Serial solution

Roger Gann shows you how to maximise comms speed by making the most of that unsung PC hero, the serial port. Plus, do *you* know the difference between baud rate and bps?

This month I'm taking a look at that unsung hero of the internet, the humble PC serial port — without it, where would modems be? I'll be showing you various ways to ensure you're squeezing every last drop of performance out of it, so that it doesn't bottleneck your modem and mar your surfing "experience".

The first port of call

The original PC serial interface used the 8250 (later 16450) Universal Asynchronous Receive/Transmit or UART chip. The 8250 UART has only a one-byte receive buffer, which means that after the 8250 receives a character, it has to be retrieved before the next character arrives or it's overwritten, i.e. lost. This leads to CRC errors, with consequent retransmissions and an inevitable impact on throughput.

At 1,200bps, a new character arrives about every 8ms — plenty of time for an interrupt handler to fetch the character from the UART and store it in a receive buffer. But at 19,200bps, a new character arrives about every 0.5ms, and at 115,200 baud (the maximum speed of the standard PC COM port) about every 90 microseconds. Depending on your CPU speed, there may not be enough time to reliably service one interrupt per character. As a result, the effective throughput ceiling of an 8250 UART is 38,400bps.

The superior 16550A or AFN UART overcomes many of the limitations of the 8250 and is reliable at speeds of up to 115,200bps. It has two main improvements over the 8250. Firstly, it has a pair of 16-byte send and receive buffers, thus giving it 16 times the buffering capacity of the 8250. Secondly, it has a First In, First Out (FIFO) buffering scheme, which can dramatically improve performance on modem transfer speeds of 9,600bps or higher.

To see which UARTs are installed on your system, exit Windows and run the MSD utility included with DOS 5.0 and later by entering MSD at the DOS prompt. Once in the program, press C to see your COM ports. The type of UART installed for each of your ports appears on-screen.

Serial port upgrades

If you have an 8250 UART you have two choices: you can either upgrade to a 16550A or buy a third-party card, such as the Hayes ESP II. Note that if you have an internal card modem, it will most probably have its own 16550 UART aboard, so upgrading your PC's UARTs won't be necessary.

Upgrading 8250 UARTs

If you have a pretty old I/O card in your PC, the chances are it's got a 40-pin socketed 8250 UART. In this case, upgrading is simple. The 16550A is pin-compatible with the 8250, so all you've got to do is obtain a 16550AFN UART from somewhere like Maplins or Radio Spares, unplug the 8250 and replace it with the 16550A. However, if the 8250 is soldered in or you've got a more modern I/O card with a multi-function ASIC chip, then it's much less hassle to buy a £20 replacement I/O card.

Hayes ESP

Hayes Optima V.34 modems use a bigger "data dictionary" than their rivals and this enables V.42bis data compression to achieve 8:1 compression ratios, twice as much as normal. On a 28.8Kbps modem this would give you a maximum potential data transfer rate of 230Kbps, way beyond the capabilities of a "fast" 16550 serial port. So Hayes offers its own serial-card solution, the Hayes ESP Communications Accelerator. This 16-bit ISA card is outwardly 16550-compatible but features a co-processor plus a pair of 1Kb send and receive buffers designed to eliminate buffer overflow problems. The ESP has had a spot of bother on the driver front over the years and the NT drivers have yet to emerge from interminable beta, but the Windows 95 drivers work just fine. The ESP II single port card goes for around £60 plus VAT. If you're serious about high-speed comms, this is the card to use.

Software tweaks

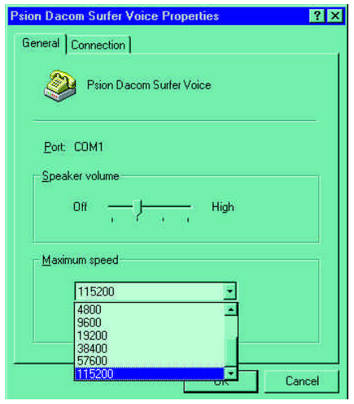
As well as upgrading your serial port hardware, there are a number of software tweaks you can make to improve data throughput.

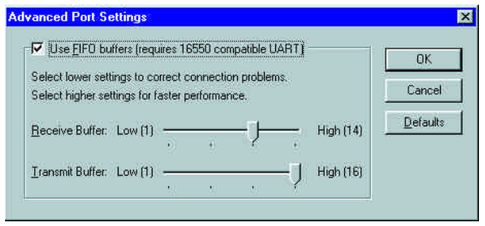
■ Flow control

With today's fast data transfer rates it's essential for the computer to be able to exercise some control over the flow of data to and from the modem, in order to prevent data loss. After all, there's no point in sending more data if the last lot hasn't been dealt with. This process is called flow control and there are several ways of doing it. The slowest way is software flow control, which makes use of two special characters, XON (Transmit on) and XOFF (Transmit off). When software flow control is in effect, sending a modem an XOFF character stops the flow of data. Sending the modem an XON character causes data to flow again. This is an inefficient method, significantly reducing bandwidth by adding two control characters for every eight sent.

The alternative is hardware flow control or RTS/CTS, which is more responsive and efficient. This uses a set of pins in the serial port, "Ready to Send" and "Clear to Send", and depending on their "state", data is either sent or paused. A minor caveat: you will need a decent, fully-wired serial cable with the CTS/RTS wires in place for hardware handshaking to work. Note that this type of flow control isn't the exclusive preserve of external modems — internal modems can also take advantage of it.

So how do you choose hardware flow control? In Windows 3.1x open Control Panel, then double-click on the Ports icon. The resulting window offers an icon for each COM port. Double-click on the icon of the port you want to configure, then click on the Settings button. Select Hardware from the Flow Control list box, and then click on OK. In Windows 95 click on the Modem applet in Control Panel, highlight your modem and click on the Properties button. Click on the Connection tab and then on the Advanced button. Check the "Use Flow Control" box and select Hardware (RTS/CTS). Click OK a few times to back out.





Top Windows traditionally errs on the side of caution when it comes to modem line speeds, but it's normally quite safe to throw caution to the wind and crank it up to its maximum speed: 115,200bps in the case of V.34 modems

Above Once again, Windows errs on the side of caution when it comes to the FIFO buffers in the 16550 UART. Experiment by upping the Receive Buffer from 14 to 16

You may also need to tell your comms apps to use hardware flow control. For example, if you use WinCIM 2.01, simply add the line

```
FlowControl=3
```

to the [Connector (CIS Connection)] in the CIS.INI file.

You'll also need to tell your modem which type of flow control to use. Check through its "initialisation string" and see if you can see "&K3", the usual Hayes command for turning on hardware flow control. You'll need to do this for DOS and Windows 3.1x comms apps; under Windows 95, provided you've installed the modem correctly, turning on the hardware flow control via the Control Panel modems applet is sufficient.

■ Windows 3.1x

Windows 3.1 was the first version of Windows to support the FIFO feature of the 16550 UART for Windows-based applications. (DOS apps running under Windows 3.1, 3.11, or Windows for Workgroups 3.1 don't support the FIFO feature.) However, Windows for Workgroups 3.11 does give DOS comm apps access to the FIFO feature. An upgrade for the Windows for Workgroups communications driver SERIAL.386 is available which can also use the transmit buffer (TX) of the 16550 UART. Previous versions of the communications driver use the receive buffer (RX) only. Check the Microsoft KnowledgeBase on the web for WG1001.EXE, which contains the updated driver.

You may be able to improve throughput without the need for new hardware, simply by adjusting a couple of Windows SYSTEM.INI settings. Incoming data is held in an internal buffer by COMM.DRV, the Windows 3.1x serial port driver, until it's retrieved by the comms program you're using. However, it's not always possible for the comms program to retrieve this data in a timely manner. If the buffer isn't cleared, then data can get overwritten. Luckily, it's possible to adjust the size of the buffer. By default, this is set to 128 bytes.

By simply adding a line to the [386enh] section of your SYSTEM.INI file, you can create a buffer as large as 10,000 characters. Actually, since the COMM driver maintains a separate buffer for each of your PC's COM ports, you can add up to four lines, each controlling the size of a particular port's buffer.

The syntax looks like this:

```
COMxBuffer=num
```

where x is the COM port number, i.e. a number between 1 and 4, and "num" the buffer size. Thus:

```
COM4Buffer=1024
```

sets the size of the buffer for COM 4 to 1,024 bytes. Don't forget that you've got to exit and restart Windows for the change to take effect.

How big should your COMM buffer be? If characters are being dropped from incoming data transmissions, create a new buffer of at least 2,048 bytes. If problems

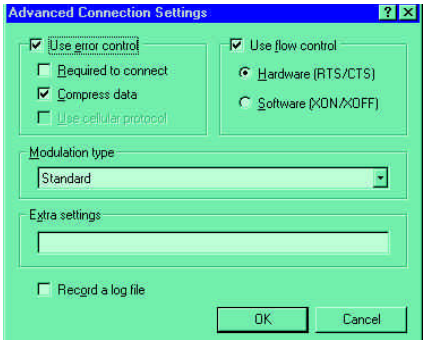
Universal Serial Bus (USB)

Luckily, in the nick of time, a solution to the inherent inadequacies of the serial port has arrived: the Universal Serial Bus (USB) which, by the end of the century, will probably replace both serial and parallel ports on PCs. With speeds of up to 12Mbits/sec, USB is aimed at simplifying and standardising the interconnection of PCs with peripherals such as modems, printers, mice, keyboards, digital speakers, joysticks, gamepads, telephones, telephone networks, scanners and digital cameras.

USB uses a tiny four-pin connector (two for data, one for power and one for ground). In fact, there are two types of connector: "Series A", a plug for a peripheral and a socket for a PC platform, is for applications permitting the cable to be moulded into its peripheral, while "Series B" is used for applications requiring a removable cable. The two-connector series is keyed differently to avoid mis-matching. PCs will most likely feature a USB "expansion hub" built into the keyboard or monitor, which will allow peripherals to be daisy-chained together — USB can connect up to 127 different devices to a single PC.

Instead of relying on the intelligence embedded in the host PC, the USB detects which devices are added or removed, automatically determines what host resources each peripheral needs, including driver software and bus bandwidth, and makes them available. Future versions of Windows 95 will be equipped with drivers that allow a PC to recognise USB peripherals. The plug-and-play feature enables users to attach or remove peripherals with the system running and without requiring the nuisance of rebooting. According to Dataquest, by the summer, 75 percent of the 22 million PCs manufactured worldwide will be equipped with USB ports.

p304 ➢



If you want reliable, high-speed modem connections it's important to make sure you're using hardware rather than software handshaking; use this dialog to choose which

persist, increase the size of the buffer to 4,096 or even 8,192bytes. Note also that the 16550A by default wakes up in 8250 mode and has to be switched in to 16550 mode via software. So if you've got a 16550 buffered serial port, you have to explicitly tell Windows 3.1x to use it. Add this line to the [386Enh] section of your SYSTEM.INI file:

```
COMnFIFO=On
```

where n represents the number of the COM port for which you are activating the buffer.

Another tweakable parameter is the COMBoostTime. This SYSTEM.INI entry specifies the time allowed a "virtual machine" to process a COM interrupt: if it's too short, you can lose characters. Its default value is 2 but you could try increasing it to 4. Once again, it's located in the [386enh] section:

```
COMBoostTime=4
```

Note: If you don't find it here, just add it. You can specify an even higher value but be conservative, since setting it too high can actually slow down communications.

Windows 95

Mercifully, Windows 95 makes optimising your modem's data throughput considerably easier than Windows 3.1x. Provided the modem is installed correctly, then there are only a couple of things to check. You should have received an INF setup file with your modem, which correctly describes all the modem's features for Windows 95. But if it didn't come with one and your modem doesn't feature in Windows 95's list of modems, you can always install it as a generic "standard" modem. This is OK-ish but not optimal, and if at all possible I'd recommend paying a visit to your modem maker's web site to see if proper Windows 95 INF files are available for your modem, for download.

Once installed, open Control Panel and click on the Modem applet. Highlight the modem and click the Properties button. On the General Tab set the maximum speed to four times that of your modem's speed: if you have a 28,800bps modem, set it to 115,200bps. Windows traditionally errs on the side of caution and always underrates the modem speed, typically setting the maximum speed to 57,600bps.

Next, click on the Connection tab and then on the Port Settings button. Make sure the Use FIFO buffers option is ticked and experiment by sliding the Receive buffer all the way up to 16. Click OK and then click on the Advanced button. Make sure the Use Error Control and Use Flow Control options are checked. Click OK several times to quit. And, er, that's it!

Rant corner — baud and bps

It's embarrassing, the number of otherwise knowledgeable people who use the terms baud rate and bits per second interchangeably. For example, the brand new CompuServe front-end, v3.0, continues to refer to "baud rate" when it really means bits per second. Baud actually refers to modulation rate: the number of times the line changes state every second. The difference between baud and bps is a result of the different forms of modulation used to encode digital data into analogue waveforms. At 300bps transmission rates, bps and baud happen to be the same; but above this speed, the 1:1 relationship ends. The maximum baud rate possible over normal phone lines is 2,400; the maximum bits per second rate is, at present, 33,600bps. So please stop using baud when you mean bits per second. Purleeze!

PCW Contact

Roger Gann can be contacted by post c/o *PCW* at the usual address, or via email at hardware@pcw.vnu.co.uk