KitProg3 user guide

A newer version of this document may be available online here.

About this document

Scope and purpose

KitProg3 is our low-level communication firmware for programming and debugging. It provides communication between a programming tool (such as ModusToolbox™ Programmer or PSOC™ Programmer) and a target, such as a PSE84 MCU. KitProg3 supports a variety of development kits. It is also the communication firmware found in the MiniProg4 debug probe.

Our 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.

Document conventions

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

Table of contents

1 Introduction 3

1.1 What's in this Guide 4
1.2 KitProg3 Tools Support and Compatibility 4

2 Installing and Using KitProg3 6

2.1 Installing KitProg3 6
2.2 Using KitProg3 6

3 KitProg3 Design 14

3.1 Supported Kits 14
3.2 Operating Speeds 17

4 DAPLink Mode 18

4.1 Supported Kits 18
4.2 Using DAPLink 18
4.3 Features 19
4.4 How To 20

5 KitProg3 vs. KitProg2 22

5.1 Feature Comparison 22
5.2 Upgrading to KitProg3 22

6 Updating KitProg3 24

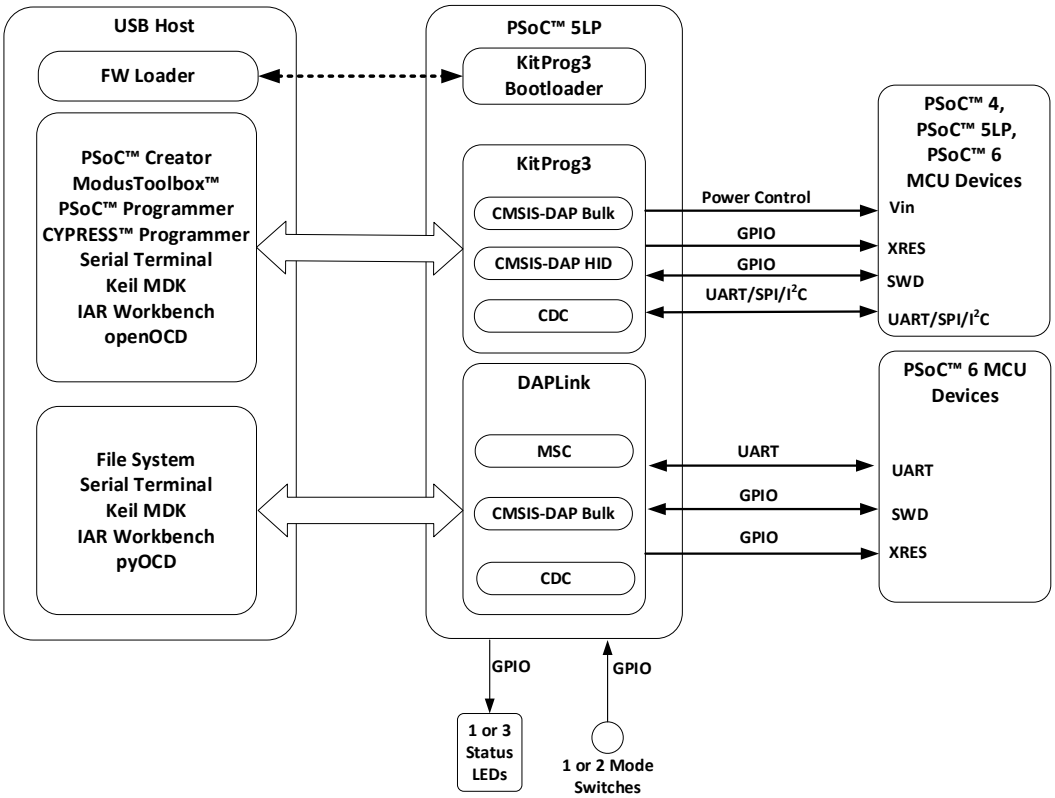
7 Troubleshooting 25

1 Introduction

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.

Figure 1-1. KitProg3 High-Level Architecture



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.

Attention: ***Arm® Mbed™ DAPLink mode will be deprecated and removed from KitProg3 package in the future with corresponding changes in kit functionality and documentation. After that, Infineon development kits will become unusable in the Mbed ecosystem.***

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, also, includes full details about KitProg3 User interface, mode switching and status LEDs behavior.
- Section 3: KitProg3 Design – includes the list of supported devices, supported operating speeds for each supported interface/bridge.
- 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 – 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 ModusToolbox™ Programmer
  - PSOC™ Creator and PSOC™ Programmer (v 3.28 or later)
  - openOCD CLI
  - µVision
  - IAR Embedded Workbench
  - Visual Studio Code
- DAPLink mode
  - ModusToolbox™
  - pyOCD
  - µVision
  - IAR Embedded Workbench
  - Visual Studio Code

Table 1-1. KitProg Compatibility

KitProg	IDE	Programmer	Bridging Tools
KitProg3	ModusToolbox™ & PSOC™ Creator	ModusToolbox™ Programmer & PSOC™ Programmer	Bridge Control Panel (PSOC™ Programmer) [1] CAPSENSE™ Tuner (PSOC™ Creator and ModusToolbox™)
KitProg2	PSOC™ Creator	PSOC™ Programmer	

[1] Bridge Control Panel is not supported by ModusToolbox™ 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 I^2C/SPI/UART/GPIO Bridging Voltage control SOM identification EEPROM
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 ModusToolbox™ Programmer
- PSOC™ Creator and PSOC™ Programmer
- Any IDE that supports CMSIS-DAP Bulk protocol
- Bridge Control Panel (BCP) 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™, ModusToolbox™ Programming tools, 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 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.

Support 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.

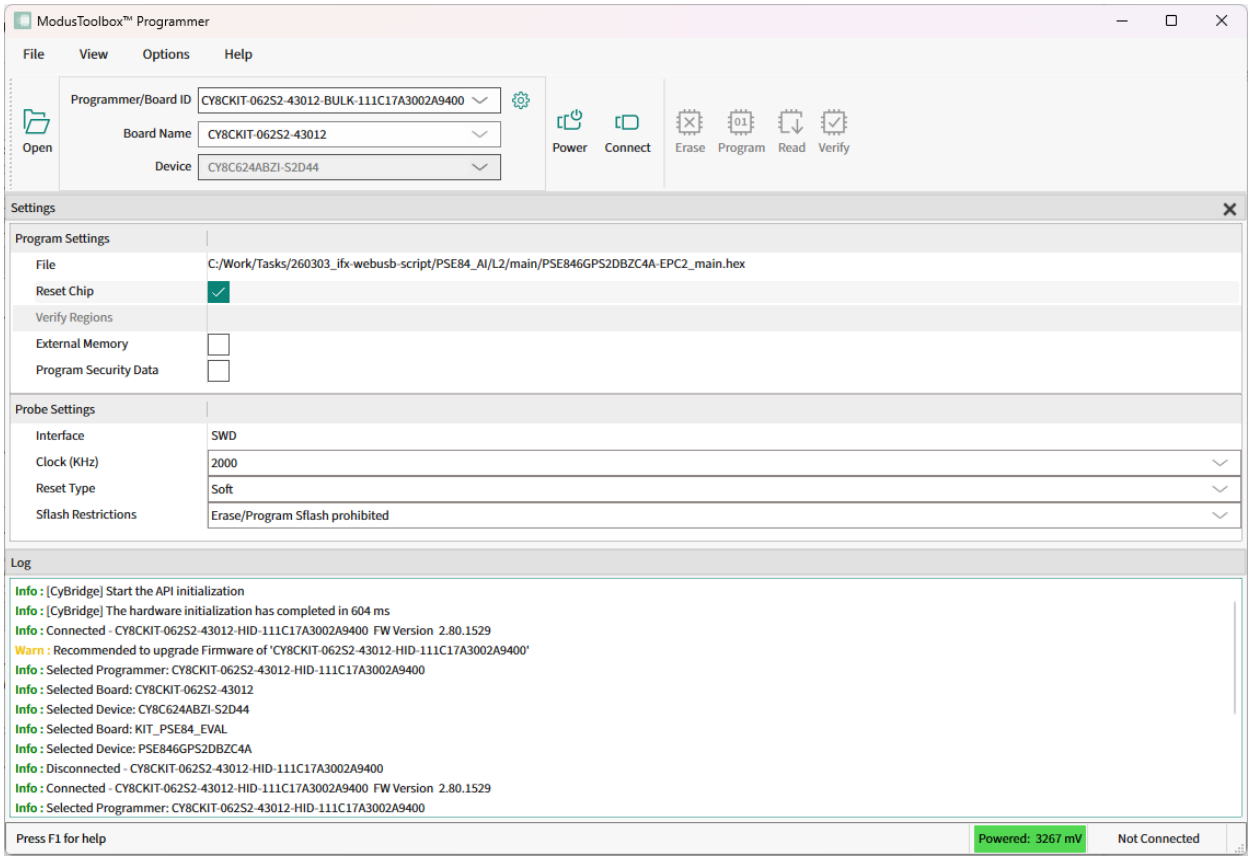
If the LED is steady (out of box behavior), KitProg3 is using Bulk endpoints for faster communication. If the LED is ramping at 2 Hz, KitProg3 is in DAPLink mode. If the LED is ramping 1 Hz, KitProg3 is using HID endpoints, which means slower communication.

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.

Installing and Using KitProg3

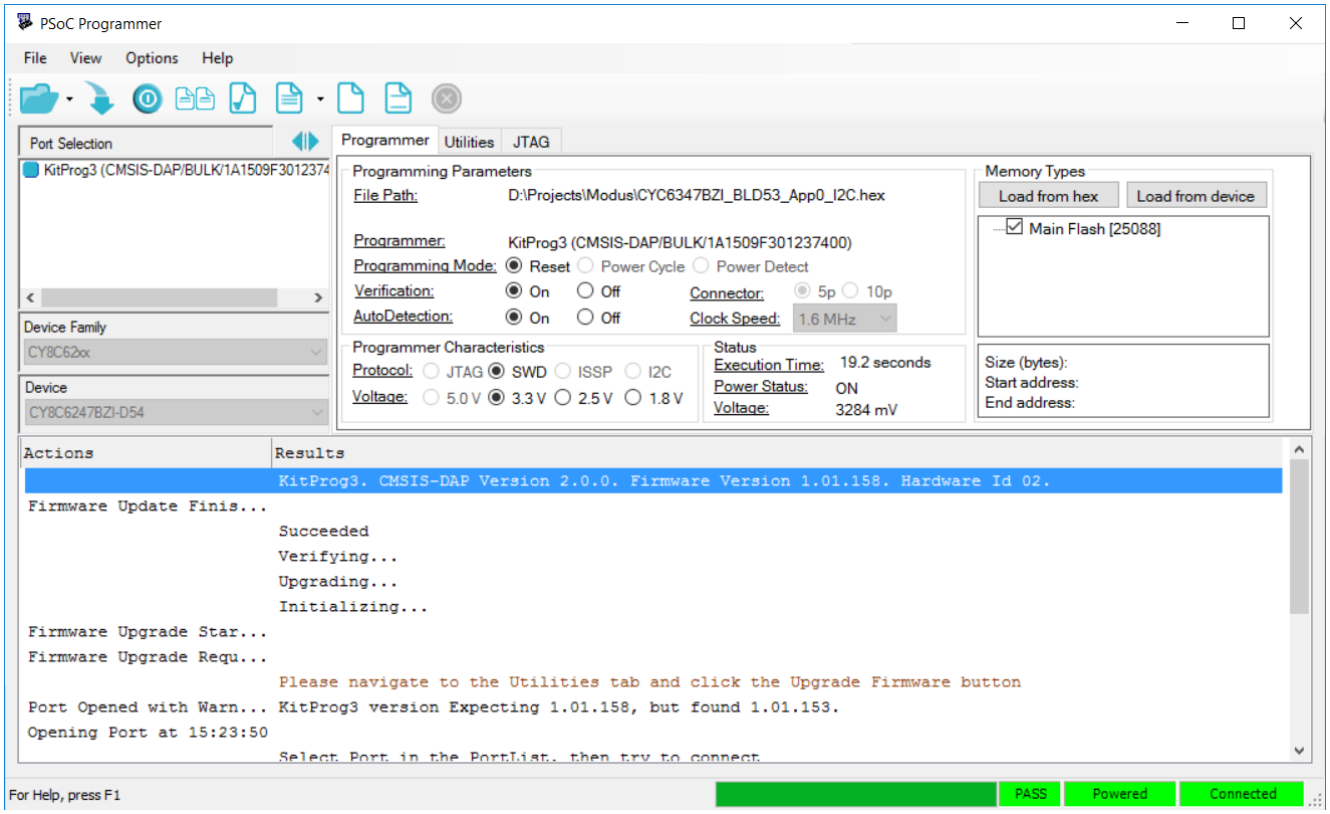
Launch your programming tool or the Eclipse IDE for ModusToolbox™. The tool connects to KitProg3 automatically. The KitProg3 connection appears in the UI of the programming tool. Note that at this time, some programming tools do not recognize the kit if it is in DAPLink mode. Switch the kit to Bulk or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

Figure 2-1. Connected via ModusToolbox™ Programmer



Installing and Using KitProg3

Figure 2-2. Connected via PSOC™ Programmer

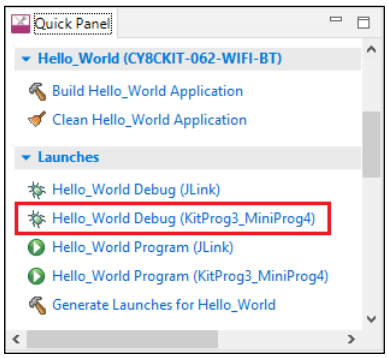


2.2.2 Programming and Debugging

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

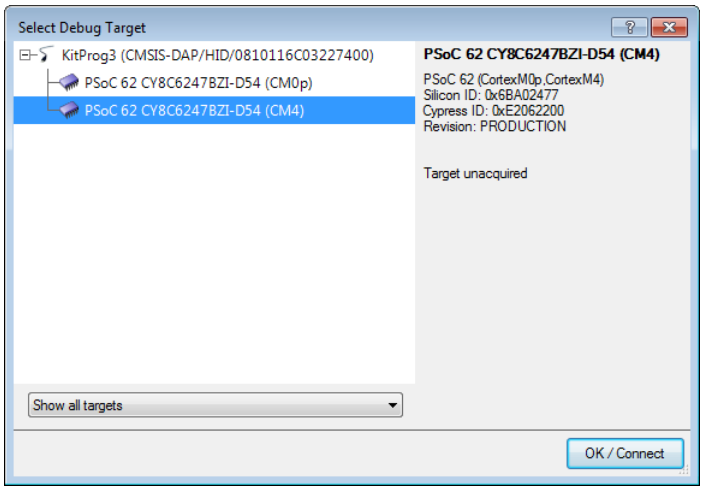
For example, for the Eclipse IDE, click the **Debug (KitProg3_MiniProg4)** link in the **Quick Panel**. To program the device without debugging, use the **Program (KitProg3_MiniProg4)** link.

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



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

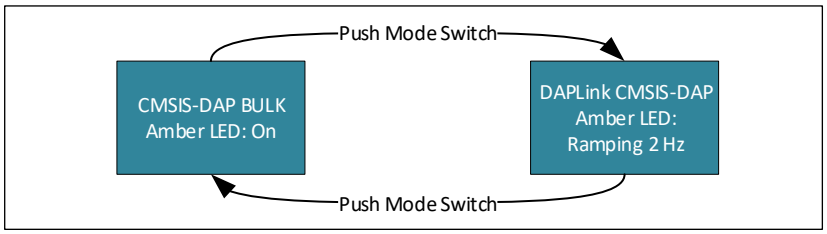
Figure 2-4. Selecting the Debug Target in PSOC™ Creator



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.

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).

To enter UARTx2 if the kit is in DAPLink mode use the Firmware Loader tool. 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 UART Half-Duplex Mode

MiniProg4 supports a special UART configuration for half-duplex (single-wire) communication. This allows the programmer to communicate with a target device where only one line is used for both transmitting and receiving data.

Unlike operational modes such as UARTx2 or Bootloader, Half-Duplex mode is not activated using the mode switch. Instead, it is enabled or disabled by sending a specific software command to the programmer. The selected setting (either Full-Duplex or Half-Duplex) is stored in the programmer's non-volatile memory, and it is retained across power cycles.

To exit Half-Duplex mode, the corresponding software command must be sent to re-enable Full-Duplex mode.

2.2.3.3 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, and it 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

A second USB-UART interface is available only on some devices. To access this feature, see Mode Switching.

If the device supports the UART Hardware Flow Control feature (refer to the "Kit Support" section for details), you can configure use of this feature via Firmware Loader (fw-loader) commands. This configuration allows you to switch between "no hardware flow control" and "hardware flow control" as per your requirement.

Use this command to set UART flow control mode:

```
fw-loader --set-kp3-flow-conrol <port_number> <mode>
```

Use this command to read the current flow control mode set:

```
fw-loader --get-kp3-flow-control <port_number>
```

For supported values of `<port_number>` and `<mode>` parameters refer to Firmware Loader User Guide.

2.2.5.2 GPIO Bridge Feature

KitProg3 can be used for single wire communication with a target device via GPIO pins. There are two ways to use this feature; either through a USB HID/Bulk terminal or with the fw-loader command line tool (see Firmware Loader User Guide for more information). Two KitProg3 GPIO pins are connected to pins on the target device for this purpose: 3[5] (Port 3, Pin 5), 3[6] (Port 3, Pin 6).

You can send USB HID/Bulk requests to set drive mode, set state, read state, and detect state transitions of GPIO pins. Refer to the KitProg host protocol interface specification for descriptions of the commands.

You can also use fw-loader GPIO dedicated command line options to set mode and state of GPIO pin:

```
fw-loader --set-kp3-gpio-pin <pin_number> <pin_mode> <state>
```

Use this command to read the current state of a pin:

```
fw-loader --read-kp3-gpio-pin <pin_number>
```

For example, to set 3[5] (port 3, pin 5) GPIO pin to High state in ResUp operational mode, use

```
fw-loader --set-kp3-gpio-pin 35 ResUp 1
```

To read current state of 3[6] (port 3, pin 6) GPIO pin, use:

```
fw-loader --read-kp3-gpio-pin 36
```

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 not 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.

2.2.6 SOM Identification

Some development kits are built as a baseboard with a replaceable system on module (SOM). On some of these kits, the SOM carries an EEPROM holding an identification record that describes the module. KitProg3 reads this record and makes it available to host tools, which can provide additional information about the connected SOM. For availability on supported devices, refer to Kit Support.

KitProg3 reads the record once at start-up. If you want to replace the SOM, disconnect and reconnect the kit so that the new module is detected.

The record is read over the same I²C bus that the USB-I²C bridge uses, at I²C addresses 0x50 to 0x57. On these kits, I²C addresses 0x50 to 0x57 are reserved for the SOM identification EEPROM. Do not assign an address in this range to your own I²C devices on the bridge bus.

To obtain this information, refer to the KitProg3 host protocol interface specification for a description of the new dedicated Read Extended Target Info command.

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

The following table lists the development kits that support KitProg3: "A" feature is supported; "N/A" feature is not supported. Kits released with KitProg2 can be upgraded to KitProg3; see Upgrading to KitProg3. For devices that support GPIO Bridging, see the GPIO Bridge Feature section for the list of GPIO pins.

Table 3-1. Kit Support

Development Kits	Mode Switches	USB-UART	USB-2xUART	UART HW Flow Control	USB-I²C	USB-SPI	USB-GPIO	SOM ID
CY8CKIT-041-40XX	One	A	N/A	N/A	A	N/A	N/A	N/A
CY8CKIT-041-41XX (PSOC™ 4100S Pioneer Kit)								
CY8CKIT-045S (PSOC™ 4500S Pioneer Kit)								
CY8CKIT-145-40XX (PSOC™ 4000S CAPSENSE™ Prototyping Kit)								
CY8CKIT-146 (PSOC™ 4200DS Prototyping Kit)								
CY8CKIT-147 (PSOC™ 4100PS Prototyping Kit)								
CY8CKIT-148 (PSOC™ 4700S Inductive Sensing Evaluation Kit)								
CY8CKIT-149 (PSOC™ 4100S Plus Prototyping Kit)								
CY8CPROTO-063-BLE (PSOC™ 6 BLE Prototyping Kit)								
CY8CPROTO-064S1-SB (PSOC™ 64 "Secure Boot" Prototyping Kit)								
CY7110 (EZ-PD™ PMG1-S0 prototyping kit)								
CY7111 (EZ-PD™ PMG1-S1 Prototyping Kit)								
CY7112 (EZ-PD™ PMG1-S2 Prototyping Kit)								
CY7113 (EZ-PD™ PMG1-S3 prototyping kit)								
CY8CPROTO-064B0S3 (PSOC™ 64 "Secure Boot" Prototyping Kit)								
CY8CPROTO-064B0S1-BLE (PSOC™ 64 Bluetooth® LE "Secure Boot" Prototyping Kit)								
CY8CKIT-062S4 (PSOC™ 62S4 Pioneer Kit)								
KIT_PSOC4-HVPA-144K_LITE (PSOCTM HV PA 144k Lite kit)								
CY8CPROTO-040T (PSOC™ 4000T CAPSENSE™ prototyping kit)								

KitProg3 Design

Development Kits	Mode Switches	USB-UART	USB-2xUART	UART HW Flow Control	USB-I²C	USB-SPI	USB-GPIO	SOM ID
KIT_PSOC4-HVMS-128k_LITE (PSOC™ 4 HVMS Lite Kit)								
KIT_XMC71_EVK_LITE_V1 (XMC7100 Evaluation Board)								
CY8CKIT-062S2-AI (PSOC™ 6 Artificial Intelligence evaluation kit)								
KIT_PSOC4-HVMS-64K_LITE								
EVAL_PMG1_B1_DRP								
EVAL_PMG1_S1_DRP								
EVAL_PMG1_S3_DUALDRP								
CY8CPROTO-041TP (PSOC™ 4100T Plus Prototyping Kit)								
CY8CPROTO-040T-MS (PSOC™ 4000T Multi-Sense Prototyping Kit)								
KIT_PSC3M5_EVK (PSOC™ Control C3M5 Evaluation Kit)								
KIT_T2G_C-2D-6M_LITE (TRAVEO™ T2G Cluster 6M Lite Kit)								
KIT_PSE84_AI (PSOC™ Edge E84 AI Kit)								
KIT_PSE84_HMI (PSOC™ Edge E84 HMI Kit)								
KIT_PSC3M8_EVK (PSOC™ CONTROL C3M8 EVALUATION KIT)								
KIT_XMC52_EVK								
KIT_T2G_C-2D-4M_LITE								
CY8CKIT-062-BLE (PSOC™ 6 Bluetooth® LE Pioneer Kit)	Two	A	N/A	A	A	A	N/A	N/A
CY8CKIT-062-WiFi-BT (PSOC™ 6 Wi-Fi-Bluetooth® Pioneer Kit)								
CY8CKIT-005 (MiniProg4 Program and Debug Kit)								
CY8CPROTO-062S2-4343W (PSOC™ 6 Wi-Fi BT Prototyping Kit)	One	A	N/A	A	A	N/A	N/A	N/A
CY8CPROTO-062S2-43012 (PSOC™ 62S2 Wi-Fi BT Pioneer Kit)								
CY8CPROTO-062S2-43439 (PSOC™ 62S2 Wi-Fi Bluetooth® Prototyping Kit)								
CYSBSYSKIT-DEV-01 (Rapid IoT Connect Developer Kit)								
CY8CKIT-041S-MAX (PSOC™ 4100S Max Pioneer Kit)								
CYW920829M2EVK-02 (AIROC™ CYW20829 Bluetooth® LE MCU Evaluation Kit)								
CYW920829B0M2P4TAI100-EVK Evaluation Kit								
CYW920829B0M2P4EPI100-EVK Evaluation Kit								
CY8CPROTO-062S3-4343W (PSOC™ 62S3 Wi-Fi Bluetooth® Prototyping Kit)								
CY8CPROTO-062-4343W (PSOC™ 6 Wi-Fi BT Prototyping Kit)								

KitProg3 Design

Development Kits	Mode Switches	USB-UART	USB-2xUART	UART HW Flow Control	USB-I²C	USB-SPI	USB-GPIO	SOM ID
CYTVII-B-E-1M-SK (TRAVEO™ T2G Low-Cost, Arduino Compatible Evaluation Board)	One	A	N/A	N/A	A	N/A	A	N/A
CY8CKIT-040T (PSOC™ 4000T CAPSENSE™ evaluation kit)								
CY8CKIT-062S2-43012 (PSOC™ 62S2 Wi-Fi Bluetooth® Pioneer Kit)	One	A	A	A	A	N/A	A	N/A
CYW9P62S1-43438EVB-01 (PSOC™ 62S1 Wi-Fi Bluetooth® Pioneer Kit)								
CYW9P62S1-43012EVB-01 (PSOC™ 62S1 Wi-Fi Bluetooth® Pioneer Kit)								
CY8CKIT-064B0S2-4343W (PSOC™ 64 Wi-Fi Bluetooth® "Secure Boot" Pioneer Kit)								
CY8CKIT-064S0S2-4343W (PSOC™ 64 "Standard Secure" - AWS Wi-Fi Bluetooth® Pioneer Kit)								
KIT_T2G-B-E_LITE (TRAVEO™ T2G Body Entry Lite evaluation kit)								
KIT_T2G-B-E_EVK								
CY8CEVAL-062S2 (PSOC™ 62S2 Evaluation Kit)								
KIT_XMC72_EVK (XMC7200 evaluation kit)								
KIT_T2G-B-H-16M_LITE (TRAVEO™ T2G BODY HIGH LITE KIT)								
CYW955X13EVK-01 (AIROC™ CYW55X13 1x1 Wi-Fi 6E and Bluetooth® Low Energy 5.4)	One	A	A	A	A	A	A	N/A
KIT_PSE84_EVAL (PSOC™ Edge E84 Evaluation Kit)	One	A	A	A	A	N/A	N/A	N/A
KIT_PSE846_EVAL	One	A	A	A	A	N/A	N/A	A

3.2 Operating Speeds

Table 3-2. KitProg3 Operating Speeds

Functionality	Supported Speed	Units	Comments
SWD Programming	100 – 16000	kHz	Minimum validated speed is 100 kHz. Lower speeds are accepted by the adapter but are not validated.
JTAG Programming	100 – 1600	kHz	Minimum validated speed is 100 kHz. Lower speeds are accepted by the adapter but are not validated.
USB-UART Bridge [1] [2]	300, 600, 1200, 1800, 2400, 4800, 7200, 9600, 14400, 19200, 28800, 33600, 38400, 57600, 115200, 128000, 230400, 250000, 256000, 460800, 500000, 921600, 1000000, 1152000, 1500000, 1600000, 2000000, 3000000, 4000000	Baud	• Data bits – 8, Parity – None, • Stop bits – 1 (This is configured by the KitProg3 and cannot be changed by the user)
Half-Duplex UART Mode [3] [4]	19200, 28800, 33600, 38400, 57600, 115200, 128000, 230400, 250000, 256000, 460800, 500000, 1000000	Baud	• MiniProg4 only. • Data bits – 8, Parity – None, Stop bits – 1.
USB-I^{2}C Bridge	50, 100, 400, 1000	kHz	Minimum 50 kHz. Only the listed speeds can be selected.
USB-SPI Bridge	50–6000	kHz	Minimum 50 kHz. Available only on kits that support USB-SPI; see Kit Support.

1 The following baud rates might not be supported on macOS if using a custom communication library: 250000, 500000, 1000000, 1152000, 1500000, 1600000, 2000000, 3000000, 4000000.

2 Standard UART clock tolerance is in range of 3.9%, in some cases deviation between target clock and KitProg3 UART clock might be bigger and result in data loss. To avoid such issue, append data frames with at least 1-bit interval in target application.

3 A delay between transmit and receive operations is recommended.

4 Baud rates 28800, 33600, 128000, 250000, and 256000 are not supported on Ubuntu.

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™ software
- Any IDE that supports CMSIS-DAP Bulk protocol

4.1 Supported Kits

The following table shows KitProg3-based kits that have DAPLink mode available.

Table 4-1. Kits with DAPLink Support

Development Kits
CY8CKIT-062-BLE PSOC™ 6 Bluetooth® LE Pioneer Kit
CY8CKIT-062-WiFi-BT PSOC™ 6 Wi-Fi-Bluetooth® Pioneer Kit
CY8CPROTO-062S2-4343W PSOC™ 6 Wi-Fi Bluetooth® Prototyping Kit
CY8CPROTO-064S1-SB PSOC™ 64 "Secure Boot" Prototyping Kit
CY8CKIT-062S2-43012 PSOC™ 62S2 Wi-Fi Bluetooth® Pioneer Kit
CY8CPROTO-062S3-4343W PSOC™ 62S3 Wi-Fi Bluetooth® Prototyping Kit
CYW9P62S1-43438EVB-01 PSOC™ 62S1 Wi-Fi Bluetooth® Pioneer Kit
CYW9P62S1-43012EVB-01 PSOC™ 62S1 Wi-Fi Bluetooth® Pioneer Kit
CY8CKIT-005 MiniProg4 Program and Debug Kit
CY8CPROTO-064B0S3 PSOC™ 64 "Secure Boot" Prototyping Kit
CY8CPROTO-064B0S1-BLE PSOC™ 64 Bluetooth® LE "Secure Boot" Prototyping Kit
CY8CKIT-064B0S2-4343W PSOC™ 64 Wi-Fi Bluetooth® "Secure Boot" Pioneer Kit
CY8CKIT-064S0S2-4343W PSOC™ 64 "Standard Secure" – AWS Wi-Fi Bluetooth® Pioneer Kit
CYSBSYSKIT-DEV-01 Rapid IoT Connect Developer Kit
CY8CKIT-062S2-AI PSOC™ 6 AI Evaluation Kit
PSOC™ 4100T Plus CAPSENSE™ Prototyping Kit CY8CPROTO-041TP
CY8CPROTO-040T-MS PSOC™ 4000T Multi-Sense Prototyping 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 Using DAPLink

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

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™
- µVision
- IAR Workbench

4.3.4 User Interface

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

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

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.

4.4.3 Half-Duplex UART Mode

KitProg3 provides a Half-Duplex UART mode to communicate with target devices that use a single wire for both transmitting and receiving data.

4.4.3.1 Hardware Support and Limitations

The Half-Duplex UART functionality has the following limitations:

- Hardware: It is supported exclusively on MiniProg4 hardware.
- Port: It is only available on the primary UART port.
- Baud Rate: It supports baud rates in the range of 19200 to 1,000,000 bps.

4.4.3.2 Wiring and Implementation

In this configuration, the MiniProg4's separate RX and TX lines should be connected to the single RX_TX UART pin on the target board. The firmware manages the direction of data flow on this shared line.

4.4.3.3 Communication Delay and supported Baud Rates

For UART Half-Duplex mode, a delay is recommended between transmit and receive operations.

DAPLink Mode

Table 4-3. Supported baud rates for UART Half-Duplex mode

Baud Rate (bps)	Windows	MacOS	Ubuntu
19200	+	+	+
28800	+	+	-*
33600	+	+	-*
38400	+	+	+
57600	+	+	+
115200	+	+	+
128000	+	+	-*
230400	+	+	+
250000	+	+	-*
256000	+	+	-*
460800	+	+	+
500000	+	+	+
1,000,000	+	+	+

Note: ** - KitProg3 is not compatible with this baud rate on Ubuntu.*

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™ IAR Embedded Workbench Keil µVision	ModusToolbox™ PSOC™ Creator IAR Embedded Workbench Keil µVision	PSOC™ Creator
Programmer Support	N/A	ModusToolbox™ Programmer PSOC™ Programmer	PSOC™ Programmer
Kit support	See Arm Mbed Enabled™ Kit Support	The same, see Supported Kits	

Note: The Bridge Control Panel is not supported by ModusToolbox™ 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)
- ModusToolbox™ 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 a Programmer tool. 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 a Programmer tool 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 Programmer tool log window provides progress information and confirms connection to the new KitProg3 firmware on the kit.

Figure 5-1. Upgrading to KitProg3 in ModusToolbox™ Programmer

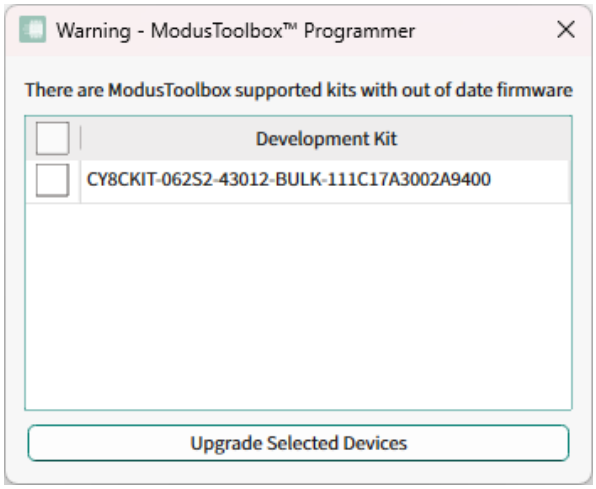
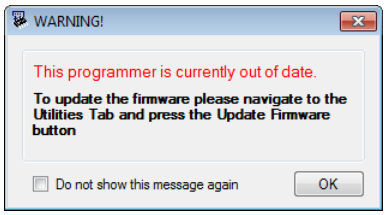


Figure 5-2. KitProg3 Firmware Update Warning in PSOC™ Programmer



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 tool 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 the GitHub repository. Download the appropriate zip archive for your OS. This tool is also installed with the ModusToolbox tools package installer.

Note: *Support for macOS on Intel-based architectures is not provided.*

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 fw-loader tool documentation

The Firmware Loader tool user guide can be found in the "docs" directory where the fw-loader package is installed. Refer to user guide for more information on supported command line options and examples of usage.

7 Troubleshooting

This section lists known issues, along with any workarounds.

Wrong COM port name() for KitProg3-based devices with AIROC™ targets on Windows

COM ports for a KitProg3-based device with AIROC targets may be wrongly recognized in the system for two reasons:

- Driver is not up-to-date - COM port(s) is/are listed in the Device Manager as a "USB Serial Device", instead of expected "HCI UART"/"Peripheral UART" name(s).

  Solution is to manually update driver(s) for each COM port to the latest one, which is delivered with the latest version of fw-loader tool (See Where to get it section).

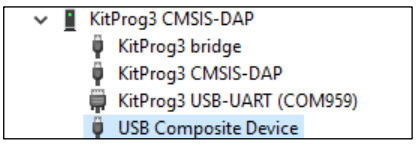
  In the Device Manager, go to the **USB Serial Device** device(s) and click on **Update driver > Browse my computer for drivers** and navigate to the kp-firmware\drivers\KitProg3\KitProg3UART folder, delivered with fw-loader tool. Click **Next** and **Install**.
- KitProg3 FW on the device is not up-to-date - DualUart COM ports are detected as "KitProg3 Primary USB-UART" and "KitProg3 Secondary USB-UART" instead of "HCI UART" and "Peripheral UART". In bulk/HID modes, the COM port is listed as "KitProg3 USB-UART" instead of the expected "HCI UART."

  Solution is to update the device FW to the latest one (See Updating KitProg3 section).

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.



2. Right click on **USB Composite Device** and choose **Uninstall device**.
3. Click on **Scan for hardware changes button**.

Troubleshooting

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.

On Windows 10 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. If I switch to HID mode, it works.

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.

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 check box **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 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 ModusToolbox™ Programmer, which automatically updates the KitProg3 firmware.

  OR

- Launch PSOC™ Programmer, and update KitProg3 firmware via **Utilities** > **Update firmware** option.

Revision history

Revision	Date	Description of Change
**	2018-10-26	New kit guide.
*A	2018-11-08	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. KitProg3 vs. KitProg2. Updated description. Updated Troubleshooting. Updated description.
*B	2018-11-22	Added PSOC™ Creator/PSOC™ Programmer information
*C	2019-02-19	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	2019-02-21	Remove listing of an unsupported kit
*E	2019-05-29	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	2019-07-24	Minor edits.
*G	2019-09-26	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.

Revision history

Revision	Date	Description of Change
*H	2019-12-12	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	2020-03-18	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	2020-06-11	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	2020-07-28	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	2020-08-19	Added footnotes about Serial Port limitation on macOS in section 3.2 and section 4.3.2
*M	2021-01-29	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.

Revision history

Revision	Date	Description of Change
*N	2021-04-20	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
*O	2021-08-12	Updated list of supported devices in section 3.1 Updated list of supported baud rates in section 3.2 Updated footnote about Serial Port limitation using custom communicational library on macOS in section 3.2. Added footnote about intervals between UART frames in section 3.2 Updated mode switching from DAPlink to Uartx2 mode in section 2.2.3.1 Added information about switching to Uartx2 mode via firmware loader tool in section 6.1.3
*P	2022-05-02	Added note for DAPLink mode availability in the next release in section 1 Updated sections 3.1 and 2.2.5.2 with information about usage of GPIO Bridging feature Added section 6.1.3 Documentation on fw-loader tool Removing references to Mbed OS, including KitProg3 High-Level Architecture in Figure 1-1. Updated list of supported devices in section 3.1.
*Q	2023-04-13	Updated Troubleshooting section with possible COM port issues and ways to solve them Added info on UART flow configuration in section 2.2.5.1 Updated Kit Support table with availability of the UART HW Flow Control feature support in section 3.1 Removed info regarding deprecated support of Mbed CLI from Troubleshooting section
*R	2024-04-26	Removed info regarding no longer supported Windows 7 from Troubleshooting section Updated section 2.2.5.1 USB-UART Bridge Feature Updated and reformatted list of supported devices in section 3.1 Updated links/references to and screenshots of ModusToolbox™ Programmer throughout document
*S	2024-10-28	Updated list of supported kits in section 3.1 and section 4.1.
*T	2025-02-21	Fixed font size in Kit-Support 3.1 table in section 3.1 Updated list of supported kits in section 3.1 and section 4.1 Fixed command typos in section 2.2.5.2
*U	2025-04-22	Updated list of supported kits in section 3.1 Updated Fig. 2.1 and 5.1

Revision history

Revision	Date	Description of Change
*V	2026-03-11	Added info on UART Half-Duplex configuration in sections 2.2.3.2, 4.3.3
*W	2026-06-05	Clarified info on UART Half-Duplex configuration in section 4.3.3.2 Updated list of supported kits in section 3.1
*X	2026-09-23	Added SOM Identification section 2.2.6. Updated Kit Support table with availability of the SOM identification feature in section 3.1. Updated Table 1-2 with the SOM identification feature. Added minimum supported speeds for SWD, JTAG, UART, I2C and SPI interfaces.

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

Edition 2026-09-23
Published by
Infineon Technologies AG
81726 Munich, Germany

© 2026 Infineon Technologies AG. All Rights Reserved.

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

Document reference
002-24616 Rev. *X

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

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

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

The data contained in this document is exclusively intended for technically trained staff. It is the responsibility of customer's technical departments to evaluate the suitability of the product for the intended application and the completeness of the product information given in this document with respect to such application.

For further information on the product, technology, delivery terms and conditions and prices please contact your nearest Infineon Technologies office (**www.infineon.com**).

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

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