ModusToolbox®
Device Firmware Update Host Tool

Version 1.40

Overview

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 MCU that has already been programmed with an application that includes the DFU capability. This tool is provided as a graphical user interface (GUI) and a command-line interface (CLI). The DFU Host tool allows you to:

- Program new application data onto a PSoC 6 MCU
- Verify the program data that is already contained on the device
- Erase the application from the device
- Select a *.cyacd2 file output from ModusToolbox 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^{2}C, SPI, and UART. They are displayed to configure their settings and view the programming status. For I^{2}C and SPI, PSoC 6 kits include the KitProg3 firmware module, which implements USB-UART, USB-I^{2}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^{2}C, USB, etc.) to download new applications from a host.

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.

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.

Launch the DFU Host Tool GUI

You can launch the DFU Host tool various ways: as a stand-alone tool, from the Eclipse IDE for ModusToolbox or from the command line.

As a Stand-Alone Tool

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.

From the Eclipse IDE

If your Eclipse IDE application is suitable to use the DFU Host tool, right-click on the appropriate project and select **ModusToolbox > Device Firmware Update Tool**.

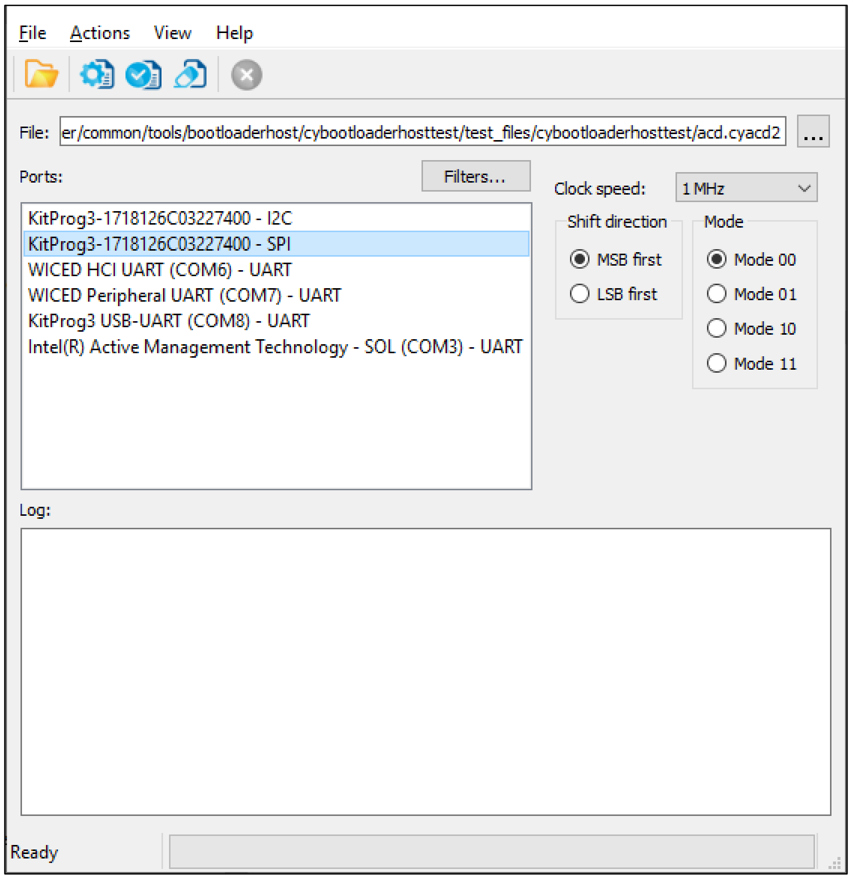
From the Command Line

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

Main window



File: er/common/tools/bootloaderhost/cybootloaderhosttest/test_files/cybootloaderhosttest/acd.cyacd2

Ports:

KitProg3-1718126C03227400 - I2C
KitProg3-1718126C03227400 - SPI
WICED HCI UART (COM6) - UART
WICED Peripheral UART (COM7) - UART
KitProg3 USB-UART (COM8) - UART
Intel(R) Active Management Technology - SOL (COM3) - UART

Clock speed: 1 MHz

Shift direction: [✓ MSB first] LSB first

Mode: [✓ Mode 00] Mode 01 Mode 10 Mode 11

Log:

Ready

File Selection

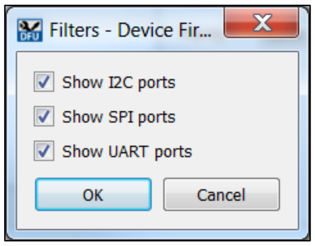
Use this section to select the cyacd2 file that contains the DFU image.

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.

Filters

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



[✓] Show I2C ports
[✓] Show SPI ports
[✓] Show UART ports

Port Configuration

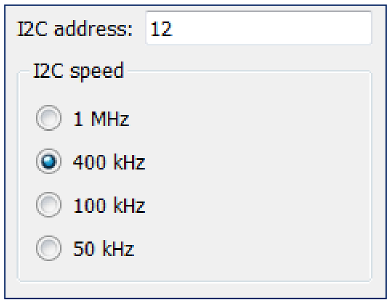
This section allows for configuring the interface-specific options for communicating with the DFU system. This is necessary to ensure both the DFU and host computer are configured the same.

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

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

I²C

For I²C communication, there are two pieces of required information:



I2C address: 12

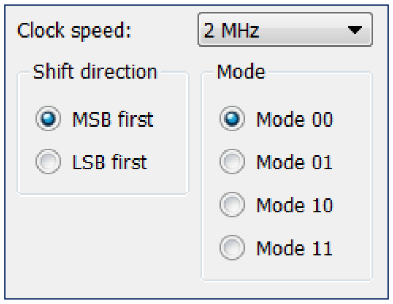
I2C speed

( ) 1 MHz
(●) 400 kHz
( ) 100 kHz
( ) 50 kHz

- The address of the I²C-based target DFU system with which the host is communicating. The range for valid addresses is from 8 -120.
- The I²C SCK signal frequency.

SPI

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



Clock speed: 2 MHz

Shift direction

(●) MSB first
( ) LSB first

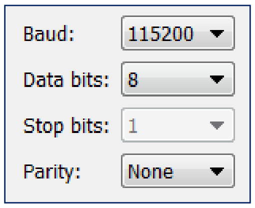
Mode

(●) Mode 00
( ) Mode 01
( ) Mode 10
( ) Mode 11

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

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.

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.

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

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:

`--program-device <cyacd2_file>`	Program the device with the specified file and exit.
`--verify-device <cyacd2_file>`	Verify the programming of the device with the specified file and exit.
`--erase-device <cyacd2_file>`	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:

`--hwid <string>`	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 **I2C** flags are:

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

The **SPI** flags are:

`--spi-clockspeed <float>`	Sets the clock speed for the SPI protocol in MHz.
`--spi-mode <int>`	Sets the mode for the SPI protocol in binary. Valid values are 00, 01, 10 and 11.
`--spi-lsb-first`	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:

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

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.

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
```

Troubleshooting

Problem	Workaround
On common Linux distributions, the serial UART ports (usually /dev/ttySx or /dev/ttyUSBx devices) belong to the root user and to the dialout group. Standard users are not allowed to access these devices.	An easy way to allow the current user access to the Linux machine's serial ports is by adding the user to the dialout group. This can be done using the following command: `$sudo usermod -a -G dialout $USER` **Note** For this command to take effect, you must log out and then log back in.
On Linux, attempts to set up or start DFU Host tool communication causes the tool to close without any messages.	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): `sh $CYSDK/tools/fw-loader/udev_rules/install_rules.sh` Reconnect the KitProg device.
On common Linux distributions, the DFU Host tool forbids communication protocol selection after re-plugging KitProg during communication.	Refer to the “Installation Procedure on Ubuntu Linux (x64)” section in the *Cypress Programmer 2.1 CLI User Guide*.
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.	• Use a third-party UART to USB bridge. • Update the KitProg3 firmware to version 1.11.243 or above.
After updating firmware and middleware for your application, SPI transfer speed is not as fast as expected.	You may be able to improve performance by modifying the *src/backend/cychannelspi.cpp* file. Remove the calls to `QThread::msleep(1)` when building your bootloader host tool.

References

Refer to the following documents for more information, as needed:

- Eclipse IDE for ModusToolbox User Guide
- Device Datasheets
- Device Technical Reference Manuals
- KitProg User Guide

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.