Implementation Guide for building a Raspberry PI OS-based ADS-B Application Server

This documentation is provided to assist with the quick provisioning and configuration of a Raspberry Pi with your preferred Raspberry Pi OS distribution, and then the installation of software to run an ADS-B application server.

Raspberry Pi Operating System

Required Components:

- Raspberry Pi card with built-in Bluetooth and Wi-Fi
- An appropriate case with cooling fan

Installing the Operating System

The operating system installation process is implemented by performing the following steps and is essentially complete at the end of step 20, however additional steps are included to help you implement two way network file sharing, and enable audio and Bluetooth audio capabilities. These last two audio related capabilities simply don't work "out of the box" and require quite a bit of configuration file editing to get everything to work properly.

The instructions here are for a Raspberry Pi 4 running the Raspberry Pi OS and may require adaption on different operating systems or earlier versions of the card.

1. Download your preferred Raspberry Pi OS distribution from one of the following sites:
   - Official Raspberry Pi OS at https://www.raspberrypi.org/downloads
   - 64-bit Raspberry Pi OS Beta here
   - Ubuntu Mate LTS at https://ubuntu.com/download/raspberry-pi
   - Twister OS (Windows & iOS look-and-feel) at https://raspbian-x.com (also see here)
2. Download and install either *NOOBS* (https://www.raspberrypi.org/downloads/noobs) or *balena Etcher* (https://www.balena.io/etcher), or an equivalent image flashing software
3. Use your chosen image flashing software to flash the OS file from step 1 to a microSD card (~10 mins). On completion it will automatically be ejected at the end of the process. You will need to re-insert the SD card if you wish to perform the optional steps 4 through 6 below.
4. For security reasons, SSH access is disabled by default. To enable SSH, create an empty file on the /boot partition of the SD card with the filename of **ssh** only (with no file extension). When this file is present, SSH will be automatically enabled plus you will be able to use the **scp** command to copy files between ssh enabled devices.

5. If you want to overclock your Raspberry Pi, refer to the topic *Overclocking Profiles* below before continuing. Only the 1.7 GHz overclock profile worked for me, the rest either hung the Pi device or Wi-Fi didn't work – go figure.
6. If you would like to specify your Wi-Fi parameters in advance, create a new file called wpu_supplicant.conf on the /boot partition of the SD card and enter the following lines:
```
country=AU
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1
network={
 ssid="Enter_YOURSSID_here"
 scan_ssid=1
 psk="Enter_YOURPASSWORD_here"
 key_mgmt=WPA-PSK
}
```
Replace the *ssid* and *psk* variables with your Wi-Fi access point's details. Also ensure the *country* variable is correct for your country (see ISO 3166-1 alpha-2 in Wikipedia).
7. Save any files and eject the SD card from your computer.
8. If you have an Access Control List (ACL) whitelist defined on your Router/Wi-Fi Access Point, you might want to temporarily *disable* that at this point to allow the Pi to connect wirelessly, and then later when you have finished configuring the OS you can then run the **`ifconfig`** command in a Terminal session to determine the new device's Wi-Fi adapter's mac address. When discovered, that mac address can then be added to the ACL whitelist and the ACL re-activated on your Router/Access Point.
9. Insert the flashed microSD card into the Raspberry Pi (RPi) together with the Ethernet, HDMI, USB Mouse & Keyboard, and power cables; and power on the RPi.
10. On power up, step through the *Welcome to Raspberry Pi* process answering questions appropriately (ensure you enter a unique secure password) and let the software update process run its course (don't worry if you get Wi-Fi access or software update errors here as we will revisit those issues in later steps).
11. Click the *Restart* button at the end of the setup process to reboot the RPi to allow it to implement all your changes.
12. When the Raspberry Pi OS UI re-appears, start a Terminal session (or press CTRL+ALT+T as a keyboard shortcut)
13. To enable Windows Remote Desktop (RDP) to remotely access the Raspberry Pi OS GUI it will be necessary to install a remote desktop client server on the RPi OS (such as XRDP), as follows:
```
$ sudo apt-get install tightvncserver xrdp -y
```
14. Optional: From this point on you may wish to continue the rest of the configuration process by logging in remotely using Windows Remote Desktop (RDP). This will allow you to disconnect all the cables from the RPi, except for the power and Ethernet cables (don't forget to shut down the Pi before doing this) and remove that mess of mouse, keyboard and cables from the desk. Now using a remote desktop you will be able to cut and paste commands from this document directly into the RPi Terminal session, speeding up the configuration process.

15. In a Terminal session run the following commands to ensure the software update process worked correctly during the setup process:
    $ sudo apt-get update && sudo apt-get upgrade –y
16. Optional: If you were previously unable to connect to a Wi-Fi access point, ensure your access point is broadcasting its SSID and has had its Access Control List (ACL) whitelist updated with this Pi's mac address; then click on the network icon in the top right of the UI, ensure Wi-Fi is turned On and select the desired access point from the drop down list.
17. Test the speed of your SD card by clicking the Raspberry Pi icon in the upper left-hand corner of the UI and then choose
    Accessories -> Raspberry Pi Diagnostics
18. Click the Raspberry Pi icon in the upper left-hand corner of the UI and then choose **Preferences –> Raspberry Pi Configuration**
    On the *System* tab, review the options. Here you can elect to boot to CLI instead of the Desktop, set the system hostname, and can disable auto-login to the Desktop. For added security, it is recommended you uncheck this last setting, so that when the system reboots it will prompt for a username and password to log in.
    Click on the *Interfaces* tab and select the *Enable* the **SSH** radio button. SSH will enable remote CLI access to the Raspberry Pi (i.e. via PuTTY or PowerShell in Windows). If required the VNC interface is used to enable you to open a remote desktop using a VNC Viewer.
    Click on the *Performance* tab and change the GPU Memory to **512**. Click **OK** to finish.
19. If you have a large SD card, you can expand the usable space on the card by running the following command:
    $ sudo raspi-config
    Then, in the GUI interface select "*Advanced Options*" followed by "*Expand Filesystem*" from the menu. Also in *Advanced Options* you might want to select *"Audio"* and choose your preferred audio device. The device will offer the chance to reboot when the process has completed. Log back in.
20. If the RPi is mounted in an ARGON ONE case, enter and run the following command to install the case's Power Button and Fan Control functionality and reboot to apply the changes:
    $ curl https://download.argon40.com/argon1.sh | bash && sudo reboot
    When the Raspberry Pi OS UI starts again and if the RPi is mounted in an ARGON ONE case you will notice two new icons on the screen. Double click on the *Argon One Configuration* icon and when prompted, click on the **Execute** button. When prompted, press the **Y** key to continue. Select fan mode **2** (recommended) and enter the following fan speeds: 55c = **10**%, 60c = **55**%, 65c = **100**%
21. If you would like to enable other Windows computers to access directories on your Raspberry Pi you will need to run the following command to install the Samba package:
    $ sudo apt-get install samba samba-common-bin -y
    See the topic *Configuring Samba for Windows Computers* below for examples of using Samba.
22. If you would like to have access to networked Windows Shared Folders, run the following command to ensure CIFS support is installed:
    $ sudo apt-get install cifs-utils -y
    See the topic *Mounting Other File Systems* below for examples of how to attach to external file systems.

23. If you are looking for a replacement over the native Chromium browser on the Pi and would like to install the popular Mozilla Firefox browser, run the following command:
    $ sudo apt-get install firefox-esr
24. Click on and make sure the volume control (the speaker icon in top right of window) is turned up fully. Now, test the sound card by plugging in some headphones and using the following command:
    $ aplay /usr/share/sounds/alsa/Noise.wav
    If for any reason you do not hear any sound try using the **raspi-config** command to alter the *Audio* settings in *Advanced Options* from the Automatic to Headphones setting, and retry. The Noise.wav file plays as a brief static sound and finishes when control returns to the terminal. If this still does not work, try running the following audio "glitch" workaround:
    $ sudo apt-get remove pulseaudio -y
    Now if that comes back indicating that pulseaudio is not installed, you will need to install it with:
    $ sudo apt-get install pulseaudio -y
    $ sudo reboot
    After the reboot you will need to re-run the removal command you ran earlier:
    $ sudo apt-get remove pulseaudio -y
    Test if you have sound now, it might not be working just yet, as we have a couple more commands to run but you should hear something:
    $ aplay /usr/share/sounds/alsa/Noise.wav
    OK, a few more steps to complete now that pulseaudio is removed, we are going to run the following
    $ sudo apt autoremove
    $ sudo /etc/init.d/alsa-utils reset
    Try testing the sound one last time before rebooting to ensure you can hear a sound with the *aplay* command:
    $ aplay /usr/share/sounds/alsa/Noise.wav
    ….and reboot:
    $ sudo reboot

Bluetooth

Before you can get connected to any Bluetooth speakers or headphones, you will need to install Pulse Audio and its associated Bluetooth module. Pulse Audio is a sound server that receives audio input from multiple channels and filters them through to one single output or sink, as it's known. You will then need to apply a fix that helps to use audio files from the default *pi* user, without having to elevate privileges for each bluetooth action.

Simply use the following commands to add user *pi* to the bluetooth group and reboot the device:

$ sudo usermod -G bluetooth -a pi
$ sudo reboot

Next you will need to install Bluealsa and the Pulseaudio Module. This module will be useful later to:

$ sudo apt-get install bluealsa pulseaudio

There seems to be a pulse audio Bluetooth module issue where module-bluetooth-discover only works when started after the X11 session is up. The following workaround seems to fix this:

Edit the file */etc/pulse/default.pa*

$ sudo nano /etc/pulse/default.pa

…and comment out (by placing an # at the beginning of the line) the following line:

#load-module module-bluetooth-discover

Now edit the file */usr/bin/start-pulseaudio-x11*

$ sudo nano /usr/bin/start-pulseaudio-x11

…and after the lines:

```
if [ x"$SESSION_MANAGER" != x ] ; then
/usr/bin/pactl load-module module-x11-xsmp "display=$DISPLAY
session_manager=$SESSION_MANAGER" > /dev/null
fi
```

…add the following line:

/usr/bin/pactl load-module module-bluetooth-discover
…and save the file.

Other issues still exist and to solve for these you need to edit the Bluetooth configuration files (so buckle up for the ride):

$ sudo nano /etc/bluetooth/main.conf

…and set the *Controller* statement to **Controller = le**

Unfortunately when you check the Bluetooth service starting status (by running the **$ sudo systemctl status bluetooth.service**) you will discover additional issues which can be fixed by changing some configurations filee; first edit the file */lib/systemd/system/bluetooth.service*:

$ sudo nano /lib/systemd/system/bluetooth.service

…and add some additional options to the end of the "*ExecStart=/usr/lib/bluetooth/bluetoothd*" statement in the **[Service]** section of the file. It should be changed to look like this:

ExecStart=/usr/lib/bluetooth/bluetoothd **--compat --noplugin=sap -E**

Then edit the */lib/systemd/system/bthelper@.service* file:

sudo nano /lib/systemd/system/bthelper@.service

…and add the additional **bolded** statement to the **[Service]** section so that it looks like this:

```
[Service]
Type=simple
ExecStartPre=/bin/sleep 2
ExecStart=/usr/bin/bthelper %I
```

Now reboot:

$ sudo reboot

Check that Pulseaudio is running:

$ ps aux | grep pulseaudio

If the command returns just two lines:

```
pi@raspberrypi:~ $ ps aux | grep pulseaudio
pi 538 0.0 0.0 7348 504 pts/0 S+ 22:01 0:00 grep --color=auto pulseaudio
```

…then you will need to manually launch pulseaudio by running:

$ pulseaudio –start

Two utilities are used to manage Bluetooth and associated devices: *bluetoothctl* is a shell for managing Bluetooth and *hciconfig* is used to configure HCI devices.

$ sudo hciconfig –a → should display UP RUNNING
$ bluetoothctl

Now power on and pair the Bluetooth speakers:

[bluetooth]# help → to see list of available commands
[bluetooth]# power on
[bluetooth]# agent on
[bluetooth]# scan on
[bluetooth]# devices → to display mac address of discovered Bluetooth devices
[bluetooth]# pair *[mac address of your chosen device]*
[bluetooth]# paired-devices → to display paired devices
[bluetooth]# trust *[mac address]*
[bluetooth]# connect *[mac address]*
[bluetooth]# exit

[more stuff to come…]

PiAware

Required Components:

- Raspberry Pi provisioned with appropriate OS
- FlightAware or similar Software Defined Radio (RTL-**SDR**) USB dongle based on the RTL2832U chipset to receive 1090 MHz ADS-B and MLAT transmissions
- A suitable 1090 MHz antenna. A simple antenna is often provided with the RTL-SDR dongles; however for a better reception range see the following Youtube videos about building your own specialised antennas:
    - A cheap antenna - https://www.youtube.com/watch?v=ffWmr6p0dho
    - Ground Plane Spider Antenna - https://www.youtube.com/watch?v=quRwIzIAzfM
    - Franklin Antenna - https://www.youtube.com/watch?v=6N_4I10WWII
    - Coaxial Colinear (CoCo) Antenna - https://www.youtube.com/watch?v=kcDEi_YIOeU, https://www.youtube.com/watch?v=zMoKs1eiyO4, https://www.youtube.com/watch?v=TkUYdCPFXXs&t=126s, https://www.youtube.com/watch?v=7AovZTQfgbo&t=7s
- Dump1090 - open source code app to decode Mode S ADB-S signals from RTLSDR dongles into textual information and also feed that same data into the FlightAware PiAware
- FlightAware PiAware - web server app on the Raspberry Pi that can be interrogated from any attached device running a web browser to dynamically graphically display the locations on a map of aircraft broadcasting ADS-B information

Installing PiAware:

The following commands will download and install PiAware and any required dependencies on your Raspberry Pi:

```
wget https://flightaware.com/adsb/piaware/files/packages/pool/piaware/p/piaware-support/piaware-repository_3.8.1_all.deb
```

```
sudo dpkg -i piaware-repository_3.8.1_all.deb
sudo apt-get update
sudo apt-get install piaware
```

This will enable automatic and manual (web-based, via your request) PiAware software updates. These updates are disabled by default. To leave updates disabled, skip this step:

```
sudo piaware-config allow-auto-updates yes
sudo piaware-config allow-manual-updates yes
```

Now install dump1090:

```
sudo apt-get install dump1090-fa
```

Once you have finished installing and configuring the packages, reboot your Raspberry Pi to ensure that everything starts correctly.

```
sudo reboot
```

Whilst rebooting, go to http://latlong.net and enter your address to obtain and take note of the Latitude and Longitude. After reboot, edit the dump1090 configuration file:

```
sudo nano /etc/default/dump1090-fa
```

Add the above coordinates to the end of the RECEIVER_OPTIONS statement in the dump1090 configuration file:

```
RECEIVER_OPTIONS="--device-index 0 --gain -10 --ppm 0 --lat -33.645149 --lon
151.105865"
```

Restart dump 1090-fa to effect the change:

```
sudo systemctl restart dump1090-fa
```

From any computer now use your browser to navigate to: `http://[your_rapberrypi]:8080`

Overclocking Profiles

The following profiles can be used to overclock the Raspberry Pi ***only*** if you have active cooling on your device. Use NotePad++ to edit the *config.txt* file in the SD card's Boot partition and then copy & paste the following text into the end of the file. Uncomment one of the two overclocking profiles before saving the file and loading the SD card into the Pi.

 # ## WARNING: Ensure that you have sufficient cooling before overclocking your device ##

 # Copy one of the following overclocking profile text blocks (below) into the /boot/config.txt file

 # If the OS hangs or panics for any reason, re-edit to try another profile, and reboot.

 # Overclock CPU 1,700 MHz (1.7 GHz)

```
# over_voltage=2
# arm_freq=1700
```

 # Overclock CPU 2,000 MHz (2 GHz), Video 750 MHz

```
# over_voltage=6
# arm_freq=2000
# gpu_freq=750
```

If the operating system doesn't load, re-edit the *config.txt* file and try a lower overclocking profile, save the file and reboot the Pi.

Use the following table to determine what *over voltage* and *arm freq* overclocking parameters work together:

Clock (MHz)	Overvoltage	Vcore	Max temp. (C \ F)	Power (Watts)	Performance increase	Notes
1500	0	0.8625	82 | 180	7	—	
1600	1	0.8875	80 |176	7.6	6.6%	
1700	2	0.9125	78 | 172	8.3	13.3%	
1800	3	0.9375	77 | 170	8.9	20%	
1900	4	0.9625	75 |167	9.5	26.6%	
2000	6	1.0125	72 | 162	11	33.3%	
2100	6	1.0125	72 | 162	11	40%	

Mounting Other File Systems

Any Windows or NAS shared folders can be *temporarily* mounted until the next reboot/shutdown in Raspberry Pi OS as in the following example:

```
$ sudo mkdir /mnt/WinTemp
$ sudo chmod 777 /mnt/WinTemp
$ sudo mount.cifs //[host name or ip addr]/Temp /mnt/WinTemp -o
username=[windowsuser],password=[password]
```

Creating a permanent mount point using fstab

Now that we have mounted a network directory, wouldn't it be nice if your Raspberry Pi can mount these directories automatically every time we boot up? In order to mount the network shares on each boot up, you will need to modify the /etc/fstab file.

$ sudo nano /etc/fstab

There may be entries in there already and all you will need to do is add an entry to the end of the file as in the following example:

```
//192.168.1.7/ShareFile  /mnt/sharefile  cifs
username=yourWindowsUser,password=yourWindowsPassword,iocharset=utf8,sec=n
tlm  0  0
```

Save the fstab file and test it by running the following in a terminal window:

$ sudo mount -a;

If no errors are present, then you've successfully added the configuration needed. At this point it may be worth rebooting the Raspberry Pi for good measure and to fully test the change.

Configuring Samba for Windows Computers

To share a folder from a Raspberry Pi with a Windows computer you need to install Samba software on the Raspberry Pi. The Samba software package implements the SMB protocol and provides support for the Windows naming service (WINS) and for joining a Windows Workgroup. If you didn't install this earlier, you can do that now by logging into your Raspberry Pi and running the following command from a Terminal window:

$ sudo apt-get install samba samba-common-bin

After installation configure the software by opening the file `/etc/samba/smb.conf` using the command:

$ sudo nano /etc/samba/smb.conf

Read through the file and make sure you have the following parameters set:

workgroup = WORKGROUP
wins support = yes

You can use anything as your workgroup name as long as it is alphanumerical and matches the workgroup you would like to join. The default workgroup in Windows is WORKGROUP.

Setup the Raspberry Pi OS folder to share

Next step is to create the folder you would like to share. To create a folder called "share" in your home directory do the following:

$ mkdir ~/share

With the folder created you can now tell the Samba software to share it on the network. If not already open in an editor, open the file `/etc/samba/smb.conf` using the command:

$ sudo nano /etc/samba/smb.conf

Scroll to the bottom and add the following:

```
[PiShare]
    comment=Raspberry Pi Share
    path=/home/pi/share
    browseable=Yes
    writeable=Yes
    only guest=no
    create mask=0777
    directory mask=0777
    public=no
```

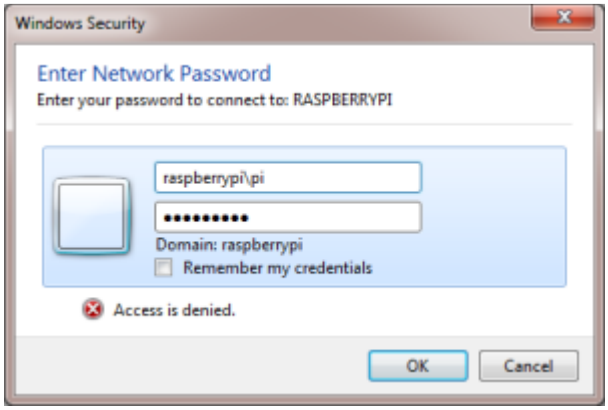
Notice how we tell Samba that public access is not allowed via “public=no” – this means that anyone wanting to access the shared folder must first login with a valid Raspberry Pi OS userid and password.

In this case the valid user is the user called “pi”. To let Samba know that “pi” is a network user run the command:

```
$ sudo smbpasswd -a pi
```

And enter pi’s password twice (default: raspberry).

At this point you can now login to the share from a Windows computer – use Domain: ***raspberrypi***, User: ***pi*** and Password: ***raspberry*** (unless you changed the password), as shown in the example below:



If you do not want to deal with logging in you can always make the share publicly available by changing the config file to say:

```
public=yes
```

However please note that this is extremely dangerous since anyone will be able to access, modify and delete your files.

Updating Raspberry Pi OS and Installed Application Packages

It is worth ensuring your Raspberry Pi OS device is kept updated with all the latest software fixes and releases to leverage any new capabilities and to plug any known security exploits. The following

bash script has been provided for your convenience to perform this process. Don't forget to give it a .sh filename extension and run the **$ sudo chmod –x *filename*** command to make the file executable.

```
#!/bin/bash

echo "This script will first update your Raspberry Pi OS OS followed by
all installed applications."
echo "Ensure you are connected to the Internet and really have the time
and want to run this process now."
echo "Also, from time to time, run the >>sudo apt clean<< command to free
up archive space consumed by the downloaded package files"
echo "Press Enter key to continue or Ctrl-c to cancel... "
read -r key
echo "Updating Raspbian..."
sudo apt update
echo "Raspberry Pi OS updated!!!"
echo "Now updating all installed apps..."
sudo apt full-upgrade
echo "---- Raspberry Pi OS OS and all apps have now been updated ----"
echo "Press Enter key to finish... "
read -r key
```

How to use Pulse to manage sound on Linux

https://opensource.com/article/17/1/linux-plays-sound