

# Guide: Parallel port passthrough in Proxmox VE 8.3.1

## Why?

Recently I have been messing around with a 30+ year old film recorder, which runs over parallel on Windows 95/98.

I don't have the space to dedicate a vintage PC to it, but I do have a low power Proxmox server with a parallel port!

However, there's fairly little documentation online about passing parallel ports through to a VM.

I was only able to find [one slightly confusing thread](#) from 2009 on the PVE forums!

So, here's a nice clear guide on how to do it!

## Prerequisites:

- A Proxmox 8 server with a parallel port  
Honestly, this will probably work on any version, but I used PVE 8.3.1 running on an Asrock N3050B-ITX board.
- PuTTY, or whatever SSH client you're used to
- Physical access to said machine
- A keyboard and monitor

## Disclaimer:



**Follow this guide at your own risk!**

**I am not responsible for any mistakes, explosions, dead film recorders, or anything else!**

That said, I have done my best to ensure this guide is as accurate as possible, and these instructions did work for me!

(if I did make a mistake though, please [contact me](#), and I'll fix it!)

However, I will not provide any support, so be warned!

## 1: BIOS Setup

Go into the BIOS or UEFI of your chosen server, and find the setting for "Parallel Port Mode".

For my board, that was under Advanced > Super IO Configuration.



Make sure the parallel port is enabled, and that it's in ECP+EPP mode. The version of this mode doesn't really matter! (mine has 1.9 and 1.7)

Then, save and reboot into Proxmox.

## 2: Proxmox Setup

Open an SSH connection to your Proxmox host (not to a VM!), and login as root.

Then, check that Proxmox can see your parallel port with the following command

```
dmesg | grep parport
```

That should return something like this.

```
root@poggers:~# dmesg | grep parport
[  9.003225] parport_pc 00:03: reported by Plug and Play ACPI
[  9.003398] parport0: PC-style at 0x378 (0x778), irq 5, dma 3 [PCSP,TRISTATE,COMPAT,EPP,ECP,DMA]
root@poggers:~#
```

If it doesn't, double check your BIOS settings then try again.

Next, use this command to enable Proxmox' parallel port kernel module, *ppdev*.

```
modprobe ppdev
```

To make sure that it's enabled, run

```
ls -la /dev/parport*
```

You should now see something like this :)

```
root@poggers:~# ls -la /dev/parport*
crw-rw---- 1 root lp 99, 0 Feb  1 18:16 /dev/parport0
root@poggers:~#
```

Now you'll need to make sure that module is loaded at every boot, by adding it to */etc/modules*.

Open */etc/modules* in your favourite text editor (I used nano), then add *ppdev* on a new line.

```
nano /etc/modules
```

```
GNU nano 7.2 /etc/modules
# /etc/modules: kernel modules to load at boot time.
#
# This file contains the names of kernel modules that should be loaded
# at boot time, one per line. Lines beginning with "#" are ignored.
# Parameters can be specified after the module name.
ppdev
```

Save (ctrl+O in nano), exit (ctrl+X), and reboot your Proxmox host.

```
reboot now
```

Finally, to make double sure that it's enabled on boot, run this again after Proxmox has rebooted

```
ls -la /dev/parport*
```

You should see exactly the same output as the first time you ran it :)

If so, you're now ready to pass it through to a VM!

## 3: VM Setup

To pass the port through to a virtual machine, just run this

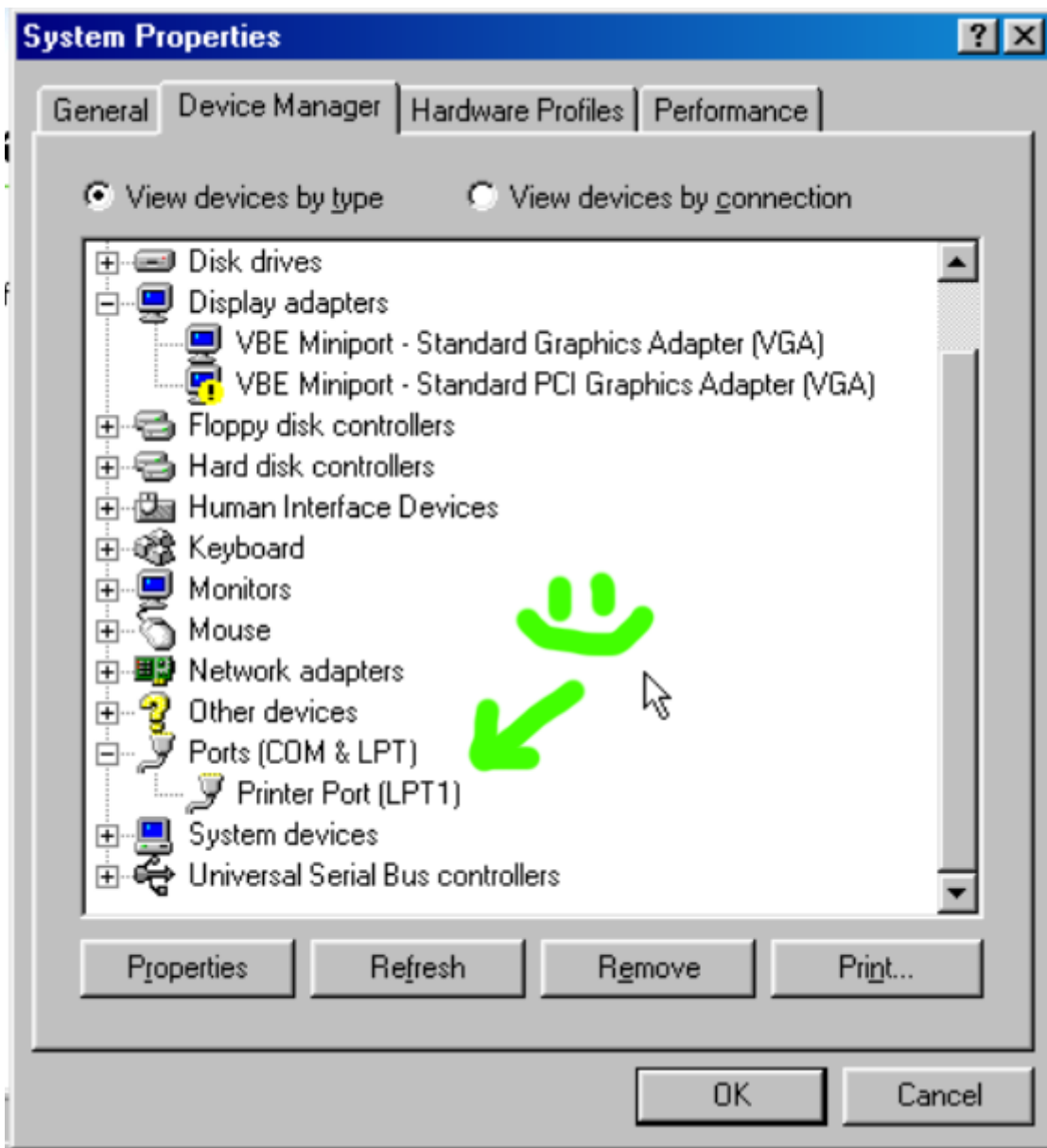
```
qm set 100 -parallel /dev/parport0
```

Replace 100 with the ID of your virtual machine!

Make sure the VM isn't running!

I've only tested this with i440fx/SeaBIOS type VMs, but it'll probably work with q35 too.

Now start/restart your chosen VM, and check to see if the port is available.



and that's it, you are now ready to print like it's 1995!

Also, if you're interested,  
I'll be doing a full project post soon about getting my film recorder running, so keep your eyes  
peeled over at [nycto.io](https://nycto.io)!

Happy printing! (or film recording!)

---

Revision #1

Created 1 February 2025 15:53:14 by nycto

Updated 1 February 2025 19:06:42 by nycto