

Installation of an Operating System on Raspberry Pi

e-Yantra Team

June 4, 2015

Contents

0.1	Objective	2
0.2	Prerequisites	2
0.3	Hardware Requirement	2
0.4	Software Requirement	2
0.5	Theory and Description	3
0.5.1	Different versions of R-pi	3
0.5.2	A comparison between different models and versions of Raspberry pi	5
0.6	Experiment	6
0.7	References	10

0.1 Objective

This tutorial will help a Windows or a Linux user to install an operating system on Raspberry Pi successfully.

0.2 Prerequisites

- An idea about Windows or Linux operating system and their features.

0.3 Hardware Requirement

1. Raspberry Pi (I will be using Version 2 Model-B)
2. Adapter (To power up an R-pi)
3. SD card reader
4. PC (either Linux or Windows machine)

0.4 Software Requirement

1. Utorrent
2. 7-zip (or any other zip tool)

0.5 Theory and Description

The Raspberry Pi is a low cost, credit-card sized computer that plugs into a computer monitor or TV, and uses a standard keyboard and mouse. It is a capable little device that enables people of all ages to explore computing, and to learn how to program in languages like Scratch and Python. Its capable of doing everything youd expect a desktop computer to do, from browsing the internet and playing high-definition video, to making spreadsheets, word-processing, and playing games.

0.5.1 Different versions of R-pi

There exists two versions of Raspberry Pi:

- The original Raspberry Pi is based on the 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 (models B and B+) to 512 MB. The system has Secure Digital (SD) (models A and B) or MicroSD (models A+ and B+) sockets for boot media and persistent storage.[1]

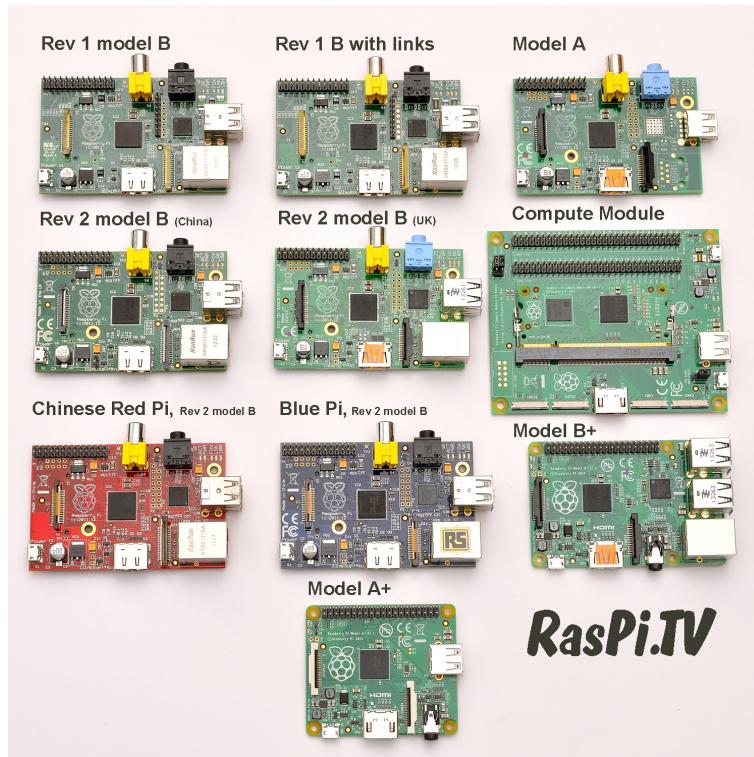


Figure 1: [3]

- The latest version Raspberry Pi 2 is available only in one configuration (model B) and features a Broadcom BCM2836 SoC, with a quad-core ARM Cortex-A7 CPU and a VideoCore IV dual-core GPU; 1 GB of RAM with remaining specifications being similar to those of the previous generation model B+. [1]

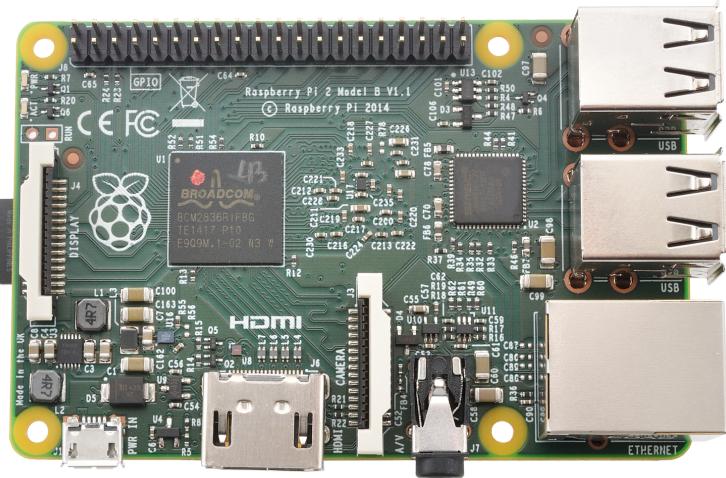


Figure 2: [4]

0.5.2 A comparison between different models and versions of Raspberry pi

Parameter	Model A	Model A+	Model B	Model B+	Generation 2 Model B
Target price(in US dollars):	25	20	35	25	35
SoC:	Broadcom BCM2835 (CPU, GPU, DSP, SDRAM, one USB port)	Broadcom BCM2835 (CPU, GPU, DSP, SDRAM, one USB port)	Broadcom BCM2835 (CPU, GPU, DSP, SDRAM, one USB port)	Broadcom BCM2835 (CPU, GPU, DSP, SDRAM, one USB port)	Broadcom BCM2836 (CPU, GPU, DSP, SDRAM, one USB port)
CPU:	700 MHz single-core ARM1176JZF-S	700 MHz single-core ARM1176JZF-S	700 MHz single-core ARM1176JZF-S	700 MHz single-core ARM1176JZF-S	900 MHz quad-core ARM Cortex-A7
Memory (SDRAM):	256 MB (shared with GPU) 512 MB	256 MB (shared with GPU)	512 MB (shared with GPU)	512 MB (shared with GPU)	1 GB (shared with GPU)
USB 2.0 ports:	1 (direct from BCM2835 chip)	1 (direct from BCM2835 chip)	2 (via the on-board 3-port USB hub)[38]	4 (via the on-board 5-port USB hub)	4 (via the on-board 5-port USB hub)
On-board storage:	SD / MMC / SDIO card slot (3.3 V with card power only)	MicroSD slot[32]	SD / MMC / SDIO card slot	MicroSD slot	MicroSD slot
On-board network:	None	None	10/100 Mbit/s Ethernet (8P8C) USB adapter on the third/fifth port of the USB hub (SMSC lan9514-jzx)	10/100 Mbit/s Ethernet (8P8C) USB adapter on the third/fifth port of the USB hub (SMSC lan9514-jzx)	10/100 Mbit/s Ethernet (8P8C) USB adapter on the third/fifth port of the USB hub (SMSC lan9514-jzx)
Low-level peripherals:	8 GPIO	17 GPIO plus the same specific functions, and HAT ID bus	8 GPIO	17 GPIO	17 GPIO
Power ratings:	300 mA (1.5 W)	200 mA (1 W)	700 mA (3.5 W)	600 mA (3.0 W)	800 mA(4.0 W)

Ref: [2]

0.6 Experiment

Instructions for installing operating system in Raspberry Pi:

Download Raspbian software from raspberry pi website. It should be available in the form of a zip file. Download it from the following link: <https://www.raspberrypi.org/downloads/> Also download win32DiskManager from the following link:<http://sourceforge.net/projects/win32diskimager/> and then install it.

A windows user should follow these steps:

- Insert an sd card(4GB or greater size) into the laptop in the memory card slot if available or use a memory card reader. Run win32DiskImager and choose the Raspbian image and select the drive corresponding to your sd card.
- Click "Write" to copy the files of the image on to the sd card.
- Eject SD card and insert it into the sd card slot of the Raspberry Pi.

A Linux user should follow these steps:

- Insert an sd card(4GB or greater size) into the laptop in the memory card slot if available or use a memory card reader.
- Run df -h to see what devices are currently mounted. The new device that has appeared is your SD card. The left column gives the device name of your SD card; it will be listed as something like /dev/mmcblk0p1 or /dev/sdd1. The last part (p1 or 1 respectively) is the partition number but you want to write to the whole SD card, not just one partition. Therefore you need to remove that part from the name (getting, for example, /dev/mmcblk0 or /dev/sdd) as the device for the whole SD card. Note that the SD card can show up more than once in the output of df; it will do this if you have previously written a Raspberry Pi image to this SD card, because the Raspberry Pi SD images have more than one partition.
- Note down your device name. You need to unmount it so that files can't be read or written to the SD card while you are copying over the SD image.
- Run umount /dev/sdd1, replacing sdd1 with whatever your SD card's device name is (including the partition number). If your SD card shows up more than once in the output of df due to having multiple partitions on the SD card, you should unmount all of these partitions.

- In the terminal, write the image to the card with the command below, making sure you replace the input file if= argument with the path to your .img file, and the /dev/sdd in the output file of= argument with the right device name. This is very important, as you will lose all data on the hard drive if you provide the wrong device name. Make sure the device name is the name of the whole SD card as described above, not just a partition of it; for example sdd, not sdds1 or sddp1; or mmcblk0, not mmcblk0p1.

```
dd bs=4M if=2015-05-05-raspbian-wheezy.img of=/dev/sdd
```

Please note that block size set to 4M will work most of the time; if not, please try 1M, although this will take considerably longer.

Also note that if you are not logged in as root you will need to prefix this with sudo.

- The dd command does not give any information of its progress and so may appear to have frozen; it could take more than five minutes to finish writing to the card. If your card reader has an LED it may blink during the write process. To see the progress of the copy operation you can run pkill -USR1 -n -x dd in another terminal, prefixed with sudo if you are not logged in as root. The progress will be displayed in the original window and not the window with the pkill command; it may not display immediately, due to buffering.

Instead of dd you can use dcfldd; it will give a progress report about how much has been written.

You can check what's written to the SD card by dd-ing from the card back to another image on your hard disk, truncating the new image to the same size as the original, and then running diff (or md5sum) on those two images.

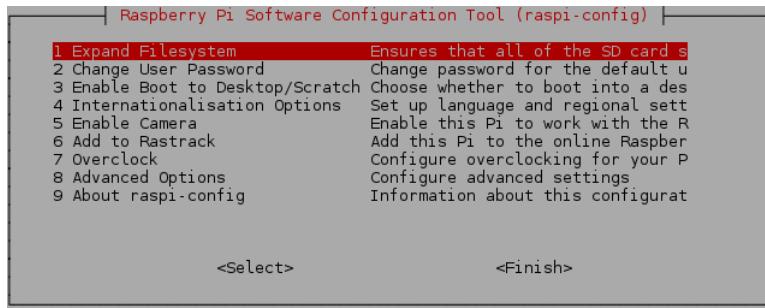
- The SD card might be bigger than the original image, and dd will make a copy of the whole card. We must therefore truncate the new image to the size of the original image. Make sure you replace the input file if= argument with the right device name. diff should report that the files are identical.

```
dd bs=4M if=/dev/sdd of=from-sd-card.img truncate --reference  
2015-05-05-raspbian-wheezy.img from-sd-card.img diff -s  
from-sd-card.img 2015-05-05-raspbian-wheezy.img
```

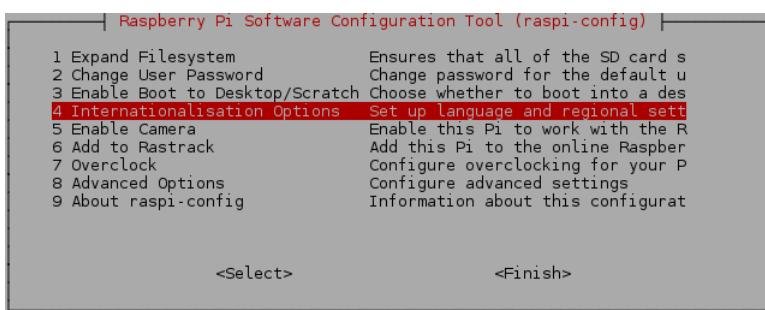
- Run sync; this will ensure the write cache is flushed and that it is safe to umount your SD card.
- Remove the SD card from the card reader and insert it into Raspberry Pi's SD card slot. [1]

After inserting the SD card into Raspberry Pi follow these steps:

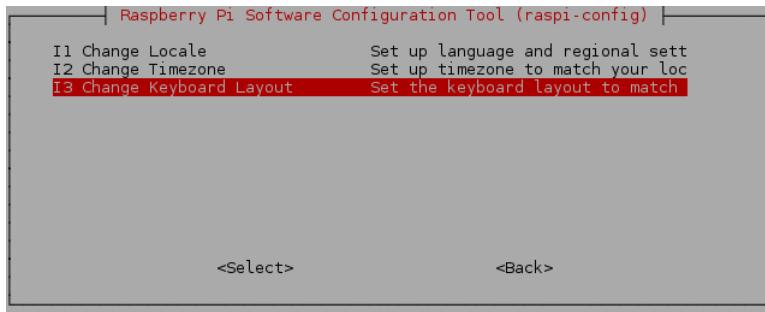
1. Connect the keyboard and mouse. Use HDMI cable to connect the board to the VGA monitor
2. Power on the board and the monitor. You will notice a set of code running on the monitor.
3. After a while a software configuration tool opens.



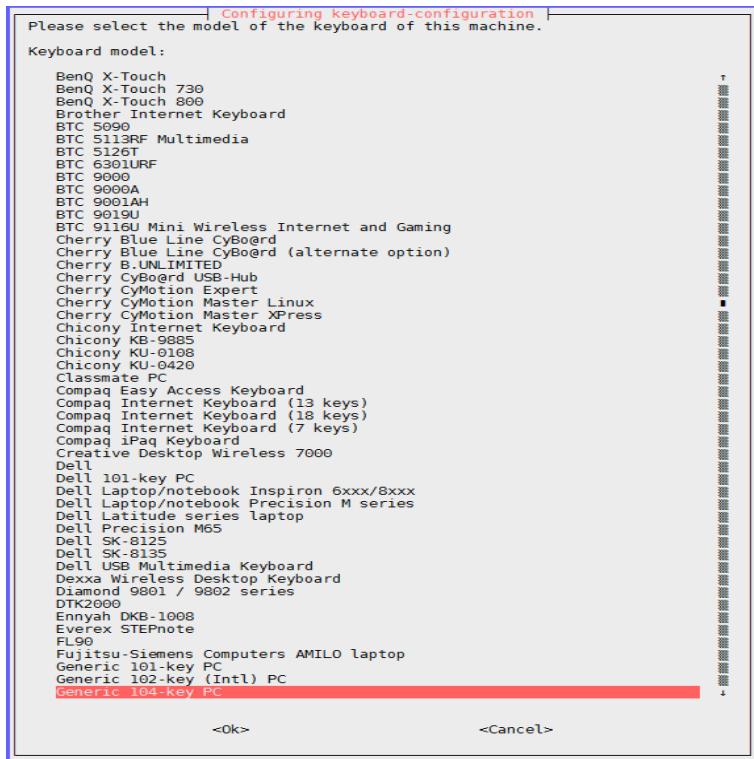
4. In case the size of the SD card is greater than 4GB you can select the 1st option i.e. Expand file system in the configuration tool as shown above.
5. Select the internationalisation options.



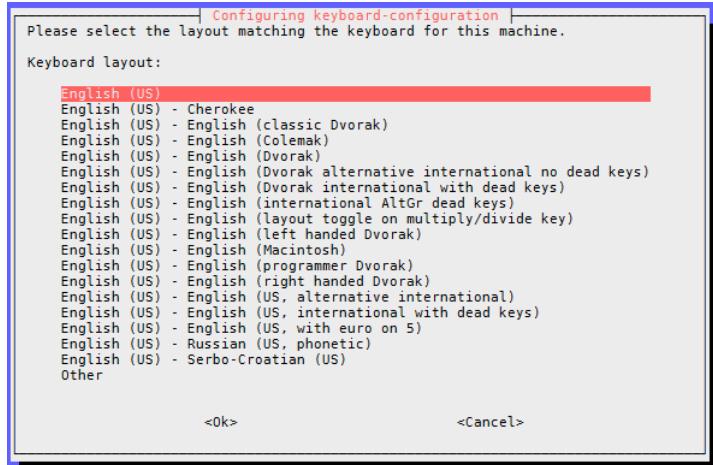
6. Change the keyboard and language settings as per your choice. Default keyboard will be of UK. Step by step change to US keyboard is as shown:
 - First select the option change Keyboard layout and click ok



- Then select the respective keyboard you are using and if you dont find your kind then select any of the generic types.



- Select "English US" which will be available at the top. Then for the rest of the process simply click enter till you reach the main configuration menu.



7. Click on finish using the keyboard by using arrow keys and enter key.
8. Click on "ok" when prompted for reboot.
9. Upon reboot, enter 'pi' as user id and 'raspberry' as password. Then type 'startx' to enter the graphical interface.
10. The linux based raspbian interface will be displayed on the monitor. Some of the softwares will be preloaded such as wolfram, python etc.
11. The command prompt will be the lx terminal.

With this we end the installation of Raspbian OS in R-Pi.

0.7 References

1. <https://www.raspberrypi.org/>
2. http://en.wikipedia.org/wiki/Raspberry_Pi
3. https://www.raspberrypi.org/wp-content/uploads/2014/11/Raspberry_Pi_Family_A-annotated-15001.jpg
4. http://assets.windowsphone.com/3f82dfe6-a179-4ddf-9738-91989190c3fa/IoT-rpi2-board_InvariantCulture_Default.png