Cooling a Raspberry Pi Device

Various mechanisms to help keep a Pi cool.

Colophon

Legal disclaimer notice

Document version history

Release	Date	Description
1.0	1 Jan 2022	Initial release
1.1	1 Feb 2025	Added information on underclocking to reduce heat.

Scope of document

This document applies to the following Raspberry Pi products:

Pi 0			Pi 1		Pi 2		Pi 3	Pi 4	Pi 5	Pi 400	CM1	CM3	CM4	CM5	Pico
0	W	H	A	B	A	B	B	All	All	All	All	All	All	All	All
*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	

Introduction

Ever since the release of the first Raspberry Pi, people have talked about cooling. Whether it's to keep things cool in extreme environments, or trying to stop overclocks from overheating, much has been said and written on the various ways of keeping the temperature down. As each new model has been released, the need for cooling in these circumstances, while not compulsory, has increased.

This whitepaper goes through the reasons why your Raspberry Pi may get hot and why you might want to cool it back down, and gives various options for achieving that cooling process.

This whitepaper assumes that the Raspberry Pi is running the Raspberry Pi operating system (OS), and is fully up to date with the latest firmware and kernels.

Thermal considerations

Why does my Raspberry Pi get hot?

All silicon-based devices warm up when they are in use. For example, on Raspberry Pi 4B there are three major silicon devices on the board: the SoC (system on a chip, the main processor), the wireless/Bluetooth device, and the memory device. In addition, there are power control chips and circuitry. When working, each of these silicon devices has millions upon millions of tiny electronic transistor gates switching on and off rapidly, each producing heat as they switch. While each gate produces only a tiny amount of heat, there are millions of them switching very, very fast, and it is their combined heat we can feel if we touch one of these silicon devices while it is running.

⊖ **WARNING**

> The chips, especially the SoC, can get pretty hot, so be careful if you do actually touch any of them!

The number of gates switching in a chip also depends on the workload the chip is being asked to service. For example, if your Raspberry Pi is just sitting at a desktop prompt, with no video playing or 3D graphics being produced, then the workload is fairly low; fewer gates will be switching, and the chip may be running more slowly. However, as soon as you start up a compute-intensive application, more gates come into play and start to switch faster, producing more heat. This rise in temperature can actually happen quite quickly.

Over the years, Raspberry Pi devices have become steadily more powerful. The first Raspberry Pi used the BCM2835 processor, which only has a single Armv6 core. This means it has many fewer silicon gates than more recent devices, and therefore generates much less heat. In fact, except in very extreme environments, it is extremely unlikely you would ever need to add extra cooling to the devices based on the BCM2835, which includes Raspberry Pi Zero (W/H) and Raspberry Pi Compute Module 1.

As our devices have become more powerful,they have moved to quad-core processors running at higher frequencies, producing more heat. However, internal thermal management techniques have also improved at the same time, and in many cases, despite the extra power, there is still no need for extra cooling.

❗ **IMPORTANT**

> You cannot damage a Raspberry Pi device by letting it run hot, so it is always safe *not* to apply any sort of cooling mitigations. In fact, Raspberry Pi devices have been tested to well over 120°C with no problems. Their operational lifetime will decrease at these very high temperatures (which should never be reached due to the thermal management involved), but even then, those lifetimes can be measured in decades.

Internal thermal management

All devices in the Raspberry Pi range include some sort of thermal management internally on the SoC. The SoCs were designed with low-power applications in mind and incorporate various techniques to reduce power requirements; reducing power also decreases the overall temperatures the devices can reach.

Clock gating

It seems fairly obvious, but one way of reducing power consumption (and therefore heat) is to simply turn things off when they are not in use. We already do that with TVs overnight, and it would be daft to leave your car running if you were not using it, for instance.

However, though it sounds simple, doing the same thing inside a silicon chip is a little more complicated than turning

your TV off when you go to bed! The VideoCore graphics processing units inside the SoCs on Raspberry Pi devices contain special circuitry that can turn off chunks of silicon that are not in use. For example, if you are not using the H264 encoder, it is powered down. In fact, it is even cleverer than that — it will turn things off and on again in a split second if that saves power. If you are outputting video information 30 times a second, that's 33ms or so per frame. If the work needed from a chunk of silicon can produce the frame in 10ms, you can turn off that piece of silicon for 23ms each frame! Of course, if you double the frame rate to 60Hz, you not only double the amount of work needed per second, but you also have less time (only 6ms) in which that bit of silicon can be turned off, which explains why things get hotter when you increase the frame rate or resolution!

So, the VideoCore processor does some of this *clock gating*, which can save a lot of power.

Frequency and voltage management

As mentioned above, increasing the frequency that the SoC is running at increases its performance, but it also increases the heat produced. Something else that comes into this equation is the voltage that the core silicon runs at: the higher the voltage, the higher the frequency that the silicon can handle. A corollary of this is that, if you run at low frequencies, you can drop the voltage driving the silicon, and dropping the voltage means less power, which in turn means less heat.

So, on some Raspberry Pi models we use a scheme called *dynamic voltage and frequency scaling* (DVFS). This is the same technology used in laptops and the like to reduce power consumption and therefore increase battery life. This technology varies the voltage and the frequency supplied to the SoC according to the computing demands being made. So, if the device is mostly idle, the frequency and voltage will be dropped down. If computing demand rises, the voltage and frequency will be increased to provide the extra performance needed. This is a great scheme that for most people means the Raspberry Pi will never really get too hot, since in most cases the device only needs to run occasionally at full speed.

The only fly in the ointment is the very compute-intensive workloads that last for a long time, such as the compilation of large projects or video processing. Under these loads, the SoC never gets a chance to drop the voltage or frequency and let itself cool down, which brings us to the next topic.

Thermal throttling

Raspberry Pi SoCs all have internal temperature sensors, which are constantly being monitored by firmware that runs, in the background, all the time. This code tests the temperature, and if it reaches a predefined limit, which for Raspberry Pi devices is 85°C, the voltage and frequencies are forced down, even when the workload is high. This gives the processor a chance to cool down, but does mean that performance is reduced, so compute-intensive tasks will take longer if this thermal throttling point is reached.

Monitoring temperatures

It is possible to monitor temperatures from the command line using `vcgencmd`:

```
pi@raspberrypi:~ $ vcgencmd measure_temp
temp=49.6C
```

In addition, if you are using the Raspberry Pi OS desktop, you can add a CPU monitor to the menu bar (known as a panel). Right click on the menu, click `Add/Remove Panel items...` and the Panel Preferences dialog box will appear. Click on `Add` and select the `CPU Temperature Monitor` plugin. Once added, a graph will appear on the menu bar with the current temperature overlaid. You can right click on this graph to select various customisation options.

So when might I need to add extra cooling?

The thermal management techniques already in use will mean that for most use cases, no extra actions are required. There are, however, some circumstances when some sort of extra cooling may be needed:

- Very high ambient temperatures
- High, persistent workloads
- Airtight enclosures
- More extreme overclocking

If you find that your Raspberry Pi is throttling during your usual workload, then you may need to add extra cooling. Although no harm can come to the device if it throttles, you will be losing some performance that can possibly be regained, often with very simple changes.

Dealing with excess heat

The first question to ask when deciding on a cooling solution is whether any extra cooling is actually needed. The vast majority of Raspberry Pi devices have no extra cooling added, and rely entirely on the internal DVFS and thermal throttling to keep temperatures within the working range. But, if you are running high and persistent workloads, or are in a high ambient temperature, then there may be some benefit to adding extra cooling.

There are some things that can be done to improve cooling before adding extra hardware like heatsinks or fans.

Bare boards

If you run a Raspberry Pi in the open air, outside of a case, then simple convection will keep it pretty cool. However, if it is laid flat on a desk, then hot air has difficulty circulating under the board. Increasing the gap by using stand-offs can help, but a very simple way to improve cooling is to simply mount the board on its edge. This allows hot air to rise up from both sides of the board, meaning there is no trapped air and convection can drop the temperature of the board considerably.

If you are prototyping on a desk using bare boards, a quick and easy way to keep the device cool is a desktop fan! Any extra airflow around the board will greatly increase the cooling.

Adding a heatsink

Heatsinks improve cooling by moving the heat away from the processor and by providing a much larger surface area from which the heat can dissipate. There are many third-party suppliers of heatsinks for Raspberry Pi devices — some better than others — but a very important part of the heatsink is its thermal connection to the processor. Thermal tape can be used, but a good thermal paste is usually better.

Figure 1. A Raspberry Pi in a case with a heatsink

It is important to understand that heatsinks still need to dissipate heat to their environment, which is usually air. If there is little or no airflow over the heatsink, then it will have problems moving heat away, and so, as for bare boards, this airflow is important.

Heatsinks can also help even when there is very limited airflow, as long as the processing load is intermittent. This is because they provide more thermal mass into which to dump heat. If the workload is intense but infrequent — such that thermal throttling is reached during the peak — but then there is a long gap before the next peak, a heatsink acting as thermal mass can absorb that heat, preventing thermal throttling. It then has the time between peaks to cool down. As long as the cool-down time is long enough, the heatsink can continue to absorb the peak heating and prevent throttling. If the time between peaks is short, however, you will need extra airflow to cool the heatsink down enough so that it doesn't gradually increase in temperature until it can no longer absorb the peaks.

Using cases

Once a Raspberry Pi is placed in a case, it is clear that airflow will be reduced over the device. Holes in the case can help, but you will find that a Raspberry Pi in a case will, unless other mitigations are in place, run hotter. In most situations, though, it should still be able to maintain a sensible operating temperature through the use of DVFS and throttling.

Although adding a heatsink to a Raspberry Pi in a case can help for burst loads (as described above), adding one to a case without decent airflow or some way of moving the heat from the heatsink to outside the case means the device will eventually still heat up.

Some cases are, in effect, large heatsinks. They thermally connect the processor to the outside of the case, which is usually made of metal. This can be very effective at keeping temperatures down.

Figure 2. Argon Neo
heatsink case

Figure 3. FLIRC
heatsink case

> 💡 **TIP**
>
> The AstroPi case is used on the International Space Station, where there is very little airflow. It is machined out of a solid block of aluminium and acts as a very large heatsink. Getting the heat dissipation up to the required standards took quite a lot of work!

Figure 4. AstroPi case

Fans

If all the preceding mitigations have failed to reduce the temperature of your Raspberry Pi to your satisfaction, then you may wish to try a fan. Fans ensure high airflow over the device, blowing or dragging heat away, and can often be programmed to turn on and off as required to keep the processor within a required temperature range. Raspberry Pi Ltd sells a fan that fits inside the standard Raspberry Pi case for a very reasonable price, and this also comes with a heatsink for extra cooling capabilities.

Figure 5. Raspberry Pi case fan
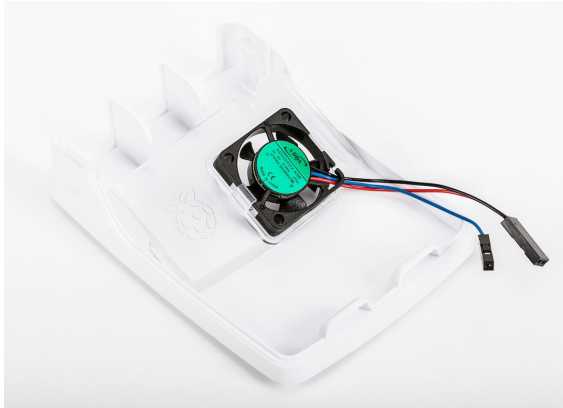
When combined with a heatsink, fans are the most effective way to keep a Raspberry Pi cool; their main disadvantage is that they require power, and so increase the total power budget for the Raspberry Pi. They can also be a little noisy if running quickly.

Power over Ethernet (PoE) HAT

Raspberry Pi Ltd sells a PoE HAT (hardware attached on top), a PCB that attaches to a Raspberry Pi device on the GPIO header. As well as providing the ability to power the Raspberry Pi over the Ethernet cable (with the appropriate router PoE capabilities), this accessory also incorporates a fan to cool both the Raspberry Pi and the PoE HAT itself.

Figure 6. Raspberry Pi 4B with PoE HAT

Reducing clock speed

One way of preventing excess heat is to underclock the device. This results in less heat being produced by the SoC, but does reduce performance. The actual reduction in temperature may be minimal depending on the circumstances, so experimentation is advised to determine whether this can be a useful change.

There are three main options when setting clock speeds:

- arm_freq
  Sets the maximum frequency of the Arm cores, in MHz, at the firmware level

- arm_freq_min
  Sets the idle frequency of the Arm cores, in MHz, at the firmware level

- powersave governor
  Sets the scaling frequency governor to powersave mode at the Linux OS level; see https://www.kernel.org/doc/Documentation/cpu-freq/governors.txt

Firmware

The value to which you underclock will depend on the device being used. For example, Raspberry Pi 5 has a default maximum Arm frequency of 2400MHz, whilst Raspberry Pi 4's is 1800MHz (or 1500MHz, depending on board revision and OS version), so setting a specific underclock will depend on the device in use. You can use `config.txt` conditional filters (https://www.raspberrypi.com/documentation/computers/config_txt.html#model-filters) to tell the system which to apply, depending on the model.

Here is an example `config.txt` file with entries to underclock a Raspberry Pi 5 and Raspberry Pi 4.

```
[pi5]
# Set the Arm A76 core frequency in MHz
arm_freq=1500
arm_freq_min=600

[pi4]
# Set the Arm A72 core frequency in MHz
arm_freq=1000
arm_freq_min=600
```

Raspberry Pi's online documentation has a section on over- and under-clocking that defines all the default values and limits: https://www.raspberrypi.com/documentation/computers/config_txt.html#overclocking-options.

Linux

To set the Linux frequency scaling governor to powersave mode, use the following on the command line (or in a startup script so that it is applied on each boot).

```
echo powersave | sudo tee /sys/devices/system/cpu/cpu0/cpufreq/scaling_governor
```

Conclusions

In most situations, no extra cooling is needed, but there are a number of ways to ensure that heavy workloads or high ambient temperatures do not stop your Raspberry Pi from performing at maximum speed. They vary greatly in capability and price. You will need to decide whether you want passive or active cooling, and whether you can put up with the extra power or noise of a fan, or with the extra cost of a heatsink-style case. As always, the ultimate decision depends on your particular situation.