

Programming the TMX320F206

This file explains the utilities that are necessary to program the TMS320F206 flash arrays using JTAG loader.

JTAG Loader : PRG2XXW.EXE with EMU2XXD.DLL

Environment : IBMPC compatible with Windows 95 or Windows 3.1 only

TI tools : XDS510 JTAG emulator and software installed in PCs

Program type : DOS window command.

IMPORTANT CONSIDERATIONS FOR VERSION 1.10 OF THE 'F206 PROGRAM LOADER

The current version 1.10 of the loader is a beta version. The loader and flash algorithms are being revised to give error free operation.

1. The supply voltage should be maintained to be at 5v (5%). Close tolerance to 5V will help flash programming.
2. F206 should run only at 20 MIPS, that is CLKOUT1 should be 20Mhz/50ns. All the flash program timings are based on this clock. DO NOT CHANGE CLOCK speed with these flash programming utilities.
3. Error -114 can occur during CLEAR or ERASE as the program currently shares the same error flag and program function.
 - 3a. ERROR 114 during CLEAR.
If it occurs during CLEAR - array is being written 0, then the device is in depletion. The current algorithms do not have the recovery features. Future versions of the software will discuss about this issue.
 - 3b. ERROR 114 during ERASE.
This could be due to supply voltage (below 4.75V). Check for the voltage and re-run CLEAR and then ERASE algorithm.
If the supply voltage is maintained at 5V during CLEAR/ERASE and the entire array is cleared and erased, this error should not occur.
4. Algorithms:

The flash algorithms for CLEAR, ERASE and PROGRAM will be available only as object files (.obj). DO NOT use these files except in the form suggested. More documentation will be provided in future revisions of these utilities. This will provide more flexibility in using these functions along with any application.

5. For any issues on F206 programming, please contact TI. Send an email to DSP hotline with the subject marked as 'F206 - Programming'. Refer to TI's WEB site for future updates on software and device.
TI DSP hotline address: dsph@ti.com TI Internet address : www.ti.com/dsps

1. PRG2XXW.exe

Function : Used to download control code, algorithms and flash code using JTAG link on the TMS320F206 devices. Could be used to clear, erase and program Flash0 -16k or Flash1 -16k. One 16k array at a time.

Resources : The Flash array programming can be done either using the SARAM and used B1 RAM (OPTION 1) or B0 and B1 RAM (OPTION 2).

Steps to get familiar with this documentation :

- 1.Understand the flash programming logic and flow. ITEMS 3,4.
- 2.Understand the JTAG loader PRG2XXW command line format and its components. ITEM 2.
- 3.See the f206p2.ppt, foils or PDF files for flash program memory maps and variable mapping.
- 4.Read about the utilities provided for OPTION 1 and 2 and choose the option suitable for your application. ITEM 5,6.
- 5.Check the list of batch files for testing the loader, with the XDS510 JTAG emulator.
- 6.Run any of the test programs to check the connections.
- 7.Choose the appropriate batch files for CLEAR,ERASE and PROGRAM functions.

2. JTAG loader Command format

PRG2XXW [-options] c2xxprog.out flashcode.out

2.1 [options] - Descriptions

- p 280 - specifies IO address for XDS510 card in the PC.- optional default 240.
- w (1-6) - specifies timeout limit for the host while programming. Necessary to match fast and slow PCs. -optional
- h - Help, Lists the options
- i xxxxxh - I/O address to be initialized before program loading
Used initialize the SARAM mapping in the PMST register at 0xFFE4h.Default 0xFFE4
- m xxxxxh - Value to be written in the I/O address mentioned with i-option
Default value to be written in the PMST register is 0x0006.
To initialize PMST register at 0xFFE4, just use -m option with 0x0006.-i option is not required.
- o - No flashcode COFF file for programming
- t xxxxxh - Initialize ST1 with XXXXX, default 0x17fc
- e - Run PRG_erase function before executing PRG_prog function
- v - Run PRG_verify function after executing PRG_prog functions
- default - Will run PRG_init first followed by PRG_prog and PRG_stop.
These functions are defined in the c2xxprog.out

Current JTAG loader ver1.10 WILL NOT USE -v options!!!

2.1 c2xxprog - Description

C2XXPROG.out - Generic name for the assembler program c2xxprog.asm which carries all the control functions, and flash algorithms

Functions defined in this program are:

PRG_init - Initializes the device with customer specific logic if necessary, before running other functions.

PRG_program - Default function executed after PRG_init.

PRG_erase - Used with -e option, currently used for CLEAR and ERASE in OPTION 1 and CLEAR only in OPTION 2.

PRG_verify - Used with -v option, currently not used

PRG_stop. - Exit function with software break point after executing PRG_program

Refer to any c2xxprog.asm program for more details on these functions. The F206 devices are generally erased when shipped. They can be used to program directly.

Resources used:

Uses B1 ram 0x300h -0x31fh as algorithm variable space

2.2 Flashcode - Descriptions

Flashcode.out - This is a generic name for the code to be flashed in FLASH0 or FLASH1 array. In version 1.10 of the loader these COFF files are to be generated only with the program code, located anywhere in the Flash array. That is use only one .text section for your code, NO DATA SECTIONS should be specified clearly under .data directive.

Resources used:

Uses B1 ram (from 0x320h),200 locations or SARAM (from 0xc00) 2000 locations as a temporary buffer for partially loading the COFF file before being programmed into the Flash area specified by the COFF header.

PRG2XXW has the COFF loader to load the flash code in blocks of 200 words into B1 ram (OPTION 2) before executing the functions that transfer the buffer data to flash memory. In OPTION 2 the same COFF loader is used to load SARAM in blocks of 2000 words.

3. Flash programming logic:

The flash array has to be prepared in the following sequence. This sequence is to be followed for flash0 and flash1 array separately. The current version does not allow contiguous programming of flash 0 and flash1 array. This suggests that the program has to be broken for each array into separate COFF files.

clear - make all bits zero 0
erase - make all bits one 1
program - make selected bits 0

4. Flash programming flow.

1. Prepare your COFF file (flashcode.out) for FLASH0 or FLASH1 array with a single .text section. Do not explicitly place data under .data section.
2. If its a fresh F206 sample the should be erased and hence proceed to step 8.
Its a good idea to check if all the required flash array is erased to 0xFFFFh.
3. YES, all the array locations read 0xFFFFh. Proceed to step 8.
If it is not 0xFFFFh, go to step 4 or 5.
4. NO. If all the array locations read 0x0000h, then the array is to be ERASED. It is in cleared state. Proceed to step 7.
5. NO, partially cleared/erased. Proceed to step 6.
The array is to be CLEARED and ERASED before proceeding to program.
6. Run CLEAR batch files or through the debugger to CLEAR the required array. The current version of the CLEAR algorithm will clear the entire flash array0 or 1. Once the array reads 0x0000h, proceed to next step. If the array never gets CLEARED (ERROR 114), then exit to 10.
7. Run ERASE batch files or through the debugger to ERASE the required array. The current version of the ERASE algorithm will erase the entire flash array0 or 1. Once the array reads 0xFFFFh, proceed to next step. If the array never gets ERASED (ERROR 114), then exit to 10.
8. This is the programming stage. The entire array is to be in ERASED condition -0xFFFFh. If you are programming for the first time run any of the test batch files in PROGRAM examples. ELSE go to step 9.

Verify through the debugger. If its TI debugger use 'take command' using sf0.cmd or sf1.cmd files. This would display a watch window with data variables used in the loader. If these variables are as per requirement then proceed to program, step 9. This stage also confirms that the JTAG connections are working correctly.
9. The programming is done using any of the program batch files, with your specific flashcode COFF file. The PRG2XXW should display the status of the programming (ERROR 112). If necessary, check the programmed section through the debugger.
- 10.EXIT. The array is over-erased or bad. Await future revisions of the flash utilities, to explain more on these. If any abnormal error is observed, please contact TI.

5.OPTION 1. Programs for Flash programming using SARAM

5.1. Memory reserved for flash programming:

Uses SARAM ram as program and data space

0x8000 - 0x83ff - 1Kx16 Program space for algorithms
0x0c00 - 0x13cf - 2Kx16 Data space for flash code
2000 words buffer.

Uses B2 ram as variable space

0x60h - 0x68h is reserved for PRG2XXW.exe

Uses B1 RAM

0x300h - 0x30fh as algorithm variable space
0x310h - 0x31fh as c2xxprog variable space

5.2. Batch files for executing flash functions using JTAG loader

; TEST PROGRAMS

ST0 BAT ; Flash 0 program loader test file using SARAM
ST1 BAT ; Flash 1 program loader test file using SARAM

; CLEAR and ERASE ALGORITHMS

SCE0 BAT ; Clear and Erase - Flash 0 array
If Clear operation fails, the program will exit without executing Erase algorithm.
Note in OPTION 1 the CLEAR and ERASE are done in one batch file.

SCE1 BAT ; Clear and Erase - Flash 1 array
If Clear operation fails, the program will exit without executing Erase algorithm.

; PROGRAM ALGORITHM

; Note the flashcode .out should be replaced with the required COFF file.
The COFF files used are only for demonstration of this function.

SP0 BAT ; Program Flash 0 array with l20.out
SP1 BAT ; Program Flash 1 array with l21.out
SPK0 BAT ; Program Flash 0 array with l16k0.out
SPK1 BAT ; Program Flash 1 array with l16k1.out

CAUTION: The following programs should be used after the individual flash functions are fully functional on the respective F2xx hardware.

SCEP0 BAT ; Clear,Erase and Program - Flash 0 array
SCEP1 BAT ; Clear,Erase and Program - Flash 1 array
; If Clear operation fails, the program will exit
without executing Erase and Program algorithms.

5.3. Assembler (c2xxprog) source files for SARAM

Note: Programs for Flash 0 and Flash 1 array differ only in address variable definition in c2xxprog files. Specifically, the variables FL_ST, FL_END and PRG_paddr are defined according to the array address map.

C2XX_SP0 ASM ; Program with PROGRAM,CLEAR and ERASE algorithms - FLASH0
C2XX_SP1 ASM ; Program with PROGRAM,CLEAR and ERASE algorithms - FLASH1

C2XX_ST0 ASM ; Flash 0 program loader test file
C2XX_ST1 ASM ; Flash 1 program loader test file

5.4 Linker command files for linking c2xxprog and flash algorithms for flash0 and flash1 array

C2XX_SP0 CMD ; SARAM loader for flash 0
C2XX_SP1 CMD ; SARAM loader for flash 1

C2XX_ST0 CMD ; SARAM test loader for flash 0
C2XX_ST1 CMD ; SARAM test loader for flash 1

5.5. Command files to initialize memory windows in TI debugger

SF0 CMD ; Displays FLASH0 variables and memory
SF1 CMD ; Displays FLASH1 variables and memory
SR0 CMD ; Initializes SARAM program to run from TI debugger
BR0 CMD ; Initializes B0 program to run from TI debugger

5.6 Time estimates based on the JTAG loader running on a 486PC at 66Mhz :

- a. Clear - 16K - approx. 8 secs
- b. Erase - 16K - approx. 2 secs
- c. Program - 16k - approx.18 secs

6. OPTION 2. Programs for Flash programming using B0 - RAM

6.1. Memory map reserved for programming:

Uses B0 ram as program space

Uses B2 ram as variable space

0x60h - 0x68h is reserved for PRG2XXW.exe

Uses B1 RAM

0x300h - 0x30fh as algorithm variable space

0x310h - 0x31fh as c2xxprog variable space

0x320h - 0x3e7h as temporary buffer for flash code
200 words buffer.

6.2. Batch files for executing flash functions using JTAG loader

; TEST PROGRAMS

BT0 BAT ; Flash 0 program loader test file using B0 RAM

BT1 BAT ; Flash 1 program loader test file using B0 RAM

; CLEAR ALGORITHM

BC0 BAT ; Clear - Flash 0 array

BC1 BAT ; Clear - Flash 1 array

; ERASE ALGORITHM

BE0 BAT ; Erase - Flash 0 array

BE1 BAT ; Erase - Flash 1 array

; PROGRAM ALGORITHM

BP0 BAT ; Program - Flash 0 array with l20.out

BP1 BAT ; Program - Flash 1 array with l21.out

6.3. Assembler (c2xxprog) source files for B0-RAM

Note: Programs for Flash 0 and Flash 1 array differ only in address variable definition in c2xxprog files. Specifically, the variables FL_ST ,FL_END and PRG_paddr are defined according to the array address.

C2XX_BC0 ASM ; CLEAR program for FLASH0

C2XX_BC1 ASM ; CLEAR program for FLASH1

C2XX_BE0 ASM ; ERASE program for FLASH0

C2XX_BE1 ASM ; ERASE program for FLASH1

C2XX_BP0 ASM ; PROGRAM algorithm for FLASH0

C2XX_BP1 ASM ; PROGRAM algorithm for FLASH1

C2XX_BT0 ASM ; B0 Program loader test for FLASH0

C2XX_BT1 ASM ; B0 Program loader test for FLASH1

6.4. Linker command files for linking c2xxprog and flash algorithms for flash0 and flash1 array

C2XX_BC0 CMD ; CLEAR
C2XX_BC1 CMD

C2XX_BE0 CMD ; ERASE
C2XX_BE1 CMD

C2XX_BP0 CMD ; PROGRAM
C2XX_BP1 CMD

C2XX_BT0 CMD ; B0 Program loader
C2XX_BT1 CMD

6.6 Time estimates based on the JTAG loader running on a 486PC at 66Mhz :

- a. Clear - 16K - approx. 8 secs
- b. Erase - 16K - approx. 2 secs
- c. Program - 16k - approx. 22 secs

7. B0/SARAM FLASH Programming algorithms object files.

All these programs reference PRG_parm table (0x310- 0x31f) for flash array variables. This area is to be protected while programming.

SCLR0 OBJ ; GCLR Clear algorithm
SERA0 OBJ ; GERS Erase algorithm
SPGMV0 OBJ ; GPGM Program alogrithm
SUTILS0 OBJ ; Flash utilities
SVAR0 H ; Flash programming algorithm variables.

The flash algorithms for CLEAR, ERASE and PROGRAM will be available only as object files (.obj). DO NOT use these files except in the form suggested. More documentation will be provided in future revisions of these utilities. This will provide more flexibility in using these functions along with any application.

7. List of files shipped with FLREV1.ZIP

7.1 Exe.files:

Prg2xxw.exe and Emu2xxd.dll - JTAG loader version 1.10

7.2 Text files:

Readme.txt

7.3 Assembler and COFF files:

.asm files for c2xxprog files - clear, erase and program
.cmd files for rebuilding files
.obj files for flash algorithms - clear, erase and program
.out files for using the XDS510 debugger and Prg2xxw
.h files used to declare flash0 or flash1 array programming variables

7.4 Batch files:

These files execute PRG2XXW with necessary options to perform clear, erase and programming of the flash.

7.5 Take files:

BR0.cmd - to configure PC and ST1 register using TI debugger

SR0.cmd - to configure PC and ST1 register using TI debugger

SF0.cmd - configures flash0 memory for display. Displays (watch window,in TI debugger)the variables used while programming

SF1.cmd - configures flash1 memory for display. Displays (watch window,in TI debugger)the variables used while programming

7.6 COFF files:

l20.out	: Sample Flashcode files for Flash0 numbers 0-1f
l21.out	: Sample Flashcode files for Flash1 numbers 0-1f
l16k0.out	: Sample Flashcode files for Flash0 numbers 0-3fff
l16k1.out	: Sample Flashcode files for Flash1 numbers 0-3fff
l20.asm,cmd	: Assembler source and linker command files for sample
l16k.asm,cmd	: COFF files.

7.7 PowerPoint file (Version 4.0)

F206P2.PPT - Slides on the 'F206 device and flash utility.

PDF files

F206P2.PDF - Adobe Acrobat version of F206P1.PPT

Readme.PDF - Acrobat version of this file.

Microsoft Word Files

Readme.doc - Word 2.0 version of this file

8. Status and Error Messages:

Once the PRG2xxw is executed, the status of the flash algorithm will be displayed. The following codes explain the common status and error messages.

"Processor Initialization",	/*ERR_INIT	100 */
"Processor Reset",	/*ERR_RESET	101 */
"Processor Register Write",	/*ERR_REGWR	102 */
"Processor Memory Write",	/*ERR_MEMWR	103 */
"Processor Memory Read",	/*ERR_MEMRD	104 */
"Processor Memory Fill Not Allowed",	/*ERR_MEMFI	105 */
"Processor Run",	/*ERR_RUN	106 */
"Processor Halt",	/*ERR_HALT	107 */
"Processor Status",	/*ERR_STATUS	108 */
"Processor Timeout",	/*ERR_TIMEOUT	109 */
"File Open",	/*ERR_FOPEN	110 */
"COFF Load",	/*ERR_LOAD	111 */
"Verify",	/*ERR_VERIFY	112 */
"Program",	/*ERR_PROGRAM	113 */
"Erase"	/*ERR_ERASE	114 */
"Missing symbol"	/*ERR_SYMBOL	115 */