

# Raspberry Pi处理器模块5

作者：Raspberry Pi Ltd

译者：上海晶珩电子科技有限公司

# 前言

© 2024 Raspberry Pi Ltd

有关 Raspberry Pi 处理器模块 5 的文档已根据 Creative Commons 获得许可 [-Attribution-NoDerivatives 4.0 International](#) (CC BY-ND)。

日期: 2024-05-23

版本: githash: b7c9aa6-clean

## 法律免责声明

RASPBERRY PI 产品的技术和可靠性数据 (包括 Datasheet) (下文称“资源”) 的修改由 RASPBERRY PI LTD (“RPL”) 按“原样”提供, 拒绝任何明示或暗示的保证, 包括但不限于对适销性和特定用途适用性的暗示保证。在适用法律允许的最大范围内, RPL 不对任何直接的、间接的、附带的、特殊的、惩戒性的或后果性的损害负责 (包括但不限于购买替代商品或服务; 用途、数据或利润的损失; 或业务中断) 以及任何责任理论, 无论是合同责任、严格责任还是因使用资源而产生的侵权行为 (包括疏忽或其他), 即使已被告知此类损害的可能性。

RPL 保留随时对资源或其中描述的任何产品进行任何增强、改进、更正或任何其他修改的权利, 恕不另行通知。

这些资源面向具有适当设计知识水平的熟练用户。用户自行负责选择和使用资源以及资源中描述的产品的任何应用。用户同意保障 RPL 免于承担因其使用资源而产生的所有责任、费用、损害或其他损失。

RPL 允许用户仅将资源与 Raspberry Pi 产品结合使用。禁止将资源用于其他任何用途。没有向任何其他 RPL 或其他第三方知识产权授予许可。

**高风险活动。** Raspberry Pi 产品的设计、制造或设计目的不是用于需要故障安全性能的危险环境, 例如核设施、飞机导航或通信系统、空中交通管制、武器系统或安全关键应用 (包括生命支持系统和其他医疗设备), 在这些环境中, 产品故障可能会直接导致死亡、人身伤害或严重的身体或环境损害 (“**高风险活动**”)。RPL 明确否认对高风险活动适用性的任何明示或暗示担保, 并且不接受在高风险活动中使用或包含 Raspberry Pi 产品的任何责任。

Raspberry Pi 产品根据 RPL 的 [标准条款](#) 提供。RPL 提供的资源不会扩大或以其他方式改变 RPL 的资源 [标准条款](#) 包括但不限于其中表达的免责声明和保证。

## 译者免责声明

本文档由**上海晶珩电子科技有限公司**翻译。

由于翻译水平有限，可能会存在翻译不准确的地方。如果发现翻译后的内容不完整或者不准确，请及时联系(可发送邮件至[support@edatec.cn](mailto:support@edatec.cn))。

上海晶珩电子科技有限公司不保证翻译后的手册中的信息是最新的、正确的、完整的或高质量的，且不对这些信息的进一步使用做出保证。如果由于使用或不使用本手册中的信息，或由于使用错误或不完整的信息而造成的物质或非物质相关损失，上海晶珩电子科技有限公司不承担任何责任。

# 目录

法律免责声明 .....	2
译者免责声明 .....	3
目录 .....	2
1 简介 .....	1
1.1. 简介	1
1.2. 特征	2
2 接口 .....	3
2.1. 无线	3
2.1.1. WL_nDisable .....	3
2.1.2. BT_nDisable .....	4
2.2. 以太网 .....	4
2.3. PCIe (Gen2x1) .....	5
2.4. USB 2.0 (高速) .....	5
2.5. GPIO	6
2.5.1. 可供选择功能 .....	6
2.6. 备用GPIO功能 .....	7
2.7. 双HDMI 2.0 .....	8
2.8. MIPI (DSI/CSI-2) .....	8
2.9. CAM_GPIO (CAM_GPIO0 CAM_GPIO1) .....	8
2.10. I2C (SDA0 SCL0) .....	9
2.11. I2C (ID_SD ID_SC) .....	9
2.12. SDIO/eMMC (仅CM5Lite) .....	9
2.13. Type C (CC0 CC1) .....	10
2.14. PMIC_EN .....	10
2.15. PWR_BUT .....	10
2.16. VBAT .....	10
2.17. nRPI_BOOT .....	10
2.18. LED_nACT .....	10
2.19. EEPROM_nWP .....	11
2.20. Debug UART .....	11
3 电气和机械 .....	12
3.1. 机械	12
3.2. 散热	13
3.3. 电气特性 .....	13
4 引脚 .....	16
4.1. 差分对22	22
4.1.1. 100Ω差分对信号长度 .....	22
4.1.2. 90Ω差分对信号长度 .....	24
5 电源 .....	25
5.1. 上电时序控制 .....	25
2. PMIC_ENABLE 上升 .....	25
4. 在CM5_+3.3V后, +1.8V上升至至少1ms .....	25
5.2. 掉电序列 .....	25
5.3. 功耗	25
5.4. 调节器输出 .....	25
附录A: 故障排除 .....	26
硬件清单 .....	26
启动程序 .....	26
rpi-eprom更新 .....	26
EEPROM写保护 .....	26
固件 .....	27
内核 .....	27
附录B: 测试点 .....	28

测试点 .....	28
电压输入.....	30
调试串口.....	30
RPI 启动.....	30
以太网启动.....	30
附录C: CM4与CM5之间的差异 .....	31
Pinout 变化 .....	31
走线长度 .....	32
功率分配 .....	32
附录D: 可用性.....	33
支持.....	33
订购代码 .....	33
包装.....	34

# 1 简介

## 1.1. 简介

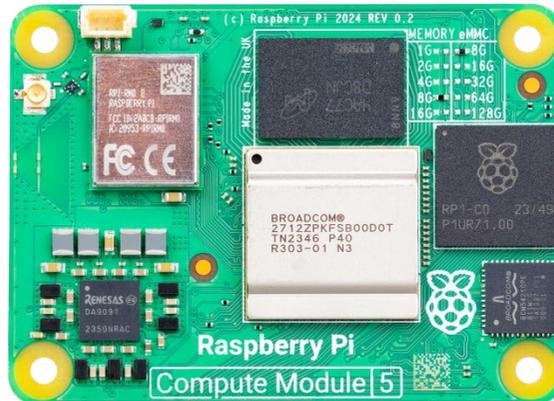


图 1 Raspberry Pi处理器模块5（CM5）

Raspberry Pi处理器模块5（CM5）是一种系统级模块（SoM），包含处理器、内存、eMMC闪存以及配套电源电路。这些模块允许设计人员将Raspberry Pi硬件和软件堆叠在自己的定制系统和外形中。此外，这些模块相较于Raspberry Pi主板，还提供了额外的输入输出（IO）接口，为设计者提供了更多选择。

CM5的设计大致基于Raspberry Pi 5，对于成本敏感型应用，它可以不选择eMMC；这个版本被称为Raspberry Pi处理器模块5精简版（CM5Lite）。

新一代CM5和CM5Lite在很大程度上与上一代处理模块兼容。上一代c处理器模块仍在销售，并将至少持续生产至2034年1月。CM5的电气接口通过两个100针的高密度连接器实现。

引脚分配的主要变化为增加2个USB 3.0端口。

### 注意

除非另有说明，本文档中的CM5也指CM5Lite。

## 1.2. 特征

CM5的主要特点如下:

- Broadcom [BCM2712](#), 四核Cortex-A76 (ARM v8) 64位2.4GHz SoC
- 4 x M2.5安装孔
- 4Kp60 HEVC 解码器
- OpenGL ES 3.1 图形, Vulkan 1.2
- 可选择带ECC的1GB、2GB、4GB、8GB 或16GLPDDR4-4267 SDRAM(请参见[订购代码](#))
- 可选择0GB (CM5Lite)、8GB、16GB或32GB eMMC闪存(请参见[订购代码](#))
  - 峰值eMMC带宽200MBps(比以前的处理器模块快2倍)
- 可选(参见[订购代码](#))经过无线认证的无线模块, 支持:
  - 2.4 GHz & 5.0 GHz IEEE 802.11 b/g/n/ac 无线通信协议
  - 蓝牙5.0, BLE无线通信协议
  - 板载电子开关, 用于选择PCB板载天线或外置天线
- 支持IEEE 1588协议的千兆以太网PHY
- 1 x PCIe单通道主机, 第二代(5Gbps)
- 1 x USB 2.0端口(高速)
- 1个USB 3.0端口, 支持同时以5Gbps的速度运行
- 高达30 x GPIO, 支持1.8V或3.3V信号和外设选项:
  - 最多5 x UART
  - 最多5 x I2C
  - 最多5 x SPI
  - 1 x SDIO接口
  - 1 x DPI (并行RGB显示)
  - 1 x I2S
  - 最多4 x PWM输出
  - 最多3 x GPCLK输出
- 2 x HDMI 2.0接口 (支持2端口同时输出最高4Kp60)
- 2 x 4通道MIPI接口:
  - 同时支持DSI (显示端口) 和CSI-2 (相机端口)
- 1 x SDIO 2.0 (CM5Lite)
- 单路+5V PSU输入, 支持USB PD, 最高可达5A@5V
- 实时时钟 (RTC), 由外部电池供电

## 2 接口

### 2.1. 无线

CM5可选择基于Cypress CYW43455的板载无线模块，同时支持以下两种功能：

- 2.4 GH & 5.0 GHz IEEE 802.11 b/g/n/ac 无线通信协议
- 蓝牙5.0, BLE无线通信协议

无线通信接口可以根据需要单独启用或禁用。例如在kiosk应用的情况下，服务工程师可以启用无线操作，然后在完成后禁用它。

CM5包含一个板载天线。如果使用，应将其放置在产品中，使其不被金属包围，包括任何接地层(参见[机械](#)了解更多详情)。CM5模块上有一个标准的U.FL连接器，参见图 1，便于连接外置天线。

Raspberry Pi Ltd可提供一个天线套件，经认证可用于CM5。如果使用不同的天线，则需要单独的认证。

#### 警告

Raspberry Pi Ltd无法协助对第三方天线进行认证。

系统启动时使用 `config.txt` 文件完成内置或外置天线的选择，且在运行期间不能更改。在 `config.txt` 文件中添加以下行之一以选择天线：

- `dtparam=ant1` 选择内置天线
- `dtparam=ant2` 选择外置天线

#### 2.1.1. WL\_nDisable

该引脚具有如下功能：

1. 用于监测WLAN无线通信功能的启动/禁用的状态，逻辑高表示WLAN功能被启用。
2. 该引脚被驱动或连接到低电平时，会禁止WLN模块上电，助于降低系统功耗，或者应用在需要从物理上禁用无线模块的应用中。该接口在被禁用后，如果再被重新启用，需要重新初始化WLAN驱动程序。

#### 注意

在不带无线功能的CM5模块上，此引脚为预留引脚。

### 2.1.2. BT\_nDisable

该引脚具有如下功能：

- 1.用于监测蓝牙通信功能启用/禁用的状态，逻辑高表示蓝牙功能被启用。
- 2.当该引脚被驱动或连接到低电平时，会阻止蓝牙模块上电，有助于降低系统功耗，或者应用在需要从物理上确保蓝牙被禁用的应用中。该接口在禁用后，如果再被重新启用，需要重新初始化蓝牙驱动程序。

**i 注意**

在不带无线功能的CM5模块上，此引脚为预留引脚。

## 2.2. 以太网

CM5包含一个板载千兆以太网PHY芯片，型号为Broadcom BCM54210PE，该PHY芯片的一些主要特征包括：

- 兼容IEEE 1588-2008协议
- MDI交叉、线对偏斜和线对极性校正

为CM5提供以太网连接只需要一个标准的1:1 RJ45连接器，支持PoE并增加了ESD保护，如图 2所示。

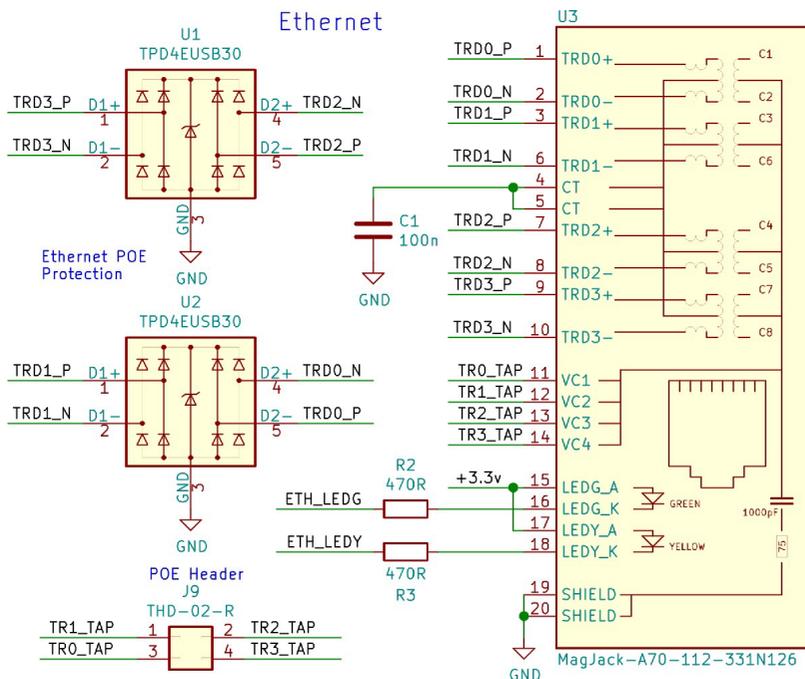


图 2 Raspberry Pi处理器模块5的以太网原理图接口

以太网信号应以100Ω阻抗的差分对进行走线，并留有适当的间隙。差分对之间的长度差异应小于等于50mm，一般情况下差分对之间不需要等长处理。然而差分对内的信号需要等长处理，差异要求小于或者等于0.15mm。

该以太网PHY还支持驱动高达3个LED指示灯，以便向用户提供各种状态反馈，这些LED指示灯驱动信号都是低电平有效。这些指示灯可实现一系列功能，可以查看你的操作系统驱动程序，以了解你的驱动程序支持哪些功能。

该以太网PHY还提供3.3V的SYNC\_OUT信号来支持IEEE 1588-2008协议。如有需要，可将该引脚定义为输入。

## 2.3. PCIe (Gen2x1)

CM5有一个内部PCIe 2.0 x1主机控制器。它可用于Gen 3.0，但不支持这种模式，而且可能无法在所有情况下使用。

### 警告

在制作产品原型机之前，你应该确保选择的任何主机控制器都有合适的操作系统驱动程序。

### 注意

板载PCIe主机控制器不支持来自ARM的64位访问，它们必须分成两个32位访问。

连接PCIe设备遵循标准的PCIe规范。CM5为PCIe\_TX信号配备了板载交流耦合电容器。但是，如果使用的是外部PCIe/NVMe卡，PCIe\_RX信号则需要在靠近驱动源（设备TX）的位置外接耦合电容器。PCIe规范规定，如果直接连接到集成电路（IC），则TX和RX对需要互换（即TX → RX，RX → TX）。如果连接到连接器，则通常从主机角度进行标记，因此无需进行TX/RX互换。此外，必须连接PCIe\_CLK\_nREQ以确保CM5产生时钟信号，还应连接PCIe\_nRST以确保在需要时正确重置设备。还提供PCIe\_nWAKE信号，但目前软件不支持该信号。

PCIe信号以90Ω阻抗差分对进行走线，并留有适当的间隙。差分对之间不需要等长处理，差分内等长的信号需要等长处理，要求差异小于或等于0.1mm。

### 提示

5.10及更新的内核版本增加了对MSI-X的支持。最多32个可用的IRQ。如果设备有中断问题，将pci=noms添加/cmdline.txt(并重新启动)可以解决问题。

## 2.4. USB 2.0 (高速)

USB 2.0接口支持高达480Mbps的数据传输。USB 2.0信号应以90Ω阻抗的差分对进行走线。差分对对内的P/N信号做等长处理，要求差异小于或者等于0.15mm。

### 提示

要在config.txt文件中启用USB接口，请使用dtoverlay=dwc2,dr\_mode=host来设置。

### 注意

该端口能够用作真正的USB移动(OTG)端口。虽然没有官方文档，但一些用户已经成功地做到了这一点。USB\_OTG\_ID引脚用于在USB主机和通常连接到Micro USB连接器ID引脚的设备之间进行选择。要使用此功能，必须在操作系统中启用。如果用作固定从机或固定主机，请将USB\_OTG\_ID引脚接地。

## 2.5. GPIO

有28个引脚可用于通用I/O（GPIO），这些引脚与Raspberry Pi 4 Model B的40针接头上的GPIO引脚相对应。这些引脚可以访问内部外设，包括SMI、DPI、I2C、PWM、SPI和UART。[RP1外设数据手册](#)描述了这些特性以及可用的复用选项。为了减少电磁兼容性（EMC）问题，理想情况下应将驱动强度和摆率设置为尽可能低。GPIO2和GPIO3配备了1.8kΩ的上拉电阻。

GPIO组由GPIO\_VREF供电，该引脚可以连接到CM5\_1.8V以进行1.8V信号的GPIO传输，或者连接到CM5\_3.3V以进行3.3V信号的传输。您应确保28个GPIO引脚上的负载总和不超过50mA。为确保CM5正确启动，GPIO\_VREF必须供电。如果使用外部+2.5v电源为GPIO\_VREF供电，则可以实现+2.5v信号传输。该外部电源仅在CM5\_1.8v开启时有效，当CM5\_1.8v降低时，需要在1ms内将2.5v电源放电。

### 2.5.1. 可供选择功能

支持6种功能可选，[BCM2712 ARM设备手册](#)详细说明了这些功能，[表 1](#)展示功能的概述。

**表 1 GPIO功能选择**

		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_CS[0]
1	SPI0_SIO[2]	DPI_DE	UART1_RX	I2C0_SCL		SYS_RIO[1]	PROC_RIO[1]	PIO[1]	SPI2_SIO[1]
2	SPI0_CS[3]	DPI_VSYN C	UART1_CTS	I2C1_SDA	UART0_IR_RX	SYS_RIO[2]	PROC_RIO[2]	PIO[2]	SPI2_SIO[0]
3	SPI0_CS[2]	DPI_HSYN C	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_CS[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_CS[1]	DPI_D[3]	UART2_RTS	I2C3_SCL	UART0_DSR	SYS_RIO[7]	PROC_RIO[7]	PIO[7]	SPI3_SCLK
8	SPI0_CS[0]	DPI_D[4]	UART3_TX	I2C0_SDA		SYS_RIO[8]	PROC_RIO[8]	PIO[8]	SPI4_CS[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_CS[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_CS[2]	DPI_D[12]			UART0_CTS	SYS_RIO[16]	PROC_RIO[16]	PIO[16]	
17	SPI1_CS[1]	DPI_D[13]			UART0_RTS	SYS_RIO[17]	PROC_RIO[17]	PIO[17]	
18	SPI1_CS[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]	
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]

每个GPIO每次只能选择一个功能。同样，每个外设输入（例如I2C3\_SCL）每次也只能在一个GPIO上被选中。如果同一个外设输入连接到多个GPIO，则该外设将看到这些GPIO输入的逻辑（OR）结果。

**注意**

此列表中未命名功能的选项是预留的。

## 2.6. 备用GPIO功能

- 5x 串口
  - 4个4线接口 (TX, RX, CTS, RTS)
  - UART0 支持8线接口 (TX, RX, CTS, RTS, DTR, DCD, DSR, RI) or IrDA (IR\_TX, IR\_RX)
- 1x SDIO (4 bit)
- 4x PWMs
- 1x I2S, 主控制器 (ISC0), 四通道
- 1x I2S, 从控制器 (ISC1), 四通道
- 2x AUDIO\_OUT PWM 音频输出，需要使用低电源噪声缓冲器进行缓冲，并通过22KHZ的一阶RC网络进行滤波
- 2x AUDIO\_IN: 数字PDM输入
- 2x GPCLK 时钟输出
- 1x DPI (数字显示接口) (PCLK, DE, VSYNC, HSYNC and 最多24位数据)
- 28x GPIO (SYS\_RIO)
- 4x I2C (SDA, SCL)
- SPI:

实例ID	主站/从站	片选计数	最大 I/O 宽度
SPI0	M	4	Quad
SPI1	M	3	Dual

SPI2	M	2	Dual
SPI3	M	2	Dual
SPI4	S	1	Single
SPI5	M	2	Dual

对于传统SPI连接，SIO[0]是MOSI（主输出从输入），而SIO[1]是MISO（主输入从输出）。其他的SIO引脚则不需要。

## 2.7. 双HDMI 2.0

CM5支持两路HDMI 2.0接口，每路接口都支持4K图像。

HDMI信号应以100Ω差分对的形式进行布线。理想情况下，一对内的每个信号之间的匹配精度应优于0.15mm。通常，差分对不需要任何额外的匹配，因为它们只需匹配到25mm的范围内即可。

支持CEC，CM5内置一个27kΩ的上拉电阻。

提供了EDID信号的内部上拉电阻。在Raspberry Pi 5上，HDMI信号没有额外的静电放电（ESD）保护。根据应用的不同，可能需要额外的ESD保护。

## 2.8. MIPI（DSI/CSI-2）

CM5支持两个4通道的MIPI接口。每个MIPI接口均可以用作DSI（显示接口）或CSI（摄像头接口）。MIPI信号应以100Ω差分对的形式进行布线。理想情况下，一对内的每个信号之间的匹配精度应优于0.15mm。

有关CSI接口的文档可以在[Raspberry Pi网站](#)上找到，而Linux内核驱动程序则可以在[GitHub](#)上找到。尽管有可用的Linux内核驱动程序，但目前还没有关于DSI接口的文档。仅支持由Raspberry Pi官方固件支持的DSI显示器。

### **i** 注意

官方的Raspberry Pi固件支持OmniVision OV5647、Sony IMX219、Sony IMX296、Sony IMX477和Sony IMX708相机传感器。在处理器模块设备上使用这些相机传感器，不需要安全设备。

### **i** 注意

虽然仅支持官方DSI显示器，但可以使用作为GPIO备用功能的并行DPI接口添加其他显示器。CM5一次最多支持三种任何类型的显示器（HDMI、DSI、DPI）中的一种。

## 2.9. CAM\_GPIO ( CAM\_GPIO0 CAM\_GPIO1 )

CAM\_GPIO0通常连接到相机连接器的第17脚，用于控制相机模块的电源开关。

CAM\_GPIO1是一个之前处理器模块上没有的新信号。建议将其连接到相机连接器的第18脚，以备将来扩展。

## 2.10. I2C (SDA0 SCL0)

此内部I2C总线通常分配给CSI1和DSI1，这些设备由固件控制。如果CSI1和DSI1接口没有被使用，或者被固件控制，其可以被用作通用I2C总线。例如，`libcamera`运行在ARM上，如果CSI1和DSI1接口没有被使用，或者被固件控制；在这种情况下，可以使用CSI1和I2C总线。SDA0连接到BCM2712上的GPIO44，SCL0连接到GPIO45。

## 2.11. I2C (ID\_SD ID\_SC)

该I2C总线通常用于识别HATs和控制CSI0和DSI0设备。如果固件没有使用I2C总线，例如没有使用CSI0和DSI0，则这些引脚可以在需要时用作GPIO 0和GPIO 1。

### 注意

如果这些引脚用作GPIO引脚，则为了防止固件检查是否有HAT EEPROM可用，请将`force_eeprom_read=0`和`disable_poe_fan=1`添加到`config.txt`文件中。

## 2.12. SDIO/eMMC (仅CM5Lite)

CM5Lite不支持板载eMMC，eMMC信号可以在连接器上使用，因此可以使用外部eMMC或SD卡。

`SD_PWR_ON`信号用于使能外部电源开关，以打开SD卡的电源；对于eMMC，通常不使用。如果需从SD卡启动，则需要安装一个上拉电阻来默认打开电源开关。当`SD_VDD_OVERRIDE`为高电平(3.3V)时，会在SDIO接口上强制发送1.8V信号，通常与eMMC存储一起使用。

注意：SD卡需要一个由`SD_PWR_ON`控制的电源开关，因为这是重置SD卡的唯一方法。

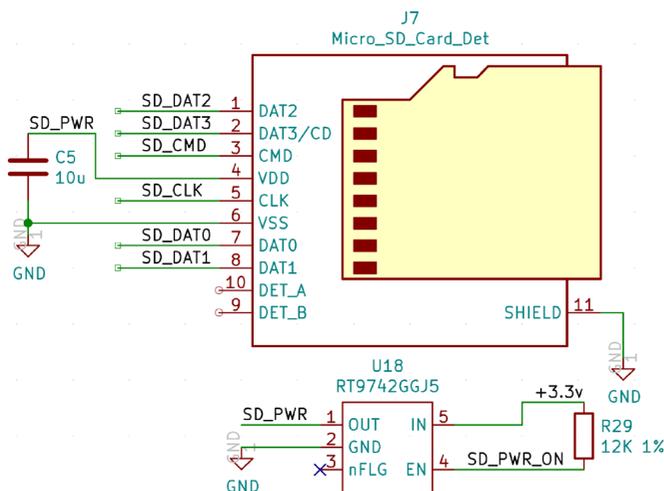


图 3 CM5Lite SD卡接口

## 2.13. Type C ( CC0 CC1 )

CC0和CC1是在Type-C连接器上用于电源协商的信号。CM5将使用这些信号来尝试协商5V @ 5A的电源供应。

## 2.14. PMIC\_EN

将该引脚拉低可使CM5进入可能的最低功耗状态。

### 提示

建议在操作系统关闭后，再将此引脚拉低。

## 2.15. PWR\_BUTTON

当这个引脚被拉低时，如果CM5之前已经关闭，那么它将启动CM5；如果保持低电平，则关闭CM5。这个信号通常连接到一个按钮上，作为电源按钮。

## 2.16. VBAT

这是一个2.7至3v的电源输入，用于为RTC供电。它的设计目的是在电源中断期间保持RTC运行，而不是用于长期为RTC供电。当CM5正在运行时，会有一个小的静态负载。当CM5不运行时，负载会增加为RTC供电。当CM5未通电时，一个的CR2032电池应该能够持续使用三年以上。

## 2.17. nRPI\_BOOT

在启动过程中，如果此引脚为低电平，将停止从eMMC启动，会通过USB转移到rpi启动。

## 2.18. LED\_nACT

该引脚设计用于驱动一个LED来复制Raspberry Pi 5上的绿色LED。在Linux系统下，此引脚闪烁表示eMMC访问。如果在启动过程中出现任何错误，此LED将闪烁错误模式，可以在Raspberry网站上查找[解码表](#)。

## 2.19. LED\_nPWR

这个引脚需要经过缓冲才能驱动LED。该信号的设计是为了复制Raspberry Pi 5上的红色LED的功能。当电路板通电但系统关闭时，LED会亮起。

## 2.20. EEPROM\_nWP

建议将该引脚拉低，以防止最终用户改变板载EEPROM的内容。有关支持所需的软件设置的说明，请参见Raspberry Pi 5文档[EEPROM写保护](#)。

## 2.21. Debug UART

为用户提供了安装调试UART连接器的空间。该连接器与Raspberry Pi 5具有相同的功能。连接器是一个三针1mm间距的JST-SH连接器，零件号为BM03B-SRSS-TB。信号在底部以测试点的形式复制。

## 3 电气和机械

### 3.1. 机械

CM5是一个紧凑的40mmx55mm模块。该模块厚度为4.6mm，连接后的高度将为4.94mm或7.44mm，具体取决于所选的堆叠高度。

1. 4xM2.5安装孔(距离模块边缘3.5mm)
2. PCB厚度 $1.24\text{mm}\pm 10\%$
3. BCM2712 含焊球的SoC高度 $2.2\pm 0.15\text{mm}$
4. 堆叠高度:
  - a. 配套1.5mm高度的连接器(CM5以下的间隙为0mm): P/N 10164227-1001A1RLF
  - b. 配套4.0mm高度的连接器 (CM5下的间隙为2.5mm): P/N 10164227-1004A1RLF

如果使用板载无线天线(参见2.1无线)，必须朝向塑料外壳的边缘，任何附近的金属都必须有切口，否则无线性能会下降。建议PCB天线周围至少有10mm的无金属净空区域，但设计人员必须检查性能。

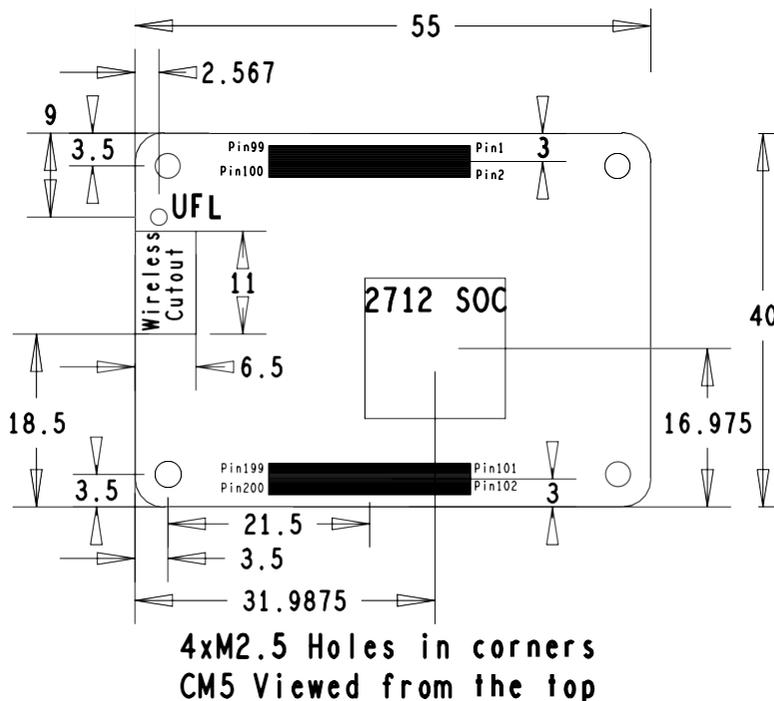


图 4 Raspberry Pi处理器模块5的机械规格

天线下方不能有任何金属，包括接地层。如有接地层，则对应接地层开口最少为 $6.5\text{mm}\times 11\text{mm}$ ，但理想情况下至少为 $8\text{mm}\times 15\text{mm}$ 。如果不能满足这些要求，无线性能可能会下降，尤其是在2.4GHz频谱中。建议尽可能使用外置天线。

#### **i** 注意

由于成本和制造方面的考虑，组件在处理器模块上的位置和排列可能会随着时间的推移而略有变化；但最大元件高度和PCB厚度将保持不变。

CM5的step文件是CM5设计数据包的一部分。这仅供参考，可能会随着时间的推移因修订而发生变化。

## 3.2. 散热

与Raspberry Pi 5相比，CM5的印刷电路板（PCB）中所含的金属更少，连接器也更少，这意味着它的被动散热能力不如Raspberry Pi 5。

BCM2712会降低主时钟的工作频率，以控制芯片内部温度保持在85°C以下。故在高温环境下，主时钟的工作频率也可能会自动降低。如果BCM2712无法将其内部时钟降低到足以降低内部温度的程度，其金属外壳温度将升至85°C以上。选择的任何散热解决方案都要将CM5上其它硅器件的环境温度保持在工作温度范围内。

工作温度范围：-20°C-+85°C非冷凝。注意：最佳射频无线性能介于-20°C和+75°C之间。

## 3.3. 电气特性

### 警告

高于下表中所列的限度值可能会对设备造成永久性损坏。这只是一个压力等级，不暗示设备在这些或本规范操作部分所列条件之外的任何其他条件下的功能操作。长时间暴露在绝对最大额定值条件下可能会影响设备的可靠性。

表 2 绝对最大额定值

标志	参数	最小值	最大值	单位
V <sub>IN</sub>	5V输入电压	-0.5	6.0	V
V <sub>GPIO_VREF</sub>	GPIO电压	-0.5	3.6	V
V <sub>gpio</sub>	GPIO输入电压	-0.5	V <sub>GPIO_VREF</sub> + 0.5	V

### 注意

V<sub>GPIO\_VREF</sub>是GPIO组电压，必须连接到CM5的3.3V或1.8V供电。

表 3 DC特征

标志	参数	条件	最小值	典型值	最大值	单位
V <sub>IL(gpio)</sub>	输入低电压	V <sub>GPIO_VREF</sub> = 3.3V	0	-	0.8	V
V <sub>IH(gpio)</sub>	输入高电压	V <sub>GPIO_VREF</sub> = 3.3V	2.0	-	V <sub>GPIO_VREF</sub>	V
V <sub>IL(gpio)</sub>	输入低电压	V <sub>GPIO_VREF</sub> = 2.5V	0	-	0.7	V
V <sub>IH(gpio)</sub>	输入高电压	V <sub>GPIO_VREF</sub> = 2.5V	1.7	-	V <sub>GPIO_VREF</sub>	V
V <sub>IL(gpio)</sub>	输入低电压	V <sub>GPIO_VREF</sub> = 1.8V	0	-	0.35*V <sub>GPIO_VREF</sub>	V
V <sub>IH(gpio)</sub>	输入高电压	V <sub>GPIO_VREF</sub> = 1.8V	0.65*V <sub>GPIO_VREF</sub>	-	V <sub>GPIO_VREF</sub>	V

$I_{L(gpio)}$	输入泄漏电流	$V_{GPIO\_VREF} = 3.3V$	-	-	3	$\mu A$
$I_{L(gpio)}$	输入泄漏电流	$V_{GPIO\_VREF} = 2.5V$	-	-	5	$\mu A$
$I_{L(gpio)}$	输入泄漏电流	$V_{GPIO\_VREF} = 1.8V$	-	-	7	$\mu A$
$V_{OL(gpio)}$	输出低电压	-	-	-	0.4	V
$V_{OH(gpio)}$	输出高电压	$V_{GPIO\_VREF} = 3.3V$	$V_{GPIO\_VREF} - 0.4$	-	-	V
$V_{OH(gpio)}$	输出高电压	$V_{GPIO\_VREF} = 2.5V$	$V_{GPIO\_VREF} - 0.5$	-	-	V
$V_{OH(gpio)}$	输出高电压	$V_{GPIO\_VREF} = 1.8V$	$V_{GPIO\_VREF} - 0.4$	-	-	V
$I_{OL(gpio)}$	输出电流	2mA, $V_{GPIO\_VREF} = 3.3V$	6.1	9.6	13.5	mA
$I_{OL(gpio)}$	输出电流	4mA, $V_{GPIO\_VREF} = 3.3V$	9.2	14.3	20.2	mA
$I_{OL(gpio)}$	输出电流	8mA, $V_{GPIO\_VREF} = 3.3V$	15.3	23.9	33.7	mA
$I_{OL(gpio)}$	输出电流	12mA, $V_{GPIO\_VREF} = 3.3V$	18.4	28.7	40.5	mA
$I_{OH(gpio)}$	输出电流	2mA, $V_{GPIO\_VREF} = 3.3V$	4.5	6.3	8.4	mA
$I_{OH(gpio)}$	输出电流	4mA, $V_{GPIO\_VREF} = 3.3V$	6.8	9.5	12.6	mA
$I_{OH(gpio)}$	输出电流	8mA, $V_{GPIO\_VREF} = 3.3V$	11.4	15.8	21	mA
$I_{OH(gpio)}$	输出电流	12mA, $V_{GPIO\_VREF} = 3.3V$	13.6	19	25.2	mA
$I_{OL(gpio)}$	输出电流	2mA, $V_{GPIO\_VREF} = 2.5V$	4.7	8	12.2	mA
$I_{OL(gpio)}$	输出电流	4mA, $V_{GPIO\_VREF} = 2.5V$	7.1	12	18.2	mA
$I_{OL(gpio)}$	输出电流	8mA, $V_{GPIO\_VREF} = 2.5V$	11.8	20	30.4	mA
$I_{OL(gpio)}$	输出电流	12mA, $V_{GPIO\_VREF} = 2.5V$	14.1	24	36.4	mA
$I_{OH(gpio)}$	输出电流	2mA, $V_{GPIO\_VREF} = 2.5V$	3.5	5.1	7	mA
$I_{OH(gpio)}$	输出电流	4mA, $V_{GPIO\_VREF} = 2.5V$	5.2	7.6	10.5	mA

$I_{OH(gpio)}$	输出电流	12mA, $V_{GPIO\_VREF} = 2.5V$	10.4	15.2	21.1	mA
$I_{OL(gpio)}$	输出电流	2mA, $V_{GPIO\_VREF} = 1.8V$	4.4	8.1	13.6	mA
$I_{OL(gpio)}$	输出电流	4mA, $V_{GPIO\_VREF} = 1.8V$	8.8	16.3	27.2	mA
$I_{OL(gpio)}$	输出电流	8mA, $V_{GPIO\_VREF} = 1.8V$	11.8	21.7	36.3	mA
$I_{OL(gpio)}$	输出电流	12mA, $V_{GPIO\_VREF} = 1.8V$	16.2	29.2	49.9	mA
$I_{OH(gpio)}$	输出电流	2mA, $V_{GPIO\_VREF} = 1.8V$	3.4	5.3	7.7	mA
$I_{OH(gpio)}$	输出电流	4mA, $V_{GPIO\_VREF} = 1.8V$	6.9	10.5	15.4	mA
$I_{OH(gpio)}$	输出电流	8mA, $V_{GPIO\_VREF} = 1.8V$	9.1	14	20.6	mA
$I_{OH(gpio)}$	输出电流	12mA, $V_{GPIO\_VREF} = 1.8V$	12.6	19.3	28.3	mA
$R_{PU(gpio)}$	上拉电阻	$V_{GPIO\_VREF} = 3.3V$	37	55	86	k $\Omega$
$R_{PD(gpio)}$	下拉电阻	$V_{GPIO\_VREF} = 3.3V$	35	55	98	k $\Omega$
$R_{PU(gpio)}$	上拉电阻	$V_{GPIO\_VREF} = 2.5V$	49	77	123	k $\Omega$
$R_{PD(gpio)}$	下拉电阻	$V_{GPIO\_VREF} = 2.5V$	49	84	155	k $\Omega$
$R_{PU(gpio)}$	上拉电阻	$V_{GPIO\_VREF} = 1.8V$	38	64	106	k $\Omega$
$R_{PD(gpio)}$	下拉电阻	$V_{GPIO\_VREF} = 1.8V$	58	103	189	k $\Omega$

有关其他接口的电气细节，请参阅接口规格（参见第二章）。

**表 4 功耗**

标志	参数	条件	最小值	典型值	最大值	单位
$I_{shutdown}$	关闭电流	$PMIC\_ENABLE = 0V$	-	1.5	-	mA
$I_{shutdown}$	关闭电流	$PMIC\_ENABLE > 2V$	-	2.8	-	mA
$I_{idle}$	空闲电流	$PMIC\_ENABLE > 2V$	-	450	-	mA
$I_{load}$	工作电流	$PMIC\_ENABLE > 2V$	-	900	-	mA
$I_{Vbat}$	RTC 电流	$V_{in} = +5v$	-	1.7	-	$\mu A$
$I_{Vbat}$	RTC 电流	$V_{in} = 0v$	-	6	-	$\mu A$

**注意**

上表中的数值很大程度上取决于最终应用。

# 4 引脚

表 5 Raspberry Pi处理器模块5引脚列表

引脚	信号	描述
1	GND	接地(0V)
2	GND	接地(0V)
3	Ethernet_Pair3_P	以太网对3正极(连接到变压器或磁插孔)
4	Ethernet_Pair1_P	以太网对1正极(连接到变压器或磁插孔)
5	Ethernet_Pair3_N	以太网对3负极(连接到变压器或磁插孔)
6	Ethernet_Pair1_N	以太网对1负极(连接到变压器或磁插孔)
7	GND	接地(0V)
8	GND	接地(0V)
9	Ethernet_Pair2_N	以太网对2负极(连接到变压器或磁插孔)
10	Ethernet_Pair0_N	以太网对0负极(连接到变压器或磁插孔)
11	Ethernet_Pair2_P	以太网对2正极(连接到变压器或磁插孔)
12	Ethernet_Pair0_P	以太网对0正极(连接到变压器或磁插孔)
13	GND	接地(0V)
14	GND	接地(0V)
15	Ethernet_nLED3	低电平有效以太网活动指示器(CM5_3.3V信号):通常绿色LED连接到此引脚。IOL = 8mA @ VOL < 0.4V
16	Fan_Tacho	Fan Tacho Input 引脚通过 1.8K 电阻内部上拉到 CM5_3.3V
17	Ethernet_nLED2	低电平有效以太网速度指示器(CM5_3.3V信号):通常黄色LED连接到此引脚。低状态表示1gb或100Mbit链路:IOL = 8mA @ VOL < 0.4V
18	Ethernet_SYNC_OUT	IEEE1588同步输出引脚(CM5_1.8V信号:IOL = 8mA @ VOL < 0.4V)
19	Fan_PWM	( 开漏输出 ):
20	EEPROM_nWP	让浮动NB通过100kΩ在内部上拉至CM5_3.3V(VIL < 0.8V), 但可以接地以防止写入存储引导代码的片上EEPROM
21	LED_nACT	低电平有效的Pi活动指示灯。最大电流20mA, 可承受5V电压 (VOL<0.4V)。(这是驱动Raspberry Pi 5上绿色指示灯的信号)
22	GND	接地(0V)
23	GND	接地(0V)
24	GPIO26	GPIO:通常是3.3V信号, 但通过将GPIO_VREF连接到1.8V, 也可以是CM5_1.8V信号
25	GPIO21	GPIO:通常是3.3V信号, 但通过将GPIO_VREF连接到1.8V, 也可以是CM5_1.8V信号
26	GPIO19	GPIO:通常是3.3V信号, 但通过将GPIO_VREF连接到1.8V, 也可以是CM5_1.8V信号
27	GPIO20	GPIO:通常是3.3V信号, 但通过将GPIO_VREF连接到1.8V, 也可以是CM5_1.8V信号
28	GPIO13	GPIO:通常是3.3V信号, 但通过将GPIO_VREF连接到1.8V, 也可以是CM5_1.8V信号

29	GPIO16	GPIO:通常是3.3V信号, 但通过将GPIO_VREF连接到1.8V, 也可以是CM5_1.8V信号
30	GPIO6	GPIO:通常是3.3V信号, 但通过将GPIO_VREF连接到1.8V, 也可以是CM5_1.8V信号
31	GPIO12	GPIO:通常是3.3V信号, 但通过将GPIO_VREF连接到1.8V, 也可以是CM5_1.8V信号
32	GND	接地(0V)
33	GND	接地(0V)
34	GPIO5	GPIO:通常是3.3V信号, 但通过将GPIO_VREF连接到1.8V, 也可以是CM5_1.8V信号
35	ID_SC	(BCM2712 GPIO 1) GPIO:通常是3.3V信号, 但通过将GPIO_VREF连接到1.8V, 也可以是CM5_1.8V信号
36	ID_SD	(BCM2712 GPIO 0) GPIO:通常是3.3V信号, 但通过将GPIO_VREF连接到1.8V, 也可以是CM5_1.8V信号
37	GPIO7	GPIO:通常是3.3V信号, 但通过将GPIO_VREF连接到1.8V, 也可以是CM5_1.8V信号
38	GPIO11	GPIO:通常是3.3V信号, 但通过将GPIO_VREF连接到1.8V, 也可以是CM5_1.8V信号
39	GPIO8	GPIO:通常是3.3V信号, 但通过将GPIO_VREF连接到1.8V, 也可以是CM5_1.8V信号
40	GPIO9	GPIO:通常是3.3V信号, 但通过将GPIO_VREF连接到1.8V, 也可以是CM5_1.8V信号
41	GPIO25	GPIO:通常是3.3V信号, 但通过将GPIO_VREF连接到1.8V, 也可以是CM5_1.8V信号
42	GND	接地(0V)
43	GND	接地(0V)
44	GPIO10	GPIO:通常是3.3V信号, 但通过将GPIO_VREF连接到1.8V, 也可以是CM5_1.8V信号
45	GPIO24	GPIO:通常是3.3V信号, 但通过将GPIO_VREF连接到1.8V, 也可以是CM5_1.8V信号
46	GPIO22	GPIO:通常是3.3V信号, 但通过将GPIO_VREF连接到1.8V, 也可以是CM5_1.8V信号
47	GPIO23	GPIO:通常是3.3V信号, 但通过将GPIO_VREF连接到1.8V, 也可以是CM5_1.8V信号
48	GPIO27	GPIO:通常是3.3V信号, 但通过将GPIO_VREF连接到1.8V, 也可以是CM5_1.8V信号
49	GPIO18	GPIO:通常是3.3V信号, 但通过将GPIO_VREF连接到1.8V, 也可以是CM5_1.8V信号
50	GPIO17	GPIO:通常是3.3V信号, 但通过将GPIO_VREF连接到1.8V, 也可以是CM5_1.8V信号
51	GPIO15	GPIO:通常是3.3V信号, 但通过将GPIO_VREF连接到1.8V, 也可以是CM5_1.8V信号
52	GND	接地(0V)
53	GND	接地(0V)
54	GPIO4	GPIO:通常是3.3V信号, 但通过将GPIO_VREF连接到1.8V, 也可以是CM5_1.8V信号
55	GPIO14	GPIO:通常是3.3V信号, 但通过将GPIO_VREF连接到1.8V, 也可以是CM5_1.8V信号
56	GPIO3	GPIO:通常是3.3V信号, 但通过将GPIO_VREF连接到1.8V, 也可以是CM5_1.8V信号 内部1.8kΩ上拉到GPIO_VREF
57	SD_CLK	SD卡时钟信号(仅适用于CM5Lite)
58	GPIO2	GPIO:通常是3.3V信号, 但通过将GPIO_VREF连接到1.8V, 也可以是CM5_1.8V信号 内部1.8kΩ上拉到GPIO_VREF
59	GND	接地(0V)
60	GND	接地(0V)
61	SD_DAT3	SD卡/eMMC数据3信号(仅适用于CM5Lite)
62	SD_CMD	SD卡/eMMC命令信号(仅在CM5Lite上可用)

63	SD_DAT0	SD卡/eMMC数据0信号(仅适用于CM5Lite)
64	SD_DAT5	SD卡/eMMC数据5信号(仅适用于CM5Lite)
65	GND	接地(0V)
66	GND	接地(0V)
67	SD_DAT1	SD卡/eMMC数据1信号(仅适用于CM5Lite)
68	SD_DAT4	SD卡/eMMC数据4信号(仅适用于CM5Lite)
69	SD_DAT2	SD卡/eMMC数据2信号(仅适用于CM5Lite)
70	SD_DAT7	SD卡/eMMC数据7信号(仅适用于CM5Lite)
71	GND	接地(0V)
72	SD_DAT6	SD卡/eMMC数据6信号(仅适用于CM5Lite)
73	SD_VDD_OVERRIDE	连接到CM5_3.3V以强制SD卡/eMMC接口为1.8V信号，而不是3.3V，否则保持不连接。通常仅在连接外部eMMC时使用。
74	GND	接地(0V)
75	SD_PWR_ON	输出到SD卡的电源开关。CM5将此引脚设为高电平(CM5_3.3V)，表示SD卡的电源应该打开。如果需要从SD卡启动，那么也应该安装一个上拉开关，这样电源开关默认为on。(仅在CM5Lite上提供)
76	VBAT	RTC电池输入，典型值为3V，范围在2.5V至3.5V之间
77	+5V (Input)	4.75V-5.25V。主电源输入
78	GPIO_VREF	对于3.3V GPIO，必须连接到CM5_3.3V(引脚84和86)，对于1.8V GPIO，必须连接到CM5_1.8V(引脚88和90)。此引脚不能悬空或接地。
79	+5V (Input)	4.75V-5.25V。主电源输入
80	SCL0	I2C时钟引脚(BCM2711 GPIO45):通常用于摄像头和显示器。内部1.8kΩ上拉至CM5_3.3V
81	+5V (Input)	4.75V-5.25V。主电源输入
82	SDA0	I2C数据引脚(BCM2712 GPIO44):通常用于摄像头和显示器。内部1.8kΩ上拉至CM5_3.3V
83	+5V (Input)	4.75V-5.25V。主电源输入
84	CM5_3.3V (Output)	3.3V 2.5%。每个引脚的最大功率输出为300毫安，总共为600毫安。这将在关断或GLOBAL_EN设为低电平期间关断
85	+5V (Input)	4.75V-5.25V。主电源输入
86	CM5_3.3V (Output)	3.3V 2.5%。每个引脚的最大功率输出为300毫安，总共为600毫安。这将在关断或GLOBAL_EN设为低电平期间关断
87	+5V (Input)	4.75V-5.25V。主电源输入
88	CM5_1.8V (Output)	1.8V 2.5%。每个引脚的最大功率输出为300毫安，总共为600毫安。这将在关断或GLOBAL_EN设为低电平期间关断
89	WL_nDisable	可以保持浮动；如果被拉低，无线接口将被禁用。通过1.8kΩ内部上拉至CM5_3.3V
90	CM5_1.8V (Output)	1.8V 2.5%。每个引脚的最大功率输出为300毫安，总共为600毫安。这将在关断或GLOBAL_EN设为低电平期间关断
91	BT_nDisable	可以保持浮动；如果驱动为低电平，蓝牙接口将被禁用。通过1.8kΩ内部上拉至CM5_3.3V

92	PWR_Button	将其拉低以强制关机或从之前的软件关机状态开机。内部通过10kΩ上拉到+5V。
93	nRPIBOOT	此引脚为低电平时，会强制从RPI服务器(如PC或Raspberry Pi)启动；如果不使用，保持悬空。通过10kΩ内部上拉至CM5_3.3V
94	CC1	USB PSU的电源传输PD信号，将其连接到Type-C接口以实现5V下5A电流的协商。
95	LED_nPWR	(+3.3V信号) 低电平有效输出，用于驱动电源指示灯。此信号需要缓冲。
96	CC2	USB PSU的电源传输PD信号，将其连接到Type-C接口以实现5V下5A电流的协商。
97	CAM_GPIO0	(+3.3V信号) 可以是GPIO，也可以是包含引脚100的总线的一部分
98	GND	接地(0V)
99	PMIC_Enable	输入。将其拉低以关闭CM5的电源。内部通过100kΩ上拉到+5V。
100	CAM_GPIO1	(+3.3V信号) 内部通过15K上拉到CM5_3.3V。
101	USB_OTG_ID	输入(+3.3V信号) USB OTG引脚。内部上拉。当接地时，CM5成为USB主机，但也需要使用正确的操作系统驱动程序。
102	PCle_CLK_nREQ	输入 (+3.3V信号) PCIe时钟请求引脚（拉低以请求PCI时钟）。内部上拉。
103	USB_N	USB2.0 D-
104	PCIE_nWAKE	(+3.3V信号) 如果不需要唤醒功能，PCIE WAKE#信号可以不连接，内部已上拉。
105	USB_P	USB2.0 D+
106	PCIE_PWR_EN	(+3.3V信号) 高电平有效，当信号为低时，表示PCIE设备可以被关闭电源。
107	GND	接地(0V)
108	GND	接地(0V)
109	PCle_nRST	输出 (+3.3V信号) PCIe复位信号，低电平有效
110	PCle_CLK_P	PCIE时钟输出正极 (100MHz)
111	VBUS_EN	(+3.3V信号) 高电平有效，表示USB3端口应该被供电
112	PCle_CLK_N	PCIE时钟输出负极 (100MHz)
113	GND	接地(0V)
114	GND	接地(0V)
115	MIPI0_D0_N	MIPI0 D0 负极
116	PCle_RX_P	输入PCIE GEN 2 RX正极，需要外部NB交流耦合电容器
117	MIPI0_D0_P	MIPI0 D0 正极
118	PCle_RX_N	输入PCIE GEN 2 RX正极，需要外部NB交流耦合电容器
119	GND	接地(0V)
120	GND	接地(0V)
121	MIPI0_D1_N	MIPI0 D1 负极
122	PCle_TX_P	输出PCIE GEN 2 TX正极，CM5上包含NB交流耦合电容器
123	MIPI0_D1_P	MIPI0 D1正极
124	PCle_TX_N	输出PCIE GEN 2 TX负极，CM5上包含NB交流耦合电容器
125	GND	接地(0V)

126	GND	接地(0V)
127	MIPI0_C_N	MIPI0 时钟负极
128	USB3-0-RX_N	USB3 端口0 RX 输入负极
129	MIPI0_C_P	MIPI0 时钟正极
130	USB3-0-RX_P	USB3 端口0 RX 输入正极
131	GND	接地(0V)
132	GND	接地(0V)
133	MIPI0_D2_N	MIPI0 D2 负极
134	USB3-0-DP	USB3端口 0 DP
135	MIPI0_D2_P	MIPI0 D2 正极
136	USB3-0-DM	USB3 端口 0 DM
137	GND	接地(0V)
138	GND	接地(0V)
139	MIPI0_D3_N	MIPI0 D3 负极
140	USB3-0-TX_N	CM5上包含USB3端口0的TX输出负极以及NB交流耦合电容器
141	MIPI0_D3_P	MIPI0 D3 正极
142	USB3-0-TX_P	CM5上包含USB3端口0的TX输出正极以及NB交流耦合电容器
143	HDMI1_HOTPLUG	输入HDMI1热插拔检测。内部通过一个100kΩ电阻下拉。可承受5V电压。(可直接连接到HDMI连接器上)
144	GND	接地(0V)
145	HDMI1_SDA	双向HDMI1 SDA。内部通过一个1.8kΩ电阻上拉。可承受5V电压。(可直接连接到HDMI连接器上)
146	HDMI1_TX2_P	输出 HDMI1 TX2 正极
147	HDMI1_SCL	双向HDMI1 SCL。内部通过一个1.8kΩ电阻上拉。可承受5V电压。(可直接连接到HDMI连接器上)
148	HDMI1_TX2_N	输出 HDMI1 TX2 负极
149	HDMI1_CEC	输入HDMI1 CEC信号。内部通过一个27kΩ电阻上拉。可承受5V电压。(可直接连接到HDMI连接器上)
150	GND	接地(0V)
151	HDMI0_CEC	输入HDMI0 CEC信号。内部通过一个27kΩ电阻上拉。可承受5V电压。(可直接连接到HDMI连接器上)
152	HDMI1_TX1_P	输出HDMI1 TX1 正极
153	HDMI0_HOTPLUG	输入HDMI0热插拔检测。内部通过100kΩ下拉。支持5V电压。(可直接连接到HDMI连接器)
154	HDMI1_TX1_N	输出HDMI1 TX1 负极
155	GND	接地(0V)
156	GND	接地(0V)
157	USB3-1-RX_N	USB3 端口1 RX 输入负极

158	HDMI1_TX0_P	输出 HDMI1 TX0正极
159	USB3-1-RX_P	USB3 端口 1 RX 输入正极
160	HDMI1_TX0_N	输出 HDMI1 TX0负极
161	GND	接地(0V)
162	GND	接地(0V)
163	USB3-1-DP	USB3 端口 1 DP
164	HDMI1_CLK_P	输出 HDMI1 时钟正极
165	USB3-1-DM	USB3 端口 1 DM
166	HDMI1_CLK_N	输出 HDMI1 时钟负极
167	GND	接地(0V)
168	GND	接地(0V)
169	USB3-1-TX_N	CM5上包含USB3端口1的TX输出负极, 以及NB交流耦合电容器
170	HDMI0_TX2_P	输出 HDMI0 TX2 正极
171	USB3-1-TX_P	CM5上包含USB3端口1的TX输出正极, 以及NB交流耦合电容器
172	HDMI0_TX2_N	输出 HDMI0 TX2 负极
173	GND	接地(0V)
174	GND	接地(0V)
175	MIPI1_D0_N	MIPI1 D0 负极
176	HDMI0_TX1_P	输出 HDMI0 TX1 正极
177	MIPI1_D0_P	MIPI1 D0 正极
178	HDMI0_TX1_N	输出 HDMI0 TX1 负极
179	GND	接地(0V)
180	GND	接地(0V)
181	MIPI1_D1_N	MIPI1 D1 负极
182	HDMI0_TX0_P	Output HDMI0 TX0 正极
183	MIPI1_D1_P	MIPI1 D1 正极
184	HDMI0_TX0_N	输出 HDMI0 TX0 负极
185	GND	接地(0V)
186	GND	接地(0V)
187	MIPI1_C_N	MIPI1时钟负极
188	HDMI0_CLK_P	输出 HDMI0时钟正极
189	MIPI1_C_P	MIPI1 时钟正极
190	HDMI0_CLK_N	输出HDMI0时钟负极
191	GND	接地(0V)
192	GND	接地(0V)
193	MIPI1_D2_N	MIPI1 D2 负极

194	MIPI1_D3_N	MIPI1 D3 负极
195	MIPI1_D2_P	MIPI1 D2 正极
196	MIPI1_D3_P	MIPI1 D3 正极
197	GND	接地(0V)
198	GND	接地(0V)
199	HDMI0_SDA	双向HDMI0 SDA。内部用1.8kΩ上拉。5V容差。(可以直接连接一个HDMI接口；CM5通过板载HDMI05-CL02F3提供少量ESD保护)
200	HDMI0_SCL	双向HDMI0 SCL。内部用1.8kΩ上拉。5V容差。(可以直接连接一个HDMI接口；CM5通过板载HDMI05-CL02F3提供少量ESD保护)

所有接地引脚都应连接。如果第二个连接器(引脚101至200)上的信号都没有使用，则可以节省该连接器以降低成本，但需要考虑机械稳定性。

如果使用+3.3V信号，GPIO引脚0-27上的电压不得超过**CM5\_3.3V**；如果使用+1.8V信号，则不得超过**CM5\_1.8V**。这些引脚与Raspberry Pi 5上的40-Pin连接器相同。

如果**CM5\_1.8V**供电轨用于为GPIO\_VREF以外的其他器件供电，则应确保在CM5意外断电(例如+5V引脚低于+4.5V)的情况下，**CM5\_1.8V**上的负载必须为零

如果**CM5\_3.3V**供电轨用于为GPIO\_VREF之外的其它器件供电，则应确保在意外断电的情况下，**CM5\_3.3V**供电轨不会低于**CM5\_1.8V**供电轨。这是典型的情况，但是需要在设计中检查。如果其低于**CM5\_1.8V**供电轨，则需要额外的电路来断开**CM5\_3.3V**负载。

不得对任何引脚施加反向电压，否则可能会阻止上电；即在关断/断电期间，任何引脚都不能施加外部电压，否则可能会妨碍后续上电。

## 4.1. 差分对

建议任意差分对对内的P/N信号长度差异小于和等于0.15mm，差分对与差分对之间的匹配并不重要。例如HDMI线差分对与差分对之间的长度差异应小于或者等于25mm，因此在典型电路板上不需要额外匹配。

### 4.1.1. 100Ω差分对信号长度

在CM5上，所有100Ω的差分对，其差分对对内的长度差异小于或者等于0.05mm(P/N信号)。

#### **i** 注意

建议在接口板上也匹配线对。

在CM5上，线对之间并不总是匹配的，因为许多接口不要求线对之间非常精确的匹配。下表记录每个组内的CM5磁道长度差异。(非零值表示与长度差为零的信号相比，轨道长度以mm为单位。)

**表 6 100Ω差分对信号长度**

信号	长度
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差分对信号长度

在CM5上，所有90Ω差分对（包含USB），其差分对对内的长度差异小于或者等于0.01mm(P/N信号)。USB 3 配对不需要端口组中的任何配对。

#### **NOTE**

建议在接口板上也匹配线对。

线对之间并不总是匹配的，因为许多接口不要求线对之间非常精确的匹配。下表记录每个组内的CM5磁道长度差异。(非零值表示与长度差为零的信号相比，轨道长度以mm为单位)

**表 7 90Ω差分对信号长度**

信号	长度
PCle_CLK_P	0.00
PCle_CLK_N	0.01
PCle_TX_P	3.71
PCle_TX_N	3.72
PCle_RX_P	0.84
PCle_RX_N	0.84

# 5 电源

## 5.1. 上电时序控制

CM5需要单通道+5V电源，能以+3.3V和+1.8V向外提供高达600mA的电源。在+5V供电之前，所有引脚都不应加载任何电源。

如果要对EEPROM进行写保护，则EEPROM\_nWP在上电前应处于低电平。

如果要使用USB启动CM5，则RPI\_nBOOT需要在+5V上升的2ms内为低电平。

+5V电压应在整个CM5操作过程中单调上升至4.75V并保持在该电压之上，且其上升速度应快于TBA毫秒（TBA为待确定时间）。

当+5V电压高于4.75V且PMIC\_EN信号上升时，将启动上电序列。事件顺序如下：

1. +5V上升
2. PMIC\_ENABLE 上升
3. CM5\_+3.3V上升
4. 在CM5\_+3.3V后，+1.8V上升至少1ms

## 5.2. 掉电序列

操作系统应在断电前关闭，以确保文件系统保持一致。如果做不到这一点，则应该考虑btrfs、f2fs或overlayfs(使用raspi-config来实现)这样的文件系统。

操作系统关闭后，可以移除+5V供电或拉低PMIC\_EN引脚，使CM4进入最低功耗模式。

在关断序列期间，+1.8V将在+3.3V供电之前放电。

## 5.3. 功耗

CM5的确切功耗将极大地取决于在其上运行的任务。最低关机功耗模式是当PMIC\_EN被拉低时，通常功耗为TBA微安（TBA为待确定数值）。当PMIC\_EN为高电平但软件关闭时，典型功耗为TBA毫安。空闲功耗通常为TBA毫安，但这会根据操作系统的不同而有很大差异。工作功耗通常在TBA毫安左右；同样，这也极大地依赖于操作系统和正在执行的任务。

## 5.4. 调节器输出

为了更容易连接到CM5，板载稳压器(CM5\_+3.3V和 CM5\_+1.8V)可以向连接到CM4的器件提供600mA电流。这些输出端的负载未计入功耗数据。

# 附录A：故障排除

在CPU启动之前，CM5有多个上电阶段。如果任何一个阶段出现错误，将停止上电。

## 硬件清单

1. +5V电源是否良好？将PMIC\_EN拉低，并在+5V电源上添加一个外部2A负载。是否保持>+4.75V(包括噪声)？理想情况下，应保持在+4.9V以上，包括任何噪声。
2. 移除外部2A负载，但保持PMIC\_EN拉低。
3. 检查CM5 +3.3V供电是否小于200mV。如果不是这种情况，则有一个外部电源路径直接或间接地反馈给CM4。这也可以通过数字引脚实现，例如以太网。
4. 仍然在PMIC\_EN拉低的情况下，检查CM5 +1.8V供电是否小于200mV。如果+1.8V供电高于200mV，则有一个外部路径反馈给1.8V供电。(如果这些引脚没有连接任何东西，可以忽略此检查。)
5. 移除PMIC\_EN上的下拉。
6. 检查PMIC\_EN为高电平(在CM5上被内部上拉)。
7. 检查+3.3V电源是否升至+3.15V以上。如果没有，则表明+3.3V供电上的负载过大。
8. 检查+1.8V供电轨是否达到+1.71V以上。如果没有，则表明+1.8V供电上的负载过大。
9. 检查ACT\_LED开始振荡表示开机；检查没有闪烁错误代码。

## 启动程序

1. 连接HDMI电缆，查看HDMI诊断屏幕是否出现。
2. 将USB串口线连接到GPIO引脚14和15。详情请见：[Configuring UARTs](#)
3. 将nRPIBOOT引脚短接至地，强制切换至USB启动模式。CM5IO板有一个nRPIBOOT跳线，可用于启用不同的启动模式(如网络)和使能UART日志。详情请见：[Flashing the Compute Module EMMC](#)

## rpi-eeprom更新

CM5不支持从EMMC(或CM5Lite上的SD卡)运行recovery.bin，故更新启动加载程序EEPROM的唯一方法是通过usbboot或自我更新。

## EEPROM写保护

通过将EEPROM\_nWP短接至地，可以对板上EEPROM进行写保护。CM5IO板有一个跳线用于EEPROM\_nWP。详情请见：[Raspberry Pi boot EEPROM](#)

## 固件

请确保您拥有最新版本的固件，因为新版本中包含了系统改进。

## 内核

请确保您拥有最新版本的内核，因为新版本通常包含安全修复和改进。

# 附录B：测试点

## 测试点

CM5主板上设有多个测试点。这些测试点可以在不使用主100针连接器的情况下为CM5供电和编程。大多数信号都会复制100针连接器上的引脚。

以下是测试点坐标和安装孔位置的表格。坐标是从CM5的正面（或指定面）看过去的。

表8. Raspberry Pi 处理器模块5测试点表

X	Y	名字
51.5	36.5	Mounting Hole
3.5	36.5	Mounting Hole
51.5	3.5	Mounting Hole
3.5	3.5	Mounting Hole
14.34	17.54	+5v
8.8	1.3	RUN
51.2	32.6	GND
4.8	13	reserved
24.2	7.5	GND
1.65	15.05	GND
1.5	10.5	reserved
48.4	15.1	reserved
42.6	7.3	GND
14.7	6.6	reserved
9.3	34.9	nRPIBOOT
37.4	8.1	reserved
23.4	23.55	reserved
24.5125	14.025	RESET_OUT B
13.0875	11.225	reserved
17.7	20.2	GND
43.6	22.3	reserved
15.4	16	reserved
23.65	21.55	reserved
37.2	34.9	reserved
9.1	3.2	reserved

TP32	1.5	13	reserved
TP33	47	36	CM5_3V3
TP34	50.5	15.5	CM5_1V8
TP35	11	37.8	UART_TX
TP36	8.5	37.1	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
------	-------	-------	-----

## 电压输入

两个Vin测试点（**TP1**和**TP78**）都应使用+5V电源。接地必须至少连接到**TP26**、**TP61**和**TP8**。如果可能的话，还应使用其他接地引脚。

## 调试串口

**TP35**和**TP36**分别是调试UART的发送（**TX**）和接收（**RX**）引脚。**TP46**应作为接地引脚使用。在编程和初始启动时，能够访问这些引脚非常有用。

## RPI 启动

**TP65**和**TP66**是USB引脚。必须使用**TP7**的GND（接地）引脚。同时，应将**TP16 nRPI\_BOOT**引脚接地，以强制RPI进入启动模式。

## 以太网启动

**TP69-TP70**应连接到外部磁道插座。

## 附录C：CM4与CM5之间的差异

本节描述了Raspberry Pi处理器模块5（CM5）与之前的Raspberry Pi处理器模块4（CM4）之间的差异。

### Pinout 变化

连接器已更换品牌，并且已经过更高电流的测试以支持CM5。

CAM1信号变为双用途，既可以用于CSI摄像头，也可以用于DSI显示屏。

DSI1信号也变为双用途，既可以用于CSI摄像头，也可以用于DSI显示屏。

CM4在HDMI、SDA、SCL、HPD和CEC信号上增加了额外的静电放电（ESD）保护，但在Raspberry Pi Compute Module 5中已移除这一保护。

CM4上标记为CAM0的端口现在支持USB 3.0端口。标记为DSI0的端口现在也支持USB 3.0端口。CM4上原来的VDAC\_COMP引脚现在变为了两个USB 3.0端口的VBUS使能引脚，并且高电平有效。

CM4上的两个ADC通道现在已经变成了PD CC信号。

Pin	CM4	CM5	备注
16	SYNC_IN	Fan_tacho	Fan tacho 输入
19	Ethernet nLED1	Fan_PWM	Fan PWM 输出
76	Reserved	VBAT	RTC电池。注意，即使CM5已通电，也会有一个几微安的恒定负载。
92	RUN_PG	PWR_Button	复制了Raspberry Pi 5上的电源按钮。短按表示设备应唤醒或关闭。长按则强制关闭。
93	nRPIBOOT	nRPIBOOT	在开机后的一段时间内，如果PWR_button为低电平，则该引脚也将被设置为低电平。
94	AnalogIP1	CC1	此引脚可以连接到Type C USB连接器的CC1线路，以使电源管理集成电路（PMIC）能够协商5A电流。
96	AnalogIP0	CC2	此引脚可以连接到Type C USB连接器的CC2线路，以使电源管理集成电路（PMIC）能够协商5A电流。
99	Global_EN	PMIC_ENABLE	无需外部更改。
100	nEXTRST	CAM_GPIO1	在树莓派计算模块5上被上拉，但可以被强制拉低以模拟RESET信号。

104	Reserved	PCIE_DET_nWAKE	PCIE nWAKE。通过一个8.2K的电阻上拉到CM5_3v3。
106	Reserved	PCIE_PWR_EN	表示PCIe设备是否可以上电或断电的信号。高电平有效。
111	VDAC_COMP	VBUS_EN	输出信号，表示应启用USB VBUS。
128	CAM0_D0_N	USB3-0-RX_N	可能存在正负极（P/N）互换的情况
130	CAM0_D0_P	USB3-0-RX_P	可能存在正负极（P/N）互换的情况
134	CAM0_D1_N	USB3-0-DP	USB 2信号。
136	CAM0_D1_P	USB3-0-DM	USB 2信号。
140	CAM0_C_N	USB3-0-TX_N	可能存在正负极（P/N）互换的情况
142	CAM0_C_P	USB3-0-TX_P	可能存在正负极（P/N）互换的情况
157	DSI0_D0_N	USB3-1-RX_N	可能存在正负极（P/N）互换的情况
159	DSI0_D0_P	USB3-1-RX_P	可能存在正负极（P/N）互换的情况
163	DSI0_D1_N	USB3-1-DP	USB 2信号。
165	DSI0_D1_P	USB3-1-DM	USB 2信号。
169	DSI0_C_N	USB3-1-TX_N	可能存在正负极（P/N）互换的情况
171	DSI0_C_P	USB3-1-TX_P	可能存在正负极（P/N）互换的情况

除了上述内容外，PCIe的CLK信号不再通过电容耦合。

PCB的厚度增加了0.04mm，但主处理器的厚度有所减小。

## 走线长度

HDMI0走线长度已更改。每个P/N对保持匹配，但是对于现有的主板，对之间的偏斜现在小于1毫米。这不太可能造成差异，因为线对之间的偏斜可能在25mm左右。

HDMI1走线长度也发生了变化。每个P/N对保持匹配，但是对于现有主板来说，对之间的偏斜现在小于5毫米。这不太可能造成差异，因为线对之间的偏斜可能在25毫米左右。

以太网轨道长度已经改变。每个P/N对都保持匹配，但对于现有主板，线对之间的偏斜现在小于4毫米。这不可能产生差异，因为线对之间的偏斜可能在12mm的数量级。

## 功率分配

由于CM5比CM4强大得多，所以它消耗更多的功率。电源设计预算应为5V(最大2.5A)。如果这给现有主板设计带来问题，请降低CPU时钟频率以降低峰值功耗。

# 附录D：可用性

## 支持

有关文档，请参见[Raspberry Pi网站的计算模块硬件文档](#)。支持问题可以张贴到[Raspberry Pi论坛](#)。

## 订购代码

表 8 部件选择

型号	无线	RAM LPDDR4	eMMC存储
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
部件示例			
CM5	1	02	032

表 9 订购选项

无线	RAM LPDDR4	存储eMMC	RPL#	部件编码	订单倍数	建议零售价
-	2GB	Lite	SC1156	CM5002000	1+ / Bulk	
-	2GB	16GB	SC1158	CM5002016	1+ / Bulk	
-	2GB	32GB	SC1159	CM5002032	1+ / Bulk	
Yes	2GB	Lite	SC1580	CM5102000	1+ / Bulk	
Yes	2GB	16GB	sC1582	CM5102016	1+ / Bulk	
Yes	2GB	32GB	SC1583	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	
Yes	8GB	16GB	SC1600	CM5108016	1+ / Bulk	
Yes	8GB	32GB	SC1601	CM5108032	1+ / Bulk	

**i 注意**

还可以提供其他选项作为定制订单。

**i 注意**

RRP在发布时是正确的，并且不含税。

## 包装

少量装在单独的纸箱中。这些都有一个内部防静电涂层，因此不需要单独的防静电袋。这种包装是可回收的，减少了浪费。



Raspberry Pi is a trademark of Raspberry Pi Ltd