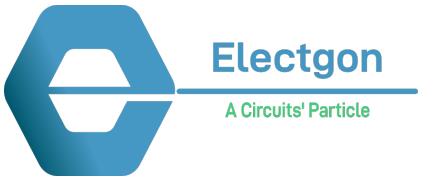


Raspberry Pi

Software View

ELECTGON
www.electgon.com
ma_ext@gmx.net

28.04.2018

Contents

1 Introduction 1
2 Installing Raspbian 1
3 First Settings with Raspbian 3
3.1 Expand Filesystem 3
3.2 Keyboard 4
3.3 Assign Static IP 8
3.4 Wireless Connection Setup 9
3.4.1 Credentials 10
3.5 Enable SSH 10
3.6 Other Packages 12
4 Remote Connection 12
5 BackUp Operations 13

Abstract

Previous tutorial was introduced to give overview of Raspberry Pi board as a Hardware view. Software settings for Raspberry Pi are needed to be declared including its Operating System. Only basic settings are needed to be understood. Afterwards, Raspberry Pi can act as simple computer machine that allow user to install his applications as long it is compatible with the running operating system. This document provides needed steps to have Raspbian as a running operating system and some other configurations that may be needed.

1 Introduction

Once you have your Raspberry Pi board is ready and connected with needed peripherals, you can start working with it as long you have an operating system is installed in its attached micro SD card. Available operating system images for Raspberry Pi can be found in its website

https://www.raspberrypi.org/downloads/

On this page you can find ready images like Ubuntu mate, Ubuntu Snappy, Windows 10 and Raspbian. This Raspbian is built to be optimized for Raspberry Pi. This operating system (or root file system to be more specific) is built based on Linux kernel which enables user configurations to be applied (like realtime patch for instance). The following discussion will use Raspbian to show how to work with software settings for Raspberry Pi.

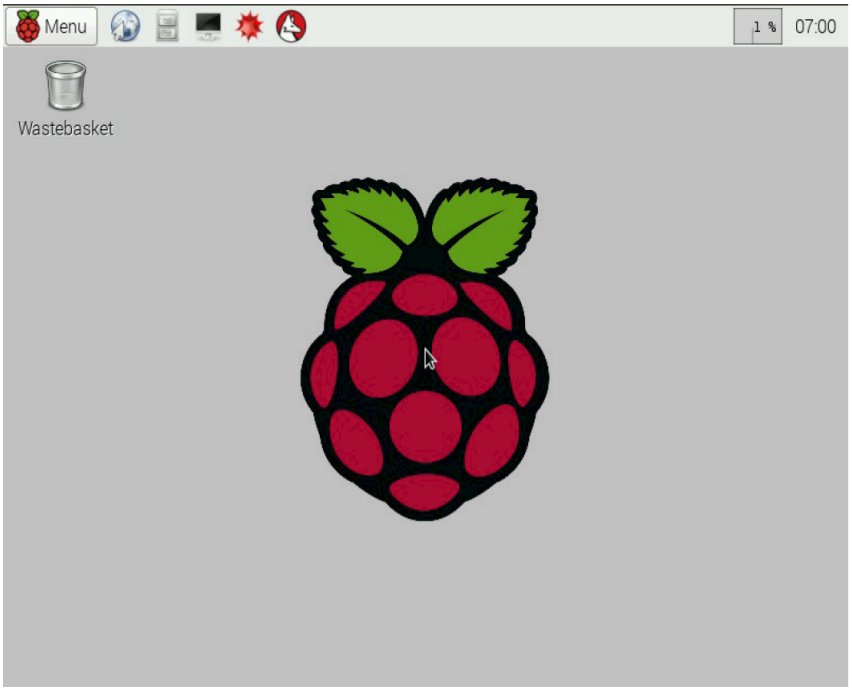


Figure 1: Raspbian Desktop

2 Installing Raspbian

When you choose which operating system to download. You have to prepare SD card to install this operating system (Raspbian in our case) into this SD card. Also we have to find a tool which we can use to install Raspbian on the SD card. Raspbian will be downloaded as an operating system image (file with extension .img). This means we need to write this .img file to the SD card. If your host machine (machine that you will use to write the .img file into SD card) is windows, you can use one software called Etcher to write the image. As shown in figure 2, you can use this tool easily.

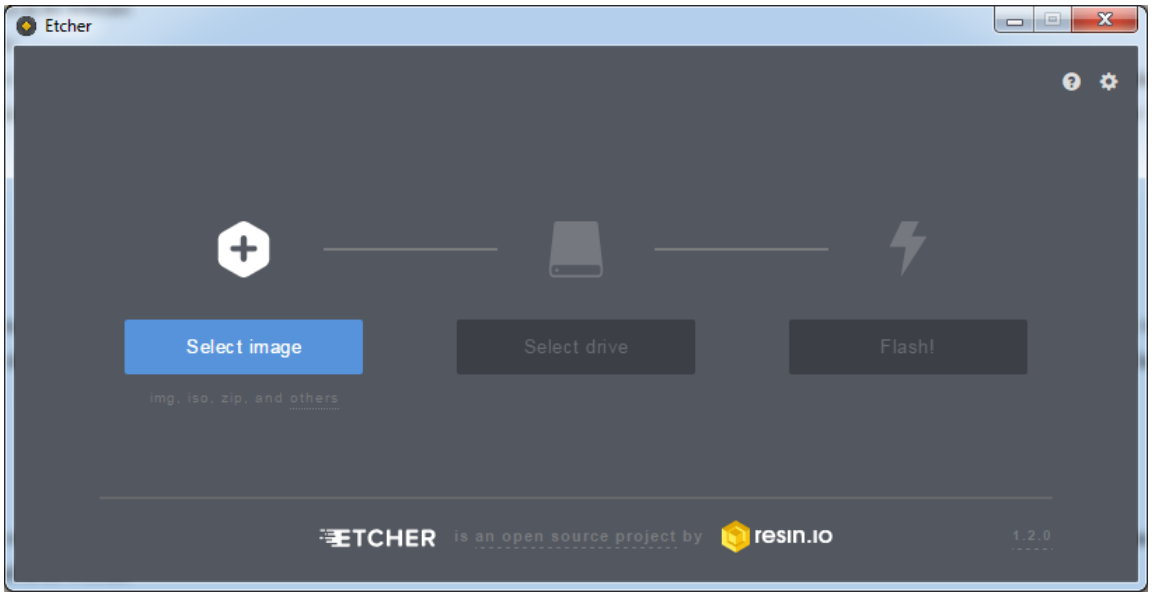


Figure 2: Image Writing Tool in Windows

In case you have Linux as the host machine, dd package can be used to write the image. The following steps explains how to write the image using dd.

- Connect your SD card to your machine.
- Open shell interface.
- Type *df -h* to know driver of your SD card
- Unmount this driver to start read/write, use this command to unmount

```
$umount /dev/sdb
```

- Usually SD card is seen at sdb or sdd, so whatever the driver you have, unmount it. Also note that the SD card may appear in 2 drivers. If happens, unmount both drivers

```
$umount /dev/sdb1
$umount /dev/sdb2
```

- Now navigate to your downloaded Raspbian and start to install it on the SD card using dd command

```
$sudo dd bs=4M if=2015-11-21-raspbian-jessie.img of=/dev/sdb
```

This may take sometime but after finished, you are done with a ready SD card to work with Raspbian.

Important to note here that the SD card will be partitioned into two partitions boot & root. It is highly recommended to open the boot partition and add an empty file named as 'ssh' without any extension. This file then will enable you to access the Raspberry Pi remotely using ssh protocol.

- Plug in this SD card then start working with Raspbian.

3 First Settings with Raspbian

Some important settings for the Raspberry Pi shall be set before continuing. These settings will make working with Raspberry Pi more easier. The following steps assume that Raspbian has been successfully installed and booted up. So when it starts open a shell interface in Raspbian to execute the following.

3.1 Expand Filesystem

In order to make use of all of your SD card space, make sure that Raspbian has been successfully expanded on the disk space. You can check first how the sd card is partitioned by typing

```
$df -h
```

then type:

```
$sudo raspi-config
```

the following window shall be opened

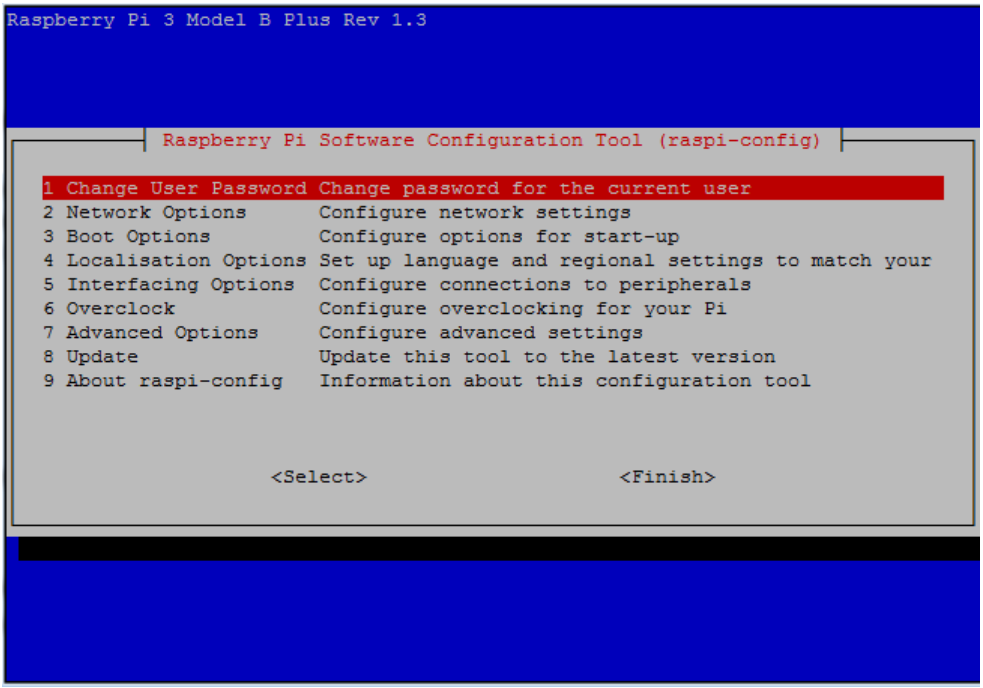


Figure 3: Main RaspiConfig Window

Enter through 'Advanced Options' then Choose First option "Expand Filesystem"

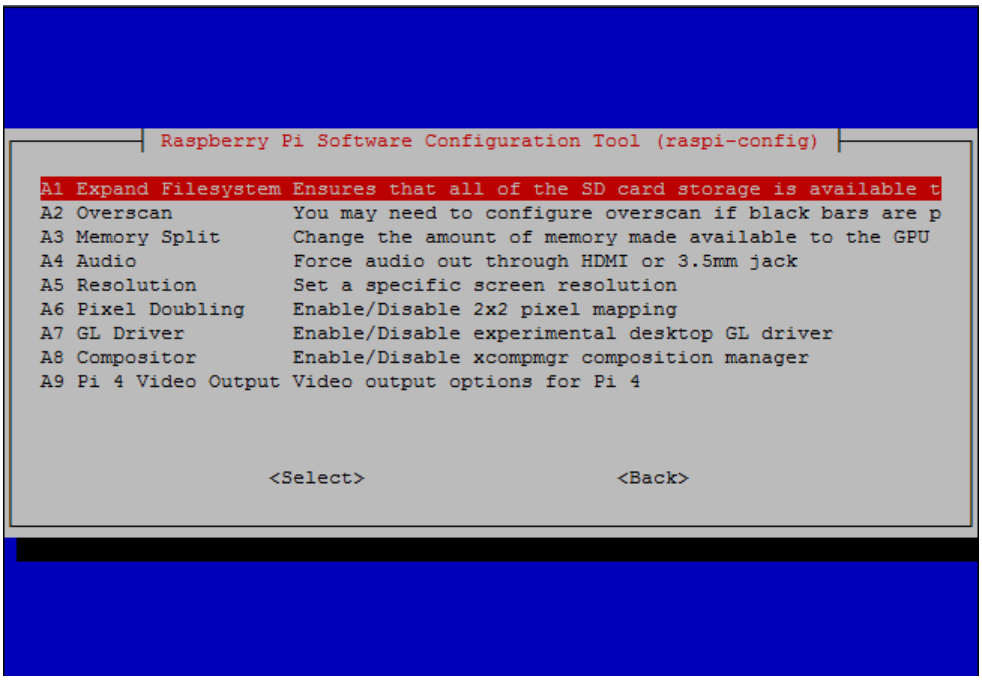


Figure 4: Expand File System

Click OK to confirm operation.

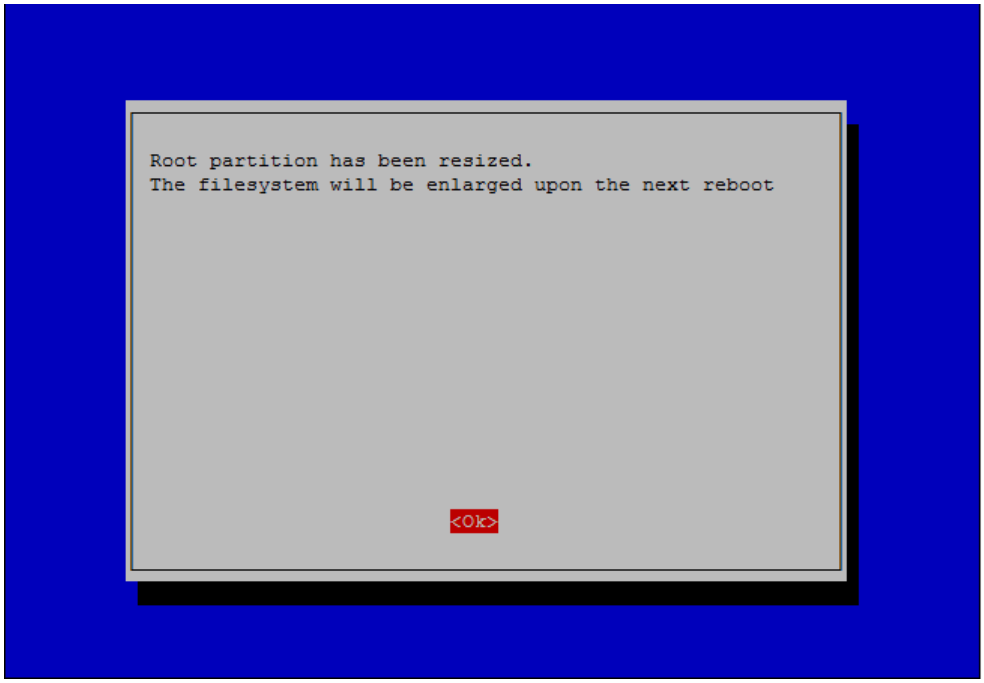


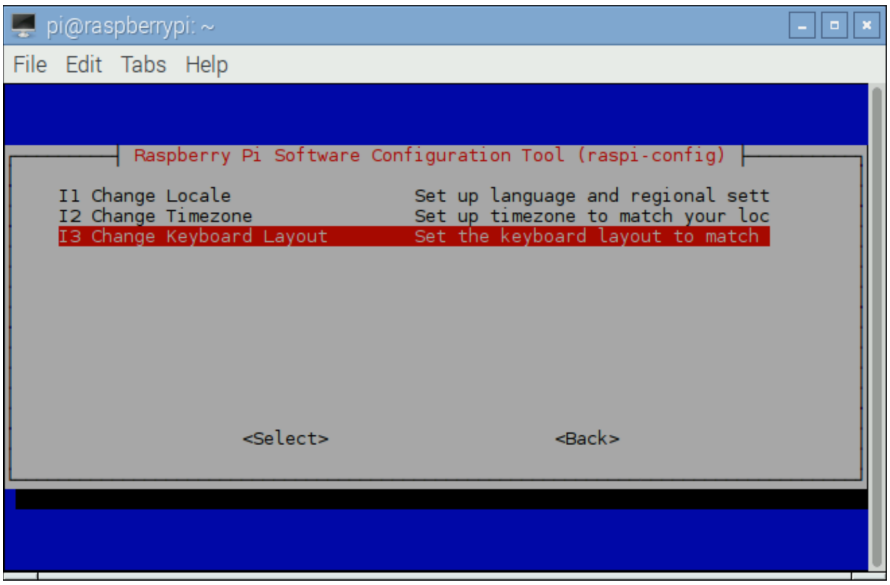
Figure 5: Confirm Expand File System

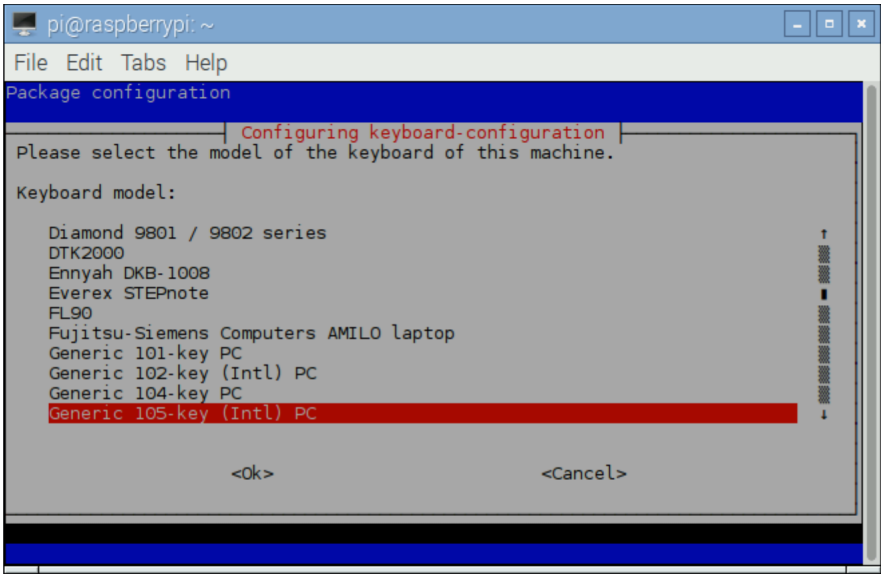
3.2 Keyboard

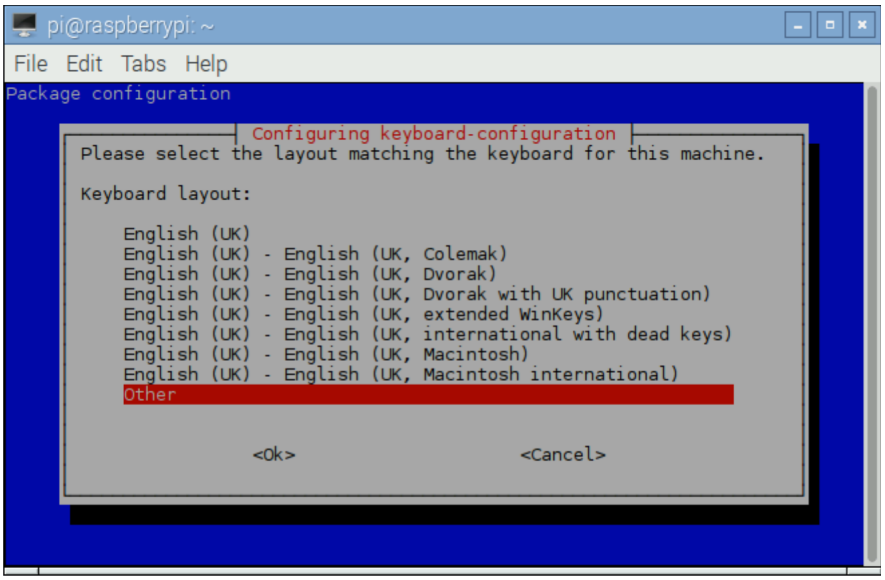
According to your language, you can choose which keyboard layout you use

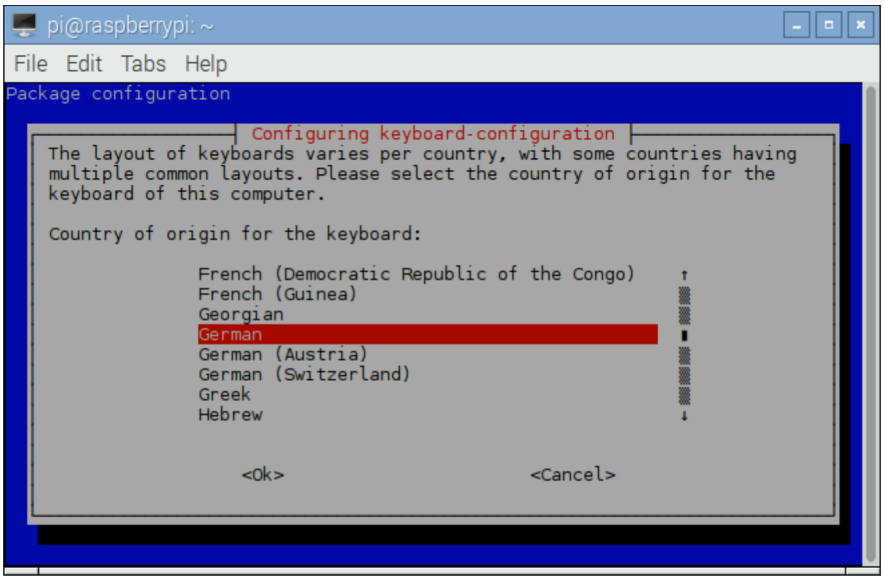
```
$sudo raspi-config
```

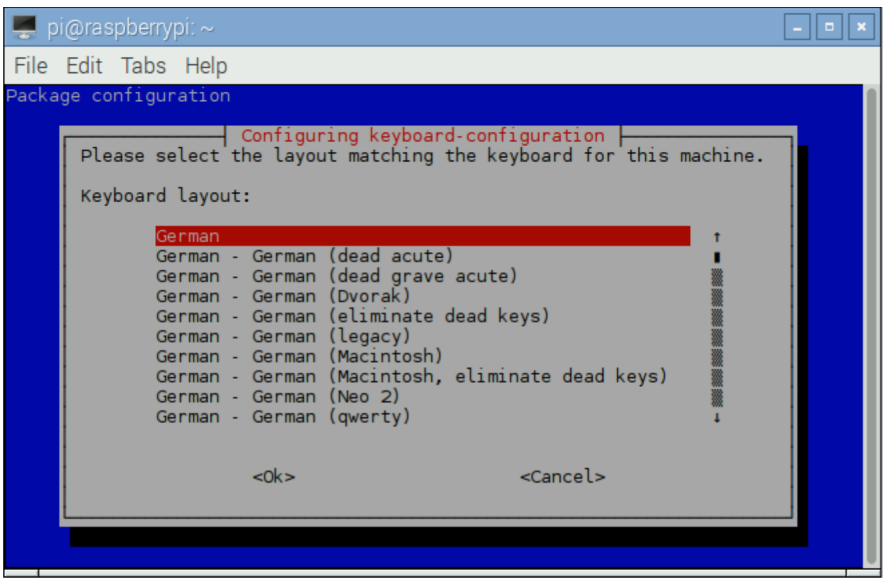
Enter through 'Localisation Options' then follow the following options as shown in shots. Important Note: in new releases of Raspbians, the following steps are not needed anymore. Keyboard layout is configured automatically.

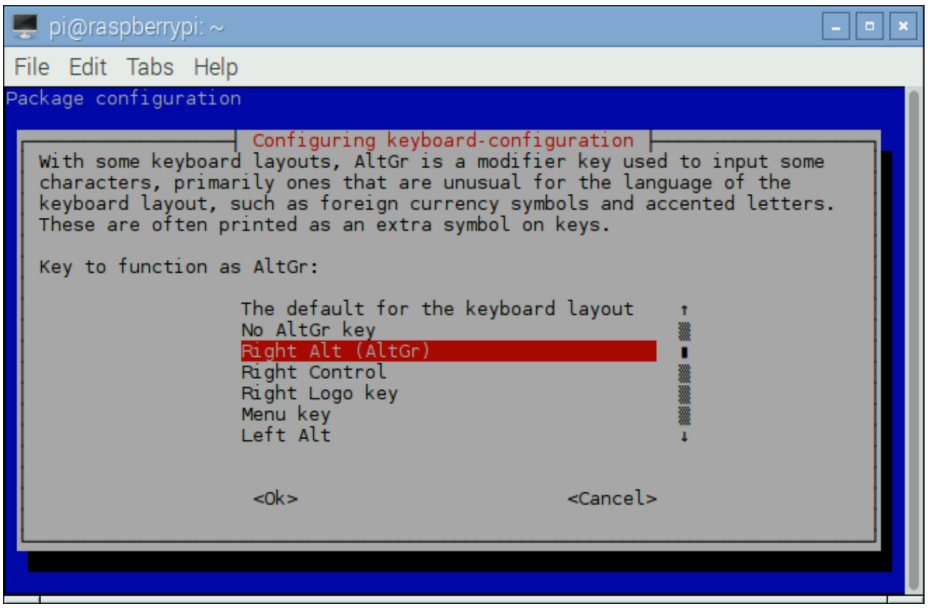


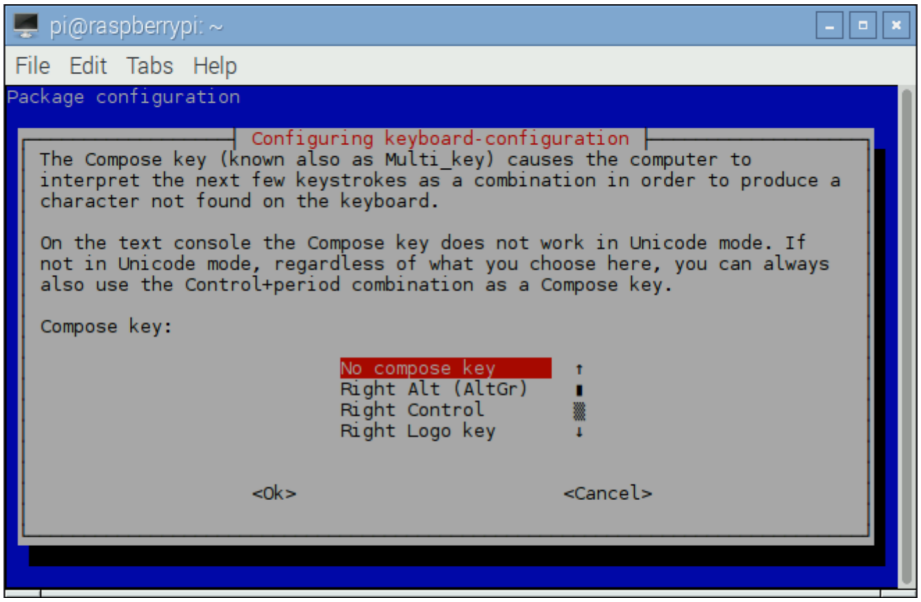



pi@raspberrypi: ~
File Edit Tabs Help
Package configuration
Configuring keyboard-configuration
Please select the layout matching the keyboard for this machine.
Keyboard layout:
English (UK)
English (UK) - English (UK, Colemak)
English (UK) - English (UK, Dvorak)
English (UK) - English (UK, Dvorak with UK punctuation)
English (UK) - English (UK, extended WinKeys)
English (UK) - English (UK, international with dead keys)
English (UK) - English (UK, Macintosh)
English (UK) - English (UK, Macintosh international)
Other
<Ok>
<Cancel>


pi@raspberrypi: ~
File Edit Tabs Help
Package configuration
Configuring keyboard-configuration
The layout of keyboards varies per country, with some countries having multiple common layouts. Please select the country of origin for the keyboard of this computer.
Country of origin for the keyboard:
French (Democratic Republic of the Congo)
French (Guinea)
Georgian
German
German (Austria)
German (Switzerland)
Greek
Hebrew
<Ok>
<Cancel>


pi@raspberrypi: ~
File Edit Tabs Help
Package configuration
Configuring keyboard-configuration
Please select the layout matching the keyboard for this machine.
Keyboard layout:
German
German - German (dead acute)
German - German (dead grave acute)
German - German (Dvorak)
German - German (eliminate dead keys)
German - German (legacy)
German - German (Macintosh)
German - German (Macintosh, eliminate dead keys)
German - German (Neo 2)
German - German (qwerty)
<Ok>
<Cancel>


pi@raspberrypi: ~
File Edit Tabs Help
Package configuration
Configuring keyboard-configuration
With some keyboard layouts, AltGr is a modifier key used to input some characters, primarily ones that are unusual for the language of the keyboard layout, such as foreign currency symbols and accented letters. These are often printed as an extra symbol on keys.
Key to function as AltGr:
The default for the keyboard layout
No AltGr key
Right Alt (AltGr)
Right Control
Right Logo key
Menu key
Left Alt
<Ok>
<Cancel>


pi@raspberrypi: ~
File Edit Tabs Help
Package configuration
Configuring keyboard-configuration
The Compose key (known also as Multi_key) causes the computer to interpret the next few keystrokes as a combination in order to produce a character not found on the keyboard.
On the text console the Compose key does not work in Unicode mode. If not in Unicode mode, regardless of what you choose here, you can always also use the Control+period combination as a Compose key.
Compose key:
No compose key
Right Alt (AltGr)
Right Control
Right Logo key
<Ok>
<Cancel>

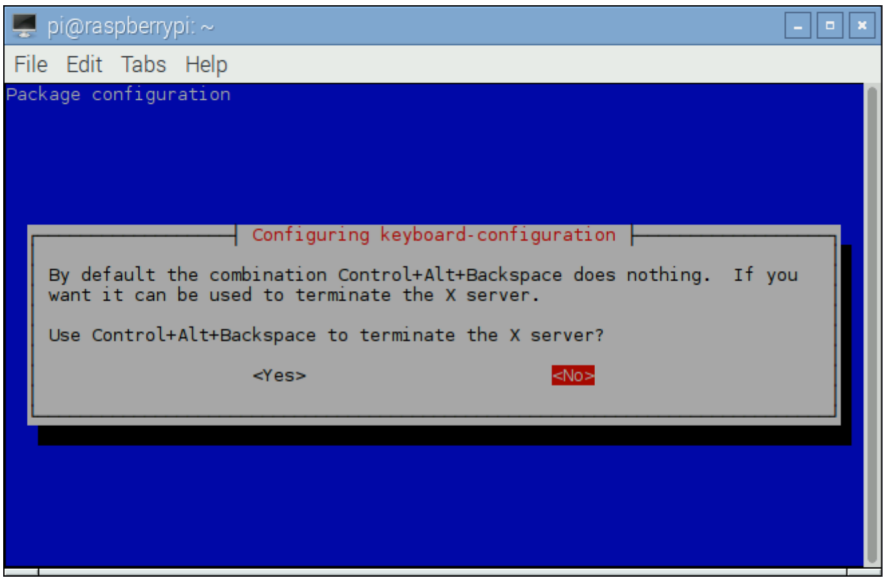


Figure 6: Keyboard Settings in Raspbian

3.3 Assign Static IP

Most of work done in Raspberry Pi, can be executed with remote connection (using SSH). It is good idea then to have fixed IP address in order to access Raspberry Pi remotely. Default setting in Raspberry Pi makes IP address is assigned by dhcp server, which means most probably you will have different IP address each time you reboot your Raspberry Pi. That is why it is important to have fixed IP address.

In order to give your Raspberry Pi static IP address, use the following to open network interfaces file

```
$sudo nano /etc/network/interfaces
```

Add the following before definition of eth0 (if it is not already exist)

```
allow-hotplug eth0
```

if you found definition of eth0 using dhcp, modify it to be as follows

```
iface eth0 inet static
```

and add the following

```
address 192.168.1.100
netmask 255.255.255.0
```

where address is the static IP address that you wish to be assigned always to your Raspberry Pi, so change it according to your preference.

The interfaces file should look like the following

```
pi@raspberrypi:~ $ cat /etc/network/interfaces
# interfaces(5) file used by ifup(8) and ifdown(8)

# Please note that this file is written to be used with dhcpcd
# For static IP, consult /etc/dhcpcd.conf and 'man dhcpcd.conf'

# Include files from /etc/network/interfaces.d:
source-directory /etc/network/interfaces.d

auto eth0
allow-hotplug eth0
iface eth0 inet static
address 192.168.1.100
netmask 255.255.255.0
gateway 192.168.1.1
```

Figure 7: Ethernet Interface Setting

3.4 Wireless Connection Setup

Starting from Raspberry Pi 3, the board comes with a wireless interface that is powerful and make the Raspberry Pi accessable through many interfaces. Enabling the Wireless interface is pretty similar to ethernet interface. So open the network interface file

```
$sudo nano /etc/network/interfaces
```

then add the following lines at the end

```
allow-hotplug wlan0
iface wlan0 inet manual
wpa-roam /etc/wpa_supplicant/wpa_supplicant.conf
iface default inet static
address 192.168.178.25
netmask 255.255.255.0
gateway 192.168.178.1
```

Also here the address and netmask and gateway depend on the address of your available wireless network. The file should look like figure 8

```
root@raspberrypi:/boot# cat /etc/network/interfaces
# interfaces(5) file used by ifup(8) and ifdown(8)

# Please note that this file is written to be used with dhcpcd
# For static IP, consult /etc/dhcpcd.conf and 'man dhcpcd.conf'

# Include files from /etc/network/interfaces.d:
source-directory /etc/network/interfaces.d

auto eth0
allow-hotplug eth0
iface eth0 inet static
address 192.168.1.100
netmask 255.255.255.0
gateway 192.168.1.1

allow-hotplug wlan0
iface wlan0 inet manual
wpa-roam /etc/wpa_supplicant/wpa_supplicant.conf
iface default inet static
address 192.168.178.25
netmask 255.255.255.0
gateway 192.168.178.1
```

Figure 8: Wireless Settings

3.4.1 Credentials

The or difference or important setting in wireless interface definition is the credentials to join the network. As shown in figure 8, the wpa_supplicant.conf is used to provide the network credentials. To add credentials of your network, open wpa_supplicant.conf

```
$sudo nano /etc/wpa_supplicant/wpa_supplicant.conf
```

At end of the file, add the following section

```
network={
    ssid="Wifi_Name"
    psk="WifiPassword"
}
```

Where “Wifi_Name” is the name of the wireless network, “WifiPassword” is the password, So they have to be changed according to your network settings.

Sometimes it is desired to have the password encrypted, this can be done using the command ’wpa_passphrase’. To generate encrypted psk use the following

```
$wpa_passphrase "Wifi_Name"
```

You will be asked for the password. After providing it, you will have an encrypted psk. Just copy the new encrypted psk instead of the un-encrypted one

```
network={
    ssid="Wifi_Name"
    psk=250648005ff2fac9182665c004de85300f9c2bb7b9f1de0bf840cdfe
}
```

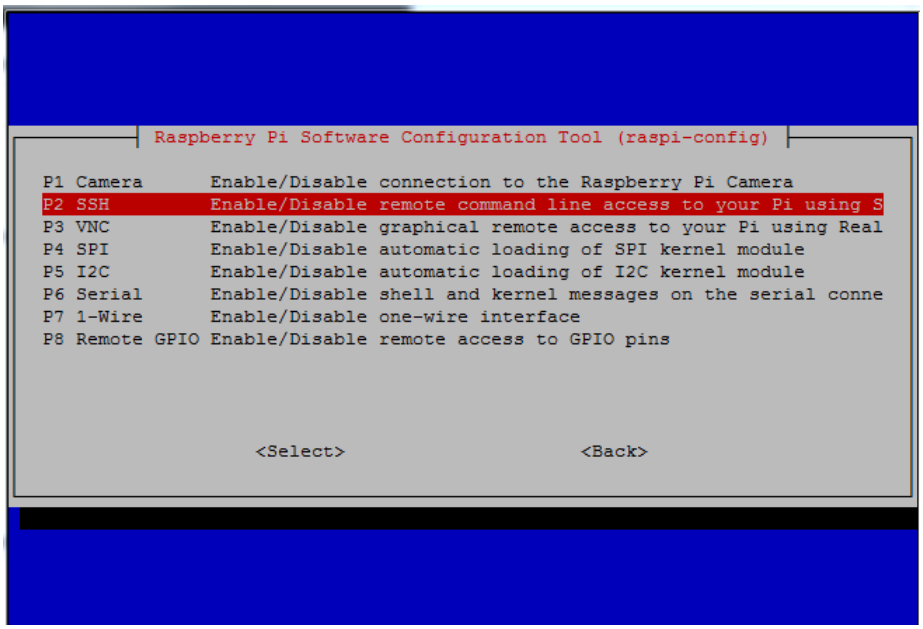
Ctrl+O then Ctrl+X to save your modifications in wpa_supplicant.conf and exit.

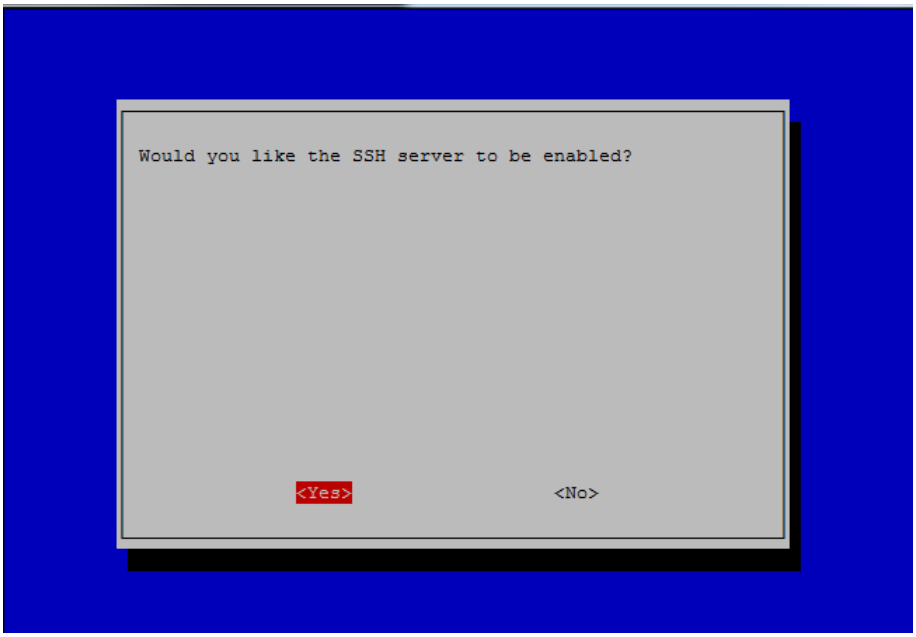
3.5 Enable SSH

If you didn’t enable ssh at beginnig, now it is time to enable the SSH service in order to access the Raspberry Pi remotely.

```
$sudo raspi-config
```

choose “Interfaceing Options”, the following panel should appear. Choose to enable SSH.


Raspberry Pi Software Configuration Tool (raspi-config)
P1 Camera Enable/Disable connection to the Raspberry Pi Camera
P2 SSH Enable/Disable remote command line access to your Pi using S
P3 VNC Enable/Disable graphical remote access to your Pi using Real
P4 SPI Enable/Disable automatic loading of SPI kernel module
P5 I2C Enable/Disable automatic loading of I2C kernel module
P6 Serial Enable/Disable shell and kernel messages on the serial conne
P7 1-Wire Enable/Disable one-wire interface
P8 Remote GPIO Enable/Disable remote access to GPIO pins
<Select>
<Back>


Would you like the SSH server to be enabled?
<Yes>
<No>

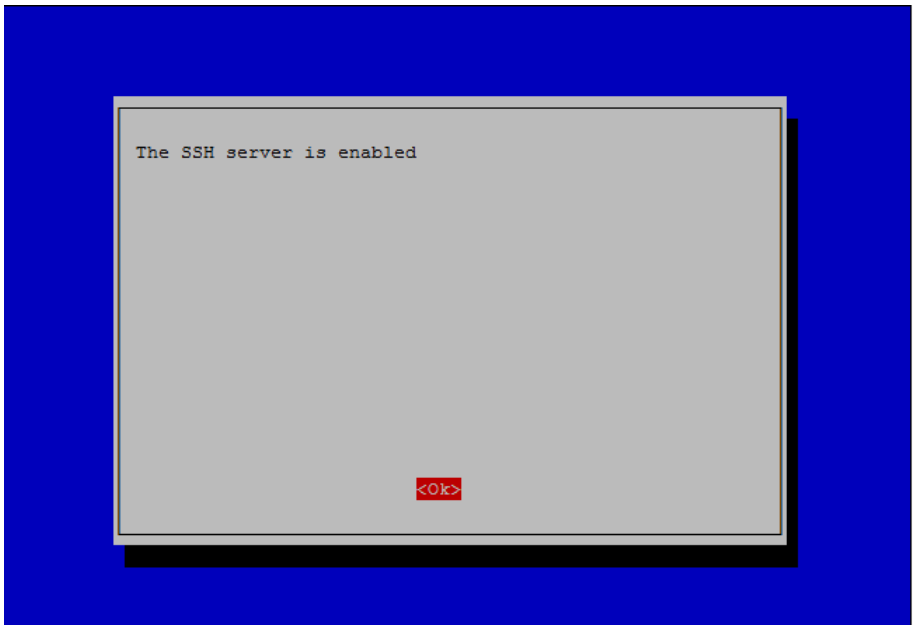


Figure 9: SSH Setting

3.6 Other Packages

Some other installations are advised to be performed in order to prepare Raspbian for further application to be set. The following packages can be installed once you have an active internet connection to Raspbian.

```
$sudo apt-get install build-essential
$sudo apt-get install htop
$sudo apt-get install git
$sudo apt-get install sysbench
```

4 Remote Connection

All previous instructions assumed that user is interacting with the Raspbian directly by connecting a Monitor to Raspberry Pi to command the Raspbian directly. This is not necessary as you can work with Raspberry Pi remotely by means of SSH. The point here is SSH has to be enabled in Raspbian, this has been discussed in section 2. So when you install Raspbian in the SD card, you have to add empty file called 'ssh' in the created 'boot' partition.

So all what you need to do is to install Raspbian to the SD card, connect the Raspberry Pi then into your network. Raspberry Pi will be assigned with IP address. This IP address is what we will use to connect using SSH. The following steps assume that you need to connect to Raspbian from another Linux machine.

In order to know this IP address, you can you use nmap package in your local machine that will connect to Raspberry Pi remotely. Install this package if it is not installed by

```
$sudo apt-get install nmap
```

Then to figure out what it is the assigned IP for the Raspberry Pi, execute this command

```
$nmap -sn -A 192.168.1.*
```

where 192.168.147 is the first section of your network IPs. This command will list all connected devices in the network along with its assigned IP. Have a look then on these devices names until you find Raspberry Pi device, capture its IP then.

You can connect to Raspberry Pi then by

```
$ssh pi@192.168.1.xxx
```

Replace xxx with the number you have found from nmap operation. For first time connection, your local machine will ask for authentication of connection and adding ECDSA key. Respond with yes to proceed with the connection. You can now work with Raspberry Pi remotely and perform any possible operation.

In some cases, If you have more than Raspberry Pi board, you may want to connect your prepared SD card to other Raspberry Pi board. If you want to connect remotely then, your local machine will notice that the connected device has been changed and the saved ECDSA key is not matching with the detected device. You can't connect remotely then if you got this warning message.

To solve this point, you can delete the old ECDSA key and establish new one. You can do that simply by

```
$ssh-keygen -R 192.168.1.xxx
```

then connect with ssh again to establish new ECDSA key.

In case of Windows operating system, nmap can be installed with a gui, it is free and downloadable from the internet.

5 BackUp Operations

To make backup of your current installed Raspbian

```
df -h
```

this is to know what is the driver of your SD Card. Assume it is in /dev/sdb1 & /dev/sdb2

```
$sudo dd if=/dev/sdb of=/path/to/backup/location bs=4M
```

To reinstall this back up again the same dd command shall be used.

If it happened that your image bigger than available SD Card, then you can use the following script to resize the image to fit into the card. This script is taken from

http://sirlagz.net/2013/03/10/script-automatic-rpi-image-downsizer/

```
#!/bin/bash
# Automatic Image file resizer
# Written by SirLagz

strImgFile=$1
if [[ ! $(whoami) =~ "root" ]]; then
  echo ""
```

```
  echo "**********************************"
  echo "*** This should be run as root ***"
  echo "**********************************"
  echo ""
  exit
fi

if [[ -z $1 ]]; then
  echo "Usage: ./autosizer.sh "
  exit
fi

if [[ ! -e $1 || ! $(file $1) =~ "x86" ]]; then
  echo "Error : Not an image file, or file doesn't exist"
  exit
fi

partinfo=`parted -m $1 unit B print`
partnumber=`echo "$partinfo" | grep ext4 | awk -F: ' { print $1 } '`
partstart=`echo "$partinfo" | grep ext4 | awk -F: ' { print
substr($2,0,length($2)-1) } '`
loopback=`losetup -f --show -o $partstart $1`
e2fsck -f $loopback
minsize=`resize2fs -P $loopback | awk -F': ' ' { print $2 } '`
minsize=`echo $minsize+1000 | bc`
resize2fs -p $loopback $minsize
sleep 1
losetup -d $loopback
partnewsize=`echo "$minsize * 4096" | bc`
newpartend=`echo "$partstart + $partnewsize" | bc`
part1=`parted $1 rm 2`
part2=`parted $1 unit B mkpart primary $partstart $newpartend`
endresult=`parted -m $1 unit B print free | tail -1 | awk -F: ' { print
substr($2,0,length($2)-1) } '`
truncate -s $endresult $1
```

Bibliography

[1] https://www.raspberrypi.org/

[2] http://sirlagz.net/2013/03/10/script-automatic-rpi-image-downsizer/