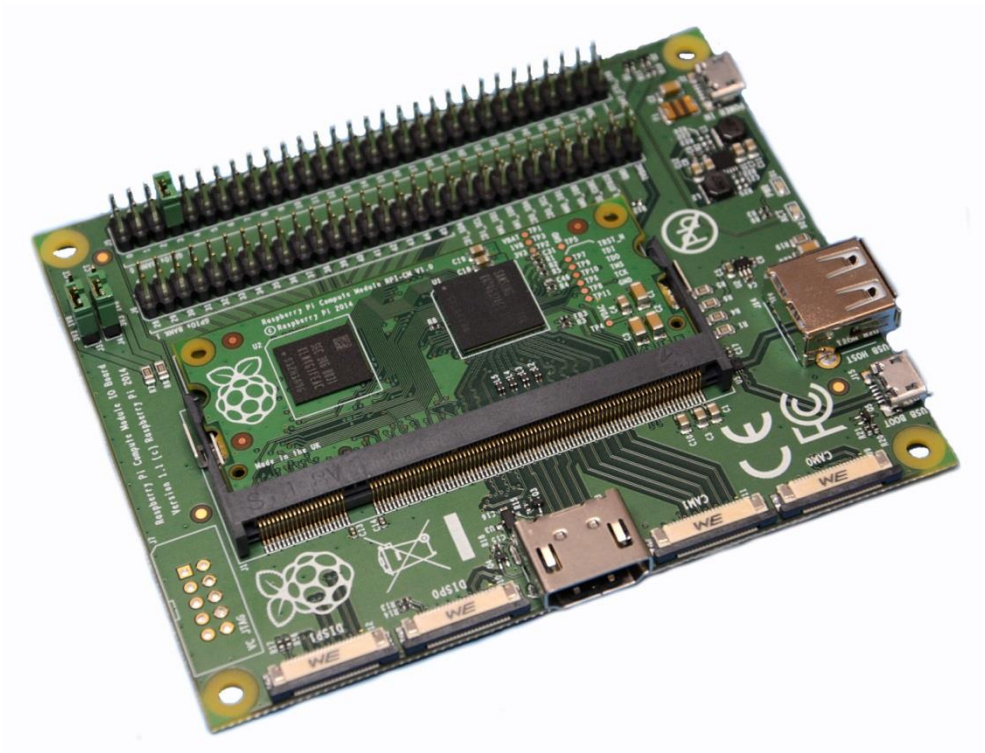
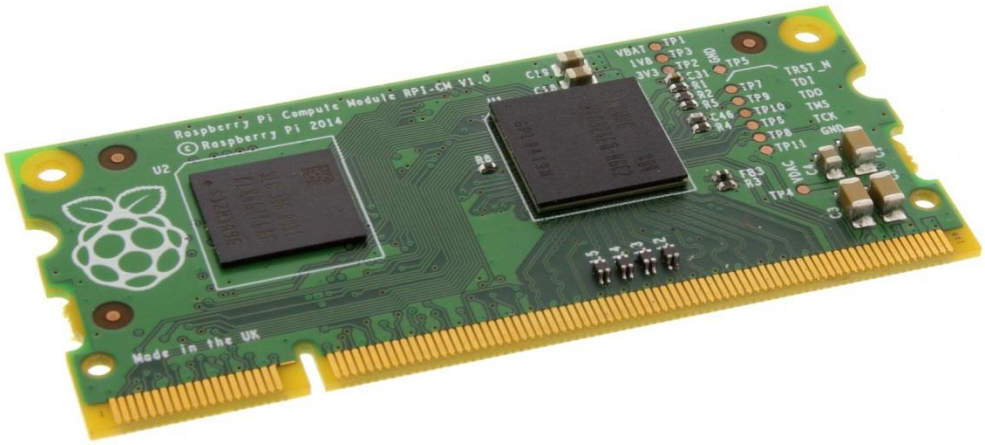
Raspberry Pi Compute Module

Hardware Design Guide

Rev 1.0



Contents



1. Compute Module Hardware Design

1.1. Powering the module

The Compute Module has six separate supplies that must be present and powered at all times; you cannot leave any of them unpowered, even if a specific interface or GPIO bank is unused. The six supplies are as follows:

1. VBAT powers the BCM2835 processor core. It feeds the SMPS that generates the chip core voltage.
2. 3V3 powers various BCM2835 PHYs, IO and the eMMC Flash memory.
3. 1V8 powers various BCM2835 PHYs, IO and SDRAM.
4. VDAC powers the composite (TV-out) DAC.
5. GPIO0-27_VREF powers the GPIO 0-27 IO bank.
6. GPIO28-45_VREF powers the GPIO 28-45 IO bank.

Supply	Voltage / Voltage Range	Tolerance
VBAT	2.3 - 5V [1]	±5%
3V3	3.3V	±5%
1V8	1.8V	±5%
VDAC	2.5 - 2.8V (can connect to 3V3 if unused)	±5%
GPIO0-27_VREF	1.8 - 3.3V	±5%
GPIO28-45_VREF	1.8 - 3.3V	±5%

[1] Note that the voltage range for best SMPS efficiency is ~3.3 - 4.3V.

1.1.1. Power sequencing

Supplies must be synchronised to come up at exactly the same time. Alternatively, they should be staggered so that the highest voltage comes up first, then the remaining voltages in descending order. This is to avoid forward biasing internal (on-chip) diodes between supplies, and causing latch-up.

1.1.2. Power requirements

Exact power requirements will be heavily dependent upon the individual use case. If an on-chip subsystem is unused, it is usually in a low power state or completely turned off. For instance, if your application does not use 3D graphics then a large part of the core digital logic will never turn on and need power. This is also the case for camera and display interfaces, HDMI, USB interfaces, video encoders and decoders, and so on.

Powerchain design is critical for stable and reliable operation of the Compute Module. We strongly recommend that designers spend time measuring and verifying power requirements for their particular application, as well as paying careful attention to power supply sequencing and maximum supply voltage tolerance.

The following table gives a rough guide to minimum supply requirements. **However, the user is responsible for verifying that their powerchain is able to supply sufficient current for their application. In some cases these minimum requirements may well be too low!**

Supply	Minimum Requirement (mA or mW)
VBAT	2000mW [1]
3V3	250mA
1V8	250mA
VDAC	25mA
GPIO0-27_VREF	See note [2]
GPIO28-45_VREF	See note [2]

[1] Note that VBAT is heavily dependent upon the application. For example, with video encoding, 3D and the camera all running the power requirements can be substantial.

[2] Note that each GPIO bank will only need a few mW if unused; however when in use, the requirements will vary depending on the number of active I/O ports and the load on each. The user is responsible for calculating or measuring this based on their particular design.

1.2. Module booting

The 4GB eMMC Flash memory device on the Compute Module is directly connected to the primary BCM2835 SD/eMMC interface. These signals are not accessible from the module edge connector.

When initially powered on, or after the RUN pin has been held low and then released, the BCM2835 will try to access the eMMC device. It will then look for a file called **bootcode.bin** on the primary partition (which must be FAT format) to start booting the system. If it cannot access the eMMC device or the boot code cannot be found, it will fall back to waiting for boot code to be written to it over USB; in other words, its USB port is in slave mode waiting to accept boot code from a suitable host.

A USB boot tool is available on github which allows a host PC running Linux to write the BCM2835 boot code over USB to the module. That boot code then runs and provides access to the eMMC as a USB mass storage device, which can then be read and written using the host PC. Note that a Raspberry Pi can be used as the host machine. **See Section 2 for details of Flash programming the eMMC.**

The Compute Module has a pin called EMMC_DISABLE_N which when shorted to GND will disable the eMMC, forcing the BCM2835 to boot from USB. Note that when the eMMC is disabled in this way, it takes a few seconds from powering up for the processor to stop attempting to talk to the eMMC device and fall back to booting from USB.

Also note that once booted over USB, BCM2835 needs to re-enable the eMMC device (by releasing EMMC_DISABLE_N) to allow access to it as a mass storage device. It expects to be able to do this by driving the GPIO47_1V8 pin LOW, which at boot is initially an input with a pull up to 1V8. If an end user wishes to add the ability to access the eMMC over USB in their product, similar circuitry to that used on the Compute Module IO Board to enable/disable the USB boot and eMMC must be used; that is, EMMC_DISABLE_N pulled low via MOSFET(s) and released again by MOSFET, with the gate controlled by GPIO47_1V8. Ensure you use MOSFETs suitable for switching at 1.8V (i.e. use a device with Vt << 1.8V).

1.3. Compute Module I/O interfaces

1.3.1. GPIO

The GPIO46_1V8 and GPIO47_1V8 pins are 1.8V IO only and are reserved for special functions (HDMI hot plug detect and boot control respectively). Do not use these pins for any other purpose, as the software for the Compute Module will always expect these pins to have these special functions. If they are unused, leave them unconnected.

The remaining GPIOs are available for general use and are split into two banks. GPIO0 to GPIO27 are bank 0 and GPIO28-45 make up bank 1. GPIO0-27_VREF is the power supply for bank 0 and GPIO28-45_VREF is the power supply for bank 1. These supplies can be in the range 1.8 to 3.3V. They are not optional; each bank must be powered, even when none of the GPIOs for that bank are used.

All GPIOs except GPIO28, 29, 44 and 45 have weak in-pad pull-ups or pull-downs enabled when the device is powered on. Whether the GPIO is pulled up or down is documented in the BCM2835 peripherals document section 6.2. **It is recommended to add off-chip pulls to GPIO28, 29, 44 and 45 to make sure they never float during power on and initial boot.**

1.3.2. CSI (MIPI serial camera)

The Compute Module has two MIPI serial camera interfaces (CSI): Interface 0 and Interface 1.

Interface 0 is a 2-lane interface; one clock lane and two data lanes.

Interface 1 is a 4-lane interface; one clock lane and four data lanes.

Note that the Raspberry Pi Model A/B camera connector uses Interface 1, but only in a 2-lane configuration.

The camera interface(s) clock and data pins must be routed as matched-length, matched-phase 100Ω differential PCB traces.

1.3.3. DSI (MIPI serial display)

The Compute Module has 2 MIPI serial display interfaces (DSI): Interface 0 and Interface 1.

Interface 0 is a 2-lane interface; one clock lane and two data lanes.

Interface 1 is a 4-lane interface; one clock lane and four data lanes.

Note that the Raspberry Pi Model A/B display connector uses Interface 1, but only in a 2-lane configuration.

The display interface(s) clock and data pins must be routed as matched length, matched phase 100Ω differential PCB traces.

1.3.4. USB OTG

The BCM2835 USB port is On-The-Go (OTG) capable. When using either as a fixed slave or fixed master, tie the USB_OTGID pin to ground.

The USB port (Pins USB_DP and USB_DM) must be routed as matched-phase 90Ω differential PCB traces.

Note that the port is capable of being used as a true OTG port, but there is currently no documentation, code or examples for this use case.

1.3.5. HDMI

It is recommended that users follow a similar arrangement to the Compute Module IO Board circuitry for HDMI output.

The HDMI CK_P/N (clock) and D0-D3_P/N (data) pins must each be routed as matched length 100Ω differential PCB traces. It is also important to make sure that each differential pair is closely phase matched. Finally, keep HDMI traces well away from other noise sources and as short as possible.

Failure to observe these design rules is likely to result in performance reduction and/or EMC conformance failure.

1.3.6. Composite Video (TVDAC)

The TVDAC pin can be used to output composite video. Route this signal away from noise sources and use a 75Ω PCB trace.

Note that the TV DAC is powered from the VDAC supply which must be a clean supply of 2.5 to 2.8V. It is recommended developers generate this supply from 3V3 using a low noise LDO regulator chip.

If the TVDAC output is not used, VDAC can be connected to 3V3, but it must be powered even if the TV-out functionality is unused.

1.4. Compute Module temperature range

The operating temperature range of the module is set by the lowest maximum and highest minimum of any of the components.

The Samsung eMMC and Samsung LPDDR2 RAM are both rated for -25 to +80°C, setting the range for the whole module at -25 to +80°C. The BCM2835 processor and the analogue switch have a greater range. The ceramic capacitors are specified from -25 to +85°C.

However, this range is the maximum for the silicon die; therefore developers should take into account the heat generated when in use and make sure this does not cause the device temperature to exceed 80°C.

The developer is responsible for designing and testing their system so that these limits are not exceeded.

1.5. Compute Module form factor

The Compute Module conforms to JEDEC MO-224 mechanical specification for 200 pin DDR2 (1.8V) SODIMM modules. Please note that the pinout of the Compute Module is not the same as for a DDR2 SODIMM memory module; they are **not** electrically compatible.

The maximum component height on the underside of the Compute Module is 1.2mm.

The maximum component height on the top side of the Compute Module is 1.5mm.

The Compute Module PCB thickness is 1.0mm ±10%.

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

2. Flashing the Compute Module eMMC

The Compute module has an on-board eMMC Flash memory device connected to the primary SD card interface. This guide explains how to write data to the eMMC storage using a Compute Module IO Board.

2.1. Steps to Flash the eMMC on a Compute Module

You need a host Linux system; a Raspberry Pi will do.

2.1.1. On your Compute Module IO Board:

Make sure that J4 (USB SLAVE BOOT ENABLE) is set to the 'EN' position.

2.1.2. On your host system:

Git may produce an error if the date/time is not set correctly, so on a Raspberry Pi enter the following:

sudo date MMDDhhmm

where MM is month, DD day and hh mm hours and minutes respectively.

Clone the usbboot tool repository and install libusb:

```
sudo git clone --depth=1 https://github.com/raspberrypi/tools
cd tools/usbboot
sudo apt-get install libusb-1.0-0-dev
```

Build the usbboot tool:

sudo make

Run the usbboot tool and it will wait for a connection:

sudo ./rpiboot

Now plug the host machine into the Compute Module IO Board USB slave port (J15) and power on the CMIO board. The usbboot tool will discover the Compute Module and send boot code to allow access to the eMMC. Once complete you will see a new device appear; this is commonly /dev/sda but it could be another location such as /dev/sdb, so check in /dev/ before running rpiboot so you can see what changes.

You now need to write a raw OS image (such as Raspbian) to the device. Note the following command may take some time to complete, depending on the size of the image:

sudo dd if=raw_os_image_of_your_choice.img of=/dev/sda bs=4MiB

Once the image has been written, unplug and re-plug the USB; you should see 2 partitions appear (for Raspian) in /dev. In total you should see something similar to this:

```
/dev/sda    <- Device
/dev/sda1   <- First partition (FAT)
/dev/sda2   <- Second partition (Linux filesystem)
```

The /dev/sda1 and /dev/sda2 partitions can now be mounted normally.

Make sure J4 (USB SLAVE BOOT ENABLE) is set to the disabled position and/or nothing is plugged into the USB slave port. Power cycling the IO board should now result in the Compute Module booting from eMMC.