

TMS320F/C/R281x SPI Serial EEPROM Programming Utilities

SDFlash EEPROM Programming Algo Version 1.0

This download includes algorithm files that can be used with SDFlash to program a SPI Serial EEPROM connected to SPI-A of an F281x, C281x or R281x device.

**SDFlash is a product of Spectrum Digital Inc.
(www.spectrumdigital.com)**

**Document updated:
June 17, 2004**

Contents:

1.	Revision History	2
2.	Release Notes	2
3.	Known Limitations	2
4.	SDFlash Overview	3
5.	Quick Start Guide	5
6.	Preparing the Application .out File for Programming	9

1. Revision History

Version 1.0

Initial release.

2. Release Notes

- This release of the SDFlash 281x Serial EEPROM Programming Algo is based on utility functions written to write and read from a Serial EEPROM connected to SPI-A of a 281x device. The source code for the utility functions is provided as-is.
- SDFlash does not support the XDS560 scan controller.
- You should not run SDFlash and Code Composer at the same time. This results in two different applications trying to control the DSP. During programming, SDFlash will have complete control of the device. No user application code can be running in parallel.

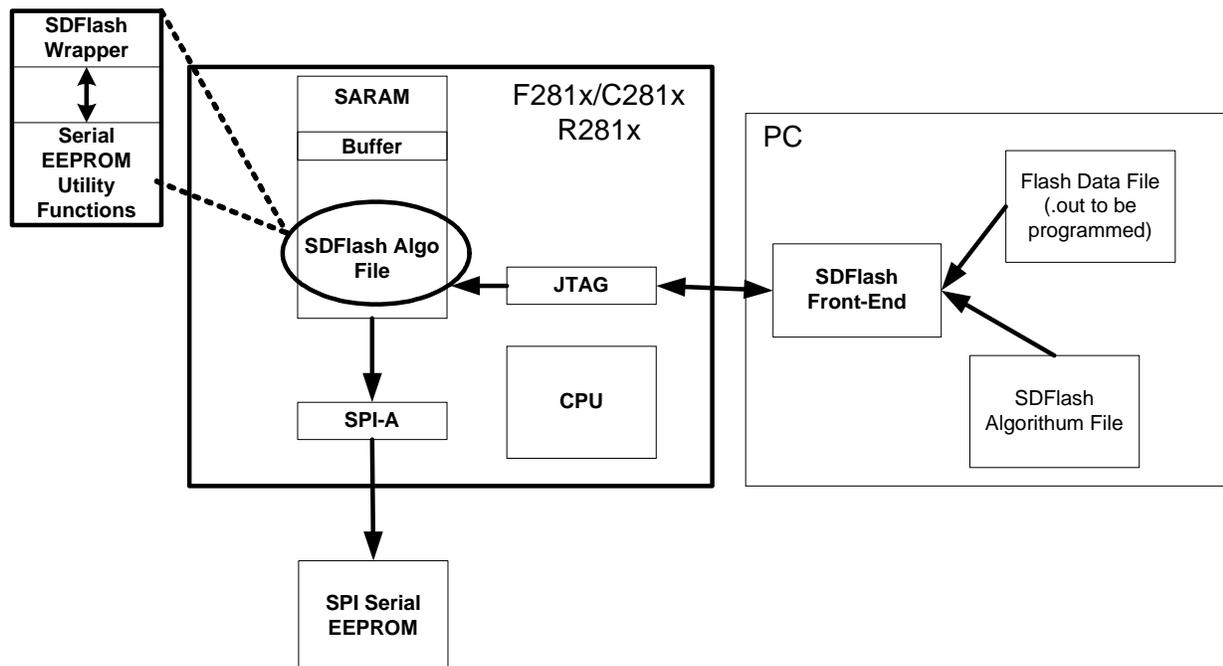
3. Known Limitations

- Support is currently provided for the Atmel SPI Serial EEPROMs (AT25HP256/512). The utility functions can be modified by the customer to support other EEPROMs if required.

4. SDFlash Overview

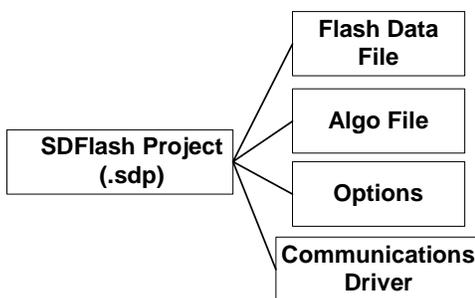
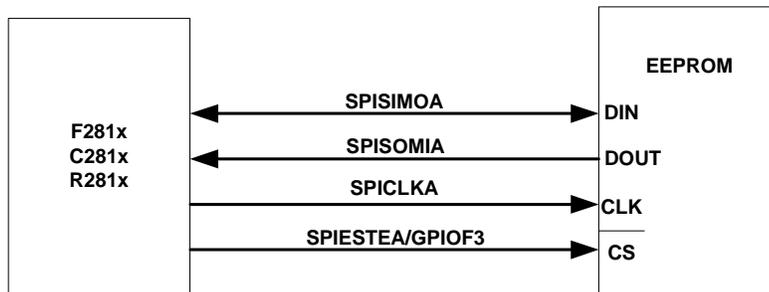
SDFlash is a generic front-end application owned by Spectrum Digital Inc (www.spectrumdigital.com). This application provides a generic interface to the JTAG communications channel that can be used to support programming of internal or external flash.

In order to perform the required operations, SDFlash downloads an algorithm file onto the DSP. This algorithm file is an executable (.out) file for the DSP being programmed and is executed on the DSP target. In this case, the SDFlash algorithm file consists of an SDFlash wrapper and 281x serial EEPROM programming utilities. The SDFlash wrapper has well defined standard functions and variables that are accessed by SDFlash over the JTAG channel. These functions in turn make calls to the EEPROM programming utility functions to perform operations on the serial EEPROM device. Note: because the SDFlash interface is separate from the algorithm file, the version of the SDFlash interface will differ from the version of the algorithm file.



The flash data file is first prepared to include the header information required by the on-chip SPI bootloader. With this information included, the “boot to SPI” option can then be used to copy the application into the 281x device at boot time. Detailed information on the SPI boot header and the 281x boot mode options can be found in *TMS320F28x Boot ROM Reference Guide* (spru095) available on TI’s website.

The R2812 eZdsp from Spectrum Digital Inc (www.spectrumdigital.com) comes with the required hardware setup. The EEPROM is connected to SPI-A and the boot mode pins are set to “boot to SPI” mode by default. The connection between the DSP and the EEPROM are shown below:



For SDFlash to program the EEPROM, it must have the following information:

- Location of the EEPROM algorithm file
- Location of the flash data file – that is the prepared data (.out file) to program into EEPROM device.
- Which JTAG driver to use
- Information about the JTAG scan chain

All of this information is stored in an SDFlash project file (.sdp) that can be edited through the SDFlash GUI interface. A sample SDFlash

project has been provided in this download.

Section 5 of this document will guide you through the SDFlash setup and show you how to use the provided sample SDFlash project to create your own project to program the Serial EEPROM.

5. Quick Start Guide

The following is a step-by-step guide for using the SDFlash utility to program your F281x device.

This quick start guide will refer to the following default directory locations:

<CCS base> default Code Composer Studio install directory: "c:\ti"
 <SDFlash base> default SDFlash directory <CCS base>\specdig\SDFlash

These directories may be different for your particular installation.

5.1. Prepare the .out (COFF) file of the application to be stored in the SPI Serial EEPROM.

The .out file of the application to be programmed must be re-formatted before it can be programmed into the Serial EEPROM device.

If you are using the example image provided you can skip this step and use the prepared image that is provided. When you are ready to prepare your own application, follow the detailed steps in section 6.

5.2. Make sure the target and emulator are setup properly:

- If you are using the R2812 onboard eZdsp USB emulator, run the R2812 EzDSP Diagnostics utility. This diagnostic is included as part of the R2812 eZdsp installation.
- If you are using an XDS510/XDS510pp+/SPI515 emulation controller, use the SDConfig utility. The R2812 on-board USB emulator is not supported by SDConfig. SDConfig is part of the default emulation installation and will be installed in your <CCS base>\specdig\sdconfig directory

5.3. Install the SDFlash flash support utility.

A version of SDFlash is included on the R2812 eZdsp CD. SDFlash will typically be installed in your <CCS base>\specdig\SDFlash directory.

SDFlash is a generic utility supplied by Spectrum Digital Inc. to interface to user written algorithms. In this case, Texas Instruments Inc has supplied the algorithm file. Users should check the SD website for updates to this utility.

5.4. Download the latest 281x SDFlash SPI EEPROM Programming algorithm files.

A version of the algorithms is included on the R2812 eZdsp CD. Future releases and upgrades will be available for download from the Spectrum Digital website in the *Support Utilities->SdFlashAlgo's* section.

5.5. Unzip the Serial EEPROM programming algorithm files into the myprojects subdirectory of SDFlash. For a typical install this will be the <CCS base>\specdig\SDFlash\myprojects directory.

This will automatically create a directory indicating the processor and version of the utilities.

Note: If you had installed an earlier release of SDFlash on your system you may have additional sub-directories to those shown, such as an algo directory. With the release of SDFlash V1.3 the algo directory was replaced with the myprojects directory. Presence of this directory will not effect the operation of SDFlash.



5.6. Make any required changes to the EEPROM Programming Algorithms.

The EEPROM programming algorithms configure the 281x device with the following configuration parameters:

```

PLLCR = 10 for SYSCLKOUT = 30*10/5 = 150Mhz
LOSPCP = 0x0001 for LSPCLK = 150Mhz/2 = 75Mhz
SPIBRR = 0x002B

```

These parameters result in an SPI clock rate of $75\text{MHz}/(0x002B+1) = 1.829\text{MHz}$

This configuration and algorithms have been tested with the R2812 eZdsp with an Atmel AT28HP256/512 serial EEPROM. If your system has other requirements you must modify the algorithms for your system.

To change the algorithms, load the SDFlash algo project into Code Composer Studio. This project is located at:

```
<SDFlash base>\myprojects\tif281x_eeprog_v1\SDFlash_Algo\SDFlash281x_Eeprog.pjt
```

To modify the LOSPCP and SPIBRR values, edit the file: SPIEEPROM28_Eeprog_Library.
To modify the PLLCR value, edit the file: SDFlash28x_Eeprog_Wrapper.h

After your modifications, re-build the algorithm file.

5.7. Make sure Code Composer Studio is not running and Run SDFlash.

5.8. Load the supplied SDFlash sample project. SDFlash uses project files to store information required to erase a device and program an .out file into a device. Sample 281x projects have been included in tif281x_eeprog_v1.zip for use as project templates.

Using *File->Open Project* in SDFlash, browse to and load the appropriate sample SDFlash project. For a typical installation, these files will be found in the following location:

```

R2812 eZdsp*:      <SDFlash base>\myprojects\tif281x_eeprog_v1\Sample_R2812USBeZdsp.sdp
281x 510PP+:      <SDFlash base>\myprojects\tif281x_eeprog_v1\Sample_281x_Eeprog.sdp

```

* This project uses the eZdsp locked on-board USB emulation driver.

5.9. Modify the SDFlash project (if required) to locate the various elements such as device driver, algorithm file and flash data file.

If you installed CCS and SDFlash in the <CCS base> and <SDFlash base> directories shown below, then usually only the *Flash Data File* on the **Program Tab**, and possibly the *Emulator Address/ID* on the **Target Tab** needs to be changed.

By default all flash projects are setup relative to the default TI CCS base directory "c:\ti". For example:

```

<CCS base>      default Code Composer install directory: "c:\ti"
<SDFlash base> default is <CCS base>\specdig\SDFlash
SDFlash binary default is <CCS base>\specdig\SDFlash
Flash projects  default is <CCS base>\specdig\SDFlash\myprojects\<projectname>

```

To change any of the directory paths or project settings from their default values, open the project settings dialog box: *Project->Settings*

Target Tab:

- Driver:* This is the Code Composer Studio™ emulation driver (*.dvr) file that is used to communicate with the target. The driver files can be found in the <CCS base>\drivers\ directory.
 default for the R2812 USB eZdsp: <CCS base>\drivers\sdgo2812eZdspusb.dvr
 default for other 510PP+/SPI515 etc: <CCS base>\drivers\sdgo28x.dvr
Emulator Address/ID:
 default for R2812 USB: 510
 default for 510PP+, etc: 378
- Board file:* File that provides SDFlash information on how many devices are on the JTAG scan chain. For a single 28x device on the scan chain, the default board file can be used. For systems with more devices on the scan chain, use the board file generated by Code Composer Studio to access your device. This file is found in the <CCS base>\cc\bin\BrdDat directory.
 The default board file is <SDFlash base>\myprojects\tif281x_eeprog_v1\ccBrd028x.dat
- Processor name:* default is cpu_0

Erase Tab:

- Algorithm File:*
 <SDFlash base>\myprojects\tif281x_eeprog_v1\SDFlash_Algo\Debug\ SDFlash28x_eeprog.out
- Timeout:* leave as 200 or higher
- For all other boxes the default is blank.*

Program Tab:

- Algorithm File:*
 <SDFlash base>\myprojects\tif281x_eeprog_v1\SDFlash_Algo\Debug\ SDFlash28x_eeprog.out
Flash Data File: This is the prepared .out file that you want to program into the SPI serial EEPROM. The steps for preparing the source application file to the correct format are described in section 6. For the example image, the prepared file can be found at:
 <SDFlash base>\myprojects\tif281x_eeprog_v1\test_image\Debug\spi_rom_281x.out
 The example application toggles the XF LED while checking an initialized memory pattern. The memory is initialized during boot time. If anything is found incorrect in the initialized memory the LED will stop. On the R2812 eZdsp the XF line is attached to LED DS2.
- Timeout:* leave as 200 or higher
- For all other boxes the default is blank.*

Verify Tab:

- Algorithm File:*
 <SDFlash base>\myprojects\tif281x_eeprog_v1\SDFlash_Algo\Debug\ SDFlash28x_eeprog.out
- Timeout:* leave as 200 or higher
- For all other boxes the default is blank.*

- 5.10. **Save the SDFlash project file: *File->Save Project As.*** Once you have made the required changes select ok and save the project using the name of your choice: *File->Save Project As.*

If you changed the *Emulator Address/ID* setting on the **Target Tab**, you should get a message that the current driver was unloaded and a new driver has been loaded. This operation is required to synchronize the SDFlash project settings with SDConfig. If you did not change the *Emulator Address/ID*, then you will not get this message.

Should you want to program a different .out file into the EEPROM, use this project as a template and change the *Flash Data File* on the **Program Tab**.

- 5.11. **Reset the device: *Device->Reset.*** You will get a pass/fail message in the output window.

- 5.12. **Erase/Program/Verify your device: *Device->Flash*.** Check or un-check the operation(s) you want to perform then select start. Each checked operation is executed from left to right, with continue on success and abort on fail.

Note that the erase function has been included only for completeness. For an EEPROM it is not required to erase the device before programming and thus this step is not required.

- 5.13. **Optional: Repeat erase/programming for each device to be programmed.** If additional devices are to be programmed, the target can be powered down and a new target connected without closing the SDFlash utility. Once the new target is connected, reset the part (*Device->Reset*) and erase, program, verify (*Device->Flash*) the device as described in 5.11 and 5.12.

- 5.14. **Boot the code from SPI-A**

Power cycle or hardware reset the device. The boot mode options should be set to “boot to SPI-A” which is the default on the R2812 eZdsp. The code from the EEPROM will be loaded into the device. If you are using the example image, you should see the XF LED (DS2) blink on the eZdsp. Note, if you reset the device from the debugger, the PLLCR register is not reset. This can cause a baud rate issue due to SYSCLKOUT being higher than expected. For power-on or a hardware reset PLLCR is cleared and this is not an issue.

6. Preparing the Application .out File for Programming

There are a few basic steps that need to be completed before programming the application into the Serial EEPROM device. These two steps are described in detail in this section.

6.1. Locate and modify the included `fmt_eZ281x.bat` batch (.bat) file.

This batch file will be used to help you automate the file preparation process. This .bat file can be found in the following location:

<SDFlash base>myprojects\tif281x_eeprog_v1\make_image\fmt_eZ281x.bat

and can be used as a template and modified to suite your preferences. This file was tested on a Windows XP system.

6.1.1. Specify the location of files that will be used.

Create a copy of `fmt_eZ281x.bat` and edit the top of this of the file to specify the directory path to the required utilities, name, and directory location of the .out file that you want to prepare. As supplied the file is setup for a default installation and will prepare the sample image.

```
REM Code Composer Studio install location. This is used to access the hex2000 utility.
SET CCS_DIR=c:\ti
```

```
REM Specify the directory where the image will be made.
SET MAKE_IMAGE_DIR=c:\ti\specdig\sdf\flash\myprojects\tif281x_eeprog_v1\test_image\debug
```

```
REM Specify the name of the file to prepare
SET SPI_COFF=image.out
```

```
REM Specify location of Asciihexlib.exe
SET ASCIIHEX_DIR=c:\ti\specdig\sdf\flash\bin
```

6.1.2. Specify the SPICLK rate that will be used during the boot load process.

To do this, specify the initial value for the LOSPCP register and the SPIBRR register in your copy of the batch file. These values will be included in the SPI boot header. These values will be used to initialize the LOSPCP register and SPIBRR register by the SPI boot loader.

```
REM Specify the LOSPCP register value for the SPI bootlaoder
SET LOSPCP_REG=0x00
```

```
REM Specify the SPIBRR register value for the SPI bootloader
SET SPIBRR_REG=0x07
```

Caution:

The user should select the SPICLK rate based on their design specific EEPROM requirements. The F/C/R281x SPI boot loader has an option to change the SPICLK rate based on information included in the SPI boot header. In this example the LOSPCP is set to 0x0 and SPIBRR is set to 0x7. For this example, the SPICLK rates are adjusted for an ATMEL AT25HP256/512 EEPROM for 1.875MHz.

Assuming the PLL is not disabled, the R2812 EzDSP default clock rate at reset is $30\text{MHz}/2 = 15\text{MHz}$. LSPCLK is set to times 1 (`lospcp = 0x0`) and the SPI baud rate register is set to 7 (`spibrr = 0x7`). Hence SPI clock rate is adjusted to $15\text{MHz}/(7+1) = 1.875\text{MHz}$ after by the SPI boot loader.

6.2. Invoke the batch file.

When invoked, the batch file will process the specified out file in two steps. During the 2nd step you will be prompted to make a simple edit to one of the files. This section describes the steps and the modification you will be asked to make.

Batch File Step 1: Run hex2000 to add the boot loader header information to each section.

The boot header information is required for the 281x SPI boot loader to properly download the application into the device for execution. This header information is added to the file by converting the .out file to ASCII-HEX using the C2000 hex utility that is included with the codegen tools in Code Composer Studio. The example below shows how the file *sourcefile.out* would be converted to the ASCII-HEX file *spi_rom_out.asc*:

```
hex2000 -a -boot -spi8 -lospcp %LOSPCP_REG -spibr SPIBRR_REG -o
spi_rom_281x.asc spi_source.out
```

where:

-a	Create an ASCII-Hex file.
-boot	Use boot header format.
-spi8	Use the SPI 8-bit boot header format
-	Specify the initial value for the LOSPCP register. This value should be selected based on the
lospcp	system's specific requirements the EEPROM device being used.
-spibr	Specify the initial value for the SPIBRR register. This value should be selected based on the
	system's specific requirements the EEPROM device being used.
-e	Specify the code entry point to begin execution after boot loading.

Refer to *TMS320F28x Boot ROM Reference Guide* (spru095) for more information on the SPI-boot header requirements and hex2000 options.

Note that while the code entry point for the application can be passed to the hex2000 utility by using the `-e` option, it is normally included as part of the .out file built by Code Composer Studio. For example, by default for a C program the entry point defined in the .out file is the C-initialization routine `_c_int00`. If required, during compile time the linker's `-e` option can be used to manually specify the application's entry point. The `-e` option allows you to manually specify a symbol that corresponds to the start of the section which should be the entry point. This information is then included in the SPI-boot header and used by the boot loader to transfer control to the application that has been downloaded into the device through the SPI module.

Batch File Step 2: Create a single-section COFF file to program into the EEPROM.

The ASCII-HEX file is then converted to a single section COFF file that will be sequentially programmed into the Serial EEPROM device.

The batch file will open the ASCII-Hex file created by the hex2000 utility in the previous step in notepad. You will be prompted to edit the ASCII-HEX file to include a base address required by the `Asciihexlib.exe` utility from Spectrum Digital.

Before you edit the file, the top of the .asc file will look similar to the following:

```
□
AA 08 02 7E 00 00 00 00 00 00 00 00 00 00 00 00 00 3F 00 00 90 00 04
00 00 00 00 00 00 01 00 02 00 03 00 04 00 05 00 06 00 07 00 08 00 09 00
. . .
```

To add the base address, after the first non-printable character, add the following:

```
<enter>$A0000,
```

Where <enter> is the enter key on your keyboard. The file will now look like:

```
□  
$A0000,  
AA 08 02 7E 00 00 00 00 00 00 00 00 00 00 00 00 00 00 3F 00 00 90 00 04  
00 00 00 00 00 00 01 00 02 00 03 00 04 00 05 00 06 00 07 00 08 00 09 00  
. . .
```

Be sure to include all 4 zeros and the comma. There is no return after the comma.

Save the file once you are finished editing it.

- 6.2.1. Use the Spectrum Digital Asciihexlib utility to convert the ASCII-HEX file to a single-section COFF file. The Asciihexlib file is included in the This is done as follows:

```
Asciihexlib.exe -hex spi_rom_281x.asc -out spi_rom_281x.out
```