

INSA TOULOUSE

PROJET TUTORÉ

SEMESTRE 8 4IR-I

---

# Installation de Xenomai sur Raspberry Pi 3

---

*Etudiants:*

Alexandre BENAZECH,  
Vincent PERA et  
Romain RIVIÈRE

*Tuteurs:*

Ion HAZYUK,  
Claude BARON



## Abstract

Ce document s'appuie sur la démarche de Christophe Blaess dans son article *"Xenomai sur Raspberry Pi 3 : bon espoir mais bilan mitigé"* [1]. Au lieu de cross-compiler un noyau pour le Raspberry on installe une distribution raspbian directement sur le raspberry. On compile ensuite le noyau Xenomai directement sur le raspberry, rendant l'installation plus simple mais avec une compilation plus longue que sur un PC.

On récupère tout d'abord une image Raspbian sur le site officiel [2]. Ici la version utilisée est celle du 26 Février 2017. Il ne faut pas utiliser de distribution Noob. La distribution choisie est la dernière en date, soit **2017-04-10-raspbian-jessie-lite**.

On flash la carte sd avec Win32DiskImager si on est sous windows ou avec la commande suivante sous Linux.

```
1 $ sudo dd if=2017-02-16-raspbian-jessie-lite.img of=/dev/sdc bs=4M
```

On branche la raspberry à un clavier et un écran et on peut se connecter avec les identifiants suivants.

```
2 raspberrypi login: pi
Password: raspberry
```

On active le ssh [3](optionnel)

1. Entrer **sudo raspi-config** dans le terminal
2. Sélectionner **Interfacing Options**
3. Sélectionner **SSH**
4. Choisir **Yes**
5. Sélectionner **OK**
6. Choisir **Finish**

Il est aussi possible de passer le clavier en azerty depuis **raspi-config**.

On installe quelques outils supplémentaires.

```
pi@raspberrypi:~ $ sudo apt install -y git ncurses-dev bc autoconf libtool
```

On récupère la version 3.0.3 de Xenomai et on affiche nos versions de ipipe

```
1 pi@raspberrypi:~$ wget http://xenomai.org/downloads/xenomai/stable/
  xenomai-3.0.3.tar.bz2
pi@raspberrypi:~$ tar xjf xenomai-3.0.3.tar.bz2
3 pi@raspberrypi:~$ ls xenomai-3.0.3/kernel/cobalt/arch/arm/patches/
```

On récupère le patch ipipe.

```
1 pi@raspberrypi:~$ git clone https://github.com/raspberrypi/linux.git -b
  rpi-4.1.y --depth 1
```

```

pi@raspberrypi:~$ head -4 linux/Makefile
3 VERSION = 4
  PATCHLEVEL = 1
5 SUBLEVEL = 21
  EXTRAVERSION =
7 pi@raspberrypi:~$

```

Le noyau est un 4.1.21, le patch ipipe est pour un noyau 4.1.18. Il y a peu de différences entre les deux mais il est quand même nécessaire d'appliquer un patch fourni par Christophe Blaess. On patch un patch.

```

1 pi@raspberrypi:~$ wget https://www.blaess.fr/christophe/files/article
  -2017-03-20/001-adapt-4.1.18-patch-to-rpi-4.1.21-kernel.patch
pi@raspberrypi:~$ ls
3 001-adapt-4.1.18-patch-to-rpi-4.1.21-kernel.patch  linux  xenomai-3.0.3
  xenomai-3.0.3.tar.bz2
pi@raspberrypi:~$ cd xenomai-3.0.3/
5 pi@raspberrypi:~/xenomai-3.0.3$ patch -p1 < ../001*
patching file kernel/cobalt/arch/arm/patches/ipipe-core-4.1.18-arm-8.
  patch
7 pi@raspberrypi:~/xenomai-3.0.3$

```

On applique le patch ipipe à Xenomai

```

1 pi@raspberrypi:~/xenomai-3.0.3$ ./scripts/prepare-kernel.sh --arch=arm --
  linux=../linux --ipipe=kernel/cobalt/arch/arm/patches/ipipe-core
  -4.1.18-arm-8.patch

```

On applique un patch correctif pour la gestion des timers du raspberry Pi.

```

1 pi@raspberrypi:~/xenomai-3.0.3$ cd ..
pi@raspberrypi:~$ wget https://www.blaess.fr/christophe/files/article
  -2017-03-20/002-xenomai-3-on-bcm-2709.patch
3 pi@raspberrypi:~$ cd linux
pi@raspberrypi:~/linux$ patch -p1 < ../002*

```

Nous allons bientôt compiler le noyau. On récupère un fichier de configuration fourni par Christophe Blaess. On doit le modifier pour ajouter le support du port série en temps réel.

```

pi@raspberrypi:~/linux$ wget https://www.blaess.fr/christophe/files/
  article-2017-03-20/linux.config
2 pi@raspberrypi:~/linux $ make menuconfig

```

1. Aller dans Xenomai/cobalt

2. Drivers
3. Serial Drivers
4. sélectionner 16550A UART driver et appuyer sur la touche Y
5. Quitter

```
pi@raspberrypi:~/linux$ mv linux.config .config
```

Compiler un noyau linux est long et va beaucoup solliciter le processeur du Raspberry Pi 3. Il est possible d'utiliser les 4 coeurs du Raspberry avec l'option `-j 4`. Cependant cela fera chauffer d'autant plus la raspberry, c'est donc à éviter si le Raspberry se trouve dans un boitier non ventilé sans dissipateur. Dans mon cas le Raspberry était simplement posé sur une table et a tout de même beaucoup chauffé en utilisant les 4 coeurs.

```
1 pi@raspberrypi:~/linux$ time make -j 4
```

On lance la compilation en récupérant le temps d'exécution.

```
1 real 69m29.841s
user 149m13.470s
3 sys 7m4.460s
```

Dans mon cas la compilation a duré plus d'une heure.

On finit d'installer le kernel.

```
1 pi@raspberrypi:~/linux$ sudo cp arch/arm/boot/zImage /boot/kernel-xenomai
img
pi@raspberrypi:~/linux$ sudo make modules_install
3 pi@raspberrypi:~/linux$ sudo make dtbs_install
pi@raspberrypi:~/linux$ ls /boot/dtbs/4.1.21-xenomai+/
5 bcm2709-rpi-2-b.dtb bcm2710-rpi-3-b.dtb overlays
```

On modifie le fichier de configuration du bootloader afin de lui indiquer qu'il faut charger le kernel de Xenomai.

```
1 pi@raspberrypi:~/linux$ sudo nano /boot/config.txt
```

On ajoute ces deux lignes à la fin du fichier.

```
1 kernel=kernel-xenomai.img
device_tree=dtbs/4.1.21-xenomai+/bcm2710-rpi-3-b.dtb
```

On redémarre le Raspberry pour tester notre nouveau noyau.

```
pi@raspberrypi:~/linux$ sudo reboot
```

```

1 pi@raspberrypi:~$ uname -a
Linux raspberrypi 4.1.21-xenomai+ #1 SMP PREEMPT Sat Mar 18 10:12:21 UTC
  2017 armv7l GNU/Linux
3 pi@raspberrypi:~$ cat /proc/ipipe/version
8
5 pi@raspberrypi:~$ cat /proc/xenomai/version
3.0.3
7 pi@raspberrypi:~$

```

On doit compiler les bibliothèques de Xenomai. Pour les préparer on exécute un script. Il est normal qu'il s'exécute pendant quelques minutes sans rien afficher.

```

1 pi@raspberrypi:~$ cd xenomai-3.0.3/
pi@raspberrypi:~/xenomai-3.0.3$ ./scripts/bootstrap

```

On peut enfin compiler la bibliothèque.

```

pi@raspberrypi:~/xenomai-3.0.3$ ./configure --enable-smp
2 pi@raspberrypi:~/xenomai-3.0.3$ make
pi@raspberrypi:~/xenomai-3.0.3$ sudo make install

```

Juste pour s'assurer

```

1 pi@raspberrypi:~$ sudo apt update

```

Ajouter les bibliothèques de Xenomai au PATH:

Pour une raison inconnue les bibliothèques de xenomai ne sont pas toujours trouvées à l'exécution, il faut donc ajouter ces bibliothèques au Path. Pour cela il faut créer un fichier utilisateur en `.conf` dans le dossier `/etc/ld.so.conf.d`

```

pi@raspberrypi:/etc/ld.so.conf.d $ sudo nano usr.conf

```

Dans ce fichier écrire l'adresse de notre librairie Xenomai

```

1 /usr/xenomai/lib

```

On met à jour le cache

```

1 pi@raspberrypi:/etc/ld.so.conf.d $ sudo ldconfig

```

On vérifie que les bibliothèques de xenomai apparaissent

```
1 pi@raspberrypi:/etc/ld.so.conf.d $ sudo ldconfig -p
288 libs found in cache `/etc/ld.so.cache'
3   libz.so.1 (libc6,hard-float) => /lib/arm-linux-gnueabihf/libz.so.1
4   [...]
5   libtrank.so.0 (libc6,hard-float) => /usr/xenomai/lib/libtrank.so.0
6   libtrank.so (libc6,hard-float) => /usr/xenomai/lib/libtrank.so
7   [...]
```

On a bien trouvé des librairies

## References

- [1] Christophe Blaess : *"Xenomai sur Raspberry Pi 3 : bon espoir mais bilan mitigé"*. url : <https://www.blaess.fr/christophe/2017/03/20/xenomai-sur-raspberry-pi-3-bilan-mitige/>
- [2] Télécharger raspbian : [http://downloads.raspberrypi.org/raspbian\\_lite/images/raspbian\\_lite-2017-02-27/](http://downloads.raspberrypi.org/raspbian_lite/images/raspbian_lite-2017-02-27/)
- [3] Autoriser le ssh : <https://www.raspberrypi.org/documentation/remote-access/ssh/>