Multimedia

Contents

[2 The Raspberry Pi 3](#_Toc398122284)

[3 The Operating System 4](#_Toc398122285)

[3.1 Installing Raspbian 4](#_Toc398122286)

[3.2 Booting the Raspberry Pi 5](#_Toc398122287)

[3.2.1 Graphical Interface 5](#_Toc398122288)

[3.2.2 SSH connection 5](#_Toc398122289)

[3.2.3 RS232 Connection 7](#_Toc398122290)

[3.3 Initial Configuration 10](#_Toc398122291)

[3.3.1 Expanding the Filesystem 10](#_Toc398122292)

[3.3.2 Change User Password 10](#_Toc398122293)

[3.3.3 Enable Boot to Desktop / Scratch 10](#_Toc398122294)

[3.3.4 Internationalization Options 10](#_Toc398122295)

[3.3.5 Enable Camera 11](#_Toc398122296)

[3.3.6 Add to RasTrack 11](#_Toc398122297)

[3.3.7 Overclock 11](#_Toc398122298)

[3.3.8 Advanced Options 11](#_Toc398122299)

[3.4 Checking the configuration 12](#_Toc398122300)

[4 Buildroot: Making Embedded Linux easy 14](#_Toc398122301)

[4.1 Introduction 14](#_Toc398122302)

[4.2 Getting Buildroot 14](#_Toc398122303)

[4.3 Requirements 14](#_Toc398122304)

[5 Important Command Overview 15](#_Toc398122305)

[6 References 15](#_Toc398122306)

# The Raspberry Pi

The Raspberry Pi is a credit card-sized single-board computer developed in the UK by the Raspberry Pi Foundation with the intention of promoting the teaching of basic computer science in schools.

The Raspberry Pi has a Broadcom BCM2835 system on a chip (SoC), which includes an ARM1176JZF-S 700 MHz processor, VideoCore IV GPU, and was originally shipped with 256 megabytes of RAM, later upgraded (Model B & Model B+) to 512 MB. It does not include a built-in hard disk or solid-state drive, but it uses an SD card for booting and persistent storage, with the Model B+ using a MicroSD.

# The Operating System

The Raspberry Pi foundation provides several ready to use operating system images for the Pi. At the moment this writing the following are available:

* RASPBIAN - Debian Wheezy
* PIDORA - Fedora Remix
* OPENELEC - An XBMC Media Centre
* RASPBMC - An XBMC Media Centre
* RISC OS - A non-Linux distribution
* ARCH LINUX - A lightweight Linux distribution

For the start of this course we will be using the Raspbian image. Later we will create our own distribution.

## Installing Raspbian

You can download the latest image from the Raspberry Pi website (<http://www.raspberrypi.org/downloads/>). Make sure to pick the “Download ZIP” option. Extract the zip file on your local disk. You should get an image file (.img extension).

The current version at the moment of this writing is of June 2014 with a Linux kernel version of 3.12. You can always check out the release notes on <http://downloads.raspberrypi.org/raspbian/release_notes.txt>.

To boot this Linux distribution we will need to write the image file to an SD card of at least 4GB. A popular tool to write the image to an SD card is “Win32 Disk Imager”[[1]](#footnote-1) which can be downloaded on <http://sourceforge.net/projects/win32diskimager/>

Select the correct device letter and load the Raspbian image from your local drive as shown in Figure 1. If you’re ready hit the write button and grab a cup of coffee. You can also create a backup of your current SD card by reading from the SD card to an image file. Just make sure to select a new image file name.

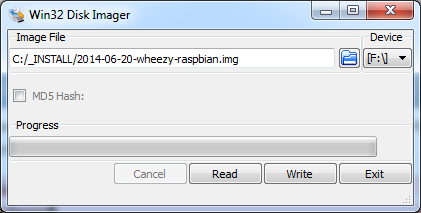


Figure 1 Win32 Disk Imager

Once the write process is finished you can remove the SD card and plug it in the Raspberry Pi. Just make sure to disconnect the power first.

If you want your Pi to be connected to your local area network (LAN) you will have to plug in the Ethernet cable before booting the Pi. The Pi is default configured to acquire an IP address using DHCP.

## Booting the Raspberry Pi

Booting the Raspberry Pi is really simple. All you have to do is plug in the supply adapter and it automatically boots from the SD card. Interacting with the Linux operating system from that point on can be a bit harder in certain situations.

### Graphical Interface

If you deployed an OS such as Raspbian than you can attach an HDMI display or RCA Video compatible device (yellow connector on the board, cfr. Figure xxxxxxxxx). You will also have to connect a USB keyboard to the Raspberry Pi to be able to navigate through the configuration menu you will be presented with on the initial boot.

Jump to Section 2.3 to configure the Pi for initial use using the configuration menu.

### SSH connection

Raspbian comes default with the SSH[[2]](#footnote-2) daemon enabled. This allows us to connect to the Pi from a computer using the SSH protocol. Before we can do this we will have to determine the IP address of the Pi. In case of a home network you can log on to your router and look for the last IP address that was given by your DHCP server running on the router.

Another option can be a network scan tool such as SoftPerfect Network Scanner[[3]](#footnote-3) which allows you to scan a range of IP addresses and display some basic information about them such as the MAC (Media Access Control) address and the hostname.

This would not be an option in a LAB if there are 12 Pi’s connected to the same subnet all with the default configuration of Raspbian. However for your convenience we added labels on the Pi’s with their respective MAC addresses so you can identify which Pi is yours.

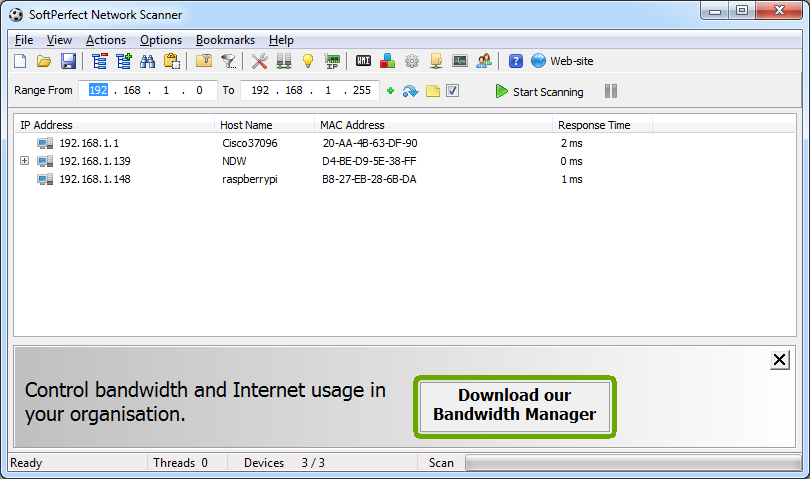


Figure 2 Network scan using SoftPerfect Network Scanner

Another option is using WireShark[[4]](#footnote-4) and watch the communication on the network. Especially the DHCP traffic which distributes IP addresses to the connected client devices. This way you can also identify what IP address is given to your device (if you know the MAC address of your device).

TODO

Connecting to a device using the SSH protocol can be easily achieved using a terminal tool such as Putty[[5]](#footnote-5). All you have to do is start Putty and select the SSH connection option and specify the IP address of the device as shown in Figure 2. Once the connection is configured you can open it.

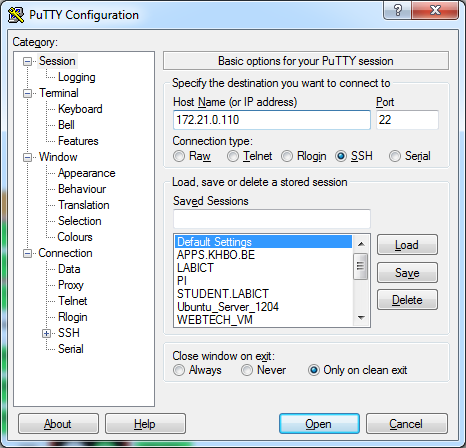


Figure 3 Opening an SSH connection using Putty

You will be presented with the command line interface (CLI) of the Linux operating system running on your device. The first thing you will see is a login screen similar to the one shown in Figure 3.



Figure 4 The login screen of the Raspbian distribution running on the Pi

The default username and password can be found on the Raspberry Pi website. For Raspbian it is “pi” as username and “raspberry” as password. Once you login with these credentials you are presented with the command line interface as shown in Figure 5. From this point on you can start to execute commands on the Pi.

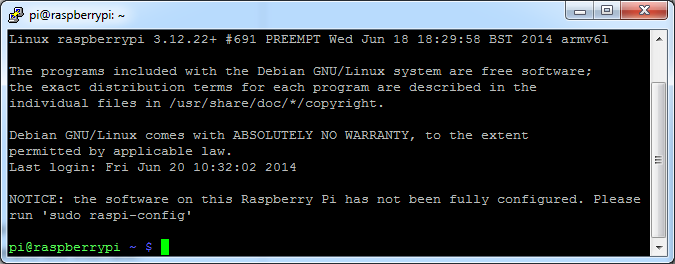


Figure 5 The command line interface after logging in

One of the most useful commands you should remember is the “ifconfig” command which displays the current network interfaces and their configuration parameters. If you execute the command you should get a similar output to the one shown in Figure 6. Try to identify the IP address and MAC address of the primary Ethernet interface (eth0).

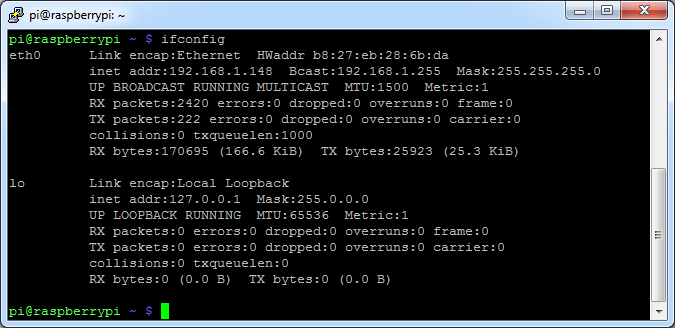


Figure 6 Output of the ifconfig command

Jump to Section 2.3 to configure the Pi for initial use using the configuration menu.

### RS232 Connection

A last option we can use to connect to the Raspberry Pi is by using a serial connection. This is often used for debugging embedded systems because it is a very basic connection type. Because of this the kernel will also output its kernel messages (debugging information and errors) to this connection. Since most computers these days lack the serial interface we can use a simple RS232 to USB converter such as the PL-2303HX[[6]](#footnote-6).

To attach the converter we do have to take a look at the pinout of the GPIO connector on the Raspberry Pi board, shown in Figure 7.

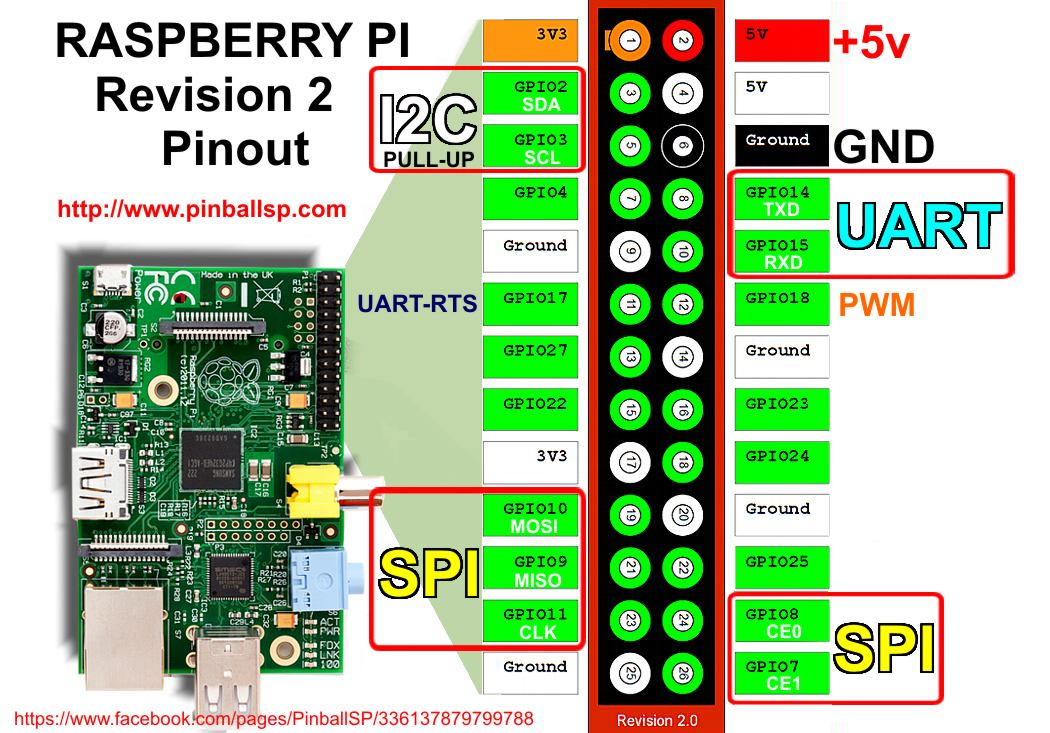


Figure 7 Raspberry Pi GPIO pinout

The serial-to-USB converter has three pins that need to be connected to the UART of the Pi. A Tx (transmit), an Rx (receive) and a GND (ground) pin. To allow communication with the Raspberry Pi the Tx of the Pi has to be connected with the Rx of the converter, while the Rx of the Pi has to be connected with the Tx of the converter. The GND pin of the converter needs to be connected to a GND pin of the Pi. Do NOT connect the 3V3 or 5V pin of the connector to the Pi! The necessary connections are shown in Figure 8.

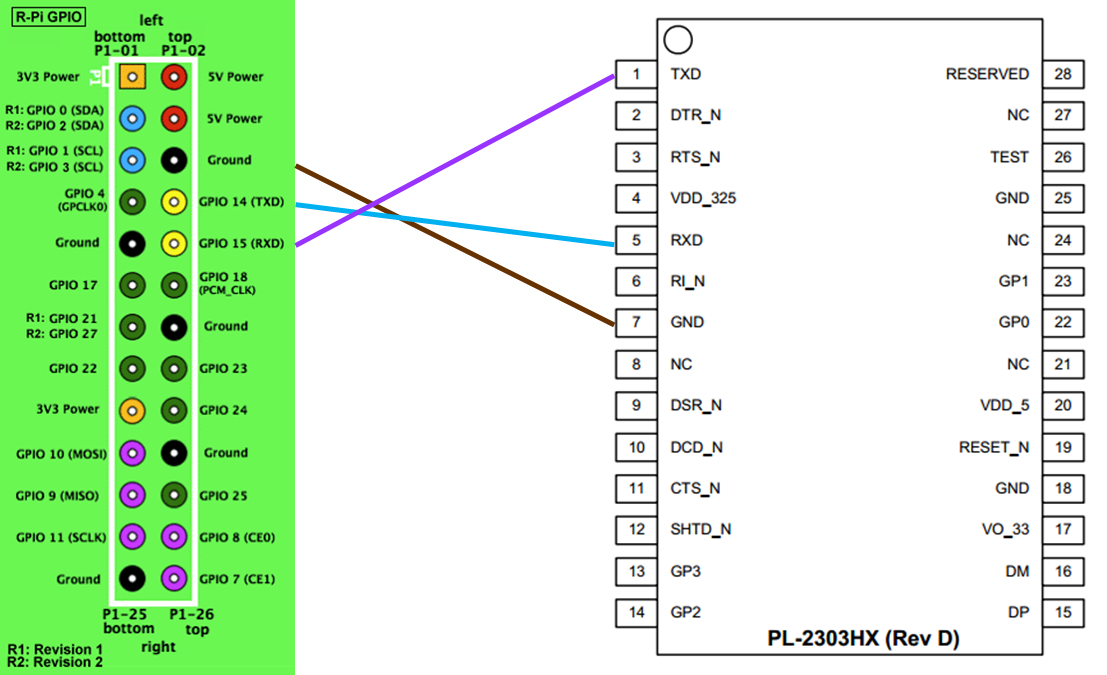


Figure 8 Connecting the PL2303HX to the UART of the Raspberry Pi

Just as with SSH, you can use Putty for the serial terminal. Just select “serial” as connection type, “COMx”[[7]](#footnote-7) (where x is an integer number) as serial line and “115200” as speed. An example is shown in Figure 9. Choose open and you will a get a command line interface similar to the one of SSH.

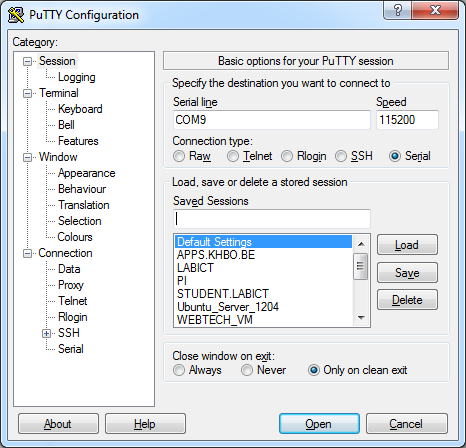


Figure 9 Serial line connection parameters

If you reboot your Raspberry Pi at this moment you will see the kernel messages shown in Figure 10 mentioned earlier.

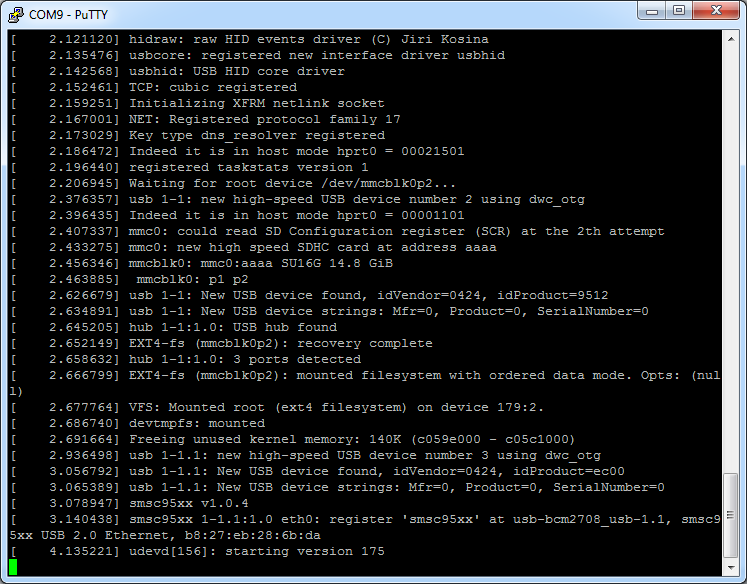


Figure 10 Kernel messages on serial line

## Initial Configuration

If you choose the to attach an external display to the Pi, then will already have seen the blue screen menu with the initial configuration options shown in Figure 11.

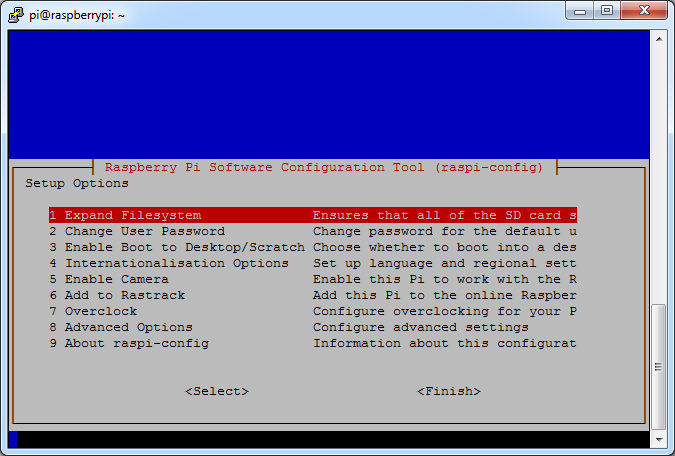


Figure 11 Initial configuration tool

If you choose to use the SSH or serial method you will need to run the command

sudo raspi-config

to get this configuration menu. Don’t worry too much about how this works. This will be explained in a later chapter.

Once you’re done configuring the Pi, choose the Finish option and let the Pi reboot.

### Expanding the Filesystem

The first thing we need to do is expand the filesystem. Currently we are using SD cards of 8GB or 16GB but the root file system only takes up about 3GB of which over 80% is used. So to expand the root filesystem to the full SD card we can use the “Expand Filesystem” configuration script. You will need to reboot the Raspberry Pi to make this available.

### Change User Password

The “Change User Password” tool allows you to change the default password of the pi user. Make sure to do this before continuing. Watch out if you do this using an external keyboard as the keyboard layout may be configured to qwerty. In this case configure the keyboard layout first by selecting the internationalization menu option.

### Enable Boot to Desktop / Scratch

You can change what happens when your Pi boots. Use this option to change your boot preference to command line, desktop, or straight to Scratch.

### Internationalization Options

This will open up a sub menu with internationalization options to configure.

#### Change Locale

Locales are a framework to switch between multiple languages and allow users to use their language, country, characters, collation order, etc.

In the first screen you will be asked which locales to generate. UTF-8 locales should be chosen by default, particularly for new installations. Other character sets may be useful for backwards compatibility with older systems and software.

Select both “en\_GB.UTF-8 UTF-8” and “nl\_BE.UTF-8 UTF-8”. On the next screen you can pick any of these two to be the default locale.

#### Change Timezone

The time zone should be changed to “Europe – Brussels” to reflect our own time zone.

#### Change Keyboard Layout

This option opens another menu which allows you to select your keyboard layout. It will take a long time to display while it reads all the keyboard types. Changes usually take effect immediately, but may require a reboot.

This option however does not seem to work with the current version of the configuration tool.

You can however change the keyboard layout to a typical Belgian azerty layout by executing the following command the next time you get to the command line interface:

$ sudo setxkbmap be

### Enable Camera

In order to use the Raspberry Pi camera module, you must enable it here. This option will also make sure at least 128MB of RAM is dedicated to the GPU.

We will come back to this later.

### Add to RasTrack

Rastrack is a user-contributed Google Map to which Pi users in the community have added their location; it shows a heat map of where Pi users are known to be around the world. This was set up by young Pi enthusiast Ryan Walmsley in 2012. Rastrack is located at rastrack.co.uk.

Skip this option for the LAB.

### Overclock

It is possible to overclock your Raspberry Pi's CPU. The default is 700MHz but it can be set up to 1000MHz. The overclocking you can achieve will vary; overclocking too high may result in instability.

Selecting this option shows the following warning:

Be aware that overclocking may reduce the lifetime of your Raspberry Pi. If overclocking at a certain level causes system instability, try a more modest overclock. Hold down `shift` during boot to temporarily disable overclock.

### Advanced Options

The advanced options allow the configuration of the SSH daemon, the hostname, the division of the memory with the GPU and so on.

#### Overscan

Old TV sets had a significant variation in the size of the picture they produced; some had cabinets that overlapped the screen. TV pictures were therefore given a black border so that none of the picture was lost; this is called overscan. Modern TVs and monitors don't need the border, and the signal doesn't allow for it. If the initial text shown on the screen disappears off the edge, you need to enable overscan to bring the border back.

On some displays, particularly monitors, disabling overscan will make the picture fill the whole screen and correct the resolution. For other displays, it may be necessary to leave overscan enabled and adjust its values.

Any changes will take effect after a reboot.

#### Hostname

Set the visible name for this Pi on a network. Pick a unique name, different from the other students.

#### Memory Split

Change the amount of memory made available to the GPU (Graphics Processing Unit). For the initial LAB exercises you can change this to 16MB.

#### SSH

Enable/disable remote command line access to your Pi using SSH.

SSH allows you to remotely access the command line of the Raspberry Pi from another computer. Disabling this ensures the SSH service does not start on boot, freeing up processing resources. Note that SSH is enabled by default. If connecting your Pi directly to a public network, you should disable SSH unless you have set up secure passwords for all users.

#### SPI (Serial Peripheral Interface)

Enable/disable automatic loading of SPI kernel module, needed for products such as PiFace.

More on this later.

#### Audio

Force audio out through HDMI or a 3.5mm jack.

#### Update

Update this tool to the latest version. This requires an active internet connection.

## Checking the configuration

At this point you should be able to login to the Pi using the pi user with the new password. If you execute the disk free command you should get a similar output to the one shown in Figure 12.

$ df –h

df stands for disk free and the –h option requests a human readable output formatting.

Notice how the disk size has increased and the used disk space percentage has dropped significantly.

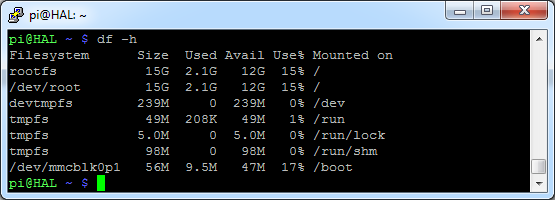


Figure 12 Output of the disk free command after initial configuration

Also notice how the hostname of the device changed (HAL[[8]](#footnote-8) in this case).

# Buildroot: Making Embedded Linux easy

## Introduction

Buildroot is a tool that simplifies and automates the process of building a complete Linux system for an embedded system, using cross-compilation.

In order to achieve this, Buildroot is able to generate a cross-compilation toolchain, a root filesystem, a Linux kernel image and a bootloader for your target. Buildroot can be used for any combination of these options, independently (you can for example use an existing cross-compilation toolchain, and build only your root filesystem with Buildroot).

Buildroot is useful mainly for people working with embedded systems. Embedded systems often use processors that are not the regular x86 processors everyone is used to having in his PC. They can be PowerPC processors, MIPS processors, ARM processors, etc.

Buildroot supports numerous processors and their variants; it also comes with default configurations for several boards available off-the-shelf. Besides this, a number of third-party projects are based on, or develop their BSP 1 or SDK 2 on top of Buildroot.

Buildroot is designed to run on Linux systems.

## Getting Buildroot

## Requirements

While Buildroot itself will build most host packages it needs for the compilation process, certain standard Linux utilities are expected to be already installed on the host system. Below you will find an overview of the mandatory packages and how to install them on a Debian based Linux system.

# Important Command Overview

|  |  |
| --- | --- |
| Command | Description |
| ifconfig | Configure a network interface |

# References

|  |  |
| --- | --- |
| Description | URL |
| Configuration options of the P | http://www.raspberrypi.org/documentation/configuration/config-txt.md |

1. Check out <http://www.raspberrypi.org/documentation/installation/installing-images/README.md> for instructions for different host operating systems such as Linux or Mac. [↑](#footnote-ref-1)
2. SSH or Secure Shell is a secure way to connect to a device and execute commands from a distance. Default SSH daemon listen on port 22. See chapter xxxx for more information on SSH. [↑](#footnote-ref-2)
3. <http://www.softperfect.com/products/networkscanner/> [↑](#footnote-ref-3)
4. Wireshark, originally named Ethereal, is a free and open-source packet analyzer. It is used for network troubleshooting, analysis, software and communications protocol development, and education. It can be downloaded from <https://www.wireshark.org/> [↑](#footnote-ref-4)
5. PuTTY is a free implementation of Telnet and SSH for Windows and Unix platforms, along with an xterm terminal emulator. It can be downloaded from <http://www.chiark.greenend.org.uk/~sgtatham/putty/> [↑](#footnote-ref-5)
6. The datasheet can be downloaded from <https://www.adafruit.com/datasheets/PL2303HX.pdf> [↑](#footnote-ref-6)
7. You can find the COM port number using the device manager. Select the “Ports (COM & LPT)” category and look for a “USB-to-Serial Comm Port (COMx)” device. [↑](#footnote-ref-7)
8. <http://en.wikipedia.org/wiki/HAL_9000> [↑](#footnote-ref-8)