

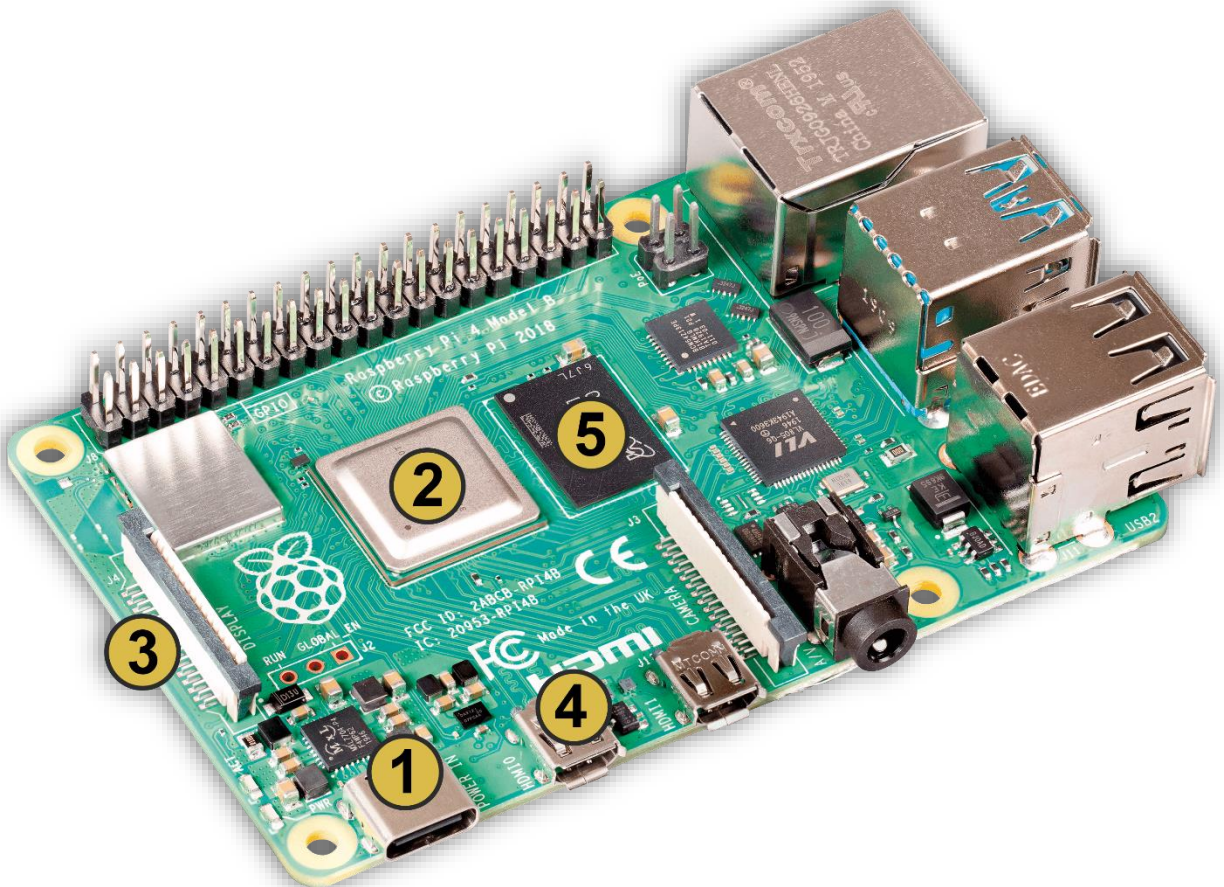
Tutoriel N°2 : Optimiser son Raspberry Pi

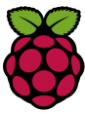
Bien que les Raspberry Pi deviennent de plus en plus puissants au fil des générations, il est quand même indispensable d'optimiser leurs fonctionnements et leur durée de vie. Ce tutoriel va regrouper toutes les techniques, autant matériels que logiciels, qui vont permettre de donner un second souffle aux plus vieux modèles mais également de les faire vivre le plus longtemps possible.

Avant toute chose, il est clair qu'une des méthodes pour redonner un coup de boost aux plus anciens modèles passent par l'overclocking (augmentation volontaire de la fréquence du processeur). Bien que je pratique cette opération assez souvent et sur presque tous mes Raspberry Pi, il est clair que cela réduit leurs durées de vie. Même en le faisant dans d'excellentes conditions, cela passe par de nombreuses étapes pour trouver la bonne valeur, chaque puce ayant ses propres « limites ». Je ne parlerai donc pas de cette méthode dans ce tutoriel mais ce sujet sera le thème d'un tutoriel complet très prochainement.

1. Introduction et présentation des axes d'optimisation

Les 5 axes d'amélioration vont cibler 5 parties précises du Raspberry Pi :





- 1 : L'alimentation
- 2 : Le processeur
- 3 : La carte micro-SD
- 4 : Le / les port(s) HDMI
- 5 : La mémoire vive (RAM)

Une chose importante à savoir, est que, comparé à un ordinateur classique, le Raspberry Pi n'a pas vraiment de BIOS ou d'UEFI. C'est le système, principalement, qui se chargera des fonctions de surveillance du nano-ordinateur, mais également du bootloader ou encore du firmware du GPU, qui seront exécutés au démarrage, depuis la carte Micro-SD.

2. L'alimentation électrique

Une mauvaise alimentation électrique sur un Raspberry Pi peut causer des problèmes divers et variés : baisse de fréquence du processeur, redémarrage intempestif, corruption des données, fin de vie prématurée de la carte Micro-SD, instabilité système...

Vu que le Raspberry Pi est vendu sans alimentation, il a vite tendance à se retrouver avec le vieux chargeur de téléphone qui traîne dans un tiroir, ce qui est, pour certain modèle, malheureusement très insuffisant.

Voici un repère simple pour savoir quelle puissance est nécessaire pour chaque modèle :

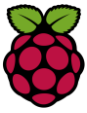
	Puissance	Voltage / Intensité	Connectique
1 ^{ère} génération	5W	5V 1A	Micro USB
2 ^{ème} génération	10W	5V 2A	Micro USB
3 ^{ème} génération	12,5 - 15W	5V 2,5 - 3A	Micro USB
4 ^{ème} génération	15W	5V 3A	USB-C

Les valeurs indiquées sont des valeurs de base. Evidemment, la consommation est dépendante du nombre d'éléments que vous allez brancher dessus que ce soit via les ports USB ou encore les ports GPIO.

En résumé, une bonne alimentation est **primordiale** et doit faire partie intégrante de votre réflexion en fonction du rôle que vous voulez donner à votre Raspberry Pi.

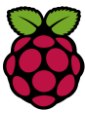
Pour information, le Raspberry Pi indique, lorsqu'il est branché à un écran, qu'il est sous-alimenté au moyen d'une icône présente en haut à droite de l'écran :

-  : sur les anciennes versions de Raspberry OS (Raspbian)



Tutoriel N°2 : Optimiser son Raspberry Pi

-  : sur les nouvelles versions de Raspberry OS (Raspbian)



Tutoriel N°2 : Optimiser son Raspberry Pi

Mais comment faire pour savoir si le Raspberry Pi est sous-alimenté si nous n'avons pas d'écran ?

Il existe heureusement un moyen simple via une ligne de commande. Il s'agit même, en fait, de la méthode la plus précise.

Ouvrez le terminal et tapez :

```
/opt/vc/bin/vcgencmd get_throttled
```

Cette commande va, en fonction des cas rencontrés, retourner plusieurs réponses possibles :

Code	Description
0X0	Aucune anomalie
0X1	Sous-alimentation détectée
0X2	Fréquence processeur réduite
0X4	Ralentissement du CPU : problème électrique ou de température
0X8	Limite de température atteinte
10000	Une sous-alimentation a eu lieu
20000	Une fréquence réduite du processeur a eu lieu
40000	Un ralentissement du CPU a eu lieu
80000	Une limite de température atteinte a eu lieu

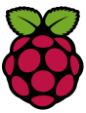
Ce qui est très pratique avec cette commande, c'est qu'en plus, les codes peuvent se cumuler. Si par exemple vous obtenez le retour : 30000 c'est que non seulement vous avez eu une sous-alimentation mais, qu'en plus, la fréquence du processeur a été volontairement réduite par le système.

Dans tous les cas, si vous avez un code erreur différent de 0X0, commencez par changer d'alimentation. Si vous continuez d'avoir un code retour différent, c'est qu'il y a un problème de température, nous aborderons ce sujet un peu plus loin.

En résumé, privilégiez en premier lieu une bonne alimentation et ne branchez pas de chargeur sur lequel aucune indication de puissance (volt ou ampère) n'est indiquée. Pour information, si la puissance en Watt n'est pas présente, il suffit de multiplier le voltage par l'intensité pour obtenir la puissance ($P = U \times I$).

3. Le microprocesseur

Depuis la toute première version du Raspberry Pi, le microprocesseur est devenu de plus en plus puissant, pour atteindre le quad-core (ArmV8) 64-bit de 1,8 Ghz présent sur le Raspberry Pi 400 (le seul actuellement avec cette fréquence).



Tutoriel N°2 : Optimiser son Raspberry Pi

Cette montée en puissance du processeur a engendré, indubitablement, une montée en température. Le Raspberry Pi 4 a beaucoup été critiqué à sa sortie sur ce point.

Il faut savoir qu'une température élevée du processeur entrainera également beaucoup de problème : limite de fréquence du processeur (underclocking), redémarrage, corruption des données sur la carte. Bref, comme vous le voyez, presque les mêmes conséquences qu'un problème d'alimentation.

Il est donc primordial de refroidir correctement le processeur du Raspberry Pi. Encore une fois, la solution à préconiser est dépendante du rôle que vous allez lui faire endosser. Si celui-ci est voué à effectuer beaucoup de calculs de façon constante, il vaut mieux partir sur une solution de refroidissement active, sinon, une solution passive peut généralement être suffisante.

Il existe également des boîtiers qui permettent une meilleure dissipation de la chaleur.

Maintenant, comment savoir si la solution choisie est suffisante ?

Dans un premier temps, nous allons afficher la température actuelle du processeur. Ouvrez un terminal et tapez la commande suivante :

```
cat /sys/class/thermal/thermal_zone0/temp
```

Il faudra diviser la valeur par 1000 pour obtenir la température en °C.

Vous pouvez aussi exécuter la commande suivante pour obtenir la valeur directement en °C (moins précise) :

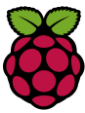
```
vcgencmd measure_temp
```

La commande que nous avons vu dans le chapitre précédent nous permettra de savoir si une limite de fréquence du processeur a eu lieu à cause d'une température trop élevée.

D'après mon expérience, il est tout à fait possible de laisser le Raspberry Pi avec un refroidissement passif sur les générations 1, 2 et 3. La 4^{ème} nécessite généralement un refroidissement actif mais certains boîtiers arrivent à proposer un refroidissement passif de qualité.

Je vous recommande les boîtiers suivants :

- Boîtier « Argon Neo » avec refroidissement passif pour RPI4
- Boîtier « Argon One » avec refroidissement actif pour RPI4



4. La carte Micro-SD

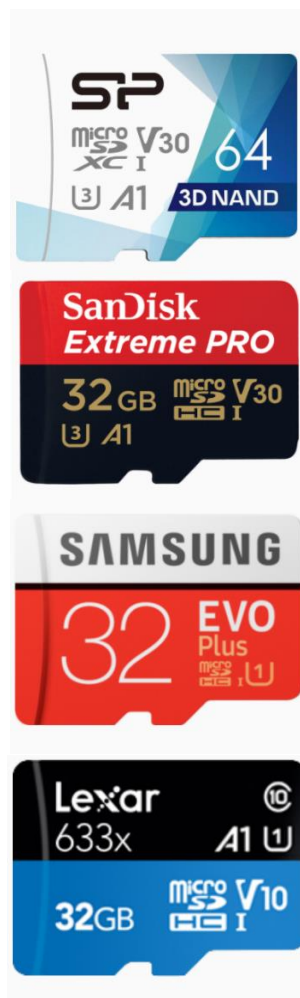
a. Quel matériel choisir ?

La carte Micro-SD est, de loin, l'élément le plus sensible de toute l'installation. Une simple mauvaise extinction peut corrompre l'intégralité des données présentes sur celle-ci. Il est donc très important de choisir une carte avec des caractéristiques bien précises. Au-delà des problèmes qui peuvent survenir en choisissant une mauvaise carte, les performances de la machine peuvent être grandement réduite à cause d'une carte de piètre qualité.

La meilleure solution, mais plus couteuse, reste de remplacer cette carte par un autre type de stockage : clé USB, disque dur mécanique, SSD ou encore M.2 SSD. Les possibilités sont grandes et les performances qui en découlent également.

Nous ne verrons pas dans ce tutoriel l'installation sur ces types de stockage, car ils feront l'objet d'un tutoriel complet.

Revenons-en à nos cartes Micro-SD.



Silicon Power 3D NAND

Prix : Environ 12€ pour 32 Go

SanDisk Extreme Pro

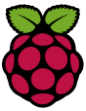
Prix : Environ 20€ pour 32 Go

Samsung Evo Plus

Prix : Environ 10€ pour 32 Go

Lexar 633x

Prix : Environ 10 € pour 32 Go



Tutoriel N°2 : Optimiser son Raspberry Pi

Cette liste est très subjective car elle a été sélectionnée à partir de plusieurs sites internet ayant effectués des tests sur des dizaines de cartes. Celles que j'ai choisi sont celles les plus fréquemment citées dans les résultats de ces tests.

Globalement, les éléments à prendre en compte sont :

- *La vitesse maximum en lecture*
- *La vitesse maximum en écriture*
- *La capacité*
- *Le type de mémoire flash utilisé*

Certaines de ces informations sont parfois dures à trouver, notamment la *vitesse maximum en écriture* qui est rarement indiquée. Le mieux reste encore de se renseigner sur des sites spécialisés pour comparer au mieux possible. Généralement, les cartes dédiées à la vidéo 4K sont de très bonnes cartes.

b. Comment bien entretenir sa carte Micro-SD ?

- o Logs systèmes

Autant la première partie de ce chapitre est très subjective, autant quand il s'agit d'entretien et d'optimisation, des solutions précises existent.

Un des premiers axes d'amélioration sera de diminuer autant que possible le volume de données écrit sur la carte. En effet, ces accès très fréquents réduisent drastiquement la durée de vie de celle-ci. Le champion toute catégorie sur ce point : *les logs système*.

La solution la plus efficace est de supprimer l'écriture de tous les logs système. C'est brutal, mais redoutable !

Pour effectuer cette opération, ouvrez un terminal et saisissez la commande suivante :

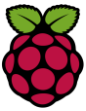
```
sudo service rsyslog stop
```

Cette commande va stopper pour la durée de la session, le service d'écriture des logs. Cependant, si vous redémarrez le Raspberry Pi, le service redémarrera également. Pour éviter qu'il ne s'exécute à chaque démarrage, saisissez la commande suivante :

```
sudo systemctl disable rsyslog
```

Pour réactiver le démarrage automatique à nouveau :

```
sudo systemctl enable rsyslog
```

Tutoriel N°2 : Optimiser son Raspberry Pi

Afin d'éviter de désactiver la totalité des logs systèmes, il est possible de configurer le service de gestion des logs de la façon que vous souhaitez. Pour éditer cette configuration, saisissez la commande suivante dans la fenêtre de votre terminal :

```
sudo nano /etc/rsyslog.conf
```

Cette commande va démarrer dans l'éditeur de texte « *nano* », l'édition du fichier « *rsyslog.conf* ».

La compréhension et la modification de ce fichier est, cependant, très complexe. Je vous déconseille de le modifier si vous ne savez pas ce que vous faites. Je vous donne juste une piste à suivre si vous souhaitez faire une configuration précise des logs du système. Si vous souhaitez plus d'informations à ce sujet, faites une recherche avec le mot clé « *rsyslog* ». Sachez que dans la plupart des utilisations fréquentes répandues, l'analyse des logs n'est pas vraiment utile et qu'il n'est pas forcément nécessaire de passer du temps sur ce point.

○ Taille de la partition système

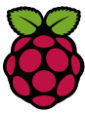
Il faut savoir que, lorsque vous installez votre système, la taille de la partition système utilisée n'est pas du tout celle de la carte Micro-SD (environ 2Go). Si aucune opération n'est faite pour la modifier, vous ne pourrez jamais utiliser la capacité totale de la carte. C'est quand même dommage si vous investissez dans une carte avec une grande capacité !

Il est toutefois possible de modifier cette taille très simplement et de l'étendre à la capacité totale de la carte.

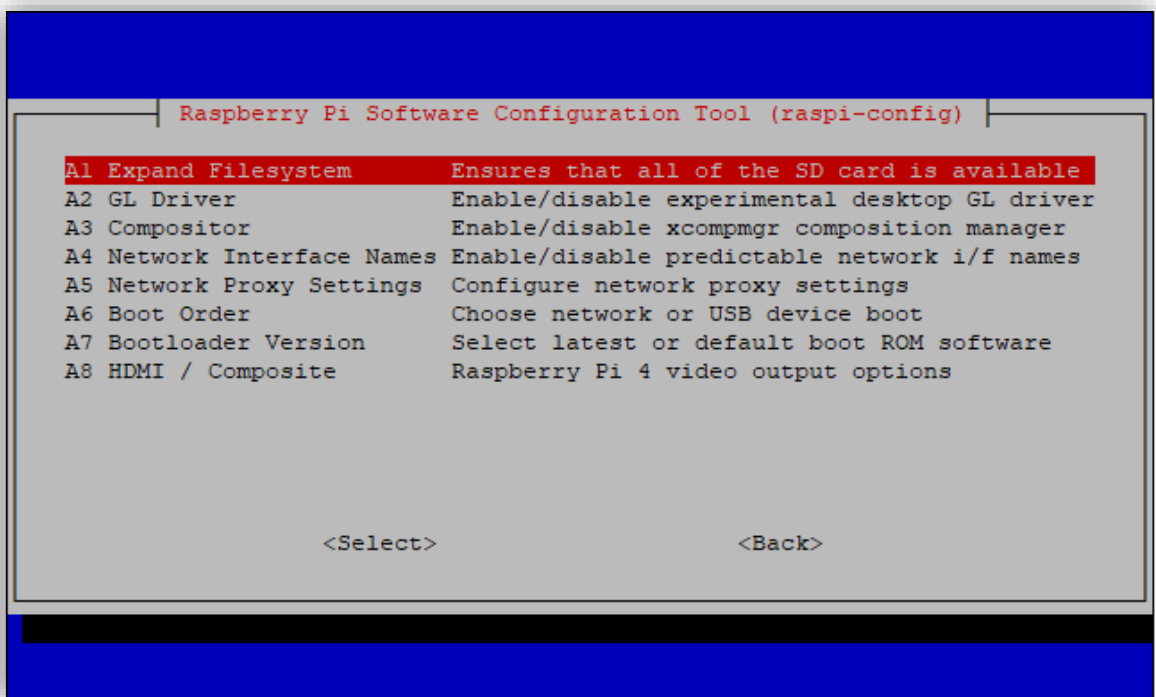
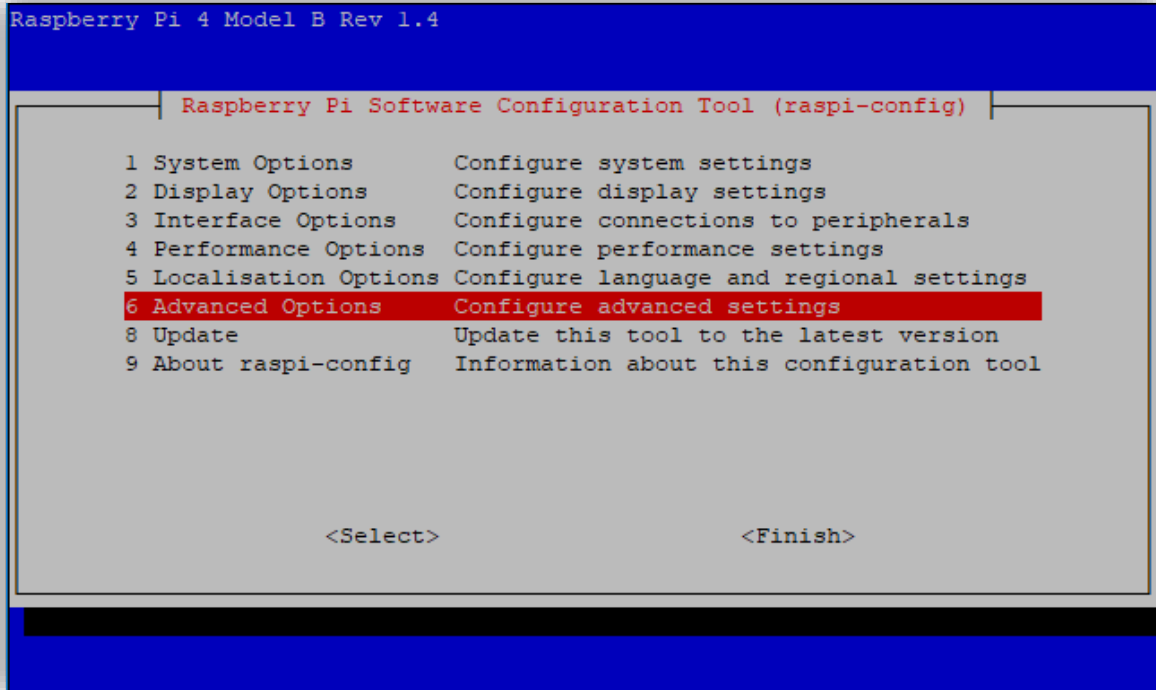
Pour effectuer cette opération, ouvrez le terminal et saisissez la commande :

```
sudo raspi-config
```

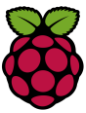
Cette commande aura pour but d'ouvrir l'utilitaire de configuration du Raspberry Pi qui affichera cet écran :



Tutoriel N°2 : Optimiser son Raspberry Pi



Sélectionnez « *Advanced Options* » puis « *Expand FileSystem* ». Un écran de confirmation s'affichera :



Vous pouvez maintenant utiliser la totalité de l'espace disponible sur votre carte.

- Supprimer le « Swap »

Nous avons vu que les logs systèmes étaient les champions toutes catégories en termes d'usure de carte micro-SD. Cependant, il a un redoutable adversaire : le Swap. Pour faire simple, le Swap est une « mémoire virtuelle » présente sur l'espace de stockage qui prendra le relai si la mémoire physique (RAM) arrive à saturation. Habituellement, le Swap est présent sous la forme d'une partition spécifique sur l'espace de stockage. Sous Raspberry OS, il est présent sous la forme d'un fichier présent dans « /var ».

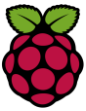
Nous verrons plus tard, comment améliorer la gestion de la mémoire du Raspberry Pi, mais en attendant, nous allons nous débarrasser définitivement du Swap pour qu'aucune opération mémoire ne soit écrite sur la carte Micro SD.

Pour effectuer cette opération, saisissez la commande suivante dans un terminal :

```
sudo apt-get remove dphys-swapfile && apt-get autoremove
```

Cette commande va désinstaller « *dphys-swapfile* » qui gère le Swap et procéder au nettoyage des dépendances.

Le Swap étant désactivé, il ne reste qu'à supprimer le fichier dédié de la carte :



```
sudo rm -f /var/swap
```

Puis, faites un redémarrage du système avec la commande :

```
sudo reboot
```

Le Swap n'est maintenant qu'un vieux souvenir.

5. Les ports HDMI

Vous allez me dire, pourquoi irions nous optimiser les ports HDMI ? Et bien c'est simple, dans la majeure partie des cas d'utilisation d'un Raspberry Pi, le ou les ports HDMI ne sont... jamais utilisés. Pourquoi donc faudrait-il continuer de les alimenter ? Pour information, un port HDMI non utilisé consomme en moyenne 20 à 25 mA, il serait donc dommage de perdre cette énergie.

Si vous aussi vous n'avez pas d'écran branché à votre Pi, voici la comment permettant de désactiver les ports HDMI :

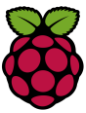
```
/usr/bin/tvservice -o
```

Cette commande désactivera les ports HDMI pour la durée de la session. Pour définitivement couper leur alimentation, il faudra éditer le fichier « *rc.local* » pour que le système exécute cette commande au démarrage.

Procédez de la façon suivante :

```
sudo nano /etc/rc.local
```

Puis, insérez la même commande que pour la désactivation « */usr/bin/tvservice -o* » juste avant la ligne « *exit 0* ». Vous devriez obtenir ceci :



```
GNU nano 3.2 /etc/rc.local Modified
#
# By default this script does nothing.
# Print the IP address
_IP=$(hostname -I) || true
if [ "$_IP" ]; then
    printf "My IP address is %s\n" "$_IP"
fi

/usr/bin/tvservice -o
exit 0

^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File ^\ Replace   ^U Uncut Text ^T To Spell  ^_ Go To Line
```

Faites **CTRL + X** puis **Y** pour sauvegarder et quitter l'éditeur de texte.

Les ports HDMI seront automatiquement désactivés à chaque démarrage.

6. La mémoire vive (RAM)

a. Réduction de la mémoire graphique partagée

Il y a également quelques optimisations à faire du côté de la mémoire vive. En effet, il faut savoir que le processeur graphique du Raspberry Pi utilise une partie de la RAM en tant que mémoire « partagée » pour gérer la partie graphique. Par défaut, sur le Raspberry Pi, elle est de 64Mo.

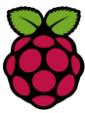
Dans le cas où, comme précédemment, vous n'avez pas d'écran ou pas d'interface graphique, il est totalement inutile de conserver cette mémoire partagée. Nous ne pourrions pas la supprimer totalement, mais nous allons la réduire à 16Mo.

Pour effectuer cette opération, saisissez dans votre terminal :

```
sudo raspi-config
```

Encore une fois, nous nous retrouverons dans la configuration du Raspberry Pi.

Commencez par aller dans le menu suivant :



Tutoriel N°2 : Optimiser son Raspberry Pi

```
Raspberry Pi 4 Model B Rev 1.4

Raspberry Pi Software Configuration Tool (raspi-config)

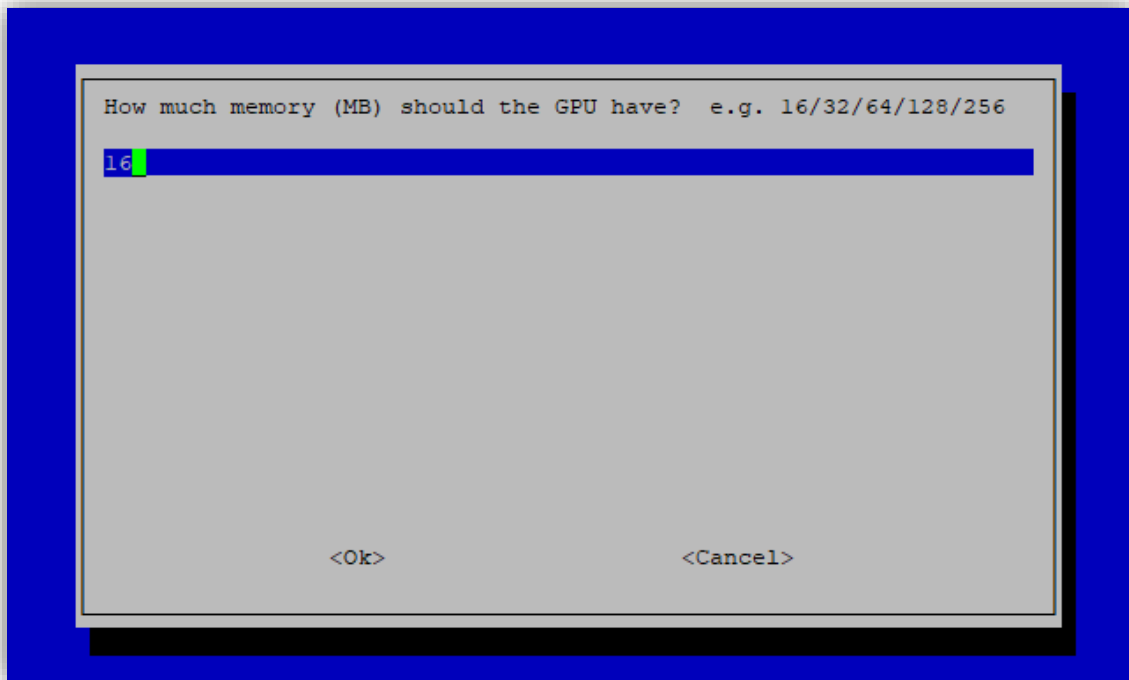
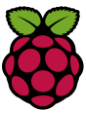
1 System Options      Configure system settings
2 Display Options     Configure display settings
3 Interface Options   Configure connections to peripherals
4 Performance Options Configure performance settings
5 Localisation Options Configure language and regional settings
6 Advanced Options    Configure advanced settings
8 Update              Update this tool to the latest version
9 About raspi-config  Information about this configuration tool

<Select>                <Finish>
```

```
Raspberry Pi Software Configuration Tool (raspi-config)

P1 Overclock          Configure CPU overclocking
P2 GPU Memory          Change the amount of memory made available to the GPU
P3 Overlay File System Enable/disable read-only file system
P4 Fan                 Set behaviour of GPIO fan

<Select>                <Back>
```



Saisissez « 16 » à la place de « 64 ». Sachez que vous pouvez également procéder à l'opération l'inverse. Si vous souhaitez disposer de plus de mémoire graphique, vous pouvez l'étendre jusqu'à 256Mo en saisissant « 256 ».

Il est nécessaire de redémarrer le Raspberry Pi pour que cette modification soit prise en compte.

Vous commencez à connaître la commande :

```
sudo reboot
```

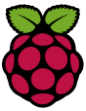
b. Optimisation de la gestion de la mémoire :

Maintenant que nous n'avons plus de Swap et que nous avons la quasi-totalité de la mémoire vive disponible, nous allons pouvoir optimiser son fonctionnement. Nous allons utiliser « zRam ».

« zRAM », de son ancien nom « compcache », est un module utilisé pour créer des « blocs » compressés en mémoire, ou « disques virtuels. Les blocs créés avec « zRAM » pourront alors être alloué à l'utilisation d'un Swap ou bien comme emplacement de RAM classique. La compression des données en mémoire sera faite à la volée.

Dans votre terminal, saisissez la commande suivante :

```
sudo wget -O /usr/bin/zram.sh  
https://raw.githubusercontent.com/novaspirit/rpi\_zram/master/zram.sh
```



Tutoriel N°2 : Optimiser son Raspberry Pi

Si vous obtenez un message d'erreur indiquant « *wget – unknown command* », saisissez cette nouvelle commande :

```
sudo apt-get install wget
```

Puis, après l'opération, ressaisissez la commande précédente.

Nous avons téléchargé un script qui démarrera zRam lors de son exécution, mais il faut tout d'abord le rendre exécutable grâce à cette commande :

```
sudo chmod +x /usr/bin/zram.sh
```

Nous allons maintenant demander au système d'exécuter ce script à chaque démarrage du Raspberry Pi. Nous allons procéder de la même façon que précédemment, en éditant le fichier « *rc.local* ».

Saisissez la commande suivante dans la fenêtre du terminal :

```
sudo nano /etc/rc.local
```

Vous devrez, comme pour la désactivation du port HDMI au démarrage, ajouter la ligne suivante « */usr/bin.zram.Sh &* ». Vous devriez obtenir ceci si vous avez exécuté les deux opérations :

```
GNU nano 3.2 /etc/rc.local Modified

#
# By default this script does nothing.

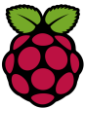
# Print the IP address
_IP=$(hostname -I) || true
if [ "$_IP" ]; then
    printf "My IP address is %s\n" "$_IP"
fi

/usr/bin/zram.sh &
/usr/bin/tvservice -o

exit 0

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line
```

Appuyez sur **CTRL + X** pour quitter puis sur **Y** pour sauvegarder.



Tutoriel N°2 : Optimiser son Raspberry Pi

Faites un dernier redémarrage du système.

```
sudo reboot
```

Votre Raspberry Pi est maintenant complètement optimisé 😊