

---

# Changing the MAC Address on Raspberry Pi Single-Board Computers

Raspberry Pi Ltd

2022-04-29: githash: ba7441c-clean

# Colophon

© 2020-2022 Raspberry Pi Ltd (formerly Raspberry Pi (Trading) Ltd.)

This documentation is licensed under a Creative Commons [Attribution-NoDerivatives 4.0 International](#) (CC BY-ND).

build-date: 2022-04-29

build-version: githash: ba7441c-clean

## Legal Disclaimer Notice

TECHNICAL AND RELIABILITY DATA FOR RASPBERRY PI PRODUCTS (INCLUDING DATASHEETS) AS MODIFIED FROM TIME TO TIME ("RESOURCES") ARE PROVIDED BY RASPBERRY PI LTD ("RPL") "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW IN NO EVENT SHALL RPL BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THE RESOURCES, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

RPL reserves the right to make any enhancements, improvements, corrections or any other modifications to the RESOURCES or any products described in them at any time and without further notice.

The RESOURCES are intended for skilled users with suitable levels of design knowledge. Users are solely responsible for their selection and use of the RESOURCES and any application of the products described in them. User agrees to indemnify and hold RPL harmless against all liabilities, costs, damages or other losses arising out of their use of the RESOURCES.

RPL grants users permission to use the RESOURCES solely in conjunction with the Raspberry Pi products. All other use of the RESOURCES is prohibited. No licence is granted to any other RPL or other third party intellectual property right.

HIGH RISK ACTIVITIES. Raspberry Pi products are not designed, manufactured or intended for use in hazardous environments requiring fail safe performance, such as in the operation of nuclear facilities, aircraft navigation or communication systems, air traffic control, weapons systems or safety-critical applications (including life support systems and other medical devices), in which the failure of the products could lead directly to death, personal injury or severe physical or environmental damage ("High Risk Activities"). RPL specifically disclaims any express or implied warranty of fitness for High Risk Activities and accepts no liability for use or inclusions of Raspberry Pi products in High Risk Activities.

Raspberry Pi products are provided subject to RPL's [Standard Terms](#). RPL's provision of the RESOURCES does not expand or otherwise modify RPL's [Standard Terms](#) including but not limited to the disclaimers and warranties expressed in them.

## Document version history

Release	Date	Description
1.0	14 April 2021	Initial release
1.1	16 June 2021	Update for software-level setting
1.2	1 December 2021	Update for bootloader-based MAC changes
1.3	27 April 2022	Copy edit, public release

## Scope of document

This document applies to the following Raspberry Pi products:

Pi 0			Pi 1		Pi 2		Pi 3	Pi 4	Pi 400	CM 1	CM 3	CM 4	Pico
0	W	H	A	B	A	B	B	All	All	All	All	All	All
	*	*		*		*	*	*	*	*	*	*	

# Introduction

This whitepaper describes how to set a specific media access control (MAC) address on your Raspberry Pi device. It applies to both Ethernet and wireless interfaces.

Two mechanisms are described, one for devices up to the Pi 3 and one for the Pi 4/400 and CM 4. These latter three devices use the BCM2711 system on a chip (SoC), which has a built-in Ethernet port and hence defines its MAC addresses in a different way.

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.

# Techniques

## How are MAC addresses generated?

You can display the current MAC address for any Raspberry Pi device using

```
ip link
```

or, for a specific device (e.g. `eth0`), using

```
ip link show eth0
```

The MAC address is the set of six hexadecimal numbers of the form `aa:bb:cc:dd:ee:ff`.

On devices prior to the Raspberry Pi 4x the MAC address is generated from the Raspberry Pi serial number. For example, if your Raspberry Pi serial number is `58d2ec5c`, the MAC address will be generated from the bottom six nibbles, combined with the Raspberry Pi Foundation Organizationally Unique Identifier (OUI), which is `b8:27:eb`, so the final MAC address would be `b8:27:eb:d2:ec:5c`.

This address is generated on startup by the firmware, and passed on to the Linux kernel for use by the Ethernet driver.

On the Raspberry Pi 4x range with the 2711 SoC, the MAC address is generated by combining a Raspberry Pi Ltd OUI (`dc:a6:32` or `e4:5f:01`) with a final three bytes that are generated at production time from a sequential set, for example `dc:a6:32:c4:7c:5f`.

This MAC address is stored in the one-time programmable (OTP) memory of the SoC and cannot be permanently changed, but can be overridden in software as described in a later section.

### Raspberry Pi Ltd OUIs

Raspberry Pi MAC addresses are currently allocated from three OUI ranges, although new ranges will be added as each existing one is used up. Each range provides `0xfffff` (six nibbles, 16777216 decimal) different IDs.

OUI	Owner
DC:A6:32	Raspberry Pi Ltd
E4:5F:01	Raspberry Pi Ltd
B8:27:EB	Raspberry Pi Foundation

## Setting a new MAC address from the command line

You can temporarily change the MAC address of the Ethernet adapter with the `ip` command:

```
sudo ip link set eth0 down
sudo ip link set eth0 address aa:bb:cc:dd:ee:ff
sudo ip link set eth0 up
```

The same process also applies to the wireless networking MAC address:

```
sudo ip link set wlan0 down
sudo ip link set wlan0 address aa:bb:cc:dd:ee:ff
```

```
sudo ip link set wlan0 up
```

**NOTE**

These commands are *not* persistent over a reboot, so will need to be retyped each time or added to a login script to make it automatic.

## Setting a persistent alternative MAC address on devices using SoCs prior to 2711 (Raspberry Pi 1–3)

On these devices there is an area in the OTP memory that can be set to the new MAC address; if present, this address will be used instead of the address based on the serial number.

Programming the OTP requires using the `SET_MAC_ADDRESS_OTP` mailbox call (ID: 0x00030051). There is a utility provided on all Raspberry Pi OS installations for making mailbox calls to the firmware. The mailbox system accepts data in 32-bit words in little-endian format, so to set a MAC address of aa:bb:cc:dd:ee:ff you will need to do the following:

```
/opt/vc/bin/vcmailbox 0x00030051 6 6 0xddccbaa 0x0000ffee
```

The two '6' parameters are the length, in bytes, of the question and response buffers.

Note that this operation is irreversible: once an OTP bit has moved from 0 to 1 it can never be set back to 0, so please be careful when using this command.

## Setting a persistent alternative MAC address on devices using the 2711 SoC (Raspberry Pi 4, 400, CM 4)

On the 2711 SoC the MAC address is already stored in the OTP memory during production, and cannot be permanently changed. However, there is another option to override the preprogrammed MAC address, now available in all boot modes. The boot EEPROM configuration on these devices has a setting to update the MAC address:

```
MAC_ADDRESS=aa:bb:cc:dd:ee:ff
```

To edit the EEPROM configuration, use the following command:

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

Add the required `MAC_ADDRESS`, save, and reboot. The device should now use that MAC address when booting.

## Possible schemes for programming the MAC address on Raspberry Pi 4 devices during production

While there are any number of mechanisms for getting customer-specific MAC addresses on to a Raspberry Pi 4 device on a production line, here are some suggestions that may give some idea of the options available.

In the vast majority of cases each device will require a unique MAC address to be assigned, which means an image with a predefined MAC address is inappropriate. Schemes are therefore needed to provide a unique address after the image has been installed. Raspberry Pi 4 devices during production are assigned unique MAC addresses, but with a Raspberry Pi Ltd OUI.

One possible algorithm that should give almost non-repeating numbers could be to take the entire MAC address and replace the Raspberry Pi Ltd OUI segment (the first three bytes) with a manufacturer-specific OUI, leaving the last three bytes as programmed. However, because of the large number of Raspberry Pi 4 devices that have been made there are multiple OUI ranges, so simply replacing the last three bytes could result in duplicate values. One option here is to have two or more customer OUI ranges to replace the two or more OUI ranges allocated to Raspberry Pi Ltd.

An alternative scheme could be to use an external system to generate unique MAC addresses. The programmed image could be designed, on first boot, to read from a MAC address provider via a script, which subsequently sets up the system to use the provided MAC address. The initial boot could be part of a provisioning/testing process on the production line; a server could provide unique numbers on request, and connect to a backend database for further provisioning information.



**Raspberry Pi**

Raspberry Pi is a trademark of the Raspberry Pi Foundation

---

**Raspberry Pi Ltd**