|  |
| --- |
|  |
| Multimedia  Multimedia |
| |  |  |  | | --- | --- | --- | | Nico De Witte | 9/10/14 | [Course title] | |

Contents

[2 The Raspberry Pi 2](#_Toc399321051)

[2.1 What’s the Raspberry Pi Made off 3](#_Toc399321052)

[2.1.1 Specifications 5](#_Toc399321053)

[3 An Operating System 6](#_Toc399321054)

[3.1 Installing Raspbian 6](#_Toc399321055)

[3.2 Booting the Raspberry Pi 7](#_Toc399321056)

[3.2.1 Graphical Interface 7](#_Toc399321057)

[3.2.2 SSH connection 7](#_Toc399321058)

[3.2.3 RS232 Connection 9](#_Toc399321059)

[3.2.4 Checking The Kernel Messages 11](#_Toc399321060)

[3.3 Initial Configuration 12](#_Toc399321061)

[3.3.1 Expanding the Filesystem 12](#_Toc399321062)

[3.3.2 Change User Password 12](#_Toc399321063)

[3.3.3 Enable Boot to Desktop / Scratch 12](#_Toc399321064)

[3.3.4 Internationalization Options 12](#_Toc399321065)

[3.3.5 Enable Camera 13](#_Toc399321066)

[3.3.6 Add to RasTrack 13](#_Toc399321067)

[3.3.7 Overclock 13](#_Toc399321068)

[3.3.8 Advanced Options 13](#_Toc399321069)

[3.4 Checking the configuration 15](#_Toc399321070)

[4 Linux Basics 16](#_Toc399321071)

[4.1 The MAN-Pages 16](#_Toc399321072)

[4.2 Traversing and Manipulating the File System 16](#_Toc399321073)

[4.3 Installing Packages 18](#_Toc399321074)

[5 Running a Virtual Machine 19](#_Toc399321075)

[5.1 Installing Oracle VirtualBox 19](#_Toc399321076)

[6 Buildroot: Making Embedded Linux easy 21](#_Toc399321077)

[6.1 Introduction 21](#_Toc399321078)

[6.2 Getting Buildroot 21](#_Toc399321079)

[6.3 Requirements 21](#_Toc399321080)

[7 Important Commands Overview 22](#_Toc399321081)

[8 References 22](#_Toc399321082)

# 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[[1]](#footnote-1) system on a chip (SoC[[2]](#footnote-2)), which includes an ARM1176JZF-S 700 MHz processor, a 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.

ARM is a family of instruction set architectures for computer processors based on a reduced instruction set computing (RISC[[3]](#footnote-3)) architecture developed by British company ARM Holdings.

The Broadcom SoC used in the Raspberry Pi is equivalent to a chip used in an old smartphone (Android or iPhone). While operating at 700 MHz by default, the Raspberry Pi provides a real world performance roughly equivalent to the 0.041 GFLOPS. On the CPU level the performance is similar to a 300 MHz Pentium II of 1997-1999, but the GPU, however, provides 1 Gpixel/s, 1.5 Gtexel/s or 24 GFLOPS of general purpose compute and the graphics capabilities of the Raspberry Pi are roughly equivalent to the level of performance of the Xbox of 2001. The Raspberry Pi chip operating at 700 MHz by default, will not become hot enough to need a heatsink or special cooling.

The Raspberry Pi became popular very fast as the system was cheap but powerful. Some projects include:

* **Use a Raspberry Pi to Automate Time-Lapse Photos** (<https://sites.google.com/site/raspilapse/home>)



* **Raspberry Pi Supercomputer** (<http://www.southampton.ac.uk/~sjc/raspberrypi/>)



* **Automatic DeviantArt Picture Frame** (<http://theswitchtolinux.blogspot.be/2012/12/raspberry-pi-daily-deviations-picture.html>)



And the list keeps growing every day.

## What’s the Raspberry Pi Made off

The Raspberry has everything onboard you need for a small media center, simple NAS or home automation system. Figure 1 shows the placement of the different components on the Raspberry Pi model B. While the Pi is at its third major board revision, we will be focusing on model B in this course.

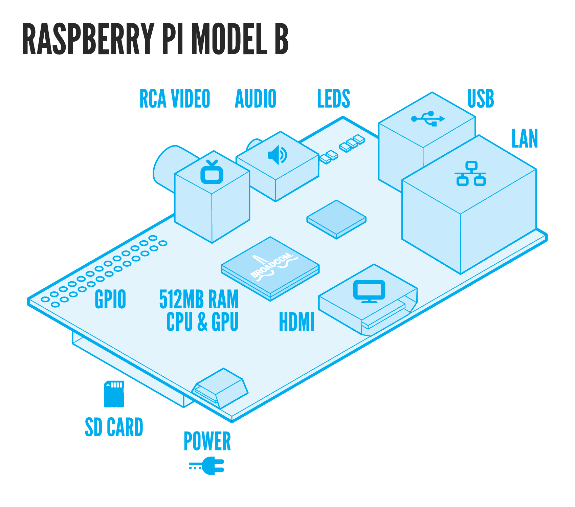
 

Figure Raspberry Pi model B - Component placements (left) – Real Board (right)

The Pi model B offers:

* Full size SD card interface
* HDMI output port
* Composite video output
* 26 pin expansion header exposing GPIO, I2C, etc
* 3.5mm audio jack
* Camera interface port (CSI-2)
* LCD display interface port (DSI)
* One microUSB power connector for powering the device
* One Ethernet port
* Two USB ports

Check out Section 2.1.1 for more detailed information on the Pi’s onboard hardware.

If you're wondering why you can't see the Broadcom SoC anywhere on the Raspberry Pi, it's because you're looking in the wrong place. The processor is hidden beneath the Hynix memory chip at the centre of the board, which uses a package-on-package (PoP) (example shown in Figure 2) mounting process to create a RAM-SoC sandwich.

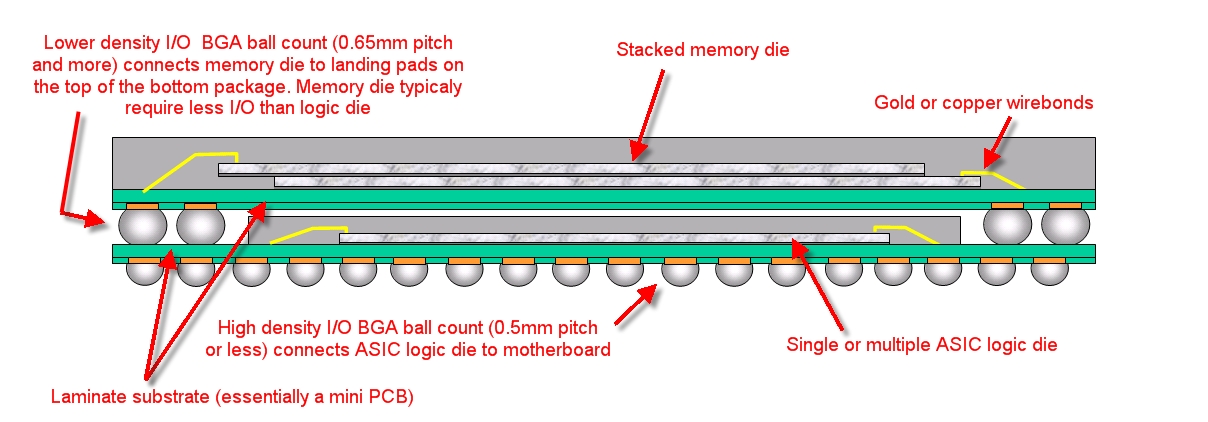


Figure Package-on-Package

Important to know is that the Ethernet controller (shown in Figure 2) on the Pi has a build-in USB hub. The upstream port of this hub is connected to the Broadcom processor while the two downstream ports are made available using a dual-stacked USB port.

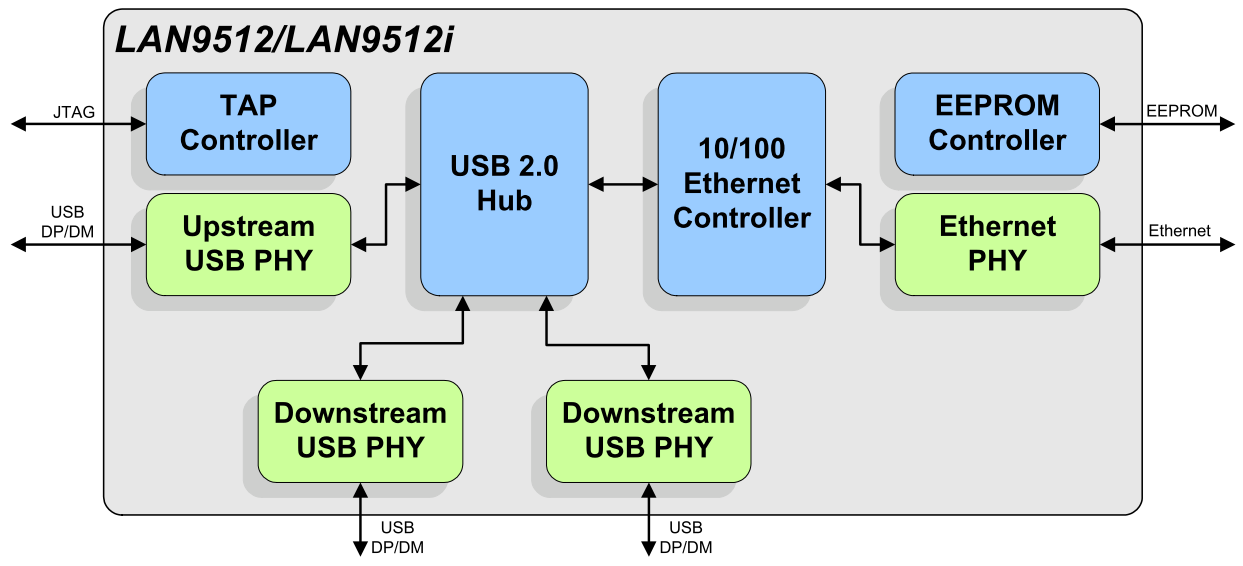


Figure The Pi's Ethernet Controller with Integrated USB HUB

Detailed schematics of the Pi can be found at <http://www.raspberrypi.org/documentation/>.

### Specifications

Table 1 shows an overview of the differences between the different major board revisions of the Pi.

Table RPi Major Board Revisions

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Model A** | **Model B** | **Model B+** |
| **Target price** | US$25 Ext tax | US$35 Ext tax | US$35 Ext tax |
| **System-on-a-chip (SoC)** | Broadcom BCM2835 (CPU + GPU. SDRAM is a separate chip stacked on top) | | |
| **CPU** | 700 MHz ARM11 ARM1176JZF-S core | | |
| **GPU** | Broadcom VideoCore IV, OpenGL ES 2.0, OpenVG 1080p30 H.264 high-profile encode/decode | | |
| **Memory (SDRAM)iB** | 256 MiB (planned with 128 MiB, upgraded to 256 MiB on 29 Feb 2012) | 256 MiB (until 15 Oct 2012)  512 MiB (since 15 Oct 2012) | 512 MiB |
| **USB 2.0 ports** | 1 (provided by the BCM2835) | 2 (via integrated USB hub) | 4 (via integrated USB hub) |
| **Video outputs** | Composite video | Composite RCA, HDMI (not at the same time) | | Composite video requires 4 Pole Adapter |
| **Audio outputs** | TRS connector | 3.5 mm jack, HDMI | | |
| **Audio inputs** | None, but a USB mic or sound-card could be added | |  |
| **Onboard Storage** | Secure Digital|SD / MMC / SDIO card slot | | Micro Secure Digital / MicroSD slot |
| **Onboard Network** | None | 10/100 wired Ethernet RJ45 | |
| **Low-level peripherals** | 26 General Purpose Input/Output (GPIO) pins, Serial Peripheral Interface Bus (SPI), I²C, I²S, Universal asynchronous receiver/transmitter (UART) | | 40 General Purpose Input/Output (GPIO) pins, Serial Peripheral Interface Bus (SPI), I²C, I²S, I2C IDC Pins, Universal asynchronous receiver/transmitter (UART) |
| **Real-time clock** | None | | |
| **Power ratings** | 300 mA, (1.5 W) | 700 mA, (3.5 W) | ~650 mA, (3.0 W) |
| **Power source** | 5 V (DC) via Micro USB type B or GPIO header | | |
| **Size** | 85.0 x 56.0 mm x 15mm | 85.0 x 56.0 mm x 17mm | 85.0 x 56.0 mm x 17mm |
| **Weight** | 31g | 40g | 40g |

While the specs recommend a 700mA power supply, it’s best to use a 1000mA power supply when connecting the Pi to an Ethernet network. Also make sure not to connect external HDD drives, which are powered by USB, directly to the Pi. For these setups always use an externally powered USB hub.

# An Operating System

The Raspberry Pi foundation provides several ready to use operating system images for the Pi. At the moment of 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”[[4]](#footnote-4) which can be downloaded at <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.

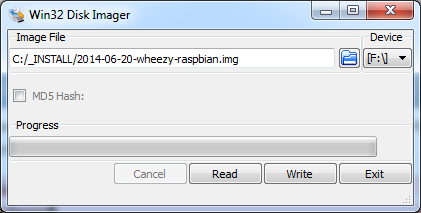


Figure Win32 Disk Imager

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 before inserting the SD card.

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 1). You will also have to connect a USB keyboard to the Pi to be able to navigate through the configuration menu you will be presented with on the initial boot.

Jump to Section 3.3 to configure the Pi for initial use by means of the configuration menu.

### SSH connection

Raspbian comes default with the SSH[[5]](#footnote-5) daemon enabled. This allows us to connect to the Pi from a remote 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[[6]](#footnote-6) 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.

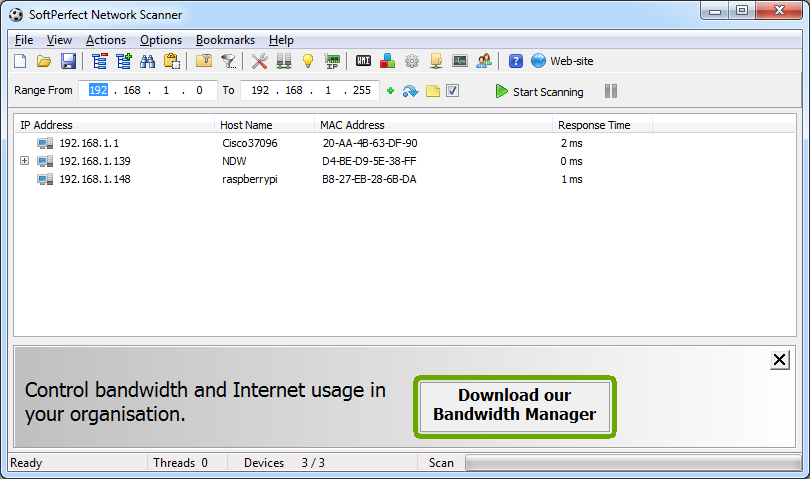


Figure Network scan using SoftPerfect Network Scanner

Another option is using WireShark[[7]](#footnote-7) 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[[8]](#footnote-8). 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 3. Once the connection is configured you can open it.

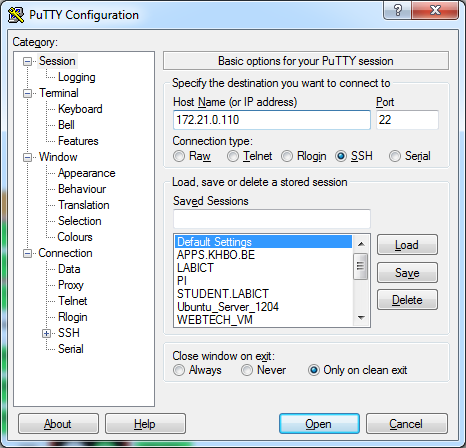


Figure Opening an SSH connection using Putty

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 4.



Figure 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.

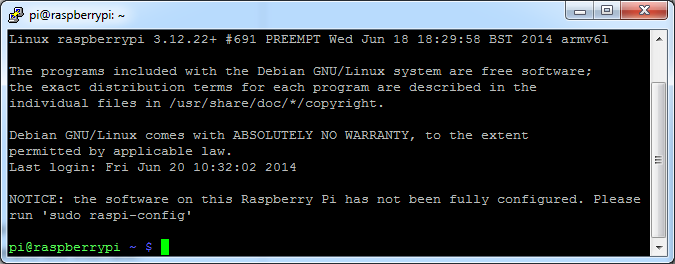


Figure The command line interface after logging in

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).

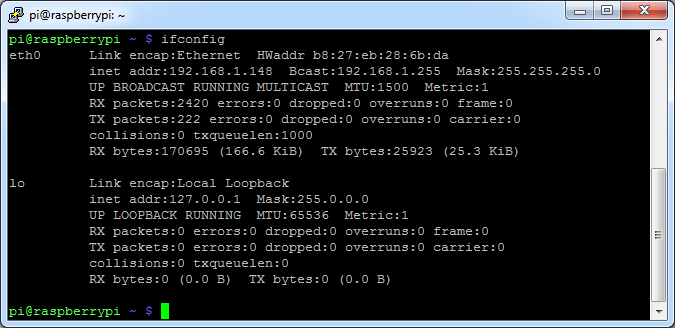


Figure Output of the ifconfig command

Jump to Section 3.3 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[[9]](#footnote-9).

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.

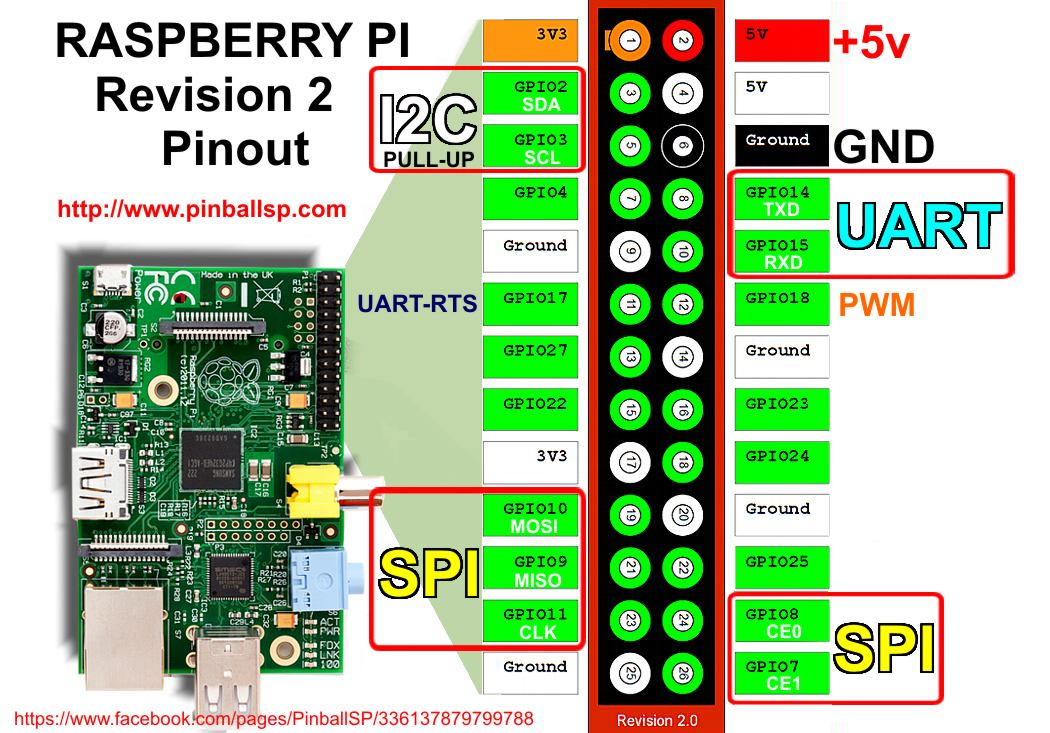


Figure Raspberry Pi GPIO pinout

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.

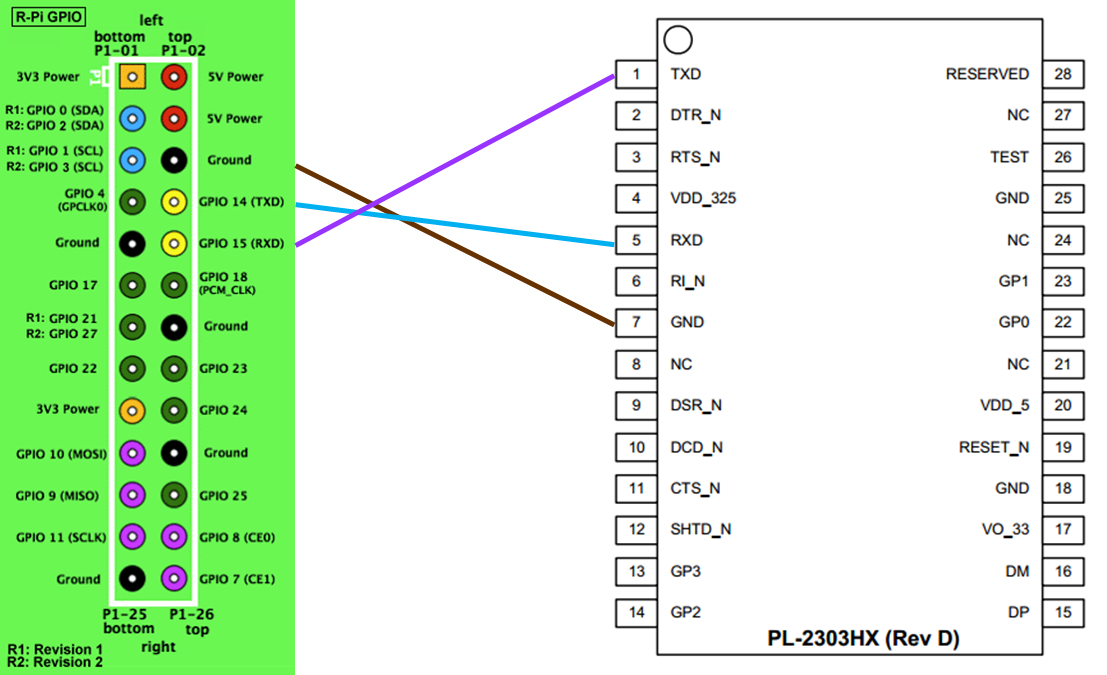


Figure Connecting the PL2303HX to the UART of the Raspberry Pi

Just as with SSH, you can use Putty for the serial terminal. Just select “serial” as connection type, “COMx”[[10]](#footnote-10) (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.

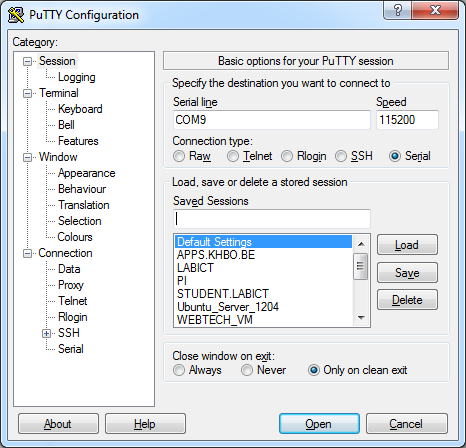


Figure Serial line connection parameters

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

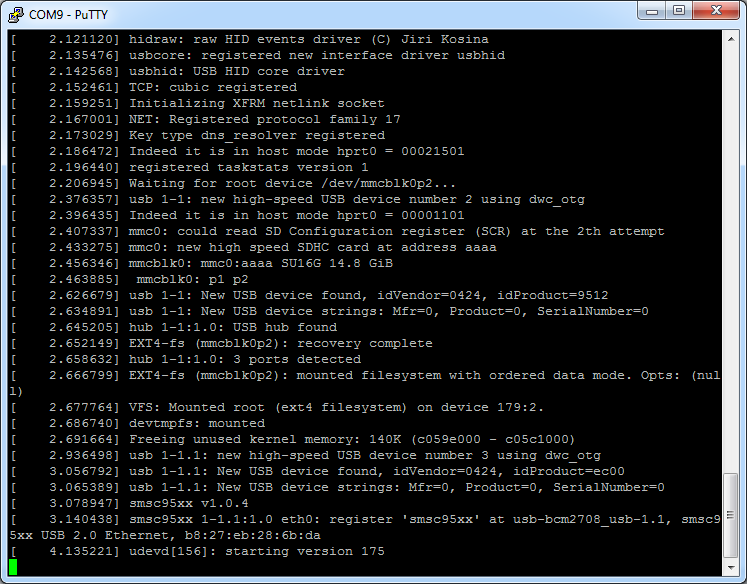


Figure Kernel messages on serial line

### Checking The Kernel Messages

While the RS232 connection will automatically show the kernel messages while booting, you can also retrieve the output from an SSH connection by using the command below:

$ dmesg

which outputs the kernel messages to your current terminal.

## Initial Configuration

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

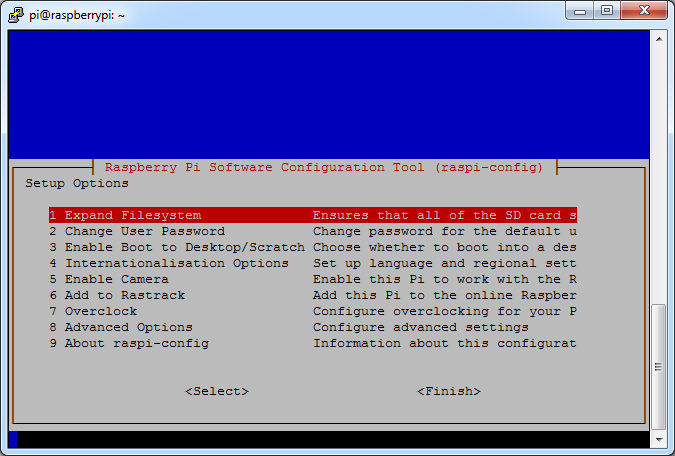


Figure Initial configuration tool

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 (with more than 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. In our case we will be using the command line interface.

### 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.

For our LABs we will leave the Pi with its default settings.

### 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.

Make sure to leave this enabled or you will be required to reconfigure your Pi using an external display and keyboard or by using a serial connection.

#### SPI (Serial Peripheral Interface)

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

#### 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.

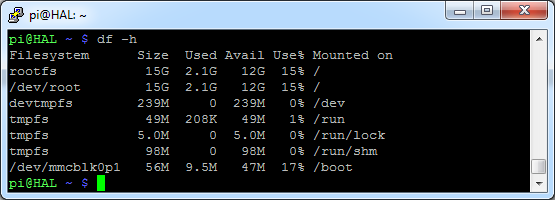


Figure Output of the disk free command after initial configuration

Also notice how the hostname of the device changed (HAL[[11]](#footnote-11) in this case). Rescanning the subnet using SoftPerfect Network Scanner should result in systems with all different hostnames.

# Linux Basics

While the information and examples presented here will be valid for most Linux distro’s, some information and commands may only work for Debian based distro’s.

## The MAN-Pages

The most important command you need to know is the “man” command, which provides an interface to the online reference manuals. By adding a command after the man command you can consult the man-pages for the particular command. These pages provide all the information you need to know to use the command such as general information, a detailed description of the arguments and usage examples. Let’s see an example:

$ man ls

Which gives the output shown in Figure 16. We can for example see if we add “-a” after the “ls” command it will also display hidden files[[12]](#footnote-12). You can scroll through the man-pages using the arrow keys.

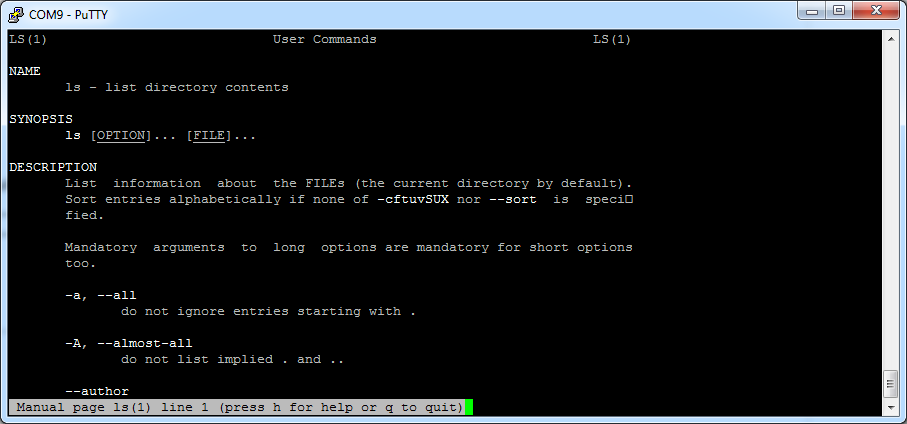


Figure Output of "man ls" command

Searching the current man-page can be done by first typing a slash (“/”), followed by your search term. Jumping to the next hit can be done by hitting the “n” key, while jumping back is done with “SHIFT-n”.

Exiting the man-pages is achieved using the “CTRL-c” combination.

## Traversing and Manipulating the File System

One of the principles guiding the organization of the file system is to allow it to be split across multiple disk partitions (or multiple disks) in a rational manner, and to allow appropriate pieces of it to be shared between machines. Key to this is the notion of the root partition.

When Linux boots, the kernel attaches a single file system partition all by itself. This is known as the root partition. Any other partitions that need to be attached are mounted by the mount command, usually under control of entries in the file “/etc/fstab”. Because in the early stages of startup, only the root file system is available, it must contain everything needed for the system to function and attach the other pieces of the file system.

Tools on the root partition include the init program (which starts all the other processes), a shell, mount and the “/etc/fstab” file. The File System Hierarchy standard specifies a number of directories that must lie within the root partition.

Figure xxx shows a typical Linux file system hierarchy.

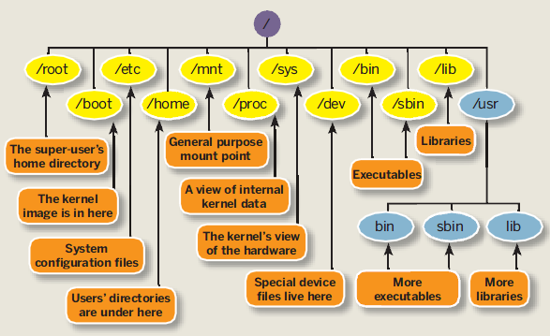


Figure A typical Linux file system hierarchy

The Linux file system basics are not really part of this course. However throughout this course you will gain a deeper knowledge of the file system by means of practice and self-education.

The most used commands to traverse and manipulate the file system of a Linux system are listed in Table 2. You can always use the man-command to get a detailed description.

Table Basic Linux commands

|  |  |
| --- | --- |
| Command | Description |
| ls | List files |
| cp | Copy files |
| rm | Remove files |
| mv | Move files |
| cd | Change working dir |
| chmod | Change file permission mode |
| chown | Change owner of file |
| cat | Concatenate files and output to terminal |
| touch | Create an empty file |
| mkdir | Make directory |

When interacting with the file system there are two ways to specify a filename. First is by using the absolute path, for example:

$ cat /proc/cpuinfo

Second is by using relative paths (relative to your current working directory). In this situation you can use the dot “.” as the current directory and the double dot “..” as the parent directory. Let’s for example say we are currently located in the “/usr” dir and we want to change the current dir to the pi’s home dir:

pi@HAL:/usr$ cd ../home/pi/

There is another shortcut to specify your own homedir, and that is by using the tilde “~”. You can immediately jump to your homedir by for example executing the following command:

$ cd ~

## Installing Packages

# Running a Virtual Machine

A virtual machine (VM) is an emulation of a particular computer system. This system can be based on an existing or hypothetical machine. As a user we can create such virtual machines and install an operating system of choice on them. This allows us to run a Linux distribution while working on a Windows machine and vice versa.

Several software packages are available to create and run virtual machines. Examples are VMware, Hyper-V[[13]](#footnote-13) which comes with Windows 8, Oracle VirtualBox, … For our labs we will be using VirtualBox as this is free, lightweight, easy to use and available for Windows, Linux, Mac and Solaris.

## Installing Oracle VirtualBox

Start by going to the download section of the website of VirtualBox (<https://www.virtualbox.org/>). Download the VirtualBox platform package for your system. At the moment of this writing the current version of VirtualBox is 4.3.16.

When running the installer package make sure to install VirtualBox with all features enabled as shown in Figure 18.

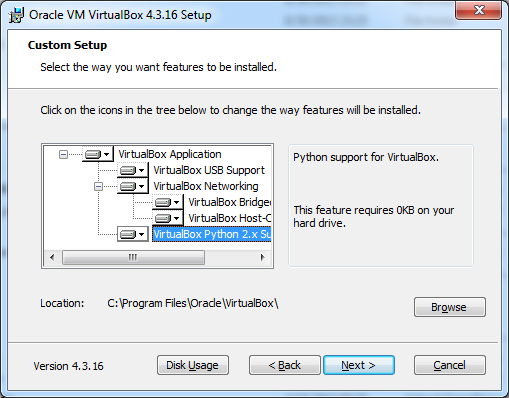


Figure Installing VirtualBox with all features enabled

Next we also need to install the extension pack which introduces USB2.0 support and some other extra features. You can download the extension pack on the same page as you downloaded the installer for VirtualBox. Just make sure to pick the correct version for your current VirtualBox version.

The installer of VirtualBox should also have created a virtual network adapter (such as can be seen in xxxxxxxx) which is used for private networking between the host machine and the virtual machine.

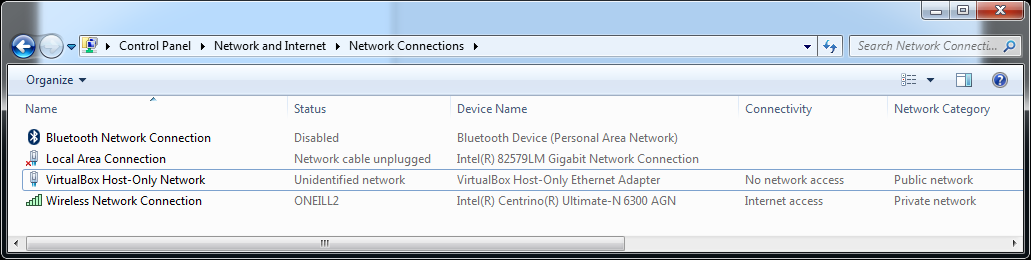


Figure VirtualBox virtual network adapter

Once you’re finished you can start the VirtualBox client and you should get the interface presented in Figure 19.

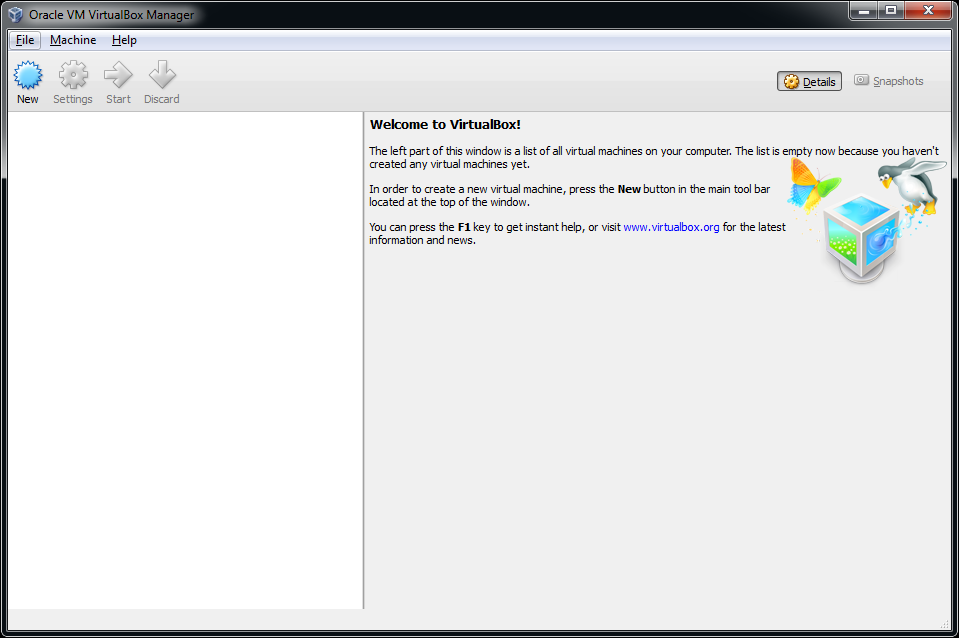


Figure Launching VirtualBox after installation

Under *Files => Preferences => General* you can change the default path for your virtual machines. Do take note that you will need about 25GB of free space for each VM. For these labs you will most likely only need 1 VM.

Under *Files => Preferences => Language* you can also change the default interface language if you wish.

## Creating a Virtual Machine

Creating a virtual machine is very simple as it just following the steps presented to you by the wizard. To start the process of creating a VM hit the *New* button on the main interface of VirtualBox.

The first step consist of giving your VM a name and selecting the operating system you will be running on the VM as shown in Figure 21. In our case we will use Ubuntu 12.04 LTS[[14]](#footnote-14) 64 bit Desktop edition. More on this later.

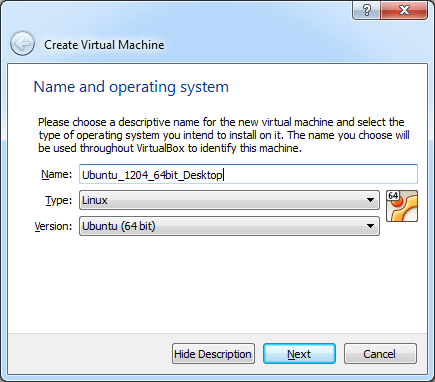


Figure Creating a VM - Step 1 - The name and OS

In step 2 we need to select the amount of memory we want to assign to the virtual machine (as seen in xxxxxxxxx). The recommended amount is 512MB. However if you have more than 4GB, select 1024MB or even 2048MB, which will improve the responsiveness and performance of the VM.

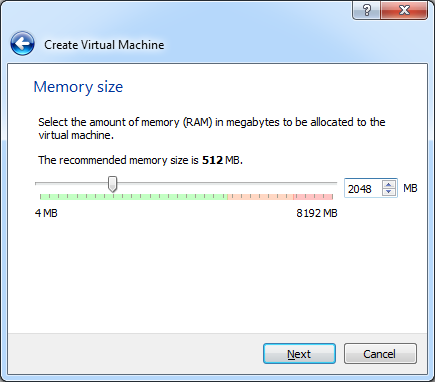


Figure Creating a VM - Step 2 - Amount of memory

Next we need to choose what if we want to create a new or use an existing virtual hard drive. Pick the option to create one now as depicted in Figure 23.This will launch another wizard which will lead us through the creation process of a virtual drive.

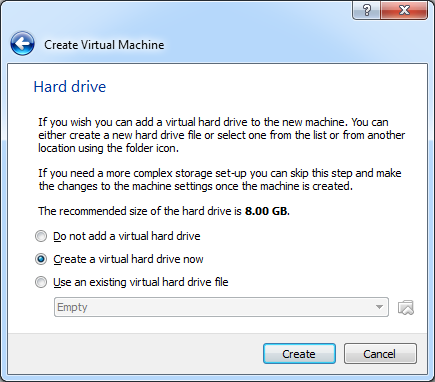


Figure Creating a VM - Step 3 - Virtual hard drive

The first screen will allow us to select what type of virtual drive file we want to create. Just leave the default option (VDI – Virtual Disk Image) as shown in Figure 24.

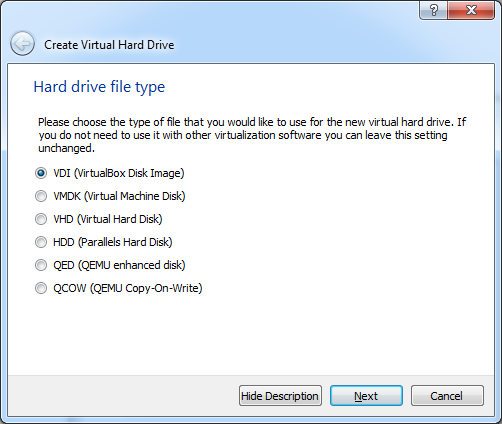


Figure Creating a VM - Step 3a - Hard drive file type

In the next step we get the option to create a dynamically allocated image or a fixed size image as can be seen in Figure 25. A fixed size image is faster but will take up the full space we select for the size of the virtual drive. A dynamically allocated image is faster but will only grow in size when needed. You will need to decide this for yourself based on the free space available on your host system.

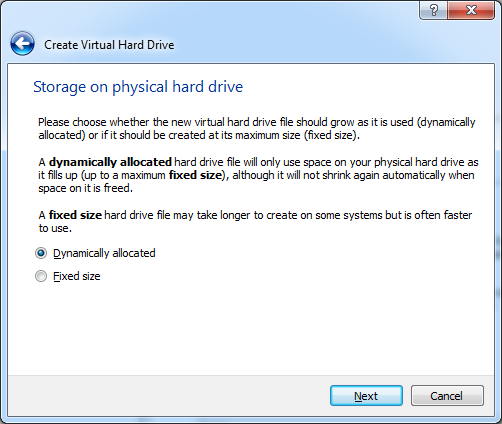


Figure Creating a VM - Step 3b - Virtual drive allocation method

Now we need to select the hard drive file location (leave it as is) and size of the drive. Make sure to select at least 25GB for the size as shown in Figure 26. Hitting create will finish the process of creating a VM.

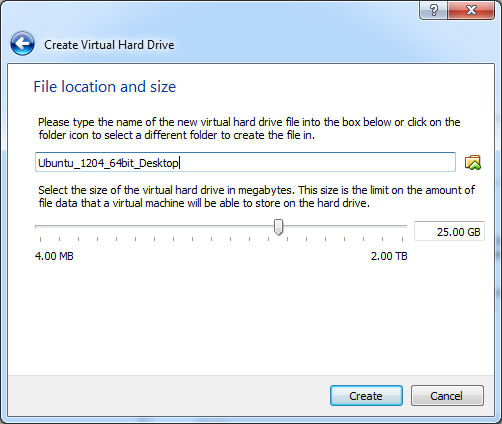


Figure Creating a VM - Step 3c - Location and size of the virtual drive

Your new VM should now appear in the list of VM’s on the left side of the VirtualBox main interface as seen in Figure 27. Selecting a VM in the list also displays some basic information about the VM.

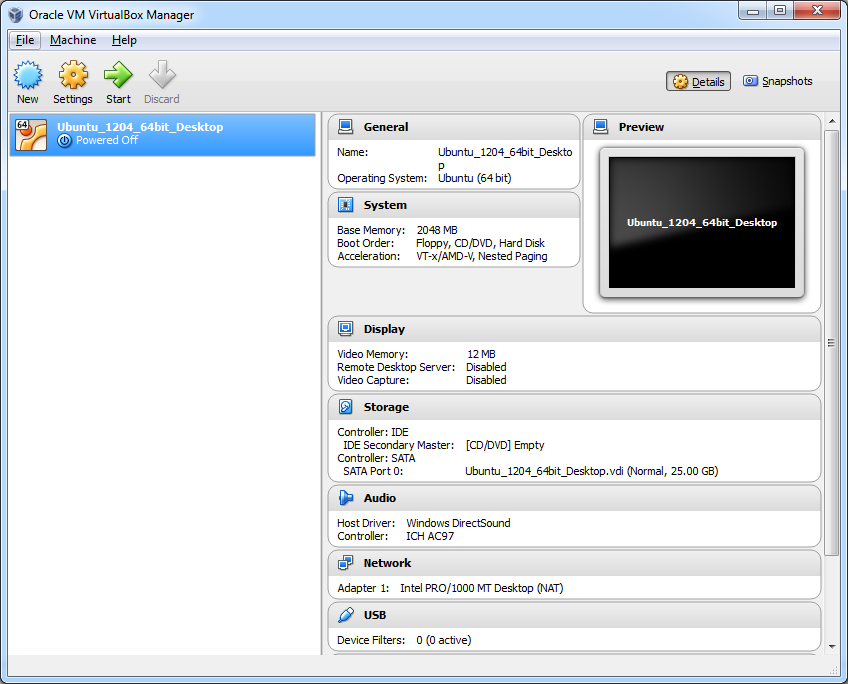


Figure A new VM is added to your current list of VM’s

## Configuring the Virtual Machine

# 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 Commands Overview

|  |  |
| --- | --- |
| Command | Description |
| ifconfig | Configure a network interface |
| dmesg | Print or control the kernel ring buffer |
| df | Disk free |
|  |  |

# References

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

1. The BCM2835 is a High Definition 1080p Embedded Multimedia Applications Processor [↑](#footnote-ref-1)
2. SoC, or System on a Chip, is a method of placing all necessary electronics for running a computer on a single chip. Instead of having an individual chip for the CPU, GPU, USB controller, RAM, Northbridge, Southbridge, etc., everything is compressed down into one tidy package. [↑](#footnote-ref-2)
3. A RISC-based computer design approach means ARM processors require significantly fewer transistors than typical CISC x86 processors in most personal computers. This approach reduces costs, heat and power use. These are desirable traits for light, portable, battery-powered devices—​including smartphones, laptops, tablet and notepad computers, and other embedded systems. A simpler design facilitates more efficient multi-core CPUs and higher core counts at lower cost, providing improved energy efficiency for servers. [↑](#footnote-ref-3)
4. 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-4)
5. 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-5)
6. <http://www.softperfect.com/products/networkscanner/> [↑](#footnote-ref-6)
7. 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-7)
8. 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-8)
9. The datasheet can be downloaded from <https://www.adafruit.com/datasheets/PL2303HX.pdf> [↑](#footnote-ref-9)
10. 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-10)
11. <http://en.wikipedia.org/wiki/HAL_9000> [↑](#footnote-ref-11)
12. In Linux hidden files start with a dot, for example “.ssh”. [↑](#footnote-ref-12)
13. Hyper-V, codenamed Viridian and formerly known as Windows Server Virtualization, is a native hypervisor; it can create virtual machines on x86-64 systems. Starting with Windows 8, Hyper-V supersedes Windows Virtual PC as the hardware virtualization component of the client editions of Windows NT. [↑](#footnote-ref-13)
14. Long-term support (LTS) is a term used to describe special versions or editions of software designed to be supported for a longer than normal period. It is particularly applicable to open-source software projects. [↑](#footnote-ref-14)