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

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

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

ModusToolbox® 2.2 Tools Package Release Notes

Production Release

Overview

ModusToolbox software is a set of tools that enable you to integrate Cypress 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 User Guide.

This release is an update to ModusToolbox 2.1 tools installation package. It does not replace the existing version; it installs alongside it. By default, the newer version of tools will be used. For more details about installation, refer to the ModusToolbox Installation Guide.

This document describes the features and known limitations for the ModusToolbox software provided as part of the ModusToolbox 2.2 tools package included with the installer.

Contents

What's Changed

This release of the ModusToolbox tools package includes the following updates and features:

Shared Library Management

Beginning with the ModusToolbox 2.2 release, we've developed a new way of structuring applications, called the MTB flow. Using this flow, applications can share Board Support Packages (BSPs) and libraries. If needed, different applications can use different versions of the same BSP/library. Sharing resources reduces the number of files on your computer and speeds up subsequent application creation time. Shared BSPs, libraries, and versions are located in a new "*mtb_shared*" directory adjacent to your application directories. You can easily switch a shared BSP or library to become local to a specific application, or back to being shared. Refer to the Library Manager User Guide for details.

Looking ahead, most example applications will use the new MTB flow. However, there are still various applications that use the previous flow, now called the LIB flow, and these applications generally do not share BSPs and libraries. ModusToolbox fully supports both flows.

Note When the BTSDK v2.8 is released, it will use the MTB flow.

Tools available from Windows Start menu

The Eclipse IDE, Configurators, and various tools can now be launched from Start menu.

Tools can now be patched

Instead of having to wait for a completely new tools package, Cypress can update individual Configurators and utilities to a new minor version to fix a defect or implement a small feature. Patching does not apply to the Eclipse IDE. When a patched tool is released, it will be available in a new version of the installer on the Cypress website.

Library Manager and Project Creator improvements

As part of the MTB flow, as well as to improve overall usability, the Library Manager and Project Creator tools have undergone substantial changes. Review the change section in each tool's user guide for the specific changes.

KitProg3 and OpenOCD Performance Improvements

KitProg3 has been updated to version 2.10 and it offers enhanced USB-I2C/SPI bridging implemented on bulk endpoints, which speeds-up USB communication. OpenOCD has also been updated to version 4.1 and provides fixes for various defects.

Enable SAR Connections to On-Chip Resources

In previous versions of ModusToolbox, the only supported connections were SAR port pins. As of this release, the SAR Sequencer block supports connection to all its routed connections (not all are routed on every device) in the Device Configurator. All valid SAR Sequencer connections are shown in the Analog Routing Viewer.

Update GCC compiler

GCC was updated to version 9.3.1. See GCC Update with Multiple Versions of ModusToolbox.

What's Included

This release includes the following tools and versions:

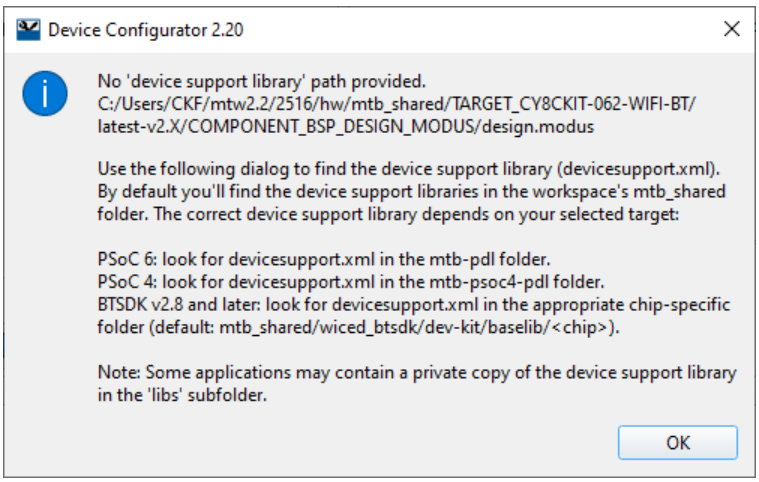
Tool Name	Version
Eclipse IDE for ModusToolbox	2.2.0
Bluetooth Configurator	2.20
CapSense® Configurator	3.10
CapSense Tuner	3.10
Configurator Backend CLI	2.20
Cy MCU Elf Tool	1.0
Device Configurator	2.20
Device Firmware Update (DFU) Host Tool	1.30
Firmware Loader	3.1.0 (KitProg3 2.10)
GCC	9.3.1
JRE	1.8.0_252
Library Manager	1.2.0
GNU make Build System	1.2
modus-shell	1.1.0
OpenOCD (Cypress-specific)	4.1.0
Project Creator	1.2.0
Proxy Helper	1.1.0
Power Estimator	1.2
Python (for Windows)	3.7.7
QSPI Configurator	2.20
Segment LCD Configurator	1.20
Smart I/O™ Configurator	2.20
USB Configurator	2.20

Design Impact

This section includes issues that might impact current designs.

Launching Device Configurator

When you launch the Device Configurator as a stand-alone tool without an association to the application, you will see the following dialog. This means you need to find the *devicesupport.xml* file. This file is located in the appropriate library directory as indicated on the dialog. Depending on your application, this directory can be in the *mtb_shared* directory or in your application's *libs* subdirectory.



You can avoid this by opening the Device Configurator from the Eclipse IDE, or by using the following command in your application directory:

```
make config
```

Updating ModusToolbox 2.1 Applications to Version 2.2

ModusToolbox 2.2 fully supports 2.1 designs, so you do not need to update your application. However, if you want to take advantage of the shared library feature, follow these steps:

1. Create a new 2.2 application using a similar BSP and code example to yours.
2. Open the Library Manager and add any needed libraries.
3. Copy the source code from the old application to the new one.

ModusToolbox 2.2.0 Requires Python 3.7

As part of this release, ModusToolbox 2.2.0 now requires Python 3.7.

- For **Windows**, Python 3.7 is installed in the *tools_2.2* directory, and the make build system has been configured to use it. You don't need to do anything if you use the modus-shell/Cygwin.bat file to run command line tools.

  However, if you plan to use your own version of Cygwin or some other type of bash, you will need to ensure your system is configured correctly to use Python 3.7. Use the `CY_PYTHON_PATH` as appropriate.

- For **macOS and Linux**, you must install and configure Python 3.7 manually.

GCC Update with Multiple Versions of ModusToolbox

With the 2.2 release of ModusToolbox, we have updated the GCC compiler that is shipped with the tools package to 9.3.1. In addition, we have renamed the directory that contains GCC in order to support more seamless upgrades in the future. As a result, some older applications may have leftover references to the old location, which was named *gcc-7.2.1*.

In those cases you will generally see an error that refers to a "missing gcc-7.2.1" when you open the application, try to build, or try to use one of the links in the Quick Panel. In most cases, clicking the "Generate Launches for ..." link will update your application so that everything will work. In rare cases, you will need to recreate your application using the 2.2 tools and then manually copy your source code from the old application to the new application.

Supported Tool Chains

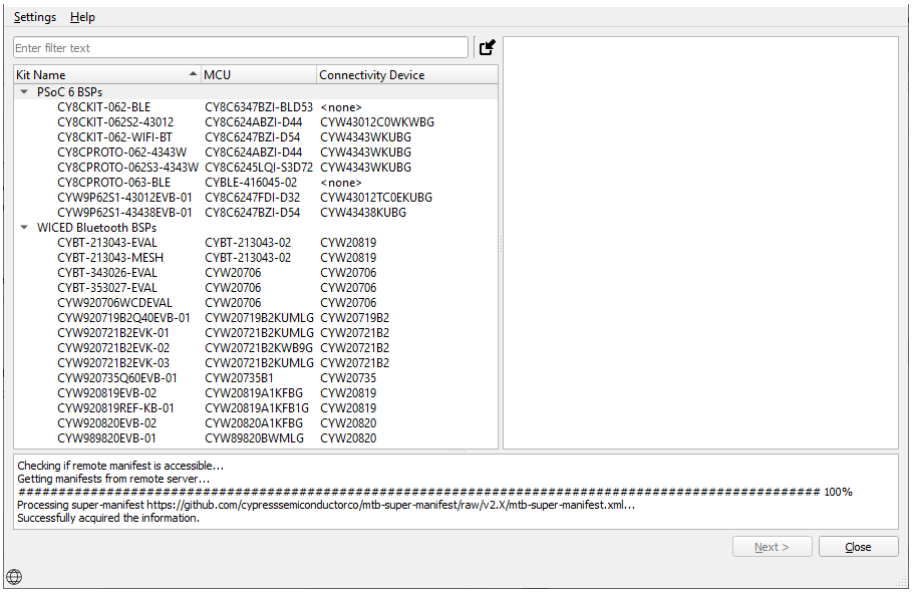
The GCC Arm Embedded toolchain GCC 9.3.1 is installed with the ModusToolbox software. This toolchain has no use restrictions and does not require license activation (it is distributed under the terms of the GNU Public License).

Although not installed with ModusToolbox software, the build system also supports these tool chains for PSoC 6 MCU applications:

- Arm compiler v6.11 (Windows and Linux hosts)
- IAR Embedded Workbench v8.32 (Windows only)

Supported Boards

The Cypress 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:



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

Known Issues/Limitations

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

Installation

Problem	Workaround
On common Linux distributions, the serial UART ports (usually /dev/ttySx or /dev/ttyUSBx devices) belong to the root user and to the dialout and plugdev groups. 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 or plugdev group. This can be done using the following command: `$sudo usermod -a -G dialout,plugdev $USER` **Note** For this command to take effect, the user must log out and then log back in.

Proxy

Problem	Workaround
When trying to create a new project, you may see the following error message: *Unable to open file at http://github.com/cypresssemiconductorco/mtb-super-manifest/raw/v2.X/mtb-super-manifest.xml.* Some boards and apps may be missing. Check the logfile for a detailed error message.	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 or use the Proxy Helper tool from the **Settings** menu. You can also find it in: <install_path>/ModusToolbox/tools_2.2/proxy-helper
In some cases, incorrect proxy settings in the Project Creator or Library Manager tool can prevent the proxy server settings from being edited to correct values.	Run the Proxy Helper tool from the **Settings** menu to reset the proxy mode to direct. The tool is also located in: <install_path>/ModusToolbox/tools_2.2/proxy-helper For example: `./proxy-helper --config set mode=direct`

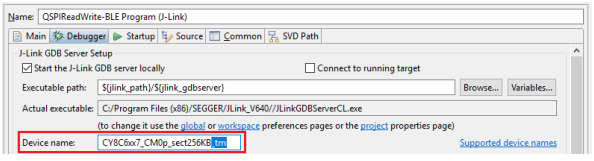
Building/Programming/Debugging

Problem	Workaround

When attempting to Program/Debug with Miniprog4 or J-Link probes in JTAG mode, the tool displays errors similar to the following:
`Info : KitProg3: Pipelined transfers enabled`
`Info : VTarget = 3.276 V`
`Info : clock speed 2000 kHz`
`Error: JTAG scan chain interrogation failed: all zeroes`
`Error: Check JTAG interface, timings, target power, etc.`
`Error: Trying to use configured scan chain anyway...`
`Error: psoc6.cpu: IR capture error; saw 0x00 not 0x01`
`Warn : Bypassing JTAG setup events due to errors`
`Error: Invalid ACK (0) in DAP response`
`Error: Invalid ACK (0) in DAP response`
`Error: Invalid ACK (0) in DAP response`
	These errors are likely caused by the device being in sleep mode. Open the Device Configurator, and set the **System Idle Power Mode** setting to 'Active' to turn off DeepSleep mode. JTAG is not available in sleep mode.
	Alternately, use SWD to acquire the target, and then switch to JTAG.

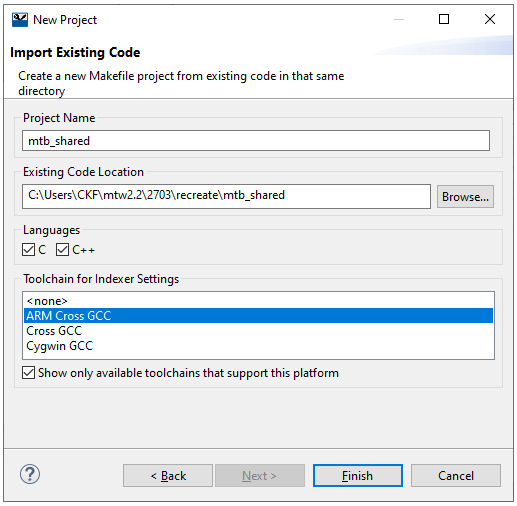
Problem	Workaround
If an application includes multiple BSPs that span major version numbers (for example, v1.x and v2.x) as local dependencies, there will be build conflicts. This is because of changes in the names of the BSP's dependencies.	There are two options: 1) The application can be updated to use the new MTB flow with the BSPs (and dependencies) marked as shared. 2) All local BSPs within a application must have the same major version number. Once the application is updated to only have a single major version of the BSP, any of the dependent libraries that were pulled down by the removed BSP version need to be removed from the application. This includes removing any *.lib* or *.mtb* files. The problematic dependencies that must be removed include the following: 1.x BSP dependencies: psoc6pdl, psoc6hal 2.x BSP dependencies: mtb-pdl, mtb-hal
KitProg3 and J-Link launch configurations do not work when you change the build configuration from Debug to Release or vice-versa. This applies to the Eclipse IDE for ModusToolbox and Visual Studio (VS) Code. Launch configurations contain the path to the executable, and that changes for each build configuration.	After you change from Debug to Release or vice-versa, click the "Generate Launches" command in the Quick Panel. You can alternately run the following command to create new launch configurations: • `make eclipse` for Eclipse IDE for ModusToolbox • `make vscode` for VS Code Be aware that this option overwrites any changes you may have made to the configurations.
J-Link launch configurations do not work for a PSoC 64-related project in VS Code.	Use KitProg3 launch configs. Starting from psoc6make v1.5.0, J-Link launch configurations are not available in VS Code for a PSoC 64-related project.
On macOS for a PSoC 64 project, using the Eclipse IDE for ModusToolbox, the "Attach" launch configuration fails every second time it is used: • for an RTOS-based project • if "Attach" is launched when the target is halted on breakpoint	Reset the device each time you run the "Attach" configuration.

For the Arm Cortex CM4 core, some code in main() executes before the debugger stops at start of main(). This means that some code executes twice; once before the debugger stops execution, and again after the debugger resets the program counter to the start of main() and you start debugging.	If you observe this issue, and it affects your application, put a delay loop at the start of main() to allow time for debugging subsystem initialization. The following code tests for the presence of the debugger before entering the delay loop:
	`int main(void)`
	`{`
	`/* If an active debugger is detected, give it some time to execute SYSRESETREQ`
	`* otherwise, the application runs before the debugger is fully operational */`
	`if (CoreDebug->DHCSR & CoreDebug_DHCSR_C_DEBUGEN_Msk)`
	`{`
	`Cy_SysLib_Delay(400u);`
	`}"`
	See KBA231071 for details.

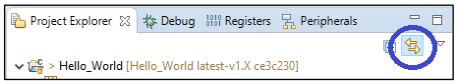
Problem	Workaround
Junk characters might be observed in a UART terminal during programming of the connected kit or after programming is completed.	Clear UART buffers after programming is completed.
Some kits with KitProg3 version 1.20 may hang after several rapid erase, program or read operations happening in sequence via openOCD. As a result, you might see a message like: LIBUSB_ERORR_TIMEOUT	You have two options: • Disconnect and reconnect the kit, or • Get latest update of the fw-loader from github.com/cypresssemiconductorco/Firmware-loader.
While running the “make getlibs" command or running the Library Manager or Project Creator tools, you may see an error similar to the following: ERROR: Commit/tag/branch "<Insert commit hash>" does not exist. : Check that the commit/tag/branch in the .lib file is valid. : If working offline, check that your offline content is up to date.	This occurs when a .lib file is pointing at a branch of the repo. Either replace it with a tag or a specific commit in this file. If you must use a branch, then run "make getlibs" without the cache enabled by setting the environment variable: CY_GETLIBS_NO_CACHE=1
On some occasions, the IDE will fail to run launch configurations with various errors, such as: "XXX has encountered a problem. Debug session already started. Terminate the first one before restarting"	Depending on the steps taken prior to launching a configuration, there are several reasons this may occur. The easiest way to resolve the issue is to restart Eclipse.
There is a programming error for Cypress platforms that connect via FTDI on macOS Catalina. The boards include: • CYW920819EVB-02 • CYW920820EVB-02 • CYW920719B2Q40EVB-01 • CYW920721B2EVK-01 • CYW920721B2EVK-02 • CYW920721B2EVK-03 • CYW989820EVB-01 • CYW920706WCDEVAL • CYW920735Q60EVB-01	This only happens in macOS Catalina 10.15.5 because of a serial port detection error. The macOS Catalina FTDI driver is missing the necessary device identification information. To resolve this issue, update to macOS Catalina 10.15.6.
A “JTAG-DP STICKY ERROR” message may appear in the IDE when connected via the JTAG interface of a MiniProg4 probe in CMSIS-DAP HID mode.	You can safely ignore this error, or switch the MiniProg4 to CMSIS-DAP bulk mode.
JTAG performance on MiniProg4 may be significantly slower than SWD.	There is no workaround except using the SWD interface if JTAG performance is not acceptable.
You must manually reset after programming PSoC 6 kits when using GDB SEGGER + Jlink + JTAG interface.	Update each of the following Launch Configurations under the **Debugger** tab. In the Device Name field, delete the "_tm" suffix.  • "<app-name> Erase (JLink)" • "<app-name> Program (JLink)"

Problem	Workaround
In VS Code, the J-Link launch configuration does not stop at Reset_Handler after 'Restart', but at some other line of code. This is associated with any "monitor" command in GDB. For example, if you press the Restart button, the Cortex-Debug plugin sends "monitor halt" and "monitor reset" commands to the GDB causing GDB to go out-of-sync with the target. GDB does not know what the "monitor xxx" command does. It does not expect that "monitor reset" will change the state of the target.	Step through code to enter the Reset_Handler.
Starting from KitProg3 v2.10, when KitProg3 is in CMSIS-DAP Bulk mode, it is not possible to debug and use USB-I2C/SPI bridging (for example, in the CapSense Tuner, Bridge Control Panel) at the same time. This affects Windows OS only. It does not affect Linux or macOS users.	If you would like to use debug and USB-I2C/SPI bridging at the same time, there are two possible workarounds: • If performance for programming and debug is not critical, switch KitProg3 to **CMSIS-DAP HID** mode via the fw-loader utility. Firmware Loader is installed with ModusToolbox software, and is available separately on GitHub. • If you need faster performance for programming and debug, use the onboard KitProg3 for programming purposes and MiniProg4 for bridging purposes or vice versa. Both devices can be in **CMSIS-DAP bulk** mode. Details are in KBA231025.
Starting from KitProg3 v2.10, in some cases Windows 7 does not recognize the KitProg3 bridge. So the USB-I2C/SPI bridge devices are not available in either CMSIS-DAP HID or CMSIS-DAP bulk mode.	Install a digitally signed driver manually from the Windows Update Catalog. Follow steps from KBA231026.
In Linux OS, with KitProg3 in CMSIS-DAP HID mode, a debug session in ModusToolbox can be destroyed if you use the Firmware Loader `--device-list` command while debugging. This is limitation of hidapi library used on Linux. MacOS and Windows OSs are not impacted.	If you have a debug session running, don't use the firmware loader tool.
KitProg3 v2.10 is installed as part of the ModusToolbox 2.2 tools package. This version of KitProg3 will not work with PSoC Creator 4.3 or PSoC Programmer 3.28.7.	If you updated your kit to KitProg3 v2.10 and wish to use the kit with PSoC Creator 4.3 and PSoC Programmer 3.28.7, get a previous version of fw-loader (with an earlier version of KitProg3) and update the kit. The fw-loader tool is available here: github.com/cypresssemiconductorco/Firmware-loader If this is not urgent, you can wait for a newer PSoC Programmer version with support for KitProg3 v2.10, expected soon.

Eclipse IDE

Problem	Workaround
When opening the Eclipse IDE for ModusToolbox using macOS version 11.0.1 (Big Sur), you may encounter a java.lang.NullPointerException error message, which indicates that the IDE cannot run due to a Java conflict between Eclipse and macOS.	If you are using macOS 11.0.1 (Big Sur), do not use the Eclipse IDE. You can use various tools such as the Project Creator, Library Manager, and Configurators in stand-alone mode. You can also export your application to various IDEs/code editors, such as Visual Studio Code. Refer to the ModusToolbox User Guide for details.
If you delete the shared library directory (named *mtb_shared*, by default) from disk and then regenerate it using `make getlibs` or the Library Manager, the directory will not be restored properly for use in the Eclipse IDE. This is because several files required by the Eclipse IDE are not restored as they were when the application was created. **Note** You can delete the *mtb_shared* directory at any time because it can be recreated. You might do this when sharing the application, for example. The shared library directory only contains files that are already controlled and versioned, so you should NOT check it into a revision control system.	After regenerating the *mtb_shared* directory and assorted libraries, open the Eclipse IDE and follow these steps: 1. Delete the mtb_shared folder shown in the IDE Project Explorer. Do **NOT** select the check box "Delete project contents on disk" (if you do, you will have to regenerate it again). 2. Select **File > Import > C/C++ > Existing Code as Makefile Project** and click **Next >**.  a. Under **Existing Code Location**, click **Browse…**, navigate to the application's root directory, select the mtb_shared folder, and click **Select Folder**. b. Under **Toolchain for Indexer Settings**, select **ARM Cross GCC**. c. Click **Finish**. 3. After the import completes, build the application.

If your application includes a shared libraries directory, such as mtb_shared, when you use the **Build All** command you may see error such as:
`**** Build of configuration Debug for project mtb_shared ****`
`make: *** No rule to make target 'all'.  Stop.`
`Build Failed. 1 errors, 0 warnings.`
	Make sure that the build failure is only for the shared directory. If so, you can safely ignore it. This occurs because the shared directory is included in the workspace, but it is not intended to be built.

Problem	Workaround
Switching from a file in one project to a file in another project might cause a delay before showing it in the File Editor/Viewer. 	Click the **Link with Editor** button in the Project Explorer to toggle the feature off.  The "Link with Editor" feature always shows the selected file from the File Editor/Viewer in the Project Explorer. By turning the feature off, Eclipse stops trying to show the selected file, and thus eliminates the delay. You can see the selected project/file in the Eclipse IDE Title Bar.
For projects that use external resources located outside of the application folder root (such as mtb_shared, wiced_btsdk, or 3rd-party libraries), some IDE code browsing and analysis features (such as resolving includes or opening object declarations for objects and headers located in those external resources) may not be usable or show unresolved includes immediately after project creation.	These will be resolved after building the project, as the IDE parses the output messages from the build to find paths to those resources.
On macOS Catalina, some IDE GUI windows/elements do not display correctly in some cases.	Resize the window or scroll to fix the display problem. In some cases, use the Refresh option. Cypress is investigating these issues and will address them in a future release.

Some applications when imported into the Eclipse IDE from Mbed OS fail to build with an error, such as:
cc1.exe fatal error: .mbed_config.h: No such file or directory
	This happens because the application is generating a path to the *mbed_config.h* header file that make build system cannot find. Fix this by removing the relative path '.\' in the makefile. Change this:
	`ASM_FLAGS += -include`
	`ASM_FLAGS += .\mbed_config.h`
	
	To this:
	`ASM_FLAGS += -include`
	`ASM_FLAGS += mbed_config.h`

On Windows, when building a project and/or programming the device, the IDE reports one or more errors similar to the following: *** fatal error - cygheap base mismatch detected - 0x18032C408/0x18032D408	This occurs because you likely have multiple versions of Cygwin in your build environment path. The IDE uses a version of Cygwin in the ModusToolbox installation directory. Remove the instance of "C:\cygwin64\bin" from your path.
Markdown (*.md) files do not render correctly in the IDE. For example, tables do not show rows and columns. Also, the IDE may show an error for the file such as: "Cannot resolve element with id 'figure-1'".	This is a known issue with viewing markdown files in Eclipse. Various rendering errors can be safely ignored. Cypress recommends using an external editor, such as Visual Studio Code or Typora to view markdown files.
When creating a new application, the IDE attempts to open any found readme.md files. Occasionally (and randomly), some of these files are opened in an external text editor instead of in the IDE.	This appears to be an Eclipse bug with no workaround. However, since *.md files do not render well in Eclipse (as noted above), you should use an external editor, such as Visual Studio Code or Typora, and set that editor as the default.
Sometimes, the Eclipse "egit" plugin locks directories. This prevents the Library Manager from removing BSPs/libraries from these locked directories. When this happens, you will see an error message in the Library Manager console indicating that permission is denied for removing a particular directory.	Before running the Library Manager from the Eclipse IDE, right-click on the project and select **Team > Disconnect**. When you are done with the Library Manager, go to **Team > Share Project** and select the correct project to reconnect. **Note** Various projects may be set up differently, and the process to use the Team options will vary as well.

Problem	Workaround

The IDE **Project >Build Configurations > Active** menu item to set Debug or Release mode is non-functional. This is due to the project's Makefile.	To set Debug or Release mode, edit the project's Makefile, which contains the following:
	`# Default build configuration. Options include:`
	`#`
	`# Debug -- build with minimal optimizations, focus on debugging.`
	`# Release -- build with full optimizations`
	`CONFIG=Debug`

If you include external folders/files in your application, the Eclipse IDE will occasionally unselect your application project in the Project Explorer. However, it will leave the **Launch** section populated with links in the Quick Panel. Using those Launch links may result in a reported error of "Could not resolve cy_prj_path."	To resolve this, click on the appropriate project for your application in the Project Explorer, and then click on the **Launch** link again.

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: https://www.cypress.com/products/modustoolbox-software-environment

Project Creator

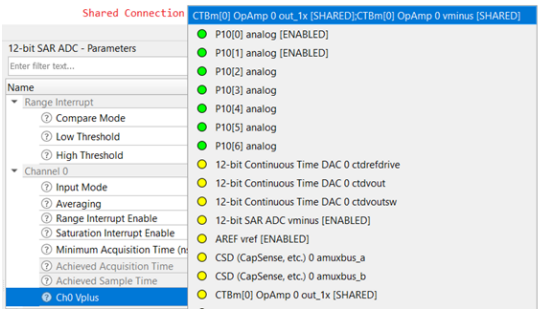
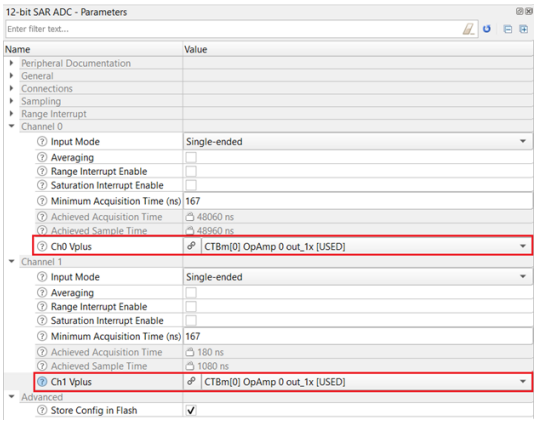
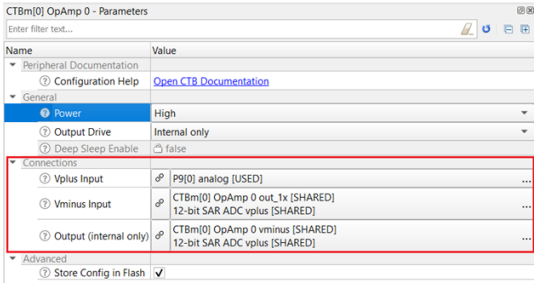
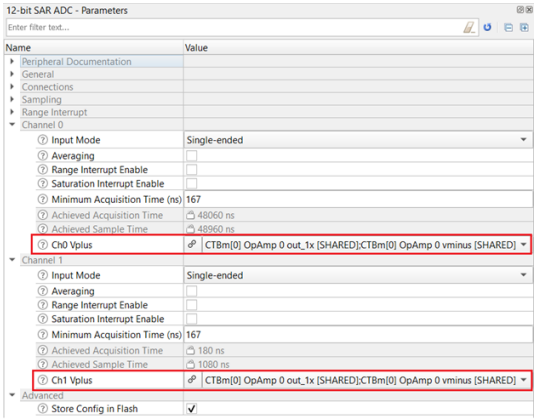
Problem	Workaround
When using the Project Creator for BTDSK devices and the **Import** feature, some imported applications may not be configured correctly. If you select a BSP that is not already included with the imported application files, the Project Creator tool generates a new *.mtb* file; however, the "location" field of the *.mtb* file does not contain any "prefix" data. This results in the BSP being created in a non-standard location.	To create a BTSDK BSP in the standard location, do the following after creating the imported application: 1. Open the Library Manager for the application. 2. Deselect the **Shared** check box for the applicable BSP, and then click **Update**. 3. After the update has completed, reselect the **Shared** check box for the applicable BSP, and then click **Update** again. 4. When the update completes, close the Library Manager. 5. Open a File Explorer tool on your computer, and navigate to the *mtb_shared* directory adjacent to the application directory. 6. Delete the entire top-level BSP directory (*mtb_shared/TARGET_<BTSDK-BSP-NAME>*). **Note** The Library Manager does not delete shared BSPs, so after the **Update** operations there will be two BSPs. The new standard BTSDK BSP is generated in *mtb_shared/wiced_btsdk/dev-kit/bsp/* by default.

Problem	Workaround
When setting the proxy information in the Project Creator, the tool does not refresh and find BSPs at the specified manifest.	To refresh the Project Creator tool with the new proxy settings, close it and reopen it. This will be addressed in a future release.
When trying to create a new project you may see the following error message: Unable to open file at http://github.com/cypresssemiconductorco/mtb-super-manifest/raw/v2.X/mtb-super-manifest-fv2.xml Some boards and apps may be missing. Check the console for a detailed error message.	Verify that you have a network connection and see the instructions in the Installation problem section above concerning proxy settings.

Library Manager

Problem	Workaround
After running the Library Manager, the files shown in the Eclipse IDE Project Explorer are not updated.	To see the updated files, click on the project in the IDE Project Explorer and press [F5] to refresh the Eclipse view.
Opening multiple instances of the Library Manager at the same time could cause confusion and indeterminate states in applications.	Do not open multiple instances of the Library Manager. This issue will be addressed in a later release.
Sometimes when you click **Update** in the Library Manager, the tool will begin to process your request and then appear to freeze.	Terminate the Library Manager and relaunch it. This issue will be addressed in a future release.
When you remove a library item, the Library Manager deletes the associated directory from the libraries directory (typically this is called "libs"). On Windows, if there are any processes that have a lock on that directory, or a file in that directory, the directory removal will not work completely. The Library Manager will remove most of the directory contents and will also mark the library item as removed.	To remove the library completely, you must release the lock on the folder or file, and then manually delete the directory. The steps to release the lock depend entirely on the process that is holding the lock. Common scenarios include: • A command-line prompt is in that directory. In this case, "cd" to a different directory. • A text editor has a file from that directory open. In this case, close the file in the text editor. Depending on the text editor, you may have to exit the entire text editor. • The Eclipse git "egit" plugin has a lock on the folder. In this case, exit and restart the ModusToolbox Eclipse IDE. In all cases, once the lock is removed you must remove the associated directory from the libraries directory before the Library Manager will be able to work with that particular library again.

Device Configurator

Problem	Workaround
SAR ADC Vplus parameter for multiple channels cannot be configured to the same shared connection in the user interface directly (even though there is no HW limitation that prevents it). For example, if the opamp is configured as follower, wherein, Vminus of opamp is connected to its output, then the shared connection (which involves Vminus and output) is not available to select for the Vplus parameter of more than one SAR channel. In the following image, the shared connection appears for one of the channels. Yet once selected, it will not appear for rest of the channels. 	First configure the Vplus parameter of the selected SAR channels and then configure the resource providing the input to SAR Vplus. In the example of Opamp as follower, configure Vplus of the selected SAR channels as opamp out 1x or 10x.  Then configure the opamp parameters. In the opamp parameters UI, there is a '...' button that opens a dialog to select multiple items at once forming a shared connection.  Once the opamp connections are completed, the SAR Vplus parameter of the selected channels are automatically updated to the shared connection. 

Problem	Workaround
When connecting a pin in a port other than the preferred SAR port to more than one ADC channel, the router may fail to find a solution.	Use a pin in the preferred SAR port when connection to multiple ADC channels is required.
The analog router may route across pins that are not intended to be routed (either because they are digital inputs, or because the project routes them in software). This can cause signal corruption or other connection problems.	Use the Analog Route Editor to manually route around such pins, if necessary.

CapSense Tuner

Problem	Workaround
On macOS, UART communication can fail in the CapSense Tuner with a baud rate over 115200. This occurs because the maximum baud rate supported in macOS is 230400 (defined in the *termios.h* system file), and next baud rate after 115200 in the CapSense Tuner is 250000. So, baud rates from 250000 to 4000000 cannot be used on macOS.	On macOS, use a baud rate of 115200 for the CapSense Tuner.
If using the UART communication interface, a low packet-transferring rate may cause the CapSense Tuner to disconnect due to a data-reading timeout error.	You can resolve this using either of these solutions: • Increase the UART communication baud rate. • Increase the value of the `uartSingleReadTimeout` parameter available in the tuner INI file. You can find the tuner INI file as follows: • Windows: *<user_home>/AppData/Roaming/Cypress Semiconductor Corporation/CapSense Tuner.ini* • Linux: */home/<user_home>/.config/Cypress Semiconductor Corporation/CapSense Tuner.ini* • macOS: /Users/<user_home>/.config/Cypress *Semiconductor Corporation/CapSense Tuner.ini*

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
- ModusToolbox User Guide
- Cypress Programmer Release Notes

Other documentation includes (but is not limited to):

- Device Datasheets
- Application Notes
- Training

Contact your Cypress representative, as needed.

Cypress Semiconductor
An Infineon Technologies Company
198 Champion Ct.
San Jose, CA 95134-1709 USA
www.cypress.com
www.infineon.com