

Practica: Compartir un Puerto USB con WSL.

Equipo:

Salazar Lares Aylin Ximena.

Rodríguez Osorio Diego.

Benavides Villanueva Bernabé Manuel.

Carreón Hernández Emmanuel.

Medellín López Martin Josafath.

Cano Guzman Jaime.

Torres Juarez Humberto Alexander (Vespertino, F).

Salazar Castro Bryan Alfonso (Vespertino, F).

Ingeniero en sistemas computacionales.

Profesor: Daniel López Piña.

Materia: PIP, 6J.

13-05-25



Contenido:

Objetivo de la Práctica:	3
Procedimiento:	3
Conclusión.	6

Objetivo de la Práctica:

Habilitar un dispositivo USB (en este caso, un adaptador serial CH340) conectado al sistema operativo Windows, para que sea accesible desde WSL 2 (subsistema de Windows para Linux), específicamente desde la distribución Ubuntu. Esta configuración permite utilizar hardware como placas Arduino desde un entorno Linux que corre dentro de Windows.

Procedimiento:

1. Instalación de usbipd-win en Windows

Pasos realizados:

- ✓ Abrir PowerShell con privilegios de administrador.
- ✓ Ejecutamos el siguiente comando:

```
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Instale la versión más reciente de PowerShell para obtener nuevas características y mejoras. https://aka.ms/PSWindows

PS C:\WINDOWS\system32> winget install usbipd
Se encontró un paquete existente ya instalado. Intentando actualizar el paquete instalado...
No se ha encontrado ninguna actualización disponible.
No hay versiones más recientes del paquete disponibles en las fuentes configuradas.
PS C:\WINDOWS\system32> usbipd list
Connected:
BUSID  VID:PID  DEVICE                                STATE
1-3    1a86:7523  USB-SERIAL CH340 (COM5)              Not shared
1-6    0bda:5539  Integrated Webcam                   Not shared
1-10   8087:0026  Intel(R) Wireless Bluetooth(R)      Not shared

Persisted:
GUID                                DEVICE
be3f24be-4873-4091-90c5-15864fdbelcd  USB-SERIAL CH340 (COM9)

PS C:\WINDOWS\system32> usbipd bind --busid=1-3
PS C:\WINDOWS\system32> usbipd list
Connected:
BUSID  VID:PID  DEVICE                                STATE
1-3    1a86:7523  USB-SERIAL CH340 (COM5)              Shared
1-6    0bda:5539  Integrated Webcam                   Not shared
1-10   8087:0026  Intel(R) Wireless Bluetooth(R)      Not shared

Persisted:
GUID                                DEVICE
be3f24be-4873-4091-90c5-15864fdbelcd  USB-SERIAL CH340 (COM9)

PS C:\WINDOWS\system32>
```

2. Listar Dispositivos USB Disponibles

Antes de compartir un dispositivo USB, es necesario identificarlo mediante su BUSID.

Pasos realizados:

- ✓ Ejecutamos:

```
PS C:\WINDOWS\system32> usbipd list
Connected:
BUSID  VID:PID  DEVICE                                STATE
1-3    1a86:7523  USB-SERIAL CH340 (COM5)              Not shared
1-6    0bda:5539  Integrated Webcam                   Not shared
1-10   8087:0026  Intel(R) Wireless Bluetooth(R)      Not shared
```

3. Compartir (Bind) el Dispositivo USB

Se prepara el dispositivo para ser accesible por WSL mediante el comando bind.

Pasos realizados:

- ✓ Ejecutamos en PowerShell (admin):

```
usbipd bind --busid=1-4
```

El dispositivo pasa al estado “Shared”, lo cual indica que ya está disponible para ser adjuntado a una distribución WSL.

```
PS C:\WINDOWS\system32> wsl --list --verbose
NAME      STATE      VERSION
* Ubuntu   Running    2
PS C:\WINDOWS\system32> usbipd attach --wsl --busid=1-4 # Reemplaza "1-4" con tu BUSID
usbipd: info: Using WSL distribution 'Ubuntu' to attach; the device will be available in all WSL 2 distributions.
usbipd: info: Detected networking mode 'nat'.
usbipd: info: Using IP address 172.17.144.1 to reach the host.
PS C:\WINDOWS\system32>
```

4. Adjuntar el Dispositivo a WSL 2

Conecta virtualmente el dispositivo compartido a la instancia de Ubuntu en WSL 2.

Pasos realizados:

- ✓ Ejecutamos en PowerShell (admin):

```
Reading state information... Done
The following additional packages will be installed:
  libusb-1.0-0
Suggested packages:
  usbutils-py
The following NEW packages will be installed:
  libusb-1.0-0 usbutils
0 upgraded, 2 newly installed, 0 to remove and 99 not upgraded.
Need to get 139 kB of archives.
After this operation, 469 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://archive.ubuntu.com/ubuntu noble/main amd64 libusb-1.0-0 amd64 2:1.0.27-1 [54.0 kB]
Get:2 http://archive.ubuntu.com/ubuntu noble/main amd64 usbutils amd64 1:017-3build1 [84.5 kB]
Fetched 139 kB in 2s (86.2 kB/s)
Selecting previously unselected package libusb-1.0-0:amd64.
(Reading database ... 40768 files and directories currently installed.)
Preparing to unpack .../libusb-1.0-0_2%3a1.0.27-1_amd64.deb ...
Unpacking libusb-1.0-0:amd64 (2:1.0.27-1) ...
Selecting previously unselected package usbutils.
Preparing to unpack .../usbutils_1%3a017-3build1_amd64.deb ...
Unpacking usbutils (1:017-3build1) ...
Setting up libusb-1.0-0:amd64 (2:1.0.27-1) ...
Setting up usbutils (1:017-3build1) ...
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for libc-bin (2.39-0ubuntu8.4) ...
diego@DESKTOP-UB0JE7E:~$ lsusb
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 001 Device 002: ID 1a86:7523 QinHeng Electronics CH340 serial converter
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
diego@DESKTOP-UB0JE7E:~$
```

5. Verificación del Dispositivo en WSL (Ubuntu)

Se comprueba que el sistema Linux dentro de WSL detecta el dispositivo correctamente.

6. Identificación del Puerto Serial

```
diego@DESKTOP-UB0JE7E:~$ lsusb
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 001 Device 002: ID 1a86:7523 QinHeng Electronics CH340 serial converter
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
diego@DESKTOP-UB0JE7E:~$
```

Se busca el nombre del archivo de dispositivo asociado al convertidor serial

Pasos realizados:

- ✓ Ejecutamos en Ubuntu:

```
diego@DESKTOP-UB0JE7E:~$ ls /dev
autofs      fuse         loop0       ptmx        ram5        shm         tty18       tty31       tty45       tty59       vcs         vcsu1
block       hugepages   loop1       ptp0        ram6        stderr      tty19       tty32       tty46       tty6        vcs1       vcsu2
bsg         hvc0        loop2       ptp_hyperv  ram7        stdin       tty2         tty33       tty47       tty60       vcs2       vcsu3
btrfs-control hvc1        loop3       pts         ram8        stdout      tty20       tty34       tty48       tty61       vcs3       vcsu4
bus         hvc2        loop4       ram0        ram9        tty         tty21       tty35       tty49       tty62       vcs4       vcsu5
char        hvc3        loop5       ram1        random      tty0        tty22       tty36       tty5        tty63       vcs5       vcsu6
console     hvc4        loop6       ram10       rtc         tty1        tty23       tty37       tty50       tty7        vcs6       vfio
core        hvc5        loop7       ram11       rtc0        tty10       tty24       tty38       tty51       tty8        vcsa       vhost-net
cpu_dma_latency hvc6        mapper      ram12       sda         tty11       tty25       tty39       tty52       tty9        vcsa1     virtio-ports
cuse        hvc7        mem         ram13       sdb         tty12       tty26       tty4        tty53       ttyS0       vcsa2     vport0p0
disk        initctl     mqueue      ram14       sdc         tty13       tty27       tty40       tty54       ttyS1       vcsa3     vport0p1
dri         kmsg        net         ram15       serial      tty14       tty28       tty41       tty55       ttyS2       vcsa4     vport0p2
dxg         kvm         null        ram2        sg0         tty15       tty29       tty42       tty56       ttyS3       vcsa5     vsock
fd          log         nvram       ram3        sg1         tty16       tty3        tty43       tty57       ttyUSB0     vcsa6     zero
full        loop-control ppp         ram4        sg2         tty17       tty30       tty44       tty58       urandom     vcsu
```

Conclusión.

Con esta práctica aprendimos a compartir un puerto USB desde Windows hacia WSL 2 (Ubuntu) usando la herramienta usbipd-win. Esto nos permitió usar un adaptador serial CH340 dentro de Linux, como si estuviera conectado directamente. Así, ahora podemos trabajar con dispositivos como Arduino desde WSL, lo cual es útil para programación y pruebas de hardware sin salir del entorno Linux.