**Running ARDOPC**

**Some notes on ardopc (my C version of ARDOP\_WIN)**

ardopc is a command line program. It supports two host interfaces, TCP (using two TCP ports) or Serial using a Virtual COM port to link to the host. The serial interface can emulate an SCS Dragon controller, allowing programs that don't support ardop but support the Dragon (such as Winlink Express or Airmail) to use ardop.

If run without any parameters it listens on TCP port 8515, on Linux it uses the virtual sound device ARDOP, and on Windows the first available capture and playback device. You can specify command line parameters port capture device playback device.

For TCP Host connection, port is TCP Port Number.
For Serial Host connection port must start with "COM" or "com"
On Windows use the name of the BPQ Virtual COM Port, eg COM4. See below for info on installing the BPQ Virtual COM port driver
On Linux the program will create a pty and symlink it to the specified name.

On Linux the capture and playback devices are in hw:1,0 format. In Windows it is a number starting from zero or a string that matches part of the device name. When you start the program it will list all the sound devices so you can select the right one. If you are specifying sound devices you must also include the port, eg

./ardopc 8515 hw:1,0 hw:1,0 (Linux TCP)

ardopc 8515 1 1 (Windows TCP using second device in list)

./ardopc com10 hw:1,0 hw:1,0 (Linux Serial)

ardopc COM10 USB USB (Windows Serial using device USB Audio CODEC)

On Linux not many soundcards support the required sampling rate of 12000 samples/sec. See paragraph below on rate conversion if your card doesn't.

The TNC supports a number of options for PTT. When using the TCP interface it will send commands to the host to allow it to contol PTT. This is most useful when the host program is running more than one sound card mode (eg ARDOP and WINMOR). The TNC can also key the radio via a hardware or software (CAT) interface. There are a number of optional command line parameters to control this

-p device or --ptt device Device to use for PTT control using RTS or GPIO Pin (Raspbery Pi only)

-c device or --cat device Device to use for CAT Control

-k string or --keystring string String (In HEX) to send to the radio to key PTT

-u string or --unkeystring string String (In HEX) to send to the radio to unkeykey PTT

Hardware control is normally via the RTS line on a COM port (possibly a Virtual COM Port, eg with the IC7300). On a Raspberry PI a GPIO pin can also be used. For example
-c COM10
-c /dev/ttyUSB0
-c GPIO (Using default pin 17) or -c GPIO=n (Using a different pin).

The -c or --cat paramter is specified in the form port:speed, eg com1:19200 or /dev/tty0:19200. Some host programs (such as BPQ32/LinBPQ and Winlink Express) can send radio control commands for frequency and mode setting over this port, but for reasons of latency the PTT commands have to be generated by the TNC. Some host programs (such as BPQ32/LinBPQ) can set these commands over the ardop control interface, but for others the -k and -u commands are used. The value is the string (specified in hex) to be sent to key or unkey the radio. For example for most ICOM radios you could use

-k FEFE88E01C0001FD -u FEFE88E01C0000FD

CAT and RTS PTT can share the same com port, so for radios like the IC7300 which support keying via RTS on the virtual com port you could have the application controlling frequency and the TNC controlling PTT. Apart from the command line params above there is no configuration. All parameters are set by the application, though there are reasonable defaults for most of them.

The program writes a debug log as well as debugging information to the program console.

To simplify setting the audio input the program displays the peak input level every 10 seconds. The program works with signed 16 bit samples, so the values will be in the range -32768 to 32767. Clipping will cause distortion, so it is import to keep away from the limits. I've found in practice that it will work down to quite low levels, but try to keep in the range +-20000.

The software can be downloaded from

<http://www.cantab.net/users/john.wiseman/Downloads/Beta/ardopc>
<http://www.cantab.net/users/john.wiseman/Downloads/Beta/piardopc>
<http://www.cantab.net/users/john.wiseman/Downloads/Beta/ARDOPC.exe>

**Sampling Rate Conversion**

ARDOP uses a sampling rate of 12000 samples/second, and not all that many Linux sound cards support this directly. So if you are using Linux you will probably have to enable sample rate conversion. This is achieved by adding the following to the alsa configuration. I suggest you add it to the user configuration file .asoundrc, creating the file if it doesn't already exist. The file should be in the user's home directory. Note there is a dot on the front of the name, which isn't obvious with some fonts.

pcm.ARDOP {type rate slave {pcm "hw:1,0" rate 48000}}

This creates a virtual rate conversion device called ARDOP, using the real device hw:1,0, and running the real device at 48000. If the sound card you are using isn't hw:1,0, then change the line to match your device.

You can also enable sample rate converison using the ALSA "plug" device, eg plughw:1,0, but as this doesn't allow you to specify the real sampling rate I prefer using the virtual device.

**Virtual COM Port Installation**

If you have installed BPQ32 the needed files are in folders VCOM32 or VCOM64 under your BPQ32 Programs folder. If you don't use BPQ32 you can download the files as part of the CAT7200 installation. Download <http://www.cantab.net/users/john.wiseman/Downloads/CAT7200.zip> and unzip to a convenient folder.

Note that there are separate versions for 32bit and 64bit Windows, in the VCOM32 and VCOM64 subdirectories - make sure you install the right one.

For Installation on Windows XP, see intructions [here](http://www.cantab.net/users/john.wiseman/Documents/VCUSMInstallXP.htm).

To Install on a Windows 8 or Windows 10 64 bit system you must [reboot with advanced options](http://www.makeuseof.com/tag/how-can-i-install-hardware-with-unsigned-drivers-in-windows-8/) to disable driver signature enforcement.

With Windows 7 and above, you should be able to install the driver by right clicking on InstallBPQVCOM.exe (or InstallBPQVCOM32.exe if 32 bit) and selecting "Run as Administrator". If that fails, [follow these intructions](http://www.cantab.net/users/john.wiseman/Documents/VCUSMInstallW7.htm).

Use Device manager to identify the COM port number allocated to your Virtual COM Port.

John Wiseman G8BPQ
January 201