



# Using Soft-ATM for OS-9

## Version 1.1

[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 B  
March 2000

## Copyright and publication information

This manual reflects version 1.1 of Soft-ATM for OS-9.

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.

---

March 2000  
Copyright ©2000 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: Introduction to Soft-ATM™ for OS-9** **5**

---

- 6     ATM Signalling Overview
- 6         IEs and Defining ATM Connections
- 7         ATM Signalling Protocol Layers
- 8     ILMI Overview
- 8         ILMI Communication
- 9         MIB Information
- 10    Contents of Soft-ATM™ for OS-9
- 11         File Location
- 12         Soft-ATM™ for OS-9 Documentation
- 13         For More Information

## **Chapter 2: Installing and Running Soft-ATM™ for OS-9** **15**

---

- 16    Installing Soft-ATM™ for OS-9
- 17    Hardware and Software Requirements
- 17         Hardware Requirements
- 19         Software Requirements
- 20         Demonstration Testing Environment
- 21    Building the Boot ROM
- 22         Stage 1: Modifying the Parameter List
- 24         Stage 2: Configuring for IP Over ATM
- 24         Stage 3: Selecting Files for the Boot ROM
- 26         Stage 4: Executing the Build Command
- 27    Connecting the Host and Target
- 29    Loading the Boot ROM on the 8XXFADS
- 30    Loading the Boot ROM on the SBC8260

<b>Chapter 3: Protocol Driver spatmsig and its Descriptors</b>	<b>33</b>
34 The External Tasking Model	
35 User Profiles	
36 Protocol Data Units	
37 Device Descriptors	
<b>Chapter 4: Building the Libraries, Driver and Descriptors</b>	<b>39</b>
40 Building the Libraries	
42 Building the Driver spatmsig and Descriptors	
<b>Product Discrepancy Report</b>	<b>43</b>

---

# Chapter 1: Introduction to Soft-ATM™

## for OS-9

---

Soft-ATM™ for OS-9 is an integrated ATM signalling stack that provides the switched virtual circuits (SVC) necessary for on-demand ATM services. Soft-ATM™ for OS-9 is designed for developing both user-side and network-side ATM products.

This chapter includes the following sections:

- **ATM Signalling Overview**
- **ILMI Overview**
- **Contents of Soft-ATM™ for OS-9**
- **For More Information**



---

### Note

Soft-ATM™ for OS-9 is an extension of the ATM Base Pak. It is recommended that you become familiar with the ATM Base Pak before implementing Soft-ATM™ for OS-9.

---



---

### For More Information

For definitions of ATM terms and acronyms, see the glossary in the ***Using ATM Base Pak*** manual.

---



MICROWARE SOFTWARE

# ATM Signalling Overview

---

ATM Signalling implements procedures to establish Switched Virtual Circuits (SVC). SVCs enable you to dynamically establish, maintain, and clear ATM connections at the ATM User-Network Interface (UNI).

The signalling procedures are defined in terms of messages used exclusively for ATM connections and Information Elements (IEs) used to define the ATM connections. They apply to the interface between terminal (endpoint) equipment and a public or private network consisting of at least one ATM switch.

## IEs and Defining ATM Connections

Some of the characteristics of an ATM connection that IEs define are:

- The ATM address of the caller (node requesting a connection).
- The ATM address of the called node.
- The Quality of Service (QoS) for the connection, for example CBR, VBR, ABR.
- AAL parameters, for example AAL-5 or AAL-1 for CBR
- ATM traffic parameters such as forward and backward peak cell rates, sustainable cell rates, and maximum burst size.
- The connection identifier (the VPI/VCI of the SVC).

## ATM Signalling Protocol Layers

ATM signalling as defined by the ATM Forum is based upon the Q.2931 standard of the ITU. The ATM protocol layers required for ATM signalling are given below in order from highest to lowest:

- ATM UNI Signalling, ITU specification Q.2931.
- Service Specific Coordination Function (SSCF), Q.2130.
- Service Specific Connection Oriented Protocol (SSCOP), Q.2110.
- ATM Adaptation Layer Type 5 (AAL-5).
  - CPCS Sublayer
  - SAR Sublayer
- ATM Layer (generally on the ATM hardware device).



---

### Note

In this document, and in general, ATM signalling protocol layers are referred to by their documentation names rather than the actual protocol name. For example, Q.2110 is used instead of SSCOP.

---

# ILMI Overview

---

The Soft-ATM™ Integrated Layer Management Interface (ILMI) subsystem manages the User side and the Network side of the User-to-Network ATM interface. This allows access to management objects on the peer ATM networking equipment. ILMI also provides configuration and exchange of interface information between ATM devices, including the following:

- Virtual Path Connections (VPC).
- Virtual Channel Connections VCC).
- Registered ATM network prefixes.
- Registered ATM addresses.
- Registered services and capabilities available at ATM Interfaces.

ILMI uses the Simple Network Management Protocol (SNMP) and an ATM interface Management Information Base (MIB) to provide these services. The MIB stores and manages the interface information.

## ILMI Communication

Each MIB is associated with an ATM interface and has an associated interface manager. The interface manager accesses—via ILMI—only the MIB associated with the interface manager on a physically adjacent ATM Interface. Each interface manager contains an agent application and a management application.

ILMI communication takes place between adjacent interface managers over the well known VPI/VCI 0, 16.



## MIB Information

The following categories of information are available in the ATM Interface MIB

- The 7 “System Group” objects of RFC 1213.
- Physical Layer.
- ATM Layer - size of VPI, VCI address fields, count of permanent Virtual Path Connections (VPC), Virtual Channel Connections (VCC).
- Virtual Path Connection to adjacent interface manager.
- Virtual Channel Connection to adjacent interface manager.
- Address Registration Information.
- Service Registry.

VPI 0, VCI 16 is used for all SNMP/AAL-5 messages used by ILMI.



---

### Note

In the version 3.0 and 3.1 specifications, ILMI was an acronym for Interim Local Management Interface. In version 4.0, the acronym changed to Integrated Layer Management Interface.

---

## Contents of Soft-ATM™ for OS-9

---

Soft-ATM™ for OS-9 provides signalling for on-demand connection establishment over ATM. It was ported from the Harris & Jeffries, Inc. *Soft-ATM™ Signalling Software Subsystem* and the *SoftATM™ ILMI Subsystem*, which include the following:

- User-side of the UNI
- Network side of the UNI
- Support for Interim Inter-switch Signalling Protocol (IISP) and Private Network-to-Network Interface (PNNI) v1.0
- ILMI



---

### Note

The Harris & Jeffries, Inc. PNNI product is not included in Soft-ATM™ for OS-9.

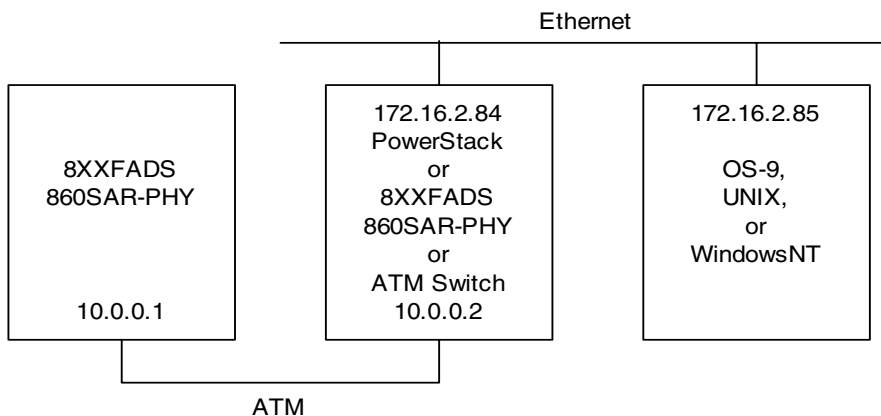
---

The Soft-ATM™ for OS-9 product consists of the following elements:

- Protocol driver
- Device descriptors
- Source code for building and modifying the driver
- Descriptors
- Documentation

Device descriptors are provided for terminal (endpoint) equipment and equipment emulating an ATM switch. The descriptors for emulating an ATM switch are provided so that Soft-ATM™ for OS-9 can be exercised in a testbed similar to **Figure 1-1**.

**Figure 1-1 Testbed for Emulating Switch**



## File Location

The driver, descriptor, and \*.stb files are listed below. The \*.stb files are included for use with Hawk and rombug. Drivers in the MON directories write debugging information to debug data modules, spatmsig\_dbg for MON/spatmsig (Soft-ATM™ for OS-9) and dbg\_llcsnap for MON/sp1577 and MON/sp860sar (ATM Base Pak).

- The ATM Signalling protocol driver consisting of layers Q.2931, Q.2130, Q.2110 and CPCS and ILMI in files:

```
/mwos/OS9000/PPC/CMDS/BOOTOBSJS/SPF/spatmsig
```

```
/mwos/OS9000/PPC/CMDS/BOOTOBSJS/SPF/STB/spatmsig.stb
```

```
/mwos/OS9000/PPC/CMDS/BOOTOBSJS/SPF/MON/spatmsig
```

```
/mwos/OS9000/PPC/CMDS/BOOTOBSJS/SPF/MON/STB/spatmsig.stb
```

- The device descriptor for terminal (endpoint) equipment is:  
`/mwos/OS9000/PPC/CMDS/BOOTOBS/SPF/hju0`
- The device descriptor for equipment emulating an ATM switch is:  
`/mwos/OS9000/PPC/CMDS/BOOTOBS/SPF/hjn0`
- The IPOA device descriptor for stacking device driver sp1577 over spatmsg, terminal (endpoint) equipment side is:  
`/mwos/OS9000/PPC/CMDS/BOOTOBS/SPF/llcsnap_hju`
- The IPOA device descriptor for stacking the device driver sp1577 over spatmsg, switch emulation side, is:  
`/mwos/OS9000/PPC/CMDS/BOOTOBS/SPF/llcsnap_hjn`

All source files for spatmsg and its descriptors are located in the following directory:

`/mwos/SRC/DPIO/SPF/DRVR/SPHJ`

## Soft-ATM™ for OS-9 Documentation

The following documentation is included with Soft-ATM™ for OS-9:

- ***Using Soft-ATM™ for OS-9***
- ***Soft-ATM™ Signalling Software Subsystem***  
(Harris & Jeffries, Inc.)
- ***Soft-ATM™ ILMI Subsystem*** (Harris & Jeffries, Inc.)

## For More Information

---

For more information about ATM signalling and Soft-ATM™ for OS-9, see the following references:

- **Soft-ATM™ Signalling Software Subsystem** - release 1.60 (Harris & Jeffries, Inc.), which contains an ATM signalling bibliography.
- **Soft-ATM™ ILMI Subsystem** - release 1.20 (Harris & Jeffries, Inc.).
- For documents approved by the ATM Forum use the ATM Forum web site at [www.atmforum.com](http://www.atmforum.com). The current approved document list can be found at:

<http://www.atmforum.com/atmforum/specs/approved.html>

- **ATM User-Network Interface (UNI) Signalling Specification Version 4.0 af-sig-0061.000**. July 1996. ATM Forum.
- **Anchorage Accord** (ISBN 0-07-856654-1). Available from the ATM Forum ftp server and from McGraw-Hill on CD ROM ([www.books.mcgraw-hill.com](http://www.books.mcgraw-hill.com)).

This book contains 60 ATM specifications.

- **Integrated Local Management Interface (ILMI) Specification Version 4.0 af-ilmi-0065.00**. July 1996. ATM Forum.
- **MPC8BUG Software Users Manual for the MPC821ADS**. This document is available on the Motorola web site.



---

# Chapter 2: Installing and Running Soft-ATM™ for OS-9

---

This chapter describes installing and running Soft-ATM™ for OS-9 and includes the following sections:

- **Installing Soft-ATM™ for OS-9**
- **Hardware and Software Requirements**
- **Building the Boot ROM**
- **Connecting the Host and Target**
- **Loading the Boot ROM on the 8XXFADS**
- **Loading the Boot ROM on the SBC8260**



MICROWARE SOFTWARE

## Installing Soft-ATM™ for OS-9

---

The steps for installing Soft-ATM™ for OS-9 on your host development system are described in ***Getting Started with Microware Products***, which is part of your product package.

You should also read either ***Enhanced OS-9 for the EST SBC8260 Board Guide*** or ***Enhanced OS-9 for the 8XXFADS Board Guide***.

Once the software is installed, you can start building your ATM signalling system described in this chapter.



---

### Note

You must install an Enhanced OS-9 software package as well as the ATM Base Pak before installing Soft-ATM™ for OS-9.

---



# Hardware and Software Requirements

---

## Hardware Requirements

### Host Development System Hardware Requirements

Your host PC must meet the following minimum requirements:

- 32MB of free disk space (an additional 235MB of free disk space is required to run PersonalJava for OS-9)
- 16MB of RAM (64MB recommended)
- ADI - PC board
- For the 8XXFADS target, the Motorola MPC8BUG ISA bus ADI card and 37 pin cable.
- For the SBCATMF target, the Embedded Support Tools Corp. (EST) visionICE incircuit emulator.

### Target System Hardware Requirements

One of the following board sets is required:

Set for Motorola MPC860SAR processor:

- Motorola MPC8XXFADS board.
- Motorola 860SR FADSDB daughter board.
- Motorola 860SAR-PHY board.

Set for Motorola MPC8260 processor:

- EST SBC8260 board.
- EST SBCATMF daughter board with the MPC8260.

The drivers and descriptors were developed and tested on the following hardware:

- SBC8260/SBCATMF board with Motorola MPC8260 processor

- MPC8XXFADS/860SR FADSDB/860SAR-PHY board.
- Motorola PowerStacks with Zeitnet ATM interface cards.

## Software Requirements

### Host Development System Software Requirements

Your host development system must have the following applications:

- Windows 95, Windows 98 or Windows NT 4.0
- A terminal emulation program (such as `Hyperterminal` that comes with Microsoft Windows 95, Windows 98, and Windows NT 4.0).
- Enhanced OS-9 for PowerPC
- ATM Base Pak
- Soft-ATM™ for OS-9
- For the 8XXFADS target, the Motorola MPC8bug or MPC8bug95 software.
- For the SBCATMF target, the EST visionICE Software Tools.

### Target System Software Requirements

All software for using Soft-ATM™ for OS-9 on your target system is included with the Enhanced OS-9 for PowerPC CD.

## Demonstration Testing Environment

A simple demonstration testing environment can consist of the following elements:

- MPC8XXFADS/860SR FADSDB/860SAR-PHY board or SBC8260/SBCATMF board with device descriptors hju0 and llcsnap\_hju installed (where “u” represents user side).
- PowerStack, emulating an ATM switch, with descriptors hjn0 and llcsnapn\_hjn installed (where “n” represents network side).
- Ethernet Local Area Network (LAN)

In this environment, the 8XXFADS and PowerStack are linked only by their ATM interfaces. The PowerStack is connected to an Ethernet LAN. This configuration is described in **Figure 1-1**.

When you are able to establish telnet sessions between the 8XXFADS and the IP nodes on the LAN, IPOA using ATM signalling is functioning.

## Building the Boot ROM

The boot ROM for your reference board can be built using the Configuration Wizard. The Configuration Wizard generates a S-record file, called `rom.S`, for downloading the boot ROM to the reference board. For the SBC8260 target the `rom.S` file must be converted to a `rom.bin` file using the EST convert program.



## Note

You must run shell commands in order to start IPOA. However, by modifying the `Parameter List` in the Configuration Wizard, you can avoid entering the commands at the shell prompt each time the reference board is booted. Once the list is modified, these commands are run at boot time.



## Note

The example below describes building a boot ROM for a Motorola PowerPC 8XXFADS reference board and the SBCATMF reference board.

To modify the `Parameter List` in the Configuration Wizard, you must know the following characteristics about your system:

- [illegible]

- The ATM address for the adjacent node, 3.3 in the example below.

The ATM addresses contain 20 bytes (numbers). When connecting the target to an ATM switch, you must know the ATM format of the ATM addresses on the network to be joined. You must also correctly initialize the following fields of the ATM address:

- AFI - Authority and Format Identifier (1 byte)
- IDI - Initial Domain Part, AFI and one of DCC, IDC, E.164
- HO-DSP - High Order Domain Specific Part
- ESI - End System Identifier
- SEL - Selector




---

## For More Information

See ***ATM User-Network Interface (UNI) Signalling Specification Version 4.0***, section 3.0, for addressing specifications. This is available from the ATM Forum.

---

Building the boot ROM performed in following three basic stages:

- **Stage 1: Modifying the Parameter List**
- **Stage 2: Configuring for IP Over ATM**
- **Stage 3: Selecting Files for the Boot ROM**
- **Stage 4: Executing the Build Command**

## Stage 1: Modifying the Parameter List

---

Step 1. Click the **Start** button on the Windows desktop.

- Step 2. Select **Programs --> Enhanced OS-9 for PowerPC --> Configuration Wizard**. The Configuration Wizard opening screen will be displayed.



---

### Note

If this is the first OS-9 boot ROM built for your reference board, or this is the first time you have used the Configuration Wizard, it is highly recommended at this point that you read the **Enhanced OS-9 Board Guide** for your particular reference board and complete the Enhanced OS-9 Tutorial.

---

- Step 3. Configure the opening the Configuration Wizard screen.
1. Select the appropriate reference board in the Port Selection menu **860SARFADS** or **SBC8260**.
  2. Select the proper MWOS location and port.
  3. Name your configuration.
  4. Select Advanced Mode.
- Step 4. Click **OK**.
- At this point, the Configuration Wizard main configuration window will be displayed.
- Step 5. From the Configuration Wizard main configuration window, select **Configure -> Bootfile -> Disk Configuration**.
- Step 6. From the Disk Configuration window, select the **Init Options** tab.
- Step 7. Under the Initial Device Name section, click the **User button**. This enables the Parameter List option.
- Step 8. In the Parameter List concatenate, in order, the strings below:
- ```
setenv SHELL shell;  
mbinstall -m1024k;  
ipstart;  
ifconfig atm0 10.0.0.1 binding /llcsnap_hju;
```

[illegible]

- Step 9. Click **OK**. The main configuration window is displayed.

## Stage 2: Configuring for IP Over ATM

When using the Configuration Wizard to build a boot ROM containing modules from the ATM Base Pak for a testbed exercising IPOA, you may wish to build a boot ROM that does not enable the Ethernet device on the target, relying instead on IP over ATM.

- Step 1. In **Configure -> Bootfile -> Network Configuration** select the **Interface** tab. In window **Disable/Enable Interface** disable Ethernet.
- Step 2. In **Configure -> Coreboot -> Main Configuration** select the **Debugger** tab. In the **Remote Debug Connection** window do not select Ethernet.

## Stage 3: Selecting Files for the Boot ROM

- Step 1. From the Configuration Wizard main configuration window, select **Configure -> Bootfile -> Network Configuration -> SoftStax Options**.
- Step 2. Configure the SoftStax Options window as follows:  
Select: **ifconfig, ndbmod, route, routed, telnet**.  
If you are using Soft-ATM™:



Select: **Soft-ATM** if the 8XXFADS is to be an ATM end node.

Select: **Soft-ATM Switch** if the 8XXFADS is to emulate an ATM switch.

---

## Stage 4: Executing the Build Command

- 
- Step 1. Build rom.S. Select **Configure -> Build Image**. The Master Builder window will be displayed.
- Step 2. From the Master Builder window, select **Coreboot + Bootfile** in the Build Type/Options section.
- Step 3. Make the following selections in the Include section:
- **ROM Utility Set**
  - **Disk Support**
  - **Disk Utilities [fdisk, format....]**
  - **SoftStax [SPF] Support Modules**
  - **User Modules**
- Step 4. Click Build. rom.S is saved to the following directory:

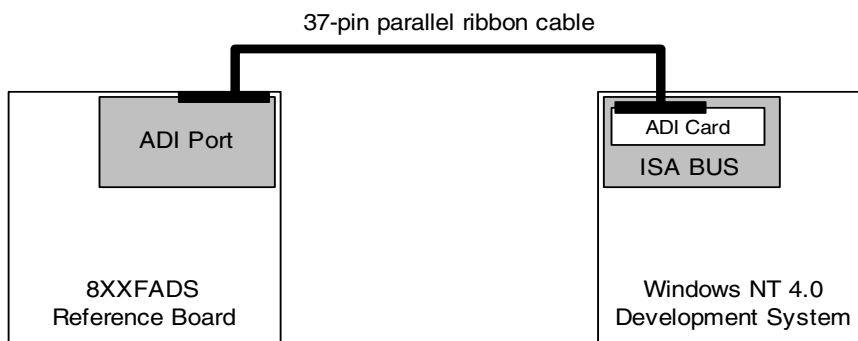
/MWOS/OS9000/821/PORTS/8XXFADS/BOOTS/INSTALL/PORTBOOT/rom.S

---

## Connecting the Host and Target

**Figure 2-1** shows a typical development system (host) and the 8XXFADS reference board (target).

**Figure 2-1 Connecting the Host and Target**



The application development interface (ADI) card, 37-pin parallel ribbon cable, and MPC8bug software are available for the SUN4 workstation (S-bus) and Windows PC (ISA bus). MPC8bug requires 600K of memory. It communicates with the 8XXFADS system through the onboard ADI debug port on the 8XXFADS via the ADI connection.

The ADI parallel port supplies a parallel link from the 8XXFADS motherboard to various host computers. They are connected with a 37-line cable to the ADI board installed in the host computer. There are four versions of the ADI board available to support a connection to a PC, Macintosh, VMEbus computer, or SUN4 workstation. You can connect the 8XXFADS board to these computers if they have the appropriate software drivers installed. Each 8XXFADS can have eight possible slave addresses set for its ADI port, which enables up to eight ADS boards to be connected to the same ADI.

If the host system used to generate rom.S is not equipped with the ADI board and MPC8bug software, you must transfer rom.S to the node that has this equipment.

The Windows host and SBC8260 target are connected via Ethernet and the EST visionICE incircuit emulator. Obtain an IP address for the incircuit emulator and connect it to the Ethernet. Connect the incircuit emulator to the SBC8260 board at the JTAG JP5 10 pin connector.

## Loading the Boot ROM on the 8XXFADS

---

The Configuration Wizard generates a bootrom file in S-record format named `rom.S`. To download `rom.S` to the 8XXFADS you must have a node configured with the Motorola ADI (Application Development Interface) board and MPC8bug software. The ADI board, MPC8bug software and manuals are available from Motorola.

Complete the following steps to download `rom.S`.

- 
- Step 1. FTP the following file to the node that has the Motorola MPC8bug software and hardware.

```
\MWOS\OS9000\821\PORTS\8XXFADS\BOOTS\INSTALL\PORTBOOT\rom.S
```

- Step 2. Power down the 8XXFADS.

- Step 3. Connect the mpc8bug ISA bus card to the 8XXFADS using the 37 pin cable and connectors.

- Step 4. Power up the 8XXFADS.

- Step 5. Run the `mpc8bug` command with no parameters to see the syntax and semantics of the command. Then run `mcp8bug` using a command similar to:

```
c:\mpc8bug> mpc8bug 1 0
```

The drive specification in your command will vary according to what drive Enhanced OS-9 is installed on your system.

- Step 6. Download `rom.S` using the commands

```
f860SARbug> reset :h
f860SARbug> loadf rom.S 0
```

- Step 7. Power down the 8XXFADS and disconnect the 37 pin cable.

- Step 8. Power up and boot the 8XXFADS.

# Loading the Boot ROM on the SBC8260

The Configuration Wizard generates a bootrom file in S-record format named rom.S. The rom.S file must be converted to before it can be loaded into the flash memory on the SBC8260.

Complete the following steps to download rom.bin.

- 
- Step 1. Connect the visionICE to the Ethernet and to the JTAG JP5 10 pin connector on the SBC8260.
  - Step 2. Power on the visionICE and the SBC8260.
  - Step 3. Start the visionICE software using:  

```
Start> Programs> visionICE> visionICE Utility Panel
```

 Connect to visionICE Over NET xxx.xxx.xxx.xxx? **Yes**  
 If the >ERR> prompt appears enter:  

```
>ERR> in
```

 The >BKM> prompt should then be presented
  - Step 4. Open the Configuration and Communications Dialog window using the 1st icon on the visionICE Utilities window.
  - Step 5. Select the **Communications** tab. Select the **Connect** button and wait for the connect to be established.
  - Step 6. Open the Program Target Flash or visionICE Flash Card window, 4th icon on the visionICE Utilities window.
  - Step 7. Select the PC Host File Name and PATH:  

```
\mwos\OS9000\8260\PORTS\SBC8260\BOOTS\SYSTEMS\PORTBOOT\rom.bin
```
  - Step 8. Select the Programming Algorithm (Flash Device(s) and Configuration:  

```
AMD 29F080 [1024 x 8] 4 devices
```
  - Step 9. Select the Base Address **FE000000**.
  - Step 10. Select **Erase All**.

Step 11. Select Start Address 00000000.

Step 12. Select Erase and Program.

---





---

## Chapter 3: Protocol Driver spatmsig and its Descriptors

---

The code for the Harris & Jeffries, Inc., Soft-ATM™ Signalling Software Subsystem and the Soft-ATM™ ILMI Subsystem was significantly modified for porting to OS-9. These changes enable the spatmsig protocol driver module to run in system-state.

This chapter describes the modifications made to the spatmsig protocol driver for porting to OS-9. It includes the following sections:

- **The External Tasking Model**
- **User Profiles**
- **Protocol Data Units**
- **Device Descriptors**



MICROWARE SOFTWARE

# The External Tasking Model

---

In the Harris & Jeffries, Inc., internal tasking model the ATM Signalling layer and/or the SSCOP layer run as user-state processes communicating via mailboxes.

As a result, the Harris & Jeffries, Inc., external tasking—or non-internal tasking—model was chosen for porting Soft-ATM™ Signalling to OS-9.

In the External tasking model, all the signalling layers (including ILMI) are contained in one protocol driver module. This model operates in the following manner:

1. In the receive direction, a protocol data unit (PDU) destined for the IPoA layer is delivered to a thread that runs to completion. Completion occurs when the PDU is delivered to process `spf_rx`.
2. `spf_rx` delivers the PDU to the driver stacked immediately above `spatmsig`—in this case the IPoA driver `sp1577`.
3. PDUs containing signalling messages destined for the layers Q.2110 or Q.2931 are handled within those layers.




---

## For More Information

See file `SPHJ/SIG/SYSTEM/hjsssopt.h` for the selection of the tasking model.

---




---

## Note

Global variables were moved into per path static storage or into driver static storage.

---

## User Profiles

---

The ATM Signalling layer API uses a “user profile” as an abstraction to represent a client. A client registers a user profile with ATM signalling.

The user profile is a structure—`HJCC_USR_PROF`—defined in the `SPHJ/INC/usr_prof.h` file. This structure is used when the inbound and outbound signal handlers are registered.

In addition, the `HJCONTEXT` field of `HJCC_USR_PROF` is a pointer to the per path static storage, which is allocated in `dr_setstat`, `SPF_SS_OPEN`. The one-to-one mapping of paths to per path static storage in the `HJCONTEXT` field enables the Soft-ATM™ Signalling Subsystem to identify a client.

## Protocol Data Units

---

In OS-9, a protocol data unit (PDU) is passed from one end of a protocol stack to the other end in memory buffers called mbufs.

PDUs passing through the Harris & Jeffries, Inc. portion of spatmsig, however, must be handled using the Harris & Jeffries, Inc., buffer management system—inside memory buffers called usrbufs.

As a result, PDUs handled within spatmsig (Q.2110 and Q.2931 layer messages) must be copied from mbufs into usrbufs. PDUs destined for drivers or applications above spatmsig are handled within spatmsig by usrbufs containing one datum, a pointer to an mbuf.

## Device Descriptors

---

The `SPHJ/spatmsig/spf_desc.h` file contains constant definitions for building the device descriptors `hjn0` and `hju0`.

The constant `NODE_TYPE` initializes the interface type field in the Q.2931 Signalling Control Block structure (QSCB) and the SSCOP Control Block structure (SCB).

Possible values for `NODE_TYPE` are defined in the `SPHJ/INC/hjiftype.h` file. In this release, the value `SIG_IF_TYPE_UNI_NETWORK_4_0` is used for descriptor `hjn0` and `SIG_IF_TYPE_UNI_USER_4_0` is used for `hju0`.



---

# Chapter 4: Building the Libraries, Driver and Descriptors

---

This chapter lists various procedures for building libraries and the driver and descriptors for Soft-ATM™ for OS-9. It includes the following sections:

- **Building the Libraries**
- **Building the Driver spatmsig and Descriptors**



MICROWARE SOFTWARE

## Building the Libraries

---

The `SPHJ/SYSTEMS/hj_opt.h` file selects the features included in the `spatmsig` module. In this release, ATM Signalling and ILMI are selected. The `hj_opt.h` file also selects the following characteristics:

- Big-endian mode
- Debugging options
- Support for ATM Signalling versions 3.0, 3.1 and 4.0
- Private Network-to-Network Interface (PNNI) options
- Interim Inter-switch Signalling Protocol (IISP)



---

### Note

The size of the `spatmsig` module can be reduced by deselecting some of these items.

---

The `SPHJ/LMI/SYSTEM/hjil_opt.h` file selects features to be included in the ILMI agent and manager parts.



Logging to data module `spatmsig_dbg` can be disabled using

```
#define HJIL_LOG    0
```

in the `SPHJ/ILMI/SYSTEM/hjil_opt.h` file. This has been done to greatly reduce the data written to `spatmsig_dbg`.

To compile and link the libraries for linking `spatmsig` execute:

```
\MWOS\SRC\DPIO\SPF\DRVR\SPHJ\hjsss.bat  
\MWOS\SRC\DPIO\SPF\DRVR\SPHJ\ilmi.bat  
\MWOS\SRC\DPIO\SPF\DRVR\SPHJ\src.bat  
\MWOS\SRC\DPIO\SPF\DRVR\SPHJ\sysfcpcs.bat
```

# Building the Driver spatmsig and Descriptors

---

Complete the following steps to build spatmsig and its descriptors:

---

Step 1. Change to the following directory:

`\MWOS\SRC\DPIO\SPF\DRVR\SPHJ`

Step 2. Compile and link the spatmsig driver by entering the following command:

`os9make -f spfdrv.mak`

Step 3. Compile and link the debug version of spatmsig by entering the following command:

`os9make -f spfdbg.mak`

Step 4. Compile and link the descriptors for spatmsig by entering the following command:

`os9make -f spfdesc.mak`

---

---

# Product Discrepancy Report

---

To: Microware Customer Support

FAX: 515-224-1352

From: \_\_\_\_\_

Company: \_\_\_\_\_

Phone: \_\_\_\_\_

Fax: \_\_\_\_\_ Email: \_\_\_\_\_

Product Name: Soft-ATM™ for OS-9

Description of Problem:

---

---

---

---

---

---

---

---

---

---

---

---

Host Platform \_\_\_\_\_

Target Platform \_\_\_\_\_



MICROWARE SOFTWARE