

RPiconfig "valeur" où valeur est un entier. On ne peut spécifier qu'une seule option par ligne. Des commentaires peuvent être ajoutés en commençant une ligne par le caractère '#'.
Dans les nouveaux modèles de Raspberry Pi, il y a un # avant chaque ligne, pour que les modifications aient un effet, alors 'décommenter' signifie supprimer le #.

Table of Contents



Format de fichier

Mémoire

CMA - répartition dynamique de la mémoire

Caméra

Réseau

Audio

Vidéo

Codecs sous licence

Boot

Overclocking

Rotation des horloges

Valeurs testées

Alerte de la température et de la tension**

Avertissement de sous tension

Test de stabilité d'overclocking

Alimentation USB

Arborescence des appareils

Activer ARMv8 sur RPi3B

Voici un exemple de fichier

```
# Règle le mode sdtv sur PAL (tel qu'il est utilisé en Europe)
sdtv mode=2
# Forcer le moniteur en mode HDMI pour que le son soit envoyé via le câble
HDMI
lecteur_hdmi=2
# Règle le mode moniteur sur DMT
groupe_hdmi=2
# Règle la résolution du moniteur sur 1024x768 XGA 60 Hz (HDMI_DMT_XGA_60)
hdmi mode=16
# Réduit l'affichage pour empêcher le texte de déborder de l'écran
overscan_left=20
```

Le Raspberry Pi n'a pas de BIOS conventionnel, les différents paramètres de configuration du système qui sont normalement conservés et définis à l'aide du BIOS sont désormais stockés dans un fichier texte nommé "config.txt".

Le fichier config.txt du Raspberry Pi est lu par le GPU avant l'initialisation du noyau ARM.

Mémoire

Ce fichier est un fichier facultatif sur la partition de démarrage. Il serait normalement accessible en tant que /boot/config.txt sur Linux, mais sur Windows (ou OSX), il sera considéré comme un fichier dans la partie accessible de la carte.

disable_12c	Désactive l'accès ARM au cache L2 du GPU (Nécessite le noyau désactivé h2 correspondant. Par défaut 0
gpu mem	Mémoire GPU en mégaoctets. Définit la répartition de la mémoire entre l'ARM et le GPU. ARM obtient la mémoire restante. Min 16. Par défaut 64
gpu_mem_256	Mémoire GPU en mégaoctet pour le Raspberry Pi de 256 Mo. Ignoré si la RAM n'est pas de 256 Mo. Remplace gpu_mem . Max 192. Par défaut non défini
gpu_mem_512	Mémoire GPU en mégaoctet pour le Raspberry Pi de 512 Mo. Ignoré si la RAM n'est pas de 512 Mo. Remplace gpu_mem . Max 448. Par défaut non défini
gpu_mem_1024	Mémoire GPU en mégaoctets pour Raspberry Pis avec 1024 Mo ou plus de mémoire. Ignoré si la RAM est inférieure à 1024 Mo. Remplace gpu_mem . Max 944. Par défaut non défini
disable_pvt	Désactive l'ajustement du taux de rafraîchissement de la RAM toutes les 500 ms (mesure de la température de la RAM).

CMA - répartition dynamique de la mémoire

Le firmware et le noyau en date du 19 novembre 2012 prennent en charge CMA, ce qui signifie que la répartition de la mémoire entre ARM et GPU est gérée dynamiquement au moment de l'exécution.

cma_lwm	Lorsque le GPU a moins de cma_lwm (low water mark) de mémoire disponible, il en demandera à ARM.
----------------	---

cma_hwm	Lorsque le GPU a plus que cma_hwm (high water mark) de mémoire disponible, il en libère une partie vers ARM.
----------------	---

Les options suivantes doivent être dans cmdline.txt pour que CMA fonctionne :

```
coherent_pool=6M smsc95xx.turbo_mode=N
```



selon <https://github.com/raspberrypi/linux/issues/503>, popcornmix indique que CMA n'est pas officiellement pris en charge.

Caméra

start_x	Activer le module caméra.	start_x=1
disable_camera_led	Éteindre le voyant rouge de la caméra lors de l'enregistrement d'une vidéo ou de la prise d'une photo	disable_camera_led=1
gpu_mem	mémoire GPU minimale pour l'utilisation de la caméra	gpu_mem=128

Réseau

smsc95xx.macaddr	Indique au pilote smsc95xx d'utiliser une adresse mac personnalisée au lieu d'utiliser l'adresse mac par défaut.	smsc95xx.macaddr=B8:AA:BC:DE:F0:12
-------------------------	--	------------------------------------

Audio

disable_audio_dither	Désactive le dithering ¹⁾ sur l'algorithme audio PWM. Essayer ceci si vous rencontrez des problèmes de bruit blanc sur la prise audio.	disable_audio_dither=1
-----------------------------	---	------------------------

Vidéo



L'ancien pilote **vc4-fkms-v3d** (driver Broadcom VideoCore 4 présent dans Raspberry Pi) ne sera plus pris en charge à l'avenir. Il s'appuie fortement sur des éléments spécifiques aux ordinateurs Pi et non directement pris en charge par Linux. Le nouveau pilote **vc4-kms-v3d** introduit avec Bullseye est privilégié, mais comme il est nouveau, il y aura des ajustements nécessaires (en particulier pour les personnes qui dépendaient des anciens pilotes et logiciels fkms).

Lorsque le nouveau pilote fonctionne mal, il faut utiliser: `dtoverlay=vc4-fkms-v3d`

Options du mode vidéo

sdtv_mode	définit la norme TV pour la sortie composite (par défaut=0)	sdtv_mode=0 NTSC normal sdtv_mode=1 version japonaise de NTSC sdtv_mode=2 PAL normal sdtv_mode=3 Version brésilienne de PAL - 525/60 plutôt que 625/50, sous-porteuse différente
sdtv_aspect	définit le rapport hauteur/largeur pour la sortie composite (par défaut=1)	sdtv_aspect=1 4:3 sdtv_aspect=2 14:9 sdtv_aspect=3 16:9
sdtv_disable_colourburst	désactive la rafale de couleur (color burst) sur la sortie composite. L'image sera monochrome, mais peut-être plus nette	sdtv_disable_colourburst=1 la rafale de couleurs est désactivée
hdmi_safe	Utilise les paramètres "mode sans échec" pour essayer de démarrer avec une compatibilité HDMI maximale. C'est la même chose que la combinaison de hdmi_force_hotplug=1, hdmi_ignore_edid=0xa5000080, config_hdmi_boost=4, hdmi_group=2, hdmi_mode=4, disable_overscan=0, overscan_left=24, overscan_right=24, overscan_top=24, overscan_bot	hdmi_safe=1
hdmi_ignore_edid	Permet d'ignorer les données EDID/d'affichage si l'affichage n'a pas d'EDID précis.	hdmi_ignore_edid=0xa5000080
hdmi_edid_file	lorsqu'il est défini sur 1, lira les données edid à partir du fichier edid.dat au lieu de celles du moniteur.	hdmi_edid_file=1
hdmi_force_edid_audio	Prétend que tous les formats audio sont pris en charge par l'affichage, permettant le passage de DTS/AC3 même lorsqu'ils ne sont pas signalés comme pris en charge.	hdmi_force_edid_audio=1
hdmi_ignore_edid_audio	Prétend que tous les formats audio ne sont pas pris en charge par l'affichage. Cela signifie que ALSA passera par défaut à l'analogique.	hdmi_ignore_edid_audio=1
hdmi_force_edid_3d	Prétend que tous les modes CEA prennent en charge la 3D même lorsque edid n'indique pas leur prise en charge.	hdmi_force_edid_3d=1
Avoid_edid_fuzzy_match	Evite la correspondance floue des modes décrits dans edid. Choisit le mode standard avec la résolution correspondante et la fréquence d'images la plus proche, même si la suppression est incorrecte.	Avoid_edid_fuzzy_match=1

hdmi_ignore_cec_init	N'envoie pas le message initial de la source active. Évite de sortir le téléviseur (compatible CEC) de la veille et du changement de chaîne lors du redémarrage.	hdmi_ignore_cec_init=1
hdmi_ignore_cec	Prétend que CEC n'est pas du tout pris en charge par la télévision. Aucune fonction CEC ne sera prise en charge.	hdmi_ignore_cec=1
hdmi_force_hotplug	Prétend que le signal HDMI hotplug est affirmé de sorte qu'il semble qu'un écran HDMI soit connecté	hdmi_force_hotplug=1 Utilise le mode HDMI même si aucun moniteur HDMI n'est détecté
hdmi_ignore_hotplug	Prétend que le signal de connexion à chaud HDMI n'est pas affirmé, il semble donc qu'un écran HDMI n'est pas connecté	hdmi_ignore_hotplug=1 Utilise le mode composite même si un moniteur HDMI est détecté
hdmi_pixel_encoding	Force le mode d'encodage des pixels. Par défaut, il utilisera le mode demandé par edid et ne devrait donc pas avoir besoin d'être modifié.	hdmi_pixel_encoding=0 par défaut (limité pour le CEA, complet pour le DMT) hdmi_pixel_encoding=1 RVB limité (16-235) hdmi_pixel_encoding=2 RVB plein (0-255) hdmi_pixel_encoding=3 YCbCr limité (16-235) hdmi_pixel_encoding=4 YCbCr plein (0-255)
hdmi_drive	choisit entre les modes HDMI et DVI	hdmi_drive=1 Mode DVI normal (pas de son) hdmi_drive=2 Mode HDMI normal (le son sera envoyé s'il est pris en charge et activé)
hdmi_group	Ne pas spécifier le groupe ou définir sur 0 utilisera le groupe préféré signalé par l'edid.	hdmi_group=1 CEA hdmi_group=2 DMT
hdmi_mode	définit la résolution de l'écran au format CEA ou DMT (pour consulter la liste des modes voir RPI: Accélération matérielle et GPU)	
overscan_left	nombre de pixels à ignorer à gauche	
overscan_right	nombre de pixels à sauter à droite	
overscan_top	nombre de pixels à sauter en haut	
overscan_bottom	nombre de pixels à ignorer en bas	
framebuffer_width	largeur du framebuffer de la console en pixels. La valeur par défaut est la largeur d'affichage moins le surbalayage.	
framebuffer_height	hauteur du framebuffer de la console en pixels. La valeur par défaut est la hauteur d'affichage moins le surbalayage.	

framebuffer_depth	Profondeur du framebuffer de la console en bits par pixel. La valeur par défaut est 16. 8 bits est valide, mais la palette RVB par défaut rend l'écran illisible. 24 bits semble mieux mais a des problèmes de corruption à partir de 20120615. 32 bits n'a pas de problèmes de corruption mais nécessite framebuffer_ignore_alpha=1 et affiche les mauvaises couleurs à partir de 20120615.	
framebuffer_ignore_alpha	défini sur 1 pour désactiver le canal alpha. Aide avec 32 bits.	
test_mode	permet de tester le son/l'image pendant le démarrage pour le test de fabrication.	
disable_overscan	défini sur 1 pour désactiver l'overscan.	
config_hdmi_boost	configure la force du signal de l'interface HDMI. La valeur par défaut est 2 sur le modèle Pi 1 B et 5 sur les cartes ultérieures. Essayer d'augmenter si vous rencontrez des problèmes d'interférence avec HDMI (par exemple jusqu'à 7) 11 est le maximum.	
display_rotate	fait pivoter l'affichage dans le sens des aiguilles d'une montre sur l'écran (par défaut = 0) ou inverse l'affichage ²⁾	display_rotate=0 Normal display_rotate=1 90 degrés display_rotate=2 180 degrés display_rotate=3 270 degrés display_rotate=0x10000 retournement horizontal display_rotate=0x20000 basculement vertical

Comment Déterminer les valeurs valables pour un moniteur ? : Un moniteur HDMI peut ne prendre en charge qu'un ensemble limité de formats. Pour savoir quels formats sont pris en charge, utiliser la méthode suivante:

- Régler le format de sortie sur VGA 60 Hz (hdmi_group=1 hdmi_mode=1) et démarrer le Raspberry Pi
- Entrer la commande suivante pour donner une liste des modes pris en charge par le CE:
/opt/vc/bin/tvservice -m CEA
- Entrer la commande suivante pour donner une liste des modes pris en charge par DMT:
/opt/vc/bin/tvservice -m DMT
- Entrer la commande suivante pour afficher votre état actuel:
/opt/vc/bin/tvservice -s
- Entrer les commandes suivantes pour vider des informations plus détaillées à partir de votre moniteur:
/opt/vc/bin/tvservice -d edid.dat
/opt/vc/bin/edidparser edid.dat





Le edid.dat doit également être fourni lors du dépannage des problèmes avec le mode HDMI par défaut

Codecs sous licence

Le décodage matériel de codecs supplémentaires peut être activé en achetant une licence verrouillée sur le numéro de série du processeur du Raspberry Pi.

Il faut affecter au moins 32 mégaoctets au GPU pour activer les codecs.

decode_MPG2	Clé de licence permettant le décodage matériel MPEG-2.	decode_MPG2=0x12345678
decode_WVC1	Clé de licence pour permettre le décodage matériel VC-1.	decode_WVC1=0x12345678

Configuration de la licence pour partager de carte SD entre plusieurs Raspberry Pis. Maximum de 8 licences à la fois.

```
decode_XXXX=0x12345678,0xabcdabcd,0x87654321,...
```

Boot

disable_commandline_tags	empêche start.elf de remplir ATAGS (mémoire à partir de 0x100) avant de lancer le noyau
cmdline (chaîne)	paramètres de ligne de commande. Peut être utilisé à la place du fichier cmdline.txt
kernel (string)	nom alternatif à utiliser lors du chargement du kernel. Par défaut "kernel.img"
kernel_address	adresse pour charger le fichier kernel.img
kernel_old (bool)	si 1, charge le noyau à 0x0
ramfsfile (string)	fichier ramfs à charger
ramfsaddr	adresse pour charger le fichier ramfs
initramfs (string address)	fichier ramfs et adresse pour le charger (c'est comme ramfsfile+ramfsaddr dans une option) ³⁾
device_tree_address	adresse pour charger device_tree
init_uart_baud	débit en bauds UART initial. Par défaut 115200
init_uart_clock	horloge UART initiale. Par défaut 3000000 (3 MHz)
init_emmc_clock	horloge eMMC initiale. Par défaut 100000000 (100 MHz)
boot_delay	attend un nombre donné de secondes dans start.elf avant de charger le noyau. délai = 1000 * boot_delay + boot_delay_ms. Par défaut 1
boot_delay_ms	attend un nombre donné de millisecondes dans start.elf avant de charger le noyau. Par défaut 0
Avoid_safe_mode	s'il est défini sur 1, le démarrage safe_mode ne sera pas activé. Par défaut 0
disable_splash	si défini sur 1, évite l'écran de démarrage arc-en-ciel au démarrage

Overclocking



La définition de paramètres autres que ceux fournis par « raspi-config » définira un bit permanent dans le SoC, ce qui permettra peut-être de détecter que le Raspberry Pi a été overclocké. Cela était destiné à annuler la garantie si l'appareil a été overclocké. Depuis le 19 septembre 2012, on peut overclocker le Raspberry Pi sans affecter sa garantie

Le dernier noyau a un pilote de noyau cpufreq avec le gouverneur "à la demande" activé par défaut. Cela n'a aucun effet si on n'a pas de paramètres d'overclocking. Mais lorsqu'on le fait, la fréquence arm varie en fonction de la charge du processeur. Les valeurs autres que celles par défaut ne sont utilisées que lorsque cela est nécessaire en fonction du gouverneur utilisé. On peut ajuster les valeurs minimales avec les options de configuration *_min ou désactiver la synchronisation dynamique avec force_turbo=1.

L'overclocking et la surtension seront désactivés au moment de l'exécution lorsque le SoC atteint 85 °C pour le refroidir. On ne doit pas atteindre la limite, même avec des réglages maximum à une température ambiante de 25 °C.

Options d'overclocking	Description des options	Valeur par défaut
arm_freq	Fréquence de l'ARM en MHz.	700
gpu_freq	Ensemble core_freq, h264_freq, isp_freq, v3d_freq.	250
core_freq	Fréquence du cœur du processeur GPU en MHz. Pour les modèles antérieurs au Pi2, cela a un impact sur les performances ARM car il pilote le cache L2. L'ARM sur le Pi2 a un cache L2 séparé.	250
h264_freq	Fréquence du bloc vidéo matériel en MHz.	250
isp_freq	Fréquence du bloc pipeline du capteur d'image en MHz.	250
v3d_freq	Fréquence du bloc 3D en MHz.	250
Avoid_pwm_pll	Ne dédie pas de pll à l'audio PWM. Cela réduira légèrement la qualité audio analogique. La PLL de rechange permet au core_freq d'être défini indépendamment du reste du GPU, ce qui permet plus de contrôle sur l'overclocking.	0
sdram_freq	Fréquence de la SDRAM en MHz. Par défaut 400	
over_voltage	ajustement de la tension du noyau ARM/GPU. [-16,8] équivaut à [0,8 V, 1,4 V] avec des pas de 0,025 V. Les valeurs supérieures à 6 ne sont autorisées que lorsque force_turbo ou current_limit_override sont spécifiés (ce qui définit le bit de garantie).	0 (1,2 V).
over_voltage_sdram	Définit over_voltage_sdram_c, over_voltage_sdram_i, over_voltage_sdram_p ensemble	
over_voltage_sdram_c	Réglage de la tension du contrôleur SDRAM. [-16,8] équivaut à [0,8 V, 1,4 V] avec des pas de 0,025 V.	0 (1,2 V)
over_voltage_sdram_i	Réglage de la tension des E/S SDRAM. [-16,8] équivaut à [0,8 V, 1,4 V] avec des pas de 0,025 V.	0 (1,2 V)
over_voltage_sdram_p	Réglage de la tension physique de la SDRAM. [-16,8] équivaut à [0,8 V, 1,4 V] avec des pas de 0,025 V.	0 (1,2 V)

Options d'overclocking	Description des options	Valeur par défaut
force_turbo	Désactive le pilote cpufreq dynamique et les paramètres minimaux ci-dessous. Active les options d'overclocking H.264/V3D/ISP. Peut définir le bit de garantie	0
initial_turbo	Active le mode turbo à partir du démarrage pour la valeur donnée en secondes (jusqu'à 60) ou jusqu'à ce que cpufreq définisse une fréquence.	0
arm_freq_min	Valeur minimale de arm_freq utilisée pour la synchronisation dynamique.	700
core_freq_min	Valeur minimale de core_freq utilisée pour la synchronisation dynamique.	250
sdram_freq_min	Valeur minimale de sdram_freq utilisée pour la synchronisation dynamique.	400
over_voltage_min	Valeur minimale de over_voltage utilisée pour la synchronisation dynamique.	0
temp_limit	Protection contre la surchauffe. Définit les horloges et les tensions par défaut lorsque le SoC atteint cette valeur Celsius. Un réglage supérieur à la garantie par défaut annule la garantie.	85
current_limit_override	Ceci est activé par défaut.	

force_turbo=0

permet des horloges et une tension dynamiques pour le cœur ARM, le cœur GPU et la SDRAM. Lorsqu'il est occupé, la fréquence ARM monte à **arm_freq** et descend à **arm_freq_min** au repos. **core_freq**, **sdram_freq** et **over_voltage** se comportent de la même manière. **over_voltage** est limité à 6 (1,35 V). Les valeurs autres que celles par défaut pour les parties H.264/V3D/ISP sont ignorées.

force_turbo=1

désactive la synchronisation dynamique, de sorte que toutes les fréquences et tensions restent élevées. L'overclocking des composants GPU H.264/V3D/ISP est autorisé ainsi que le réglage de **over_voltage** sur 8 (1,4 V).

Relation des horloges

Le cœur GPU, H.264, V3D et ISP partagent une PLL, donc doivent avoir des fréquences liées. ARM, SDRAM et GPU ont chacun leurs propres PLL et peuvent avoir des fréquences indépendantes.[9]

Ce qui suit n'est pas nécessaire avec "evoid_pwm_pll=1".

```
pll_freq = floor(2400 / (2 * core_freq)) * (2 * core_freq)
gpu_freq = pll_freq / [even number]
```

Le gpu_freq effectif est automatiquement arrondi à l'entier pair le plus proche, donc demander core_freq=500 et gpu_freq=300 donnera un diviseur de $2000/300 = 6,666 \Rightarrow 6$ et donc 333,33 MHz.

Valeurs testées

Le tableau suivant montre quelques tentatives réussies d'overclocking, qui peuvent être utilisées pour l'orientation. Ces paramètres peuvent ne pas fonctionner sur tous les appareils et peuvent raccourcir la durée de vie du SoC Broadcom.

arm_freq	gpu_freq	core_freq	h264_freq	isp_freq	v3d_freq	sdram_freq	over_voltage	over_voltage_sdram
800								
900	275					500		
900		450				450		
930	350					500		
1000		500				500	6	
1050							6	
1150		500				600	8	

Certains rapports indiquent que la RAM Hynix n'est pas aussi bonne que la RAM Samsung pour l'overclocking.

Surveillance de la température et de la tension**

- Pour surveiller la température du Raspberry Pi, regarder :
/sys/class/thermal/thermal_zone0/temp
- Pour surveiller la fréquence actuelle du Raspberry Pi, regarder :
/sys/devices/system/cpu/cpu0/cpufreq/scaling_cur_freq
- Pour surveiller la tension du bloc d'alimentation du Raspberry Pi, il faut utiliser un multimètre, entre les points de test d'alimentation ou l'en-tête d'extension.

C'est généralement une bonne idée de maintenir la température centrale en dessous de 70 degrés et la tension au-dessus de 4,8 V. (certaines alimentations USB, pas nécessairement bon marché, tombent à 4,2 V, car elles sont généralement conçues pour charger une batterie au lithium polymère de 3,7 V, plutôt que de fournir un solide 5 V à un ordinateur). De plus, un dissipateur thermique peut être utile, surtout si le Raspberry Pi doit être exécuté à l'intérieur d'un boîtier. Un dissipateur thermique approprié est le dissipateur thermique autocollant BGA (ball-grid-array) 14x14x10 mm, référence 674-4756 de RS Components.

Avertissement de sous tension

Le B+ a un déclencheur de détection de sous-tension qui entraîne l'extinction du voyant d'alimentation lorsque la tension chute en dessous d'environ 4,65 V. Le signal est également disponible sur une ligne gpio (GPIO35).

La dernière mise à jour du micrologiciel affichera un symbole d'avertissement en haut à droite de l'écran lorsque cela est détecté. Il désactivera également le mode turbo pendant que l'avertissement est affiché pour essayer de réduire les risques de plantage.

Il y a aussi un symbole d'avertissement à la condition de surchauffe existante (> 85°C), qui a également désactivé le mode turbo. Actuellement, les symboles sont :

- **Carré rouge:** surchauffe

- **Carré arc-en-ciel:** sous-tension

Comme d'habitude, vous pouvez remplacer le comportement dans config.txt si vous comprenez les risques :

```
Avoid_warnings=1 supprime la superposition d'avertissement.  
Avoid_warnings=2 autorise en plus le turbo lorsque la basse tension est  
présente.`</WRAP>
```

Test de stabilité d'overclocking

La plupart des problèmes d'overclocking apparaissent immédiatement avec un échec de démarrage, mais il est possible d'obtenir une corruption du système de fichiers qui survient au fil du temps. Voici un script pour tester la stabilité du système, en particulier la carte SD. Si ce script s'exécute jusqu'à la fin, sans qu'aucune erreur ne s'affiche dans dmesg, alors le Raspberry Pi est probablement stable avec ces paramètres.

Si le système plante, maintenir la touche Maj enfoncée pendant le démarrage, ce qui désactivera temporairement tout overclocking. Noter également que les problèmes de carte SD sont généralement affectés par **core_freq**, plutôt que par **arm_freq**, et qu'il y a un saut étonnamment important (de 250 MHz à 500 MHz) entre le haut débit (950 MHz) et le turbo (1 GHz) pré-réglages dans raspi-config.

```
#!/bin/bash  
#Test de stress simple pour le système. S'il survit à cela, il est  
probablement stable  
#Free software, GPL2+  
  
echo "Test de la stabilité de l'overclock..."  
  
#Maximise tous les cœurs du processeur. Le réchauffe, charge l'alimentation.  
for ((i=0; i<$(nproc --all); i++)); do nice yes >/dev/null & done  
  
#Lit l'intégralité de la carte SD 10x. Essais RAM et E/S  
for i in `seq 1 10`; do echo reading: $i; sudo dd if=/dev/mmcblk0  
of=/dev/null bs=4M; done  
  
#Écrit un fichier de test de 512 Mo, 10x.  
for i in `seq 1 10`; do echo writing: $i; dd if=/dev/zero of=deleteme.dat  
bs=1M count=512; sync; done  
  
#Nettoyage  
killall yes  
rm deleteme.dat  
  
#Imprimer le résumé. Tout ce qui est méchant apparaîtra dans dmesg..  
echo -n "CPU freq: " ; cat  
/sys/devices/system/cpu/cpu0/cpufreq/scaling_cur_freq
```

```
echo -n "CPU temp: " ; cat /sys/class/thermal/thermal_zone0/temp
dmesg | tail
```

```
echo "Pas encore planté, probablement stable."
```

Alimentation USB

max_usb_current Avec le Raspberry PI B+, un nouveau paramètre config.txt a été introduit.

```
max_usb_current=1
```

lors de l'ajout de cette ligne, le gestionnaire d'alimentation USB changera sa limite de courant de sortie (pour les 4 ports USB combinés) de 600 mA au double de celle de 1200 mA.

```
max_usb_current=0
```

est la valeur par défaut et limite le courant USB à 600 mA.

Il faut s'assurer que votre alimentation (et le cordon microUSB) sont à la hauteur de la tâche de fournir le courant supplémentaire de 0,6 A.

Arborescence des appareils

dtparam=i2c_arm=on	Active I2C sur les broches GPIO.
dtparam=i2s=on	Active le matériel audio I2S.
dtparam=spi=on	Active le pilote SPI.
dtoverlay=xxx	Ajoute une superposition /boot/overlays/xxx-overlay.dtb à l'arborescence des périphériques.

Activer ARMv8 sur RPi3B

Lors de l'exécution de **lscpu** ou **cat /proc/cpuinfo**, le processeur signalé est ARMv7. Pour dire au RPi3B de démarrer en mode ARMv8, il faut ajouter une nouvelle ligne à config.txt.

```
arm_control=0x200
```

1)

La diffusion d'erreur, **dithering** est une technique d'amélioration du rendu des images infographiques en cas de diminution du nombre de codes de couleurs. Elle se combine à l'anticrénelage.

2)

les options de rotation à 90 et 270 degrés nécessitent de la mémoire supplémentaire sur le GPU, elles ne fonctionneront donc pas avec la division GPU de 16 M. Probablement la raison de :

- Fait planter le Raspberry Pi avant le démarrage de Linux s'il est défini sur "1" – REW 20120913.
- **overscan_scale** est mentionné en tant que fonctionnalité du nouveau firmware.
- **overscan_scale=1** obligeait la sortie d'openFrameworks à respecter les paramètres *overscanleft*, *overscanright*, *top* et *bottom* avec l'utilisation d'un écran LCD connecté via composite.

3)

initramfs (string address) utilise une syntaxe différente de toutes les autres options - il ne faut pas utiliser le caractère "=" ici. Par exemple: `initramfs initramf.gz 0x00800000`. Les adresses valides dépendent de la taille du noyau. `0x00800000` fonctionne pour 3.6-trunk-rpi, `0x00a00000` fonctionne pour 3.14-1-rpi.

From:

<http://www.ouarte.garden/> - **dwndoc**

Permanent link:

<http://www.ouarte.garden/doku.php?id=prive:rpiconfig>Last update: **2025/02/19 10:59**