Home

OS-9® for MC68328ADS Board Guide

Version 3.3

RadiSys
THE POWER OF WE

www.radisys.com
Revision C • July 2006

Copyright and publication information

This manual reflects version 3.3 of Microware OS-9 for 68K. Reproduction of this document, in part or whole, by any means, electrical, mechanical, magnetic, optical, chemical, manual, or otherwise is prohibited, without written permission from RadiSys Microware Communications Software Division, Inc.

Disclaimer

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

Reproduction notice

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

Table of Contents

Chapter 1: Supported Uses of This Guide

This chapter contains the following sections:

- **Purpose**
- **Using This Document For Evaluation**
- **Using This Document For Reference and Tutorial**

Purpose

This guide gives you a head start in working with the MC68328ADS board. The guide provides specific information on the use of the board with Microware products. It is possible you received this guide as part of a board evaluation package or an OEM Developer's Kit.

This document serves two purposes:

- Evaluation — Use with a stand alone software package, enabling evaluation of the OS-9® system using a single configuration of a specific board.
- Tutorial — Use with other documents in an OEM Developer's Kit, enabling walk through of a single configuration of the board. At various steps, other Microware manuals providing greater detail for custom configuration of a hardware port are referenced.

Chapters 3 and 4 of this document provide reference information specific to the MC68328ADS board and to the 328 processor.

Subsystem demos are provided in the latter part of this manual.

Using This Document For Evaluation

To evaluate an OS-9 package for the MC68328ADS, see Chapter 3: Example Installation and Configuration as a step-by-step guide to set up the hardware, software, and OS-9 as needed to perform initial installation of OS-9 on the MC68328ADS board.

A general software development background is recommended but previous OS-9 experience or previous configuration and porting experience is not necessary. This installation may require minimal code reconfiguration to support debugging. Specific hardware and software listed in Chapter 3: Example Installation and Configuration are prerequisites.

For More Information

To gain understanding of OS-9 theory and operation and for configuration and porting details supporting a custom configuration, reference ***OS-9 OEM Installation Manual***.

Chapter 3 describes a single, limited example of an OS-9 installation designed with this specific board configuration in mind and does not explain all of the configuration possibilities available.

Using This Document For Reference and Tutorial

To perform a custom configuration, use the example in Chapter 3: Example Installation and Configuration of this manual as a tutorial for the initial installation of OS-9 on the MC68328ADS board if you have the required hardware and software. After you follow the steps in this tutorial, it should be relatively easy to port OS-9 to target hardware that closely resembles the standard microprocessor evaluation board that is used as the example target system. Hardware that uses a significantly different design lengthens the porting process.

For More Information

Porting OS-9 is explained in detail in ***OS-9 OEM Installation Manual***. It is imperative that this manual be reviewed prior to attempting a port of OS-9 to new hardware.

Use Chapter 4: Board-Specific Considerations of this manual as a reference for board or processor specific information, complemented by the configuration and porting information in the ***OS-9 OEM Installation Manual***.

Chapter 2: Installation Considerations

This chapter contains the following chapters:

- MC68328ADS Overview
- Development System Requirements
- Target Hardware Requirements
- Installation Summary and Examples

MC68328ADS Overview

This section provides:

- Summary of the supported hardware
- Special hardware considerations for this board
- Summary of the supported installation options
- Specific hardware and software used in the example installation documented in this chapter

Development System Requirements

The list below identifies minimum hardware and software required to port OS-9 to a target system. The OS-9 for Embedded Systems installation is a source release for OEMs. It is designed for installation on a host system.

- Computer with an 80x86 family processor running.
- Windows NT, Windows 2000, or Windows XP.
- The Microware Hawk integrated development environment.

Microware OS-9 for 68K includes the Ultra C/ C++ compiler, the assembler and linker, and all utilities necessary to rebuild OS-9.The installation procedure may vary at times according to the type of development system being used. This is noted where applicable.

Additional Host Hardware Requirements

The following hardware is also required for some targets.

- EPROM device programmer

Target Hardware Requirements

This software package supports the Motorola 68328ADS board using a 328 processor. The software in this package is organized within the `MWOS\OS9\68000\PORTS\MC328ADS` directory.

Special Hardware Considerations

Not all platforms supported by this package have the same set of peripherals. Therefore, the example boot image contained within is tuned to support as many platforms as possible. The following section contains only one specific example. After booting OS-9 initially from the sample boot, you may reconfigure the system to directly fit your requirements.

Note

Read the ***OS-9 OEM Installation Manual*** included with your software distribution for a better understanding of the purpose of each of the modules contained in this distribution. The variety of ways that the software can be configured to meet your needs are explained in this manual.

The following MC68328ADS capabilities are supported:

- Real time clock
- Counter/Time
- Power Management
- Liquid crystal display (LCD - 1 bit and 2 bit grayscale)
- Touch-panel A/D interface
- Memory only PCMCIA cards (Linear Flash)

The following MC68328ADS capabilities are not supported:

- Infrared data association (IRDA)
- Master and slave serial peripheral interface (SPI) ports
- Hardware and software watchdog timers
- Universal asynchronous receiver/transmitter (UART)
- Pulse width modulation (PWM) output

Installation Summary and Examples

This section briefly explains the installation options available for the MC68328ADS series boards. The option selected may be dependent upon the following:

- Development tools available, especially the debugger. The following methods of debugging are available:
  - System-state debugging using the low-level system through RomBug. (Example port default.)
  - Process level debugging using SoftStax through Hawk. (This example installation walk-through configures the boot for this option.)
- Specific target board configuration.
- Specific hardware available for use as a host system and the methods available to transfer data to the target system.
- Degree of OS-9 configuration desired or required prior to installation.

The installation method supported for this board is burning the distribution image into EPROMs. This option is explained in the example walk through in the next chapter.

For More Information

Read the ***OS-9 OEM Installation Manual*** before custom configuring OS-9 for a target system.

Chapter 3: Example Installation and Configuration

This chapter contains the following sections:

Set-Up Used In This Port Example

The example in the following section assumes the following:

- An MC68328ADS target, Revision 1.1 or 2.0.
- Windows-based PC for use as a cross host system.
- Erasable programmable read only memory (EPROM) device programmer and related software.
- Two 512K, model 27C040 EPROM chips.
- Four 128 K or two 512 K static random access memory (SRAM) chips for "Boot from ROM".
- Eight 128 K or four 512 K static random access memory (SRAM) chips for "Boot from RAM".
- Microware distribution software is loaded on the host system. Follow the instructions included with the distribution media to load the distribution software and read all .txt files or README files included on the media in the root before proceeding.
- Process-level debugging using SoftStax through Hawk.
- The example uses `<MWOS>` as the root directory. If necessary, replace directory names in the example with directory names specified during your installation of the software.

Booting the Distribution Image

The distribution boot image resides in the PORTS directory in CMDS\BOOTOBJS\NOBUG\bootrom. Use this image to boot the MC68328ADS series board from a PC, following instructions provided in this section.

- **Making Boot Modules and the Bootrom File**
- **Burning the Distribution Image to EPROMs**
- **Inserting EPROMs and SRAMs and Setting Jumpers**
- **Target Configuration**

The MC68328ADS boards are designed for programming an operating system into EPROMs. This example supports that mode of operation exclusively.

Making Boot Modules and the Bootrom File

Microware provides a distribution image. However, if so desired, you can change the image to meet your specifications at this point.

After making any changes to the boot files, perform the following steps from a DOS prompt to create boot modules and the bootrom file.

Step 1. Change directories:

```
cd \<MWOS>\OS9\68000\PORTS\MC328ADS
```

Step 2. Type:

```
os9make
```

Note

This step is only necessary of you have changed the distribution image.

Burning the Distribution Image to EPROMs

Burn an image of the distribution from a host to two, 512 K, model 27C040 EPROMs.

Step 1. Change directories:

```
cd \<MWOS>\OS9\68000\PORTS\MC328ADS\CMDS\BOOTOBJS\NOBUG
```

Step 2. Transfer the `bootrom` file to the device programmer housing the EPROM.

Inserting EPROMs and SRAMs and Setting Jumpers

Figure 3-1 and Figure 3-2 depict Revision 1.1 and Revision 2.0 of the MC68328ADS board respectively. EPROM and SRAM requirements and insertion banks are identified in Table 3-1. Jumper settings are identified in Table 3-2.

Table 3-1 EPROM and SRAM Requirements

Board Revision	Device	Size Required	Bank
1.1 and 2.0	EPROM	1 MB (2 x 512 K)	Bank 1
1.1	SRAM	1 MB (4 x 128 K or 2 x 512 K)	Start at bank 1
2.0	SRAM	1 MB (4 x 128 K or 2 x 512 K)	Start at bank 1

Table 3-2 Jumper Settings

Board Revision	Jumper Number	Jumper State	Effect
1.1	JP28	Closed	Recognize the EPROM set as 512 K each EPROM.
	JP27	Open	Recognizes SRAM as 128 K each.
		Closed	Recognizes SRAM as 512 K each.
2.0	JP16	Closed	Recognize the EPROM set as 512 K each EPROM.
	JP17	Open	Recognizes SRAM as 128 K each.
		Closed	Recognizes SRAM as 512 K each.

Figure 3-1 MC68328ADS Revision 1.1

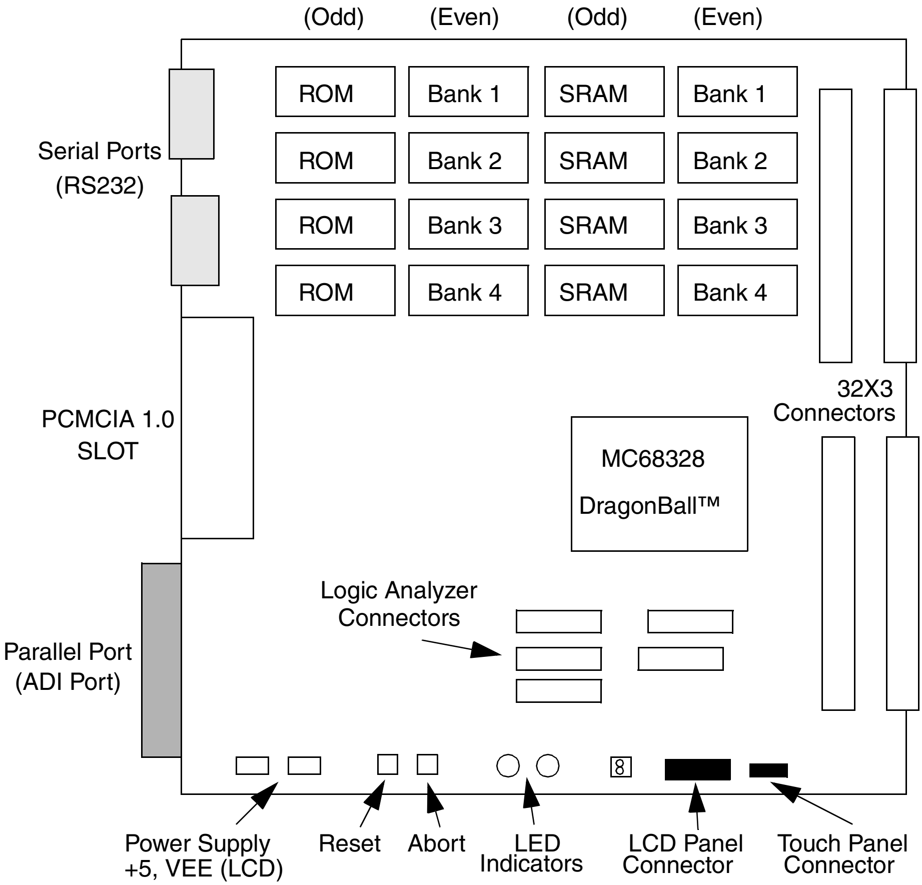
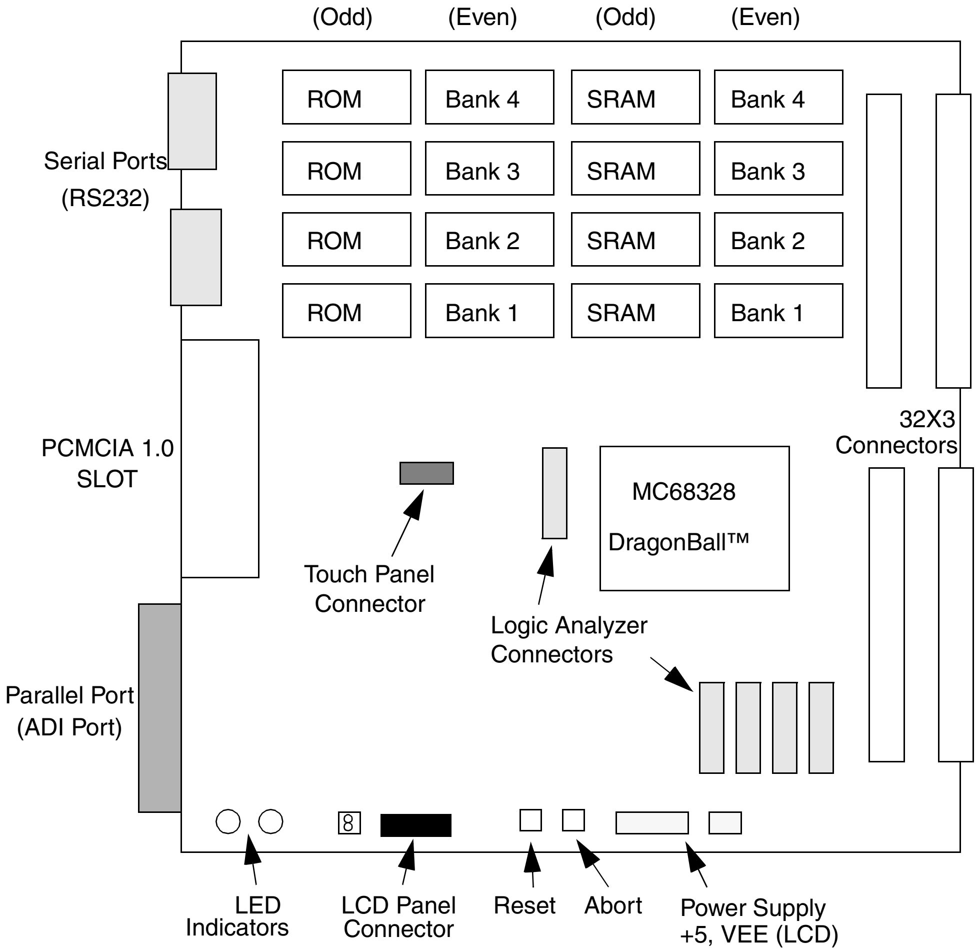


Figure 3-2 MC68328ADS Revision 2.0



Target Configuration

Step 1. Connect the MC68328ADS serial port P3 to the PC serial port.

Step 2. Configure your terminal or terminal emulation software for the following:

9600 baud

8 data bits

1 stop bit

No parity

XON\XOFF flow control

Step 3. Connect the power to the MC68328ADS board and verify that it is working correctly. The Power light emitting diode (LED) should be lit and the Run LED should flash slowly. See the MC68328ADS installation manual for specific diagnostics information.

Booting OS-9

Verify that the system is initialized properly. After the system is powered up, an OS-9 bootstrap message and boot menu similar to the following displays.

```
OS-9/68K System Bootstrap
```

Step 1. The following message, boot menu, and prompt displays:

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

Boot Manually Loaded Bootfile Image - <ml>
Boot from ROM ----------------------- <ro>
Load from ROM ----------------------- <lr>
Restart the system ------------------ <q>

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

Step 2. At the prompt, enter:

```
ro
```

The following message displays upon entry of a boot method:

```
Now searching memory ($00400400 - $004FFFFF) for an OS-9
Kernel...

An OS-9 kernel module was found at $00414584
A valid OS-9 bootfile was found.

$
```

You are now ready to reconfigure the operating system to use various OS-9 subsystems. The next section describes configuration by subsystem.

Debugging with Hawk Over SLIP

Using Hawk for debugging over a SLIP connection involves the following basic steps:

- Setting up the Hardware connections.
- Setting up the Windows host system.
- Setting up the OS-9 target system.
- Using the debugger.

For More Information

For a complete description of this procedure see the **Debugging with Hawk Over SLIP** section in the ***Using Hawk*** manual.

Chapter 4: Board-Specific Considerations

This chapter contains the following sections:

- **Introduction**
- **Software Overview**

Introduction

This chapter contains information that is specific to the MC68328ADS boards.

For More Information

For general information on porting OS-9, see the ***OS-9 OEM Installation Manual***.

Software Overview

This section contains a list of the board specific low-level system modules and the high-level system modules with a brief description. Low-level modules are those that are used before the operating system comes up. The high-level modules are used after the kernel is brought up.

For More Information

For a list of all of the OS-9 modules common to all boards, see the ***OS-9 OEM Installation Manual***.

Low-Level System Modules

The following low-level system modules are tailored specifically for the MC68328ADS target platform. These modules reside in the following directory, unless otherwise noted:

```
<MWOS>/OS9/68000/PORTS/MC328ADS/CMDS/BOOTOBJS
```

Cache

`flush68k` Provides cache flushing routine to the low-level system. This module is in the following directory:

```
<MWOS>/OS9/68000/CMDS/BOOTOBJS/ROM
```

Console Drivers

`io68681` Provides console services for the 328.

High-Level System Modules

The following OS-9 system modules are tailored specifically for your MC68328ADS platform. Unless otherwise specified, each module can be found in a file of the same name in the following directory:

`<MWOS>/OS9/68000/PORTS/MC328ADS/CMDS/BOOTOBJS`

Real Time Clock Driver

`rtc68328`	Provides OS-9 access to the real time clock.

Ticker

`tk68328`	Provides the system ticker.

Abort Handler

Serial Ports

`sc68681`	Provides support for the CPM SMC and SCC UARTS serial port. The descriptors provided for this driver are named `term` and `t1` and are located in the /DESC subdirectory.

RAM Disk Descriptors

The RAM disk device is configurable so its descriptor is built in the `PORTS` directory. Sample descriptors can be found for each of `ram` and `r0`. The r0 descriptor is in the following subdirectory: `/DESC/RAMDISK`

Pipe Descriptors

There are three pipe descriptors: `pipeman`, `pipe`, and `pipe2`.

Chapter 5: Subsystem Demos

This chapter contains the following sections:

- **Introduction**
- **MAUI**
- **Power Management Subsystem**

Introduction

OS-9 subsystem demos documented in this chapter support:

- MAUI®
- Power Management Subsystem

MAUI

This section identifies a variety of demonstrations.

Running the Demos

It is important to use a supported LCD for the demonstrations. Table 5-1 identifies characteristics of the supported Alps display.

Table 5-1 Display Characteristics

Display	Resolution	Number of Bits
Alps	240 x 160	1 = Black/White 2/4 = Grayscale

The following table identifies modules required in the bootlist to startup MAUI demos.

Table 5-2 Bootlist for Demo start-up

Module	demo	mauidemo	slidshow
`mail`		●	
`email`		●	
`falarm`	●		●
`game`		●	
`intro`	●		

Table 5-2 Bootlist for Demo start-up (continued)

Module	demo	mauidemo	slidshow
mainmenu		●	●
message1			●
message2			●
secure1	●		●
secure2	●		●
secure3	●		●
secure4	●		●
secure5	●		●
warnalarm	●		●
slide.img	●		●
default.fnt	●	●	

slidshow

Function: Cycles through ten images. The images are designed to illustrate the application of MAUI to a personal communication device in an industrial setting.

Display: Alps

Usage: `slidshow <option>`

Example: `slidshow -r=2`

MAUI Descriptor: `gfx`

MAUI Driver: `mauidrvr`

Options:

`-d` Display each slide once at 5 sec/image.

`-r=<num>`
Display demo recursively at <`num`> where <`num`> = sec/image. Valid <`num`> values are 2, 5 or 10.

`-h=<num>`
Halt demo on image <`num`>. Valid <`num`> values are 1 through 10.

Note

Specification of an option is required.

To exit the `slidshow` demo, press [CTRL-E].

penp

Function:	Acts as an etch-a-sketch, echoing user input on the display.
Display:	Alps

Note

The following command must be issued at the OS-9 prompt before executing the `penp` demo:

```
maui_inp<>>>/nil&
```

This command initializes the display to receive input from the touchscreen.

Usage:	`penp <MAUI descriptor name> <touchpad descriptor>`
Example:	`penp gfx tpad`
MAUI Driver:	`mauidrvr`
Touchpad Driver:	`tpaddrvr`

demo

Function:	Displays images and demonstrates touchscreen capabilities.
Display:	Alps

Note

The following command must be issued at the OS-9 prompt before executing `demo`:

```
maui_inp<>>>/nil&
```

This command initializes the display to receive input from the touchscreen.

Usage:	`demo`
MAUI Driver:	`mauidrvr`
Touchpad Driver:	`tpaddrvr`
MAUI Descriptor:	`gfx`
Touchpad Descriptor:	`tpad`

mauidemo

Function:	Demo program showing how to output graphics and text to a drawmap in MAUI and wait for user input.
Display:	Alps
Usage:	`mauidemo /<descriptor name>`
Example:	`mauidemo /gfx`
MAUI Driver:	`mauidrvr`

draw

Function:	Displays images of 1, 2 or 4 bits.
Display:	Alps
Usage:	`draw <descriptor name> N` Where *N* equals the number of bits per pixel (1, 2, or 4).
Example:	`draw /gfx 2`
MAUI Driver:	`mauidrvr`

Note

The MC328ADS supports only modes of 1 or 2 bits per pixel.

MAUI Demo Summary

```
Module:  slidshow
Source:  \MWOS\SRC\WIRELESS\DEMO\SLIDESHOW
Makefile:\MWOS\OS9\68000\PORTS\MC328ADS\MAUI\DEMOS\SLIDESHOW

Module:  penp
Source:  \MWOS\SRC\WIRELESS\DEMO\PENP
Makefile:\MWOS\OS9\68000\PORTS\MC328ADS\MAUI\DEMOS\PENP

Module:  demo
Source:  \MWOS\SRC\WIRELESS\DEMO\DEMO328
Makefile:\MWOS\OS9\68000\PORTS\MC328ADS\MAUI\DEMOS\DEMO

Module:  mauidemo
Source:  \MWOS\SRC\WIRELESS\DEMO\MAUIDEMO
Makefile:\MWOS\OS9\68000\PORTS\MC328ADS\MAUI\DEMOS\MAUIDEMO

Module:  draw
Source:  \MWOS\SRC\WIRELESS\DEMO\DRAW
Makefile:\MWOS\OS9\68000\PORTS\MC328ADS\MAUI\DEMOS\DRAW
```

Power Management Subsystem

Power Management is automatically installed. Disable it by following the instructions:

Step 1. At the DOS prompt, change directories:

```
cd \MWOS\OS9\68000\PORTS\MC328ADS\INIT\DD_DISK
```

Step 2. Edit `makefile`. Replace the following line:

```
   PWRMAN = -asaPWRMAN
```

with

```
   PWRMAN = #-asaPWRMAN
```

Step 3. Save changes to `makefile` and exit.

Step 4. At the DOS prompt, change directories:

```
cd \MWOS\OS9\68000\PORTS\MC328ADS
```

and type:

```
   os9make
```

Step 5. Burn EPROMS as outlined in Burning the Distribution Image to EPROMs section.

For More Information

See ***Power Management Subsystem Specification*** for more information about the `pwrstat` utility.

Using the pwrstat Utility

The `pwrstat` utility enables viewing of some internal Power Management Subsystem structures, such as version label, Device Registry Table entries, and Power State Table entries. Commands enabling viewing of Power Management Subsystem tables are defined following.

To view the subsystem version, at the prompt type:

```
pwrstat -v
```

An example version display follows:

```
$ pwrstat -v
Motorola M68328ADS Board  OS-9/68K V3.1

Installed PwrMan Version ID = 'PwrMan 1.0.Beta'
```

The Device Registry Table identifies power aware device to be power managed and device driver or `SysIF` callback functions or parameters. To view the Device Registry Table, at the prompt type:

```
pwrstat -d
```

An example Device Registry Table display follows:

```
Motorola M68328ADS Board OS-9/68K V3.0.3

Device Registry Table

       Id         C/B Ptr   Data Addr Device Param
---------------- --------- --------- ------------
cpu              $000a3528 $000bb714 $00000000
```

The Power State Table identifies device power control. To view the Power State Table, at the prompt type:

```
pwrstat -p
```

A Power State Table display follows:

```
Motorola M68328ADS Board OS-9/68K V3.0.3

Power State Table

       Id          Syslevel   Devlevel   Priority  Data Addr  C/B Ptr
---------------- ---------- ---------- ---------- --------- ---------
gfx              109        0          1          $00000000 $00000000
cpu              109        48         4294967295 $000bb714 $000a3528
cpu              110        32         4294967295 $000bb714 $000a3528
gfx              111        255        1          $00000000 $00000000
cpu              111        16         4294967295 $000bb714 $000a3528
```

To display other options available, at the DOS prompt type:

```
$ pwrstat -?
Syntax:    pwrstat [<opts>]
Function:    Microware Power Manager Display Table
 Options:
    -a[=]id syslevel devlevel priority     Add a power state entry
    -c syslevel                            Change to system level
    -d                                     Display the device table
    -p                                     Display the power policy table
    -r[=]id syslevel devlevel priority     Remove a power state entry
    -v                                     Print out version
```

To observe power management of the LCD display, run `mauidemo`. At the prompt, type:

```
mauidemo /gfx
```

Press enter several times and observe the image on the display. After about two minutes, the screen blanks, illustrating power management of the display. Press enter again and the next image in the sequence displays on the screen.