Installing Com0Com Null Modem Emulator (an alternate to vCom)

Last Updated: 29-October-2008

The null-modem emulator (Com0Com) is a kernel-mode virtual serial port driver for Windows. You can create an unlimited number of virtual comport pairs and use any pair to connect one comport based application to another.

The null-modem emulator is available freely under GPL license. It allows you to create the virtual serial port pairs to interconnect comport based applications. Com0com operates in the same manner as vCom and can be used, as an alternative to vCom if so desired.

Here is the Com0Com SourceForge project page: http://Com0Com.sourceforge.net/

Getting Started:

First, you need to download and install the appropriate Com0Com driver for your PC platform. It is located in the SourceForge.net repository. The download page for the Com0Com project is:

http://sourceforge.net/project/showfiles.php?group_id=129551&package_id=141993

You will see two types of platform specific files listed. Download the one that is correct for the operating system running on your PC to a folder on your PC:

- 32-bit Windows operating systems are denoted by the "i386" designation in the file name: **com0com-2.1.0.0-i386-chk.zip**
- 64-bit Windows operating systems are denoted by the "x64" designation in the file name: **com0com-2.1.0.0-x64-chk.zip**

Note: Download the most current version of Com0Com that is appropriate for the operating systems you are running it on.

WARNING: Before installing Com0Com, it is recommended that you uninstall vCom if you have it already installed so that the two virtual null modem emulators do not have the possibility of conflicting with one another.

NOTE: (Windows Vista): Before installing/uninstalling the Com0Com driver or adding/removing/changing ports on Windows Vista, the User Account Control (UAC) should be turned off. Turning off UAC or enabling test signing will impair computer security. Use with caution.

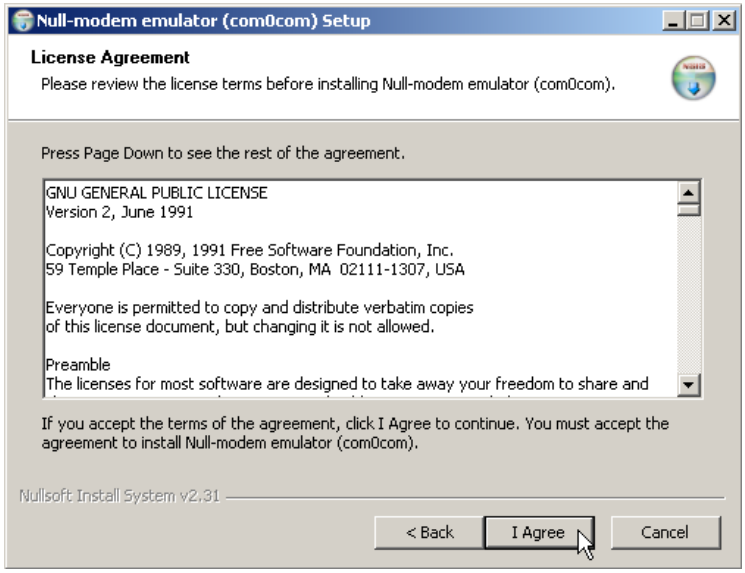
NOTE: (x64-based Windows Vista): The Com0Com.sys is a test-signed kernel-mode driver that will not load by default on x64-based Windows Vista. To enable test signing, use the following: BCDedit command: bcdedit.exe -set TESTSIGNING ON

Installing Com0Com:

After you have extracted the files to a standard Windows folder, **double left click** on the **setup.exe** file and follow the instructions for installing the Com0Com application and kernel mode drivers on your system.

At the "*Welcome to the Null-modem emulator (Com0Com) Setup Wizard*" screen, **left click** on the **Next>** button to start the installation of the software. **Left click** on **Cancel** to abort the software installation. See the screen example below.

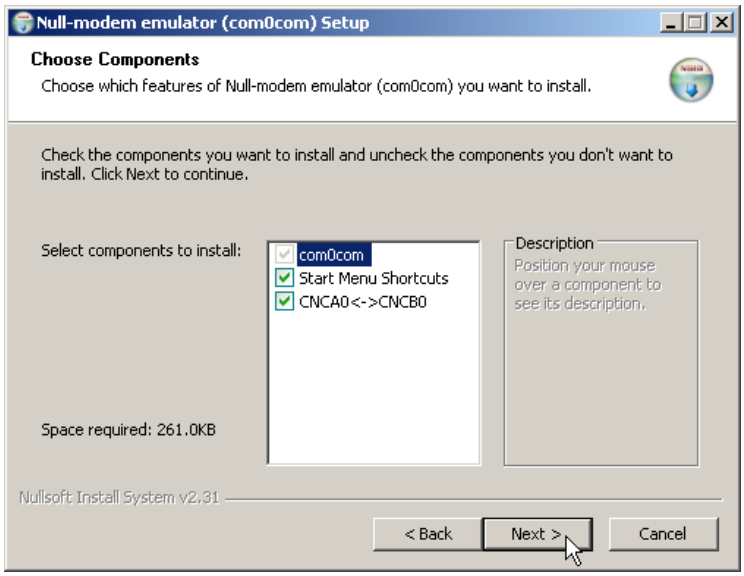
After clicking on Next at the "*Welcome to the Null-modem emulator (Com0Com) Setup Wizard*" screen, the "*License Agreement*" screen is displayed. **Left click** on the **I Agree** button to continue the installation of the software. **Left click** on **Cancel** to abort the software installation. See the screen example below.



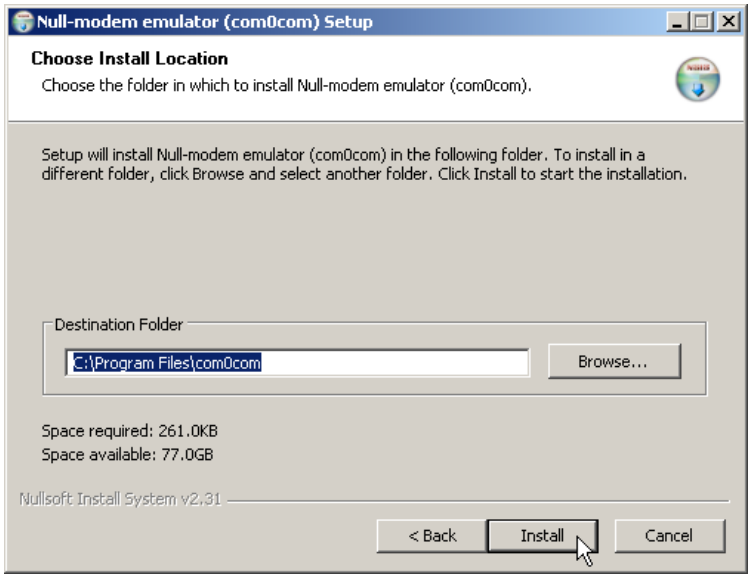
After clicking on I Agree at the "*License Agreement*" screen, the "*Choose Components*" screen is displayed. Do not change the default settings.

The bottom option, **CNCA0<->CNCB0** represents the definition of a default Virtual Port Pair and should be left checked so that one new com port pair is created during installation. You should notice that the Virtual Port Pair comport names do not have the standard naming convention such as COMx, where x is the comport number. Do not worry about this. We will be changing the Virtual Port Pair's comports names shortly to match the COMx, syntax.

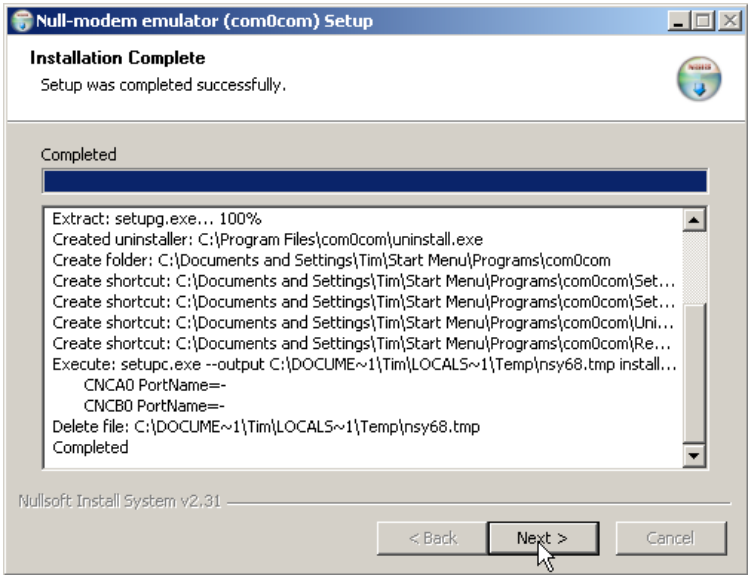
Left click on the **Next>** button to continue the installation of the software. **Left click** on **Cancel** to abort the software installation. See the screen example below.



After clicking on Next at the "*Choose Components*" screen, the "*Choose Install Location*" screen is displayed. It is recommended that you use the default location unless your system root dick drive is not drive C: **Left click** on the **Install** button to continue the installation of the software. **Left click** on **Cancel** to abort the software installation. See the screen example below.



After clicking on Next at the "*Choose Install Location*" screen, the software installation process will begin. You can monitor the installation progress. Once it is complete, the "*Installation Complete*" screen is displayed. **Left click** on the **Next>** button to continue the installation of the software. **Left click** on **Cancel** to abort the software installation. See the screen example below.

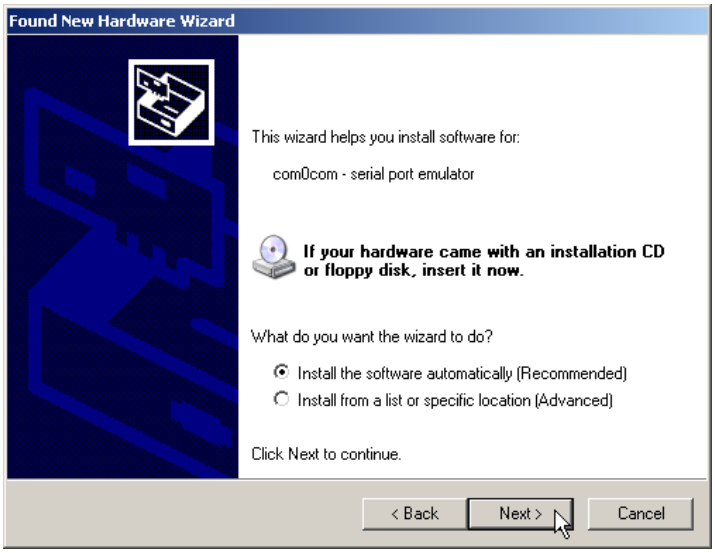


Since you install the default Virtual Port Pair as part of the software installation, Windows will recognize that new hardware has been discovered.

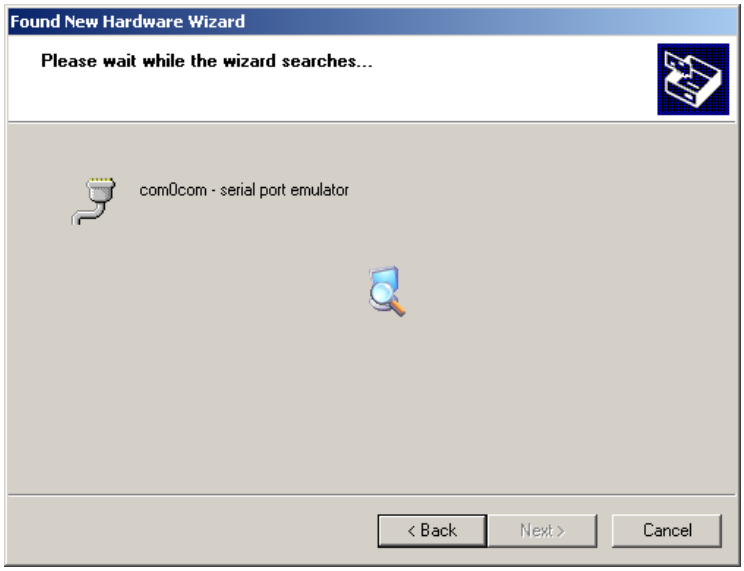
Note: Since there are two comports defined in each Virtual Port Pair, the following hardware installation process will occur twice for every Virtual Port Pair you define.

When the "*Welcome to the Found New Hardware Wizard*" screen is displayed, select the "*No, not this time*" option and **left click** on the **Next>** button. See the screen example below.

When the "*Found New Hardware Wizard*" screen that prompts you for a location of the hardware driver is displayed, up select "*Install the software automatically (recommended)*" option and **left click** on the **Next>** button. See the screen example below.



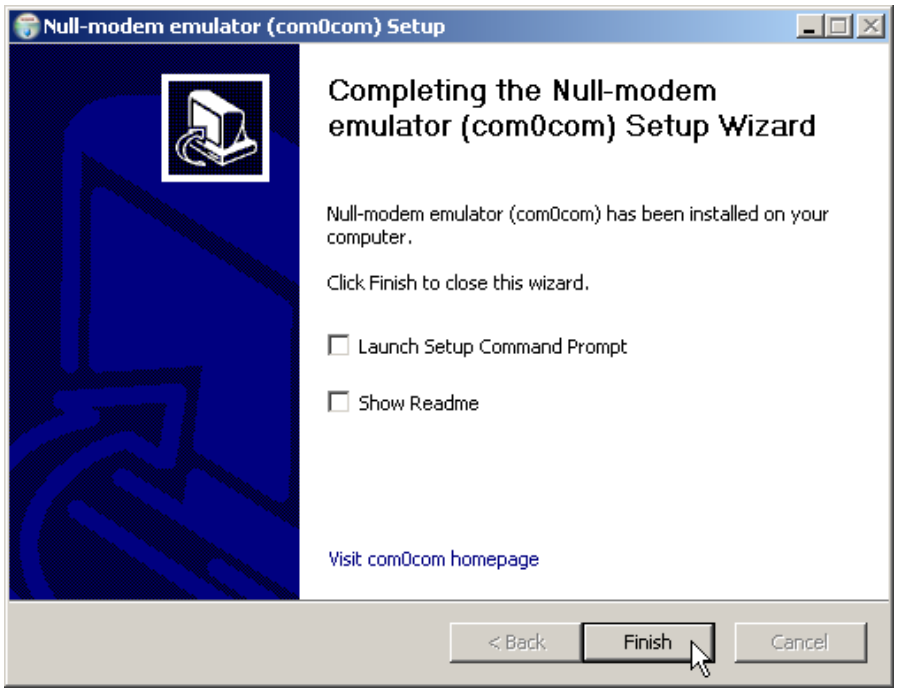
The "Please wait while the wizard searches…" progress screen will be displayed while the driver is installed. See the screen example below.



Once the driver installation is complete, you will receive the "Completing the Found New Hardware Wizard" confirmation screen. **Left click** on the **Finish** button to complete the installation of the comport driver. See the screen example below.

You should see an indication in the bottom right hand portion of the screen that the new hardware has been found and is ready to use indicating that the comport driver successfully installed. Since this is just the first comport of the Virtual Port Pair that has been installed, immediately following the installation of the first comport, the second comport will start its installation. Follow the instructions above for the second comport installation.

After both comports of the Virtual Port Pair have been installed, you must **left click** on the **Finish** button to complete the installation of the Com0Com software. See the screen example below.



Configuring Com0Com:

After the comport hardware drivers have been successfully installed, it is time to add additional Virtual Port Pairs and change the comport names of the two comports in the pair to names that PowerSDR™ and other software packages recognize so they can be used.

The reason Com0Com uses this unique comport naming convention is to prevent newly created comports from conflicting with existing virtual and physical com ports already defined on your computer.

First, lets add an additional Virtual Port Pair before renaming the comports using the Com0Com Setup program.

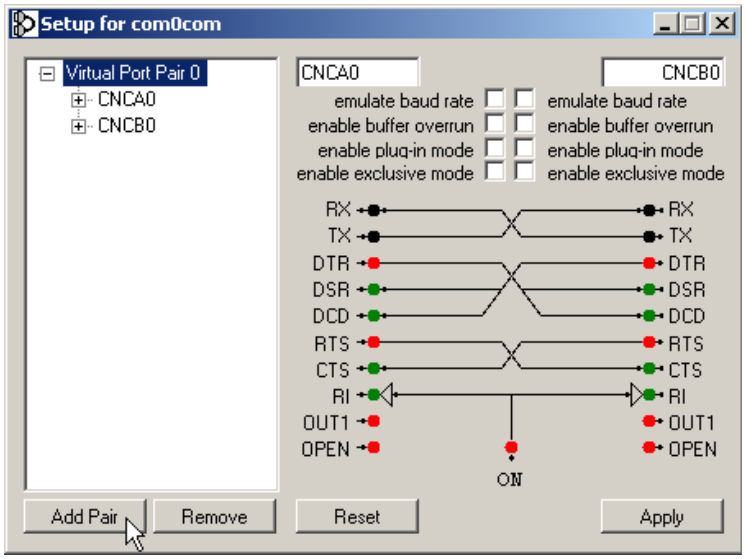
The Com0Com installation program creates an application menu in your *All Programs Start Menu* called "Com0Com".

NOTE: Microsoft .NET Framework 2.0 is required to run the Com0Com Setup program.

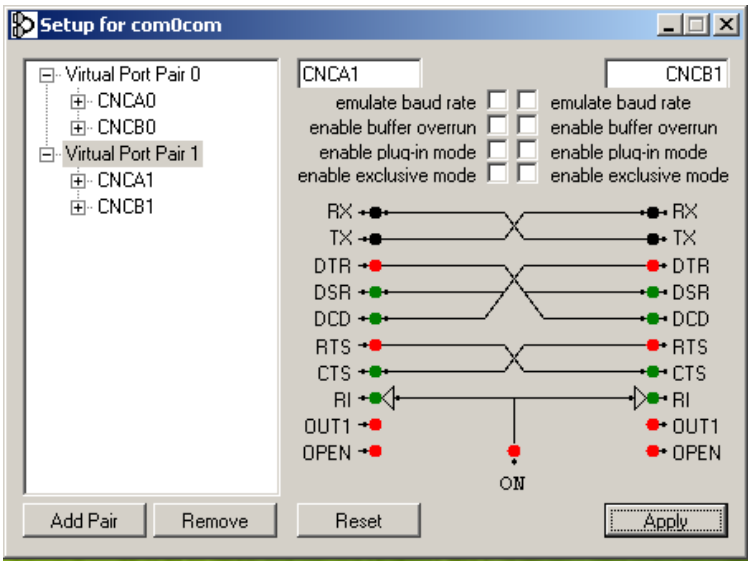
To start the Com0Com **Setup** application, **Left click** on the Windows **Start** button and place your mouse cursor over the **All Programs** at the bottom of the *Start Menu* to display your installed applications.

Place your mouse cursor over the **Com0Com application menu** to display the menu options. Move your cursor over the *Setup* option and **left click** the mouse to select it. The example below shows the four (4) Com0Com menu options with the *Setup* menu option highlighted. See the screen example below.

Below is the "Setup for Com0Com" application screen. Note that there is already one Virtual Port Pair defined using comports **CNCA0** and **CNCB0.** **Left click** on the **Add Pair** button to create a new Virtual Port Pair. See the screen example below.



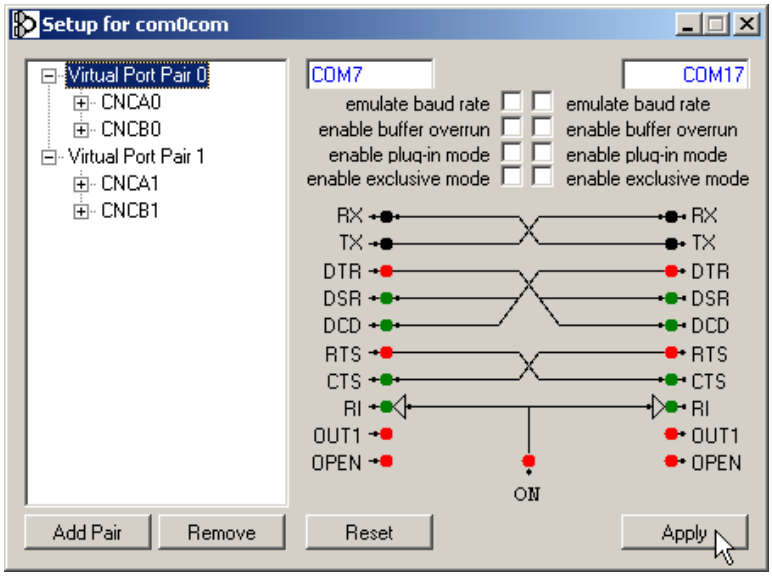
Once the new Virtual Port Pair has been created, it will be displayed in the Setup Screen. See the screen example below.



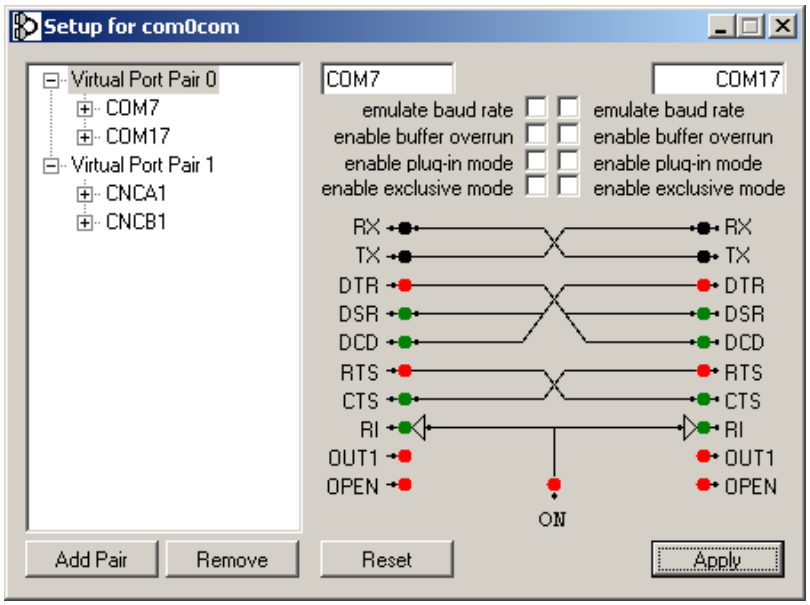
Now we need to change the names of the individual comports in the Virtual Port Pair to names that are recognizable by other software programs and PowerSDR.

NOTE: Enter the comport names of the Virtual Port Pair in capital letters to ensure the greatest compatibility with other programs using the Virtual Port Pairs

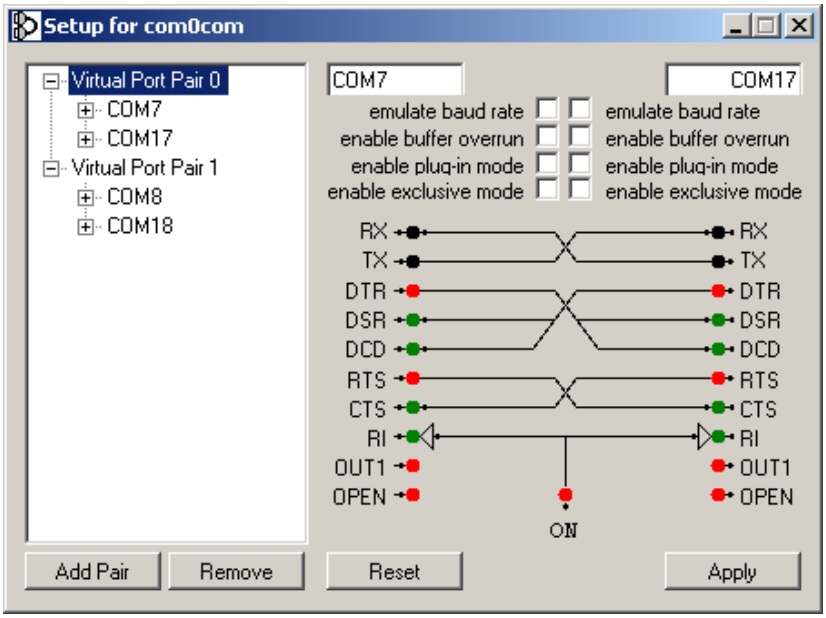
Highlight the Virtual Port Pair where you want to change the two (2) comport names. Once highlighted, change the comport names in the two text boxes in the top right hand side of the *Setup* screen. In the example below, Virtual Port Pair 0 has been selected and **CNCA0** has been changed to **COM7** and **CNCB0** has been changed to **COM17**. The new port names will be in **blue** until they have been activated. **Left click** on the **Apply** button to affect the individual comport name changes. See the screen example below.



It will take several moments to reconfigure both comports in the Virtual Port Pair so please be patient. Once the change is completed, the comport names will be displayed in black. See the screen example below.



Repeat this process for all new the Virtual Port Pairs that you define. See the screen example below.



Optimizing Com0Com Virtual Port Pairs:

There are a few optimizations you may want to perform on particular Virtual Port Pairs that are being used for CAT communication. When a Virtual Port Pair is used for CAT functionality, there can be a lot of communication taking place between the controlling program and PowerSDR. This is especially true if the application is polling or scanning multiple frequencies.

In these cases you may experience times where the application can "hang" or miss some CAT commands because of a receive buffer overrun. By default, receive buffer overrun is disabled for all Virtual Port Pairs. You can mitigate the problem by enabling receive buffer overrun for the receiving port, which would be the end or comport of the Virtual Port Pair that is used by PowerSDR. In addition, to prevent some flow control issues you need to enable baud rate emulation for the sending port, which would be the program controlling PowerSDR.

In the example below, Virtual Port Pair 0 (com7/17) is used for CAT. Com7 is the comport used by the controlling program (a logger, PC-ALE, MixW, etc) and Com17 is configured as the comport in PowerSDR.

Since Com7 is the sending port and Com17 is the receiving port, you would want to configure **emulate baud rate** on Com7 and **enable buffer overrun** on Com17.

All you need to do is place the appropriate check mark in the proper check box and **left click** on the **Apply** button to enable the comport options.

NOTE: It is best to make changes to the Virtual Port Pair when both PowerSDR and the controlling program are not active.