

Problem:

After rebooting the Raspberry Pi, the Com port assignments for attached A/D's may have unexpectedly changed from what they were previously. This means that, at least, you may have to change the Com Port number in WinSDR's System settings and, worse, if you are running more than one A/D, it may end up confusing and corrupting their 'WinSDR.ini' system-settings files. However, it is possible to have the R Pi use information it reads from the USB devices at boot-up to assign specific USB and Com port numbers, so that each A/D always gets the same ones.

Here's how I got it to do that.

First, find the Vendor, Product and Serial numbers for each of your attached A/D's:

With WinSDR(s) **not** running and all A/D's powered up but **not** connected to the R Pi, reboot it. Then, in a Terminal window run 'lsusb'

```
pi@raspberrypi:~ $ lsusb
```

You should see a listing that looks something like this:

```
pi@raspberrypi:~ $ lsusb
```

```
Bus 001 Device 005: ID 04ca:004b Lite-On Technology Corp.
```

```
Bus 001 Device 004: ID 093a:2510 Pixart Imaging, Inc. Optical Mouse
```

```
Bus 001 Device 003: ID 0424:ec00 Standard Microsystems Corp. SMSC9512/9514 Fast Ethernet Adapter
```

```
Bus 001 Device 002: ID 0424:9514 Standard Microsystems Corp.
```

```
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```

Devices 001, 002 and 003 are internal devices, native to the R Pi, while 004 and 005 are my USB mouse and USB keyboard.

Now hot-plug one A/D to a free USB port and run 'lsusb' again to look for the new entry.

In my case the 24bit A/D's internal USB UART IC, created a new line which read:

```
'Bus 001 Device 006: ID 0403:6001 Future Technology Devices International, Ltd FT232 USB-Serial (UART) IC'
```

Record the Device number, '6' and which A/D it was for.

Plug in the next A/D or its serial-USB adapter, if any, and again run 'lsusb', record its Device number, etc.....

My 16-bit A/D with its serial converter created a new line which read:

```
'Bus 001 Device 007: ID 050d:0109 Belkin Components F5U109/F5U409 PDA Adapter'
```

Now you need to find the data which individually identifies each of your A/D's.

Run 'dmesg' from the Terminal window to read the kernel messages, and send its output to a text file 'dmesgout' on the Desktop.

```
pi@raspberrypi:~ $ dmesg > ~/Desktop/dmesgout
```

Though nothing will appear to happen, you should see the text file 'dmesgout' appear on the Desktop.

Then you can open it with the text editor, and search it (Search > Find) for lines of interest.

For my 24 bit A/d, which was Device 006, I searched for the text 'USB device number 6' and located the following lines:

```
[ 137.355919] usb 1-1.3: new full-speed USB device number 6 using dwc_otg
[ 137.483516] usb 1-1.3: New USB device found, idVendor=0403, idProduct=6001
[ 137.483549] usb 1-1.3: New USB device strings: Mfr=1, Product=2, SerialNumber=3
[ 137.483567] usb 1-1.3: Product: FT232R USB UART
[ 137.483584] usb 1-1.3: Manufacturer: FTDI
[ 137.483599] usb 1-1.3: SerialNumber: A402V6DK
```

which let me record the 'device number', 'idVendor' and 'SerialNumber' for the 24-bit USB A/D.

Then I searched for 'USB device number 7' and located the lines:

```
[ 226.456083] usb 1-1.2: new full-speed USB device number 7 using dwc_otg
[ 226.590145] usb 1-1.2: New USB device found, idVendor=050d, idProduct=0109
[ 226.590178] usb 1-1.2: New USB device strings: Mfr=1, Product=2, SerialNumber=3
[ 226.590196] usb 1-1.2: Manufacturer: Staples USB to Adapter
[ 226.590212] usb 1-1.2: SerialNumber: 246436
```

which recorded the device, vendor and serial numbers for the Serial-USB adapter on my 16-bit Serial A/D. Note that the serial numbers will be unique to each individual A/D or other USB device.

Now, to force specific USB device number(s) to always get assigned to the same USB device(s):

In a Terminal window use **sudo nano /etc/udev/rules.d/50-usb-serial.rules** to enter and save one line for each attached A/D in file '50-usb-serial.rules', like the two lines below, but using the values you found for your own USB device(s).

```
SUBSYSTEM=="tty", ATTRS{idVendor}=="0403", ATTRS{idProduct}=="6001",
ATTRS{serial}=="A402V6DK", SYMLINK+="ttyUSB7"
SUBSYSTEM=="tty", ATTRS{idVendor}=="050d", ATTRS{idProduct}=="0109",
ATTRS{serial}=="246436", SYMLINK+="ttyUSB8"
```

These were typed as two single lines, one for each USB connected device.

Then <ctrl> O <enter> to save the file and <ctrl> X to exit the nano editor.

This example will cause: "ttyUSB7" to always get assigned to the output from that 24-bit A/D and "ttyUSB8" to that serial-USB converter. You can assign them any 'ttyUSB' values you want, but so far I haven't been able to figure out how to use numbers larger than one digit.

Now create symbolic links in directory '~/.wine/dosdevices/', linking those 'ttyUSB' assignments and corresponding Com port numbers, by running:

```
ln -s /dev/ttyUSB7 ~/.wine/dosdevices/com7
```

```
and ln -s /dev/ttyUSB8 ~/.wine/dosdevices/com8
```

which, in my case, linked "com7 to ttyUSB7" and "com8 to ttyUSB8"

Now, this specific 24-bit A/D will always get assigned to com7 and this 16-bit A/D always to com8, even after re-booting the Raspberry Pi.

Note that you can assign any Com numbers you want; they don't need to be the same as the 'ttyUSB' numbers, as I chose to do here.