

USB Port Management Guide

Whitham D. Reeve

1. Introduction

It often happens that when a new USB device, such as a USB-Serial Adapter, is plugged into a PC, its COM port is automatically set to a high number, say, 27. If the application program that is to use the device can only handle COM ports up to, say, 16, then you must manually adjust the device's port number. This is rather straight forward and is described in Part I of this article.

However, another problem can arise. You may notice that when you try to change the port number, Windows reports that the desired COM port already is in-use even though you have no other USB devices plugged into the PC. You can try to pick an arbitrary port number, but Windows will warn of a potential conflict. This may be handled as described in Part II by deleting COM ports that are marked "in use" but are not actually associated with an active device.

With the introduction of Windows 7, many USB-Serial Adapters stopped working, produced corrupted data streams or worked with some application programs but not others. This invariably is caused by out-of-date device drivers or, in some cases, incompatible or obsolete chipsets in the USB-Serial Adapter. This is addressed in Part III.

Finally, serial port interface wiring always has been a huge problem, especially when hardware flow control is involved. This is briefly discussed in Part IV and a reference is given for various wiring combinations of both the DB9 and DB25 interface connector.

2. Port Assignment Example

Port conflicts occur because Windows assigns different COM port numbers to each USB device in each USB jack. This assignment remains even when the devices are unplugged (not present). For example, if your laptop has three physical USB jacks, and if a single USB device is plugged into each one of them in succession, Windows will assign three different COM port numbers (figure Ex-1). Even after the device is unplugged, the COM ports remain assigned to their previous values (figure Ex-2).

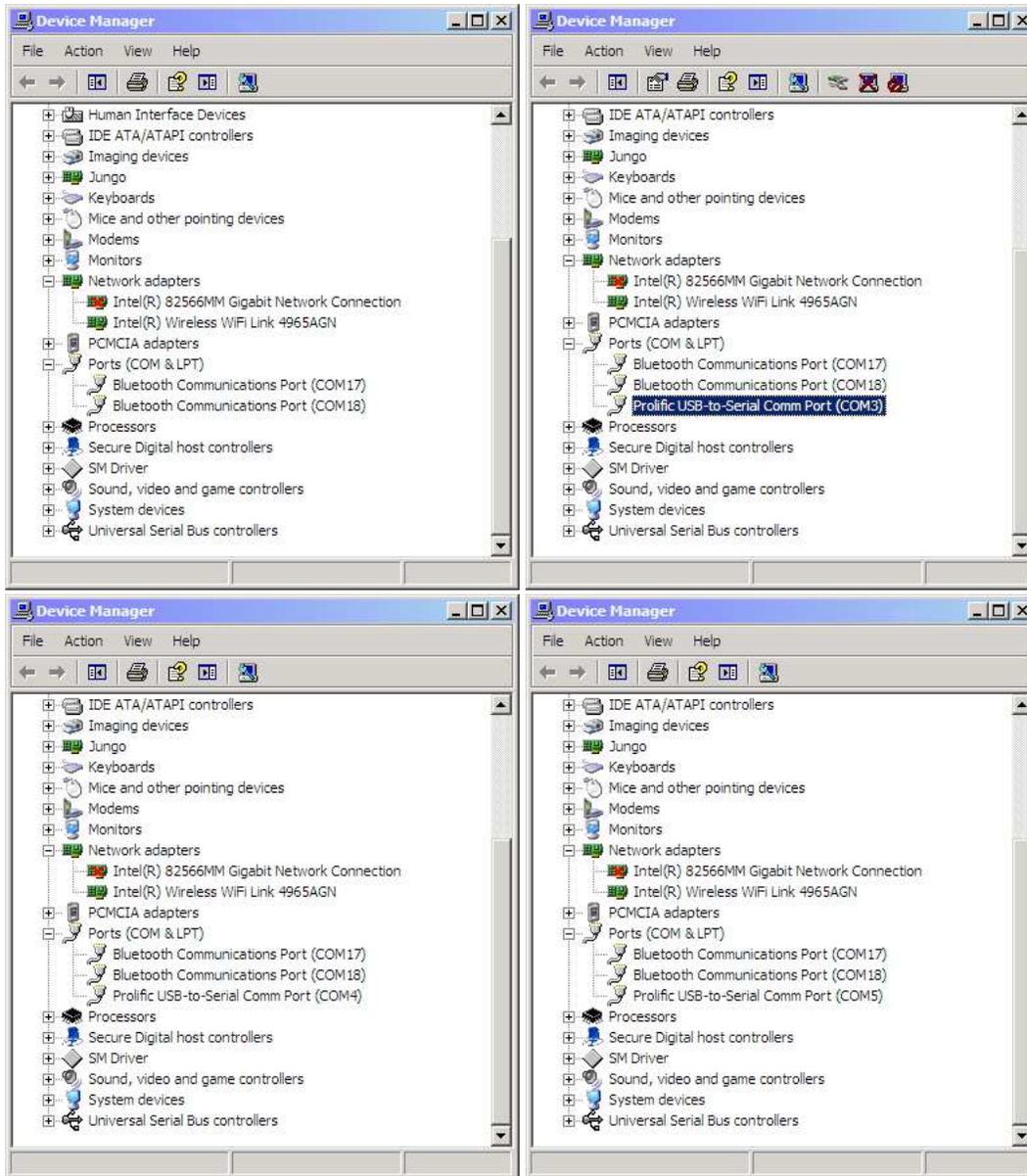


Figure Ex-1 ~ A USB-Serial Adapter was plugged into each of three USB jacks on a laptop. The first screen (upper-left) shows the Ports in Device Manager prior to plugging in the adapter. The next screen (upper-right) shows that the device was assigned to COM3 when it was plugged into the first USB jack. The device was unplugged from the first jack and plugged into the second jack (lower-left) and then the third jack (lower-right) and Windows assigned COM4 and COM5.

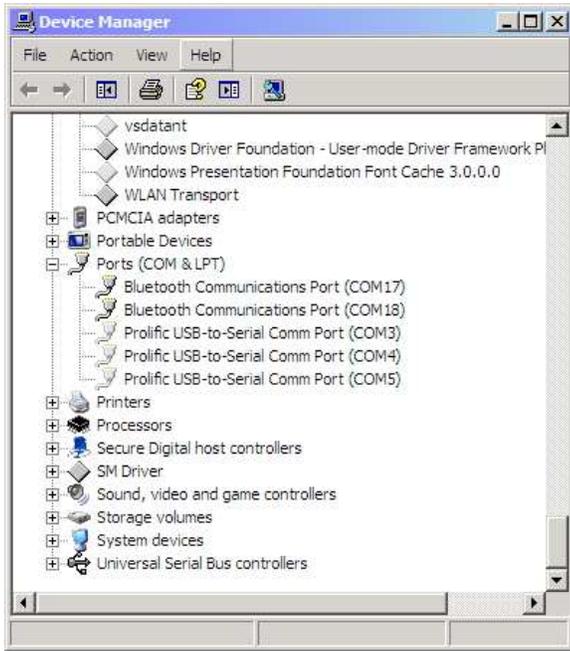


Figure Ex-2 ~ After the USB-Serial Adapter was unplugged, the COM ports stay assigned. If the same device is plugged back into a given port, the previous COM port assignment will be used. In the screen shown here, **Device Manager** was configured as described in Part II to show “non-present” (hidden) COM ports.

Part I ~ Reassign COM Ports to USB Devices

Over time, as different USB devices are plugged into different USB jacks on a PC, Windows assigns COM ports sequentially to avoid conflicts. Eventually, many COM ports are assigned even though the devices are unplugged. Some of these devices may no longer be used or were discarded long ago. If a new USB device is installed, and if Windows assigns a COM port outside the range of the application program that uses that device, it is necessary to manually reassign the COM port to within the usable range as described below. Before following these steps, unplug all unused USB devices.

1. Open **Device Manager** by right-clicking **My Computer** and selecting **Properties**. The **System Properties** window will open (figure 1.1-left).
2. Select the **Hardware** tab and click the **Device Manager** button (figure 1.1-right).

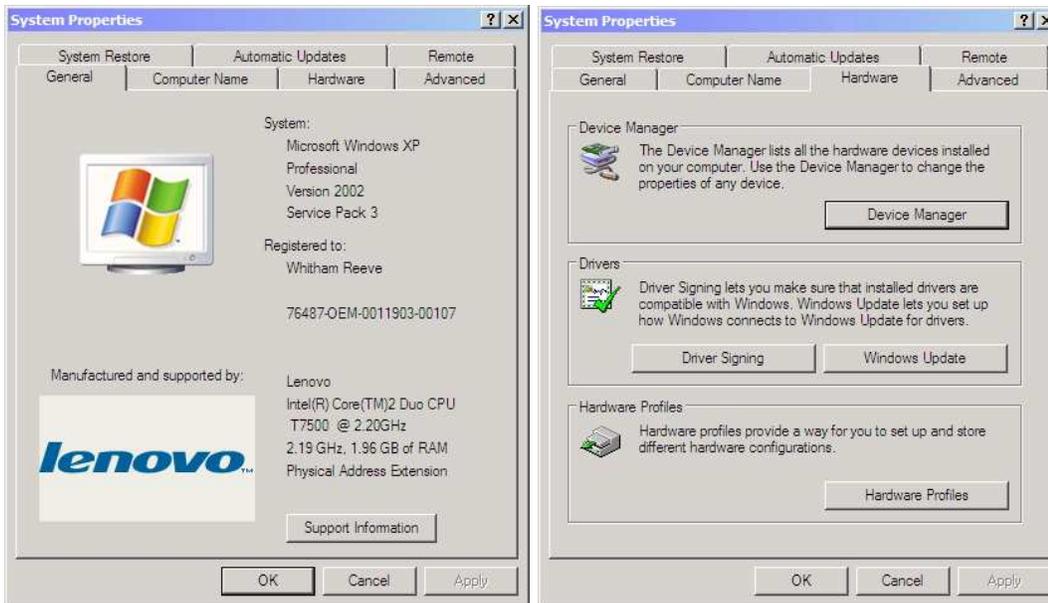


Figure 1.1 ~

Left: **System Properties** window.

Right: **System Properties Hardware** tab.

3. Scroll down to the **Ports** section and click the **+** to expose the various COM port assignments (figure 1.2-left). To identify a given device, unplug it and then plug it back into the USB jack and note which COM port disappears and then reappears in the list. Make note of the COM ports you wish to retain.
4. As an example, change the COM port assignment for the USB-Serial Adapter, marked *Prolific USB-to-Serial Comm Port (COM3)*, from COM3 to COM4. Right-click that item and select **Properties** (figure 1.2-right).

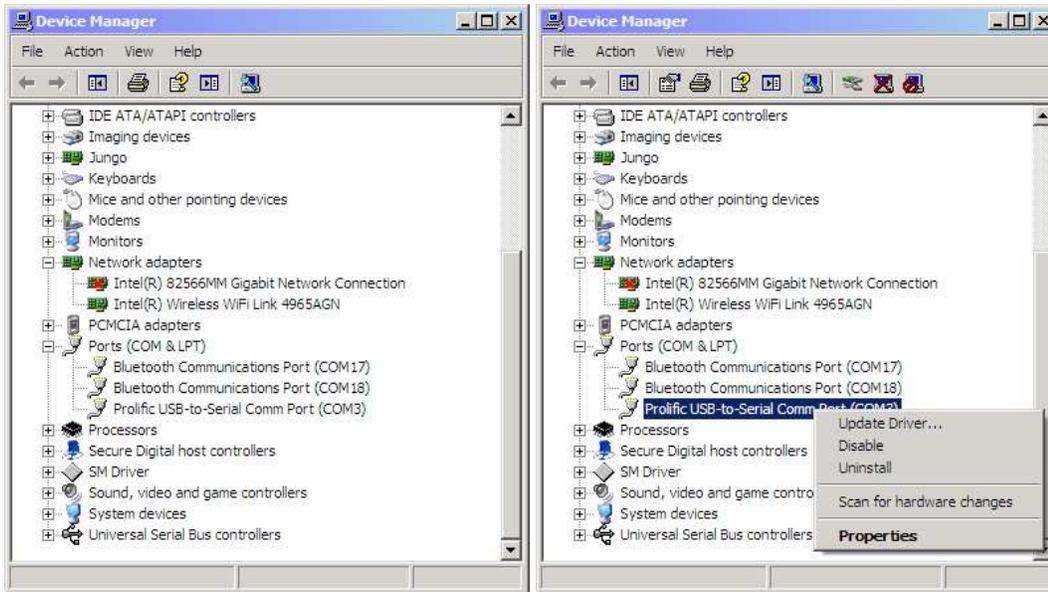


Figure 1.2 ~
Left: Device Manager with COM Ports exposed. In this example, Windows has assigned a Bluetooth device to COM17 and COM18 and a USB-Serial Adapter to COM3.
Right: Right-click a Port (COM3 in the example) to select its Properties.

5. The device's **Properties** window will open. Select the **Port Settings** tab (figure 1.3).

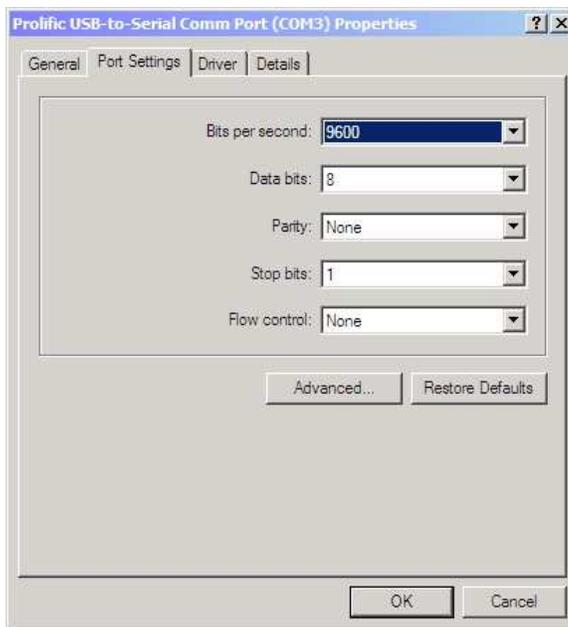


Figure 1.3 ~ Device **Port Settings** tab.

6. Click on the **Advanced...** button (figure 1.4).

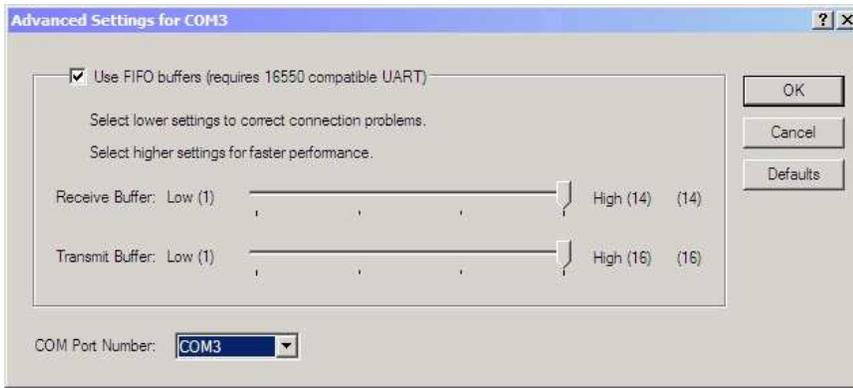


Figure 1.4 ~ **Advanced Settings** for selected COM port.

- The existing COM port assigned to this device is shown at the bottom of the **Advanced Settings for COMx** window. Click on the **COM Port Number** drop-down box, and you will see a list of COM ports and their existing assignments. If all ports in the desired range (typically 1 ~ 16) are marked “in use”, select one from step 3 that you do not wish to retain or you did not identify. In this example, COM4 is selected. It is shown in the list as “in use” (figure 1.5).

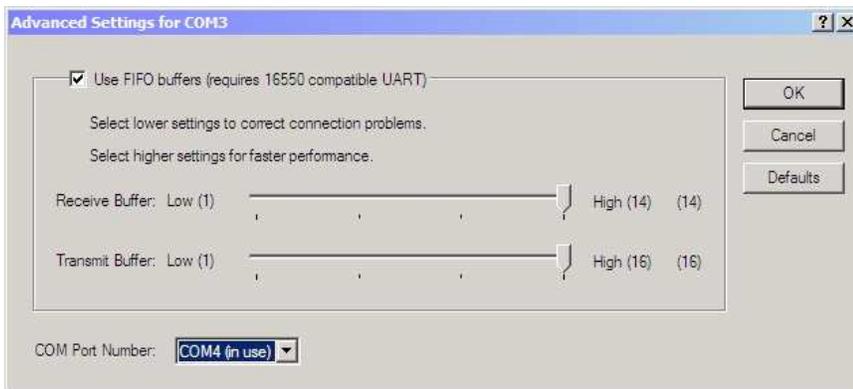


Figure 1.5 ~ Select a different COM port in the **COM Port Number** drop-down box. It may be marked “in use” even though no active device is connected to that port.

- Click the **OK** button. If the port was marked “in use”, a **Communications Port Properties** dialog window with a warning will pop up (figure 1.6).

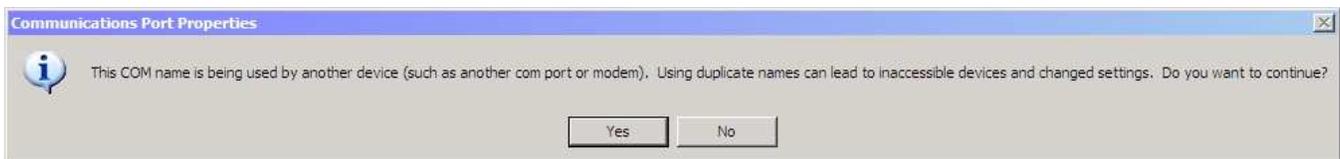


Figure 1.6 ~ **Communications Port Properties** dialog window warns that the selected COM port already has been assigned by Windows. This does not mean it is actually in use, only that it has been assigned.

- Click the **Yes** button and, after a moment, the warning window will close. Click **OK** on the device’s **Properties** window.
- Unplug the USB device and plug it back in. Check **Device Manager** and it will show the device has been assigned to selected COM port (COM4 in this example) (figure 1.5). In some cases, it may be necessary to restart the PC for it to recognize the device on the reassigned COM port.

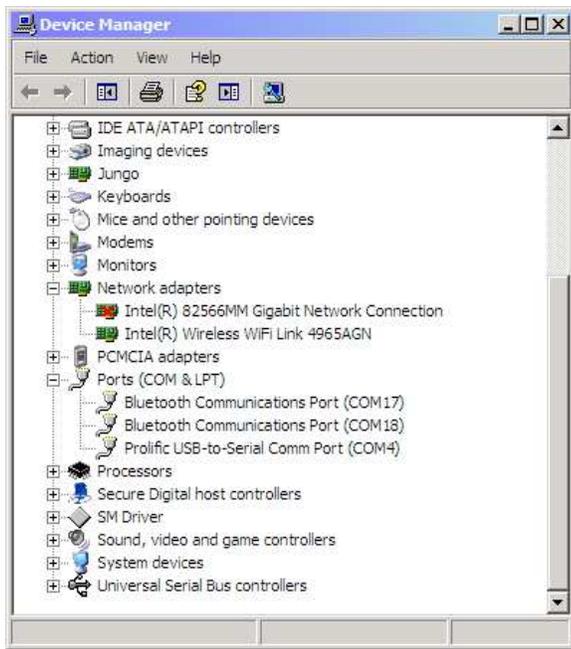


Figure 1.5 ~ When the USB device is unplugged and then plugged back in, Windows will recognize the device and its assignment to the selected COM port.

11. To delete unused and hidden COM ports, go to Part II.

Part II ~ Delete Unused USB Ports

2.1 USB Devices

As already mentioned, Windows remembers previous COM port assignments, even for devices that are not plugged in or that might no longer exist. To reduce the possibility of conflict and problems, it is worthwhile to delete all unused COM ports before installing a new USB device. If by chance you delete a port that is needed some time later, Windows will automatically re-install the port driver at that time (but it may be installed on a different port than before and the application that uses the device may need to be reset to the new port). The procedures given in section 2.1, allow hidden devices to be viewed until the PC is rebooted. If desired, the viewing can be made persistent as described in section 2.3.

2.2 Temporarily View USB Devices

Unused devices normally are hidden in the **Device Manger** window and must be made visible before they can be deleted. You will notice in **Device Manager** that if you click **View – Show hidden devices**, the **Device Manager** window shows many previously hidden devices, but it does not show all unused COM ports. To expose and then delete hidden COM ports, follow the steps given here. These steps do not provide a persistent display of all hidden and unused devices; the settings disappear when the PC is rebooted. Additional steps are provided later to permanent enable display of hidden and unused devices.

1. Open the **Command Prompt** window by clicking **Start – Run....** In the **Open:** box type **cmd** and click **OK** (figure 2.1).



Figure 2.1 ~ Run dialog box and entry to open the **Command Prompt**.

2. At the command prompt, type **set devmgr_show_nonpresent_devices=1** and press the Enter key (figure 2.2).



Figure 2.2 ~ **Command Line** dialog box and command to show non-present devices.

Note: In some Windows environments it may be necessary to change to the System32 folder before performing the next step. To do this type **cd\windows\system32** and press the **Enter** key.

3. Type **start devmgmt.msc** and press the **Enter** key (figure 2.3).

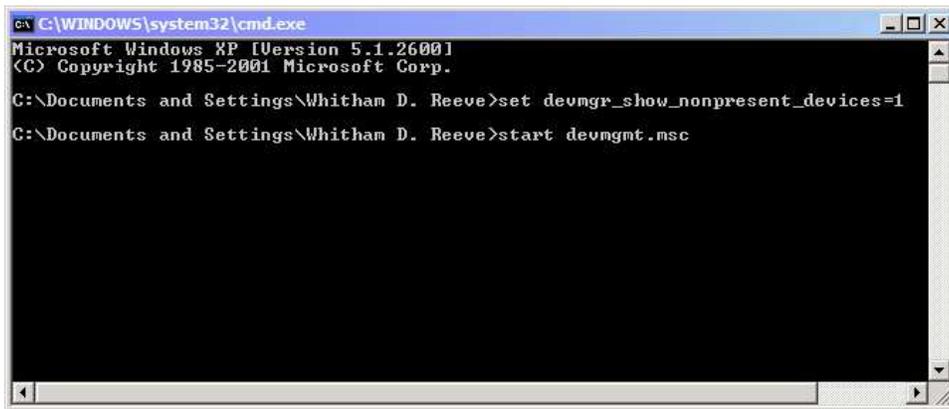


Figure 2.3 ~ **Command Line** dialog box and command to start the **Device Manager**.

4. When the **Device Manager** window opens, click **View – Show hidden devices** (figure 2.4)

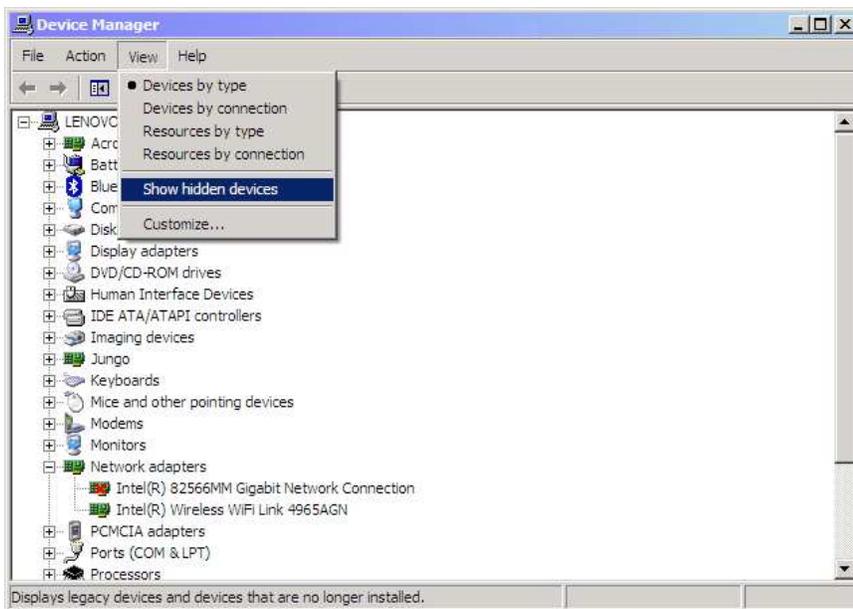


Figure 2.4 ~ Device Manager being set to Show hidden devices.

5. Scroll down to the Ports devices and click on the + sign to see the full list of COM ports in use and the USB devices that are not present and were previously hidden (figure 2.5).

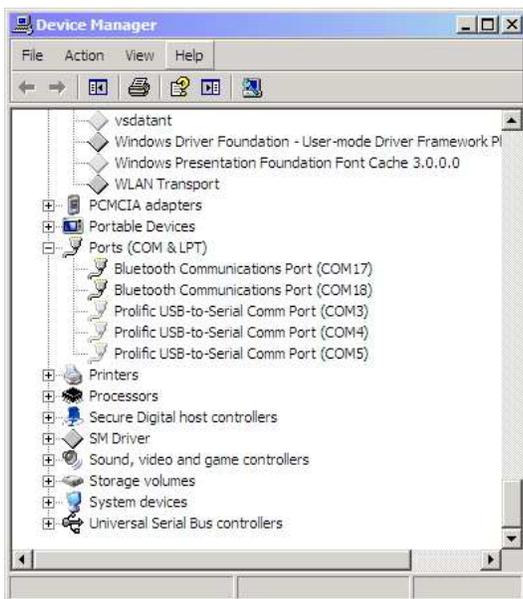


Figure 2.5 ~ Device Manager shows all Ports including those previously hidden.

6. Highlight the port you wish to delete and then press the **Delete** key. Repeat as necessary to eliminate additional unused COM ports.

2.3 Permanently View Hidden USB Devices

The changes described above to view hidden devices are not persistent. If it is necessary to delete additional devices after a PC reboot, the procedures will have to be followed again. To permanently change the Device

Manager to allow display hidden devices, follow the steps below. The screens and procedures are for Windows 7 but are similar for Windows XP.

1. It is necessary to make the changes in **Safe Mode** while logged in as Administrator. Restart the PC and press and hold the **F8** key or press it repeatedly while the BIOS startup is displayed but before Windows starts to load. Eventually, a list of startup modes will appear. The screen most likely will be displayed at lower resolution than in **Normal Mode**.
2. Select **Safe Mode** and login as Administrator. When the **Safe Mode** screen appears, click Start, right-click **My Computer** and select **Properties**. The **System** window will appear (figure 2.6).



Figure 2.6 ~ System window

3. Click on **Advanced system settings**. The **System Properties** window will open (figure 2.7).

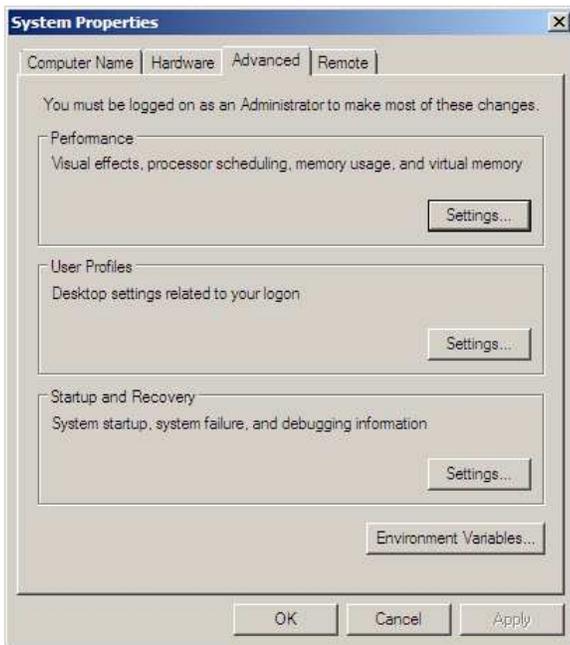


Figure 2.7 ~ System Properties window.

4. Click on the **Environment Variables** button. The **Environment Variables** window will open (figure 2.8).

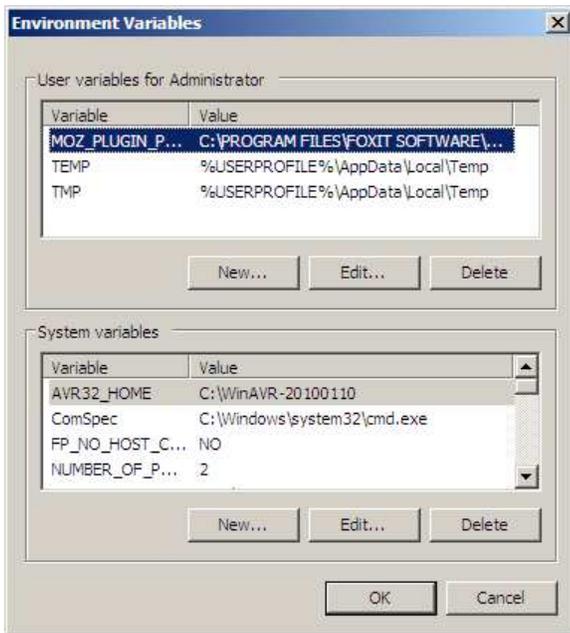


Figure 2.8 ~ Environment Variables window.

- Under the **System variables** frame at the bottom, click **New...** In the **Variable Name** field of the New System Variable window (figure 2.9) enter **DevMgr_Show_NonPresent_Devices**, and in the **Variable Value** field enter **1**. Click **OK**.



Figure 2.9 ~ New System Variable window.

- Check that the new **Environment Variable** is displayed in the **System Variables** window. When satisfied, click **OK** to close the **Properties** dialog window and **OK** again to close the **Environment Variables** window.
- Once back at the **Desktop**, restart the PC. The PC should automatically start in **Normal Mode**.
- Navigate to the **Environment Variables – System variables** window as above and check that the **DevMgr_Show_NonPresent_Devices** variable is displayed (figure 2.10). Return to the **Desktop**.

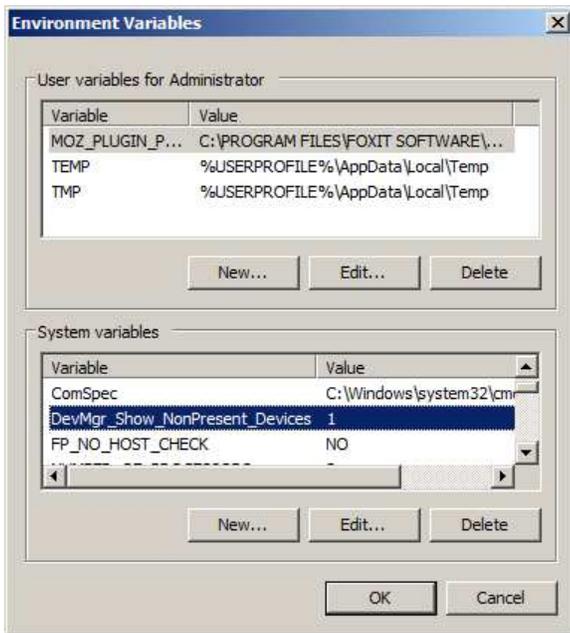


Figure 2.10 ~ Environment Variables window with changes.

- To view USB-Serial Adapter devices, at least one adapter must be plugged into a USB port. Open the **Device Manager** and scroll down to the **Ports (COM & LPT)** devices. Click on the + sign to see the COM port presently used by the adapter you plugged in (figure 2.11).

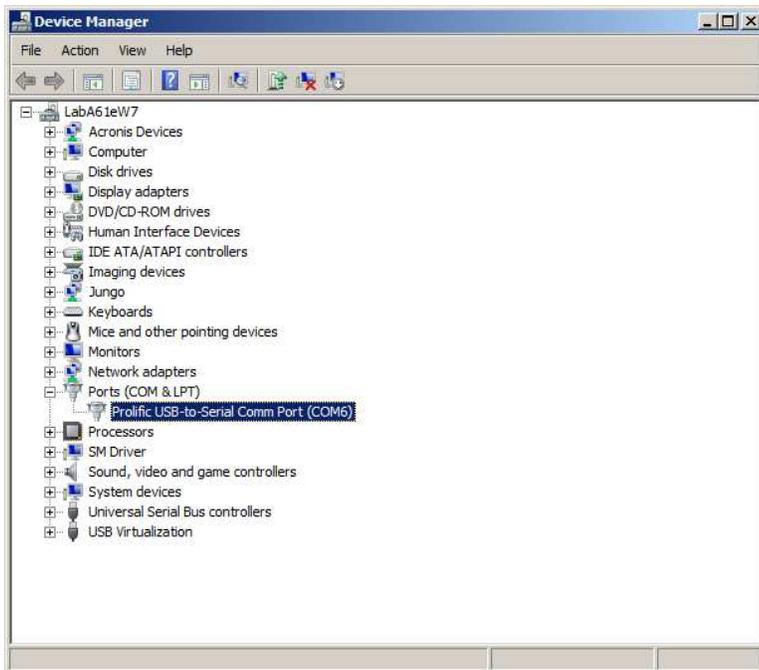


Figure 2.11 ~ Device Manager window.

- Now, click on the **View** menu and then click on **Show hidden devices**. The **Ports** list should be much longer and can be accessed anytime without going through the command line interface. Devices can be deleted as previously described.

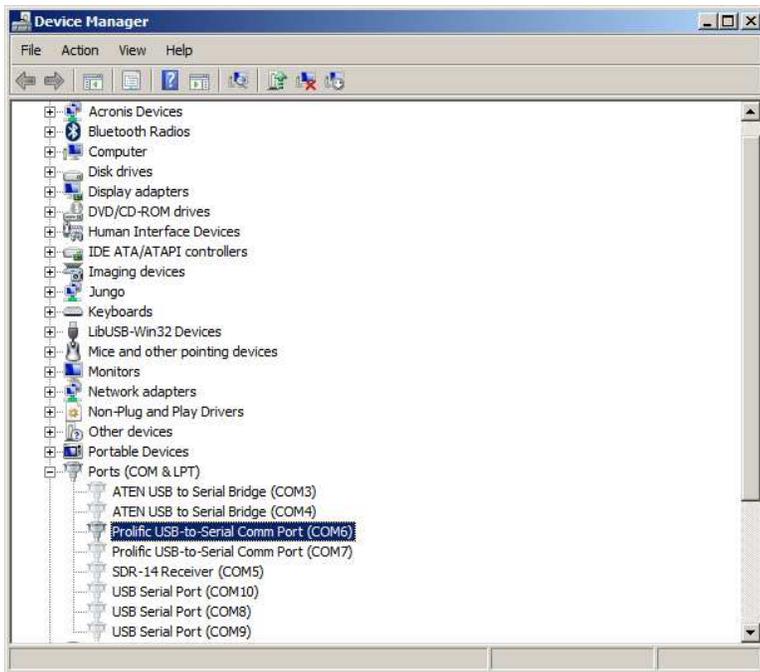


Figure 2.12 ~ Device Manager window with Hidden Devices exposed.

Part III ~ USB-Serial Adapters and Associated Device Drivers

3.1 Problems That May Be Observed

A number of problems may be observed when running application programs on Windows 7 that previously worked well on Windows XP:

1. Windows updates have made drivers inoperable that previously worked with the program
2. USB-Serial Adapter devices and drivers that worked with application programs at lower speeds (for example, 9600 b/s) do not work at a higher speed (for example, 115200 b/s)
3. Older USB-Serial Adapters that worked with Windows XP do not work with Windows 7 even though the driver may be current or claimed by the manufacturer to be Windows 7 compatible
4. A USB-Serial Adapter may work on a Windows 7 x32 PC but not on a Windows 7 x64 PC
5. Nothing happens when the application is opened – the application software user interface window is not visible on the Windows desktop. Subsequent attempts to open the application do not work, resulting in one or more instances seen in the Windows Task Manager (called “zombies”)
6. Opening the application the first time after PC restart or plugging in an adapter may not be successful (the application software user interface window is not visible), resulting in a “zombie” instance (figure 3.1)

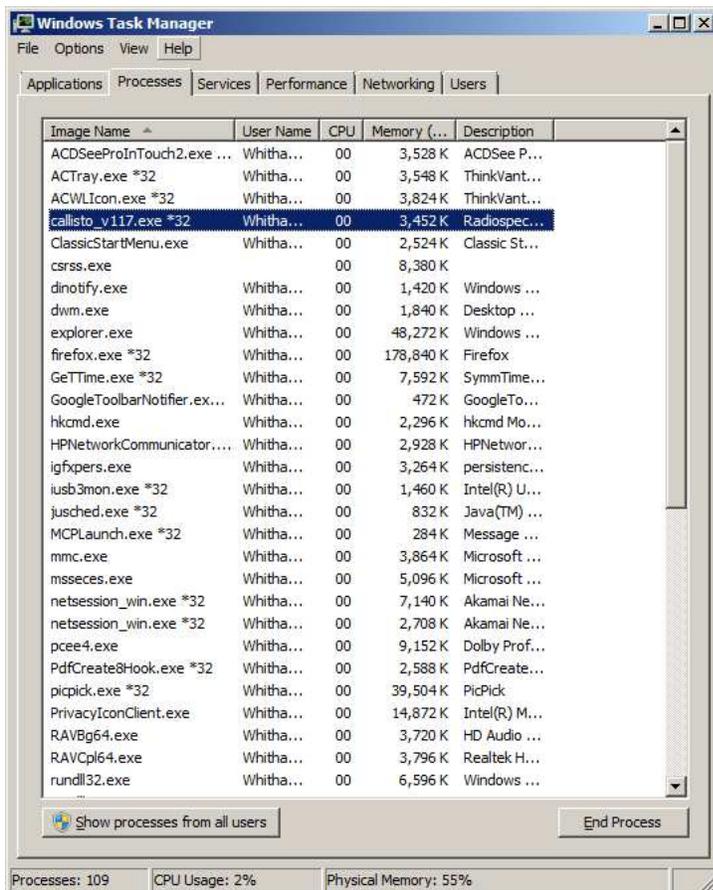
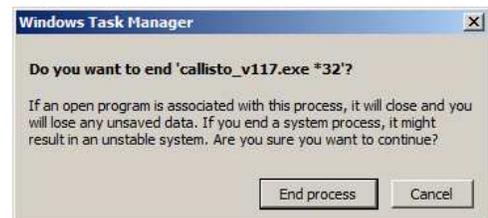


Figure 3.1 ~ Left: In this example, the callisto.exe application is opened but its window does not appear on the desktop. It is seen in Task Manager as a “zombie”. Below: Task Manager is used to close (End Process) the application (callisto.exe in this example).



3.2. USB-Serial Adapters that have been tested with Window 7

Adapters that I have tested with Windows 7 x32 and x64 are tabulated (table 1). Two adapters known to be incompatible are included at the end of the list. Note that only adapters using the FTDI and Prolific chips were tested. In all cases, the latest device drivers were installed as described in this part. Other chip manufacturers exist (for example, MOXA and MicroChip) but have not been tested.

Table 1 ~ Tested USB-Serial Adapters

Manufacturer	Model Number	Chip	Website	Remarks
StartTech	ICUSB2321F	FTDI	www.startech.com	Note 1
USConverters	XS880	FTDI	www.usconverters.com	Ultimate, Note 2
USConverters	XS8801	FTDI	www.usconverters.com	Professional, Note 2
USBGear	USBG-10ft-232-STF	FTDI	www.usbgear.com/	Note 2, 3
Plugable	USB Serial Converter	Prolific HXD	plugable.com/	
Trendnet	TU-S9 V2 (white)	Prolific HXD	www.trendnet.com/	
Trendnet	TU-S9 V1 (blue)	Prolific HXD		Incompatible, Note 4
IOGear	GUC232A	Prolific HXA		Incompatible, Note 5

Table Notes:

1. The "F" in the model number indicates FTDI chip. StarTech supplies several versions of the ICUSB232 adapter, including some with different chips, but only the model number shown was tested
2. These adapters are equipped with Rx, Tx and power LEDs, which are very helpful for troubleshooting
3. The model shown has a 10 ft cable; a similar model with 6 ft cable, USBG-6ft-232-STF, is available. The "F" in the model number indicates FTDI chip
4. The Trendnet TU-S9 V1 (blue) is an older model and is not recommended even though it uses the HXD chip
5. The IOGear GUC232A is an older model with an incompatible chip (HXA) and ATEN drivers and is not recommended

3.3. USB-Serial Adapter Device Drivers

If the adapter was obtained prior to 2010, it probably will not work with Windows 7 even though a Windows 7 x32 or x64 driver may be available. A web search or the manufacturer's website may yield the necessary compatibility information. However, many adapters are unbranded, making it impossible to find reliable information on them and, for this reason, are not recommended.

Before attempting to update, troubleshoot or repair USB-Serial adapter problems, remove all unused drivers using the procedures described in Part II of this article. Next, determine the chipset used in the adapter. This usually can be found by examining the Device Manager as follows:

1. Unplug all adapters except the one being investigated
2. Right-click My Computer, click on Properties and then select the Device Manager
3. Click the + next to Ports (COM & LPT) to expose the adapter
4. Right-click the adapter, select Properties and then select the Driver tab. The driver provider should be indicated (figure 3.2)



Figure 3.2 ~ Device Driver Properties – Driver tab for a USB-Serial Adapter.

- Alternately, click the Details tab and select Manufacturer and then Driver Assy Version from the drop-down (figure 3.3)

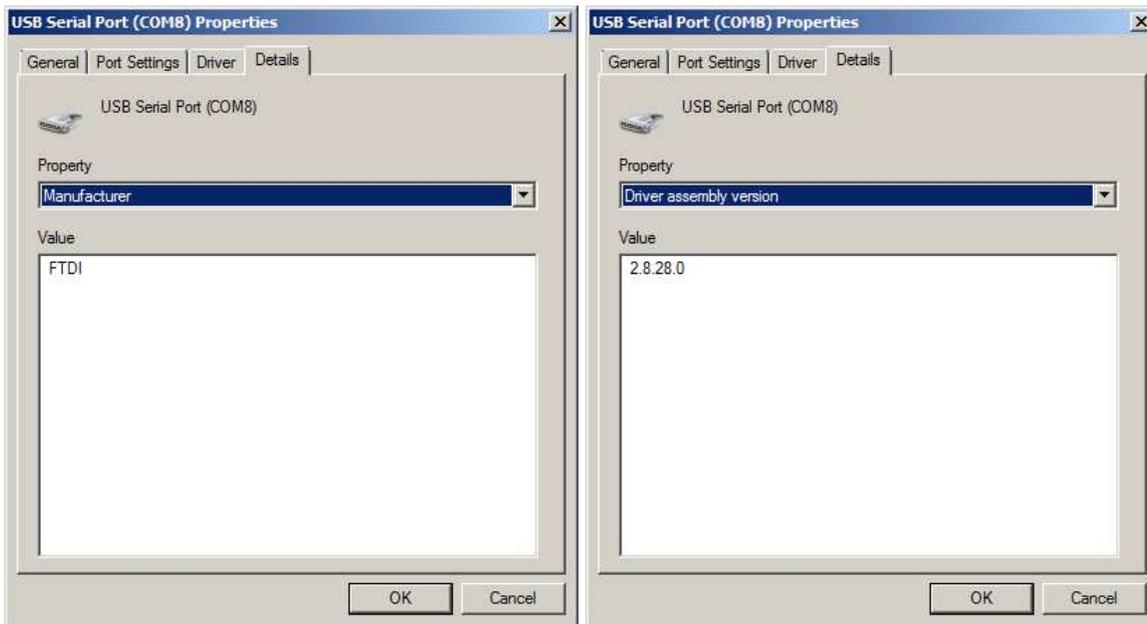


Figure 3.3 ~ Device Driver Properties – Details tab for a USB-Serial Adapter

- Make a note of the driver date and version. If the driver is dated prior to 2013, then it is out of date and likely will not work properly under Windows 7. However, it is not guaranteed to work even if it is dated 2013.

Websites for FTDI and Prolific drivers are provided in the next section. Prolific provides a program with their driver software that identifies the chipset used (figure 3.4). This may indicate an older, incompatible chip.

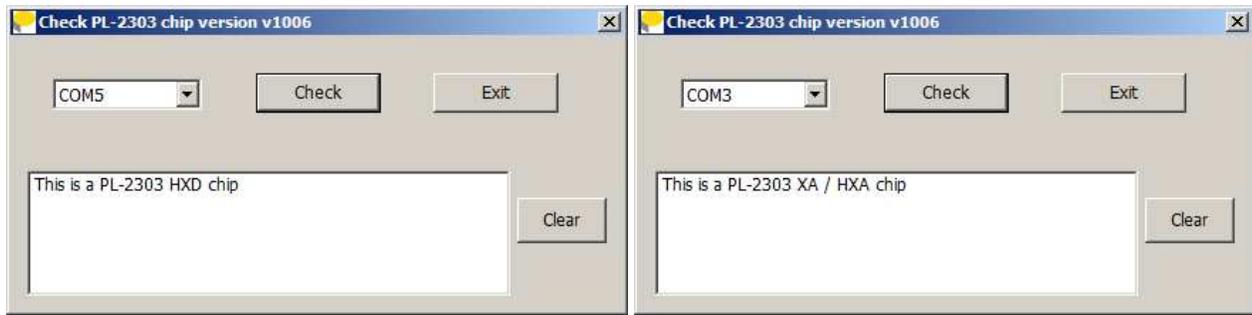


Figure 3.4 ~ Prolific chip identification tool. Left: If the chip is identified as HXD, it is compatible with Windows 7 when used with the latest Windows 7 device driver. Right: If it is identified as any other chipset, such as HXA, it is incompatible even if the manufacturer claims the driver is Windows 7 compatible.

3.4. Driver Updates

The USB-Serial Adapters that were tested use either FTDI or Prolific chips. The respective websites for the drivers are:

<http://www.ftdichip.com/FTDrivers.htm>

<http://prolificusa.com/pl-2303hx-drivers/>

Do not attempt to update device drivers from the Device Manager > Driver tab. It has been found that this method often does not find the latest drivers, so avoid using it. The recommended procedure is to visit the website of the adapter chipset manufacturer and obtain the driver directly.

Some driver update programs will first uninstall the old driver. This means the update program must be run twice – first to remove the old driver and second time to install the new driver. When the new driver is installed, the update program may indicate the PC must be restarted. Do not skip this step.

If the new driver does not automatically uninstall the old one, then the old driver must be manually uninstalled. This may be done from the Device Manager, Driver tab. The USB-Serial Adapter must be plugged in for the Uninstall process.

Some driver installation programs require that the USB-Serial Adapter be plugged in when running the update program. When a new adapter is plugged in, and no driver exists for its chip, Windows 7 will attempt to automatically install the needed drivers, and this can lead to an out-of-date driver. Allow Windows to finish its process and then go to the Device Driver Properties and select the Driver or Details tab to check the driver version. If a newer one can be found on the adapter manufacturer's website, then uninstall the existing driver and install the new driver.

After installing the new driver and restarting the PC, go once again to the Device Manager and check the driver date and version. It should correspond to the information on the driver provider's website.

After the correct, up-to-date driver is installed for a given type of chip, it can be applied to any USB-Serial Adapter that uses that chip. When a new USB-Serial Adapter is plugged in, Windows 7 should automatically use the new driver. However, if that adapter was previously installed, Windows 7 may use the old drivers if they still exist. Otherwise, it will use the new driver. If Windows 7 selects the old driver, then follow the above procedures to install the new driver for that device.

The driver update process as described above needs to be repeated for each USB-Serial Adapter.

Part IV ~ EIA-232 Serial Port Interface Wiring

4.1 DTE and DCE

The EIA-232 (formerly RS-232) serial communications interface is used with data terminal equipment (DTE) and data communications equipment (DCE). The signaling and flow control directions of DTE and DCE are complementary. A PC is DTE and a modem is DCE. A PC that uses a USB port for serial communications requires a USB-Serial Adapter and a corresponding virtual communications port driver. In this case, the USB-Serial Adapter is simply a hardware extension of the PC's virtual DTE. The interfaces on other devices may be DTE or DCE. A DTE may be connected to another DTE and a DCE may be connected to another DCE through a null modem cable.

All modern USB-Serial Adapters use a DB9-M connector and their drivers support hardware and software flow control. Many interfaces to be connected to the USB-Serial Adapter do not require hardware flow control and use only 3 leads, in which case the USB-Serial Adapter pin 2 is TD (transmit data, or outgoing), pin 3 is RD (receive data, or incoming) and pin 5 is common (figure 4.1). Where a null modem cable is required for a 3-wire setup, pins 2 and 3 are simply reversed at one end of the cable.

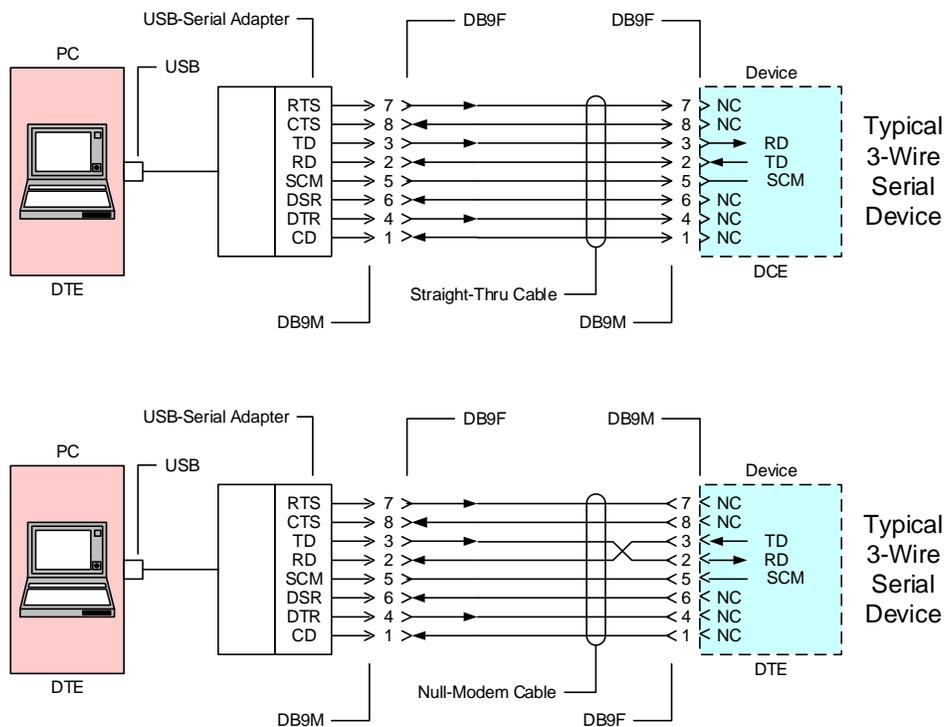


Figure 4.1 ~ Interface wiring diagram for connecting typical devices to a USB-Serial Adapter with a straight-thru cable (upper) and null modem cable (lower). In the null modem cable, pins 2 and 3 are reversed at one end, and the cable usually has DB9F connectors at both ends.

When hardware flow control is involved, the interface wiring is more complicated. Readers are referred to: http://www.reeve.com/Documents/Articles%20Papers/Reeve_SerialPortAdapters.pdf, which shows serial port wiring diagrams for many different types of equipment with and without flow control. The referenced document has not been updated to include new equipment since 2008.

Document information:

Author: Whitham D. Reeve
Copyright: © 2012, 2015, 2016 W. Reeve
Issue: 0.0 (Draft started, 28 October 2012)
0.1 (Reformat and minor edits, 30 October 2012)
1.0 (Issued for distribution, 30 October 2012)
1.1 (Minor edits, 8 November 2012)
1.2 (Added section on permanent display of hidden devices, 8 Nov 2015)
2.0 (Added Part III, taken from Callisto investigations paper, 23 Jan 2016)
2.1 (Added Part IV, 24 Mar 2016)