



# PicoMuon Detector RPI Python code software manual

Pico<sup>μ</sup>

<https://ukraa.com/>

The UK Radio Astronomy Association

A charitable incorporated organisation

Registered charity in England and Wales No. 1123866



## Python code for the UKRAA PicoMuon

Set of Python code to run on an RPi4/5 to get, process and present data from the UKRAA PicoMuon detector

This software was written to suit a specific set-up, feel free to use as you see fit.

## Table of Contents

Python code for the UKRAA PicoMuon .....	2
Requirements .....	4
File structure.....	5
Scripts .....	5
Logfiles.....	5
Data .....	6
Plots .....	7
Temp.....	7
Website.....	8
Using the code .....	9
Set-up instructions.....	9
Raspberry Pi OS.....	10
RPi4 configuration.....	19
Headless access to RPi via VNC .....	25
Add additional software to your RPi .....	28
pandas .....	28
gnuplot.....	29
Get code from github.....	32
Where is my detector? .....	36
Running the code to get data from the detector .....	37
Running the code to process and plot the data from the detector.....	42
Creating simple home intranet web server on your RPi .....	46
Creating simple website for you PicoMuon plots to view your results on your smart phone.....	47
Get daily plots to your website .....	51
How to Set Up a Raspberry Pi Web Server.....	52
License .....	54
Contact us .....	54

---



# Requirements

The software needed to run will depend on what you intend. You will need the following:


- Essential: [Python 3](#)
  - Optional: [Pandas](#) for neutron data
  - Optional: [gnuplot](#) for graphing processed data
-

# File structure

## Scripts

- └─  scripts
  - └─  \_\_pycache\_\_
    - └─ nest.cpython-311.pyc
  - └─ GetDataNeutron.py
  - └─ GetDataRawACM0.py
  - └─ nest.py
  - └─ PlotDataAdcACM0.gp
  - └─ PlotDataCpmACM0.gp
  - └─ PlotDataDayACM0.gp
  - └─ PlotDataFreqACM0.gp
  - └─ PlotDataMonthACM0.gp
  - └─ PlotDataWeekACM0.gp
  - └─ ProcessDataAdcACM0.py
  - └─ ProcessDataCpmACM0.py
  - └─ ProcessDataDayACM0.py
  - └─ ProcessDataMonthACM0.py
  - └─ ProcessDataWeekACM0.py














## Logfiles

- └─  logfiles
  - └─ cron-PicoMuonACM0.log


## Data

```
└─ data
  └─ neutrons
    └─ 2025
      └─ 2025-01
        └─ 2025-01-16.txt
  └─ processed
    └─ adc
      └─ ACM0
        └─ 2025
          └─ 2025-01
            └─ 2025-01-16.txt
  └─ cpm
    └─ ACM0
      └─ 2025
        └─ 2025-01
          └─ 2025-01-16.txt
  └─ day
    └─ ACM0
      └─ 2025
        └─ 2025-01
          └─ 2025-01-16.txt
  └─ month
    └─ ACM0
      └─ 2025
        └─ 2025-01
          └─ 2025-01-16.txt
          └─ temp.txt
  └─ week
    └─ ACM0
      └─ 2025
        └─ 2025-01
          └─ 2025-01-16.txt
          └─ temp.txt
  └─ raw
    └─ ACM0
      └─ 2025
        └─ 2025-01
          └─ 2025-01-16.txt
```




## Plots

- └─  plots
  - └─  adc
    - └─  ACM0
      - └─ 2025-01-16\_ADC\_T\_plot.png
  - └─  cpm
    - └─  ACM0
      - └─ 2025-01-16\_CPM\_T\_plot.png
  - └─  day
    - └─  ACM0
      - └─ 2025-01-16\_day\_plot.png
  - └─  freq
    - └─  ACM0
      - └─ 2025-01-16\_Freq\_T\_plot.png
  - └─  month
    - └─  ACM0
      - └─ 2025-01-16\_month\_plot.png
  - └─  week
    - └─  ACM0
      - └─ 2025-01-16\_week\_plot.png

## Temp

- └─  temp
  - └─ ACM0\_B-Adc.png
  - └─ ACM0\_B-Cpm.png
  - └─ ACM0\_B-Freq.png
  - └─ ACM0\_C-Adc.png
  - └─ ACM0\_C-Cpm.png
  - └─ ACM0\_C-Freq.png
  - └─ ACM0\_day\_plot.png
  - └─ ACM0\_month\_plot.png
  - └─ ACM0\_T-Adc.png
  - └─ ACM0\_T-Cpm.png
  - └─ ACM0\_T-Freq.png
  - └─ ACM0\_week\_plot.png

## Website

- └─  WWW
    - └─  images
      - └─ default.png
      - └─ favicon.ico
      - └─ PicoMuon\_1.png
      - └─ PicoMuon\_2.png
      - └─ PicoMuon\_3.png
      - └─ SunHeader.png
      - └─ UKRAA\_logo\_black.png
      - └─ UKRAA\_logo\_white.png
    - └─  temp
      - └─ ACM0\_B-Adc.png
      - └─ ACM0\_B-Cpm.png
      - └─ ACM0\_B-Freq.png
      - └─ ACM0\_C-Adc.png
      - └─ ACM0\_C-Cpm.png
      - └─ ACM0\_C-Freq.png
      - └─ ACM0\_day\_plot.png
      - └─ ACM0\_T-Adc.png
      - └─ ACM0\_T-Cpm.png
      - └─ ACM0\_T-Freq.png
      - └─ ACM0\_week\_plot.png
  - └─ index.html
-



## Using the code

The code assumes that you the UKRAA PicoMuon detector is connected to the RPi4/5 via supplied USB cable and that it is `/dev/ttyACM0` - you can check this by using `ls /dev/tty*` in a terminal window on the RPi4/5 and reviewing the response.

The code assumes username is `pi`. If `pi` is not the username, then you will need to change all occurrences of `/home/pi` to `/home/username` in the python and gnuplot scripts, where *username* is the username you have selected for your RPi4/5.

The code assumes one detector connected to the RPi4/5 USB and will be `/dev/ttyACM0`, if there are other devices connected to the RPi and your detector is not `/dev/ttyACM0`, then you will need to change `/dev/ttyACM0` to `/dev/ttyACMx` in the `GetDataRawACM0.py` python script, where *ttyACMx* is the tty address of you connected detector.

`GetDataRawACM0.py` is run as a service.

Other scripts (Python and gnuplot) are run from `cron`

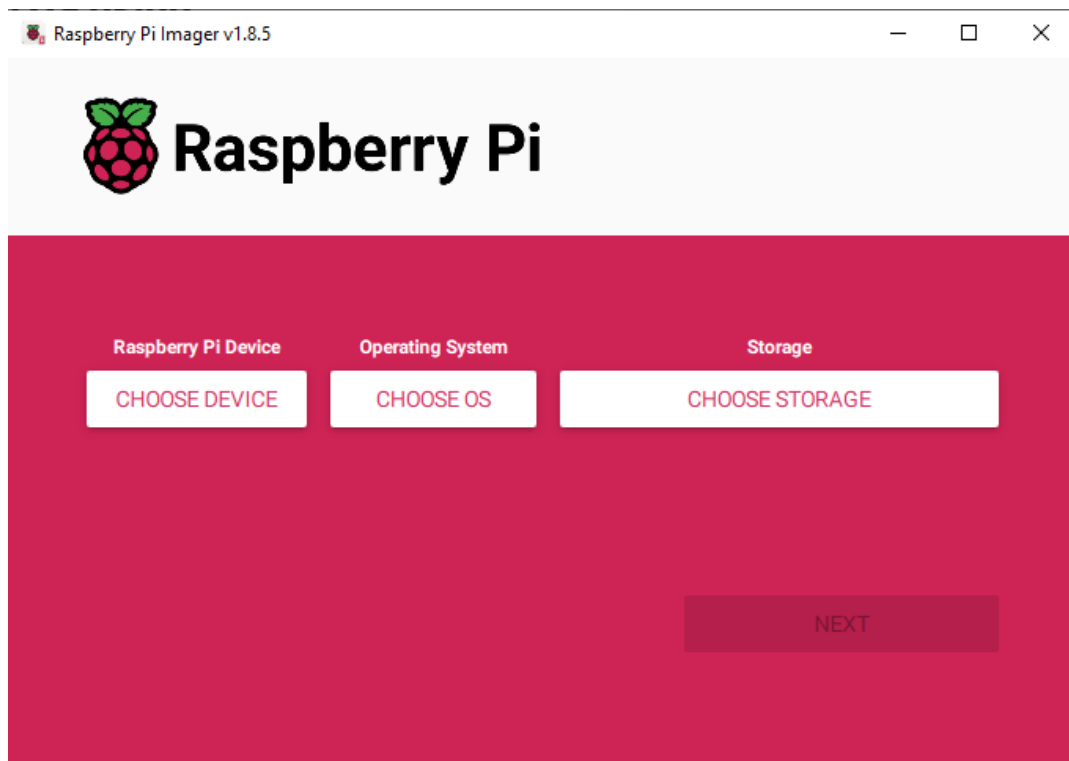
---

## Set-up instructions

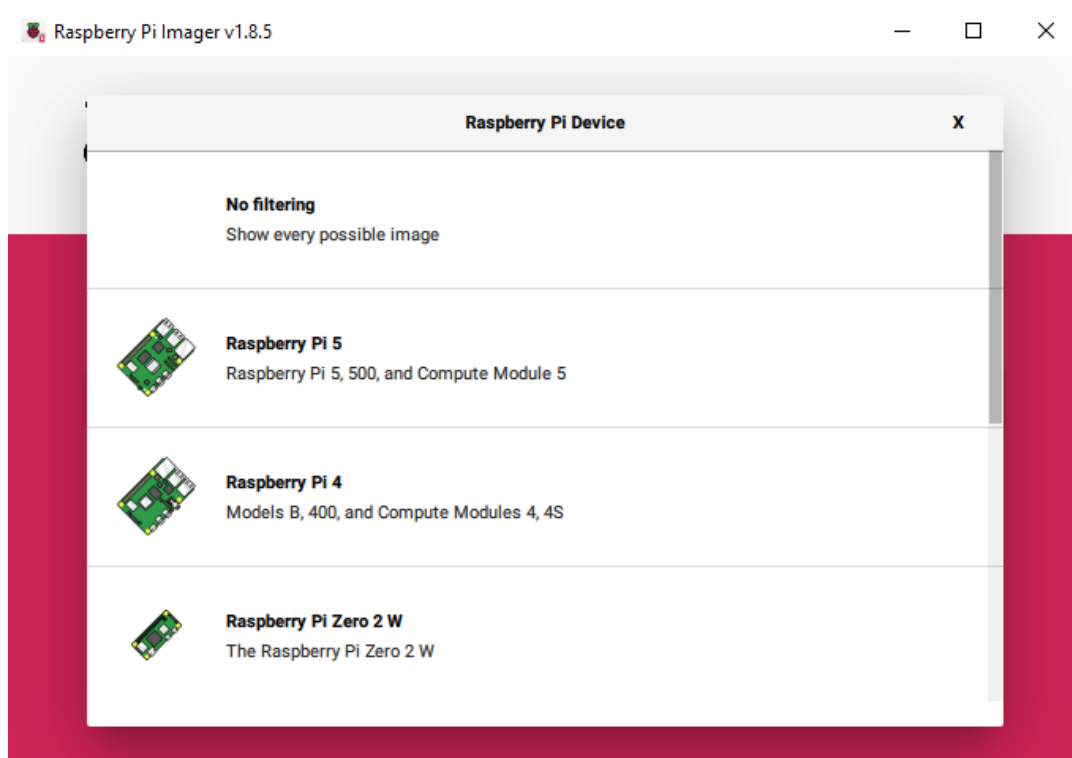
---

# Raspberry Pi OS

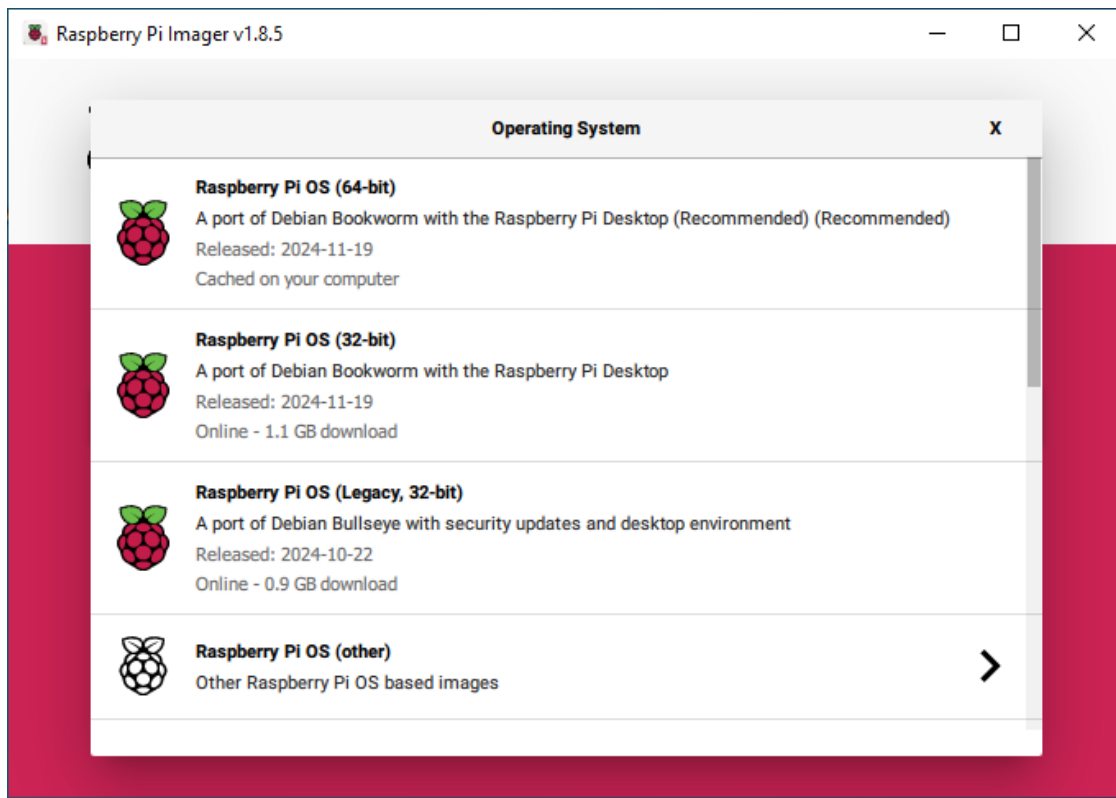
1. Download [Raspberry Pi Imager](#) for your operating system.



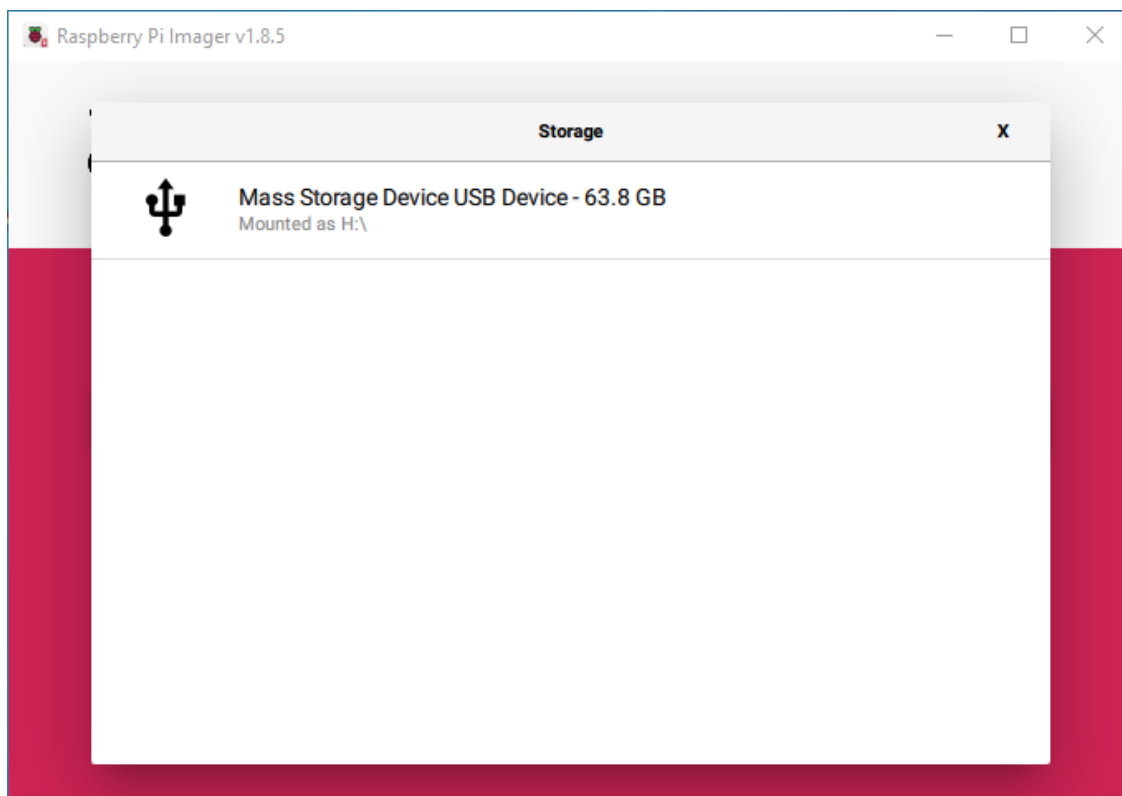
2. Run Raspberry Pi Imager.



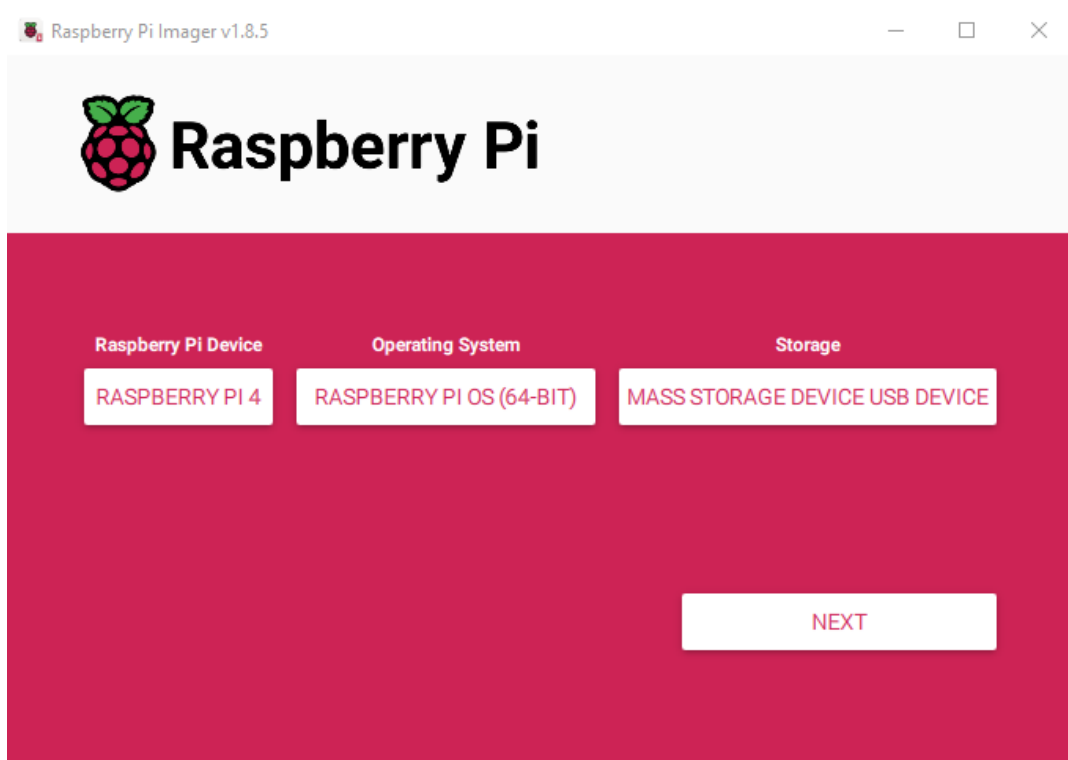
3. Select **CHOOSE DEVICE** and then select your **Raspberry Pi device**.



4. Select **CHOOSE OS** and then select **Raspberry Pi OS (64bit)**.

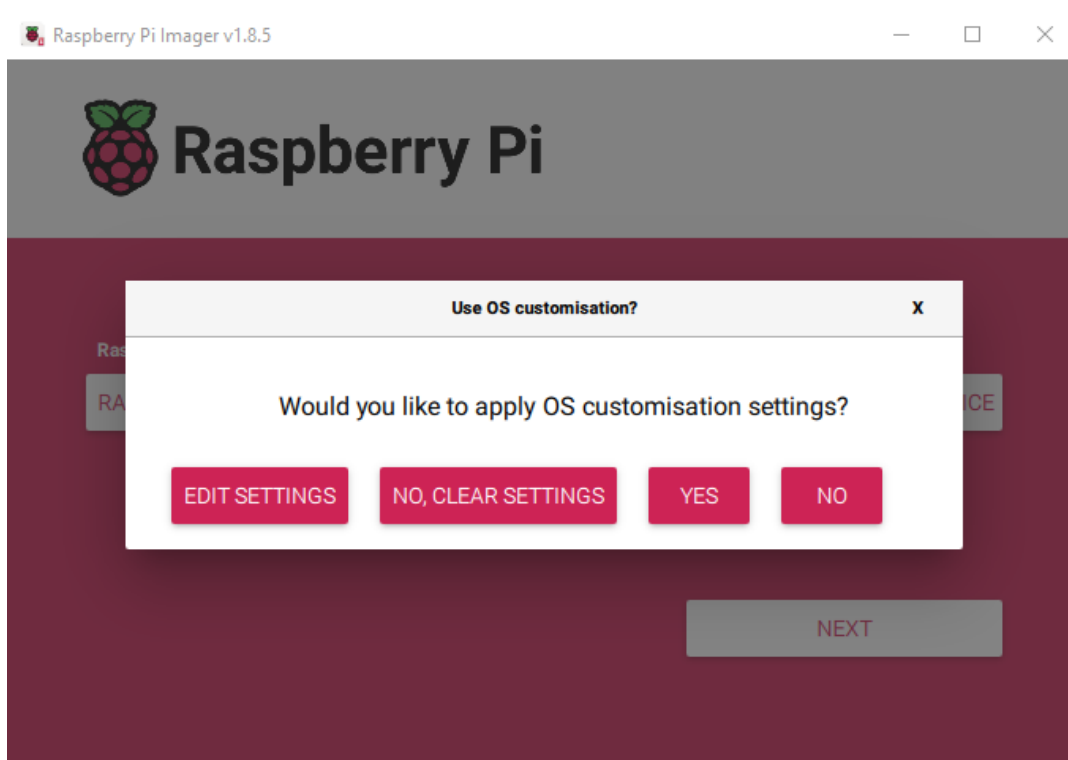


5. Select **CHOOSE STORAGE** and select your **microSD card** for your RPi.



6. Select **NEXT**

7. You are presented with a **Use OS customisation?** window, select **EDIT SETTINGS**.



8. On the **GENERAL** page...

- Set **hostname** - set to whatever you want - but write it down, we will need it latter!
- Set **Username:** and **Password:**
  - Keep username as **pi**
  - set your own password
- Configure wireless LAN
  - if you wish to use you RPi wirelessly
  - Enter your wireless network's **SSID:**, **Password:** and **Wireless LAN country:**
- Set locale settings
  - set **Time zone:** and **Keyboard layout** to your preferences

OS Customisation

GENERAL

SERVICES

OPTIONS

☒ Set hostname: .local

☒ Set username and password

Username:

Password:

☒ Configure wireless LAN

SSID:

Password:

☒ Show password ☐ Hidden SSID

Wireless LAN country:

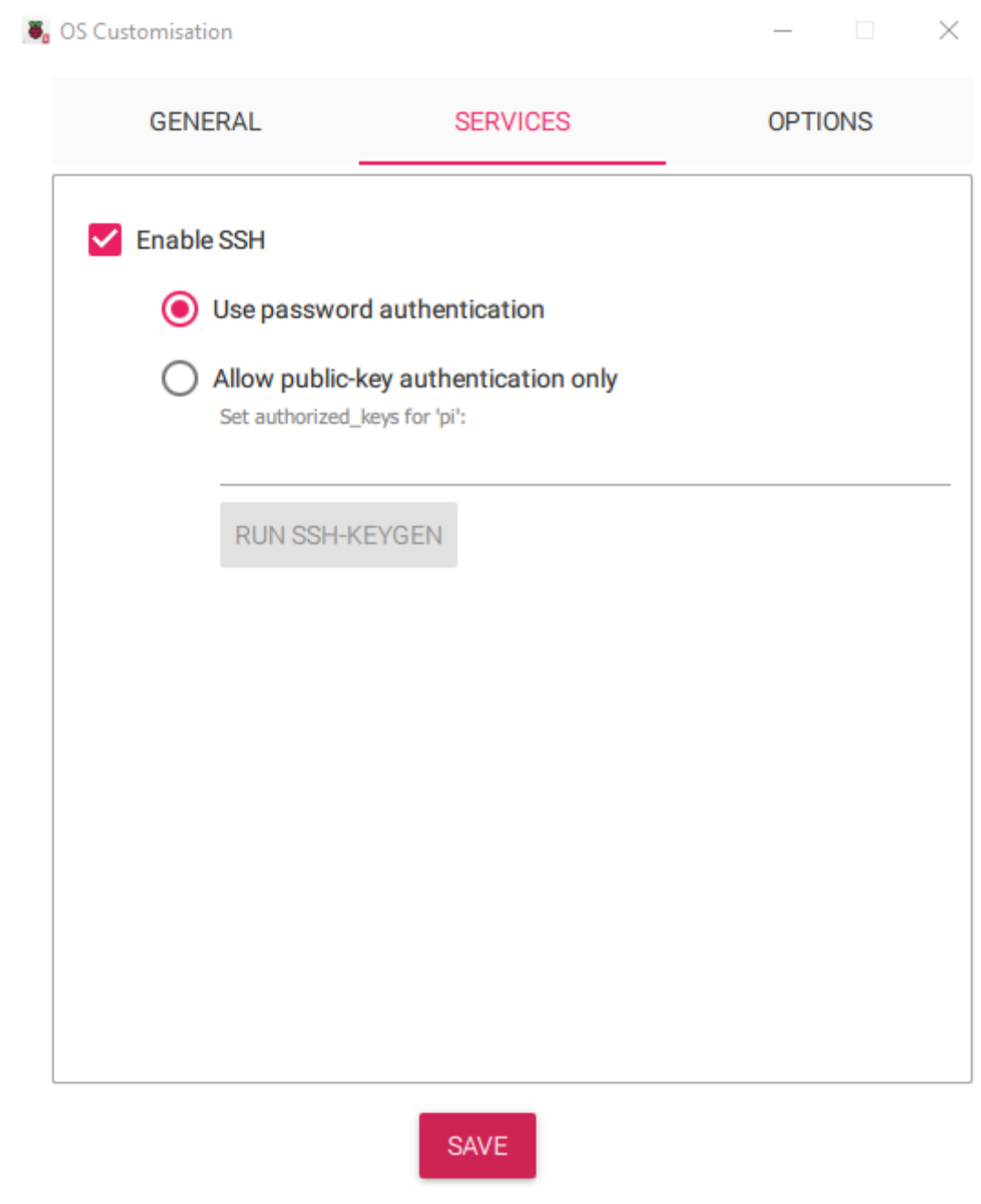
☒ Set locale settings

Time zone:

Keyboard layout:

SAVE

9. Select **SERVICES**
  - Keep as shown



The screenshot shows a window titled "OS Customisation" with three tabs: "GENERAL", "SERVICES", and "OPTIONS". The "SERVICES" tab is selected and highlighted with a red underline. Inside the "SERVICES" tab, there is a section for SSH configuration. It starts with a checked checkbox labeled "Enable SSH". Below this, there are two radio button options: "Use password authentication" (which is selected) and "Allow public-key authentication only". Under the second option, there is a text input field with the placeholder text "Set authorized\_keys for 'pi':". Below the input field is a button labeled "RUN SSH-KEYGEN". At the bottom of the window, there is a large red button labeled "SAVE".

OS Customisation

GENERAL SERVICES OPTIONS

☒ Enable SSH

☒ Use password authentication

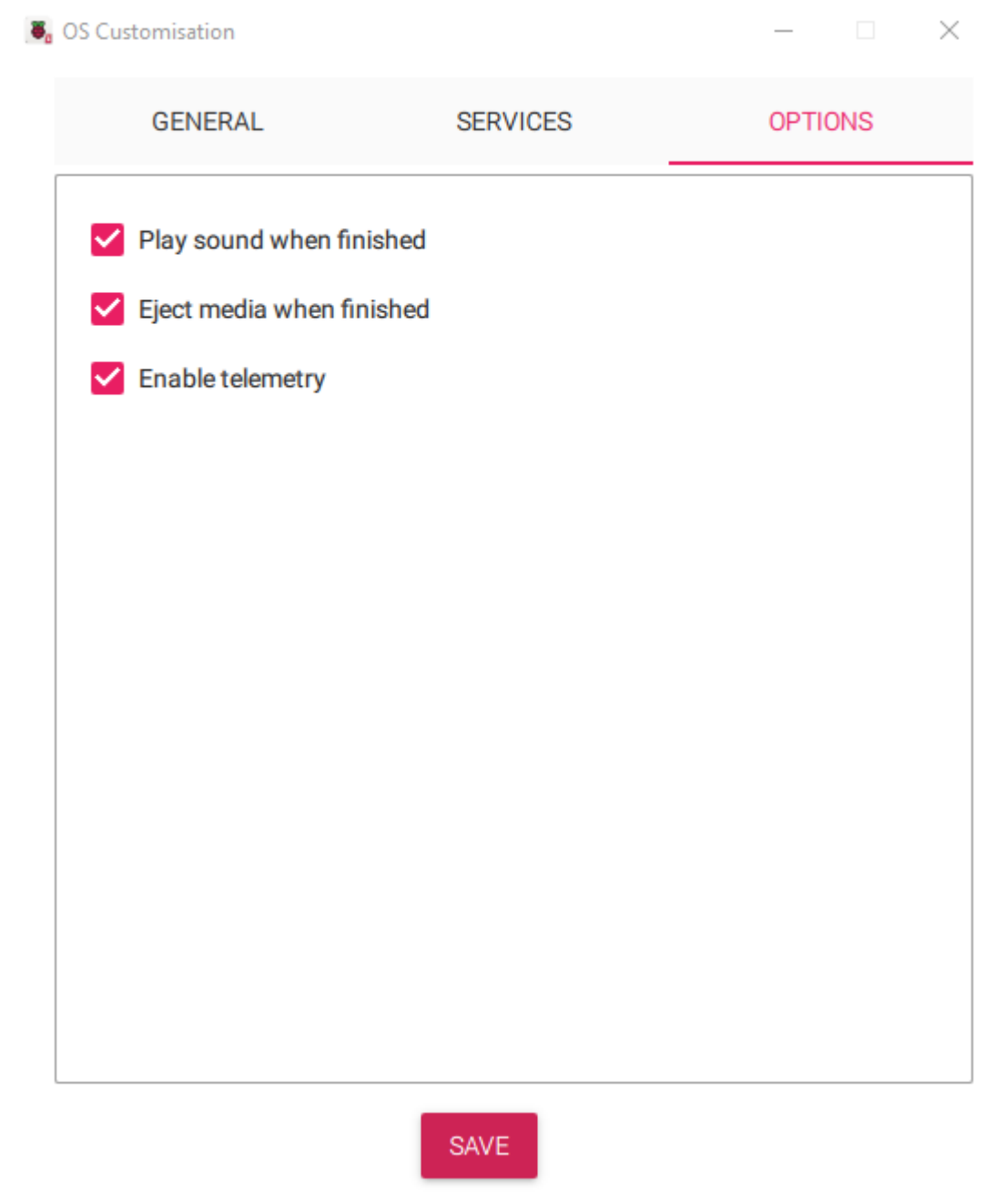
☐ Allow public-key authentication only  
Set authorized\_keys for 'pi':

RUN SSH-KEYGEN

SAVE

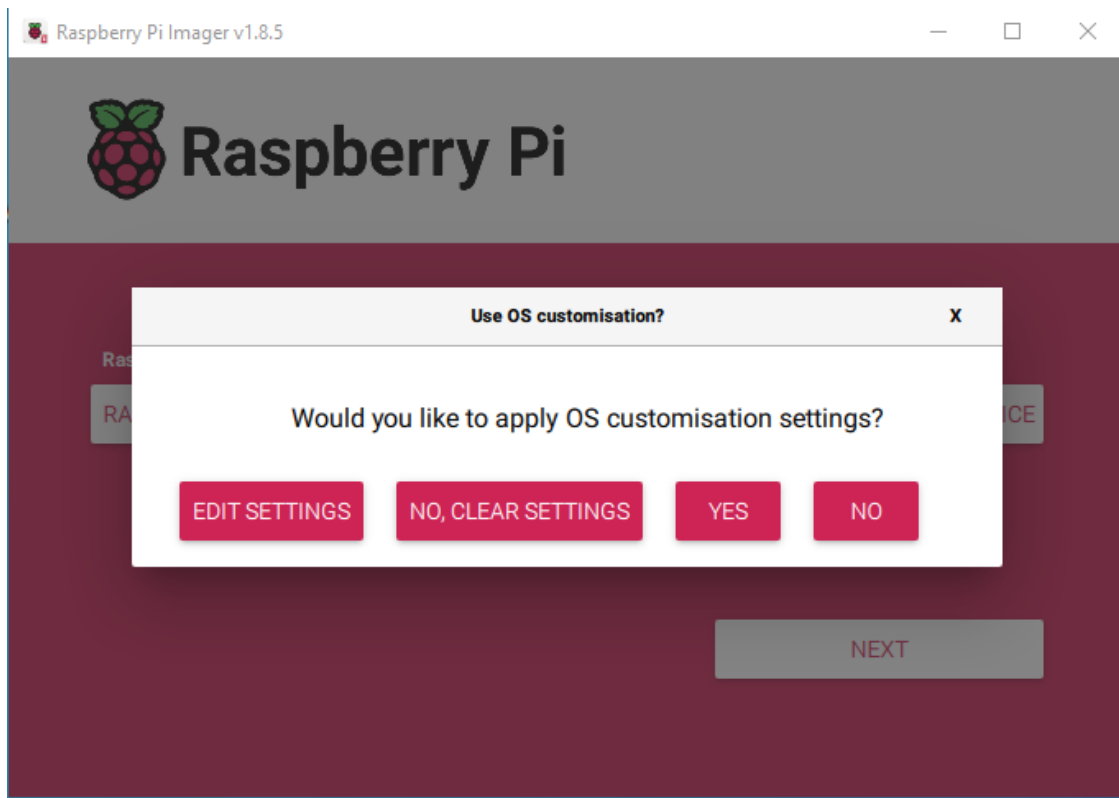
10. Select **OPTIONS**

- Keep as shown

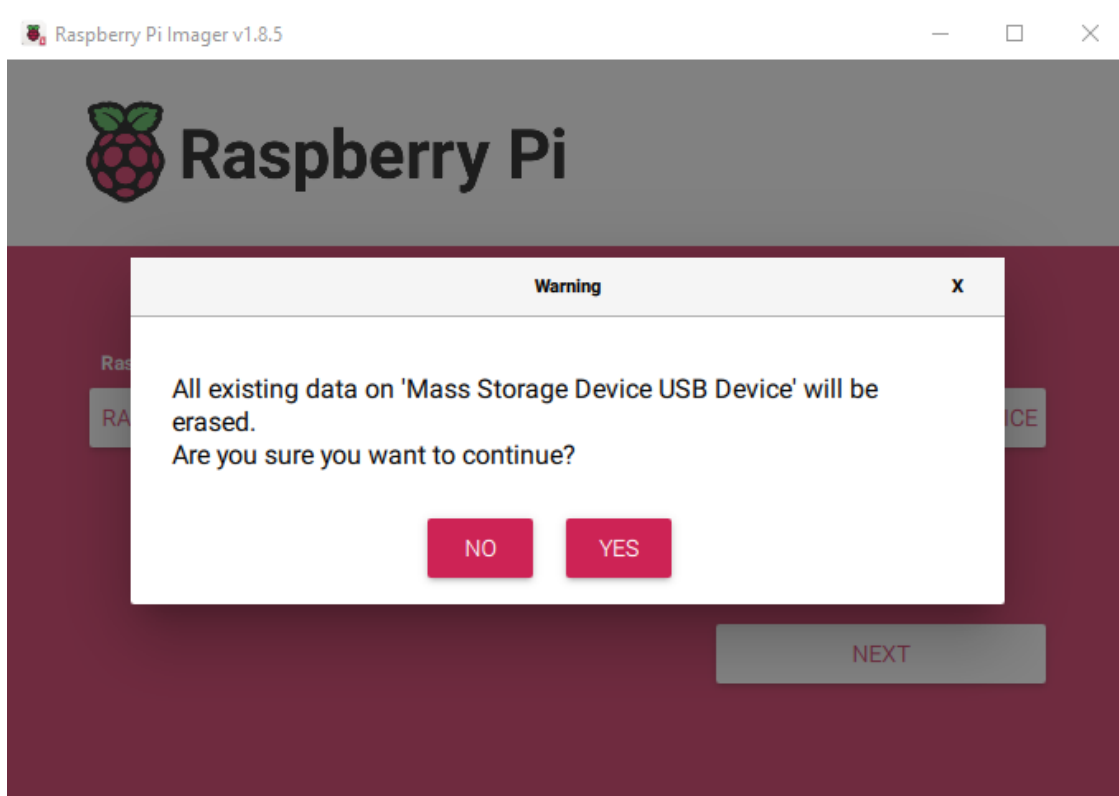


11. Select **SAVE** at bottom of window.

12. Now apply the customised OS settings by selecting **YES**.

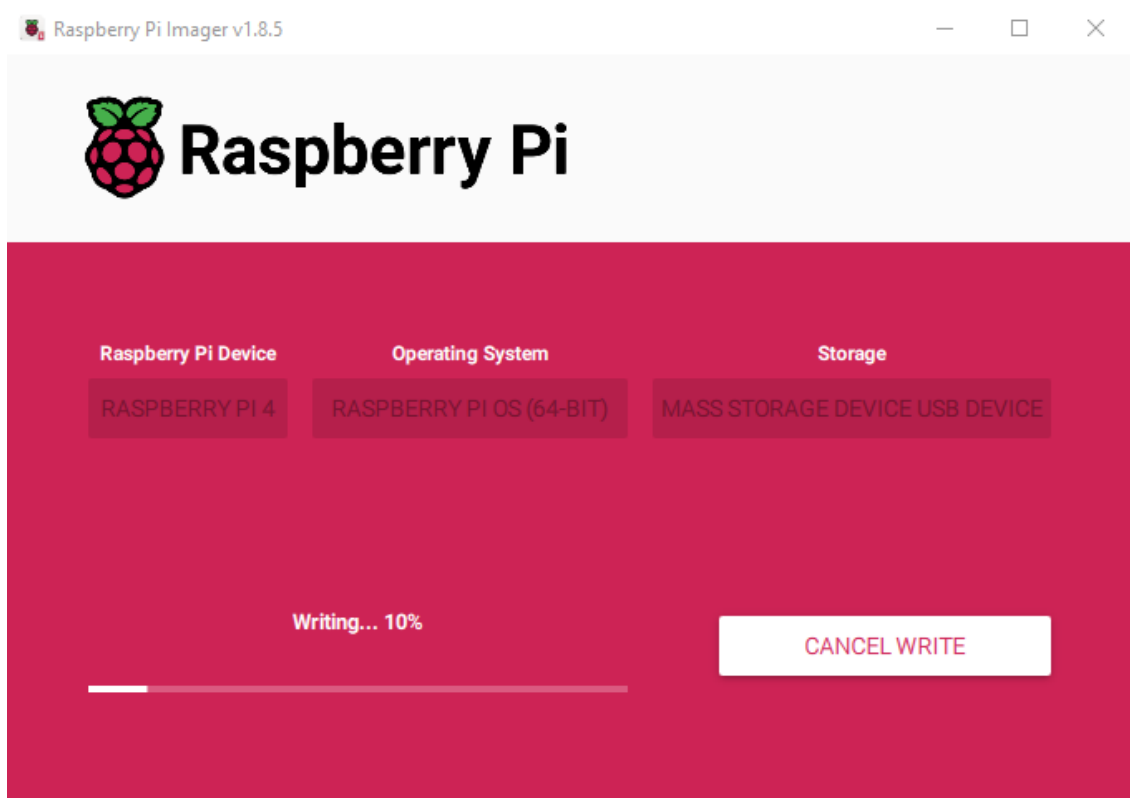


13. You will be asked if you wish to proceed, select **YES**.

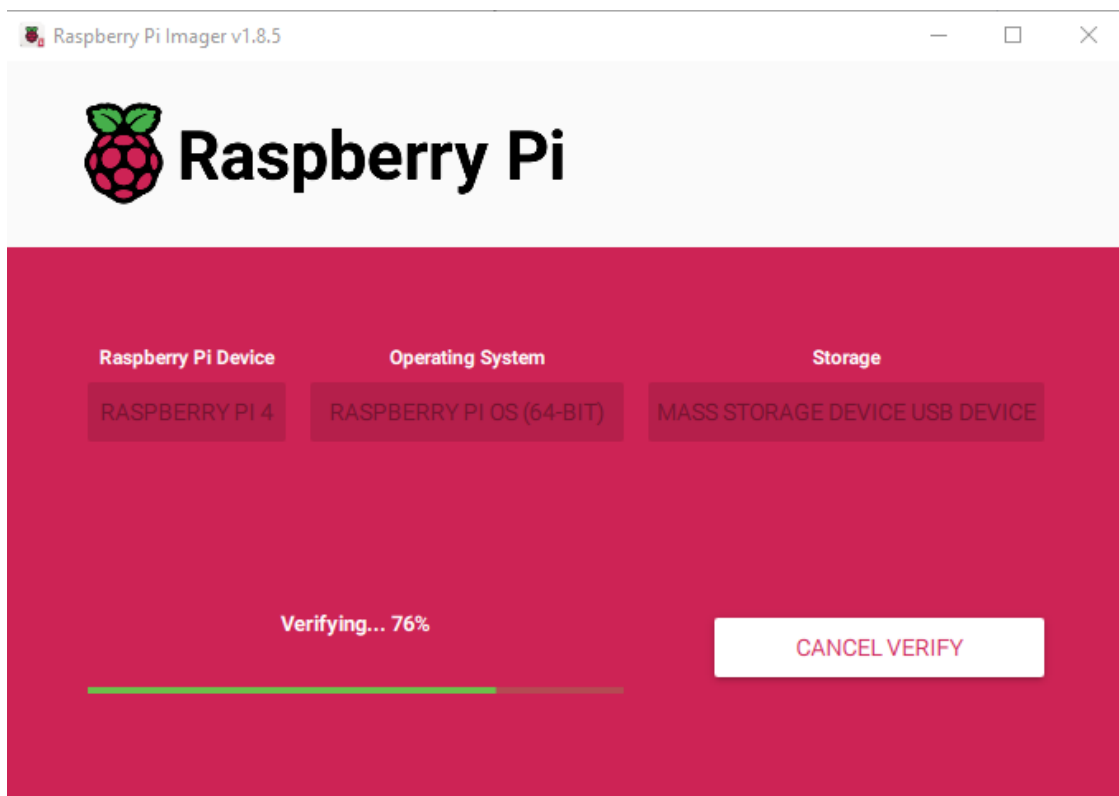




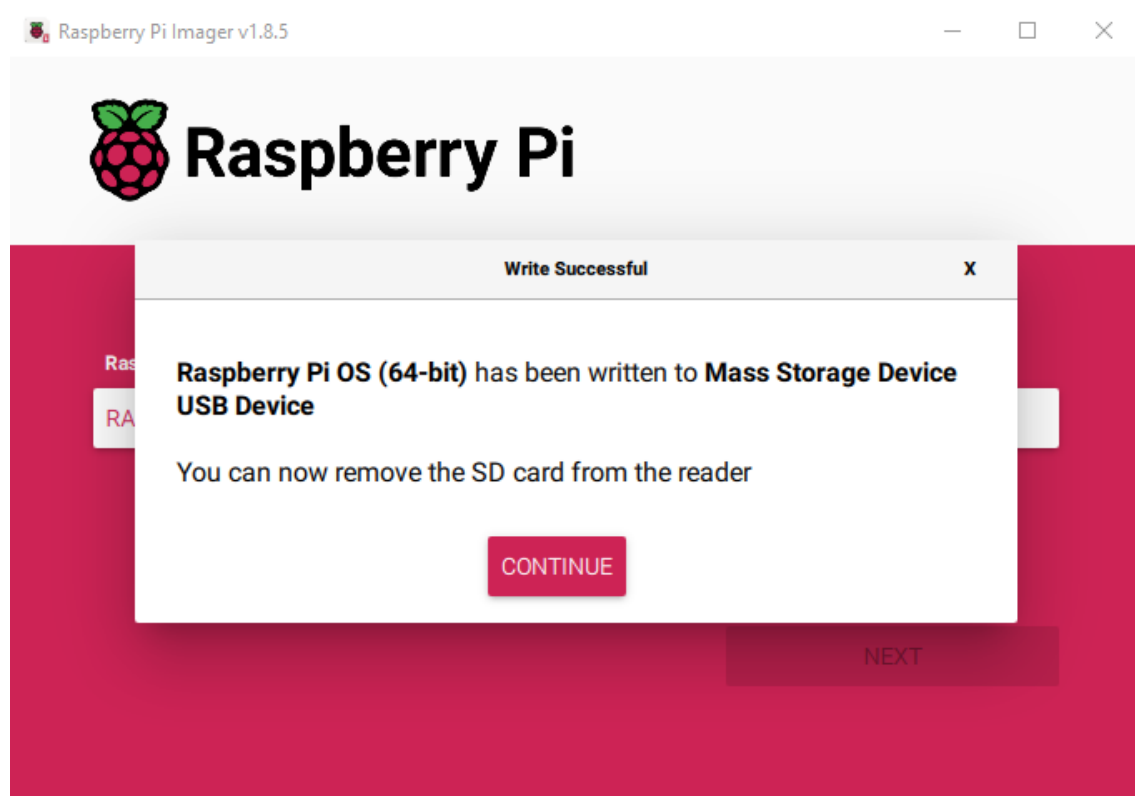
14. The operating system will now be written to the microSD card



15. and then verified, this take a bit of time...



16. When finished, you can remove the microSD card and select **CONTINUE** and close the **Raspberry Pi Imager**.

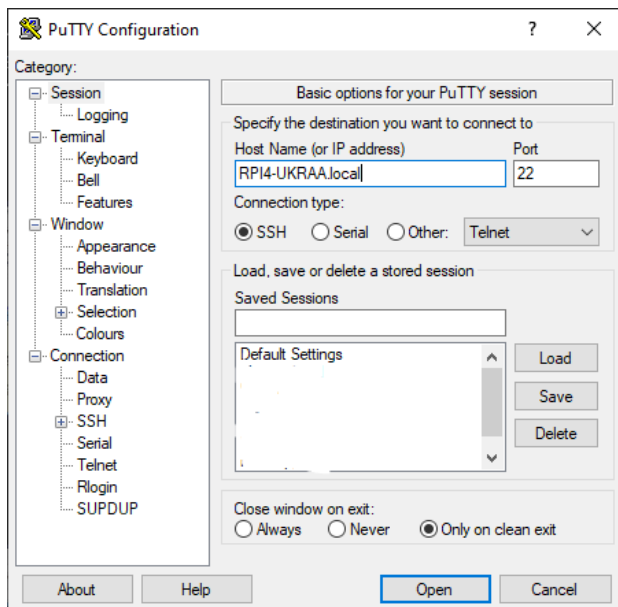


## RPi4 configuration

Put the microSD card you have just created into your RPi4, connect the RPi to your LAN if necessary, insert the power cable and turn on.

Using PuTTY (<https://www.putty.org/>)

1. In **Host Name (or IP address)**, type the host name of your RPi that you wrote down earlier... and select **Open**

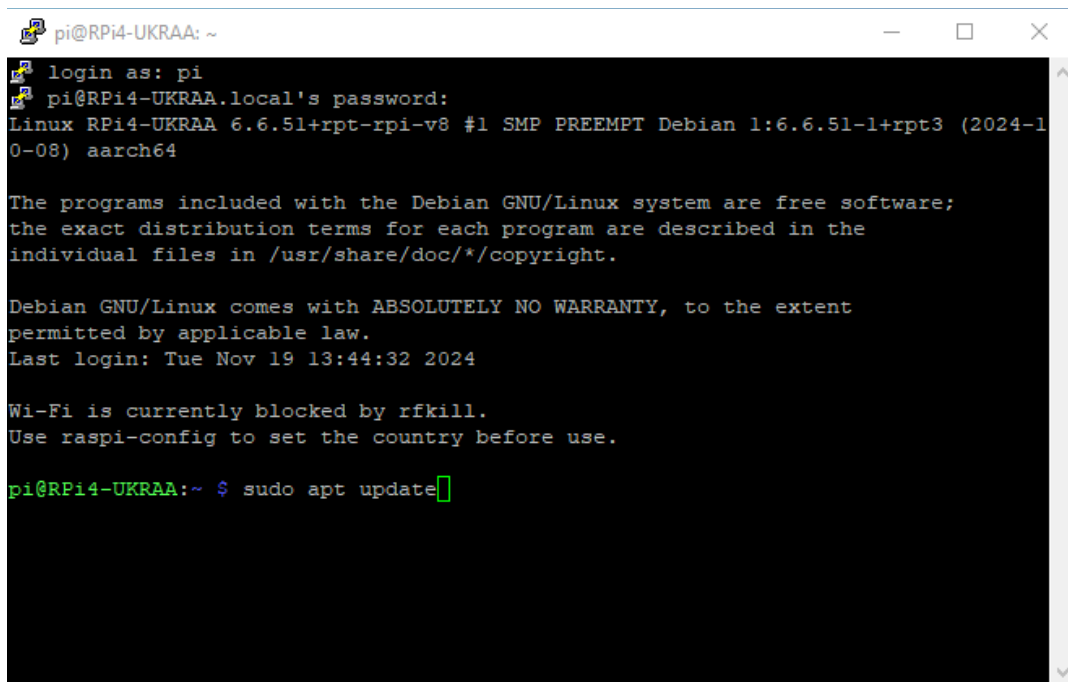


2. You will be presented with a PuTTY Security Alert - select **Accept**. You will now have a PuTTY terminal window.



3. Login as **pi** and enter your **password**
4. Now type the command below and press enter. This will update the RPi OS

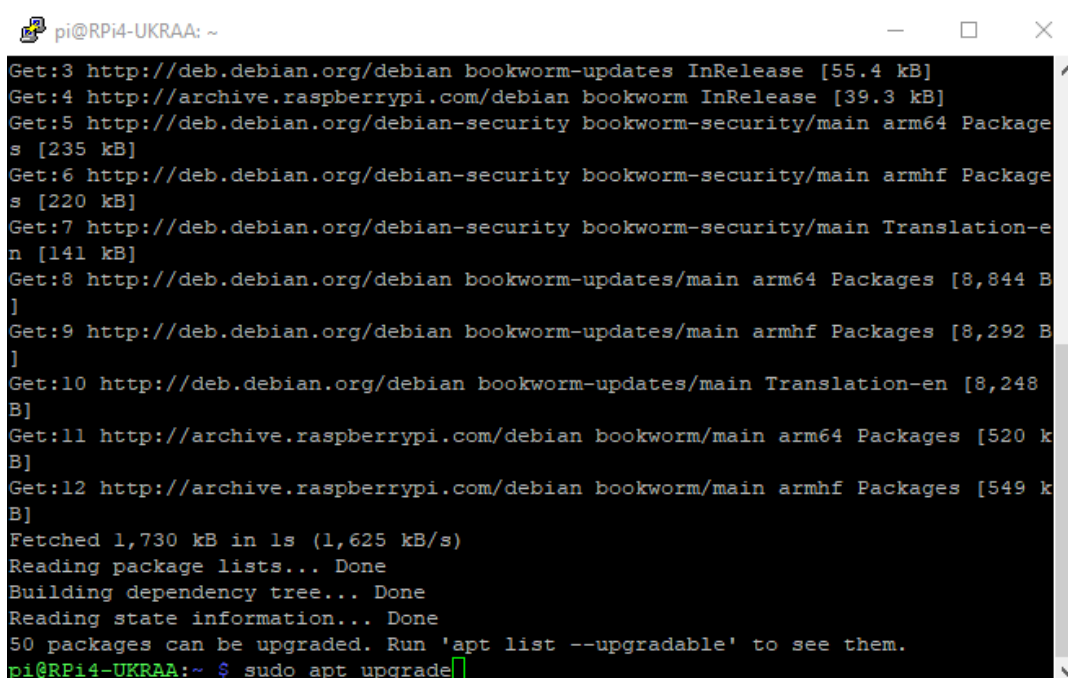
## sudo apt update



```
pi@RPi4-UKRAA: ~  
login as: pi  
pi@RPi4-UKRAA.local's password:  
Linux RPi4-UKRAA 6.6.51+rpt-rpi-v8 #1 SMP PREEMPT Debian 1:6.6.51-1+rpt3 (2024-10-08) aarch64  
  
The programs included with the Debian GNU/Linux system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.  
Last login: Tue Nov 19 13:44:32 2024  
  
Wi-Fi is currently blocked by rfkill.  
Use raspi-config to set the country before use.  
pi@RPi4-UKRAA:~ $ sudo apt update
```

5. Now type the command below and press enter. This will update any preinstalled software packages. You will be asked to type **y** to proceed.

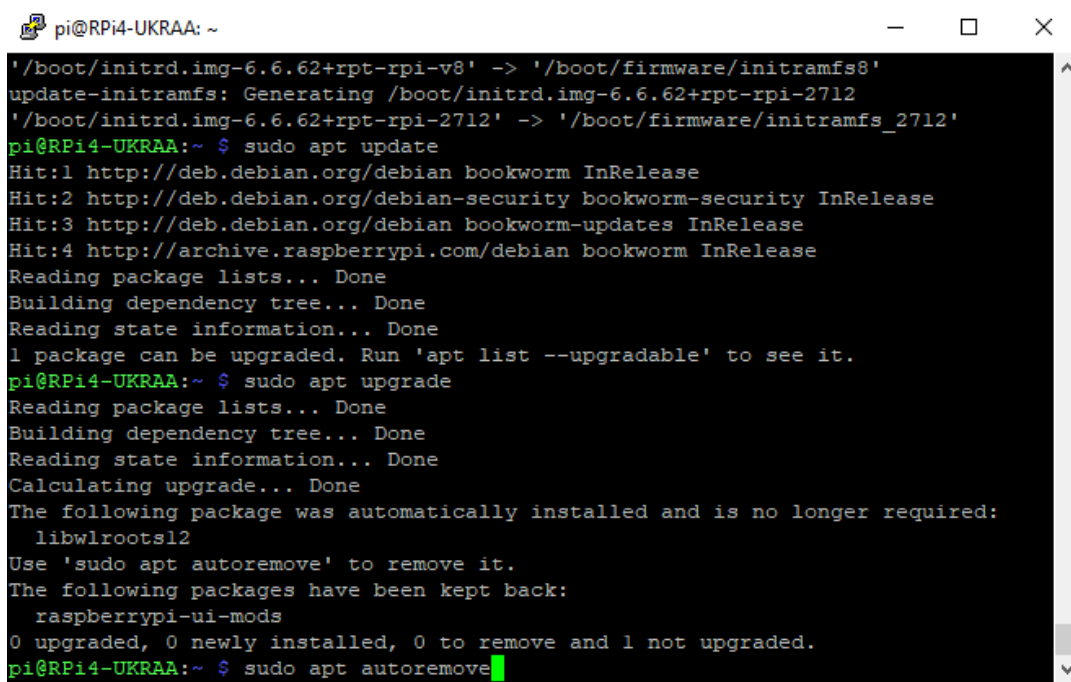
## sudo apt upgrade



```
pi@RPi4-UKRAA: ~  
Get:3 http://deb.debian.org/debian bookworm-updates InRelease [55.4 kB]  
Get:4 http://archive.raspberrypi.com/debian bookworm InRelease [39.3 kB]  
Get:5 http://deb.debian.org/debian-security bookworm-security/main arm64 Packages [235 kB]  
Get:6 http://deb.debian.org/debian-security bookworm-security/main armhf Packages [220 kB]  
Get:7 http://deb.debian.org/debian-security bookworm-security/main Translation-en [141 kB]  
Get:8 http://deb.debian.org/debian bookworm-updates/main arm64 Packages [8,844 B]  
Get:9 http://deb.debian.org/debian bookworm-updates/main armhf Packages [8,292 B]  
Get:10 http://deb.debian.org/debian bookworm-updates/main Translation-en [8,248 B]  
Get:11 http://archive.raspberrypi.com/debian bookworm/main arm64 Packages [520 kB]  
Get:12 http://archive.raspberrypi.com/debian bookworm/main armhf Packages [549 kB]  
Fetched 1,730 kB in 1s (1,625 kB/s)  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
50 packages can be upgraded. Run 'apt list --upgradable' to see them.  
pi@RPi4-UKRAA:~ $ sudo apt upgrade
```

6. You may be advised that you can remove **no longer required** packages. You can type the command below and press enter. Again, you will be asked to type **y** to proceed.

## sudo apt autoremove

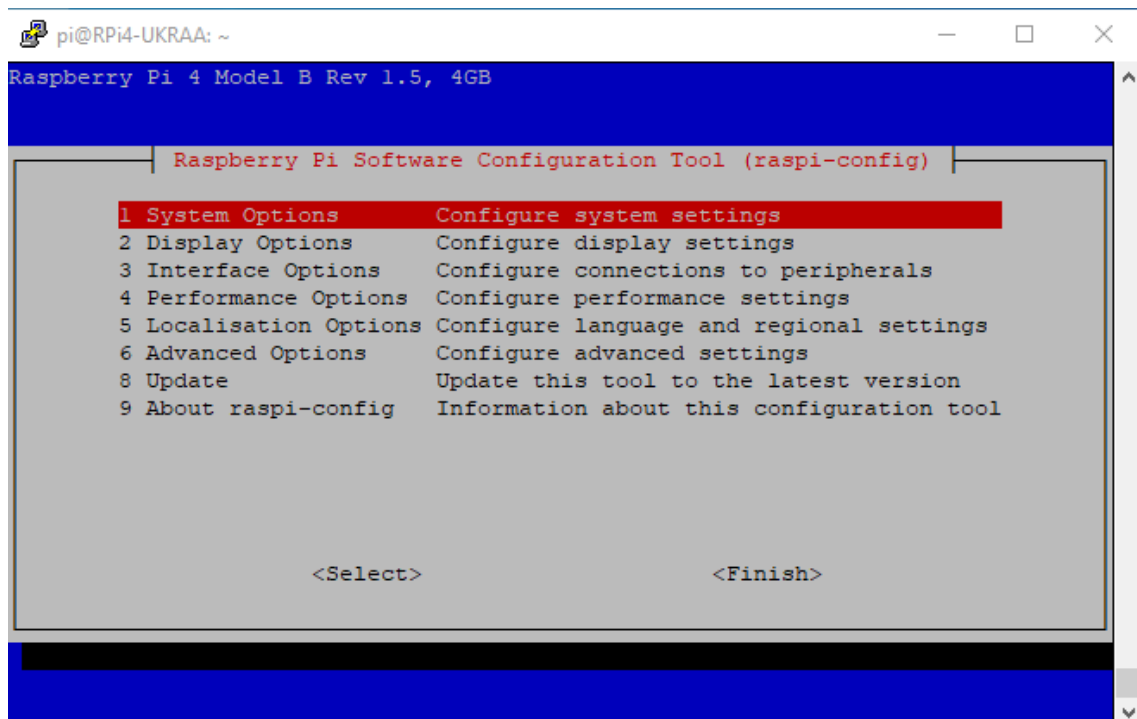


```
pi@RPi4-UKRAA: ~  
'/boot/initrd.img-6.6.62+rpt-rpi-v8' -> '/boot/firmware/initramfs8'  
update-initramfs: Generating /boot/initrd.img-6.6.62+rpt-rpi-2712  
'/boot/initrd.img-6.6.62+rpt-rpi-2712' -> '/boot/firmware/initramfs_2712'  
pi@RPi4-UKRAA:~$ sudo apt update  
Hit:1 http://deb.debian.org/debian bookworm InRelease  
Hit:2 http://deb.debian.org/debian-security bookworm-security InRelease  
Hit:3 http://deb.debian.org/debian bookworm-updates InRelease  
Hit:4 http://archive.raspberrypi.com/debian bookworm InRelease  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
1 package can be upgraded. Run 'apt list --upgradable' to see it.  
pi@RPi4-UKRAA:~$ sudo apt upgrade  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
Calculating upgrade... Done  
The following package was automatically installed and is no longer required:  
  libwlroots12  
Use 'sudo apt autoremove' to remove it.  
The following packages have been kept back:  
  raspberrypi-ui-mods  
0 upgraded, 0 newly installed, 0 to remove and 1 not upgraded.  
pi@RPi4-UKRAA:~$ sudo apt autoremove
```

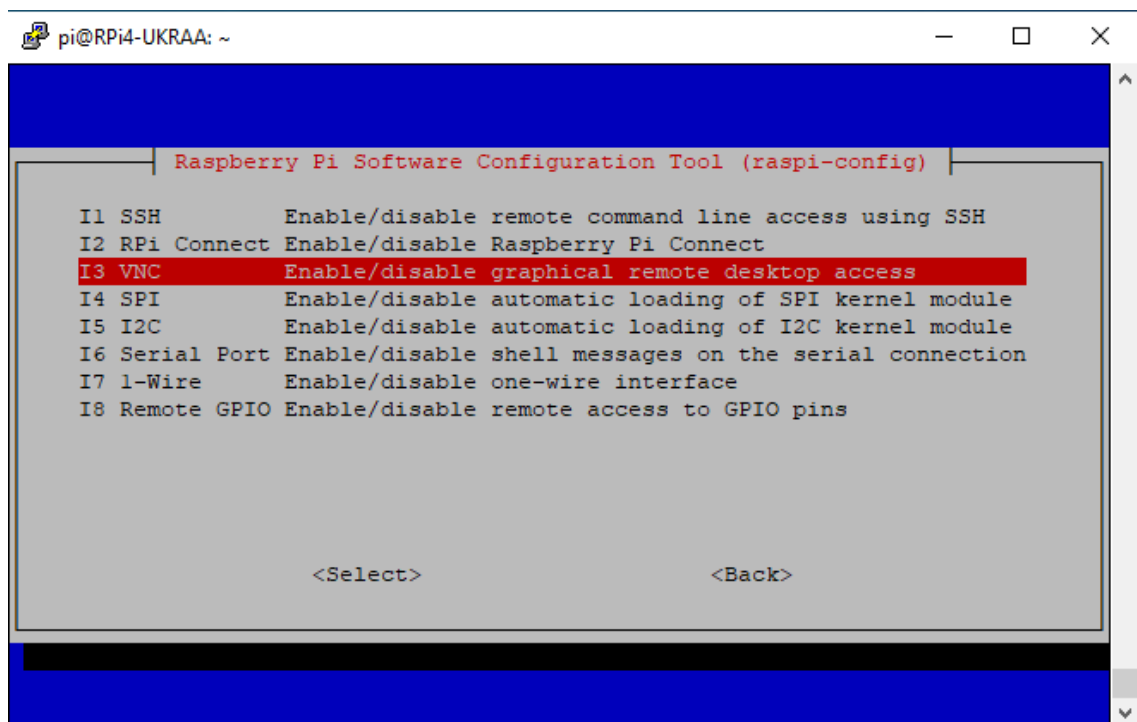
7. We can now set up VNC to access the RPi from our desktop PC. Type the command below and press enter. We will be presented with the RPi configuration tool.

## sudo raspi-config

8. Select **3 Interface Options** and selecting **Select** and press enter.



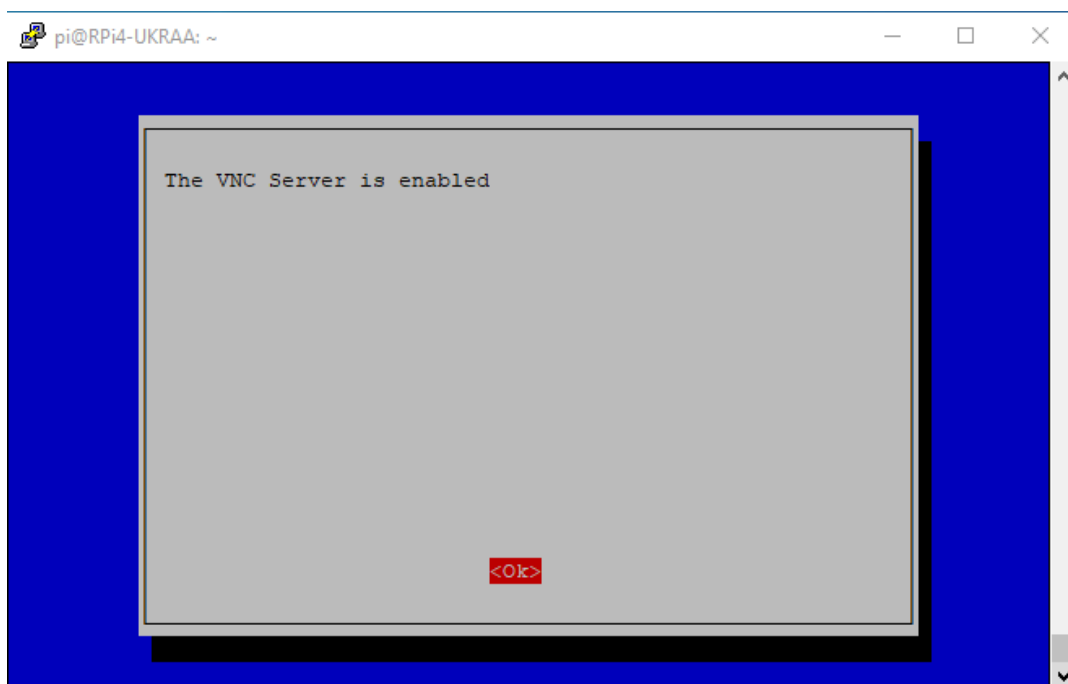
9. Now navigate down to **I3 VNC** and selecting **Select** and press enter.



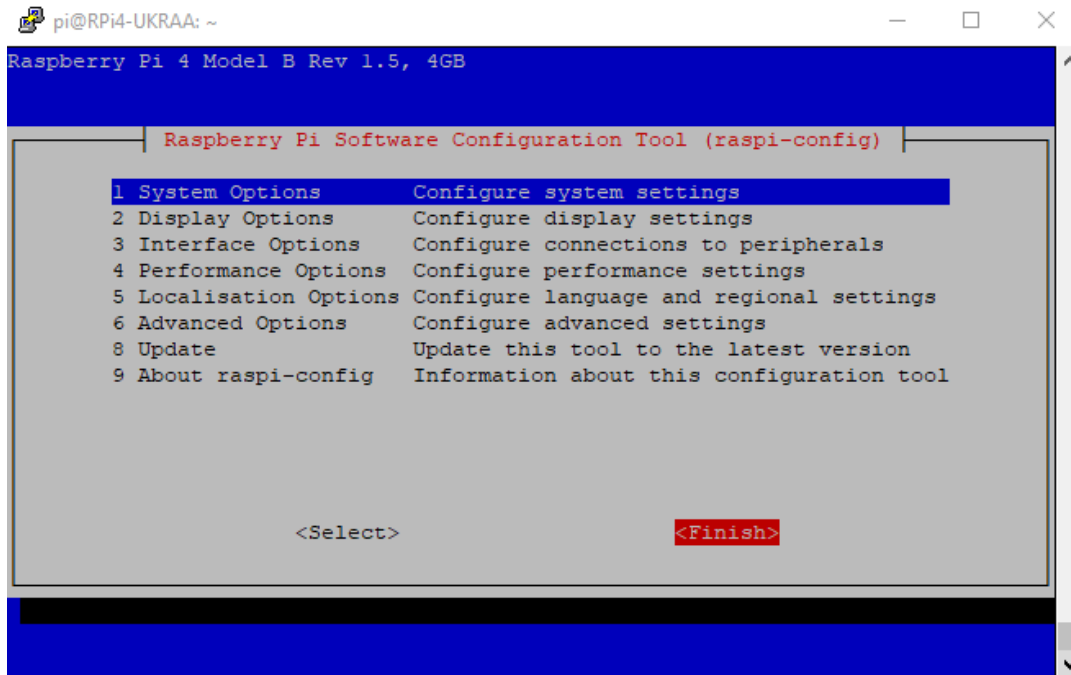
10. Select **Yes** and press enter to enable the VNC Server.



11. VNC server will now be enabled, press enter.



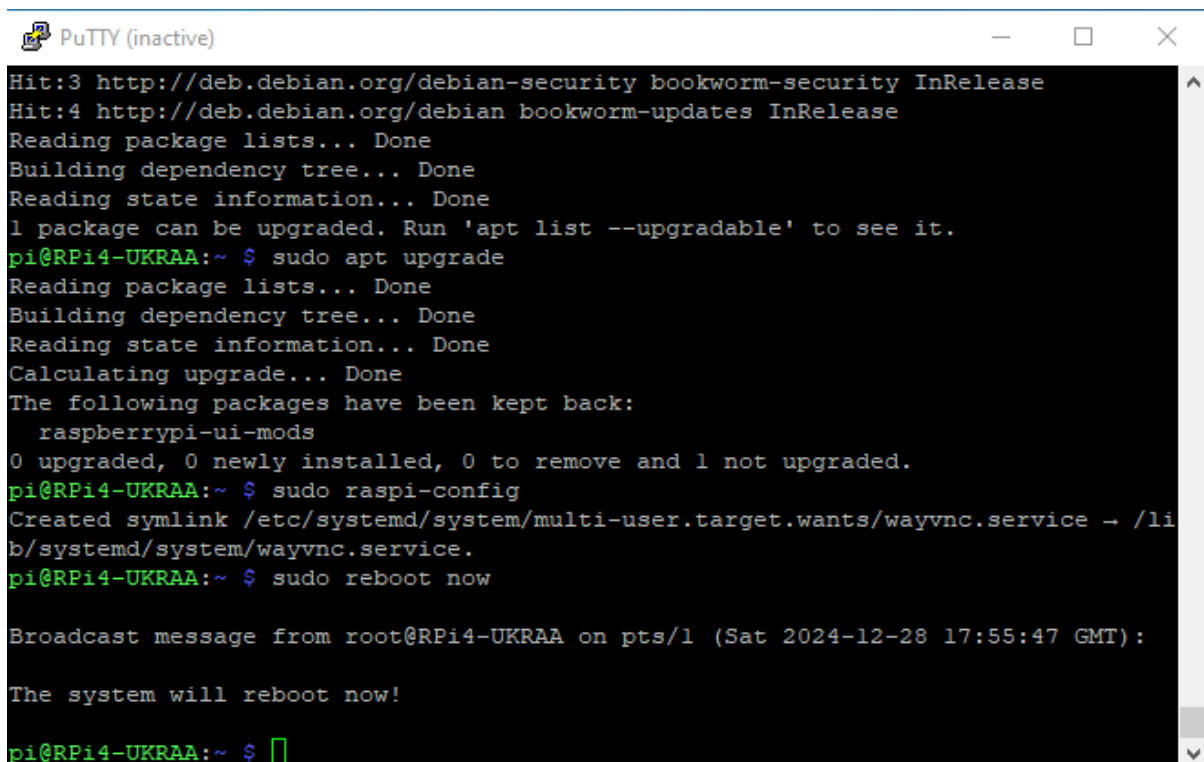
12. Finish with the Configuration Tool by selecting **Finish** and press enter.



The screenshot shows the Raspberry Pi Software Configuration Tool (raspi-config) interface. The title bar indicates the user is 'pi@RPi4-UKRAA: ~'. The main window has a blue header with 'Raspberry Pi 4 Model B Rev 1.5, 4GB'. Below this, a red title bar reads 'Raspberry Pi Software Configuration Tool (raspi-config)'. The main area is a list of options: 1 System Options (Configure system settings), 2 Display Options (Configure display settings), 3 Interface Options (Configure connections to peripherals), 4 Performance Options (Configure performance settings), 5 Localisation Options (Configure language and regional settings), 6 Advanced Options (Configure advanced settings), 8 Update (Update this tool to the latest version), and 9 About raspi-config (Information about this configuration tool). The first option is highlighted. At the bottom, there are two red buttons: '<Select>' and '<Finish>'.

13. Reboot the RPi, type the command below and press enter. This will close the PuTTY connection. You can close the Putty window.

## sudo reboot



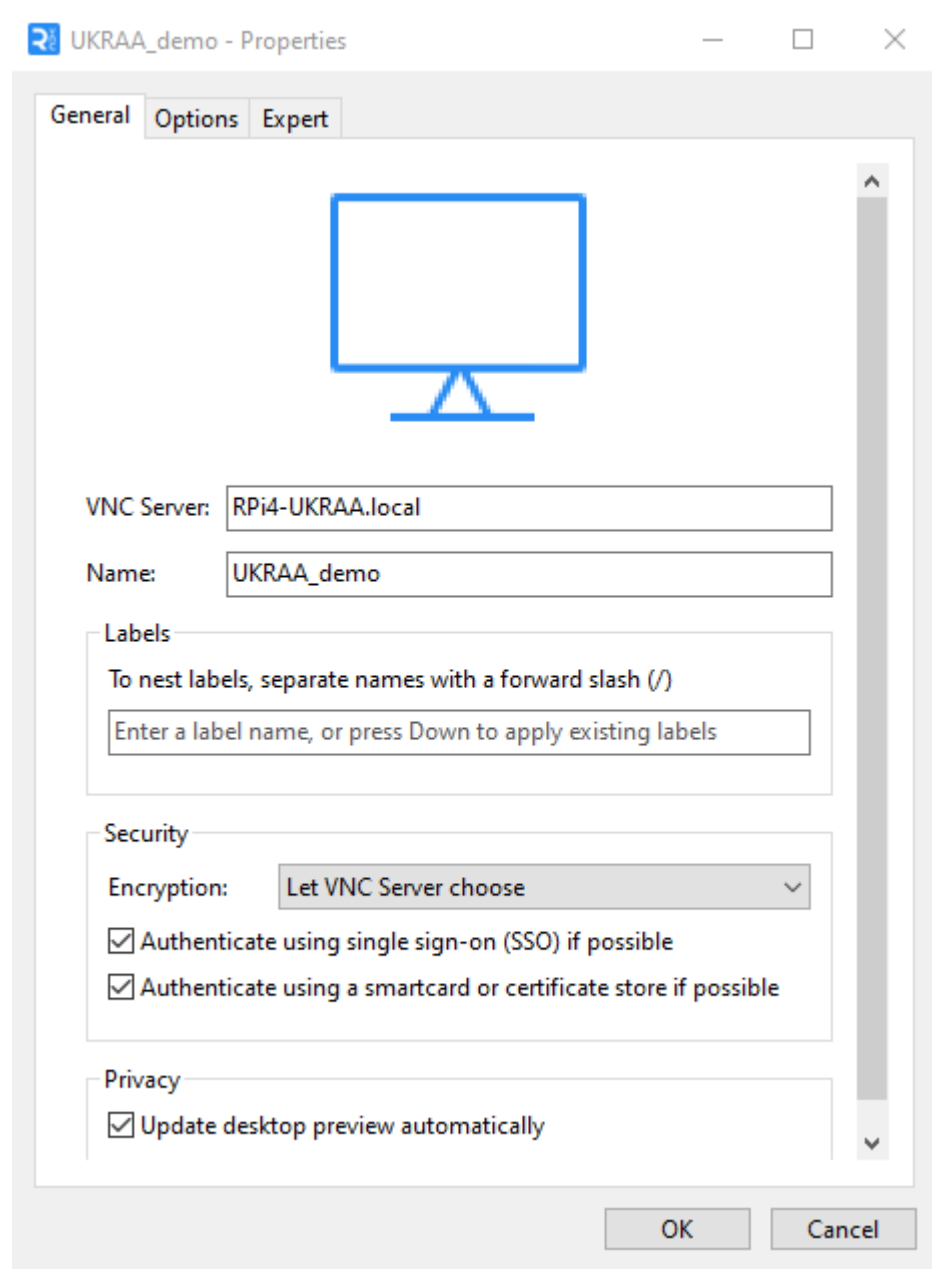
The screenshot shows a PuTTY terminal window titled 'PuTTY (inactive)'. The terminal output shows the following sequence of commands and their results: 'Hit:3 http://deb.debian.org/debian-security bookworm-security InRelease', 'Hit:4 http://deb.debian.org/debian bookworm-updates InRelease', 'Reading package lists... Done', 'Building dependency tree... Done', 'Reading state information... Done', '1 package can be upgraded. Run 'apt list --upgradable' to see it.', 'pi@RPi4-UKRAA:~ \$ sudo apt upgrade', 'Reading package lists... Done', 'Building dependency tree... Done', 'Reading state information... Done', 'Calculating upgrade... Done', 'The following packages have been kept back: raspberrypi-ui-mods', '0 upgraded, 0 newly installed, 0 to remove and 1 not upgraded.', 'pi@RPi4-UKRAA:~ \$ sudo raspi-config', 'Created symlink /etc/systemd/system/multi-user.target.wants/wayvnc.service -> /lib/systemd/system/wayvnc.service.', 'pi@RPi4-UKRAA:~ \$ sudo reboot now', 'Broadcast message from root@RPi4-UKRAA on pts/1 (Sat 2024-12-28 17:55:47 GMT):', 'The system will reboot now!', and finally 'pi@RPi4-UKRAA:~ \$' with a cursor.



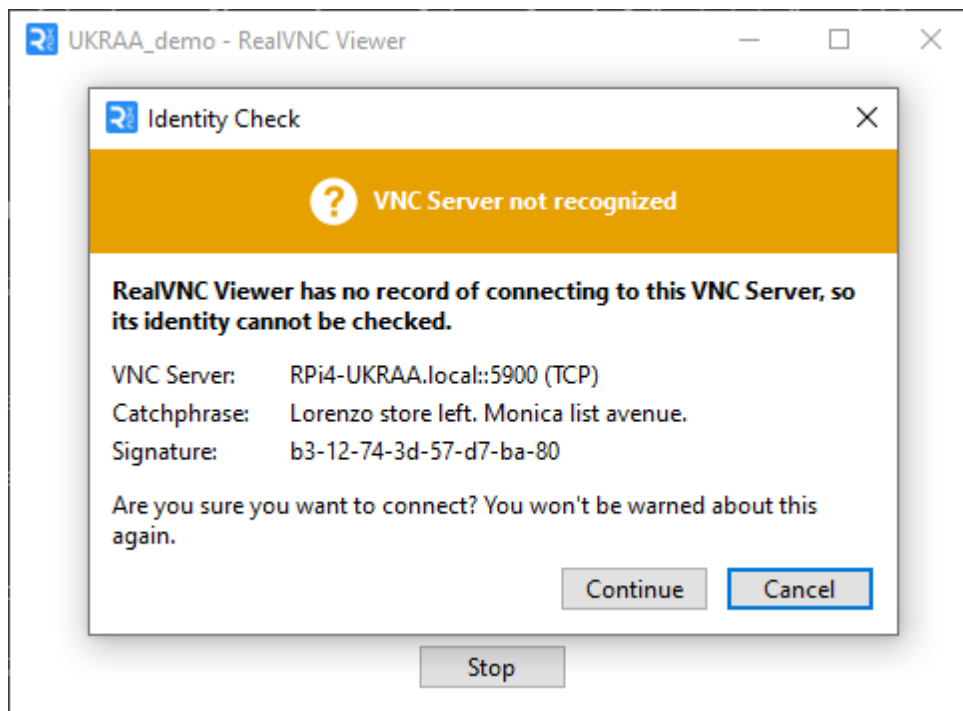
## Headless access to RPi via VNC

Using a VNC program like RealVNC (<https://www.realvnc.com/en/>), connect to your RPi using your desktop PC...

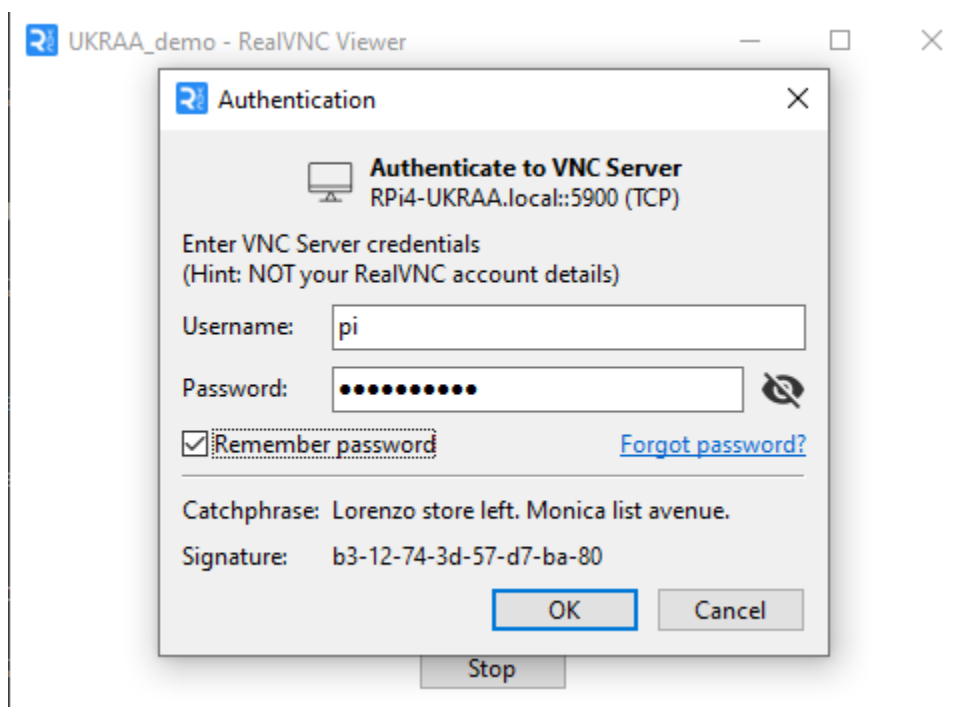
1. Create a **New connections** (CTRL-N) and enter the hostname into the **VNC Server:** box, add friendly name in the **Name** box and select **OK**.



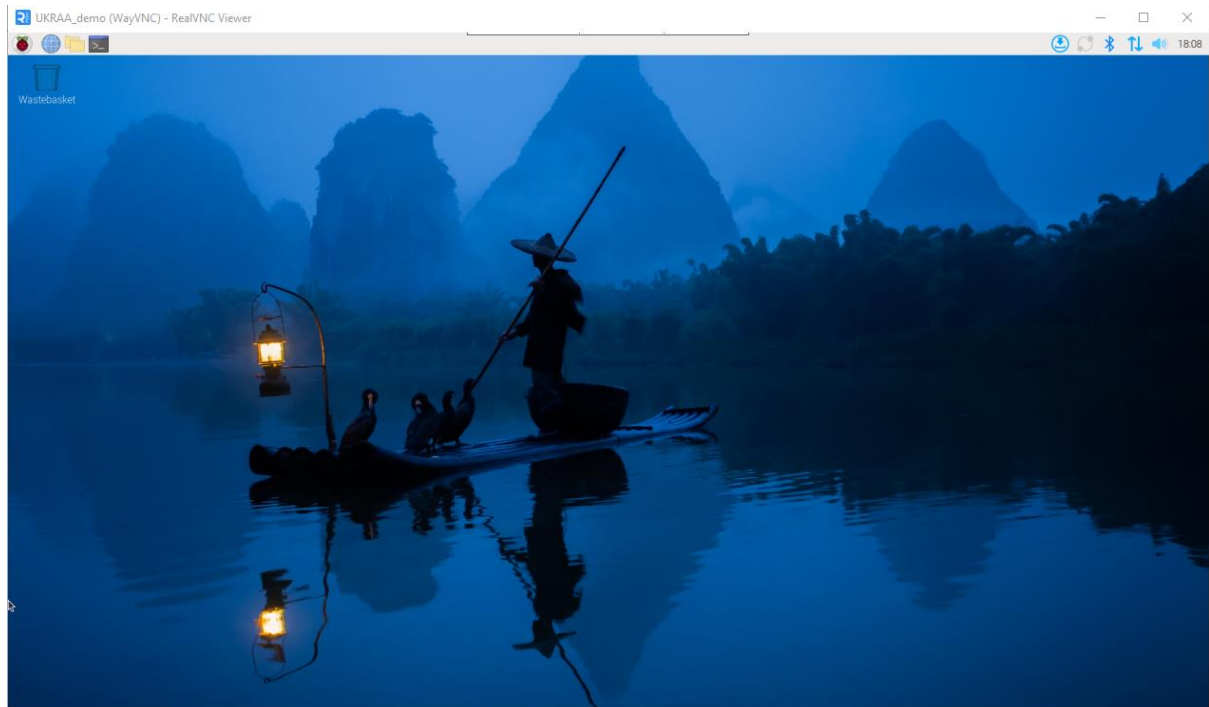
2. Open this VNC connection to your RPi; you will get a **VNC server not recognised** window from RealVNC. Select **Continue**



3. You will now get an **Authentication** window from RealVNC, enter your **Username** and **Password** details and select **Remember password** to make it easier to login in future.



4. You are now remotely accessing your RPi from your desktop PC...



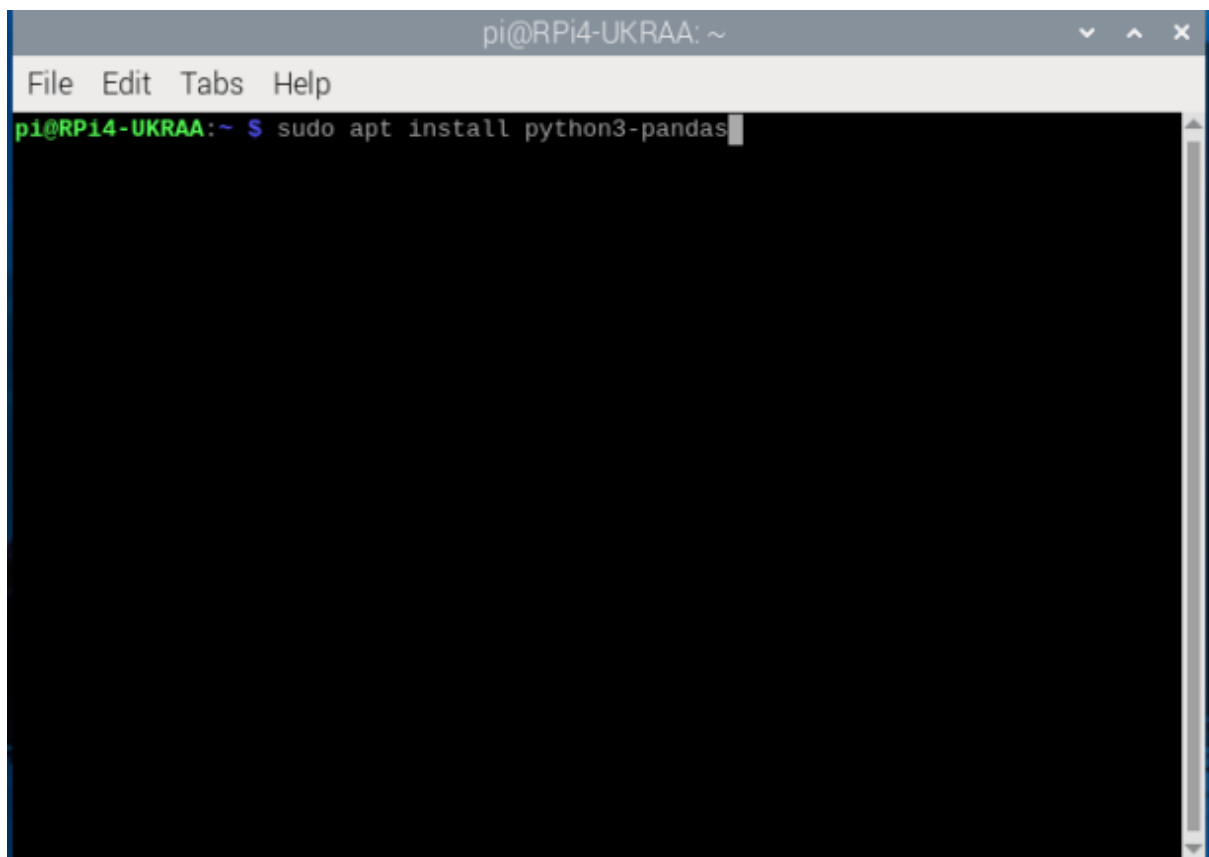
## Add additional software to your RPi

We need to add some dependencies to the RPi.

### pandas

1. Open a terminal window and type the command below and press enter. You will be asked to type **y** to proceed. This will install pandas and other dependencies related to pandas.

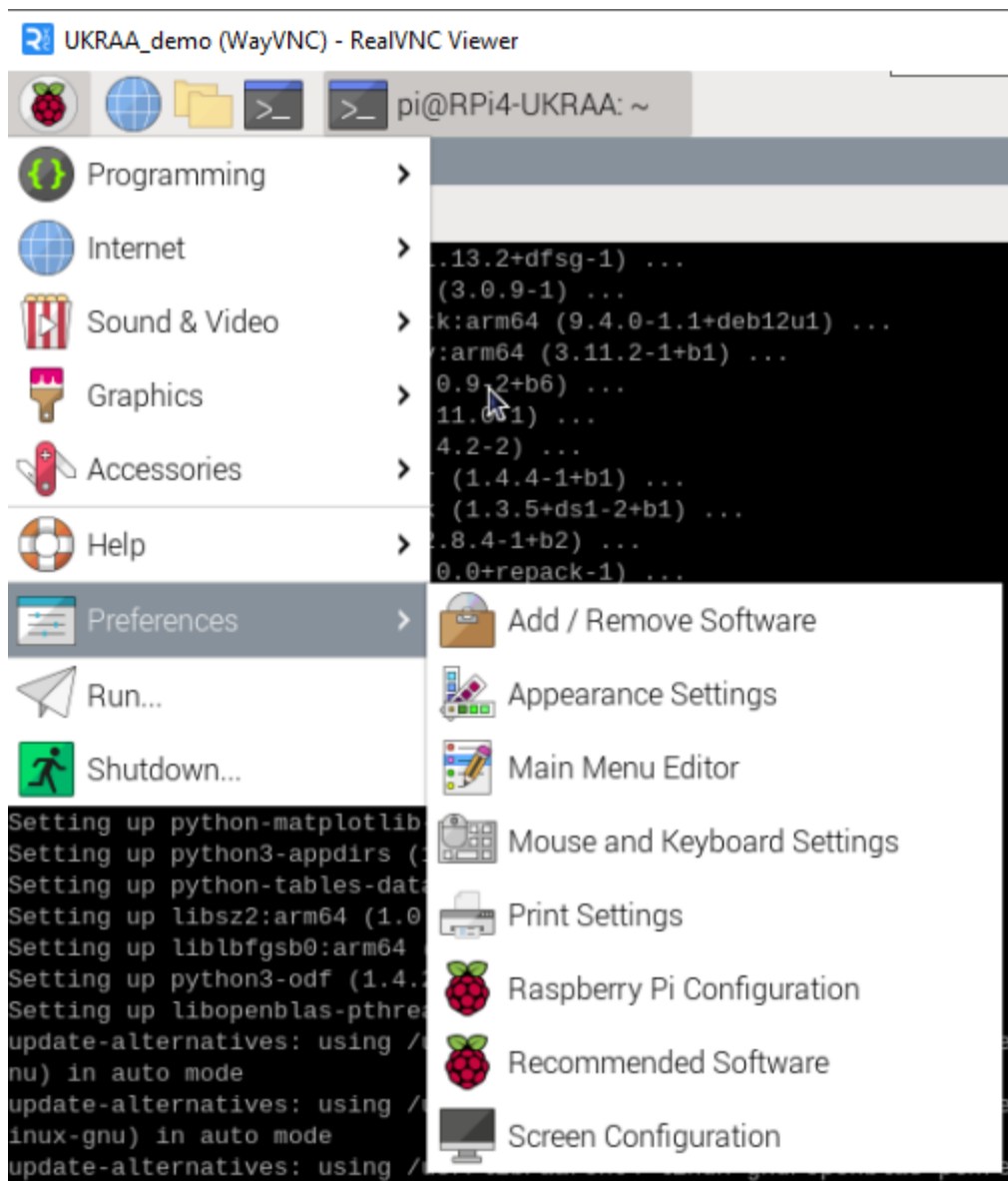
**sudo apt install python3-pandas**



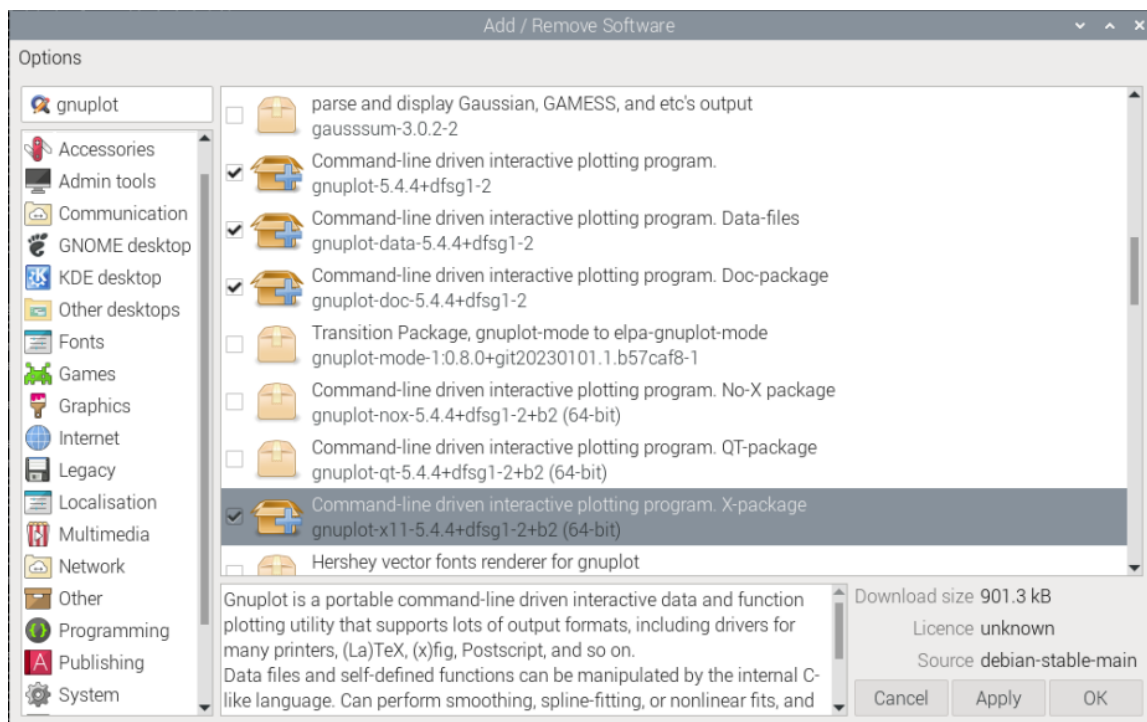
```
pi@RPi4-UKRAA: ~
File Edit Tabs Help
pi@RPi4-UKRAA:~ $ sudo apt install python3-pandas
```

## gnuplot

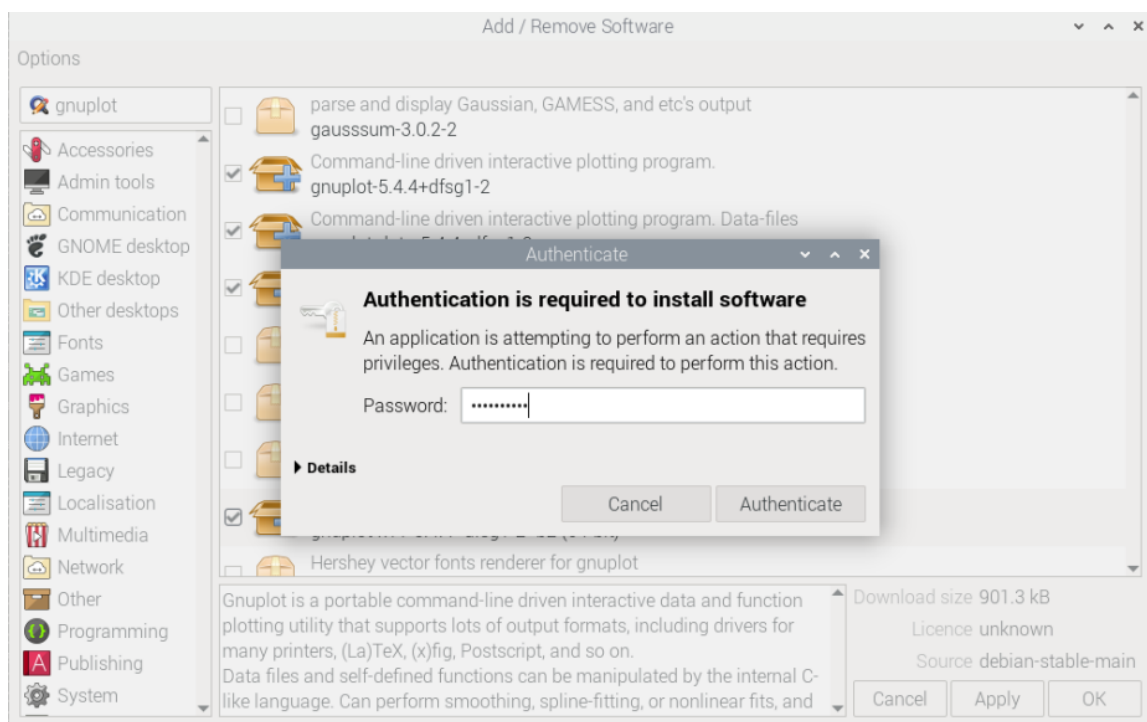
1. Open the application menu (Raspberry logo) and select **Preferences** and **Add/Remove Software**.



- From the **Add/Remove Software** search window type **gnuplot** into the search bar and press enter. When the available packages have been found, select the packages in the following image and select **Apply**.



- You will be asked to enter the pi **password**. Enter your pi **password** and select **Authenticate**. This will then install the gnuplot package and any dependencies required.



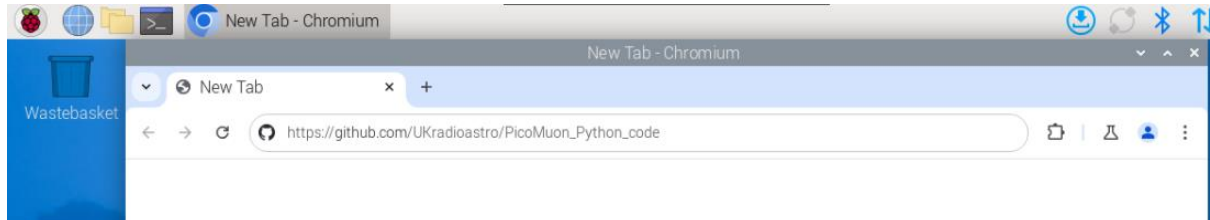
4. Select **OK** to close the window.



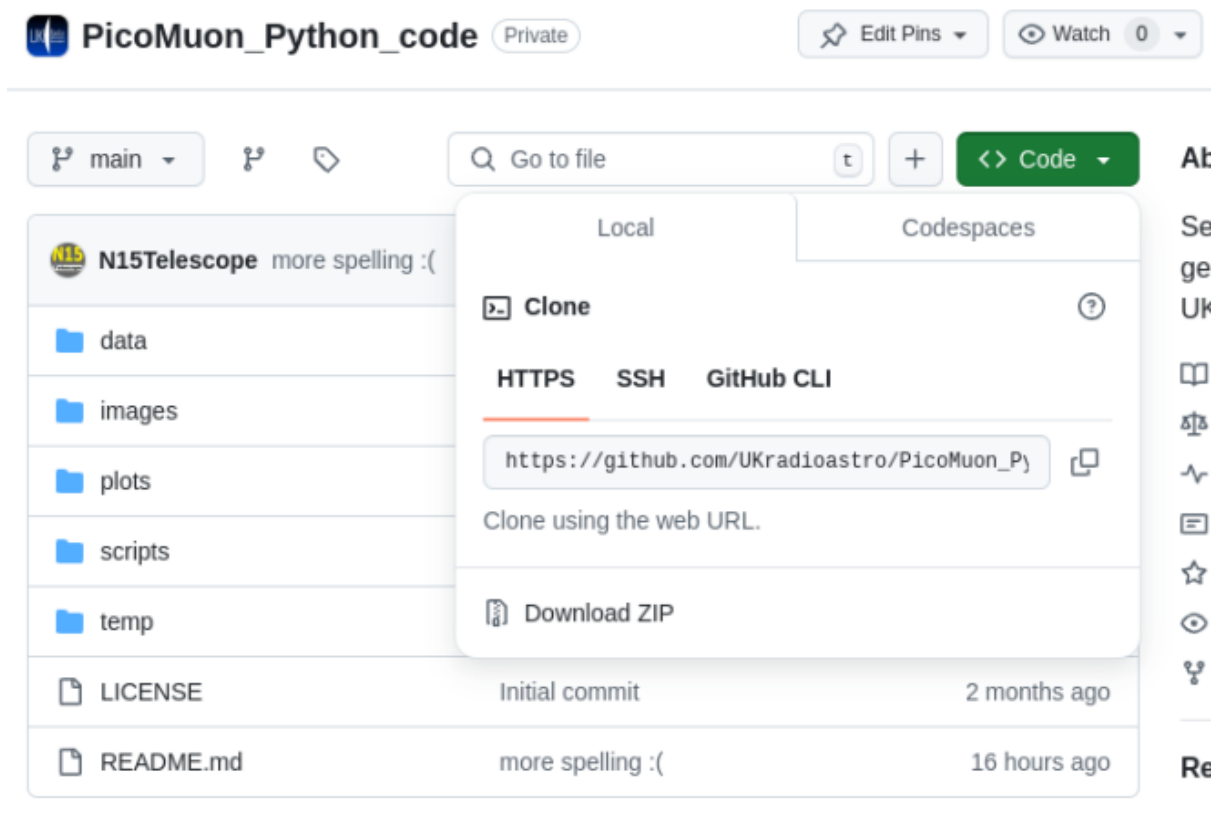
# Get code from github

1. Open the web browser on the RPi. In the address bar type the following command and press enter.

**[https://github.com/UKradioastro/PicoMuon\\_Python\\_code](https://github.com/UKradioastro/PicoMuon_Python_code)**

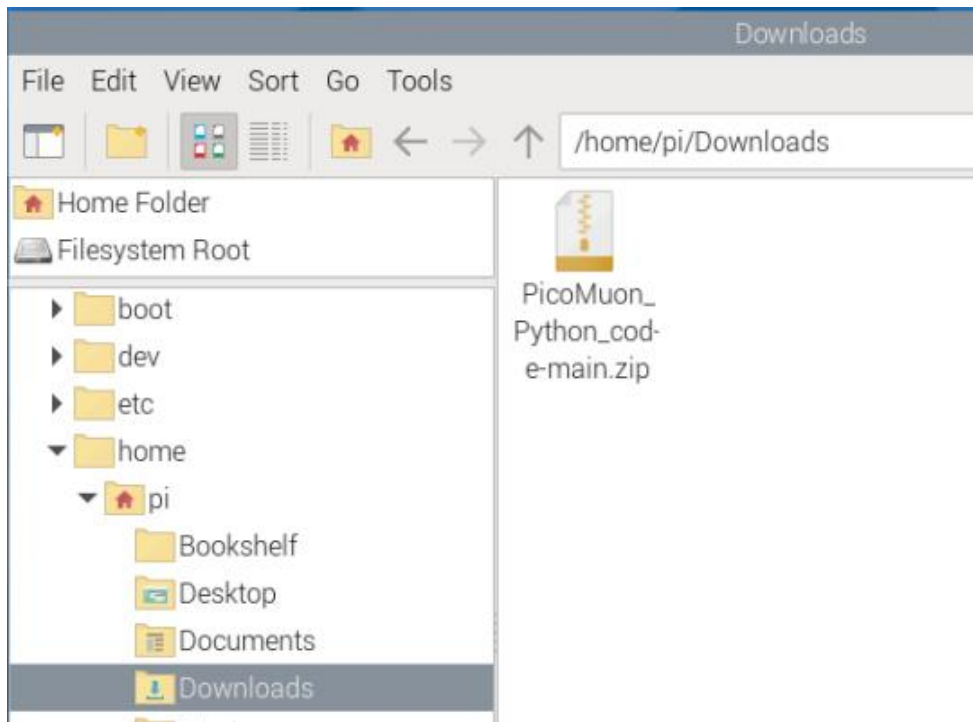


2. Select the **<> Code** tab and then select **Download ZIP**. This will download all the necessary code/files for the Python code. Close the web browser.

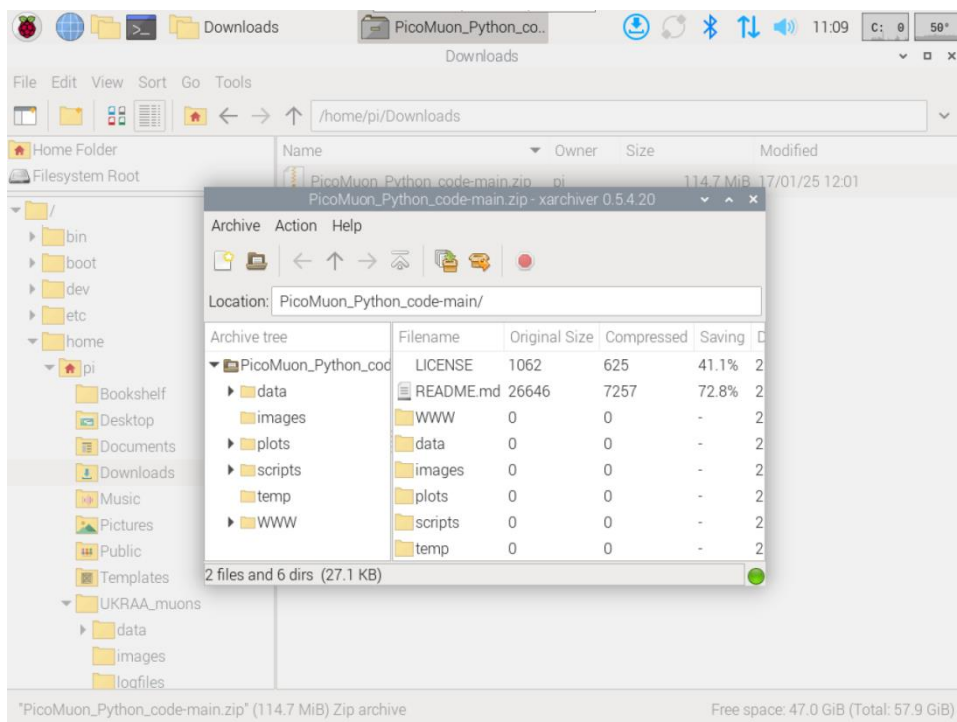




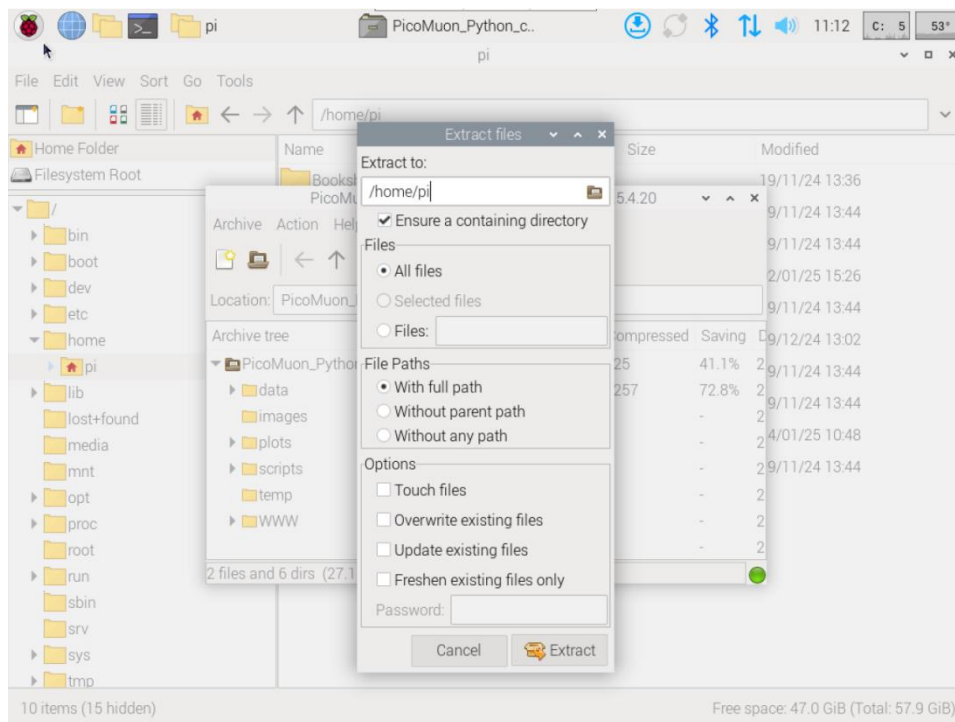
3. Open file manager and navigate to the **Downloads** folder - you should see a zip folder of the downloaded files.



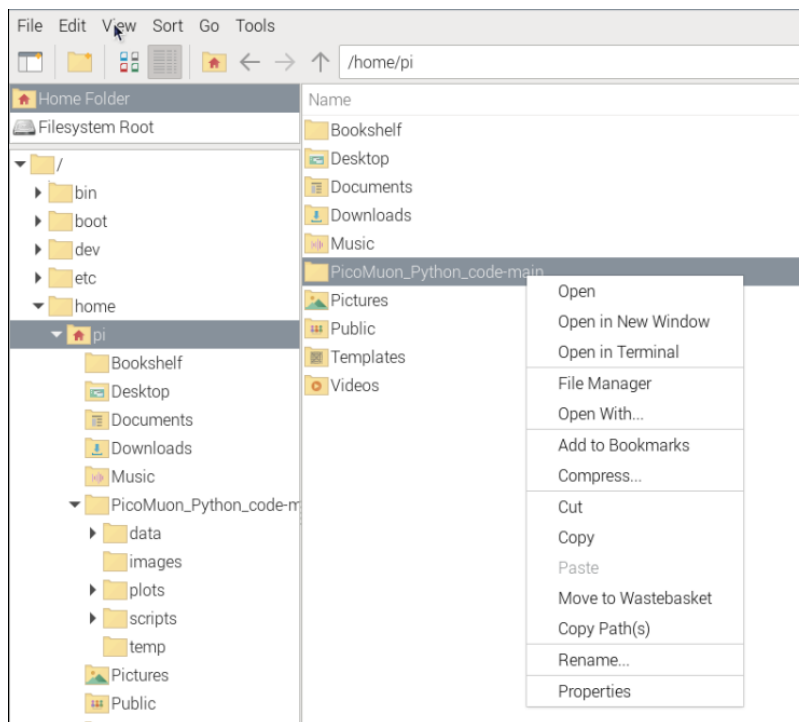
4. Double click on the zip folder to bring up the extraction tool. Select the **extract files** to extract all the files in the zip folder.



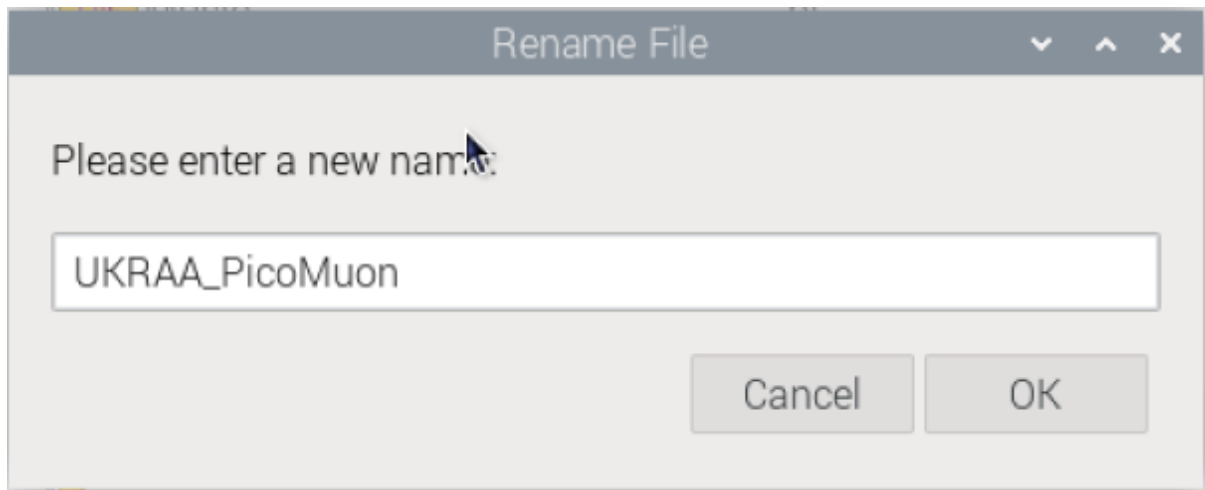
- You will be asked where you want the files to be extracted to - change the selection to the pi home directory - **/home/pi**. Select **Extract** to extract the files and then close the extraction tool when finished.



- From the open file manager navigate to **/home/pi** directory and you will see a new folder **PicoMuon\_Python\_code-main**. Select this folder and right-click mouse and select **Rename....**



7. Change the name of the folder to **UKRAA\_PicoMuon** and select **OK**.



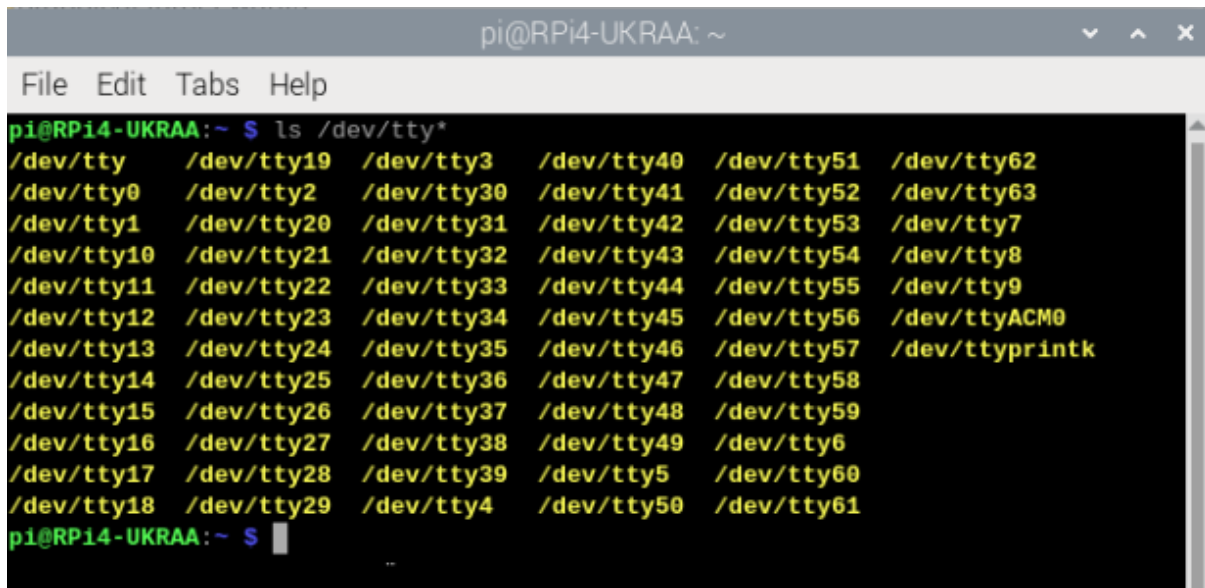
8. Scripts and folder structure should now be in place to run the code.
-

# Where is my detector?

Plug your detector into any of the RPi USB ports - I normally use the blue ports (USB3).

1. Open a terminal window and type the following command and press enter

**ls /dev/tty\***



```
pi@RPi4-UKRAA: ~  
File Edit Tabs Help  
pi@RPi4-UKRAA:~$ ls /dev/tty*  
/dev/tty /dev/tty19 /dev/tty3 /dev/tty40 /dev/tty51 /dev/tty62  
/dev/tty0 /dev/tty2 /dev/tty30 /dev/tty41 /dev/tty52 /dev/tty63  
/dev/tty1 /dev/tty20 /dev/tty31 /dev/tty42 /dev/tty53 /dev/tty7  
/dev/tty10 /dev/tty21 /dev/tty32 /dev/tty43 /dev/tty54 /dev/tty8  
/dev/tty11 /dev/tty22 /dev/tty33 /dev/tty44 /dev/tty55 /dev/tty9  
/dev/tty12 /dev/tty23 /dev/tty34 /dev/tty45 /dev/tty56 /dev/ttyACM0  
/dev/tty13 /dev/tty24 /dev/tty35 /dev/tty46 /dev/tty57 /dev/ttyprintk  
/dev/tty14 /dev/tty25 /dev/tty36 /dev/tty47 /dev/tty58  
/dev/tty15 /dev/tty26 /dev/tty37 /dev/tty48 /dev/tty59  
/dev/tty16 /dev/tty27 /dev/tty38 /dev/tty49 /dev/tty6  
/dev/tty17 /dev/tty28 /dev/tty39 /dev/tty5 /dev/tty60  
/dev/tty18 /dev/tty29 /dev/tty4 /dev/tty50 /dev/tty61  
pi@RPi4-UKRAA:~$
```

2. You are looking for **/dev/ttyACM0** - this is on the right hand side of the screen shot above.
3. This is the USB address for your attached detector - if you have more than one detector attached you may see **/dev/ttyACM1** etc.
4. If you do not see **/dev/ttyACM0**, then unplug and plug the detector back in and try again.
5. As long as we see **/dev/ttyACM0** then we do not have to make any changes to the python scripts, because they are looking for **ACM0**.

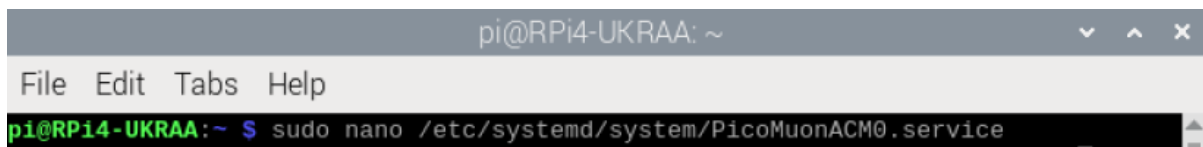
# Running the code to get data from the detector

It is an easy process to set up services to run the code, check it is running and restart the code if it stops.

To do this we need to create the service, and then enable the service.

1. Open terminal window
2. Type the following command and press enter - this will open nano text editor.

**`sudo nano /etc/systemd/system/PicoMuonACM0.service`**



```
pi@RPi4-UKRAA: ~  
File Edit Tabs Help  
pi@RPi4-UKRAA:~$ sudo nano /etc/systemd/system/PicoMuonACM0.service
```

3. Type in the following text...

**[Unit]**

**Description=**start GetDataRawACM0.py

**After=**multi-user.target

**[Service]**

**Type=**idle

**User=**pi

**ExecStart=**/usr/bin/python3

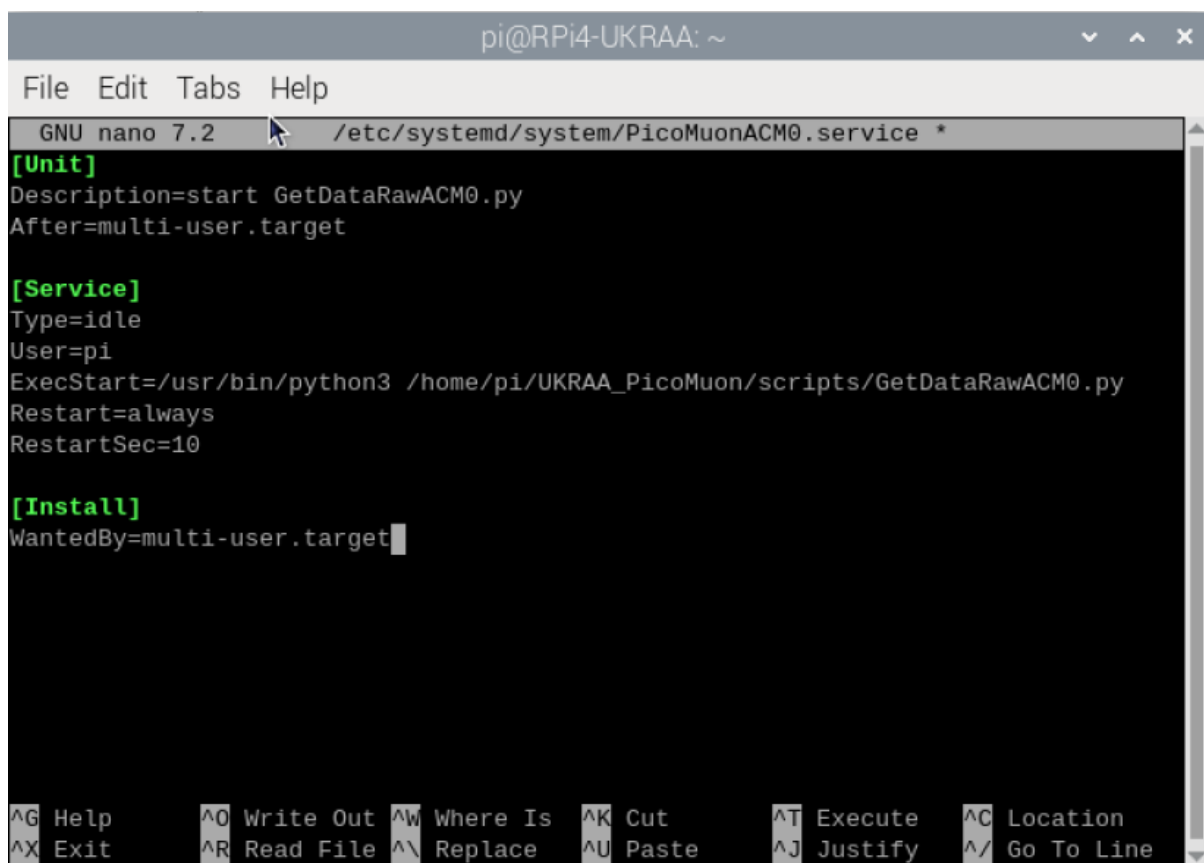
**/home/pi/UKRAA\_PicoMuon/scripts/GetDataRawACM0.py**

**Restart=**always

**RestartSec=**10

**[Install]**

**WantedBy=**multi-user.target



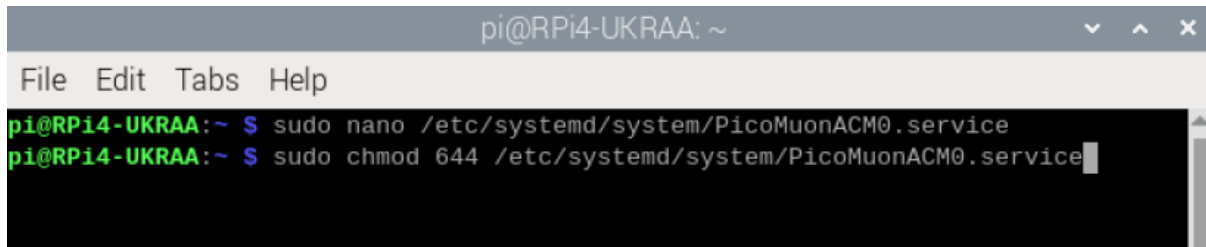
```
pi@RPi4-UKRAA: ~
File Edit Tabs Help
GNU nano 7.2 /etc/systemd/system/PicoMuonACM0.service *
[Unit]
Description=start GetDataRawACM0.py
After=multi-user.target

[Service]
Type=idle
User=pi
ExecStart=/usr/bin/python3 /home/pi/UKRAA_PicoMuon/scripts/GetDataRawACM0.py
Restart=always
RestartSec=10

[Install]
WantedBy=multi-user.target
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute  ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify  ^_ Go To Line
```

4. Save (Ctrl + s) and exit (Ctrl + x)
5. Type the following command and press enter - this will change permissions of your created service file and press enter.

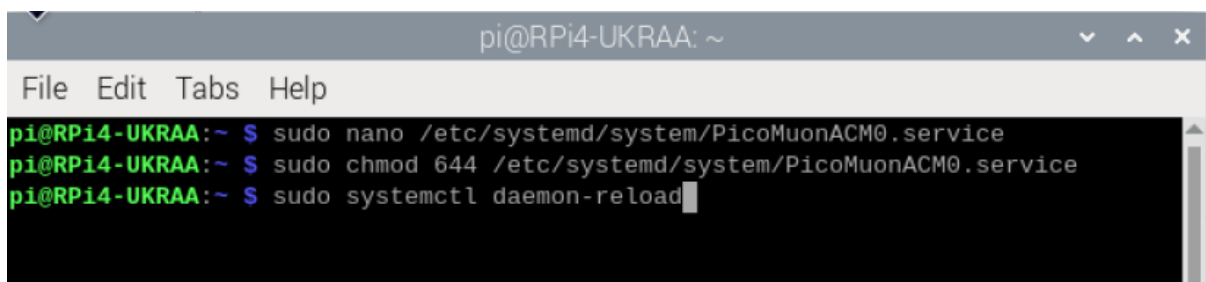
**sudo chmod 644 /etc/systemd/system/PicoMuonACM0.service**

A terminal window titled 'pi@RPi4-UKRAA: ~' with a menu bar 'File Edit Tabs Help'. The terminal shows two commands: 'sudo nano /etc/systemd/system/PicoMuonACM0.service' and 'sudo chmod 644 /etc/systemd/system/PicoMuonACM0.service'. The cursor is at the end of the second command.

```
pi@RPi4-UKRAA:~ $ sudo nano /etc/systemd/system/PicoMuonACM0.service
pi@RPi4-UKRAA:~ $ sudo chmod 644 /etc/systemd/system/PicoMuonACM0.service
```

6. Type the following command and press enter - this will reload services.

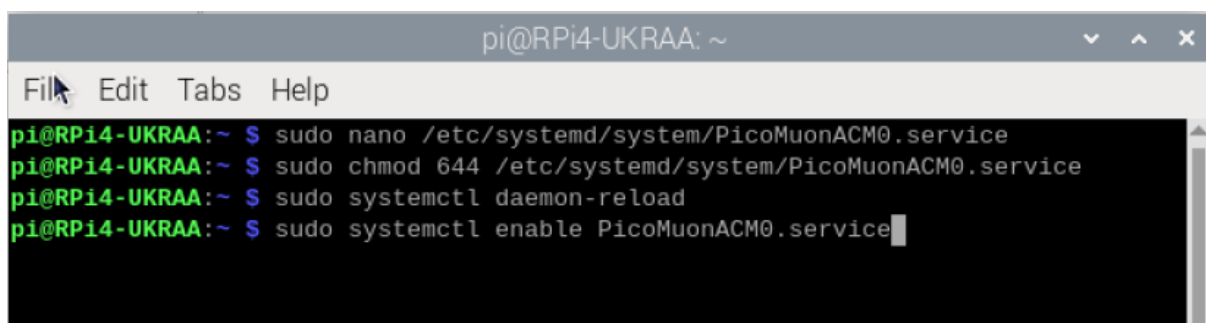
**sudo systemctl daemon-reload**

A terminal window titled 'pi@RPi4-UKRAA: ~' with a menu bar 'File Edit Tabs Help'. The terminal shows three commands: 'sudo nano /etc/systemd/system/PicoMuonACM0.service', 'sudo chmod 644 /etc/systemd/system/PicoMuonACM0.service', and 'sudo systemctl daemon-reload'. The cursor is at the end of the third command.

```
pi@RPi4-UKRAA:~ $ sudo nano /etc/systemd/system/PicoMuonACM0.service
pi@RPi4-UKRAA:~ $ sudo chmod 644 /etc/systemd/system/PicoMuonACM0.service
pi@RPi4-UKRAA:~ $ sudo systemctl daemon-reload
```

7. Type the following command and press enter - this will **enable** your new service.

**sudo systemctl enable PicoMuonACM0.service**

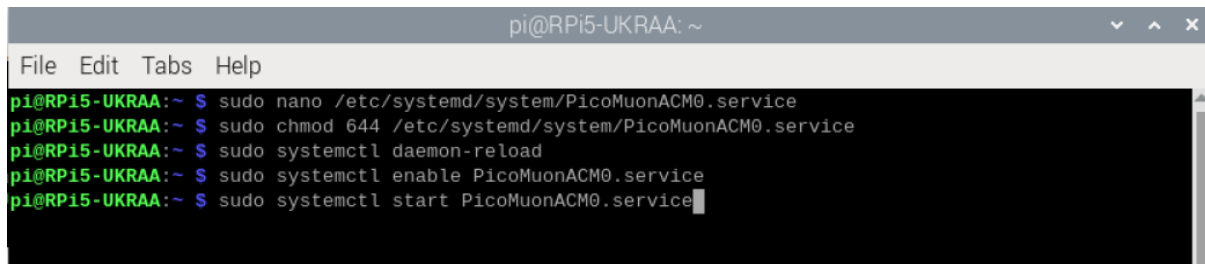
A terminal window titled 'pi@RPi4-UKRAA: ~' with a menu bar 'File Edit Tabs Help'. The terminal shows four commands: 'sudo nano /etc/systemd/system/PicoMuonACM0.service', 'sudo chmod 644 /etc/systemd/system/PicoMuonACM0.service', 'sudo systemctl daemon-reload', and 'sudo systemctl enable PicoMuonACM0.service'. The cursor is at the end of the fourth command.

```
pi@RPi4-UKRAA:~ $ sudo nano /etc/systemd/system/PicoMuonACM0.service
pi@RPi4-UKRAA:~ $ sudo chmod 644 /etc/systemd/system/PicoMuonACM0.service
pi@RPi4-UKRAA:~ $ sudo systemctl daemon-reload
pi@RPi4-UKRAA:~ $ sudo systemctl enable PicoMuonACM0.service
```

8. The service you have set will automatically start whenever the RPi is started or rebooted.

9. We need to **start** the service, type the following command and press enter

**sudo systemctl start PicoMuonACM0.service**

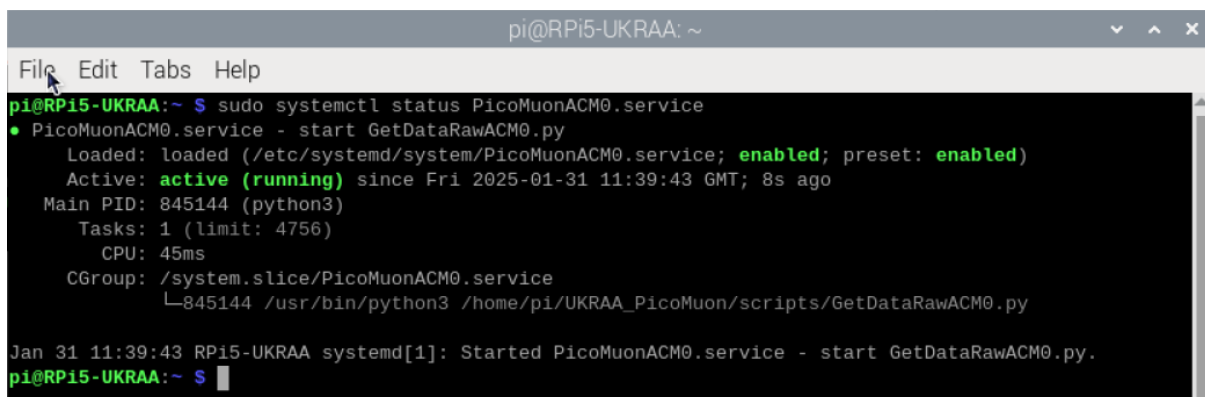
A terminal window titled 'pi@RPi5-UKRAA: ~' with a menu bar (File, Edit, Tabs, Help). The terminal shows a series of commands being entered: 'sudo nano /etc/systemd/system/PicoMuonACM0.service', 'sudo chmod 644 /etc/systemd/system/PicoMuonACM0.service', 'sudo systemctl daemon-reload', 'sudo systemctl enable PicoMuonACM0.service', and finally 'sudo systemctl start PicoMuonACM0.service' with the cursor at the end of the line.

```
pi@RPi5-UKRAA:~ $ sudo nano /etc/systemd/system/PicoMuonACM0.service
pi@RPi5-UKRAA:~ $ sudo chmod 644 /etc/systemd/system/PicoMuonACM0.service
pi@RPi5-UKRAA:~ $ sudo systemctl daemon-reload
pi@RPi5-UKRAA:~ $ sudo systemctl enable PicoMuonACM0.service
pi@RPi5-UKRAA:~ $ sudo systemctl start PicoMuonACM0.service
```

11. To check **status** of your service, type the following command and press enter

**sudo systemctl status PicoMuonACM0.service**

12. We expect to see...

A terminal window titled 'pi@RPi5-UKRAA: ~' with a menu bar (File, Edit, Tabs, Help). The terminal shows the command 'sudo systemctl status PicoMuonACM0.service' being entered. The output shows the service is loaded and active (running).

```
pi@RPi5-UKRAA:~ $ sudo systemctl status PicoMuonACM0.service
● PicoMuonACM0.service - start GetDataRawACM0.py
   Loaded: loaded (/etc/systemd/system/PicoMuonACM0.service; enabled; preset: enabled)
   Active: active (running) since Fri 2025-01-31 11:39:43 GMT; 8s ago
     Main PID: 845144 (python3)
        Tasks: 1 (limit: 4756)
           CPU: 45ms
      CGroup: /system.slice/PicoMuonACM0.service
              └─845144 /usr/bin/python3 /home/pi/UKRAA_PicoMuon/scripts/GetDataRawACM0.py

Jan 31 11:39:43 RPi5-UKRAA systemd[1]: Started PicoMuonACM0.service - start GetDataRawACM0.py.
pi@RPi5-UKRAA:~ $
```

13. If we don't see green **enabled** and **active**, then we have a typo in the **PicoMuonACM0.service** file.

14. To **start** your service, type the following command and press enter.

**sudo systemctl start PicoMuonACM0.service**

15. To **stop** your service, type the following command and press enter.

**sudo systemctl stop PicoMuonACM0.service**

16. To check the **status** of your service, type the following command and press enter.

**sudo systemctl status PicoMuonACM0.service**



17. To **enable** your service, type the following command and press enter.

**sudo systemctl enable PicoMuonACM0.service**

18. To **disable** your service, type the following command and press enter.

**sudo systemctl disable PicoMuonACM0.service**

---

# Running the code to process and plot the data from the detector

The data will be processed to get counts per minute and the frequency of the adc values for the previous day.

A request will be made to [NMDB](#) via NEST to get the previous days recorded neutron count; this data will be overlaid onto the counts per minute graphs.

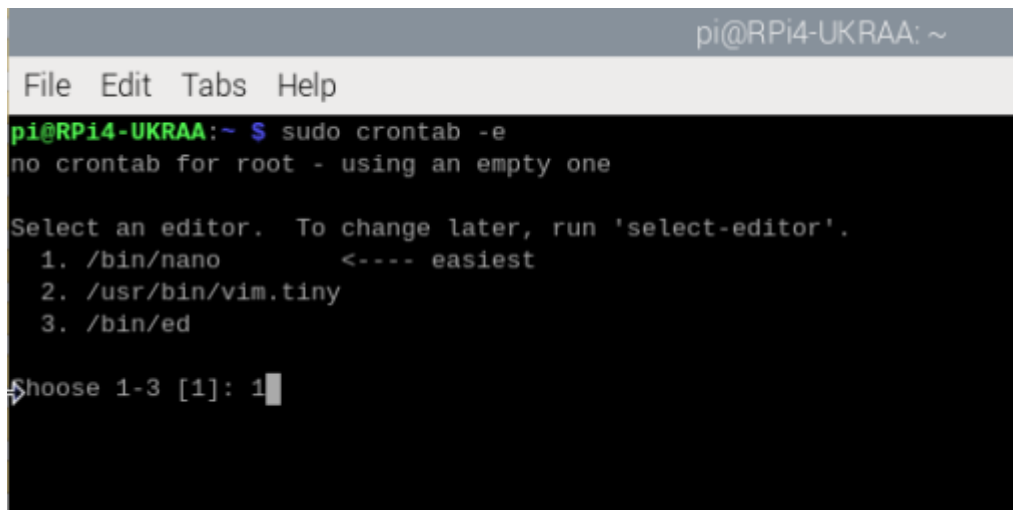
Three plots will be created:

- counts per minute
- frequency of counts per minute
- frequency of ADC values recorded
- 

This can be done after midnight automatically using **CRON** because the processing of the cpm takes about 6 hours and the processing of the adc values takes about 1 hour on a RPi4. It takes less time to run these two processes on a RPi5.

1. Open terminal window
2. Type the following command and press enter - this will open crontab text editor. The first time it will ask what editor you prefer - I prefer nano.

**sudo crontab -e**



```
pi@RPi4-UKRAA: ~  
File Edit Tabs Help  
pi@RPi4-UKRAA:~ $ sudo crontab -e  
no crontab for root - using an empty one  
  
Select an editor. To change later, run 'select-editor'.  
 1. /bin/nano          <---- easiest  
 2. /usr/bin/vim.tiny  
 3. /bin/ed  
  
$ choose 1-3 [1]: 1
```

3. Scroll to bottom and type the following after **# m h dom mon dow command...**

**# m h dom mon dow command**

**# cron entry to get neutron data from NMDC NEST**

```
10 00 * * * su pi -c "/usr/bin/python3
/home/pi/UKRAA_PicoMuon/scripts/GetDataNeutron.py >>
/home/pi/UKRAA_PicoMuon/logfiles/cron-PicoMuonACM0.log 2>&1"
```

**# cron entry to process yesterdays raw muon counts per minute**

```
30 00 * * * su pi -c "/usr/bin/python3
/home/pi/UKRAA_PicoMuon/scripts/ProcessDataCpmACM0.py >>
/home/pi/UKRAA_PicoMuon/logfiles/cron-PicoMuonACM0.log 2>&1"
```

```
30 02 * * * su pi -c "/usr/bin/python3
/home/pi/UKRAA_PicoMuon/scripts/ProcessDataDayACM0.py >>
/home/pi/UKRAA_PicoMuon/logfiles/cron-PicoMuonACM0.log 2>&1"
```

```
30 04 * * * su pi -c "/usr/bin/python3
/home/pi/UKRAA_PicoMuon/scripts/ProcessDataWeekACM0.py >>
/home/pi/UKRAA_PicoMuon/logfiles/cron-PicoMuonACM0.log 2>&1"
```

```
30 05 * * * su pi -c "/usr/bin/python3
/home/pi/UKRAA_PicoMuon/scripts/ProcessDataMonthACM0.py >>
/home/pi/UKRAA_PicoMuon/logfiles/cron-PicoMuonACM0.log 2>&1"
```

**# cron entry to process yesterdays raw muon adc values**

```
45 00 * * * su pi -c "/usr/bin/python3
/home/pi/UKRAA_PicoMuon/scripts/ProcessDataAdcACM0.py >>
/home/pi/UKRAA_PicoMuon/logfiles/cron-PicoMuonACM0.log 2>&1"
```

**# cron entry to plot yesterdays counts per minute**

```
50 07 * * * su pi -c "/usr/bin/gnuplot
/home/pi/UKRAA_PicoMuon/scripts/PlotDataCpmACM0.gp >>
/home/pi/UKRAA_PicoMuon/logfiles/cron-PicoMuonACM0.log 2>&1"
```

**# cron entry to plot yesterdays counts per minute frequency**

```
51 07 * * * su pi -c "/usr/bin/gnuplot
/home/pi/UKRAA_PicoMuon/scripts/PlotDataFreqACM0.gp >>
/home/pi/UKRAA_PicoMuon/logfiles/cron-PicoMuonACM0.log 2>&1"
```

**# cron entry to plot yesterdays adc frequency**

```
52 07 * * * su pi -c "/usr/bin/gnuplot
/home/pi/UKRAA_PicoMuon/scripts/PlotDataAdcACM0.gp  >>
/home/pi/UKRAA_PicoMuon/logfiles/cron-PicoMuonACM0.log 2>&1"
```

# cron entry to plot yesterdays % muons & % neutron deviation

```
53 07 * * * su pi -c "/usr/bin/gnuplot
/home/pi/UKRAA_PicoMuon/scripts/PlotDataDayACM0.gp  >>
/home/pi/UKRAA_PicoMuon/logfiles/cron-PicoMuonACM0.log 2>&1"
```

# cron entry to plot last weeks % muons & % neutron deviation

```
54 07 * * * su pi -c "/usr/bin/gnuplot
/home/pi/UKRAA_PicoMuon/scripts/PlotDataWeekACM0.gp  >>
/home/pi/UKRAA_PicoMuon/logfiles/cron-PicoMuonACM0.log 2>&1">
```

# cron entry to plot last months % muons & % neutron deviation

```
55 07 * * * su pi -c "/usr/bin/gnuplot
/home/pi/UKRAA_PicoMuon/scripts/PlotDataMonthACM0.gp  >>
/home/pi/UKRAA_PicoMuon/logfiles/cron-PicoMuonACM0.log 2>&1"
```

```
# m h dom mon dow   command
# cron entry to get neutron data from NMDC NEST
10 00 * * * su pi -c "/usr/bin/python3 /home/pi/UKRAA_PicoMuon/scripts/GetDataNeutron.py      >> /home/pi/UKRAA_PicoMuon/log>
# cron entry to process yesterdays raw muon counts per minute
30 00 * * * su pi -c "/usr/bin/python3 /home/pi/UKRAA_PicoMuon/scripts/ProcessDataCpmACM0.py      >> /home/pi/UKRAA_PicoMuon/log>
30 02 * * * su pi -c "/usr/bin/python3 /home/pi/UKRAA_PicoMuon/scripts/ProcessDataDayACM0.py      >> /home/pi/UKRAA_PicoMuon/log>
30 04 * * * su pi -c "/usr/bin/python3 /home/pi/UKRAA_PicoMuon/scripts/ProcessDataWeekACM0.py      >> /home/pi/UKRAA_PicoMuon/log>
30 05 * * * su pi -c "/usr/bin/python3 /home/pi/UKRAA_PicoMuon/scripts/ProcessDataMonthACM0.py      >> /home/pi/UKRAA_PicoMuon/log>
# cron entry to process yesterdays raw muon adc values
45 00 * * * su pi -c "/usr/bin/python3 /home/pi/UKRAA_PicoMuon/scripts/ProcessDataAdcACM0.py      >> /home/pi/UKRAA_PicoMuon/log>
# cron entry to plot yesterdays counts per minute
50 07 * * * su pi -c "/usr/bin/gnuplot /home/pi/UKRAA_PicoMuon/scripts/PlotDataCpmACM0.gp      >> /home/pi/UKRAA_PicoMuon/log>
# cron entry to plot yesterdays counts per minute frequency
51 07 * * * su pi -c "/usr/bin/gnuplot /home/pi/UKRAA_PicoMuon/scripts/PlotDataFreqACM0.gp      >> /home/pi/UKRAA_PicoMuon/log>
# cron entry to plot yesterdays adc frequency
52 07 * * * su pi -c "/usr/bin/gnuplot /home/pi/UKRAA_PicoMuon/scripts/PlotDataAdcACM0.gp      >> /home/pi/UKRAA_PicoMuon/log>
# cron entry to plot yesterdays % muons & % neutron deviation
53 07 * * * su pi -c "/usr/bin/gnuplot /home/pi/UKRAA_PicoMuon/scripts/PlotDataDayACM0.gp      >> /home/pi/UKRAA_PicoMuon/log>
# cron entry to plot last weeks % muons & % neutron deviation
54 07 * * * su pi -c "/usr/bin/gnuplot /home/pi/UKRAA_PicoMuon/scripts/PlotDataWeekACM0.gp      >> /home/pi/UKRAA_PicoMuon/log>
# cron entry to plot last months % muons & % neutron deviation
55 07 * * * su pi -c "/usr/bin/gnuplot /home/pi/UKRAA_PicoMuon/scripts/PlotDataMonthACM0.gp      >> /home/pi/UKRAA_PicoMuon/log>
```

4. Save (Ctrl + s) and exit (Ctrl + x).
5. If this went well you will see **crontab: installing new crontab** on the terminal window.

This should now process and graph yesterday's data from the detector overnight.

Plots will appear in the **plots** folder and the **temp** folder.

The **plots** folder holds plots for each day in year/month folder.

The **temp** folder only holds plots for yesterday, updated each day, for pushing to a website/etc.

---

# Creating simple home intranet web server on your RPi

We can create a simple web server on our RPi so that we can view our detector's results on our smart phone when connected to our home network.

To set up the web server on the RPi, follow the instruction from [tom's HARDWARE](#). Only need to do first section – up to **9. Build your website**.

Distilled versions of these instructions are included at the end of this manual.

---

# Creating simple website for you PicoMuon plots to view your results on your smart phone

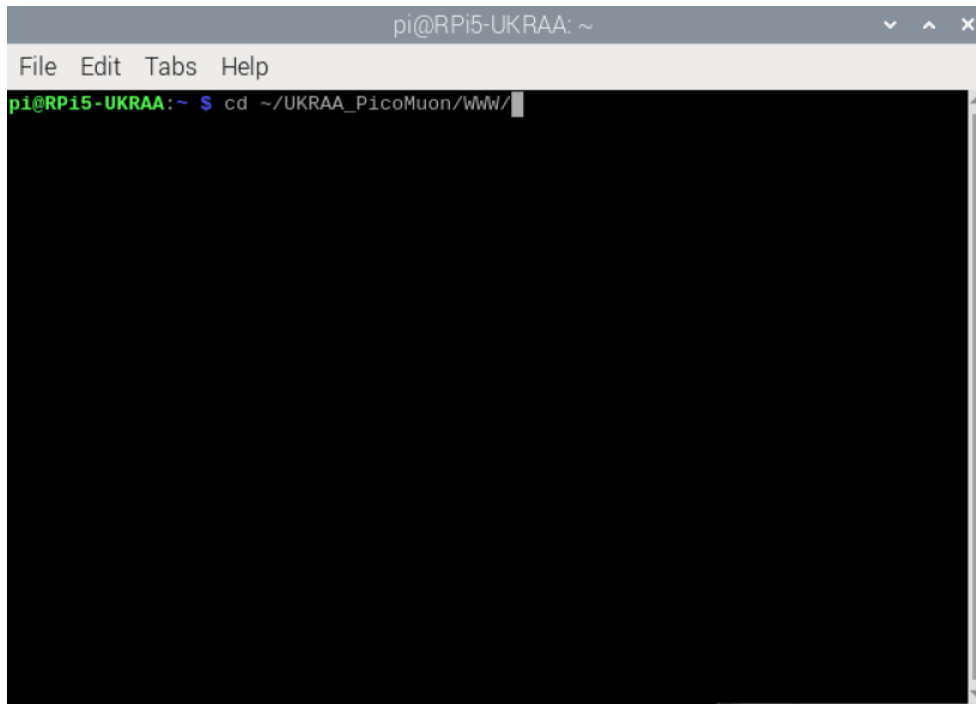
Once you have completed building your RPi web server, we need to move some files over to the `/var/www/html` folder on your RPi; these are all located in the WWW folder from your downloaded git zip file - as shown below.

```
└─ 📁 WWW
  └─ 📁 images
    └─ default.png
    └─ favicon.ico
    └─ PicoMuon_1.png
    └─ PicoMuon_2.png
    └─ PicoMuon_3.png
    └─ SunHeader.png
    └─ UKRAA_logo_black.png
    └─ UKRAA_logo_white.png
  └─ 📁 temp
    └─ ACM0_B-Adc.png
    └─ ACM0_B-Cpm.png
    └─ ACM0_B-Freq.png
    └─ ACM0_C-Adc.png
    └─ ACM0_C-Cpm.png
    └─ ACM0_C-Freq.png
    └─ ACM0_day_plot.png
    └─ ACM0_T-Adc.png
    └─ ACM0_T-Cpm.png
    └─ ACM0_T-Freq.png
    └─ ACM0_week_plot.png
  └─ index.html
```

1. Open terminal window

2. Type the following command and press enter - this will take you to the files in the WWW folder.

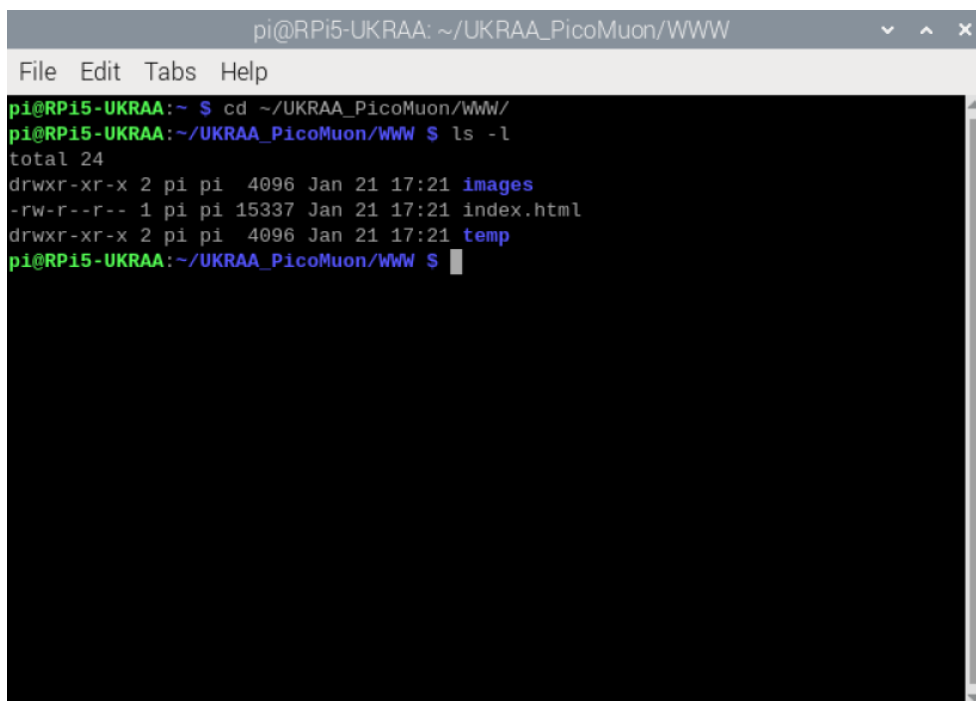
**cd ~/UKRAA\_PicoMuon/WWW/**



```
pi@RPi5-UKRAA: ~  
File Edit Tabs Help  
pi@RPi5-UKRAA:~ $ cd ~/UKRAA_PicoMuon/WWW/
```

3. We can check we are in the correct location by typing the following command and press enter, we should see the following...

**ls -l**

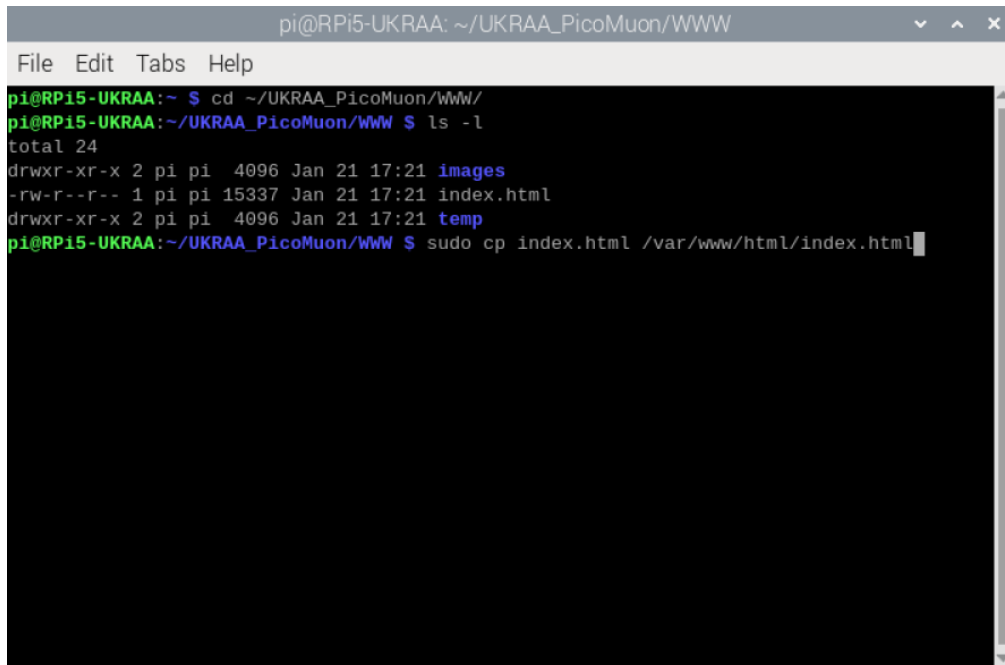


```
pi@RPi5-UKRAA: ~/UKRAA_PicoMuon/WWW  
File Edit Tabs Help  
pi@RPi5-UKRAA:~ $ cd ~/UKRAA_PicoMuon/WWW/  
pi@RPi5-UKRAA:~/UKRAA_PicoMuon/WWW $ ls -l  
total 24  
drwxr-xr-x 2 pi pi 4096 Jan 21 17:21 images  
-rw-r--r-- 1 pi pi 15337 Jan 21 17:21 index.html  
drwxr-xr-x 2 pi pi 4096 Jan 21 17:21 temp  
pi@RPi5-UKRAA:~/UKRAA_PicoMuon/WWW $
```



4. We now need to copy the files and folders from `~/UKRAA_PicoMuon/WWW` to `/var/www/html`.
5. Type the following command and press enter - this will copy the **index.html** file.

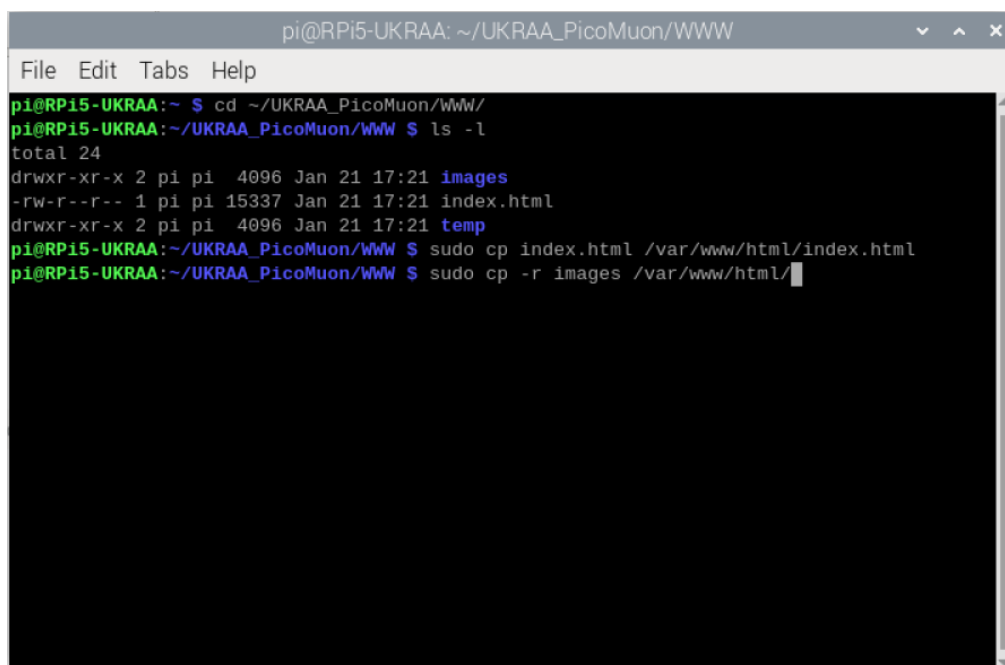
**`sudo cp index.html /var/www/html/index.html`**



```
pi@RPi5-UKRAA: ~/UKRAA_PicoMuon/WWW
File Edit Tabs Help
pi@RPi5-UKRAA:~ $ cd ~/UKRAA_PicoMuon/WWW/
pi@RPi5-UKRAA:~/UKRAA_PicoMuon/WWW $ ls -l
total 24
drwxr-xr-x 2 pi pi 4096 Jan 21 17:21 images
-rw-r--r-- 1 pi pi 15337 Jan 21 17:21 index.html
drwxr-xr-x 2 pi pi 4096 Jan 21 17:21 temp
pi@RPi5-UKRAA:~/UKRAA_PicoMuon/WWW $ sudo cp index.html /var/www/html/index.html
```

6. Type the following command and press enter - this will copy the **images** directory and content.

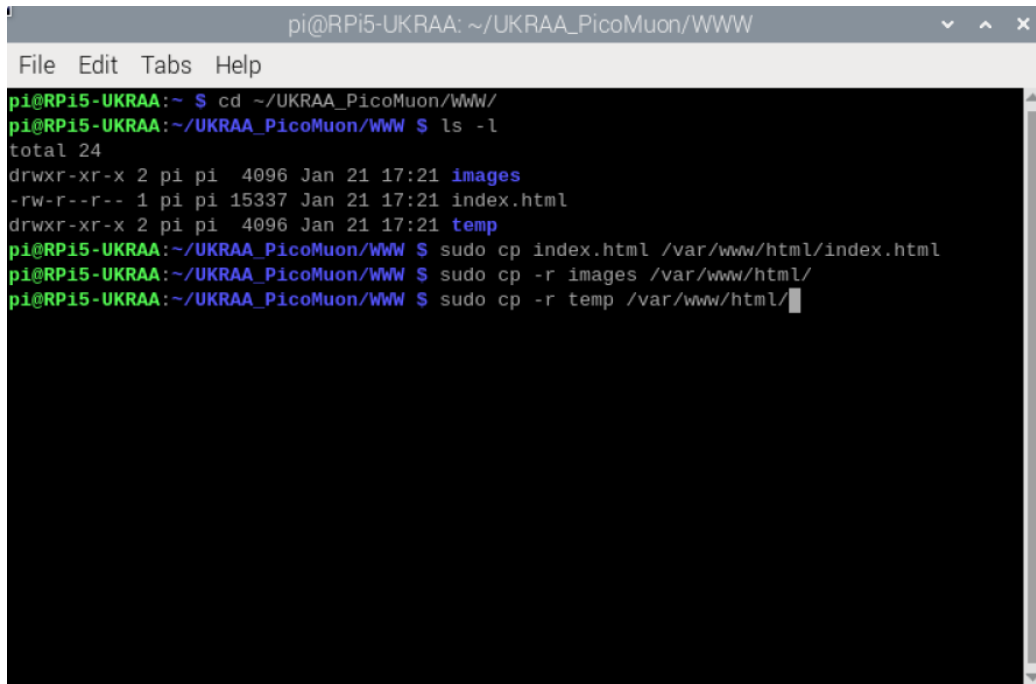
**`sudo cp -r images /var/www/html/`**



```
pi@RPi5-UKRAA: ~/UKRAA_PicoMuon/WWW
File Edit Tabs Help
pi@RPi5-UKRAA:~ $ cd ~/UKRAA_PicoMuon/WWW/
pi@RPi5-UKRAA:~/UKRAA_PicoMuon/WWW $ ls -l
total 24
drwxr-xr-x 2 pi pi 4096 Jan 21 17:21 images
-rw-r--r-- 1 pi pi 15337 Jan 21 17:21 index.html
drwxr-xr-x 2 pi pi 4096 Jan 21 17:21 temp
pi@RPi5-UKRAA:~/UKRAA_PicoMuon/WWW $ sudo cp index.html /var/www/html/index.html
pi@RPi5-UKRAA:~/UKRAA_PicoMuon/WWW $ sudo cp -r images /var/www/html/
```

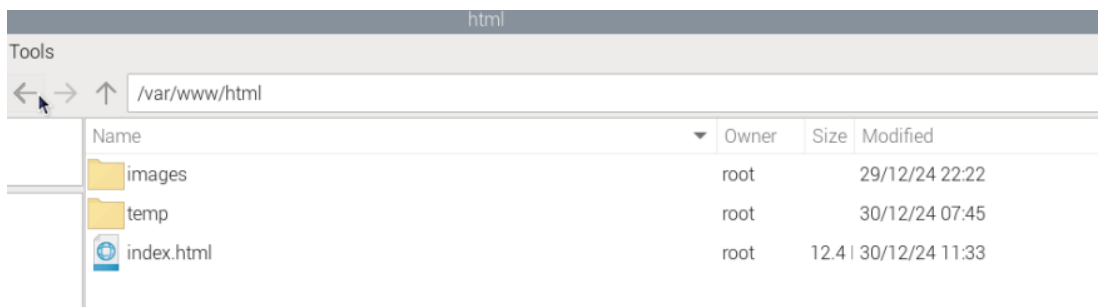
7. Type the following command and press enter - this will copy the **temp** directory and content.

**sudo cp -r temp /var/www/html/**

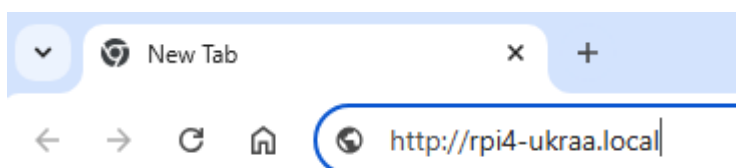


```
pi@RPi5-UKRAA: ~/UKRAA_PicoMuon/WWW
File Edit Tabs Help
pi@RPi5-UKRAA:~$ cd ~/UKRAA_PicoMuon/WWW/
pi@RPi5-UKRAA:~/UKRAA_PicoMuon/WWW$ ls -l
total 24
drwxr-xr-x 2 pi pi 4096 Jan 21 17:21 images
-rw-r--r-- 1 pi pi 15337 Jan 21 17:21 index.html
drwxr-xr-x 2 pi pi 4096 Jan 21 17:21 temp
pi@RPi5-UKRAA:~/UKRAA_PicoMuon/WWW$ sudo cp index.html /var/www/html/index.html
pi@RPi5-UKRAA:~/UKRAA_PicoMuon/WWW$ sudo cp -r images /var/www/html/
pi@RPi5-UKRAA:~/UKRAA_PicoMuon/WWW$ sudo cp -r temp /var/www/html/
```

8. From file manager, you can check that the file/folders have been copied correctly, by navigating to **/var/www/html** and you should see the following.



9. You should now be able to access the webpage from your smart phone. On your smart phone open your preferred web application (Safari, chrome, etc...). In the search bar type the hostname of your RPi, e.g. <http://rpi4-ukraa.local>, and press enter - should access the web page.



# Get daily plots to your website

The data on the website is static - we need to push the daily plots to the website so that they can be updated and viewed.

This can be done after the plots are completed automatically using **CRON**.

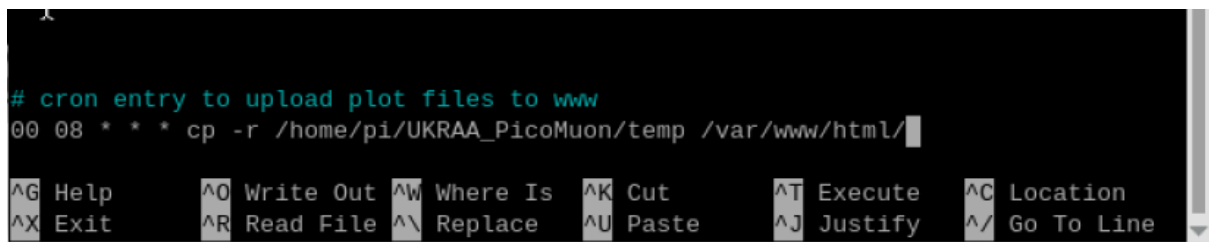
1. Open terminal window
2. Type the following command and press enter - this will open crontab text editor.

## **sudo crontab -e**

3. Scroll to bottom and type the following after what you had previously typed in above...

### **# cron entry to upload plot files to www**

**00 08 \* \* \* cp -r /home/pi/UKRAA\_PicoMuon/temp /var/www/html/**



```
# cron entry to upload plot files to www
00 08 * * * cp -r /home/pi/UKRAA_PicoMuon/temp /var/www/html/
```

^G Help    ^O Write Out    ^W Where Is    ^K Cut    ^T Execute    ^C Location  
^X Exit    ^R Read File    ^\ Replace    ^U Paste    ^J Justify    ^\_ Go To Line

4. Save (Ctrl + s) and exit (Ctrl + x).
5. If this went well you will see **crontab: installing new crontab** on the terminal window.

This should copy yesterday's plots at 8.00am each morning to the website for viewing.

# How to Set Up a Raspberry Pi Web Server

How-to by Avram Piltch last updated 17 September 2022

*Host a website from your Raspberry Pi.*

One of the most popular uses of the Raspberry Pi is as a web server that lives on your local network. Whether you need an Intranet for your office or a small server for doing web development, the Pi is a great choice. In fact, at Tom's Hardware, we have a local Pi web server that we use to deliver the content for our laptop battery test, which involves continuous surfing over Wi-Fi.

To get your web server working, you'll need a Raspberry Pi that's connected to your local network and running a fairly-recent version of the Raspbian operating system. These instructions will work on just about any model, including the powerful Raspberry Pi 4 and diminutive Raspberry Pi Zero W. If you need to install Raspbian, see our tutorial on how to set up a Raspberry Pi or, better yet, how to do a headless install (no keyboard or screen required).

1. **Navigate to the command prompt / terminal.** You can get there by hitting CTRL+ALT+T from the Raspbian desktop or connecting remotely via SSH if you have that configured. Some users also configure the Raspberry Pi to boot directly to the command prompt.

2. **Update your packages** by typing:

**sudo apt-get update**

This will make sure that you get the latest versions of every file you download after this.

3. **Install apache2** with the command:

**sudo apt-get install apache2 -y**

4. **Install php** for your sever by typing:

**sudo apt-get install php libapache2-mod-php -y**

5. **Install mariadb** so you can use a mysql database with your website. You start by typing:

**sudo apt-get install mariadb-server**

Then, after the download is finished, you must do the formal install by typing:

**sudo mysql\_secure\_installation**

You will be asked for a root password. You can leave it blank.

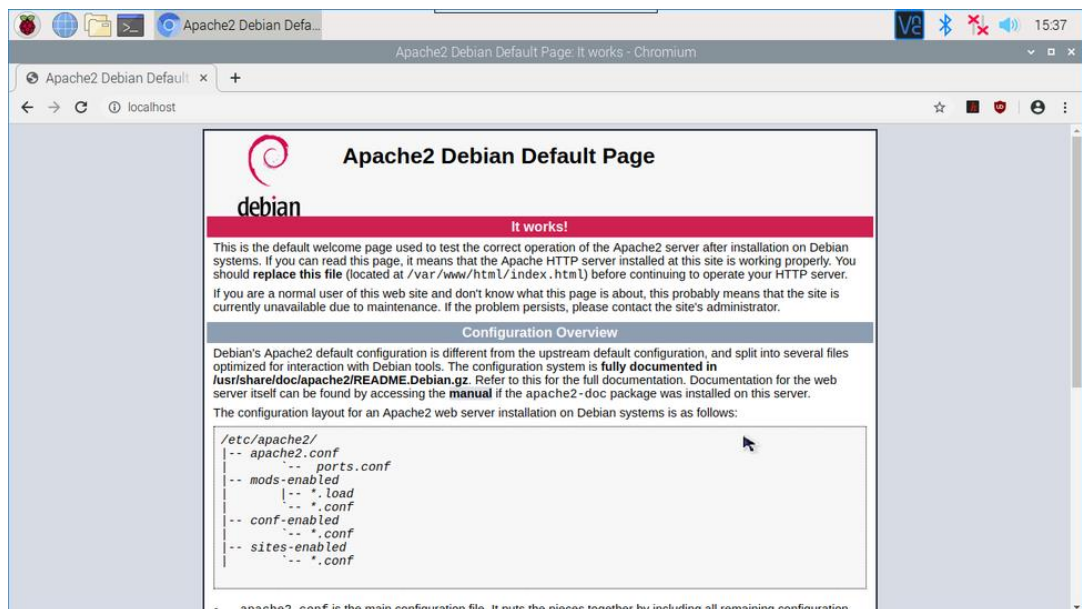
6. **Install the php-mysql connector** so php pages can access the DB.

**sudo apt install php-mysql**

7. **Restart apache2** so all of the changes are running.

**sudo service apache2 restart**

8. **Test your server.** On the Raspberry Pi itself, you should be able to go to `http://localhost` and see a test page. From another computer on the same network, you should be able to get there by visiting `http://raspberrypi.local` or `http://raspberrypi`, provided that your Raspberry Pi's hostname is `raspberrypi`.



9. **Build your website** by putting html or PHP files in the `/var/www/html` directory.

This guide was taken from **tom's HARDWARE** website February 2025.

URL to full page:

<https://www.tomshardware.com/news/raspberry-pi-web-server,40174.html>

# License

MIT License

Copyright (c) 2024 UKRAA

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the **Software**), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED **AS IS**, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

---

## Contact us

Please send an e-mail to [picomuon@ukraa.com](mailto:picomuon@ukraa.com)

---

