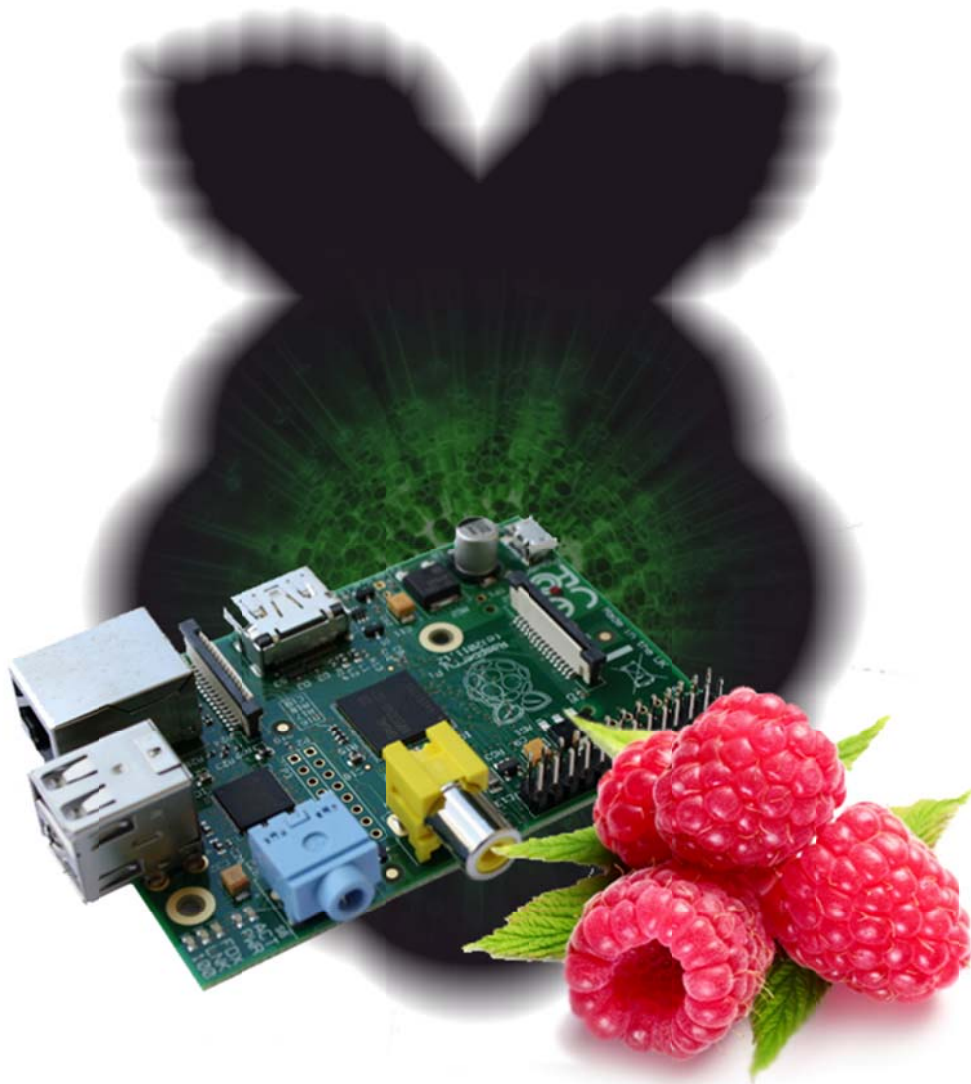


# **Raspberry Pi**

**un nou concepte de mini-ordinador**



Pol Serra Bertran

2n BTX A

Gener 2014

Tutor: Jesús Arbués

"Innovation distinguishes between a leader and a follower"

"Creativity is just connecting things. When you ask creative people how they did something, they feel a little guilty because they didn't really do it, they just saw something. It seemed obvious to them after a while"

Steve Jobs -cofundador de Apple-

# Índex

<b>Introducció .....</b>	<b>6</b>
<b>Objectiu de Recerca .....</b>	<b>8</b>
<b>Metodologia.....</b>	<b>9</b>
<b>1. La Raspberry Pi.....</b>	<b>10</b>
1.1: Què és la Raspberry Pi ? .....	10
1.2: Com és la Raspberry Pi ? .....	10
1.3: Adquirir una Raspberry Pi.....	13
<b>2. Primers Passos .....</b>	<b>14</b>
2.1: El sistema operatiu .....	14
2.1.1: Què és el sistema operatiu?.....	14
2.1.2: Instal·lació del sistema operatiu Raspbian .....	15
2.2: Posada en marxa.....	17
2.2.1: Configuració inicial – El configurador raspi-config.....	17
2.2.2: Configuració inicial .....	18
2.2.3: Començant a utilitzar la Raspberry Pi .....	19
2.3: Funcionament i comandaments .....	21
2.3.1: Comandaments Útils .....	21
2.3.2: Modificació d'un arxiu de text .....	21
2.3.3: Instal·lació d'un programa .....	22
2.4: Connexió escriptori remot (SSH) .....	23
2.4.1: Instal·lació de l'xrdp.....	23
2.4.2: Connexió SSH amb cable ethernet d'ordinador a ordinador .....	23
2.4.3: Connexió SSH a través de LAN (cable i wifi) .....	26
2.4.4: Connexió SSH a través d'internet (des de qualsevol lloc del món) .....	27
2.4.5: Connexió SSH a través del mòbil (amb el VNC) .....	31

<b>3. Utilitats de la Raspberry .....</b>	<b>33</b>
3.1: Programes Interessants:.....	33
3.1.1: Minecraft Pi .....	33
3.1.2: Scrot.....	33
3.1.3: VLC Player .....	34
3.1.4: LibreOffice .....	34
3.1.5: Buscar altres programes .....	34
3.2: Utilitats com a Sistema Operatiu:.....	35
3.2.1: OpenELEC .....	35
3.2.2: RaspBMC .....	35
3.2.3: PiMAME .....	35
<b>4. Els pins GPIO de la Raspberry .....</b>	<b>36</b>
4.1: Què són els pins GPIO ? .....	36
4.2: Com funcionen els pins GPIO?.....	38
4.3: Interactuant amb el GPIO de la línia de comandaments (LXTerminal) .....	40
4.3.1: Llista de Comandaments.....	40
4.3.2: Comprovació dels comandaments .....	41
4.4: Programació dels GPIO .....	42
4.5: El QT Creator .....	43
4.5.1: Per què el QT Creator? .....	43
4.5.2: Instal·lació del QT Creator a la Raspberry Pi .....	43
4.5.3: Configuració inicial del QT Creator.....	44
4.5.4: L'entorn de QT Creator .....	46
4.5.5: Començant un programa.....	48
4.5.6: La programació .....	49
4.5.6.1: Elements en el full de la programació:.....	49
4.5.6.2: Tipus de variables: Tipus més freqüents .....	50
4.5.6.3: Declaració de variables.....	50
4.5.6.4: Funcions i utilitats .....	51
4.5.6.5: Funcions i utilitats lligades a la interfície gràfica .....	52

4.6: Aplicació del control dels GPIO.....	54
4.6.1: Simulació d'un semàfor .....	54
4.6.2: Joc de reacció amb LEDs .....	58
<b>Conclusions .....</b>	<b>62</b>
<b>Webgrafia .....</b>	<b>64</b>

# Introducció

Sóc un de un alumne de 2n de Batxillerat de l'institut Vilatzara i que curso l'itinerari tecnològic. Inicialment no tenia gens clar quin treball fer, dubtava entre diferents àrees i possibilitats però mai vaig aconseguir que se me n'acudís un que em deixés convençut. Per tant, vaig decidir esperar que sortís la llista de treballs proposats o que algun professor en proposés un.

Abans d'acabar 1r de Batxillerat el professor de Plàstica , ara també tutor del treball, em va comentar que estaria bé fer alguna cosa amb una Raspberry Pi, ja que es una cosa bastant nova. I allà es va quedar la cosa, va passar l'estiu i en arribar el setembre sense haver triat el tema del treball, i sense alternatives a la Raspberry, va quedar decidit. Tot i això, cal dir que el tema em semblava interessant perquè tractava temes de tecnològics informàtics.

Ja només faltava concretar què fer amb la Raspberry, però en començar a investigar vaig descobrir que a internet hi havia força desordre en els tutorials, i per tant, vaig trobar interessant muntar un tutorial propi per deixar la base ben muntada per si algú en el futur ho podia aprofitar per estalviar-se tota aquesta feina prèvia.

## Resum del treball

Aquest treball és un tutorial sobre la Raspberry Pi, començant dels elements més bàsics, passant per utilitats i acabant amb la programació dels pins de la Raspberry amb un parell de senzilles pràctiques.

El cos del treball l'he dividit en quatre parts. La primera part és una introducció al concepte de la Raspberry Pi i quines característiques te. La segona part és un tutorial de com posar en marxa la Raspberry per primer cop, així com els elements i programes necessaris i els passos que s'han de seguir. També inclou una àmplia part on s'explica l'accés a la raspberry des d'un altre ordinador o mòbil. La tercera part és un recull de utilitats, programes i modes d'utilització que pot tenir la raspberry

Pi, ja siguin sistemes operatius emuladors de consoles o simples programes per a la raspberry. L'última part està enfocada a la programació dels pins GPIO de la Raspberry a través del Qt Creator, així que també inclou uns quants conceptes i elements de programació en c++.

# **Objectiu de Recerca**

L'objectiu de recerca que té aquest treball és aconseguir el control dels pins GPIO de la Raspberry Pi i fer-ne una aplicació pràctica.

## **Objectius Generals**

Aquest treball té com a objectius crear un tutorial que pretén estendre el concepte de la Raspberry pi i fer-la més accessible a tothom. El tutorial haurà de explicar els aspectes bàsics, com fer-la servir, i aquelles utilitats pot tenir. També caldrà deixar ben clar el procediment en la programació del GPIO, perquè es pugui aprofitar posteriorment per fer una aplicació real i pràctica utilitzant la Raspberry.



# Metodologia

La metodologia que s'ha fet servir en aquest treball és la cerca d'informació a Internet i comprovar si funciona tot tal com posa i si encara és vàlid ( com que una bona part depèn de la informàtica, això està subjecte a actualitzacions, revisions i canvis constants.

En el cas que la informació trobada a internet sigui veraç i funcioni, llavors he procedit a redactar el procés que he fet servir jo de la millor manera explicada, perquè sigui comprensible per tothom i evitar un dels grans problemes que es té normalment quan busques per internet: els tutorials i respostes sobreentenen moltes coses, i si no saps del que estàs parlant, no s'entén res. L'objectiu és que es pugui seguir perfectament les instruccions quan t'hi trobes al davant.

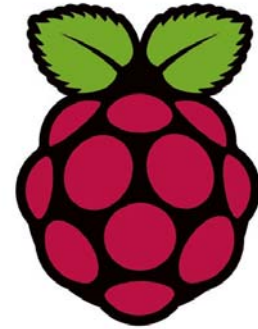
Una bona eina que he fet servir a l'hora de redactar el tutorial són les captures de pantalla, que ajuden a situar-se i que la explicació no sigui tan densa. Hi ha captures de pantalla tant del meu ordinador com de la Raspberry.

# La Raspberry Pi

## 1.1: Què és la Raspberry Pi ?

La Raspberry Pi és un ordinador d'una sola placa ( tots els seus components es troben en una placa ), i d'una mida de 65x95mm, una mica més gran que una targeta de crèdit (55x85 mm).

Va ser desenvolupat per la fundació Raspberry Pi com a un ordinador de baix cost i amb la finalitat educativa d'impulsar la programació informàtica a escoles i instituts.



Logotip de la Raspberry Pi

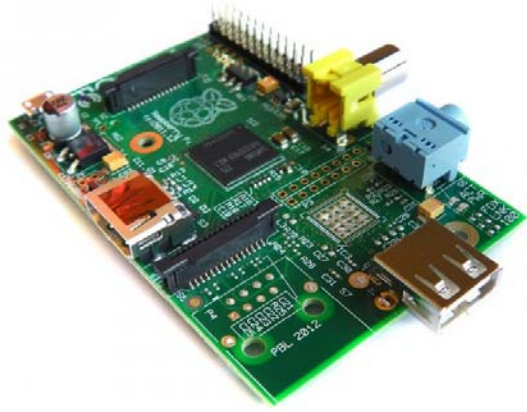
## 1.2: Com és la Raspberry Pi ?

Existeixen dos models de la Raspberry Pi, el model A i B. El model B és més complet que l'A, ja que té més connectors i més prestacions.

A continuació la taula que compara els dos models,

<b>Característica</b>	<b>Model A</b>	<b>Model B</b>
Preu	25€	32.5€~35€
Processador	ARM (..) de 700 MHz	
T. Gràfica	Broadcom Videocore IV: té OpenGL i compatible amb alta definició.	
Memòria (RAM) a la targeta SD	256 MiB *	512 MiB *
Ports USB 2.0	1	2
Sortida d'àudio	HDMI i jack	
Sortida de vídeo	HDMI i RCA	
Emmagatzematge	Espai per targeta SD	
Internet	No en té	Connector ethernet
Alimentació	5V 500mA via MicroUSB	5V 700mA MicroUSB

\* MiB: Mebibyte, és igual a  $2^{20}$  bytes, 1048576 bytes



Model A

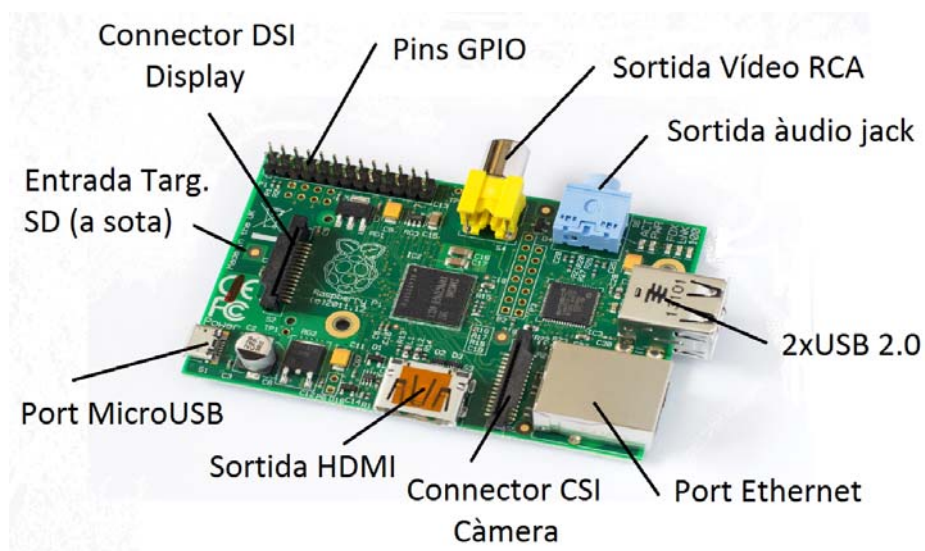


Model B

A partir d'ara ens centrarem en el model B, que és el que he fet servir.

### Components de la Raspberry Pi (B)

- Port MicroUSB: Alimentació de la Raspberry,
- Sortida HDMI: Sortida de vídeo i àudio
- Sortida RCA: Sortida de vídeo
- Sortida jack (3,5mm): Sortida d'àudio
- 2 USB 2.0: Per connectar-hi accessoris: ratolí, teclat, USB (de dades), adaptador wifi, etc.. com els d'un ordinador qualsevol
- Port ethernet: Connexió a internet via cable o a un altre ordinador
- GPIO: Pins per a la connexió de circuits electrònics, d'entrada i sortida.
- Connector DSI: Aquí hi podem connectar un petit display
- Connector CSI: Aquí s'hi pot connectar una càmera



## Hardware necessari:

Per poder posar la Raspberry Pi en marxa necessitem obligatòriament els següents elements:



- Font d'alimentació microUSB que ens doni 5V i entre 700mA i 1A de corrent continu, el més normal és fer servir un carregador d'un smartphone, que normalment porten aquest tipus de connector, en el que el voltatge i la intensitat són adequats. Tot i això, comproveu que sigui així abans de provar-ho per evitar danys a la Raspberry, ja que amb una intensitat menor pot ser que falli quan s'hi connectin altres coses. La Raspberry Pi no té cap interruptor encès/apagat, si el cable de la corrent esta connectat estarà encesa, i si no, apagada. Podem saber quan esta encesa o apagada per un llumet vermell que hi ha entre els USB i la sortida jack.
- L'altre element obligatori és una targeta SD de com a mínim 4GB. La Raspberry Pi no porta disc dur ni cap tipus de sistema d'emmagatzematge, i per tant, el sistema operatiu, els programes i els fitxers s'han de trobar a la targeta SD quan es connecta la corrent.

Amb tot això ja podrem encendre la Raspberry Pi, però per poder-la fer servir necessitarem un ratolí, un teclat i una pantalla on poder-hi connectar l'HDMI o l'RCA. Si no disposem d'una pantalla amb aquestes entrades haurem d'utilitzar un aparell de televisió, per incòmode que sigui, això ja ho arreglarem més endavant.

## 1.3: Adquirir una Raspberry Pi

Podem comprar una Raspberry Pi a una botiga d'electrònica, però l'empresa oficial distribuïdora de Rasperrys a Europa és RSComponents. Podem comprar-ne una a través d'internet a la web:

<http://raspberrypi.rsdelivers.com/>.



Cal tenir en compte que els preus de la Raspberry Pi oficials són sense IVA (o VAT, ja que la RSComponents és anglesa) de manera que s'ha d'afegir un 20% al preu.

A banda de la Raspberry, a la web hi ha packs on ja porta una SD, i també accessoris i complements, com una càmera, capsas protectores, expansions, ratolins i teclats, adaptadors wifi, cables, etc.

Suggereixo comprar una capsa protectora perquè a part de que queda millor, protegirà la Raspberry contra possibles accidents com ara la caiguda. També esta bé tenir un adaptador wifi per poder connectar la Raspberry a internet sense necessitat de cable.



Raspberry Pi en una capsa protectora



Adaptador wifi USB

# Primers Passos

A continuació s'expliquen els primers passos en una Raspberry Pi, l'encendre-la per primer cop, posar-hi el sistema operatiu i allò més essencial per començar a fer-la servir.

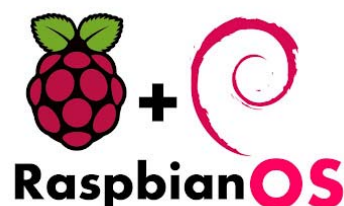
També hi ha un apartat important en fer servir una Raspberry Pi, que és fer-la anar des d'un altre ordinador o fins i tot des del telèfon mòbil, cosa que és molt més còmode ja que no hi has de connectar ni teclat, ni ratolí, ni pantalla, a més a més de deixar els ports USB lliures.

## 2.1: El sistema operatiu

### 2.1.1: Què és el sistema operatiu?

El sistema operatiu (SO / OS) és el conjunt de programes (software) que controlen i coordinen les funcions bàsiques d'un ordinador (com el processador, les targetes, i els altres programes) i els processos d'encesca i apagada, instal·lacions i transferències de dades. Els sistemes operatius es troben en aparells com telèfons mòbils, consoles de videojocs i evidentment en ordinadors, a més a més d'altres aparells.

Per tant, la Raspberry Pi, com a bon ordinador que és, ha de tenir el seu sistema operatiu. La Raspberry Pi esta pensada per anar amb un sistema operatiu Linux, tot i que també pot anar amb d'altres com AROS i RISC OS, entre d'altres. Dins dels sistemes operatius Linux, hi ha moltes variants, la més comuna per la Raspberry Pi i la que farem servir, és el Raspbian, que és una variant del Debian, que esta fet exclusivament per a la Raspberry Pi.



## 2.1.2: Instal·lació del sistema operatiu Raspbian

Com tots els programes i fitxers, el sistema operatiu de la Raspberry Pi s'ha d'instal·lar a la targeta SD, i per tant, necessitem un lector de targetes SD.

### 1- Descarregar el SO:

Podem descarregar el SO Raspbian a la pagina oficial de la Raspberry Pi: <http://www.raspberrypi.org/downloads>. En aquesta web hi ha diferents sistemes operatius i altres programes per la Raspberry Pi. Per descarregar el Raspbian simplement hem de fer clic a una de les opcions de descàrrega (directe i torrent), la que es prefereixi.

**Raspbian**




Image	<a href="#">2013-12-20-wheezy-raspbian.zip</a>	➔ Links de descàrrega
Torrent	<a href="#">2013-12-20-wheezy-raspbian.zip.torrent</a>	
SHA-1 Checksum	ade48c874f8e4b694175de4c87d7357960961fbf	
Default login	pi / raspberry	
Description	A community-created port of Debian wheezy, optimised for the Raspberry Pi	
Release Date	2013-12-20	
Version	wheezy	
Kernel	3.10	
URL	<a href="#">Link</a>	
Release Notes	<a href="#">release notes.txt</a>	

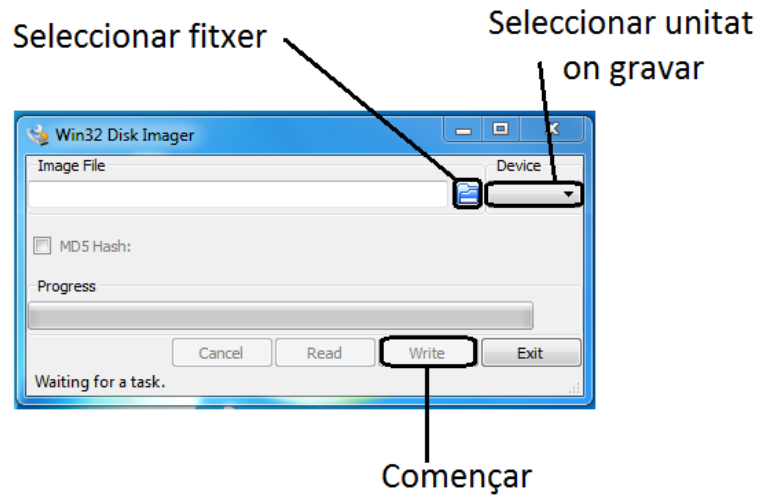
Cal tenir en compte que el fitxer pesa uns 750MB, i la descàrrega pot durar força estona en funció de la connexió. El fitxer va comprimit a dins d'un fitxer ".zip", i per tant, simplement l'hem d'extreure i ja el tindrem. L'extensió del fitxer serà ".img"

### 2- Muntar la imatge del SO a la targeta SD:

Un cop tinguem el sistema operatiu descarregat, l'hem de muntar a la targeta SD amb el programa Win32DiskImager, que podem descarregar a la pàgina següent: <http://sourceforge.net/projects/win32diskimager/>

Nota: la opció de copiar els fitxers directament a la targeta SD no funciona.

Per muntar el SO a la targeta SD, hem d'obrir el programa, i seleccionar el fitxer que volem gravar (correspon a la primera opció) i a continuació a on es vol gravar (correspon a la segona opció). Lògicament, seleccionarem el fitxer del sistema operatiu Raspbian (".img") i la targeta SD. Llavors ja només cal clicar a *Write* i esperar a que acabi el procés.



En aquest moment ja tenim la targeta SD a punt per posar-la a la Raspberry Pi i poder arrencar el sistema operatiu.

**IMPORTANT:** S'ha de tenir en compte que aquest procés és un formateig de la targeta SD i per tant, tot allò que hi hagués anteriorment a dins quedarà eliminat.

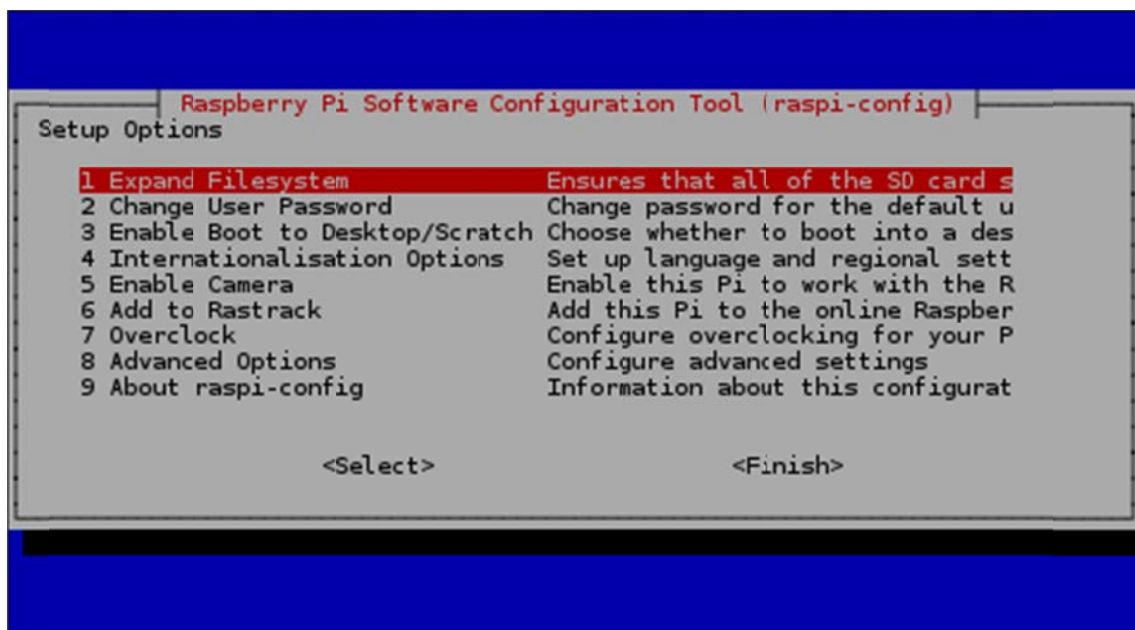


## 2.2: Posada en marxa

Ara el que toca és encendre la Raspberry Pi per primer cop introduint la targeta SD al seu lloc i connectant a la font d'alimentació (carregador mòbil) l'entrada microUSB. També haurem de connectar-hi un ratolí, un teclat ( als connectors USB ) i també una pantalla/tele a l'HDMI o a l'RCA.

### 2.2.1: Configuració inicial – El configurador raspi-config

Quan la Raspberry Pi ja estigui encesa, a la pantalla apareixerà el panell de configuració *raspi-config* (només apareixerà el primer cop que la poses en marxa si ho configures correctament)



#### **1-Expand Filesystem:**

El sistema operatiu agafa per defecte una mica més de 2 GB de la targeta SD, i aquesta opció serveix per fer augmentar l'espai de disc que tindrem fins al màxim de la targeta SD.

#### **2-Change User Password:**

Per accedir al sistema hi ha un usuari i una contrasenya, aquesta opció és per si es vol canviar.

#### **3-Enable boot to Desktop/Scratch:**

Aquí es configura com es vol que la Raspberry Pi arranqui. Per defecte arrencarà en la consola bàsica de text, però es pot canviar perquè arranqui directament a l'escriptori gràfic.

#### **4-Internationalisation Options:**

Opció per configurar l'idioma, la zona horària i el tipus de teclat.

#### **5-Enable Camera:**

Configuració necessària si n'hi vols posar una.

#### **6-Add to Rastrack:**

Opció per afegir la teva raspberry Pi a una comunitat mundial, requereix que et registris. Si tens connexió a internet a la raspberry ho pots fer directament des d'aquí.

#### **7-Overclock:**

L'overclock és la sobrecàrrega a la freqüència en què treballa un processador, dins d'aquesta opció és pot sobrecarregar el processador de la Raspberry Pi per augmentar el seu rendiment. La sobrecàrrega té diversos valors, des de "gens" fins a "turbo", passant pels entremitjos. Sobrecarregar el processador pot fer que es sobreescalfi si no esta ben ventilat, pot consumir més electricitat del compte i ocupar més SDRAM.

#### **8-Advanced Options:**

Altres opcions com l'Overscan (adaptar una pantalla quadrada a una rectangular), quin nom té la nostra Raspberry a la xarxa, la memòria assignada al processador, l'activació del control remot (SSH), la configuració de per on surt el so (HDMI o 3.5mm jack) i també hi ha la opció d'actualitzar la Raspberry Pi (connexió a internet necessària).

#### **9-About raspi-config:**

Informació sobre el configurador

### **2.2.2: Configuració inicial**

Recomano expandir l'espai de la targeta SD però no el primer cop que configures la Raspberry, sinó més endavant.

S'ha de configurar necessàriament la zona horària i el tipus de teclat. La zona horària s'ha de seleccionar qualsevol ciutat a la mateixa zona horària que nosaltres i assegurar-nos que el teclat és un teclat qwerty i de llengua

espanyola si es pot. Tot i així, s'ha de dir que es poden tenir problemes amb la configuració del teclat. Ens podem trobar que tot i escriure perfectament, els signes poden aparèixer desplaçats al teclat.


Una cosa que també s'ha de comprovar és que estigui habilitada la connexió SSH, perquè l'haurèm de fer servir després (Advanced Options)

També s'ha d'actualitzar el sistema operatiu de la Raspberry (a Advanced Options) perquè és necessari per instal·lar correctament altres programes. Els altres aspectes són totalment opcionals de modificar, però en un principi es recomana no tocar res més.

### **2.2.3: Començant a utilitzar la Raspberry Pi**

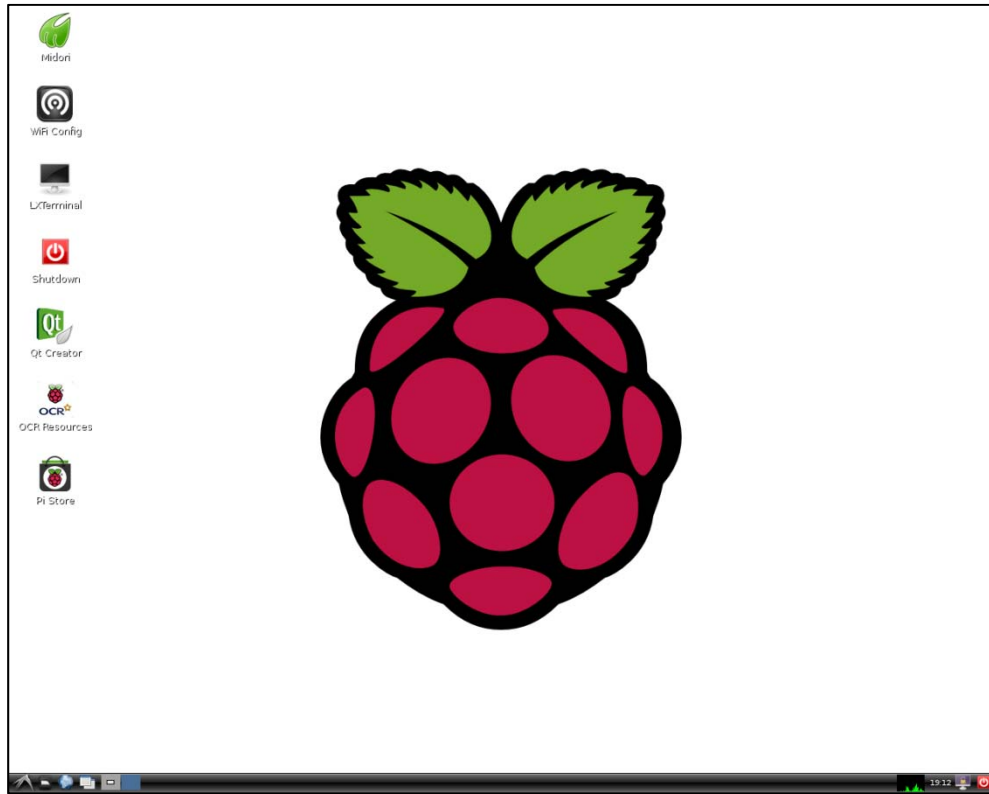
Un cop sortim del configurador, ens demanaran l'usuari i la contrasenya (per defecte són usuari: pi, contrasenya: raspberry). Un cop iniciem sessió, ens apareixerà la línia de comandaments bàsics. Aquí ja tenim l'ordinador en ple funcionament. La línia de comandaments serveix per donar les instruccions a l'ordinador, navegar per carpetes, instal·lar programes, obrir documents.. i altres coses. L'inconvenient és que tot és text i instruccions en codi, i per tant, és més pràctica la visualització de l'escriptori.

Per iniciar l'escriptori, hem d'escriure a la línia de comandaments "startx" i llavors prémer l'enter. Es pot configurar a la opció 3 del "raspi-config" perquè surti directament l'escriptori quan s'engegui la màquina.



```
pi@raspberrypi ~ $
```

Com podem comprovar, ja vénen uns quants programes instal·lats amb el sistema operatiu, com ara l'Scratch, el navegador web Midori i un editor de llenguatge de programació python entre d'altres. A partir d'aquí ja ho podem considerar com un ordinador normal i corrent. Podem accedir a la línia de comandament clicant "LXTerminal".



## 2.3: Funcionament i comandaments

### 2.3.1: Comandaments Útils

Llista d'uns quants comandaments que poden ser d'interès:

- sudo: Executar el comandament a continuació com a super-usuari
- sudo -i: Iniciar sessió com a super-usuari, a partir de llavors serà com si hi hagués un “sudo” davant de cada comandament
- ls: llista de fitxers i carpetes que hi ha a la carpeta on et trobes
- cd carpeta: accedir a la carpeta
- sd: tornar a la carpeta base ( /home/pi)
- mkdir “carpeta”: crear carpeta
- rm “fitxer”: eliminar fitxer
- halt: Apagar el sistema
- reboot: Reiniciar el sistema

### 2.3.2: Modificació d'un arxiu de text

Un arxiu normal es pot modificar des del mateix entorn gràfic, accedint a la carpeta on es troba i obrin-lo mitjançant el “Leafpad”, que ja ve instal·lat. El “Leafpad”, és com el típic bloc de notes del Windows.

Ara bé; si el vol modificar algun arxiu protegit, o que només el pot modificar el super-usuari, s'ha d'obrir des del LXTerminal.

Per obrir un fitxer de text a l'LXTerminal, es fa servir l'editor de textos “nano”. Per obrir el fitxer s'ha de fer de la següent manera:

-sudo nano “localització del fitxer”.

Així ens apareixerà el document en qüestió. Per guardar s'ha de prémer ctrl+x. Si poséssim la localització d'un fitxer que no existeix, el programa s'obriria igual, i quan guardéssim, es crearia el nou fitxer.

### **2.3.3: Instal·lació d'un programa**

Els programes es descarreguen, per tant, cal tenir connexió a internet per poder-ne instal·lar. Abans d'instal·lar un programa, es recomana tenir actualitzada la Raspberry:

```
-sudo apt-get update  
-sudo apt-get upgrade
```

Quan ja estigui actualitzada ja podem instal·lar el que vulguem.

```
-sudo apt-get install "nomdelprograma"
```

La Raspberry es posarà a buscar el programa i quan estigui ens informarà quan ocupa i altres dades, per continuar, ens demanarà que introduïm una "y" de "yes"

## 2.4: Connexió escriptori remot (SSH)

Com ja he comentat abans, una cosa important per la comoditat és la connexió per escriptori remot, que es tracta de connectar-te a través d'internet i del teu ordinador, per fer servir la Raspberry Pi, i per tant, no seran necessaris ni ratolí, ni teclat ni pantalla específics per a la Raspberry Pi. Hi ha diverses maneres de realitzar l'escriptori remot.

### 2.4.1: Instal·lació de l'xrdp

L'xrdp és un programa que haurem d'instal·lar a la Raspberry Pi, per poder-hi accedir remotament. Per instal·lar-lo, obrim l' LXterminal i hi introduïm "*sudo apt-get install xrdp*". Hi haurà un moment que haurem de escriure una "y" de "yes" per confirmar la instal·lació.

### 2.4.2: Connexió SSH amb cable ethernet d'ordinador a ordinador

Aquesta connexió, al meu parer, és la més pràctica de totes. Es tracta de connectar el cable ethernet directament de la Raspberry Pi a l'ordinador que vulguis fer servir, i per tant, no es necessita cap xarxa d'internet i es pot fer a qualsevol lloc. Per contra, mitjançant aquest sistema la nostra Raspberry no tindrà connexió a internet.

Per aconseguir-ho hem de configurar tant l'ordinador com la Raspberry Pi. Comencem configurant la Raspberry Pi:

El que hem de fer és fixar la ip ethernet (eth0) a la que es connectarà la Raspberry a través del cable. Obrim la línia de comandaments i hi escrivim "*sudo nano /etc/network/interfaces*" (sudo: executar com a superusuari (administrador), nano: editor de text simple, i a continuació la ubicació del fitxer que volem modificar.) El que ens apareixerà serà el següent:

```
pi@raspberrypi: ~
File Edit Tabs Help
GNU nano 2.2.6 File: /etc/network/interfaces Modified
auto lo
iface lo inet loopback
iface eth0 inet manual

allow-hotplug wlan0
iface wlan0 inet manual
wpa-roam /etc/wpa_supplicant/wpa_supplicant.conf
iface default inet dhcp

^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell
```

El que hem de fer a continuació és modificar el fitxer de la següent manera:

```
pi@raspberrypi: ~
File Edit Tabs Help
GNU nano 2.2.6 File: /etc/network/interfaces Modified
auto lo
iface lo inet loopback
iface eth0 inet static
address 192.168.120.2 — IP Raspberry
netmask 255.255.255.0
gateway 192.168.120.1 — IP Ordinador

allow-hotplug wlan0
iface wlan0 inet manual
wpa-roam /etc/wpa_supplicant/wpa_supplicant.conf
iface default inet dhcp

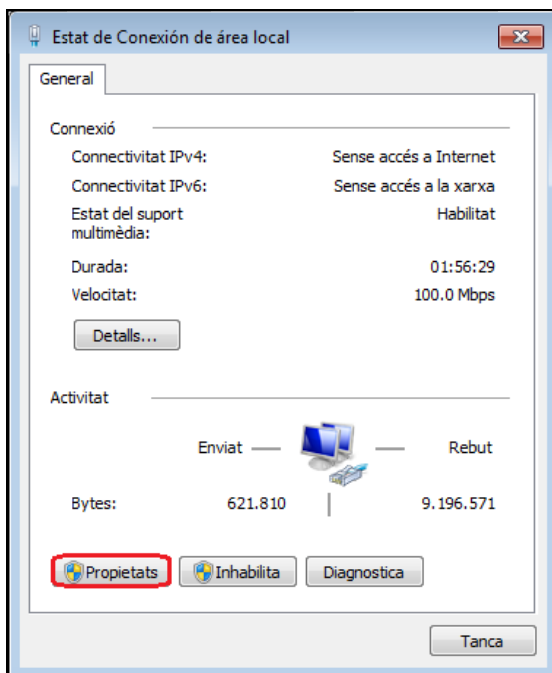
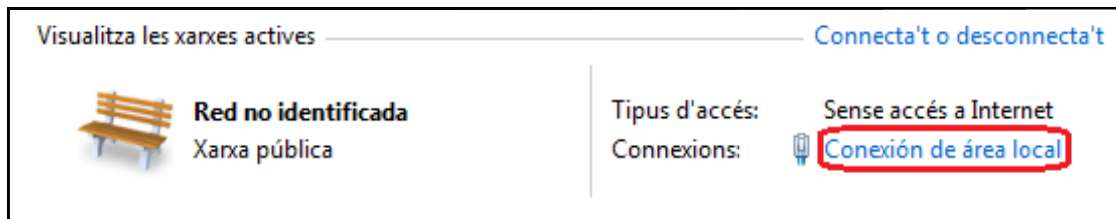
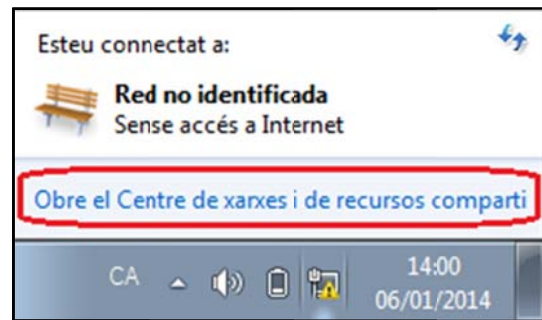
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell
```

Al punt “address”, hi posarem qualsevol número d’ip però que comenci amb 192.168.X.Y (X i Y el que vulguis). Aquest número serà el de la Raspberry. Al punt “gateway” hi posarem el número que correspondrà a l’ordinador: 192.168.X.Z (X, la mateixa que abans, i la Z un número diferent al de Y). Jo he fet servir el 192.168.120.2 per a la Raspberry i el 192.168.120.1 per a l’ordinador. Per guardar els canvis, “ctrl+x”, llavors “y”, i després “enter”. Reiniciem la Raspberry a continuació (*sudo reboot*).

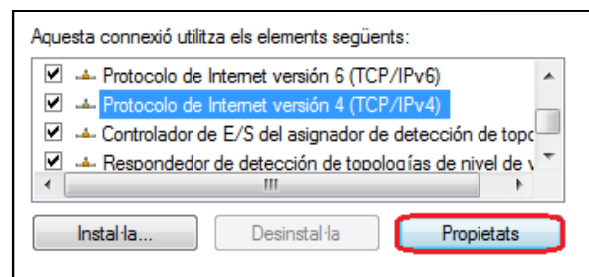
Un cop configurada correctament la Raspberry ens toca configurar l’ordinador:



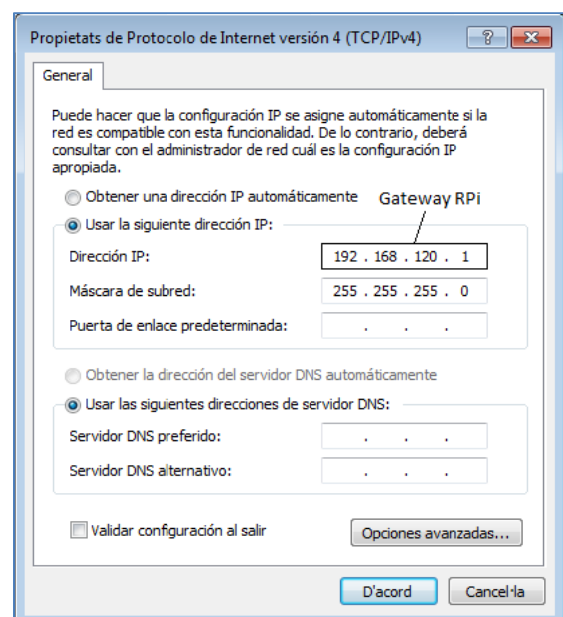
Amb el cable de xarxa connectat de Raspberry a ordinador, obrim les opcions de la connexió d'àrea local al Centre de Xarxes i Recursos Compartits, i a continuació les propietats de la connexió àrea local.



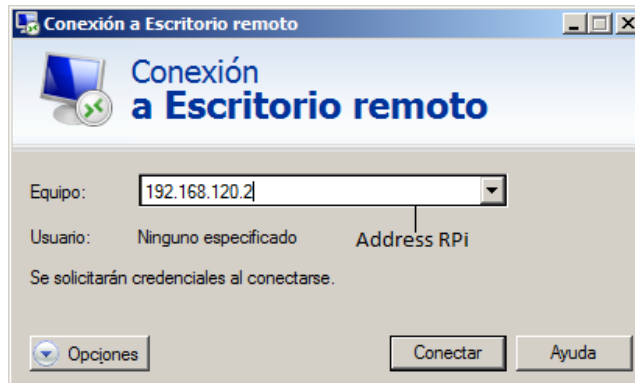
Lavors obrim les propietats del protocol d'internet versió 4 (TCP/IPv4)



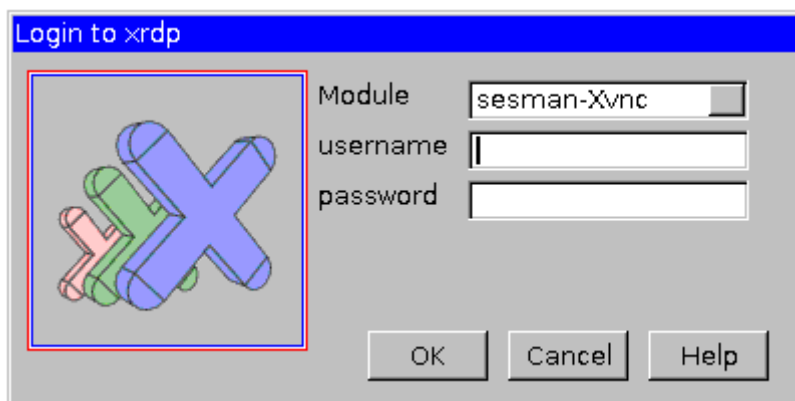
Ja només faltaria posar la direcció d'ip que hem posat a la gateway de la Raspberry. Cal tenir en compte que quan connectem l'ordinador a internet amb cable es probable que no funcioni, llavors només caldrà restablir els valors que hi havia abans.



Per acabar, només ens queda accedir a la raspberry a través del programa “Conexión a escritorio remoto” que ja ve de sèrie als Windows vista, 7 i 8. En cas de tenir un Windows anterior, s’haurà d’instal·lar un programa com el PuttY, que fa la mateixa funció. Per connectar-nos-hi simplement hem de posar la IP que hem assignat a la raspberry en el punt “Address”.



Si tot va bé, ens apareixerà la següent pantalla per iniciar sessió, només cal tenir en compte de tenir ben posada la primera opció. L’usuari i la contrasenya són els de la Raspberry Pi.



### **2.4.3: Connexió SSH a través de LAN (cable i wifi)**

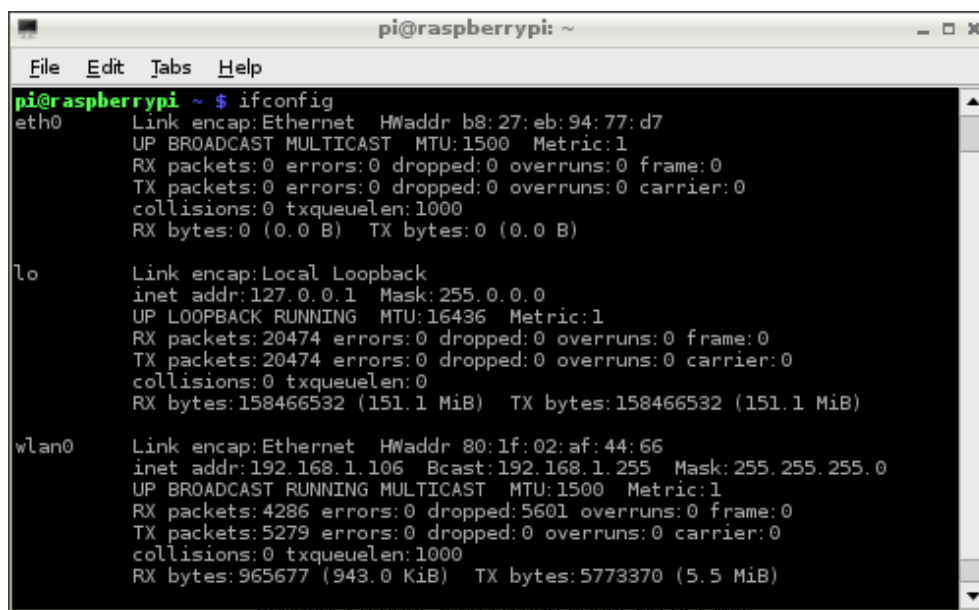
És un mètode més senzill de connectar-se i amb el qual tindrem connexió a internet a la Raspberry, però el que passa és que no és tan accessible com el cable directe, ja que es necessita una xarxa local sigui amb cable o a través de wifi. També una cosa que és molt important és que el nostre router estigui configurat de manera que doni una ip fixe, és a dir, que sempre ens doni la mateixa ip dins de la xarxa local. Això és important ja que si no és així, cada cop

hem de mirar quina ip s'ha assignat i com que necessitem una pantalla per fer-ho, llavors tot perd sentit.

**Configuració de la xarxa wifi:** Per configurar la xarxa wifi hem de fer servir el programa “WIFI Config”, que ja tindrem a l'escriptori. Aquest programa ens trobarà les xarxes disponibles i nosaltres n'haurem de seleccionar la correcta i introduir-hi la contrasenya.

Per connectar-nos a la Raspberry des de l'ordinador, ho farem igual que abans, amb el programa “Conexión a Escritorio Remoto” però posant la ip que ens sortirà al programa “WIFI Config”

En el cas que fem servir el cable ethernet, la ip la podem trobar executant el comandament “ifconfig”. Aquest comandament ens mostrarà tots es aspectes de la nostra connexió actual.



```
pi@raspberrypi ~ $ ifconfig
eth0      Link encap:Ethernet  HWaddr b8:27:eb:94:77:d7
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:20474 errors:0 dropped:0 overruns:0 frame:0
          TX packets:20474 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:158466532 (151.1 MiB)  TX bytes:158466532 (151.1 MiB)

wlan0     Link encap:Ethernet  HWaddr 80:1f:02:af:44:66
          inet addr:192.168.1.106  Bcast:192.168.1.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:4286 errors:0 dropped:5601 overruns:0 frame:0
          TX packets:5279 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:965677 (943.0 KiB)  TX bytes:5773370 (5.5 MiB)
```

#### **2.4.4: Connexió SSH a través d'internet (des de qualsevol lloc del món)**

Aquest tipus de connexió serveix per connectar-nos des de qualsevol ordinador a la Raspberry a través del programa VNC i el NO-IP. El VNC és un programa que ens ofereix el servei de control remot sobre un ordinador i el NO-IP és una pàgina web on podem crear un host (amb una ip) on vincularem la nostra Raspberry.

## Instal·lació del VNC Server

Abans de res hem d'instal·lar el vncserver a la Raspberry:

```
-sudo apt-get install tightvncserver
```

A continuació executem el programa. Ens demanarà quina contrasenya hi volem posar, aquesta contrasenya l'haurem de fer servir quan ens hi vulguem connectar. Hi haurà un punt que ens preguntarà si només ho volem per observar, hem de respondre que NO.

```
-tightvncserver
```

Reiniciem la Raspberry (sudo reboot) i llavors activem el vncserver

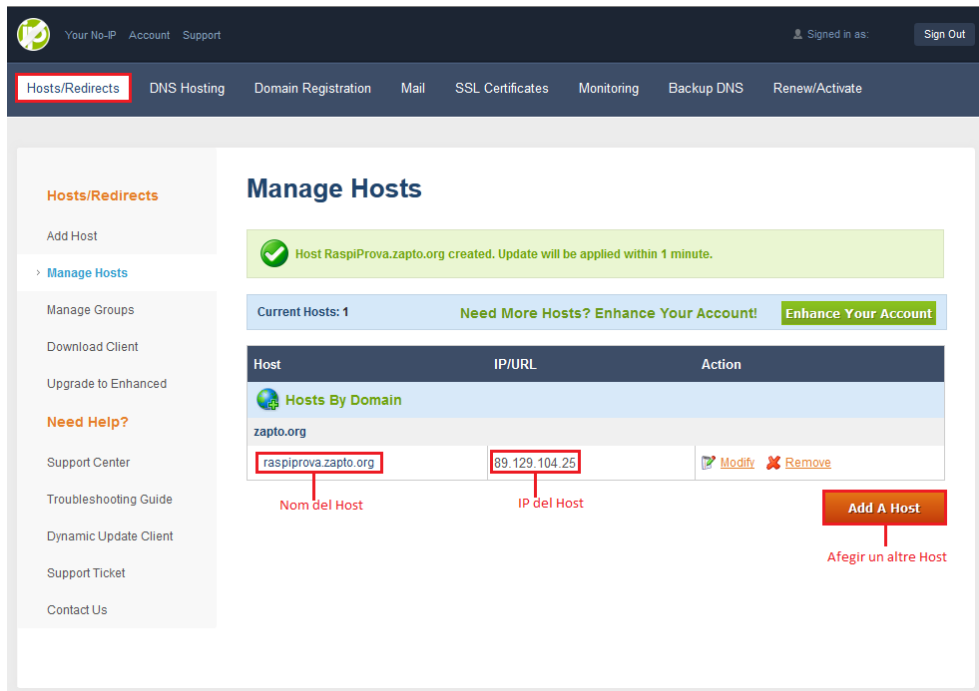
```
-vncserver :0 o també vncserver :1
```

això determinarà el port 5900 o 5901 després per connectar-nos .

Aquest pas s'ha de fer cada cop que s'encén la Raspberry per poder-nos-hi connectar.

## EI NO-IP

El primer que hem de fer és registrar-nos a [www.noip.com](http://www.noip.com). Un cop registrats hem de crear un host al que després vincularem la Raspberry i a través del qual ens connectarem.



Llavors ens cal instal·lar un paquet del no-ip necessari a la Raspberry; durant el procés, ens demanarà unes quantes dades, entre les quals dades sobre el compte que ens haurem fet a noip.com. Els altres paràmetres ja va bé deixar els que hi ha per defecte.

```
-mkdir no-ip  
-cd no-ip  
-wget http://www.no-ip.com/client/linux/noip-duc-linux.tar.gz  
-tar -zxvf noip-duc-linux.tar.gz  
-cd noip-2.1.9-1/  
-make  
-sudo make install
```

Ara toca crear un fitxer de text que servirà per iniciar el servei del no-ip.

*-sudo nano /etc/init.d/noip2*

A dins del fitxer hi copiarem el següent i el guardarem (ctrl+x,y,enter)

```
#!/bin/bash
### BEGIN INIT INFO
# Provides: Servicio No-IP
# Required-Start: $syslog
# Required-Stop: $syslog
# Default-Start: 2 3 4 5
# Default-Stop: 0 1 6
# Short-Description: arranque automatico para no-ip
# Description:
#
### END INIT INFO
sudo /usr/local/bin/noip2
```



The screenshot shows a terminal window titled 'pi@raspberrypi: ~'. The window contains the GNU nano 2.2.6 editor with the file '/etc/init.d/noip2' open. The editor's content is identical to the code block above. The terminal window also shows a status bar at the bottom with various keyboard shortcuts like '^G Get Help', '^O WriteOut', etc.

Com s'ha de veure el fitxer

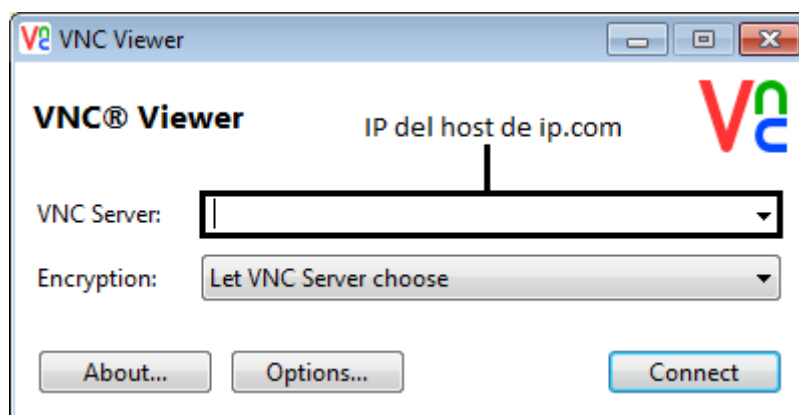
I per acabar li donem permisos d'execució i el posem a la cua per arrancar-se quan s'engegui la Raspberry Pi

```
-sudo chmod +x /etc/init.d/noip2
-sudo update-rc.d noip2 defaults
```

## El VNC Viewer

Per connectar-nos a la Raspberry amb aquest servei necessitem el programa concret del VNC, que és el VNC Viewer, que podem descarregar-nos des de la pàgina web [www.realvnc.com](http://www.realvnc.com).

El programa funciona com la "Conexión a Escritorio Remoto" del Windows, hi hem d'escriure la IP que tindrà la Raspberry, que serà la que tindrà el nostre host creat a ip.com. També caldrà introduir la contrasenya que vam posar a la Raspberry quan hi vam instal·lar el VNC Server.



### IMPORTANT:

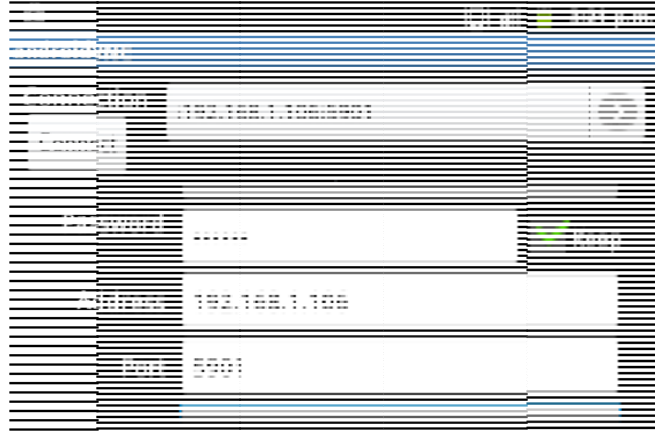
Cal tenir en compte que perquè la connexió sigui possible, s'han d'obrir uns quants ports del router on està connectada la Raspberry. Els ports que s'han d'obrir són el 22, el 80, i del 5900 al 5910.

### 2.4.5: Connexió SSH a través del mòbil (amb el VNC)

Aquesta manera de connectar-se és idèntica de com ho faríem amb l'ordinador, podem veure l'escriptori i interactuar-hi a través de la pantalla del telèfon mòbil. Per connectar-nos-hi necessitem l'aplicació VNC per mòbil, en el cas d'android, "AndroidVNC".

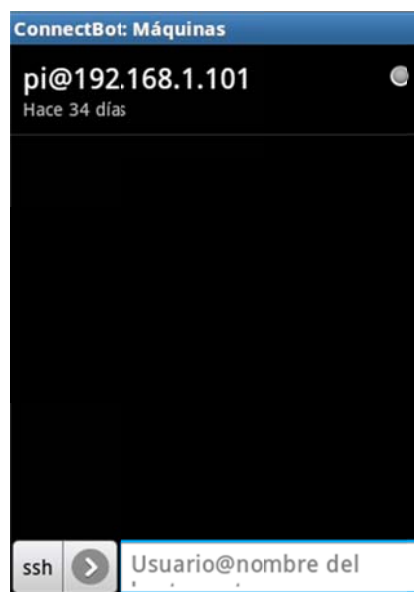
Aquesta aplicació funciona de manera semblant al VNC Viewer de l'ordinador, hem d'omplir els camps de contrasenya ( amb la contrasenya que vam posar a la Raspberry quan vam instal·lar el VNC Server), la direcció d'IP (la del no-ip o

la que tingui a la xarxa local, si és que ens trobem a la mateixa xarxa), i el port ( si el comandament d'activar el VNC Server de la Raspberry va ser “vncserver :0”, serà 5900, si va ser “vncserver :1”, serà 5901).



Aplicació androidVNC del mòbil

En el cas que el servidor VNC de la Raspberry no estigui activat i ens trobem a la mateixa xarxa que la Raspberry (connexió LAN), podem activar-lo amb una altra aplicació que és el ConectBot. Aquesta aplicació no ens permet una visualització gràfica, però ens permet accedir a la línia de comandaments, on hi podrem executar el comandament “vncserver :1” o “vncserver:0” i per tant, llavors ja podrem accedir-hi mitjançant el VNC.



Aplicació ConnectBot del mòbil



# Utilitats de la Raspberry

## 3.1: Programes Interessants:

### 3.1.1: Minecraft Pi

Hi ha una versió del minecraft específica per la Raspberry Pi!

Potser és una mica antiquada per serveix perfectament i amb poc lagg.

***Instal·lar i extreure,***

```
"wget -c https://s3.amazonaws.com/assets.minecraft.net/pi/minecraft-pi-0.1.1.tar.gz"
```

```
"tar -zxvf minecraft-pi-0.1.1.tar.gz"
```

Per executar, entrem a la carpeta /home/pi/mcpi i executem el fitxer minecraft-pi

```
"cd mcpi"
```

```
"./minecraft-pi"
```

### 3.1.2: Scrot

L'Scrot és un programa que ens permet fer captures de pantalla. Aquest és el programa que he fet servir per fer totes les captures de la Raspberry d'aquest treball.

***Instal·lar,***

```
"sudo apt-get install scrot"
```

Comandaments de l'Scrot, que s'han de fer servir des de l'LXTerminal

```
"scrot" → Captura de pantalla instantània
```

`“scrot -d X → Captura de pantalla d'aquí X segons`

`“scrot /home/pi/carpeta/nom.png” → Canviar carpeta on apareixen les captures i el format (un altre exemple seria : “scrot /home/pi/captures/cap.jpg”).`

### **3.1.3: VLC Player**

Un excel·lent reproductor de vídeo també disponible a la Raspberry Pi, com les altres versions, seleccionar pistes d'àudio i de subtítols. Té l'avantatge que porta la majoria de còdecs instal·lats

***Instal·lar,***

`“sudo apt-get install vlc”`

Atenció: Reproduir un vídeo estant connectat a la Raspberry a través l'escriptori remot provocarà problemes i anirà molt relantitzat.

### **3.1.4: LibreOffice**

Versió de l'openoffice que podem instal·lar a la nostra Raspberry.

És un paquet d'office on s'inclou un editor de texts, un editor de fulls de càlcul, un administrador de bases de dades, una utilitats per fer dibuixos i logotips i també un editor de presentacions gràfiques. En definitiva un conjunt d'utilitats que no poden faltar en cap ordinador.

***Instal·lació,***

`“sudo apt-get install openoffice.org”`

### **3.1.5: Buscar altres programes**

Si fan falta altres programes, s'han de buscar programes compatibles amb el sistema operatiu Debian a l'ordinador.

Si un programa és compatible amb Debian, és molt probable que també ho sigui amb el Raspbian. Per instal·lar el programa, abans de buscar alternatives mireu si funciona el procediment habitual.

## 3.2: Utilitats com a Sistema Operatiu:

Aquest tipus d'utilitats són alternatives al Raspbian, perquè són altres sistemes operatius. No es pot tenir dos sistemes operatius a la mateixa targeta SD.

Se n'han de fer servir de diferents

### **3.2.1: OpenELEC**

L'openELEC és un sistema operatiu que convertirà la nostra Raspberry en un Media Center per reproduir tot tipus de vídeo i música. Cal aprofitar la sortida HDMI per reproduir vídeos en alta definició al nostre televisor. L'openELEC té una interfície gràfica que ens recordarà a la de qualsevol DVD o fins i tot els reproductors de les SmartTV!

Podem descarregar l'OpenELEC des de la pàgina oficial de la Raspberry Pi, igual que vam fer amb el Raspbian: descarreguem el fitxer comprimit muntem la imatge en una SD (que no hi hagi res ! recordeu que és un formateig !).

### **3.2.2: RaspBMC**

Es tracta d'una altra distribució per convertir la Raspberry en un Media Center. És molt semblant a l'OpenELEC i també es pot descarregar a la pàgina oficial de la Raspberry Pi.

### **3.2.3: PiMAME**

El PiMAME convertirà la nostra Raspberry en un emulador d'una consola arcade .També és capaç d'emular una PlayStation, Game Boys d'altres consoles velles.

La instal·lació d'aquest sistema operatiu funcionaria exactament igual que la resta.

Podem descarregar el piMAME a <http://pimame.org>

# Els pins GPIO de la Raspberry

## 4.1: Què són els pins GPIO ?

Els GPIO (General-purpose input/output), són uns pins que serveixen per fer qualsevol tipus de circuit elèctric que requereixi entrades i sortides digitals. Això vol dir que mitjançant aquests pins podem crear aplicacions que encenguin i apaguin algun dispositiu controlades per algun interruptor o polsador.

La Raspberry Pi disposa de 26 pins col·locats en dues files, i no tots els pins són GPIOs, sinó que hi ha sortides de corrent a 3.3 V i 5V, pins terra (GRND) i diferents varietats de pins com ara els SDA i SCL, els UART TX i RX, i els SPI que poden tenir altres funcions a més a més de GPIO normal.

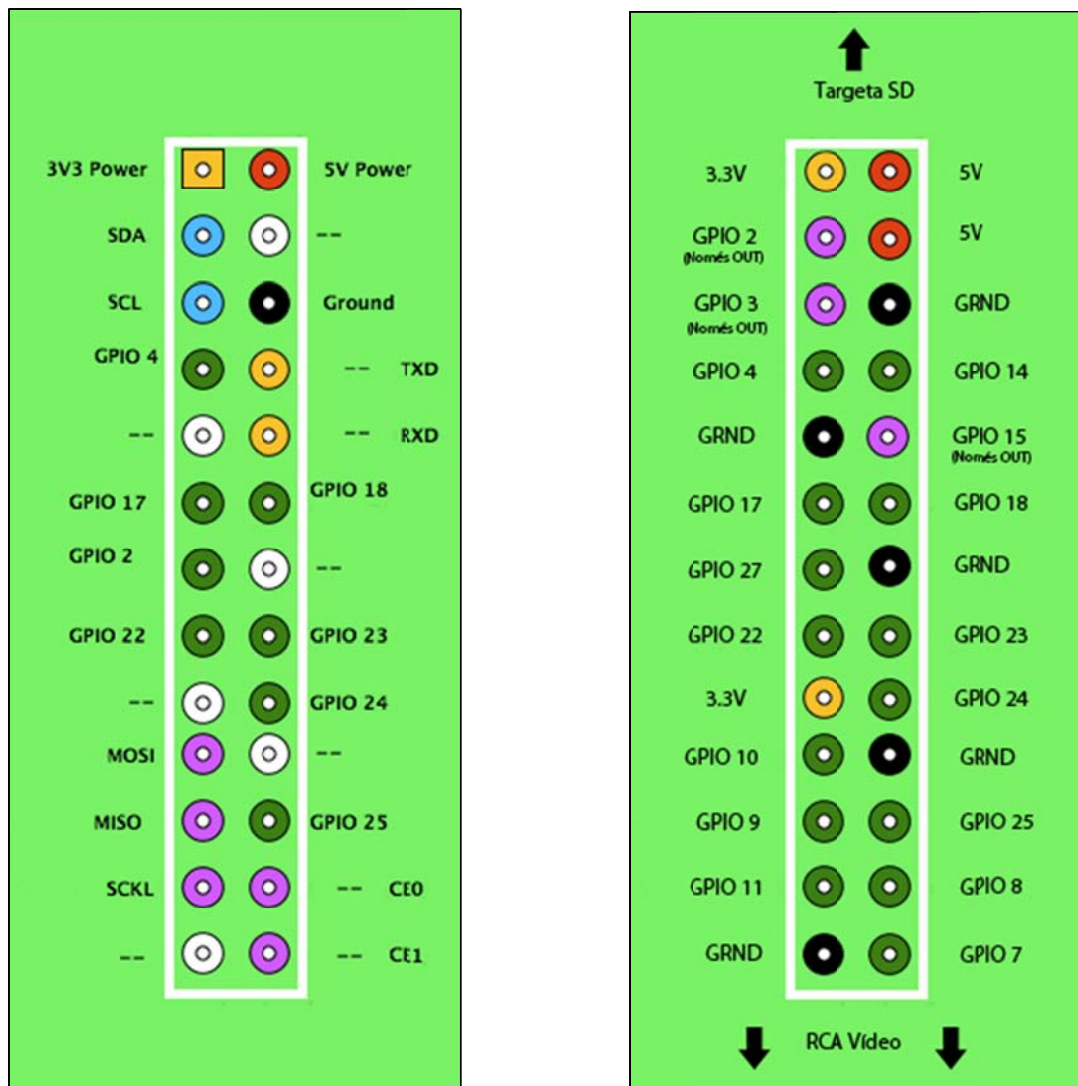


En aquest tutorial, farem servir els pins únicament com a GPIO de manera que només ens hem de preocupar de diferenciar els pins GPIO de les terres i les corrents.

La Raspberry Pi té un sistema de protecció contra els curts-circuits, que fa que quan se'n produeix un (connexió directa accidental entre un pin de corrent i un de terra) la Raspberry es manté apagada fins que no s'acabi el curtcircuit, llavors es reinicia. Per tant, cal parar anar amb compte de guardar tots els

fixers mentre remenes els pins perquè un curt-circuit pot representar una pèrdua important d'informació.

A continuació hi ha uns esquemes dels pins de la Raspberry, a la dreta trobem l'esquema amb tots els tipus de pins. A l'esquerra veiem l'esquema que farem servir on només apareix quin GPIO és cada pin i els que només farem servir de sortida per evitar complicar-nos.



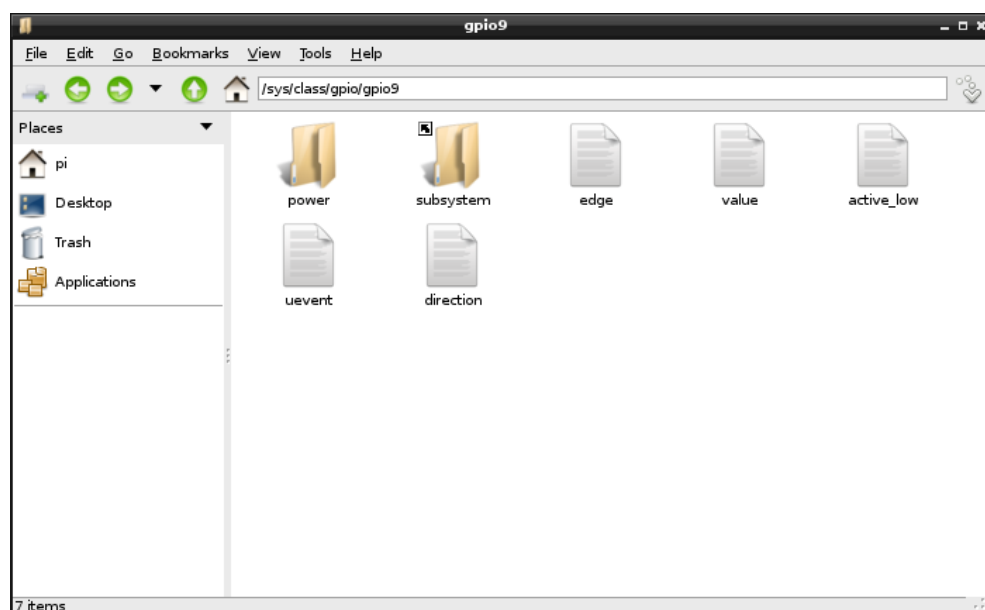
Com podem veure, gairebé podem fer servir tots els GPIO de forma normal, però tal com es veu a l'esquema el GPIO 2, el 3 i el 15 només els farem servir com a sortides. L'esquema l'he muntat testejant cada GPIO amb el multímetre i he pogut comprovar que a l'hora de rebre la senyal dels GPIO 2,3 i 15 es comporten de manera diferent als altres, per tant, per evitar problemes he evitat fer-los servir com a entrada.

## 4.2: Com funcionen els pins GPIO?

En altres tipus de plaques electròniques, com per exemple una “arduino” o una “picaxe”, la part de la utilització dels pins és bàsica. En canvi, en una Raspberry Pi, com que és un ordinador (i per què li fan falta pins a un ordinador perquè funcioni?) la part del control dels pins no és evident i està d'entrada una mica amagada.

El sistema operatiu Raspbian, que cal recordar que està basat en Linux, fa funcionar els GPIO a través de fitxers, que contenen la informació de cada GPIO com ara la direcció (in/out) o el valor que tenen tant com per entrada com per sortida. Per tant, interactuar amb els GPIO es basa en la manipulació d'aquests fitxers.

Per defecte, quan engeguem la Raspberry, tots els GPIO estan deshabilitats de manera que abans de res caldrà habilitar-los. Els GPIO s'han d'habilitar mitjançant el fitxer “export” que es troba a la carpeta dels GPIO, que és “/sys/class/gpio”. Quan haurem exportat un GPIO ens apareixerà una carpeta de nom gpio+”número del GPIO que hàgim exportat” (per exemple gpio9) dins de la qual hi haurà els fitxers que he comentat abans.



Carpeta “gpio9” al directori “/sys/class/gpio/gpio9”  
on trobem els fitxers relacionats amb el GPIO 9.

Els únics fitxers en que ens fixarem són el fitxer “direction” i el fitxer “value” i són simples fitxers de text (com un .txt de Windows),

- **Direction:** Fitxer que determina si el GPIO en qüestió és una entrada o una sortida (input/output). Si hi escrivim “in”, serà una entrada, si hi escrivint “out”, serà una sortida, així de senzill.
- **Value:** Fitxer que determina si el GPIO dóna corrent o no (en el cas d’una sortida) o si rep corrent o no (en el cas d’una entrada). Com que tot és digital només hi pot haver-hi escrit el valor 0 o 1.

En aquest apartat farem servir el GPIO de manera manual des de la línia de comandaments (LXTerminal).

#### **4.3.1: Llista de Comandaments**

**Habilitar pin:** Es tracta d'introduir el valor del pin al fitxer "export". El comandament que hem de fer servir és:

*"echo (número GPIO) > /sys/class/gpio/export"*

```
pi@raspberrypi ~ $ echo 14 > /sys/class/gpio/export
```

Habilitar el GPIO 14

Amb aquesta ordre es crearà el la carpeta del GPIO i aquest quedarà habilitat. Per defecte queda configurat com una entrada.

**Configurar com a entrada/sortida:** Es tracta d'introduir "in" o "out" al fitxer "direction". El comandament que hem de fer servir és:

*"echo (in/out) > /sys/class/gpio/gpio(número GPIO)/direction"*

```
pi@raspberrypi ~ $ echo out > /sys/class/gpio/gpio14/direction
```

Configurar el GPIO 14 com una sortida

```
pi@raspberrypi ~ $ echo in > /sys/class/gpio/gpio14/direction
```

Configurar el GPIO 14 com una entrada

**Canviar estat de GPIO (sortida) :** Es tracta d'introduir "0" o "1" al fitxer "value" quan el GPIO és una sortida. Això determinarà si don corrent o no. El comandament que hem de fer servir és:

*"echo (1/0) > /sys/class/gpio/gpio(número GPIO)/value"*

```
pi@raspberrypi ~ $ echo 1 > /sys/class/gpio/gpio14/value
```

Encendre el GPIO 14



```
pi@raspberrypi ~ $ echo 0 > /sys/class/gpio/gpio14/value
```

Apagar el GPIO 14

**Llegir el valor d'entrada** : Es tracta de llegir quin valor hi arriba a un GPIO que sigui una entrada. La Raspberry ens respondrà amb un 1 o un 0 en funció de si arriba corrent o no al GPIO. El comandament que hem de fer servir és:

*“cat /sys/class/gpio/gpio(número GPIO)/value”*

```
pi@raspberrypi ~ $ cat /sys/class/gpio/gpio14/value
```

Llegir el valor del GPIO 14 (el rectangle al costat del “cat” és el cursor del text, allà hi ha un simple espai)

#### **4.3.2: Comprovació dels comandaments**

És bastant útil comprovar físicament com funcionen els GPIO per acabar-se'n de fer la idea. Podem comprovar el funcionament de tots aquests comandaments mitjançant un multímetre (comprovació sortides) i un simple cable (comprovació entrades).

Per exemple, podem comprovar que la sortida GPIO 17 funciona mirant que hi ha 0V entre el pin GRND i el GPIO quan aquest te assignat el valor 0, i com ascendeix a 3.3V quan li assignem el valor 1. Com a entrada, demanem el valor que te el pin sense res, ( hauria de donar 0) i llavors connectat directament amb un cable a pin de 3.3V ( hauria de donar 1)

Tot i això pot ser útil una comprovació visual muntant un simple circuit amb un led i un altre amb un polsador.

## 4.4: Programació dels GPIO

Com hem vist, el control dels pins GPIO manualment va molt bé i és senzill, però no podem pas parlar de gaire pràctic. El que necessitem és poder programar-los.

Hi ha moltes maneres de programar els pins GPIO de la Raspberry i una pila de llenguatges de programació que poden servir-nos. Alguns dels llenguatges de programació útils són els següents: el python (és el més adaptat a la Raspberry. Es basa en una programació seqüencial és a dir, que el programa s'executa línia rere línia, en llenguatge python. El sistema operatiu ja ve amb un editor de python instal·lat, el IDLE). També es pot programar amb scratch (tot i que no és tant freqüent), amb programes especialitzats (algun programa fet expressament per programar els pins corre per internet..) i amb programes amb algun altra llenguatge de programació com el C o el C++.



El programa que he fet servir és el QT Creator, que és un programa lliure i basat amb el C++ amb el que es poden desenvolupar programes directament a través d'una interfície gràfica.



## 4.5: El QT Creator

### 4.5.1: Per què el QT Creator?

Vaig triar fer servir el QT Creator per diferents motius,

És un programa que funciona a partir de la programació d'advents i en un entorn gràfic on escrivim un codi de programació per cada event que vulguis (com per exemple clicar un botó). Així doncs, com que es disposa d'una interfície gràfica, vaig pensar que això seria molt útil en termes d'accessibilitat (disposant de l'entorn gràfic podrem interactuar fàcilment amb el programa que fem i també des d'un dispositiu mòbil que ens permeti veure l'escriptori de la Raspberry).

Un altre motiu és que fa un temps vaig fer una petita aplicació amb el visual basic, i com que també funciona creant un entorn gràfic vaig considerar que el QT Creator seria semblant i per tant, ja hi estaria habituat.

### 4.5.2: Instal·lació del QT Creator a la Raspberry Pi

Abans d'instal·lar res, recordem-nos d'actualitzar correctament la Raspberry i que cal accés a internet per qualsevol actualització o instal·lació.

```
-sudo apt-get update
```

```
-sudo apt-get upgrade
```

A continuació cal comprovar que hi ha instal·lada la llibreria g++, que es necessita pel QTCreator. Si la Raspberry està actualitzada, és força probable que et digui que no cal instal·lar res perquè ja hi és, però cal comprovar-ho.

```
-sudo apt-get install g++
```

Finalment, instal·lem el QT Creator i també un paquet d'eines complementàries pel propi programa que instal·larem abans.

```
-sudo apt-get install qt4-dev-tools
```

-sudo apt-get install qtcreator

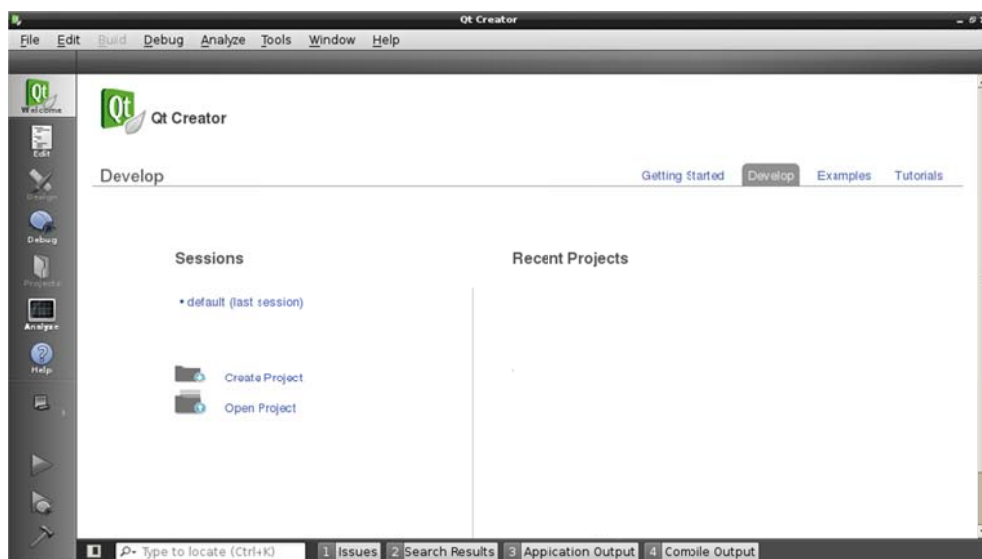
El QT Creator és un programa força gros i que per tant, tardarà força estona a descarregar-se ( evidentment en funció de la connexió que tinguem ) i una estona més per instal·lar-se. Quan hagi acabat es recomana reiniciar el sistema, però el podem executar sense fer-ho.

El programa ens apareixerà al menú de programes juntament amb els seus paquets d'eines. Hem de fer clic al que posa simplement "QT Creator.

### **4.5.3: Configuració inicial del QT Creator**

Quan obrim el QT Creator ens apareixerà la pantalla de benvinguda, on hi trobem enllaços a tutorials i exemples. Abans però, de començar cap projecte hem de configurar correctament el compilador dels programes.

L'opció que busquem és la "Build & Run" que és a dins de "Options", al menú de "Tools".

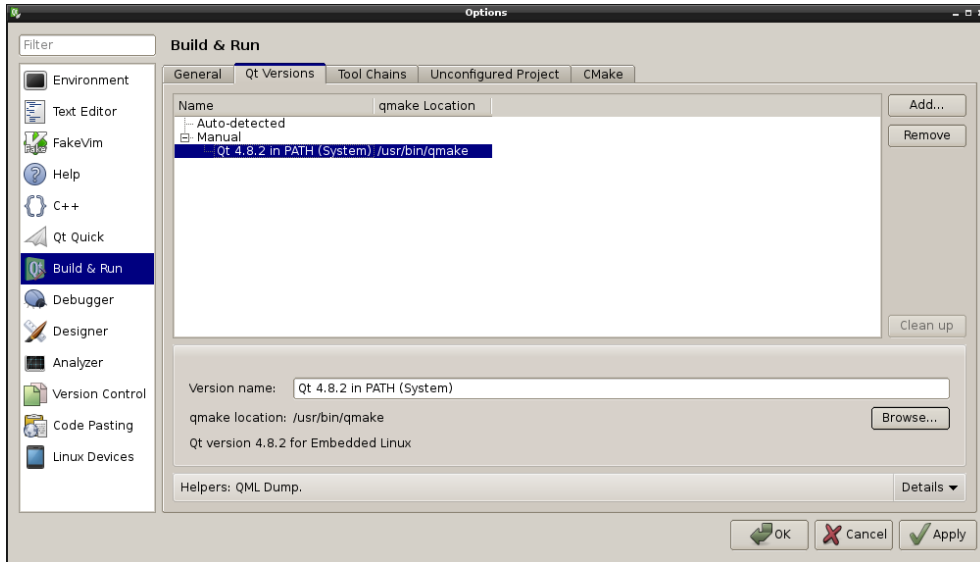


Pantalla inicial del QT Creator

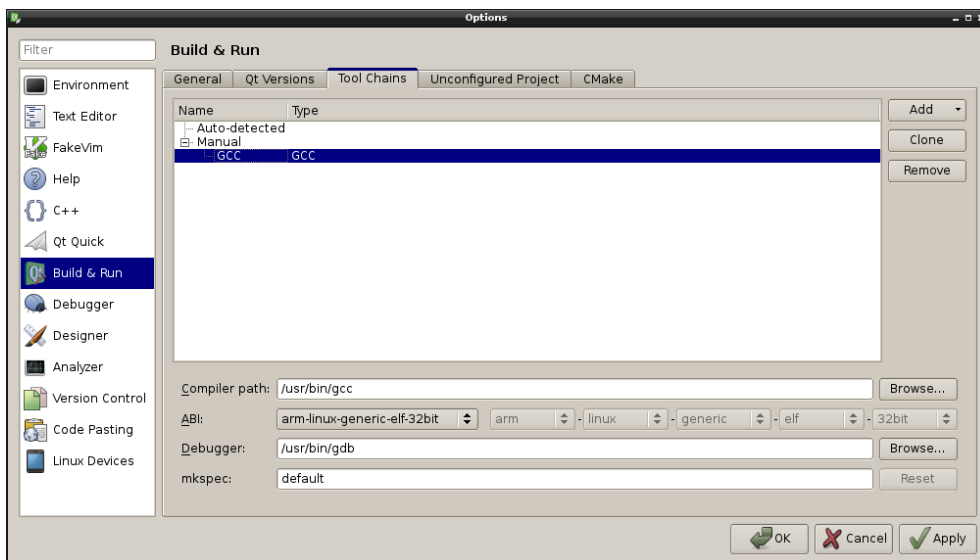
Dins de "Build & Run", hem d'anar a "QT Versions", seleccionar "Manual" i a continuació clicar "Add". A sota ens demanarà el "Version Name", i clicant "browse", hem de seleccionar el fitxer "qmake" que serà a la carpeta "/usr/bin".

A continuació, anem a “Tool Chains”, seleccionem “manual”, cliquem a “Add” opció “GCC”, i a sota, seleccionem l’arxiu “gcc” de la carpeta “/usr/bin” allà on posa “Compiler path”, i l’arxiu “gdb” de la mateixa carpeta allà on posa “Debugger”. La resta es deixa com està.

Llavors només falta clicar l’OK, i ja ho tindrem configurat



La configuració de “QT Versions” hauria de quedar així



La configuració de “Tool Chains” hauria de quedar així

#### **4.5.4: L'entorn de QT Creator**

**Crear un nou projecte:** Simplement hem d'anar a "File > New project", el tipus de programa que farem serà una "Qt Gui Application".

Seguidament ens demanarà quin títol li volem posar i a on volem que es guardi.

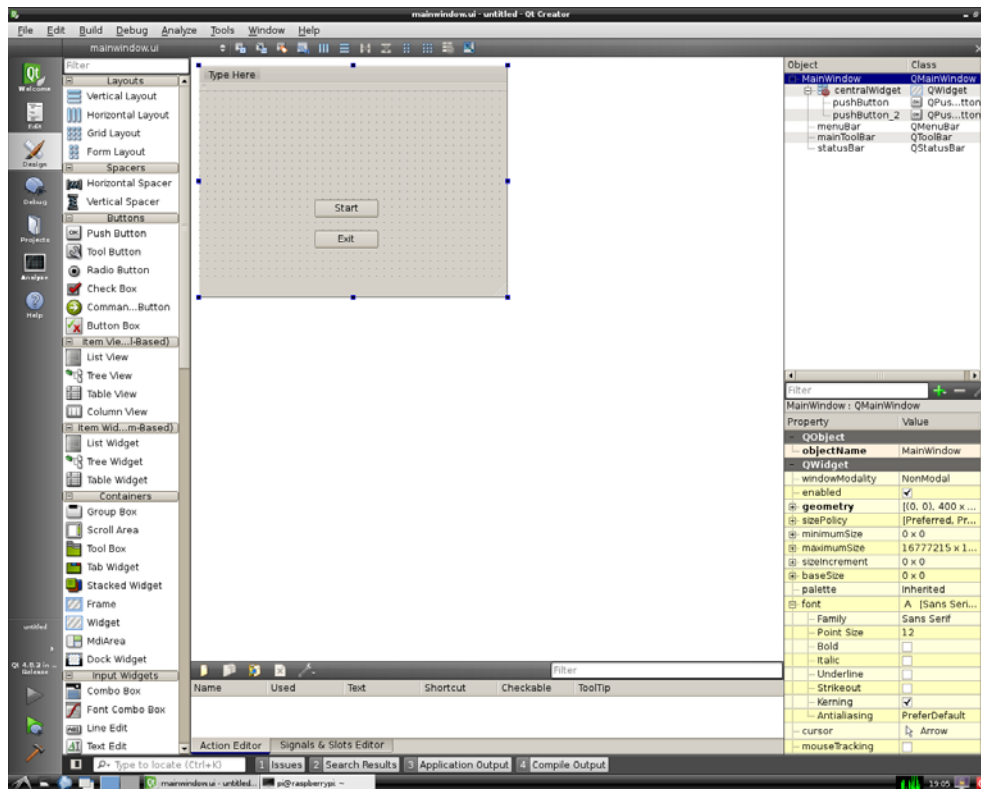
El QT Creator treballa a dos nivells, el nivell gràfic, on podem veure com és el nostra programa i quins elements el formen ( botons, textos, barres...), i el nivell de programació, on hem d'escriure que volem que faci el programa quan es produeixi un event (el més comú: que es cliqui un botó).

Cada projecte té quatre tipus diferents de fitxers:

- **Fitxer .pro**  
Fitxer per lligar tots els altres en el mateix programa
- **Fitxer "header" (.h)**  
Cada finestra en té un. Allà s'hi troben les classes que es faran servir en el programa.
- **Fitxer "source"**  
Cada finestra en té un. Fitxer on hi ha tota la programació.
- **Fitxer "form"**  
Cada finestra en té un. Fitxer on hi ha tots els paràmetres gràfics de la finestra i de tots els objectes que hi ha.

#### **La interfície de disseny gràfic:**

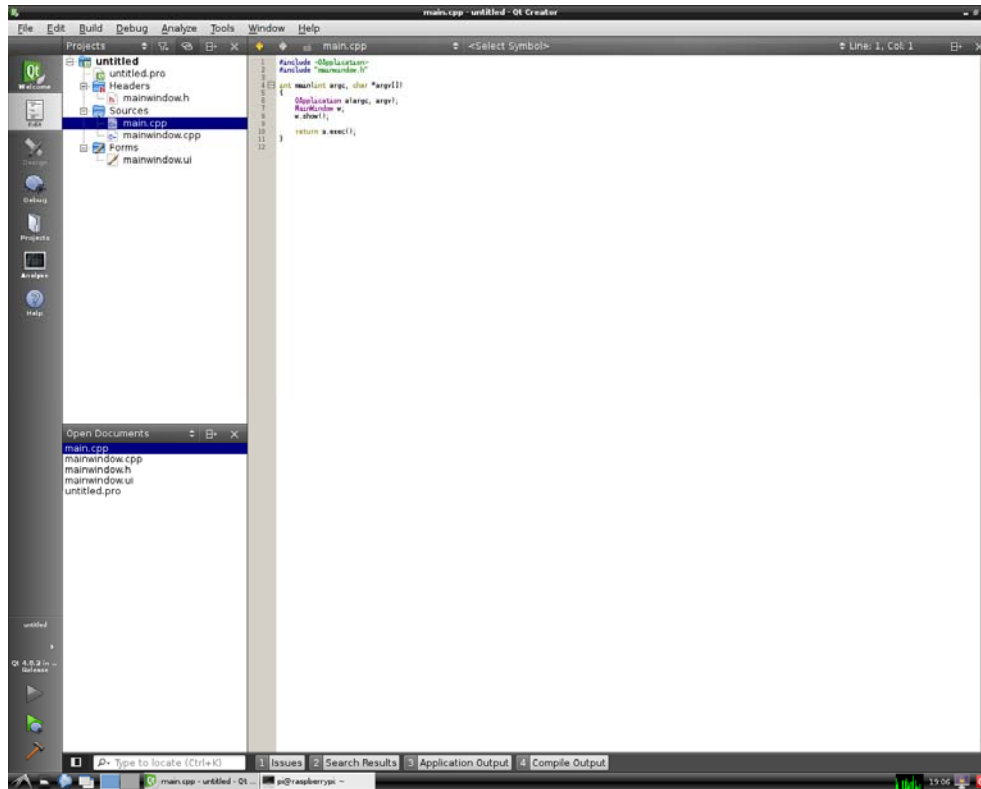
- **Menú del QT Creator:** Situat a l'extrem esquerre, ens permet anar al menú principal, a la zona d'edició de la programació i a altres opcions del programa. A la zona de baix, hi ha el "run", per provar el programa (el "play") i el "build", per muntar el programa un cop funcioni be (el martell)
- **Els objectes:** Situada a l'esquerra, la llista d'objectes que podem posar a la nostra finestra



- **La nostra finestra:** Situada a la part central
- **Llista d'objectes de la nostra finestra:** Situada a dalt a la dreta, hi ha la llista de tots els objectes que hem posat a la nostra finestra.
- **Propietats de l'objecte seleccionat:** Situatedes a baix a la dreta, hi ha totes les característiques de cada objecte, com ara la mida, el color, la lletra, si es veu.... Allà podem canviar el nom dels objectes per a la nostra conveniència ( millor “començar” i sortir”, que no “pushbutton1”, “bushbutton2”...)

### La interfície de programació:

- **Menú del QT Creator**
- **Navegador d'arxius:** Situat a dalt a l'esquerra hi ha el navegador d'arxius, des d'on podem accedir a tots els arxius del programa que estem fent i veure'n el contingut



- **Historial d'arxius oberts:** Situat a baix a l'esquerra, hi ha l'historial de els arxius que hem anat obrint amb el temps
- **Espai de programació:** La part central. Aquí és on escriurem el programa
- **Barra de correcció:** És la barra que hi ha a baix de tot on ens apareixeran els errors que hi hagin el programa quan l'intentem fer anar o construir.

#### **4.5.5: Començant un programa**

Quan es comença un programa, normalment es comença dissenyant la part gràfica. Accedim al disseny gràfic fent doble-clic al fitxer "form (.ui).

El funcionament bàsic és arrossegar l'objecte de la llista a la nostra finestra, donar-li les mides corresponents i canviar-li el text en cas que en tinguin. Els objectes més comuns són:

- **El Botó (Push Button)**
- **Label:** És un objecte on hi ha text
- **LineEdit:** Objecte que permet escriure



Quan hàgim creat tots els objectes que creiem que necessitem és hora de programar. Si més tard en falten o en sobren, en podem afegir o esborrar els objectes que calguin.

Per començar a programar, cal seleccionar a quina acció programem, és a dir, quan el nostre codi començarà a utilitzar-se ( en tots els programes que he fet, ha estat l'acció "clicar un botó"). Per seleccionar una acció fem clic dret sobre un objecte i cliquem "go to slot", llavors seleccionem l'acció que vulguem (en el cas d'un botó, és "Clicked()")

#### **4.5.6: La programació**

A continuació explicaré quins elements trobem dins del full de programació (source .> .cpp) i els comandaments que he fet servir als meus programes.

IMPORTANT: Al final de cada línia de programació (que no inclogui programació dins seu com ara com un if, un else, un while, una acció.. etc), ha de portar un punt i coma al final per declarar que s'ha acabat la línia.

##### **4.5.6.1: Elements en el full de la programació:**

- **Llibreries:** Es troben a dalt de tot del programa i estan escrites de color blau i verd. Les llibreries el que fan és incloure funcions que podem fer servir, i que per defecte el programa no les reconeix
- **Declaració de variables:** S'acostuma a situar just a continuació de les llibreries. En aquest punt es creen les variables, indicant-ne el tipus, nom i valor (el valor no és necessari)
  - **Una variable** són uns quants bytes de memòria que guarden un valor, una lletra, una paraula, o cadenes de números i paraules, depenent de quin tipus de variable es tracti.
- **Accions automàtiques:** Són dues accions que apareixen escrites automàticament el programa, que corresponen a la de fer aparèixer la nostra finestra quan obrim el programa, i fer-la desaparèixer quan el tanquem.

- **Accions designades:** Són les accions que nosaltres hem creat (a partir de l'entorn gràfic, go to slot ..).  
A dins d'aquestes accions és on s'hi trobarà la major part del nostre programa.

#### 4.5.6.2: Tipus de variables: Tipus més freqüents

- **INT:** Variable que conté un valor numèric enter
- **CHAR:** Variable que conté una lletra
- **CHAR\*:** Variable que conté una paraula
- **BOOL:** Variable que conté un valor "true" o "false"
- **QSTRING (String del QT Creator):** Variable que conté una cadena de caràcters i valors
- **FLOAT:** Variable que conté un valor numèric real
- **IFSTREAM:** Variable que serveix per emmagatzemar el contingut d'un fitxer de text que s'ha llegit.\*
- **OFSTREAM:** Variable que serveix per emmagatzemar un contingut que s'escriurà en un arxiu de text.\*

\*Les variables ifstream i ofstream necessiten llibreries addicionals

#### 4.5.6.3: Declaració de variables

Per declarar una variable s'escriu el tipus de la variable que es vol declarar i a continuació el nom que se li vol posar. Un mateix nom no es pot utilitzar en més d'una variable. Eviteu posar noms a les variables que ja apareguin al programa com per exemple el nom d'un objecte, d'una finestra, o el del programa.

Exemple:

*-char inicial;* Hem creat la variable "inicial", que és del tipus "char"

*-float num = 47.5;* Hem creat la variable "num" que és del tipus "float",  
i que conté el nombre "47.5"

#### 4.5.6.4: Funcions i utilitats

- **Canviar valor d'una variable:**

*(nomvariable) = (nouvalor o també operacions entre variables)*

- **Enviar comandament al sistema (LXTerminal):**

*system("comandament entre cometes")*

- **Establir condició**

```
if (condició) //Per exemple -num ==0, num<0, num>0, num !=0  
{  
-Programació que s'executarà només si es compleix la condició  
}
```

- **Si no es compleix la condició.. (després d'un if)**

```
else  
{  
-Programació que s'executarà (condició if no complerta)  
}
```

- **Crear bucle si..**

```
while (condició)  
{  
-Programació que s'executarà indefinidament si es compleix la  
condició fins que es deixi de fer-ho  
}
```

NOTA: Alerta en crear bucles d'aquest tipus perquè poden penjar el programa si no es resolen mai.

- **Aturar el programa durant..**

*usleep(microsegons)*

- **Llegir la primera paraula o número d'un text (.txt)**

Requereix la llibreria <fstream>: “*#include <fstream>*” i també escriure “*using namespace std*” al final de totes les llibreries

```
variableifstream.open(ubicació fitxer (carpeta) entre cometes)  
variableifstream >> variable  
variableifstream.close()
```

La variable ha d'acceptar el que contingui el text (una variable int no podrà acceptar lletres!)

Per guardar més paraules o lletres, s'han de guardar a variables diferents substituint el “variableifstream >> variable” per

“variableifstream >> variable1 >> variable2 >> variable 3 ...”

Cada variable guardarà una paraula o número corresponent al que aquest ocupi al document de text.

#### **4.5.6.5: Funcions i utilitats lligades a la interfície gràfica**

- **Canviar el text d'una label:**

*ui.>nomlabel.>SetText(QString)*

Quan es demana una QString, s'hi ha de posar una variable QString, un text concret entre cometes, o una variable numèrica posada de la següent manera:

*QString::number(nomvariablenumèrica)*

- **Ajustar la mida d'una label al text que hi té escrit**

```
ui.>nomlabel.>AdjustSize()
```

- **Amagar/Mostrar un objecte**

```
ui.>nomobjecte.>hide()            Amagar
```

```
ui.>nomobjecte.>show()          Mostrar
```

- **Desplegar finestra emergent informativa**

Requereix la llibreria <QMessageBox>: “*#include <QMessageBox>*”

```
QMessageBox::information(this,títol,text)
```

Títol i text són QStrings (vegeu 4.5.6.5.1)

- **Introduir text a una QLineEdit**

```
ui.>nomlineedit.>SetText(QString)
```

- **Llegir text d'una QLineEdit**

```
variable = ui.>nomlineedit.>text()
```

La variable ha de ser una QString o s'ha de convertir el valor que et donen:

```
variable = ui->nomlineedit.>text().toInt()
```

En aquest cas a una variable int, però també: *toFloat()*

## 4.6: Aplicació del control dels GPIO amb el QT Creator

Per comprovar el bon funcionament de la programació dels GPIO mitjançant el QT Creator, he fet dues petites pràctiques amb leds i pulsadors.

Una pràctica es tracta d'una simulació d'un semàfor de cotxes i vianants, i l'altre un petit joc de reacció amb els leds.

### 4.6.1: Simulació d'un semàfor

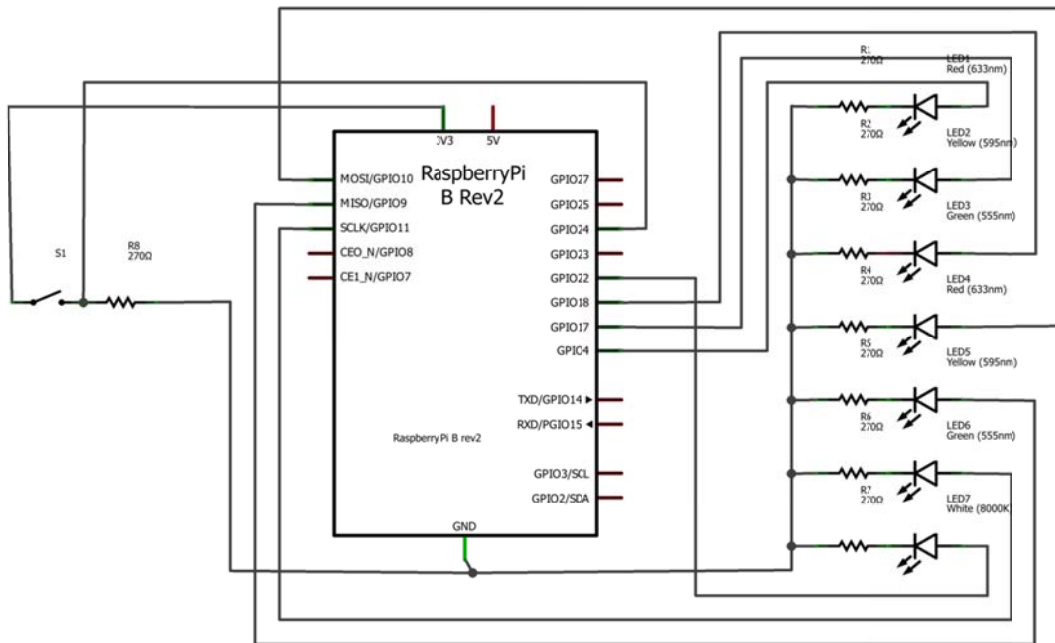
Es tracta d'una simulació d'un semàfor de cotxes i un de vianants d'un carrer unidireccional. El semàfor de vianants tindria un botó amb el qual poden posar verd el semàfor dels vianants. Els llums del semàfor van canviant segons com estiguin programats.

Cada cop que es prem el pulsador, s'activa un comptador que impedirà que l'acció del botó tingui efecte en un temps determinat i així evitant que s'abusi de parar el trànsit.

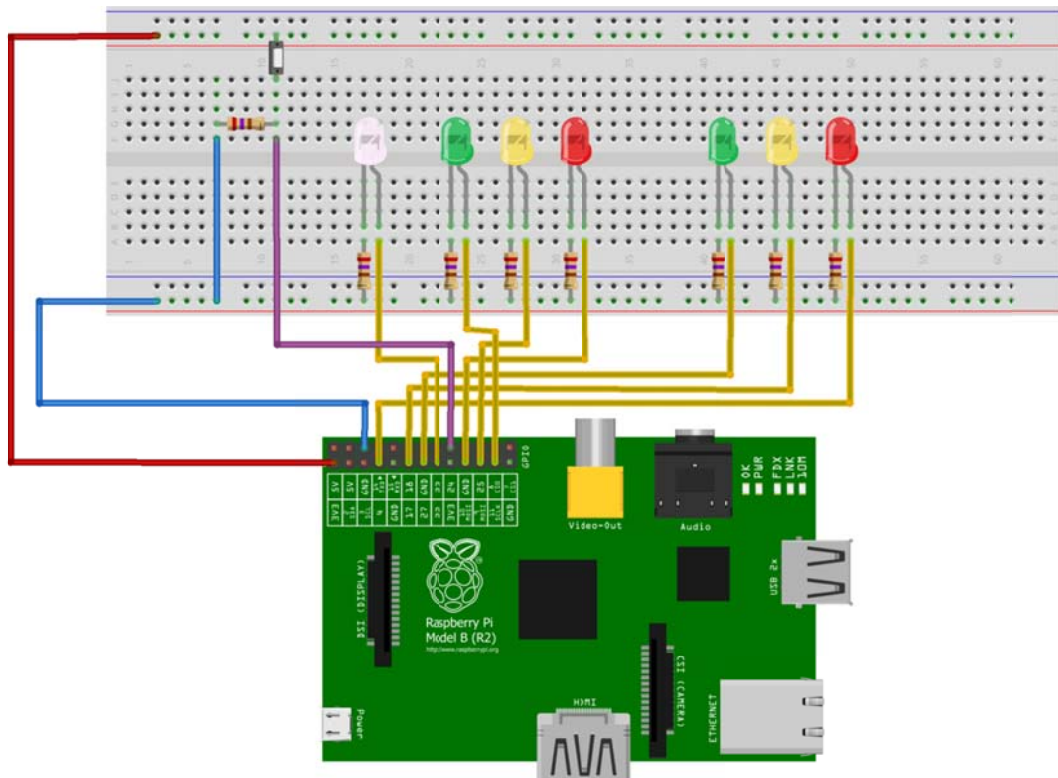
#### **Materials:**

- 7 LEDs, (vermell, groc, verd i blanc)
- 1 pulsador NO
- 8 resistències de 270 Ohms
- cables
- placa protoboard

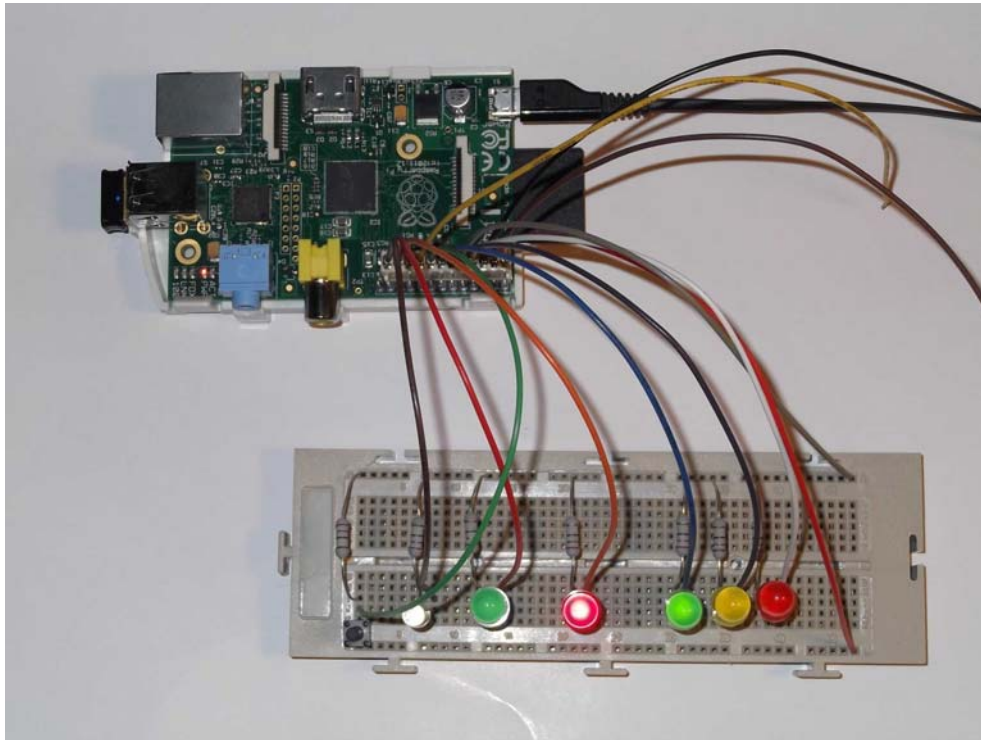
## Esquema del circuit:



## Representació del muntatge del circuit:



## Foto del Muntatge:



Exemple d'una fase del muntatge

## Programació en QT Creator (fitxer principal .cpp):

```
#include "finprincipal.h"
#include "ui_finprincipal.h"
#include <fstream>
using namespace std;

ifstream llegir;
int boto;
int parada;
int esperant;
int temps = 150;

FinPrincipal::FinPrincipal(QWidget *parent) :
    QMainWindow(parent),
    ui(new Ui::FinPrincipal)
{
    ui->setupUi(this);
    system("echo 4 > /sys/class/gpio/export");
    system("echo 17 > /sys/class/gpio/export");
    system("echo 27 > /sys/class/gpio/export");
    system("echo 22 > /sys/class/gpio/export");
    system("echo 10 > /sys/class/gpio/export");
    system("echo 9 > /sys/class/gpio/export");
    system("echo 11 > /sys/class/gpio/export");
}
```



```

FinPrincipal::~FinPrincipal()
{
    delete ui;
    system("echo 4 > /sys/class/gpio/unexport");
    system("echo 17 > /sys/class/gpio/unexport");
    system("echo 27 > /sys/class/gpio/unexport");
    system("echo 22 > /sys/class/gpio/unexport");
    system("echo 10 > /sys/class/gpio/unexport");
    system("echo 9 > /sys/class/gpio/unexport");
    system("echo 11 > /sys/class/gpio/unexport");
}

void FinPrincipal::on_Habilitar_clicked()
{
    system("echo out > /sys/class/gpio/gpio4/direction");
    system("echo out > /sys/class/gpio/gpio17/direction");
    system("echo out > /sys/class/gpio/gpio27/direction");
    system("echo out > /sys/class/gpio/gpio10/direction");
    system("echo out > /sys/class/gpio/gpio9/direction");
    system("echo out > /sys/class/gpio/gpio11/direction");
}

void FinPrincipal::on_Start_clicked()
{
    while (parada==0)
    {
        system("echo 1 > /sys/class/gpio/gpio27/value");
        system("echo 1 > /sys/class/gpio/gpio10/value");
        while ((boto==0) && (esperant == 0))
        {
            system("echo 1 > /sys/class/gpio/gpio27/value");
            system("echo 1 > /sys/class/gpio/gpio10/value");
            llegir.open("/sys/class/gpio/gpio22/value");
            llegir >> boto;
            llegir.close();
            temps++;
            usleep(100000);
        }
        temps++;
        usleep(100000);
        boto=0;
        esperant = 1;
        system("echo 1 > /sys/class/gpio/gpio11/value");
        if (temps>150)
        {
            system("echo 0 > /sys/class/gpio/gpio11/value");
            system("echo 0 > /sys/class/gpio/gpio27/value");
            system("echo 1 > /sys/class/gpio/gpio17/value");
            usleep(2000000);
            system("echo 0 > /sys/class/gpio/gpio17/value");
            system("echo 1 > /sys/class/gpio/gpio4/value");
            usleep(2000000);
            system("echo 0 > /sys/class/gpio/gpio10/value");
            system("echo 1 > /sys/class/gpio/gpio9/value");
            usleep(5000000);
            system("echo 0 > /sys/class/gpio/gpio9/value");
            system("echo 1 > /sys/class/gpio/gpio10/value");
            system("echo 0 > /sys/class/gpio/gpio4/value");
            temps = 0;
            esperant = 0;
        }
    }
}

```

}  
}

#### 4.6.2: Joc de reacció amb LEDs

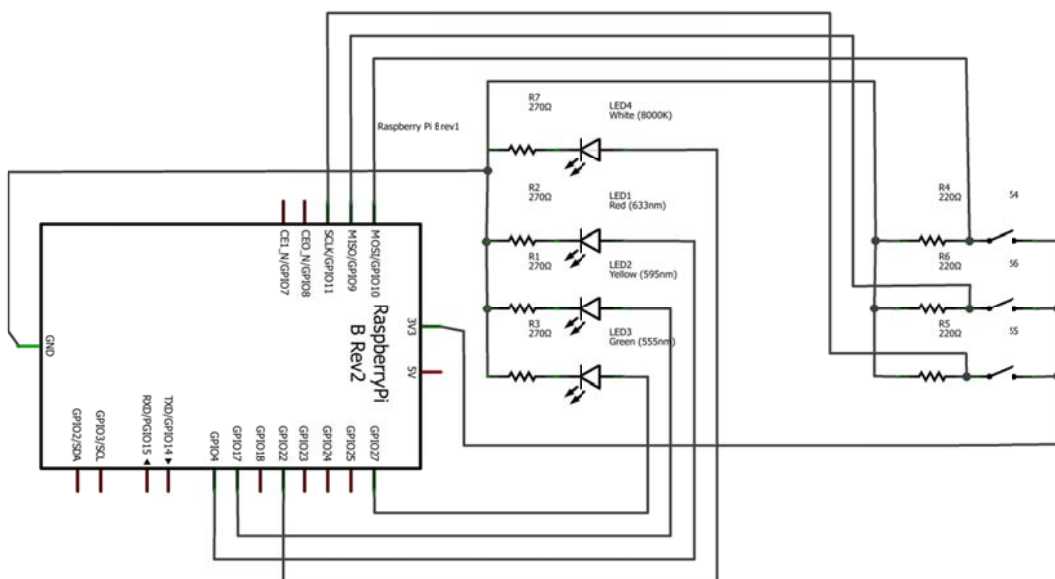
Aquest joc consisteix en prémer ràpidament el polsador assignat al led que s'encén. Un dels tres leds s'encén aleatòriament, i el programa comença a comptar. Quan es prem el polsador correcte s'avisava que s'ha encertat i en 2 segons s'encén un altre led (pot ser el mateix). El procés es repeteix tants cops com s'hagi posat al programa.

Quan acabes el joc, et diu el temps total que has tardat entre que s'encenia el led i premies el polsador correcte i també la mitjana dels temps, que equivaldria a la mitjana de temps de reacció

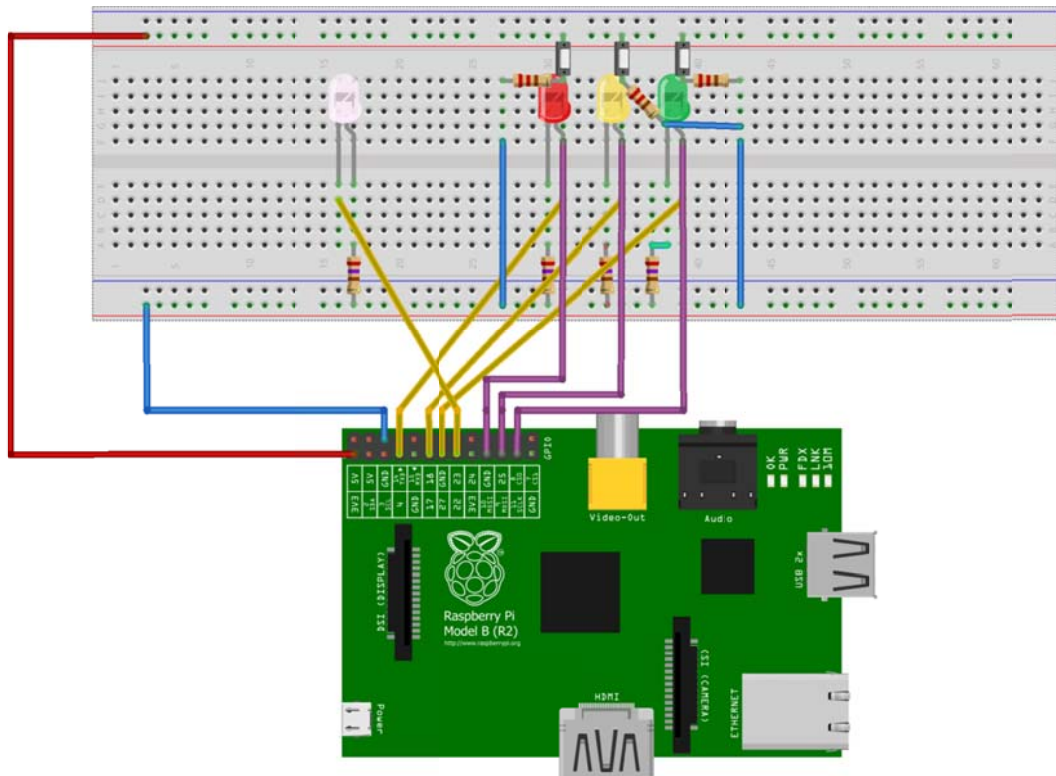
#### **Materials:**

- 4 LEDs, (vermell, groc, verd i blanc)
- 3 polsadors NO
- 7 resistències de 270 Ohms
- cables
- placa protoboard

#### **Esquema del circuit:**



## Representació del muntatge del circuit:



## Programació en QT Creator (fitxer principal .cpp):

```
#include "finestra.h"
#include "ui_finestra.h"
#include <cstdlib>
#include <QMessageBox>
#include <fstream>
using namespace std;
```

```
int rep;
int aleat;
int cont = 0;
int prem = 0;
int temps = 0;
int tot = 0;
ifstream boto;
```

```
Finestra::Finestra(QWidget *parent) :
    QMainWindow(parent),
    ui(new Ui::Finestra)
{
    ui->setupUi(this);
    system("echo 4 > /sys/class/gpio/export");
    system("echo 17 > /sys/class/gpio/export");
    system("echo 27 > /sys/class/gpio/export");
    system("echo 22 > /sys/class/gpio/export");
    system("echo 10 > /sys/class/gpio/export");
```

```

system("echo 9 > /sys/class/gpio/export");
system("echo 11 > /sys/class/gpio/export");
ui->Start->hide();
ui->Rein->hide();
}

Finestra::~Finestra()
{
    delete ui;
}

void Finestra::on_Habilitar_clicked()
{
    system("echo out > /sys/class/gpio/gpio4/direction");
    system("echo out > /sys/class/gpio/gpio17/direction");
    system("echo out > /sys/class/gpio/gpio27/direction");
    system("echo out > /sys/class/gpio/gpio22/direction");
    ui->Start->show();
    ui->Habilitar->hide();
    ui->Rein->show();
}

void Finestra::on_Start_clicked()
{
    rep = ui->lineEdit->text().toInt();
    while (cont<rep)
    {
        aleat = rand() % 3;
        if (aleat==0)
        {
            system("echo 1 > /sys/class/gpio/gpio4/value");
            system("echo 0 > /sys/class/gpio/gpio17/value");
            system("echo 0 > /sys/class/gpio/gpio27/value");
            while (prem==0)
            {
                boto.open("/sys/class/gpio/gpio10/value");
                boto >> prem;
                boto.close();
                temps++;
                usleep(1000);
            }
            system("echo 1 > /sys/class/gpio/gpio22/value");
        }
        if (aleat==1)
        {
            system("echo 0 > /sys/class/gpio/gpio4/value");
            system("echo 1 > /sys/class/gpio/gpio17/value");
            system("echo 0 > /sys/class/gpio/gpio27/value");
            while (prem==0)
            {
                boto.open("/sys/class/gpio/gpio9/value");
                boto >> prem;
                boto.close();
                temps++;
                usleep(1000);
            }
            system("echo 1 > /sys/class/gpio/gpio22/value");
        }
        if (aleat==2)
        {
            system("echo 0 > /sys/class/gpio/gpio4/value");
            system("echo 0 > /sys/class/gpio/gpio17/value");
        }
    }
}

```

```

system("echo 1 > /sys/class/gpio/gpio27/value");
while (prem==0)
{
    boto.open("/sys/class/gpio/gpio11/value");
    boto >> prem;
    boto.close();
    temps++;
    usleep(1000);
}
system("echo 1 > /sys/class/gpio/gpio22/value");
}
usleep(3000000);
system("echo 0 > /sys/class/gpio/gpio4/value");
system("echo 0 > /sys/class/gpio/gpio17/value");
system("echo 0 > /sys/class/gpio/gpio27/value");
system("echo 0 > /sys/class/gpio/gpio22/value");
prem = 0;
cont++;
tot = tot + temps;
temps = 0;
}
QMessageBox::information(this,"Resultat","Molt Be");
temps = tot/rep;
ui->total->setText(QString::number(tot) + "msec");
ui->mijana->setText(QString::number(temps) + "msec");
cont=0;
temps = 0;
tot = 0;
}

```

# Conclusions

Les primeres conclusions que trec del treball, són a nivell pedagògic, i relacionades amb tot allò que he anat aprenent mentre investigava el funcionament de la Raspberry i redactant el tutorial. Es tracta d'un mètode de recerca en temes informàtics.

El primer que has de fer és trobar la pregunta correcta, ja que si no és així pots estar molt de temps donant voltes sobre allà mateix. Un exemple d'això seria la manera d'obrir un fitxer de text del Qt Creator: com que el que volia era obtenir la resposta d'un pin com a entrada, vaig buscar i buscar com es podia rebre una resposta del sistema ( la funció recordeu que era "cat /sys/class...") mitjançant l'ordre "System" del Qt Creator, i emmagatzemar el valor en una variable. Fins que no vaig descobrir que el fitxer "value" era un simple fitxer de text, no se'm va acudir buscar com es podia obrir un fitxer de text amb el Qt Creator.

A continuació toca la part de prova i error, que consisteix en provar la informació que has trobat fins que allò que vulguis que vagi funcioni correctament i el més important, fins que s'entengui, perquè de no gaire serveixen aquelles webs que et donen una resposta sense cap explicació, ja que potser ho podràs aplicar en aquell cas concret però no en el general.

El tercer i últim pas és adaptar el procediment que has trobat a la teva manera perquè et sigui còmode, i que normalment, acaba sent una combinació personalitzada de conceptes i procediments. La prova final de que te n'has sortit, ha de res veure's en cor d'explicar el teu nou procediment.

D'aquest treball també es poden treure unes conclusions de contingut. La Raspberry Pi és un aparell que et sorprèn amb totes les seves utilitats i tot allò que pot fer malgrat la seva mida i el petit processador que porta. El primer impactant de la Raspberry Pi és la seva capacitat gràfica, de poder reproduir imatges i vídeos en alta definició a través de la seva sortida HDMI. Llavors vénen tots aquells programes que fan de la nostra Raspberry un autèntic ordinador complet.

Finalment, queda comprovada la possibilitat de fer de la Raspberry una Placa de control de circuits com si fos una placa arduino, si més no, s'hi acostava. Segurament és menys pràctica però té un munt d'opcions en l'accessibilitat al programa que una arduino no té.

## Opcions de seguir endavant

Aquest treball es podria continuar, creant una aplicació pràctica a la realitat mitjançant els GPIO, crear algun aparell que tingui una finalitat real i aprofitar les prestacions en imatge i so per millorar-ho. Accés al programa per internet i a través del mòbil.

En el camp de continuar investigant la Raspberry, s'hi podria incloure l'opció de mirar si es poden convertir algunes entrades/sortides GPIO en analògiques, mirar si recollint dades de sensors es pot vincular directament a una base de dades, i també l'apartat de crear un servidor al núvol d'emmagatzematge de dades a la nostra Raspberry.

# Webgrafia

## Informació General:

wikipedia.org

[es.wikipedia.org/wiki/Raspberry\\_Pi](http://es.wikipedia.org/wiki/Raspberry_Pi)

[www.raspberrypi.org](http://www.raspberrypi.org)

## Tutorials Generals:

www.aquihayapuntes.com

<http://www.aquihayapuntes.com/%C3%ADndice-sistemas-embebidos/raspberry-pi-primeros-pasos.html>

deeiivid.wordpress.com

<http://deeiivid.wordpress.com/2013/02/23/guia-completa-raspberry-pi-espanol/>

## Escritori remot:

www.electroensaimada.com

<http://www.electroensaimada.com/escritorio-remoto.html>

## Qt Creator:

dis.um.es

<http://dis.um.es/~ginesgm/files/doc/pav/quion1.pdf>

## GPIO:

www.aquihayapuntes.com

<http://www.aquihayapuntes.com/%C3%ADndice-sistemas-embebidos/raspberry-pi-programando-el-puerto-gpio-con-qt-creator.html>

hertaville.com

<http://hertaville.com/2012/11/18/introduction-to-accessing-the-raspberry-pis-gpio-in-c/>

projects.drogon.net

<https://projects.drogon.net/raspberry-pi/wiringpi/pins/>

## Programació C++:

Diverses pàgines de respostes de stackoverflow.com

Com que la metodologia d'aquest treball inclou una recerca d'informació molt general per internet, hi ha moltes pàgines on he buscat i que no surten a la webgrafia perquè només n'he extret algun concepte concret que feia falta en un moment en concret.