Getting Started with the Ethernet Router Application

Version 3.2

www.radisys.com

World Headquarters
5445 NE Dawson Creek Drive • Hillsboro, OR
97124 USA
Phone: 503-615-1100 • Fax: 503-615-1121
Toll-Free: 800-950-0044

International Headquarters
Gebouw Flevopoort • Televisieweg 1A
NL-1322 AC • Almere, The Netherlands
Phone: 31 36 5365595 • Fax: 31 36 5365620

RadiSys Microware Communications Software Division, Inc.
1500 N.W. 118th Street
Des Moines, Iowa 50325
515-223-8000

Revision A
December 2001

Copyright and publication information

This manual reflects version 3.2 of Enhanced OS-9 for IXP1200.
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

Table of Contents

Chapter 1: Ethernet Router Application 3

4 Overview
6 Target Setup
6 ENP-3511 Jumper Settings
7 ENP-2505 DIP Switch Settings
7 Setting up the Demonstration System
10 The Boot Manager
11 Boot Manager Commands
14 Configuring the Network Interfaces
14 Configuring the RIP Daemon
17 Configuring the OSPF Daemon
19 Configuring the Network Interfaces for Ethernet Ports
21 Running the Application
22 Explanation of the Software Components
22 IXP1200 Forwarding Table Manager
22 IP Infustion ZebOS
23 EMANATE®/Lite SNMP Modules

Chapter 1: Ethernet Router Application

The following document contains information on the RadiSys 4-port Ethernet Router Application. The following sections are included:

- **Overview**
- **Target Setup**
- **Configuring the Network Interfaces**
- **Running the Application**
- **Explanation of the Software Components**

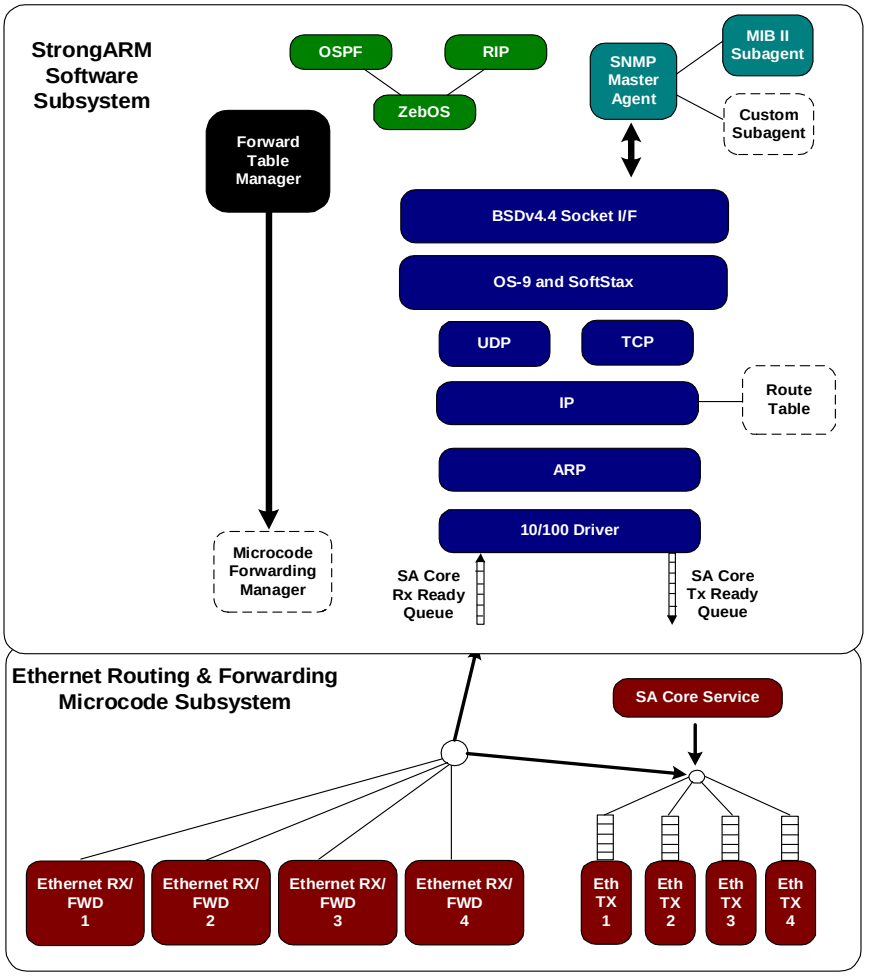
Overview

The RadiSys 4-port Ethernet Router Application demonstrates a forwarding and routing solution using Intel IXA building blocks to develop next-generation networking applications.

Figure 1-1 shows the components of the RadiSys 4-port Ethernet Router Application. The application contains the following software:

- Microware OS-9 real-time operating system with the SoftStax™ communications framework
- IXP1200 Forwarding Table Manager for OS-9
- IP Infusion Advanced Routing Suite for OS-9 (which includes IP Infusion's ZebOS, RIPv1v2, and OSPFv2)
- SNMP Research EMANATE®/Lite SNMP for OS-9

Figure 1-1 RadiSys 4-port Ethernet Router Application

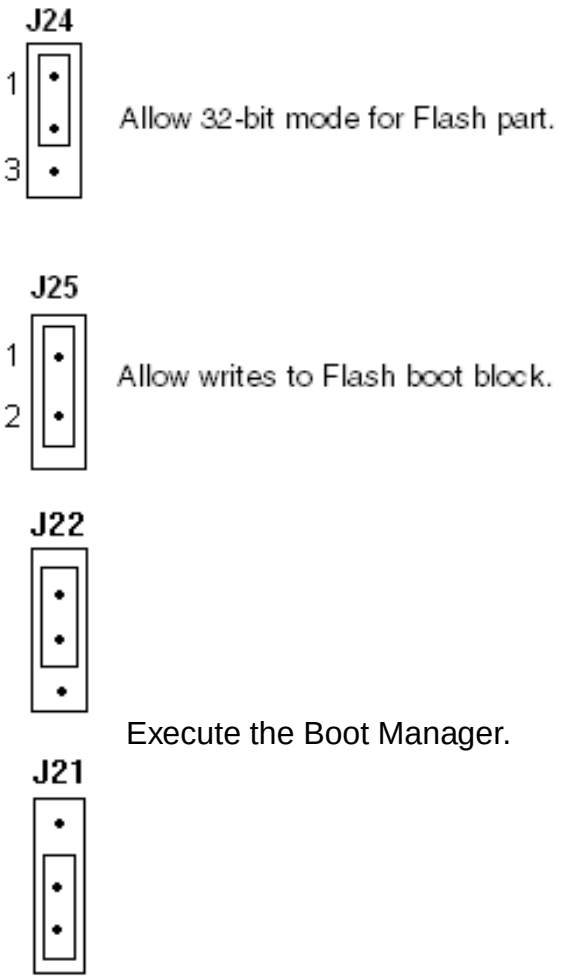


Target Setup

ENP-3511 Jumper Settings

If you are using the ENP-3511 board, you will need to modify the jumper settings according to those shown in **Figure 1-2**.

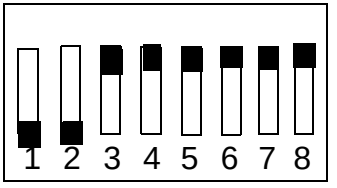
Figure 1-2 Jumper Settings for the ENP-3511



ENP-2505 DIP Switch Settings

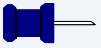
If you are using the ENP-2505 board, you will need to configure your settings according to **Figure 1-3** in order to use the Boot Manager. This configuration will allow the Configuration Register to boot the Boot Manager.

Figure 1-3 DIP Settings for ENP-2505 Board



Setting up the Demonstration System

To set up the demonstration system to run the Ethernet Router Application, complete the following steps.



Note

Be sure the board is powered off before you begin these steps.

Step 1. If you are connecting to a hub:

- Connect a straight-through Ethernet cable to the Intel Ethernet Pro port located on the front of the board. In addition, connect reverse Ethernet cables to each port used on the transition module.

If you are not connecting to a hub:

- Use crossover cables to make the connection.

In addition, you must connect the Intel Ethernet Pro port to a network running a DHCP server. This is necessary because the interface to the Intel Ethernet Pro port is configured via DHCP.

WARNING

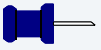
Do not connect any of the Ethernet ports (IXP1200 microcode ports) on the transition module to the same network that connects the Intel Ethernet Pro port.

Step 2. Connect the COM port to your PC, using an RS-232 serial cable.

For More Information

For more information on how to configure and establish a connection through the serial port, refer to the board guide for your respective processor, included with the Microware documentation on this CD.

Step 3. Boot the board.



Note

You may need to press the **Reset** button in order for it to boot properly.

Upon booting the board, you should see the following message on your terminal. If the message below does not appear, however, OS-9 has not been configured as the default operating system for your board. For information on configuring OS-9 as the default operating system, proceed to the **The Boot Manager** section.

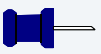
```
OS-9 Bootstrap for the ARM (Edition 65)

Now trying to Override autobooters.

Press the spacebar for a booter menu

Now trying to Copy embedded OS-9000 to RAM and boot.
Now searching memory ($00040000 - $007fffff) for an OS-9 Kernel...

An OS-9 kernel was found at $00040000
A valid OS-9 bootfile was found.
+3
tmode -w=l nopause
*
*OS-9 - Version 3.2
*Copywright 2001 by RadiSys Corporation
*
*The commands in this file are highly system dependent and should
*be modified by the user.
*
link mshel csl             ;*make “mshell” and “csl” stay in memory
iniz rl
copy /mm/zebos.conf -w=/r0/ZEBOS
copy /mm/ripd.conf -w=/r0/ZEBOS
copy /mm/ospfd.conf -w=/r0/ZEBOS
copy /mm/start_ethroute -w=/r0/SYS
copy /mm/v1.cf /r0/ETC/SRCONF/AGT/snmpd.cnf
copy /mm/motd /r0/SYS/motd
list sys/motd

************Welcome to Enhanced OS-9 for IXP1200 version 3.2******************
*                                                                             *
*       Thank you for selecting the RadiSys OS-9 Operating System         *
*                                                                             *
*                                                                             *
*             Please consule your documentation for information               *
*                          regarding the board.                               *
*                                                                             *
*******************************************************************************

ixp_engine &
+7
uenginit
ixp_engine &
+7
uenginit
```

```
Engine 0 is inactive
Engine 1 is inactive
Engine 2 is inactive
Engine 3 is inactive
Engine 4 is inactive
tmode -w=1 pause

$
```

The above script file copies the necessary IP Infusion configuration files, script files, and the SNMP configuration file from FLASH onto the RAM disk. This allows you to edit the files. It also starts two modules: `ixp_engine` and `uenginit`. `ixp_engine` installs the microcode engine services and enables interrupts. `uenginit` enables loading and initializing microcode directly on the network processor's microengines without using the Intel Developer's Workbench.

The Boot Manager

The Boot Manager (BootMgr) allows you to select which of the operating systems residing in Flash will be booted by the board. The BootMgr is accesible via a terminal connected to the board's serial port. Your current settings are saved into the Flash; thus, if no interaction is detected by the BootMgr, the current default operating system is automatically booted.



Note

The Boot Manager region in Flash is write protected and the reflashing tools described here will not update that region of the flash memory.

When the BootMgr starts, it prints out a banner on the serial port (at 38400 Baud):

```
Press space bar to stop auto-boot...
     10
```

This value (which is configurable) then counts down to zero. If it reaches zero before you press the space bar, the current default operating system will be booted. After auto-booting is stopped, the BootMgr prompt is displayed. You can then enter a command. To display all possible commands, type h, as shown below:

```
[BootMgr]: h

BootMgr commands:
p : Print boot parameters
c : Change boot parameters
b : Boot with current parameters
b <os> : Boot given os without changing parameters
h : Print this help message
```

Boot Manager Commands

The following section describes some of the available Boot Manager commands:

The print command p will display the current parameters:

```
[BootMgr]: p
BootMgr Version 1.0.2
CPU Revision 6901C123
OS list:
     0 Flash Utility
     1 Monitor
     2 Reserved
     3 VxWorks
     4 OS-9
Default OS: 4
Countdown value: 10
Disable initial display: 0

[BootMgr]:
```

The four parameters that may be varied include:

default operating system	This is the OS that will be booted if the autoboot countdown gets to zero. It's value is limited to the displayed list.
countdown value:	This is the number of seconds that the BootMgr will wait before auto-booting the default OS. The tradeoff is that with smaller values, there will be less delay in booting the desired OS, but the user will also have to be quicker in changing the boot parameters. This parameter is limited to values between 1 and 60 inclusive; i.e. a value of 0 is not allowed.
disable initial display:	If this value is set to 1, then the countdown display will not be printed. The BootMgr will still count down before auto booting, but no messages will be displayed unless the autoboot is stopped.

These parameters can be modified with the change command c. This command will print the current parameters and then prompt the user for new values. A blank line will leave the current value unchanged. This dialog looks similar to:

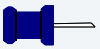
```
BootMgr Version 1.0.2
CPU Revision 6901C123
OS list:
     0 Flash Utility
     1 Monitor
     2 Reserved
     3 VxWorks
     4 OS-9
Disable initial display: 0
Enter blank line to leave value unchanged
Default OS:
Countdown value: 3
Disable initial display:

[BootMgr]:
```

In the example above, the countdown value was changed from 10 to 3, and the other parameters reamained unchanged.

The boot command b will boot the default operating system. Alternately, it can be given a numeric argument (such as `b 0`), which will boot the indicated operating system without changing the default. This could be used, for example, to boot the flash utility without modifying the default operating system.

Configuring the Network Interfaces

After booting the board, you need to configure the routing protocols and network interface. These include the RIP and OSPF daemons. Configuration information for the RIP and OSPF daemons are stored in the files `ripd.conf` and `ospfd.conf`, respectively.



Note

The example `ripd.conf` and `ospfd.conf` files assume that RIP and OSPF are running on two separate networks.

Configuring the RIP Daemon

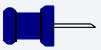
Below is an example of how the `ripd.conf` file is configured, by default, to use RIPv2. The file is located in `/r0/ZEBOS`, on the target machine. This configuration can be used if you are planning to run RIPv2 over one interface. (The file below shows RIPv2 running on the `enet0` interface.)

For More Information

To run RIPv1 or RIPv2 over more than one interface, refer to the ***IP Infusion RIPv1v2 for OS-9*** documentation, included with this CD.

```
! This is a simple configuration file for IP Infusion's rip daemon. On an OS-9
! target the default location for this file is /dd/zebos. The -f command line
! option may be used to specify a different location.
!
! Instead of using a text file, the <mwos>/SRC/ETC/zebos.mak makefile can be
! used to convert the text version to a data module. If this method is used,
! ripd should be started using:
!       ripd -f /mm/ripd.conf &
!
```

```
! See the IP Infusion documentation for additional configuration information.

hostname ripd
password radisys
enable password radisys

router rip

! Define which interface should run rip. This may be done either by using the
! interface name (enet0) or by specifying an IP address and netmask
! (192.168.3.0./2.4)
network enet0

log stdout
```



Note

The RIP daemon supports only one version of RIP; by default, RIPv2 is enabled. If you are running RIPv1, you need to add the line `version 1` to the `ripd.conf` file.

In order to modify the `ripd.conf` text file, the `edt` utility must be used. The `edt` utility is a line-oriented text editor. Below is the output you should receive when you type `edt -?` at the OS-9 prompt.

```
Syntax:   edt [<opts>] <path> [<opts>]
Function: line editor
Options:
  -b[=]<size> use larger buffer, <size> in k bytes (default file size + 2k)
Commands:
<cr>         : move line pointer down one line
<num>        : mo: move line pointer down <num> lines(default: 1)
-[<num>]     : move line pointer up <num> lines(default: 1)
c            : search/replace from current line
d[<num>]     : delete lines(default: 1)
l[<num>]     : list <num> lines(default: 1)
s            : search from current line
<space>      : insert line
<esc> or q   : write file and return to shell
```

For More Information

For more information on the `edt` utility, refer to Chapter 1 of the ***Utilities Reference*** manual in the Microware documentation set, included with this CD.

To modify the RIP daemon, complete the following steps:

Step 1. At the OS-9 prompt, type `edt ripd.conf`. You should see the following output:

```
*0001    !
  E:
```

Step 2. At the `E:` prompt, type `23`. This moves the line pointer to line 23. You should now see the following output:

```
*0023
        network enet0
  E:
```

Step 3. At the `E:` prompt, type `c/enet0/enet1`. This searches the current line (23) for the string enet0 and replaces it with enet1 (the name of the interface in this example). You should now see the following output:

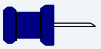
```
*0023
        network enet1
  E:
```

If you are running RIPv1, proceed to step four. If not, proceed directly to step six.

Step 4. At the `E:` prompt, type `<enter>`. This moves the line pointer to line 24. You should now see the following output:

```
*0024
 E:
```

Step 5. At the `E:` prompt type `<space> version 1`.



Note

Be sure to press <space> before entering this command. If you do not, the line will not be written to the file.

You should now see the following output:

```
*0025
E:
```

Step 6. At the `E:` prompt, type `<esc>`. This will write your changes to the file and return you to the OS-9 prompt.

Configuring the OSPF Daemon

Configuration information for the OSPF daemon is stored in the file `ospfd.conf`. Modify this file to provide the appropriate IP address for your network.

For More Information

For information on running OSPFv2 on more than one interface or information regarding the OSPF daemon, refer to the ***IP Infusion OSPFv2 for OS-9*** documentation, included with this CD.

Below is the default configuration for `ospfd.conf`:

```
!
!  This is a simple configuration file for IP Infusion's ospf daemon. On
!  an OS-9 target the default location for this file is /dd/zebos. The -f
!  command line option may be used to specify a different location.
!
!  Instead of using a text file, the <mwos>/SRC/ETC/zebos.mak makefile
!  can be used to convert the text version to a data module. If this
!  method is used, ospfd should be started using:
!       ospfd -f /mm/ospfd.conf &
!
!  See the IP Infusion documentation for additional configuration
!  information

hostname ospfd
password radisys
enable password radisys

router ospf

! This line must be replaced with the IP address and mask length of
! the subnet(s) that should run ospf.
network 192.168.3.0/24 area 0

log stdou
```

To modify the `ospfd.conf` file, complete the following steps:

Step 1. At the OS-9 prompt, type `edt ospfd.conf`. You should see the following output:

```
*0001    !
  E:
```

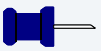
Step 2. At the `E:` prompt, type `22`. This moves the line pointer to line 22. You should see the following output:

```
*0022
        network 192.168.3.0/24 area 0
  E:
```

Step 3. At the `E:` prompt, type `d 22`. This deletes line 22. You should now see the following output:

```
*0022
E:
```

Step 4. At the `E:` prompt, type `<space> network <IP address>/<netmask> area <num>`.



Note

Be sure to press <space> before entering this code. If you do not, the line will not be written to the file.

Below is an example of what your line might look like:

```
E: network 192.168.3.202/24 area 0
```

You should now see the following output:

```
*0023
   E:
```

Step 5. At the `E:` prompt, type `<esc>`. This writes your changes to the file and return you to the OS-9 prompt.

Configuring the Network Interfaces for Ethernet Ports

It is necessary to configure the network interfaces for the Ethernet ports (IXP1200 microcode ports) used on the transition module. This can be done with the `ifconfig` utility. The `ifconfig` utility configures the network interface as such:

```
ifconfig <name of interface> <IP address> binding <device>
```

Note

To configure the network interfaces with `ifconfig`, use the name of the interface you specified in Step three under the section **Configuring the RIP Daemon**. This example uses `enet1` for the network running RIPv2 and `enet2` for the network running OSPFv2.

To configure the network interfaces with `ifconfig`, complete the following steps:

Step 1. Match the `<device>` to the Ethernet port (IXP1200 microcode port) on the transition module. For example, if port 0 is connected to the network running RIPv2, put the string `/spixp0/enet` in for `<device>`. If you are using port 1, be sure you put `/spixp1/enet` in for <device>. The number in the Ethernet driver descriptor corresponds to the Ethernet port number.

Step 2. To configure the interface to a network that runs RIPv2 connected to Ethernet port 0 on the transition module, type the following command at the OS-9 prompt:

```
ifconfig enet1 192.168.3.201 binding /spixp0/enet
```

To configure the interface to a network that runs OSPFv2 connected to Ethernet port 1 on the transition module, use the same IP address you used in Step four of the section, **Configuring the OSPF Daemon**.

If OSPFv2 is running on the same network as RIPv1 or RIPv2, you do not need to configure any other interfaces. However, be sure to use the same IP address you specified in Step four under the section **Configuring the OSPF Daemon** (in the example case of `enet1`). Otherwise, at the OS-9 prompt you should type the following line:

```
ifconfig enet2 192.168.3.202 binding /spixp1/enet
```

At this time, you may configure other interfaces.

Running the Application

Start the Ethernet Router Application by running the script file `start_ethroute`. To run this file, type the following code at the OS-9 prompt:

```
/r0/sys/start_ethroute
```

You should see the following output (or something similar):

```
ftm &
+8
zebos -d
ripd -d
ospfd -d
snmpd &
+13
$ SNMP Research SNMP Agent Resident Module Version
15.3.1.7
Copyright 1989, 1990, 1991, 1992, 1993, 1994, 1995,
1996, 1997, 1998, 1999, 2000, 2001 SNMP Research,
Inc.
```

At this point, you should be able to see and verify the RIP and OSPF routes by performing a telnet into the appropriate protocol module. You should also be able to see these routes using the `netstat` utility. To see which routes correspond with the interfaces, type `netstat –rn` at the OS-9 prompt.

For More Information

For more information on other options or networking utilities, refer to the ***Using LAN Communications Pak*** manual in the Microware documentation set included with this CD.

Explanation of the Software Components

Below is a brief summary of what each module does in the `start_ethroute` script file.

IXP1200 Forwarding Table Manager

The IXP1200 Forwarding Table Manager interfaces with the various packet processing pieces and extracts all of the information that is required in order to forward packets. It stores this information in lookup tables optimized for IP packet forwarding. These tables can be searched for using a low level driver. Forwarding decisions are made without the overhead of network stack processing.

IP Infustion ZebOS

The ZebOS daemon module, `zebosd`, provides kernel routing table management and distribution between the different routing protocols. The demo version of the `zebosd` only runs for eight hours. At the end of the eight hours, you must reboot your machine in order to get the IP Infusion routing protocols to work again. Below is the message you will see when the module times out:

```
Time Out! Please reset the board for continuation.
```

```
The routing subsystem, including ZebOS, RIP & OSPF is
a time-bombed product. Please contact RadiSys'
Microware Software at 888-642-7609 for a fully
licensed product.
```

The `ripd` module runs in daemon mode and manages RIPv1 & RIPv2.

The `ospfd` module runs in daemon mode and manages OSPFv2.

EMANATE®/Lite SNMP Modules

The snmpd module is an EMANATE®/Lite SNMP agent that runs in daemon mode. It contains an SNMP engine, command responder application, and notification originator application. It supports objects from MIB-II.

For More Information

Refer to the ***IP Infusion RIPv1v2 for OS-9*** & ***IP Infusion OSPFv2 for OS-9*** documentation for more information.