

# ModusToolbox™ Device Configurator user guide

ModusToolbox<sup>™</sup> tools package version 3.0.0

Device Configurator version 4.0.0

# About this document

#### Scope and purpose

This guide provides information and instructions for using the Device Configurator to update various parameters for you ModusToolbox<sup>™</sup> application.

#### Intended audience

This document helps application developers understand how to use the Device Configurator as part of creating a ModusToolbox<sup>™</sup> application.

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

#### **Document conventions**

#### **Abbreviations and definitions**

- **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 *props.json*. It is used to find things like other tools, devices, and personalities.

#### **Reference documents**

Refer to the following documents for more information as needed:

- Eclipse IDE for ModusToolbox<sup>™</sup> user guide
- <u>ModusToolbox<sup>™</sup> tools package user guide</u>
- <u>ModusToolbox™ tools package release notes</u>
- API reference guides



# Table of contents

# **Table of contents**

1	Overview
2	Launch the Device Configurator
2.1	make command4
2.2	Eclipse IDE4
2.3	Executable (GUI)4
2.4	Open configuration file5
2.5	Create new configuration5
2.6	Executable (CLI)6
3	Quick start7
4	Code generation9
5	GUI description10
5.1	Menus
5.2	Device tabs11
5.3	Resource tabs
5.4	Panes
5.5	lcons11
6	Resources tabs12
6.1	Peripherals13
6.2	Pins14
6.3	Analog-Routing tab15
6.4	System tab18
6.5	Peripheral-Clocks tab19
6.6	DMA tab19
7	Panes
7.1	Parameters pane
7.2	Notice List
7.3	Code Preview pane23
8	Version changes



#### Overview

# 1 Overview

The Device Configurator is part of a collection of tools included with the ModusToolbox<sup>™</sup> 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 CAPSENSE<sup>™</sup>, SegLCD, etc., have specialized configuration tools, and the Device Configurator provides links to launch those separate tools (see <u>Launch other configurators</u>). After configuring and saving a particular device's settings, the Device Configurator generates firmware for use in your application (see <u>Code generation</u>).

C:/Users/follettcj/mtw3.0/8297/dev-config/Dual-CPU_Empty_PSoC6_App/bsps/TARGET_APP_	CY8CKIT-062S2-43012/config/design.modus - Device Configurator 4.0 -	ı ×
<u>File E</u> dit <u>V</u> iew <u>H</u> elp		
🗋 🚰 🔚 🖄 (M		
CY8C624ABZI-S2D44 CYW43012C0WKWBG	CSD (CapSense, etc.) 0 (CYBSP_CSD) - Parameters	ð ×
Peripherals Pins Analog-Routing System Peripheral-Clocks DMA	Enter filter text	<b>5</b> 🖻 🕀
Enter filter text 🖉 🖻 🖽 🐇 🗎 🕅	Name Value	^
Resource Name(s) Personality	V Overview	
> Analog	Configuration Help <u>Open CSD Documentation</u>	
> Communication	⑦ CAPSENSE™ Middleware <u>Open CAPSENSE™ Documentation</u>	
> Digital	CSDADC Middleware Open CSDADC Documentation	
✓ CSD (CapSense, etc.) 0 CYBSP_CSD CSD-3.0 ✓	CSDIDAC Middleware Open CSDIDAC Documentation	
LCD Direct Drive 0	✓ Inputs	
Multi-Counter Watchdog Timer (MCWDT) 0 srss_0_mcwdt_0	Clock Clock Ø lbit Divider 0 clk (CYBSP_CSD_CLK_DIV, CYBSP_CS_CLK_DIV) [USE	D]
Multi-Counter Watchdog Timer (MCWDT) 1 srss_0_ncwdt_1	✓ CAPSENSE™	
Real Time Clock (RTC)	⑦ Enable CAPSENSE™	
	⑦ Target CPU core Cortex M4	
	Y External Tools	
	⑦ CAPSENSE™ Configurator Launch CAPSENSE™ Configurator	
	⑦ CAPSENSE™ Tuner Launch CAPSENSE™ Tuner	
	✓ CSDADC	
	② Enable CSDADC	
	✓ CSDIDAC	~
		>
	CSD (CapSense, etc.) 0 (CYBSP_CSD) - Parameters Code Preview	
Notice List		ēΧ
😮 0 Errors 🔥 0 Warnings 🧮 0 Tasks 🕕 3 Infos		
Fix Description		Location
Ready		
		-



Launch the Device Configurator

# 2 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 *design.modus* file contains all the required hardware configuration information about the device for the application. When you save updates, the Device Configurator generates/updates source code in the *GeneratedSource* directory next to the *design.modus* file. Applications use the *design.modus* file and generated source code in future application builds.

## 2.1 make command

As described in the <u>ModusToolbox<sup>™</sup> tools package user guide</u> build system chapter, you can run numerous make commands in the application directory, such as launching the Device Configurator. After you have created a ModusToolbox<sup>™</sup> application, navigate to the application directory and type the following command in the appropriate bash terminal window:

make device-configurator

This opens the Device Configurator using the application's *design.modus* file.

# 2.2 Eclipse IDE

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

🔓 P 🙁 🎄 D	1010 0101	R 🖫 P 🗆 🗆			
> 👺 Hello_Worl > 👺 mtb_shared		New Go Into	>		
		Open in New Window Show In	Alt+Shift+W >		
		ModusToolbox™ Show in Local Terminal	>	Library Manager 2.0 Ctrl+ Bluetooth Configurator 2.60 (new configuration)	8
		<b>Copy</b> Paste	Ctrl+C Ctrl+V	CapSense Configurator 5.0 CapSense Tuner 5.0	
	×	Delete Source	Delete >	Device Configurator 4.0 Device Firmware Update Host Tool 1.50	
		Move Rename	F2	ML Configurator 1.30 QSPI Configurator 4.0	
		Import Export		Smart I/O Configurator 4.0 USB Configurator 2.50 (new configuration)	

Quick Panel (x)= Variables 🙀 Expressions 💁 Breakpoints
Eclipse IDE for
ModusToolbox™
Start
Hello_World (CY8CPROTO-062-4343W)
Launches
🔛 Library Manager 2.0
Bluetooth Configurator 2.60 (new configuration)
CapSense Configurator 5.0
CapSense Tuner 5.0
Device Configurator 4.0
Device Firmware Update Host Tool 1.50
ML Configurator 1.30

Similar to the make command method, launching the Device Configurator using the Eclipse IDE opens the tool using the application's *design.modus* file. Refer to the Eclipse IDE for ModusToolbox<sup>™</sup> user guide for details about the IDE.

# 2.3 Executable (GUI)

If you don't have an application or if you just want to see what the configurator looks like, you can launch the Device Configurator GUI by running its executable as appropriate for your operating system (for example, double-click it or select it using the Windows **Start** menu). By default, it is installed here:

#### <install\_dir>/ModusToolbox/tools\_<version>/device-configurator-<version>

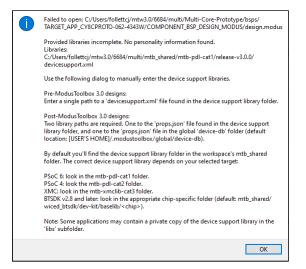
When launched this way, the Device Configurator opens without any configuration information. You can either open a specific \*.*modus* file or create a new one.



Launch the Device Configurator

# 2.4 Open configuration file

Click **File > Open** and navigate to the location of the \*.*modus* file to open. The tool may open the following dialog, indicating that it cannot find the device support library (*props.json*) file:



The dialog provides pointers for where you can find the appropriate file for your device. Click **OK** to close the dialog, and then navigate to the appropriate directory containing the *props.json* file.

# 2.5 Create new configuration

Click **File > New** to open the Create Design dialog, which provides fields to create new \*.*modus* files.

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 **Libraries**, navigate to the location of the device-db *props.json* file, and select it. Also, find and select the appropriate PDL *props.json* file.
- 3. Select a **Device** from the pull-down menu.
- 4. If applicable, select one or more Additional devices from the list.
- 5. Click **OK** to close the dialog and load the new \*.*modus* file into the Device Configurator.

🕍 Create Design - I	Device Configurator 4.0		>
File name:	C:/Users/follettcj/example-bsps/TARGET_MyBSP-5-2-22/COMPONENT_BSP_DESIGN_MODUS/design.modus		
Libraries:	C:/Users/follettcj/.modustoolbox/global/device-db/release-v4.0.0/props.json C:/Users/follettcj/mtw3.0/7156/multi/mtb_shared/mtb-pdl-cat1/release-v3.0.0/props.json		
Device:	CYB06447BZI-BLD53	~	
	Enter filter text		
Additional devices:		•	
Documentation:	ITE PSoC 4 Product Selector PSoC 6 Product Selector XMC1000 Product Selector XMC4000 Product Selector		
	ОК	Can	cel



Launch the Device Configurator

# 2.6 Executable (CLI)

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



### Quick start

# 3 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 <u>Peripherals tab</u> by clicking the enable check box. Notice the <u>Parameters</u> <u>pane</u> becomes populated with fields.

C:/Users/follettcj/mtw3.0/8297/dev-config/Dual-C	PU_Empty_PSoC6_App/bsps/TARGET_APP_CY	/8CKIT-062S2-43012/config/design.modus* - Devi	ice Configurator 4.0 —		×
<u>F</u> ile <u>E</u> dit <u>V</u> iew <u>H</u> elp					
🗋 🖆 🔚 🔊 (*					
CY8C624ABZI-S2D44 CYW43012C0WKWBG		12-bit SAR ADC - Parameters		ć	8 ×
Peripherals Pins Analog-Routing System	Peripheral-Clocks DMA	Enter filter text	4	🖉 😈 🗏 🖻	
Enter filter text	🖉 🔻 🖻 🖷 🔏 🗎 🛍	Name	Value		^
Resource	Name(s) Personality	Overview     Overview     Overview     Overview	Open SAR Documentation		- 1
✓ Analog		<ul> <li>Configuration Help</li> <li>General</li> </ul>	Open SAK Documentation		
Low-Power Comparator 0	lpcomp_0_comp_0	? Vref Select	Internal Reference (from AREF Resource)		
Low-Power Comparator 1	lpcomp_0_comp_1	<ul> <li>Viel Select</li> <li>Vref Voltage (V)</li> </ul>			-
✓ Programmable Analog 0	pass 0 sar 0 SAR-6.0 V	Number of Channels	2		-
I2-bit SAR ADC     AREF		<ul> <li>Injection Channel</li> </ul>			-
Communication	pass_0_aref_0	<u> </u>			
> Digital		<ul> <li>(?) Target Scan Rate (sps)</li> </ul>	2000		-
✓ System		Achieved Free-Run Scan Rate (sps)			-
CSD (CapSense, etc.) 0	CYBSP_CSD CSD-3.0 V	Achieved Scan Duration	<u> </u>		-
LCD Direct Drive 0	Icd_0_drive_0	✓ Connections			
Multi-Counter Watchdog Timer (MCWDT) 0	srss_0_mcwdt_0	Clock	<unassigned></unassigned>		$\sim$
Multi-Counter Watchdog Timer (MCWDT) 1	srss_0_mcwdt_1	Clock Frequency	<u>A</u>		~
Real Time Clock (RTC)	srss_0_rtc_0	<			>
		12-bit SAR ADC - Parameters Code Previe	2W		
Notice List				ć	8 ×
😢 0 Errors 🥂 0 Warnings 📋 5 Tasks 🚺 3	Infos				
Fix Description			Location		
📋 🥆 The AREF resource must be enabled for pass	_0_sar_0 to work.		CY8C624ABZI-S2D44: 12-bit SAR ADC		
The SAR ADC reference voltage must be hig	her than 0.85 V		CY8C624ABZI-S2D44: 12-bit SAR ADC		
The connection list for signal "Ch1 Vplus" ca	an't be empty		CY8C624ABZI-S2D44: 12-bit SAR ADC [V	plus Conne	ction]
The connection list for signal "Ch0 Vplus" ca	an't be empty		CY8C624ABZI-S2D44: 12-bit SAR ADC [V	plus Conne	ction]
The 'Clock' parameter must not be empty.			CY8C624ABZI-S2D44: 12-bit SAR ADC [C	lock]	
Ready					

3. Notice also that a new task may appear in the <u>Notice List pane</u>.

See <u>lcons</u> 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.

Name	Value	^
<ul> <li>Peripheral Documentation</li> </ul>		
? Configuration Help	Open SAR Documentation	
✓ General		
? Vref Select	Internal Reference (from AREF Resource)	-
Vref Voltage (V)	0.000	
? Number of Channels	2	]
Injection Channel		_
? Vref Bypass		
? Target Scan Rate (sps)	20000	
💧 Achieved Free-Run Scan Rate (sps)		_
💧 Achieved Scan Duration	<b>A</b>	
✓ Connections		
Clock	<unassigned></unassigned>	
🔺 Cld 🗐 Task: The 'Clock' parameter		
(?) EO must not be empty.	<unassigned></unassigned>	
Clock that operates this block	-	~
12-bit SAR ADC - Parameters Code Previo	ew	

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



#### Quick start

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

Code Preview é	7	×				
Enter search text	0	0,				
<pre>/* NOTE: This is a preview only. It combines elements of the   * cycfg_peripherals.c and cycfg_peripherals.h files located in the folder   * C:/Users/follettcj/mtw2.4/5645/hw/Hello_World/libs/TARGET_CY8CKIT-062-WIFI-BT/COMPONENT_BS   */</pre>	P_	>				
<pre>#include "cy_sar.h" #include "cycfg_routing.h" #include "cycfg_routing.h"</pre>						
<pre>#include "cy_sysclk.h" #if defined (CY_USING_HAL)     #include "cyhal_hwmgr.h" #endif //defined (CY_USING_HAL)</pre>						
<pre>#define pass_0_sar_0_HW SAR #define pass_0_sar_0_IRQ pass_interrupt_sar_IRQn #define pass_0_sar_0_CTL ((uint32_t)CY_SAR_VREF_PWR_100   (uint32_t)CY_SAR_VREF_SEL_BGR   (ui #define pass_0_sar_0_SAMPLE ((uint32_t)SAR_SAMPLE_CTRL_EOS_DSI_OUT_EN_Msk   (uint32_t)CY_SAR_ #define pass_0_sar_0_CH0_CONFIG (((uint32_t)SAR0_VPLUS0_PORT &lt;&lt; SAR_CHAN_CONFIG_POS_PORT_ADDR #define pass_0_sar_0_CH1_CONFIG (((uint32_t)SAR0_VPLUS1_PORT &lt;&lt; SAR_CHAN_CONFIG_POS_PORT_ADDR #define pass_0_sar_0_VREF_MV 1200UL</pre>	RI P					
<pre>const cy_stc_sar_config_t pass_0_sar_0_config =</pre>	>	¥				
12-bit SAR ADC - Parameters Code Preview						

- 7. Use the <u>various tabs</u> 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**

# 4 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 <u>Parameters pane</u>.

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 <code>init\_cycfg\_<resource-category></code>. The <code>init</code> 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 <code>init\_cycfg\_all()</code>, which calls all other generated functions, for example <code>init\_cycfg\_pins()</code>.

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**

# 5 GUI description

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

#### 5.1 Menus

## 5.1.1 File

- **New** Creates a new \*.*modus* file. See <u>Create Design dialog</u> 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 Libraries** Opens a dialog to select a different device support library (*props.json*) file used for resource <u>Personalities</u>.

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 <u>Personalities</u>. This opens a dialog showing all the personalities in the design.

For example, if you load a \*.modus file made with an older device support library, there might be many warnings in the <u>Notice List pane</u> 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.

## 5.1.2 Edit

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

#### 5.1.3 View

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

#### 5.1.4 Help

- View Help Opens this document.
- **About** Opens the **About** box for version information, with links to open Infineon.com and the current session log file.



#### **GUI description**

# 5.2 Device tabs

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

 CY8C624ABZI-S2D44
 CYW43012C0WKWBG

 Peripherals
 Pins
 Analog-Routing
 System
 Peripheral-Clocks
 DMA

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 <u>ModusToolbox<sup>™</sup> user guide</u> for more details.

#### 5.3 Resource tabs

See <u>Resource tabs</u>.

#### 5.4 Panes

See <u>Panes</u>.

#### 5.5 Icons

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

0	
(?)	Indicates there is a tooltip. Hover over the icon to display a brief message about the setting.
<b>V</b>	Enables or disables a specific resource.
8 🛦 🗐 🚺	There may be occasions where an error, warning, task, or info icon displays for an enabled resource. See <u>Notice List pane</u> for more details.
<b>A</b>	When shown in <b>Parameters</b> , this indicates that it is a read-only field. When shown for a <b>Resource</b> , this indicates the resource is locked and disabled. There is a tooltip explaining why the resource is locked.
2	When shown for a Resource, this indicates the resource is locked and enabled. There is a tooltip explaining why the resource is locked.
8	After assigning a signal, clicking this icon jumps to the linked resource(s).



# 6 **Resources tabs**

For most device families, the Device Configurator contains several tabs, each of which provides access to specific resources. However, for some device families, 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 <u>Parameters pane</u> displays various configuration options. As described under <u>lcons</u>, some enabled resources may contain errors, warnings, tasks, or infos that indicate some action might be required to resolve the issue. See <u>Notice List pane</u> 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.

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

Each of the tabs (except the **Analog-Routing** tab) 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 <u>Personality</u> 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.

Enter any string in this field. The tool converts the name into a legal C identifier and replaces nonlegal 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<sup>2</sup>C.
  - 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.

Note:



# 6.1 **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 <u>Names</u> for the resource. It also shows the selected <u>Personality</u>, where applicable.

# 6.1.1 Device families with tabs

Periphe	erals	Pins	Anal	og-Rout	ing	System	Peripheral-Clocks	DMA							
Enter filt	er text									R	7	E	Ŧ	K	D 1
Resource	e						Name(s)	Perso	nality						
> Anal															
	imunica							_							
	✓ Inter-	-IC Sour	nd Bus	(I2S) 0			audioss_0_i2s_0	I2S-1.	.0						
	Inter	-IC Sour	nd Bus	(I2S) 1			audioss_1_i2s_0								
	Quad	d Serial I	Memo	ry Interfa	ace (QS	PI) 0	smif_0								
Ć	SD ⊢	lost Cor	ntrolle	r (SDHC)	0		sdhc_0								
	SD H	ost Con	ntroller	(SDHC)	1		sdhc_1								
5	Seria	l Comm	nunicat	tion Bloc	k (SCB	) 0	scb_0	SPI-1.	.0	$\sim$					
	Seria	l Comm	nunicat	tion Bloc	k (SCB	) 1	scb_1								
E	Seria	l Comm	nunicat	tion Bloc	k (SCB	) 2	scb_2								
	Seria	l Comm	nunicat	tion Bloc	k (SCB	) 3	scb_3								
5	2 🗐 s	Serial Co	ommu	nication	Block (	(SCB) 4	scb_4	UART	-1.0	~					
	Seria	l Comm	nunicat	tion Bloc	k (SCB	) 5	scb_5								
. E	Seria	l Comm	nunicat	tion Bloc	k (SCB	) 6	scb_6								
	Seria	l Comm	nunicat	tion Bloc	k (SCB	) 7	scb_7								
E	Seria	l Comm	nunicat	tion Bloc	k (SCB	) 8	scb_8								
	Seria	l Comm	nunicat	tion Bloc	k (SCB	) 9	scb_9								
E	Seria	l Comm	nunicat	tion Bloc	k (SCB	) 10	scb_10								
	Seria	l Comm	nunicat	tion Bloc	k (SCB	) 11	scb_11								
E	Seria	l Comm	nunicat	tion Bloc	k (SCB	) 12	scb_12								
6	Z 🗐 L	Jniversa	al Seria	l Bus (US	B) 0		usb_0	USBFS	S Device-1.1	~					
> Digit															
✓ Syste								_							
6				c.) 0			CYBSP_CSD	CSD-2	2.0	$\sim$					
	_	Direct D					lcd_0_drive_0								
6	✓ Multi	i-Count	ter Wat	tchdog T	îmer (I	VICWDT) 0	srss_0_mcwdt_0	Multi	-counter watchdog-1	~ 0.					
	Mult	i-Count	ter Wat	tchdog T	îmer (l	VICWDT) 1	srss_0_mcwdt_1								
6	Real	Time Cl	lock (R	TC)			srss_0_rtc_0	Real T	Time Clock-1.1						
<															>

# 6.1.2 Device families without tabs

Enter filter text 🖉 🔻 🖻 🖏 🖏 🖏 🖏													
Resource	Name(s)	Personality			8	7	6	5	4	3	2	1	
✓ Peripherals						· ·	Ŭ	5	-	9	-	- 1	
ADC	adc_0	ADC-1.0	$\sim$										
ARM IO	armio_0				ADC_AVSS	X810,10	( P8 )	( P3 )	( P6 )	( P17 )	( P9 )	( P12 )	A
Audio	audio_0												
Bluetooth	bluetooth_0	Bluetooth-1.	0 ~		VDDO1	*140 TR	P15	P2	PS	P14	P11	P13	В
Clock	clock_0										$\bigcirc$		D
GCI-SECI	coex_0												-
☑ 12C	i2c_0	I2C-1.0			VDDC	THE REPORT	vssc	(P4)	P2.8	VSSC	P10	( P1 )	С
Keyboard Scanner	keyscan_0					_		$\smile$					
Low Noise/External Power Amplifiers	amplifiers_0				(HOST,WAKE)	LART, BRD	UART THE		P2.6	P29	( PO )	VDDO2	D
D PDM	pdm_0												-
PWM 0	pwm_0							RR.DO. DIGLDO	PMILIND, DNN SIMP				-
PWM 1	pwm_1				XTALO	DEV,WAKE		VDDIN	COMP STORE	VSSC	P37	VDDC	E
PWM 2	pwm_2												
PWM 3	pwm_3				XTALI	PLLVSS	(IFV DD)	Reserved	(PMU_AVSS		( P32 )	( P27 )	F
PWM 4	pwm_4							-					
PWM 5	pwm_5				PLLVDD	vcovss	PAVSS	Reserved	DIGLDO_	PMU AVDD	TAS 81	RSTIN	G
Quadrature Decoder	quadrature_0								VDDOUT				G
SPI 1	spi_0								RFLDO_				
SPI 2	spi_1	SPI-1.0			VCOVDD	IFVSS	RF	PAVID	VDDOUT	SR.PVDD	SR_VLX	SR_PVSS	Н
SWD	swd_0												
UART 1 (HCI UART)	uart_0					CVIA	12081	QA 1k	FBG	62_F			
UART 2 (PUART)	uart_1	UART-1.0	$\sim$		_					•		-	
> Pins	ioss_0	Pins-1.0				No (	Connect	Ass	igned	O Po	wer		
<				>		🔴 Erro	r	😑 De	dicated	O GF	10		



### 6.2 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 <u>Names</u> for the resource. It also shows the selected <u>Personality</u>, where applicable.

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.

# 6.2.1 Device families with tabs

CY8C624ABZI-S	2D44 CYW43012C0WKWBG														
Peripherals	Pins Analog-Routing System Peripher	al-Clocks DMA													
Enter filter text									2	7 🖻	Ŧ	K	ì	¢	<b>t</b> 22
Resource	Name(s)	Personality	13	12 11	I 10	9	8	7	6	5	4	3	2	1	
✓ Port 0						$\sim$	$\sim$	$\frown$	$\frown$	$\frown$		$\frown$			
	CYBSP_WCO_IN	Pin-1.1 🗠	VODIOA	VDDA P107	1) (P10[4]	) (P1Q7]	(P11(2)	(P11(5))	P12[0]	P12[3]	P12(6)	P13[1]	VCCD	VDDD	Α
	CYBSP_WCO_OUT	Pin-1.1 🔍	Var	VSS P10;	P 105	(P110)	(P11(3)	P11(6)	(P12(1))	P12[4]	P12[7]			P13[0]	в
P0[2]	ioss_0_port_0_pin_2														в
P0[3]	ioss_0_port_0_pin_3		P9(7)	(P10(9) (P10(9)	1) (P10(6)	) (P11[1]	P11(4)	(911(7))	P12[2]	P12(5)	V0000	VSS	(P13(4)	(P13[5]	с
✓ P0[4]	CYBSP_SW2,CYBSP_USER_BTN1,CYBSP_USER_BTN	Pin-1.1 🗠									_			$\leq$	-
P0[5]	CYBSP_LED_RGB_GREEN, CYBSP_USER_LED4		(P9(4)	(P9[5]) (P9[5	a) Azz	)						(P13[6])	(P13[7]	V BAC KUP	D
Y Port 1			$\square$	õõ							_			$\overline{\frown}$	-
✓ P1[0]	CYBSP_CSD_RX,CYBSP_CS_RX,CYBSP_CS_TX_RX	Pin-1.1 $\sim$	P9(2)	(P9[1]) (P9[0								PO[0]	P0[1]	P0[2]	E
P1[1]	CYBSP_LED_RGB_RED,CYBSP_USER_LED3		P8(6)	P8(7) P9(1								P0[3]	P0541		F
P1[2]	ioss_0_port_1_pin_2				2									-	· .
P1[3]	ioss_0_port_1_pin_3		P8(3)	(P8(4) (P8(1	n)							PO[S]	P1[0]	(P1[1])	G
P1[4]	CYBSP_SW4,CYBSP_USER_BTN2														
P1[5]	CYBSP_LED8, CYBSP_USER_LED1, CYBSP_USER_LEE	0	P8(0)	P8(1) P8(2	n)							P1[2]	P1[3]	P1[4]	н
> Port 2			P7(5)	P7[6] P7[								P1[5]		VDD.NS	
> Port 3			(Pr(p)	(Prile)	2							Pilpi	-	ADD'UR	,
> Port 4 > Port 5			P7(2)	V00101 P7[4	22V	)									к
> Port 6											_				
> Port 7			(P7(0)	(P7[1] (P6[	I) (P6[1]	) (P5[4])	(PS[1])	( P40	(18(3))	PS(0)	V00102		(PH/BL/ USECP	USBON	L
> Port 8				õč							$\overline{}$				
> Port 9			P6(7)	P6[5] P6[3	I) (P6[0]	) (P5[5]	PSZ	P41	P3[4]	PS[1]				VIDUSB	М
> Port 10 > Port 11			P7[3]	P6[6] P6[4	P5[7]	P5[6]	PSI	PSQ	PSIS	P3[2]	P2[7]				Ν
> Port 12															
> Port 13					C	/8C62	4ABZ	-S2D	44 (12	4-BG	A)				
> Port 14						Assigne		Power		Error	<u> </u>				
<			>		•	Dedicat	ed 🕓	GPIO							
L									_		_				

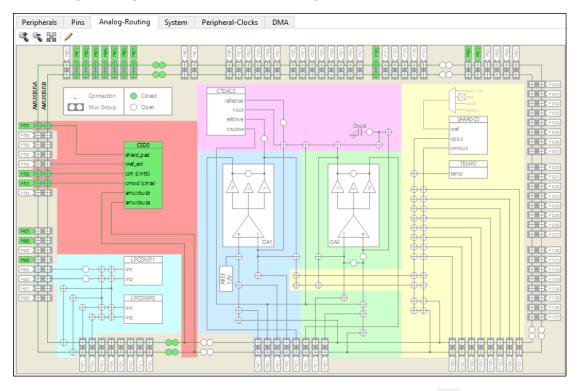


#### Enter filter text. 🥖 🔻 🖻 🖌 🗎 🗖 🔍 🕺 Resource Personality Name(s) Peripherals Pins 8 7 6 5 4 3 2 1 Pins-1.0 🗸 ioss\_0 DEV\_WAKE ioss\_0\_pin\_40 P8 P6 P17 P1 2 А ioss\_0\_pin\_41 🗹 P0 CYBSP\_D2,SW3,USER\_BUTTON1 Pin-1.0 ✓ P1 CYBSP\_RST Pin-1.0 В ✓ P2 CYBSP D4 🗹 РЗ CYBSP\_D5 С 🗹 P4 CYBSP\_D6 ✓ P5 Pin-1.0 CYBSP\_D7 ✓ P6 CYBSP\_D11,SPI2\_MOSI D CYBSP\_A0, CYBSP\_THERM\_TEMP\_SENSE ✓ P8 P9 CYBSP D13, SPI2 CLK Е ✓ P10 CYBSP\_A2 ✓ P11 CYBSP\_RSVD9,SPI2\_CS Pin-1.0 ✓ P12 CYBSP\_A4 Pin-1.0 F P2 7 P32 P13 CYBSP A5 ✓ P14 CYBSP\_D8 G 🗹 P15 CYBSP\_D10 ✓ P17 CYBSP\_D12,SPI2\_MISO Pin-1.0 ✓ P26 LED2 Pin-1.0 Н P27 LED1 P28 12C SCL CYW20819A1KFBG (62-FBGA) ✓ P29 I2C\_SDA ✓ P32 CYBSP\_D1,UART\_TX Pin-1.0 No Connect Assigned Power ✓ P37 CYBSP\_D0,UART\_RX Error Dedicated GPIO

# 6.2.2 Device families without tabs

# 6.3 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** *C* command opens the <u>Analog Route</u> <u>Editor</u>.

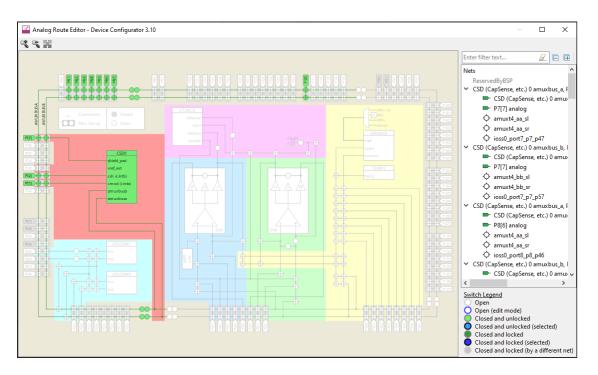


# 6.3.1 Analog Route Editor

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:

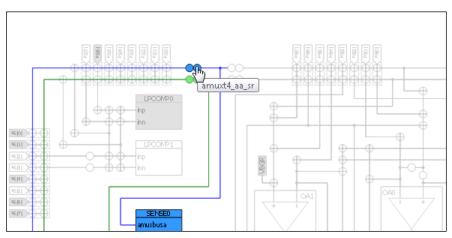
The Analog Editor can be some times be unresponsive during route recalculation. Please wait for it to finish algorithm execution.



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.

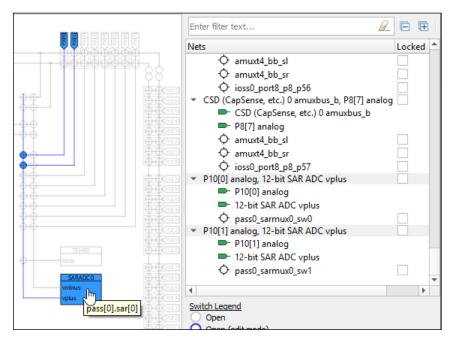
## 6.3.1.1 Select a resource

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 <u>Edit</u> <u>Route</u>.

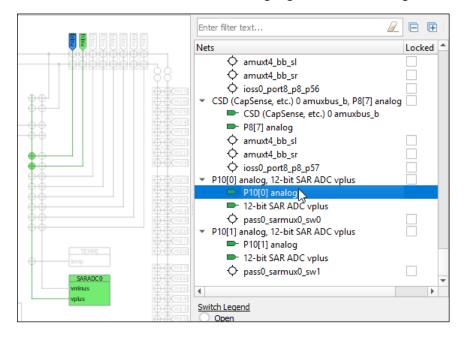




At the same time, the selected analog resource(s) is highlighted in the Nets tree.



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



# 6.3.1.2 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.



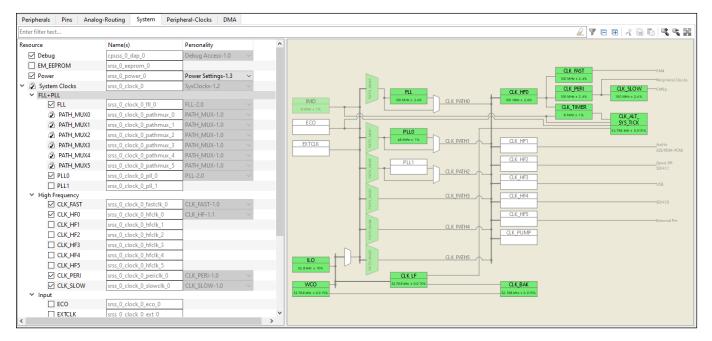
If a change results in an error, a message displays. The routes are automatically rolled back to the previous state, so you will lose at most the last invalid change.

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.

## 6.4 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 <u>Names</u> for the resource. It also shows the selected <u>Personality</u>, where applicable.



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.



# 6.5 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 <u>Names</u> for the resource. It also shows the selected <u>Personality</u>, where applicable.

Peripherals	s Pins	Analog-Routing	System	Peripheral-C	locks	DMA					
Enter filter t	ext						$\overline{\mathbf{v}}$	E	Ħ	K	Đ
Resource		Name(s)			Persor	nality					
✓ 8 bit											
2	8 bit Divider	0 CYBSP_CSD_CL	K_DIV,CYBS	P_CS_CLK_DIV	Periph	eral Clock	-1.0				
2	8 bit Divider	1 peri_0_div_8_1			Periph	eral Clock	-1.0				
L 8	8 bit Divider	2 peri_0_div_8_2									
L 8	8 bit Divider	3 peri_0_div_8_3									
🗌 8	8 bit Divider	4 peri_0_div_8_4									
L 8	8 bit Divider	5 peri_0_div_8_5									
🗌 8	8 bit Divider	6 peri_0_div_8_6									
	8 bit Divider	7 peri_0_div_8_7			]						
> 16 bit ✓ 16.5 bit											
	16.5 bit Divid	ler 0 peri_0_div_16_5	0								
		ler 1 peri_0_div_16_5	-								
		ler 2 peri_0_div_16_5									
	16.5 bit Divid	ler 3 peri_0_div_16_5	3								
> 24.5 bit			-		J						
c											2

# 6.6 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 <u>Names</u> for the resource. It also shows the selected <u>Personality</u>, where applicable.

Peripherals	Pins	Analog-Routin	ng S	System	Peripher	ral-Clocks	DI	MA				
Enter filter text						R	7	E	Ħ	K	Ð	P
Resource			Nam	e(s)		Personality	1					^
DMA Cont	roller					_						
🗹 DM	A Chann	el 0	cpuss	_0_dmad	_0_chan_0	DMAC-1.0						
🗌 DM	A Chann	el 1	cpuss	_0_dmad	_0_chan_1							
DM	A Chann	el 2	cpuss	_0_dmad	_0_chan_2	]						
🗌 DM	A Chann	el 3	cpuss	_0_dmad	_0_chan_3	1						
DMA Data	Vire 0					_						
🗌 DM	A DataW	ire 0: Channel 0	cpuss	_0_dw0_	0_chan_0							
DM	A DataW	ire 0: Channel 1	cpuss	_0_dw0_	0_chan_1							
🗌 DM	A DataW	ire 0: Channel 2	cpuss	_0_dw0_	0_chan_2							
DM	A DataW	ire 0: Channel 3	cpuss	_0_dw0_	0_chan_3	]						
🗌 DM	A DataW	ire 0: Channel 4	cpuss	_0_dw0_	0_chan_4							
DM	A DataW	ire 0: Channel 5	cpuss	_0_dw0_	0_chan_5	1						
DM	A DataW	ire 0: Channel 6	cpuss	_0_dw0_	0_chan_6	1						
DM	A DataW	ire 0: Channel 7	cpuss	_0_dw0_	0_chan_7	1						
🗌 DM	A DataW	ire 0: Channel 8	cpuss	_0_dw0_	0_chan_8	1						
DM	A DataW	ire 0: Channel 9	cpuss	_0_dw0_	0_chan_9	1						
🗆 DM	A DataW	ire 0: Channel 10	cpuss	_0_dw0_	0_chan_10	1						
DM	A DataW	ire 0: Channel 11	cpuss	_0_dw0_	0_chan_11	1						
	A DataW	ire 0: Channel 12	cpuss	; 0 dw0	0 chan 12							
		ire 0: Channel 13	-			1						
 DM	A DataW	ire 0: Channel 14	cpuss	0 dw0	 0 chan 14							
		ire 0: Channel 15										
		ire 0: Channel 16	<u> </u>									
		ire 0: Channel 17	-									
		ire 0: Channel 17 ire 0: Channel 18	-			-						
	A Dataw	ne o. channel 10	repuss	_o_uwo_	o_cnan_lo	1					;	



# 7 Panes

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

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

## 7.1 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 <u>launch a separate configurator</u>.

TCF	WM[0] 32-bit Counter 0 - Par	ameters		Ð	×
Ent	er filter text		🖉 🖉	E	Ŧ
Na	me	Value			
~	Peripheral Documentation				
	⑦ Configuration Help	Open Timer - Counter (TCPWM) Documentation			
~	General				
	? Clock Prescaler	Divide by 1			~
	⑦ Counter Resolution	32-bits			
	? Run Mode	Continuous			~
	⑦ Count Direction	Up			~
	? Period	32768			_
	⑦ Compare or Capture	Capture			~
~	Capture				
	⑦ Capture Input	Disabled			~
~	Interrupts				
	⑦ Interrupt Source	None			~
~	Inputs				
	Clock Signal	<unassigned></unassigned>			~
	⑦ Count Input	Disabled			~
	③ Stop Input	Disabled			~
	? Reload Input	Disabled			~
	⑦ Start Input	Disabled			~
~	Outputs				
	Overflow	<unassigned></unassigned>			
	⑦ Underflow	<unassigned></unassigned>			
	? Capture (cc_match)	<unassigned></unassigned>			
~	Advanced				
	③ Store Config in Flash				

# 7.1.1 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.

# 7.1.2 Parameter descriptions

As described under <u>lcons</u>, all parameters have a tooltip icon ⑦ to indicate there is information about the parameter. Hover the mouse cursor over the icon to display a description of the parameter.



## 7.1.3 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**  $\mathscr{P}$  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:

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

# 7.1.4 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.

✓ CapSense Sensors			
③ Button0_Rx0	C	P8[1] analog (CYBSP_CSD_BTN0, CYBSP_CS_BTN0) [SHARED]	~
③ Button0_Tx	P	P1[0] analog (CYBSP_CSD_TX, CYBSP_CS_TX) [SHARED]	~
⑦ Button1_Rx0	C	<ul> <li>P10[5] analog (CYBSP_A5, CYBSP_J2_11)</li> <li>P10[6] analog (CYBSP_A6, CYBSP_J2_13)</li> </ul>	^
② Button1_Tx	C	P10[7] analog	
② LinearSlider0_Sns0	C	<ul> <li>P0[0] analog (CYBSP_WCO_IN) [USED]</li> <li>P0[1] analog (CYBSP_WCO_OUT) [USED]</li> </ul>	
② LinearSlider0_Sns1	C	P0[1] analog (CYBSP_WCO_OOT) [USED]	
② LinearSlider0_Sns2	C	P0[3] analog (CYBSP_LED_RGB_RED, CYBSP_USER_LED3) P0[4] analog (CYBSP_SW2, CYBSP_USER_BTN1, CYBSP_USER_BTN)	
② LinearSlider0_Sns3	C	P0[4] analog (CYBSP_SW2, CYBSP_USER_BIN1, CYBSP_USER_BIN1) P0[5] analog	
② LinearSlider0_Sns4	P	P1[0] analog (CYBSP_CSD_TX, CYBSP_CS_TX) [SHARED]	~

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**  $\mathscr{O}$  button to jump to the selected signal or open a dialog to select from multiple signals.



# 7.1.5 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 <u>Peripherals Tab</u>, the **Parameters** pane contains a **<Configurator>** parameter, where **<**Configurator**>** is the name of the other configurator.

`	~	External Tools	
		? CapSense Configurator	Launch CapSense Configurator
		? CapSense Tuner	Launch CapSense Tuner

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

# 7.2 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.

Notice List	t	문 ×		
21 Err	rors 🔥 0 Warnings 🗐 1 Task 🚺 0 Infos			
Fix	Description	Location ^		
	The 'Clock Signal' parameter must not be empty. CY8C62478ZI-D54: TCPWM[0] 32-bit Counter 6 [Clock Sign			
🖸 🔧 🔻	The personality 'lp_assist_wifi-1.0', instantiated at 'wifi[0].power[0]', is not available in any selected library.	CYW4343WKUBG: Wi-Fi		
😢 🍾 🔻	The personality 'lp_assist_bt-1.0', instantiated at 'bt[0].power[0]', is not available in any selected library.	CYW4343WKUBG: BT		
8	The Low Frequency Clock Source (typically ILO) should be enabled	CY8C6247BZI-D54: LCD Direct Drive 0		
👩 🔧 🗸	The CLK_HE0 clock is not enabled	CY8C6247B7I-D54: CSD (CapSense_etc.) 0		
Ready				

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):

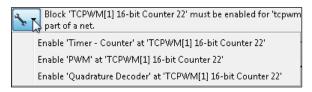
- Icon Displays the icons for the error, warning, task, or info.
- **Fix** This may display a wrench icon, which can be used to automatically address the required notice.
- **Description** Displays a brief description of the notice.
- **Location** Displays the specific line number or other location of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

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.



# 7.2.1 Fix a task/error

When a wrench `` icon displays in the **Fix** column, click on it and select the appropriate action from the pulldown 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.

# 7.2.2 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.

# 7.3 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			ð
hal			0
<pre>/* NOTE: This is a preview only. It combines * cycfg_peripherals.c and cycfg_peripherals * C:/Users/cycfg_per/mtw2.3/3728/hw/mtb_sha */</pre>	.h files loo	cated in the folder	-v2.
<pre>#include "cy_smif.h" #include "cycfg_qspi_memslot.h" #if defined (CY_USING_HAL) #include "cyhal_hwmgr.h" #endif //defined (CY_USING_HAL)</pre>			
Adofine anif 0 UN SMIRO	<u>C</u> opy		
<pre>#define smif_0_HW SMIF0 #define smif 0 IRQ smif interrupt IRQn</pre>	Select Al	I	
#define smif 0 MEMORY MODE ALIGMENT ERROR (0	u		
#define smif_0_RX_DATA_FIF0_UNDERFLOW (0UL)	Find	Ctrl+F	
<pre>#define smif_0_TX_COMMAND_FIF0_OVERFLOW (0UL</pre>	) 🔍 Find next	t F3	
<pre>#define smif_0_TX_DATA_FIF0_OVERFLOW (OUL)</pre>	Find prev	vious Shift+F3	
<pre>#define smif_0_RX_FIF0_TRIGEER_LEVEL (OUL) #define smif_0_TX_FIF0_TRIGEER_LEVEL (OUL)</pre>	Find prev	Nous Shirt+F5	
<pre>#define smif_0_TX_FIF0_TRIGEER_LEVEL (0UL) #define smif 0 DATALINES0 1 (0UL)</pre>			
#define smif 0 DATALINES2 3 (OUL)			
#define smif 0 DATALINES4 5 (OUL)			
<pre>#define smif_0_DATALINES6_7 (0UL)</pre>			
<pre>#define smif_0_SS0 (OUL)</pre>			
<pre>#define smif_0_SS1 (0UL)</pre>			
#define smif 0 SS2 (OUL)			
Quad Serial Memory Interface (QSPI) 0 - Parameters	ode Preview		r

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.



# Version changes

# 8 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 <b>Platform</b> tab to <b>System</b> 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 <b>File</b> 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.							
2.21	Added feature to support incremental patch updates.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.							
	Removed Change Devices dialog.	Use command line tools instead.						
3.0	Updated the tool to recognize obsolete devices.	Displays an error in the Notice List instead of blocking the file from being opened.						
	Added path to files in the Code Preview Pane.	ed device is changed to not check against non-						
3.10	Fixed the name conflict warning notices when the selected device is changed to not check against non- existent chip locations.         Added a 'Fix' to name conflict warnings in the notice list.         Locked down number entry to the English format (decimal place is always a period).         Made various performance improvements.         Added a search filter to the dialog that allows for the selection of multiple signals.         Added a log file that can be accessed from the About box. Debug messages are redirected to it.         Added back the 'Fix' to the MPN consistency notice.         Fixed error message displayed when the same *.modus file is opened in multiple instances of the Device-Configurator and mpn checking command line arguments are provided.							

# ModusToolbox™ Device Configurator user guide



# Version changes

Version	Change Descriptions	Notes
	Updated back-end to support multi-core applications.	
4.0	Updated Create Design dialog.	
	Changed device library file from xml to <i>props.json</i> .	



# **Revision history**

# **Revision history**

Revision	Date	Description
**	2018-11-21	New document.
*A	2019-03-01	Updated to version 1.1.
*В	2019-10-17	Updated to version 2.0.
*C	2020-03-27	Updated to version 2.1.
*D	2020-09-01	Updated to version 2.20.
*E	2020-12-10	Updated to version 2.21.
*F	2021-03-15	Updated to version 3.0.
*G	2021-09-24	Updated to version 3.10.
*H	2022-09-12	Updated to version 4.0.

#### Trademarks

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

Edition 2022-09-12

**Published by** 

Infineon Technologies AG

81726 Munich, Germany

© 2022 Infineon Technologies AG. All Rights Reserved.

Do you have a question about this document? www.cypress.com/support

Document reference 002-24377 Rev. \*H

#### IMPORTANT NOTICE

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

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

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

The data contained in this document is exclusively intended for technically trained staff. It is the responsibility of customer's technical departments to evaluate the suitability of the product for the intended application and the completeness of the product information given in this document with respect to such application. For further information on the product, technology, delivery terms and conditions and prices please contact your nearest Infineon Technologies office (www.infineon.com).

#### WARNINGS

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

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