



Compute Module IO Board Plus

用户手册

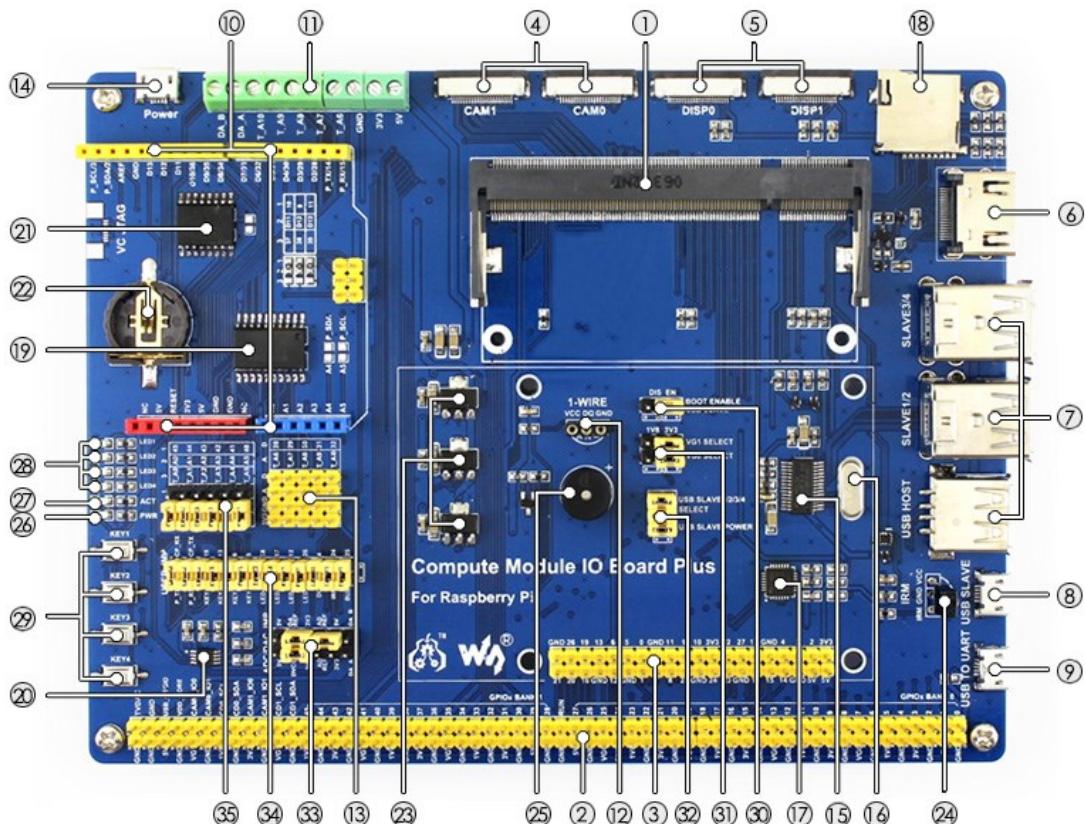
产品概述

- 本品是一块基于树莓派计算模块的外扩板，与树莓派基金会提供的 Compute Module IO Board V3 兼容，并扩展了多种常用的资源。支持树莓派官方的 Compute Module 3 Lite, Compute Module 3, Compute Module 3+ Lite, Compute Module 3+ (8G/16G/32G).

特点

- 兼容树莓派基金会提供的 Compute Module IO Board V3
- 带有 Raspberry Pi GPIO 接口，方便接入各种树莓派扩展板
- 带有 Arduino 接口，方便接入各种 Arduino 扩展板
- 带有 1-WIRE 接口，方便接入各种单总线设备（如 DS18B20）
- 板载 4 个按键、4 个 LED 和 1 个蜂鸣器，方便进行 I/O 测试
- 板载 USB HUB，方便接入更多的 USB 设备
- 板载红外接收器，方便使用红外遥控功能
- 板载 USB TO UART，方便进行串口调试
- 板载传感器接口，方便接入各种传感器
- 板载 ADC，10 位，38KSPS，11 通道（6 通道用于 Arduino 接口，5 通道用于接入传感器）
- 板载 DAC，2 通道，16 位精度
- 板载 RTC，这是常用的功能之一

板载资源



【核心接口简介】

1. Compute Module 接口

可接入 Compute Module 3, Compute Module 3 Lite, Compute Module 3+(8G/16G/32G) 或者 Compute Module 3+ Lite

2. Compute Module GPIO 接口

引出 Compute Module 所有接口

3. Raspberry Pi GPIO 接口

可接入各种 Raspberry Pi HAT

4. CSI 摄像头接口

可接入 Raspberry Pi Camera

5. DSI 显示接口

可接入 Raspberry Pi LCD

6. HDMI 接口

可接入各种 HDMI 设备

7. USB 接口

可接入各种 USB 设备

8. USB SLAVE 接口

Compute Module 3 可通过此接口烧录系统镜像

9. USB TO UART 接口

方便进行串口调试

10. Arduino 接口

方便接入各种 Arduino 扩展板(Shield)

11. AD/DA 输入输出接口（接线端子）

方便在各种场合使用

12. 1-WIRE 接口

可接入 DS18B20 等单总线设备

13. 传感器接口

方便接入各类传感器

14. 电源接口

一般接入 5V 2.5A 电源

【器件简介】**15. FE1.1S**

USB HUB 芯片

16. 12MHz 晶振**17. CP2102**

USB 转串口芯片

18. Micro SD 卡槽

用于接入带系统的 Micro SD 卡，以启动

Compute Module 3 Lite

19. TLC1543

AD 转换芯片

20. DAC8552

16 位高精度 DAC，2 通道输出

21. DS3231

高精度 RTC 芯片，I2C 接口

22. RTC 实时时钟供电电池座

可接入 CR1220 纽扣电池

23. 电源稳压芯片

从上到下，分别是 3.3V / 2.5V / 1.8V 稳压芯片

24. LFN0038 红外接收头**25. 蜂鸣器****26. 电源 LED****27. ACTLED**

指示 Micro SD 卡运行状态

28. 用户 LED

便于 I/O 输出测试或显示程序运行状态

【跳线说明】**29. 用户按键**

便于 I/O 输入测试或控制程序运行状态

30. BOOT 选择跳线

EN: PC 机可通过 USB SLAVE，读写 SD 卡或 eMMC

DIS: 计算模块从 SD 卡或者 eMMC 启动

31. VGx 电源选择跳线

选择 I/O 电平

32. HUB 使能跳线

HUB 使能和 USB SLAVE 供电选择

33. ADC/DAC 配置跳线

配置 ADC/DAC 供电电压和参考电压

34. 多种外设配置跳线

配置 UART、用户按键、用户 LED、1-WIRE 接口、红外接收头和蜂鸣器的控制引脚

35. Arduino AD 选择跳线

短接 1 与 2: Arduino 接口的 A0-A5 作为数字控制脚

短接 2 与 3: Arduino 接口的 A0-A5 作为 AD 输入脚

目录

产品概述.....	1
特点.....	1
板载资源.....	2
目录	4
使用方法.....	7
烧录镜像.....	7
烧写镜像到 CM3/CM3+.....	7
烧写镜像到 CM3L.....	7
连接屏幕和摄像头	7
连接树莓派官方 7 寸屏幕	8
连接微雪 7inch HDMI LCD (C)	8
连接摄像头	8
示例程序.....	9
蜂鸣器 pwm 实验	10
python 程序.....	10
wiringpi 程序.....	10
预期结果	10
DAC 实验.....	10
BCM2835 程序.....	10
预期结果	11
DS18B20 实验.....	11
sysfs 程序	11

python 程序.....	11
预期结果	11
RTC DS3231 实验.....	11
BCM2835 程序.....	11
wiringPi 程序.....	11
python 程序.....	12
预期结果	12
IRM 红外遥控实验.....	12
BCM2835 程序.....	12
wiringPi 程序.....	12
python 程序.....	12
预期结果	13
按键实验.....	13
BCM2835 程序.....	13
WiringPi 程序.....	13
python 程序.....	13
预期结果	13
LED 实验.....	13
BCM2835 程序.....	13
WiringPi 程序.....	14
python 程序.....	14
预期结果	14
ADC 实验.....	14

BCM2835 程序.....	14
wiringPi 程序.....	14
python 程序.....	15
预期结果	15
串口通信实验	15
WiringPi 程序.....	15
python 程序.....	15
预期结果	15

使用方法

烧录镜像

烧写镜像到 CM3/CM3+

如果你使用的是树莓派 Compute Module 3 或者 Compute Module 3+. 这几款带有 EMMC 的计算模块的话，使用的时候需要把镜像烧写到计算模块上的 EMMC 内，否则无法使用。

1. 拔掉 Compute Module IO Board Plus 板（后面简称 IO 板）上的 USB SLAVE 1/2/3/4 SELECT 跳线帽（注意这要拔掉的是两个跳线帽）。将 BOOT ENABLE USB SLAVE 跳线帽接到 EN 端。
2. 电脑上运行 `rpiboot_setup` 软件（请在微雪百科界面下载）安装树莓派 USB 驱动，安装前建议先关掉杀毒软件。安装成功后再安装录下会有一个 `rpiboot.exe` 的可执行文件
3. Power 接口连接外部 5V 电源，用 USB 线将 IO 板的 USB SLAVE 接口连接到电脑。
4. 运行 `rpiboot.exe` 可执行文件，电脑会把 Compute Module IO Board Plus 识别为 U 盘
5. 运行 `WinDiskImager.exe` 烧录工具，选择 CM3 镜像，将镜像烧录到 EMMC 中。
6. 烧写完成后，把 IO 板上的跳线帽恢复原样。

注：

在烧写过程中注意 PC 不要另外对其他 USB 设备进行写操作，避免出现冲突
由于 CM3 的 EMMC 只有 4G（最新的 CM3+ 扩大了 EMMC 容量，可以支持正常的树莓派镜像了），所以烧录的镜像大小不可以超过 4G。因此，如果烧录 `rasbian`，只能使用 Lite 版本的镜像。如果想要 GUI 的。可以在安装完成后运行下面的指令安装 GUI

```
sudo apt-get update  
sudo apt-get install raspberrypi-ui-mods
```

烧写镜像到 CM3L

1. 下载 CM3L 镜像。
2. 将 SD 卡用读卡器插到电脑，SD 卡要求大于 8G。
3. 打开烧录工具 `Win32DiskImager.exe`，选择 CM3L 镜像烧录。（同烧录树莓派镜像一样）
4. 烧录成功后，把 SD 卡插到底板的 SD 卡卡槽即可

连接屏幕和摄像头

下载我们提供的测试镜像并烧录，烧录方式见上文

注：使用前检查一下 **BOOT ENABLE USB SLAVE** 跳线帽是否设置在 **DIS** 一端。

连接树莓派官方 7 寸屏幕

1. 用 LCD 转接板和 22PIN FFC 线将屏幕连接到 Compute Module IO Board Plus the DISP1 接口
2. 杜邦线将 5V 和 GND 引脚连接到 Plus 扩展板的 5V 和 GND 引脚
3. 使用杜邦线短接：

GPIO0<->CD1_SDA

GPIO1<->CD1_SCL

4. 连接电源

5. 等待几秒后屏幕启动

连接微雪 7inch HDMI LCD (C)

1. 确保树莓派官方 7 寸 DISP 接口电容屏没有接到底板上，系统才会默认进入 7inch 1024x600 HDMI 电容显示屏
2. HDMI 线连到底板的 HDMI 接口和触摸屏的 HDMI 接口。 USB 线连接到底板的 SLAVE 接口和屏幕的 USB 接口
3. 连接电源
4. 等待几秒后屏幕启动

连接摄像头

1. 将摄像头连接到 CAM1 接口（[需要 RPi Zero V1.3 Camera Cable 15cm](#)）

2. 用杜邦线短接：

GPIO0<->CD1_SDA

GPIO1<->CD1_SCL

GPIO4<->CAM1_IO1

GPIO5<->CAM1_IO0

3. 启动电源

4. 接入多个摄像头，可以将另一个摄像头接入到 CAM0 接口。

5. 使用 CAM0 接口的时候需要用杜邦线短接：

GPIO28<->CD0_SDA

GPIO29<->CD0_SCL

GPIO30<->CAM0_IO1

GPIO31<->CAM0_IO0

6. 摄像头显示命令，可以分别执行：

```
raspivid -t 0 -cs 0  
raspivid -t 0 -cs 1
```

注：

1. -cs 命令用于选择摄像头 0 或者 1，参数 0 表示 CAM1，参数 1 表示 CAM0
2. 官方 Raspbian 系统默认是没有加官方摄像头和官方 7 寸屏配置启动文件。使用官方 Raspbian 系统需要把官方提供的 dts 文件转换为 bin 文件放入到系统/boot/下。（dts 文件示例程序中有提供，位于/Program/dt 目录下）

转换命令：

```
sudo dtc -l dts -O dtb -o /boot/dt-blob.bin dt-blob-disp1-cam2.dts
```

使用我们提供的系统，系统已配置好，无需另外再做转换



示例程序

示例程序使用的时候，底板连接一个屏幕用于显示，连接一个键盘用于输入。

如果是使用官方 Raspbian 镜像，使用示例程序的时候需要安装 `bcm2835`、`wiringPi`、`python` 函数库，函数库安装方法详见为[树莓派安装必要的函数库](#)

蜂鸣器 PWM 实验

python 程序

- 终端执行指令进入程序所在目录:

```
cd /home/pi/CM3/Buzzer_PWM/python/
```

- 执行指令运行程序:

```
sudo ./buzzer.py
```

wiringpi 程序

- 终端执行指令进入程序所在目录:

```
cd /home/pi/CM3/Buzzer_PWM/wiringPi/
```

- 执行指令运行程序:

```
sudo ./buzzer
```

预期结果

蜂鸣器会响，声音有小变大，然后再变小

注：

Computer Module IO Board Plus 在使用过程中，蜂鸣器有时候由于干扰会响，这时候可以把 `USER_JMP` 最后一个跳线帽拔掉（`BUZ<->25`）。

DAC 实验

BCM2835 程序

- 用两条杜邦线（最好是 2PIN 的），分别连接 `DA_A`, `DA_B` 到 `USER JMP` 的 `LED1`, `LED2` 排针
- 终端执行指令进入程序目录:

```
cd /home/pi/CM3/DAC8532
```

- 执行指令运行程序:

```
sudo ./dac8532
```

预期结果

LED1 和 LED2 交替由暗变亮，再由亮变暗。

DS18B20 实验

需要 DS18B20，按照丝印指示，将 DS18B20 半圆边对着蜂鸣器插入到 1-WIRE 接口。注意不要插反，否则容易产生高温灼伤手指。

sysfs 程序

- 终端执行指令进入程序所在目录:
`cd /home/pi/CM3/DS18B20/fs/`
- 执行指令运行程序:
`sudo ./ds18b20`

python 程序

- 终端执行指令进入程序所在目录:
`cd /home/pi/CM3/DS18B20/python/`
- 执行指令运行程序:
`sudo ./ds18b20.py`

预期结果

输出当前测量的温度结果。可以 `Ctrl+C` 退出程序。

RTC DS3231 实验

BCM2835 程序

- 终端执行指令进入程序所在目录:
`cd /home/pi/CM3/DS3231/bcm2835/`
- 执行指令运行程序:
`sudo ./ds3231`

wiringPi 程序

- 终端执行指令进入程序所在目录:

```
cd /home/pi/CM3/DS3231/wiringPi/
```

- 执行指令运行程序:

```
sudo ./ds3231
```

python 程序

- 终端执行指令进入程序所在目录:

```
cd /home/pi/CM3/DS3231/python/
```

- 执行指令运行程序:

```
sudo ./ds3231.py
```

预期结果

输出时间信息。按 **Ctrl+C** 可以退出程序

IRM 红外遥控实验

需要用到红外遥控器。使用的时候注意取下隔离纸

BCM2835 程序

- 终端执行指令进入程序所在目录:

```
cd /home/pi/CM3/IRM/bcm2835/
```

- 执行指令运行程序:

```
sudo ./irm
```

wiringPi 程序

- 终端执行指令进入程序所在目录:

```
cd /home/pi/CM3/IRM/wiringPi/
```

- 执行指令运行程序:

```
sudo ./irm
```

python 程序

- 终端执行指令进入程序所在目录:

```
cd /home/pi/CM3/IRM/python/
```

- 执行指令运行程序:

```
sudo ./irm.py
```

预期结果

按下红外遥控器的按键，终端会显示按下的按键的相应数值。按 **Ctrl+C** 退出程序。

按键实验

BCM2835 程序

- 终端执行指令进入程序所在目录
`cd /home/pi/CM3/KEY/bcm2835/`
- 执行指令运行程序：
`sudo ./key`

WiringPi 程序

- 终端执行指令进入程序所在目录：
`cd /home/pi/CM3/KEY/wiringPi/`
- 执行指令运行程序：
`sudo ./key`

python 程序

- 终端执行指令进入程序所在目录：
`cd /home/pi/CM3/KEY/python/`
- 执行指令运行程序：
`sudo ./key.py`

预期结果

按下底板上面的按键 KEY1、KEY2、KEY3 和 KEY4，终端会显示按下的按键的信息，例如按下 KEY1，显示 `press the key : 0`。按 **Ctrl+C** 退出程序。

LED 实验

BCM2835 程序

- 终端执行指令进入程序所在目录

```
cd /home/pi/CM3/LED/bcm2835/
```

- 执行指令运行程序:

```
sudo ./led
```

WiringPi 程序

- 终端执行指令进入程序所在目录:

```
cd /home/pi/CM3/LED/wiringPi/
```

- 执行指令运行程序:

```
sudo ./led
```

python 程序

- 终端执行指令进入程序所在目录:

```
cd /home/pi/CM3/LED/python/
```

- 执行指令运行程序:

```
sudo ./led.py
```

预期结果

4 个 LED 灯轮流闪烁。Ctrl+C 退出程序

ADC 实验

BCM2835 程序

- 终端执行指令进入程序所在目录

```
cd /home/pi/CM3/TLC1543/bcm2835/
```

- 执行指令运行程序:

```
sudo ./tlc1543
```

wiringPi 程序

- 终端执行指令进入程序所在目录:

```
cd /home/pi/CM3/TLC1543/wiringPi/
```

- 执行指令运行程序:

```
sudo ./tlc1543
```

python 程序

- 终端执行指令进入程序所在目录:
`cd /home/pi/CM3/TLC1543/python/`
- 执行指令运行程序:
`sudo ./tlc1543.py`

预期结果

输出 AD 数据。Ctrl+C 退出程序

串口通信实验

用 USB 线将底板的 USB TO UART 接口连接到电脑。电脑打开 Putty 软件，设置串口通信波特率为：115200。

电脑端 Putty 输入用户名和密码登录 CM3/CM3L （用户名和密码默认为 pi 和 raspberry）

WiringPi 程序

- 终端执行指令进入程序所在目录:
`cd /home/pi/CM3/UART/wiringPi/`
- 执行指令运行程序:
`sudo ./uart`

python 程序

- 终端执行指令进入程序所在目录:
`cd /home/pi/CM3/UART/python/`
- 执行指令运行程序:
`sudo ./usart.py`

预期结果

程序运行后可以看到 PC 端 Putty 界面打印出 Hello Wrold!!!字样。Ctrl+C 退出程序。