Transitioning From Raspberry Pi Compute Module 1/3 To Compute Module 4

Raspberry Pi Ltd

Colophon

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	30 September 2021	Initial release
1.1	27 April 2022	Copy edit, public release
1.2	11 Dec 2023	Change URL from .org to .com, grammar fixes.

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 is for those who wish to move from using a Raspberry Pi Compute Module (CM) 1 or 3 to a Raspberry Pi CM 4.

There are a number of reasons why this might be desirable:

- Greater computing power
- More memory
- More High-Definition Multimedia Interface (HDMI) ports
- Higher-resolution output up to 4Kp60
- Faster Ethernet
- Faster wireless
- Faster Universal Serial Bus (USB)
- Better availability

From a software perspective the move from Raspberry Pi CM 1/3 to Raspberry Pi CM 4 is relatively painless, as a Raspberry Pi operating system (OS) image should work on all platforms. If, however, you are using a custom kernel, some things will need to be considered in the move. The hardware changes are considerable, and the differences are described in a later section.

Terminology

Legacy graphics stack: A graphics stack wholly implemented in the VideoCore firmware blob with a shim application programming interface exposed to the kernel. This is what has been used on the majority of Raspberry Pi Ltd Pi devices since launch, but is gradually being replaced by (F)KMS/DRM.

FKMS (vc4-fkms-v3d): Fake Kernel Mode Setting. While the firmware still controls the low-level hardware (for example the High-Definition Multimedia Interface (HDMI) ports, Display Serial Interface (DSI), etc.), standard Linux libraries are used in the kernel itself. Deprecated in the latest Raspberry Pi OS release in favour of KMS.

KMS: The full Kernel Mode Setting driver. Controls the entire display process, including talking to the hardware directly with no firmware interaction.

DRM: Direct Rendering Manager, a subsystem of the Linux kernel used to communicate with graphical processing units. Used in partnership with FKMS and KMS.

Compute Module comparison

Functional differences

The following table gives some idea of the basic electrical and functional differences between the models.

Feature	CM 1	CM 3+	CM 4
Processor	BCM2835	BCM2837	BCM2711
Random access memory	512MB	1GB	1/2/4/8GB
Embedded MultiMediaCard memory	—	8/16/32GB	8/16/32GB
Ethernet	None	None	1.0Gbps
USB	1 × USB 2.0	1 × USB 2.0	1 × USB 2.0
HDMI	1 × 1080p60	1 × 1080p60	2 × 4K
Form factor	SODIMM	SODIMM	55mm × 40mm

Physical differences

The CM 1 and CM 3/3+ form factor is based around a small-outline dual inline memory module (SODIMM) connector. This provides a physically compatible upgrade path between these devices.

> ℹ️ **NOTE**
>
> The CM 1 and CM 3/3+ *cannot* be used in a memory slot as a SODIMM device.

Due to the increased functionality available on the BCM2711 system on a chip (SoC), a new form factor was developed based on two 100-pin connectors. This was launched as the CM 4. This form factor has a higher density and is not backward compatible with the CM 1/3 form factor.

Hardware changes

The new form factor of the CM 4 means that any baseboards developed for CM 1/3 will require a redesign.

Electrical differences

There are a significant number of electrical differences between the CM 1 and CM 4; consult the CM 4 datasheet for full details. Some of the major differences are:

- Reduced number of general purpose input/output (GPIO) pins due to the on-board wireless and Ethernet interfaces

- Addition of a second HDMI port

- Addition of a Peripheral Component Interconnect Express (PCIe) interface

- Simplified external power supply unit (PSU), meaning that only a +5V supply is required

- On-board PSUs capable of supplying power to the baseboard

SODIMM to 55mm × 40mm

The new connectors used on the CM 4 (2 × 100 pin) require careful alignment during manufacture, the SODIMM being much more forgiving due to its design. Please refer to the Raspberry Pi CM 4 datasheet for more details on the manufacturing tolerances for the 100-pin connectors.

Software changes required

If you are using a fully updated Raspberry Pi OS then the software changes needed when moving between any Raspberry Pi Ltd boards are minimal; the system automatically detects which board is running and will set up the operating system appropriately. So, for example, you can move your OS image from a Raspberry Pi CM 3+ to a Raspberry Pi CM 4 and it should work without changes.

> ℹ️ **NOTE**
>
> You should ensure that your Raspberry Pi OS installation is up to date by going through the standard update mechanism. This will ensure that all firmware and kernel software is appropriate for the device in use.

If you are developing your own minimal kernel build or have any customisations in the `boot` folder then there may be some areas where you will need to ensure you are using the correct setup, overlays, and drivers.

While using an updated Raspberry Pi OS *should* mean that the transition is fairly transparent, for some 'bare metal' applications it is possible that some memory addresses have changed and a recompilation of the application is required. See the BCM2711 peripherals documentation for more details on the extra features of the BCM2711 and register addresses.

Graphics

By default the Raspberry Pi CM 1–3+ use the legacy graphics stack, while the Raspberry Pi CM 4 uses the KMS graphics stack when using the latest Raspberry Pi OS.

While it is possible to use the legacy graphics stack on the Raspberry Pi CM 4 or 4S, this does not support 3D acceleration, so moving to KMS is recommended (or FKMS if using Raspberry Pi OS Buster).

HDMI

The Raspberry Pi CM 4 has two HDMI ports that are capable of dual 4Kp30 output, or a single port at 4Kp60. From a software perspective, there is little that needs to be done to take advantage of the extra ports and resolution. Although all graphics stacks can detect the dual ports, when using FKMS or KMS the graphics system provides stretching of the desktop over both displays, duplication, rotation, and flipping.

There are some caveats to using the higher-resolution output facilities:

- 4Kp60 is only available on HDMI port 0 and is disabled by default
    - Enable 4Kp60 using `hdmi_enable_4kp60=1` in `config.txt`
- The H264 decoder can handle a maximum of 1080p60
    - Use the High-Efficiency Video Coding (HEVC) decoder for higher-resolution video

Ethernet

On the Raspberry Pi 3B+ the Ethernet interface is provided by a separate chip, a Microchip LAN7515, which is connected via USB 2.0, giving a maximum throughput of about 0.35Gbps. This chip uses the LAN78xx driver, which is enabled using the `CONFIG_USB_LAN78XX` kernel option.

The Raspberry Pi 4 SoC has a built-in Ethernet media access controller (MAC) and physical layer transceiver (Phy), which provides full gigabit networking, so has a theoretical maximum of 1.0Gbps. This Phy is supplied by Broadcom and uses the Broadcom GENET (Gigabit Ethernet) controller driver. This is enabled in a kernel build using `CONFIG_BCMGENET`.

Apart from ensuring that the correct drivers are installed when building your Linux kernel, no other changes are required; aside from a difference in speed, no functional difference should be seen at the user level.

ⓘ NOTE

If you require any sort of network-based root file system or network booting, then the Ethernet drivers need to be built into the kernel, not as modules.

Raspberry Pi