OS-9 for EST SBC8260 Board Guide

Version 3.2

www.radisys.com
World Headquarters
5445 NE Dawson Creek Drive • Hillsboro, OR
97124 USA
Phone: 503-615-1100 • Fax: 503-615-1121
Toll-Free: 800-950-0044
International Headquarters
Gebouw Flevopoort • Televisieweg 1A
NL-1322 AC • Almere, The Netherlands
Phone: 31 36 5365595 • Fax: 31 36 5365620
RadiSys Microware Communications Software Division, Inc.
1500 N.W. 118th Street
Des Moines, Iowa 50325
515-223-8000

Revision A
December 2001

Copyright and publication information

This manual reflects version 3.2 of Enhanced OS-9 for PowerPC.
Reproduction of this document, in part or whole, by any means, electrical, mechanical, magnetic, optical, chemical, manual, or otherwise is prohibited, without written permission from RadiSys Microware Communications Software Division, Inc.

Disclaimer

The information contained herein is believed to be accurate as of the date of publication. However, RadiSys Corporation will not be liable for any damages including indirect or consequential, from use of the OS-9 operating system, Microware-provided software, or reliance on the accuracy of this documentation. The information contained herein is subject to change without notice.

Reproduction notice

The software described in this document is intended to be used on a single computer system. RadiSys Corporation expressly prohibits any reproduction of the software on tape, disk, or any other medium except for backup purposes. Distribution of this software, in part or whole, to any other party or on any other system may constitute copyright infringements and misappropriation of trade secrets and confidential processes which are the property of RadiSys Corporation and/or other parties. Unauthorized distribution of software may cause damages far in excess of the value of the copies involved.

Table of Contents

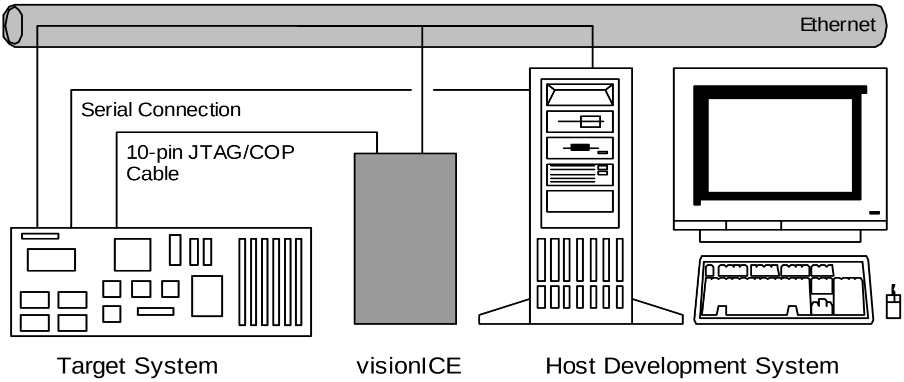
Chapter 1: Installing and Configuring OS-9

This chapter describes installing and configuring OS-9 on the EST SBC8260 board. It includes the following sections:

Development Environment Overview

Figure 1-2 shows a typical development environment for the SBC8260 board. The components shown include the minimum required to enable OS-9 to run on the PowerPC.

Figure 1-1 EST SBC8260 Development Environment



Requirements and Compatibility

Host Hardware Requirements (PC Compatible)

Your host PC must meet the following minimum requirements:

- Windows 95, 98, ME, 2000, or NT
- 300-400 MB of free disk space

  An additional 235MB of free disk space is required to run PersonalJava for OS-9. The SBC8260 Board Level Support Package requires about 100 MB of free disk space.
- 32MB of RAM (64MB recommended)
- Serial or Ethernet connection to the visionICE emulator.

Host Software Requirements (PC Compatible)

Your host PC must have the following applications:

- a terminal emulation program (such as `Hyperterminal` that comes with Microsoft Windows 95, Windows 98, and Windows NT 4.0)
- visionICE software from EST

Target Hardware Requirements

Your EST SBC8260 target system requires the following hardware:

- display terminal
- RS-232 serial connectors and cables

For More Information

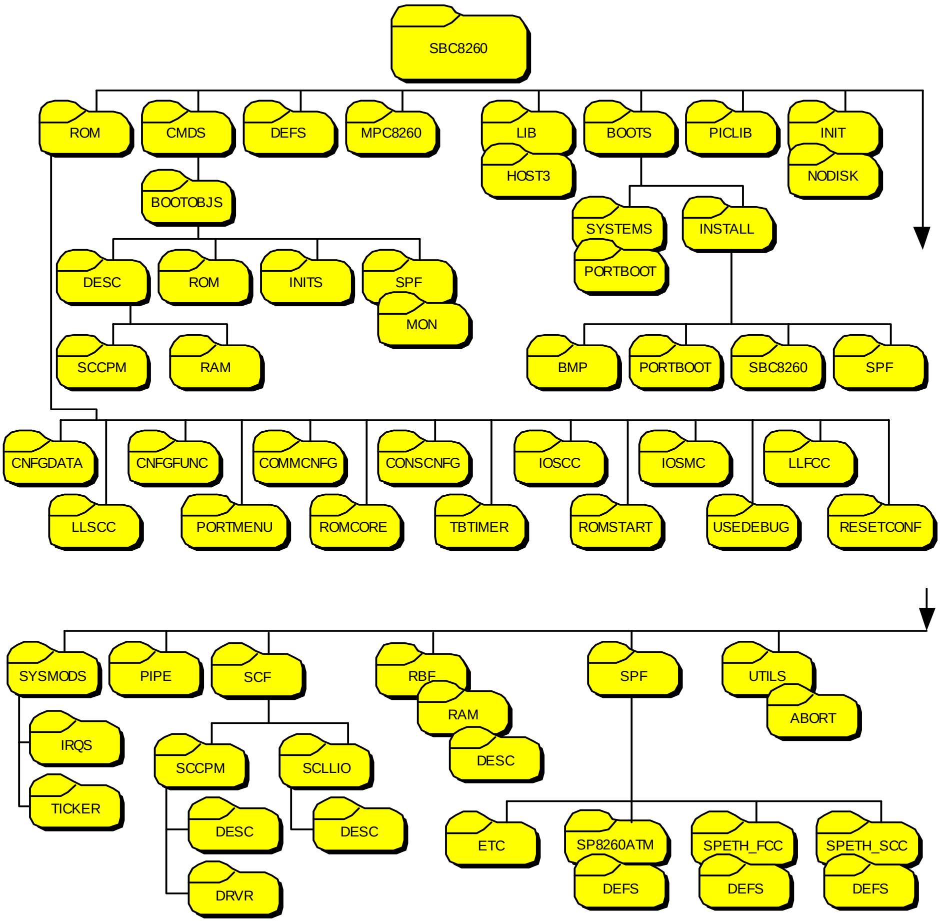
The ***OS-9 Device Descriptor and Configuration Module Reference*** manual included with your software distribution contains information to help you understand the purpose of each of the modules contained in this distribution and the variety of ways that the software can be configured to meet your needs.

OS-9 Architecture

The source and example code and makefiles for OS-9 for PowerPC are located in the following directory. The directory structure is shown in Figure 1-2.

```
/mwos/OS9000/8260/PORTS/SBC8260
```

Figure 1-2 PowerPC Directories



Connecting the Target to the Host

This section describes connecting the target board to the host PC via the visionICE (in-circuit emulator) device, serial, and Ethernet connections. It also describes using a terminal emulation program for the target.

Note

Your development system must have the following basic elements to complete this procedure:

- Ethernet connection from your Host to a network
- Ethernet connection from the visionICE device to a network
- 10-pin JTAG/COP cable connection from the visionICE device to the Target
- Serial connection between the Host PC and the Target
- a terminal emulation program (for example `Hyperterminal`)
- appropriate power supply to the Target and visionICE device

Overview

The OS-9 system software is installed on the SBC8260 board using the visionICE emulator. Before the software can be installed, the EST visionICE emulator must be able to communicate with the host PC and also must be connected to the SBC8260 board.

For More Information

See the Establishing Communications and Hardware Setup sections of the ***visionICE Getting Started*** manual for information about connecting the visionICE emulator to your PC and to the target.

Once loaded onto the SBC8260, the OS-9 system software uses the Com1 port on the SBC8260 as the system console. This section describes how to connect the SBC8260 Com1 port to your host PC.

You need a terminal emulation program (such as Hyperterminal) and the serial cable supplied with the visionICE to establish the connection between the host PC and the SBC8260 target machine.

Establishing the Serial Connection

Step 1. With the target system powered off, use the serial cable to connect the SBC8260 board's RS232 COM1 port to an unused RS-232 COM port on your host PC.

Step 2. Start Hyperterminal on the host PC by selecting `Start -> Programs -> Accessories -> Hyperterminal`.

Step 3. Enter a name for your Hyperterminal session.

Step 4. Select an icon for the new Hyperterminal session. A new icon will be created with the name of your session associated with it.

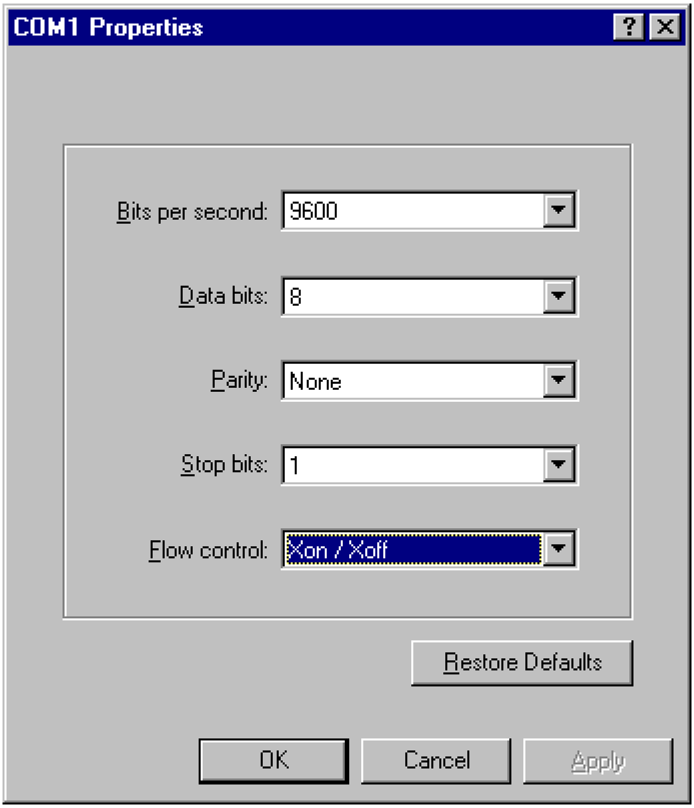
Step 5. Click `OK`.

Step 6. In the **Connect To** dialog box, go to the **Connect using** pull-down menu and enter the communications port to be used to connect to the target system.

Step 7. Click OK

Step 8. Configure the **Port Settings** tab, as shown in **Figure 1-3**.

Figure 1-3 COM Port Settings



Step 9. Click OK

Step 10. In the Hyperterminal window, select `File/Properties`. Click on the `Settings` tab and select the following:

`Terminal Keys`

`Emulation = Auto Detect`

`Backscroll Buffer Lines = 500`

Step 11. Click `OK`.

Step 12. Go to the Hyperterminal menu and select `Call/Connect` from the pull-down menu to establish your terminal session with the target. If you are connected, the bottom left corner of your Hyperterminal screen will display the word *connected*.

Step 13. Leave the Hyperterminal window open on your desktop (or minimized); you will use the window again later in this procedure.

Building the OS-9 ROM Image

The OS-9 ROM Image is a set of files and modules that collectively make up the OS-9 operating system. The specific ROM Image contents can vary from system to system depending on hardware capabilities and user requirements.

To simplify the process of loading and testing OS-9, the ROM Image is generally divided into two parts: the low-level image, called `coreboot`, and the high-level image, called `bootfile`.

Note

This section provides an example of an OS-9 ROM image successfully built on a host PC and transferred to an MTX603 target board. You may have to modify your selections depending on your application.

Coreboot

The coreboot image is generally responsible for initializing hardware devices and locating the high-level (or bootfile) image as specified by its configuration. For example from a FLASH part, a harddisk, or Ethernet. It is also responsible for building basic structures based on the image it finds and passing control to the kernel to bring up the OS-9 system.

Bootfile

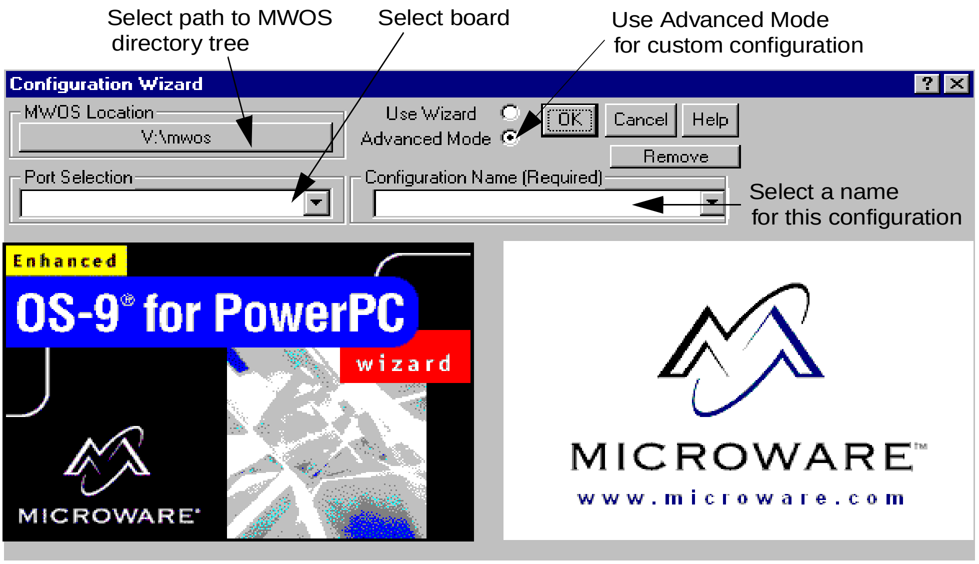
The bootfile image contains the kernel and other high-level modules (initialization module, file managers, drivers, descriptors, applications). The image is loaded into memory based on the device you select from the boot menu. The bootfile image normally brings up an OS-9 shell prompt, but can be configured to automatically start an application.

Using the Configuration Wizard

This section describes using the configuration wizard to build the OS-9 ROM image. To open the wizard, perform the following steps:

Step 1. On the Windows desktop, select `Start` --> `Programs` --> `Microware` -> `Enhanced OS-9 for PowerPC` --> `Microware Configuration Wizard`. You should see the following opening screen.

Figure 1-4 The Configuration Wizard Opening Window



Step 2. Select the path where the MWOS directory structure can be located by clicking in the MWOS location button.

Step 3. Select the target board from the **Port Selection** pull-down menu.

Step 4. Select a name for your configuration in the **Configuration Name** field. Your settings are saved. This enables you to modify the ROM image incrementally, without having to reselect every option for each change.

Step 5. Select `Advanced Mode` and click `OK`. The **Main Configuration** window is displayed. Advanced mode enables you to make more detailed and specific choices about what modules are included in your ROM image.

Creating and Configuring the ROM Image

This section describes how to use the Configuration Wizard to create and configure your OS-9 ROM image.

Note

The ROM Image created from this procedure expects a CPM frequency of 133 MHz and a Core Frequency of 200 MHz. To set these frequencies, the switches on DIP switch S2 are set as follows:

6 (open):	MODCK1 = 1
7 (closed):	MODCK2 = 0
8 (closed):	MODCK3 = 0

Note

The OS-9 ROM Image comprises two files—`coreboot` and `bootfile`. For the EST SBC8260 target board, these two files are combined and transferred from the host PC to the target board as one file (`rom`).

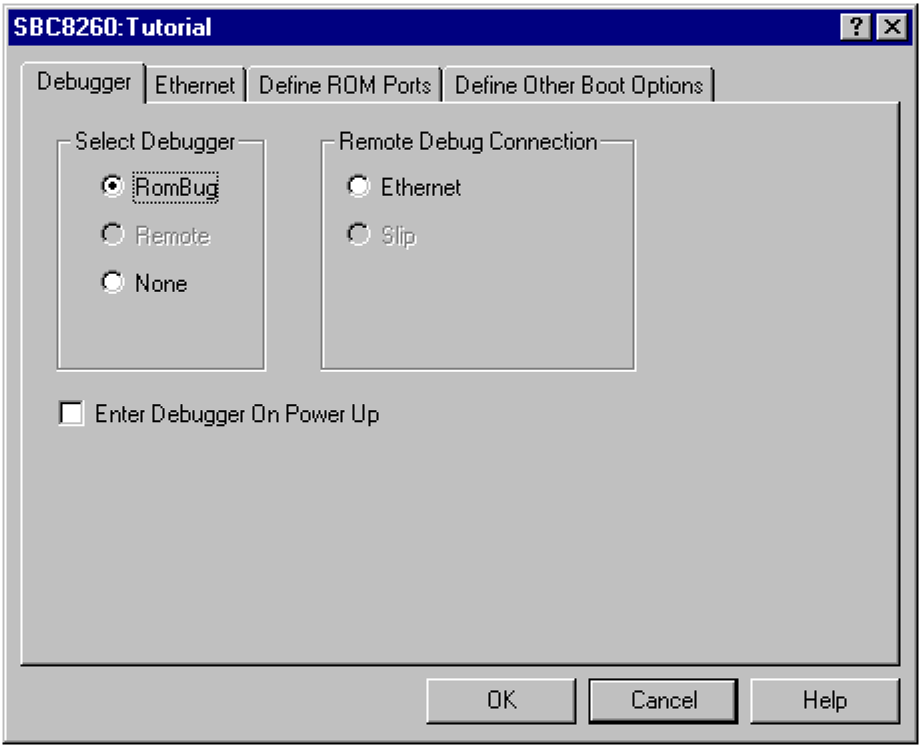
This section provides an example of an OS-9 ROM image successfully built on a host PC and transferred to an EST SBC8260 target board. You may have to modify your selections depending on your application.

Configure Coreboot Options

Step 1. From the **Main Configuration** window, select `Configure -> Coreboot -> Main configuration`.

Step 2. Select the **Debugger** tab. The following window is displayed.

Figure 1-5 Coreboot Configuration—Debugger Tab



Step 3. Under **Select Debugger**, select `RomBug`. This sets Ethernet as the method for user state debugging. Select `None` if you do not want to debug your program.

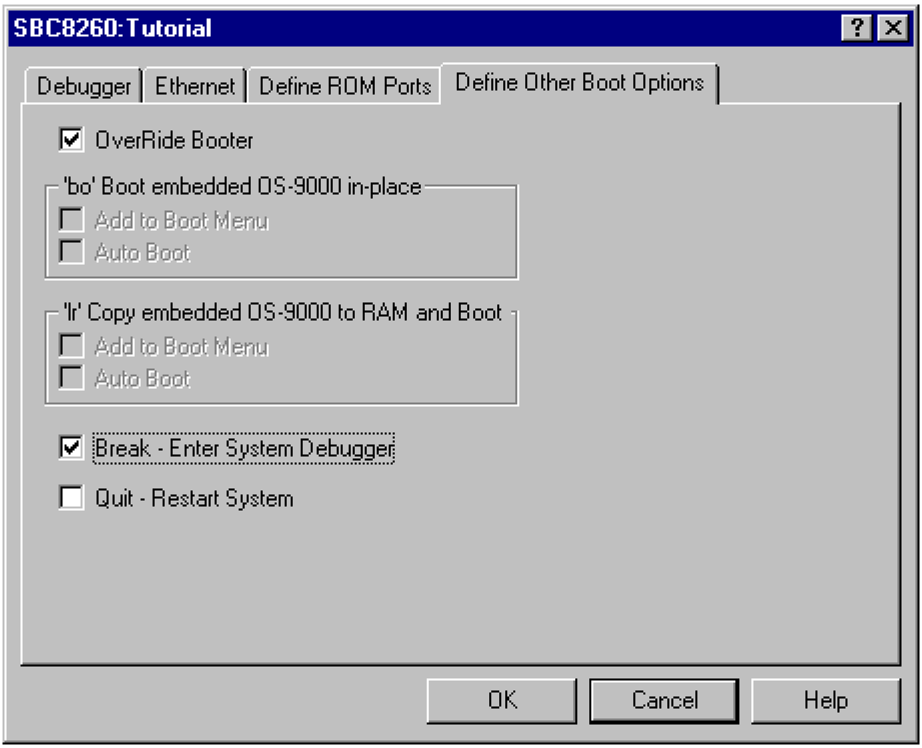
Note

To perform system state debugging, select `Ethernet` under Remote Debug Connection. If you set Ethernet as the method for system state debugging, you will not be able to perform user state debugging via Ethernet.

For system state debugging, you must also set the parameters in the **Ethernet** tab of the coreboot configuration.

Step 4. Click the **Define Other Boot Options** tab. The following window is displayed.

Figure 1-6 Coreboot Configuration—Define Other Boot Options Tab



Step 5. Select Break-Enter System Debugger radio button.

Note

Use the default settings for the **Ethernet** and **Define ROM Ports** tabs.

Step 6. Click OK and return to the **Main Configuration** window.

Configure System Options

When you select Configure -> Bootfile -> Configure System Options the **System Options** window appears. This window contains the **Define /term Port** tab and the **Bootfile Options** tab. Use the default settings for your selections.

Network Configuration

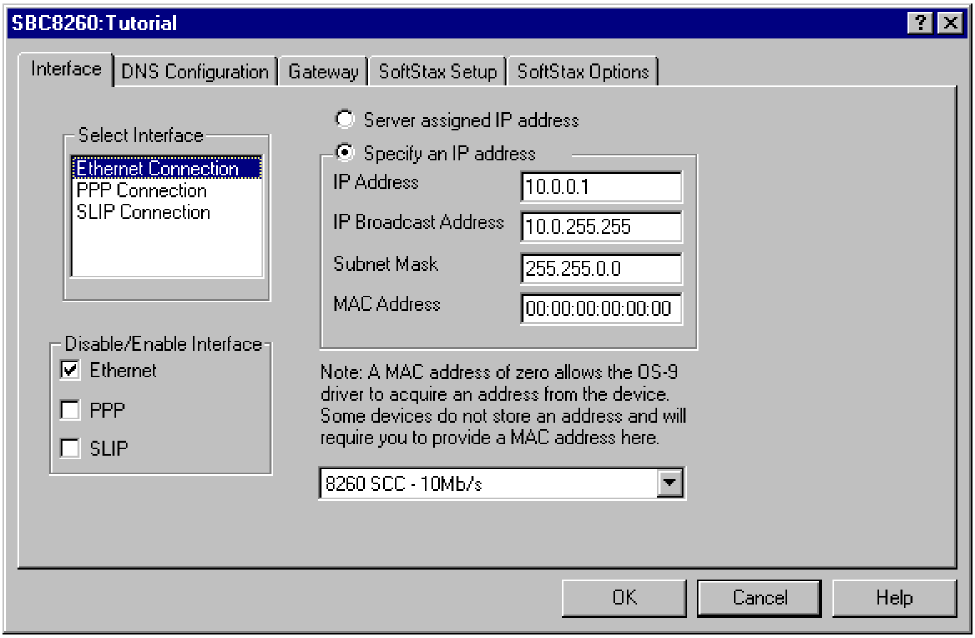
To use the target board across a network—once the target is booted—you must enable the Ethernet network settings. The **IP Address**, **DNS Configuration**, and **Gateway** tabs of the network configuration are similar to the **TCP/IP Properties** window in Windows.

To configure your network settings, complete the following steps:

Step 1. From the Main Configuration window, select `Configure -> Bootfile -> Network Configuration`.

Step 2. Select the **IP Address** tab. The following window is displayed.

Figure 1-7 Bootfile Configuration—IP Address Tab



Note

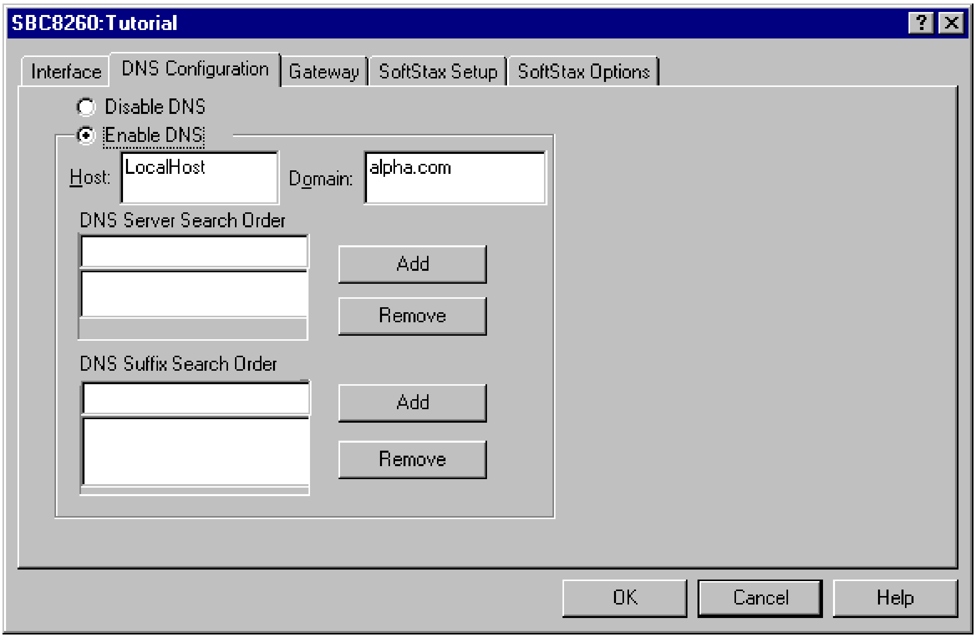
The addresses shown are for demonstration only. Contact your network administrator to obtain your IP Setup information. The SBC8260 board does not have a hardware assigned MAC address.

Note

You can select either 10Mb/s or 100Mb/s.

Step 3. Select the **DNS Configuration** tab. The following window is displayed. More than one DNS server can be added in this dialog box.

Figure 1-8 Bootfile Configuration—DNS Configuration Tab



If your network does not use DNS, click `Disable DNS`, and move to the **Gateway** tab.

If you have DNS available, click `Enable DNS` and type your host name and domain.

Note

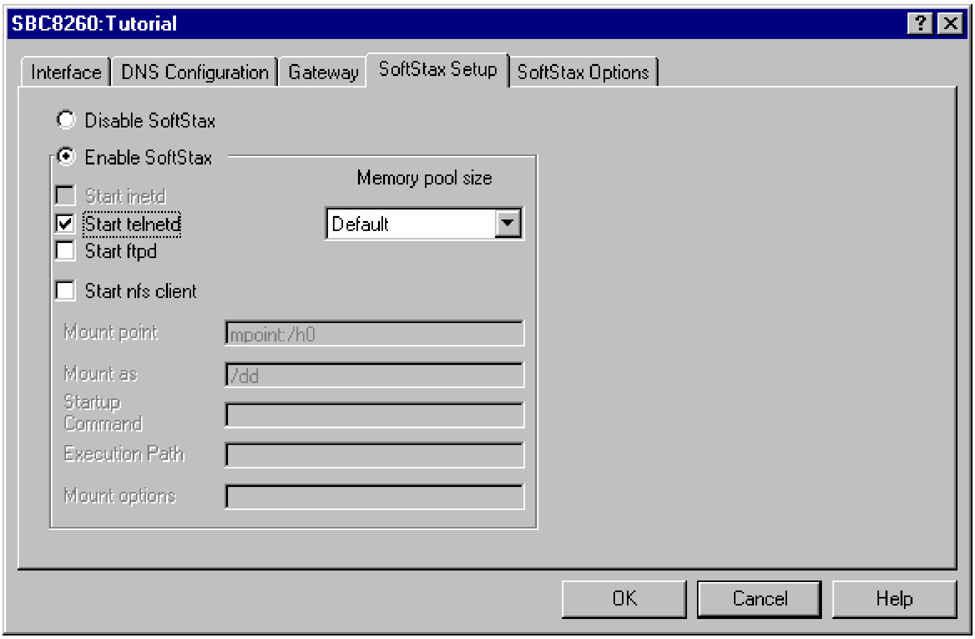
You add DNS IP addresses by clicking on the box directly under **DNS Server Search Order** and typing the IP address. Click the Add button when complete.

Step 4. Select the **Gateway** tab. You add new gateway addresses by clicking on the box and typing in the gateway name. Click the Add button when complete.

Step 5. Select the **SoftStax Setup** tab. The following window is displayed.

The options below represent daemons that can be automatically started if you want to FTP or telnet from a PC to the OS-9 target. **Start NFS Client** enables you to remote mount the target. For this demonstration, you will telnet to the target and establish a sender and a receiver window.

Figure 1-9 Bootfile Configuration—SoftStax Setup Tab



Step 6. Click `Enable SoftStax`.

Step 7. Click `Start telnetd`. (The only checked box on this tab should be the Start telnetd box.)

Step 8. Click `OK`.

Note
Use the default settings for the **SoftStax Options** tab.

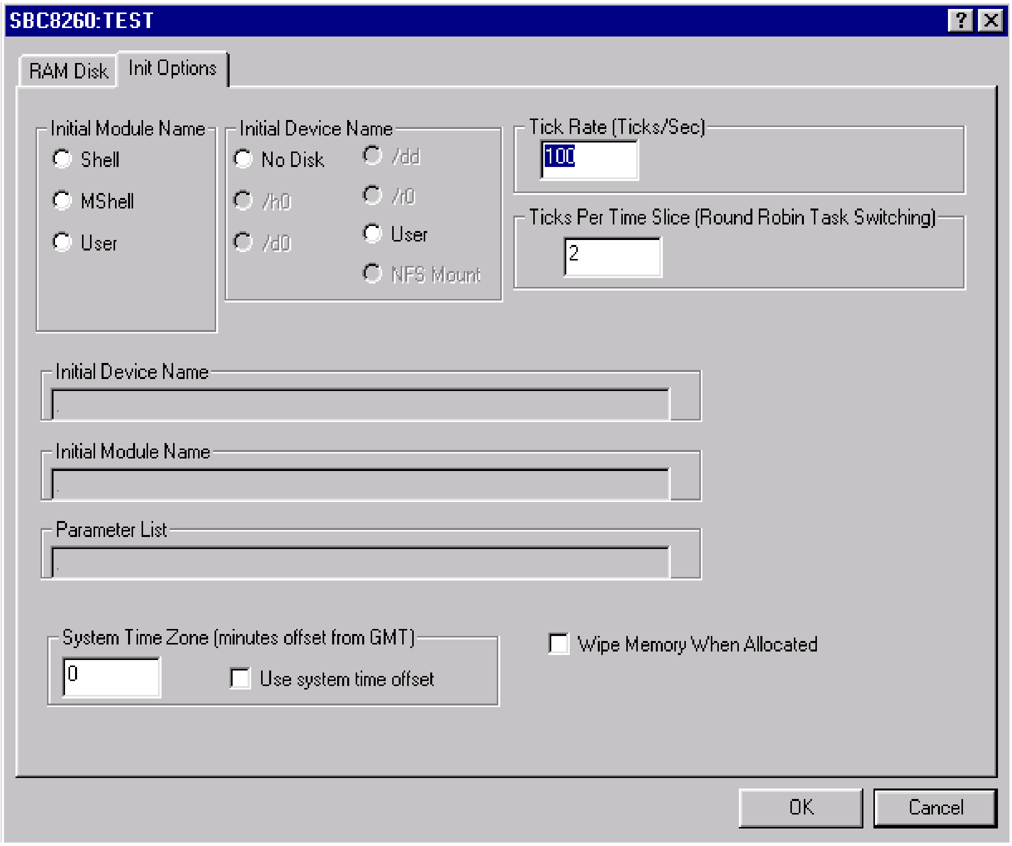
Disk Configuration

Step 1. From the **Main Configuration** window, select `Configure -> Bootfile -> Disk Configuration`.

Step 2. Select the **RAM Disk** tab. This enables you to create a RAM disk of any size for loading modules onto the target. This is an options setting.

Step 3. Select the **Init Options** tab. The following window is displayed. This sets the configuration for OS-9 to initialize itself on the target.

Figure 1-10 Bootfile Configuration—Init Options Tab



Select the `Mshell` option for the initial module name. This causes OS-9 to start a console shell usable from your terminal window. **Initial Device Name** should be selected as `No Disk`.

The tick rate is `100` and ticks per timeslice is set to 2. If you look at the Parameter List box, you see the commands that OS-9 executes upon system start-up.

Step 4. Click `OK` to return to the **Main Configuration** window.

Build Image

Complete the following steps to build the ROM Image for the target board.

Step 1. From the **Main Configuration** window, select `Configure -> Build Image`. The **Master Builder** window appears.

Step 2. Select the `Coreboot + Bootfile` option.

Step 3. Select the `ROM Utility Set`, `User State Debugging Modules`**,** and the `SoftStax (SPF) Support` boxes under the Include options.

Step 4. Click `Build`. It should display progress information and show the statistics of the image just created.

Step 5. The `rom` and `rom.s` files are created in the following directory:
`MWOS/OS9000/8260/PORTS/SBC8260/BOOTS/INSTALL/PORTBOOT`

Transferring the ROM Image to the Target

Complete the following steps to transfer your ROM image to the reference board.

For More Information

This process uses the EST Utilities Panel GUI. For more information about this software, refer to the ***visionICE User Manual***.

Step 1. Program the SBC8260 Flash. Following is the image to be loaded:

`MWOS/OS9000/8260/PORTS/SBC8260/BOOTS/INSTALL/PORTBOOT/rom.s.`

`rom.s` must be converted to the.BIN format. Use the following information to load the OS-9 ROM image on to the SBC8260 board:

- Programming Algorithm: `AMD 29F080 (1024 x 8) 4 Devices`
- Start Address value: `FE000000`
- End Address value: `FE3FFFFF`.

For More Information

The Flash memory on the SBC8260 is programmed as described in the Flash Programming chapter of the ***visionICE User Manual***.

Step 2. Boot the OS-9 system. This step can be completed with or without the visionICE emulator connected to the SBC8260.

To boot the system with the visionICE connected, use the command `go FFF00100` in the Terminal Window.

To boot the system without the visionICE connected, power off the SBC8260 board, remove the visionICE COM connector and power on the SBC8260.

In either case the system will display a boot menu on the console.

Note

The boot menu can have different selections, depending upon your selections using the configuration wizard.

Step 3. Type the bo command to select booting OS-9 in-place. Your screen should display the following:

```
OS-9000 Bootstrap for the PowerPC(tm)

Now trying to Override autobooters.

BOOTING PROCEDURES AVAILABLE ---------- <INPUT>

Boot embedded OS-9000 in-place -------- <bo>
Copy embedded OS-9000 to RAM and boot - <lr>
Enter ROM Debugger -------------------- <break>
Restart the System -------------------- <q>

Select a boot method from the above menu: bo

Now searching memory ($02840000 - $029fffff) for an
OS-9000 Kernel...

An OS-9000 kernel was found at $02840000
A valid OS-9000 bootfile was found.
+3
[1]$
```

Optional Procedures

Preliminary Testing

Once you have established an OS-9 prompt on your target system, you can perform the following procedures to test your system:

Step 1. Type `mdir` at the prompt.

`mdir` displays all the modules in memory. You may have to hit the space bar to scroll the output.

Step 2. Type `procs` at the prompt.

`procs` displays the processes currently running in the system.

Step 3. Test the networking on your system.

Select a host on the Ethernet network and run the `ping` utility. The following display shows a successful `ping` to a machine called solkanar.

```
$ ping solkanar
PING solkanar.microware.com (172.16.2.51): 56 data bytes
64 bytes from 172.16.2.51: ttl=128 time=0 ms
```

Step 4. Test `telnet`.

Select a host machine that allows telnet access and try the OS-9 `telnet` utility. The following display shows a successful `telnet` to a machine called delta.

```
$ telnet delta
Trying 172.16.1.40...Connected to delta.microware.com.
Escape character is '^]'.
capture closed.
```

```
OS-9/68K V3.0.3 Delta VME177 - 68060 98/12/24 14:41:51
User name?: curt
Password:
Process #101 logged on 98/12/24 14:41:56
Welcome!
***********************************************************
*      WELCOME TO DELTA - THE :OS-9 68K: MACHINE *
```

Step 5. Test telnet from your host PC to the reference board.

From the Windows Start menu, select `Run` and type `telnet` <`hostname`> and click `OK`. A telnet window should display with a $ prompt. Type `mdir` from the prompt. You should see the same module listing as on the serial console port.

You have now created your OS-9 ROM image, loaded the ROM image to the target, and established network connectivity with the target.

Chapter 2: Board Specific Reference

This chapter contains information that is specific to the EST SBC8260 reference boards. The following sections are included:

- **Boot Menu Options**
- **OS-9 Vector Mapping**

Boot Menu Options

You select your boot device menu options using the Configuration Wizard. For each boot device option, you can select whether you want it to be displayed on a boot menu, set up to autoboot, or both. The autoboot option enables the device selected to automatically boot up the high-level bootfile, bypassing the boot device menu.

Note

When using the Configuration Wizard, you should select only one device for autoboot on your system.

Following is an example of the Boot menu displayed in the terminal emulation window (using Hyperterminal):

```
OS-9000 Bootstrap for the PowerPC(tm)

Now trying to Override autobooters.

BOOTING PROCEDURES AVAILABLE ----- <INPUT>

Scan SCSI devices ---------------- <ioi>
Boot FDC floppy ------------------ <fd>
Boot from PC-Floppy -------------- <pf>
Boot from Teac SCSI floppy drive - <fs>
Boot from SCSI PC-Floppy --------- <pfs>
Boot from Viper tape drive ------- <vs>
Boot over Ethernet --------------- <eb>
Boot from SCSI(SCCS) hard drive -- <hs>
Boot embedded OS-9000 in-place --- <bo>
Enter system debugger ------------ <break>
Restart the System --------------- <q>

Select a boot method from the above menu:
```

What you select for boot options in the Configuration Wizard determines what modules are included in the coreboot image. **Table 2-1** lists some of the supported boot devices for OS-9:

Table 2-1 Supported Boot Methods

Type of Boot	Description
Boot from RBF hard disk	Boot from a standard SCSI hard disk (`hs`).
Floppy Disk	Boot from floppy disk. You must select if the floppy is controlled by a Random Block File System (RBF) (`fd` or `fs`) or PC File System (`pf` or `pfs`).
Boot embedded OS-9 in-place	Boot OS-9 from FLASH (`bo`).
Copy embedded OS-9 to RAM and Boot	Copy OS-9 from FLASH (if stored there) to RAM and boot (`lr`).
Boot using bootp over Ethernet	OS-9 is downloaded via TFTP from a server system.

OS-9 Vector Mapping

This section contains the vector mappings and dual-port RAM mappings for the MPC8260 processors.

The system module `pq2irq` maps interrupts coming from the MPC8260 into the OS-9 vector table according to the following mappings.

Interrup vectors are mapped starting at vector 0x40 in the order shown in Table 4-3 of the ***MPC8260 PowerQuicc II^{tm} User's Manual,*** and as shown in the following table.

Table 2-2 System Interface Unit Vectors

Vector	Source
0x40	Error
0x41	I2C
0x42	SPI
0x43	RISC Timer
0x44	SMC1
0x45	SMC2
0x46	IDMA1
0x47	IDMA2
0x48	IDMA3
0x49	IDMA4
0x4a	SDMA

Table 2-2 System Interface Unit Vectors (continued)

Vector	Source
0x4b	Reserved
0x4c	Timer1
0x4d	Timer2
0x4e	Timer3
0x4f	Timer4
0x50	TMCNT
0x51	PIT
0x52	PCI
0x53	IRQ1
0x54	IRQ2
0x55	IRQ3
0x56	IRQ4
0x57	IRQ5
0x58	IRQ6
0x59	IRQ7
0x5a	Reserved
0x5b	Reserved

Table 2-2 System Interface Unit Vectors (continued)

Vector	Source
0x5c	Reserved
0x5d	Reserved
0x5e	Reserved
0x5f	Reserved
0x60	FCC1
0x61	FCC2
0x62	FCC3
0x63	Reserved
0x64	MCC1
0x65	MCC2
0x66	Reserved
0x67	Reserved
0x68	SCC1
0x69	SCC2
0x6a	SCC3
0x6b	SCC4
0x6c	Reserved

Table 2-2 System Interface Unit Vectors (continued)

Vector	Source
0x6d	Reserved
0x6e	Reserved
0x6f	Reserved
0x70	PC15
0x71	PC14
0x72	PC13
0x73	PC12
0x74	PC11
0x75	PC10
0x76	PC9
0x77	PC8
0x78	PC7
0x79	PC6
0x7a	PC5
0x7b	PC4
0x7c	PC3
0x7d	PC2

Table 2-2 System Interface Unit Vectors (continued)

Vector	Source
0x7e	PC1
0x7f	PC0

Dual-port RAM Mapping

The SBC8260 processors include 64K bytes of dual-port RAM for buffer descriptor and microcode use. Since the high- and low-level drivers both use this area and must agree on their usage of it, the following locations have been reserved for the following uses:

Table 2-3 Dual Port RAM Use Map

Offset into DPRAM	Use
0x0 - 0x7f	SCC1 Ethernet
0x80-0xbf	SMC1 Parameter RAM
0xc0-0xff	SMC2 Parameter RAM
0x110 - 0x11f	SCC2
0x120 - 0x12f	SCC3
0x130 - 0x13f	SCC4
0x140 - 0x14f	SMC1
0x150 - 0x15f	SMC2
0x160 - 0x1ff	reserved

Table 2-3 Dual Port RAM Use Map (continued)

Offset into DPRAM	Use
0x200 - 0x3ff	FCC2 Ethernet
0x400 - 0x2fff	FCC1 ATM

Appendix A: Board Specific Modules

This appendix contains lists of high and low-level modules. the following sections are included:

- **Low-Level System Modules**
- **High-Level System Modules**
- **Common System Modules List**

Low-Level System Modules

The following low-level system modules are tailored specifically for the EST SBC8260 target platform. These modules can be found in the following directory:

`MWOS/OS9000/8260/PORTS/SBC8260/CMDS/BOOTOBJS/ROM`

Configuration Modules

`cnfgdata`	provides low-level configuration data including configuration of a serial console.
`cnfgfunc`	retrieves configuration parameters from the `cnfgdata` module.
`commcnfg`	retrieves the name of the low-level auxiliary communication port driver from the `cnfgdata` module.
`conscnfg`	retrieves the name of the low-level console driver from the `cnfgdata` module.
`resetconf`	contains the Hard Reset Configuration Word. See the ***MPC8260 PowerQUICC II User's Manual*** for details.

Console Drivers

`iosmc`	provides console services for the SMC UART on the MPC8260.

Debugging Modules

`usedebug`	is a debugger configuration module.

Ethernet Driver

`llf8260`	provides network driver services for the Ethernet port.
`lls8260`	provides network driver services for the Ethernet port.

System Modules

`portmenu`	retrieves a list of configured booter names from the ROM `cnfgdata` module.
`romcore`	provides bootstrap code.
`romstart`	provides the PowerPC exception vectors.

Timer Modules

`tbtimer`	provides polling timer services using the `tblo` and `tbhi` registers in the MPC8260 processor.

High-Level System Modules

The following OS-9 system modules are tailored specifically for your SBC8260 platform. Unless otherwise specified, each module can be found in a file of the same name in the following directory:

`<MWOS>/OS9000/8260/PORTS/SBC8260/CMDS/BOOTOBJS`

Ticker

`tk821pit`	provides the system ticker based on the SIU periodic interrupt timer.
`tkcpm`	provides the system ticker based on the CPM general purpose timer.
`tkdec`	provides the system ticker based on the PowerPC decrementer.

Abort Handler

`abort`	provides handler for the abort interrupt which calls into the system-state debugger. If no system state debugger is configured, the system performs a soft reset.

Shared Libraries

`picsub`	provides interrupt enable and disable routines to handle platform specific interrupt controller issues for device drivers. This module is called by all drivers, and should be included in your bootfile.

Serial and Console Drivers

`sccpm`	provides support for the CPM SMC and SCC UARTS serial port. The descriptors provided for this driver are named `term`, `t1`, `t2`, `t3`, `t4`, and `t5`, and are located in the following directory: `<MWOS>/OS9000/8260/PORTS/SBC8260/CMDS/BOOTOBJS/DESC/SCCPM`

Note

Only the SMC ports are wired for RS232.

`scllio`	is a configurable driver whose descriptor is built into the `PORTS` directory.

Common System Modules List

The following low-level system modules provide generic services for OS9000 modular ROM. They are located in the following directory:

`MWOS/OS9000/PPC/CMDS/BOOTOBJS/ROM`

Table 2-4 Common System Modules List

Module	Description
`bootsys`	provides booter services.
`console`	provides high-level I/O hooks into low-level console serial driver.
`dbgentry`	provides hooks to low-level debugger server.
`dbgserv`	is a debugger server module.
`excption`	is a service module.
`fdc765`	provides PC style floppy support.
`fdman`	is a target-independent booter support module providing general booting services for RBF file systems.
`flboot`	is a SCSI floptical drive disk booter.
`flshcach`	provides the cache flushing routine.
`fsboot`	is a SCSI TEAC floppy disk drive booter.
`hlproto`	allows user-state debugging.
`hsboot`	is a SCSI hard disk drive booter.

Table 2-4 Common System Modules List (continued)

Module	Description
`ide`	provides target-specific standard IDE support, including PCMCIA ATA PC cards.
`iovcons`	is a hardware independent virtual console driver that provides a telnetd-like interface to the low-level system console.
`llbootp`	is a target-independent BOOTP protocol booter module.
`llip`	is a target-independent internet protocol module.
`llkermit`	is a kermit booter (serial down loader).
`llslip`	is a target-independent serial line internet protocol module. This modules uses the auxiliary communications port driver to perform serial I/O
`lltcp`	is a target-independent transmission control protocol module.
`lludp`	is a target-independent user datagram protocol modules.
`notify`	coordinates use of low-level I/O drivers in system and user-state debugging.
`override`	enables overriding of the autobooter. If the space bar is pressed within three seconds after booting the target, a boot menu is displayed. Otherwise, booting proceeds with the first autobooter.

Table 2-4 Common System Modules List (continued)

Module	Description
`parser`	parses key fields from the `cnfgdata` module and the user parameter fields.
`pcman`	is a target-independent booter support module providing general booting services for PCF file systems (PC FAT file systems).
`protoman`	is a target-independent protocol module manager. This module provides the initial communication entry points into the protocol module stack.
`restart`	restarts boot process.
`romboot`	locates the OS-9 bootfile in ROM, FLASH, NVRAM.
`rombreak`	enables break option from the boot menu.
`rombug`	is a debugger client module.
`scsiman`	is a target-independent booter support module that provides general SCSI command protocol services
`sndp`	is a target-independent system-state network debugging protocol module. This module acts as a debugging client on the target, invoking the services of dbgserv to perform debug tasks.
`srecord`	receives a Motorola S-record format file from the communications port and loads it into memory.
`swtimer`	is a software timer.

Table 2-4 Common System Modules List (continued)

Module	Description
`tsboot`	is a SCSI TEAC tape driver booter.
`type41`	is a primary partition type.
`vcons`	provides the console terminal pathlist.
`vsboot`	is a SCSI archive viper tape drive booter.

Product Discrepancy Report

To: Microware Customer Support

FAX: 515-224-1352

From:____

Company:_____________________________________

Phone:_______________________________________

Fax:____________________Email:_________________

Product Name:

Description of Problem:

____

____

____

____

____

____

____

____

____

____

Host Platform____________________________________

Target Platform__________________________________

RadiSys.

MICROWARE SOFTWARE