

## Slackware ARM sur Raspberry Pi 3

Le Raspberry Pi 3 a un processeur Broadcom BCM2837 SoC qui int gre un CPU Quad-core ARMv8 Cortex-A53 [64 bit] @ 1.2GHz et un GPU VideoCore IV @ 400MHz, et il arrive avec une carte SDRAM 1GB LPDDR2 @ 900MHz. Cette carte m re r vis e et mise   jour succ de au [Raspberry Pi \(2\)](#); elle est consid rablement plus rapide et beaucoup plus puissante. Le Wi-Fi et le Bluetooth sont maintenant fournis sur la carte. Pas encore de RTC cependant. Voyez le RPI3 comme une r novation plut t que comme une innovation. Slackware ARM, comme vous pouvez vous y attendre, tourne sans souci sur cet appareil avec une vitesse nettement accrue par rapport aux versions pr c dentes. Les temps de compilation sont beaucoup plus courts par rapport au RPI2 par exemple.

Le Raspberry Pi 3 est support  en dehors de la branche officielle de Slackware ARM par la communaut  Slackware.

### Slackware version 14.2, -current

Slackware ARM current ou Slackware ARM 14.2 peuvent  tre install s sur le Raspberry Pi 3.

Suivez les liens dans la table ci-dessous. Ils sont maintenus par un d veloppeur s par  qui fait partie de la communaut  Slackware autour du Raspberry Pi.

Site	Versions Slackware	Utilisation des paquetages Slackware officiels	M�thodes d'installation	Notes
<a href="#">FatDog</a>	14.2,-current	Oui	Installateur Slackware	Un guide complet d'installation et de param�trage

Aussi longtemps que vous utilisez l'image Raspbian et le firmware les plus r cents la [m thode d'installation manuelle du Raspberry Pi 1](#) marchera  galement pour le Pi2 et le Pi3.

### Installation manuelle sans image Raspbian

Cette m thode permet d'installer Slackware ARM 14.2 sur un Raspberry Pi 3 Model B qui n'a pas d'image Raspbian. Cependant elle doit aussi fonctionner pour d'autres versions de Slackware ARM et de Raspberry Pi.

#### 1. Partitionnement et formatage de la carte SD

```
$ sudo fdisk -l /dev/mmcblk0
```

```
Disk /dev/mmcblk0: 31.9 GB, 31914983424 bytes
4 heads, 16 sectors/track, 973968 cylinders, total 62333952 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
```

Disk identifier: 0x00000000

Device	Boot	Start	End	Blocks	Id	System
/dev/mmcblk0p1		2048	67583	32768	b	W95 FAT32
/dev/mmcblk0p2		67584	62333951	31133184	83	Linux

```
$ sudo mkfs.vfat /dev/mmcblk0p1  
$ sudo mkfs.ext4 /dev/mmcblk0p2
```

Remarques:

- J'utilise une carte SD de 32 Go
- Je choisis 32 Mo comme taille de la première partition
- Je laisse tout l'espace vide pour la seconde partition

## 2. Mettre le firmware du Raspberry Pi sur la carte SD

```
$ git clone https://github.com/raspberrypi/firmware.git  
$ sudo mount /dev/mmcblk0p1 ~/mnt  
$ sudo cp -r firmware/boot/* ~/mnt  
$ sudo umount ~/mnt  
$ sudo mount /dev/mmcblk0p2 ~/mnt  
$ sudo mkdir -p ~/mnt/lib/modules  
$ sudo cp -r firmware/modules/* ~/mnt/lib/modules  
$ sudo umount ~/mnt
```

## 3. Mettre le filesystem mini root de Slackware ARM sur la carte SD

```
$ wget -c  
ftp://ftp.arm.slackware.com/slackwarearm/slackwarearm-devtools/minirootfs/roots/slack-14.2-miniroot_01Jul16.tar.xz  
$ sudo mount /dev/mmcblk0p2 ~/mnt  
$ sudo tar -C ~/mnt -xf slack-14.2-miniroot_01Jul16.tar.xz  
$ echo "/dev/mmcblk0p1 /boot vfat defaults 0 0" | sudo tee ~/mnt/etc/fstab  
$ echo "/dev/mmcblk0p2 / ext4 defaults 0 0" | sudo tee -a  
~/mnt/etc/fstab  
$ echo "proc /proc proc defaults 0 0" | sudo tee -a  
~/mnt/etc/fstab  
$ PASSWD=$(openssl passwd -1 -salt cetkq/enZx6/c2 password)  
$ sudo sed -i "s|\\(root:\\).*\\(:16983:0:::::)|\\1${PASSWD}\\2|"  
~/mnt/etc/shadow  
$ sudo sed -i 's|USE_DHCP\\[1\\]=""|USE_DHCP\\[1\\]="yes"|'  
~/mnt/etc/rc.d/rc.inet1.conf  
$ echo "PermitRootLogin yes" | sudo tee -a ~/mnt/etc/ssh/sshd_config  
$ sudo umount ~/mnt
```

Remarques:

- Je mets "password" comme mot de passe pour l'utilisateur "root"

- J'active DHCP sur l'interface réseau "eth1"
- J'autorise l'utilisateur "root" à se connecter par SSH

## 4. Insérer la carte SD dans le Raspberry Pi

Votre carte SD est prête et vous pouvez l'insérer dans le Raspberry Pi et le démarrer.

Vous pouvez vous connecter à distance sur votre Raspberry Pi par SSH en tant que "root".

```
$ ssh root@raspberrypi
```

Dès que vous êtes connecté, vous pouvez installer les paquetages additionnels de Slackware ARM:

```
$ wget --mirror ftp://ftp.arm.slackware.com/slackwarearm/slackwarearm-14.2
$ upgradepkg --install-new
ftp.arm.slackware.com/slackwarearm/slackwarearm-14.2/slackware/*/*.txz
$ removepkg
ftp.arm.slackware.com/slackwarearm/slackwarearm-14.2/slackware/*/kernel_*.txz
```

- Je considère que nom d'hôte du Raspberry Pi est "raspberrypi"
- Je recommande de choisir un utilisateur normal plutôt que "root"
- Je recommande de changer le mot de passe de "root"
- Je recommande d'interdire à "root" de se connecter par ssh
- Je recommande de [créer vos propres paquetages du noyau Linux](#) car le noyau que vous utilisez ne coïncide pas avec les paquetages installés de Slackware ARM

## 5. Trucs et astuces

### 5.1. Processeur

Le processeur du Raspberry Pi peut atteindre 1.2GHz. Cependant, par défaut, il est bridé à 600MHz même s'il est utilisé à 100 %. Vous pouvez vérifier la fréquence courante du processeur en tapant:

```
$ cpufreq-info
```

Pour atteindre 1.2GHz quand le processeur est utilisé à 100 % ( autrement dit pour faire de l'adaptation de fréquence ), il faut changer les "gouverneurs" par défaut. Ajoutez cette ligne à la fin du fichier /etc/rc.d/rc.local:

```
echo ondemand | sudo tee
/sys/devices/system/cpu/cpu*/cpufreq/scaling_governor
```

Maintenant le processeur est correctement configuré.

## 5.2. Temps

Malheureusement, le Raspberry Pi ne possède pas d'horloge temps réel (RTC). C'est la raison pour laquelle il n'y a pas de pile sur la carte mère. Cela veut dire que chaque fois que vous éteignez le Raspberry Pi, l'horloge est remise à 0! Cependant, si vous un accès Internet, vous pouvez mettre l'horloge à l'heure durant le démarrage de Slackware ARM. Ajoutez la ligne suivante à la fin du fichier `/etc/rc.d/rc.local`:

```
ntpdate pool.ntp.org
```

Maintenant, l'horloge est à la bonne heure.

## 5.3. Video

Malheureusement, le Raspberry Pi n'est pas compatible OpenGL (il est compatible OpenGL ES qui est un sous-ensemble d'OpenGL). Cela veut dire que, par défaut, toutes les applications qui ont besoin d'OpenGL seront lentes. Cependant, vous pouvez atteindre 60 images par seconde pour des applications OpenGL sur le Raspberry Pi en utilisant le bon pilote.

En premier, il faut construire Mesa ( $\geq 17.0.4$ ) avec le pilote VC4 DRI:

```
$ CFLAGS="-O2 -march=armv8-a -mtune=cortex-a53 -mfloat-abi=softfp -  
mfpu=neon-vfpv4" \  
  CXXFLAGS="-O2 -march=armv8-a -mtune=cortex-a53 -mfloat-abi=softfp -  
mfpu=neon-vfpv4" \  
  ./configure \  
  --prefix=/usr \  
  --sysconfdir=/etc \  
  --with-dri-driverdir=/usr/lib/xorg/modules/dri \  
  --with-egl-platforms=x11,drm \  
  --with-gallium-drivers=vc4  
$ make -j4  
$ make install DESTDIR=/where/you/want/to/install
```

Ensuite, fabriquez votre propre paquetage Mesa pour Slackware ARM et installez le.

Deuxièmement, ajoutez la ligne suivante à la fin du fichier `/boot/config.txt`:

```
dtoverlay=vc4-fkms-v3d
```

Puis redémarrez le Raspberry Pi. Vous pouvez vérifier que vous pouvez atteindre les 60 images par seconde pour des applications OpenGL sur le Raspberry Pi en tapant la commande suivante dans un terminal X:

```
$ glxgears
```

Désormais la vidéo est correctement configurée.

## Sources

- Originally written by [Exaga](#)
- Contributions by [yugiohj](#)
- Traduction française par [desiderius](#)

[howtos](#), [hardware](#), [arm](#), [author exaga](#)

From:

<https://docs.slackware.com/> - **SlackDocs**

Permanent link:

<https://docs.slackware.com/fr:howtos:hardware:arm:raspberrypi3>

Last update: **2017/07/21 09:45 (BST)**

