

# **TMS320C3x** ***Evaluation Module***

*Installation  
Guide*



# ***TMS320C3x Evaluation Module Installation Guide***

SPRU120  
November 1993



## **IMPORTANT NOTICE**

Texas Instruments (TI) reserves the right to make changes to its products or to discontinue any semiconductor product or service without notice, and advises its customers to obtain the latest version of relevant information to verify, before placing orders, that the information being relied on is current.

TI warrants performance of its semiconductor products and related software to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are utilized to the extent TI deems necessary to support this warranty. Specific testing of all parameters of each device is not necessarily performed, except those mandated by government requirements.

Certain applications using semiconductor products may involve potential risks of death, personal injury, or severe property or environmental damage ("Critical Applications").

**TI SEMICONDUCTOR PRODUCTS ARE NOT DESIGNED, INTENDED, AUTHORIZED, OR WARRANTED TO BE SUITABLE FOR USE IN LIFE-SUPPORT APPLICATIONS, DEVICES OR SYSTEMS OR OTHER CRITICAL APPLICATIONS.**

Inclusion of TI products in such applications is understood to be fully at the risk of the customer. Use of TI products in such applications requires the written approval of an appropriate TI officer. Questions concerning potential risk applications should be directed to TI through a local SC sales office.

In order to minimize risks associated with the customer's applications, adequate design and operating safeguards should be provided by the customer to minimize inherent or procedural hazards.

TI assumes no liability for applications assistance, customer product design, software performance, or infringement of patents or services described herein. Nor does TI warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right of TI covering or relating to any combination, machine, or process in which such semiconductor products or services might be or are used.

### **WARNING**

This equipment is intended for use in a laboratory test environment only. It generates, uses, and can radiate radio frequency energy and has not been tested for compliance with the limits of computing devices pursuant to subpart J of part 15 of FCC rules, which are designed to provide reasonable protection against radio frequency interference. Operation of this equipment in other environments may cause interference with radio communications, in which case the user at his own expense will be required to take whatever measures may be required to correct this interference.

### **TRADEMARKS**

MS-DOS and MS-Windows are registered trademarks of Microsoft Corp.

PC-DOS is a trademark of International Business Machines Corp.

# Installing the Evaluation Module and the C Source Debugger

---

---

---

This guide helps you install the TMS320C3x evaluation module (EVM) and the C source debugger on a PC running MS-DOS or PC-DOS. You can also use the debugger with MS-Windows. When you complete the installation, turn to the *TMS320C3x C Source Debugger User's Guide*.

Topic	Page
<b>1. What You'll Need</b>	<b>2</b>
Hardware checklist	2
Software checklist	3
<b>2. Step 1: Installing the EVM Board in Your PC</b>	<b>4</b>
Preparing the EVM board for installation	4
Setting the EVM board into your PC	5
<b>3. Step 2: Installing the Debugger Software</b>	<b>7</b>
<b>4. Step 3: Setting Up the Debugger Environment</b>	<b>7</b>
Invoking the new or modified batch file	8
Modifying the PATH statement	8
Setting up the environment variables	9
Identifying the correct I/O switches	10
<b>5. Step 4: Verifying the Installation</b>	<b>11</b>
Installation error messages	12
<b>6. Using the Debugger With MS-Windows</b>	<b>12</b>

## 1. What You'll Need

The following checklists detail items that are shipped with the 'C3x C source debugger and EVM and additional items you'll need to use these tools.

### **Hardware checklist**

- |                          |                                     |  |
|--------------------------|-------------------------------------|--|
| <input type="checkbox"/> | <b>host</b>                         | An IBM PC/AT or 100% compatible ISA/EISA-based PC with a hard-disk system and a 1.2-Mbyte floppy-disk drive  |
| <input type="checkbox"/> | <b>memory</b>                       | Minimum of 640K bytes; in addition, if you are running under MS-Windows, you'll need at least 256K bytes of extended memory.   |
| <input type="checkbox"/> | <b>display</b>                      | Monochrome or color (color recommended)  |
| <input type="checkbox"/> | <b>slot</b>                         | One 8- or 16-bit slot  |
| <input type="checkbox"/> | <b>EVM board power requirements</b> | Approximately 1 ampere @ 5 volts (5 watts)   |
| <input type="checkbox"/> | <b>optional hardware</b>            | A Microsoft-compatible mouse   |
| <input type="checkbox"/> |                                     | An EGA- or VGA-compatible graphics display card and a large monitor. The debugger has several options that allow you to change the overall size of the debugger display. If you have an EGA- or VGA-compatible graphics card, you can take advantage of some of these larger screen sizes. These larger screen sizes are most effective when used with a large (17" or 19") monitor. (To use a larger screen size, you must invoke the debugger with an appropriate option. For more information about options, refer to the invocation section in Chapter 1, <i>Overview of a Code Development and Debugging System</i> , in the <i>TMS320C3x C Source Debugger User's Guide</i> .) |
| <input type="checkbox"/> | <b>miscellaneous materials</b>      | Blank, formatted disks   |

**WARNING**

To minimize the risk of electric shock and fire hazard, be sure that all major components that you interface with Texas Instruments devices are limited in energy and certified by one or more of the following agencies: UL, CSA, VDE, or TUV.

**Software checklist**

- |                          |                         |   |
|--------------------------|-------------------------|---|
| <input type="checkbox"/> | <b>operating system</b> | MS-DOS or PC-DOS (version 3.0 or later)<br>Optional: MS-Windows (version 3.0 or later)  |
| <input type="checkbox"/> | <b>software tools</b>   | TMS320 floating-point family DSP ('C3x/'C4x) assembler and linker<br>Optional: TMS320C3x/C4x C compiler   |
| <input type="checkbox"/> | <b>required files</b> † | <i>evmrst.exe</i> resets the EVM  |
| <input type="checkbox"/> | <b>optional files</b> † | <i>init.cmd</i> is a file that contains debugger commands. The version of this file that's shipped with the debugger defines a 'C3x memory map. If this file isn't present when you invoke the debugger, then all memory is invalid at first. When you first start using the EVM, this memory map should be sufficient for your needs. Later, you may want to define your own memory map. For information about setting up your own memory map, refer to Chapter 5, <i>Defining a Memory Map</i> , in the <i>TMS320C3x C Source Debugger User's Guide</i> . |
| <input type="checkbox"/> | †                       | <i>init.clr</i> is a general-purpose screen configuration file. If this file isn't present when you invoke the debugger, the debugger uses the default screen configuration.  |
| <input type="checkbox"/> | †                       | The default configuration is for color monitors; an additional file, <i>mono.clr</i> , can be used for monochrome monitors. When you first start to use the debugger, the default screen configuration should be sufficient for your needs. Later, you may want to define your own custom configuration.<br><br>For information about these files and about setting up your own screen configuration, refer to Chapter 9, <i>Customizing the Debugger Display</i> , in the <i>TMS320C3x C Source Debugger User's Guide</i> .                                |

† Included as part of the debugger package

## 2. Step 1: Installing the EVM Board in Your PC

This section contains the hardware installation information for the EVM.

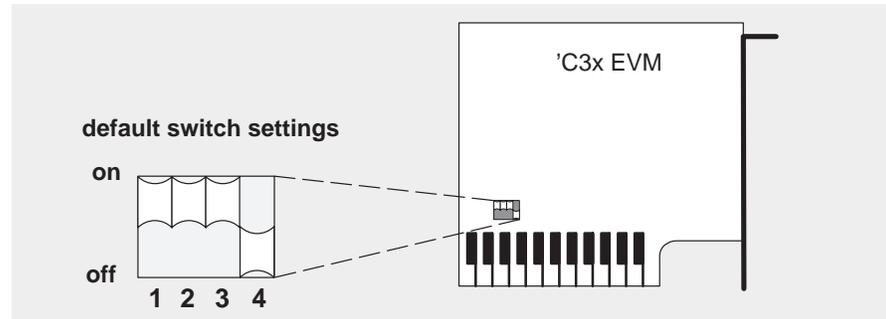
### ***Preparing the EVM board for installation***

Before you install the EVM board, you must be sure that the board's switches are set to correctly identify the I/O space that the board can use. The 'C3x EVM board has four switches:

- Switches 1 and 2 identify your system's I/O address space. You can change these switch settings to identify the I/O address space that the EVM uses in your system.
- Switches 3 and 4 are for manufacturing test. *Leave these switches in the default positions.*

Figure 1 shows where these switches are on the EVM board and identifies the switch numbers.

Figure 1. EVM Board I/O Switches



Switches are shipped in the default settings shown here and described in Table 1. If you use an I/O space that differs from the default, change the switch settings for switches 1 and 2. Table 1 shows you how to do this.

In most cases, you can leave the switch settings in the default position. However, you must ensure that the 'C3x EVM I/O address space does not conflict with other bus settings. For example, if you've installed a bus mouse in your system, you may not be able to use the default switch settings for the I/O address space—the mouse might use this space. Refer to your PC technical reference manual and your other hardware-board manuals to see if there are any I/O space conflicts. If you find a conflict, use one of the settings in Table 1.

Table 1. EVM Board Switch Settings

	Address Range	Switch #	
		1	2
default	0x0240–0x025F	on	on
	0x0280–0x029F	on	off
	0x0320–0x033F	off	on
	0x0340–0x035F	off	off

Some of the other installation steps require you to know which switch settings you used. If you reset the I/O switches, note the modified settings here for later reference.

Table 2. Your Switch Settings

	Address Range	Switch #	
		1	2
			

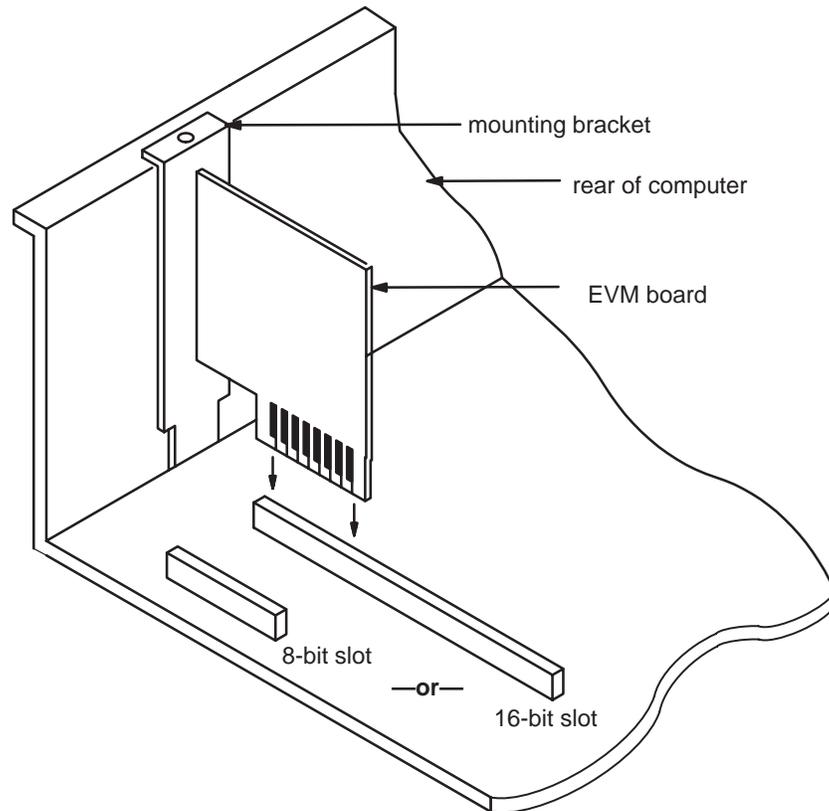
### Setting the EVM board into your PC

**WARNING**  
To minimize risk of personal injury, *always* turn off the power to all computer-system components and unplug power cords.

After you've prepared the EVM board for installation, follow these steps.

- Step 1:** Turn off your PC's power and unplug the power cord.
- Step 2:** Remove the cover of your PC.
- Step 3:** Remove the mounting bracket from an unused 8-bit or 16-bit slot.
- Step 4:** Install the EVM board in an 8-bit or 16-bit slot (see Figure 2).
- Step 5:** Tighten down the mounting bracket.
- Step 6:** Replace the PC cover.
- Step 7:** Plug in the power cord and turn on the PC's power.

Figure 2. EVM Board Installation



### 3. Step 2: Installing the Debugger Software

This section explains the simple process of installing the debugger software on a hard-disk system.

- 1) Make a backup copy of the DOS and/or MS-Windows debugger product disk. (If necessary, refer to the DOS manual that came with your computer.)
- 2) On your hard disk or system disk, create a directory named *c3xhll*. This directory will contain the 'C3x C source debugger software. To create this directory, enter:

```
MD C:\C3XHLL
```

- 3) Insert either the DOS or MS-Windows debugger product disk into drive A. Copy the contents of the disk.

```
COPY A:\*.* C:\C3XHLL\*.* /V
```

Repeat this step for the other product disk if you want to be able to run both the DOS and MS-Windows versions of the debugger.

The DOS version of the debugger executable is called *evm30.exe*, and the MS-Windows version of the debugger executable is called *evm30w.exe*. Throughout this document, the executable for the debugger is referred to as simply *evm30*.

### 4. Step 3: Setting Up the Debugger Environment

To ensure that your debugger works correctly, you must:

- Modify the PATH statement to identify the *c3xhll* directory.
- Define environment variables so that the debugger can find the files it needs.
- Identify any nondefault I/O space used by the EVM.



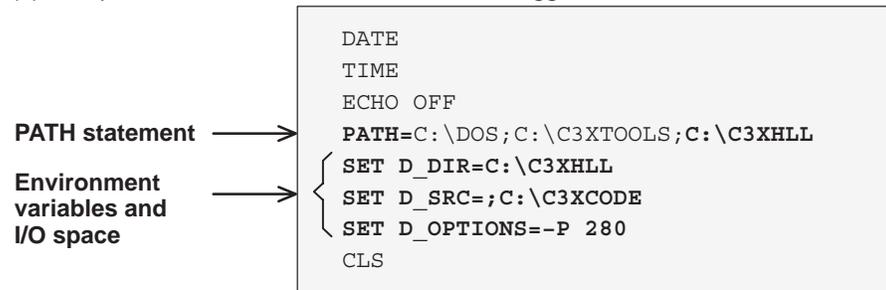
Not only must you do these things before you invoke the debugger for the first time, *you must do them any time you power up or reboot your PC.*

You can accomplish these tasks by entering individual DOS commands, but it's simpler to put the commands in a batch file. You can edit your system's *autoexec.bat* file; in some cases, modifying the *autoexec* may interfere with other applications running on your PC. So, if you prefer, you can create a separate batch file that performs these tasks.

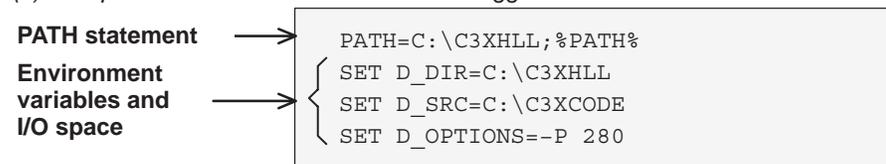
Figure 3 (a) shows an example of an autoexec.bat file that contains the suggested modifications (highlighted in bold type). Figure 3 (b) shows a sample batch file that you could create instead of editing the autoexec.bat file. (For the purpose of discussion, assume that this sample file is named *initdb.bat*.) The subsections following the figure explain these modifications.

Figure 3. DOS-Command Setup for the Debugger

(a) Sample autoexec.bat file to use with the debugger and EVM



(b) Sample initdb.bat file to use with the debugger and EVM



### Invoking the new or modified batch file

- If you modify the autoexec.bat file, be sure to invoke it before invoking the debugger for the first time. To invoke this file, enter:

**AUTOEXEC**

- If you create an initdb.bat file, you must invoke it before invoking the debugger for the first time. If you are using MS-Windows, invoke initdb.bat *before* entering MS-Windows. You'll need to invoke initdb.bat any time that you power-up or reboot your PC. To invoke this file, enter:

**INITDB**

### Modifying the PATH statement

Define a path to the debugger directory. The general format for doing this is:

**PATH=C:\C3XHLL**

This allows you to invoke the debugger without specifying the name of the directory that contains the debugger executable file.

- If you are modifying an autoexec that already contains a PATH statement, simply include ;C:\c3xhll at the end of the statement as shown in Figure 3 (a).
- If you are creating an initdb.bat file, use a different format for the PATH statement:

```
PATH=C:\C3XHLL;%PATH%
```

The addition of ;%path% ensures that this PATH statement won't undo PATH statements in any other batch files (including the autoexec.bat file).

### Setting up the environment variables

An environment variable is a special system symbol that the debugger uses for finding or obtaining certain types of information. The debugger uses three environment variables, named D\_DIR, D\_SRC, and D\_OPTIONS. The following tells you how to set up these environment variables. The format for doing this is the same for both the autoexec.bat and initdb.bat files.

- Set up the D\_DIR environment variable to identify the c3xhll directory:

```
SET D_DIR=C:\C3XHLL
```

(Be careful not to precede the equal sign with a space.)

This directory contains auxiliary files (evmrst, evminit.cmd, etc.) that the debugger needs.

- Set up the D\_SRC environment variable to identify any directories that contain program source files that you'll want to look at while you're debugging code. The general format for doing this is:

```
SET D_SRC=pathname1;pathname2...
```

For example, if your 'C3x programs were in a directory named c3xsrc on drive C, the D\_SRC setup would be:

```
SET D_SRC=C:\C3XSRC
```

- You can use several options when you invoke the debugger. If you use the same options over and over, it's convenient to specify them with D\_OPTIONS. The general format for doing this is:

```
SET D_OPTIONS= [object filename] [debugger options]
```

This tells the debugger to load the specified object file and use the specified options each time you invoke the debugger. These are the options that you can identify with D\_OPTIONS:

-b	-bb	-i <i>pathname</i>
-p <i>port address</i>	-s	-t <i>filename</i>
-profile	-v	

Note that you can override D\_OPTIONS by invoking the debugger with the -x option.

For more information about options, refer to the invocation section in Chapter 1, *Overview of a Code Development and Debugging System*, in the *TMS320C3x C Source Debugger User's Guide*.

### Identifying the correct I/O switches

Refer to your entries in Table 2 (page 5). If you didn't modify the I/O switches, skip this step.

If you modified the I/O switch settings, you must use the debugger's -p option to identify the I/O space that the EVM is using. You can do this each time you invoke the debugger, or you can specify this information by using the D\_OPTIONS environment variable. Table 3 lists the nondefault I/O switch setting and the appropriate line that you can add to the autoexec.bat or initdb.bat file.

Table 3. Identifying Nondefault I/O Address Space

Address Range	switch #		Add this line to the batch file
	1	2	
0x0280-0x029F	on	off	SET D_OPTIONS= -p 280
0x0320-0x033F	off	on	SET D_OPTIONS= -p 320
0x0340-0x035F	off	off	SET D_OPTIONS= -p 340

**Notes:**

- 1) The 'C3x EVM uses 96 bytes of the PC I/O space.
- 2) If you didn't note the I/O switch settings, you may use a trial-and-error approach to find the correct -p setting. **If you use the wrong setting, you'll see this error message when you try to invoke the debugger:**

```

CANNOT INITIALIZE THE EVM ! !
- Check I/O configuration
    
```

- 3) Never reset the 'C3x EVM with evmrst unless you have first loaded a valid object file to the EVM.

## 5. Step 4: Verifying the Installation

To ensure that you have correctly installed the EVM and debugger software, enter this command at the system prompt:

```
evm30 c:\c3xh11\sample
```

You should see a display similar to this one:

The screenshot displays the debugger's main window with several panes:

- DISASSEMBLY:** Shows assembly instructions with addresses, hex values, mnemonics, and operands. A watchpoint is set on `c_int00:`.
 

Address	Hex	Mnemonic	Operands
f00075	00f000b2	ABS	178, DP
f00076	087000f0	LDI	240, DP
f00077	08340074	LDI	@074H, SP
f00078	080b0014	LDI	SP, AR3
f00079	087000f0	LDI	240, DP
f0007a	08280075	LDI	@075H, AR0
f0007b	04e8ffff	CMPI	-1, AR0
f0007c	6a05000c	BZ	f00089
f0007d	08412001	LDI	*AR0++(1), R1
f0007e	6a250008	BZD	f00089
f0007f	08492001	LDI	*AR0++(1), AR1
f00080	08402001	LDI	*AR0++(1), R0
f00081	18610001	SUBI	1, R1
f00082	139b9991	RPTS	R1
f00083	da002120	LDI	*AR0++(1), R0    STI
- CPU:** Shows the status of various registers.
 

Register	Value	Register	Value
PC	00f00076	SP	00000755
R0	00000003	R1	00000005
R2	00000007	R3	00000000
R4	00000000	R5	00000000
R6	00000000	R7	00000000
AR0	00001802	AR1	00000000
AR2	00000000	AR3	00000000
AR4	00000000	AR5	00000000
AR6	00000000	AR7	00000000
IR0	00000000	IR1	00000000
ST	00000000	RC	00000000
RS	00000000	RE	00000000
DP	00000000	BK	00000000
IE	00000000	IF	00000000
- COMMAND:** Shows the execution of the `evm30` command and the loading of `sample.out`.
 

```
TMS3203x Debugger
Copyright (c) 1989, 1993 Texas In
TMS320C3x
Loading sample.out
Done
>>>
```
- MEMORY:** Shows a dump of memory addresses and their corresponding values.
 

Address	Value	Address	Value	Address	Value
000000	0000004b	00000040	00000041	00000042	
000004	00000043	00000044	00000045	00000046	
000008	00000047	00000048	00000049	0000004a	
00000c	00000000	00000000	00000000	00000000	
000010	00000000	00000000	00000000	00000000	
000014	00000000	00000000	00000000	00000000	

- If you see a display similar to this one, you have correctly installed your EVM and debugger.
- If you see a display and the lines of code show ADD instructions or say *Invalid address*, your EVM board may not be installed snugly. Check your board to see if it is correctly installed, and re-enter the command above.
- If you don't see a display, then your debugger or board may not be installed properly. Go back through the installation instructions and be sure that you have followed each step correctly; then re-enter the command above.

### **Installation error messages**

While invoking the debugger, you may see the following message:

```
CANNOT INITIALIZE THE EVM ! !  
- Check I/O configuration
```

To determine the problem, follow these actions:

- Check the EVM board to be sure it is installed snugly.
- Ensure that your port address is set correctly:
  - Check to be sure the `-p` option used with the `D_OPTIONS` environment variable matches the I/O address defined by your switch settings (refer to *Your Switch Settings*, Table 2, and *Identifying Nondefault I/O Address Space*, Table 3).
  - Check to see if you have a conflict in address space with another bus setting. If you have a conflict, change the switches on your board to one of the alternate settings in Table 1. Modify the `-p` option of the `D_OPTIONS` environment variable to reflect the change in your switch settings.

## **6. Using the Debugger With MS-Windows**

If you're using MS-Windows, you can freely move or resize the debugger display on the screen. If the resized display is bigger than the debugger requires, the extra space is not used. If the resized display is smaller than required, the display is clipped. Note that when the display is clipped, it can't be scrolled.

You may want to create an icon to make it easier to invoke the debugger from within the MS-Windows environment. Refer to your MS-Windows manual for details.

You should run MS-Windows in either the standard mode or the 386 enhanced mode to get the best results.