



FREE eBook

LEARNING Raspberry Pi

Free unaffiliated eBook created from
Stack Overflow contributors.

#raspberry-

pi

Table of Contents

About.....	1
Chapter 1: Getting started with Raspberry Pi.....	2
Remarks.....	2
Examples.....	2
SD card Setup.....	2
Raspberry Pi Installation - Windows.....	2
Raspberry Pi Introduction - Hello World in C.....	5
Pretext.....	5
Hello World - My first program.....	5
Login with IPv6.....	6
Proxy Configuration.....	7
Raspberry Pi v2 and v3 Arch Linux Installation --- from Mac or Linux.....	7
Chapter 2: Create a scan station with scanbd (Raspbian).....	9
Examples.....	9
Introduction and installation.....	9
Configure sane to scan from the network.....	10
Configure scanbd to poll the local scanner.....	10
Identify the local scanner.....	10
Confirm that the local scanner is found by scanbd.....	11
What if "found device" doesn't appear ?.....	12
Configure xinetd to make scanbm listening on the network.....	12
Trigger actions from the scanner buttons.....	12
Chapter 3: Enable SSH on Raspbian.....	15
Introduction.....	15
Remarks.....	15
Examples.....	15
Enable SSH using a Windows computer.....	15
Use Raspberry Pi as headless system.....	16
Chapter 4: Operating systems.....	18
Introduction.....	18

Examples.....	18
Raspbian.....	18
Yocto based operating systems.....	18
Windows 10 IoT Core.....	18
OSMC.....	18
LibreELEC.....	18
Kali.....	19
Credits.....	20

About

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [raspberrypi](#)

It is an unofficial and free Raspberry Pi ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official Raspberry Pi.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Chapter 1: Getting started with Raspberry Pi

Remarks

This section provides an overview of what raspberry-pi is {it may also be described on Stack Exchange with the shortened term of "RPI"}, and why a developer might want to use it.

It has its own dedicated Stack Exchange site here at: <http://raspberrypi.stackexchange.com/>.

This section provides an overview of why a developer might want to use it.

It should also mention any large subjects within raspberry-pi, and link out to the related topics. Since the Documentation for raspberry-pi is new, you may need to create initial versions of those related topics.

Examples

SD card Setup

Start by installing the OS on the MicroSD card, either NOOBS or Raspbian, both provided by the Raspberry Pi foundation, available on [their site](#). NOOBS, which stands for New Out Of Box Software, is designed for beginners, and is the easiest to install on your SD card. You can either follow their [official instructions](#), or see below.

Format the SD card with a single partition of FAT32, using either the system drive management tool or a third party tool, such as the SD Association's [SD Formatter 4.0](#), or [GParted](#). Download the NOOBS [ZIP file](#) and unzip it. next, copy the files over to the partition that you just created. Ensure that you eject it before disconnecting it.

Raspberry Pi Installation - Windows

Pretext:

These are detailed instructions on how to set up a Raspberry Pi with the **Raspbian** operating system.

These instructions are somewhat Windows specific.

Some installation steps may apply to other operating systems as well, but keep the former in mind.

Contents

- Requirements
- Choosing an OS
- Installing the OS
- Basic Configuration

Requirements

1. Raspberry Pi (version 1, 2 or 3 - any model)
 2. Power supply (5V, 2000mAh usb adapter - recommended)
 3. MicroSD card (Class 10 recommended with at least 8GB - see list of compatible cards here: [eLinux](#))
 4. Computer with SD Card reader (or a USB SD Card reader)
 5. Monitor / TV with HDMI Port
 6. HDMI Cable
 7. USB Keyboard
 8. [Win32 Disk Imager](#)
-

Choosing an Operating System

- [NOOBS](#)

A collection of different operating systems for the Raspberry Pi, it contains Raspbian. In addition to Raspbian, NOOBS also includes Pidora, LibreELEC, OSMC, RISK OS, Arch Linux, Windows 10 IOT Core, and a few other choices. It is important to note that not all of these choices are available on every model of Raspberry Pi (for example, Windows IOT Core is only available on Raspberry Pi 2 and newer) . Installing any other OS than Raspbian will require an ethernet connection on your Raspberry Pi.

- [Raspbian](#)

The officially supported OS for the Raspberry Pi, it is a port of the OS known as Debian. Pre-installed with educational software and with a large community - this is the recommended OS for the Raspberry Pi

- [Windows 10 IOT Core](#) A simplified version of windows specifically designed for IOT (Internet Of Things) devices. It is important to note this is not a full version of windows, and does not have things user might expect, such as a start menu and the ability to run most Windows applications.
-

Installing the Operating System

SD card

1. Connect your SD Card to your computer
2. Download the Operating System you have chosen in **.img** format
(It might be compressed in a **.zip** file, which should be extracted after download)
3. Open Win32DiskImager
4. Select your SD Card
5. Select your OS **.img** file
6. Click Write

Raspberry Pi

1. Insert the SD Card into the Raspberry Pi
2. Connect your USB Keyboard
3. Connect your Monitor/TV
4. Connect an Ethernet cable (optional: but recommended!)

5. Plug in the USB Power cable in the Raspberry Pi

Operating System

If the OS was written correctly to the SD Card and the OS is a valid ARM Operating System, it should be installed automatically to the Raspberry Pi - with little to no interaction.

(This does however depend greatly on what kind of OS that has been chosen)

If you chose NOOBS an interface will show up and you will be able to choose what to install.

Basic Configuration

Command Line:

The command `sudo raspi-config` gives you access to the Raspberry Pi configuration.

Setup Options:

1 Expand Filesystem	Ensures that all of the SD card storage is available to the OS
2 Change User Password	Change password for the default user (pi)
3 Enable Boot to Desktop/Scratch	Choose whether to boot into a desktop environment, Scratch, or the command line
4 Internationalisation Options	Set up language and regional settings to match your location
5 Enable Camera	Enable this Pi to work with the Raspberry Pi camera
6 Add to Rastrack	Add this Pi to the online Raspberry Pi Map (Rastrack)
7 Overclock	Configure overclocking for your Pi
8 Advanced Options	Configure advanced settings
9 About `raspi-config`	Information about this configuration tool

- After a new and clean installation you are recommended to choose the first option: `Expand Filesystem`
- If you're European and not familiar with the English keyboard layout you can go to `Internationalisation Options` and change the keyboard language and more.
- It is recommended to avoid the `Overclock` menu as a beginner. Especially if you have the Raspberry Pi 3. The RPI 3 is known to get very hot, even if it is not overclocked. If you are thinking of overclocking the RPI 3, please obtain heatsinks or a fan for your board, to keep it from overheating.
- Enabling SSH access can be done in the `Advanced Options` menu, the credentials will be the default username and password.

GUI:

Installing Raspbian Jessie automatically boots into a graphical user interface, the aforementioned options are presented in the Options menu in the top left corner.

It is recommended for beginners to keep using the GUI, but if you bought a Raspberry Pi to learn the command line. The option to boot into the command line can be found in the Settings menu.

Note

Changing settings on the Raspberry Pi usually prompts a restart which happens when you accept the changes you've made.

Extra Info

Default Username: pi

Default Password: raspberry

Shutting down your Raspberry Pi is done by pulling the USB power plug.

(Choosing shutdown in the Raspbian GUI will simply put the Raspberry Pi in a diminished state, still using power - but less)

Restarting the Raspberry Pi is done by simply plugging in the cable after unplugging it.

(Unless the reboot option is chosen in the GUI or `sudo reboot` in the command line.

Raspberry Pi Introduction - Hello World in C

Pretext

This is an introduction to the Hello World program example on the Raspberry Pi written in C.

The following example makes use of the command line interface and is set up as a step-by-step guide.

Along with creating a Hello World program, the reader will be introduced to simple linux command line commands.

Note: This introduction was written for beginners.

Hello World - My first program

First step:

Making a directory that will contain source code.

- Locate your home directory by writing the following command `cd`
 - Make a new folder for your source code `mkdir programs`
(Tip: Linux is case sensitive when managing file and directory names)
 - Change to your new directory `cd programs`
-

Second step:

Writing your first program.

Linux systems offer a great variety of text editors, natively you will find Vim or Nano.

This example will make use of the Nano text editor.

- Create your source code file `nano helloworld.c`
- This will open an editor - exiting and saving will create the file in your folder.

The following code is the source code for the Hello World program:

```
/* My first program */
```



```
#include<stdio.h>

int main()
{
    printf("Hello World\n");
}
```

- After writing the code hit `ctrl + x` to exit the editor, hit `y` and then `enter` to save the changes.
- Write the command `ls` to check if the file is present in your directory.

Third step:

Compiling your first program.

- To compile our source code file `helloworld.c` we need to use a compiler, we will use the **GNU Compiler Collection** - also known as GCC.
- The following command compiles the source code to an executable binary program
`gcc helloworld.c -o myfirstprogram.bin`

The source code file is offered as an argument to the GCC compiler and `-o` defines another argument expressing that we would like the compiler to output something.

In this case we want it to output a `.bin` file that we named ourselves.

There is several other arguments you can use when compiling with GCC, an example would be `-Wall` which enables all warnings. This gives you information about any error GCC might encounter.

Fourth step:

Running your first program.

- Running a program on the Raspberry Pi is done by adding `./` in front of the name of the program that you want to run.
- We can run our program by writing `./myfirstprogram.bin`

The command should execute the program and produce `Hello World` in the console window.

Login with IPv6

Usually we struggle through the process of making login in the Raspberry Pi using `ssh` or other similar tools. But we can make the process more pleasant.

Once your Raspberry Pi is connected to a network it gets an IPv4 address and an **IPv6** address, that is based on the NIC's MAC address. The good thing about this is that the IPv6 address doesn't change even if the network has a DHCP service.

That being said, let's discover our IPv6 address. You only need to login once using IPv4 or using a monitor (HDMI or VGA). Open a terminal and type:

```
ifconfig
```

You will see something like:

```
eth0      Link encap:Ethernet  HWaddr 00:1C:C0:AE:B5:E6
          inet addr:192.168.0.1  Bcast:192.168.0.255  Mask:255.255.255.0
          inet6 addr: fe80::21c:c0ff:feae:b5e6/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:41620 errors:0 dropped:0 overruns:0 frame:0
          TX packets:40231 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:21601203 (20.6 MiB)  TX bytes:6145876 (5.8 MiB)
          Interrupt:21 Base address:0xe000
```

As you can see your IPv4 would look like `192.168.0.1` and your IPv6 would be `fe80::21c:c0ff:feae:b5e6` (notice the line that starts with `inet6 addr`).

So, with this information you can login using the `ssh` command with the following syntax:

```
ssh -6 pi@fe80::21c:c0ff:feae:b5e6%eth0
```

This approach seems more complicated, but the IPv6 address is defined as the `link local`, and, as it is based on the MAC address, unless you change it yourself, this will always work regardless of your IPv4 address.

Proxy Configuration

If you are behind a proxy and need to connect to the internet, you can use:

```
export http_proxy="http://username:password@host:port/"
```

For configuring the proxy inside `apt-get`:

```
cd /etc/apt/apt.conf.d
```

Create a file named `10proxy`:

```
sudo nano 10proxy
```

Without authentication add the following line:

```
Acquire::http::Proxy "http://yourproxyaddress:proxyport/";
Acquire::https::Proxy "http://yourproxyaddress:proxyport/";
```

With authentication:

```
Acquire::http::Proxy "http://username:password@yourproxyaddress:proxyport/";
```

The `/` at the very end was important. Without it it does not work.

Raspberry Pi v2 and v3 Arch Linux Installation --- from Mac or Linux

One of the best Linux distributions currently for Raspberry Pi (from now on, "RPI") is [Arch Linux](#). This web shows the installation for:

- [RPI2](#). ARMv7 architecture (32 bits).
- [RPI3](#). There are two options:
 - ARMv7 architecture (32 bits) or
 - AArch architecture (64 bits).

This tutorial advises you *not* to install the AArch option --- "Some of the hardware on the board may not work, or it may perform poorly".

Read [Getting started with Raspberry Pi online](#): <https://riptutorial.com/raspberry-pi/topic/5701/getting-started-with-raspberry-pi>

Chapter 2: Create a scan station with scanbd (Raspbian)

Examples

Introduction and installation

The purpose of this documentation is to use a scanner without any user interface. A common use is to upload a PDF scanned file directly to Google Drive or Dropbox just by pressing scanner-buttons.

scanbd permits to trigger actions from the scanner buttons, it uses sane.

What make things a little bit trickier is that by polling the scanner, scanbd locks the device. As we also need to scan from local tools (like scanimage), we need to set a proxy that's going to interrupt scanbd from polling when needed. This is the job of scanbm, which monitors the network and give priority over scanbd to any local access to the scanner.

Thus, we need to :

- Set up the local sane configuration to only scan from the network (through scanbm)
- Configure scanbd to have access and poll the local scanner

As root:

```
apt-get update
apt-get install libconfuse-dev libsane-dev libudev-dev libusb-dev xinetd
```

The latest version of scanbd can be find here :

<https://sourceforge.net/projects/scanbd/files/releases/>

```
# Download the latest version of **scanbd**.
wget https://sourceforge.net/projects/scanbd/files/releases/scanbd-1.4.4.tgz/download

# Uncompress
tar xvzf scanbd-1.4.4.tgz && cd 1.4.4

# Configure with the installation path in /etc/
./configure --disable-Werror

# install it
make
make install

# Add a dbus policy to authorize "saned" user to manage scanbd :
cp integration/scanbd_dbus.conf /etc/dbus-1/system.d/

# Add the init script to manage the daemon :
cp integration/scanbd.debian /etc/init.d/scanbd
```

scanbd is now installed but not yet configured.

Configure sane to scan from the network

The local configuration of sane is inside `/etc/saned.d`

`/etc/sane.d/dll.conf`

```
# /etc/sane.d/dll.conf - Configuration file for the SANE dynamic backend loader
#
# Backends can also be enabled by configuration snippets under
# /etc/sane.d/dll.d directory -- packages providing backends should drop
# a config file similar to dll.conf in this directory, named after the package.
#
# The next line enables the network backend; comment it out if you don't need
# to use a remote SANE scanner over the network - see sane-net(5) and saned(8)

net
```

`dll.conf` contains only the `net` backend.

`/etc/sane.d/net.conf` :

```
# This is the net backend config file.

## net backend options
# Timeout for the initial connection to saned. This will prevent the backend
# from blocking for several minutes trying to connect to an unresponsive
# saned host (network outage, host down, ...). Value in seconds.
connect_timeout = 3

## saned hosts
# Each line names a host to attach to.
# If you list "localhost" then your backends can be accessed either
# directly or through the net backend. Going through the net backend
# may be necessary to access devices that need special privileges.

localhost
```

At this point, **scanbm** is not yet configured, no scanner can be reached from the network. We need to set up `scanbm` along with `scanbd` to test the set up.

Configure scanbd to poll the local scanner

Identify the local scanner

By using `lsusb`, identify the **productid** (1909 here) :

```
pi:# lsusb
pi:# Bus 001 Device 005: ID 04a9:1909 Canon, Inc. CanoScan LiDE 110
```

With that **productId**, grep the correct configuration file (it depends of your scanner model, for me it is **genesys.conf**) :

```
pi:# grep 1909 /etc/sane.d/*conf
pi:# /etc/sane.d/genesys.conf:usb 0x04a9 0x1909
```

Copy the file inside your scanbd configuration folder:

```
cp /etc/sane.d/genesys.conf /usr/local/etc/scanbd/
cp /etc/sane.d/dll.conf /usr/local/etc/scanbd/
```

Edit **/usr/local/etc/scanbd/dll.conf** and replace `net` with the correct scanner backend:

```
# /etc/sane.d/dll.conf - Configuration file for the SANE dynamic backend loader
#
# Backends can also be enabled by configuration snippets under
# /etc/sane.d/dll.d directory -- packages providing backends should drop
# a config file similar to dll.conf in this directory, named after the package.
#
# The next line enables the network backend; comment it out if you don't need
# to use a remote SANE scanner over the network - see sane-net(5) and saned(8)

genesys
```

Confirm that the local scanner is found by scanbd

Now we are ready to test :

```
scanbd -d7 -f
```

Avoid as much as you can to manipulate the env var `SANE_CONFIG_DIR`. It can break things, the default value is already configured in `/usr/local/etc/scanbd/scanbd.conf` and in the init script.

Output should be similar to :

```
scanbd: foreground
scanbd: reading config file /usr/local/etc/scanbd/scanbd.conf
scanbd: debug on: level: 7
scanbd: dropping privs to uid saned
scanbd: dropping privs to gid scanner
scanbd: group scanner has member:
scanbd: saned
scanbd: pi
scanbd: drop privileges to gid: 110
scanbd: Running as effective gid 110
scanbd: drop privileges to uid: 110
scanbd: Running as effective uid 110
scanbd: dbus_init
scanbd: dbus match type='signal',interface='org.freedesktop.Hal.Manager'
scanbd: SANE_CONFIG_DIR=/usr/local/etc/scanbd/sane.d
```

```
scanbd: sane version 1.0
scanbd: Scanning for local-only devices
scanbd: found device: genesys:libusb:001:005 Canon LiDE 110 flatbed scanner
```

The most important here is `found device`.

What if "found device" doesn't appear ?

You should check again your configuration, especially the identification part of this documentation and clear any eventual `SANE_CONFIG_DIR` env var wrongly set by typing :

```
unset SANE_CONFIG_DIR
```

Configure xinetd to make scanbm listening on the network

Create `/etc/xinetd.d/sane-port` which contains :

```
service sane-port
{
    port          = 6566
    socket_type   = stream
    wait         = no
    user         = saned
    group        = scanner
    server       = /usr/local/sbin/scanbm
    server_args  = scanbm -c /usr/local/etc/scanbd/scanbd.conf
    disable     = no
}
```

!/ Double-check the path of `scanbm` with `which scanbm`

Then, restart `xinetd` and start `scanbd` :

```
service xinetd restart
/etc/init.d/scanbd start
```

`scanimage -L` should now display the scanner.

Most issues at this stage concern the permissions. Ensure that `scanimage -L` works when you are logged as `saned` by doing `sudo -u saned -s`.

If not, check USB permissions : `ls -al /dev/bus/usb/***`

Trigger actions from the scanner buttons

Actions are located in `/usr/local/etc/scanbd/scanbd.conf`. I have 4 buttons that are **scan**, **copy**, **email** and **file**.

The default config file doesn't include all actions per default, you will probably have to add the block manually. You can have less or more buttons depending of your scanner

model.

For each action, we will set a custom path for the **script** option.

```
action scan {
    filter = "^scan.*"
    numerical-trigger {
        from-value = 1
        to-value    = 0
    }
    desc    = "Scan to file"
    # script must be an relative path starting from scriptdir (see above),
    # or an absolute pathname.
    # It must contain the path to the action script without arguments
    # Absolute path example: script = "/some/path/foo.script"
    script = "/home/pi/scan.sh"
}
```

Don't forget to comment any other default action at the end of scanbd.conf :

```
# devices
# each device can have actions and functions, you can disable not relevant devices
#include(scanner.d/avision.conf)
#include(scanner.d/fujitsu.conf)
#include(scanner.d/hp.conf)
#include(scanner.d/pixma.conf)
#include(scanner.d/snapscan.conf)
#include(scanner.d/canon.conf)
```

You can now create your custom script to handle each action :

Each line relative to `/sys/class/leds/led0/trigger` are for controlling the LED to monitor what's going on. You can do whatever you want, `cat /sys/class/leds/led0/trigger` gives you all different pattern of lights.

/home/pi/scan.sh

```
#!/bin/bash

# don't forget to create the folder
scan_dir=/home/pi/scanned-files
datetime=`date +%F_%H%M%S`

echo none >/sys/class/leds/led0/trigger

case $SCANBD_ACTION in
    scan)
        filename=file-$datetime
        logger -t "scanbd: $0" "$SCANBD_DEVICE $SCANBD_ACTION - scanning --resolution 150 --mode
Color --depth 8 --format=tiff to $scan_dir/$filename.jpg"
        echo timer >/sys/class/leds/led0/trigger
        scanimage -d $SCANBD_DEVICE --resolution 150 --mode Color --depth 8 --format=tiff --
brightness 5 --contrast 20 | convert tiff:- -compress jpeg $scan_dir/$filename.pdf
        echo none >/sys/class/leds/led0/trigger
        logger -t "scanbd: $0" "Finished scanning"
    ;;
```



```
email)
  logger -t "scanbd: $0" "Emailing $scan_dir/file-*.pdf"
  echo heartbeat >/sys/class/leds/led0/trigger
  # here are the lines to send the file
  echo none >/sys/class/leds/led0/trigger

esac
```

Read [Create a scan station with scanbd \(Raspbian\)](https://riptutorial.com/raspberry-pi/topic/6701/create-a-scan-station-with-scanbd--raspbian-) online: <https://riptutorial.com/raspberry-pi/topic/6701/create-a-scan-station-with-scanbd--raspbian->

Chapter 3: Enable SSH on Raspbian

Introduction

Previous versions of Raspbian (prior to November 2016) had SSH enabled by default. They also had a default username (pi) and password (raspberry) this was done to make first time setup for new users easier. But this obviously represents a large security hole. The new release (November 2016) disables SSH by default, and will also display a warning if SSH is enabled without changing the the default password.

Remarks

The contents of the ssh file do not matter - you can create an empty file like we did above.

During boot up the Pi will look for this file. If it exists SSH will be enabled and the file deleted.

Examples

Enable SSH using a Windows computer

To enable SSH create a file called ssh in the /boot directory of your SD card.

1. Identify the drive letter associated with your SD card.
2. Open a command prompt (press **Win+R** on your keyboard to open the Run window. Then, type **cmd**)
3. Enter the following at the command prompt (replacing DriveLetter with the letter you identified in step 1):

```
echo.>DriveLetter:/ssh
```

4. Press `Enter`

Verify that your SD card contains a file called ssh.

1. Enter the following at the command prompt (again replacing DriveLetter with the letter you identified in step 1)

```
dir DriveLetter:
```

2. Press `Enter`

You should see an ssh file in the directory listing.

To avoid the security warning mentioned above (re: SSH enabled with the default password) change the default password for the Pi user. This can be done either from the command line with

the `passwd`, the `raspi-config` script, or settings menu in the Pixel desktop.

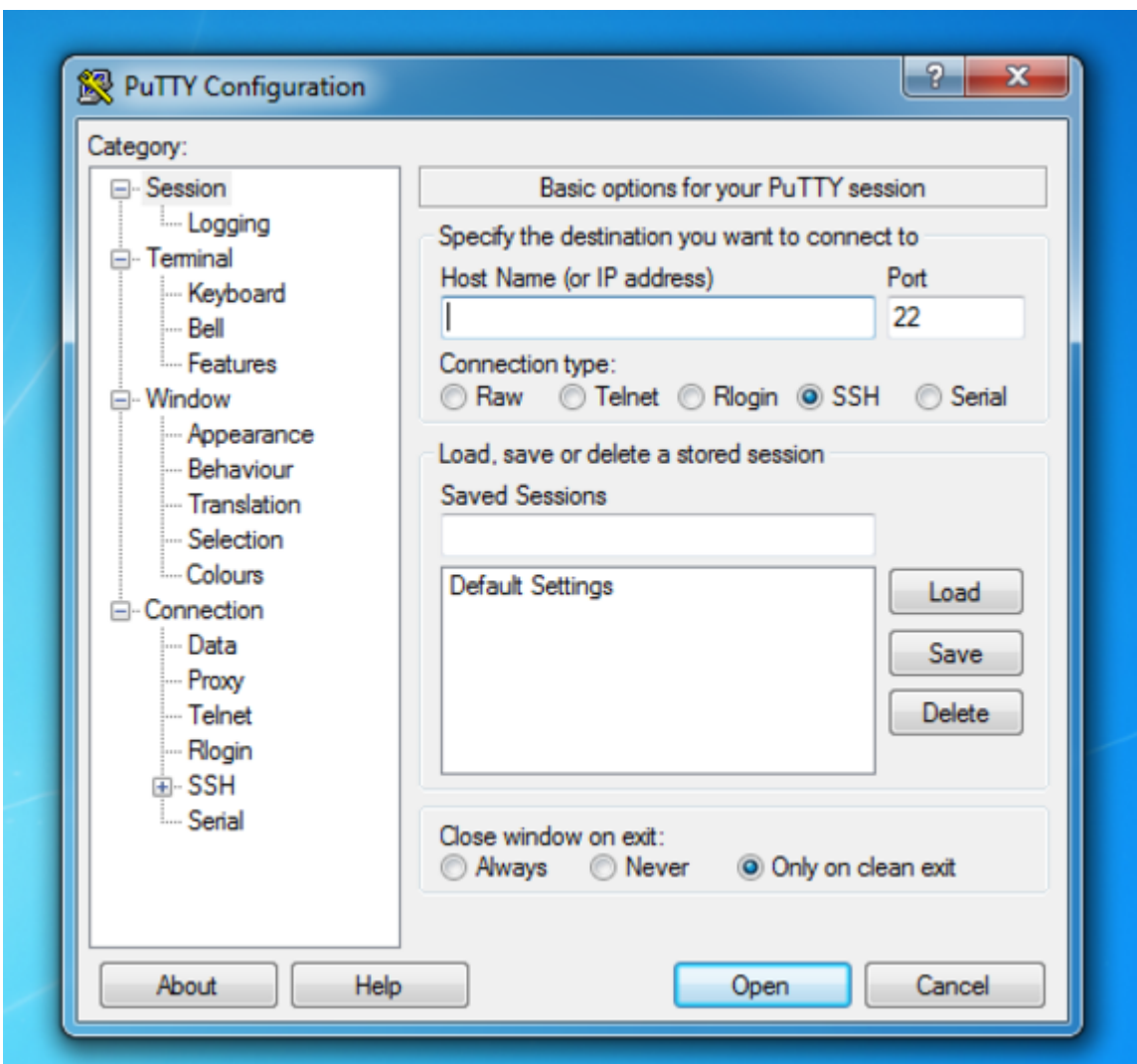
Use Raspberry Pi as headless system

Above example helps us how to turn on SSH on Pi. This example is the prerequisite of above example.

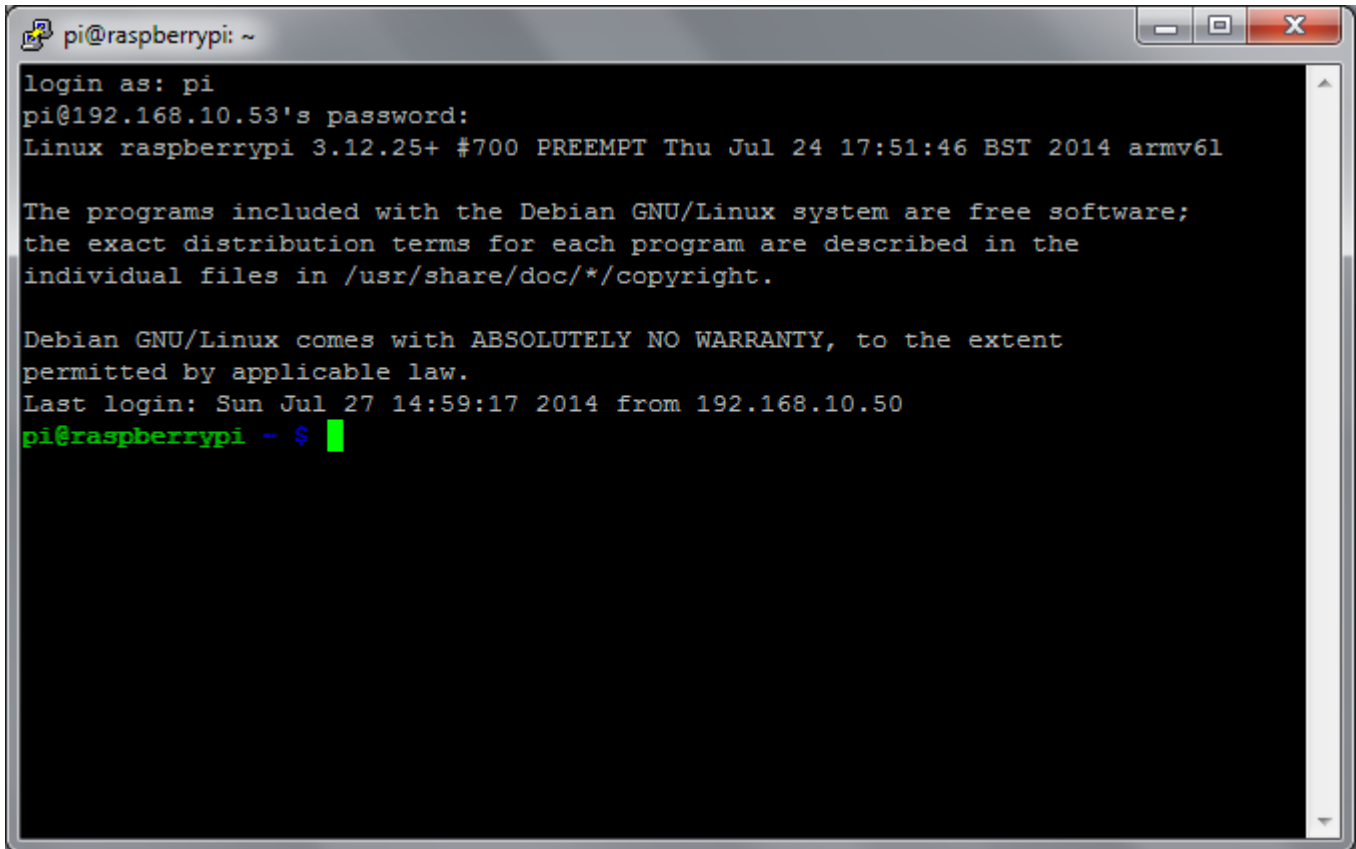
For Linux OS: Open the terminal and write down this `ssh pi@your local IP address`. Then you will need to provide the password.

Note: Here **pi**: your raspberry pi **username** and **your local IP address**: the IP address of your device. You will get this IP address from your router, DHCP client list. However, this topic is out of scope.

For Windows OS: We can use [Putty](#) to login via ssh. At Host Name write your Raspberry pi IP address.



After pressing **Open**. You should provide your **username** and **password**. After the successful attempt, you will get this screen.

A terminal window titled "pi@raspberrypi: ~" with standard window controls. The terminal output shows a successful SSH login for user 'pi' at IP 192.168.10.53. It displays the system version (Linux raspberrypi 3.12.25+ #700 PREEMPT Thu Jul 24 17:51:46 BST 2014 armv6l), a copyright notice for Debian GNU/Linux, a warranty disclaimer, and the last login time (Sun Jul 27 14:59:17 2014 from 192.168.10.50). The prompt is now "pi@raspberrypi - \$" with a green cursor.

```
login as: pi
pi@192.168.10.53's password:
Linux raspberrypi 3.12.25+ #700 PREEMPT Thu Jul 24 17:51:46 BST 2014 armv6l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sun Jul 27 14:59:17 2014 from 192.168.10.50
pi@raspberrypi - $ █
```

Read Enable SSH on Raspbian online: <https://riptutorial.com/raspberry-pi/topic/8355/enable-ssh-on-raspbian>

Chapter 4: Operating systems

Introduction

This section contains information about various operating systems available for the Raspberry Pi's.

Examples

Raspbian

The most popular operating system for the Raspberry Pi is a Debian based Raspbian.

It is officially supported by the Raspberry Pi Foundation.

Raspbian can be downloaded from official Raspberry Pi site in one of two variants:

- With desktop environment
- Lite- Minimal image

Starting September 2016 Raspbian is shipped with PIXEL desktop environment ("Pi Improved Xwindows Environment, Lightweight")

Yocto based operating systems

It is possible to create own Linux distribution using Yocto Project.

For Raspberry Pi- there is a specific layer [meta-raspberrypi](#) that needs to be used to create an image.

Windows 10 IoT Core

Windows 10 IoT Core is available only for Raspberry Pi 2 and 3. It is important to note that this is not a full version of windows as one might expect, it is designed specifically for IoT (Internet of Things) applications. It will not run most standard windows applications, and does not have a start menu. It boots to a page with some example code and videos, and has a command line interface.

OSMC

OSMC is a free and open source media player based on Linux.

LibreELEC

Linux distribution with features required to run [Kodi](#).

Kali

A fully featured OS designed around penetration testing.

Download Link:

<https://www.offensive-security.com/kali-linux-arm-images/>

Requirements:

1. 8GB Class 10 SD Card Minimum
2. 16GB Class 10 SD Card Recommended

Default Login:

```
Username: root  
Password: toor
```

Notes

1. Install gparted and expand partition to full size
2. Run apt-get install kali-linux-full to get all the standard tools available
3. Runs XFCE Desktop Environment

Read Operating systems online: <https://riptutorial.com/raspberry-pi/topic/8656/operating-systems>

Credits

S. No	Chapters	Contributors
1	Getting started with Raspberry Pi	An Epic Person , andpei , atilacamura , Cicero , Community , ctafur , Patrick L , RvT , S.Spieker , SlySven
2	Create a scan station with scanbd (Raspbian)	Pierre Maoui
3	Enable SSH on Raspbian	M Hornbacher , Saiful Azad , Steve Robillard
4	Operating systems	lewiatan , M Hornbacher , Patrick L