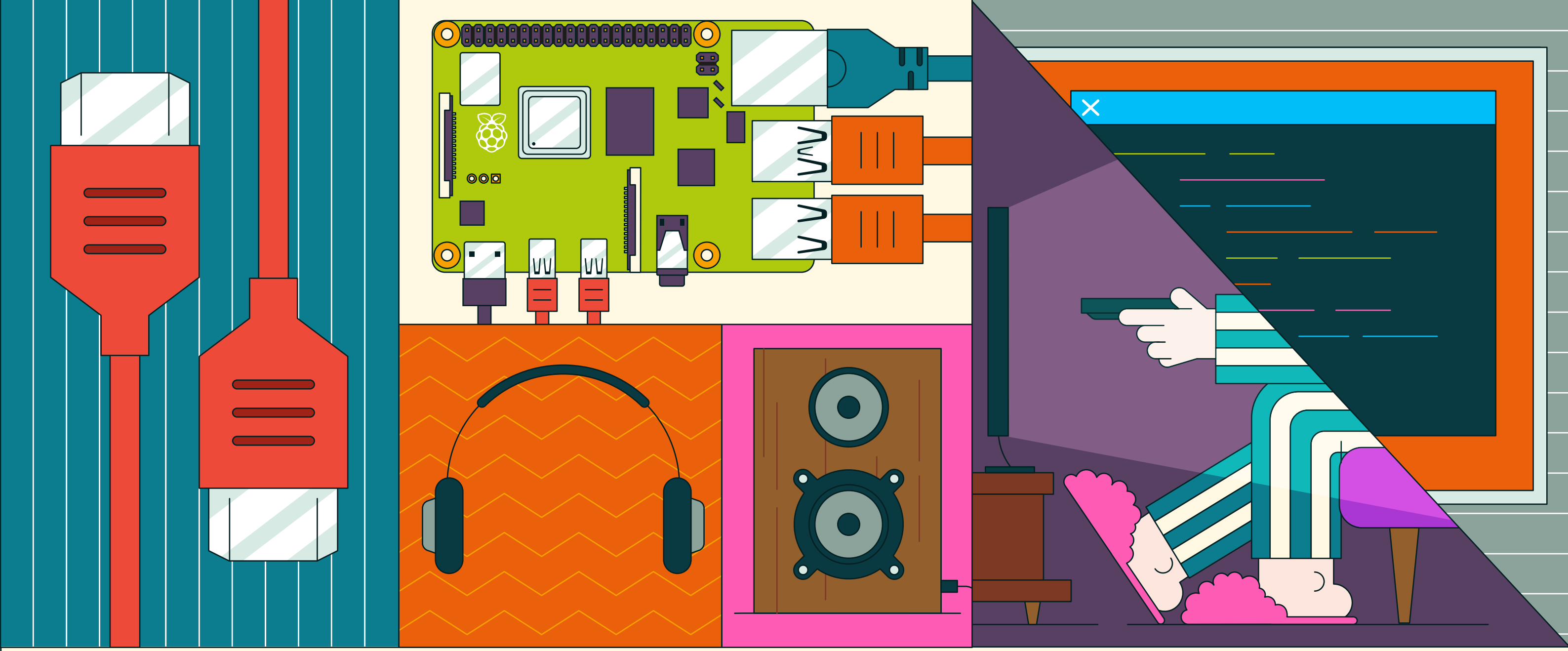
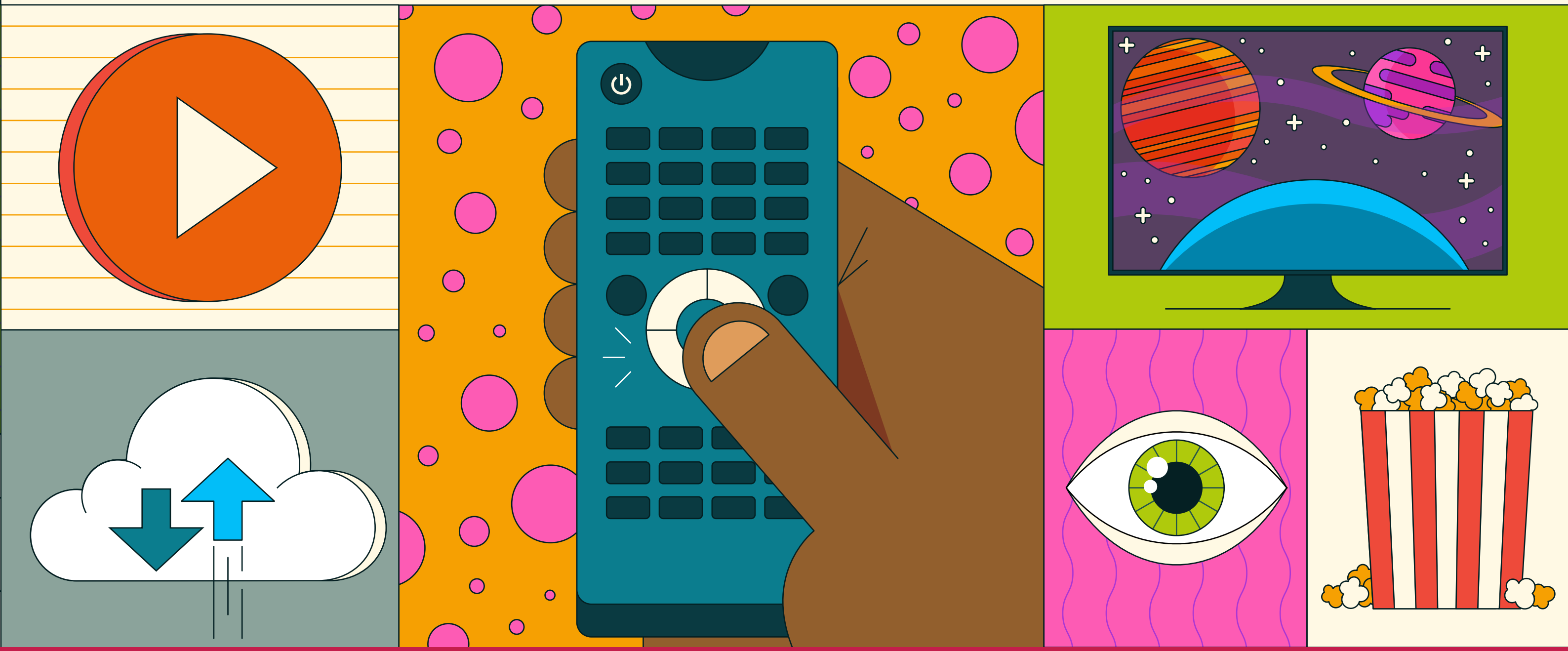
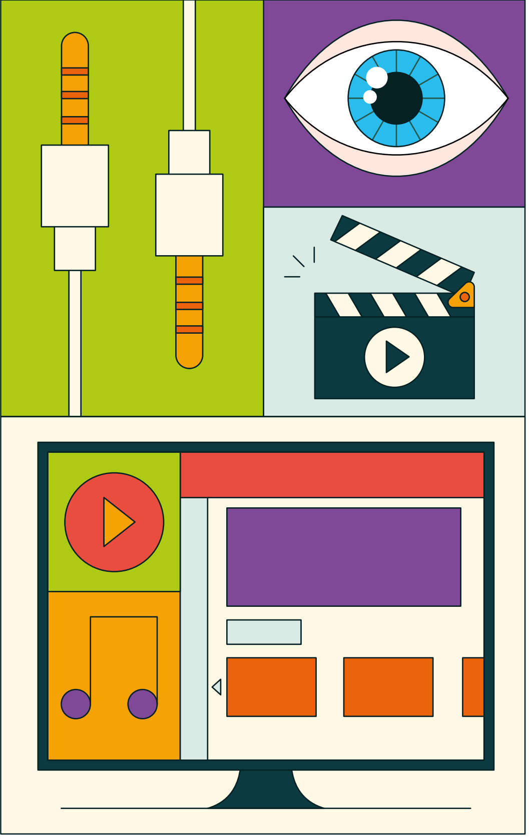


BUILD A RASPBERRY PI MEDIA PLAYER

Power up your TV and music system



FROM THE MAKERS OF The MagPi THE OFFICIAL RASPBERRY PI MAGAZINE



WELCOME!

Thank you for picking up this copy of *Build a Digital Media Player*. This is an exciting Raspberry Pi project that's fun and functional.

Raspberry Pi is small computer that can be picked up from just $15. Add some simple software, and a remote control, and you will build a digital media box that puts shop-bought solutions to shame.

For starters, it's cheaper to make your own media player, but more importantly, you have complete freedom to play whatever you want. Play your own movie files, music and photographs alongside streaming video and music from the big services.

In this book, we have collected together some of the best articles from The MagPi magazine written on this hugely popular subject. We'll show you how to build a media centre, create a home server that stores all your video files, make your own music system and all the best kit and projects to look at.

Happy making and enjoy your personal media player.

Lucy Hattersley

FIND US ONLINE magpi.cc **GET IN TOUCH** magpi@raspberrypi.com

The MagPi

Available on the App Store
GET IT ON Google Play

EDITORIAL
Editor: **Lucy Hattersley**
Features Editor: **Rob Zwetsloot**
Contributor: **PJ Evans**

DISTRIBUTION
Seymour Distribution Ltd
2 East Poultry Ave, London,
EC1A 9PT | **+44 (0)207 429 4000**

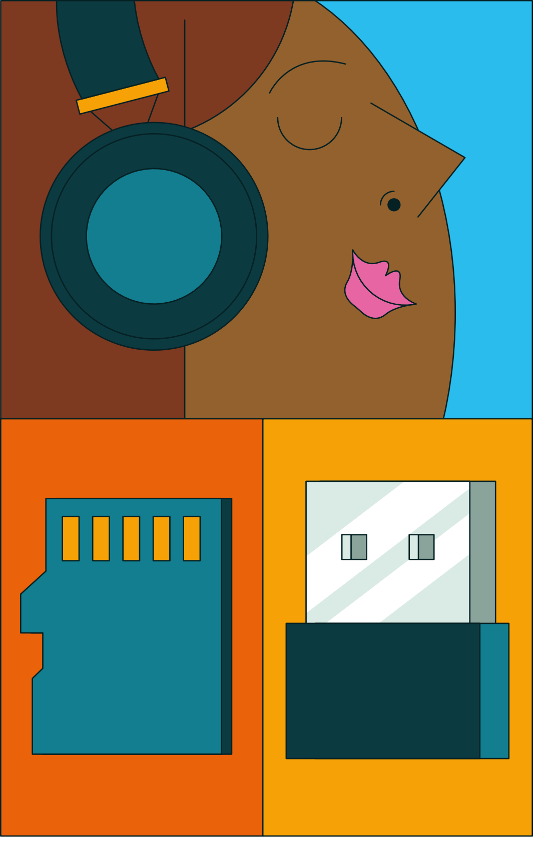
DESIGN
Designers: **Natalie Turner** and **Sara Parodi**
Illustrator: **Sam Alder**

MAGAZINE SUBSCRIPTIONS
Unit 6, The Enterprise Centre, Kelvin Lane,
Manor Royal, Crawley, West Sussex,
RH10 9PE | **+44 (0)207 429 4000**
magpi.cc/subscribe
magpi@subscriptionhelpline.co.uk

PUBLISHING
Publishing Director: **Russell Barnes**
russell@raspberrypi.com

Advertising: **Charlotte Milligan**
charlotte.milligan@raspberrypi.com
Tel: +44 (0)7725 368887

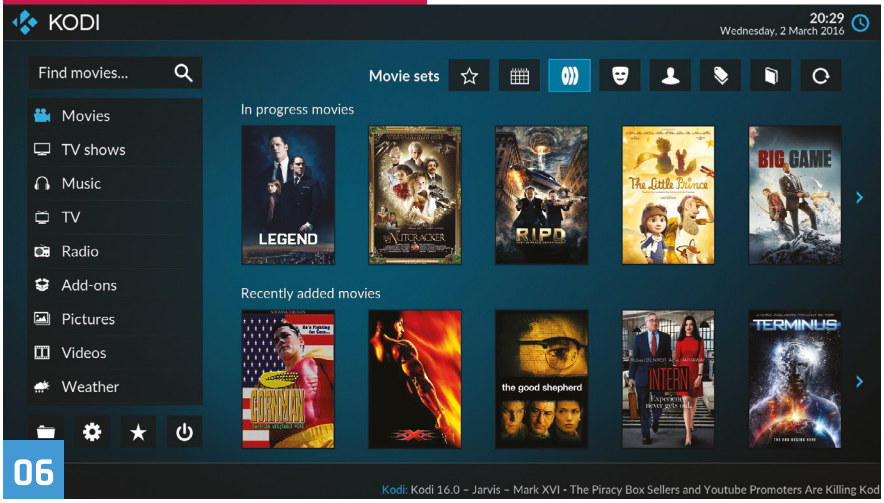
Director of Communications: **Liz Upton**
CEO: **Eben Upton**



Contents

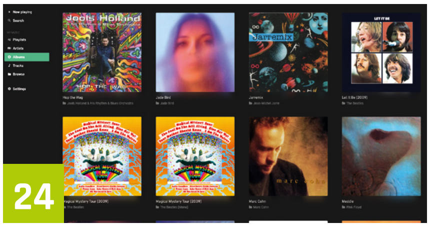
Build the ultimate media centre

12



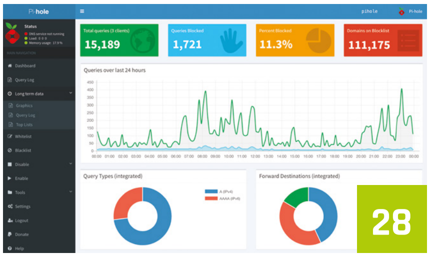
Kodi: Kodi 16.0 – Jarvis – Mark XVI - The Piracy Box Sellers and Youtube Promoters Are Killing Kod

06

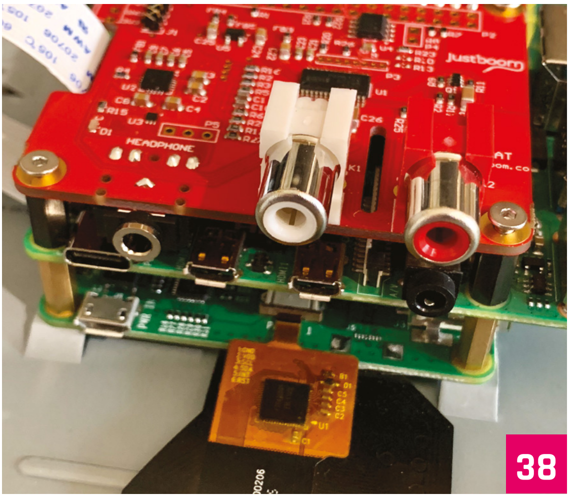


24

Build the ultimate home server

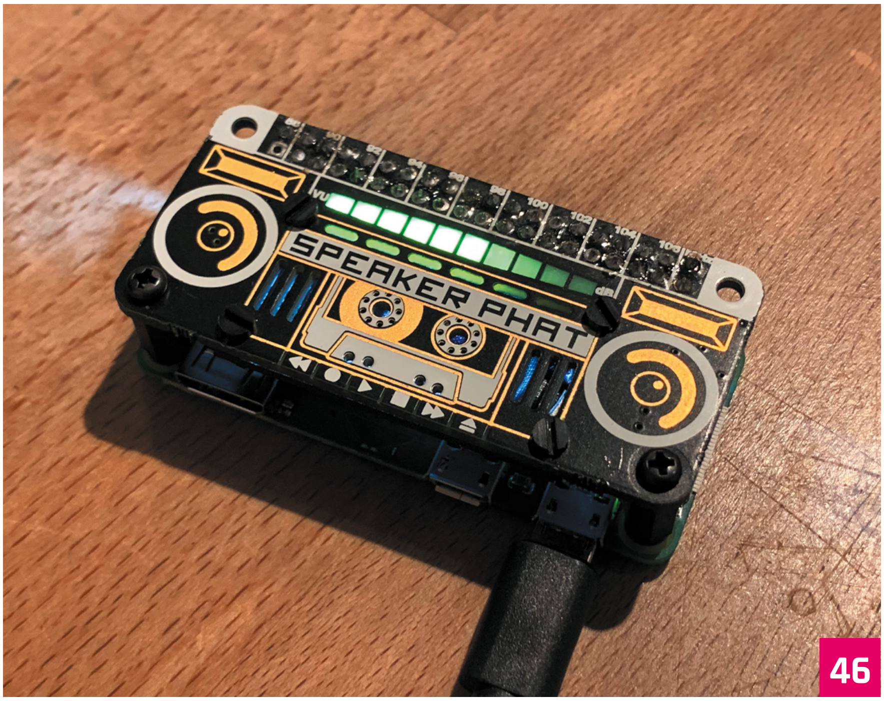


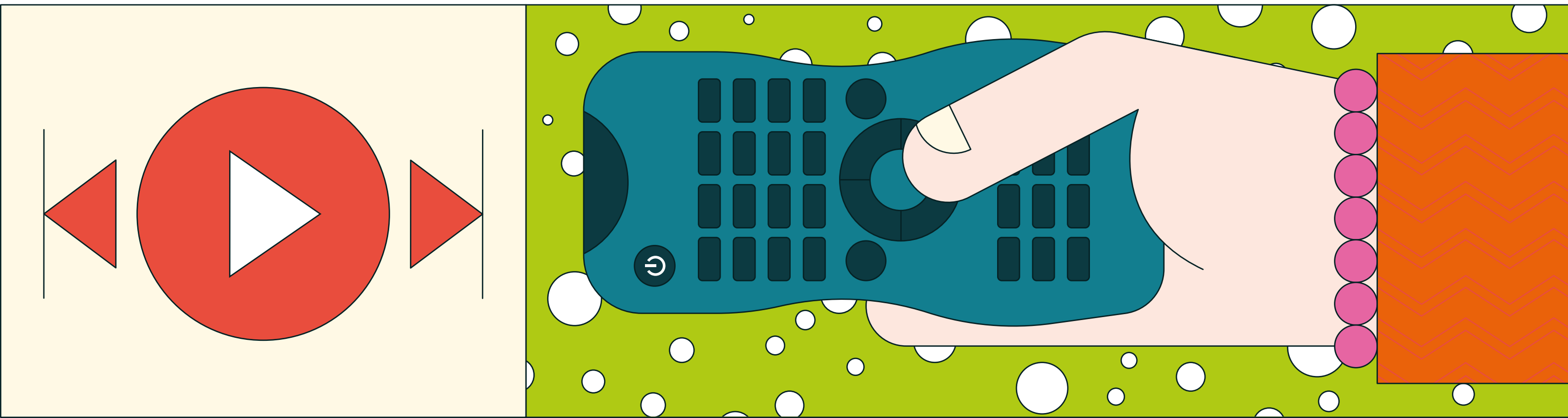
28



Build the ultimate music system

Reviews





BUILD THE ULTIMATE MEDIA CENTRE

Create the ultimate media box using just a Raspberry Pi

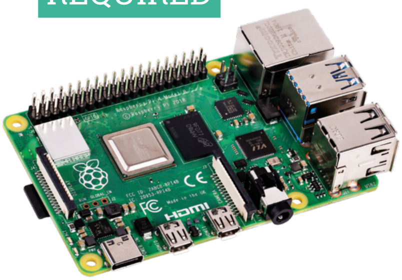
Raspberry Pi is capable of many things, and we've built a lot of projects with one. Having several Raspberry Pi computers performing similar functions seemed a littler redundant to us, though, which gave us a brainwave – what if we could combine multiple uses into one amazing project?

Thus we decided to see how many media services we could squeeze into just one Raspberry Pi 4. Luckily, the majority of the tools are already available – you just need to know how to combine them all. Get ready for the perfected media experience.

Parts list

Everything you need to put together your ultimate media centre

REQUIRED



▲ **Raspberry Pi 4**

£34 / $35 | **magpi.cc/raspberrypi4**

▼ **Case (Argon One M.2)**

£42 / $45 | **magpi.cc/argononem2**

▲ **Mini Remote Control**

£5 / $6 | **magpi.cc/miniremote**

▼ **Official Raspberry Pi 4 power supply**

£8 / $10 | **magpi.cc/psu4**

▲ **HDMI cable**

▼ **Game controllers**

▲ **Flirc USB**

£20 / $23 | **magpi.cc/flircusb**

◀ **16GB microSD card**

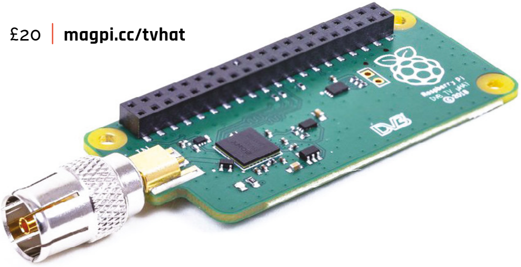
OPTIONAL

▲ **M.2 storage drive**

▶ **Ethernet cable**

▼ **Raspberry Pi TV HAT**

£20 | **magpi.cc/tvhat**



◀ **Wireless keyboard**

▶ **Removable USB storage**

Set up your system

Build your media PC and install RetroPie

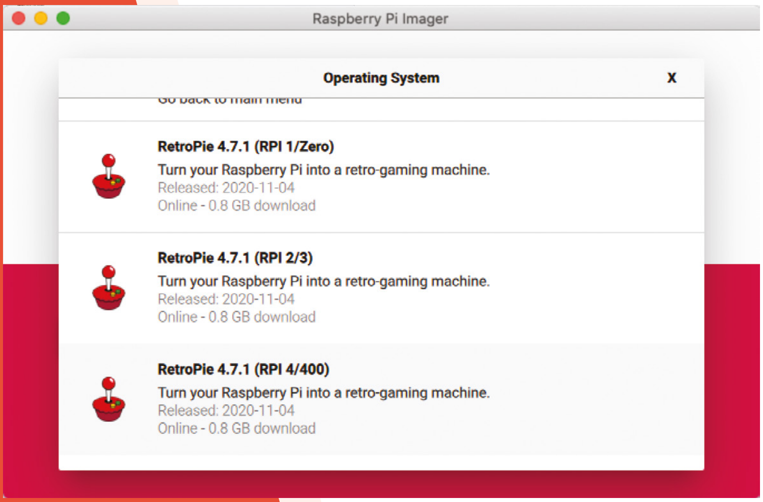
PUT IT TOGETHER

01 Install RetroPie

We're using RetroPie. This software allows you to emulate several game systems, as well as installing Kodi under it for video and music. You'll need to burn your microSD card manually for this – we advise using Raspberry Pi Imager (download it from **magpi.cc/imager**). It's easy to use: click 'Choose OS', select RetroPie, then choose the 'RPI 4/400' version for Raspberry 4 and 400. Then click 'Choose SD card', select your microSD card. Finally, click 'Write'.

02 Install Raspberry Pi

We're using the Argon One M.2 case for a few reasons – it's secure, it keeps your Raspberry Pi cool, it can use a Nanosound DAC if that's your thing, and it routes all the inputs to the rear of the case. Because it has all these bells and whistles, it does take an extra step to install. The instruction manual that comes with the Argon One shows you how to add the daughterboard to Raspberry Pi and then install it into the case with some screws. Follow these instructions, and keep the power button pins in the default position.



RetroPie 4.7.1 (RPI 1/Zero)
Turn your Raspberry Pi into a retro-gaming machine.
Released: 2020-11-04
Online - 0.8 GB download

RetroPie 4.7.1 (RPI 2/3)
Turn your Raspberry Pi into a retro-gaming machine.
Released: 2020-11-04
Online - 0.8 GB download

RetroPie 4.7.1 (RPI 4/400)
Turn your Raspberry Pi into a retro-gaming machine.
Released: 2020-11-04
Online - 0.8 GB download

▲ Raspberry Pi Imager makes it easy to install RetroPie – choose the RPi 4/400 version

03 Hook it up

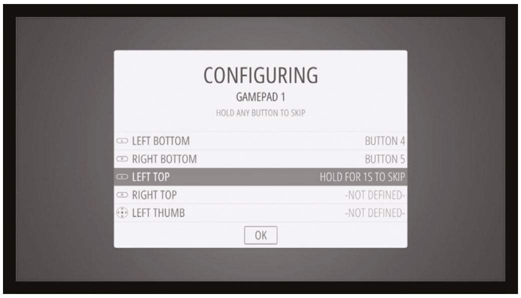
For first-time setup, we recommend connecting a USB keyboard, along with having it connected to a monitor instead of a TV for comfort – however, connecting it to your selected TV is also fine. If you also plan to use a wired game controller, connecting it now is also a good idea. The last thing to connect should be the power supply.

If you want to add an M.2 SSD drive for increased internal storage, do so now!

INITIAL SETUP

01 First boot

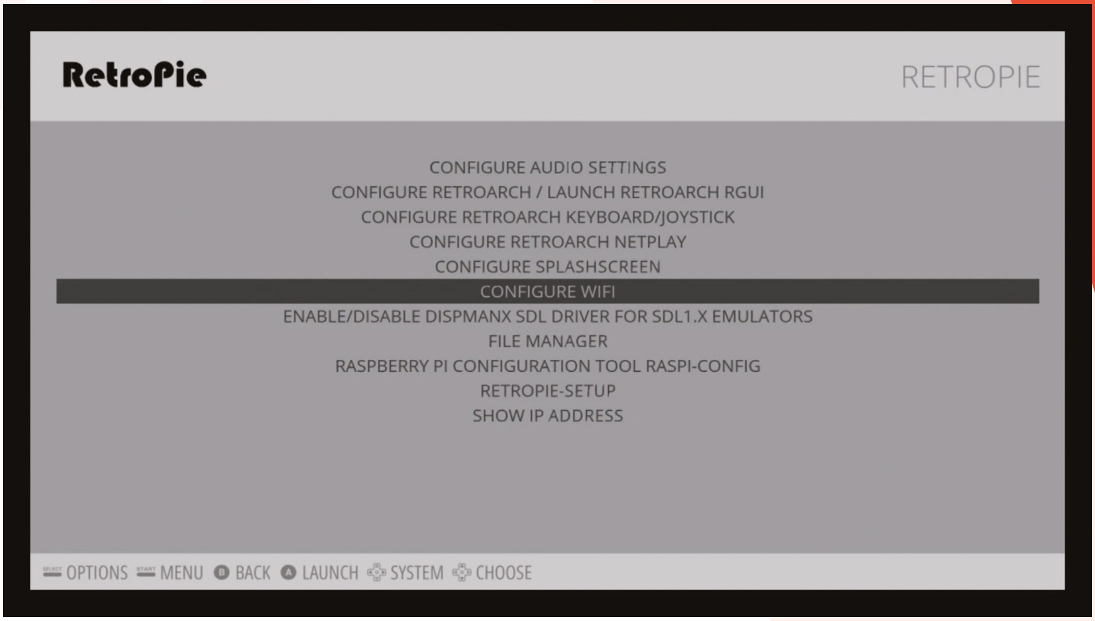
With a power supply plugged in and switched on, press the power button on the back of the case to boot up your Raspberry Pi. RetroPie will do some initial automated setup stuff, then ask you to configure your games controller



LEFT BOTTOM BUTTON 4
RIGHT BOTTOM BUTTON 5
LEFT TOP HOLD FOR 1S TO SKIP
RIGHT TOP -NOT DEFINED-
LEFT THUMB -NOT DEFINED-

▲ Press the relevant buttons to configure your games controller; hold any button to skip a function

For troubleshooting and tips on specific controllers, especially for Sony, Microsoft, and Nintendo consoles, look at the docs: **retropie.org.uk/docs**


RetroPie — RETROPIE

CONFIGURE AUDIO SETTINGS
CONFIGURE RETROARCH / LAUNCH RETROARCH RGUI
CONFIGURE RETROARCH KEYBOARD/JOYSTICK
CONFIGURE RETROARCH NETPLAY
CONFIGURE SPLASHSCREEN
CONFIGURE WIFI
ENABLE/DISABLE DISPMANX SDL DRIVER FOR SDL1.X EMULATORS
FILE MANAGER
RASPBERRY PI CONFIGURATION TOOL RASPI-CONFIG
RETROPIE-SETUP
SHOW IP ADDRESS

▲ Select the Configure WiFi option from the RetroPie menu and enter your network details

buttons. If you don't have a controller connected yet, you can press **F4** on the keyboard to get to the main menu. If you've run out of buttons to assign, hold down any button and it will skip the function. Keep doing this until you get to the end, and confirm 'OK' to finish.

If you're using a wired network and controllers, you can stop here! Just add ROMs over the network or via a USB stick to play games!

> Make sure you turn the controller on just after the reboot

02 Wireless connection

If you don't intend to use an Ethernet connection, you can connect to your wireless network in the RetroPie menu. Select 'Configure WiFi' and it will open a text menu. Select 'Connect to WiFi' and choose your network from the list. Enter your password and hit OK – you may need to wait a moment or two, but it should then be fully connected.

To transfer ROMs over the network, go to **\\RETROPIE** in the Windows File Explorer, or **smb://retropie** on Mac, and open up the **roms** folder

Stream from Steam

RetroPie allows you to use Steam Link software to stream games from a gaming PC straight to your TV! To do this, go to the RetroPie menu and then RetroPie Setup. Go down to Manage Packages and choose Experimental Packages from it. In that list will be 'steamlink' – install it and it will then appear in the main games menu, ready for you to start streaming from a gaming PC running Steam with Remote Play on.

03 Bluetooth controllers

Press **F4** and you'll open the command line. From there, make sure Bluetooth is installed using the command:

```
sudo apt install pi-bluetooth
```

Type **exit** to return to the graphical interface. Go to the menu and select Bluetooth Configuration. Select Register and Connect Bluetooth Devices while your Bluetooth controller is in pairing mode, then select it from the screen and follow the pairing instructions. Once connected, you may need to reboot your Raspberry Pi before configuring the buttons – make sure you turn the controller on just after the reboot.

Install Argon ONE software

To install the fan control to the system, press **F4** to get to the command line, and enter the following:

```
curl https://download.argon40.com/argon1.sh | bash
```

Configure Kodi

Go beyond games to stream your own media files over a network

01 Install Kodi

Kodi is our preferred media centre software for watching video and playing music, and you can install it to RetroPie. Once again from the RetroPie menu, go to RetroPie Setup and then Manage Packages, Optional Packages. From there you'll be able to install Kodi – although it may take a while. Once installed, you'll find it in the Ports menu as an option after a reboot.

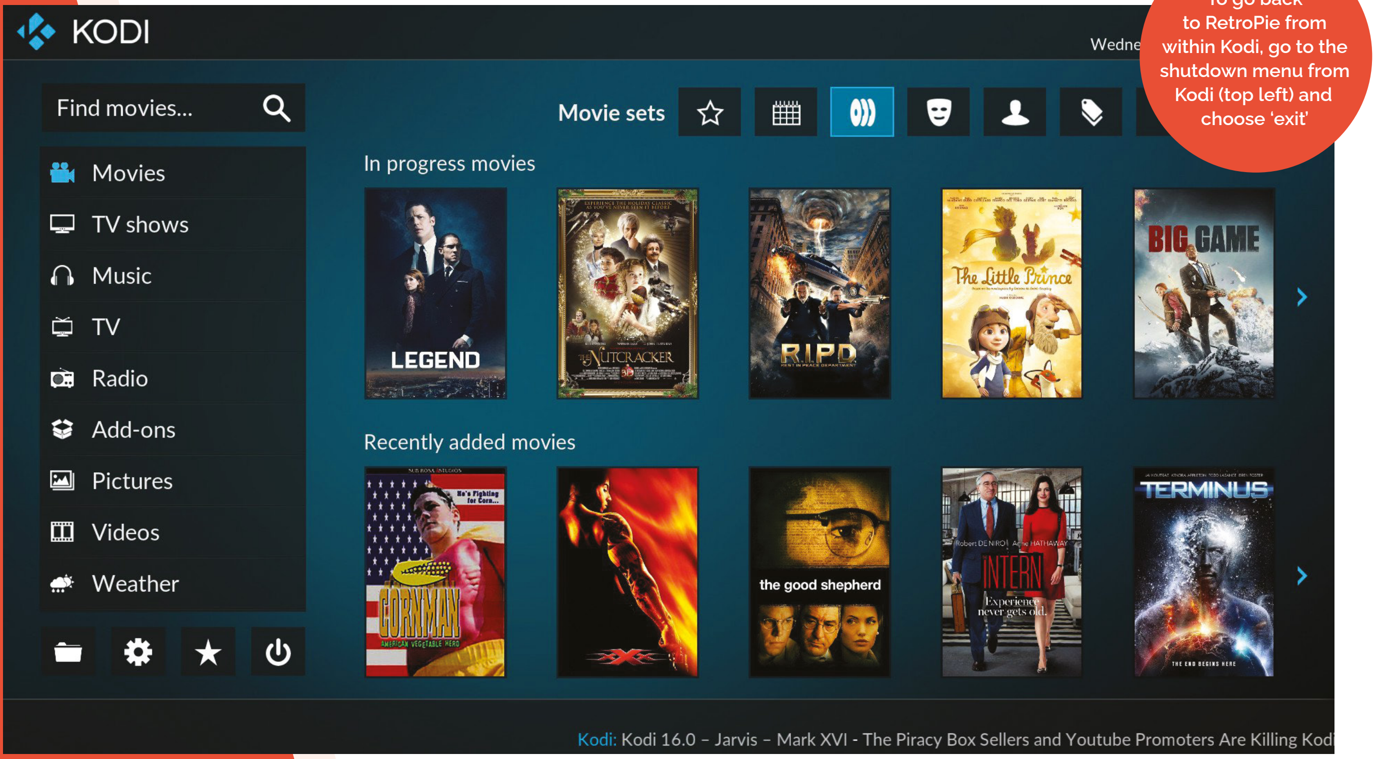
02 Setup Kodi

Kodi will be automatically set up after installation, however you may want to change some settings, especially if you're experiencing overscan issues. Find them in the settings menu, and be sure to give the name of the system an update if you want to as well so you can identify it on your network.

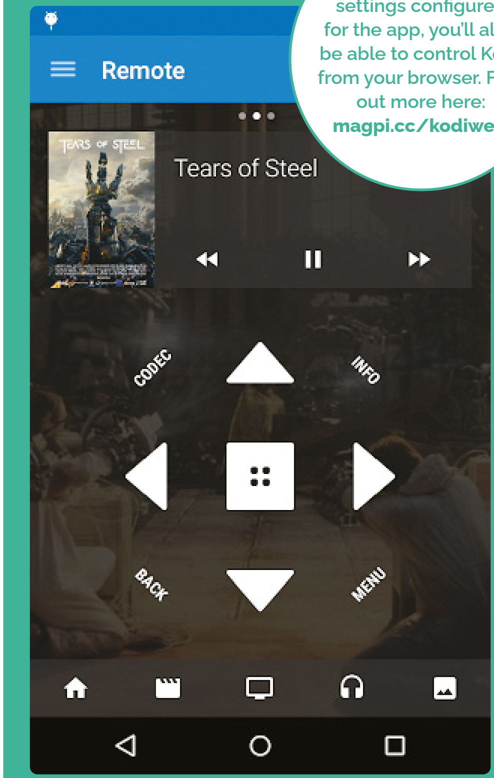
03 Add networked sources

The easiest way to access media (videos, music, or pictures) over your local network is to add them as media sources. Head back to the settings meny from the top of the interface. From there, go to File Manager and select Add Source. You'll be able to browse your available networked machines, or alternatively write in the SMB route if you know it.

To go back to RetroPie from within Kodi, go to the shutdown menu from Kodi (top left) and choose 'exit'



Kodi: Kodi 16.0 – Jarvis – Mark XVI - The Piracy Box Sellers and Youtube Promoters Are Killing Kodi



With the settings configured for the app, you'll also be able to control Kodi from your browser. Find out more here: **magpi.cc/kodiweb**

Smart device remotes

You can also control Kodi using Android and iOS devices, and there's an official app for it as well, which you can find here: **magpi.cc/kodiremote**.

While it's downloading to your desired device, you'll need to tell Kodi to allow you to use it. From the Settings menu, go to Services and find the Control option. From this list, you want to turn the following options to ON: 'Allow programs on other systems to control Kodi', 'Allow control of Kodi via HTTP', and 'Announce these services to other systems'.

Open the app and you should be able to select the name you gave your Kodi setup from the list and immediately start controlling it.

Using IR remotes

The Flirc USB dongle allows you to program specific media commands into itself from another computer before plugging it into a media centre. You may need a USB extender so it's visible, though!

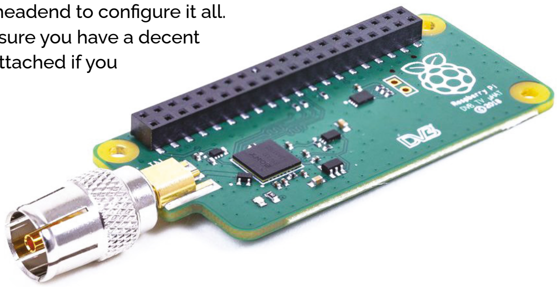
The Argon ONE case does have an IR receiver, but it seems to have a couple issues with working at the moment. You can always try installing the IR software with `argonone-ir` from the command line, as there will likely be a future update for it.

> "The first time you start Kodi, you'll have to do a basic setup"

Watching TV

With a Raspberry Pi TV HAT, you can watch and record TV using Kodi. You can have it streaming TV via a different Raspberry Pi if you don't have an aerial socket near your home theatre, or have it as a server and viewer from the same device. We have a full guide you can find here: **magpi.cc/tvhatguide.**

The system uses Tvheadend to configure it all. You'll need to make sure you have a decent amount of storage attached if you plan to record TV.



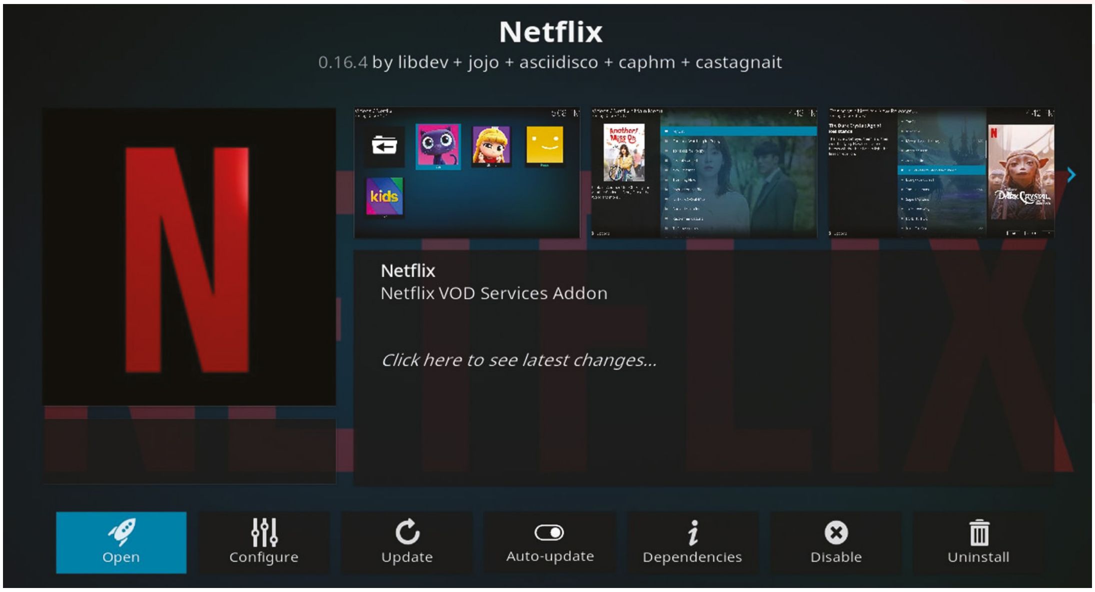
A Raspberry Pi TV HAT enables you to watch and record digital terrestrial TV

Essential add-ons

Get streaming services and TV running on Kodi for the full smart TV experience

Warning!
Unofficial Add-on

Check that Netflix add-ons like this are legal in your region.



Netflix
0.16.4 by libdev + jojo + asciidisco + caphm + castagnait

Netflix
Netflix VOD Services Addon

Click here to see latest changes...

Catch your favourite Netflix shows and movies on Kodi using a third-party source

INSTALL NETFLIX

01 Get the source

While Kodi doesn't have an official Netflix add-on, you can easily add a third-party one that allows easy access to the streaming service. First of all, head to Settings and go to System. Scroll down to Add-ons and turn 'Unknown sources' to on. Go back and head to the File Manager – go to Add Source and enter the following address:

https://castagnait.github.io/repository.castagnait/

Call it 'netflix' to make things easier.

02 Install the add-on

From the Settings menu, go to Add-ons and scroll down to 'Install from zip file'. As 'netflix' is now a source, you can select that to find a file called **repository.castagnait-x.x.x.zip** to download. Go back to Add-ons and go to 'Install from repository', where you'll find an option for CastagnaIT Repository for Kodi. Select it, then Video Add-ons, and you'll see the Netflix add-on. Go to install it, choosing the latest version, and say OK to extra stuff that will be installed.

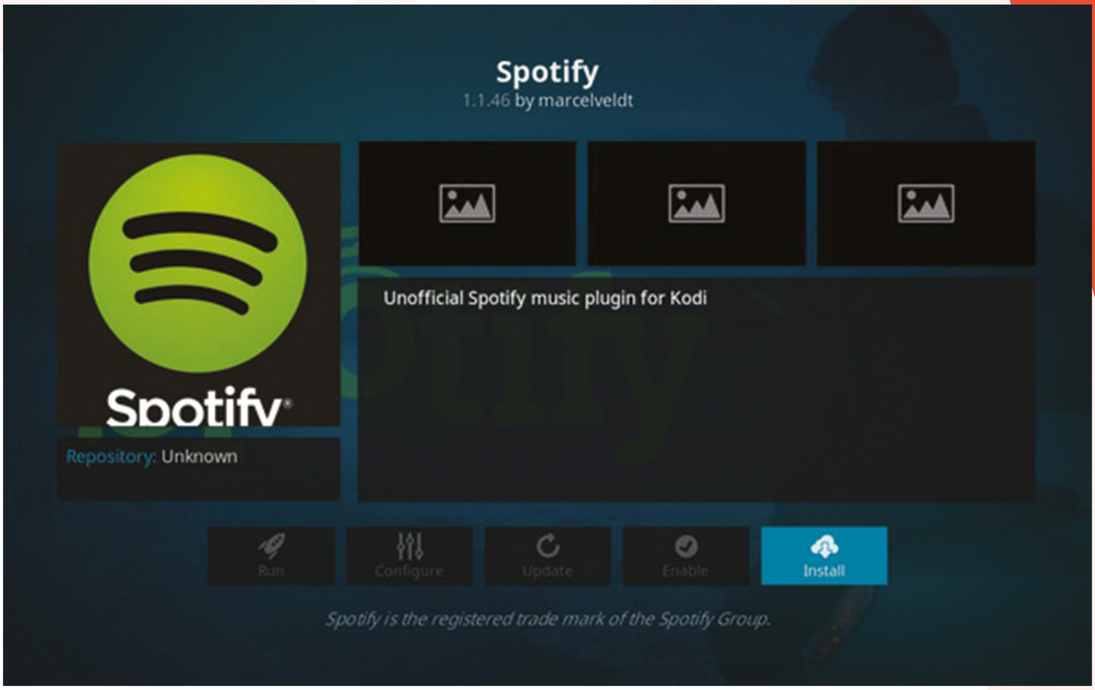
> "Go to install it, choosing the latest version, and say OK to extra stuff"

03 Using Netflix

You may need to add extra software for Netflix if you're having issues. Exit Kodi, and press **F4** in RetroPie to access the command line. Enter the following:

```
sudo apt install python3-pip
pip3 install cryptodome
sudo apt install libnss3
```

Head back to Kodi and Netflix should launch.



Spotify

1.1.46 by marcelveldt

Unofficial Spotify music plugin for Kodi

Repository: Unknown

Spotify is the registered trade mark of the Spotify Group.

Listen to your favourite tunes with an unofficial Spotify add-on for Kodi

INSTALL SPOTIFY

01 Spotify source

There isn't an official Spotify app on Kodi, so you'll need to install from a third-party source like with the Netflix add-on, although this one you'll need to transfer from another computer. Head to **magpi.cc/spotifykodi** on a computer and download the zip file from the link under 'Install with repository'. If you're able to transfer it to your media PC over the internet, do so, or load it to a USB stick and plug it into the media box we've been building.

02 Install source and add-on

Like with Netflix, head to Settings, Add-ons, and 'Install from zip file'. Head to the folder you downloaded it to, or go through the USB storage you're transferring it with to find the zip file. Head to 'Install from repository' and go to Marcelveldt's BETA repository. Go to Music Add-ons and then install the Spotify app from there. If you're having trouble installing, you may need to disable the Netflix add-on and restart.

Install YouTube

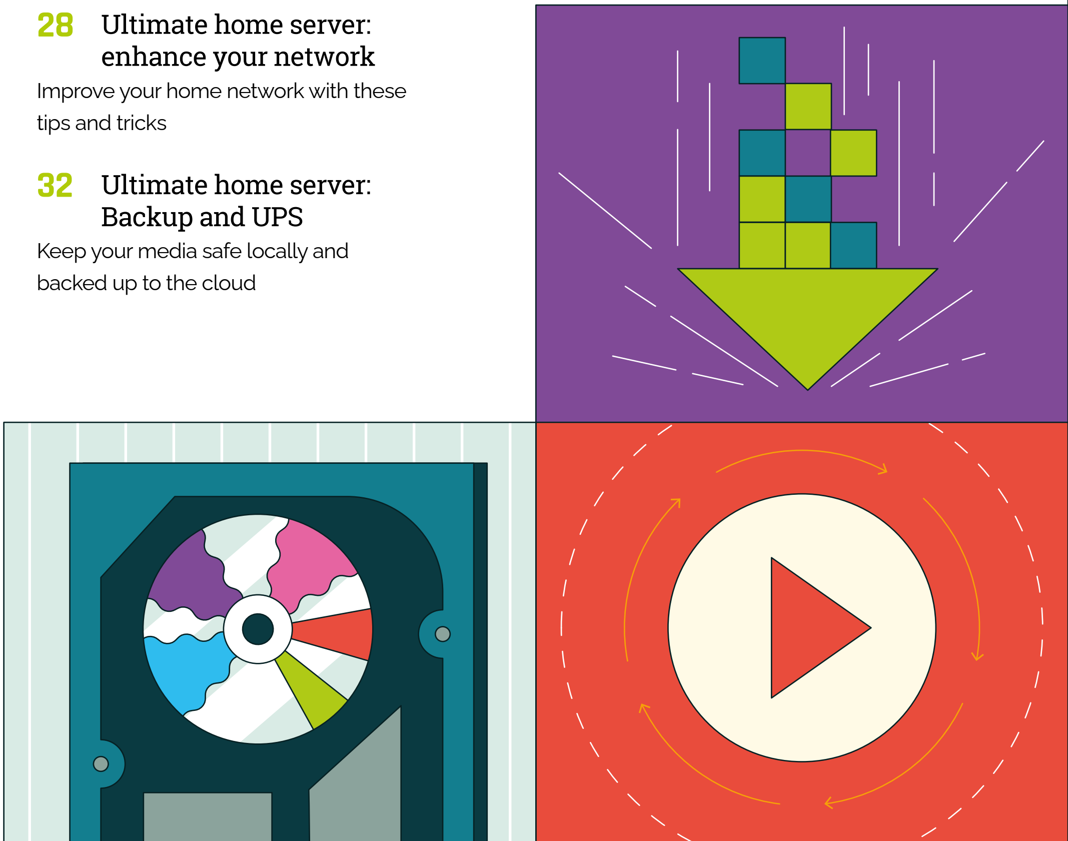
Unlike Spotify and Netflix, this one is a bit easier – a YouTube app is available in the standard Add-ons list for Kodi! From the Add-ons menu, go to Download, Video Add-ons, and scroll down to YouTube. Install it from there and it will live in the Add-ons menu from now on. Like the other apps, you'll have to use the standard Kodi interface to navigate it.

03 Configure and use Spotify

Use the Configure option from the screen where you installed the Spotify app and enter your Spotify login details. Once that's done, head back to the home screen and you'll find Spotify under the Add-ons menu. Open it to start browsing and listening – once again, it will use the standard Kodi interface instead of the usual Spotify one.

Build the ultimate home server with Raspberry Pi

Creating your own media server, media centre, HTPC, or whatever you want to call the box under your TV has never been easier. We have a comprehensive guide on how to create and optimise your perfect project.



Part 01

Ultimate home server: assemble your equipment

Revolutionise your home network with shared storage, media serving, network management, home automation and more. All in one tiny box!

MAKER

PJ Evans

PJ is a writer, software engineer and tinkerer. His server has just told him it's time for another coffee.

twitter.com/mrpjevans

The rear of the case organises all the ports neatly and the bridge connects the M.2 drive to your Raspberry Pi 4

You'll Need

- Raspberry Pi 4 **magpi.cc/raspberrypi4**
- Argon One M.2 case **magpi.cc/argononem2**
- 1TB M.2 SATA SSD drive **magpi.cc/wdbluem2**

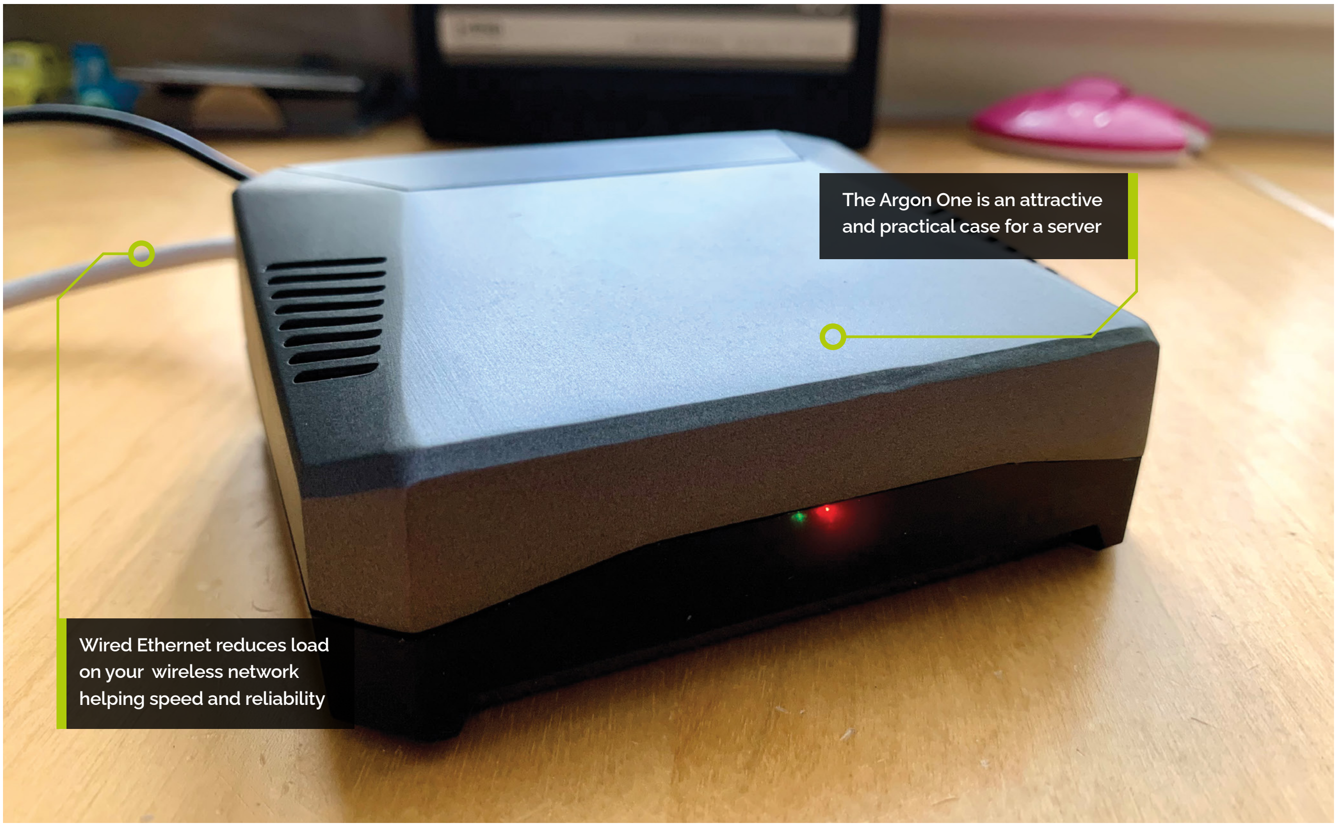
Raspberry Pi also makes a great home server. Why do you need a home server? Well, there's all kinds of uses: file sharing, protecting your network from dodgy advertisers, controlling your smart devices; the list goes on. In this new series, we'll build a home server with all the bells and whistles to inspire you to create your own ultimate build, with plenty of ideas and tips along the way.

01 Choose the right model

It's probably no surprise that we're using a Raspberry Pi 4 with 8GB RAM model for our ultimate server. It's the most powerful Raspberry Pi. However, you can go for lower memory sizes and get similar performance. If you're interested in using advanced file systems like ZFS for protecting your data, you'll need at least 4GB for it to work effectively. Then consider what the server will be doing. Serving files doesn't require much effort and is well within the capabilities of Raspberry Pi Zero, and we strongly recommend a wired network connection.

02 Choose a case

Our Raspberry Pi 4 needs a good home if it's going to be a server. There's no shortage of cases available and if you're lucky enough to have a 3D printer, many more designs are available for download. Have a think about storage requirements such as additional disks or access to USB or GPIO connectors. We've chosen the Argon One case. Not only is it beautiful, it also features


The Argon One is an attractive and practical case for a server

Wired Ethernet reduces load on your wireless network helping speed and reliability

passive cooling (as well as a fan) and support for M.2 storage devices. As a server is intended to be running 24/7, make sure your choice of case is well ventilated and will be placed where it can 'breathe'.

03 Choose storage

The standard application for a server is file sharing: the ability for anyone in your household to access a library of files. If you're thinking of a media server, consider carefully what type of storage to use and how much you need. Movies, especially in high-definition formats, eat up a lot of space but don't require fast storage to play back reliably. We're making use of the Argon One's M.2 interface to add 1TB of fast SSD-based storage, but you may prefer one or more USB SSD drives. Using a microSD card is still an option, but slower and less reliable.

04 Update your bootloader

By default, a Raspberry Pi 4 and Raspberry Pi 400 will try to boot from the microSD card and then from a USB attached drive. So you will not need to do any additional tweaks to get the M.2 drive working. USB boot is also available on most other Raspberry Pi models, including Raspberry Pi 2, 3A, 3B, and Compute Modules 3+ and 4. However, you will need to update the bootloader on these devices. See **magpi.cc/bootloader** for info and instructions.

05 Argon assemble!

Now we have the computer, case and storage, it's time to put them together. The Argon One comes with a comprehensive installation guide. Add the expansion board to bring all the connections to the rear of the case, then mount in the case as instructed, using the supplied thermal pads to get a good connection between the board and the heatsink. Install your M.2 SSD drive in the lower part of the case, then screw the two halves together. The M.2 SSD device is connected to Raspberry Pi 4 using an external USB-to-USB bridge.

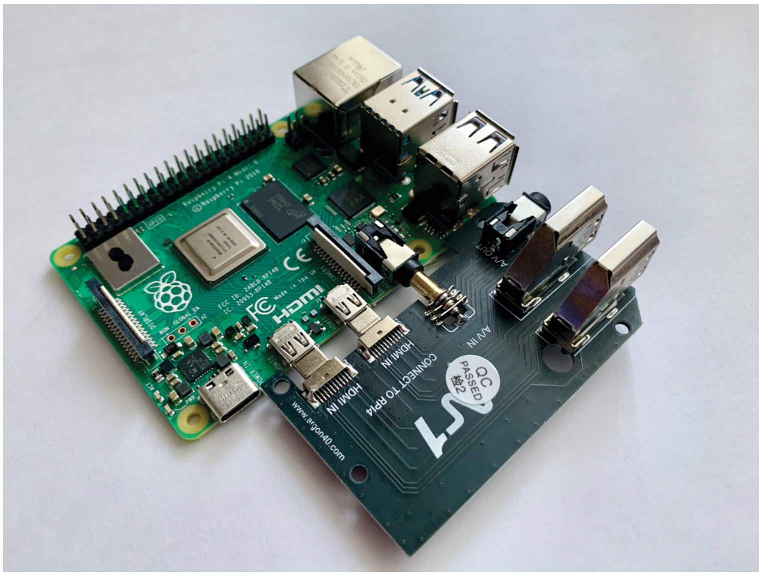
06 Prepare the operating systems

Preparing an M.2 SSD device is identical to preparing an SD card, but connecting the M.2 device may be tricky. Either purchase an inexpensive M.2 to USB device or start with an existing Raspberry Pi build and use Argon One M.2's USB bridge to connect it. Then use Raspberry Pi Imager to 'burn' a Raspberry Pi OS disk image onto the SSD device so

Top Tip

USB drives

If you're going for mass storage and fancy adding an array of drives, don't forget a powered USB hub to avoid undervolting – this can damage your data and hardware.



The Argon One's clever expansion board brings all the connectors to the back and converts the HDMI ports to full-size

The aluminium case connects to the CPU and memory using thermal paste to act as a giant heatsink

it's ready to boot. As this is a server, we don't need a graphical user interface, so we can keep things lightweight. Using Raspberry Pi Imager, select Raspberry Pi OS Lite and check out the advanced menu (**CTRL/CMD+SHIFT+X**) to set things like hostname and networking before burning the image to the card. We are naming our hostname 'ultimate' so it is easy to find on the network.

Top Tip

Which M.2?

Be careful when selecting your M.2 SSD. The Argon One supports SATA devices, not newer NVMe types.

07 Networking choices

Full-time servers are not normally connected to your network over wireless LAN. Whilst wireless is a great technology for end-users, servers crave reliability and the best way of delivering that is a wired Ethernet connection. We recommend placing your server close to your home broadband router (or whichever device handles your network traffic). That way, the network is not trying to handle two wireless conversations: one between you and the router and another between the router and the server. Wired will be much faster and reliable. If this isn't an option, don't worry: it'll still work, but larger files may be slow to transfer.

08 Boot time

Let your server boot for the first time, making sure no microSD card is inserted. Even if you configured it as headless, it's worth plugging a monitor in so you can check everything is OK. It will quickly reboot to resize the file system and then start up. We used the Advanced menu in Raspberry Pi Imager to set the hostname to 'ultimate', so with a wired connection we can ping 'ultimate.local' from another computer to confirm it's on the network. If you didn't do this, set up your hostname by running `sudo raspi-config` from a Terminal. You can also set up networking here if you need to. Update everything with:

```
sudo apt update && sudo apt -y full-upgrade
```

09 Boot from microSD

There is an argument to be made for still using a microSD card as the primary boot device. This dedicates your M.2 SSD to data only. In the event of an operating system failure, your data can still be accessed by removing the SSD drive. Likewise, if you need to increase your storage capability, you avoid having to rebuild your operating system from scratch. If you want to take this approach, make sure your bootloader prioritises the SD card (you can check in raspi-config > Advanced Options > Bootloader Version). You will you'll need to partition and format the second USB drive with an ext4 file system.

10 Fan and power button support

The Argon One case incorporates not only passive cooling but also has a fan for keeping our Raspberry Pi 4B nice and cool. It's not the quietest but does support active control. To install this, enter the following command in a Terminal:

```
curl https://download.argon40.com/argon1.
sh | bash
```

This will download the drivers and install them automatically. The fan will now adjust its speed

> We recommend placing your server close to your home broadband router

based on temperature. It will also allow the power button to cleanly shut down after being pressed for three seconds. To change these settings, run `argonone-config` from the command line.

11 Access from other devices

Let's go back to networking. Our server's hostname is 'ultimate', so using the ZeroConf DNS system (aka mDNS), it should be on the network as 'ultimate.local'. To test this from another machine, run `ping ultimate.local` from the command line and see if you get a response from the server. If you get no response, check your network settings on Raspberry Pi using `sudo raspi-config`. If using wireless LAN, try a wired connection instead. You can also use utilities like Fing (**fing.com**) to see if your server has acquired an IP address. (Note that versions of Windows prior to 10 do not natively support .local addresses.)

12 Create a user

All Raspberry Pi operating systems start with a pre-installed user called 'pi'. So we can share files and manage permissions, it makes sense to create our own user accounts. To do this, open a Terminal as the 'pi' user and run this command:

```
sudo adduser <name>
```

...where <name> is the username you would like (avoid spaces and special characters). It will ask you a few optional questions about your full name and location. Finally, you can set a password (be sensible!).

To give the user 'super user' or 'sudo' rights:

```
sudo adduser <name> sudo
```

To give the new user all the same permissions as the 'pi' user:

```
sudo usermod -a -G
```

The lower part of the case accepts an M.2 SSD for large storage needs

```
sudo usermod -a -G adm, dialout, cdrom,
sudo, audio, video, plugdev, games, users,
input, netdev, gpio, i2c, spi <name>
```

You should now be able to log in as your new user. You can create as many of these as you need.

13 Create and exchange SSH keys

A fast and secure way to log into your new server is SSH key exchange. If you anticipate lots of logging into your new server with SSH, this is an essential step. If you're using Windows, start by installing the free OpenSSH client from Microsoft's instructions (**magpi.cc/msopenssh**).

From a command line, first generate your keypair (if you don't already have one):

```
ssh-keygen
```

This stores a public and private key in your home directory under **~/.ssh**.

Copy your public key over to your server:

```
ssh-copy-id <name>@ultimate.local
```

...where <name> is the username you created previously. Now you can log into your server without a password using SSH:

```
ssh <name>@ultimate.local
```

If your local username matches the server's, you don't even need to include it or the '@'.

You've now got the basis for building a powerful home server. In the following tutorial, we'll move on to the most useful aspect of any home server: file sharing.

Part 02

Ultimate home server: sharing and storing files

Sharing and storing large files on your network equals lots of fun and productivity without filling up your computer

MAKER

PJ Evans

PJ is a writer, software engineer and tinkerer. His server needs to go on a diet.

twitter.com/mrpjevans

"Have you got that file? Can you email it to me?" "But it's 10GB! It's too big!" We've all been there. If you're in a house full of data-hungry family members, or working in a small business where a single repository of information would be enormously helpful, then a file-sharing device (sometimes called 'network-attached storage' or NAS) is what you need. We're going to take the Argon M.2 Raspberry Pi server we have built and add file sharing and increase disk capacity so there's plenty of room for all your stuff.

01 Choose the right model

To create our ultimate file sharing device, we need to install some file-sharing software. There are many different protocols and options available to us. It makes sense to choose a protocol that is supported by as many different platforms as possible, such as Linux, Windows, and macOS. The de facto standard is SMB (Server Message Block) and the software that implements SMB on Raspberry Pi OS is called Samba. Installation is simple:

```
sudo apt -y update && sudo apt -y upgrade
sudo apt install samba
```

If you get asked a question about WINS, just say no. After installation completes, you have a running file server, but with nothing configured.

02 Prepare your sharing location

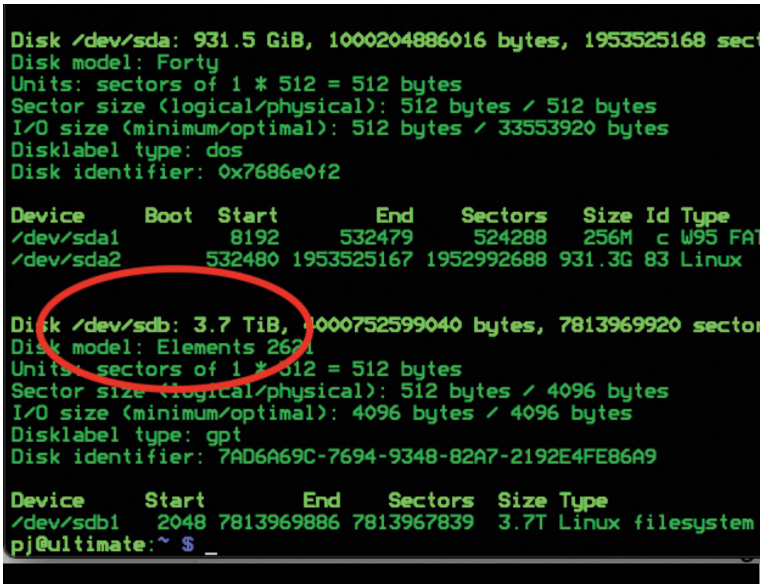
The next choice to make is where to store the files that we wish to share on the network. There's no hard and fast rule on this. You can use your user's home directory or create a dedicated area. We're going to create a new directory under **/var**, which is an area in the Linux file system designated for files that will 'vary'.

```
sudo mkdir /var/shared
```

We're now going to allow any user to read and write files in that area:

```
sudo chmod -R 777 /var/shared
```

Putting the 'welcome mat' down like this is not considered best practice, but it serves our purposes. For critical implementations, investigate more granular read/write settings.

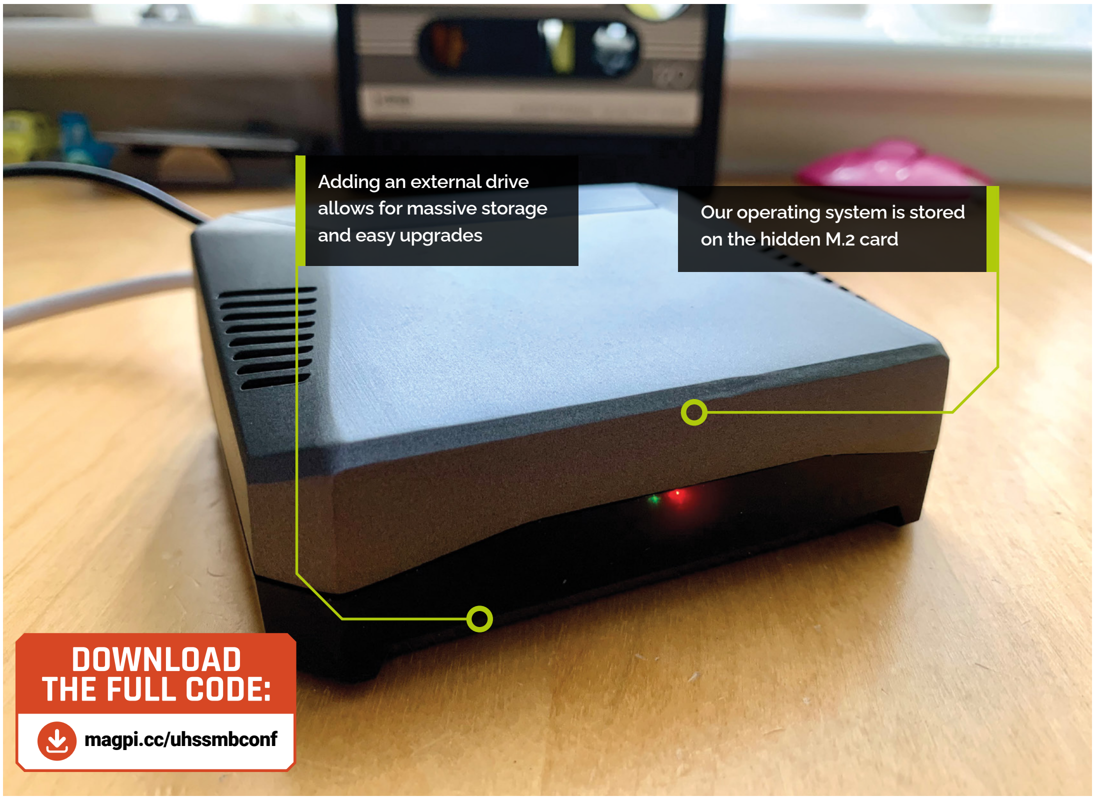


```
Disk /dev/sda: 931.5 GiB, 1000204886016 bytes, 1953525168 sect
Disk model: Forty
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 33553920 bytes
Disklabel type: dos
Disk identifier: 0x7686e0f2

Device     Boot  Start        End    Sectors   Size Id Type
/dev/sda1         8192     532479     524288   256M  c W95 FAT
/dev/sda2       532480 1953525167 1952992688 931.3G 83 Linux

Disk /dev/sdb: 3.7 TiB, 4000752599040 bytes, 7813969920 sector
Disk model: Elements 2621
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 4096 bytes
I/O size (minimum/optimal): 4096 bytes / 4096 bytes
Disklabel type: gpt
Disk identifier: 7AD6A69C-7694-9348-82A7-2192E4FE86A9

Device     Start        End    Sectors Size Type
/dev/sdb1   2048 7813969886 7813967839 3.7T Linux filesystem
pj@ultimate:~ $ _
```

Identify your drive carefully before removing and creating partitions

You'll Need

- Raspberry Pi ultimate server **magpi.cc/108**
- 4TB Western Digital USB 3.0 external drive (or similar) **magpi.cc/wdelements**
- Powered USB hub **magpi.cc/usbhub**



Adding an external drive allows for massive storage and easy upgrades

Our operating system is stored on the hidden M.2 card

DOWNLOAD THE FULL CODE:

magpi.cc/uhssmbconf

03 Configure Samba

We need to tell Samba to share our new directory. All the sharing settings and configuration options are stored in the file **/etc/samba/smb.conf**, so open that up in an editor:

```
sudo nano /etc/samba/smb.conf
```

This is a large file, but don't be put off: we're just going to add to the end of it. Add the text shown in the code listing for **smb.conf**.

Getting connected

How you connect to remote servers varies by platform, but we'll cover some common ones here.

- **Raspberry Pi Desktop:** Open the File Manager, click 'Go' in the menu bar, click 'Network', select 'ULTIMATE' (or whatever you named your server), click the 'Shared' directory, and enter your credentials (leave the domain as WORKGROUP).
- **macOS:** Open Finder, click on 'Network' on the sidebar, select 'ULTIMATE' (or whatever you named your server), click 'Connect As…', then enter your username and Samba password. Click 'Shared'.
- **Windows:** Open File Explorer and right-click on 'This PC', then select 'Add a network location'. You'll go through a short wizard; when asked which server to connect to, enter **\\ULTIMATE\Shared** and enter your username and Samba password.

Take care to indent the code correctly. These lines set up the most basic kind of sharing. Save the file and exit the editor (**CTRL+X**), then restart Samba so the new configuration is picked up.

```
sudo systemctl restart smbd
```

smb.conf

Language: **Config / Bash**

```
[Shared]
    comment = Shared
    path = /var/shared
    read only = no
    browsable = yes
```

04 Configure your users

Security constraints mean Samba cannot check your password when you connect. To mitigate this, you can set an additional password for Samba access. This can be the same as your regular password for something different. To add your user account to Samba and set your access password, just run the following command:

```
sudo smbpasswd -a <user>
```

…Where `<user>` is your current username or any other username that needs Samba permissions. Enter your new password twice and you're ready to go.

05 Get yourself connected

Samba is now ready for use. You can also mount shared drives from the command line in Raspberry Pi OS using `smbmount`, or connect to your Raspberry Pi from another computer on your network – see 'Getting connected'.

06 Adding more storage

Now you have a shared drive up and running, you may want to load it up with videos, audio, and more. Soon you may need more space than our M.2 drive provides. You may also wish to consider separating data and the OS onto different disks so it's easier and safer to perform upgrades or expansions. Thankfully, the powerful USB 3.0 ports on Raspberry Pi 4 are perfect for adding inexpensive larger external drives, such as the whopping 4TB Western Digital drive we've chosen. If you want to go for multiple drives, we recommend a powered USB hub to make sure there are no 'brown-outs' that could cause lost data.

07 Identify the new drive

When you plug in a USB 3.0 external drive, Raspberry Pi OS will recognise it and add it as a new device in the **/dev** directory. We need to find the assigned name of the device. Typically it will be **/dev/sdb**, but not always. We can check with this command:

```
sudo fdisk -l
```

The output will show two physical devices, labelled as 'Disks': your M.2 drive and the new external drive. It should be easy to tell them apart by the size and manufacturer name. Look for the 'logical name' of the external drive and make a note of it.

08 Partition the disk

To create a suitable file system, we first need to partition the new disk. Run the following command (assuming your external drive's logical name is **/dev/sdb**):

```
sudo fdisk /dev/sdb
```

FDISK is a utility for managing your disk's partitions – the areas where data can be stored.

Now complete the following steps (note that this will erase any data on the disk). Repeatedly enter 'd' followed by **RETURN** to delete any current partitions until you get a 'no partition' error. Enter 'n' followed by **RETURN** to create the new partition, accepting all the defaults. Enter 'w' to write the changes to the disk and exit

09 Format the disk

Once a disk has been partitioned, it needs a file system. There are many to choose from, but a popular choice in the Linux world (and therefore Raspberry Pi OS) is ext4. To start, we need the partition name. If you have a look in the device directory (**ls /dev**), you will see that sdb (our drive) has been joined by sdb1 (our partition). To format the partition:

```
sudo mkfs.ext4 /dev/sdb1
```

This will probably take a few minutes to run based on the size of the disk.

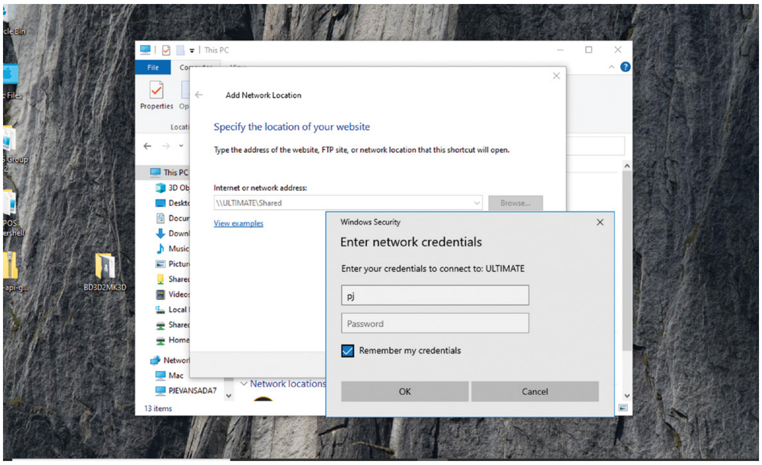
10 Mount the disk

To access your fancy new drive, you need to 'mount' it. This is Linux-speak for making the formatted partition available as a directory on the

Top Tip

RAID

Did you know you can use multiple disks to provide protection against drive failure? RAID is a system for keeping two or more drives in constant sync, so if one fails, the other has your data!



Add Network Location

Specify the location of your website

Type the address of the website, FTP site, or network location that this shortcut will open.

Internet or network address: \\ULTIMATE\Shared

View examples

Windows Security

Enter network credentials

Enter your credentials to connect to: ULTIMATE

pj

Password

☑ Remember my credentials

Windows supports connections too, but it's a little tricker

DOWNLOAD THE FULL CODE:

magpi.cc/uhssmbconf2

Our server connected to an external drive

Top Tip

Data danger!

Using FDISK and MKFS can cause data loss. Check all your device logical names are correct and that there's no data on the disk that you need to keep.

system. The first step is to create an 'endpoint': a folder that will act as the gateway to the new disk. The normal directory to use for this is 'mnt'.

```
sudo mkdir /mnt/huge
```

You don't have to call it 'huge' – use any name you like. You only have to do this once; the folder will persist like any other.

To mount the drive:

```
sudo mount -t ext4 /dev/sdb1 /mnt/huge
```

That's it. Any files written into that folder will be placed on our new disk. To unmount:

```
sudo umount /mnt/huge
```

11 Automount on boot

The problem with the mount command is that it will not persist after a reboot. You can add a mount command to something like **rc.local**, but the best way to deal with this is to add the drive to the file system table (fstab). This is a dangerous step as a misconfigured fstab can make your Raspberry Pi unbootable. First, get the UUID of your new drive. Run this command:

```
sudo blkid
```

Find the line containing our new drive (**/dev/sdb**) and copy the list of numbers and letters quoted in UUID="". Now edit the table:

smb.conf (part 2)

Language: **Config / Bash**

```
[Huge]
    comment = Huge
    path = /mnt/huge
    read only = no
    browsable = yes
```

```
sudo nano /etc/fstab
```

Carefully append the following line:

```
UUID=c13507ef-190d-400d-9fc0-96f7cedea521
/mnt/huge ext4 rw,user,auto 0 0
```

This is an example UUID: Make sure everything after UUID= is the UUID of your new drive. Save and reboot. Your drive should be automatically mounted.

> To access your fancy new drive, you need to 'mount' it

12 Share the disk

The final step is to let Samba know about our new disk. Set the permissions and edit the Samba config file as we did earlier:

```
sudo chmod -R 777 /mnt/huge
sudo nano /etc/samba/smb.conf
```

At the bottom of the file, add a new line and add the contents of the **smb.conf (part 2)** listing. Save the file and exit (**CTRL+X**). You can now restart Samba to start the share without rebooting:

```
sudo systemctl restart smbd
```

Now try mapping from a remote device. All being well, you now have a huge amount of storage at your disposal.

We've now got our server sharing files across your network. So, what else can you do with an always-on device? In the next tutorial we'll look at one of the most popular applications: media streaming.

Part 03

Ultimate home server: add sound and vision

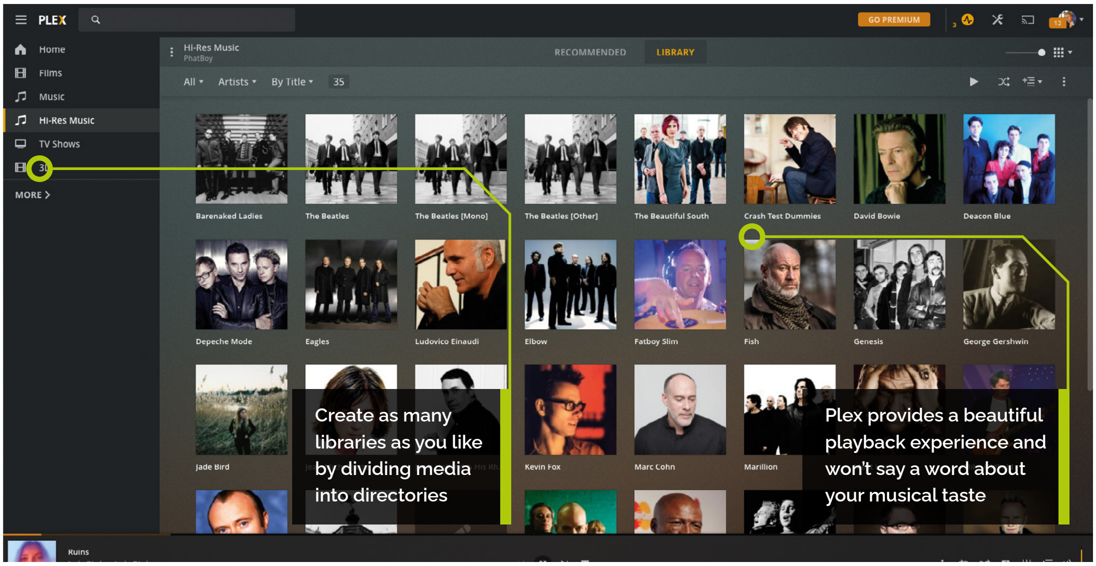
Turn your server into a media powerhouse with Plex and Mopidy

MAKER

PJ Evans

PJ is a writer, software engineer, and tinkerer. His ambient music collection has been described as "inspiring" (PJ Evans) and "just a load of random noise" (Mrs Evans, music teacher).

twitter.com/mrpjevans



Create as many libraries as you like by dividing media into directories

Plex provides a beautiful playback experience and won't say a word about your musical taste

Sound and vision is where we're at for our ultimate server project in this tutorial. Sure, serving files is all very well (see the previous tutorial) but what about a decent media interface? Luckily we're spoilt for choice. We're going to add two media serving solutions, Plex and Mopidy, to our server. Plex is a great choice for playing back your media anywhere in the world and Mopidy can turn your server into a jukebox as well as a multi-room audio streaming system. We'll also look at how to create your file structure and how to add content from different sources. Let's hit the play button.

01 Check, check, and check again

Before we plunge into the world of media streaming, let's make sure your ultimate server is correctly configured as per the previous tutorial. We want to be sure our external drive is mounting on startup. Reboot and then issue this one simple command in the terminal:

```
df -h
```

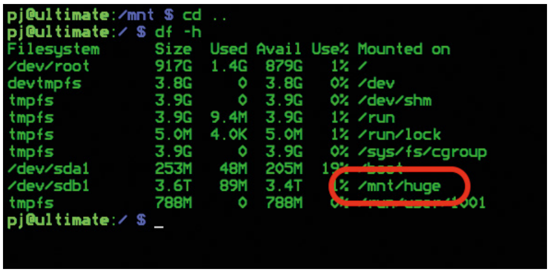
The 'df' command is short for 'disk free' and shows you a summary of disk usage. If all is well, you'll see the external drive (partition **sdb1**) mounted to **/mnt/huge** (or whatever you decided to call the mount point). If /mnt/huge does not appear in the list, refer back to the previous tutorial and make sure the entry in **/etc/fstab** is correct.

02 Set up your library

Media streaming tends to involve a lot of files. Before you find yourself in a mess, take some time to consider how you would like to structure all your media. How you do this is up to you, but

You'll Need

- Raspberry Pi ultimate server **magpi.cc/108**
- Plex **magpi.cc/plex**
- Mopidy **magpi.cc/mopidy**



```
pj@ultimate:/mnt $ cd ..
pj@ultimate:/ $ df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/root       917G  1.4G  879G   1% /
devtmpfs        3.8G     0  3.8G   0% /dev
tmpfs           3.9G     0  3.9G   0% /dev/shm
tmpfs           3.9G  9.4M  3.9G   1% /run
tmpfs           5.0M  4.0K  5.0M   1% /run/lock
tmpfs           3.9G     0  3.9G   0% /sys/fs/cgroup
/dev/sda1       253M   48M  205M  19% /boot
/dev/sdb1       3.6T   89M  3.4T   1% /mnt/huge
tmpfs           788M     0  788M   0% /run/user/1001
pj@ultimate:/ $ _
```

The command `df -h` shows that the external drive has correctly mounted

when it comes to ripping CDs, many encoders such as abcde (A Better CD Encoder, **magpi.cc/abcde**) will automatically rip to a given structure, so it can be helpful to plan this out now. See **Figure 1** for an example of a well-laid-out file structure. Don't feel that you have to follow this – if you would prefer more complex (or simple!) divisions, that's completely up to you.

Top Tip

Quickly copy your media

Moving a lot of media files over the network can be very time-consuming. Luckily, our USB drive can be mounted onto the source machine for super-fast copying. Remember, the host operating system will need a driver that can understand ext4 partitions.

03 Add some sounds

Now we have lots of fast USB 3.0 space to play with, we can add some media that we'd like to stream. This is a massive topic in itself, so we'll just give you some pointers here. You can now legally buy MP3s from many sources such as Amazon and Apple. These are free from DRM (Digital Rights Management), meaning you can play them back on your Raspberry Pi media system without issues. If you want to get the best quality possible, consider buying in FLAC format (no lossy compression) from services such as HDTracks (**hdtracks.com**) or Qobuz (**qobuz.com**).

Figure 1

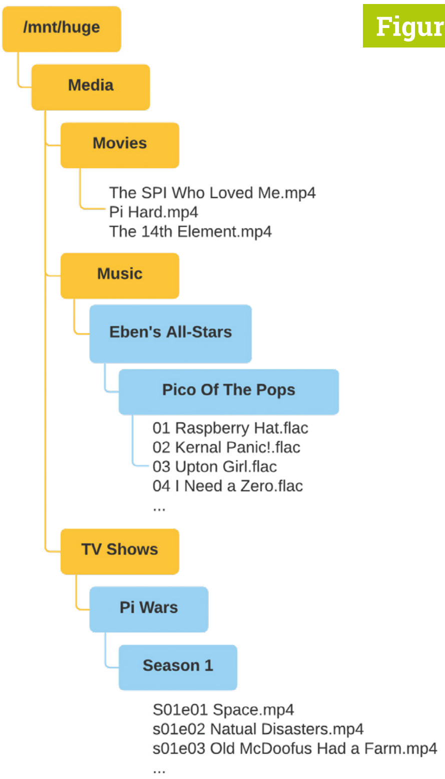


Figure 1 Here's an example of a sensible directory structure for your media files. Remember to create it on your external mounted drive

04 Going to the movies

Adding mainstream movies can be a bit challenging as there are no legal vendors of DRM-free movies. However, you can use Plex's own free streaming service to access content with a wide range of (sometimes cheesy) movies to watch, as well as video podcasts. You can also use plug-ins to access all of YouTube. Of course, if you have a collection of home movies, it's a great way to get them up on the big screen.

05 Let's Plex!

We're going to install Plex, one of the most popular media streaming services available. It supports a dizzying array of file formats for both audio and video. Plus, clever built-in transcoding systems mean it can compress on-the-fly to ensure the smoothest of playbacks. All with a very friendly, good-looking user interface. Installation takes a few steps, so follow the commands in the **install_plex.sh** listing to add Plex's repository, update, and then install and configure Plex. Once complete, we can check Plex is running using this command:

```
sudo systemctl status plexmediaserver
```

You should see that Plex is 'loaded' and 'active (running)'. Press **Q** to exit and you're ready to configure!

06 First-time Plexing

When we installed Plex, it started a web server on port 32400. So all you need to do is open a web browser on another machine (or the server itself if you've installed a desktop) and go to:

```
http://ultimate.local:32400/web
```

(Replace the server name if you didn't use 'ultimate' when setting it up in part 01 of this series.)

What you should see now is the Plex welcome screen. Be patient: it can take a few seconds on the first run. To get the best out of Plex, create a free account as prompted using an authentication provider such as Google or just your own email address.

07 Initial setup

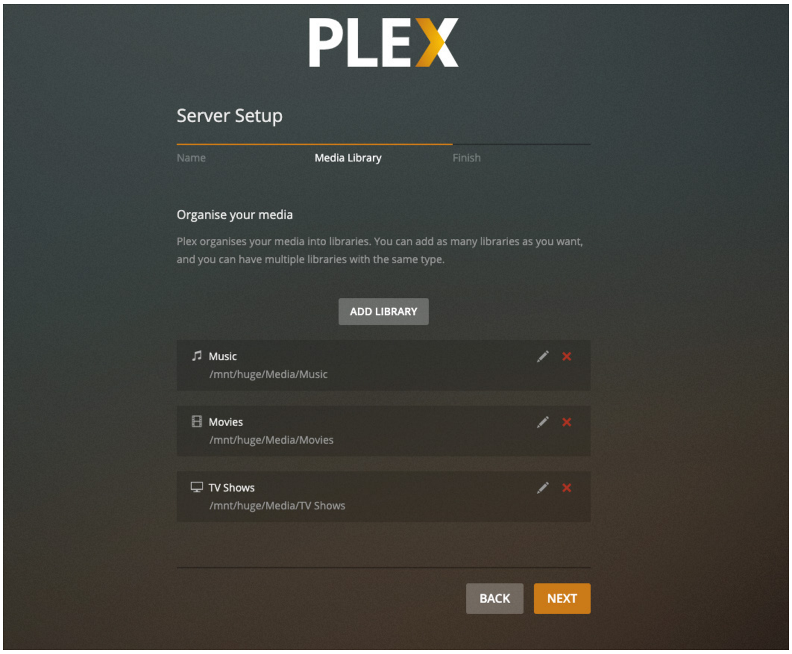
Plex will take you through a short wizard to get things set up. Start by giving your Plex server a name; we've chosen 'ultimate' to be consistent. You can also opt for access to your Plex library from anywhere on the internet. Plex will attempt to open a port on our home router so you can access your media from the **plex.tv** website. If you don't want to do this, ensure the checkbox is clear. Click 'Next' to start adding your libraries. One-by-one, select a library 'type' (music, movies, etc.) when navigating to the corresponding folder on your external drive; e.g. for 'Music', use **/mnt/huge/Media/Music**.

08 Library management

Plex will now start scanning your folders for content. It will also access online databases to add metadata to its library to make your collection as complete as possible. Depending on the size of your collection, this process will take anywhere between seconds and hours. It's a background process though, so you can start using Plex right away as your media appears. You'll also notice that Plex will also offer you online content – you can disable this if you wish or sign up for a 'Plex Pass' for a lot more features. If you don't like what Plex has 'found' for your media types, you can always click the pen icon and edit the metadata; Plex doesn't always get it right.

09 Access it anywhere

One of the great things about Plex is the availability of its client applications. Not only is there a rock-solid web-based interface (which works great on Raspberry Pi OS's Chromium) but support for a wide range of devices including PlayStation, Xbox, Apple TV, Amazon Fire devices – and for mobile, iOS and Android. If you created an account earlier, all you have to do is install the app and log in. Your ultimate server will be detected and added. Now you can access and stream your media collection from any location on the internet.



PLEX

Server Setup

Name · Media Library · Finish

Organise your media

Plex organises your media into libraries. You can add as many libraries as you want, and you can have multiple libraries with the same type.

ADD LIBRARY

Music
/mnt/huge/Media/Music

Movies
/mnt/huge/Media/Movies

TV Shows
/mnt/huge/Media/TV Shows

BACK · NEXT

Once Plex is installed, you need to add each directory and specify what type of media it contains

10 Streaming to headless devices

One restriction of Plex is that it requires a device with some kind of interface to work and plays audio to that interface. If we wanted to play audio on the ultimate server itself, or stream to another Raspberry Pi computer that is connected to speakers, we need to use Mopidy, a purpose-built audio streaming solution. Luckily, Plex and Mopidy can live side-by-side and even share libraries. To install Mopidy and its user interface, Iris, follow the commands in the **install_mopidy.sh** listing. Once complete, you can access the user interface here:

```
http://ultimate.local:6680/iris/
```

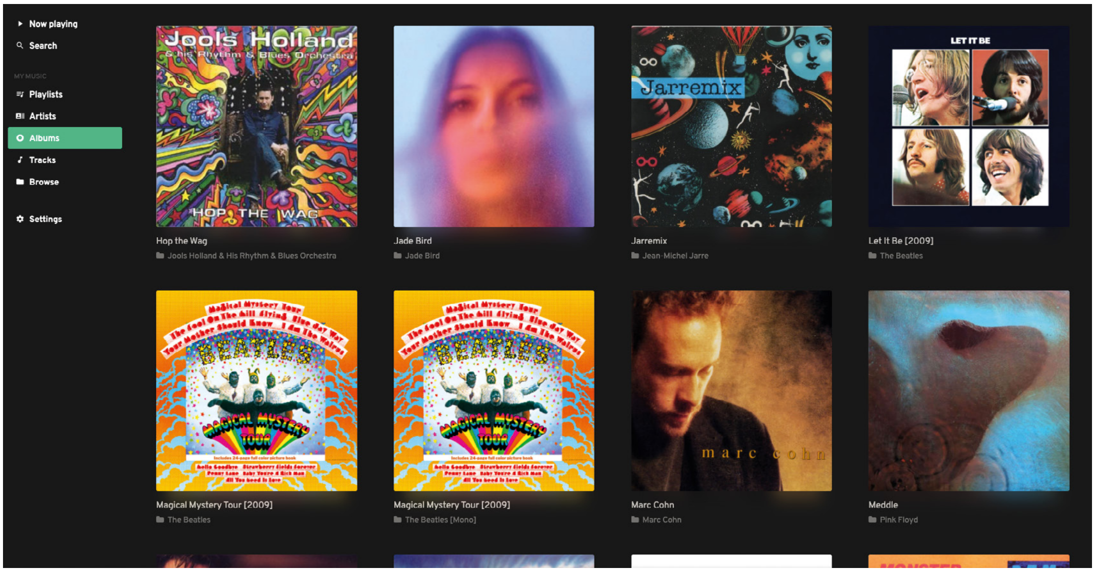
install_plex.sh

Language: **Bash**

DOWNLOAD THE FULL CODE:

magpi.cc/installplexsh

```bash
#!/bin/sh

# Plex's repository uses HTTPS as the transport, so we need to
install support
sudo apt update
sudo apt install apt-transport-https ca-certificates curl

# Get Plex's public key and add the Plex repo
curl https://downloads.plex.tv/plex-keys/PlexSign.key | sudo apt
key add -
echo deb https://downloads.plex.tv/repo/deb public main | sudo
tee /etc/apt/sources.list.d/plexmediaserver.list

# Install the server
sudo apt update
sudo apt install plexmediaserver
```



Mopidy is an alternative to Plex, designed for local playback and streaming

Warning!

The laws covering the ripping of CDs, DVDs and Blu-rays vary from country to country. The private copying of copyrighted material is illegal in the United Kingdom. Check what you want to do is legal before proceeding!

magpi.cc/ripping

11 Adding media to Mopidy

In the previous step we installed the 'local' plugin, which allows Mopidy to access local file systems for media. One restriction is that it can only support a single directory. Start by editing the config file:

```
sudo nano /etc/mopidy/mopidy.conf
```

Add the following at the end of the file:

```
[local]
media_dir = /mnt/huge/Media/Music
```

Remember to change this path if you're using a different naming system or structure. Save the file (**CTRL+X**), and instruct Mopidy to scan the contents, then restart:

```
sudo mopidyctl local scan
sudo systemctl restart mopidy
```

Reload Mopidy in your web browser and the media should be available. Now you can play music using your server as the audio source should you wish.

12 Advanced Mopidy

Now you have Mopidy installed, you can upgrade to a few special features. Mopidy supports a wealth of plug-ins and one of the most popular is mopidy-spotify, which, you guessed it, turns Mopidy into a Spotify client (although you need a premium account for this). Also, you can install Snapcast to create a synced multi-room audio system for a fraction of the price of solutions such as Sonos. For more detail and tutorials on all of this, check out *The MagPi* issues #96, #97, and #98.

In the fourth part of this tutorial series, we're going to look at how Raspberry Pi OS can be put to work to make your network a better place. Access your files securely from anywhere in the world, protect yourself from tracking ads, and more.

install_mopidy.sh

Language: **Bash**

DOWNLOAD THE FULL CODE:

magpi.cc/installmopidysh

```
#!/bin/sh

# Add Mopidy repo and install
wget -q -O - https://apt.mopidy.com/mopidy.gpg | sudo apt-key add -
sudo wget -q -O /etc/apt/sources.list.d/mopidy.list https://apt.mopidy.com/buster.list
sudo apt update
sudo apt install mopidy
sudo adduser mopidy video

# Configure (be careful with this line!)
printf "[http]\nhostname = 0.0.0.0" | sudo tee -a  /etc/mopidy/mopidy.conf

# Install user interface
sudo apt install python3-pip
sudo pip3 install Mopidy-Iris
sudo pip3 install Mopidy-Local
sudo sh -c 'echo "mopidy ALL=NOPASSWD: /usr/local/lib/python3.7/dist-packages/mopidy_iris/system.sh" >> /etc/sudoers'

# Enable and start service
sudo systemctl enable mopidy
sudo systemctl start mopidy
```

Part 04

Ultimate home server: enhance your network

Make your ultimate server a network genie with these apps

MAKER

PJ Evans

PJ is a writer, software developer and tinkerer. His server has just told him it's time for lunch. This might be getting out of hand.

twitter.com/mrpjevans

In the previous parts of this tutorial, we've taken a Raspberry Pi 4 and turned it into a file sharing, media streaming powerhouse. Next we're taking on your network. There are so many ways to enhance your home (or office) network that we can't hope to fit them into a few pages. However, these steps will point you in the right direction and hopefully inspire you to make your home network smarter, faster, and more convenient to use. Follow along and we'll automate your home, allow you to access your data securely from anywhere in the world, and automatically protect you from nefarious websites.

01 Settle on an address

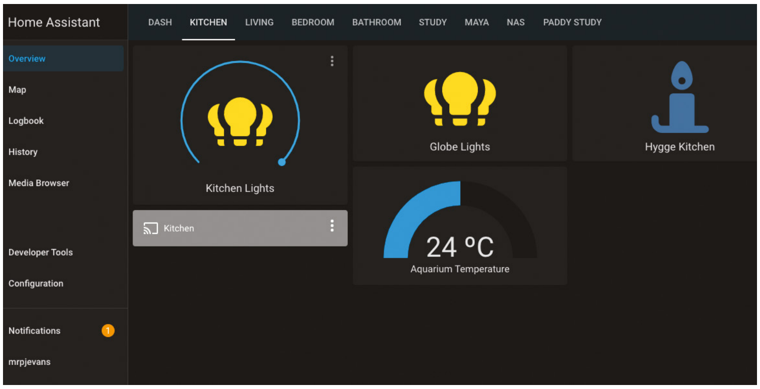
A key aspect for any server is that you should know where it is. It seems obvious, but a typical home network will assign IP addresses dynamically (a system known as DHCP) and there's a fair chance they will change from time to time. We normally get around this by using zero-config domain names like '.local', but we're going to need a set IP address for our server going forward. The easiest way to do this is to ask your router to reserve an IP address for your ultimate server. We recommend you look at doing this before proceeding. Sorry for the vagueness, but every router does this differently.

02 Home control the easy way

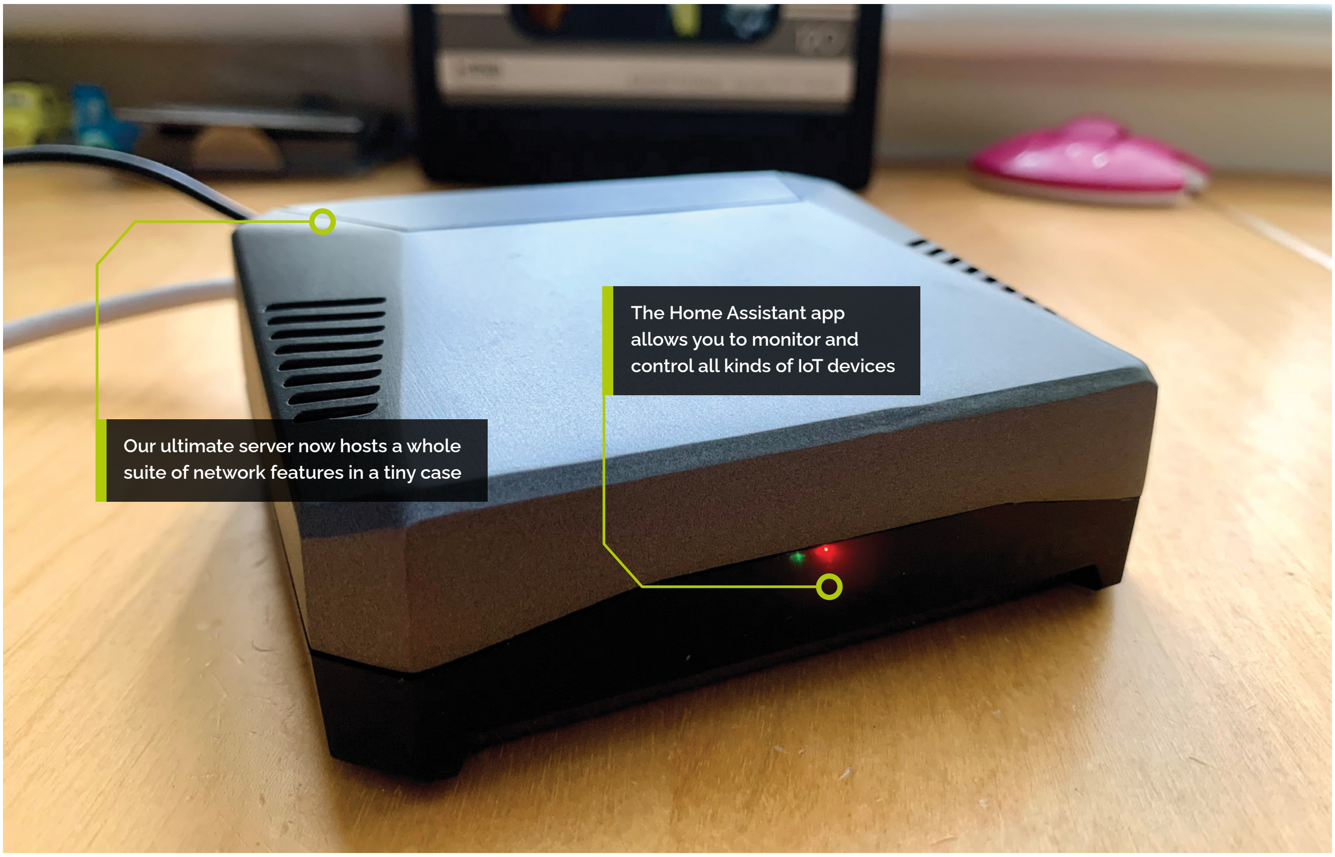
Want to automate your home? Fancy being able to control lights or check a home webcam from anywhere in the world? How about if that software cost you absolutely nothing? Home Assistant is an incredible package that integrates a wide range of inexpensive smart devices such as Philips Hue, Ikea Trädfri, and many more. Best of all, it's open enough that you can create your own activities and integrate your own projects relatively easily. Installation on an existing server requires a few steps, but a great guide has been written by the HA team – **magpi.cc/hacore** – that we have summarised here: **magpi.cc/hasummary**.

03 Getting started with Home Assistant

The first time you run Home Assistant, it'll take a few minutes or so to get its act together as it installs everything it needs. HA will also scan your network for any existing devices you have. Once running, you can access the dashboard at **http://ultimate.local:8123** (change the name if you didn't call your server 'ultimate'). You may be surprised to see what is already detected and available. If not, there are hundreds of plug-ins to extend HA's capabilities. Home Assistant is a massive topic, so you may want to have a read of our tutorial in *The MagPi* #99 (**magpi.cc/99**).



Home Assistant can control hundreds of IoT devices with customisable dashboards like this



Our ultimate server now hosts a whole suite of network features in a tiny case

The Home Assistant app allows you to monitor and control all kinds of IoT devices

04 Add an MQTT broker

We will absolutely forgive you if this title sounds like gibberish. MQTT is a very simple protocol for sending messages from device to device. It is loved by the home automation community, so it's no surprise it's well supported by Home Assistant. The principle is simple: any device can publish its data (e.g. a temperature reading) to an MQTT 'topic' (a label to describe the data stream). Then, any other device can 'subscribe' to the MQTT topic and receive updates in real-time. It makes brewing your own sensors or switches so much easier as you don't have to know about the target.

> “ A key aspect for any server is that you should know where it is ”

05 A nice kind of mosquito

One of the most popular MQTT brokers (a server that receives and broadcasts MQTT events) is Mosquitto. Installation is very straightforward:

```
  sudo apt install mosquitto mosquitto-clients
```

This will install Mosquitto and make sure it's running as a service at all times. To test it's working, open two Terminal windows and run these commands, one in each:

```
  mosquitto_sub -t "test"
  mosquitto_pub -m "message from mosquitto_pub
client" -t "test"
```

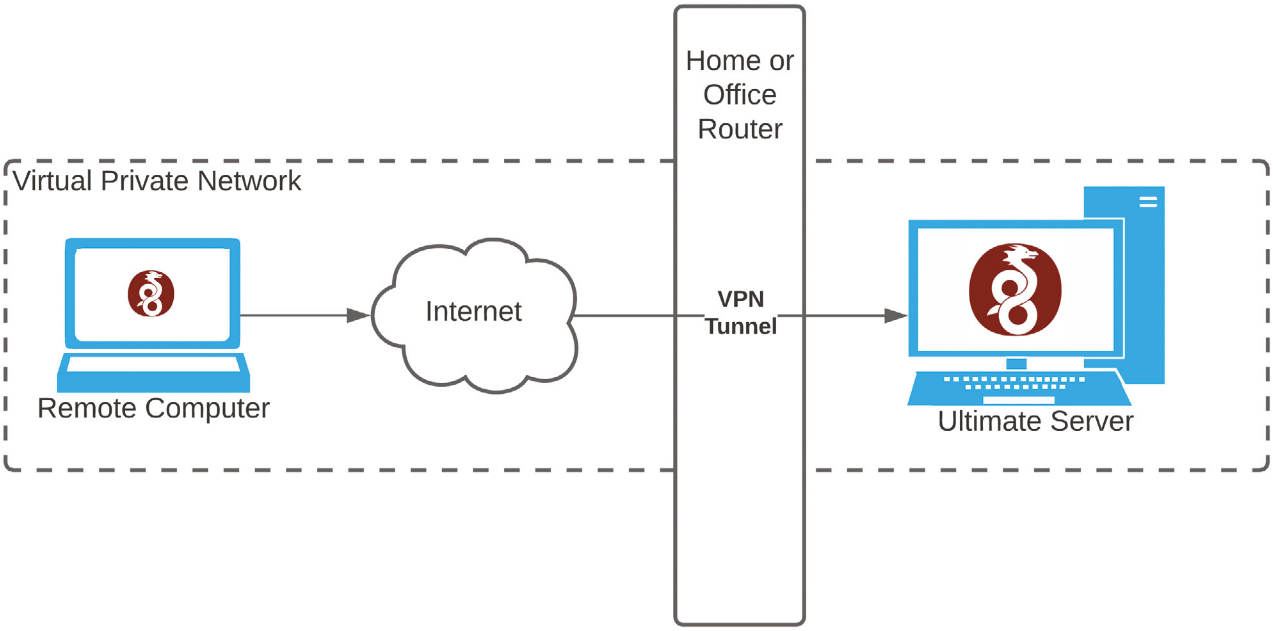
If all is working, you will see a window 'talking' to the other. Now anything that generates MQTT messages can connect with you and your Home Assistant server. Just add the MQTT integration in HA and you can create graphs, alerts, and more.

06 Adding remote access

Previously, we've configured our server to stream audio and video and now we can switch lights, check the living room, feed the cat, and more. So, if you're not home, how do you do this? Remote access to your home network is not something to take lightly, as if you can get access, what stops someone else? The answer is PiVPN, a script that installs a VPN (virtual private

You'll Need

- Raspberry Pi Ultimate Server as built in the previous tutorials



▲ A virtual private network such as Wireguard creates a secure encrypted link to your home devices from anywhere

network) that you can connect to securely from anywhere in the world. Previously PiVPN only supported OpenVPN, which can be difficult to work with, but now the much simpler Wireguard protocol is supported.

07 Installing Wireguard

For lots of information on PiVPN and Wireguard, head over to The PiVPN Project at **pivpn.io**. To get started with installation, run the following command from a Terminal:

```
curl -L https://install.pivpn.io | bash
```

The installer will take you through a series of questions. An important point is to ensure you have a static IP address as your router will need to know where to find your Wireguard server. When asked what type of server to install, answer 'Wireguard'. Continue through the installer, accepting the default answers. If you don't have a static IP address for your router, you may need to consider a dynamic DNS service such as DynDNS (**account.dyn.com**) so you can easily reach your Wireguard service.

08 Using Wireguard

Before you can use Wireguard to get secure remote access to your network, you need to allow the internet to 'see' it. This is done by creating a port mapping. Your router needs to forward all traffic to port 51820 (default) over UDP to your server. How to do this varies from router to router, so search for your particular router's instructions.

To create users that can access the network, run this command:

```
pivpn add
```

Give a username when prompted and a config file will be created. Copy that config file to the device you want to use to access the network and import it using the free Wireguard client software. Now you can access your home network from a laptop or smartphone from anywhere in the world.

09 Add a web server

The fun thing about web servers is they are actually very simple things. You can place a text file in the home directory and the server will happily serve it anywhere in the world. If you would like to be able to access files through a web browser, you can combine Wireguard and a web server to make access both easy and secure. As an example, we're going to configure access to the media library we have previously created. First, install a web server:

```
sudo apt install apache2
```

After that single line, you would be able to access a simple welcome page at **http://ultimate.local**.

Top Tip

Building your Home Assistant setup

The Home Assistant community maintains a list of compatible smart devices. These either work 'out of the box' or with a free plug-in: **magpi.cc/haplugin**

10 Adding directories to your web server

What a web server can provide you with is only limited by imagination. Let's get simple access to our music and videos, which are stored in **/mnt/huge/Media**. Apache has a complex configuration system, but we're going to play a little UNIX trick to make things easier: a symbolic link. This allows us to map one part of the file system to another, creating a kind of short cut.

```
  sudo ln -s /mnt/huge/Media /var/www/html/
media
```

Now you should be able to go to **http://ultimate.local/media/** and browse your files. It isn't pretty, but it's a start! Now you can connect to your network securely and access files easily.

> “ Pi-hole can protect your entire network, even your smartphone ”

11 Avoid the dodgy advertising

One of the most popular Raspberry Pi applications is Pi-hole (**pi-hole.net**), a clever system designed to 'sink' internet advertisements, leaving your browser clear of noise. While we strongly recommend supporting your favourite sites by allowing adverts, many sites contain ads that are not only scams, but can lead to tricking you into downloading malware that can compromise your data. Pi-hole can protect your entire network, even your smartphone. To install it, simply run this command from a Terminal:

```
  curl -sSL https://install.pi-hole.net | bash
```

The installer will ask several questions, but is well-designed and you're probably going to be OK to answer every one with the default answer.

12 Starting off with Pi-hole

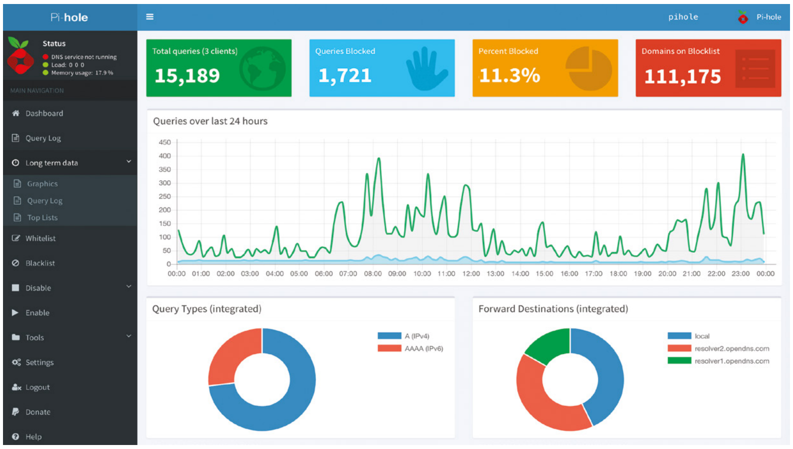
Pi-hole works by looking at DNS requests. These are requests to servers that map domain names such as **raspberrypi.com** to actual servers. It has a maintained blacklist of DNS servers that are for advertising and intercepts any such requests. To do this, Pi-hole must be your DNS server. Here things get complicated. For best results, your new Pi-hole server should also be your network DHCP server (the server that allocates IP addresses to your local devices). This requires some additional setup and we recommend you have a read of *The MagPi* #104 (**magpi.cc/104**).

13 Careful with those ports

A final warning. In this tutorial we've talked about opening ports to the internet using your router's port forwarding features. You may be tempted to just directly open ports to the various services we've installed so you can get access wherever you are. One tip: don't. The reason that services such as Wireguard exist is to ensure that bad actors can't find any easy way to access your stuff. Wireguard uses state-of-the-art encryption. So long as your config file is not compromised, you're more than likely completely safe. If you open your web server to the planet, we guarantee you will have bots sniffing around trying to find access within minutes. VPN isn't easy, but it's worth the time.

Well, we've now built a pretty impressive server. File sharing, media streaming, remote access, network protection, and home automation. In the next tutorial we will look at how to protect all this hard work, and your precious data, from unexpected loss. M

▼ Pi-hole is a network-wide ad blocker that's effective and easy to use



Status
- DNS service not running
- Load: 0 0 0
- Memory usage: 17.9 %

Total queries (3 clients) 15,189

Queries Blocked 1,721

Percent Blocked 11.3%

Domains on Blocklist 111,175

Part 05

Ultimate home server: Backup and UPS

MAKER

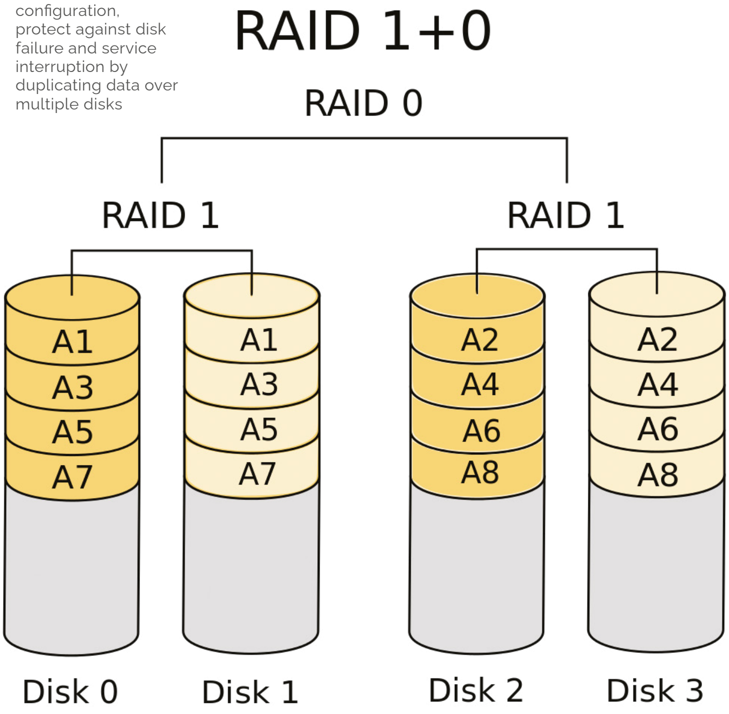
PJ Evans

PJ is a writer, software developer and tinkerer. He keeps several clones of himself in storage just in case.

twitter.com/mrpjevans

Now you have the ultimate server, let's keep your data safe with an uninterruptable power supply

If you've been following our home server tutorials over the past months, you should now have a pretty sweet Raspberry Pi-powered server up and running, with file sharing, media streaming, remote access, and more. In this final part of our Ultimate Server tutorial, we're going to look at protecting both your hard work and all that data you have entrusted to our favourite little computer.



RAID systems, like this RAID 10 configuration, protect against disk failure and service interruption by duplicating data over multiple disks

We're going to cover the three key pillars of server management: Backup, Reliability, and Monitoring. Luckily, these are not hard to achieve, even for a home setup, and you never know: there may come a day when you're really glad you took the time to protect your investment.

01 Don't be that person

When you've completed a project like this, it's tempting to stop now and enjoy the fruits of your labour. Sadly, the annals of computer history are filled with stories of those who did not heed the warnings and didn't back up. It's caused heartbreak when families have lost their photo libraries and bankruptcy when companies have lost their precious customer data. Never trust your data to one single source, no matter how much you've spent, or the reputation of the supplier. Think carefully about how to keep your data safe, not only from hardware or software failure, but also theft and disaster.

02 Three is the magic number

There's a long-held mantra amongst system administrators known as the 3-2-1 rule. You need three copies of your data, on at least two different media types with one off-site. Only then does your data safely 'exist'. This principle protects you against hardware failure and physical problems such as fire. Although this sounds initially complicated or at least a faff to set up, don't worry. Modern tools have made this easier than ever before. We can get your data to a healthy backed-up state without too

A simple way to keep data safe is to have a second copy ready to go

You can theoretically add up to 256 disk drives, but don't forget a powered USB hub even with just two, or you risk data loss

> You need three copies of your data, on at least two different media types

much work. What needs to be done is twofold: make sure the data is available elsewhere and that if you delete a file, you can get it back.

03 A local drive for local data

Let's start with a very simple way of backing up your data. We connected a 4TB drive to our ultimate server for all our lovely files. The easiest thing to do is to double it so we can mirror that data. Adding a second 4TB drive may sound expensive for little benefit, but when your original drive fails or you delete a critical file by accident, you'll be holding a parade through the streets when you realise you've got that data safe and sound on the second drive. All you need to do is set up a regular backup routine to copy the contents of your first drive to the second.

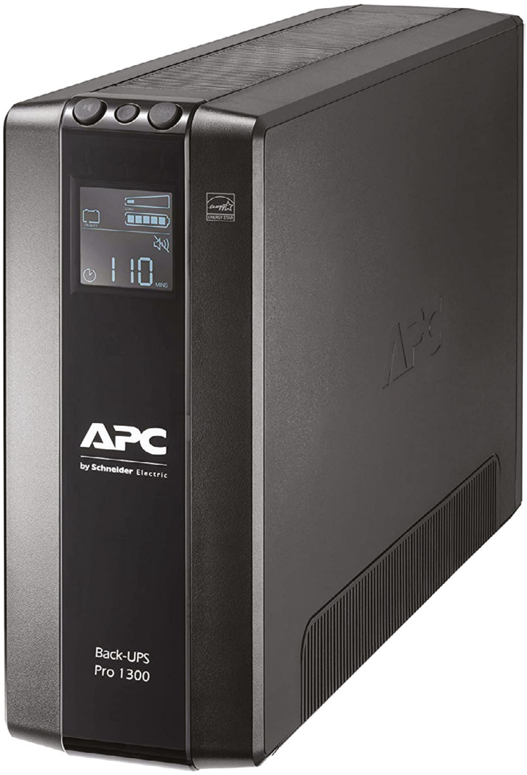
04 Quick and easy backup

Assuming you're adding an identical USB drive to your server, make sure you've partitioned and formatted the drive identically to the original. See tutorial part 1 for details on how to do this. Next, do an initial backup to ensure everything is OK. This may take some time, depending on how much data you have. If your first disk is mounted to **/mnt/huge** and your new disk is **/mnt/backup**, the command would be:

```
sudo rsync -av /mnt/huge /mnt/backup
```

You'll Need

- Raspberry Pi Ultimate Server (*The MagPi* issue #107 to #110) **magpi.cc/issues**
- Uninterruptible power supply (optional)
- Additional USB 3.0 external hard disks (optional)



There are many uninterruptible power supply (UPS) models available that provide a small footprint and adequate time for a clean shutdown, perfect for home use

Top Tip

Time Machine

Not only can you back up your ultimate server, but it can back up your Mac as well. Make your file shares Time Machine-compatible by following these instructions: **magpi.cc/timemachinegit**

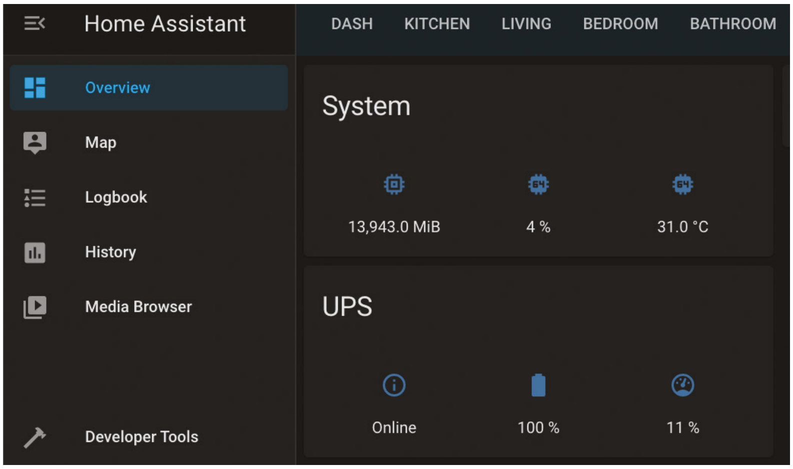
Rsync will only copy changes, so it's faster after the initial backup. To run this on a regular basis, add it to crontab (a schedule of regular jobs):

```
sudo nano /etc/crontab
```

Add this line at the end:

```
1 12 * * * root  rsync -av /mnt/huge /mnt/backup > /var/log/backup.txt 2>&1
```

This will run the backup every night at one minute past twelve. You can change it to any time or to be as (in)frequent as you like.



Home Assistant can monitor your server and your UPS, triggering alerts when necessary

05 Let's go offsite

A central principle of a good backup policy is that at least one copy is nowhere near you. This is to protect against all those things we really don't want to think about such as fire, flood, theft, or other types of damage. A wide range of services exist to securely store your data for you for a reasonable fee. If the worst happens, a copy of all your precious data will be available to retrieve. Some even offer a hard-drive-by-post service. There are too many services to cover here, but some tried and tested solutions for Raspberry Pi OS include iDrive, Amazon S3, and BackBlaze B2.

06 Introducing iDrive

If you want to keep things easy, we can recommend iDrive (**idrive.com**) as a contender. Not only do they offer very attractive first-year terms: they are one of the few off-site backup companies to offer full Linux (i.e. Raspberry Pi OS) support. Although you'll need to do some navigation of the command line, you can install the iDrive tools and create a destination and a backup-set in a surprisingly short amount of time. If you have a lot of data, the first backup may take some time, but iDrive can throttle the bandwidth used. Once uploaded, a regular incremental backup keeps you up to date. Best of all, deleted files and different versions of files can be archived.

07 The amazing rclone

There are many services available that offer some form of 'block' or 'bucket' storage. Most popular is Amazon S3, but there are a growing number of contenders such as BackBlaze B2, Box, or Digital Ocean Spaces. Working out how to communicate with all these can be a challenge, but luckily rClone (**rclone.org**) knows how to talk to over 40 different services. Sign up for the service you want, get your credentials, and configure rClone using its menus. Although trickier to configure than iDrive, this can be a very cheap way of backing up in the long term.

08 Let's get physical

In the modern age of the cloud, this may seem like an unnecessary step, but hear us out. How far you go with backup is dependent on how valuable you perceive your data to be. Typically, photographs fall into this category: memories that are irreplaceable. This is where the 'two types of media' rule can come into play. If you absolutely need that extra peace of mind, consider either burning the data to good-quality DVD or Blu-ray ROMs or, as many photographers do, never reusing an SD card and archiving them as you go. Many banks offer a cheap safety-box service where you can store physical media.

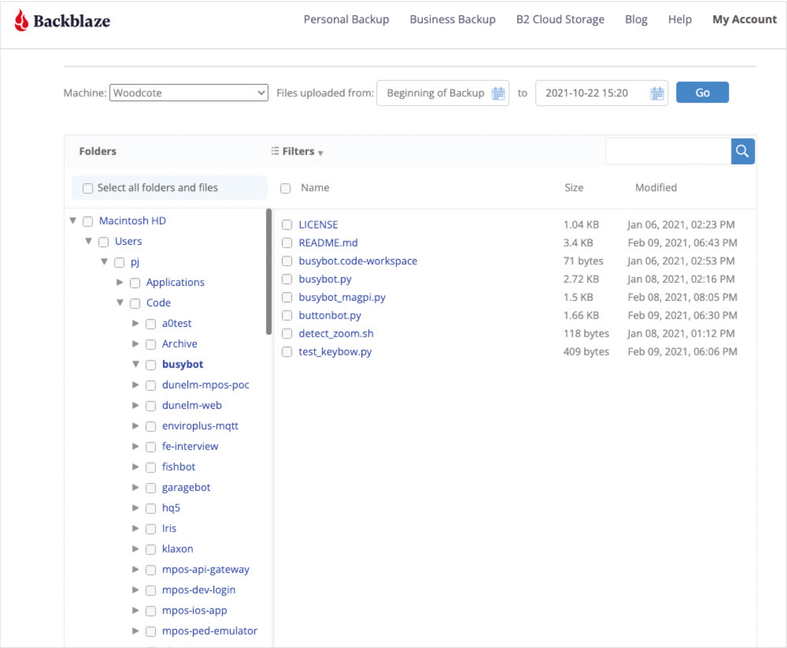
> " As for off-site data storage, there really are no practical limits "

09 Please don't interrupt

Now backups are sorted, let's look at making sure you don't need them. One of the classic causes of data loss is a sudden power cut. This is due to the operating system constantly writing and reading its storage and sometimes caching reads and writes in memory (known as 'journaling'). An inopportune power cut can use file system corruption and the loss of data. An uninterruptible power supply (UPS) is a battery that can keep your server going during a power cut. In the case of a Raspberry Pi computer, sometimes for hours. Best of all, they can communicate with your server over USB and invoke a safe shutdown when the battery gets low, protecting your data. Good UPSs for home use start at around £80.

10 Redundancy is a good thing

Another aspect of good server management is availability. If you have all your cool data on the server, you want to make sure it can handle failure and keep going. Although the two-drive system will prevent data loss, it's going to be a pain to reconfigure everything. For a truly robust system, you need a redundant array of inexpensive disks (RAID). This is a file system that allows a number of disks to operate as one and can tolerate at least one drive failure. You can then replace the drive and have the data protected again with no downtime at all. The ZFS file system is especially good at this.



Backblaze

Personal Backup | Business Backup | B2 Cloud Storage | Blog | Help | My Account

Machine: Woodcote | Files uploaded from: Beginning of Backup | to | 2021-10-22 15:20 | Go

Folders | Filters

□ Select all folders and files

- ▼ □ Macintosh HD
  - ▼ □ Users
    - ▼ □ pj
      - ▶ □ Applications
      - ▼ □ Code
        - ▶ □ a0test
        - ▶ □ Archive
        - ▼ □ **busybot**
        - ▶ □ dunelm-mpos-poc
        - ▶ □ dunelm-web
        - ▶ □ enviroplus-mqtt
        - ▶ □ fe-interview
        - ▶ □ fishbot
        - ▶ □ garagebot
        - ▶ □ hq5
        - ▶ □ Iris
        - ▶ □ klaxon
        - ▶ □ mpos-api-gateway
        - ▶ □ mpos-dev-login
        - ▶ □ mpos-ios-app
        - ▶ □ mpos-ped-emulator

□ Name	Size	Modified
□ LICENSE	1.04 KB	Jan 06, 2021, 02:23 PM
□ README.md	3.4 KB	Feb 09, 2021, 06:43 PM
□ busybot.code-workspace	71 bytes	Jan 06, 2021, 02:53 PM
□ busybot.py	2.72 KB	Jan 08, 2021, 02:16 PM
□ busybot_magpi.py	1.5 KB	Feb 08, 2021, 08:05 PM
□ buttonbot.py	1.66 KB	Feb 09, 2021, 06:30 PM
□ detect_zoom.sh	118 bytes	Jan 08, 2021, 01:12 PM
□ test_keybow.py	409 bytes	Feb 09, 2021, 06:06 PM

▲ Companies like BackBlaze and iDrive offer cheap off-site backup services so you know your data is safe

11 Two heads are better than one

Server reliability can often turn into a rabbit hole: when do you stop? The principles of high availability (HA) will keep a server running no matter what. If you really need to keep things ticking along, you need to eradicate every single point of failure, and that includes the server itself. In the last step, we removed a single hard drive as a point of failure. Using the principle of load balancing, we can remove the server as well. Software such as Varnish can split the load between two or more Raspberry Pi servers and if one fails, everything keeps on working. Magic!

12 Let it grow

Nothing we have discussed here prevents your server from growing in the future. Modern RAID systems such as Z2 allow you to increase disk capacity with no downtime. Using load-balancing software such as Varnish means you can add in more servers, or replace failed hardware, without anyone noticing. You're free to add more software and features whenever you want. As for off-site data storage, there really are no practical limits. Any time you need to boost your hardware, it's possible to do it with no interruption, again and again, truly making it the ultimate server.

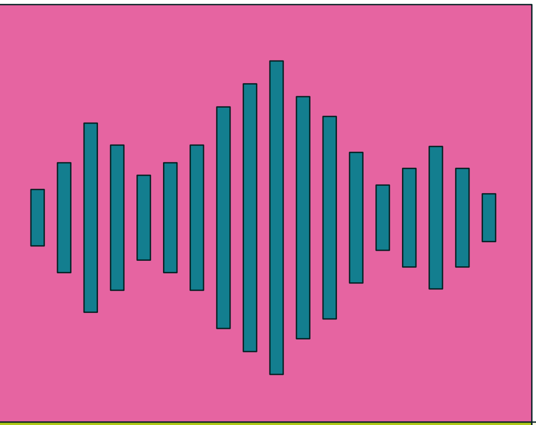
Top Tip

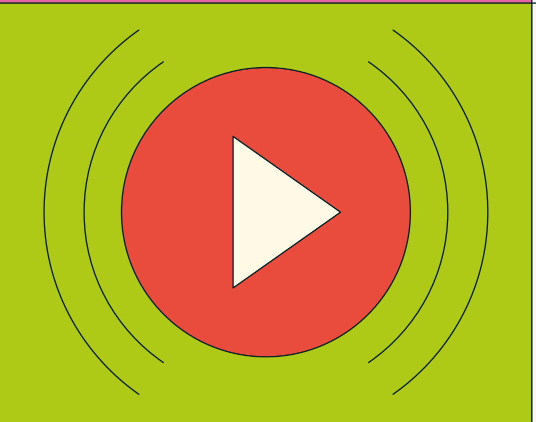
Remote Monitoring

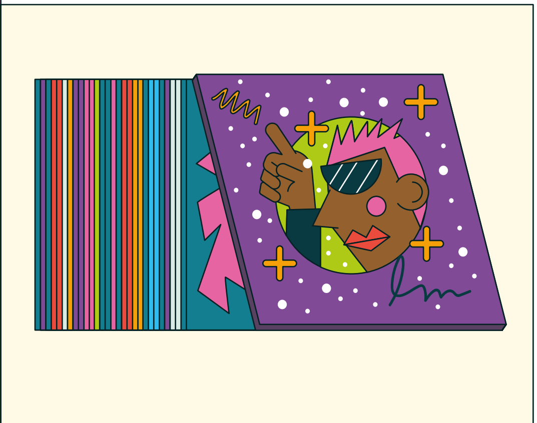
If you want to monitor your server from afar and have exposed a web server to the internet, services such as **uptimerobot.com** will alert you if they can no longer reach your site.

Build the ultimate Home Music System with Raspberry Pi

Add a digital-to-analogue (DAC) board to Raspberry Pi for the ultimate sound system.







Part 01

Ultimate home music: get started

Want your own whole-house audio without the price tag? In this three-part series, we create a digital jukebox using open-source software

MAKER

PJ Evans

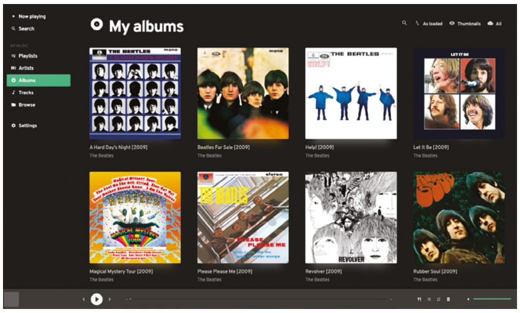
PJ is a writer, software engineer, and lover of open-source software. It's his turn to choose the next track, thank you very much.

@mrpjevans

The ability to play music throughout your home, in perfect sync and controllable from your phone, has been around for a few years now. Many solutions are available, from simple Bluetooth speakers to high-end systems like Sonos. Often the price can be off-putting and there has been recent controversy over the closed nature of these systems. Can we build a whole-house audio system using nothing but open-source technology and our favourite little computer? This month, we'll start by building a dedicated music-playback station that can produce audiophile-level sound and web-based control.

01 Hardware assembly

The JustBoom DAC (digital-to-analogue converter) HAT provides high-quality audio output for our project. It also blocks the GPIO header, which we need to power the touchscreen. Luckily, the HAT has a GPIO breakout, so we soldered a two-pin header to the 5 V and GND for the touchscreen so the whole system can be powered by a single connector. You can also use two power supplies if you prefer. Mount the DAC HAT on the GPIO pins and secure. We used 20 mm 2.5M screws to secure the HAT to the computer and screen controller standoffs. Connect the 5 V and GND pins for the touchscreen to their partners on the DAC HAT.



Iris is a touchscreen-friendly, web-based interface for Mopidy. It supports add-ons for popular platforms such as Spotify and Soundcloud

02 Prepare Raspberry Pi OS

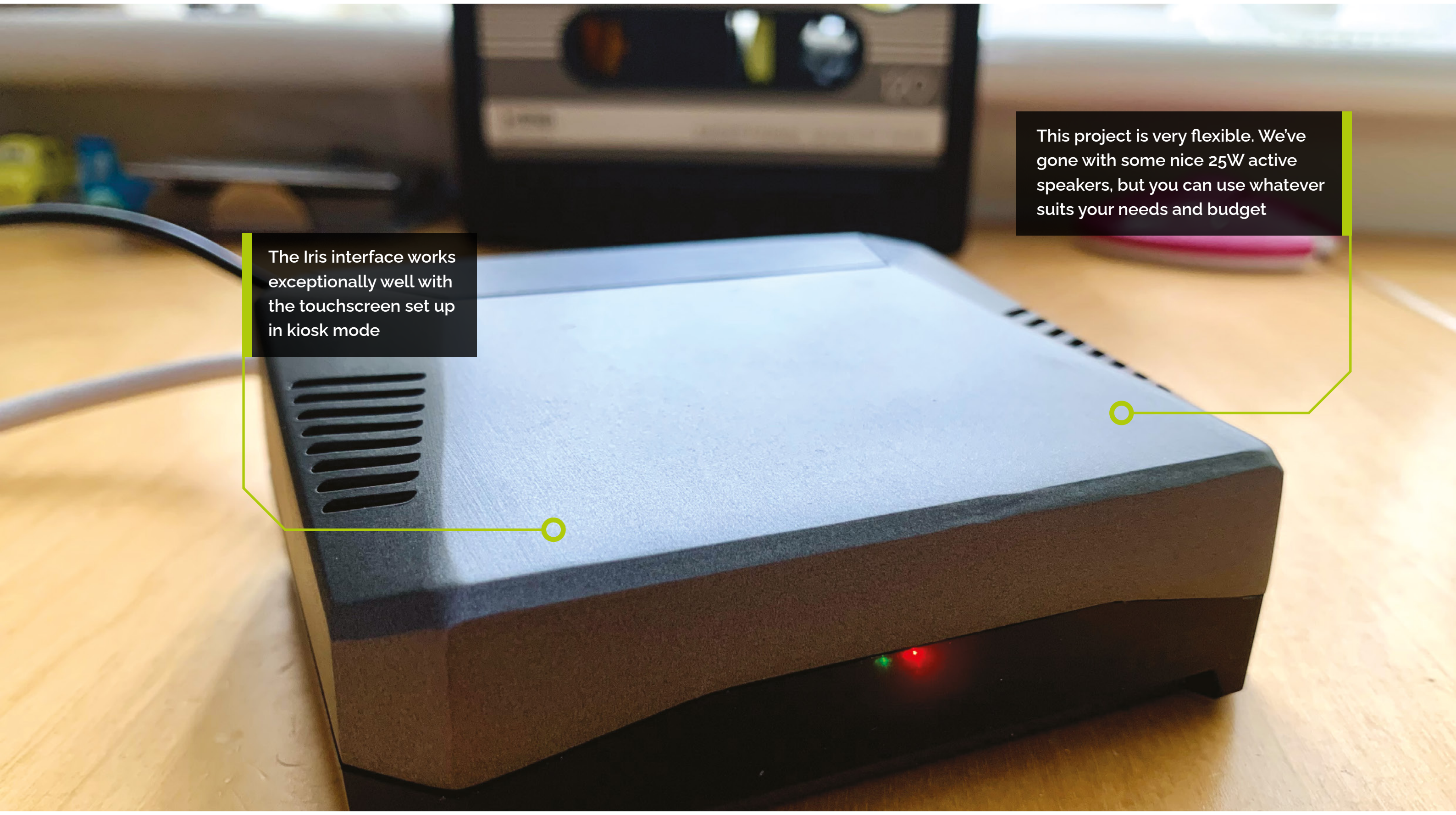
The case prevents access to the microSD card, so make sure you've got a basic setup of Raspberry Pi OS before assembling everything. For this project we're going to be using Chromium, so we need the 'standard' OS image with Desktop (but we don't need the Recommended Software version). Burn this image with your preferred method and make sure you're connected to the network with SSH enabled, or hook up a mouse and keyboard if you prefer. Finally, as ever, make sure you're running the latest version of everything with `sudo apt update && sudo apt upgrade`.

03 Mount into the case

The Raspberry Pi 4B case used here will not accept the assembly with the DAC HAT. If you're not using a HAT, you can skip this step. Otherwise, carefully remove the two wall-mounting screw holes and then line up the HAT with the top ventilation spacers and remove any that are blocking the two phono sockets and the 3.5 mm socket from sitting flush in the case. Measure twice and cut once! The plastic is soft, so it's fairly easy to do. Once the assembly is sitting comfortably in the case, you can screw in the touchscreen and it should now sit up securely.

You'll Need

- JustBoom DAC HAT **magpi.cc/justboomdac**
- Official Raspberry Pi 7" Touchscreen **magpi.cc/touch**
- Touchscreen Case **magpi.cc/touchcase**
- Active speakers, e.g. **magpi.cc/r1000t4**



This project is very flexible. We've gone with some nice 25W active speakers, but you can use whatever suits your needs and budget

The Iris interface works exceptionally well with the touchscreen set up in kiosk mode

04 Enable DAC output

No sound will come from the DAC HAT by default. Luckily, the required drivers are already installed and waiting to be enabled. From the command line, run the following:

```
sudo nano /boot/config.txt
```

Find the line that reads:

```
dtparam=audio=on
```

Remove the line (or add a # to the front to comment it out), then add these lines:

```
dtparam=audio=off
dtoverlay=justboom-dac
lcd_rotate=2
```

That last line flips the screen if you've found, as we did, that the touchscreen display is now upside down. Reboot your Raspberry Pi computer and from the command line, test audio output:

```
aplay /usr/share/sounds/alsa/Front_Center.wav
```

You should hear a calm voice say 'front centre'.

> “Luckily, the required drivers are already installed and waiting to be enabled”

05 Install Mopidy

Mopidy is a headless music server. This means it has no user interface of its own, so we'll add one later. To install Mopidy, we need to add its repository to APT. Run the following commands one-by-one:

```
wget -q -O - https://apt.mopidy.com/mopidy.gpg | sudo apt-key add -
sudo wget -q -O /etc/apt/sources.list.d/mopidy.list https://apt.mopidy.com/buster.list
sudo apt update
```

Now install Mopidy and set its user's permissions:

```
sudo apt install mopidy
sudo adduser mopidy video
```

Set Mopidy to run as a background service on boot:

```
sudo systemctl enable mopidy
```

Top Tip

New album blues

New music not appearing? Go to Settings and then click 'Run Local Scan'. If tracks still don't appear, click 'Reset Cache' on the same screen.

Top Tip

Keep it simple

Change things to suit your budget. You can run Mopidy on a full-size Raspberry Pi with the built-in headphone socket.

06 Configure and test Mopidy

Before Mopidy can be run, we need to configure a few things. The default settings are sensible, but if you want to access the web interface (once installed) from another device, we'll need to allow that.

```
sudo nano /etc/mopidy/mopidy.conf
```

After the comments block, add this:

```
[http]
hostname = 0.0.0.0

[audio]
output = alsasink
```

Save the file with **CTRL+X** and then start Mopidy:

```
sudo systemctl enable mopidy
```

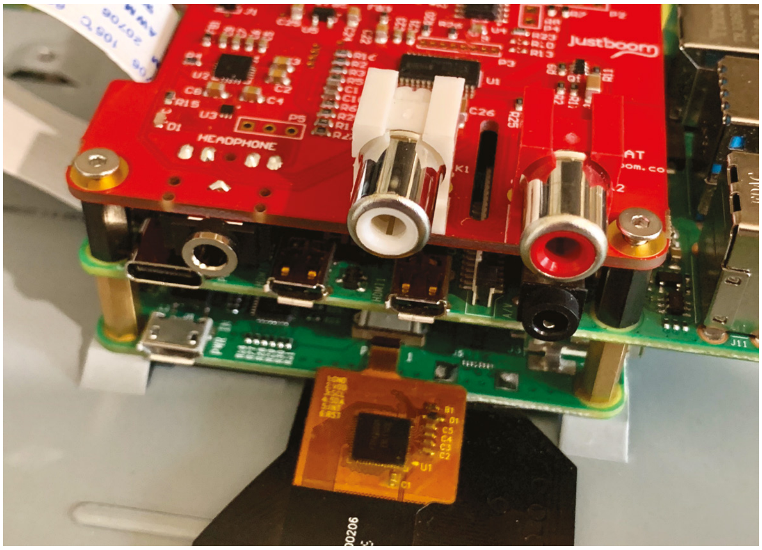
In a browser, go to **http://<hostname>:6680/** (replace <hostname> with your Raspberry Pi's name – 'raspberrypi.local' by default). If you get a welcome page displayed, all is working well.

07 Install the UI & file support

Next, we're going to install Iris, a user interface, and support for playing music stored on the local system. Run these commands:

```
 sudo pip3 install Mopidy-Iris
 sudo pip3 install Mopidy-Local
 sudo sh -c 'echo "mopidy ALL=NOPASSWD:
/usr/local/lib/python3.7/dist-packages/
mopidy_iris/system.sh" >> /etc/sudoers'
 sudo systemctl restart mopidy
```

This installs the Iris and Local modules, then gives Iris permission to run Mopidy commands as the superuser. Finally, we restart Mopidy to pick up the changes. In a web browser, go to **http://<hostname>:6680/iris/** to see the interface.



The DAC HAT provides high-resolution output at line level. You can swap this with an amplifier HAT and use passive speakers instead

08 Just add music

Mopidy supports a wide range of formats, including MP3 and FLAC. Whatever you decide to use, create a space for your music and upload some files there. You'll find a **Music** folder already exists in your home directory. We need to tell Mopidy where to look for your files. Edit the Mopidy configuration file:

```
sudo nano /etc/mopidy/mopidy.conf
```

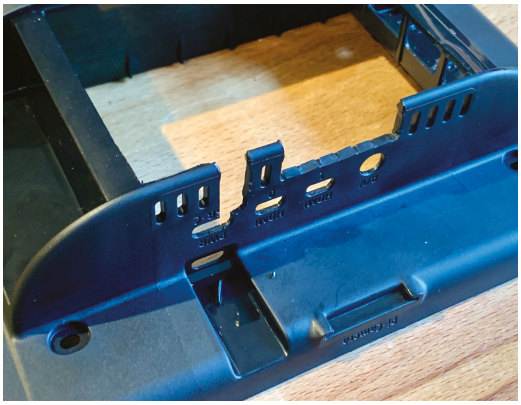
Add append the following:

```
[local]
media_dir = /home/pi/Music
```

Restart Mopidy to pick up the configuration changes and then initiate the first music scan:

```
sudo mopidyctl local scan
```

Now in Iris, you should be able to see your albums appear. Try playing one!



We've modified the case and removed the back so everything fits. The screen takes its 5 V feed from the GPIO breakout on the DAC HAT

09 Kiosk mode

To get Iris running on the touchscreen, we can make Chromium run in 'kiosk' mode so that no other elements of Raspberry Pi OS Desktop can be seen. The first step is to install 'unclutter', a handy app for hiding the mouse pointer when not in use.

```
sudo apt install unclutter
```

Now we'll create a script to launch Chromium with all the correct settings:

```
nano ~/kiosk.sh
```

Add the code shown in the listing. When run, this will enable cursor hiding, prevent some Chromium error messages from being displayed, and then start Chromium in kiosk mode, going directly to the Iris website in full-screen.

kiosk.sh

Language: **Bash/Shell**

DOWNLOAD THE FULL CODE:

magpi.cc/iriskiosk

```
#!/bin/bash
unclutter -idle 0.5 -root &
sed -i 's/"exited_cleanly":false/"exited_cleanly":true/' /home/pi/.config/chromium/Default/Preferences
sed -i 's/"exit_type":"Crashed"/"exit_type":"Normal"/' /home/pi/.config/chromium/Default/Preferences
/usr/bin/chromium-browser --noerrdialogs --disable-infobars --kiosk http://127.0.0.1:6680/iris &
```

10 Implement the kiosk script

Before we can use the script, it needs to be flagged as an executable:

```
chmod +x ~/kiosk.sh
```

Now try it out by running `~/kiosk.sh`. If you're entering commands remotely via SSH, use this command instead:

```
DISPLAY=:0 ~/kiosk.sh
```

All being well, Iris should start full-screen and be controllable by touch. We recommend going into settings and enabling 'Wide scrollbars' to make it easier to control with fingers. You can also choose between light and dark themes, the darker theme being a little kinder on the eyes when used with the touchscreen.

11 Run on boot

The final step for our music system is to make it run on boot and go straight into kiosk mode. We'll ask the Desktop itself to do this. Enter the following in the command line:

```
sudo nano /etc/xdg/lxsession/LXDE-pi/
autostart
```

Add this line, including the '@' sign:

```
@/home/pi/kiosk.sh
```

Save the file and reboot your system. The desktop will appear and then Iris will boot full-screen. Your touchscreen music system is now ready for use. You can control everything from either the touchscreen, a web browser on any device, or your smartphone.

12 Load up the music

If you want to add some audiophile-grade music to your collection, you can use your Raspberry Pi computer to rip CDs to FLAC, a lossless compression system. The files are large, but no music detail is lost. A very easy-to-use ripper is 'A Better CD Encoder' aka 'abcde'.

```
sudo apt install abcde
```

Hook up a CD drive to the USB port, insert a CD, and enter this:

```
abcde -o flac
```

Abcde will guide you through the whole process, including automatically naming the tracks. Place the resulting directory in your **Music** directory and go to Settings in Iris to initiate a rescan.

Mopidy, with a little help from its friends, can play music from many different services, including Spotify. In the next tutorial, we'll explore how.

*We'd like to acknowledge PiMyLifeUp for their excellent tutorial on kiosk mode: **magpi.cc/pikiosk**.*

Part 02

Ultimate home music: services

Extend the capabilities of your music system with Mopidy extensions and access a world of audio

MAKER

PJ Evans

PJ is a writer, software engineer and can't decide whether to arrange his collection by genre, title, or year of release.

@mrpjevans

Previously, we built a standalone music server using a DAC (digital-to-analogue converter) HAT, Mopidy, Iris, and a Raspberry Pi touchscreen. All these things, together with a pair of decent speakers, guarantee audio bliss for your home or workplace. Now we're going to extend the capabilities of Mopidy using its plug-in architecture. With a few commands, Mopidy can be extended to stream audio from a wide range of online services, from giants like Spotify to the gargantuan collections of the Internet Archive. This month we're looking at the best Mopidy extensions and how to install them.

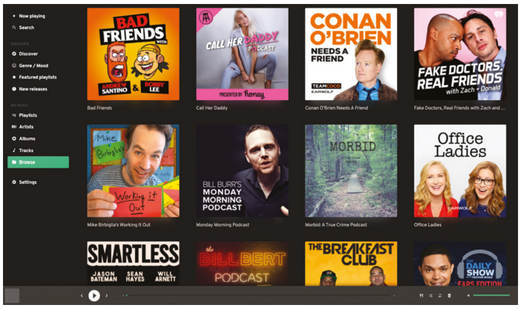
01 Local library

Let's start by looking again at our local collection. Storing music locally allows you reliable playback and complete control over how your music is organised. The downside is storage costs, so before embarking on ripping all your CDs, have a think about how you want to store them. Last month we discussed using abcde (A Better CD Encoder) to rip to FLAC, which produces large files with no quality loss. You can also consider using Ogg Vorbis. This open-source codec does involve quality loss, but produces much smaller files, and there's every chance you won't be able to tell the difference. To rip a CD to Ogg Vorbis:

```
abcde -o ogg
```

02 Tag, you're it

Many CD rippers, abcde included, will use online services to find information about your CD and automatically name files after the artist, album, and track name. They will also record that information inside the file itself so servers like Mopidy can display details about what's playing. This is called *tagging*. Unfortunately, the online services don't always get it right, so if you find a tag is misnamed or contains mistakes, you can use Kid3 to set the tags yourself. To install from the command line:


BAD FRIENDS
CALL HER DADDY
CONAN O'BRIEN NEEDS A FRIEND
FAKE DOCTORS, REAL FRIENDS
Office Ladies
MORBID
BILL BURR'S MONDAY MORNING PODCAST
SMARTLESS
THE BREAKFAST CLUB
DAILY SHOW

Listen to all your favourite podcasts directly from Mopidy and discover new shows with the iTunes Podcast catalogue

```
sudo apt install kid3
```

This desktop app provides easy bulk retagging of files and can even import artwork.

> " If you find a tag is misnamed, you can use Kid3 to set tags yourself "

03 Figuring configuring

In the following steps, we'll be setting up a number of online services to interface directly with your Mopidy installation. In many cases, you'll need to make changes to the Mopidy config file,

Warning! Copyright

Where you live may or may not allow the copying of data from a CD. Check here for more info:

magpi.cc/ripping

You'll Need

- Mopidy Music Server **magpi.cc/mopidy**

Adding Spotify to your Mopidy server gives you access to literally millions of recordings

Iris's clever interface allows you to search across many different services at once

so it's worth explaining how it's formatted. Each extension is named 'Mopidy-Extension'. In the config file located at **/etc/mopidy/mopidy.conf**, *Extension* becomes `[extension]` and any relevant options go underneath that line in the form 'key = value'. For example:

```
[local]
media_dir = /home/pi/Music
```

We've provided an example config file to help make sure things are properly set up.

04 Spot on

Now your local collection is set up, let's move on to the first of our extensions. Spotify is an amazing collection of music. Wouldn't it be great to be able to access not only Spotify's archive but also your playlists and tagged artists? If you've got a Spotify Premium account, you can do just that. Start by installing the extension from the command line:

```
sudo apt install mopidy-spotify
```

We now need to authorise Mopidy to use your account by visiting **magpi.cc/mpdyspotify**. Once you're authorised, a config file will be displayed. Cut and paste this into **/etc/mopidy/mopidy.conf** and add your Spotify username and password. See the code listing overleaf for an example.

Now restart Mopidy:

```
sudo systemctl restart mopidy
```

Go back to Iris and the world of Spotify is literally at your fingertips.

05 Radio waves

How about adding radio stations to your Mopidy setup? There is easy and free access to thousands of radio stations streaming across the planet, catering for all interests. We have the option of two extensions for playing radio: TuneIn and RadioNet. The TuneIn extension provides access not only to live stations, but also historical recordings and is more suited to the curious browser rather that just getting to your favourite station. RadioNet is a simpler but easier-to-navigate service that allows you to create a list of favourite stations. To install:

```
sudo apt install mopidy-tunein
sudo pip3 install Mopidy-RadioNet
sudo systemctl restart mopidy
```

Top Tip

Lossless or lossy?

Lossless (FLAC) compresses audio with no loss of quality. Lossy (Ogg Vorbis, MP3) removes audio information to produce smaller file sizes.

Top Tip

Enabled or disabled?

Every extension supports 'enabled' in its config. If you want to remove something temporarily, just add 'enabled = false' to the extension's block in **/etc/mopidy/mopidy.conf**.

06 Casting out

If you prefer your radio programmes 'on demand', you're probably a fan of podcasts. You've got two options for listening to your favourite shows on Mopidy. If you like to subscribe direct to podcast RSS feeds, then it's as easy as this:

```
sudo pip3 install Mopidy-Podcast
sudo systemctl restart mopidy
```

To use it, you need to point to an OPML file (a list of podcast locations), as shown in the listing. Most podcast clients can generate this for you. If you prefer to browse a huge catalogue of programming, then additionally install the iTunes plug-in:

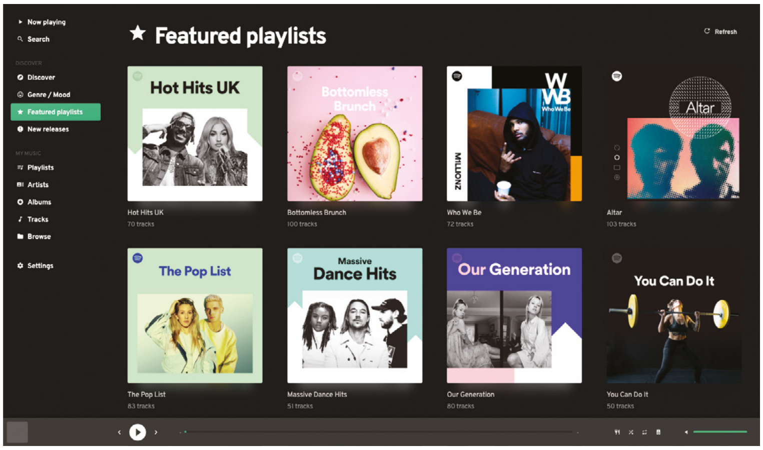
```
sudo pip3 install Mopidy-Podcast-iTunes
sudo systemctl restart mopidy
```

Now you can use Browse to find your favourite show and listen on demand.

07 SomaFM

SomaFM was one of the early pioneers of internet audio streaming. Working through a donations model, it has expanded to over 30 channels, all with the theme of ambience but covering different genres from electronica to jazz. Often SomaFM makes for perfect background music. To add SomaFM to Mopidy (you've probably spotted a pattern emerging by now):

```
sudo pip3 install Mopidy-SomaFM
sudo systemctl restart mopidy
```



▲ Add audio services, like Spotify, SomaFM, and YouTube for a fully-fledged modern music system

08 SoundCloud

SoundCloud is an excellent online showcase of music and other forms of audio. As well as providing a way for bands to sell direct to their fans, it's a platform for new and upcoming material from unsigned musicians. You can browse SoundCloud's vast collection from Mopidy, but as with Spotify, you need to have an account (it's free to join) and authorise your installation.

First, visit **mopidy.com/ext/soundcloud** and follow the instructions to authorise Mopidy. Copy and paste the resulting config file, and add it to **/etc/mopidy/mopidy.conf**. Now install the SoundCloud extension:

```
sudo pip3 install Mopidy-SoundCloud
sudo systemctl restart mopidy
```

> "You can browse SoundCloud's vast collection from Mopidy"

09 YouTube

The ubiquitous YouTube contains countless hours of audio-specific content. Either listen to one of the many audio recordings uploaded or just the soundtrack of a favourite video. Again, the standard installation pattern applies:

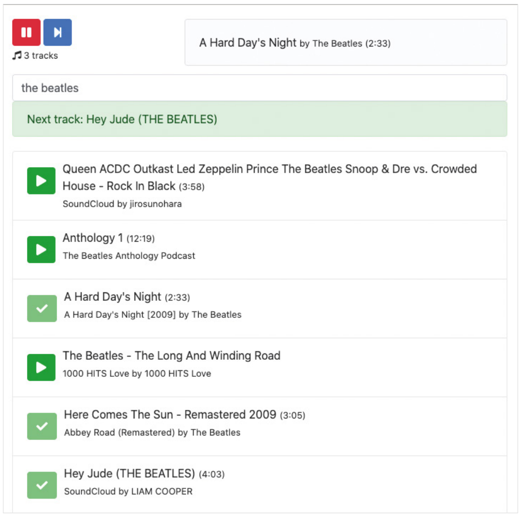
```
sudo pip3 install Mopidy-YouTube
sudo systemctl restart mopidy
```

One difference with this extension is the need to specifically enable it in **/etc/mopidy/mopidy.conf**. Make sure this is in the file or YouTube files will not appear in search results:

```
[youtube]
enabled = true
```

10 Digging through the archives

The Internet Archive (**archive.org**) is one of the crowning achievements of the internet. This vast resource aims to preserve anything that can be expressed in digital form. Unsurprisingly, its audio collection is vast, containing thousands of hours or public domain recordings, from vintage music to public speeches, documentaries, and a massive



3 tracks

A Hard Day's Night by The Beatles (2:33)

the beatles

Next track: Hey Jude (THE BEATLES)

Queen ACDC Outkast Led Zeppelin Prince The Beatles Snoop & Dre vs. Crowded House - Rock In Black (3:58)
SoundCloud by jirosunohara

Anthology 1 (12:19)
The Beatles Anthology Podcast

A Hard Day's Night (2:33)
A Hard Day's Night [2009] by The Beatles

The Beatles - The Long And Winding Road
1000 HITS Love by 1000 HITS Love

Here Comes The Sun - Remastered 2009 (3:05)
Abbey Road (Remastered) by The Beatles

Hey Jude (THE BEATLES) (4:03)
SoundCloud by LIAM COOPER

Mopidy-Party lets your friends choose the music, which may or may not be a good thing

catalogue of Grateful Dead recordings! You can access all of these from your Mopidy system:

```
sudo pip3 install Mopidy-InternetArchive
sudo systemctl restart mopidy
```

We recommend using the 'Browse' feature of Iris to navigate this resource. Search results can be overwhelming!

11 Party time, excellent!

You now have access to a mind-boggling array of music and audio. Time to celebrate. We've been using Iris as our interface to Mopidy, but in fact you can install different interfaces and even run them concurrently. If you're having a few friends around, Mopidy-Party allows you to safely give everyone access to your music collection from their phones and add their picks to a playback queue. Now, who put Showaddywaddy on?

12 Make your own

Something missing from your collection? Is there no access to the latest and greatest streaming service? If you're handy with a bit of Python coding, then you're free to make your own Mopidy extensions and share them with users all over the world. Not only have the Mopidy developers provided detailed documentation at **magpi.cc/mpdyext**, they have also provided a 'cookiecutter' project at **magpi.cc/mpdycookie** which will get you up and running in no time. Contributions to improve existing extensions are also welcomed.

mopidy.conf

DOWNLOAD THE FULL CODE:

magpi.cc/modipyconf

Language: **Config file**

```
[http]
# Make sure the web interface can be accessed by the local network
hostname = 0.0.0.0

[audio]
# Direct audio to the DAC, not HDMI
output = alsasink

# The File extension isn't of much use, so we can disable it
[file]
enabled = false

# If enabled, this adds support for M3U playlists
[m3u]
enabled = false

# Location of local music
[local]
media_dir = /home/pi/Music

# Replace with the information given when you authorise
# then add in you Spotify username and password
[spotify]
username = <Spotify username>
password = <Spotify password>
client_id = <Client ID>
client_secret = <Client secret>

# Getting to many results in TuneIn? Add this so you only get stations.
[tunein]
filter = station

# Once authorised, add the provided token here
[soundcloud]
auth_token = <Auth token>

# Most extensions are enabled by default. You must explicitly enable YouTube.
[youtube]
enabled = true
```

Part 03

Ultimate home music: multi-room

You've built your music system and enhanced it with streaming services, now let's fill your home with sound

MAKER

PJ Evans

PJ is a writer, software engineer, and now has a house with too many speakers.

@mrpjevans

In the previous two tutorials, we've built and configured an all-singing, all-dancing, tricked-out music system using a Raspberry Pi computer and Mopidy, the Python-based extendable music server. We can listen to our music in beautiful FLAC quality and stream from the worlds of Spotify, SoundCloud, and YouTube. There's one amazing feature that systems such as Sonos offer that we haven't yet covered: multi-room audio. Not just playing music in different rooms, but also playing in perfect sync. In this final part, we'll do just that and learn a few new Mopidy tricks along the way.

01 Going multi-room

Sure, playback from our Mopidy setup is great, but you don't want to be carrying that setup around the house. With Snapcast we can play music anywhere in perfect sync, so you can wander around your home without interruption. This clever piece of open-source software sends out audio in 'frames', each one with a time code attached. Any device that's part of the stream matches the frame's time code to its own internal clock to ensure playback happens at the same time, providing in-sync audio. The only downside is a short delay in starting playback as everything syncs up.

You'll Need

- Mopidy Music System
- Hi-Berry DAC+Zero **magpi.cc/hifiberrydaczero**
- Case (optional) **magpi.cc/hifiberrycase**

Hiding inside are a Raspberry Pi Zero W and HiFiBerry DAC+ Zero digital-to-analogue converter

Connect this box to an amplifier or active speakers for synchronous whole-house audio

02 Snap to it

Before we can add Snapcast clients, our original Mopidy needs to become one itself, so it can keep in sync with everything else. We start by installing the Snapcast client and server on the same machine (it's effectively streaming to itself). Enter the following on the command line to download the client:

```
  get https://github.com/badaix/
snapcast/releases/download/v0.20.0/
snapclient_0.20.0-1_armhf.deb
  sudo dpkg -i snapclient_0.20.0-1_armhf.deb
```

If you get an error here, run this to fix it:

```
  sudo apt -f install
```

This automated install sets everything up and will restart the service on reboot.

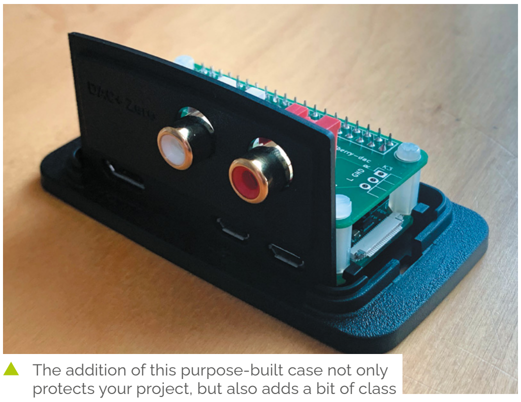
03 Snappy service

We have the client running on your Mopidy system, but nothing to serve music. So, now install the server:

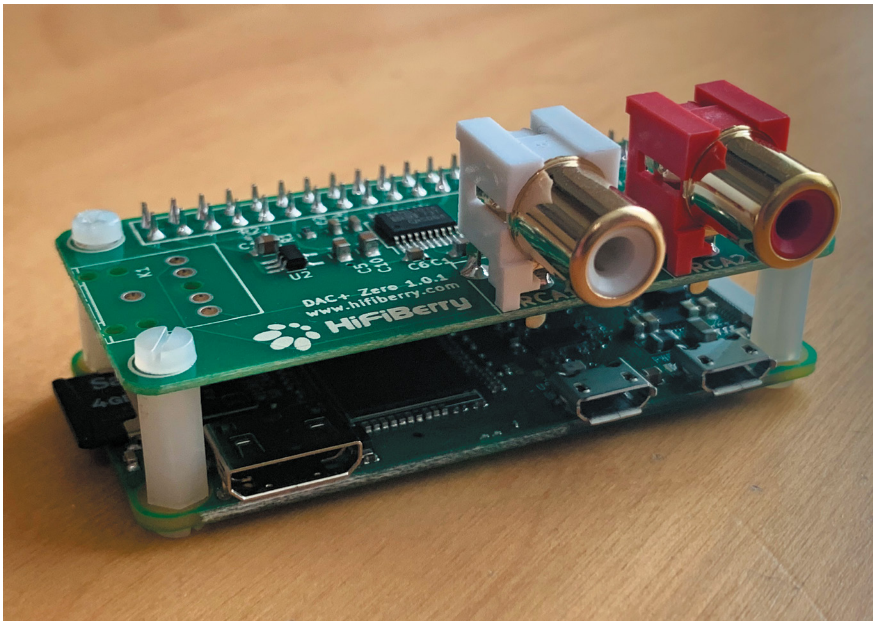
```
  get https://github.com/badaix/
snapcast/releases/download/v0.20.0/
snapclient_0.20.0-1_armhf.deb
  sudo dpkg -i snapclient_0.20.0-1_armhf.deb
```

This will also restart on boot. The client will automatically find the server as its local. Now tell Mopidy to send its audio stream to the Snapcast server instead of the DAC.

```
  sudo nano /etc/mopidy/mopidy.conf
```



The addition of this purpose-built case not only protects your project, but also adds a bit of class



The HiFiBerry DAC is an entry-level digital-to-analogue converter that provides much higher quality than the standard audio-out from a Raspberry Pi

Change…

```
  [audio]
  output = autoaudiosink
```

…to (the second part on one line):

```
  [audio]
  output = audioresample ! audioconvert !
audio/x-raw,rate=48000,channels=2,format=S16
LE ! wavenc ! filesink location=/tmp/snapfifo
```

Now restart Mopidy:

```
  sudo systemctl restart mopidy
```

04 Set me up

Iris comes with full control over the Snapcast system. After restarting Mopidy, go to the Iris interface and into Settings. You'll see a Snapcast icon. Click on it and then click 'Enable'. You should see a 'Connected' message appear. There will also be a group which represents your local Mopidy setup. Click the group to rename it to something memorable. Before proceeding further, make sure that playback still works. It'll take a second or two longer to start as Snapcast syncs up, but should otherwise be unaffected. If it works well, your local Mopidy setup is complete.

Top Tip

Choose your DAC

There are a wide range of digital audio convertors out there catering for every budget, from £12 to thousands – choose wisely.

▲ The case includes access to the HDMI and USB ports, so you can connect a touchscreen monitor and use it as a Mopidy control centre

▶ You can use Iris on any web device to control which group is listening to which stream

05 Just add clients

Your system is now in effect streaming to itself, which means it can play in sync with other devices, so let's add one. We're using the HiFiBerry DAC+Zero, a great DAC for a small price. Start by connecting the DAC to a Raspberry Pi Zero W, ideally using standoffs to ensure a secure fit. Install Raspberry Pi OS Lite on this device as we're going to be running it headless. Using raspi-config, make sure you've configured WiFi and set a suitable hostname. Now update everything to the latest version using:

```
sudo apt -y update && sudo apt -y full-upgrade
```

06 Make some noise

To enable the DAC+ Zero, obtain command-line access to your Raspberry Pi Zero W (using SSH or connect up a monitor and keyboard) and edit the main configuration file:

```
sudo nano /boot/config.txt
```

Near the end of the file, find the line reading:

```
dtparam=audio=on
```

Delete it (or comment out with a #) and add:

```
dtoverlay=hifiberry-dac
```

Save (**CTRL+X**) and reboot your computer. Once back up, connect the phono sockets on the DAC to an amplifier and test the output:

```
aplay /usr/share/sounds/alsa/Front_Center.wav
```

If everything is working well, a nice person will say "front centre".

07 Join the band

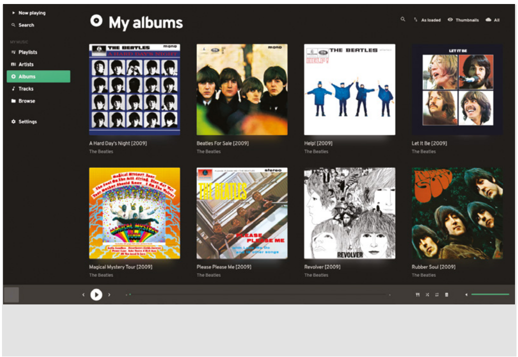
To get streaming working, we now repeat the process for installing the Snapcast client. Start by downloading and installing the client:

```
 wget https://github.com/badaix/
snapcast/releases/download/v0.20.0/
snapclient_0.20.0-1_armhf.deb
 sudo dpkg -i snapclient_0.20.0-1_armhf.deb
```

If you see errors at the end of the process with the line 'dependency problems – leaving unconfigured', run the following command:

```
sudo apt -f install
```

This will detect and install all the dependencies required and then finish configuration. Snapcast will be configured to automatically start on boot.



08 Configure and test

The Snapcast client is now running, but we need to tell it where the server is. Edit the configuration file:

```
sudo nano /etc/default/snapclient
```

Find the line that reads `SNAPCLIENT_OPTS=""` and add your Mopidy server hostname as follows:

```
SNAPCLIENT_OPTS="--host jukebox.local"
```

Replace 'jukebox.local' with whatever you named your server. Save and exit, then restart the client:

```
sudo systemctl restart snapclient
```

In a web browser, open up Iris on your main server and go to Settings, then click on Snapcast. You should see a new group (something like 'Group 8ec'); that's your device. You can rename it if you wish.

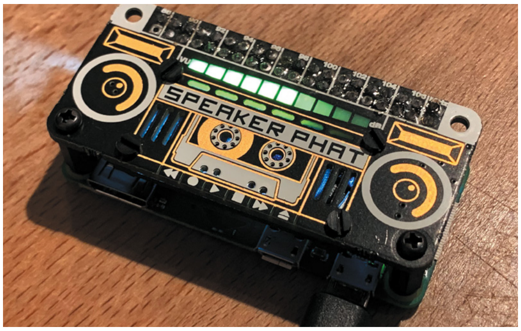
Warning! Copyright

Where you live may or may not allow the copying of data from a CD. Check here for more info:
magpi.cc/ripping

09 Play, tweak, repeat

Now try playing something from Mopidy with your new client hooked up to an amplifier or active speaker. Everything should be in sync. If not, you can adjust latency under your group settings to fine-tune the playback. At the bottom of the screen, you can click the speaker icon to control which devices are playing and set their individual volume levels.

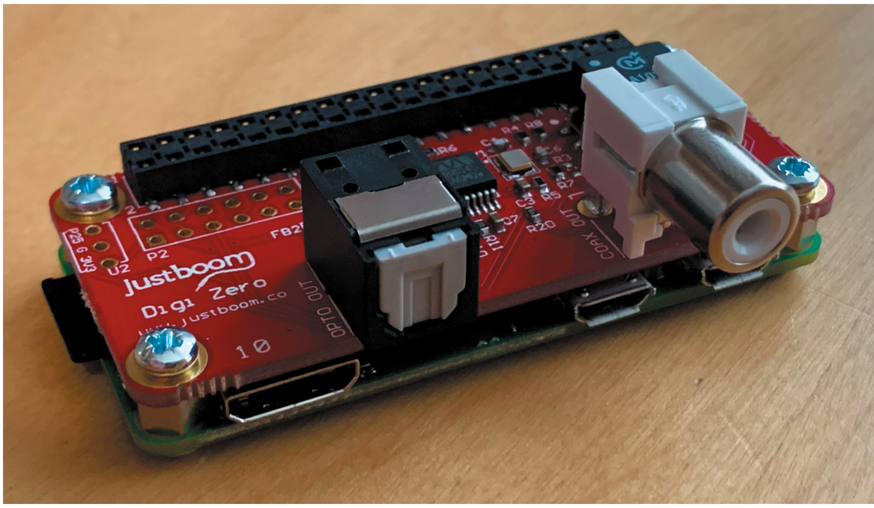
You can add as many Snapcast clients as you like. It's a great use for an older Raspberry Pi and you don't have to use a DAC, you can just use the standard audio/video jack to an active speaker. We built another client using Pimoroni's Speaker pHAT and a battery for portable tunes.



10 Add some Apple

Snapcast supports multiple streams, allowing clients to switch between them. We can add Apple AirPlay 2 support as a Snapcast stream that runs alongside Mopidy. We can't just install a package, though: we have to build and install Shairport Sync with the following commands:

```
 cd
 sudo apt install build-essential git
xmltoman autoconf automake libtool libpopt-
dev libconfig-dev libasound2-dev avahi-daemon
libavahi-client-dev libssl-dev libsoxr-dev

 git clone https://github.com/mikebrady/
shairport-sync.git

 autoreconf -fi

 ./configure --with-stdout --with-avahi
--with-ssl=openssl --with-metadata

 make

 sudo make install
```



11 Multistream!

The final step is to configure Snapcast to enable Shairport Sync as a stream. Open up the Snapcast configuration file as follows:

```
 sudo nano /etc/snapserver.conf
```

In the section starting `[stream]` you'll see an entry starting `[stream] =`. Directly under there, add the following line:

```
 stream = airplay:///shairport-sync?name=
Airplay&devicename=Jukebox
```

(You can change the names to anything you like.) Now restart everything:

```
 sudo systemctl restart snapserver.service
 sudo systemctl restart mopidy.service
```

In Iris, go to the Snapcast settings and change the default stream to AirPlay. You can now push audio from iOS devices and Macs to your music system or any other Snapcast client.

▲ If your home audio setup features coax or S/PDIF digital interfaces, this JustBoom adapter is perfect and can be set up identically to the DAC+ Zero

◀ You can have a lot of fun with Snapcast. We've used a Pimoroni Speaker pHAT here for the ultimate in portable tunes

12 Going further

If you've played along and built this setup, you now have a pretty sweet audio player setup. The great thing about this project is the wider range of budgets and platforms for which it caters. You can reuse older hardware, upcycle speakers, and turn just about anything with a processor into a streaming client. You can also get HATs with pure digital S/PDIF or coaxial output so you can use an amplifier's DAC if you prefer. How about using a USB audio capture device to stream audio around the house from your record player? As ever, it's over to you.

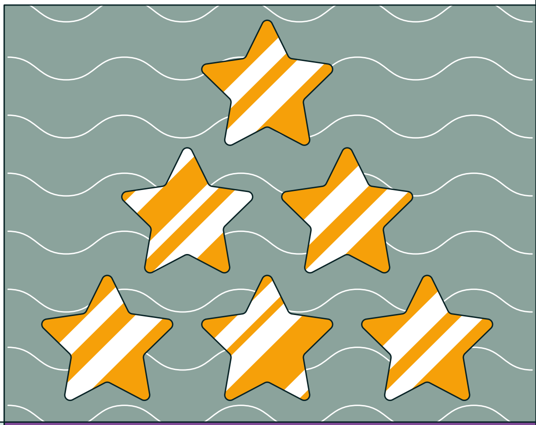
Top Tip

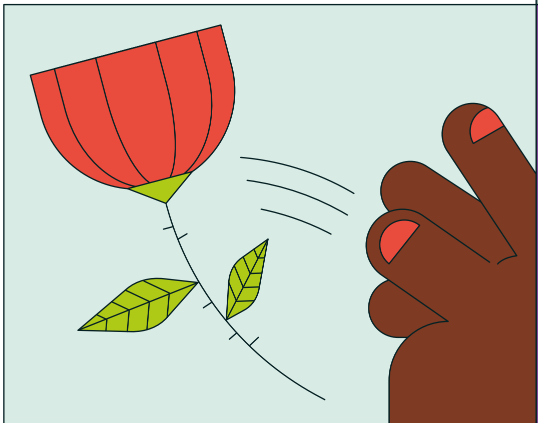
DAC's not for me

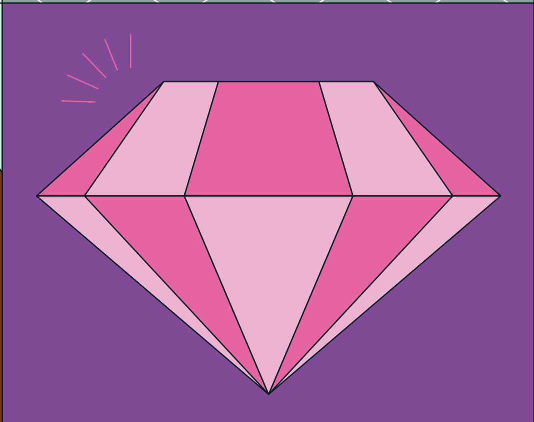
If you're on a tight budget, you don't need a DAC at all. The 3.5 mm AV jack found on most Raspberry Pi computers will suffice for smaller projects.

Reviews

When it comes to hardware and software, it can be tricky to figure out what the best item for the job is. Let us show you some of the best Raspberry Pi accessories that will help improve your media centre project.







1

Argon ONE **M.2**

SPECS

DRIVE TYPE: M.2 SATA SSD with Key-B or Key-B&M

DISPLAY PORTS: Dual full-size HDMI

FEATURES: UASP support, power management, active and passive cooling

▶ Argon 40 ▶ **magpi.cc/argononem2** ▶ £43 / $45

Add a super-fast M.2 SATA SSD to Raspberry Pi 4 along with dual full-sized HDMI sockets, a power button, and fan cooling. By **Lucy Hattersley**

The Argon ONE M.2 case for Raspberry Pi 4 has been on the radar for many Raspberry Pi power users. We're glad to report that your anticipation has not gone unrewarded.

The marquee addition here is support for an M.2 SATA SSD drive (the blade drives commonly found in laptops). This sits snugly in a bottom compartment and is connected to Raspberry Pi using the USB 3.0 connection (via a U-shaped adapter). It features UASP (USB Attached SCSI Protocol) support to increase performance.

In the spirit of things, we tested it with a Raspberry Pi 4 8GB RAM and Raspberry Pi OS (64-bit, **magpi.cc/64bit**).

An internal board repositions the ports to the rear, and converts the micro-HDMI ports of Raspberry Pi 4 to two full-sized HDMI sockets. GPIO remains accessible via a removable flap with a pin guide printed to the side.

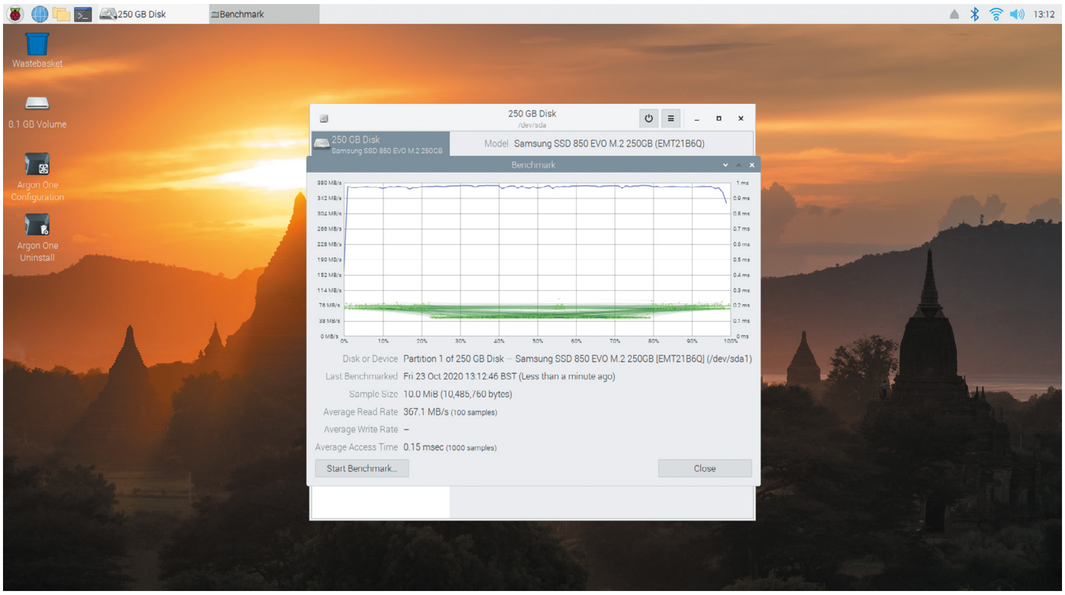
The aluminium alloy lid acts as a heatsink and a 30 mm fan provides active cooling. The fan is controlled via a script provided by Argon 40. Default settings start at 10% power when the CPU temperature is at 55°C, increasing to 55% at 60°C, and 100% at 65°C.

Putting it together

Your M.2 SATA SSD clips into the lower part of the case and is held in place with a single screw. The two halves are then screwed together and a U-shaped USB adapter is used to bridge the lower part of the case the port on the upper part.

It's an ingenious solution, but you lose a USB 3.0 socket in the final build. More importantly, the microSD card socket is sealed inside the case, requiring disassembly to access it.

Raspberry Pi EEPROM recently provided USB boot support in the default 'critical' firmware,

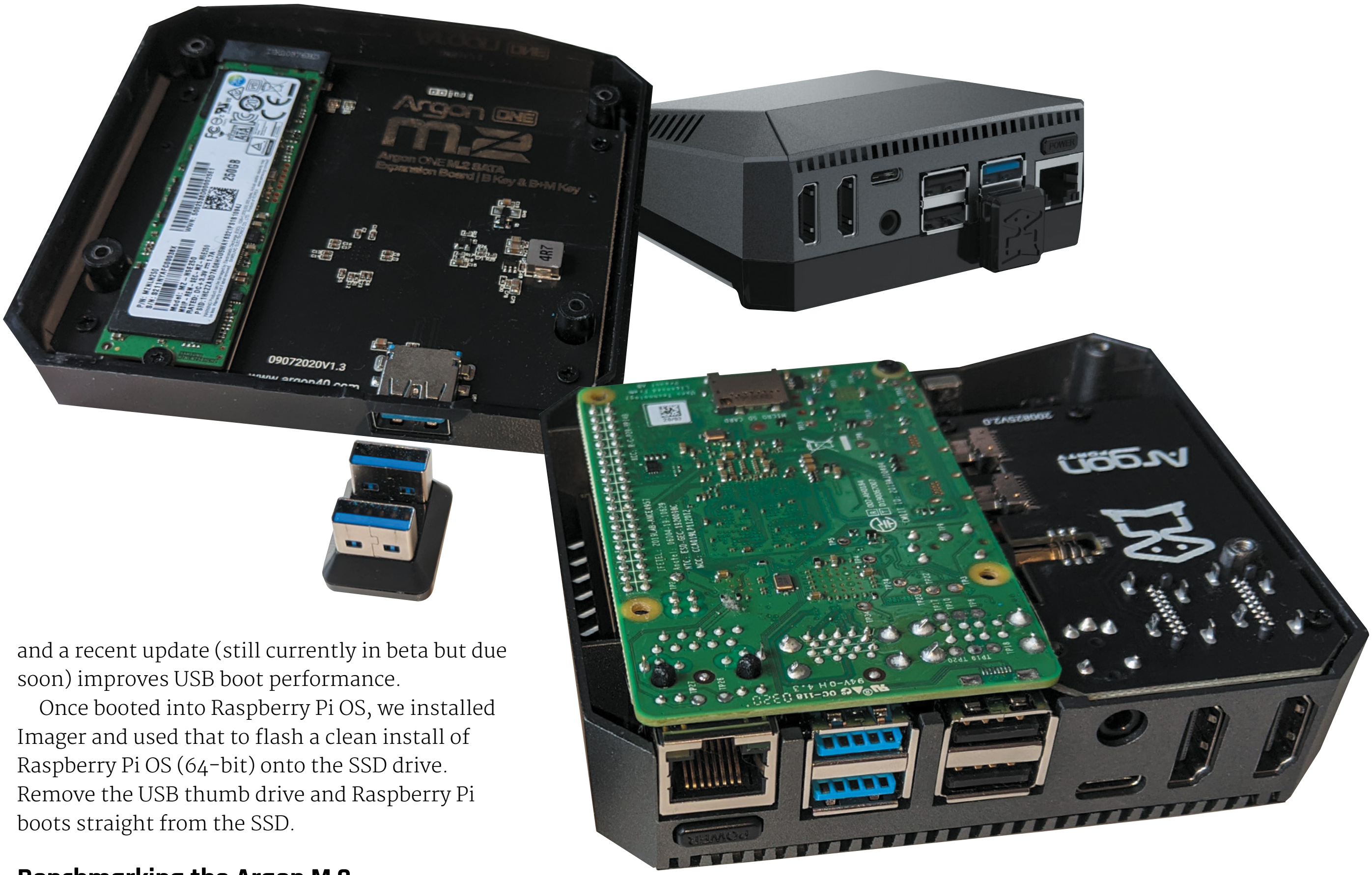


250 GB Disk
Samsung SSD 850 EVO M.2 250GB

Model Samsung SSD 850 EVO M.2 250GB (EMT21B6Q)

Disk or Device Partition 1 of 250 GB Disk — Samsung SSD 850 EVO M.2 250GB [EMT21B6Q] (/dev/sda1)
Last Benchmarked Fri 23 Oct 2020 13:12:46 BST (Less than a minute ago)
Sample Size 10.0 MiB (10,485,760 bytes)
Average Read Rate 367.1 MB/s (100 samples)
Average Write Rate –
Average Access Time 0.15 msec (1000 samples)

▶ Gnome Disk Utility benchmarking the rootfs partition of our SSD



▲ The two halves of the case connect together with a U-shaped USB 3.0 connector bridging the M.2 SSD SATA drive to Raspberry Pi 4

and a recent update (still currently in beta but due soon) improves USB boot performance.

Once booted into Raspberry Pi OS, we installed Imager and used that to flash a clean install of Raspberry Pi OS (64-bit) onto the SSD drive. Remove the USB thumb drive and Raspberry Pi boots straight from the SSD.

Benchmarking the Argon M.2

We used a Samsung 850 EVO SSD supplied by our friends at Custom PC magazine (**magpi.cc/custompc**) and installed Gnome Disks for its benchmarking test.

The M.2 did not disappoint, with an average read rate of 367.1MB/s compared to 45MB/s for our microSD card. That's an 800% speed increase.

> “You get unrestrained performance with an overclocked Raspberry Pi 4 coupled with an ultra-fast SSD storage solution”

Overjoyed with our SSD throughput rate, we set about testing CPU speed and thermal performance (the two are related as Raspberry Pi OS throttles performance if the CPU temperature exceeds 80°C).

We performed a similar test to the one Gareth Halfacree uses when thermal-testing Raspberry Pi boards (**magpi.cc/thermaltesting**). We ran stress-ng and glxgears for ten minutes while monitoring CPU speed and temperature recordings, followed by a five-minute cool-down period.

Raspberry Pi remained cool under stress, maxing out at 53°C – not even high enough for the fan to kick in, with the passive case doing all the cooling. We took our Raspberry Pi clock speed up to 1800MHz (**magpi.cc/overclock**) and, after leaving Raspberry Pi to recover, ran the test again. This time the board got a little more stressed, maxing out at 73°C – enough for the Argon fan to kick in at 100% speed, but below the critical 80°C point where Raspberry Pi OS begins to throttle back the CPU speed.

So at all times you get unrestrained performance with an overclocked Raspberry Pi 4 coupled with an ultra-fast SSD storage solution. If that's not nifty, we don't know what is.

We did also install Ubuntu, but we could not install the scripts for the fan in this case (as it depends on GPIO Zero, **magpi.cc/gpiozero**), so the fan was on permanently. Raspberry Pi OS remains our preferred operating system at any rate.

We spent a delightful couple of days knocking together the scripts for testing in Visual Studio Code and creating graphs of the results in LibreOffice, all while doing browser research and writing up the results. Our Argon ONE M.2 build remained a joy to use throughout. M

Verdict

If you want to take Raspberry Pi 4 performance as far as it can go, this is the way to do it.

10/10

▲ Flirc Raspberry Pi 4 case is a stylish affair, with an aluminium case topped and tailed with soft-touch plastic

Flirc Raspberry Pi 4

▶ Flirc ▶ **flirc.tv** ▶ £16 / $16

Stylish and clever case with a body that becomes a heat sink. By **Lucy Hattersley**

SPECS

DIMENSIONS: 66×91×25 mm

CASE: Aluminium housing

FEATURES: Built-in heat sink

We see a lot of cases here at *The MagPi*, but the Flirc Raspberry Pi 4 case came recommended to us via Raspberry Pi's engineering team, so we had to take a look.

The Flirc Raspberry Pi 4 case comes in two halves: the bottom is a soft-touch plastic shell, and the top half is milled aluminium coated with more soft-touch. The result is a stylish and lovely-to-hold case made from solid metal.

Here's where things get clever: inside the case is a protruding heat sink that reaches down to Raspberry Pi 4's CPU. This turns the whole of the aluminium case into a giant heat sink, cooling down your Raspberry Pi 4.

Inside the pack is a square thermal pad (similar in substance to Blu Tack). You use this to squidge the Raspberry Pi to the heat sink. Putting together the case is ludicrously simple: you simply drop a Raspberry Pi in the bottom half, attach the thermal paste, squidge down the lid, and use four screws to hold everything together.

Flirc claims that this is "the most beautifully crafted Raspberry Pi 4 case" and it's not a wholly unwarranted claim. It certainly has a sense of style. Everything is neatly constructed from high-quality materials and there's considerable charm to the heat sink. The microSD card slot is easily accessible, and a small cut-out on the enclosure enables the LEDs to shine through.

On the box

One downside to the sealed approach is that the GPIO pins are hidden away inside the case. Unlike the official case, the lid cannot be quickly removed to provide access to the pins.

To Flirc's credit, it has addressed this issue via a small gap on the underneath of the Flirc Raspberry Pi 4 case, which could be used with a breakout I/O cable. But it's an ungainly addition to such a lovely looking case.

If you plan to use Raspberry Pi as a desktop computer, then this might be a valid trade-off.

Warning!

Using stress should only be done for short periods. Watch your Raspberry Pi and restart afterwards.

▲ The case cunningly doubles as a heat sink, enabling you to run your Raspberry Pi 4 more coolly (and overclock to faster speeds)

However, for many of us, GPIO pins are the very essence of Raspberry Pi.

Under pressure

We stress-tested a Raspberry Pi 4 board on its own vs a Raspberry Pi 4 inside the Flirc case to see what temperatures it reported.

We used stress (`apt install stress`) and the following script from Core Electronics (**magpi.cc/7Q0oX3**) to test our Raspberry Pi 4 while measuring temperature:

```
 while true; do vcgencmd measure_clock arm;
vcgencmd measure_temp; sleep 10; done& stress
-c 4 -t 300s
```

This puts all four cores of Raspberry Pi 4 under stress. For each test, we left the Raspberry Pi to run for five minutes. Warning! Don't do this at home without doing your research first.

Unsurprisingly, the Raspberry Pi with no heat sink attached quickly went up to 79°C and hovered at that level for the rest of the test, nudging up against (but not pushing over) the level where Raspbian starts to throttle the CPU.

> “ We're going to experiment some more with overclocking, which makes this a fun case ”

Next, we put a Raspberry Pi into the Flirc Raspberry Pi 4 case and ran the same test. This time it idled at a mere 28°C and our five-minute stress test took it up to a mere 46°C.

Because this is comfortably below the threshold, it opens up a world of overclocking (something that has been reintroduced on Raspberry Pi 4 – see **magpi.cc/sED3id**).

We took the CPU clock speed up to 1.75GHz. The overclocked Raspberry Pi (inside the Flirc case) idled at 41°C, and running our five-minute stress test took it up to 67°C. Again, comfortably within a threshold. We also played around with CPU clock speeds up to 2.0GHz, which idled at 48°C and maxed out at 69°C.

We're going to experiment some more with overclocking, which makes this a fun case.

Verdict

We love the style of the Flirc case, and its heat sink opens up a world of overclocking. If only the GPIO pins remained accessible. If that's not a deal-breaker for you, though, then this is a great case to get.

8/10

NanoSound ONE Hi-Fi DAC Case

▶ Nanomesher ▶ **magpi.cc/nanosound1** ▶ £59 / $80

Combine the best of both worlds with a melding of the excellent NanoSound DACs and the Argon ONE case. **Rob Zwetsloot** grabs his listenin' ears

It's no secret that we're big fans of the Argon ONE cases – they're fantastic for keeping your Raspberry Pi cool, they look great, and they have neat features like GPIO pins under a magnetic lid and all the ports rearranged to the rear. We're also a fan of the NanoSound DACs, so we were quite intrigued when we found out about a melding of the two into one product: NanoSound ONE.

> You have your choice of bottom part for the case

On paper it's a simple change – the board attached to the roof of the case that provides a fan and heatsink is replaced with one that provides a DAC and heatsink. Because of this, the ports on the rear are slightly changed, going from two HDMI ports and a 3.5 mm jack to one micro HDMI port and two RCA jacks.

This does mean you're losing the active cooling. However, passive cooling of the heatsink (which is the entire case) is definitely good enough.

Snug fit

Installing a Raspberry Pi 4 is simple – you add the little breakout board that plugs into HDMI 0 and the 3.5 mm jack on Raspberry Pi, and then connect the GPIO pins with the GPIO header on the top of the case. The breakout board is a bit of a snug fit, so don't be worried if it looks like it's not quite spaced out correctly. It is!

You have your choice of bottom part for the case. The standard piece just encloses the whole thing and provides an SD card port, while there's also the M.2 SSD storage bay option which we reviewed last month.

The whole process is very quick, and we were able to experience some excellent sounds from the DAC part in no time, which uses the same tech as the NanoSound DAC 2. You're paying a little bit of a premium for the DAC and case together (only about $5), but you are getting a custom version of one of our favourite cases – so it's definitely worth it.

◀ Control it with a smartphone remote or the included remote

SPECS

CHIP: Texas Instruments PCM5122

SAMPLE RATE: 192 kHz

RESOLUTION: 24-bit

SIGNAL-TO-NOISE RATIO: 112 dB

AUDIO OUTPUT: RCA

Verdict

A great case with great sound at a reasonable price. This could be the one for folks who need the best audio out of a media centre.

9/10

Argon IR Remote

▶ Argon40 ▶ **magpi.cc/argonir** ▶ £9 / $10

Looking for a sleek minimalist remote for your Raspberry Pi media centre? **PJ Evans** sits back on his sofa and gets clicking

SPECS

DIMENSIONS: 150 mm × 35 mm × 10 mm

POWER: 2 × AAA batteries

COMMUNICATION: Infrared

HARDWARE COMPATIBILITY: Argon ONE & EON cases

SOFTWARE COMPATIBILITY: LibreELEC or Raspberry Pi OS

Using a Raspberry Pi 4 as a media centre makes for a brilliant way to access your favourite sounds and movies. Add Argon40's sleek ONE case and it looks as good, if not better, as any other set-top box. There's even mature software in the form of the LibreELEC operating system and Kodi media centre app. All that's missing is an easy way to control the action. The eagle-eyed may have noticed the ONE case's circuitry includes an IR sensor, and now Argon40 has made the companion Argon IR infrared remote control available.

Thankfully, Argon40 has been watching other tech design teams rather than those of most modern TVs. What you get for your very reasonable £9 is a minimalist, slim remote with just enough to make controlling Kodi a breeze. Basic directional control, selection, volume, and standard navigation work alongside a power button that works out-of-the box with any Raspberry Pi 4, regardless of operating system. It's light too, at just 65 g with two AAA batteries.

▲ The lack of extraneous buttons and a small form factor make the Argon IR comfortable to hold and use

More than media

If media centres are not your thing, or you have other epic plans for a remote control, Argon40 provides software that allows you to assign any key press to a button using the standard LIRC library with ease, making the addition of IR remote control to your project easier than ever before. In our tests, this worked flawlessly. However, we did have some issues configuring Kodi due to some missing steps in the supplied instructions. Once resolved, the remote performed admirably from across the room.

> “ A minimalist, slim remote with just enough to make controlling Kodi a breeze ”

▲ You'll need an Argon ONE case (or the forthcoming EON) to use the Argon IR

Verdict

Although the setup instructions could be improved, Argon40 has delivered a sleek, simple, and reasonably priced IR remote, both for Raspberry Pi media centres and homebrew projects.

9/10

LibreELEC **9.1 Beta**

▶ LibreELEC ▶ **libreelec.tv** ▶ Free

LibreELEC comes to Raspberry Pi 4. **Rob Zwetsloot** tests it out to see if it's ready for prime time yet

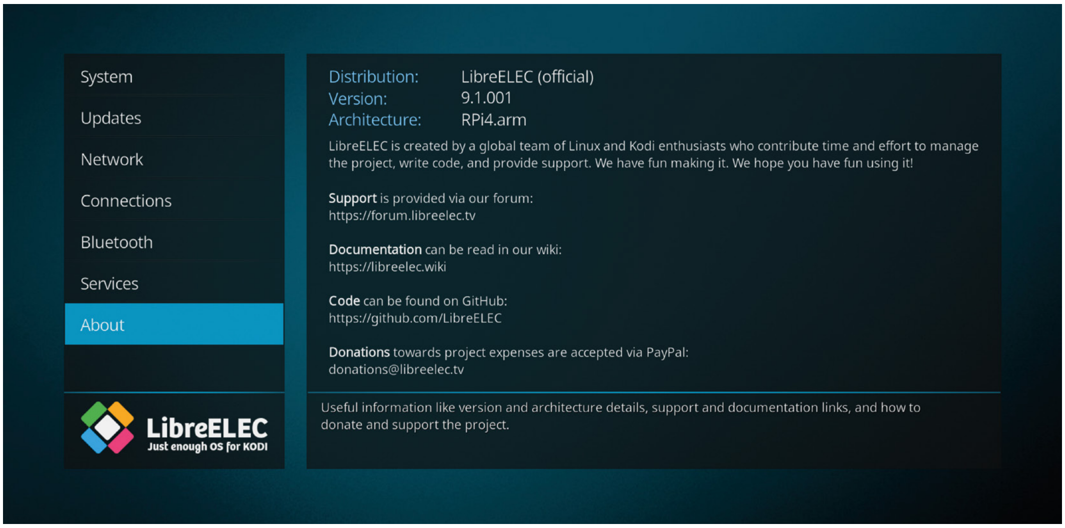
SPECS

KODI VERSION:
18.3 'Leia'

COMPATIBILITY:
All Raspberry Pi versions

INSTALLATION:
Graphical wizard or burnable image

IMAGE SIZE:
129MB



Distribution: LibreELEC (official)
Version: 9.1.001
Architecture: RPi4.arm

LibreELEC is created by a global team of Linux and Kodi enthusiasts who contribute time and effort to manage the project, write code, and provide support. We have fun making it. We hope you have fun using it!

Support is provided via our forum:
https://forum.libreelec.tv

Documentation can be read in our wiki:
https://libreelec.wiki

Code can be found on GitHub:
https://github.com/LibreELEC

Donations towards project expenses are accepted via PayPal:
donations@libreelec.tv

Useful information like version and architecture details, support and documentation links, and how to donate and support the project.

▶ 4K works fine for now, but will get better

You could easily argue that Raspberry Pi was already the perfect media PC – why get excited for a Raspberry Pi 4-compatible version of LibreELEC/Kodi, then? One of the many improvements to Raspberry Pi 4 was the inclusion of hardware H.265 decoding – also known as HEVC. This means it can support 10-bit colour, as well as the 4K output of the new Raspberry Pi 4.

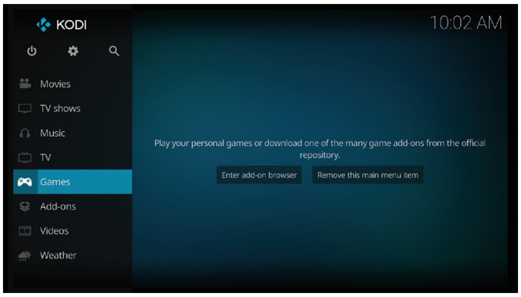
And we can say right off the bat that with H.265 encoded media, it's noticeable immediately. Colour tearing and glitches are completely gone, with media running at full speed without it seeming to struggle. 4K works fine as well – not great, just fine – at 30Hz.

Size matters?

That's not the only reason we decided to do a quick re-review of LibreELEC, though – there's also the question of what version of Raspberry Pi 4 do you need. Could you save $20 and get the 1GB version over the 4GB, or is there a difference?

Yes, there is a difference, but no it's not worth the $20 (for this purpose) in our opinion. After some extensive testing with the three versions, playback was the exact same, even with the most demanding bit rates. The UI, however, was noticeably just a little bit more responsive on the higher gigabyte models. Just the slightest bit, though – nothing to splurge on. At the very least, the 1GB model is a bit better than previous Raspberry Pi computers, in our experience.

So, to recap: you don't even need to buy the best version of Raspberry Pi 4 to get the major benefits of the hardware redesign. That's a huge win.



Play your personal games or download one of the many game add-ons from the official repository.

▲ The video game part has a way to go

Verdict

LibreELEC 9.x still needs some work upstream. However, the beta is well worth jumping on board if you have a Raspberry Pi 4 and the need to use the extra power.

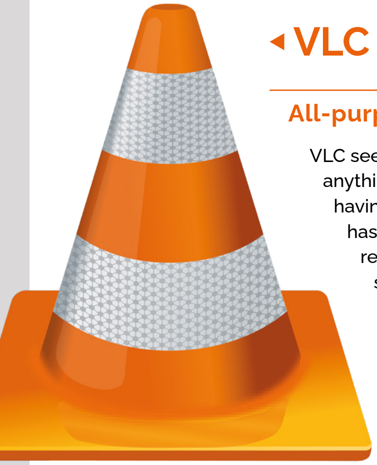
9/10

10 Best:

Raspberry Pi media players

Here's how to play music and video on your Raspberry Pi

Raspberry Pi has been used to play video and music since the very first board landed in the hands of an enthusiast. Its use as a media platform is legendary, and for good reason. Want to take advantage of Raspberry Pi's media powers? Here are ten excellent ways…



◂ VLC Media Player

All-purpose player

VLC seems to be able to play just about anything you throw at it, while still having a very small install size. It also has great streaming abilities, and can receive web streams and network streams as well. It also has a GUI.

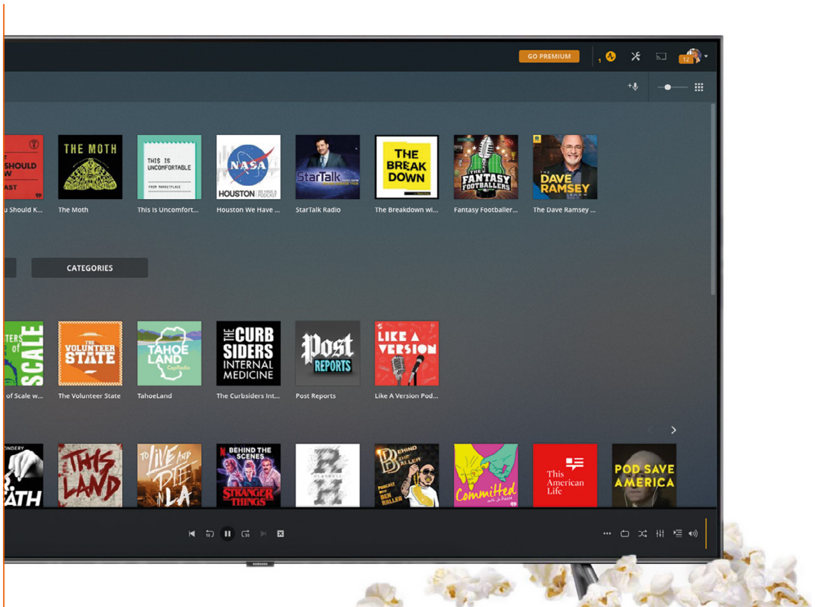
magpi.cc/qsXYHH

▸ MPlayer

The legend

MPlayer is still adored by many long-term Linux and open-source software users, and for good reason: it's great. You'll probably have to compile it from source to get a version you prefer, though.

mplayerhq.hu



▴ Plex

Ultimate media library

Need a tiny media server that will stream to all your devices that support Plex? Using Raspberry Pi you can create the ultimate streaming media server, with the tiniest footprint in both physical space and on your electricity bill.

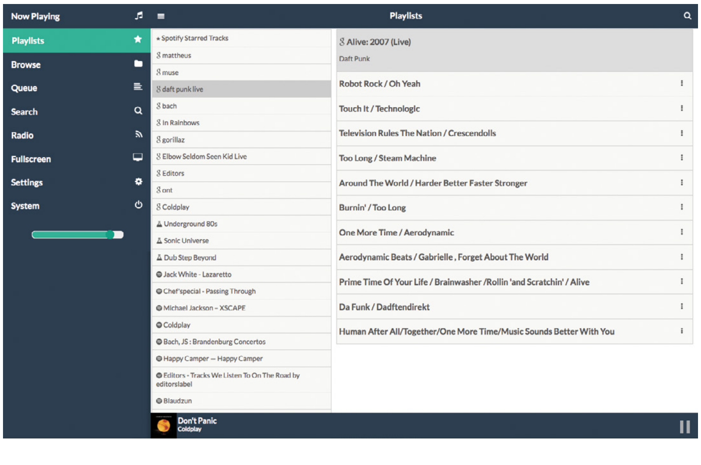
plex.tv

▾ Pi MusicBox

Music streaming software

Another excellent streaming solution for your home. As well as being able to connect to your local music collection, it connects well to online services such as Spotify and SoundCloud.

pimusicbox.com



Now Playing
Playlists
Browse
Queue
Search
Radio
Fullscreen
Settings
System

Playlists

★ Spotify Starred Tracks
mattheus
muse
daft punk live
bach
In Rainbows
gorillaz
Elbow Seldom Seen Kid Live
Editors
ont
Coldplay
Underground 80s
Sonic Universe
Dub Step Beyond
Jack White - Lazaretto
Chef'special - Passing Through
Michael Jackson – XSCAPE
Coldplay
Bach, JS : Brandenburg Concertos
Happy Camper – Happy Camper
Editors - Tracks We Listen To On The Road by editorslabel
Blaudzun

Alive: 2007 (Live)
Daft Punk

Robot Rock / Oh Yeah
Touch It / Technologic
Television Rules The Nation / Crescendolls
Too Long / Steam Machine
Around The World / Harder Better Faster Stronger
Burnin' / Too Long
One More Time / Aerodynamic
Aerodynamic Beats / Gabrielle , Forget About The World
Prime Time Of Your Life / Brainwasher /Rollin 'and Scratchin' / Alive
Da Funk / Dadftendirekt
Human After All/Together/One More Time/Music Sounds Better With You

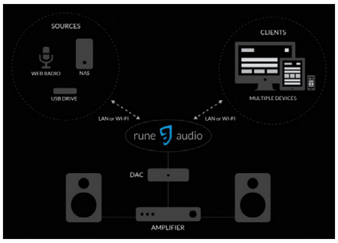
Don't Panic
Coldplay

OSMC

Raspberry Pi HTPC

A Kodi alternative for people who like slightly more modern user interfaces. It has a special installer that can get a Raspberry Pi microSD card set up in minutes, and it also supports streaming to other devices.

osmc.tv



RuneAudio

In-home music system

RuneAudio is a complete home hi-fi system, connecting to local network storage and internet streams. It can be controlled by multiple clients if run headless, and plays music perfectly through a DAC. It's also open-source.

runeaudio.com

Chromium

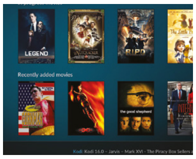
Web browser streaming

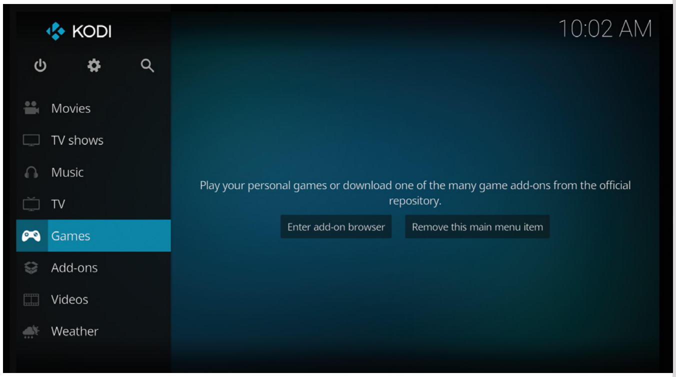
Using the default Raspberry Pi OS browser, you can access a world of content online from various video sources like YouTube, as well as stuff on your home network with the right add-ons.

chromium.org

KODI

Kodi is software used in many home-theatre PCs and other kinds of media PCs. You can install it on Raspbian on Raspberry Pi; however, we find it works a lot better if you're using it on a dedicated media device, especially in conjunction with operating systems that make use of it.





Play your personal games or download one of the many game add-ons from the official repository.

LibreELEC

The purest Kodi

Our favourite Kodi-based OS gives you the pure, familiar, highly customisable Kodi experience. It can play just about anything you can throw at it, including a lot of video games now.

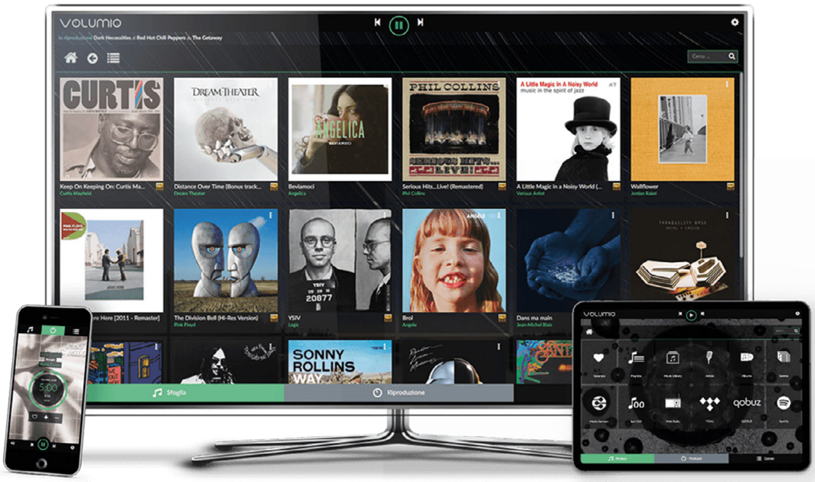
libreelec.tv

Volumio

All-in-one audio

Volumio wants you to have everything connected to one device, and has created a system to allow you to do that. Not only can you store music on it, you can stream music from other places to it.

volumio.org



OMXPlayer

Command-line player

This is great for having media play when you turn a Raspberry Pi on, or to control it from an SSH-connected remote computer. It makes use of minimal resources, while also being hardware-accelerated for great picture quality.

magpi.cc/ABdFRJ

10 Amazing: Media player projects

The best ways to groove, dance, chill, and more using a Raspberry Pi

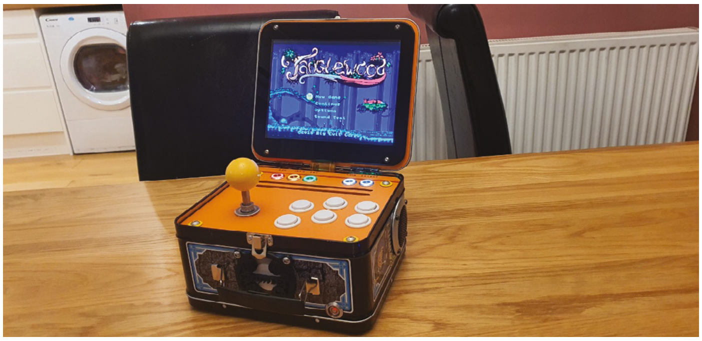
You can use Raspberry Pi to play all kinds of media thanks to its small but mighty system-on-a-chip. We've seen people use Raspberry Pi to create incredible and inventive projects involving media – here are some of the very best.

▼ Big and Smart Raspberry Pi Picture Frame

Digital photo supremacy

With a spare monitor, a nice frame, and a Raspberry Pi loaded with photos or connected to a cloud service, you can easily have a nice, ever-changing series of pictures displayed on your wall.

magpi.cc/bigframe



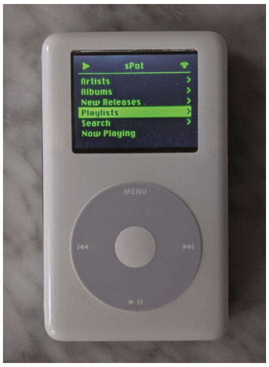
▲ Lunchbox Arcade Game

Lunch-time fun

There are many great and cool video game projects, but this portable arcade machine disguised as a lunchbox is still one of our favourites. Playing it at home or at work on your break seems like the perfect fit for it.

magpi.cc/lunchbox

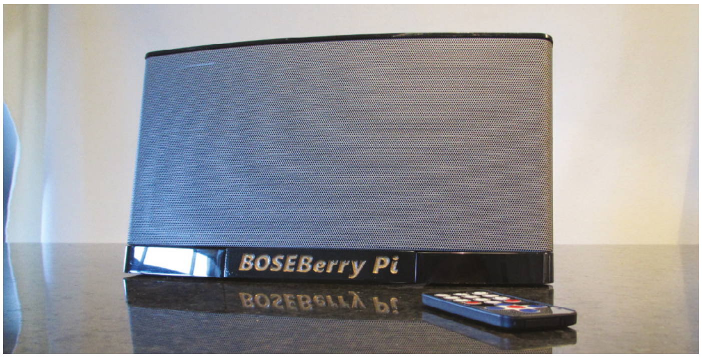
▶ iPod Classic Spotify music player



Upcycled classic

The clicky wheel iPod is a very important device in the history of portable media players. So of course, Guy here ripped one apart and upgraded it with a Raspberry Pi Zero W and Spotify.

magpi.cc/ipodspotify



▲ BOSEBerry Pi

A 2010 throwback

The iPod docks of yore are remembered fondly by many and are still available in a lot of hotel rooms. The form factor (and used price) really appealed to maker David Hunt, who turn one into an internet radio box.

magpi.cc/boseberrypi

▲ Atomic TV

Custom-built nostalgia

Atompunk is a style that takes inspiration from American ideas of the future from the 1940s and 1950s. This retro-future aesthetic has been perfectly captured in Atomic TV, with bezels and curves and switches to match. It's made from scratch as well and wouldn't look out of place in the Carousel of Progress.

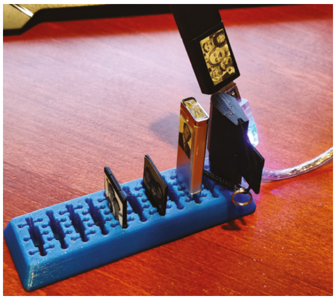
magpi.cc/atomictv

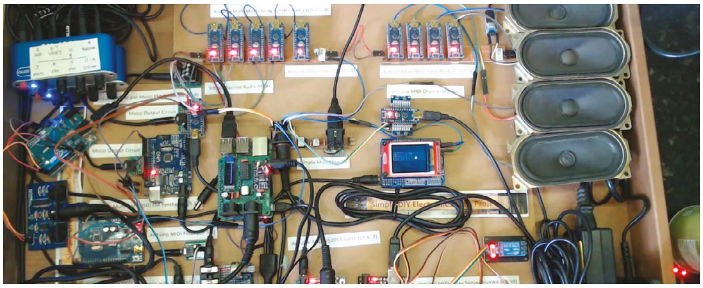
▸ PiVidBox

Simple solution

This basic media player just requires you to plug in an SD card or USB stick with various media on it. It uses old storage that would otherwise never get used, and each has a curated selection of content, labelled to make sure you know what's on it.

magpi.cc/pividbox





▲ Lo-Fi Orchestra

Impractical yet cool

Using a whole host of microcontrollers and microcomputers – including Raspberry Pi – Kevin, aka DIY Electro Music, has managed to recreate a lo-fi version of Gustav Holst's *The Planets* suite. He's making his way through the suite, but *Jupiter* is one of our favourites.

magpi.cc/lofiorchestra

▲ Vintage television

Digital to analogue

Portable TVs were the coolest thing in the eighties and nineties. However, for obvious smartphone-related reasons, they have fallen out of favour. This 1975 portable TV has been upcycled by the great Martin Mander to use a Raspberry Pi and a TV HAT for portable digital telly. Nostalgic.

magpi.cc/vintagetv

▲ Vinyl emulator

RFID records

Using Sonos and Spotify and some very lovely, RFID-enabled cards to represent records, you too can experience the joy of physical media. There are even cassette tape cases for specific functions like play and pause or for specific playlists. Very nice.

magpi.cc/vinylemulator

▸ Retro gaming NES frame

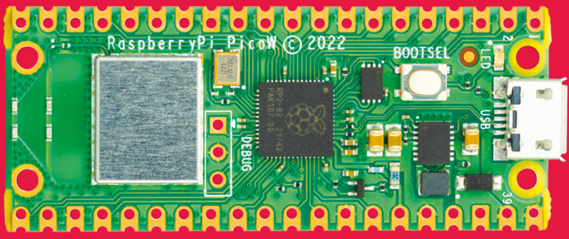
Video game art

Pixel art and old game cover art have their place in pop culture, as does modding the NES. Combine them and you have this framed NES cartridge cycling through art.

magpi.cc/nesframe

SUBSCRIBE TODAY FOR JUST £10

Get 3 issues + FREE Pico W



Subscriber Benefits

- FREE Delivery
  Get it fast and for FREE
- Exclusive Offers
  Great gifts, offers, and discounts
- Great Savings
  Save up to 35% compared to stores

Subscribe for £10

- Free Pico W
- 3 issues of The MagPi
- Free delivery to your door
- £10 (UK only)

Subscribe for 12 Months

- Free Pico W
- 12 issues of The MagPi
- Free delivery to your door

£55 (UK)	£90 (USA)
£80 (EU)	£90 (Rest of World)

☎ Subscribe by phone: **01293 312193**

Subscribe online: **magpi.cc/subscribe**

✉ Email: **magpi@subscriptionhelpline.co.uk**

Subscribe for £10 is a UK-only offer. The subscription will renew at £30 every six months unless cancelled. A free Pico W is included with a 12-month subscription in USA, Europe and Rest of World. (No Raspberry Pi Pico W with £5 Rolling Monthly Subscription).

SUBSCRIBE TODAY AND GET A

FREE Raspberry Pi Pico W

Subscribe in print today and get a **FREE development board**

WORTH $6

- A brand new RP2040-based Raspberry Pi Pico W development board
- Learn to code with electronics and build your own projects
- Make your own home automation projects, handheld consoles, tiny robots, and much, much more

This is a limited offer. Not included with renewals. Offer subject to change or withdrawal at any time.

YOUR OFFICIAL RASPBERRY PI MAGAZINE
The MagPi
Issue 118 | June 2022 | magpi.cc
The official Raspberry Pi magazine
RASPBERRY PI PHOTOGRAPHY
SETUP • EDITING • PROJECTS
TIME-LAPSE & BATCH CONVERT TUTORIALS
BUILD YOUR OWN VECTREX MINI
CRAFT WEARABLE KIT
10 FACE TRACKING PROJECTS
WIN! CUTIEPI COMPUTE MODULE 4 TABLET

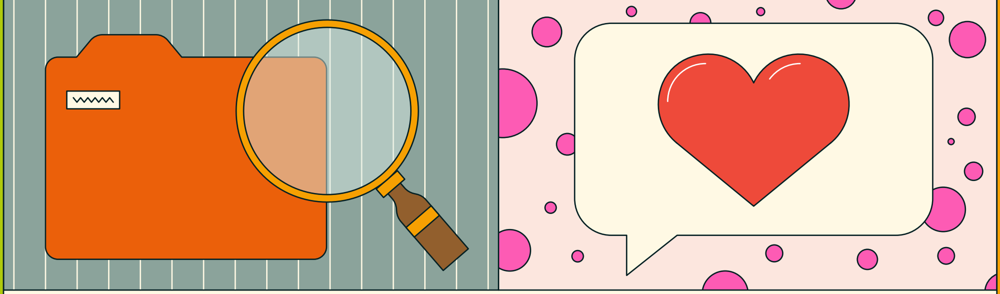
Buy now: magpi.cc/subscribe

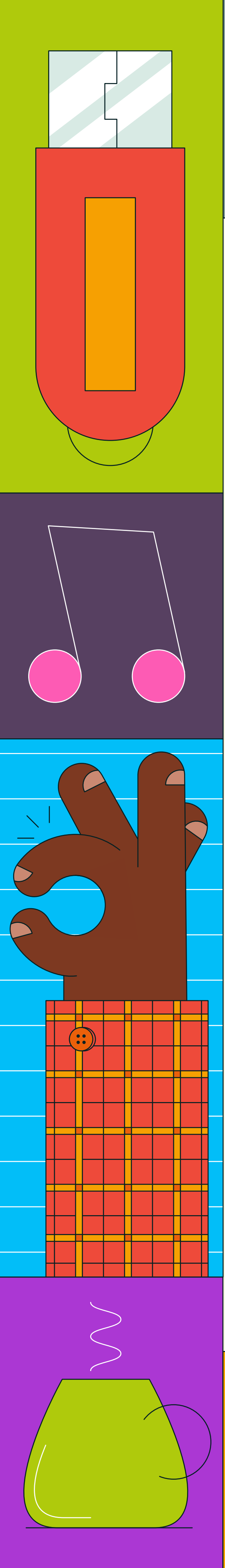
SUBSCRIBE on app stores

From **£2.29**

Available on the App Store

GET IT ON Google Play



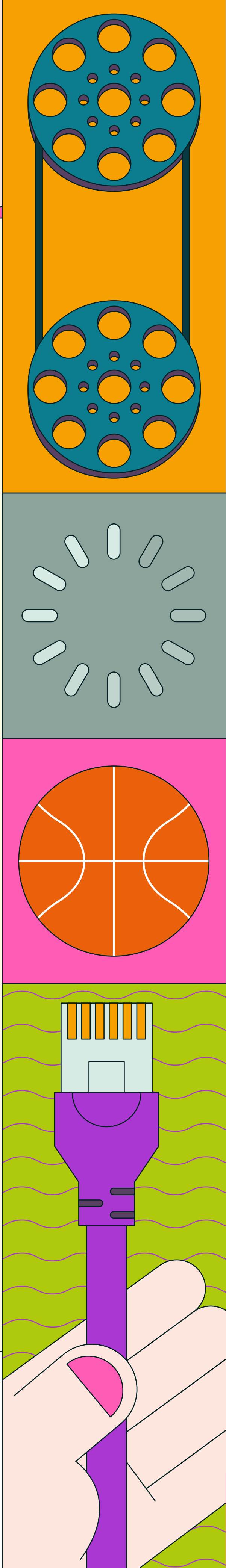


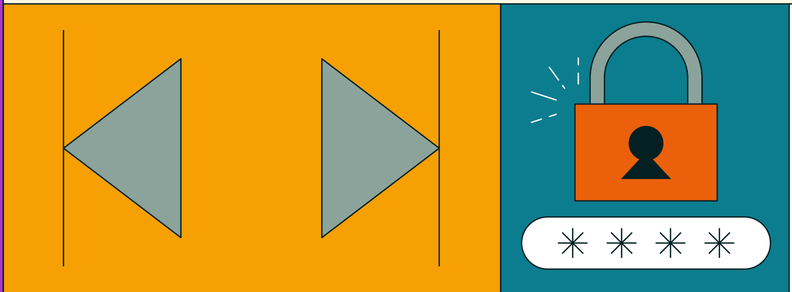
THANK YOU FOR READING!
Build a Raspberry Pi Media Player!

Raspberry Pi is a small, clever, British-built computer that's packed with potential. Made using the same technology you find in a smartphone, Want more articles and guides about all the things you can make with Raspberry Pi?

Read The MagPi, the Official Raspberry Pi magazine

- Sign up to our newsletter and get a free PDF delivered to your inbox: **magpi.cc**
- Get a free Raspberry Pi computer with a twelve-month print subscription: **magpi.cc/subscribe**
- Buy the magazine in print, and get books from us and the rest of Raspberry Pi Press at **store.rpipress.cc**





Raspberry Pi PRESS
raspberrypi.com