

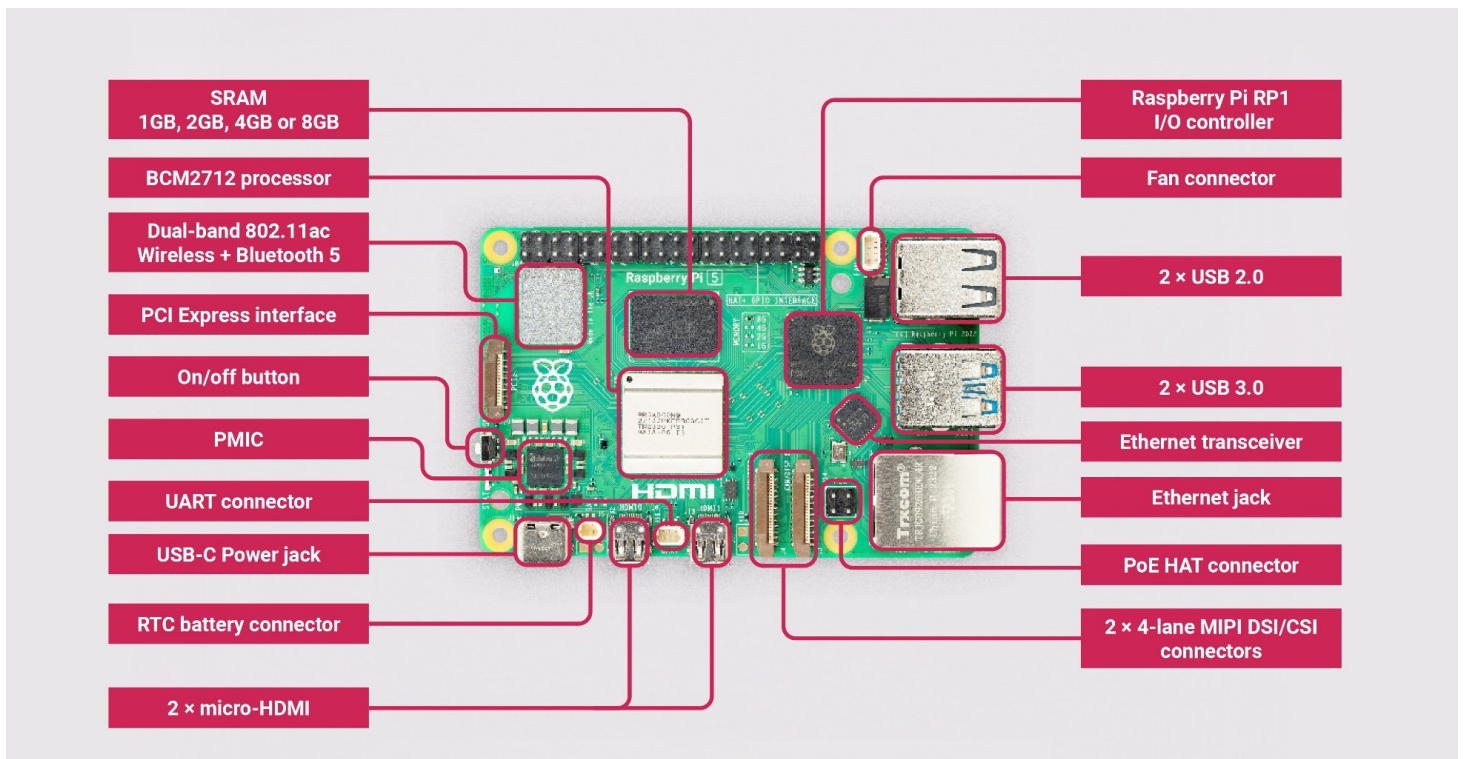
Raspberry Pi 5

Introduction

Edit this on GitHub

IMPORTANT

If you are using a Raspberry Pi for the first time please see our instructions on how to get started.



Raspberry Pi 5

Featuring a 64-bit quad-core Arm Cortex-A76 processor running at 2.4GHz, Raspberry Pi 5 delivers a $\times 2$ to $\times 3$ increase in CPU performance relative to Raspberry Pi 4. Alongside a substantial uplift in graphics performance from an 800MHz VideoCore VII GPU; dual 4Kp60 display output over HDMI; and state-of-the-art camera support from a rearchitected Raspberry Pi Image Signal Processor, it provides a smooth desktop experience for consumers, and opens the door to new applications for industrial customers.

For the first time, this is a full-size Raspberry Pi computer using silicon built in-house at Raspberry Pi. The RP1 provides the bulk of the I/O capabilities for Raspberry Pi 5, and delivers a step change in peripheral performance and functionality. Aggregate USB bandwidth is more than doubled, yielding faster transfer speeds to external UAS drives and other high-speed peripherals; the dedicated two-lane 1Gbps MIPI camera and display interfaces present on earlier models have been replaced by a pair of four-lane 1.5Gbps MIPI transceivers, tripling total bandwidth, and supporting any combination of up to two cameras or displays; peak

SD card performance is doubled, through support for the SDR104 high-speed mode; and for the first time the platform exposes a single-lane PCI Express 2.0 interface, providing support for high-bandwidth peripherals.

Key features include:

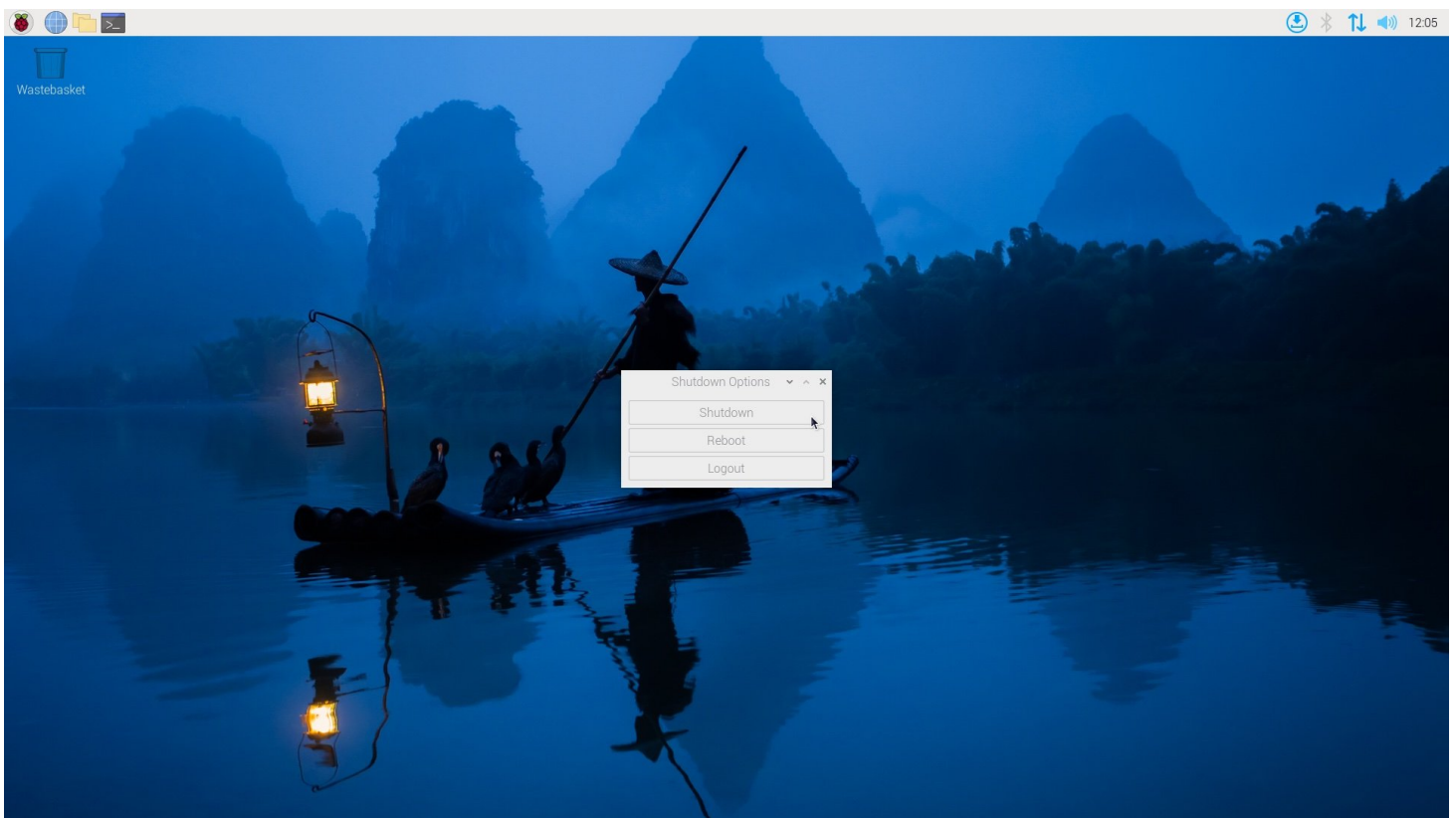
- Quad Arm Cortex-A76 @ 2.4GHz
 - Cryptographic Extension support (AES on hardware)
 - 512KB per-core L2 caches
 - 2MB L3 cache
- 4GB or 8GB LPDDR4X-4267 SDRAM
- Dual 4kp60 HDMI display output with HDR support
- 4kp60 HEVC decoder
- VideoCore VII graphics, with OpenGL-ES 3.1, Vulkan 1.2
- Raspberry Pi Image Sensor Processor (ISP)
- Raspberry Pi connector for PCIe (1 x 2.0 port, additional HAT required)
- 802.11ac dual-band Wi-Fi
- Bluetooth 5.0 (with BLE support)
- Gigabit Ethernet
- 2x USB 3.0 (capable of simultaneous full throughput)
- 2x USB 2.0
- PoE support (additional HAT required)
- Dual 4-lane MIPI CSI/DSI transceivers, supporting
 - 2x display; or
 - 2x camera; or
 - 1x display + 1x camera
- Raspberry Pi 40-pin GPIO header
- Fan connector
- Real Time Clock (RTC)
 - RTC battery connector

- Power button

Turning it off and back on again

When you plug your Raspberry Pi into power for the first time it will automatically turn on and boot into the operating system without having to push the button.

If you are running the Raspberry Pi Desktop you can initiate a clean shutdown by briefly pressing the power button located on the left-hand side of the board, and then releasing it. This will cause a Desktop menu to appear asking whether you want to Shutdown, Reboot, or Logout. You can select whether you want to shutdown or reboot, or alternatively pressing the power button briefly for a second time will initiate a clean shutdown.



Shutting down your Raspberry Pi 5 using the power button.

NOTE

If you are running your Raspberry Pi OS with a Desktop you can press the power button twice in quick succession to shutdown. However if you are running your Raspberry Pi OS

If you are running Raspberry Pi OS Lite in headless mode pressing the power button a single time will initiate a shutdown.

If you want to force a hard shutdown, you should press and hold the Power Button.

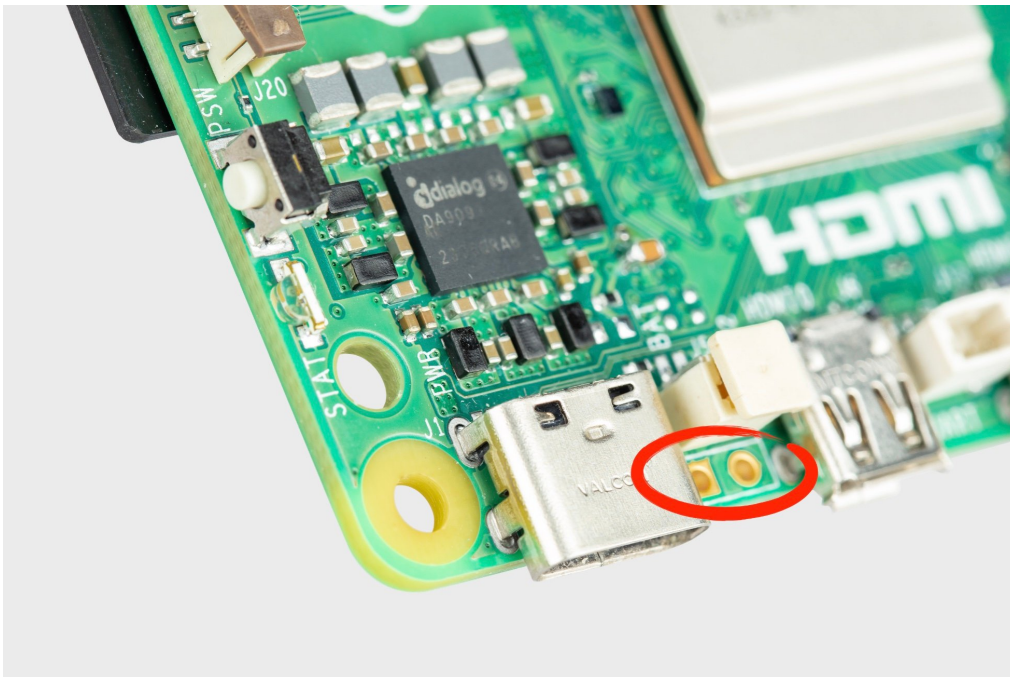
If the Raspberry Pi board is shutdown, but still powered, pressing the power button will restart the board.

Default shutdown wattage is around 1 to 1.4W. However this can be decreased by manually editing the EEPROM configuration, `sudo rpi-eeprom-config -e` and change the settings to:

```
BOOT_UART=1
POWER_OFF_ON_HALT=1
BOOT_ORDER=0xf416
```

This should drop the shutdown power consumption to around 0.01W.

Adding your own power button



The J2 jumper

The J2 jumper, located between the RTC battery connector and the board edge, is a breakout allowing you to add your own power button to Raspberry Pi 5 by adding a Normally Open (NO) momentary switch bridging the two pads. Briefly closing this switch will perform the same actions as the onboard power button.

Updating your firmware

WARNING

Do not update the firmware on your Raspberry Pi unless instructed by a Raspberry Pi engineer.

While firmware updates are rare, if you run into a problem with your Raspberry Pi you may want to check to see if one is available. You can do so as below:

```
sudo apt update
sudo apt upgrade
sudo rpi-eeeprom-update -a
```

If a firmware update is available you will be prompted to install it and your Raspberry Pi will reboot. More information about updating the firmware on your Raspberry Pi is available elsewhere in the documentation.

Powering Raspberry Pi 5

Edit this on GitHub

Raspberry Pi manufactures two different USB-C power supplies.

The first is the Raspberry Pi 15W USB-C Power Supply, which is the recommended supply for Raspberry Pi 4 and Raspberry Pi 400. The second is the Raspberry Pi 27W USB-C Power Supply, which provides up to 5A current at +5.1V, and is the recommended supply for Raspberry Pi 5.

The USB-C standard, used by the Raspberry Pi 4 supply, is limited to 5V at 3A, for a total power output of 15W. The USB-PD standard allows higher voltages and currents to be negotiated via software but requires an appropriate Power Management Integrated Circuit (PMIC) which is only present on Raspberry Pi 5. These supplies require that you use suitable USB-PD compatible cables.

NOTE

It's not possible to use older USB "dumb" cables and connectors to provide more than 15W, even with a USB-PD capable supply.

While USB-PD capable phone chargers advertise greater than 15W of power, virtually all of them achieve this by increasing the voltage instead of providing more current at +5V. If you are using a power supply that cannot provide 5A at +5V on first boot you will be warned by the operating system that the current draw to peripherals will be restricted to 600mA.



On-screen warning of under-voltage event

For users who wish to drive high-power peripherals like hard drives and SSDs, while retaining margin for peak workloads, a USB-PD enabled power supply capable of supplying a 5A at +5V (25W) should be used. If the Raspberry Pi 5 firmware detects such a supply, it increases the USB current limit for peripherals to 1.6A, providing 5W of extra power for downstream USB devices, and 5W of extra onboard power budget.

We recommend the official Raspberry Pi Power Supply, which has been designed to consistently provide +5V despite rapid fluctuations in current draw. Those fluctuations in demand are common and frequent when you're using peripherals with the Raspberry Pi. Other power supplies — usually designed to provide consistent current for charging cellphones — don't cope with power fluctuations all that well. The official supply also has an attached (captive) USB cable, which means that you don't accidentally use a poor-quality or "dumb" cable, which can be an issue with other supplies.

WARNING

If you are using a third-party USB-PD capable multi-port power supply, plugging an additional device into the supply as well as your Raspberry Pi will cause a renegotiation between the supply and the Raspberry Pi. This will happen seamlessly if the Raspberry Pi is powered, but if the Raspberry Pi is plugged in and powered down, this renegotiation may cause the Raspberry Pi to boot.

You can check the status of power output to the USB ports using `vcgencmd`.

```
vcgencmd get_config usb_max_current_enable
```

USB boot and power supplies

By default, USB boot is not enabled with a 3A power supply. However, setting `usb_max_current_enable=1` in the `/boot/firmware/config.txt` file will override this and enable USB boot with a lower wattage power supply.

When booting using a 3A supply without this statement in the `config.txt`, a warning message is displayed on the bootloader HDMI diagnostics screen and the bootloader will skip to the next boot mode.

```
Trying partition: 0
type: 32 lba: 8192 'mkfs.fat' ' bootfs      ' clusters 130554 (4)
rsc 32 fat-sectors 1020 root dir cluster 2 sectors 0 entries 0
FAT32 clusters 130554
[MSD [01:00] 2.00 000000:02] autoboot.txt not found
Trying partition: 0
type: 32 lba: 8192 'mkfs.fat' ' bootfs      ' clusters 130554 (4)
rsc 32 fat-sectors 1020 root dir cluster 2 sectors 0 entries 0
FAT32 clusters 130554
Read config.txt bytes      2109 hnd 0x10a
[MSD [01:00] 2.00 000000:02] pieeprom.upd not found
usb_max_current_enable default 0 max-current 900
Read bcm2712-rpi-5-b.dtb bytes  71862 hnd 0x5101
dt-match: compatible: raspberrypi,5-model-b match: brcm,bcm2712
dt-match: compatible: brcm,bcm2712 match: brcm,bcm2712
***
USB boot requires a high current (5V 5A) power supply.
To disable this check set usb_max_current_enable=1 in config.txt
or press the power button to temporarily enable usb_max_current_enable
and continue booting.
See https://rptl.io/rp5-power\_supply for more information
***
```

The bootloader will prompt you to "press power button to continue", which effectively sets `usb_max_current_enable=1` for the current boot temporarily to allow you to boot from USB. Unless your disk has its own external power supply it is possible that there may not be sufficient current available to your Raspberry Pi.

Power supplies and Raspberry Pi OS

The bootloader passes information about the power supply via device-tree `/proc/device-tree/chosen/power`. Users will typically not read this directly.

`max_current`

The max current in mA

`uspd_power_data_objects`

A dump of the PDOs - debug for advanced users

`usb_max_current_enable`

Whether the current limiter was set to high or low

`usb_over_current_detected`

Whether any USB over current occurred during boot before transferring control to the OS

reset_event

The PMIC reset reason e.g. watchdog, over- or under-voltage, over-temperature

The PMIC has built-in ADCs that, among other things, can measure the supply voltage EXT5V_V.

```
vcgencmd pmic_read_adc
```

NOTE

You can't see USB current or anything else connected directly to 5V because this bypasses the PMIC. You should not expect this to add up to the wattage of the source power supply. However, it can be useful to monitor things like the core voltage.

Cooling Raspberry Pi 5

Edit this on GitHub

Raspberry Pi 5 has been designed to handle typical client workloads, uncased, with no active cooling. For heavier loads there are two official options when it comes to cooling your Raspberry Pi 5. Both of these plug into the four-pin JST PWM fan connector located in the upper right of the board between the 40-pin GPIO header and the USB 2 ports.

Both of the available official accessories are actively managed by the Raspberry Pi firmware: at 60°C the fan will be turned on, at 67.5°C fan speed increases, and finally at 75°C the fan steps up to full speed. When the temperature drops back below these limits, the fan will spin down automatically.

NOTE

The USB ports and fan connector both pull from the same 1.6A current limit.

At boot up the fan is turned on and the tacho input is checked to see if the fan is spinning, if it is then the `cooling_fan` device tree overlay is enabled. This overlay is in `bcm2712-rpi-5-b.dtb` by default but with `status=disabled`.

Fan Case

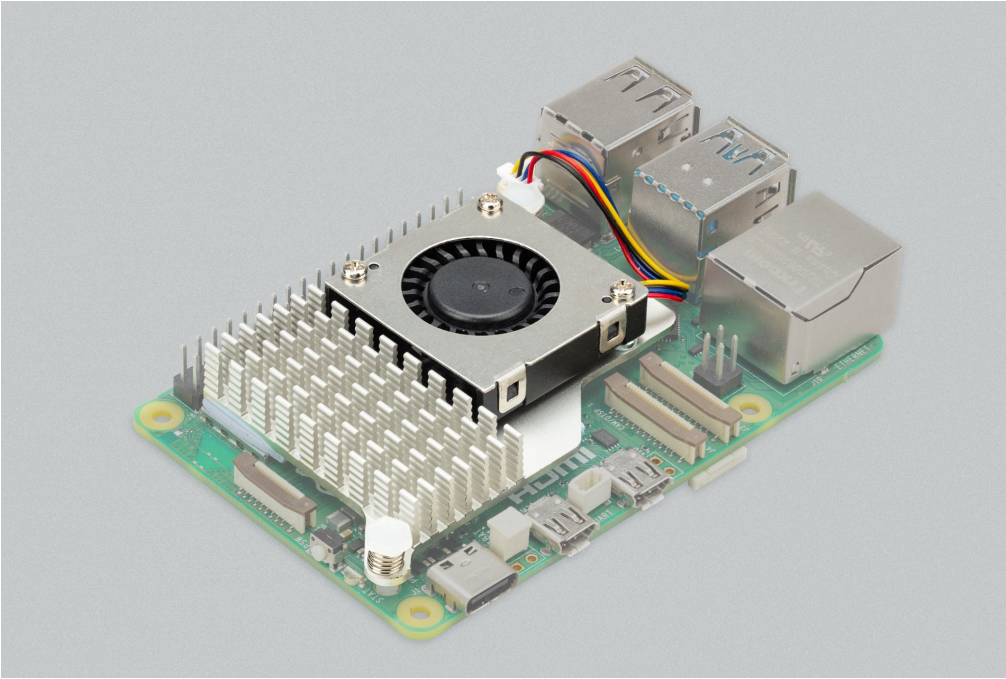


The official Case for Raspberry Pi 5

The official Case for Raspberry Pi 5 comes with an integrated fan. HATs can be fitted directly on top of Raspberry Pi 5 by removing the fan, or on top of the fan by use of 14mm high standoffs and a 19mm GPIO extender. These are available separately from Authorised Resellers.

The integrated 2.79 (max) CFM fan has fluid dynamic bearings selected for low noise and an extended operating lifetime. Air is drawn in through a 360° slot under the lid and blown over a heatsink attached to the BCM2712, then exhausted through connector apertures and vents in the base.

Active Cooler



The Active Cooler for Raspberry Pi 5

The Active Cooler is a single-piece anodised aluminium heatsink with an integrated blower. It has pre-applied thermal pads for heat transfer, and is mounted to the Raspberry Pi 5 board directly using spring-loaded push pins. It connects to the same 4-pin JST connector as the case fan.

A radial blower, again selected for low noise and extended operating lifetime, pushes air through an extruded and milled aluminium heatsink. The cooling performance of the Active Cooler is somewhat superior to the Fan Case, making it particularly suitable for overclockers.

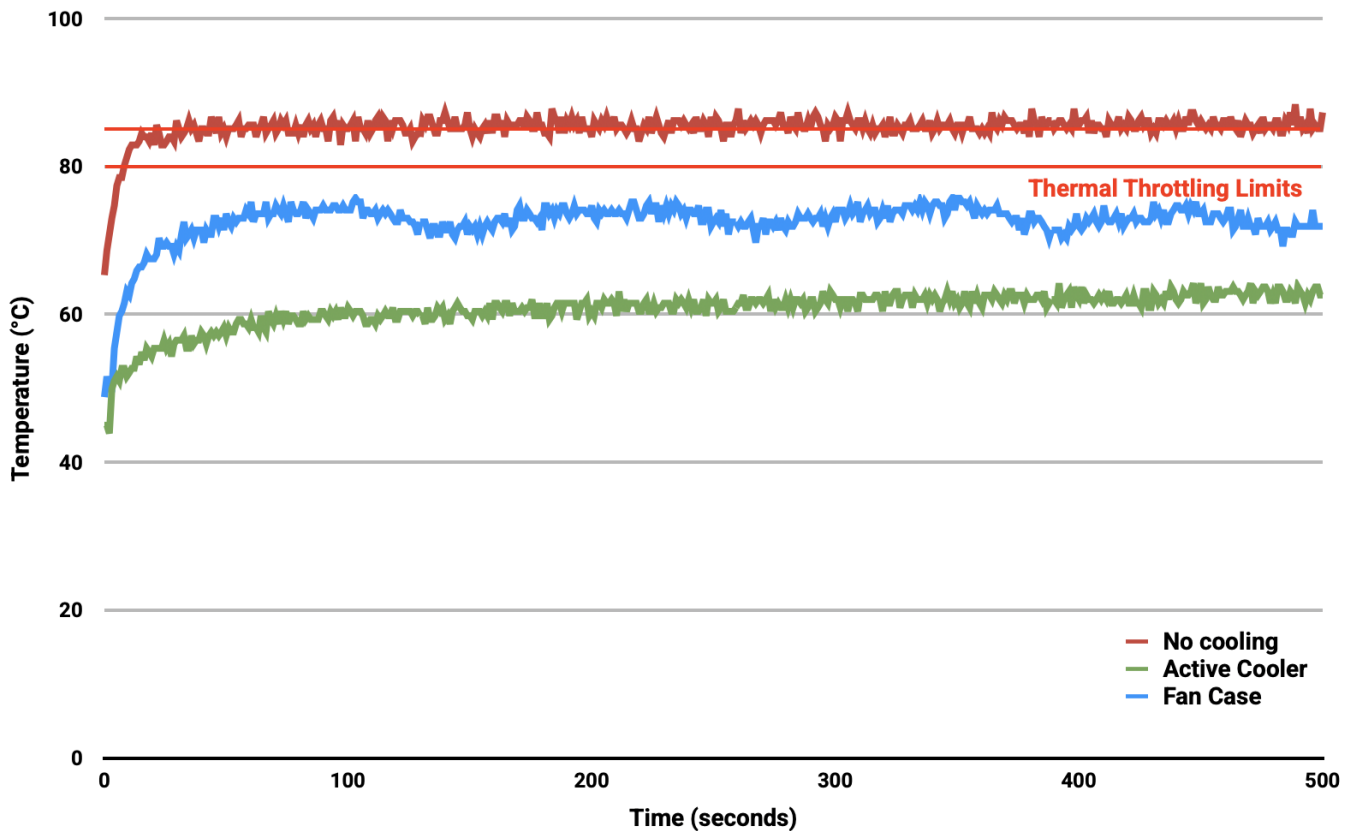
NOTE

Blowers are electromechanical systems that direct air towards a specific direction using a fan and designated channels. While fans displace large volumes of air and promote circulation within a designated area, blowers provide a degree of air pressurisation to drive it out and forward.

The Active Cooler has not been designed to be repeatedly removed from the Raspberry Pi 5.

Thermals

A stress test was started on all four cores of the Raspberry Pi 5 to load the CPU. Two main thermal management regimes were examined during the test: no cooling, and managed active cooling. The managed active cooling tests were run using both the Active Cooler, and the Raspberry Pi Case for Raspberry Pi 5.



Graph of CPU temperature against time.

With no cooling, we see an idle temperature of approximately 65°C, with a maximum temperature above 85°C during extended testing leading to sustained thermal throttling after the temperature reported by the processor rises above throttling limits.

Thanks to the passive heatsink that is fitted with the Active Cooler, we see a much lower idle temperature, around 45°C. During extended testing under load, the fan of the Cooler spins up at low speed to stabilise the CPU temperature at 60°C, with a maximum temperatures of 62 to 63°C being seen during the tests.

Using the fan case we see idle temperatures a couple of degrees hotter than with the Active Cooler on its own, at around 48°C. With the lid removed we see a maximum temperature of approximately 72°C under sustained load, and with the lid in place we see a marginally higher maximum of around 74°C under load. While temperature under load is higher than with the Active Cooler, the maximum temperature under load is still well below the 80 and 85°C throttling temperatures.

Raspberry Pi 5 has been designed to handle typical client workloads, uncased, with no active cooling. For normal use adding cooling is optional, although performance may improve with active cooling. However a heavy continuous load, such as rebuilding the Linux kernel, will force the new Raspberry Pi 5 into thermal throttling. For heavy loads thermal throttling can extend processing times, and passive cooling is probably insufficient thermal management.

However, cooling of any type isn't mandatory, and no harm will come to the Raspberry Pi if left uncooled.

Pinout

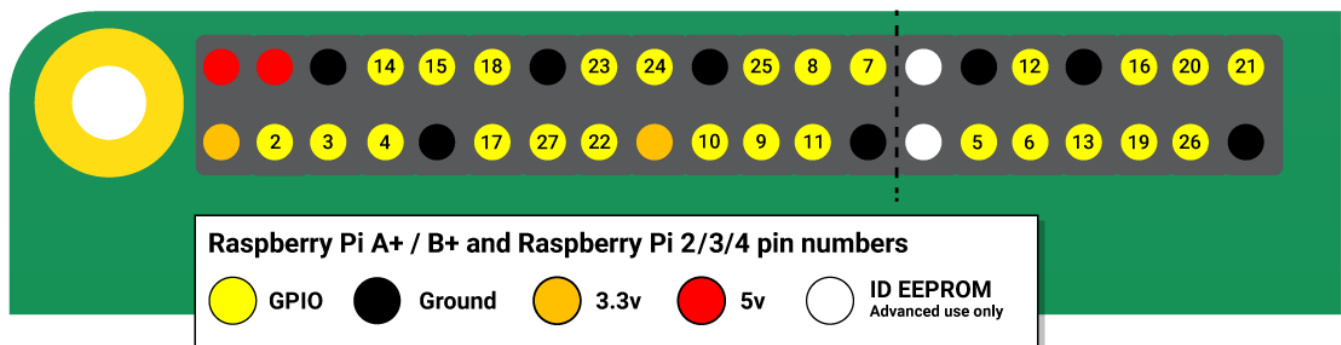
The fan connector has four pins:

Pin	Function
1	+5V
2	PWM
3	GND
4	Tach

Raspberry Pi and GPIO

Edit this on GitHub

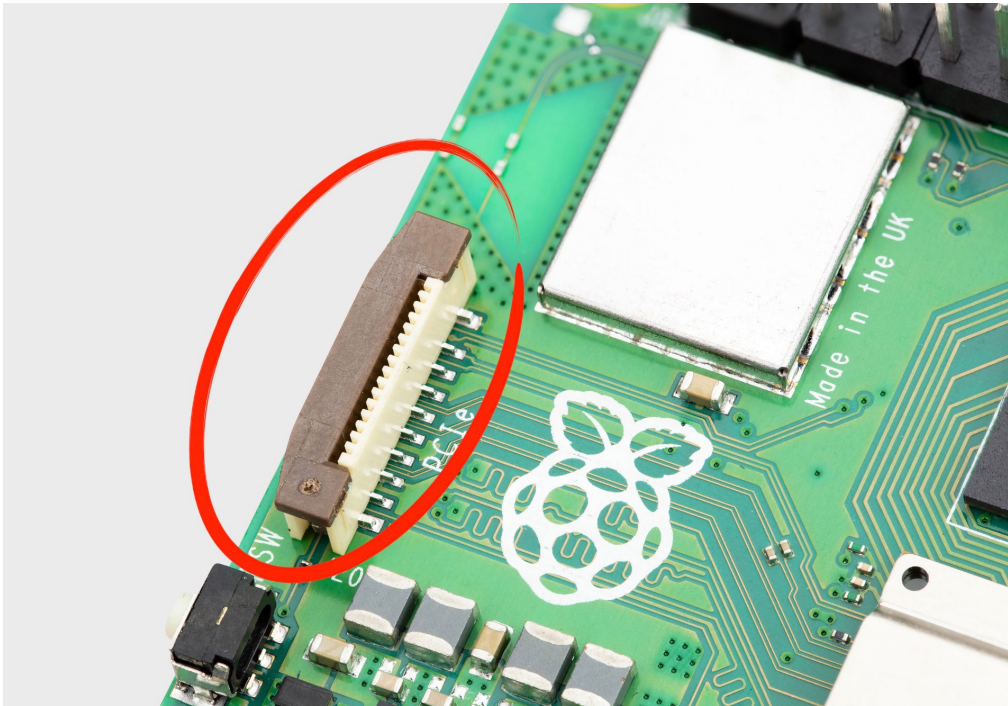
Using the GPIO Zero library makes it easy to get started with controlling GPIO devices with Python. The library is comprehensively documented at gpiozero.readthedocs.io.



More information about using the GPIO pins on Raspberry Pi can be found elsewhere in the documentation.

Raspberry Pi connector for PCIe

Edit this on GitHub



Raspberry Pi connector for PCIe

Raspberry Pi 5 has an FPC connector on the right-hand side of the board. This connector breaks out PCIe Gen 2.0 x1 interface for fast peripherals.

NOTE

An M.2 HAT which provides an M.2 connector to allow the use of NVMe drives is still in prototype, and has not yet been released.

NOTE

More information around the PCIe FPC connector pin out, and other details needed to create third-party devices, accessories, and HATs, will be released at a later point in time.

Enabling PCIe

By default the PCIe connector is not enabled. To enable it you should add the following option into `/boot/firmware/config.txt` and reboot:

```
# Enable the PCIe External connector.  
dtparam=pciex1
```

A more memorable alias for `pciex1` exists, so you can alternatively add `dtparam=nvme` to the `/boot/firmware/config.txt` file.

NOTE

Enumeration of PCIe devices behind a switch is not currently supported.

PCIe Gen 3.0

The connection is certified for Gen 2.0 speeds (5 GT/sec), but you can force it to Gen 3.0 (10 GT/sec) if you add the following lines to your `/boot/firmware/config.txt`.

```
# Enable the PCIe external connector
dtparam=pciex1

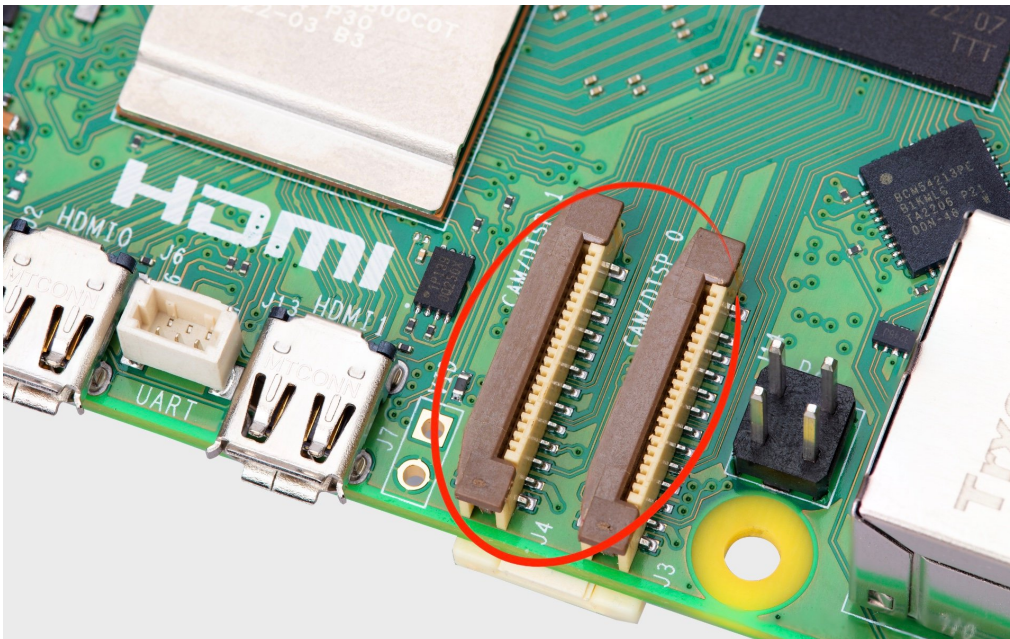
# Force Gen 3.0 speeds
dtparam=pciex1_gen=3
```

WARNING

The Raspberry Pi 5 is not certified for Gen 3.0 speeds, and connections to PCIe devices at these speeds may be unstable.

MIPI CSI/DSI connectors

[Edit this on GitHub](#)



The two MIPI connectors

The CSI and DSI ports found on previous models of Raspberry Pi have been combined into two dual-purpose CSI/DSI (MIPI) ports. To fit onto the board these now use a denser connector pinout, previously only found on

Raspberry Pi Zero and the CM4IO board. You can connect two displays, two cameras, or one camera and one display to these ports.

Attaching cameras

There is no configuration necessary for Raspberry Pi devices like cameras. If you plug a camera into either MIPI connector you will get an appropriate CSI connection, and your camera will be made available to the OS.

Camera cables



Camera cables

There are three Flat Flexible Cables (FFC) available to connect CSI cameras to the Raspberry Pi 5:

- 200 mm camera cable standard to mini
- 300 mm camera cable standard to mini
- 500 mm camera cable standard to mini

NOTE

The Compute Module cable adapters, see schematics and design files, can be used to connect standard cables to the higher-density mini connectors.

Attaching a display

If you are using our 7-inch Touch Display with Raspberry Pi 5, it will not automatically be configured. You will need to add one of the following two lines to your `/boot/firmware/config.txt` file. Attaching the display to the CAM/DISP 1 connector you should add:

```
dtoverlay=vc4-kms-dsi-7inch
```

alternatively, attaching it to the CAM/DISP 0 connector you can add the following line:

```
dtoverlay=vc4-kms-dsi-7inch,dsi0
```

WARNING

At the time of writing, to get touch support on the 7-inch Touch Display you will need to `rpi-update` to get a newer kernel.

Available display cables



Display cables

There are three Flat Flexible Cables (FFC) available to connect DSI displays to the Raspberry Pi 5:

- 200 mm display cable standard to mini
- 300 mm display cable standard to mini
- 500 mm display cable standard to mini

Using non-Raspberry Pi devices

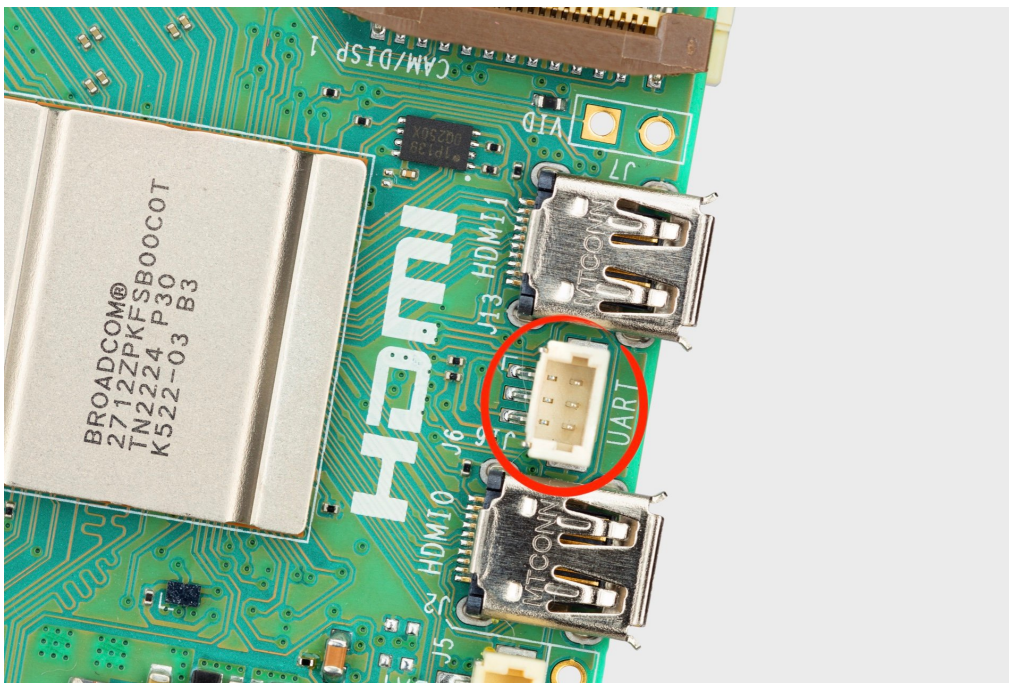
If you are using a non-Raspberry Pi MIPI device — either a camera, or a display — it will not be automatically configured for your Raspberry Pi 5. Instead you will need to add a `dtoverlay` setting into the `/boot/firmware/config.txt` file to correctly configure the right port for the right camera or display.

These `dtoverlay` settings should be provided by the manufacturer of your device. For example, adding `dtoverlay=ov9281` would configure an Omnivision OV9281-based camera on CSI/DSI1, while adding `dtoverlay=ov9281,cam0` would add the same camera to CSI/DSI0.

UART connector

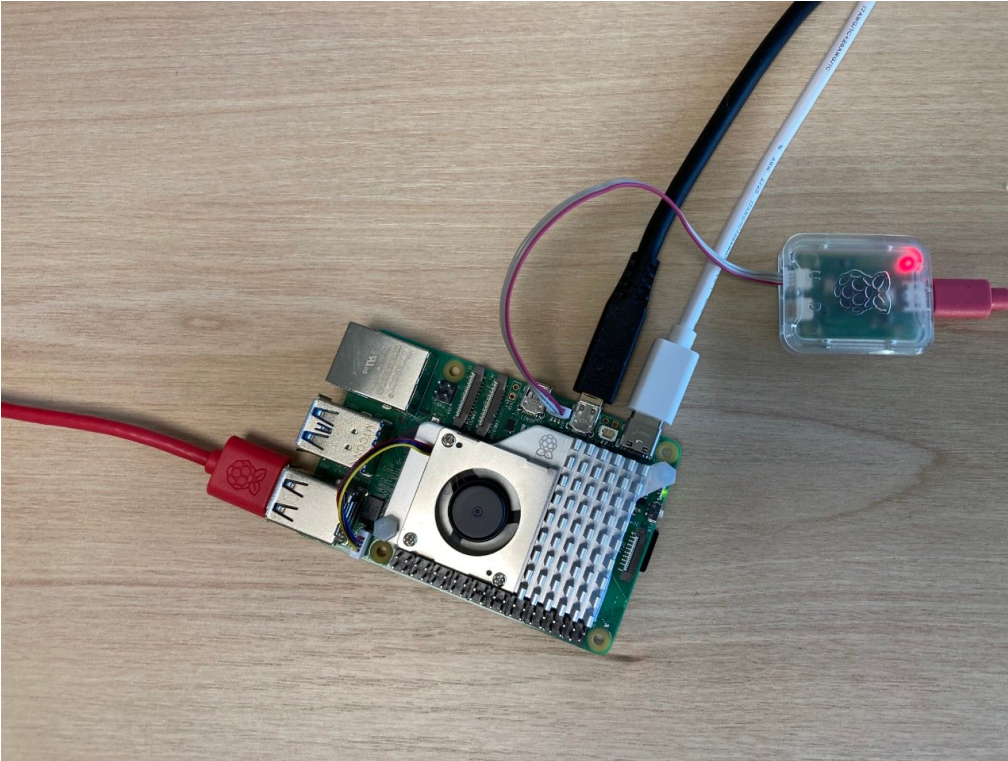
[Edit this on GitHub](#)

The Raspberry Pi 5 has a dedicated debug UART connector. It has a maximum baud rate of 921,600bps, which can be configured with an EEPROM parameter. It is always active and enabled, and provides direct access to early boot and firmware information; as well as a Linux console after boot.



The three-pin UART connector

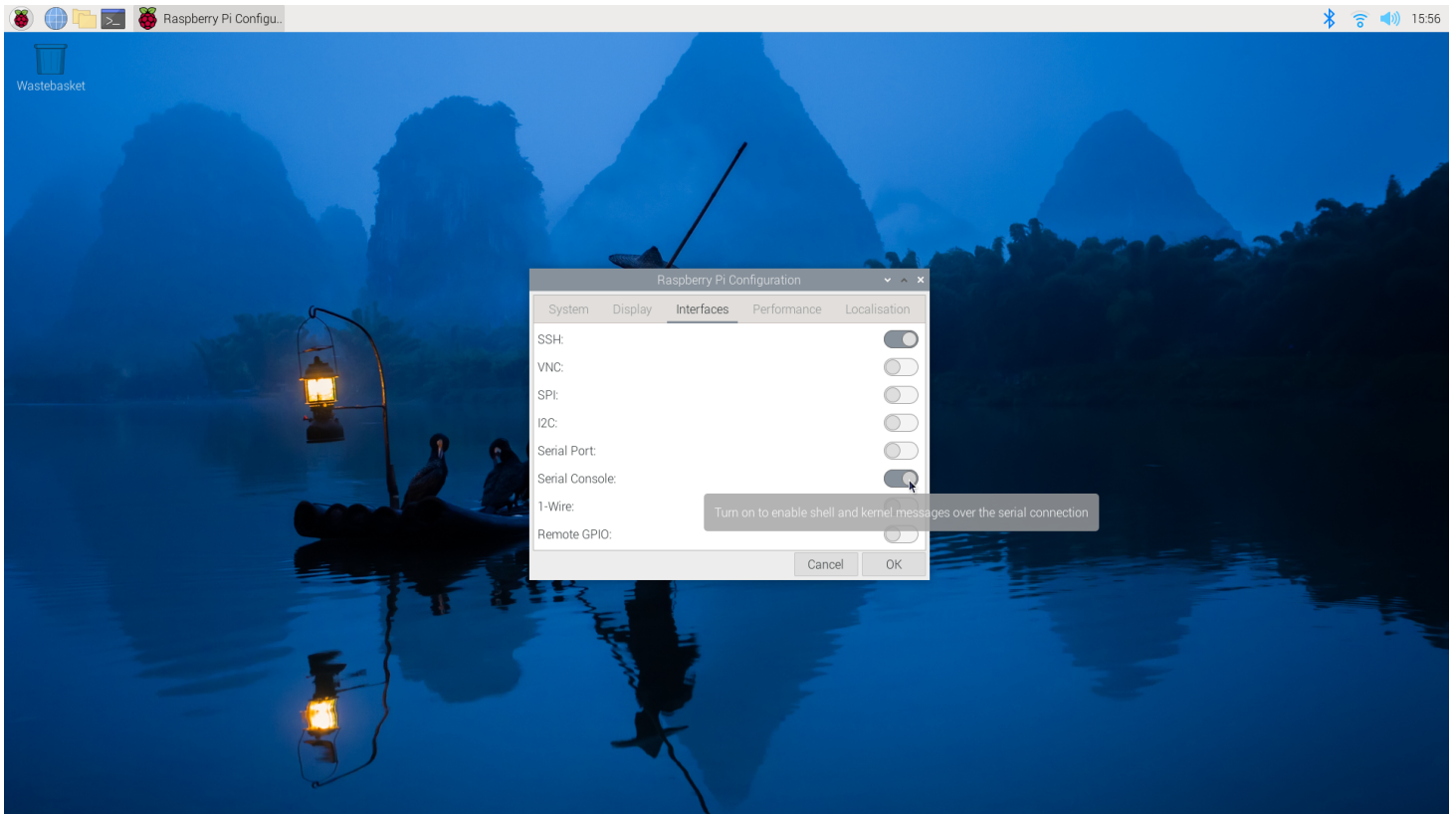
The UART connector is a three-pin header compatible with the Raspberry Pi Debug Connector specification. A cable terminating in an appropriate JST connector allowing you to connect to the UART is provided with the Debug Probe.



The Raspberry Pi Debug Probe connected to Raspberry Pi 5

By default it offers a 115,200 baud 8N1 connection on `/dev/ttyAMA0` (aliased to `/dev/serial0`).

Creating a serial console



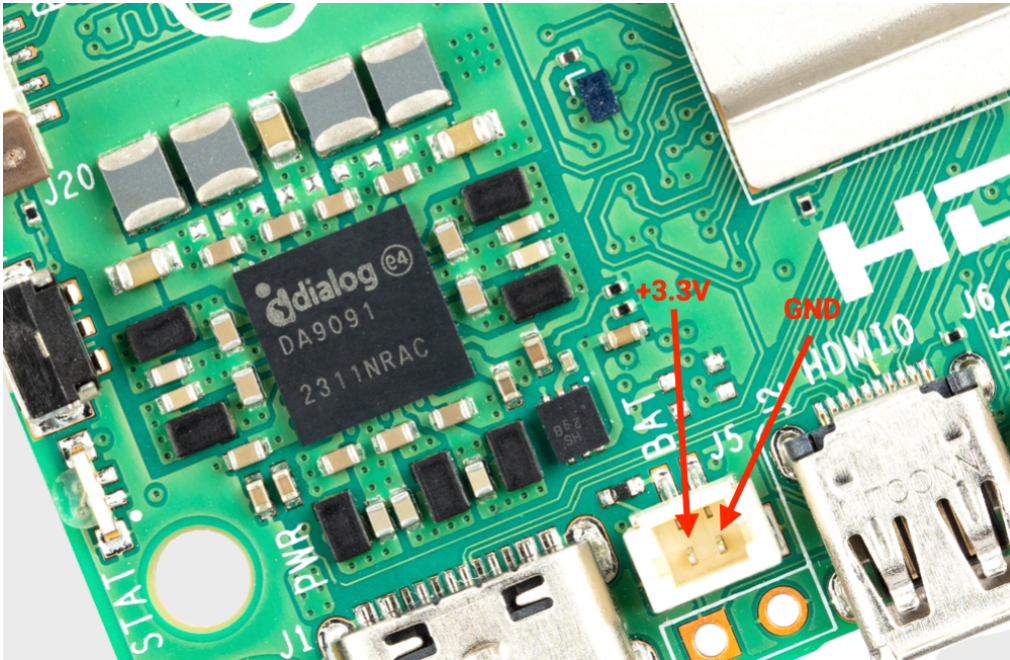
Enabling a serial console

Unlike previous Raspberry Pi models, where you could enable a serial console on GPIO14 and 15, enabling a serial console from `raspi-config` will enable a serial console on the new UART connector on `/dev/ttyAMA0` on Raspberry Pi 5.

Real Time Clock (RTC)

Edit this on GitHub

The Raspberry Pi 5 includes an RTC module. This can be battery powered via the J5 (BAT) connector on the board located to the right of the USB-C power connector.



The J5 battery connector

You can set a wake alarm which will switch the board to a very low-power state (approximately 3mA). When the alarm time is reached, the board will power back on. This can be useful for periodic jobs like time-lapse imagery.

To support the low-power mode for wake alarms you should edit the bootloader config:

```
sudo -E rpi-eeeprom-config --edit
```

adding the following two lines.

```
POWER_OFF_ON_HALT=1  
WAKE_ON_GPIO=0
```

You can test the functionality with:

```
echo +600 | sudo tee /sys/class/rtc/rtc0/wakealarm  
sudo halt
```

That will halt the board into a very low-power state, then wake and restart after 10 minutes.

The RTC also provides the time on boot e.g. in dmesg:

```
[ 1.295799] rpi-rtc soc:rpi_rtc: setting system clock to 2023-08-16T15:58:50 UTC (1692201530)
```

...which may be useful if you don't have an internet connection for getting time through NTP.

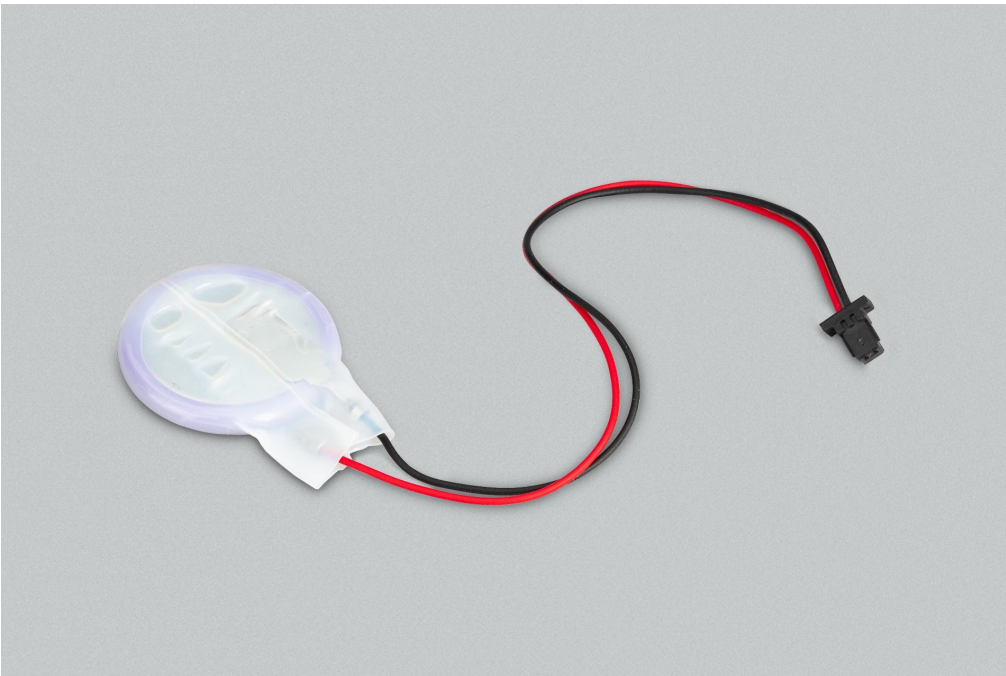
NOTE

The RTC is still usable even when there is no backup battery attached to the J5 connector.

Adding a backup battery

WARNING

We do not recommend using a primary lithium cell for the RTC, as it has a trickle charge circuit which is disabled by default. If enabled, this will kill the cell quickly.



Lithium-manganese rechargeable RTC battery

The official battery part is a rechargeable lithium manganese coin cell, with a pre-fitted two-pin JST plug and an adhesive mounting pad. This is suitable for powering the Raspberry Pi 5 RTC when the main power supply for the board is disconnected, with a power-off current draw measuring in single-digit μA , giving a retention time that can be measured in months.

Enabling trickle charging

Trickle charging of the battery is disabled by default. There are `sysfs` files that show the current trickle charging voltage and limits:

```
/sys/devices/platform/soc/soc:rpi_rtc/rtc/rtc0/charging_voltage:0
/sys/devices/platform/soc/soc:rpi_rtc/rtc/rtc0/charging_voltage_max:4400000
/sys/devices/platform/soc/soc:rpi_rtc/rtc/rtc0/charging_voltage_min:1300000
```

If you add `rtc_bbat_vchg` to `/boot/firmware/config.txt`:

```
dtparam=rtc_bbat_vchg=3000000
```

...and reboot, you'll see:

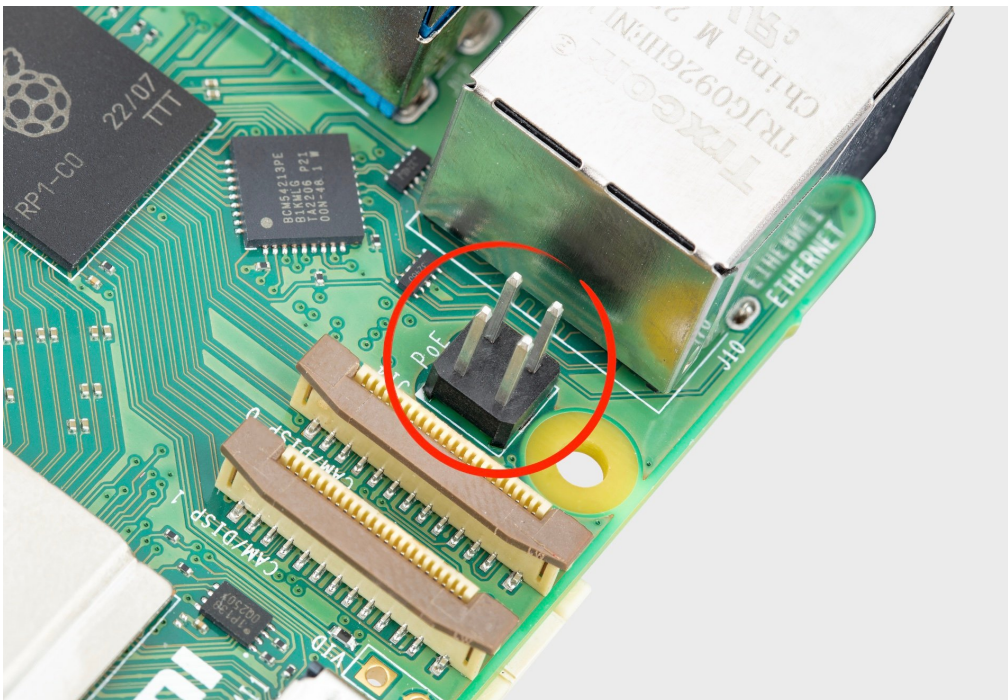
```
/sys/devices/platform/soc/soc:rpi_rtc/rtc/rtc0/charging_voltage:3000000
/sys/devices/platform/soc/soc:rpi_rtc/rtc/rtc0/charging_voltage_max:4400000
/sys/devices/platform/soc/soc:rpi_rtc/rtc/rtc0/charging_voltage_min:1300000
```

The battery will be trickle charging. Remove the `dtparam` line from `config.txt` to stop the trickle charging.

Power over Ethernet (PoE) connector

[Edit this on GitHub](#)

The Ethernet jack on Raspberry Pi 5 is PoE+ capable, supporting the IEEE 802.3at-2009 PoE standard. The Raspberry Pi 5 has a 4-pin Power-over-Ethernet (PoE) header block, located between the Ethernet jack and the two MIPI connectors.



Raspberry Pi 5 PoE header

The Raspberry Pi PoE+ HAT for Raspberry Pi 5 is an add-on board for Raspberry Pi 5 computers which connects to the PoE header. It is used to power your Raspberry Pi via an Ethernet cable, provided that power-sourcing equipment is installed on the Ethernet network.

NOTE

The PoE+ HAT for Raspberry Pi 5 is still in prototype, and has not yet been released.