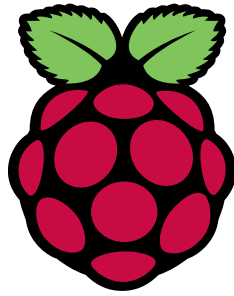


DATASHEET



Raspberry Pi Compute Module (CM1)

Raspberry Pi Compute Module 3 (CM3)

Raspberry Pi Compute Module 3 Lite (CM3L)

Version 1.0, October 2016

Copyright 2016 Raspberry Pi (Trading) Ltd. All rights reserved.



Table 1: Revision History

Revision	Date	Description
1.0	13/10/2016	First release



Contents

1	Introduction	5
2	Features	6
2.1	Hardware	6
2.2	Peripherals	6
2.3	Software	6
3	Block Diagram	7
4	Mechanical Specification	9
5	Pin Assignments	11
6	Electrical Specification	13
7	Power Supplies	14
7.1	Supply Sequencing	15
7.2	Power Requirements	15
8	Booting	16
9	Peripherals	17
9.1	GPIO	17
9.1.1	GPIO Alternate Functions	18
9.1.2	Secondary Memory Interface (SMI)	19
9.1.3	Display Parallel Interface (DPI)	19
9.1.4	SD/SDIO Interface	20
9.2	CSI (MIPI Serial Camera)	20
9.3	DSI (MIPI Serial Display)	20
9.4	USB	20
9.5	HDMI	20
9.6	Composite (TV Out)	21
10	Thermals	21
10.1	Temperature Range	21
11	Availability	21
12	Support	21



List of Figures

1	CM1 Block Diagram	7
2	CM3/CM3L Block Diagram	8
3	CM1 Mechanical Dimensions	9
4	CM3 and CM3L Mechanical Dimensions	10
5	Digital IO Characteristics	14



List of Tables

1	Revision History	1
2	Compute Module SODIMM Connector Pinout	11
3	Pin Functions	12
4	Absolute Maximum Ratings	13
5	DC Characteristics	13
6	Digital I/O Pin AC Characteristics	14
7	Power Supply Operating Ranges	15
8	Mimimum Power Supply Requirements	16
9	GPIO Bank0 Alternate Functions	18
10	GPIO Bank1 Alternate Functions	19



1 Introduction

The Raspberry Pi Compute Module (CM1), Compute Module 3 (CM3) and Compute Module 3 Lite (CM3L) are DDR2-SODIMM-mechanically-compatible System on Modules (SoMs) containing processor, memory, eMMC Flash (for CM1 and CM3) 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 module have extra IO interfaces over and above what is available on the Raspberry Pi model A/B boards opening up more options for the designer.

The CM1 contains a BCM2835 processor (as used on the original Raspberry Pi and Raspberry Pi B+ models), 512MByte LPDDR2 RAM and 4Gbytes eMMC Flash. The CM3 contains a BCM2837 processor (as used on the Raspberry Pi 3), 1Gbyte LPDDR2 RAM and 4Gbytes eMMC Flash. Finally the CM3L product is the same as CM3 except the eMMC Flash is not fitted, and the SD/eMMC interface pins are available for the user to connect their own SD/eMMC device.

Note that the BCM2837 processor is an evolution of the BCM2835 processor. The only real differences are that the BCM2837 can address more RAM (up to 1Gbyte) and the ARM CPU complex has been upgraded from a single core ARM11 in BCM2835 to a Quad core Cortex A53 with dedicated 512Kbyte L2 cache in BCM2837. All IO interfaces and peripherals stay the same and hence the two chips are largely software and hardware compatible.

The pinout of CM1 and CM3 are identical. Apart from the CPU upgrade and increase in RAM the other significant hardware differences to be aware of are that CM3 has grown from 30mm to 31mm in height, the VBAT supply can now draw significantly more power under heavy CPU load, and the HDMI_HPD_N_1V8 (GPIO46_1V8 on CM1) and EMMC_EN_N_1V8 (GPIO47_1V8 on CM1) are now driven from an IO expander rather than the processor. If a designer of a CM1 product has a suitably specified VBAT, can accommodate the extra 1mm module height increase and has followed the design rules with respect to GPIO46_1V8 and GPIO47_1V8 then a CM3 should work fine in a board designed for a CM1.



2 Features

2.1 Hardware

- Low cost
- Low power
- High availability
- High reliability
 - Tested over millions of Raspberry Pis Produced to date
 - Module IO pins have 35u hard gold plating

2.2 Peripherals

- 48x GPIO
- 2x I2C
- 2x SPI
- 2x UART
- 2x SD/SDIO
- 1x HDMI 1.3a
- 1x USB2 HOST/OTG
- 1x DPI (Parallel RGB Display)
- 1x NAND interface (SMI)
- 1x 4-lane CSI Camera Interface (up to 1Gbps per lane)
- 1x 2-lane CSI Camera Interface (up to 1Gbps per lane)
- 1x 4-lane DSI Display Interface (up to 1Gbps per lane)
- 1x 2-lane DSI Display Interface (up to 1Gbps per lane)

2.3 Software

- ARMv6 (CM1) or ARMv7 (CM3, CM3L) Instruction Set
- Mature and stable Linux software stack
 - Latest Linux Kernel support
 - Many drivers upstreamed
 - Stable and well supported userland
 - Full availability of GPU functions using standard APIs



3 Block Diagram

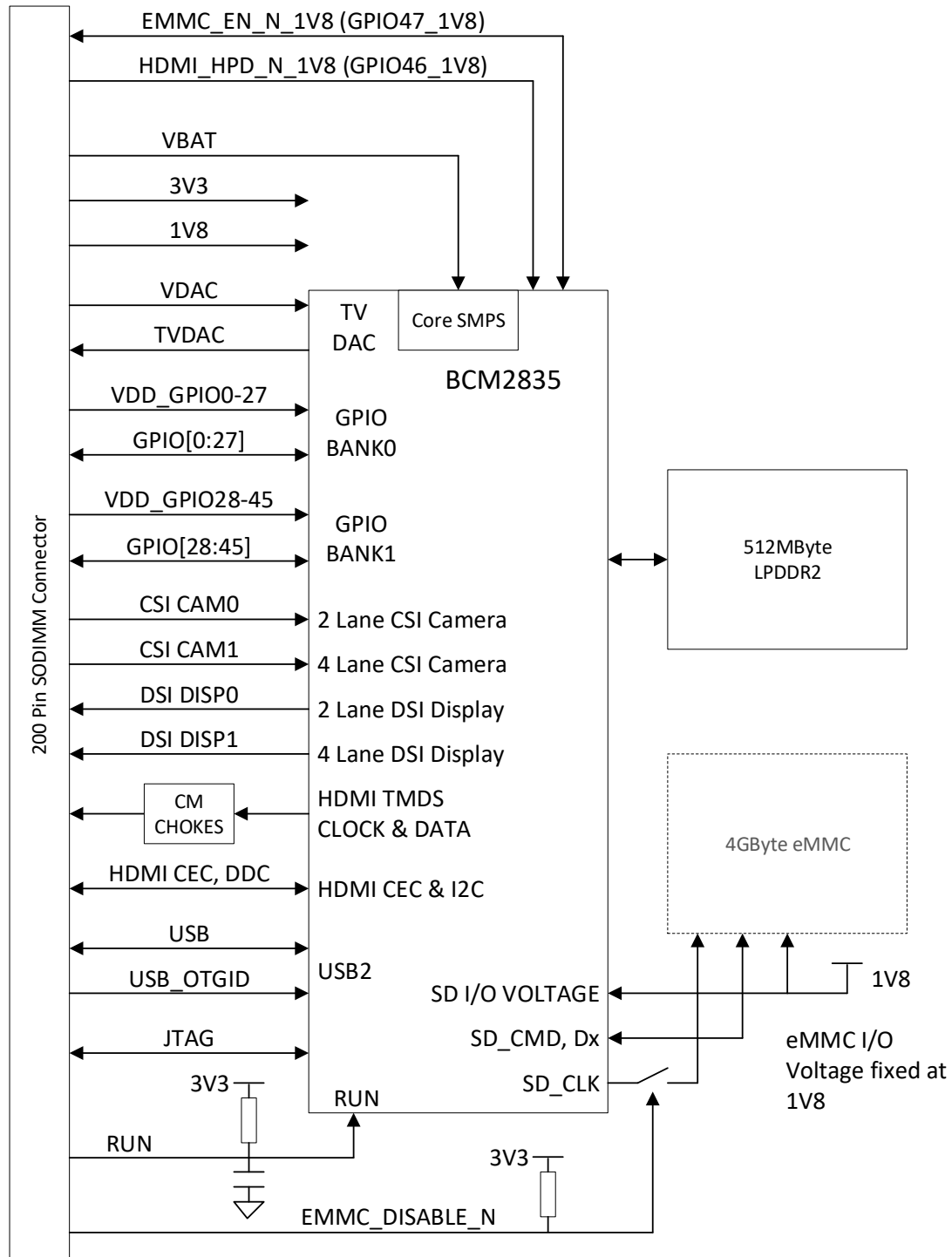


Figure 1: CM1 Block Diagram

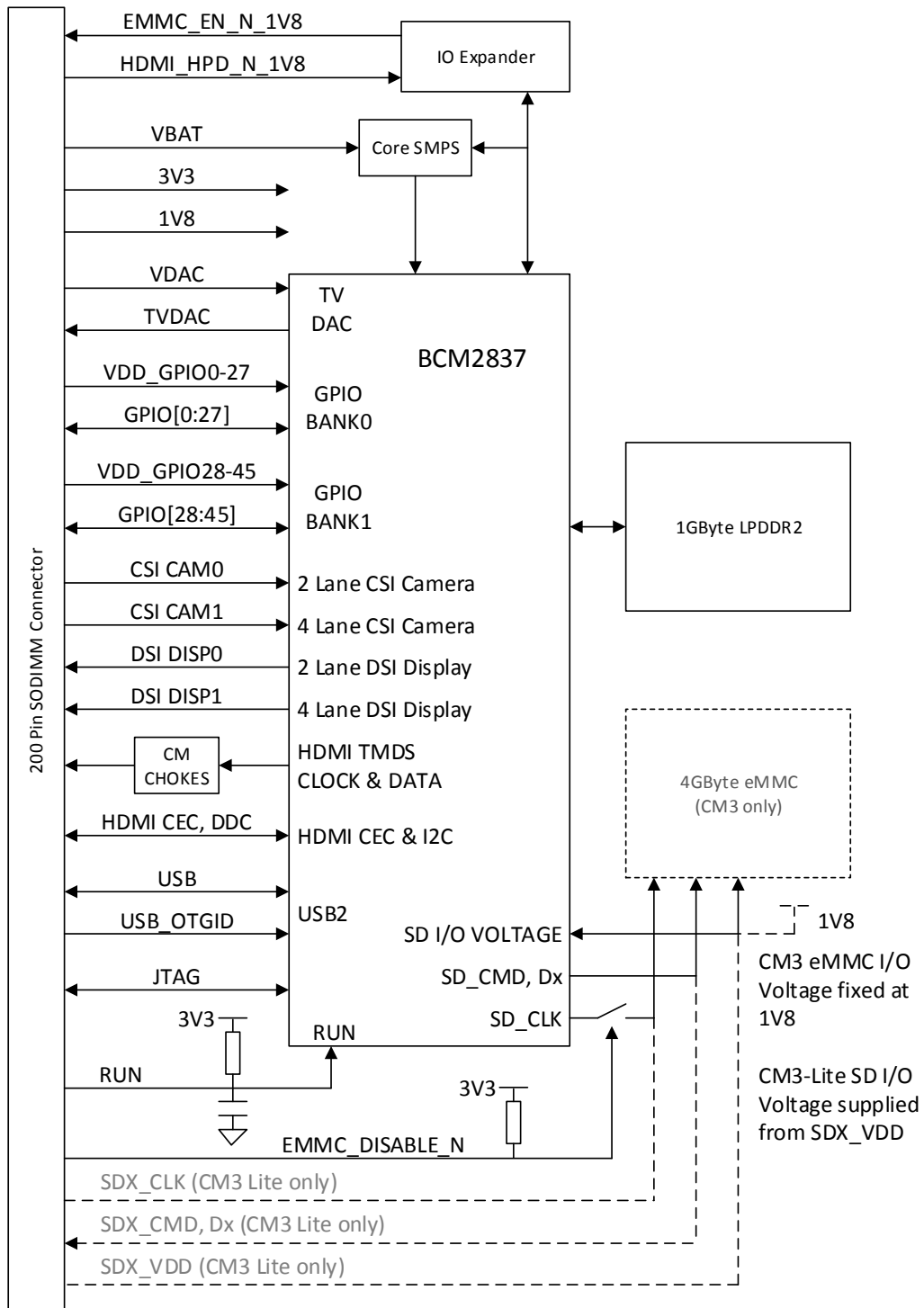


Figure 2: CM3/CM3L Block Diagram



4 Mechanical Specification

The Compute Modules conform to JEDEC MO-224 mechanical specification for 200 pin DDR2 (1.8V) SODIMM modules (with the exception that the CM3, CM3L modules are 31mm in height rather than 30mm of CM1) and therefore should work with the many DDR2 SODIMM sockets available on the market. **(Please note that the pinout of the Compute Module is not the same as a DDR2 SODIMM module; they are not electrically compatible.)**

The SODIMM form factor was chosen as a way to provide the 200 pin connections using a standard, readily available and low cost connector compatible with low cost PCB manufacture.

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 +/- 0.1mm.

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.

Figure 3 gives the CM1 mechanical dimensions. Figure 4 gives the CM3 and CM3L mechanical dimensions.

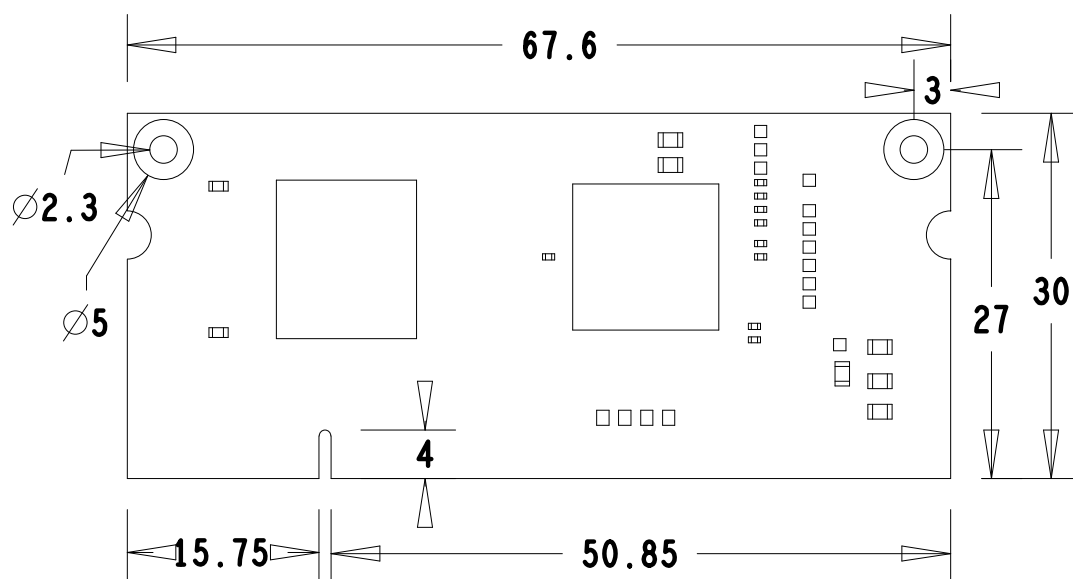


Figure 3: CM1 Mechanical Dimensions

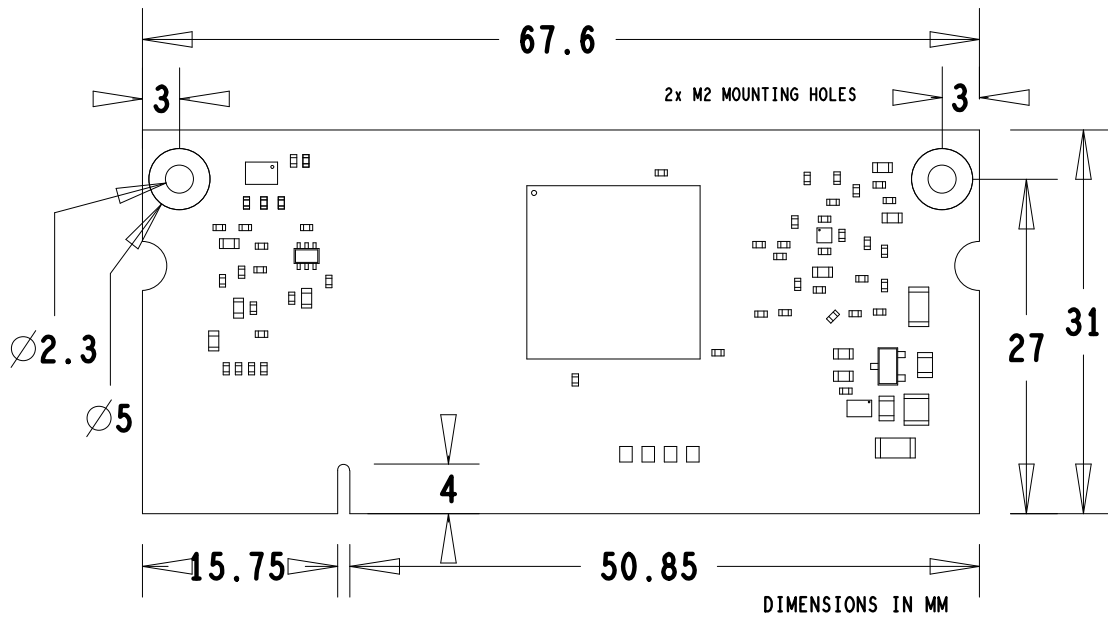


Figure 4: CM3 and CM3L Mechanical Dimensions



5 Pin Assignments

CM1	CM3-Lite	CM3	PIN	PIN	CM3	CM3-Lite	CM1
	GND		1	2	EMMC_DISABLE_N		
	GPIO0		3	4	NC	SDX_VDD	NC
	GPIO1		5	6	NC	SDX_VDD	NC
	GND		7	8	GND		NC
	GPIO2		9	10	NC	SDX_CLK	NC
	GPIO3		11	12	NC	SDX_CMD	NC
	GND		13	14	GND		NC
	GPIO4		15	16	NC	SDX_D0	NC
	GPIO5		17	18	NC	SDX_D1	NC
	GND		19	20	GND		NC
	GPIO6		21	22	NC	SDX_D2	NC
	GPIO7		23	24	NC	SDX_D3	NC
	GND		25	26	GND		
	GPIO8		27	28	GPIO28		
	GPIO9		29	30	GPIO29		
	GND		31	32	GND		
	GPIO10		33	34	GPIO30		
	GPIO11		35	36	GPIO31		
	GND		37	38	GND		
	GPIO0-27_VDD		39	40	GPIO0-27_VDD		
					KEY		
	GPIO28-45_VDD		41	42	GPIO28-45_VDD		
	GND		43	44	GND		
	GPIO12		45	46	GPIO32		
	GPIO13		47	48	GPIO33		
	GND		49	50	GND		
	GPIO14		51	52	GPIO34		
	GPIO15		53	54	GPIO35		
	GND		55	56	GND		
	GPIO16		57	58	GPIO36		
	GPIO17		59	60	GPIO37		
	GND		61	62	GND		
	GPIO18		63	64	GPIO38		
	GPIO19		65	66	GPIO39		
	GND		67	68	GND		
	GPIO20		69	70	GPIO40		
	GPIO21		71	72	GPIO41		
	GND		73	74	GND		
	GPIO22		75	76	GPIO42		
	GPIO23		77	78	GPIO43		
	GND		79	80	GND		
	GPIO24		81	82	GPIO44		
	GPIO25		83	84	GPIO45		
	GND		85	86	GND		
	GPIO26		87	88	HDMI_HPD_N_1V8	GPIO46_1V8	
	GPIO27		89	90	EMMC_EN_N_1V8	GPIO47_1V8	
	GND		91	92	GND		
	DSIO_DN1		93	94	DSI1_DP0		
	DSIO_DP1		95	96	DSI1_DN0		
	GND		97	98	GND		
	DSIO_DN0		99	100	DSI1_CP		
	DSIO_DP0		101	102	DSI1_CN		
	GND		103	104	GND		
	DSIO_CN		105	106	DSI1_DP3		
	DSIO_CP		107	108	DSI1_DN3		
	GND		109	110	GND		
	HDMI_CLK_N		111	112	DSI1_DP2		
	HDMI_CLK_P		113	114	DSI1_DN2		
	GND		115	116	GND		
	HDMI_D0_N		117	118	DSI1_DP1		
	HDMI_D0_P		119	120	DSI1_DN1		
	GND		121	122	GND		
	HDMI_D1_N		123	124	NC		
	HDMI_D1_P		125	126	NC		
	GND		127	128	NC		
	HDMI_D2_N		129	130	NC		
	HDMI_D2_P		131	132	NC		
	GND		133	134	GND		
	CAM1_DP3		135	136	CAM0_DP0		
	CAM1_DN3		137	138	CAM0_DN0		
	GND		139	140	GND		
	CAM1_DP2		141	142	CAM0_CP		
	CAM1_DN2		143	144	CAM0_CN		
	GND		145	146	GND		
	CAM1_CP		147	148	CAM0_DP1		
	CAM1_CN		149	150	CAM0_DN1		
	GND		151	152	GND		
	CAM1_DP1		153	154	NC		
	CAM1_DN1		155	156	NC		
	GND		157	158	NC		
	CAM1_DP0		159	160	NC		
	CAM1_DN0		161	162	NC		
	GND		163	164	GND		
	USB_DP		165	166	TVDAC		
	USB_DM		167	168	USB_OTGID		
	GND		169	170	GND		
	HDMI_CEC		171	172	VC_TRST_N		
	HDMI_SDA		173	174	VC_TDI		
	HDMI_SCL		175	176	VC_TMS		
	RUN		177	178	VC_TDO		
	VDD_CORE (DO NOT CONNECT)		179	180	VC_TCK		
	GND		181	182	GND		
	1V8		183	184	1V8		
	1V8		185	186	1V8		
	GND		187	188	GND		
	VDAC		189	190	VDAC		
	3V3		191	192	3V3		
	3V3		193	194	3V3		
	GND		195	196	GND		
	VBAT		197	198	VBAT		
	VBAT		199	200	VBAT		

Table 2: Compute Module SODIMM Connector Pinout

Table 2 gives the Compute Module pinout and Table 3 gives the Compute Module pin functions.



Pin Name	DIR	Voltage Ref	PDN ^a State	If Unused	Description/Notes
<i>RUN and Boot Control (see text for usage guide)</i>					
RUN	I	3V3 ^b	Pull High	Leave open	Has internal 10k pull up
EMMC_DISABLE_N	I	3V3 ^b	Pull High	Leave open	Has internal 10k pull up
EMMC_EN_N_1V8	O	1V8	Pull High	Leave open	Has internal 2k2 pull up
<i>GPIO</i>					
GPIO[27:0]	I/O	GPIO0-27_VDD	Pull or Hi-Z ^c	Leave open	GPIO Bank 0
GPIO[45:28]	I/O	GPIO28-45_VDD	Pull or Hi-Z ^c	Leave open	GPIO Bank 1
<i>Primary SD Interface^{d,e}</i>					
SDX_CLK	O	SDX_VDD	Pull High	Leave open	Primary SD interface CLK
SDX_CMD	I/O	SDX_VDD	Pull High	Leave open	Primary SD interface CMD
SDX_Dx	I/O	SDX_VDD	Pull High	Leave open	Primary SD interface DATA
<i>USB Interface</i>					
USB_Dx	I/O	-	Z	Leave open	Serial interface
USB_OTGID	I	3V3		Tie to GND	OTG pin detect
<i>HDMI Interface</i>					
HDMI_SCL	I/O	3V3 ^b	Z ^f	Leave open	DDC Clock (5.5V tolerant)
HDMI_SDA	I/O	3V3 ^b	Z ^f	Leave open	DDC Data (5.5V tolerant)
HDMI_CEC	I/O	3V3	Z	Leave open	CEC (has internal 27k pull up)
HDMI_CLKx	O	-	Z	Leave open	HDMI serial clock
HDMI_Dx	O	-	Z	Leave open	HDMI serial data
HDMI_HPD_N_1V8	I	1V8	Pull High	Leave open	HDMI hotplug detect
<i>CAM0 (CSI0) 2-lane Interface</i>					
CAM0_Cx	I	-	Z	Leave open	Serial clock
CAM0_Dx	I	-	Z	Leave open	Serial data
<i>CAM1 (CSI1) 4-lane Interface</i>					
CAM1_Cx	I	-	Z	Leave open	Serial clock
CAM1_Dx	I	-	Z	Leave open	Serial data
<i>DSI0 (Display 0) 2-lane Interface</i>					
DSI0_Cx	O	-	Z	Leave open	Serial clock
DSI0_Dx	O	-	Z	Leave open	Serial data
<i>DSI1 (Display 1) 4-lane Interface</i>					
DSI1_Cx	O	-	Z	Leave open	Serial clock
DSI1_Dx	O	-	Z	Leave open	Serial data
<i>TV Out</i>					
TVDAC	O	-	Z	Leave open	Composite video DAC output
<i>JTAG Interface</i>					
TMS	I	3V3	Z	Leave open	Has internal 50k pull up
TRST_N	I	3V3	Z	Leave open	Has internal 50k pull up
TCK	I	3V3	Z	Leave open	Has internal 50k pull up
TDI	I	3V3	Z	Leave open	Has internal 50k pull up
TDO	O	3V3	O	Leave open	Has internal 50k pull up

^a The PDN column indicates power-down state (when RUN pin LOW)

^b Must be driven by an open-collector driver

^c GPIO have software enabled pulls which keep state over power-down

^d Only available on Lite variants

^e The CM will always try to boot from this interface first

^f Requires external pull-up resistor to 5V as per HDMI spec

Table 3: Pin Functions



6 Electrical Specification

Caution! Stresses above those listed in Table 4 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.

Symbol	Parameter	Minimum	Maximum	Unit
V _{BAT}	Core SMPS Supply	-0.5	6.0	V
3V3	3V3 Supply Voltage	-0.5	4.10	V
1V8	1V8 Supply Voltage	-0.5	2.10	V
V _{DAC}	TV DAC Supply	-0.5	4.10	V
GPIO0-27_VDD	GPIO0-27 I/O Supply Voltage	-0.5	4.10	V
GPIO28-45_VDD	GPIO28-27 I/O Supply Voltage	-0.5	4.10	V
SDX_VDD	Primary SD/eMMC Supply Voltage	-0.5	4.10	V

Table 4: Absolute Maximum Ratings

DC Characteristics are defined in Table 5

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Unit
V_{IL}	Input low voltage ^a	VDD_IO = 1.8V	-	-	0.6	V
		VDD_IO = 2.7V	-	-	0.8	V
V_{IH}	Input high voltage ^a	VDD_IO = 1.8V	1.0	-	-	V
		VDD_IO = 2.7V	1.3	-	-	V
I_{IL}	Input leakage current	TA = +85°C	-	-	5	μA
C_{IN}	Input capacitance	-	-	5	-	pF
V_{OL}	Output low voltage ^b	VDD_IO = 1.8V, I _{OL} = -2mA	-	-	0.2	V
		VDD_IO = 2.7V, I _{OL} = -2mA	-	-	0.15	V
V_{OH}	Output high voltage ^b	VDD_IO = 1.8V, I _{OH} = 2mA	1.6	-	-	V
		VDD_IO = 2.7V, I _{OH} = 2mA	2.5	-	-	V
I_{OL}	Output low current ^c	VDD_IO = 1.8V, V _O = 0.4V	12	-	-	mA
		VDD_IO = 2.7V, V _O = 0.4V	17	-	-	mA
I_{OH}	Output high current ^c	VDD_IO = 1.8V, V _O = 1.4V	10	-	-	mA
		VDD_IO = 2.7V, V _O = 2.3V	16	-	-	mA
R_{PU}	Pullup resistor	-	50	-	65	kΩ
R_{PD}	Pulldown resistor	-	50	-	65	kΩ

^a Hysteresis enabled

^b Default drive strength (8mA)

^c Maximum drive strength (16mA)

Table 5: DC Characteristics



AC Characteristics are defined in Table 6 and Fig. 5.

Pin Name	Symbol	Parameter	Minimum	Typical	Maximum	Unit
Digital outputs	t_{rise}	10-90% rise time ^a	-	1.6	-	ns
Digital outputs	t_{fall}	90-10% fall time ^a	-	1.7	-	ns
GPCLK	t_{JOSC}	Oscillator-derived GPCLK cycle-cycle jitter (RMS)	-	-	20	ps
GPCLK	t_{JPLL}	PLL-derived GPCLK cycle-cycle jitter (RMS)	-	-	48	ps

^a Default drive strength, CL = 5pF, VDD_IOx = 3.3V

Table 6: Digital I/O Pin AC Characteristics

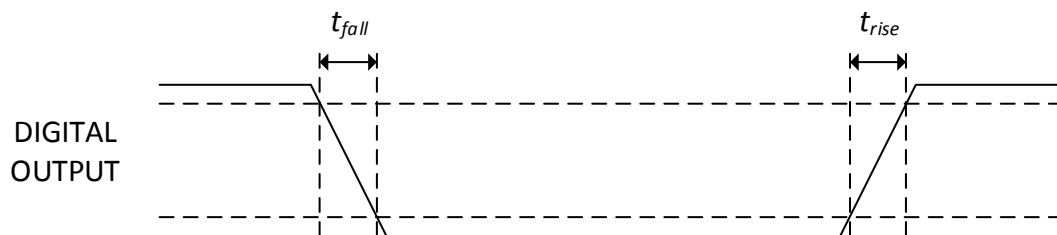


Figure 5: Digital IO Characteristics

7 Power Supplies

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 is used to power the BCM283x processor core. It feeds the SMPS that generates the chip core voltage.
2. 3V3 powers various BCM283x PHYs, IO and the eMMC Flash.
3. 1V8 powers various BCM283x 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	Description	Minimum	Typical	Maximum	Unit
VBAT	Core SMPS Supply	2.5	-	5.0 + 5%	V
3V3	3V3 Supply Voltage	3.3 - 5%	3.3	3.3 + 5%	V
1V8	1V8 Supply Voltage	1.8 - 5%	1.8	1.8 + 5%	V
VDAC	TV DAC Supply ^a	2.5 - 5%	2.8	3.3 + 5%	V
GPIO0-27_VDD	GPIO0-27 I/O Supply Voltage	1.8 - 5%	-	3.3 + 5%	V
GPIO28-45_VDD	GPIO28-27 I/O Supply Voltage	1.8 - 5%	-	3.3 + 5%	V
SDX_VDD	Primary SD/eMMC Supply Voltage	1.8 - 5%	-	3.3 + 5%	V

^a Requires a clean 2.5-2.8V supply if TV DAC is used, else connect to 3V3

Table 7: Power Supply Operating Ranges

7.1 Supply Sequencing

Supplies 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. Alternatively supplies can be synchronised to come up at exactly the same time as long as at no point a lower voltage supply rail voltage exceeds a higher voltage supply rail voltage.

7.2 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 use case and application, as well as paying careful attention to power supply sequencing and maximum supply voltage tolerance.

Table 8 specifies the recommended minimum power supply outputs required to power the Compute Module.



Supply	Minimum Requirement	Unit
VBAT (CM1)	2000 ^a	mW
VBAT (CM3,3L)	3500 ^a	mW
3V3	250	mA
1V8	250	mA
VDAC	25	mA
GPIO0-27_VDD	50 ^b	mA
GPIO28-45_VDD	50 ^b	mA
SDX_VDD	50 ^b	mA

^a Recommended minimum. Actual power drawn is very dependent on use-case

^b Each GPIO can supply up to 16mA, aggregate current per bank must not exceed 50mA

Table 8: Minimum Power Supply Requirements

8 Booting

The 4GB eMMC Flash device on CM3 is directly connected to the primary BCM2837 SD/eMMC interface. These connections are not accessible on the module pins. On CM3L this SD interface is available on the SDX_ pins.

When initially powered on, or after the RUN pin has been held low and then released, the BCM2837 will try to access the primary SD/eMMC interface. It will then look for a file called bootcode.bin on the primary partition (which must be FAT) to start booting the system. If it cannot access the SD/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 BCM2837 boot code over USB to the module. That boot code then runs and provides access to the SD/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. For those using Windows a precompiled and packaged tool is available. For more information see [here](#).

The Compute Module has a pin called EMMC_DISABLE_N which when shorted to GND will disable the SD/eMMC interface (by physically disconnecting the SD_CMD pin), forcing BCM2837 to boot from USB. Note that when the eMMC is disabled in this way, it takes a couple of seconds from powering up for the processor to stop attempting to talk to the SD/eMMC device and fall back to booting from USB.

Note that once booted over USB, BCM2837 needs to re-enable the SD/eMMC device (by releasing EMMC_DISABLE_N) to allow access to it as mass storage. It expects to be able to do this by driving the EMMC_EN_N_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 SD/eMMC over USB in their product, similar circuitry to that used on the Compute Module IO Board to enable/disable the USB boot and SD/eMMC must be used; that is, EMMC_DISABLE_N pulled low via MOSFET(s) and released again by MOSFET, with the gate controlled by EMMC_EN_N_1V8. Ensure you use MOSFETs suitable for switching at 1.8V (i.e. use a device with gate threshold voltage, V_t , suitable for 1.8V switching).



9 Peripherals

9.1 GPIO

BCM283x has in total 54 GPIO lines in 3 separate voltage banks. All GPIO pins have at least two alternative functions within the SoC. When not used for the alternate peripheral function, each GPIO pin may be set as an input (optionally as an interrupt) or an output. The alternate functions are usually peripheral I/Os, and most peripherals appear twice to allow flexibility on the choice of I/O voltage.

On CM1, CM3 and CM3L bank2 is used on the module to connect to the eMMC device and, on CM3 and CM3L, for an on-board I2C bus (to talk to the core SMPS and control the special function pins). On CM3L most of bank 2 is exposed to allow a user to connect their choice of SD card or eMMC device (if required).

Bank0 and 1 GPIOs are available for general use. GPIO0 to GPIO27 are bank 0 and GPIO28-45 make up bank1. GPIO0-27_VDD is the power supply for bank0 and GPIO28-45_VDD is the power supply for bank1. SDX_VDD is the supply for bank2 on CM3L. These supplies can be in the range 1.8V-3.3V (see Table 7) and are not optional; each bank must be powered, even when none of the GPIOs for that bank are used.

Note that the HDMI_HPD_N_1V8 and EMMC_EN_N_1V8 pins (on CM1 these were called GPIO46_1V8 and GPIO47_1V8 respectively) are 1.8V IO and are used for special functions (HDMI hot plug detect and boot control respectively). Please 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 please leave them unconnected.

All GPIOs except GPIO28, 29, 44 and 45 have weak in-pad pull-ups or pull-downs enabled when the device is powered on. 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.



9.1.1 GPIO Alternate Functions

GPIO	Default						
	Pull	ALT0	ALT1	ALT2	ALT3	ALT4	ALT5
0	High	SDA0	SA5	PCLK	-	-	-
1	High	SCL0	SA4	DE	-	-	-
2	High	SDA1	SA3	LCD_VSYNC	-	-	-
3	High	SCL1	SA2	LCD_HSYNC	-	-	-
4	High	GPCLK0	SA1	DPI.D0	-	-	ARM_TDI
5	High	GPCLK1	SA0	DPI.D1	-	-	ARM_TDO
6	High	GPCLK2	SOE_N	DPI.D2	-	-	ARM_RTCK
7	High	SPI0_CE1_N	SWE_N	DPI.D3	-	-	-
8	High	SPI0_CE0_N	SD0	DPI.D4	-	-	-
9	Low	SPI0_MISO	SD1	DPI.D5	-	-	-
10	Low	SPI0_MOSI	SD2	DPI.D6	-	-	-
11	Low	SPI0_SCLK	SD3	DPI.D7	-	-	-
12	Low	PWM0	SD4	DPI.D8	-	-	ARM_TMS
13	Low	PWM1	SD5	DPI.D9	-	-	ARM_TCK
14	Low	TXD0	SD6	DPI.D10	-	-	TXD1
15	Low	RXD0	SD7	DPI.D11	-	-	RXD1
16	Low	FL0	SD8	DPI.D12	CTS0	SPI1_CE2_N	CTS1
17	Low	FL1	SD9	DPI.D13	RTS0	SPI1_CE1_N	RTS1
18	Low	PCM_CLK	SD10	DPI.D14	-	SPI1_CE0_N	PWM0
19	Low	PCM_FS	SD11	DPI.D15	-	SPI1_MISO	PWM1
20	Low	PCM_DIN	SD12	DPI.D16	-	SPI1_MOSI	GPCLK0
21	Low	PCM_DOUT	SD13	DPI.D17	-	SPI1_SCLK	GPCLK1
22	Low	SD0_CLK	SD14	DPI.D18	SD1_CLK	ARM_TRST	-
23	Low	SD0_CMD	SD15	DPI.D19	SD1_CMD	ARM_RTCK	-
24	Low	SD0_DAT0	SD16	DPI.D20	SD1_DAT0	ARM_TDO	-
25	Low	SD0_DAT1	SD17	DPI.D21	SD1_DAT1	ARM_TCK	-
26	Low	SD0_DAT2	TE0	DPI.D22	SD1_DAT2	ARM_TDI	-
27	Low	SD0_DAT3	TE1	DPI.D23	SD1_DAT3	ARM_TMS	-

Table 9: GPIO Bank0 Alternate Functions



GPIO	Default						
	Pull	ALT0	ALT1	ALT2	ALT3	ALT4	ALT5
28	None	SDA0	SA5	PCM_CLK	FL0	-	-
29	None	SCL0	SA4	PCM_FS	FL1	-	-
30	Low	TE0	SA3	PCM_DIN	CTS0	-	CTS1
31	Low	FL0	SA2	PCM_DOUT	RTS0	-	RTS1
32	Low	GPCLK0	SA1	RING_OCLK	TXD0	-	TXD1
33	Low	FL1	SA0	TE1	RXD0	-	RXD1
34	High	GPCLK0	SOE_N	TE2	SD1_CLK	-	-
35	High	SPI0_CE1_N	SWE_N	-	SD1_CMD	-	-
36	High	SPI0_CE0_N	SD0	TXD0	SD1_DAT0	-	-
37	Low	SPI0_MISO	SD1	RXD0	SD1_DAT1	-	-
38	Low	SPI0_MOSI	SD2	RTS0	SD1_DAT2	-	-
39	Low	SPI0_SCLK	SD3	CTS0	SD1_DAT3	-	-
40	Low	PWM0	SD4	-	SD1_DAT4	SPI2_MISO	TXD1
41	Low	PWM1	SD5	TE0	SD1_DAT5	SPI2_MOSI	RXD1
42	Low	GPCLK1	SD6	TE1	SD1_DAT6	SPI2_SCLK	RTS1
43	Low	GPCLK2	SD7	TE2	SD1_DAT7	SPI2_CE0_N	CTS1
44	None	GPCLK1	SDA0	SDA1	TE0	SPI2_CE1_N	-
45	None	PWM1	SCL0	SCL1	TE1	SPI2_CE2_N	-

Table 10: GPIO Bank1 Alternate Functions

Table 9 and Table 10 detail the default pin pull state and available alternate GPIO functions. Most of these alternate peripheral functions are described in detail in the Broadcom Peripherals Specification document and have Linux drivers available.

9.1.2 Secondary Memory Interface (SMI)

The SMI peripheral is an asynchronous NAND type bus supporting Intel mode80 type transfers at 8 or 16 bit widths and available in the ALT1 positions on GPIO banks 0 and 1 (see Table 9 and Table 10). It is not publicly documented in the Broadcom Peripherals Specification but a Linux driver is available in the Raspberry Pi Github Linux repository (`bcm2835_smi.c` in `linux/drivers/misc`).

9.1.3 Display Parallel Interface (DPI)

A standard parallel RGB (DPI) interface is available on bank 0 GPIOs. This up-to-24-bit parallel interface can support a secondary display. Again this interface is not documented in the Broadcom Peripherals Specification but documentation can be found [here](#).



9.1.4 SD/SDIO Interface

The BCM283x supports two SD card interfaces, SD0 and SD1.

The first (SD0) is a proprietary Broadcom controller that does not support SDIO and is the primary interface used to boot and talk to the eMMC or SDX_x signals.

The second interface (SD1) is standards compliant and can interface to SD, SDIO and eMMC devices; for example on a Raspberry Pi 3 it is used to talk to the on-board BCM43438 WiFi device in SDIO mode.

Both interfaces can support speeds up to 50MHz single ended (SD High Speed Mode).

9.2 CSI (MIPI Serial Camera)

Currently the CSI interface is not openly documented and only CSI camera sensors supported by the official Raspberry Pi firmware will work with this interface. Supported sensors are the OmniVision OV5647 and Sony IMX219.

It is recommended to attach other cameras via USB.

9.3 DSI (MIPI Serial Display)

Currently the DSI interface is not openly documented and only DSI displays supported by the official Raspberry Pi firmware will work with this interface.

Displays can also be added via the parallel DPI interface which is available as a GPIO alternate function - see Table 9 and Section 9.1.3

9.4 USB

The BCM283x USB port is On-The-Go (OTG) capable. If using either as a fixed slave or fixed master, please tie the USB_OTGID pin to ground.

The USB port (Pins USB_DP and USB_DM) must be routed as 90 ohm differential PCB traces.

Note that the port is capable of being used as a true OTG port however there is no official documentation. Some users have had success making this work.

9.5 HDMI

BCM283x supports HDMI V1.3a.

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-D2_P/N (data) pins must each be routed as matched length 100 ohm 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 EMC failure.



9.6 Composite (TV Out)

The TVDAC pin can be used to output composite video (PAL or NTSC). Please route this signal away from noise sources and use a 75 ohm PCB trace.

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

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.

10 Thermals

The BCM283x SoC employs DVFS (Dynamic Voltage and Frequency Scaling) on the core voltage. When the processor is idle (low CPU utilisation), it will reduce the core frequency and voltage to reduce current draw and heat output. When the core utilisation exceeds a certain threshold the core voltage is increased and the core frequency is boosted to the maximum working frequency. The voltage and frequency are throttled back when the CPU load reduces back to an 'idle' level OR when the silicon temperature as measured by the on-chip temperature sensor exceeds 85C (thermal throttling).

A designer must pay careful attention to the thermal design of products using the CM3/CM3L so that performance is not artificially curtailed due to the processor thermal throttling, as the Quad ARM complex in the BCM2837 can generate significant heat output.

10.1 Temperature Range

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

The eMMC and LPDDR2 have the narrowest range, these are rated for -25 to +80 degrees Celsius. Therefore the nominal range for the CM3 and CM3L is -25C to +80C.

However, this range is the maximum for the silicon die; therefore, users would have to take into account the heat generated when in use and make sure this does not cause the temperature to exceed 80 degrees Celsius.

11 Availability

Raspberry Pi guarantee availability of CM1, CM3 and CM3 Lite until at least January 2023.

12 Support

For support please see the hardware documentation section of the Raspberry Pi website and post questions to the Raspberry Pi forum.



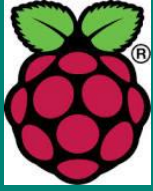
Compute Module IO Board V3



The Compute Module IO Board V3 is a development kit for those who wish to make use of the Raspberry Pi in a more flexible form factor, intended for industrial applications. The IO Board V3 is made for developing with CM3, CM3L, and CM1. This kit replaced the original Compute Module IO Board in January 2017.

The Compute Module contains the guts of a Raspberry Pi 3 (the BCM2837 processor and 1GB RAM). The accompanying IO Board is a simple, **open-source** breakout board that you can plug a Compute Module into. The board hosts 120 GPIO pins, an HDMI port, a USB port, two camera ports, and two display ports. Designing the Module into a custom system should be relatively straightforward as we've put all the tricky bits onto the Module itself, and you have the freedom to add extra components and place parts exactly where your product needs them.

The CM1, CM3, and CM3L are available for purchase in single units, or in batches of hundreds or thousands.



FLASHING THE COMPUTE MODULE EMMC

The Compute Module has an on-board eMMC 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.

Please also read the section on the [Compute Module Datasheet](#)

Steps to flash the eMMC on a Compute Module

To flash the Compute Module eMMC, you either need a Linux system (a Raspberry Pi is recommended, or Ubuntu on a PC) or a Windows system (Windows 7 is recommended). For BCM2837 (CM3), a bug which affected the Mac has been fixed, so this will also work.

Note There is a bug in the BCM2835 (CM1) bootloader which returns a slightly incorrect USB packet to the host. Most USB hosts seem to ignore this benign bug and work fine; we do, however, see some USB ports that don't work due to this bug. We don't quite understand why some ports fail, as it doesn't seem to be correlated with whether they are USB2 or USB3 (we have seen both types working), but it's likely to be specific to the host controller and driver. This bug has been fixed in BCM2837.

For Windows users

Under Windows, an installer is available to install the required drivers and boot tool automatically. Alternatively, a user can compile and run it using Cygwin and/or install the drivers manually.

WINDOWS INSTALLER

For those who just want to enable the Compute Module eMMC as a mass storage device under Windows, the stand-alone installer is the recommended option. This installer has been tested on Windows 10 32-bit and 64-bit, and Windows XP 32-bit.

Please ensure you are not writing to any USB devices whilst the installer is running.

1. Download and run the [Windows installer](#) to install the drivers and boot tool.
2. Plug your host PC USB into the CMIO USB SLAVE port, making sure J4 is set to the EN position.
3. Apply power to the CMIO board; Windows should now find the hardware and install the driver.
4. Once the driver installation is complete, run the `RPiBoot.exe` tool that was previously installed.
5. After a few seconds, the Compute Module eMMC will pop up under Windows as a disk (USB mass storage device).

SETTING UP THE COMPUTE MODULE IO BOARD

Ensure the Compute Module itself is correctly installed on the IO board. It should lie parallel with the board, with the engagement clips clicked into place.

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

Use a micro USB cable to connect the IO board to the host device.

Do not power up yet.

BUILDING RPIBOOT ON YOUR HOST SYSTEM (CYGWIN/LINUX)

We will be using Git to get the rpiboot source code, so ensure Git is installed. In Cygwin, use the Cygwin installer. On a Pi or other Debian-based Linux machine, use the following command:

```
sudo apt-get install git
```

Git may produce an error if the date is not set correctly. On a Raspberry Pi, enter the following to correct this:

```
sudo date MMDDhhmm
```

where `MM` is the month, `DD` is the date, and `hh` and `mm` are hours and minutes respectively.

Clone the `usbboot` tool repository:

```
git clone --depth=1 https://github.com/raspberrypi/usbboot
cd usbboot
```

`libusb` must be installed. If you are using Cygwin, please make sure `libusb` is installed as previously described. On the Raspberry Pi or other Debian-based Linux, enter the following command:

```
sudo apt-get install libusb-1.0-0-dev
```

Now build and install the `usbboot` tool:

```
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 the CMIO board on. The `rpiboot` tool will discover the Compute Module and send boot code to allow access to the eMMC.

WRITING TO THE EMMC - WINDOWS

After `rpiboot` completes, a new USB mass storage drive will appear in Windows. We recommend following this [guide](#) and using Win32DiskImager to write images to the drive, rather than trying to use `/dev/sda` etc. from Cygwin.

Once you have written an OS image, 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 result in the Compute Module booting the OS image from eMMC.

WRITING TO THE EMMC - LINUX

After `rpiboot` completes, you will see a new device appear; this is commonly `/dev/sda` on a Pi 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 two 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.

[View/Edit this page on GitHub](#)

[Read our usage and contributions policy](#)