Installation with screen shots

The new ‘Bookworm’ OS release

After the design, software and articles for this project were fully completed, a new Raspberry Pi and Operating System were announced. The clock would not install or work with the new version of OS and you can imagine how happy I was. But that’s not important right now because I’ve now addressed all of the new OS’s differences.

What’s important for you is that there are some advantages to choosing the recently released ‘Bookworm’ version of the OS instead of the previous ‘Bullseye’ version. For example, the Y2038 time bug present in the 32-bit version of Bullseye has been fixed in 32-bit Bookworm and the volume level for bluetooth audio streams can now be adjusted from the originating bluetooth device (as well as using the clock’s volume control). Bookworm will receive support until mid 2028 whereas Bullseye’s ends in 2026. However even after the OS version has passed its end of life and is no longer supported, clocks should keep running indefinitely.

Although you can now run 64-bit Bookworm on a Pi 3 (which runs faster and more efficiently than its the 32-bit counterpart), many of the 64-bit applications still do not work on a Pi3, including the mpv media player used by the alarm-clock. So for the time being, stick with 32-bit Bookworm on the Pi3.

At the time of writing, Bookworm seems to be stable. For the first few weeks after its release, many updates were released in quick succession, some of them changing behaviours compared with Bullseye and so introducing new problems with code that had been written to work with the former version of the operating system. The stream of changes now seems to have settled down.

I’ve implemented work-arounds within the installer, web pages and the clock software to detect which version of the operating system is being used, and to change behaviour accordingly. My various clocks are once again working nicely with the new version of OS and with the older Bullseye version too.

I trust that by the time you read this, Bookworm will be stable and you’ll not need to worry. If per-chance you strike inexplicable trouble with either the installer or clock software running under Bookworm, consider reverting to Bullseye which has been stable and working well with this clock design since early 2023.

Preparing the SD card – with screenshots

The installation process is fully covered within the magazine articles, but there was insufficient space to include screenshots.

Start with a blank SD card sized with at least 4Gbyte storage. If you need to buy an SD card, you’ll likely find it difficult to source a card as small as this. Anything larger will be fine and you can use the extra storage to hold your media library.

With Raspberry Pis, the read/write speed and quality of the SD card makes a difference. Cheap SD cards often have slow read/write speeds and can perform poorly. A faster SD card is recommended and you can find more information on the Internet if you’re interested.

The Raspberry Pi and its installation script require the SD card to be formatted with either Debian Bullseye Lite or Debian Bookworm Lite. Images older than Bullseye are not compatible. Bookworm is the latest release.

The main difference between ‘Lite’ and ‘Full’ SD images is that the Lite versions do not include a Linux GUI nor support HDMI video output. That’s perfect for the alarm-clock, which doesn’t require a monitor. We’ll be using SSH to connect to the clock, and the Lite Debian Bullseye and Lite Debian Bookworm versions will meet this need.

The easiest way to prepare the SD card is with ‘Raspberry Pi Imager’, freely available for Windows, Mac OSX and Linux. You may have used this before and once you’re familiar with it, the procedure is easy. However as fate would have it, the user interface to the Raspberry Pi Imager programme was changed in November 2023 after the release of the Pi5, making it a little more tricky and confusing to describe the procedure in a unified way.

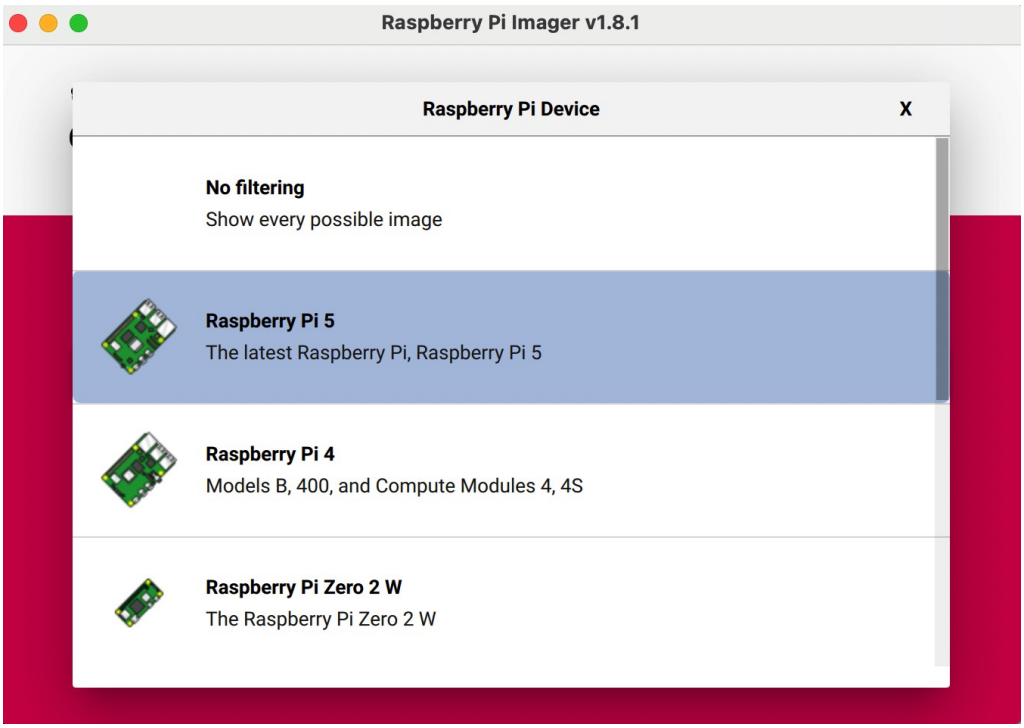
I've reproduced the Raspberry Pi Imager instructions below including screenshots for both the new and the old versions. After you have prepared the SD card image, insert it into your Raspberry Pi and continue with the installation process as documented afterwards.

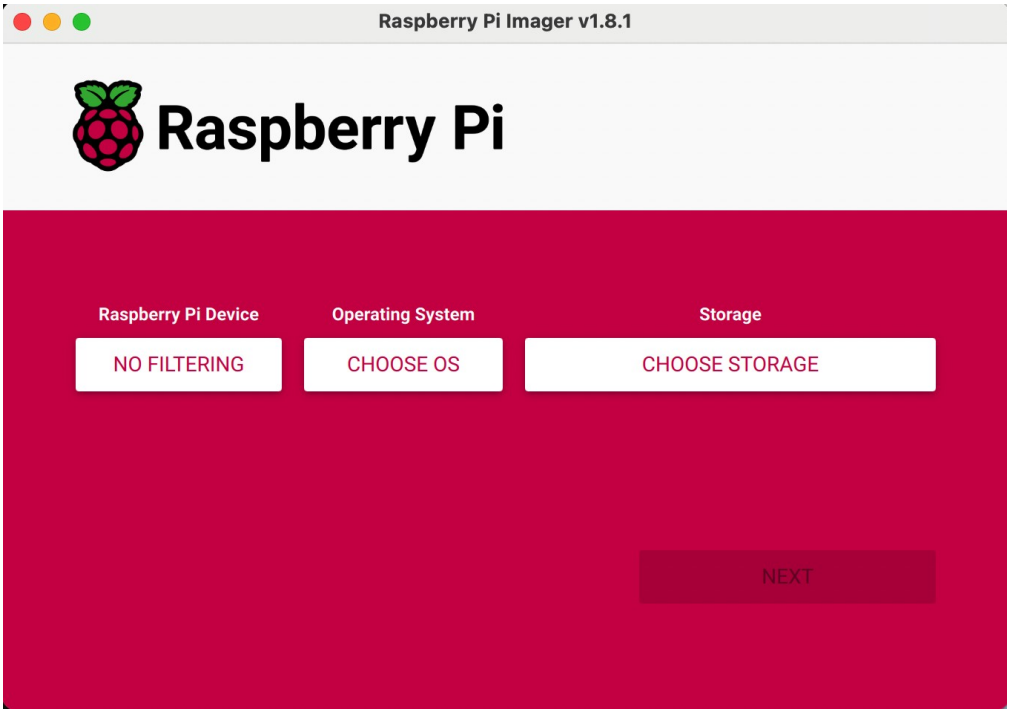
Installation using newer Raspberry Pi Imager version

Launch the newer version of Raspberry Pi Imager, insert an SD card into your computer and click on the Raspberry Pi Device button.

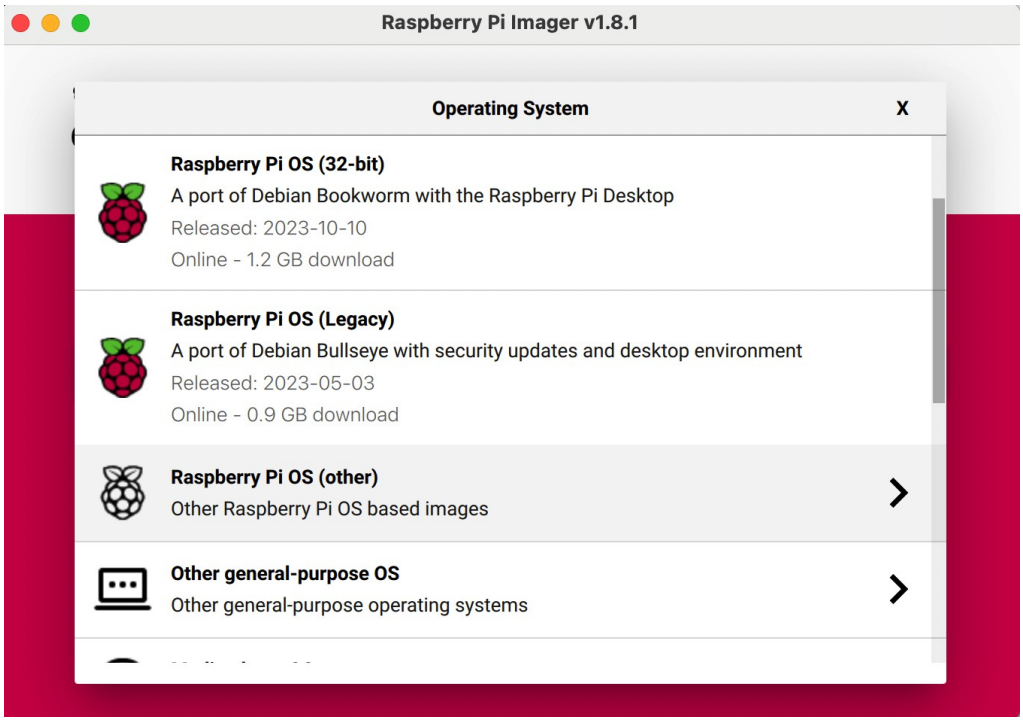
At the time of writing, what you need to do next will be a surprise. The new version of Raspberry Pi Imager has introduced an option to identify the type of Pi hardware you're preparing the image for. Based on what you tell it, the programme will restrict the images it offers to install, presumably to make it easier to ensure you choose one that's compatible with your hardware. The problem is that for Pi models 3, it doesn't offer Bookworm at all, or even offer a 32 bit Lite image, which is what we need for the clock! I suspect this is an imager bug that will be fixed in the future. Thankfully, there's a work around.

When you click on 'CHOOSE DEVICE', you'll see the window below. Set the device type to 'No Filtering' and you'll get to choose from all available images.





Next, click on ‘CHOOSE OS’.



Next, scroll downwards and click ‘Raspberry Pi OS (Other)’.

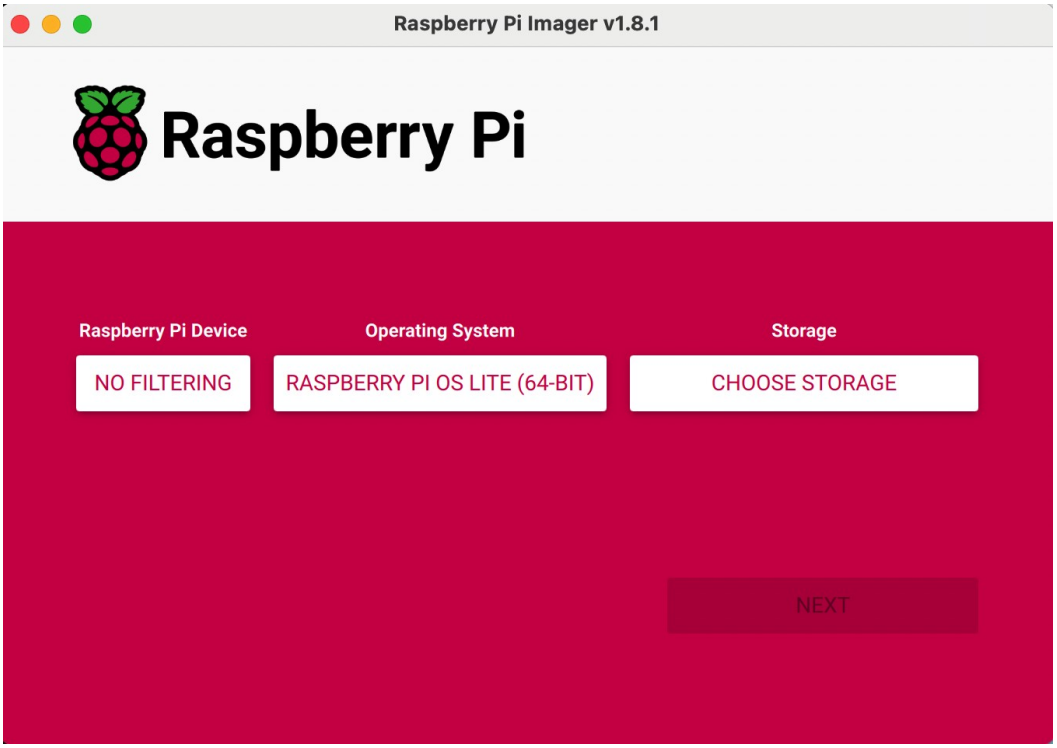
Depending upon whether you wish to install Bookworm or Bullseye, and depending upon your Pi hardware, carefully read and choose one of the following:

Bookworm (the new version of the OS):

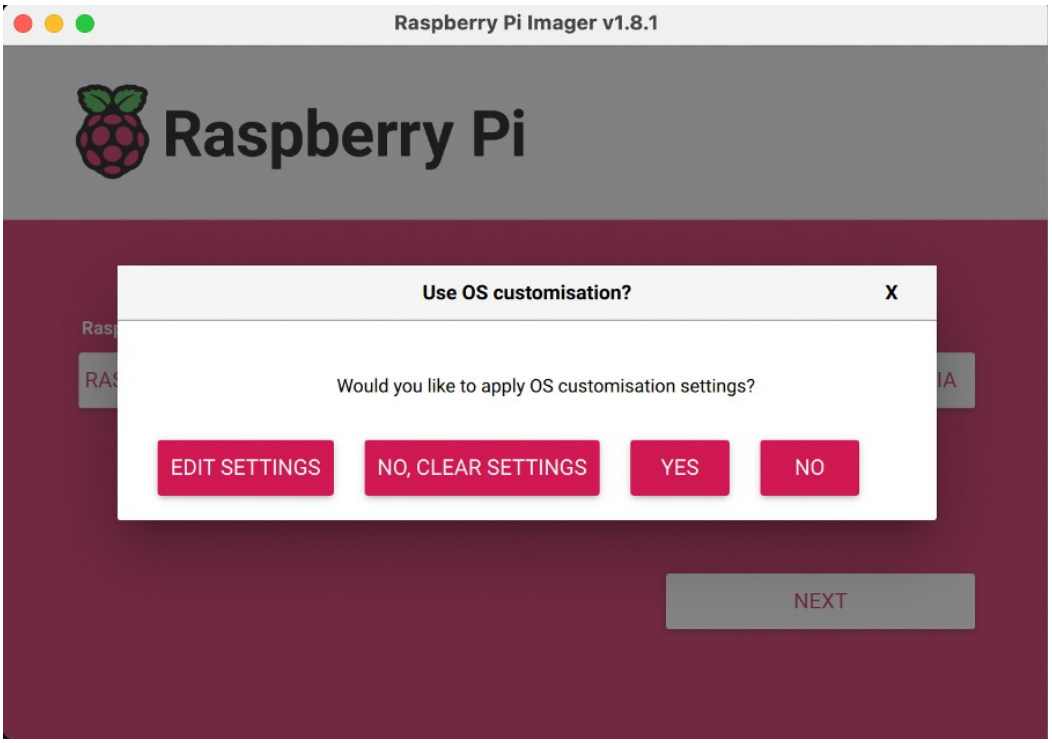
- For a Pi4, Pi5 or Pi Zero 2W, choose **Raspberry Pi OS Lite (64-bit)**’.
- For other models including the Pi3, choose ‘**Raspberry Pi OS Lite (32-bit)**’.

Bullseye (the previous version of the OS):

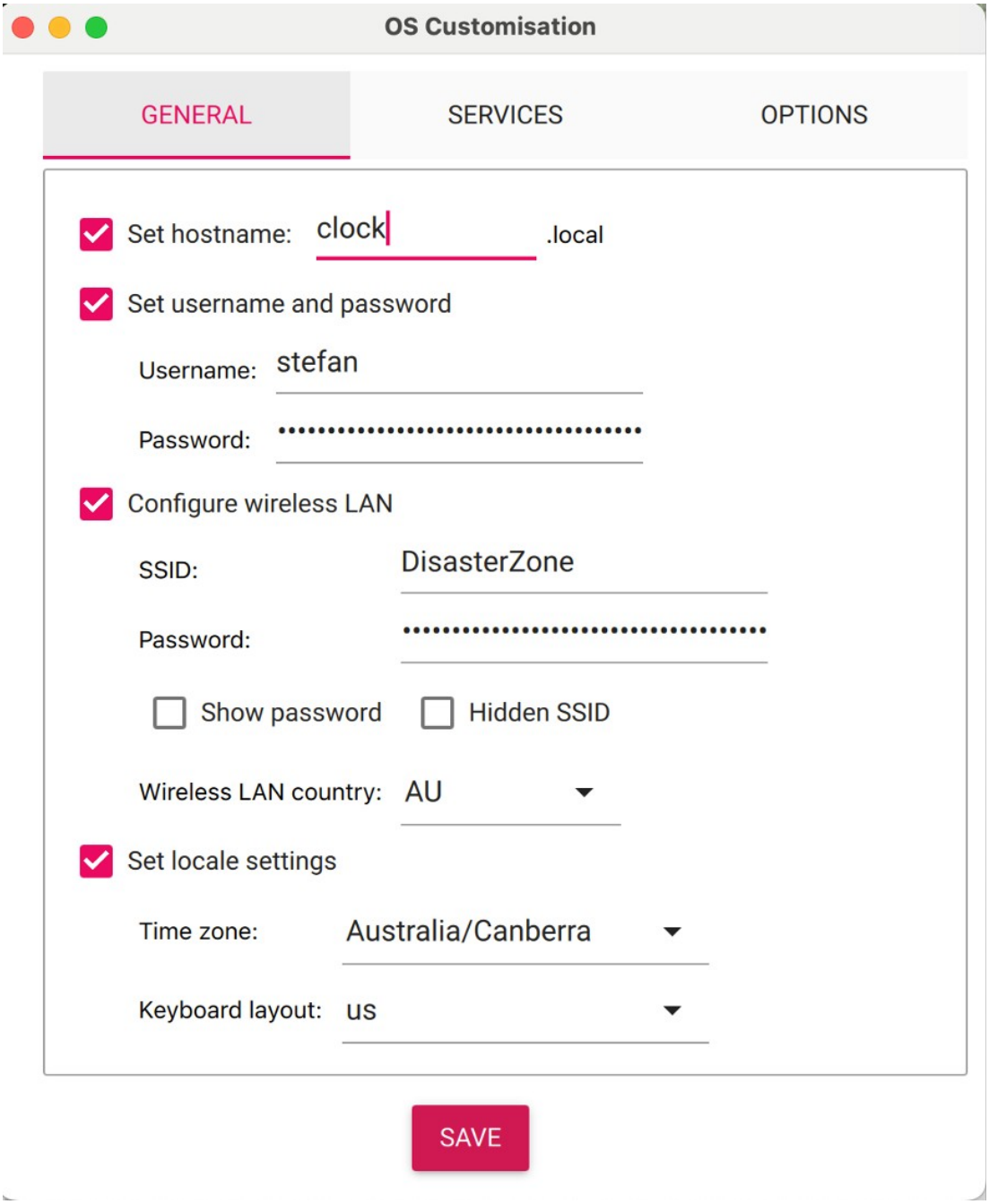
- For a Pi4 or Pi Zero 2W, choose **Raspberry Pi OS (Legacy, 64-bit) Lite**’.
- For other models including the Pi3, choose ‘**Raspberry Pi OS (Legacy) Lite**’.
- (Note: Bullseye is not compatible with a Pi5)



Next click the Choose Storage option and select the SD card onto which you would like to write the image. You will then be able to click ‘NEXT’ at the bottom right corner of the window which only becomes available after choosing storage.

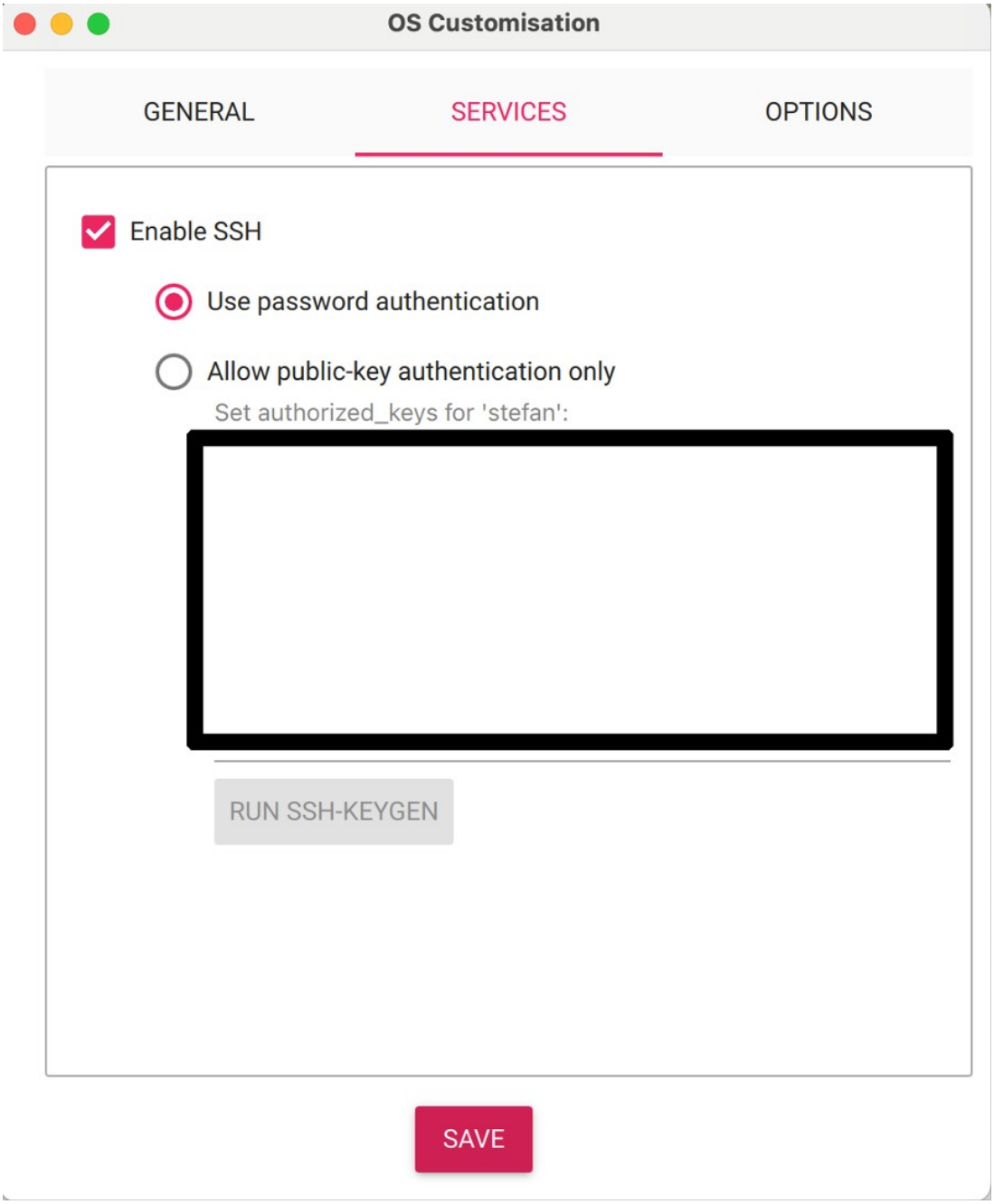


You’ll now be asked if you would like to apply OS customisation settings. Choose ‘EDIT SETTINGS’. A window with several tabs will appear. Go through these tabs one at a time and fill in details to suit your situation. If you are unsure, refer to the notes below.

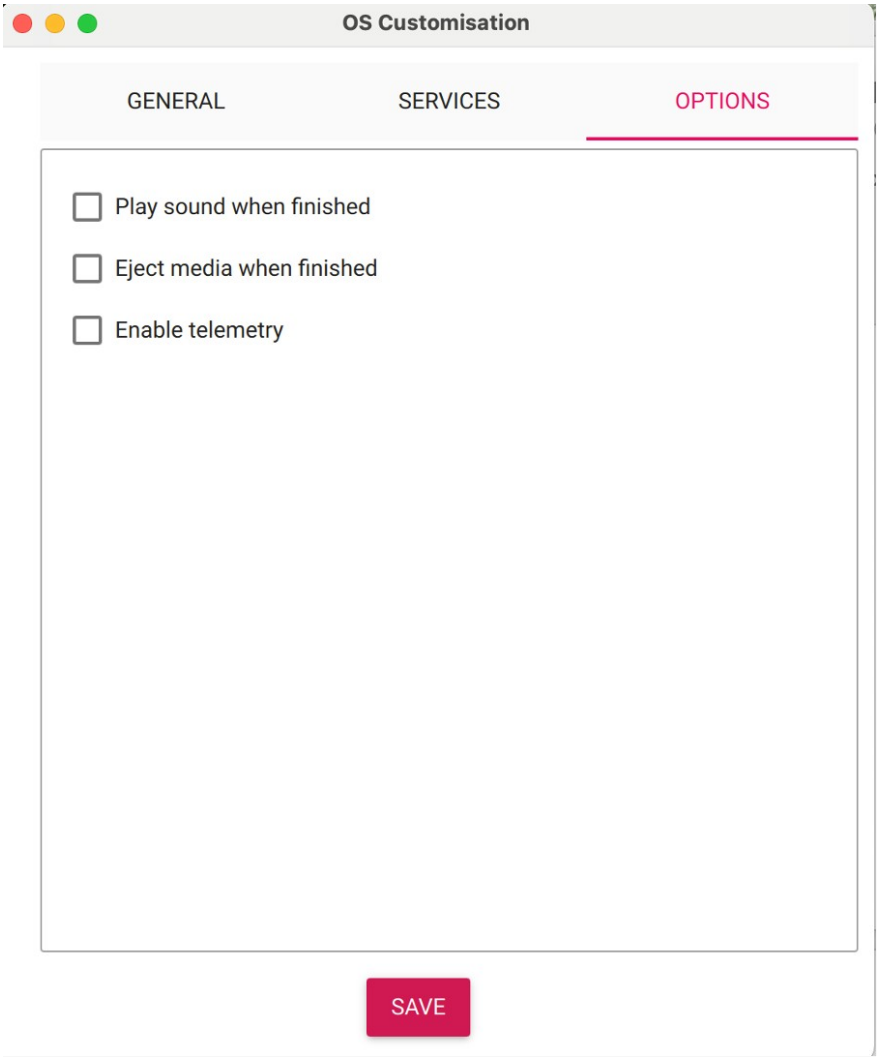


Under the 'GENERAL' tab, you'll want to

(a) set a unique hostname for your clock (such as 'clock' if you can't think of anything else, but something different if you have previously used that name),
(b) set a username and password for logging in via SSH,
(c) configure your wireless LAN details (SSID, password and country) so the Pi can automatically connect with the correct wi-fi frequencies,
(d) set locale settings for your area so that the right time zone and daylight savings settings are applied



(e) Under the ‘SERVICES’ tab, you’ll want to enable SSH using password authentication

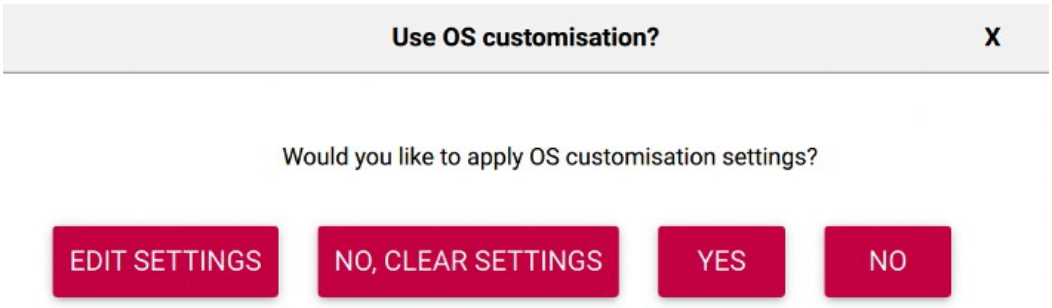


Under the ‘OPTIONS’ tab, you’ll want to

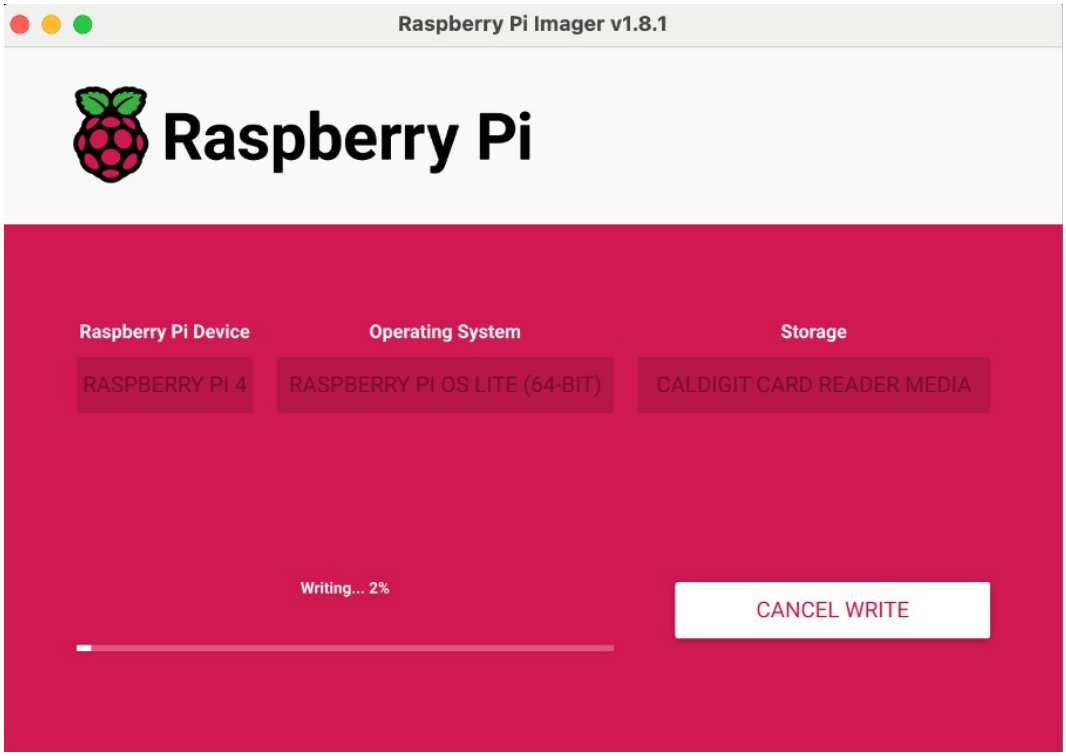
(f) Deselect the option to eject media (i.e. the SD card) when finished, as you’ll also be copying the clock software to the SD card before ejecting.

(g) Deselect the ‘enable telemetry’ option (the clock installer script will configure everything the clock requires).

Before clicking ‘SAVE’ on the last tab, make sure you write down all information to remember it later when you need to log into the Pi.



After saving, make sure to click ‘YES’ to apply the settings you just edited!

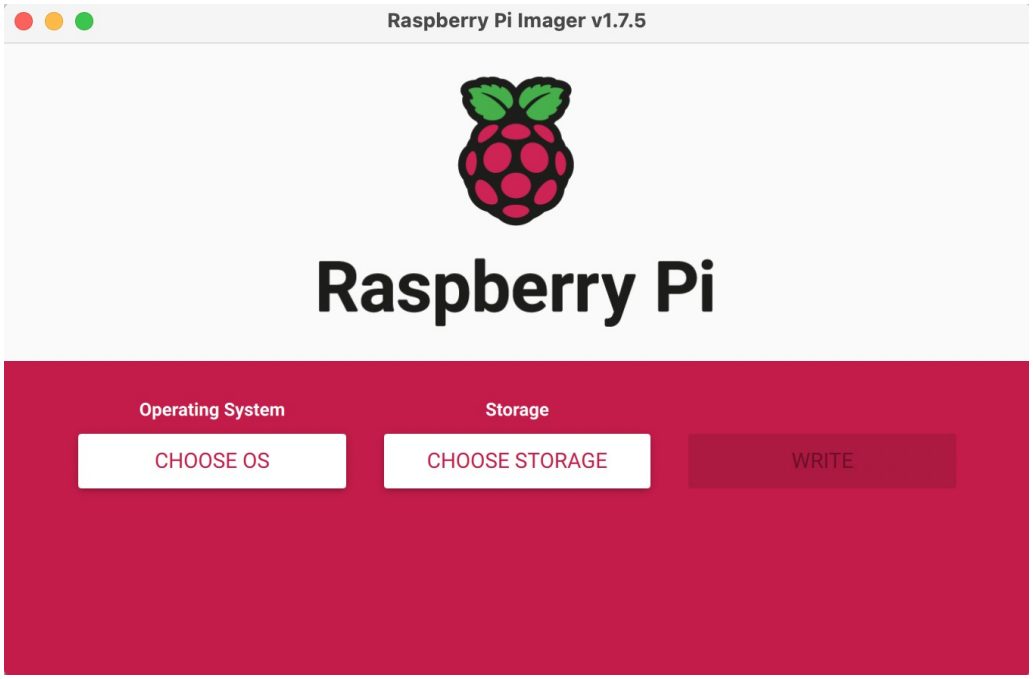


Next click ‘WRITE’ to initiate the formatting and writing of the SD card. And you can go and make a cup of tea as it writes.

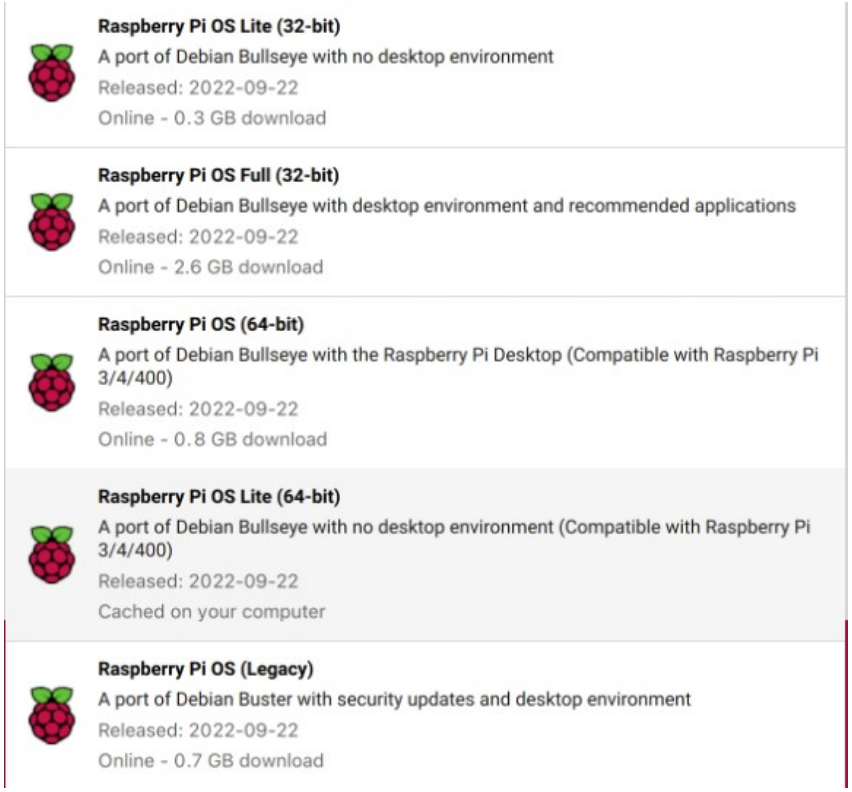
When the SD card has been written, there’s one last task before ejecting. You’ve already downloaded the clock software zip file from the Silicon Chip website because you are reading this! Inside the same zip file, there is a tar-ball that you now need to copy onto the SD card. The tar-ball is called ‘alarm-clock_v01.tgz’. Copy it to the root of the ‘bootfs’ partition on the SD card the same way you transfer files to a thumb drive. The clock tar-ball should be called ‘alarm-clock_v01.tgz’ or with a number greater than ‘01’. If there are updates in the future, they’ll be numbered incrementally.

Eject the SD card, insert it into the Pi and apply power.

Installation using older Raspberry Pi Imager version

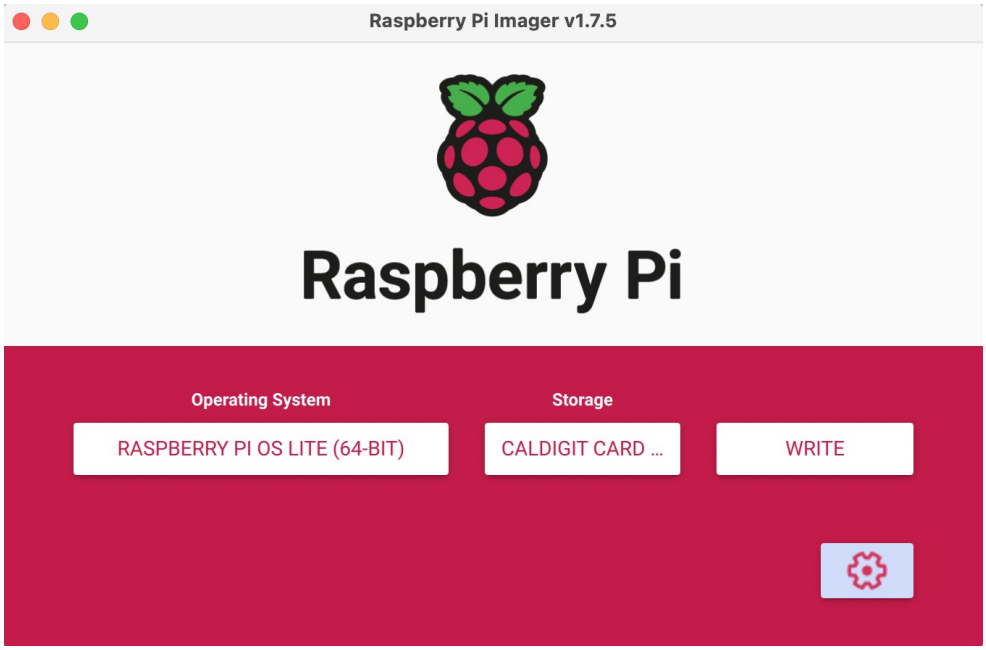


Launch Raspberry Pi Imager, insert an SD card into your computer and click on the CHOOSE OS button.

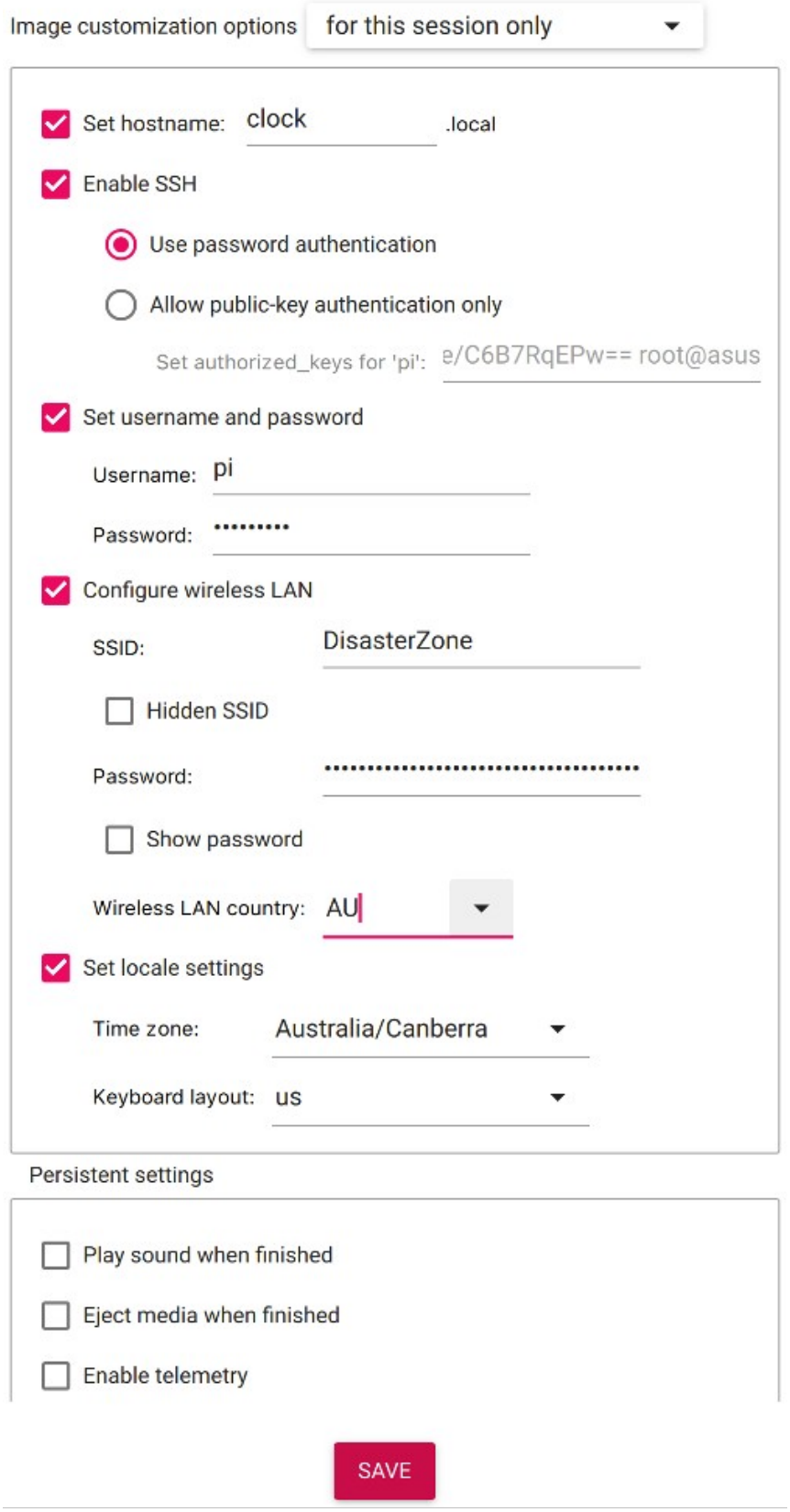


(The image above shows Bullseye, but the default version in Raspberry Pi Imager is now Bookworm)

- For a Pi4, Pi5 or Pi Zero 2W, choose 'Raspberry Pi OS (Other) -> Raspberry Pi OS Lite **(64-bit)**'.
- For other models, choose 'Raspberry Pi OS (Other) -> Raspberry Pi OS Lite **(32-bit)**'.



You'll see a cog-wheel icon appear in the main window after you've selected the OS. Click it and fill out the form.



You'll want to

(a) set a unique hostname for your clock (such as 'clock' if you can't think of anything else, but something different if you have previously used that name),
(b) enable SSH using password authentication,
(c) set a username and password for logging in via SSH,
(d) configure your wireless LAN details (SSID, password and country) so the Pi can automatically connect with the correct wi-fi frequencies,
(e) set locale settings for your area so that the right time zone and daylight savings settings are applied, and
(f) deselect the option to eject media (i.e. the SD card) when finished, as you'll also be copying the clock software to the SD card before ejecting.

Before saving, make sure you write down all information to remember it later when you need to log into the Pi.

Next click CHOOSE STORAGE to specify the media and finally click WRITE to format the SD card.

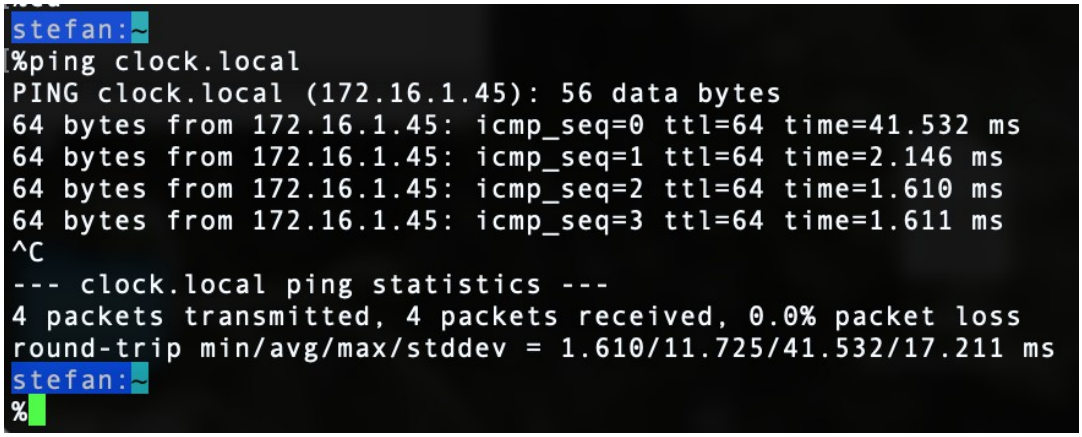
When the SD card has been written, there's one last task before ejecting. You've already downloaded the clock software zip file from the Silicon Chip website because you are reading this! Inside the same zip file, there is a tar-ball that you now need to copy onto the SD card. The tar-ball is called 'alarm-clock_v01.tgz'. Copy it to the root of the 'bootfs' partition on the SD card the same way you transfer files to a thumb drive. The clock tar-ball should be called 'alarm-clock_v01.tgz' or with a number greater than '01'. If there are updates in the future, they'll be numbered incrementally.

Eject the SD card, insert it into the Pi and apply power.

Installing clock software onto fresh OS image

Finding the Pi on your network and connecting

Because there's no video output, the only way to know the Pi is ready to proceed is that you'll eventually be able to ping it, either over wi-fi or a wired network connection. The first time a Pi boots a fresh image, it could take a few minutes longer than normal to respond. To avoid frustration, apply power, go and make a cup of tea, then come back and ping it.



In order to ping it, you'll either need its IP address, or you can use the hostname specified when you prepared the SD image earlier. In general, ***most*** home routers publish local hostnames using a '.local' suffix as suggested in the Raspberry Pi Imager programme. So you can try to ping clock.local (if you chose to call the hostname 'clock'). Older home routers may not conform with the '.local' naming standard and may not even publish a suffix at all. Other routers may publish a different suffix than '.local'. If the '.local' suffix does not work for you, consult your router's documentation or just look at the router's DHCP leases table to find the IP address that has been allocated to the new Pi. After you find its IP address, ping that address to know that the Pi is up and responding.

When you're receiving ping responses, you can ssh to the same hostname or IP address using your favourite method (for example PuTTY or OpenSSH for Windows, or the command line ssh tool for MAC and Linux). When you connect, the Pi will prompt for the username and password that you specified during the SD card setup. When you are logged in, you will be in the home directory of the account you created for yourself.

Installing the clock software

On the Pi, the tar-ball you copied to the SD card earlier is available within the bootfs partition which gets named '/boot' on the Pi. From your home directory on the Pi and assuming you copied the tar-ball to the root of the bootfs partition, you can now extract the contents using one of the two command options shown below:

For the Bookworm version of the OS:

```
tar zxf /boot/firmware/alarm-clock_v01.tgz
```

For the Bullseye version of the OS:

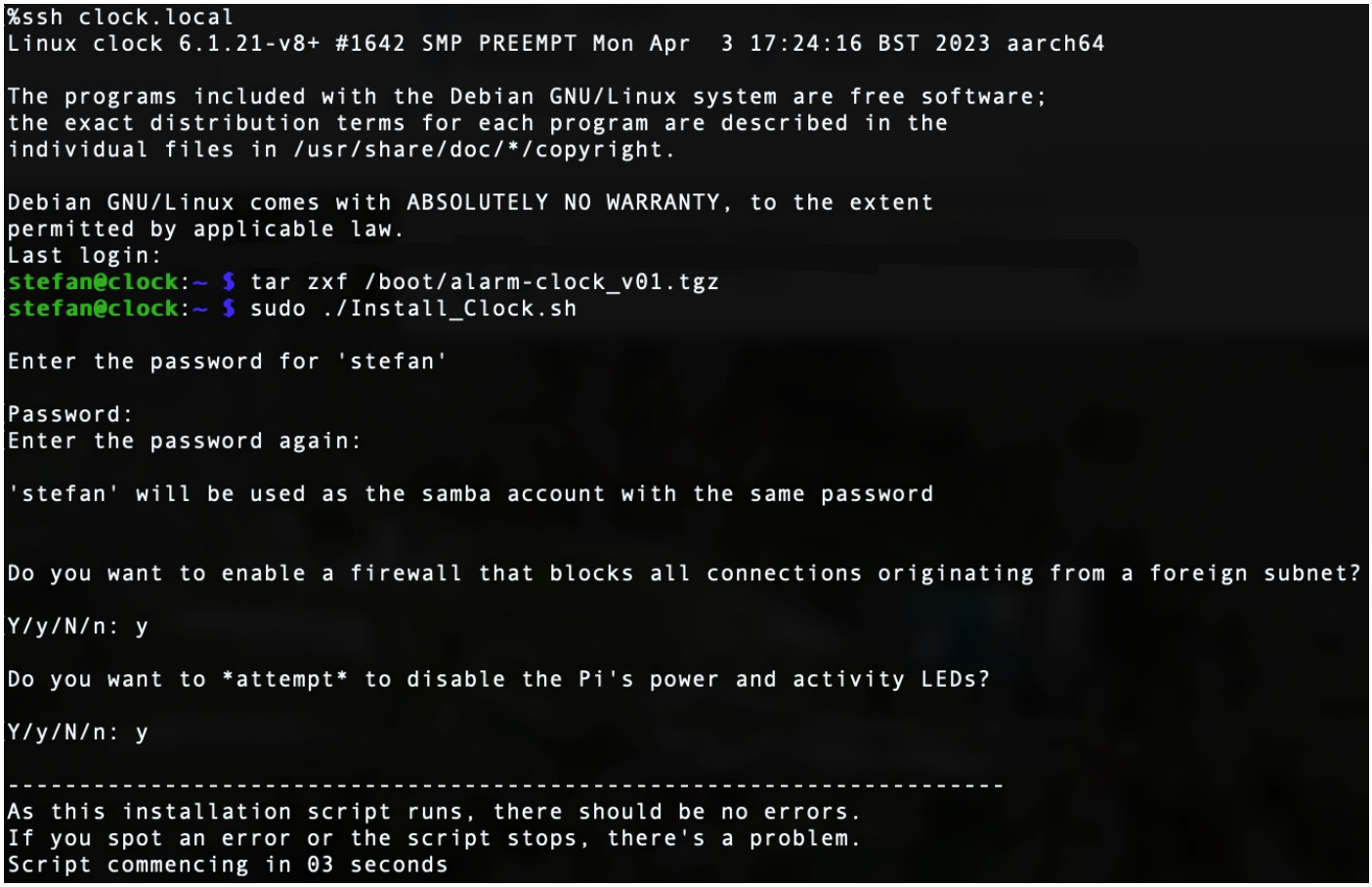
```
tar zxf /boot/alarm-clock_v01.tgz
```

The command will create a subdirectory called alarm-clock containing the source code, and will also leave an installation script in your current directory.

The last stage in the software installation is to run the installation script:

```
sudo ./Install_Clock.sh
```

The installation script asks for your password twice, whether you would like to install firewall rules that prevent access from IP addresses originating on a different subnet: you will probably want to say yes; and asks if you would like to attempt to disable the power and activity LEDs.



That's it, except for waiting for the script to run and do its thing. It will download recent Linux updates, install libraries required by the alarm-clock and then compile and install the alarm-clock software.

If you specified, the script will attempt to extinguish the Pi's power and activity LEDs so that they aren't 'beacons in the night'. However even Pis of the same model but different vintage can be inconsistent with respect to LED configuration so there's a chance on your particular Pi that one or both of the LEDs will remain enabled. If this happens, you'll need to search the Internet about how to disable the LEDs for your version of Pi.

If you'd like to re-enable the LEDs after having previously disabled them, rerun the 'sudo ./Install_Clock.sh' script again and answer the question about disabling the LEDs differently.

When done, which may take ten minutes or more on slower broadband connections or slow SD cards, the script asks if its OK to reboot. When the Pi comes back online, the clock software will be running regardless of whether there is any hardware connected and you'll be able to access its web interface and network file share.

The Install_Clock.sh script should run without errors, so if you see any, something is wrong.

New versions of the alarm-clock software

The first version of software for this project will be released in a tar-ball file named 'alarm-clock_v01.tgz'. If for any reason a new version of the software needs to be released, it will be called 'alarm-clock_v02.tgz' and so on.

After installing the clock software for the first time, there are two methods you could use to install an update.

1. Open up the clock, remove the SD card from the Pi and plug it into your computer, copy the new tar-ball file onto the SD card, put the SD card back into the Pi, exact the tar-ball contents and then rerun the Install_Clock.sh script. The disadvantage of this method is that you'd need to open up your clock.

2. You can avoid opening up the clock by network-copying the new tar-ball into the root of your home account on the Pi using the Pi's samba file-server.

Either extract its contents and re-run the installation script as described above to compile and install the new version. Alternately you can manually compile and install the new software. The following commands assume you network-copied the tar-ball into the root of your Pi's home directory:

```
tar zxf alarm-clock_v02.tgz
cd alarm-clock
make
make install
```