# Introductie en configuratie Raspberry pi

## Inleiding

De raspberry pi is een volledige PC met de afmetingen van een creditcard. De pi heeft een ARM processor. Er bestaan verschillende types van model A, model B, model B+ en de raspberry pi 2 model B. De belangrijkste verschillen zijn de aansluit mogelijkheden (26 of 40 polige connector) en de processor. Ik raad zeker de raspberry pi 2 omdat die ook windows 10 IoT Core kan draaien. Sinds maart ’16 kan je ook de raspberry pi 3 aankopen. De pi 3 is ietske sneller dan de pi 2, maar zijn grootste voordeel is dat hij bluetooth 4.0 of bluetooth low energy en wifi aan boord heeft.

Momenteel heb je ook de pi zero die je kan aankopen voor 5$. Helaas staat die nog steeds als “sold out”

<https://www.raspberrypi.org/>

Op de pi kan je dus verschillende OS laten draaien, een database installeren, webserver configureren, …

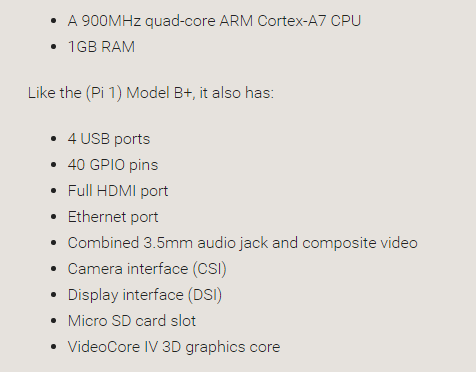
Een arduino daarentegen is een microcontroller met een Atmega chip, die je in een soort C-syntax kan programmeren. Het voordeel van een arduino is, dat er heel wat shields bestaan die je gewoon op de arduino kan inpluggen. De arduino beschikt ook over heel wat analoge in-en uitgangen.

Bij een groter project kan het interessant zijn om een pi samen te laten werken met een aantal arduino’s, de arduino’s verzorgen dan de sturing, terwijl de pi zorgt voor de back-up, database, webserver, …

Opmerking:

Bovenstaand materiaal wordt vooral gebruikt als prototyping. Heb je een bedrijfszekere oplossing nodig dan zou ik toch eerder voor een plc oplossing opteren. Een plc is een veel robuuster apparaat. De in-en uitgangen zijn ook veel beter beveiligd. Een mooi compromis is b.v. de Controllino

## Raspberry pi board

Belangrijk hierbij zijn de volgende opmerkingen :

De pi heeft GEEN spanningsregelaar. Dus enkel gebruik maken van voedingen die exact 5V genereren ofwel een usb poort van de laptop gebruiken.

De GPIO pins (general pupose input output), hiermee kan je allehande sensoren en actuatoren aansluiten. Heel wat embedded devices beschikken over deze pinnen. Standaard heeft de pi libraries geïnstalleerd om met deze pinnen te werken, zowel in C of Python.

Let op : De GPIO pinnen zijn zeer kwetsbaar!!!

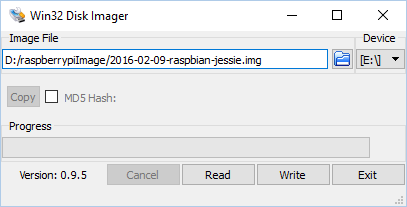
## Installatie van de sd card.

De makkelijkste manier om de pi aan de praat te krijgen is de installatie van een NOOBS (new out of the box software, of zo iets). Via noobs kan je makkelijk verschillende os installeren. O.a. Raspbian, Pidora, OpenElec. Via NOOBS heb je wel een complexere mappenstructuur en verloopt de configuratie wat moeilijker. Je kan immers verschillende OS staan hebben.

Vandaar dat ik verkies om enkel Raspbian als OS te installeren.

Op de volgende pagina kan je heel wat verschillende os selecteren.

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

Druk op de button Raspbian. Nu hebben we de keuze tussen Raspbian Jessie, Raspbian Jessie Lite en Raspbian Wheezy. Er wordt het meest met Raspbian Wheezy gewerkt en daar vind je ook de meeste info van terug. De Jessie is de nieuwere versie.

Download Jessie en pak het zipbestand uit.

Met b.v. win32 disk imager kan je image op de sd card plaatsen.

Tip : Win32 disk imager kan je ook gebruiken om een backup van je image te maken.

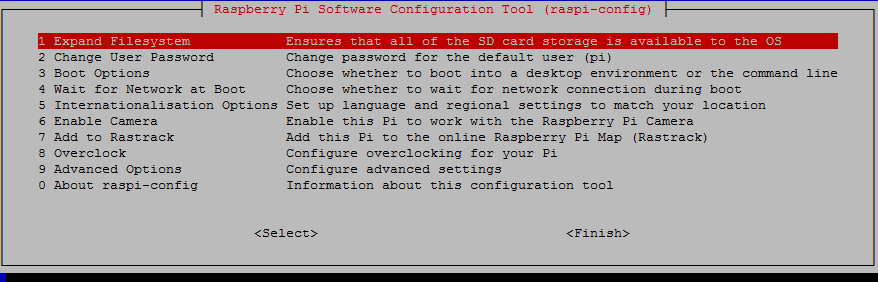
<http://sourceforge.net/projects/win32diskimager/>

## De pi configureren

Na het opstarten van de pi krijgen we een configuratie scherm te zien. Belangrijk hier is om het keyboard in te stellen. Eventueel kan je de default login en het paswoord nog wijzigen. (pi en paswoord raspberry) Dit gaan de studenten zeker moeten aanpassen.

Later kan je dit scherm natuurlijk ook nog aanpassen.





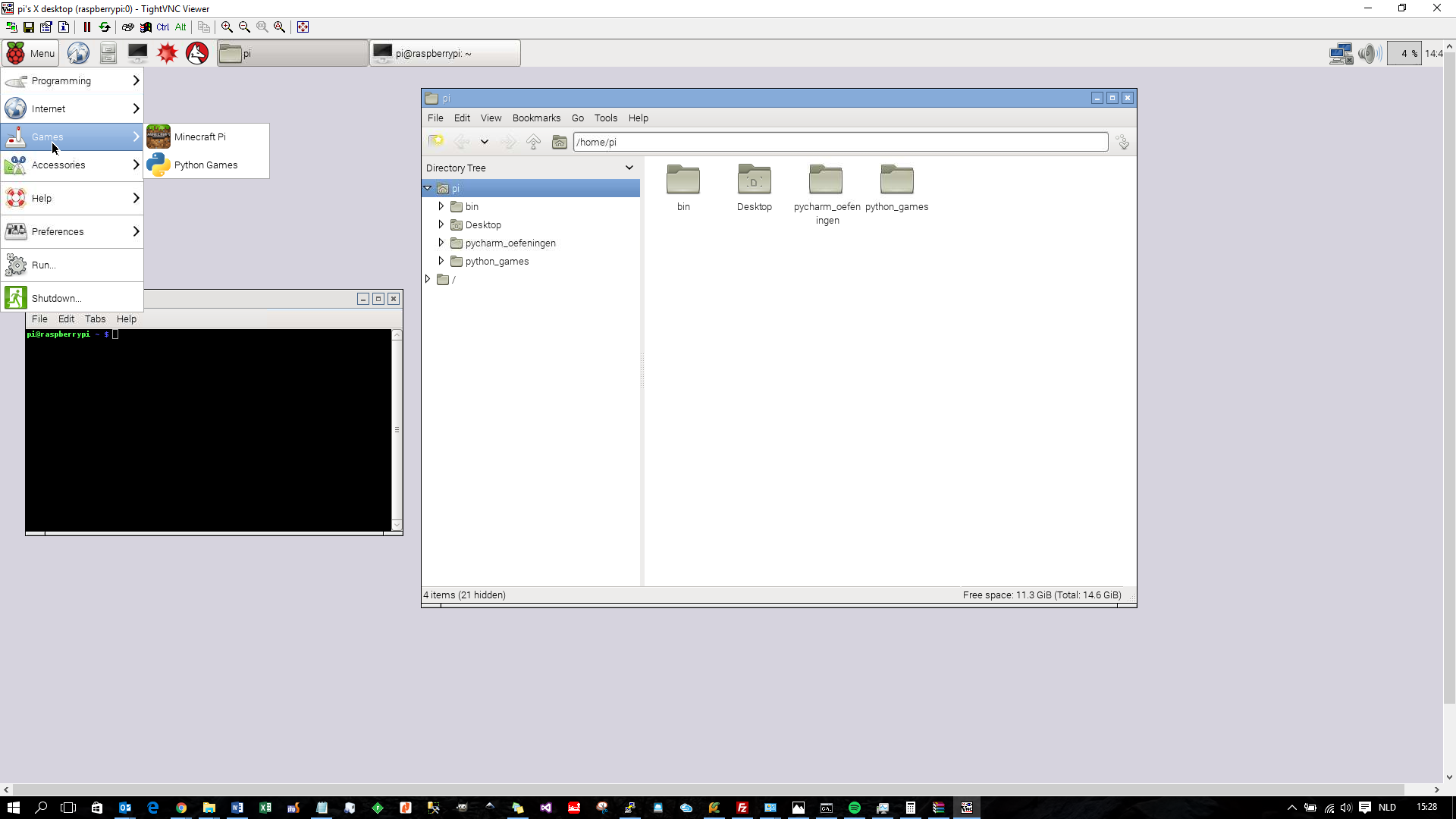
Belangrijk is keuze 1 : expand filesystem

Op die manier wordt de sd card volledig benut.

Via “advanced options” kan je je de hostname wijzigen en de verschillende bussen enablen. Die bussen kan je nodig hebben om met verschillende sensoren te communiceren.

Tik startx om de desktopomgeving op de starten.

$startx



Ga naar de command prompt

Er komen verschillende versie en updates van de Wheezy of de Jessie uit, m.b.v. de volgende commando’s kunnen we met de laatste versie werken.

$ sudo apt-get update //update van de system package list

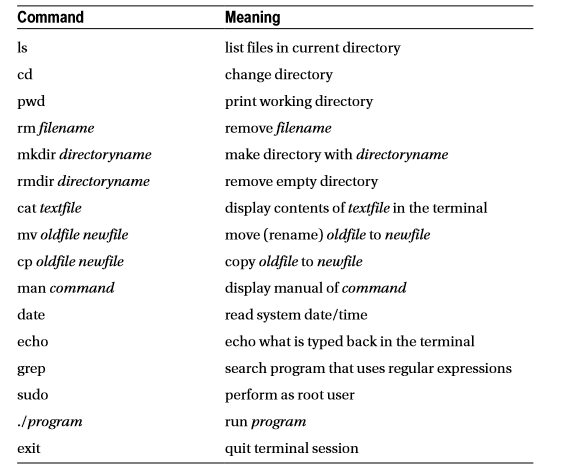
$ sudo apt-get upgrade //upgrade van de geïnstalleerde package list

## Linux

Het linux file system is opgebouwd rond de root folder, voorgesteld door een /.

Enkel een root user is gemachtigd over alle files. Een linux gebruiker zal nooit inloggen als een root user. Een ervaren linux gebruiker maakt gebruik van sudo (super user do) om ‘administror’ commands uit te voeren.

Enkele veel gebruikte commando’s zijn :



cd Returns you to your login directory

cd ~ Returns you to your login directory

cd / Takes you to the entire systems root directory

cd .. Moves you up one directory

## Bestanden **(bron wikipedia)**

<https://nl.wikibooks.org/wiki/Linux_voor_beginners/De_terminal>

Om er voor te zorgen dat verschillende gebruikers niet in de bestanden van andere gebruikers gaan rommelen of zelfs in de systeembestanden gaan prutsen, heeft iedere gebruiker per bestand rechten om dat bestand te bekijken, aan te passen of uit te voeren. Dit komt er op neer dat ieder bestand een eigenaar heeft, deze eigenaar heeft volledige controle over het bestand. Voor ieder bestand wordt er ook een groep bijgehouden. De gebruikers die in de groep van het bestand zitten hebben meer rechten om het te gebruiken.

Om te kijken wat de rechten van een bepaald bestand zijn gebruik je het commando ls -l. We geven een voorbeeld:

$ ls -l Bestand1

-rw-r----- 1 ik mijnGroep 9 2010-02-02 10:18 Bestand1

^^^^^^^^^^

||||||||||

|||||||**\\\-**---------- andere gebruikers

|||||||

||||**\\\-**------------- groep

||||

|**\\\-**---------------- eigenaar

|

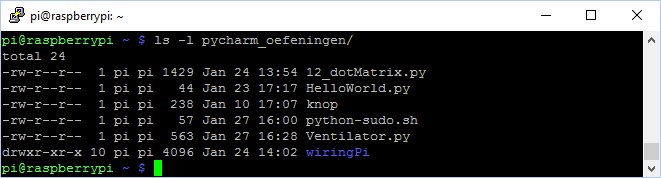
**\-**------------------- bestand = "-" (mogelijk: -, d, l, b, c, q)

Je ziet dus wie er eigenaar is, in dit geval de gebruiker met de naam *ik*. En de groep *mijnGroep* is de groep die het bestand kan controleren. Om te zien wie welke rechten heeft kijk je naar de eerste code:

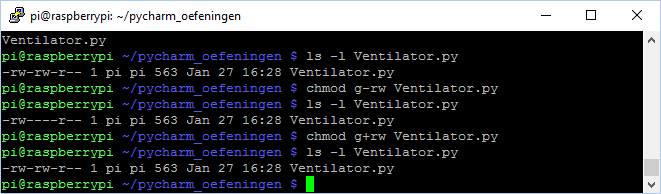
* Als er een - als eerste teken staat, dan is het een gewoon bestand. Voor directory's wordt een d gebruikt en voor links een l. Er zijn ook nog meer exotische letters.
* Na de eerste letter volgen drie blokjes met elk drie tekens. Het eerste blokje, rw- in ons geval, zijn de rechten die aan de eigenaar gegeven worden. Het tweede blokje, r-- in ons geval, zijn de rechten die aan de controlerende groep gegeven worden en het laatste blokje --- zijn de rechten voor iedere gebruiker.

Zo'n blokje rechten heeft iedere keer een gelijkaardige structuur. Er worden drie letters gebruikt: de r staat voor leesrecht, de w staat voor schrijfrecht en de x staat voor het recht om het bestand uit te voeren. Als de gebruiker waarvoor het blokje telt de letter heeft, dan heeft hij dat recht op het bestand. Als de gebruiker waarvoor het blokje telt die letter niet heeft maar een - heeft in plaats van die letter, dan heeft de gebruiker dat recht niet.

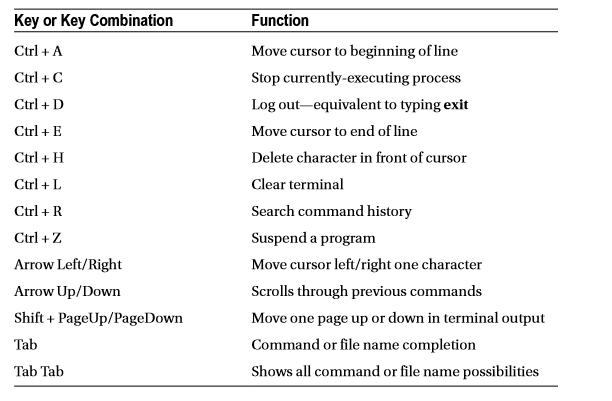
Bij het voorbeeld zien we dus dat *ik* het bestand mag bewerken en bekijken (rw-). De gebruikers die in de groep *mijnGroep* zitten mogen het bestand bekijken (r--) en andere gebruikers mogen niets (---).



Met het commando chmod kan je de permissions van een file aanpassen.



## Shells in linux



## lscPackage managers

Een veel gebruikte manager is de apt packagemanager. Dit is een Advance Packaging Tool van Debian. In de terminal kunnen we dan b.v. het volgende commando ingeven :

sudo apt-get install package name

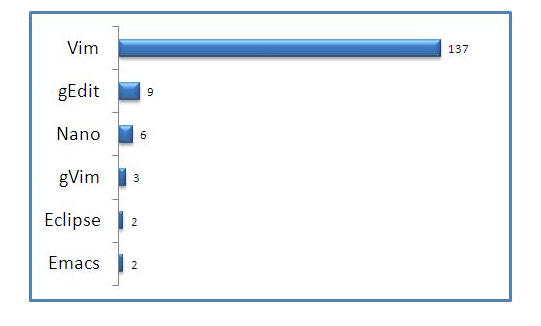
of

sudo apt-get remove package name

Via sudo reboot of

Sudo shutdown –h (shutdown halt)

## Text editors



Een echte linux gebruiker werkt blijkbaar met Vi of Vim.

Ik verkies echter nano. Deze staat standard geïnstalleerd op raspbian. Er staan ook enkele shortcuts vermeld in nano. Vi is waarschijnlijk wel handig, maar dan moet je deze tool dikwijls gebruiken. In nano kunnen we dus b.v. onze eerste python script schrijven. Ik vind het werken in nano niet zo aangenaam, vandaar dat ik verkies om aan remote programming te doen. Dit kan b.v. in pycharm, netbeans of via een plugin van visual studio. (deze is wel betalend).

Om een file aan te maken in nano kan je b.v. tikken :

nano testfile.txt



## Python

Python is een script taal, dus is iets trager dan C. Meestal wordt in python gewerkt, is snelheid echt een issue kan je het probleem in C oplossen.

Wij gaan natuurlijk van start met het verplichte “Hello world”.

Standaard is python geïnstalleerd in Raspbian.

$ cd / toont ons de root van het systeem

In de map /usr/local/lib vind je de python libraries terug. Momenteel zijn de python 2.7 en de 3.2 geïnstalleerd.

Tik :

print (“Hello world!)

en sla de file Hello.py op.

Vervolgens kan tik je :

python Hello.py

Ons eerste script is uitgevoerd!!!

