TEXAS INSTRUMENTS

TMS320C54x Evaluation Module

Installation Guide

1995 Microprocessor Development Systems

Printed in U.S.A., September 1995
D425003-9741 revision *

SPRU134

TMS320C54x Evaluation Module Installation Guide

SPRU134
September 1995

TEXAS INSTRUMENTS

Printed on Recycled Paper

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

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

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

Installing the Evaluation Module and the C Source Debugger

This guide helps you install the TMS320C54x evaluation module (EVM) board 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, refer to the *TMS320C5xx C Source Debugger User's Guide*.

Topic **Page**

1. What You'll Need

In addition to the materials that are shipped with the 'C54x EVM and C source debugger, you'll need the following hardware and software.

Hardware checklist

☐	**host**	An IBM PC/AT or 100% compatible ISA/EISA-based PC with a hard-disk system and a 1.44-megabyte, 3.5-inch floppy-disk drive
☐	**memory**	Minimum of 640K bytes. In addition, if you are running the tools under MS-Windows, you will need at least 256K bytes of extended memory.
☐	**display**	Monochrome or color (color recommended)
☐	**expansion slot**	One 16-bit slot
☐	**EVM board power requirements**	1.5 amperes at 5 volts 0.1 amperes at –5 volts 0.5 amperes at 12 volts 0.1 amperes at –12 volts
☐	**optional hardware**	A Microsoft-compatible mouse
☐		An EGA- or VGA-compatible graphics display card and a large monitor. The debugger has two 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 the *Overview of a Code Development and Debugging System* chapter of the *TMS320C5xx C Source Debugger User's Guide*.)
☐	**miscellaneous materials**	Blank, formatted disks

Software checklist

☐	**operating system**	MS-DOS or PC-DOS (version 3.0 or later) Optional: MS-Windows (version 3.0 or later)
☐	**software tools**	TMS320C54x assembler and linker Optional: TMS320C54x C compiler
☐	**optional files** †	*evmrst.exe* resets the EVM
☐	†	*evminit.cmd* is a batch file that contains debugger commands. The version of this file that's shipped with the debugger defines a 'C54x memory map. If this file isn't present when you first invoke the debugger, then all memory is invalid at first. When you first start using the debugger, 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 the *Defining a Memory Map* chapter of the *TMS320C5xx C Source Debugger User's Guide*.
☐	†	*init.clr* 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.
☐	†	*init.25* and *init.50* have been provided for basic 80x25 and 80x50 screen sizes, respectively. The init.clr file brings up the debugger in 80x25 mode. To bring the debugger up in another mode, copy one of the init.xx files to the init.clr file.
☐	†	The default configuration is for color monitors; an additional file, *mono.clr*, 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. For information about these files and about setting up your own screen configuration, refer to the *Customizing the Debugger Display* chapter of the *TMS320C5xx C Source Debugger User's Guide*.

† Included as part of the debugger package

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

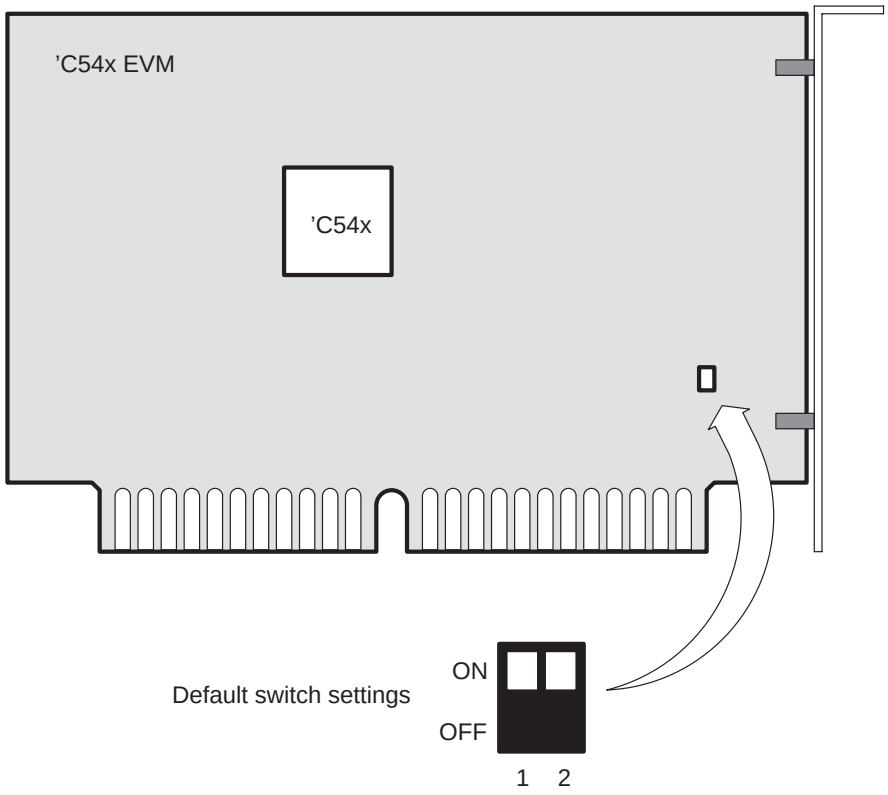
This section contains hardware installation information for the EVM.

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 EVM board has two switches that identify your system's I/O address space. You can change the switch settings to identify the I/O address space that the EVM uses in your system.

Figure 1 shows the location of these switches on the EVM board and identifies the switch numbers.

Figure 1. EVM Board I/O Switches



In most cases, you can leave the switch settings in the default position. However, you must ensure that the EVM I/O address space does not conflict with other system hardware. For example, if you have 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 circumvent any I/O space conflicts. If you find a conflict, use one of the settings in Table 1.

Table 1. EVM Board Switch Settings

		Switch #	
	Address Range	**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 in Table 2 below for later reference.

Table 2. Your Switch Settings

		Switch #	
	Address Range	**1**	**2**
⇨			

Setting the EVM board into your PC

After you have prepared the EVM board for installation, follow these steps:

Step 1: Turn off the power to the PC, and unplug the power cord.

Step 2: Remove the cover of the PC.

Step 3: Remove the mounting bracket from an unused 16-bit slot.

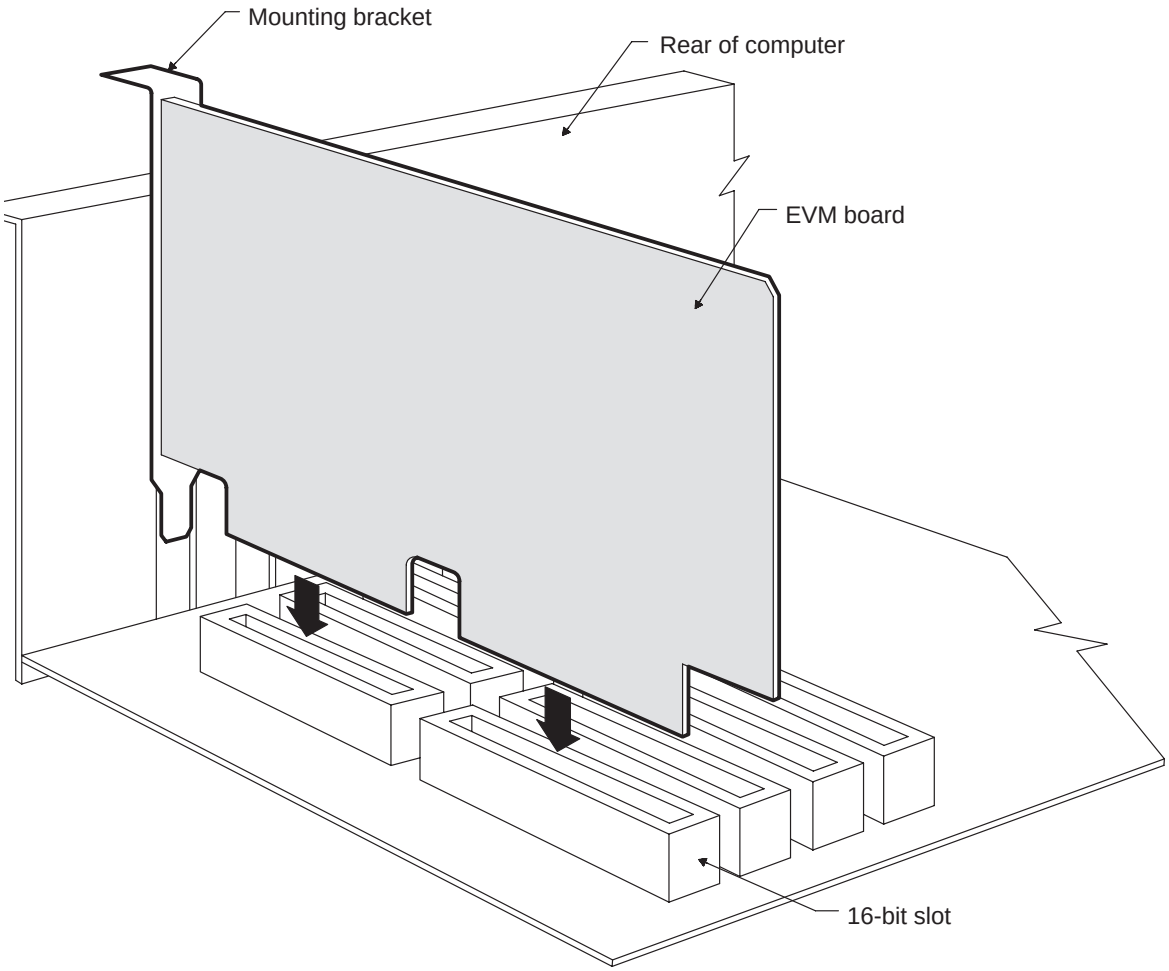
Step 4: Install the EVM board in a 16-bit slot (see Figure 2).

Step 5: Tighten the mounting bracket.

Step 6: Replace the PC cover.

Step 7: Plug in the power cord, and turn on the power to the PC.

Figure 2. EVM Board Installation



3. Step 2: Installing the Debugger Software

This section explains the 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 *c5xxhll*. This directory will contain the 'C54x C source debugger software. To create this directory, enter:

   MKDIR C:\C5xxHLL ↵

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

   COPY A:*.* C:\C5xxHLL*.* /V ↵

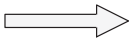
   Repeat this step for the other product disk if you want to be able to run both the DOS and MS-Windows versions of the debugger.

   The DOS version of the debugger executable is called evm5xx.exe, and the MS-Windows version of the debugger executable is called evm5xxw.exe. Throughout this document, the executable for the debugger is referred to as simply evm5xx.

4. Step 3: Setting Up the Debugger Environment

To ensure that your debugger works correctly, you must:

- ❑ Modify the PATH statement to include the C5xxhll directory.
- ❑ Define environment variables so that the debugger can locate all required files.
- ❑ 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 tasks by entering individual DOS commands, but you may find it simpler to put the commands in a batch file. You can edit the system's autoexec.bat file, but in some cases, modifying it may interfere with other applications running on your PC. So, you may prefer to create a separate batch file that performs these tasks.

Example 1 *(a)* shows an autoexec.bat file that contains the suggested modifications highlighted in bold type. Example 1 *(b)* shows a sample batch file that you could create instead of editing autoexec.bat. For the purpose of this discussion, assume that this sample file is named initdb.bat. The subsections following the example explain these modifications.

Example 1. DOS-Command Setup for the Debugger

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

PATH Statement →

Environment Variables and I/O Space →

```
DATE
TIME
ECHO OFF
PATH=C:\DOS;C:\C5xxTOOLS;C:\C5xxHLL
SET D_DIR=C:\C5xxHLL
SET D_SRC=C:\C5xxCODE
SET D_OPTIONS=-P 280
CLS
```

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

PATH Statement →

Environment Variables and I/O Space →

```
PATH=C:\C5xxHLL;%PATH%
SET D_DIR=C:\C5xxHLL
SET D_SRC=C:\C5xxCODE
SET D_OPTIONS=-P 280
```

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

To invoke the debugger without specifying the name of the directory that contains the debugger executable file, define a path to the debugger directory. The general format is:

```
PATH=C:\C5XXHLL
```

- If you are modifying an autoexec.bat file that already contains a PATH statement, simply include **;C:\c5xxhll** at the end of the statement, as shown in Example 1 *(a)*.

- If you are creating an initdb.bat file, use a different format for the PATH statement, as shown in Example 1 *(b)*:

  ```
  PATH=C:\C5XXHLL;%PATH%
  ```

  The addition of **;%path%** ensures that this PATH statement won't undo PATH statements in 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 and obtaining certain types of information. The debugger uses three environment variables named D_DIR, D_SRC, and D_OPTIONS. This section tells you how to set up these environment variables. For both the autoexec.bat and initdb.bat files, the format is the same.

- Set up the D_DIR environment variable to identify the c5xxhll directory:

  ```
  SET D_DIR=C:\C5XXHLL
  ```

  (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=*pathname*$_1$*;pathname*$_2$*...*

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

  For example, if your 'C54x programs were in a directory named *c5xxsrc* on drive C, the D_SRC setup would be:

  ```
  SET D_SRC=C:\C5XXSRC
  ```

- 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*]

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

  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[b]	–i *pathname*	–p *port address*
–s	–t *filename*	–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 the *Overview of a Code Development and Debugging System* chapter of the *TMS320C5xx 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 emulator 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

	switch #		Add this line to the
Address Range	**1**	**2**	**batch file**
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 'C54x 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 will encounter an error message when you try to invoke the debugger:

```
CANNOT INITIALIZE THE EVM !!
– Check I/O Configuration
```

When you use the EVM, you must first load a valid object file into the EVM. To do this, invoke the debugger and load the object file:

evm5xx [*object filename*]

Once you have entered the debugging environment and loaded the object file, exit the debugger:

`quit` ↵

Resetting the EVM

At the DOS prompt, reset the EVM by entering the evmrst command:

```
evmrst ↵
```

If you decide to change your I/O switch settings, you can specify a different I/O space by entering the –p option following evmrst. You can identify the I/O space by modifying the D_OPTIONS environment variable in your autoexec.bat or initdb.bat file and invoking evmrst. You can override D_OPTIONS by entering evmrst followed by the –x option.

Notes:

1) Never reset the ’C54x EVM with evmrst unless you have first loaded a valid object file to the EVM.

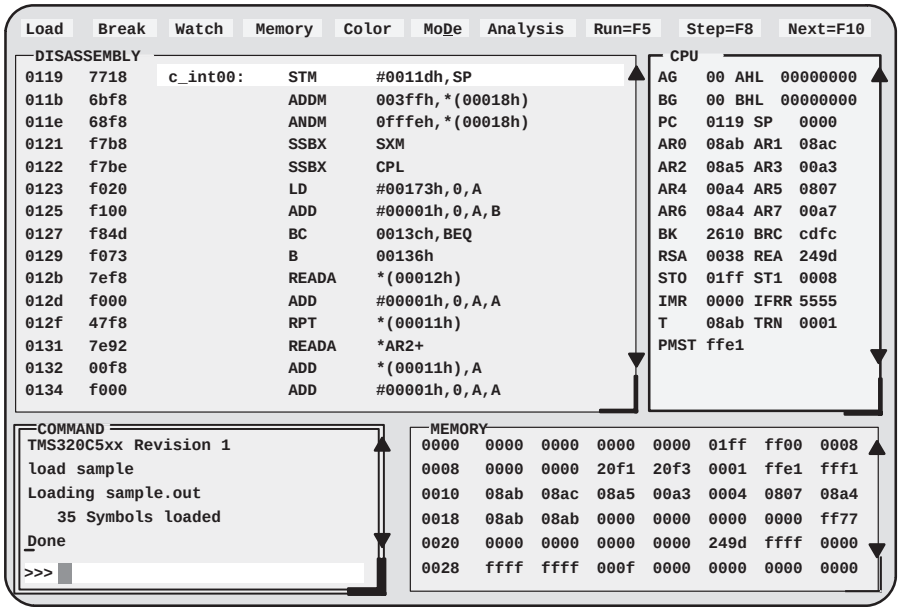
2) If you plan to use the debugger with the EVM, you don’t need to reset the EVM with evmrst before invoking the debugger.

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:

evm5xx c:\c5xxhll\sample ↵

You should see a display similar to this one:



- ❑ 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 *Invalid address*, your EVM board may not be installed snugly. Check your board to see if it is seated correctly and reenter the command above.
- ❑ If you do not 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 reenter the command above.

Installation error messages

While invoking the debugger, you may encounter the following error message.

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

One of several of the following conditions may be the cause; check:

- ❑ Is the EVM board installed snugly?
- ❑ Is your port address 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 Table 2 on page 5, and Table 3 on page 11).
  - ■ 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 3. Modify the –p option of the D_OPTIONS environment variable to reflect the change in your switch settings.

After you have checked all of the above, repeat the verification instructions in Section 5.

6. Using the Debugger With MS-Windows

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 user's manual for details.

In the MS-Windows environment, 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.