

Raspberry Pi alarm-clock, media player and bluetooth speakers project

Preparing the SD card

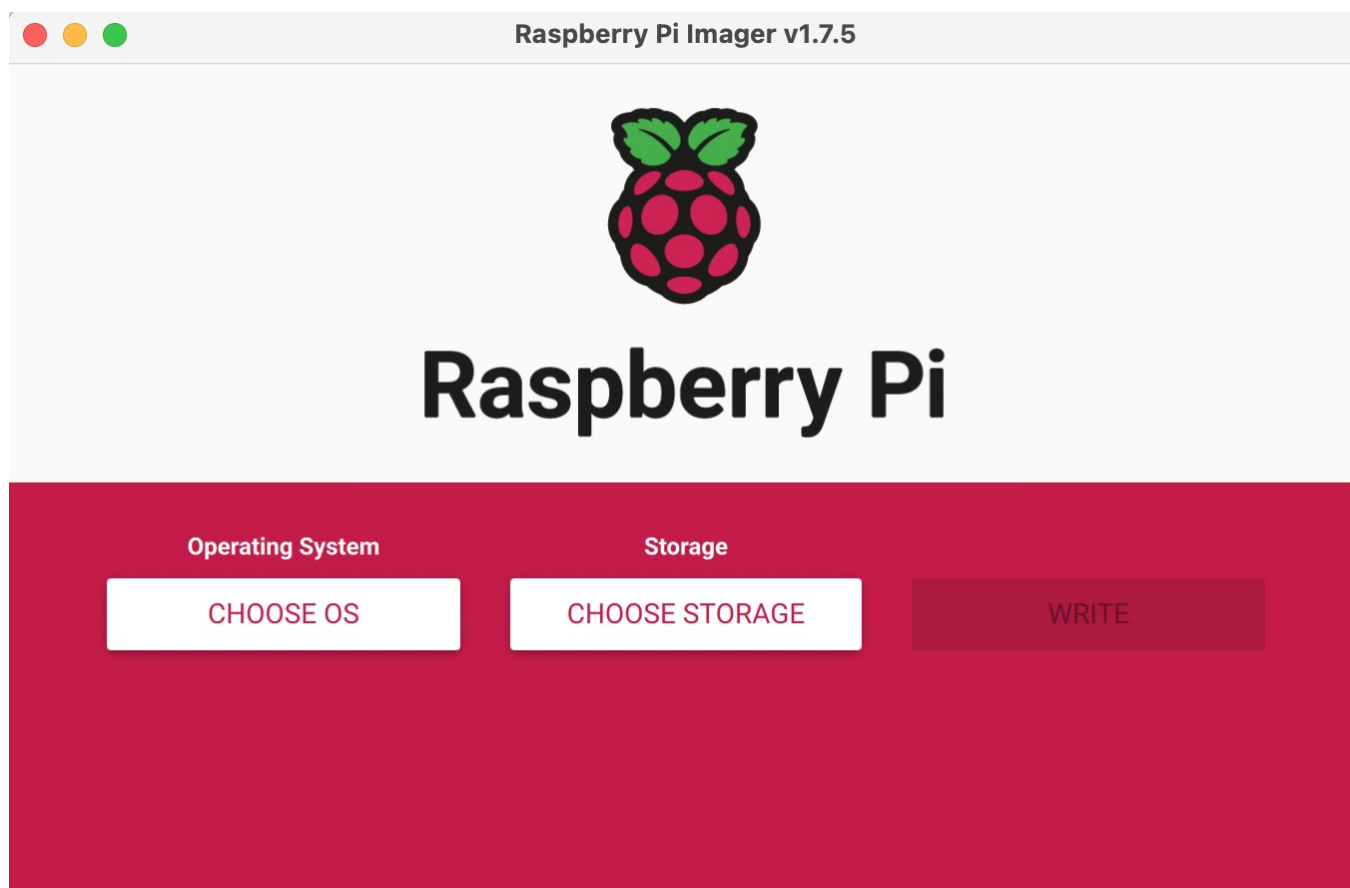
Start with a blank SD card sized with at least 4Gbyte storage. If you need to buy an SD card, you'll likely find it difficult to source a card as small as this. Anything larger will be fine and you can use the extra storage to hold your media library.

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. Bookworm is the later release and is recommended, but the clock has been tested with either of these versions and works OK. Images older than Bullseye are not compatible.

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.

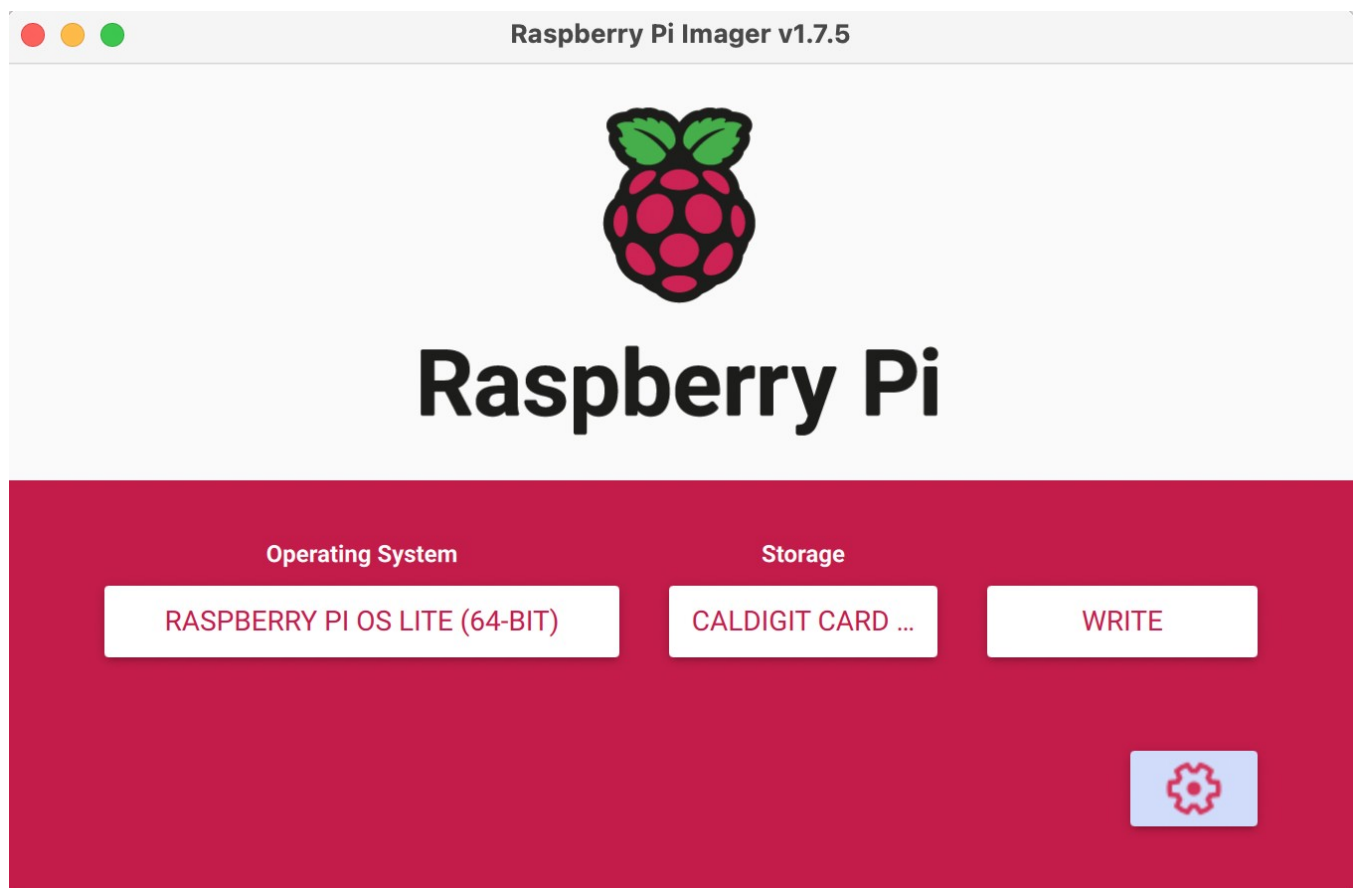


Launch Raspberry Pi Imager, insert an SD card into your computer and click on the CHOOSE OS button.

	Raspberry Pi OS Lite (32-bit) A port of Debian Bullseye with no desktop environment Released: 2022-09-22 Online - 0.3 GB download
	Raspberry Pi OS Full (32-bit) A port of Debian Bullseye with desktop environment and recommended applications Released: 2022-09-22 Online - 2.6 GB download
	Raspberry Pi OS (64-bit) A port of Debian Bullseye with the Raspberry Pi Desktop (Compatible with Raspberry Pi 3/4/400) Released: 2022-09-22 Online - 0.8 GB download
	Raspberry Pi OS Lite (64-bit) A port of Debian Bullseye with no desktop environment (Compatible with Raspberry Pi 3/4/400) Released: 2022-09-22 Cached on your computer
	Raspberry Pi OS (Legacy) A port of Debian Buster with security updates and desktop environment Released: 2022-09-22 Online - 0.7 GB download

(The image above shows Bullseye, but the default version is now Bookworm)

- For a Pi4 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.

Image customization options for this session only

☒ Set hostname: .local

☒ Enable SSH

☒ Use password authentication

☐ Allow public-key authentication only

Set authorized_keys for 'pi':

☒ Set username and password

Username:

Password:

☒ Configure wireless LAN

SSID:

☐ Hidden SSID

Password:

☐ Show password

Wireless LAN country: ▼

☒ Set locale settings

Time zone: ▼

Keyboard layout: ▼

Persistent settings

☐ Play sound when finished

☐ Eject media when finished

☐ Enable telemetry

SAVE

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.

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.

```
stefan:~  
%ping clock.local  
PING clock.local (172.16.1.45): 56 data bytes  
64 bytes from 172.16.1.45: icmp_seq=0 ttl=64 time=41.532 ms  
64 bytes from 172.16.1.45: icmp_seq=1 ttl=64 time=2.146 ms  
64 bytes from 172.16.1.45: icmp_seq=2 ttl=64 time=1.610 ms  
64 bytes from 172.16.1.45: icmp_seq=3 ttl=64 time=1.611 ms  
^C  
--- clock.local ping statistics ---  
4 packets transmitted, 4 packets received, 0.0% packet loss  
round-trip min/avg/max/stddev = 1.610/11.725/41.532/17.211 ms  
stefan:~  
%
```

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 xzf /boot/firmware/alarm-clock_v01.tgz
```

For the Bullseye version of the OS:

```
tar xzf /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.

```
%ssh clock.local
Linux clock 6.1.21-v8+ #1642 SMP PREEMPT Mon Apr  3 17:24:16 BST 2023 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:
stefan@clock:~ $ tar xzf /boot/alarm-clock_v01.tgz
stefan@clock:~ $ sudo ./Install_Clock.sh

Enter the password for 'stefan'

Password:
Enter the password again:

'stefan' will be used as the samba account with the same password

Do you want to enable a firewall that blocks all connections originating from a foreign subnet?
Y/y/N/n: y

Do you want to *attempt* to disable the Pi's power and activity LEDs?
Y/y/N/n: y

-----
As this installation script runs, there should be no errors.
If you spot an error or the script stops, there's a problem.
Script commencing in 03 seconds
```

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, extract 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 fileserver.

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 xzf alarm-clock_v02.tgz
cd alarm-clock
make
make install
```

Configuring the clock via the web interface

To reach the web interface, open a browser and surf to <http://clock.local>, or whatever system name or IP address you used to ssh into the clock. You're greeted by the clock's home page which contains links to the various configuration and media player functions, a summary of the configured alarms, the playlist if media is currently playing, and a list of any other clocks found on the local network if any exist. There may be short pauses as the web page builds out, but even without hardware connected, web pages and links will still work.

6:48 pm Sun 6 Aug
88%

AA

clock.local

🔍

🔄

📶

+

🗪

Raspberry Pi Alarm Clock Version 1.2 (3rd August 2023)

Alarm Configuration

Clock Setup

WiFi Configuration

Media Player

Current Alarm Settings

Today is **Sunday**. The time is **18:48**.

Time	Duration	Alarm Days	Alarm File or Stream	Vol Adj
06:40	20 minutes	Mon Tue Wed Thu Fri	http://www.abc.net.au/res/streaming/audio/aac/classic_fm.pls	-15 to +0
07:00	11 minutes	Mon Tue Wed Thu Fri Sat Sun	http://www.abc.net.au/res/streaming/audio/aac/local_canberra.pls	-4
07:11	49 minutes	Mon Tue Wed Thu Fri Sat Sun	http://www.abc.net.au/res/streaming/audio/aac/classic_fm.pls	---
08:00	6 minutes	Sat Sun	http://www.abc.net.au/res/streaming/audio/aac/local_canberra.pls	-4
08:06	30 minutes	Sat Sun	http://www.abc.net.au/res/streaming/audio/aac/classic_fm.pls	---

Click [here](#) to open a tab with information about using the buttons.

Click [here](#) to open a tab with the schematic diagrams.

Other clocks on this network

This clock type: Silent Clustered

IP address	System Name	Clock type
172.16.1.97	clock2	Audible clustered
172.16.1.54	stefan-pizero2w	Stand alone

Note that the image above shows other clocks that have been discovered on the network. If you only have one clock, this boxed section will not be displayed.

Navigating should be reasonably intuitive. Check out the Clock Setup page to configure your preferences. When you hover the pointer over some options, the browser will display hints and a description.

Creating alarms is a matter of filling in details from top to bottom on the Alarm Configuration page: what time, how long do you want it to run, on what days of the week and should it recur indefinitely. A volume adjustment is relative to the current master volume, adding or subtracting from the current master volume level, which is useful when chaining a sequence of alarms from different sources (where each source has different audio levels), or to slowly increase or decrease volume using a succession of alarms.

Current Alarms

Today is Friday.

Time	Duration	Alarm Days	Alarm File or Stream	Vol Adj	
☐ 06:40	20 minutes	Mon Tue Wed Thu Fri	http://www.abc.net.au/res/streaming/audio/aac/classic_fm.pls	-15 to +0	Copy
☐ 07:00	11 minutes	Mon Tue Wed Thu Fri Sat Sun	http://www.abc.net.au/res/streaming/audio/aac/local_canberra.pls	-4	Copy
☐ 07:11	49 minutes	Mon Tue Wed Thu Fri Sat Sun	http://www.abc.net.au/res/streaming/audio/aac/classic_fm.pls	---	Copy
☐ 08:00	6 minutes	Sat Sun	http://www.abc.net.au/res/streaming/audio/aac/local_canberra.pls	-4	Copy
☐ 08:06	30 minutes	Sat Sun	http://www.abc.net.au/res/streaming/audio/aac/classic_fm.pls	---	Copy

Delete Selected Alarms
Suspend Selected Alarms

Create a new alarm

New alarm time: 07:00 am

Alarm duration: 120 (minutes)

Day or days of the week for the new alarm:

Monday ☒ Alarm ☐ No Alarm
Tuesday ☒ Alarm ☐ No Alarm
Wednesday ☒ Alarm ☐ No Alarm
Thursday ☒ Alarm ☐ No Alarm
Friday ☒ Alarm ☐ No Alarm
Saturday ☒ Alarm ☐ No Alarm
Sunday ☒ Alarm ☐ No Alarm

Alarm type ☐ One-off ☒ Recurring

Volume Adjustment: 0 (added to the system-wide default volume level when this alarm first trips)

Target Adjustment: 0 (if different, volume gradually adjusts to this target as the alarm is playing)

Accept the following defaults OR choose one of the following options, THEN click 'Create new alarm':

a. [Browse for a file, folder or playlist](#)

b. Type a streaming URL OR a path to a media file, folder or playlist OR specify radio:
http://www.abc.net.au/res/streaming/audio/aac/abc_jazz.pls
Click [here](#) to open a tab with links to streaming search sites, or search the net for streaming URLs.

Create new alarm

Return to main page

Anywhere you see a text box for entering media, you can specify either a streaming radio site URL, the full Linux path to a media file or playlist, or you can type the word 'radio' (without the quotes) to power-on and use the external audio input.

b. Type a streaming URL OR a path to a media file, folder or playlist OR specify radio:
http://www.abc.net.au/res/streaming/audio/aac/abc_jazz.pls
Click [here](#) to open a tab with links to streaming search sites, or search the net for streaming URLs.

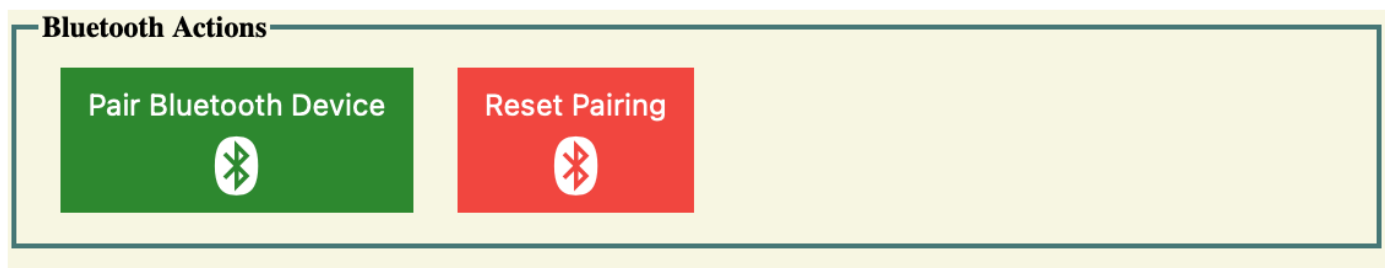
Finding Internet streaming sources is easy: any Internet radio station that is accessible in your country and that can be listened to with a web-browser will work with the clock. However determining the right URL to use with the clock can be tricky. Instructions on how to use a browser to discover the syntax for a streaming URL are included on the clock's in-built web pages.

Using the clock as a Bluetooth speaker

Raspberry Pi models other than the Pi2 (and earlier) support wi-fi and bluetooth, so the alarm clock is able to emulate bluetooth speakers. That means computers, phones and tablets can be paired with the clock in a similar manner to the way they can be paired with car sound systems and commercial bluetooth speakers.

The pairing process is a one-off and you can think of pairing as an initial introduction of the devices to each other. After their 'pairing', devices only need to be 'connected' whenever you'd like to stream podcasts and music via the clock's speakers.

For security reasons, bluetooth interfaces are not permanently pairable so you'll need to temporarily activate this function on the clock. There's a 'Pair Bluetooth Device' icon on the web setup page that will enable discoverability, or you can alternately enable the discoverability mode by simultaneously pressing two of the clock's physical alarm selection switches. When activated, you'll have three minutes to complete pairing after which you'll need to reactivate discovery to pair something else.



You can pair as many devices with the clock as you'd like. The clock will remember the identity of each so when you wish to connect any of them in the future, you won't need to go through the pairing process again – just open the bluetooth settings on the device and ask it to connect with the clock.

When your device is connected, audio will be played through the clock's speakers. Set the volume using either the physical volume up/down switch on the clock, or using a volume slider on one of the clock's web pages. Normally, you'll probably want to set the volume on your computer, phone or tablet to maximum and then use the volume controls on the clock to adjust the level, but the other way around will work too.

If you're not yet experienced with bluetooth, you may find the process of pairing and connecting devices to be less obvious than you'd expect. The two devices that you wish to pair each need to be in pairing / discovery mode at the same time or they won't see each other and you'll need to carefully follow the documented pairing and connection process for your device. If you strike difficulties pairing with the clock and you're new to this process, confirm the procedure by pairing with something such as your car's sound system.

It's not uncommon for bluetooth devices to experience pairing issues after you've asked one of the two devices to forget a previous association, but have not done the same on the other. If this happens and you cannot re-pair your device with the clock, there's a 'Reset Pairing' button on the web setup page which will cause all of the clock's pairings to be forgotten. After using this function, try to re-pair once more.

Devices also experience reconnection issues. A phone or tablet may indicate that it's connected, but won't output any audio. In these cases, use the bluetooth disconnect function on the computer, phone or tablet, and then reconnect it before restarting the audio stream. If that still fails, forget the pairing on both devices and re-pair from scratch.

Why does Bluetooth audio sometimes stutter and why do my devices disconnect?

The clock's bluetooth streaming has been tested with Windows and Mac OSX, Android and iPhones, and it has worked well using either the Pi3 or the Pi4. However there are certain circumstances where it is expected you'll notice occasional audio stutter and bluetooth disconnections.

The most obvious situation is when connected bluetooth devices move beyond their working range. This will happen when moving a phone or tablet outside the room where the clock is located. If it happens, move closer, reconnect the phone or tablet and restart the now-paused audio app.

A less obvious reason for bluetooth stuttering stems from the fact that bluetooth and 2.4 GHz wi-fi share the same spectrum. The Pi 3 doesn't support 5 GHz wi-fi and although the clock won't need to access the Internet often when you're streaming bluetooth audio, you'll certainly notice audio stuttering or disconnection if you access your Pi3 clock's web interface at the same time as you're streaming bluetooth audio to it.

If you're unsure what's happening in the wi-fi spectrum at your place, you can run a wi-fi scan from within an ssh session to the Pi using the following command:

```
sudo iwlist wlan0 scan
```

It will scan all the wi-fi channels and show you what access points are using them. You'll not know which channels your bluetooth devices have paired on, but if you can see that the clock is using a 5 GHz channel, you're on the right track towards eliminating bluetooth interference issues.

The following command will display which wi-fi frequency the Pi is using.

```
iwconfig wlan0
```

Bluetooth channel assignment is normally statically allocated during the pairing process. If your bluetooth channel experiences interference, you can try resetting pairing and starting over. Refer to the magazine article for some additional tips about bluetooth technology and interference from 2.4 GHz wi-fi devices.

About the software

The web serving component of the project is implemented upon a well-known application called Apache and the clock is implemented in a programme called alarm-clock. All of the GPIO activity is implemented in a third programme called pigpiod (the Pi GPIO daemon). There are other support functions such as the network time keeping implemented in something called chrony, and the Linux operating system itself has many other tasks of its own such as managing the wi-fi, samba share, usb ports, other network functions and looking for time zone daylight savings updates, all of this done through what ultimately is dozens of different programmes and modules that seemingly run simultaneously.

When you ssh to the Raspberry Pi and type the following command, it displays information about each of the CPU cores within the Pi's ARM microprocessor.

```
cat /proc/cpuinfo
```

For most Linux Pis, you'll see four different cores meaning the hardware supports four simultaneous programmes running 'at full CPU speed' across its four cores. If a programme supports multi-threading, different parts of the same programme can be running at precisely the same time and if you can arrange its different tasks to take advantage of this, the programme will run faster overall than it could using just a single CPU core.

Apache is an example of a multi-threaded programme and can connect with many different browsers at exactly the same time. If you try, you'll see the clock supports this too and the fair test is to launch multiple browsers to the clock's web page simultaneously.

The alarm-clock programme is written in multi-threaded C. It integrates the media player functions by incorporating the open sourced 'mpv' media player libraries. For this reason alone, multi-threading was worthwhile and because there are many other things the clock needs to do, multi-threading seemed to be a good fit. Each of the threads are somewhat like different programmes, but you can also think of them as different subroutines running on (potentially) different CPU cores at (potentially) the same time. The sequencing of clock software events can therefore change depending upon what other things the Linux operating system is running on the different CPU cores alongside the clock. Those that are

interested can explore the source code, but you can be a spectator without looking at any code at all and just watch it play out as it runs.

Testing the hardware: you'll need to enable software debugging mode

When testing the hardware to ensure the switches are being recognised and ambient light levels are being measured correctly, you'll need to use the clock's software debugging mode to view the debugging log. This means you'll need to temporarily stop the operating system from managing the clock. You can also explore the software on its own, without hardware, as soon as you have a Raspberry Pi with clock software up and running.

By default, alarm-clock is configured to run automatically at boot time and if it ever crashes or detects a problem of its own accord, the operating system will restart it. Normally its output will be hidden. To reveal and watch the log, run the alarm-clock programme from within an ssh session and enable debugging.

```
stefan:~  
%stop-alarm-clock  
*****  
Commands have been issued to stop the alarm-clock software.  
  
If you have a web page opened to the alarm-clock while it is  
stopped, the crash detection algorithm is liable to reboot the  
Pi to restart the clock.  
  
To avoid this, either run alarm-clock on the command line  
specifying any options you would like, such as:  
    sudo alarm-clock -V  
  
or close all open web pages to the clock NOW!  
*****  
stefan:~  
%
```

To take a peek from within an ssh session, temporarily stop the operating system managing the alarm-clock by issuing the command 'stop-alarm-clock'.

```

stefan:~
%alarm-clock -h
Usage: alarm-clock {-acdjhllmpqrvxyzVABCDMSTWXYZ}
-a          Display current ambient light measurement, volume and remaining playing time (in minutes).
-c [cmd]    Send command to the media player.
            cmd must be one of 'stop', 'pause', 'next', 'prev', 'restart', 'seek-fwd', 'seek-back',
            'clear', 'playlist', 'shuffle', 'clocks'.
-d          Daemonise this process (run in the background and don't die when shell closes).
-j NUM      Jump to the specified track number in currently playing playlist.
-h          Display this usage information.
-l          Display copyleft information.
-L [filename] Save logging output to the specified filename.
-m 'media'  Append media path/URL to the current playlist. Quote strings containing special
            characters. Note: -m must be used in conjunction with -t
-p fraction  Temporarily set PWM ratio to fraction between 0.0 and 1.0.
-q          Query the software version.
-r          Reload configuration from '/etc/alarm-clock/setup.conf'
            and alarms from '/etc/alarm-clock/alarms.csv'.
-t duration  Specify a time duration (in minutes) to play media stream specified with -m.
-v volume    Set volume to specified level. 0=silent, 31=maximum.
-x          Stop the alarm-clock daemon.
-y          Initiate a brightness test to check for uniform segment brightness.
-z          Initiate a display test to check for broken LED segments.

-V          Be verbose. Enable ALL the debug and trace information listed below.
-A          Display debug information from the amp manager thread.
-B          Display debug information from the button manager thread.
-C          Display debug information from the clock thread.
-D          Display debug information from the display driver thread.
-M          Display debug information from the media player thread.
-S          Display debug information from the setup manager thread.
-T          Display debug information from the system timer thread.
-U          Display debug information from the bluetooth threads.
-W          Display debug information from the multicast threads.
-X          Display debug information from the display dimmer thread.
-Y          Display debug information about alarm setting and tripping.
-Z          Display reduced debug information from the the media and amp manager threads.

(This is version 1.2 (3rd August 2023))
stefan:~
%

```

Execute the alarm-clock programme by entering 'sudo alarm-clock', or 'alarm-clock -h' to display command line option help.

```
sudo alarm-clock -h
```

Choose one or more options or type 'sudo alarm-clock -V' to enable absolutely everything, and watch. As it runs, messages and time stamps from different alarm-clock threads will intermix with each other in different ways and with different interval spacing every time you restart the programme, to the whims of the randomness of whatever the Operating System is doing at the time. The Pi will revert to automatic clock management the next time you reboot.

```

stefan~
%sudo alarm-clock -V
07/08/2023 13:35:20.175: Commencing initialisation
07/08/2023 13:35:20.204: pigpiod PID: 1397
07/08/2023 13:35:20.204: Attempting to start the bluetoothctl process
07/08/2023 13:35:20.216: Bluetoothctl PID: 1449
07/08/2023 13:35:20.217: My networking details are [stefan-pizero2w / 172.16.1.54]
07/08/2023 13:35:20.218: Launching threads!
07/08/2023 13:35:20.229: Threads launched!
07/08/2023 13:35:20.230: Digital Alarm Clock V1.2 (3rd August 2023). Setup verbosity=0x800, Command line verbosity=0xcff
07/08/2023 13:35:20.230: Running on Raspberry Pi Zero 2 W Rev 1.0 [0.410 GB RAM available to Linux]
07/08/2023 13:35:20.253: Architecture=aarch64, kernel=6.1.21-v8+
07/08/2023 13:35:22.937: mpv 0.32.0 Copyright © 2000-2020 mpv/MPlayer/mplayer2 projects
07/08/2023 13:35:22.956: This PID: 1448
07/08/2023 13:35:22.956: Real UID: 0, Effective UID: 0
07/08/2023 13:35:22.957: Real GID: 0, Effective GID: 0
07/08/2023 13:35:22.957: File creation mask: 664 at launch. Now set to 111.
07/08/2023 13:35:22.957:
07/08/2023 13:35:22.957: Reloaded setup:
07/08/2023 13:35:22.957:   Clock type: S
07/08/2023 13:35:22.957:   Default 12 or 24H display: 12
07/08/2023 13:35:22.957:   Leading zero blanking: N
07/08/2023 13:35:22.957:   Default alarm time: 07:00
07/08/2023 13:35:22.958:   Default alarm duration: 120 (minutes)
07/08/2023 13:35:22.958:   Default media duration: 60 (minutes)
07/08/2023 13:35:22.958:   Default snooze duration: 600 (seconds)
07/08/2023 13:35:22.958:   Second snooze press ignore: N
07/08/2023 13:35:22.958:   Default volume: 12
07/08/2023 13:35:22.958:   Default stream or file: 'radio'
07/08/2023 13:35:22.958:   Fallback alarm file: '/media'
07/08/2023 13:35:22.958:   Default minimum ambient light threshold: 300
07/08/2023 13:35:22.959:   Default maximum ambient light threshold: 1000
07/08/2023 13:35:22.959:   Minimum PWM ratio: 0.03
07/08/2023 13:35:22.959:   Maximum PWM ratio: 1.00
07/08/2023 13:35:22.959:   Command line verbose bitmask: 0xCFF
07/08/2023 13:35:22.959:
07/08/2023 13:35:22.959: Cannot open alarm file and determine its size, or file '/etc/alarm-clock/alarms.csv' is empty: No such file or directory
07/08/2023 13:35:22.960: Setup Loader Thread initialised
07/08/2023 13:35:22.960: Media Manager Thread initialised
07/08/2023 13:35:22.961: Button Manager Thread initialised
07/08/2023 13:35:22.962: Bluetooth Rx Thread: Waiting to connect to bluetoothd...
07/08/2023 13:35:22.962: Display Driver Thread initialised
07/08/2023 13:35:22.963: Clock Thread initialised
07/08/2023 13:35:22.963: Clock Thread received tick
07/08/2023 13:35:22.963: Display Driver Thread received message: Type=0, H=13, M=35, S=22, dots=0x00
07/08/2023 13:35:22.964: LED Dimmer Thread initialised
07/08/2023 13:35:22.964: Multicast Tx Thread sent 'H,S,0,1,stefan-pizero2w,172.16.1.54'
07/08/2023 13:35:22.964: Multicast Rx Thread received 42 bytes 'H,C,21112ff2,1691286729,clock2,172.16.1.97' from 172.16.1.97
07/08/2023 13:35:22.964: Multicast Rx Thread ignoring 35 byte msg from self
07/08/2023 13:35:22.964: Amp Manager Thread initialised
07/08/2023 13:35:22.963: Bluetooth Rx Thread: [bluetooth]#
07/08/2023 13:35:22.964: Media Player Thread initialised
07/08/2023 13:35:22.964: Timer Thread initialised
07/08/2023 13:35:22.965: Bluetooth Rx Thread: Agent registered
07/08/2023 13:35:22.966: Bluetooth Rx Thread: [bluetooth]#
07/08/2023 13:35:22.966: Bluetooth Rx Thread: [CHG] Controller B8:27:EB:0D:9F:6A Pairable: yes
07/08/2023 13:35:23.000: Clock Thread received tick
07/08/2023 13:35:23.000: Display Driver Thread received message: Type=0, H=13, M=35, S=23, dots=0x00
07/08/2023 13:35:23.210: Timer Thread: A to D instantaneous=4095, Moving Average=2191.3, PWM ratio=1.000, time to next alarm=-1s
07/08/2023 13:35:23.210: LED Dimmer Thread received message: type=0x00, Duration=0, PWM_ratio=1.000, alternate PWM freq=0

```

If required, there's a setting that enables logging to a file for catching issues when you're not around to look. The clock's setup file is located in '/etc/alarm-clock/setup.conf' and you can edit it to enable file logging as follows:

```
sudo nano /etc/alarm-clock/setup.conf
```

Look for the two following lines and remove the '#' symbol in the first column to uncomment them. Set the 'VERBOSE' symbol to reflect the logging you're interested in by forming a hex number of the logging option bitmask described in the file. Up to ten log files will be kept, creating a fresh log each time the programme restarts.

```
#VERBOSE=0x090
#LOG_FILE="/var/log/alarm-clock.log"
```

For more information about the configuration, compiling and playing with the code, consult the readme files in the alarm-clock directory tree you extracted from the tar-ball. There are some example programmes I knocked up to help me as I wrote the code but I've left them there if you're interested to explore.

Checking software health in general

If you're new to linux, you're probably not familiar with all of the commands for checking on how things are running. You'll find a couple of basic commands useful. To check that the alarm-clock programme is running, issue the following command. If its running, you'll see a response showing how much CPU time it has used so far. If you see nothing, its not running.

```
ps -A | grep alarm
```

Similarly, you can check for a running pigpio daemon:

```
ps -A | grep pigpiod
```

And the bluetooth control process and its daemon (there should be two items displayed):

```
ps -A | grep bluetooth
```

And the web server and its workers waiting to receive connections (there should be several items displayed):

```
ps -A | grep apache
```

To see the full list of all the running processes along with their memory consumption, type 'top' to start a self-refreshing display (and type the letter 'q' to exit when you're done):

```
top
```

The 'top' command has lots of useful information including how much CPU time each process is currently consuming expressed as a percentage of a single CPU core.

If a process is using more than one of the Pi's CPU cores and maxing them out, you will even see percentages greater than 100% however you should never see such large utilisation on the clock. As there are four CPU cores on most Pis, when the sum of all current CPU utilisation reaches 400%, the Pi will be fully maxed out. The largest CPU hog is the pigpio daemon.

The simple commands above are just a start and will tell you enough to know things are working. There are many more than these which you can search for on the Internet.

The philosophy behind the digital volume control

When I describe the philosophy behind the way the volume control works, it may seem obvious. But it hasn't been obvious to everybody that's used this clock and media player so I will describe it to be sure.

The up/down volume control buttons work just like a volume control knob would work on the kinds of device you are already very familiar with. Putting volume up or down a couple of steps will make it louder and softer just as you would expect.

The clock will remember the last volume setting even after a power outage. This too is just like any other device with a rotating volume control knob. It also means that if you turn the volume down while an alarm is playing, the next time an alarm starts, the volume will be down at the level you last left it.

On the alarm settings pages, there's a 'volume adjustment' setting that can be applied to each alarm. Normally, I imagine you'll want to set an adjustment of zero. However if you are chaining alarms from different streaming sources, you may find one source to have a significantly different level than another. The volume adjustment can be used to equalise the levels of the chained sources.

Whenever you adjust the system-wide volume, the volume of all future alarms will adjust accordingly, exactly as would happen if you rotated the volume control knob on your traditional alarm clock and waited for the next alarm to trip. If you've specified any non-zero volume adjustments for any alarm, those adjustments will be added to the new system-wide volume to maintain the volume relativity.

The Clock Setup page also has a setting for the 'minimum alarm volume'. In the case that you've set a low system-wide volume, the minimum alarm volume will ensure your alarms will not be quieter than this. Adjustments still apply to the minimum alarm volume to maintain relativity between different alarm levels, and to ensure adjustments continue to work as you expect: but you won't be able to accidentally set the system-wide volume low and then sleep through an alarm.

Waking up gently. Varying the volume.

You can use alarm chaining to be awoken 'gently' by setting the volume adjustment low on the initial alarm, and then chaining a series of subsequent alarms with progressively increasing volume adjustments. But there's a better feature for achieving the same thing.

Each alarm includes a 'target adjustment' setting as well as the 'volume adjustment' setting. The volume adjustment is applied when the alarm first trips. Each minute thereafter, the digital volume control will make one step towards the target until the target volume is reached (or the next chained alarm trips and causes its own adjustment settings to be applied on top of the previous alarm's settings). So in order to gently increase volume when an alarm is tripped, you could set the

‘volume adjustment’ to -15, and the ‘target adjustment’ to +0 as shown in the example above, and the alarm will start quietly and gradually get louder until it reaches its normal level with no (i.e. +0) adjustment.

The volume adjustments are added to the current system-wide volume setting and could therefore go beyond range. If the sum of the system-wide volume and the adjustment falls outside 0 .. 31 (which are the digital volume control’s limits), then the volume will be pegged at 0 or 31.

Remembering the position within a playlist

You can use an mp3 playlist from your own media library for your alarm and even chain alarms between your local playlist, then to the radio or Internet stream, and back to the original playlist after the news is over.

When you initially create an alarm and specify a directory as an alarm source, the clock’s web server will build a playlist file of the underlying directory tree. Similarly, a playlist is created if you use the media player function and specify a directory.

When playing a playlist, the clock remembers the last track and will recommence from the following track when it uses the same playlist again. You can therefore chain from the playlist to the news and back again, and the playlist will continue from the track after where it left off. The playlist memory will not record the offset within the track that was interrupted: it recommences from the beginning of the following track.

Two or more clocks playing audio at the same time

If you build more than one clock as I did and position them within earshot of each other such as on either side of your bed, you’ll sometimes notice that streaming audio is not precisely time-aligned. Although different clocks will maintain the same time within an accuracy of a millisecond or so, if they launch an Internet stream, they will not necessarily time-align. If the time difference is very slight, you’ll not notice however if its more than around ten milliseconds, you probably will. For this reason, there’s two modes for the clustered clock configuration: audible and silent.

In the audible clustered mode, the clock will activate its audio output when an alarm is tripped. In the silent clustered mode, the clock will share alarm and button information with other clustered clocks, but will not sound out an alarm when the alarm trips, avoiding situations where one clock is playing audio slightly out of phase with the other.

In my case, I’ve positioned two clocks on either side of the bed. One is audible and the other is silent. I’ve positioned the two speakers from the audible clock on either side of the bed so we get a good stereo effect. From the perspective of using the clock’s switches, you can’t tell which of the two clocks is the silent one and which is the audible one: because they are clustered, they share their alarm configuration and button presses and function as if they are the same clock.

If you build a free-to-air radio into your clocks and plan to use this as your alarm source, the audio from the two radios will align perfectly every time and you won’t need to worry about making one of the clocks silent.

Starting the media player when a Bluetooth device is connected

You can connect a bluetooth device to the clock at any time and stream audio from your device. The bluetooth device will remain connected to the clock even when the audio stream stops so you can play another stream without having to reconnect your device again.

However if at the same time you are streaming bluetooth audio you start a media player stream on the clock, including when an alarm trips and plays a media source other than the radio, the clock will disconnect from any currently connected bluetooth devices so that it can play the media stream correctly. Afterwards, you’ll need to reconnect your bluetooth device to the clock in order to stream using bluetooth once more.

12 and 24 hour time display

The setup web page includes a configuration option to choose between 12 hour and 24 hour times. This refers to the LED display on the clock itself and does not refer to the way times are displayed on the web pages. It turns out that the times displayed in most web browsers are formatted according to the locale setting on the device running the web browser. This

means that although you may have configured one particular format on the web settings screen, your browser might steadfastly display a different format. That's OK – its just the way that web browsers work. The LEDs on the clock will correctly reflect your preference.

Hardware versions of the Pi that have been tested

Because I don't own every variant of the Pi, I've not been able to verify that the software installs and works correctly on every variant. However I've installed and tested the hardware with a couple of Pi3s, Pi4s, and a PiZero2W. I also installed and unsuccessfully tried to run the software on one of the early Pi Model Bs.

As the article was being prepared for publication, a new model Pi (the Pi5) was announced along with a new operating system release called Bookworm. Using the Pi5 is not recommended for the following reasons:

- Although the Pi5 is faster than any of the earlier hardware platforms, the additional speed is not necessary for this clock.
- The Pi5 does not support an analogue audio output.
- The lead author of the pigpio library has indicated that it may be some time before the pigpio library supports the Pi5. That's because the development team are not only waiting for the new hardware to be released, but the documentation for the new chips that will drive the GPIO pins. Its not clear when a pigpio library that supports the Pi5 will be available.

Other than on the very old Model B, the installer and the alarm-clock programme ran fine on the Pi3, Pi4 and Pi Zero 2W devices I have available to me, including working with the hardware design as published. The Model B processor is slow and could not keep up, and it crashed.

In terms of performance, the speed difference is very noticeable between the different models of Pi, most so when accessing the in-built web pages. The Pi4 is obviously the fastest, followed by the Pi3 and then PiZero2W. I suspect that the Pi2 and PiZeroW will be noticeably slower again.

If you would like to stream media, using either a Pi3 or Pi4 is recommended. If you're instead planning to only use the external audio input from an external audio device, the Pi2 or PiZero models should support your needs, or the Pi5 once the pigpiod library is released for this platform.