



SWE 4211 Robot Wi-Fi Setup

1 Setting Up Wi-Fi on the Raspberry Pi

At some point during this lab, you will want to enable Wi-Fi on the Raspberry Pi. You can do this early in the lab or later; it does not matter when. Wi-Fi makes driving the robot easier, since it is no longer tethered. However, wired connections are easier to diagnose. On a wired connection you can easily ping devices and use Wireshark to monitor the network.

Your Pi can remember more than one Wi-Fi network. Once a network is saved, the Pi reconnects to it automatically whenever it is in range, including after a reboot. You can also return to a wired connection at any time, even after Wi-Fi is set up.

1.1 Choose your flow

This document has two flows. Pick the one that matches your situation.

Your situation	Use
Wi-Fi has never been set up on this Pi.	Flow A: First-time setup or repair
Wi-Fi was set up but is not working, and you want to start clean.	Flow A: First-time setup or repair
Wi-Fi works on at least one network, and you want the robot to also join another one (your apartment, home, or a different lab).	Flow B: Add another Wi-Fi network

Flow A deletes every saved Wi-Fi network on the Pi. Do not use it just to add a second network. Use Flow B instead.

Command conventions. Commands in bold blue are typed on the Pi. Commands in bold red are typed on your laptop. Replace anything in <angle brackets> with your own value, and do not type the brackets. Type commands exactly as shown. If you copy and paste them, check that quotation marks and dashes were not converted to "smart" characters.

2 Flow A: First-Time Setup or Repair

Use this flow if Wi-Fi has never been set up on your Pi, or if it is not working and you want to start over. Flow A removes all saved Wi-Fi networks, prepares the radio, connects to one network, and then moves you from the cable to Wi-Fi.

2.1 Before you begin

Start with your Pi connected by Ethernet and an SSH session open to it from your laptop. Every step through Section 2.6 is done over this wired session, so you never lose access while setting up Wi-Fi.

You will need:

- A Raspberry Pi running Raspberry Pi OS Trixie Lite, with a working SSH login over Ethernet.
- The SSID (network name) and passphrase of the lab Wi-Fi network, provided by your instructor.
- Your laptop connected to that same Wi-Fi network, so it can reach the Pi once the cable is removed.
- A place to record information. You will write down your Pi's wired IP address, Wi-Fi IP address, and hostname.

Before starting, record the wired IP address you are currently using to SSH to the Pi:

hostname -I

Wired IP address: _____



2.2 Clear existing Wi-Fi connections

Remove every saved Wi-Fi connection so your Pi starts from a clean state. This only removes wireless connections. Your wired Ethernet connection, and your SSH session, are left alone.

1. List the connections currently saved on the Pi:

nmcli connection show

Each row is one saved connection. The TYPE column shows wifi for wireless connections and ethernet for wired ones. Note how many Wi-Fi connections are listed. It may be zero.

2. Delete all saved Wi-Fi connections at once:

nmcli -t -f UUID,TYPE connection show | awk -F: ' \$2=="802-11-wireless" {print \$1}' | xargs -r sudo nmcli connection delete

This pipeline lists every connection by its unique ID (UUID) and type, keeps only the wireless ones, and deletes each one. (In this terse listing, Wi-Fi connections are reported as 802-11-wireless rather than wifi.) If there are no Wi-Fi connections, it does nothing.

3. Confirm that only wired connections remain:

nmcli connection show

You should see no rows with type wifi. A wired connection and possibly a loopback (lo) entry will still be listed. That is expected.

Do not delete the wired connection. If you do, you will likely lose your SSH session and need a keyboard and monitor to recover.

2.3 Prepare the Wi-Fi radio

The Pi keeps its Wi-Fi radio disabled until a country code is set, because legal radio channels differ by country. Set the country and make sure the radio is turned on.

1. Set the Wi-Fi country to the United States:

sudo raspi-config nonint do_wifi_country US

2. Make sure the radio is not blocked:

sudo rfkill unblock wifi

3. Check the status of the network devices:

nmcli device status

Look for the wlan0 row. Its STATE should be disconnected. If it says unavailable, repeat steps 1 and 2. If there is no wlan0 row at all, ask your instructor. Your Pi model may not have built-in Wi-Fi, or there may be a hardware failure.

2.4 Scan for available networks

Ask the Pi to scan for nearby wireless networks and find the lab network in the results.

1. Trigger a fresh scan:

sudo nmcli device wifi rescan

2. Wait about 5 seconds, then list what the Pi found:

nmcli device wifi list

Each row is one network. The most useful columns are:



Column	Meaning
SSID	The network name. Must be typed exactly, including capitalization and spaces.
CHAN	The radio channel. Channels 1–14 are 2.4 GHz; 36 and above are 5 GHz.
SIGNAL	Signal strength from 0 to 100. Above 50 is good.
SECURITY	The security type. WPA2 or WPA3 means a passphrase is required.

- Find the lab network in the list and record its details:

SSID: _____ Signal: _____ Security: _____

If the lab network does not appear, run the rescan again and wait a little longer. Some older Pi models only support 2.4 GHz and will not see 5 GHz-only networks.

2.5 Connect to the Wi-Fi network

Connect to the lab network using the SSID you recorded in Section 2.4. NetworkManager saves the connection under the same name as the SSID. It will reconnect automatically whenever the network is in range, including after a reboot.

- Connect, replacing <SSID> with the exact SSID. Keep the quotation marks, especially if the SSID contains spaces:

```
sudo nmcli --ask device wifi connect "<SSID>"
```

The `--ask` option makes nmcli prompt you for the passphrase. Type it and press Enter. The passphrase will not be shown on screen.

- Wait for the result. On success you will see a message similar to:

```
Device 'wlan0' successfully activated with '<uuid>'.
```

If you see an error such as Secrets were required, but not provided, the passphrase was wrong. Run the command again.

Note on passphrases. You may see guides that put the passphrase directly on the command line with password "...". That works, but the passphrase is then saved in your shell history (`~/.bash_history`) in plain text. Using `--ask` avoids this.

Either way, NetworkManager stores the passphrase in plain text in a root-only file under `/etc/NetworkManager/system-connections/`. Anyone with physical access to your microSD card can read it. Keep this in mind before saving your home network's passphrase on the Pi.

2.6 Verify the connection and record the Wi-Fi IP address

Confirm the Pi is on Wi-Fi and find the IP address it was given. This address is different from your wired address. It is the one you will use once the cable is unplugged.

- Check the device status:

```
nmcli device status
```

Both `eth0` and `wlan0` should now show connected.

- Show the Wi-Fi interface's address:

```
ip -4 addr show wlan0
```

Find the line beginning with `inet`. The address before the `/` is your Wi-Fi IP address. For example, in `inet 192.168.1.57/24`, the address is `192.168.1.57`.

- Record it:

Wi-Fi IP address: _____



4. Test that the Pi can reach the network over Wi-Fi specifically:

```
ping -c 4 -I wlan0 8.8.8.8
```

The -I wlan0 option forces the ping out through the wireless interface. You should see 4 replies and 0% packet loss. If the lab network blocks internet access, ping your laptop's IP address instead.

5. Check the Pi's hostname, which gives you a second way to reach it:

```
hostname
```

Hostname: _____

2.7 Switch from the wired connection to Wi-Fi

Prove the Wi-Fi connection works before removing the cable, then unplug and log in over Wi-Fi only. Testing first means that if something is wrong, you still have your wired session to fix it.

1. Leave your current wired SSH session open. On your laptop, open a second terminal window.
2. Make sure your laptop is connected to the same Wi-Fi network as the Pi.
3. From the second terminal, ping the Pi's Wi-Fi IP address recorded in Section 2.6:

```
ping -c 4 <wifi-ip-address>
```

On Windows, use ping -n 4 instead of ping -c 4.

4. Still in the second terminal, open a new SSH session to the Pi's Wi-Fi address, replacing <username> with your Pi login:

```
ssh <username>@<wifi-ip-address>
```

If this works, you now have two sessions to the same Pi: one wired, one wireless. If it does not work, stop here and go to Troubleshooting. Do not unplug the cable.

5. Type exit in the wired SSH session to close it.
6. Unplug the Ethernet cable from the Pi.
7. Return to your Wi-Fi SSH session and confirm it is still responsive:

```
nmcli device status
```

eth0 should now show unavailable and wlan0 should show connected. Your session survived because it never used the cable.

8. Close the session with exit, then reconnect from scratch to show you can reach the Pi wirelessly from a cold start:

```
ssh <username>@<wifi-ip-address>
```

You can also try connecting by hostname, which keeps working even if the IP address changes later:

```
ssh <username>@<hostname>.local
```

Connecting by hostname relies on mDNS. It works on most Linux, macOS, and current Windows systems, but some networks block it. If it fails, use the IP address.

You can return to a wired connection at any time by plugging the cable back in and using your wired IP address. Both connections can be active at once.



3 Flow B: Add Another Wi-Fi Network

Use this flow when Wi-Fi already works on at least one network and you want the robot to also join another one. Flow B saves the new network alongside the ones you already have. It does not disconnect your current session, and you do not need to be in range of the new network to add it.

3.1 Before you begin

You will need:

- A working SSH session to the Pi, over either Wi-Fi or Ethernet.
- The exact SSID and passphrase of the new network.
- The new network's security type. Flow B supports WPA2-Personal and WPA3-Personal networks, which use a single passphrase. Networks that ask for a username and password (WPA2-Enterprise, such as many campus networks) are not covered.

Do not use the Flow A connect command (nmcli device wifi connect) while you are logged in over Wi-Fi. It immediately moves wlan0 to the new network and drops your SSH session. If the passphrase is wrong, you will not be able to reach the Pi until you plug in a cable.

3.2 Review the saved networks

1. List the connections already saved on the Pi:

nmcli connection show

Note the names in the NAME column for rows of type wifi. Each name must be unique, so pick a different name for the new network in the next section.

2. If the network you want to add is already listed, you are done. If it is listed but its passphrase has changed, delete it and add it again:

sudo nmcli connection delete "<name>"

3.3 Add the new network

This saves the new network without connecting to it.

1. Store the passphrase in a temporary shell variable. You will be prompted for it, and it will not be shown on screen or saved in your shell history:

read -rsp "Passphrase: " PSK; echo

2. Add the connection. Replace <name> with a short label you choose (for example, Home) and <SSID> with the exact network name:

sudo nmcli connection add type wifi ifname wlan0 con-name "<name>" ssid "<SSID>" wifi-sec.key-mgmt wpa-psk wifi-sec.psk "\$PSK"

Type "\$PSK" exactly as shown, including the quotation marks. Your shell fills in the passphrase you typed in step 1. On success you will see a message similar to:

Connection '<name>' (<uuid>) successfully added.

If the network is WPA3-only, use wifi-sec.key-mgmt sae in place of wpa-psk. If the network is hidden (it does not broadcast its SSID), add 802-11-wireless.hidden yes to the end of the command.

3. Clear the temporary variable:

unset PSK



4. Confirm the new network is saved:

`nmcli connection show`

The new name should appear with type wifi. Your existing networks should still be listed.

3.4 Choose a preferred network (optional)

When more than one saved network is in range, the Pi joins the one with the highest priority. All networks start at priority 0. When priorities are tied, the network used most recently wins. To make a network preferred, give it a higher number:

`sudo nmcli connection modify "<name>" connection.autoconnect-priority 10`

To view the priorities of all saved networks:

`nmcli -f NAME,TYPE,AUTOCONNECT-PRIORITY connection show`

3.5 Test the new network

How you test depends on whether you are in range of the new network right now.

If you are in range now and connected by Ethernet:

1. Switch wlan0 to the new network:
`sudo nmcli connection up "<name>"`
2. Confirm wlan0 shows connected, and record the new Wi-Fi IP address:
`nmcli device status`
`ip -4 addr show wlan0`
3. Switch back to your usual network if needed, using its name from the saved list:
`sudo nmcli connection up "<previous-name>"`

Only run nmcli connection up while connected by Ethernet. Over Wi-Fi it drops your session, as described above.

If you are not in range now: nothing more is needed. When the Pi is powered on within range of the new network, it joins automatically. Continue with the next section to find it.

3.6 Find the Pi on the new network

The Pi will have a different IP address on each network. To find it after the Pi joins the new network, use any of these:

- Connect by hostname: `ssh <username>@<hostname>.local` (requires mDNS, which some networks block).
- Look up the Pi in the router's list of connected devices, if you have access to the router.
- Plug in an Ethernet cable, or a keyboard and monitor, and run `hostname -I` on the Pi.

Record the address for this network: _____

If the Pi does not join the new network, connect to it by Ethernet and run `nmcli device wifi list` to confirm the network is visible. Then delete the saved connection and repeat Section 3.3, checking the SSID and passphrase carefully.



4 Reference

4.1 Troubleshooting

Most problems come from the radio being blocked, a mistyped SSID or passphrase, or the laptop and Pi being on different networks.

Symptom	Likely cause	What to try
wlan0 shows unavailable	Country not set or radio blocked	Repeat Section 2.3. Check with rfkill list; Wi-Fi should show Soft blocked: no.
No wlan0 device at all	Pi has no built-in Wi-Fi	Ask your instructor about a USB Wi-Fi adapter.
Lab network missing from scan	Out of range, 5 GHz-only network, or scan too soon	Run <code>sudo nmcli device wifi rescan</code> , wait 10 seconds, and list again. Move closer to the access point.
Secrets were required, but not provided	Wrong passphrase	Flow A: run the connect command again and retype carefully. Flow B: delete the connection and repeat Section 3.3.
No network with SSID ... found	SSID typed incorrectly	Copy the SSID exactly from <code>nmcli device wifi list</code> , including capitalization and spaces.
Connect command asks for a username, or fails on a campus network	Network uses WPA2-Enterprise (username and password login)	The commands in this document do not support this. Ask your instructor for the lab network settings.
Pi shows connected, but the laptop cannot ping or SSH to it	Laptop is on a different network, or the network blocks devices from reaching each other (common on guest and public Wi-Fi)	Confirm both devices are on the same SSID. If they are, the network may isolate clients; use the lab network or a wired connection.
SSH session froze after running a connect or connection up command	wlan0 moved to a different network	Reconnect over Ethernet. Use Flow B to add networks while on Wi-Fi.
ssh to <hostname>.local fails	mDNS blocked or not supported	Use the IP address instead.
Pi never joins a network added in Flow B	Wrong SSID, wrong passphrase, or wrong security type	Over Ethernet, check <code>nmcli device wifi list</code> . Delete and re-add the connection. For WPA3-only networks, use <code>sae</code> .

4.2 Command quick reference

Commands below run on the Pi.

Task	Command
List saved connections	<code>nmcli connection show</code>
Show device status	<code>nmcli device status</code>
Delete one saved connection	<code>sudo nmcli connection delete "<name>"</code>
Delete all saved Wi-Fi connections (Flow A)	<code>nmcli -t -f UUID,TYPE connection show awk -F: '\$2=="802-11-wireless" {print \$1}' xargs -r sudo nmcli connection delete</code>
Set Wi-Fi country	<code>sudo raspi-config nonint do_wifi_country US</code>



Task	Command
Unblock the Wi-Fi radio	<code>sudo rfkill unblock wifi</code>
Scan for networks	<code>sudo nmcli device wifi rescan</code>
List nearby networks	<code>nmcli device wifi list</code>
Connect to a network now (Flow A; drops any current Wi-Fi session)	<code>sudo nmcli --ask device wifi connect "<SSID>"</code>
Connect to a hidden network now	<code>sudo nmcli --ask device wifi connect "<SSID>" hidden yes</code>
Save a network without connecting (Flow B)	<pre>read -rsp "Passphrase: " PSK; echo sudo nmcli connection add type wifi ifname wlan0 con-name "<name>" ssid "<SSID>" wifi-sec.key-mgmt wpa-psk wifi- sec.psk "\$PSK" unset PSK</pre>
Switch to a saved network	<code>sudo nmcli connection up "<name>"</code>
Disconnect without forgetting	<code>sudo nmcli connection down "<name>"</code>
Set a preferred network	<pre>sudo nmcli connection modify "<name>" connection.autoconnect-priority 10</pre>
Show network priorities	<code>nmcli -f NAME,TYPE,AUTOCONNECT-PRIORITY connection show</code>
Show the Wi-Fi IP address	<code>ip -4 addr show wlan0</code>
Show all IP addresses	<code>hostname -I</code>
Show the hostname	<code>hostname</code>
Menu-based alternative	<code>sudo nmtui</code>