ASL · SIGNALLAB - RF EXECUTOR

One or Two Alfa Radios on Windows

Set up an **AWUS036ACM / MT7612U**, an **AWUS036AXML / MT7921U**, or both together with the tested SignalLab WSL2 kernel and USB/IP path. Authorized lab use only.

What you end up with

Ubuntu 24.04 under WSL2 running the exact **6.18.33.2-microsoft-standard-WSL2** kernel and matching module tree. Attach either supported Alfa through **usbipd-win**, or attach both and operate two independent monitor interfaces in the same WSL instance.

ACM · 0e8d:7612 · mt76x2u | AXML · 0e8d:7961 · mt7921u | one or both attached | monitor + bounded injection verified

Current tested path: WSL2 plus usbipd-win carries both radios. Simultaneous attach, firmware initialization, passive scans, and bounded bidirectional synthetic-frame injection passed. The MT7612U later disconnected after about 11 minutes of repeated monitor/capture work and recovered through software reattach. Treat the ACM USB/IP path as bounded-session engineering, not unattended service. The older VirtualBox path remains an ACM fallback and is covered near the end of this guide.

0. Choose the layout

ACM only	AWUS036ACM, USB `0e8d:7612`, driver `mt76x2u`. Monitor and bounded injection pass. The USB/IP long-duration test did not pass.
AXML only	AWUS036AXML, USB `0e8d:7961`, driver `mt7921u`. Monitor and bounded injection pass. This is the cleaner single-radio WSL path in the current evidence.
Both	Bind and attach both USB identities. The drivers create separate interfaces. Simultaneous initialization, passive scan, and bounded cross-radio injection pass.

Use direct USB 3 ports and USB 3 data cables. Do not use the old USB 2 cable, and avoid a shared hub during setup. Bus IDs and interface names change with ports and enumeration, so identify each adapter by USB identity and Linux driver instead of copying example names.

A. Prerequisites

- Windows 11 with WSL2 and Ubuntu 24.04.
- `usbipd-win`. The accepted workstation used version 5.3.0 with its matching `7.2.2.170484` host driver.
- At least 4 GB free for the kernel bundle and extracted module tree.
- An administrator PowerShell for the one-time USB bind. Normal attach commands do not require an elevated shell.
- The SignalLab kernel archive and its published checksum.

If WSL or Ubuntu is not installed, run these in an administrator PowerShell and restart Windows if asked:

```
wsl --install -d Ubuntu-24.04
wsl --update
winget install --exact --id dorssel.usbipd-win
```

B. Install the tested dual-driver WSL kernel

B1 · Download and verify

Download the dual-Alfa WSL kernel bundle and its SHA-256 file. Then verify it in PowerShell:

```
$bundle = "$env:USERPROFILE\Downloads\signallab-wsl-kernel-6.18.33.2-dual-alfa.tar.zst"
$expected = "99804e797c0b2f1894243768a88407e923c0468bf1fee61e7021c479441121ae"
$actual = (Get-FileHash -Algorithm SHA256 -LiteralPath $bundle).Hash.ToLowerInvariant()
if ($actual -ne $expected) { throw "SignalLab kernel checksum mismatch" }
```

B2 · Extract the Windows-side package

```
$extract = "$env:USERPROFILE\SignalLab-WSL"
$kernelDir = "$env:USERPROFILE\wsl-kernel-mt76"
New-Item -ItemType Directory -Path $extract,$kernelDir -Force | Out-Null
tar -xf $bundle -C $extract
$package = Join-Path $extract "SignalLab-Dual-Alfa-WSL-Kernel-6.18.33.2-20260827"
Copy-Item -LiteralPath "$package\bzImage" -Destination "$kernelDir\bzImage" -Force
```

B3 · Install the matching modules and firmware in Ubuntu

Open Ubuntu 24.04. Replace `YOURNAME` with the Windows account folder name:

```
sudo apt-get update
sudo apt-get install -y zstd iw usbutils tcpdump aircrack-ng

PKG=/mnt/c/Users/YOURNAME/SignalLab-WSL/SignalLab-Dual-Alfa-WSL-Kernel-6.18.33.2-20260827
sudo tar --zstd -xf "$PKG/modules-6.18.33.2-microsoft-standard-WSL2.tar.zst" -C /lib/modules
sudo install -D -m 0644 "$PKG/firmware/mediatek/WIFI_MT7961_patch_mcu_1_2_hdr.bin" \
  /lib/firmware/mediatek/WIFI_MT7961_patch_mcu_1_2_hdr.bin
sudo install -D -m 0644 "$PKG/firmware/mediatek/WIFI_RAM_CODE_MT7961_1.bin" \
  /lib/firmware/mediatek/WIFI_RAM_CODE_MT7961_1.bin
sudo depmod 6.18.33.2-microsoft-standard-WSL2
```

B4 · Point WSL at the kernel

On Windows, merge the following into `%USERPROFILE%\.wslconfig`. Preserve any other existing settings and keep only one `[wsl2]` section:

```
[wsl2]
kernel=C:\\Users\\YOURNAME\\wsl-kernel-mt76\\bzImage
```

Restart WSL and verify the exact kernel:

```
wsl --shutdown
wsl -d Ubuntu-24.04 -- uname -r
```

> **Required result:** `6.18.33.2-microsoft-standard-WSL2`. Do not attach either radio until that exact kernel and its matching module tree are active.

C. Bind and attach one or both adapters

C1 · Identify the current bus IDs

Plug each radio directly into a USB 3 port. In an administrator PowerShell:

```
usbipd --version
usbipd list
```

Find `0e8d:7961` for the AXML and `0e8d:7612` for the ACM. Record the current `BUSID` beside each one. The accepted workstation used `1-4` and `1-5`, but those are examples, not fixed values.

C2 · Bind once on Windows

Run only the command for the model you own, or run both for the dual layout:

```
# AXML / MT7921U
usbipd bind --busid AXML-BUSID

# ACM / MT7612U - forced sharing was required on the accepted workstation
usbipd bind --force --busid ACM-BUSID
```

Binding persists. It does not mean the radio is attached to WSL yet.

C3 · Start Ubuntu, then attach

Start Ubuntu and leave it running. In a normal PowerShell, attach the model you want. Run both attach commands for a two-radio session:

```
# AXML only, or first radio in the dual layout
usbipd attach --wsl --busid AXML-BUSID

# ACM only, or second radio in the dual layout
usbipd attach --wsl --busid ACM-BUSID

usbipd list
```

> **Required result:** every selected radio reads `Attached`. A WSL shutdown ends the attachment; the one-time bind remains, but the attach command must be run again for the next WSL session.

D. Map the interfaces and enter monitor mode

D1 · Identify interfaces by driver

Inside Ubuntu, do not assume `wlan0`, `wlan1`, or a saved `wlx...` name.

```
lsusb | grep -Ei '0e8d:(7612|7961)'
iw dev

for path in /sys/class/net/*; do
  [ -e "$path/phy80211" ] || continue
  iface=$(basename "$path")
  driver=$(basename "$(readlink -f "$path/device/driver")")
  printf '%-24s driver=%s address=%s\n' "$iface" "$driver" "$(cat "$path/address")"
done
```

The ACM interface reports `mt76x2u`. The AXML interface reports `mt7921u`.

D2 · Select one or both interfaces safely

```
interface_for_driver() {
  for path in /sys/class/net/*; do
    [ -e "$path/phy80211" ] || continue
    [ "$(basename "$(readlink -f "$path/device/driver")")" = "$1" ] || continue
    basename "$path"
    return
  done
}

ACM_IF=$(interface_for_driver mt76x2u)
AXML_IF=$(interface_for_driver mt7921u)
printf 'ACM=%s AXML=%s\n' "${ACM_IF:-not-attached}" "${AXML_IF:-not-attached}"
```

D3 · Put every attached radio on the same test channel

```
set_monitor() {
  IF=$1
  [ -n "$IF" ] || return
  sudo ip link set "$IF" down
  sudo iw dev "$IF" set type monitor
  sudo ip link set "$IF" up
  sudo iw dev "$IF" set channel 1
}

set_monitor "$ACM_IF"
set_monitor "$AXML_IF"
iw dev
```

With one radio, one interface must report `type monitor`. With both attached, both interfaces must report `type monitor` and channel 1 before a cross-radio test.

D4 · Passive verification

Use the interface variable for the model you are checking:

```
sudo timeout 15 tcpdump -i "$AXML_IF" -e -n -s 256 'type mgt'
# or
sudo timeout 15 tcpdump -i "$ACM_IF" -e -n -s 256 'type mgt'
```

For a two-radio capture, write separate files so the evidence remains attributable:

```
sudo timeout 20 tcpdump -i "$ACM_IF" -e -n -s 256 -w acm-monitor.pcap &
ACM_PID=$!
sudo timeout 20 tcpdump -i "$AXML_IF" -e -n -s 256 -w axml-monitor.pcap &
AXML_PID=$!
wait "$ACM_PID" "$AXML_PID"
```

> **Injection boundary:** setup is complete when the intended interfaces are attached, mapped to the correct drivers, in monitor mode, and capturing. Do not use generic injection tests against nearby access points. SignalLab's accepted transmit result used eight fixed synthetic data frames in each direction, locally administered addresses, no disruptive management/control frames, and no real-network target.

E. Stop cleanly and recover USB/IP

E1 · Return interfaces to managed mode

```
set_managed() {
  IF=$1
  [ -n "$IF" ] || return
  sudo ip link set "$IF" down
  sudo iw dev "$IF" set type managed
  sudo ip link set "$IF" up
}

set_managed "$ACM_IF"
set_managed "$AXML_IF"
```

E2 · Detach on Windows

```
usbipd detach --busid AXML-BUSID
usbipd detach --busid ACM-BUSID
```

Run only the command for a connected model. Detach before changing USB ports or shutting down a long test session.

E3 · MT7612U recovery

- If the ACM changes from `Attached` to `Shared (forced)`, run its attach command again. The accepted long-duration failure recovered this way without a physical power cycle.
- If Linux logs `rx urb failed: -32` followed by host `sendmsg failed, ret=-104`, stop the session, detach the ACM, and reattach it. Do not call the interrupted session stable.
- If Windows reports *Unknown USB Device (Set Address Failed)* or no longer lists `0e8d:7612`, physically power-cycle only the ACM and reconnect it to a direct USB 3 port.
- Do not reset a shared USB root hub to recover the radio; that can also reset unrelated storage devices.

F. VirtualBox fallback and dashboard

The downloadable executor kit builds the older `asl-rf-executor` VirtualBox VM with native USB passthrough and the Windows dashboard. Use that route when the ACM needs the dependable native-passthrough executor. It is not the current dual-radio path.

- **AWUS036ACM:** dependable in the VirtualBox executor with native USB passthrough.
- **AWUS036AXML:** bounded manual tests passed, but sustained monitor traffic reproduced VirtualBox xHCI completion-code-13 warnings and a Linux kernel panic. Keep its persistent VM USB filter disabled.
- **Both together:** use the WSL procedure in this guide. The VirtualBox route is not accepted as a simultaneous dual-radio configuration.
- **Dashboard:** the current dashboard reaches the VirtualBox VM over SSH. It does not control the WSL interfaces described above.

A physical USB device cannot be owned by WSL and VirtualBox at the same time. Detach it from one before assigning it to the other.

G. Quick reference

Validated kernel	`6.18.33.2-microsoft-standard-WSL2`
Kernel SHA-256	`99804e797c0b2f1894243768a88407e923c0468bf1fee61e7021c479441121ae`
ACM	AWUS036ACM · `0e8d:7612` · `mt76x2u` · forced bind on accepted host
AXML	AWUS036AXML · `0e8d:7961` · `mt7921u`
Host status	`usbipd list`
Linux status	`uname -r` · `lsusb` · `iw dev`
Attach	`usbipd attach --wsl --busid BUSID`
Detach	`usbipd detach --busid BUSID`
Dual-radio rule	Map by driver, not interface order; put both on the same channel before cross-capture.
ACM stability	Short bounded work passed; long-duration USB/IP acceptance failed after about 11 minutes.