Please note that Cypress is an Infineon Technologies Company.
The document following this cover page is marked as “Cypress” document as this is the company that originally developed the product. Please note that Infineon will continue to offer the product to new and existing customers as part of the Infineon product portfolio.

Continuity of document content
The fact that Infineon offers the following product as part of the Infineon product portfolio does not lead to any changes to this document. Future revisions will occur when appropriate, and any changes will be set out on the document history page.

Continuity of ordering part numbers
Infineon continues to support existing part numbers. Please continue to use the ordering part numbers listed in the datasheet for ordering.

ModusToolbox® Device Configurator Guide

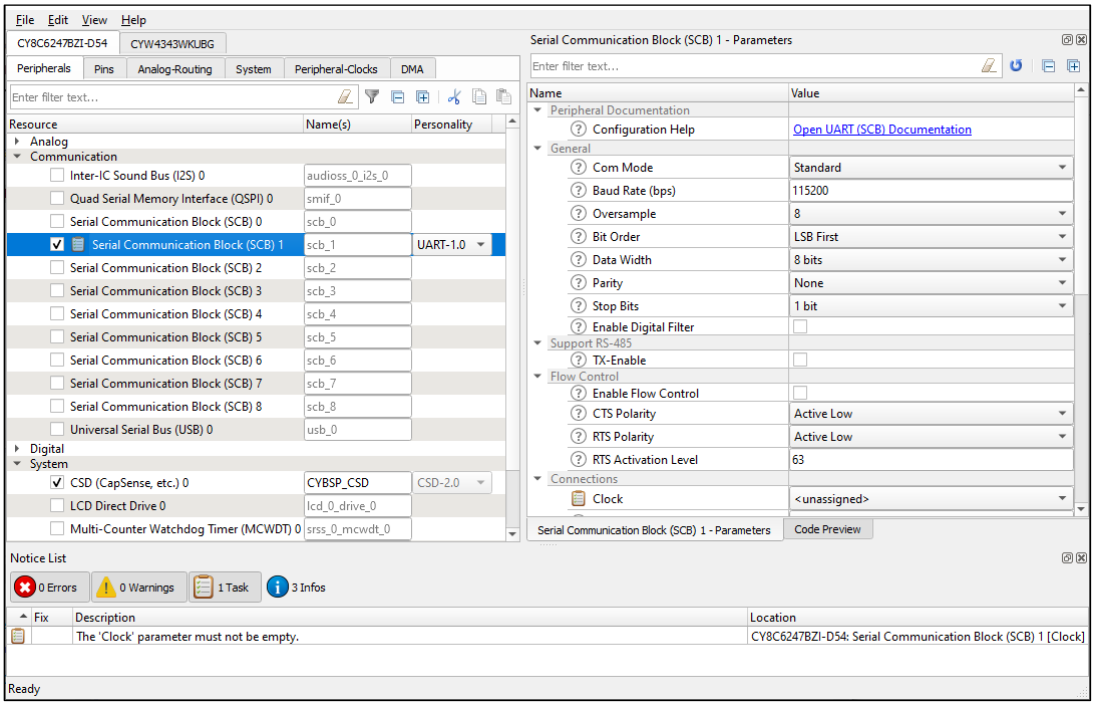
Version 3.0

Overview

The Device Configurator is part of a collection of tools included with the ModusToolbox software. Use the Device Configurator to enable and configure device peripherals, such as clocks and pins, as well as standard MCU peripherals that do not require their own tool.

Note If you make changes to Device Configurator settings, you are making changes to a standard Board Support Package (BSP) library, which will cause the repo to become dirty. Additionally, if the BSP is in the shared asset repository, changes will impact all applications that use the shared BSP. If you wish to make changes, you should first copy the configuration information to the application and override the BSP configuration or create a custom BSP. For more information, refer to the ModusToolbox User Guide.

Some complex peripherals, such as CapSense®, SegLCD, etc., have specialized configuration tools, and the Device Configurator provides links to launch those separate tools (see Launch Other Configurators). After configuring and saving a particular device's settings, the Device Configurator generates firmware for use in your application (see Code Generation).



Definitions

The following are the terms used in this guide that you may not be familiar with:

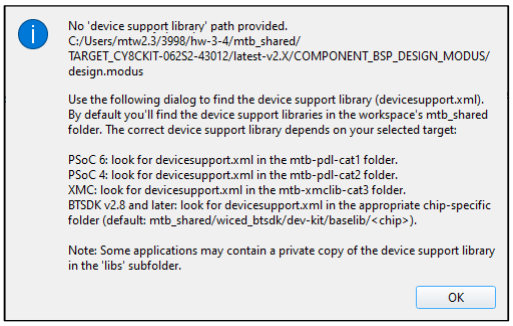
- **Resource –** Includes peripherals, pins, clocks, etc. used in an application.
- **Configurator –** A GUI-based tool used to configure a resource.
- **Application –** One or more projects related to each other.
- **Personality –** A file that defines a resource behavior.

- **Device Support Library –** A device support library provides critical firmware and device data files to configurators. Device support libraries are identified with a file named *devicesupport.xml*. It is used to find things like other tools, devices, and personalities.

Launch the Device Configurator

You can launch the Device Configurator in various ways as described in this section; however, the tool's configuration (*design.modus*) file requires an association to an application in order to obtain device configuration information. The easiest way to do that is to launch the Device Configurator from an application within the Eclipse IDE. If you prefer not to use the Eclipse IDE, you can launch the tool using a make command inside the application directory.

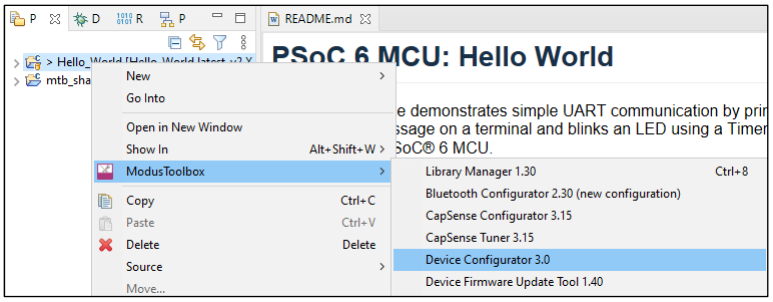
If you launch the Device Configurator and open a configuration file **without** an association to an application, the tool may open the following dialog, indicating that it cannot find the device support library:



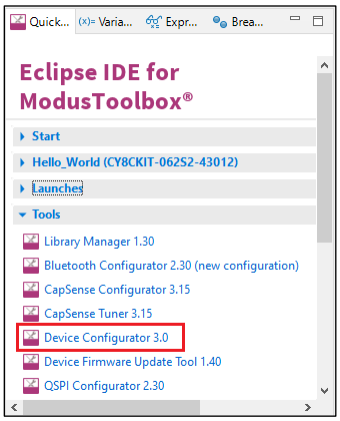
In this case, after closing the dialog, you must navigate to the appropriate directory to find the device support library file.

From the Eclipse IDE

To launch the Device Configurator from the Eclipse IDE, right-click on the project and select **ModusToolbox > Device Configurator <version>**.



You can also open the Device Configurator by clicking the link in the Eclipse IDE for ModusToolbox Quick Panel:



This opens the Device Configurator using the application's *design.modus* file, which contains all the required hardware configuration information about the device for the application. When you save updates to the *design.modus* file, the tool generates/updates source code in the *GeneratedSource* directory next to the *design.modus* file. Eclipse IDE applications use the *design.modus* file and generated source code in future application builds.

From the Command Line

If you prefer not to use the Eclipse IDE, you can launch the Device Configurator from the command line. Run the appropriate shell program that is configured to use the ModusToolbox build system. Navigate to your application directory where the makefile is located, and type the following command:

```
make config
```

This also opens the Device Configurator using the application's *design.modus* file, the same as opening the tool from the Eclipse IDE.

For more information about the ModusToolbox build system and make commands, refer to the ModusToolbox User Guide.

As a Stand-Alone Tool

You can launch the Device Configurator as a stand-alone tool. As stated previously, if you launch it this way, you will likely have to specify the path to the device support library. By default, the Device Configurator is installed here:

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

On Windows, launch the tool from the **Start** menu. For other operating systems, navigate to the install location and run the executable.

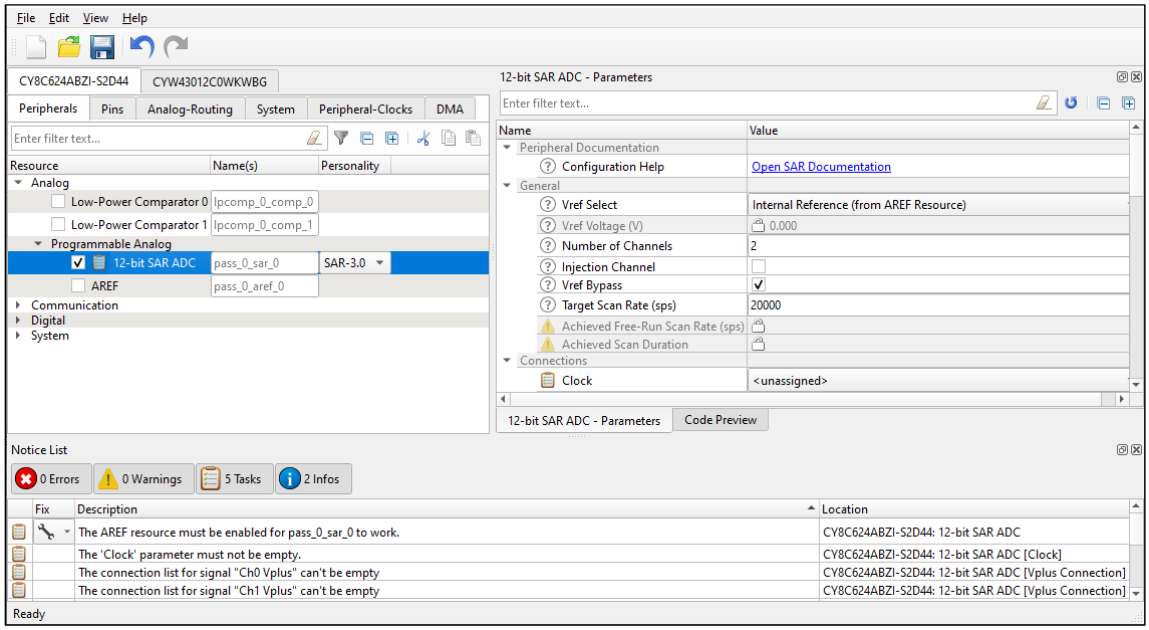
When it is run as a stand-alone tool, the Device Configurator opens without any content. You can either open a specific **.modus* file or create a new one. See Menus for more information.

- If you create a new **.modus* file, specify the file name and location to store the new **.modus* file, and select a part number for the application. See Create Design dialog for more information.
- If you open an existing **.modus* file from a **non**-Eclipse IDE application, it will be your preferred working environment flow.
- If you open a *design.modus* file for an Eclipse IDE application, it will be the same flow as if you opened it from an Eclipse IDE application.

Quick Start

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

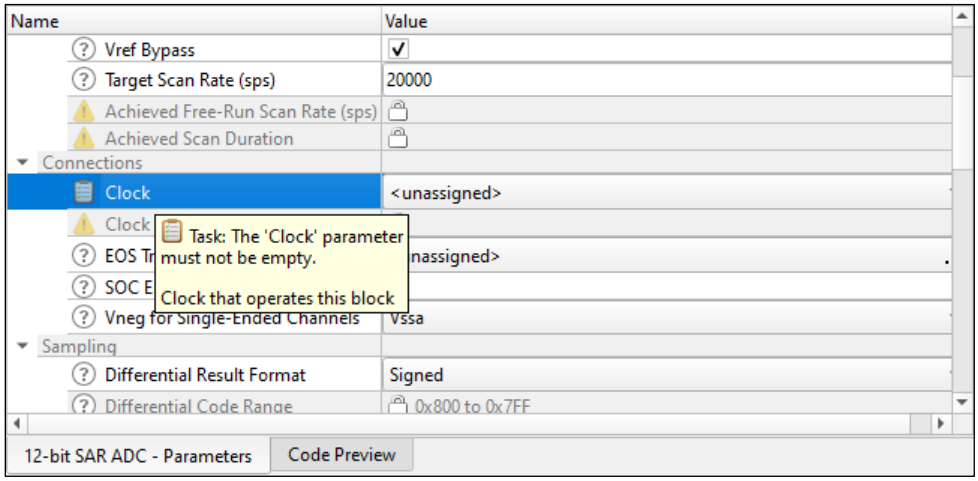
1. Launch the Device Configurator.
2. Enable a desired peripheral on the Peripherals Tab by clicking the enable check box. Notice the Parameters Pane becomes populated with fields.



3. Notice also that a new task may appear in the Notice List Pane.

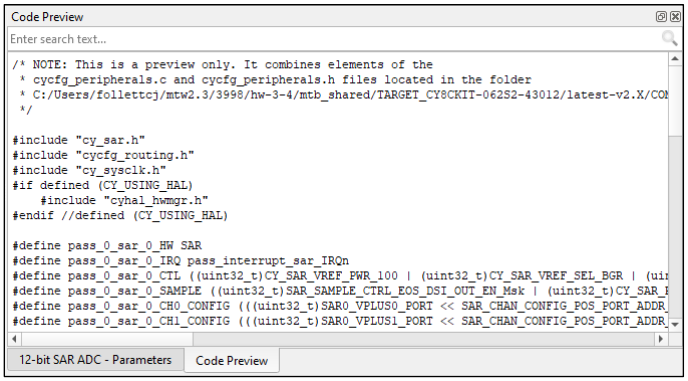
   See Icons for descriptions of the various icons displayed in the Device Configurator.

4. Double-click on a task to jump to the parameter that needs to be addressed.



5. Select an appropriate parameter value and the task should be removed from the Notice List.

6. When all tasks have been completed, select the Code Preview Pane to see a preview of the code that will be generated upon saving.



7. Use the various tabs to enable and configure other resources as needed in the same manner as peripherals.
8. Save the **.modus* file to generate source code.

   The Device Configurator generates code into a “GeneratedSource” directory in your Eclipse IDE application, or in the same location you saved the **.modus* file for non-Eclipse IDE applications. That directory contains the necessary source (.c) and header (.h) files that use the relevant driver APIs to configure the hardware. Application code then uses this code to configure the system.
9. Use the appropriate API in your application code.

Code Generation

The Device Configurator generates structures, defines, and initialization code for the blocks on the chip. All generated code is located in the *GeneratedSource* folder next to the **.modus* file. Refer to the Peripheral Driver Library (PDL) API Reference for more information about this code. Each enabled resource has a link to the specific driver documentation in the Parameters Pane.

Note The Device Configurator generates code based on the hardware resources that are enabled. If a resource is not enabled, no configuration will be generated for it. This means the resource will retain its default reset state. In most cases, this is powered off. However, some features are enabled by default, such as debug connectivity. To disconnect these features, you must call the appropriate API functions to turn the feature off.

The defines and structures are all named based on the resource that created it. In general, these have the form *<resource-name>_config*. These structures can be passed to the PDL functions that are responsible for configuring the hardware block.

The functions are specific to a resource category and have names of the form *init_cycfg_<resource-category>*. The init function for a particular resource type is located in *GeneratedSource/cycfg_<resource-category>.h*. There are also the *cycfg.h* and *cycfg.c* files. Include the *cycfg.h* file in your application to access the generated header files. The *cycfg.c* file implements init_cycfg_all(), which calls all other generated functions, for example init_cycfg_pins().

The resource types include:

- Clocks: Peripheral clocks.
- Connectivity: Configuration of the programmable analog and digital routing resources.
- Peripherals: Fixed function analog and digital peripherals.
- System: Overall configuration function to setup all power and clock options.

It is up to you to make use of the generated code based on the application's needs. This can be done as part of the application’s main() loop.

GUI Description

The Device Configurator GUI contains menus, icons, tabs, and several panes used to configure MCU peripherals.

Menus

File

- **New** – Creates a new **.modus* file. See Create Design dialog for more information. The current file, if any, will be closed.
- **Open** – Opens an existing **.modus* file. The current file, if any, will be closed.
- **Close** – Closes the current file. If there are pending changes, you will be prompted to save the file.
- **Save** – Saves the current file and generates code for the related application. If there are errors in the application, a dialog will indicate such. The file will still be saved.
- **Save As** – Saves the current file with a different name and/or location.
- **Open in System Explorer** – This opens your computer's file explorer tool to the folder that contains the **.modus* file.
- **Change Library** – Opens a dialog to select a different device support library (*devicesupport.xml*) file used for resource Personalities.

  The path to this file is usually stored relative to the **.modus* file. If for any reason the device support library cannot be found (for example, the **.modus* file has been moved on disk), you will be prompted to manually enter the path.

  Note This menu option is not available if a command-line override has been used.
- **Recent Files** – Shows up to five recent files that you can open directly.
- **Update All Personalities** – Use this item to update all resource Personalities.

  For example, if you load a **.modus* file made with an older device support library, there might be many warnings in the Notice List Pane to update personalities or that a personality is no longer supported. Each warning must be addressed, and doing so one at a time can be annoying. The **Update All Personalities** menu item addresses them all at once.
- **Exit** – Closes the tool. You will be prompted to save any pending changes.

Edit

- **Undo** – Undoes the last action or sequence of actions.
- **Redo** – Redoes the last undone action or sequence of undone actions.

View

Contains toggles to hide or show different panes. All panes are shown by default. There is also a command to show or hide the Toolbar (hidden by default).

Help

Opens this document and the **About** box.

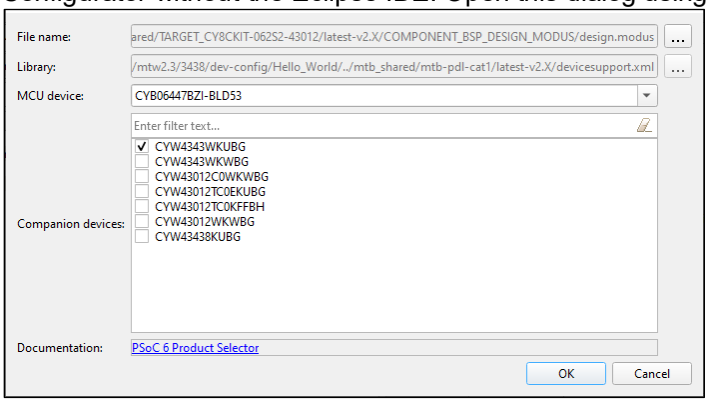
Icons

When configuring various options with this tool, you will see the following icons:

	Indicates there is a tooltip. Hover over the icon to display a brief message about the setting.
	Enables or disables a specific resource.
	There may be occasions where an error, warning, task, or info icon displays for an enabled resource. See Notice List Pane for more details.
	When shown in **Parameters**, this indicates that it is a read-only field. When shown for a **Resource**, this indicates the resource is locked and disabled. There is a tooltip explaining why the resource is locked.
	When shown for a Resource, this indicates the resource is locked and enabled. There is a tooltip explaining why the resource is locked.
	After assigning a signal, clicking this icon jumps to the linked resource(s).

Create Design Dialog

Use the Create Design dialog to create new *.modus files. You usually do this when launching the Device Configurator without the Eclipse IDE. Open this dialog using the **File > New** option or pressing [**Ctrl**] + [**N**].

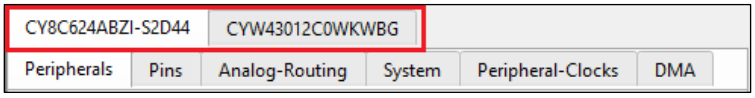


To create a new *.modus file:

1. Click the **Browse** [**. . .**] button next to **File name**, navigate to the location to save the new file, and enter a file name.
2. Click the **Browse** [**. . .**] button next to **Library**, navigate to the location of the library (*devicesupport.xml*) file, and select it.
3. Select an **MCU Device** from the pull-down menu.
4. If applicable, select one or more **Companion Device(s)** from the list.
5. Click **OK** to close the dialog and load the new *.modus file into the Device Configurator.

Device Tabs

The Device Configurator can be used to configure multiple devices. All devices selected in the Create Design dialog display in top-level tabs above the **Resource** tabs.



All the settings for each **Device** tab are configured separately.

Note If you need to update or change devices, you must use the command line option or make changes manually by editing the makefile. You must also make changes to the BSP makefile or you will encounter build errors. The best way to change devices is to create a custom BSP. Refer to the ModusToolbox User Guide for more details.

Resources Tabs

For PSoC 6 MCUs, the Device Configurator contains several tabs, each of which provides access to specific resources. For WICED Bluetooth and Wi-Fi devices, there are no separate tabs; resources are shown in a single pane, sometimes under collapsible trees.

When you enable a resource, or select an enabled resource, the Parameters pane displays various configuration options. As described under Icons, some enabled resources may contain errors, warnings, tasks, or infos that indicate some action might be required to resolve the issue. See Notice List Pane for more details.

Note Only the tabs relevant for a selected device are displayed, so some of the tabs may not be included for some devices.

- Peripherals – Options to enable any of the analog, digital, system, and communication hardware capabilities of the chip that are not related to the platform.
- Pins – Options for all the pin related resources.
- Analog-Routing – This tab shows all the analog resources, whether enabled or not, and how they connect. It also allows you to edit routes.
- System – Options for chip-wide configuration settings such as system clocks, power management, and debug interfaces.
- Peripheral-Clocks – Options for all the peripheral clocks.
- DMA – Provides configuration of the DMA channel and transaction descriptors.

Each of the tabs (except the Analog-Routing) has the following features:

- **Filter** – The **Resource** column shows all available resources in an expandable tree. The filter box above the list of peripherals allows you to limit the peripherals shown in the tree as well as a **Hide** disabled resources filter button. There are also **Expand** and **Collapse** commands.
- **Cut, Copy, Paste** – Use these commands to move and copy settings from one resource of the same type to another.
  - When you use **Cut**, the settings will be copied to the clipboard, and the selected resource will be disabled.
  - When you use **Copy**, the settings will just be copied to the clipboard.
  - When you use **Paste**, the selected resource will be enabled if needed. The selected resource must support the same Personality name and version as the cut/copied resource.

- **Name(s)** – This displays the current resource name(s). This is an editable field where you can specify optional, alternate names for this resource. This is also used in generated code.

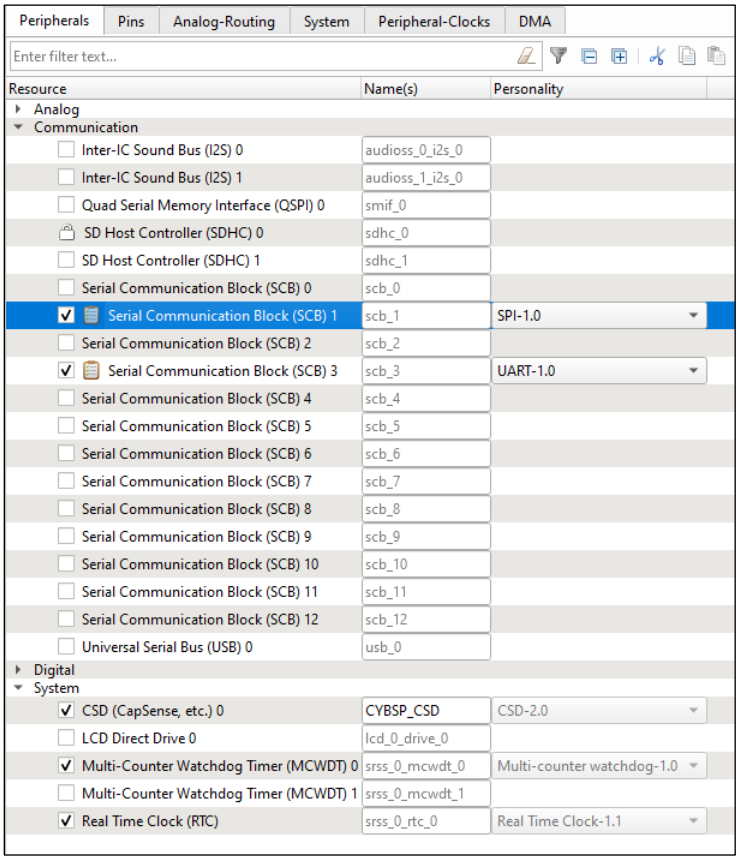
  Note Enter any string in this field. The tool converts the name into a legal C identifier and replaces non-legal characters with underscores. If entering more than one name, use a coma-separated list.

- **Personality** – Each resource has a "Personality" file that contains the information for the given resource.
  - Some peripherals, such as Serial Communication Block (SCB) and Timer, Counter, Pulse Width Modulator (TCPWM), have a pull-down menu to select a specific personality, such as UART, SPI, or I^2C.
  - Some peripherals have multiple personality versions from which you can select.
  - Some peripherals have a read-only field that only shows the name of this resource's personality file.

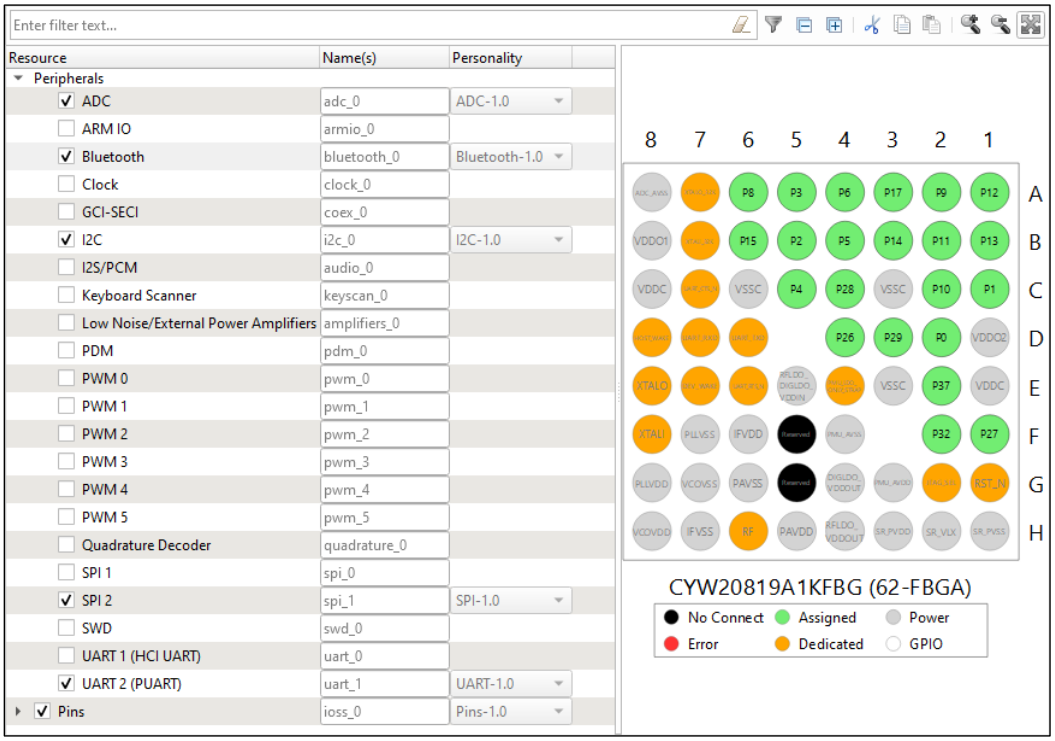
Peripherals

The **Peripherals** tab/tree is where you enable various analog, digital, system, and communication peripherals for the device to include in your application. The filter box and the hide disabled button above the list of peripherals allows you to limit the resources shown in the tree. This tab allows you to enter one or more Names for the resource, and it shows the selected Personality.

PSoC 6 Applications



Bluetooth Applications



Pins

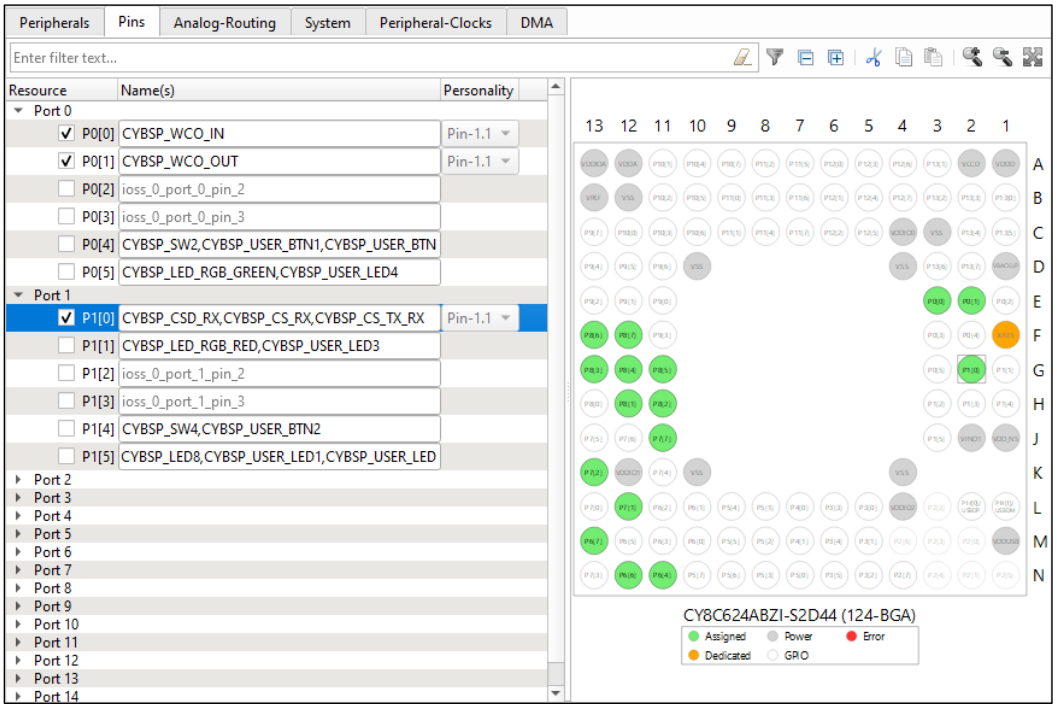
The **Pins** tab/tree is where you enable all the pin related resources. All available pins are shown in an expandable tree, arranged by port number. The filter box and the hide disabled button above the list of pins allows you to limit the pins shown in the tree. This tab allows you to enter one or more Names for the resource, and it shows the selected Personality.

The interactive pin package diagram shows the different states of the pins; there is a legend on the diagram. You can enable/disable a pin by double-clicking it in the diagram. There are also zoom commands to resize the diagram as needed. If you zoom the image larger than the frame area, scroll bars appear to move to different area of the diagram. You can also press the [**Alt**] key to use the pan tool.

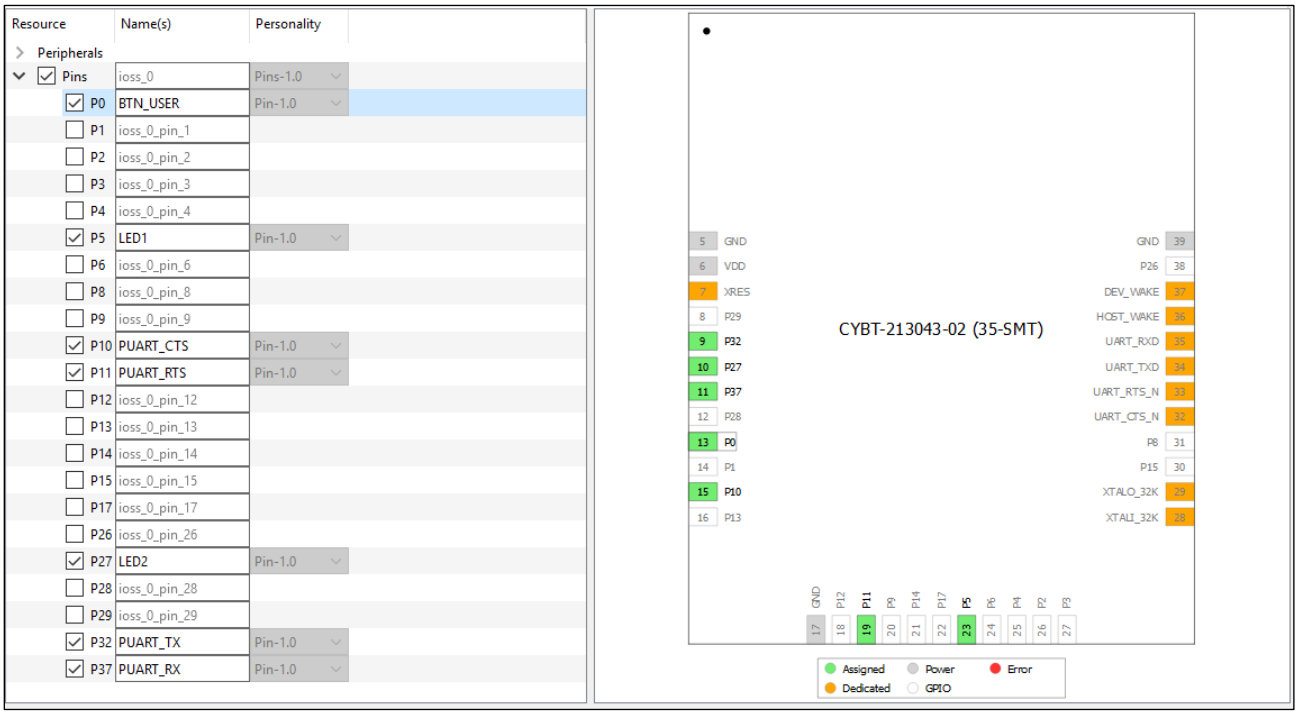
Pin states are shown in different colors:

- Black – No connect
- White – Disabled
- Green – Enabled
- Grey – Power/ground pins
- Orange – Fixed function pins
- Red – Error state
- Semi-transparent – The hardware resource's enabled state has been locked.

PSoC 6 Applications

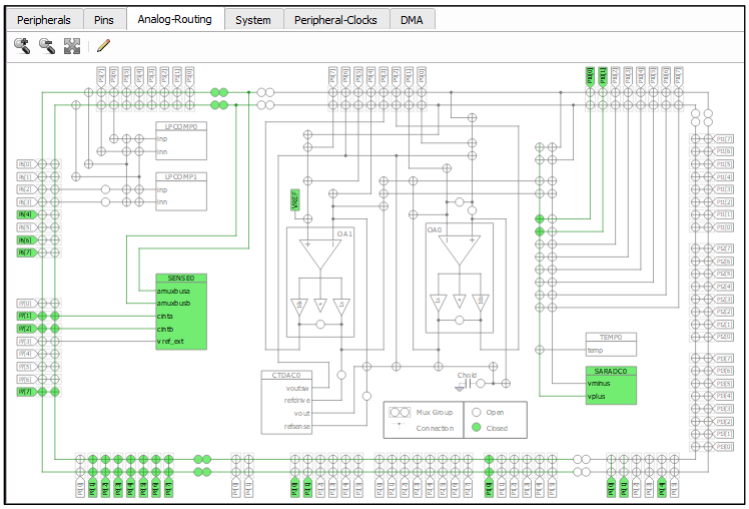


Bluetooth Applications



Analog-Routing Tab

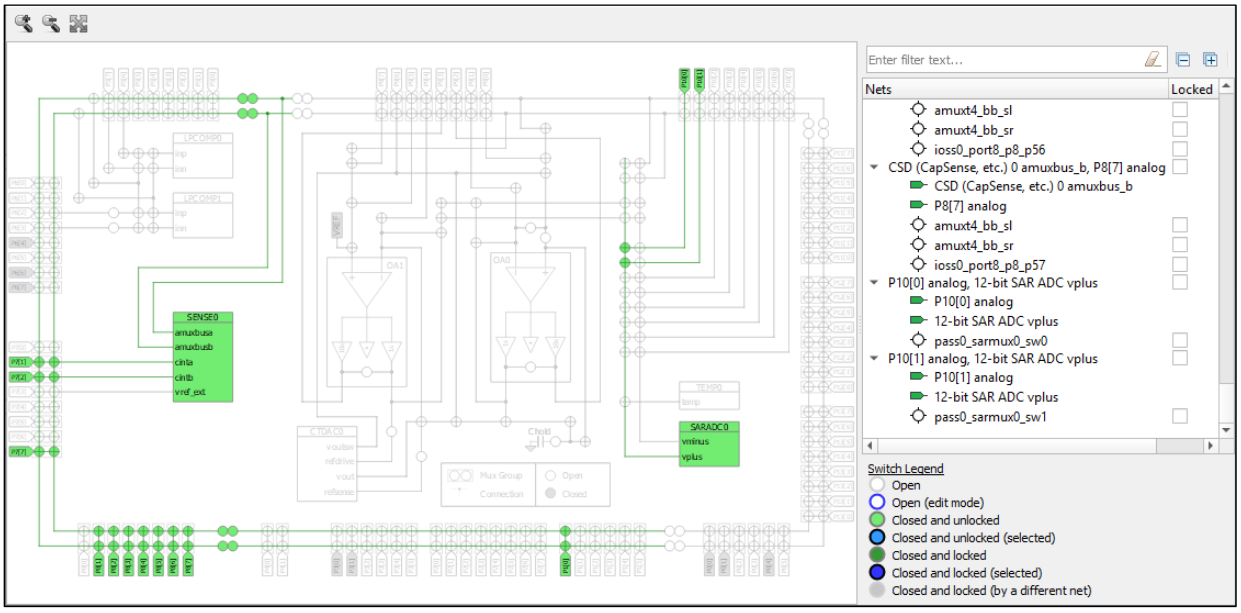
The **Analog-Routing** tab shows the various analog resources in your application. Enabled resources are green.



There are zoom commands to resize the diagram as needed. The **Edit** command opens the Analog Route Editor.

Analog Route Editor

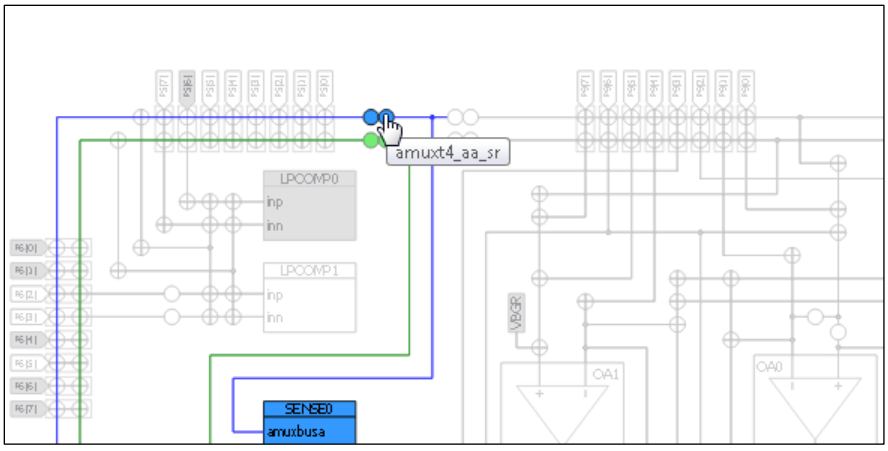
The Analog Route Editor allows you to manually edit the routing of analog resources in your application. It also provides the ability to lock-down all or some of the results.



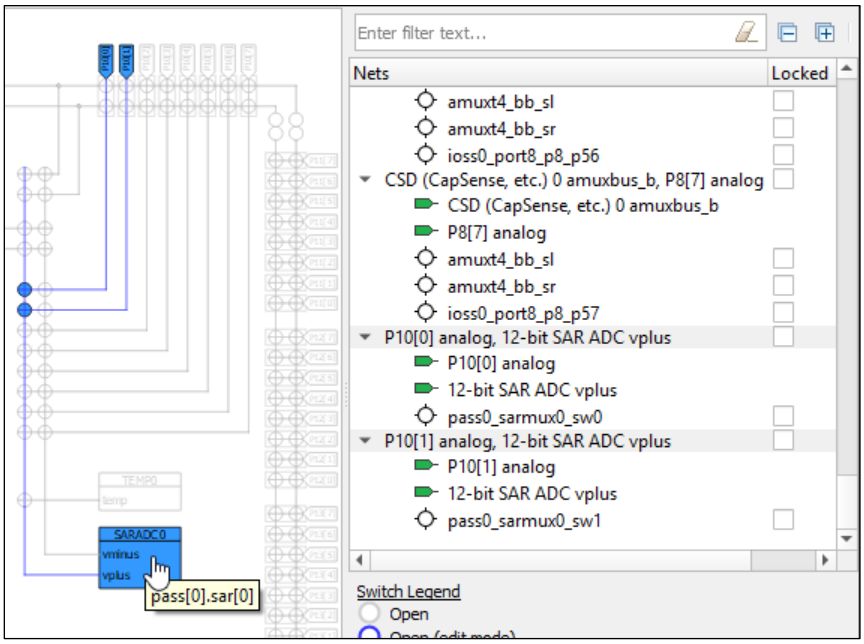
Note If there are configuration errors, complete routing results will not be available; only locked resources. If you open the Analog Editor in this error state, a warning message will display. You can still lock and unlock switches, but you won't get complete routing results as long as the configuration has errors.

Select a Resource

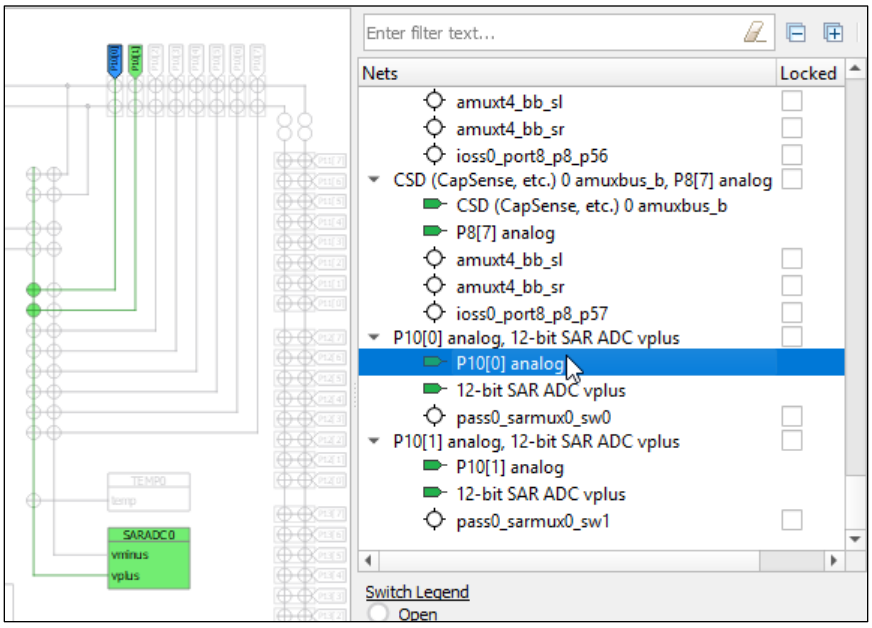
To select an analog resource, click on it. Any enabled (green) element in the tree can be selected. The resource and the associated route(s) become blue. Also, the **Edit Route** command appears on the toolbar. See Edit Route.



At the same time, the selected analog resource(s) is highlighted in the Nets tree.



You can also select items in the tree to highlight them in the diagram.



Edit Route

With an editable analog resource selected, click the **Edit Route** command to enable edit mode. If multiple routes are selected, a pull-down menu displays to select the route to edit. You cannot edit multiple routes at the same time.

In edit mode, the net tree shows only the applicable route entries, and you cannot select resources using the tree. However, the lock/unlock check boxes remain enabled for use. The inactive switches change color to indicate they can be selected to use for the route being edited.

Route changes are live with updates applied automatically as you make changes. Selecting a switch adds it to the current route in a locked state and the route tree is updated to reflect the modifications.

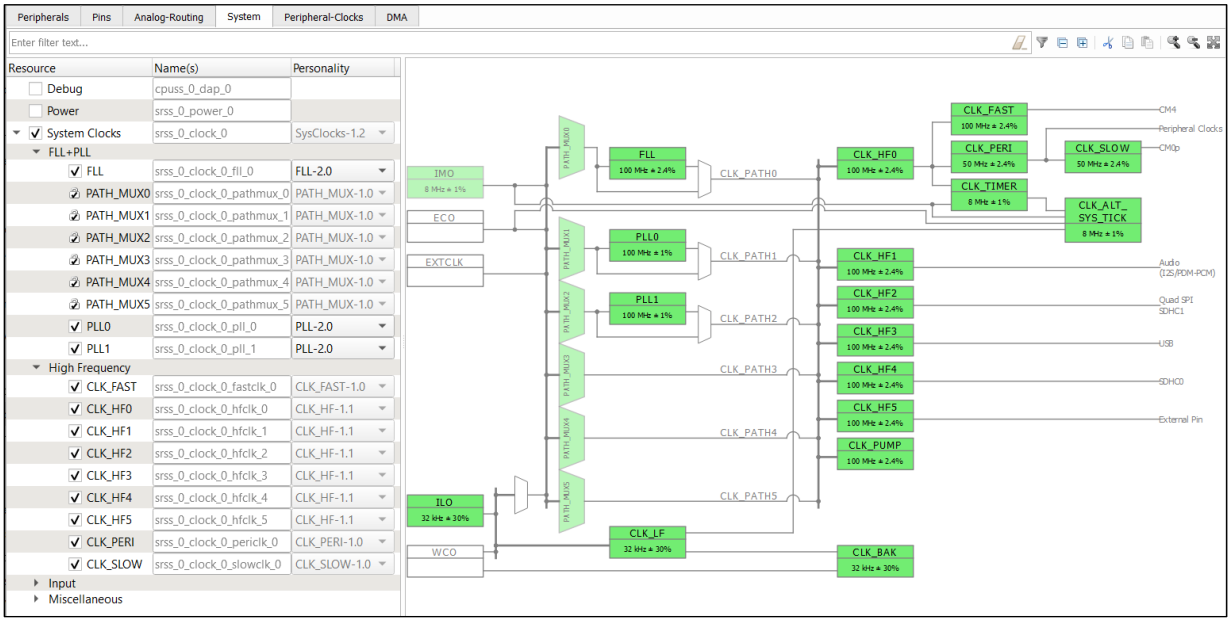
If a change results in an error, a message displays. The routes are automatically rolled back to the previous state, so you will lose at most the last invalid change.

The toolbar shows the **Finish edit** command to return the editor to selection mode.

Note If a route is edited so that it uses switches associated with a location where no personalities are instantiated, you must manually power on the containing block at startup in order for the switches to function. Refer to the *PDL API Reference Guide* and the *Device Technical Reference Manual* for more details.

System Tab

The **System** tab provides access to system-level items, such as system clocks, power management, and debug interfaces. All available resources are shown in an expandable tree. The filter box and the hide disabled button above the list of resources allows you to limit the items shown in the tree. This tab allows you to enter one or more Names for the resource, and it shows the selected Personality.

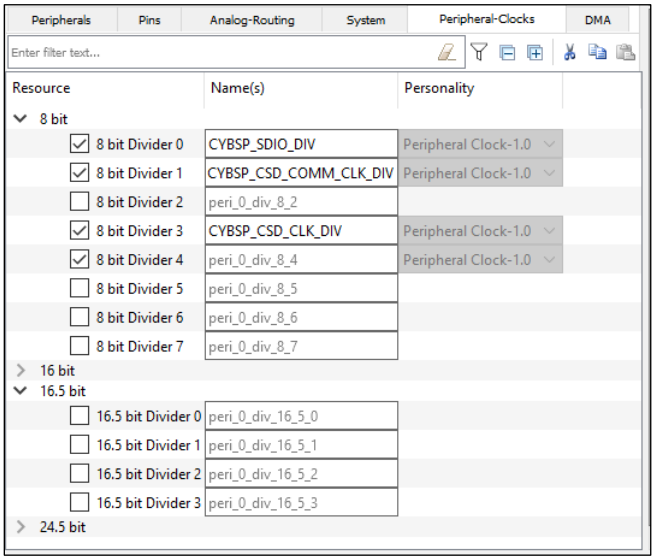


The interactive clock diagram shows all the system clocks and how they connect to each other. You can enable/disable a clock by double-clicking it in the diagram. Enabled clocks are green, disabled clocks are white, and clocks in error state are red. There are also zoom commands to resize the diagram as needed.

Note The semi-transparent (faded) elements in the diagram indicate that their enabled state is locked.

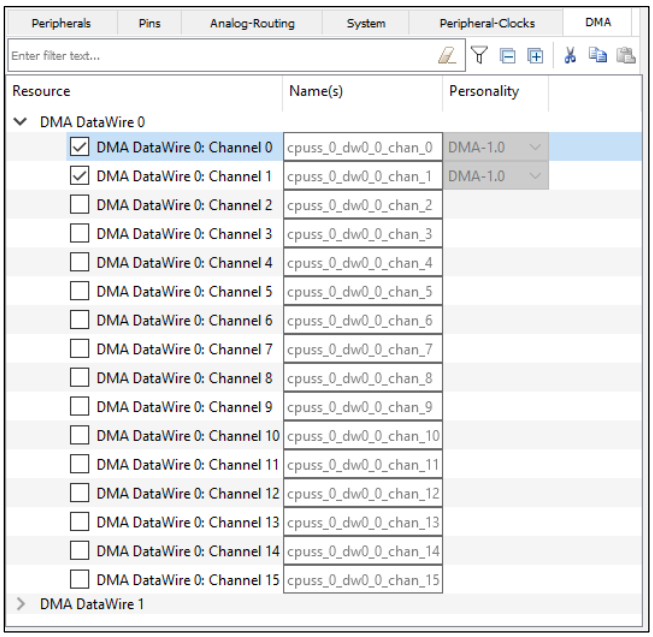
Peripheral-Clocks Tab

The **Peripheral-Clocks** tab lists all the clocks in a design used to drive the various peripherals. All available clocks are shown in an expandable tree. The filter box and the hide disabled button above the list of resources allows you to limit the items shown in the tree. This tab allows you to enter one or more Names for the resource, and it shows the selected Personality.



DMA Tab

The **DMA** tab lists all the DMA resources in the design. All available DMA channels are shown in an expandable tree. The filter box and the hide disabled button above the list of resources allows you to limit the items shown in the tree. This tab allows you to enter one or more Names for the resource, and it shows the selected Personality.



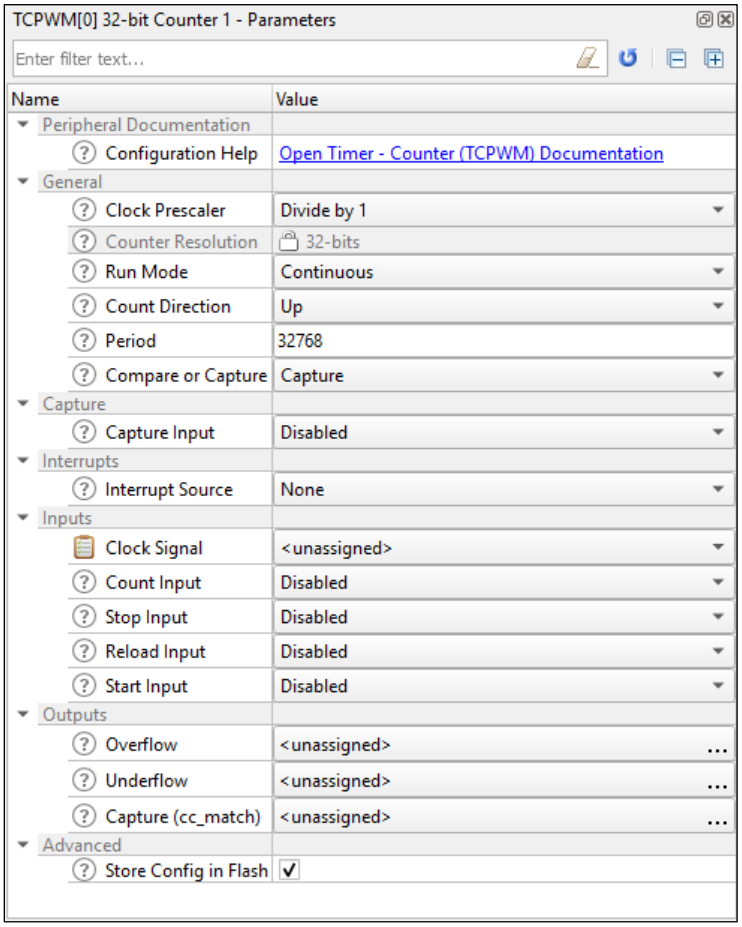
Panes

The Device Configurator tool contains the following primary panes that display information based on what is selected in a particular resource tab:

- Parameters – This pane shows the various parameters for any specific resource enabled in one of the tabs.
- Notice List – This pane shows any errors, warnings, tasks, and infos for the application.
- Code Preview – This pane shows a preview of the code that will be generated for the selected resource when you save the **.modus* file.

Parameters Pane

The **Parameters** pane contains all the parameters for a selected, enabled resource. This pane will show different parameters for each resource, grouped by various categories. For example, the parameters for the TCPWM peripheral are completely different than those for a pin resource. The filter box above the list of parameters allows you to limit the items shown in the pane. Some resources also provide a link to launch a separate configurator.



Configuration Help

Nearly all resources provide a link to open the Peripheral Driver Library (PDL) documentation to the specific driver. This is the Doxygen-generated HTML file located in the installation directory. To see links to the documentation, simply highlight a resource; you do not need to enable it.

Parameter Descriptions

As described under Icons, all parameters have a tooltip icon to indicate there is information about the parameter. Hover the mouse cursor over the icon to display a description of the parameter.

Parameter Values

Different parameter types have different ways to specify a value, as follows:

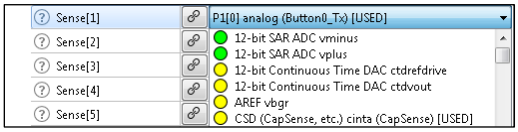
- **Pull-down Menu –** For parameters with a specific set of values, use the pull-down menu to select the appropriate value.
- **Selection Box –** For parameters with a variable set of values, click the ellipsis […] button to open a selection box. There, use the check boxes to select one or more appropriate values for the parameter.

  Note After selecting these parameter types, use the **Go To** button to jump to the selected resource.
- **Check Box –** For parameters with a true or false value, use the check box to enable or disable the parameter.
- **Text Box –** For parameters with editable values, type the value in the text box.

  Note Values preceded by '0' are interpreted as octal; values preceded by '0x', '0X', and '#' are interpreted as hexadecimal.

Signal Select Indicators

For parameters where you select a signal, there is a pull-down menu for single-select signals and a button to open a dialog for multi-select signals.



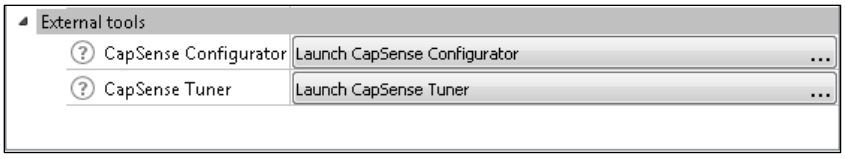
The signals have guidance icons next to them to indicate the status of the signal, as follows:

- Green – preferred
- Yellow – valid
- Red – constrained

After selecting one or more signals, use the **Go To** button to jump to the selected signal or open a dialog to select from multiple signals.

Launch Other Configurators

For peripherals with their own configuration tools (CapSense, SegLCD, etc.), the Device Configurator provides links to launch those separate configurators. After enabling the peripheral on the Peripherals Tab, the **Parameters** pane contains a **<Configurator>** parameter, where <Configurator> is the name of the other configurator.



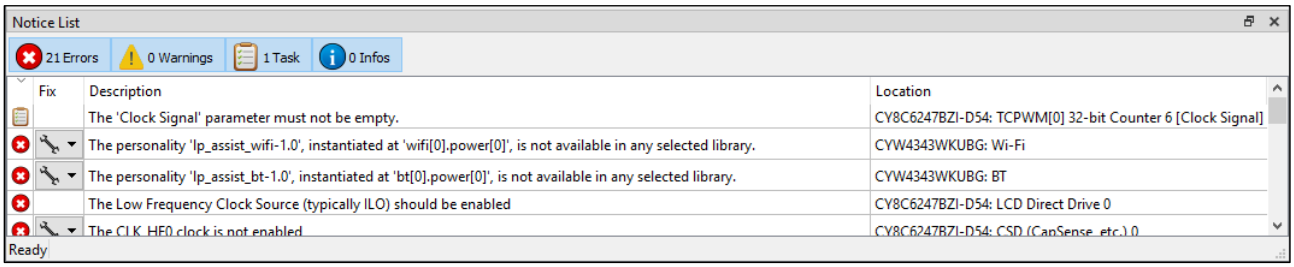
Click on **[Launch < Configurator > . . .]** to launch the configurator.

Note When launching another configurator from the Device Configurator, it passes information, such as the location of the **.modus* file and any configuration data, to that other configurator. Those other configurators can be launched independently from the Device Configurator. When launched independently, you will need to either

open or create the appropriate configuration file for that tool. If you want to use the configuration tools independently for the same application, make sure to save the source files in the correct “GeneratedSource” folder for the appropriate application.

Notice List

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



Notices display in rows. Use the filters above the notices to show or hide different types of notices, as follows:

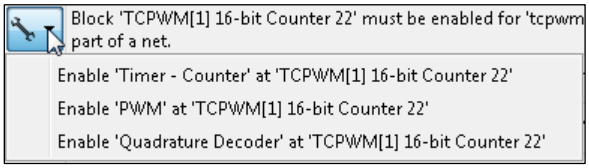
- **Errors** – These indicate there is at least one problem that must be addressed before you can build your application. Typical errors could include compiler build errors and connectivity errors.
- **Warnings** – These report unusual conditions that might indicate a problem, although you can usually build the application regardless.
- **Tasks** – These are actions you need to perform to resolve an issue, such as enabling a resource. If you save without resolving a task, it becomes an error.
- **Infos** – These are informational messages from the system to indicate something occurred.

The Notice List pane contains the following columns (each column header contains an arrow control to change the sorting of the notices in the table):

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

Fix a Task/Error

When a wrench  icon displays in the **Fix** column, click on it and select the appropriate action from the pull-down menu. When all related issues have been addressed, the notice will be removed from the Notice List pane.



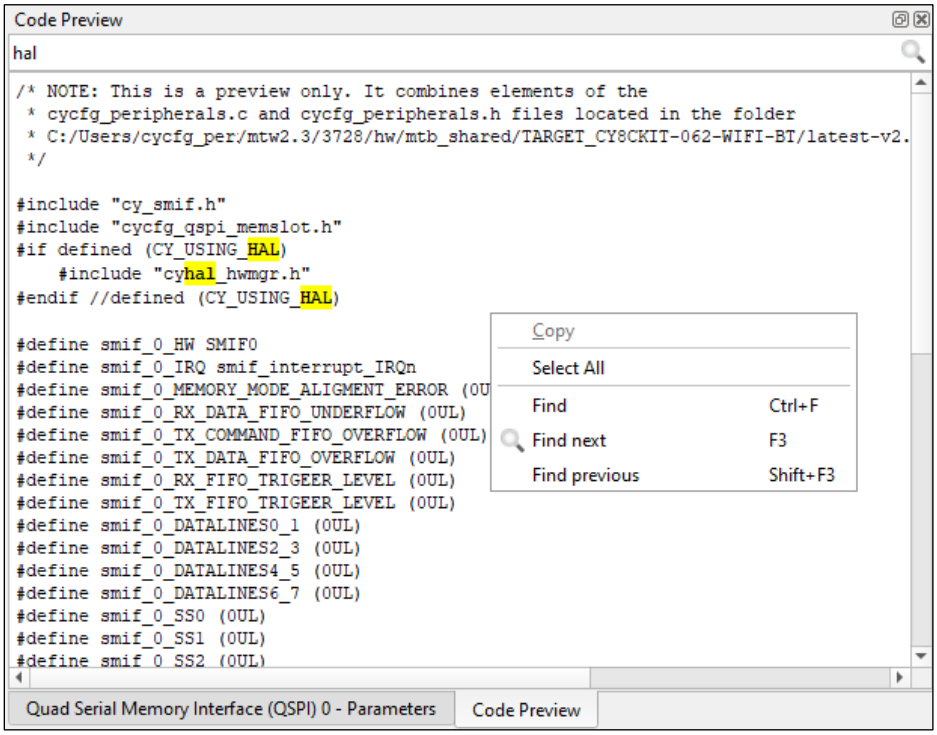
Note The fixes listed are not necessarily the only way to fix the issue. They are merely common options. Also, if you save the **.modus* file with outstanding tasks, they become errors saved in the *GeneratedSource/cycfg_notices.h* file.

Copy a Notice

You can copy a notice to your system’s clipboard using [**Ctrl**]+[**c**] or right-click and select **Copy.** Then, paste the notice text into an email, document, and so on.

Code Preview Pane

The **Code Preview** pane is a read-only preview of the code that will be generated for the currently selected resource when you save the **.modus* file. As you update configuration options, the Code Preview pane updates the code shown. This code will be written to the appropriate file(s) located in the *GeneratedSource* folder of your application.



You can select and copy code from this pane using [**Ctrl**]+[**C**] key or using the right-click menu option.

You can use the **Search** feature to find instances of specific terms.

References

Refer to the following documents for more information, as needed:

- Eclipse IDE for ModusToolbox User Guide
- ModusToolbox User Guide
- ModusToolbox Release Notes
- API Reference Guides
- Device Datasheets
- Device Technical Reference Manuals

Version Changes

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

Version	Change Descriptions	Notes
1.0	New tool.	
1.1	Added support for WICED Bluetooth devices.	
2.0	Changed the **Platform** tab to **System** tab.	This affects the file name generated during code generation. Older versions of the Device Configurator generated *cycfg_platform.(c/h)* files; it now generates *cycfg_system.(c/h)* files. If you are updating a design from a previous version, manually remove the old *cycfg_platform.(c/h)* files and update any references you created to use the new file names.
	Moved the Analog-Routing Editor to a tab.	
	Updated the **File** menu for library settings.	
	Added **Update All Personalities** menu item.	
	Added the ability to enter multiple resource Names using comma-separated list.	
2.1	Added Open System Explorer menu item. Added Change Devices menu item and dialog. Added Undo/Redo operations to the Edit menu. Added major/minor version number to the title bar.	
2.20	Added Copy feature to the Notice List. Added ability to see documentation links without enabling a resource. Added frequencies to clock diagram. Added feature to support incremental patch updates.	
2.21	Added Search feature to Code Preview. Implemented various performance improvements. Fixed an issue with multiple font sizes in the GUI. Fixed the PDL display name when using the MTB flow. Updated to allow analog routing from the SAR to any analog resource or pin.	
3.0	Removed Change Devices dialog. Updated the tool to recognize obsolete devices. Added path to files in the Code Preview Pane.	Use command line tools instead. Displays an error in the Notice List instead of blocking the file from being opened.