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.

KitProg3 User Guide

Document Number: 002-24616 Rev. *N

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

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. No computing device can be absolutely secure. Therefore, despite security measures implemented in Cypress hardware or software products, Cypress shall have no liability arising out of any security breach, such as unauthorized access to or use of a Cypress product. CYPRESS DOES NOT REPRESENT, WARRANT, OR GUARANTEE THAT CYPRESS PRODUCTS, OR SYSTEMS CREATED USING CYPRESS PRODUCTS, WILL BE FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION (collectively, “Security Breach”). Cypress disclaims any liability relating to any Security Breach, and you shall and hereby do release Cypress from any claim, damage, or other liability arising from any Security Breach. In addition, the products described in these materials may contain design defects or errors known as errata which may cause the product to deviate from published specifications. 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. “High-Risk Device” means any device or system whose failure could cause personal injury, death, or property damage. Examples of High-Risk Devices are weapons, nuclear installations, surgical implants, and other medical devices. “Critical Component” means any component of a High-Risk Device whose failure to perform can be reasonably expected to cause, directly or indirectly, the failure of the High-Risk Device, 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 any use of a Cypress product as a Critical Component in a High-Risk Device. You shall indemnify and hold Cypress, its directors, officers, employees, agents, affiliates, distributors, and assigns harmless from and against all claims, costs, damages, and expenses, arising out of any claim, including claims for product liability, personal injury or death, or property damage arising from any use of a Cypress product as a Critical Component in a High-Risk Device. Cypress products are not intended or authorized for use as a Critical Component in any High-Risk Device except to the limited extent that (i) Cypress’s published data sheet for the product explicitly states Cypress has qualified the product for use in a specific High-Risk Device, or (ii) Cypress has given you advance written authorization to use the product as a Critical Component in the specific High-Risk Device and you have signed a separate indemnification agreement.

Contents

1 Introduction

KitProg3 is the Cypress low-level communication firmware for programming and debugging. It provides communication between a programming tool (such as Cypress Programmer or PSoC® Programmer™) and a target, such as a PSoC 6 MCU. KitProg3 supports a variety of Cypress development kits. It is also the communication firmware found in the MiniProg4 debug probe.

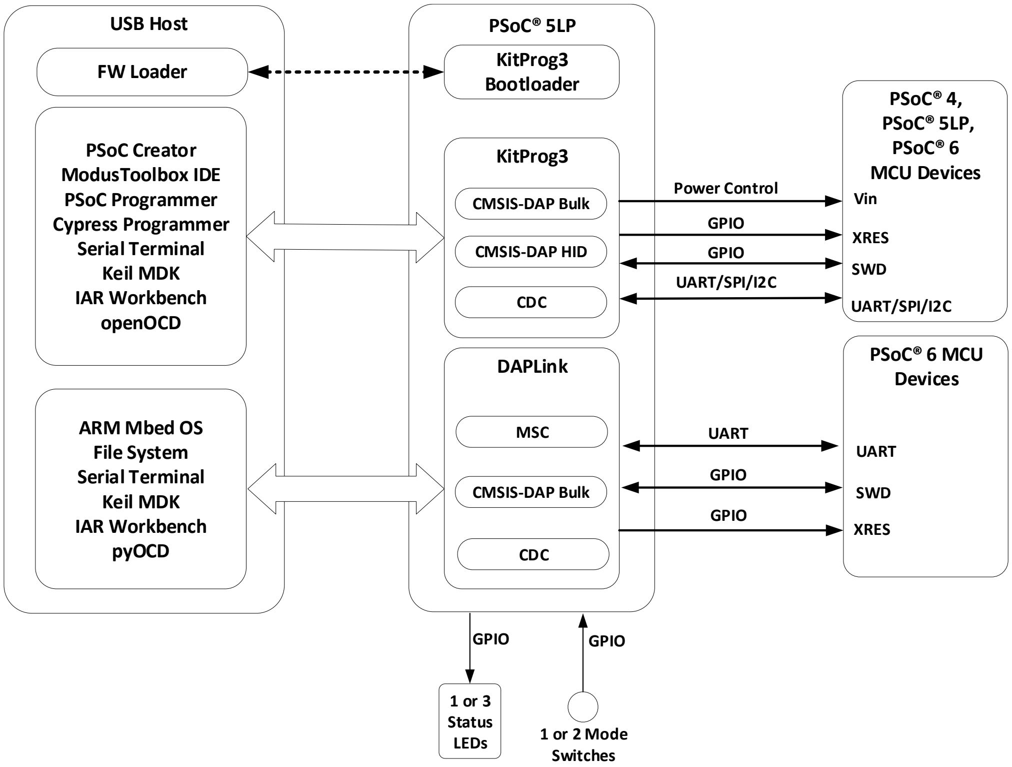
Cypress development kits have KitProg firmware installed to provide the necessary communication between the host and target. As a result, when you plug the kit into your host computer, programming and debugging just work.

KitProg3 uses the industry-standard Serial Wire Debug (SWD) and JTAG protocols. It uses CMSIS-DAP V2.0.0 and V1.2.0 as the Bulk and HID endpoints transport mechanisms. CMSIS-DAP is also an industry standard. KitProg3 implements USB Bulk endpoints for faster communication. It also supports HID endpoints for use cases that require them, but communication is slower. Out of the box, KitProg3 uses Bulk endpoints.

KitProg3 also supports bridging: USB-UART, USB-I2C, USB-SPI, and GPIO Bridging.

The KitProg3 package also includes the Arm® Mbed™ DAPLink that enables programming and debugging applications (IoT) for Arm Cortex® CPUs (PSoC 6 MCUs only). DAPLink is platform-independent and provides Drag-and-Drop programming via a Mass Storage Controller (MSC), CMSIS-DAP debugging (Bulk endpoints), and a virtual serial port via USB Communications Device Class (CDC). You can switch between KitProg3 and DAPLink with a simple push of a Mode button.

Figure 1-1. KitProg3 High-Level Architecture



1.1 What’s in this Guide

This user guide provides comprehensive information about KitProg3 in PSoC development kits:

- Section 2: Installing and Using KitProg3 – provides all the information you need to get up and running for the common use cases.
- Section 3: KitProg3 Design – includes full details about KitProg3 User interface, mode switching and status LEDs behavior.
- Section 4: DAPLink Mode – includes details on how to upgrade kit firmware to KitProg3 with DAPLink, how to switch to DAPLink mode, and useful references to Arm resources.
- Section 5: KitProg3 vs KitProg2 – includes details about how to tell what’s installed, the differences, and how to upgrade a kit to KitProg3.
- Section 6: Updating KitProg3 Firmware – includes details on how to update KitProg3 firmware, how to use Firmware Loader to update, downgrade firmware and switch modes.

In case of any issues, see the Troubleshooting section.

1.2 KitProg3 Tools Support and Compatibility

KitProg3 is supported by the following tool combinations:

- KitProg3 CMSIS-DAP Bulk and HID modes
  - ModusToolbox and Cypress Programmer
  - PSoC Creator and PSoC Programmer (v 3.28 or later)
  - openOCD CLI
  - uVision
  - IAR Embedded Workbench
  - Visual Studio Code
- DAPLink mode
  - ModusToolbox
  - Mbed CLI
  - Mbed Online IDE
  - Mbed Studio
  - pyOCD
  - µVision
  - IAR Embedded Workbench
  - Visual Studio Code

Table 1-1. KitProg Compatibility

KitProg	IDE	Programmer	Bridging Tools
KitProg3	ModusToolbox PSoC Creator	Cypress Programmer PSoC Programmer	Bridge Control Panel (PSoC Programmer) * CapSense® Tuner (PSoC Creator and ModusToolbox)
KitProg2	PSoC Creator	PSoC Programmer	

* Note that the Bridge Control Panel is not supported by Cypress Programmer or ModusToolbox.

Table 1-2. KitProg Modes

Mode	USB devices	Features
KitProg3 mode	CMSIS-DAP Bulk CDC UART CMSIS-DAP HID Bridge Bulk	CMSIS-DAP Programming/Debugging I2C/SPI/UART/GPIO Bridging Voltage control
DAPLink mode	Mass Storage Device CDC UART CMSIS-DAP Bulk	CMSIS-DAP Programming/Debugging UART Bridging Drag-And-Drop Programming

2 Installing and Using KitProg3

To use KitProg3, you need one or more of the following tools:

- ModusToolbox and Cypress Programmer
- PSoC Creator and PSoC Programmer
- Any IDE that supports CMSIS-DAP Bulk protocol
- Bridge Control Panel (BCP) from Cypress for USB-I2C and USB-SPI bridging
- A terminal emulator for USB-UART bridging
- A terminal that supports HID/Bulk USB communicational protocol
- A supported kit (see Supported Kits)

KitProg3 is communication firmware used by these tools. Bridge Control Panel is installed with PSoC Programmer. PSoC Programmer versions before 3.28.x do not support KitProg3.

2.1 Installing KitProg3

Install ModusToolbox, Cypress Programmer, PSoC Creator, or PSoC Programmer before using any kit with KitProg3. Any required driver is installed by the tools that use KitProg3. There is no separate installer for KitProg3.

You can also get the latest version of KitProg3 delivered with the Firmware Loader available at the Cypress GitHub repository. The Firmware Loader does not install any drivers, but you can use it to upgrade (or downgrade) the KitProg firmware on a kit.

The supported Cypress kits have either KitProg3 or KitProg2 already installed. See Upgrading to KitProg3 to learn how to tell what's installed, and how to upgrade.

When you plug in a kit, depending upon your circumstances and host operating system, you may see a message that drivers are being installed.

KitProg3 enumerates as a root USB Composite Device with subordinate CMSIS-DAP, Bridge, and USB-UART interfaces.

2.2 Using KitProg3

You do not use KitProg3 directly. You use a programming tool or IDE that automatically connects to and uses KitProg3. In most cases, KitProg3 is completely transparent.

2.2.1 Connecting

Plug in the kit. Use the USB cable that came with the kit and connect the host computer to the kit. KitProg3 is powered via the USB cable.

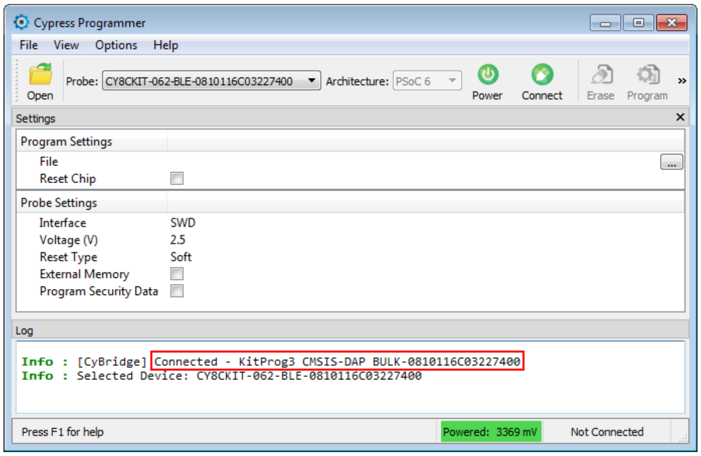
When you plug in the kit, an amber status LED indicates the current mode. On most kits, this is LED2. On MiniProg4, it is labeled **Mode**. The precise designation varies by kit.

If the LED is steady (out of box behavior), KitProg3 is using Bulk endpoints for faster communication. If the LED is ramping at 2 Hz, KitProg3 is in DAPLink mode. If the LED is ramping 1 Hz, KitProg3 is using HID endpoints, which means slower communication.

KitProg3 from the factory defaults to Bulk endpoints because they are faster. You can switch KitProg3 between Bulk and HID endpoints should you need to. See Mode Switching and Command-Line Options.

Launch your programming tool or the Eclipse IDE for ModusToolbox. The tool connects to KitProg3 automatically. The KitProg3 connection appears in the UI of the programming tool. Note that at this time, some Cypress programming tools do not recognize the kit if it is in DAPLink mode. Switch the kit to Bulk or HID, and the tool can see and work with the kit. See Mode Switching and Command-Line Options.

Figure 2-1. Connected via Cypress Programmer



Probe: CY8CKIT-062-BLE-0810116C03227400 Architecture: PSoC 6

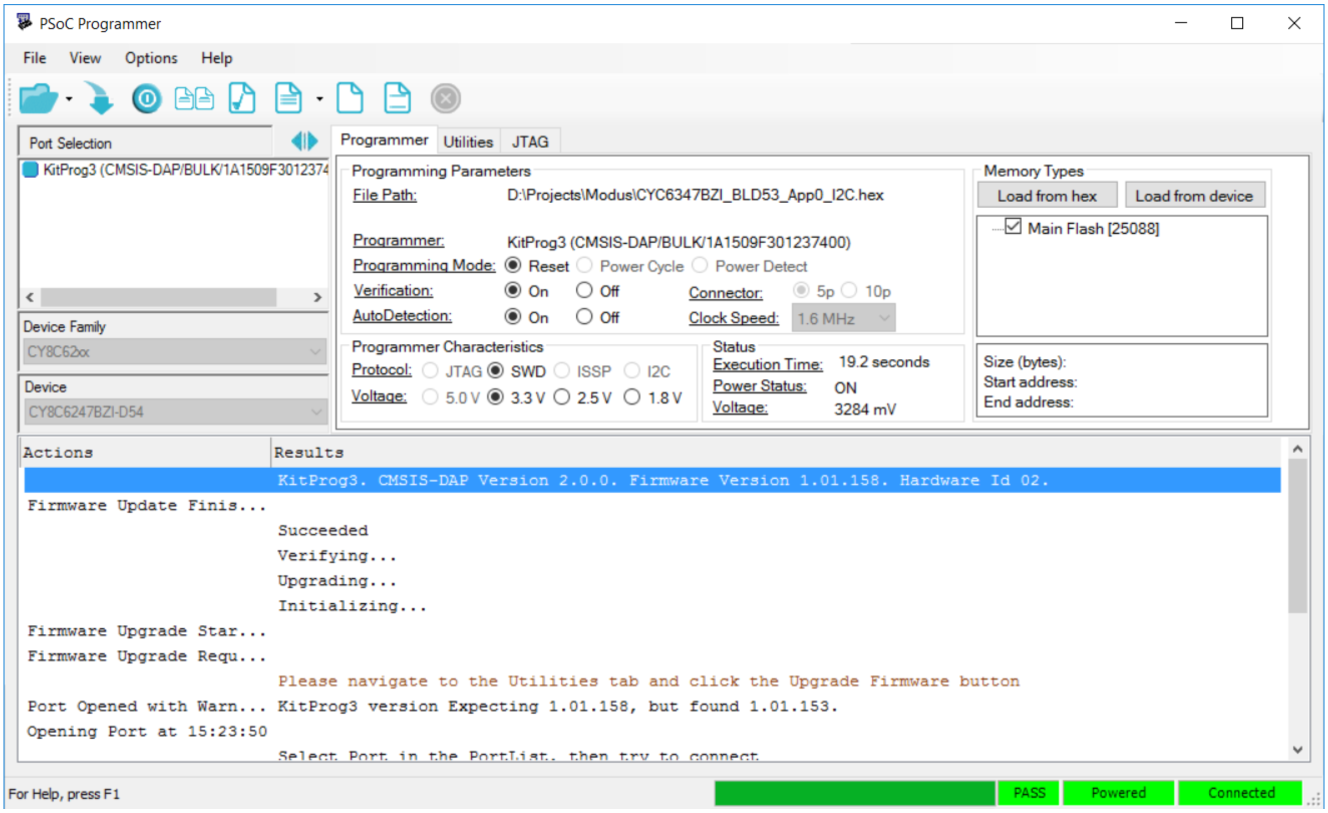
Program Settings
File
Reset Chip □

Probe Settings
Interface SWD
Voltage (V) 2.5
Reset Type Soft
External Memory □
Program Security Data □

```
Info : [CyBridge] Connected - KitProg3 CMSIS-DAP BULK-0810116C03227400
Info : Selected Device: CY8CKIT-062-BLE-0810116C03227400
```

Press F1 for help Powered: 3369 mV Not Connected

Figure 2-2. Connected via PSoC Programmer



PSoC Programmer

File View Options Help

Port Selection

KitProg3 (CMSIS-DAP/BULK/1A1509F3012374

Device Family: CY8C62xx

Device: CY8C6247BZI-D54

Programmer | Utilities | JTAG

Programming Parameters

File Path: D:\Projects\Modus\CYC6347BZI_BLD53_App0_I2C.hex

Programmer: KitProg3 (CMSIS-DAP/BULK/1A1509F301237400)

Programming Mode: ● Reset ○ Power Cycle ○ Power Detect

Verification: ● On ○ Off

AutoDetection: ● On ○ Off

Connector: ● 5p ○ 10p

Clock Speed: 1.6 MHz

Memory Types

Load from hex | Load from device

☑ Main Flash [25088]

Programmer Characteristics

Protocol: ○ JTAG ● SWD ○ ISSP ○ I2C

Voltage: ○ 5.0 V ● 3.3 V ○ 2.5 V ○ 1.8 V

Status

Execution Time: 19.2 seconds

Power Status: ON

Voltage: 3284 mV

Size (bytes):

Start address:

End address:

```
Actions                       Results
                              KitProg3. CMSIS-DAP Version 2.0.0. Firmware Version 1.01.158. Hardware Id 02.
Firmware Update Finis...
                              Succeeded
                              Verifying...
                              Upgrading...
                              Initializing...
Firmware Upgrade Star...
Firmware Upgrade Requ...
                              Please navigate to the Utilities tab and click the Upgrade Firmware button
Port Opened with Warn...      KitProg3 version Expecting 1.01.158, but found 1.01.153.
Opening Port at 15:23:50
                              Select Port in the PortList, then try to connect
```

For Help, press F1

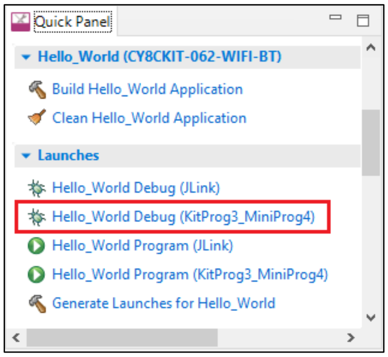
PASS | Powered | Connected

2.2.2 Programming and Debugging

Use the program or debug commands in your tool. See your IDE’s documentation for details.

For example, for the Eclipse IDE, click the **Debug (KitProg3_MiniProg4)** link in the **Quick Panel**. To program the device without debugging, use the **Program (KitProg3_MiniProg4)** link.

Figure 2-3. Launching a Debug Session in Eclipse IDE for ModusToolbox



Quick Panel

Hello_World (CY8CKIT-062-WIFI-BT)

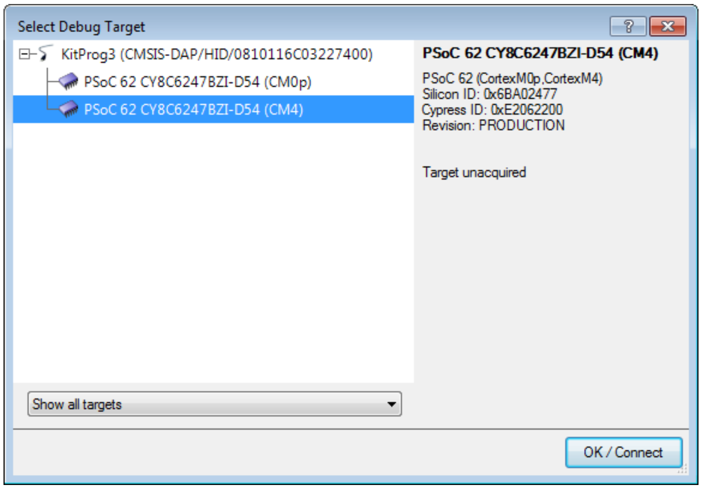
- Build Hello_World Application
- Clean Hello_World Application

Launches

- Hello_World Debug (JLink)
- Hello_World Debug (KitProg3_MiniProg4)
- Hello_World Program (JLink)
- Hello_World Program (KitProg3_MiniProg4)
- Generate Launches for Hello_World

For PSoC Creator, use the **Debug** or **Program** commands in the **Debug** menu**.** Then select your target and click **OK**.

Figure 2-4. Selecting the Debug Target in PSoC Creator



Select Debug Target

KitProg3 (CMSIS-DAP/HID/0810116C03227400)
- PSoC 62 CY8C6247BZI-D54 (CM0p)
- PSoC 62 CY8C6247BZI-D54 (CM4)

PSoC 62 CY8C6247BZI-D54 (CM4)

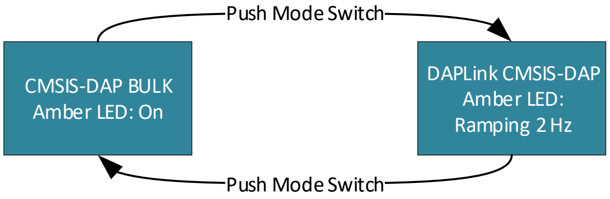
PSoC 62 (CortexM0p,CortexM4)
Silicon ID: 0x6BA02477
Cypress ID: 0xE2062200
Revision: PRODUCTION

Target unacquired

2.2.3 Mode Switching

KitProg3 v1.20 and later supports CMSIS-DAP Bulk by default, and it may support Arm DAPLink mode. Each supported kit has a **Mode Select** switch (mode switch). Push the switch to cycle through the two modes: CMSIS-DAP Bulk and DAPLink. KitProg3 also supports CMSIS-DAP HID mode, but only through the Firmware Loader tool. See Command-Line Options.

Figure 2-5. Mode switching in KitProg3



Note If a kit does not support DAPLink mode, mode switch will have no effect. See DAPLink Mode for supported kits.

The precise designation for the mode switch varies based on the kit. For example, on some Cypress Pioneer kits and the MiniProg4 debug probe, it is labeled **Mode Select**. Use your kit documentation if you can't find the switch. Starting from KitProg3 v1.11, the **Custom App** switch is deprecated and does not perform any action.

Switch to HID mode only if needed for your design or hardware. Otherwise, stay in Bulk mode for better performance during programming and debugging operations. Use DAPLink mode when required by your development workflow (for example, Arm Mbed development).

When in Bulk mode, the amber LED is ON and steady. In DAPLink mode the LED ramps at 2 Hz frequency. When in HID mode, the amber LED ramps up and down at 1 Hz frequency. In Bulk and HID modes, bridging (USB-I2C, USB-SPI, or USB-UART) is available while debugging with one exception: starting from KitProg3 v2.10, USB-I2C and USB-SPI bridging and debugging are mutually exclusive for Windows OS in CMSIS-DAP Bulk mode – see Troubleshooting for details. See KitProg3 LEDs for information on how KitProg uses the LEDs.

2.2.3.1 *UARTx2 Mode*

Some kits support a special operating mode that allows for two UART connections, rather than a single UART plus bridging (for example, USB-I2C or USB-SPI).

You cannot enter UARTx2 if the kit is in DAPLink mode. When the kit is in CMSIS-DAP Bulk or HID mode, press and hold the mode switch for at least two seconds. In UARTx2 mode, the amber LED blinks 1 second at 2 Hz then stays on for another second. To exit, press and hold the mode switch for at least two seconds. You return to CMSIS-DAP Bulk mode.

2.2.3.2 *In and out of Bootloader Mode*

If your KitProg3 image is corrupted, you can use bootloader mode to update the firmware. See Installing KitProg3 for information on where to get the KitProg3 image.

- To get into Bootloader mode, press the mode switch while plugging in the board.
- To get out of Bootloader mode and return to normal operation, unplug the kit and reconnect without pressing the mode switch button.

2.2.4 KitProg3 LEDs

The KitProg3 user interface is limited to one or two mode switches and one or three status LEDs, depending upon the kit. The name and location of the mode switch(es) vary per kit. See the kit documentation to understand what switches and LEDs on the kit are used for KitProg3. See Mode Switching for information on how to use the mode switches.

Table 2-1 describes how KitProg3 uses LEDs to let you know what's going on. In CMSIS-DAP Bulk and CMSIS-DAP HID modes, green means success and red means there was a problem. In DAPLink mode the green LED flashes when the USB MSC interface is active, the red LED flashes when USB CDC interface is active.

Table 2-1. Status LEDs

Mode Type	Programming Mode	Programming Status	Three LED Kit			Single LED Kit
			Amber LED	**Green LED**	**Red LED**	**Amber LED**
User modes	CMSIS-DAP Bulk	Programming	ON	8 Hz	OFF	8 Hz
		Success		ON	OFF	ON
		Error		OFF	ON	Flashing 0.5 Hz Duty cycle = 5%
	DAPLink		Ramping 2 Hz	Flashes when the USB MSC interface is active	Flashes when the USB CDC interface is active	Ramping 2 Hz
	CMSIS-DAP Bulk with two UARTs	Programming	N/A	N/A	N/A	8 Hz
		Success				2 pulse of 2 Hz then 1 second on
		Error				Flashing 0.5 Hz Duty cycle = 5%
	CMSIS-DAP HID	Programming	Ramping 1 Hz	8 Hz	OFF	8 Hz
		Success		ON	OFF	Ramping 1 Hz
		Error		OFF	ON	Flashing 0.5 Hz Duty cycle = 5%
Advanced modes	Bootloader	Not applicable	1 Hz	N/A	N/A	1 Hz

2.2.5 Bridging Feature

For more information on bridging feature availability on supported devices refer to Kit Support.

2.2.5.1 USB-UART Bridge Feature

This feature is available on all supported devices, can be accessed via any Serial Communicational Terminal Software to communicate with target device. Operating baud rates are described in section 3.2, values of data bits, stop bit and parity are not configurable and always must be set to default values: Data bits – 8, Parity – None, Stop bits – 1.

Second USB-UART interface is available only on some devices, how to access this feature see Mode Switching.

2.2.5.2 GPIO Bridge Feature

KitProg3 can be used for single wire communication with a target device via GPIO pins. To use this feature, you must have a USB HID/Bulk terminal and one of the supported kits.

You can send USB HID/Bulk requests to the KitProg3 device, which sets drive mode, sets state, reads state, and detects state transitions of GPIO pins. Refer to KitProg Host Protocol Interface specification for descriptions of the commands. Two KitProg3 GPIO pins that are connected to pins on the target device are available 3[5], 3[6].

2.2.5.3 USB-I2C/USB-SPI Bridge Feature

KitProg3 USB-SPI feature enables user to have communication between host and target device via I2C communication protocol, this feature is available for all supported devices. For operating speeds refer **here**.

USB-SPI feature enables user to communicate between host and target device via SPI communicational interface, and is not available on all supported devices.

For more details on how use USB-I2C/USB-SPI Bridging via Bridge Control Panel (BCP), refer to Help documentation issued with the BCP.

3 KitProg3 Design

KitProg3 firmware runs on specific hardware using a PSoC 5LP device. The hardware design is unchanged between KitProg2 and KitProg3. As a result, any kit that supports KitProg2 can be upgraded to KitProg3.

3.1 Supported Kits

Table 3-1 lists the development kits that support KitProg3. Kits released with KitProg2 can be upgraded to KitProg3. See Upgrading to KitProg3.

Table 3-1. Kit Support

Development Kits	Mode Switches	USB-UART	USB-2xUART	USB-I2C	USB-SPI	USB-GPIO
CY8CKIT-041-40XX PSoC 4 S-Series Pioneer Kit	One	+	-	+	-	-
CY8CKIT-041-41XX PSoC 4100S Pioneer Kit	One	+	-	+	-	-
CY8CKIT-041S-MAX PSoC 4100S Max Pioneer Kit	One	+	-	+	-	-
CY8CKIT-045S PSoC 4500S Pioneer Kit	One	+	-	+	-	-
CY8CKIT-145-40XX PSoC 4 S-Series Prototyping Kit	One	+	-	+	-	-
CY8CKIT-146 PSoC 4200DS Prototyping Kit	One	+	-	+	-	-
CY8CKIT-147 PSoC 4100PS Prototyping Kit	One	+	-	+	-	-
CY8CKIT-148 PSoC 4700S Inductive Sensing Evaluation Kit	One	+	-	+	-	-
CY8CKIT-149 PSoC 4100S Plus Prototyping Kit	One	+	-	+	-	-
CY8CKIT-062-BLE PSoC 6 BLE Pioneer Kit	Two	+	-	+	+	-
CY8CKIT-062-WiFi-BT PSoC 6 WiFi-BT Pioneer Kit	Two	+	-	+	+	-
CY8CPROTO-063-BLE PSoC 6 BLE Prototyping Kit	One	+	-	+	-	-
CY8CKIT-005 MiniProg4 Program and Debug Kit	Two	+	-	+	+	-
CY8CPROTO-062S2-4343W PSoC 6 Wi-Fi BT Prototyping Kit	One	+	-	+	-	-
CY8CPROTO-064S1-SB PSoC 64 "Secure Boot" Prototyping Kit	One	+	-	+	-	-
CY8CKIT-062S2-43012 PSoC 62S2 Wi-Fi BT Pioneer Kit	One	+	+	+	-	+
CY8CPROTO-062S3-4343W PSoC 62S3 Wi-Fi BT Prototyping Kit	One	+	-	+	-	-
CYW9P62S1-43438EVB-01 PSoC 62S1 Wi-Fi BT Pioneer Kit	One	+	+	+	-	+
CYW9P62S1-43012EVB-01 PSoC 62S1 Wi-Fi BT Pioneer Kit	One	+	+	+	-	+
CY8CPROTO-064B0S3 PSoC 64 "Secure Boot" Prototyping Kit	One	+	-	+	-	-
CY8CPROTO-064B0S1-BLE PSoC 64 BLE "Secure Boot" Prototyping Kit	One	+	-	+	-	-
CY8CKIT-064B0S2-4343W PSoC 64 Wi-Fi BT "Secure Boot" Pioneer Kit	One	+	+	+	-	+
CY8CKIT-064S0S2-4343W PSoC 64 "Standard Secure" - AWS Wi-Fi BT Pioneer Kit	One	+	+	+	-	+
CY8CKIT-062S4 PSoC 62S4 Pioneer Kit	One	+	-	+	-	-

Note "+" feature is supported, "-" feature is not supported.

3.2 Operating Speeds

Table 3-2. KitProg3 Operating Speeds

Functionality	Supported Speed	Units	Comments
SWD Programming	Up to 16	MHz	-
JTAG Programming	Up to 1.6	MHz	-
USB-UART Bridge[1]	300, 600, 1200, 1800, 2400, 4800, 7200, 9600, 14400, 19200, 28800, 33600, 38400, 57600, 115200, 128000, 230400, 250000, 256000, 500000, 1000000, 1152000, 1500000, 1600000, 2000000, 3000000, 4000000	Baud	Data bits – 8, Parity – None, Stop bits – 1 (This is configured by the KitProg and cannot be changed.)
USB-I2C Bridge	50, 100, 400, 1000	kHz	-
USB-SPI Bridge	50–6000	kHz	-

[1] The following baud rates are not supported on macOS: 250000, 500000, 1000000, 1152000, 1500000, 1600000, 2000000, 3000000, 4000000.

4 DAPLink Mode

Arm Mbed DAPLink is open-source software that provides alternative platform-independent programming/debugging interfaces between the target application and host PC. KitProg3 includes DAPLink.

DAPLink is supported by:

- ModusToolbox
- Mbed Online IDE (Arm® Mbed Enabled™ technology)
- Mbed CLI (Arm® Mbed Enabled™ technology)
- Mbed Studio (Arm® Mbed Enabled™ technology)
- Any IDE that supports CMSIS-DAP Bulk protocol

4.1 Supported Kits

The list of development kits that support the Arm® Mbed Enabled™ program is available on Arm Mbed website.

Table 4-1 shows KitProg3-based kits that have DAPLink mode available but might not be Arm Mbed enabled until program certification is completed.

Table 4-1. Kits with DAPLink Support

Development Kits
CY8CKIT-062-BLE PSoC 6 BLE Pioneer Kit
CY8CKIT-062-WiFi-BT PSoC 6 WiFi-BT Pioneer Kit
CY8CPROTO-062S2-4343W PSoC 6 Wi-Fi BT Prototyping Kit
CY8CPROTO-064S1-SB PSoC 64 "Secure Boot" Prototyping Kit
CY8CKIT-062S2-43012 PSoC 62S2 Wi-Fi BT Pioneer Kit
CY8CPROTO-062S3-4343W PSoC 62S3 Wi-Fi BT Prototyping Kit
CYW9P62S1-43438EVB-01 PSoC 62S1 Wi-Fi BT Pioneer Kit
CYW9P62S1-43012EVB-01 PSoC 62S1 Wi-Fi BT Pioneer Kit
CY8CKIT-005 MiniProg4 Program and Debug Kit
CY8CPROTO-064B0S3 PSoC 64 "Secure Boot" Prototyping Kit
CY8CPROTO-064B0S1-BLE PSoC 64 BLE "Secure Boot" Prototyping Kit
CY8CKIT-064B0S2-4343W PSoC 64 Wi-Fi BT "Secure Boot" Pioneer Kit
CY8CKIT-064S0S2-4343W PSoC 64 "Standard Secure" – AWS Wi-Fi BT Pioneer Kit

Note MiniProg4 in DAPLink mode supports only SWD Programming and Serial Communication. Drag-and-Drop Programming is not available via the CY8CKIT-005 MiniProg4.

4.2 Mbed Ecosystem

To use DAPLink, you need to upgrade kit firmware to KitProg3 v1.10 or later. See How-To: Update DAPLink Firmware section for details.

See also the *Mbed OS to ModusToolbox Flow* section of the Eclipse IDE for ModusToolbox User Guide. You can download it from the ModusToolbox website (under the Documentation tab).

4.3 Features

DAPLink provides three interfaces: drag-and-drop programming, a serial port, and debugging support.

4.3.1 Drag-and-Drop Programming

Program the target PSoC 6 MCU device by copying or saving a file in one of the supported formats to the DAPLink drive. Upon completion, the drive remounts. If a failure occurs, the file FAIL.TXT appears on the drive containing information about the failure. Drag-and-Drop Programming is not supported for MiniProg4 in DAPLink mode. Attempting to use MSD device programming with MiniProg4 will result in failure.

Supported file formats:

- Raw binary file
- Intel Hex

You can control DAPLink with MSD Commands.

Note DAPLink provided with KitProg3 does not implement the **start_bl.act** MSD command because it uses the KitProg3's own Bootloader.

4.3.2 Serial Port

The serial port is connected directly to the target PSoC 4 / PSoC 6 MCU device allowing for bidirectional communication. It also allows the target to be reset by sending a break command over the serial port.

Table 4-2. UART Parameters and Speed

Functionality	Supported Speed	Units	Comments
USB-UART Bridge[1]	1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200, 250000, 500000, 1000000, 2000000, 3000000, 4000000	Baud	Data bits – 8, Parity – None, Stop bits – 1

4.3.3 Debugging

You can debug with any IDE that supports the CMSIS-DAP protocol. Some tools capable of debugging are:

- ModusToolbox
- Mbed CLI
- Mbed Studio
- µVision
- IAR Workbench

[1] The following baud rates are not supported on macOS: 250000, 500000, 1000000, 2000000, 3000000, 4000000.

4.3.4 User Interface

See KitProg3 LEDs to learn how they are used to let you know what’s going on with DAPLink.

4.4 How To

4.4.1 Switch to and from DAPLink Mode

See Mode Switching.

4.4.2 Update DAPLink Firmware

DAPLink is part of the latest KitProg3 package when you update KitProg to a version that includes DAPLink. See Installing KitProg3 for information on where to get KitProg3.

If you wish to upgrade KitProg to a version that includes DAPLink, you need a tool called Firmware Loader. See Command-Line Options for information how to use Firmware Loader tool.

5 KitProg3 vs. KitProg2

Use this chapter to understand the differences between KitProg versions and decide which to use.

5.1 Feature Comparison

Table 5-1. KitProg Feature Comparison

Feature	DAPLink	KitProg3	KitProg2
Protocol	Serial Wire Debug (SWD)	Serial Wire Debug (SWD) JTAG	Serial Wire Debug (SWD)
Transport Mechanism	CMSIS DAP v2.0.0	CMSIS DAP v2.0.0 CMSIS DAP v1.2.0	CMSIS DAP v1.1.0 Proprietary
USB Mass Storage Device	Yes	No	Yes
USB Endpoints	Bulk, CDC, MSC	Bulk and HID, CDC	HID, CDC
IDE Support	ModusToolbox Mbed CLI Mbed Studio IAR Embedded Workbench Keil µVision	ModusToolbox PSoC Creator IDE IAR Embedded Workbench Keil µVision	PSoC Creator IDE
Programmer Support	N/A	Cypress Programmer PSoC Programmer	PSoC Programmer
Kit support	See Arm Mbed Enabled™ Kit Support	The same, see Supported Kits	

* Note that the Bridge Control Panel is not supported by Cypress Programmer or ModusToolbox.

This comparison does not include programming speed because that depends upon several variables, such as the target flash memory (type and size), programming tool overhead, and the transport mechanism. These vary widely from tool to tool, and kit to kit. However, KitProg3 is 2-4x faster than KitProg2 using the CMSIS-DAP transport mechanism, because it uses Bulk endpoints for faster data transfer. JTAG programming interface is available only for MiniProg4.

5.2 Upgrading to KitProg3

You can upgrade the kit firmware by using one of these:

- PSoC Programmer (may not include the latest KitProg3 Firmware)
- Cypress Programmer (may not include the latest KitProg3 Firmware)
- Firmware Loader Tool (includes latest KitProg3 Firmware with DAPLink)

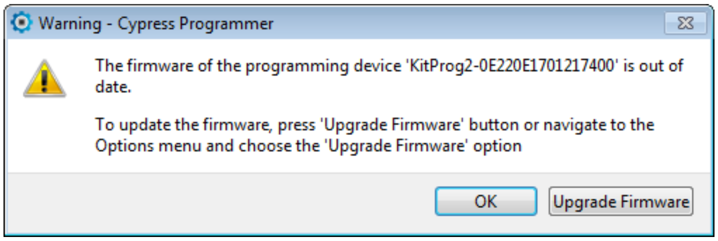
This section describes how to upgrade kit firmware to KitProg3 by using PSoC Programmer or Cypress Programmer. See How-To: Update DAPLink Firmware for instructions on how to upgrade kit firmware to latest KitProg3 FW with DAPLink.

Because tools are released individually and on their own schedule, the most recent release of a programming tool may not have the very latest version of KitProg3 to install and update. You can also use the Firmware Loader (fw-loader) tool to upgrade the kit. See Updating KitProg3 for more information.

Use Cypress Programmer or PSoC Programmer to connect to a kit. The tool connection tells you what firmware is on the kit. If KitProg2 is installed on the kit, the tool notifies you, and gives you the option to upgrade firmware. Click **OK** to leave the firmware unchanged.

Click **Upgrade Firmware** and KitProg3 is loaded into the kit. The Cypress Programmer or PSoC Programmer log window provides progress information and confirms connection to the new KitProg3 firmware on the kit.

Figure 5-1. Upgrading to KitProg3 in Cypress Programmer

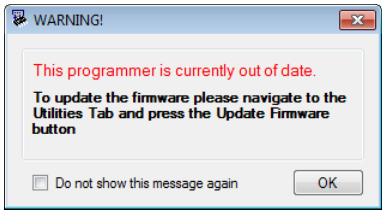


Warning - Cypress Programmer

The firmware of the programming device 'KitProg2-0E220E1701217400' is out of date.

To update the firmware, press 'Upgrade Firmware' button or navigate to the Options menu and choose the 'Upgrade Firmware' option

Figure 5-2. KitProg3 Firmware Update Warning in PSoC Programmer



WARNING!

This programmer is currently out of date.

To update the firmware please navigate to the Utilities Tab and press the Update Firmware button

□ Do not show this message again

6 Updating KitProg3

You can update your KitProg device to latest version of KitProg3 firmware with DAPLink using the cross-platform, command line Firmware Loader (fw-loader) tool. The also allows you to switch between different modes programmatically. See Mode Switching for a discussion of the different modes.

6.1.1 Where to get it

You can find the latest version of fw-loader tool on Cypress GitHub repository. Download the appropriate zip archive for your OS. This tool is also installed with the ModusToolbox tools package installer.

6.1.2 Install fw-loader Tool

Unzip downloaded archive tool to any convenient location. Move to bin directory.

Note On a Linux machine, you must run the *udev_rules/install_rules.sh* script before the first run of the fw-loader tool.

6.1.3 Command-Line Options

The fw-loader tool provides several commands. Run the tool from the command line, as follows:

```
        [install_path]/fw-loader/bin/fw-loader
```

For Windows, you can use the *fw-loader.bat* file provided with the ModusToolbox tools package installer.

Use the following options, as needed:

Option	Description
`--help or /?` (or no arguments)	Display a list of supported commands with their descriptions.
`--device-list`	Display a list of connected devices.
`--update-kp3 [device-name]`	Update the firmware of the specified device to KitProg3.
`--update-kp2 [device-name]`	Downgrades the firmware of the specified device to KitProg2.
`--mode <mode> [device-name]`	Switches KitProg3 mode of the specific device. Supported modes are: 'kp3-hid', 'kp3-bulk', 'kp3-bootloader', 'kp3-daplink'.

Note If you have only one device attached, the `[device-name]` switch is optional.

Examples of `[device-name]` include:

- KitProg3 device: `KitProg3 CMSIS-DAP BULK-1A1509F301237400`
- KitProg2 device: `KitProg2-1A1509F301237400`

To get the name of a device use the `--device-list` option.

7 Troubleshooting

This section lists known issues, along with any workarounds.

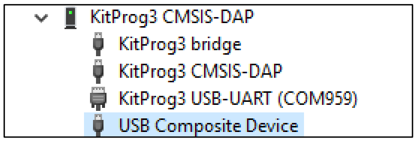
On Windows, after updating to KitProg3 v2.10, PSoC Creator, PSoC Programmer, CapSense Tuner, or the Bridge Control Panel can’t connect to a device.

Previous versions of KitProg3 implement I2C/SPI bridging using HID endpoints. In version 2.10 and newer, bridging is implemented on bulk endpoints for improved performance. After updating the KitProg3, Windows may continue using the old driver “hidusb” for I2C/SPI bridging instead of the proper driver for bulk endpoints - “winusb”. To fix this issue, follow these steps:

1. In the Device Manager, find Composite Device for the KitProg3 device.

    a. Change View to **Devices by container**.

    b. Find **KitProg3 CMSIS-DAP** or **MiniProg4 CMSIS-DAP** and open the container.



KitProg3 CMSIS-DAP
KitProg3 bridge
KitProg3 CMSIS-DAP
KitProg3 USB-UART (COM959)
USB Composite Device

2. Right click on **USB Composite Device** and choose **Uninstall device**.

3. Click on **Scan for hardware changes button**.

On Windows, after upgrading to KitProg3 v2.10 in CMSIS-DAP Bulk mode, simultaneous use of the USB-I2C/SPI bridging and debugging is not possible.

Previous versions of KitProg3 implement I2C/SPI bridging using HID endpoints. In version 2.10 and later, bridging is implemented on bulk endpoints to improve performance. After the upgrade, Windows cannot use the I2C/SPI bridging interface and the CMSIS-DAP bulk interface at the same time because of a WinUSB driver limitation. If you would like to use debug and 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 fw-loader command.
- 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.

After update to KitProg3 v2.10 my Windows 7 fails to install "KitProg3 bridge" driver.

In some cases, Windows 7 cannot recognize the KitProg3 bridge Microsoft OS descriptor, and the result is that it cannot recognize KitProg3 bridge as a WinUSB device. In some editions of Windows 7, the .inf file that enables Windows 7 to recognize that descriptor may be missing. The solution is to install a digitally signed driver manually from the Windows Update Catalog. Windows 8 and later auto install drivers without a .inf file, so this is not an issue. Follow these steps to recover the driver for Windows 7:

1. Download the driver and extract it to a local folder on your computer:

   https://www.catalog.update.microsoft.com/Search.aspx?q=Microsoft%20Other%20hardware%20WinUsb%20Device

2. In the Device Manager, find “KitProg3 bridge” with a yellow exclamation mark and update it manually by providing path to the extracted driver.

   a. Browse my computer for driver software

   b. Select “Let me pick from a list of device drivers on my computer and “Have disk.”

   c. Browse and provide path to the driver

   d. Click on the “winusbcompat.inf” file

   e. Click open and then **Next**.

I use Windows 7 or 10. When in Bulk mode (amber status LED on), the kit is not recognized by the programming tool, and debug does not work with ModusToolbox or PSoC Creator IDE. If I switch to HID mode, it works.

#1: This is a known issue with Windows 7 only driver updates. Instead of using the correct driver, Windows update installs the HP Printer (BIDI) driver *if the machine is connected to the Internet*. As a result, KitProg3 will not work when in Bulk mode. The problematic HP Printer driver is no longer available in the Windows update system. However, it is possible that the wrong driver was installed at an earlier time. To fix this issue, follow these steps:

1. Uninstall the driver from the Device Manager.

2. Close any internet connection.

3. Attach the kit to your computer and rescan the device in the Device Manager.

#2: Several versions of the KitProg3 CMSIS-DAP driver may be available on your machine. Instead of using the correct driver, Windows uses a wrong or faulty driver. As a result, the kit in Bulk mode will not be recognized by programming/debug tools.

For Windows 7, the solution is to install the proper driver that is provided with PSoC Programmer or fw-loader software:

1. Switch KitProg3 to Bulk mode.

2. In the Device Manager, go to the KitProg3 CMSIS-DAP device and update the driver manually:

   a. Browse my computer for driver software.

   b. Select “Let me pick from a list of device drivers on my computer” and “Have disk.”

   c. Open drivers.

   d. Open KitProg3.

   e. Click on KitProg3.inf and click **Next**.

For Windows 10, the solution is to reinstall the native driver:

1. Open the Device Manager and go to the KitProg3 CMSIS-DAP device.

2. Right-click the device and select Uninstall from the context menu (Select checkbox Delete the driver software for this device if present)

3. After uninstalling re-plug the device.

4. Right-click the device and select Update driver software... from the context menu.

5. In the wizard, select Search automatically for updated driver software.

CMSIS-DAPv2 interface driver for DAPLink mode isn't installed properly on Windows 7.

In some cases, Windows 7 cannot recognize the CMSIS-DAP v2 Microsoft OS descriptor, and it cannot detect CMSIS-DAP v2 as a WinUSB device. In some editions of Windows 7, the .inf file that enables Windows 7 to recognize that descriptor may be missing. The solution is to install a digitally unsigned driver manually from the https://arm-software.github.io/CMSIS_5/DAP/html/group__DAP__ConfigUSB__gr.html. Windows 8 and later auto install drivers without a .inf file, so this is not an issue. Follow these steps to recover the driver for Windows 7:

1. Uninstall the Mbed composite device driver (if installed). In the Device Manager, find "mbed Composite Device" and uninstall it.

2. Create a "CMSIS-DAP v2.inf" file by copying the contents of the .inf file from https://arm-software.github.io/CMSIS_5/DAP/html/group__DAP__ConfigUSB__gr.html and pasting it to "CMSIS-DAP_v2.inf" file on your computer.

3. In the Device Manager, find "CMSIS-DAP v2" with a yellow exclamation mark and update it manually by providing path to the extracted driver.

    a. Browse my computer for driver software

    b. Select "Let me pick from a list of device drivers on my computer and "Have disk."

    c. Browse and provide path to the "CMSIS-DAP_v2.inf" file created in step 2.

    d. Click on the "CMSIS-DAP_v2.inf" file and click on **Open**.

    e. Click "**OK** and then **Next**.

CMSIS-DAPv2 interface driver for DAPLink mode isn't installed properly on Windows 10

It is possible to have incorrect drivers installed for the device, for example, if custom drivers were installed. Solution to this issue is to reinstall drivers:

1. Uninstall the mbed serial driver and the USB Composite Device Driver

    a. In the Device Manager, select the **View** menu and choose the "Devices by container" option.

    b. In the list of devices, find "mbed Serial Port (COMXX)" and uninstall it.

        □ Right-click the device and select **Uninstall** from the context menu.

        □ Select the **Delete the driver software for this device** check box, if present.

    c. In the list of devices, find "Device" or "DAPLink CMSIS-DAP".

        □ Open it and make sure it contains the entry "DAPLINK" or "mbed Composite Device".

        □ Uninstall "CMSIS-DAP v2", "USB Composite Device", and "mbed Composite Device" devices from this container.

            – Right-click the device and select **Uninstall** from the context menu.

            – Select the **Delete the driver software for this device check box**, if present)

2. Click the **Scan for Hardware Changes** button in the Device Manager to install the correct drivers.

How do I recover a corrupted KitProg3 image?

Although unlikely, it is possible to corrupt the KitProg3 image, for example, if a firmware update is interrupted.

To fix this issue, put the KitProg3 into bootloader mode. (Press the Mode switch while plugging in the kit.) Then follow the instructions for your programmer.

- Launch Cypress Programmer. Cypress Programmer automatically updates the KitProg3 firmware.

  OR

- Launch PSoC Programmer. Update KitProg3 firmware via **Utilities** > **Update firmware** option.

Mbed CLI interface shows errors and warning while running Mbed on Cypress kits.

This is a known issue caused by Mbed OS installation problems. See the *Install and configure Mbed CLI* section of the Eclipse IDE for ModusToolbox User Guide. You can download from the ModusToolbox Software website (Documentation tab).

Revision History

Document Title: KitProg3 User Guide **Document Number: 002-24616**		
Revision	**Date**	**Description of Change**
**	10/26/2018	New kit guide.
*A	11/08/2018	Updated Introduction. Updated Description. Updated KitProg3 Tools Support and Compatibility. Updated Installing and Using KitProg3. Updated Using KitProg3. Updated Programming and Debugging. Updated description. Updated Mode Switching. Updated description. Updated KitProg3 Design. Updated KitProg3 . Updated description. Updated KitProg3 vs. KitProg2. Updated description. Updated Troubleshooting. Updated description.
*B	11/22/2018	Added PSoC Creator/PSoC Programmer information
*C	02/19/2019	Updates throughout for KitProg3 v1.1 and DAPLink Updated Introduction. Added DAPLink Mode. Updated KitProg3 User Interface. Updated Troubleshooting. Updated KitProg3 Operating Speeds. Updated KitProg3 Status LEDs. Updated Downgrading to KitProg2. Updated Upgrading to KitProg3.
*D	2/21/2019	Remove listing of an unsupported kit
*E	5/29/2019	Updated DAPLink CMSIS-DAP HID protocol to DAPLink CMSIS-DAP Bulk protocol Updated description of Mode Switching between HID, Bulk and DAPLink interfaces Added description of Mode Switching between CMSIS-DAP Bulk with two UARTs and UART + bridging Added new supported kits for KitProg3 firmware Added DAPLink CMSIS-DAP v2 driver issue on Windows 7 into Troubleshooting section
*F	7/24/19	Minor edits.

Document Title: KitProg3 User Guide **Document Number: 002-24616**		
*G	9/26/2019	Added new supported kits for KitProg3 firmware Updated description of status LEDs behavior Added section 6 Updating KitProg3 Updated Update DAPLink Firmware and Upgrading to KitProg3 sections. Removed Downgrading to KitProg2 section.
*H	12/12/2019	Updated mode switching diagram Added description about UARTx2 mode Updated KitProg3 Status LEDs. Added new supported kits for KitProg3 firmware Updated figure of KitProg3 High-Level Architecture
*I	3/18/2020	Updated KitProg3 High-Level Architecture figure Updated mode switching diagram Updated mode switching information for CMSIS-DAP HID Added new supported kit for KitProg3 firmware in DAPLink mode Updated KitProg3 vs. KitProg2 section Updated lists of supported IDEs for KitProg3 firmware in sections Introduction, DAPLink mode and Installing and Using KitProg3 Added information about JTAG programming via MiniProg4 Added information about MiniProg4 in DAPLink mode Updated Operating speeds table in KitProg Design section Added DAPLink driver issue on Windows 10 into Troubleshooting section
*J	6/11/2020	Clarified LED connection in section 2.2.1. Updated list of supported kits in section 3.1 and section 4.1. Clarified Troubleshooting process for updating drivers in Chapter 7.
*K	7/28/2020	Updated KitProg3 Tools Support and Compatibility. Added Windows driver issues into Troubleshooting section Added information about using simultaneous use of the USB-I2C/SPI bridging and debugging in Windows
*L	8/19/2020	Added footnotes about Serial Port limitation on macOS in section 3.2 and section 4.3.2
*M	01/29/2021	Updated Launching a Debug Session in Eclipse IDE for ModusToolbox picture in section 2.2.2. Updated mode switching information in section 2.2.3. Added Bridging Features section 2.2.5. Updated list of supported kits in section 3.1. Updated list of supported baud rates in section 3.2. Updated footnote about Serial Port limitation on macOS in section 3.2. Updated DAPLink Serial Port information in section 4.3.2 Updated Updating KitProg3 information section 6. Updated solution to DAPLink CMSIS-DAP v2 issue on Windows 7 in section 7.
*N	04/20/21	Added reference to KitProg Host Protocol Interface in section 2.2.5 Added note on Supported kits in section 3.1 Added example of “device-name” in section 6.1.3 Updated Operating speeds table in section 3.2