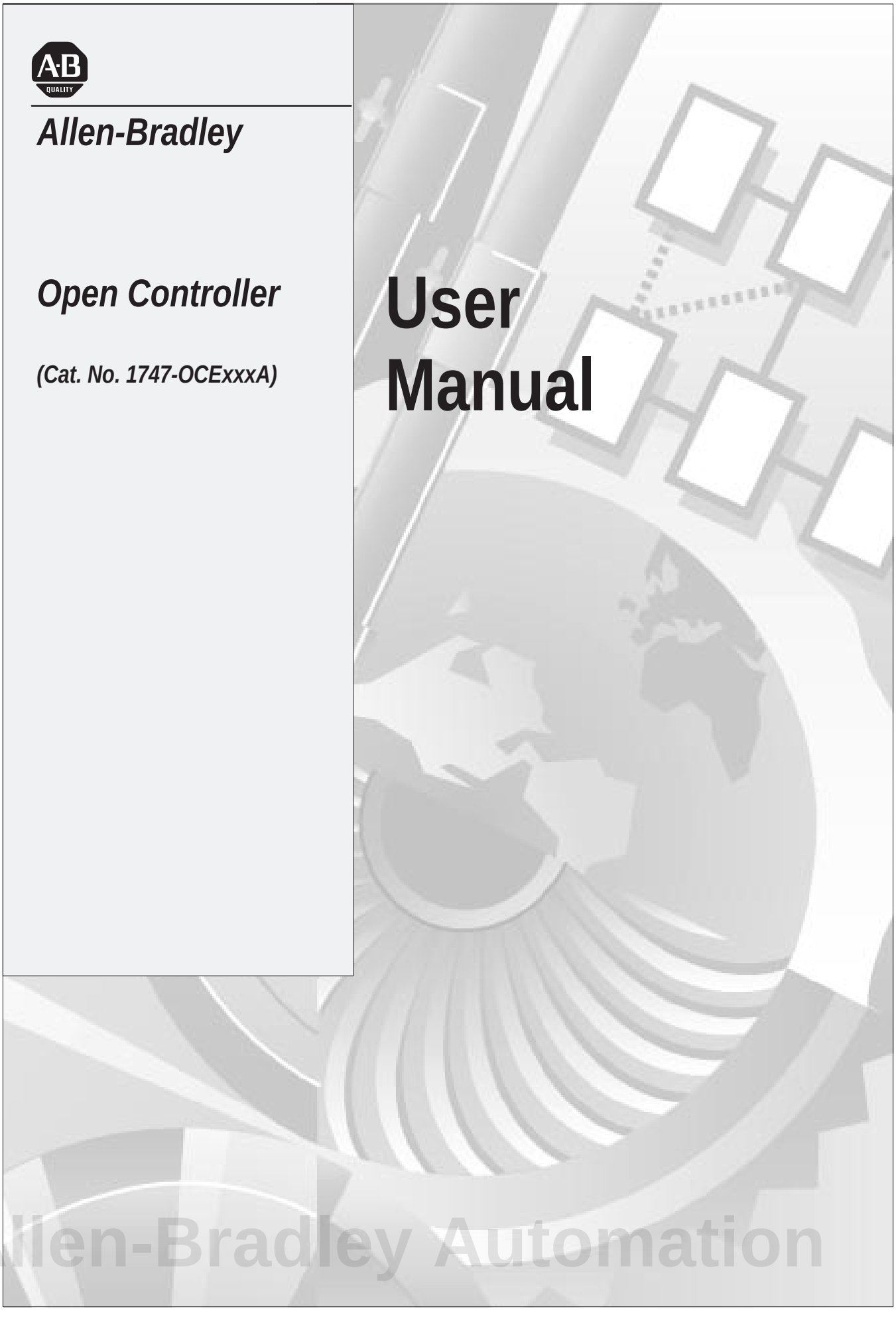


Allen-Bradley

Open Controller

(Cat. No. 1747-OCExxxA)

User Manual

Important User Information

Because of the variety of uses for the products described in this publication, those responsible for the application and use of this control equipment must satisfy themselves that all necessary steps have been taken to assure that each application and use meets all performance and safety requirements, including any applicable laws, regulations, codes and standards.

The illustrations, charts, sample programs and layout examples shown in this guide are intended solely for purposes of example. Since there are many variables and requirements associated with any particular installation, Allen-Bradley does not assume responsibility or liability (to include intellectual property liability) for actual use based upon the examples shown in this publication.

Allen-Bradley publication SGI-1.1, *Safety Guidelines for the Application, Installation, and Maintenance of Solid-State Control* (available from your local Allen-Bradley office), describes some important differences between solid-state equipment and electromechanical devices that should be taken into consideration when applying products such as those described in this publication.

Throughout this manual we use notes to make you aware of safety considerations:

ATTENTION: Identifies information about practices or circumstances that can lead to personal injury or death, property damage or economic loss.

Attention statements help you to:

- identify a hazard
- avoid the hazard
- recognize the consequences

Important: Identifies information that is critical for successful application and understanding of the product.

Using This Manual

Who Should Use This Manual

Use this manual if you are responsible for designing, installing, programming, or troubleshooting control systems that use the Allen-Bradley open controller.

This manual is a guide for using the open controller. How you program the open controller or how your application functions depends on the software options you install on the open controller.

Additional Open Controller Documentation

The following documents are available for additional information about the open controller and its options.

This document:	Has this publication number:
1747 Open Controller User Manual	1747-6.16
1747 Open Controller System Quick Start	1747-10.3
1747 Open Controller PCI Expansion Bus Installation Instructions	1747-5.16
1747 Open Controller Video Interface Module Installation Instructions	1747-5.15
1747 Open Controller PCMCIA Interface Module Installation Instructions	1747-5.13
1747 Open Controller A-B Communication Interface Module Installation Instructions	1747-5.14
1747 Open Controller A-B Communication Interface Module User Manual	1747-6.18
1747 Open Controller FlashDrive Installation Instructions	1747-5.17
1747 Open Controller System Memory Installation Instructions	1747-5.22
1747 Open Controller Chassis Fan Installation Instructions	1747-5.23
1747 Open Controller IDE Interface Module for IDE-Compatible ATA Devices (PC Cards) Installation Instructions	1747-5.29
1747 Open Controller IDE Interface Module for an Internally-Mounted 2.5” ATA Drive Installation Instructions	1747-5.30
1747 Open Controller API Software User Manual	1747-6.19

Each open controller component ships with installation instructions. The user manuals are part of the open controller documentation set (catalog number 1747-OCDOC1) so you can order as many copies as you need. The documentation set includes one copy of each open controller document (as listed above).

In addition to the above documentation, the:

- 1747-OCVGA1 video interface module includes a disk with documentation for the video drivers
- 1747-OCPCM2 PCMCIA interface module includes a disk with CardSoft™ documentation.

Reference Material

The following books might be useful as you develop your open controller applications:

This document:	By:	Has this ISBN number:
PC System Architecture Series PCI System Architecture	MindShare, Inc. Addison-Wesley Publishing Company	ISBN: 0–201–40993–3
PC System Architecture Series ISA System Architecture	MindShare, Inc. Addison-Wesley Publishing Company	ISBN: 0–201–40996–8
PC System Architecture Series PCMCIA System Architecture	MindShare, Inc. Addison-Wesley Publishing Company	ISBN: 0–201–40991–7
The PCMCIA Developer's Guide	Michael T. Mori and W. Dean Welder	ISBN: 0–9640342–1–2
PCI Hardware and Software Architecture and Design	Edward Solari and George Willse	ISBN: 0–929392–28–0

Table of Contents

Defining System Settings — Chapter 5

Installing Software — Chapter 6

Using the Open Controller — Chapter 7

Troubleshooting — Chapter 8

Specifications

Appendix A

Handling the 1747-BA Battery

Appendix B

Notes:

Introducing the Open Controller

Introducing the Open Controller CPU Module

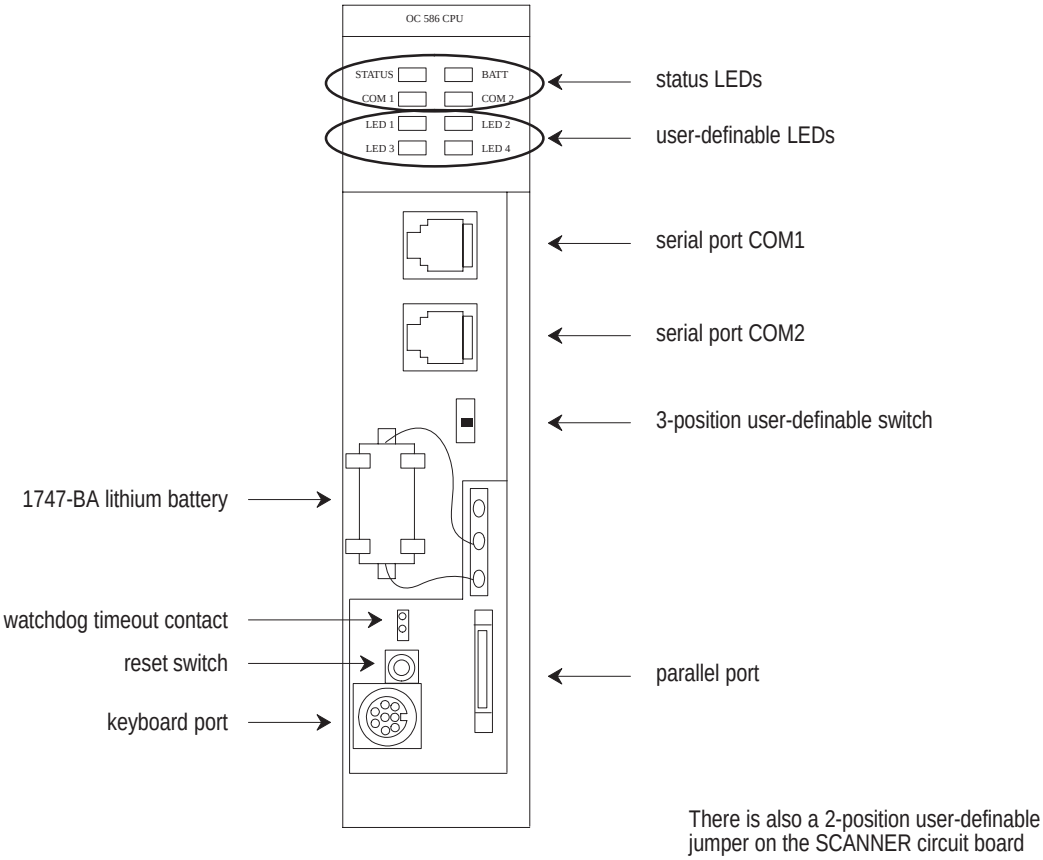
The open controller CPU module (Figure 1.1):

- has a 5x86 class CPU and a 1746 local I/O scanner
- resides in the left slot of any 1746 chassis
  (you need a series B or greater chassis if you plan to use any open controller option modules)
- can address all 1746 I/O and communication modules
  (including 1747-SN remote I/O and 1747-SDN DeviceNet™ scanners)
- supports 4, 8, 16, 32, or 64 Mbytes of system memory (DRAM)

  Note: Although A-B does not currently offer 64 Mbytes of DRAM, you can purchase the DRAM from other vendors as long as it meets the specifications in appendix A.
- supports the optional FlashDrive™ for non-volatile storage

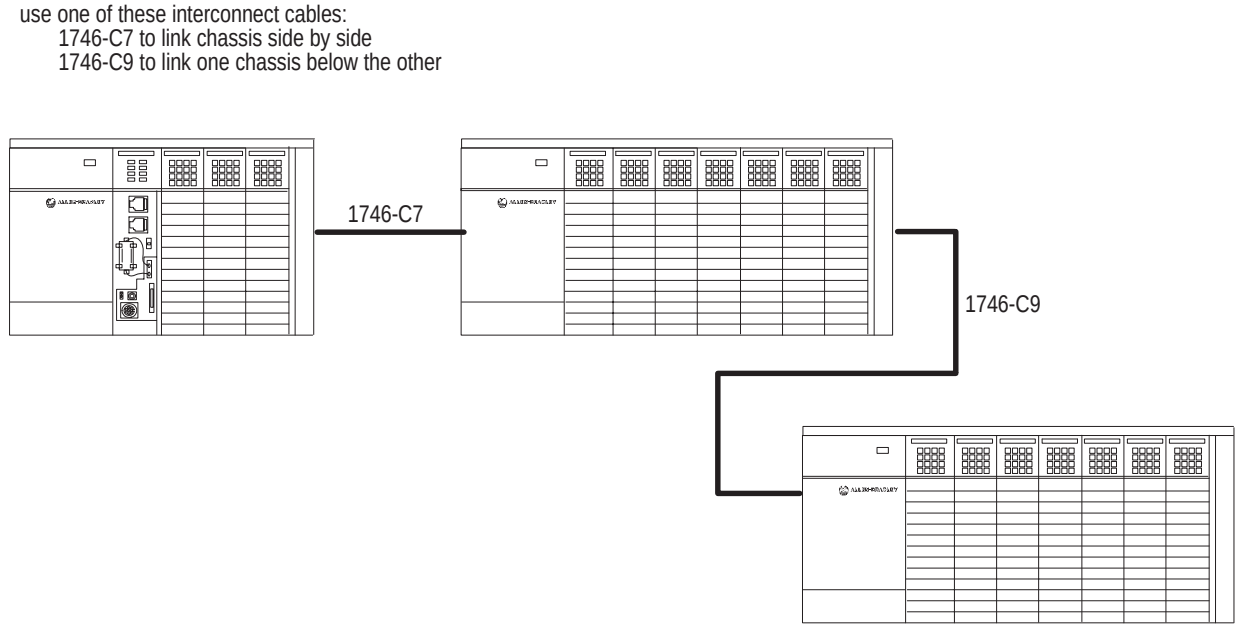
  The FlashDrive appears as an IDE hard drive to the operating system
- has a dual-port (8K bytes of shared memory) area that provides an interface to the integrated 1746 local I/O scanner
- can run any PC-based software package

Figure 1.1
Front panel of a 1747 open controller CPU module



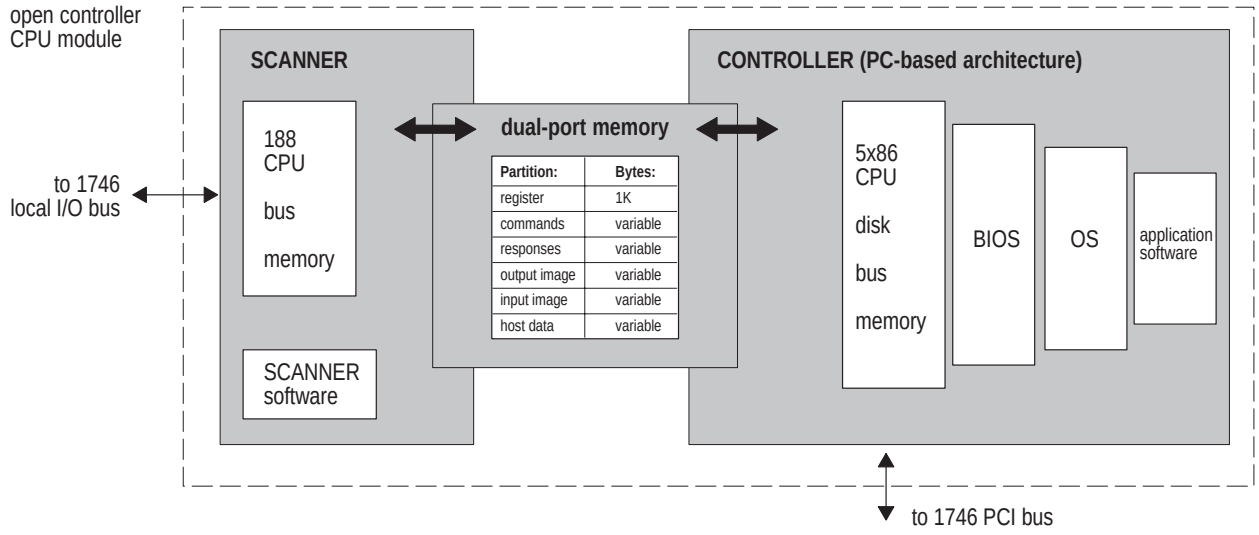
The open controller can address any valid 1746 chassis configuration. You can interconnect as many as 3 chassis (for a maximum of 30 slots) to create an expanded, local-chassis system for one open controller. Figure 1.2 shows how to interconnect chassis.

Figure 1.2
Interconnecting 1746 chassis to create an expanded, local chassis



Understanding the Open Controller Architecture

The open controller architecture consists of two CPUs (SCANNER and CONTROLLER) that share dual-port memory. The SCANNER scans the 1746 local I/O bus and reads/writes inputs and outputs to/from the dual-port registers. The CONTROLLER has a PC-based architecture with a 5x86 class CPU to run your application.



The dual port is an 8K byte memory partition that provides an interface between the integrated SCANNER and your application software that resides on the CONTROLLER.

Your application (the code you develop) uses the dual-port memory to communicate with the SCANNER and to handle control functions on the 1746 backplane, such as:

- SCANNER commands and responses
- battery and SCANNER status
- scan rate frequency and timing
- I/O image counters
- priority messages and interrupts
- semaphores to ensure data integrity
- software-generated watchdogs
- control of the 4 user-definable LEDs, the 3-position switch, and the 2-position jumper

The SCANNER functionality of the dual port supports I/O control functions, such as:

- synchronizing scans to the application
- forcing I/O
- discrete-input interrupts
- I/O module-driven interrupts (such as for the 1746-BAS module)
- I/O slot enables and disables
- I/O resets

In addition to providing access to the control SCANNER, the dual-port memory also provides non-volatile storage for:

- I/O values
- application parameters (timers, counters, presets)

Communicating with the Open Controller

Your CONTROLLER application software communicates with the SCANNER to control I/O by reading and writing to the dual-port registers. Your application can access the dual-port registers using:

- the function calls in the open controller API (application program interface) to access dual-port registers

  Choose 1747-OCAPID for DOS or 1747-OCAPINT for Windows NT.
- a software package, such as Controlware, that has built-in drivers that access the open controller dual-port registers
- the drivers you develop

  The dual-port documentation is available through the A-B technology licensing program. For more information, contact:

  Bill Waltz
  1747-OC Product Manager
  (216) 646-3870

Using the API software

The APIs (1747-OCAPID for DOS and 1747-OCAPINT for Windows NT) provide libraries of C function calls for interfacing with the open controller dual-port memory. Each library provides calls for typical control functions, such as:

- configure I/O files
- initialize the SCANNER
- define user LEDs, 3-position switch, 2-position jumper, and external watchdog
- read CONTROLLER status
- read/write input/output data
- enable/disable forces

The APIs support Microsoft and Borland C compilers in the DOS and Windows NT environments. The DOS API is compiled as a 16-bit MS-DOS library using the 80386 instruction set. The Windows NT API is a standard 32-bit DLL.

Using Controlware

Controlware is a deterministic, multi-tasking environment that runs on DOS systems. Controlware incorporates a real-time control executive and development tools for machine, process, and motion control applications.

The Controlware development tools include:

- Commander for Controlware: an MMI package that lets you create graphical operator interface screens
- PRO for Controlware: a relay-ladder programming package that allows on-line, real-time programming

Controlware is best suited for:

- distributed control applications
- applications or users that require the control to be augmented by C/C++ or other standard, PC-type programming languages

Controlware lets you mix multiple programming languages, such as ladder, C, BASIC, and assembler.

Controlware interfaces with the open controller dual-port memory to control local I/O. Controlware can:

- read and write I/O data
- configure 1746 I/O and 1747 scanners using G files
- transfer data using M0/M1 files

Controlware provides the necessary drivers to control remote I/O through the 1747-OCKTX, -OCKTXD communication interface module or the 1747-SN module.

Installing the Open Controller

Before You Begin

You can install only one open controller CPU module per open controller chassis.

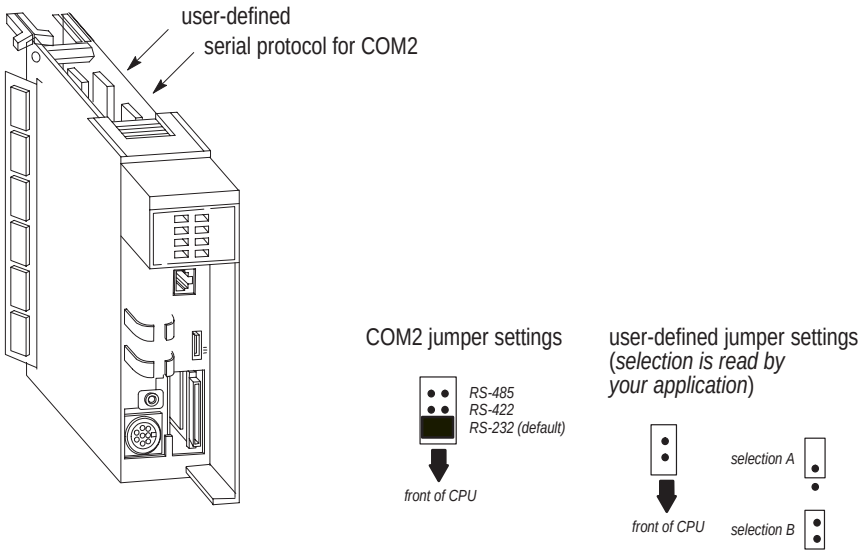
Important: Before you install the open controller CPU module:

- turn off power to the chassis
- set the jumpers on the open controller CPU module
- if needed, install the system memory and the FlashDrive

Setting Jumpers

Figure 2.1 shows how to set the jumpers on the open controller CPU module.

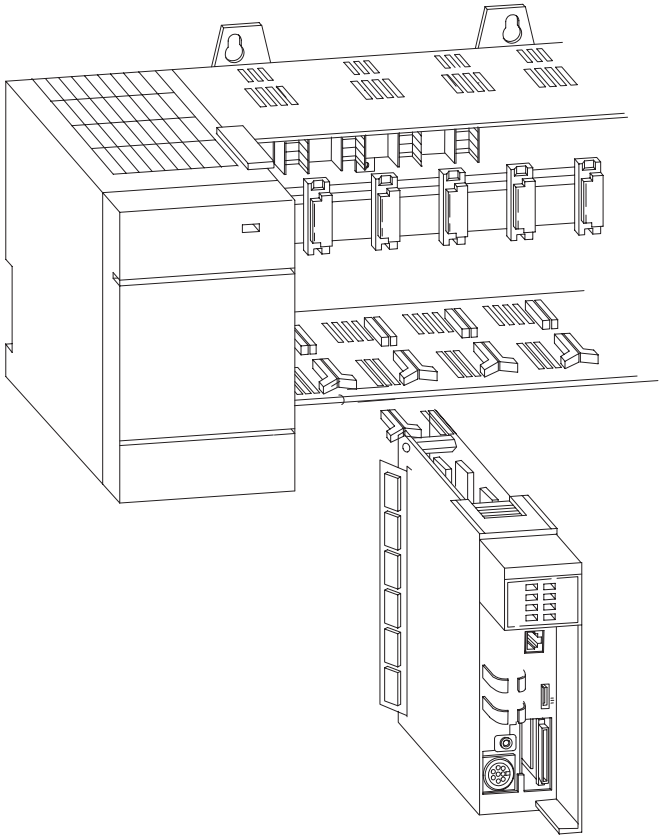
Figure 2.1
Setting open controller jumpers



Inserting the Module

Place the open controller CPU module in the far-left slot of a 1746 chassis.

If you will be using any of the open controller option modules, the PCI expansion bus must already be installed. For more information, see the 1747 Open Controller PCI Expansion Bus Installation Instructions, publication 1747-5.16.



For information about installing an open controller system, see the Open Controller System Quick Start, publication 1747-10.3.

For information about handling the battery, see appendix B.

Changing the LED mask

Important: When you receive your open controller CPU module, the LED cover is installed upside down. This lets you remove the cover and change the LED mask if you want. Once you install the LED cover upright, you cannot remove the cover without damaging it.

For information about producing your own LED mask, see page A–5.

Modifying BIOS Settings

Before You Begin

To boot an open controller or install software, you need to change BIOS settings for the open controller system. This chapter shows you how to:

- modify BIOS settings
- save a set of BIOS settings to copy to another system

This chapter also lists and describes the factory-set BIOS settings.

Important: Your open controller must have a good battery installed in order to change BIOS settings. Otherwise, the BIOS settings revert to the failsafe settings.

Preparing the Open Controller

To change the BIOS settings of the open controller, you need one of the following configurations:

- 1747-OCVGA1 module, monitor, and keyboard

  or

- remote PC, serial cable, and terminal emulation software

Remote BIOS setup

You can use a remote PC to change BIOS settings and boot the open controller. You need:

- open controller CPU module with BIOS version 1.03 or later
- remote PC
- null-modem serial cable
- terminal emulation software capable of emulating a VT100

To remotely change BIOS settings:

1. Connect the null-modem serial cable to COM2 of the open controller CPU module and to any serial port on the remote PC.

   Important: The remote setup function only works on COM2 of the open controller CPU module.

2. Run the terminal emulation software on the remote PC.

   Select the port to which the open controller is connected (COM2) and set the serial port parameters to:

   - 19200 bps
   - 8 data bits
   - 1 stop bit
   - no parity
   - software flow control (XON/XOFF)

   Configure the terminal emulation software for VT100 or VT102.

3. Activate the remote setup by holding down the space bar on the remote PC while turning on or resetting the open controller.

   Release the space bar when the BIOS sign-on message appears on the remote PC.

   If a keyboard is connected to the open controller, you can also activate the remote setup from that keyboard.

4. When prompted during the memory test, press `[Del]` on the remote PC to enter the setup.

Once you enter the setup, the keys function differently than if you entered setup from the open controller. Table 3.A describes the keys and their functions.

Table 3.A
Key functions during remote BIOS setup

This key:	Performs this function:
+	page up, instead of [Page Up]
–	page down, instead of [Page Down]
arrow keys	move between fields on the screen
[Tab] > [Ctrl–F]	move one field to the right use any of these keys if the right arrow doesn't function properly
[Backspace] < [Ctrl–D]	move one field to the left use any of these keys if the left arrow doesn't function properly
[Ctrl–R]	move one field up use this key if the up arrow doesn't function properly
[Ctrl–C]	move one field down use this key if the down arrow doesn't function properly

The [F10] key doesn't work during remote BIOS setup. Use the direction keys to select Save Settings and Exit option from the main menu.

Changing BIOS Settings

Several open controller configurations require BIOS changes in order to work properly. To modify BIOS, boot the open controller system. During the boot-up process, the monitor displays several messages. When you see the following message, press [Del]:

```
Hit <DEL> if you want to run SETUP
```

Sometimes, this message disappears too quickly to see. If this happens, reboot the open controller and press [Del] immediately, even while the screen is still blank.

After you make BIOS changes save the changes and reboot the open controller.

Navigating the menu system

To access the various options within the BIOS setup screens, use these keys:

Press this key:	To:
[Page Up] [Page Down]	change the present item to the next/previous possible value
[Enter]	select a sub-menu
[F2] [F3]	change the screen colors
arrow keys	move to another item
[Esc]	return to the previous menu

Using the main menu

The main menu provides access to the different BIOS areas:

Use this menu:	To specify or change:
Standard CMOS Setup	date/time floppy drives hard/CDROM drive sizes boot sector virus checking
Advanced CMOS Setup	bootup sequence numlock setting system keyboard primary display OS/2 compatibility mode internal cache external cache PCMCIA configuration boot channel
PCI/Plug and Play Setup	Plug & Play aware operating system CPU to PCI write buffer PCI to DRAM buffer PCI VGA palette snoop 1747-OCIDE properties IRQs reserved memory
Peripheral Setup	onboard serial ports onboard parallel port onboard IDE
Change Supervisor Password	the password that is necessary to gain access to the BIOS setup
Auto Configuration with Fail Safe Settings	BIOS settings to defaults **Important:** This function does not change any of the settings on the Standard CMOS Setup screen.
Save Settings and Exit	save the BIOS changes you made
Exit Without Saving	exit without saving the BIOS changes you made

Saving BIOS Settings

The diagnostic and utilities disk that comes with the open controller CPU module includes a `getcmos.exe` utility that saves current BIOS settings to a file. You can also use this utility to restore BIOS settings from information previously saved in a file (using this utility).

Storing BIOS settings

To save current BIOS settings to a file:
`getcmos>`*`filename`*

The `getcmos.exe` utility works properly under DOS, a DOS shell in Windows, and a DOS shell in Windows 95. The utility doesn't work under a DOS shell in Windows NT.

Restoring BIOS settings

The revision of BIOS must be the same between the open controller CPU module and the BIOS data saved to a file. To restore BIOS settings from an existing file:
`getcmos` *`filename`*

After you restore BIOS settings, reboot to make those settings active.

The real time clock will not be correct after you restore BIOS settings.

Factory-Set BIOS Settings

The following tables list the factory-set (default) BIOS settings when you receive an open controller CPU module.

Standard CMOS Setup

This field:	Description:
Date/Time	Set the time and date for the real time clock that is built into the open controller CPU module. The clock keeps track of time even when the open controller system is off. *Default* Current date and time
Floppy Drives	Use this setting to tell the open controller CPU module if any floppy drives are installed and what size they are. *Default* Not Installed

Hard Drives	There is one primary master, IDE connector built into the open controller CPU module for the FlashDrive. Most of the time the open controller can automatically determine the settings on a hard drive. If this is the case, leave the settings on Auto. If the open controller doesn't determine the size of a hard drive, consult the drive documentation and enter the appropriate settings.
	Default
	`                    LBA   BLK   PIO   32Bit`
	`                    Mode  Mode  Mode  Mode`
	`Pri Master   Auto  Off   Off   Auto  Off`
	`Pri Slave    Not Installed`
	`Sec Master   Auto  Off   Off   Auto  Off`
	`Sec Slave    Not Installed`
	Important: Set `LBA Mode` to `On` if the drive is greater the 512 Mbytes.
	To view or change drive settings, place the cursor on `Auto` and press `[Enter]`. This changes `Auto` to `User` and displays the drive settings.

Boot Sector Virus Protection	This option allows the BIOS to scan the boot sector on startup for viruses. When enabled, AMIBIOS warns the user when any program attempts to write to or format the boot sector and allows the user to intervene. *Default* Disabled

Advanced CMOS Setup

This field:	Description:
BootUp Sequence	This is the order the BIOS uses to find a bootable drive. *Default* C:, A:, CDROM Set the sequence to A:, C:, CDROM if you are using the `hostsvr` utility or are booting from a floppy drive connected to the parallel port. Set the sequence to CDROM, A:, C; if booting from a CDROM attached to an IDE interface module.
BootUp Num-Lock	Set whether the Num Lock is on or off at startup. *Default* On
System Keyboard	Use this setting to tell the BIOS whether a keyboard will be present or not. *Default* Present
Primary Display	Use this setting to tell the open controller CPU module the type of video display, if any. *Default* VGA/EGA
OS/2 Compatibility Mode	Use this setting to enable or disable compatibility with the OS/2 operating system. Set this option to Enabled to permit AMIBIOS to run with IBM OS/2. *Default* Disabled
Internal Cache	This option specifies the caching algorithm used for L1 internal cache memory. The settings are: Disabled — Neither L1 internal cache memory on the CPU or L2 secondary cache memory is enabled. WriteBack *(default)* — Use the write-back caching algorithm. WriteThru — Use the write-through caching algorithm.
External Cache	This option specifies the caching algorithm used for L2 external cache memory. The settings are: Disabled — Neither L1 internal cache memory on the CPU or L2 secondary cache memory is enabled. WriteBack *(default)* — Use the write-back caching algorithm. WriteThru — Use the write-through caching algorithm.
PCMCIA Configuration	Set how an ATA memory device in a PCMCIA interface module behaves. *Default* Disabled AMIBIOS allows access to an ATA memory device within a PCMCIA interface module without the need for card and socket services to be running on the open controller. Set PCMCIA Configuration to Secondary to enable this access. This sets the ATA memory device as a secondary master. If you want the ATA device to be the primary master, remove or disable the embedded FlashDrive and disable the onboard IDE.
Boot Channel	Use this setting to tell the BIOS which drive to try to boot off of, the primary master or the secondary master. *Default* Primary You should only set the boot channel to secondary as a temporary solution, such as using a PC Card in a PCMCIA interface module to load software onto an embedded FlashDrive. Setting the boot channel to secondary is nonstandard. DOS will boot from a secondary drive, but other operating systems, such as Windows NT, will not boot from a secondary drive.

PCI/Plug and Play Setup

This field:	Description:
Plug and Play Aware O/S	**This setting does not affect the open controller.** *Default* No
CPU to PCI Write Buffer	Enables the CPU to PCI write buffer for increased speed. *Default* Enabled
PCI to DRAM Buffer	Enabled the PCI to DRAM buffer for increased speed. *Default* Enabled
PCI VGA Palette Snoop	**This setting does not affect the open controller.** *Default* Enabled
1747-OCIDE Module 1747-OCIDE Primary Master 1747-OCIDE Secondary Master	Use this setting to tell AMIBIOS the settings of an open controller IDE interface module. *Default* Disabled *indicate the slot the IDE interface module is in* Disabled *enable if using the primary channel on an IDE interface module* Disabled *enable if using the secondary channel on an IDE interface module* For more information, see the IDE Interface Module Installation Instructions, publications 1747-5.29 and 1747-5.30.
IRQ3 through IRQ15	Set whether an IRQ is ISA/EISA or PCI/Plug and Play. *Default* IRQ3 ISA/EISA IRQ4 ISA/EISA IRQ5 PCI/PnP IRQ7 ISA/EISA IRQ9 PCI/PnP IRQ10 PCI/PnP IRQ11 ISA/EISA IRQ14 PCI/PnP IRQ15 ISA/EISA You cannot change IRQ 12. IRQ 12 is always ISA/EISA.
Reserve Memory Size and Address	Specify a memory area that will not be assigned to PCI devices. If you are using CardSoft card and socket services, this memory area should match the memory resources allocated for CardSoft to use. *Default* Size 64K Address D0000

Peripheral Setup

This field:	Description:
Onboard Serial Port1/Port2	Set the base I/O port address used by the onboard serial port 1 and serial port 2. *Default* Port 1 3F8h Port 2 2F8h
Onboard Parallel Port	Set the base I/O port address used by the onboard parallel port. This option also enables floppy-drive support via the parallel port. *Default* 378H
Parallel Port Mode	Set the parallel port mode. *Default* Normal
Parallel Port IRQ	Set the parallel port interrupt. *Default* IRQ 7
Onboard IDE	Enables or disables the internal IDE controller (internal FlashDrive). If enabled, the onboard IDE controller is always the primary IDE channel. *Default* Primary Set the `Onboard IDE` to `Disabled` if you do not have an embedded FlashDrive. Set the `1747-OCIDE Module` to `Primary` if you do have an embedded FlashDrive.

Starting the Open Controller

Before You Begin

This chapter describes how the open controller hardware powers up. This chapter also describes the open controller configurations you can use when powering-up a system.

Powering Up an Open Controller CPU Module

When you power up the open controller, the CONTROLLER and the SCANNER simultaneously perform power-on self tests (POST). The CONTROLLER goes through a boot-up process that is similar to that of a personal computer. If you have a video module, boot-up messages appear on the screen. The SCANNER indicates which test it is performing by lighting the user-defined LEDS (LED 1, LED 2, LED 3, and LED 4). Table 4.A describes the scanner POST and provides approximate times for each test.

Table 4.A
LED patterns for LED 1, LED 2, LED 3, LED 4 during POST

This LED:	Has these states (in this order)	To indicate these tests:	That take this long (seconds):
LED 1	solid red	software CRC checksum	0.5
	solid green	128K bytes RAM	1.5
	flashing red	reserved	0.5
	flashing green	dual-port RAM	0.5
LED 2	solid red	backplane	0.5
	solid green	reserved	0.5
	flashing red	reserved	0.5
	flashing green	reserved	0.5
LED 3	solid red	reserved	0.5
	solid green	temperature sensor	0.5
LED 4	solid red	interrupt controller	0.5
	solid green	timer	0.5
		Total Time	7.0

The STATUS LED indicates the success of the POST (table 4.B).

Table 4.B
LED patterns for STATUS during POST

These states are possible:	Which mean:
yellow	running POST
flashing green	passed scanner post
solid red	internal fault / POST failure

The CONTROLLER and the SCANNER perform their power-on self tests when you press the reset switch on the front panel.

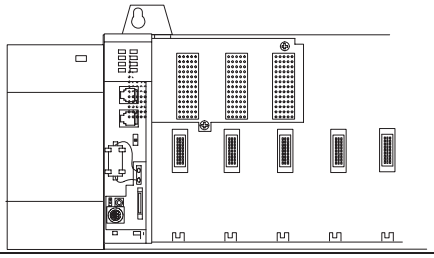
How you power up the open controller depends on what software is already installed on the native IDE drive. The native IDE drive can be one of the following:

- embedded FlashDrive (Sandisk 1.3” FlashDrive)
- 2.5” IDE hard drive installed in the 1747-OCIDE25 interface module
- IDE-compatible, ATA memory device installed in the 1747-OCIDE1 interface module

Booting when an operating system is loaded

If the native IDE drive already has an operating system installed, apply power to the open controller system (table 4.C).

Table 4.C
Configurations for booting the open controller (native IDE drive is bootable)

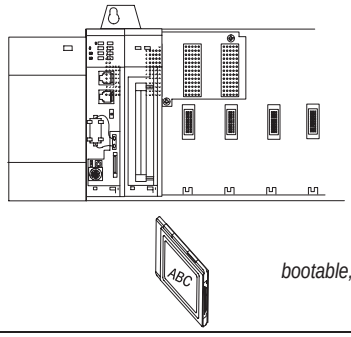
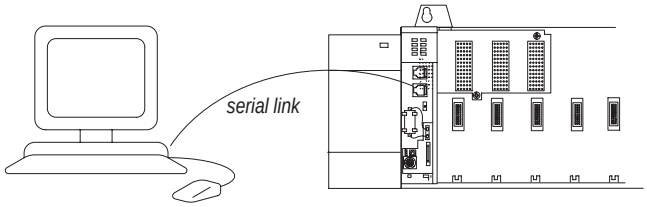
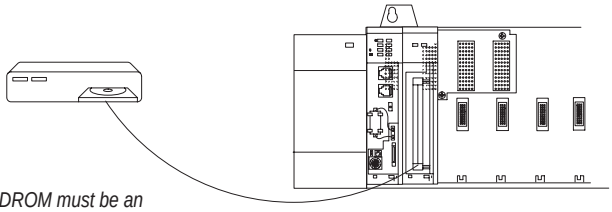
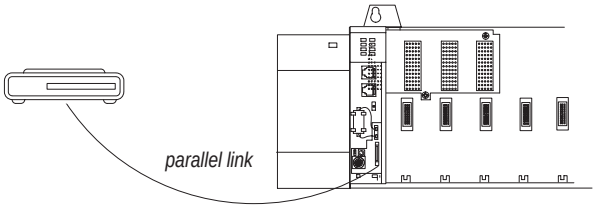
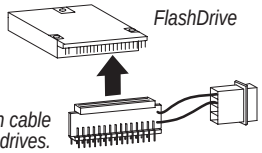
Do this:		For more information see:
Boot from an already installed, native IDE drive that is formatted and bootable	 *The operating system is already installed on the native IDE drive*	FlashDrive Installation Instructions publication 1747-5.17 IDE Interface Module Installation Instructions publications 1747-5.29 and 1747-5.30

If you have an open controller CPU module that came with an internal FlashDrive but DOS was not installed, the internal FlashDrive was only formatted and is not bootable.

Booting when an operating system is not loaded

If the native IDE drive is not bootable, use one of the following configurations (table 4.D) to boot the open controller. Once the system is up and running, either use the DOS `sys` command to make the native IDE drive DOS bootable or install a different operating system.

Table 4.D
Configurations for booting the open controller
(native IDE drive is not bootable or is not installed)

Do this:	For more information see page:
Boot from a bootable PC Card within an open controller PCMCIA interface module	4–4
 *bootable, ATA-compatible PC Card*	
Boot from a remote PC through COM2 using the `hostsvr` utility	4–6
	
Boot from an external, CDROM on an IDE interface module	4–8
 *External CDROM must be an IDE-compatible ATA device*	
Boot from an external, parallel-port floppy drive	4–9
	
Temporarily install the native IDE drive in another PC	4–10
 *Attach this end to the ribbon cable that controls the disk drives.* *Attach this end to the power source in the personal computer*	

Booting from a PC Card in a PCMCIA interface module

This method lets you transfer software from a bootable ATA memory device to a native IDE drive. The ATA memory device will be drive C: and the native IDE drive will be drive D:.

Requirements

You need a:

- 1747-OCPCM1 module
- 1747-OCVGA1 module, monitor, and keyboard

  or

  remote PC, serial cable, and terminal emulation software
- bootable, ATA-compatible PC Card

  You can make a PC Card DOS bootable by using your personal computer and running the `sys` command on the PC card. For example, if the PC card is drive D:, enter **`sys d:`**

Considerations

Considerations of this configuration include:

- This method only works for some operating systems, such as DOS or Windows 3.x. Operating systems with built-in card and socket services, such as Windows 95 and Windows NT, will not boot using this method.
- Card and socket services software (such as CardSoft) cannot be loaded because it breaks the BIOS connection to the PCMCIA interface module. This means that you cannot use the second PC Card slot in the PCMCIA interface module.

  Important: If you want to boot from an ATA PC Card and use other PC Cards at the same time, place the ATA PC Card in an 1747-OCIDE1 IDE interface module. The ATA PC Card must be IDE compatible. Then you can load card and socket services software and use other PC Cards in the PCMCIA interface module.
- The ATA memory device will not appear as an IDE drive, so functionality and performance will differ from that of native IDE drives. Only use this method temporarily to load an operating system or other software on the native IDE drive.

Steps Follow these steps:

5. Insert a bootable ATA memory device into the leftmost slot (slot 0) of the rightmost 1747-OCPCM1 interface module in the open controller chassis.

6. Change BIOS settings. Table 4.E shows the required BIOS settings – assuming that you are using an internal FlashDrive. The BIOS settings not shown don't affect booting from a PCMCIA interface module. See chapter 3 for additional BIOS information.

Table 4.E
BIOS settings for booting from a PCMCIA interface module

This BIOS area:	Should have these settings:	
Standard CMOS Setup	`Pri Master:`	`Auto`
	`Pri Slave:`	`Not Installed`
	`Sec Master:`	`Auto`
	`Sec Slave:`	`Not Installed`
Advanced CMOS Setup	`Boot Up Sequence:`	`C:, A:, CDROM`
	`PCMCIA Configuration:`	`Secondary` (disabled by default)
	`Boot Channel:`	`Secondary` (primary by default)
	The PCMCIA Configuration and the Boot Channel selection must match (both must be primary or both must be secondary). If you are using an embedded Flashdrive, you must select secondary for the PCMCIA configuration.	
Peripheral Setup	`OnBoard IDE:`	`Primary`
PCI / Plug and Play	`IRQ 15`	`ISA/EISA`
	Make sure no other device in your system uses IRQ 15	

7. Cycle power to the open controller.

The ATA memory device acts as a secondary master, which uses IRQ 15. The BIOS default for this interrupt is ISA/EISA. For information about defining interrupts, see chapter 5.

Booting from a remote PC through COM2

This boot method makes a remote PC appear as a floppy drive on the open controller. This enables the open controller to boot from a bootable floppy in the remote PC floppy drive, after which you can transfer an operating system to the native IDE drive.

Requirements

You need a:

- 1747-OCVGA1 module, monitor, and keyboard

  You use the monitor and keyboard connected to the open controller, not the remote PC.
- remote PC (laptop or desktop)
- remote boot-up utility (`hostsvr.exe`) on the remote PC (this utility is on the diagnostic/utility disk that comes with the open controller)
- 9-pin null-modem serial cable (such as catalog number 1747-OCSBC)
- bootable floppy disk in drive A: of the remote PC (use `format` *`n`*`: /s` to make a floppy disk DOS bootable)

Considerations

Considerations of this configuration include:

- Some operating systems, such as QNX, can't be loaded from a remote PC in this manner because their built-in floppy drivers break the BIOS connection.
- This method allows access only to the remote PC floppy drive, not the hard drive.
- The keyboard of the remote PC is inactive. Using the keyboard of the remote PC, breaks the host/server connection.

Steps

Follow these steps:

1. Connect a serial port of the remote PC to COM2 on the open controller. You can use the 1747-OCSBC serial cable.
2. Place a DOS bootable floppy disk in drive A: of the remote PC.

3. Run `hostsvr /com`*n* on the remote PC, where *n* is the COM port on the remote PC that is connected to the open controller.

   If the remote PC is running Windows 95 or Windows NT, you must reboot to MS-DOS mode to use `hostsvr`. This utility won't run reliably in a DOS shell.

   This utility displays an information screen on the remote PC. Once you escape from the display, the remote connection ends.

   ATTENTION: Don't use the keyboard of the remote PC to run any applications because that would disrupt the remote PC connection.

4. Make sure COM2 on the open controller is configured for RS-232. For information about configuring COM2, see page 7–3.

5. Change BIOS settings on the open controller. Table 4.F shows the required BIOS settings. The BIOS settings not shown don't affect booting from a remote PC. See chapter 3 for additional BIOS information.

Table 4.F
BIOS settings for booting from a remote PC

This BIOS area:	Should have these settings:	
Advanced CMOS Setup	`BootUp Sequence`	`A:, C:, CDROM`

6. Cycle power to the open controller.

   During system boot, the open controller monitor displays this message underneath the AMIBIOS system configuration screen:
   `COM2: Emulating Drive A:`

7. Drive A: on the remote PC appears as drive A: to the open controller. You can use drive A: on the remote PC to make the native IDE drive bootable or to install an operating system or other software on the native IDE drive (see page 6–3).

   ATTENTION: The `hostsvr` utility makes the remote PC emulate a floppy drive. You only have access to drive A: on the remote PC. Don't use the monitor, keyboard, or other drives.

   If there is not a bootable floppy disk in drive A: of the remote PC, the open controller will boot from the C: drive of the open controller, as long as the C: drive is bootable. If the open controller boots from the C: drive, you can still access the A: drive of the remote PC, as long as `hostsvr` is running on the remote PC.

Booting from a external CDROM on an IDE interface module

This configuration lets you boot the open controller from a CDROM device in a 1747-OCIDE1 or 1747-OCIDE25 interface module.

Requirements

You need a:

- 1747-OCIDE1 or 1747-OCIDE25 IDE interface module
- 1747-OCVGA1 module, monitor, and keyboard

  or

  remote PC, serial cable, and terminal emulation software
- IDE-compatible CDROM device connected to an IDE interface module

Important: Each IDE interface module supports as many as two external 3.5” devices – to be used for development purposes only.

Considerations

Considerations of this configuration include:

- It is recommended that you configure the CDROM device as a secondary master. You can configure the CDROM device as a primary slave as long as there is a primary master on the same IDE interface module.
- The CDROM device will appear as a IDE drive.

Important: The specifications of the CDROM device require that you derate the overall open controller system specifications to that of the CDROM device.

Steps

Follow these steps:

1. Make sure the jumpers on the IDE interface modules are set correctly. For more information, see the IDE Interface Module Installation Instructions, publications 1747-5.29 and 1747-5.30.
2. Change BIOS. Table 4.G shows the required BIOS settings. The BIOS settings not shown don’t affect booting from an IDE interface module. See chapter 3 for additional BIOS information.

Table 4.G
BIOS settings for booting from a CDROM in a 1747-OCIDE1 interface module

This BIOS area:	Should have these settings:
Advanced CMOS Setup	`Boot Sequence:` `CDROM, A:, C:` `Boot Channel:` `Primary`
PCI / Plug and Play	`1747-OCIDE Secondary Master` To enable the secondary channel: `Enabled` It is recommended that you configure the CDROM as a secondary master. You can configure the CDROM as a primary slave as long as there is a primary master on the same IDE interface module.

Booting from an external, parallel-port floppy

This method is useful for providing external, floppy-drive access to the open controller.

Requirements

You need a:

- floppy drive and appropriate cable to connect to the open controller parallel port (see “Considerations” below)
- 1747-OCVGA1 module, monitor, and keyboard

Considerations

Considerations of this configuration include:

- The recommended cable and floppy drive are bundled under catalog number 1747AV-OCFD, which is available from:
  Automation Value, LLC
  423-609-0222
  sales@avllc.com
- The cable is wired for floppy A only.

Steps Follow these steps:

1. Connect the external floppy drive to the parallel port of the open controller.
2. Change BIOS. Table 4.H shows the required BIOS settings assuming that you are using the cable and floppy drive from Automation Value. The BIOS settings not shown don't affect booting from an IDE interface module. See chapter 3 for additional BIOS information.

Table 4.H
BIOS settings for booting from an external, parallel port floppy

This BIOS area:	Should have these settings:	
Standard CMOS Setup	`Floppy Drive A:`	`1.44 MB 3 1/2`
	`Boot Channel:`	`Primary`
Advanced CMOS Setup	`Boot Sequence:`	`A:, C:, CDROM`
Peripheral Setup	`OnBoard Parallel Port:`	`Floppy`

Temporarily installing the native IDE drive in another PC

Temporarily installing the native IDE drive in another PC, as a D: drive for example, makes it relatively easy to download an operating system to the native IDE drive. Then you can re-install the native IDE drive in the open controller. The following requirements and steps explain how to install a FlashDrive in another PC.

Requirements You need a:

- PC into which to install the FlashDrive
- 3.5" to 2.5" FlashDrive adapter cable (comes with 1747-OCSDCK)
- 2mm jumper for the FlashDrive (comes with 1747-OCSDCK)

Steps Follow these steps:

1. If the embedded FlashDrive needs to be set as a slave drive, attach the 2mm jumper.
2. Attach the 2.5" to 3.5" adapter cable to the FlashDrive.
3. Attach the 2.5" to 3.5" adapter cable to an IDE ribbon cable in the other PC.
4. Attach the power connector from the 2.5" to 3.5" adapter cable to a power source in the other PC.

For more information, see the 1747 Open Controller FlashDrive Installation Instructions, publication 1747-5.17.

Defining System Settings

Before You Begin

This chapter shows you how to:

- define system interrupts
- define drive types
- set up memory managers

This chapter also lists some operating considerations and the default `autoexec.bat` and `config.sys` files.

Defining System Interrupts

The open controller and its option modules use a combination of PCI/PnP (plug-and-play) interrupts and ISA/EISA interrupts. You define interrupts as either PCI/PnP or ISA/EISA on the PCI / Plug and Play setup screen in AMIBIOS. This lets you set interrupt types to match functions.

- You generally set ISA/EISA interrupts through jumpers or in software.
- PCI/PnP interrupts are assigned by AMIBIOS when the open controller boots up. AMIBIOS begins allocating PCI/PnP interrupts with the highest-numbered IRQ defined as PCI/PnP in BIOS.
- ISA/EISA interrupts cannot be shared.
- PCI interrupts can be shared by multiple PCI devices. AMIBIOS tries to allocate one IRQ for each PCI/PnP device that requires an interrupt. If there are not enough IRQs defined as PCI/PnP, BIOS assigns the same IRQ to more than one device.
- ISA/EISA interrupts are edge-level high true interrupts.
- PCI/PnP interrupts are low true interrupts.

How ISA/EISA interrupts and PCI/PnP interrupts interact

PCI interrupts are automatically chosen from those defined as PCI/PnP (IRQ 5, 9, and 10 by default). IRQ 14 is always used by the primary IDE channel and is never assigned to a PCI/PnP module.

Within an open controller system:

- PC Cards in a PCMCIA interface module always use ISA/EISA interrupts.
- The PCI interrupt assigned to the PCMCIA interface module is not used by the module, so another device can share that interrupt.
- The 1747-OCKTX, -OCKTXD channels can use ISA/EISA interrupts or PCI/PnP interrupts, depending on how you set the jumpers on the module.

The PCI resources are assigned independently of the operating system being run. The PCI resources only change if you modify the hardware in the open controller system.

The open controller comes with a diagnostic/utility disk that includes an `ocpci` utility (`ocpci.exe` for DOS and `ocpci_nt.exe` for Windows NT). The `ocpci` utility reports the currently assigned memory addresses and PCI interrupts for all the open controller option modules.

Table 5.A shows the default IRQ assignments:

Table 5.A
Default open controller system IRQ assignments

Interrupts:	Default Type:	Typical Assignments:
IRQ 3	ISA/EISA	serial port 2
IRQ 4	ISA/EISA	serial port 1
IRQ 5	PCI/PnP	**available**
IRQ 7	ISA/EISA	parallel port 1
IRQ 9	PCI/PnP	**available**
IRQ 10	PCI/PnP	
IRQ 11	ISA/EISA	recommended for use by the open controller SCANNER and should be defined as ISA/EISA
IRQ 12	ISA/EISA	**available** always available as ISA/EISA; cannot be set to PCI/PnP
IRQ 14	PCI/PnP	primary drives (FlashDrive)
IRQ 15	ISA/EISA	secondary drives

IRQ 3, 4, and 7 are used by the internal COM and LPT ports. These IRQs can be used as PCI/PnP interrupts if you first disable the corresponding port.

If you use a jumper to set a 1747-OCKTX, -OCKTXD **channel** to a specific interrupt instead of using the PCI interrupt assigned to the 1747-OCKTX, -OCKTXD **module**, make sure the interrupt you assigned to the **channel** is configured as ISA/EISA on the AMIBIOS screen.

If you are using a PC Card in a PCMCIA interface module, make sure the interrupt for the card is set as ISA/EISA on the AMIBIOS screen.

IRQ 12 is not included on the AMIBIOS screen. It is permanently defined as an ISA interrupt.

Defining the primary IRQ (IRQ 14)

The primary master and primary slave drives share IRQ 14. Leave this interrupt defined as PCI/PnP. No open controller option module is ever assigned this interrupt.

If your open controller has an embedded FlashDrive, it always acts as the primary master. The FlashDrive uses IRQ 14. Other possible primary drives are:

- ATA memory devices in a PCMCIA interface module
- devices connected to an IDE interface module

Defining the secondary IRQ (IRQ 15)

The secondary master and secondary slave drives share IRQ 15. Leave this interrupt defined as ISA/EISA if you are using an ATA drive in a PCMCIA interface module. Change this interrupt to PCI/PnP if you are using an IDE interface module. No open controller option module is ever assigned this interrupt, even if you change the interrupt to PCI/PnP in BIOS.

Possible secondary drives are:

- ATA memory devices in a PCMCIA interface module
- devices connected to an IDE interface module

If your open controller system does not have a secondary drive, you can use IRQ 15 for another device, such as an Ethernet card or a 1747-OCKTX, -OCKTXD channel.

Defining the interrupts for PC Cards in PCMCIA interface modules

All PC Cards that use interrupts, such as Ethernet cards, use ISA interrupts. Make sure you define the interrupts as ISA/EISA on the AMIBIOS screen.

AMIBIOS assigns a PCI interrupt to the PCMCIA interface module, but the module never uses that interrupt.

Defining interrupts for a 1747-OCKTX, -OCKTXD module

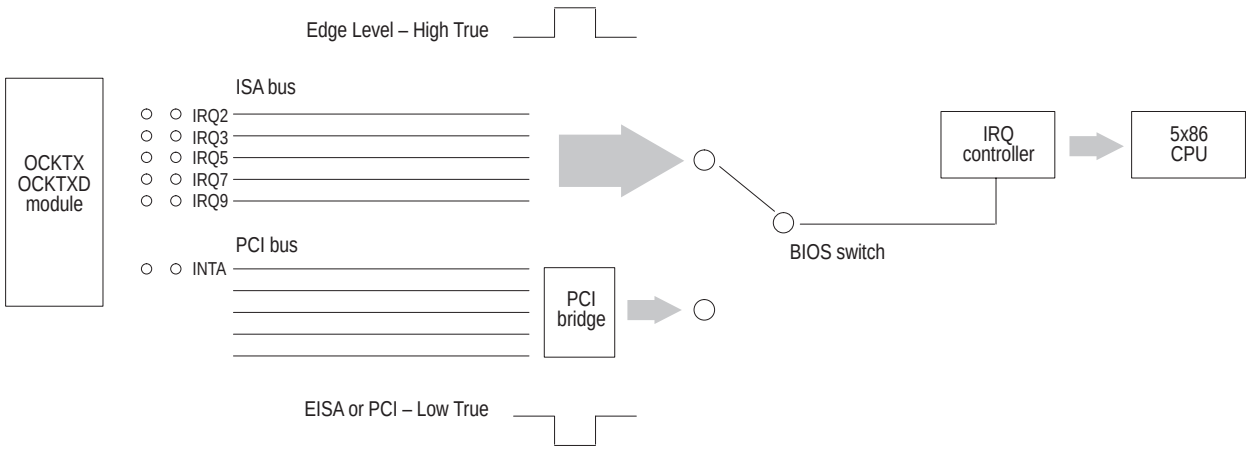
Each channel on a 1747-OCKTX, -OCKTXD module can have an ISA or PCI interrupt, based on the jumper settings you make (table 5.B). If you set the jumpers for ISA interrupts, make sure the IRQs selected by the jumpers are defined as ISA/EISA in AMIBIOS. If you use ISA interrupts, each channel must have a unique interrupt. Each ISA interrupt is a unique interrupt that cannot be used by any other device.

ISA interrupts are edge-level triggered. A high true signal indicates an interrupt. Multiple devices cannot share an ISA interrupt because another module on the bus cannot drive the signal high when another module on the bus is pulling the signal low.

PCI interrupts are low true. The signal is floating high so multiple devices can pull the signal low, which indicates an interrupt.

The 1747-OCKTX, -OCKTXD module can generate either kind of interrupt, based on the jumper settings you make. The BIOS configuration screen for interrupts determines the types of interrupts and whether an interrupt should pass through the PCI bridge device to convert it to the correct signal (figure 5.1).

Figure 5.1
Triggering ISA/EISA and PCI/PnP interrupts



If you set a 1747-OCKTX, -OCKTXD jumper for INTA, the channel uses the PCI interrupt assigned to the module. Make sure there is at least one PCI/PnP interrupt, besides IRQ 14 or IRQ 15, defined in AMIBIOS. IRQ 14 and IRQ 15 cannot be assigned to an open controller option module.

Table 5.B
Assigning PCI and ISA interrupts for an OCKTX module

If you set the jumpers so that:	Results:
both channels have a unique ISA interrupt	Each channel has a unique ISA interrupt. Make sure the IRQs selected by the jumpers are defined as ISA/EISA in BIOS BIOS also assigns the module a PCI interrupt – the module does not use this interrupt but it is reported by the `ocpci` utility.
one channel has an ISA interrupt other channel has INTA selected	Each channel has a unique interrupt (one ISA, one PCI). Make sure the ISA IRQ selected by the jumper is defined as ISA/EISA in BIOS. Make sure there is at least one PCI/PnP interrupt defined, besides IRQ 14 or IRQ 15. Use the `ocpci` utility to determine which PCI interrupt was assigned to the channel set for INTA.
both channels have INTA selected	BIOS assigns the same PCI interrupt to both channels. Make sure there is at least one PCI/PnP interrupt defined, besides IRQ 14 or IRQ 15. Both channels can share the same PCI interrupt as long as your application supports shared interrupts. Use the `ocpci` utility to determine which PCI interrupt was assigned to the channels.

Since the PCMCIA interface module does not use its PCI interrupt, it can share that interrupt with a 1747-OCKTX, -OCKTXD module. Two 1747-OCKTX, -OCKTXD channels can share a PCI interrupt only if the drive software supports shared interrupts.

For more information, see the A-B Communication Interface Module User Manual, publication 1747-6.18.

Defining interrupts for the open controller SCANNER

The open controller API and the open controller diagnostic software use IRQ 11 by default. Keep IRQ 11 defined as ISA/EISA so that it does not get assigned to open controller option modules. Also, make sure that no other devices, such as Ethernet cards or 1747-OCKTX, -OCKTXD channels, are assigned to IRQ 11.

Defining Drive Types

Corresponding master and slave devices must be on the same open controller module. For example, if the FlashDrive is a primary master, you cannot have a primary slave on an IDE interface module. Likewise, if a PCMCIA interface module is a secondary master, you cannot have a secondary slave on an IDE interface module.

You cannot have a slave drive without a master drive. For example, you cannot have a CDROM connected to an IDE interface module secondary channel and set it up as a slave (via jumpers on the CDROM) unless the the same IDE interface module already has a secondary master.

Using primary drives

The primary master and primary slave drives share IRQ 14. Leave this interrupt defined as PCI/PnP. No open controller option module is ever assigned this interrupt.

Important: Some operating systems, such as Windows NT, only boot from a primary master drive.

Using secondary drives

The secondary master and secondary slave drives share IRQ 15. Leave this interrupt defined as ISA/EISA if you are using an ATA drive in a PCMCIA interface module. Change this interrupt to PCI/PnP if you are using an IDE interface module. No open controller option module is ever assigned this interrupt.

Using 3.5” IDE devices

When you connect devices to the 3.5” connector on the IDE interface module, you select master/slave by jumpers on the external device. You select primary/secondary by which connector you connect to on the IDE interface module.

If you only have one IDE interface module with an external CDROM device, you will get better performance if you configure the IDE module as a primary master and the external CDROM as a secondary master, rather than have both share the same primary or secondary channel.

Using Memory Managers

If you are using a memory manager, you must exclude from the memory manager the memory:

- used by the I/O scanner dual-port memory, as defined by the open controller API software or as reported by the open controller diagnostic utility (the recommended memory is C800–C9FF).
- assigned to the 1747-OCKTX, -OCKTXD module, if you are using one, as reported by the `ocpci` or `ocpci_nt` utilities (see chapter 8).
- allocated in the `\cardsoft\config` utility (the default is D000–DFFF) if you are using CardSoft services. See the CardSoft documentation that comes on disk with 1747-OCPCM2.

Operating System Considerations

The following table lists system considerations for some of the operating systems you can install on an open controller.

Table 5.C
Operating system considerations

For this operating system:	Considerations:
DOS and Windows 3.x	The SystemSoft™ CardSoft™ card and socket services for DOS/Windows 3.x, version 3.1, limit an open controller system under DOS or Windows 3.x to two 1747-OCPCM1 modules (4 slots). SystemSoft CardSoft card and socket services come with the 1747-OCPCM2 kit. For more information about configuring and using these services, see the documentation that comes in the 1747-OCPCM2 kit.
Windows 95	You can install as many as two PCMCIA modules (four slots) per open controller chassis. If you use 2 PCMCIA interface modules, define IRQ 3–11 as ISA/EISA. Leave INT 14 defined as a PCI/PnP interrupt (for the primary master drive). If you use 2 PCMCIA interface modules and you have a 1747-OCKTX, -OCKTXD module in this same open controller chassis, all the channel jumpers must be set for ISA/EISA interrupts.
Windows NT	Windows NT limits itself to one 1747-OCPCM1 module. Not including IRQ 14 and IRQ 15, you must define one PCI/PnP interrupt for each 1747-OCKTX, -OCKTXD interface module and for each PCMCIA interface module. If you want to use IRQ 3, IRQ 4, or IRQ 7 for 1747-OCKTX, -OCKTXD channels or PC Cards, you must disable the COM port within Windows NT. Windows NT 4.0 does not support Plug and Play (PnP), so it has no way to detect that BIOS assigned an IRQ to a PCI/PnP device. Take care to reserve in BIOS any IRQ that is used by an ISA/EISA device (such as the open controller scanner, an Ethernet communication card, etc.) so that the BIOS does not assign the IRQ to a PCI device.

Pre-Installed DOS Files

An open controller CPU module with a factory-installed FlashDrive that has MS-DOS installed has these MS-DOS, version 6.22, files (table 5.D):

Table 5.D
DOS files on pre-installed FlashDrive

These files:	With these file names:

boot files	`command.com`
	`io.sys`
	`msdos.sys`

system files	`interlnk.exe`
	`intersvr.exe`
	`edit.com`
	`qbasic.exe`
	`format.com`
	`fdisk.exe`
	`xcopy.exe`
	`edit.hlp`
	`sys.com`

If you want all the DOS files, use `hostsvr` (or any other method described in chapter 6) to re-install DOS using the DOS disks that came with the open controller CPU module.

Default `autoexec.bat` and `config.sys` files

An open controller CPU module with a factory-installed FlashDrive that has MS-DOS installed has these `autoexec.bat` and `config.sys` files:

`autoexec.bat`
```
prompt $P$G
path c:\; c:\dos
```

`config.sys`
```
buffers = 20
files = 20
device = c:\dos\interlnk.exe
```

Installing Software

Before You Begin

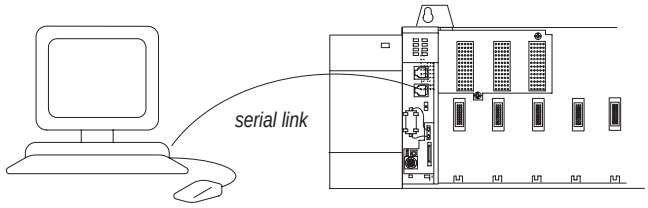
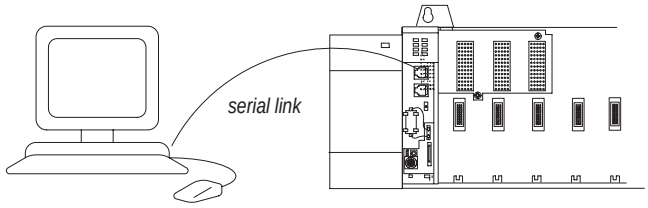
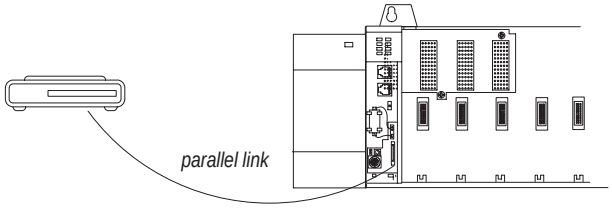
This chapter shows you how to:

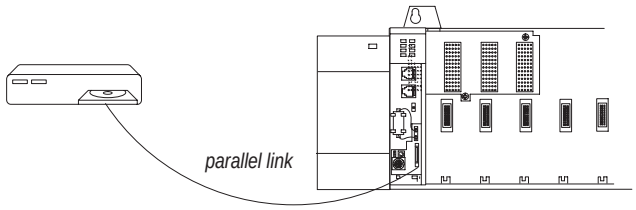
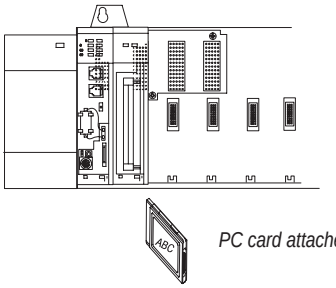
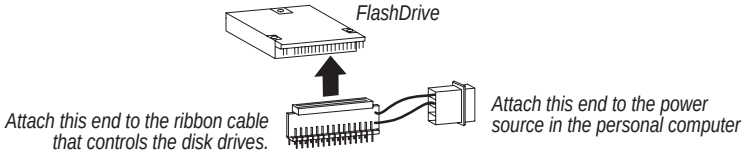
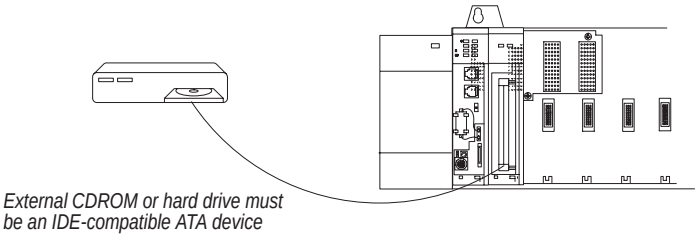
- install software
- install operating systems

Installing Software onto a Native IDE Drive

How you install software onto a native IDE drive, such as programming tools or applications, depends on the hardware in your open controller system. Once the open controller system boots up, you can use the following methods (table 6.A) to install software onto the native IDE drive.

Table 6.A
Installing software onto a native IDE drive

To do this:	See:
Install from a remote PC through COM2  *Run `hostsvr` on the remote PC* *The open controller will only have access to drive A: on the remote PC*	page 6–3
Install from a host computer using DOS `interlnk` on COM1 or COM2  *Run `intersvr` on the remote PC* *The `interlnk` device statement must be in `config.sys` of the open controller*	page 6–5
Install from an external, parallel-port floppy drive  	page 6–6

To do this:	See:
Install from an external, parallel-port SCSI device 	page 6–7
Install from a PC card connected to a CDROM or ATA memory device 	page 6–7
Temporarily install the native IDE drive in another PC 	page 6–8 FlashDrive Installation Instructions publication 1747-5.17
Install from an external, IDE-compatible ATA device on an IDE interface module 	page 6–9 IDE Interface Module Installation Instructions publication 1747-5.29 and 1747-5.30

Installing software from a remote PC through COM2

This boot method makes a remote PC appear as a floppy drive on the open controller. This enables the open controller to boot from a bootable floppy in the remote PC floppy drive, after which you can transfer an operating system to the native IDE drive.

Requirements

You need a:

- 1747-OCVGA1 module, monitor, and keyboard

  You use the monitor and keyboard connected to the open controller, not the remote PC.
- remote PC (laptop or desktop)
- remote boot-up utility (`hostsvr.exe`) on the remote PC (this utility is on the diagnostic/utility disk that comes with the open controller)
- 9-pin null-modem serial cable (such as catalog number 1747-OCSBC)
- bootable floppy disk in drive A: of the remote PC (use `format` *`n`*`: /s` to make a floppy disk DOS bootable)

Considerations

Considerations of this configuration include:

- Some operating systems, such as QNX, can't be loaded from a remote PC in this manner because their built-in floppy drivers break the BIOS connection.
- This method allows access only to the remote PC floppy drive, not the hard drive.
- The keyboard of the remote PC is inactive. Using the keyboard of the remote PC, breaks the host/server connection.

Steps

Follow these steps:

1. Connect a serial port of the remote PC to COM2 on the open controller. You can use the 1747-OCSBC serial cable.
2. Place a DOS bootable floppy disk in drive A: of the remote PC.

3. Run `hostsvr /com`*n* on the remote PC, where *n* is the COM port on the remote PC that is connected to the open controller.

   If the remote PC is running Windows 95 or Windows NT, you must reboot to MS-DOS mode to use `hostsvr`. This utility won't run reliably in a DOS shell.

   This utility displays an information screen on the remote PC. Once you escape from the display, the remote connection ends.

   ATTENTION: Don't use the keyboard of the remote PC to run any applications because that would disrupt the remote PC connection.

4. Make sure COM2 on the open controller is configured for RS-232. For information about configuring COM2, see page 7–3.

5. Change BIOS settings on the open controller. Table 6.B shows the required BIOS settings. The BIOS settings not shown don't affect booting from a remote PC. See chapter 3 for additional BIOS information.

Table 6.B
BIOS settings for installing software from a remote PC

This BIOS area:	Should have these settings:	
Advanced CMOS Setup	`BootUp Sequence`	`A:, C:, CDROM`

6. Cycle power to the open controller.

   During system boot, the open controller monitor displays this message underneath the AMIBIOS system configuration screen:
   `COM2: Emulating Drive A:`

7. Drive A: on the remote PC appears as drive A: to the open controller. You can use drive A: on the remote PC to make the native IDE drive bootable or to install an operating system or other software on the internal FlashDrive (see page 6–3).

   ATTENTION: The `hostsvr` utility makes the remote PC emulate a floppy drive. You only have access to drive A: on the remote PC. Don't use the monitor, keyboard, or other drives.

   If there is not a bootable floppy disk in drive A: of the remote PC, the open controller will boot from the C: drive of the open controller, as long as the C: drive is bootable. If the open controller boots from the C: drive, you can still access the A: drive of the remote PC, as long as `hostsvr` is running on the remote PC.

Once you boot the open controller through this configuration, drive A: on the remote PC appears as drive A: on the open controller. You can use drive A: on the remote PC to install software on the open controller. The open controller doesn't have access to any other drives on the remote PC. Don't use the keyboard of the remote PC to run any applications because that would disrupt the remote PC connection.

Installing software through DOS `interlnk`

This method lets you transfer files from a remote PC to the open controller. Use DOS `interlnk` to make the drives of the remote PC appear as extra drives to the open controller. Connect the remote PC with a serial cable (9-pin null-modem serial cable) to COM1 or COM2 of the open controller. You can use the serial boot cable, catalog number 1747-OCSBC.

Requirements

You need a:

- 1747-OCVGA1 module, monitor, and keyboard
- remote PC or laptop
- 9-pin null-modem serial cable
- DOS installed
- `interlnk.exe` installed on both systems

Steps

Follow these steps:

1. Add the statement `device=c:\dos\interlnk.exe` to the `config.sys` file for the open controller and the remote PC.
2. Type **`intersvr`** at the DOS prompt of the remote PC (server). The monitor on the remote PC displays information about the redirected drives. For more information about `interlnk`, see your DOS documentation.

   If you run `intersvr` on the remote PC after the open controller boots, run `interlnk` on the open controller to initiate the communication.

Installing software from an external, parallel-port floppy

This method is useful for providing external, floppy-drive access to the open controller.

Requirements

You need a:

- floppy drive and appropriate cable to connect to the open controller parallel port (see "Considerations" below)
- 1747-OCVGA1 module, monitor, and keyboard

Considerations

Considerations of this configuration include:

- The recommended cable and floppy drive are bundled under catalog number 1747AV-OCFD, which is available from:
  Automation Value, LLC
  423-609-0222
  sales@avllc.com
- The cable is wired for floppy A only.

Steps

Follow these steps:

1. Connect the external floppy drive to the parallel port of the open controller.
2. Change BIOS. Table 4.H shows the required BIOS settings assuming that you are using the cable and floppy drive from Automation Value. The BIOS settings not shown don't affect booting from an IDE interface module. See chapter 3 for additional BIOS information.

Table 6.C
BIOS settings for booting from an external, parallel port floppy

This BIOS area:	Should have these settings:	
Standard CMOS Setup	`Floppy Drive A:`	`1.44 MB 3 1/2`
	`Boot Channel:`	`Primary`
Peripheral Setup	`OnBoard Parallel Port:`	`Floppy`

Installing software from an external, parallel port SCSI device

This method is useful for providing external, floppy-drive or CDROM access to the open controller. You cannot use this method to boot the open controller.

Requirements

You need a:

- external, parallel-port SCSI device (should come with appropriate cable and device driver)
- 1747-OCVGA1 module, monitor, and keyboard

Steps

Follow these steps:

1. Connect the external SCSI device to the parallel port of the open controller (the cable comes with the device)
2. Install a driver (that comes with the device) for the device onto the native IDE drive
3. Add a device statement to the open controller `config.sys` file to allow the open controller to recognize the device

For more information, see the documentation that comes with the SCSI device.

Installing software from a PCMCIA CDROM or ATA memory device

This method is useful for installing most software if you have a PCMCIA interface module.

Requirements

You need a:

- 1747-OCPCM1, -OCPCM2 PCMCIA interface module
- PCMCIA CDROM or ATA memory device
- 1747-OCVGA1 module, monitor, and keyboard

Steps

Follow these steps:

1. Install and configure card and socket services
2. Install and connect the PCMCIA CDROM or ATA memory device
3. Install any necessary drivers (that come with the device)

You can load software from the ATA memory device without having card and socket services loaded by changing the BIOS setting for `PCMCIA Configuration` under `Advanced CMOS Setup` to `Secondary`. See chapter 3 for more information.

Temporarily installing the native IDE drive in another PC

Transporting the native IDE drive to another PC, as a D: drive for example, makes it relatively easy to download an operating system to the native IDE drive. Then you can re-install the native IDE drive in the open controller. The following requirements and steps explain how to install a FlashDrive in another PC.

Requirements

You need a:

- PC into which to install the FlashDrive
- 3.5” to 2.5” FlashDrive adapter cable (comes with 1747-OCSDCK)
- 2mm jumper for the FlashDrive (comes with 1747-OCSDCK)

Steps

Follow these steps:

1. If the embedded FlashDrive needs to be set as a slave drive, attach the 2mm jumper.
2. Attach the 2.5” to 3.5” adapter cable to the FlashDrive.
3. Attach the 2.5” to 3.5” adapter cable to an IDE ribbon cable in the other PC.
4. Attach the power connector from the 2.5” to 3.5” adapter cable to a power source in the other PC.

For more information, see the 1747 Open Controller FlashDrive Installation Instructions, publication 1747-5.17.

Installing software from an external, IDE-compatible device

This method is useful for providing external access to the open controller through an IDE interface module.

Requirements

You need a:

- 1747-OCIDE1 or 1747-OCIDE25 IDE interface module
- 1747-OCVGA1 module, monitor, and keyboard

  or

  remote PC, serial cable, and terminal emulation software
- IDE-compatible CDROM device connected to an IDE interface module

Important: Each IDE interface module supports as many as two external 3.5” devices – to be used for development purposes only.

Steps

Follow these steps:

- Connect the external CDROM or hard drive to one of the external 3.5” connectors on the IDE interface module
- Install the drive according to the instructions that come with the drive
- Provide power to the external device

Important: The external 3.5” connectors are for development purposes only.

You cannot have a slave drive without a master drive. For example, you cannot have a CDROM connected to an IDE interface module secondary channel and set it up as a slave (via jumpers on the CDROM) unless the open controller system already has a secondary master.

When you connect devices to the 3.5” connector on the IDE interface module, you select master/slave by jumpers on the external device. You select primary/secondary by which connector you connect to on the IDE interface module.

Installing Operating Systems

How you install an operating system depends on the hardware in your open controller system. Table 6.D provides some guidelines.

Table 6.D
How to install operating systems

Operating system:	Considerations:
DOS	You can install DOS on the FlashDrive, on a PC Card in a PCMCIA interface module, or on a drive in an IDE interface module.
Windows 3.x	You can install Windows 3.x on the FlashDrive, on a PC Card in a PCMCIA interface module, or on a drive in an IDE interface module.
Windows 95	You can install Windows 95 on the FlashDrive or on an IDE-compatible device installed in an IDE interface module. You can't run or install Windows 95 from a PC Card in a PCMCIA interface module because the operating system requires full access to the backplane and can't work around the resources that are assigned to the PCMCIA module.
Windows NT	You can install Windows NT on an IDE-compatible device installed in an IDE interface module (for more information, see the section following this table). You can't run or install Windows NT from a PC Card in a PCMCIA interface module because the operating system requires full access to the backplane and can't work around the resources that are assigned to the PCMCIA module.

Installing Windows NT

Before you install Windows NT, make sure you have:

- IDE drive (flash or rotating) greater than 180 Mbytes
- 1747-OCIDE1 or 1747-OCIDE25 IDE interface module
- external CDROM drive (through parallel port) in DOS
  or
  external IDE CDROM drive directly connected to the external connector on the IDE interface module
  or
  notebook computer with embedded CDROM drive

Without a floppy drive

See "Installing software from a parallel port floppy" on page 6–6 for information about installing a floppy drive that Windows NT recognizes.

With a floppy drive

Important: First install DOS 6.22. Then install Windows NT in FAT32 format. Don't convert Windows NT to NTFS until after you finish configuring your open controller system. This lets you transfer device drivers or other files to the hard drive using the DOS `interlnk` or `hostsvr` utilities.

If your Windows NT drive is configured as NTFS, you can still boot using `hostsvr` and run the open controller diagnostics utilities, but you cannot transfer files to the hard drive.

Installing Windows NT from an external CDROM drive

1. Make sure the CDROM drive is configured and the proper DOS drivers are loaded and running.
2. Run the WINNT utility (which bypasses the 3 setup disks for installing Windows NT):
   ```
   winnt /B /S:d:\i386
   ```
`/B`	loading without floppy disks
`/S`	alternate path to CDROM
`d:`	CDROM drive
3. Follow the installation instructions.

The advantage of this method is that the load time is fairly fast. The disadvantage of this method is that is can take a while to get the CDROM drivers configured and running so that the CDROM appears as an extra drive in DOS.

If you have room on the hard drive, consider copying the entire `\i386` subdirectory to the hard drive (`xcopy /s`). This makes the available drivers more accessible if you ever change your Windows NT setup.

Installing Windows NT from a notebook CDROM drive

1. Place your PCMCIA hard drive in a PCMCIA slot on a notebook computer that has an accessible CDROM drive.
2. Copy the entire `\i386` subdirectory to the PCMCIA hard drive.
3. Place the PCMCIA hard drive in the IDE interface module.
4. Run the WINNT utility (which bypasses the 3 setup disk for installing Windows NT):
   `winnt /B /S:c:\i386`

`/B`	loading without floppy disks
`/S`	alternate path to CDROM
`c:`	PCMCIA drive

5. Follow the installation instructions.

Installing from the `\i386` directory on an ATA drive takes much longer than installing directly from a CD.

Using the Open Controller

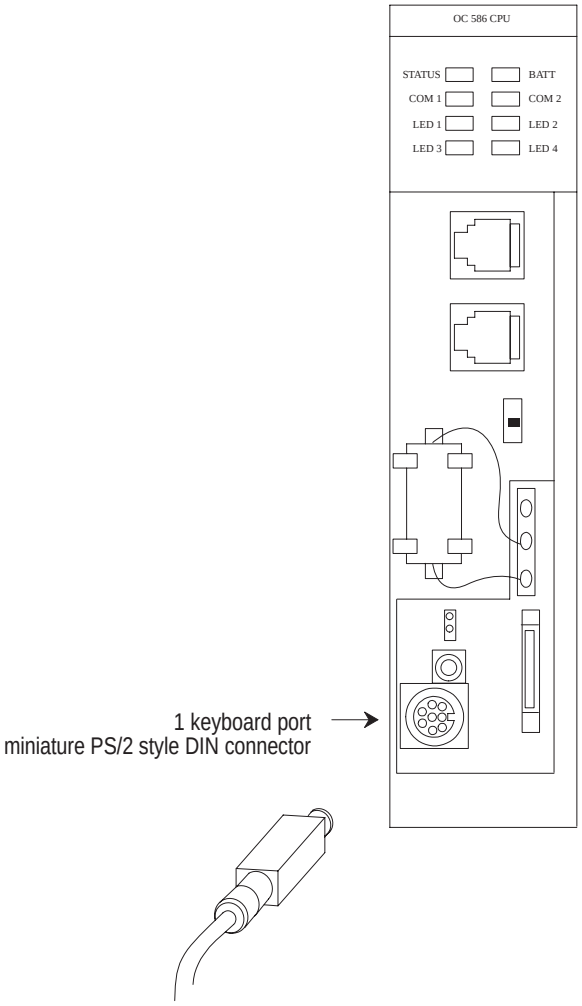
Before You Begin

This chapter shows how to make connections to the open controller CPU module and includes cable pinouts for serial and parallel cables. This chapter also describes the open controller LEDs.

Connecting the Keyboard

The open controller CPU module has one keyboard port (figure 7.1).

Figure 7.1
Connecting to the keyboard port

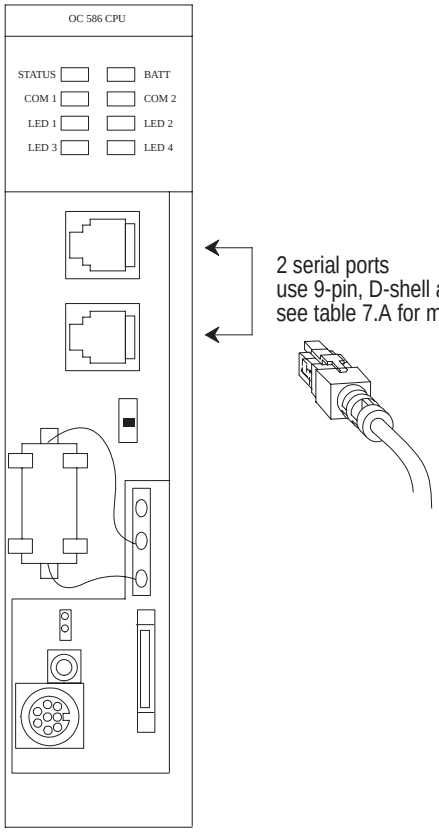


Using the Serial Ports

The open controller CPU module has two serial ports that are electrically isolated from the backplane but share a common isolated ground (figure 7.2).

COM1 and COM2 are configurable through the DOS `mode` command (`mode com`*n b*) for 110, 150, 300, 600, 1200, 2400, 4800, 9600, 19200 bit/sec. For example, at the DOS prompt, enter: `MODE COM2 48` to specify 4800 baud rate.

Figure 7.2
Connecting to the serial ports



Pin	RS-232	RS-422 (COM2 only)	RS-485 (COM2 only)
1	DCD	422 TXD+	TRXD
2	RXD	422 RXD+	
3	TXD		
4	DTR		
5	common	common	common
6	DSR	422 RXD–	
7	RTS		
8	CTS	422 TXD–	TRXD
9	N/C	N/C	N/C

In RS-422, the TXD lines are enabled when RTS is enabled. When RTS is disabled, the TXD lines go to tristate (offline).

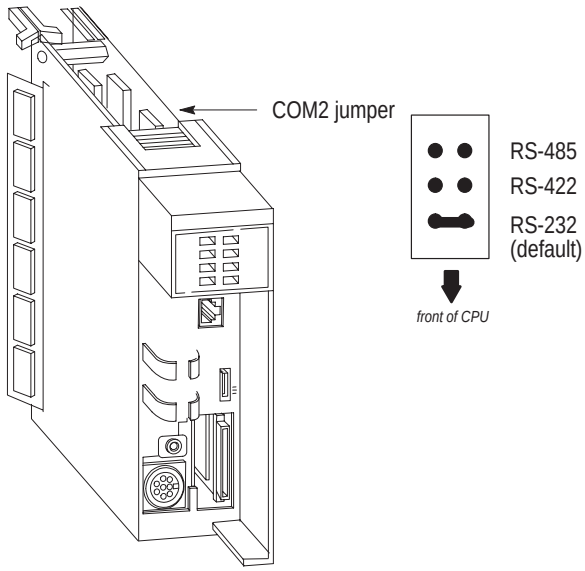
COM1 supports RS-232 only.

COM2 supports RS-232, RS-422, and RS-485. You set a jumper on the open controller CPU to select the COM2 mode of operation (figure 7.3).

Table 7.A
Maximum cable lengths for serial cables

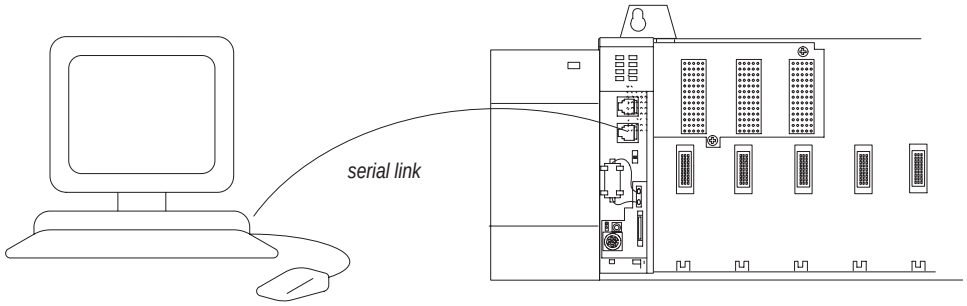
For this:	The maximum cable length is:
RS-232	50 ft (15.24 m)
RS-422	4000 ft (1219.20 m)
RS-485	4000 ft (1219.20 m)

Figure 7.3
Setting the jumper for COM2



You can also use COM2 to boot the open controller from a remote PC (figure 7.4). For more information, see page 4–6.

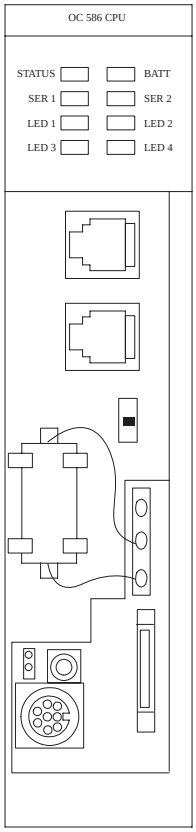
Figure 7.4
Using COM2 to boot the open controller



Using the Parallel Port

The open controller CPU module has one PC-compatible, bi-directional parallel port (figure 7.5). The parallel port is not electrically isolated from the backplane.

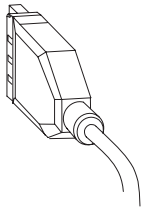
Figure 7.5
Connecting to the parallel port



Pin	Description	Pin	Description
1	strobe①	14	auto feed①
2	data bit 0	15	error①
3	data bit 1	16	initialize printer①
4	data bit 2	17	select input①
5	data bit 3	18	common
6	data bit 4	19	common
7	data bit 5	20	common
8	data bit 6	21	common
9	data bit 7	22	common
10	acknowledge①	23	common
11	busy	24	common
12	paper end	25	common
13	select		

① low true

1 parallel port
use 25-pin micro D-shell adapter cable 1747-OCP252
maximum cable length 10 ft (3.05 m)

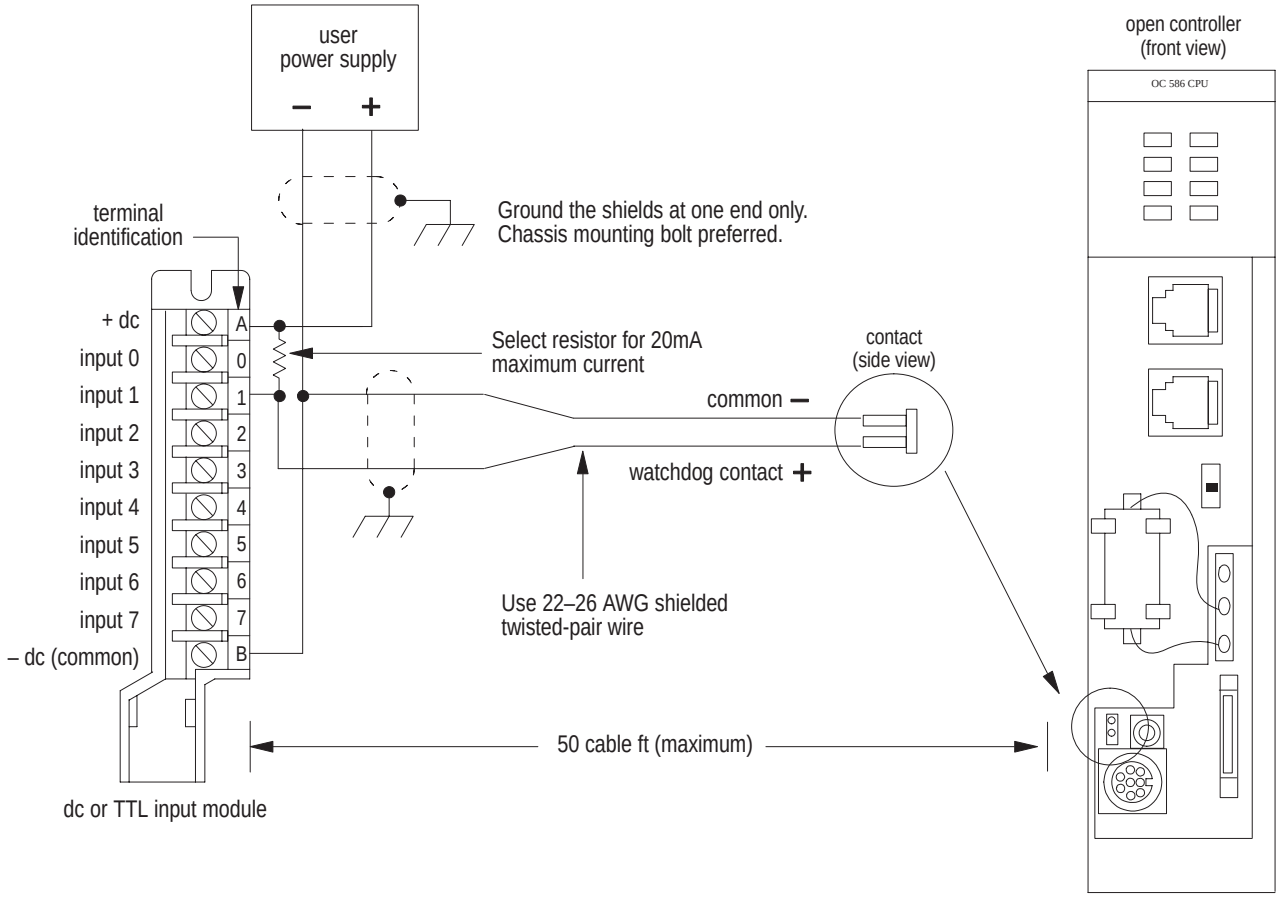


Using the External Watchdog Contact

The watchdog contact output is a solid state switch that can switch a voltage range of 4.5V dc to 26.4V dc.

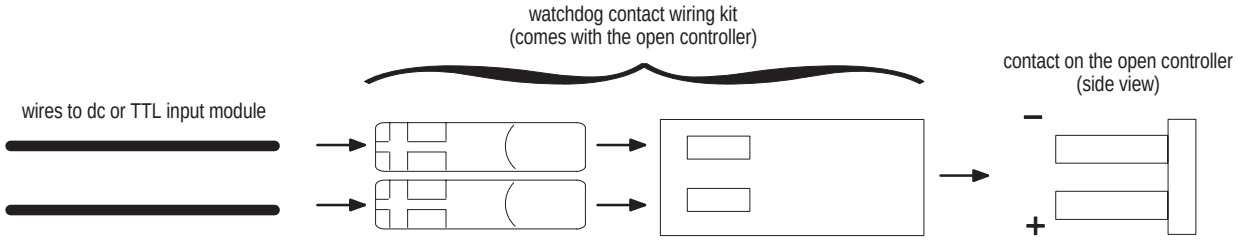
You need an external power supply to use the external watchdog. Adjust the external load switched by the watchdog output to 20 mA or less. The watchdog output is normally off and turns on as a result of a watchdog timeout condition. You must adhere to the watchdog output polarization to guarantee proper operation.

Figure 7.6
External watchdog contact output wiring diagram



A watchdog wiring kit comes with the open controller CPU module. It includes four pins and one connector.

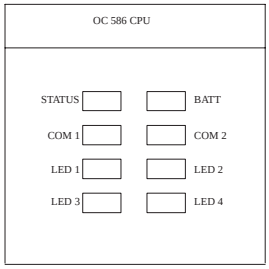
Figure 7.7
Connecting the external watchdog contact



The following specifications define watchdog components:

Component:	Specification:
housing	AMP 87499–4
pins	AMP 1–87309–4
recommended hand crimp	AMP 90202–2

Using the LEDs



STATUS

The STATUS indicator reports the status of the open controller scanner CPU. Table 7.B lists the LED patterns for STATUS.

Table 7.B
LED patterns for STATUS during normal operation

These states are possible:	Which mean:
yellow	running POST
flashing green	idle mode; not scanning I/O
solid green	scan mode; scanning I/O
flashing red	I/O fault
solid red	internal fault / POST failure
off	invalid state

BATT

The BATT indicator reports the health of the open controller battery (1747-BA). Table 7.C lists the LED patterns for BATT.

Table 7.C
LED patterns for BATT

These states are possible:	Which mean:
off	battery OK
red	battery low / battery dead

COM 1, COM 2

The COM 1 and COM 2 indicators report serial port activity and are not controlled by the open controller CPU module. These LEDs flash green any time there is activity (receive or transmit) over the serial link.

LED 1, LED 2, LED 3, LED 4

The LED 1, LED 2, LED 3, and LED 4 indicators are user-defined. Your application uses the open controller API software to define LED conditions. Table 7.D lists the possible LED patterns for LED 1, LED 2, LED 3, and LED 4.

Table 7.D
LED patterns for LED 1, LED 2, LED 3, and LED 4

This LED:	Can have these states:
LED 1 and LED 2	solid red flashing red solid green flashing green off
LED 3 and LED 4	solid red solid green off

During the power-on self test (POST), LED 1, LED 2, LED 3, and LED 4 indicate which test is running (if the STATUS LED is yellow) or which test failed (if the STATUS LED is red). Table 7.E lists the LED patterns during POST.

Table 7.E
LED patterns for LED 1, LED 2, LED 3, LED 4 during POST

This LED:	Can have these states:	Which indicate these tests:
LED 1	solid red	software CRC checksum
	flashing red	reserved
	solid green	128K bytes RAM
	flashing green	dual-port RAM
LED 2	solid red	backplane
	flashing red	reserved
	solid green	reserved
	flashing green	reserved
LED 3	solid red	reserved
	solid green	temperature sensor
LED 4	solid red	interrupt controller
	solid green	timer

Notes:

Troubleshooting

Before You Begin

This chapter describes the utilities that come on the diagnostic/utilities disk with the open controller CPU module. These utilities are also on the FlashDrive if you ordered a preconfigured open controller with a FlashDrive and MS-DOS installed. The smaller-memory versions of the FlashDrive have a subset of these utilities, as space allows.

In addition to the open controller diagnostic/utilitiees disk, you can also use standard PC troubleshooting software packages, such as QAPLUS, on the open controller system.

Using the Diagnostics Utility

The diagnostic/utility disk has a menu system for selecting the utility you want to run. You can either run the program `ocmenu` to start the menu system or you can make the disk a bootable disk and use our `autoexec.bat` file that launches the menu system automatically.

Important: The diagnostic utility disk includes a `readme.txt` file that contains important information. This file also contains instructions for making the disk bootable.

We recommend that you make the diagnostic/utility disk bootable (use the DOS `sys` command) so you can run diagnostic utilities from a remote PC that is connected to the open controller CPU module.

Most of the diagnostic utilities come in two versions:

- DOS, Windows 3.x, and Windows 95
- Windows NT

The Windows NT version requires that you load the open controller runtime API for Windows NT.

If you are using a different operating system, connect to the open controller using the `hostsvr` utility and run the diagnostic utility.

The diagnostic utility disk contains these utilities:

This utility selection:	Uses this executable:
OC scanner diagnostics	`\scanner\ocdiag.exe` This utility exercises the functionality of the open controller and tests scanner communication to the I/O modules in the 1746 chassis. This utility runs under DOS or in a DOS shell under Windows 3.x or Windows 95. A version of `ocdiag.exe` that runs under Windows NT is available by contacting Allen-Bradley Technical Support Services (TSS) at 216-646-6800. You can also download a copy from the TSS Bulletin Board using a modem at 216-646-5441.
KTX DH+/DH-485 diagnostics	`\ktxdiag\ktxdiag.exe` This utility verifies KTX/D operation on the DH+ or DH-485 network. This utility performs onboard KTX/D tests and verifies memory location and interrupt selection for the KTX/D. This is the same utility that ships with the 1784-KTx communication card.
KTX RIO scanner diagnostics	`\ktsdiag\kts_diag.exe` This utility verifies operation of the KTX/D as a remote I/O scanner. Use this utility to modify I/O image tables and perform block-transfers. The `kts_diag.exe` utility requires `ansi.sys` to be loaded in the `config.sys` file.
PCI backplane/card utility for DOS	`\pci\ocpci.exe` This utility prints to the screen the current PCI information, such as assigned PCI interrupts and memory locations of modules installed in the PCI expansion bus. This utility is a DOS-executable program only. Use the `–v` (verbose mode) switch to display additional PCI information. This mode also displays an extra PCI slot that physically is not there and is intended for engineering development purposes only. You can run `ocpci.exe` in DOS 6.22 or in a DOS shell within Windows 3.x or Windows 95.
PCI backplane/card utility for Windows NT	`\pci\ocpci_nt.exe` This utility prints to the screen the current PCI information, such as assigned PCI interrupts and memory locations of modules installed in the PCI expansion bus. This utility is a Windows NT-executable program only. Use the `–v` (verbose mode) switch to display additional PCI information. This mode also displays an extra PCI slot that physically is not there and is intended for engineering development purposes only. The runtime API for Windows NT must be loaded before executing this utility. If you don't have the runtime API, you can boot to DOS (possibly by using the `hostsvr` utility) and run `ocpci.exe`. The system resources are allocated independently of the operating system being run and don't change unless you modify system hardware.
BIOS save utility	`\bios\getcmos.exe` This utility saves the current BIOS settings to a file, which can later be restored to the same or different open controller CPU module running the same version of BIOS. See page 3–4.

This utility selection:	Uses this executable:
I/O configuration utility	`\scanner\ioconfig.exe` This utility creates I/O configuration files for the open controller, either in an offline mode or online to the open controller. This utility runs under DOS or in a DOS shell under Windows 3.x or Windows 95. A version of `ioconfig.exe` that runs under Windows NT is available by contacting Allen-Bradley Technical Support Services (TSS) at 216-646-6800. You can also download a copy from the TSS Bulletin Board using a modem at 216-646-5441.
remote connection utility	`hostsvr.exe` This utility is a DOS program that makes a remote PC connected to COM2 of the open controller appear as a floppy drive on the open controller. See page 4–6

Notes:

Specifications

Open Controller CPU Module Specifications

functional specifications

Characteristic:	Description:
main PC-based CPU	5x86 @ 100 MHz
local I/O scanner	188 @ 33 MHz
battery backup	for 8K dual port RAM and real-time clock
BIOS	AMIBIOS
output	external watchdog contact

hardware specifications

Characteristic:	Description:
battery	1747-BA lithium battery (contains 0.23 g lithium) two (2) year life span (may vary based on temperature)
serial communication	two (2) isolated serial ports COM1 supports RS-232 COM2 supports RS-232, RS-422, RS-485 configurable baud rates: 110, 150, 300, 600, 1200, 2400, 4800, 9600, 19200, 115200 bps
parallel communication	one (1) port
DRAM	72-pin SIMM gold-plated real parity 70 nanosecond access time
keyboard connection	miniature PS/2 style DIN connector keyboard not required
LEDs	four (4) diagnostic/status LEDs four (4) user-definable LEDs (LED1 – LED4) by application
switches	CPU reset of both the 586 CPU and local I/O scanner user-definable 3-position switch
jumpers	3-position jumper for selecting COM2 serial protocol user-definable (by API software) 2-position jumper

environmental specifications

Characteristic:	Values/Ranges:		
slot temperature			
operating	**with chassis fan**	0° to 60° C	(32 to 140° F)
storage	-40° to 85° C	(-40 to 185° F)	
relative humidity	5% to 95% noncondensing		
vibration	10 to 500 Hz 2.0 G maximum peak acceleration .012 in (peak-to-peak) displacement		
shock			
operating	30G peak for 11ms		
storage	50G peak for 11ms		
weight	14 oz (396.9 g)		
backplane current	2.25 A @ 5V dc		
agency certification (when product or packaging is marked)	UL A191 identified CE for all applicable directives CSA Class 1, Division 2, Groups A, B, C, D, Temp Code T5		

The operating temperature without the chassis fan is only 0° to 30° C (32° to 86° F) so we recommend that you always use the chassis fan.

internal FlashDrive disk specifications

FlashDrive version:	Specifications:
4 Mbytes	123 cylinders, 2 heads, and 32 sectors/track
10 Mbytes	320 cylinders, 2 heads, and 32 sectors/track
20 Mbytes	615 cylinders, 4 heads, and 17 sectors/track
60 Mbytes	640 cylinders, 6 heads, and 32 sectors/track

The maximum number of writes per sector (for any version of the FlashDrive) is 300,000 writes.

communication cables (not provided with the CPU)

Communication:	Specifications:	
serial	requires adapter cable 1747-OCS92, which contains two 2 ft (0.61 m) serial adapter cables	
	Communication rate:	**Maximum cable length:**
	RS-232	50 ft (15.24 m)
	RS-422	4000 ft (1219.20 m)
	RS-485	4000 ft (1219.20 m)
parallel	requires adapter cable 1747-OCP252, which contains one 2 ft (0.61 m) parallel adapter cable maximum cable length 10 ft (3.05 m)	

Calculating Scan Time

Add the rows in the following tables to calculate input and output scan times. Fill in the shaded boxes with the correct values. All times are in microseconds.

estimate minimum input scan time

Discrete I/O Modules: number of class 0 input words:		x 48 =	
Specialty I/O Modules (Class 1) number of 1746-BAS modules (class 1): number of 1746-NIO4V modules: number of 1746-FIO4V modules: number of 1746-NI4V modules:		 x 490 = x 265 = x 265 = x 325 =	
Specialty I/O (Class 3 and 4) number of 1746-HSCE modules: number of 1747-SN modules: number of 1746-BAS modules (class 4):		 x 712 = x 1185 = x 650 =	
Forced Input Overhead number of class 0 input words: number of specialty input words: number of specialty I/O modules:		 x 6 = x 9 = x 30 =	
Input Scan Overhead			36
		Minimum Input Scan Time (total)	

estimate maximum input scan time

Minimum Input Scan Time (from above table)			
Maximum Specialty I/O Input Scan Time number of class 1 specialty I/O modules: number of class 3 and 4 specialty I/O modules:		 x 50 = x 200 =	
		Maximum Input Scan Time (total)	

estimate minimum output scan time

Discrete I/O Modules: number of class 0 input words:		x 46 =	
Specialty I/O Modules (Class 1) number of 1746-BAS modules (class 1): number of 1746-NIO4V modules: number of 1746-FIO4V modules: number of 1746-NO4V modules:		 x 500 = x 277 = x 277 = x 330 =	
Specialty I/O (Class 3 and 4) number of 1746-HSCE modules: number of 1747-SN modules: number of 1746-BAS modules (class 4):		 x 608 = x 1208 = x 655 =	
Forced Output Overhead number of class 0 output words: number of specialty output words: number of specialty I/O modules:		 x 6 = x 9 = x 30 =	
Output Scan Overhead			33
		Minimum Output Scan Time (total)	

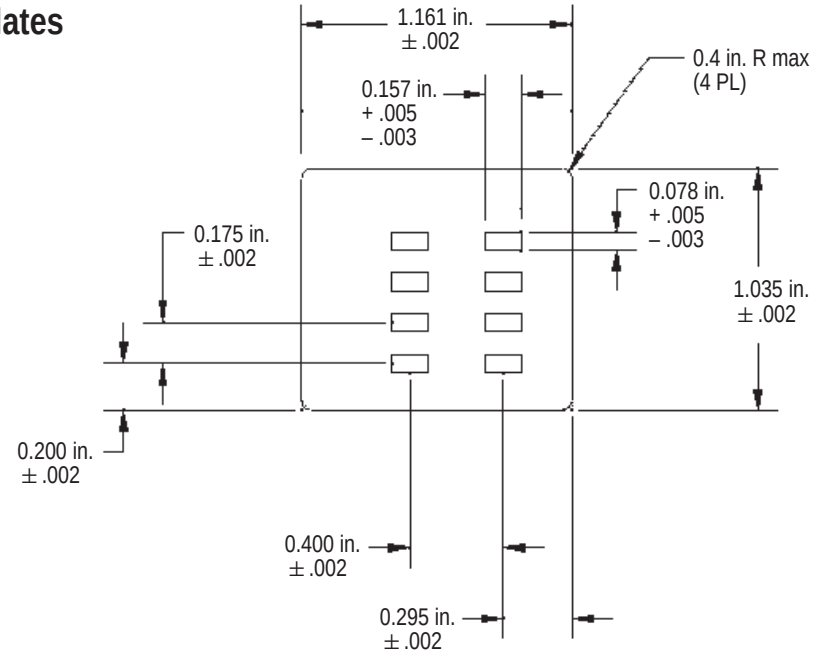
estimate maximum output scan time

Minimum Output Scan Time (from above table)			
Maximum Specialty I/O Output Scan Time number of class 1 specialty I/O modules: number of class 3 and 4 specialty I/O modules:		 x 50 = x 200 =	
		Maximum Output Scan Time (total)	

estimate output scan time plus processor overhead

Minimum or Maximum Output Scan Time (from above tables)		
Overhead if DII Enabled	60	
Processor Overhead	minimum = 140 maximum = 200	
	Output Scan Time Plus Processor Overhead (total)	

Open Controller CPU Module LED Templates



Notes:
1) Color is Pantone #431 grey over white for opacity, opposite adhesive side
2) Selective adhesive around the perimeter on the back
3) Clear windows
4) Protective cling mask
5) Material .019 in ± .002 glossy polycarbonate
6) Reference A-B drawing 98587701-101 for screen legend

The Allen-Bradley manufacturer for masks is:

Metallics
P.O. Box 99
Onalaska, WI 54650

608-781-5200

Notes:

Handling the 1747-BA Battery

Storing the Battery

Store the lithium battery in a cool, dry environment, typically 20° C to 25° C (68° F to 77° F) and 40% to 60% relative humidity. Store the batteries and a copy of the battery instruction sheet in the original container, away from flammable materials.

Handling the Battery

Follow these guidelines when using the 1747-BA battery:

- Use only for the intended operation.
- Do not ship or dispose of cells except according to recommended procedures.
- Do not ship on passenger aircraft.

ATTENTION: Do not charge the batteries. An explosion could result or the cells could overheat causing burns.

Do not open, puncture, crush, or otherwise mutilate the batteries. A possibility of an explosion exists and/or toxic, corrosive, and flammable liquids would be exposed.

Do not incinerate or expose the batteries to high temperatures. Do not attempt to solder batteries. An explosion could result.

Do not short positive and negative terminals together. Excessive heat can build up and cause severe burns.

Transporting the Battery

One or Two Batteries — Each battery contains 0.23 grams of lithium. Therefore, two batteries can be shipped together within the United States without restriction. Regulations governing shipment to or within other countries may differ.

Three or More Batteries — Procedures for the transportation of three or more batteries shipped together within the United States are specified by the Department of Transportation (DOT) in the Code of Federal Regulations, CFR49, "Transportation." An exemption to these regulations, DOT - E7052, covers the transport of certain hazardous materials classified as flammable solids. This exemption authorizes transport of lithium batteries by motor vehicle, rail freight, cargo vessel, and cargo-only aircraft, providing certain conditions are met. Transport by passenger aircraft is not permitted.

A special provision of DOT-E7052 (11th Rev., October 21, 1982, par. 8-a) provides that:

> "Persons that receive cell and batteries covered by this exemption may reship them pursuant to the provisions of 49 CFR 173.22a in any of these packages authorized in this exemption including those in which they were received."

The Code of Federal Regulations, 49 CFR 173.22a, relates to the use of packaging authorized under exemptions. In part, it requires that you must maintain a copy of the exemption at each facility where the packaging is being used in connection with shipment under the exemption.

Shipment of depleted batteries for disposal may be subject to specific regulation of the countries involved or to regulations endorsed by those countries, such as the IATA Restricted Articles Regulations of the International Air Transport Association, Geneva, Switzerland.

Important: Regulations for transportation of lithium batteries are periodically revised.

ATTENTION: Do not incinerate or dispose of lithium batteries in general trash collection. Explosion or violent rupture is possible. Batteries should be collected for disposal in a manner to prevent against short circuiting, compacting, or destruction of case integrity and hermetic seal.

For disposal, batteries must be packaged and shipped in accordance with transportation regulations, to a proper disposal site. The U.S. Department of Transportation authorizes shipment of "Lithium batteries for disposal" by motor vehicle only in regulation 173.1015 of CFR 49 (effective January 5, 1983). For information contact:

U.S. Department of Transportation
Research and Special Programs Administration
400 Seventh Street, S.W.
Washington, D.C. 20590

Although the Environmental Protection Agency at this time has no regulations specific to lithium batteries, the material contained may be considered toxic, reactive, or corrosive. The person disposing of the material is responsible for any hazard created in doing so. State and local regulations may exist regarding the disposal of these materials.

For a lithium battery product safety data sheet, contact the manufacturer:

Sanyo Energy Corporation
600 Supreme Drive
Bensenville, Il 60106

Numbers

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Rockwell Automation

Allen-Bradley

Allen-Bradley, a Rockwell Automation Business, has been helping its customers improve productivity and quality for more than 90 years. We design, manufacture and support a broad range of automation products worldwide. They include logic processors, power and motion control devices, operator interfaces, sensors and a variety of software. Rockwell is one of the world's leading technology companies.

Worldwide representation.

Argentina • Australia • Austria • Bahrain • Belgium • Brazil • Bulgaria • Canada • Chile • China, PRC • Colombia • Costa Rica • Croatia • Cyprus • Czech Republic • Denmark • Ecuador • Egypt • El Salvador • Finland • France • Germany • Greece • Guatemala • Honduras • Hong Kong • Hungary • Iceland • India • Indonesia • Ireland • Israel • Italy • Jamaica • Japan • Jordan • Korea • Kuwait • Lebanon • Malaysia • Mexico • Netherlands • New Zealand • Norway • Pakistan • Peru • Philippines • Poland • Portugal • Puerto Rico • Qatar • Romania • Russia–CIS • Saudi Arabia • Singapore • Slovakia • Slovenia • South Africa, Republic • Spain • Sweden • Switzerland • Taiwan • Thailand • Turkey • United Arab Emirates • United Kingdom • United States • Uruguay • Venezuela • Yugoslavia

Allen-Bradley Headquarters, 1201 South Second Street, Milwaukee, WI 53204 USA, Tel: (1) 414 382-2000 Fax: (1) 414 382-4444

Publication 1747-6.16 July 1997
Supersedes 1747-6.16 October 1996
1747-6.16-RN1 March 1997

PN 955128-44