



# **OS-9 for STPC Board Guide**

## **Version 3.2**

[www.radisys.com](http://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 Microwave 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 X86.

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.

---

December 2001  
Copyright ©2001 by RadiSys Corporation.  
All rights reserved.

EPC, INtime, iRMX, MultiPro, RadiSys, The Inside Advantage, and ValuPro are registered trademarks of RadiSys Corporation. ASM, Brahma, DAL, DAQ, MultiPro, SAIB, Spirit, and ValuePro are trademarks of RadiSys Corporation.

DAVID, MAUI, OS-9, and OS-9000, are registered trademarks of RadiSys Microware Communications Software Division, Inc. FasTrak, Hawk, SoftStax, and UpLink are trademarks of RadiSys Microware Communications Software Division, Inc.

† All other trademarks, registered trademarks, service marks, and trade names are the property of their respective owners.

---

# Table of Contents

---

## Chapter 1: Installing and Configuring OS-9

9

---

10	Development Environment Overview
11	Requirements and Compatibility
11	Host Hardware Requirements (PC Compatible)
11	Host Software Requirements (PC Compatible)
12	Target Hardware Requirements
13	Target Hardware Setup
13	Supported Devices
13	Ethernet Controllers
14	Maui VGA Support
15	Sequential Device Support
15	Physical Disk Media
15	System Devices
16	Additional Devices
16	SIMM Modules
16	CMOS Settings
17	Connecting the Target to the Host
18	Building the Bootfile Image
18	Using the Configuration Wizard
20	Building a Bootable Floppy Disk
25	Building an OS-9 Image on the Target
25	Preparing the Hard Disk
25	Partitioning the Drive
27	Formatting the Hard Drive
28	FTP to the Target
30	Configuring the Hard Drive
31	Creating a Startup File
32	Example Startup File

33	Optional Procedures
33	Advanced Configurations

## **Chapter 2: Board Specific Reference** **43**

---

44	MAUI Graphics Support
44	Getting Started
44	Configuring the Display

## **Appendix A: Board Specific Modules** **47**

---

48	Low-Level System Modules
49	High-Level System Modules
49	MAUI Support
49	PersonalJava™ Support

## **Appendix B: Configuring Hardware Devices** **51**

---

52	Ethernet Controllers
52	3COM PCI
52	Default Settings
53	Solving Configuration Issues
56	3COM ISA
56	System State Debugging
56	Default Settings
57	Solving Configuration Issues
60	3COM PCMCIA
60	System State Debugging
61	Default Settings
61	Solving Configuration Issues
64	DEC 21140
64	System State Debugging
64	Default Settings
65	Solving Configuration Issues
67	AM79C961 & AM79C73A

67	System State Debugging
67	Default Settings
67	Solving Configuration Issues
68	NE2000
68	System State Debugging
68	Default Settings
69	Board Setup Issues
69	PCI Specific Settings Information
73	NE2000 PCMCIA
73	Cirrus Logic CS8900
73	Hardware Configuration
74	OS9 Software Configuration
76	NetGear
77	LAN
78	RealTek
78	SMC
78	INTEL Ethernet PRO100 Series
79	Sequential Device Support
79	VGA Graphics / Keyboard
80	Language Support Options
81	Serial Mouse
81	PS2 Mouse
82	16550 Serial
83	Making the Descriptors
83	Digiboard
84	Making the Descriptors
84	Hostessl
85	Making the Descriptors
85	Risicom
86	Making the Descriptors
86	Parallel Printer
87	Making the Descriptors
88	Physical Disk Media

88	IDE Standard
88	Benefits
91	Using IDE in PCI Mode
91	RBF
92	PCF
93	Special Note
93	Descriptors
94	ROM BOOTING
94	Advanced Notes
94	PCMCIA IDE
95	Benefits
98	RBF
98	PCF
99	Special Note
100	Descriptors
100	ROM BOOTING
101	Advanced Notes
101	IDE Descriptors
101	Standard IDE - RBF Descriptors
102	Standard IDE - PCF Descriptors
103	CDROM IDE Descriptors
103	PCMCIA IDE - RBF Descriptors
103	PCMCIA IDE - PCF Descriptors
104	DiskOnChip
104	Benefits
104	Low-Level Boot Support
104	High-Level Support
105	Descriptors Used by Wizard
105	Building a DiskOnChip Boot Image
106	Bootting OS-9 and Formatting DiskOnChip
106	Creating the OS-9 Partition
107	Formatting the Partition
107	Installing Boot Files on DiskOnChip

107	Booting OS-9 from the DiskOnChip
107	Updating the DiskOnChip Firmware
109	Booting OS-9 from DiskOnChip
109	DiskOnChip Descriptors
109	DiskOnChip - RBF Descriptors
110	DiskOnChip - PCF Descriptors
111	PCAT Style Floppy
111	Floppy Descriptors
111	Floppy - RBF Descriptors
111	Floppy - PCF Descriptors
112	Symbios 810,810A,825,825A and 875 PCI SCSI controllers—Wide, Ultra and Ultra Wide
112	Benefits
117	Controller Dependency
117	Device Descriptors
117	Using Multiple SCSI Controllers
118	Creating Driver-Specific Versions
118	Diamond FirePort20 and FirePort40—Wide, Ultra and Ultra Wide
118	Benefits
119	Additional Notes
122	Controller Dependency
123	Device Descriptors
123	Using Multiple SCSI Controllers
124	Creating Driver-Specific Versions
124	Adaptec 1540/1542 ISA
125	Adaptec 2940, 2940U and 2940UW
125	SCSI HARD - RBF Descriptors
125	SCSI HARD - PCF Descriptors
125	SCSI FLOPPY - RBF Descriptors
125	SCSI FLOPPY - PCF Descriptors
125	SCSI TAPE Descriptors
125	SCSI CDROM Descriptors
127	System Devices

127	Real Time Clock
128	Additional Devices
128	PPP and SLIP
128	PPP Setup
129	SLIP Setup

## **Product Discrepancy Report**

---

**131**



---

# Chapter 1: Installing and Configuring OS-9

---

This chapter describes installing and configuring OS-9 for X86. It includes the following sections:

- **Development Environment Overview**
- **Requirements and Compatibility**
- **Target Hardware Setup**
- **Connecting the Target to the Host**
- **Building the Bootfile Image**
- **Building an OS-9 Image on the Target**
- **Creating a Startup File**
- **Optional Procedures**

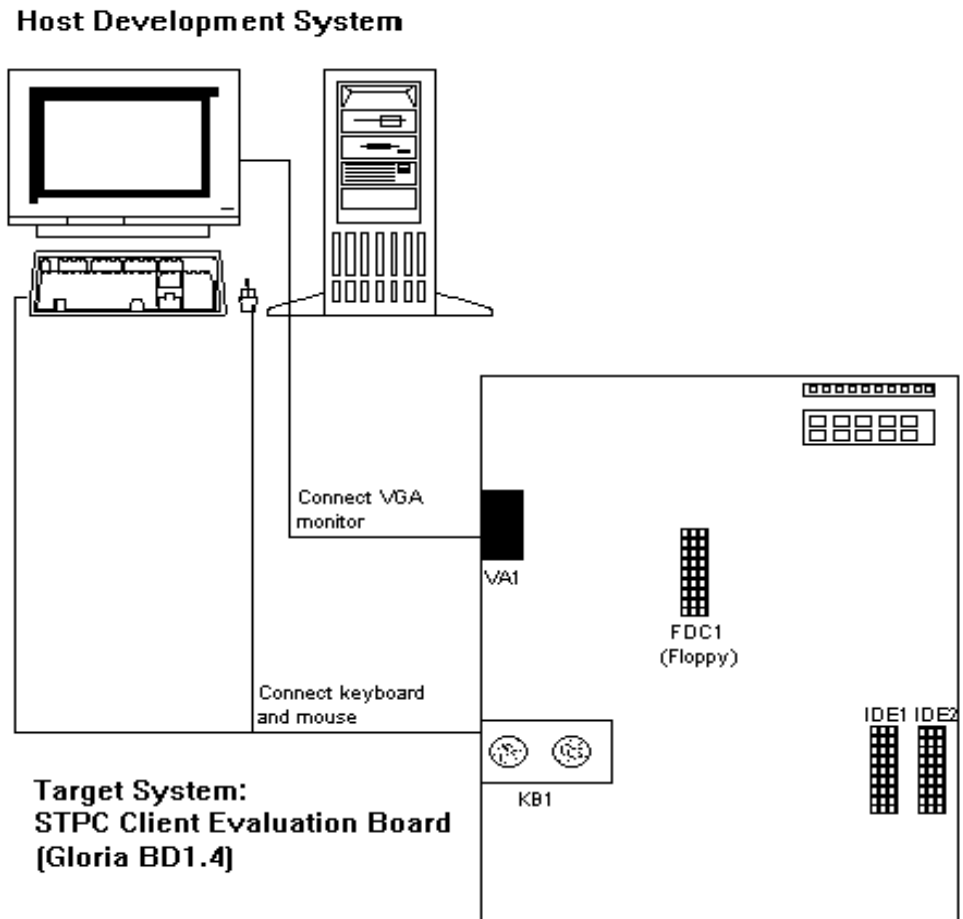


MICROWARE SOFTWARE

# Development Environment Overview

**Figure 1-1** shows a typical development environment for the PCAT board.

**Figure 1-1 STPC Development Environment**



# Requirements and Compatibility

---



---

## Note

Before you begin, install the *Enhanced OS-9 for X86* CD-ROM on your host PC.

---

## Host Hardware Requirements (PC Compatible)

The host PC must have the following minimum hardware characteristics:

- 64 MB RAM
- 250 - 350 MB free disk space
- CD-ROM drive
- network card

## Host Software Requirements (PC Compatible)

The host PC must have the following software installed:

- Enhanced OS-9
- Microsoft Windows 95, 98, ME, 2000, or NT

## Target Hardware Requirements

The following is required of your target hardware:

- an evaluation board with STPC client (consumer or industrial)
- VGA monitor
- IDE hard disk
- floppy drive
- OS-9 supported network card
- keyboard (may be removed once OS-9 is installed)

## Target Hardware Setup

---

The following sections detail the target hardware setup for the STPC client and consumer microprocessors.

## Supported Devices

The following sections list the supported devices for OS-9 for PCAT.



---

### For More Information

Refer to **Appendix B: Configuring Hardware Devices** of this document for detailed information on configuring and troubleshooting specific devices with OS-9.

---

## Ethernet Controllers

To complete development work, you will need an ISA, PCI or PCMCIA network card supported by OS-9. Driver support is also included for network cards from the following manufacturers:

- 3COM PCI
- NE2000 PCMCIA (PC Card)
- NetGear FA311/FA312
- DEC 21140
- AM79C961 & AM79C73A
- NE2000 PCI/ISA
- Cirrus Logic CS8900
- RealTek RTL-8029 and RL8139
- SMC 91C94/96

- INTEL Ethernet PRO100
- Winbond 89C940 and w89C940
- KTI ET32P2
- NetVin NV5000SC
- Via 92C926
- SureCom NE34
- Holteck HT80229/232
- LAN79C961/AM79C973
- Ready LINK COMPEX RL2000-PCI (use ne2000 Configuration Wizard setting)
- Etherlink XL PCI 3C900B-TPO
- Asanté Fast 10/100 PCI Adapter for the Mac & PC (use Dec 21140 Configuration Wizard setting)

For some Ethernet cards, the I/O base address and interrupt settings must be configured on the card to match the settings used by OS-9. A setup disk, provided with the network card, may be needed to configure the card to the correct settings. The default settings for an NE2000 card are I/O Base 0x340 and IRQ 9.

## **Maui VGA Support**

- Generic VGA mode 13 ( 320x200x8bpp )
- Generic VGA mode 12 & "X" ( 640x480x4bpp & 360x480x8bpp )
- Cirrus Alpine Series - CL-GD5434, CL-GD5480 etc. ( up to 1024x768x24bpp)
- VESA ( INT 10h ) driver
- ISA banked

## Sequential Device Support

- VGA Graphics / Keyboard
- Serial Mouse
- PS2 Mouse
- 16550 Serial
- Digiboard
- HostessI
- Risicom
- Parallel Printer

## Physical Disk Media

- IDE Standard
- PCMCIA IDE
- IDE Standard
- DiskOnChip
- DiskOnChip Descriptors
- PCAT Style Floppy
- Floppy Descriptors
- Symbios 810,810A,825,825A and 875 PCI SCSI controllers—Wide, Ultra and Ultra Wide
- Diamond FirePort20 and FirePort40—Wide, Ultra and Ultra Wide
- Adaptec 1540/1542 ISA
- Adaptec 2940, 2940U and 2940UW
- SCSI Descriptors

## System Devices

- Real Time Clock

## Additional Devices

- PPP and SLIP

## SIMM Modules

OS-9 will run with as little as 2 MB of RAM; however, it may be more convenient to install additional memory when developing and testing graphic intensive applications.



---

### For More Information

Refer to the **Hardware Installation** section of the board manual when installing the SIMM modules.

---

## CMOS Settings

It may be necessary to modify the BIOS settings in CMOS to boot from a hard disk. To do this, press **DEL** after rebooting and configure the board with the correct settings for the attached peripherals. The boot sequence should be to try floppy first, and then the IDE disk.

Additionally, an attached television monitor may be installed by selecting **NTSC** from the **TV Output** menu in the **Chipset Setup** window.



---

### For More Information

Refer to the **MAUI Graphics Support** section of this document for information on configuring OS-9 to use the optional TV Output port.

---



## Connecting the Target to the Host

---

To connect the target to the host, complete the following steps:



---

### Note

While OS-9 can be configured as a ROM based system, these instructions assume that a floppy drive and IDE hard disk will be used for initial development and testing.

---

- 
- Step 1. Connect a power supply to the board. Make sure power is in the OFF position.
  - Step 2. Connect the floppy drive to the motherboard, using the FDC1 board connector.
  - Step 3. Connect the IDE drive to the motherboard, using the IDE1 board connector.
  - Step 4. Connect a VGA monitor to the VA1 connector on the board. (You may optionally connect an NTSC monitor to the P2 connector on the board instead.)
  - Step 5. Connect a PS/2 style keyboard and mouse to the KB1 connectors on the board.
-

# Building the Bootfile Image

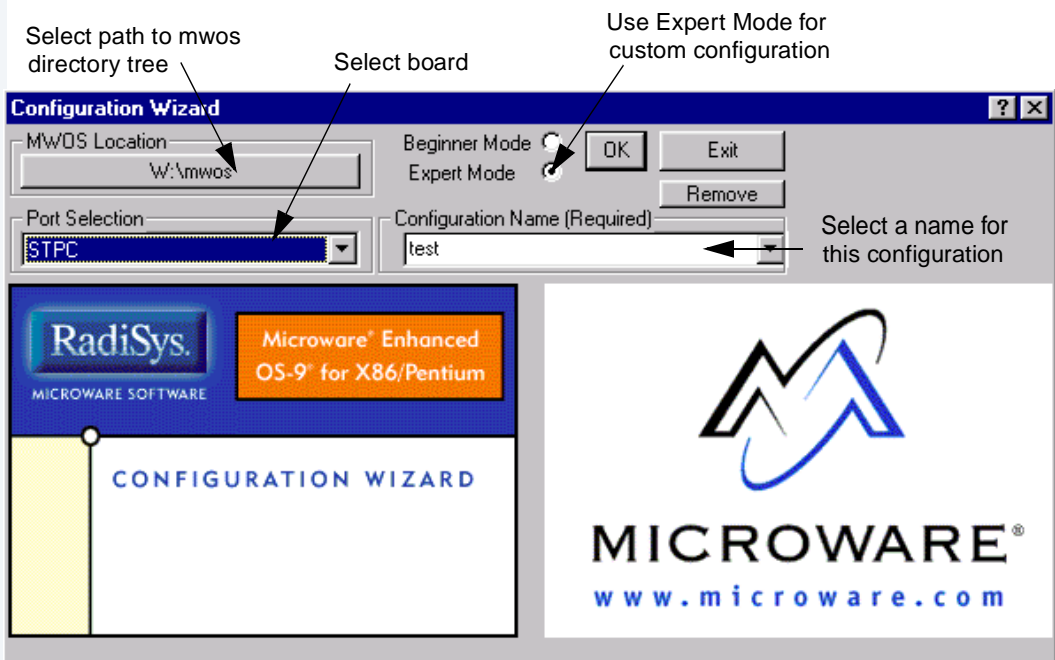
The following sections detail the preferable method for building the bootfile image. This preferable method includes building a bootable floppy disk using Microware's Configuration Wizard.

## Using the Configuration Wizard

To use the Configuration Wizard, perform the following steps:

- Step 1. Click the **Start** button on the Windows desktop.
- Step 2. Select **Programs -> Microware -> Enhanced OS-9 for X86 -> Microware Configuration Wizard**. The following dialog box appears:

**Figure 1-2 X86 Configuration Wizard**



- Step 3. Select the path where the MWOS directory structure can be located by clicking the MWOS location button.
  - Step 4. Select the target board from the Port Selection pull-down menu.
  - Step 5. Select a name for your configuration in the Configuration Name field. Your settings will be saved for future use. This enables you to modify the ROM image incrementally, without having to reselect every option for each change.
  - Step 6. Select **Use Wizard** and click **OK**. You are ready to begin preparing your floppy disk for the build.
-

## Building a Bootable Floppy Disk

Once you have opened the Configuration Wizard in **Use Wizard** mode, the **IP Address** window appears (shown in **Figure 1-3**). This window is where you will begin configuring your floppy disk for the build.

**Figure 1-3 IP Address Window**

**IP Address (SYSTEM OPTION)**

☐ Server assigned IP address  
☒ Specify an IP address

IP Address: 172.16.3.156  
 IP Broadcast Address: 172.16.255.255  
 Subnet Mask: 255.255.0.0

3COM PCI series

< Back    Next >    Cancel    Help

- Step 1. From the **IP Address** window, select the **Specify an IP address** radio button and enter your IP address, broadcast address, and subnet mask values.



## Note

Contact your system administrator if you do not know the appropriate IP values for your system.

- Step 2. Select your network adaptor model by scrolling through the choices on the pull-down list box. If you do not want networking enabled, simply select **None** at the bottom of the list. Click **Next**.
- Step 3. The **DNS Configuration** window appears (shown in **Figure 1-4**). Select the **Enable DNS** radio button and fill in the appropriate values for your network. If you do not want DNS enabled, simply select the **Disable DNS** radio button. Click **Next**.

**Figure 1-4 DNS Configuration Window**

**DNS Configuration (SYSTEM OPTION)**

☒ Disable DNS  
☐ Enable DNS

Host: LocalHost Domain: alpha.com

DNS Server Search Order

Add Remove

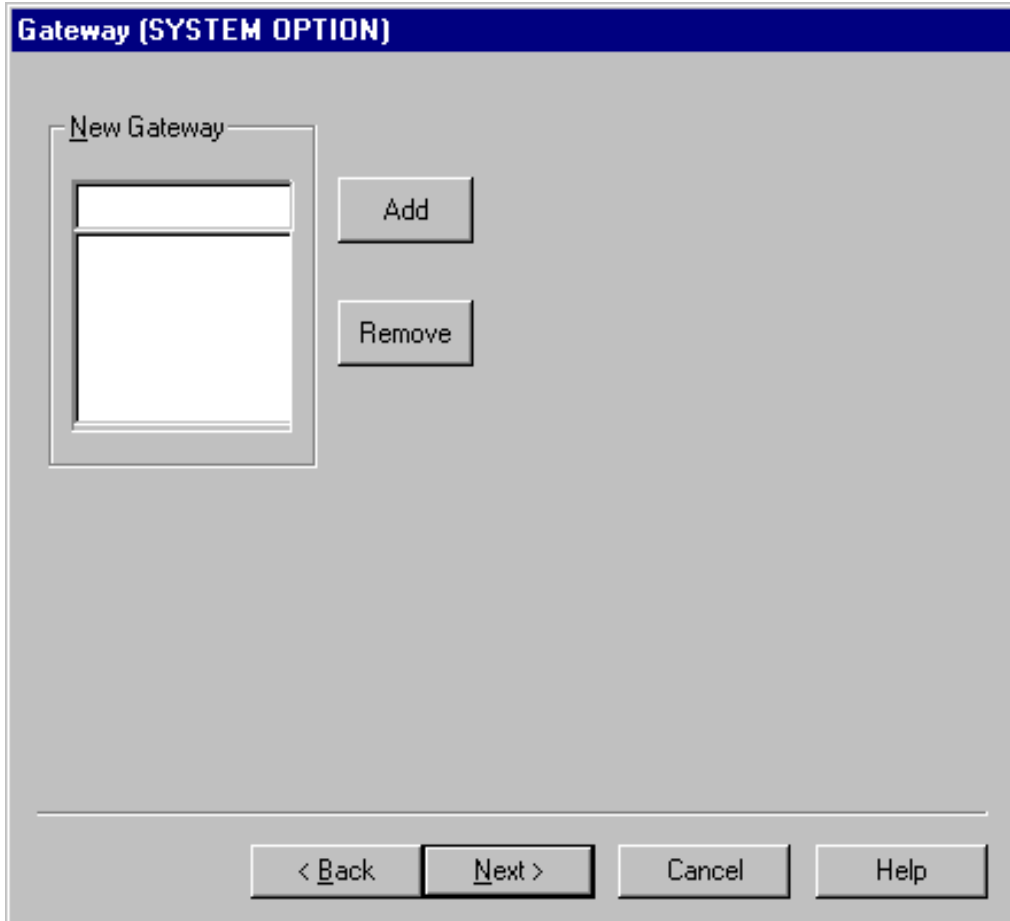
DNS Suffix Search Order

Add Remove

< Back Next > Cancel Help

- Step 4. The **Gateway** window appears (shown in **Figure 1-5**). Add in the appropriate information for your network. Click **Next**.

**Figure 1-5 Gateway Window**

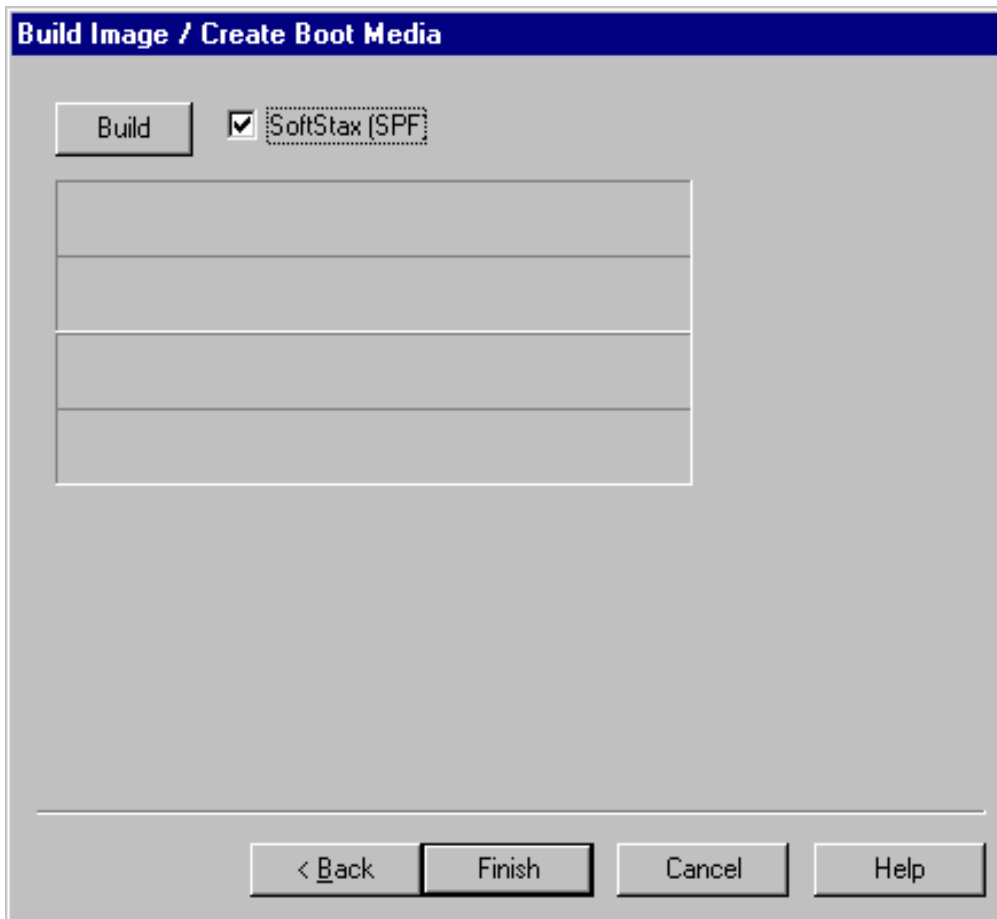


### **Note**

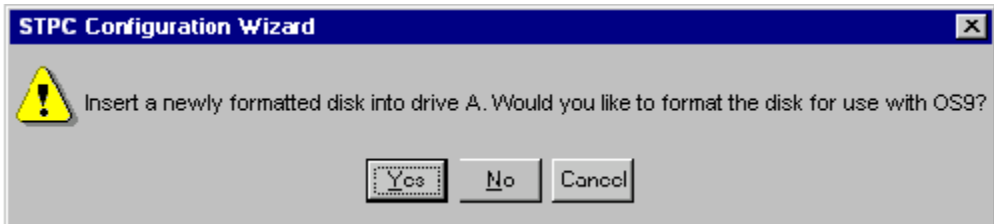
Contact your system administrator if you do not know the appropriate gateway values for your system.

- Step 5. The **Build Image/Create Boot Media** window appears (shown in **Figure 1-6**). Select the **SoftStax** check box if you would like SoftStax enabled and click **Build**.

**Figure 1-6 Build Image Window**



Step 6. When the build is complete, the following dialog box appears:



Insert a new floppy disk into your computer's floppy drive and click **Yes** to format it for OS-9 and copy the boot image.

Step 7. Click **Finish**. A dialog appears prompting you to save the file.

The newly created boot floppy can be used to bring OS-9 up on the X86 target hardware. The default OS-9 console is the target's monitor.

If you enabled networking, you should be able to telnet into the target from your host PC. To do this, select **Run** from the Windows start menu and type the following command:

```
telnet OS9test
```

The login user name is `super`; the password is `user`.

---



## Building an OS-9 Image on the Target

---

This section describes using the floppy you built in the previous sections on the target machine.

### Preparing the Hard Disk

The newly created boot floppy may be used to format a local hard disk with the OS-9 file system. A network connection between the OS-9 target machine and the Windows host computer may be used to load the OS-9 system files onto the hard disk.

### Partitioning the Drive

Complete the following steps to partition the hard drive:

- Step 1. Boot the target system, using the floppy made in the previous section. The console menu appears. An example of the console menu is shown below.

```

BOOTING PROCEDURES AVAILABLE ----- <INPUT>

Boot FDC floppy ----- <fd>
Boot from PC-Floppy ----- <pf>
Boot from (RBF) IDE Primary Master ----- <ide>
Boot from PCMCIA(PC) IDE Primary Master-Socket #0 -- <pcm_pc>
Enter ROM Debugger ----- <break>
Restart the System ----- <q>

Select a boot method from the above menu:

```

- Step 2. Select **fd** from the boot menu to boot the floppy disk you inserted in the disk drive. Press **Enter** on the keyboard.

Step 3. Once the bootfile is read, the OS-9 console prompt appears (\$). From the prompt, run `fdisk` by typing the following command:

```
fdisk -d=/hc<n>fmt -e
```

Step 4. The **Fdisk Options** menu appears. An example of this menu is shown below.

```
Current fixed disk device: /hc<n>fmt@
```

```
Choose operation to execute:
```

1. Create OS-9000 partition
2. Set active partition
3. Delete partition
4. Display partition information
5. Change extended Dos partition to OS9000 partition
6. Write master boot record (MBR)

Select **1** to create the OS-9 partition. The partition information is displayed.

Step 5. The **Enter the partition size in cylinders: [ ]** prompt appears at the bottom of the screen. Press **Enter** to accept the size.

Step 6. The following partition type options are displayed:

1. OS9000/386 type partition
2. Extended Type 41 partition

Select **1** and press **Enter**. The partition information is displayed again. Press **Esc** to return to the **Fdisk Options** menu.

Step 7. The next step is to make the partition active. Once returned to the **Fdisk Options** menu, select **2** to set the active partition. Press **Enter**.

Step 8. The partition information is displayed once again. When prompted, select the number that corresponds to the partition you would like to make active and press **Enter**.

- Step 9. The partition information displays your new information. Press **ESC** to return to the **Fdisk Options** menu. At the menu, you can do one of two things:
- If the disk you are using is new and contains no other operating systems, proceed to step ten.
  - If at least one other operating systems exists on your disk, proceed directly to the section, **Formatting the Hard Drive**.
- Step 10. If the disk you are using contains no other operating system, you will need to write master boot record (MBR) to the disk. To do this, select **6** from the **Fdisk Options** menu.
- Step 11. The display information prompts you to select the partition on which you wish to write MBR. Once you have done this, press **Enter**. The **Fdisk Options** menu is displayed. Press **ESC** to exit this menu.
- Step 12. The **want to save new partition information** prompt appears. Press **y** to save the partition information and press **Enter**. The partition information is written to the drive.
- 

## Formatting the Hard Drive

To format the hard drive, complete the following steps:

- 
- Step 1. Create the OS-9 RBF file system by running `format` from the console:
- ```
format /hc<n>fmt
```
- Press **Enter**.
- Step 2. The disk format utility parameters display. At the prompt, select **y** if you are ready to begin performing the selected partition. Press **Enter** to continue.

Step 3. From here you are prompted to enter the following information, respectively:

- physical format
- disk name
- physical verify



---

### Note

It is not typically necessary to complete a physical format and verify.

---

After entering the appropriate information, your formatting information is displayed and the OS-9 prompt returns.

---

## FTP to the Target

The following steps discuss the procedure for transferring required files from the host to the target machine.

- 
- Step 1. On the host machine, open the command prompt window.
- Step 2. From the command prompt, move to the directory which contains the diskcache file. (This file is located in `MWOS\OS9000\80386\CMD5.`)
- Step 3. Begin the FTP session with the target machine by typing the following command:

```
ftp <target>
```

- Step 4. To log in to the system, type the appropriate username and password. (The username is `Super`. The password is `User`.)
- Step 5. At the next `ftp` prompt, type the following command to set binary mode:

```
bin
```

Step 6. Move to the target hard drive directory by typing the following command:

```
cd hc<n>fmt
```

Step 7. Type the following command:

```
put tar
```

Step 8. To place the `diskcache` file on the target system, type the following command:

```
put diskcache
```

Step 9. To place the MWOS directory tree onto the target system, type the following command:

```
put mw86sm.tar
```



---

### Note

The following instructions assume a minimal configuration with limited disk space (less than 4MB). If additional disk space is available, then you may wish to download `mw86.tar` in place of `mw86sm.tar`. `mw86.tar` includes the full command set and descriptors.

---

Step 10. To exit the program, type `quit`.

Step 11. To turn on disk cache support from the OS-9 console, type the following command:

```
$ chd /hc1fmt ; load -d diskcache ; diskcache -e /hc1fmt=1024k
```

Step 12. Expand the system files by typing the following command:

```
$ load -d tar ; tmode nopause ; tar xvpf mw86sm.tar
```

The disk is now formatted and the OS-9 system files have been copied to disk.

---

## Configuring the Hard Drive

This section finishes the hard disk configuration by using the OS-9 `bootgen` utility to install a boot image onto the hard disk. The hard disk will be made bootable using the OS-9 boot image created in the previous sections.

- 
- Step 1. Boot the target system using the floppy made in the previous section.
  - Step 2. At the OS-9 prompt, verify that you can access the hard disk by executing the following command:  

```
$ dir /h0
```

An OS-9 directory listing should be displayed.
  - Step 3. Turn disk caching off by typing the following command at the OS-9 console:  

```
$ diskcache -d /hclfmt
```
  - Step 4. To ready `bootgen` on the new system, type the following command at the OS-9 console:  

```
$ bootgen /hclfmt -i=/d0/iplhdnoq -l=/d0/firstboot /d0/sysboot -nb400
```
  - Step 5. Remove the floppy from the drive and reboot the system. The system should boot from the hard disk, with the OS-9 system prompt appearing on the console.
-

## Creating a Startup File

---

When the Configuration Wizard is set to use the hard drive as the default device, it automatically sets up the init module to call the following file:

```
/dd/sys/startup
```

However, this startup file will not exist until you create it. To create and configure the startup file, complete the following steps:

- 
- Step 1. Create a `SYS` directory on your target machine. This is the directory in which the startup file will reside (for example, `mkdir /dd/sys`).
  - Step 2. On your host machine, navigate to the following location:  
`MWOS/OS9000/SRC/SYS`  
In this directory, you will see the following three files:
    - `password`
    - `termcap`
    - `startup`
  - Step 3. Transfer the `password` and `termcap` files from their location on the host machine (`MWOS/OS9000/SRC/SYS`) to the newly created `SYS` directory on the target machine. These files do not require modification.
  - Step 4. Once you have transferred the `password` and `termcap` files, you will need to transfer the `startup` file from the `SYS` directory on the host to the `SYS` directory on the target machine. However, please note that because the commands in the startup file are system dependent, it may be necessary to modify the file to fit your system configuration. Moreover, it is recommended that you modify the startup file before transferring it to your target machine.
-

## Example Startup File

Below is the example startup file as it appears in  
MWOS/OS9000/SRC/SYS:

```
-tnxnp
tmode -w=1 nopause
*
*OS-9000 - Version 3.0
*Copyright 2001 by Microware Systems Corporation
*Copyright 2001 by RadiSys Corporation
*The commands in this file are highly system dependent and *should be
modified by the user.
*
*setime </term                ;* start system clock
setime -s                    ;* start system clock
link mshell csl              ;* make "mshell" and "csl" stay in memory
* iniz r0 h0 d0 t1 p1 term   ;* initialize devices
* load utils                  ;* make some utilities stay in memory
* tsmon /term /t1 &          ;* start other terminals
list sys/motd
setenv TERM vt100
tmode -w=1 pause
mshell<>>>/term -l&
```




---

### For More Information

Refer to the section, **Making a Startup File**, in Chapter 9 of the *Using OS-9* manual for more information on startup files.

---



## Optional Procedures

---

The following sections discuss optional procedures you may perform once you have set up and configured the PCAT board.

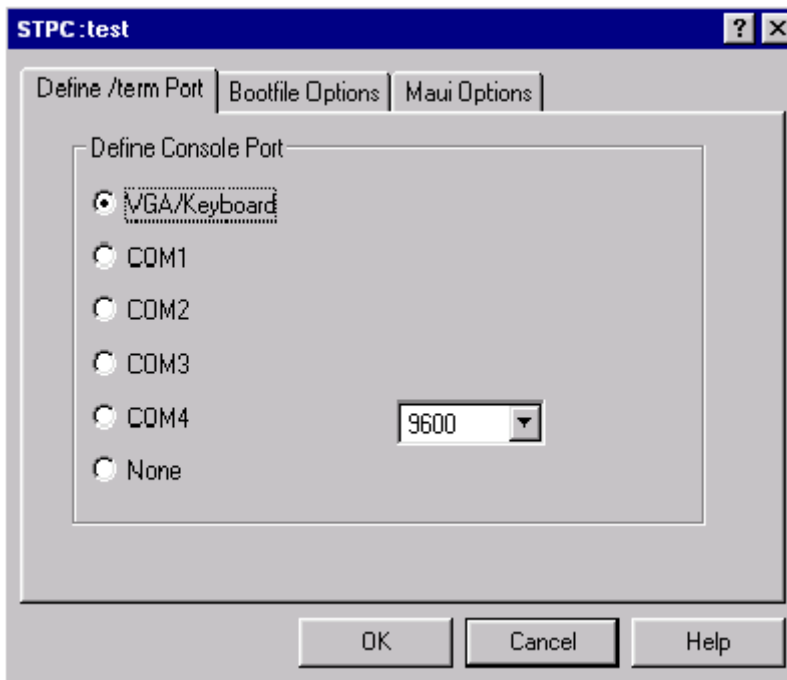
### Advanced Configurations

The following steps detail configuring the target system to boot from a local hard disk and to moving the OS-9 console to a serial port. This might be beneficial if you wish to remove the monitor or use it specifically for graphic applications.

- 
- Step 1. Open the Configuration Wizard in **Advanced Mode**. The **Main Configuration** window is displayed.

- Step 2. Select **Configure -> Bootfile -> Configure System Options** from the menu. The following window appears:

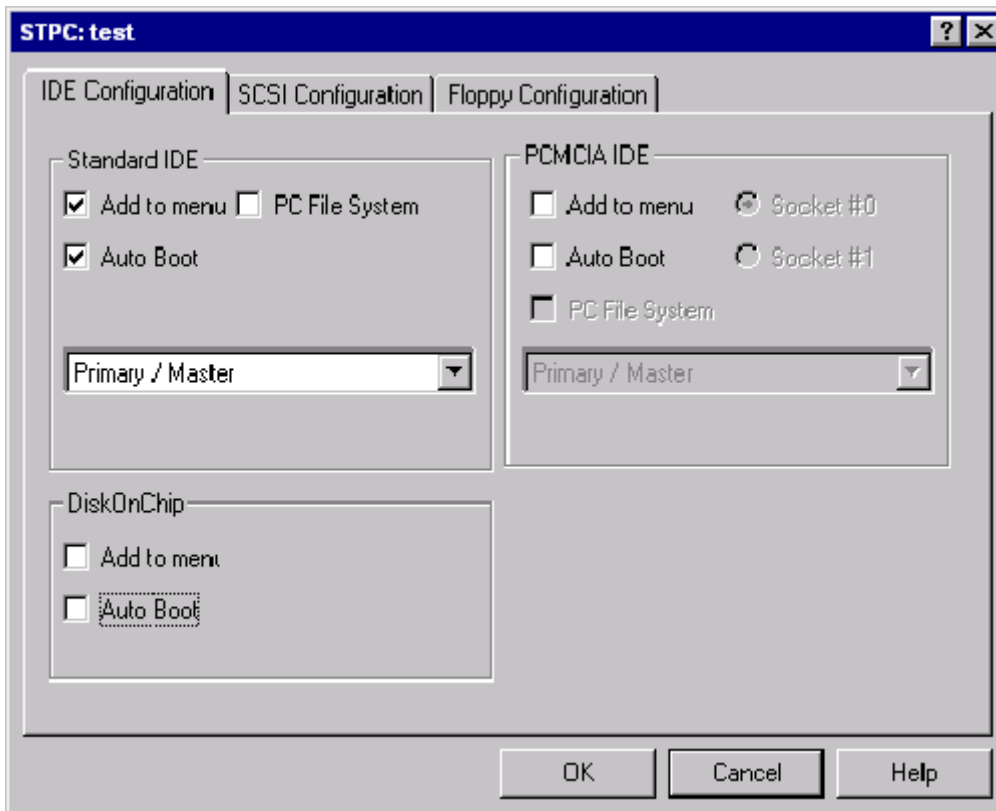
**Figure 1-7 Configure System Options Window-Define /term Port tab**



- Step 3. If you plan on using the monitor and keyboard as the OS-9 system console, leave the **VGA/Keyboard** radio button checked. Otherwise, click on the **COM1** radio button on the **Define /term Port** tab. This moves the high-level console to serial port one.
- Step 4. Verify that the baud rate is set to 9600. Click **OK**.

- Step 5. Select **Configure -> Coreboot -> Disk Configuration**. The following window appears:

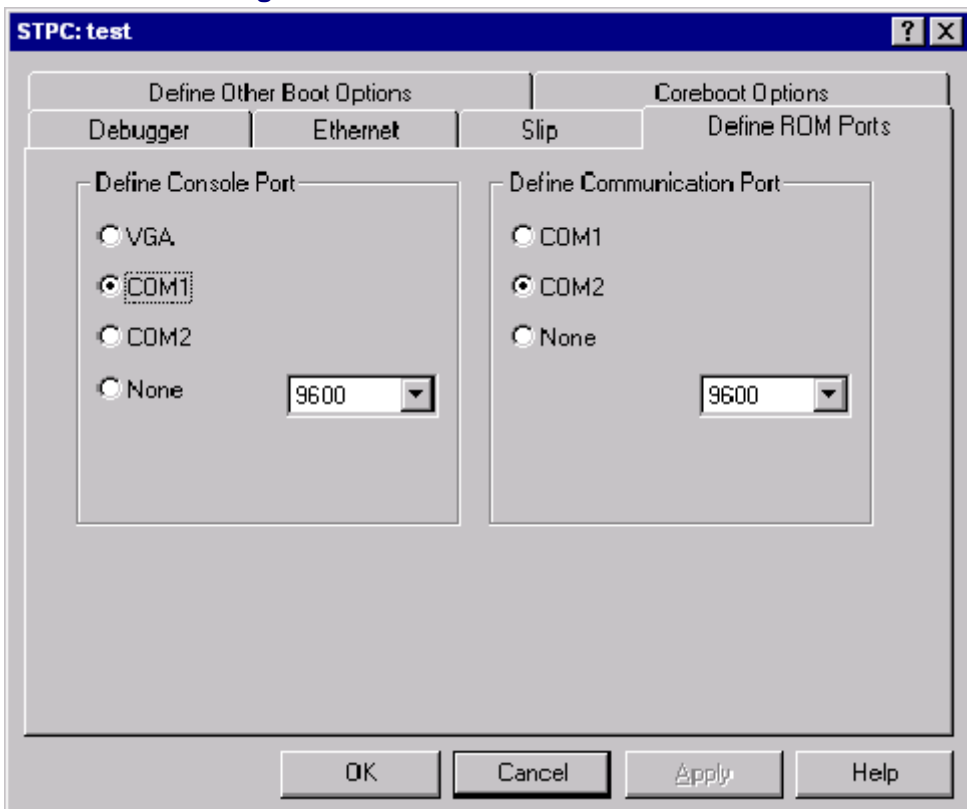
**Figure 1-8 Disk Configuration Window-IDE Configuration tab**



- Step 6. Click the **Auto Boot** check box and click **OK**.
- Step 7. Select **Configure -> Coreboot -> Main Configuration** from the menu.

Step 8. Select the **Define ROM Ports** tab. The following window appears:

**Figure 1-9 Main Configuration Window-Define ROM Ports tab**

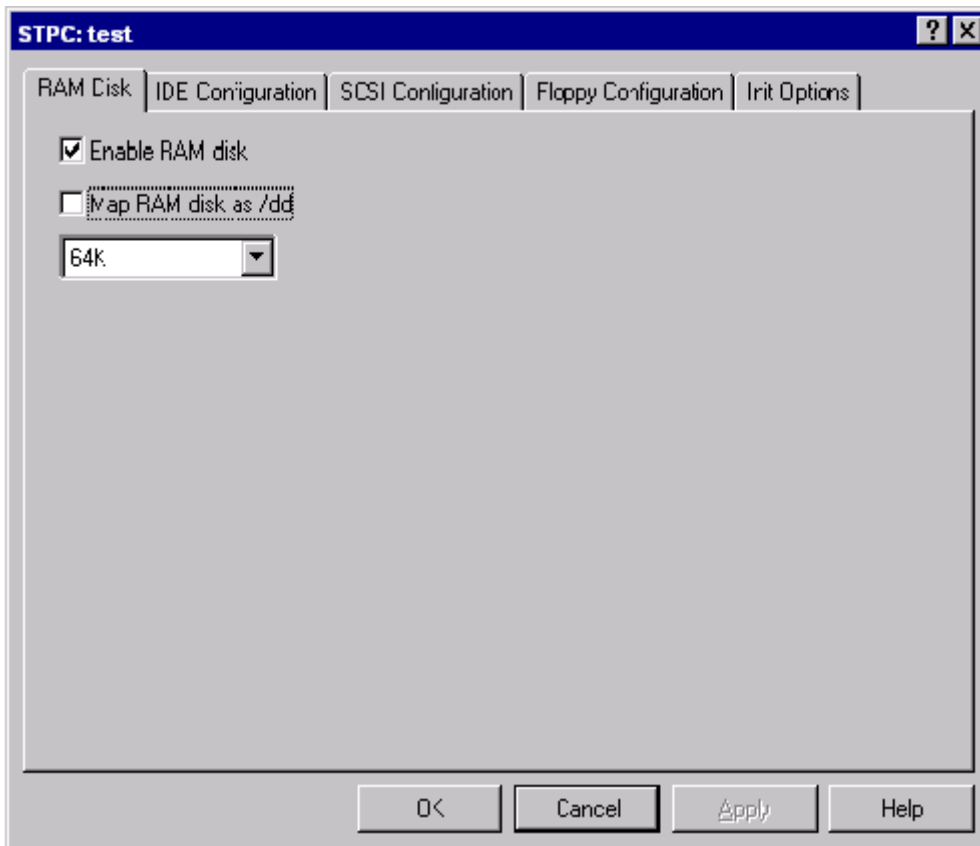


Step 9. If you plan on using the monitor and keyboard as the OS-9 system console, leave the **VGA/Keyboard** radio button checked. Otherwise, click on the **COM1** radio button on the **Define Console Port** and **Define Communication Port** areas. This moves the high-level console to serial port one.

Step 10. Click **OK**.

Step 11. Select **Configure** -> **Bootfile** -> **Disk Configuration** from the menu. The following window is displayed:

**Figure 1-10 Disk Configuration Window-RAM Disk tab**

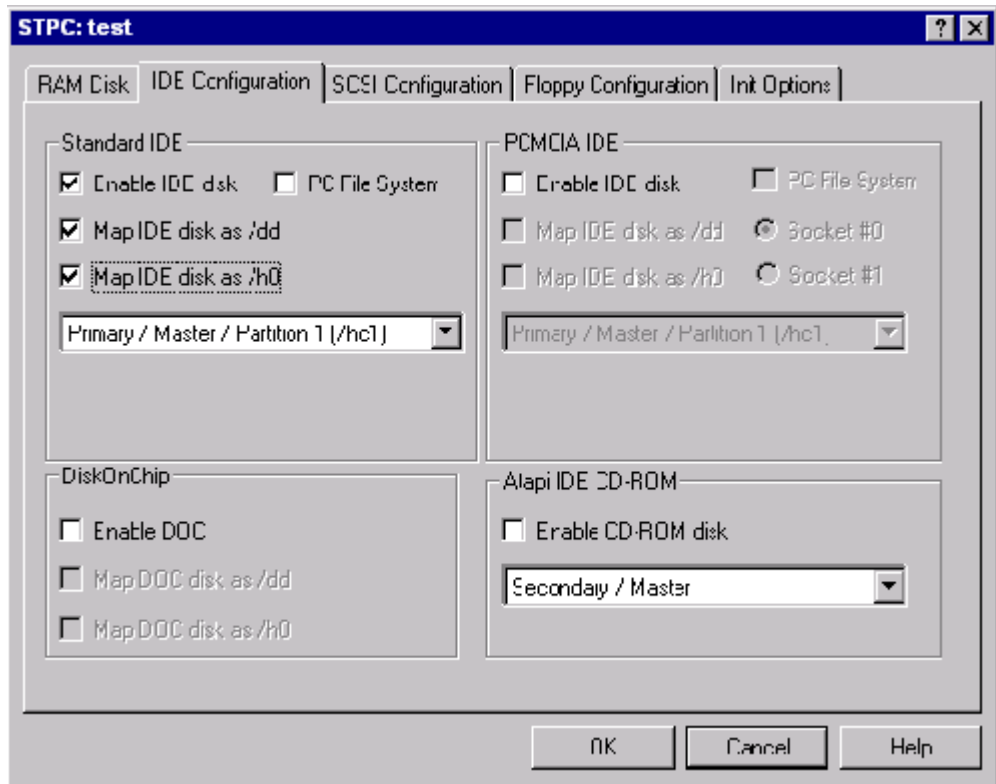


Step 12. From the **RAM Disk** tab, verify that the **Enable RAM disk** box is checked and other boxes remain unchecked.

Step 13. Select the RAM disk size from the drop down list box. Use of a RAM disk is optional, and you may disable it by unchecking the **Enable RAM disk** box. If enabled, the RAM disk may be accessed as `/r0` on the target system.

Step 14. Select the **IDE Configuration** tab. The following window is displayed:

**Figure 1-11 Disk Configuration Window-IDE Configuration tab**



Step 15. Click the **Enable IDE disk**, **Map IDE disk as /dd**, and **Map IDE disk as /h0** check boxes to enable them.



## Note

The standard IDE hard disk will be accessed as device `/hc1` from the OS-9 console. The same device may also be accessed as `/h0` or `/dd`.

An IDE CD-ROM drive may be attached to the target system and accessed as device `/cd0`. The CD-ROM must be the master device on the second IDE channel.

Step 16. Click on the **Init Options** tab. The following window appears:

**Figure 1-12 Disk Configuration Window-Init Options tab**

**STPC: test**

FAM Disk | IDE Configuration | SCSI Configuration | Floppy Configuration | **Init Options**

Initial Module Name: ☐ Shell ☒ MShell ☐ User

Initial Device Name: ☐ Nc Disk ☐ /dd ☒ /h0 ☐ /d0 ☐ User ☐ NFS Mount

Tick Rate:

Ticks Per Time Slice:

Initial Device Name:

Initial Module Name:

Parameter List:

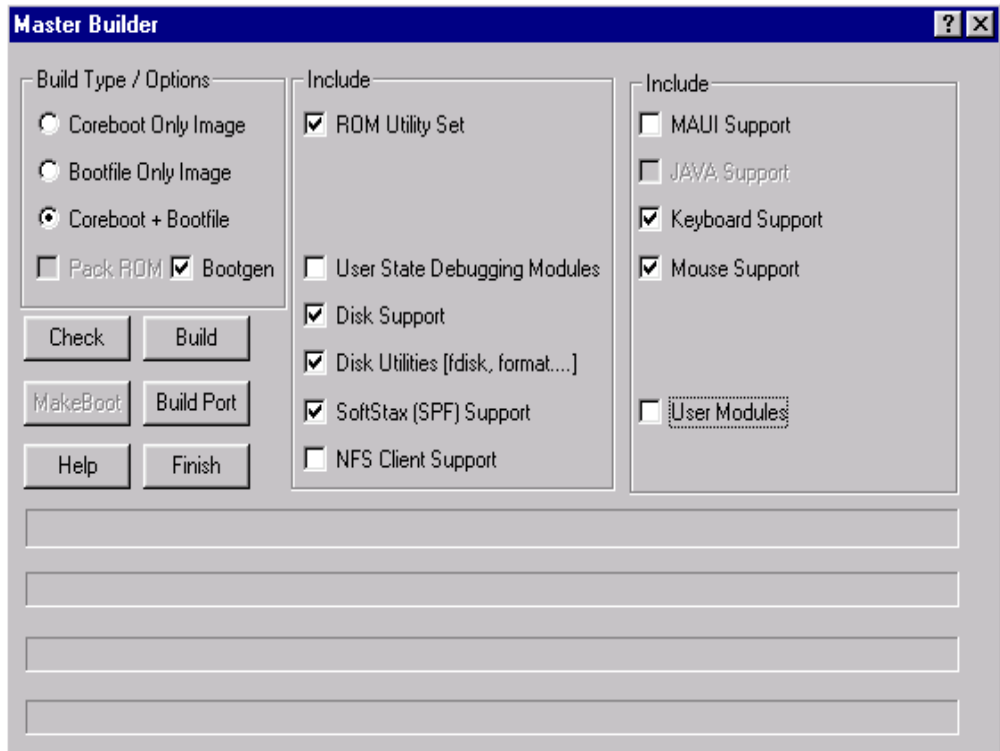
system time zone:minutes offset from GMT:  ☐ Use system time offset

☐ Wipe Memory Flag ☐ Allow User State I/C

OK Cancel Apply Help

- Step 17. Select the **/h0** radio button to use the as the initial device. Click **OK**.
- Step 18. Select **Configure -> Build Image** from the menu. The following window appears:

**Figure 1-13 Master Builder Window**





Step 19. Verify that the following options are checked:

- Bootgen
- Coreboot + Bootfile
- ROM Utility Set
- Disk Support
- Disk Utilities [fdisk, format...]
- SoftStax (SPF) Support Modules
- Keyboard Support
- Mouse Support (Enables support for a PS/2 style mouse)

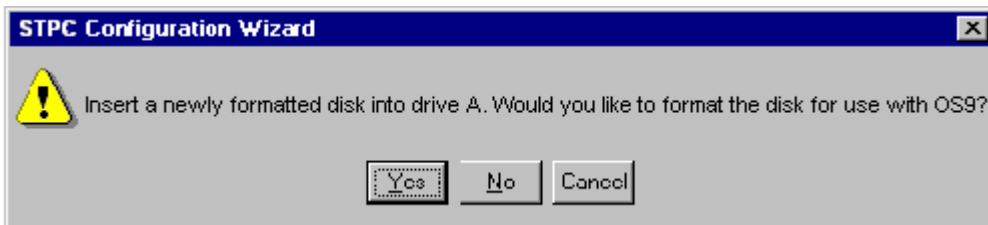


### Note

Select the **User State Debugging Modules** check box to include the Hawk debugging modules on the target system. Alternately, you may load and run the modules from the hard disk on the target.

Step 20. Click **Build** to create the OS-9 boot image.

Step 21. Click the **MakeBoot** button once it is enabled. The following dialog box appears:



Step 22. Insert a new floppy disk into your computer's floppy drive and click **Yes** to format it for OS-9 and copy the boot image.

Step 23. Once the boot image has been written to floppy, select **Finish**.

Step 24. Save your changes and exit the Configuration Wizard by selecting **File** -> **Exit**.

If you enabled networking, you should be able to telnet into the target from your host PC. To do this, select **Run** from the Windows start menu and type the following command:

```
telnet OS9test
```

The login user name is `super`; the password is `user`.

---

---

## Chapter 2: Board Specific Reference

---

This chapter contains porting information that is specific to the X86 board. It includes the following sections:

- **MAUI Graphics Support**



MICROWARE SOFTWARE

# MAUI Graphics Support

---

This section details information for using MAUI (Multimedia Application User Interface) for the STPC board.

## Getting Started

To start MAUI from the OS-9 console, complete the following steps:

- 
- Step 1. From the command prompt, navigate to the `/h0/sys` directory.
  - Step 2. At the prompt, type the following command:

```
loadmaui
```

To verify that MAUI is running, try executing one of the demo programs, such as `fdraw` or `fcopy`, from the OS-9 console.

---

## Configuring the Display

OS-9 for STPC supports 256 colors with 640 x 480, 800 x 600 or 1024 x 768 resolutions. You can set this resolution by editing the `loadmaui` script and finding the STPC sections at the end of the file. You can also uncomment the code block for the desired resolution by removing the leading asterisks. The unused resolution must be commented out.

The STPC evaluation board may be configured to use the NTSC television output in place of the normal SVGA output by removing the leading asterisks from the NTSC section of the `loadmaui` file. The unused SVGA resolutions must be commented out. The video output must also be changed in the BIOS CMOS settings.

The following code fragments, from the `loadmaui` file, configure the display for 640 x 480 resolution.

```
* STPC - Gloria Board
* The gd_???? is for the client version and
* the gd_????_co is for the consumer version
* of the cpu.
* MAUI port - STPC at 640x480 256 color
*
load -d CMDS/BOOTOBJS/MAUI/cdb_stpc
load -d CMDS/BOOTOBJS/MAUI/svgga_stpc
load -d CMDS/BOOTOBJS/MAUI/gx_stpc6
load -d CMDS/BOOTOBJS/MAUI/gd_stpc
*load -d CMDS/BOOTOBJS/MAUI/gd_stpc_co
*
* MAUI port - STPC at 800x600 256 color
*
*load -d CMDS/BOOTOBJS/MAUI/cdb_stpc
*load -d CMDS/BOOTOBJS/MAUI/svgga_stpc
*load -d CMDS/BOOTOBJS/MAUI/gx_stpc8
*load -d CMDS/BOOTOBJS/MAUI/gd_stpc
*load -d CMDS/BOOTOBJS/MAUI/gd_stpc_co
*
* MAUI port - STPC at 1024x768 256 color
*
*load -d CMDS/BOOTOBJS/MAUI/cdb_stpc
*load -d CMDS/BOOTOBJS/MAUI/svgga_stpc
*load -d CMDS/BOOTOBJS/MAUI/gx_stpc1
*load -d CMDS/BOOTOBJS/MAUI/gd_stpc
```

```
*load -d CMDS/BOOTOBJS/MAUI/gd_stpc_co
*
* MAUI port - STPC at 640x480 256 color NTSC output
*
*
*load -d CMDS/BOOTOBJS/MAUI/cdb_stpc
*load -d CMDS/BOOTOBJS/MAUI/svgastpc
*load -d CMDS/BOOTOBJS/MAUI/gx_stpc6
*load -d CMDS/BOOTOBJS/MAUI/gd_stpcntsc
*load -d CMDS/BOOTOBJS/MAUI/gd_stpcntsc_co
*
* End of bootlist
```

In the above example, the file `cdb_stpc` is the control data block for the mouse. `svgastpc` is the descriptor file for the graphics display. `Gx_stpc8` is the 800 x 600 resolution driver, `gx_stpc6` is the 640 x 480 resolution driver and `gx_stpc1` is the 1024 x 768 resolution driver. `Gd_stpc` and `Gd_stpcntsc` are data modules used by the graphic driver.

---

# Appendix A: Board Specific Modules

---

This chapter describes the modules specifically written for the target board. It includes the following sections:

- **Low-Level System Modules**
- **High-Level System Modules**



MICROWARE SOFTWARE

# Low-Level System Modules

---

The following low-level system modules are tailored specifically for the STPC evaluation board. These modules are located in the following directory:

MWOS/OS9000-80386/PORTS/STPC/CMD5/BOOTOBJS/ROM




---

## For More Information

For information on additional X86 modules, refer to Appendix A of the ***OS-9 for PCAT Board Guide*** included with this CD.

---

|          |                                                     |
|----------|-----------------------------------------------------|
| cnfgdata | data module containing the configuration parameters |
| romcore  | system initialization                               |



## High-Level System Modules

---

The following OS-9 system modules are tailored specifically for the STPC evaluation board. Unless otherwise specified, each module is located in a file of the same name in the following directory:

MWOS/OS9000-80386/PORTS/STPC/CMD5/BOOTOBJS



---

### For More Information

For information on additional X86 modules, refer to Appendix A of the ***OS-9 for PCAT Board Guide*** included with this CD.

---

## MAUI Support

cdb\_stpc  
gd\_stpc  
gd\_stpc\_co  
gd\_stpcntsc  
gd\_stpcntsc\_co  
gx\_stpc1  
gx\_stpc6  
gx\_stpc8  
svga\_stpc

## PersonalJava™ Support

pjruntime  
pjruntime\_g



---

# Appendix B: Configuring Hardware Devices

---

This appendix contains detailed information for configuring and troubleshooting specific devices with OS-9. The following sections are included:

- **Ethernet Controllers**
- **Sequential Device Support**
- **Physical Disk Media**
- **System Devices**
- **Additional Devices**



MICROWARE SOFTWARE

# Ethernet Controllers

---



## Note

Some Network Interface Cards require that a setup disk, included with the card, is ran before the card is installed in a system running OS-9.

The setup disk is required for configuring the connection type for cards which support multiple interfaces, such as connections for 10Base-T, 10Base-2 or AUI. The setup disk may also be needed to configure the card for a specific interrupt or I/O address.

---

## 3COM PCI

3C900B-TPO - 10Base-T TPO NIC

3C900B-CMB - 10Base-T/10Base-2/AUI Combo

3C905-T4 - 10/100 Base-T4 (RJ-45) - 3C905-T4 Fast Etherlink XL

3C905B-TX - 10/100Base-TX NIC

3CSOHO100-TX - 10/100 Base-TX NIC - Office Connect 10/100

3C900-TPO - 10Base-T TPO NIC

System State Debugging - Supported

## Default Settings

PORTADDR      NA

IRQVECTOR     NA

CONNTYPE      INF\_EXT /\* Auto \*/

## Solving Configuration Issues

### Connection Type

The default connection type is set to INF\_EXT (auto). For the 3COM ISA card, this implies the card setup program has been used and has setup the card connection type. If you are unable to communicate with this card and netstat -in shows the device, the connection type may be incorrect. To correct it, you may do one of the following:

1. Use the 3COM setup disk to configure the card for the connection used.
2. Change the OS-9 device descriptor for the type of connection in use.
3. Try one of the other connections on the card (if using AUI type, try the RJ45 connector).

```
(Super)[/h0/sys/>] netstat -in
```

```
NameMtu NetworkAddressIpktslerrsOpktsOerrsColl
lo0 1536<Link>00 0 0 0
lo0 1536127127.0.0.10000 0
enet01500<Link>00.00.C0.91.4F.96551103500
enet01500182.52.109182.52.109.2555103500
```

### Modifying the OS-9 Descriptor

Edit the file

MWOS/OS9000/80386/PORTS/PCAT/SPF/SPE509/DEFS/spf\_desc.h, looking in the "#ifdef spe30\_pci" section for CONNTYPE, which you should set to the appropriate value from the following list:

INF\_AUI = AUI Connection type

INF\_BNC = BNC connection type

INF\_UPT = 10BaseT (RJ45)

INF\_EXT = Use same connection type determined in 3COM setup program

```
/*
```

```
* From spf_desc.h
```

```
*/
```

```
/* options for CONNTYPE: INF_AUI, INF_BNC, INF_UTP, INF_EXT (auto) */
```

```
#define CONNTYPEINF_EXT
```

Finally, remake the descriptor: change to the MWOS/OS9000/80386/PORTS/PCAT/SPF/SPE509 directory and type "os9make -f=spfdesc.mak".

Next time you run the Wizard, it will use the new descriptor.

## Boomerang

The source code for the spe509 driver includes “#if defined(BOOMERANG)” sections to allow only including support for the newer 3COM PCI based cards. Each card is now defined in a constant table and as such the driver makefile used must be modified to include both the define for “BOOMERANG” and the compiler option “-c” to force constant code data.

```
/* spfdrvr.mak - add the following define and compiler option */
```

```
DEFINES = -c -dBOOMERANG
```

```
/* spfdesc.mak - add the following define */
```

```
MACROS = -dBOOMERANG
```

## DMA

To allow support for the newer 3COM “B” based cards, DMA support with ring buffers has been added. The size of the ring buffers may be set in the “spf\_desc.h” file.

```
#define RX_RING_CNT32/* Number of buffers in BOOMERANG rcv ring */
```

```
#define TX_RING_CNT16/* Number of buffers in BOOMERANG xmit ring */
```

## Time-out Options

To allow support with switches and slow hubs the time-out for checking for link beat has been increased. This change effects 3COM NON-B parts as well as PCMCIA CARDS using UTP connections. The default time-out prior to this change was 750ms. Most switches take two to three seconds to sync. A loop count has been added.

```
/*
```

```
* When a connection type is tried we will wait for the time
```

```
* specified in LINK_BEAT_ITER and LINK_BEAT_SLEEP_TIME.
```

```
* This should address the problem with not being able to work
```

```
* with switches. Most switches will take 2 to 3 seconds, we will wait up to
```

```
* 5.25 seconds (192/256ths)*7.
```

```
*
```

```
*/
```

```
#define LINK_BEAT_ITER 7
```

```
#define LINK_BEAT_SLEEP_TIME 0x800000c0 /* 192/256ths of a second (750 ms) */
```



## Note

The **PCIV** utility may be used to examine a network card. This utility displays vendor and device ID's for each installed PCI device.

To find out if your card has been tested with OS-9, run the `pciv` command and look at the vendor and device ID's. The vendor ID should be 0x10B7 for all 3COM network cards. Network cards with the following device ID's have been tested with OS-9 drivers shipping with this release.

|                   |        |     |
|-------------------|--------|-----|
| 3COM 3C509        | 0x5900 |     |
| 3COM 3C900-TPO    | 0x9000 |     |
| 3COM 3C900        | 0x9001 |     |
| 3COM 3C900B-TPO   | 0x9004 |     |
| 3COM 3C900B-CMB   | 0x9005 |     |
| 3COM 3C905-T4     | 0x9051 | (2) |
| 3COM 3C905B-TX    | 0x9055 | (1) |
| 3COM 3CSOHO100-TX | 0x7646 | (1) |

Support for the following cards is included with the driver, however, these cards were not tested prior to the release.

|                            |        |
|----------------------------|--------|
| 3COM 3C905-TX              | 0x9050 |
| 3COM 10/100 COMBO Deluxe   | 0x9058 |
| 3COM 10Base-T/10Base-2/TPC | 0x9006 |
| 3COM 10Base-FL NIC         | 0x900A |
| 3COM 100Base-FX NIC        | 0x905A |

|                                    |        |
|------------------------------------|--------|
| 3COM Tornado NIC                   | 0x9200 |
| 3COM 10/100 Base-TX NIC (Python-H) | 0x9800 |
| 3COM 10/100 Base-TX NIC (Python-T) | 0x9805 |

### Additional Notes

1. 100BaseT support is included for the 3C905B-TX and 3CSOHO100-TX.
  2. The 3C905-T4 has been tested with 10BaseT only.
- 

## 3COM ISA

3COM ISA EtherLink III

### System State Debugging

Supported

### Default Settings

|           |         |                       |
|-----------|---------|-----------------------|
| PORTADDR  | 0x340   | /* IO port for ISA */ |
| IRQVECTOR | 0x43    | /* IRQ vector */      |
| CONNTYPE  | INF_EXT | /* Auto */            |



## Solving Configuration Issues

### Connection Type

The default connection type is set to INF\_EXT (auto). For the 3COM ISA card, this implies the card setup program has been used and has setup the card connection type. If you are unable to communicate with this card and netstat -in shows the device, the connection type may be incorrect. To correct it, you may do one of the following.

1. Use the 3COM setup disk to configure the card for the connection used.
2. Change the OS-9 device descriptor for the type of connection in use.
3. Try one of the other connections on the card (if using AUI type, try the RJ45 connector).

### Interrupt Conflict

Another problem may be the interrupt used. The default interrupt is IRQ3. In this case you have the following options.

1. Disable the COM2 serial port from the BIOS to allow IRQ3 to function with this card.
2. Choose a interrupt that matches the system configuration such as IRQ10 (0x4a). In this case the OS-9 device descriptor must be changed.

If an interrupt conflict exists the device will either not work at all or will hang when the conflicting device is accessed. Mapping the interrupts used in the system is recommended.

If it seems like we should be getting interrupts this can be tested.

Use the command `irqs` to see a list of interrupts, e.g.:

```
(Super)[/h0/sys/>] irqs
```

PC-AT Compatible 80386 OS9 For Embedded Systems

```
vector ($) prior drivstat irq svc driver dev list
-----
 7 ($07) 10 $0003c444 $0010f7b4 fpu <na>
14 ($0e) 1 $0003c3a4 $00110113 vectors <na>
64 ($40) 10 $00ff40b0 $0011098f tk8253 <na>
65 ($41) 10 $00ffa680 $00120582 sc8042m <na>
65 ($41) 10 $00e85db0 $00120582 sc8042m <na>
65 ($41) 10 $00e84a40 $00120582 sc8042m <na>
65 ($41) 10 $00e82980 $00120582 sc8042m <na>
74 ($4a) 1 $00ff02d0 $001f9504 spe509 <na>
78 ($4e) 10 $00ff4f30 $00137906 rb1003 <na>
```

In this case, we can go into RomBug by typing `break` and placing a breakpoint at the ISR.

```
$ break
RomBug: b 1f9504
RomBug: g
```

and then pinging a machine on the net:

```
$ ping 182.52.109.13
```

( using the actual address of another machine on the network, rather than the one shown above).

If interrupts are running you should be presented a Rombug prompt at the breakpoint address. You can type `g` to see if you get another interrupt or `k` to kill the breakpoint.

## Port Address Conflict

It is also possible that the port address used for this card is used by another device in the system. If this is the case, the OS-9 command `netstat -in` will not show the card as available.

The following netstat example shows a working network card configured with IP address 182.52.109.25 and MAC address of 00.00.C0.91.4F.96.

(Super)[/h0/sys/>] netstat -in

| Name  | Mtu  | Network    | Address           | Ipkts | Ierrs | Opkts | Oerrs | Coll |
|-------|------|------------|-------------------|-------|-------|-------|-------|------|
| lo0   | 1536 | <Link>     |                   | 0     | 0     | 0     | 0     | 0    |
| lo0   | 1536 | 127        | 127.0.0.1         | 0     | 0     | 0     | 0     | 0    |
| enet0 | 1500 | <Link>     | 00.00.C0.91.4F.96 | 55    | 110   | 35    | 0     | 0    |
| enet0 | 1500 | 182.52.109 | 182.52.109.25     | 55    | 110   | 35    | 0     | 0    |

## Modifying the OS-9 Descriptor

Edit the file

MWOS/OS9000/80386/PORTS/PCAT/SPF/SPE509/DEFS/spf\_desc.h and look for the #ifdef spe30\_isa section.

Change the fields below as required.

INF\_AUI = AUI Connection type

INF\_BNC = BNC connection type

INF\_UPT = 10BaseT (RJ45)

INF\_EXT = Use same connection type determined in 3COM setup program

/\*

\* From spf\_desc.h

\*/

#define PORTADDR0x340/\* IO port for ISA\*/

#define IRQVECTOR0x43/\* IRQ vector \*/

/\* options for CONNTYPE: INF\_AUI, INF\_BNC, INF\_UTP, INF\_EXT (auto) \*/

#define CONNTYPEINF\_EXT

Finally, remake the descriptor: change to the

MWOS/OS9000/80386/PORTS/PCAT/SPF/SPE509 directory and type:

C:> os9make -f=spfdesc.mak

Next time you run the Wizard the new descriptor will be used.

## Low-level System Changes

If system state debugging is used, you must change the low level system by modifying the following lines from the file

MWOS/OS9000/80386/PORTS/PCAT/ROM/cnfgdata.des:

```
#define LLE509_PORT_ADDRESS 0x340
#define LLE509_IF_VECTOR 0x43
```

as required by the system. For example, for IRQ10, here are the changes required.

```
#define LLE509_PORT_ADDRESS 0x340
#define LLE509_IF_VECTOR 0x4a
```

The Wizard will automatically re-make the cnfgdata module.

## 3COM PCMCIA



### Note

When making bootfile only images care should be taken to make sure PCMCIA support is enabled in the low-level 'coreboot' system if PCMCIA devices are to be employed once the system is booted.

3COM EtherLink III PC CARD

3COM Megahertz LAN (3CCE589ET) - 10 Mbps LAN PC Card

## System State Debugging

Supported

## Default Settings

|           |         |                       |
|-----------|---------|-----------------------|
| PORTADDR  | 0x340   | /* IO port for ISA */ |
| IRQVECTOR | 0x43    | /* IRQ vector */      |
| CONNTYPE  | INF_EXT | /* Auto */            |

## Solving Configuration Issues

### Connection Type

The default connection type is set to INF\_EXT (auto). For the 3COM PCMCIA card this implies the card will detect the connection type used. If desired the connection type may be forced. To force the connection type the descriptor must be changed.

### Interrupt Conflict

Another problem may be the interrupt used. The default interrupt is IRQ3. In this case you have the following options.

1. Disable the COM2 serial port from the BIOS to allow IRQ3 to function with this card.
2. Choose a interrupt that matches the system configuration such as IRQ10 (0x4a). In this case the OS-9 device descriptor must be changed. Also the PCMCIA socket services setup must be changed to assign the new interrupt to the PCMCIA Ethernet Card.

If an interrupt conflict exists the device will either not work at all or will hang when the conflicting device is accessed. Mapping the interrupts used in the system is recommended.

Use the command `irqs` to see a list of interrupts.

```
(Super)[/h0/sys/>] irqs
```

```
vector ($)  prior drivstat  irq svc   driver dev list
-----
  7 ($07)   10 $0003c444 $0010f7b4 fpu   <na>
 14 ($0e)    1 $0003c3a4 $00110113 vectors <na>
 64 ($40)   10 $00ff40b0 $0011098f tk8253 <na>
 65 ($41)   10 $00fa680 $00120582 sc8042m <na>
 65 ($41)   10 $00e85db0 $00120582 sc8042m <na>
 65 ($41)   10 $00e84a40 $00120582 sc8042m <na>
 65 ($41)   10 $00e82980 $00120582 sc8042m <na>
 74 ($4a)    1 $00ff02d0 $001f9504 spe509 <na>
 78 ($4e)   10 $00ff4f30 $00137906 rb1003 <na>
```

In the case above we can go into RomBug by typing `break` and placing a break at the ISR.

```
$ break
```

```
RomBug: b 1f9504
```

```
RomBug: g
```

and then pinging a machine on the net.

```
$ ping 182.52.109.13
```

(Using the actual address of another machine on the network, rather than the one shown above.)

If interrupts are running you should be presented a *Rombug* prompt at the breakpoint address. You can type `g` to see if you get another interrupt or `k` to kill the breakpoint.

## Port Address Conflict

It is also possible that the port address used for this card is used by another device in the system. If this is the case the OS-9 command `netstat -in` will not show the card as available.

```
(Super)[/h0/sys/>] netstat -in
```

| Name  | Mtu  | Network    | Address           | lpkts | lerrs | Opkts | Oerrs | Coll |
|-------|------|------------|-------------------|-------|-------|-------|-------|------|
| lo0   | 1536 | <Link>     |                   | 0     | 0     | 0     | 0     | 0    |
| lo0   | 1536 | 127        | 127.0.0.1         | 0     | 0     | 0     | 0     | 0    |
| enet0 | 1500 | <Link>     | 00.00.C0.91.4F.96 | 55    | 110   | 35    | 0     | 0    |
| enet0 | 1500 | 182.52.109 | 182.52.109.25     | 55    | 110   | 35    | 0     | 0    |

## Modifying the OS-9 Descriptor

Edit the file

MWOS/OS9000/80386/PORTS/PCAT/SPF/SPE509/DEFS/spf\_desc.h

Look for the `#ifdef spe30_isa` section and change the `PORTADDR`, `IRQVECTOR`, and `CONNTYPE` as required.

The permissible values for `CONNTYPE` are:

`INF_AUI` = AUI Connection type

`INF_BNC` = BNC connection type

`INF_UPT` = 10BaseT (RJ45)

`INF_EXT` = Probe connection type

```
/*
 * From spf_desc.h
 */

#define PORTADDR0x340 /* IO port for ISA */
#define IRQVECTOR0x43 /* IRQ vector */
/* options for CONNTYPE: INF_AUI, INF_BNC, INF_UTP, INF_EXT (auto) */
#define CONNTYPEINF_EXT
```

Finally, remake the descriptor by changing to the  
MWOS/OS9000/80386/PORTS/PCAT/SPF/SPE509 directory and typing

`os9make -f=spfdesc.mak.`

## Low-level System Changes

System state debugging requires a change to the low level system, as well as the PCMCIA socket services information. This is controlled by the contents of the file

MWOS/OS9000/80386/PORTS/PCAT/ROM/cnfgdata.des.

Find the following lines:

```
#define LLE509_PORT_ADDRESS0x340
#define LLE509_IF_VECTOR0x43
#define ETH_CIS_PARAMS"3com=0x340,3"
```

The above port addresses and/or IRQ information should be changed as required by the system. For IRQ 10, here are the changes required:

```
#define LLE509_PORT_ADDRESS0x340
#define LLE509_IF_VECTOR0x4a
#define ETH_CIS_PARAMS"3com=0x340,10"
```

The Wizard will automatically re-make the cnfgdata module.

## DEC 21140

Intra Server DE504-BA (Quad)

Asante' Fast 10/100

D-Link DFE-500TX ProFast 10/100 Adapter

## System State Debugging

Not Supported

## Default Settings

|           |         |
|-----------|---------|
| PORTADDR  | NA      |
| IRQVECTOR | NA      |
| CONNTYPE  | INF_UTP |



## Solving Configuration Issues

### Connection Type

The default connection type is set to INF\_AUI.

The following netstat example shows a working network card configured with IP address 182.52.109.25 and MAC address of 00.00.C0.91.4F.96.

(Super)[/h0/sys/>] netstat -in

| Name  | Mtu  | Network    | Address           | Ipkts | lerrs | Opkts | Oerrs | Coll |
|-------|------|------------|-------------------|-------|-------|-------|-------|------|
| lo0   | 1536 | <Link>     |                   | 0     | 0     | 0     | 0     | 0    |
| lo0   | 1536 | 127        | 127.0.0.1         | 0     | 0     | 0     | 0     | 0    |
| enet0 | 1500 | <Link>     | 00.00.C0.91.4F.96 | 55    | 110   | 35    | 0     | 0    |
| enet0 | 1500 | 182.52.109 | 182.52.109.25     | 55    | 110   | 35    | 0     | 0    |

### Modifying the OS-9 Descriptor

Edit the file

MWOS/OS9000/80386/PORTS/PCAT/SPF/SP21140/DEFS/spf\_desc.h, changing the definition of CONNTYPE as required. The possible values for CONNTYPE are:

```
*
* From spf_desc.h
*/
/*
* Interface/connection type
* Common values:
*
* INF_UTP    ==  MII_10MB == 10Mb/s 21140
* INF_AUI    ==  SRL_10MB == Conventional 10Mb/s 21140
* INF_UTP100 ==  MII_100MBTX == MII 100Mb/s 21140
* INF_FX100  ==  MII_100MBFX == MII 100Mb/s 21140
* INF_MII10  ==  MII_10MB == 10Mb/s 21140
* INF_MII100 ==  MII_100MB == MII 100Mb/s 21140
*
* Note: Not all common values will work. Below are common
* values used for different cards supported. Much work at
* driver level still remains to allow auto and NWay support.
* Support for DEC21143 may be added in the future once the
* NWay support is added.
*
* Intra Server DE504-BA (Quad)
* 10BaseT = INF_UTP ( note: preliminary release support for 21143. No 100BaseT support)
```

```
*
* Asante' Fast 10/100
*   10BaseT = INF_UTP
* 100BaseT = INF_MII100
*
* D-Link DFE-500TX ProFast 10/100 Adapter
*
*   10BaseT = INF_UTP
*   10BaseT = INF_MII10
* 100BaseT = INF_MII100
*
*/
```

```
#define CONNTYPE   INF_UTP
```

Finally, remake the descriptor: change to the  
MWOS/OS9000/80386/PORTS/PCAT/SPF/SP21140 directory and type:

```
os9make -f=spfdesc.mak
```

You have now created a new descriptor. Next time you run the Wizard, it will use the new descriptor.

## Adding support for Dual and Quad Channel Cards

The descriptors for the additional Ethernet ports must be added. Edit the spf.ml file in the

MWOS/OS9000/80386/PORTS/PCAT/BOOTS/INSTALL/PORTBOOT directory. Find the entry for spde0. Add spde1 for a dual card or spde1, spde2 and spde3 for a quad card.

Next edit the pcat.ini file located in  
MWOS/OS9000/80386/PORTS/PCAT/BOOTS/INSTALL/INI directory. Look for the ETHER\_OPTION\_ string and add the entries as required. You must specify the Ethernet information for all extra Ethernet ports used.

The following example adds the three extra Ethernet ports for a quad card.

```
ETHER_OPTION_2=enet1 address 112.16.1.237 broadcast
112.16.255.255 netmask 255.255.000.000 binding /spde1/enet
```

```
ETHER_OPTION_3=enet2 address 122.16.1.237 broadcast
122.16.255.255 netmask 255.255.000.000 binding /spde2/enet
```

```
ETHER_OPTION_4=enet3 address 132.16.1.237 broadcast
132.16.255.255 netmask 255.255.000.000 binding /spde3/enet
```

Once the boot image is created you may boot OS-9 and use "netstat" to see that all cards are active and ready for use. You should see entries for enet0, enet1, enet2 and enet3 if you are using a quad card.

## AM79C961 & AM79C73A

### System State Debugging

Not Supported

### Default Settings

|           |       |
|-----------|-------|
| PORTADDR  | 0x300 |
| IRQVECTOR | NA    |
| CONNTYPE  | NA    |

### Solving Configuration Issues

The AM79C961A driver is designed to work in systems where DMA BUS MASTER mode is employed with respect to the AM79C961 or AM79C973 interfaces.

The AM79C961A driver is PLUG & PLAY. Only the base address should be defined to allow multiple card usage.

## Modifying the OS-9 Descriptor

1. Edit the file.  
MWOS/OS9000/80386/PORTS/PCAT/SPF/SP79C961/DEFS/spf\_desc.h, changing the line defining PORTADDR, which reads #define PORTADDR 0x300 /\* Base address of hardware \*/, to give PORTADDR the desired value.
2. Next re-make the descriptor: change to the MWOS/OS9000/80386/PORTS/PCAT/SPF/SP79C961 directory and type the command os9make -f=spfdesc.mak

You have now created a new descriptor. The next time you run the Wizard, it will use the new descriptor.

## NE2000

ZF NetDisplay

ACCTON - EN166X MPX 2 Ethernet

D-LINK DE-220PCT - 10Mbps Combo 16-Bit Ethernet ISA Adapter

Compex - ReadyLink 2000 - PCI 32-bit

## System State Debugging

Supported

## Default Settings

```
PORTADDR    0x340      /* IO port for ISA */
IRQVECTOR    0x49      /* IRQ vector */
CONNTYPE     INF_EXT    /* Auto */
```

## Board Setup Issues

### ZF NetDisplay

use <CDROM>:\Drivers\Ethernet\Realtek\RSET8019.EXE"

to determine the IO address and IRQ required.

IO=0x340 VECTOR=0x49 is typical. Settings are system dependent.

### ACCTON - EN166X MPX 2 Ethernet

use "1step" program located on the setup disk to set card to

jumpered "ne2000" mode.

IO=0x300 VECTOR=0x43 is typical. Settings are system dependent.

### D-LINK DE-220PCT - 10Mbps Combo 16-Bit Ethernet ISA Adapter

Use "setup" program located on the setup disk "A:\SETUP\setup.exe" to setup the card. Disable PNP and setup Interrupt and I/O base address.

### Compex - ReadyLink 2000 - PCI 32-bit

Just plug and go. Multiple cards may be used by using the

PCI Specific Settings listed below.

## PCI Specific Settings Information

When using multiple NE2000 PCI cards in a system you may force the driver to use a specific slot or card number for the device being used. PCIINDEX may be used to specify the card instance to be used. Keep in mind the PCIINDEX method is based on a first found basis, so moving cards in the system will change the configuration used. You may also use the PCIBUS and PCIDEV to force the use of the device to a specific slot. To find out the current PCIBUS and PCIDEV values use the OS-9 command *pciv*.

/\*

\* PCI Specific Settings

\*/

#define PCIINDEX0x00/\* 0 picks first card \*/

#define PCIBUS 0x00/\* 0 indicates to search \*/

#define PCIDEV0x00/\* 0 indicates to search \*/

### Connection Type

The default connection type is set by either the configuration setup program that came with the card or by hardware jumpers employed. If you are unable to communicate with this card and netstat -in shows the device, the connection type may be incorrect. To correct it, you may do one of the following.

1. Use the NE2000 setup disk to configure the card for the connection used.
2. Change the OS-9 device descriptor for the type of connection in use.
3. Try one of the other connections on the card (if using AUI type, try the RJ45 connector).

### Interrupt Conflict Options

Another problem may be the interrupt used. The default interrupt is IRQ9. In this case you have the following options.

1. Choose a interrupt that matches the system configuration such as IRQ10 (0x4a). In this case the OS-9 device descriptor must be changed.

If an interrupt conflict exists the device will either not work at all or will hang when the conflicting device is accessed. Mapping the interrupts used in the system is recommended.

Use the command `irqs` to see a list of interrupts.

(Super)[/h0/sys/>] `irqs`

```
vector ($)  prior drivstat  irq svc   driver dev list
-----
  7 ($07)    10 $0003c444 $0010f7b4 fpu    <na>
 14 ($0e)     1 $0003c3a4 $00110113 vectors <na>
 64 ($40)    10 $00ff40b0 $0011098f tk8253  <na>
 65 ($41)    10 $00ffa680 $00120582 sc8042m <na>
 65 ($41)    10 $00e85db0 $00120582 sc8042m <na>
 65 ($41)    10 $00e84a40 $00120582 sc8042m <na>
 65 ($41)    10 $00e82980 $00120582 sc8042m <na>
 74 ($49)     1 $00ff02d0 $001f9504 spne2000 <na>
 78 ($4e)    10 $00ff4f30 $00137906 rb1003  <na>
```

In the case above, we can go into RomBug by typing break and placing a breakpoint at the ISR.

```
$ break
RomBug: b 1f9504
RomBug: g
```

and then pinging a machine on the net:

```
$ ping 182.52.109.13
```

( using the actual address of another machine on the network, rather than the one shown above).

If interrupts are running you should be presented a *Rombug* prompt at the breakpoint address. You can type *g* to see if you get another interrupt or *k* to kill the breakpoint.

### Port Address Conflict

It is also possible that the port address used for this card is used by another device in the system. If this is the case, the OS-9 command netstat -in will not show the card as available.

The following netstat example shows a working network card configured with IP address 182.52.109.25 and MAC address of 00.00.C0.91.4F.96.

```
(Super)[/h0/sys/>] netstat -in
```

| Name  | Mtu  | Network    | Address           | Ipkts | Ierrs | Opkts | Oerrs | Coll |
|-------|------|------------|-------------------|-------|-------|-------|-------|------|
| lo0   | 1536 | <Link>     |                   | 0     | 0     | 0     | 0     | 0    |
| lo0   | 1536 | 127        | 127.0.0.1         | 0     | 0     | 0     | 0     | 0    |
| enet0 | 1500 | <Link>     | 00.00.C0.91.4F.96 | 55    | 110   | 35    | 0     | 0    |
| enet0 | 1500 | 182.52.109 | 182.52.109.25     | 55    | 110   | 35    | 0     | 0    |

## Modifying the OS-9 Descriptor

Edit the file

MWOS/OS9000/80386/PORTS/PCAT/SPF/NE2000/DEFS/spf\_desc.h.

Change the fields below as required.

```
/*
 * From spf_desc.h
 */

#define PORTADDR0x00000340/* Base address of hardware */
#define VECTOR0x49/* Port vector */

/*
 * PCI Specific Settings
 */

#define PCIINDEX0x00/* 0 picks first card */
#define PCIBUS0x00/* 0 indicates to search */
#define PCIDEV0x00/* 0 indicates to search */
```

Finally, remake the descriptor: change to the  
MWOS/OS9000/80386/PORTS/PCAT/SPF/NE2000 directory and type:

C:> os9make -f=spfdesc.mak

Next time you run the Wizard the new descriptor will be used.

## Low-level System Changes

If system state debugging is used, you must change the low level system  
by modifying the following lines from the file

MWOS/OS9000/80386/PORTS/PCAT/ROM/cnfgdata.des:

```
#define LLNE2000_PORT_ADDRESS0x340
#define LLNE2000_IF_VECTOR0x49
```

as required by the system. For example, for IRQ10, here are the changes  
required.

```
#define LLNE2000_PORT_ADDRESS0x340
#define LLNE2000_IF_VECTOR0x4a
```

The Wizard will automatically re-make the cnfgdata module.



## NE2000 PCMCIA

|                  |        |        |
|------------------|--------|--------|
| RealTek RTL-8029 | 0x10ec | 0x8029 |
| Winbond 89C940   | 0x1050 | 0x0940 |
| Winbond w89C940  | 0x1050 | 0x5a5a |
| KTI ET32P2       | 0x8e2e | 0x3000 |
| NetVin NV5000SC  | 0x4a14 | 0x5000 |
| Via 82C926       | 0x1106 | 0x0926 |
| SureCom NE34     | 0x10bd | 0x0e34 |
| Holtek HT80232   | 0x12c3 | 0x0058 |
| Holtek HT80229   | 0x12c3 | 0x5598 |

## Cirrus Logic CS8900

The OS9 sp8900 software driver provides support for the Cirrus Logic CS8900a Ethernet Controller. This allows the device to be used as part of an OS9 SoftStax network implementation.

The Cirrus Logic CS8900a provides single chip support for IEEE 802.3 Ethernet. It has a direct ISA bus interface and is therefore commonly found in PC-AT type environments.

The OS9 sp8900 driver takes advantage, where appropriate, of the Plug and Play capability of the cs8900a device. This reduces the time taken to configure the cs8900a for use within an OS9 environment.

## Hardware Configuration

The CS8900a should be supplied with an MSDOS hosted configuration program. This should be used to pre-configure the device for use. This program assumes the cs8900a has the associated EEPROM as recommended. This EEPROM is used to store the cs8900 configuration parameters. At this time OS9 will only support devices that have this configuration.

## Using the Setup Program

Before using the setup program, you should determine the network adaptor's IO address and Interrupt level. The cs8900 has a limited number of possible combinations, these should be chosen with care. As a default OS9 will assume IO port 0x300 and IRQ Level 10. It is also important to note that OS9 drives the device using the PC-AT I/O Bus for ALL operations. Therefore shared memory should be disabled for OS9 operation.

Having selected the correct choices you may run the setup program and configure the cs8900 accordingly.

If the device was supplied without a configuration utility it will be necessary to obtain this from the vendor or try the cirrus logic Web site at <http://www.cirrus.com/drivers/>

The setup program also incorporates a self test utility that may be used to confirm correct operation of the device before proceeding.

## OS9 Software Configuration

### Configuring PnP Firmware

The OS9 sp8900 driver will use the PnP (Plug and Play) capability of the cs8900a. This will only be used if it is enabled in the OS9 device descriptor. When enabled the OS9 driver will search all possible I/O locations for a cs8900a device. If found, the first one, starting at the lowest valid I/O address, will be used. The software will confirm that the EEPROM is present. The OS9 driver extracts the necessary configuration details from this device and initializes the cs8900a.

### Configuring OS9 Descriptors

The OS9 device descriptor allows you to override the PnP default configuration. At this time only a subset of all the possible configuration parameters may be overridden. To change the PnP values the following fields must be modified. This should be performed using a text editor and the OS9 tools provided within the Microware Hawk package.

Once modified the descriptor should be regenerated and tested.

## Device Descriptor Fields

The standard device descriptor is as follows. This file may be found in

```
.../MWOS/OS9000/<processor>/PORTS/<port>/SPF/SP8900/DEFS/spf_desc.h
#define SPF_DIR_NONE0xFF
#define SPF_DIR_IN0x00
#define SPF_DIR_OUT0x01

#include <SPF/item.h>

#ifdef spcs0

/** Device Descriptor for SPF 8900 ethernet driver */
#define PNPON 1 /* do plug and play */
#define PNPOFF 0 /* Use descriptor values ( see manual ) */

/*****
 * User configuration defines
 *****/
/* Port configuration defines */

/* Macros that initialize device descriptor common fields */

/* 300/320/340/360 */

#define PORTADDR0x300 /* Base address of hardware */
#define LUN0x7F /* logical unit number */

#define VECTOR0x4a /* Port vector */
#define PRIORITY8 /* IRQ polling priority */
#define IRQLEVEL0 /* Port IRQ Level */
#define PNP8900PNPON /* Do plug and play ( Normal setting ) */

#define TB486COMPATTRUE
-----*/
```

Any information ( not shown ) beyond this point MUST not be changed.

## User Configurable Fields

The following fields are user configurable.

| Field Name   | Default Value | Possible Values     |
|--------------|---------------|---------------------|
| PORTADDR     | 0x300         | 0x200..0x360        |
| VECTOR       | 0x4a          | 0x45,0x4a,0x4b,0x4c |
| PRIORITY     | 0x08          | 0..255              |
| PNP8900      | PNPON         | PNPON or PNPOFF     |
| TB486COMPAT* | TRUE          | TRUE or FALSE       |

\*note: The tb486 board is a special case and this flag should be set false for any other board type.

## Generating a New Device Descriptor

Having located and edited the field as desired the new device descriptor may be generated with the following steps

Change directory to: ../MWOS/OS9000/80386/PORTS/PCAT/SPF/SP8900

Enter the command: `os9make -f=sppfdesc.mak -u MOPTS=-u`

The new descriptor will be built.

## OS9 SP8900 Components

The complete driver consists of two OS9 load modules. You should refer to the appropriate Microware manual for further information concerning system configuration.

The SP8900 component files are :

SP8900 -- cs8900a Ethernet Driver

spcs0 -- cs8900 Device Descriptor

## NetGear

FA311

## FA312



### Note

NetGear FA311 and FA312 support autosense of 10/100BaseT Full/Half Duplex and contain multicasting support.

The high-level driver for these cards is `SPFA311`. The low-level driver is `LLFA311`.

## LAN

LAN79C961

LANAM79C973

The low-level driver for this card is `LL79C961`. The high-level driver is `SP79C961`.



### Note

Multicasting support is available for the `SP79C961`.

## RealTek

RL8139

The low-level driver for this card is `LL8139`. The high-level driver is `SP8139`.

## SMC

SMC91C94

SMC91C96

The low-level driver for this card is `LLC91C94`. The high-level driver is `SP91C94`.

## INTEL Ethernet PRO100 Series

82557

82558

82559

The low-level driver for this card is `LLPRO100`. The high level driver is `SPPRO100`.

# Sequential Device Support

---

## VGA Graphics / Keyboard

VGA support is provided using standard VGA graphics screen and keyboard. Most PC based systems use VGA keyboard as the default device for user input. While this is not required for OS-9 based systems it is a convenient way to initially setup systems for use with OS-9.

During the development of MAUI user applications, a serial console may be the preferred method since the text based console may interfere with the graphics application on the same device.

MULTI-TERM is a feature of the VGA Graphics/Keyboard console driver which provides up to four virtual screens. If you are a console user, you may switch between screens by pressing an alternate function key combination, such as <Alt> <F1>, <Alt> <F2>, <Alt><F3> or <Alt><F4>. MULTI-TERM may be started automatically in the /h0/sys/startup file or manually from the console by executing the following commands:

```
$ mshell -l <>>>/mterm1&
$ mshell -l <>>>/mterm2&
$ mshell -l <>>>/mterm3&
```

### VGA TERMINAL Descriptors Notes

```
/mterm0Multi-term descriptor 0
/mterm1Multi-term descriptor 1
/mterm2Multi-term descriptor 2
/mterm3Multi-term descriptor 3
```

The following optional settings apply to the VGA/Keyboard console:

```
#define DS_ROMBREAK1/* Enter RomBug - Shift PrintScreen. */
#define DS_RESTART1/* Reset System - Ctrl/Alt/Del. 0=disabled */
#define DS_NUM_LOCK1/* Keyboard Number lock 0=off 1=on */
#define DS_SHIFT_LOCK0/* Keyboard Caps lock 0=off 1=on */
```

To change these options, edit the file  
MWOS/OS9000/80386/PORTS/PCAT/SCF/SC8042M/config.des. Find the  
sections as outlined above. Change as desired. Then, change to the  
MWOS/OS9000/80386/PORTS/PCAT/SCF/SC8042M/DRVR directory and  
type os9make.

## Language Support Options

To change the language support for the keyboard use the advanced mode  
from the Wizard and select BOOTFILE OPTIONS tab. Select the language  
desired.

```
MWOS\OS9000\80386\PORTS\PCAT\CMDS\BOOTOBS\DESC\SC8042M\term0
MWOS\OS9000\80386\PORTS\PCAT\CMDS\BOOTOBS\DESC\SC8042M\mterm0
MWOS\OS9000\80386\PORTS\PCAT\CMDS\BOOTOBS\DESC\SC8042M\mterm1
MWOS\OS9000\80386\PORTS\PCAT\CMDS\BOOTOBS\DESC\SC8042M\mterm2
MWOS\OS9000\80386\PORTS\PCAT\CMDS\BOOTOBS\DESC\SC8042M\mterm3
MWOS\OS9000\80386\PORTS\PCAT\CMDS\BOOTOBS\DESC\SC8042M\term0_fr
MWOS\OS9000\80386\PORTS\PCAT\CMDS\BOOTOBS\DESC\SC8042M\mterm0_fr
MWOS\OS9000\80386\PORTS\PCAT\CMDS\BOOTOBS\DESC\SC8042M\mterm1_fr
MWOS\OS9000\80386\PORTS\PCAT\CMDS\BOOTOBS\DESC\SC8042M\mterm2_fr
MWOS\OS9000\80386\PORTS\PCAT\CMDS\BOOTOBS\DESC\SC8042M\mterm3_fr
MWOS\OS9000\80386\PORTS\PCAT\CMDS\BOOTOBS\DESC\SC8042M\term0_gr
MWOS\OS9000\80386\PORTS\PCAT\CMDS\BOOTOBS\DESC\SC8042M\mterm0_gr
MWOS\OS9000\80386\PORTS\PCAT\CMDS\BOOTOBS\DESC\SC8042M\mterm1_gr
MWOS\OS9000\80386\PORTS\PCAT\CMDS\BOOTOBS\DESC\SC8042M\mterm2_gr
MWOS\OS9000\80386\PORTS\PCAT\CMDS\BOOTOBS\DESC\SC8042M\mterm3_gr
MWOS\OS9000\80386\PORTS\PCAT\CMDS\BOOTOBS\DESC\SC8042M\term0_nw
MWOS\OS9000\80386\PORTS\PCAT\CMDS\BOOTOBS\DESC\SC8042M\mterm0_nw
MWOS\OS9000\80386\PORTS\PCAT\CMDS\BOOTOBS\DESC\SC8042M\mterm1_nw
MWOS\OS9000\80386\PORTS\PCAT\CMDS\BOOTOBS\DESC\SC8042M\mterm2_nw
MWOS\OS9000\80386\PORTS\PCAT\CMDS\BOOTOBS\DESC\SC8042M\mterm3_nw
MWOS\OS9000\80386\PORTS\PCAT\CMDS\BOOTOBS\DESC\SC8042M\term0_uk
MWOS\OS9000\80386\PORTS\PCAT\CMDS\BOOTOBS\DESC\SC8042M\mterm0_uk
MWOS\OS9000\80386\PORTS\PCAT\CMDS\BOOTOBS\DESC\SC8042M\mterm1_uk
MWOS\OS9000\80386\PORTS\PCAT\CMDS\BOOTOBS\DESC\SC8042M\mterm2_uk
MWOS\OS9000\80386\PORTS\PCAT\CMDS\BOOTOBS\DESC\SC8042M\mterm3_uk
```



## Serial Mouse

Configuration modules for a Serial Mouse is included in the system image when the Mouse option is not selected in the Configuration Wizard's Master Builder screen. Serial mouse support is only included when sc16550 support is enabled in the Configuration Wizards BOOTFILE OPTIONS dialog box.

The default port is COM1. The MWOS/OS9000/80386/PORTS/PCAT/BOOTS/INSTALL/PORTBOOT/bootfile.ml file may be changed to allow a different port to be used.

Default (Serial Mouse configured using COM1)

```
*
* [OPTION4 && !MOUSE] serial mouse
*
../../../../CMDS/BOOTOBJS/DESC/SC16550/m0_t1
../../../../CMDS/BOOTOBJS/DESC/SC16550/m0_t2
../../../../CMDS/BOOTOBJS/DESC/SC16550/m0_t3
../../../../CMDS/BOOTOBJS/DESC/SC16550/m0_t4
*
```

Changed to use COM3

```
*
* [OPTION4 && !MOUSE] serial mouse
*
../../../../CMDS/BOOTOBJS/DESC/SC16550/m0_t1
../../../../CMDS/BOOTOBJS/DESC/SC16550/m0_t2
../../../../CMDS/BOOTOBJS/DESC/SC16550/m0_t3
../../../../CMDS/BOOTOBJS/DESC/SC16550/m0_t4
```

## PS2 Mouse

PS2 mouse support is automatically included when the Mouse option is selected from the Configuration Wizard's Master Build screen.

## 16550 Serial

Standard PC type serial ports are supported. By default, four descriptors are available, but you may add more as needed.

Use of COM1 and COM2 are standard on PC based systems. COM3 and COM4 are not. Since COM1 and COM2 use IRQ3 and IRQ4, most systems will not allow COM3 and COM4 to also use IRQ3 and or IRQ4. The main reason for this is that IRQ3 and IRQ4 are normally edge based interrupts, and the 16550 is normally implemented in a edge based configuration. Therefore, anytime COM3 and or COM4 are used, you must determine the interrupt vector to use for these ports.

To change the vector you must edit the `systype.h` file located in the port directory.

MWOS/OS9000/80386/PORTS/PCAT/systype.h

```
#define T1BASE_165500x000003f8/* SC16550 port 1 */
#define T1VECT_165500x44/* IRQ 4 */
#define T1PRI_165505/* Priority */

#define T2BASE_165500x000002f8/* SC16550 port 2 */
#define T2VECT_165500x43/* IRQ 3 */
#define T2PRI_165505/* Priority */

#define T3BASE_165500x000003e8/* SC16550 port 3 */

#define T3VECT_165500x44/* IRQ 4 */
#define T3PRI_1655010/* Priority */

#define T4BASE_165500x000002e8/* SC16550 port 4 */
#define T4VECT_165500x43/* IRQ 3 */
#define T4PRI_1655010/* Priority */
```

## Making the Descriptors

Once the `systype.h` file has been updated the new descriptors may be created.

Change to directory:

`MWOS/OS9000/80386/PORTS/PCAT/SCF/SC16550/DESC`

Type `os9make`; the following descriptors will be made:

```
MWOS\OS9000\80386\PORTS\PCAT\CMD5\BOOTOBJS\DESC\SC16550\term1
MWOS\OS9000\80386\PORTS\PCAT\CMD5\BOOTOBJS\DESC\SC16550\t1
MWOS\OS9000\80386\PORTS\PCAT\CMD5\BOOTOBJS\DESC\SC16550\term2
MWOS\OS9000\80386\PORTS\PCAT\CMD5\BOOTOBJS\DESC\SC16550\t2
MWOS\OS9000\80386\PORTS\PCAT\CMD5\BOOTOBJS\DESC\SC16550\term3
MWOS\OS9000\80386\PORTS\PCAT\CMD5\BOOTOBJS\DESC\SC16550\t3
MWOS\OS9000\80386\PORTS\PCAT\CMD5\BOOTOBJS\DESC\SC16550\term4
MWOS\OS9000\80386\PORTS\PCAT\CMD5\BOOTOBJS\DESC\SC16550\t4
MWOS\OS9000\80386\PORTS\PCAT\CMD5\BOOTOBJS\DESC\SC16550\ps
MWOS\OS9000\80386\PORTS\PCAT\CMD5\BOOTOBJS\DESC\SC16550\m0_t1
MWOS\OS9000\80386\PORTS\PCAT\CMD5\BOOTOBJS\DESC\SC16550\m0_t2
MWOS\OS9000\80386\PORTS\PCAT\CMD5\BOOTOBJS\DESC\SC16550\m0_t3
MWOS\OS9000\80386\PORTS\PCAT\CMD5\BOOTOBJS\DESC\SC16550\m0_t4
```

## Digiboard

Support for the Digiboard intelligent serial card is included by selecting the Digiboard option in the Configuration Wizard's Bootfile Options dialog box.

To change the vector you must edit the `systype.h` file located in the port directory.

`MWOS/OS9000/80386/PORTS/PCAT/systype.h`

```
#define DIGIPORT    0xe0      /* port address of DIGI board status reg. */
#define DIGILEVEL   0x45      /* 16450 keyboard controller */
#define DIGIVECTOR  DIGILEVEL /* irq vector same as irq level */

#define T10PORT     0x320     /* t10 onboard port address */
#define T11PORT     0x328     /* t11 onboard port address */
#define T12PORT     0x330     /* t12 onboard port address */
#define T13PORT     0x338     /* t13 onboard port address */
#define T14PORT     0x340     /* t14 onboard port address */
#define T15PORT     0x348     /* t15 onboard port address */
#define T16PORT     0x350     /* t16 onboard port address */
#define T17PORT     0x358     /* t17 onboard port address */
```

## Making the Descriptors

Once the `systype.h` file has been updated the new descriptors may be created.

Change to directory:

`MWOS/OS9000/80386/PORTS/PCAT/SCF/SCPC8/DESC`

Type *os9make*; the following descriptors will be made:

```
MWOS\OS9000\80386\PORTS\PCAT\CMDS\BOOTOBJS\DESC\SCPC8\t10
MWOS\OS9000\80386\PORTS\PCAT\CMDS\BOOTOBJS\DESC\SCPC8\t11
MWOS\OS9000\80386\PORTS\PCAT\CMDS\BOOTOBJS\DESC\SCPC8\t12
MWOS\OS9000\80386\PORTS\PCAT\CMDS\BOOTOBJS\DESC\SCPC8\t13
MWOS\OS9000\80386\PORTS\PCAT\CMDS\BOOTOBJS\DESC\SCPC8\t14
MWOS\OS9000\80386\PORTS\PCAT\CMDS\BOOTOBJS\DESC\SCPC8\t15
MWOS\OS9000\80386\PORTS\PCAT\CMDS\BOOTOBJS\DESC\SCPC8\t16
MWOS\OS9000\80386\PORTS\PCAT\CMDS\BOOTOBJS\DESC\SCPC8\t17
```

## Hostessl

Support for the Hostessl intelligent serial card is included by selecting the Hostessl option in the Configuration Wizard's Bootfile Options dialog box.

To change the vector you must edit the `systype.h` file located in the port directory.

`MWOS/OS9000/80386/PORTS/PCAT/systype.h`

```
#define HS_PORT0x00000218/* Hostess i board. serial adapter board */
#define HS_VECT0x4f/* IRQ 15 */
#define HS_BOARDMEM0xd0000/* onboard memory place in the system address space */
#define HS_NBLINES16/* number lines on the board (8/16) */
```

```
/* Old board doesn't permit 16 bits mode. */
#define HS_BUSSIZE8/* size of the bus the board uses (8/16) */
```

## Making the Descriptors

Once the `systype.h` file has been updated the new descriptors may be created.

Change to directory:

`MWOS/OS9000/80386/PORTS/PCAT/SCF/SCHOST/DESC`

Type `os9make` the following descriptors will be made:

```
MWOS\OS9000\80386\PORTS\PCAT\CMDS\BOOTOBS\DESC\SCHOST\t40
MWOS\OS9000\80386\PORTS\PCAT\CMDS\BOOTOBS\DESC\SCHOST\t41
MWOS\OS9000\80386\PORTS\PCAT\CMDS\BOOTOBS\DESC\SCHOST\t42
MWOS\OS9000\80386\PORTS\PCAT\CMDS\BOOTOBS\DESC\SCHOST\t43
MWOS\OS9000\80386\PORTS\PCAT\CMDS\BOOTOBS\DESC\SCHOST\t44
MWOS\OS9000\80386\PORTS\PCAT\CMDS\BOOTOBS\DESC\SCHOST\t45
MWOS\OS9000\80386\PORTS\PCAT\CMDS\BOOTOBS\DESC\SCHOST\t46
MWOS\OS9000\80386\PORTS\PCAT\CMDS\BOOTOBS\DESC\SCHOST\t47
MWOS\OS9000\80386\PORTS\PCAT\CMDS\BOOTOBS\DESC\SCHOST\t48
MWOS\OS9000\80386\PORTS\PCAT\CMDS\BOOTOBS\DESC\SCHOST\t49
MWOS\OS9000\80386\PORTS\PCAT\CMDS\BOOTOBS\DESC\SCHOST\t50
MWOS\OS9000\80386\PORTS\PCAT\CMDS\BOOTOBS\DESC\SCHOST\t51
MWOS\OS9000\80386\PORTS\PCAT\CMDS\BOOTOBS\DESC\SCHOST\t52
MWOS\OS9000\80386\PORTS\PCAT\CMDS\BOOTOBS\DESC\SCHOST\t53
MWOS\OS9000\80386\PORTS\PCAT\CMDS\BOOTOBS\DESC\SCHOST\t54
MWOS\OS9000\80386\PORTS\PCAT\CMDS\BOOTOBS\DESC\SCHOST\t55
```

## Risicom

Support for the Risicom8 intelligent serial card is included.

To change the vector you must edit the `systype.h` file located in the port directory.

`MWOS/OS9000/80386/PORTS/PCAT/systype.h`

```
#define RC8BASE0x00000220/* Risicom8 serial port adapter */
#define RC8VECT0x45/* IRQ 5 */
```

## Making the Descriptors

Once the systype.h file has been updated the new descriptors may be created.

Change to directory:

MWOS/OS9000/80386/PORTS/PCAT/SCF/SCPC8/DESC

Type os9make; the following descriptors will be made:

```
MWOS\OS9000\80386\PORTS\PCAT\CMDS\BOOTOBJS\DESC\SCRISCOM\t20
MWOS\OS9000\80386\PORTS\PCAT\CMDS\BOOTOBJS\DESC\SCRISCOM\t21
MWOS\OS9000\80386\PORTS\PCAT\CMDS\BOOTOBJS\DESC\SCRISCOM\t22
MWOS\OS9000\80386\PORTS\PCAT\CMDS\BOOTOBJS\DESC\SCRISCOM\t23
MWOS\OS9000\80386\PORTS\PCAT\CMDS\BOOTOBJS\DESC\SCRISCOM\t24
MWOS\OS9000\80386\PORTS\PCAT\CMDS\BOOTOBJS\DESC\SCRISCOM\t25
MWOS\OS9000\80386\PORTS\PCAT\CMDS\BOOTOBJS\DESC\SCRISCOM\t26
MWOS\OS9000\80386\PORTS\PCAT\CMDS\BOOTOBJS\DESC\SCRISCOM\t27
```



### WARNING

We have not been able to obtain a Riscom8 board to verify this driver with this release.

## Parallel Printer

Standard PC style printer support is included.

To change the vector or port address you must edit the `systype.h` file located in the port directory.

MWOS/OS9000/80386/PORTS/PCAT/systype.h

```
#define PLEVEL0x47/* scp87303 parallel port */
#define PVECTPLEVEL/* irq vector same as irq level */
#define LPT1BASE0x000003bc/* base address of first parallel port */
#define LPT2BASE0x00000378/* base address of second parallel port */
#define LPT3BASE0x00000278/* base address of third parallel port */
```

## Making the Descriptors

Once the systype.h file has been updated the new descriptors may be created.

Change to directory:

MWOS/OS9000/80386/PORTS/PCAT/SCF/SCP87303/DESC

Type os9make; the following descriptors will be made:

```
MWOS\OS9000\80386\PORTS\PCAT\CMD5\BOOTOBJ5\DESC\SCP87303\p.lp1
MWOS\OS9000\80386\PORTS\PCAT\CMD5\BOOTOBJ5\DESC\SCP87303\p.lp2
MWOS\OS9000\80386\PORTS\PCAT\CMD5\BOOTOBJ5\DESC\SCP87303\p.lp3
```

# Physical Disk Media

---

## IDE Standard

Support for IDE based devices, including standard IDE based hard disk. Primary and secondary controllers with master and slave drive support. On some embedded systems Compact Flash supported devices may be used as if they were standard PC AT based devices.

### Benefits

- Supports large media (8.5GB maximum).
- PIO mode three supported.
- PC File system supported including long filenames (FAT32 is not supported). Boot support (requires OS-9 coreboot load).
- Native RBF file system supported. Full boot support including IPL boot technology.

The standard configuration assumes the primary controller is located at 0x1f0 with IRQ 14 and secondary controller at 0x170 with IRQ 15. You may, however, change these values as needed to suit the target. The values are based on the contents of the files  
MWOS/OS9000/80386/PORTS/PCAT/RBF/RB1003/config.des and  
MWOS/OS9000/80386/PORTS/PCATsystype.h.



The pertinent lines in `MWOS/OS9000/80386/PORTS/PCATsysstype.h` are shown below:

```
#if defined(RB1003_SPEC_IO_ADDRESS) /* PCMCIA */

#defineBASE_RB1003_PRI 0x00000320/* IDE controller port addr */
#defineVECT_RB1003_PRI0x0/* IDE controller vector */
#defineBASE_RB1003_SEC0x00000360/* IDE 2nd controller port */
#defineVECT_RB1003_SEC0x0/* IDE 2nd controller vector */

#else

#defineBASE_RB1003_PRI0x000001f0/* IDE controller port addr */
#defineVECT_RB1003_PRI0x4e/* IDE controller vector */
#defineBASE_RB1003_SEC0x00000170/* IDE 2nd controller port */

#defineVECT_RB1003_SEC0x4f/* IDE 2nd controller vector */

#endif
```

while in `MWOS/OS9000/80386/PORTS/PCAT/RBF/RB1003/config.des` the portion of interest is shown below:

```
/*
 * Device specific defines
 *
 * ds_idetype = IDE interface type
 *     IDE_TYPE_STANDARD
 *     IDE_TYPE_PCI
 *     IDE_TYPE_PCMCIA
 *
 * ds_polled = IDE_POLLED
 *     IDE_INTERRUPTS
 *
 * ds_altstat = HD_DEFAULT_ALTSTAT (Standard IDE offset)
 *
 *     HD_PCMCIA_ALTSTAT (PCMCIA IDE offset)
 *
 * ds_timeout = Drive ready timeout in seconds.
 *     IDE specification allows for up to
 *     30 seconds. We will allow the max here.
 *     Users are free to reduce this amount
 *     if desired. PCMCIA IDE FLASH type cards
 *     require only a few milliseconds. Rotating
 *     devices will require more time.
 */

#define IDE_TYPE_STANDARD 0

#define IDE_TYPE_PCI 1
#define IDE_TYPE_PCMCIA 2
```

```
#define IDE_INTERRUPTS 0
#define IDE_POLLED 1

#define HD_DEFAULT_ALTSTAT 0x0206
#define HD_PCMCIA_ALTSTAT 0xe

init dev_specific {

#if defined(RB1003_SPEC_IO_ADDRESS)
    ds_idetype = IDE_TYPE_PCMCIA;
    ds_polled = IDE_POLLED;
    ds_altstat = HD_PCMCIA_ALTSTAT;
    ds_timeout = 30;

#else

    ds_idetype = IDE_TYPE_STANDARD;
    ds_polled = IDE_INTERRUPTS;
    ds_altstat = HD_DEFAULT_ALTSTAT;
    ds_timeout = 30;

#endif
};
```



## Note

Since OS-9 does not require the BIOS to use IDE it is possible on some systems to use IDE without interrupts. Keep in mind that on some systems disabling the IDE from the BIOS also disables the IDE controller as well.

Drive time-out may also fail on drives that are extremely old. If you are having problems using drives that are less than 540MB you may want to disable the time-out. This can be done by setting time-out value to zero in config.des and re-making the descriptors and boot image.

## Using IDE in PCI Mode

Support is included to support IDE devices as PCI specific devices. PCI based IDE support is not automatic and may not work on some PCI bridges. The `rb1003` driver must be re-made with the following changes to the makefile.

```
PCILIB      =      -I=$(PORT)/LIB/pcilib.l

LIB         =      $(PICLIB) $(PCILIB) \
                  $(CPULIB) $(CLIB) $(P2LIB) $(OS_LIB) $(SYS)

SPEC_COPTS  =      -a -c -r -t=0 -bepg -dNEWINFO $(PICISR) $(IRQMASK) \
                  -dPCI
```

In this case we have added PCILIB as well as defined PCI in the SPEC\_COPTS section. On some systems that use both primary and secondary controllers that allow level interrupt to be set and used in PCI standard method, you can save one interrupt vector. You must also set the device type to PCI in the config.des file shown above. You must have the sources for RB1003 for the ability to make this change using the cross hosted utilities.

If the PCI bridge does not work in PCI mode you can modify the RB1003 init code as need for the PCI bridge device used. The sources are located in `MWOS/OS9000/SRC/IO/RBF/DRVR/RB1003`, and are included with the Embedded Systems package.

Use of IDE in PCI mode adds about 2K to the driver size.

## RBF

OS-9 RBF native file system may be used on any IDE drive. For more information see BootGen and [IDE Descriptors](#).

## PCF

A PC style file system is also supported. FAT32, however, is not supported in this release. If access to partitions other than the primary are required you may use the pinfo utility to obtain the information required to create specific device descriptors. For more information see [IDE Descriptors](#). You may select the PCF file system as the boot media.

For example. If the drive is Fat (not Fat32) you may place the bootfile image on the root. Make sure it is called `os9kboot`. Next, create a CMDS and SYS directory at the root level. Copy whatever CMDS you need to the CMDS directory. Create a startup and or password file as needed. This method allows you to use the same partition as Windows95 or NT when you actually run OS-9.

Prepare Windows95/NT based system for use with OS-9.

```
md C:\CMDS
```

```
md C:\SYS
```

```
copy MWOS\OS9000\80386\CMDS\* C:\CMDS
```

```
copy MWOS\OS9000\80386\PORTS\PCAT\BOOTS\INSTALL\SYS\MSHELL\startup C:\SYS
```

```
copy MWOS\OS9000\80386\PORTS\PCAT\BOOTS\INSTALL\SYS\MSHELL\password C:\SYS
```

```
cd C:\SYS
```

```
cudo -cdo startup
```

```
cudo -cdo password
```

Although RBF is the preferred file system for use with OS-9 the convenience of using FAT file systems should be taken into consideration when deciding how you want to setup your system.

## Special Note

In the following example the IDE device for /h0 and /dd is set for IDE primary partition four.

If the init dialog is set to /h0 the following is generated. In this case we also have SoftStax SPF enabled.

```
setenv SHELL mshell; alias /dd /hc4;chd /h0 ; chx /h0/cmds;mbinstall;
```

```
ipstart;inetd <>>>/nil&./h0/sys/startup &\n
```

If the init dialog is set to /dd the following is generated. In this case we also have SPF enabled.

```
setenv SHELL mshell; alias /dd /hc4;chd /dd ; chx /dd/cmds;mbinstall ;
```

```
ipstart;inetd <>>>/nil&./dd/sys/startup &\n
```

In both cases the script file on hc4 in sys/startup will be executed. When building systems this file must exist, but does not have to contain data. The following commands suffice to create the expected directory and file:

```
$ mkdir /hc4/SYS
```

```
$ touch /hc4/startup
```

It is usually best to create the initial boot image to not use /h0. /dd should be set for RAM disk. This will allow downloading the TAR images. Next setup the final boot image and select /h0 as initial device name.

## Descriptors

Refer to [IDE Descriptors](#) for information on descriptor naming conventions. The descriptors for RB1003 are located in MWOS/OS9000/80386/PORTS/PCAT/CMDS/BOOTOBJS/DESC/RB1003. Also the RB1003 driver is located in MWOS/OS9000/80386/PORTS/PCAT/CMDS/BOOTOBJS.

## ROM BOOTING

If changes to the IDE addresses of time-out values are employed, then the ROM boot system may also require changes.

MWOS/OS9000/80386/PORTS/PCAT/ROM/cnfgdata.des

Find the following sections:

```
#define IDE_PRIMARY_PARAMS_PORT"port=0x1f0 timeout=30"
#define IDE_SECONDARY_PARAMS_PORT"port=0x170 timeout=30"
```

To remove time-out for example we could change the above to:

```
#define IDE_PRIMARY_PARAMS_PORT"port=0x1f0 timeout=0"
#define IDE_SECONDARY_PARAMS_PORT"port=0x170 timeout=0"
```

Or we could make the time-out shorter. IDE specification indicates we can wait up to 30 seconds.

```
#define IDE_PRIMARY_PARAMS_PORT"port=0x1f0 timeout=5"
#define IDE_SECONDARY_PARAMS_PORT"port=0x170 timeout=2"
```

## Advanced Notes

Some embedded systems include support for Compact Flash, which looks like a standard IDE device. In these cases, we may decide that RBF is the file system of choice, since we can boot the embedded board with no other boot devices installed. How do we place the RBF file system on such small embedded systems? Compact Flash devices will work in PCMCIA systems with a carrier, so that we can use a standard PC with PCMCIA support to build up the PCMCIA disk. Once the disk is built, we can then remove the Compact Flash from the carrier and place it in the target system for use.

## PCMCIA IDE



### Note

Microware PCMCIA socket services are included with all PCMCIA selections.



## Note

When making bootfile only images care should be taken to make sure PCMCIA support is enabled in the low-level 'coreboot' system if PCMCIA devices are to be employed once the system is booted.

Support for IDE based devices including standard PCMCIA IDE based hard disk.

## Benefits

- Supports large media(8.5GB maximum).
- PIO mode three supported.
- PC File system supported including long filenames (FAT32 is not supported). Boot support (requires OS-9 coreboot load).
- Native RBF file system supported. Full boot support including IPL boot technology (PCMCIA BIOS BOOT support required if this option is used).
- Requires no interrupts. Interrupts are optional.

The standard configuration assumes socket #0 is mapped to 0x320 and socket #1 is mapped to 0x360. The default configuration does not use interrupts. You may however enable interrupts if desired.

Example (Enable interrupts on PCMCIA device in socket #0 only - IRQ5 used)

```
/*
 * MWOS/OS9000/80386/PORTS/PCATsystype.h file.
 */
#defineBASE_RB1003_PRI0x00000320/* IDE controller port addr */
#defineVECT_RB1003_PRI0x45/* IDE controller vector */
/*
 * MWOS/OS9000/80386/PORTS/PCAT/RBF/RB1003/config.des
 */
```

```
ds_idetype = IDE_TYPE_PCMCIA;
ds_polled = IDE_INTERRUPTS;
ds_altstat = HD_PCMCIA_ALTSTAT;
ds_timeout = 30;
```

MWOS/OS9000/80386/PORTS/PCAT/ROM/cnfgdata.des

```
#define IDE_CIS_PARAMS "ide0=0x320,5 ide1=0x360,0"
```

Once the changes are made change to the MWOS/OS9000/80386/PORTS/PCAT/RBF/RB1003/DESC directory and type os9make. The changes to MWOS/OS9000/80386/PORTS/PCAT/RBF/RB1003/config.des are automatically taken care of next time you run the Wizard.

Changes to the default values are based on the MWOS/OS9000/80386/PORTS/PCAT/RBF/RB1003/config.des file as well as the MWOS/OS9000/80386/PORTS/PCATsystype.h file.

MWOS/OS9000/80386/PORTS/PCATsystype.h

```
#if defined(RB1003_SPEC_IO_ADDRESS) /* PCMCIA */

#defineBASE_RB1003_PRI0x00000320/* IDE controller port addr */
#defineVECT_RB1003_PRI0x0 /* IDE controller vector */
#defineBASE_RB1003_SEC0x00000360/* IDE 2nd controller port */
#defineVECT_RB1003_SEC0x0/* IDE 2nd controller vector */

#else

#defineBASE_RB1003_PRI0x000001f0/* IDE controller port addr */
#defineVECT_RB1003_PRI0x4e/* IDE controller vector */

#defineBASE_RB1003_SEC0x00000170/* IDE 2nd controller port */
#defineVECT_RB1003_SEC0x4f/* IDE 2nd controller vector */

#endif
```

MWOS/OS9000/80386/PORTS/PCAT/RBF/RB1003/config.des

```
/*
 * Device specific defines
 *
 * ds_idetype = IDE interface type
 * IDE_TYPE_STANDARD
 * IDE_TYPE_PCI
 * IDE_TYPE_PCMCIA
 *
 * ds_polled = IDE_POLLED
 * IDE_INTERRUPTS
 *
```



```

* ds_altstat = HD_DEFAULT_ALTSTAT (Standard IDE offset)

*      HD_PCMCIA_ALTSTAT (PCMCIA IDE offset)
*
* ds_timeout = Drive ready timeout in seconds.
*      IDE specification allows for up to
*      30 seconds. We will allow the max here.
*      Users are free to reduce this amount
*      if desired. PCMCIA IDE FLASH type cards
*      require only a few milliseconds. Rotating
*      devices will require more time.
*
*/

#define IDE_TYPE_STANDARD0

#define IDE_TYPE_PC11
#define IDE_TYPE_PCMCIA2

#define IDE_INTERRUPTS0
#define IDE_POLLED1

#define HD_DEFAULT_ALTSTAT0x0206
#define HD_PCMCIA_ALTSTAT0xe

init dev_specific {

#if defined(RB1003_SPEC_IO_ADDRESS)
    ds_idetype = IDE_TYPE_PCMCIA;
    ds_polled = IDE_POLLED;
    ds_altstat = HD_PCMCIA_ALTSTAT;
    ds_timeout = 30;

```

```
#else  
  
    ds_idetype = IDE_TYPE_STANDARD;  
    ds_polled = IDE_INTERRUPTS;  
    ds_altstat = HD_DEFAULT_ALTSTAT;  
    ds_timeout = 30;  
  
#endif  
};
```

## RBF

OS-9 RBF native file system may be used on any IDE drive including PCMCIA devices. For more information see BootGen and IDE DESCRIPTORS. When using RBF with PCMCIA only OS-9 will be able to access the media. When running FDISK on PCMCIA media, be sure to write down the ID type. You will need this value if you decide to later restore the media for use with DOS/ Windows. `fdisk -d=/pchcfmt -s` will show the type. If you need to restore the PCMCIA IDE card for use with DOS/Windows you must restore the ID type. If you have PCMCIA support at the DOS level you may be able to use FDISK. If not you can use Linux to change the ID type. We may add this feature to OS-9 fdisk in the future but be warned: once the device is changed to RBF if you do not have the tools then this disk will have to stay RBF.

## PCF

PC style file system is also supported. FAT32 is, however, not supported in this release. For more information see [IDE Descriptors](#).

You may select the PCF file system as the boot media.

For example, if the drive is Fat (not Fat32) you may place the bootfile image on the root. Make sure it is called `os9kboot`. Next create a CMDS and SYS directory at the root level. Copy whatever CMDS you need to the CMDS directory. Create a startup and or password file as needed. This method allows you to use the same partition as Windows95 or NT when you actually run OS-9.

## Prepare Windows95/NT based system for use with OS-9.

```
md C:\CMDS
```

```
md C:\SYS
```

```
copy MWOS\OS9000\80386\CMDS\* C:\CMDS
```

```
copy MWOS\OS9000\80386\PORTS\PCAT\BOOTS\INSTALL\SYS\MSHELL\startup C:\SYS
```

```
copy MWOS\OS9000\80386\PORTS\PCAT\BOOTS\INSTALL\SYS\MSHELL\password C:\SYS
```

```
cd C:\SYS
```

```
cudo -cdo startup
```

```
cudo -cdo password
```

Although RBF is the preferred file system for use with OS-9 the convenience of using FAT file systems should be taken into consideration when deciding how you want to setup your system.

## Special Note

In the following example the IDE device for /h0 and /dd is set for PCMCIA IDE using socket #0.

If the init dialog is set to /h0 the following is generated. In this case we also have SoftStax SPF enabled.

```
setenv SHELL mshell; alias /dd /pcmhc1;chd /h0 ; chx /h0/cmds;mbinstall ;
```

```
ipstart;inetd <>>>/nil&./h0/sys/startup &\n
```

If the init dialog is set to /dd the following is generated. In this case we also have SoftStax SPF enabled.

```
setenv SHELL shell; alias /dd /pcmhc1;chd /dd ; chx /dd/cmds;mbinstall ;
```

```
ipstart;inetd <>>>/nil&./dd/sys/startup &\n
```

In both cases above the script file on hc4 in sys/startup will be executed. When building systems this file must exist but does not have to contain any data. To create the needed directory and file, the following commands suffice:

```
$ mkdir /pcmhc1/SYS
```

```
$ touch /pcmhc1/startup
```

It is usually best to create the initial boot image to not use /h0. /dd should be set for RAM disk. This will allow downloading the TAR images. Next setup the final boot image and select /h0 if as initial device name.

## Descriptors

Refer to [IDE Descriptors](#) for information on descriptor naming conventions. The descriptors for RB1003 are located in

MWOS/OS9000/80386/PORTS/PCAT/CMD5/BOOTOBS/DESC/RB1003. Also the RB1003 driver is located in

MWOS/OS9000/80386/PORTS/PCAT/CMD5/BOOTOBS.

## ROM BOOTING

If changes to the IDE addresses of time-out values are employed then the ROM boot system may also require changes.

MWOS/OS9000/80386/PORTS/PCAT/ROM/cnfgdata.des

Find the following sections:

```
#define IDE_CIS_PARAMS "ide0=0x320,0 ide1=0x360,0"
```

```
#define PCMCIA_IDE_PRIMARY_PARAMS_PORT "port=0x320 timeout=30 altstat=0xe"
```

```
#define PCMCIA_IDE_SECONDARY_PARAMS_PORT "port=0x360 timeout=30 altstat=0xe"
```

To remove time-out for example we could change the above to:

```
#define PCMCIA_IDE_PRIMARY_PARAMS_PORT "port=0x320 timeout=0 altstat=0xe"
```

```
#define PCMCIA_IDE_SECONDARY_PARAMS_PORT "port=0x360 timeout=0 altstat=0xe"
```

Or we could make the time-out shorter. IDE specification indicates we should wait up to 30 seconds.

```
#define PCMCIA_IDE_PRIMARY_PARAMS_PORT "port=0x320 timeout=5 altstat=0xe"
```

```
#define PCMCIA_IDE_SECONDARY_PARAMS_PORT "port=0x360 timeout=2 altstat=0xe"
```

To explain the definition of IDE\_CIS\_PARAMS in detail: "ide0=0x320,5 ide1=0x360,0" indicates that IDE0 (socket 0) has a base address of 0x320 and uses IRQ 5, while IDE1 (socket 1) has a base address of 0x360 and uses no interrupt.

## Advanced Notes

Some embedded systems support Compact Flash, which looks like a standard IDE device. In these cases, we may decide that RBF is the file system of choice, since we can boot the embedded board with no other boot devices installed. How do we place the RBF file system on such small embedded systems? Compact Flash devices will work in PCMCIA systems with a carrier, so that we can use a standard PC with PCMCIA support to build up the PCMCIA disk. Once the disk is built, we can then remove the Compact Flash from the carrier and place it in the target system for use.

## IDE Descriptors

For Standard IDE devices the devices are referenced as shown in the following table.

### Standard IDE - RBF Descriptors

|         |                                             |
|---------|---------------------------------------------|
| /hcfmt  | IDE primary master - Entire disk            |
| /hc1fmt | IDE primary master - Primary partition #1   |
| /hc2fmt | IDE primary master - Primary partition #2   |
| /hc3fmt | IDE primary master - Primary partition #3   |
| /hc4fmt | IDE primary master - Primary partition #4   |
| /hdfmt  | IDE primary slave - Entire disk             |
| /hd1fmt | IDE primary slave - Primary partition #1    |
| /hd2fmt | IDE primary slave - Primary partition #2    |
| /hd3fmt | IDE primary slave - Primary partition #3    |
| /hd4fmt | IDE primary slave - Primary partition #4    |
| /hefmt  | IDE secondary master - Entire disk          |
| /he1fmt | IDE secondary master - Primary partition #1 |
| /he2fmt | IDE secondary master - Primary partition #2 |
| /he3fmt | IDE secondary master - Primary partition #3 |

|         |                                             |
|---------|---------------------------------------------|
| /he4fmt | IDE secondary master - Primary partition #4 |
| /hffmt  | IDE secondary slave - Entire disk           |
| /hf1fmt | IDE secondary slave - Primary partition #1  |
| /hf2fmt | IDE secondary slave - Primary partition #2  |
| /hf3fmt | IDE secondary slave - Primary partition #3  |
| /hf4fmt | IDE secondary slave - Primary partition #4  |

## Standard IDE - PCF Descriptors

|        |                                             |
|--------|---------------------------------------------|
| /mhc1  | IDE primary master - Primary partition #1   |
| /mhc2f | IDE primary master - Primary partition #2   |
| /mhc3  | IDE primary master - Primary partition #3   |
| /mhc4  | IDE primary master - Primary partition #4   |
|        |                                             |
| /mhd1  | IDE primary slave - Primary partition #1    |
| /mhd2  | IDE primary slave - Primary partition #2    |
| /mhd3  | IDE primary slave - Primary partition #3    |
| /mhd4  | IDE primary slave - Primary partition #4    |
|        |                                             |
| /mhe1  | IDE secondary master - Primary partition #1 |
| /mhe2  | IDE secondary master - Primary partition #2 |
| /mhe3  | IDE secondary master - Primary partition #3 |
| /mhe4  | IDE secondary master - Primary partition #4 |
|        |                                             |
| /mhf1  | IDE secondary slave - Primary partition #1  |
| /mhf2  | IDE secondary slave - Primary partition #2  |
| /mhf3  | IDE secondary slave - Primary partition #3  |
| /mhf4  | IDE secondary slave - Primary partition #4  |

## CDROM IDE Descriptors

/cd0            IDE secondary master

## PCMCIA IDE - RBF Descriptors

/pchcfmt      PCMCIA IDE Socket #0 - Entire disk

/pchc1fmt     PCMCIA IDE Socket #0 - Primary partition #1

/pchefmt      PCMCIA IDE Socket #1 - Entire disk

/pche1fmt     PCMCIA IDE Socket #1 - Primary partition #1

## PCMCIA IDE - PCF Descriptors

/pcmhc1       PCMCIA IDE Socket #0 - Primary partition #1

/pcmhe1       PCMCIA IDE Socket #1 - Primary partition #1



---

### Note

The descriptors for IDE are automatically included when using the Wizard. You may also access the descriptors in the MWOS directory structure at:

MWOS/OS9000/80386/PORTS/PCAT/CMDS/BOOTOBS/DESC/RB1003

---

## DiskOnChip

M-Systems' DiskOnChip™ is a new generation of single-chip flash disks. The DiskOnChip device contains built-in firmware which provides full hard disk emulation and allows the DiskOnChip to operate as a boot device.

When used under OS-9, the DiskOnChip is managed by a TrueFFS™, technology based device driver, attached to the standard OS-9 file system (RBF) or to a DOS compatible file system (PCF).

This section is intended for systems integrators designing with the DiskOnChip 2000, DiskOnChip Millennium or DiskOnChip DIMM and describes how to use the DiskOnChip as a bootable data storage device under the OS-9 Operating System. In the remainder of this application note the term DiskOnChip is used to describe the above mentioned DiskOnChip Family Products.

### Benefits

- Small footprint for embedded boards.
- Native RBF file system supported. Full boot support including IPL boot technology.

### Low-Level Boot Support

OS9000/80386/CMDS/BOOTOBSJS/ROM/doc

### High-Level Support

OS9000/80386/PORTS/PCAT/CMDS/BOOTOBSJS/rbdoc



## Descriptors Used by Wizard

OS9000/80386/PORTS/PCAT/CMDS/BOOTOBJS/DESC/RBDOC/dochcfmt  
OS9000/80386/PORTS/PCAT/CMDS/BOOTOBJS/DESC/RBDOC/dochc1  
OS9000/80386/PORTS/PCAT/CMDS/BOOTOBJS/DESC/RBDOC/dochc1fmt  
OS9000/80386/PORTS/PCAT/CMDS/BOOTOBJS/DESC/RBDOC/dochc1.h0

Additional descriptors are provided in the "OS9000/80386/PORTS/PCAT/CMDS/BOOTOBJS/DESC/RBDOC" directory. Please refer to [DiskOnChip Descriptors](#) for information on the use of these other descriptors.

## Building a DiskOnChip Boot Image

The DiskOnChip distribution is made up of two primary modules: *doc*, which is the OS-9 low-level booter module, and *rbdoc*, the OS-9 device driver for DiskOnChip. The following procedure uses the Configuration Wizard to configure a boot device using DiskOnChip. The DiskOnChip device appears as a disk drive to the high-level system and is accessed using the RBF descriptor */dochc1*.

1. On your host PC, select Microware Configuration Wizard from the Program Menu.
2. At the main screen of the wizard, select the radio button labeled Advanced Mode.
3. In the text box labeled Configuration Name (Required), type DiskOnChip, Click on the OK button. A new window appears with a menu bar. This is the Advanced Mode window of the Configuration Wizard. It is here that we must tell the wizard to include DiskOnChip support.
4. Navigate to Configure->Coreboot->Disk Configuration. A window appears with several tabs.
5. Select the tab labeled *IDE Configuration*.
6. Look for the area of the window labeled DiskOnChip and click on the checkmark boxes labeled *Add to menu* and *Auto Boot*. This will cause the booter to include the proper modules for allowing the system to boot from the DiskOnChip device.
7. Click on the OK button to dismiss the window.

8. Navigate to Configure->Bootfile->Disk Configuration. A window appears with several tabs.
9. Select the tab labeled *IDE Configuration*.
10. Look for the area of the window labeled DiskOnChip and click on the checkmark box labeled *Enable DOC*. This will cause the booter to include the proper modules for allowing us to format the DiskOnChip device.
11. Click on the tab labeled Init Options.
12. Click on the OK button to dismiss the window.
13. Navigate to Configure->Build Image and another window appears.
14. Click on the *Check* button, then the *Build* button to build the coreboot and bootfile.
15. Once this process is finished, click on the *MakeBoot* button and follow the prompts to create a bootable OS-9 floppy.

## Booting OS-9 and Formatting DiskOnChip

Once the boot floppy has been created, insert it into the target's floppy drive and boot OS-9. Once the shell prompt (\$) appears, format the DiskOnChip device:

```
$ chd /d0
$ iniz /dochcfmt
$ format /dochcfmt -nv -np -r -v
```

## Creating the OS-9 Partition

Start fdisk by typing:

```
$ fdisk -d=/dochcfmt
```

From fdisk, select option 6 to create the Master Boot Record. Respond with y <ENTER>, then <ESC> to return to the fdisk main menu

Use option 1 to create an OS-9000 partition. Press <ENTER> to accept the partition size, then selection option 1 to create an OS-9000/386 partition. Press <ESC> to return to the fdisk main menu.

At the main menu, select option 2 to make the newly created partition active. You must type the partition number (1 in this case) and press <ENTER>. Press <ESC> to return to the fdisk main menu.

Press <ESC> to exit fdisk. You will be prompted for confirmation to change the disk. Type y <ENTER>

## Formatting the Partition

Now the DiskOnChip contains an OS-9000 partition. Format this partition with the following command:

```
$ format /dochc1fmt -np -nv -r -v
```

## Installing Boot Files on DiskOnChip

Use the following commands on your OS-9 computer to install the boot image onto the DiskOnChip device.

```
$ bootgen -iplhdnoq -lfirstboot /dochc1fmt  
$ bootgen /dochc1fmt sysboot
```

## Booting OS-9 from the DiskOnChip

Booting OS-9 from the DiskOnChip device allows you to use the DiskOnChip as the only disk in the system, holding the operating system itself in addition to all other applications and files.

Please follow the steps described in the following paragraphs in order to use the DiskOnChip as the boot device.

## Updating the DiskOnChip Firmware

By default, the DiskOnChip firmware installs the DiskOnChip as an additional disk in the system. This default allows you to boot an Operating System from the DiskOnChip on a diskless machine. In case your machine is equipped with other hard disk(s) and you still want to boot from the DiskOnChip, you need to install the DiskOnChip as the first drive.

In order to install the DiskOnChip as the first drive, boot your target system into MS-DOS and perform the following command:

DUPDATE /WIN:{address} /S:DOC121.EXB /FIRST

With the {address} being the base address of the DiskOnChip, i.e. D000, D400, etc. “121” in the file DOC121.EXB represents the firmware version. The actual firmware version used might be a higher version, i.e. DOC122.EXB, etc.



---

## Note

The DUPDATE utility and firmware files are provided with the DiskOnChip ISA evaluation board available from M-Systems.

The default base address for the M-System’s evaluation board is D000h. Refer to the documentation included with your hardware for the base address and board jumper settings.

If you do not need to access additional hard disk(s) under OS-9, you may also disable them in the CMOS setup. In this case, the above DOS command is not necessary.

---

In some cases it is useful to prevent the DiskOnChip firmware from installing at boot time. You can achieve this by performing the following DOS command:

DUPDATE /WIN:{address} /S:DOC2.FFF

With the {address} being the base address of the DiskOnChip, i.e. D000h, D400h, etc.

## Booting OS-9 from DiskOnChip

Remove any floppy disks from the floppy drive and reset your target. The OS-9 IPL message should appear briefly, then the following screen:

```
OS-9000/x86 Bootstrap
Now trying to Override autobooters.
```

At this point, the floppy booter will be called and will fail because there is no floppy disk in the drive. At that point, the DiskOnChip booter will read the OS-9 bootfile from the DiskOnChip device. The system will then boot to a shell prompt.

Your system is now fully booting from DiskOnChip!

## DiskOnChip Descriptors

In the `MWOS\OS9000\80386\PORTS\PCAT\CMD5\BOOTOBS\DESC\RBD0C` directory there are numerous device descriptors for both RBF and PCF filesystems. Note that the table below omits descriptors with the filename extension `.h0` - these files are also present, and contain device descriptors with the canonical name `h0`, useful for systems whose main disk unit will be a DiskOnChip device.

### DiskOnChip - RBF Descriptors

|                         |                                            |
|-------------------------|--------------------------------------------|
| <code>/dochcfmt</code>  | DiskOnChip Device 0 - Entire disk          |
| <code>/dochc1fmt</code> | DiskOnChip Device 0 - Primary partition #1 |
| <code>/dochc2fmt</code> | DiskOnChip Device 0 - Primary partition #2 |
| <code>/dochc3fmt</code> | DiskOnChip Device 0 - Primary partition #3 |
| <code>/dochc4fmt</code> | DiskOnChip Device 0 - Primary partition #4 |
| <code>/dochdfmt</code>  | DiskOnChip Device 1 - Entire disk          |
| <code>/dochd1fmt</code> | DiskOnChip Device 1 - Primary partition #1 |
| <code>/dochd2fmt</code> | DiskOnChip Device 1 - Primary partition #2 |
| <code>/dochd3fmt</code> | DiskOnChip Device 1 - Primary partition #3 |
| <code>/dochd4fmt</code> | DiskOnChip Device 1 - Primary partition #4 |

/dochefmt     DiskOnChip Device 2 - Entire disk  
 /doche1fmt   DiskOnChip Device 2 - Primary partition #1  
 /doche2fmt   DiskOnChip Device 2 - Primary partition #2  
 /doche3fmt   DiskOnChip Device 2 - Primary partition #3  
 /doche4fmt   DiskOnChip Device 2 - Primary partition #4  
 /dochffmt     DiskOnChip Device 3 - Entire disk  
 /dochf1fmt   DiskOnChip Device 3 - Primary partition #1  
 /dochf2fmt   DiskOnChip Device 3 - Primary partition #2  
 /dochf3fmt   DiskOnChip Device 3 - Primary partition #3  
 /dochf4fmt   DiskOnChip Device 3 - Primary partition #4

## **DiskOnChip - PCF Descriptors**

/docmhcfmt   DiskOnChip Device 0 - Entire disk  
 /docmhc1fmt   DiskOnChip Device 0 - Primary partition #1  
 /docmhc2fmt   DiskOnChip Device 0 - Primary partition #2  
 /docmhc3fmt   DiskOnChip Device 0 - Primary partition #3  
 /docmhc4fmt   DiskOnChip Device 0 - Primary partition #4  
 /docmhdfmt   DiskOnChip Device 1 - Entire disk  
 /docmhd1fmt   DiskOnChip Device 1 - Primary partition #1  
 /docmhd2fmt   DiskOnChip Device 1 - Primary partition #2  
 /docmhd3fmt   DiskOnChip Device 1 - Primary partition #3  
 /docmhd4fmt   DiskOnChip Device 1 - Primary partition #4  
 /docmhcfmt   DiskOnChip Device 2 - Entire disk  
 /docmhe1fmt   DiskOnChip Device 2 - Primary partition #1  
 /docmhe2fmt   DiskOnChip Device 2 - Primary partition #2  
 /docmhe3fmt   DiskOnChip Device 2 - Primary partition #3  
 /docmhe4fmt   DiskOnChip Device 2 - Primary partition #4

/docmhffmt DiskOnChip Device 3 - Entire disk  
/docmh1fmt DiskOnChip Device 3 - Primary partition #1  
/docmh2fmt DiskOnChip Device 3 - Primary partition #2  
/docmh3fmt DiskOnChip Device 3 - Primary partition #3  
/docmh4fmt DiskOnChip Device 3 - Primary partition #4

## PCAT Style Floppy

Standard floppy support is provided using the RB765 driver. /d0 may be used to access RBF native file system. /md0 may be used to access PC style floppy devices.

## Floppy Descriptors

### Floppy - RBF Descriptors

/d0 Floppy drive A:

### Floppy - PCF Descriptors

/md0 Floppy drive A:



---

#### Note

When using the Wizard the descriptors for floppy devices are automatically included. You may also access the descriptors in the MWOS directory structure:

MWOS/OS9000/80386/PORTS/PCAT/CMDS/BOOTOBS/DESC/RB765

---

## **Symbios 810,810A,825,825A and 875 PCI SCSI controllers—Wide, Ultra and Ultra Wide**

### **Benefits**

- Wide support
- Ultra FAST20 support
- SCRIPTS RAM support ( able to run scripts from on-chip RAM )
- Large FIFO enabled
- Increased burst rates to 128 where supported
- Special PCI cache features enabled
- PCI IO Mode selectable (PCI I/O or PCI Memory )






---

**Note**

The SCRIPTS RAM support is currently only available on OS9000 X86 based systems. Requires non translation of PCI memory. To use SCRIPTS RAM support include the "-dSCRIPTS\_RAM" in the compile line when making the driver.

Instruction prefetch is not enabled by default. Maximum burst rate and large fifo's are enabled.

By default the Microware Symbios driver will use the PCI I/O model. To speed up transfers, especially on X86, platforms the memory module may be used. In the PCI memory mode no in/out instructions are used. For the X86 platform this removes the CPU related waits added by the use of "inc", "outc" etc. If you want to run the driver in PCI Memory mode the driver may be recompiled with the "-dPCI\_IO\_MAPPED" flag removed from the

MWOS/OS9000/80386/PORTS/PCAT/SCSI/SCSI8XX/makefile

---

```
IO_MAPPED = -dPCI_IO_MAPPED
```

To use memory model change to:

```
IO_MAPPED = # -dPCI_IO_MAPPED
```




---

**Note**

The default has changed to IO\_MAPPED for X86 due to problems on PCAT based motherboards.

---

Prior to this release the following Symbios devices were supported the following devices:

### Number of devices supported (2)

DEVICEWIDEULTRA1ULTRA2FIFO\_SIZEBURST

Symbios 53c810N/AN/AN/A6416

Symbios 53c825NoN/AN/A8816

This release adds the following:

### Number of devices supported (12)

DEVICEWIDEULTRA1ULTRA2FIFO\_SIZEBURST

Symbios 53c810N/AN/AN/A6416

Symbios 53c810APN/AN/AN/A6416 (1)

Symbios 53c815N/AN/AN/A6416 (1)

Symbios 53c820YesN/AN/A8816 (1)

Symbios 53c825YesN/AN/A8816

Symbios 53c825AYesN/AN/A536128

Symbios 53c875YesYESN/A536128

Diamond FirePort20YesN/AN/A536128 (825A)

Diamond FirePort40YesYESN/A536128 (875)

Symbios 53c860YesYESN/A536128 (1)

Symbios 53c885YesYESN/A536128 (1)

Symbios 53c895YesYESYES536128 (1,2)

Symbios 53c896YesYESYES536128 (1,2)

(1) Support is included but untested.

(2) Support for 895 and 896 is only available with out ULTRA support. The 160Mhz clock will be enabled on a future release.



### Note

The 895 and 896 have not been tested.

[Symbios 53C810]

[Symbios 53C810A]

Device supports burst op code fetch

Device supports instruction prefetch

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |  |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|
| Device supports Cache Line Size and Cache Commands<br>[Symbios 53C810ALV] * same as 810<br>Device supports burst op code fetch<br>Device supports instruction prefetch<br>Device supports Cache Line Size and Cache Commands<br>[Symbios 53C815]<br>Device supports burst op code fetch<br>[Symbios 53C825]<br>Device supports Wide SCSI data transfers<br>Device supports burst op code fetch<br>[Symbios 53C825A]<br>Device supports Wide SCSI data transfers<br>Device supports burst op code fetch<br>Device supports instruction prefetch<br>Device has Scripts RAM<br>Device supports Cache Line Size and Cache Commands<br>[Symbios 53C860]<br>Device supports burst op code fetch<br>Device supports instruction prefetch<br>Device supports Cache Line Size and Cache Commands<br>Device supports Fast-20 transfers<br>[Symbios 53C875]<br>Device supports Wide SCSI data transfers<br>Device supports burst op code fetch<br>Device supports instruction prefetch<br>Device has Scripts RAM<br>Device supports Cache Line Size and Cache Commands |  |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|

Device supports Fast-20 transfers

[Symbios 53C885]

Device supports Wide SCSI data transfers

Device supports burst op code fetch

Device supports instruction prefetch

Device has Scripts RAM

Device supports Cache Line Size and Cache Commands

Device supports Fast-20 transfers

Device supports Clock Doubler

[Symbios 53C895]

Device supports Wide SCSI data transfers

Device supports burst op code fetch

Device supports instruction prefetch

Device has Scripts RAM

Device supports Cache Line Size and Cache Commands

Device supports Fast-20 transfers ( Not supported yet )

Device supports Clock Doubler ( Not supported yet )

Device supports Fast-40 transfers ( Not supported yet )

[Symbios 53C896]

Device supports Wide SCSI data transfers

Device supports burst op code fetch

Device supports instruction prefetch

Device has Scripts RAM

Device supports Cache Line Size and Cache Commands

Device supports Fast-20 transfers ( Not supported yet )

Device supports Clock Doubler ( Not supported yet )

Device supports Fast-40 transfers ( Not supported yet )

Using Ultra Fast20 and Wide support.

## Controller Dependency

For FAST20 support the controller must support FAST20.

## Device Descriptors

To use a device with disconnect, wide, synchronous data transfer, and FAST20 Ultra the following should be added to the device descriptor entry in "systype.h". Be sure to re-make the descriptors.

```
#define SCSI_OPTS SCSI_ATN|SCSI_SYNC|SCSI_WIDE|SCSI_ULTRA
```

Optionally you may use EditMod to change the SCSI\_OPTS field. For SYNC and ATN the SCSI\_OPTS value is "5".

## Using Multiple SCSI Controllers

It is possible to use multiple SCSI controllers with the Symbios family of controllers.

The port address is used to specify the card to use.

PortAddress format.

[0xff] [device] [index] [SCSI\_ID]

device = device number. Use PCIV to discover index to match. This is system dependent and slot dependent.

Index = you may instead use index to specify the index of the card found. Zero indicates first card, one indicates second card, etc.

The same address information may be used from the OS-9 boot menu to access additional SCSI controllers, e.g.:

: hs port=0xff000100 id=3 ? Boot from second SCSI controller SCSI ID=3

## Creating Driver-Specific Versions

By default, the Symbios scsi8xx driver will look for any Symbios SCSI card based on table usage. You may however re-compile the driver to only look for the card desired.

MWOS/OS9000/80386/PORTS/PCAT/SCSI/SCSI8XX/makefile

```
PCI_DEV_ID    =    # -dSYMBIOS_DEVICE_ID=0xf
```

Remove the # and specify the ID required.

Driver name: scsi8xx

Rom driver name: ncr8xx

## Diamond FirePort20 and FirePort40—Wide, Ultra and Ultra Wide

### Benefits

- Wide support
- Ultra FAST20 support
- SCRIPTS RAM support ( able to run scripts from on chip RAM ) (1)
- Large FIFO enabled
- Increased burst rates to 128 where supported
- Special PCI cache features enabled (2)
- PCI IO Mode selectable (PCI I/O or PCI Memory ) (3)

## Additional Notes

1. The SCRIPTS RAM support is currently only available on OS-9, X86 based systems. Requires non translation of PCI memory. To use SCRIPTS RAM support include the "-dSCRIPTS\_RAM" in the compile line when making the driver.
2. Instruction prefetch is not enabled by default. Maximum burst rate and large fifo's are enabled.
3. By default the Microware Symbios driver will use the PCI I/O model. To speed up transfers especially on X86 platforms the memory module may be used. In the PCI memory mode no in/out instructions are used. For the X86 platform this removes the CPU related waits added by the use of "inc", "outc" etc... If you would like to run the driver in PCI Memory mode the driver may be recompiled with the "-dPCI\_IO\_MAPPED" flag removed.

MWOS/OS9000/80386/PORTS/PCAT/SCSI/SCSI8XX/makefile

IO\_MAPPED = -dPCI\_IO\_MAPPED

To use memory model change to:

IO\_MAPPED = # -dPCI\_IO\_MAPPED



---

### Note

We have changed the default to IO\_MAPPED for X86 due to problems on PCAT based motherboards.

---

Prior to this release the following Symbios devices were supported:

## Number of devices supported (2)

DEVICEWIDEULTRA1ULTRA2FIFO\_SIZEBURST

-----  
Symbios 53c810N/AN/AN/A6416  
Symbios 53c825NoN/AN/A8816

This release adds the following:

## Number of devices supported (12)

DEVICEWIDEULTRA1ULTRA2FIFO\_SIZEBURST

-----  
Symbios 53c810N/AN/AN/A6416  
Symbios 53c810APN/AN/AN/A6416 (1)  
Symbios 53c815N/AN/AN/A6416 (1)  
Symbios 53c820YesN/AN/A8816 (1)  
Symbios 53c825YesN/AN/A8816  
Symbios 53c825AYesN/AN/A536128  
  
Symbios 53c875YesYESN/A536128  
Diamond FirePort20YesN/AN/A536128 (825A)  
Diamond FirePort40YesYESN/A536128 (875)  
Symbios 53c860YesYESN/A536128 (1)  
Symbios 53c885YesYESN/A536128 (1)  
Symbios 53c895YesYESYES536128 (1,2)  
Symbios 53c896YesYESYES536128 (1,2)

1. Support is included but untested.
2. Support for 895 and 896 is only available with out ULTRA support. The 160Mhz clock will be enabled on a future release. Note the 895 and 896 have not been tested.

[Symbios 53C810]

[Symbios 53C810A]

Device supports burst op code fetch

Device supports instruction prefetch

Device supports Cache Line Size and Cache Commands

[Symbios 53C810ALV] \* same as 810

Device supports burst op code fetch

Device supports instruction prefetch

Device supports Cache Line Size and Cache Commands



[Symbios 53C815]

Device supports burst op code fetch

[Symbios 53C825]

Device supports Wide SCSI data transfers

Device supports burst op code fetch

[Symbios 53C825A]

Device supports Wide SCSI data transfers

Device supports burst op code fetch

Device supports instruction prefetch

Device has Scripts RAM

Device supports Cache Line Size and Cache Commands

[Symbios 53C860]

Device supports burst op code fetch

Device supports instruction prefetch

Device supports Cache Line Size and Cache Commands

Device supports Fast-20 transfers

[Symbios 53C875]

Device supports Wide SCSI data transfers

Device supports burst op code fetch

Device supports instruction prefetch

Device has Scripts RAM

Device supports Cache Line Size and Cache Commands

Device supports Fast-20 transfers

[Symbios 53C885]

Device supports Wide SCSI data transfers

Device supports burst op code fetch

Device supports instruction prefetch

Device has Scripts RAM

Device supports Cache Line Size and Cache Commands

Device supports Fast-20 transfers

Device supports Clock Doubler

[Symbios 53C895]

Device supports Wide SCSI data transfers

Device supports burst op code fetch

Device supports instruction prefetch

Device has Scripts RAM

Device supports Cache Line Size and Cache Commands

Device supports Fast-20 transfers ( Not supported yet )

Device supports Clock Doubler ( Not supported yet )

Device supports Fast-40 transfers ( Not supported yet )

[Symbios 53C896]

Device supports Wide SCSI data transfers

Device supports burst op code fetch

Device supports instruction prefetch

Device has Scripts RAM

Device supports Cache Line Size and Cache Commands

Device supports Fast-20 transfers ( Not supported yet )

Device supports Clock Doubler ( Not supported yet )

Device supports Fast-40 transfers ( Not supported yet )

Using Ultra Fast20 and Wide support.

## Controller Dependency

For FAST20 support the controller must support FAST20.

## Device Descriptors

To use a device with disconnect, wide, synchronous data transfer, and FAST20 Ultra the following should be added to the device descriptor entry in "systype.h". Be sure to re-make the descriptors.

```
#define SCSI_OPTS SCSI_ATN|SCSI_SYNC|SCSI_WIDE|SCSI_ULTRA
```

Optionally you may use EditMod to change the SCSI\_OPTS field. For SYNC and ATN the SCSI\_OPTS value is "5".

## Using Multiple SCSI Controllers

It is possible to use multiple SCSI controllers with the Symbios family of controllers.

The port address is used to specify the card to use.

PortAddress format.

[0xff] [device] [index] [SCSI\_ID]

device = device number. Use PCIV to discover index to match. This is system dependent and slot dependent.

Index = you may instead use index to specify the index of the card found. Zero indicates first card, one indicates second card, and so on.

The same address information may be used from the OS-9 boot menu to access additional SCSI controllers, e.g.:

: hs port=0xff000100 id=3 ? Boot from second SCSI controller SCSI ID=3

## Creating Driver-Specific Versions

By default, the Symbios scsi8xx driver will look for any Symbios SCSI card based on table usage. You may however re-compile the driver to only look for the card desired.

MWOS/OS9000/80386/PORTS/PCAT/SCSI/SCSI8XX/makefile

```
PCI_DEV_ID = # -dSYMBIOS_DEVICE_ID=0xf
```

Remove the # and specify the ID required.

Driver name: scsi8xx

Rom driver name: ncr8xx

## Adaptec 1540/1542 ISA

Support for Adaptec 1540 series is provided, this includes 1540, 1542 and 1542CP. The driver probes the DMA channel used, but the port address and interrupt are fixed. If the vector does not match the card, a Bad Mode error is returned. You may set up the descriptor to use vector zero, which forces the driver to use what the card reports.

```
#defineBASE_AHA15400x00000330  
#defineVECT_AHA15400x4b
```

Driver name: aha1540

Rom driver name: ll1540

## Adaptec 2940, 2940U and 2940UW

Support for Adaptec PCI series AHA2940, 2940U and 2940UW is provided. Only one SCSI controller of this type is allowed.

Driver name: aic7870

Rom driver name: ll7870

### SCSI HARD - RBF Descriptors

/hs<id>fmt id= SCSI ID (1-f) - Entire disk

/hs<id><part>fmtid= SCSI ID (1-f) part= partition

/hs<id><part> id= SCSI ID (1-f) part= partition

### SCSI HARD - PCF Descriptors

/mhs<id><part> id= SCSI ID (1-f) part= partition

### SCSI FLOPPY - RBF Descriptors

d<id>\_3.d0 id= SCSI ID (1-f) mapped as drive d0

### SCSI FLOPPY - PCF Descriptors

md<id>\_3.d0 id= SCSI ID (1-f) mapped as drive md0

### SCSI TAPE Descriptors

/mt<id> id= SCSI ID (1-f)

### SCSI CDROM Descriptors

/cd0 SCSI ID is set to 5



---

## Note

When using the Wizard, the descriptors for SCSI are automatically included or created as needed for the SCSI controller selected. You may also access the descriptors in the MWOS directory structure:

```
MWOS/OS9000/80386/PORTS/PCAT/CMDS/BOOTOBS/DESC/SCSI8XX  
MWOS/OS9000/80386/PORTS/PCAT/CMDS/BOOTOBS/DESC/AHA1540  
MWOS/OS9000/80386/PORTS/PCAT/CMDS/BOOTOBS/DESC/AIC7870
```

---

## System Devices

---

### Real Time Clock

Real-time clock (RTC) devices with battery backup enable the system clock to be set without operator intervention. The bootfile options dialog in the Configuration Wizard may be used to include one of two possible real-time clock drivers.

The local time driver assumes that the time stored in the RTC device is local time. This option maintains compatibility when another O.S. is installed on the same machine.

The GMT driver assumes that the time stored in the RTC is Greenwich Mean Time.

The driver communicates with the OS-9 kernel using GMT, with the System Time Zone field in the init. module converting between GMT and local time. Refer to the Configuration Wizard Init dialog for information on setting the system time zone.

# Additional Devices

---

## PPP and SLIP



### Note

PPP and SLIP is not directly supported by the Wizard.

---

Although PPP and SLIP are not directly supported by the Wizard, you can use the Wizard to configure and use both PPP and SLIP.

You may select any or all of Ethernet, PPP, or SLIP. When using PPP or SLIP, the SPF options must be enabled. You can do this from the **SPF/Options** tab by selecting either SLIP or PPP or both. When you do this, make sure SPF is checked when building the boot image. If Ethernet is not desired, select **None** for the Ethernet controller name.

## PPP Setup

Set up PPP by completing the following steps:

- 
- Step 1. Edit the `pcat.ini` file in the following directory:  
`MWOS\OS900\80386\PORTS\PCAT\BOOTS\INSTALL\INI`  
 Search for `ETHER_OPTION_1`.
  - Step 2. By default, the PPP setup will obtain the address from the server. If desired this may be changed.  
`ETHER_OPTION_1=ppp0 binding /ipcp0 iff_pointopoint`
  - Step 3. Make sure PPP is selected in the **SPF/Options** tab.
  - Step 4. Go into the Wizard and select **Enable SoftStax**. Build the boot.
-



## SLIP Setup

Set up SLIP by completing the following steps:

---

Step 1. Edit the `pcat.ini` in the following directory:

`MWOS\OS900\80386\PORTS\PCAT\BOOTS\INSTALL\INI`

Search for `ETHER_OPTION_0`.

Step 2. Setup SLIP as required.

`ETHER_OPTION_0=slip0 address 10.0.0.1 destaddr 10.0.0.2 binding /spsl0`

Step 3. Make sure SLIP is selected in the **SPF/Options** tab.

Step 4. Go into the Wizard and select **Enable SoftStax**. Build the boot.

---



---

# 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 \_\_\_\_\_



MICROWARE SOFTWARE