

# ModusToolbox™

## Device Firmware Update Host tool guide

### About this document

#### Version

1.50

#### Scope and purpose

The Device Firmware Update (DFU) Host tool is a stand-alone program included with the ModusToolbox™ software. This tool is used to communicate with a PSoC™ 6 or PSoC™ 4 MCU that has already been programmed with an application that includes the DFU capability.

#### Intended audience

This document helps application developers understand how to use the DFU Host tool as part of creating a ModusToolbox™ application.

#### Document conventions

Convention	Explanation
<b>Bold</b>	Emphasizes heading levels, column headings, menus and sub-menus
<i>Italics</i>	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
<b>File &gt; New</b>	Indicates that a cascading sub-menu opens when you select a menu item

#### Reference documents

Refer to the following documents for more information as needed:

- [Eclipse IDE for ModusToolbox™ user guide](#)
- Device datasheets
- Device technical reference manuals
- [KitProg3 user guide](#)

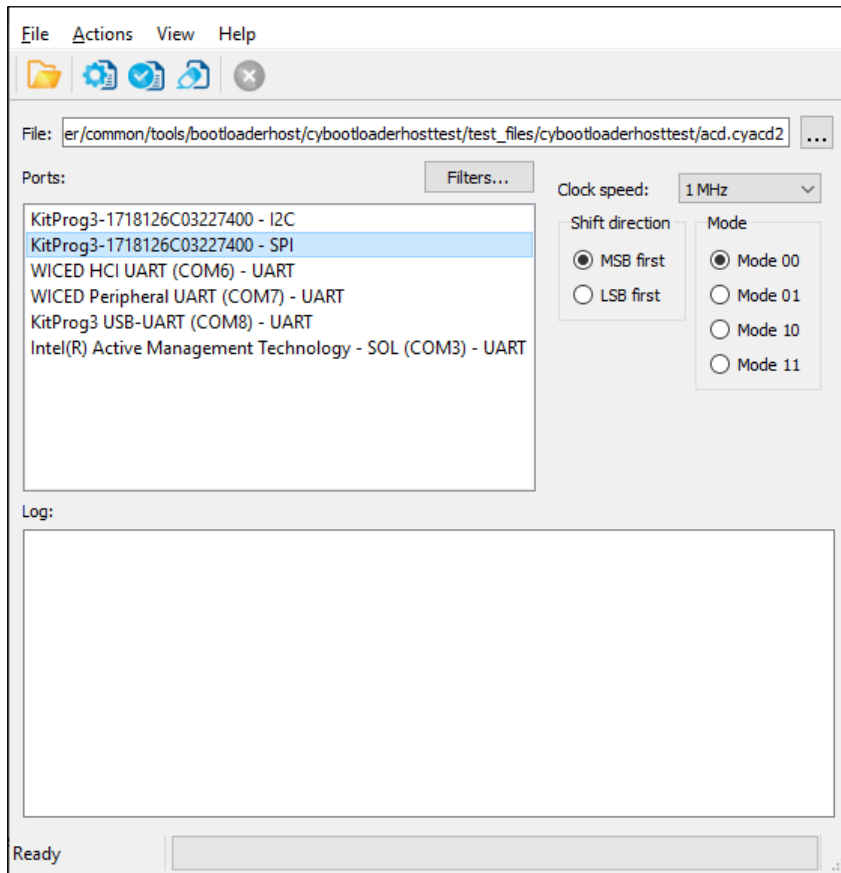
**Table of contents**

- 1 Overview..... 3**
- 2 Example code ..... 4**
- 3 Walkthrough..... 5**
- 4 Launch the DFU Host tool ..... 6**
  - 4.1 make command .....6
  - 4.2 Eclipse IDE .....6
  - 4.3 Executable (GUI).....6
  - 4.4 Executable (CLI).....6
- 5 GUI description ..... 7**
  - 5.1 Main window .....7
  - 5.2 Port configuration .....8
  - 5.3 Log .....9
  - 5.4 Errors .....9
- 6 CLI description.....10**
  - 6.1 Command-line flags ..... 11
  - 6.2 CLI example ..... 11
- 7 Troubleshooting .....12**
- 8 Version changes .....13**

Overview

# 1 Overview

The DFU Host tool is provided as a [graphical user interface \(GUI\)](#) and a [command-line interface \(CLI\)](#).



This tool allows you to:

- Program new application data onto a PSoC™ 6 or PSoC™ 4 MCU
- Verify the program data that is already contained on the device
- Erase the application from the device
- Select a \*.cyacd2 file output from a ModusToolbox™ application for programming
- Abort an operation.

*Note: This operation leaves the device in whatever state it is in when the abort message is acted upon.*

The DFU Host tool supports communicating via I<sup>2</sup>C, SPI, and UART. They are displayed to configure their settings and view the programming status. For I<sup>2</sup>C and SPI, PSoC™ 6 and PSoC™ 4 kits include the KitProg3 firmware module, which implements USB-UART, USB-I<sup>2</sup>C, and USB-SPI bridges. For UART, communication can be done directly from the PC simply by connecting an appropriate cable.

*Note: The process of initializing the device, two CPUs therein, and executing code in the SROM and supervisory flash, is referred to as “bootloading.” The process of installing and updating applications in the field is referred to as “device firmware update”. This process uses standard communication channels (UART, I<sup>2</sup>C, USB, etc.) to download new applications from a host.*

**Example code**

## 2 Example code

We provide source code for the DFU Host tool in the installation directory, as follows:

- **Windows/Linux:** `~/ModusToolbox/tools_2.3/dfuh-tool/sample_code/`
- **Linux:** `~/ModusToolbox/tools_2.3/dfuh-tool/share/dfuh-tool/sample_code/`
- **macOS:** `/Applications/ModusToolbox/tools_2.3/dfuh-tool/dfuh-tool.app/Contents/sample_code/`

Use this source code as a reference to build your own tool for your needs. Refer to the `CMakeLists.txt` file for instructions on how to build the example.

*Note:* This example requires CMake 3.12 or later.

### 3 Walkthrough

This section contains a basic walkthrough using the DFU Host tool.

1. First, make sure you have a PSoC™ MCU that has been programmed with an application that includes the DFU capability.
2. [Launch the DFU Host tool GUI.](#)
3. In the [graphical DFU Host tool](#), click **File > Open** to browse to the location of your application file (\*.cyacd).
4. Connect a hardware port that supports the communication selected in your bootloader project.
5. Wire the hardware port pins to the corresponding pins on the target device.
6. Update the [port configuration](#). These values are set from the communication component used by the bootloader component.

After updating configuration information, click **Program** to load the new application.

## 4 Launch the DFU Host tool

There are several ways to launch the DFU Host tool, and those ways depend on how you use the various tools in ModusToolbox™ software.

### 4.1 make command

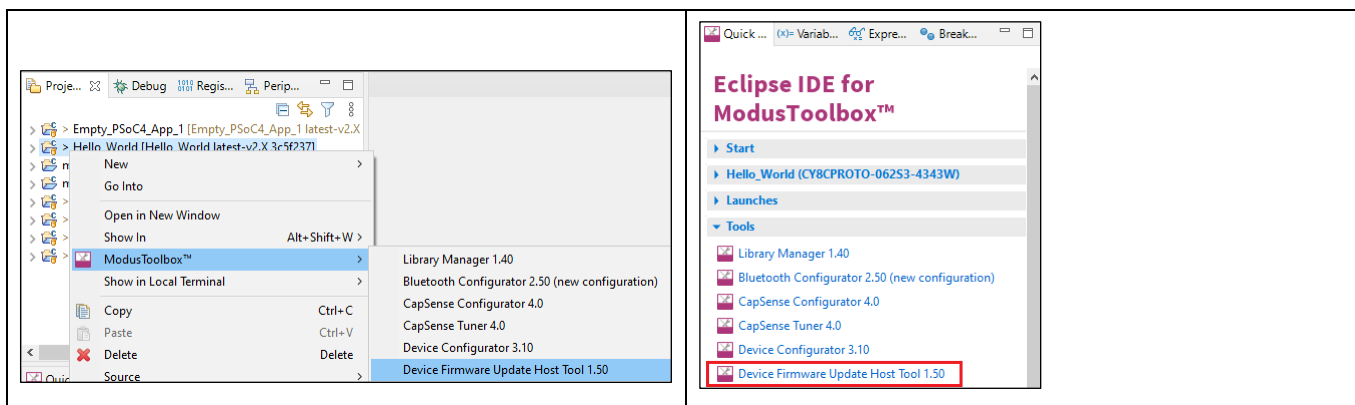
As described in the [ModusToolbox™ user guide "Build System"](#) chapter, you can run numerous make commands in the application directory, such as launching the DFU Host tool. After you have created a ModusToolbox™ application, navigate to the application directory and type the following command in the appropriate bash terminal window:

```
make open CY_OPEN_TYPE=dfuh-tool
```

This command opens the DFU Host tool GUI for the specific application in which you are working.

### 4.2 Eclipse IDE

If the selected Eclipse IDE application is suitable to use the DFU Host tool, right-click on the appropriate project and select **ModusToolbox™ > Device Firmware Update Tool**. You can also click the DFU Host tool link in the IDE Quick Panel.



### 4.3 Executable (GUI)

You can launch the DFU Host tool as a stand-alone tool without the Eclipse IDE. By default, it is installed here:

```
<install_dir>/ModusToolbox_<version>/tools_<version>/dfuh-tool
```

On Windows, you can launch the tool from the **Start** menu. For other operating systems, the installation directory will vary, based on how the software was installed.

### 4.4 Executable (CLI)

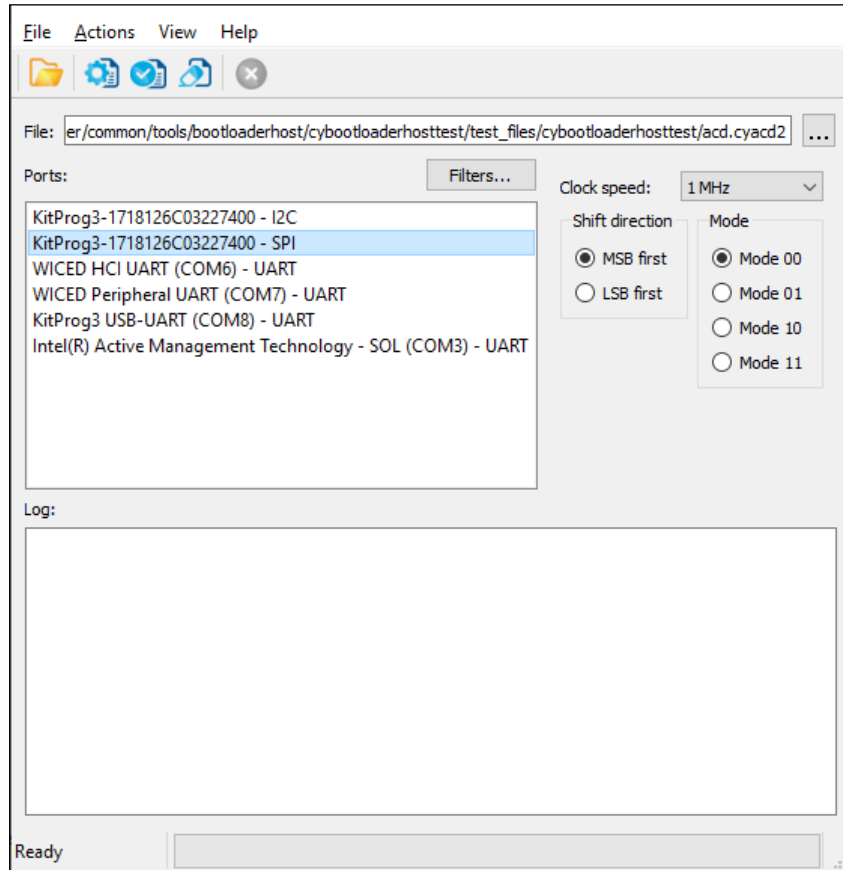
Refer to [CLI Description](#) to run the *dfuh-cli* tool. You can also run the DFU Host tool GUI executable using the following command line arguments:

-?, -h, --help	Displays information about all valid command-line arguments and exits.
-v, --version	Displays version information and exits.
--debug <filename>	Appends detailed debugging information to the specified file.

GUI description

## 5 GUI description

### 5.1 Main window



#### 5.1.1 File selection

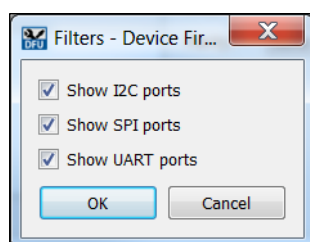
Use this section to select the cyacd2 file that contains the DFU image.

#### 5.1.2 Ports

This is a list of all the ports attached to the computer that can be used to communicate with the bootloader. Based on which item is selected, different options are available for the Port Configuration.

#### 5.1.3 Filters

The **Filters** button allows for restricting which ports are displayed in the ports list.



## GUI description

## 5.2 Port configuration

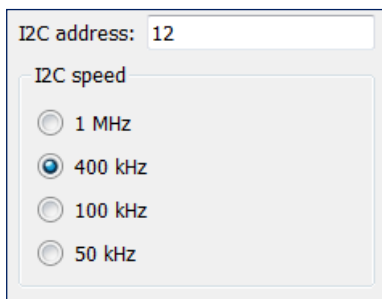
This section allows for configuring the interface-specific options for communicating with the DFU system. This is necessary to ensure both the DFU and host computer are configured the same.

Refer to the appropriate probe documentation for a list of supported modes.

*Note: Not all SPI and UART communication properties combinations are supported.*

### 5.2.1 I<sup>2</sup>C

For I<sup>2</sup>C communication, there are two pieces of required information:

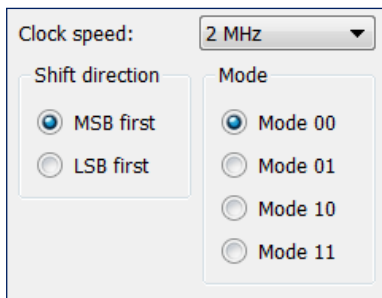


The screenshot shows a configuration window for I<sup>2</sup>C. It features a text input field for 'I2C address' containing the value '12'. Below it is a section titled 'I2C speed' with four radio button options: '1 MHz', '400 kHz' (which is selected), '100 kHz', and '50 kHz'.

- The address of the I<sup>2</sup>C-based target DFU system with which the host is communicating. The range for valid addresses is from 8 - 120.
- The I<sup>2</sup>C SCK signal frequency.

### 5.2.2 SPI

For SPI communication, there are three pieces of required information:



The screenshot shows a configuration window for SPI. It features a dropdown menu for 'Clock speed' set to '2 MHz'. Below it are two sections: 'Shift direction' with radio buttons for 'MSB first' (selected) and 'LSB first'; and 'Mode' with radio buttons for 'Mode 00' (selected), 'Mode 01', 'Mode 10', and 'Mode 11'.

- The SPI SCLK signal frequency.
- The bit ordering of transferred data.
- The SPI operating mode.



## GUI description

### 5.2.3 UART

For UART communication, there are four pieces of required information:

Baud:	115200 ▼
Data bits:	8 ▼
Stop bits:	1 ▼
Parity:	None ▼

- The baud (bit) rate at which data is transferred.
- The number of data bits per byte.
- The number of stop bits indicating the termination of a byte.
- The parity bit that is added to a byte.

### 5.3 Log

The log displays:

- history of what happened while the Host was open
- information about user-initiated operations
- when items started/completed
- the error messages during an operation if any.

### 5.4 Errors

Any errors for various fields display as a red X in the field containing the error, and it contains a tooltip when you hover the mouse cursor on it.

## CLI description

## 6 CLI description

In addition to the *dfuh-tool* GUI executable, there is also a *dfuh-cli* executable. The CLI allows programming, verifying, and erasing of devices from a command-line prompt or from within batch files or shell scripts. The exit code for the *dfuh-cli* executable is zero if the operation is successful, or non-zero if the operation encounters an error.

In order to use the *dfuh-cli* executable, you must provide exactly one of the following flags:

<code>--program-device &lt;cyacd2_file&gt;</code>	Program the device with the specified file and exit.
<code>--verify-device &lt;cyacd2_file&gt;</code>	Verify the programming of the device with the specified file and exit.
<code>--erase-device &lt;cyacd2_file&gt;</code>	Erase the specified program from the device and exit.

If there is more than one device connected to the host, use the following flag to specify which device to use:

<code>--hwid &lt;string&gt;</code>	Specifies the ID of the hardware to program/verify/erase. If this option is skipped, the first appropriate device found will be used.
------------------------------------	---

In addition, you must provide the appropriate configuration values for exactly one of the following protocols:

The **I<sup>2</sup>C** flags are:

<code>--i2c-address &lt;int&gt;</code>	Sets the address for the I2C protocol. Valid values are between 8 (0x08) and 120 (0x78).
<code>--i2c-speed &lt;int&gt;</code>	Sets the speed for the I2C protocol in kHz. Common values are 50, 100, 400 and 1000.

The **SPI** flags are:

<code>--spi-clockspeed &lt;float&gt;</code>	Sets the clock speed for the SPI protocol in MHz.
<code>--spi-mode &lt;int&gt;</code>	Sets the mode for the SPI protocol in binary. Valid values are 00, 01, 10 and 11.
<code>--spi-lsb-first</code>	Specifies that the least-significant bit should be sent first for the SPI protocol. Otherwise, the most-significant bit will be sent first.

The **UART** flags are:

<code>--uart-baudrate &lt;int&gt;</code>	Sets the baud rate for the UART protocol.
<code>--uart-databits &lt;int&gt;</code>	Sets the number of data bits for the UART protocol.
<code>--uart-paritytype &lt;string&gt;</code>	Sets the parity type for the UART protocol. Valid strings are “None”, “Odd” and “Even”.
<code>--uart-stopbits &lt;float&gt;</code>	Sets the stop bits for the UART protocol. Valid values are 1, 1.5 and 2.

## CLI description

## 6.1 Command-line flags

The following flags change the overall functioning of the tool:

-?, -h, --help	Displays information about all valid command-line arguments and exits.
-v, --version	Displays version information and exits.
--debug	Outputs debugging information to the terminal running the CLI tool during programming, verifying or erasing.
--display-hw	Outputs all compatible hardware attached to the computer and exits.

## 6.2 CLI example

The following shows a simple example for using the *dfu-cli* executable

```
dfuh-cli --program-device test_app.cyacd2 --i2c-address 8 --i2c-speed 100
```

## 7 Troubleshooting

Problem	Workaround
<p>On common Linux distributions, the serial UART ports (usually /dev/ttySx or /dev/ttyUSBx devices) belong to the root user and to the dialout group. Standard users are not allowed to access these devices.</p>	<p>An easy way to allow the current user access to the Linux machine's serial ports is by adding the user to the dialout group. This can be done using the following command:</p> <pre data-bbox="603 450 1139 479">\$sudo usermod -a -G dialout \$USER</pre> <p><i>Note:</i> For this command to take effect, you must log out and then log back in.</p>
<p>On Linux, attempts to set up or start DFU Host tool communication causes the tool to close without any messages.</p>	<p>To enable DFU Host tool communication under Linux, install the udev rules for KitProg3:            Disconnect the KitProg device.            Execute in the terminal (root access required):</p> <pre data-bbox="603 792 1461 822">sh \$CYSDK/tools/fw-loader/udev_rules/install_rules.sh</pre> <p>Reconnect the KitProg device.</p>
<p>On common Linux distributions, the DFU Host tool forbids communication protocol selection after re-plugging KitProg during communication.</p>	<p>Refer to the “Installation Procedure on Ubuntu Linux (x64)” section in the CYPRESS™ Programmer 2.1 CLI User Guide.</p>
<p>KitProg3 UART is accessible but not able to read data on Linux kernel 4.15 and above or Mac OS X 10.13 and above.</p>	<p>Use a third-party UART to USB bridge.            Update the KitProg3 firmware to version 1.11.243 or above.</p>
<p>After updating firmware and middleware for your application, SPI transfer speed is not as fast as expected.</p>	<p>You may be able to improve performance by modifying the <code>src/backend/cychannelspi.cpp</code> file. Remove the calls to <code>QThread::msleep(1)</code> when building your bootloader host tool.</p>

## 8 Version changes

This section lists and describes the changes for each version of this tool.

Version	Change Descriptions
1.0	New tool.
1.1	Added Notice List. Added command-line interface (CLI) and handling of invalid command line arguments. Added logging for firmware update process.
1.2	Removed Notice List.
1.30	Updated versioning to support patches.
1.40	Fixed minor defects.
1.50	Added PSoC™ 4 device support.

## Revision history

Revision	Date	Description
**	11/09/2018	New document.
*A	10/16/2019	Updated to version 1.1.
*B	03/27/2020	Updated to version 1.2.
*C	09/01/2020	Updated to version 1.30.
*D	10/16/2020	Added information about code example.
*E	03/11/2021	Updated to version 1.40.
*F	09/20/2021	Updated to version 1.50.

**Trademarks**

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

**Edition 2021-09-20**

**Published by**

**Infineon Technologies AG**

**81726 Munich, Germany**

**© 2021 Infineon Technologies AG.**

**All Rights Reserved.**

**Do you have a question about this document?**

**Email: [erratum@infineon.com](mailto:erratum@infineon.com)**

**Document reference**

**002-25258 Rev. \*F**

**IMPORTANT NOTICE**

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

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](http://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.