# ModusToolbox™ Programmer GUI user guide

**Version**

5.0.0

## About this document

### Scope and purpose

ModusToolbox™ Programmer is a stand-alone, cross-platform, flash programmer tool that provides a graphical user interface to Program, Erase, Verify, and Read the flash of the target device. It is delivered with the ModusToolbox™ Programming tools package, and it supports HEX, SREC, ELF, and BIN programming file formats.

### Intended audience

This document helps you learn how to use the ModusToolbox™ Programmer GUI to perform various operations on devices.

### Reference documents

Refer to the Infineon programming solutions website for more information as needed.

### 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 |

### Abbreviations and definitions

The following define the abbreviations and terms used in this document:

| Term | Description |
|---|---|
| CMSIS | Arm® Cortex® Microcontroller Software Interface Standard. |
| CMSIS-DAP | CMSIS Debug Access Port. |
| OpenOCD | The Open On-Chip Debugger is the debugger tool that provides on-chip programming support. This tool acts as a backend of the ModusToolbox™ Programmer application. |
| Data File | The data file for programming in the hex or binary format |
| DP | The Debug Port register of the Arm Cortex CPU. Used for programming and debugging, along with the corresponding SWD-address bit selections. |
| Flash kernel/loader | The firmware file loaded into the MCU's RAM. Sometimes referred to as RAM program, Flash kernel, Flash loader. |
| GDB | GNU Project Debugger – GNU.org. |

## About this document

| Term | Description |
|---|---|
| JTAG | Joint Test Action Group. Specifies the use of a dedicated debug port while implementing a serial communication interface for low-overhead access without requiring direct external access to the system address and data buses. |
| MCU | Microcontroller Unit. |
| PSoC™ | A family of microcontroller integrated circuits. These chips include a CPU core and mixed-signal arrays of configurable integrated analog and digital peripherals. |
| Region | Logical areas within the target device the programmer operates on. |
| SWD | Serial Wire Debug interface. |
| QSPI | Quad Serial Peripheral Interface. A name used for SPI external memory interfaces |

# Table of contents

**1 Overview ..... 4**

**2 Installing ModusToolbox™ Programmer ..... 5**

**3 Getting started ..... 6**

3.1 Run ModusToolbox™ Programmer ..... 6
3.2 Load programming file ..... 7
3.3 Connect device ..... 8
3.4 Program device ..... 9
3.5 Save log file ..... 9

**4 GUI description ..... 10**

4.1 Menus ..... 10
4.2 Toolbar ..... 12

**5 Programming Operations ..... 15**

5.1 Erase Device ..... 15
5.2 Program Device ..... 16
5.3 Program Device and Reset Chip ..... 17
5.4 Program Binary File with Offset ..... 18
5.5 Program External Memory ..... 19
5.6 Program PSoC™ 6 MCU in JTAG Chain ..... 21
5.7 Verify Device ..... 24
5.8 Verify Device with External Memory ..... 25
5.9 Verify Custom Flash Regions of PSoC™ 6 MCU ..... 27
5.10 Read Device ..... 29
5.11 Program eFuse Region of PSoC™ 6/TRAVEO™ T2G/XMC7xxx MCU ..... 31
5.12 Program PSoC™ 4 MCU With Protected Flash ..... 33
5.13 Program Chip-Protected PSoC™ 4 MCU ..... 34
5.14 Program Secure AIROC™ CYW20829 MCU ..... 35
5.15 Program QSPI memory with patched flashloader ..... 35

**6 Troubleshooting ..... 37**

6.1 Limitations ..... 37
6.2 How to Recover AIROC™ Bluetooth® Devices on Failure ..... 37

**7 Upgrading firmware ..... 38**

7.1 Upgrade KitProg2 firmware ..... 38
7.2 Upgrade KitProg3 on kit or MiniProg4 firmware ..... 39

# 1 Overview

ModusToolbox™ Programmer supports the following features:

- Programming Infineon MCUs' internal and external memories
  - PSoC™ 6 MCUs, including corresponding starter kits and evaluation boards
  - PSoC™ 4 MCUs, including corresponding starter kits and evaluation boards
  - PMG1 and WLC1 MCUs, including corresponding evaluation boards
  - EZ-PD™ CCG7S and CCG7D MCUs, CCG4, CCG3PA, CCG8
  - XMC7100/7200, CYT4BB/BF, CYT2Bx
- AIROC™ Wi-Fi/Bluetooth® platforms (CYW4390x, CYW4343W, CYW20829, CYW208xx, CYW55513 and others), including corresponding starter kits and evaluation boards
- Connectivity devices via support of development boards
- Windows, Linux, and macOS
- Programming external memory devices using PSoC™ 6 external memory interfaces (EBI / QSPI)
- Programming external memory of AIROC™ Wi-Fi devices
- KitProg3 and MiniProg4 hardware
- SEGGER J-Link Base and J-Link Ultra hardware
- OpenOCD via machine interface (MI) to 3rd party debug hardware
- Cross-platform Bridge library – $I^2C$, SPI, UART communications
- KitProg3 firmware update

# 2 Installing ModusToolbox™ Programmer

ModusToolbox™ Programmer is delivered as part of the ModusToolbox™ Programming tools package. A link is available to download/install from the IDC webpage here:

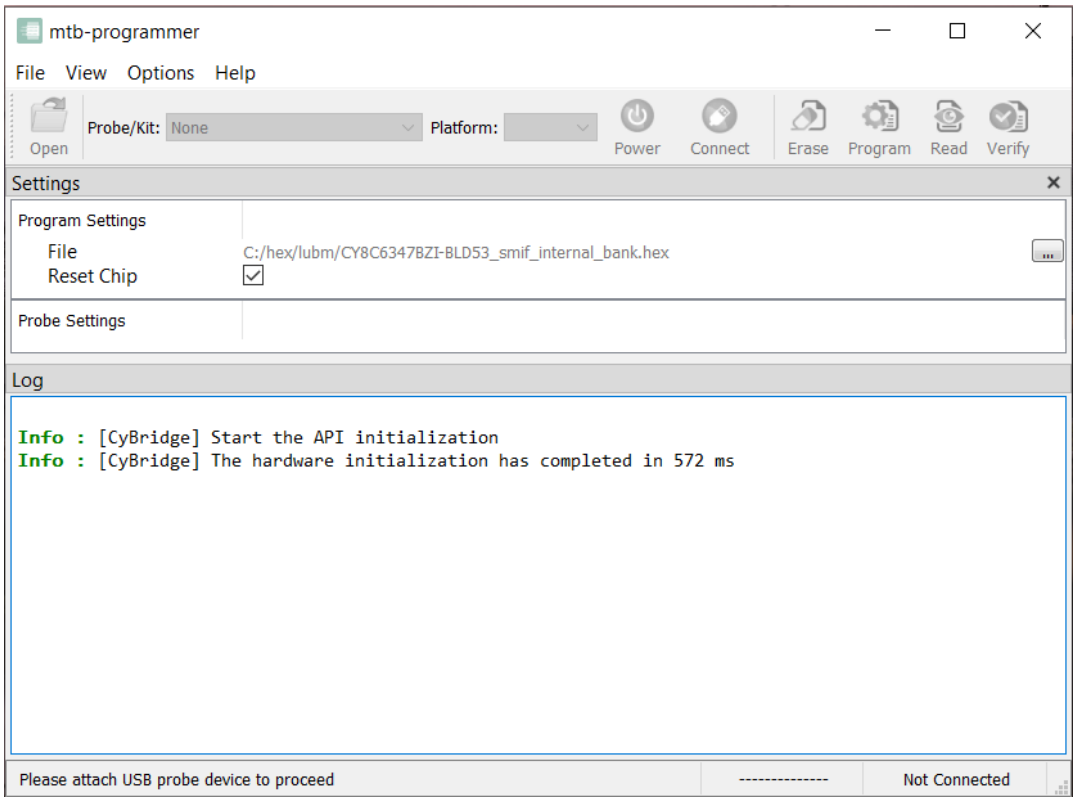
https://www.infineon.com/cms/en/design-support/tools/programming-testing/psoc-programming-solutions/

For installation details, see section 2 of the ModusToolbox™ Programming tools release notes.

# 3 Getting started

## 3.1 Run ModusToolbox™ Programmer

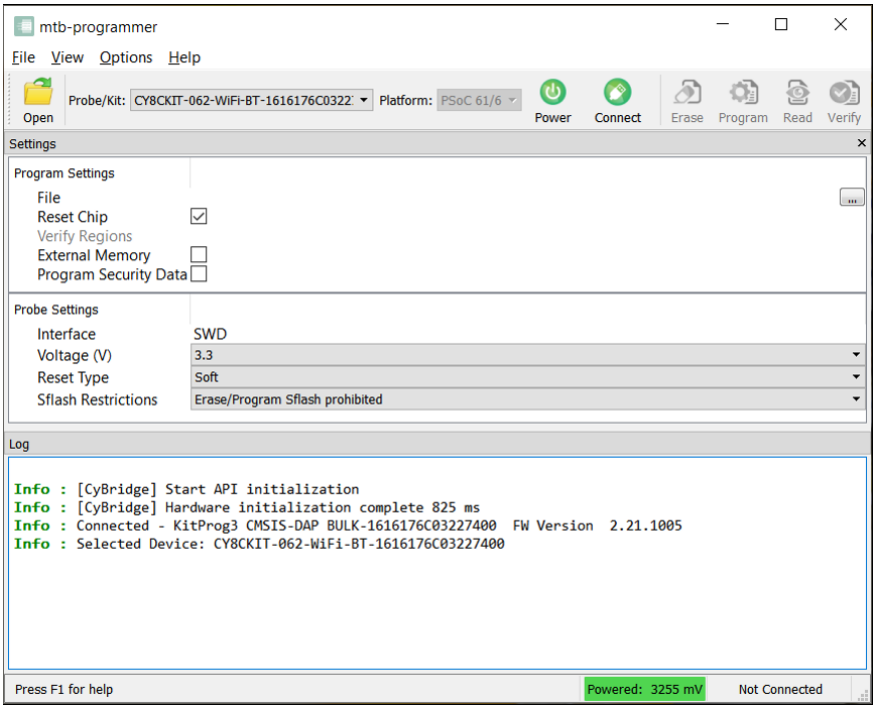
To run the ModusToolbox™ Programmer GUI application, navigate to the ModusToolbox™ Programming tools install location, open the *mtb-programmer* folder, and run the executable. See Installing ModusToolbox™ Programmer. The GUI opens and looks similar to this:



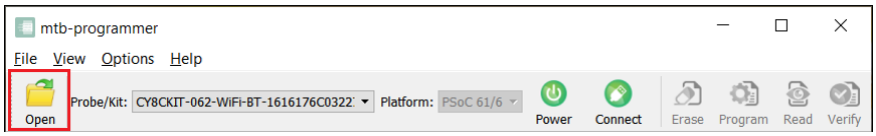

In this case, no kit or device is connected, and a message displays asking you to connect a device.

## 3.2 Load programming file

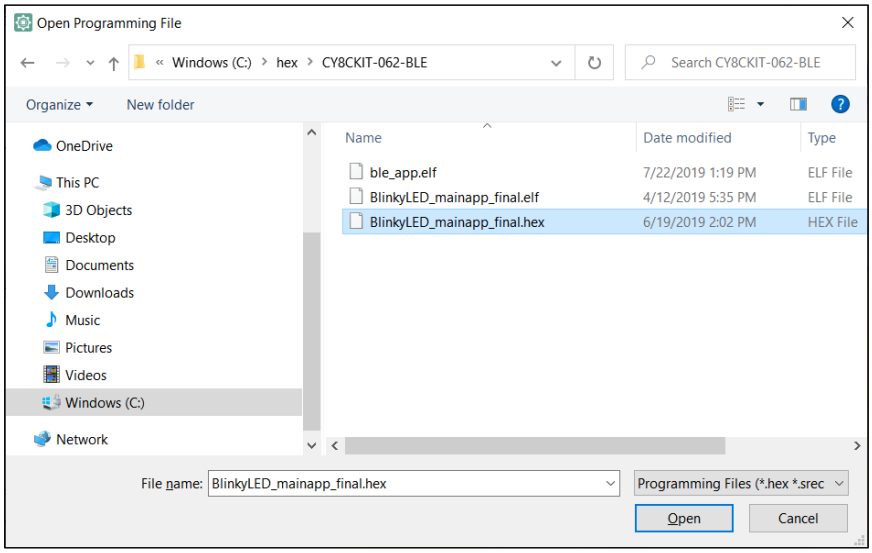
1. Connect the device to the host computer. Select the device name in the **Probe/Kit** drop-down, and ModusToolbox™ Programmer will display information under **Probe Settings** (if the **Settings** section is viewable).




2. Click **Open**.

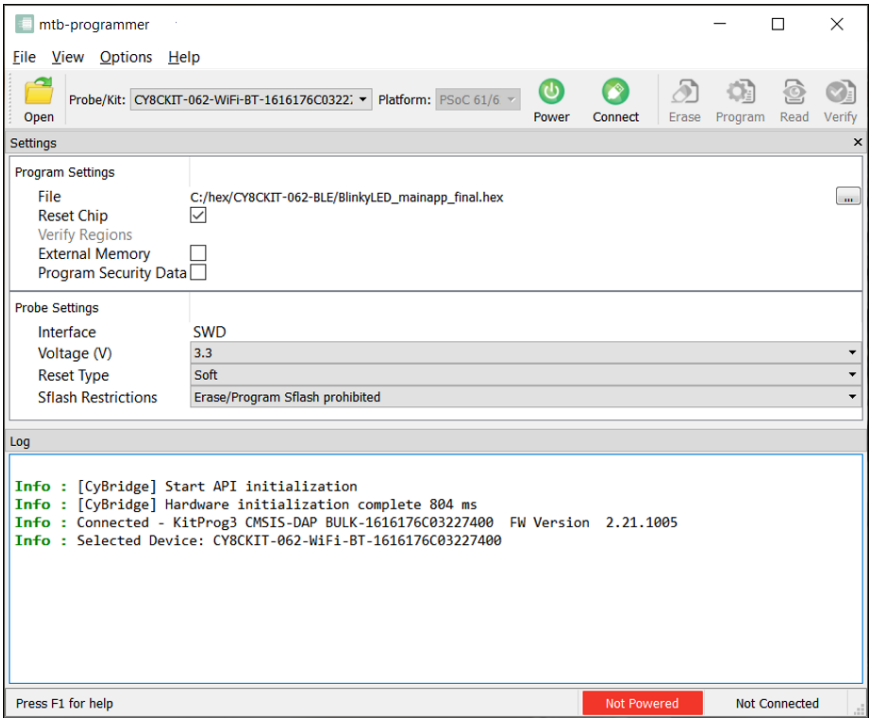



On the Open Programming File dialog, navigate to the location of the HEX, SREC, ELF, or BIN file to load, select it, and click **Open**.

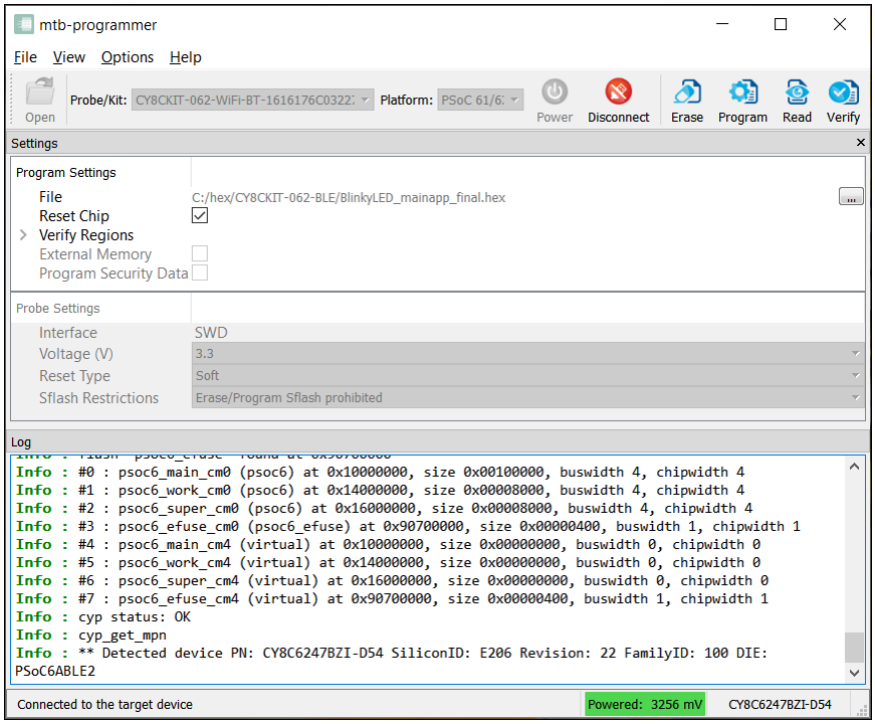


## 3.3 Connect device

1. If the device is not powered, the status message "Not Powered" is displayed in the Status Bar. Click **Power** to power up the device.

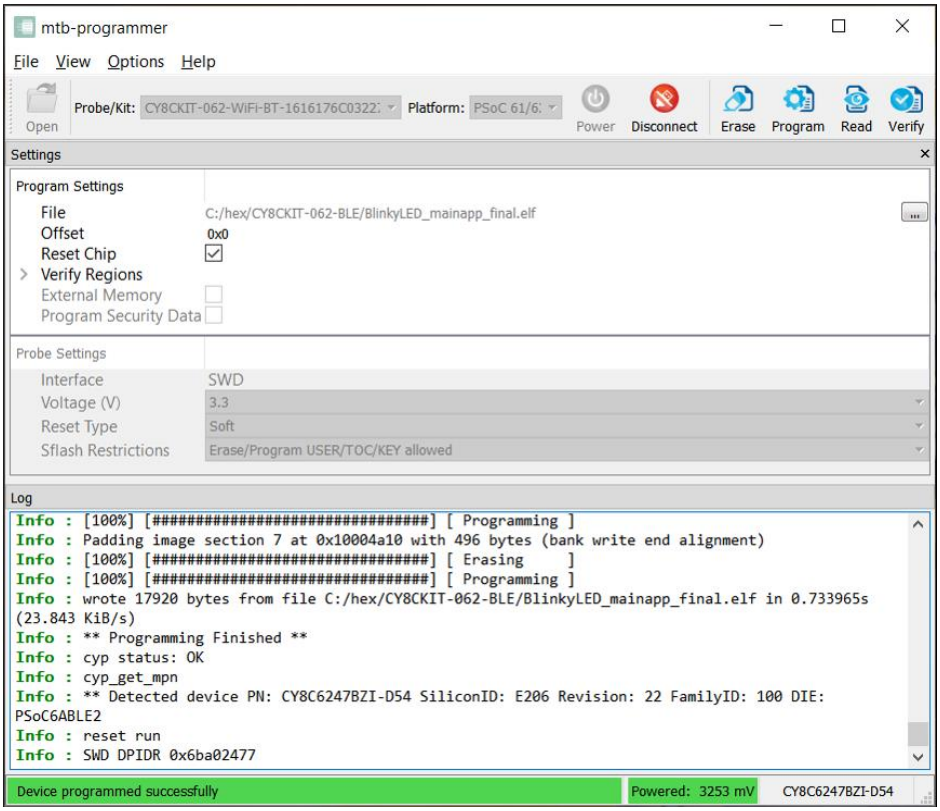



2. Click **Connect**. ModusToolbox™ Programmer communicates with the device and displays various messages in the **Log**. Then, a message in the Status Bar indicates that it is connected.



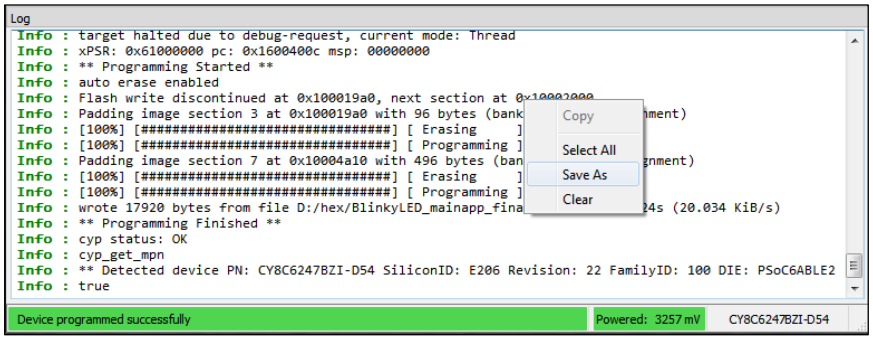
## 3.4 Program device

Click **Program**. ModusToolbox™ Programmer downloads the program file onto the device and displays messages in the **Log**.




## 3.5 Save log file

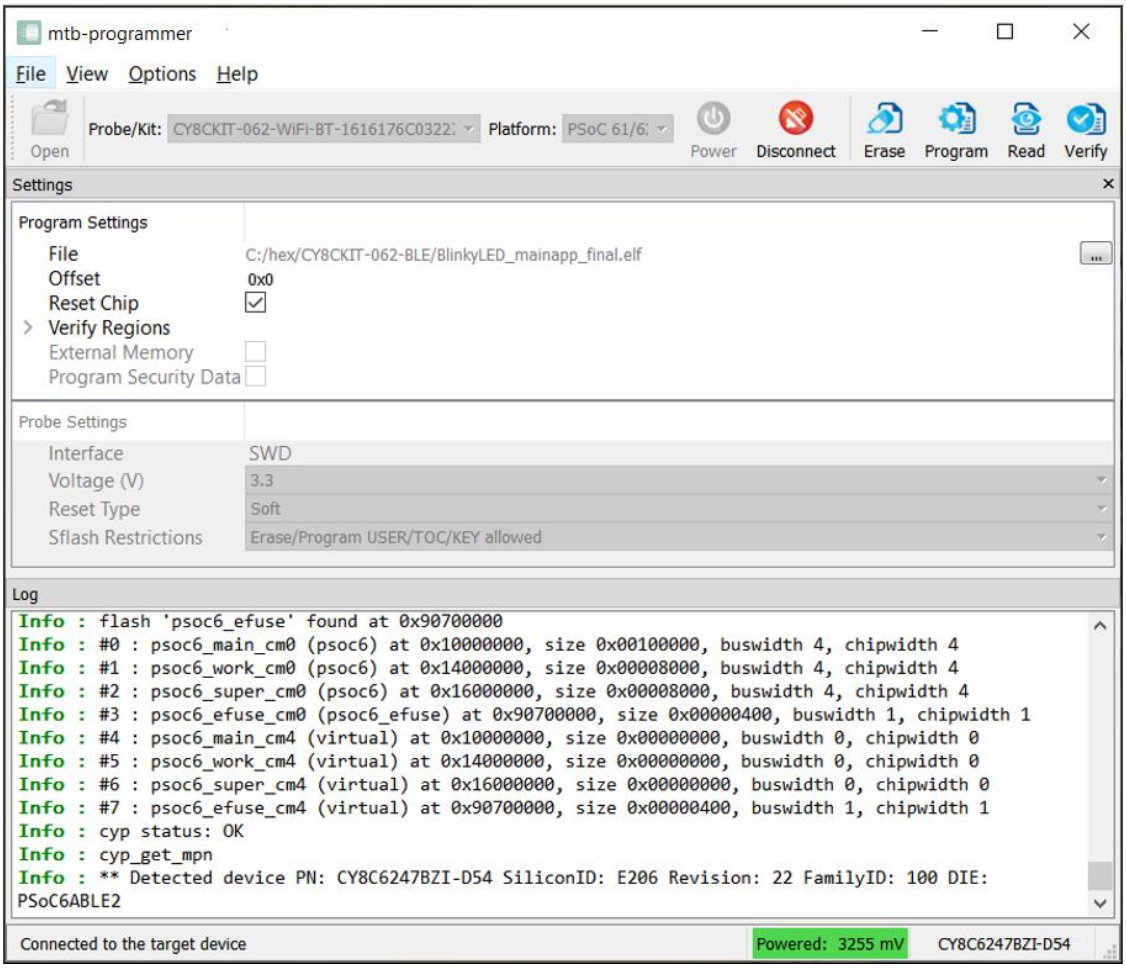
Right-click in the **Log** section and select **Save As**.




*Note:* *You can also select the **Select All** command to select the text, then copy and paste the text to the file you selected.*

# 4 GUI description

ModusToolbox™ Programmer contains menus and toolbar commands to perform actions. This chapter describes the various GUI elements.




## 4.1 Menus

### 4.1.1 File

The **File** menu contains the following commands:

- **Open (Ctrl+O)** – Opens the programming file.
- **Connect/Disconnect (Alt+Q)** – Connects and disconnects the selected device.
- **Program (Alt+G)** – Programs the selected device with the selected file.
- **Erase (Alt+E)** – Erases the selected device.
- **Read (Alt+R)** – Reads flash of the selected device into a HEX or SREC file.
- **Verify (Alt+Y)** – Verifies that the selected device is programmed correctly.
- **Recent Files** – Lists up to five recently loaded programming files.
- **Exit (Alt+F4)** – Closes the ModusToolbox™ Programmer application.

### 4.1.2 View

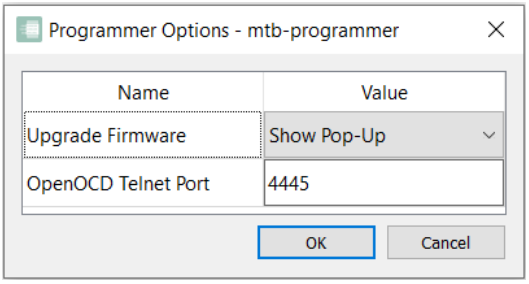
The **View** menu contains the **Settings** check box. Select it to view the **Settings** section of the window; unselect it to hide the **Settings** section. See Settings.

### 4.1.3 Options

The **Options** menu contains the following commands:

#### 4.1.3.1 Programmer Options (Alt+T)

Opens the **Programmer Options** dialog to set the **Upgrade Firmware** mode and other options.




- **Upgrade Firmware** mode:
  - **Automatically** – The firmware is updated automatically when the tool opens.
  - **Show Pop-up** – A dialog displays asking if you want to upgrade the firmware.
  - **Ignore** – The firmware is not updated and no prompt displays.
- **OpenOCD Telnet Port**: This option specifies the port number of the OpenOCD telnet connection.

#### 4.1.3.2 Upgrade Firmware (Alt+U)

When this command is enabled, select it to upgrade the programmer firmware on the device.

### 4.1.4 Verify Regions

The **Verify Regions** menu is available only if **Verify Regions** option is selected in **Program Settings**.

- **Add Region –** Adds a custom flash region to the **Verify Regions** list.
- **Reload Regions –** Resets the **Verify Regions** list to the default state corresponding to the flash map of the target.
- **Undo (Ctrl+Z) –** Undo the last change in the **Verify Regions** list.
- **Redo (Ctrl+Y) –** Redo the last change in the **Verify Regions** list.
- **Verify –** Initiates the Verify device operation.

### 4.1.5 Help

The Help menu contains the following commands:

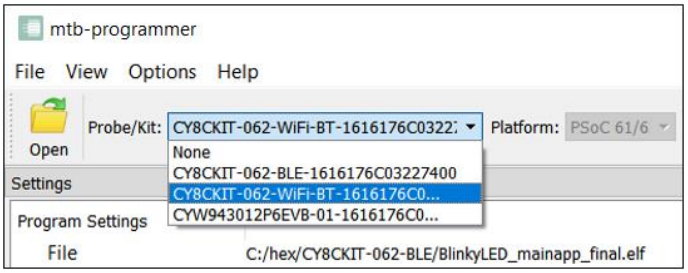
- **View Help (F1)** – Opens this document.
- **About mtb-programmer** – Opens the **About** box.

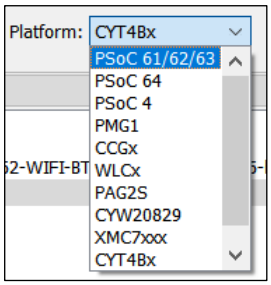
## 4.2 Toolbar

The toolbar contains the **Open, Connect, Erase, Program, Read,** and **Verify** commands, which are also located on the File menu. This area also contains the following:

### 4.2.1 Probe/kit and platforms

If you have more than one device connected to your computer, use these pull-down menus to select the specific probe and target platform to use.




### 4.2.2 Power

Use the **Power** button to power on and off the selected device.

### 4.2.3 Connect

Use the **Connect** button to connect to and disconnect from the selected device.

### 4.2.4 Settings

The **Settings** section of the tool allows you to update the program and probe/target settings as follows:

#### 4.2.4.1 Program Settings

- **File** – Use this to select the programming file to perform actions on or with.
- **Offset** – This is an optional offset parameter; it can be an integer or hexadecimal value. The relocation offset is added to the base address for each section in the image when the image is programmed. This option is visible only if a binary or elf file is selected for programming operations.
- **Reset Chip** – Use this to reset the chip after the Program operation completes. This option resets the target chip and runs the programmed firmware on it.
- **Verify Regions** – Use this option to define flash regions used during device verification. This allows to verify user defined flash regions of the PSoC™ MCU. This option does not change behavior of the Program operation. See Verify Custom Flash Regions of PSoC™ 6 MCU for details.
- **External Memory** – Enables/disables the programming of external memory in the target device. For PSoC™ 6x MCUs, this option enables programming of the QSPI regions. This is also used for programming external memory of AIROC™ Wi-Fi devices.
- **Program Security Data** – Allows programming security regions if the target device supports this capability. For example, for PSoC™ 61 PSoC™ 62, and PSoC™ 63 MCUs, this option enables programming the eFuse region.
- **Target AP** – Allows you to select the target access port (DAP) that will be used for programming. Possible values include: CM0, CM4, and SYS_AP. This option is available only for PSoC™ 64 MCUs.
- **Flash Size Limit** – Limits the size of application flash available for programming operations. This option is available only for PSoC™ 64 MCUs.
- **Programming Mode** – Use this option to define programming mode for PMG1 devices. The mode options include:
  - Reset: This programming mode enables acquisition of the target device in the Test mode.
  - PowerCycle: In this mode, the programmer cycles power to acquire the device.
- **Flashloader** – Use this option to select the patched QSPI CMSIS flashloader file (in FLM format). This flashloader is used for external flash programming.

*Note:* *To be able to program custom external flash you should also provide the appropriate QSPI configuration file (qspi_config.cfg), generated by the ModusToolbox™ QSPI Configurator tool. This file should be located in the same directory as the patched flashloader file.*

- **Debug Certificate** – Use this option to specify the location of the debug certificate binary file. The debug certificate is used for programming AIROC™ CYW20829 targets in Secure lifecycle mode.
- **ECC Config** – Use this option to enable or disable the ECC error reporting. This option is only applicable for read flash operations of some MCUs.

### 4.2.4.2 Probe Settings

The Probe settings allow you to configure the programming and target device before you connect to it. These settings are available when ModusToolbox™ Programmer is not connected to the device.

- **Interface** – To select the hardware (debug) interface for communication with the target device. The possible values include: SWD and JTAG if supported by the device.
- **JTAG Chain** – To select the interested target device in the JTAG chain. This option is only available for probes supporting JTAG interface. See Program PSoC™ 6 MCU in JTAG Chain for details.
- **Voltage (V)** –To select the power supply voltage of the target device in Volts. This option is available only if the selected probe has the power control capability.
- **Clock (KHz)** – To select the frequency of the hardware interface in KHz. This option is available only if the selected probe supports configurable frequencies.
- **Reset Type** – Specifies the type of the Reset Chip operation. The possible values include: Soft and XRES:
  - Soft is a software reset type that sends the system reset request to the ARM core.
  - XRES is a hardware reset type that toggles the XRES hardware line.
- **Sflash Restrictions** – Specifies the Sflash programming behavior. This option is available only for PSoC™ 61/62/63, XMC7100/7200, FX3G2, CYT4Bx and CYT2Bx MCUs. The possible values include:
  - Erase/Program of Sflash is prohibited.
  - Erase and Program of USER/TOC/KEY is allowed.
  - Erase of USER/TOC/KEY and Program of USER/TOC/KEY/NAR is allowed.
  - Erase and Program of entire Sflash is allowed.

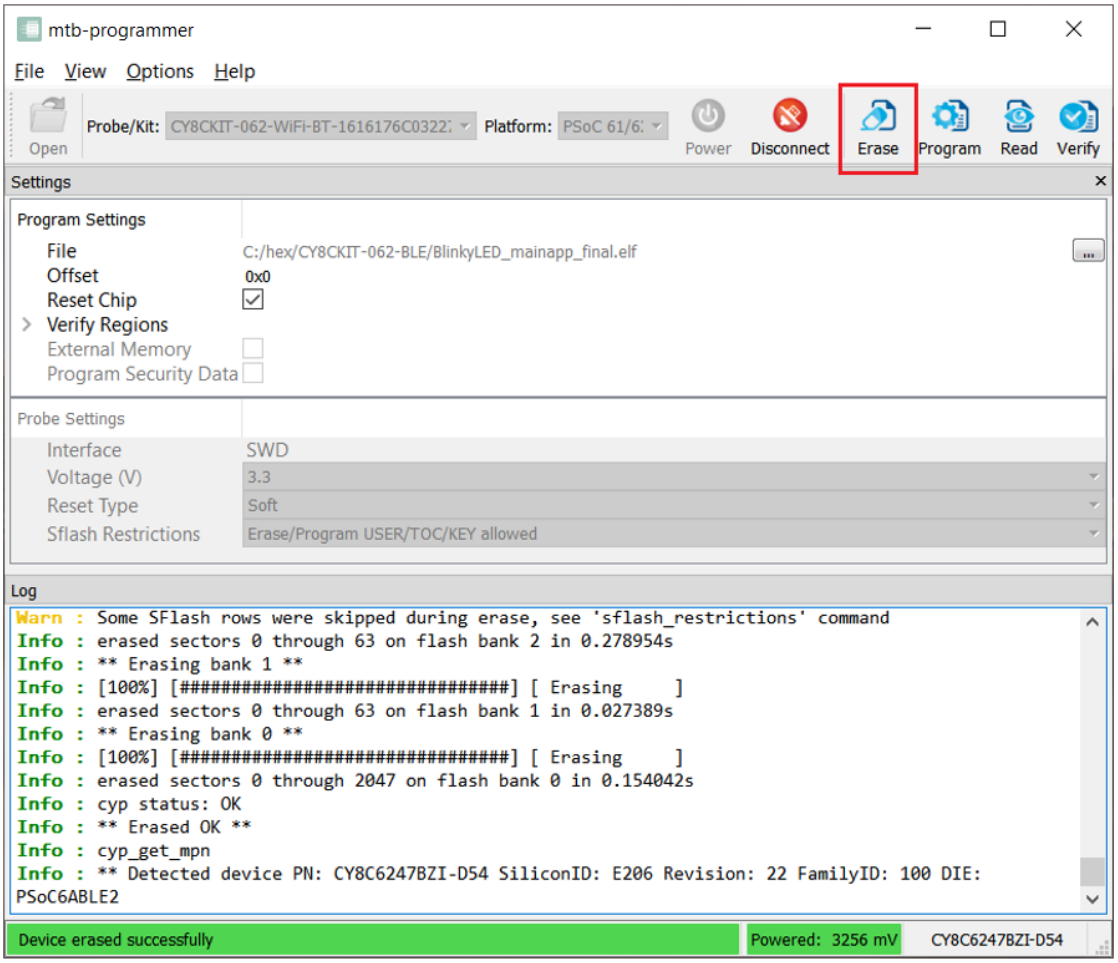
# 5 Programming Operations

This chapter covers the various programming operations you can perform using the ModusToolbox™ Programmer tool.

## 5.1 Erase Device

1. Connect to the device (see Connect Device).
2. Click the **Erase** button.

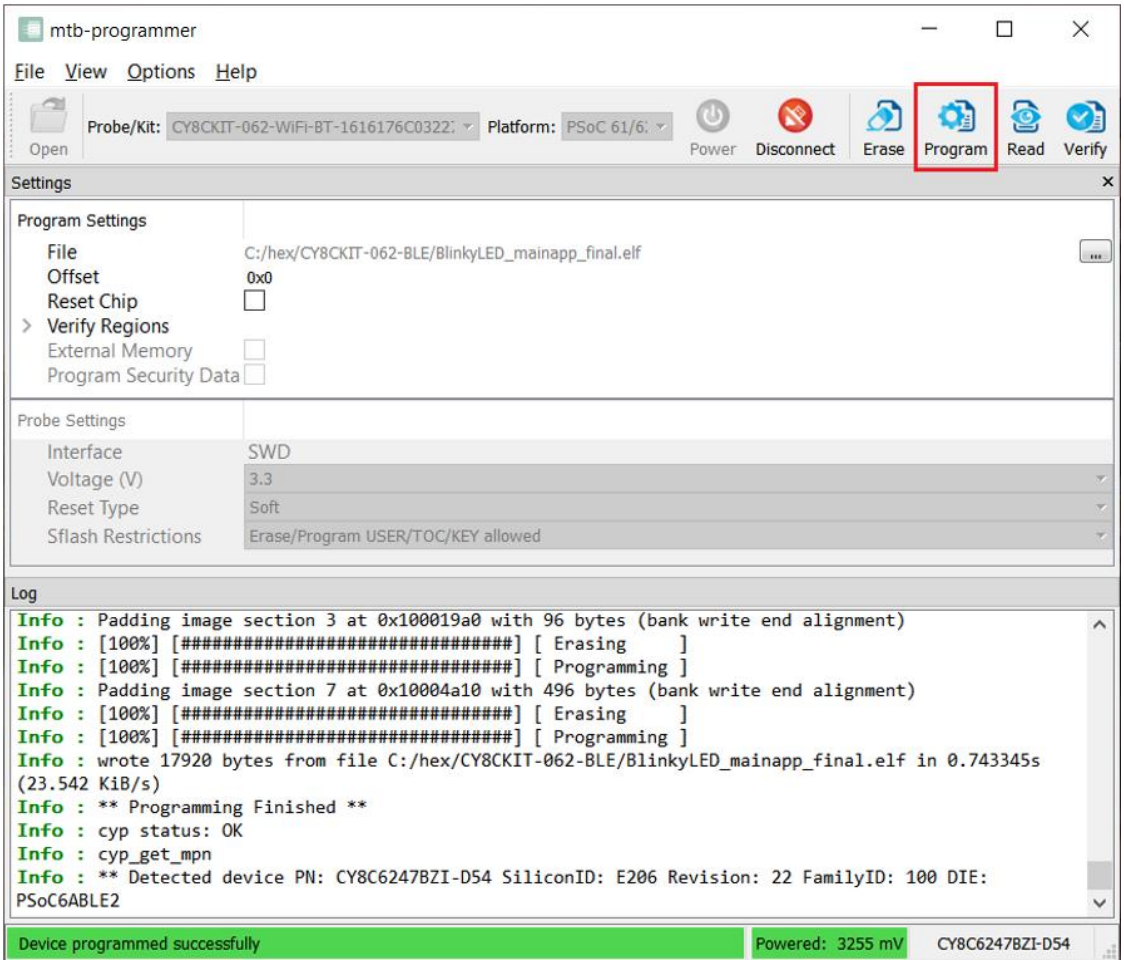
ModusToolbox™ Programmer erases the device and displays various messages in the **Log**. Then, a message in the Status Bar indicates that the device was erased successfully or that an error occurred.



## 5.2 Program Device

1. Connect the device to the host computer and select it in the **Probe/Kit** drop-down.
2. Select the programming file as described in Load Programming File section.
3. Connect to the device (see Connect Device).
4. Click the **Program** button.

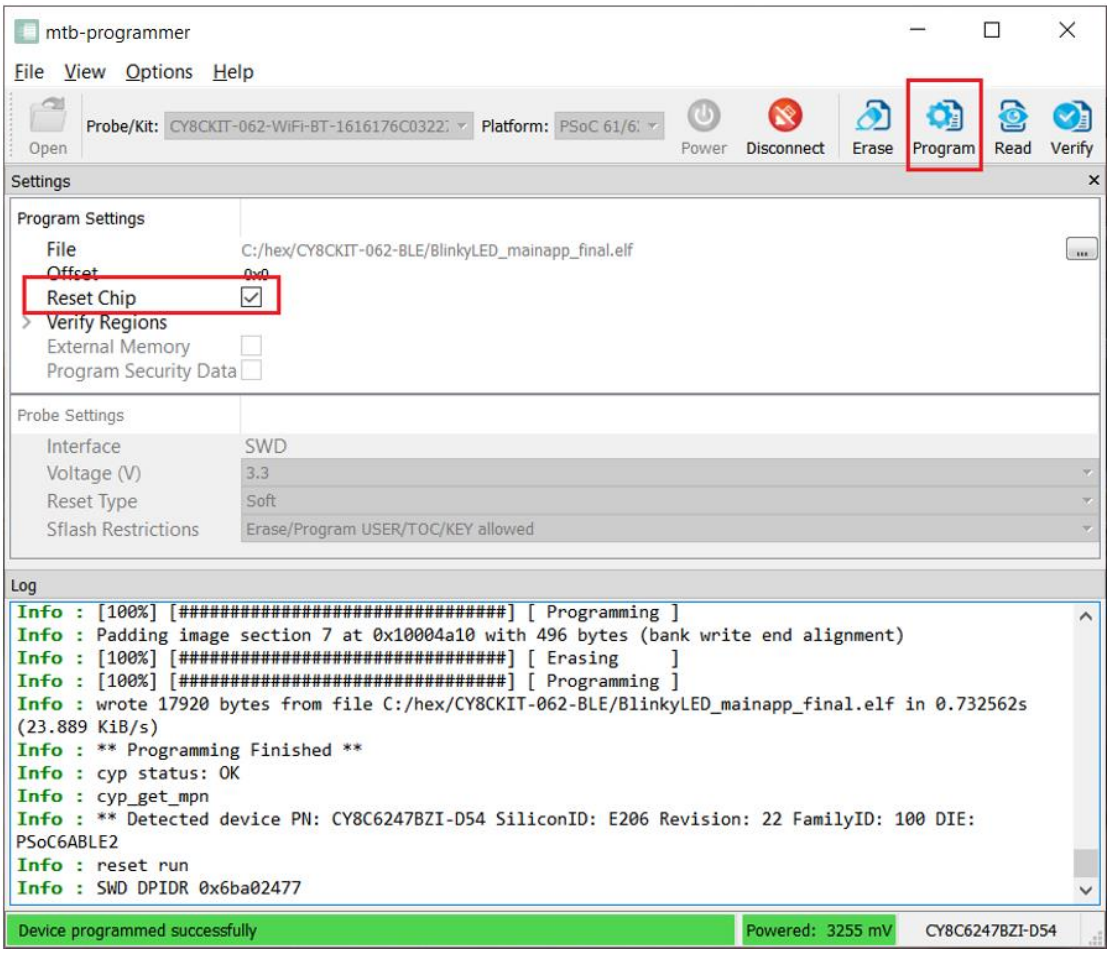
ModusToolbox™ Programmer programs the device and displays various messages in the **Log**. Then, a message in the Status Bar indicates that device was programmed successfully or that an error occurred.



## 5.3 Program Device and Reset Chip

1. Connect the device to the host computer and select it in the **Probe/Kit** drop-down.
2. Select the programming file as described in the Load Programming File section.
3. Connect to the device (see Connect Device).
4. Select the **Reset Chip** check box under **Program Settings**.
5. Click the **Program** button.

ModusToolbox™ Programmer programs the device and displays various messages in the **Log**. Then, a message in the Status Bar indicates that the device was programmed successfully or that an error occurred.

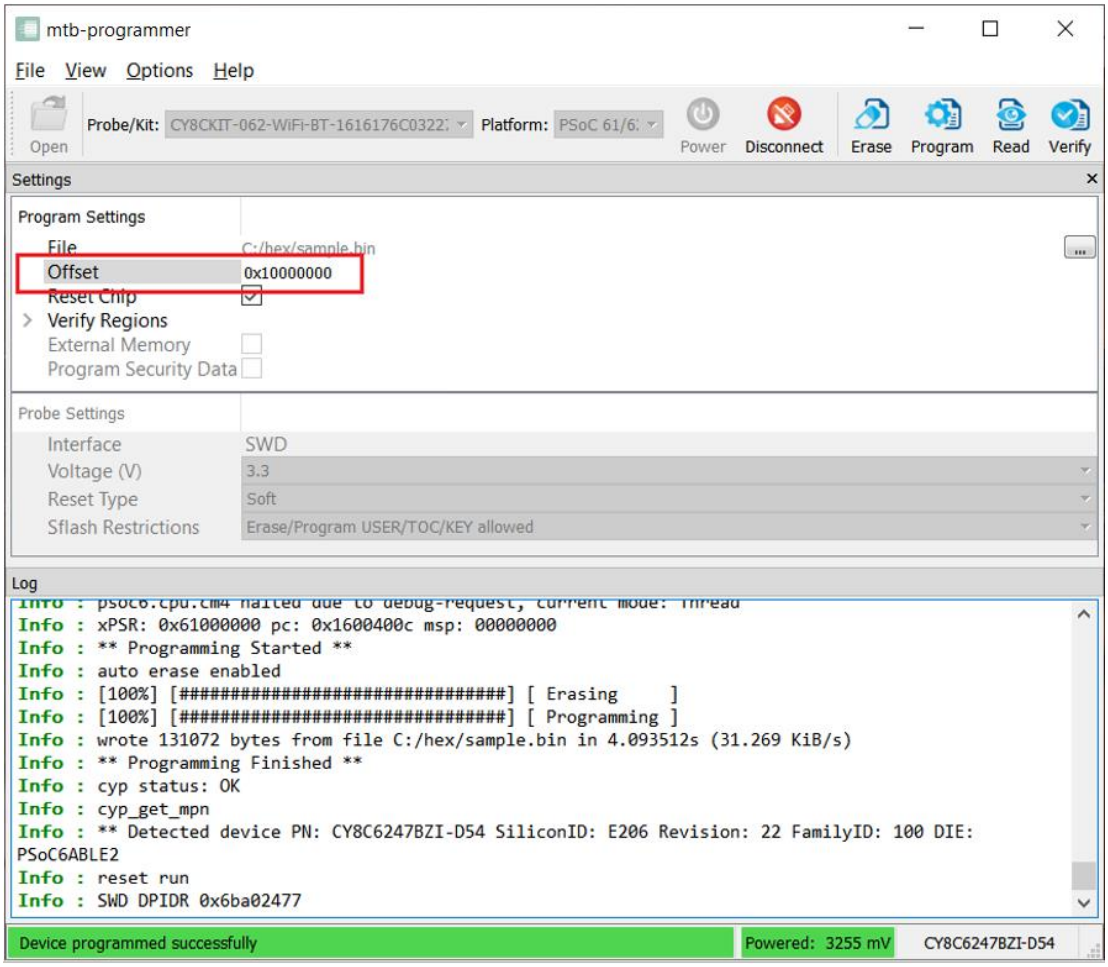



The target device is reset and running.

## 5.4 Program Binary File with Offset

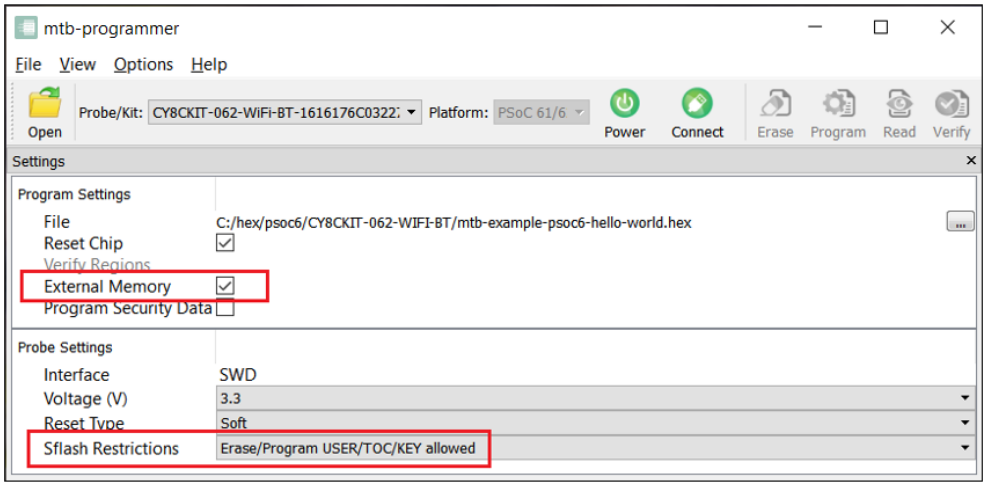
1. Connect the device to the host computer and select it in the **Probe/Kit** drop-down.
2. Select the binary programming file as described in the Load Programming File section.
3. Connect to the device (see Connect Device).
4. Enter the desired address in the **Offset** field under **Program Settings**.
5. Click the **Program** button.

ModusToolbox™ Programmer programs the device and displays various messages in the **Log**. Then, a message in the Status Bar indicates that the device was programmed successfully or that an error occurred.



## 5.5 Program External Memory

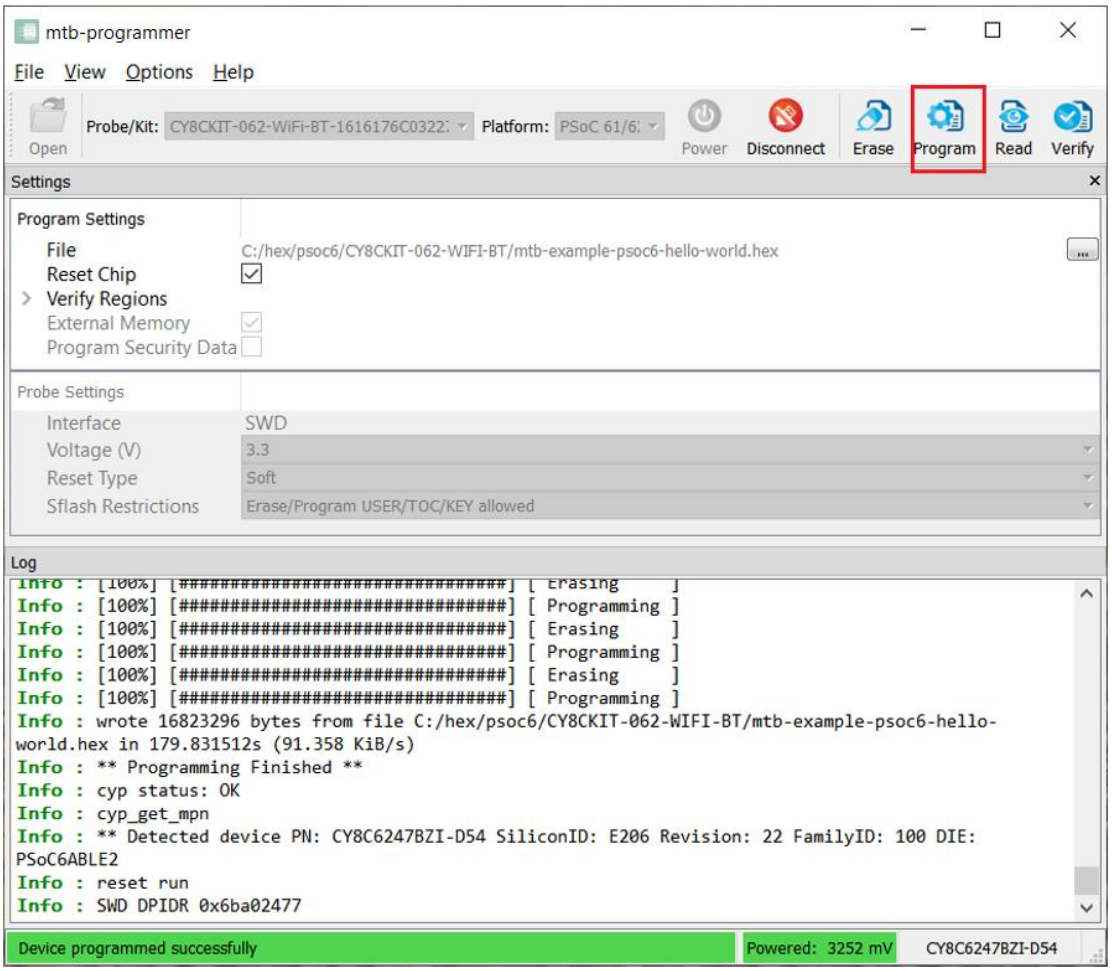
1. Attach and select a device that supports external memory (for example, CY8CKIT-062-WiFi-BT with QSPI support).
2. Select the **External Memory** option under **Program Settings**.




3. Select the programming file as described in the Load Programming File section. The programming file should contain external memory region(s) and correct QSPI configuration data.
4. Select **Erase/Program USER/TOC/KEY allowed** option under **Probe Settings > Sflash Restrictions.**
5. Connect to the device (see Connect Device).
6. Click the **Program** button.

## Programming Operations

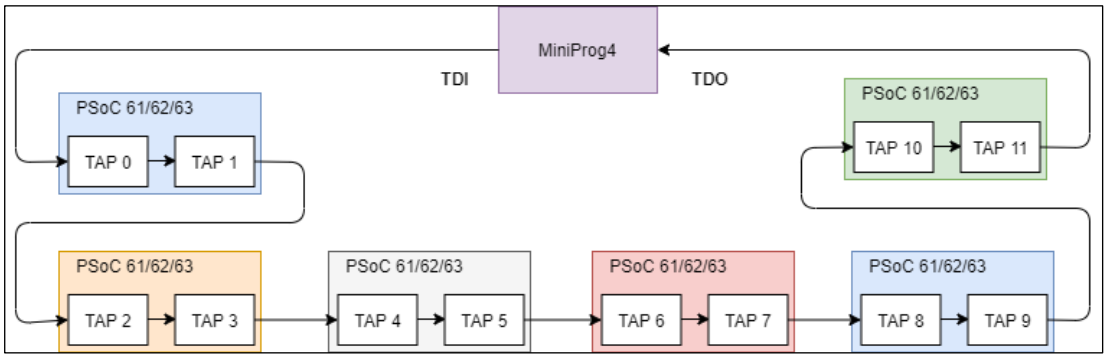
ModusToolbox™ Programmer programs the device and displays various messages in the **Log**. Then, a message in the Status Bar indicates that the device was programmed successfully or that an error occurred.



## 5.6 Program PSoC™ 6 MCU in JTAG Chain

1. Connect the host computer to a MiniProg4 or J-Link probe attached to several MCU targets in the JTAG chain.

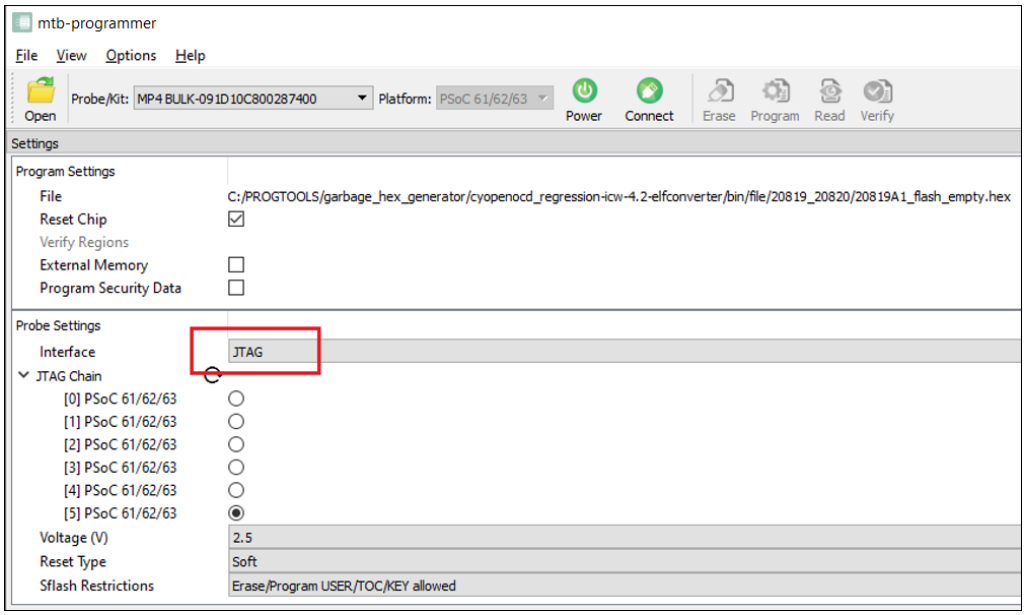
   The following hardware configuration is used in this example:




   The sample JTAG chain configuration contains six serially connected PSoC™ 6 MCU targets.

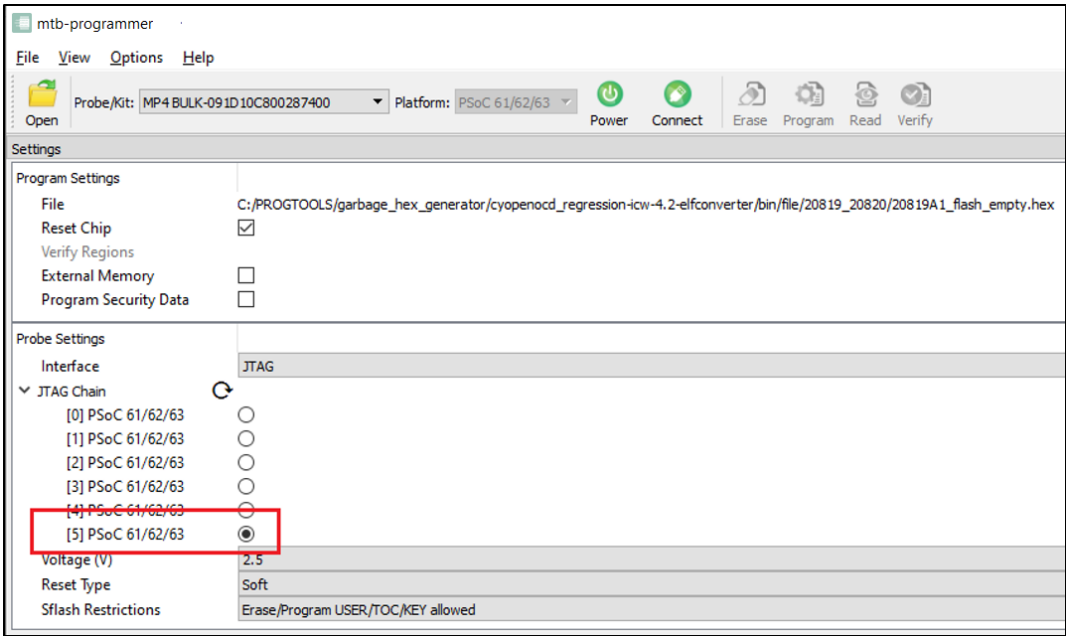
2. Select the MiniProg4 probe in the **Probe/Kit** drop-down and ModusToolbox™ Programmer will display information under **Probe Settings**. Ensure the JTAG chain is powered.

3. Select the JTAG interface in the **Interface** drop-down.



## Programming Operations

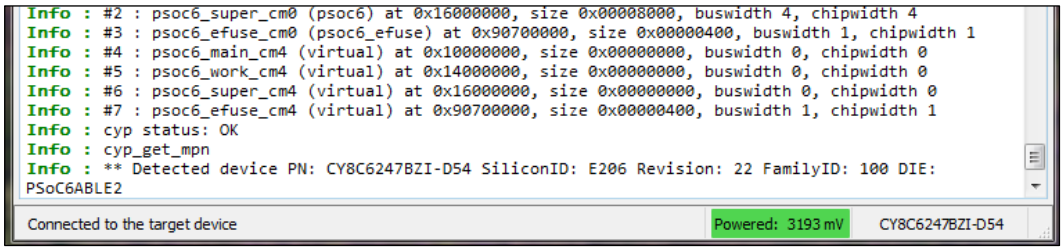
4. ModusToolbox™ Programmer queries the JTAG chain and displays detected devices under the **JTAG Chain** option in **Probe Settings**. The list of devices in the chain contains target names for supported devices and ID codes for those which are not supported.




5. Select the desired target device in the list by clicking the radio button next to the target name.

6. Select the programming file as described in the Load Programming File section.

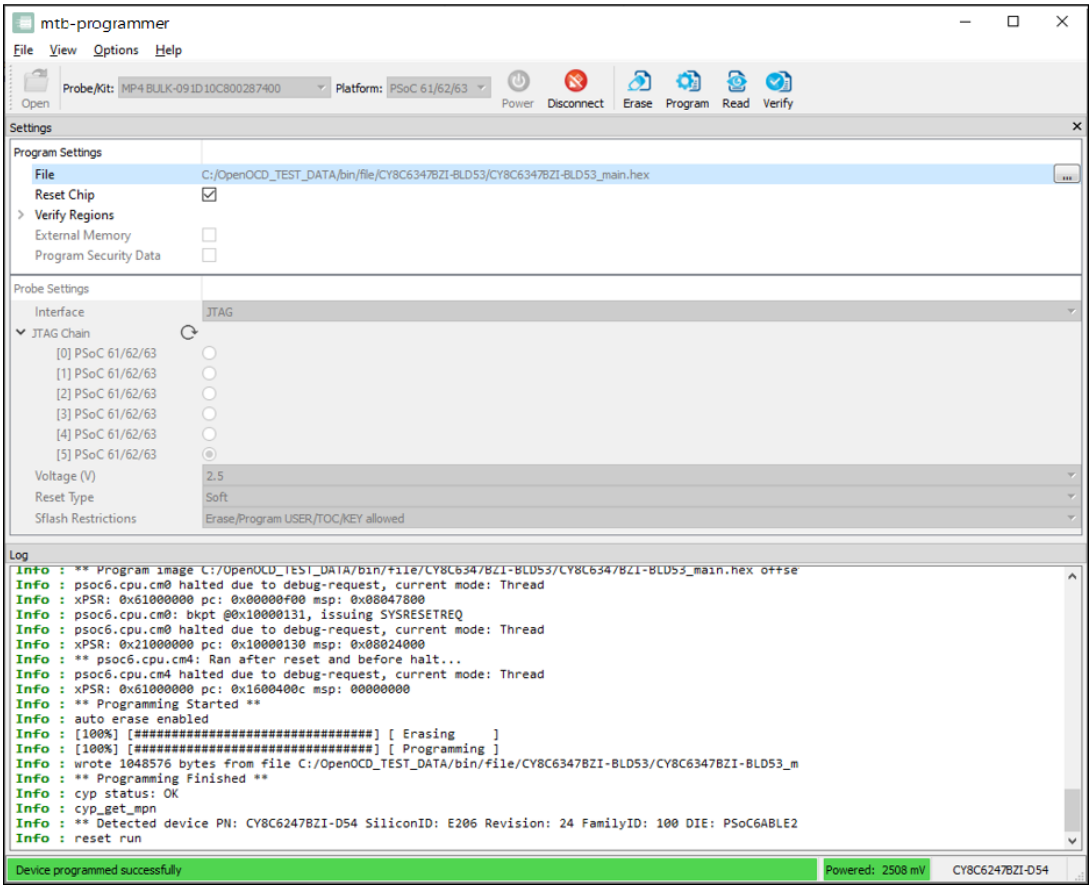
7. Click **Connect**. ModusToolbox™ Programmer communicates with the device and displays various messages in the **Log**. Then, a message in the Status Bar indicates that it is connected.



## Programming Operations

8. Click the **Program** button.

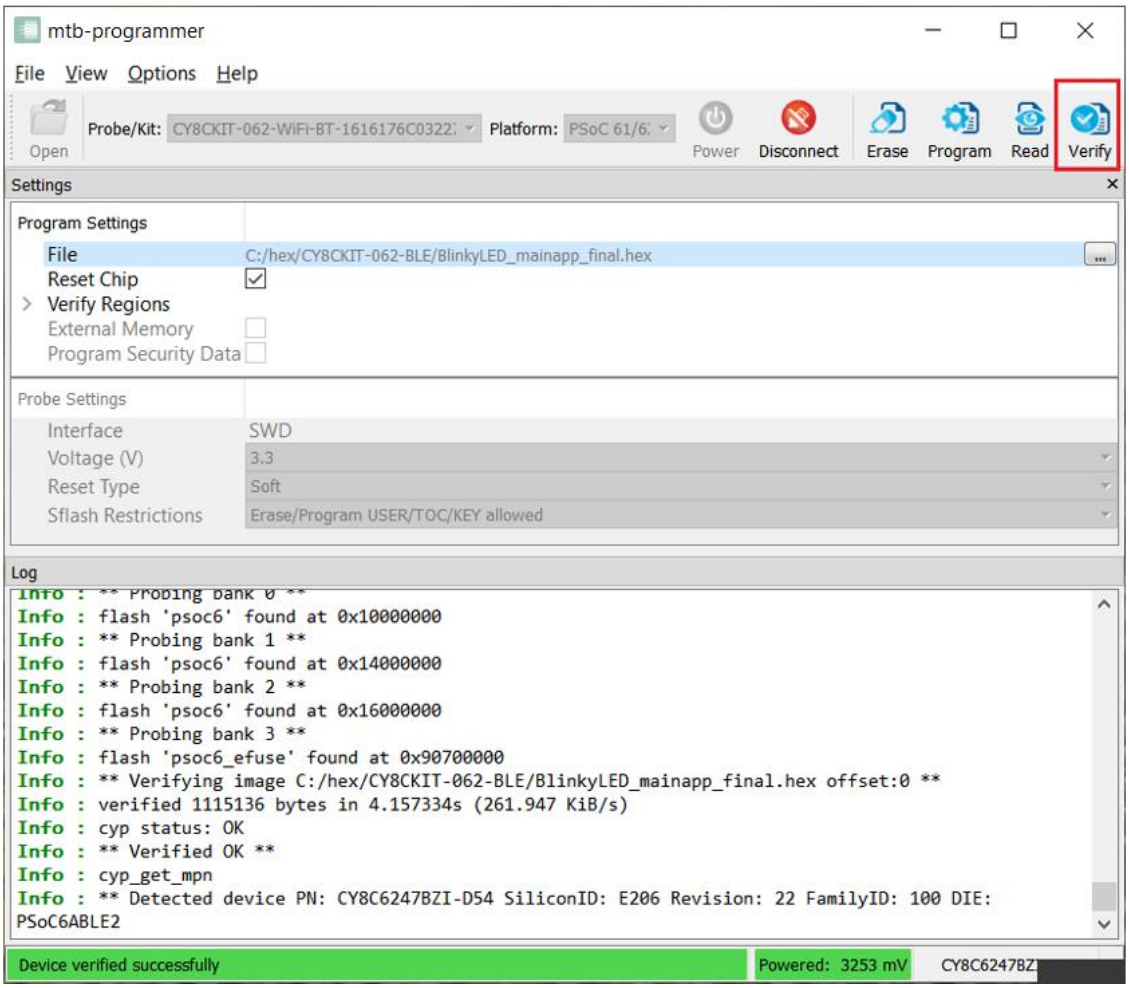
   ModusToolbox™ Programmer programs the device and displays various messages in the **Log**. Then, a message in the Status Bar indicates that the device was programmed successfully or that an error occurred.



## 5.7 Verify Device

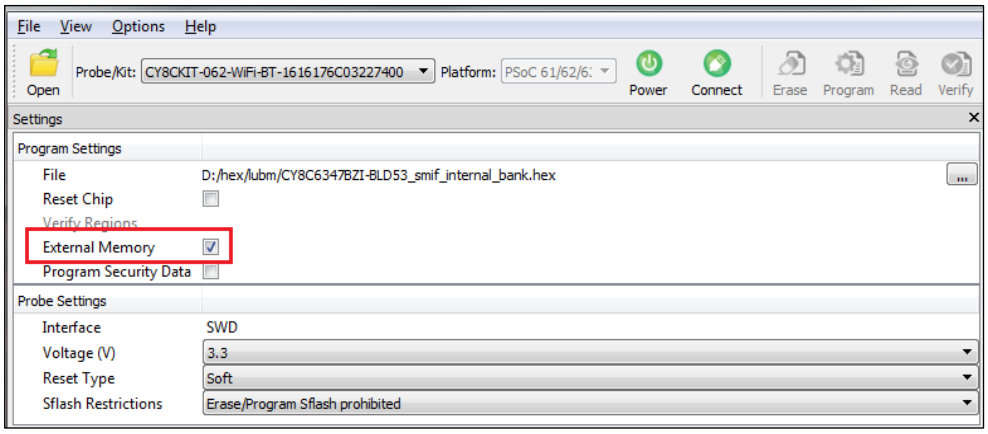
1. Connect the device to the host computer and select it in the **Probe/Kit** drop-down.
2. Select the programming file as described in the Load Programming File section.
3. Connect to the device (see Connect Device).
4. Click the **Verify** button.

ModusToolbox™ Programmer performs the Verify device operation and displays various messages in the **Log**. Then, a message in the Status Bar indicates that the device was verified successfully or that an error occurred.



## 5.8 Verify Device with External Memory

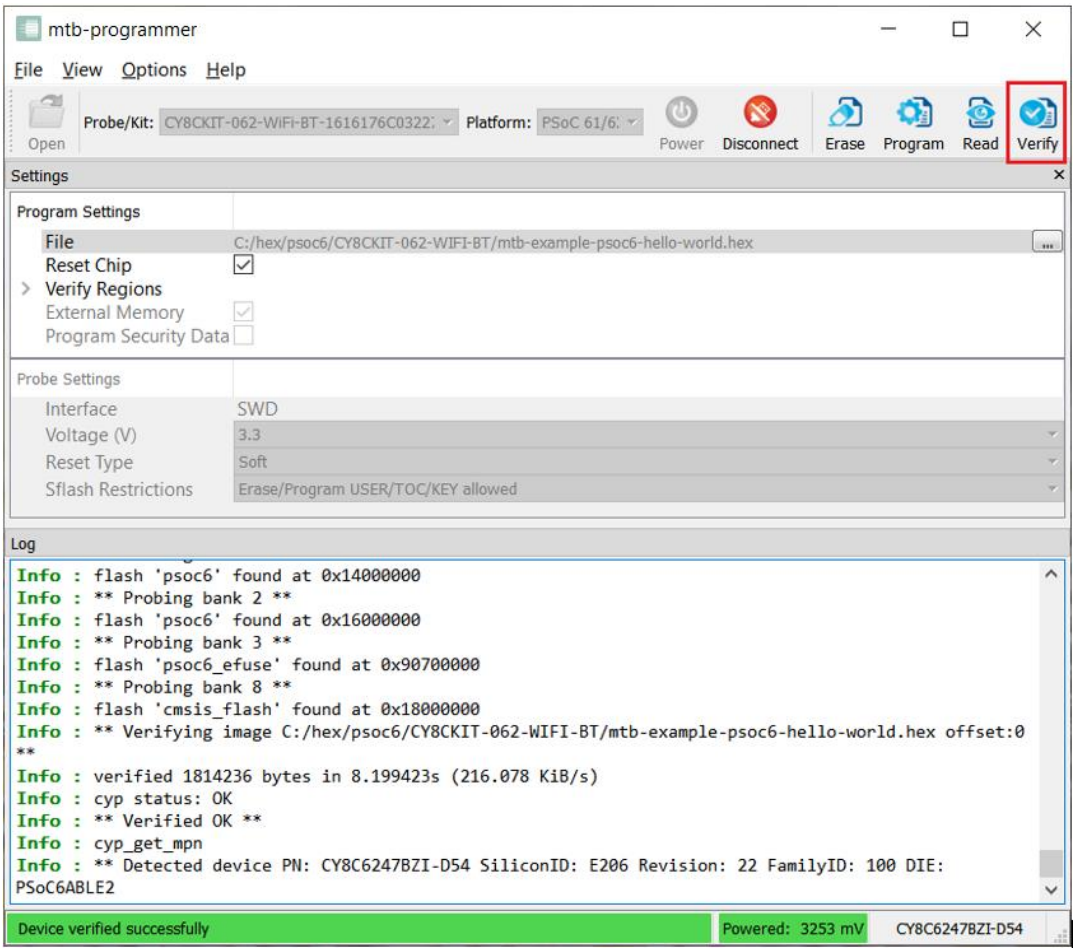
1. Connect the device that supports external memory (for example, CY8CKIT-062-WiFi-BT with QSPI support) to the host computer and select it in the **Probe/Kit** drop-down.
2. Select the programming file as described in the Load Programming File section. The programming file should have external memory region(s).
3. Select the External Memory option under Program Settings.




4. Connect to the device (see Connect Device).
5. Click the **Verify** button.

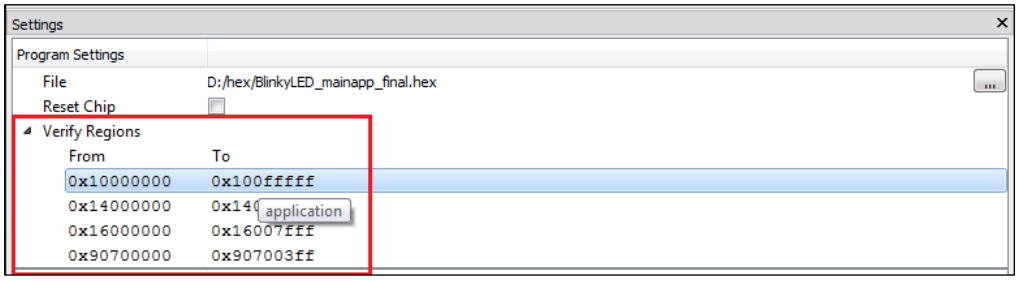
### Programming Operations

ModusToolbox™ Programmer verifies the device and displays various messages in the **Log**. Then, a message in the Status Bar indicates that the device was verified successfully or that an error occurred.

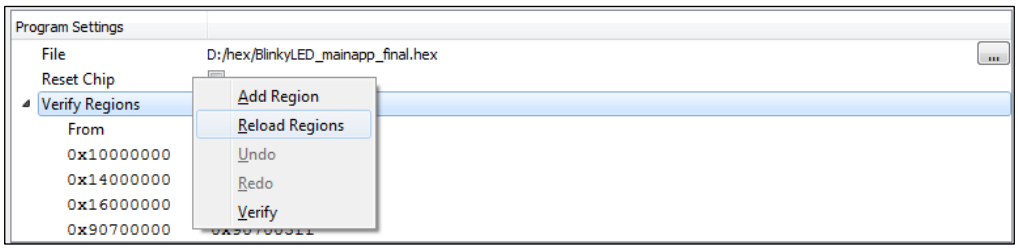


## 5.9 Verify Custom Flash Regions of PSoC™ 6 MCU

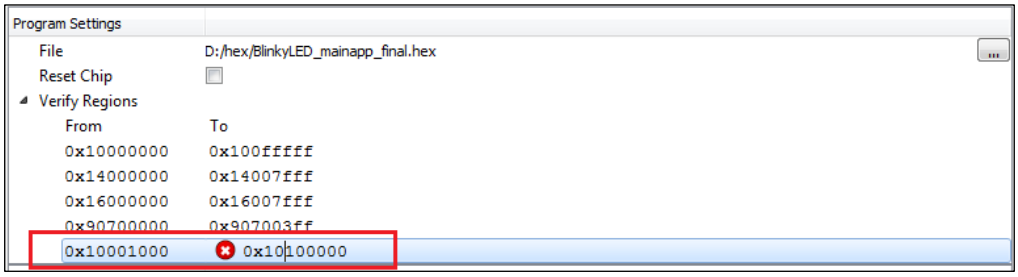
1. Connect the device to the host computer and select it in the **Probe/Kit** drop-down.
2. Select the programming file as described in the Load Programming File section.
3. Connect to the device (see Connect Device).
4. Expand **Verify Regions** option in **Program Settings** to see the list of flash regions available for verification. By default, only supported by target device regions are displayed:
   - application
   - AUXflash
   - Sflash
   - eFuse
   - QSPI




5. Right-click on **Verify Regions,** or any region entry, to open the context menu.

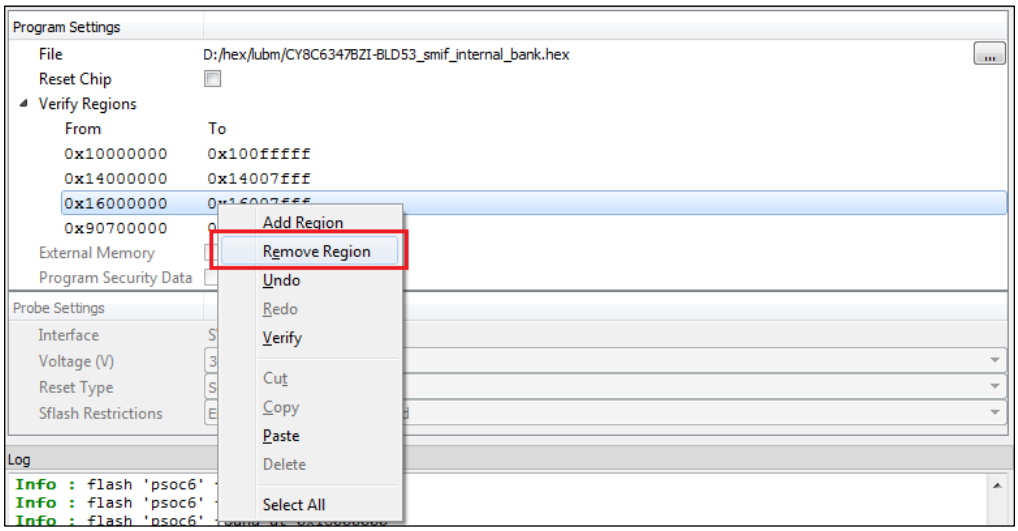



6. To add a custom flash region, select **Add Region**. Select the added list entry, and enter the correct values for start and end addresses of the region.



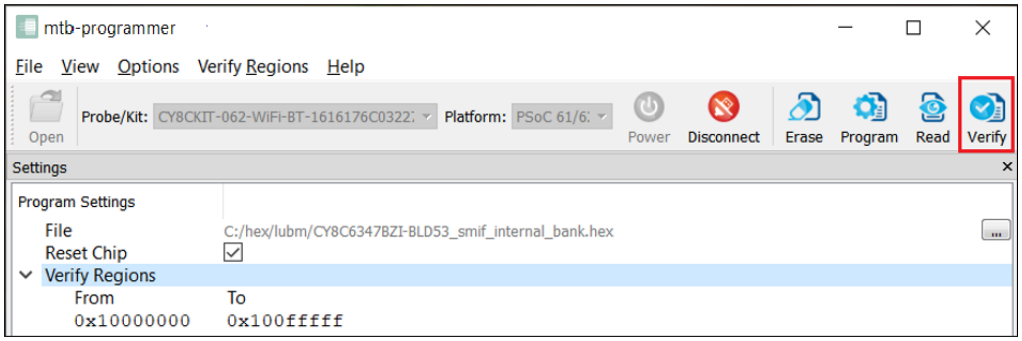
## Programming Operations

7. To remove any region in the list, right-click the desired region entry and select **Remove Region**.

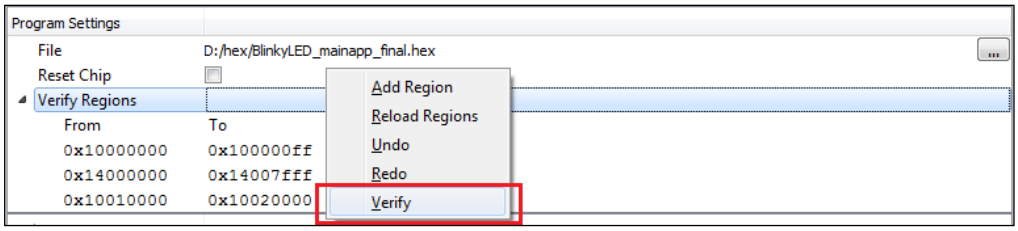



8. To revert any previous change, select **Undo**.

9. When finished with the list of regions, start device verification by clicking the **Verify** button on the toolbar.

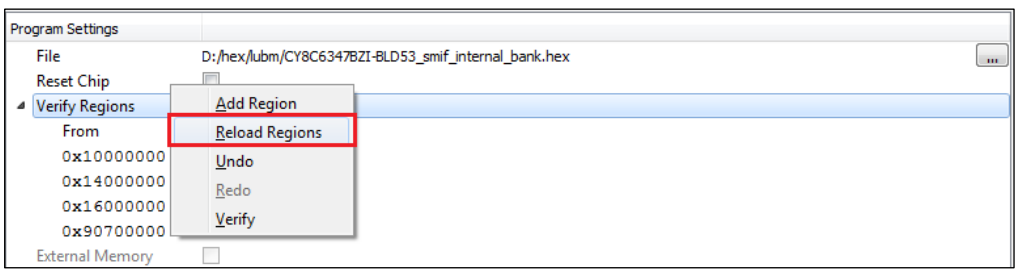



You can also select **Verify** on the context menu.



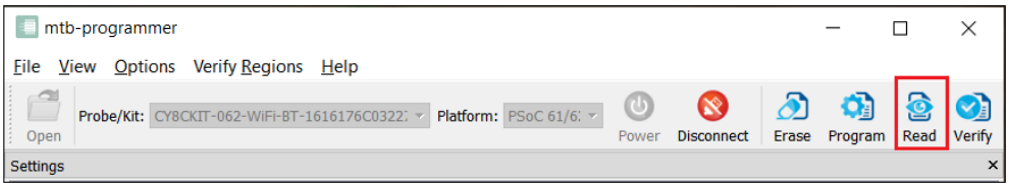

ModusToolbox™ Programmer verifies only for the regions specified in the **Verify Regions** list. Then, a message in the Status Bar indicates that the device was verified successfully or that an error occurred.

10. To reset the **Verify Regions** list to its default state select **Reload Regions** from the context menu. This action will remove all custom regions and load default regions corresponding to the flash map of the PSoC™ 6 MCU.

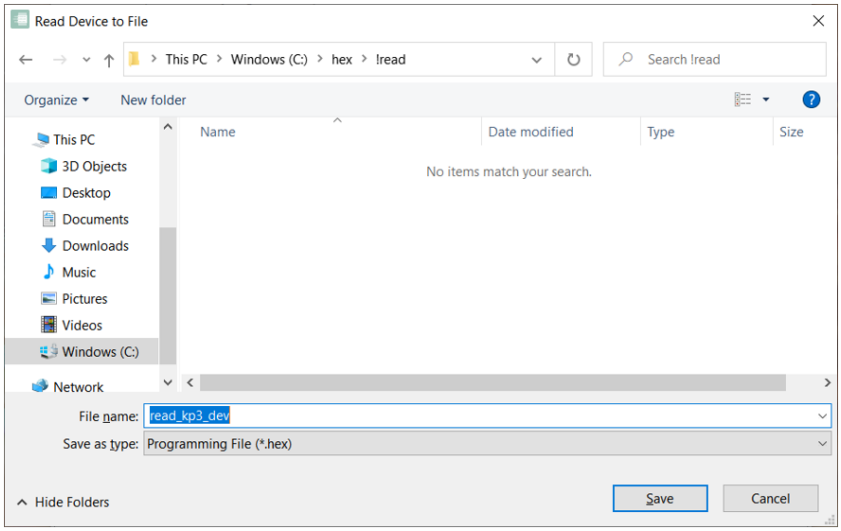


## 5.10 Read Device

1. Connect the device to the host computer and select it in the **Probe/Kit** drop-down.
2. Select the programming file as described in the Load Programming File section.
3. Connect to the device (see Connect Device).
4. Click the **Read** button.



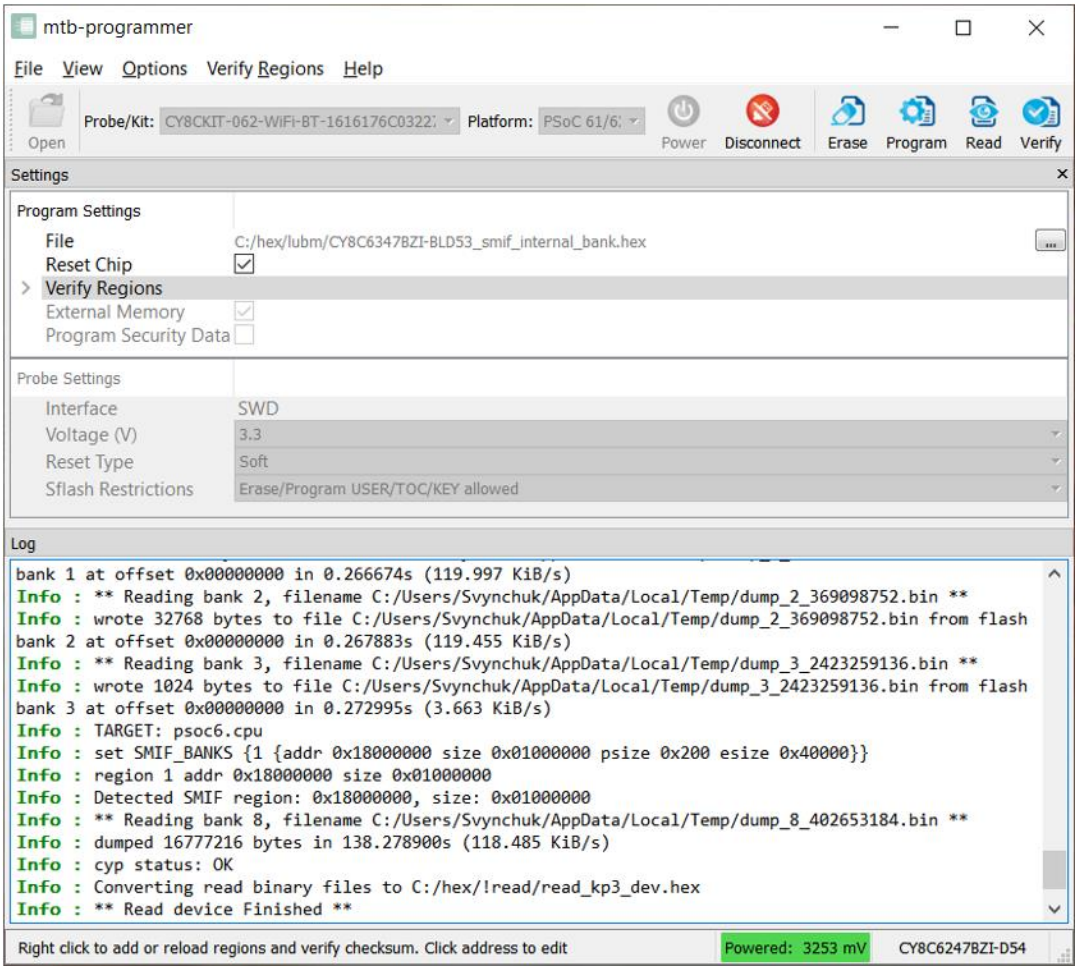

5. On the **Read Device to File** dialog, navigate to the location of the HEX or SREC file to be saved, enter the file name, select the file type in the **Save as type** drop-down, and click **Save**.




*Note:* *Under Ubuntu Linux, specify the full file name with an extension (e.g. kp3-dev.srec); otherwise, the file will be saved in HEX format.*

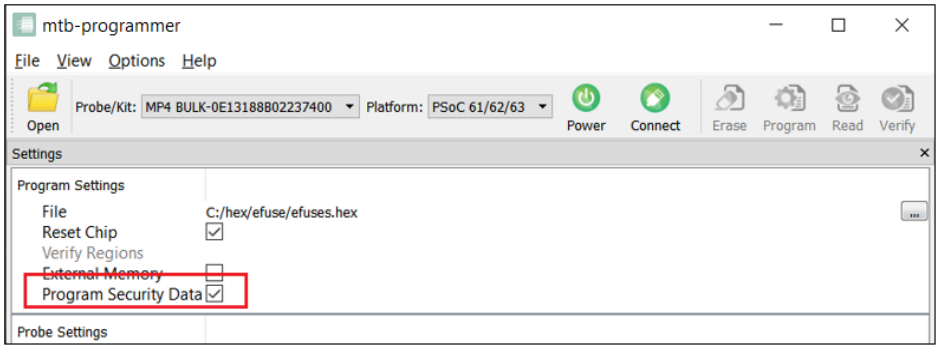
### Programming Operations

ModusToolbox™ Programmer performs the Read device operation and displays various messages in the **Log**. Then, a message in the Status Bar indicates that the device was read successfully or that an error occurred.



## 5.11 Program eFuse Region of PSoC™ 6/TRAVEO™ T2G/XMC7xxx MCU

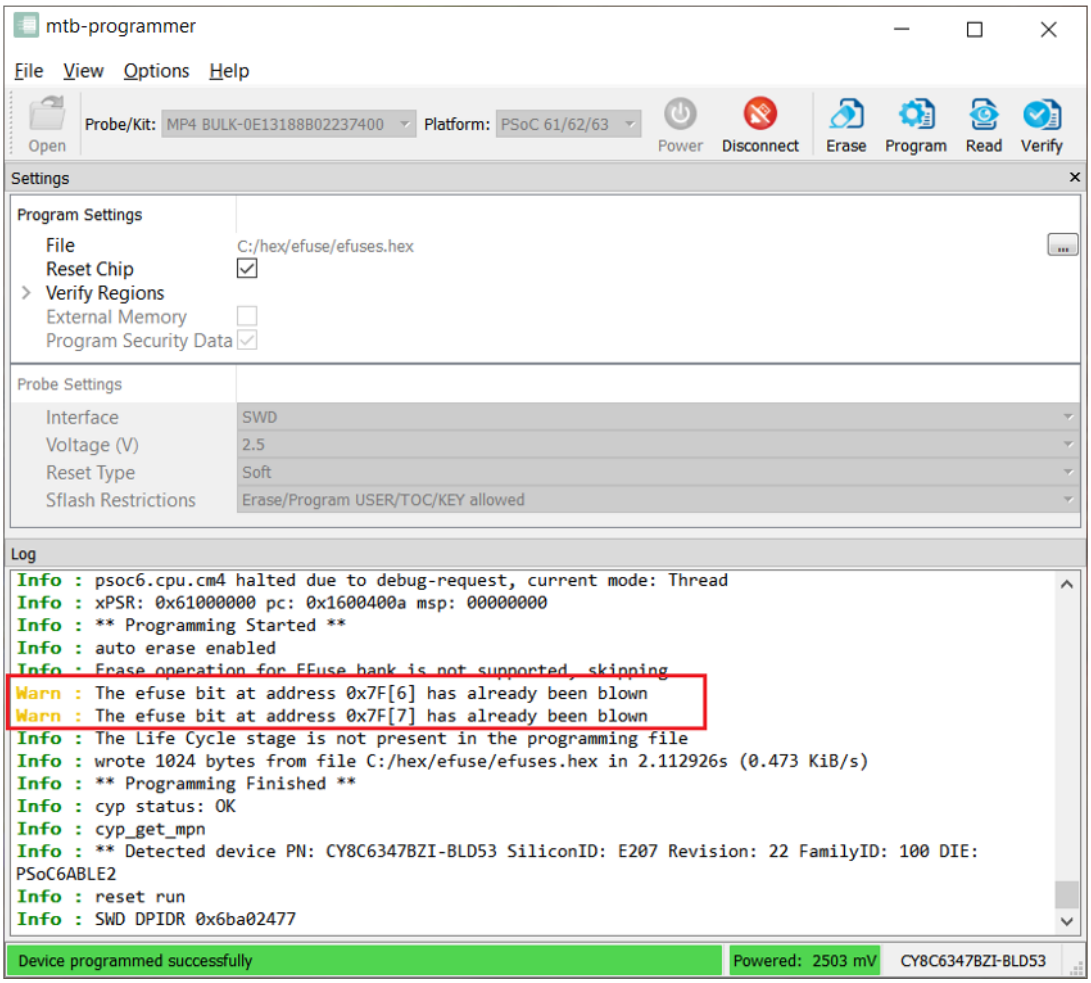
1. Connect the device to the host computer and select it in the Probe/Kit drop-down.
2. Select the programming file as described in the Load Programming File section. The programming file should contain valid eFuse data region (at address 0x90700000).
3. Select the **Program Security Data** check box under Program Settings.




4. Connect to the device (see Connect Device).
5. Click the **Program** button.

### Programming Operations

ModusToolbox™ Programmer programs the device and displays various messages in the **Log**. Then, a message in the Status Bar indicates that the device was programmed successfully or that an error occurred.

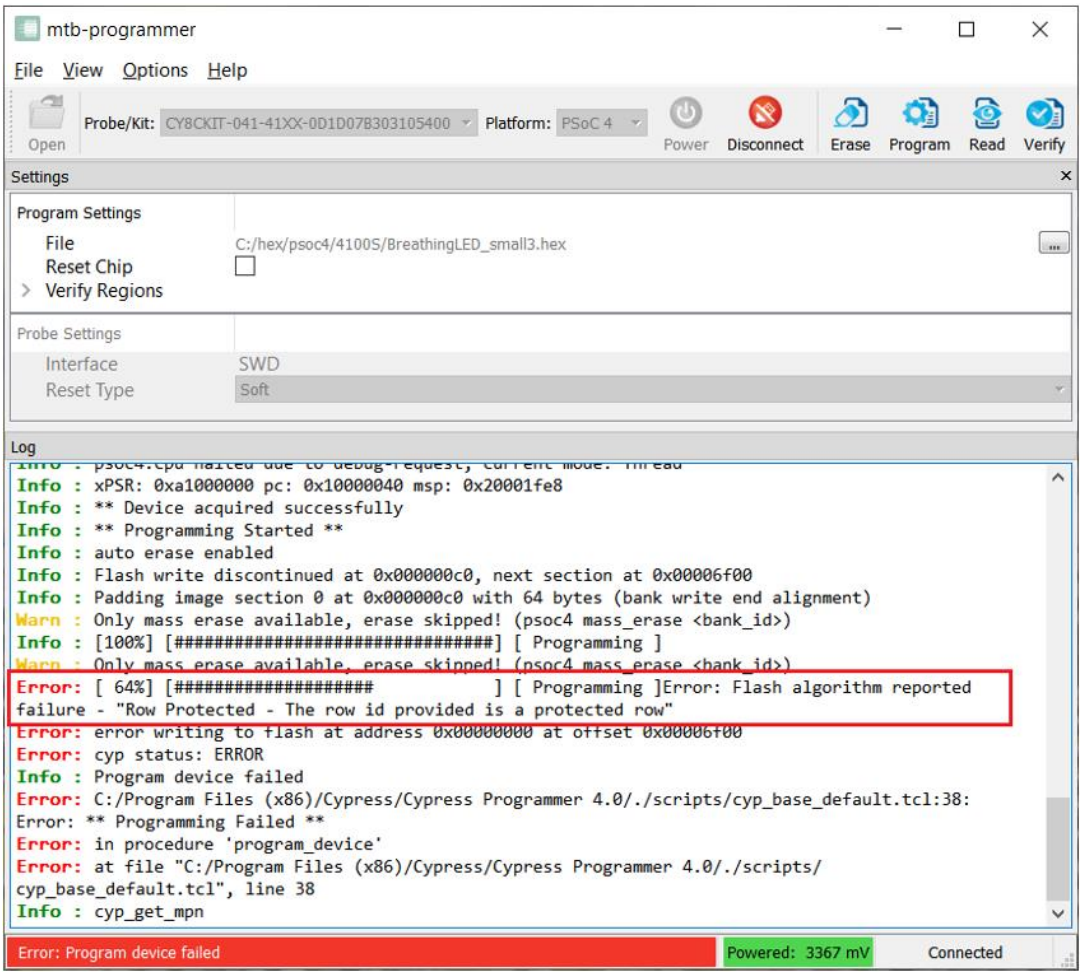



If some eFuse bits have been already programmed before, a warning message "The efuse bit at address xx has been already blown" appears in **Log**.

## 5.12 Program PSoC™ 4 MCU With Protected Flash

Flash protection allows you to protect any PSoC™ 4 flash rows from being written. Applying protection to the user data with ModusToolbox™ Programmer is nothing more than programming a data file containing appropriate flash protection region (at the address 0x90400000).

Programming data into protected flash region will fail with "Programming Failed" error.

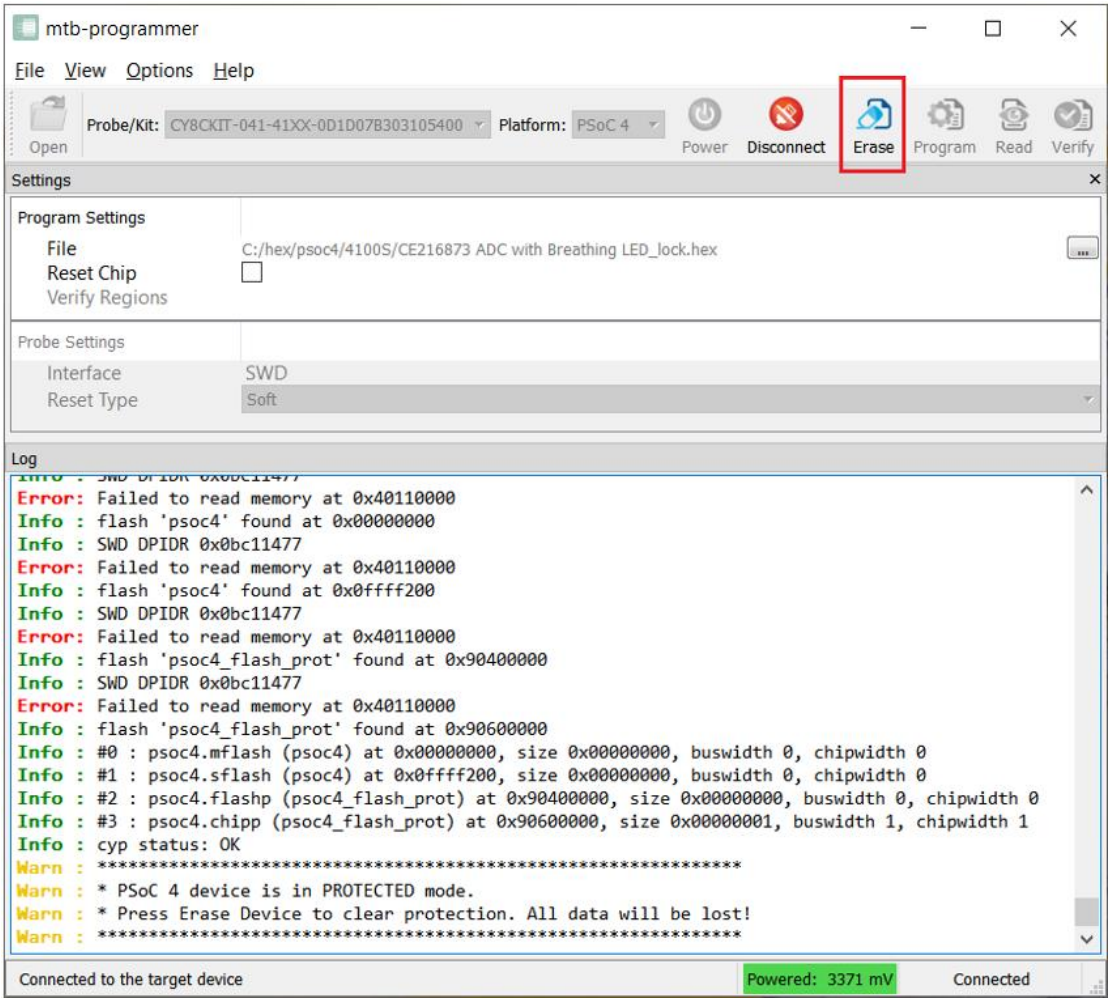



Use the **Erase** button to un-protect user flash rows from writing.

## 5.13 Program Chip-Protected PSoC™ 4 MCU

The chip-level protection mechanism restricts access of the programmer application to silicon resources. In this mode, access to flash, SRAM, and most of the registers in the PSoC™ 4 are disabled. Chip protection can be activated by programming a HEX file with a special protection region at address 0x90600000.

If you try to connect ModusToolbox™ Programmer to a chip-protected PSoC™ 4/PMG1/WLC1 device, a warning message indicates that device is in protected mode. The only available operation is **Erase** device in this case.



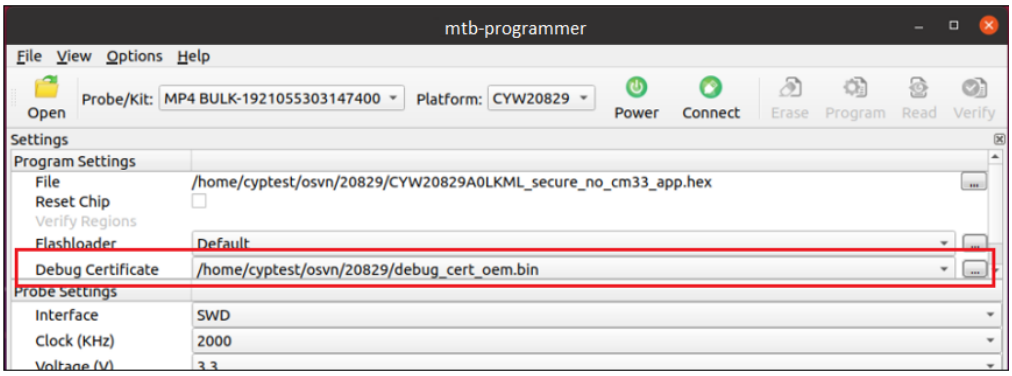

Use the **Erase** button to clear chip protection and move target to the open state.

## 5.14 Program Secure AIROC™ CYW20829 MCU

You can program the flash of an AIROC™ CYW20829 device in Secure lifecycle mode only by providing a valid debug certificate file.

1. Connect the device to the host computer and select it in the Probe/Kit drop-down.
2. Select the programming file as described in the Load Programming File section.
3. Click on the **Debug Certificate** option under **Program Settings** and select the certificate file in the dialog.

   The path will display under **Settings**.



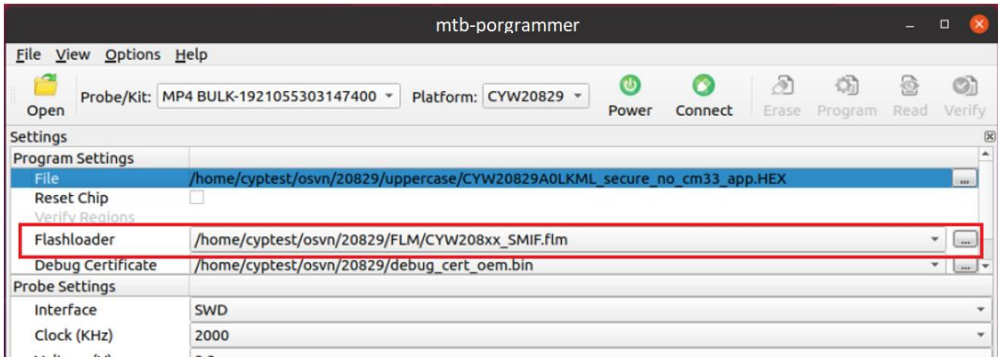

4. Connect to the device (see Connect Device).
5. Click the **Program** button.

## 5.15 Program QSPI memory with patched flashloader

This feature is supported by AIROC™ CYW20829 devices, as well as XMC7100/7200 and CYT4Bx MCUs. It allows you to specify the patched flashloader file in FLM format along with the appropriate QSPI configuration. Patched flashloaders contain data about how your external memory is configured.

1. Connect the device to the host computer and select it in the Probe/Kit drop-down.
2. Select the programming file as described in the Load Programming File section.
3. Select the **External Memory** option under **Program Settings** (if supported).
4. Click on the Flashloader option under **Program Settings** and select the patched flashloader (FLM) file in the dialog.

   The path will display under **Settings**.




5. Connect to the device (see Connect Device).

6. Click the **Program** button.

*Note:* *To be able to program custom QSPI memory, you should also provide the appropriate QSPI configuration file (qspi_config.cfg) generated by the ModusToolbox™ QSPI Configurator tool. This file should be placed in the same directory as the patched flashloader file; it is located and read by ModusToolbox™ Programmer automatically.*
*Refer to the QSPI Configurator user guide for more information about patching flashloaders.*

# 6 Troubleshooting

## 6.1 Limitations

- ModusToolbox™ Programmer does not support RAM programming, you can program only the flash memory of a target device, except for devices that support only RAM such as CYW9M2BASE-43012BT and CYW955513EVK-01 kits.
- Programming devices in DAPLink mode is not supported. You have to switch your device into CMSIS-DAP BULK mode by pressing the mode selection button.

## 6.2 How to Recover AIROC™ Bluetooth® Devices on Failure

If the program operation for an AIROC™ Bluetooth® device fails, it is possible the memory on the board has been corrupted by a previously loaded application, or the application used a custom baud rate that the download process does not detect.

To recover from this, it may be necessary to reset the board to factory defaults, as follows:

1. Press and hold the **Recovery** button (SW1).
2. Press the **Reset** button (SW2).
3. Release the **Reset** button (SW2).
4. Release the **Recovery** button (SW1).

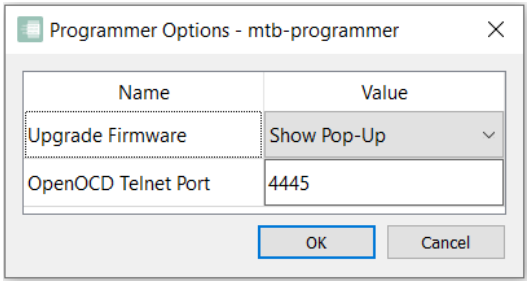
# 7 Upgrading firmware

The ModusToolbox™ Programmer application allows you to upgrade KitProg2, KitProg3, and MiniProg4 device firmware.

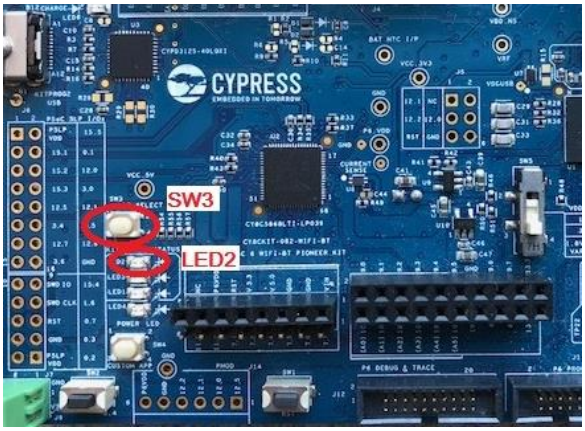
## 7.1 Upgrade KitProg2 firmware

The following upgrade process is based on the CY8CKIT-062-WIFI-BT hardware.

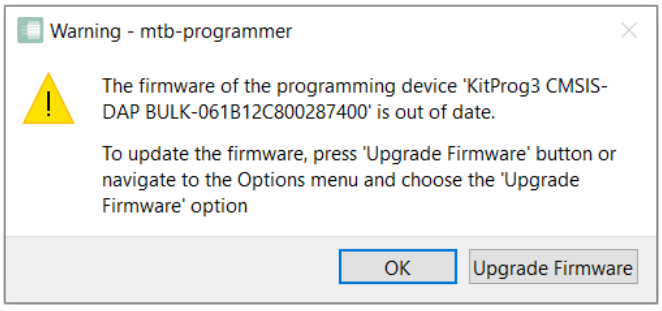
1. Run the ModusToolbox™ Programmer application.
2. Go to **Options > Programmer Options** and select the Show Pop-Up value for the **Upgrade Firmware** option.




3. Click **OK** to apply the changes.
4. Connect the device with the KitProg2 firmware to the host PC. Ensure that the KitProg2 device is in Native KP2 mode.
5. If **LED2** is off, press the **SW3** (Mode Select) button and hold it for about 1 second. When **LED2** is on, the device is ready for upgrading the firmware.



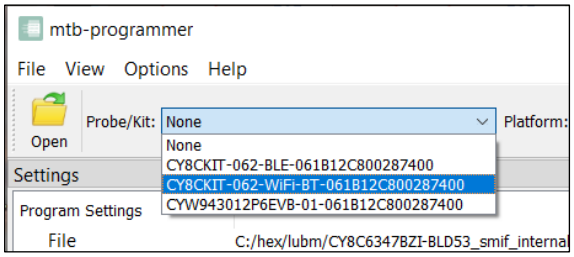

A warning dialog displays with the "Firmware is Out of Date" message.



### Upgrading firmware

6. Click the **Upgrade Firmware** button to start the upgrade process.

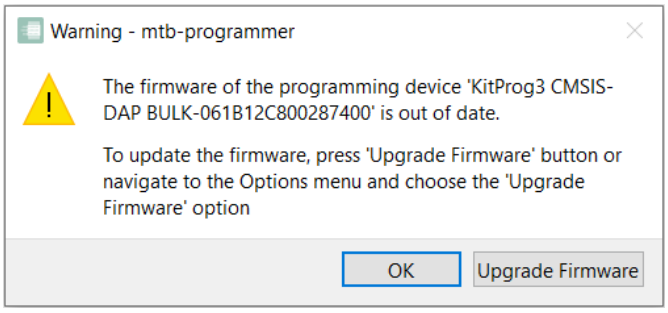
After the upgrade process completes, the message "Firmware of 'KitProg2-xx' upgraded successfully" displays in the **Log** view. The KitProg2-xx device disappears from the **Probe/Kit** drop-down. The **Probe/Kit** drop-down is populated with the names of the supported KitProg3 Kits.




## 7.2 Upgrade KitProg3 on kit or MiniProg4 firmware

Follow this process to upgrade KitProg3 on a kit or MiniProg4 firmware:

1. Run the ModusToolbox™ Programmer application.
2. As needed, go to **Options > Programmer Options** and select the Show Pop-Up value for the **Upgrade Firmware** option.
3. Connect the KitProg3/MiniProg4 device to the host PC. A warning dialog with the "Firmware is Out of Date" message displays.




4. Click the **Upgrade Firmware** button to start the upgrade process.

After the upgrade process completes, the message "Firmware of 'KitProg3-xx' upgraded successfully" or "Firmware of 'MiniProg4 xx' upgraded successfully" displays in the **Log** view.

# Revision history

| Version | Date | Description |
|---|---|---|
| ** | 2023-07-18 | New document. |
| *A | 2018-10-30 | Updates for Production milestone:<br>• Various screen captures<br>• Description of the Platforms pull-down menu<br>• Description of the Clock option in Probe Settings |
| *B | 2019-07-19 | Updated to version 2.1.<br>• Added Verify Regions menu.<br>• Added JTAG Chain instructions.<br>• Added Verify Custom Flash Regions section. |
| *C | 2019-10-11 | List of changes:<br>• User Guide clean-up in whole document<br>• Updated section "Features"<br>• Updated section "Limitations"<br>• Updated section "Settings"<br>• Added section "How to Recover AIROC™ Bluetooth® Devices on Failure" |
| *D | 2020-05-04 | List of changes:<br>• Updated to version 3.0<br>• Updated screenshots across the document<br>• Made modifications with new features of CYP 3.0<br>• Added section "Program eFuse Region of PSoC™ 6 MCU"<br>• Corrected mistakes. |
| *E | 2021-03-18 | List of changes:<br>• Updated to version 4.0.<br>• Updated all sections with new screenshots<br>• Updated sections "Getting Started", "Settings", "Program PSoC™ 6 MCU in JTAG Chain"<br>• Added section "Program Chip Protected PSoC™ 4 MCU"<br>• Corrected mistakes |
| *F | 2022-03-15 | List of changes:<br>• Updated to version 4.0.1<br>• Updated section "Introduction"<br>• Updated section "Installing CYP" |
| *G | 2022-10-28 | List of changes:<br>• Updated to version 4.1.0<br>• Minor updates across the document |
| *H | 2023-02-14 | List of changes:<br>• Updated to version 4.2.0<br>• Updated section 4.2.4.1<br>• Added sections 5.14 and 5.15<br>• Minor updates across the document |

## Revision history

| Version | Date | Description |
|---|---|---|
| *I | 2023-09-12 | List of changes:<br>• Updated to version 5.0.0<br>• Changed title of the tool and user guide to ModusToolbox™ Programmer.<br>• Tool is now part of Programming tools package; installation instructions now included in the release notes. |

**Trademarks**
All referenced product or service names and trademarks are the property of their respective owners.
The Bluetooth® word mark and logos are registered trademarks owned by Bluetooth SIG, Inc., and any use of such marks by Infineon is under license.

**Edition 2023-09-13**
**Published by**
**Infineon Technologies AG**
**81726 Munich, Germany**



**Do you have a question about this document?**
**Email:** erratum@infineon.com

**Document reference**
**002-25230 Rev.*I**

**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.