

# 树莓派底层配置

由于树莓派起源于以BCM2835为核心的嵌入式计算设备，树莓派并没有像传统PC那样有BIOS菜单和丰富的高权限级别的系统设置选项可以被配置。作为替代，其依赖于包含系统配置字符串的文本文件，这一文本文件在电源打开的时候可以被芯片载入读取。

在查看这几个有着丰富选项的文件--`config.txt` , `cmdline.txt` , `start.elf`-之前,需要先警告几件事：将默认配置选项改为其他最佳实践配置选项之前，树莓派可能不会启动，这需要您将相应文件回滚；更糟糕的实践甚至会在物理上损坏您的系统。在这个章节中，这些潜在的危险选项需要在这个章节中高度重视。

## 硬件设置--`config.txt`

树莓派的硬件设置配置包含在一个命名为 `config.txt` 的文件中，其位于 `/boot` 目录中（参见图片6-1）。这个文件通告派如何设置它的各式输入和输出的格式，以及BCM2835芯片的频率和运行时所连接的内存模块。

如果您对图像的输出有疑问的话，例如为什么图片没有全屏或者越过边缘，`config.txt`是您能够修复这些问题的文件。通常来讲，这个文件仅仅是在目前为空或者在某些发行版中不为空，这只是意味着派能够使用预先设置的文件配置。如果您想做些变化、或者这个文件并不在这里的话，那就创建一个新的名为`config.txt`的文本文件并且填入您想要改变的配置文件。

`config.txt` 文件能够控制几乎树莓派硬件中几乎所有的层面，除了BCM2835的中央处理器CPU和图形处理器GPU使用的内存部分。您也能够在下个章节中学会如何变更内存分区的容量-`start.elf`。

在系统第一次启动时，`config.txt`文件是只读的。所有的文件变更只在下一次派重新启动或者关机后再次启动时才会生效。在不想变更生效时，只需要简单的在 `/boot` 目录中删除对应的文件就足够将默认配置生效。如果派没有根据您的新的选项进行启动，只需要简单的移除SD卡和另一个PC上从boot分区中删除`config.txt`文件，然后再次重新插入SD卡，再重复之前步骤。

## 显示配置

通常来讲，树莓派会自动检测其所连接的显示器且自动根据情况变更设置。然而在有些时候，自动检测并没有正常工作。这通常发生在树莓派发行地区和一个okler电视发行地区不一致的时候。如果您将您的树莓派连上您的电视时什么都没有看到，您或许需要覆写这些地区选项以保持一致。

在config.txt中极其丰富的选项设置可以被用来改善或者变更视频输出。这些设置和他们的可能的值，将在下面的列表中进行描述。

### 警告

通常而言，校正HDMI设置或者复杂的视频输出设置可能让您的派不能够很好的和您的显示器连接。大多数情况下，最佳实践仍然是自动检测的设置，除非您不能很好的在显示器上看到开启时的树莓派的第一幅图片。

- overscan\_left
  - 这将图片向内移动一列像素以适配电视的扫描方式。如果在树莓派上的字符显示消失在屏幕的边缘的时候，调节扫描方式将可以修复之一问题。需要调整的像素的行数值也需要被给出。
- overscan\_right
  - 这和overscan\_left做着相同的工作，不过是对屏幕的右方。
- overscan\_top
  - 再一次的，这会忽略一系列的像素值，但是这次是对屏幕的顶部。
- overscan\_bottom
  - 这可能被用来跳过屏幕底部一系列的像素点。典型情况下，所有的overscan\_的设置值都将是相同的，以在显示器上创建规则的边缘。
- disable\_overscan
  - 如果您使用一个显示器或者电视来连接HDMI，您或许会发现您的显示图像有一个黑色边界在围绕。为了避免这一黑边，任何其他的默认的扫描设置都可能被设置为1--默认设置值。
- framebuffer\_width

- 这个值由像素计量出，调整之可以改变控制台的宽度。如果在屏幕上文本显示字体过小，试图改变其到更小的值上远比调整为显示器的默认值更有效。
- framebuffer\_height
  - 该值以和framebuffer\_width相同的方式影响控制台字符的大小，但是在竖直高度而不是水平高度。
- framebuffer\_depth
  - 控制每bit像素在控制台的色彩深度。默认为16bit，这有65536种色彩。其他值包括8bit（256种色彩），24bit（大约1670万色彩）和32bit（约10亿色彩）都是合理的，但是许多导致图像错误。
- framebuffer\_ignore\_alpha
  - 设置为1时，这个值默认会禁止alpha通道，该通道是为控制控制台的色彩透明度。静止alpha通道常规上不需要，但是设置framebuffer\_depth为32bit每像素时可能出现图像错误。
- sdtv\_mode
  - 该值影响树莓派的模拟制式视频输出，调整其以适用于各类国家。但是默认上，树莓派使用北美的NTSC制式标准；用户在其他的国家或许需要改变这一值来在模拟信号的电视上显示图像，可能的值有：
    - 0--NTSC，北美电视输出格式
    - 1--NTSC-J，日本的电视标准
    - 2--PAL，英国和其他国家电视输出标准
    - 3--PAL-M，巴西电视输出标准
- sdtv\_aspect--控制模拟信号输出的长宽比，如果图像被拉伸或者挤压，将使用合适的电视长宽比替代。可能的值为：
  - 1--4:3的长宽比，常见于老型号
  - 2--14:9的长宽比，常见于小型的宽屏电视
  - 3--16:9的长宽比，常见于现代的宽屏电视
- hdmi\_mode
  - 设置为针对模拟信号视频模式输出的设置，可以能覆写了HDMI接口上的自动检测解决方案。如果您想让您的树莓派的显示输出分辨率比原生分辨率更小，来在一定范围内更加的可读。在附表B中--HDMI显示模式--列出可能的设置的值。
- hdmi\_drive
  - 可以变更HDMI端口的电压输出。当使用HDMI转DVI接口适配器的时候，这是十分的重要的。因为HDMI接口和DVI接口电压有轻微的不同。如果您发现图像出现了雪花点或在一幅图像中有轻微的亮光，试图变更下面的值，可能的值如下：

- 1--DVI 输出电压，在这个模式中，HDMI输出中并不包括音频。
- 2--HDMI 输出电压，在这一模式中，HDMI输出中包括音频。
- `hdmi_force_hotplug`
  - 树莓派强制使用HDMI端口，即使其并没有检测到已连接的显示器。0的值允许树莓派尝试检测显示器，而为1的值可使树莓派自由使用HDMI输出。
- `hdmi_group`
  - 设置HDMI组模式为CEA或者DMT。在使用`hdmi_mode`前来控制输出分辨率和刷新频率前，您应该根据您想连接的显示器的情况改变相应的设定值。下面是可能的值：
    - 1--设置HDMI组驱动定义为CEA，当树莓派通过HDMI接口连接上HDTV时，可以使用附表B中的第一个设置。
    - 2--在DMT类别中设置HDMI输出为VESA驱动模式，用VESA设置树莓派可以轻松的通过DVI连接上电脑显示器，或者使用第二备选项--附表B“HDMI显示模式”。
- `hdmi_safe`
  - 强制使树莓派使用预先定义的显示设置集合，提供最大兼容性的HDMI端口的适配。这个为1的值等价于  
`hdmi_force_hotplug=1` , `config_hdmi_boost=4` ,  
`hdmi_group=1` , `hdmi_mode=1` 和 `disable_overscan=0`
- `config_hdmi_boost`
  - 一些显示器需要HDMI线提供更多的电力时使用，如果您的图像呈现雪花状，尝试增加这个值的范围，从1（短线缆）到7（长线缆）。

`config.txt` 的每个选项都独占一行，同时用等价符号跟随。例如，告知树莓派使用模拟制式PAL规范长宽比为4:3的显示器，且每秒扫描20线时，将下面的行数放在`config.txt` 中：

```
sdtv_mode=2
sdtv_aspect=1
overscan_left=20
overscan_right=20
overscan_top=20
overscan_bottom=20
```

说明了树莓派使用通过HDMI转接器连接DVI显示器接口，格式输出为720p60，非完全扫描方式，可以用下面的代替：

```
hdmi_group=1
hdmi_mode=4
hdmi_drive=1
disable_overscan=1
```

为了使下面的改动生效，树莓派必须重启。如果您的关于视频输出的改动失效的话，将SD卡插入到其他的计算机中，或者再次改动config.txt 到新的设置项，当然您也可以删除文件以恢复默认设置。

## 启动选项

config.txt 文件也可以被用来控制树莓派上Linux载入的方式。然而，最常见的控制Linux内核的方式就是使用分离的文件--cmdline.txt--在后面的章节中您将了解这一文件。当然只用config.txt也是可以的。下面的选项控制了启动进程：

- `disable_commandline_tags`
  - 在载入Linux内核前跳过填充0x100前的内存区域（告知start.elf）。这一选项不应被禁止，因为可能造成Linux载入错误甚至崩溃。
- `cmdline`
  - 可被Linux内核解读的参数，可被用来替代cmdline.txt,通常可以在/boot 目录中查看。
- `kernel`
  - 和内核文件同名的文件载入。通常作为一种紧急恢复模式载入使用。
- `ramfsfile`
  - 载入的初始化ramfs文件系统，很少被定义，除非需要建立一个专门的新的初始文件系统。
- `init_uart_baud`
  - 串口的传输速率，单位为每秒比特率。默认为115200bps，在更老的串口通信时更低的值可以提高连接稳定性。



## 树莓派的超频

config.txt文件只控制树莓派的BCM2835处理器的GPU输出，但也能以另一种方式控制芯片。尤其是，它能变更芯片运行的频率，在相同费用周期的前提下提高对应的性能。处理器中常说的超频。

### 警告：

下面的列出的设置选项可能导致您的树莓派损坏，尤其是改动关于内存的选项，GPU或CPU的电压将熔断芯片的保险，可能导致不可逆转的损伤--哪怕参数设置回滚到正常范围--损伤已经造成。

这类的问题将无法在树莓派基金会和相应的您购买树莓派的分销商上获得合理的保修权利。如果有疑问，不要改变下面的设置：借助超频获得的更高性能提升几乎不值得对树莓派的风险。

BCM2835多媒体处理器是树莓派的核心，作为一个片上芯片系统，设计中被划分为两个主要部分：CPU和GPU。简单来说，CPU负责处理常见的所有每日处理进程队列，于此同时，GPU处理屏幕上的绘图功能，无论是2D的还是3D的。

使用config.txt, 您能超频一个或所有的BCM2835的部件。您能增加内存模块处理的速度--作为一个层级挂载操作方式的上方模块。

提高工作频率后的树莓派的部件性能的小幅度增加将导致：增加GPU的时钟频率意味着3D图形（如游戏图形）将会表现出更快的速度，并且视频解码的播放将更快更流畅;提高CPU的时钟频率将提高设备的整体性能，如将增加内存的频率。

Pi 不将提供更高的运行速度放在第一位的原因是有关芯片的寿命。BCM2835 由其制造商博通生产，额定的操作速度是700兆赫。在正式额定频率上增加频率可能正常工作，但它也会对芯片的寿命有影响。与台式机处理器不同，SoC设计上很少有为超频的空间。

### 锁频设置

如果您仍然承担树莓派损坏的风险--一个在嵌入式圈子中已知的常识--只为了小小的性能提升的话，下面的config.txt的设置可能会有帮助，下面的设置选项控制树莓派片上系统的性能：

- `arm_freq` —— 设置核心时钟频率的 CPU 部分的 BCM2835，为通用的性能提升。默认速度是 700 MHz。
- `gpu_freq` —— 设置 BCM2835，所有操作的图形性能提升的 GPU 部分的时钟频率。默认速度是 250 兆赫。另外，您可以调整单个部分的 GPU 硬件使用以下选项：
  - `core_freq` —— 设置 GPU，留下其它的频率，以提高整体的 GPU 性能的核心时钟频率。默认速度是 250 兆赫。
  - `h264_freq` —— 设置了 GPU 的硬件视频解码器，提高播放 H.264 视频数据的时钟频率。默认速度是 250 兆赫。
  - `isp_freq` —— 设置时钟频率的图像传感器管道，提高捕获率的连接视频硬件（例如摄像机）。默认速度是 250 兆赫。
  - `v3d_freq` —— 设置了 GPU 的 3D 渲染硬件，为可视化和游戏性能提高的时钟频率。默认速度是 250 兆赫。
- `sdram_freq` —— 在 Pi，给整个系统小的增加在性能上设置随机存取内存 (RAM) 芯片的时钟速度。默认的速度是 400 MHz。
- `init_uart_clock` —— 设置默认时钟速度的通用异步收发器 (UART)，用于控制串行控制台。默认值是的 3000000，哪套的 3 兆赫。改变这速度可能超出损坏的串行控制台输出几乎没有影响。
- `init_emmc_clock` —— 设置默认的 SD 卡控制器的时钟速度。默认值为 80000000，哪套速度的 80 兆赫。增加这个值可以导致更快的阅读和写作从 SD 卡，但也可能会导致数据损坏。

例如，CPU超频到 800 MHz，GPU超频到 280 兆赫和内存频率到 420 MHz时，在`config.txt`下输入下面的选项，每个一行：

```
arm_freq=800
gpu_freq=280
sdram_freq=420
```

## 超频设置

如果您正在超频您的树莓派，您将最终遇上一种“超频墙”，也就是一台设备无法越过的极限。由于现实设备在制造过程中芯片生产的复杂多样性，这点依赖于各自的树莓派设备的元件上的可靠程度。对于一些用户来说，这将限制到最低800MHz的频率上，或者其他用户可能将他们的树莓派的CPU频率提高到1GHz而不发生任何问题。

如果您想从树莓派中获取更多的性能，一种可能的方式是增加处理器的上限，已知的处理器性能的超频和加压。树莓派的 BCM2835 偏上芯片和所连接的内存模块通常运行在 1.2 伏特。如果可能，尽管这种方式不被建议，覆写默认配置并迫使组件运行在更高或更低的电压之下。加压意味着增加芯片上可能的时钟频率。这使得可能有更高的频率，这也可能意味着芯片运行过热，相比只进行超频的树莓派，缩减了 CPU 可能的生命周期。

### 警告：

在 `config.txt` 中设置的任何电压选项导致内部 BCM2835 温度安全界限失效，重启不能重置的方式。它是万无一失的方式来告诉是否有人一直在试图超频芯片超过额定的规格，并会使您的保修无效——即使失败的原因是无关超频。如果您在保修期内返回 Pi 不幸出现问题，它不会被取代。不要试图对接地 Pi，你不能来代替你。

与前面所述的设置，在 `config.txt` 提供的作为绝对值，不同电压的调节进行使用相对于 Pi 的基准 1.2V 设置的值。对每个整数零度以上，电压被上调 0.025V 从基准值。每个完整的数字低于零，电压下跌 0.025V 从基准值。电压调整设置有 8 和 -16，等于 0.2 V 层级电压高于或低于层级电压 1.4 V 绝对值和 0.4 V 或 0.8 V 绝对值的上限和下限。必须在整个数字调节电压和 0.8 V (-16) 低于或高于 1.4 V (8.)，但不能调整。

下面在 `config.txt` 中的设置调节是可行的：

- `over_voltage` —— 调整 BCM2835 的核心电压。完整的数字（一个整数），对应于 0.025 V 的上方或下方的默认值 (0, ) 8 上限与下限的 -16 给出值。
- `over_voltage_sdram` —— 调整给内存芯片在 Pi 上的电压。与 `over_voltage`，完整的数字，对应于 0.025 V 的上方或下方的基准 (0, ) 8 上限与下限的 -16 给出值。此外，您可以调整电压为单个内存组件使用以下选项：
- `over_voltage_sdram_c` —— 调整给内存控制器的电压。可接受的值是与 `over_voltage_sdram` 相同。
  - `over_voltage_sdram_i` —— 调整到内存中的输入/输出 (I/O) 系统给出的电压。可接受的值是与 `over_voltage_sdram` 相同。
  - `over_voltage_sdram_p` —— 调整电压给内存中的物理层 (PHY) 组件。可接受的值是与 `over_voltage_sdram` 相同。



下面输入下面的行到config.txt中，将树莓派的BCM2835的电压小幅度增加0.05V到1.25V，内存的芯片更大幅度增加0.1V到1.3V：

```
over_voltage=2  
over_voltage_sdram=4
```

在其他的设置中，从config.txt中删除对应的行数，将其返回至正常。不像这个章节其他的设置，，证据会留在 BCM2835 安全机制中存在——表现出 Pi 的保修无效，甚至后恢复默认设置。

## 禁用L2 Cache

树莓派的BCM2835 SoC处理器在板子上有大小为128KB的L2 Cache。尽管这个缓存很小，它还是十分的迅速，L2缓存通常是临时存储数据和指令集。在更慢的主存和更快的处理器中缓冲，以改善性能。

由于BCM2835开始设计为多媒体处理芯片目的是为机顶盒，这个L2 Cache被设计为单独给GPU的芯片部分使用，不像传统的处理器，CPU部分本身并不包含自己的L2Cache。

借助config.txt,您能告知BCM2835允许使用CPU部分访问L2 cache缓存，在多数情况下，这能改善性能。在其他几个情况下，这可能损伤性能。由于物理的缓存位置问题，CPU到缓存的距离比GPU到缓存的距离远得多。

使用L2 缓存也需要Linux发行版编译缓存相关文件。在编译中启用L2 cache 或许能比标准的禁用L2 cache的方式更陌生和未预料的行为。

为了转换L2 cache在CPU的权限上，简单在config.txt 上加入下面这一行即可：

```
disable_l2cache=0
```

在所有的config.txt设定完成后，系统将会自动重启便于变更生效。要禁止CPU的对缓存的访问权限，将0替代为1。

## 启用测试模式

在 `config.txt` 这最后一个选项是一个绝大多数 Pi 用户不需要触碰，但出于完整性的考虑的，这里包括：测试模式。树莓派厂，测试模式在生产过程中使用——结合特殊的硬件，用于检查良品率——允许工厂工作人员测试以确保 Pi 运行正常。

### 警告：

启用测试模式不会造成永久性伤害，但 Pi 不会启动到其操作系统，直到再次禁用模式并且 Pi 的电源被关闭然后重新打开时。

如果您十分好奇想知道树莓派在工厂模式中的样子，您能够进一步启用测试模式到 `config.txt` 文件中：

```
test_mode=1
```

在其他的 `config.txt` 设置中，测试模式并没有启用，直到树莓派重启。测试模式可能再次被禁用，通过去除配置文件中的行数。

## 内存分区--start.elf

尽管树莓派有256MB的内存芯片，还是能被各种各样的方式使用。BCM2835被划分为两个主要的分区：通用计算CPU和图形计算GPU。所有的这两个都需要内存进行操作，意味着256MB的内存存在树莓派上需要被划分成两段。这个分割有名叫 `start.elf` 的文件分割。

这个典型的分割由安装在树莓派上的Linux发行版的特性决定，一些选择中间划分，128MB对应到每个CPU和GPU，确保每个图形硬件能够展现全部的性能，其他的允许CPU有更大的共享内存便于改善通用的计算性能。

主要的发行版中，包括了三种类别的 `start.elf` 复制项，以便于树莓派启动时添加载入：`arm128_start.elf`, `arm192_start.elf`, and `arm224_start.elf`。这三种文件是理想的除了一个更小的改变项：预留为BCM2835的CPU的部分。

第一个文件，`arm128_start.elf`，配置中间拆分下来，128 MB 可用到 BCM2835 的 ARM 处理器和 128 MB 可用到的 VideoCore 四 GPU 内存。第二和第三个文件逐渐减少到 GPU 可用的内存量。`arm192_start.elf` 给出了 192 MB 到 CPU 和 GPU，虽然 `arm224_start.elf` 给出了 224 MB 到 64 MB 到 CPU 和 GPU 到 32 MB。通过讲述 Pi 使用这些文件，你可以增加通用计算可用的内存量。

### 警告：

有着很高图像负载工作的应用，像3D游戏和高分辨率的视频处理软件，通常需要128MB的内存给GPU。缩减这个可能导致动态的性能变化。

大多数的通用目的发行版中工作于192MB和64MB的划分，但是你能自由释放大多数的内存容量给程序，通过迫使树莓派使用 224MB/32MB的分割方式。这能够改善通用目的计算机性能而不需要加大相对应的超频的树莓派的风险。

若要更改内存分配的方式，从那 `/boot` 目录删除 `start.elf` 和复制其他三个版本之一在的地方。请确保文件重命名为 `start.elf`，否则 Pi 将无法启动。最简单的实现这一目标的方法是在终端键入以下命令：

```
sudo cp /boot/arm224_start.elf /boot/start.elf
```

当你下次重新启动 Pi 时，它会有更多的内存提供给 ARM 系列的 CPU。要检查可用的内存量，请在命令行中键入 `free` 查看。若要更改到另一个内存划分，仅重复前面的命令使用不同版本的 `start.elf` 作为源文件：`arm_192_start.elf` 或 `arm128_start.elf` 即可。

## 软件的设置-- `cmdline.txt`

除了 `config.txt` 文件可以控制树莓派的各种功能的硬件外，还有另一个重要的文本文件在 `/boot` 目录：`cmdline.txt`（请参见图 6-3）。此文件包含所谓的内核模式行——该选项传递给 Linux 内核作为 Pi 启动的载入文件之一。

在基于 Linux 的台式机或笔记本电脑中，这些选项通常被称为引导工具，由它自己的配置文件传递到内核。基于 Pi，选项可以简单地直接输入到 `cmdline.txt`，让 Pi 在启动时读取。几乎任何支持的 Linux 的内核选项都可以更改要改变控制台或加载的内核的外观之类的 `cmdline.txt` 文件。这里是一个例子，来自 Debian 的分布，应书面文件作为一个连续的行中的 `cmdline.txt` 文件：

```
dwc_otg.lpm_enable=0 console=ttyAMA0,115200  
kgdboc=ttyAMA0,115200 console=tty1 root=/dev/mmcblk0p2  
rootfstype=ext4 rootwait
```

第一个选项，**`dwc_otg.lpm_enable`**，告诉 Pi 要禁用其 USB 控制器，以防止问题时有适当的支持，在操作系统中启用的功能可能会出现对 Go (OTG) 模式。在大多数 Linux 发行版上，树莓派禁用此模式。

**`console`**选项告诉 Linux，它应该创建一个串行控制台——设备 **`ttyAMA0`**——和什么样的速度去运作。在大多数情况下，速度应该留在默认的 115,200 Kb/s（每秒千字节为单位）。如果正在使用 Pi 与较旧的设备进行通信，对应值可以相应降低。

**`Kgdboc`** 内核选项启用调试 Linux 内核的在使用控制台参数创建串行控制台。对于大多数用户，这是不必要的。对于开发人员来说，访问到内核调试通过串行连接是最有用的。很多发行版离开只是为了以防万一启用该选项。

第二个**`console`**创建设备 **`tty1`**，这是你看到当你第一次启动 Pi 的文本填充的屏幕。没有此项，您不能将东西连接到串行控制台创建的第一个控制台选项的情况下使用 Pi。

**`root`**选项告诉 Linux 内核它在哪里可以找到它的根文件系统中包含的所有文件和目录所需的系统操作。在默认的 Debian 分布，这是对第二个分区的 SD 卡——设备 **`mmcblk0p2`**。此选项可以改变到外部存储设备通过 USB，可以加快 Pi 较大幅度的运作,具有存储在 SD 卡上的根文件系统进行连接的地址。

最后，**`rootwait`**字符段告知内核其不仅仅是试图启动系统，，设备中还包含可行的root文件系统，没有这一选项时，树莓派会陷入慢启动，在相关的SD开启动之前准备好访问。

除了 **dwc\_otg** 设置，这些内核参数都独有的 Pi。任何 Linux 发行版本的引导加载程序配置将包括非常类似于 **cmdline.txt** 的选项的列表。

通常情况下，你最好不要编辑 **cmdline.txt** 文件。它由专门为该版本的 Linux 分布维护人员维护，不同版本该文件可能不同。在 Fedora 版本下可能混音有效的组件可能无法工作在 Debian 版本中，反之亦然。对 **cmdline.txt** 的可用选项取决于分布用上了什么样的内核和内核始建时包含哪些功能。

如果你是一个内核开发人员，您可以使用 **cmdline.txt** 来传递参数以用于启用或禁用你已经编译进内核的新功能。和之前的一样，**config.txt** 中的任何更改需要重新启动才能生效。