

# ModusToolbox™ 1.1 Release Notes

## Production Release

### Overview

ModusToolbox™ software is a set of tools that enable you to integrate Cypress devices into your existing development methodology. This release is an update to ModusToolbox 1.0. It does not replace the existing version; it installs alongside it. The main pieces of this software include the PSoC 6 software development kit (SDK), Bluetooth SDK, the ModusToolbox IDE, and Cypress Programmer. ModusToolbox is also available through the Mbed ecosystem, supporting PSoC 6 plus Wi-Fi connectivity devices.

This document describes the features and known limitations for the ModusToolbox software included in this release.

### Contents

Overview .....	1
Contents .....	1
What's Changed.....	2
What's Included.....	2
PSoC 6 Software Library .....	3
Bluetooth SDK .....	3
Mbed Ecosystem Support .....	4
ModusToolbox IDE .....	4
Configurators and Tools .....	4
Utilities .....	5
Design Impact .....	5
Updating from ModusToolbox 1.0 .....	5
Supported Tool Chains .....	5
Supported Boards .....	6
Known Issues/Limitations.....	6
Installation.....	6
ModusToolbox IDE .....	7
Documentation.....	10
Source Control.....	10
Programming Tools .....	10
PSoC 6 Software Library .....	10
Bluetooth SDK.....	11
General Configurator Issues.....	12
Device Configurator .....	12
QSPI Configurator .....	12

Device Firmware Update Host Tool.....	12
Open Source .....	12
Further Reading .....	13

## What's Changed

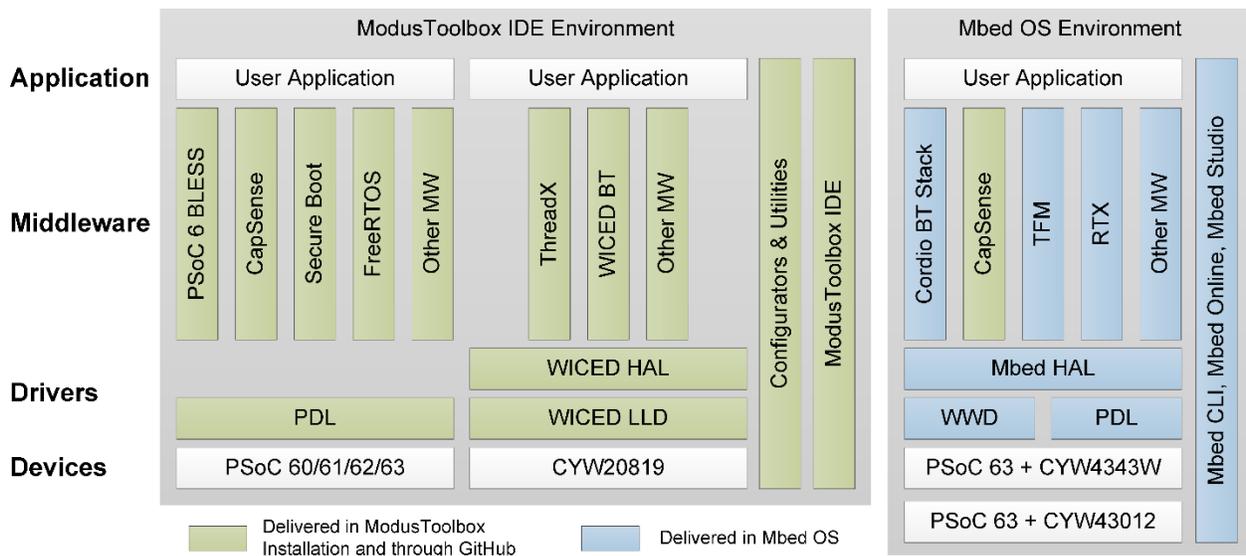
This section provides a high-level overview about what changed from ModusToolbox 1.0 to 1.1.

- Added support for WICED Bluetooth CYW20819 device in the IDE
- Enabled usage of ModusToolbox software elements for Wi-Fi applications through the Mbed ecosystem
- Updated PSoC Software Library to version 1.1
  - Updated Configurators
  - Updated PDL

**Note** ModusToolbox 1.1 is not backward compatible with the 1.0 release. You must create a new application and copy across your various files. See [Updating from ModusToolbox 1.0](#) for more details.

## What's Included

The following diagram shows a high-level view of everything included in the ModusToolbox software.



**Note** Some pieces of Cypress Programmer are included as part of command line tools with the ModusToolbox software. You can also install Cypress Programmer, including a GUI, separately.

## ***PSoC 6 Software Library***

Cypress provides the PSoC 6 Software Library (psoc6sw-1.1) as a part of the ModusToolbox installation. This library simplifies software development for the PSoC 6 family of devices. It consists of two main parts: Peripheral Driver Library (PDL) and middleware/software libraries. The combination of driver-level code with integrated middleware and RTOS provides a path to high-level software solutions for PSoC 6 applications.

Code examples demonstrate solution usage, serve as the evaluation platform, and can be used as a project starting point for customers to start product prototyping and development. The library enables customers to focus on application development and differentiating products, rather than spending energy creating a software development platform.

This release provides the following features:

- PSoC 6 digital and analog peripheral drivers, which enable rapid peripheral software development
- The ARM Cortex Microcontroller Software Interface Standard (CMSIS) core access header files directly from the CMSIS 5.3.0 release
- CMSIS compliant device header files, startup code (platform initialization) and device configuration header files
- A Device Firmware Update library that can be readily integrated for any application need
- FreeRTOS source code integrated with the PDL
- Bluetooth Low Energy (BLE) middleware and stack library
- USB device peripheral driver and middleware library
- CapSense and CSD ADC middleware library
- Emulated EEPROM middleware library
- SEGGER EmWin middleware library
- Secure Image reference design, which demonstrates a Trusted Execution Environment (TEE) on the CM0+ core
- PDL Application Programming Interface (API) Reference Guide

This release supports all devices in the PSoC 6 MCU architecture. Some drivers support peripherals only available on the PSoC 62 series with 2M flash (CY8C6xx8, CY8C6xx9, CY8C6xxA).

For full details on both the drivers and the middleware libraries, refer to the library documentation, located here: `<install_dir>\libraries\psoc6sw-1.1\docs`, which includes the *PSoC 6 Software PDL User Guide* and the *PDL API Reference Guide*.

## ***Bluetooth SDK***

The Bluetooth SDK is targeted for the CYW20819 ultra low power Bluetooth 5.0 SoC and the ModusToolbox IDE. ModusToolbox 1.1 with the Bluetooth SDK provides a complete development environment to allow you to quickly create Bluetooth enabled IoT solutions like smart watches, medical devices, or home automation platforms. The Bluetooth SDK includes the following:

- Bluetooth firmware
- Platform and board support packages
- Build system

- Local tools including BTSpy trace utility and debugger
- Various sample applications

### ***Mbed Ecosystem Support***

Mbed OS is an open-source real-time operating system (RTOS) for the Internet of Things. Cypress has enabled several kits (see [Supported Boards](#)) for use with the Mbed OS to leverage their ecosystem including TLS, networking, storage and drivers, and many code examples and libraries. For more information about Mbed OS, refer to:

<https://os.mbed.com/docs/mbed-os/v5.11/introduction/index.html>

### ***ModusToolbox IDE***

This is an Eclipse-based IDE that uses parts of the SDK. The IDE facilitates creating applications and adding additional Cypress firmware to them. There are several parts of the ModusToolbox IDE, including:

- ModusToolbox Perspective – This is a plugin that provides ease-of-use features in the IDE, including the Quick panel and debug mode.
- New Application Wizard – This is an Eclipse-specific wizard used for creating new applications for Cypress devices.
- Update Manager – This is an Eclipse-specific dialog used for updating and adding SDKs.
- Code Editor – This is the standard Eclipse code editing environment.
- Middleware Selector – This is an Eclipse-specific dialog used for transitioning your application to include/remove assorted devices, as well as adding/removing middleware.
- CYW20819 Bluetooth Project Settings – This is an Eclipse-specific dialog used for changing various platform and applications settings.

### ***Configurators and Tools***

Configurators are GUI-based tools provided as part of the SDK. In general, you interact with these tools from the ModusToolbox IDE. However, these tools can also be used separately, and they can be run from the command line. Each of these tools provides a guide, available from the tool's **Help** menu.

- Device Configurator: Set up the system (platform) functions, as well as the basic peripherals (e.g., UART and Timer, etc.).
- CapSense Configurator and Tuner: Configure CapSense, test it, and generate the required firmware.
- USB Configurator: Configure USB settings and generate the required firmware.
- QSPI Configurator: Configure external memory and generate the required firmware.
- Bluetooth Configurator: Configure Bluetooth settings and generate the required firmware.
- Smart I/O™ Configurator: Configure Smart I/O settings and generate the required firmware.
- BT Spy: This is a trace viewer utility for CYW20189 Bluetooth platforms to view protocol and application trace messages from the embedded device.
- Client Control: These are tools for CYW20189 Bluetooth BLE Profile and BLE Mesh applications. One emulates the host MCU applications for BLE and BR/EDR profile and the other emulates the host MCU applications for BLE Mesh models.

## Utilities

The PSoC SDK includes the following utilities.

- CyMCUEIfTool: Merges core executable images into a single file for programming / debugging.
- OpenOCD: Open On-Chip Debugger provides debugging, in-system programming, and boundary-scan testing for embedded target devices.
- Cypress KitProg Firmware Loader: Used to upgrade firmware on Cypress kits. It allows you to switch the KitProg firmware from KitProg2 to KitProg3, and back
- JRE: Java Runtime Environment. Required for Eclipse plugin.

## Design Impact

### Updating from ModusToolbox 1.0

ModusToolbox 1.1 does not support the Cortex M0+ core in PSoC 6 devices. Accordingly, if you selected the “Cortex M0+” as the target CPU core in ModusToolbox 1.0 in the respective configurators for CapSense, BLE, or USB, you must re-design the application to run wholly on the Cortex M4.

1. From the ModusToolbox IDE, create a new application targeting the same board used in the original application. Use the “EmptyPSoC6App” template.
2. Copy (overwrite) your CM4 application sources from the original to the new application.
3. Copy (overwrite) the whole GeneratedSources folder from the original to the new application.
4. Copy (overwrite) the *design.modus* file from the original to the new application.
5. Open the Device Configurator and select **Update Referenced Libraries** from the **File** menu.
6. On the dialog, use the **psoc6sw** pull-down menu to select “1.1 [USED]” and click **Commit**.
7. If any of the BLE, CapSense, QSPI, Smart I/O, and USB resources are used in your application, launch the associated configurator, save the configuration, and close the configurator.
8. Save the configuration from the Device Configurator and close it.
9. From the ModusToolbox IDE, open the Middleware Selector and add all the packages used in the original application (including the middleware for the functions that were implemented on the Cortex M0+).

## Supported Tool Chains

The GCC Arm Embedded toolchain GCC 7.2.1 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).

## Supported Boards

This release provides support for the following boards.

Board	MCU	Connectivity <sup>[1]</sup>	Mbed Support
CY8CPROTO-062-4343W	CY8C624ABZI-D44	CYW4343W Wi-Fi + Bluetooth	Yes
CY8CMOD-062-4343W	CY8C624ABZI-D44	CYW4343W Wi-Fi + Bluetooth	Yes
CY8CKIT-062-4343W	CY8C624ABZI-D44	CYW4343W Wi-Fi + Bluetooth	Yes
CY8CKIT-062-WIFI-BT	CY8C6247BZI-D54	CYW4343W Wi-Fi + Bluetooth	Yes
CY8CKIT-062-BLE	CY8C6347BZI-BLD53	On-chip Bluetooth	Yes
CY8CPROTO-063-BLE	CYBLE-416045-02	On-chip Bluetooth	No
CYBLE-416045-02 EZ-BLE	CYBLE-416045-02	On-chip Bluetooth	No
CYW920819EVb-02	CYW20819	On-chip Bluetooth	No
CYBT-213043-MESH	CYW20819	On-chip Bluetooth	No

## Known Issues/Limitations

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

### Installation

Problem	Workaround
ModusToolbox can only be installed with Administrator Privileges and only once on a Windows machine. If a user other than the one who installed it wants to use ModusToolbox, that user will need Administrator Privileges also.	This issue will be addressed in the next release.
On common Linux distributions, the serial UART ports (usually /dev/ttySx or /dev/ttyUSBx devices) belong to the root user and to the dialout group. Standard users are not allowed to access these devices.	An easy way to allow the current user access to the Linux machine's serial ports is by adding the user to the dialout group. This can be done using the following command: <pre>\$sudo usermod -a -G dialout \$USER</pre> <b>Note</b> For this command to take effect, the user must log out and then log back in.
On some Windows machines, the Norton Security Suite antivirus software detects <i>wiceddriverinstaller_64.exe</i> as containing a virus.	This is a false alert. If "Auto-Protect" is enabled in Norton, it will delete the <i>wiceddriverinstaller_64.exe</i> file and drivers won't be installed.
On Windows 10, installation might be slow due to Anti-Malware virus scanning.	You may wish to disable virus scanning while installing.
The CY8CKIT-062-BLE and CY8CKIT-062-WIFI-BT boards are programmed with KitProg2 firmware by default. This firmware is not supported.	Update the boards to KitProg3 for use with ModusToolbox software. Refer to the <i>ModusToolbox IDE User Guide</i> , "KitProg Firmware Loader" section.

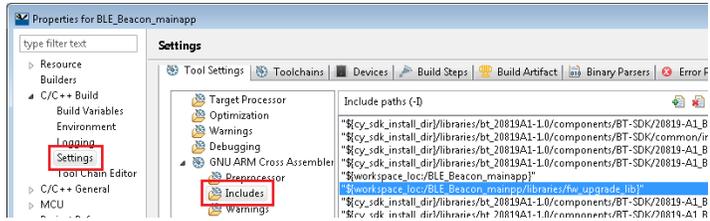
<sup>1</sup> Wi-Fi is only supported through the Mbed OS flow.

## ModusToolbox IDE

### Application Changes

Problem	Workaround
Changes made to an application outside of the ModusToolbox IDE will not automatically be included in the application.	After making changes to an application, use the Refresh command to include those changes in the ModusToolbox IDE. You can also select the Eclipse workspace setting "Refresh using native hooks or polling" to ensure updates are included automatically.

### Missing Project Libraries

Problem	Workaround
<p>Libraries that appear in the IDE Project Explorer under the &lt;project_name&gt;_mainapp\libraries\&lt;library_name&gt;\ folder are not added to the Project's -I Include path list automatically. If a source file in the project includes a header file under such folder, it will actually include the template copy of that header file, not the local copy under the libraries\&lt;library_name&gt; folder. This is only an issue if the header under such folder needs to be modified.</p>	<p>Add the local copy path to the Project Settings for the Compiler and Assembler Include paths, directly under the &lt;project_name&gt;_mainapp entry:</p> <ol style="list-style-type: none"> <li>Open the Project's Properties dialog and select <b>C/C++ Build &gt; Settings &gt; GNU ARM Cross Assembler &gt; Includes</b>.</li> <li>In the Include paths list in the right pane, select the entry for "\${workspace_loc:/&lt;project_name&gt;_mainapp}" (for example, "\${workspace_loc:/BLE_Beacon_mainapp}" in the screen capture below), and click the <b>Add</b> icon to create a new entry: <pre>"\${workspace_loc:/&lt;project_name&gt;_mainapp/libraries/&lt;library_name&gt;}"</pre> </li> <li>Manually enter the &lt;project_name&gt; instead of using the \$ProjName variable. <p>For example, to work around the issue in the BLE_Beacon application for the libraries\fw_upgrade_lib library, the end result would be:</p> <pre>"\${workspace_loc:/BLE_Beacon_mainpp/libraries/fw_upgrade_lib}"</pre> </li> </ol>  <li>Repeat the above process for the similar entry under the <b>GNU Arm Cross C Compiler &gt; Includes</b> setting.</li>

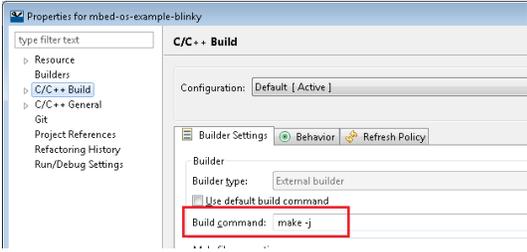
### Quick Panel

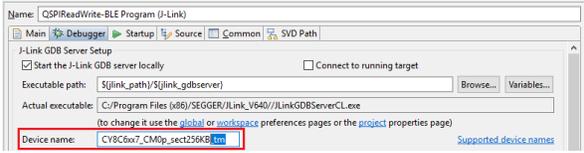
Problem	Workaround
After creating an application, the <b>Quick Panel</b> links are not available.	You need to select some project or file in the Project Explorer to enable links in the <b>Quick Panel</b> , such as "Device Datasheet" or "Program."

### Error/Warning Messages

Problem	Workaround
<p>On Windows, if you see one of the following errors, it indicates that a rare corruption condition has occurred in the MSYS shared memory sections.</p> <p><i>"No modus.mk based examples are available"</i></p> <p>or</p> <p><i>**** Couldn't reserve space for cygwin's heap, Win32 error 0"</i></p>	<p>The fix is to logout and login to your machine, effectively clearing the problem condition. If this problem persists, try rebooting the system.</p>
<p>There are known issues with the Eclipse CDT Indexer that can show incorrect errors and/or warnings in source files.</p>	<p>If errors and/or warnings are encountered, try forcing Eclipse to re-index (Project -&gt; C/C++ Index -&gt; Re-build).</p>
<p>The following warning is displayed while building a ModusToolbox application:</p> <p><i>"/bin/sh: line 0: igncr: invalid option name"</i></p>	<p>This warning does not affect the application. It means that SHELLOPTS=igncr is set in your environment. To stop seeing this warning, remove this environment variable.</p>

### Build, Program, Debug

Problem	Workaround
<p>On Windows 7 machines, the first build/compile of an imported Mbed project may fail with the message "Error 2".</p> <p>This is caused by "-j" parallel make option.</p>	<p>Open the Project Properties to <b>C/C++ Build</b>, locate the <b>Build command</b> field, and remove the "-j" option from the make command.</p>  <p>Apply, close, and build the project. It will take a bit longer than with the "-j" option; however, it is only needed for the first build of an imported Mbed project. You can add the -j back for future builds.</p>
<p>Adding multiple pre-build or post-build steps causes build failures.</p>	<p>To include more than one step as a pre-build or post-build command, create a script for all of the pre-build steps and another for all the post-build steps. Then include only the appropriate script as a pre-build or post-build command.</p>

Problem	Workaround
ModusToolbox IDE or command line applications fail to build when the workspace name, application name, or ModusToolbox installation path includes non-ASCII characters.	When installing ModusToolbox, or creating workspaces and applications, use only ASCII characters in the paths or names.
Cannot start a second debug session for CM4 during multi-core multi-device debugging on Linux and macOS.	Simultaneously starting more than three sessions of debug/program on Linux and macOS is not supported.
User must manually reset after programming PSoC 6 kits when using GDB SEGGER + Jlink + JTAG interface.	<p>Update each of the following Launch Configurations under the <b>Debugger</b> tab. In the Device Name field, delete the "_tm" suffix.</p>  <ul style="list-style-type: none"> <li>• "&lt;app-name&gt; Debug CM0+ (J-Link)",</li> <li>• "&lt;app-name&gt; Erase (J-Link)",</li> <li>• "&lt;app-name&gt; Program (J-Link)",</li> <li>• "&lt;app-name&gt; Program CM0+ (J-Link)"</li> </ul>
Debug does not halt on CM0 using GDB SEGGER + Jlink + JTAG interface.	<p>Update the Debug Configuration for "&lt;app-name&gt; Debug CM0+ (J-Link)", under the <b>Startup</b> tab:</p> <p>Unselect <b>Pre-run/Restart reset</b>, and add the following commands in the text box :</p> <pre>set \$sp = *(uint32_t*)0x10000000 set \$pc = *(uint32_t*)0x10000004</pre>
After programming a PSoC 6 project when using a J-Link probe, a debug session may be left behind. This prevents further programming or debugging.	You must manually stop the debug session to resolve this.
For multi-CPU devices only, the CM4 CPU hangs in the while loop when the CM0+ is stopped (debug breakpoint) in the following PDL drivers: syspm, flash, and effuse.	Release the CM0+ CPU to execute a syscall, requested by the CM4.

### Miscellaneous

- The workspace “Build Automatically” setting is programmatically turned off by ModusToolbox IDE project builds. This helps avoid certain errors that can otherwise occur.
- Applications can be migrated from one operating system (e.g., macOS) to another (e.g., Windows) by adjusting the post-build step. However, applications cannot be shared between two people on different operating systems at the same time.
- During some long-running operations within the ModusToolbox IDE (including ModusToolbox Middleware Selector), a dialog may indicate that a project “is out of sync with the file system.” This happens because files on disk can be changed early in the process, but the project refresh request does not happen until the end. The dialog can be ignored.

## 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 these release notes, may be available online at: <a href="http://www.cypress.com/modustoolbox">www.cypress.com/modustoolbox</a>

## Source Control

Problem	Workaround
When using ModusToolbox applications with egit or other git clients, there may be problems with building or updating the application.	<ul style="list-style-type: none"> <li>If using Eclipse / egit: Make sure your project builds cleanly before adding it to git using "Team → Add to Index". Failure to do so may cause Eclipse to add inappropriate files to source code control.</li> <li>If using another Git client, make sure you have a .gitignore file with the following contents BEFORE adding files to git:            Debug/            Release/            .metadata/</li> </ul>

## Programming Tools

Refer also to the Cypress Programmer 2.1 Release Notes.

Problem	Workaround
Segger J-Link Commander and J-Flash Lite tools cannot read and do not write eFuse bits correctly. These tools do not fill in gaps in the hex file with the correct values for eFuse bits, and use a different read technique than the Segger J-Flash tool.	Use the Segger J-Flash tool. To program eFuse bits with J-Link Commander and J-Flash Lite tools. Manually edit the hex file so that the eFuse region (0x90700000-0x907003FF) is filled in with data. Fuses, which must not be touched, should be filled with the 0xFF (ignore) value in that hex file.
When upgrading firmware with Firmware Loader you see this error: "The application is not valid and cannot be set active" If a target device is disconnected while Firmware Loader is upgrading firmware, the device cannot subsequently be updated.	Upon connecting with the Firmware Loader, downgrade to KitProg2: <code>fw-loader --update-kp2 [device-name]</code> Then upgrade to KitProg3: <code>fw-loader --update-kp3 [device-name]</code>
After changing the UART communication speed of the DAPLink via a terminal window, the communication speed doesn't change.	Send a <Break> command to the target from the terminal window

## PSoC 6 Software Library

Problem	Workaround
The Watchdog Timer (WDT) lock state is not retained during system Deep Sleep power mode. After wakeup from system Deep Sleep, the WDT is locked.	If you want to change the configuration of WDT, Clk LF, ILO, or PILO, unlock the WDT (call <code>Cy_WDT_Unlock()</code> ) after wakeup from system Deep Sleep.

Problem	Workaround
PSoC 6 does not support cache coherency. As a result when a particular row of flash that executes instructions is written/updated, the updated information will not be reflected in the cache.	The cache should be invalidated in the firmware during such instances. This is applicable for both CM4 and CM0+ cache. In other words, the appropriate (CM0+ and/or CM4) cache should be invalidated.
The PDL reserves certain system resources for internal use. These resources include IPC resources, like the first 16 IPC semaphores and interrupt lines to the Cortex M0+ CPU.	Using any of these resources in your design will lead to unexpected behavior. Refer to the PDL API Reference Guide for details.
The function malloc() does not return an error when the allocation size is bigger than the heap size, because PDL does not implement the _sbrk function.	This will be addressed in a future release.

### Bluetooth SDK

Problem	Workaround
The Bluetooth SDK does not warn of duplicate definitions.	Ensure that your application code does not have duplicate data definitions. This is existing issue with WICED SDK and will be addressed in the next release.
OTA Secure update fails if Linux/Mac tools are used to generate keys	Use the Windows tools to generate security keys.
When using BLE HID Keyboard app with Client Control, 'Enter Pairing' button does not work sometimes.	The enter pairing button might need to be pressed twice. Run the BT Spy tool and verify that the device is advertising.
OTA Firmware upgrade does not appear to work with Windows peer application.	Some old models of Windows PC using older BT devices with BT version 4.0 or earlier may not work correctly in responding to GATT notifications. Use newer PC model or updated BT device.
P5 pin may not generate interrupt sometimes for dev kit.	This issue will be addressed in next release. Currently there are no applications that use the P5 pin.
Sometimes unable to disconnect AVRC TG connection with client control	This issue is seen infrequently. Run the BT Spy tool and click the button again and verify using traces that the connection is disconnected.
On Linux, an error is seen in console when launching client control. Error - "QMetaObject::connectSlotsByName"	The console treats warnings as errors. The warning is harmless and can be ignored.
BLE_EnvironmentSensingTemperature builds with warning when compiled with Secure OTA compile option.	The warning is harmless can be ignored.
BLE conformance test MESH/CFGCL/CFG/MP/BV-01-C and MESH/CFGCL/CFG/CFGR/BV-01-C fails.	A change in needed to mesh client control application to pass the test and will be addressed in the next release.
When using ANS application, the UI does not allow to generate all possible alerts simultaneously.	This is current UI limitation, the application can handle generating alerts. The UI can generate individual alerts.
Unable to open multiple Client Control applications on Mac OS.	This is a limitation only if you use 2 or more boards on same Mac OS simultaneously and need to use Client Control for each app. In such a case, 2 different PCs will be needed.

### General Configurator Issues

Problem	Workaround
On macOS, when launching an external application from a Configurator and then switching focus back to the Configurator, it may not be fully focused. This can be seen by a lack of tooltips or proper cursor updates.	Click on the Configurator icon in the title bar or press Command+Tab to select the Configurator to fix the focus issue.
Code example <i>design.modus</i> files only support the CY8CPROTO-062-4343W, CY8CKIT-062-WIFI-BT, and CY8CKIT-062-BLE kits.	Manually modify the <i>design.modus</i> file for your target kit. This issue will be fixed in the next software release.

### Device Configurator

Problem	Workaround
For the CY8C6xx6, CY8C6xx7, CY8C6xx8, CY8C6xx9, and CY8C6xxA devices, the pin personality sets are not recommended at pin full drive strength. All pins except QSPI clock must limit the drive mode to one-quarter strength to limit noise. For the QSPI clock, pin the maximum drive mode can be one-half strength.	In the pin personality, manually decrease pin strength to one-quarter strength. If QSPI is used, decrease pin strength to one-half strength. For more details refer to the device datasheet.
In this release, the "12-bit SAR ADC" personality only supports connecting inputs to the dedicated SAR port. Other sources, such as Opamps or pins outside of the dedicated SAR port, are not permitted.	This will be updated in a future release.

### QSPI Configurator

Problem	Workaround
The QSPI Configurator cannot be used with CYBLE-416045-02 parts.	Select another part.

### Device Firmware Update Host Tool

Problem	Workaround
Closing the DFUH Tool on macOS causes a crash	Ignore the crash. There is no impact of the crash on the functionality of the tool.

## 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 the Cypress license agreement covering this software package. The applicable license agreements are available online:

<http://www.cypress.com/documentation/software-and-drivers/free-and-open-source-software-download-page>

## Further Reading

There are several related documents provided with ModusToolbox software. These documents include (but are not limited to):

- ModusToolbox Installation Guide
- Cypress Programmer 2.1 Release Notes
- PSoC 6 Software API Reference Guides
- Bluetooth API Documentation
- ModusToolbox IDE Quick Start Guide
- ModusToolbox IDE User Guide
- ModusToolbox Configurator Guides (for each Configurator)

Other documentation includes (but is not limited to):

- Device Datasheets
- Application Notes
- Training

[Contact your Cypress representative](#), as needed.

Cypress Semiconductor  
198 Champion Ct.  
San Jose, CA 95134-1709 USA  
[www.cypress.com](http://www.cypress.com)

© Cypress Semiconductor Corporation, 2017-2019. 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](http://cypress.com). Other names and brands may be claimed as property of their respective owners.