PC Configuration for NEXTSTEP

Terry Williams

Undoubtedly, one of the most challenging tasks facing a NEXTSTEP system administrator or installer is configuring PCs to run NEXTSTEP. There's a plethora of PC-compatible adapters and products to deal with, and each has its own quirks.

There are, however, a few *de facto* rules that make PC configuration easier. In most cases, the defaults that the NEXTSTEP installation software uses will be fine, and you will have to do only a little tweaking. In a few instances, though, you might have to do more than tweak. This article describes the configuration pieces that are common in PCs, and helps you see how they fit together.

HARDWARE INTERRUPT REQUEST LINES

Probably the most difficult and frustrating task in setting up a PC is dealing with hardware interrupts. Each peripheral system and device uses hardware interrupts to gain access to the CPU, using an *Interrupt Request line* (*IRQ line*). Each connected peripheral must be set to a unique hardware interrupt. There are 16 distinct levels of these interrupts in a PC, and therefore 16 IRQ lines. You have to make sure they're all assigned properly.

How interrupts work

When a device needs access to the CPU, it raises a signal (or ªinterruptº) on its IRQ line. If the CPU's current task has a lower priority than the interrupt, the CPU puts aside what it's doing to immediately handle the interrupt. If more than one interrupt arrives at a time or if the CPU is already servicing an interrupt, the CPU honors the lowest numbered (highest priority) IRQ line first. For example, if the CPU is reading something from a disk and you type something, the keyboard interrupt causes the CPU to put the disk read temporarily on hold. When the CPU completes the interrupt task, it clears that device's IRQ line and continues with what it was doing

before.

Problems arise when you have multiple devices on the same IRQ line (except in special cases, as described below). The CPU and driver software that manage the device expect particular interrupts from particular devices, and things get confusing if the CPU sees an interrupt from a fax modem, for example, on an IRQ line that is set for something else, like an Ethernet adapter.

Side note: Devices can sometimes share interrupts, but only if the device driver software can handle such situations. Typically, interrupt sharing is managed by adapters that support multiple serial interface or SCSI ports. Some EISA and MCA adapters support interrupt sharing. However, NEXTSTEP for Intel Processors doesn't currently support any interrupt sharing.

Fixed and available IRQs

When configuring a PC, you need to find a unique IRQ line for each device or peripheral system connected to your PC. Some IRQs are traditionally assigned to particular devices, while others are available for you to assign.

Some IRQs that are traditionally designated for particular devices can sometimes be available anyway. For example, if a system doesn't have a serial port 2, or if nothing is connected to it, its traditional interrupt line (IRQ 3) would be available for, say, the Ethernet adapter.

However, some IRQ lines are fixed and can't ever be used by adapters. For example, IRQ lines 0, 1, 2, 8, and 13 are never available.

The following table shows common and required settings for IRQ lines.

IRQ line	Device
0	System clock (fixed)
1	Keyboard (fixed)
2	Cascade from interrupt controller #2 (fixed)
3	Serial port 2
4	Serial port 1
5	Parallel port 2
6	Floppy disk drive controller
7	Parallel port 1

8	Real-time clock (fixed)
9	Redirected as IRQ2
10	Available
11	Available
12	PS/2 mouse
13	Math coprocessor (fixed)
14	Hard disk controller
15	Available

I/O PORT ADDRESSES

The CPU uses I/O ports to communicate directly with devices. An *I/O port* tells the CPU where to find data, control information, and status information for the associated device. Ports are much like memory addresses and occupy address space, but they aren't used for storage. A port is where the CPU goes to get data from the port's assigned device or to check the status of the device, and where the CPU puts any data it's sending to the device.

Each device must have an I/O port and, like IRQ lines, devices often have traditional port address ranges. However, many adapters allow you to set or change their port addresses. You usually view and change the addresses through the **BIOS Setup** utility. (If you have an EISA system, you might also need to use the EISA Configuration Utility (ECU) that comes with your system to work with port addresses.)

The following table lists a few common I/O port address ranges.

Address range	Device
0A0 - 0BF	Interrupt controller 2
0F8 - 0FF	Math coprocessor
1F0 - 1F8	Primary hard disk controller
170 - 178	Secondary hard disk controller
278 - 27F	Parallel port 2
2B0 - 2DF	VGA
2F8 - 2FF	Serial port 2
378 - 37F	Parallel port 1

380 - 38F	SDLC, bisynchronous 2
3A0 - 3AF	Bisynchronous 1
3B0 - 3BF	Monochrome adapter with printer port
3F0 - 3F7	Floppy disk drive controller
3F8 - 3FF	Serial port 1

DIRECT MEMORY ACCESS CHANNELS

Directly Memory Access channels (*DMA channels*) are communication channels for the CPU and peripherals that need direct access to data in memory. A special controller called a *DMA controller* deals with transferring data to and from memory, freeing the CPU for other tasks. Some peripherals can make requests of the DMA controller without having to bother the CPUÐthis speeds things along quite a bit.

DMA channels are usually assigned to high speed communication devices like Ethernet adapters, floppy disk drives, and SCSI controllers. A DMA channel should be unique to a particular device; however, it can be shared as long as it's not used simultaneously by multiple devices. Below is a table of standard DMA assignments.

DMA channel	Standard usage
0	Available for 16 bit
1	Available
2	Floppy disk drive controller
3	Available
4	Cascade for DMA 0 - 3
5	Available for 16 bit
6	Available for 16 bit
7	Available for 16 bit

RAM BASE ADDRESS

For some devices, such as video boards, you have to specify the starting address of the program that controls the device (the driver software or the adapter's ROM) or the data used by the device. This location is called the *RAM base address*. A number of things can occupy this space. For example, a video device might have data memory in these addresses. Make sure that the RAM for one device doesn't

overlap that of another deviceÐthat is, make sure its RAM base address is not within the range of another device. The following table shows how some of this reserve memory is usually laid out.

Address range	Purpose
A0000h - BFFFFh	Video RAM
C0000h - D0000h	Adapter board ROM area (available)
E0000h - FFFFFh	Motherboard ROM BIOS (some of this may be available)
100000h - 1FFFFFh	DOS high memory area

HOW DO YOU SET THESE THINGS?

Now that you know what everything is, how do you work with it? It depends on the architecture of your PC and devices.

Industry Standard Architecture (ISA)

Most ISA components are jumper configured, which means you have to physically set switches on the board or device. However, if the device comes with configuration software, it may be more convenient to use the software than to fiddle with the jumper settings. Follow the manufacturer's suggested settings, unless NEXTSTEP calls for something differentÐthere are rare occasions when the manufacturer defaults for NEXTSTEP-supported devices are not appropriate for use with NEXTSTEP.

Check the documentation for the device to find the manufacturer's suggestions, and see the individual configuration settings in your NEXTSTEP installation guidelines or NeXTanswers for exceptions to the defaults.

Extended Industry Standard Architecture (EISA)

Some EISA adapters have jumpers that you set, but most do not. When you install an EISA adapter, you must add it to the EISA ROM before you can use it. The ECU that ships with your system's motherboard probably adds the board for you automatically when you first run it; see the ECU documentation. When the board is added and identified, you're asked to insert a floppy disk that has a particular configuration fileÐa floppy disk comes with each EISA adapter that you install. Even if the board is automatically added, check the details of the board's settings in the

ECU program. While doing that, make performance adjustments as necessary. In general, follow the EISA adapter documentation.

The ECU also configures the bus slot for the particular adapter. If you change the slot position of the adapter, you must rerun the ECU. The changes to the EISA configuration information are loaded into EISA ROM when you save and exit. In fact, EISA system configuration is downloaded each time you turn on the computer.

You configure ISA boards in an EISA system exactly as if you were installing them in an ISA system.

IS IT A PC YET?

To see if the thing works, you have to install at least the minimum necessary components that allow the computer to boot and display video. You need:

· A video adapter and display (for testing you can use the internal video adapter, if your computer has one)

· A keyboard

· A floppy disk drive

· A bootable floppy disk (for DOS or some other operating system)

· Some memory

To run NEXTSTEP you also need:

· A NEXTSTEP-supported video adapter

· A NEXTSTEP-supported mouse

· A 250MB or larger hard disk

· 16MB or more of memory

You need to know:

· How to access the ROM BIOS of your system (usually there's a key sequence that you press after the computer finishes its memory test). See your PC's documentation.

- · How to load and start the ECU program if the computer is an EISA computer.

- · The devices that are present and their proper hardware configuration.

YOU'RE OFF!

So, configuring your PC doesn't have to be dreadful, if you read the manufacturer's documentation and double-check for conflicts between devices. For additional tips on installing a particular device, be sure to check the configuration information available from NeXTanswers. *(Editor's note: See ªTips & Techniquesº for details.)*

SUGGESTED READING

There are many good books on PC architecture available. Here are two we particularly like:

Rosch, Winn L. *The Winn Rosch Hardware Bible*, 2nd ed. New York: Brady Pub., 1992.

Angermeyer, John, et al. *The Waite Group's Tricks of the MS-DOS Masters*, 2nd ed. Carmel, IN: Howard W. Sams & Co., 1990.

*Terry Williams is a System Administration Instructor in NeXT Education. You can reach him by e-mail at **Terry_Williams@next.com**.*

CONFIGURATION TROUBLESHOOTING

No video

If no video is present, make sure that your video system is set up properly.

- · Check that the cabling is installed correctly.

- · If you are using an expansion bus video adapter, be sure that any on-board video systems are

disabled.

- · If you are using on-board video, check that it's enabled.

- · Check that the ROM BIOS setup is configured properly for your video application.

Poor video

Poor video can mean ªsnowº in the output, poor horizontal or vertical synchronization (flipping screens), dim output, or bright output.

- · Check that your cabling is installed correctly.

- · Check the configuration of the monitor.

- · Check the configuration of the video adapter.

System won't boot past memory test

Memory sails past, but the system doesn't load the software.

- · If you are using an expansion bus hard disk controller, make sure its IRQ line doesn't conflict with anything.

- · Disable either an integrated IDE controller or the expansion bus hard disk controller.

- · Check that the hard disk controller's IRQ line is not set to 6 (which must be the floppy disk drive controller).

System boots DOS (not NEXTSTEP) by default

This assumes that NEXTSTEP still boots properly when you type **n** at the ªn for NEXTSTEP, d for DOSº prompt at startup. Usually, this is because a DOS partition is the active partition.

- · Use **fdisk** to set the NEXTSTEP partition to be active.

- · The boot sector or file may have been corrupted. Run the UNIX utility **/usr/etc/disk** to rewrite the NEXTSTEP boot blocks.

System fails to boot NEXTSTEP in multi-user mode

The most likely causes for this are improperly configured NEXTSTEP drivers or incompatibility between the drivers and the hardware.

- · Ensure that all IRQ lines, I/O port addresses, and RAM base addresses are set properly and

are unique between devices.

· Try to boot in the default configuration. To do this, type the following at the boot prompt:

```
config=Default
```

Now, run Configure to reconfigure the system.

· Boot in single-user mode and create a new configuration table. To do this, type the following at the boot prompt:

```
-s config=Default
```

Now copy /usr/Devices/System.config/Default.table to /usr/Devices/System.config/Instance0.table.

Other problems

In general, check NeXTanswers for ways to avoid configuration gotchas for all devices, and to fix them if they already got you.ÐTW