Modems & Windows

by Fredrick Lovret

Microsoft has an excellent technical document, #WW0654, which goes into great detail about serial communications and Windows. Unfortunately, many users find this paper a bit too technical. So, after seeing a continuous stream of similar questions on Compuserve, I took the time to write this. Feel free to pass it around and, hopefully, it will make things a little easier for you.

Recently, while testing a communications application that I had just written, I had the opportunity to spend a day experimenting with a variety of modems. You may find the results instructive.

I had three 28.8k, two 14.4k, and two 2400bps modems. All of these were from different manufacturers and internal. I also had two computers opened up: a low-quality 386/33 and a high-quality 486/66. In addition, I had a 486/66 at a remote site available. Finally, I had several communications and fax applications (in addition to the one I wrote).

I spent a full day swapping modems among these devices, checking different configurations, and some of the results were surprising.

Background

First, a bit of history. A few years ago, computers came with one serial port in the form of an add-on card, and this was set to Comm1. Internal modems were normally factory-set to Comm2, and all you had to do was plug them in. Modern computers, however, usually come with two serial ports (Comm1 and Comm2) built in to the motherboard. The problem is that most internal modems are still being preset at the factory to Comm2. This causes problems.

Now for some technical data—Don't get excited; I'm only going to give you the absolute minimum. A comm port, such as Comm1, is really just a memory address in your computer. The standards are:

Comm1	3f8
Comm2	2f8
Comm3	3e8
Comm4	2f8

The odd looking numbers are in hex format (3f8 hex = 1016 decimal).

Unless you have a strange system, there is no reason to change the default settings. These addresses tell your computer where (in its memory) to look for information from the modem (or where to send it when transmitting).

In addition to where, your computer needs to know when. This is taken care of by interrupts, signals which tell your computer to stop what it's doing and pay attention to the modem. These are known as IRQs, and the defaults are:

Comm1	4
Comm2	3
Comm3	4
Comm4	3

Note that they are shared. This is because you only have a limited number available. If you use MSD.EXE (it's in your Windows directory) you can see these assignments.

So, you pull the new modem out of the box, note that it is set to Comm2/IRQ3 and plug it in. What happens? Probably nothing good. It has an address and interrupt conflict with the serial port on your motherboard, and the results will be erratic. Why erratic? That has to do with the way communications applications are written.

There are three basic ways to write a comm app: BIOS, DOS, and Windows API. You don't really need to know the details about these methods, just remember that it means that while one program may work in a given system, another won't.

Communications applications written with the Windows API are much more sensitive to conflicts than are the other types. Unfortunately, while computer technicians are knowledgeable about hardware, few are experts on Windows. You can, therefore, have a technician who states that your modem is properly installed when it isn't.

Finally, the settings themselves. These are normally made with hardware. That means that you set the address and interrupt by physically changing a switch or jumper on the card itself. (Some devices, especially sound cards, allow you to do this through software with a thing called flash-bios.) Changing the settings in your comm app, or the ones in the Ports section of Control Panel, has no effect on the card. Again, Control Panel is used ***only*** to tell Windows what the physical settings on the board are; it does ***not*** change the existing settings.

Experiments

As a programmer, I understood how comm ports worked. But, I was still quite surprised at the results of some of these tests.

• Although a low-speed (2400bps) modem would tolerate IRQ conflicts, high-speed devices wouldn’t. Sharing an IRQ would work sometimes with some comm apps on a 14.4k modem, but never on most 28.8k modems. This seemed to vary with manufacturer. What this means is that, if you replace your 2400bps modem with a high-speed modem, and set the new device to the same settings as the old board, there is no guarantee that it will work.

• With a couple of combinations, disabling the on-board serial port through CMOS did not give reliable results. With one combination of computer and modem, it didn’t work at all.

• Given a marginal installation, some Windows comm apps would work, while others wouldn’t. This was especially true for Terminal: there were a couple of configurations where I could send an “AT” to the modem with Terminal and get an “OK” response, but a more advanced application still would not work. (This was probably due to the difference between sending signals slowly at typing speed and sending them in a burst at full speed.)

• Setting a modem to Comm3/IRQ4 (without changing the motherboard Comm1) and installing the mouse on Comm2 worked with both the 2400 and 14.4 modems, but not with the 28.8 modems.

• Although you may see complaints about the Windows comm driver, I had no problem using it at high speeds. Although I wasn’t set up to measure actual transfer speed, a comm app I was using was equipped to measure time from initial request to completion of transfer (when the incoming data was actually written to disk and the file closed) and the results were always well over 30k bps.

• Sometimes two 28.8k modems would fail to connect, even though both were from the same manufacturer. Immediately trying again always worked.

• The remote computer, which was advertised as having automatic port configuration, didn’t. And it didn’t provide a method of manually changing the motherboard settings.

I still don’t understand some of these results—after all, an address is an address, and an IRQ is an IRQ, so speed or manufacturer shouldn't matter —but they were repeatable and I couldn't ignore them.

A Proper Installation

After switching and swapping all day long, and observing the results, I strongly advise you to do the following (even if your system appears to be running perfectly):

1) Open your computer and examine and record the jumper (or switch) settings on each card. Compare what you see with the manuals and make a chart. It is important that you make this a physical inspection; software searches, such as with MSD, do not detect all devices. Do this for ***everything***, including your motherboard and video card.

2) Looking at the chart, set the cards so that no two addresses or IRQs are the same. This may require some juggling, as the range of choices on some cards is very limited. Again, don't share ***anything***. Although you may be able to get away with it now, it will come back to haunt you in the future (especially if you upgrade to WindowsNT).

3) When everything is set, record the settings in a CardFile database so that you never have to go through this again. Also, make a hard-copy of the data and tape it to the back of your computer.

Again (and again and again), make this a physical inspection and check everything. I am embarrassed to admit that, when I did it, things weren't set up as I had thought they were.

For most systems, the easiest installation is to set the new modem to Comm3 and use one of the high-level IRQs (such as 11 or 12). Use MSD.EXE to see what is available: The ones labeled "Reserved" are free (unless you have some type of device which MSD can't see). Also, you can usually get away with using #7.

Next simplest is to disable a motherboard port and use its settings for your modem. Do this by booting your computer and pressing the Delete key when prompted. There should be an area in the CMOS Set Up screen called "Peripheral Management", or something like that. Select the correct port, disable it, and then save the settings at the prompt. Be advised that some computers do not provide this feature.

In the worst case, where their isn't an available address or interrupt available for the modem, you may have to install a combination IO card. These, often inexpensive, boards consolidate several functions on one card and can usually take full advantage of the high-level IRQs. They can, therefore, often free a low-level IRQ for use by the modem.

Standard Problems

Here are a few situations that I see regularly with my customers...

• Two serial ports on the motherboard, with the mouse plugged in to Comm1. If you put the modem on Comm2, it will conflict with the on-board port's address; if you put it on Comm1 or Comm3 (IRQ4), it conflicts with the mouse. You may install the modem on Comm4/IRQ3, but you must add the line

```
Com3IRQ=-1
```

to the [386Enh] section of the system.ini file. (This tells Windows that you don't have Comm3 installed. Without this line, it won't work.) I don't advise this, as it may cause problems in the future.

• Same as above, but Comm2 on the motherboard was disabled in CMOS and the modem takes its place. As previously noted, this caused erratic performance with certain combinations.

• Modem is improperly installed, and the user attempts to fix things with ControlPanel\Ports. Again, this only tells Windows what the physical settings on the card are; it does ***not*** change those settings.

• Speed is incorrect. When you set speed in your communication application, this should be the maximum throughput of the modem, not its bps speed. (e.g., 57600 for a 14.4k modem). The speed setting tells your computer how fast to talk to the modem; it does not tell the modem how fast to talk to another modem. (To do that you have to make several other command-line changes.) If speed is set too low, a buffer overrun can occur. This will usually cause a GPF.

• WindowsNT. Windows 3.1 is more sensitive to IRQ settings than was DOS, and WinNT is much more sensitive than Win3. This means that a configuration that worked fine under Windows 3.1 may not work at all under NT.

• Application conflict. Under Windows, only one app can use a port at a time. So, if you have the fax program for your fax-modem running, you must stop it before using the modem's communication program. (There is software available to get around this limitation.)

A final word of advice: After learning more about serial ports, I am now much more selective when purchasing hardware. I always check, in addition to the performance figures of a new device, what the address and interupt range is. And, with experience, I have come to prefer flexibility over performance. So, before purchasing a new motherboard, make sure to check that disabling the on-board devices (such as ports and IDE drive connectors) is possible. For new modems, check the IRQ options just as carefully as you check maximum speed.

In conclusion, you can get away with marginal installations most of the time. (By marginal, I mean an address or interrupt conflict, even if the conflicting device is not in use.) However, sooner or later, this can cause a problem. And, as we have a new version of Windows that will soon be available, I advise that you make sure (absolutely 100% sure) that your hardware is set up perfectly.

###