1 Introduction 4
  1.1 Quick Start .......................................................... 5
    1.1.1 System Requirements ........................................... 5
    1.1.2 Support ........................................................ 5

2 Software overview 6
  2.1 Design tools ....................................................... 7
    2.1.1 Interface overview ............................................. 8
    2.1.2 Working with project files ................................... 13
    2.1.3 NVM viewer .................................................... 18
    2.1.4 Components ................................................... 19
    2.1.5 Connections ................................................... 22
    2.1.6 Keyboard commands ............................................ 24
    2.1.7 Settings ....................................................... 26
    2.1.8 Legend box .................................................... 30
    2.1.9 Help window ................................................... 30
    2.1.10 Snipping tool ................................................ 32
    2.1.11 Rule checker ................................................ 33
    2.1.12 Comparison tool ............................................. 34
    2.1.13 Power Dissipation Calculator ............................... 37
  2.2 Debug tools ........................................................ 42
    2.2.1 Hardware sources .............................................. 43
    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 ....................................... 84
    2.2.11 I2C I/O Tool ................................................ 86
    2.2.12 Logic Analyzer .............................................. 88
5.1 Failed Socket Test .......................................................... 175

6 Appendices ...................................................................... 177
  6.1 Main menu commands .................................................. 178
  6.2 Keyboard and mouse controls ....................................... 180
  6.3 Debugging Controls feature availability ......................... 183
  6.4 Abbreviations ............................................................... 184
  6.5 Changelog ................................................................. 185
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: 8 GB
- GPU: 128 MB
- Free disk space: 2 GB
- OS: Windows 10/11, macOS 11 or higher, Ubuntu 20.04/22.04, Debian 11/Testing

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.
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

![Go Configure Software Hub User Interface](image)
2.1.1 Interface overview

The basic interface elements are main menu, toolbar, work area, work area controls, properties panel, and component 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 6.1 Main menu commands.

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.
**Work area**

The work area displays the selected components (macrocells) from the component 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](image-url)
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.
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

Component 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.

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

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).

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.

**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*.

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
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](image)

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

![Customer project name in Print Preview](image)

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](image)
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.

![NVM viewer top bar](image)

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

![NVM hint](image)

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 6.1 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](image)

### 2.1.4 Components

After you add a macrocell from the component 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 6.2 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.

**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.

- **OUT**: open for connection
- **IN**: connection limit is reached
- **PWR_UP**: temporarily closed for connection (you can change macrocells properties to make it available for connection)
- **OUT**: used for hard-wired connections
- **EXT. IN/OUT**: 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.

- **POR**: open for connection
- **TEMP SENSOR**: temporarily closed for connection (you can change macrocell settings on its Properties panel to make it available for connection)
- **TEMP SENSOR**: used for hard-wired connection
- **POR**: 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).

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

**Hard-wired** connections which are predefined and depend on block settings

**Bus** same as hard-wired

**State dependent** connections set to state dependent components (e.g. Dynamic Memory, F or State blocks within ASM subsystem in SLG46880V chip)

**External** 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:

- **Matrix** the most common connection which can be set between green pins

- **Hard-wired** 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.

---

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.

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

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.

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 in 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.

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.

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 6.2 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

**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
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 colors](image)

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.

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

You may also see the information buttons (ℹ️) 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*).

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

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

The Rule 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

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

**Rule Checker window**

**Rule checker controls**

**Rule Checker** window shows three message types:

- **Fail** — 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).
- **Warning** — 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.
- **Note** — 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*.

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** — *Go Configure Software Hub* project file with the following extensions: .aap, .can, .gp3, .gp4, .gp5, .gp6, .hvp, and .ppak
- **Text File** — exported NVM file in .txt format
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.

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.

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 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

The tool helps to determine the amount of heat dissipation of the device, which is an important step in developing an effective thermal management solution. *Power Dissipation Calculator* is available only for the HVPAK Part Numbers and mainly applies to the HV OUT CTRL macrocell.
To launch the tool, click the *Power Dissipation* button on the toolbar or find it in the main menu, *Tools*.

![Power Dissipation Calculator preview window](image)

The tool consists of four main parts:

- **Power supply** — voltage applied to VDD pin and estimated current consumption parameters

- **HV OUT 0 and HV OUT 1 settings** — the main parameters of the power pins used in the design
  - *Design configuration* — reflects the *HV OUT mode* parameter selection, set on the HV OUT CTRL macrocell’s *Properties* panel
  - *Fall Time/Rise Time/HS FET on resistance (mOhm)/LS FET on resistance (mOhm)* — configurations, which are automatically adjusted based on the conditions and configuration of the HV OUT CTRL component
  - *High Voltage Power Supply* — set the supply voltage (VDD2) for HV OUT CTRL
HV OUT RMS — responsible for the output current and cannot exceed 1.5 A per HV OUT

PWM 0 Frequency (kHz) — set 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.

HV OUT properties panel

Note: When the output pins connected to the HV OUT CTRL macrocell are in the Hi-Z state, HV OUT 0 and HV OUT 1 settings are displayed as locked. Change the HV Pin’s Output mode on its Properties panel to unlock the settings.

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

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

- **Thermal Resistance** — you need only the ambient temperature of the device for calculations. This method is the least accurate. If the platform you are using contains thermal vias, you can tick the respective checkbox to take it into account.

- **Junction to Board** — the PCB temperature is used to calculate the Die temperature. It should be measured 1 mm from the device. This method has a higher accuracy compared to Thermal Resistance.
– *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.
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
- Pull down
- Signal Generator
- Logic Generator
- I2C Generator
- Constant Voltage
- Parametric Generator
- Synchronous Logic Generator

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]

![PowerPAK 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.

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.
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.
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 Settings](image)

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.

![I2C Generator](image)

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](image)
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.

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

- **Constant voltage waveform type**

![Signal Generator Diagram](image)

![Signal Generator Settings](image)

Configuration options of Signal Generator, type Constant Voltage
Configuration options of Signal Generator, type Sine
Trapeze waveform type

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 |
| Umin: | 0.000 |
| Pattern: | 0110101 |

<table>
<thead>
<tr>
<th>T</th>
<th>U</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>50.000 ms</td>
</tr>
<tr>
<td>2</td>
<td>50.000 ms</td>
</tr>
<tr>
<td>3</td>
<td>50.000 ms</td>
</tr>
<tr>
<td>4</td>
<td>50.000 ms</td>
</tr>
<tr>
<td>5</td>
<td>50.000 ms</td>
</tr>
<tr>
<td>6</td>
<td>50.000 ms</td>
</tr>
<tr>
<td>7</td>
<td>50.000 ms</td>
</tr>
<tr>
<td>8</td>
<td>50.000 ms</td>
</tr>
</tbody>
</table>

Insert before row: 1
Remove row: 1

Levels count: 0

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.
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.

---

**Signal Wizard for Synchronous Logic Generator**

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

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

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
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](image)

- **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](image)

- **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](image)

> **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](image)

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.

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.

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.

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](image)

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](image)

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

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 6.3 *Debugging Controls* feature availability.

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

- GreenPAK Advanced Dev. Platform
- GreenPAK DIP Dev. Platform
- GreenPAK Lite Dev. Platform
- GreenPAK Serial Debugger
- ForgeFPGA Advanced Dev. Platform
- ForgeFPGA Evaluation Board
- PowerPAK Dev. Platform
- PowerPAK Demo Board
- GreenPAK Serial Debugger (with SLG5100x)

**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](image)

- **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/Chip in socket** — search the device placed on the board
- **External** — search the device on the selected I2C slave address
- **Auto detect/External, connected to I2C OUT** — search the device on all available I2C slave addresses, starting from 0 until the first one is found

![Device selector](image)

**Note:** Make sure you do not connect the socket to the platform and external chip at the same time.

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](image)

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)

**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)
Generator controls

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

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
TP map

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

![TP map](image)

**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](image)

**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](image)

**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](image)

GPIO SW control

- **GPIO SW Control** — add buttons for software control of GPIOs, GPIO(SDA), GPIO(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](image)

Info details

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

![Info details](image)

- 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.

Blink  Blink with an LED of the board in use

Update the chip and board Info details

Run the Socket Test

The Socket Test checks if the chip is successfully connected to the board. Click Show Details to get the Socket Test report.

If the Socket Test fails for any reason, you can refer to the Troubleshooting section for assistance.
2.2.7 Demo Board and Demo mode

The Demo mode allows exploring possible applications of a specific Part Number. The Demo tab on the Hub window contains the list of the preconfigured projects. Click Details to find out more about the design. To open the project click Run.

Select a Demo project

To interact with the design, you can connect the specific Demo Board. Such board contains a soldered IC with a preprogrammed project.

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

After the board detection is successful, the following controls become available:

- **Write** — load the current project’s 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 refer to section 2.2.8 I2C Tools.
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.

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.

The tool is comprised of I2C Virtual Inputs, I2C Virtual Outputs, and I2C Reconfigurator that can be launched from the I2C 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 tool 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** tool allows you to **read** and **write** the value of all available virtual inputs via **I2C**.
The *Registers* page shows entire chip NVM. The columns at the left, R and W show if a register is readable and writable, respectively.

- **Green** square: I2C operations are supported
- **Red** square: I2C operations are not supported
- **Yellow** square: 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](image)

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
Searching for a register, filtering out bytes

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

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.

![Image](image_url)

**Context menus**

**Data Buffers**

The Data Buffers tool allows you to read data from the buffers and the ADC data register.

For automatic data actualization, click Read while the Auto read every 1s option is enabled.

![Image](image_url)

**Managing Data Buffers**
Memory Table

The Memory Table tool allows you to read data from the Memory Table macrocell in the Column X section. To change the amount of displayed columns use the Columns: selector. In addition to the decimal and hexadecimal formats, applicable for both column types, the Address columns data view can be also changed using the following options:

- **Absolute** — show the actual address of the register in the macrocell
- **Relative** — show the register address starting with zero

You can save the read data in .csv format by clicking the Export to file button.

Managing Memory Table

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 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.

![Snapshot configuration](image)

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:

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

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 a 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 a 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

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 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.

**Project data window table:**

- **R and W** — show if a register is readable and writable
  - Green: I2C operations allowed
  - Red: 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.
- **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

![Image of Project data window](image)
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](image)

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](image)
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
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.

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).

Internal trigger list

Trigger configuration

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

![Protocol settings](image)

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

![Decoded data](image)

**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](image)

**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.

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

**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.
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*.

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

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
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.

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 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.

2.2.14 Voltage Monitor

The Voltage Monitor tool helps you to measure the voltage on the ADC channels. The tool appears on the toolbar after you click Debug and select the appropriate development platform.
Once the platform is connected, the controls on the tool become active, and you can update the table data.

The following controls are available:

- **Refresh** — update the ADC channels measurement data in the table
- **Auto-refresh** — auto-update the ADC channels measurement data
- **Last Update** — time of the last measurement

The tool also contains the **Channel/Value** table.

- **Channel** — channel types with VIN and VOUT relations
- **Value** — value measured in volts. The accuracy of measurement is three decimal places
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’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.

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.

- Regular state
- Selected state
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.

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.

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*. 

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.

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.
Use the toolbar controls to work with the tool.

![Timing diagram options](image1)

### 2.3.4 Memory Table Editor

*Memory Table Editor* allows you to configure the Memory table macrocell (make sure you select the ROM mode on the macrocell’s *Properties* panel). To open the tool, click the *Memory Table Editor* button on the toolbar or double-click the corresponding component on the work area.

![Memory Table Editor toolbar button](image2)

The *Memory Table Editor* window visualizes the output data of the Width Converter (WC) in graphical format (WC provides data from Memory Table to the connection matrix).

![Memory Table Editor window](image3)
The tool provides you with the following possibilities:

- Add and configure a generator using the offered settings, e.g. specify which Width Converter output corresponds to which generator signal. Use Manual Editor to customize the data (read more below).

![Generator configuration](image)

**Note**: If a generator is locked and cannot be added, change the Width converter mode.

- Check the Resources bar, indicating the number of available cells. The number of cells, as well as the displayed output signals, depends on the selected Width converter mode, which are:
  - \(12 \rightarrow 12\) (12-bit parallel output)
  - \(12 \rightarrow 4\) (12-bit word to three 4-bit words)
  - \(12 \rightarrow 2\) (12-bit word to six 2-bit words)
  - \(12 \rightarrow 1\) (12-bit word to serial bit stream)

![Resources/Width converter mode](image)

**Manual Editor** allows you to modify the macrocell’s bit values of the specific registers, and consists of three main parts:

- Navigation controls — enter the bit number and click Go to bit button to jump to the desired bit.
– Registers value configuration — configure the registers by setting values in binary, decimal, or hexadecimal format (bits set to 1 are indicated with white highlight on the sidebar)

– Bottom controls — Import/Export or Clear the data

Memory Table Data Editor

▶ The additional settings provide flexibility in table usage defining the access to the memory cells. To activate the Two range mode checkbox, the following conditions should be met:

– Memory Control Counter (MCC): Range mode select = Two ranges; Initial CNT data value != 0 (make configurations on MCC’s Properties panel)

– Generator is added

▶ Save/Load the file with configurations using the corresponding bottom controls
See some additional plot controls:

- Zoom the graphs with Ctrl + mouse wheel. Reset scale via the context menu
- Enable legend for each plot via the context menu
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](image)

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](image)

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:

![Import subcircuit dialog](image)

**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.

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.

![Circuit Diagram]

**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.

![Circuit Diagram]

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,
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.

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.

**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.

Add a probe by selecting a component and a property in the drop-down lists on the respective panel.
Right-click the parametric probe sticker to facilitate adding/removing a probe and editing the probe parameters.

![Parametric probe sticker](image)

### 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](image)

*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.
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

**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).
2.4.8 Simulation progress

The *Simulation in progress* window appears while processing data. Simulation time remaining is estimated dynamically.

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.

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.
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.

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.
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, .vcd, 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 $\Delta t$ and $\Delta V$ parameters on the Plot widget:

![Plot widget and Measurements](image)

**Group List**

The *Group 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](image)
A sine simulated with huge interval settings

The Group List bottom controls let you operate the waveforms.

Group List

Group controls include:

- ♦  v / ^ — 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 parametric 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.
2.5 FPGA Editor

The *FPGA Editor* is a complex software tool designed to create and configure the FPGA logic. The editor allows you to create designs for Field-Programmable Gate Arrays (FPGAs) using Verilog, a Hardware Description Language (HDL). Before you program your design onto the FPGA, the editor provides an option to simulate it first and ensure it’s correct.

The *FPGA Editor* includes a graphical representation of the Register Transfer Level (RTL) for the FPGA logic you designed. You can also use an additional macrocell constructor instead of writing a Verilog code, manually place the required blocks to the work area, connect them, and convert the complete design into the code representation.

In addition, *FPGA Editor* allows you to work with external designs by importing files with the created logic mapped to the components. Find the description of these and some additional features in the sections below.

2.5.1 Flowchart

We have created a flowchart to narrow down and follow the basic steps from start to finish. The flowchart shows all the important aspects of using the ForgeFPGA software.

```
Software flow
```

![Flowchart diagram](image)
2.5.2 FPGA Editor interface

To open FPGA Editor, click the respective button on the toolbar or double-click the FPGA Core component on the work area.

FPGA Editor button on the toolbar

Toolbar

The top toolbar provides quick access to a set of tools and actions. See the description of all available tools in the next sections.

Control panel

From the left control panel, you can access:

- **Resources Report** — an extract from the resource usage report. It shows the part of available resources utilized by the design sources
Sources — a list of all the Verilog sources and external netlists you have in the current project file. Here you can find the following subcategories:

- Custom Code
- IP Blocks
- Testbenches
- External Netlists

You can customize the Sources list via context menu with the options to add, delete, or rename the sources.

Synthesize and Generate Bitstream — main controls to run toolchain on your design to produce chip configuration

Logger/Issues panel
Here you can see the generated messages that appear after you use Synthesize, Generate Bitstream, or Simulation procedure.

- Logger — shows the information messages along with warnings and errors that were recorded while processing the design

- Issues — displays the warning and error messages that are automatically generated by the software when required. Once you receive a syntax error, you can right-click it and select Show in Editor to highlight a line in your Verilog code with a detected mistake
Additional controls

See the list of FPGA Editor controls that can enhance your user experience (the controls availability may depend on the selected tool):

- **Split** — divide your work area side-by-side to work simultaneously with more than one tool

- **Find** — activate the search field in the main menu, *Edit → Find (Ctrl + F)*

- **Footer controls:**
  - **Zoom** — zoom in/out the work area
  - **Pan mode** — click, hold and drag the cursor to move the work area (use the middle mouse button as an alternative)
  - **Zoom to selection** — click, hold and drag a cross cursor over the element you want to zoom
  - **Align Vertical/Horizontal** — align the macrocells relative to each other on the work area
2.5.3 Writing Verilog code


There are a few key code attributes that are important while working with the synthesis tool:

- (* top *) — the main module of your design should be marked with this attribute so the toolchain can successfully recognize which of the modules in the design is the top one

- (* clkbuf_inhibit *) — clocks signals in the input list of the main module should be marked with this attribute. This prevents clock buffer insertion by the synthesis tool, which may lead to the distortion of the clock signal name in the resulting netlist

- (* iopad_external_pin *) — all the external pins that are used in any source code need to be marked with this attribute

Example of working with Verilog
2.5.4 Modules Library

*Modules Library* is a comprehensive repository of pre-designed and pre-verified easy-to-integrate modules. This tool provides Verilog code for various hardware modules, accompanied by the testbenches to check their functionality using *Simulation*.

To open *Modules Library*, click a corresponding button on the toolbar or reach the main menu, *File → Modules Library*. Choose the module, set the required configurations, and add the name to complete the module creation.

Inside the *Modules Library* GUI, you can find the schematic, resource estimation along with the description of the block selected. The GUI gives a detailed explanation of all the input and output pins of the block and allows you to change the parameters as desired. This gives you the flexibility to create the Verilog code of the module needed and its associated testbench with just a few clicks.

![Modules Library GUI](image)
You can check the added block on the Sources control panel under the Ip Blocks group. Along with the block, the testbench module is added, and you can find it under the Testbenches group. You can also add multiple modules to your design and work with them simultaneously.

2.5.5 Synthesis

FPGA Editor has a built-in synthesis tool that takes an input design and produces a netlist file. While performing synthesis, the input design is analyzed and converted into gate-level representation. To run synthesis on your design, click the Synthesize button in the bottom left corner of FPGA Editor or from the main menu Tools → Run Synthesis.

During the synthesis the software also checks the Verilog code for syntax errors. You can check the log output in the Logger section. The synthesis is successful only when the code is syntax error-free.

Netlist

The system generates a netlist file after the successful synthesis. It describes the components and connectivity of the source design and is required to perform the subsequent place-and-route procedure. You can access the netlist by clicking the toolbar Netlist button or from the main menu,
You can also import an existing netlist by clicking \textit{File} → \textit{Import} → \textit{Netlist}.

\textbf{Post-Synth RTL}

At the Register-Transfer Level, the design is represented by combinational data paths and registers. RTL synthesis is easy as each circuit node element in the netlist is replaced with an equivalent gate-level circuit. In Post-Synthesis RTL, the synthesized inputs are taken as a netlist. It helps in providing information about the clock and other clock-related logic in the design, which enables additional I/O planning.

You can find the Post-Synth RTL report by clicking the respective toolbar button or from the main menu, \textit{Window} → \textit{Post-Synth RTL}. The report contains all the connections made within the module between the LUTs, FDREs, and Carry-Chain logics.
2.5.6 Generating Bitstream

To prepare your design to be sent to the device, you need to perform the place-and-route and the bitstream generation procedures. You can do this once the netlist and bitstream files are successfully generated. Place-and-route takes the elements of the synthesized netlist and maps its primitives to FPGA physical resources. To receive bitstream data, click the Generate Bitstream button in the bottom left corner of FPGA Editor or from the main menu, Tools → Generate Bitstream.

You can check Logger to inspect the background steps after you click Generate Bitstream. In the background, the software automatically performs technology mapping, clustering and floor planning, placement and optimization, routing and resource calculation. If the issue occurs in any of these steps, the bitstream generation will be incomplete, and you will see outcome on the Logger and Issues panels.

The generated bitstream is a hex file which, if generated correctly, you can further use for debugging your design. Read more about the chip/flash procedures and where to find them in section 2.2.6 Debugging Controls.

After bitstream generation is completed, you can see the generated info for the tools described later in this section.

I/O Planner

Each I/O port available on FPGA has a dedicated function that can be mapped to your design using the I/O Planner tool. Click the corresponding button on the toolbar, or go to the main menu, Window → I/O Planner to launch the tool.

The I/O Planner tool is represented as a table with the following columns:

- **Position** — the coordinate of a certain device I/O port on FPGA
- **Function** — dedicated function, assigned to the port
- **Port** — editable column, where you can input ports from your Verilog design to connect them to the desired functionality. Double-click a cell to see the list of all the ports defined in the Verilog code

![I/O Planner](image)
For easier navigation, use the additional controls to filter out the desired data.

![Filter controls]

You can also clear, import, or export the port data via main menu, Tools → I/O Planner, or using the table's context menu.

**Floorplan**

The results of the place-and-route procedure are visualized in the *Floorplan* tool. Check how the primitives from the netlist are placed and interconnected, as well as how the I/O ports are mapped to the internal blocks and GPIOs. Launch the tool via the toolbar or Window section in the main menu. Use the bottom toolbar controls to take a closer look and navigate inside the tool.

![Floorplan]

Click the internal component to see its details on the block configuration info panel. Also, see the
components and connections color scheme by clicking the Legend box icon at the right bottom.

Informational elements

In addition, you can manually upload the configurations by accessing Load custom PNR in the main menu, Tools → Floorplan.

Resources Report

After you perform the bitstream generation procedure, a full report of the used resources is generated. This report shows the amount of CLBs, FFs, I/Os, and LUTs used for the design synthesis. You can also see the list of utilized resources on the control panel. To open the Resources Report window, click the corresponding button on the toolbar or reach the main menu, Window → Resources Report.
Timing Analysis

The tool is designed to extract timing and check for any timing violations associated with any of the internal registers. The results show whether all set-up, hold, and pulse-width time is being met.

You can find the tool on the toolbar, or by clicking Window → Timing Analysis in the main menu.

2.5.7 Build folder

As soon as we perform the procedures (synthesis, or both synthesis and bitstream generation), the software creates a folder containing the generated files. This is what we call a build folder, and its name format is build_projectFileName (where the second part is the name of your project file). The build folder appears in the project file destination folder.

The list of generated files depends on the performed procedures. Below, you can see the description of the most important files present in the build folder:

- **FPGA_bitstream_AXI.log** — contains the generated bitstream that represents your project’s logic. This file is used while interacting with the development hardware upon performing the chip/flash procedures
- **io_spec_in.txt** — lists the I/O ports specification added in the I/O Planner, which is created right upon clicking Generate Bitstream
- **PNR_IO.log** — I/O ports mapping file generated after successful bitstream generation procedure
- **netlist.edif** — the result of Yosys data processing, describing the components and connectivity within the source design. You can also import an external netlist from another software/synthesis tool
- **PNR_PACK_PLACE.log** — source data for building a floorplan
- **post_synth_results.v** — Yosys output data in the Verilog code format
- **resource-utilization-report.log** — information about the resources amount required for your design
2.5.8 Simulation

Simulation allows you to verify the overall functionality of the FPGA design and its response to different inputs without the need for physical hardware.

FPGA Editor works in correspondence with 3rd party software for simulating the testbench, called Icarus Verilog, and for verifying the functionality of the design by viewing simulation results we use GTKWave. Please refer to the ForgeFPGA GTKWave User Guide for the installation process of the additional software.

Writing a testbench

To work with Simulation you should have a testbench module for the FPGA design. You can add a testbench from the Modules Library, or open a module from the main menu, File → New Custom Testbench and save the name of the module `modulename_tb`. FPGA Editor opens the new testbench module with a few lines of code to act as a guideline for writing the testbench.

Note: The system recognizes any module as testbench only if it is saved as `modulename_tb`. 

Custom Testbench example
Simulating a testbench
After adding the testbench, click the Simulate Testbench button on the toolbar to launch the Icarus Verilog and GTKWave or from the main menu, Tools → Simulation. The simulation stage handled by Icarus Verilog is performed in the background, while the GTKWave has a visual representation. If the selected testbench is correct and contains no syntax errors, then the GTKWave software will launch automatically. In case it does not launch, check the Logger for any syntax related issues in the testbench code and make necessary changes.

2.5.9 Macrocell Editor
Macrocell Editor is a tool which allows you to create the desired circuit in a schematic view. Launch the tool by clicking the corresponding button on the toolbar, or go to main menu, Window → Macrocell
The **Macrocells Library** panel contains the list of components, which you can use for circuit design. Click the desired component from the library and then the work area to add it to the circuit. Clicking two ports creates the connection between the macrocells. You can also change the name of the input/output pin components.

![Macrocell Editor](image)

**Macrocell name change**

To generate the Verilog code based on the created design, click the **Generate Verilog** button. The created Verilog code is listed as `mcm_generated` under the **Custom Code** section of the control panel. You can use this code for the synthesis along with other modules added in the design.

![Verilog generation](image)
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.

![GreenPAK Advanced Development Platform Diagram]

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:

<table>
<thead>
<tr>
<th>Debug Configuration</th>
<th>Read</th>
<th>Power source selector</th>
</tr>
</thead>
<tbody>
<tr>
<td>Device selector</td>
<td>Program</td>
<td>TP map</td>
</tr>
<tr>
<td>Emulation</td>
<td>Project data window</td>
<td>LEDs ON and LEDs OFF</td>
</tr>
<tr>
<td>Emulation (sync)</td>
<td>Generator controls</td>
<td>Info details</td>
</tr>
<tr>
<td>Test Mode</td>
<td>Expansion connectors</td>
<td></td>
</tr>
</tbody>
</table>

Find a comparison of the controls available on different GreenPAK boards in the appendix, section 6.3 *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 micro-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

Find a comparison of the controls available on different GreenPAK boards in the appendix, section 6.3 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 *DIP Proto Board*. 

![GreenPAK Lite Development Platform](image)

- Reset button
- I2C line connector
- Expansion connector
- Two socket connector
- LEDs

---

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

<table>
<thead>
<tr>
<th>Feature</th>
<th>Debug Configuration</th>
<th>Device selector</th>
<th>External device modes</th>
<th>I2C Reset</th>
<th>Emulation</th>
<th>Emulation(sync)</th>
<th>Test Mode</th>
<th>TP map</th>
</tr>
</thead>
<tbody>
<tr>
<td>Debug Configuration</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Device selector</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>External device modes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I2C Reset</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Emulation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Emulation(sync)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Test Mode</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TP map</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LEDs ON and LEDs OFF</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Voltage level controls</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Info details</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Expansion connectors</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Power source selector</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**GreenPAK Lite Development Platform controls**

Find a comparison of the controls available on different GreenPAK boards in the appendix, section 6.3 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).
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

Find a comparison of the controls available on different GreenPAK boards in the appendix, section 6.3 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* (Va) 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 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 an 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.
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, connect the platform to the computer via a USB Type-C cable and connect the power supply. 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**

Find a comparison of the controls available on different FPGA boards in the appendix, section 6.3 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**

Find a comparison of the controls available on different FPGA boards in the appendix, section 6.3 *Debugging Controls feature availability.*
3.3 PowerPAK Devices

3.3.1 PowerPAK Development Platform

The PowerPAK Development Platform provides full debugging capabilities for the Power GreenPAK chips family. The Development Board has configurable voltage sources and GPIO control capability.

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**
- **Device selector**
- **Emulation**
- **Sync**
- **Test Mode**

**PowerPAK Development Platform controls**

Find a comparison of the controls available on different PowerPAK boards in the appendix, section 6.3 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**
- **Emulation**
- **Sync**
- **Test Mode**
- **GPIO SW Control**
- **Voltage level controls**
- **PowerPAK voltage controls**
- **Info details**

Find a comparison of the controls available on different PowerPAK boards in the appendix, section 6.3 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](image)

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

- Connect a chip with wires to the Serial Debugger pins (CLK, SDA, GND). If you use SLG5100xCRT Socket Eval., use the DUT I2C connector.

- Ensure the Evaluation board’s CS pin has been pulled-up to VDD or VDDIO. Connect the external power supply to VDD and VDDIO inputs.

- 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:

<table>
<thead>
<tr>
<th>Debug Configuration</th>
<th>Sync</th>
<th>Info details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Device selector</td>
<td>Test Mode</td>
<td>Voltage level controls</td>
</tr>
<tr>
<td>Emulation</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

![PowerPAK Serial debugger controls](image-url)
This chapter presents the methods to keep your software version up to date and introduces various approaches for updating software. Additionally, you will discover how to create and save block diagrams in a graphical format for integration into datasheets and other project-related documents. Lastly, you will gain insights into debug tool-related instructions designed to enhance your implementation process and ensure a smooth experience while implementing your designs.
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.

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](image)

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](image)

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

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

5.1 Failed Socket Test

Here is the list of steps to solve the most common causes of the Socket Test procedure failure:

- Ensure the contacts connecting a socket and chip pins are clean and undamaged
- Disconnect any external circuits or signal sources (generator, voltage supply, etc) connected to a board or a socket itself
  \textbf{Note:} External devices connected to expansions connectors do not affect the Socket Test procedure.
- Reconnect a board to a USB port
- Use an empty chip, if possible
  \textbf{Note:} Certain combinations of chip memory protection bits or specific I2C configurations on already programmed silicons may cause Socket Test to fail.

\textbf{SLG47011V Evaluation Board}

- \textit{Socket Test failed on Test Point 10} → Check the PIN15/VREF jumper position. This jumper should connect PIN15 to Test Point 10. To pass Socket Test, please try the following configuration:
SLG47011V Evaluation Board, default PIN15/VREF jumper position
In the appendix, you can find helpful information for the Go Configure Software Hub user experience enhancement.
6.1 Main menu commands

File

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

Edit

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

View

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

Tools

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

Options

- **Settings** set the default projects folder, autosave interval, toolbars position, recovery, shortcuts, and update options

Help

- **Help** show help window
<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Guides</td>
<td>open User guides on the web</td>
</tr>
<tr>
<td>Legend box</td>
<td>show the color legend box</td>
</tr>
<tr>
<td>Renesas web site</td>
<td>open the Renesas official website</td>
</tr>
<tr>
<td>Software and documentation</td>
<td>open Software and Documentation web page</td>
</tr>
<tr>
<td>Renesas web store</td>
<td>open Renesas chip store</td>
</tr>
<tr>
<td>Design support</td>
<td>web page with training courses and videos</td>
</tr>
<tr>
<td>Contact Us</td>
<td>web form with request</td>
</tr>
<tr>
<td>Social</td>
<td>Renesas in social networks</td>
</tr>
<tr>
<td>Application Notes</td>
<td>open examples web page</td>
</tr>
<tr>
<td>Datasheet</td>
<td>open documentation web page</td>
</tr>
<tr>
<td>Updater</td>
<td>open software update tool</td>
</tr>
<tr>
<td>About Go Configure Software Hub</td>
<td>show information about software tools version modification</td>
</tr>
</tbody>
</table>
### 6.2 Keyboard and mouse controls

#### Basic tools

<table>
<thead>
<tr>
<th>Action</th>
<th>Windows/Linux controls</th>
<th>macOS controls</th>
</tr>
</thead>
<tbody>
<tr>
<td>NVM Viewer</td>
<td>F2</td>
<td>F2</td>
</tr>
<tr>
<td>Component properties</td>
<td>F3</td>
<td>F3</td>
</tr>
<tr>
<td>Apply changes</td>
<td>Ctrl + A</td>
<td>Command + A</td>
</tr>
<tr>
<td>Revert changes</td>
<td>Ctrl + U</td>
<td>Command + U</td>
</tr>
<tr>
<td>Reset settings to default</td>
<td>Ctrl + Shift + R</td>
<td>Command + Shift + R</td>
</tr>
<tr>
<td>Reset connections to default</td>
<td>Ctrl + Shift + C</td>
<td>Command + Shift + C</td>
</tr>
<tr>
<td>Component List</td>
<td>F4</td>
<td>F4</td>
</tr>
<tr>
<td>Filter on the Component List</td>
<td>Ctrl + F</td>
<td>Command + F</td>
</tr>
<tr>
<td>Rule Checker</td>
<td>F5</td>
<td>F5</td>
</tr>
<tr>
<td>Debug tool</td>
<td>F9</td>
<td>F9</td>
</tr>
</tbody>
</table>

#### Components and connections

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

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

## Import model/subcircuit

<table>
<thead>
<tr>
<th>Action</th>
<th>Windows/Linux controls</th>
<th>macOS controls</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open search field*</td>
<td>Ctrl + F</td>
<td>Command + F</td>
</tr>
<tr>
<td>Close search field*</td>
<td>Esc</td>
<td>Esc</td>
</tr>
</tbody>
</table>

## Debug

<table>
<thead>
<tr>
<th>Action</th>
<th>Windows/Linux controls</th>
<th>macOS controls</th>
</tr>
</thead>
<tbody>
<tr>
<td>Emulation</td>
<td>Shift + E</td>
<td>Shift + E</td>
</tr>
<tr>
<td>Program</td>
<td>Shift + P</td>
<td>Shift + P</td>
</tr>
<tr>
<td>Read</td>
<td>Shift + R</td>
<td>Shift + R</td>
</tr>
<tr>
<td>Test Mode</td>
<td>Shift + T</td>
<td>Shift + T</td>
</tr>
<tr>
<td>NVM Data</td>
<td>Shift + N</td>
<td>Shift + N</td>
</tr>
<tr>
<td>Info</td>
<td>Shift + I</td>
<td>Shift + I</td>
</tr>
<tr>
<td>Log</td>
<td>Shift + L</td>
<td>Shift + L</td>
</tr>
</tbody>
</table>

## I2C reconfigurator

<table>
<thead>
<tr>
<th>Action</th>
<th>Windows/Linux controls</th>
<th>macOS controls</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create snapshot*</td>
<td>Shift + A</td>
<td>Shift + A</td>
</tr>
<tr>
<td>Add delay*</td>
<td>Shift + D</td>
<td>Shift + D</td>
</tr>
<tr>
<td>Move up*</td>
<td>Shift + up</td>
<td>Shift + up</td>
</tr>
<tr>
<td>Move down*</td>
<td>Shift + down</td>
<td>Shift + E</td>
</tr>
<tr>
<td>Send one snapshot*</td>
<td>F7</td>
<td>F7</td>
</tr>
<tr>
<td>Send all snapshots*</td>
<td>F6</td>
<td>F6</td>
</tr>
<tr>
<td>Stop sending snapshots*</td>
<td>Shift + F7</td>
<td>Shift + F7</td>
</tr>
</tbody>
</table>

## ASM editor

<table>
<thead>
<tr>
<th>Action</th>
<th>Windows/Linux controls</th>
<th>macOS controls</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set link</td>
<td>Ctrl + W</td>
<td>Command + W</td>
</tr>
<tr>
<td>Erase link</td>
<td>Ctrl + E</td>
<td>Command + E</td>
</tr>
<tr>
<td>Clear</td>
<td>Ctrl + N</td>
<td>Command + N</td>
</tr>
<tr>
<td>Undo</td>
<td>Ctrl + Z</td>
<td>Command + Z</td>
</tr>
<tr>
<td>Redo</td>
<td>Ctrl + Y</td>
<td>Command + Y</td>
</tr>
<tr>
<td>Help</td>
<td>F1</td>
<td>F1</td>
</tr>
</tbody>
</table>
## General

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

## Software tool launcher

<table>
<thead>
<tr>
<th>Action</th>
<th>Windows/Linux controls</th>
<th>macOS controls</th>
</tr>
</thead>
<tbody>
<tr>
<td>Welcome tab*</td>
<td>Ctrl + 1</td>
<td>Command + 1</td>
</tr>
<tr>
<td>Recent files tab*</td>
<td>Ctrl + 2</td>
<td>Command + 2</td>
</tr>
<tr>
<td>Develop tab*</td>
<td>Ctrl + 3</td>
<td>Command + 3</td>
</tr>
<tr>
<td>Demo tab*</td>
<td>Ctrl + 4</td>
<td>Command + 4</td>
</tr>
<tr>
<td>Recovery files tab*</td>
<td>Ctrl + 5</td>
<td>Command + 5</td>
</tr>
<tr>
<td>Datasheets*</td>
<td>Ctrl + 6</td>
<td>Command + 6</td>
</tr>
<tr>
<td>User guide*</td>
<td>Ctrl + 7</td>
<td>Command + 7</td>
</tr>
</tbody>
</table>

**Note:** Non-configurable controls are marked with an asterisk (*).
6.3 Debugging Controls feature availability

**GreenPAK platforms**

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

**FPGA platforms**

<table>
<thead>
<tr>
<th>Feature</th>
<th>ForgeFPGA advanced board</th>
<th>ForgeFPGA Evaluation board</th>
</tr>
</thead>
<tbody>
<tr>
<td>Debug Configuration</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Chip procedures</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Flash procedures</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Generator controls</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>TP map</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Voltage level controls</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Info details</td>
<td>✓</td>
<td></td>
</tr>
</tbody>
</table>

**PowerPAK platforms**

<table>
<thead>
<tr>
<th>Feature</th>
<th>PowerPAK Development Platform</th>
<th>SLG51002CTR Demo Board</th>
<th>Serial Debugger(with SLG5100x)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Debug Configuration</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Device selector</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Chip procedures</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>GPIO SW Control</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TP map</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Voltage level controls</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>PowerPAK voltage controls</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Info details</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
</tbody>
</table>
### 6.4 Abbreviations

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

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2.0</td>
<td>2023-12-22</td>
<td>• Added a new Memory Table Editor section</td>
</tr>
</tbody>
</table>
| 2.1.0   | 2023-12-08 | • Added a new FPGA Editor section  
• Added a new Troubleshooting section with the Failed Socket Test solutions  
• Added a new section with the Voltage Monitor tool  
• Extended the I2C Tools section with the new Memory Table tool  
• Extended the I2C Tools section with the new Data Buffers tool  
• Extended Debugging Controls for the PowerPAK Dev. Platform with the LEDs ON and LEDs OFF feature description  
• Extended Debugging Controls for the PowerPAK Dev. Platform with the Device selector (external chip support) feature description  
• Updated the Board Selector in the Debugging Controls with the new information about the Socket Test  
• Updated the notification of the processing data time for Transient Simulation Analysis  
• Updated the System Requirements section  
• Improved the Demo Board and Demo mode section  
• Improved the Hardware source image list  
• Fixed typos                                                                 |
| 2.0.1   | 2023-10-25 | • Fixed typos  
• Added 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 who are 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 to develop 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 from 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.1.01 Jan 2024)

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-us/.

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