

# ***TMS470R1x Emulator Software***

## *Getting Started Guide*



*Book Type*  
*Two Lines*  
*Volume #*

*Book Type*  
*Volume #*

*Book Type*  
*Two Lines*

**Title**  
**Two Lines**  
**Subtitle**  
**Line Two**

**Title**  
**Two Lines**  
**Subtitle**

**Title**  
**Two Lines**

**Title**  
**Subtitle**  
**Line Two**

**Title**  
**Subtitle**

*year*



*Book Type*

**Title**

*year*

# ***TMS470R1x Emulator Software Getting Started Guide***

Literature Number: SPNU123B  
Manufacturing Part Number: M414009-9741 revision A  
October 1996



## **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.

## Preface

# Read This First

---

---

---

---

### ***About This Manual***

The book tells you how to install release 1.xx of the TMS470R1x emulator debugging tools on your system. It also does the following:

- Tells you how to set environment variables for parameters that you use often
- Gives helpful information for troubleshooting
- Gives information on new or changed features for this release of the emulator

### ***How to Use This Manual***

The goal of this book is to get you started using the emulator specifically designed for the TMS470R1x. Following are the topics covered in this getting started guide:

<b>For information about ...</b>	<b>See ...</b>
Installing the emulator, setting environment variables, and verifying the installation on Windows™ 3.1 systems	Chapter 1
Installing the emulator, setting environment variables, and verifying the installation on Windows NT™ systems	Chapter 2
Installing the emulator, setting environment variables, and verifying the installation on Windows™ 95 systems	Chapter 3
Installing the emulator, setting environment variables, and verifying the installation on a SPARCstation™	Chapter 4
Installing the emulator, setting environment variables, and verifying the installation on HP systems	Chapter 5
Workstation Troubleshooting	Appendix A
Installing the Emulator Device Driver Manually	Appendix B

## Notational Conventions

This document uses the following conventions.

- The TMS470R1x family of processors is referred to as '470.
- Program listings, program examples, and interactive displays are shown in a special *typeface* similar to a typewriter's. Examples use a **bold version** of the special typeface for emphasis; interactive displays use a **bold version** of the special typeface to distinguish commands that you enter from items that the system displays (such as prompts, command output, error messages, etc.).

Here is an example of a command that you might enter:

```
cd /cdrom/hp
```

- In syntax descriptions, the instruction, command, or directive is in a **bold typeface** font and parameters are in an *italic typeface*. Portions of a syntax that are in **bold** should be entered as shown; portions of a syntax that are in *italics* describe the type of information that should be entered. Here is an example of a command syntax:

```
PATH=C:\pathname1;pathname2
```

PATH is the command. This command has one parameter, indicated by *pathname*.

- Square brackets ( [ and ] ) identify an optional parameter. If you use an optional parameter, you specify the information within the brackets; you don't enter the brackets themselves. Here's an example of a command that has an optional parameter:

```
emurst [options]
```

This command allows you to specify one or more options.

### **Related Documentation From Texas Instruments**

The following books describe the TMS470R1x and related support tools. To obtain a copy of any of these TI documents, call the Texas Instruments Literature Response Center at (800) 477-8924. When ordering, please identify the book by its title and literature number.

**TMS470R1x Assembly Language Tools User's Guide** (literature number SPNU118) describes the assembly language tools (assembler, linker, and other tools used to develop assembly language code), assembler directives, macros, common object file format, and symbolic debugging directives for the TMS470R1x devices.

**TMS470R1x C Source Debugger User's Guide** (literature number SPNU124) describes the TMS470R1x emulator and simulator versions of the C source debugger interface. This book discusses various aspects of the debugger interface, including window management, command entry, code execution, data management, and breakpoints. It also includes a tutorial that introduces basic debugger functionality.

**TMS470R1x Optimizing C Compiler User's Guide** (literature number SPNU119) describes the TMS470R1x C compiler. This C compiler accepts ANSI standard C source code and produces assembly language source code for the TMS470R1x devices.

**TMS470R1x User's Guide** (literature number SPNU134) gives a detailed description of the TMS470R1x RISC microcontroller: its architecture including registers, ICEBreaker module, and interfaces (memory, coprocessor, and debug), its instruction sets (both 16- and 32-bit), and electrical specifications.

**XDS51x Emulator Installation Guide** (literature number SPNU070) describes the installation of the XDS510™, XDS510PP™, and XDS510WS™ emulator controllers. The installation of the XDS511™ emulator is also described.

### **FCC 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**

HP-UX, HP 9000 Series 700, and PA-RISC are trademarks of Hewlett-Packard Company.

Microsoft, MS-DOS, Windows, and Windows NT are registered trademarks of Microsoft Corporation.

OpenWindows, Solaris, and SunOS are trademarks of Sun Microsystems, Inc.

PC and PC-DOS are trademarks of International Business Machines Corporation.

Pentium is a trademark of Intel Corporation.

SPARCstation is trademark of SPARC International, Inc., but licensed exclusively to Sun Microsystems, Inc.

UNIX is a registered trademark in the United States and other countries, licensed exclusively through X/Open Company Limited.

X Window System is a trademark of the Massachusetts Institute of Technology.

XDS510, XDS510PP, and XDS510WS are trademarks of Texas Instruments Incorporated.

**If You Need Assistance. . .**

<input type="checkbox"/> <b>World-Wide Web Sites</b> TI Online <span style="float: right;">http://www.ti.com</span> Semiconductor Product Information Center (PIC) <span style="float: right;">http://www.ti.com/sc/docs/pic/home.htm</span> Microcontroller Home Page <span style="float: right;">http://www.ti.com/sc/micro</span>		
<input type="checkbox"/> <b>North America, South America, Central America</b> Product Information Center (PIC) <span style="float: right;">(972) 644-5580</span> TI Literature Response Center U.S.A. <span style="float: right;">(800) 477-8924</span> Software Registration/Upgrades <span style="float: right;">(214) 638-0333</span> <span style="float: right;">Fax: (214) 638-7742</span> U.S.A. Factory Repair/Hardware Upgrades <span style="float: right;">(713) 274-2285</span> U.S. Technical Training Organization <span style="float: right;">(972) 644-5580</span> Microcontroller Hotline <span style="float: right;">(713) 274-2370</span> <span style="float: right;">Fax: (713) 274-4203</span> <span style="float: right;">Email:*H370@msg.ti.com</span> Microcontroller Modem BBS <span style="float: right;">(713) 274-3700 8-N-1</span>		
<input type="checkbox"/> <b>Europe, Middle East, Africa</b> European Product Information Center (EPIC) Hotlines: Multi-Language Support <span style="float: right;">+33 1 30 70 11 69</span> <span style="float: right;">Fax: +33 1 30 70 10 32</span> <span style="float: right;">Email: epic@ti.com</span> Deutsch <span style="float: right;">+49 8161 80 33 11</span> or <span style="float: right;">+33 1 30 70 11 68</span> English <span style="float: right;">+33 1 30 70 11 65</span> Francais <span style="float: right;">+33 1 30 70 11 64</span> Italiano <span style="float: right;">+33 1 30 70 11 67</span> EPIC Modem BBS <span style="float: right;">+33 1 30 70 11 99</span> European Factory Repair <span style="float: right;">+33 1 93 22 25 40</span> Europe Customer Training Helpline <span style="float: right;">Fax: +49 81 61 80 40 10</span>		
<input type="checkbox"/> <b>Asia-Pacific</b> Literature Response Center <span style="float: right;">+852 2 956 7288</span> <span style="float: right;">Fax: +852 2 956 2200</span>		
<input type="checkbox"/> <b>Japan</b> Product Information Center <span style="float: right;">+0120-81-0026 (in Japan)</span> <span style="float: right;">Fax: +0120-81-0036 (in Japan)</span> <span style="float: right;">+03-3457-0972 or (INTL) 813-3457-0972</span> <span style="float: right;">Fax: +03-3457-1259 or (INTL) 813-3457-1259</span>		
<input type="checkbox"/> <b>Documentation</b> When making suggestions or reporting errors in documentation, please include the following information that is on the title page: the full title of the book, the publication date, and the literature number. Mail: Texas Instruments Incorporated <span style="float: right;">Email: comments@books.sc.ti.com</span> Technical Documentation Services, MS 702 P.O. Box 1443 Houston, Texas 77251-1443		

**Note:** When calling a Literature Response Center to order documentation, please specify the literature number of the book.



# Contents

---

---

---

<b>1</b>	<b>Installing the Debugger With Windows 3.1x</b> .....	<b>1-1</b>
	<i>Lists the hardware and software you need to install the emulator board and C source debugger; provides installation instructions for PC systems running Windows 3.1x.</i>	
1.1	System Requirements .....	1-2
	Hardware checklist .....	1-2
	Software checklist .....	1-3
1.2	Step 1: Installing the XDS510 or XDS510PP Emulator Controller .....	1-4
1.3	Step 2: Installing the Debugger Software .....	1-4
1.4	Step 3: Setting Up a DLL File for Your XDS510 or XDS510PP .....	1-5
1.5	Additional XDS510PP Setup .....	1-6
	Diagnostic files .....	1-7
1.6	Step 4: Setting Up the Debugger Environment .....	1-8
	Identifying the directory that contains the executable files (PATH statement) .....	1-8
	Identifying alternate directories for the debugger (D_DIR) .....	1-8
	Identifying directories that contain source files (D_SRC) .....	1-9
	Setting default debugger options (D_OPTIONS) .....	1-9
1.7	Step 5: Resetting the Emulator .....	1-11
	Special Considerations When Resetting the XDS510PP .....	1-12
	The smcmode command .....	1-12
1.8	Step 6: Describing Your Target System to the Debugger .....	1-13
1.9	Step 7: Verifying the Installation .....	1-14
1.10	Installation Error Messages .....	1-15
<b>2</b>	<b>Installing the Debugger With Windows NT</b> .....	<b>2-1</b>
	<i>Lists the hardware and software you need to install the emulator version of the C source debugger; provides installation instructions for PC systems running Windows NT.</i>	
2.1	System Requirements .....	2-2
	Hardware checklist .....	2-2
	Software checklist .....	2-3
2.2	Step 1: Installing the XDS510 Emulator Controller .....	2-4
2.3	Step 2: Installing the Debugger Software .....	2-4
2.4	Step 3: Installing the Emulator Device Driver .....	2-5
2.5	Step 4: Verifying the Emulator Device Driver Installation .....	2-7
	Driver installation error messages .....	2-7

2.6	Step 5: Setting Up the Debugger Environment .....	2-9
	Identifying the directory that contains the executable files (PATH statement) .....	2-9
	Identifying alternate directories for the debugger (D_DIR) .....	2-10
	Identifying directories that contain source files (D_SRC) .....	2-10
	Setting default debugger options (D_OPTIONS) .....	2-10
2.7	Step 6: Resetting the Emulator .....	2-12
2.8	Step 7: Describing Your Target System to the Debugger .....	2-13
2.9	Step 8: Verifying the Installation .....	2-14
2.10	Installation Error Messages .....	2-15
<b>3</b>	<b>Installing the Debugger With Windows 95 .....</b>	<b>3-1</b>
	<i>Lists the hardware and software you need to install the emulator version of the C source debugger; provides installation instructions for PC systems running Windows 95.</i>	
3.1	System Requirements .....	3-2
	Hardware checklist .....	3-2
	Software checklist .....	3-3
3.2	Step 1: Installing the XDS510 or XDS510PP Emulator Controller .....	3-4
3.3	Step 2: Installing the Debugger Software .....	3-4
3.4	Step 3: Setting Up a DLL File for Your XDS510 or XDS510PP .....	3-5
3.5	Additional XDS510PP Setup .....	3-6
	Diagnostic files .....	3-7
3.6	Step 4: Setting Up the Debugger Environment .....	3-8
	Identifying the directory that contains the executable files (PATH statement) .....	3-8
	Identifying alternate directories for the debugger (D_DIR) .....	3-8
	Identifying directories that contain source files (D_SRC) .....	3-9
	Setting default debugger options (D_OPTIONS) .....	3-9
3.7	Step 5: Resetting the Emulator .....	3-11
	Special Considerations When Resetting the XDS510PP .....	3-12
	The smcmode command .....	3-12
3.8	Step 6: Describing Your Target System to the Debugger .....	3-13
3.9	Step 7: Verifying the Installation .....	3-14
3.10	Installation Error Messages .....	3-15

<b>4</b>	<b>Installing the Debugger on a SPARCstation .....</b>	<b>4-1</b>
	<i>Lists the hardware and software you need to install the workstation emulator and C source debugger; provides installation instructions for SPARCstations running SunOS.</i>	
4.1	System Requirements .....	4-2
	Hardware checklist .....	4-2
	Software checklist .....	4-2
4.2	Step 1: Installing the XDS510WS Emulator Controller .....	4-4
4.3	Step 2: Installing the Debugger Software .....	4-4
	Mounting the CD-ROM .....	4-4
	Copying the files .....	4-5
	Unmounting the CD-ROM .....	4-5
4.4	Step 3: Ensuring That the Emulator Supports the Debugger .....	4-6
4.5	Step 4: Describing Your Target System to the Debugger .....	4-7
4.6	Step 5: Setting Up the Debugger Environment .....	4-8
	Modifying the path shell variable .....	4-8
	Setting up the environment variables .....	4-8
	Invoking the new or modified .cshrc file .....	4-10
4.7	Step 6: Verifying the Installation .....	4-11
4.8	Using the Debugger With the X Window System .....	4-12
	Using the keyboard's special keys .....	4-12
	Changing the debugger font .....	4-13
	Color mappings on monochrome screens .....	4-13
<b>5</b>	<b>Installing the Debugger on an HP Workstation .....</b>	<b>5-1</b>
	<i>Lists the hardware and software you need to install the workstation emulator and C source debugger; provides installation instructions for HP workstations running HP-UX.</i>	
5.1	System Requirements .....	5-2
	Hardware checklist .....	5-2
	Software checklist .....	5-3
5.2	Step 1: Installing the XDS510WS Emulator Controller .....	5-4
5.3	Step 2: Installing the Debugger Software .....	5-4
	Mounting the CD-ROM .....	5-4
	Copying the files .....	5-5
	Unmounting the CD-ROM .....	5-5
5.4	Step 3: Ensuring That the Emulator Supports the Debugger .....	5-5
5.5	Step 4: Describing Your Target System to the Debugger .....	5-6
5.6	Step 5: Setting Up the Debugger Environment .....	5-7
	Modifying the path shell variable .....	5-7
	Setting up the environment variables .....	5-7
	Invoking the new or modified .cshrc file .....	5-9

5.7	Step 6: Verifying the Installation .....	5-10
5.8	Using the Debugger With the X Window System .....	5-11
	Using the keyboard's special keys .....	5-11
	Changing the debugger font .....	5-12
	Color mappings on monochrome screens .....	5-12
<b>A</b>	<b>Workstation Troubleshooting .....</b>	<b>A-1</b>
	<i>Describes problems that you may encounter while installing and using the emulator on your SPARCstation or HP Workstation and gives suggestions for resolving those problems.</i>	
A.1	Problems When Booting Your Workstation .....	A-2
A.2	Problems With Multiple Emulators on SunOS .....	A-2
A.3	Problems When Resetting the Emulator .....	A-3
A.4	Problems When Invoking the Debugger .....	A-5
A.5	Additional Emulator and Debugger Problems .....	A-7
<b>B</b>	<b>Installing the Emulator Device Driver Manually .....</b>	<b>B-1</b>
	<i>Provides instructions for installing the emulator device driver manually.</i>	
B.1	Copying the Device Driver File and Invoking the Registry Editor .....	B-2
B.2	Setting Up the Genport Directory .....	B-3
B.3	Setting Up the Values for the Genport Directory .....	B-4
B.4	Setting Up the Parameters Directory and Values .....	B-6

# Installing the Debugger With Windows 3.1x

---

---

---

---

This chapter provides instructions for installing the C source debugger on a PC™ running Windows 3.1x. When you complete the installation, turn to the *TMS470R1x C Source Debugger User's Guide*.

To install the emulator controller, see the *XDS51x Emulator Installation Guide*, which is included in your kit.

<b>Topic</b>	<b>Page</b>
<b>1.1 System Requirements</b> .....	<b>1-2</b>
<b>1.2 Step 1: Installing the XDS510 or XDS510PP Emulator Controller</b> ..	<b>1-4</b>
<b>1.3 Step 2: Installing the Debugger Software</b> .....	<b>1-4</b>
<b>1.4 Step 3: Setting Up a DLL File for Your XDS510 or XDS510PP</b> .....	<b>1-5</b>
<b>1.5 Additional XDS510PP Setup</b> .....	<b>1-6</b>
<b>1.6 Step 4: Setting Up the Debugger Environment</b> .....	<b>1-8</b>
<b>1.7 Step 5: Resetting the Emulator</b> .....	<b>1-11</b>
<b>1.8 Step 6: Describing Your Target System to the Debugger</b> .....	<b>1-13</b>
<b>1.9 Step 7: Verifying the Installation</b> .....	<b>1-14</b>
<b>1.10 Installation Error Messages</b> .....	<b>1-15</b>

## 1.1 System Requirements

To install the '470 C source debugger and TMS470R1x emulator, you need the items in the following hardware and software checklists.

### **Hardware checklist**

- |                          |                                   |  |
|--------------------------|-----------------------------------|--|
| <input type="checkbox"/> | <b>System</b>                     | 80386-, 80486, or Pentium™-based PC                          |
| <input type="checkbox"/> | <b>Memory</b>                     | 4–16 Mbytes of free memory                                   |
| <input type="checkbox"/> | <b>Disk space</b>                 | 10 Mbytes available disk space for executables and libraries |
| <input type="checkbox"/> | <b>Display</b>                    | Color monitor  |
| <input type="checkbox"/> | <b>Required hardware</b>          | CD-ROM drive   |
| <input type="checkbox"/> | <b>Optional hardware</b>          | Microsoft™-compatible mouse                                  |
| <input type="checkbox"/> | <b>XDS510 emulator controller</b> | XDS510™ or XDS510PP™ emulator controller                     |

**Note:**

The speed at which your system operates depends on the amount of RAM available on your PC and the number of debuggers running simultaneously.

**Software checklist**

- |                          |  |  |
|--------------------------|--|--|
| <input type="checkbox"/> | <b>Operating system</b>                                  | Windows 3.1x   |
| <input type="checkbox"/> | <b>Software tools</b>                                    | TMS470R1x assembler and linker<br>Optional: TMS470R1x C compiler   |
| <input type="checkbox"/> | <b>Required files included with the debugger package</b> | <i>emu470.exe</i> is the debugger executable file  |
| <input type="checkbox"/> |  | <i>emurst.exe</i> resets the XDS510 emulator   |
| <input type="checkbox"/> |  | <i>emurstpp.exe</i> resets the XDS510PP emulator   |
| <input type="checkbox"/> |  | <i>board.dat</i> describes your target system to the debugger in terms of what devices are on the emulation scan path. The board.dat file included in the debugger package is for a target board with one TMS470R1x named <i>tms470</i> .  |
| <input type="checkbox"/> |  | <i>board.cfg</i> is a text file used to describe your target system to the debugger in terms of what devices are on the emulation scan path. The board.cfg file included in the debugger package is for a target board with one TMS470R1x named <i>tms470</i> .  |
| <input type="checkbox"/> |  | The <i>composer</i> utility allows you to convert your text board configuration file (board.cfg) into a format the debugger can read (board.dat). For the emulator to initialize properly, you must create a new board.dat file with this release of the composer or use the samples that are provided for you.  |
| <input type="checkbox"/> | <b>Optional files included with the debugger package</b> | <i>emuinit.cmd</i> is a general-purpose batch file that contains debugger commands. The version of this file that is shipped with the debugger defines a '470 memory map. 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, see the <i>TMS470R1x C Source Debugger User's Guide</i> . |

## 1.2 Step 1: Installing the XDS510 or XDS510PP Emulator Controller

Before installing the '470 debugger software, you must install an XDS510 emulator controller: the XDS510 or XDS510PP.

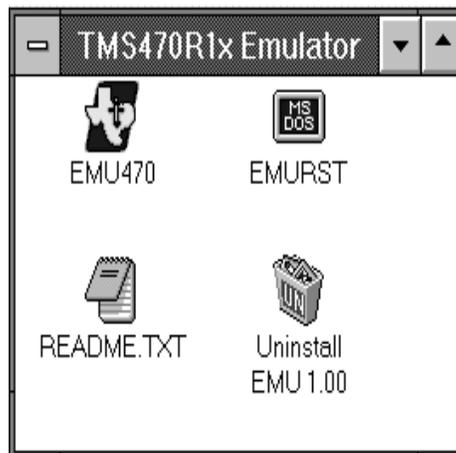
Follow the instructions in the *XDS51x Emulator Installation Guide* to install the XDS510 emulator controller.

## 1.3 Step 2: Installing the Debugger Software

To install the debugger on a Windows 3.1x system, follow these steps:

- 1) Insert the debugger CD-ROM into your CD-ROM drive.
- 2) Start Windows 3.1x.
- 3) From the File menu, select Run.
- 4) In the dialog box, enter the following command (replace d with the letter of your CD-ROM drive):  
`d:\setup.exe`
- 5) Click on OK.
- 6) Follow the on-screen instructions.

The setup.exe program creates a program group for the TMS470R1x Emulator that looks similar to the following example:



## **1.4 Step 3: Setting Up a DLL File for Your XDS510 or XDS510PP**

After you run the installation program (setup.exe), you must set up a DLL (dynamic link library) file for your XDS510 or XDS510PP. You can set up the smg510w.dll file for your XDS510 or XDS510PP by running a batch file, or you can do it manually.

For the XDS510:

■ Run set510.bat. This file sets up the smg510w.dll file.

or

■ Copy the smg510.xds file to the smg510w.dll file.

For the XDS510PP:

■ Run set510pp.bat. This file sets up the smg510w.dll file.

or

■ Copy the smg510.pp file to the smg510w.dll file.

## 1.5 Additional XDS510PP Setup

If you are using the XDS510PP, you need to complete the following:

- Check your BIOS (basic input/output system) to ensure that your printer is in one of the following modes:

EPP: Enhanced printer port

SPP4: 4-bit unidirectional mode

SPP8: 8-bit bidirectional mode

EPP mode provides the best performance. If your BIOS offers the options *EPP+ECP* and *EPP+SPP*, select *EPP+SPP*. The *EPP+ECP* option is not supported. Note the port address of your printer.

- Edit the parameters in the xds510pp.ini file to match your system. The xds510pp.ini file contains parameters for the printer address, printer mode, and printer speed. Make sure the xds510pp.ini file is in the same directory as your debugger software. The following is a sample xds510pp.ini file:

```
port = 378
mode = EPP
speed = 10
```

Where:

- Port is the I/O address of your printer port.
- Mode is the printer port mode (EPP, SPP4, or SPP8).
- Speed is a delay parameter that may be required on some systems. Set this parameter to 10 initially. You can reduce this number once your XDS510PP is up and running.

## Diagnostic files

The XDS510PP software includes diagnostic files that help you determine system information (see Table 1–1). Use these files if you have problems setting up your parallel printer port.

Table 1–1. Diagnostic Files

File	Purpose
smcmode.exe	Determines if your parallel printer port uses a port controller device manufactured by SMC.
-h	gives you more information about the options.
-r	resets the status register.
-m x	sets your printer port to mode x where: Mode 0: Standard bidirectional mode. Mode 1: EPP and SPP mode. Mode 2: ECP mode. Mode 3: ECP and EPP mode. Mode 4: Non bidirectional mode.
nscmode.exe	Determines if your parallel printer port uses a port controller device manufactured by National Semiconductor.
portchk.exe	Determines if your parallel printer port is bidirectional.

For more information about the smcmode.exe program, see Section 1.7, *Special Considerations When Resetting the XDS510PP*.

## 1.6 Step 4: Setting Up the Debugger Environment

You can define *environment variables* that set certain debugger parameters you normally use. An environment variable is a system symbol that you define and assign to a string. When you use environment variables, default values are set, making each individual invocation of the debugger simpler because these parameters are automatically specified. The debugger uses three environment variables: `D_DIR`, `D_SRC`, and `D_OPTIONS`.

By default, the installation program sets up these environment variables in your `autoexec.bat` file:

```
SET PATH=C:\EMU470;%PATH%
SET D_DIR=C:\EMU470;
```

If you choose not to have the environment variables set up automatically, you can modify your `autoexec.bat` file to exclude the `SET` commands above.

The remainder of this section describes these environment variables and other variables that you can define.

### ***Identifying the directory that contains the executable files (PATH statement)***

You must include the emulator directory in your `PATH` statement. This allows you to specify the debugger executable without specifying the name of the directory that contains the executable file.

- If you modify your `autoexec.bat` file to change the path information, add the following to the end of the `PATH` statement:

```
;C:\EMU470
```

- If you set the `PATH` statement from the command line, use this format:

```
SET PATH=C:\EMU470;%PATH%
```

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

The addition of `;%PATH%` ensures that this `PATH` statement does not undo the `PATH` statements in any other batch files (including the `autoexec.bat` file).

### ***Identifying alternate directories for the debugger (D\_DIR)***

The debugger uses the `D_DIR` environment variable to name alternative directories that contain auxiliary files (`emurst`, `emuinit.cmd`, etc.) that the debugger needs. The command for assigning the environment variable is as follows:

```
SET D_DIR=C:\EMU470
```

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

### ***Identifying directories that contain source files (D\_SRC)***

The debugger uses the D\_SRC environment variable to find directories that contain program source files. The command for assigning the environment variable is as follows:

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

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

The *pathnames* are directories that contain program source files. You can separate pathnames with a semicolon or with blanks.

### ***Setting default debugger options (D\_OPTIONS)***

You might find it useful to set default debugger options using the D\_OPTIONS environment variable. When you use the D\_OPTIONS environment variable, the debugger uses the options and/or input filenames that you name with D\_OPTIONS every time you run the debugger. The command for assigning the environment variable is as follows:

```
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. The options that you can identify with D\_OPTIONS are listed on page 1-10.

These are the options you can identify with D\_OPTIONS:

Option	Brief Description
-b[b]	Select the screen size.
-bl	Select screen length.
-bw	Select screen width.
-c	Clear the .bss section.
-f <i>filename</i>	Identify a new board configuration file.
-font <i>size</i>	Select font point size.
-i <i>pathname</i>	Identify additional directories.
-me	Select little-endian format.
-min	Select the minimal debugging mode.
-n <i>processor name</i>	Identify the processor for debugging (this processor name must match the processor name as stated in the configuration file).
-p <i>port address</i>	Identify the port address.
-profile	Enter the profiling environment.
-s	Load the symbol table only.
-t <i>filename</i>	Identify a new initialization file.
-v	Load without the symbol table.

**Note: Overriding D\_OPTIONS**

You can override D\_OPTIONS by invoking the debugger or emurst with the -x option.

For more information about options, see the invocation instructions in the *TMS470R1x C Source Debugger User's Guide*.

## 1.7 Step 5: Resetting the Emulator

You must reset your XDS510 or XDS510PP *before* invoking the debugger. A successful reset can occur only after you have powered up the target board. You can reset the emulator by adding one of the following commands to the autoexec.bat file:

XDS510: **emurst** [-x] [-p *port address*]

XDS510PP: **emurstpp** [-x]

The -x option tells the emurst/emurstpp utility to ignore any options specified with the D\_OPTIONS environment variable.

The -p option identifies the port address.

When you want to reset your XDS510 or XDS510PP without rebooting your system, use the icons provided in the program group.

- To reset the XDS510 from Windows, click on the EMURST icon in the TMS470R1x Emulator program group.



EMURST

- To reset the XDS510PP from Windows, click on the EMURSTPP icon in the TMS470R1x Emulator program group.



EMURSTPP

### Note:

If the debugger is running, emurst/emurstpp will not reset the emulator. The debugger displays the following message:

```
RESET DISALLOWED : DEBUGGER RUNNING
```

If an error message appears after the emulator is reset, see Section 1.10, *Installation Error Messages*.

### **Special Considerations When Resetting the XDS510PP**

You must reset your XDS510PP with the `emurstpp` command *before* invoking the debugger. After resetting the XDS510PP with the `emurstpp` command, you will see one of two messages.

- If you see the following message, the reset was successful:

```
EMURST FOR THE XDS510PP VERSION 1.0  
XDS510PP IS RESET, HARDWARE VERSION 1
```

- If you see the following message, the reset was unsuccessful:

```
EMURST FOR THE XDS510PP VERSION 1.0  
COMMUNICATIONS ERROR, OR POD HAS NO POWER
```

### **The `smcmode` command**

If `emurstpp` does not reset your XDS510PP, it is possible that the printer port device in your computer and the SMC chip in the XDS510PP are confused during power cycles. Enter the following command:

```
smcmode -r 
```

If your computer has an SMC device, this command resets and initializes the device for proper operation.

The `smcmode` command works only if your computer uses an SMC printer port controller chip. To determine if your computer uses an SMC printer port controller chip, enter the following command with no parameters:

```
smcmode 
```

This command tells you if an SMC device is installed and lists the device's configuration.

## 1.8 Step 6: Describing Your Target System to the Debugger

In order for the debugger to understand how you have configured your target system, you must supply a file for the debugger to read. You can either use the default configuration file, *board.dat*, or create your own file.

- If you are using an emulation scan path that contains only one '470 and no other devices, you can use the *board.dat* file that comes with the '470 emulator kit. This file describes the single '470 in the scan path and gives the '470 the name *tms470*. Since the debugger automatically looks for a file called *board.dat* in the current directory and in the directories specified with the *D\_DIR* environment variable, you don't need to create your own board configuration file.
- If you want to use a different name for the target device or you want the debugger to recognize a different target configuration, you must follow these steps:
  - 1) Create the board configuration file.
  - 2) Use *composer* to translate the board configuration file to binary so that the debugger can read it.
  - 3) Specify the target processor name as stated in the configuration file. This name is used with the *-n* option when invoking the debugger.

These steps are described in the *TMS470R1x C Source Debugger User's Guide*.

## 1.9 Step 7: Verifying the Installation

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

```
emu470 -p port address -n tms470
```

You should see a display similar to this one:

Load	Break	Watch	Memory	Color	MoDe	Analysis	Run=F5	Step=F8	Next=F10
DISASSEMBLY									
00000000	55555555					LDRPLB R5 [R5, #-		PC	aaaaaaaa
00000004	55555555					LDRPLB R5 [R5, #-		SP	aaaaaaaa
00000003	55555555					LDRPLB R5 [R5, #-		LR	aaaaaaaa
0000000c	55555555					LDRPLB R5 [R5, #-		CPSR	aaaaaaaa
00000010	55555555					LDRPLB R5 [R5, #-		R0	aaaaaaaa
00000014	55555555					LDRPLB R5 [R5, #-		R1	aaaaaaaa
00000013	55555555					LDRPLB R5 [R5, #-		R2	aaaaaaaa
0000001c	55555555					LDRPLB R5 [R5, #-		R3	aaaaaaaa
00000020	55555555					LDRPLB R5 [R5, #-		R4	aaaaaaaa
00000024	55555555					LDRPLB R5 [R5, #-		R5	aaaaaaaa
00000023	55555555					LDRPLB R5 [R5, #-		R6	aaaaaaaa
0000002c	55555555					LDRPLB R5 [R5, #-		R7	aaaaaaaa
00000030	55555555					LDRPLB R5 [R5, #-		R8	aaaaaaaa
00000034	55555555					LDRPLB R5 [R5, #-		R9	aaaaaaaa
00000033	55555555					LDRPLB R5 [R5, #-		R10	aaaaaaaa

COMMAND		WATCH STATUS REGISTER	
ARM7TDMI	Silicone Revision 0.1.7	3: C-bit	1
XDS510	Emulator Revision 1	4: V-bit	0
		5: I-bit	1
		6: F-bit	0
		7: T-bit	1
		8: MODE	0x0000000a

- If you see a display similar to this one, you have correctly installed your emulator and debugger.
- If you see a display and the lines of code say *Invalid address* or the fields in the MEMORY window are shown in red, the debugger may not be able to find the emuinit.cmd file. Check for the file in the directories specified by the D\_DIR environment variable or ensure that the file is in the current directory. Reenter the command above.
- If you don't see a display, 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.

## 1.10 Installation Error Messages

While invoking the debugger, you may see one of the following messages:

```
CANNOT INITIALIZE THE TARGET SYSTEM ! !  
- Check I/O configuration  
- Check cabling and target power
```

```
CANNOT DETECT TARGET POWER ! !  
- Check I/O configuration  
- Check cabling and target power
```

One or several of the following conditions may be the cause:

- Is the target power on?
- Is the XDS510/XDS510PP installed snugly?
- Is the device installed snugly?
- Is the cable connecting your emulator and target system loose?
- Is your target board getting the correct voltage?
- Is your emulator scan path interrupted? One or more devices on the emulator scan path may have been removed. Check the connections; either they are not connected, or they are connected improperly.
- Did you use the `-n` option? Was it used with an incorrect device name? You must supply a valid device name with the `-n` option.
- After you powered up the target board, did you execute the `emurst/ emurstpp` command? This command must be executed *after* you powered up the target board.

- Did you use the `-p` option? Is your port address correct (XDS510 only)?
  - Check to be sure the `-p` option used with the `D_OPTIONS` environment variable matches the I/O address defined by your switch settings. For information about the switch settings, see the XDS510 installation instructions in the *XDS51x Emulator Installation Guide*.
  - 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 alternative settings. Modify the `-p` option of the `D_OPTIONS` environment variable to reflect the change in your switch settings.
- Is the `board.dat` file in the current directory or in a directory specified by `D_DIR`?

After you have checked all of the above, repeat the verification instructions on page 1-14.

# Installing the Debugger With Windows NT

---

---

---

---

This chapter provides instructions for installing the C source debugger on a PC running Windows NT™. When you complete the installation, turn to the *TMS470R1x C Source Debugger User's Guide*.

To install the emulator controller, see the *XDS51x Emulator Installation Guide*, which is included in your kit.

<b>Topic</b>	<b>Page</b>
<b>2.1 System Requirements</b> .....	<b>2-2</b>
<b>2.2 Step 1: Installing the XDS510 Emulator Controller</b> .....	<b>2-4</b>
<b>2.3 Step 2: Installing the Debugger Software</b> .....	<b>2-4</b>
<b>2.4 Step 3: Installing the Emulator Device Driver</b> .....	<b>2-5</b>
<b>2.5 Step 4: Verifying the Emulator Device Driver Installation</b> .....	<b>2-7</b>
<b>2.6 Step 5: Setting Up the Debugger Environment</b> .....	<b>2-9</b>
<b>2.7 Step 6: Resetting the Emulator</b> .....	<b>2-12</b>
<b>2.8 Step 7: Describing Your Target System to the Debugger</b> .....	<b>2-13</b>
<b>2.9 Step 8: Verifying the Installation</b> .....	<b>2-14</b>
<b>2.10 Installation Error Messages</b> .....	<b>2-15</b>

## 2.1 System Requirements

To install the '470 C source debugger and TMS470R1x emulator, you need the items in the following hardware and software checklists.

### ***Hardware checklist***

- |                          |                                   |   |
|--------------------------|-----------------------------------|---|
| <input type="checkbox"/> | <b>System</b>                     | 32-bit x86- or Pentium-based PC with an ISA/EYES, VESA local, or PCI bus. |
| <input type="checkbox"/> | <b>Memory</b>                     | 4–16 Mbytes of free memory  |
| <input type="checkbox"/> | <b>Disk space</b>                 | 10 Mbytes available disk space for executables and libraries              |
| <input type="checkbox"/> | <b>Display</b>                    | Color monitor   |
| <input type="checkbox"/> | <b>Required hardware</b>          | CD-ROM drive  |
| <input type="checkbox"/> | <b>Optional hardware</b>          | Microsoft-compatible mouse  |
| <input type="checkbox"/> | <b>XDS510 emulator controller</b> | XDS510 emulator controller  |

**Software checklist**

- |                          |  |  |
|--------------------------|--|--|
| <input type="checkbox"/> | <b>Operating system</b>                                  | Windows NT version 3.5 or higher   |
| <input type="checkbox"/> | <b>Software tools</b>                                    | TMS470R1x assembler and linker<br>Optional: TMS470R1x C compiler   |
| <input type="checkbox"/> | <b>Required files included with the debugger package</b> | <i>emu470.exe</i> is the debugger executable file  |
| <input type="checkbox"/> |  | <i>emurst.exe</i> resets the XDS510 emulator   |
| <input type="checkbox"/> |  | <i>board.dat</i> describes your target system to the debugger in terms of what devices are on the emulation scan path. The <i>board.dat</i> file included in the debugger package is for a target board with one TMS470R1x named <i>tms470</i> .   |
| <input type="checkbox"/> |  | <i>board.cfg</i> is a text file used to describe your target system to the debugger in terms of what devices are on the emulation scan path. The <i>board.cfg</i> file included in the debugger package is for a target board with one TMS470R1x named <i>tms470</i> .   |
| <input type="checkbox"/> |  | The <i>composer</i> utility allows you to convert your text board configuration file ( <i>board.cfg</i> ) into a format the debugger can read ( <i>board.dat</i> ). For the emulator to initialize properly, you must create a new <i>board.dat</i> file with this release of the composer or use the samples provided for you.  |
| <input type="checkbox"/> | <b>Optional files included with the debugger package</b> | <i>emuinit.cmd</i> is a general-purpose batch file that contains debugger commands. The version of this file that is shipped with the debugger defines a '470 memory map. 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, see the <i>TMS470R1x C Source Debugger User's Guide</i> . |

## 2.2 Step 1: Installing the XDS510 Emulator Controller

Before installing the '470 debugger software, you must install the XDS510 emulator controller.

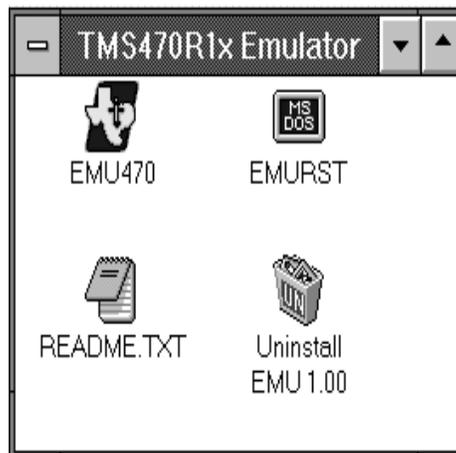
Follow the instructions in the *XDS51x Emulator Installation Guide* to install the XDS510 emulator controller.

## 2.3 Step 2: Installing the Debugger Software

To install the debugger on a Windows NT system, follow these steps:

- 1) Insert the debugger CD-ROM into your CD-ROM drive.
- 2) Start Windows NT.
- 3) From the File menu, select Run.
- 4) In the dialog box, enter the following command (replace d with the letter of your CD-ROM drive):  
`d:\setup.exe`
- 5) Click on OK.
- 6) Follow the on-screen instructions.

The setup.exe program creates a program group for the TMS470R1x Emulator that looks similar to the following example:



## 2.4 Step 3: Installing the Emulator Device Driver

You must install the emulator device driver, `genport.sys`, in the Windows NT registry to enable the debugger to communicate with the emulator. To install the emulator device driver, you can use the `regdrv.exe` driver installation program. The `regdrv.exe` program is stored in the `emu470\drivers` subdirectory of the emulation software and performs the following tasks:

- Copies `genport.sys` to the Windows NT drivers directory. In most cases, this directory is `C:\WINDOWS\SYSTEM32\DRIVERS` or `C:\WINNT35\SYSTEM32\DRIVERS`. However, the directory name may be different on your system.
- Registers the `genport.sys` parameters in the Windows NT registry.

---

**Note:**

The Windows NT registry (also referred to as the configuration registry) serves as a repository for all information about the computer hardware that the Windows NT operating system is running on, the software installed on the system, and the users of the system. Device drivers, applications, and users can also place information in the registry and can later query the registry for the information they need.

---

To install the the emulator device driver, `genport.sys`, follow these steps:

- 1) Set your current directory to the *driver* subdirectory where the emulator software is installed.

For example, enter a command like the following:

```
CD: \EMU470\DRIVER
```

- 2) Execute the driver installation program.

`regdrv.exe` 

The `regdrv.exe` program prompts you to enter the path to the directory where the `genport.sys` device driver is located. If you don't enter a path, the `regdrv.sys` program searches the current directory.

- 3) To accept the default parameters, press enter.

The regdrv.exe program prompts you to register the emulator driver parameters.

<Y/N>

- If you enter Y, the regdrv.exe program checks to see if the emulator driver is already installed. If it is, the regdrv.exe program prompts you to replace/update the driver with the new emulator driver.
- If you enter N, the regdrv.exe program exits without updating the emulator driver parameters in the Windows NT registry.

- 4) To complete the installation, enter the following:

Y 

When the registration is complete, the regdrv.exe program displays the status of the registration.

- 5) Reboot your PC.

If you get error messages during the installation process, see the *Driver Installation Error Messages* subsection on page 2-7.

For information about installing the emulator device driver, see Appendix B, *Installing the Emulator Device Driver Manually*.

## 2.5 Step 4: Verifying the Emulator Device Driver Installation

To ensure that you have correctly installed the emulator device driver, follow these steps:

- 1) Turn off the power to the TMS470 target board.
- 2) Connect the JTAG cable to the TMS470 target board.
- 3) Turn on the power to the TMS470 target board.

If a hard reset is not generated automatically after power up, apply a hard reset to the TMS470 device.

- 4) From the command prompt in an MS-DOS™ window, enter:

```
emurst 
```

If the port address has a value other than 240, you must use the `-p` option with the `emurst` command to specify the appropriate port address.

- If you see a message similar to the following, you have correctly installed the emulator device driver.

```
XDS510 IS RESET, HARDWARE VERSION 3
```

- If you see this message:

```
CANNOT DETECT TARGET POWER
```

follow the troubleshooting tips below.

### ***Driver installation error messages***

#### **Invalid Source Path**

*Description* The directory path that you entered does not exist.

*Action* Be sure that the path to the emulator software is correct. Discontinue the driver installation program by entering **N** at the next prompt. Reinvoke the `regdrv` program and reenter the emulator software directory.

#### **Can't find the emulator driver**

*Description* The directory path that you entered does not contain the emulator driver `genport.sys`.

*Action* Check the directory path and make sure it corresponds to the emulator software directory.

### **Can't find Windows NT SYSTEM32\DRIVERS\ directory**

*Description* The Windows NT drivers directory should be c:\pathname\system32\drivers. Typically, the *pathname* is c:\windows\ or c:\winnt35.

*Action* Reinvoke the regdrv program and enter the correct pathname to the system32\drivers directory. If you continue to see this message, you may need to install the emulator driver manually. Refer to Appendix B, *Installing the Emulator Device Driver Manually*, for more information.

### **Can't overwrite emulator driver – access denied**

*Description* The permissions for the emulator driver are not correct.

*Action* Check the permissions on the emulator driver that is already installed. If the driver is loaded in memory, from the command line, type **NET STOP GENPORT** to stop the driver. Reinstall the driver by following the directions in Section 2.4, *Installing the Emulator Device Driver*.

## 2.6 Step 5: Setting Up the Debugger Environment

You can define *environment variables* that set certain debugger parameters you normally use. An environment variable is a system symbol that you define and assign to a string. When you use environment variables, default values are set, making each individual invocation of the debugger simpler because these parameters are automatically specified. The debugger uses three environment variables—D\_DIR, D\_SRC, and D\_OPTIONS.

The debugger uses environment variables for finding or obtaining certain types of information. By default, the installation program sets up these environment variables:

```
SET PATH=C:\EMU470;%PATH%
SET D_DIR=C:\EMU470
```

These variables are set up in the registry under:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SessionManager\Environment
```

If you choose not to have the environment variables set up automatically, you can set up the environment variables in the System applet of the Control Panel.

The remainder of this section describes these environment variables and other variables that you can define.

### ***Identifying the directory that contains the executable files (PATH statement)***

You must include the emulator directory in your PATH statement. This allows you to specify the debugger executable without specifying the name of the directory that contains the executable file.

- If you modify your System applet to change the path information, add the following to the end of the PATH statement:

```
;C:\EMU470
```

- If you set the PATH statement from the command line, use this format:

```
SET PATH=C:\EMU470;%PATH%
```

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

The addition of **;%PATH%** ensures that this PATH statement does not undo the PATH statements in any other batch files (including the autoexec.bat file).

### ***Identifying alternate directories for the debugger (D\_DIR)***

The debugger uses the D\_DIR environment variable to name alternative directories that contain auxiliary files (emurst, emuinit.cmd, etc.) that the debugger needs. The command for assigning the environment variable is as follows:

```
SET D_DIR=C:\EMU470
```

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

### ***Identifying directories that contain source files (D\_SRC)***

The debugger uses the D\_SRC environment variable to name directories that contain program source files. The command for assigning the environment variable is as follows:

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

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

The *pathnames* are directories that contain program source files. You can separate pathnames with a semicolon or with blanks.

### ***Setting default debugger options (D\_OPTIONS)***

You might find it useful to set default debugger options using the D\_OPTIONS environment variable. When you use the D\_OPTIONS environment variable, the debugger uses the options and/or input filenames that you name with D\_OPTIONS every time you run the debugger. The command for assigning the environment variable is as follows:

```
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. The options that you can identify with D\_OPTIONS are listed on page 2-11.

These are the options you can identify with D\_OPTIONS:

Option	Brief Description
-b[b]	Select the screen size.
-bl	Select screen length.
-bw	Select screen width.
-c	Clear the .bss section.
-f <i>filename</i>	Identify a new board configuration file.
-font <i>size</i>	Select font point size.
-i <i>pathname</i>	Identify additional directories.
-me	Select little-endian format.
-min	Select the minimal debugging mode.
-n <i>processor name</i>	Identify the processor for debugging (this processor name must match the processor name as stated in the configuration file).
-p <i>port address</i>	Identify the port address.
-profile	Enter the profiling environment.
-s	Load the symbol table only.
-t <i>filename</i>	Identify a new initialization file.
-v	Load without the symbol table.

**Note: Overriding D\_OPTIONS**

You can override D\_OPTIONS by invoking the debugger or emurst with the -x option.

For more information about options, see the invocation instructions in the *TMS470R1x C Source Debugger User's Guide*.

## 2.7 Step 6: Resetting the Emulator

You must reset the emulator *before* invoking the debugger. Reset can occur only *after* you have powered up the target board. You can reset the emulator by adding the following command to the autoexec.bat file:

**emurst** [-x] [-p *port address*]

The -x option tells the emurst utility to ignore any options specified with the D\_OPTIONS environment variable.

The -p option identifies the port address.

When you want to reset your XDS510 or XDS510PP without rebooting your system, use the icons provided in the program group.

To reset the XDS510 from Windows, click on the EMURST icon in the TMS470R1x Emulator program group.



---

**Note:**

If the debugger is running, emurst will not reset the emulator. The debugger displays the following message:

```
RESET DISALLOWED : DEBUGGER RUNNING
```

---

If an error message appears after the emulator is reset, see Section 2.10, *Installation Error Messages*.

## 2.8 Step 7: Describing Your Target System to the Debugger

In order for the debugger to understand how you have configured your target system, you must supply a file for the debugger to read. You can either use the default configuration file, *board.dat*, or create your own file.

- If you are using an emulation scan path that contains only one '470 and no other devices, you can use the *board.dat* file that comes with the '470 emulator kit. This file describes the single '470 in the scan path and gives the '470 the name *tms470*. Since the debugger automatically looks for a file called *board.dat* in the current directory and in the directories specified with the *D\_DIR* environment variable, you don't need to create your own board configuration file.
- If you want to use a different name for the target device or you want the debugger to recognize a different target configuration, you must follow these steps:
  - 1) Create the board configuration file.
  - 2) Use *composer* to translate the board configuration file to binary so that the debugger can read it.
  - 3) Specify the target processor name as stated in the configuration file. This name is used with the *-n* option when invoking the debugger.

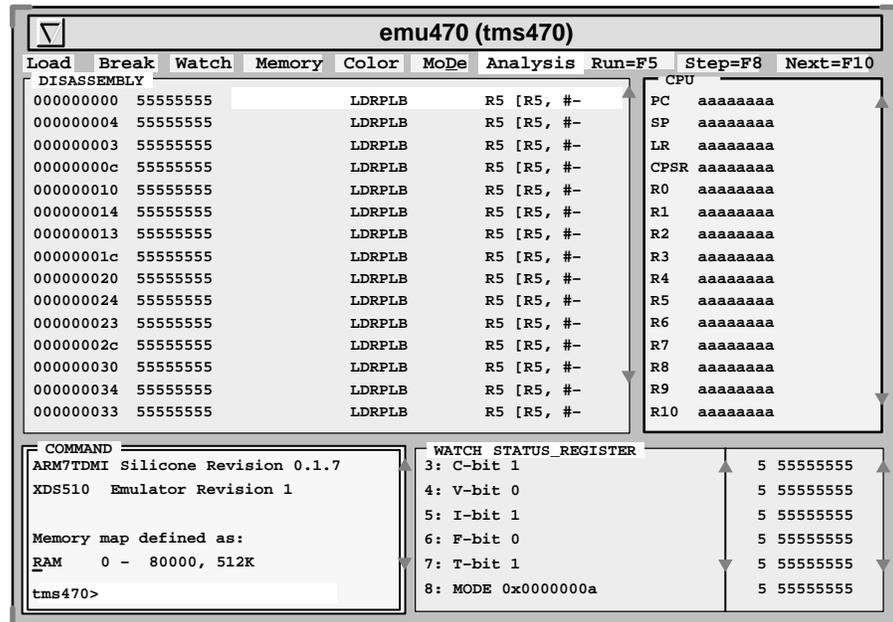
These steps are described in the *TMS470R1x C Source Debugger User's Guide*.

## 2.9 Step 8: Verifying the Installation

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

```
emu470 -p port address -n tms470
```

You should see a display similar to this one:



- If you see a display similar to this one, you have correctly installed your emulator and debugger.
- If you see a display and the lines of code say *Invalid address* or the fields in the MEMORY window are shown in red, the debugger may not be able to find the emuinit.cmd file. Check for the file in the directories specified by the D\_DIR environment variable or ensure that the file is in the current directory. Reenter the command above.
- If you don't see a display, 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.

## 2.10 Installation Error Messages

While invoking the debugger, you may see one of the following messages:

```
CANNOT INITIALIZE THE TARGET SYSTEM ! !  
- Check I/O configuration  
- Check cabling and target power
```

```
CANNOT DETECT TARGET POWER ! !  
- Check I/O configuration  
- Check cabling and target power
```

One or several of the following conditions may be the cause:

- Is the target power on?
- Is the XDS510 board installed snugly?
- Is the device installed snugly?
- Is the cable connecting your emulator and target system loose?
- Is your target board getting the correct voltage?
- Is your emulator scan path interrupted? One or more devices on the emulator scan path may have been removed. Check the connections; either they are not connected, or they are connected improperly.
- Did you use the `-n` option? Was it used with an incorrect device name? You must supply a valid device name with the `-n` option.
- After you powered up the target board, did you execute the `emurst` command? This command must be executed *after* you powered up the target board.
- Is the `board.dat` file in the current directory or in a directory specified by `D_DIR`?
- Is the emulator driver installed properly?

- Did you use the `-p` option? Is your port address correct?
  - Check to be sure the `-p` option used with the `D_OPTIONS` environment variable matches the I/O address defined by your switch settings. For information about the switch settings, see the XDS510 installation instructions in the *XDS51x Emulator Installation Guide*.
  - 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 alternative settings. 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 on page 2-14.

# Installing the Debugger With Windows 95

---

---

---

---

This chapter provides instructions for installing the C source debugger on a PC running Windows 95. When you complete the installation, turn to the *TMS470R1x C Source Debugger User's Guide*.

To install the emulator controller, see the *XDS51x Emulator Installation Guide*, which is included in your kit.

<b>Topic</b>	<b>Page</b>
<b>3.1 System Requirements</b> .....	<b>3-2</b>
<b>3.2 Step 1: Installing the XDS510 or XDS510PP Emulator Controller</b> ..	<b>3-4</b>
<b>3.3 Step 2: Installing the Debugger Software</b> .....	<b>3-4</b>
<b>3.4 Step 3: Setting Up a DLL File for Your XDS510 or XDS510PP</b> .....	<b>3-5</b>
<b>3.5 Additional XDS510PP Setup</b> .....	<b>3-6</b>
<b>3.6 Step 4: Setting Up the Debugger Environment</b> .....	<b>3-8</b>
<b>3.7 Step 5: Resetting the Emulator</b> .....	<b>3-11</b>
<b>3.8 Step 6: Describing Your Target System to the Debugger</b> .....	<b>3-13</b>
<b>3.9 Step 7: Verifying the Installation</b> .....	<b>3-14</b>
<b>3.10 Installation Error Messages</b> .....	<b>3-15</b>

### 3.1 System Requirements

To install the '470 C source debugger and TMS470R1x emulator, you need the items in the following hardware and software checklists.

#### **Hardware checklist**

- |                          |                                   |  |
|--------------------------|-----------------------------------|--|
| <input type="checkbox"/> | <b>System</b>                     | 32-bit x86- or Pentium-based PC with an ISA/EISA, VESA local, or PCI bus |
| <input type="checkbox"/> | <b>Memory</b>                     | 4–16 Mbytes of free memory   |
| <input type="checkbox"/> | <b>Disk space</b>                 | 10 Mbytes available disk space for executables and libraries             |
| <input type="checkbox"/> | <b>Display</b>                    | Color monitor  |
| <input type="checkbox"/> | <b>Required hardware</b>          | CD-ROM drive   |
| <input type="checkbox"/> | <b>Optional hardware</b>          | Microsoft-compatible mouse   |
| <input type="checkbox"/> | <b>XDS510 emulator controller</b> | XDS510 or XDS510PP emulator controller                                   |

**Note:**

The speed at which your system operates depends on the amount of RAM available on your PC and the number of debuggers running simultaneously.

**Software checklist**

- |                          |  |  |
|--------------------------|--|--|
| <input type="checkbox"/> | <b>Operating system</b>                                  | Windows 95   |
| <input type="checkbox"/> | <b>Software tools</b>                                    | TMS470R1x assembler and linker<br>Optional: TMS470R1x C compiler   |
| <input type="checkbox"/> | <b>Required files included with the debugger package</b> | <i>emu470.exe</i> is the debugger executable file  |
| <input type="checkbox"/> |  | <i>emurst.exe</i> resets the XDS510 emulator   |
| <input type="checkbox"/> |  | <i>emurstpp.exe</i> resets the XDS510PP emulator   |
| <input type="checkbox"/> |  | <i>board.dat</i> describes your target system to the debugger in terms of what devices are on the emulation scan path. The board.dat file included in the debugger package is for a target board with one TMS470R1x named <i>tms470</i> .  |
| <input type="checkbox"/> |  | <i>board.cfg</i> is a text file used to describe your target system to the debugger in terms of what devices are on the emulation scan path. The board.cfg file included in the debugger package is for a target board with one TMS470R1x named <i>tms470</i> .  |
| <input type="checkbox"/> |  | The <i>composer</i> utility allows you to convert your text board configuration file (board.cfg) into a format the debugger can read (board.dat). For the emulator to initialize properly, you must create a new board.dat file with this release of the composer or use the samples that are provided for you.  |
| <input type="checkbox"/> | <b>Optional files included with the debugger package</b> | <i>emuinit.cmd</i> is a general-purpose batch file that contains debugger commands. The version of this file that is shipped with the debugger defines a '470 memory map. 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, see the <i>TMS470R1x C Source Debugger User's Guide</i> . |

### 3.2 Step 1: Installing the XDS510 or XDS510PP Emulator Controller

Before installing the '470 debugger software, you must install an XDS510 emulator controller: the XDS510 or XDS510PP.

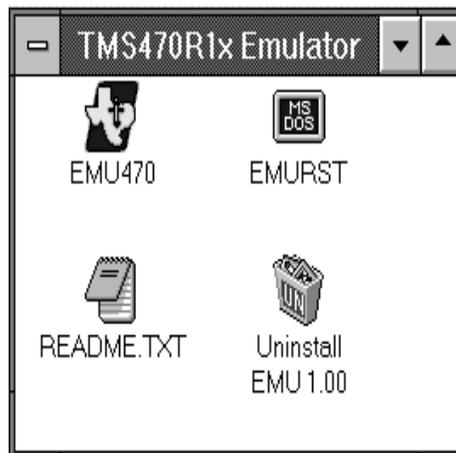
Follow the instructions in the *XDS51x Emulator Installation Guide* to install the XDS510 emulator controller.

### 3.3 Step 2: Installing the Debugger Software

To install the debugger on a Windows 95 system, follow these steps:

- 1) Insert the debugger CD-ROM into your CD-ROM drive.
- 2) Start Windows 95.
- 3) From the Start menu, select Run.
- 4) In the dialog box, enter the following command (where d: is the name of your CD-ROM drive):  
d:\setup.exe
- 5) Click on OK.
- 6) Follow the on-screen instructions.

The setup.exe program creates a program group for the TMS470R1x Emulator that looks similar to the following example:



### **3.4 Step 3: Setting Up a DLL File for Your XDS510 or XDS510PP**

After you run the installation program (setup.exe), you must setup a DLL (dynamic link library) file for your XDS510 or XDS510PP. You can set up the smg510w.dll file for your XDS510 or XDS510PP by running a batch file, or you can do it manually.

- For the XDS510:
  - Run set510.bat. This file sets up smg510w.dll file.
  - or
  - Copy the smg1095.xds file to the smg510w.dll file.
- For the XDS510PP:
  - Run set510pp.bat. This file sets up the smg510w.dll file.
  - or
  - Copy the smg1095.pp file to the smg510w.dll file.

### 3.5 Additional XDS510PP Setup

If you are using the XDS510PP, you need to complete the following:

- Check your BIOS (basic input/output system) to ensure that your printer is in one of the following modes:

EPP: Enhanced printer port

SPP4: 4-bit unidirectional mode

SPP8: 8-bit bidirectional mode

EPP mode provides the best performance. If your BIOS offers the options *EPP+ECP* and *EPP+SPP*, select *EPP+SPP*. The *EPP+ECP* option is not supported. Note the port address of your printer.

- Edit the parameters in the xds510pp.ini file to match your system. The xds510pp.ini file contains parameters for the printer address, printer mode, and printer speed. Make sure the xds510pp.ini file is in the same directory as your debugger software. The following is a sample xds510pp.ini file:

```
port = 378
mode = EPP
speed = 10
```

Where:

- Port is the I/O address of your printer port.
- Mode is the printer port mode (EPP, SPP4, or SPP8).
- Speed is a delay parameter that may be required on some systems. Set this parameter to 10 initially. You can reduce this number once your XDS510PP is up and running.

## Diagnostic files

The XDS510PP software includes diagnostic files that help you determine system information (see Table 3–1). Use these files if you have problems setting up your parallel printer port.

Table 3–1. Diagnostic Files

File	Purpose
smcmode.exe	Determines if your parallel printer port uses a port controller device manufactured by SMC.
-h	gives you more information about the options.
-r	resets the status register.
-m x	sets your printer port to mode x where: Mode 0: Standard bidirectional mode. Mode 1: EPP and SPP mode. Mode 2: ECP mode. Mode 3: ECP and EPP mode. Mode 4: Non bidirectional mode.
nscmode.exe	Determines if your parallel printer port uses a port controller device manufactured by National Semiconductor.
portchk.exe	Determines if your parallel printer port is bidirectional.

For more information about the smcmode.exe program, see Section 3.7, *Special Considerations When Resetting the XDS510PP*.

### 3.6 Step 4: Setting Up the Debugger Environment

You can define *environment variables* that set certain debugger parameters you normally use. An environment variable is a system symbol that you define and assign to a string. When you use environment variables, default values are set, making each individual invocation of the debugger simpler because these parameters are automatically specified. The debugger uses three environment variables: D\_DIR, D\_SRC, and D\_OPTIONS.

By default, the installation program sets up these environment variables in your autoexec.bat file:

```
SET PATH=C:\EMU470;%PATH%
SET D_DIR=C:\EMU470
```

If you choose not to have the environment variables set up automatically, you can modify your autoexec.bat file to exclude the SET commands above.

The remainder of this section describes these environment variables and other variables that you can define.

#### ***Identifying the directory that contains the executable files (PATH statement)***

You must include the emulator directory in your PATH statement. This allows you to specify the debugger executable without specifying the name of the directory that contains the executable file.

- If you modify your autoexec.bat file to change the path information, add the following to the end of the PATH statement:

```
;C:\EMU470
```

- If you set the PATH statement from the command line, use this format:

```
SET PATH=C:\EMU470;%PATH%
```

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

The addition of **;%PATH%** ensures that this PATH statement does not undo the PATH statements in any other batch files (including the autoexec.bat file).

#### ***Identifying alternate directories for the debugger (D\_DIR)***

The debugger uses the D\_DIR environment variable to name alternative directories that contain auxiliary files (emurst, emuinit.cmd, etc.) that the debugger needs. The command for assigning the environment variable is as follows:

```
SET D_DIR=C:\EMU470
```

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

### ***Identifying directories that contain source files (D\_SRC)***

The debugger uses the D\_SRC environment variable to name directories that contain program source files. The command for assigning the environment variable is as follows:

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

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

The *pathnames* are directories that contain program source files. You can separate pathnames with a semicolon or with blanks.

### ***Setting default debugger options (D\_OPTIONS)***

You might find it useful to set default debugger options using the D\_OPTIONS environment variable. When you use the D\_OPTIONS environment variable, the debugger uses the options and/or input filenames that you name with D\_OPTIONS every time you run the debugger. The command for assigning the environment variable is as follows:

```
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. The options that you can identify with D\_OPTIONS are listed on page 3-10.

These are the options that you can identify with D\_OPTIONS:

Option	Brief Description
-b[b]	Select the screen size.
-bl	Select screen length.
-bw	Select screen width.
-c	Clear the .bss section.
-f <i>filename</i>	Identify a new board configuration file.
-font <i>size</i>	Select font point size.
-i <i>pathname</i>	Identify additional directories.
-me	Select little-endian format.
-min	Select the minimal debugging mode.
-n <i>processor name</i>	Identify the processor for debugging (this processor name must match the processor name as stated in the configuration file).
-p <i>port address</i>	Identify the port address.
-profile	Enter the profiling environment.
-s	Load the symbol table only.
-t <i>filename</i>	Identify a new initialization file.
-v	Load without the symbol table.

**Note:**

You can override D\_OPTIONS by invoking the debugger or emurst with the -x option.

For more information about options, see the invocation instructions in the *TMS470R1x C Source Debugger User's Guide*.

### 3.7 Step 5: Resetting the Emulator

You must reset the XDS510 or XDS510PP *before* invoking the debugger. A successful reset can occur only after you have powered up the target board. You can reset the emulator by adding one of the following commands to the autoexec.bat file:

XDS510: **emurst** [-x] [-p *port address*]

XDS510PP: **emurstpp** [-x]

The -x option tells the emurst/emurstpp utility to ignore any options specified with the D\_OPTIONS environment variable.

The -p option *port address* identifies the port address.

When you want to reset your XDS510 or XDS510PP without rebooting your system, use the icons provided in the program group.

- To reset the XDS510 from Windows, click on the EMURST icon in the TMS470R1x Emulator program group.



EMURST

- To reset the XDS510PP from Windows, click on the EMURSTPP icon in the TMS470R1x Emulator program group.



EMURSTPP

**Note:**

If a debugger is running, emurst/emurstpp will not reset the emulator. The debugger displays the following message:

```
RESET DISALLOWED : DEBUGGER RUNNING
```

If an error message appears after the emulator is reset, see Section 3.10, *Installation Error Messages*.

### **Special Considerations When Resetting the XDS510PP**

You must reset your XDS510PP with the `emurstpp` command *before* invoking the debugger. After resetting the XDS510PP with the `emurstpp` command, you will see one of two messages.

- If you see the following message, the reset was successful:

```
EMURST FOR THE XDS510PP VERSION 1.0  
XDS510PP IS RESET, HARDWARE VERSION 1
```

- If you see the following message, the reset was unsuccessful:

```
EMURST FOR THE XDS510PP VERSION 1.0  
COMMUNICATIONS ERROR, OR POD HAS NO POWER
```

### **The `smcmode` command**

If `emurstpp` does not reset your XDS510PP, it is possible that the printer port device in your computer and the SMC chip in the XDS510PP are confused during power cycles. Enter the following command:

```
smcmode -r 
```

If your computer has an SMC device, this command resets and initializes the device for proper operation.

The `smcmode` command works only if your computer uses an SMC printer port controller chip. To determine if your computer uses an SMC printer port controller chip, enter the following command with no parameters:

```
smcmode 
```

This command tells you if an SMC device is installed and lists the device's configuration.

### 3.8 Step 6: Describing Your Target System to the Debugger

In order for the debugger to understand how you have configured your target system, you must supply a file for the debugger to read. You can either use the default configuration file, *board.dat*, or create your own file.

- If you are using an emulation scan path that contains only one '470 and no other devices, you can use the *board.dat* file that comes with the '470 emulator kit. This file describes the single '470 in the scan path and gives the '470 the name *tms470*. Since the debugger automatically looks for a file called *board.dat* in the current directory and in the directories specified with the *D\_DIR* environment variable, you don't need to create your own board configuration file.
- If you want to use a different name for the target device or you want the debugger to recognize a different target configuration, you must follow these steps:
  - 1) Create the board configuration file.
  - 2) Use *composer* to translate the board configuration file to binary so that the debugger can read it.
  - 3) Specify the target processor name as stated in the configuration file. This name is used with the *-n* option when invoking the debugger.

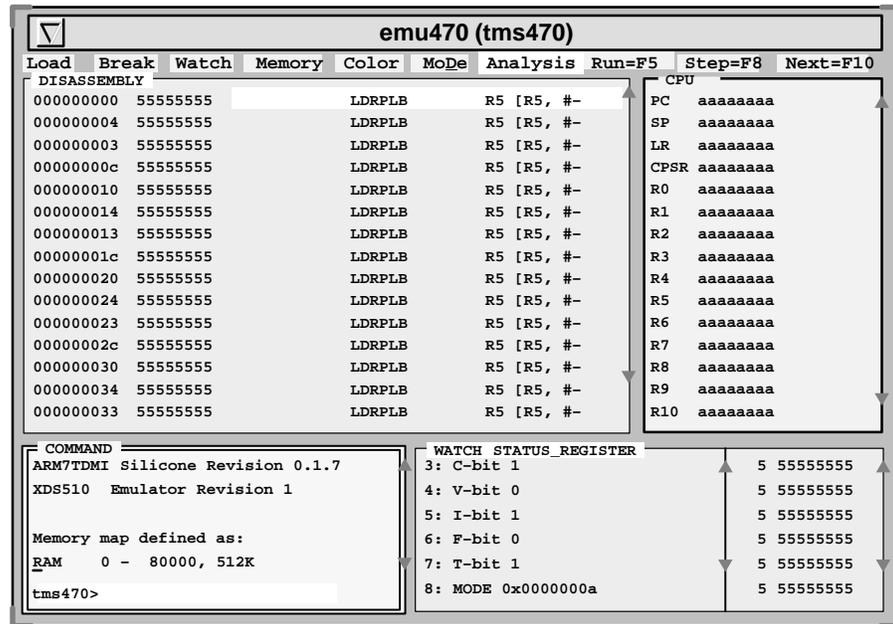
These steps are described in the *TMS470R1x C Source Debugger User's Guide*.

### 3.9 Step 7: Verifying the Installation

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

```
emu470 -p port address -n tms470
```

You should see a display similar to this one:



- If you see a display similar to this one, you have correctly installed your emulator and debugger.
- If you see a display and the lines of code say *Invalid address* or the fields in the MEMORY window are shown in red, the debugger may not be able to find the emuinit.cmd file. Check for the file in the directories specified by the D\_DIR environment variable or ensure that the file is in the current directory. Reenter the command above.
- If you don't see a display, 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.

### 3.10 Installation Error Messages

While invoking the debugger, you may see one of the following messages:

```
CANNOT INITIALIZE THE TARGET SYSTEM ! !  
- Check I/O configuration  
- Check cabling and target power
```

```
CANNOT DETECT TARGET POWER ! !  
- Check I/O configuration  
- Check cabling and target power
```

One or several of the following conditions may be the cause:

- Is the target power on?
- Is the XDS510/XDS510PP board installed snugly?
- Is the device installed snugly?
- Is the cable connecting your emulator and target system loose?
- Is your target board getting the correct voltage?
- Is your emulator scan path interrupted? One or more devices on the emulator scan path may have been removed. Check the connections; either they are not connected, or they are connected improperly.
- Did you use the `-n` option? Was it used with an incorrect device name? You must supply a valid device name with the `-n` option.
- After you powered up the target board, did you execute the `emurst/ emurstpp` command? This command must be executed *after* you powered up the target board.

- Did you use the `-p` option? Is your port address correct (XDS510 only)?
  - Check to be sure the `-p` option used with the `D_OPTIONS` environment variable matches the I/O address defined by your switch settings. For information about the switch settings, see the XDS510 installation instructions in the *XDS51x Emulator Installation Guide*.
  - 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 alternative settings. Modify the `-p` option of the `D_OPTIONS` environment variable to reflect the change in your switch settings.
- Is the `board.dat` file in the current directory or in a directory specified by `D_DIR`?

After you have checked all of the above, repeat the verification instructions on page 3-14.

# Installing the Debugger on a SPARCstation

---

---

---

This chapter provides instructions for installing the TMS470R1x C source debugger on a SPARCstation running OpenWindows™ under SunOS™ version 4.1.x. After completing the installation, see the *TMS470R1x C Source Debugger User's Guide*.

To install the emulator controller, see the *XDS51x Emulator Installation Guide*.

<b>Topic</b>	<b>Page</b>
<b>4.1 System Requirements .....</b>	<b>4-2</b>
<b>4.2 Step 1: Installing the XDS510WS Emulator Controller .....</b>	<b>4-4</b>
<b>4.3 Step 2: Installing the Debugger Software .....</b>	<b>4-4</b>
<b>4.4 Step 3: Ensuring That the Emulator Supports the Debugger .....</b>	<b>4-6</b>
<b>4.5 Step 4: Describing Your Target System to the Debugger .....</b>	<b>4-7</b>
<b>4.6 Step 5: Setting Up the Debugger Environment .....</b>	<b>4-8</b>
<b>4.7 Step 6: Verifying the Installation .....</b>	<b>4-11</b>
<b>4.8 Using the Debugger With the X Window System .....</b>	<b>4-12</b>

## 4.1 System Requirements

To install the '470 C source debugger and TMS470R1x emulator, you need the items in the following hardware and software checklists.

### Hardware checklist

- |                          |                                   |  |
|--------------------------|-----------------------------------|--|
| <input type="checkbox"/> | <b>System</b>                     | SPARCstation or 100%-compatible system |
| <input type="checkbox"/> | <b>Display</b>                    | Color monitor                          |
| <input type="checkbox"/> | <b>CD-ROM drive</b>               | CD-ROM drive                           |
| <input type="checkbox"/> | <b>XDS510 emulator controller</b> | XDS510WS™ emulator controller          |

### Software checklist

- |                          |  |  |
|--------------------------|--|--|
| <input type="checkbox"/> | <b>Operating system</b>                                  | SunOS version 4.1.x or SunOS version 5.x (also known as Solaris™ 2.x) using an X Window System™-based window manager, such as OpenWindows version 3.x.   |
| <input type="checkbox"/> | <b>Root privileges</b>                                   | If you are running SunOS 4.1.x, 5.0, or 5.1, you <i>must</i> have root privileges to mount and unmount the CD-ROM. If you don't, get help from your system administrator.  |
| <input type="checkbox"/> | <b>Software tools</b>                                    | TMS470R1x assembler and linker<br>Optional: TMS470R1x C compiler   |
| <input type="checkbox"/> | <b>Required files included with the debugger package</b> | <i>emu470</i> is the debugger executable file.   |
| <input type="checkbox"/> |  | <i>r470510ws.out</i> is the executable portion of the debugger that runs on the emulator.  |
| <input type="checkbox"/> |  | <i>emurst</i> resets the emulator and downloads <i>r470510ws.out</i> to the emulator.  |
| <input type="checkbox"/> |  | <i>board.dat</i> describes your target board to your debugger in terms of what devices are in the emulation scan path. The <i>board.dat</i> file included in the debugger package is for a target board with one TMS470R1x named <i>tms470</i> . |



*board.cfg* is a text file used to describe your target system to the debugger in terms of what devices are on the emulation scan path. The *board.cfg* file included in the debugger package is for a target board with one TMS470R1x named *tms470*.



The *composer* utility allows you to convert your text board configuration file (*board.cfg*) into a format the debugger can read (*board.dat*). For the emulator to initialize properly, you must create a new *board.dat* file with this release of the composer or use the samples that are provided for you.



**Optional files  
included with the  
debugger package**

*emuinit.cmd* is a general-purpose batch file that contains debugger commands. The version of this file that is shipped with the debugger defines a '470 memory map. 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, see the *TMS470R1x C Source Debugger User's Guide*.

## 4.2 Step 1: Installing the XDS510WS Emulator Controller

Before installing the '470 debugger software, you must install an XDS510WS emulator controller.

Follow the instructions in the *XDS51x Emulator Installation Guide* to install the XDS510WS emulator controller.

## 4.3 Step 2: Installing the Debugger Software

This section explains how to install the debugger software on your hard disk system. The software package is shipped on CD-ROM. To install the emulator software, you must mount the CD-ROM, copy the files, and unmount the CD-ROM.

### Note:

If you are running SunOS 4.1.x, 5.0, or 5.1, you *must* have root privileges to mount or unmount the CD-ROM. If you don't, get help from your system administrator.

### Mounting the CD-ROM

The steps to mount the CD-ROM vary according to your operating system version:

- If you have a SunOS 4.1.x, load the CD-ROM into the drive. As root, enter the following from a command shell:

```
mount -rt hsfs /dev/sr0 /cdrom   
exit   
cd /cdrom/sparc 
```

- If you have SunOS 5.0 or 5.1, load the CD-ROM into the drive. As root, enter the following from a command shell:

```
mount -rF hsfs /dev/sr0 /cdrom   
exit   
cd /cdrom/cdrom0/sparc 
```

- If you have SunOS 5.2 or higher:
  - If your CD-ROM drive is already attached, load the CD-ROM into the drive and enter the following from a command shell:

```
cd /cdrom/cdrom0/sparc
```

- If you do not have a CD-ROM drive attached, you must shut down your system to the PROM level, attach the CD-ROM drive, and enter the following:

```
boot -r
```

After you log into your system, load the CD-ROM into the drive and enter the following from a command shell:

```
cd /cdrom/cdrom0/sparc
```

### Copying the files

After you mount the CD-ROM, you must create the directory that will contain the debugger software and copy the software to that directory.

- 1) Create a directory named emu470 on your hard disk. To create this directory, enter:

```
mkdir /your_pathname/emu470
```

- 2) Copy the files from the CD-ROM to your hard-disk system:

```
cp -r * /your_pathname/emu470
```

### Unmounting the CD-ROM

You must unmount the CD-ROM after copying the files.

- If you have a SunOS 4.1.x, 5.0, or 5.1, as root, enter the following from a command shell:

```
cd  
umount /cdrom  
eject /dev/sr0  
exit
```

- If you have SunOS 5.2 or higher, enter the following from a command shell:

```
cd  
eject
```

#### 4.4 Step 3: Ensuring That the Emulator Supports the Debugger

The ROM code for the XDS510WS does not contain the information necessary to communicate with the target; that code must be downloaded from the host. The *emurst* program downloads the necessary code for proper emulation.

To run this program, enter the *emurst* command in the following format:

**emurst [-x] [-p *port address*] r470510ws.out**

The `-x` option tells the *emurst* utility to ignore any options specified with the `D_OPTIONS` environment variable.

The `-p` option identifies the SCSI port address.

`r470510ws.out` contains the information necessary to communicate with the target.

You can be sure that *emurst* succeeded when only the first and second LEDs from the left are on. For more information about XDS510WS LEDs, see the *XDS51x Emulator Installation Guide*.

## 4.5 Step 4: Describing Your Target System to the Debugger

In order for the debugger to understand how you have configured your target system, you must supply a file for the debugger to read. You can either use the default configuration file, *board.dat*, or create your own file.

- If you are using an emulation scan path that contains only one '470 and no other devices, you can use the *board.dat* file that comes with the '470 emulator kit. This file describes the single '470 in the scan path and gives the '470 the name *tms470*. Since the debugger automatically looks for a file called *board.dat* in the current directory and in the directories specified with the *D\_DIR* environment variable, you don't need to create your own board configuration file.
- If you want to use a different name for the target device or you want the debugger to recognize a different target configuration, you must follow these steps:
  - 1) Create the board configuration file.
  - 2) Use *composer* to translate the board configuration file to binary so that the debugger can read it.
  - 3) Specify the target processor name as stated in the configuration file. This name is used with the *-n* option when invoking the debugger.

These steps are described in the *TMS470R1x C Source Debugger User's Guide*.

## 4.6 Step 5: Setting Up the Debugger Environment

To ensure that your debugger works correctly, you must:

- Modify the path shell variable to identify the emu470 directory.
- Set up the environment variables that you want to use.
- Invoke the new or modified .cshrc file.

### **Modifying the path shell variable**

You must include the debugger directory in your shell path. To do this, you need to modify your shell configuration file in your home directory (for example, the .cshrc file for a C shell). Include the pathname to your emu470 directory in your path. The following statement is an example of what a typical path-variable definition looks like:

```
set path = ( . /bin /usr/ucb /usr/contrib/bin /usr/bin \  
/usr/openwin/bin )
```

The following is an example of a modified path variable. The part of the path that is boldface is an example of a pathname that identifies the emu470 directory:

```
set path = ( . /bin /usr/ucb /usr/contrib/bin /usr/bin \  
/usr/openwin/bin /user/fred/emu470 )
```

### **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. You can set up the environment variables in your shell configuration file. The debugger uses four environment variables, named D\_DIR, D\_SRC, D\_OPTIONS, and DISPLAY:

- Set up the D\_DIR environment variable to identify the emu470 directory by defining the D\_DIR environment variable:

```
setenv D_DIR "/user/fred/emu470"
```

(Be sure to enclose the directory name within quotes.)

This directory contains auxiliary files (such as emuinit.cmd) 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 access from the debugger. The general format for doing this is:

```
setenv D_SRC "pathname1;pathname2..."
```

(Be sure to enclose the path names within one set of quotes.)

For example, if your programs were in a directory named */user/fred/emu470/samples*, the D\_SRC setup would be:

```
setenv D_SRC "/user/fred/emu470/samples"
```

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

**setenv D\_OPTIONS "[object filename] [debugger options]"**

(Be sure to enclose the options and filenames within one set of quotes.)

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:

Option	Brief Description
-b[b]	Select the screen size.
-bl	Select screen length.
-bw	Select screen width.
-c	Clear the .bss section.
-f <i>filename</i>	Identify a new board configuration file.
-font <i>size</i>	Select font point size.
-i <i>pathname</i>	Identify additional directories.
-me	Select little-endian format.
-min	Select the minimal debugging mode.
-n <i>processor name</i>	Identify the processor for debugging (this processor name must match the processor name as stated in the configuration file).
-p <i>port address</i>	Identify the port address.
-profile	Enter the profiling environment.
-s	Load the symbol table only.
-t <i>filename</i>	Identify a new initialization file.
-v	Load without the symbol table.

#### **Note: Overriding D\_OPTIONS**

You can override D\_OPTIONS by invoking the debugger or emurst with the -x option.

For more information about options, see the invocation section in the *TMS470R1x C Source Debugger User's Guide*.

- If you are using the X Window System, you can use the DISPLAY environment variable to display the debugger on a different machine from the one the debugger is running on. The general format for doing this is:

**setenv DISPLAY *machine name***

For example, if you are running the debugger on a machine called opie and you want the debugger display to appear on a machine called barney, the DISPLAY setup would be:

```
setenv DISPLAY barney:0
```

You can also display the debugger on a different machine by using the -d option when invoking the debugger.

```
emu470 -d barney:0
```

For more information about using the debugger under the X Window system, see Section 4.8, *Using the Debugger With the X Window System*.

### ***Invoking the new or modified .cshrc file***

When you modify your shell configuration file, you must ensure that the changes are made to your current session. For example, if you are using a C shell, use this command to reread the .cshrc file:

```
source ~/.cshrc
```

## 4.7 Step 6: Verifying the Installation

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

```
emu470 -p port address -n tms470
```

You should see a display similar to this one:

The screenshot shows the emu470 (tms470) debugger interface. The main window is titled "emu470 (tms470)" and contains several panes:

- DISASSEMBLY:** A table with columns for Load, Break, Watch, Memory, Color, MoDe, Analysis, Run=F5, Step=F8, and Next=F10. The table lists assembly instructions (LDRPLB) with their addresses (00000000 to 00000033) and operands (R5 [R5, #-]).
- CPU:** A list of CPU registers (PC, SP, LR, CPSR, R0-R10) with their current values (all are "aaaaaaaa").
- COMMAND:** A text area showing the output of the "ARM7TDMI Silicone Revision 0.1.7" and "XDS510 Emulator Revision 1" commands, along with the memory map definition: "RAM 0 - 80000, 512K".
- WATCH STATUS REGISTER:** A table showing the status of various watch registers (3: C-bit 1, 4: V-bit 0, 5: I-bit 1, 6: F-bit 0, 7: T-bit 1, 8: MODE 0x0000000a) with their values (all are "5 5555555").

- If you see a display similar to this one, you have correctly installed your emulator and debugger.
- If you do not see a display, then your debugger, board, or XDS510WS 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.
- If you continue to experience problems, see Appendix A, *Workstation Troubleshooting*.

## 4.8 Using the Debugger With the X Window System

If you're using the X Window System to run the '470 debugger, you need to know about the keyboard's special keys, the debugger fonts, and using the debugger on a monochrome monitor.

### Using the keyboard's special keys

The debugger uses some special keys that you can map differently from your particular keyboard. Some keyboards, such as the Sun Type 5 keyboard, have these special symbols on separate keys. Other keyboards, such as the Sun Type 4 keyboard, do not have the special keys, but the functions are available.

The special keys that the debugger uses are shown in the following table with their corresponding keysym. A *keysym* is a label that interprets a keystroke; it allows you to modify the action of a key on the keyboard.

Debugger Key Needed	Keysym for That Function
(F1) to (F10)	F1 to F10
(PAGE UP)	Prior
(PAGE DOWN)	Next
(HOME)	Home
(END)	End
(INSERT)	Insert
(→)	Right
(←)	Left
(↑)	Up
(↓)	Down

Use the X utility `xev` to check the keysyms associated with your keyboard. If you need to change the keysym definitions, use the `xmodmap` utility. For example, you could create a file that contains the following commands and use that file with `xmodmap` to map a Sun Type 4 keyboard to the keys listed above:

```
key code      keysym
keysym R13    = End
keysym Down   = Down
keysym F35    = Next
keysym Left   = Left
keysym Right  = Right
keysym F27    = Home
keysym Up     = Up
keysym F29    = Prior
keysym Insert = Insert
```

Refer to your X Window System documentation for more information about using `xev` and `xmodmap`.

### **Changing the debugger font**

You can change the font of the debugger screen by using the `xrdb` utility and modifying the `.Xdefaults` file in your root directory. For example, to change the '470 debugger fonts to Courier, add the following line to the `.Xdefaults` file:

```
emu470*font:courier
```

For more information about using `xrdb` to change the font, refer to your X Window System documentation.

### **Color mappings on monochrome screens**

Although a color monitor is recommended, you can use a monochrome monitor. The following table shows the color mappings for monochrome screens:

<b>Color</b>	<b>Appearance on Monochrome Screen</b>
black	black
blue	black
green	white
cyan	white
red	black
magenta	black
yellow	white
white	white



# Installing the Debugger on an HP Workstation

---

---

---

---

This chapter provides instructions for installing the TMS470R1x C source debugger on an HP 9000 Series 700™ PA-RISC™ computer with HP-UX™ 9.0x. After completing the installation, turn to the *TMS470R1x C Source Debugger User's Guide*.

To install the emulator controller, see the *XDS51x Emulator Installation Guide*.

<b>Topic</b>	<b>Page</b>
<b>5.1 System Requirements</b> .....	<b>5-2</b>
<b>5.2 Step 1: Installing the XDS510WS Emulator Controller</b> .....	<b>5-4</b>
<b>5.3 Step 2: Installing the Debugger Software</b> .....	<b>5-4</b>
<b>5.4 Step 3: Ensuring That the Emulator Supports the Debugger</b> .....	<b>5-5</b>
<b>5.5 Step 4: Describing Your Target System to the Debugger</b> .....	<b>5-6</b>
<b>5.6 Step 5: Setting Up the Debugger Environment</b> .....	<b>5-7</b>
<b>5.7 Step 6: Verifying the Installation</b> .....	<b>5-10</b>
<b>5.8 Using the Debugger With the X Window System</b> .....	<b>5-11</b>

## 5.1 System Requirements

To install the '470 C source debugger and TMS470R1x emulator, you need the items in the following hardware and software checklists.

### *Hardware checklist*

- |                          |                                   |                                     |
|--------------------------|-----------------------------------|-------------------------------------|
| <input type="checkbox"/> | <b>System</b>                     | HP 9000 Series 700 PA-RISC computer |
| <input type="checkbox"/> | <b>Display</b>                    | Color monitor                       |
| <input type="checkbox"/> | <b>CD-ROM drive</b>               | CD-ROM drive                        |
| <input type="checkbox"/> | <b>XDS510 emulator controller</b> | XDS510WS emulator controller        |

**Software checklist**

- Operating system** HP-UX version 9.0x or higher
- Root privileges** You *must* have root privileges to mount and unmount the CD-ROM. If you don't, get help from your system administrator.
- Software tools** TMS470R1x assembler and linker  
Optional: TMS470R1x C compiler
- Required files included with the debugger package** *emu470* is the debugger executable file.
- r470510ws.out* is the executable portion of the debugger that runs on the emulator.
- emurst* resets the emulator and downloads *r470510ws.out* to the emulator.
- board.dat* describes your target board to your debugger in terms of what devices are on the emulation scan path. The *board.dat* file included in the debugger package is for a target board with one TMS470R1x named *tms470*.
- board.cfg* is a text file used to describe your target system to the debugger in terms of what devices are on the emulation scan path. The *board.cfg* file included in the debugger package is for a target board with one TMS470R1x named *tms470*.
- The *composer* utility allows you to convert your text board configuration file (*board.cfg*) into a format the debugger can read (*board.dat*). For the emulator to initialize properly, you must create a new *board.dat* file with this release of the composer or use the samples that are provided for you.
- Optional files included with the debugger package** *emuinit.cmd* is a general-purpose batch file that contains debugger commands. The version of this file that's shipped with the debugger defines a '470 memory map. 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, see the *TMS470R1x C Source Debugger User's Guide*.

## 5.2 Step 1: Installing the XDS510WS Emulator Controller

Before installing the '470 debugger software, you must install an XDS510WS emulator controller.

Follow the instructions in the *XDS51x Emulator Installation Guide* to install the XDS510WS emulator controller.

## 5.3 Step 2: Installing the Debugger Software

This section explains how to install the debugger software on your hard-disk system. The software package is shipped on a CD-ROM. To install the software, you must mount the CD-ROM, copy the files, and unmount the CD-ROM.

**Note:**

You *must* have root privileges to mount or unmount the CD-ROM. If you don't, get help from your system administrator.

### **Mounting the CD-ROM**

As root, mount the CD-ROM using the UNIX mount command or the SAM (system administration manager):

- To use the UNIX mount command, enter:

```
mount -rt cdfs /dev/dsk/your_cdrom_device /cdrom   
exit 
```

Make the hp directory on the CD-ROM the current directory. For example, if the CD-ROM is mounted at /cdrom, enter:

```
cd /cdrom/hp 
```

- To use SAM to mount the CD-ROM, see *System Administration Tasks*, the HP documentation about SAM, for instructions.

### **Copying the files**

After you have mounted the CD-ROM, you must create the directory that will contain the debugger software and copy the software to that directory.

- 1) Create a directory named emu470 on your hard disk. To create this directory, enter:

```
mkdir /your_pathname/emu470
```

- 2) Copy the files from the CD-ROM to your hard-disk system:

```
cp -r * /your_pathname/emu470
```

### **Unmounting the CD-ROM**

You must unmount the CD-ROM after copying the files. As root, enter:

```
cd  
umount /cdrom  
exit
```

## **5.4 Step 3: Ensuring That the Emulator Supports the Debugger**

The ROM code for the XDS510WS does not contain the information necessary to communicate with the target; that code must be downloaded from the host. The *emurst* program downloads the necessary code for proper emulation.

To run this program, enter the *emurst* command in the following format:

```
emurst [-x] [-p port address] pathname-filename
```

The *-x* option tells the *emurst* utility to ignore any options specified with the *D\_OPTIONS* environment variable.

The *-p* option identifies the SCSI port address.

*r470510ws.out* contains the information necessary to communicate with the target.

You can be sure that *emurst* succeeded when only the first and second LEDs from the left are on. For more information about XDS510WS LEDs, see the *XDS51x Emulator Installation Guide*.

## 5.5 Step 4: Describing Your Target System to the Debugger

In order for the debugger to understand how you have configured your target system, you must supply a file for the debugger to read. You can either use the default configuration file, *board.dat*, or create your own file.

- If you are using an emulation scan path that contains only one '470 and no other devices, you can use the *board.dat* file that comes with the '470 emulator kit. This file describes the single '470 in the scan path and gives the '470 the name *tms470*. Since the debugger automatically looks for a file called *board.dat* in the current directory and in the directories specified with the *D\_DIR* environment variable, you don't need to create your own board configuration file.
- If you want to use a different name for the target device or you want the debugger to recognize a different target configuration, you must follow these steps:
  - 1) Create the board configuration file.
  - 2) Use *composer* to translate the board configuration file to binary so that the debugger can read it.
  - 3) Specify the target processor name as stated in the configuration file. This name is used with the *-n* option when invoking the debugger.

These steps are described in the *TMS470R1x C Source Debugger User's Guide*.

## 5.6 Step 5: Setting Up the Debugger Environment

To ensure that your debugger works correctly, you must:

- Modify the path shell variable to identify the emu470 directory.
- Set up the environment variables that you want to use.
- Invoke the new or modified .cshrc file.

### Modifying the path shell variable

You must include the debugger directory in your shell path. To do this, you need to modify your shell configuration file in your home directory (for example, the .cshrc file for a C shell). Include the pathname to your emu470 directory in your path. The following statement is an example of what a typical path-variable definition looks like:

```
set path = ( . /bin /usr/ucb /usr/contrib/bin /usr/bin \
/usr/openwin/bin)
```

The following is an example of a modified path variable. The part of the path that is boldface is an example of a pathname that identifies the emu470 directory:

```
set path ( . /bin /usr/ucb /usr/contrib/bin /usr/bin \
/usr/openwin/bin /user/fred/emu470)
```

### 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. You can set up the environment variables in your shell configuration file. The debugger uses four environment variables, named D\_DIR, D\_SRC, D\_OPTIONS, and DISPLAY:

- Set up the D\_DIR environment variable to identify the emu470 directory by defining the D\_DIR environment variable:

```
setenv D_DIR "/user/fred/emu470"
```

(Be sure to enclose the directory name within quotes.)

This directory contains auxiliary files (such as emuinit.cmd) 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 this is:

```
setenv D_SRC "pathname1;pathname2;..."
```

(Be sure to enclose the path names within one set of quotes.)

For example, if your programs were in a directory named */user/fred/emu470/samples*, the D\_SRC setup would be:

```
setenv D_SRC "/user/fred/emu470/samples"
```

- You can use several options when you invoke the debugger. If you use the same options repeatedly, it's convenient to specify them with `D_OPTIONS`. The general format for doing this is:

**setenv D\_OPTIONS “[object filename] [debugger options]”**

(Be sure to enclose the options and filenames within one set of quotes.)

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`:

Option	Brief Description
-b[b]	Select the screen size.
-bl	Select screen length.
-bw	Select screen width.
-c	Clear the .bss section.
-f <i>filename</i>	Identify a new board configuration file.
-font <i>size</i>	Select font point size.
-i <i>pathname</i>	Identify additional directories.
-me	Select little-endian format.
-min	Select the minimal debugging mode.
-n <i>processor name</i>	Identify the processor for debugging (this processor name must match the processor name as stated in the configuration file).
-p <i>port address</i>	Identify the port address.
-profile	Enter the profiling environment.
-s	Load the symbol table only.
-t <i>filename</i>	Identify a new initialization file.
-v	Load without the symbol table.

**Note: Overriding D\_OPTIONS**

You can override `D_OPTIONS` by invoking the debugger or `emurst` with the `-x` option.

For more information about options, see the invocation instructions in the *TMS470R1x C Source Debugger User's Guide*.

- If you are using the X Window System, you can use the DISPLAY environment variable to display the debugger on a different machine from the one the debugger is running on. The general format for doing this is:

**setenv DISPLAY machine name**

For example, if you are running the debugger on a machine called opie and you want the debugger display to appear on a machine called barney, the DISPLAY setup would be:

```
setenv DISPLAY barney:0
```

You can also display the debugger on a different machine by using the -d option when invoking the debugger.

```
emu470 -d barney:0
```

For more information about using the debugger under the X Window system, see Section 5.8, *Using the Debugger With the X Window System*.

### ***Invoking the new or modified .cshrc file***

When you modify your shell configuration file, you must ensure that the changes are made to your current session. For example, if you are using a C shell, use this command to reread the .cshrc file:

```
source ~/.cshrc
```

## 5.7 Step 6: Verifying the Installation

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

```
emu470 -p port address -n tms470
```

You should see a display similar to this one:

The screenshot shows the emu470 (tms470) debugger interface. The main window is titled "emu470 (tms470)" and contains several panes:

- DISASSEMBLY:** A table with columns: Load, Break, Watch, Memory, Color, MoDe, Analysis, Run=F5, Step=F8, Next=F10. The table lists assembly instructions (LDRPLB) with addresses from 00000000 to 00000033, all with a watch value of 55555555.
- CPU:** A list of CPU registers (PC, SP, LR, CPSR, R0-R10) with values of "aaaaaaaa".
- COMMAND:** A text area showing:
 

```
ARM7TDMI Silicone Revision 0.1.7
XDS510 Emulator Revision 1

Memory map defined as:
RAM 0 - 80000, 512K

tms470>
```
- WATCH STATUS REGISTER:** A list of watch registers (3-8) with their bit values and a watch value of 55555555:
 

```
3: C-bit 1      5 55555555
4: V-bit 0     5 55555555
5: I-bit 1     5 55555555
6: F-bit 0     5 55555555
7: T-bit 1     5 55555555
8: MODE 0x0000000a 5 55555555
```

- If you see a display similar to this one, you have correctly installed your emulator and debugger.
- If you do not see a display, then your debugger, board, or XDS510WS 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.
- If you continue to experience problems, see Appendix A, *Workstation Troubleshooting*.

## 5.8 Using the Debugger With the X Window System

If you're using the X Window System to run the '470 debugger, you need to know about the keyboard's special keys, the debugger fonts, and using the debugger on a monochrome monitor.

### Using the keyboard's special keys

The debugger uses some special keys that you can map differently from your particular keyboard. Some keyboards, such as the Sun Type 5 keyboard, have these special symbols on separate keys. Other keyboards, such as the Sun Type 4 keyboard, do not have the special keys, but the functions are available.

The special keys that the debugger uses are shown in the following table with their corresponding keysym. A *keysym* is a label that interprets a keystroke; it allows you to modify the action of a key on the keyboard.

Debugger Key Needed	Keysym for That Function
(F1) to (F10)	F1 to F10
(PAGE UP)	Prior
(PAGE DOWN)	Next
(HOME)	Home
(END)	End
(INSERT)	Insert
(→)	Right
(←)	Left
(↑)	Up
(↓)	Down

Use the X utility `xev` to check the keysyms associated with your keyboard. If you need to change the keysym definitions, use the `xmodmap` utility. For example, you could create a file that contains the following commands and use that file with `xmodmap` to map a Sun Type 4 keyboard to the keys listed above:

```

key code      keysym
keysym R13    = End
keysym Down   = Down
keysym F35    = Next
keysym Left   = Left
keysym Right  = Right
keysym F27    = Home
keysym Up     = Up
keysym F29    = Prior
keysym Insert = Insert

```

Refer to your X Window System documentation for more information about using `xev` and `xmodmap`.

### **Changing the debugger font**

You can change the font of the debugger screen by using the `xrdb` utility and modifying the `.Xdefaults` file in your root directory. For example, to change the '470 debugger fonts to Courier, add the following line to the `.Xdefaults` file:

```
emu470*font:courier
```

For more information about using `xrdb` to change the font, refer to your X Window System documentation.

### **Color mappings on monochrome screens**

Although a color monitor is recommended, you can use a monochrome monitor. The following table shows the color mappings for monochrome screens:

<b>Color</b>	<b>Appearance on Monochrome Screen</b>
black	black
blue	black
green	white
cyan	white
red	black
magenta	black
yellow	white
white	white

# Workstation Troubleshooting

---

---

---

This chapter describes some common problems you may encounter while using your emulator or debugger on your SPARCstation or HP workstation.

<b>Topic</b>	<b>Page</b>
<b>A.1 Problems When Booting Your Workstation .....</b>	<b>A-2</b>
<b>A.2 Problems With Multiple Emulators on SunOS .....</b>	<b>A-2</b>
<b>A.3 Problems When Resetting the Emulator .....</b>	<b>A-3</b>
<b>A.4 Problems When Invoking the Debugger .....</b>	<b>A-5</b>
<b>A.5 Additional Emulator and Debugger Problems .....</b>	<b>A-7</b>

## A.1 Problems When Booting Your Workstation

After installing your emulator, problems may occur when you attempt to boot your workstation. The following are typical problems and suggestions to help resolve these problems:

- Your workstation will not boot when connected to your emulator, even if your emulator is not turned on.
  - 1) Be sure that all of your SCSI cables are connected securely and that the SCSI bus is terminated properly.
  - 2) Remove any unnecessary SCSI devices from the bus.
  - 3) Make sure that the total length of the SCSI bus is less than six meters, including the section of the bus within the SPARCstation chassis.
- Your workstation will boot when the emulator is turned off but will not boot when the emulator is turned on.

Your emulator's SCSI ID conflicts with the SCSI ID of another device on the SCSI bus.

For more information, see the *XDS51x Emulator Installation Guide*.

## A.2 Problems With Multiple Emulators on SunOS

When you have multiple emulators running constantly, they may consume most of the CPU time. To overcome this situation, run the emulators at a lower priority using the UNIX nice command. See the man page entry on nice for more information.

### A.3 Problems When Resetting the Emulator

After you power up the emulator and the workstation, and if you have the following problems while attempting to reset the emulator, implement the applicable solutions:

- ❑ When you execute the emurst command, you receive this message:

```
emurst file [.out]:
```

You forgot to specify the *pathname–filename* of the r470510ws.out file. You can specify it at this prompt or you can reeducate emurst with *pathname–filename* specified on the command line.

- ❑ When executing the emurst command, you receive this message:

```
>> can't initialize the target system
```

- 1) You haven't set the IPCSEMAPHORE option to allow the debugger to access the emulator. Be sure that the configuration file, EMULATOR, has the options line set correctly, without comments. Then, use the corrected configuration file to build the currently executing kernel (see the XDS510WS installation instructions in the *XDS51x Emulator Installation Guide* for more information).

- 2) There are too many current semaphores on the system. Clean up the unused semaphores by using the **ipcs –st** and **ipcrm** utilities, and try to execute emurst again.

- 3) You may not have permission to access the driver file you specified with the **–p** debugger option. Normally, you specify the **–p** debugger option on the command line or in the D\_OPTIONS environment variable. Remember, if you haven't specifically reset the driver file number to another number, the default is 4. Have the root user execute the following command, and try to execute emurst again.

```
chmod a+rw /dev/rsd#a
```

- 4) The driver file you specified with the **–p** debugger option is not correctly associated to your emulator in your configuration file. Make sure your configuration file contains a line similar to this:

```
disk sd# at scsibus<m> target <s> lun 0
```

where # is the device driver number. The <m> is zero (0) unless the XDS510WS is connected to a second SCSI bus that you added to your SPARCstation, which causes <m> to change. The <s> is the SCSI ID of the XDS510WS you set with the switch at the front of the XDS510WS. Use the corrected configuration file to build the currently executing kernel (see the XDS510WS installation instructions in the *XDS51x Emulator Installation Guide* for more information).

5) You have not turned on the XDS510WS, or it has not completed its self test. Turn on the XDS510WS and wait for the self test to complete successfully before executing emurst. The self test has completed once the sixth LED from the left is off and the first, second, and fifth LEDs from the left are on.

When executing the emurst command, you receive this message:

```
>> error loading file
```

- 1) The emurst utility can't find the r470510ws.out file as specified. If you didn't specify the *pathname-filename* with an extension as part of the name, the emurst utility appends the default extension *.out* to the name.
- 2) If you didn't provide path information (just the filename), emurst searches first in the current directory and then in all of the directories specified in the *D\_DIR* environment variable before returning this error. Make sure the correct file is located where emurst can find it.
- 3) The file that you specified to emurst is not appropriate for this use. Use the r470510ws.out file that is included with the debugger software.

## A.4 Problems When Invoking the Debugger

If you encounter these problems when you invoke the debugger, the suggested solutions may resolve the problems:

- You receive the following message when executing the `emu470` command:

```
CANNOT INITIALIZE THE TARGET !!  
- Check I/O configuration  
- Check cabling and target power
```

- 1) The `emurst` command didn't successfully execute before you tried to invoke the debugger. Execute `emurst` (see Section 4.4, *Step 3: Ensuring That the Emulator Supports the Debugger*, on page 4-6 for SPARCstations and Section 5.4, *Step 3: Ensuring That the Emulator Supports the Debugger*, on page 5-5, for HP workstations). The `emurst` has completed successfully if you see your command prompt after this message:

```
EMURST for XDS510WS loading <pathname-filename> at #  
where <pathname-filename> is the location of the r470510ws.out file,  
and # refers to the file /dev/rsd#a, which is associated with the emula-  
tor in the configuration file, EMULATOR. Also, you can be sure that  
emurst succeeded when only the first and second LEDs from the left  
are on.
```

- 2) The `-p` debugger option that you entered on the command line or in the `D_OPTIONS` environment variable specifies a different driver file from the one used by `emurst`. Remember, if you haven't specifically reset the driver file number to another number, the default is 4. Use the same `-p` option that you used when you executed `emurst`. (For more information on the `-p` option, see Section 4.4, *Step 3: Ensuring That the Emulator Supports the Debugger*, on page 4-6, for SPARCstations and Section 5.4, *Step 3: Ensuring That the Emulator Supports the Debugger*, on page 5-5, for HP workstations.)
- 3) The `-f` debugger option you specified on the command line or in the `D_OPTIONS` environment variable (where the default file specified by the `-f` option is `board.dat`) specifies a file that the debugger can't find.
  - If you didn't provide *any* path information with the filename, the debugger could not find the file in the current directory or in any of the directories listed in the `D_DIR` environment variable.
  - If you didn't provide the *correct* path information, reexecute the debugger, specifying the correct pathname and filename for the board configuration file.

- 4) One of these two problems could exist:
  - You didn't specify **-n tms470** debugger option.
  - The debugger couldn't find the **tms470** that you specified with the **-n** option in your board configuration file.

Re-execute the debugger with the **-n** debugger option, specifying the name of a '470 device from the board configuration file.

- 5) You may not have described your target system correctly in the board configuration file that you specified with the **-f** debugger option on the command line or in the **D\_OPTIONS** environment variable. Review your board configuration file and correctly describe the target system.
- 6) Make sure that your emulation cable is firmly attached both to the XDS510WS and to your target system.
- 7) Make sure that your target system is receiving sufficient power at the required voltage to allow all devices on the board to work properly.

- You receive the following message at the operating-system command line when trying to execute the `emu470` command:

```
emu470: display :0.0 doesn't know font 7x14
```

The default font file that the debugger uses (`7x14.ff`) couldn't be found by OpenWindows. OpenWindows searches for these font files in the directories specified in the **FONTPATH** environment variable. To correct the problem, do one of the following:

- Add the font file `7x14.ff` to a directory defined in the **FONTPATH** environment variable.
- Add to the `.Xdefaults` file in your home directory the line "`emu470*font: GoodFontName`", where *GoodFontName* is the name of a font that OpenWindows can find.
- Copy a valid font file onto `7x14.ff`.

## A.5 Additional Emulator and Debugger Problems

The operating-system window displays operating-system messages. These messages differ from the error messages that you may see in the COMMAND window of the debugger. If you receive one of these operating-system messages while executing emu470 or emurst under SunOS 4.1.x, refer to the following explanations.

These messages are status messages, not error messages.

`<date> <time> <hostname> vmunix: sd<n>: disk not responding to selection`

or under SunOS 5.x:

```
WARNING: /sbus@1,f8000000/esp@0,800000/sd@<n>,0(sd<n>):
        disk not responding to selection
```

The XDS510WS didn't respond to the SPARCstation in a certain amount of time. This can be caused by one of these conditions:

- The XDS510WS isn't powered.
- The XDS510WS is executing its self test.
- The XDS510WS is executing a lengthy debugger command such as a large memory fill.

`<date> <time> <hostname> vmunix: sd<n>: offline`

or under SunOS 5.x.

```
WARNING /sbus@1,f8000000/esp@0,800000/sd@<n>,0(sd<n>):
        offline
```

The SPARCstation is unable to select the XDS510WS after several attempts and therefore considers the emulator to be offline. This message can be generated during large memory-fill instructions and should *not* be considered an error by itself or in combination with the preceding message. The debugger automatically corrects for this situation unless a major error has taken place, in which case the debugger eventually returns an error message in the COMMAND window of the debugger.

❑ `<date> <time> <hostname> vmunix: sd<n>: disk okay`

or under SunOS 5.x:

```
WARNING: /sbus@1,f8000000/esp@0,800000/sd@<n>,0(sd<n>):
        disk okay
```

The SPARCstation has reconnected with the XDS510WS after the XDS510WS didn't respond to the selection. When the debugger recovers from the *offline* condition (described in the previous bulleted item), one of the two messages shown above is written to the operating-system window.

❑ `sd<n> at esp0 target <p> lun 0`  
`sd<n>: Vendor 'TI-ASP', product 'XDS510-WS_Rev.*', 130`  
`512 byte blocks`  
`<date> <time> <hostname> vmunix: sd<n>: corrupt label -`  
`wrong magic number`

If the emulator has been inactive on the bus since the SPARCstation's last attempt to access it, the XDS510WS returns to an active status on the bus. The above message informs you of this *new* SCSI device.

**Note:**

Since the SPARCstation interprets the emulator as a SCSI disk, the SPARCstation expects it to be formatted. When the SPARCstation first finds that the new device isn't formatted, it produces the corrupt label message.

# Installing the Emulator Device Driver Manually

---

---

---

You must install the emulator device driver (genport.sys) in the Windows NT registry to enable the debugger to communicate with the emulator. If you can't get regdrv installation program (described in Section 2.4, *Installing the Emulator Device Driver*, on page 2-5) to install and register the emulator device driver, you can install the driver manually by performing the following steps:

- 1) Copy the device driver file and invoke the registry editor.
- 2) Set up the genport directory.
- 3) Set up the values for the genport directory.
- 4) Set up the Parameters directory and values.

This appendix describes these steps.

<b>Topic</b>	<b>Page</b>
<b>B.1 Copying the Device Driver File and Invoking the Registry Editor . .</b>	<b>B-2</b>
<b>B.2 Setting Up the Genport Directory . . . . .</b>	<b>B-3</b>
<b>B.3 Setting Up the Values for the Genport Directory . . . . .</b>	<b>B-4</b>
<b>B.4 Setting Up the Parameters Directory and Values . . . . .</b>	<b>B-6</b>

## **B.1 Copying the Device Driver File and Invoking the Registry Editor**

The TMS470R1x Emulator CD-ROM includes a `genport.sys` file that contains information about the emulator device driver. You must copy this file to the Windows NT drivers directory. Once you do so, you can use the registry editor to register the `genport.sys` parameters, allowing the emulator and debugger to communicate.

- 1) Copy `genport.sys` to the Windows NT drivers directory. For example:

```
copy c:\emu470\genport.sys c:\winnt35\system32\drivers 
```

or

```
copy c:\emu470\genport.sys c:\windows\system32\drivers 
```

- 2) Invoke the registry editor:

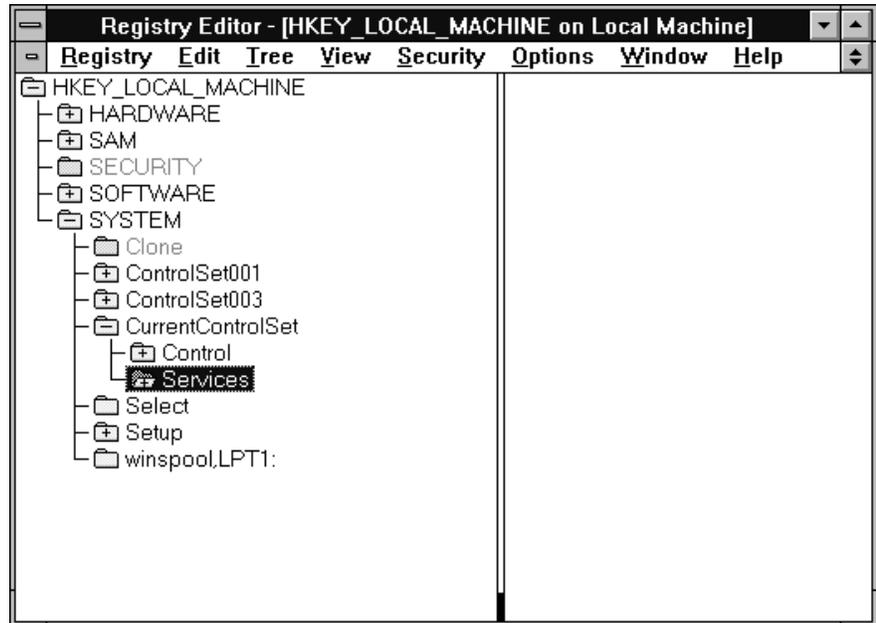
```
regedt32 
```

The registry editor brings up several windows.

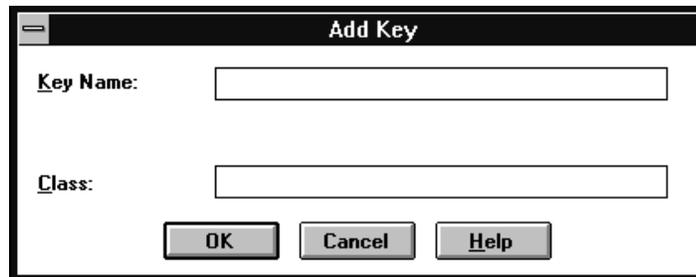
The remaining sections in this appendix tell you how to register the `genport.sys` parameters.

## B.2 Setting Up the Genport Directory

- 1) After you invoke the registry editor, locate the window called *HKEY\_LOCAL\_MACHINE on Local Machine* and select the *SYSTEM\CurrentControlSet\Services* directory icon.



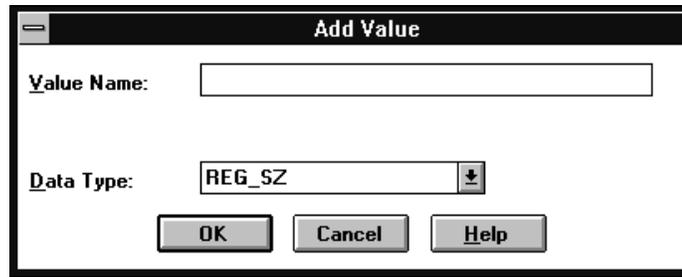
- 2) While the *Services* directory icon is highlighted, select *Edit* → *Add Key* from the menu bar. This displays the *Add Key* dialog box:



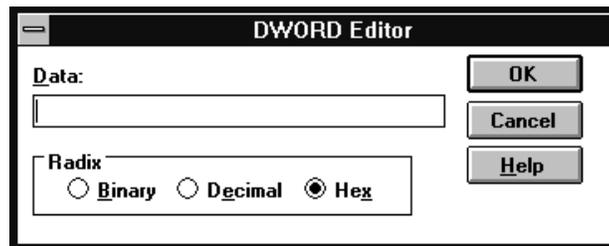
- 3) Enter *Genport* as the Key Name, then click on the *OK* button. This causes the *Genport* directory icon to appear under the *Services* directory (you may need to scroll the window to see the *Genport* directory).

### B.3 Setting Up the Values for the Genport Directory

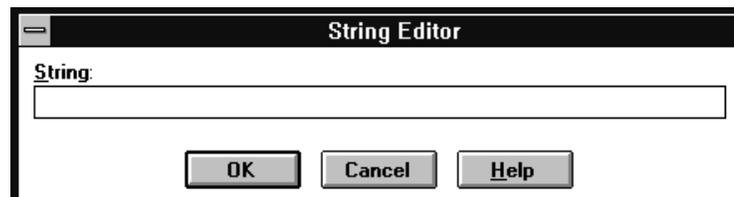
- 1) Click on the Genport directory icon to make it active (highlighted), then select Edit → Add Value from the menu bar. This displays the Add Value dialog box:



- 2) Enter *ErrorControl* as the Value Name, select *REG\_DWORD* as the Data Type, then click on the OK button. This displays the DWORD editor dialog box:



- 3) Enter *1* as the Data, select *Hex* as the Radix, then click on the OK button.
- 4) Click on the Genport directory icon to make it active (highlighted), then select Edit → Add Value from the menu bar. This displays the Add Value dialog box.
- 5) Enter *Group* as the Value Name, select *REG\_SZ* as the Data Type, then click on the OK button. This displays the String Editor dialog box:



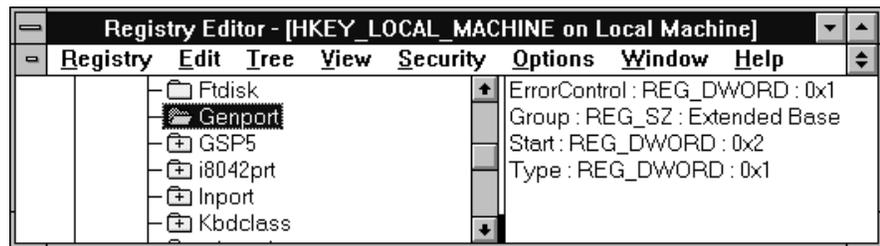
- 6) Enter *Extended Base* as the String, then click on the OK button.

7) Repeat steps 1–3 with the following information:

Value Name: Start  
Data Type: REG\_DWORD  
Data: 2  
Radix: Hex

Value Name: Type  
Data Type: REG\_DWORD  
Data: 1  
Radix: Hex

The following values should appear in the registry editor when the Genport directory icon is highlighted:



## B.4 Setting Up the Parameters Directory and Values

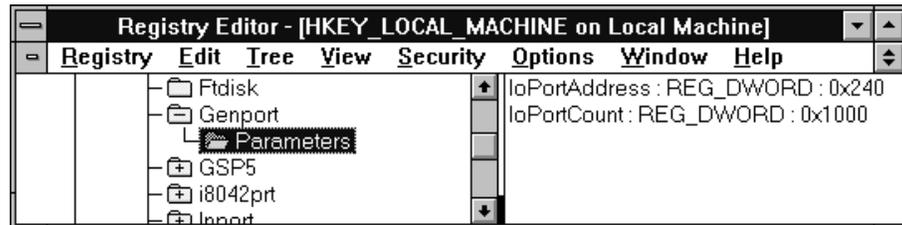
- 1) Click on the Genport directory icon to make it active, then select Edit → Add Key from the menu bar. This displays the Add Key dialog box.
- 2) Enter *Parameters* as the Key Name, then click on the OK button. This causes the Parameters directory icon to appear under the Genport directory.
- 3) Click on the Parameters directory to make it active, select Edit → Add Value from the menu bar, then enter the following information:

Value Name: IoPortAddress  
Data Type: REG\_DWORD  
Data: 240  
Radix: Hex

- 4) Repeat step 3 with the following information:

Value Name: IoPortCount  
Data Type: REG\_DWORD  
Data: 1000  
Radix: Hex

The following values should appear in the registry editor when the Parameters directory icon is highlighted:



- 5) Exit the registry editor, then reboot your PC. This completes the manual driver installation process.

*This template is for the “See” and “See also” references in your index. Since these entries do not have a page number associated with them, it’s extremely difficult to locate one if you need to modify or delete it and you don’t remember which chapter it’s in. By using this template, you can alphabetize your entries according to the first letter of the first level entry.*

**A**

**B**

**C**

**D**

**E**

**F**

**G**

**H**

**I**

**J**

**K**

**L**

**M**

**N**

**O**

**P**

**Q**

**R**

**S**

**T**

**U**

**V**

**W**

**X**

**Y**

**Z**

# Index

---

---

---

## A

- arrow keys
  - for HP workstations 5-11
  - for SPARCstations 4-12
- autoexec.bat file 1-8 to 1-10, 3-8 to 3-10

## B

- batch files
  - autoexec.bat 1-8 to 1-10, 3-8 to 3-10
  - board.cfg
    - for HP workstations 5-3
    - for SPARCstations 4-3
    - for Windows 3.1x 1-3
    - for Windows 95 3-3
    - for Windows NT 2-3
  - board.dat
    - for HP workstations 5-3
    - for SPARCstations 4-2
    - for Window 3.1x 1-3
    - for Window NT 2-3
    - for Windows 95 3-3
  - .cshrc
    - for HP workstations 5-7 to 5-9
    - for SPARCstations 4-8 to 4-10
  - emuinit.cmd
    - for HP workstations 5-3
    - for SPARCstations 4-3
    - for Windows 3.1x 1-3
    - for Windows 95 3-3
    - for Windows NT 2-3
  - emurst
    - for HP workstations 5-5
    - for SPARCstations 4-6
    - for Windows 3.1x 1-11
    - for Windows 95 3-11
    - for Windows NT 2-12

- batch files (continued)
  - initialization
    - for HP workstations 5-3
    - for SPARCstations 4-3
    - for Windows 3.1x 1-3
    - for Windows 95 3-3
    - for Windows NT 2-3
- board.cfg file
  - for HP workstations 5-3
  - for SPARCstations 4-3
  - for Windows 3.1x 1-3
  - for Windows 95 3-3
  - for Windows NT 2-3
- board.dat file
  - defined
    - for HP workstations 5-3
    - for SPARCstations 4-2
    - for Windows 3.1x 1-3
    - for Windows 95 3-3
    - for Windows NT 2-3
  - describing target system to debugger
    - for HP workstations 5-6
    - for SPARCstations 4-7
    - for Windows 3.1x 1-13
    - for Windows 95 3-13
    - for Windows NT 2-13
  - error messages
    - for Windows 3.1x 1-16
    - for Windows 95 3-16
    - for Windows NT 2-15
- booting, problems A-2

## C

- CD-ROM
  - mounting
    - for HP workstations 5-4
    - for SPARCstations 4-4
  - requirements
    - for HP workstations 5-2

CD-ROM (continued)  
  for SPARCstations 4-2  
  for Windows 3.1x 1-2  
  for Windows 95 3-2  
  for Windows NT 2-2  
retrieving files from  
  for HP workstations 5-5  
  for SPARCstations 4-5  
  for Windows 3.1x 1-4  
  for Windows 95 3-4  
  for Windows NT 2-4  
unmounting  
  for HP workstations 5-5  
  for SPARCstations 4-5  
chmod UNIX command A-3  
colors, mapping with X Windows  
  for HP workstations 5-12  
  for SPARCstations 4-13  
composer utility  
  for HP workstations 5-3  
  for SPARCstations 4-3  
  for Windows 3.1x 1-3  
  for Windows 95 3-3  
  for Windows NT 2-3  
configuration file, board.dat  
  for HP workstations 5-3, 5-6  
  for SPARCstations 4-2, 4-7  
  for Windows 3.1x 1-3, 1-13  
  for Windows 95 3-3, 3-13  
  for Windows NT 2-3, 2-13  
.cshrc file  
  for HP workstations 5-7  
  for SPARCstations 4-8 to 4-10  
invoking  
  for HP workstations 5-9  
  for SPARCstations 4-10  
customizing the display  
  for HP workstations 5-12  
  for SPARCstations 4-13

## D

D\_DIR environment variable  
  for HP workstations 5-7  
  for SPARCstations 4-8  
  for Windows 3.1x 1-8, 2-10  
D\_DIR environment variable (continued)  
  for Windows 95 3-8  
  for Windows NT 2-9  
D\_OPTIONS environment variable  
  for HP workstations 5-8  
  for SPARCstations 4-9  
  for Windows 3.1x 1-9  
  for Windows 95 3-9  
  for Windows NT 2-10  
  troubleshooting A-6  
D\_SRC environment variable  
  for HP workstations 5-7  
  for SPARCstations 4-8  
  for Windows 3.1x 1-9  
  for Windows 95 3-9  
  for Windows NT 2-10  
debugger  
  communicating with your target system  
    for HP workstations 5-6  
    for SPARCstations 4-7  
    for Windows 3.1x 1-13  
    for Windows 95 3-13  
    for Windows NT 2-13  
  displaying on a different machine  
    for HP workstations 5-9  
    for SPARCstations 4-10  
  environment setup  
    for HP workstations 5-7 to 5-9  
    for SPARCstations 4-8 to 4-10  
    for Windows 3.1x 1-8 to 1-10  
    for Windows 95 3-8 to 3-10  
    for Windows NT 2-9 to 2-11  
  font changes  
    for HP workstations 5-12  
    for SPARCstations 4-13  
  installation  
    for HP workstations 5-1 to 5-12  
    for SPARCstations 4-1 to 4-13  
    for Windows 3.1x 1-1 to 1-16  
    for Windows 95 3-1 to 3-16  
    for Windows NT 2-1 to 2-16  
    troubleshooting A-5 to A-6  
  invoking  
    for HP workstations 5-10  
    for SPARCstations 4-11  
    for Windows 3.1x 1-14  
    for Windows 95 3-14  
    for Windows NT 2-14

debugger (continued)  
   options  
     for HP workstations 5-8  
     for SPARCstations 4-9  
     for Windows 3.1x 1-9  
     for Windows 95 3-9  
     for Windows NT 2-10  
   troubleshooting A-5 to A-6, A-7 to A-8  
   using the X Window System  
     for HP workstations 5-11 to 5-12  
     for SPARCstations 4-12 to 4-13

default, memory map  
   for HP workstations 5-3  
   for SPARCstations 4-3  
   for Windows 3.1x 1-3  
   for Windows 95 3-3  
   for Windows NT 2-3

device driver, installation 2-5 to 2-6  
   manual installation B-1 to B-6

Diagnostic files  
   for Windows 3.1x  
     nscmode.exe 1-7  
     portchk.exe 1-7  
     smcmode.exe 1-7  
   for Windows 95  
     nscmode.exe 3-7  
     portchk.exe 3-7  
     smcmode.exe 3-7

directories  
   auxiliary files  
     for HP workstations 5-7  
     for SPARCstations 4-8  
     for Windows 3.1x 1-8, 2-10  
     for Windows 95 3-8  
     for Windows NT 2-9  
   debugger software  
     for HP workstations 5-5, 5-7  
     for SPARCstations 4-5, 4-8  
     for Windows 3.1x 1-8, 2-9  
     for Windows 95 3-8  
     for Windows NT 2-9  
   emu470 directory  
     for HP workstations 5-5, 5-7  
     for SPARCstations 4-5, 4-8  
     for Windows 3.1x 1-8  
     for Windows 95 3-8  
     for Windows NT 2-9  
   identifying additional source directories  
     for HP workstations 5-7

directories (continued)  
   for SPARCstations 4-8  
   for Windows 3.1x 1-9  
   for Windows 95 3-9  
   for Windows NT 2-10

display, font changes  
   for HP workstations 5-12  
   for SPARCstations 4-13

DISPLAY environment variable  
   for HP workstations 5-9  
   for SPARCstations 4-10

display requirements  
   for HP workstations 5-2  
   for SPARCstations 4-2  
   for Windows 3.1x 1-2  
   for Windows 95 3-2  
   for Windows NT 2-2

downloading code  
   for HP workstations 5-5  
   for SPARCstations 4-6

driver file, troubleshooting A-3, A-5

## E

emu470 command  
   for HP workstations 5-3  
   for SPARCstations 4-2  
   for Windows 3.1x 1-3  
   for Windows 95 3-3  
   for Windows NT 2-3  
   troubleshooting A-5 to A-6, A-7 to A-8  
   verifying the installation  
     for HP workstations 5-10  
     for SPARCstations 4-11  
     for Windows 3.1x 1-14  
     for Windows 95 3-14  
     for Windows NT 2-14

emu470 directory  
   for HP workstations 5-5, 5-7  
   for SPARCstations 4-5, 4-8  
   for Windows 3.1x 1-8  
   for Windows 95 3-8  
   for Windows NT 2-9

emuinit.cmd file  
   for HP workstations 5-3  
   for SPARCstations 4-3  
   for Windows 3.1x 1-3  
   for Windows 95 3-3  
   for Windows NT 2-3

## emulator

## additional tools

- for HP workstations* 5-3
- for SPARCstations* 4-2
- for Windows 3.1x* 1-3
- for Windows 95* 3-3
- for Windows NT* 2-3

## board.cfg file

- for HP workstations* 5-3
- for SPARCstations* 4-3
- for Windows 3.1x* 1-3
- for Windows 95* 3-3
- for Windows NT* 2-3

## board.dat file

- for HP workstations* 5-3
- for SPARCstations* 4-2
- for Windows 3.1x* 1-3
- for Windows 95* 3-3
- for Windows NT* 2-3

## booting problems A-2

## debugger environment

- for HP workstations* 5-7 to 5-9
- for Windows 3.1x* 1-8 to 1-10
- for Windows 95* 3-8 to 3-10
- for Windows NT* 2-9 to 2-11
- SPARCstation* 4-8 to 4-10

## debugger installation

- for HP workstations* 5-1 to 5-12
- for SPARCstations* 4-1 to 4-13
- for Windows 3.1x* 1-1 to 1-16
- for Windows 95* 3-1 to 3-16
- for Windows NT* 2-1 to 2-16

## driver file access A-3

## hardware requirements

- for HP workstations* 5-2
- for SPARCstations* 4-2
- for Windows 3.1x* 1-2
- for Windows 95* 3-2
- for Windows NT* 2-2

## host system

- for HP workstations* 5-2
- for SPARCstations* 4-2
- for Windows 3.1x* 1-2
- for Windows 95* 3-2
- for Windows NT* 2-2

installation. *See XDS51x Emulator Installation Guide*

## memory, default map

- for HP workstations* 5-3
- for SPARCstations* 4-3

## emulator (continued)

- for Windows 3.1x* 1-3
- for Windows 95* 3-3
- for Windows NT* 2-3

## operating system

- for HP workstations* 5-3
- for SPARCstations* 4-2
- for Windows 3.1x* 1-3
- for Windows 95* 3-3
- for Windows NT* 2-3

## resetting

- for Windows 3.1x* 1-11, 1-12
- for Windows 95* 3-11
- for Windows NT* 2-12
- problems* A-3 to A-4, A-7 to A-8

## software requirements

- for HP workstations* 5-3
- for SPARCstations* 4-2
- for Windows 3.1x* 1-3
- for Windows 95* 3-3
- for Windows NT* 2-3

## troubleshooting A-1 to A-8

## emulator controller, requirements

- for HP workstations* 5-2
- for SPARCstations* 4-2
- for Windows 3.1x* 1-2
- for Windows 95* 3-2
- for Windows NT* 2-2

## EMULATOR file, troubleshooting A-3

## emurst command

- for HP workstations* 5-3, 5-5
- for SPARCstations* 4-2, 4-6
- for Windows 3.1x* 1-3, 1-11
- for Windows 95* 3-3, 3-11
- for Windows NT* 2-3, 2-12
- specifying parameters A-3
- syntax
  - for HP workstations* 5-5
  - for SPARCstations* 4-6
- troubleshooting A-3 to A-4, A-7 to A-8
- when invoking the debugger A-5

## emurstpp command

- for Windows 3.1x* 1-3, 1-12
- for Windows 95* 3-3

## end key

- for HP workstations* 5-11
- for SPARCstations* 4-12

## environment variables

## D\_DIR

- for HP workstations 5-7
- for SPARCstations 4-8
- for Windows 3.1x 1-8, 2-10
- for Windows 95 3-8
- for Windows NT 2-9

## D\_OPTIONS

- for HP workstations 5-8
- for SPARCstations 4-9
- for Windows 3.1x 1-9
- for Windows 95 3-9
- for Windows NT 2-10

## D\_SRC

- for HP workstations 5-7
- for SPARCstations 4-8
- for Windows 3.1x 1-9
- for Windows 95 3-9
- for Windows NT 2-10

## DISPLAY

- for HP workstations 5-9
- for SPARCstations 4-10

## displaying the debugger on a different machine

- for HP workstations 5-9
- for SPARCstations 4-10

- for HP workstations 5-7 to 5-9
- for SPARCstations 4-8 to 4-10
- for Windows 3.1x 1-8 to 1-10
- for Windows 95 3-8 to 3-10
- for Windows NT 2-9 to 2-11

## error messages, installation

- for Windows 3.1x 1-15
- for Windows 95 3-15
- for Windows NT 2-15

**F**

-f debugger option, troubleshooting A-5

## font changes

- for HP workstations 5-12
- for SPARCstations 4-13

font file, troubleshooting A-6

## function key mapping

- for HP workstations 5-11
- for SPARCstations 4-12

**G**

## Genport directory

- entering values B-4 to B-5
- setting up B-3

**H**

## hardware checklist

- for HP workstations 5-2
- for SPARCstations 4-2
- for Windows 3.1x 1-2
- for Windows 95 3-2
- for Windows NT 2-2

hardware installation. *See XDS51x Emulator Installation Guide*

## home key

- for HP workstations 5-11
- for SPARCstations 4-12

## host system

- for HP workstations 5-2
- for SPARCstations 4-2
- for Windows 3.1x 1-2
- for Windows 95 3-2
- for Windows NT 2-2

## HP workstations

- CD-ROM requirements 5-2
- display requirements 5-2
- emulator controller requirements 5-2
- hardware requirements 5-2
- host system 5-2
- installation
  - debugger software* 5-4
  - emulator controller* 5-4
  - verifying* 5-10
- operating system 5-3
- setting up debugger environment 5-7 to 5-9
- software requirements 5-3
- target system 5-3

**I**

## initialization batch files, emuinit.cmd

- for HP workstations 5-3
- for SPARCstations 4-3
- for Windows 3.1x 1-3
- for Windows 95 3-3
- for Windows NT 2-3

insert key  
  for HP workstations 5-11  
  for SPARCstations 4-12

installation  
  debugger software  
    for HP workstations 5-4  
    for SPARCstations 4-4  
    for Windows 3.1x 1-4  
    for Windows 95 3-4  
    for Windows NT 2-4  
  device driver 2-5 to 2-6  
    manual installation B-1 to B-6  
  emulator. See XDS51x Emulator Installation Guide  
  emulator controller  
    for HP workstations 5-4  
    for SPARCstations 4-4  
    for Windows 3.1x 1-4  
    for Windows 95 3-4  
    for Windows NT 2-4  
  error messages  
    for Windows 3.1x 1-15  
    for Windows 95 3-15  
    for Windows NT 2-15  
  troubleshooting A-1 to A-8  
  verifying  
    for HP workstations 5-10  
    for SPARCstations 4-11  
    for Windows 3.1x 1-14  
    for Windows 95 3-14  
    for Windows NT 2-14

invoking  
  .cshrc file  
    for HP workstations 5-9  
    for SPARCstations 4-10  
  debugger  
    for HP workstations 5-10  
    for SPARCstations 4-11  
    for Windows 3.1x 1-14  
    for Windows 95 3-14  
    for Windows NT 2-14  
    troubleshooting A-5 to A-6

ipcrm UNIX command A-3

ipcs UNIX command A-3

IPCSEMAPHORE option A-3

Index-6

## K

keyboard, mapping keys  
  for HP workstations 5-11  
  for SPARCstations 4-12

keys, special keys with the X Window System  
  for HP workstations 5-11  
  for SPARCstations 4-12

keysym label  
  for HP workstations 5-11  
  for SPARCstations 4-12

## L

labels, keysym  
  for HP workstations 5-11  
  for SPARCstations 4-12

LED lights, after emurst, for SPARCstations 4-6, 5-5

## M

mapping keys for use with X Windows  
  for HP workstations 5-11  
  for SPARCstations 4-12

memory  
  default map  
    for HP workstations 5-3  
    for SPARCstations 4-3  
    for Windows 3.1x 1-3  
    for Windows 95 3-3  
    for Windows NT 2-3  
  requirements  
    for Windows 3.1x 1-2  
    for Windows 95 3-2  
    for Windows NT 2-2

memory mapping, emuinit.cmd file  
  for HP workstations 5-3  
  for SPARCstations 4-3  
  for Windows 3.1x 1-3  
  for Windows 95 3-3  
  for Windows NT 2-3

messages, installation errors  
  for Windows 3.1x 1-15  
  for Windows 95 3-15  
  for Windows NT 2-15

monochrome monitors, color mapping with X Windows  
 for HP workstations 5-12  
 for SPARCstations 4-13

mouse, requirements  
 for Windows 3.1x 1-2  
 for Windows 95 3-2  
 for Windows NT 2-2

## N

-n debugger option  
 for HP workstations 5-10  
 for SPARCstations 4-11  
 for Windows 3.1x 1-14  
 for Windows 95 3-14  
 for Windows NT 2-14  
 troubleshooting A-6

nscmode.exe  
 for Windows 3.1x 1-7  
 for Windows 95 3-7

## O

OpenWindows, finding the font file A-6

operating system  
 for HP workstations 5-3  
 for SPARCstations 4-2  
 for Windows 3.1x 1-3  
 for Windows 95 3-3  
 for Windows NT 2-3

optional files  
 for HP workstations 5-3  
 for SPARCstations 4-3  
 for Windows 3.1x 1-3  
 for Windows 95 3-3  
 for Windows NT 2-3

options, IPCSEMAPHORE A-3

## P

-p debugger option  
 selecting the driver file, troubleshooting A-3, A-5  
 using with emurst  
 for SPARCstations 4-6, 5-5  
 for Windows 3.1x 1-11  
 for Windows 95 3-11

-p debugger option (continued)  
 for Windows NT 2-12  
 with D\_OPTIONS environment variable  
 for Windows 3.1x 1-16  
 for Windows 95 3-16  
 for Windows NT 2-16

page-down key  
 for HP workstations 5-11  
 for SPARCstations 4-12

page-up key  
 for HP workstations 5-11  
 for SPARCstations 4-12

Parameters directory  
 entering values B-6  
 setting up B-6

path shell variable  
 for HP workstations 5-7  
 for SPARCstations 4-8

PATH statement  
 for Windows 3.1x 1-8, 2-9  
 for Windows 95 3-8  
 for Windows NT 2-9

permissions, root access  
 for HP workstations 5-3  
 for SPARCstations 4-2

port address  
 for Windows 3.1x 1-16  
 for Windows 95 3-16  
 for Windows NT 2-16

portchk.exe  
 for Windows 3.1x 1-7  
 for Windows 95 3-7

## R

r470510ws.out file  
 for HP workstations 5-3, 5-5  
 for SPARCstations 4-2, 4-6

RAM speed  
 for Windows 3.1x 1-2  
 for Windows 95 3-2

regedt32 file B-2

registry  
 editor, invoking B-2  
 environment variables, modifying 2-9  
 Genport directory  
 entering values B-4 to B-5  
 setting up B-3  
 installing the device driver 2-5, B-1 to B-6

- registry (continued)
  - Parameters directory
    - entering values* B-6
    - setting up* B-6
- required files
  - for HP workstations 5-3
  - for SPARCstations 4-2
  - for Windows 3.1x 1-3
  - for Windows 95 3-3
  - for Windows NT 2-3
- required tools
  - for HP workstations 5-3
  - for SPARCstations 4-2
  - for Windows 3.1x 1-3
  - for Windows 95 3-3
  - for Windows NT 2-3
- resetting
  - emurst command
    - for Windows 3.1x* 1-11
    - for Windows 95* 3-11
    - for Windows NT* 2-12
  - emurst file, troubleshooting A-3 to A-4, A-7 to A-8
  - emurstpp command, for Windows 3.1x 1-12
  - XDS510PC, emurst.exe command 1-3
  - XDS510PP, emurstpp.exe command 1-3
- retrieving files from CD-ROM
  - for HP workstations 5-5
  - for SPARCstations 4-5
  - for Windows 3.1x 1-4
  - for Windows 95 3-4
  - for Windows NT 2-4
- root privileges
  - for HP workstations 5-3
  - for SPARCstations 4-2

## S

- smcmode.exe
  - for Windows 3.1x 1-7
  - for Windows 95 3-7
- software checklist
  - for HP workstations 5-3
  - for SPARCstations 4-2
  - for Windows 3.1x 1-3
  - for Windows 95 3-3
  - for Windows NT 2-3
- SPARCstations
  - CD-ROM requirements 4-2

Index-8

- SPARCstations (continued)
  - display requirements 4-2
  - emulator controller requirements 4-2
  - hardware requirements 4-2
  - host system 4-2
  - installation
    - debugger software* 4-4
    - emulator controller* 4-4
    - verifying* 4-11
  - operating system 4-2
  - setting up debugger environment 4-8 to 4-10
  - software requirements 4-2
  - target system 4-2
- special keys, X Window System
  - for HP workstations 5-11
  - for SPARCstations 4-12

## T

- target system
  - board.dat file
    - for HP workstations* 5-3
    - for SPARCstations* 4-2
    - for Windows 3.1x* 1-3
    - for Windows 95* 3-3
    - for Windows NT* 2-3
  - describing to the debugger
    - for HP workstations* 5-3, 5-6
    - for SPARCstations* 4-3, 4-7
    - for Windows 3.1x* 1-3, 1-13
    - for Windows 95* 3-3, 3-13
    - for Windows NT* 2-3, 2-13
- troubleshooting A-1 to A-8
  - when booting workstation A-2
  - when invoking the debugger A-5 to A-6
  - when resetting emulator A-3 to A-4, A-7 to A-8
  - when using the debugger A-7 to A-8

## U

- utilities
  - xev
    - for HP workstations* 5-11
    - for SPARCstations* 4-12
  - xmodmap
    - for HP workstations* 5-11
    - for SPARCstations* 4-12
  - xrdb
    - for HP workstations* 5-12
    - for SPARCstations* 4-13

**V**

verifying, installation  
 for HP workstations 5-10  
 for SPARCstations 4-11  
 for Windows 3.1x 1-14  
 for Windows 95 3-14  
 for Windows NT 2-14  
 troubleshooting A-5

**W**

Windows 3.1x  
 CD-ROM requirements 1-2  
 display requirements 1-2  
 emulator controller requirements 1-2  
 hardware requirements 1-2  
 host system 1-2  
 installation  
   *debugger software* 1-4  
   *emulator controller* 1-4  
   *environment variables* 1-8 to 1-10  
   *error messages* 1-15  
   *verifying* 1-14  
 memory requirements 1-2  
 mouse requirements 1-2  
 operating system 1-3  
 resetting emulator 1-11 to 1-12  
 setting up debugger environment 1-8 to 1-10  
 software requirements 1-3

Windows 95  
 CD-ROM requirements 3-2  
 display requirements 3-2  
 emulator controller requirements 3-2  
 hardware requirements 3-2  
 host system 3-2  
 installation  
   *debugger software* 3-4  
   *emulator controller* 3-4  
   *environment variables* 3-8 to 3-10  
   *error messages* 3-15  
   *verifying* 3-14  
 memory requirements 3-2  
 mouse requirements 3-2  
 operating system 3-3  
 resetting emulator 3-11 to 3-12  
 setting up debugger environment 3-8 to 3-10  
 software requirements 3-3

Windows NT  
 CD-ROM requirements 2-2  
 display requirements 2-2  
 emulator controller requirements 2-2  
 hardware requirements 2-2  
 host system 2-2  
 installation  
   *debugger software* 2-4  
   *emulator controller* 2-4  
   *environment variables* 2-9 to 2-11  
   *error messages* 2-15  
   *verifying* 2-14  
 memory requirements 2-2  
 mouse requirements 2-2  
 operating system 2-3  
 registry 2-9  
 resetting emulator 2-12  
 setting up debugger environment 2-9 to 2-11  
 software requirements 2-3

workstation, problems when booting A-2

**X**

-x debugger option, using with emurst  
 for HP workstations 5-5  
 for SPARCstations 4-6  
 for Windows 3.1x 1-11  
 for Windows 95 3-11  
 for Windows NT 2-12

X Window System  
 color mapping  
   *for HP workstations* 5-12  
   *for SPARCstations* 4-13  
 displaying debugger on a different machine  
   *for HP workstations* 5-9  
   *for SPARCstations* 4-10  
 special keys  
   *for HP workstations* 5-11  
   *for SPARCstations* 4-12  
 using with the debugger  
   *for HP workstations* 5-11 to 5-12  
   *for SPARCstations* 4-12 to 4-13

.Xdefaults file  
 for HP workstations 5-12  
 for SPARCstations 4-13

xev utility  
 for HP workstations 5-11  
 for SPARCstations 4-12

*Index*

---

xmodmap utility

for HP workstations 5-11

for SPARCstations 4-12

xrdb utility

for HP workstations 5-12

for SPARCstations 4-13

## **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.