ModusToolbox™ tools package release notes

ModusToolbox™ tools package version 3.0.0

About this document

Scope and purpose

This document describes the features and known limitations for the ModusToolbox™ software provided as part of the ModusToolbox™ tools package included with the installer.

ModusToolbox™ software is a set of tools that enable you to integrate our devices into your existing development methodology. ModusToolbox™ software consists of various libraries and middleware on GitHub, as well as an IDE and tools package installed on your computer. For more details about what is included with ModusToolbox™ software, refer to the ModusToolbox™ tools package user guide. See also What's included in this document.

This ModusToolbox™ tools package is a complete release. It includes the latest features from all previous releases, including patches. This release does not replace any existing installed releases; it installs alongside them. If you have more than one release installed, refer to the ModusToolbox™ tools package user guide, "Product versioning" section.

Reference documents

Refer to the following documents for more information as needed:

- ModusToolbox™ tools package user guide
- ModusToolbox™ tools package installation guide
- Training material on GitHub

Release Notes
www.infineon.com

Please read the Important Notice and Warnings at the end of this document
page 1 of 16

002-22557 Rev. *T
2022-09-27

Table of contents

1 Primary changes

The overall version 3.0 release includes the following updates and features:

1.1 Multi-core platform support

One of the major features of version 3.0, and all subsequent 3.x releases, is support for multi-core platform device families. This change requires a substantial update to the ModusToolbox™ configuration, build, programming, and debugging capabilities. The multi-core platform enables features such as high-end graphics and instruction trace, which require various changes such as:

- Multi-core application development
- Multi-core device configuration
- BSP development flows and tools
- PSoC™ 6 ETM support in µVision and EW-ARM

1.1.1 Multi-core application development

The ModusToolbox™ build and program/debug support has been extended to support multi-core devices such that code can be developed on all (user-accessible) cores.

1.1.2 Multi-core device configuration

The Device Configurator has been enhanced to enable the generation of peripheral configuration code for all cores in a multi-core device from one *design.modus* file.

1.1.3 BSP development flows and tools

One of the major changes made for ModusToolbox™ version 3.x is that most BSPs are no longer Git repos when you create a version 3.0 application. Instead, the BSP becomes owned by the application. This means you can change various aspects of the BSP as you see fit without having to create a custom BSP, because creating a version 3.x application effectively creates a custom BSP.

This release provides an improved experience for creating and managing BSPs, including:

- A new BSP Assistant tool to create a BSP from an existing BSP and to edit an existing BSP.
- An improved process to create an application-owned BSP during application creation.
- An improved process when adding a new BSP to the application via the Library Manager.
- An import mechanism to copy template firmware files from libraries to the application during application creation.

Note: *In ModusToolbox™ version 2.4, applications were de-coupled from the Git repos for the code example templates they were based upon.*

1.1.4 PSoC™ 6 ETM support in µVision and EW-ARM

Embedded Trace Macrocell (ETM) instruction trace has been enabled on all PSoC™ 6 devices. Trace is offered on third-party IDEs only, using the SEGGER J-Trace hardware and the IDE vendor's own tools.

Primary changes

1.2 Core build infrastructure

We have replaced the core build infrastructure with C++ implementations: mtbgetlibs, mtbideexport, mtbquery, and mtbsearch. These reduce the implementation required in Makefiles, and they provide a uniform view of manifests, projects, and tools to the various C++ applications included with the ModusToolbox™ tools package.

You will see new subdirectories with these names in the *tools_3.0* directory. You are not expected to use these implementations directly, but rather continue using the existing make targets and variables.

1.3 Infineon Developer Center

The ModusToolbox™ tools package has been integrated with the Infineon Developer Center (IDC) (https://softwaretools.infineon.com/tools/com.ifx.tb.tool.modustoolbox). IDC is a software distribution system where you can find and download Infineon products from the web. Additionally, you can install the IDC desktop launcher tool to find, install, and manage various versions of Infineon products. Plus, IDC will provide notifications when there are updates.

1.4 ModusToolbox™ packs

ModusToolbox™ version 3.0 now supports delivering packs with tools and libraries separate from a regular tools package release. For example, the ModusToolbox™ Machine Learning solution has been removed from the tools package installation. However, that solution will be available to download and install from our website as a pack in the near future. This allows us to provide a complete solution targeted at machine learning, while the tools package focuses on core technology. Additionally, packs allow us to deliver early access to support new device families or to provide completely new features. This allows you to experiment with the new devices or features as we improve and update them in subsequent packs.

Watch for updates on our website about when packs will be available.

1.5 Reset button in VS Code

The Cortex-Debug plug-in for Visual Studio (VS) Code is adding a **Reset** button that causes a soft reset on the device and halts the processor at the start of the application. This is supported for both SEGGER J-Link and OpenOCD connections.

1.6 Support for new operating systems

This release officially adds support for Windows 11 and macOS Monterey.

For a complete explanation of system requirements and supported operating systems, refer to the ModusToolbox™ tools package installation guide.

2 What's included

This release includes the following tools and versions:

Tool Name	Current Release	Previous Release
Bluetooth® Configurator	2.60	2.50
BSP Assistant	1.0	N/A
CAPSENSE™ Configurator & Tuner	5.0	4.0
CyBridge	3.4.0	3.3.0
cymcuelftool	1.0 (no change)	1.0
Device Configurator	4.0	3.10
Device Firmware Update (DFU) Host Tool	1.60	1.50
Eclipse IDE for ModusToolbox™	3.0	2.4.0
EZ-PD™ Configurator	1.20	1.10
Firmware Loader (fw-loader)	3.4.0	3.3.0
GCC	10.3.1 (no change)	10.3.1
GNU make Build System (tools-make)	2.0	1.6.0
JRE	11.0 (no change)	11.0
KitProg3	2.40	2.30
Library Manager	2.0	1.40
LIN Configurator	1.20	1.10
modus-shell	1.3.1.69 (Windows) 1.3.1.69 (Linux/macOS)	1.20 (Windows) 1.1.0 (Linux/macOS)
Core build infrastructure: • mtbgetlibs • mtbideexport • mtbquery • mtbsearch	1.0.0	N/A
OpenOCD (ModusToolbox™-specific)	4.4.0	4.3.0
Project Creator	2.0	1.40
Proxy Helper	1.30	1.20
Python (for Windows) • cysecuretools • pyocd	3.8.10 • 4.1.0 • 0.27.3 (no change)	3.7.155.7 • 3.1.0 • 0.27.3
QSPI Configurator	4.0	3.0
Secure Policy Configurator	1.30	1.20
Segment LCD Configurator	1.50	1.40
Smart I/O Configurator	4.0	3.10
USB Configurator	2.50	2.40

What's included

2.1   Supported tool chains

The GCC Arm Embedded toolchain GCC 10.3 is installed with the ModusToolbox™ software. This toolchain has no use restrictions and does not require license activation (it is distributed under the terms of the GNU Public License).

Although not installed with ModusToolbox™ software, the build system also supports these tool chains for most applications and devices (see the application *README.md* file for applicable support):

- Arm compiler v6 or newer (Windows and Linux hosts)
- IAR Embedded Workbench v9 or newer (Windows only)

2.2   Supported boards

The boards available for use varies with different releases of BSPs and libraries on GitHub. You can see the current list of BSPs in the Project Creator tool using the default manifest URL:

Kit Name	MCU/SOC/SIP	Connectivity
> AIROC™ Bluetooth® BSPs		
∨ AIROC™ Connectivity BSPs		
CYW943907AEVAL1F	CYW43907KWBG	\<none\>
CYW954907AEVAL1F	CYW54907KWBG	\<none\>
∨ PMG1 BSPs		
PMG1-CY7110	CYPM1011-24LQXI	\<none\>
PMG1-CY7111	CYPM1111-40LQXIT	\<none\>
PMG1-CY7112	CYPM1211-40LQXIT	\<none\>
PMG1-CY7113	CYPM1311-48LQXI	\<none\>
∨ PSoC™ 4 BSPs		
CY8CKIT-041-41XX	CY8C4146AZI-S433	\<none\>
CY8CKIT-041S-MAX	CY8C4149AZI-S598	\<none\>
CY8CKIT-045S	CY8C4548AZI-S485	\<none\>
CY8CKIT-145-40XX	CY8C4045AZI-S413	\<none\>
CY8CKIT-149	CY8C4147AZI-S475	\<none\>
PSOC4-GENERIC	CY8C4548AZI-S485	\<none\>
∨ PSoC™ 6 BSPs		
CY8CEVAL-062S2	CY8C624ABZI-S2D44	\<none\>
CY8CEVAL-062S2-LAI-4373M2	CY8C624ABZI-S2D44	CYW4373EUBGT
CY8CEVAL-062S2-MUR-43439M2	CY8C624ABZI-S2D44	CYW43439KUBG
CY8CKIT-062-BLE	CY8C6347BZI-BLD53	\<none\>
CY8CKIT-062S2-43012	CY8C624ABZI-S2D44	CYW43012C0WKWBG
CY8CKIT-062S4	CY8C6244LQI-S4D92	\<none\>
CY8CKIT-062-WIFI-BT	CY8C6247BZI-D54	CYW4343WKUBG
CY8CKIT-064B0S2-4343W	CYB0644ABZI-S2D44	CYW4343WKUBG
CY8CKIT-064S0S2-4343W	CYS0644ABZI-S2D44	CYW4343WKUBG
CY8CPROTO-062-4343W	CY8C624ABZI-S2D44	CYW4343WKUBG
CY8CPROTO-062S3-4343W	CY8C6245LQI-S3D72	CYW4343WKUBG
CY8CPROTO-063-BLE	CYBLE-416045-02	\<none\>
CY8CPROTO-064B0S1-BLE	CYB06447BZI-BLD53	\<none\>
CY8CPROTO-064B0S3	CYB06445LQI-S3D42	\<none\>
CY8CPROTO-064S1-SB	CYB06447BZI-D54	\<none\>
CYBLE-416045-EVAL	CYBLE-416045-02	\<none\>
CYSBSYSKIT-01	CY8C624AFNI-S2D43	CYW43012TC0KFFBH
CYSBSYSKIT-DEV-01	CY8C624AFNI-S2D43	CYW43012TC0KFFBH
CYW9P62S1-43012EVB-01	CY8C6247FDI-D52	CYW43012TC0EKUBG
CYW9P62S1-43438EVB-01	CY8C6247BZI-D54	CYW43438KUBG
PSOC6-GENERIC	CY8C6347BZI-BLD53	\<none\>
∨ XMC™ BSPs		
KIT_XMC14_BOOT_001	XMC1404-Q064x0200	\<none\>
KIT_XMC47_RELAX_V1	XMC4700-F144x2048	\<none\>
XMC-GENERIC	XMC1404-Q064x0200	\<none\>

Note: Additional boards will be made available on an ongoing basis.

2.3   Open source

Portions of this software package are licensed under free and/or open source licenses such as the GNU General Public License. Such free and/or open source software is subject to the applicable license agreement and not our license agreement covering this software package. The applicable license agreements are available online:

> *https://www.infineon.com/cms/en/design-support/software/free-and-open-source-software-foss/modustoolbox-foss-packages/*

3 Design impact

This section includes issues and solutions for changes that may impact various designs.

3.1 Limited 3.x code examples

The ModusToolbox™ 3.x ecosystem has been developed looking forward to newer devices with greater capabilities and features. The 3.x software also offers newer features including expanded multi-core support and application-owned BSPs that you can modify as needed without having to create a new custom BSP.

We also have many code examples for existing devices that use the 2.x application structure. These 2.x applications fully function in the 3.x ecosystem, and they can be easily migrated to the 3.x structure. Over time, these code examples will be updated to the 3.x structure and released online. When the update is complete, you can use them to create 3.x applications without the need for migration.

3.2 Migrating ModusToolbox™ applications from version 2.x to version 3.x

To migrate a ModusToolbox™ version 2.x application to the version 3.x ecosystem, follow instructions in KBA236134. Do not update individual libraries to major new versions, because the new versions are likely not compatible. KBA236134 provides instructions for replacing your BSP and associated libraries with compatible versions.

3.3 Legacy dual-core and multi-project applications not supported

As described under [Multi-core platform support](#), there are substantial changes to the ModusToolbox™ version 3.0 build system, device configuration, and BSP development flows. These changes are required to support a broader range of devices with more than one core. Due to these changes, legacy dual-core and multi-project applications from ModusToolbox™ version 2.x are not supported in the version 3.0 ecosystem. If you try to create or import these types of legacy applications, you will see a message similar to the following:

> "the directory 'C:/views/git/mtb-env/TestData/apps/newappdir/testproject' does not contain the file 'Makefile'"

Infineon recommends that you create a new application using the ModusToolbox™ version 3.0 tools, and then copy/paste source code from the legacy application into the new application.

3.4 No arguments for make getlibs target

With the update to the make build system for 3.0, the `make getlibs` command no longer takes any arguments such as `TARGET` or `TOOLCHAIN`. If you used any arguments from previous versions of ModusToolbox™ applications, you'll have to remove them.

If you need to specify the `TARGET` or TOOLCHAIN, update the application Makefile as appropriate.

3.5 ModusToolbox™ version 2.1 "LIB" flow no longer supported

In ModusToolbox™ version 2.2, we introduced the "MTB" flow where BSPs and libraries used .mtb files. At the same time, we supported the previous "LIB" flow for versions 2.2, 2.3, and 2.4. Due to the restructuring of applications in ModusToolbox verion 3.0, the old "LIB" flow is no longer supported.

If you have an older application using the "LIB" flow, we recommend creating a new version 3.0 application, and copying over the source code.

Design impact

3.6   Issue with setting CY_BUILD_LOCATION variable

By default, build artifacts are created in the *./build* directory in the application. If you have built an application with the default location, and then later set the `CY_BUILD_LOCATION` variable in the *Makefile* to another directory, the build will fail. This is because the default build location is not being ignored as it was in version 2.x.

There are two ways to solve this problem:

- Delete the original *./build* directory, or
- Set the `CY_IGNORE` variable for the *./build* directory.

3.7   .cyignore files not detected

Due to the restructuring of applications in ModusToolbox 3.0, .cyignore files are not detected unless they are located in the project or middleware asset root. If you encounter an issue with a .cyignore file, ensure it is located in the proper location.

4 Known issues/limitations

This section lists the known issues/limitations of this release:

4.1 ModusToolbox™ issues from previous releases

This document contains only new issues pertinent to ModusToolbox™ version 3.x. All issues noted in previous ModusToolbox™ version 2.x releases have been made available online here: KBA236147.
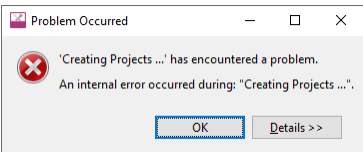
4.2 Proxy

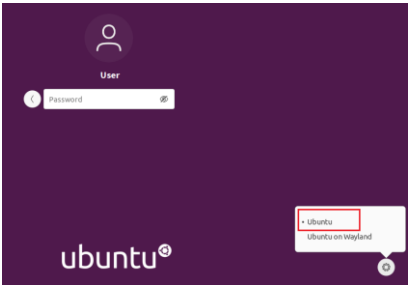| Problem	Workaround
When trying to use any of the ModusToolbox™ tools, you may see an error message similar to the following about not able to connect to the Internet: *Unable to open file at [URL]*	This can happen if you are behind a firewall and do not have your proxy settings configured. You must set your HTTP_PROXY and HTTPS_PROXY environment variables, as appropriate for your site. In previous versions of the ModusToolbox™ tools package, these types of errors only affected the Library Manager and Project Creator. In version 3.x, these errors apply to all tools. **Note**: The Proxy Helper tool only works for Library Manager and Project Creator. You must set the proxy variables manually for all other tools to work.

4.3 Eclipse IDE

Problem	Workaround
In the Eclipse IDE for multi-core applications, the code in the subprojects is not indexed (cannot go to definitions of external symbols, header includes are marked as not found).	Right-click the affected subproject in the Project Explorer and select **Build Project**. After the build is complete, indexing should work for that project. If you make changes to the application code, you may have to rebuild the project.
When renaming a multi-core application in the Eclipse IDE, the core-processor subprojects become regular folders instead of project folders.	Right-click on each of the core process folders and select **Import as Project**. Each folder will become a project folder with the appropriate name. **Note**: After changing the folders to project folders, they retain the original application prefix name. You can rename them individually if you prefer. **Note**: The Eclipse IDE may display Indexer errors during the rename process. These errors usually go away when the rename process is complete.
After completing a build/clean for a multi-core application, the Eclipse IDE reports a C/C++ Indexer error message.	The error message doesn't impact functionality. Try restarting the Eclipse IDE to clear the message.
When using the Eclipse IDE on macOS, if you delete a multi-core application with several README.md files open in the editor/viewer, you may see the following error message: "Failed to create the part's controls."	None. This error message just means the markdown viewer is trying to read a file that has been deleted. This message doesn't impact functionality. Simply close the tab and proceed.

Known issues/limitations

Problem	Workaround
For BTSDK applications, various tools such as BTSpy and ClientControl are not available from the Eclipse IDE Quick Panel. This is due to the restructuring of the ModusToolbox™ version 3.0 build system and flow.	To open various tools, navigate to where they are located in the workspace and launch them manually. By default, all the tools are located under the *../mtb_shared* directory (relative to the code example directory). For example, BTSpy is located in the following path: *../mtb_shared/wiced_btsdk/tools/btsdk-utils* ClientControl is located in the following path: *../mtb_shared/wiced_btsdk/tools/btsdk-host-apps-bt-ble* Other tools are located in different subdirectories under *mtb_shared*. The BTSDK will be updated in a future release to move these tools into a pack so that they will be available in the Quick Panel again.
Exporting and importing a multi-core application with the Eclipse IDE doesn't work.	1. When Exporting a multi-core application using the Eclipse IDE, include only the main application folder, and exclude the core project folders and the *mtb_shared* folder. 2. If you exported the application as an archive, extract it, and then run Library Manager or `make getlibs`. 3. In Eclipse, use the **Import Existing Application In-Place** option. The Eclipse IDE for ModusToolbox™ user guide has been updated to include multi-core application instructions.
When importing an application in the Eclipse IDE using the **Import Existing Application In-Place** option, there may be an error: "An internal error occurred..." 	If possible, you should start with the Eclipse IDE to launch Project Creator and create an application. Using this flow, the process just works. In some cases, this error may occur because you're attempting to manually import an application in the same workspace/folder that already exists in Eclipse. To resolve this problem, use **Import > General > Existing Projects into Workspace**.
When using the Eclipse IDE on macOS, the context menu may show tools for the previously-selected project rather than the project that was right-clicked.	Left-click to select the project before right-clicking it to launch the context menu. This issue will be addressed in a future release.
If you make changes to the BSP using the BSP Assistant, such as the connectivity module, and then launch the Device Configurator from the Eclipse IDE Quick Panel, the Device Configurator may not recognize that the changes were made.	To fix this, close the Device Configurator, click **Refresh Quick Panel** in the Eclipse IDE, and then launch the Device Configurator again.
The Eclipse IDE has indexing issues. For a single-core application, indexing doesn't work in files outside of the application directory (such as in the *mtb_shared* directory). For multi-core applications, indexing only works in the core project directories (for example, cm4) and does not work in the application's *bsp* directory. For example in the *main.c* file, if you right-click on `#include "cybsp.h"` and select **Open Declaration**, when that file opens, IntelliSense doesn't work to locate `#include "cy_result.h"`.	None. Due to the structure of applications in the Eclipse IDE, indexing can only find files in the main application and core projects. To locate headers outside the application, use the search option. However, be aware that there's no guarantee that there aren't multiple headers with the same name in different shared directories.

4.4 Visual Studio Code

Problem	Workaround
Various ModusToolbox™ GUI tools (Device Configurator, Library Manager, etc.) fail to start on Ubuntu when executed from the Terminal window in VS Code, with the following error: Failed to load client buffer integration: "wayland-egl" Available client buffer integrations: () No shell integration named "xdg-shell" found No shell integration named "xdg-shell-v6" found No shell integration named "wl-shell" found No shell integration named "ivi-shell" found Loading shell integration failed. Attempted to load the following shells ("xdg-shell", "xdg-shell-v6", "wl-shell", "ivi-shell")	**Solution 1**: Open an Ubuntu Terminal, navigate to the application directory, and use the applicable "make" command to open the GUI (e.g., `make library-manager`). **Solution 2**: Log out from the current Ubuntu session, and then log back in and make sure Ubuntu on Wayland is **not** selected on the login screen. On newer versions of Ubuntu, the login screen will show "Ubuntu on Xorg".

4.5 Library Manager/make getlibs

Problem	Workaround
The Library Manager does not list the retarget-io library for the CY8CKIT-040T.	The retarget-io library is not available currently for this device. The retarget-io library depends on mtb-hal-cat2, which is not supported by the CY8CKIT-040T.
A .mtb file added as a direct dependency to the *deps* directory is overridden and removed during the `make getlibs` operation.	This problem occurs when you add a direct dependency using the 'latest-x.y' tag instead of a specific 'release-x.y.z' tag when the BSP or another library requires the same library as an indirect dependency. To ensure your direct dependency is kept, use the 'release-x.y.z' tag in your .mtb file.

4.6 CySecureTools

Problem	Workaround

Using the policy from CySecureTools 4.0.0 in the projects created by CySecureTools 4.1.0 causes an error during re-provisioning on PSoC64-2M devices:
 	
ERROR : SFB status: CY_FB_INVALID_IMG_JWT_SIGNATURE: Invalid image certificate signature. Check the log for details
	1. Open the policy file.
	2. Navigate to the first section of boot_upgrade/firmware.
	3. Set `boot_auth` and `bootloader_keys` as follows:
	`"boot_auth": [`
	`    3`
	`],`
	`"bootloader_keys": [`
	`    { "kid": 3, "key": "../keys/cy_pub_key.json" }`
	`]`

4.7   Building/programming/debugging

Problem	Workaround
After changing the Memory Part Number for EMC72_EVK devices using the QSPI Configurator from "Auto detect SFDP" to "S25FL512S," programming the external QSPI flash using the KitProg3 may fail with the following errors: Error: flash write algorithm aborted by target Error: error writing to flash at address 0x60000000 at offset 0x00000000	Set the Memory Part Number in the QSPI Configurator back to the default "Auto detect SFDP." This works with most flash devices, including S25FL512S. If, in your specific use case, you have to specify the memory part number explicitly and observing the error as above, use the J-Link debugger.

On Windows 11, the KitProg3 UART driver is not certified, and this causes issues for the UART interface, CAPSENSE™ Tuner, and CyBridge for the StopBits, ParityType, DataBits, and BaudRates parameters.	As a short-term solution, manually install the Windows 10 driver:
	1. In the Device Manager under the **Ports** tab, find USB Serial Device item (COMX).
	2. Right-click on it and select **Update driver > Browse my computer for drivers**
	3. Navigate to the appropriate driver:
	   • The path for the drivers is <ModusToolbox™ Install_DIR>/tools_3.0/kp-firmware/drivers/.
	   • Select either KitProg3 or MiniProg4, then the UART option, then Win10 and Win10x64.
	4. Click **Next >** and then **Close**.
	This issue will be resolved in a future release.

In the Eclipse IDE, programming a device is skipped when the device was previously programmed.	This is expected behavior and there is no workaround. This was a change made to the launch configurations in Eclipse to support multi-core debugging.
If you change the MCU/SOC/SIP for the BSP using the BSP Assistant, the Register View may not available while debugging using the Eclipse IDE or VS Code.	This is because the svd file path is not present in the launch configurations. To fix this, navigate to the application folder and run `make getlibs`. The, run `make eclipse` or `make vscode`, as applicable.
Debug doesn't work on AIROC™ CYW20829 devices with a J-Link probe in JTAG in Eclipse and VS Code IDEs.	Use the SWD interface.
The **Attach** launch config does not work for the secure lifecycle of AIROC™ CYW20829 devices.	Do not use the **Attach** launch config for secure lifecycle.
The CySecuretool does not work with PyOCD as the debugger on macOS M1 CPUs.	No workaround; we do not plan to support PyOCD.
When exporting ModusToolbox™ applications created for the TRAVEO™ T2G Body High MC devices to be used with the µVision and IAR IDEs, there are some limitations using the J-Link probe (including ETM traces, program, and debug via the CM7 core).	Use native Arm and IAR probes to work with TRAVEO™ T2G Body High MC devices.
There is an issue with reset for TRAVEO™ T2G Body High MC devices in the SEGGER tool. The **Debug** launch config does not work properly for CM7_0 and CM7_1 cores via the J-Link probe in Eclipse and VS Code IDEs.	To debug the code, use the **Attach** launch config.

Known issues/limitations

Problem	Workaround
When performing various program/debug operations using the Eclipse IDE on Windows, the plugin has a defect that may prevent the debug port from powering down.	Reset the device. For example, on the CY8CKIT-062S2-40312 kit, press the **SW1/XRES** button.

4.8 BSP Assistant

Problem	Workaround
On Windows, the BSP Assistant does not receive error messages from any configurator that outputs text (such as device-configurator-cli). This means that the BSP Assistant may indicate a failure, but not report any error messages from that configurator.	As an example, if the device-configurator-cli fails when changing devices, but no message explains why, go to c:/Users/<username>/AppData/Local/Temp/device-configurator-cli and find the most recently-created log file.

4.9 Device Configurator

Problem	Workaround
In the Device Configurator, the analog routing line to the SAR is not highlighted when a connection is made in the editor for XMC7xxx devices.	None. This is a display-only issue. There is nothing wrong with the data. The issue will be addressed in a future release.
For ModusToolbox™ 2.x applications using PSoC™ 4 devices, if you open the application using ModusToolbox™ version 3.x, the Device Configurator (version 4.0) may show various errors with various types of connections, such as SAR ADC SOC Input, TCPWM Start/Stop/Capture/Count/Reload Signal, DMA Channel Trigger Input or connections between pins over Digital Input and Digital Output.	Select new signals for the invalid connections to resolve the issues.
For ModusToolbox™ 2.x applications using PSoC™ 4 or PSoC™ 6 devices, if you open the application using ModusToolbox™ version 3.x, the Device Configurator (version 4.0) may show "Resource overused" errors in some cases where the pin's Digital Input, Digital Output, and Analog parameters are set at the same time.	Select new signals for the invalid connections to resolve the issues.
If you have a project and design.modus file with different manufacturing part numbers (MPNs), and one of the MPNs is a module, the Device Configurator displays a message stating that the MPNs differ. However, included in that message is a statement that the MPNs are not supported by the device-db. This statement is incorrect.	You can safely ignore the incorrect statement in the message. There is no impact to the application.

Known issues/limitations

Problem	Workaround
If one of the WICED Radio Interface personalities is instantiated in the Universal Digital Block (UDB), it will not allow picking connections for the DMA signals.	The WICED Radio Interface personalities should not be used in practice. Instead, to enable communication with an external radio device from a PSoC 6S1 device, the BSP/Application should use the udb-sdio-whd library and set the appropriate component for the desired port. See the README.md file contained in the udb-sdio-whd library for more details. Additionally, to reserve the appropriate resources so they do not get overused by the configurator, the board should include a cyreservedresources.list file next to the design.modus file. Refer to one of the existing PSoC 6S1 boards (eg: CY8CKIT-062-WIFI-BT, CYW9P62S1-43012EVB-01, CYW9P62S1-43438EVB-01, …) for what this file should contain.

4.10 CAPSENSE™ Tuner

Problem	Workaround
Switching between Synchronized and Asynchronized mode in the CAPSENSE™ Tuner may not be executed correctly for PSoC™ 4000T projects in cases where the device is working in deep sleep mode for substantial periods of time.	In order for the CAPSENSE™ Tuner to switch between Synchronized and Asynchronized mode, the device should be in active mode with working I^{2}C or UART communication channel.
UART communication may not work reliably in the CAPSENSE™ Tuner for PSoC™ 4000T.	There is no workaround. UART communication will be available in the production release for the PSoC™ 4000T device.

4.11 Documentation

Problem	Workaround
Various documents included with the release may contain incomplete information, or may not contain up-to-date screen captures or information.	New versions of documents, including this release notes document, may be available online at: ModusToolbox™ Software website

Revision history

Revision	Date	Description of Change
**	2017-12-29	Initial Release.
*A	2018-11-21	Updates for Production.
*B	2019-09-19	Updates for version 1.1.
*C	2019-10-18	Updates for version 2.0.
*D	2019-10-21	Copyright text for FreeType.
*E	2019-11-04	Added known issues.
*F	2020-03-26	Updates for version 2.1.
*G	2020-04-07	Adding Known Issues for version 2.1; fixed Rev. in footer.
*H	2020-04-21	Adding known Issue for QSPI Configurator.
*I	2020-05-20	Removed fixed known issue.
*J	2020-06-29	Added known issues for Programming/Debugging.
*K	2020-09-01	Updates for version 2.2.
*L	2020-09-25	Added several known issues.
*M	2020-10-19	Added issue about programming in DeepSleep mode.
*N	2020-12-04	Added issue about Eclipse IDE and macOS Big Sur.
*O	2020-12-10	Added issued for no access to GitHub.
*P	2021-03-25	Updates for version 2.3. Updated text for M1 workaround. Added issue about importing into Eclipse.
*Q	2021-04-22	Updated text for known issue about build failing in Eclipse for imported projects.
*R	2021-07-07	Added issue for PSoC™ 4 error message that can be ignored.
*S	2021-09-23	Updates for version 2.4.
*T	2022-09-27	Updates for version 3.0 release.

Trademarks

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