

mmWave 1843 Installation

Koushik K
Master's Thesis

December 2025

Step 1: Prepare the SD card

1. Download the RPi Imager for Windows. <https://www.raspberrypi.com/software/>
2. Insert the SD card using a card reader and format it if necessary.
3. Open the RPi Imager:
 - **Device:** Raspberry Pi 4 or 5 (as applicable)
 - **OS:** Ubuntu 20.04 (General-purpose)
 - **Storage:** Select your SD card
4. Flash the OS and include your Wi-Fi SSID, password (user-definable), and also enable SSH during configuration.

Step 2: Boot and Set Up Ubuntu on the Raspberry Pi

1. Insert the SD card into the Raspberry Pi.
2. Connect a keyboard, mouse, monitor, and power source.
3. Log in using your Ubuntu credentials.

Step 3: Configure Wi-Fi (Hotspot)

1. On your laptop or mobile device, create a hotspot using the same SSID and password as in Step 1.4.
2. On the Raspberry Pi, open a terminal and run:

```
sudo nano /etc/netplan/50-cloud-init.yaml
```

Edit the file to include your hotspot credentials (only the password needs to be updated) as shown fig 1.

3. Apply network settings:

```
sudo netplan apply
```

Step 4: Install Desktop Environment

```
sudo apt update
sudo apt upgrade -y      (sudo reboot if an error occurs)
sudo apt install ubuntu-desktop -y
sudo reboot
```

```

yaml

network:
  version: 2
  renderer: networkd
  wifis:
    wlan0:
      optional: true
      access-points:
        "YOUR_WIFI_NAME":
          password: "YOUR_WIFI_PASSWORD"
      dhcp4: true

```

Figure 1: configure wifi

Step 5: Install ROS Noetic (ROS-Base)

Follow the ROS Noetic installation guide. <https://wiki.ros.org/noetic/Installation/Ubuntu>

- Complete Sections 1.2 and 1.3.
- In Section 1.4, install **ROS-Base** (bare-bones only).
- If “unable to fetch packages” appears, run: `sudo apt update`.
- Do all steps in 1.5 except the zsh configuration (can be skipped).
- Complete Step 1.6.

Step 6: Create ROS Workspace

Follow the official ROS instructions to create your workspace (`catkin_ws`).<https://automaticaddison.com/how-to-create-a-ros-workspace/>

Step 7: Add Radar ROS Package

1. Download the radar package ZIP file and extract it. <https://github.com/koushikkalyani/Masters-Thesis/blob/main/src.zip>
2. Copy its contents from `src/` into `catkin_ws/src/`.
3. Run: `catkin_make`.

Step 8: Fix Compiler Errors

If C/C++ compiler error occurs:

```
sudo apt update
sudo apt install g++
```

If PCL error occurs:

```
sudo apt update
sudo apt install libpcl-dev
sudo apt install ros-noetic-pcl-ros
```

Then edit `CMakeLists.txt` in `catkin_ws/src/ti_mmwave_ropkg`: Add at the top:

```
set(CMAKE_CXX_STANDARD 17)
```

Rebuild:

```
cd ~/catkin_ws  
catkin_make
```

Step 9: Configure Launch File to Record Data

In `catkin_ws/src/ti_mmwave_rosPKG/launch/awr1843boost_test.launch`, add:

```
<!-- Record bag -->
<node name="bag_record" pkg="rosbag" type="record"
      args="record -0 /home/rootname /ti_mmwave/radar_scan" />
```

Ensure there is a space before /ti_mmwave.

Step 10: Launch the Radar Node

Before launching, ensure your system time is correct (connect to Wi-Fi to auto-sync).

1. Connect the radar module to the Raspberry Pi and power it.
 2. Go to `catkin_ws/src/ti_mmwave_ropkg/launch/`.
 3. Run: `roslaunch awr1843boost_test.launch`.
 4. If an error appears, close the terminal and run the launch command again.
 5. Data will be recorded in `.bag` format.

Step 11: Convert .bag File to .txt

```
python3 /home/ACPS/Documents/code.py /home/ACPS/Documents/filename.bag filename.txt /ti_mmwave/radar_scan
```

Here, `filename.bag` is the collected radar file and `filename.txt` is user-defined.

converting code "code.py"

```
import os
import argparse
import rospy
import rosbag

def main():
    parser = argparse.ArgumentParser(description="Extract radar data from a ROS bag.")
    parser.add_argument("bag_file", help="Input ROS bag.")
    parser.add_argument("output_file", help="Output .txt file.")
    parser.add_argument("radar_topic", help="Radar topic.")

    args = parser.parse_args()

    print("Extract radar data from %s on topic %s into %s" % (args.bag_file, args.radar_topic,
                                                               args.output_file))

    bag = rosbag.Bag(args.bag_file, "r")
    with open(args.output_file, 'w') as f:
        for topic, msg, t in bag.read_messages(topics=[args.radar_topic]):
            # Access frame_id from the message header
            frame_id = msg.header.frame_id
            # Write frame_id, x, y, z to the output file
            f.write("%s,%d,%d,%f,%f,%f,%f,%f,%f,%f,%f,%f,%f,%f,%f,%f,%f,%f\n" %
                    (frame_id, msg.radar_frame_idx, msg.point_id, msg.x, msg.y, msg.z, msg.range,
                     msg.velocity, msg.doppler_bin, msg.bearing, msg.intensity,
                     msg.intensity_snr_noise, msg.noise, msg.time_human))
```

```
    print("Wrote radar data to %s" % args.output_file)

if __name__ == "__main__":
    main()
```

1 Installing TigerVNC on Raspberry Pi (One-Time Setup)

Follow the steps below to install and configure TigerVNC on a Raspberry Pi running Ubuntu 20.04. Ensure no VPN is active during setup.

1. Update System Packages

1. Run the following commands:

```
sudo apt update && sudo apt upgrade -y
```

2. Install the XFCE desktop environment:

```
sudo apt install xfce4 xfce4-goodies -y
sudo reboot
```

2. Configure Wi-Fi (Hotspot)

1. Edit the netplan configuration:

```
sudo nano /etc/netplan/50-cloud-init.yaml
```

Add your mobile hotspot details (SSID and password), then apply:

```
sudo netplan apply
```

3. Install Required Packages

1. Install Firefox (optional):

```
sudo apt install firefox -y
```

2. Install TigerVNC:

```
sudo apt install tigervnc-standalone-server tigervnc-common -y
```

3. Ensure XFCE components are installed:

```
sudo apt update
sudo apt install xfce4 xfce4-goodies xfce4-terminal -y
```

4. Configure VNC Startup Script

1. Edit the xstartup file:

```
nano ~/.vnc/xstartup
```

2. Add the following:

```
#!/bin/bash
unset SESSION_MANAGER
unset DBUS_SESSION_BUS_ADDRESS
xrdb $HOME/.Xresources
startxfce4 &
```

3. Make it executable:

```
chmod +x ~/.vnc/xstartup
```

5. Enable SSH Login

1. Edit SSH configuration:

```
sudo nano /etc/ssh/sshd_config
```

2. Ensure the following lines are set:

```
PasswordAuthentication yes
PubkeyAuthentication yes
```

3. Restart SSH:

```
sudo systemctl restart ssh
```

6. Find Raspberry Pi IP Address (troubleshooting step, skip to step 7)

1. Connect both your laptop and Raspberry Pi to the same mobile hotspot.
2. On Windows, open Command Prompt and run:

```
ipconfig
arp -a
```

3. Identify the Raspberry Pi using the MAC prefix b8:27:eb or dc:a6:32.

7. Connect via PuTTY and Start VNC

1. Open PuTTY → select SSH.
2. Enter the Raspberry Pi hostname (you can also enter IP address).
3. Go to Connection → SSH → Tunnels:
 - Source port: 5901
 - Destination: localhost:5901Click **Add**, save the session, and open PuTTY.
4. Log in with your Raspberry Pi username and password.

5. Set a VNC password:

```
vncpasswd
```

6. Start the VNC server:

```
vncserver
```

7. Open VNC Viewer and connect to:

```
localhost:1
```

Enter your VNC password to access the Raspberry Pi desktop.

data transfer

In the same laptop as putty and VNCviewer install WinSCP <https://winscp.net/eng/download.php>. Login procedure is similar to putty by credentials username and password, remaining are kept default. It opens a portal where left side sapce is your laptop and right side space is ubuntu, simply drag data.txt and drop it in laptopspace.