Howto: Using USB serial ports under wine on Linux

Background

Wine is a free, open-source, Windows compatible API that is available for most Linux distributions. Many, but by no means all, Windows programs can be run under wine with almost no loss of functionality – remarkably this includes many games. It can be installed from most Linux distribution archiving services. For Ubuntu, wine binaries can be installed easily using synaptic package manager. For more information have a look at winehq.

I have installed and tested wine under Ubuntu 8.04 through to 12.04LTS workstation on a 3.0GHz P4, on a 2GHz Centrino and on an Asus EeePC (1.6GHz Intel Atom). It will ***not*** work on the Raspberry Pi The current version of wine is 1.7.

Thankfully, much amateur radio software does not use very tricky features of the OS and will therefore run under wine with no problems. The difficulty with any wine program is i/o and most problems encountered seem to be when dealing with serial ports.

USB Serial Adapters

There are a wide range of USB serial adapters available. Many machines these days do not have native serial (COM) ports and the USB adapters have to be used to provide serial communication to peripherals. Additionally, many USB devices are merely serial devices with USB connectivity included, so they are treated by the Windows software as if they are a serial port.

Linux, on the other hand, doesn't care too much about these sorts of devices. Once a device is plugged in the OS automatically assigns a device file (located in /dev) according to what the OS can determine about the capabilities. A serial device will appear in the form of a file in /dev with filename starting with “tty”. With Ubuntu, a USB serial device defaults to the filename “ttyUSBx” where “x” is a number typically between 0 and 9 (but there is no formal limit in Linux). The number assignments appear to be based on time order of connection and so the first device plugged in will be named “ttyUSB0” and onwards.

Two things need to be done to enable these devices to be accessible to Windows programs running under wine: First, the Linux port needs to be mapped to a Windows COM port; and second, the port needs to be made accessible to any user (as it is not appropriate to run your Windows programs as root).

Assigning /dev/ttyUSBx ports to COM ports

Your Linux machine doesn't know about the limitations of memory-addressed i/o imposed by primitive Windows so a mapping needs to be made in order to make the COM ports work.

I have used a number of USB serial adapters such as those available inexpensively from Maplin (in UK) and with rig-specific connectors such as a USB-FT-817 adapter from eBay.

I have not had problems with Linux recognising these serial i/o devices. Therefore the Windows driver disks that are often provided can be put to one side.

Once such a device is plugged in to your Ubuntu based machine, you will find that a new device is automagically created in the /dev directory. To find this type:

```
ls -al /dev/ttyU*
```

You will find a device (or more than one) with the name:

```
/dev/ttyUSBx
```

where “x” is an integer.

In my case, the first serial port was called /dev/ttyUSB0. You then need to find your wine devices directory thus:

```
cd ~/.wine/dosdevices
```

Most likely this directory will be empty which you can check by typing:

```
ls -al
```

You need to create a special symbolic link to support the device under wine. This should be done using temporarily assigned root privileges:

```
sudo ln -s /dev/ttyUSBx com1
```

You will have to type your password to enable root privileges when using commands preceded by “sudo”.

On my machines, I have typically between three and four USB ports. I have therefore collected together as many of these USB devices as I might need (to force auto-creation of the /dev/ttyUSBx device files) and produced symbolic links for all of them, giving typically ports COM 1 – COM 3 as typically found on a Windows PC. The Linux device file assignments start with 0 whereas Windows numbering starts with 1.

When a USB device is removed, the automatically created device file is deleted. The symbolic link, however, remains. The next time a similar device is plugged in and allocated to the appropriate device file, the link will appear to be activated again.

Enabling user access to devices

Some users may find (particularly on more recent flavours of Linux, that the wine package still cannot see the COM port device even though the above instructions have been followed. This is because the device files have specific access permissions that are restricted to system processes, of which wine is not one (when invoked by a normal user). The most straightforward way around this is to create a udev assignment which forces the device file (created on insertion of the USB serial adaptor) to be assigned to a specific user (you) and to have general access permissions. How you deal with multiple users on one machine may be a bit more tricky.

To do this you need to edit a currently non-existent file as follows:

```
sudo vim /etc/udev/rules.d/ttyUSB.rules
```

where “vim” is your favourite editor (those very familiar with Linux could use the basic vi editor, others may prefer full screen editors, it doesn't matter which except that you should not use a word processor as it will add a lot of rubbish to the file that is not needed).

Add the following line(s) as required:

```
KERNEL==”ttyUSB[0 … 9]” SYMLINK+=”%k” GROUP=”abcd” MODE=”0666”
```

where “abcd” refers to the GROUP that your user name is in (typically the same as your username). To find this out, go to your home directory (cd $home) and type ls -al when you will see files with the designation of the owning group and name such as:

```
-rw-rw-r--  1 ipb  ipb   119047 Nov 10 21:02 IC-92d.icf
```

In this case the first “ipb” is the user and the second is the group. Put your group name in the GROUP entry in the udev rules file.

Once this is done any newly started program which uses a USB serial device (whether or not under wine) will be able to access it without special privileges. To see this, remove and re-plug a USB serial device and look at the device file (ls /dev/ttyU*) which may now appear as follows:

```
crw-rw-rw- 1 root ipb 188, 0 Nov 23  2013 /dev/ttyUSB0
```

The udev rule automatically assigns the newly inserted USB serial device to the group “ipb” on my machine, making it available to any user in that group. For good measure, the permissions are widened to cover other non-root users. Take care with these permissions when using machines in public networked environments and be aware that these instructions may need to be modified if the machine is a multi-user machine with several groups of users. Hopefully you will find that these tips allow you to make effective use of Windows programs requiring serial i/o for your applications under wine.

Disclaimer

Your mileage with this may vary. The author shall not be held responsible for any damage, disruption, misoperation or other dysfunction whatsoever howsoever caused as a result of following or attempting to follow the instructions in this note.