

ArchPi cheat book

Jérémy Bardon

December 26, 2014

Contents

1	Introduction	3
1.1	Are you interested?	3
1.2	What is a Raspberry Pi	3
1.3	ArchLinux versus Raspbian	4
2	ArchLinux installation	5
2.1	We are NOOBS	5
2.2	Installer usage	6
3	Basic setup	7
3.1	Change language	7
3.1.1	Enable yours	7
3.1.2	Change your settings	8
3.2	Configure wifi connexion	8
3.2.1	Check your dongle	8
3.2.2	Searching the internet	9
3.3	Create yourself	10
3.3.1	Simple user	10
3.3.2	Very important user	10
3.4	Setup remote connexion	11

Preface

Howto book to learn you a few things you need to know about ArchLinux ARM on RPi. From basic setup of the system to side packages installation to turn your Raspberry into a music sharing or even a versioning control server.

Structure of book

The first part of this book will be focused on system setup and basic settings as keyboard language, user account and others. The second part will describe how to install some third party softwares as git and mpd server.

Author words

I am not an expert in linux system as ArchLinux and even less in electronic stuff. However, as a developer I like to tinker with my new toy which is a Raspberry Pi.

I had a lot of troubles when I decided to find uses for it and tried to install some third party software. As a result, I am glad to write this “book” to help you to install things on your RPi with ArchLinux.

Chapter 1

Introduction

1.1 Are you interested?

This book is written by a non-specialist of ArchLinux with basic knowledge of linux system so I will try to made it as simple as possible for people who have no idea about what is console. Indeed, all commands will be explained for a better comprehension and an index will be available for you.

No matter if you are an expert or a novice, you will be able to find how to install stuff on your your Pi plus tips which includes all the problems I encounter during my first installation.

1.2 What is a Raspberry Pi

If you succeed to find this book I guess you allready know but some people buy a Raspberry with OpenELEC¹ pre-installed so here is a little explanation.

The Raspberry Pi is a credit-size computer with low performances if you compare with a common PC. Nevertheless, it means its power consumption is very low (1W for B+ version²) so it is not a problem to let it on forever.

¹Tiny linux system based on XBMC media center. More details on openelec.tv

²Most robust version of RPi with 512MB of RAM and 4 usb ports

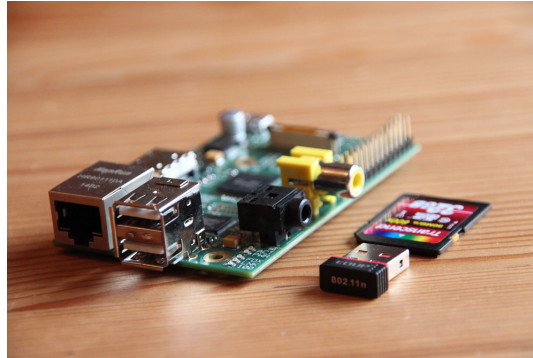


Figure 1.1: Raspberry Pi B+

Finally if you install a good linux distribution on it you can turn this old computer into a cheap server on which you will have the control. You can use it at home for file sharing, media player or others but it is also possible to host a website which will be available on the internet³.

1.3 ArchLinux versus Raspbian

The operating system recommended by the Raspberry fundation is Raspbian – a custom version of the famous Debian⁴ system – optimized for RPi hardware.

In general it will be the default choice for an inexperienced user to get a user interface and most common softwares allready installed at the first boot. However we forget the limited performances of the Raspberry and you will be able to realize that for yourself if you decided to install Raspbian.

A server does not need a user interface except a terminal which is enough to manage it everyday from anywhere. As a result, my choice has been focused on ArchLinux which is a pretty light and fast system. In addition, system updates are based on rolling release⁵ model, so it means you do not have one version of the system. You will just receive updates frequently – as soon as their availability – and it will be not necessary to reboot during the upgrade process.

³An example of website hosted by a Rpi on raspberrypi.goddess-gate.com

⁴One of the most popular linux system. See debian.org for more details

⁵Definition on wikipedia/Rolling_release

Chapter 2

ArchLinux installation

2.1 We are Noobs

There are two ways to install ArchLinux on a Raspberry Pi: the first is the ArchLinux way – no idea if it is the same with other systems – and the second is an official manager which works with many systems.

- follow instructions from archlinuxarm.org¹ which requires to already have a linux system
- use NOOBS, an operating system install manager provided by the Raspberry foundation²

I choose NOOBS because it is the easiest way to install a system on a RPi and in addition you get an extra “boot manager” which is usefull. Moreover, you can complete the full setup of your SD card on any system in few simple steps described in the next part.

NOOBS is available on two forms: one for offline installation and the other – the smallest one – downloads automatically the last release of the system online. The offline installer contains many systems – which takes a large space – but you can just keep ArchLinux and remove others (in `os` folder). Anyway, you need to know that other systems files will be kept after the installation so it is lost space. If you still want to use the offline way because you have no choices, you will have to find an older version of NOOBS because ArchLinux has been removed since the last release.

¹Specific instruction on archlinuxarm.org/platforms/armv6/raspberry-pi

²Details on www.raspberrypi.org/help/noobs-setup

2.2 Installer usage

According to the NOOBS documentation, you just have to download the last NOOBS release and format your SD card before unzip and copy NOOBS files on the card.

After putting the card into the RPi and power on, you will get the installer interface with a list of all systems you can install. If you choose the online way you have to connect your ethernet cable in the Raspberry even if you want to use a wifi dongle later.

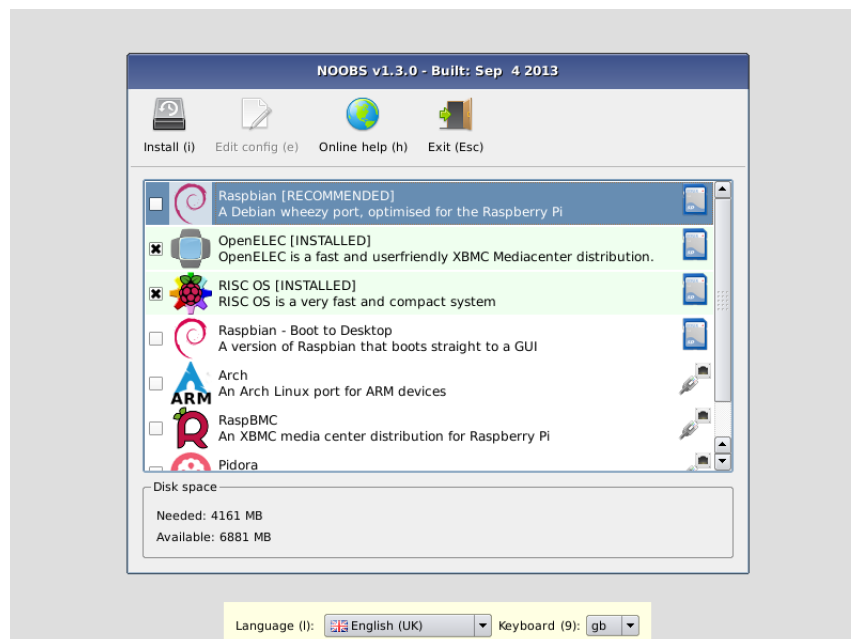


Figure 2.1: Noobs installer menu

From the menu, select ArchLinux – with and – and press to valid your choice. Then, you can change menu and keyboard language with respectively and before pressing to begin ArchLinux installation on your SD card.

Chapter 3

Basic setup

3.1 Change language

There are no default languages selected in ArchLinux but the keyboard map is set to qwerty¹ at the beginning. To choose your language you have to complete two steps and then the system will be able to use it for characters encoding and some softwares as **nano**.

3.1.1 Enable yours

Before choosing yours it is necessary to enable it in `/etc/locale.gen` file with `locale` tools.

You will need to use **nano** to edit the configuration file – `ctrl` + `W` can help you to search – and remove “#” before the language you want to enable (`fr_FR.UTF-8` for example), to save your changes press `ctrl` + `X`.

```
$ locale # Current language settings

# Edit the /etc/locale.gen file
$ nano /etc/locale.gen

$ locale-gen # Update available languages
$ locale -a # See available languages
```

Listing 3.1: Enable your language

¹Most common layout for keyboards

3.1.2 Change your settings

The second step is to set the language and configure your keyboard map. Notice that you will have to logout for the system to take into account changes you made in language setup.

```
$ localectl status
  System Locale: n/a  # System language
    VC Keymap: n/a  # Virtual console
    X11 Layout: n/a  # Graphic interface

# Change system language (choose enabled one)
$ localectl set--locale LANG=fr_FR.UTF-8

# List of keymaps, choose the one you want "fr-pc" for example
$ localectl list--keymaps

# Change settings
# no-convert not update VC with X11 and vice versa
$ localectl set--keymap --no-convert fr-pc      # VC Keymap
$ localectl set--x11-keymap --no-convert fr-pc  # X11 Layout

# Logout to apply changes
$ exit
```

Listing 3.2: Change language settings

3.2 Configure wifi connexion

3.2.1 Check your dongle

The first thing you can do is checking if your dongle has been recognized by the system and can be used.

```
$ ifconfig -a wlan # All wireless interfaces (also disabled)
```

Listing 3.3: Check wifi device

There are a lot of ways to connect your RPi to a network using a wifi dongle but all of them requires to install a package before – wifi-menu needs dialog, iw and wpa_supplicant are not installed – so it is necessary to use an ethernet wire to install them.

```
# pacman is the package manager in ArchLinux
#      -S install a new package
#
# dialog to get wifi-menu interface
# wpa_supplicant for wireless network protected with wpa keys
#
$ pacman -S dialog wpa_supplicant
```

Listing 3.4: Install wireless dependencies

3.2.2 Searching the internet

Once you installed all the packages – synonym for software – that wifi-menu needed to run you can launch it with “wifi-menu” command.

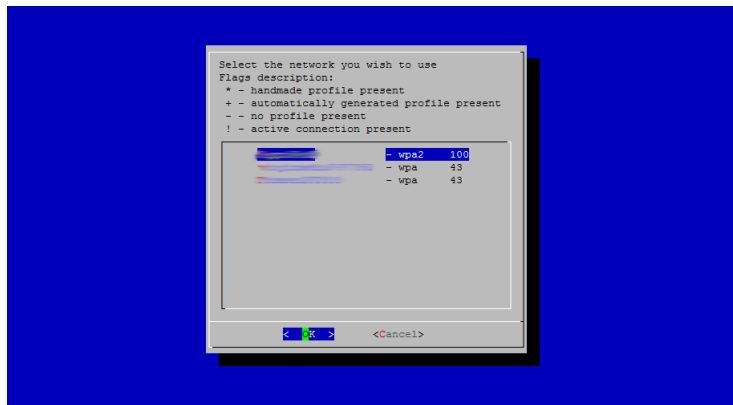


Figure 3.1: wifi-menu interface

Then, select your network – with and – , type your password (if required) and you will be connected to your wireless network.

3.3 Create yourself

3.3.1 Simple user

The default username and password for ArchLinux are **root/root**, this user got all right on the system it means he can do anything – even break the system – so it is not recommended to use it.

However, you should change this default password for security purposes and only use your new account.

```
$ passwd root          # Changes root password (for security)
$ useradd -m jeremy    # Creates user jeremy and his home folder
$ passwd jeremy        # Changes jeremy password
```

Listing 3.5: Create a new user called jeremy

Now we have a new user account for everyday usage. You can see all users on your system with “**cat /etc/passwd**” which will display the content of the config file for users.

3.3.2 Very important user

It is possible to specify user rights with **visudo** command, the general syntax for one user is the following “**username machine=(targetuser) commands**”, let’s look at some details:

username name you gave to **useradd** command

machine machine on where rights are applied, ALL in general

target user user that we takes the rights

command allowed commands separated with one coma – no spaces –, use exclamation mark for banned commands

visudo is the command which prevent you from blocking the system with bad changes in the config file **/etc/sudoers**. The default text editor used by **visudo** is **vim** but before we used **nano** so to keep using it you have tell it to the system.

```
$ pacman -S sudo    # visudo is inside

# For this session set nano as default editor
# EDITOR is an environment variable
$ export EDITOR=/usr/bin/nano

# To check your changes, use echo which print a message
# in the console and variable with "$" before
$ echo $EDITOR

# Add your account rights after root
# for example "jeremy ALL=(ALL) ALL"
$ visudo
```

Listing 3.6: Specify user rights

Now you are a regular user with root permissions but some commands requires to be root – as making changes in systems files – so if a command want not works you can try to add **sudo** before.

By doing this the system will ask you for your password – even if you are logged – to get root permissions temporarily.

```
# Logged as jeremy (regular user)
$ visudo
visudo: /etc/sudoers: Permission denied

$ sudo visudo    # Launches visudo with root rights
```

Listing 3.7: sudo command usage

This example shows you that **visudo** command requires to be root so if you are logged with a regular account you can not use it. Instead of logout and login with root user – remember do not use this account – just add **sudo** before your command.

3.4 Setup remote connexion

If you want to keep working with your regular computer on the RPi – without two screens and two keyboards – it possible to setup a remote connexion with **ssh**².

On the system there are many programs as **ssh** or web server launched

²Remote secure shell access

in background. These programs are regularly started at system startup and are called *daemons* or *services*.

Before trying to connect to your Raspberry you should check if `sshd` – for ssh daemon – is running.

```
$ systemctl status sshd  # Check service state

# If sshd is not running, launch it
$ systemctl start sshd   # Start service

# If after a reboot it is not started
# It can be disabled
$ systemctl enable sshd  # Enable service
```

Listing 3.8: Check service state

Now that `sshd` is running, you will be able to connect to it remotely from any other system. Here is a list of the methods you should use depending on your own machine:

Linux and OSX Use the build-in terminal

Windows PuTTY³ is the most used tool for this kind of work

Whatever the system you are using, two informations are necessary to perform an ssh connexion : remote account – created in section 3.3.1 – and remote machine IP or domain name.

```
# Format: username@IP (or domain)
$ ssh jeremy@192.168.0.2          # Initiate ssh connexion
jeremy@192.168.0.23's password:  # Type your account password

$ exit  # Stop ssh connexion and go back to local terminal
```

Listing 3.9: SSH with command line

³SSH client available on putty.org

Chapter 4

Web server