# Raspberry Pi Compute Module 5

A Raspberry Pi for deeply embedded applications.

# Colophon



build-date: 2024-11-27
build-version: b971495-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.

# Table of contents

# Chapter 1. Introduction

## 1.1. Introduction

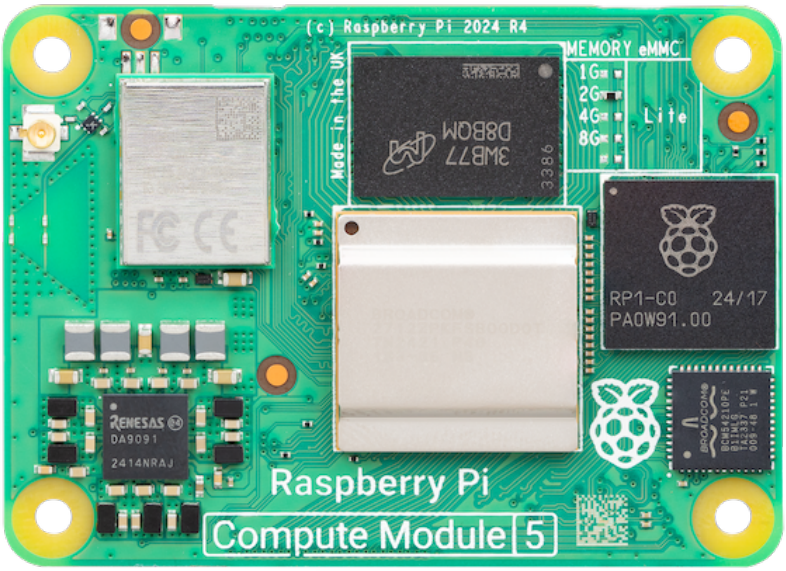



*Figure 1. The front of Raspberry Pi Compute Module 5 (CM5).*

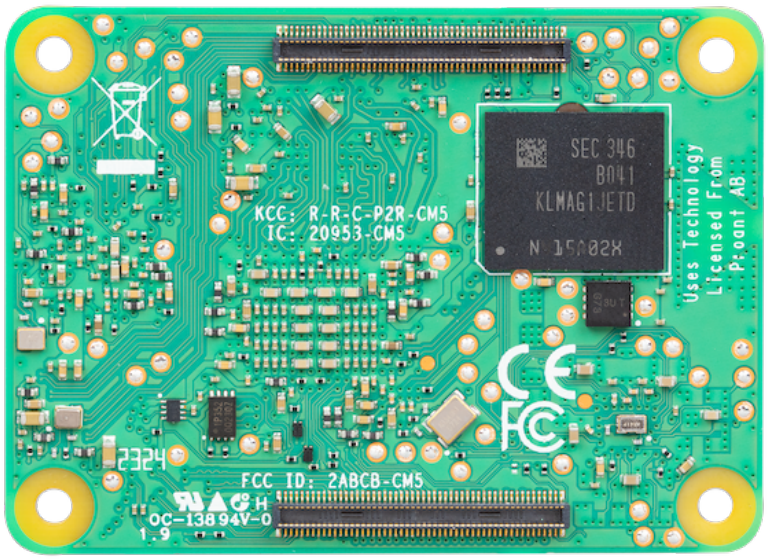



*Figure 2. The back of Raspberry Pi Compute Module 5 (CM5).*

The Raspberry Pi Compute Module 5 (CM5) is a System on Module (SoM) containing processor, memory, eMMC Flash, and supporting power circuitry. These modules allow a designer to leverage the Raspberry Pi hardware and software stack in their own custom systems and form factors. In addition, these modules have extra IO interfaces over and above what is available on the Raspberry Pi boards, opening up more options for the designer.

The design of the CM5 is loosely based on the Raspberry Pi 5, and for cost-sensitive applications it can be supplied without the eMMC fitted; this version is called the Raspberry Pi Compute Module 5 Lite (CM5Lite).

The new CM5 and CM5Lite are mostly compatible with the previous generation of compute module. The previous generation of compute modules are still for sale and will remain in production till at least January 2034. The electrical interface of the CM5 is via two 100-pin high density connectors.

The main change in pinout is to support the addition of 2 USB 3.0 ports.

**NOTE**

Unless otherwise stated, within this document CM5 also refers to CM5Lite.

## 1.2. Features

Key features of the CM5 are as follows:

- Broadcom BCM2712, quad core Cortex-A76 (ARMv8) 64-bit SoC @ 2.4GHz
- Small Footprint 55mm × 40mm × 4.7mm module
  - 4 × M2.5 mounting holes
- 4kp60 HEVC decoder
- OpenGL ES 3.1 graphics, Vulkan 1.2
- Options for 2GB, 4GB, or 8GB LPDDR4-4267 SDRAM with ECC (see Appendix D)
- Options for 0GB **(CM5Lite)**, 16GB, or 32GB eMMC flash memory (see Appendix D)
  - Peak eMMC bandwidth 400MBps (four times as faster as previous Compute Modules)
- Option (see Appendix D) for certified radio module with:
  - 2.4 GHz, 5.0 GHz IEEE 802.11 b/g/n/ac wireless
  - Bluetooth 5.0, BLE
  - On board electronic switch to select between PCB trace or external antenna
- Gigabit Ethernet PHY supporting IEEE 1588
- 1 × PCIe 1-lane Host, Gen 2 (5Gbps)
- 1 × USB 2.0 port (high speed)
- 2 × USB 3.0 ports, supporting simultaneous 5Gbps operation
- up to 30 × GPIO supporting either 1.8V or 3.3V signalling and peripheral options:
  - Up to 5 × UART
  - Up to 5 × I2C
  - Up to 5 × SPI
  - 1 × SDIO interface
  - 1 × DPI (parallel RGB display)
  - 1 × I2S
  - Up to 4× PWM channels
  - Up to 3× GPCLK outputs
- 2 × HDMI 2.0 ports (supports up to 4Kp60 on each port simultaneously)
- 2 × 4-lane MIPI ports
  - supporting both DSI (display port) and CSI-2 (camera port)

- 1 × SDIO 2.0 **(CM5Lite)**
- Single +5V PSU input supports USB PD for up to 5A @ 5V
- Real-time clock (RTC), powered from external battery

# Chapter 2. Interfaces

## 2.1. Wireless

The CM5 can be supplied with an on-board wireless module based on the Cypress CYW43455 supporting both:

- 2.4 GHz, 5.0 GHz IEEE 802.11 b/g/n/ac wireless
- Bluetooth 5.0, BLE

These wireless interfaces can be individually enabled or disabled as required. For instance, in the case of a kiosk application, a service engineer could enable wireless operation and then disable it once finished.

The CM5 has an on-board antenna. If used it should be positioned in the product such that it is not surrounded by metal, including any ground plane (see Chapter 3 for further details). Alternatively there is a standard U.FL connector on the module, see Figure 1, so that an external antenna can be used.

Raspberry Pi Ltd has an antenna kit which is certified to be used with the CM5. If you use a different antenna, separate certification is required.

**WARNING**

Raspberry Pi Ltd does not assist with certification for third-party antennas.

Antenna selection (internal or external) happens at boot time using the `config.txt` file. During operation, this selection cannot be changed. Append one of the following lines to `config.txt` to select an antenna:

- `dtparam=ant1` selects the internal antenna
- `dtparam=ant2` selects the external antenna

### 2.1.1. `WL_nDisable`

This pin serves a number of functions;

1. It can be used to monitor the enable/disable state of wireless networking. A logic high means the wireless networking module is powered up.
2. When driven or tied low it prevents the wireless network module from powering up. This is useful to reduce power consumption or in applications where it is required to physically ensure the wireless networking is disabled. If the interface is enabled after being disabled, you must reinitialise the wireless interface driver.

**NOTE**

On CM5 modules without wireless, this pin is reserved.

### 2.1.2. `BT_nDisable`

This pin serves a number of functions;

1. It can be used to monitor the enable/disable state of Bluetooth. A logic high means the Bluetooth module is powered up.
2. When driven, or tied low, it prevents the Bluetooth module from powering up. This is useful to reduce power consumption, or in applications where it is required to physically ensure the Bluetooth is disabled. If the interface is enabled after being disabled, you must reinitialise the Bluetooth interface driver.

**NOTE**

On CM5 modules without wireless, this pin is reserved.

## 2.2. Ethernet

The CM5 has an on-board Gigabit Ethernet PHY — the Broadcom BCM54210PE — some of the major features of this PHY include;

- IEEE 1588-2008 compliant
- MDI crossover, pair skew and pair polarity correction

A standard 1:1 RJ45 MagJack is all that is necessary to provide an Ethernet connection to the CM5. Typical wiring of a MagJack supporting PoE, and with added ESD protection, can be seen in Figure 3.

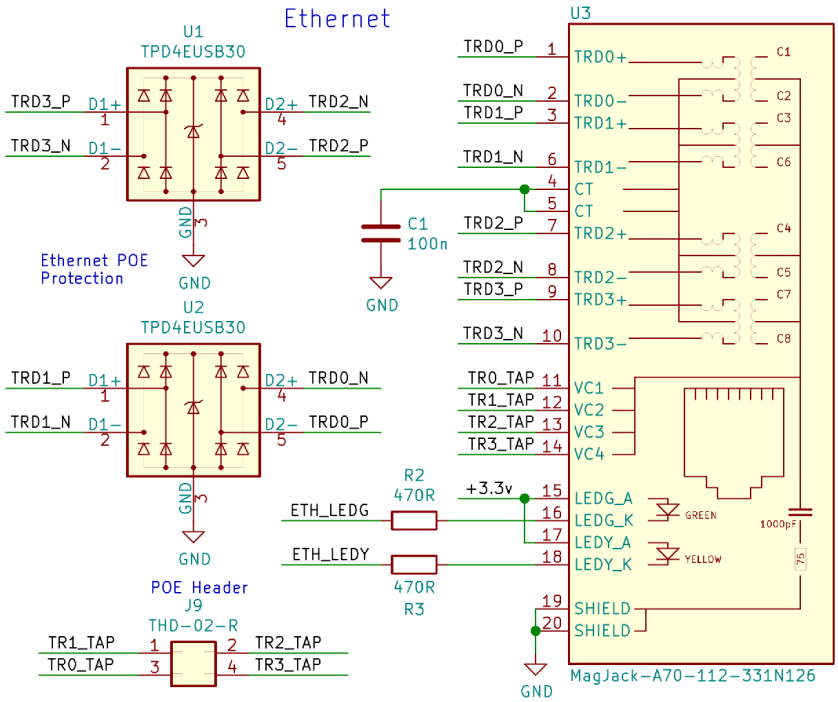



*Figure 3. Ethernet schematic interface for the Raspberry Pi Compute Module 5 supporting PoE, and with added ESD protection.*

The differential Ethernet signals should be routed as 100Ω differential pairs, with suitable clearances. Length matching between pairs should be better than 50mm, so in the typical case no length matching is required. However the signals within a pair need to be length matched, ideally to better than 0.15mm.

The PHY also supports up to 2 LEDs to give user status feedback, these are active-low. These LEDs can have a range of functions, and you should consult your OS driver to see which functions are supported by your driver.

The PHY also provides `SYNC_OUT` at 3.3V signalling to support IEEE 1588-2008. This pin can be defined as an input if required.

## 2.3. PCIe (Gen2 ×1)

The CM5 has an internal PCIe 2.0 ×1 host controller. It may be used as Gen 3.0, but this mode is unsupported and may not work in all cases.

**WARNING**

You should ensure that there is a suitable OS driver for any host controller that is chosen before proceeding to a prototype.

Connecting a PCIe device follows the standard PCIe convention. The CM5 has on-board AC coupling capacitors for the `PCIe_TX` signals. However the `PCIe_RX` signals need external coupling capacitors close to the driving source (the device `TX`), if you are using an external PCIe/NVMe card these capacitors will be on-board. The PCIe convention is that if you are wiring directly to an IC then the TX and RX pairs need to be swapped (i.e. TX → RX, RX → TX). If you are wiring to a connector then this is typically labelled from the host point of view and so TX/RX swaps aren't required. Additionally the `PCIe_CLK_nREQ` must be connected to ensure the CM5 produces a clock signal, and the `PCIe_nRST` should also be connected to ensure the device is correctly reset when required. `PCIe_nWAKE` signal is also available, but isn't currently supported in software.

The differential PCIe signals should be routed as 90Ω differential pairs, with suitable clearances. There is no need to match the lengths between pairs, only the signals within a pair need to be length matched ideally to better than 0.1mm.

## 2.4. USB 3.0 (Super speed)

Each USB 3.0 interface supports up to 5Gb/s signalling simultaneously. The differential pair should be routed as a 90Ω differential pair. There is no need to match the lengths between pairs, only the signals within a pair need to be length matched, ideally to less than 0.1mm. The P/N signals of the USB 3 signals may be P and N swapped. The USB 2 pairs can't be P/N swapped.

## 2.5. USB 2.0 (High speed)

The USB 2.0 interface supports up to 480Mbps signalling. The differential pair should be routed as a 90Ω differential pair. The length of the P/N signals should ideally be matched to better than 0.15mm.

**TIP**

To enable the USB interface use the `dtoverlay=dwc2,dr_mode=host` setting in the `config.txt` file.

**NOTE**

The port is capable of being used as a true USB On-The-Go (OTG) port. While there is no official documentation, some users have had success making this work. The `USB_OTG_ID` pin is used to select between USB host and device that is typically wired to the ID pin of a Micro USB connector. To use this functionality it must be enabled in the OS. If using either as a fixed slave or fixed master, please tie the `USB_OTG_ID` pin to ground.

## 2.6. GPIO

There are 28 pins available for general purpose I/O (GPIO), which correspond to the GPIO pins on the Raspberry Pi 5 40-pin header. These pins have access to internal peripherals: SMI, DPI, I2C, PWM, SPI, and UART. The RP1 peripherals datasheet describes these features along with the multiplexing options available. The drive strength and slew rate should ideally be set as low as possible to reduce any EMC issues. GPIO2 and GPIO3 have 1.8kΩ pull-up resistors.

The GPIO bank is powered by `GPIO_VREF`, this can either be connected to `CM5_1.8V` for 1.8V signalling GPIO, or `CM5_3.3V` for 3.3V signalling. You should keep the load on the 28 GPIO pins to below 50mA in total. `GPIO_VREF` must be powered for the CM5 to start up correctly. +2.5V signalling is possible if an external +2.5V supply is used for `GPIO_VREF`. This external supply should only be active while `CM5_1.8v` is on, the 2.5V supply will need to be discharged within 1ms of the CM5_1.8v

going low.

### 2.6.1. Alternative function assignments

Up to six alternative functions are available. The table below gives a quick overview.

*Table 1. GPIO function selection*

| | Function | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| GPIO | a0 | a1 | a2 | a3 | a4 | a5 | a6 | a7 | a8 |
| 0 | SPI0_SIO[3] | DPI_PCLK | UART1_TX | I2C0_SDA | | SYS_RIO[0] | PROC_RIO[0] | PIO[0] | SPI2_CSn[0] |
| 1 | SPI0_SIO[2] | DPI_DE | UART1_RX | I2C0_SCL | | SYS_RIO[1] | PROC_RIO[1] | PIO[1] | SPI2_SIO[1] |
| 2 | SPI0_CSn[3] | DPI_VSYNC | UART1_CTS | I2C1_SDA | UART0_IR_RX | SYS_RIO[2] | PROC_RIO[2] | PIO[2] | SPI2_SIO[0] |
| 3 | SPI0_CSn[2] | DPI_HSYNC | UART1_RTS | I2C1_SCL | UART0_IR_TX | SYS_RIO[3] | PROC_RIO[3] | PIO[3] | SPI2_SCLK |
| 4 | GPCLK[0] | DPI_D[0] | UART2_TX | I2C2_SDA | UART0_RI | SYS_RIO[4] | PROC_RIO[4] | PIO[4] | SPI3_CSn[0] |
| 5 | GPCLK[1] | DPI_D[1] | UART2_RX | I2C2_SCL | UART0_DTR | SYS_RIO[5] | PROC_RIO[5] | PIO[5] | SPI3_SIO[1] |
| 6 | GPCLK[2] | DPI_D[2] | UART2_CTS | I2C3_SDA | UART0_DCD | SYS_RIO[6] | PROC_RIO[6] | PIO[6] | SPI3_SIO[0] |
| 7 | SPI0_CSn[1] | DPI_D[3] | UART2_RTS | I2C3_SCL | UART0_DSR | SYS_RIO[7] | PROC_RIO[7] | PIO[7] | SPI3_SCLK |
| 8 | SPI0_CSn[0] | DPI_D[4] | UART3_TX | I2C0_SDA | | SYS_RIO[8] | PROC_RIO[8] | PIO[8] | SPI4_CSn[0] |
| 9 | SPI0_SIO[1] | DPI_D[5] | UART3_RX | I2C0_SCL | | SYS_RIO[9] | PROC_RIO[9] | PIO[9] | SPI4_MISO |
| 10 | SPI0_SIO[0] | DPI_D[6] | UART3_CTS | I2C1_SDA | | SYS_RIO[10] | PROC_RIO[10] | PIO[10] | SPI4_MOSI |
| 11 | SPI0_SCLK | DPI_D[7] | UART3_RTS | I2C1_SCL | | SYS_RIO[11] | PROC_RIO[11] | PIO[11] | SPI4_SCLK |
| 12 | PWM0[0] | DPI_D[8] | UART4_TX | I2C2_SDA | AUDIO_OUT_L | SYS_RIO[12] | PROC_RIO[12] | PIO[12] | SPI5_CSn[0] |
| 13 | PWM0[1] | DPI_D[9] | UART4_RX | I2C2_SCL | AUDIO_OUT_R | SYS_RIO[13] | PROC_RIO[13] | PIO[13] | SPI5_SIO[1] |
| 14 | PWM0[2] | DPI_D[10] | UART4_CTS | I2C3_SDA | UART0_TX | SYS_RIO[14] | PROC_RIO[14] | PIO[14] | SPI5_SIO[0] |
| 15 | PWM0[3] | DPI_D[11] | UART4_RTS | I2C3_SCL | UART0_RX | SYS_RIO[15] | PROC_RIO[15] | PIO[15] | SPI5_SCLK |
| 16 | SPI1_CSn[2] | DPI_D[12] | | | UART0_CTS | SYS_RIO[16] | PROC_RIO[16] | PIO[16] | |
| 17 | SPI1_CSn[1] | DPI_D[13] | | | UART0_RTS | SYS_RIO[17] | PROC_RIO[17] | PIO[17] | |
| 18 | SPI1_CSn[0] | DPI_D[14] | I2S0_SCLK | PWM0[2] | I2S1_SCLK | SYS_RIO[18] | PROC_RIO[18] | PIO[18] | GPCLK[1] |
| 19 | SPI1_SIO[1] | DPI_D[15] | I2S0_WS | PWM0[3] | I2S1_WS | SYS_RIO[19] | PROC_RIO[19] | PIO[19] | |
| 20 | SPI1_SIO[0] | DPI_D[16] | I2S0_SDI[0] | GPCLK[0] | I2S1_SDI[0] | SYS_RIO[20] | PROC_RIO[20] | PIO[20] | |
| 21 | SPI1_SCLK | DPI_D[17] | I2S0_SDO[0] | GPCLK[1] | I2S1_SDO[0] | SYS_RIO[21] | PROC_RIO[21] | PIO[21] | |

| | Function | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 22 | SDIO0_CLK | DPI_D[18] | I2S0_SDI[1] | I2C3_SDA | I2S1_SDI[1] | SYS_RIO[22] | PROC_RIO[22] | PIO[22] | |
| 23 | SDIO0_CMD | DPI_D[19] | I2S0_SDO[1] | I2C3_SCL | I2S1_SDO[1] | SYS_RIO[23] | PROC_RIO[23] | PIO[23] | |
| 24 | SDIO0_DAT[0] | DPI_D[20] | I2S0_SDI[2] | | I2S1_SDI[2] | SYS_RIO[24] | PROC_RIO[24] | PIO[24] | SPI2_CSn[1] |
| 25 | SDIO0_DAT[1] | DPI_D[21] | I2S0_SDO[2] | AUDIO_IN_CLK | I2S1_SDO[2] | SYS_RIO[25] | PROC_RIO[25] | PIO[25] | SPI3_CSn[1] |
| 26 | SDIO0_DAT[2] | DPI_D[22] | I2S0_SDI[3] | AUDIO_IN_DAT0 | I2S1_SDI[3] | SYS_RIO[26] | PROC_RIO[26] | PIO[26] | SPI5_CSn[1] |
| 27 | SDIO0_DAT[3] | DPI_D[23] | I2S0_SDO[3] | AUDIO_IN_DAT1 | I2S1_SDO[3] | SYS_RIO[27] | PROC_RIO[27] | PIO[27] | SPI1_CSn[1] |

Each GPIO can have one function selected at a time. Likewise, each peripheral input (e.g. I2C3_SCL) should only be selected on one GPIO at a time. If the same peripheral input is connected to multiple GPIOs, the peripheral sees the logical OR of these GPIO inputs.

**NOTE**

Function selections without a named function in this list are reserved.

## 2.7. Alternate GPIO functions

- 5× UARTs
  - 4 with 4 wire interface (`TX`, `RX`, `CTS`, `RTS`)
  - UART0 also supporting 8 wire interface (`TX`, `RX`, `CTS`, `RTS`, `DTR`, `DCD`, `DSR`, `RI`) or `IrDA` (`IR_TX`, `IR_RX`)
- 1× SDIO (4 bit)
- 4× PWMs
- 1× I2S, Master (`ISC0`), quadruple lane
- 1× I2S, Slave (`ISC1`), quadruple lane
- 2× `AUDIO_OUT` PWM Audio which requires buffering using a low PSU noise buffer and filtered with an 22KHz 1st order RC network.
- 2× AUDIO_IN : Digital PDM input
- 2× GPCLK Clock outputs
- 1× DPI (Digital Display Interface) (`PCLK`, `DE`, `VSYNC`, `HSYNC` and up to 24bit data)
- 28× GPIO (`SYS_RIO`)
- 4× I2C (`SDA`, `SCL`)
- SPI:

| Instance ID | Master/Slave | Chip-select count | Max I/O width |
|---|---|---|---|
| SPI0 | M | 4 | Quad |
| SPI1 | M | 3 | Dual |
| SPI2 | M | 2 | Dual |
| SPI3 | M | 2 | Dual |
| SPI4 | S | 1 | Single |
| SPI5 | M | 2 | Dual |

For conventional SPI connections on `SIO`[0] is `MOSI` and `SIO`[1] is `MISO`. The other SIO pins aren't required.

## 2.8. Dual HDMI 2.0

CM5 supports two HDMI 2.0 interfaces, each one capable of driving a 4K display.

HDMI signals should be routed as 100Ω differential pairs. Each signal within a pair should ideally be matched to better than 0.15mm. Pairs don't typically need any extra matching, as they only have to be matched to 25mm.

CEC is also supported; an internal 27kΩ pull-up resistor is included in the CM5.

Internal pull-up resistors for the EDID signals is provided. On the Raspberry Pi 5 the HDMI signals don't have any extra

ESD protection. Depending on the application, extra ESD protection may be required.

## 2.9. MIPI (DSI / CSI-2)

The CM5 supports two 4-lane MIPI interfaces. Each MIPI interface can either be DSI (Display) or CSI (Camera). The MIPI signals should be routed as 100Ω differential pairs. Each signal within a pair should ideally be matched to better than 0.15mm.

The documentation around the CSI interface can be found in the Raspberry Pi documentation. CM5 supports DSI displays supported by the official Raspberry Pi firmware or the Linux kernel tree; for other DSI displays, you must provide a custom driver.

**NOTE**

The official Raspberry Pi firmware supports the OmniVision OV5647, Sony IMX219, Sony IMX296, Sony IMX477 and Sony IMX708 camera sensors. No security device is required on Compute Module devices in order to use these camera sensors.

**NOTE**

You can also add displays using the parallel DPI interface, available as a GPIO alternative function. CM5 supports up to three displays of any type (HDMI, DSI, DPI) at any one time.

## 2.10. CAM_GPIO (`CAM_GPIO0` `CAM_GPIO1`)

`CAM_GPIO0` is typically routed to pin 17 on the camera connector to turn the power on and off on a camera module. `CAM_GPIO1` is a new signal not present on previous compute modules. It is recommend to route this to pin 18 on the camera connector for future expansion. These pins are actually GPIO pins on RP1, `CAM_GPIO0` is GPIO34 and `CAM_GPIO1` is GPIO35

## 2.11. I2C (`SDA0` `SCL0`)

This internal I2C bus is normally allocated to MIPI0. It can be used as a general I2C bus if MIPI0 interface isn't being used. SDA0 is connected to GPIO38 on the RP1 and SCL0 is connected to GPIO39.

## 2.12. I2C (`ID_SD` `ID_SC`)

This I2C bus is normally used for identifying HATs and controlling CSI0 and DSI0 devices. If the firmware isn't using the I2C bus e.g. MIPI1 isn't being used then these pins may be used as GPIO 0 and GPIO 1 if required.

**NOTE**

If these pins are used as GPIO pins, then to prevent the firmware from checking to see if there is a HAT EEPROM available, add `force_eeprom_read=0` and `disable_poe_fan=1` to the `config.txt` file.

## 2.13. SDIO/eMMC (CM5Lite only)

CM5Lite does not have onboard eMMC. eMMC signals are available on the connector, enabling the use of an external eMMC or SD card.

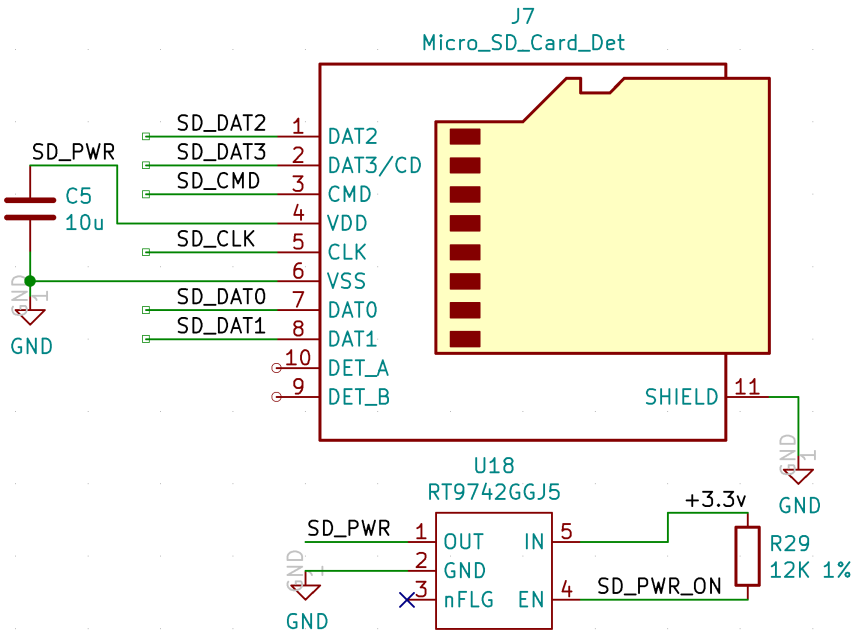
The SD_PWR_ON signal enables an external power switch to power the SD card. To boot from an SD card, fit a pull-up resistor, which turns the power switch on by default.

For eMMC, set SD_VDD_OVERRIDE to high (CM5_3.3V) to force 1.8V signalling on the SDIO interface.

**NOTE**

SD cards require a power switch controlled by SD_PWR_ON as that is the only way to reset the SD card.

*Figure 4. CM5Lite SD card interface.*




## 2.14. USB-C (CC0 CC1)

CC0 and CC1 are signals that are used on a USB-C connector for negotiation of power. CM5 will use these signals to try and negotiate 5V @ 5A.

## 2.15. Fan control (Fan_PWM Fan_Tacho)

CM5 provides two pins for PWM fan control. Fan_PWM is an open collector output suitable for connecting to a wide variety of PWM fans. Fan_Tacho is an input with an internal pull-up to CM5_3.3V suitable for Tacho output from many PWM fans.

**NOTE**

During CM5 shutdown, power supply shutdown includes `Fan_PWM`. If the PWM fan is powered from +5V, the fan will start or continue to run. To prevent the fan from running after power supply shutdown, turn off the supply to the fan. For example, you could use the same supply as the external USB ports use which is controlled by `VBUS_EN`. Alternatively, use an open collector buffer (e.g. 74LVC1G07) powered from +5V. Connect the input to `CM5_3.3V`. Connect the output in parallel with the PWM line.

## 2.16. PMIC_EN

Pulling this pin low puts the CM5 in the lowest possible power-down state.

**TIP**

It is recommended to only pull this pin low once the OS has shut down.

## 2.17. PWR_BUT

Pulling this pin low will either power up the CM5 if it has previously been shut down, or power down the CM5 if held low. This signal is typically connected to a button to function as a power button. If `PWR_EN` is low for longer than 5 seconds the CM5 will be forced shutdown.

## 2.18. VBAT

This is a 2.7 to 3V power input that powers the onboard RTC. While the CM5 is operating, there is a small static load. When the CM5 is not operating, the load increases to power the RTC. A typical CR2032 battery should last for over three years when the CM5 is unpowered.

## 2.19. nRPI_BOOT

During boot if this pin is low, booting from eMMC will be stopped and booting will be transferred to rpi boot which is via USB2.

## 2.20. LED_nACT

This pin is designed to drive an LED to replicate the green LED on the Raspberry Pi 5. Under Linux this pin will flash to signify eMMC access. If any error occurs during booting, then this LED will flash an error pattern which can be decoded using the lookup table in the Raspberry Pi documentation.

## 2.21. LED_nPWR

This pin needs to be buffered to drive an LED. The signal is designed to replicate the red LED on the Raspberry Pi 5. When the board is powered, but shut down, the LED lights up.

## 2.22. EEPROM_nWP

It is recommended that final products pull this pin low to prevent the end users changing the contents of the on-board EEPROM. See the Raspberry Pi 5 documentation for instructions on the software settings required to support EEPROM write protection.

## 2.23. Debug UART

Space is provided for the user to fit a debug UART connector. This connector provides the same functionality as Raspberry Pi 5. The connector is a three-pin 1mm pitch JST-SH connector, Part number `BM03B-SRSS-TB`. The signals are replicated on the bottom as test points.

# Chapter 3. Electrical and mechanical

## 3.1. Mechanical

The CM5 is a compact 40mm × 55mm module. The Module is 4.6mm deep, but when connected the height will be 4.94mm or 7.44mm depending on the stacking height chosen.

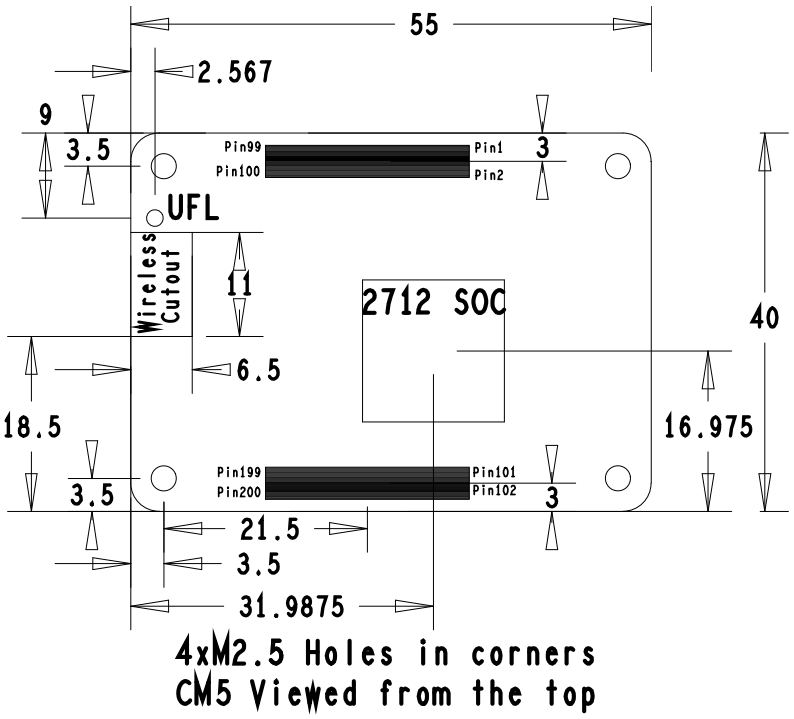
1. 4 × M2.5 mounting holes (inset 3.5mm from module edge)
2. PCB thickness 1.24mm ± 10%
3. BCM2712 SoC height including solder balls 2.2 ± 0.15mm
4. Stacking height either:
    a. 1.5mm with mating connector (clearance under CM5 0mm): Amphenol P/N 10164227-1001A1RLF.
    b. 4.0mm with mating connector (clearance under CM5 2.5mm): Amphenol P/N 10164227-1004A1RLF.

**NOTE**

For additional information see the CM5 design files.

If the onboard wireless antenna is used (see Section 2.1) it must be orientated towards the edge of the plastic enclosure and any nearby metal must have cut-outs or the wireless performance will be degraded. It is suggested that there is at least 10mm clearance around the PCB antenna, but the designer must check the performance.

*Figure 5. Mechanical specification of the Raspberry Pi Compute Module 5*




There must not be any metal, including ground planes, under the antenna. The ground plane cutout must be a minimum of 6.5mm × 11mm, but ideally at least 8mm × 15mm. If these requirements can't be met wireless performance may be degraded, especially in the 2.4GHz spectrum. It is recommended that the external antenna is used where possible.

**NOTE**

The location and arrangement of components on the Compute Module may change slightly over time due to revisions for cost and manufacturing considerations; however the maximum component heights and PCB thickness will be kept as specified.

A step file of the CM5 is available as part of the CM5 design data package. This is for guidance only and is subject to changes over time due to revisions.

## 3.2. Thermal

The CM5 contains less metal in the PCB and fewer connectors than Raspberry Pi 5, which means that it has less passive heat sinking than the Raspberry Pi 5.

The BCM2712 will reduce the clock rate to try and keep its internal temperature below 85°C. So in high ambient temperatures it is possible that the clock will also be automatically throttled back. If the BCM2712 is unable to lower its internal clocks enough to bring the temperature down, its case temperature will rise above 85°C. It is important that any thermal solution chosen keeps the ambient temperature for the other silicon devices on the CM5 within the operating temperature range.

Operating temperature range: -20°C - +85°C non-condensing. NB Optimal RF wireless performance is between -20°C and +75°C.

## 3.3. Electrical specification

**WARNING**

Stresses above those listed in Table 2 may cause permanent damage to the device. This is a stress rating only; functional operation of the device under these or any other conditions above those listed in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

*Table 2. Absolute maximum ratings*

| Symbol | Parameter | Minimum | Maximum | Unit |
|---|---|---|---|---|
| $V_{IN}$ | 5V Input Voltage | -0.5 | 6.0 | V |
| $V_{GPIO\_VREF}$ | GPIO Voltage | -0.5 | 3.6 | V |
| $V_{gpio}$ | GPIO Input voltage | -0.5 | $V_{GPIO\_VREF}$ + 0.5 | V |

**NOTE**

$V_{GPIO\_VREF}$ is the GPIO bank voltage, which must be tied to either the 3.3V or the 1.8V rail of the CM5.

*Table 3. DC characteristics*

| Symbol | Parameter | Conditions | Minimum | Typical | Maximum | Unit |
|---|---|---|---|---|---|---|
| $V_{IL(gpio)}$ | Input low voltage | $V_{GPIO\_VREF}$ = 3.3V | 0 | - | 0.8 | V |
| $V_{IH(gpio)}$ | Input high voltage | $V_{GPIO\_VREF}$ = 3.3V | 2.0 | - | $V_{GPIO\_VREF}$ | V |
| $V_{IL(gpio)}$ | Input low voltage | $V_{GPIO\_VREF}$ = 2.5V | 0 | - | 0.7 | V |

| Symbol | Parameter | Conditions | Minimum | Typical | Maximum | Unit |
|---|---|---|---|---|---|---|
| $V_{IH(gpio)}$ | Input high voltage | $V_{GPIO\_VREF}$ = 2.5V | 1.7 | - | $V_{GPIO\_VREF}$ | V |
| $V_{IL(gpio)}$ | Input low voltage | $V_{GPIO\_VREF}$ = 1.8V | 0 | - | 0.35*$V_{GPIO\_VREF}$ | V |
| $V_{IH(gpio)}$ | Input high voltage | $V_{GPIO\_VREF}$ = 1.8V | 0.65*$V_{GPIO\_VREF}$ | - | $V_{GPIO\_VREF}$ | V |
| $I_{IL(gpio)}$ | Input leakage current | $V_{GPIO\_VREF}$ = 3.3V | - | - | 3 | µA |
| $I_{IL(gpio)}$ | Input leakage current | $V_{GPIO\_VREF}$ = 2.5V | - | - | 5 | µA |
| $I_{IL(gpio)}$ | Input leakage current | $V_{GPIO\_VREF}$ = 1.8V | - | - | 7 | µA |
| $V_{OL(gpio)}$ | Output low voltage | - | - | - | 0.4 | V |
| $V_{OH(gpio)}$ | Output high voltage | $V_{GPIO\_VREF}$ = 3.3V | $V_{GPIO\_VREF}$ - 0.4 | - | - | V |
| $V_{OH(gpio)}$ | Output high voltage | $V_{GPIO\_VREF}$ = 2.5V | $V_{GPIO\_VREF}$ - 0.5 | - | - | V |
| $V_{OH(gpio)}$ | Output high voltage | $V_{GPIO\_VREF}$ = 1.8V | $V_{GPIO\_VREF}$ - 0.4 | - | - | V |
| $I_{OL(gpio)}$ | Output current | 2mA, $V_{GPIO\_VREF}$ = 3.3V | 6.1 | 9.6 | 13.5 | mA |
| $I_{OL(gpio)}$ | Output current | 4mA, $V_{GPIO\_VREF}$ = 3.3V | 9.2 | 14.3 | 20.2 | mA |
| $I_{OL(gpio)}$ | Output current | 8mA, $V_{GPIO\_VREF}$ = 3.3V | 15.3 | 23.9 | 33.7 | mA |
| $I_{OL(gpio)}$ | Output current | 12mA, $V_{GPIO\_VREF}$ = 3.3V | 18.4 | 28.7 | 40.5 | mA |
| $I_{OH(gpio)}$ | Output current | 2mA, $V_{GPIO\_VREF}$ = 3.3V | 4.5 | 6.3 | 8.4 | mA |
| $I_{OH(gpio)}$ | Output current | 4mA, $V_{GPIO\_VREF}$ = 3.3V | 6.8 | 9.5 | 12.6 | mA |
| $I_{OH(gpio)}$ | Output current | 8mA, $V_{GPIO\_VREF}$ = 3.3V | 11.4 | 15.8 | 21 | mA |
| $I_{OH(gpio)}$ | Output current | 12mA, $V_{GPIO\_VREF}$ = 3.3V | 13.6 | 19 | 25.2 | mA |
| $I_{OL(gpio)}$ | Output current | 2mA, $V_{GPIO\_VREF}$ = 2.5V | 4.7 | 8 | 12.2 | mA |
| $I_{OL(gpio)}$ | Output current | 4mA, $V_{GPIO\_VREF}$ = 2.5V | 7.1 | 12 | 18.2 | mA |
| $I_{OL(gpio)}$ | Output current | 8mA, $V_{GPIO\_VREF}$ = 2.5V | 11.8 | 20 | 30.4 | mA |

| Symbol | Parameter | Conditions | Minimum | Typical | Maximum | Unit |
|---|---|---|---|---|---|---|
| $I_{OL(gpio)}$ | Output current | 12mA, $V_{GPIO\_VREF}$ = 2.5V | 14.1 | 24 | 36.4 | mA |
| $I_{OH(gpio)}$ | Output current | 2mA, $V_{GPIO\_VREF}$ = 2.5V | 3.5 | 5.1 | 7 | mA |
| $I_{OH(gpio)}$ | Output current | 4mA, $V_{GPIO\_VREF}$ = 2.5V | 5.2 | 7.6 | 10.5 | mA |
| $I_{OH(gpio)}$ | Output current | 8mA, $V_{GPIO\_VREF}$ = 2.5V | 8.7 | 12.7 | 17.6 | mA |
| $I_{OH(gpio)}$ | Output current | 12mA, $V_{GPIO\_VREF}$ = 2.5V | 10.4 | 15.2 | 21.1 | mA |
| $I_{OL(gpio)}$ | Output current | 2mA, $V_{GPIO\_VREF}$ = 1.8V | 4.4 | 8.1 | 13.6 | mA |
| $I_{OL(gpio)}$ | Output current | 4mA, $V_{GPIO\_VREF}$ = 1.8V | 8.8 | 16.3 | 27.2 | mA |
| $I_{OL(gpio)}$ | Output current | 8mA, $V_{GPIO\_VREF}$ = 1.8V | 11.8 | 21.7 | 36.3 | mA |
| $I_{OL(gpio)}$ | Output current | 12mA, $V_{GPIO\_VREF}$ = 1.8V | 16.2 | 29.2 | 49.9 | mA |
| $I_{OH(gpio)}$ | Output current | 2mA, $V_{GPIO\_VREF}$ = 1.8V | 3.4 | 5.3 | 7.7 | mA |
| $I_{OH(gpio)}$ | Output current | 4mA, $V_{GPIO\_VREF}$ = 1.8V | 6.9 | 10.5 | 15.4 | mA |
| $I_{OH(gpio)}$ | Output current | 8mA, $V_{GPIO\_VREF}$ = 1.8V | 9.1 | 14 | 20.6 | mA |
| $I_{OH(gpio)}$ | Output current | 12mA, $V_{GPIO\_VREF}$ = 1.8V | 12.6 | 19.3 | 28.3 | mA |
| $R_{PU(gpio)}$ | Pull-up resistor | $V_{GPIO\_VREF}$ = 3.3V | 37 | 55 | 86 | kΩ |
| $R_{PD(gpio)}$ | Pull-down resistor | $V_{GPIO\_VREF}$ = 3.3V | 35 | 55 | 98 | kΩ |
| $R_{PU(gpio)}$ | Pull-up resistor | $V_{GPIO\_VREF}$ = 2.5V | 49 | 77 | 123 | kΩ |
| $R_{PD(gpio)}$ | Pull-down resistor | $V_{GPIO\_VREF}$ = 2.5V | 49 | 84 | 155 | kΩ |
| $R_{PU(gpio)}$ | Pull-up resistor | $V_{GPIO\_VREF}$ = 1.8V | 38 | 64 | 106 | kΩ |
| $R_{PD(gpio)}$ | Pull-down resistor | $V_{GPIO\_VREF}$ = 1.8V | 58 | 103 | 189 | kΩ |

Refer to interface specifications (see Chapter 2) for electrical details of other interfaces.

*Table 4. Power consumption*

| Symbol | Parameter | Conditions | Minimum | Typical | Maximum | Unit |
|---|---|---|---|---|---|---|
| $I_{shutdown}$ | Shutdown current | `PMIC_ENABLE` = 0V | - | 1.3 | - | mA |
| $I_{shutdown}$ | Shutdown current | `PMIC_ENABLE` > 2V | - | 3 | - | mA |

| Symbol | Parameter | Conditions | Minimum | Typical | Maximum | Unit |
|---|---|---|---|---|---|---|
| $I_{idle}$ | Idle current | `PMIC_ENABLE` > 2V | - | 400 | - | mA |
| $I_{load}$ | Operation current | `PMIC_ENABLE` > 2V | - | 900 | - | mA |
| $I_{Vbat}$ | RTC current | Vin = +5V | - | 1.7 | - | µA |
| $I_{Vbat}$ | RTC current | Vin = 0V | - | 6 | - | µA |

**NOTE**

The figures in Table 4 greatly depend on the end application.

# Chapter 4. Pinout

*Table 5. Pinout for the Raspberry Pi Compute Module 5*

| Pin | Signal | Description |
|---|---|---|
| 1 | GND | Ground (0V) |
| 2 | GND | Ground (0V) |
| 3 | Ethernet_Pair3_P | Ethernet pair 3 positive (connect to transformer or MagJack) |
| 4 | Ethernet_Pair1_P | Ethernet pair 1 positive (connect to transformer or MagJack) |
| 5 | Ethernet_Pair3_N | Ethernet pair 3 negative (connect to transformer or MagJack) |
| 6 | Ethernet_Pair1_N | Ethernet pair 1 negative (connect to transformer or MagJack) |
| 7 | GND | Ground (0V) |
| 8 | GND | Ground (0V) |
| 9 | Ethernet_Pair2_N | Ethernet pair 2 negative (connect to transformer or MagJack) |
| 10 | Ethernet_Pair0_N | Ethernet pair 0 negative (connect to transformer or MagJack) |
| 11 | Ethernet_Pair2_P | Ethernet pair 2 positive (connect to transformer or MagJack) |
| 12 | Ethernet_Pair0_P | Ethernet pair 0 positive (connect to transformer or MagJack) |
| 13 | GND | Ground (0V) |
| 14 | GND | Ground (0V) |
| 15 | Ethernet_nLED3 | Active-low Ethernet activity indicator (CM5_3.3V signal): typically a green LED is connected to this pin. $I_{OL}$ = 8mA @ $V_{OL}$ < 0.4V |
| 16 | Fan_Tacho | Fan Tacho Input pin internally pulled up with a 1.8K resistor to CM5_3.3V |
| 17 | Ethernet_nLED2 | Active-low Ethernet speed indicator (`CM5_3.3V`signal): typically a yellow LED is connected to this pin. A low state indicates the 1Gbit or 100Mbit link: $I_{OL}$ = 8mA @ $V_{OL}$ < 0.4V |
| 18 | Ethernet_SYNC_OUT | IEEE1588 SYNC Output pin (CM5_3.3V signal: $I_{OL}$ = 8mA @ $V_{OL}$ < 0.4V) |
| 19 | Fan_PWM | ( Open drain output ): |
| 20 | EEPROM_nWP | Leave floating NB internally pulled up to CM5_3.3V via 100kΩ ($V_{IL}$ < 0.8V), but can be grounded to prevent writing to the on-board EEPROM which stores the bootcode |
| 21 | LED_nACT | Active-low Pi activity LED. 20mA Max, 5V tolerant ($V_{OL}$ < 0.4V). (this is the signal that drives the green LED on the Raspberry Pi 5) |
| 22 | GND | Ground (0V) |
| 23 | GND | Ground (0V) |
| 24 | GPIO26 | GPIO: typically a 3.3V signal, but can be a 1.8V signal by connecting GPIO_VREF to CM5_1.8V |
| 25 | GPIO21 | GPIO: typically a 3.3V signal, but can be a 1.8V signal by connecting GPIO_VREF to CM5_1.8V |
| 26 | GPIO19 | GPIO: typically a 3.3V signal, but can be a 1.8V signal by connecting GPIO_VREF to CM5_1.8V |
| 27 | GPIO20 | GPIO: typically a 3.3V signal, but can be a 1.8V signal by connecting GPIO_VREF to CM5_1.8V |
| 28 | GPIO13 | GPIO: typically a 3.3V signal, but can be a 1.8V signal by connecting GPIO_VREF to CM5_1.8V |
| 29 | GPIO16 | GPIO: typically a 3.3V signal, but can be a 1.8V signal by connecting GPIO_VREF to CM5_1.8V |
| 30 | GPIO6 | GPIO: typically a 3.3V signal, but can be a 1.8V signal by connecting GPIO_VREF to CM5_1.8V |

| Pin | Signal | Description |
|---|---|---|
| 31 | GPIO12 | GPIO: typically a 3.3V signal, but can be a 1.8V signal by connecting GPIO_VREF to CM5_1.8V |
| 32 | GND | Ground (0V) |
| 33 | GND | Ground (0V) |
| 34 | GPIO5 | GPIO: typically a 3.3V signal, but can be a 1.8V signal by connecting GPIO_VREF to CM5_1.8V |
| 35 | ID_SC | (BCM2712 GPIO 1) GPIO: typically a 3.3V signal, but can be a 1.8V signal by connecting GPIO_VREF to CM5_1.8V |
| 36 | ID_SD | (BCM2712 GPIO 0) GPIO: typically a 3.3V signal, but can be a 1.8V signal by connecting GPIO_VREF to CM5_1.8V |
| 37 | GPIO7 | GPIO: typically a 3.3V signal, but can be a 1.8V signal by connecting GPIO_VREF to CM5_1.8V |
| 38 | GPIO11 | GPIO: typically a 3.3V signal, but can be a 1.8V signal by connecting GPIO_VREF to CM5_1.8V |
| 39 | GPIO8 | GPIO: typically a 3.3V signal, but can be a 1.8V signal by connecting GPIO_VREF to CM5_1.8V |
| 40 | GPIO9 | GPIO: typically a 3.3V signal, but can be a 1.8V signal by connecting GPIO_VREF to CM5_1.8V |
| 41 | GPIO25 | GPIO: typically a 3.3V signal, but can be a 1.8V signal by connecting GPIO_VREF to CM5_1.8V |
| 42 | GND | Ground (0V) |
| 43 | GND | Ground (0V) |
| 44 | GPIO10 | GPIO: typically a 3.3V signal, but can be a 1.8V signal by connecting GPIO_VREF to CM5_1.8V |
| 45 | GPIO24 | GPIO: typically a 3.3V signal, but can be a 1.8V signal by connecting GPIO_VREF to CM5_1.8V |
| 46 | GPIO22 | GPIO: typically a 3.3V signal, but can be a 1.8V signal by connecting GPIO_VREF to CM5_1.8V |
| 47 | GPIO23 | GPIO: typically a 3.3V signal, but can be a 1.8V signal by connecting GPIO_VREF to CM5_1.8V |
| 48 | GPIO27 | GPIO: typically a 3.3V signal, but can be a 1.8V signal by connecting GPIO_VREF to CM5_1.8V |
| 49 | GPIO18 | GPIO: typically a 3.3V signal, but can be a 1.8V signal by connecting GPIO_VREF to CM5_1.8V |
| 50 | GPIO17 | GPIO: typically a 3.3V signal, but can be a 1.8V signal by connecting GPIO_VREF to CM5_1.8V |
| 51 | GPIO15 | GPIO: typically a 3.3V signal, but can be a 1.8V signal by connecting GPIO_VREF to CM5_1.8V |
| 52 | GND | Ground (0V) |
| 53 | GND | Ground (0V) |
| 54 | GPIO4 | GPIO: typically a 3.3V signal, but can be a 1.8V signal by connecting GPIO_VREF to CM5_1.8V |
| 55 | GPIO14 | GPIO: typically a 3.3V signal, but can be a 1.8V signal by connecting GPIO_VREF to CM5_1.8V |
| 56 | GPIO3 | GPIO: typically a 3.3V signal, but can be a 1.8V signal by connecting GPIO_VREF to CM5_1.8V. Internal 1.8kΩ pull-up to GPIO_VREF |
| 57 | SD_CLK | SD card clock signal (only available on CM5Lite) |
| 58 | GPIO2 | GPIO: typically a 3.3V signal, but can be a 1.8V signal by connecting GPIO_VREF to CM5_1.8V. Internal 1.8kΩ pull-up to GPIO_VREF |
| 59 | GND | Ground (0V) |
| 60 | GND | Ground (0V) |
| 61 | SD_DAT3 | SD card/eMMC Data3 signal (only available on CM5Lite) |
| 62 | SD_CMD | SD card/eMMC Command signal (only available on CM5Lite) |
| 63 | SD_DAT0 | SD card/eMMC Data0 signal (only available on CM5Lite) |

| Pin | Signal | Description |
|---|---|---|
| 64 | SD_DAT5 | SD card/eMMC Data5 signal (only available on CM5Lite) |
| 65 | GND | Ground (0V) |
| 66 | GND | Ground (0V) |
| 67 | SD_DAT1 | SD card/eMMC Data1 signal (only available on CM5Lite) |
| 68 | SD_DAT4 | SD card/eMMC Data4 signal (only available on CM5Lite) |
| 69 | SD_DAT2 | SD card/eMMC Data2 signal (only available on CM5Lite) |
| 70 | SD_DAT7 | SD card/eMMC Data7 signal (only available on CM5Lite) |
| 71 | GND | Ground (0V) |
| 72 | SD_DAT6 | SD card/eMMC Data6 signal (only available on CM5Lite) |
| 73 | SD_VDD_OVERRIDE | Connect to CM5_3.3V to force SD card/eMMC interface to 1.8V signalling instead of 3.3V, otherwise leave unconnected. Typically only used if external eMMC is connected. |
| 74 | GND | Ground (0V) |
| 75 | SD_PWR_ON | Output to power switch for the SD card. The CM5 sets this pin high (3.3V) to signal that power to the SD card should be turned on. Internally Pulled up to CM5_3.3v with a 4.53K (only available on CM5Lite) |
| 76 | VBAT | RTC Battery input 2.5V to 3.5V typically 3V |
| 77 | +5V (Input) | 4.75V-5.25V. Main power input |
| 78 | GPIO_VREF | Must be connected to CM5_3.3V (pins 84 and 86) for 3.3V GPIO0-27 or CM5_1.8V (pins 88 and 90) for 1.8V GPIO0-27. This pin cannot be floating or connected to ground. |
| 79 | +5V (Input) | 4.75V-5.25V. Main power input |
| 80 | SCL0 | I2C clock pin (GPIO39): typically used for Camera and Display. Internal 1.8kΩ pull-up to CM5_3.3V |
| 81 | +5V (Input) | 4.75V-5.25V. Main power input |
| 82 | SDA0 | I2C Data pin (GPIO38): typically used for Camera and Display. Internal 1.8kΩ pull-up to CM5_3.3V |
| 83 | +5V (Input) | 4.75V-5.25V. Main power input |
| 84 | CM5_3.3V (Output) | 3.3V ± 5%. Power Output max 300mA per pin for a total of 600mA. This will be powered down during power-off or GLOBAL_EN being set low |
| 85 | +5V (Input) | 4.75V-5.25V. Main power input |
| 86 | CM5_3.3V (Output) | 3.3V ± 5%. Power Output max 300mA per pin for a total of 600mA. This will be powered down during power-off or GLOBAL_EN being set low |
| 87 | +5V (Input) | 4.75V-5.25V. Main power input |
| 88 | CM5_1.8V (Output) | 1.8V ± 5%. Power Output max 300mA per pin for a total of 600mA. This will be powered down during power-off or GLOBAL_EN being set low |
| 89 | WL_nDisable | Can be left floating; if driven low the wireless interface will be disabled. Internally pulled up via 1.8kΩ to CM5_3.3V |
| 90 | CM5_1.8V (Output) | 1.8V ± 5%. Power Output max 300mA per pin for a total of 600mA. This will be powered down during power-off or GLOBAL_EN being set low |
| 91 | BT_nDisable | Can be left floating; if driven low the Bluetooth interface will be disabled. Internally pulled up via 1.8kΩ to CM5_3.3V |

| Pin | Signal | Description |
|---|---|---|
| 92 | PWR_Button | Pull low to force power off or power on from previous software powered off state. Internally pulled up to +5V via 10kΩ |
| 93 | nRPIBOOT | A low on this pin forces booting from an RPI server (e.g. PC or a Raspberry Pi); if not used leave floating. Internally pulled up via 10kΩ to CM5_3.3V |
| 94 | CC1 | USB PSU PD signal , wire to a USB-C connector to enable 5A @ 5V negotiation. |
| 95 | LED_nPWR | (3.3V signal) Active-low output to drive Power On LED. This signal needs to be buffered. |
| 96 | CC2 | USB PSU PD signal , wire to a USB-C connector to enable 5A @ 5V negotiation. |
| 97 | CAM_GPIO0 | (3.3V signal) Can be a GPIO ( GPIO34) or part of the bus with pin 100 |
| 98 | GND | Ground (0V) |
| 99 | PMIC_Enable | Input. Drive low to power off CM5. Internally pulled up with a 100kΩ to +5V |
| 100 | CAM_GPIO1 | (3.3V signal) (GPIO35) Internally pulled up with 15K to CM5_3.3V |
| 101 | USB_OTG_ID | Input (3.3V signal) USB OTG Pin. Internally pulled up. When grounded the CM5 becomes a USB host but the correct OS driver also needs to be used |
| 102 | PCIe_CLK_nREQ | Input (3.3V signal) PCIe clock request pin (low to request PCI clock). Internally pulled up |
| 103 | USB_N | USB2.0 D- |
| 104 | PCIE_nWAKE | (3.3V signal) PCIe WAKE# signal can be left unconnected if wake up isn't required, internally pulled up. |
| 105 | USB_P | USB2.0 D+ |
| 106 | PCIE_PWR_EN | (3.3V signal) Active high, used to signal that a PCIe device can be powered down when low. |
| 107 | GND | Ground (0V) |
| 108 | GND | Ground (0V) |
| 109 | PCIe_nRST | Output (3.3V signal) PCIe reset active-low |
| 110 | PCIe_CLK_P | PCIe clock Out positive (100MHz) |
| 111 | VBUS_EN | (3.3V signal) Active high to signal USB3 Ports should be powered |
| 112 | PCIe_CLK_N | PCIe clock Out negative (100MHz) |
| 113 | GND | Ground (0V) |
| 114 | GND | Ground (0V) |
| 115 | MIPI0_D0_N | MIPI0 D0 negative |
| 116 | PCIe_RX_P | Input PCIe GEN 2 RX positive NB external AC coupling capacitor required |
| 117 | MIPI0_D0_P | MIPI0 D0 positive |
| 118 | PCIe_RX_N | Input PCIe GEN 2 RX negative NB external AC coupling capacitor required |
| 119 | GND | Ground (0V) |
| 120 | GND | Ground (0V) |
| 121 | MIPI0_D1_N | MIPI0 D1 negative |
| 122 | PCIe_TX_P | Output PCIe GEN 2 TX positive NB AC coupling capacitor included on CM5 |
| 123 | MIPI0_D1_P | MIPI0 D1 positive |

| Pin | Signal | Description |
|---|---|---|
| 124 | PCIe_TX_N | Output PCIe GEN 2 TX positive NB AC coupling capacitor included on CM5 |
| 125 | GND | Ground (0V) |
| 126 | GND | Ground (0V) |
| 127 | MIPI0_C_N | MIPI0 clock negative |
| 128 | USB3-0-RX_N | USB3 Port 0 RX Input negative |
| 129 | MIPI0_C_P | MIPI0 clock positive |
| 130 | USB3-0-RX_P | USB3 Port 0 RX Input positive |
| 131 | GND | Ground (0V) |
| 132 | GND | Ground (0V) |
| 133 | MIPI0_D2_N | MIPI0 D2 negative |
| 134 | USB3-0-DP | USB3 port 0 DP |
| 135 | MIPI0_D2_P | MIPI0 D2 positive |
| 136 | USB3-0-DM | USB3 Port 0 DM |
| 137 | GND | Ground (0V) |
| 138 | GND | Ground (0V) |
| 139 | MIPI0_D3_N | MIPI0 D3 negative |
| 140 | USB3-0-TX_N | USB3 Port 0 TX output negative NB AC coupling capacitor included on CM5 |
| 141 | MIPI0_D3_P | MIPI0 D3 positive |
| 142 | USB3-0-TX_P | USB3 Port 0 TX output positive NB AC coupling capacitor included on CM5 |
| 143 | HDMI1_HOTPLUG | Input HDMI1 hotplug. Internally pulled down with a 100kΩ. 5V tolerant.(It can be connected directly to a HDMI connector) |
| 144 | GND | Ground (0V) |
| 145 | HDMI1_SDA | Bidirectional HDMI1 SDA. Internally pulled up with a 1.8kΩ. 5V tolerant. (It can be connected directly to a HDMI connector) |
| 146 | HDMI1_TX2_P | Output HDMI1 TX2 positive |
| 147 | HDMI1_SCL | Bidirectional HDMI1 SCL. Internally pulled up with a 1.8kΩ. 5V tolerant. (It can be connected directly to a HDMI connector) |
| 148 | HDMI1_TX2_N | Output HDMI1 TX2 negative |
| 149 | HDMI1_CEC | Input HDMI1 CEC. Internally pulled up with a 27kΩ. 5V tolerant. (It can be connected directly to a HDMI connector) |
| 150 | GND | Ground (0V) |
| 151 | HDMI0_CEC | Input HDMI0 CEC. Internally pulled up with a 27kΩ. 5V tolerant (It can be connected directly to a HDMI connector) |
| 152 | HDMI1_TX1_P | Output HDMI1 TX1 positive |
| 153 | HDMI0_HOTPLUG | Input HDMI0 hotplug. Internally pulled down 100kΩ. 5V tolerant. (It can be connected directly to a HDMI connector) |
| 154 | HDMI1_TX1_N | Output HDMI1 TX1 negative |
| 155 | GND | Ground (0V) |

| Pin | Signal | Description |
|---|---|---|
| 156 | GND | Ground (0V) |
| 157 | USB3-1-RX_N | USB3 Port 1 RX Input negative |
| 158 | HDMI1_TX0_P | Output HDMI1 TX0 positive |
| 159 | USB3-1-RX_P | USB3 Port 1 RX Input positive |
| 160 | HDMI1_TX0_N | Output HDMI1 TX0 negative |
| 161 | GND | Ground (0V) |
| 162 | GND | Ground (0V) |
| 163 | USB3-1-DP | USB3 port 1 DP |
| 164 | HDMI1_CLK_P | Output HDMI1 clock positive |
| 165 | USB3-1-DM | USB3 Port 1 DM |
| 166 | HDMI1_CLK_N | Output HDMI1 clock negative |
| 167 | GND | Ground (0V) |
| 168 | GND | Ground (0V) |
| 169 | USB3-1-TX_N | USB3 Port 1 TX output negative NB AC coupling capacitor included on CM5 |
| 170 | HDMI0_TX2_P | Output HDMI0 TX2 positive |
| 171 | USB3-1-TX_P | USB3 Port 1 TX output positive NB AC coupling capacitor included on CM5 |
| 172 | HDMI0_TX2_N | Output HDMI0 TX2 negative |
| 173 | GND | Ground (0V) |
| 174 | GND | Ground (0V) |
| 175 | MIPI1_D0_N | MIPI1 D0 negative |
| 176 | HDMI0_TX1_P | Output HDMI0 TX1 positive |
| 177 | MIPI1_D0_P | MIPI1 D0 positive |
| 178 | HDMI0_TX1_N | Output HDMI0 TX1 negative |
| 179 | GND | Ground (0V) |
| 180 | GND | Ground (0V) |
| 181 | MIPI1_D1_N | MIPI1 D1 negative |
| 182 | HDMI0_TX0_P | Output HDMI0 TX0 positive |
| 183 | MIPI1_D1_P | MIPI1 D1 positive |
| 184 | HDMI0_TX0_N | Output HDMI0 TX0 negative |
| 185 | GND | Ground (0V) |
| 186 | GND | Ground (0V) |
| 187 | MIPI1_C_N | MIPI1 clock negative |
| 188 | HDMI0_CLK_P | Output HDMI0 clock positive |
| 189 | MIPI1_C_P | MIPI1 clock positive |
| 190 | HDMI0_CLK_N | Output HDMI0 clock negative |
| 191 | GND | Ground (0V) |

| Pin | Signal | Description |
|---|---|---|
| 192 | GND | Ground (0V) |
| 193 | MIPI1_D2_N | MIPI1 D2 negative |
| 194 | MIPI1_D3_N | MIPI1 D3 negative |
| 195 | MIPI1_D2_P | MIPI1 D2 positive |
| 196 | MIPI1_D3_P | MIPI1 D3 positive |
| 197 | GND | Ground (0V) |
| 198 | GND | Ground (0V) |
| 199 | HDMI0_SDA | Bidirectional HDMI0 SDA. Internally pulled up with a 1.8kΩ. 5V tolerant. (It can be connected directly to a HDMI connector; a small amount of ESD protection is provided on the CM5 by an on-board HDMI05-CL02F3) |
| 200 | HDMI0_SCL | Bidirectional HDMI0 SCL. Internally pulled up with a 1.8kΩ. 5V tolerant. (It can be connected directly to a HDMI connector; a small amount of ESD protection is provided on the CM5 by an on-board HDMI05-CL02F3) |

All ground pins should be connected. If none of the signals on the second connector (pins 101 to 200) are used, then you may omit the connector to reduce costs, but mechanical stability needs to be considered.

The voltage on GPIO pins 0-27 must not exceed CM5_3.3V if 3.3V signalling is used or CM5_1.8V if 1.8V signalling is used. These pins are the same as on the 40-pin connector on the Raspberry Pi 5.

If the CM5_1.8V rail is used to power other devices other than the GPIO_VREF then you should ensure that in case of surprise power removal (e.g.the +5V pin goes below +4.5V) from the CM5, the load on the CM5_1.8V must go to zero.

Similarly if the CM5_3.3V rail is used to power other devices other than the GPIO_VREF, then you should ensure that in the case of surprise power removal the CM5_3.3V rail never falls below the CM5_1.8V rail. This is the typical case, but you should check this in your design. In the case where it does fall below the CM5_1.8V rail, then extra circuitry is required to disconnect the CM5_3.3V load.

No reverse voltage must be applied to any pin, or power-up may be prevented; i.e. during power-down/off no pin may have external voltage applied, otherwise this may prevent a subsequent power-up.

## 4.1. Differential pairs

It is recommended that P/N signals within a pair are matched to better than 0.15mm. Often, matching between pairs is not so critical: e.g. HDMI pair-to-pair matching should be better than 25mm, so on a typical board no extra matching is required.

### 4.1.1. 100Ω differential pair signal lengths

On the CM5 all differential pairs are matched to better than 0.05mm (P/N signals).

**NOTE**

It is recommended that pairs are also matched on the interface board.

On the CM5, pair-to-pairs are not always matched, as many interfaces do not require very accurate matching between pairs. Table 6 documents the CM5 track-length difference within each group. (A non-zero value represents how much longer in mm that track is, when compared to the signal with zero length difference.)

*Table 6. 100Ω differential pair signal lengths*

| Signal | Length |
|---|---|
| MIPI0_C_N | 0.78 |
| MIPI0_C_P | 0.78 |
| MIPI0_D0_N | 0.01 |
| MIPI0_D0_P | 0.02 |
| MIPI0_D1_N | 0.4 |
| MIPI0_D1_P | 0.4 |
| MIPI0_D2_N | 0.04 |
| MIPI0_D2_P | 0.03 |
| MIPI0_D3_N | 0.01 |
| MIPI0_D3_P | 0 |
| | |
| MIPI1_C_N | 1.28 |
| MIPI1_C_P | 1.27 |
| MIPI1_D0_N | 0 |
| MIPI1_D0_P | 0 |
| MIPI1_D1_N | 1.06 |
| MIPI1_D1_P | 1.05 |
| MIPI1_D2_N | 0.83 |
| MIPI1_D2_P | 0.84 |
| MIPI1_D3_N | 3.79 |
| MIPI1_D3_P | 3.79 |
| | |
| HDMI0_CLK_N | 0.91 |
| HDMI0_CLK_P | 0.91 |
| HDMI0_TX0_N | 0.18 |
| HDMI0_TX0_P | 0.18 |
| HDMI0_TX1_N | 0 |
| HDMI0_TX1_P | 0 |
| HDMI0_TX2_N | 0.25 |
| HDMI0_TX2_P | 0.25 |
| | |
| HDMI1_CLK_N | 2.99 |
| HDMI1_CLK_P | 2.99 |
| HDMI1_TX0_N | 4.76 |
| HDMI1_TX0_P | 4.75 |
| HDMI1_TX1_N | 5.18 |
| HDMI1_TX1_P | 5.18 |

| | |
|---|---|
| HDMI1_TX2_N | 0 |
| HDMI1_TX2_P | 0 |
| | |
| Ethernet_Pair0_P | 2.92 |
| Ethernet_Pair0_N | 2.93 |
| Ethernet_Pair1_P | 0 |
| Ethernet_Pair1_N | 0 |
| Ethernet_Pair2_P | 0.59 |
| Ethernet_Pair2_N | 0.60 |
| Ethernet_Pair3_P | 0.38 |
| Ethernet_Pair3_N | 0.38 |

### 4.1.2. 90Ω differential pair signal lengths

On the CM5 all 90Ω differential pairs (including USB pairs) are matched to better than 0.01mm (P/N signals). USB 3 pairs don't require any pair to pair matching with in a port group.

**NOTE**

It is recommended that pairs are also matched on the interface board.

Pair-to-pairs aren't always matched as many interfaces don't require very accurate matching between pairs. Table 7 documents the CM5 track-length difference within each group. (A non-zero value represents how much longer in mm that track is, when compared to the signal with zero length difference.)

*Table 7. 90Ω differential pair signal lengths*

| Signal | Length |
|---|---|
| PCIe_CLK_P | 0.00 |
| PCIe_CLK_N | 0.01 |
| PCIe_TX_P | 3.71 |
| PCIe_TX_N | 3.72 |
| PCIe_RX_P | 0.84 |
| PCIe_RX_N | 0.84 |

# Chapter 5. Power

## 5.1. Power-up sequencing

The CM5 requires a single +5V supply, and can supply up to 600mA at +3.3V and +1.8V to peripherals.

All pins should not have any power applied to them before the +5V rail is applied.

If the on board boot EEPROM is to be write-protected, then the `EEPROM_nWP` should be low before power-up.

If the CM5 is to be booted using USB then `RPI_nBOOT` needs to be low within 2ms of +5V rising.

+5V should rise monotonically to 4.75V and stay above 4.75V for the entire operation of the CM5.

The power-up sequence will start when both +5V rail is above 4.75V and `PMIC_EN` rises. The order of events is as follows

1. +5V rises
2. `PMIC_ENABLE` rises
3. `CM5_+3.3V` rises
4. `CM5_+1.8V` rises at least 1ms after `CM5_+3.3V`

## 5.2. Power-down sequencing

The operating system should be shut down before power is removed to ensure file system consistency. If this can't be achieved, then a filesystem like `btrfs`, `f2fs` or `overlayfs` (use `raspi-config` to enable this) should be considered.

Once the operating system has shut down, the +5V rail can be removed or the `PMIC_EN` pin can be taken low to put the CM5 into the lowest power mode.

During the shutdown sequence `CM5_+1.8V` will be discharged before the `CM5_+3.3V` rail.

## 5.3. Power consumption

The exact power consumption of the CM5 will greatly depend on the tasks being run on the CM5. The lowest shutdown power consumption mode is with the `PMIC_EN` driven low, typically is 1.3µA. With `PMIC_EN` high but software shut down, the typical consumption is 3mA. Idle power consumption is typically 400mA, but this varies considerably depending on the operating system. Operating power consumption is typically around 900mA; again, this greatly depends on the operating system and the tasks being executed.

## 5.4. Regulator outputs

To make it easier to interface to the CM5 the on-board regulators (`CM5_+3.3V` and `CM5_+1.8V`) can each supply 600mA to devices connected to the CM5. The loads on these outputs isn't taken into account in the power consumption figures.

# Appendix A: Troubleshooting

The CM5 has a number of stages of power-up before the CPU starts. If there is an error at any of the stages, power-up will be halted.

## Hardware checklist

1. Is the +5V supply good? Check this by pulling `PMIC_EN` low and apply an external 2A load to the +5V supply. Does it stay > +4.75V including noise? Ideally, it should remain > +4.9V including any noise.
2. Remove external 2A load, but keep `PMIC_EN` pulled low.
3. Check the CM5 +3.3V rail is < 200mV. If this is not the case there is an external power path back-feeding the CM5, either directly or indirectly. This could also occur via the digital pins, e.g Ethernet.
4. Still with `PMIC_EN` pulled low check the CM5 +1.8V rail is < 200mV. Again if the +1.8V rail is above 200mV then there is an external path back-feeding the 1.8V rail. If nothing is connected to these pins, you can ignore this check.
5. Remove the pull-down on `PMIC_EN`.
6. Check `PMIC_EN` now goes high (it is internally pulled up on the CM5).
7. Check the +3.3V supply rises to > +3.15V. If it does not, this suggests there is too much load on the +3.3V rail.
8. Check the +1.8V rail gets to > +1.71V. If it does not, this suggests there is too much load on the +1.8V rail.
9. Check `ACT_LED` starts to oscillate to indicate booting; check it isn't flashing an error code.

## Bootloader

1. Connect a HDMI cable to see if the HDMI diagnostics screen appears.
2. Connect a USB serial cable to GPIO pins 14 and 15. For more information, see Configuring UARTs.
3. Short the `nRPIBOOT` pin to ground to force USB boot mode. The CM5IO board has a jumper for `nRPIBOOT` This can be used to enable different boot modes (e.g. network) and enable UART logging. For more information, see Flashing the Compute Module eMMC.

## rpi-eeprom-update

CM5 will not run `recovery.bin` from the eMMC (or SD card on CM5Lite). Therefore, the only way to update the bootloader EEPROM is via `usbboot` or self-update.

## EEPROM write-protect

The on-board EEPROM can be write-protected by shorting `EEPROM_nWP` to ground. The CM5IO board has a jumper for `EEPROM_nWP`.

For more information, see Raspberry Pi boot EEPROM.

## Firmware

Ensure you have the latest version of the firmware, as newer versions contain improvements to the system.

## Kernel

Ensure you have the latest version of the kernel, as newer versions often contain security fixes and improvements.

# Appendix B: Test Points

## Test Points

CM5 contains a number of test points on the board. These can be used to power and program a CM5 without using the main 100pin connectors. Most of the signals replicate pins on the 100 pin connectors.

Table of test point coordinates and mounting hole locations. The coordinates are looking though the CM5.

*Table 8. Test point table for the Raspberry Pi Compute Module 5*

| Reference | X | Y | NAME |
|---|---|---|---|
| MH4 | 51.5 | 36.5 | Mounting Hole |
| MH3 | 3.5 | 36.5 | Mounting Hole |
| MH2 | 51.5 | 3.5 | Mounting Hole |
| MH1 | 3.5 | 3.5 | Mounting Hole |
| TP1 | 14.34 | 17.54 | +5V |
| TP2 | 8.8 | 1.3 | RUN |
| TP3 | 51.2 | 32.6 | GND |
| TP4 | 4.8 | 13 | reserved |
| TP7 | 24.2 | 7.5 | GND |
| TP8 | 1.65 | 15.05 | GND |
| TP9 | 1.5 | 10.5 | reserved |
| TP10 | 48.4 | 15.1 | reserved |
| TP13 | 42.6 | 7.3 | GND |
| TP15 | 14.7 | 6.6 | reserved |
| TP16 | 9.3 | 34.9 | nRPIBOOT |
| TP17 | 37.4 | 8.1 | reserved |
| TP18 | 23.4 | 23.55 | reserved |
| TP21 | 24.5125 | 14.025 | nRESET_OUT |
| TP22 | 13.0875 | 11.225 | reserved |
| TP26 | 17.7 | 20.2 | GND |
| TP27 | 43.6 | 22.3 | reserved |
| TP28 | 15.4 | 16 | reserved |
| TP29 | 23.65 | 21.55 | reserved |
| TP30 | 37.2 | 34.9 | reserved |
| TP31 | 9.1 | 3.2 | reserved |
| TP32 | 1.5 | 13 | reserved |
| TP33 | 47 | 36 | CM5_3V3 |
| TP34 | 50.5 | 15.5 | CM5_1V8 |

| Reference | X | Y | NAME |
|---|---|---|---|
| TP35 | 11 | 37.8 | DEBUG_UART_TX |
| TP36 | 8.5 | 37.1 | DEBUG_UART_RX |
| TP39 | 22.1 | 6.1 | reserved |
| TP40 | 6.7 | 15.2 | reserved |
| TP41 | 8.7 | 15.3 | reserved |
| TP42 | 11.4 | 34.9 | PWR_BUT |
| TP44 | 51.7 | 30.2 | reserved |
| TP45 | 53.1 | 28.7 | reserved |
| TP46 | 7 | 34.7 | GND |
| TP48 | 21.6 | 15.4 | SOC_TRST_N |
| TP49 | 21.6 | 13.3 | SOC_TDI |
| TP50 | 20.4 | 17.2 | SOC_TDO |
| TP51 | 20.3 | 8.8 | SOC_TMS |
| TP52 | 19.9 | 11.9 | SOC_TCK |
| TP57 | 53.2 | 32 | reserved |
| TP60 | 48 | 38.7 | GND |
| TP61 | 6.575 | 1.225 | GND |
| TP62 | 22.2 | 31.6 | GND |
| TP63 | 8.7 | 18.2 | 5v_Sense |
| TP64 | 47.3 | 5.4 | reserved |
| TP65 | 28.2 | 7.5 | USBC_D_N |
| TP66 | 26.1 | 7.5 | USBC_D_P |
| TP67 | 7 | 38.6 | LED_nPWR |
| TP68 | 13 | 37.5 | LED_nACT |
| TP69 | 38.8 | 25.9 | ETH0_P |
| TP70 | 39.6 | 24.2 | ETH0_N |
| TP71 | 43.8 | 14.1 | ETH1_N |
| TP72 | 45.6 | 13.1 | ETH1_P |
| TP73 | 42.4 | 31.7 | ETH2_P |
| TP74 | 42.6 | 33.7 | ETH2_N |
| TP75 | 41.6 | 37.8 | ETH3_P |
| TP76 | 42.9 | 36.1 | ETH3_N |
| TP77 | 45 | 37 | GPIO_VREF |
| TP78 | 14.37 | 19.52 | +5V |

## Powering Vin

Both Vin test points (TP1 and TP78) should be used with a +5V supply. Ground must be connected to at least TP26, TP61, and TP8. Other ground pins should be also used if possible.

## Debug UART

TP35 and TP36 are a TX and RX of the debug UART. TP46 should be used as the ground. It is very useful to have access to these pins during programming and initial boot.

## RPI Boot

TP65 and TP66 are the USB pins. The TP7 GND pin must be used. TP16 nRPI_BOOT should also be grounded to force RPI boot mode.

## Ethernet Boot

TP69-TP70 should be connected to an external MagJack.

# Appendix C: Differences between CM4 and CM5

This section describes the differences between Raspberry Pi Compute Module 5 (CM5) and the previous Raspberry Pi Compute Module 4 (CM4).

## Pinout changes

The connectors have changed brand and have been tested to higher currents to support CM5.

CAM1 signals become dual-purpose and can be used for either a CSI camera or a DSI display.

DSI1 signals become dual-purpose and can be used for either a CSI camera or a DSI display.

The CM4 has extra ESD protection on the HDMI, SDA, SCL, HPD and CEC signals. This is removed from the Raspberry Pi Compute Module 5.

The port labelled `CAM0` on CM4 now is a USB 3.0 Port. The port labelled `DSI0` now is a USB 3.0 port. The original CM4 `VDAC_COMP` pin is now a VBUS enable pin for the two USB 3 ports and is active high.

The 2 ADC channels on CM4 have now become the PD CC signals.

| Pin | CM4 | CM5 | Comment |
|---|---|---|---|
| 16 | SYNC_IN | Fan_tacho | Fan tacho input |
| 19 | Ethernet nLED1 | Fan_PWM | Fan PWM output |
| 76 | Reserved | VBAT | RTC battery. Note, there will be a constant load of a few uA even if the CM5 is powered. |
| 92 | RUN_PG | PWR_Button | Replicates the power button on Raspberry Pi 5. A short press signals that the device should wake up or shut down. A long press forces shutdown. |
| 93 | nRPIBOOT | nRPIBOOT | |
| 94 | AnalogIP1 | CC1 | This pin can connect to the CC1 line of a USB-C connector to enable the PMIC to negotiate 5A. |
| 96 | AnalogIP0 | CC2 | This pin can connect to the CC2 line of a USB-C connector to enable the PMIC to negotiate 5A. |
| 99 | Global_EN | PMIC_ENABLE | No external change. |

| Pin | CM4 | CM5 | Comment |
|---|---|---|---|
| 100 | nEXTRST | CAM_GPIO1 | Pulled up on Raspberry Pi Compute Module 5, but is driven low during boot to emulate a nRESET signal |
| 104 | Reserved | PCIE_DET_nWAKE | PCIE nWAKE. Pull-up to CM5_3v3 with an 8.2K resistor. |
| 106 | Reserved | PCIE_PWR_EN | Signals if the PCIe device can be powered up or down. Active high. |
| 111 | VDAC_COMP | VBUS_EN | Output to signal USB VBUS should be enabled. |
| 128 | CAM0_D0_N | USB3-0-RX_N | May be P/N swapped. |
| 130 | CAM0_D0_P | USB3-0-RX_P | May be P/N swapped. |
| 134 | CAM0_D1_N | USB3-0-DP | USB 2 signal. |
| 136 | CAM0_D1_P | USB3-0-DM | USB 2 signal. |
| 140 | CAM0_C_N | USB3-0-TX_N | May be P/N swapped. |
| 142 | CAM0_C_P | USB3-0-TX_P | May be P/N swapped. |
| 157 | DSI0_D0_N | USB3-1-RX_N | May be P/N swapped. |
| 159 | DSI0_D0_P | USB3-1-RX_P | May be P/N swapped. |
| 163 | DSI0_D1_N | USB3-1-DP | USB 2 signal. |
| 165 | DSI0_D1_P | USB3-1-DM | USB 2 signal. |
| 169 | DSI0_C_N | USB3-1-TX_N | May be P/N swapped. |
| 171 | DSI0_C_P | USB3-1-TX_P | May be P/N swapped. |

In addition to the above, the PCIe CLK signals are no longer capacitively coupled.

The PCB is 0.04mm thicker, but the main processor is thinner

## Track lengths

HDMI0 track lengths have changed. Each P/N pair remains matched, but the skew between pairs is now <1mm. For existing motherboards, this is unlikely to make a difference as the skew between pairs can be in the order of 25mm.

HDMI1 track lengths have also changed. Each P/N pair remains matched, but the skew between pairs is now <5mm, For existing motherboards, this is unlikely to make a difference as the skew between pairs can be in the order of 25mm.

Ethernet track lengths have changed. Each P/N pair remains matched, but the skew between pairs is now <4mm. For existing motherboards, this is unlike to make a difference as the skew between pairs can be in the order of 12mm.

## Power budget

As CM5 is significantly more powerful than CM4, it consumes more power. Power supply designs should budget for 5V up to 2.5A. If this creates an issue with an existing motherboard design, reduce the CPU clock rate to reduce the peak

power consumption.

# Appendix D: Availability

## Support

For documentation please see the Compute Module Hardware documentation section of the raspberrypi.com. Support questions can be posted to the Raspberry Pi forum.

## Order codes

*Table 9. Part number options*

| Model | Wireless | RAM LPDDR4x | eMMC Storage |
|---|---|---|---|
| CM5 | 0 = No | 01 = 1GB | 000 = 0GB (Lite) |
| | 1 = Yes | 02 = 2GB | 008 = 8GB |
| | | 04 = 4GB | 016 = 16GB |
| | | 08 = 8GB | 032 = 32GB |
| | | | 064 = 64GB |
| | | | 128 = 128GB |
| **Example Part Number** | | | |
| CM5 | 1 | 02 | 032 |

*Table 10. Ordering options*

| Wireless | RAM LPDDR4x | Storage eMMC | RPL # | Part Number | Order Multiple | RRP |
|---|---|---|---|---|---|---|
| - | 2GB | Lite | SC1156 | CM5002000 | 1+ / Bulk | |
| - | 2GB | 16GB | SC1158 | CM5002016 | 1+ / Bulk | |
| - | 2GB | 32GB | SC1159 | CM5002032 | 1+ / Bulk | |
| Yes | 2GB | Lite | SC1586 | CM5102000 | 1+ / Bulk | |
| Yes | 2GB | 16GB | sC1588 | CM5102016 | 1+ / Bulk | |
| Yes | 2GB | 32GB | SC1589 | CM5102032 | 1+ / Bulk | |
| - | 4GB | Lite | SC1562 | CM5004000 | 1+ / Bulk | |
| - | 4GB | 16GB | SC1564 | CM5004016 | 1+ / Bulk | |
| - | 4GB | 32GB | SC1565 | CM5004032 | 1+ / Bulk | |
| Yes | 4GB | Lite | SC1592 | CM5104000 | 1+ / Bulk | |
| Yes | 4GB | 16GB | SC1594 | CM5104016 | 1+ / Bulk | |
| Yes | 4GB | 32GB | SC1595 | CM5104032 | 1+ / Bulk | |
| - | 8GB | Lite | SC1568 | CM5008000 | 1+ / Bulk | |
| - | 8GB | 16GB | SC1570 | CM5008016 | 1+ / Bulk | |
| - | 8GB | 32GB | SC1571 | CM5008032 | 1+ / Bulk | |
| Yes | 8GB | Lite | SC1598 | CM5108000 | 1+ / Bulk | |

| Wireless | RAM LPDDR4x | Storage eMMC | RPL # | Part Number | Order Multiple | RRP |
|---|---|---|---|---|---|---|
| Yes | 8GB | 16GB | SC1600 | CM5108016 | 1+ / Bulk | |
| Yes | 8GB | 32GB | SC1601 | CM5108032 | 1+ / Bulk | |

**NOTE**

Other options are available as a custom order

# Packaging

Small quantities are supplied in individual cardboard boxes. These have an internal ESD coating so that a separate ESD bag isn't required. This packaging is recyclable and reduces waste.

# Appendix E: Mean Time Between Failure (MTBF)

*Table 11. Mean time between failure for Raspberry Pi Compute Module 5*

| Model | Mean Time Between Failure Ground Benign (Hours) | Mean Time Between Failure Ground Mobile (Hours) |
|---|---|---|
| CM5 | 143 000 | 16 000 |
| CM5Lite | 168 000 | 16 000 |

**Ground, benign**

Applies to non-mobile, temperature and humidity controlled environments readily accessible to maintenance; includes laboratory instruments and test equipment, medical electronic equipment, business and scientific computer complexes.

**Ground, mobile**

Assumes levels of operational stress well above normal domestic or light industrial use, without temperature, humidity or vibration control: applies to equipment installed on wheeled or tracked vehicles and equipment manually transported; includes mobile and handheld communications equipment.

# Appendix H: Documentation Release History

## 27 November 2024

- Initial release.

Raspberry Pi