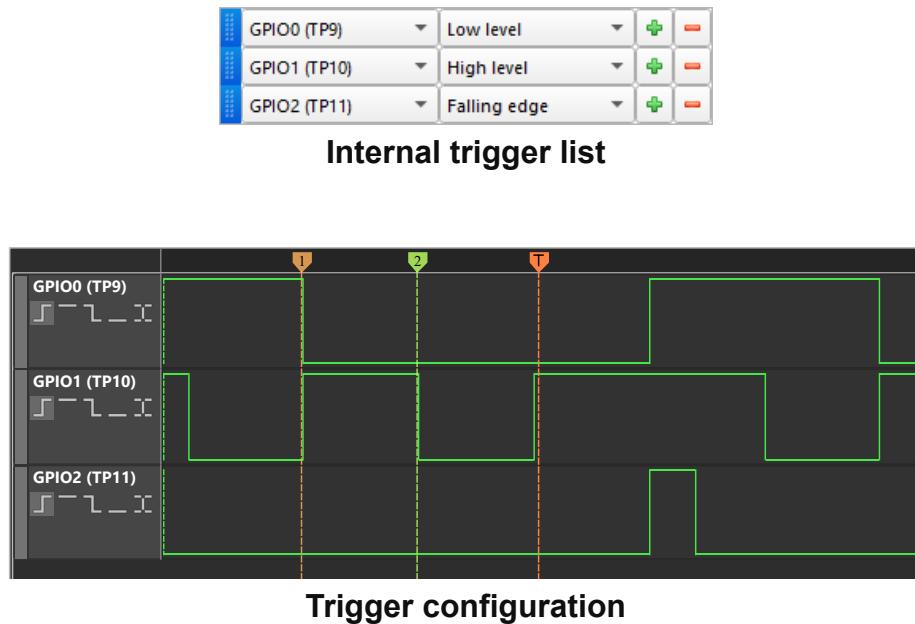
- *Or* — any of the trigger conditions added to the trigger list are met
- *And* — all trigger conditions added to the trigger list are met

An internal trigger must have the following settings specified:

- *Channel* — assign the trigger to the channel
- *Condition* — *Rising edge*, *Falling edge*, *Both edges*, *High state*, and *Low state*

Note: Set proper trigger conditions for successful sampling (e.g. *Rising edge*, *Falling edge* together

with *And* trigger mode may result in improper trigger work).

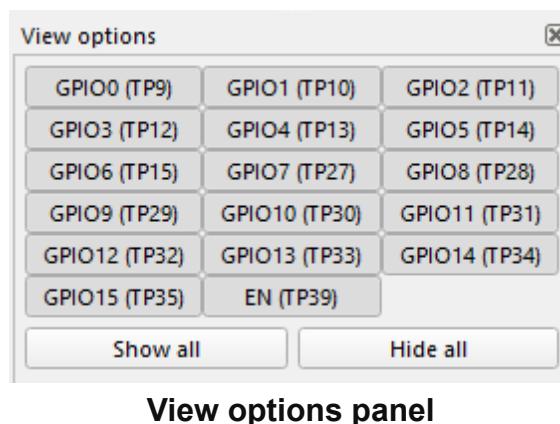


Debugging Controls

The *Debugging Controls* panel is responsible for *Emulation* and *Test mode*, and *Generator controls* functionality.

View options

You can show or hide the channels in the *View options* panel.



Presets

You can store your *Logic Analyzer* configurations in presets and restore a previous configuration when needed.

Presets			
	Default	Load	Overwrite
	PWR	Load	Overwrite
	sorted by GPIO nr	Load	Overwrite
	PWR first graph	Load	Overwrite

Presets

- — create a new preset with the current *Logic Analyzer* configuration
- Load — load a selected preset configuration
- Overwrite — overwrite preset with the current *Logic Analyzer* configuration
- — remove the selected preset
- Default — load the Default preset of the *Logic Analyzer* window
- Autosaved @ [time] — the modified preset is saved each time the *Logic Analyzer* window, the *Debug tool*, or *ForgeFPGA Workshop* is closed

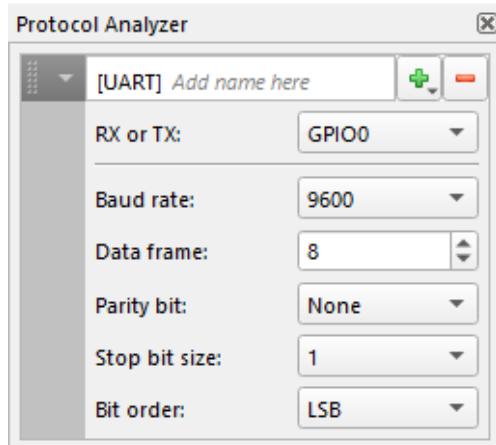
Protocol Analyzer

The *Protocol Analyzer* allows you to decode data according to a protocol.

Protocol Analyzer	
	Add new analyzer
	I2C
	SPI
	UART

Protocol Analyzer

To analyze captured data, click the  button and select one of the protocols.



Protocol settings

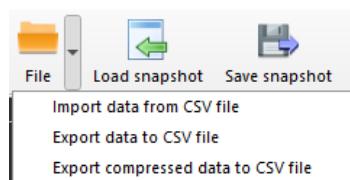
Then, choose a channel for analysis and modify protocol settings if necessary. The decoded data will be displayed above the corresponding plot.



Decoded data

Import / Export actions

You can save/import the waveform data in the CSV format. These options are grouped under the *File* button at the top toolbar.

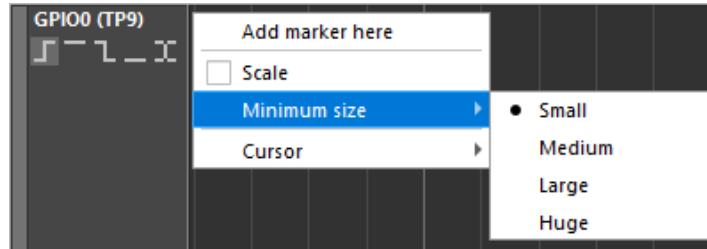


Export and import operations

Plot widget

The *plot widget* displays the waveforms in the *Logic Analyzer* window. You can change the way a plot is shown from the plot context menu. Right-click on a plot area to add a marker, show or hide the time

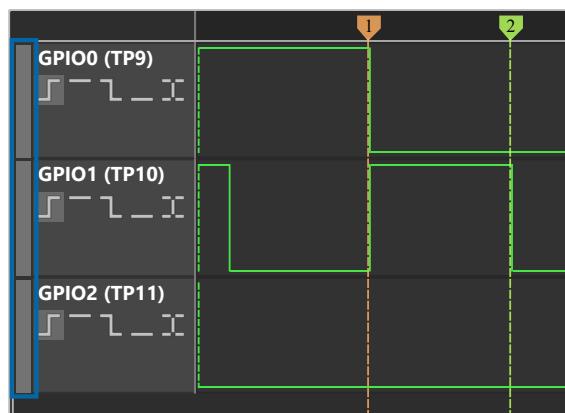
scale, change the plot height, and select the cursor width.



Logic Analyzer Plot menu

You may do the following actions with the plot area:

- Move left/right by click + drag left/right
- Scroll up/down by mouse wheel Up/Down
- Zoom in/out by *Ctrl* + mouse wheel Up/Down
- Quick zoom in/out with the *Middle mouse button* click + drag Up/Down
- Reorder waveforms by Drag and Drop



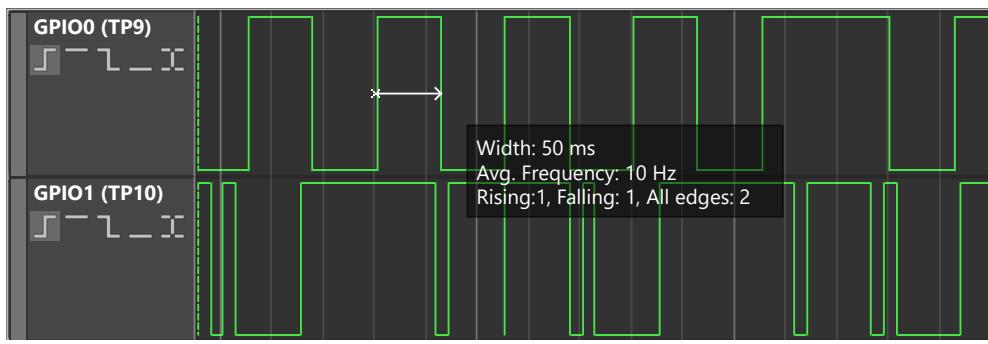
Drag and Drop Handlers

Cursors

A cursor is a measurement tool for calculating waveform data between the edges of one or more plots. To see the cursor, hover a measured section of a waveform.

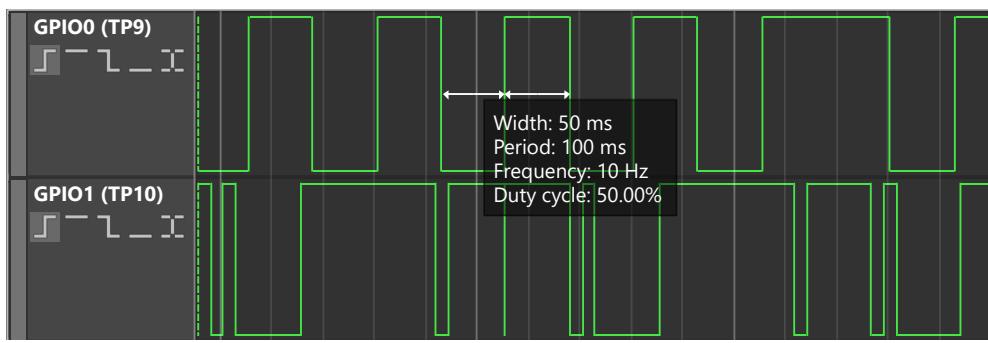
A *Half Period* cursor measures the distance between the two nearest edges at a hovered section of a

waveform.



Half Period cursor and sample measurements

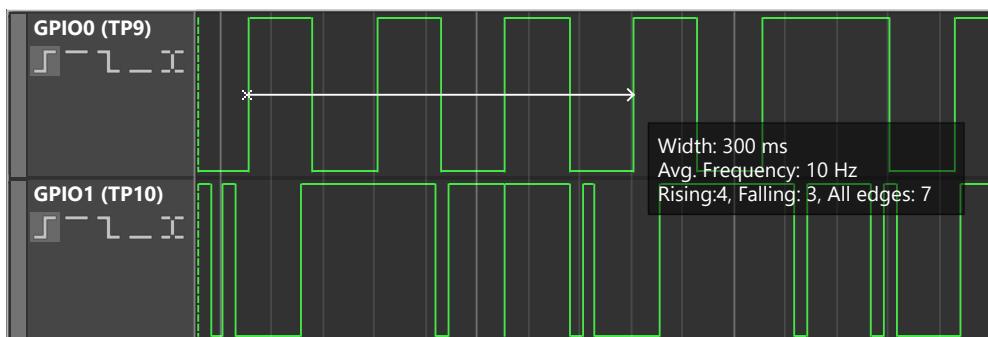
A *Period* cursor measures the width of the hovered half period, the duration of a full period, calculates the frequency, and which fraction of the full period the hovered part of the period is, which is the *Duty cycle*.



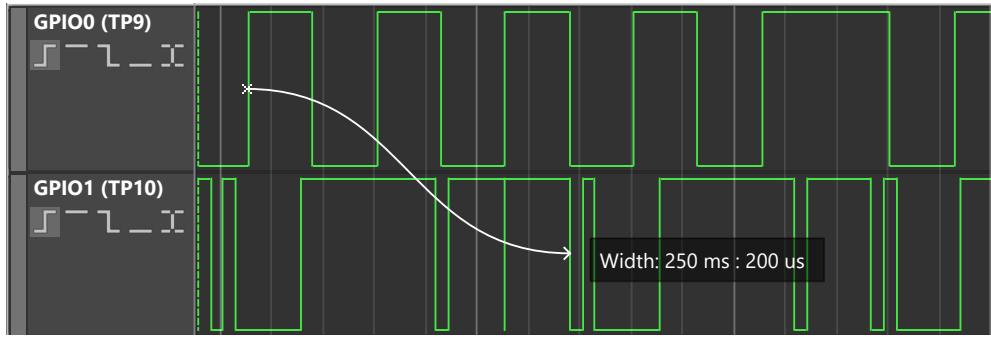
Period cursor and sample measurements

To change the cursor width, open the context menu and select either the *Period* or *Half Period* option.

To measure data at a distance that exceeds one period, click the edge you want to measure from and hover over the edge you want to measure to. This will give you the total duration of the measured section, the calculated average frequency, and the quantity of rising and falling edges as well as their sum.



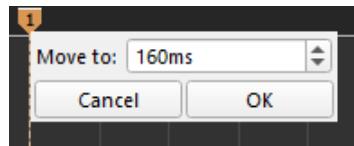
You can also measure the width between the edges on the distinct waveforms.



Markers

You can do the following actions with markers:

- Add a new marker with a *Ctrl* + left click on the markers panel
- Set a new marker from the plot's context menu
- Remove the marker with a *Ctrl* + right-click on the marker head
- Move the marker by a left click + drag
- Move the marker from the context menu by a left click on the marker's head



Marker Move To

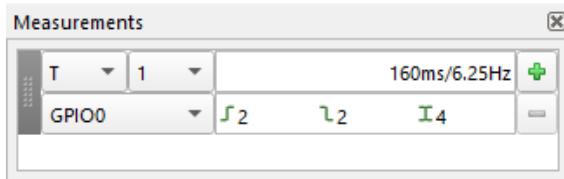
- Select the marker by clicking on the marker head: the selected marker has a white line on top.
- Move the selected marker to the closest visible left/right front by *Ctrl* + *Left/Right*
- Move the selected marker left/right by one sample with *Ctrl* + *Shift* + *Left/Right*

Marker measurements

- Measurements — period and frequency. Frequency value is rounded to four decimals
- Additional measurements — the count of *Rising Edges*, *Falling Edges*, and *All Edges* in the period between the markers

To calculate all measurements between two markers, select the markers and a channel in combo

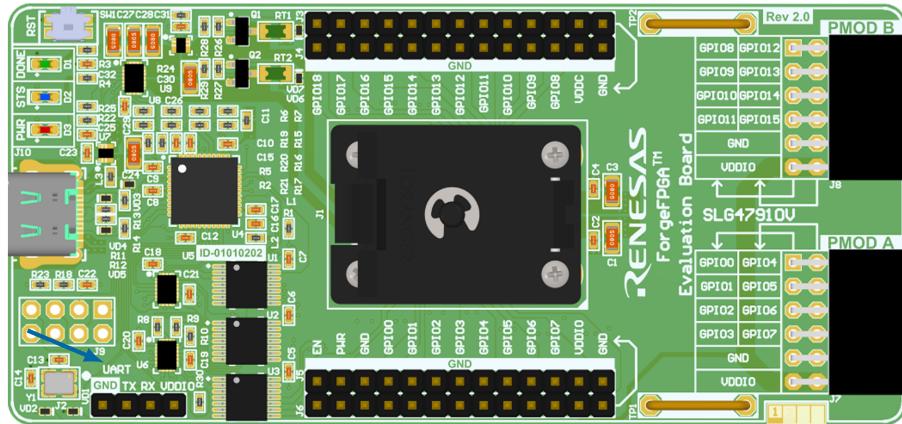
boxes.



Markers measurements

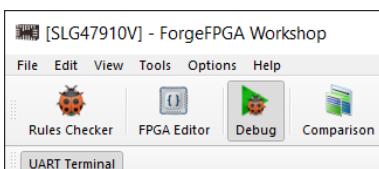
2.2.13 UART Terminal

UART terminal is a console tool used for serial communication between a board and an external device. While developing a project, the UART terminal helps you *read* and *write* via UART protocol.

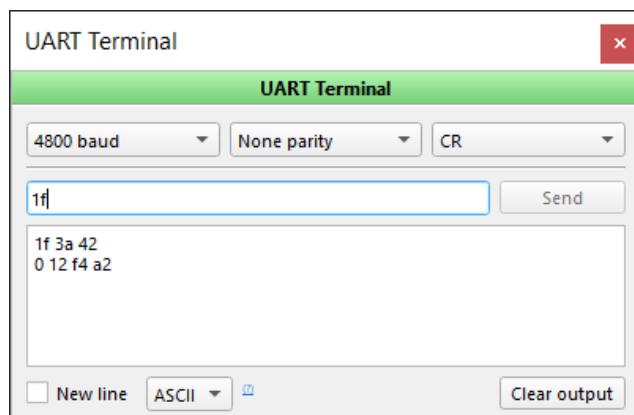


The UART port on the ForgeFPGA Evaluation Board

To start working with the terminal, ensure that the appropriate development platform is selected. Open the *UART Terminal* tool at the top toolbar.



UART Terminal tool



UART Terminal window

- *Baud rate* — select the baud rate from the list for *read* and *write* operations
- *Parity control* — select whether and how the parity control bit is used
- *Line ending* — select which character is used as a new line character: *No line ending*, *New Line (NL)*, *Carriage Return (CR)* or both *New Line* and *Carriage Return*. This option is available in ASCII mode only.
- *New line* — add a new line after each byte sent if set; otherwise, send the whole message at once
- *Data Format* — select the data format: ASCII or Hex. For the Hex, the input case is independent, numbers are separated by a space.

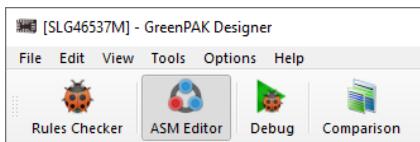
The input field allows copy/paste operations. The output field allows copying the received data.

2.3 Configuration tools

2.3.1 Asynchronous State Machine Editor

GreenPAK family devices featuring the *Asynchronous State Machine (ASM)* macrocell allows you to develop personalized state machine designs. You can establish state definitions, define permissible state transitions, and specify the signals responsible for initiating each state transition. Moreover, you can link this macrocell to various I/O PINs and other internal GreenPAK components to activate inputs for state transitions. Outputs from the macrocell can be conveniently directed to other internal macrocells or I/O PINs as needed. *ASM Editor* allows configuring the ASM component using the state diagram and setting the output configuration for the ASM Output macrocell.

To open the ASM Editor, click the *ASM Editor* button on the toolbar or double-click the ASM component.

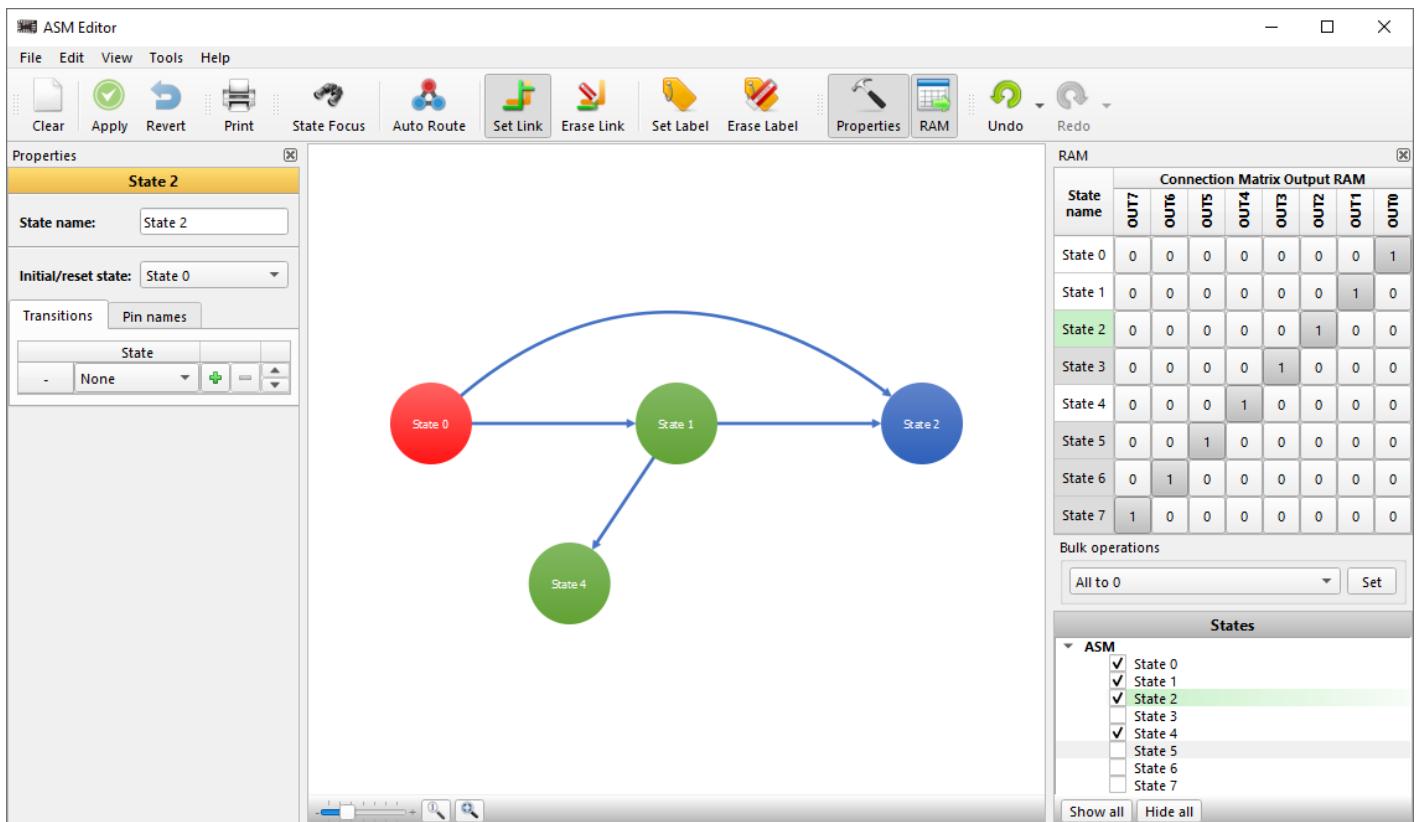


ASM Editor button on the top toolbar

ASM Editor's interface

You can view and set a state's properties in the *Properties* panel.

The *RAM* properties panel at the right contains the *Connection Matrix Output RAM*. This feature establishes the relationship between controlled outputs and specific states. Additionally, certain part numbers might have their output pin behavior regulated by these states. This behavior is indicated in the *GPOs Output RAM* table.



Setting a state transition

Bulk operations allow applying a selected operation (*All to 0*, *All to 1*, *Invert*, and resetting to *Default*) to the whole table.

You can show or hide states by toggling the corresponding checkboxes within the States section located in the lower right area of the *RAM* properties panel.

The *main area* of the ASM editor depicts states. The color of a state changes according to the state's interaction options.





initial (reset) state



state transitions limit reached

Configuring States

If states haven't been established, the *ASM Editor* will display two detached states. These states are illustrated as circles. To relocate a state, use the *left mouse button* to drag its inner circle. **Note:** dragging the outer circle will not produce any changes. To establish a connection click the outer ring of one state and then click on the ring of another state.

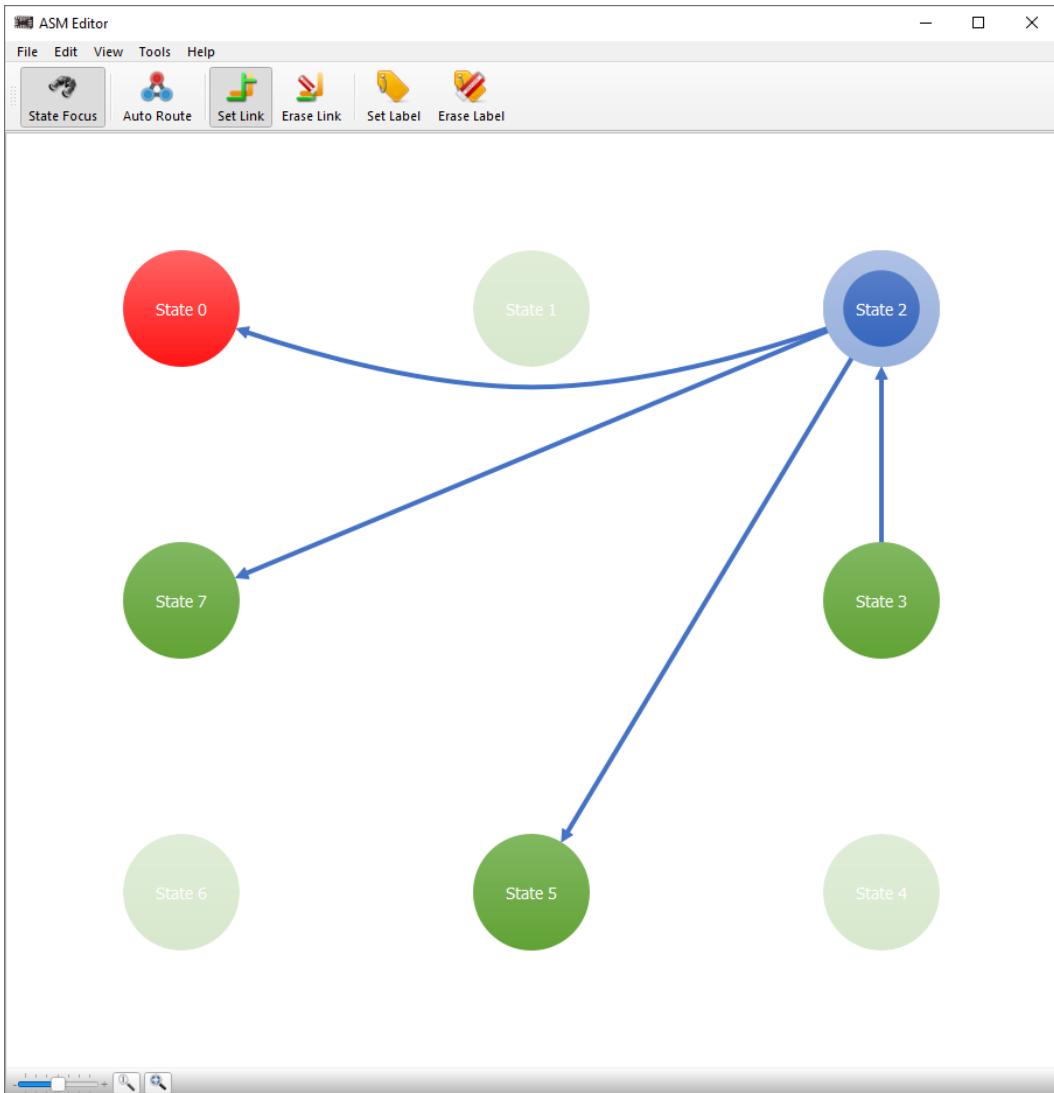
For some chips, a state may be directed to itself by selecting the state's outside ring on both the first and second click. The number of states a particular state can transition to is unrestricted; it can transition to all the utilized states or none at all.

To rename a state, double-click on the inside of the state circle or select the state to access the *State name* setting from the *Properties* docker at the left.

Transitions can be adjusted to best fit your scheme. Drag the link to change its shape. Select the *Edit path* mode from a connection's context menu if you need more tweaking. **Note:** A link's shape is updated automatically and the manual adjustments get discarded every time a state is repositioned.

You can let the software reorder the states and transitions by clicking *Auto Route* on the toolbar. You can also add a label to a transition. To do that, click *Set label* at the top menu, or click the mouse right button on an existing transition and select *Set label*.

You can also focus on the states by clicking the *State focus* button on the toolbar.



State focus mode

In order to set an *ASM* configuration to the *NVM*, apply it by clicking *Apply* on the toolbar. To return to the latest saved state of the *ASM* click *Revert*.

The context menu of a state includes:

- *Edit name* — set state's name
- *Initial state* — set the current state as the initial
- *Hide* — hides the current state

You can configure a state's properties in the *Properties* panel at the left. The available tabs are:

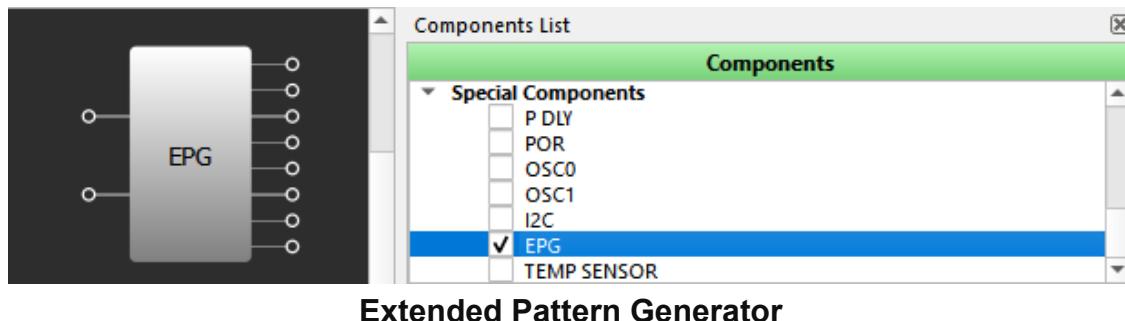
- *Transitions* — lists of the states to which the current state transitions. Allows adding a new transition by selecting another state from the list.
- *Pin names* — this tab's data is common for all states. It lists pin names set the current state as

the initial

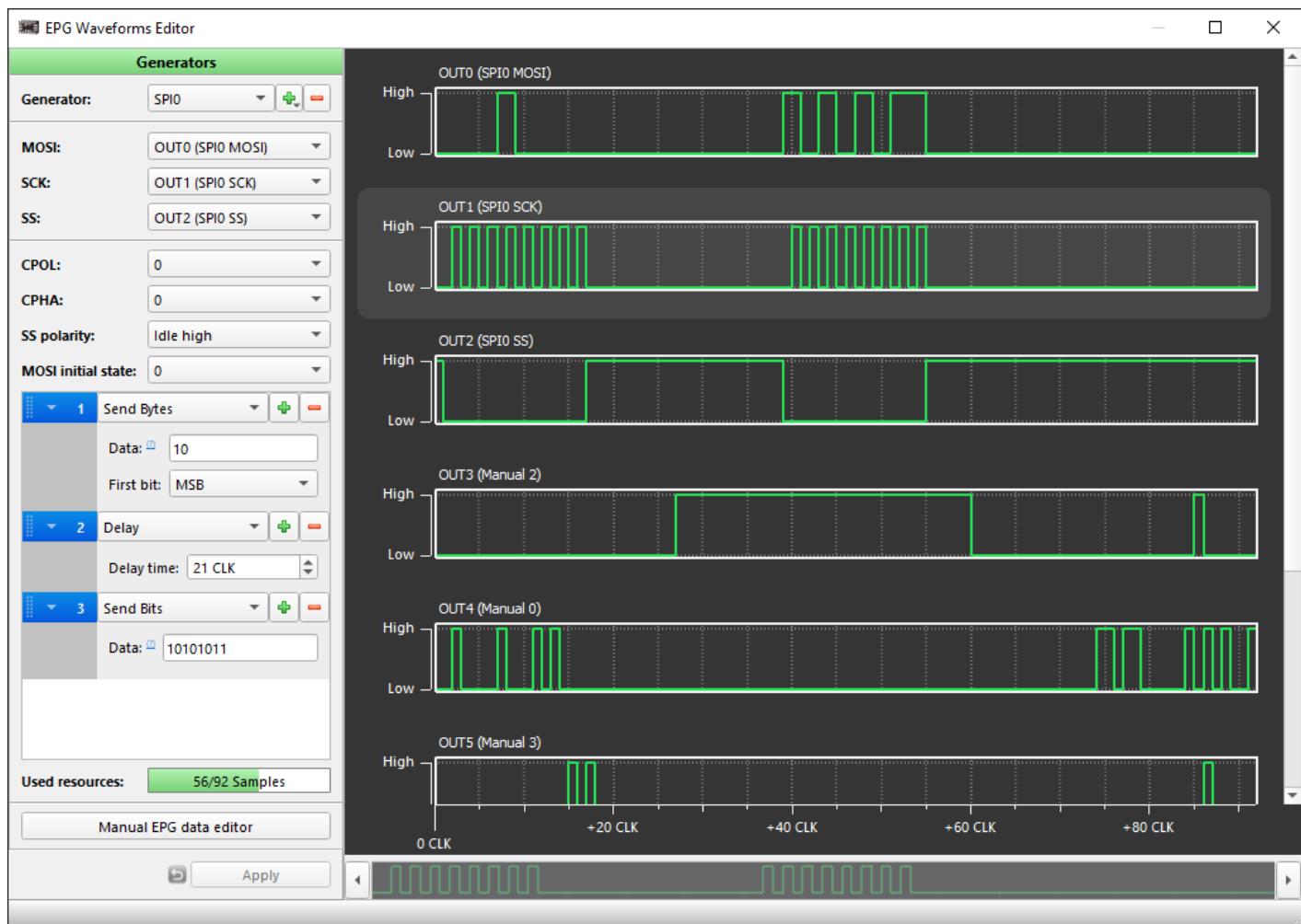
- *RE (Rising Edge)* — lists the states the current state transitions to and apply *Rising Edge* to the transition, otherwise ASM will be level sensitive.

2.3.2 EPG Waveform Editor

The *Extended Pattern Generator (EPG)* is a component featuring up to 8 outputs. Each can be individually configured using up to a 92-bit large logic pattern. It retrieves data from non-volatile memory (*NVM*) and delivers it to the outputs bit by bit on each rising edge of the *CLK input* signal.



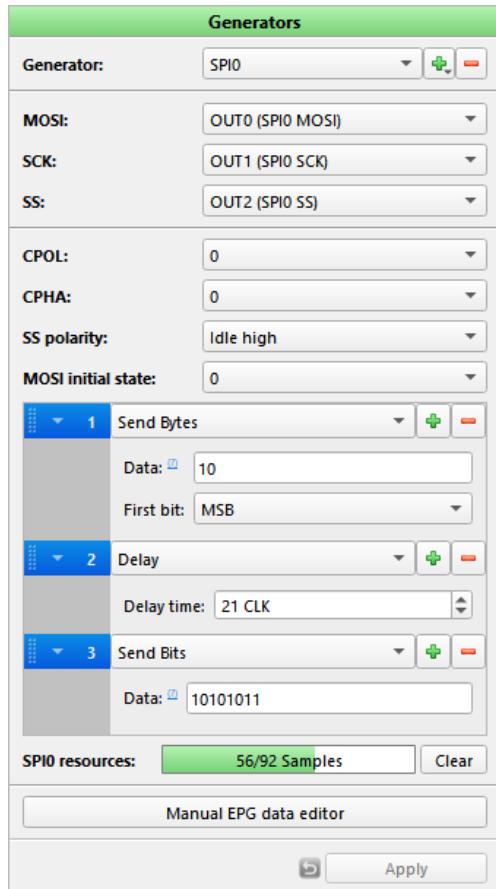
You can configure the *Extended Pattern Generator* in the *EPG Waveforms Editor*. This tool allows you to choose from the available predefined generators, including *SPI*, *I2C*, *PWM*, or *Manual*, and assign the output accordingly. Each of these generators offers a range of adjustable settings, making it more convenient for you to create desired patterns.



EPG Waveforms Editor

Within the *EPG Waveforms Editor*, you can also find a *resource meter* that visually represents the number of bits used in the pattern. This feature provides transparent view of the memory and resources allocated to the present pattern.

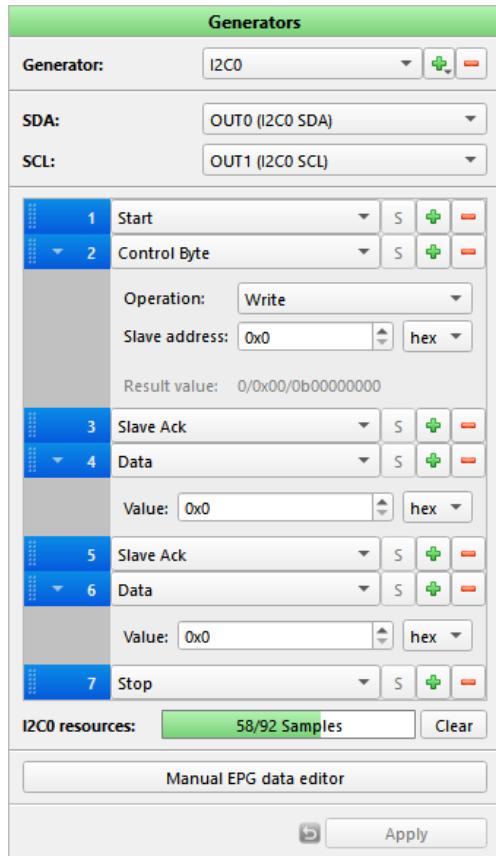
The SPI generator allows you to select and configure the output pin using the *command editor*.



SPI generator command editor and resource meter

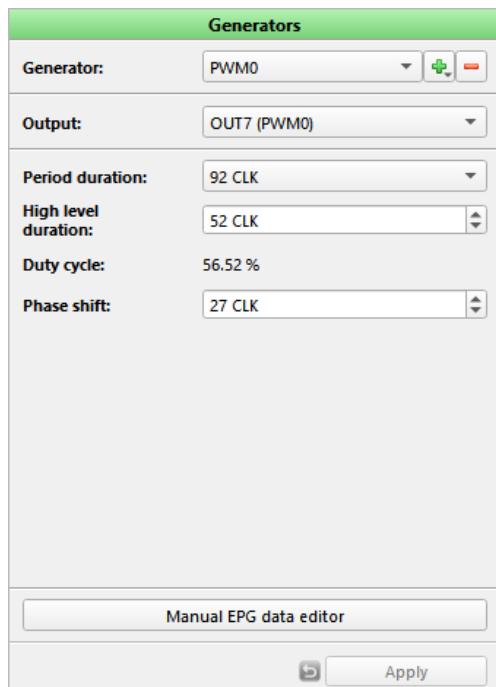
The I2C generator allows you to select the SDA and SCL output pins and define commands using the *command editor*. Within the editor, you can select basic commands from a menu, such as *Start*, *Stop*, and GreenPAK access operations like *Read/Write*. The I2C generator has an S button that allows you to

break down complex GreenPAK access commands into a series of basic commands.



I2C generator command editor and resources meter

The PWM generator configures the *output pin*, *period duration*, *high level duration*, and *phase shift*.



PWM generator options

The *Manual* generator allows configuring the bit values.

Bit #	Value
0	0
1	1
2	0
3	1
4	0
5	1
6	0
7	1
8	0
9	0
10	0

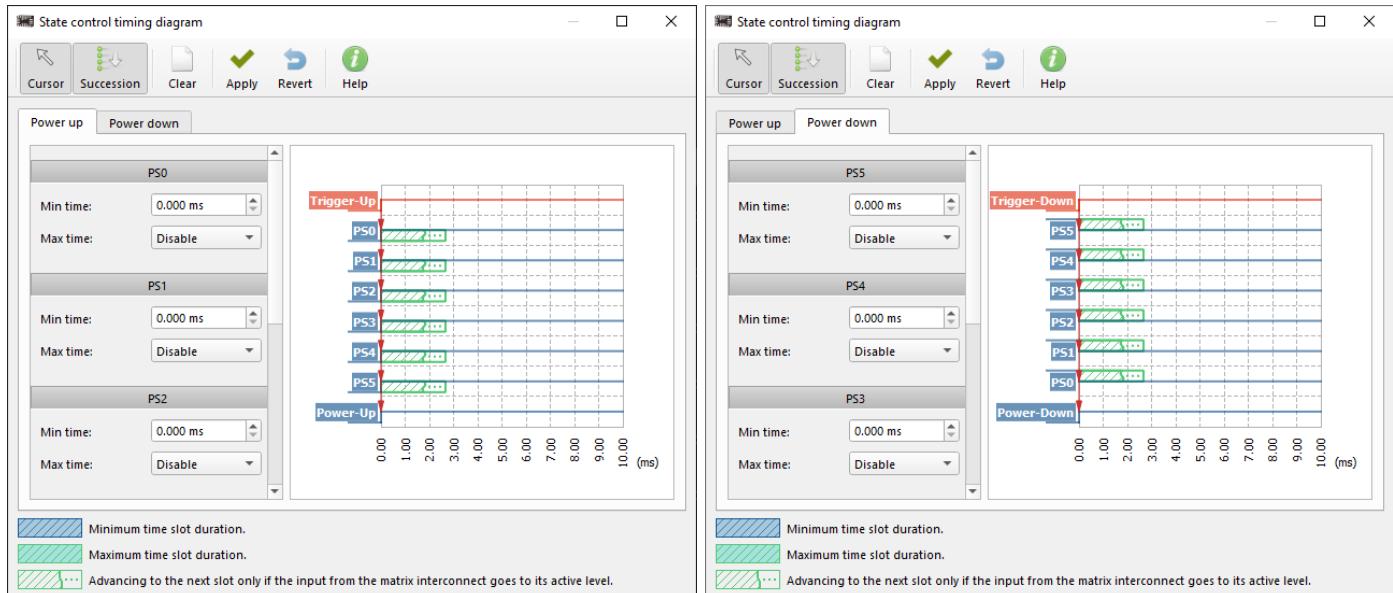
Manual generator options

When the system is powered up, the behavior of the *EPG* varies depending on the signal that received at the *nReset* input. In particular, if the *nReset* input is in an active *LOW* state, the *EPG* will display the initial value at the output. Conversely, if the *nReset* input is active *HIGH*, the *EPG* will display user-defined patterns at the output. This feature enables you to customize the output of the *EPG* to specific needs. Moreover, *EPG* can work continuously when *CLK* is being applied in *Overflow* mode or keep at the last byte in *Stop at Boundary* mode.

2.3.3 Timing Diagram

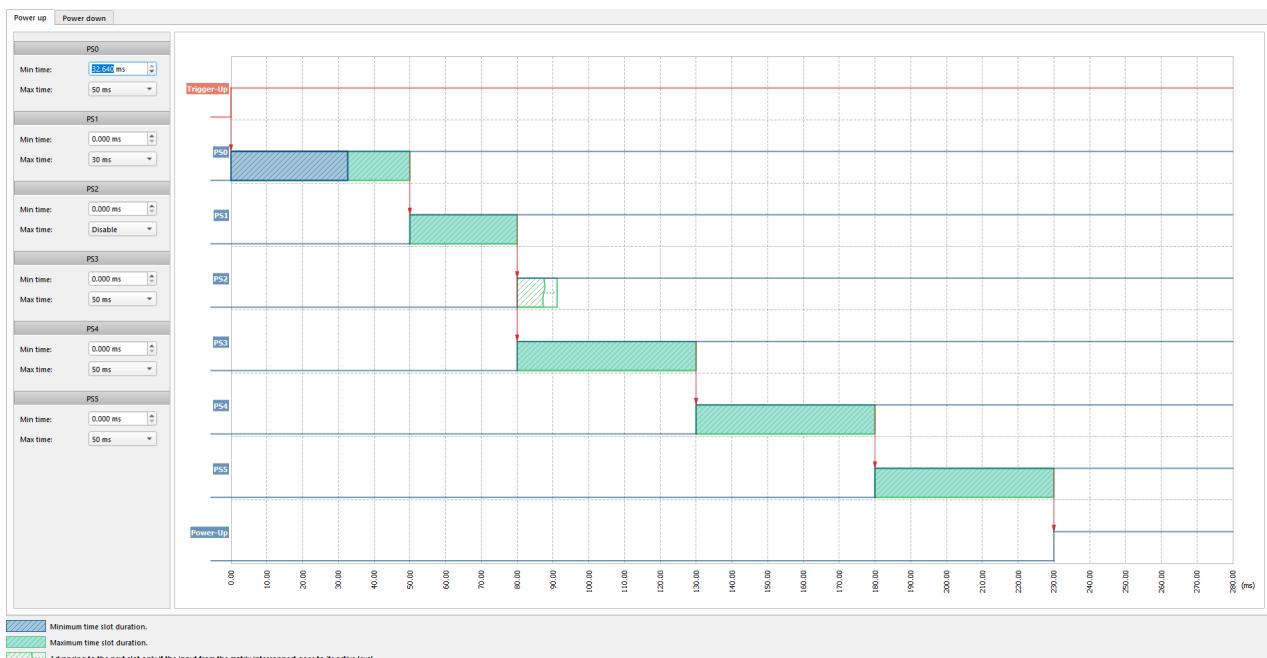
State control timing diagram is a tool, which allows managing the *Power sequencer* block configuration. The *Power sequencer* controls the *power-up* and *power-down* timings for the six resource enable outputs, which feed the matrix interconnect. The timing sequence is divided into six slots, which are periods between the events. You can set the minimum and maximum duration per each slot. The sequence is initiated with the *trigger-up* and *trigger-down* control signals from the matrix interconnect. The *Power*

sequencer and, therefore, *Timing diagram* tool are available e.g. SLG51000/1 and SLG51002.



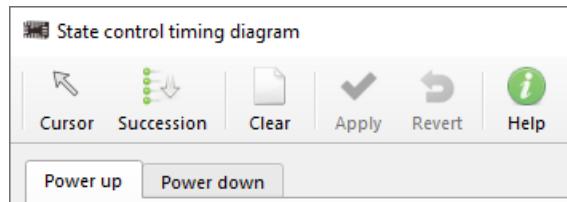
Timing diagram

The slot view depends on the Min. time and Max. time settings. You can see the legend at the bottom of the *Timing diagram* window.



Slot view examples

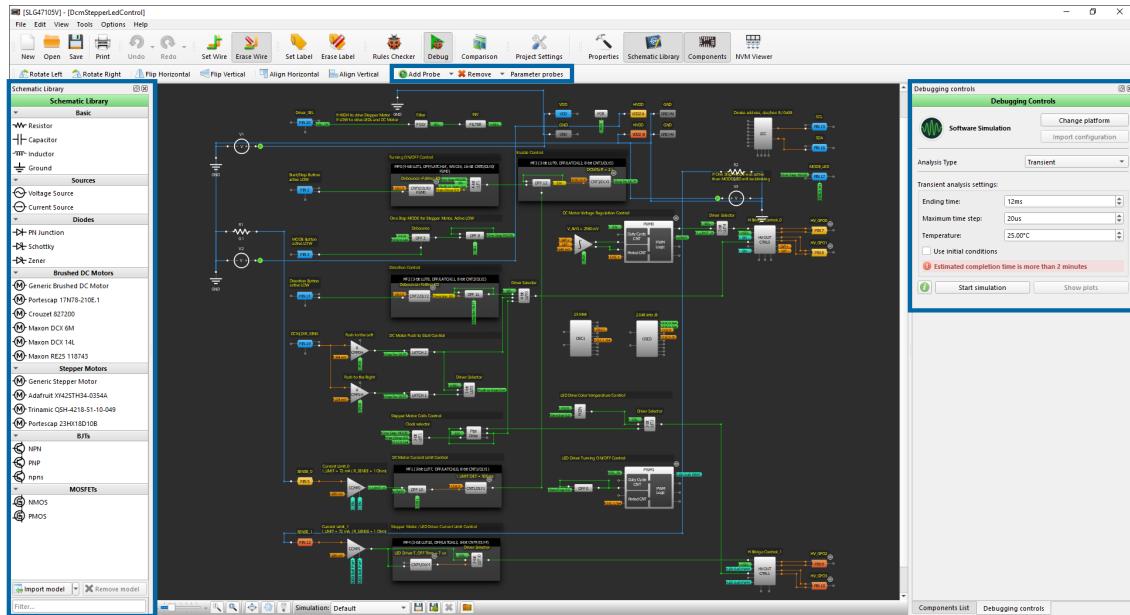
Use the toolbar controls to work with the tool.



Timing diagram options

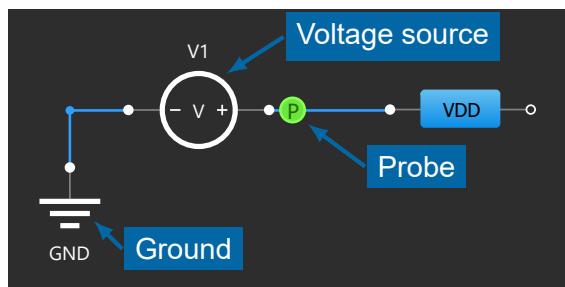
2.4 Software simulation tool

The *Software Simulation* mode enables electronic circuit simulation. It uses mathematical models to replicate the behavior of the chip macrocells and the external components. To start *Software Simulation*, click *Debug* on the toolbar and select the tool in the *Development Platform Selector*. Refer to the *Hub* window → *Development* tab → *Details* to check the simulation support availability for a certain Part Number.



Software Simulation Tools

Before starting an analysis, you can use the external components from *Schematic Library* and add *probes* to configure the simulation environment parameters.



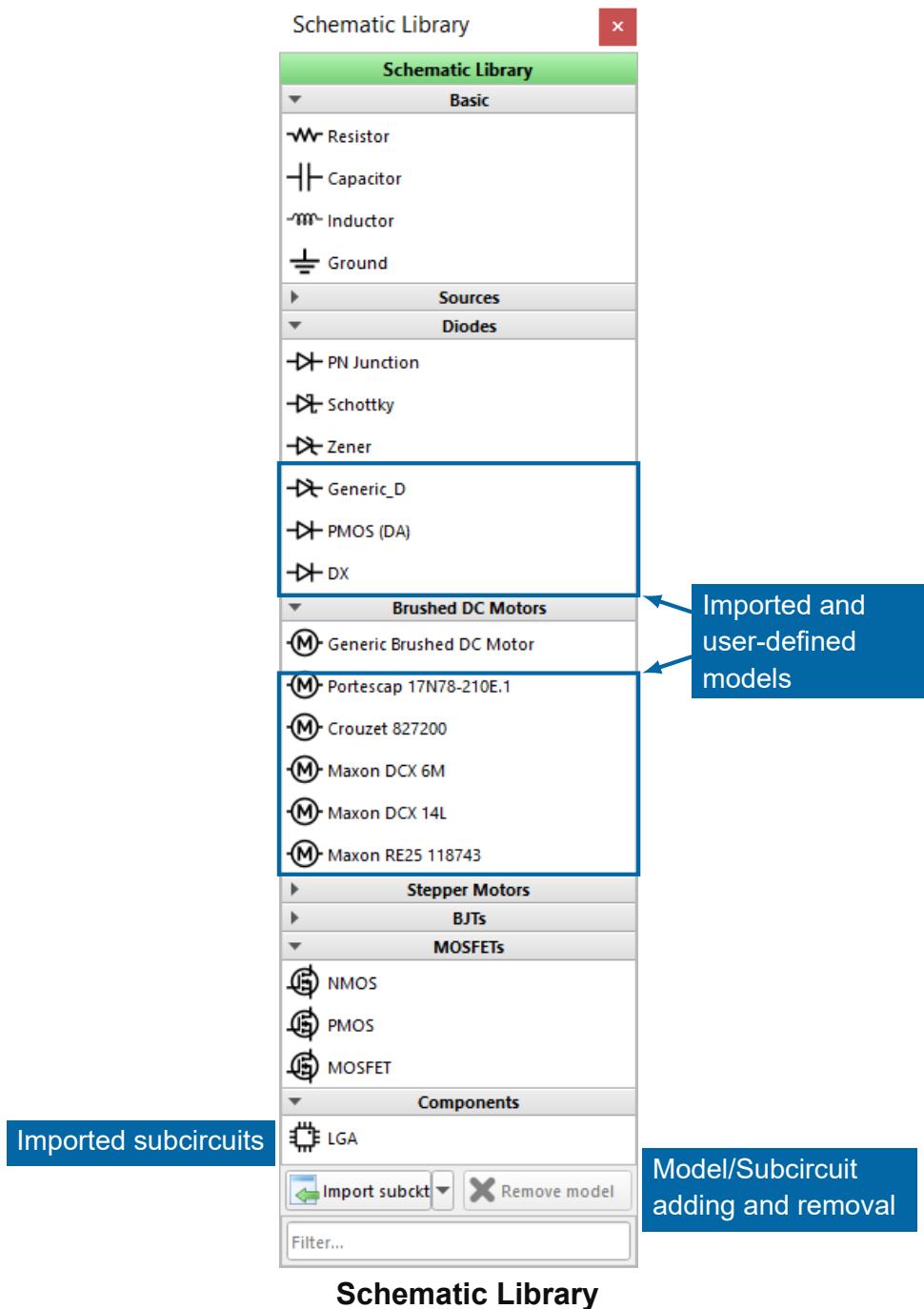
Basic Simulation Tools

2.4.1 Schematic Library

The *Schematic Library* is a repository of the *external components* you can apply to the design. External components are not a part of the chip; however, adding and configuring them allows you to simulate the system's behavior.

Click the component and then the work area to add it (discard adding with a right-click). You can connect the external components only to the external I/O macrocell pins or to the other external

component's pins. By default, voltage source(s) and GND are added to specific pins.

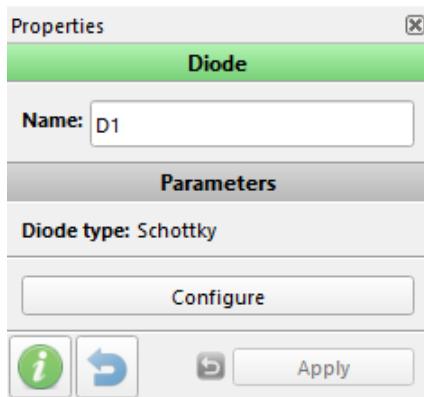


You can also add a custom component to the *Schematic Library* list by using the *Import model/subcircuit* feature. See the corresponding buttons on the *Schematic Library* panel (read more in section [2.4.3 Working with models and subcircuits](#)). Find imported models and subcircuits in the *Schematic Library* in the designated location.

Unlike the default external components, an imported model/subcircuit can be deleted from the library. Click the component and then *Remove model* below.

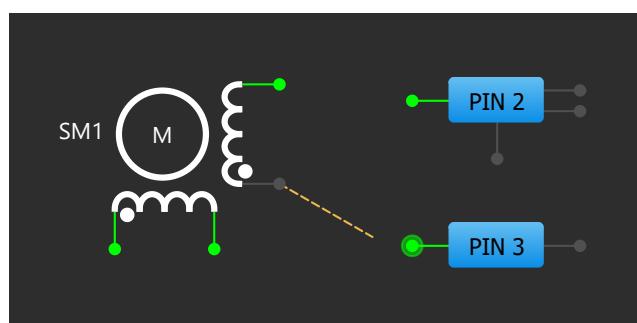
2.4.2 External components

You can find the list of the external components on the [Schematic Library](#) panel. Double-click a component on the work area to open its *Properties* panel. Click *Configure* on the panel for more advanced settings.



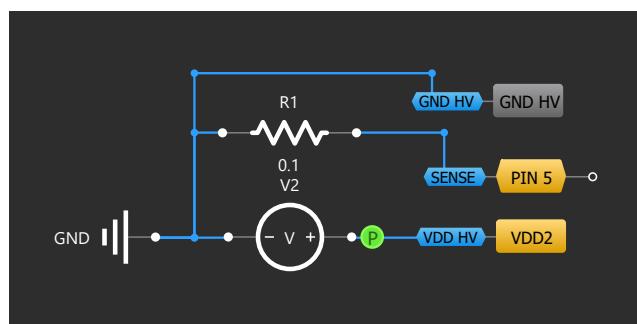
Configuring basic parameters

Connect an external component pin to an external I/O pin of the macrocell using the *Set Wire* tool. Upon setting a connection, the available pins become highlighted.



Adding wire to an external component

Here, you can see the external connection type (all connection types are described in the section [2.1.5 Connections](#)).

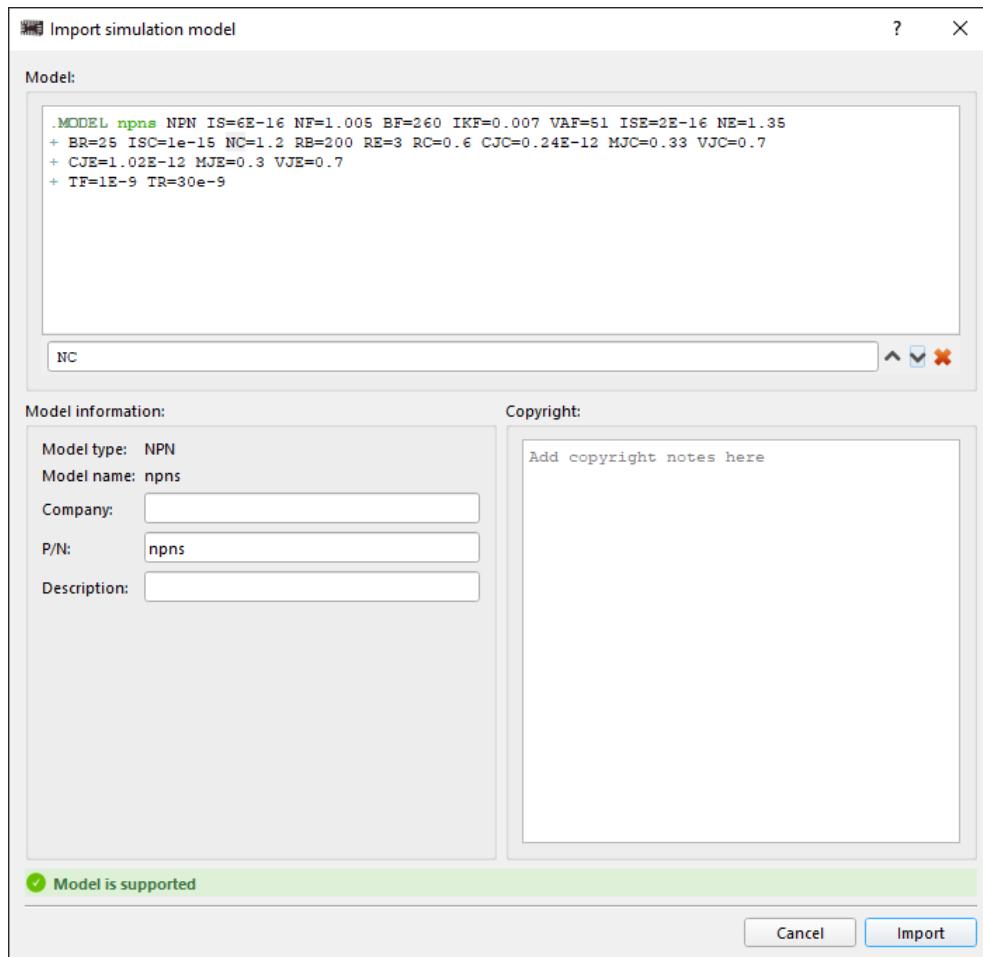


External connections

2.4.3 Working with models and subcircuits

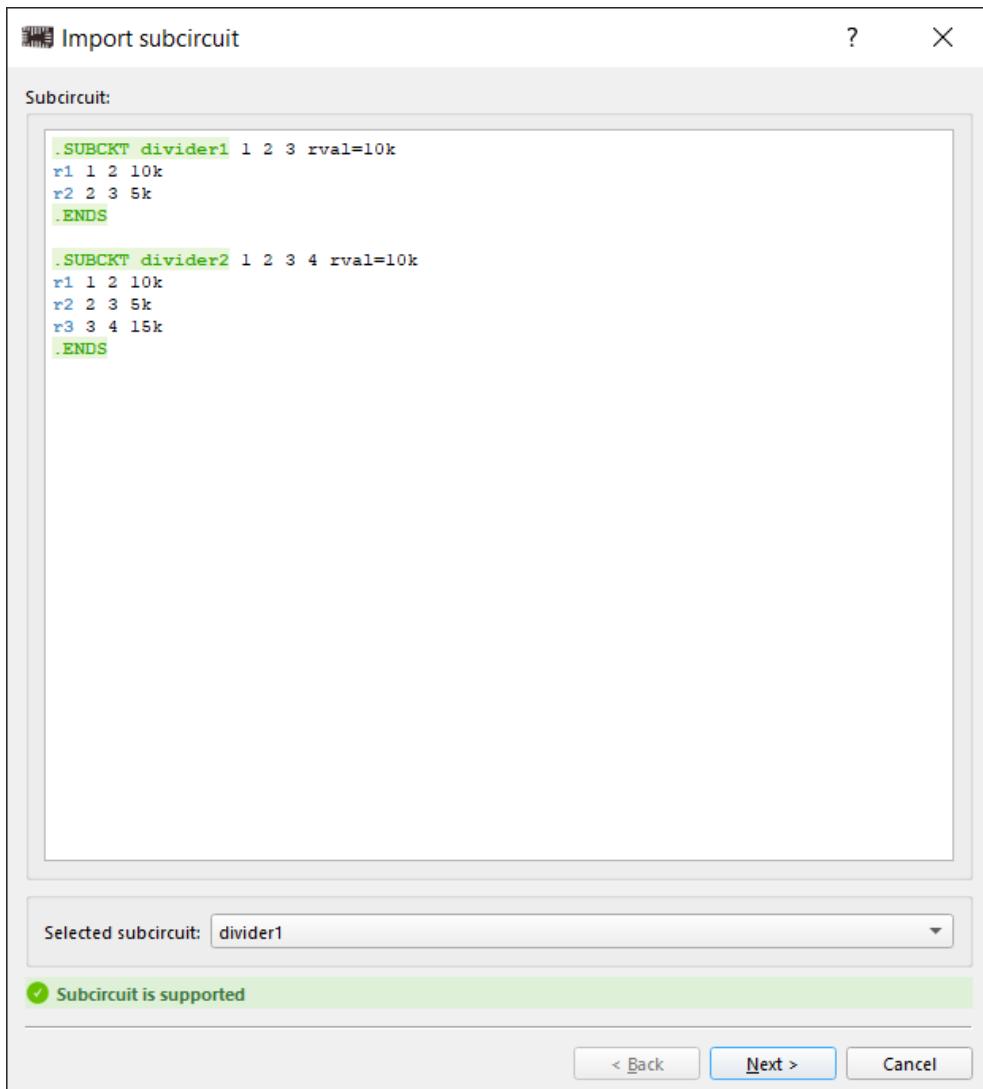
You can import third-party SPICE models and subcircuits to the project. *Import model/subcircuit* opens the *Import* window, where you can insert a SPICE simulation model/subcircuit, fill in additional information, and add it to the library.

The dialog validates a SPICE syntax of a model/subcircuit. Only one model can be imported at a time.



Import simulation model window

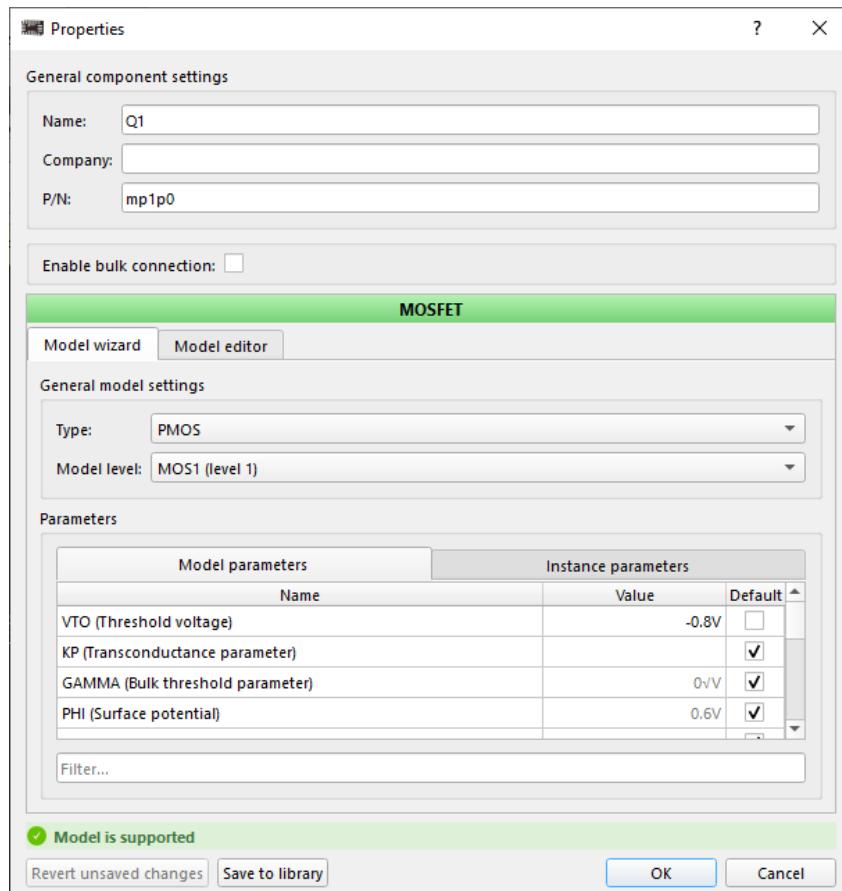
Unlike models, you can import multiple subcircuits and select one of them as main:



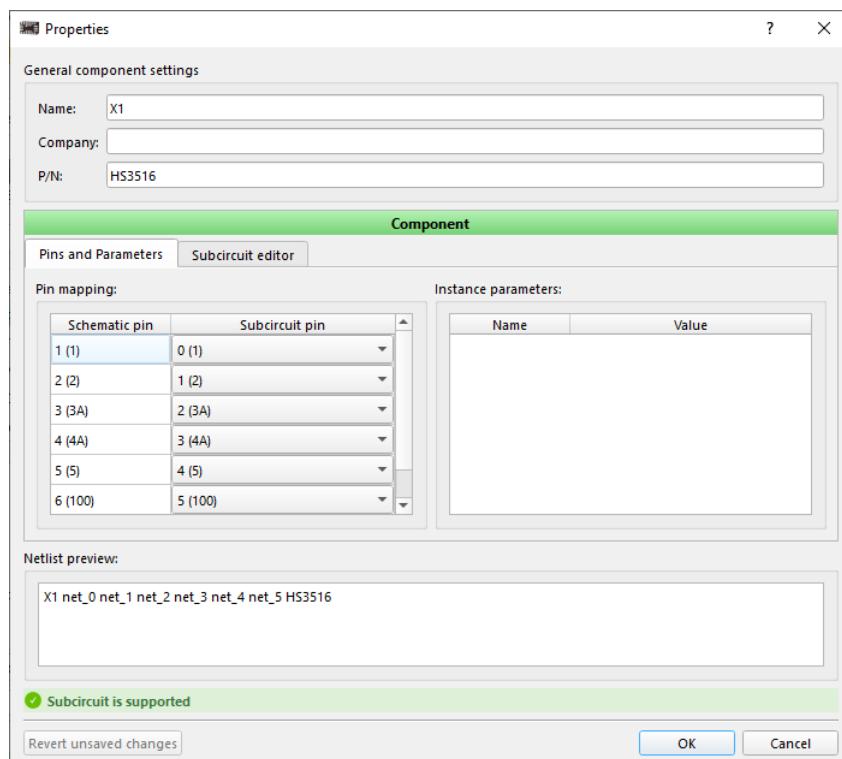
Importing a model containing multiple subcircuits

Model/Subcircuit parameters can be edited via the *Properties* dialog window. Double-click a component in the work area or click *Configure* in the *Properties* panel. The *Properties* window allows

configuring the external component's parameters and editing the model/subcircuit.



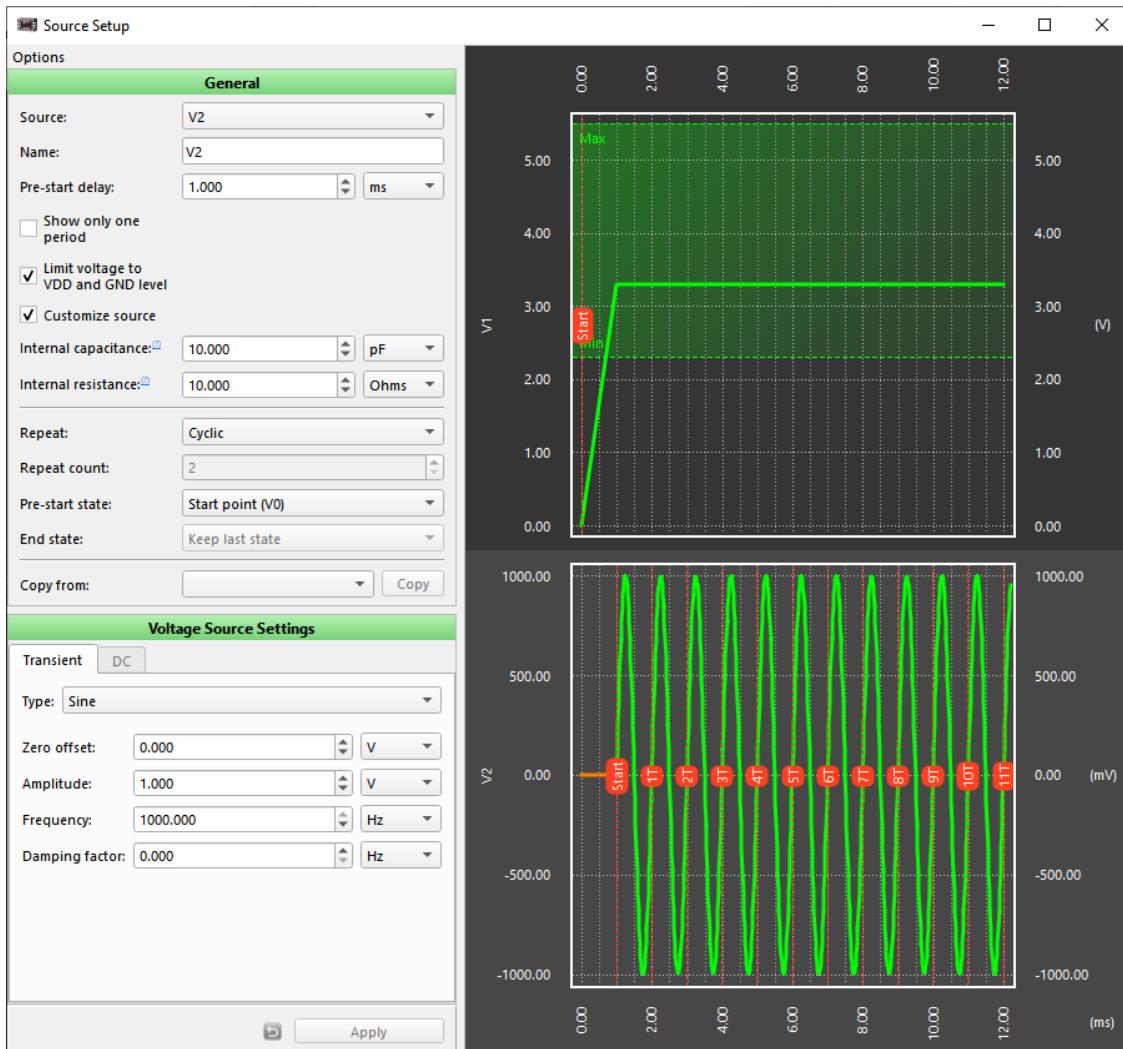
Model properties window



Subcircuit properties window

2.4.4 Source Setup

A voltage or current source can be configured in the *Source setup* window. To open the window, double-click a source or click *Configure* on the *Properties* panel.



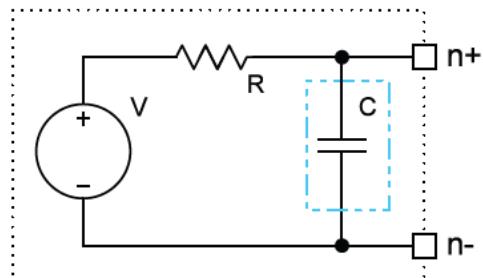
Source Setup Window

The set of settings and the window view differs depending on the simulation analysis type (read more about analysis types in section [2.4.7 Debugging Controls](#)). In the *Options* section, you can set the general and waveform-specific parameters related to the time intervals and the periods.

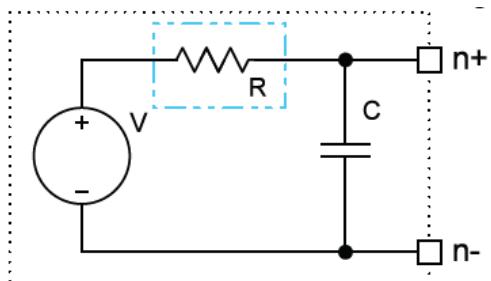
You can use the *Customize source* property to activate two additional parameters:

- *Internal capacitance* — the capacitance between the voltage source terminal and the ground.

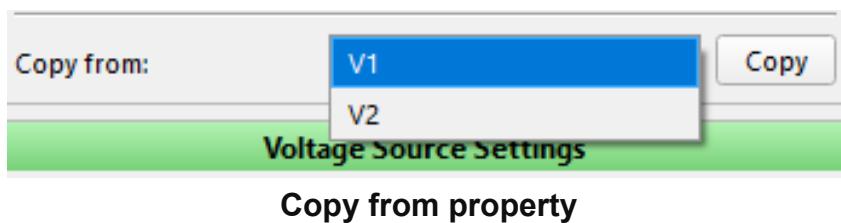
This value can be used to model a VDD bypass capacitor or IC pin parasitic capacitance



- *Internal resistance* — the output resistance of the generator. A non-ideal (real) voltage source is modeled with an ideal voltage source and resistance connected in series



If you need to duplicate the configured settings from one source to another, use the *Copy from* property.

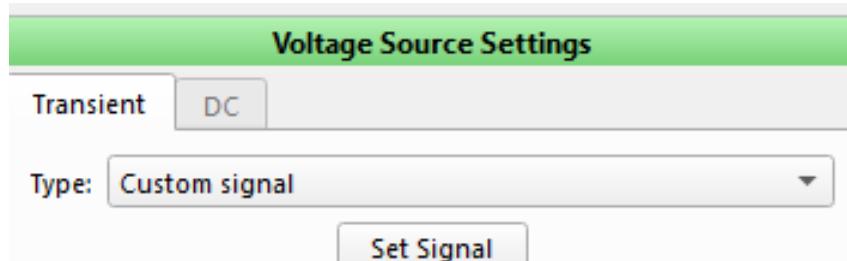


Copy from property

Custom Signal Wizard

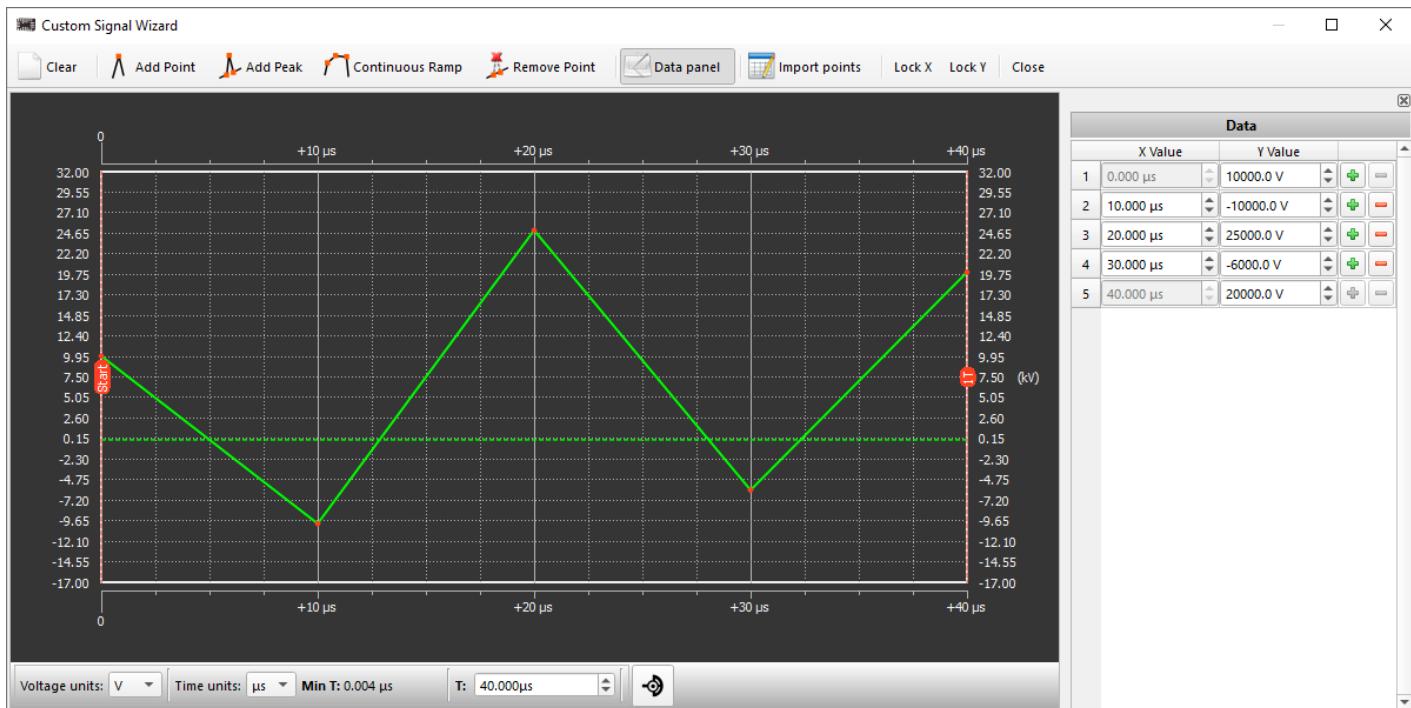
In addition to configuring the specific waveform types (e.g., DC, sine, trapeze, etc.), it is also possible to set a custom waveform shape. *Custom Signal Wizard* allows creating, importing, and editing the signal for the simulation voltage or current sources. The signal may be constructed by manually adding points or importing a custom list of points from an external source.

To open the *Custom Signal Wizard*, select *Custom signal* in the *Type* dropdown and click *Set Signal*.



Custom waveform option

The examples of possible signals are shown in the pictures below.



The standard editing process in Simulation Custom Signal Wizard

The *Custom Signal Wizard* allows importing a set of points in different formats. Signals of various complexity, duration, and amplitude are accepted. For signals consisting of large amounts of data,

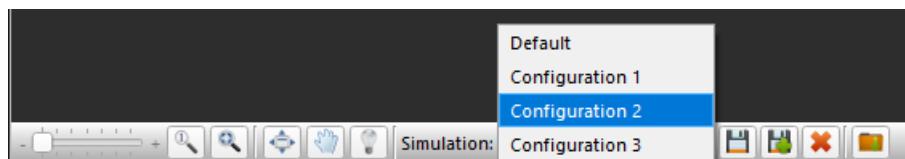
editing may be limited, and the simplified signal is shown. See the example below.



An imported waveform with a large number of points (above 1000)

2.4.5 Simulation configurations

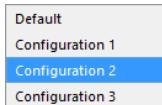
You can save the snapshots with the simulation configuration state and switch between them. The snapshot stores only the elements that belong to simulation. An asterisk next to its name indicates the modified state of the snapshot. You can also import the saved configuration state from another project for the same Part Number.



Simulation configuration

- Save the current configuration state
- Save a new configuration state
- Delete the selected configuration
- Import new configuration

The list of saved configurations



2.4.6 Probes

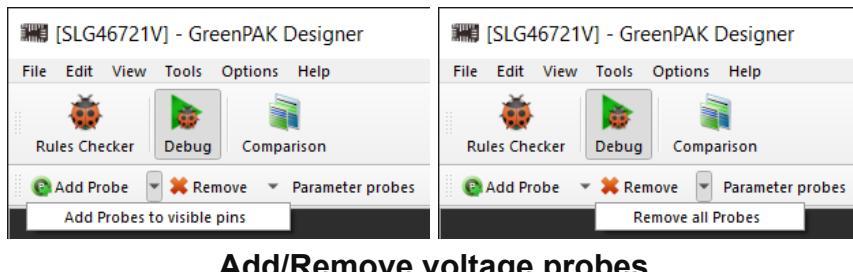
The probes provide you with the data that components yield during the simulation. You can attach a probe to a pin to reflect the component parameters. The active probes remain on the hidden

components, and their data will be displayed in the simulation results.

Two types of probes are available, namely voltage and parametric probes.

Voltage probes

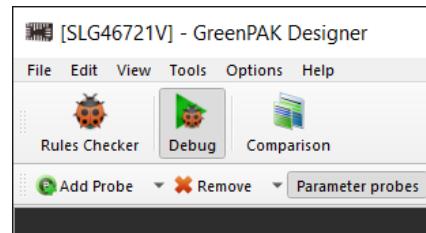
The voltage probe is used to capture an output signal from a pin. You can add a probe to the external component terminals and macrocell output pins. Click the respective buttons on the toolbar and then the pin to add or remove a probe. It is possible to add the probes to all available pins or remove all of them. In one click (see the arrow next to the add/remove button). You can also delete the probe from the context menu of a probe icon.



Add/Remove voltage probes

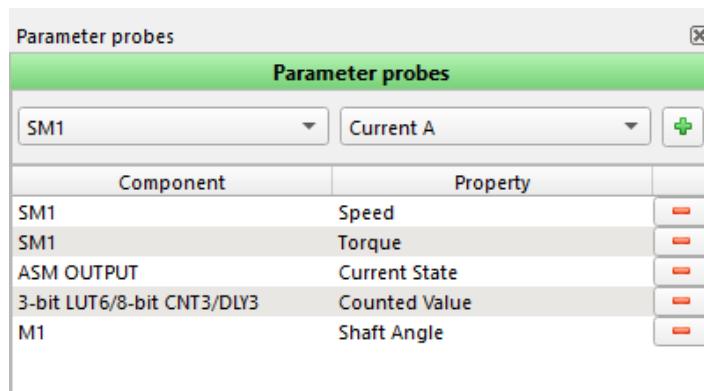
Parametric probes

The parametric probe allows monitoring the macrocell parameters in simulation, e.g., the counted value in the CNT/DLY blocks. To show the list of available probes, click the *Parametric probes* button on the toolbar.



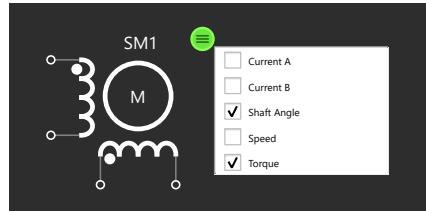
Active parametric probes button

Add a probe by selecting a component and a property in the dropdowns on the respective panel.



Parameter probes window

Right-click the parametric probe sticker to facilitate adding/removing a probe and editing the probe parameters.



Parametric probe sticker

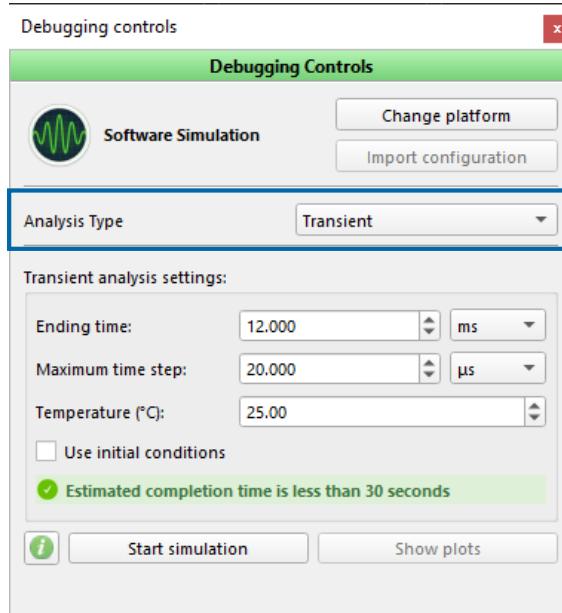
2.4.7 Debugging Controls

The *Debugging Controls* panel appears once *Software Simulation* is selected in *Development Platform Selector*.

Choose between the analysis types before starting the simulation process. There are two types of simulation analysis: *Transient* and *Parametric DC*.

Transient analysis

Transient Analysis settings define the time span for simulation, maximum time step, source voltage, and temperature.

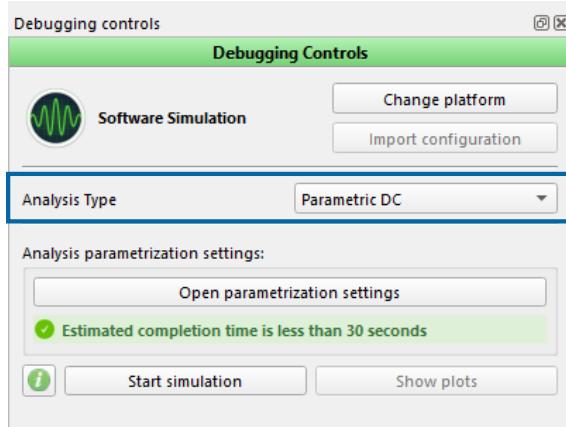


Transient Analysis Debugging controls

Use *initial conditions* is an optional checkbox that allows setting an initial charge different from the default value (0) for components like a capacitor or inductor. Also, this checkbox enables setting all node voltages in a circuit to 0V at the initial time point.

Parametric DC Analysis

You can customize the *Schematic library* objects' properties by using the *Parametric DC* analysis. The parameter sweep can be configured in the *Parametric DC analysis* parametrization window. Click *Open parametrization settings* on the *Debugging Controls* to activate it.

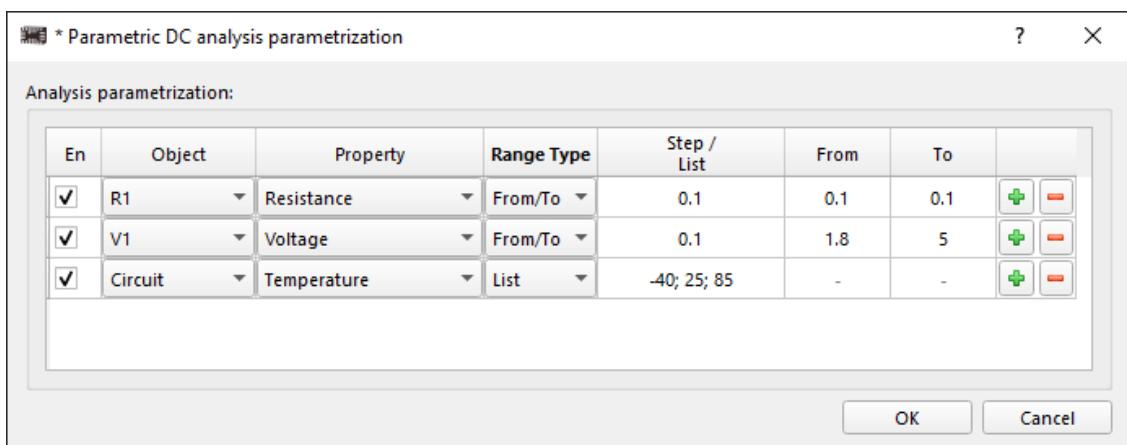


Parametric DC Analysis Debugging controls

Analysis parametrization can set active parameters for external components, circuit temperature, and macrocell properties (e.g., *Resistance* (*initial data*) for *Digital Rheostats*).

Parametric DC data has two range types:

- *From/To* — simulate data between the set *From* and *To*, incrementing by *Step*
- *List* — use data in a semicolon-separated list



Parametric DC Analysis Window

Estimated completion time

It is possible to define simulation runs that may exceed the available resources on your computer, which can potentially make your computer unstable. Longer runtimes require greater resources on your computer, including CPU and memory.

The software estimates the runtime based on sample points in three broad categories: *green*, *yellow*, and *red*. The estimated runtime may vary depending on different factors (e.g., the computer's performance).

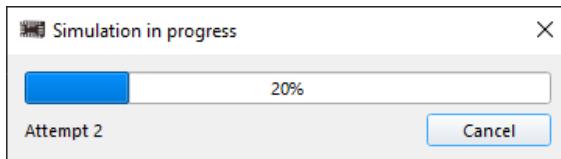
Once the required parameters are set, you can start the simulation by clicking the respective button. The *Simulation results* window appears after the simulation process ends.

2.4.8 Simulation progress

The *Simulation in progress* window appears while processing data. You can interrupt the process by clicking *Cancel*. The simulation results will still be displayed based on data that the tool managed to process before it was canceled.

Scheme issue resolution

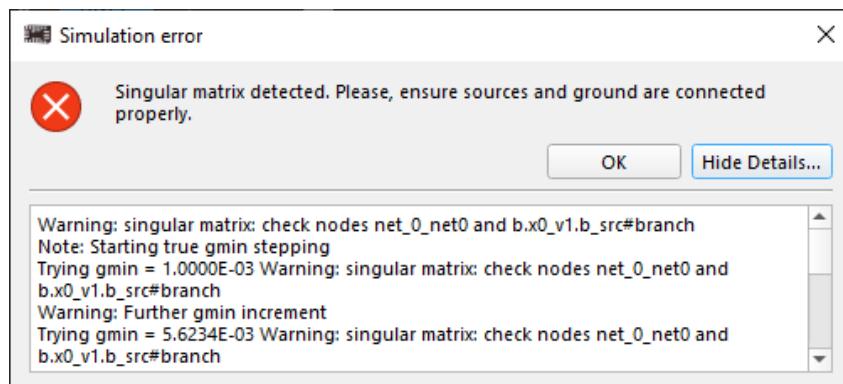
If a simulation attempt encounters a known simulation error of the scheme convergence, the simulation internal algorithm attempts to adjust the circuit by changing the component properties or scheme parameters and re-init the simulation. In this case, the *Simulation in progress* window indicates the current attempt count.



Simulation in progress (2nd attempt)

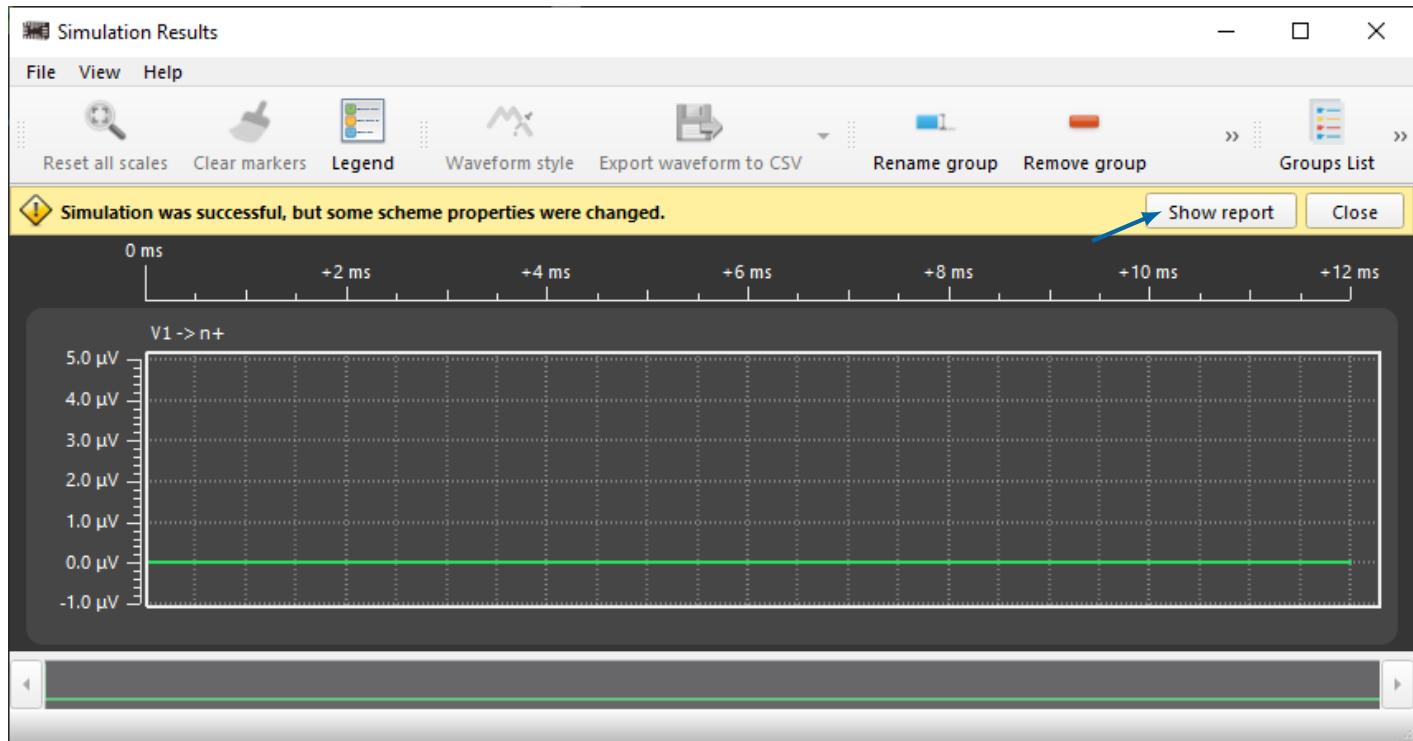
The simulation error is shown only when the algorithm cannot resolve the problem.

If the simulation process fails, the corresponding error window is shown. It may contain a brief description and details from the simulation engine about the cause.

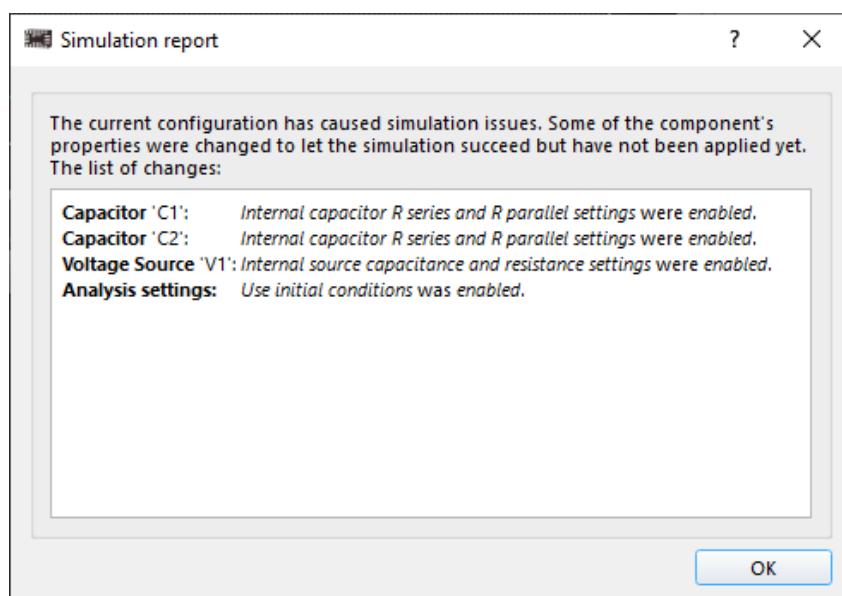


Simulation error window

After the issues are resolved and the simulation process is completed, you can observe the simulation report by clicking the *Show report* button on the *Simulation Results* window.



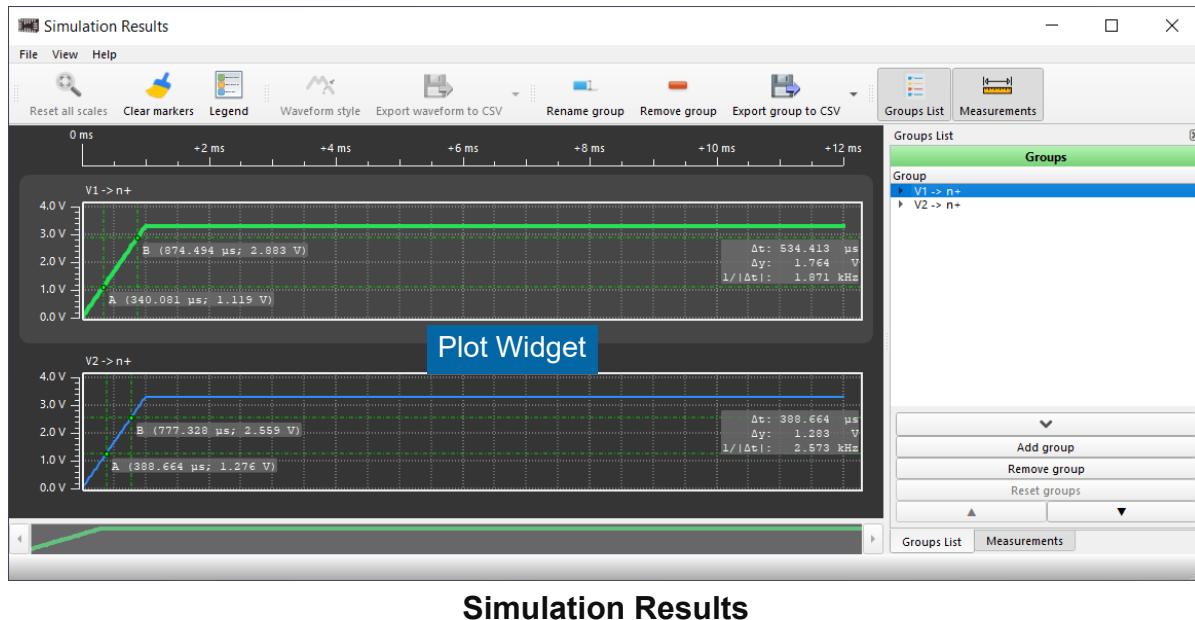
Show report button



Simulation report window

2.4.9 Simulation Results window

The *Simulation Results* window opens after the completed/interrupted simulation process. If you close the window, you can reopen the latest simulation results by clicking *Show plots* on the *Debugging Controls* panel.



Simulation Results

The *Simulation Results* window contains the following sections:

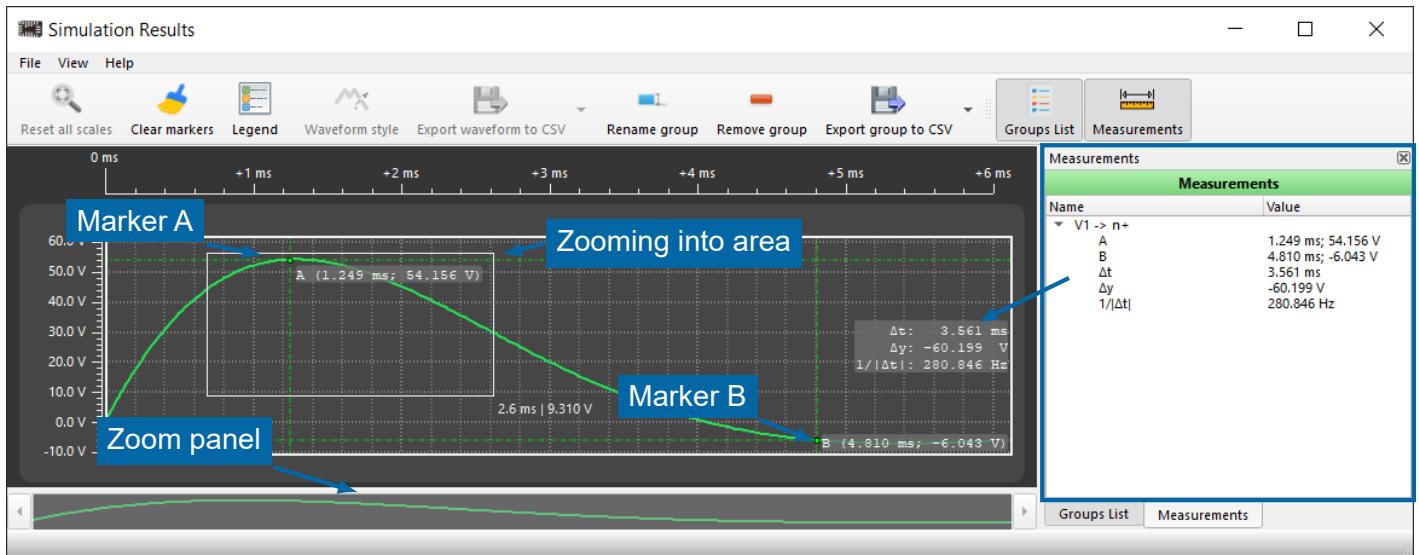
Toolbar

Toolbar provides operations with the plot area, waveform style, group management, and results data export. You can export a single waveform or a whole group to a .csv, .vcf, or .png file. You can also export all waveforms as a single .png file.

Plot widget

The *Plot* widget is the main area displaying a waveform or group of waveforms. It is possible to manipulate the plot controls in several ways. In addition to the toolbar controls, right-click the plot to open the context menu and access the available actions. Also, see the [keyboard commands](#), which can be used instead.

You can set the markers with a *Ctrl* + right and left click to see the signal information in the particular point. See the examples of markers and Δt and ΔV parameters on the *Plot* widget:

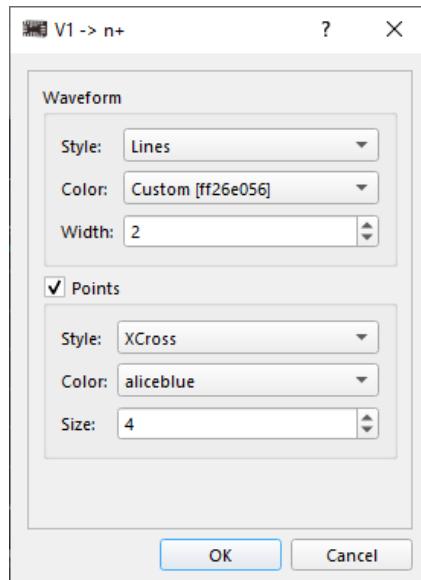


Plot widget and Measurements

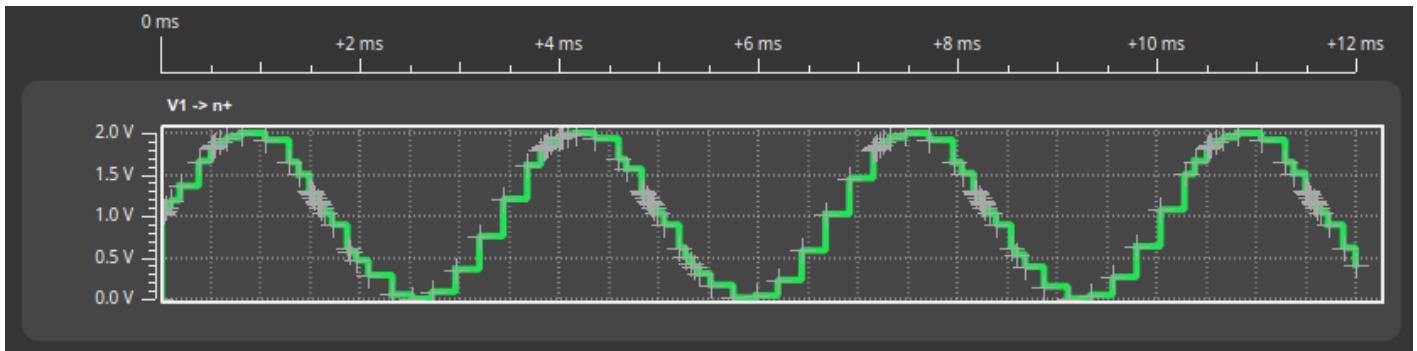
Groups List

The *Groups List* panel contains the list of all captured signals and source generators in groups.

Click the waveform on the panel to modify its style, color, or width by choosing the *Waveform style* option from the toolbar or context menu. Also, use the context menu to rename the group.

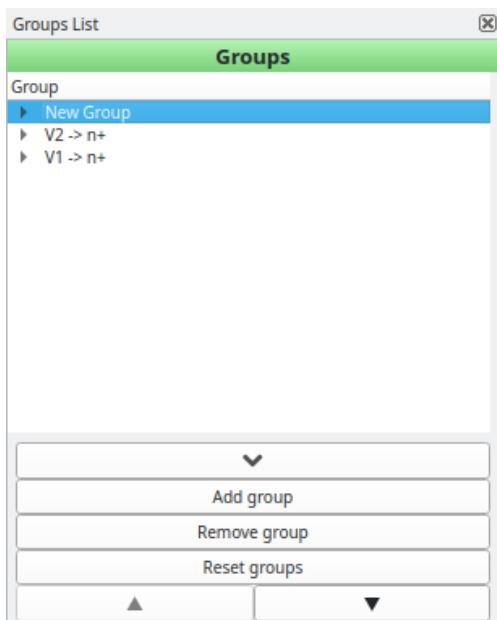


Waveform Plot Configuration window



A sine simulated with huge interval settings

The *Groups List* bottom controls let you operate the waveforms.

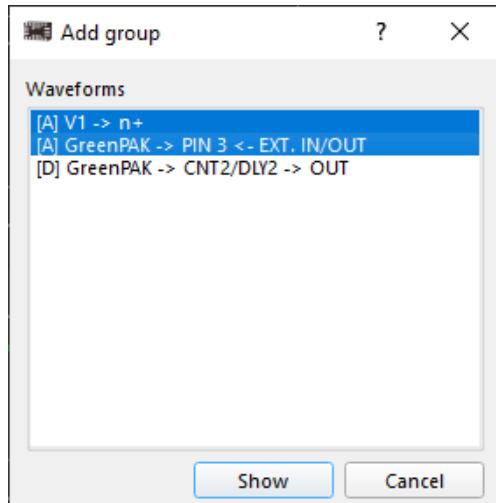


Groups List

Group controls include:

- ▾ / ^ — fold/unfold the controls
- Add group — create a group by selecting one or more signals from the list. The signal(s) can be combined in a group of *analog [A]*, *digital [D]*, or *parameter probe [B]* waveforms. You can create

multiple groups with the same waveform reused.

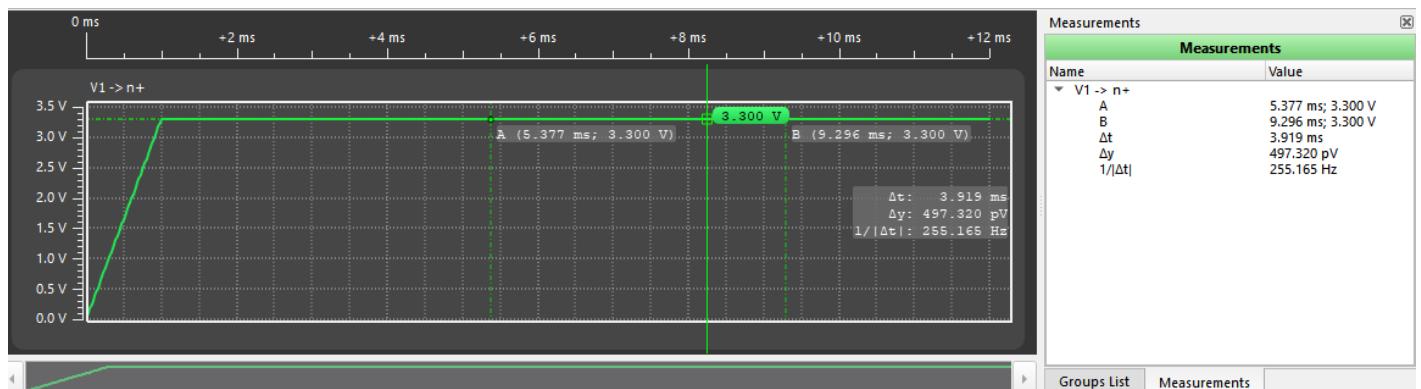


Add group menu

- Remove group — delete the selected group
- ▼ / ▲ — move a selected group up or down
- Reset groups — restore default and remove all added groups

Measurements

This panel displays measurement data for the added plot markers.



Measurements panel

3 Devices

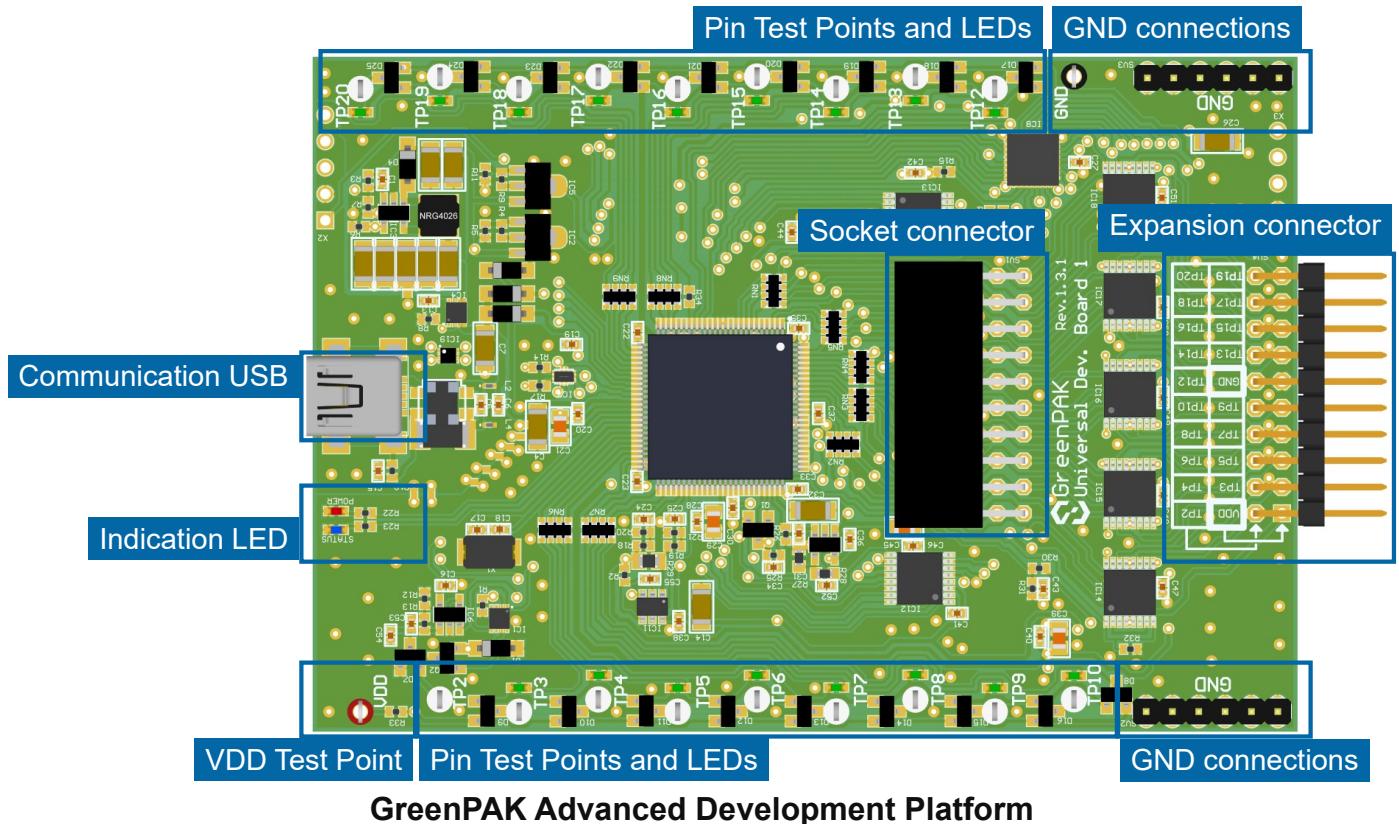
To provide communication between the chip and the software the specific hardware is required. You can refer to the *Hub* window → *Development* tab → *Details* section to check the hardware support information for a certain Part Number.

The Go Configure Software Hub allows the project debugging using the hardware platforms. Later in this chapter you can read about the supported development platforms along with their software representation.

3.1 GreenPAK devices

3.1.1 GreenPAK Advanced Development Platform

The *Advanced Development Platform* provides programming, emulation, and testing functions for GreenPAK devices. The board is compatible with 20-pin socket adapters.



The board has a USB communications interface that uses the USB mini-B connector. The USB power line is the main power source.

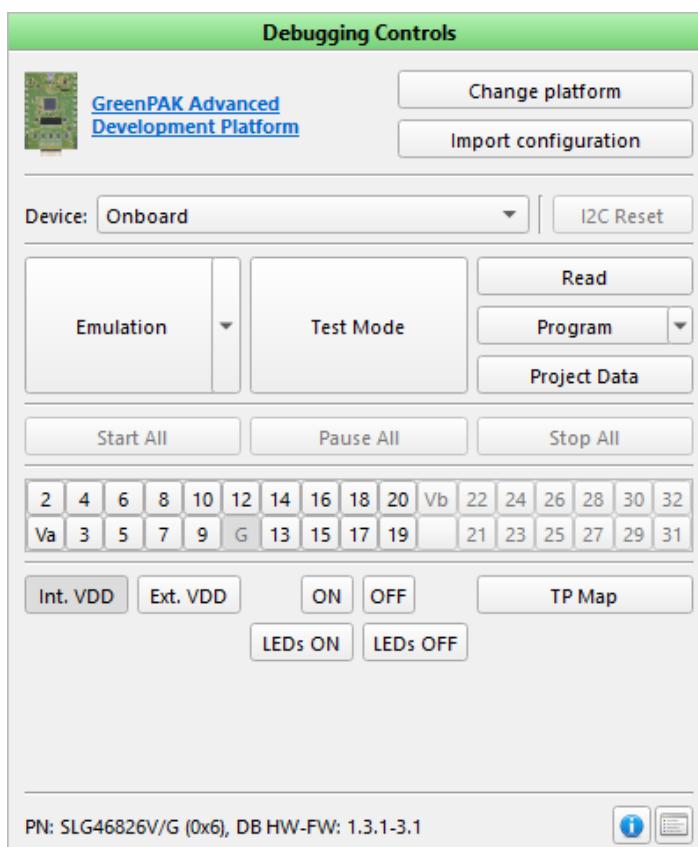
Each GreenPAK chip pin (including VDD) has its own observation test point. These test points are designed for observation of signals on the pins only. To connect an external signal source, use a software-controlled expansion connector.

Note: Do not try to connect external power/signal sources to the test points. It will affect GreenPAK Advanced Development Board functionality or may even damage it.

You can connect all the Test Points except Test Points 1, 2, and 11 to buffered LEDs. This option allows visualization of digital levels on chip pins. The *Advanced Development Board* supports connecting five types of loads and signal sources. Each source has its own particular purpose. A signal generator connection is available for VDD pins. For the communication pins, you can use the following connections: VDD, GND, Pull-up, Pull-down, Configurable Button.

The *GreenPAK Advanced Development Platform Debugging Controls* User Interface includes the following sections:

Debug Configuration	Read	Power source selector
Device selector	Program	TP map
Emulation	Project data window	LEDs ON and LEDs OFF
Emulation(sync)	Generator controls	Info details
Test Mode	Expansion connectors	

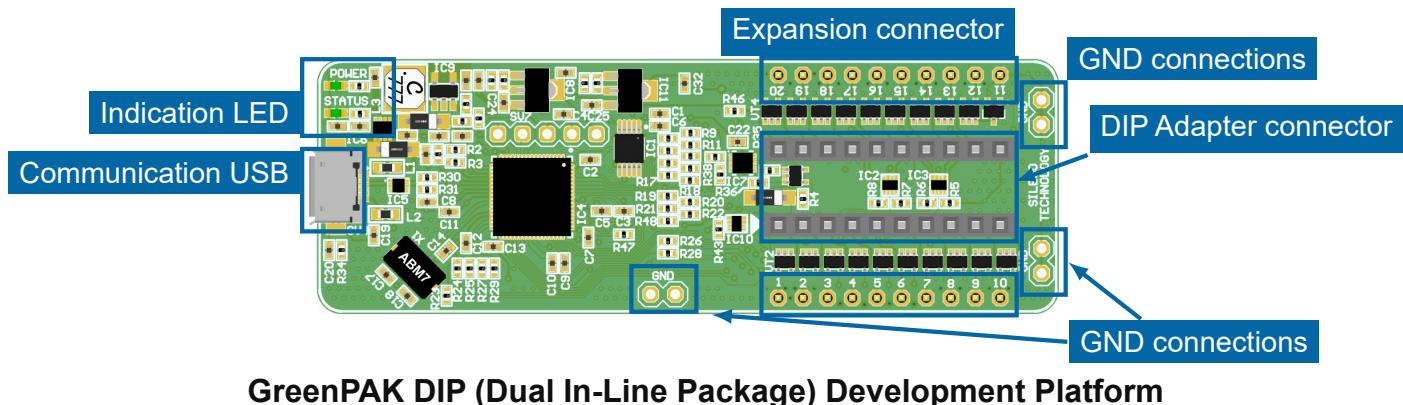


GreenPAK Advanced Development Platform controls

Find a comparison of the controls available on different GreenPAK boards in the appendix, section 5.3 Debugging Controls feature availability.

3.1.2 GreenPAK DIP Development Platform

The *GreenPAK DIP (Dual In-Line Package) Development Platform* is perfect for breadboarding and fast prototyping. It provides programming, emulation, and testing functions for *GreenPAK* devices. The *GreenPAK DIP Development Board* is compatible with the *DIP Proto Board*. The power source of the *GreenPAK DIP Development Board* is the USB power line. The board has a USB communications interface that uses the USB mini-B connector. This interface communicates with the software control tool and supplies power to the board.

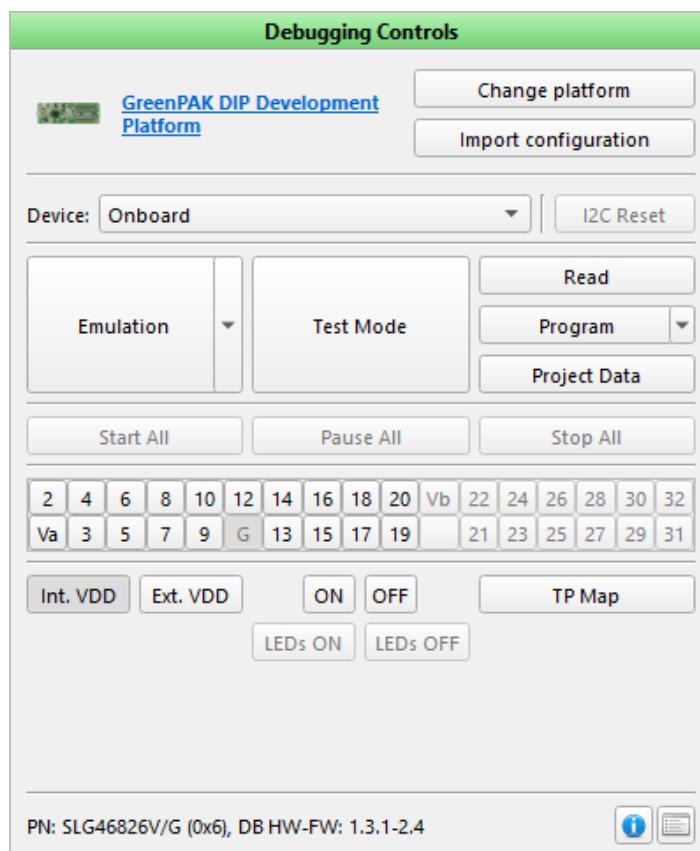


All pins of the *Expansion Connector*, except GND, are controlled simultaneously from the software: when one key is on or off, all others are also turned on or off, respectively.

For the chips that have two VDDs, VDD is always equal to VDD2.

The *GreenPAK DIP Development Platform Debugging Controls* User Interface includes the following sections:

Debug Configuration	Test Mode	Expansion connectors
Device selector	Read	Power source selector
I2C Reset	Program	TP map
Emulation	Project data window	LEDs ON and LEDs OFF
Emulation(sync)	Generator controls	Info details

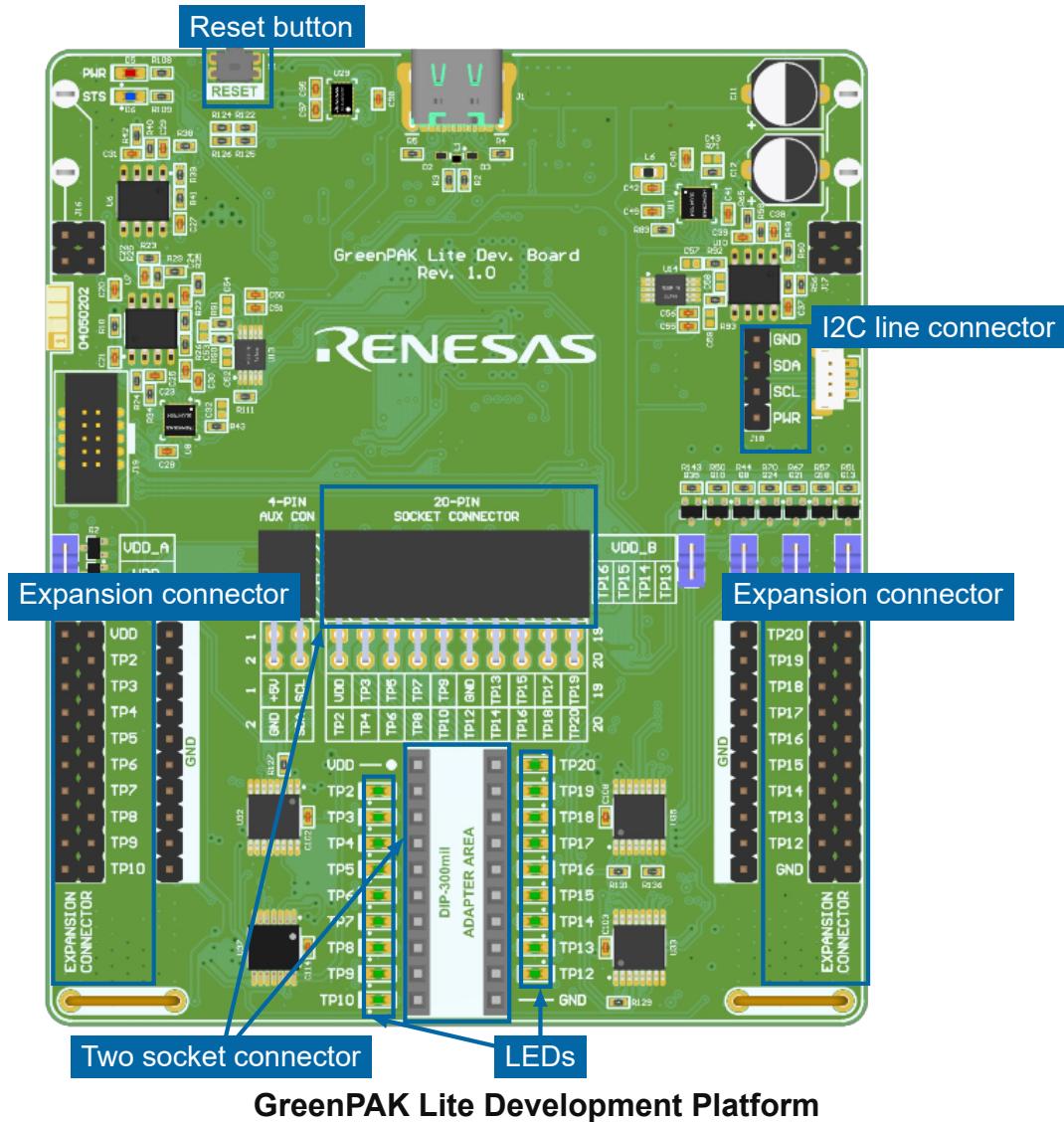


GreenPAK DIP Development Platform controls

Find a comparison of the controls available on different GreenPAK boards in the appendix, section 5.3 Debugging Controls feature availability.

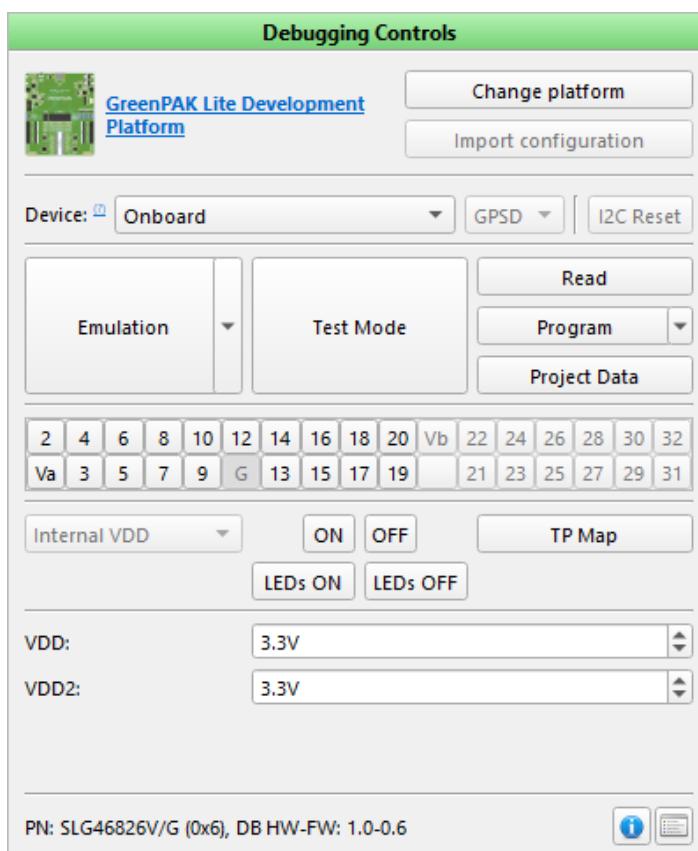
3.1.3 GreenPAK Lite Development Platform

GreenPAK Lite Development Board provides a complete set of tools to work with external chips and socket connectors of two types. The board is compatible with 20-pin socket adapters as well as with DIP Proto Board.



The *GreenPAK Lite Development Platform Debugging Controls* User Interface includes the following sections:

Debug Configuration	Test Mode	TP map
Device selector	Read	LEDs ON and LEDs OFF
External device modes	Program	Voltage level controls
I2C Reset	Project data window	Info details
Emulation	Expansion connectors	
Emulation(sync)	Power source selector	



GreenPAK Lite Development Platform controls

Find a comparison of the controls available on different GreenPAK boards in the appendix, section 5.3 Debugging Controls feature availability.

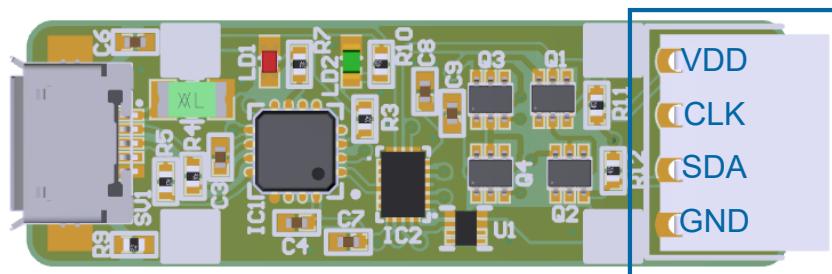
3.1.4 GreenPAK Serial Debugger

GreenPAK Serial Debugger is a solution that you can use for external chip debugging when a chip is soldered on a PCB. The *GreenPAK Serial Debugger* device works for programming GreenPAK products with multiple-time programming (MTP) Non-Volatile Memory (NVM) or for configuring the interconnect logic, the IOs, and the macrocells of all GreenPAK chips with the I2C interface.

The device runs as an I2C Master. Chip programming, emulation, and debugging are done through the I2C protocol via two chip pins: SCL and SDA. The Sync button enables immediate updating of the chip's RAM register with any change applied to the project.

The connected chip can be powered from the board, when it's connected to USB, or from an external power. The voltage level on the I2C bus can be set by a VDD drop-down in the *Debugging controls*.

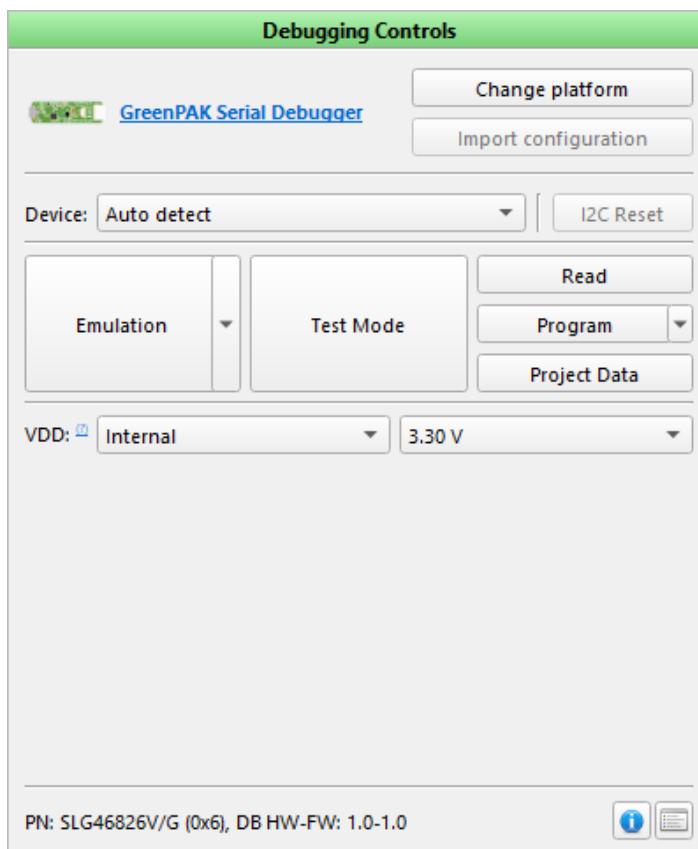
When the *GreenPAK Serial Debugger* is connected to the USB port, the Power LED turns on. You can power a chip from the board connected to a USB or from an external power. The voltage level on the I2C bus can be set by a VDD drop-down in the *Debugging controls*. When the *GreenPAK Serial Debugger* is connected to the USB port, the Power LED turns on. A programmed chip can be powered from the *GreenPAK Serial Debugger* or from the external power supply. The board detects the external power and switches off its VDD line (in this case, I2C pull-up resistors are connected to the external power supply).



GreenPAK Serial Debugger external connectors

The *GreenPAK Serial Debugger Platform Debugging Controls* User Interface includes the following sections:

Debug Configuration	Emulation(sync)	Project data window
Device selector	Test Mode	Power source selector
I2C Reset	Read	Voltage level controls
Emulation	Program	Info details



GreenPAK Serial Debugger Platform controls

Find a comparison of the controls available on different GreenPAK boards in the appendix, section 5.3 Debugging Controls feature availability.

3.1.5 Connecting external GreenPAK

Advanced, DIP, or Lite platforms

- Connect the socket with an external chip to a *Development Board* via pins:
 - Corresponding *I2C* pins (*SCL*, *SDA*)
 - *VDD* pin
 - *GND* pin

For the *DIP/Advanced/Lite Development Board*: connect a socket with an external chip to the *Expansion Connector* pins:

- Disconnect the *Onboard* chip to proceed with the external chip
- Start the *Go Configure Software Hub* and select the Part Number of the connected external chip
- Start the *Debug tool* and select the *Development Platform*
- Select the *Device address* (used by default for an empty chip) or the corresponding address programmed to the chip **Note:** If an external socket is connected to a development board correctly and the proper *Device address* is selected, a chip is detected either after clicking *Update chip info* or automatically after starting any chip operation

Once all steps are completed, the debugging controls become active.

Note: Ext. *VDD* (*V_a*) expansion can be disabled if the voltage is applied to an external *VDD* port. The *expansion connector* will automatically disconnect, and a warning message will be displayed if *Emulation/Test mode* operations for a chip with applied external voltage are running.

Serial Debugger platform

- Connect a socket with an external chip to a *Development Board* via ECs:
 - Corresponding *I2C* pins (*SCL*, *SDA*)
 - *VDD* pin
 - *GND* pin
- Connect a chip with wires to the *Serial Debugger* pins (*VDD*, *CLK*, *SDA*, *GND*):
- Start *Go Configure Software Hub* and select the Part Number of the connected external chip
- Start the *Debug tool* and select the *Serial Debugger platform*
- Select *0001b* for the *Device address* (used by default for an empty chip) or a corresponding address programmed to the chip
- Specify *VDD* configuration:

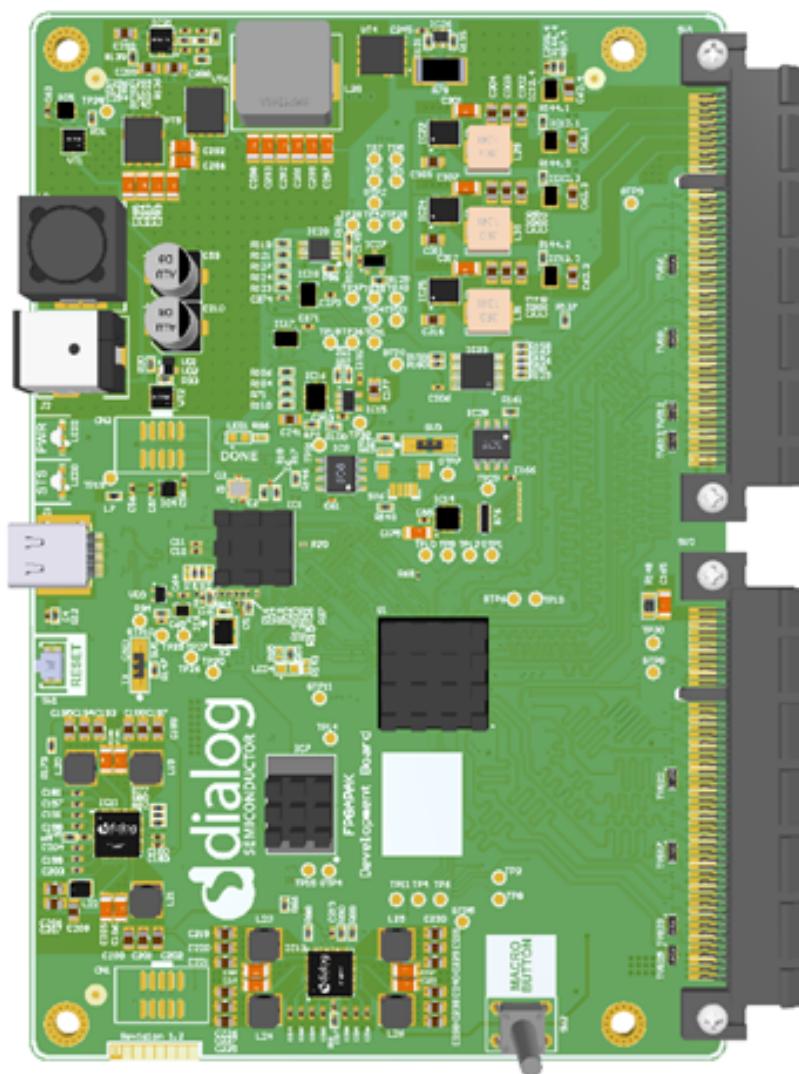
- *Internal* — chip is powered from a *Serial Debugger* board
- *External* — chip is powered from an external power source

Once all steps are completed, the debugging controls become active.

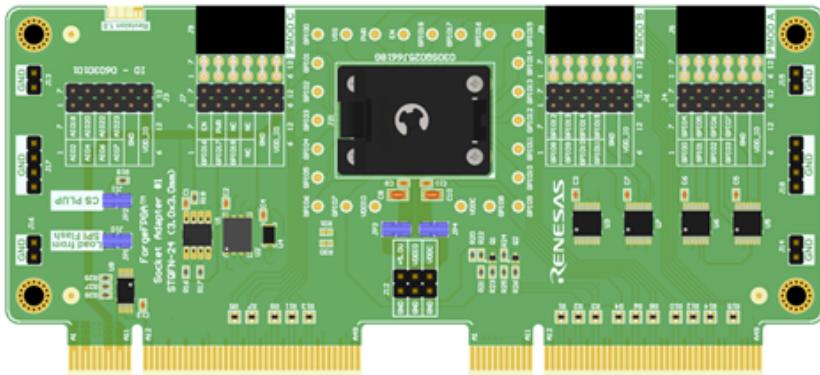
3.2 ForgeFPGA devices

3.2.1 ForgeFPGA Advanced Development Platform

The *FPGA Advanced Development Board* is a multi-functional tool that allows you to develop FPGA designs by providing onboard power source, digital and analog signal generation, and logic analysis capabilities. The platform can connect additional external boards called socket adapters. The function of the socket adapter board is to implement a stable electrical connection between the pins of the chip under test and the *FPGA Development Board*. To implement this, the platform has a Dual PCIe connector. This connector has 40 differential pairs (80 digital channels), 32 analog pins, service pins, and power pins. A dual PCIe connector is universal and can be applied to multiple socket adapter boards.

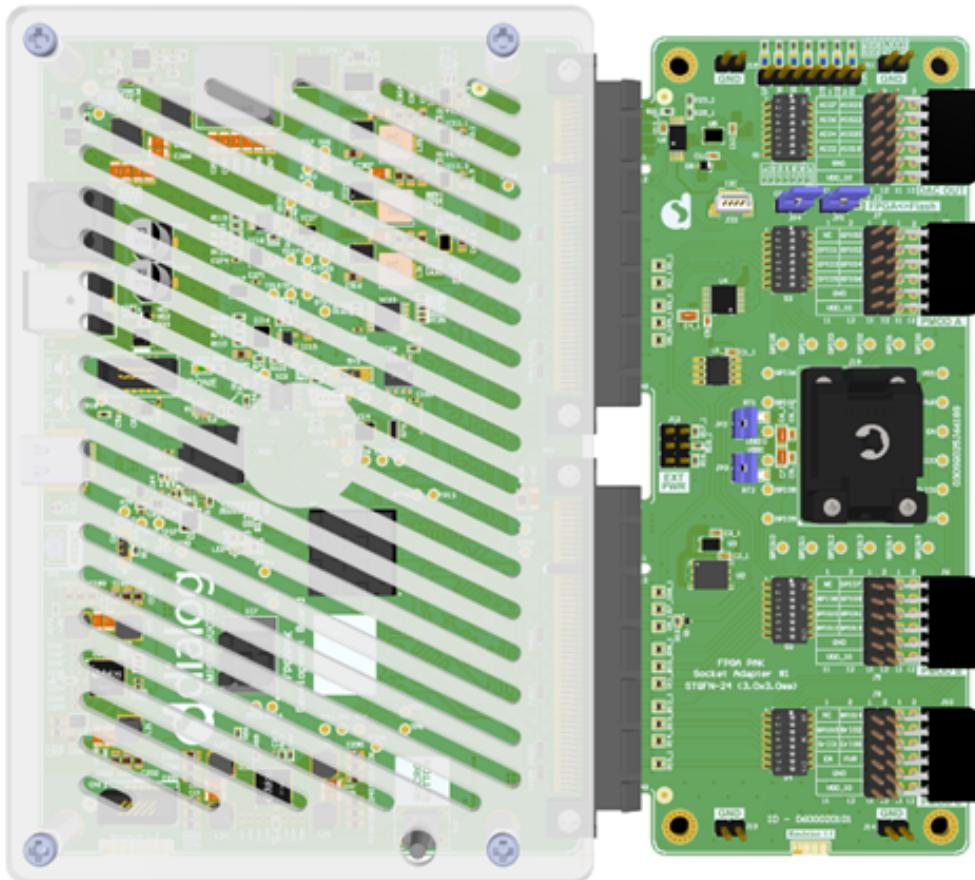


ForgeFPGA Advanced Development Platform



ForgeFPGA socket adapter

Also, you can use the board as an independent unit. The chip can be powered through the EXT PWR connector and signals can be read through the through-hole 12-pin connectors (PMOD connectors).

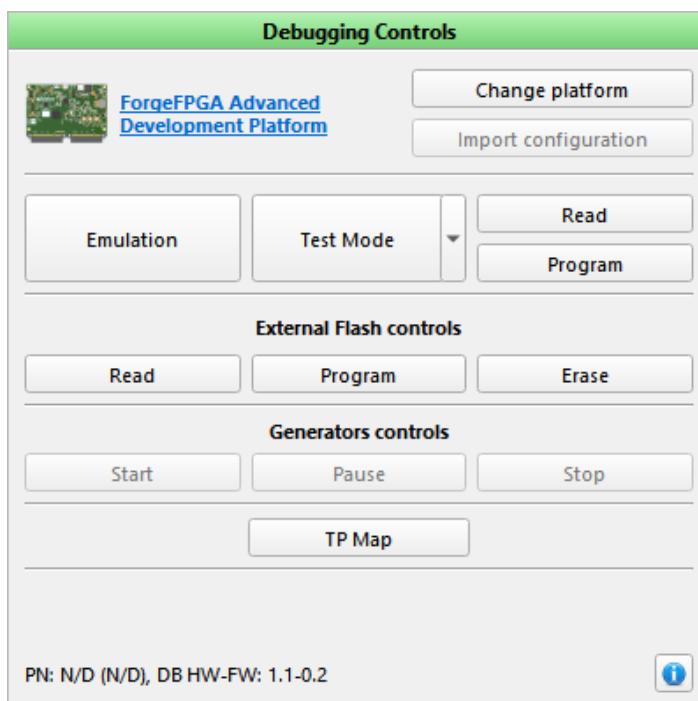


Assembled equipment for working with a chip

To start working with the FPGA devices you need to connect the platform to the computer via USB Type-C Connector and the power cord to the power supply connection on the board. If all the connections are correct, then the red LED(PWR) and blue LED(STS) light up.

The *ForgeFPGA Advanced Development Platform Debugging Controls* User Interface includes the following sections:

Debug Configuration	Program	Erase (flash)
Emulation	Test Mode* (flash)	Generator controls
Test Mode	Read (flash)	TP map
Read	Program (flash)	Info details



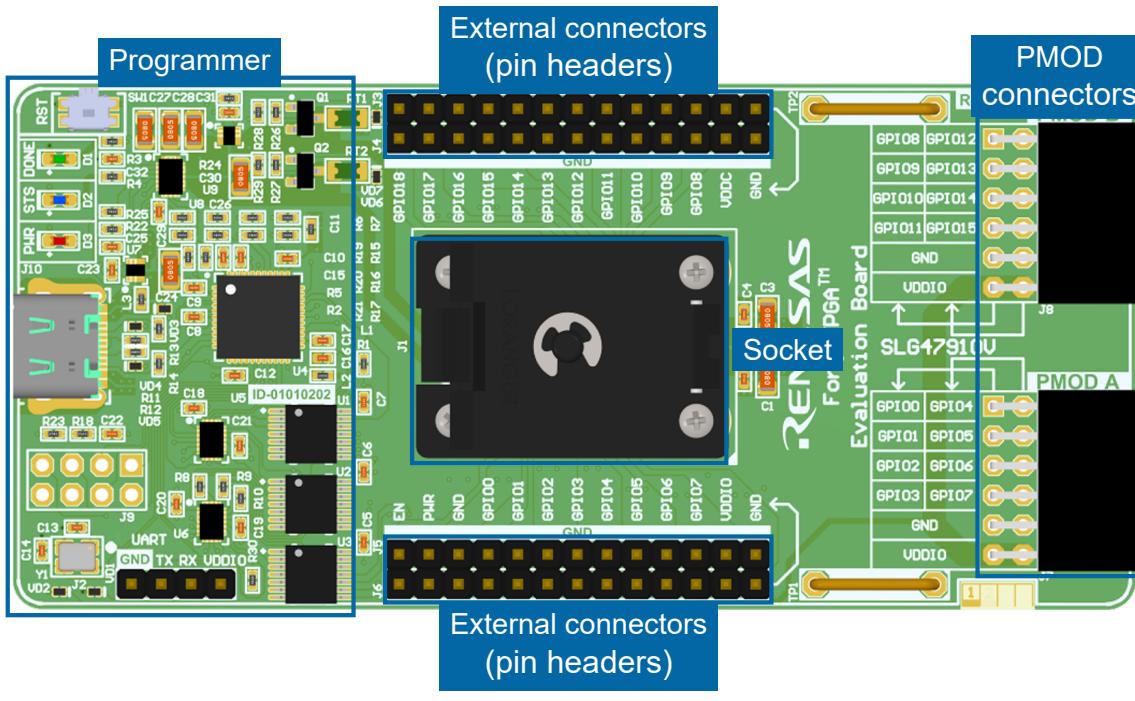
ForgeFPGA Advanced Development Platform controls

Find a comparison of the controls available on different FPGA boards in the appendix, section [5.3 Debugging Controls feature availability](#).

3.2.2 ForgeFPGA Evaluation board

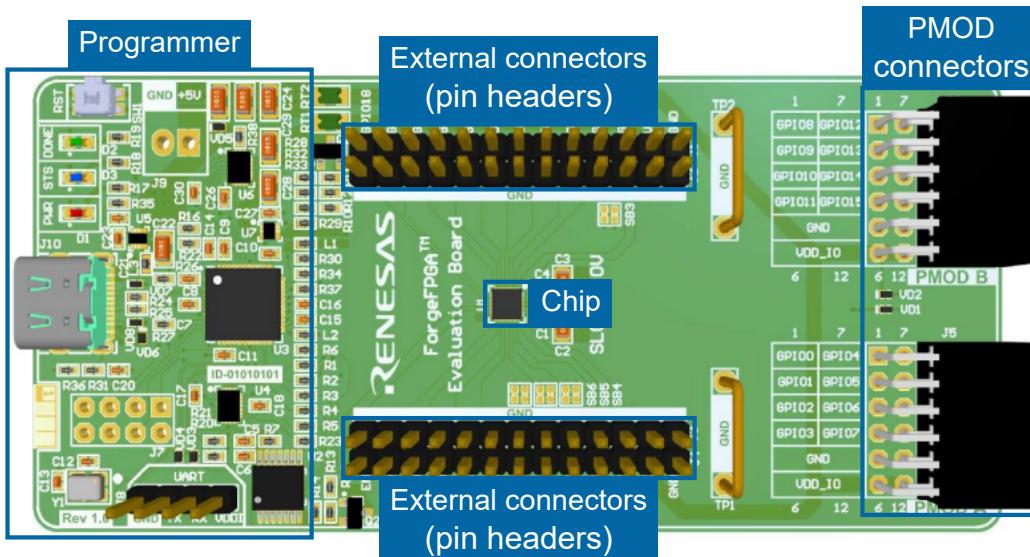
ForgeFPGA Evaluation Board is a compact, easy-to-use, USB-powered hardware tool. There are two variations of this platform: version 2.0 and 1.0.

ForgeFPGA Evaluation Board v.2.0 provides SLG47910 IC hardware support for design emulation, programming, internal UART terminal options, and real-time testing. The platform mainly consists of the following blocks – programmer, socket, GPIO external connectors, and PMOD connectors. This board uses a USB Type-C connector for communications and power supply.



ForgeFPGA Evaluation Board 2.0

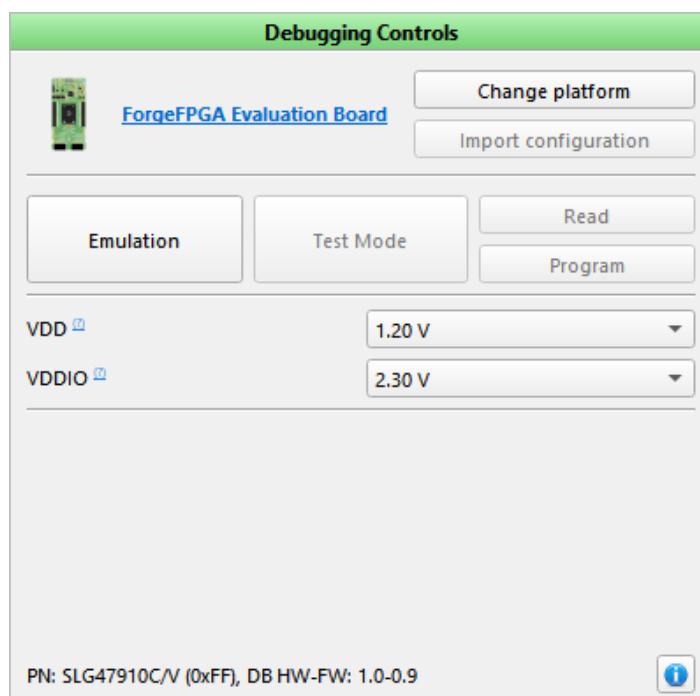
ForgeFPGA Evaluation Board v.1.0 is a simplified version of the platform and, therefore, has limited functionality. Since the socket is absent, the SLG47910 IC is soldered directly on the board. The platform provides emulation possibilities along with access to the UART terminal.



ForgeFPGA Evaluation Board 1.0

The *ForgeFPGA Evaluation Board Debugging Controls* User Interface includes the following sections:

Debug Configuration	Program
Emulation	Voltage level controls
Test Mode	Info details
Read	



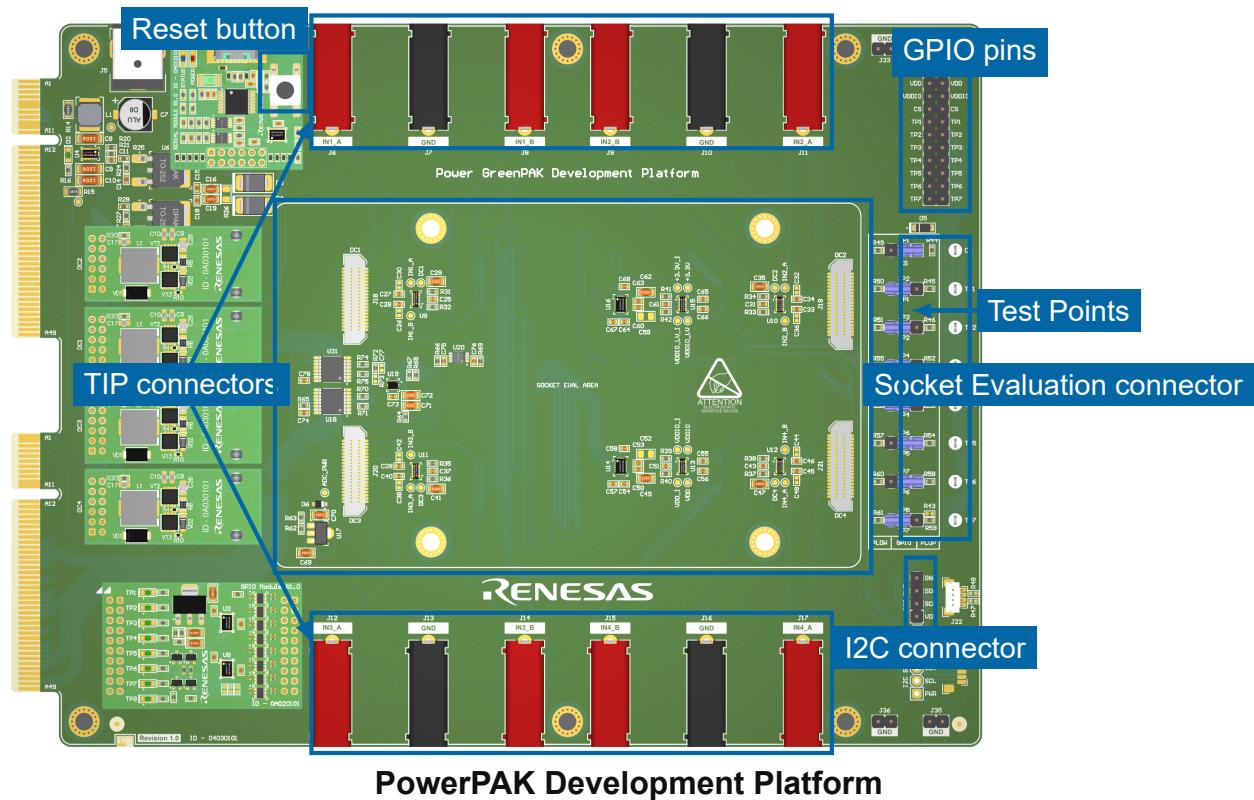
ForgeFPGA Evaluation Board controls

Find a comparison of the controls available on different FPGA boards in the appendix, section [5.3 Debugging Controls feature availability](#).

3.3 PowerPAK Devices

3.3.1 PowerPAK Development Platform

The *PowerPAK Development Platform* provides full debugging capabilities for the PowerPAK chips family. The Development Board has modules to power the IC, GPIOs control, and configurable voltage sources.



The board has a USB communications interface that uses the USB type-C connector. Also an external 12v power supply to power up the platform is required.

The *PowerPAK Development Platform Debugging Controls* User Interface includes the following sections:

Debug Configuration

Test Mode

PowerPAK voltage controls

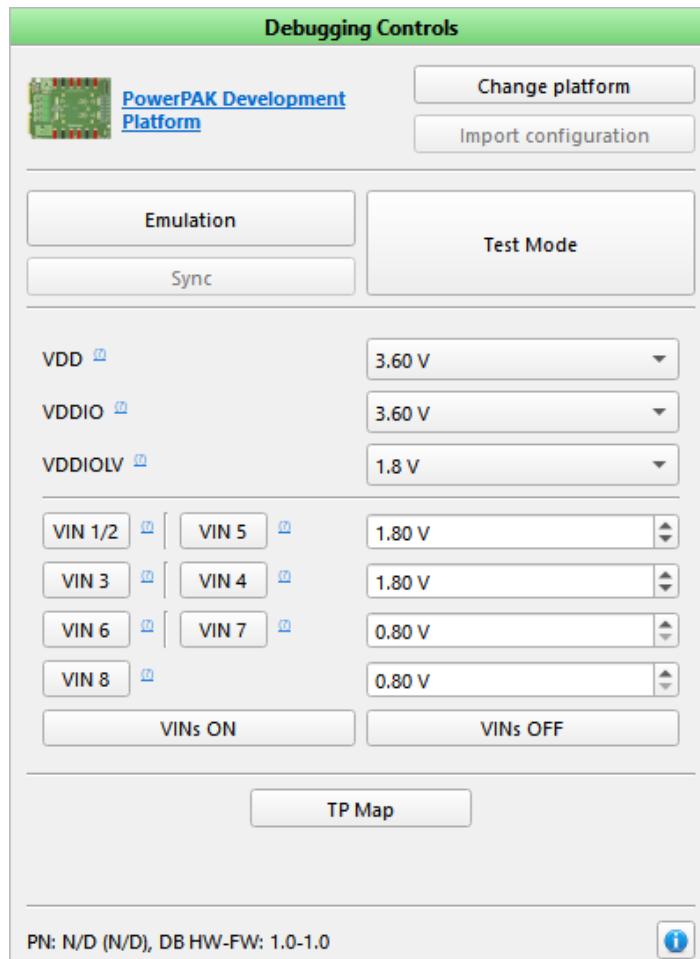
Emulation

Voltage level controls

Info details

Sync

TP map

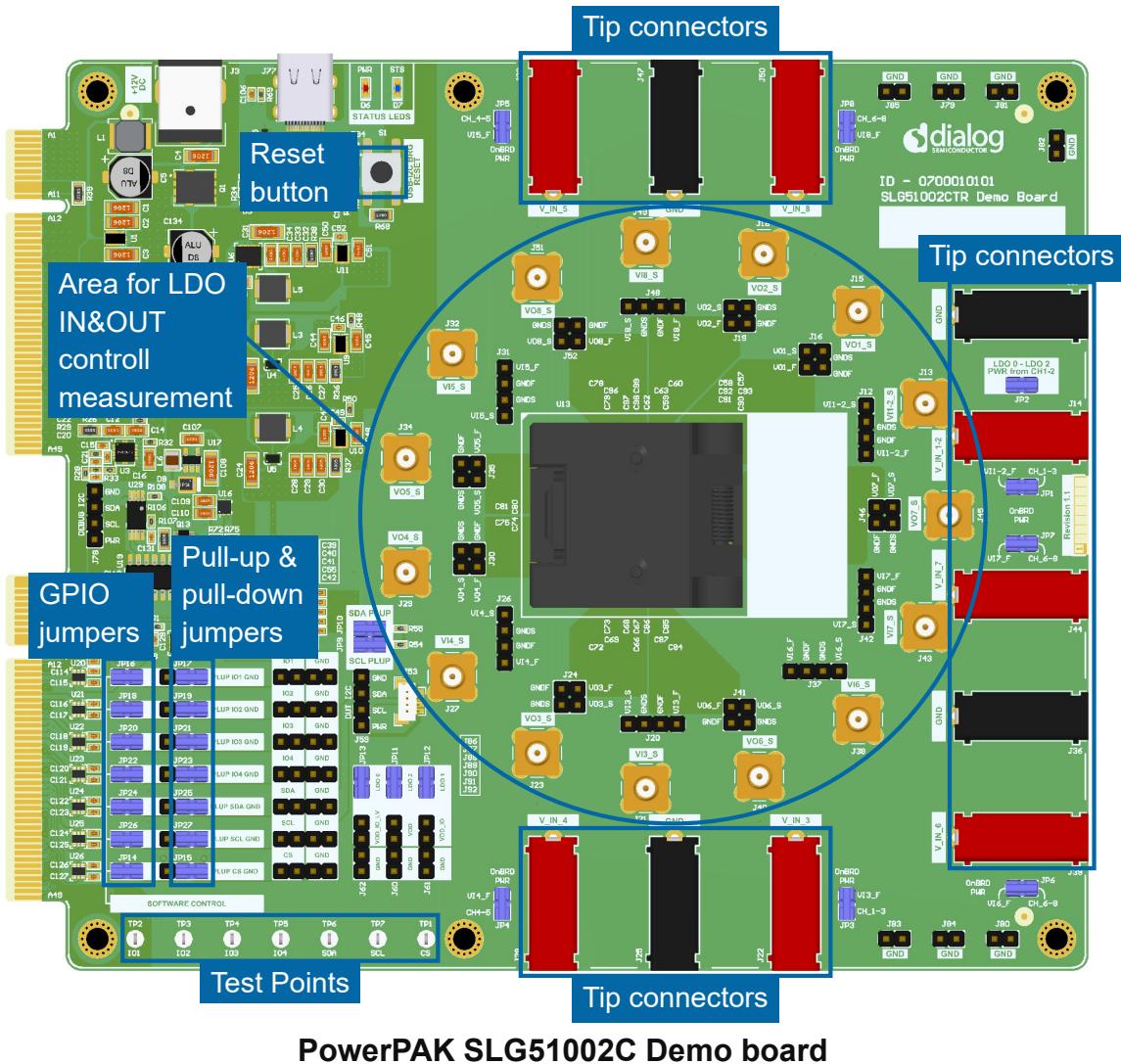


PowerPAK Development Platform controls

Find a comparison of the controls available on different PowerPAK boards in the appendix, section 5.3 Debugging Controls feature availability.

3.3.2 PowerPAK SLG51002C Demo board

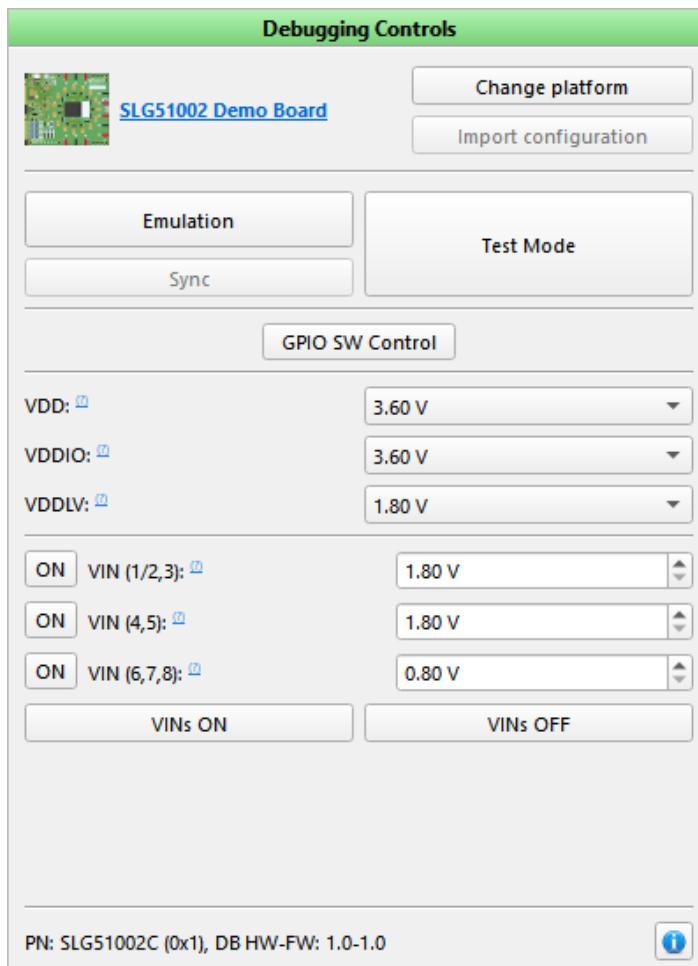
A *Demo board* is a special hardware intended to demonstrate some specific application of SLG51002C chip. It has the soldered SLG51002C chip or socket to insert the chip on the board.



The board has a USB communications interface that uses the USB type-C connector. Also, an external 12v power supply is required to power up the platform.

The PowerPAK SLG51002CTR Demo Board [Debugging Controls](#) User Interface includes the following sections:

Debug Configuration	Test Mode	PowePAK voltage controls
Emulation	GPIO SW Control	Info details
Sync	Voltage level controls	



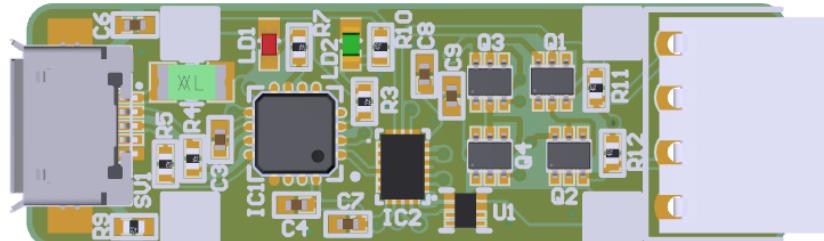
PowerPAK 002 Demo Board controls

Find a comparison of the controls available on different PowerPAK boards in the appendix, section [5.3 Debugging Controls feature availability](#).

3.3.3 Connecting external PowerPAK

Serial Debugger platform

You can use *GreenPAK Serial Debugger* for an external chip debugging of PowerPAK part numbers.

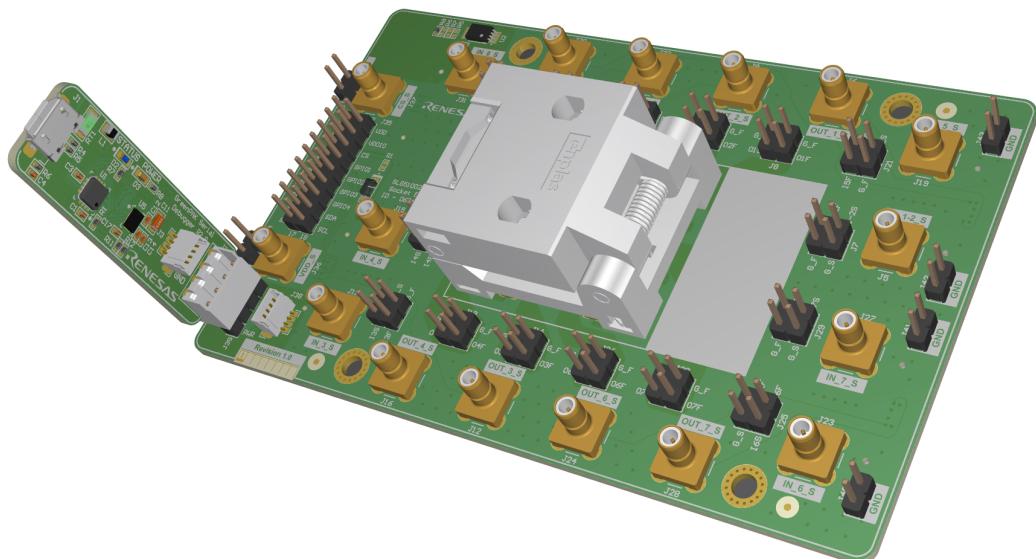


GreenPAK Serial debugger

To start working with an external chip on the *Serial Debugger* platform:

- Connect a chip with wires to the *Serial Debugger* pins (*VDD*, *CLK*, *SDA*, *GND*). If you use *SLG5100xCRT Socket Eval.*, use the DUT I2C connector.
 - Start the *Go Configure Software Hub* and select the Part Number of the connected external chip
 - Start the *Debug tool* and select the *Serial Debugger platform*
 - Select *I2C Device address* or a corresponding address programmed to the chip
 - Specify *VDD configuration* — *VDD* applied for I2C
 - Connect the chip power sources externally

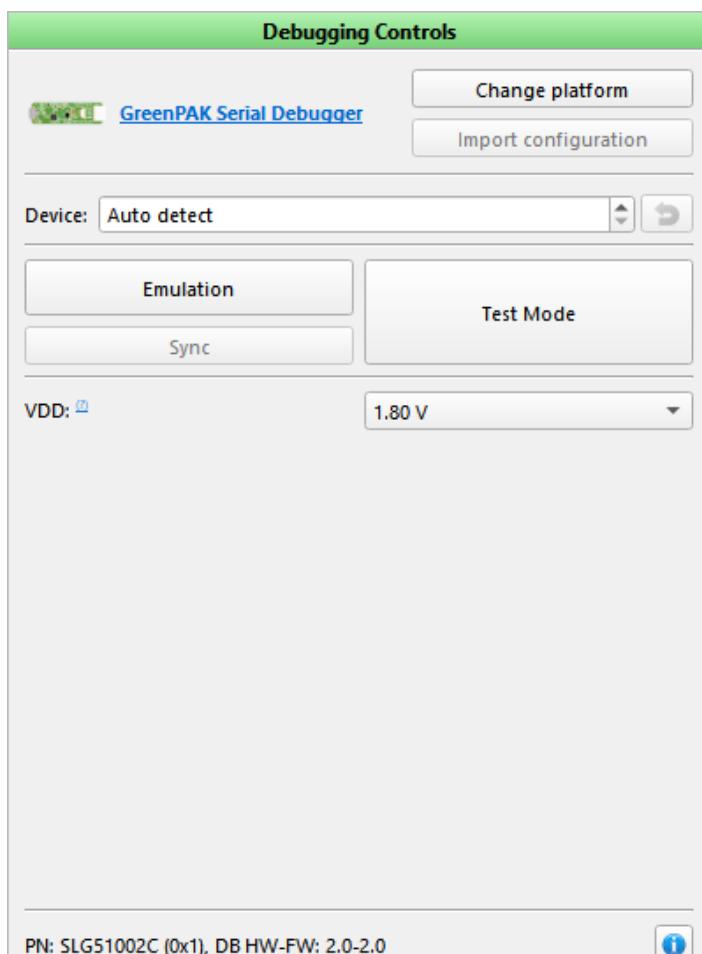
Once all steps are completed, the debugging controls become active.



GreenPAK Serial Debugger with PowerPAK socket

Debugging Controls interface for the *GreenPAK Serial Debugger Platform* (with SLG5100x device) includes:

Debug Configuration	Sync	Info details
Device selector	Test Mode	
Emulation	Voltage level controls	



PowerPAK Serial debugger controls

Find a comparison of the controls available on different *PowerPAK* boards in the appendix, section *Debugging Controls feature availability*.

4 How to

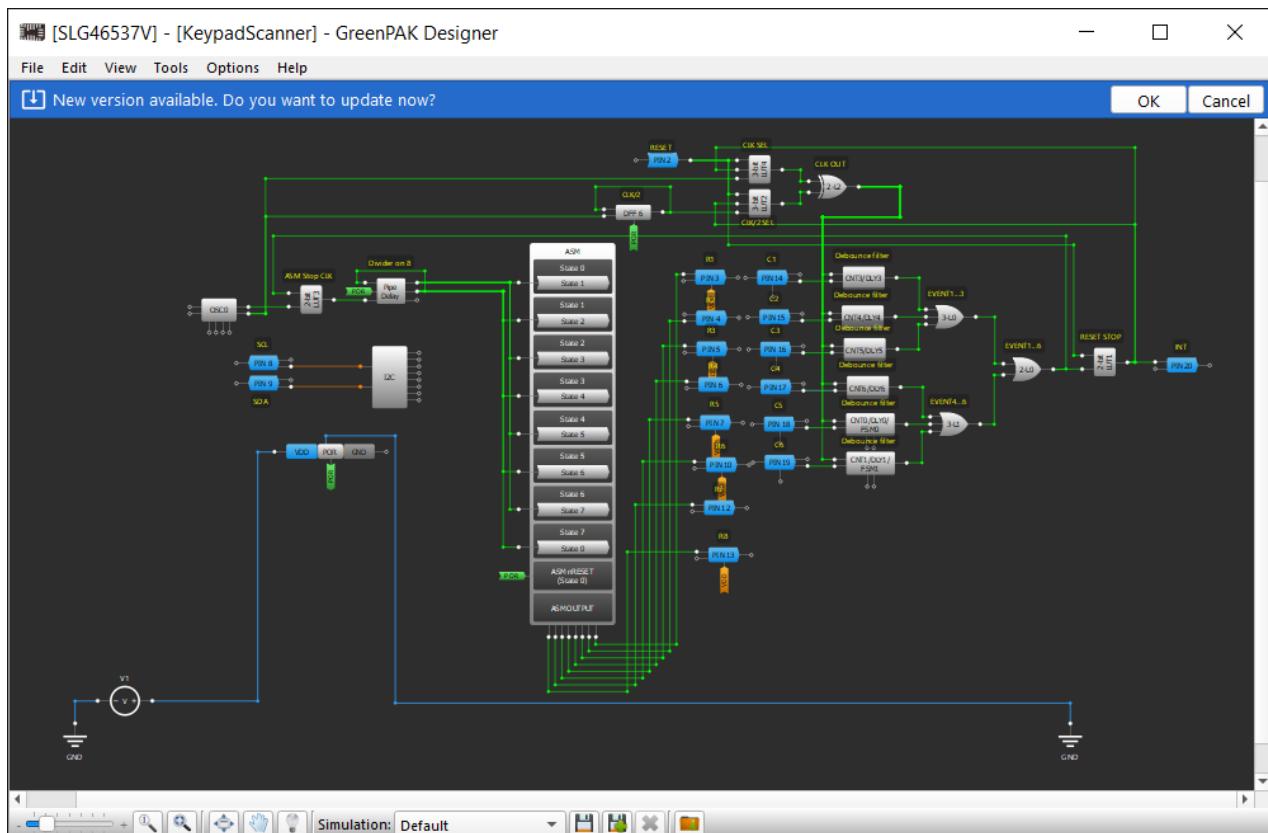
This chapter presents the methods to keep your software version up to date. Additionally, you will discover how to create and save block diagrams in a graphical format for integration into datasheets and other project-related documents.

4.1 Software update

Software updates improve user experience by addressing issues and offering new features. Updates can provide access to new templates, resources, or content libraries that can enhance the creative process.

There are two ways of updating the Go Configure Software Hub:

- When updates are available, a notification appears. You can either download the latest version or postpone the update until the program restarts. After the download is complete, the Go Configure Software Hub will handle the automatic installation of the software. **Note:** It is important to restart the software after the installation process is done.
- You can also find the latest version of the Go Configure Software Hub on the [Software page](#) on the [Renesas website](#). To ensure the best user experience, keep your software up to date.



Software update notification

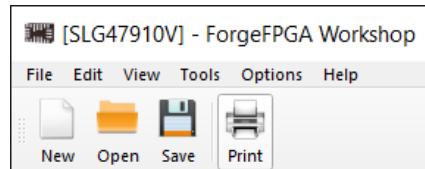
Adjust the *Updater* preferences in the *Designer Settings* window, *Updater* section (refer to Section 2.1.7 [Settings](#)).

Feel free to email suggested updates to the developer team to improve the software (click *Help* → *About Go Configure Software Hub*).

4.2 Printing

4.2.1 Print

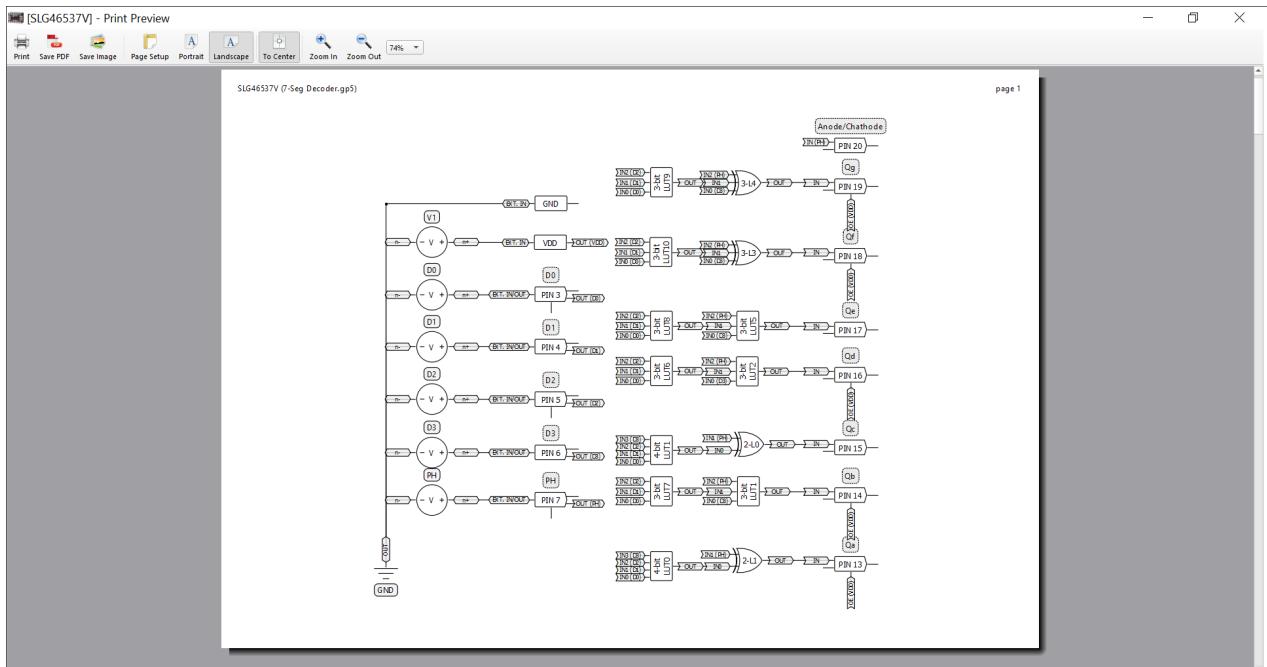
You can use the *Print tool* to present and save a block diagram in a graphic format to use these images in datasheets or other project documentation showing the interconnections between blocks and their configurations. Find the *Print tool* on the toolbar.



Print tool

Also, you can open the *Print tool* by clicking *File* → *Print* in the main menu.

To open the *Print Preview* window, click the *Print* button in the top toolbar. This window shows a ready-to-print diagram and tables with block configurations. At this point, you can't change the position of elements or lines on the diagram. To reposition components or wires, you should return to the working area, introduce the changes, and launch the *Print Tool* again.



Print Preview window

Additionally, the *Print Preview* window contains a table with a list of all blocks and their configurations.

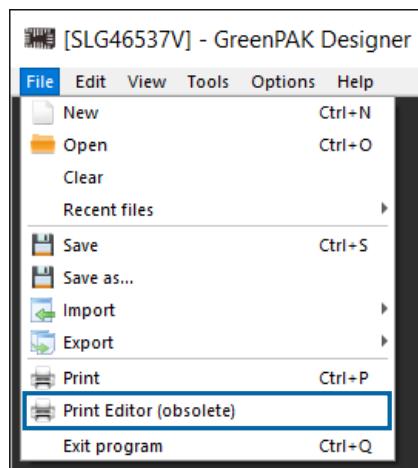
The screenshot shows the Print Preview window for the file "SLG46537V.gp5". It displays a series of tables, each representing a pin configuration:

- PIN 1 (VDD)**: Label: "D01", Property: Value, Min. value (V): 1.71, Typ. value (V): 3.30, Max. value (V): 5.50. I/O selection: Digital input, In mode: Digital in without Schmitt trigger, Out mode: None, Resistor: Floating.
- PIN 3 (IO1)**: Label: "D0", Property: Value, I/O selection: Digital input, In mode: Digital in without Schmitt trigger, Out mode: None, Resistor: Floating.
- PIN 4 (IO2)**: Label: "D1", Property: Value, I/O selection: Digital input, In mode: Digital in without Schmitt trigger, Out mode: None, Resistor: Floating.
- PIN 5 (IO3)**: Label: "D2", Property: Value, I/O selection: Digital input, In mode: Digital in without Schmitt trigger, Out mode: None, Resistor: Floating.
- PIN 6 (IO4)**: Label: "D3", Property: Value, I/O selection: Digital input, In mode: Digital in without Schmitt trigger, Out mode: None, Resistor: Floating.
- PIN 7 (IO5)**: Label: "PH", Property: Value, I/O selection: Digital input, In mode: Digital in without Schmitt trigger, Out mode: None, Resistor: Floating.
- PIN 13 (IO10)**: Label: "Qa", Property: Value, I/O selection: Digital output, In mode: None, Out mode: 1x push pull.
- PIN 14 (IO11)**: Label: "Qb", Property: Value, I/O selection: Digital output, In mode: None, Out mode: 1x push pull.
- PIN 15 (IO12)**: Label: "Qc", Property: Value, I/O selection: Digital output, In mode: None, Out mode: 1x push pull.
- PIN 16 (IO13)**: Label: "Qd", Property: Value, I/O selection: Digital output, In mode: None, Out mode: 1x push pull.
- PIN 17 (IO14)**: Label: "Qe", Property: Value, I/O selection: Digital output, In mode: None, Out mode: 1x push pull.
- PIN 18 (IO15)**: Label: "Qf", Property: Value, I/O selection: Digital output, In mode: None, Out mode: 1x push pull.
- PIN 19 (IO16)**: Label: "Qg", Property: Value, I/O selection: Digital output, In mode: None, Out mode: 1x push pull.

Table with block properties and values in Print Preview window

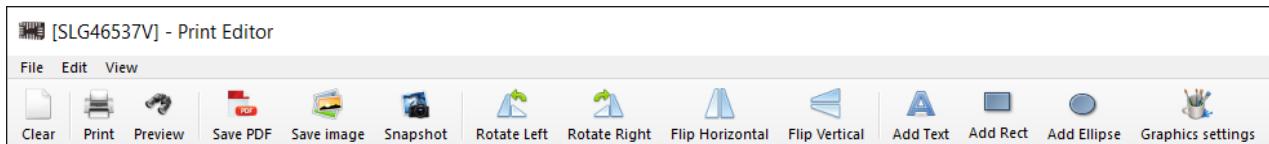
4.2.2 Print Editor (obsolete)

In the older versions of the software updates, the *Print editor (obsolete)* is available. You can find this tool by clicking *File* → *Print Editor (obsolete)*.



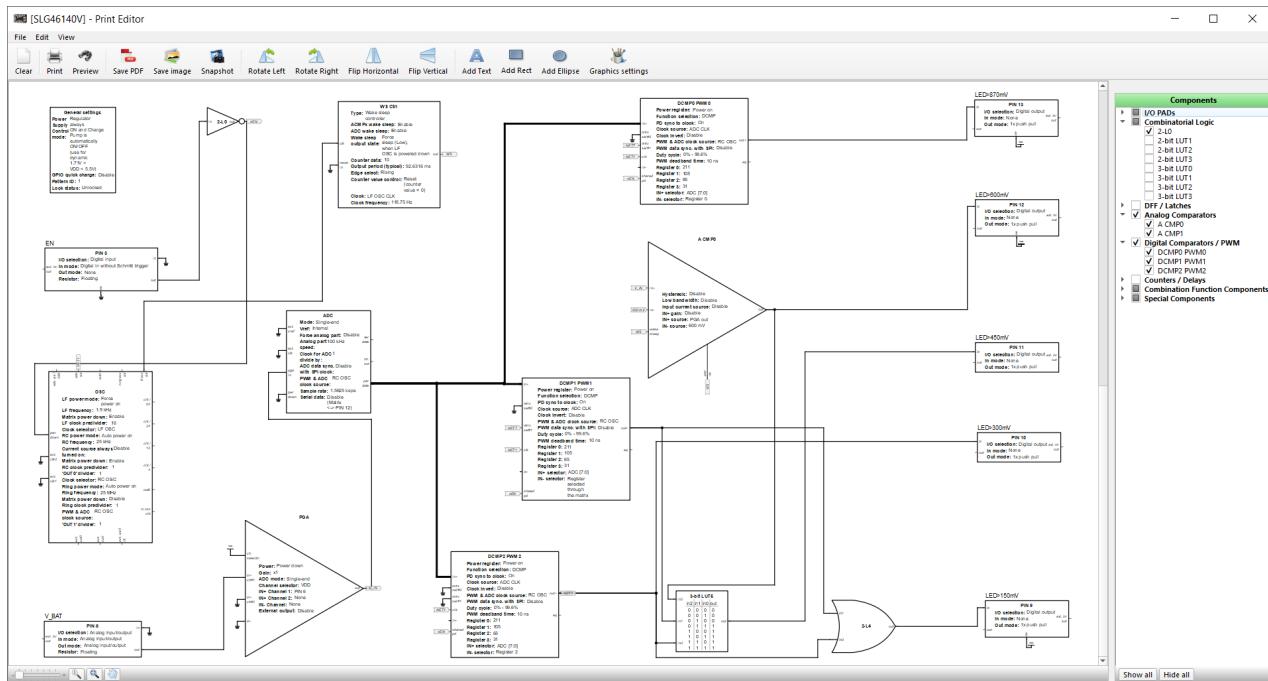
Print Editor (obsolete)

The *Print Editor* allows pre-print configuration such as rotating, flipping, and hiding some components or adding text labels.



Print Editor toolbar

An editable working area of the *Print Editor (obsolete)* contains all components used in the design. It enables customizing positions, views of components, and wires.



Print Editor (obsolete) interface

5 Appendices

In the appendix, you can find helpful information for the *Go Configure Software Hub* user experience enhancement.

5.1 Main menu commands

File

<i>New</i>	start a new or open existing project from Go Configure Software Hub
<i>Open</i>	open an existing project in software tools
<i>Save</i>	save current project
<i>Save as</i>	save the current project in a specified location
<i>Import NVM bits</i>	load configuration bits from a text file
<i>Export NVM bits</i>	save configuration bits to a text file
<i>Print</i>	a simple print feature without detailed block information
<i>Print Editor (Obsolete)</i>	start the Print Editor
<i>Exit program</i>	close software tools

Edit

<i>Rotate Left</i>	rotate a selected block counterclockwise
<i>Rotate Right</i>	rotate a selected block clockwise
<i>Flip Horizontal</i>	a horizontal reflection of a selected block
<i>Flip Vertical</i>	a vertical reflection of a selected block
<i>Align Horizontal</i>	horizontal alignment of selected blocks
<i>Align Vertical</i>	vertical alignment of selected blocks
<i>Set Label</i>	creating a text label for selected blocks
<i>Erase Label</i>	erasing text labels near selected blocks
<i>Set Wire</i>	enable wire creating mode
<i>Erase Wire</i>	enable wire erase mode

View

<i>Zoom in</i>	increase the work area scale
<i>Zoom out</i>	decrease the work area scale
<i>Fit work area</i>	tune scale to show all blocks visible in the project
<i>Zoom 1:1</i>	set default scale
<i>Full-screen mode</i>	switch to full-screen mode
<i>Pan mode</i>	enable/disable the work area move in pan mode
<i>Show hints</i>	enable/disable hints for blocks on the work area
<i>Properties</i>	show/hide Properties panel
<i>Schematic Library</i>	list of external components for Software Simulation
<i>Components</i>	show/hide software tools blocks list
<i>NVM Viewer</i>	show/hide NVM bits viewer
<i>Rules Checker Output</i>	scan the project for design errors

Tools

<i>Debug</i>	convenient project testing
<i>Rule Checker</i>	check current design for correct settings
<i>Comparison</i>	compare bits of two projects
<i>ASM Editor</i>	configure the State Machine using state diagram and set the output configuration for SM Output block
<i>I2C Tools</i>	enhanced I2C tools with I2C snapshot Reconfigurator

Options

<i>Settings</i>	set the default projects folder, autosave interval, toolbars position, recovery, shortcuts, and update options
-----------------	--

Help

<i>Help</i>	show help window
-------------	------------------

<i>User Guides</i>	open User guides on the web
<i>Legend box</i>	show the color legend box
<i>Renesas web site</i>	open the Renesas official website
<i>Software and documentation</i>	open Software and Documentation web page
<i>Renesas web store</i>	open Renesas chip store
<i>Design support</i>	web page with training courses and videos
<i>Contact Us</i>	web form with request
<i>Social</i>	Renesas in social networks
<i>Application Notes</i>	open examples web page
<i>Datasheet</i>	open documentation web page
<i>Updater</i>	open software update tool
<i>About Go Configure Software Hub</i>	show information about software tools version modification

5.2 Keyboard and mouse controls

Basic tools		
Action	Windows/Linux controls	macOS controls
NVM Viewer	<i>F2</i>	<i>F2</i>
Component properties	<i>F3</i>	<i>F3</i>
Apply changes	<i>Ctrl + A</i>	<i>Command + A</i>
Revert changes	<i>Ctrl + U</i>	<i>Command + U</i>
Reset settings to default	<i>Ctrl + Shift + R</i>	<i>Command + Shift + R</i>
Reset connections to default	<i>Ctrl + Shift + C</i>	<i>Command + Shift + C</i>
Components List	<i>F4</i>	<i>F4</i>
Filter on the Components List	<i>Ctrl + F</i>	<i>Command + F</i>
Rules Checker	<i>F5</i>	<i>F5</i>
Debug tool	<i>F9</i>	<i>F9</i>

Components and connections		
Action	Windows/Linux controls	macOS controls
Move selected block(s) by 1 pixel*	<i>Alt + left/right</i>	<i>Alt + left/right</i>
Move selected block(s) by 10 pixels*	<i>Ctrl + left/right</i>	<i>Command + left/right</i>
Rotate selected component(s) left	<i>Ctrl + L</i>	<i>Command + L</i>
Rotate selected component(s) right	<i>Ctrl + R</i>	<i>Command + R</i>
Flip selected component(s) horizontally	<i>Ctrl + H</i>	<i>Command + H</i>
Flip selected component(s) vertically	<i>Ctrl + V</i>	<i>Command + V</i>
Hide selected component(s)	<i>H</i>	<i>H</i>
Remove selected external component(s)	<i>Del</i>	<i>Backspace</i>
Set wire	<i>Ctrl + W</i>	<i>Command + W</i>
Erase wire	<i>Ctrl + E</i>	<i>Command + E</i>
Discard adding a wire*	<i>RMB</i>	<i>RMB</i>
Force Set wire while Erase wire is enabled*	<i>Hold Shift</i>	<i>Hold Shift</i>
Force Erase wire while Set wire is enabled*	<i>Hold Alt</i>	<i>Hold Alt</i>
Add multiple wires from the same source*	<i>Hold Ctrl</i>	<i>Hold Command</i>
Add multiple wires from the same source while Erase Wire is enabled*	<i>Hold Ctrl + Shift</i>	<i>Hold Command + Shift</i>
Force remove network while Set wire is enabled*	<i>Hold Ctrl + Alt</i>	<i>Hold Command + Alt</i>

Simulation results		
Action	Windows/Linux controls	macOS controls
Move the plot*	Hold MMB	Hold MMB
Add one marker on the plot*	<i>Ctrl + LMB/RMB</i>	<i>Command + LMB/RMB</i>
Add two markers on the plot*	<i>Ctrl + LMB + RMB</i>	<i>Command + LMB + RMB</i>
Clear all markers*	<i>Esc</i>	<i>Esc</i>
Zoom in/out x-axis*	<i>Ctrl + mouse wheel</i>	<i>Command + mouse wheel</i>
Zoom in/out y-axis*	<i>Shift + mouse wheel</i>	<i>Shift + mouse wheel</i>
Help*	<i>F1</i>	<i>F1</i>

Import model/subcircuit		
Action	Windows/Linux controls	macOS controls
Open search field*	<i>Ctrl + F</i>	<i>Command + F</i>
Close search field*	<i>Esc</i>	<i>Esc</i>

Debug		
Action	Windows/Linux controls	macOS controls
Emulation	<i>Shift + E</i>	<i>Shift + E</i>
Program	<i>Shift + P</i>	<i>Shift + P</i>
Read	<i>Shift + R</i>	<i>Shift + R</i>
Test Mode	<i>Shift + T</i>	<i>Shift + T</i>
NVM Data	<i>Shift + N</i>	<i>Shift + N</i>
Info	<i>Shift + I</i>	<i>Shift + I</i>
Log	<i>Shift + L</i>	<i>Shift + L</i>

I2C reconfigurator		
Action	Windows/Linux controls	macOS controls
Create snapshot*	<i>Shift + A</i>	<i>Shift + A</i>
Add delay*	<i>Shift + D</i>	<i>Shift + D</i>
Move up*	<i>Shift + up</i>	<i>Shift + up</i>
Move down*	<i>Shift + down</i>	<i>Shift + E</i>
Send one snapshot*	<i>F7</i>	<i>F7</i>
Send all snapshots*	<i>F6</i>	<i>F6</i>
Stop sending snapshots*	<i>Shift + F7</i>	<i>Shift + F7</i>

ASM editor		
Action	Windows/Linux controls	macOS controls
Set link	<i>Ctrl + W</i>	<i>Command + W</i>
Erase link	<i>Ctrl + E</i>	<i>Command + E</i>
Clear	<i>Ctrl + N</i>	<i>Command + N</i>
Undo	<i>Ctrl + Z</i>	<i>Command + Z</i>
Redo	<i>Ctrl + Y</i>	<i>Command + Y</i>
Help	<i>F1</i>	<i>F1</i>

General		
Action	Windows/Linux controls	macOS controls
New project	<i>Ctrl + N</i>	<i>Command + N</i>
Open project	<i>Ctrl + O</i>	<i>Command + O</i>
Save project	<i>Ctrl + S</i>	<i>Command + S</i>
Undo	<i>Ctrl + Z</i>	<i>Command + Z</i>
Redo	<i>Ctrl + Y</i>	<i>Command + Y</i>
Zoom in work area	+	+
Zoom out work area	-	-
Fullscreen mode	<i>F11</i>	<i>F11</i>
Print	<i>Ctrl + P</i>	<i>Command + P</i>
Help	<i>F1</i>	<i>F1</i>
Exit program	<i>Ctrl + Q</i>	<i>Command + Q</i>

Software tool launcher		
Action	Windows/Linux controls	macOS controls
Welcome tab*	<i>Ctrl + 1</i>	<i>Command + 1</i>
Recent files tab*	<i>Ctrl + 2</i>	<i>Command + 2</i>
Develop tab*	<i>Ctrl + 3</i>	<i>Command + 3</i>
Demo tab*	<i>Ctrl + 4</i>	<i>Command + 4</i>
Recovery files tab*	<i>Ctrl + 5</i>	<i>Command + 5</i>
Datasheets*	<i>Ctrl + 6</i>	<i>Command + 6</i>
User guide*	<i>Ctrl + 7</i>	<i>Command + 7</i>

Note: Non-configurable controls are marked with an asterisk (*).

5.3 Debugging Controls feature availability

GreenPAK platforms

Feature	Advanced board	DIP board	Lite board	Serial Debugger
Debug Configuration	✓	✓	✓	✓
Device selector	✓	✓	✓	✓
External device mode			✓	
Chip procedures	✓	✓	✓	✓
Project data window	✓	✓	✓	✓
Generator controls	✓	✓		
Expansion connector	✓	✓	✓	
TP map	✓	✓	✓	
Power source selector	✓	✓	✓	✓
Voltage level controls			✓	✓
LEDs ON and LEDs OFF	✓	✓	✓	
Info details	✓	✓	✓	✓

FPGA platforms

Feature	ForgeFPGA advanced board	ForgeFPGA Evaluation board
Debug Configuration	✓	✓
Chip procedures	✓	✓
Flash procedures	✓	
Generator controls	✓	
TP map	✓	
Voltage level controls		✓
Info details	✓	✓

PowerPAK platforms

Feature	PowerPAK Development Platform	SLG51002CTR Demo Board	Serial Debugger(with SLG5100x)
Debug Configuration	✓	✓	✓
Device selector			✓
Chip procedures	✓	✓	✓
GPIO SW Control		✓	
TP map	✓		
Voltage level controls	✓	✓	✓
PowerPAK voltage controls	✓	✓	
Info details	✓	✓	✓

5.4 Abbreviations

Abbreviation	Description
ASM	Asynchronous State Machine
CLK	Clock
CS	Circuit Switched
DC	Direct Current
DIP	Dual In-Line Package
EEPROM	Electrically Erasable Programmable Read-only Memory
EPG	Extended Pattern Generator
FPGA	Field-Programmable Gate Array
GND	Ground
GPI	General-Purpose Input
GPIO	General-Purpose Input/Output
I/O	Input/Output
I2C	Inter-Integrated Circuit
IC	Integrated Circuit
IDE	Integrated Development Environment
IEC	International Electrotechnical Commission
LDO	Low-Dropout Regulator
LED	Light Emitting Diode
LUT	Look-Up Table
NC	Not Connected
NVM	Non-Volatile Memory
OE	Output Enable
OTP	One-Time Programmable
POR	Power-On Reset
PWM	Pulse Width Modulation
SCL	Serial Clock
SDA	Serial Data
SPI	Serial Peripheral Interface
SPICE	Simulation Program with Integrated Circuit Emphasis
TP	Test Point
UART	Universal Asynchronous Receiver/Transmitter

5.5 Changelog

Version	Date	Changes
2.0.1	2023-10-25	Fix typos Add Changelog
2.0.0	2023-10-13	New revised structure Significantly improved navigation Carefully selected materials Improved instructions Improved graphic material

IMPORTANT NOTICE AND DISCLAIMER

RENESAS ELECTRONICS CORPORATION AND ITS SUBSIDIARIES ("RENESAS") PROVIDES TECHNICAL SPECIFICATIONS AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES "AS IS" AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for developers skilled in the art designing with Renesas products. You are solely responsible for (1) selecting the appropriate products for your application, (2) designing, validating, and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. Renesas grants you permission to use these resources only for development of an application that uses Renesas products. Other reproduction or use of these resources is strictly prohibited. No license is granted to any other Renesas intellectual property or to any third party intellectual property. Renesas disclaims responsibility for, and you will fully indemnify Renesas and its representatives against, any claims, damages, costs, losses, or liabilities arising out of your use of these resources. Renesas' products are provided only subject to Renesas' Terms and Conditions of Sale or other applicable terms agreed to in writing. No use of any Renesas resources expands or otherwise alters any applicable warranties or warranty disclaimers for these products.

(Disclaimer Rev.1.0 Mar 2020)

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

Contact Information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit:
www.renesas.com/contact/

Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.