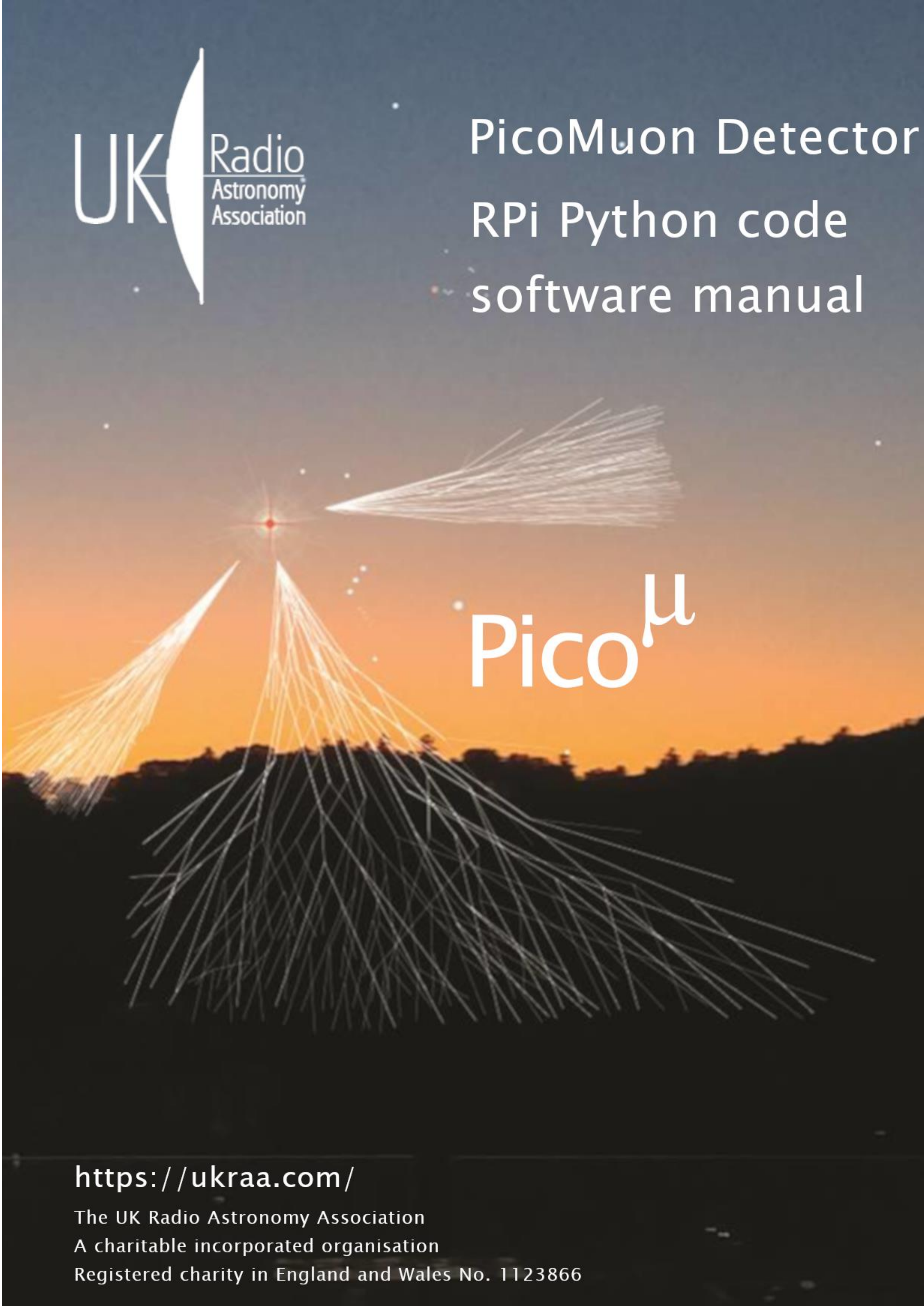
PicoMuon Detector RPi Python code software manual



https://ukraa.com/

The UK Radio Astronomy Association
A charitable incorporated organisation
Registered charity in England and Wales No. 1123866

UK Radio Astronomy Association

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

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.

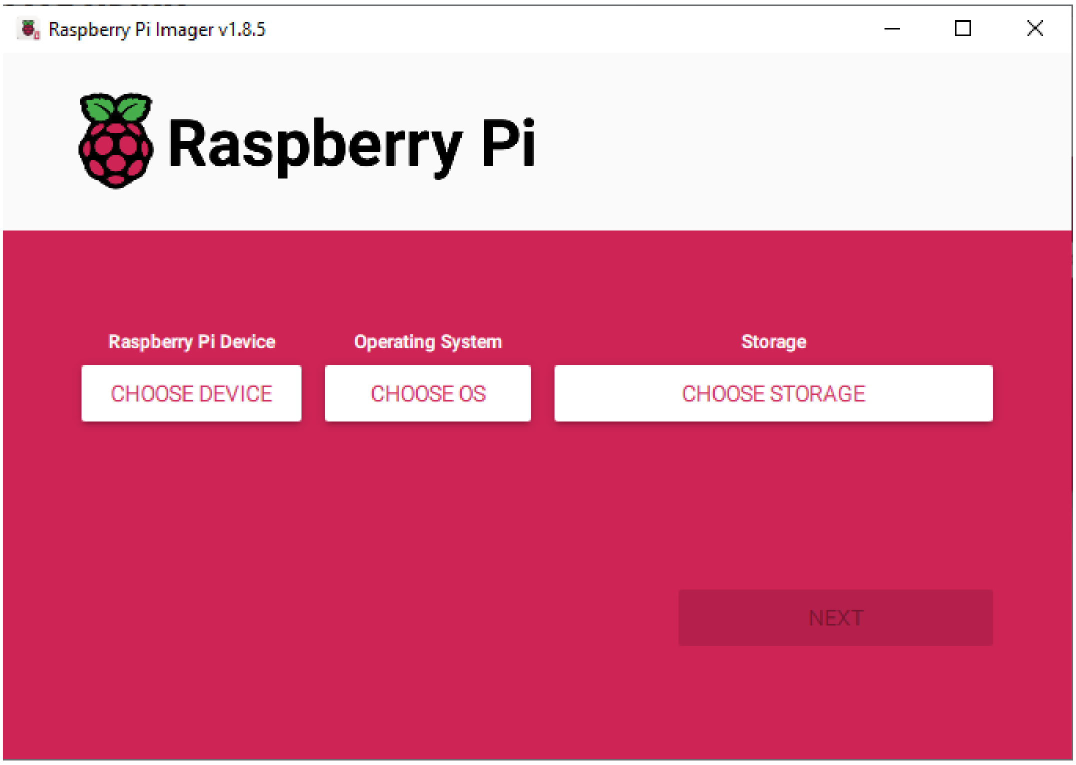
GetDataRawACM0.py is run as a service.

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

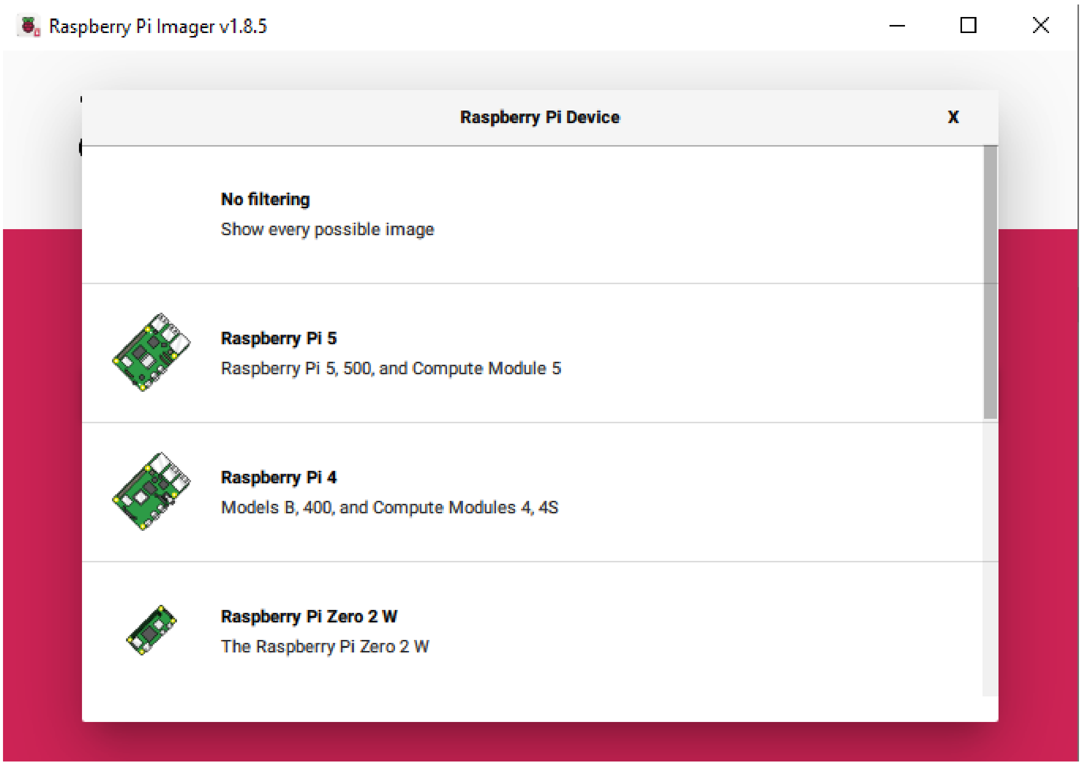
Set-up instructions

Raspberry Pi OS

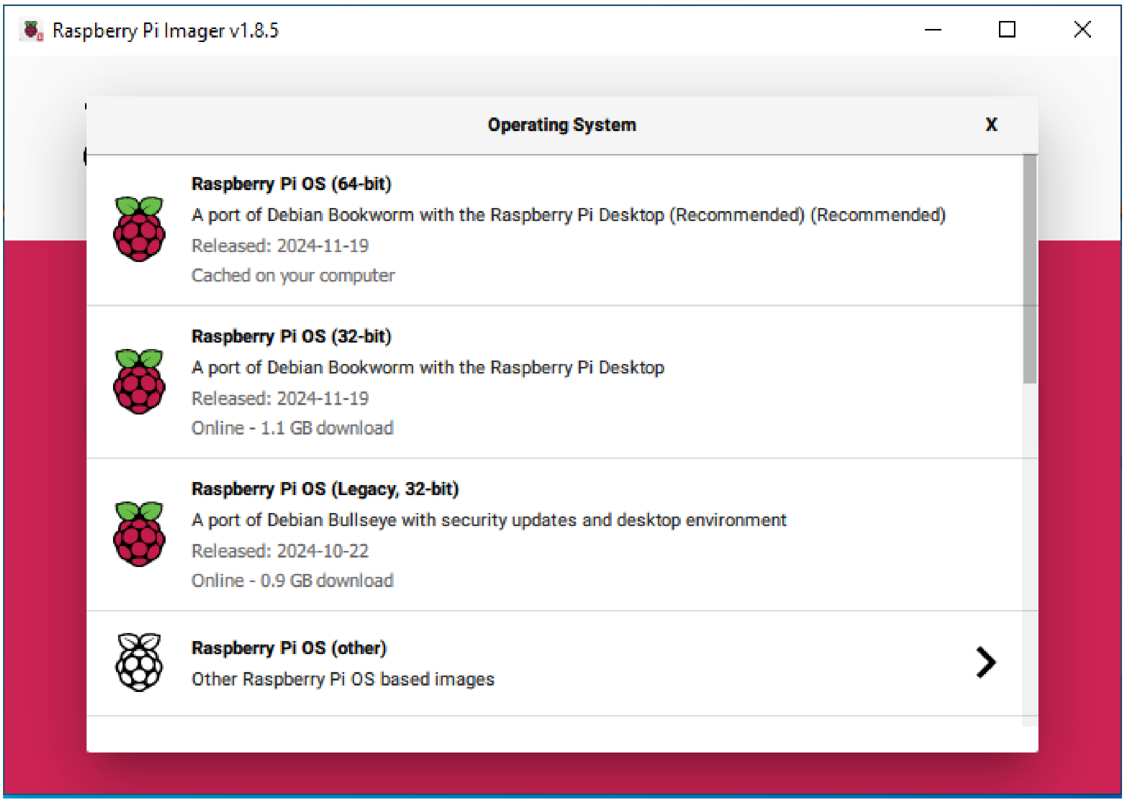
1. Download Raspberry Pi Imager for your operating system.



2. Run Raspberry Pi Imager.


Raspberry Pi Imager v1.8.5
Raspberry Pi Device
X
No filtering
Show every possible image
Raspberry Pi 5
Raspberry Pi 5, 500, and Compute Module 5
Raspberry Pi 4
Models B, 400, and Compute Modules 4, 4S
Raspberry Pi Zero 2 W
The Raspberry Pi Zero 2 W

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



Raspberry Pi Imager v1.8.5

Operating System

X

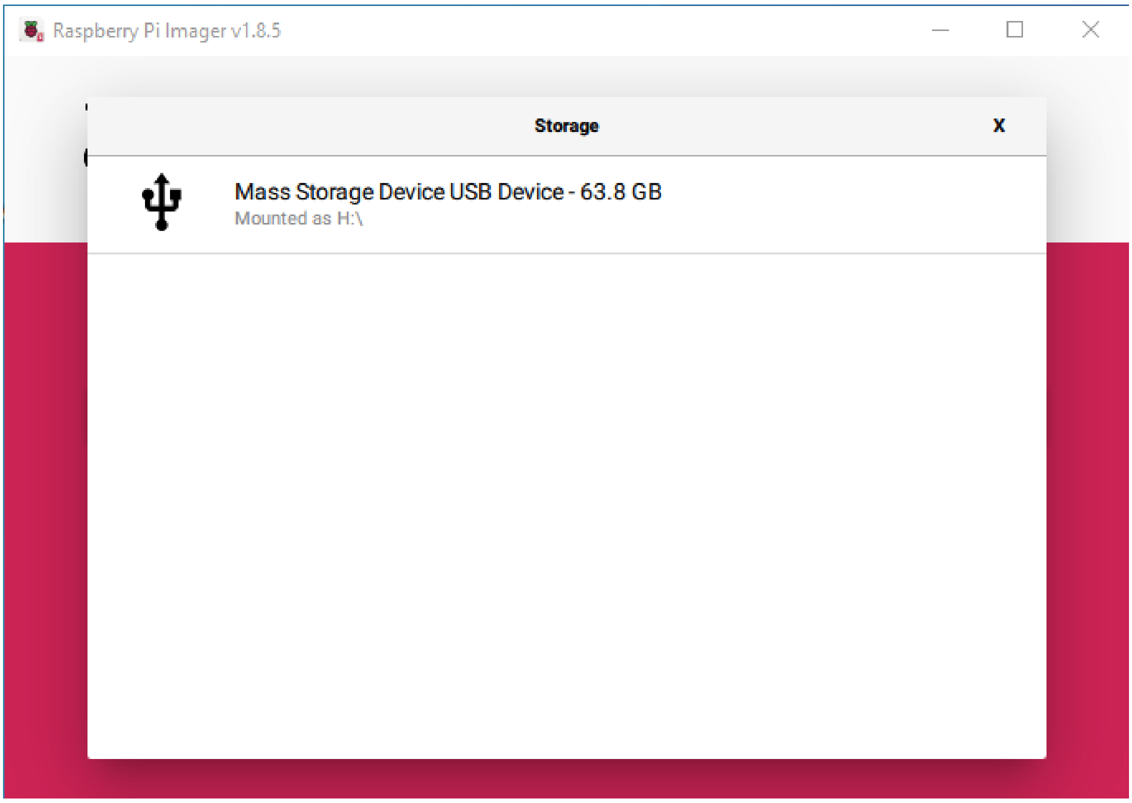
Raspberry Pi OS (64-bit)
A port of Debian Bookworm with the Raspberry Pi Desktop (Recommended) (Recommended)
Released: 2024-11-19
Cached on your computer

Raspberry Pi OS (32-bit)
A port of Debian Bookworm with the Raspberry Pi Desktop
Released: 2024-11-19
Online - 1.1 GB download

Raspberry Pi OS (Legacy, 32-bit)
A port of Debian Bullseye with security updates and desktop environment
Released: 2024-10-22
Online - 0.9 GB download

Raspberry Pi OS (other)
Other Raspberry Pi OS based images

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



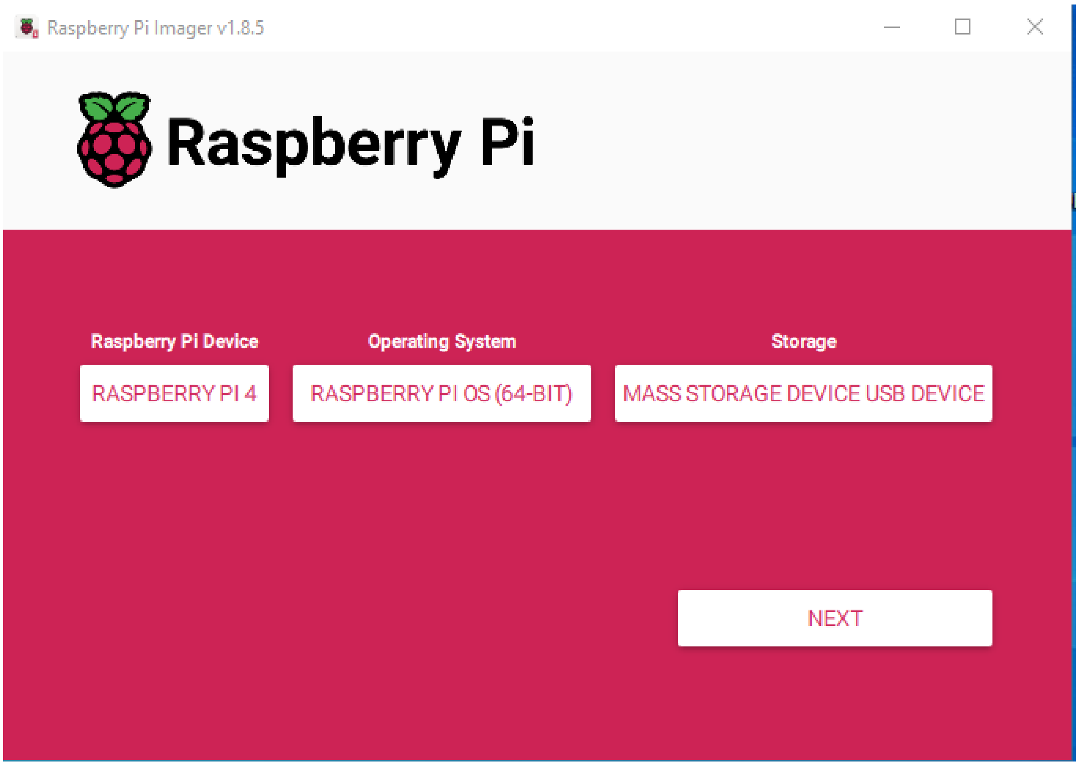
Raspberry Pi Imager v1.8.5

Storage

X

Mass Storage Device USB Device - 63.8 GB
Mounted as H:\

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



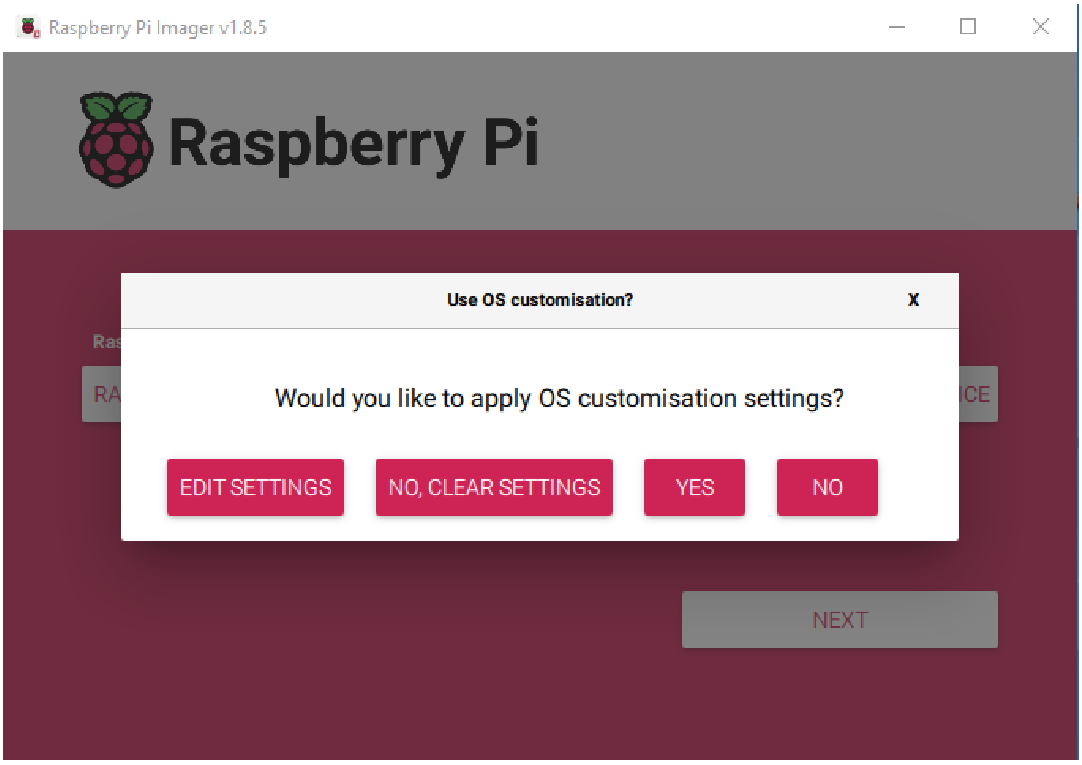
Raspberry Pi Device: RASPBERRY PI 4

Operating System: RASPBERRY PI OS (64-BIT)

Storage: MASS STORAGE DEVICE USB DEVICE

6. Select **NEXT**

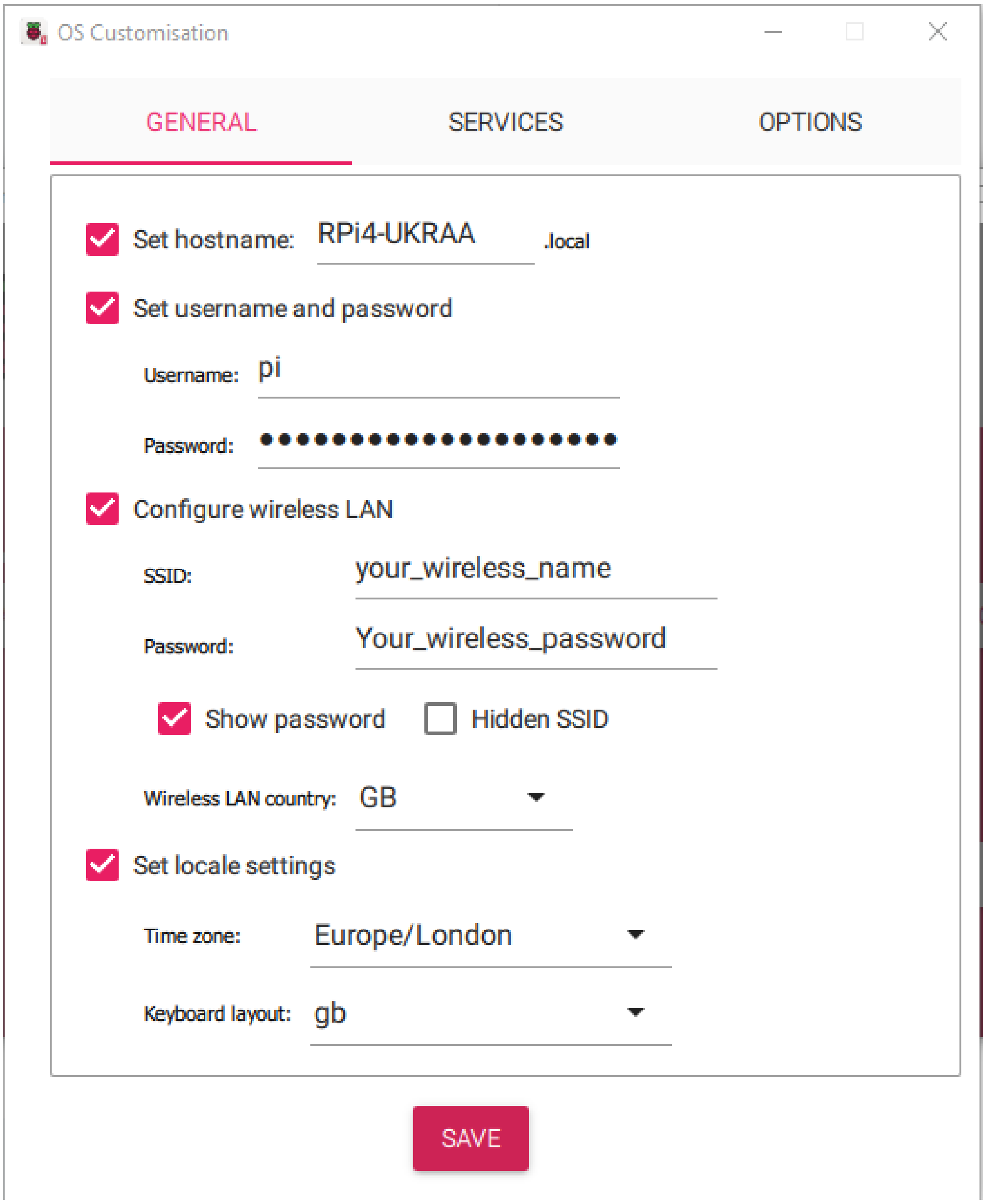
7. You are presented with a **Use OS customisation?** window, select **EDIT SETTINGS**.



Use OS customisation?

Would you like to apply OS customisation 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



☑ Set hostname: RPi4-UKRAA .local

☑ Set username and password

Username: pi

Password: ●●●●●●●●●●●●●●●●●●●●

☑ Configure wireless LAN

SSID: your_wireless_name

Password: Your_wireless_password

☑ Show password ☐ Hidden SSID

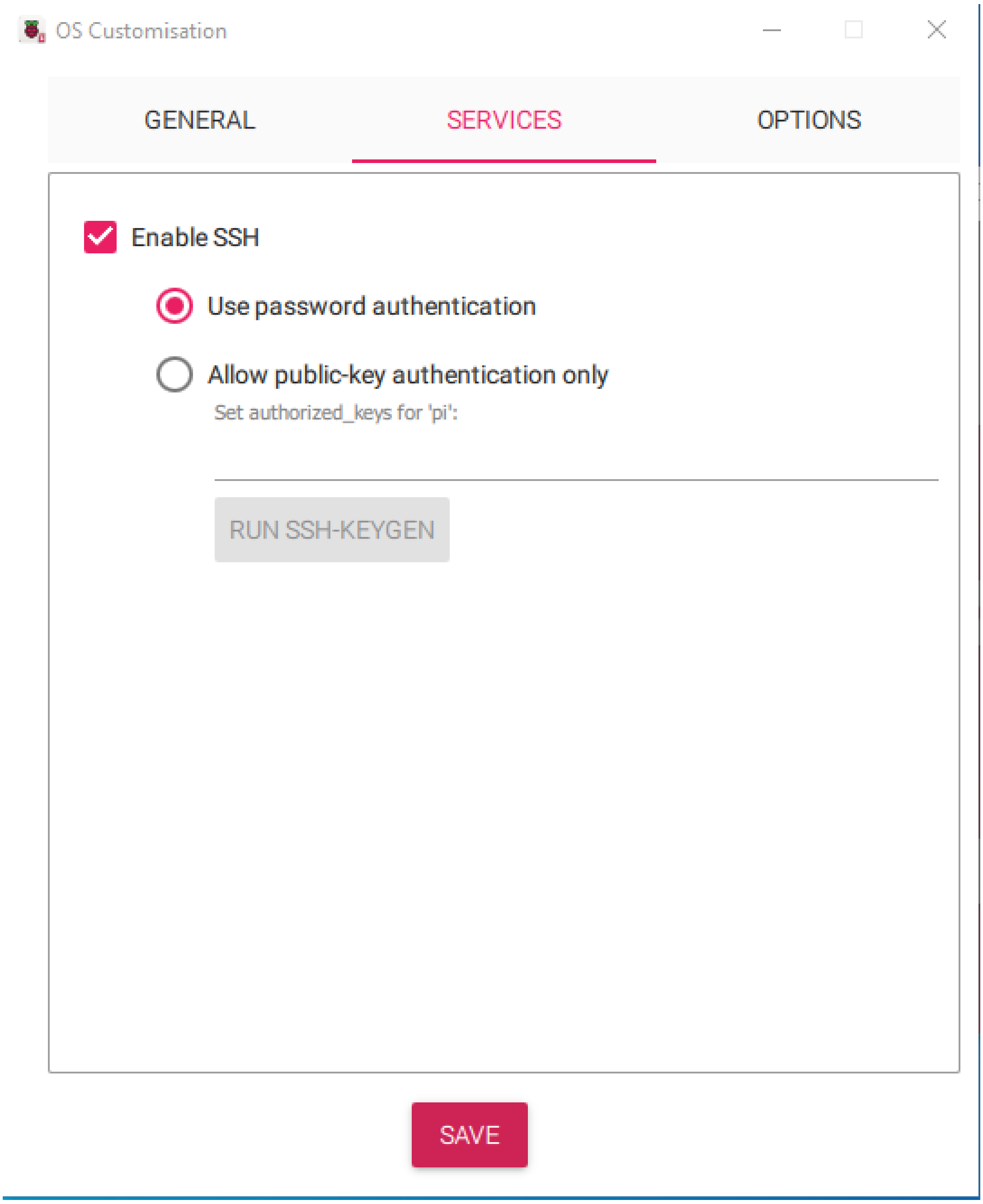
Wireless LAN country: GB

☑ Set locale settings

Time zone: Europe/London

Keyboard layout: gb

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



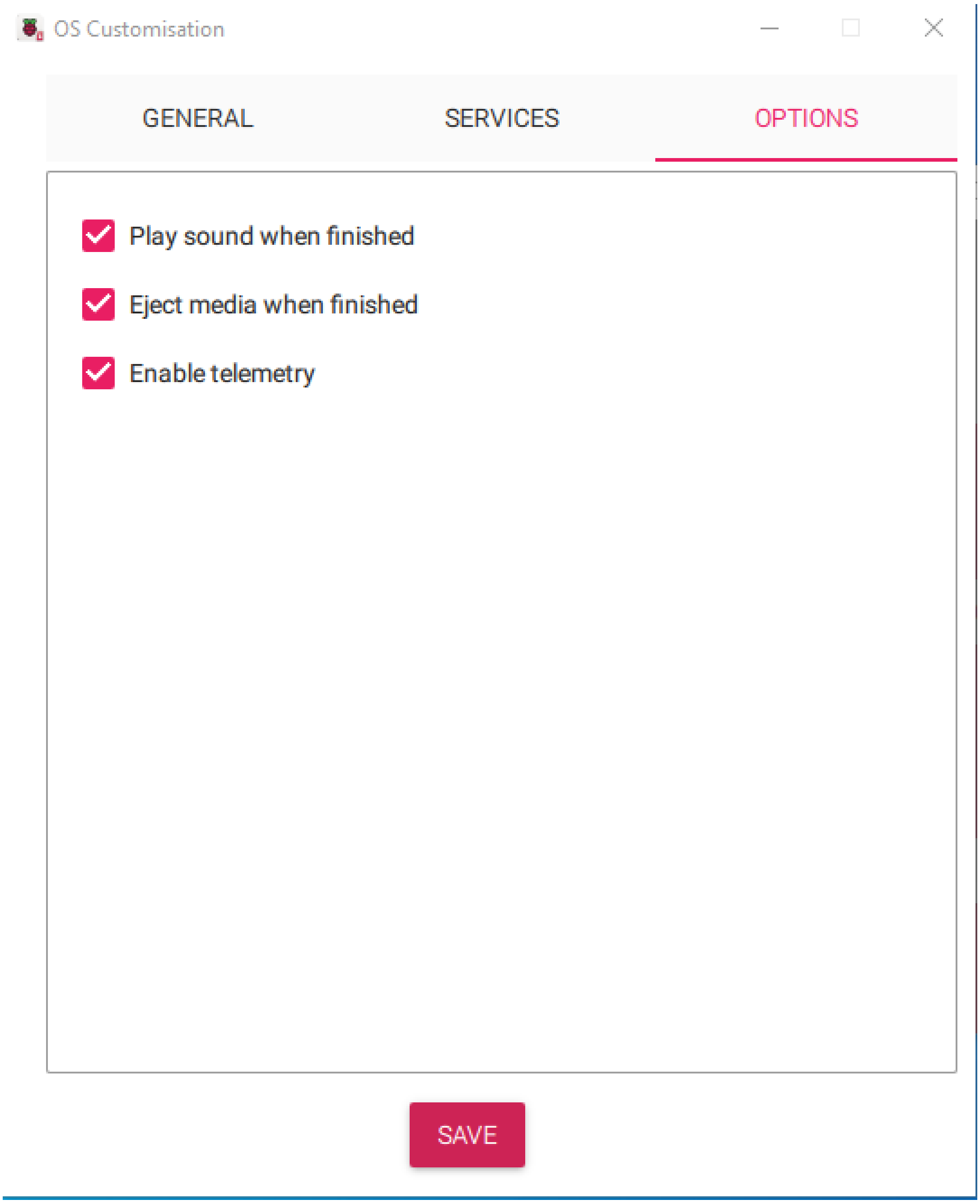
☑ Enable SSH

● Use password authentication

○ Allow public-key authentication only

Set authorized_keys for 'pi':

10. Select **OPTIONS**
    - Keep as shown



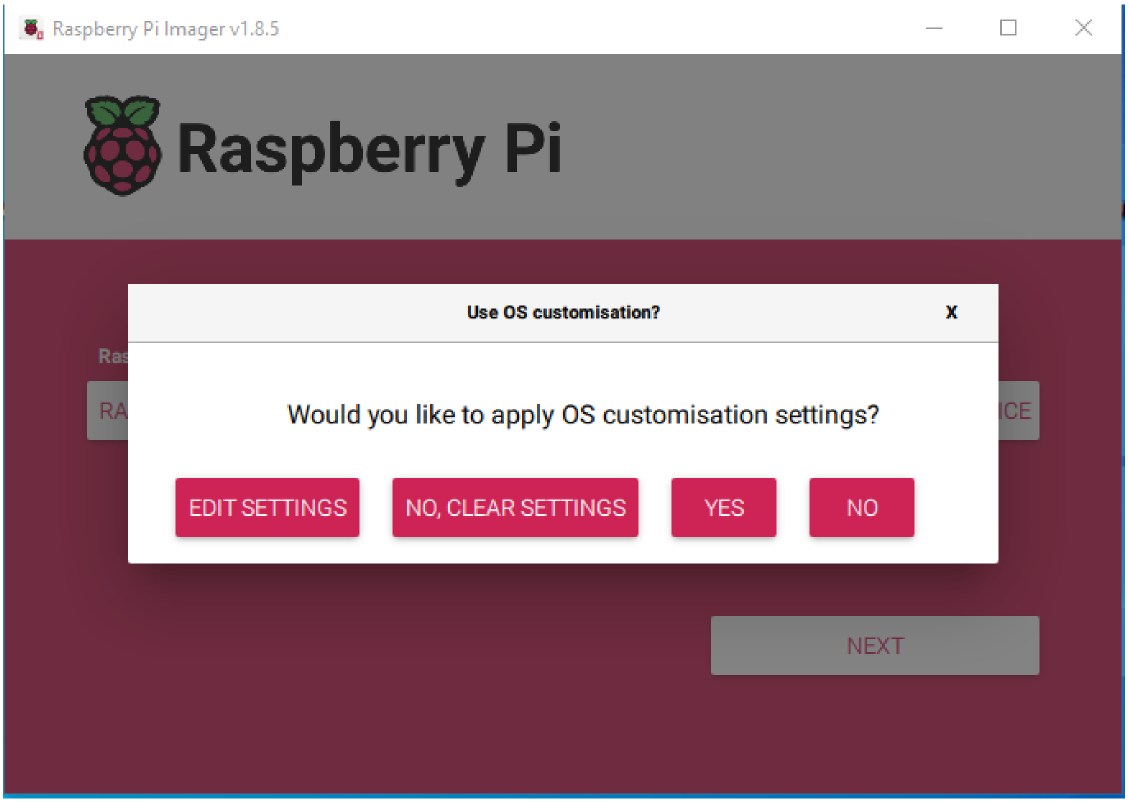
☑ Play sound when finished

☑ Eject media when finished

☑ Enable telemetry

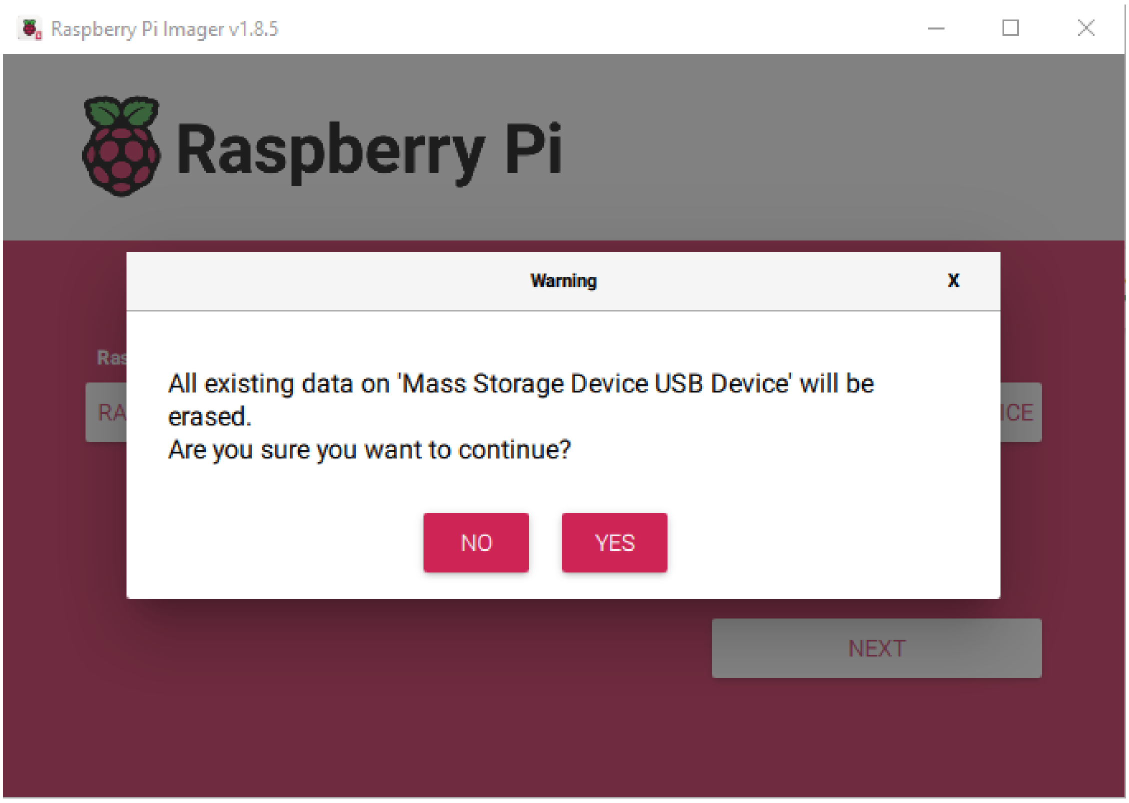
11. Select **SAVE** at bottom of window.

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



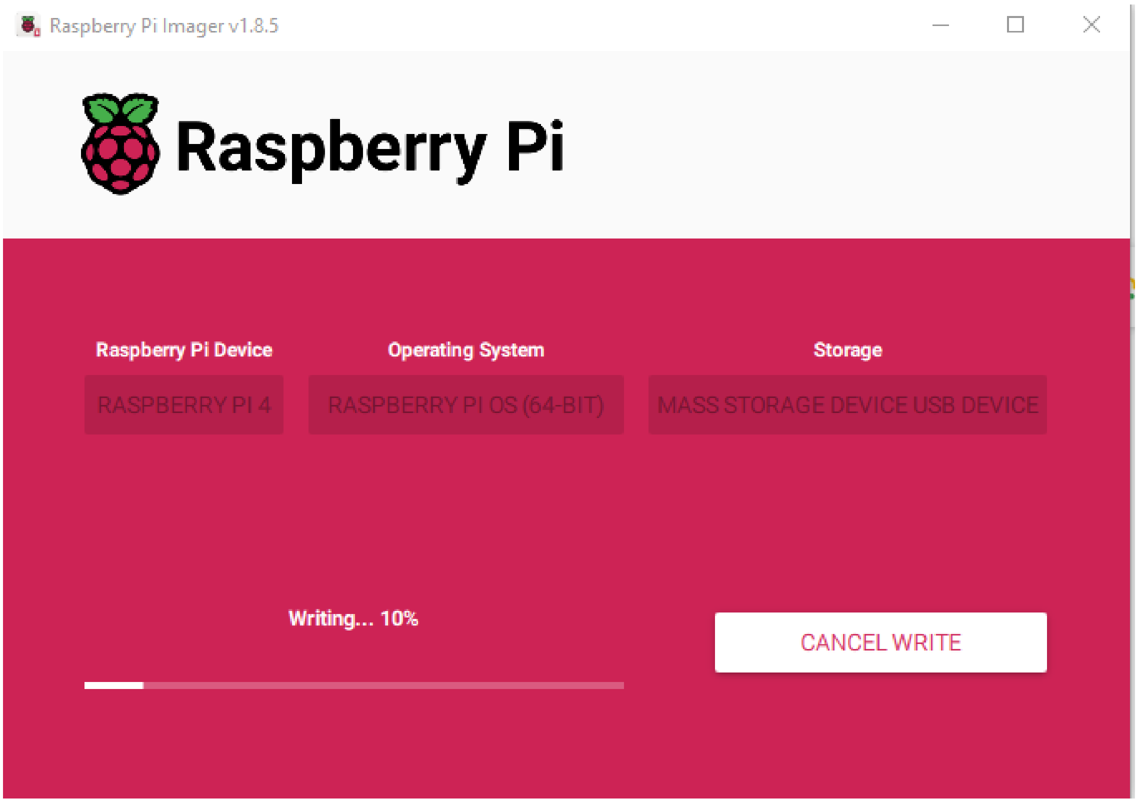
Would you like to apply OS customisation settings?

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



All existing data on 'Mass Storage Device USB Device' will be erased.
Are you sure you want to continue?

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


Raspberry Pi Imager v1.8.5

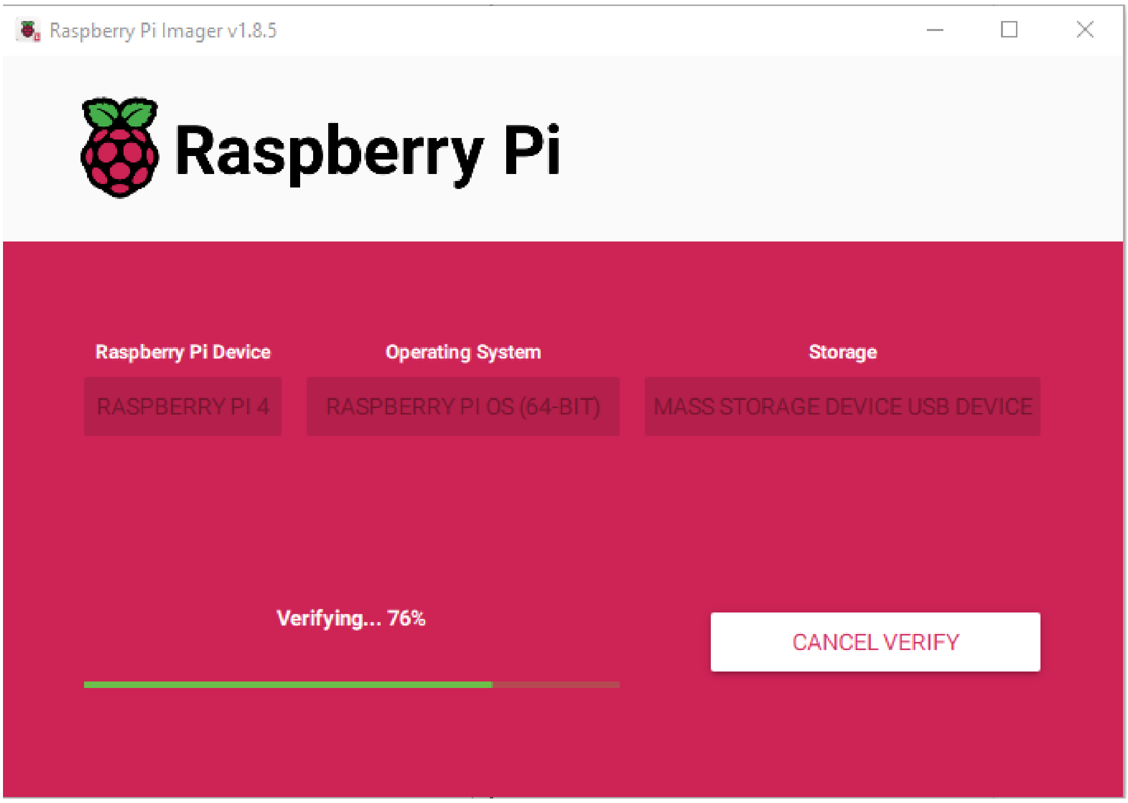
Raspberry Pi Device: RASPBERRY PI 4

Operating System: RASPBERRY PI OS (64-BIT)

Storage: MASS STORAGE DEVICE USB DEVICE

Writing... 10%

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


Raspberry Pi Imager v1.8.5

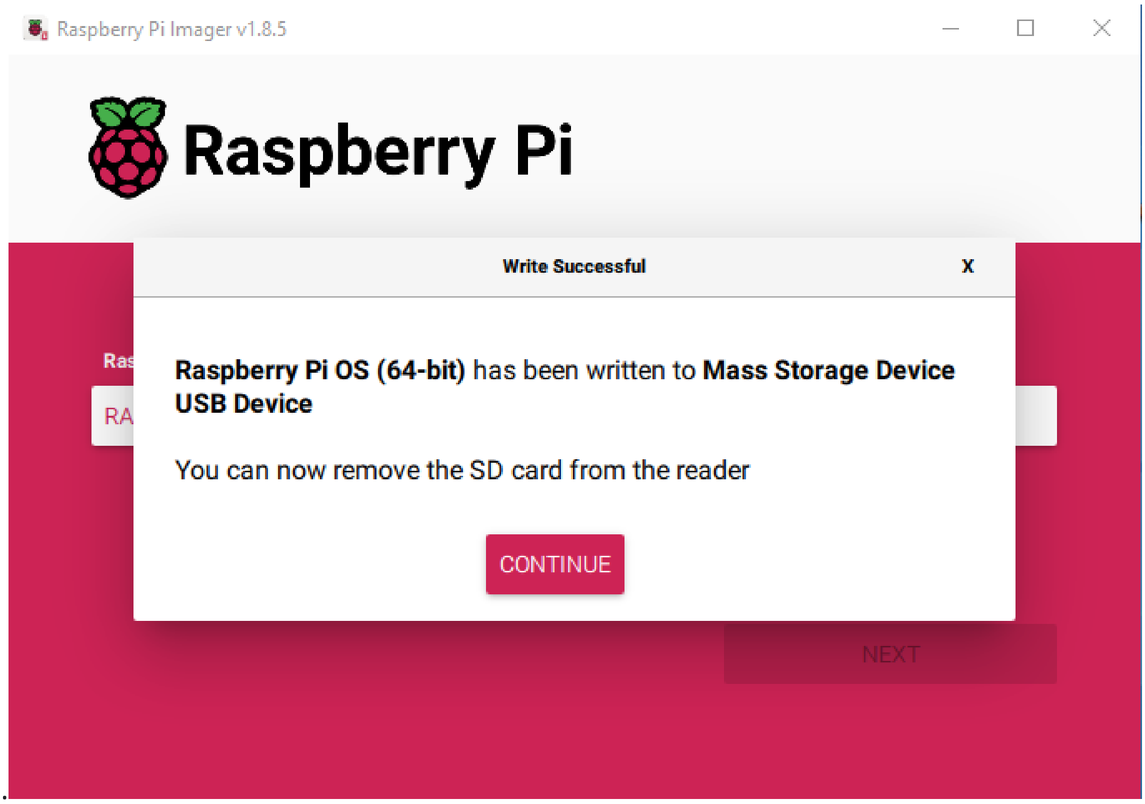
Raspberry Pi Device: RASPBERRY PI 4

Operating System: RASPBERRY PI OS (64-BIT)

Storage: MASS STORAGE DEVICE USB DEVICE

Verifying... 76%

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



Raspberry Pi OS (64-bit) has been written to **Mass Storage Device USB Device**

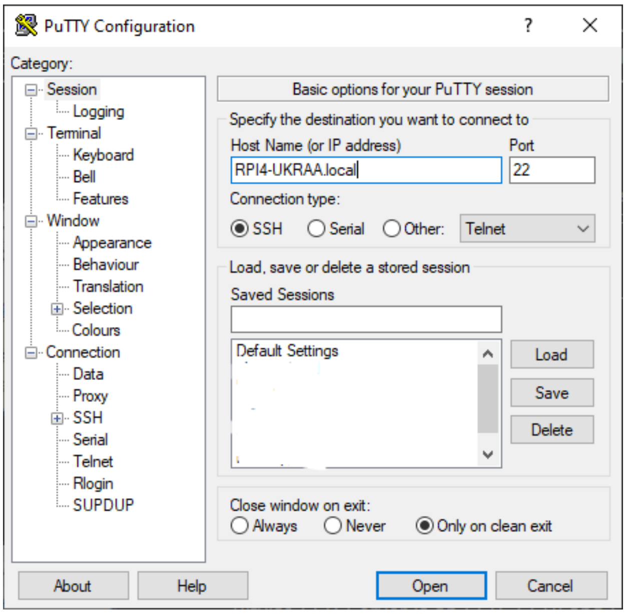
You can now remove the SD card from the reader

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**



PuTTY Configuration

Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address): RPI4-UKRAA.local

Port: 22

Connection type: ● SSH ○ Serial ○ Other: Telnet

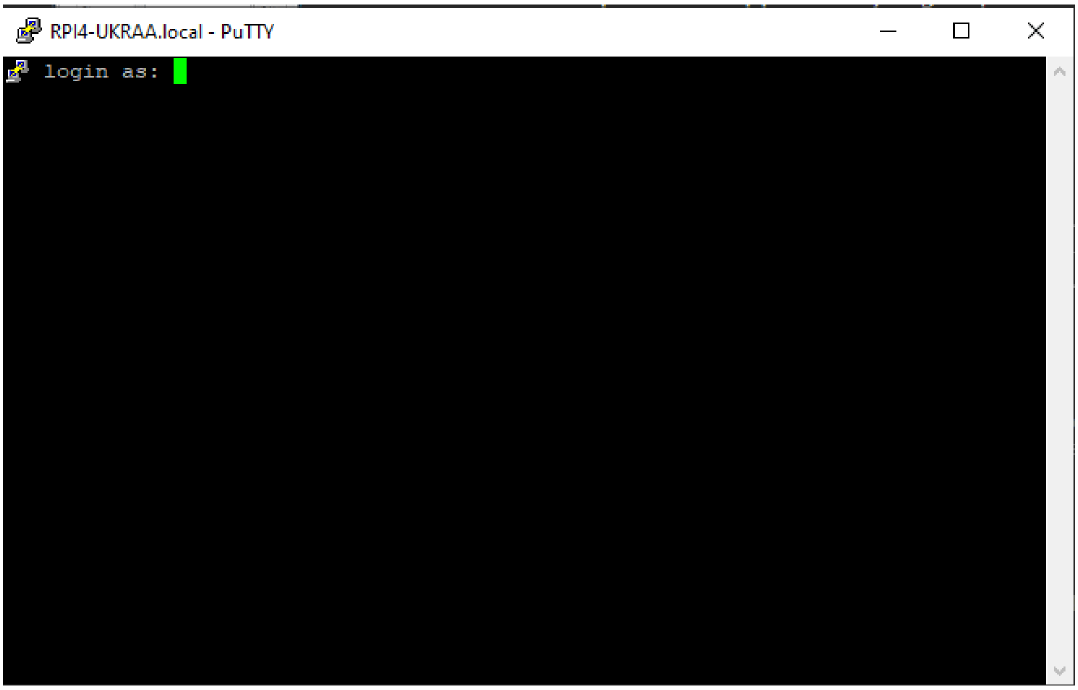
Load, save or delete a stored session

Saved Sessions

Default Settings

Close window on exit: ○ Always ○ Never ● Only on clean exit

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



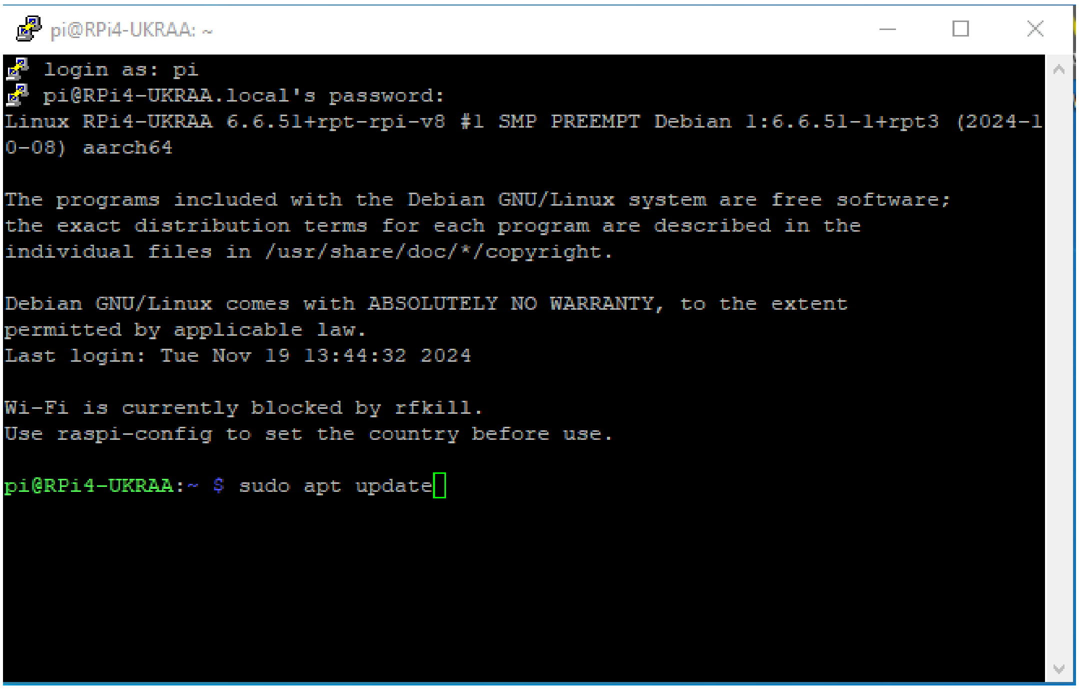
RPI4-UKRAA.local - PuTTY

```
login as: 
```

3. Login as **pi** and enter your **password**

4. Now type the command below and press enter. This will update the RPi OS

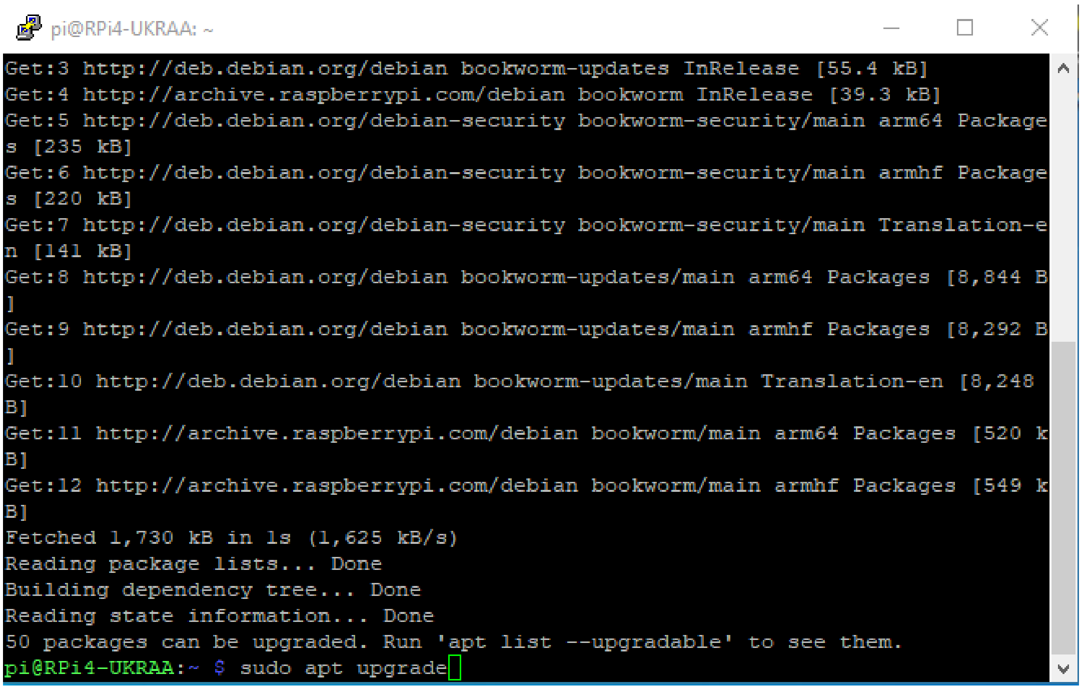
sudo apt update



```
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 (2024-1
0-08) 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.

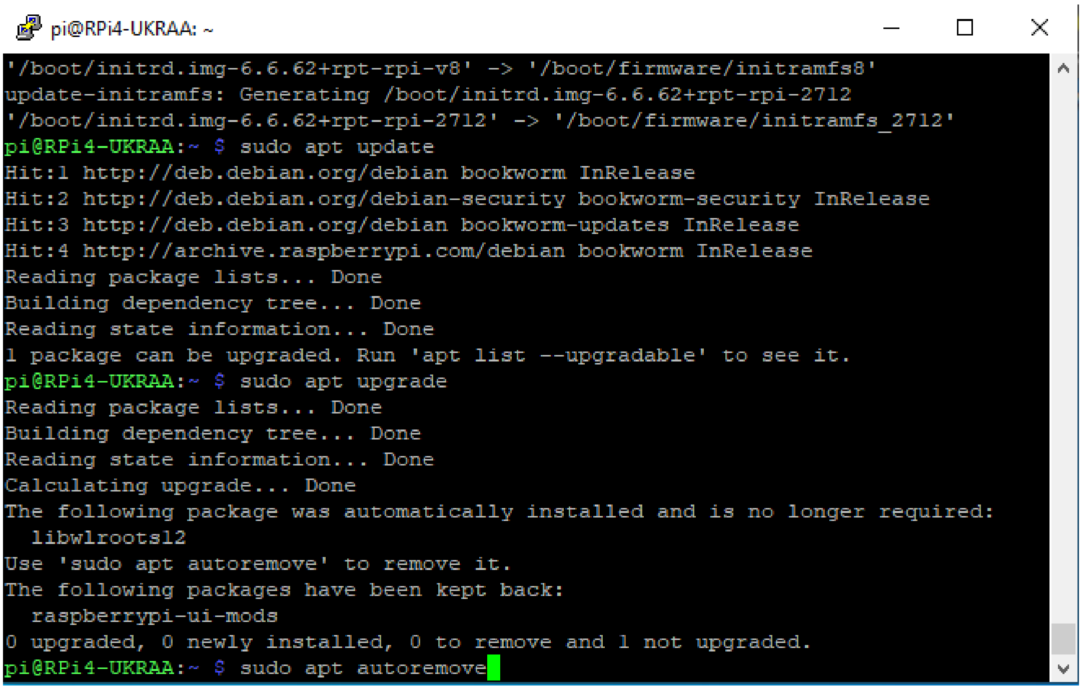
sudo apt upgrade



```
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 Package
s [235 kB]
Get:6 http://deb.debian.org/debian-security bookworm-security/main armhf Package
s [220 kB]
Get:7 http://deb.debian.org/debian-security bookworm-security/main Translation-e
n [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 k
B]
Get:12 http://archive.raspberrypi.com/debian bookworm/main armhf Packages [549 k
B]
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

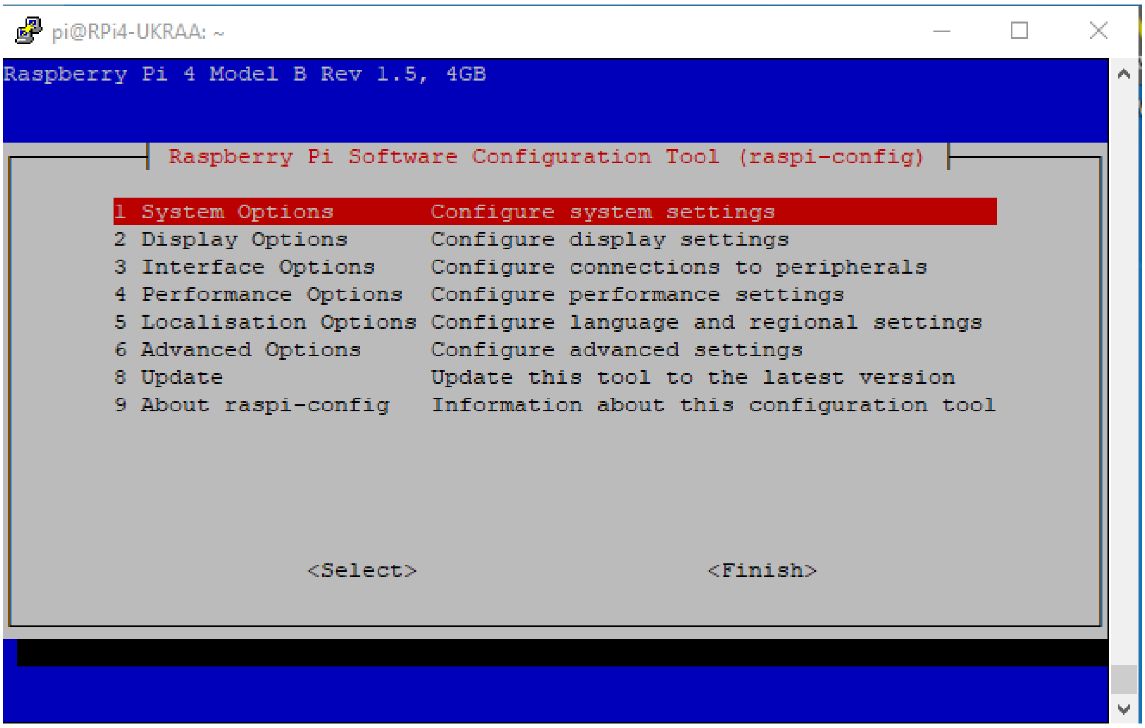


```
'/boot/initrd.img-6.6.62+rpt-rpi-v8' -> '/boot/firmware/initramfs8'
update-initramfs: Generating /boot/initrd.img-6.6.62+rpt-rpi-2712
'/boot/initrd.img-6.6.62+rpt-rpi-2712' -> '/boot/firmware/initramfs_2712'
pi@RPi4-UKRAA:~ $ sudo apt update
Hit:1 http://deb.debian.org/debian bookworm InRelease
Hit:2 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

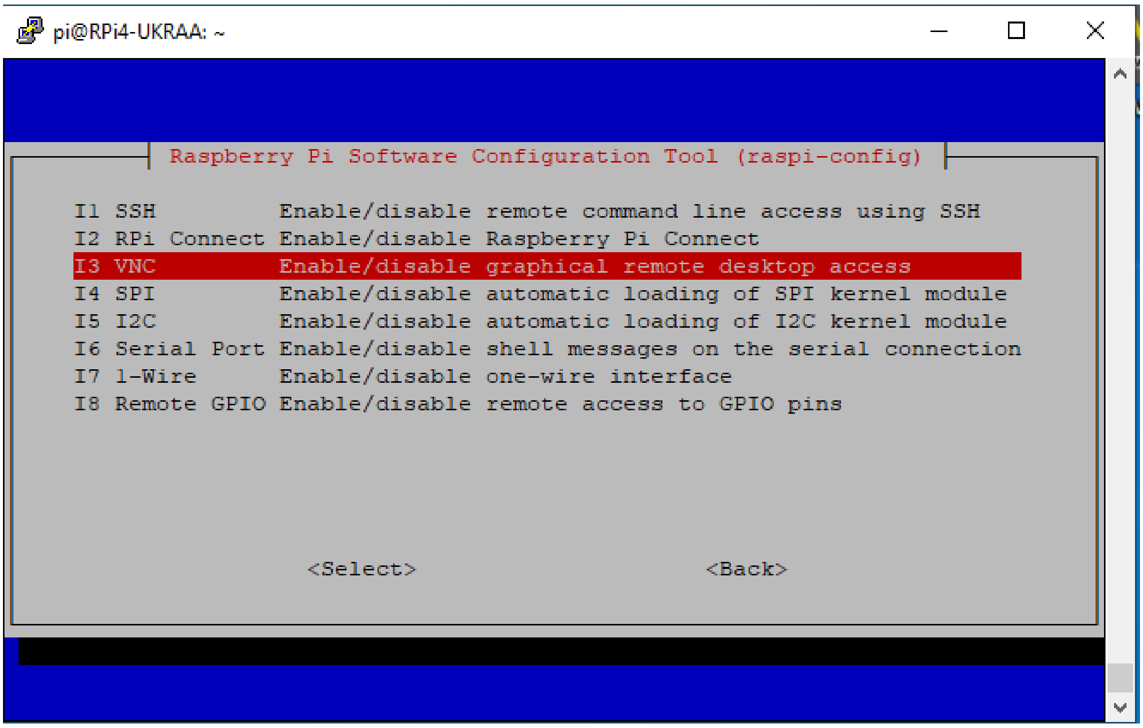
8. Select **3 Interface Options** and selecting **Select** and press enter.



pi@RPi4-UKRAA: ~

```
Raspberry Pi 4 Model B Rev 1.5, 4GB

         Raspberry Pi Software Configuration Tool (raspi-config)

       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
       9 About raspi-config   Information about this configuration tool

                     <Select>                     <Finish>
```

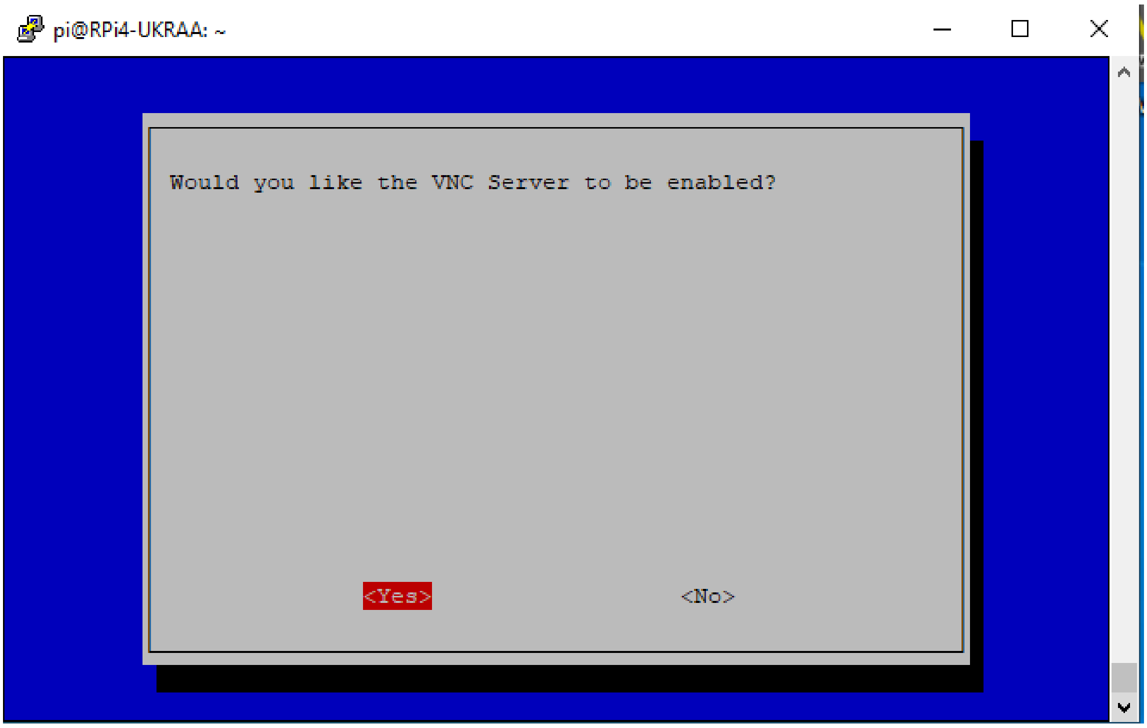
9. Now navigate down to **I3 VNC** and selecting **Select** and press enter.



pi@RPi4-UKRAA: ~

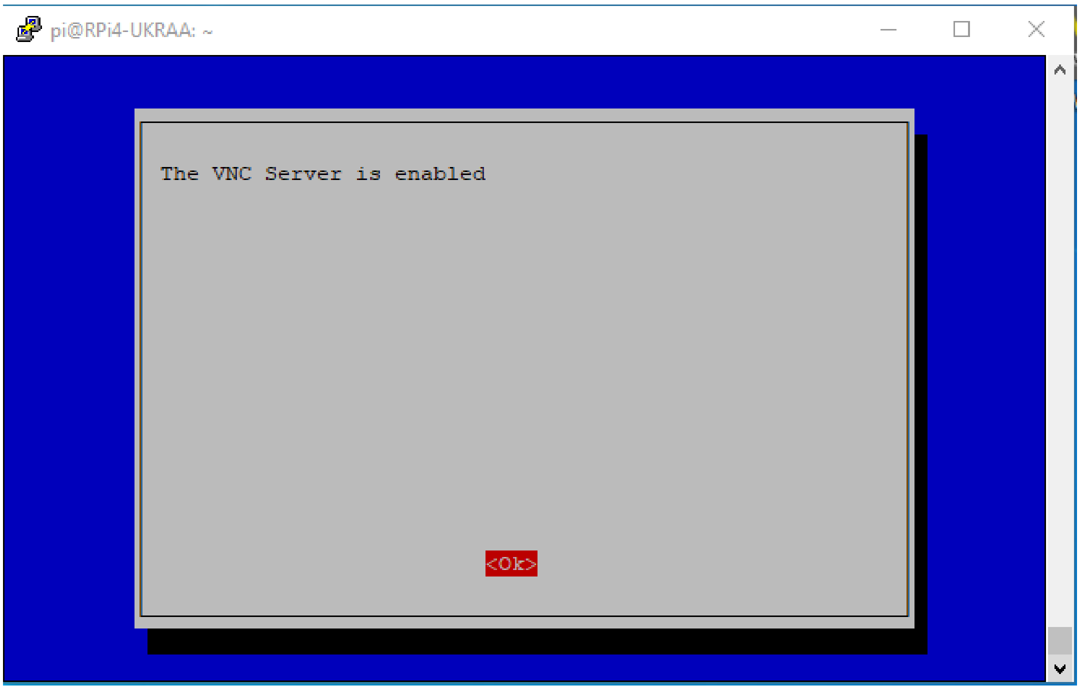
```
         Raspberry Pi Software Configuration Tool (raspi-config)

    I1 SSH         Enable/disable remote command line access using SSH
    I2 RPi Connect Enable/disable Raspberry Pi Connect
    I3 VNC         Enable/disable graphical remote desktop access
    I4 SPI         Enable/disable automatic loading of SPI kernel module
    I5 I2C         Enable/disable automatic loading of I2C kernel module
    I6 Serial Port Enable/disable shell messages on the serial connection
    I7 1-Wire      Enable/disable one-wire interface
    I8 Remote GPIO Enable/disable remote access to GPIO pins

                     <Select>                     <Back>
```

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



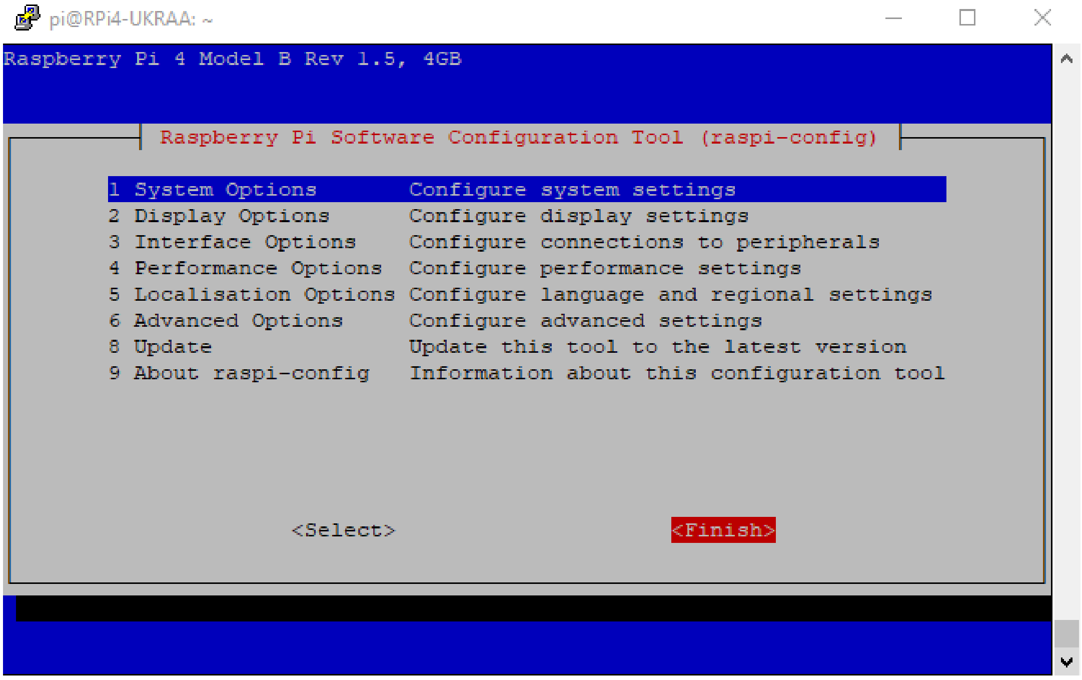
```
Would you like the VNC Server to be enabled?

<Yes>                    <No>
```

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



```
The VNC Server is enabled

<Ok>
```

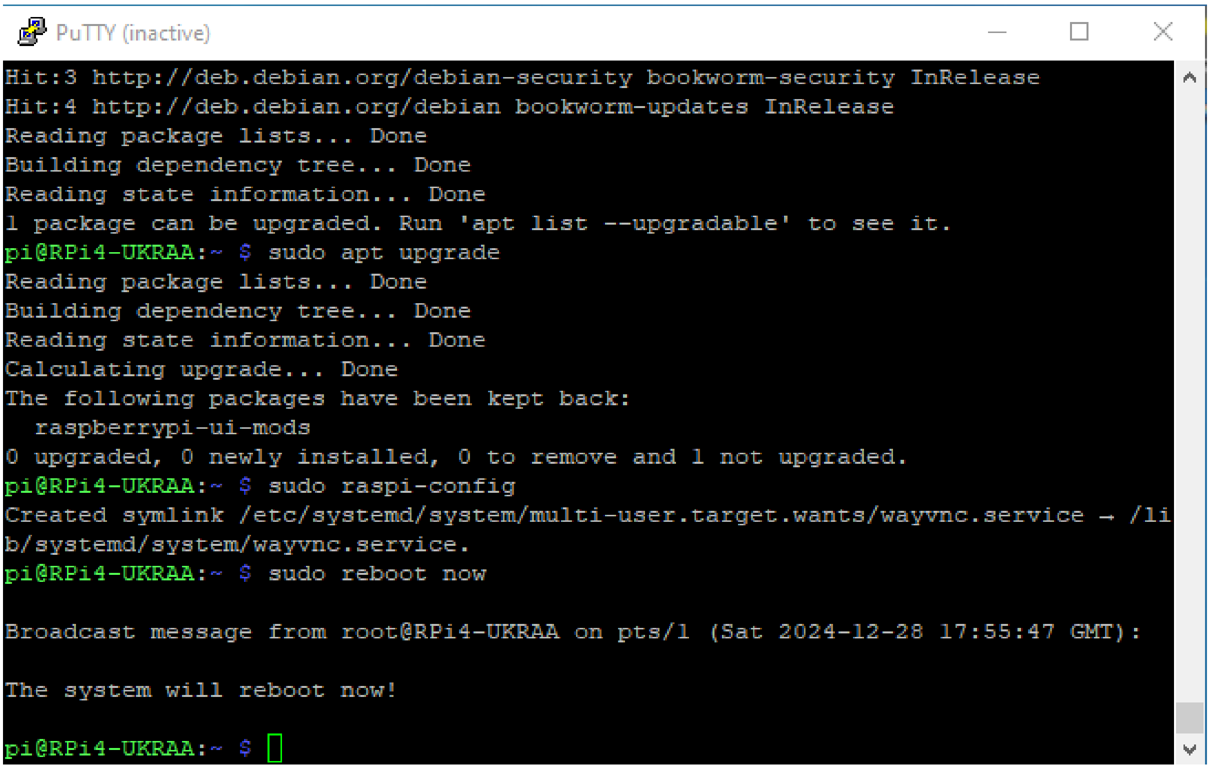
12. Finish with the Configuration Tool by selecting **Finish** and press enter.



```
Raspberry Pi 4 Model B Rev 1.5, 4GB

        Raspberry Pi Software Configuration Tool (raspi-config)

        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
        9 About raspi-config    Information about this configuration tool

                     <Select>                     <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

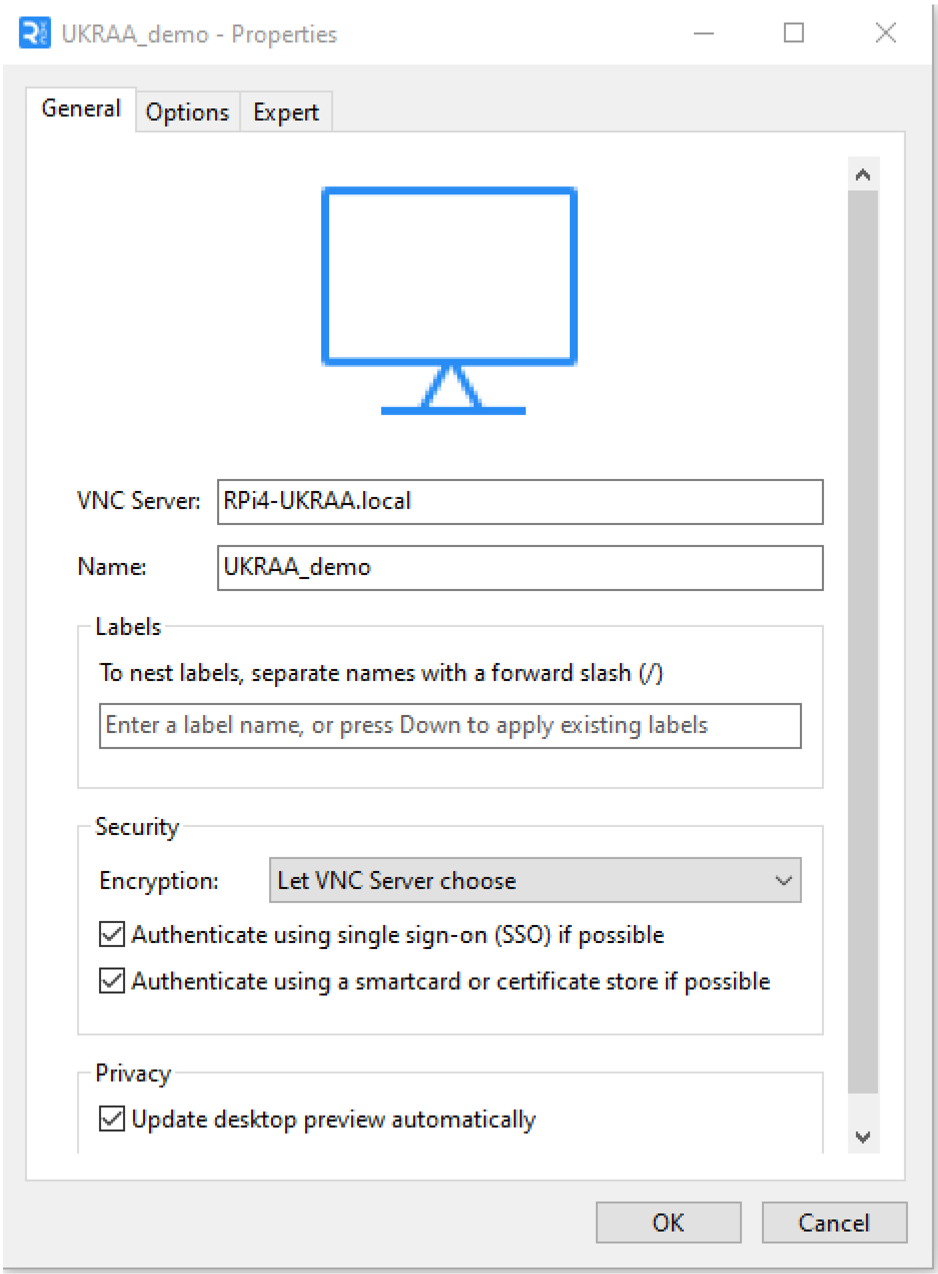


```
Hit:3 http://deb.debian.org/debian-security bookworm-security InRelease
Hit:4 http://deb.debian.org/debian bookworm-updates InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
1 package can be upgraded. Run 'apt list --upgradable' to see it.
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 → /li
b/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!

pi@RPi4-UKRAA:~ $
```

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**.



UKRAA_demo - Properties

VNC Server: RPi4-UKRAA.local

Name: UKRAA_demo

Labels

To nest labels, separate names with a forward slash (/)

Enter a label name, or press Down to apply existing labels

Security

Encryption: Let VNC Server choose

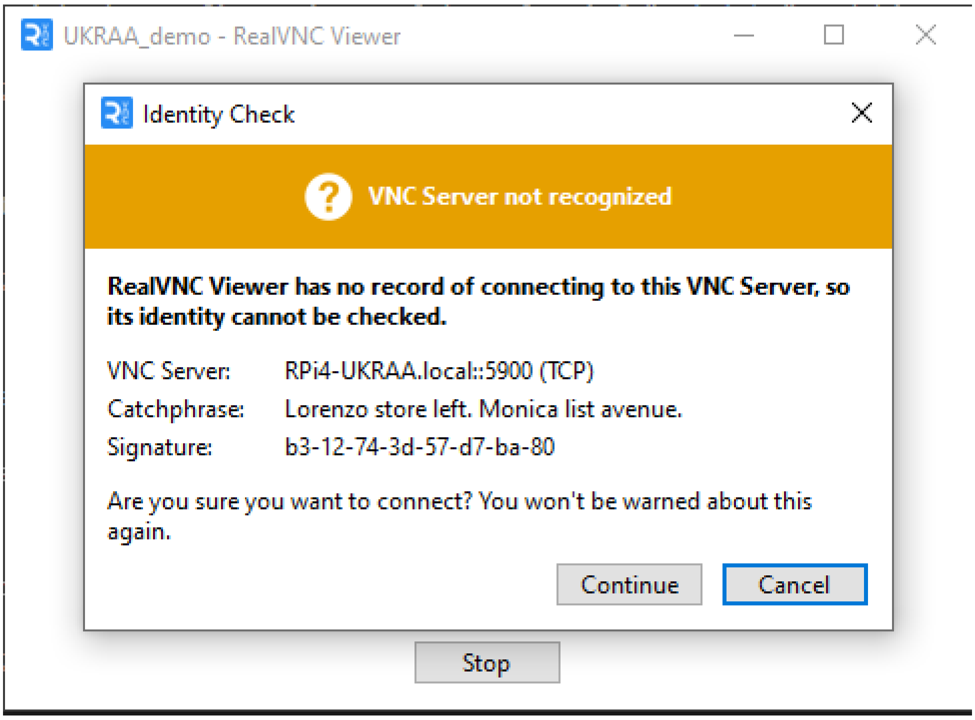
☑ Authenticate using single sign-on (SSO) if possible

☑ Authenticate using a smartcard or certificate store if possible

Privacy

☑ Update desktop preview automatically

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



UKRAA_demo - RealVNC Viewer

Identity Check

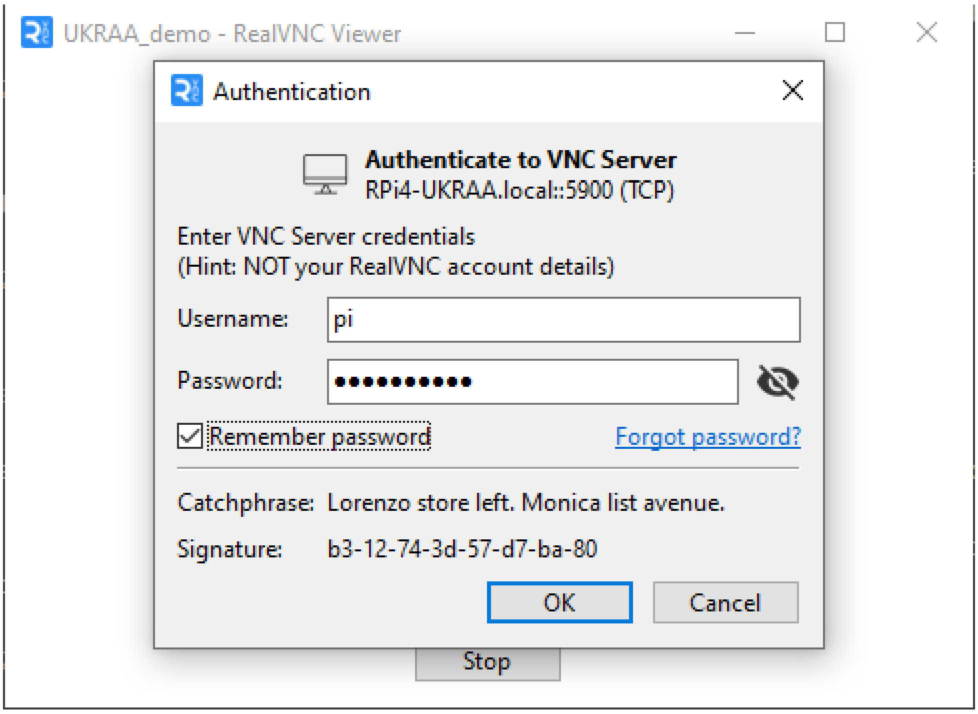
VNC Server not recognized

RealVNC Viewer has no record of connecting to this VNC Server, so its identity cannot be checked.

VNC Server: RPi4-UKRAA.local::5900 (TCP)
Catchphrase: Lorenzo store left. Monica list avenue.
Signature: b3-12-74-3d-57-d7-ba-80

Are you sure you want to connect? You won't be warned about this again.

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.



UKRAA_demo - RealVNC Viewer

Authentication

Authenticate to VNC Server
RPi4-UKRAA.local::5900 (TCP)

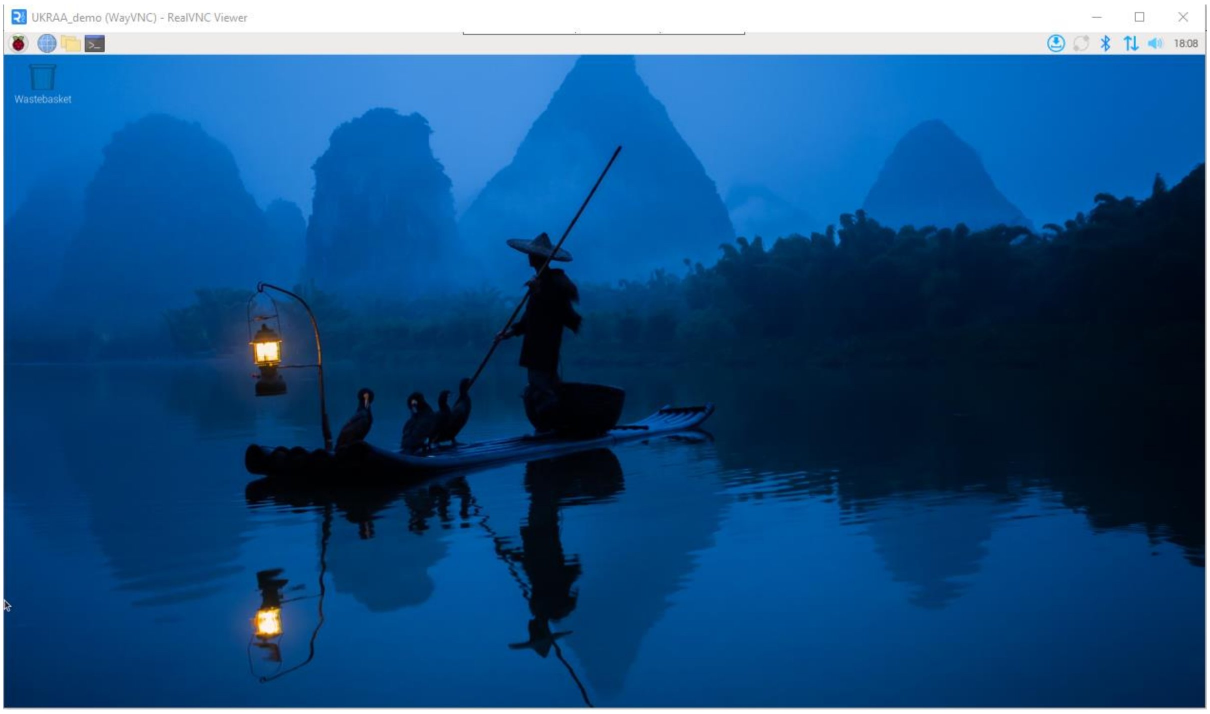
Enter VNC Server credentials
(Hint: NOT your RealVNC account details)

Username: pi
Password: ●●●●●●●●●●●

☑ Remember password

Catchphrase: Lorenzo store left. Monica list avenue.
Signature: b3-12-74-3d-57-d7-ba-80

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



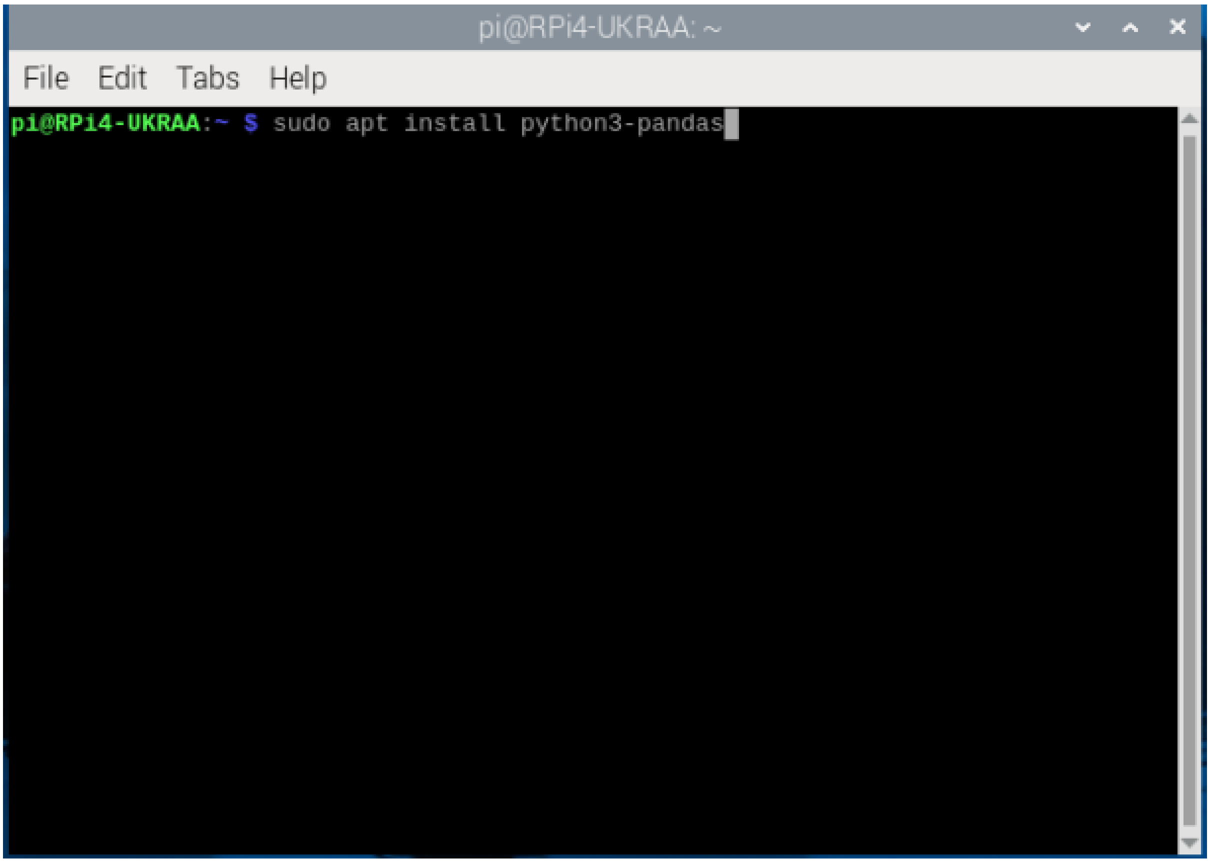
Add additional software to your RPi

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.

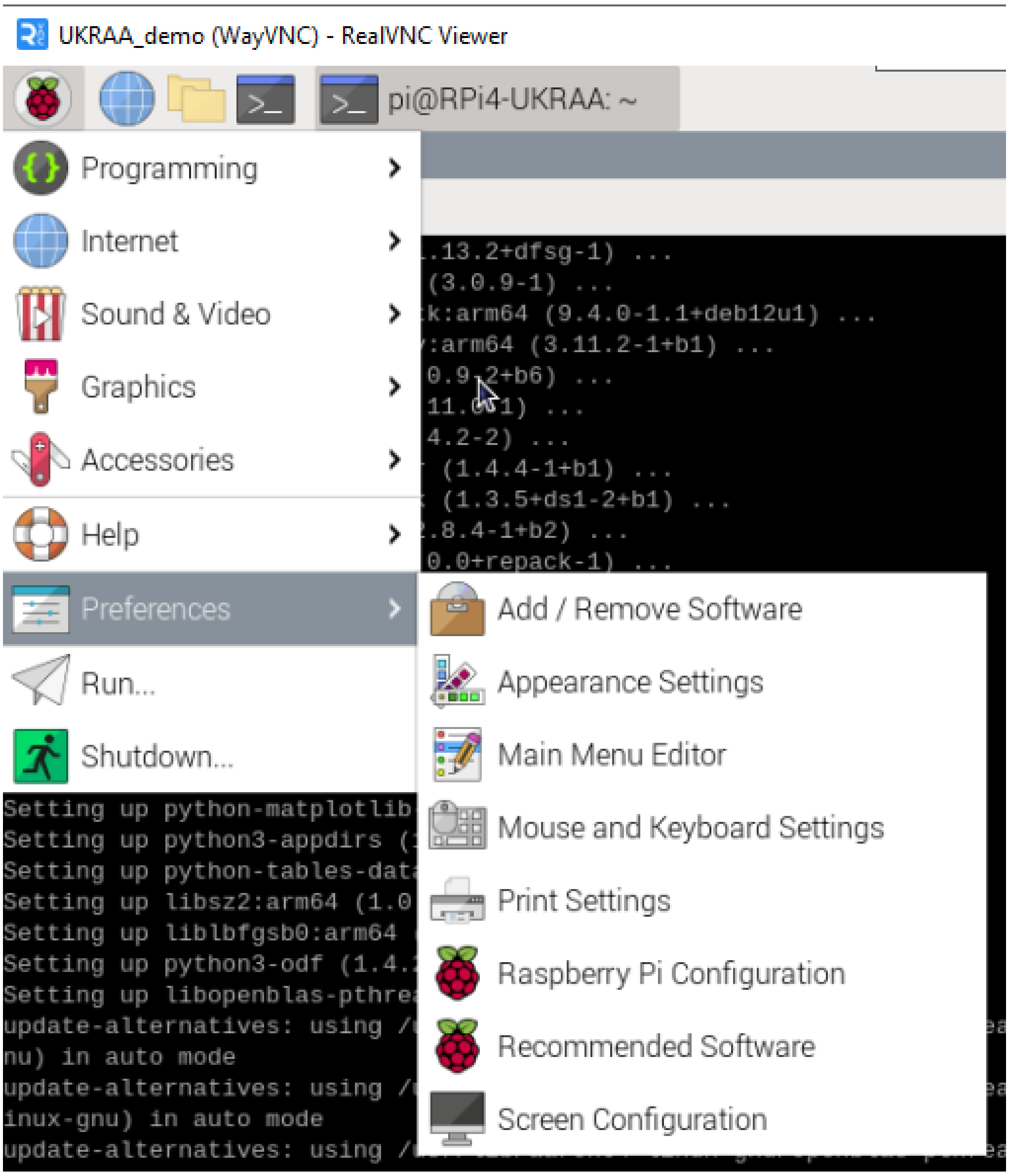
sudo apt install python3-pandas



```
pi@RPi4-UKRAA:~ $ sudo apt install python3-pandas
```

gnuplot

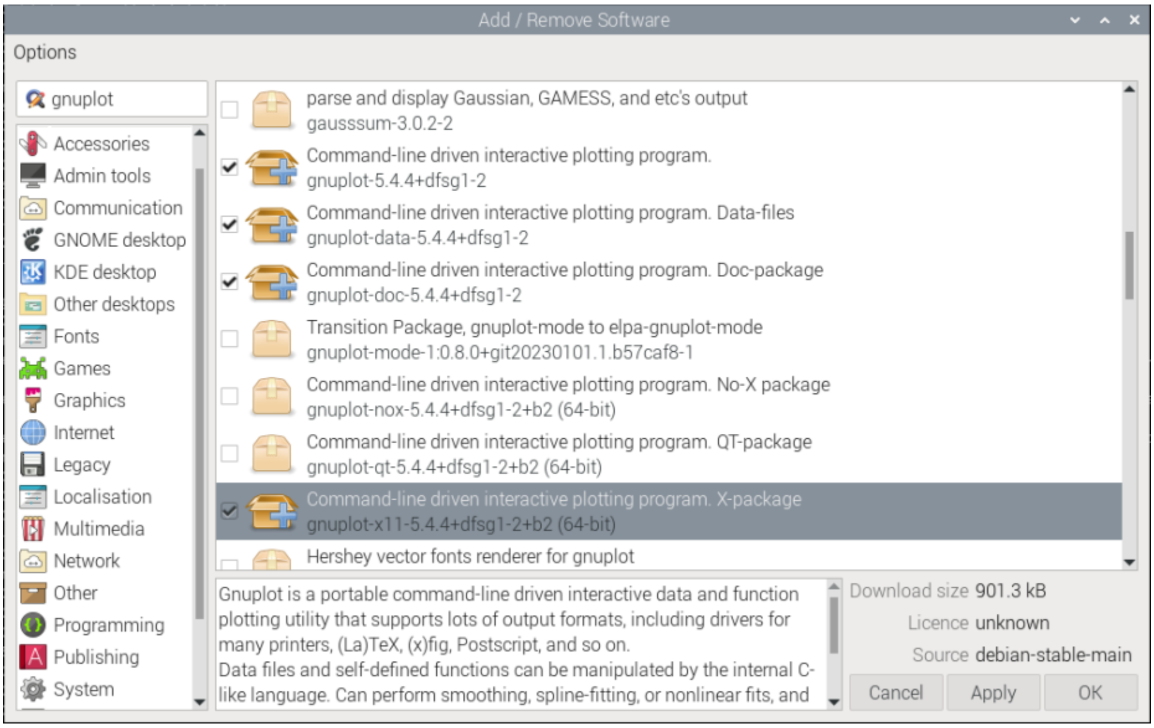
1. Open the application menu (Raspberry logo) and select **Preferences** and **Add/Remove Software**.



```
.13.2+dfsg-1) ...
(3.0.9-1) ...
k:arm64 (9.4.0-1.1+deb12u1) ...
:arm64 (3.11.2-1+b1) ...
0.9.2+b6) ...
11.0-1) ...
4.2-2) ...
 (1.4.4-1+b1) ...
 (1.3.5+ds1-2+b1) ...
.8.4-1+b2) ...
0.0+repack-1) ...
```

```
Setting up python-matplotlib
Setting up python3-appdirs (
Setting up python-tables-data
Setting up libsz2:arm64 (1.0
Setting up liblbfgsb0:arm64
Setting up python3-odf (1.4.
Setting up libopenblas-pthre
update-alternatives: using /
nu) in auto mode
update-alternatives: using /
inux-gnu) in auto mode
update-alternatives: using /
```

2. 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**.



☐ parse and display Gaussian, GAMESS, and etc's output
gausssum-3.0.2-2

☑ Command-line driven interactive plotting program.
gnuplot-5.4.4+dfsg1-2

☑ Command-line driven interactive plotting program. Data-files
gnuplot-data-5.4.4+dfsg1-2

☑ Command-line driven interactive plotting program. Doc-package
gnuplot-doc-5.4.4+dfsg1-2

☐ Transition Package, gnuplot-mode to elpa-gnuplot-mode
gnuplot-mode-1:0.8.0+git20230101.1.b57caf8-1

☐ Command-line driven interactive plotting program. No-X package
gnuplot-nox-5.4.4+dfsg1-2+b2 (64-bit)

☐ Command-line driven interactive plotting program. QT-package
gnuplot-qt-5.4.4+dfsg1-2+b2 (64-bit)

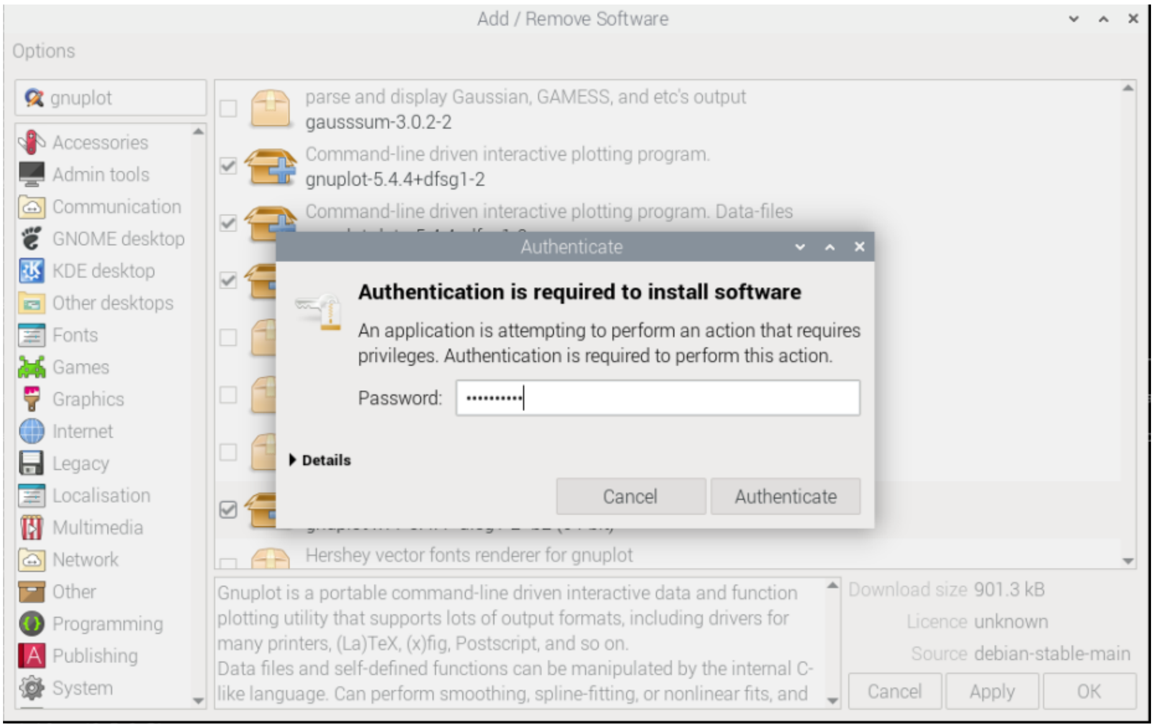
☑ Command-line driven interactive plotting program. X-package
gnuplot-x11-5.4.4+dfsg1-2+b2 (64-bit)

☐ Hershey vector fonts renderer for gnuplot

Gnuplot is a portable command-line driven interactive data and function plotting utility that supports lots of output formats, including drivers for many printers, (La)TeX, (x)fig, Postscript, and so on.
Data files and self-defined functions can be manipulated by the internal C-like language. Can perform smoothing, spline-fitting, or nonlinear fits, and

Download size 901.3 kB
Licence unknown
Source debian-stable-main

3. 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.



Authentication is required to install software

An application is attempting to perform an action that requires privileges. Authentication is required to perform this action.

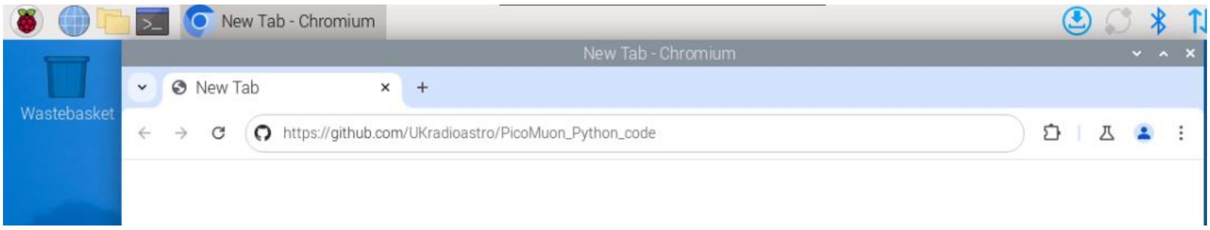
Password: ••••••••••

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

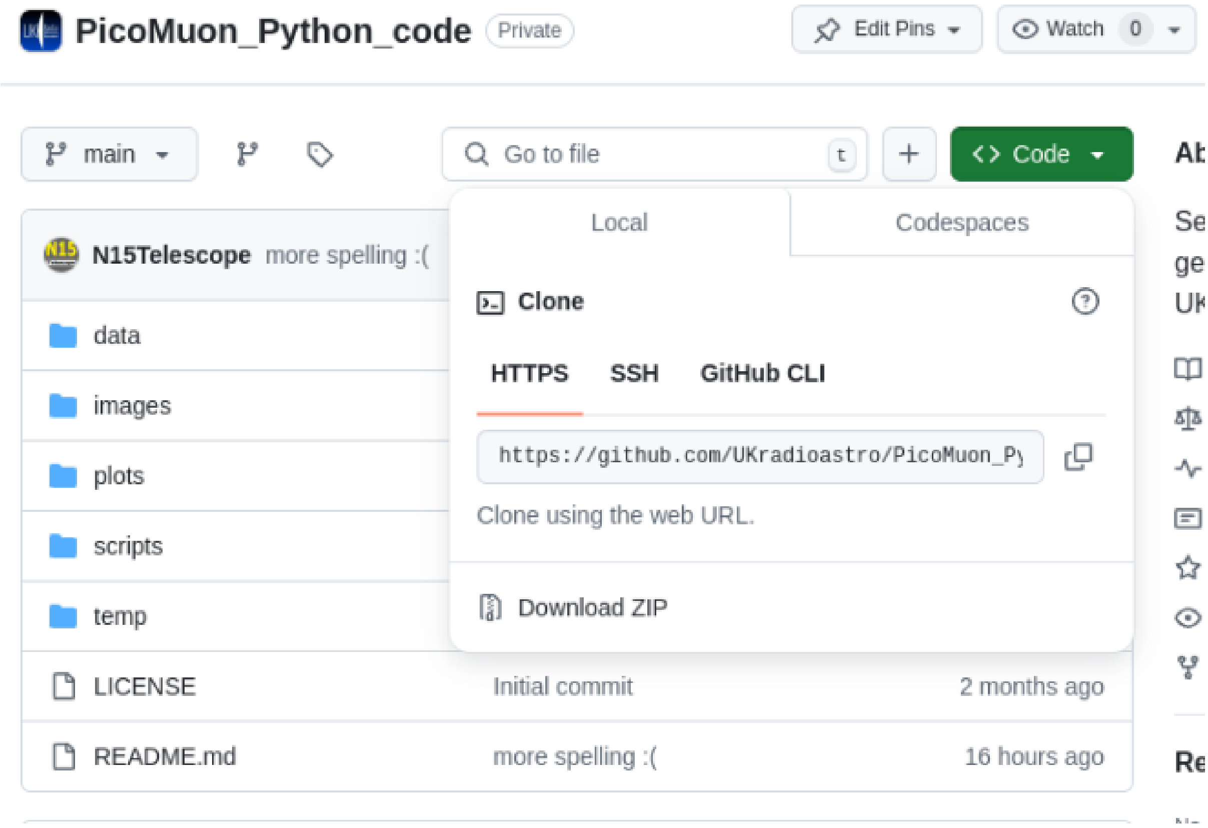
Get code from github

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

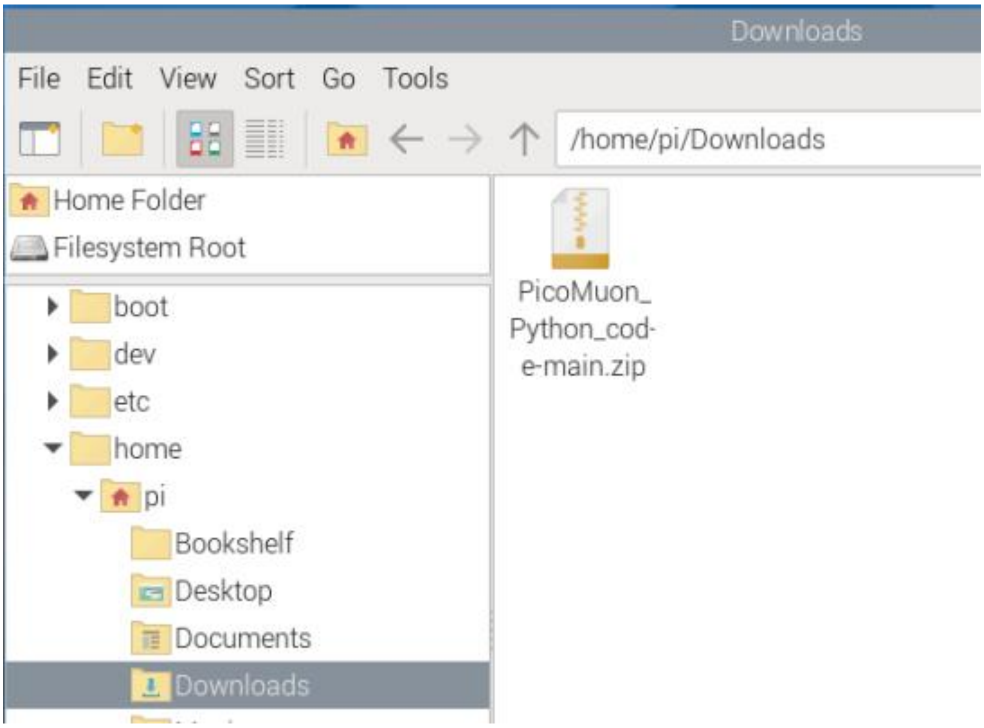
https://github.com/UKradioastro/PicoMuon_Python_code



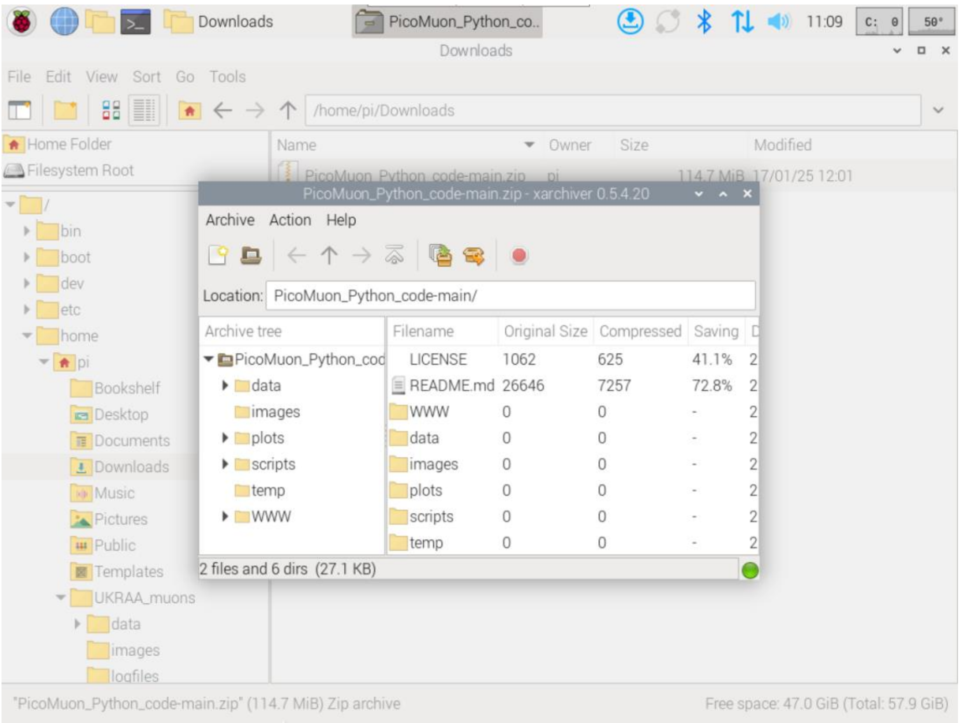
2. Select the **<> Code** tab and then select **Download ZIP**. This will download all the necessary code/files for the Python code. Close the web browser.



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.

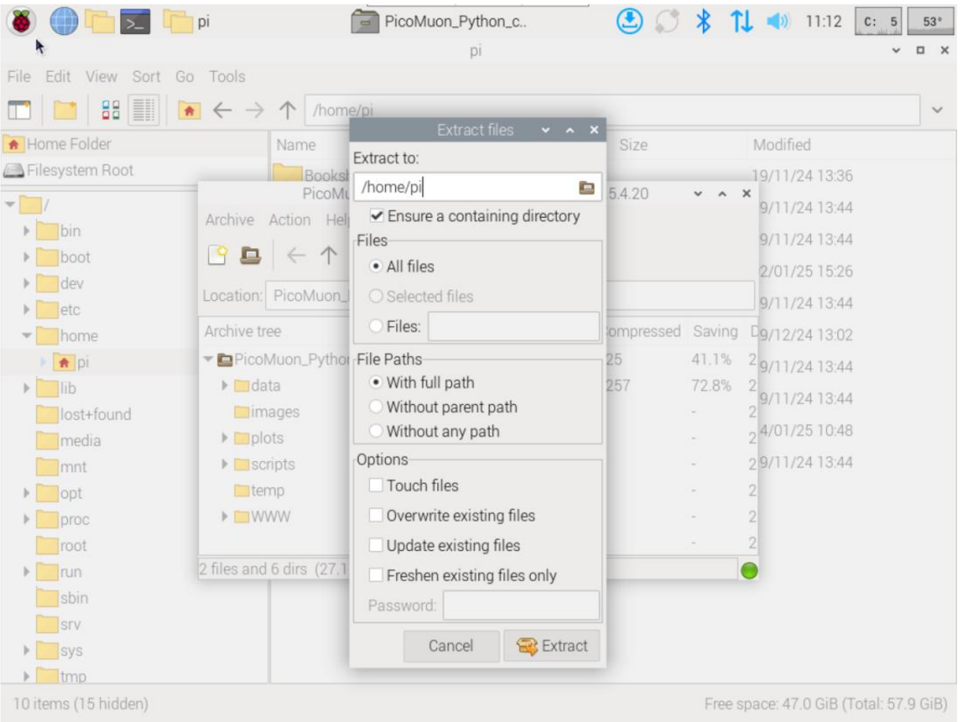


Location: PicoMuon_Python_code-main/

Filename	Original Size	Compressed	Saving	D
LICENSE	1062	625	41.1%	2
README.md	26646	7257	72.8%	2
WWW	0	0	-	2
data	0	0	-	2
images	0	0	-	2
plots	0	0	-	2
scripts	0	0	-	2
temp	0	0	-	2

2 files and 6 dirs (27.1 KB)

5. 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.



Extract files

Extract to: /home/pi

☑ Ensure a containing directory

Files

● All files

○ Selected files

○ Files:

File Paths

● With full path

○ Without parent path

○ Without any path

Options

□ Touch files

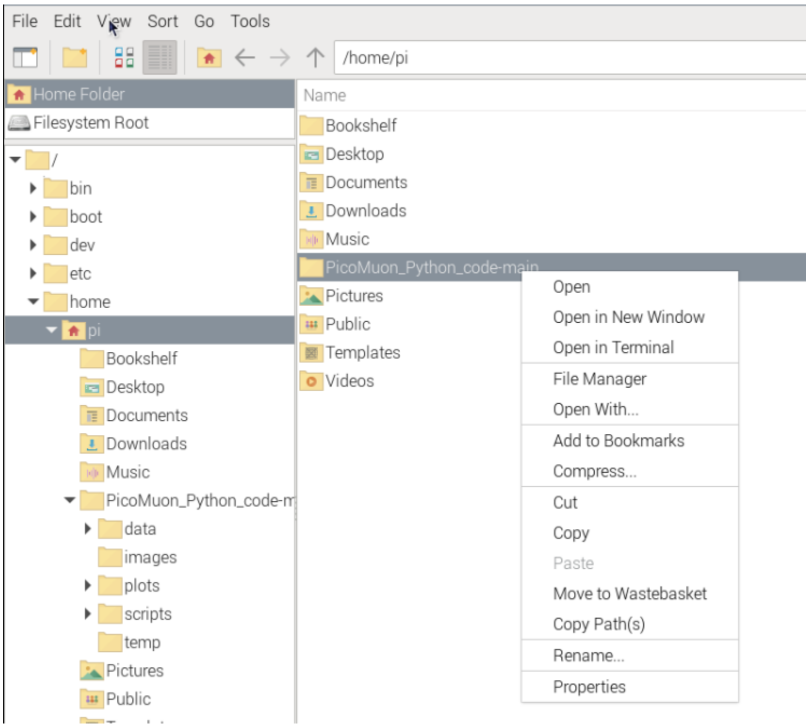
□ Overwrite existing files

□ Update existing files

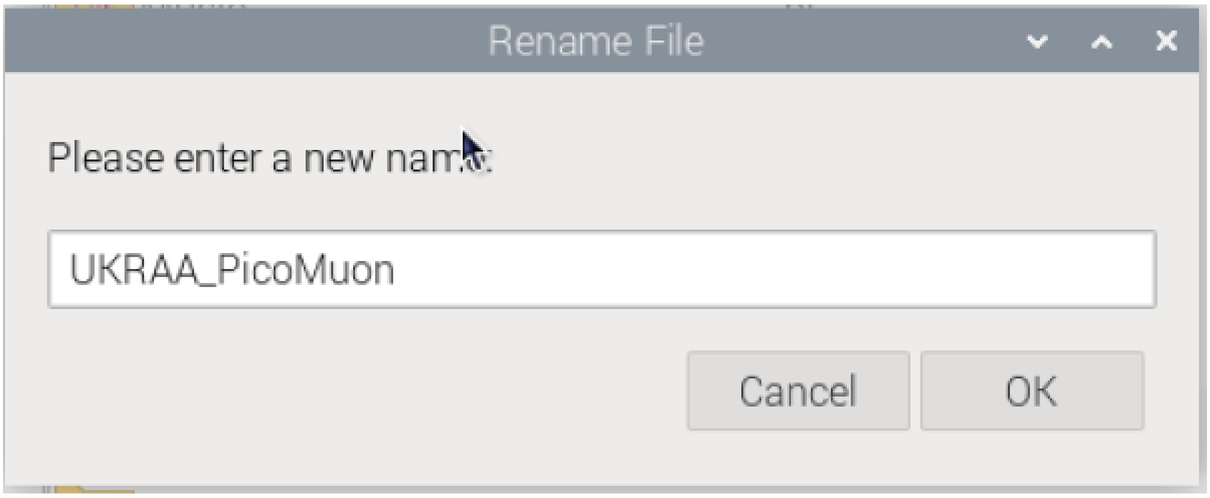
□ Freshen existing files only

Password:

6. 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**.


Rename File

Please enter a new name:

UKRAA_PicoMuon

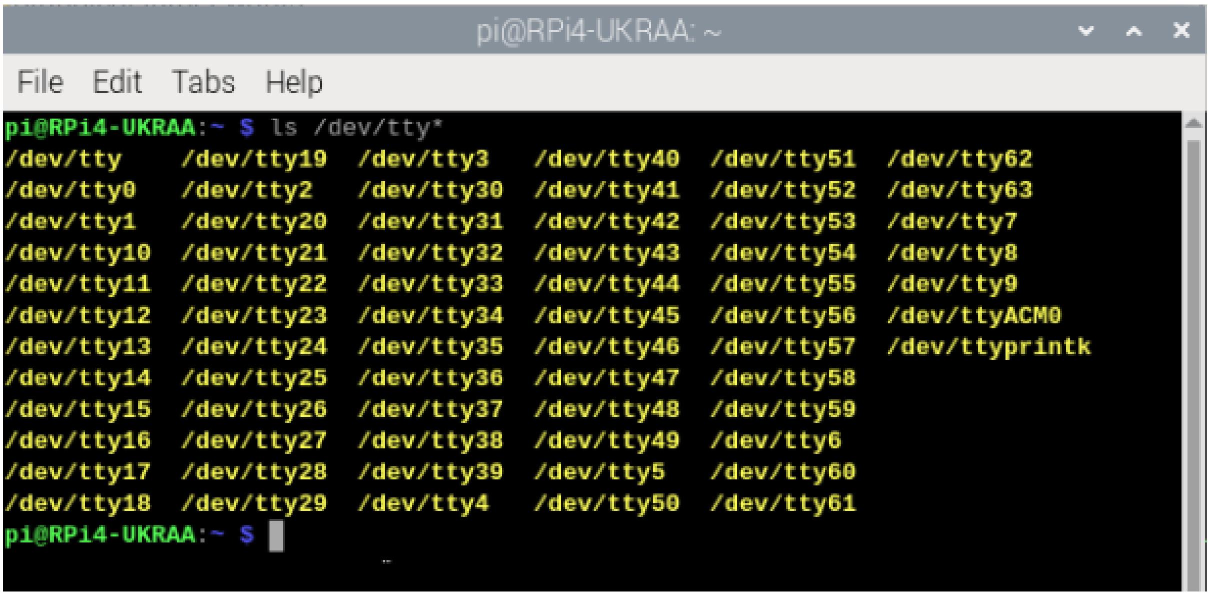
8. Scripts and folder structure should now be in place to run the code.

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:~ $ 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.

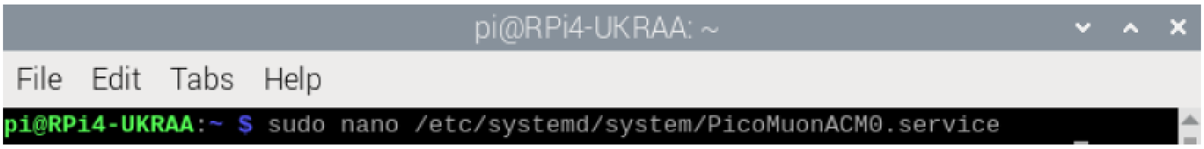
It is an easy process to set up services to run the code, check it is running and restart the code if it stops.

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.

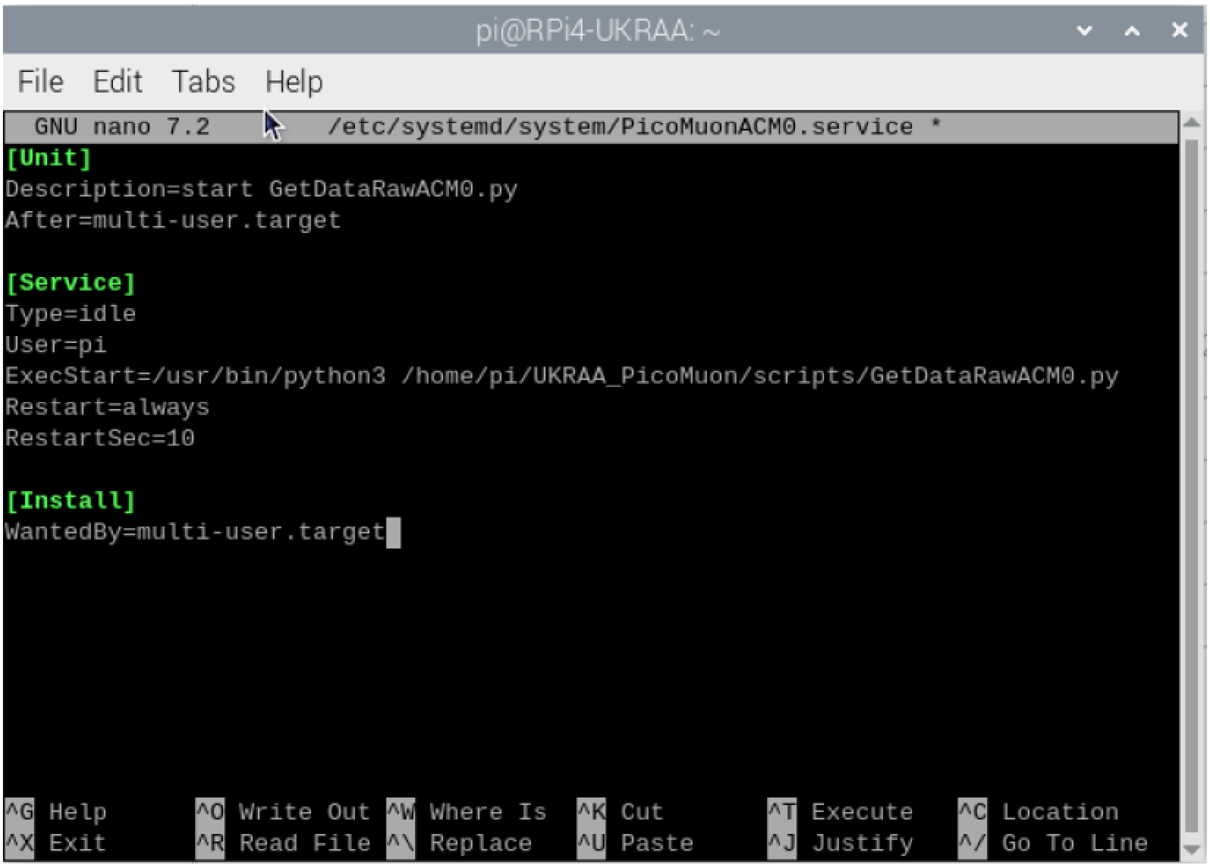
sudo nano /etc/systemd/system/PicoMuonACM0.service



```
pi@RPi4-UKRAA:~ $ 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
```

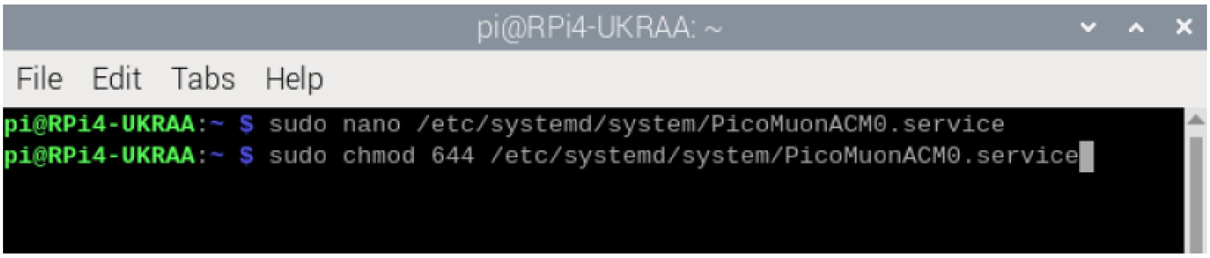


```
  GNU nano 7.2          /etc/systemd/system/PicoMuonACM0.service *
[Unit]
Description=start GetDataRawACM0.py
After=multi-user.target

[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.

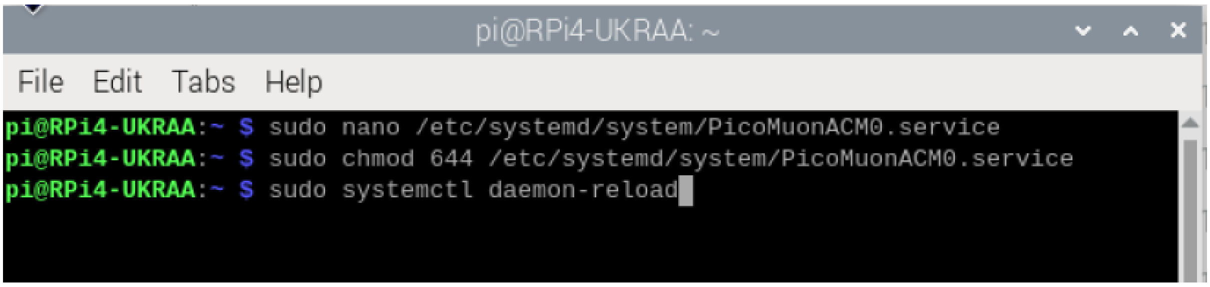
sudo chmod 644 /etc/systemd/system/PicoMuonACM0.service



```
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.

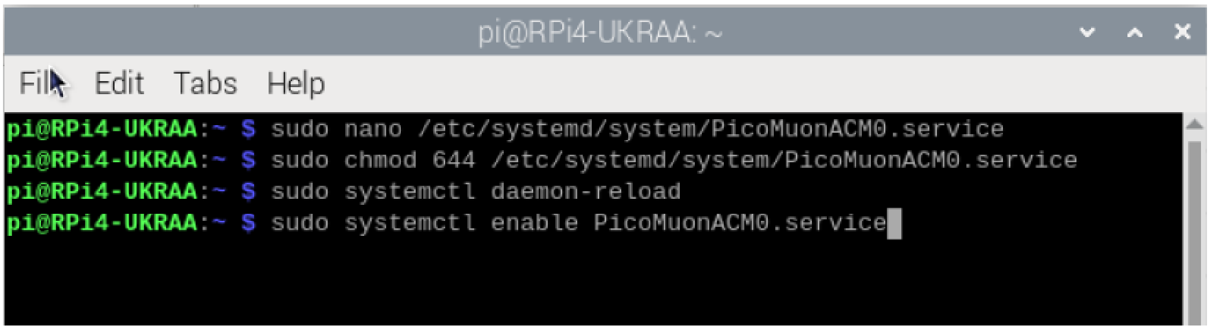
sudo systemctl daemon-reload



```
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

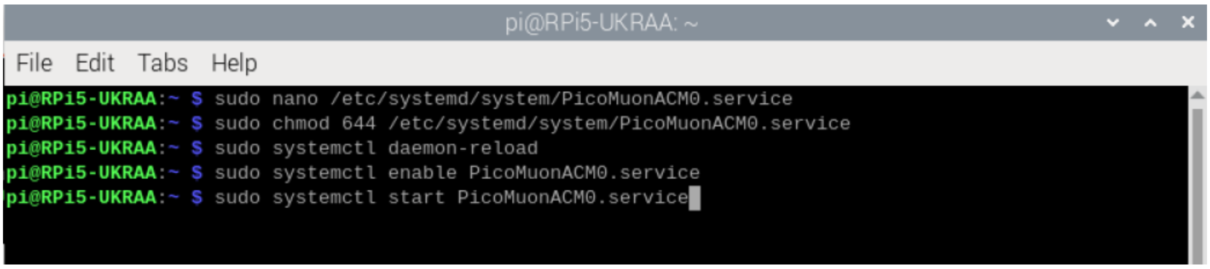


```
pi@RPi4-UKRAA:~ $ sudo nano /etc/systemd/system/PicoMuonACM0.service
pi@RPi4-UKRAA:~ $ sudo chmod 644 /etc/systemd/system/PicoMuonACM0.service
pi@RPi4-UKRAA:~ $ sudo systemctl daemon-reload
pi@RPi4-UKRAA:~ $ sudo systemctl enable PicoMuonACM0.service
```

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

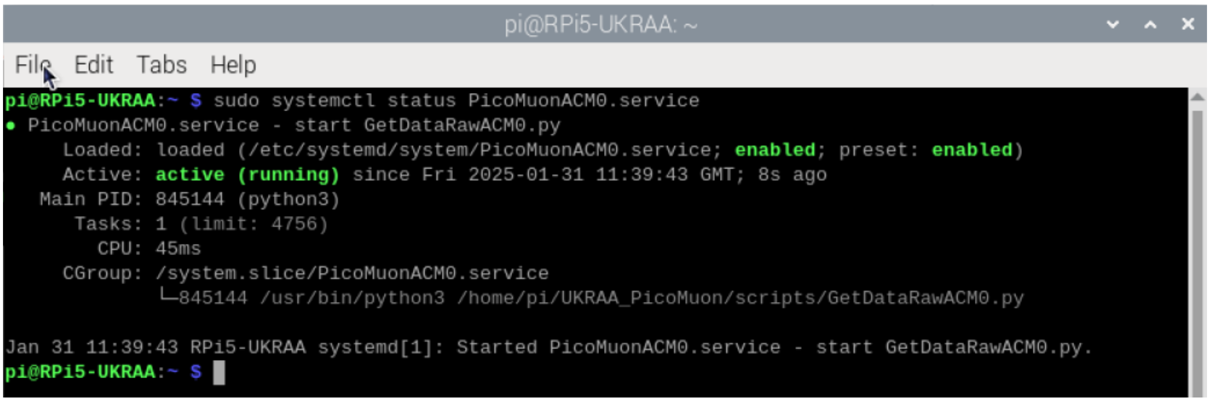


```
pi@RPi5-UKRAA:~ $ sudo nano /etc/systemd/system/PicoMuonACM0.service
pi@RPi5-UKRAA:~ $ sudo chmod 644 /etc/systemd/system/PicoMuonACM0.service
pi@RPi5-UKRAA:~ $ sudo systemctl daemon-reload
pi@RPi5-UKRAA:~ $ sudo systemctl enable PicoMuonACM0.service
pi@RPi5-UKRAA:~ $ sudo systemctl start PicoMuonACM0.service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

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

sudo systemctl status PicoMuonACM0.service

12. We expect to see...



```
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:

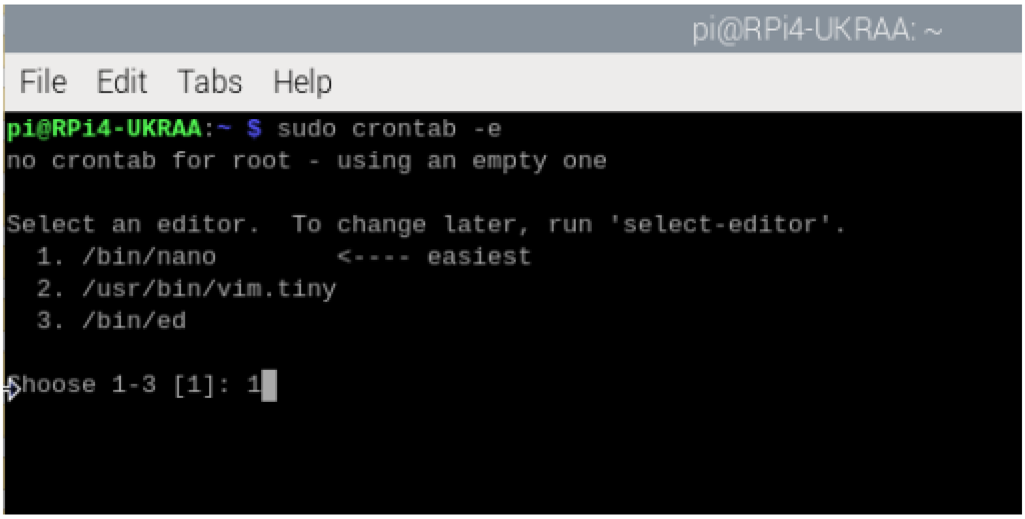
- counts per minute
- frequency of counts per minute
- frequency of ADC values recorded
-

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

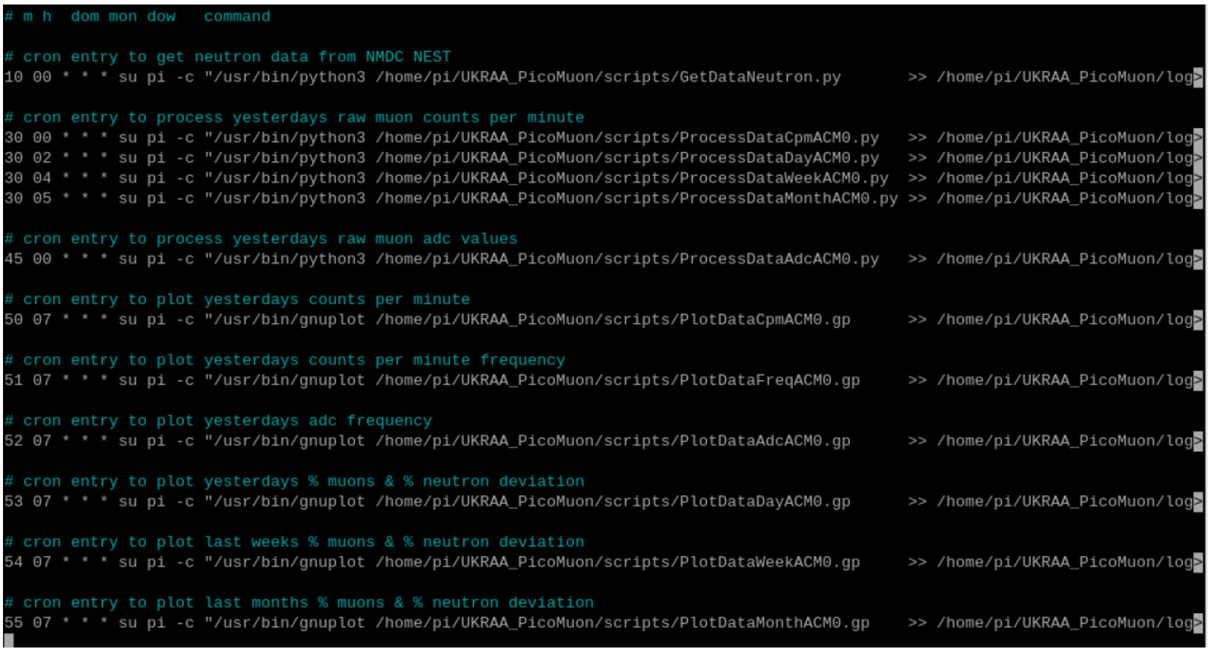


```
pi@RPi4-UKRAA:~ $ sudo crontab -e
no crontab for root - using an empty one

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**.

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

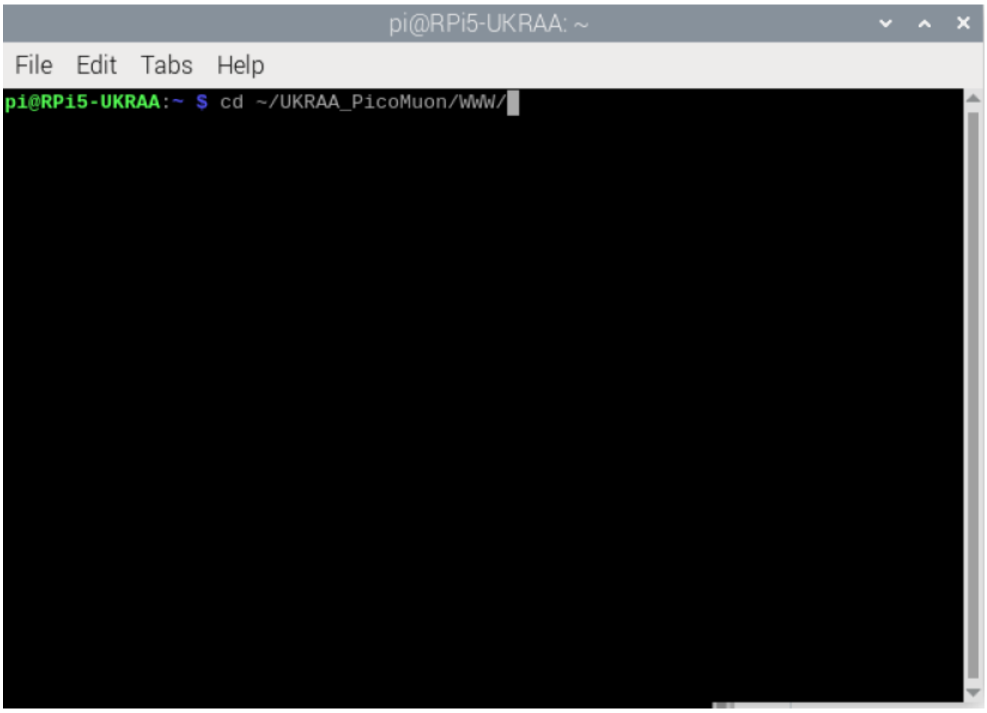
Once you have completed building your RPi web server, we need to move some files over to the **/var/www/html** folder on your RPi; these are all located in the WWW folder from your downloaded git zip file - as shown below.

```
└── 📁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.

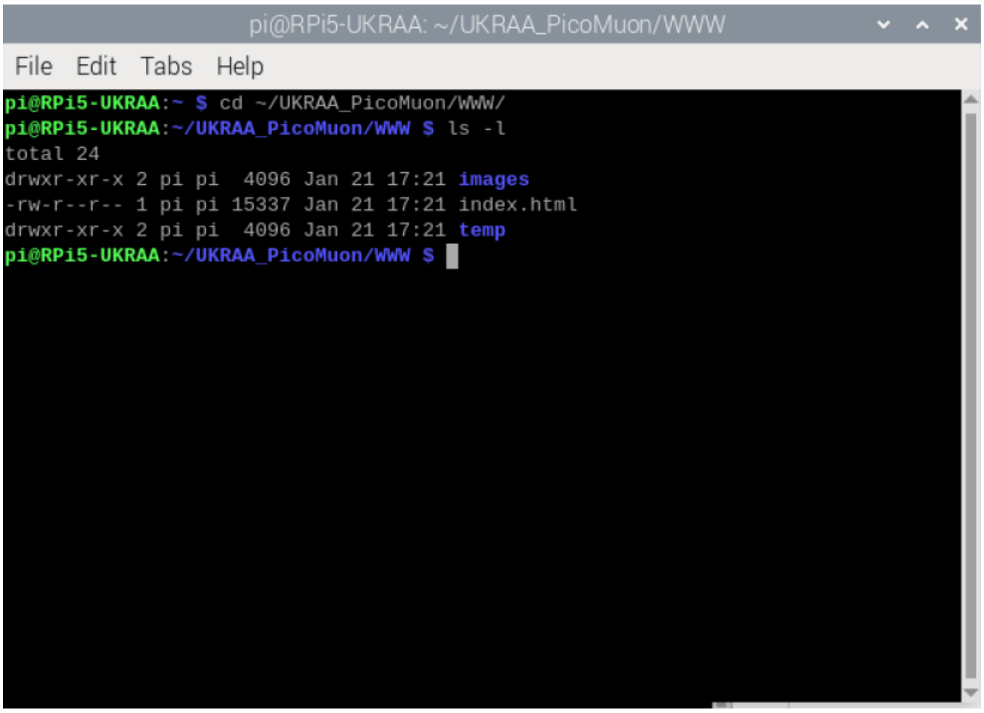
cd ~/UKRAA_PicoMuon/WWW/



```
pi@RPi5-UKRAA:~ $ 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

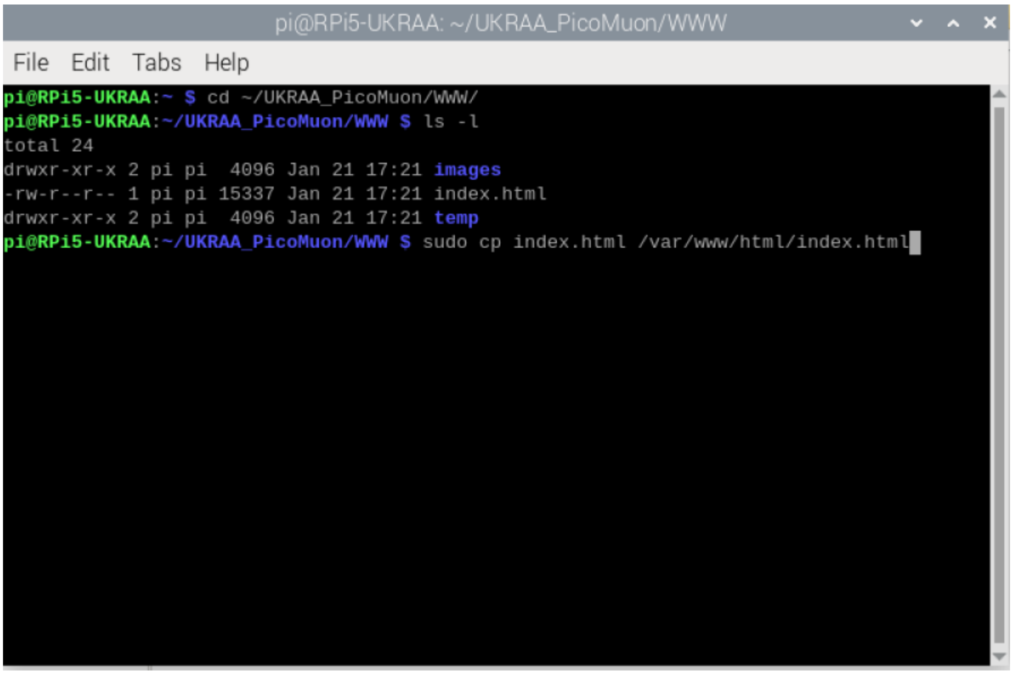


```
pi@RPi5-UKRAA:~ $ cd ~/UKRAA_PicoMuon/WWW/
pi@RPi5-UKRAA:~/UKRAA_PicoMuon/WWW $ ls -l
total 24
drwxr-xr-x 2 pi pi  4096 Jan 21 17:21 images
-rw-r--r-- 1 pi pi 15337 Jan 21 17:21 index.html
drwxr-xr-x 2 pi pi  4096 Jan 21 17:21 temp
pi@RPi5-UKRAA:~/UKRAA_PicoMuon/WWW $
```

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.

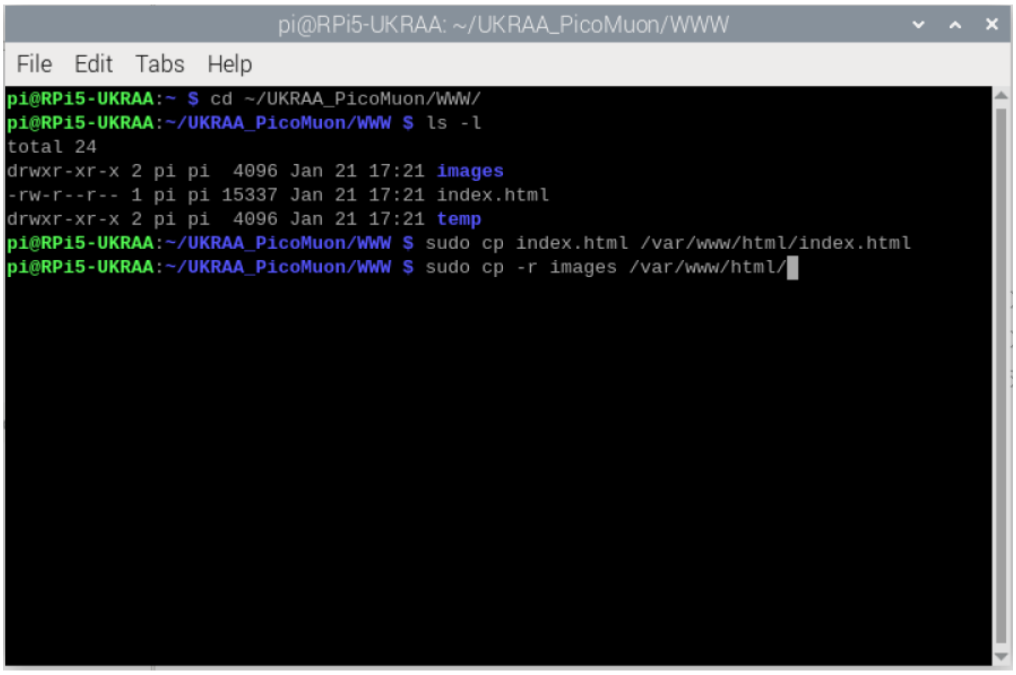
sudo cp index.html /var/www/html/index.html



```
pi@RPi5-UKRAA:~ $ 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.

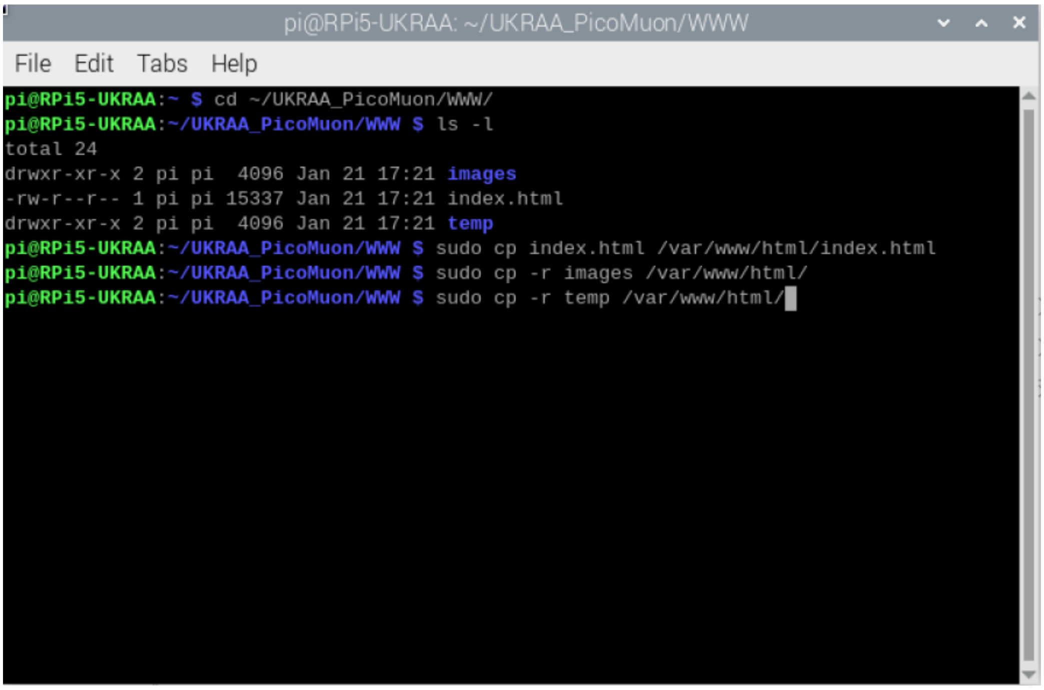
sudo cp -r images /var/www/html/



```
pi@RPi5-UKRAA:~ $ 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/
```

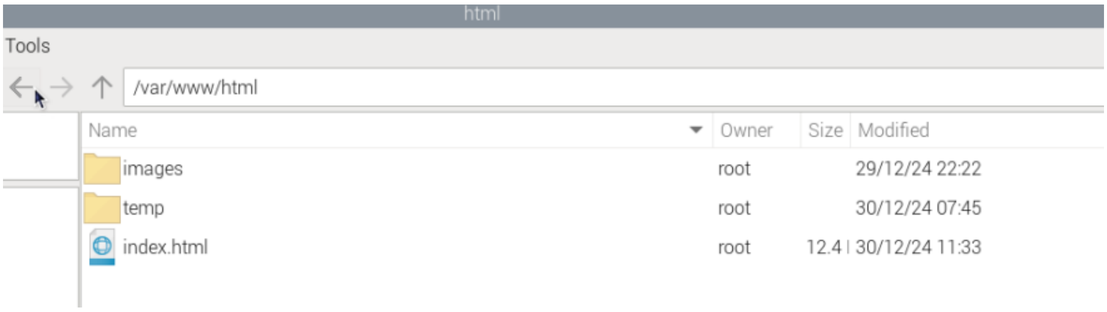
7. Type the following command and press enter - this will copy the **temp** directory and content.

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



```
pi@RPi5-UKRAA:~ $ 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/
```

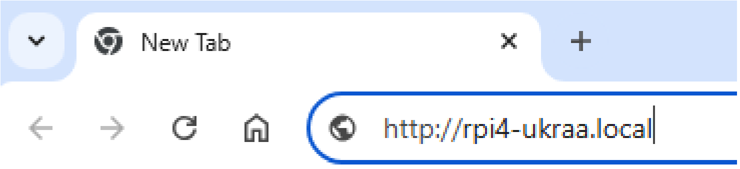
8. From file manager, you can check that the file/folders have been copied correctly, by navigating to **/var/www/html** and you should see the following.



/var/www/html

Name	Owner	Size	Modified
images	root		29/12/24 22:22
temp	root		30/12/24 07:45
index.html	root	12.4	30/12/24 11:33

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.



http://rpi4-ukraa.local

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**.

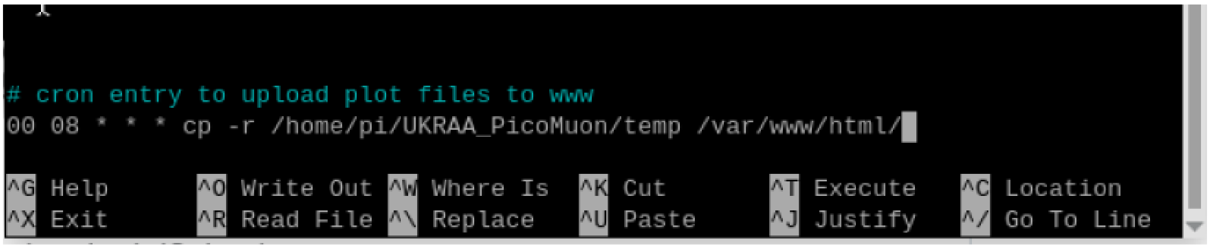
1. Open terminal window

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/
```

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.

License

Contact us

Please send an e-mail to picomuon@ukraa.com

UK Radio Astronomy Association