

TMS320LF240xA Flash Programming Utilities

October 2005 Update

**This download includes the algorithm files used to
program the TMS320LF2407A, TMS320LF2406A,
TMS320LF2403A, TMS320LF2402A and TMS320LF2401A
devices with the SDFlash JTAG Flash utility.**

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

Flash Programming Utilities Disclaimer

Texas Instruments Inc. (TI) reserves the right to update or change any material included with this release. This includes:

- ☐ **The Flash Utility Kernels based on continued TMS320LF240xA testing.**
- ☐ **Improvements in algorithm performance and functionality.**

It is the users responsibility to check for future updates to the Flash Programming Utilities and to use the latest version available for their LF240xA silicon.

Updates to the utilities, including the kernels, will be posted on the Texas Instruments Inc. website (www.ti.com) and/or the Spectrum Digital website (www.spectrumdigital.com).

This update is targeted for the LF240xA devices with Code Security. Although the algorithms should work on the older LF240x (non-A) devices without Code Security, they have not been validated, and hence are not guaranteed on those platforms. Since the LF240x (non-A) devices do not have Code Security, any section in this document on unlocking the device is not a concern for those devices.

Contents:

1. Reasons for Update.....	4
2. Description of Changes	4
3. SDFlash and CCS Plugin versus Prg2xx.....	4
4. SDFlash Quick Start Guide.....	4
5. CPU Clock Rate and Timings Configuration.....	12
6. Code Security Module (CSM) Password Considerations	15
7. Error Messages and Troubleshooting Tips.....	17
7.1 Troubleshooting: All Operations	17
7.2 Troubleshooting: Erase Fails.....	18
7.3 Troubleshooting: Programming Fails.....	18
7.4 Troubleshooting: Verify Fails	18
7.5 Troubleshooting: Programmed Application Fails To Run	18
7.6 Error Messages	19

This document is an update to the TMS320LF240xA Flash programming utilities readme files.

- ❑ The latest Prg2xx algorithm is the Aug 8, 2003 release.
- ❑ The July 2004 and later flash algorithms will not run on the Prg2xx tool
- ❑ The July 2004 and later flash algorithms require SDFlash™ version 1.62.03 or higher. SDFlash can be downloaded from Spectrum Digital's website at www.spectrumdigital.com.
- ❑ The October 2005 update does not affect the Code Composer Studio Plugin.

1. Reasons for Update

The October 2005 update is **not** a mandatory update. The algorithms did not change at the base flash level.

The July 2004 update introduced a CRC check to improve the error capabilities of the programming utilities. Prior to that revision, to change the frequency and PLL settings of the algorithms modifications were needed on only one file, VAR.h, before recompiling. With the introduction of the CRC check, modifying this file broke the algorithms. If a PLL ratio other than x4, or a SYSCLK frequency other than 40Mhz was desired, the procedure to correct the CRC key was undocumented. The October 2005 update removes the requirement to correct the CRC algorithm for the PLL ratio change and documents the steps required to correct the CRC algorithm for the SYSCLK frequency change.

2. Description of Changes

The PLL clock ratio setting was removed from the CRC checking routine. This way the PLL_RATIO_CONST setting can be modified in the VAR.h file without any special changes to the CRC key. If the change to PLL_RATIO_CONST results in a different system clock frequency other than 40Mhz, then the CRC key will need to be modified. The procedure required to change the frequency of operation of the algorithms is documented in the VAR.h file as well as later in this document.

3. SDFlash and CCS Plugin versus Prg2xx

This update is not supported for Prg2xx. The Prg2xx utility is not stable on the latest Windows operating systems, while SDFlash and CCS are supported for Windows XP. SDFlash has a "command line feel" in that the utility window can be left open while cycling power and changing devices. User need only to push the start button after the device power is turned on, similar to executing a command line batch file.

The CCS Plugin works seamlessly with the Code Composer Studio™ development software. This update does not affect the Code Composer Studio Plugin.

4. SDFlash Quick Start Guide

4.1 Run the SDConfig utility to make sure the target and emulator are setup properly.

The latest version of the SDConfig utility is included in Spectrum Digital's Code Composer emulation driver install package for C2000. This installation can be downloaded from the Spectrum Digital (www.spectrumdigital.com) website in the *CodeComposerDrivers->C2000* download section if it is not already installed on your system.

SDConfig will typically be installed in your <CCS base>\specdig\sdconfig directory.

4.2 Install the SDFlash flash support utility.

The latest version of the SDFlash utility is included in Spectrum Digital's Code Composer emulation driver install package for C2000. This installation can be downloaded from the Spectrum Digital (www.spectrumdigital.com) website in the *CodeComposerDrivers->C2000* download section if it is not already installed on your system.

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 flash algorithms. In this case, Texas Instruments Inc has supplied the algorithm file. Users should check the SD website for updates to this utility.

4.3 Download the latest LF240xA SDFlash algorithm files.

The latest LF240xA SDFlash algorithm .zip file can be downloaded from the Spectrum Digital website in the *Downloads/Drivers->C2000 Products->Utilities->SDFlash Install/Download* section. For LF2406A, LF2403A, and LF2402A devices use the LF2407A algorithms. In addition, check Spectrum Digital's website for future updates to the algorithms.

Note: The July 2004 release and later of the LF240xA algorithms require version 1.62.03 or later of SDFlash.

- 4.4 **Unzip the LF240xA SDFlash 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.



- 4.5 **Run SDFlash.** It is recommended that the CLKOUT pin is monitored prior to executing the algorithm functions. If CLKOUT is not at the expected frequency/duty cycle, turn power off, reseal the part (if possible), and turn power back on. The expected frequency of CLKOUT at power up is CLKIN/2. CLKOUT is not turned on by default on the LF2401A device.
- 4.6 **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 LF2407A and LF2401A projects have been included in the algorithm .zip file 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:

LF2407A:	<SDFlash base>\myprojects\lf2407a_Oct-2005\LF2407A.sdp
LF2406A:	<SDFlash base>\myprojects\lf2407a_Oct-2005\LF2406A.sdp
LF2403A:	<SDFlash base>\myprojects\lf2407a_Oct-2005\LF2403A.sdp
LF2402A:	<SDFlash base>\myprojects\lf2407a_Oct-2005\LF2402A.sdp
LF2401A:	<SDFlash base>\myprojects\lf2401a_Oct-2005\LF2401A.sdp

Note: The LF2406A, LF2403A, and LF2402A device project files are found in the LF2407A directory.

- 4.7 **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> directory 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 to 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 <SDFlash base>\bin
Flash projects	default is <SDFlash base>\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:

PSD_EMU_CONTROLLER_INFO

Target | Erase | Programming | Verify

Processor: C2xx

Driver: C:\ti\drivers\sdgo24x.dvr

Emulator: XDS510PP_PLUS Emulator Address/Id: 378

Board File: C:\ti\cc\bin\BrdDat\ccBrd0.dat

Processor Name: cpu_0

OK Cancel Help

- ❑ **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. The XDS510PP_PLUS default driver for the LF240xA devices is sdgo24x.dvr. The USB default driver for the LF240xA devices is sdgo24xusb.dvr.
- ❑ **Emulator Address/ID:** default for XDS510PP_PLUS is 378. For the USB emulator this should be set to 510. This address must match the setting in your SDConfig setup.
- ❑ **Board File:** File that provides SDFlash information on how many devices are on the JTAG scan chain. For a single 240x device on the scan chain, the default board file can be used. For systems with more device 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 ccBrd0.dat.
- ❑ **Processor name:** default is cpu_0.

Erase Tab:

PSD_EMU_CONTROLLER_INFO

Target Erase Programming Verify

Algorithm File:
 sh\myprojects\lf2407a_Oct-2005\Clear_Erase\C2xx_bcex.out ...

Timeout: 200

ST0:

ST1: 07FC

User Options 1: 000F

User Options 2:

User Options 3:

User Options 4:

PMST:

PMST Address:

OK Cancel Help

NOTE: The Clear and Erase algorithms are performed sequentially during the "Erase" function. Both algorithms are contained together in one algorithm file.

- ☐ **Algorithm File:**
 - LF2407A: <SDFlash base>\myprojects\lf2407a_Oct-2005\Clear_Erase\C2xx_bcex.out
 - LF2406A: <SDFlash base>\myprojects\lf2407a_Oct-2005\Clear_Erase\C2xx_bcex.out
 - LF2403A: <SDFlash base>\myprojects\lf2407a_Oct-2005\Clear_Erase\C2xx_bcex.out
 - LF2402A: <SDFlash base>\myprojects\lf2407a_Oct-2005\Clear_Erase\C2xx_bcex.out
 - LF2401A: <SDFlash base>\myprojects\lf2401a_Oct-2005\Clear_Erase\C2xx_bcex.out
- ☐ **Timeout:** leave as 200 or higher
- ☐ **User Options 1:** Sector Mask. Bit 0 is for sector 0 up to bit 3 is for sector 3. A one is enabled. Default for LF2407A and LF2406A is 000F (without the hexadecimal indicator). Default for LF2403A, LF2402A, and LF2401A, which have only two sectors, is 0003.
- ☐ **ST1:** Default is 07FC to clear the CNF bit before loading the algorithm.
- ☐ **For all other boxes the default is blank.**

Programming Tab:

PSD_EMU_CONTROLLER_INFO

Target Erase Programming Verify

Algorithm File:
dfldash\myprojects\lf2407a_Oct-2005\Prog_Ver\C2xx_bpx.out ...

Flash Data File:
C:\ti\specdig\sdfldash\myprojects\lf2407a_Oct-2005\L32kNo ...

Timeout: 6 ST0: ST1: 07FC

User Options 1: PMST: PMST Address:

User Options 2:

User Options 3:

User Options 4:

OK Cancel Help

- ❑ **Algorithm File:**
 LF2407A: <SDFlash base>\myprojects\lf2407a_Oct-2005\Prog_Ver\C2xx_bpx.out
 LF2406A: <SDFlash base>\myprojects\lf2407a_Oct-2005\Prog_Ver\C2xx_bpx.out
 LF2403A: <SDFlash base>\myprojects\lf2407a_Oct-2005\ Prog_Ver\C2xx_bpx.out
 LF2402A: <SDFlash base>\myprojects\lf2407a_Oct-2005\ Prog_Ver\C2xx_bpx.out
 LF2401A: <SDFlash base>\myprojects\lf2401a_Oct-2005\ Prog_Ver\C2xx_bpx.out
- ❑ **Flash Data File:** This is the .out file that you want to program into the flash. The sample image provided with the algorithm is a simple data = address pattern. The code security module password locations are programmed to zeros by the sample image so that the CSM can easily be unlocked. This is one of two default passwords that do not require a matching key (the other is erased, or all 0xFFFF's). If you have your own .out file ready to be programmed you can specify that file as the data file. The sample image Flash Data File is located at:
 LF2407A: <SDFlash base>\myprojects\lf2407a_Oct-2005\L32kNoPw.out
 LF2406A: <SDFlash base>\myprojects\lf2407a_Oct-2005\L32kNoPw.out
 LF2403A: <SDFlash base>\myprojects\lf2407a_Oct-2005\L16kNoPw.out
 LF2402A: <SDFlash base>\myprojects\lf2407a_Oct-2005\L8kNoPw.out
 LF2401A: <SDFlash base>\myprojects\lf2401a_Oct-2005\L8kNoPw.out
- ❑ **Timeout:** leave as 6 or higher.
- ❑ **ST1:** Default is 07FC to clear the CNF bit before loading the algorithm.
- ❑ **For all other boxes the default is blank.**

Verify Tab:

PSD_EMU_CONTROLLER_INFO

Target Erase Programming **Verify**

Algorithm File:
 dfldash\myprojects\lf2407a_Oct-2005\Prog_Ver\C2xx_bpx.out ...

Timeout: 3000

User Options 1: ST0:

User Options 2: ST1: 07FC

User Options 3: PMST:

User Options 4: PMST Address:

OK Cancel Help

- ☐ **Algorithm File:**
 LF2407A: <SDFlash base>\myprojects\lf2407a_Oct-2005\Prog_Ver\C2xx_bpx.out
 LF2406A: <SDFlash base>\myprojects\lf2407a_Oct-2005\Prog_Ver\C2xx_bpx.out
 LF2403A: <SDFlash base>\myprojects\lf2407a_Oct-2005\ Prog_Ver\C2xx_bpx.out
 LF2402A: <SDFlash base>\myprojects\lf2407a_Oct-2005\ Prog_Ver\C2xx_bpx.out
 LF2401A: <SDFlash base>\myprojects\lf2401a_Oct-2005\ Prog_Ver\C2xx_bpx.out
- ☐ **Timeout:** leave as 3000.
- ☐ **User Options 1:** Default is blank. If the field is left blank, a CRC verify to the .out file specified in the *Program Tab->Flash Data File* field will occur. If the field is a '1', a verify to all 0xFFFF's will occur. If the field is a '2', a verify to all 0x0000's will occur. Note: SDFlash will not leave the device in a cleared state (all 0x0000's). Clear and erase occur within the same algorithm step.
- ☐ **ST1:** Default is 07FC to clear the CNF bit before loading the algorithm.
- ☐ *For all other boxes the default is blank.*

- 4.8 **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.

You have now created an SDFlash project that can be used anytime you want to erase or program the device using these settings. Should you want to program a different .out file into the LF240xA flash, use this project as a template and change the *Flash Data File* on the **Programming Tab**.

4.9 Configure the algorithm for the required CPU frequency and Flash timings.

For a custom CPU frequency, you must follow the instructions in Section 5 to properly configure the algorithms before continuing. As supplied, the algorithm is configured for x4 PLL mode (40Mhz with a 10Mhz input clock).

CAUTION

The erase and program operation **MUST** be configured for the CPU clock rate (CLKOUT) at which they will run. This configuration is VITAL for proper operation of the algorithm.

As supplied, the algorithm is configured for x4 PLL mode (40Mhz with a 10Mhz input clock).

For customer CPU frequency and PLL multiplier, you must follow the instructions in Section 5 to properly configure the algorithms before continuing.

4.10 Optional: View which sections are going to be programmed (i.e. loaded) using the *View->Coff/Hex file stats* SDFlash function.

- ☐ Make sure that no RAM locations are marked as load sections.
- ☐ Constant sections (i.e. .const/.econst) must be linked to page 0 (i.e. program memory) for SDFlash to program them.
- ☐ It is suggested to not program the CSM password locations (0x0040 – 0x0043) during development since flash contents will be changed often.

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

4.12 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. Refer to Section 7 *Error Messages and Troubleshooting Tips* should a failure occur.

CAUTION

Do not press the SDFlash STOP button during the erase or program operation.

Pressing STOP will halt the CPU before the Erase algorithm and/or Program algorithm completes. This can leave the Flash in a depleted state or result in unknown Code Security Module passwords and lock the device permanently.

Other conditions that can cause the CPU to halt prior to the completion of the Erase algorithm (e.g., power loss, device reset, PC crash, etc.) can result in the same problem described above.

4.13 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 steps 4.11 and 4.12.

- 4.14 **Optional: View the flash contents using Code Composer Studio™ (CCS).** You can view the programmed flash using CCS, and compare with your source code.

Start CCS and open a memory window to view the flash contents (or use the disassembly window). In addition you can load the CCS project and load the symbols from the .out file.

Make sure that you do not press Start on SDFlash while CCS is open. Otherwise a port contention will occur between CCS and SDFlash.

5. CPU Clock Rate and Timings Configuration

CAUTION

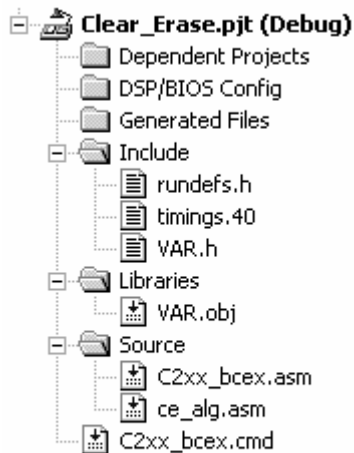
The LF240xA Flash kernel used by SDFlash contains several delay parameters that are implemented as software delays. Timing of these delays is VITAL to proper operation. To ensure the proper delays, the flash algorithms must be run at the correct speed.

As shipped, the algorithms are configured for a 10 MHz input clock and 40 MHz CPU clock (CLKOUT). To generate the 40 MHz CPU clock, the algorithms initialize the clock prescale bits (CLKPS) in the SCSR register to zero, which gives PLL x4 mode. If your hardware is running at a different CPU rate and/or requires a different PLLCR setting, then you must configure the flash programming algorithms as described below:

The following steps describe how to compile a new SDFlash algorithm file for a custom frequency and SCSR Register setting.

- 5.1. **Using Code Composer Studio (CCS), load the Clear_Erase.wks.** This workspace contains the Clear_Erase.pjt project. If you prefer a different workspace, you can just load the project file directly. This is the CCS project used to build the SDFlash clear and erase algorithm file. For a typical install, the workspace and project will be in the following directory:

```
LF2407A: <SDFlash base>\myprojects\lf2407a_Oct-2005\Clear_Erase\
LF2406A: <SDFlash base>\myprojects\lf2407a_Oct-2005\Clear_Erase\
LF2403A: <SDFlash base>\myprojects\lf2407a_Oct-2005\Clear_Erase\
LF2402A: <SDFlash base>\myprojects\lf2407a_Oct-2005\Clear_Erase\
LF2401A: <SDFlash base>\myprojects\lf2401a_Oct-2005\Clear_Erase\
```



These projects contain the following files:

C2xx_bcex.asm – Main functions called by the SDFlash front-end. These functions interface directly to the LF240xA Flash kernel.

ce_alg.asm – LF240xA clear/erase Flash kernel. Do not edit this file.

C2xx_bcex.cmd – Linker command file. Do not edit this file.

VAR.h – Include file that contains PLL_RATIO_CONST parameter. Modify to achieve desired CPU frequency.

timings.40 – Flash parameter timings file. Modify per Excel spreadsheet to match CPU frequency.

5.2. Specify the required PLL multiplier ratio in the SCSR Register.

In CCS, open and modify VAR.h to specify the PLL_RATIO_CONST parameter value. Uncomment the line corresponding to the required System Control and Status Register (SCSR) setting. This is done by removing the leading semicolon in front of the correct line. Only one line should be uncommented. For example: To have the algorithms initialize the SCSR register for a PLL frequency of x2 of the input clock frequency, uncomment the second line and comment out the remaining lines as shown:

```
;PLL_RATIO_CONST      .set  0000h      ; CPU_CLK = INPUTCLK * 4.00
PLL_RATIO_CONST       .set  0200h      ; CPU_CLK = INPUTCLK * 2.00
;PLL_RATIO_CONST      .set  0400h      ; CPU_CLK = INPUTCLK * 1.33
etc ...
```

CAUTION

For flash integrity at operation frequencies, the device should always be programmed at the fastest possible CPU frequency. For example, if the CLKIN frequency is 10 MHz, program the device at 40 MHz rather than 10 MHz or 20 MHz.

- 5.3. **Specify the Flash timing parameters to match the CPU frequency.** In CCS open VAR.h and replace the .include timings.40 file with the appropriate timing file. The timings.40 file sets up the parameters and timings loops for a 40 MHz CPU clock. A timings.30 file is also provided. If a CPU frequency of other than 30 or 40 MHz is desired, use the Timings.xls spreadsheet provided to calculate the various parameters for your specific frequency. Change the parameters to match the values calculated by the spreadsheet or save the spreadsheet as a *Formatted Text (space delimited)(*.prn)* file. Make sure to use quotations around the filename.

5.4. If using timings.30, make the following changes:

For LF2407A, LF2406A, LF2403A and LF2402A:

- Open C2xx_bcex.asm and change AlgoCRC3 from 0F06B921Fh to 0F42F5CB2h
- In the same file change AlgoCRC4 from 08994CD07h to 0922E3FEBh

For LF2401A:

- Open C2xx_bcex.asm and change AlgoCRC3 from 0CEE7DDEh to 0CAAEB373h
- In the same file change AlgoCRC4 from 0FE84E6BEh to 0B9396EF9h

5.5. If using a timing set other than timings.30 or timings.40 perform the following procedure to determine the AlgoCRC3 and AlgoCRC4 values:

- 5.5.1. Rebuild the algorithm.
- 5.5.2. Run SDFlash with the new algorithm.
- 5.5.3. Upon failure close SDFlash and reopen CCS. Load the symbols only of the Clear_Erase project (File->Load Symbols->Load Symbols Only... Choose C2xx_bcex.out).
- 5.5.4. Look at data space address 0x0313. This 16 bit variable should be labeled PRG_status and should contain the value 0x0032 meaning that the AlgoCRC3 check failed.
- 5.5.5. Look at data space address 0x031D-0x031E. This 32 bit variable should be labeled CheckSum and is in little endian format. Change AlgoCRC3 in C2xx_bcex.asm to this value (AlgoCRC3 should be set in big endian format, so you must reverse the order of the two words). For example if address 0x031D contains 0x1234 and 0x031E contains 0x5678, AlgoCRC3 should be set to 056781234h (don't forget the leading 0 or the trailing h).
- 5.5.6. Rebuild the algorithm.
- 5.5.7. Rerun SDFlash.
- 5.5.8. Upon failure close SDFlash and reopen CCS. Load the symbols of the project again and view the PRG_status variable at 0x0313 again. This time it should read 0x0033 meaning that the AlgoCRC4 check failed.
- 5.5.9. Look the CheckSum variable again and change AlgoCRC4 (swap little to big endian again) in C2xx_bcex.asm to match this value.

5.6. Rebuild the algorithm in CCS by selecting *Project->Rebuild all*. Check for Errors. Ignore any warnings about _c_int0. Loading the .out file after the build is not necessary.

5.7. Update the Program/Verify algorithm. Begin by opening the Prog_Ver.wks file, which contains the Prog_Ver.pjt project. For a typical install, the workspace and project will be in the following directory:

```
LF2407A: <SDFlash base>\myprojects\lf2407a_Oct-2005\Prog_Ver\
LF2406A: <SDFlash base>\myprojects\lf2407a_Oct-2005\Prog_Ver\
LF2403A: <SDFlash base>\myprojects\lf2407a_Oct-2005\Prog_Ver\
LF2402A: <SDFlash base>\myprojects\lf2407a_Oct-2005\Prog_Ver\
LF2401A: <SDFlash base>\myprojects\lf2401a_Oct-2005\Prog_Ver\
```

5.8. There is no need to modify the VAR.h file again. The same file is used by both the Program/Verify and the Clear/Erase algorithms.

5.9. If using timings.30, make the following change:

For LF2407A, LF2406A, LF2403A, LF2402A and LF2401A:

- Open C2xx_bpx.asm and change AlgoCRC2 from 07C07AABFh to 07ABF3729h.

- 5.10. **If using a timing set other than timings.30 or timings.40 perform the following procedure to determine the AlgoCRC2 value:**
- 5.10.1. Rebuild the algorithm.
 - 5.10.2. Run SDFlash with the new algorithm.
 - 5.10.3. Upon failure close SDFlash and reopen CCS. Load the symbols only of the Prog_Ver project (File->Load Symbols->Load Symbols Only... Choose C2xx_bpx.out).
 - 5.10.4. Look at data space address 0x09E1. This 16 bit variable should be labeled PRG_status and should contain the value 0x0021 meaning that the AlgoCRC2 check failed.
 - 5.10.5. Look at data space address 0x09EB-0x09EC. This 32 bit variable should be labeled CheckSum and is in little endian format. Change AlgoCRC2 in C2xx_bpx.asm to this value (AlgoCRC2 should be set in big endian format, so you must reverse the order of the two words).
- 5.11. **Rebuild the algorithm in CCS by selecting *Project->Rebuild all*.** Check for Errors. Ignore any warnings about _c_int0. Loading the .out file after the build is not necessary.
- 5.12. **Exit Code Composer Studio**

The SDFlash algorithm files should now be configured for your hardware's frequency requirements.

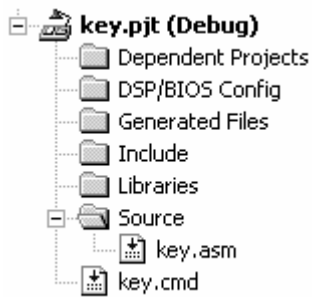
6. Code Security Module (CSM) Password Considerations

If the code security passwords are other than erased or cleared, the Code Security Module (CSM) on the device must be unlocked by the unlock algorithm before an erase, program, or verify operation. As supplied, the password locations are assumed to be all erased (all 0xFFFFs) and it is not necessary to run the unlock algorithm first. During the code development phase, it is suggested that the CSM passwords be left erased (0xFFFFs) for ease of use. If you program new passwords into the CSM password locations (0x0040 – 0x0043) and then later need to reprogram the part, you will need to configure and run the unlock algorithms so that the CSM can be unlocked. Refer to *TMS320LF/LC240xA DSP Controllers Reference Guide, System and Peripherals*, literature #SPRU357B, for details on the proper operation of the CSM. The unlock algorithm uses the CSM keys provided in the key.asm file to unlock the CSM. Follow these steps to build an SDFlash unlock algorithm file with a new set of CSM passwords.

6.1 Using Code Composer Studio, load the key.pjt project

For a typical install, the projects will be in the following directories:

```
LF2407A: <SDFlash base>\myprojects\lf2407a_Oct-2005\Key\
LF2406A: <SDFlash base>\myprojects\lf2407a_Oct-2005\Key\
LF2403A: <SDFlash base>\myprojects\lf2407a_Oct-2005\Key\
LF2402A: <SDFlash base>\myprojects\lf2407a_Oct-2005\Key\
LF2401A: <SDFlash base>\myprojects\lf2401a_Oct-2005\Key\
```



These projects contain the following files:

key.asm – Contains the password values. As supplied these values are 0x0040, 0x0041, 0x0042, and 0x0043. This would be the password for a valid “data equals address” pattern. Change these values to those programmed into your device.
key.cmd – Linker command file. Do not edit this file.

- 6.2 **In CCS, open the file key.asm.** Key.asm contains the definition of the CSM passwords used by the algorithm to unlock the CSM during the erase, program and verify operations. **Note:** The passwords in this file **will not** be programmed into the CSM password locations. The unlock algorithm uses these passwords only to unlock the CSM prior to an erase, program or verify operation.
- 6.3 **Modify the passwords in key.asm to match those already programmed into the CSM password locations.**
- 6.4 **Rebuild the algorithms in CCS by selecting *Project->Rebuild All*.** Check for Errors. Ignore any warnings about _c_int0. Loading the .out file after the build is not necessary.
- 6.5 **Exit Code Composer Studio**
- 6.6 **Open SDFlash**
- 6.7 **Load the unlock.sdp project.**

For a typical install, the projects will be in the following directories:

```
LF2407A: <SDFlash base>\myprojects\lf2407a_Oct-2005\
LF2406A: <SDFlash base>\myprojects\lf2407a_Oct-2005\
LF2403A: <SDFlash base>\myprojects\lf2407a_Oct-2005\
LF2402A: <SDFlash base>\myprojects\lf2407a_Oct-2005\
LF2401A: <SDFlash base>\myprojects\lf2401a_Oct-2005\
```

- 6.8 **Run the unlock routine: Device->Flash.** Uncheck the *Erase Box* and the *Verify Box*, leaving only the *Program Box* checked, then press *Start*. Your device is now unlocked and you can clear/erase/program/verify your device.

7. Error Messages and Troubleshooting Tips

7.1 Troubleshooting: All Operations

- ☐ **Important:** When a failure is seen during any of the operations there is a possibility that the Flash password contents are in an unknown state. If power is cycled when the passwords are unknown, the part will be permanently locked. Check the password locations (0x0040 – 0x0043) and note their contents before cycling power: *View->Target Buffer* in SDFlash, or *View->Memory* in CCS.
- ☐ It is recommended that the CLKOUT pin is monitored prior to executing the algorithm functions. If CLKOUT is not at the expected frequency/duty cycle, turn power off, reseal the part (if possible), and turn power back on. The expected frequency of CLKOUT at power up is CLKIN/2.
- ☐ LF240xA_July-2004 and later Flash algorithms are designed for SDFlash version 1.62.03 and later. The Prg2xx utility will not be updated to work with this version of the algorithms.
- ☐ Run SDConfig to make sure your target and emulator are setup properly.
- ☐ Examine the project settings *Project->Settings* and make sure the path names to all files are correct as described in section 4.7.
- ☐ Make sure the Code Security Module (CSM) is unlocked. An algo return error message of 0x0001, 0x0003 or 0x0025 can indicate the CSM as the source of the problem. If the password locations are cleared or erased, the algorithm attempts to unlock the CSM before an erase, program, or verify operation. By default, the password locations are assumed to be all erased (FFFFs). Refer to section 6 *Code Security Module (CSM) Password Considerations* if you have changed your passwords from the erased value. It is suggested that the CSM passwords be left erased (FFFFs) for initial development. Refer to the *TMS320LF/LC240xA DSP Controllers Reference Guide, System and Peripherals* for details on the CSM operation.
- ☐ If required, configure the SDFlash algorithms for a custom CPU frequency and PLL multiplier as well as flash parameters and timings. As supplied, the algorithms are configured for a CPU clock frequency (CLKOUT) of 40 MHz and to set the SCSR register to 0x0000. If your hardware has other requirements you must configure the algorithms as described in section 5.
- ☐ SD Flash should have full control of the device. That is, no user application should be running, no interrupts firing, and CCS should be shut down prior to using SDFlash.
- ☐ Make sure that the device has a clean VCCP 3.3V voltage source.
- ☐ Check the part using Code Composer Studio (CCS). Using CCS, check the SARAM blocks and make sure that you can unlock the CSM (check by attempting to view the passwords in a memory window. If you view garbage, with the same value throughout the flash, the CSM is still locked).
- ☐ Check the MPNMC pin (for LF2407A). The device must be in microcontroller mode (MPNMC is low) to modify the flash contents.

7.2 Troubleshooting: Erase Fails

- ❑ Make sure a valid sector mask is specified for User Option 1 for the erase operation. At least one sector must be specified. Do not use a 0x in front of the mask value – simply enter the hex number without the leading 0x. For example: 000F is correct, 0x000F is incorrect.
- ❑ If you programmed new security passwords, the SDFlash algorithm may no longer be able to unlock the CSM using the default passwords of all FFFFs or all 0000s. To avoid having to change the CSM passwords for different operations, it is suggested that the CSM passwords be left erased (FFFFs) for initial development. Refer to section 6 *Code Security Module (CSM) Password Considerations* if you have changed your passwords from the default (erased) value.

7.3 Troubleshooting: Programming Fails

- ❑ Check to make sure you are not programming a region of memory outside of the flash. To see what sections SDFlash will program, use the *View->Coff/Hex File Status* utility and look for sections labeled "load" outside of the flash memory region. It may be that the start address is within Flash but the end address is not. Any loaded section that starts outside or ends outside of the flash region will cause programming to fail.
- ❑ Make sure you erased the sectors being programmed prior to programming them.
- ❑ Make sure the sector mask used for the erase function did not include the leading 0x. For example: 000F is correct, 0x000F is incorrect.

7.4 Troubleshooting: Verify Fails

- ❑ Make sure that User Option 1 field is left blank for normal verifies. If this field contains a '1', the algorithm will try to verify that the Flash is erased. If this field contains a '2', the algorithm will try to verify that the Flash is cleared.

7.5 Troubleshooting: Programmed Application Fails To Run

This section offers suggestions if you find that your programmed .out file is not executing properly.

- ❑ Try programming the ToggleXF.out file provided. This simple program sets the PLL to x1 mode and toggles the XF pin at a visible frequency (approximately ½ Hz with a 10 MHz input clock). Either a scope or an LED can be used to verify that this program is running correctly. On the LF2401A device, since XF shares a pin with TMS, you will need to remove the JTAG connector from the board and reset the device to see XF toggle.
- ❑ **Getting Started in C and Assembly Code with the TMS320LF240x DSP, literature #SPRA755A**, is available for download from the TI website and provides a starting point for programming LF240xA devices.
- ❑ Constant sections such as .switch, .const/.econst should be linked to page 0 (program) memory. SDFlash will not program sections linked to page 1 (data) memory.
- ❑ To boot from internal Flash make sure that the MPNMC pin is low (for LF2407A) and the BOOTEN pin is high during a rising RSN pin. On the LF2401A the boot function is

defined by a combination of pins; make sure TDI/OPB5 pin is high during a rising RSN.

- For programs with a long C initialization routine, the watchdog may reset before the main function is reached and the watchdog disabled or serviced. In this case a small assembly routine can be inserted in the code to disable the watchdog before the branch to `_c_int0`.

7.6 Error Messages

Following is a description of error messages displayed as the Algo return value.

0x0000	Algorithm passed
0x0001	Failed Clear Operation. Failed to clear the entire sector(s) specified. This also shows up when an erase attempt is executed when the Code Security Module is locked. See section 6, Code Security Module (CSM) Password Considerations, on how to unlock the device.
0x0002	Failed Erase Operation. Failed to erase the entire sector(s) specified.
0x0003	Failed Program Operation. Failed to program the entire specified .out file into the flash. This also shows up when an erase attempt is executed when the Code Security Module is locked. See section 6, Code Security Module (CSM) Password Considerations, on how to unlock the device.
0x0004	Zero bit failure. Algorithm was asked to program a '1' into one or more bits already at a '0' state, i.e., flash not erased prior to programming.
0x000A	Zero sector mask. Clear/Erase was called without any sectors enabled. Check <i>User Options 1</i> box in <i>Erase Tab</i> . See section 4.7 for more details.
0x0010	Programming algorithm called out of sequence. The programming algorithm was called (not executed) when the Clear/Erase algorithm should have been. This error can also occur if the Clear/Erase algorithm file is specified as the <i>Algorithm File</i> in the <i>Programming Tab</i> . See section 4.7 for more details.
0x0011	Verify algorithm called out of sequence. The verify algorithm was called (not executed) when the Clear/Erase algorithm should have been. This error can also occur if the Clear/Erase algorithm file is specified as the <i>Algorithm File</i> in the <i>Verify Tab</i> . See section 4.7 for more details.
0x0014	Clear/Erase algorithm called out of sequence. The clear/erase algorithm was called (not executed) when the Programming algorithm should have been. This error can also occur if the Programming algorithm file is specified as the <i>Algorithm File</i> in the <i>Erase Tab</i> . See section 4.7 for more details.
0x0015	Incorrect Verify algorithm called. Standard, non-CRC, verify algorithm was incorrectly called (not executed).
0x0018	Return Error. An error trap at the beginning of SARAM (0x8000) was executed. Most likely cause if a communication failure between the device and the host. Check password locations, turn power off, remove and reseat device (if possible), apply power and retry operation.
0x0020	Programming algorithm wrapper CRC failure. The wrapper portion of the programming algorithm failed a CRC check. Programming algorithm did not execute. Most likely cause is a communication failure between the device and the host. Check password locations, turn power off, remove and reseat device (if possible), apply power and retry programming attempt.
0x0021	Programming algorithm kernel CRC failure. The kernel portion of the programming algorithm failed a CRC check. Programming algorithm did not execute. Most likely cause is a communication failure between the device and the host. Check password locations, turn power off, remove and reseat device (if possible), apply power and retry programming attempt.
0x0022	Programming algorithm SPL_text CRC failure. A special section of the wrapper portion of the programming algorithm failed a CRC check.

- Programming algorithm did not execute. Most likely cause is a communication failure between the device and the host. Check password locations, turn power off, remove and reseat device (if possible), apply power and retry programming attempt.
- 0x0023 **Data buffer CRC failure.** The data transferred to the buffer does not generate the transmitted CRC value. Programming algorithm did not execute. Most likely cause is a communication failure between the device and the host. Check password locations, turn power off, remove and reseat device (if possible), apply power and retry programming attempt.
- 0x0024 **Flash data CRC failure after programming.** The data that was programmed into the Flash does not generate the transmitted CRC value. **Do not cycle power. There is a possibility that the Flash password contents are in an unknown state.** If power is cycled when the passwords are unknown, the part will be permanently locked. Check the password locations (0x0040 – 0x0043) and note their contents before cycling power: *View->Target Buffer* then *Buffer->Target Load* in SDFlash, or *View->Memory* in CCS.
- 0x0025 **Flash data CRC failure during verify.** The data that was programmed into the Flash does not generate the transmitted CRC value. **Do not cycle power. There is a possibility that the Flash password contents are in an unknown state.** If power is cycled when the passwords are unknown, the part will be permanently locked. Check the password locations (0x0040 – 0x0043) and note their contents before cycling power: *View->Target Buffer* then *Buffer->Target Load* in SDFlash, or *View->Memory* in CCS.
- 0x0026 **Verify FFFF fail.** The verify routine failed when checking if the device was erased. If a verify FFFF was not intended clear the *User Options 1* box in the *Verify Tab*. See section 4.7 for more details.
- 0x0027 **Verify 0000 fail.** The verify routine failed when checking if the device was cleared. If a verify 0000 was not intended clear the *User Options 1* box in the *Verify Tab*. See section 4.7 for more details.
- 0x0030 **Clear/Erase algorithm wrapper CRC failure.** The wrapper portion of the clear/erase algorithm failed a CRC check. Clear/Erase algorithm did not execute. Most likely cause is a communication failure between the device and the host. Check password locations, turn power off, remove and reseat device (if possible), apply power and retry clear/erase attempt.
- 0x0031 **Clear/Erase algorithm SPL_text CRC failure.** A special section of the wrapper portion of the clear/erase algorithm failed a CRC check. Clear/Erase algorithm did not execute. Most likely cause is a communication failure between the device and the host. Check password locations, turn power off, remove and reseat device (if possible), apply power and retry clear/erase attempt.
- 0x0032 **Clear/Erase algorithm kernel section 1 CRC failure.** The 1st section of the kernel portion of the clear/erase algorithm failed a CRC check. Clear/Erase algorithm did not execute. Most likely cause is a communication failure between the device and the host. Check password locations, turn power off, remove and reseat device (if possible), apply power and retry clear/erase attempt.
- 0x0033 **Clear/Erase algorithm kernel section 2 CRC failure.** The 2nd section of the kernel portion of the clear/erase algorithm failed a CRC check. Clear/Erase algorithm did not execute. Most likely cause is a communication failure between the device and the host. Check password locations, turn power off, remove and reseat device (if possible), apply power and retry clear/erase attempt.
- 0x1111 **Failed execution.** Algorithm did not execute.