

# On windows with WSL

Tuesday, 10 January 2023 13:29

\$ WSL = Windows Subsystems for Linux = Fast Linux VM on widows with terminal only

- If the command is preceded by a `PS` then it's a Windows command to execute in powershell else if you see `$` then it is a linux command to type in the wsl shell

Follow this poste to install WSL

<https://lecrabeinfo.net/installer-wsl-windows-subsystem-for-linux-sur-windows-10.html>

## Requirements :

- WSL 2
- Linux Kernel version > 5.10.60 (to test : `uname -r`)

This was tested on Ubuntu 20.04 focal ad 22.04 jammy with a kernel of version 5.10.102.1

## Distribution manual installation

If you encountewsl r an error while installing the WSL distribution, download the .AppxBundle file present here:

<https://learn.microsoft.com/en-us/windows/wsl/install-manual>

Choose the linux distro here : <https://learn.microsoft.com/en-us/windows/wsl/install-manual#downloading-distributions>

Open a Powershell terminal to run the command :

`PS Add-AppxPackage .\distribution.AppxBundle`

and double-click the file afterwards. If you have no program able to run Appx file, install from Microsoft store the Application installation software.

## Upgrading the Kernel

If your version is not up to date, you can try to use

`PS wsl --update`

in powershell, this will update your kernel version.

If you get an error, this is likely because the Microsoft store was disabled on the computer. You will need to update manually. Go to <https://www.catalog.update.microsoft.com/Search.aspx?q=wsl> and get the last version (x64).

Open the .cab file and extract the .msi. Launch the .msi and let it update your machine. You may need administrator privilege s.

## Setting up USBIP

(If not already done after installing WSL, make sure to reboot at this point)

You can then follow the instructions at <https://learn.microsoft.com/en-us/windows/wsl/connect-usb#prerequisites>.

### ★ Note:

- If command `$"sudo apt install linux-tools-5.4.0-77-generic hwddata"` does not work, use this instead:
  - o `$ sudo apt install linux-tools-virtual hwddata`
  - o `$ sudo update-alternatives --install /usr/local/bin/usbip usbip `ls /usr/lib/linux-tools/*/usbip | tail -n1` 20`
  - o `$ sudo apt update -y && sudo apt upgrade -y`

You should get something like this when typing `usbipd wsl list`

```
PS C:\Users\g.defoy> usbipd wsl list
BUSID  VID:PID  DEVICE                                STATE
2-3    046d:c534 USB Input Device                       Not attached
2-5    108c:a001 Skywalker Connect, USB Serial Device (COM3) Not attached
2-8    5986:9106 Integrated Camera, Integrated IR Camera Not attached
2-9    06cb:00da Synaptics UWP WBDI                     Not attached
2-10   8087:0026 Intel(R) Wireless Bluetooth(R)         Not attached
```

### ★ Note: usbipd command must be run on Windows command prompt (not ubuntu)

Type in your **admin** prompt:

```
usbipd bind --busid 2-5 --force
```

If you get an error with the command :

```
usbipd wsl attach --busid <busid> #2-5 for me
```

The *-a* option can make it reattach automatically when re-plugging the CM

Ex:

```
usbipd wsl attach --busid 6-4
```

Then open a powershell prompt with normal privileges (not *administrator*) and retype *the attach* command.

★ Note:

If you get an error with usbip when using 'attach' command, such as :

```
PS C:\Users\o.coutureau> usbipd wsl attach --busid 2-1
usbipd: error: WSL 'usbip' client not correctly installed. See https://github.com/dorssel/usbipd-win/wiki/WSL-support for the latest instructions.
```

Run the following command in WSL and try attach again:

```
sudo update-alternatives --install /usr/local/bin/usbip usbip $(command -v ls /usr/lib/linux-tools/*/usbip | tail -n1) 20
```

### Connecting to the CM

You can then go to your wsl prompt and type *lsusb*.

```
gtwice@PCP824:~$ lsusb
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 001 Device 010: ID 108c:a001 Robert Bosch GmbH Skywalker Connect
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```

You should get see the interface when you type the command *ip a* or *ifconfig -a*.

```
enx001e5841b879: flags=4098<BROADCAST,MULTICAST> mtu 1500
    ether 00:1e:58:41:b8:79 txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

★ Note:

Sometimes you will not get the `enx001e5841b879` interface. This is due to a bug of wsl not loading the udev rules, type the following commands:

```
sudo service udev restart
```

```
sudo udevadm control --reload
```

Unplug, replug and reattach.

to avoid that in the future, edit the `.bashrc` file and add `wsl.exe -u root service udev status || wsl.exe -u root service udev restart` at the end of the file

You then need to bring the interface up: you will need to be able to use *sudo*

```
sudo ip link set enx001e5841b879 up
```

Then set your ip address on this interface: (you can use another ip address on the same subrange as the CM)

```
sudo ifconfig enx001e5841b879 169.254.1.51
```

Then you need to add a route to the CM via the interface (use the same ip address as the last command)

```
sudo ip route add 169.254.1.1/32 via 169.254.1.51 dev enx001e5841b879 src 169.254.1.51
```

★ Note:

You will need to input these commands every time the device is connected/disconnected, to go faster you can paste these five lines in your terminal. (paste them all at once !)

```
cat <<EOF >> cm_setup_iface.sh
```

```
#!/bin/sh
```

```
sudo ip link set enx001e5841b879 up
sudo ifconfig enx001e5841b879 169.254.1.51
sudo ip route add 169.254.1.1/32 via 169.254.1.51 dev enx001e5841b879 src 169.254.1.51
EOF
```

To execute the script, type

```
§ sudo sh cm_setup_iface.sh
```


You should be able to ping the device and connect with telnet


```
gtwice@PCP824:~$ ping 169.254.1.1
PING 169.254.1.1 (169.254.1.1) 56(84) bytes of data.
64 bytes from 169.254.1.1: icmp_seq=1 ttl=128 time=1.67 ms
64 bytes from 169.254.1.1: icmp_seq=2 ttl=128 time=8.99 ms
64 bytes from 169.254.1.1: icmp_seq=3 ttl=128 time=8.66 ms
64 bytes from 169.254.1.1: icmp_seq=4 ttl=128 time=9.59 ms
```

To open a TCP port on the CM, use the command :

```
nc -v 169.254.1.1 30471 //here, TCP port number is 30471
```

★ **Note:**

the previous steps may not work even if the attach command was successful. If at the end you cannot ping the device, detach it from windows (  usbipd wsl detach --busid <busid>) and attach it from linux:

1. Re-plug the board
2. Execute the  **ipconfig command in powershell**. Find the line Ethernet adapter vEthernet (WSL) and copy the ipv4 address field
3. In linux type § **sudo usbip list -r <IPV4 addr>** to find the Bosch component Bus ID
4. Then attach the usb: § **sudo usbip attach -r <IPV4 addr> -b <bus id>**
5. Repeat the commands from here : [Link](#)