Please note that Cypress is an Infineon Technologies Company.
The document following this cover page is marked as “Cypress” document as this is the company that originally developed the product. Please note that Infineon will continue to offer the product to new and existing customers as part of the Infineon product portfolio.

Continuity of document content
The fact that Infineon offers the following product as part of the Infineon product portfolio does not lead to any changes to this document. Future revisions will occur when appropriate, and any changes will be set out on the document history page.

Continuity of ordering part numbers
Infineon continues to support existing part numbers. Please continue to use the ordering part numbers listed in the datasheet for ordering.

Cypress Programmer

GUI User Guide

Document No.: 002-25230 Rev. *E

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

Copyrights

© Cypress Semiconductor Corporation (an Infineon company), 2018-2021. This document is the property of Cypress Semiconductor Corporation and its subsidiaries, including Spansion LLC (“Cypress”). This document, including any software or firmware included or referenced in this document (“Software”), is owned by Cypress under the intellectual property laws and treaties of the United States and other countries worldwide. Cypress reserves all rights under such laws and treaties and does not, except as specifically stated in this paragraph, grant any license under its patents, copyrights, trademarks, or other intellectual property rights. If the Software is not accompanied by a license agreement and you do not otherwise have a written agreement with Cypress governing the use of the Software, then Cypress hereby grants you a personal, non-exclusive, nontransferable license (without the right to sublicense) (1) under its copyright rights in the Software (a) for Software provided in source code form, to modify and reproduce the Software solely for use with Cypress hardware products, only internally within your organization, and (b) to distribute the Software in binary code form externally to end users (either directly or indirectly through resellers and distributors), solely for use on Cypress hardware product units, and (2) under those claims of Cypress’s patents that are infringed by the Software (as provided by Cypress, unmodified) to make, use, distribute, and import the Software solely for use with Cypress hardware products. Any other use, reproduction, modification, translation, or compilation of the Software is prohibited.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS DOCUMENT OR ANY SOFTWARE OR ACCOMPANYING HARDWARE, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. No computing device can be absolutely secure. Therefore, despite security measures implemented in Cypress hardware or software products, Cypress does not assume any liability arising out of any security breach, such as unauthorized access to or use of a Cypress product. In addition, the products described in these materials may contain design defects or errors known as errata which may cause the product to deviate from published specifications. To the extent permitted by applicable law, Cypress reserves the right to make changes to this document without further notice. Cypress does not assume any liability arising out of the application or use of any product or circuit described in this document. Any information provided in this document, including any sample design information or programming code, is provided only for reference purposes. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. Cypress products are not designed, intended, or authorized for use as critical components in systems designed or intended for the operation of weapons, weapons systems, nuclear installations, life-support devices or systems, other medical devices or systems (including resuscitation equipment and surgical implants), pollution control or hazardous substances management, or other uses where the failure of the device or system could cause personal injury, death, or property damage (“Unintended Uses”). A critical component is any component of a device or system whose failure to perform can be reasonably expected to cause the failure of the device or system, or to affect its safety or effectiveness. Cypress is not liable, in whole or in part, and you shall and hereby do release Cypress from any claim, damage, or other liability arising from or related to all Unintended Uses of Cypress products. You shall indemnify and hold Cypress harmless from and against all claims, costs, damages, and other liabilities, including claims for personal injury or death, arising from or related to any Unintended Uses of Cypress products.

Cypress, the Cypress logo, Spansion, the Spansion logo, and combinations thereof, ModusToolbox, WICED, PSoC, CapSense, EZ-USB, F-RAM, and Traveo are trademarks or registered trademarks of Cypress in the United States and other countries. For a more complete list of Cypress trademarks, visit cypress.com. Other names and brands may be claimed as property of their respective owners.

Contents

1 Introduction 5
- Overview 5
- Features 5
- Acronyms and Abbreviations 6
- Document Convention 6
- Revision History 7

2 Installing CYP 8
- Package Contents 8
- System Prerequisites 8
- Installing on Windows 8
- Installing on Ubuntu 9
- Installing on macOS 9

3 Getting Started 10
- Run CYP 10
- Load Programming File 11
- Connect Device 12
- Program Device 14
- Save Log File 14

4 GUI Description 15
- Menus 16
  - File 16
  - View 16
  - Options 16
  - Verify Regions 17
  - Help 17
- Toolbar 17
  - Probe/Kit and Platforms 17
  - Power 17
  - Connect 18
  - Settings 18

5 Programming Operations 20
- Erase Device 20

Program Device 21
Program Device and Reset Chip 22
Program Binary File with Offset 23
Program External Memory 24
Program PSoC 6 MCU in JTAG Chain 26
Verify Device 29
Verify Device with External Memory 30
Verify Custom Flash Regions of PSoC 6 MCU 32
Read Device 34
Program eFuse Region of PSoC 6 MCU 35
Program PSoC 4 MCU With Protected Flash 37
Program Chip-Protected PSoC 4 MCU 38
Limitations 39
How to Recover IoT BT Devices on Failure 39

6 Upgrading Firmware 40
Upgrade KitProg2 Firmware 40
Upgrade KitProg3 on Cypress Kit or MiniProg4 Firmware 41

1 Introduction

Overview

Cypress Programmer (CYP) is a stand-alone, cross-platform, flash programmer tool. It replaces several existing flash programmers for various Cypress devices for MCU and connectivity devices. CYP provides a graphical user interface to Program, Erase, Verify, and Read the flash of the target device. CYP supports HEX, SREC, ELF, and BIN programming file formats.

Features

CYP supports the following:

- Programming Cypress 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
- IoT Wi-Fi/BT platforms (4390x, 4343W, CYW208xx 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 IoT 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 – I2C, SPI, UART communications
- KitProg3 firmware update

Acronyms and Abbreviations

The following table lists various acronyms and abbreviations used in this document along with the associated descriptions.

Term	Description
CMSIS	Arm® Cortex® Microcontroller Software Interface Standard.
CMSIS-DAP	CMSIS Debug Access Port.
CYP	Cypress Programmer.
OpenOCD	The Open On-Chip Debugger is the debugger tool that provides on-chip programming support. This tool acts as a backend of the CYP 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.
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	Programmable System-on-Chip is a family of microcontroller integrated circuits by Cypress. 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

Document Convention

This guide uses the following conventions:

Convention	Usage
`Courier New`	Displays file locations and source code: `C:\ …cd\icc\`, user entered text
Italics	Displays file names and reference documentation: *sourcefile.hex*
[**bracketed, bold**]	Displays keyboard commands in procedures: [**Enter**] or [**Ctrl**] [**C**]
File > New Project	Represents menu paths: **File > New Project > Clone**
Bold	Displays commands, menu paths and selections, and icon names in procedures: Click the **Debugger** icon, and then click **Next**.
Text in gray boxes	Displays cautions or functionality unique to the software or the device.

Revision History

Document Title: Cypress Programmer GUI User Guide Document Number: 002-25230		
Revision	**Date**	**Description of Change**
**	9/28/18	New document.
*A	10/30/18	Updates for Production milestone: • Various screen captures • Description of the Platforms pull-down menu • Description of the Clock option in Probe Settings
*B	7/19/19	Updated Cypress Programmer to version 2.1. • Added Verify Regions menu. • Added JTAG Chain instructions. • Added Verify Custom Flash Regions section.
*C	10/11/19	List of changes: • User Guide clean-up in whole document • Updated section “Features” • Updated section “Limitations” • Updated section “Settings” • Added section “How to Recover IoT BT Devices on Failure”
*D	5/4/20	List of changes: • Updated Cypress Programmer to version 3.0 • Updated screenshots across the document • Made modifications with new features of CYP 3.0 • Added section “Program eFuse Region of PSoC 6 MCU” • Corrected mistakes.
*E	3/18/21	List of changes: • Updated Cypress Programmer to version 4.0. • Updated all sections with new screenshots • Updated sections “Getting Started”, “Settings”, “Program PSoC 6 MCU in JTAG Chain” • Added section “Program Chip Protected PSoC 4 MCU” • Corrected mistakes

2 Installing CYP

Package Contents

The CYP package contains:

- *CypGui* executable – The application to program flash of various Cypress devices
- The drivers and firmware for Cypress MiniProg4, KitProg2, KitProg3 hardware programmers
- The OpenOCD debugger tool that provides on-chip programming support
- The cross-platform library that provides access to USB devices.

System Prerequisites

Cypress recommends the following minimum system configuration:

- PassMark CPU score > 2000 (cpubenchmark.net)
- Minimum of 4 GB RAM
- Minimum 60 MB free disk space
- Minimum 1280x1024 screen resolution

CYP is supported on the following platforms:

- Windows 7 (x64)
- Windows 10 (x64)
- macOS 10.14, 10.15, 11
- Ubuntu 18.04 LTS (x64)

Installing on Windows

1. Run the Windows installer program: *CyProgrammer_4.0.0.<build>.exe*
2. Follow the instructions of the installation wizard.

Installing on Ubuntu

1. Open a command-line terminal.
2. Unpack the Cypress Programmer installation archive. Run:

   ```
   tar -xvzf CyProgrammer_4.0.0.<build>.tar.gz
   ```

3. Install the rules for the connected programming hardware MiniProg4, KitProg3.

   The script `install_rules.sh` script copies *57-cypress_programmer.rules* and *70-wiced-JTAG.rules* files to */etc/udev/rules.d*. It allows a non-superuser to have access to the connected devices:

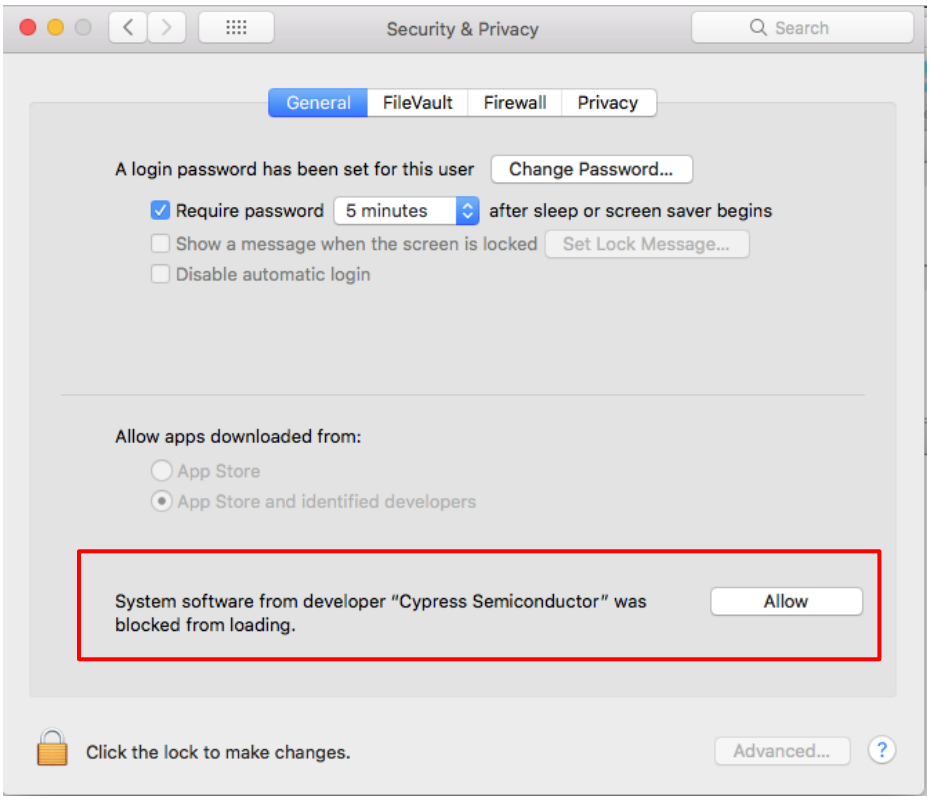
   a. Change directory to *<cyp_install_dir>/udev_rules*.

   b. Run:

   ```
   ./install_rules.sh.
   ```

Installing on macOS

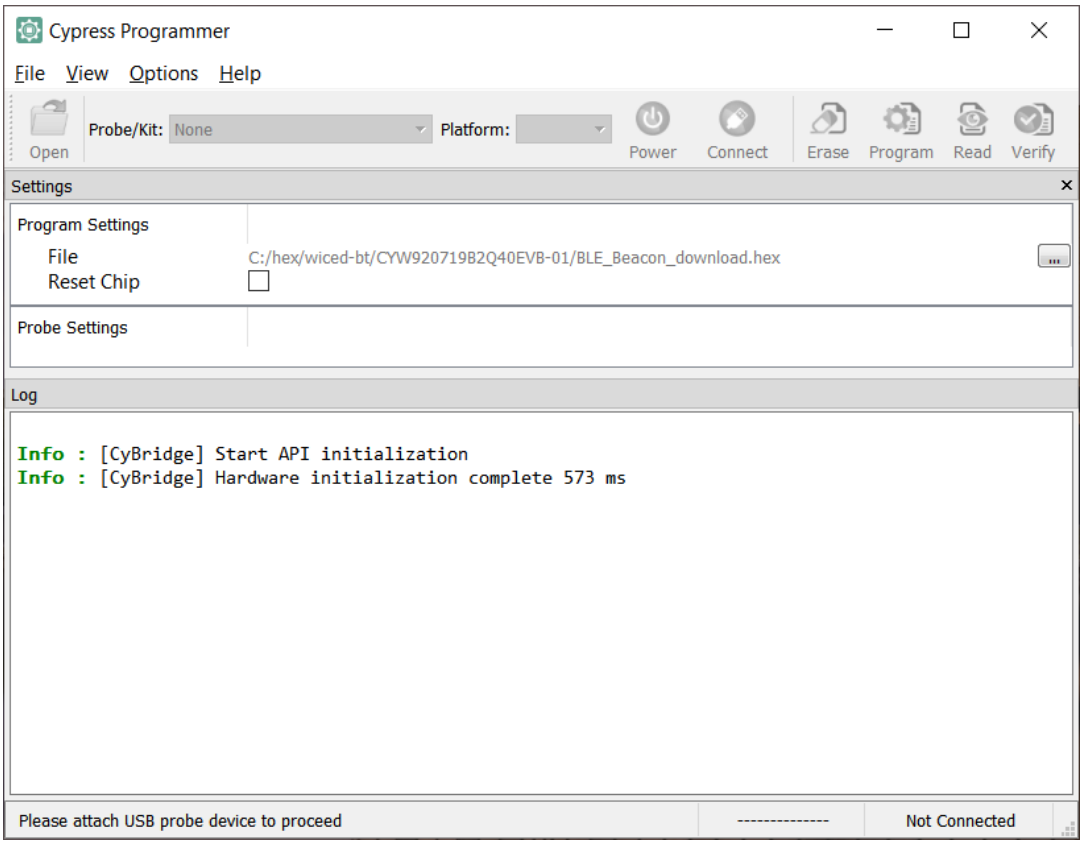
1. Run the pkg installer for the CYP - *CyProgrammer_4.0.0.<build>.pkg*.
2. Follow the instructions on the installation wizard.
3. Under **System Preferences > Security & Privacy > General**, choose to **Allow** the 'System software from developer "Cypress Semiconductor" was blocked from loading.'



3 Getting Started

Run CYP

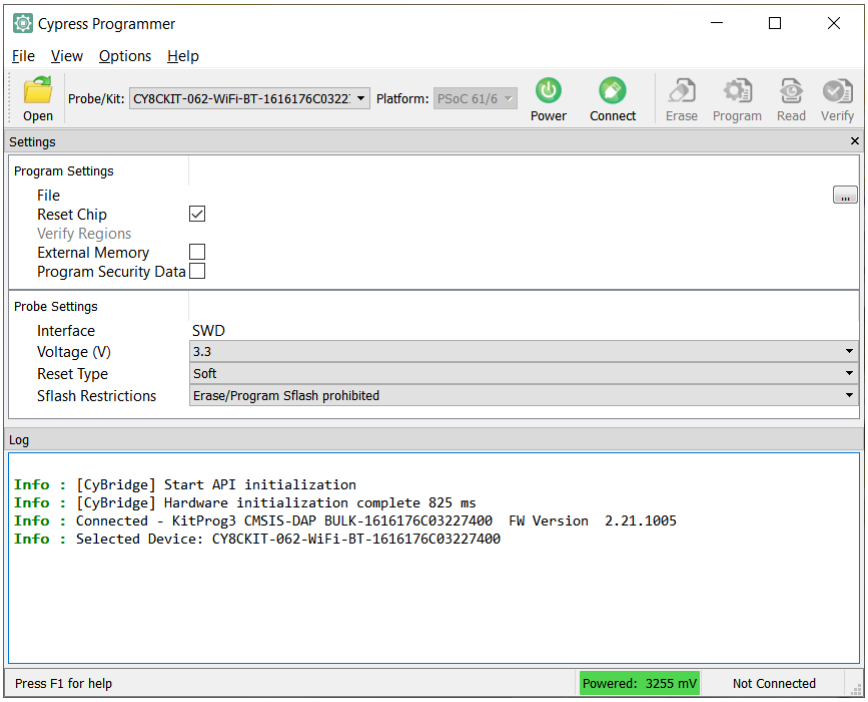
To run the CYP application, navigate to the install location and run the executable. See Installing CYP. The GUI opens and looks similar to this:



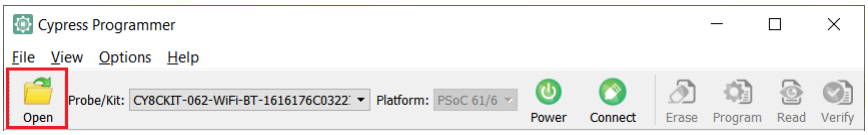
In this case, no kit or device is connected, and a message displays asking you to connect a device.

Load Programming File

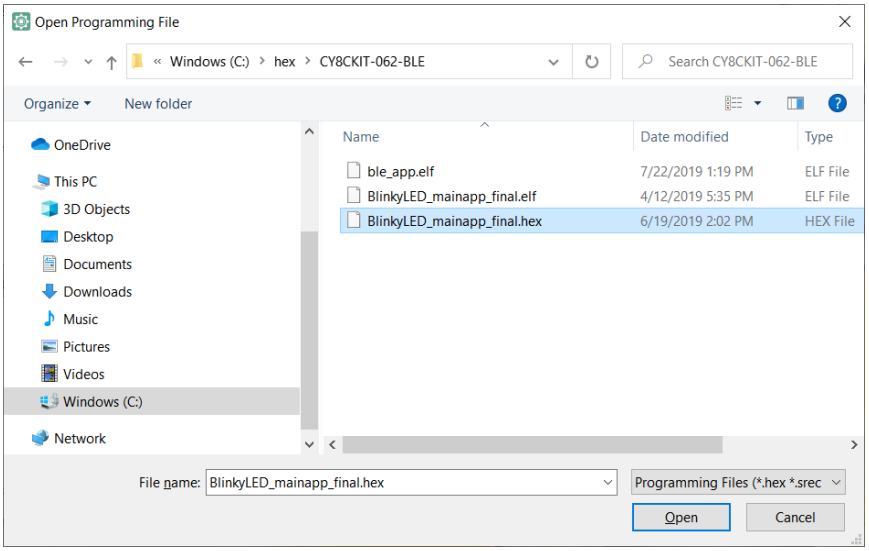
1. Connect the device to the host computer. Select the device name in the **Probe/Kit** drop-down, and CYP will display information under **Probe Settings** (if the **Settings** section is viewable).



2. Click **Open**.

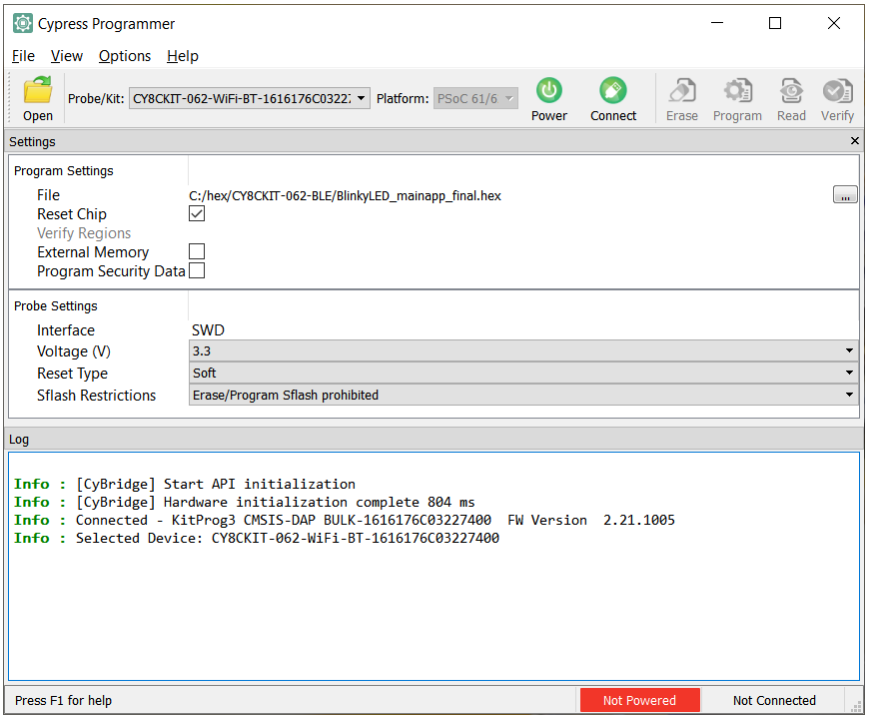


On the Open Programming File dialog, navigate to the location of the HEX, SREC, ELF, or BIN file to load, select it, and click **Open**.

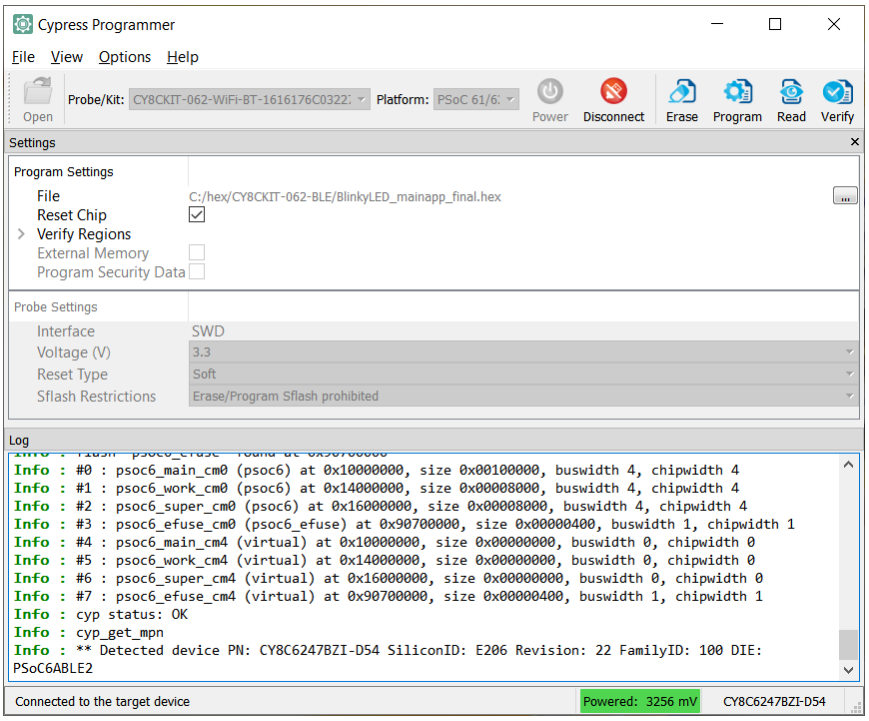


Connect Device

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.

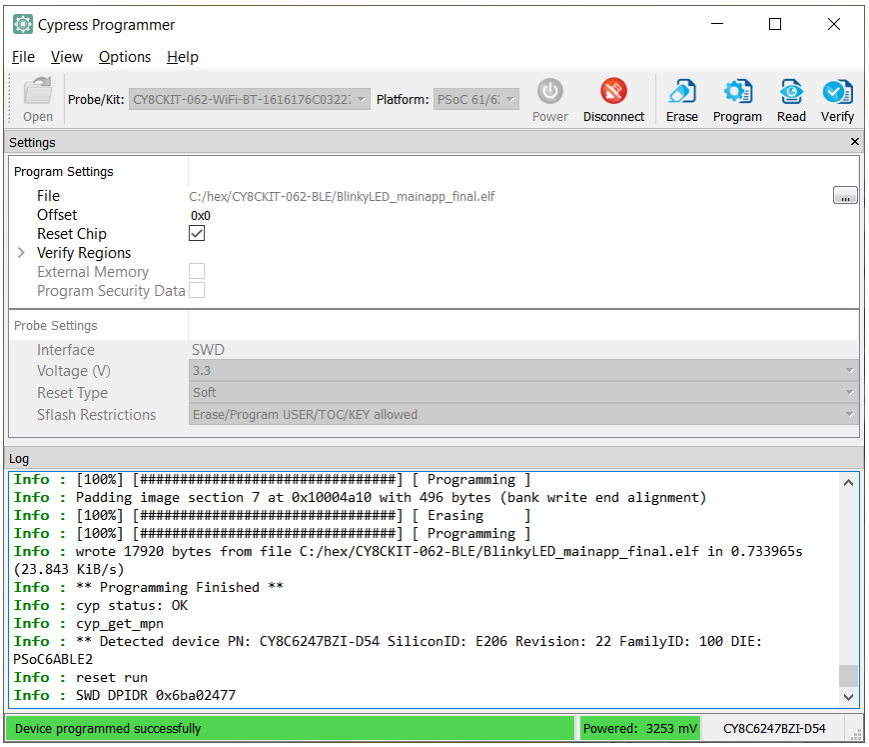


2. Click **Connect**. CYP communicates with the device and displays various messages in the **Log**. Then, a message in the Status Bar indicates that it is connected.



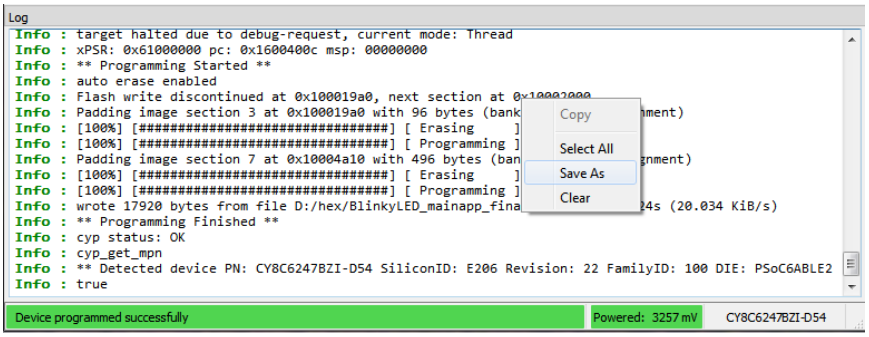
Program Device

Click **Program**. CYP downloads the program file onto the device and displays messages in the **Log**.



Save Log File

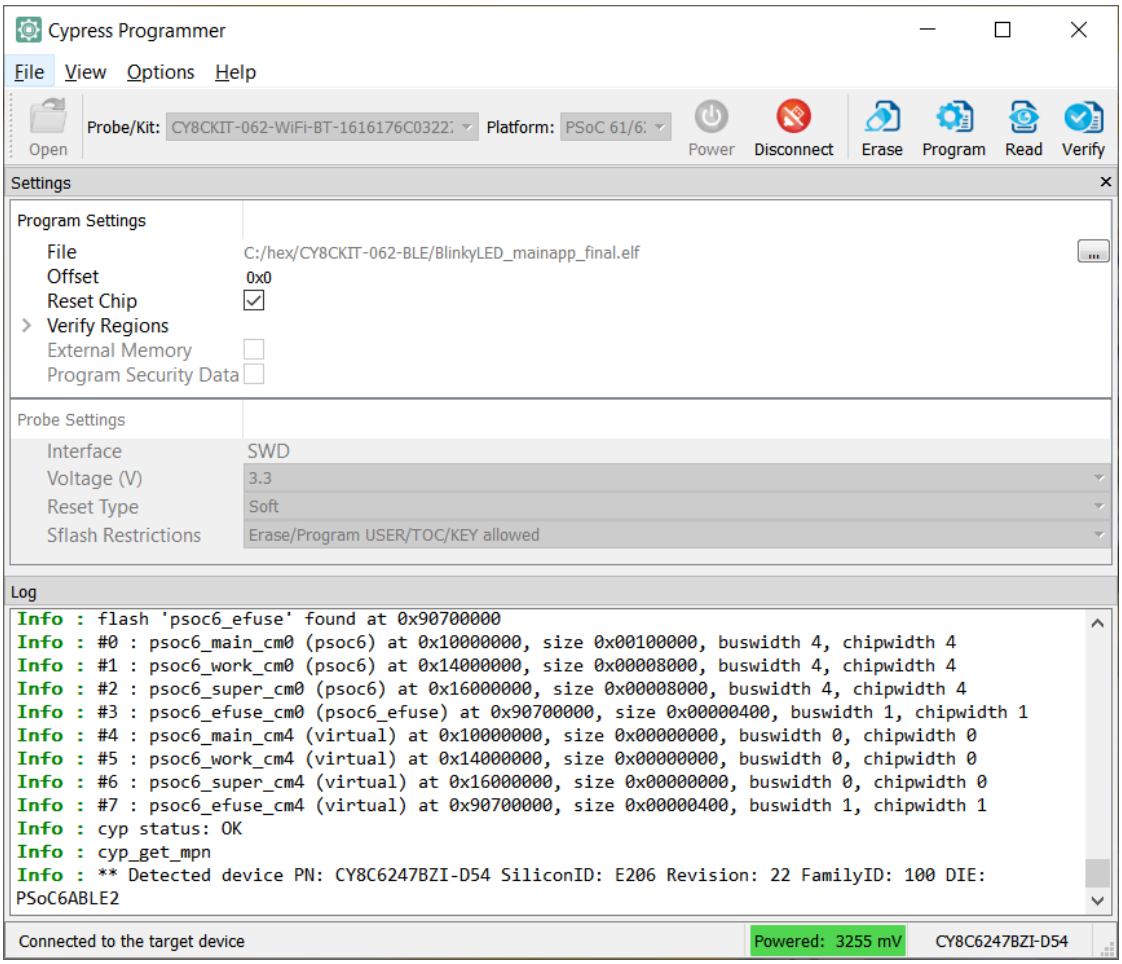
Right-click in the **Log** section and select **Save As**.



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

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



Menus

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 CYP application.

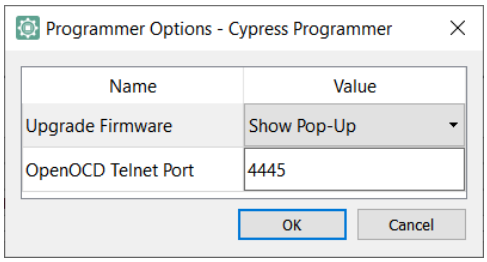
View

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.

Options

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

- **Programmer Options (Alt+T)** – Opens the **Programmer Options** dialog to set the **Upgrade Firmware** mode and other options.
  1. **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.

  2. **OpenOCD Telnet Port**:

     This option specifies the port number of the OpenOCD telnet connection.

- **Upgrade Firmware (Alt+U)** – When this command is enabled, select it to upgrade the programmer firmware on the device.

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.

Help

The Help menu contains the following commands:

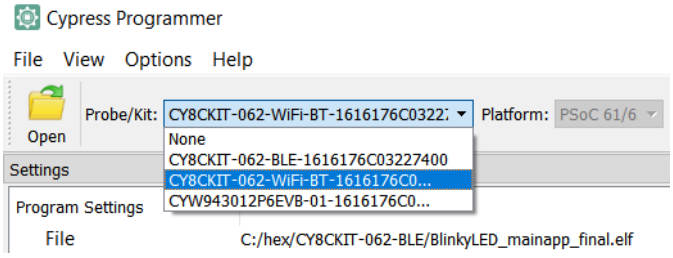
- **View Help (F1)** – Provides access to this document.
- **About Cypress Programmer –** Opens the **About** box.

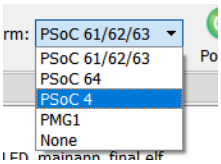
Toolbar

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:

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.





Power

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

Connect

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

Settings

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

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 IoT 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.

Probe Settings

The Probe settings allow you to configure the programming and target device before you connect to it. These settings are available when CYP 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 for PSoC 6 MCUs. This option is available only for PSoC 61/62/63 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.

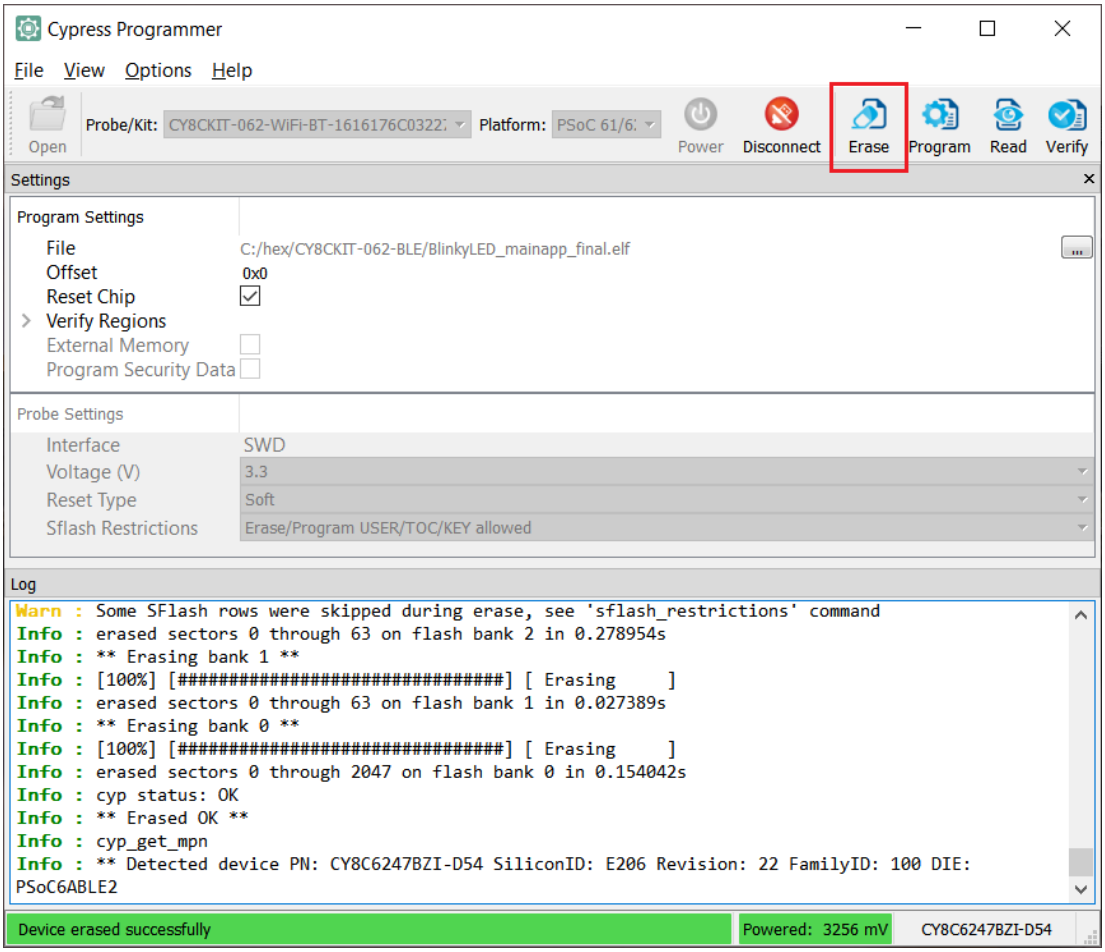
5 Programming Operations

This chapter covers the various programming operations you can perform using CYP tool.

Erase Device

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

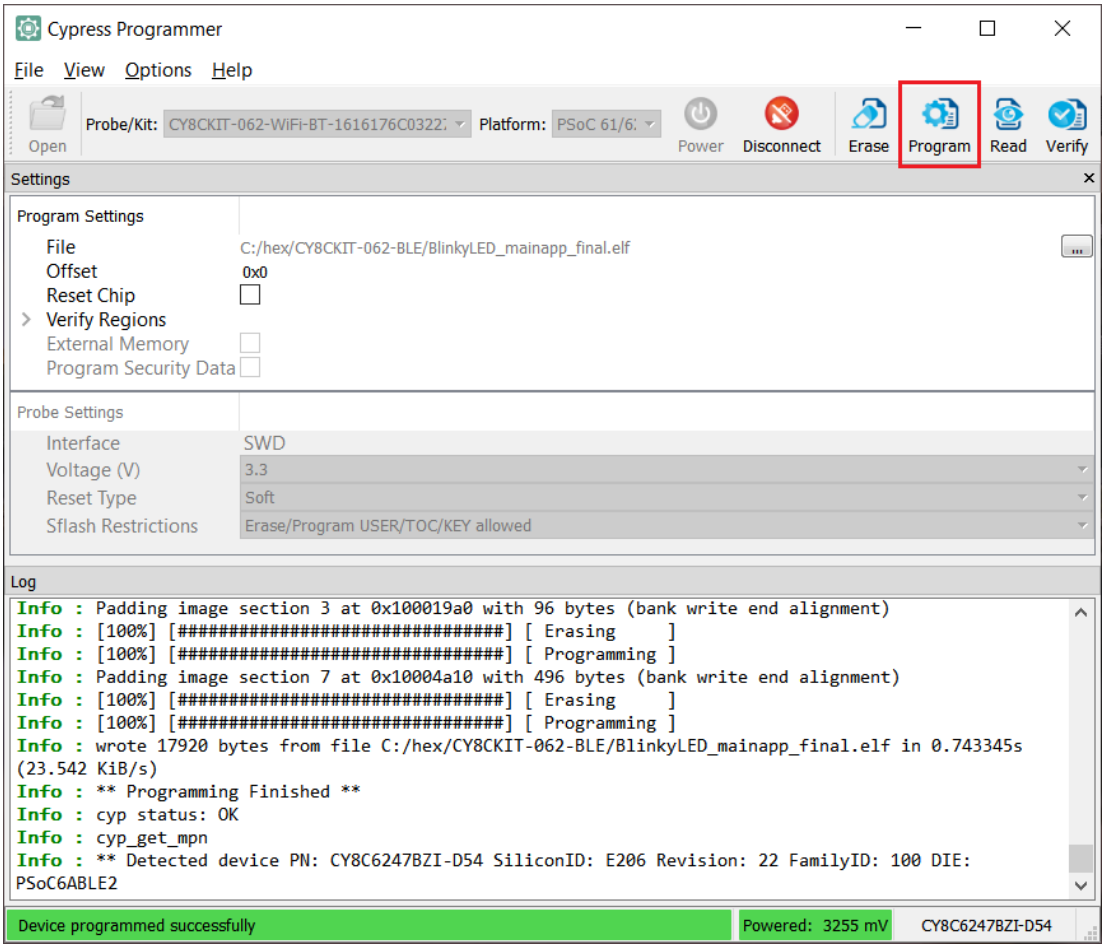
CYP erases the device and displays various messages in the **Log**. Then, a message in the Status Bar indicates that the device was erased successfully or that an error occurred.



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.

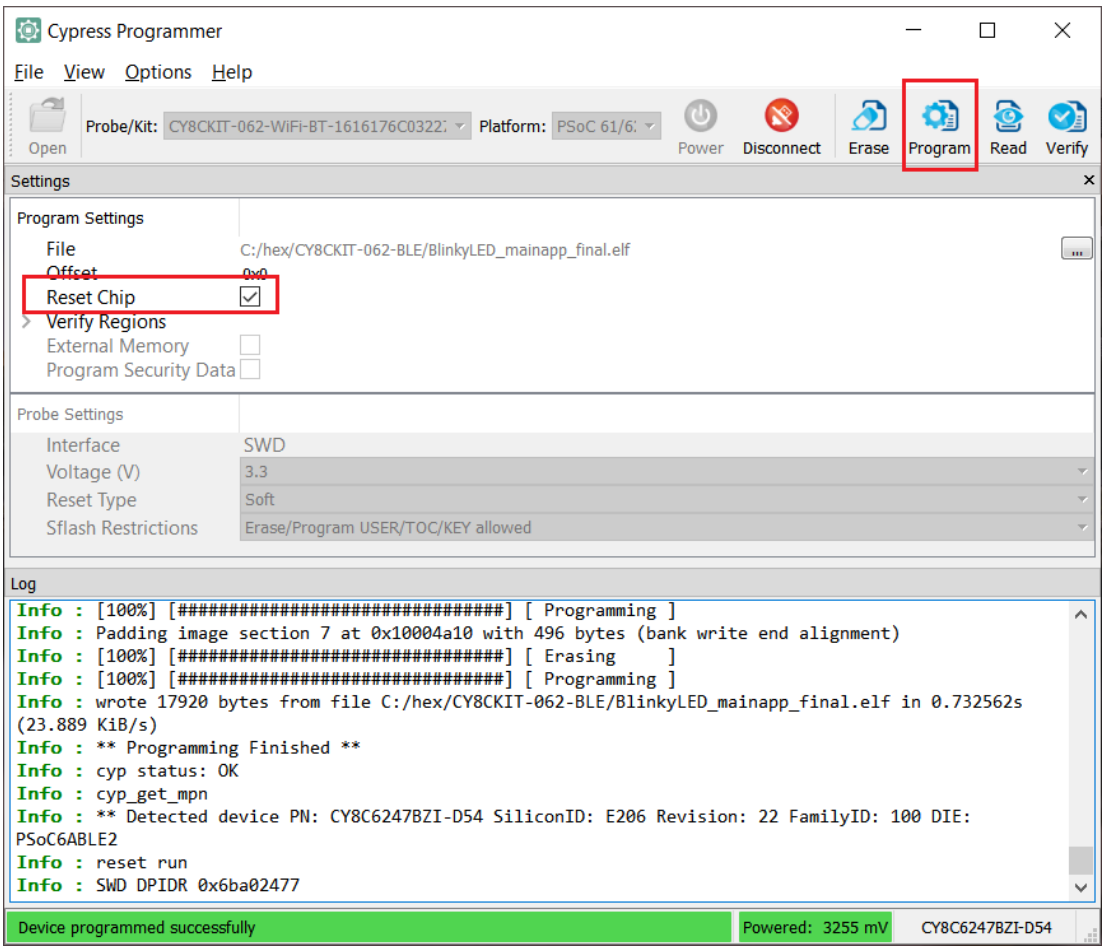
CYP programs the device and displays various messages in the **Log**. Then, a message in the Status Bar indicates that device was programmed successfully or that an error occurred.



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.

CYP 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.

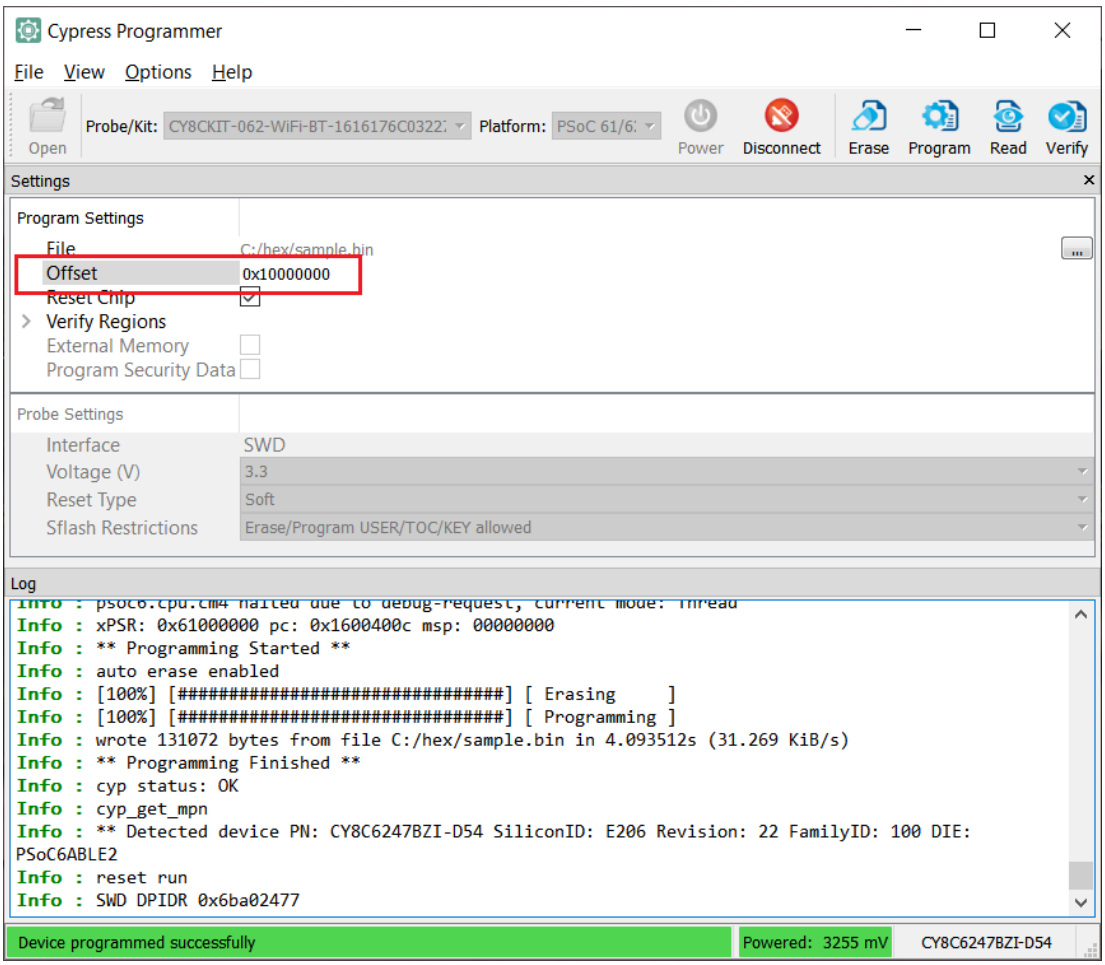


The target device is reset and running.

Program Binary File with Offset

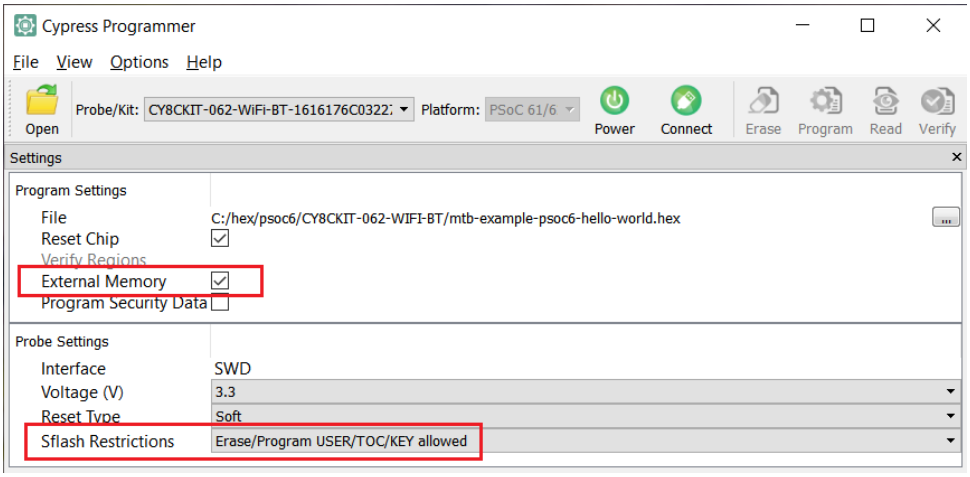
1. Connect the device to the host computer and select it in the **Probe/Kit** drop-down.
2. Select the binary programming file as described in the Load Programming File section.
3. Connect to the device (see Connect Device).
4. Enter the desired address in the **Offset** field under **Program Settings**.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

CYP 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.



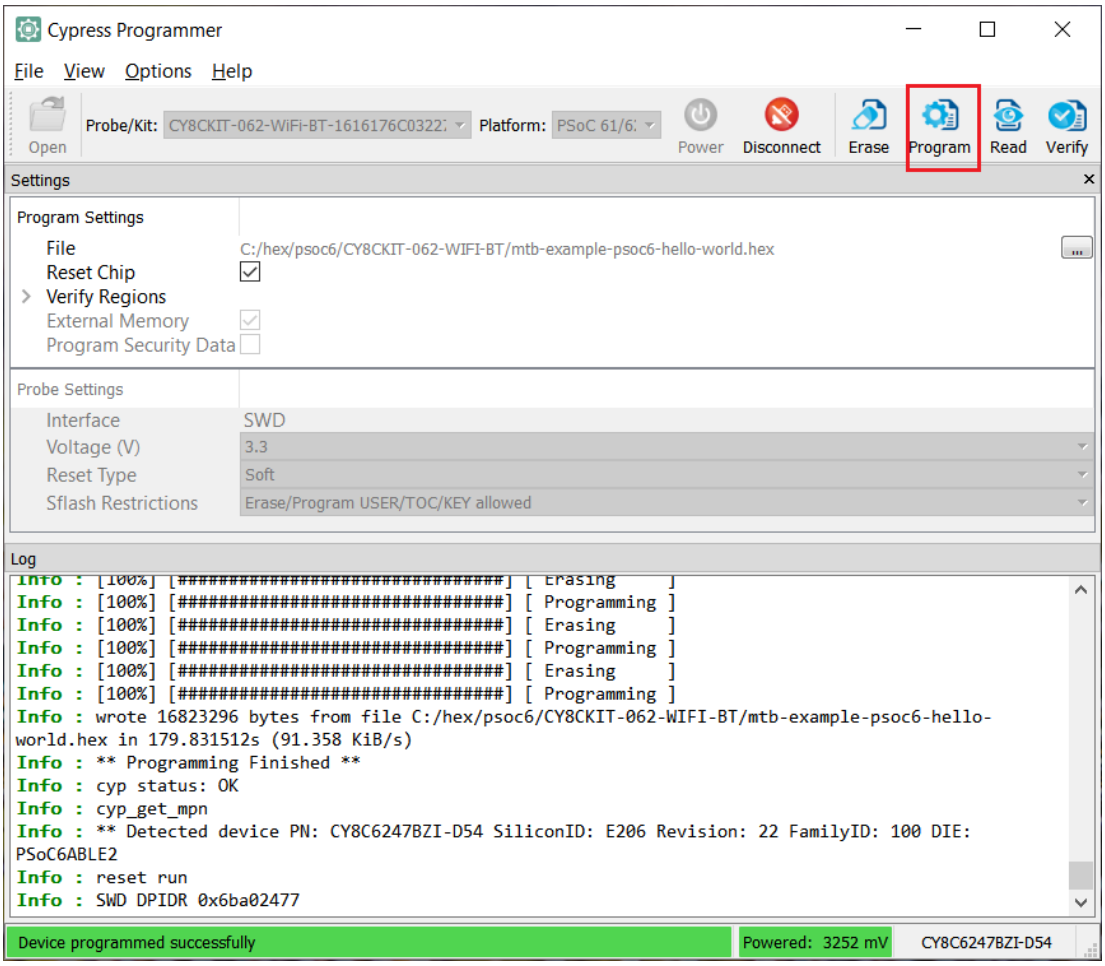
Program External Memory

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.

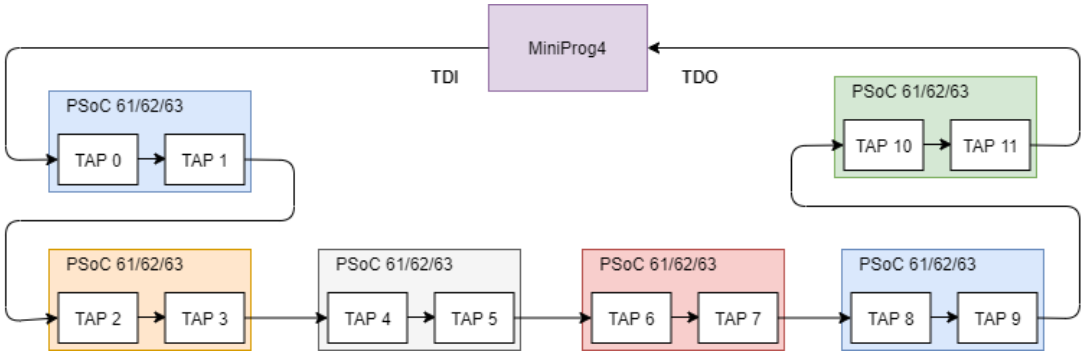
CYP programs the device and displays various messages in the **Log**. Then, a message in the Status Bar indicates that the device was programmed successfully or that an error occurred.



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.

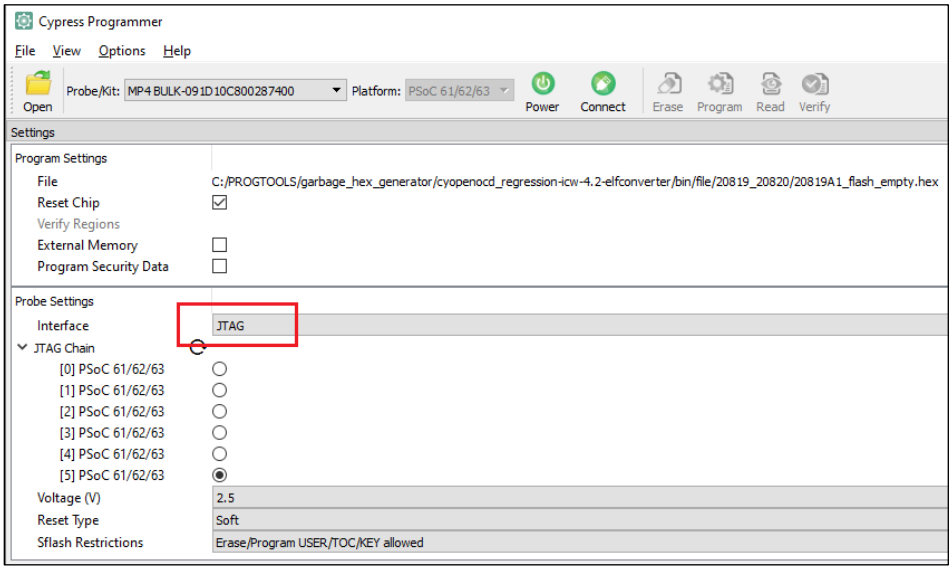
   The following hardware configuration is used in this example:



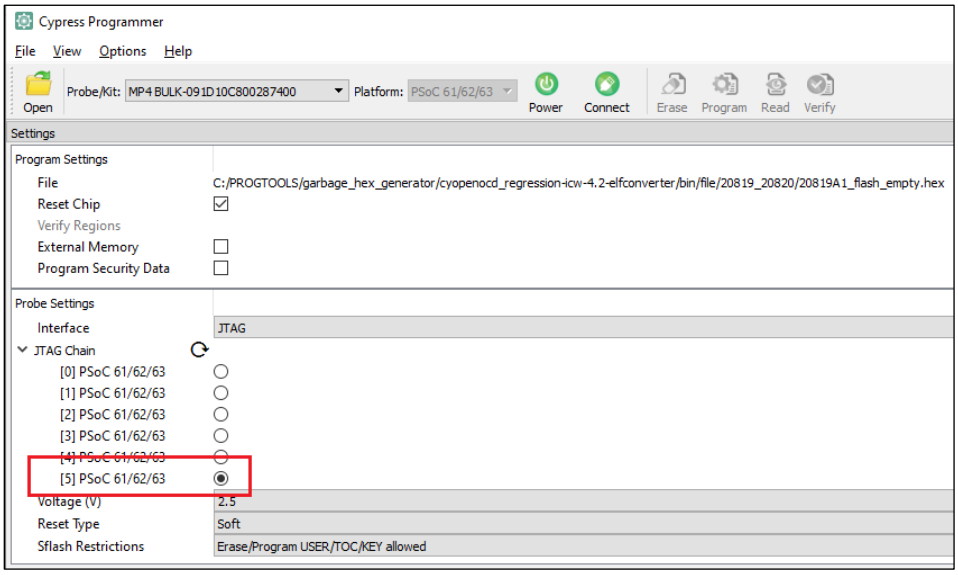
   The sample JTAG chain configuration contains six serially connected PSoC 6 MCU targets.

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

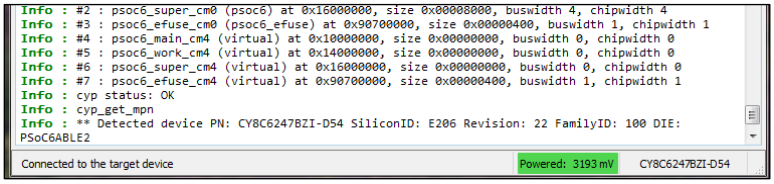
3. Select the JTAG interface in the **Interface** drop-down.



4. CYP 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.

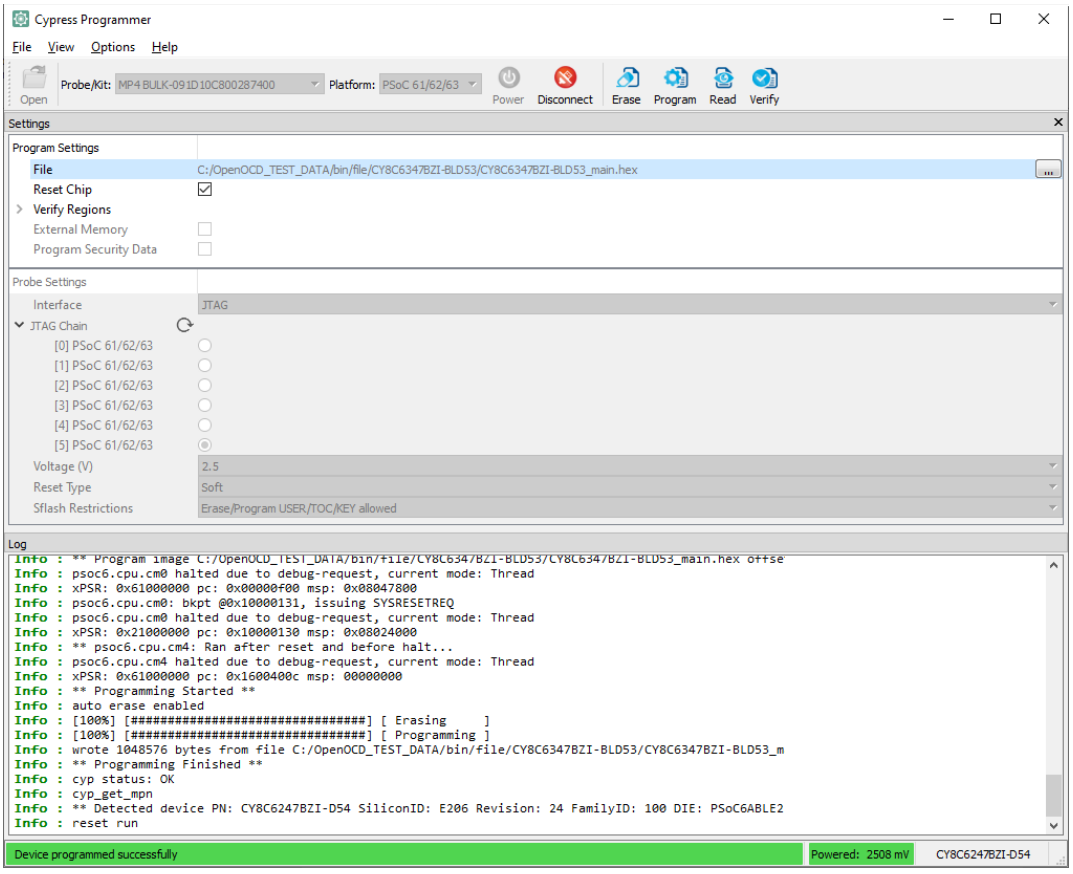


5. Select the desired target device in the list by clicking the radio button next to the target name.
6. Select the programming file as described in the Load Programming File section.
7. Click **Connect**. CYP communicates with the device and displays various messages in the **Log**. Then, a message in the Status Bar indicates that it is connected.



8. Click the **Program** button.

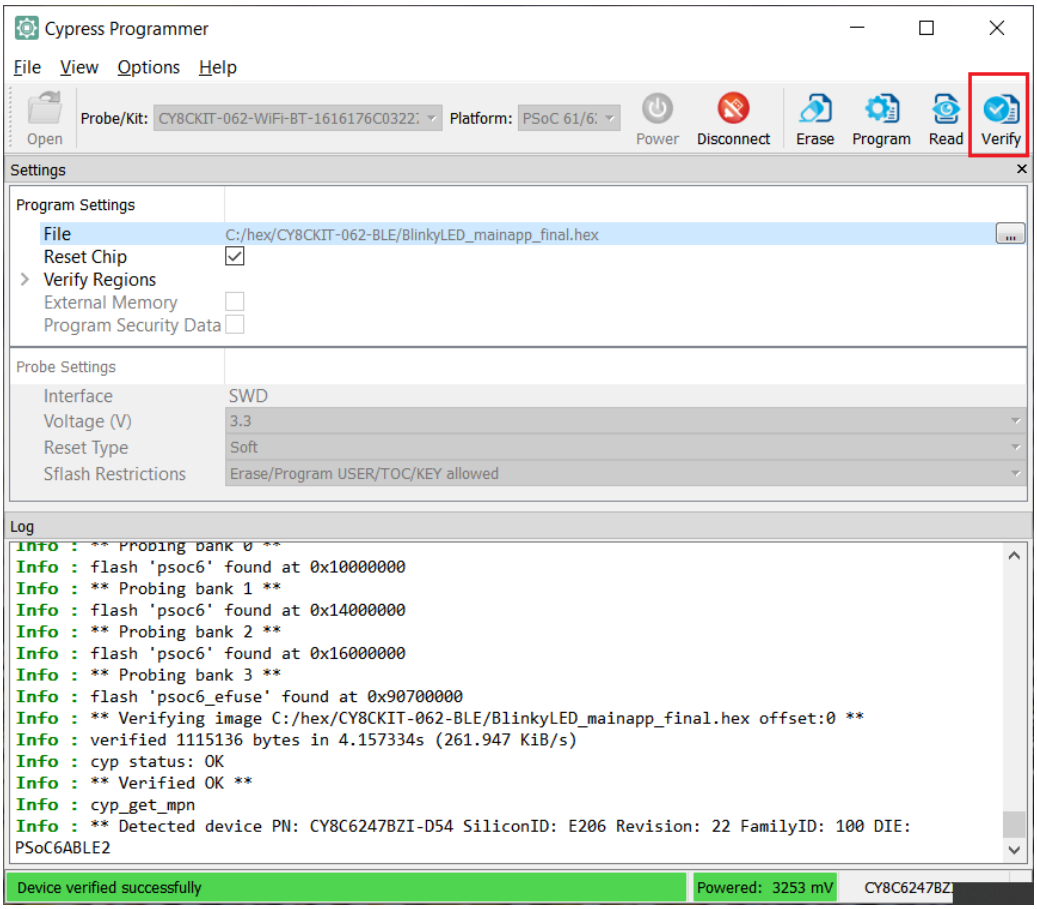
   CYP programs the device and displays various messages in the **Log**. Then, a message in the Status Bar indicates that the device was programmed successfully or that an error occurred.



Verify Device

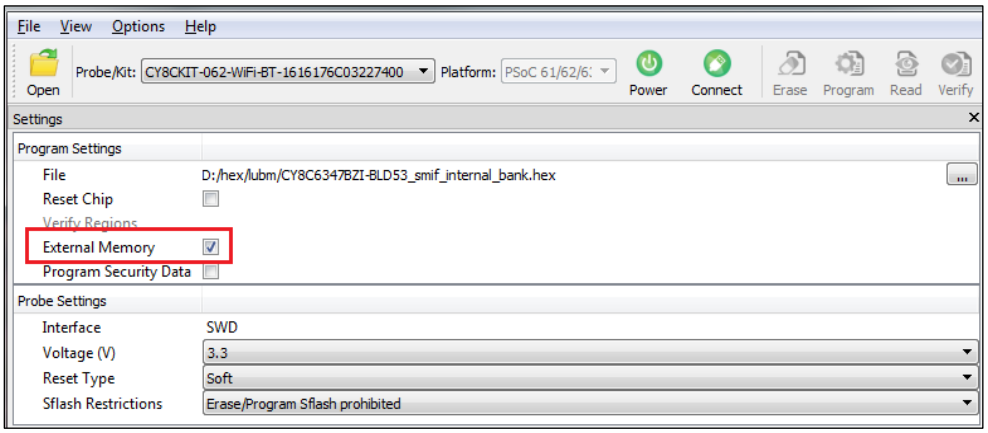
1. Connect the device to the host computer and select it in the **Probe/Kit** drop-down.
2. Select the programming file as described in the Load Programming File section.
3. Connect to the device (see Connect Device).
4. Click the **Verify** button.

CYP 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.



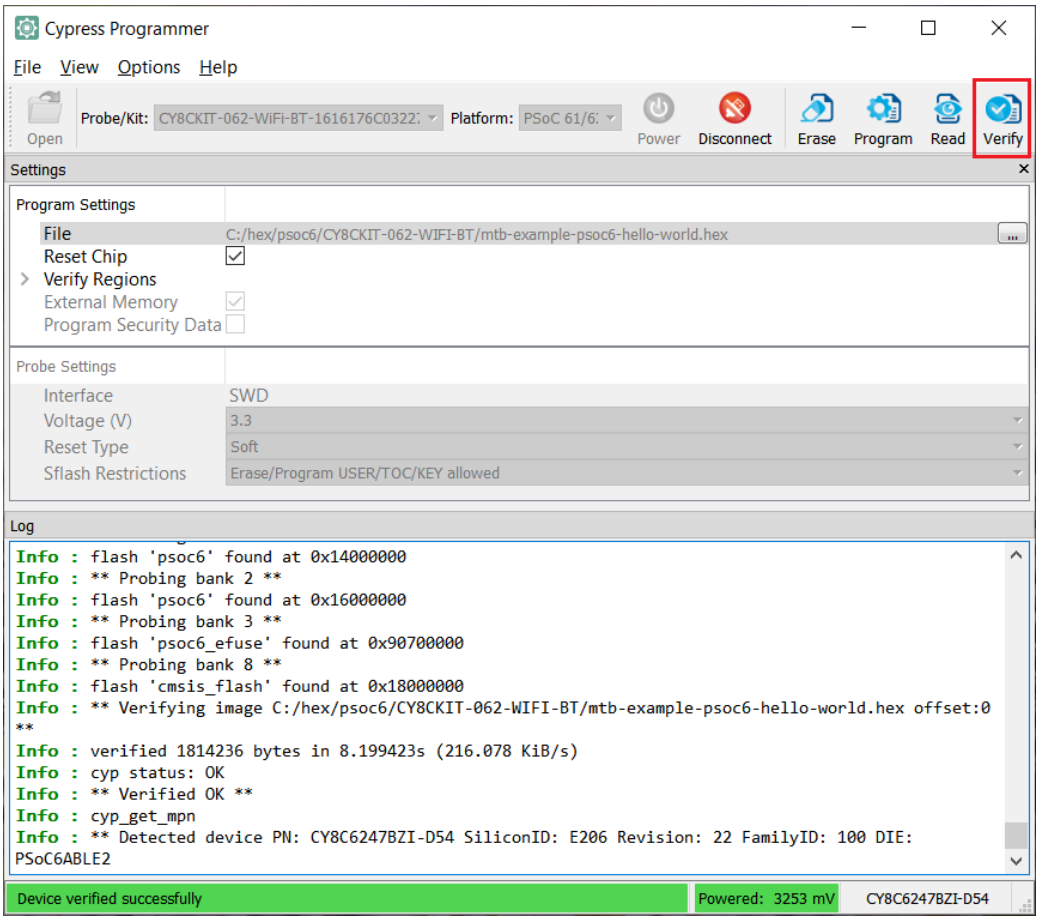
Verify Device with External Memory

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 Probe Settings.



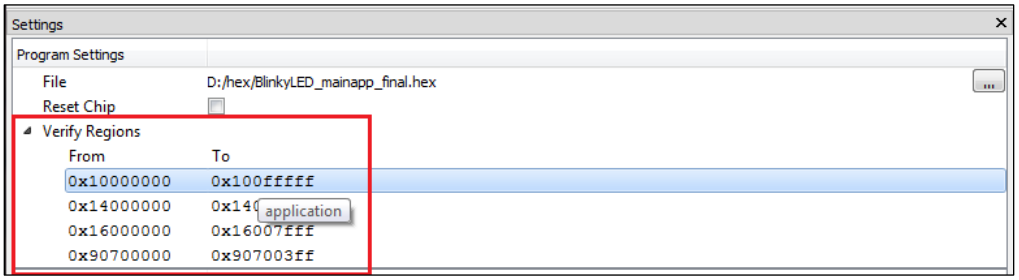
4. Connect to the device (see Connect Device).
5. Click the **Verify** button.

CYP 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.

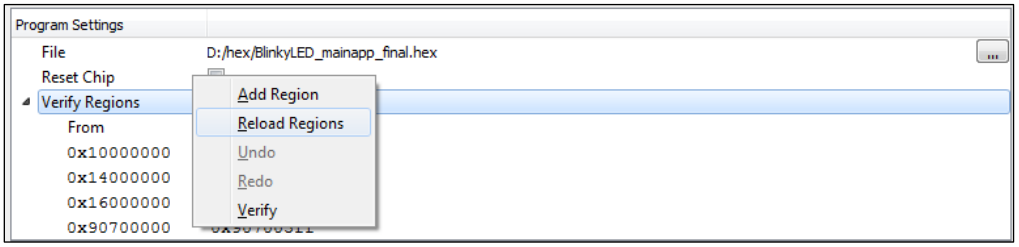


Verify Custom Flash Regions of PSoC 6 MCU

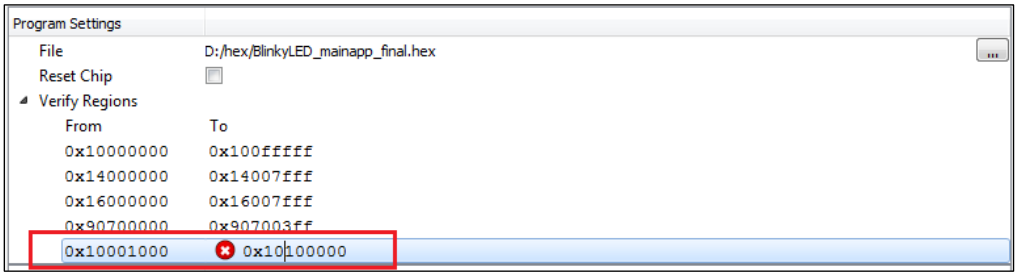
1. Connect the device to the host computer and select it in the **Probe/Kit** drop-down.
2. Select the programming file as described in the Load Programming File section.
3. Connect to the device (see Connect Device).
4. Expand **Verify Regions** option in **Program Settings** to see the list of flash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



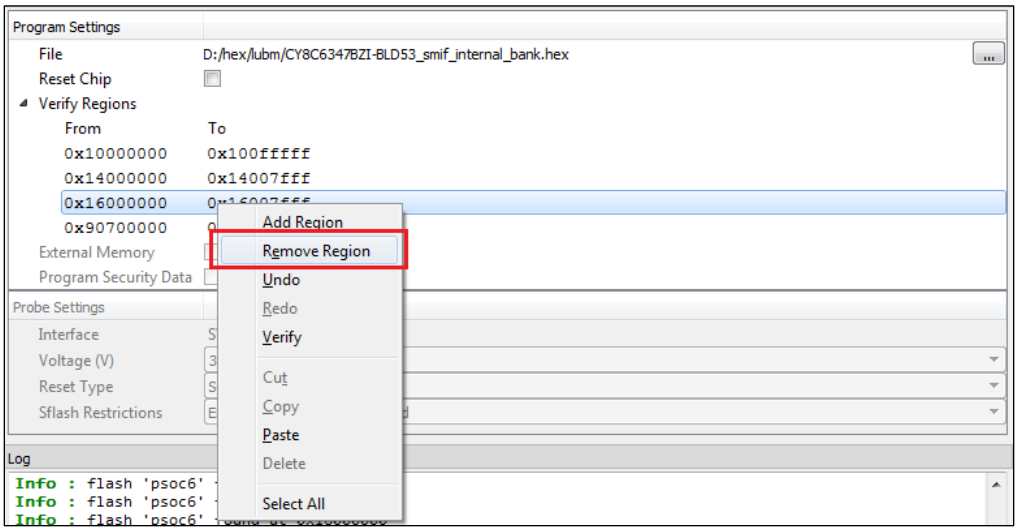
5. Right-click on **Verify Regions,** or any region entry, to open the context menu.



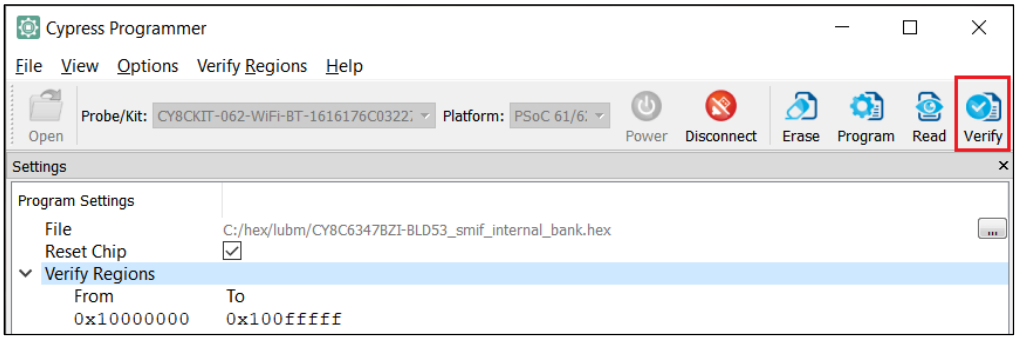
6. To add a custom flash region, select **Add Region**. Select the added list entry, and enter the correct values for start and end addresses of the region.



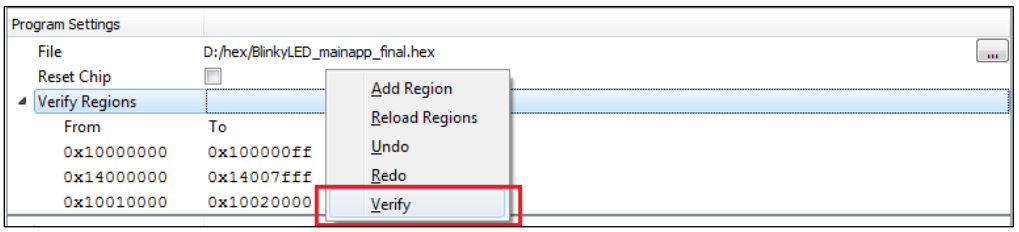
7. To remove any region in the list, right-click the desired region entry and select **Remove Region**.



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.

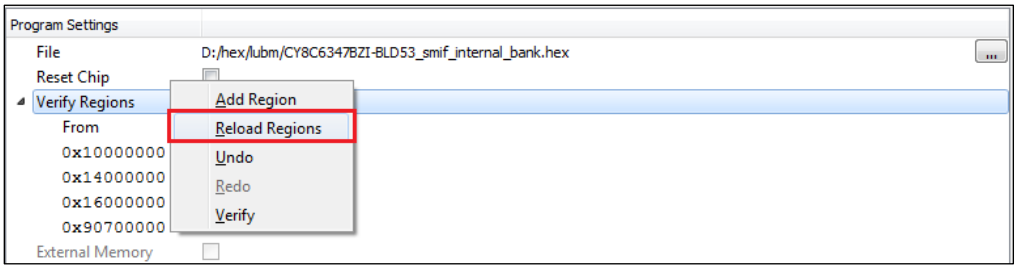


You can also select **Verify** on the context menu.



CYP 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.

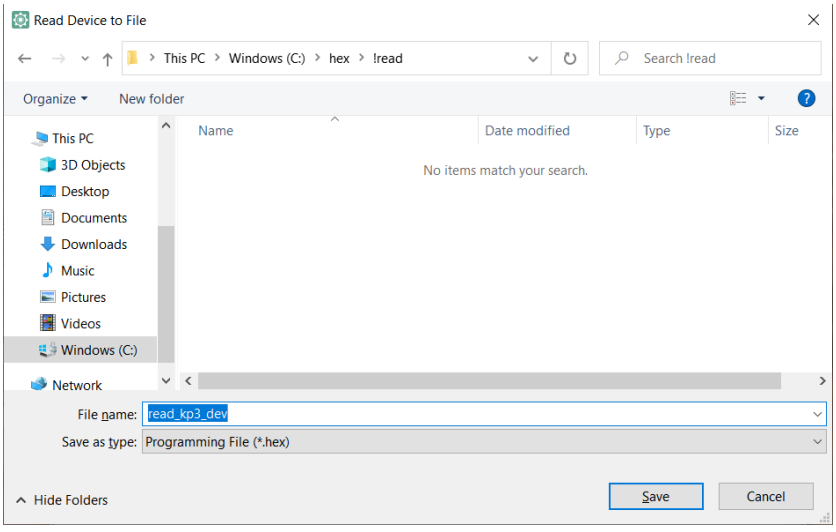


Read Device

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.

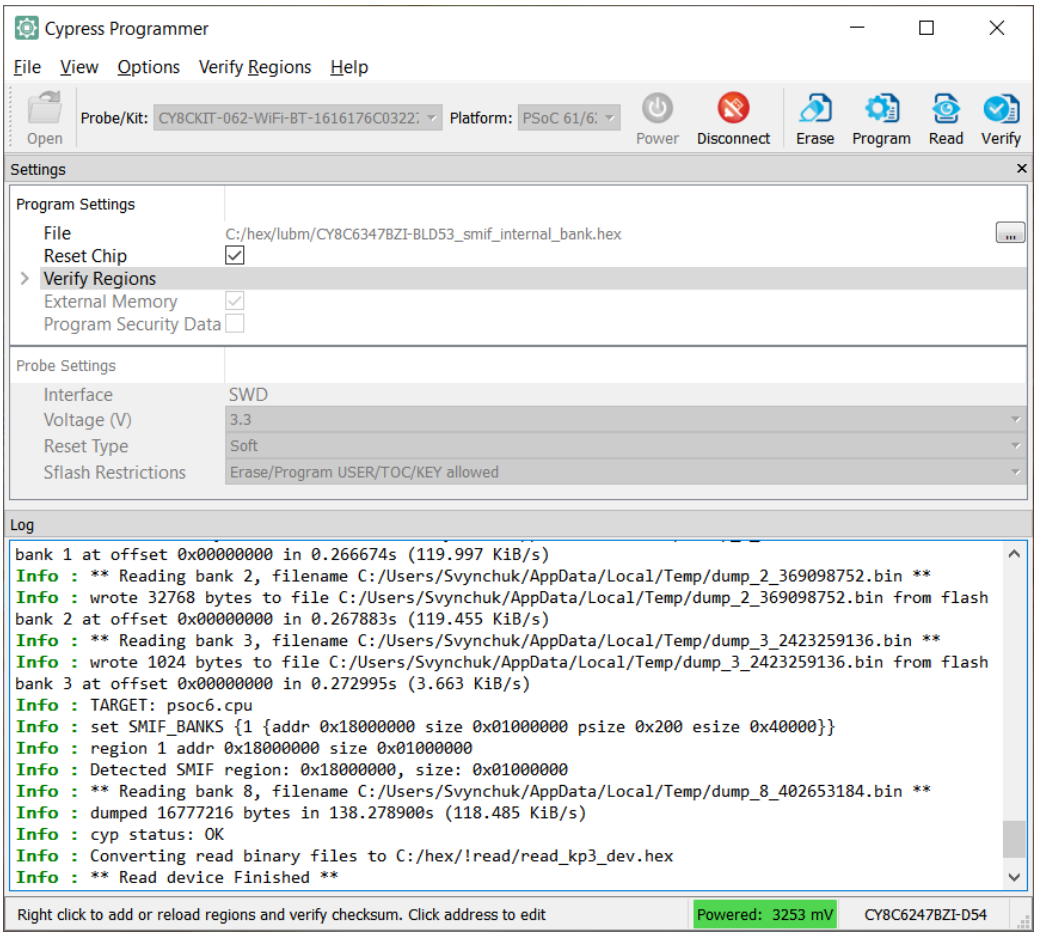


5. On the **Read Device to File** dialog, navigate to the location of the HEX or SREC file to be saved, enter the file name, select the file type in the **Save as type** drop-down, and click **Save**.



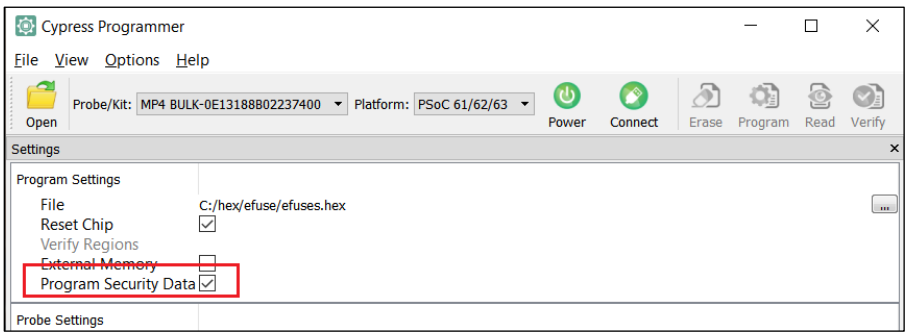
Note Under Ubuntu Linux, specify the full file name with an extension (e.g. *kp3-dev.srec*); otherwise, the file will be saved in HEX format.

CYP 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.



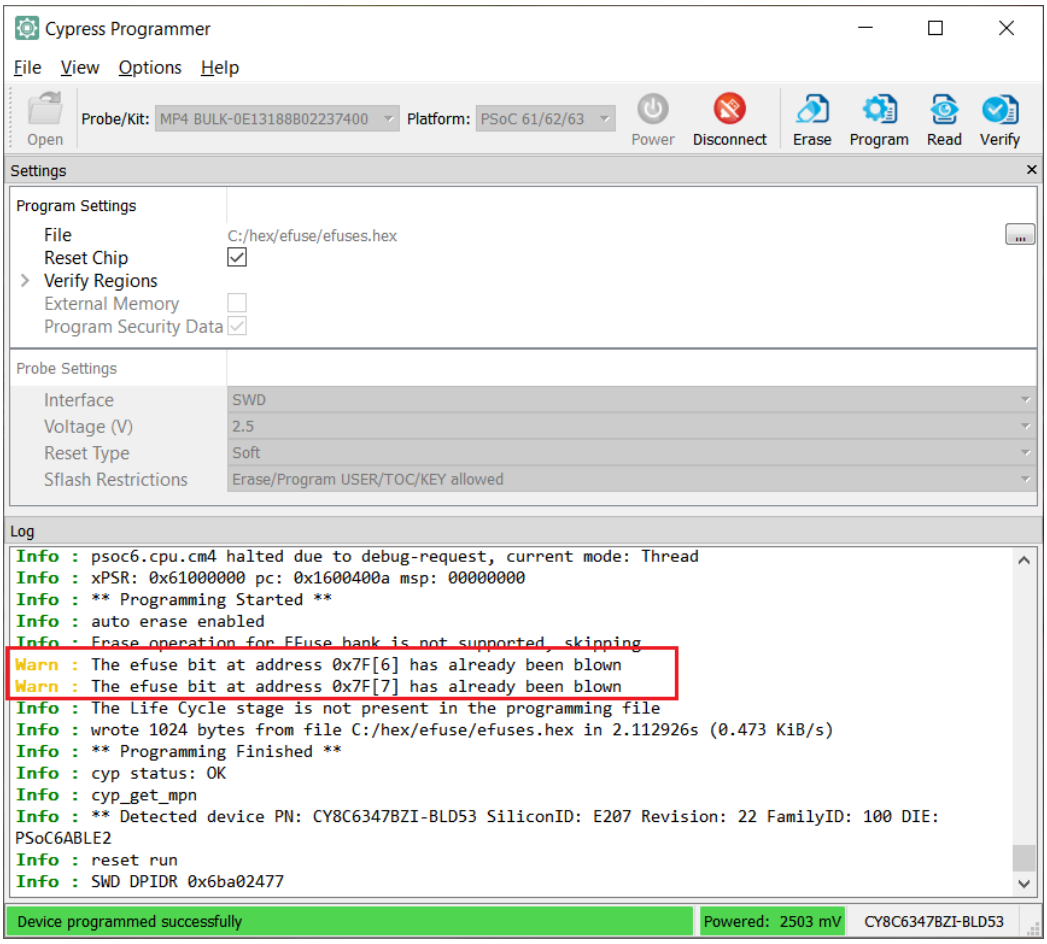
Program eFuse Region of PSoC 6 MCU

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. 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.

CYP 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.

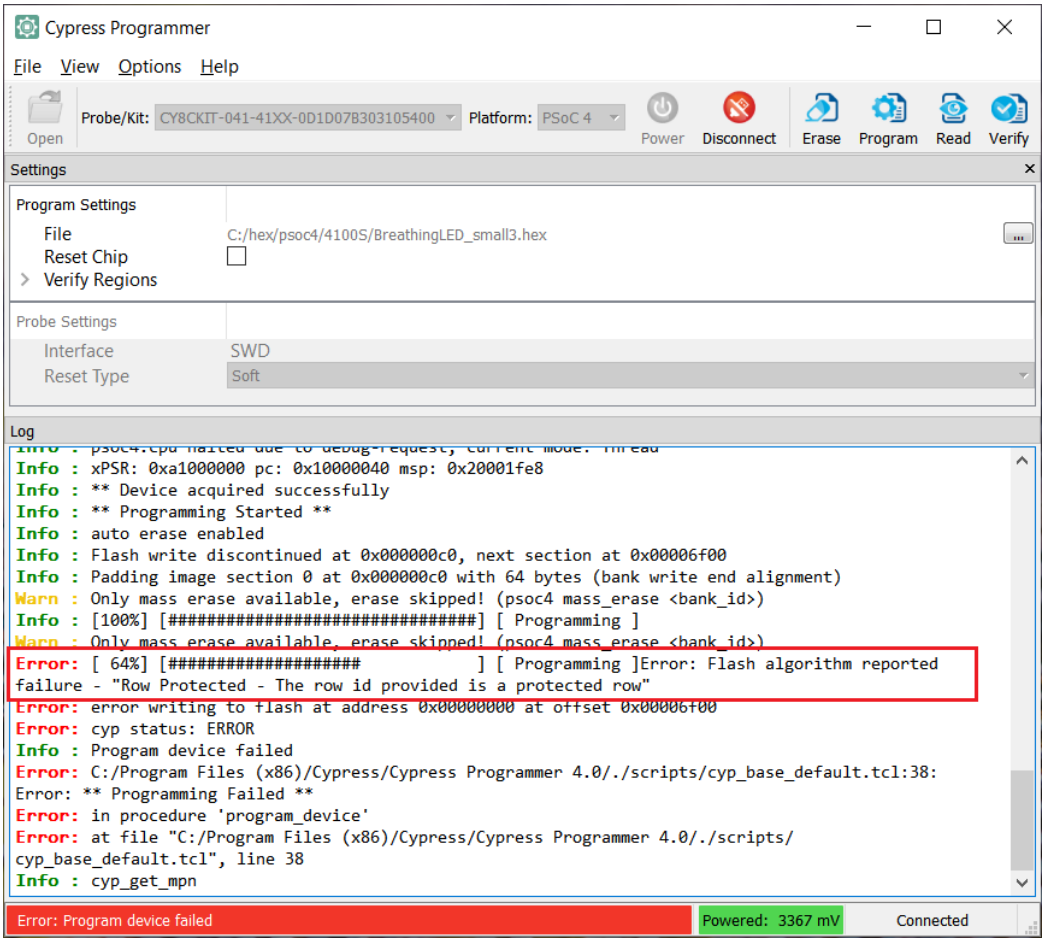


If some eFuse bits have been already programmed before, a warning message “The efuse bit at address xx has been already blown” appears in **Log**.

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 CYP 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.

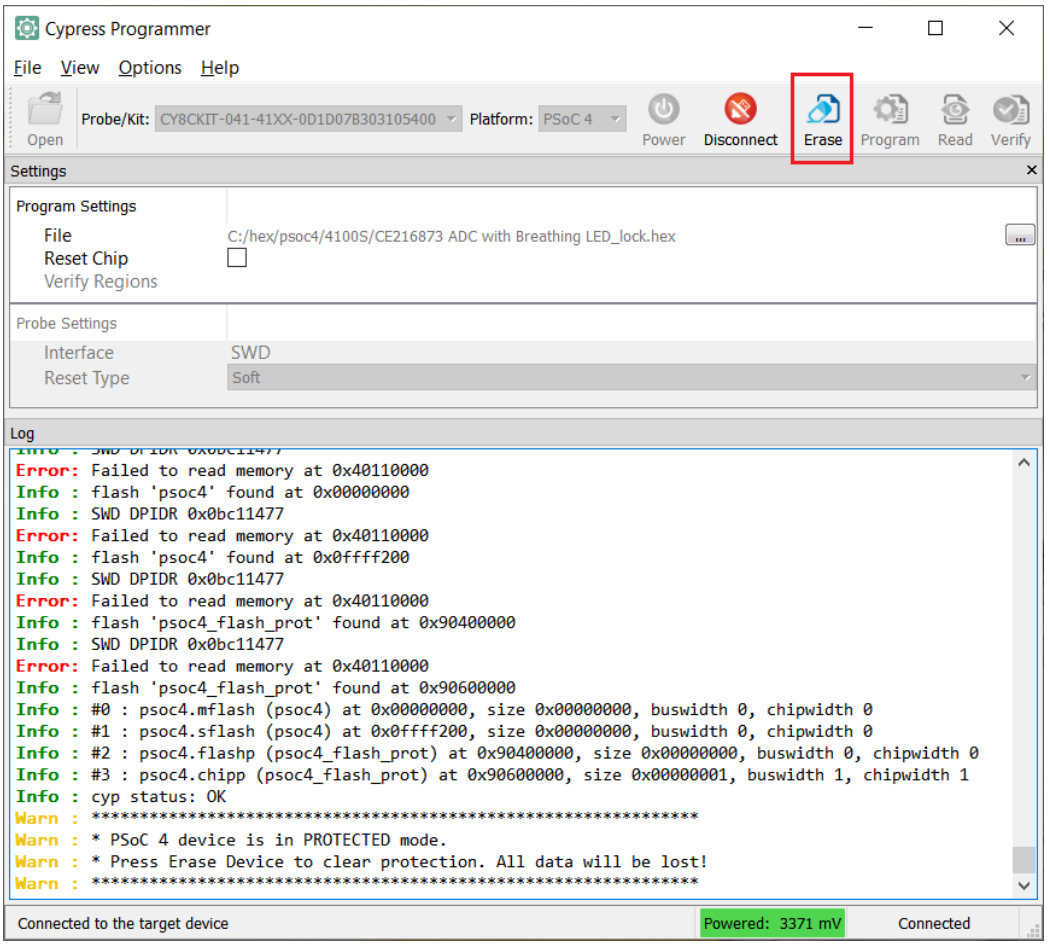


Use the **Erase** button to un-protect user flash rows from writing.

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 CYP to a chip-protected PSoC 4 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.

Limitations

- CYP does not support RAM programming, you can program only flash memory of the target device.
- Programming devices in DAPLink mode is not supported. You have to switch your device into CMSIS-DAP BULK mode by pressing mode selection button.

How to Recover IoT BT Devices on Failure

If the program operation for an IoT BT 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).

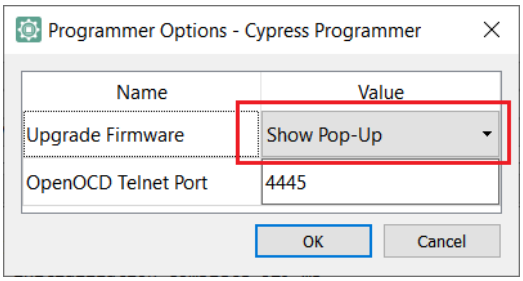
6 Upgrading Firmware

The CYP application allows you to upgrade KitProg2, KitProg3, and MiniProg4 device firmware.

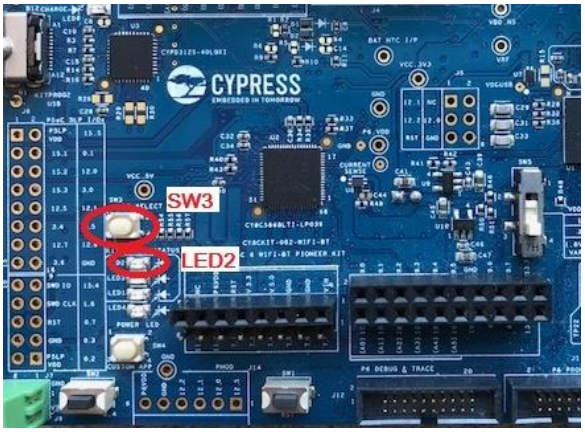
Upgrade KitProg2 Firmware

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

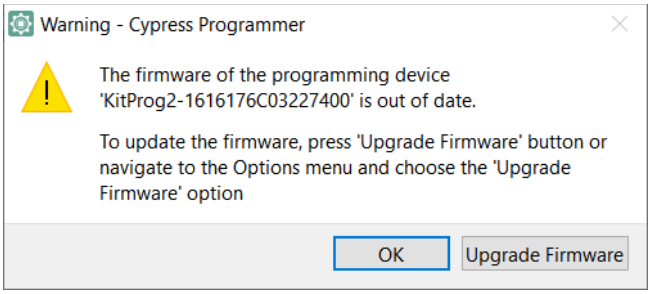
1. Run the CYP application.
2. Go to **Options > Programmer Options** and select the Show Pop-Up value for the **Upgrade Firmware** option.



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.

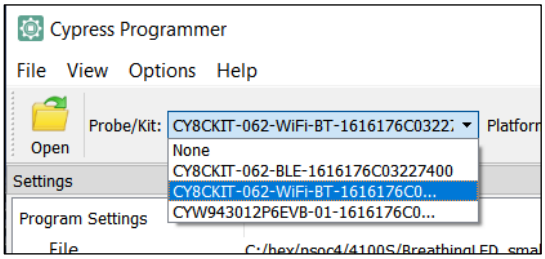


A warning dialog displays with the “Firmware is Out of Date” message.



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

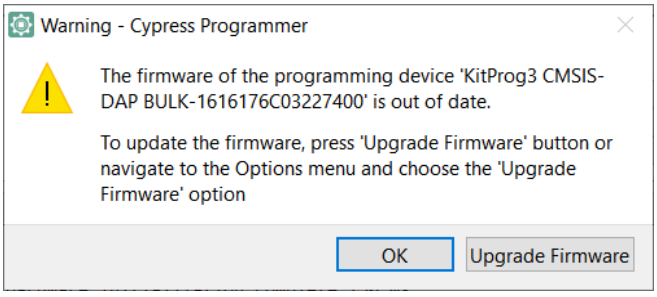
After the upgrade process completes, the message “Firmware of ‘KitProg2-xx' upgraded successfully” displays in the **Log** view. The KitProg2-xx device disappears from the **Probe/Kit** drop-down. The **Probe/Kit** drop-down is populated with the names of the supported KitProg3 Kits.



Upgrade KitProg3 on Cypress Kit or MiniProg4 Firmware

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

1. Run the CYP 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.