TMS320F28xx SDFlash Serial RS232 Flash Programming Reference Guide

Table of Contents

Revision History....3
Limitations and Assumptions....4
Installation Instructions....5
Using SDFlash Serial....9
Rebuilding the SDFlash Serial Files....18
Troubleshooting Tips....25
References....26

Revision History

v3.0: September 01, 2005. *(D. Alter)*
- Added support for F2808, F2806, F2801 using F280x Flash API algorithms v3.00.
- Upgraded F281x API Flash algorithms to v2.10.
- Removed SDFlash executable from bundle.
- Removed Win32 console test program from bundle.
- Consolidated all documentation files into this document
- Fixed linker command file for F281x algorithm build to properly include all .obj files.
- Adjusted descriptions and default values for the ERASE TIMEOUT, PROGRAM TIMEOUT, and VERIFY TIMEOUT SDFlash project options.
- Added use of Erase User Options #4 and Programming User Options #4 to specify the Host PC loop retry for erase and programming acknowledgement.

v2.2: October 15, 2004. Original release.
- Support for F2812, F2811, F2810 using F281x Flash API algorithms v1.00.

Limitations and Assumptions

The SDFlash RS232 Serial patch is provided by Spectrum Digital and Texas Instruments as an “as is” proof of concept example of serial programming for the F28xx flash DSPs. The utility as well as the source code is provided as is, without any technical support.

The patch requires use of SDFlash v1.60 or later. These versions of SDFlash support a generic communications interface that is used to communicate over the RS232 serial link.

All of the source code, project files, and documentation needed to build the .dll files for SDFlash (e.g., F281xRS232Flash.dll and F280xRS232Flash.dll), the flash programming DSP download F28yySerialFlash.out (e.g., F2812SerialFlash.out for F2812 device), and the code security password match flow DSP download (e.g., F28xxPasswordMF.out) are available in the following directories:

TMS320F281x devices: <SDFlash_install_dir>\sdflash\mydrivers\DSP281x_v3_0

TMS320F280x devices: <SDFlash_install_dir>\sdflash\mydrivers\DSP280x_v3_0

The v3.0 SDFlash serial utility supports the following devices:

TMS320F281x devices: F2812, F2811, F2810

TMS320F280x devices: F2808, F2806, F2801

The SDFlash driver files F281xRS232Flash.dll and F280xRS232Flash.dll were built using Microsoft Visual C++ v6.00. The flash programming DSP download F28yySerialFlash.out and the code security password match flow DSP download F28xxPasswordMF.out were built using Code Composer Studio™ v3.1 with C28xx C compiler v4.1.0.

This document assumes the reader has some familiarity with the following:

- SDFlash v1.60 or later for JTAG programming of the F28xx family.
- The “TMS320F281x Flash API” (SPRC125), or the “TMSF280x Flash API”(SPRC193) available on TI’s website, http://www.ti.com

---

™ Code Composer Studio is a trademark of Texas Instruments Incorporated.

Installation Instructions

Step 1. Install SDFlash V1.60 or higher.

In version 1.60 and above Spectrum Digital exposed a generic interface that can support programming via non-JTAG interfaces. The latest SDFlash can be downloaded from the Spectrum Digital website.

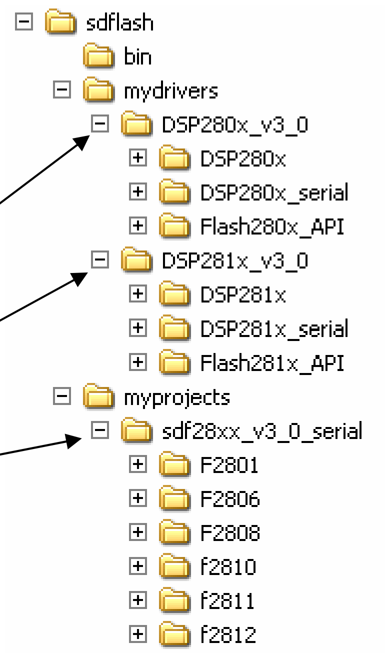
- Go to www.spectrumdigital.com and register, or login if you have previously registered. The login button is at the top right of the webpage.
- Click *"Drivers and Config Utilities"*
- Click *"C2000 Products"* on the list on the left-hand side of the screen
- Click *"Utilities"* on the list on the left-hand side of the screen.
- Click *"Install/Download"* for the SDFlash utility. This will open the following webpage:

  http://emulators.spectrumdigital.com/utilities/sdflash/
- Download and install on your PC the latest SDFlash utility (at least v1.60).

Step 2. Install the SDFlash algorithms for F28xx flash programming via an RS232 serial link.

- Go to www.spectrumdigital.com and register, or login if you have previously registered. The login button is at the top right of the webpage.
- Click *"Drivers and Config Utilities"*
- Click *"C2000 Products"* on the list on the left-hand side of the screen
- Click *"Utilities"* on the list on the left-hand side of the screen.
- Click *"Install/Download"* for the SDFlash utility. This will open the following webpage:

  http://emulators.spectrumdigital.com/utilities/sdflash/
- Download the latest SDFlash algorithms for F28xx flash programming via an RS232 serial link.
- It is recommended to install the patch in the root directory of your SDFlash installation. For example, if SDFlash is installed in *C:\CCStudio_v3.1\specdig\sdflash*, unzip the patch in the *C:\CCStudio_v3.1\specdig\sdflash*. This will create three new sub-directories under *mydrivers* and *myprojects*:



mydrivers\DSP280x_v3_0

mydrivers\DSP281x_v3_0

myprojects\sdf28xx_v3_0_serial

- The directories *mydrivers/DSP280x_v3_0* and *mydrivers/DSP281x_v3_0* contain the RS232 interface example code for F280x and F281x devices, respectively. This code is supplied as an as-is example of how to perform programming over the RS232 communications channel. This sample code is provided without technical support.
- The directory *myprojects/sdf28xx_v3_0_serial* contains all of the files specific to executing serial programming. Each flash device currently in the F28xx family has it's own specific sub directory.

  The files required to use SDFlash over a serial link are:

  - The SDFlash serial project file for the device in use:
    - F2812SerialFlash.sdp
    - F2811SerialFlash.sdp
    - F2810SerialFlash.sdp
    - F2808SerialFlash.sdp
    - F2806SerialFlash.sdp
    - F2801SerialFlash.sdp
  - The SDFlash driver file for the device in use:
    - F281xRS232Flash.dll (for F281x devices)
    - F280xRS232Flash.dll (for F280x devices)
  - The SDFlash serial flash algorithm file for the device in use:
    - F2812SerialFlash.out
    - F2811SerialFlash.out
    - F2810SerialFlash.out
    - F2808SerialFlash.out
    - F2806SerialFlash.out
    - F2801SerialFlash.out
  - The code security password match flow file (applies to F280x only):
    - F28xxPasswordMF.out
  - The CCS board.dat file. The contents of this file is not actually used, but rather is needed to satisfy a project option in SDFlash.
    - ccBrd028x.dat
  - The sdopts.cfg setup parameters.
    - AddToSdOpts.cfg

- The .out file containing the user code/data that is to be programmed into the DSP flash. This file is produced from the user's source code by the code generation tools in Code Composer Studio. For test purposes, a sample .out file is provided for each device that will load each flash locations with a value equal to the least-significant 16 bits of the address. For example, address 0x3F7123 in flash will be programmed with the value 0x7123. The code security password locations at addresses 0x3F7FF8 – 0x3F7FFF have been deliberately left unprogrammed (i.e., they will be left as 0xFFFF, which are dummy passwords). The sample .out files are:
    - ❖ F2812FlashTest.out
    - ❖ F2811FlashTest.out
    - ❖ F2810FlashTest.out
    - ❖ F2808FlashTest.out
    - ❖ F2806FlashTest.out
    - ❖ F2801FlashTest.out

Step 3. **Modify and rebuild the SDFlash serial algorithm file, if necessary** (i.e., file F28yySerialFlash.out, where yy is specific to the device you are using). The SDFlash v3.0 serial algorithm software is configured for the following:

– F281x API v2.10 for F281x revision C and later silicon, or F280x API v3.00 for F280x silicon.

– A crystal/oscillator for the DSP of 30 MHz for F281x, and 20 MHz for F280x. This is because the software will configure the DSP for PLL = x10/2, and the flash API algorithms are configured for 150 MHz operation for F281x, and 100 MHz operation for F280x.

– All code security passwords assumed to be erased (0xFFFF).

If your system does not meet the above configuration, you must make changes to the SDFlash serial algorithm file and rebuild it. See the section "Rebuilding the SDFlash Serial Flash Algorithm File" later in this document for information on rebuilding this file.

IMPORTANT: *It is strongly recommended that the Toggle Test first be run on your board to confirm that the flash algorithms are configured properly for your system. See the section "Using SDFlash Serial," Step 6, Erase Tab options, User Options 2 for information on the Toggle Test.*

Step 4. **Setup the sdopts.cfg file.** Search on your PC for the file sdopts.cfg. On a Windows XP PC, this file will be found in the C:\WINDOWS\SYSTEM32 directory. Spectrum Digital tools store emulator setup and configuration information in sdopts.cfg which is generally configured by SDConfig. SDConfig will NOT configure a non-JTAG tool so this has to be done manually. Add the following text to sdopts.cfg **before** the existing comment line "# End of sdopts.cfg".

```
[EmulatorId=C1]
EmuPortAddr=0xC1
EmuPortMode=RS232
EmuProductName=SERIAL_FLASH

[EmulatorId=C2]
EmuPortAddr=0xC2
EmuPortMode=RS232
EmuProductName=SERIAL_FLASH

[EmulatorId=C3]
EmuPortAddr=0xC3
EmuPortMode=RS232
EmuProductName=SERIAL_FLASH

[EmulatorId=C4]
EmuPortAddr=0xC4
EmuPortMode=RS232
EmuProductName=SERIAL_FLASH
```

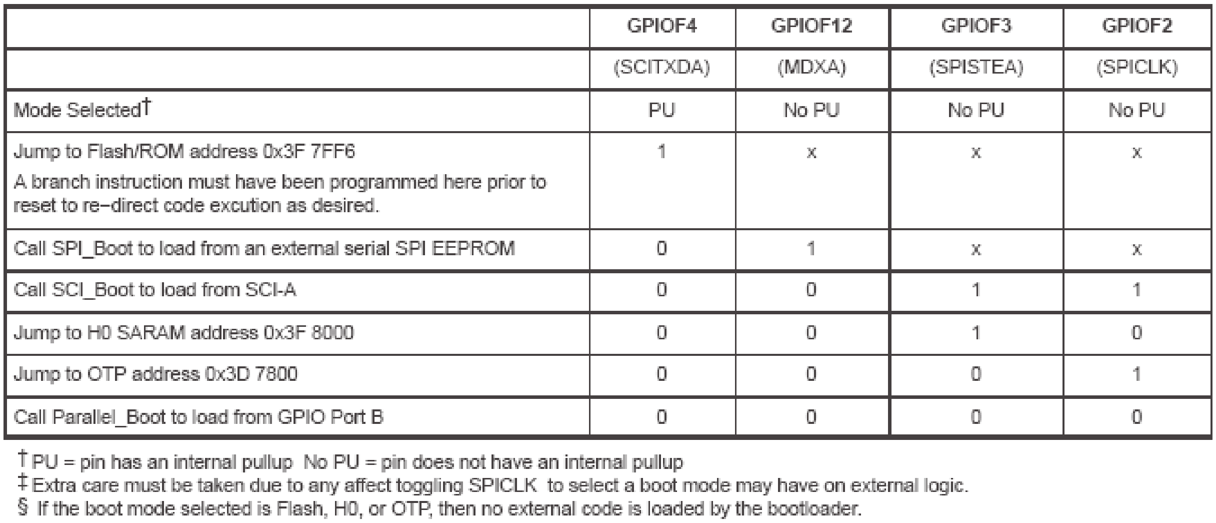
This text can be conveniently copied and pasted from the file *AddToSdOpts.cfg*.

Using SDFlash Serial

Step 1. **Connect a serial cable between the DSP serial port (SCI-A) and the host PC.** The connection must be made through a RS232 transceiver. Note that the eZdsp™ F2808 development board does come equipped with this transceiver, but the eZdsp F2812 development board does not.

Step 2. **Set the F28xx device for boot to SCI-A serial bootload**. This entails having specific GPIO pins high/low at device reset. The eZdsp boards have jumpers or dip switches that allow one to configure the boot mode (see the user's manual for the eZdsp you are using for details). On your own target board, you will need to make arrangements to configure the boot mode, if necessary. The following two tables show the GPIO pin settings needed to configure the boot modes.

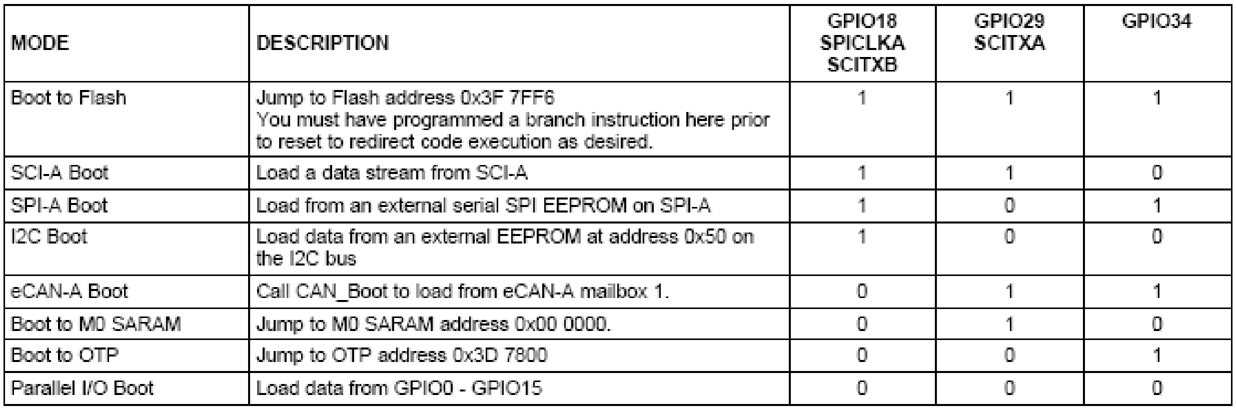
For F281x devices, configure the GPIO according to the following boot mode table (excerpt from the TMS320F2812/2811/2810 datasheet, SPRS174):



	GPIOF4	GPIOF12	GPIOF3	GPIOF2
	(SCITXDA)	(MDXA)	(SPISTEA)	(SPICLK)
Mode Selected†	PU	No PU	No PU	No PU
Jump to Flash/ROM address 0x3F 7FF6 A branch instruction must have been programmed here prior to reset to re-direct code excution as desired.	1	x	x	x
Call SPI_Boot to load from an external serial SPI EEPROM	0	1	x	x
Call SCI_Boot to load from SCI-A	0	0	1	1
Jump to H0 SARAM address 0x3F 8000	0	0	1	0
Jump to OTP address 0x3D 7800	0	0	0	1
Call Parallel_Boot to load from GPIO Port B	0	0	0	0

† PU = pin has an internal pullup No PU = pin does not have an internal pullup
‡ Extra care must be taken due to any affect toggling SPICLK to select a boot mode may have on external logic.
§ If the boot mode selected is Flash, H0, or OTP, then no external code is loaded by the bootloader.

For F280x devices, configure the GPIO according to the following boot mode table (excerpt from the TMS320F2808/2806/2801 datasheet, SPRS230):



MODE	DESCRIPTION	GPIO18 SPICLKA SCITXB	GPIO29 SCITXA	GPIO34
Boot to Flash	Jump to Flash address 0x3F 7FF6 You must have programmed a branch instruction here prior to reset to redirect code execution as desired.	1	1	1
SCI-A Boot	Load a data stream from SCI-A	1	1	0
SPI-A Boot	Load from an external serial SPI EEPROM on SPI-A	1	0	1
I2C Boot	Load data from an external EEPROM at address 0x50 on the I2C bus	1	0	0
eCAN-A Boot	Call CAN_Boot to load from eCAN-A mailbox 1.	0	1	1
Boot to M0 SARAM	Jump to M0 SARAM address 0x00 0000.	0	1	0
Boot to OTP	Jump to OTP address 0x3D 7800	0	0	1
Parallel I/O Boot	Load data from GPIO0 - GPIO15	0	0	0

NOTE: When configuring the bootmode, the SCITXDA must be pulled down through a resistor so that the F28xx can still drive the pin once the bootload process begins.

™ eZdsp is a trademark of Spectrum Digital Incorporated.

Step 3. **If programming the F2812, make sure that the XMP/MCn pin is pulled low to access the boot ROM.** On F2811, F2810, F2808, F2806, and F2801 devices, this signal is tied low internally such that the boot ROM is always active.

Step 4. **Disconnect any JTAG emulator that may be connected to the DSP board, such as a XDS510PP+ emulator.** An emulator connected to the JTAG port may hold the device in reset and thus the bootloader will not start when the device is reset.

Step 5. **Start SDFlash and load the appropriate project file.** This will be one of the following:

myprojects\sdf28xx_v3_0_serial\f2812\F2812SerialFlash.sdp

myprojects\sdf28xx_v3_0_serial\f2811\F2811SerialFlash.sdp

myprojects\sdf28xx_v3_0_serial\f2810\F2810SerialFlash.sdp

myprojects\sdf28xx_v3_0_serial\f2808\F2808SerialFlash.sdp

myprojects\sdf28xx_v3_0_serial\f2806\F2806SerialFlash.sdp

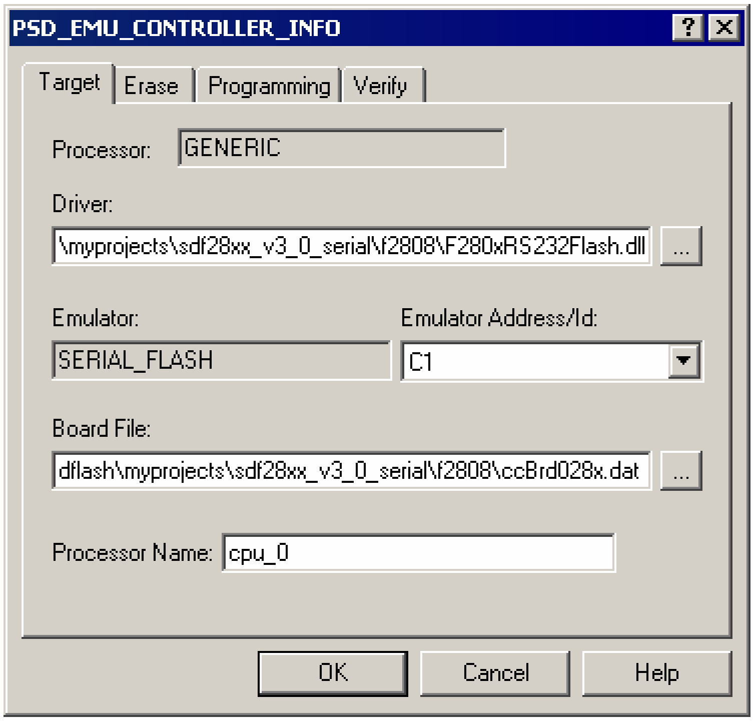
myprojects\sdf28xx_v3_0_serial\f2801\F2801SerialFlash.sdp

Step 6. **Modify the SDFlash project options (if necessary) to locate the various elements such as the device driver, algorithm file and flash data file.** The directory paths to these files assume the SDFlash is installed in the *C:\CCStudio_v3.1\specdig\sdflash* directory. To change any of the directory paths or project settings from their default values, open the project settings dialog box: *Project->Settings.* The project options that follow are specific to the SDFlash Serial Patch.

IMPORTANT

This is a very important step. If you did not install the SDFlash serial utility in the default directory of *C:\CCStudio_v3.1\specdig\sdflash*, then YOU MUST modify the file paths in the SDFlash project. Be sure that you check each tab in the project options. There are multiple places that the file paths are specified. If the wrong paths are specified, SDFlash serial may simply report a failure to connect to the DSP via the serial link, rather than telling you that your project option paths are incorrect.

Target Tab: (image shown for F2808 device)



Processor: GENERIC

Driver: \myprojects\sdf28xx_v3_0_serial\f2808\F280xRS232Flash.dll

Emulator: SERIAL_FLASH

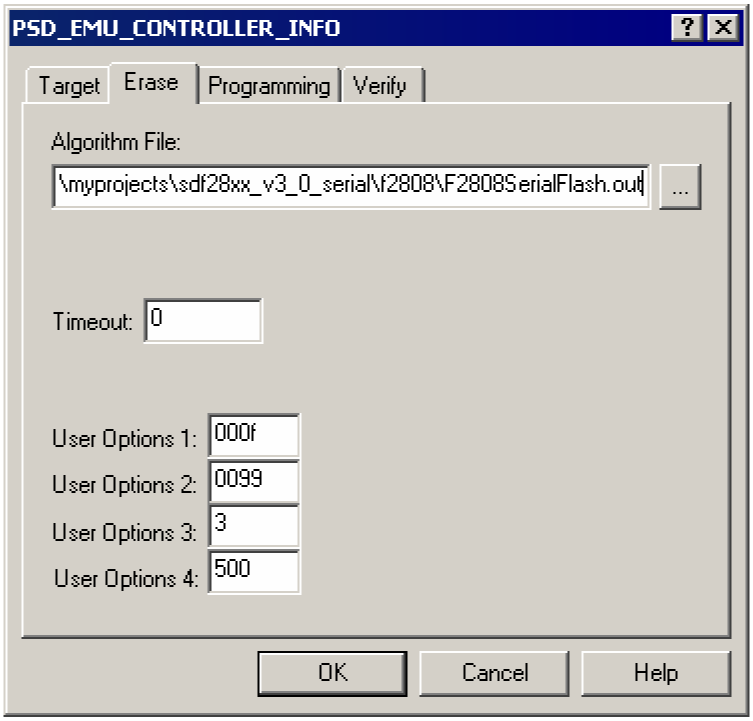
Emulator Address/Id: C1

Board File: dflash\myprojects\sdf28xx_v3_0_serial\f2808\ccBrd028x.dat

Processor Name: cpu_0

- **Driver:** This must point to either *F281xRS232Flash.dll* or *F280xRS232Flash.dll* (depending on which device you are using).
- **Emulator Address/ID:** This is the COM port on your PC that you are using to connect to the DSP. If COM1, COM2, COM3 and COM4 do not appear as options, modify the sdopts.cfg file as described previously in step 4.
- **Board file:** Leave as *ccBrd028x.dat*
- **Processor name:** Leave as *cpu_0*

Erase Tab: (image shown for F2808 device)



Algorithm File: \myprojects\sdf28xx_v3_0_serial\f2808\F2808SerialFlash.out

Timeout: 0

User Options 1: 000f
User Options 2: 0099
User Options 3: 3
User Options 4: 500

- ***Algorithm File:*** use the appropriate file for your device:

  myprojects\sdf28xx_v3_0_serial\f2801\F2801SerialFlash.out
  myprojects\sdf28xx_v3_0_serial\f2806\F2806SerialFlash.out
  myprojects\sdf28xx_v3_0_serial\f2808\F2808SerialFlash.out
  myprojects\sdf28xx_v3_0_serial\f2810\F2810SerialFlash.out
  myprojects\sdf28xx_v3_0_serial\f2811\F2811SerialFlash.out
  myprojects\sdf28xx_v3_0_serial\f2812\F2812SerialFlash.out

- ***Timeout:*** This can be left as 0. This value specifies the amount of time per flash sector (in 10's of milliseconds) that the host PC waits after sending the DSP an erase command before it checks for an acknowledge. This option has been superseded by Erase User Options #4. See the description for Erase User Options #4 below for more information.

- ***User Options 1****:* Sector Mask for erase. Default value is set to erase all sectors.
  Set Bit 0 = erase Sector A
  Set Bit 1 = erase Sector B
  ...

  For example:
    000f will erase all 4 sectors in F2808, F2806, or F2801 devices
    0001 will erase only sector A in F2808, F2806, or F2801 devices
    001f will erase all 5 sectors in the F2810 device
    03ff will erase all 10 sectors in F2812 and F2811 devices

- ***User Options 2:*** Toggle Test pin select. The toggle test toggles a selected pin at a known frequency in order to confirm proper configuration of the flash programming algorithms. While the test runs, monitor the selected pin using an oscilloscope. If the algorithms are configured correctly for your CPU rate then the pin will toggle near 10kHz (100µS +/- 10µS cycle time). If the pin is toggling at a different rate, then the algorithms are <u>not</u> configured correctly. The pin selection is made as follows:

  For F281x devices:
  Leave User Options 2 blank or 0000 for normal operation. To run the toggle test, set User Option 2 to a value of 0001 to 0005, per the below table to select the desired pin that will toggle. This will invoke the toggle test instead of the erase operation, and SDFlash will harmlessly timeout eventually. The user can click *Stop* to halt the toggle test sooner. Note that the DSP continues to run the Toggle test even after SDFlash has timed out.

Blank	Test not run
0000	Test not run
0001	GPIOF14_XF
0002	GPIOA0_PWM1
0003	Test not run
0004	GPIOG4_SCITXDB
0005	GPIOF12_MDXA
0006-FFFF	Test not run

  For F280x devices:
  Set User Options 2 to a number greater than 0022 for normal operation (e.g., default is 0099). To run the toggle test, set User Options 2 to a hexadecimal value equal to the GPIO pin number that you want to toggle. For example:

  0000 toggles GPIO0
  0001 toggles GPIO1
  ...
  0010 toggles GPIO16
  ...
  0022 toggles GPIO34

  Greater than 0022 toggle test does not run.

  This will invoke the toggle test instead of the erase operation, and SDFlash will harmlessly timeout eventually. The user can click *Stop* to halt the toggle test sooner. Note that the DSP continues to run the Toggle test even after SDFlash has timed out.

If the toggling does not occur at the expected frequency, changes need to be made to the SDFlash serial algorithm file, per the previous step 3.

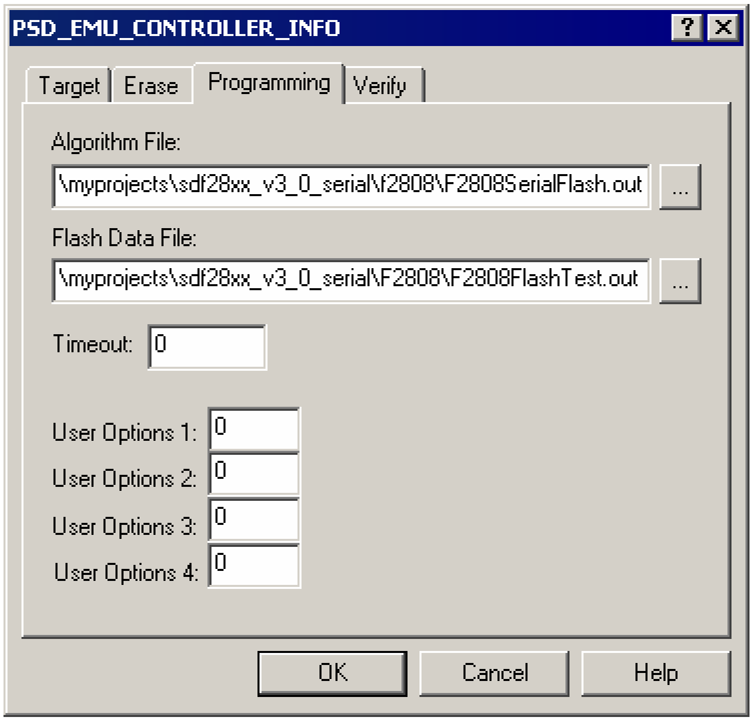
- **User Options 3:** Default Autobaud setting. Although this is on the Erase tab, it applies to all operations (toggle test, Erase, Program and Verify). Enter a number 0001 - 0004 that sets the default baud rate that SDFlash will attempt to autobaud lock with the DSP boot loader:

0001	57600 baud	
0002	38400 baud	
0003	19200 baud	(Default)
0004	9600 baud	
Other	38400 baud	

  Due to hardware, connectors, transceiver, PC UART, etc., you may need to lower the autobaud rate in order to successfully autobaud lock with the DSP.

- **User Options 4:** Erase acknowledge retry specifier. After sending the DSP an erase command, the PC runs a loop that checks for an erase acknowledge from the DSP. This option controls the maximum number of times the PC will loop. It is generally best to make this option large (e.g., 500 is the default value), which gives the DSP plenty of time to erase the flash. SDFlash will complete the erase operation as soon as the DSP reports that it is finished. In SDFlash Serial versions prior to v3.00, the user would specify a fixed amount of time for the PC to wait for the acknowledge via the Erase Timeout option. This had the disadvantage that SDFlash erase would always take the specified amount of time even if the DSP finished erasing sooner.

Programming Tab: (image shown for F2808 device)



Algorithm File: \myprojects\sdf28xx_v3_0_serial\f2808\F2808SerialFlash.out

Flash Data File: \myprojects\sdf28xx_v3_0_serial\F2808\F2808FlashTest.out

Timeout: 0

User Options 1: 0

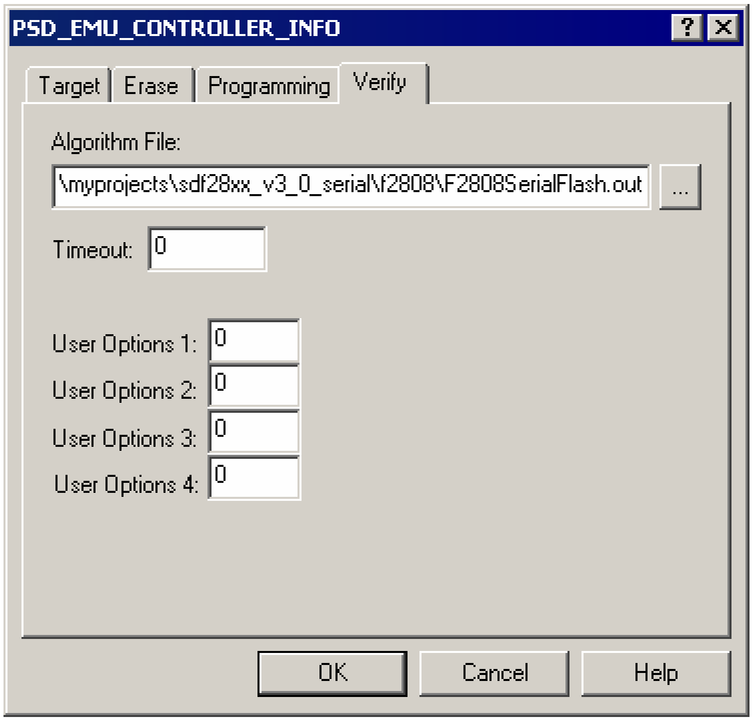
User Options 2: 0

User Options 3: 0

User Options 4: 0

- **Algorithm File***:* Use the same algorithm file that you specified on the Erase Tab.
- **Flash Data File:** This is the .out file that you want to program into the flash.
- **Timeout:** Leave as 0. This option has been superseded by Programming User Options 4.
- **User Options 1 through 3**: Unused.
- **User Options 4: :** Program acknowledge retry specifier. After sending the DSP a packet of data to program into the flash, the PC runs a loop that checks for a program acknowledge from the DSP. This option controls the maximum number of times the PC will loop. It can generally be left at the default value of 0.

Verify Tab:



Algorithm File: \myprojects\sdf28xx_v3_0_serial\f2808\F2808SerialFlash.out

Timeout: 0

User Options 1: 0

User Options 2: 0

User Options 3: 0

User Options 4: 0

- **Algorithm File:** Use the same algorithm file that you specified on the Erase Tab.

- **Timeout:** Leave as 0.

- **User Options 1 and 2:** Unused.
  Note: In the JTAG version of SDFlash, these options specify the wait-states for the OTP and Flash memory to be set before the verify operation starts. The serial version of SDFlash does not use these options however, and instead the wait-states remain set to their reset defaults (which are the maximum values).

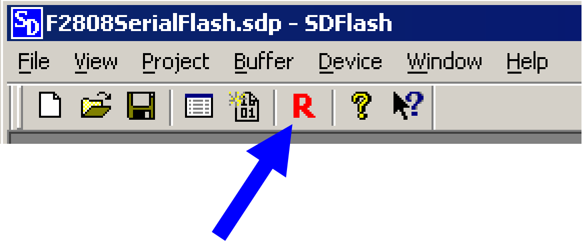
- **User Options 3 and 4:** Unused

Note that the Verify operation **does not** perform the word-by-word verify operation from the Flash API algorithm library. Rather, it performs a simple 16-bit checksum on the programmed data. Verify is done COFF section by COFF section after all sections have been programmed. A more complex verify is left to the user to implement, if desired.

Step 7. Save the SDFlash project file: Use *File->Save Project*. Alternately, you can change the project name at this point if desired by using *File->Save Project As.*

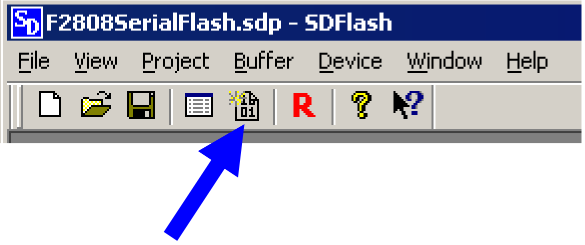
Step 8. Power up the DSP and/or reset it so that the bootloader will start. The Bootloader will wait to Autobaud lock with the host PC.

Step 9. Reset SDFlash: Click the large red R button in the SDFlash GUI, or you can use the menus: Device->Reset.

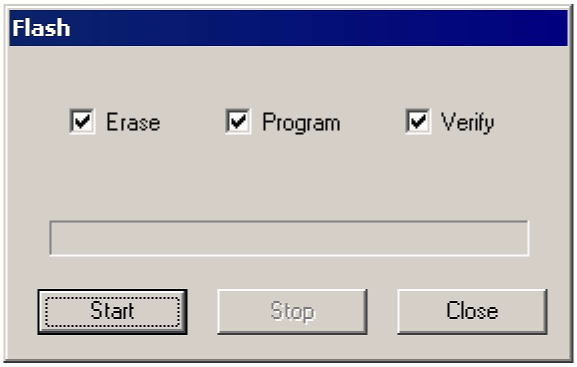

F2808SerialFlash.sdp - SDFlash
File View Project Buffer Device Window Help
R

NOTE: The reset button of SDFlash serial CANNOT reset the DSP target. If the target is in a bad state or not in bootloader mode then user must use a hardware reset to get target back to bootloader mode (per step 12). Once that is done the reset button is used to inform SDFlash that target is in bootloader mode.

Step 10. Erase/Program/Verify the flash. Click the FLASH button on the SDFlash GUI, or you can use the menus: Device -> Flash.


F2808SerialFlash.sdp - SDFlash
File View Project Buffer Device Window Help
R

Clicking the FLASH button will bring up the following pop-up window:



☑ Erase ☑ Program ☑ Verify

Select the desired operations and then click the “Start” button to execute them.

Rebuilding the SDFlash Serial Files

In general, the user will need to rebuild the SDFlash serial files only in the following three situations:

Case 1: You are using a crystal/oscillator for the DSP that does not match the assumed value. The assumed value is 30 MHz for F281x, and 20 MHz for F280x. If using a different value, the algorithm file *F28yySerialFlash.out* will need to be rebuilt to use a different PLL value.

Case 2: The code security passwords on the device are not all erased (0xFFFF). For F281x devices, the algorithm file *F28yySerialFlash.out* will need to be rebuilt to reflect the passwords in use. For F280x devices, the code security password match flow file *F28xxPasswordMF.out* will need to be rebuilt to reflect the passwords in use.

Case 3: You are using a DSP device or silicon revision that is not supported by this version of SDFlash Serial. The SDFlash Serial v3.0 release supports revision C and later of F2812, F2811, and F2810 devices, and all revisions of F2808, F2806, and F2801 devices. If using an unsupported silicon revision, the algorithm file *F28yySerialFlash.out* will need to be rebuilt with the appropriate flash programming algorithm APIs.

The Code Composer Studio project file and the source code for the SDFlash serial algorithm files are located in:

For F281x devices:

mydrivers\DSP281x_v3_0\DSP281x_serial\build\F28xxSerialFlash

mydrivers\DSP281x_v3_0\DSP281x_serial\src

For F280x devices:

mydrivers\DSP281x_v3_0\DSP281x_serial\build\F28xxSerialFlash

mydrivers\DSP281x_v3_0\DSP281x_serial\src

General instructions for rebuilding the files for the above 3 cases will now be provided.

Instructions for Case 1: You are using a crystal/oscillator for the DSP that does not match the assumed value.

1) Start Code Composer Studio.
2) Open the CCS project that corresponds to the device you are using, e.g. *F2808SerialFlash.pjt* for F2808, located in one of the following two sub-directories:

   mydrivers\DSP281x_v3_0\DSP281x_serial\build\F28xxSerialFlash

   mydrivers\DSP281x_v3_0\DSP281x_serial\build\F28xxSerialFlash
3) Edit the constant CPU_RATE in the corresponding file *Flash280x_Config_API.h* or *Flash281x_Config_API.h*, and also edit the setting for the PLLCR register in the function InitSysCtrl() in the file *f28xx_main.c* to reflect the reflect the crystal/oscillator in use. The relevant lines of code are:

   Flash28yx_Config_API.h:

   ```
   #define CPU_RATE   10.000L
   ```

   f28xx_main.c:

   ```
   SysCtrlRegs.PLLCR.bit.DIV = 0x000A;      // PLLx5
   ```
4) Rebuild the project. The output file, *F28yySerialFlash.out* (e.g., *F2808SerialFlash.out* for F2808) will appear in the relevant \bin directory:

   mydrivers\DSP281x_v3_0\DSP281x_serial\bin

   mydrivers\DSP281x_v3_0\DSP281x_serial\bin
5) Quit Code Composer Studio.
6) Copy *F28yySerialFlash.out* to overwrite the file of the same name in the SDFlash project directory you are using, e.g. one of:

   myprojects\sdf28xx_v3_0_serial\F2812\F2812SerialFlash.out

   myprojects\sdf28xx_v3_0_serial\F2811\F2811SerialFlash.out

   myprojects\sdf28xx_v3_0_serial\F2810\F2810SerialFlash.out

   myprojects\sdf28xx_v3_0_serial\F2808\F2808SerialFlash.out

   myprojects\sdf28xx_v3_0_serial\F2806\F2806SerialFlash.out

   myprojects\sdf28xx_v3_0_serial\F2801\F2801SerialFlash.out
7) Start and use SDFlash Serial as normal.

Instructions for Case 2: The code security passwords on the device are not all erased (0xFFFF). The instructions differ here for F281x and F280x devices.

For F281x devices:

1) Start Code Composer Studio.
2) Open the CCS project that corresponds to the device you are using, e.g. F2812SerialFlash.pjt for F2812, located in the following sub-directory:

   mydrivers\DSP281x_v3_0\DSP281x_serial\build\F28xxSerialFlash

3) Edit the file *Flash2812_API.c* so that the defined code security passwords reflect the passwords in use. All passwords are currently set to the value 0xFFFF. The relevant lines of code begin at roughly line 63 of the file:

```
// For erased flash the password locations will all be 0xFFFF
const  Uint16   PRG_key0  = 0xFFFF;     // PSWD bits 15-0
const  Uint16   PRG_key1  = 0xFFFF;     // PSWD bits 31-16
const  Uint16   PRG_key2  = 0xFFFF;     // PSWD bits 47-32
const  Uint16   PRG_key3  = 0xFFFF;     // PSWD bits 63-48
const  Uint16   PRG_key4  = 0xFFFF;     // PSWD bits 79-64
const  Uint16   PRG_key5  = 0xFFFF;     // PSWD bits 95-80
const  Uint16   PRG_key6  = 0xFFFF;     // PSWD bits 111-96
const  Uint16   PRG_key7  = 0xFFFF;     // PSWD bits 127-112
```

4) Rebuild the project. The output file, *F28yySerialFlash.out* (e.g., *F2812SerialFlash.out* for F2812) will appear in the \bin directory:

   mydrivers\DSP281x_v3_0\DSP281x_serial\bin

5) Quit Code Composer Studio.
6) Copy *F28yySerialFlash.out* to overwrite the file of the same name in the SDFlash project directory you are using, e.g. one of:

   myprojects\sdf28xx_v3_0_serial\F2812\F2812SerialFlash.out

   myprojects\sdf28xx_v3_0_serial\F2811\F2811SerialFlash.out

   myprojects\sdf28xx_v3_0_serial\F2810\F2810SerialFlash.out

7) Start and use SDFlash serial as normal.

For F280x devices:

1) Start Code Composer Studio.
2) Open the CCS project F28xxPasswordMF.pjt, located in the following directory:

   mydrivers\DSP281x_v3_0\DSP280x_serial\build\F28xxPasswordMF

3) Edit the file *passwords.inc* so that the defined code security passwords reflect the passwords in use. All passwords currently are set to the value 0xFFFF. The relevant lines of code are:

```
      .int      0xFFFF              ;PWL0 (LSW of 128-bit password)
      .int      0xFFFF              ;PWL1
      .int      0xFFFF              ;PWL2
      .int      0xFFFF              ;PWL3
      .int      0xFFFF              ;PWL4
      .int      0xFFFF              ;PWL5
      .int      0xFFFF              ;PWL6
      .int      0xFFFF              ;PWL7 (MSW of 128-bit password)
```

4) Rebuild the project. The output file, *F28xxPasswordMF.out* will appear in the \bin directory:

   mydrivers\DSP281x_v3_0\DSP280x_serial\bin

5) Quit Code Composer Studio.
6) Copy *F28xxPasswordMF.out* to overwrite the file of the same name in the SDFlash project directory you are using, e.g. one of:

   myprojects\sdf28xx_v3_0_serial\F2808\F28xxPasswordMF.out

   myprojects\sdf28xx_v3_0_serial\F2806\F28xxPasswordMF.out

   myprojects\sdf28xx_v3_0_serial\F2801\F28xxPasswordMF.out

7) Start and use SDFlash serial as normal.

Instructions for Case 3: You are using a DSP device or silicon revision that is not supported by this version of SDFlash Serial. For example, you may be using a new DSP. To adopt SDFlash Serial for the unsupported device, you will need to obtain from Texas Instruments the Flash Programming API algorithms for that device, assuming they currently exist for the device in question. In addition, these API algorithms must have the same function naming convention used by the API algorithms originally used to build this version of SDFlash serial. Assuming this the case, then:

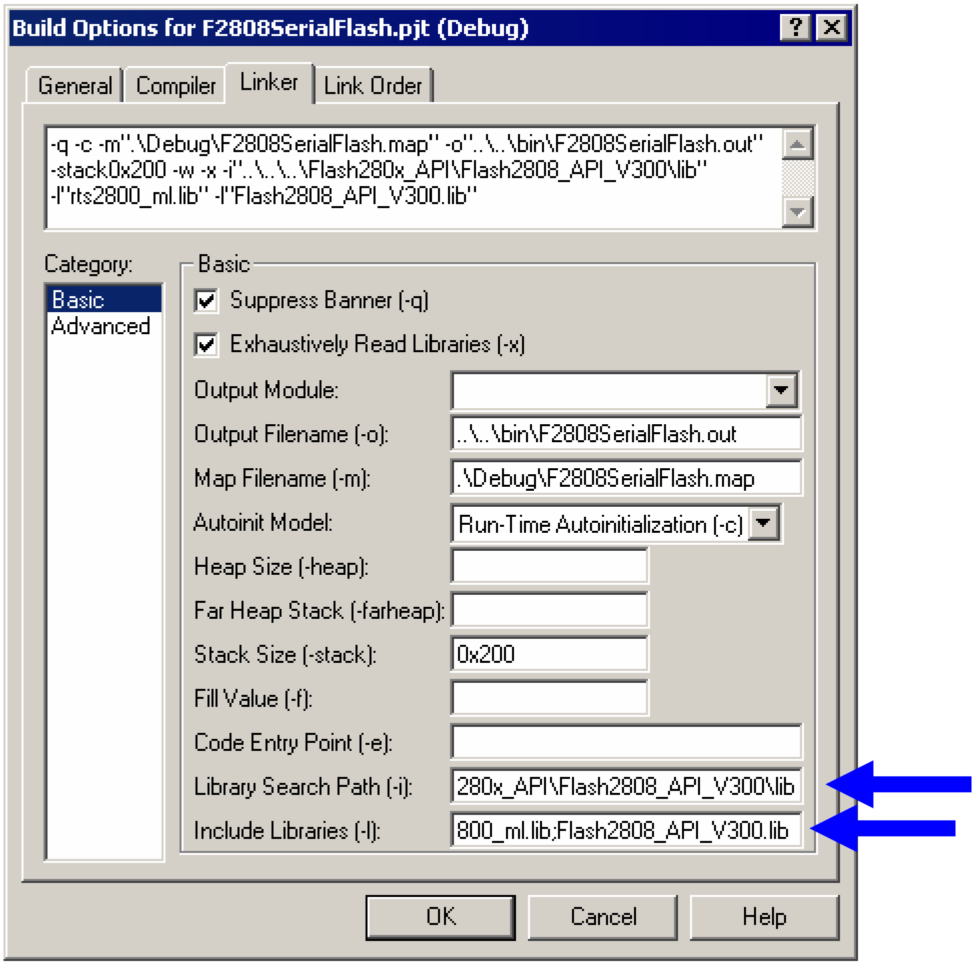
1) Start Code Composer Studio.

2) Open the CCS project that corresponds to the device you are using, e.g. *F2808SerialFlash.pjt* for F2808, located in one of the following two sub-directories:

   mydrivers\DSP281x_v3_0\DSP281x_serial\build\F28xxSerialFlash

   mydrivers\DSP281x_v3_0\DSP281x_serial\build\F28xxSerialFlash

3) In Code Composer Studio, click on the menu *Project->Build_Options*. Select the *Linker* tab on the pop-up window, and change the *Library_Search_Path* and the *Include_Libraries* option boxes to reflect the flash API algorithm library in use:



```
-q -c -m".\Debug\F2808SerialFlash.map" -o"..\..\bin\F2808SerialFlash.out"
-stack0x200 -w -x -i"..\..\..\Flash280x_API\Flash2808_API_V300\lib"
-l"rts2800_ml.lib" -l"Flash2808_API_V300.lib"
```

Basic

☑ Suppress Banner (-q)

☑ Exhaustively Read Libraries (-x)

Output Module:

Output Filename (-o): ..\..\bin\F2808SerialFlash.out

Map Filename (-m): .\Debug\F2808SerialFlash.map

Autoinit Model: Run-Time Autoinitialization (-c)

Heap Size (-heap):

Far Heap Stack (-farheap):

Stack Size (-stack): 0x200

Fill Value (-f):

Code Entry Point (-e):

Library Search Path (-i): 280x_API\Flash2808_API_V300\lib

Include Libraries (-l): 800_ml.lib;Flash2808_API_V300.lib

4) You may also need to modify needed include files or other files used by the flash API algorithms, e.g., *Flash28yy_API_Config.h*. See the documentation provided with the flash API algorithms for details on this step.

5) Rebuild the project. The output file, *F28yySerialFlash.out* (e.g., *F2808SerialFlash.out* for F2808) will appear in the relevant \bin directory:

   mydrivers\DSP281x_v3_0\DSP281x_serial\bin

   mydrivers\DSP281x_v3_0\DSP281x_serial\bin

6) Quit Code Composer Studio.

7) Copy *F28yySerialFlash.out* to overwrite the file of the same name in the SDFlash project directory you are using, e.g. one of:

   myprojects\sdf28xx_v3_0_serial\F2812\F2812SerialFlash.out

   myprojects\sdf28xx_v3_0_serial\F2811\F2811SerialFlash.out

   myprojects\sdf28xx_v3_0_serial\F2810\F2810SerialFlash.out

   myprojects\sdf28xx_v3_0_serial\F2808\F2808SerialFlash.out

   myprojects\sdf28xx_v3_0_serial\F2806\F2806SerialFlash.out

   myprojects\sdf28xx_v3_0_serial\F2801\F2801SerialFlash.out

8) Start and use SDFlash Serial as normal.

SDFlash Serial Source File Information

The following is a list of source files used in to make the SDFlash driver file (F28yxRS232Flash.dll) and the SDFlash serial flash algorithm file (F28yySerialFlash.out), along with a brief description of the files.

Directory *\mydrivers\DSP281x_v3_0\DSP280x_serial\src*: contains Win32 and F28xx shared source files:

drv2812sci.c, drv2812sci.h: Low level F28xx serial port driver.

f28xxpkt.c, f28xxpkt.h: Mid level packet code. Supports both Win32 and F28xx build.

lowlev.c, file.h: Optional cio level interface. Ported from TI rts28xx_ml library to support standard open, close, read, write file functions.

Flash2812_API.c and Flash2808_API.c: Thin wrapper between f28xx_main.c and TI FLASH API. Formats SDFlash arguments into those required by the TI FLASH API. Ported from TI FLASH API example.

f28xx_main.c, flash.h: Highest level interface for F28xx SDFlash/Serial interface.

Directory *\mydrivers\DSP281x_v3_0\DSP280x_serial\src_host*: contains Win32 source files to build the interface between the serial port and SDFlash:

f28xxsio.c, f28xxsio.h: Low level Win32 serial port interface. Uses standard Win32 communications interface.

cofflib.c, coffdefs.h, coff.h, cofflib.h: Spectrum Digital implementation of TI COFF reader.

f28xxprg.c, f28xxprg.h: The Spectrum Digital PRG interface files. Provides a generic interface for SDFlash to call. The interface is device independent and could be used to create any SDFlash based driver.

Troubleshooting Tips

If CONNECTING to the DSP via the serial link fails:

- Reset the DSP with a hardware reset (i.e., power-cycle the DSP if needed) to be sure that the bootloader was properly invoked.
- Check that the DSP is configured for SCI-A bootloader mode.
- Check that the serial port cable from the host (PC) is connected to SCI-A (as opposed to SCI-B).
- Check that the correct host (PC) communications port has been selected in the SDFlash project settings. In SDFlash, select Project->Settings, and then adjust Emulator Address/ID on the Target tab.
- Check that each file specified in the various SDFlash project settings is correct. Especially check the paths. Do this carefully, as the eyes can play tricks on you! Specifying a bad path for a file can result in SDFlash simply reporting a failure to connect.
- Try lowering the serial baud rate on in the SDFlash project settings. In SDFlash, select Project->Settings, and then adjust User_Options_3 on the Erase tab. Baud rate options are described earlier in this document.
- Make sure the SDFlash serial algorithms have been re-built to reflect the code security passwords that you are using. This was discussed earlier in this document. If you have already rebuilt the algorithms, or are not using the code security passwords (i.e., all passwords are 0xFFFF), confirm this using Code Composer Studio and an emulator to unlock the security and view the flash contents. If you cannot view the flash contents, then there is some sort of code security problem (e.g., the actual passwords are different than what is thought).
- Make sure the SDFlash serial algorithms have been re-built to reflect the DSP input clock you are using, and an appropriate PLL value. This was discussed earlier in this document.

If ERASE fails:

- Try increasing the value of the *Erase User Options #4* in the SDFlash project settings.

If Program fails:

- Try increasing the value of the *Program User Options #4* in the SDFlash project settings.

References

- **F281x Flash API**: Download SPRC125 from the TI website:

  http://focus.ti.com/docs/toolsw/folders/print/sprc125.html

- **F280x Flash API**: Download SPRC193 from the TI website:

  http://focus.ti.com/docs/toolsw/folders/print/sprc193.html

- **F281x Boot ROM and Bootloader info**: Download SPRU095 from the TI website:
  http://focus.ti.com/docs/prod/folders/print/tms320f2812.html#technicaldocuments

- **F280x Boot ROM and Bootloader info**: Download SPRU722 from the TI website:
  http://focus.ti.com/docs/prod/folders/print/tms320f2808.html#technicaldocuments