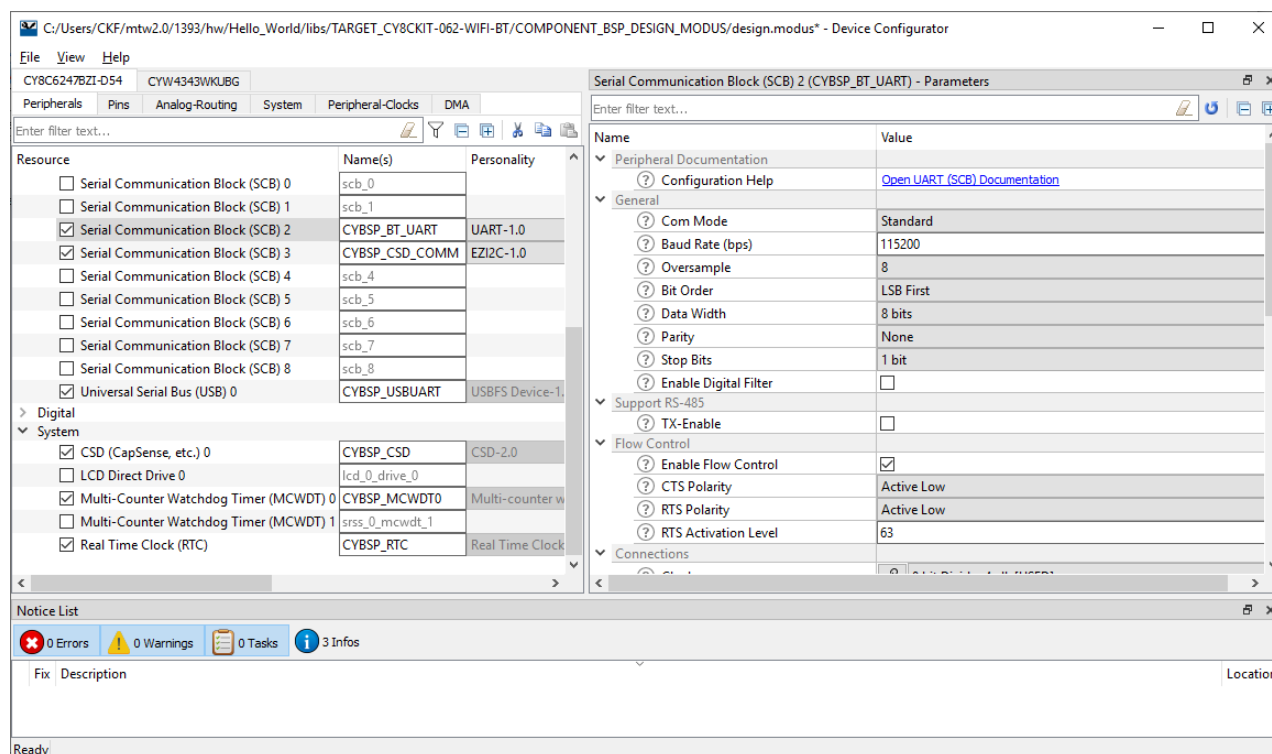


Overview

The Device Configurator is part of a collection of tools included in 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. Some complex peripherals, such as BLE, CapSense®, 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:

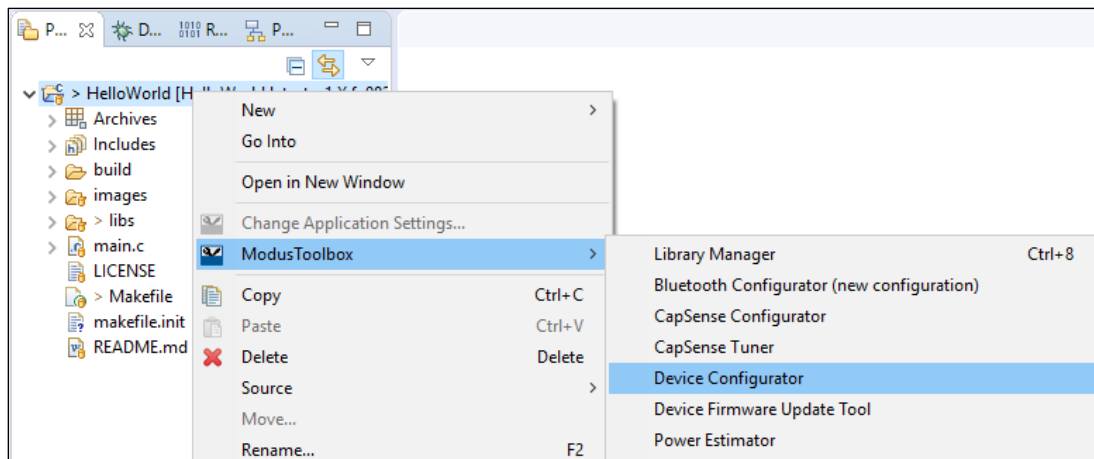
- **Resource** – Includes peripherals, pins, clocks, etc. used in an application.
- **Configurator** – A GUI-based tool used to configure a resource.
- **Application** – In ModusToolbox, an application consists of one or more projects, which are all related to each other.
- **Personality** – A file that defines a resource behavior.
- **Device Support Library** – A device support library provides critical firmware and device data files to configurators. Device support libraries are often identified with a file named *devicesupport.xml*. It is used to find things like other tools, devices, and personalities.

Launch the Device Configurator

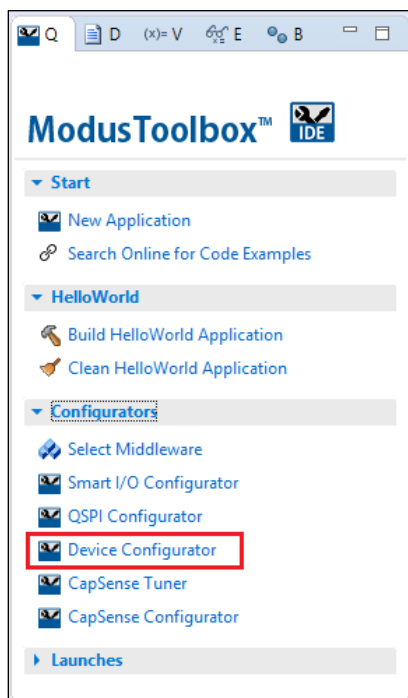
The Device Configurator is a stand-alone tool that contains [menus](#), [icons](#), [tabs](#), and several [panes](#) used to configure MCU peripherals. You launch it from, and use it with, a ModusToolbox IDE application. You can also run it independently of the ModusToolbox IDE. Then, you can either use the generated source with a ModusToolbox IDE application, or use it in any software environment you choose.

From a ModusToolbox IDE Application

To run the Device Configurator from an application within the ModusToolbox IDE, right-click on the project and select **ModusToolbox > Device Configurator**.



You can also open the Device Configurator by clicking the link in the ModusToolbox IDE 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” folder. ModusToolbox IDE applications use the *design.modus* file and generated source code in future application builds.

Independent of the ModusToolbox IDE

To run the Device Configurator independently, navigate to the install location and run the executable. On Windows, the default install location for the Device Configurator is:

```
<install_dir>\tools_<version>\device-configurator
```

For other operating systems, the installation directory will vary, based on how the software was installed.

When run independently, 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**-ModusToolbox IDE application, it will be your preferred working environment flow.
- If you open a design.modus file for a ModusToolbox IDE application, it will be the same flow as if you opened it [from within a ModusToolbox IDE application](#).

From the Command Line

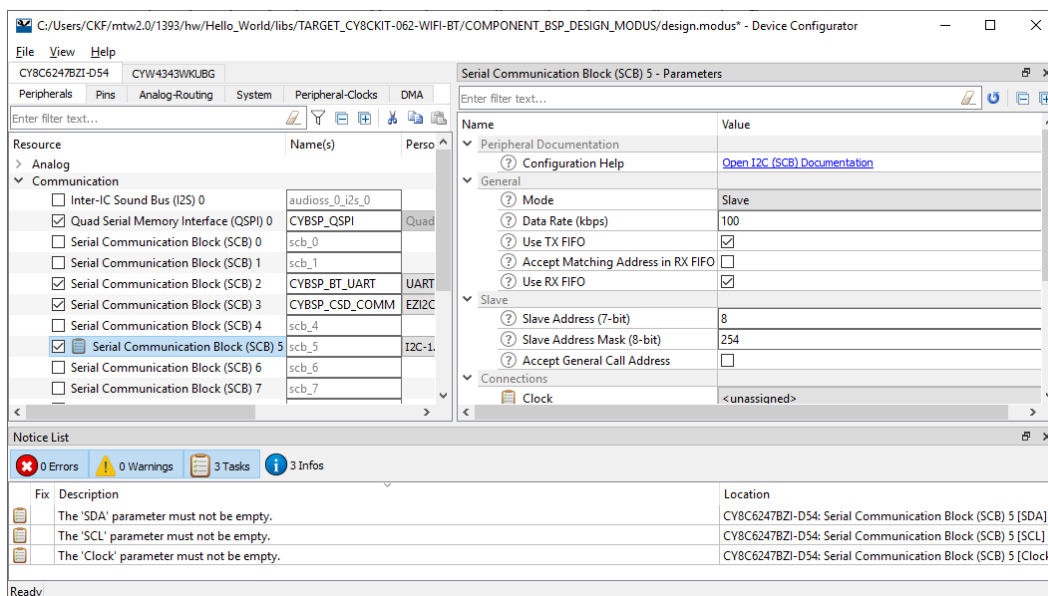
You can run the configurator from the command line. However, there are only a few reasons to do this in practice. The primary use case would be to re-generate source code based on the latest configuration settings. This would often be part of an overall build script for the entire application.

For information about command line options, run the configurator using the -h option.

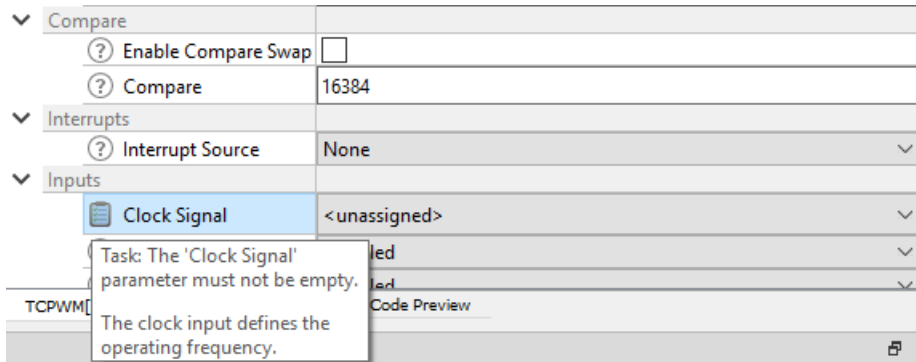
Quick Start

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

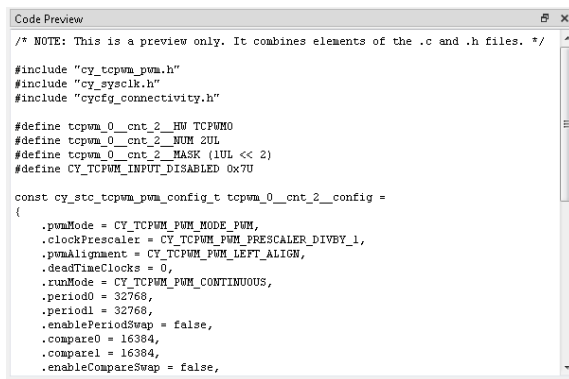
1. [Launch the Device Configurator](#).
2. Enable a desired peripheral on the [Peripherals Tab](#) by clicking the enable check box. Notice the [Parameters Pane](#) becomes populated with fields.



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



- Select an appropriate [parameter value](#) and the task should be removed from the Notice List.
- Select the [Code Preview Pane](#) to see a preview of the code that will be generated upon saving.



- Use the [various tabs](#) to enable and configure other resources as needed in the same manner as peripherals.
- Save the *.modus file to generate source code.

The Device Configurator generates code into a “GeneratedSource” directory in your ModusToolbox IDE application, or in the same location you saved the *.modus file for non-ModusToolbox 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.

- 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.

Menus

The menus include:

- **File** – Provides basic commands to open, close, and save files, as well as exit the tool.
 - **New** – Create a new `*.modus` file. See [Create Design dialog](#) for more information. The current file, if any, will be closed.
 - **Open** – Open an existing `*.modus` file. The current file, if any, will be closed.
 - **Close** – Close the current file. If there are pending changes, you will be prompted to save the file.
 - **Save** – Save the current file and generate code for the related ModusToolbox IDE application. If there are errors in the application, a dialog will indicate such. The file will still be saved.
 - **Save As** – Save the current file with a different name and/or location.
 - **Open Containing Folder** – Opens a folder on disk that contains the `design.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** – Close the tool. You will be prompted to save any pending changes.
- **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** – Provides access to this document and an About box.

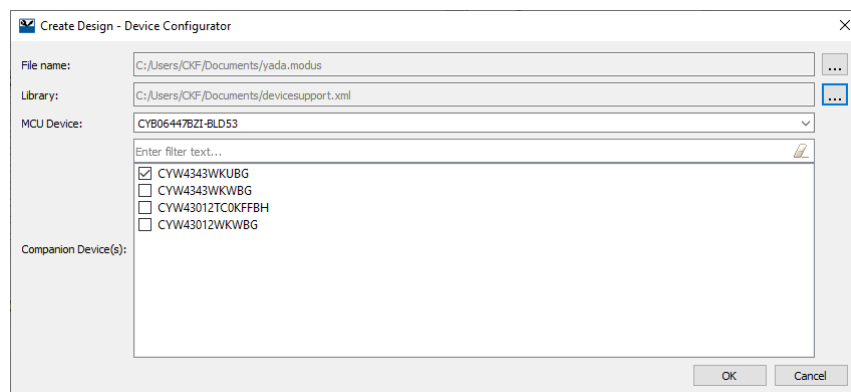
Icons

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. The Name(s) and Personality columns are read-only, and there is a tool-tip 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 [Independent of the ModusToolbox IDE](#). Open this dialog using the **File > New** option or pressing **[Ctrl] + [N]**.

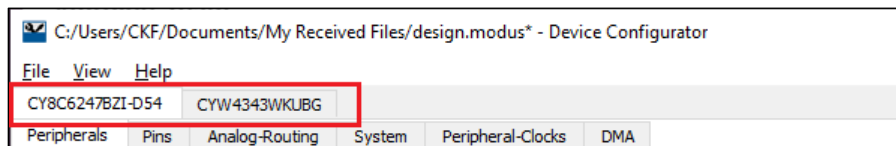


To create a new *.modus file:

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.

Devices

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](#). If there is only once device, then no **Device** tabs are shown.



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

Resources

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 have 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 I²C.
 - 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.

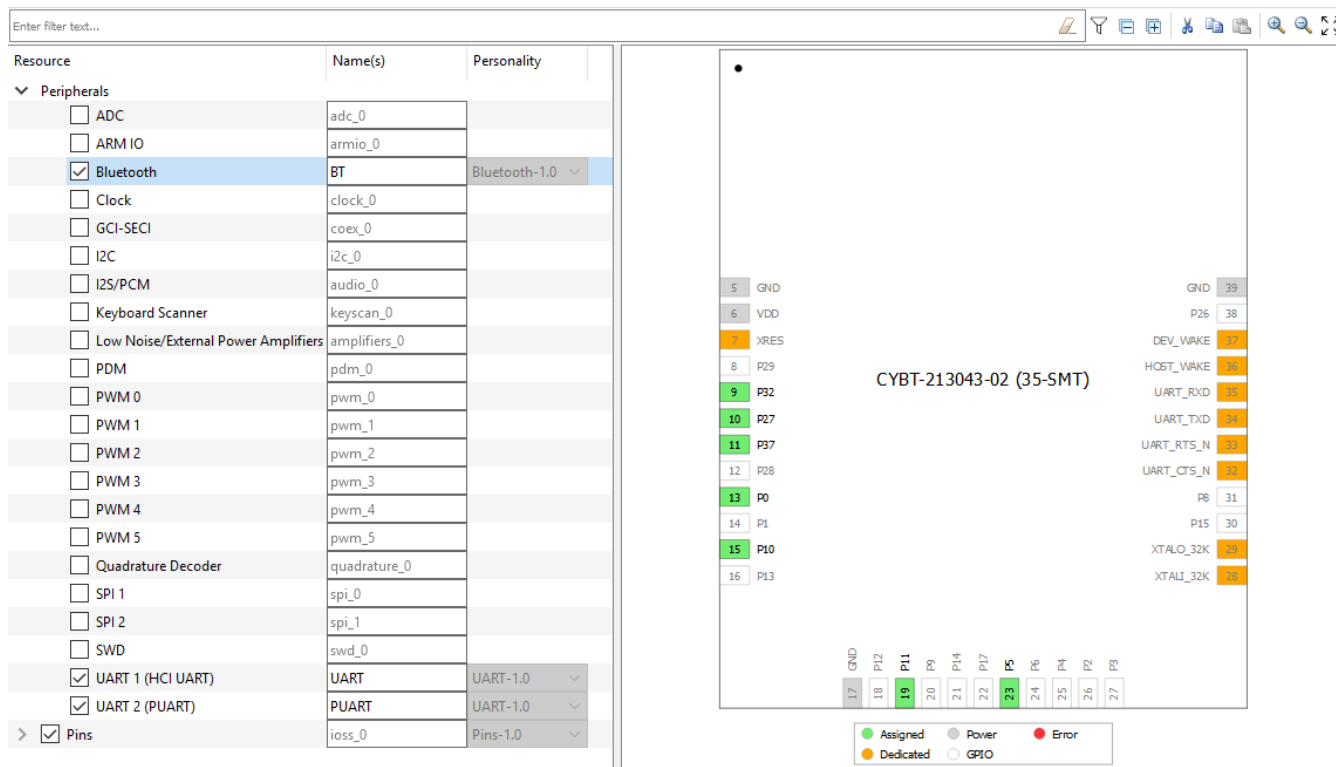
Peripherals

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

Peripherals		
Enter filter text...		
Resource	Name(s)	Personality
> Analog		
> Communication		
<input type="checkbox"/> Inter-IC Sound Bus (I2S) 0	audioss_0_i2s_0	
<input checked="" type="checkbox"/> Quad Serial Memory Interface (QSPI) 0	CYBSP_QSPI	Quad Serial Peripheral Interface (QSPI)-1.1
<input type="checkbox"/> Serial Communication Block (SCB) 0	scb_0	
<input type="checkbox"/> Serial Communication Block (SCB) 1	scb_1	
<input checked="" type="checkbox"/> Serial Communication Block (SCB) 2	CYBSP_BT_UART	UART-1.0
<input checked="" type="checkbox"/> Serial Communication Block (SCB) 3	CYBSP_CSD_COMM	EZ12C-1.0
<input type="checkbox"/> Serial Communication Block (SCB) 4	scb_4	
<input type="checkbox"/> Serial Communication Block (SCB) 5	scb_5	
<input type="checkbox"/> Serial Communication Block (SCB) 6	scb_6	
<input type="checkbox"/> Serial Communication Block (SCB) 7	scb_7	
<input type="checkbox"/> Serial Communication Block (SCB) 8	scb_8	
<input checked="" type="checkbox"/> Universal Serial Bus (USB) 0	CYBSP_USBUART	USBFS Device-1.1
> Digital		
> System		
<input checked="" type="checkbox"/> CSD (CapSense, etc.) 0	CYBSP_CSD	CSD-2.0
<input type="checkbox"/> LCD Direct Drive 0	lcd_0_drive_0	
<input checked="" type="checkbox"/> Multi-Counter Watchdog Timer (MCWDT) 0	CYBSP_MCWDT0	Multi-counter watchdog-1.0
<input type="checkbox"/> Multi-Counter Watchdog Timer (MCWDT) 1	srss_0_mcwdt_1	
<input checked="" type="checkbox"/> Real Time Clock (RTC)	CYBSP_RTC	Real Time Clock-1.1

Bluetooth Applications



Enter filter text...

Resource	Name(s)	Personality
▼ Peripherals		
<input type="checkbox"/> ADC	adc_0	
<input type="checkbox"/> ARM IO	armio_0	
<input checked="" type="checkbox"/> Bluetooth	BT	Bluetooth-1.0 ▼
<input type="checkbox"/> Clock	clock_0	
<input type="checkbox"/> GCI-SECI	coex_0	
<input type="checkbox"/> I2C	i2c_0	
<input type="checkbox"/> I2S/PCM	audio_0	
<input type="checkbox"/> Keyboard Scanner	keyscan_0	
<input type="checkbox"/> Low Noise/External Power Amplifiers	amplifiers_0	
<input type="checkbox"/> PDM	pdm_0	
<input type="checkbox"/> PWM 0	pwm_0	
<input type="checkbox"/> PWM 1	pwm_1	
<input type="checkbox"/> PWM 2	pwm_2	
<input type="checkbox"/> PWM 3	pwm_3	
<input type="checkbox"/> PWM 4	pwm_4	
<input type="checkbox"/> PWM 5	pwm_5	
<input type="checkbox"/> Quadrature Decoder	quadrature_0	
<input type="checkbox"/> SPI 1	spi_0	
<input type="checkbox"/> SPI 2	spi_1	
<input type="checkbox"/> SWD	swd_0	
<input checked="" type="checkbox"/> UART 1 (HCI UART)	UART	UART-1.0 ▼
<input checked="" type="checkbox"/> UART 2 (PUART)	PUART	UART-1.0 ▼
> <input checked="" type="checkbox"/> Pins	ioss_0	Pins-1.0 ▼

CYBT-213043-02 (35-SMT)

Legend:
 ● Assigned (Green)
 ● Dedicated (Orange)
 ● Power (Grey)
 ● Error (Red)
 ○ GPIO (White)

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:

- White – Disabled
- Green – Enabled
- Grey – Power/ground pins
- Orange – Fixed function pins
- Red – Error state

Peripherals

Pins

Analog-Routing

System

Peripheral-Clocks

DMA

Enter filter text...

Resource	Name(s)	Personality
▼ Port 0		
<input checked="" type="checkbox"/> P0[0]	CYBSP_WCO_IN	Pin-1.1
<input checked="" type="checkbox"/> P0[1]	CYBSP_WCO_OUT	Pin-1.1
<input type="checkbox"/> P0[2]	ioss_0_port_0_pin_2	
<input type="checkbox"/> P0[3]	ioss_0_port_0_pin_3	
<input type="checkbox"/> P0[4]	ioss_0_port_0_pin_4	
<input type="checkbox"/> P0[5]	ioss_0_port_0_pin_5	
▼ Port 1		
<input checked="" type="checkbox"/> P1[0]	CYBSP_CSD_TX	Pin-1.1
<input type="checkbox"/> P1[1]	ioss_0_port_1_pin_1	
<input type="checkbox"/> P1[2]	ioss_0_port_1_pin_2	
<input type="checkbox"/> P1[3]	ioss_0_port_1_pin_3	
<input type="checkbox"/> P1[4]	ioss_0_port_1_pin_4	
<input type="checkbox"/> P1[5]	ioss_0_port_1_pin_5	
▼ Port 2		
<input type="checkbox"/> P2[0]	ioss_0_port_2_pin_0	
<input type="checkbox"/> P2[1]	ioss_0_port_2_pin_1	
<input type="checkbox"/> P2[2]	ioss_0_port_2_pin_2	
<input type="checkbox"/> P2[3]	ioss_0_port_2_pin_3	
<input type="checkbox"/> P2[4]	ioss_0_port_2_pin_4	
<input type="checkbox"/> P2[5]	ioss_0_port_2_pin_5	
<input type="checkbox"/> P2[6]	ioss_0_port_2_pin_6	
<input checked="" type="checkbox"/> P2[7]	CYBSP_WIFI_HOST_WAKE	Pin-1.1
▼ Port 3		

13 12 11 10 9 8 7 6 5 4 3 2 1

CY8C6247BZI-D54 (124-BGA)

● Assigned

● Power

● Error

● Dedicated

● GPIO

Resource

> Peripherals

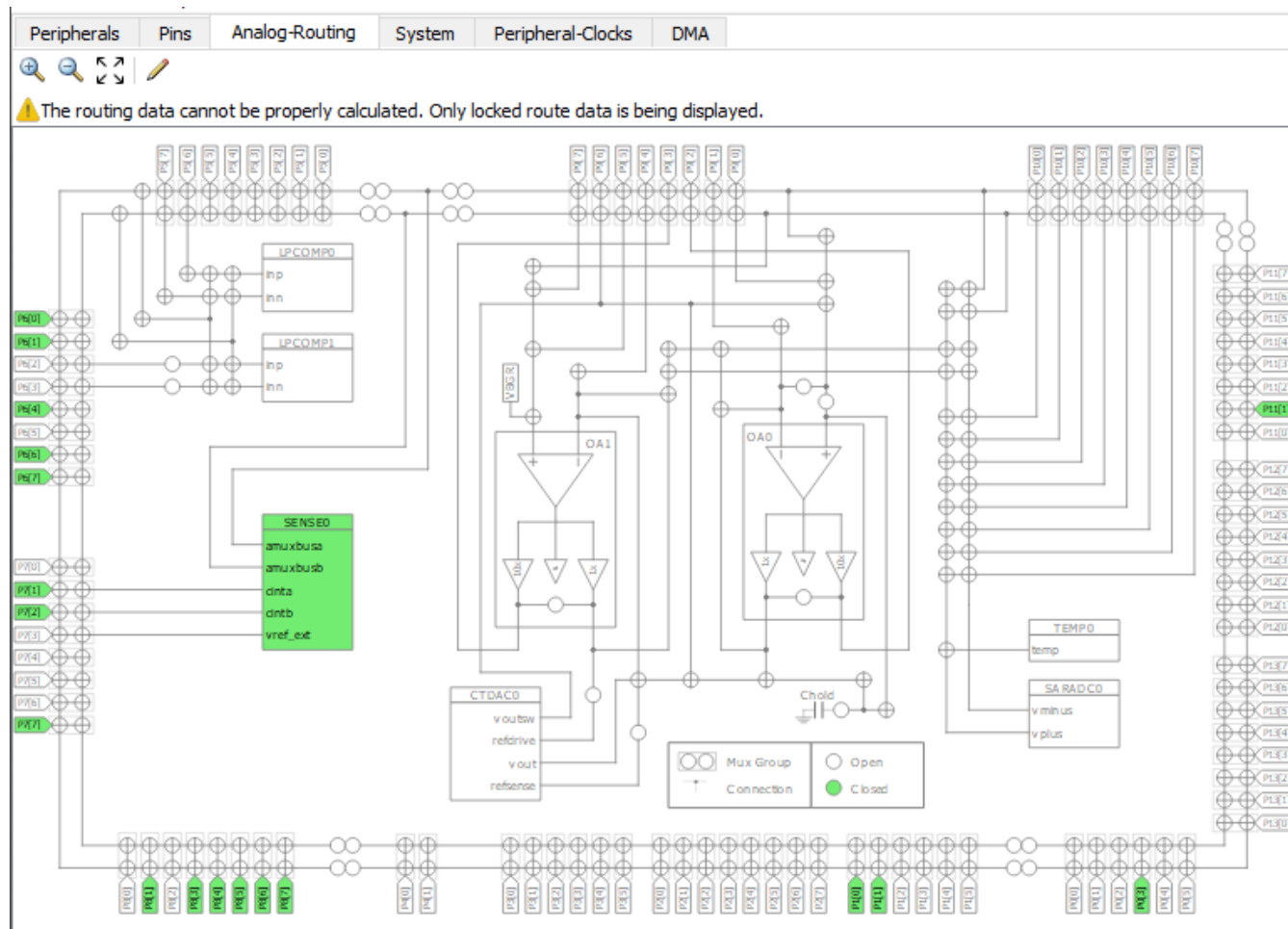
☒ Pins


	Name(s)	Personality
<input checked="" type="checkbox"/> P0	BTN_USER	Pin-1.0
<input type="checkbox"/> P1	ioss_0_pin_1	
<input type="checkbox"/> P2	ioss_0_pin_2	
<input type="checkbox"/> P3	ioss_0_pin_3	
<input type="checkbox"/> P4	ioss_0_pin_4	
<input checked="" type="checkbox"/> P5	LED1	Pin-1.0
<input type="checkbox"/> P6	ioss_0_pin_6	
<input type="checkbox"/> P8	ioss_0_pin_8	
<input type="checkbox"/> P9	ioss_0_pin_9	
<input checked="" type="checkbox"/> P10	PUART_CTS	Pin-1.0
<input checked="" type="checkbox"/> P11	PUART_RTS	Pin-1.0
<input type="checkbox"/> P12	ioss_0_pin_12	
<input type="checkbox"/> P13	ioss_0_pin_13	
<input type="checkbox"/> P14	ioss_0_pin_14	
<input type="checkbox"/> P15	ioss_0_pin_15	
<input type="checkbox"/> P17	ioss_0_pin_17	
<input type="checkbox"/> P26	ioss_0_pin_26	
<input checked="" type="checkbox"/> P27	LED2	Pin-1.0
<input type="checkbox"/> P28	ioss_0_pin_28	
<input type="checkbox"/> P29	ioss_0_pin_29	
<input checked="" type="checkbox"/> P32	PUART_TX	Pin-1.0
<input checked="" type="checkbox"/> P37	PUART_RX	Pin-1.0

CYBT-213043-02 (35-SMT)

Analog-Routing Tab

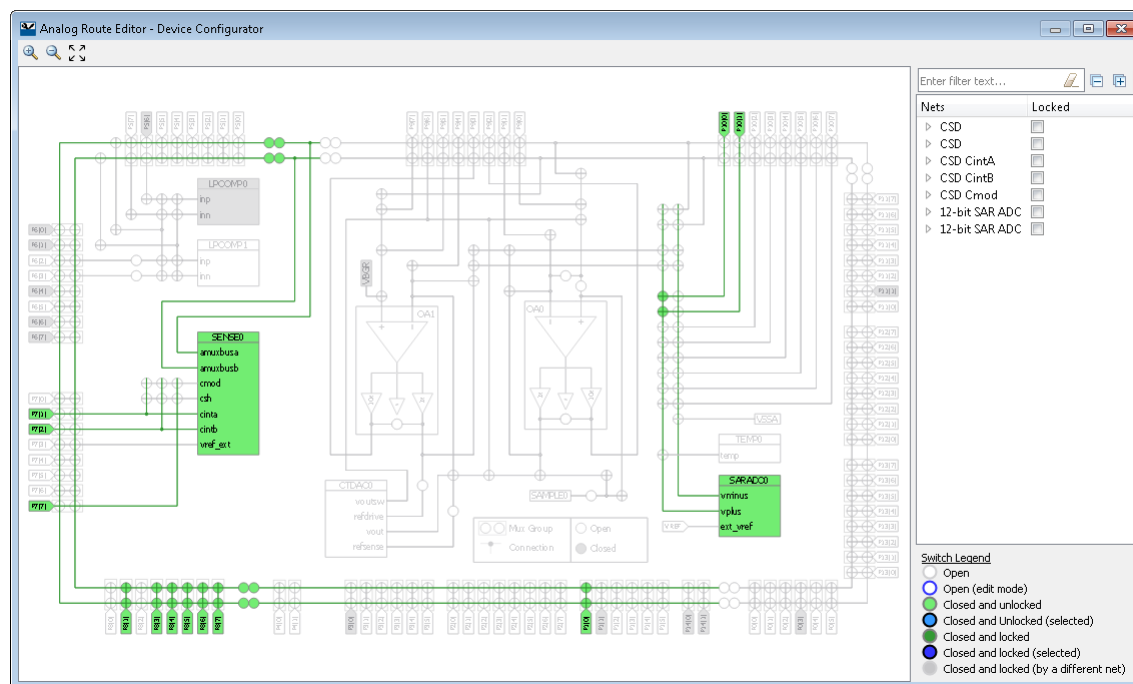
The **Analog-Routing** tab shows the various analog resources in your application. Enabled resources are green.



There are zoom commands to resize the diagram as needed. The **Edit**  command opens the [Analog Route Editor](#).

Analog Route Editor

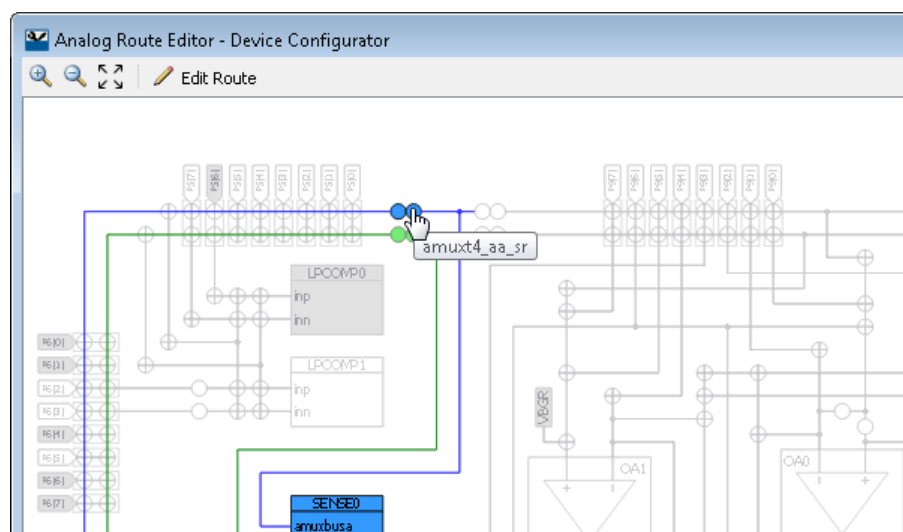
The Analog Route Editor allows you to manually edit the routing of analog resources in your application. It also provides the ability to lock-down all or some of the results.



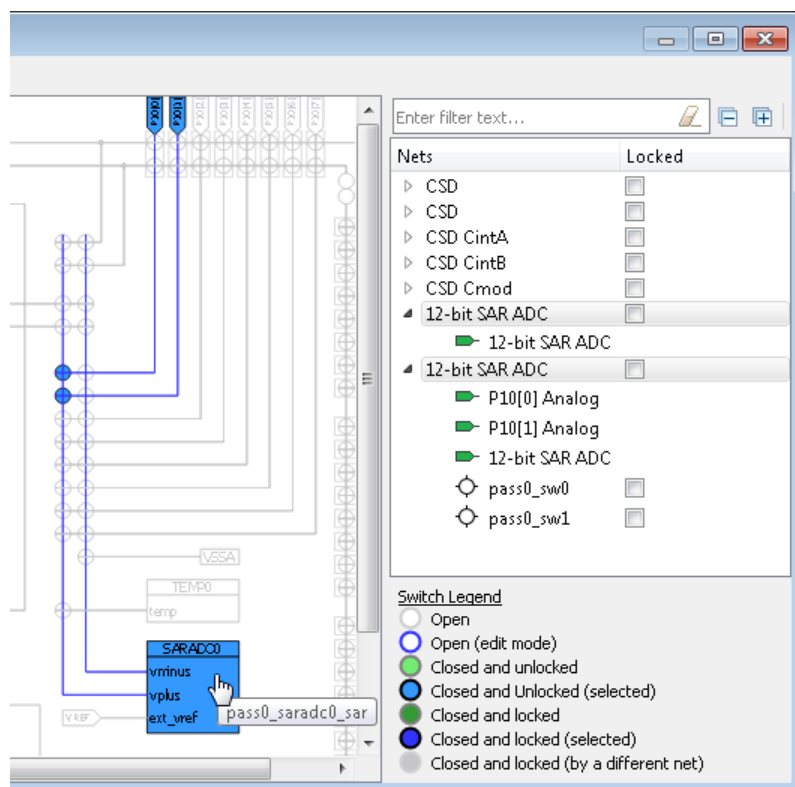
Note If there are configuration errors, complete routing results will not be available; only locked resources. If you open the Analog Editor in this error state, a warning message will display. You can still lock and unlock switches, but you won't get complete routing results as long as the configuration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

Select a Resource

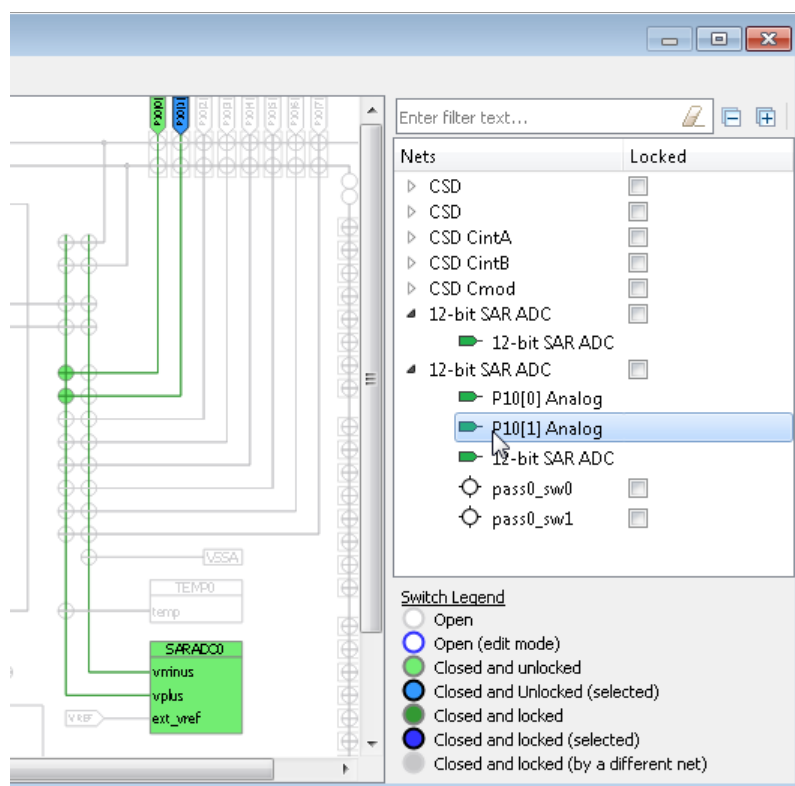
To select an analog resource, click on it. Any enabled (green) element in the tree can be selected. The resource and the associated route(s) become blue. Also, the **Edit Route** command appears on the toolbar. See [Edit Route](#).



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.

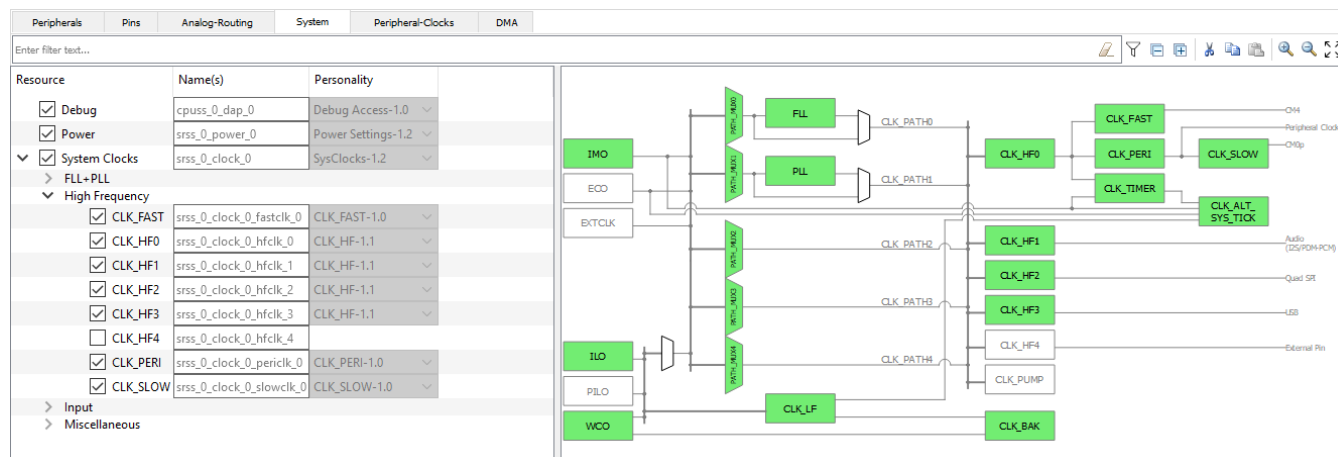
If a change results in an error, a message displays. The routes are automatically rolled back to the previous state, so you will lose at most the last invalid change.

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.

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](#).

Peripherals Pins Analog-Routing System Peripheral-Clocks DMA			
Enter filter text...			
Resource	Name(s)	Personality	
▼ 8 bit			
<input checked="" type="checkbox"/> 8 bit Divider 0	CYBSP_SDIO_DIV	Peripheral Clock-1.0 ▼	
<input checked="" type="checkbox"/> 8 bit Divider 1	CYBSP_CSD_COMM_CLK_DIV	Peripheral Clock-1.0 ▼	
<input type="checkbox"/> 8 bit Divider 2	peri_0_div_8_2		
<input checked="" type="checkbox"/> 8 bit Divider 3	CYBSP_CSD_CLK_DIV	Peripheral Clock-1.0 ▼	
<input checked="" type="checkbox"/> 8 bit Divider 4	peri_0_div_8_4	Peripheral Clock-1.0 ▼	
<input type="checkbox"/> 8 bit Divider 5	peri_0_div_8_5		
<input type="checkbox"/> 8 bit Divider 6	peri_0_div_8_6		
<input type="checkbox"/> 8 bit Divider 7	peri_0_div_8_7		
> 16 bit			
▼ 16.5 bit			
<input type="checkbox"/> 16.5 bit Divider 0	peri_0_div_16_5_0		
<input type="checkbox"/> 16.5 bit Divider 1	peri_0_div_16_5_1		
<input type="checkbox"/> 16.5 bit Divider 2	peri_0_div_16_5_2		
<input type="checkbox"/> 16.5 bit Divider 3	peri_0_div_16_5_3		
> 24.5 bit			

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](#).

Peripherals	Pins	Analog-Routing	System	Peripheral-Clocks	DMA
Enter filter text...					
Resource	Name(s)	Personality			
▼ DMA DataWire 0					
<input checked="" type="checkbox"/> DMA DataWire 0: Channel 0	cpuss_0_dw0_0_chan_0	DMA-1.0 ▾			
<input checked="" type="checkbox"/> DMA DataWire 0: Channel 1	cpuss_0_dw0_0_chan_1	DMA-1.0 ▾			
<input type="checkbox"/> DMA DataWire 0: Channel 2	cpuss_0_dw0_0_chan_2				
<input type="checkbox"/> DMA DataWire 0: Channel 3	cpuss_0_dw0_0_chan_3				
<input type="checkbox"/> DMA DataWire 0: Channel 4	cpuss_0_dw0_0_chan_4				
<input type="checkbox"/> DMA DataWire 0: Channel 5	cpuss_0_dw0_0_chan_5				
<input type="checkbox"/> DMA DataWire 0: Channel 6	cpuss_0_dw0_0_chan_6				
<input type="checkbox"/> DMA DataWire 0: Channel 7	cpuss_0_dw0_0_chan_7				
<input type="checkbox"/> DMA DataWire 0: Channel 8	cpuss_0_dw0_0_chan_8				
<input type="checkbox"/> DMA DataWire 0: Channel 9	cpuss_0_dw0_0_chan_9				
<input type="checkbox"/> DMA DataWire 0: Channel 10	cpuss_0_dw0_0_chan_10				
<input type="checkbox"/> DMA DataWire 0: Channel 11	cpuss_0_dw0_0_chan_11				
<input type="checkbox"/> DMA DataWire 0: Channel 12	cpuss_0_dw0_0_chan_12				
<input type="checkbox"/> DMA DataWire 0: Channel 13	cpuss_0_dw0_0_chan_13				
<input type="checkbox"/> DMA DataWire 0: Channel 14	cpuss_0_dw0_0_chan_14				
<input type="checkbox"/> DMA DataWire 0: Channel 15	cpuss_0_dw0_0_chan_15				
> DMA DataWire 1					

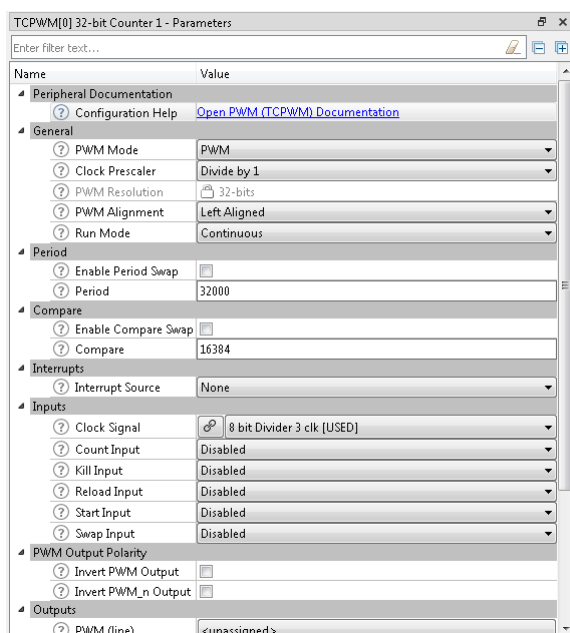
Panes

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.

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.

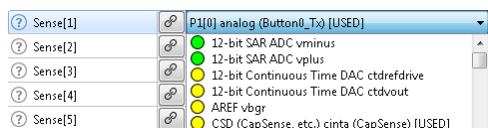
Note After selecting these parameter types, use the **Go To**  button to jump to the selected resource.

- **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.



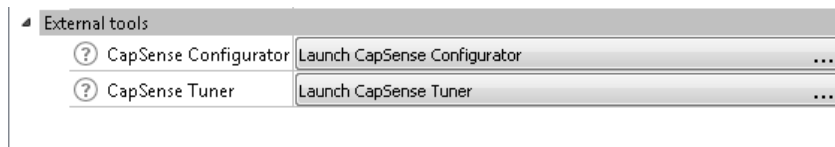
The signals have guidance icons next to them to indicate the status of the signal, as follows:

- 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 (BLE, CapSense, etc.), the Device Configurator provides links to launch those separate configurators. After enabling the peripheral on the [Peripherals Tab](#), the Parameters pane will contain a **<Configurator>** parameter, where <Configurator> is the name of the other configurator.

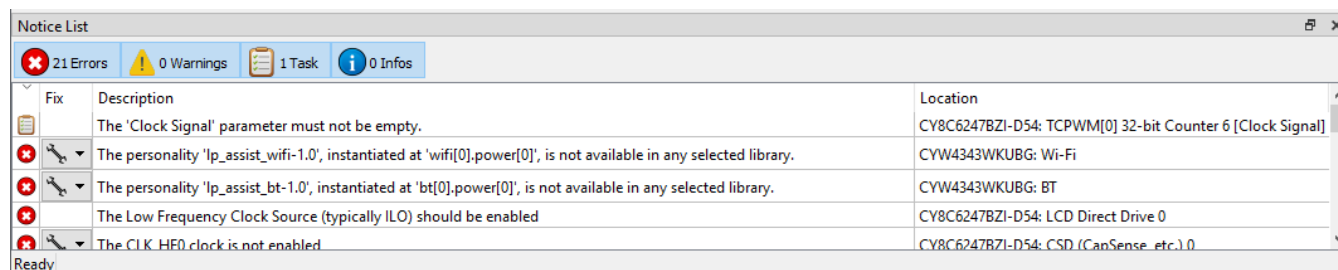


Click on **[Launch < Configurator > . . .]** to launch the configurator.

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 Pane

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:

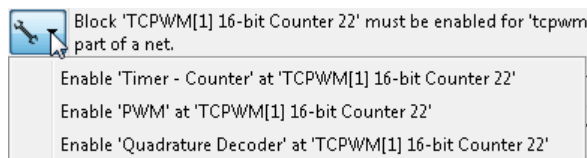
- **Errors** – These indicate there is at least one problem that must be addressed before you can build your application. Typical errors could include: compiler build errors, and connectivity errors.
- **Warnings** – These report unusual conditions that might indicate a problem, although you can usually 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

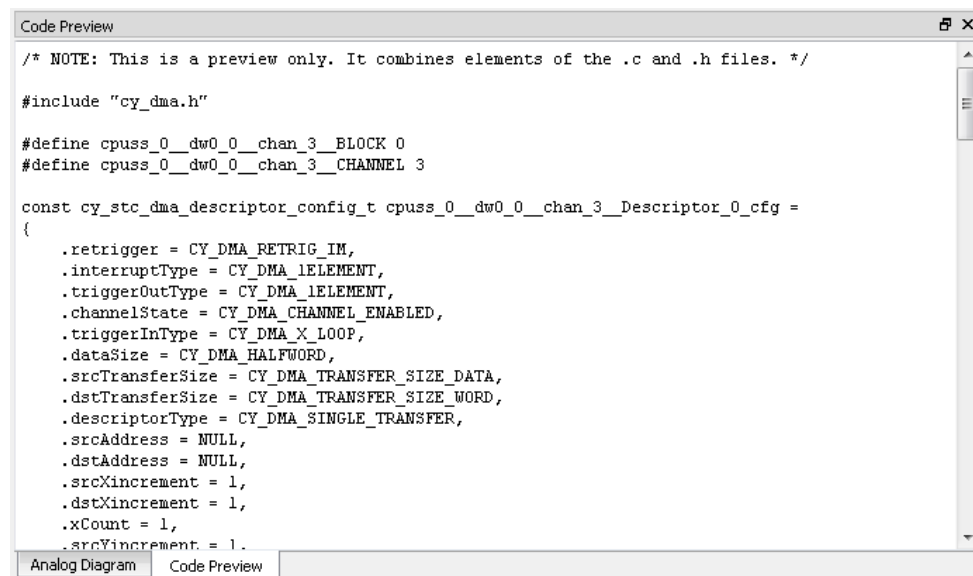
When a wrench icon displays in the **Fix** column, click on it and select the appropriate action from the pull-down menu. When all related issues have been addressed, the notice will be removed from the Notice List pane.



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.

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.



```
Code Preview
/* NOTE: This is a preview only. It combines elements of the .c and .h files. */

#include "cy_dma.h"

#define cpuss_0_dw0_0_chan_3_BLOCK 0
#define cpuss_0_dw0_0_chan_3_CHANNEL 3

const cy_stc_dma_descriptor_config_t cpuss_0_dw0_0_chan_3_Descriptor_0_cfg =
{
    .retrigger = CY_DMA_RETRIG_IM,
    .interruptType = CY_DMA_INTERRUPT,
    .triggerOutType = CY_DMA_INTERRUPT,
    .channelState = CY_DMA_CHANNEL_ENABLED,
    .triggerInType = CY_DMA_X_LOOP,
    .dataSize = CY_DMA_HALFWORD,
    .srcTransferSize = CY_DMA_TRANSFER_SIZE_DATA,
    .dstTransferSize = CY_DMA_TRANSFER_SIZE_WORD,
    .descriptorType = CY_DMA_SINGLE_TRANSFER,
    .srcAddress = NULL,
    .dstAddress = NULL,
    .srcXincrement = 1,
    .dstXincrement = 1,
    .xCount = 1,
    .srcYincrement = 1,
}
```

Analog Diagram Code Preview

References

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

- ModusToolbox IDE 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 <i>cycfg_platform.(c/h)</i> files; it now generates <i>cycfg_system.(c/h)</i> files. If you are updating a design from a previous version, manually remove the old <i>cycfg_platform.(c/h)</i> 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.	

Version	Change Descriptions	Notes
	Added Update All Personalities menu item.	
	Added the ability to enter multiple resource Names using comma-separated list.	
	Added Open Containing Folder menu item.	

© Cypress Semiconductor Corporation, 2018-2019. This document is the property of Cypress Semiconductor Corporation and its subsidiaries, including Spansion LLC ("Cypress"). This document, including any software or firmware included or referenced in this document ("Software"), is owned by Cypress under the intellectual property laws and treaties of the United States and other countries worldwide. Cypress reserves all rights under such laws and treaties and does not, except as specifically stated in this paragraph, grant any license under its patents, copyrights, trademarks, or other intellectual property rights. If the Software is not accompanied by a license agreement and you do not otherwise have a written agreement with Cypress governing the use of the Software, then Cypress hereby grants you a personal, non-exclusive, nontransferable license (without the right to sublicense) (1) under its copyright rights in the Software (a) for Software provided in source code form, to modify and reproduce the Software solely for use with Cypress hardware products, only internally within your organization, and (b) to distribute the Software in binary code form externally to end users (either directly or indirectly through resellers and distributors), solely for use on Cypress hardware product units, and (2) under those claims of Cypress's patents that are infringed by the Software (as provided by Cypress, unmodified) to make, use, distribute, and import the Software solely for use with Cypress hardware products. Any other use, reproduction, modification, translation, or compilation of the Software is prohibited.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS DOCUMENT OR ANY SOFTWARE OR ACCOMPANYING HARDWARE, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. To the extent permitted by applicable law, Cypress reserves the right to make changes to this document without further notice. Cypress does not assume any liability arising out of the application or use of any product or circuit described in this document. Any information provided in this document, including any sample design information or programming code, is provided only for reference purposes. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. Cypress products are not designed, intended, or authorized for use as critical components in systems designed or intended for the operation of weapons, weapons systems, nuclear installations, life-support devices or systems, other medical devices or systems (including resuscitation equipment and surgical implants), pollution control or hazardous substances management, or other uses where the failure of the device or system could cause personal injury, death, or property damage ("Unintended Uses"). A critical component is any component of a device or system whose failure to perform can be reasonably expected to cause the failure of the device or system, or to affect its safety or effectiveness. Cypress is not liable, in whole or in part, and you shall and hereby do release Cypress from any claim, damage, or other liability arising from or related to all Unintended Uses of Cypress products. You shall indemnify and hold Cypress harmless from and against all claims, costs, damages, and other liabilities, including claims for personal injury or death, arising from or related to any Unintended Uses of Cypress products.

Cypress, the Cypress logo, Spansion, the Spansion logo, and combinations thereof, ModusToolbox, WICED, PSoC, CapSense, EZ-USB, F-RAM, and Traveo are trademarks or registered trademarks of Cypress in the United States and other countries. For a more complete list of Cypress trademarks, visit cypress.com. Other names and brands may be claimed as property of their respective owners.