

# Five projects for Raspberry Pi at home

Ben Nuttall

The Raspberry Pi computer can be used in all kinds of settings and for a variety of purposes. It obviously has a place in education for helping students with learning programming and maker skills in the classroom and the hackspace, and it has plenty of industrial applications in the workplace and in factories. I'm going to introduce five projects you might want to build in your own home.

## Media center

One of the most common uses for Raspberry Pi in people's homes is behind the TV running media center software serving multimedia files. It's easy to set this up, and the Raspberry Pi provides plenty of GPU (Graphics Processing Unit) power to render HD TV shows and movies to your big screen TV. Kodi (formerly XBMC) on a Raspberry Pi is a great way to playback any media you have on a hard drive or network-attached storage. You can also install a plugin to play YouTube videos.

There are a few different options available, most prominently OSMC (Open Source Media Center) and LibreELEC, both based on Kodi. They both perform well at playing media content, but OSMC has a more visually appealing user interface, while LibreElec is much more lightweight. All you have to do is choose a distribution, download the image and install on an SD card (or just use NOOBS), boot it up, and you're ready to go.

Before proceeding you'll need to decide which Raspberry Pi model to use. These distributions will work on any Pi (1, 2, 3, or Zero), and video playback will essentially be matched on each of these. Apart from the Pi 3 (and Zero W) having built-in Wi-Fi, the only noticeable difference is the reaction speed of the user interface, which will be much faster on a Pi 3. A Pi 2 will not be much slower, so that's fine if you don't need Wi-Fi, but the Pi 3 will noticeably outperform the Pi 1 and Zero when it comes to flicking through the menus.

## SSH gateway

If you want to be able to access computers and devices on your home network from outside over the internet, you have to open up ports on those devices to allow outside traffic. Opening ports to the internet is a security risk, meaning you're always at risk of attack, misuse, or any kind of unauthorized access. However, if you install a Raspberry Pi on your network and set up port forwarding to allow only SSH access to that Pi, you can use that as a secure gateway to hop onto other Pis and PCs on the network.

Most routers allow you to configure port-forwarding rules. You'll need to give your Pi a fixed internal IP address and set up port 22 on your router to map to port 22 on your Raspberry Pi. If your ISP provides you with a static IP address, you'll be able to SSH into it with this as the host address (for example, **ssh pi@123.45.56.78**). If you have a domain name, you can configure a subdomain to point to this IP address, so you don't have to remember it (for example, **ssh pi@home.mydomain.com**).

However, if you're going to expose a Raspberry Pi to the internet, you should be very careful not to put your network at risk. There are a few simple procedures you can follow to make it sufficiently secure:

1. Most people suggest you change your login password (which makes sense, seeing as the default password "raspberrypi" is well known), but this does not protect against brute-force attacks. You could change your password and add a two-factor authentication (so you need your password *and* a time-dependent passcode generated by your phone), which is more secure. However, I believe the best way to secure your Raspberry Pi from intruders is to disable "password authentication" in your SSH

configuration, so you allow only SSH key access. This means that anyone trying to SSH in by guessing your password will never succeed. Only with your private SSH key can anyone gain access. Similarly, most people suggest changing the SSH port from the default 22 to something unexpected, but a simple Nmap of your IP address will reveal your true SSH port.

2. Ideally, you would not run much in the way of other software on this Pi, so you don't end up accidentally exposing anything else. If you want to run other software, you might be better running it on another Pi on the network that is not exposed to the internet. Ensure that you keep your packages up to date by upgrading regularly, particularly the **openssh-server** package, so that any security vulnerabilities are patched.

3. Install sshblack or fail2ban to blacklist any users who seem to be acting maliciously, such as attempting to brute force your SSH password.

Once you've secured your Raspberry Pi and put it online, you'll be able to log in to your network from anywhere in the world. Once you're on your Raspberry Pi, you can SSH into other devices on the network using their local IP address (for example, 192.168.1.31). If you have passwords on these devices, just use the password. If they're also SSH-key-only, you'll need to ensure your key is forwarded over SSH by using the **-A** flag: **ssh -A pi@123.45.67.89**.

## CCTV / pet camera

Another great home project is to set up a camera module to take photos or stream video, capture and save files, or streamed internally or to the internet. There are many reasons you might want to do this, but two common use cases are for a homemade security camera or to monitor a pet.

The Raspberry Pi camera module is a brilliant accessory. It provides full HD photo and video, lots of advanced configuration, and is easy to program. The infrared camera is ideal for this kind of use, and with an infrared LED (which the Pi can control) you can see in the dark!

If you want to take still images on a regular basis to keep an eye on things, you can just write a short Python script or use the command line tool raspistill, and schedule it to recur in Cron. You might want to have it save them to Dropbox or another web service, upload them to a web server, or you can even create a web app to display them.

If you want to stream video, internally or externally, that's really easy, too. A simple MJPEG (Motion JPEG) example is provided in the picamera documentation (under "web streaming"). Just download or copy that code into a file, run it and visit the Pi's IP address at port 8000, and you'll see your camera's output live.

A more advanced streaming project, pistreaming, is available, which uses JSMpeg (a JavaScript video player) with the web server and a websocket for the camera stream running separately. This method is more performant and is just as easy to get running as the previous example, but there is more code involved and if set up to stream on the internet, requires you to open two ports.

Once you have web streaming set up, you can position the camera where you want it. I have one set up to keep an eye on my pet tortoise.

If you want to be able to control where the camera actually points, you can do so using servos. A neat solution is to use Pimoroni's Pan-Tilt HAT, which allows you to move the camera easily in two dimensions. To integrate this with pistreaming, see the project's pantilthat branch.

If you want to position your Pi outside, you'll need a waterproof enclosure and some way of getting power to the Pi. PoE (Power-over-Ethernet) cables can be a good way of achieving this.

# Home automation and IoT

It's 2017 and there are internet-connected devices everywhere, especially in the home. Our lightbulbs have Wi-Fi, our toasters are smarter than they used to be, and our tea kettles are at risk of attack from Russia. As long as you keep your devices secure, or don't connect them to the internet if they don't need to be, then you can make great use of IoT devices to automate tasks around the home.

There are plenty of services you can buy or subscribe to, like Nest Thermostat or Philips Hue lightbulbs, which allow you to control your heating or your lighting from your phone, respectively—whether you're inside or away from home. You can use a Raspberry Pi to boost the power of these kinds of devices by automating interactions with them according to a set of rules involving timing or even sensors. One thing you can't do with Philips Hue is have the lights come on when you enter the room, but with a Raspberry Pi and a motion sensor, you can use a Python API to turn on the lights. Similarly, you can configure your Nest to turn on the heating when you're at home, but what if you only want it to turn on if there's at least two people home? Write some Python code to check which phones are on the network and if there are at least two, tell the Nest to turn on the heat.

You can do a great deal more without integrating with existing IoT devices and with only using simple components. A homemade burglar alarm, an automated chicken coop door opener, a night light, a music box, a timed heat lamp, an automated backup server, a print server, or whatever you can imagine.

## Tor proxy and blocking ads

Adafruit's Onion Pi is a Tor proxy that makes your web traffic anonymous, allowing you to use the internet free of snoopers and any kind of surveillance. Follow Adafruit's tutorial on setting up Onion Pi and you're on your way to a peaceful anonymous browsing experience.

You can install a Raspberry Pi on your network that intercepts all web traffic and filters out any advertising. Simply download the Pi-hole software onto the Pi, and all devices on your network will be ad-free (it even blocks in-app ads on your mobile devices).

*This article was originally published on [Opensource.com](https://opensource.com) and is licensed under a [Creative Commons Share-Alike by Attribution](https://creativecommons.org/licenses/by/4.0/) license.*