Using *com0com* and *hub4com* to Distribute a GPS Data Stream to Multiple Apps on a PC

Al Taylor, KN3U
February 21, 2025

Introduction

com0com and *hub4com* are Windows utility programs that can be used to distribute a GPS data stream to multiple apps on a Windows PC.[1] These open-source programs have been placed in the public domain under a GPL license and may be freely downloaded for use. On several of my PCs, I use these utilities to distribute data from an inexpensive GPS "hockey puck" receiver[2] to *NMEAtime*, *Winlink Express*, and *N1MM Logger+*. If you use other Windows apps that require GPS data, you can easily add them.

It can be confusing to understand how these program work and how to set them up. I hope this document will help clarify the process.

In my system, when I first plugged my GPS receiver into a USB port, Windows recognized a new serial device, found an appropriate driver, and assigned COM7 to it. So in my system, COM7 is the source for the GPS data stream. In your system, the GPS receiver will probably be assigned a different COM port, and you will have to identify it using the Windows Device Manager or some other means. Keep that in mind as we proceed through the explanation.

How *com0com* works

com0com is used to create any number of virtual COM ports in pairs. All such ports are accessible to programs running on the PC via Window COM port drivers. The outputs of one port in a pair are virtually connected to the inputs of the second port and vice versa. After installation of the *com0com* software, you run the program one time to establish the virtual COMs ports you need. They are saved in the Windows Registry and available whenever Windows starts. The virtual serial ports appear in Device Manager, as shown in Figure 1 on the following page.

com0com is a versatile tool. It has other capabilities that will not be covered in this document.

---

[1] VSPE, from Eterlogic Software, is an alternative to using *com0com* and *hub4com*. The 32-bit version of VSPE is free.

[2] See a companion document, available at https://mcacs.net/training-resources/gps-hockey-puck/ to learn more about these devices.

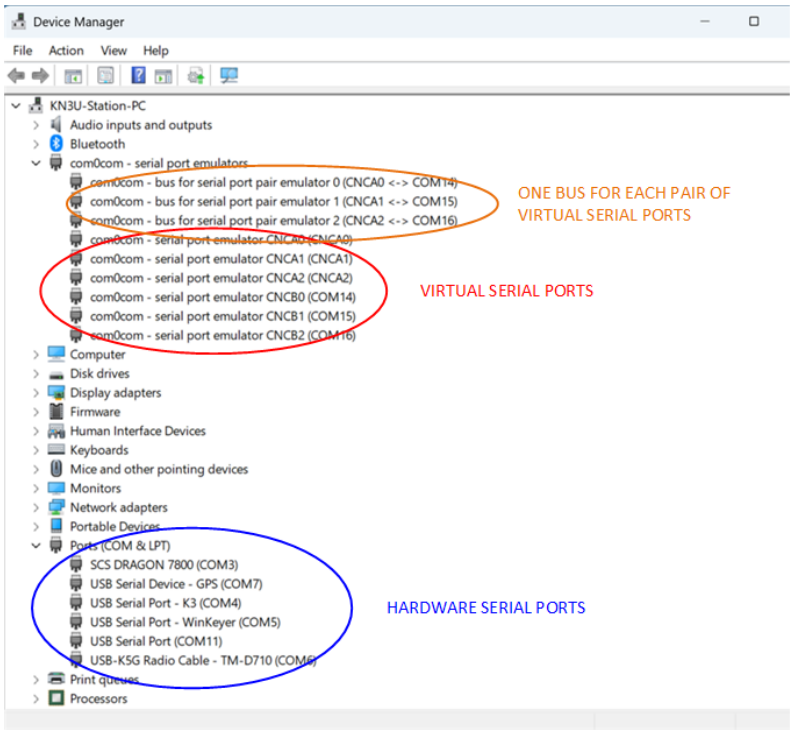

Figure 1 Device Manager screenshot showing real and virtual serial ports

The screenshot in Figure 2 shows the three virtual com port pairs I created using the *com0com* setup utility. One port in each pair is numbered like a standard COM port, specifically, COM14, 15, and 16. I someone arbitrarily chose those COM port numbers because COM14 is well beyond the number of "real" hardware COM ports in my system. That ensures that none of my virtual COM ports will conflict with a hardware COM port, now or in the future. All three of these virtual COM ports, however, appear as regular serial ports in Windows and can be accessed by any Windows app.

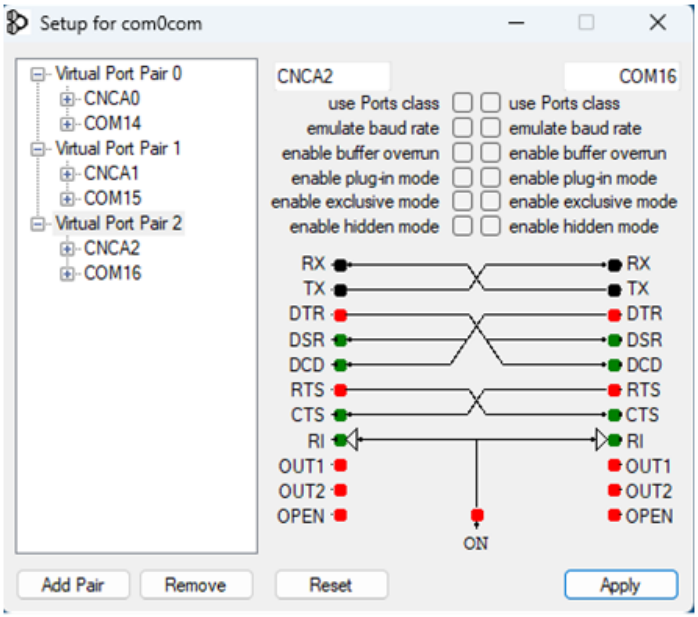

Figure 2 *com0com* setup screen

In my system, again somewhat arbitrarily, I configured *NMEAtime* to look for GPS data from COM14. I configured *N1MM Logger +* to look for GPS data from COM15, and configured *Winlink Express* to get its GPS data from COM16.

Note that the second COM port in each virtual pair is numbered CNCA0, 1, and 2. Somehow, we have to route the incoming GPS data stream to CNCA0, 1, and 2. That, in turn, will make the same GPS data available to *NMEAtime*, *Winlink Express*, and *N1MM Logger +* through their respective COM ports. Splitting the GPS data into multiple streams and routing them to CNCA0, 1, and 2 is the job of the *hub4com* utility.

How *hub4com* works

Unlike the *com0com* setup program, which needs to be run only once, the *hub4com* utility needs to be invoked each time Windows is started. I wrote a Visual Basic script to accomplish this. I saved this script in my Windows startup folder so that *hub4com* runs in the background each time Windows starts. The script instructs *hub4com* to accept the incoming GPS data stream from COM7 and route a copy to each of the virtual serial ports CNCA0, CNCA1, and CNCA2. Then *com0com* takes over and delivers the data to my apps. Figure 3 provides an overview of the entire process.

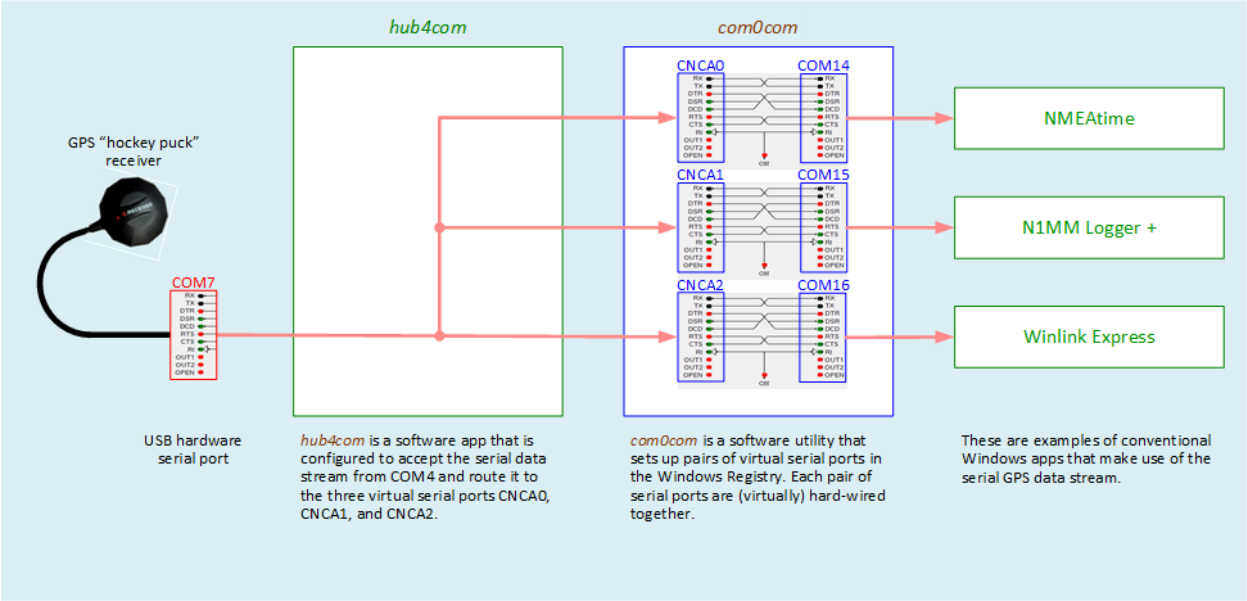

Figure 3 Overview of the entire process

hub4com Startup Script

The startup script is called GPS_Split.vbs. it runs the utility *hub4com* and instructs it to send the incoming GPS data stream on COM7 to virtual COM ports CNCA0, CNCA1, and CNCA2. The script is shown below.

```
command = "cmd /c ""C:\Program Files (x86)\hub4com\hub4com.exe"" --baud=19200 " & _
"--octs=off " & _
"--no-default-fc-route=All:All " & _
"--route=All:All \\.\COM7 \\.\CNCA0 \\.\CNCA1 \\.\CNCA2"
Set WshShell = CreateObject("WScript.Shell")
ReturnCode = WshShell.Run(command, 0, True)
Set WshShell = Nothing
```

I saved it to the Windows startup folder, which in both Windows 10 and 11 can be found here:

C:\Users\[*your username*]\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup

COM7 is the COM port that was assigned to the GPS receiver when it was installed on my computer. Change COM7 in the script to the COM port on your PC that is assigned to the GPS receiver. You can edit the script using Notepad or any other text editor.

Step-By-Step Instructions

After that somewhat convoluted explanation of how *com0com* and *hub4com* operate, actually setting up these utilities is relatively straightforward provided that you have some experience setting up regular COM ports in Windows. Here are the steps:

1. Download the latest versions of *com0com* and *hub4com* installation packages from SourceForge at the following link: https://sourceforge.net/projects/com0com/files/. If offered a choice between the signed and unsigned versions, choose the signed version.

2. Run the *com0com* installation program. Accept the default installation folder and, when prompted, check the box to create a shortcut in the Start Menu for the *com0com* executable (the setup program used to configure the virtual COM ports).

3. Use that Start Menu shortcut to start the *com0com* configuration utility. Create the three virtual COM port pairs as shown in Figure 2.

4. Open the *hub4com* zip file you downloaded. There is no setup program for this utility. Simply extract the entire *hub4com* file folder into *C:\Program Files (x86)*. As this is a protected folder, administrative privileges are required to copy the files.

5. Using Notepad or another text editor, copy the Visual Basic script shown on the preceding page into a new text file. Name it “GPS_Split.vbs” and save it in the Windows Startup folder as

described previously. Note that you will have to change “COM7” in the script to the COM port assigned to your GPS receiver.

6. Open up each of the applications that require GPS data and assign the COM ports you chose for each application.

That’s it. Now, the next time Windows is started, the *hub4com* utility will run and your applications should each be receiving their copy of the GPS data stream.

•••–•–