

SiLabs/Tiva Programming Guide

By Joseph Haas, KEØFF

8/04/2021

joe~~h~~-at-rollanet-~~dot~~-org

This document centralizes the key items and software required to program SiLabs 8-bit processors. It also features information on programming hardware for the TI Tiva microcontrollers (see page 4).

Programming the SiLabs 8051 microcontroller (MCU) on a target PCB can be accomplished using the SiLabs USB debug adapter, the SiLabs FLASH Programming Utility program, and a source file for the desired application to program.

The Debug adapter can be obtained from Mouser Electronics (www.mouser.com) or DigiKey Electronics (www.digikey.com) using the part number DEBUGADPTR1-USB. The cost (as of this writing) is \$35 plus applicable tax and shipping charges. A “standard” USB “printer” cable is also needed (these cables feature the rectangular “A” style connector that plugs into the PC at one end, and the square-ish “B” style connector that plugs into something like a printer or scanner at the other end). Mouser or Digikey are also good sources for the USB cable. The Qualtek 3021001-03 lists for about \$2.

The Flash Programming Utility software is available at:

<http://www.silabs.com/products/development-tools/software/8-bit-8051-microcontroller-software#flash>

If this link does not work, go to www.silabs.com and search for “flash programming utility”.

Follow the installation instructions on the web site and in the debug adapter documentation.

Connect the debug adapter ribbon cable connector to the target connector either directly or by using an adapter to connect to the Molex 6-pin small-form-factor programming connector (details on this adapter are at the end of this document). To program the MCU, execute the following steps:

- Obtain the object file for the application to program (typically, this is a “*.hex” file)
- Connect the debug adapter to the programming connector on the target
- Plug in the USB cable to the debug adapter and the PC. Some targets may require power to be applied to the target separately, while others may be powered from the USB Programming Adapter.
- Open the flash programming utility software
- See Figures 1 – 3:
- Under the “Download Hex File/Go/Stop” tab, select the object application file (.hex) and check the “Erase all of Code Space” box
- Under the “Connect/Disconnect” tab, make sure all of the check boxes match the image and that the “USB Debug Adapter” is checked (if not, you must make sure the debug adapter is connected and that the drivers are working properly). Click “Connect” and click “OK” when the “Connected” box appears.
- Return to the “Download Hex File/Go/Stop” tab and click “Download” (there should be erase, program, and verify progress screens displayed). Programming should take no more than 15 seconds.
- Return to the “Connect/Disconnect” and click “Disconnect”
- Remove Power from the Orion and disconnect the debug adapter.
- The target device is now ready to operate with the new application software.

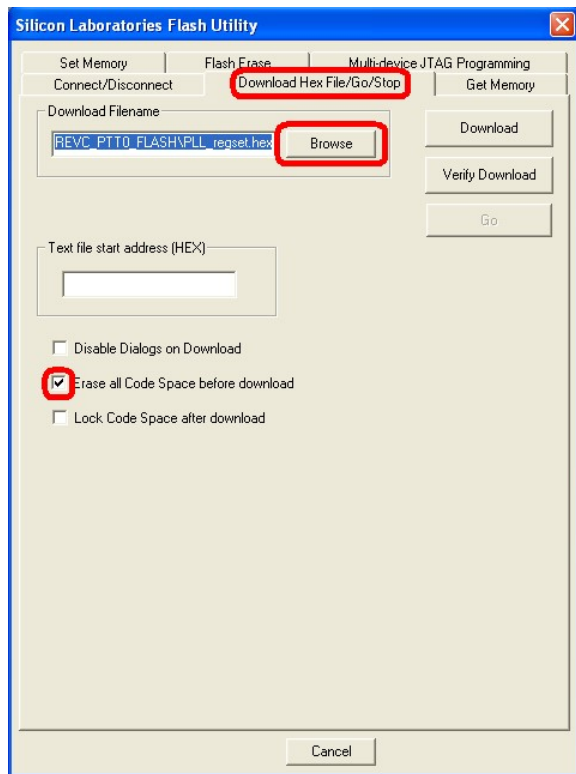


Figure 1. FLASH Utility setup dialog box.

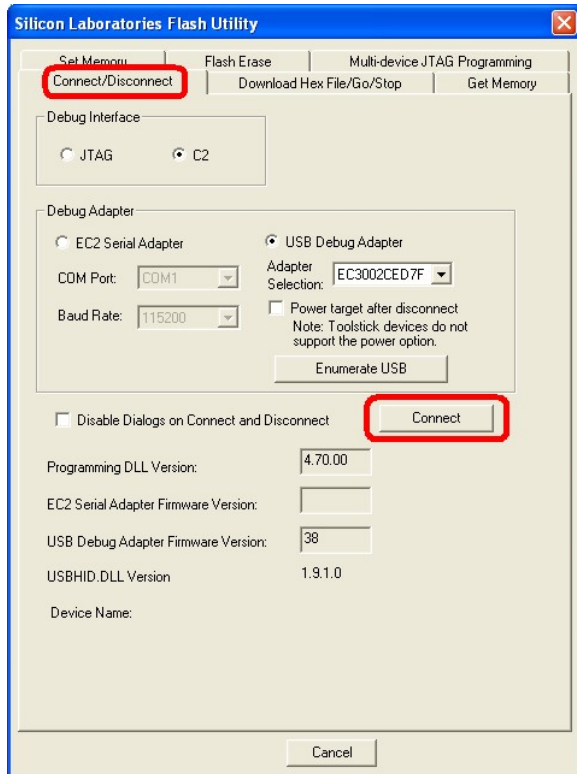


Figure 2. FLASH Utility connect dialog box.

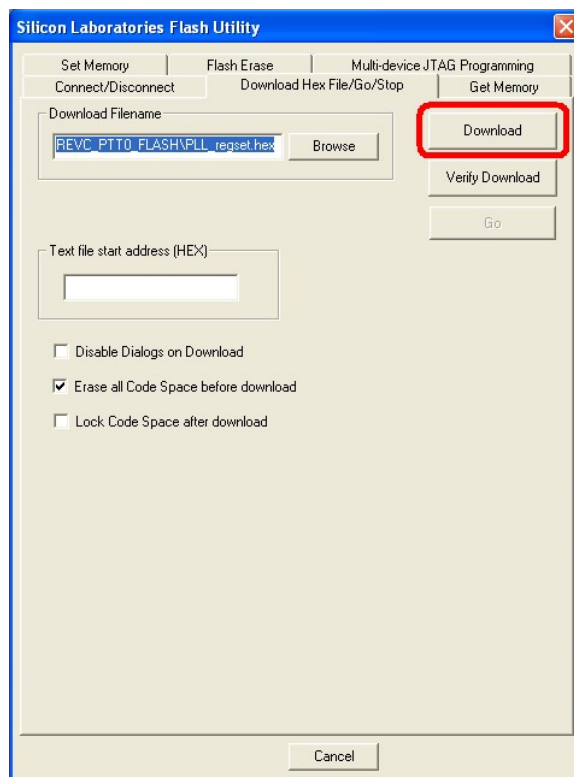


Figure 3. FLASH Utility setup dialog box (download).

Small Form-Factor Programming Connections

Some of my designs may use a small form-factor (SFF) connector to minimize the PCB space required for in-circuit programming capability. A connection pinout that I have standardized for my projects uses a Molex, 1.25mm, 6-position connector, PN 53261-0671 (R/A) or PN 53398-0671 (vertical). This connector has a small footprint and comes in right-angle and vertical versions which can both be soldered to the same PCB footprint. All that is required to connect the SiLabs programming adapter is an intermediate cable which accepts the SiLabs 10-position ribbon cable connector and converts this to a 6-position cable with the mating Molex connector for the PCB connection.

The transition cable (see Photo 1) consists of a 10-pin, dual row, 0.1" spaced ribbon header (TE Connectivity 5103309-1, or equivalent) and a cable terminated with the appropriate 6-position Molex connector (Molex PN 15134-0602, or equivalent). The 15134-0602 cable comes with two ends and should be cut in half (more or less – the other half may be saved or discarded). A small piece of pad-per-hole protoboard (approximately 0.75" square) should be used to stabilize the connectors and wires. Strip and tin the leads approximately 0.1", then form each tinned wire into a "hook". Solder the 10-pin header to the protoboard and then solder the GND net connections (pins 2, 3, and 9 – a piece of ¼W leaded resistor lead can do this nicely) then solder each wire according to Table 1.

<u>Signal</u>	<u>P1</u>	<u>10-pin</u>		<u>P1</u>	<u>Signal</u>
n/c	-	1	2	5	GND
GND	5	3	4	2	C2D
/RESET	1	5	6	4	P0.6
/RST_C2K	3	7	8	-	n/c
GND	5	9	10	6	+5V (note)

Table 1. SiLabs to SFF programming adapter. P1 is the target device connector pinout. *Note: Some targets require +3.3V supply. For these target devices, DO NOT connect +5V to P1-6.*

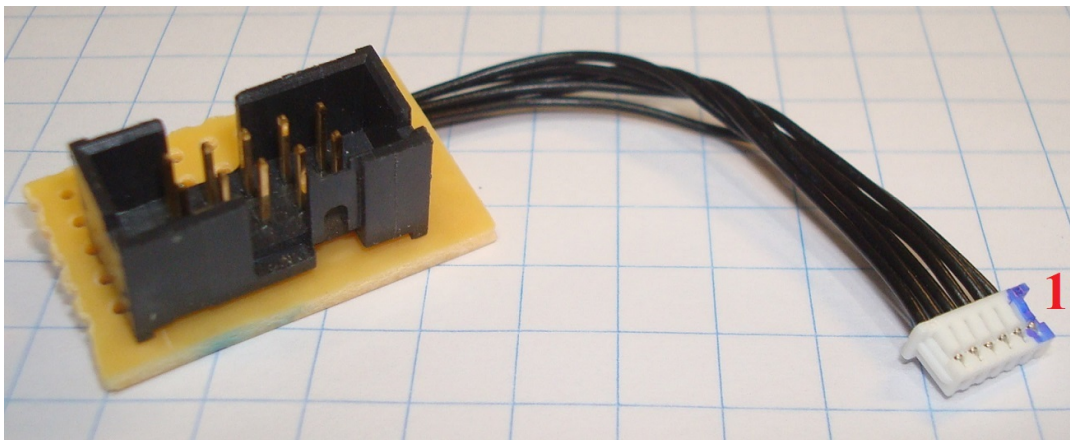


Photo 1. Programming adapter cable.

Tiva Programming Hardware Guide

The small form-factor connector used for the SiLabs processors is also used in my Tiva designs for programming. The connections are similar with a couple of exceptions noted in Table 2.

<u>LP Signal</u>	<u>P1</u>	<u>10-pin</u>		<u>P1</u>	<u>LP Signal</u>
X1-RESET	6	1	2	5	GND
GND	5	3	4	2	X1-TCK
X1-TMS	1	5	6	4	X1-TDO
X1-TDI	3	7	8	-	X1-RXD
GND	5	9	10	-	X1-TXD

Table 2. Tiva to SFF programming adapter. P1 is the target device connector pinout. *Note: X1 is located on the EK-TM4C1294XL LaunchPad.*

Any of the Tiva LaunchPad evaluation boards can be modified to act as a programming interface for a target board. This is generally much less expensive than buying a commercial programmer. Photo 2 illustrates the connection to an EK-TM4C1294XL Launchpad that has had its target MCU removed (this is not necessary to use the LaunchPad as a programmer).

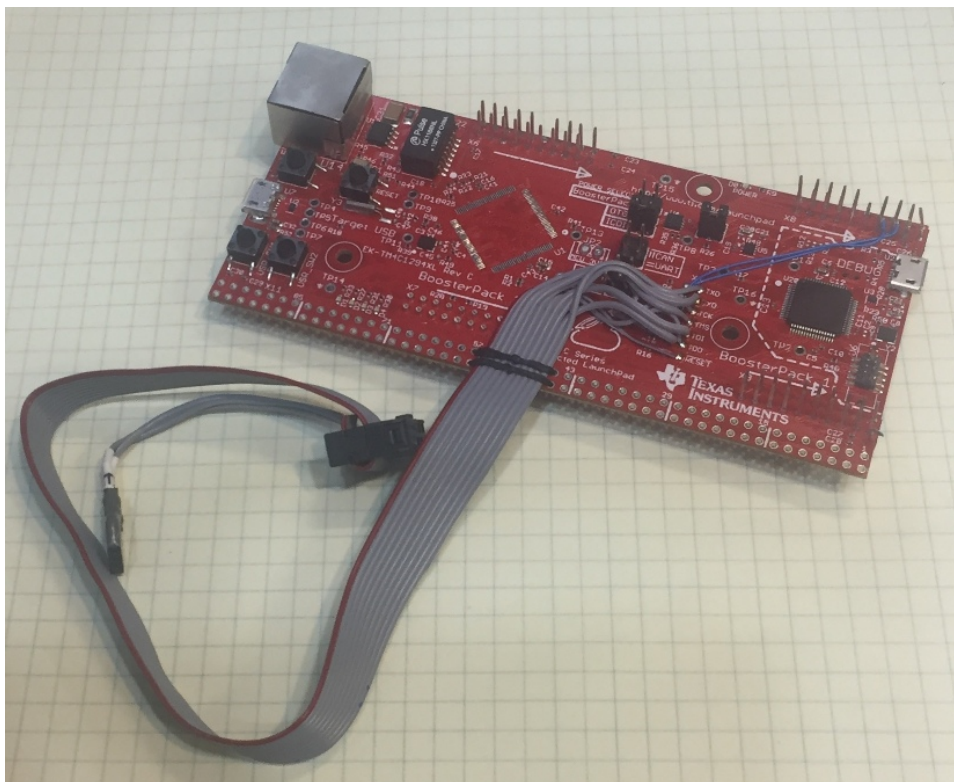


Photo 2. A modified Tiva LaunchPad for use as a programmer (*Note: the programming USB connector is on the right as shown in the photo*).

To modify a LaunchPad, start by fabricating a 10 pin ribbon cable (8", max) with an IDC connector at one end (e.g., Kobi Conn: 164-9006-E (conn) and 3M: 3801/10 (100") (cable)). Separate the wires at the un-terminated end of the ribbon cable, then strip and tin each wire 0.1". *Note: Some LaunchPads feature a*

dual row header with shorting jumpers while others have a hole-pattern for such a header-jumper but have traces connecting the holes, and no header is installed.

For the boards that have no header – cut the traces connecting each pad pair (cut BETWEEN the pads only). Then install a header (so that the LaunchPad target device can be programmed if desired) and solder the ribbon cable wires to the back side of the board on the side closest to the programming MCU. *Note that the LaunchPad shown in Photo 2 has had its dual-row header removed, so the ribbon cable is shown connected to the top side.*

For the boards with a header, remove the jumpers and store them on the same header such that they are secured on only one of the header pins. Solder the ribbon cable wires to the back side of the board on the side closest to the programming MCU.

The LaunchPad programmers also feature a serial port that can be used for debug or user interface with the target device. This connection is generally a 3-pin, 0.1” spacing header on my target boards. To use this serial connection, connect a 3-conductor cable to the appropriate target connector, and solder the appropriate wires to GND, RXD, and TXD on either the SFF 10-pin connector, or on the LaunchPad board. To use the serial port, simply connect the LaunchPad programmer USB connection to the PC, start the terminal emulator of your choice, and select the appropriate COM port.

To use the programmer with Code Composer Studio (<https://www.ti.com/tool/CCSTUDIO>), add or edit a target configuration file to the project and select the Stellaris In-Circuit Debug Interface (along with the target processor) – see Figure 4. To program the Tiva target device, simply start a debug session in Code Composer Studio.

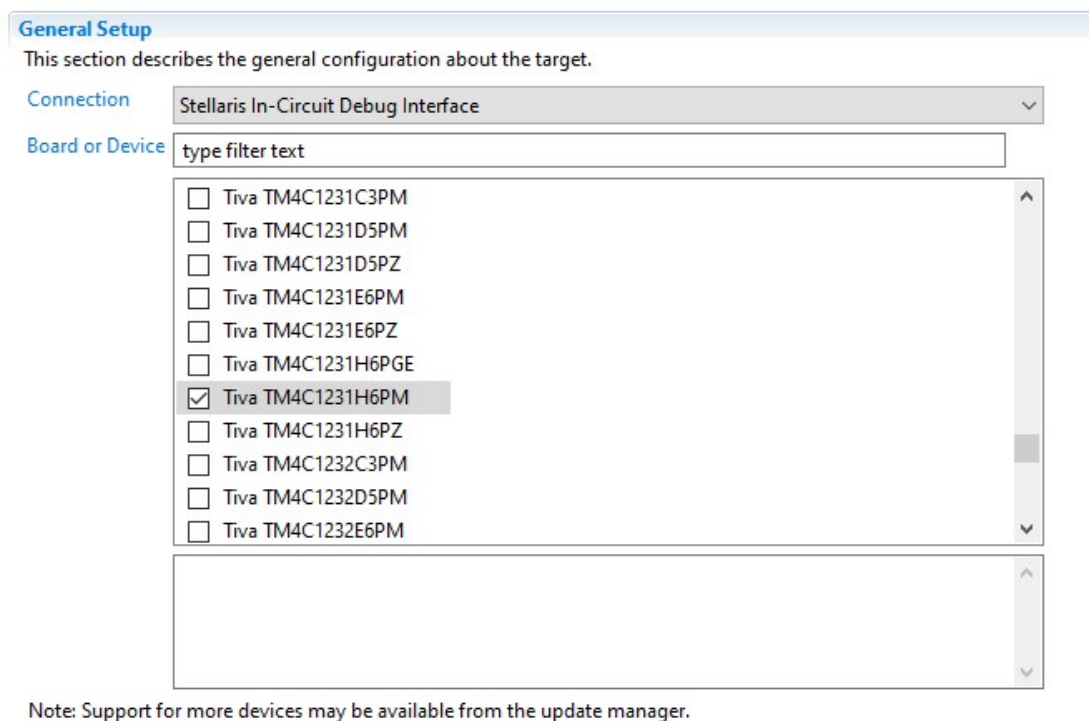


Figure 4. Code Composer Studio target configuration.