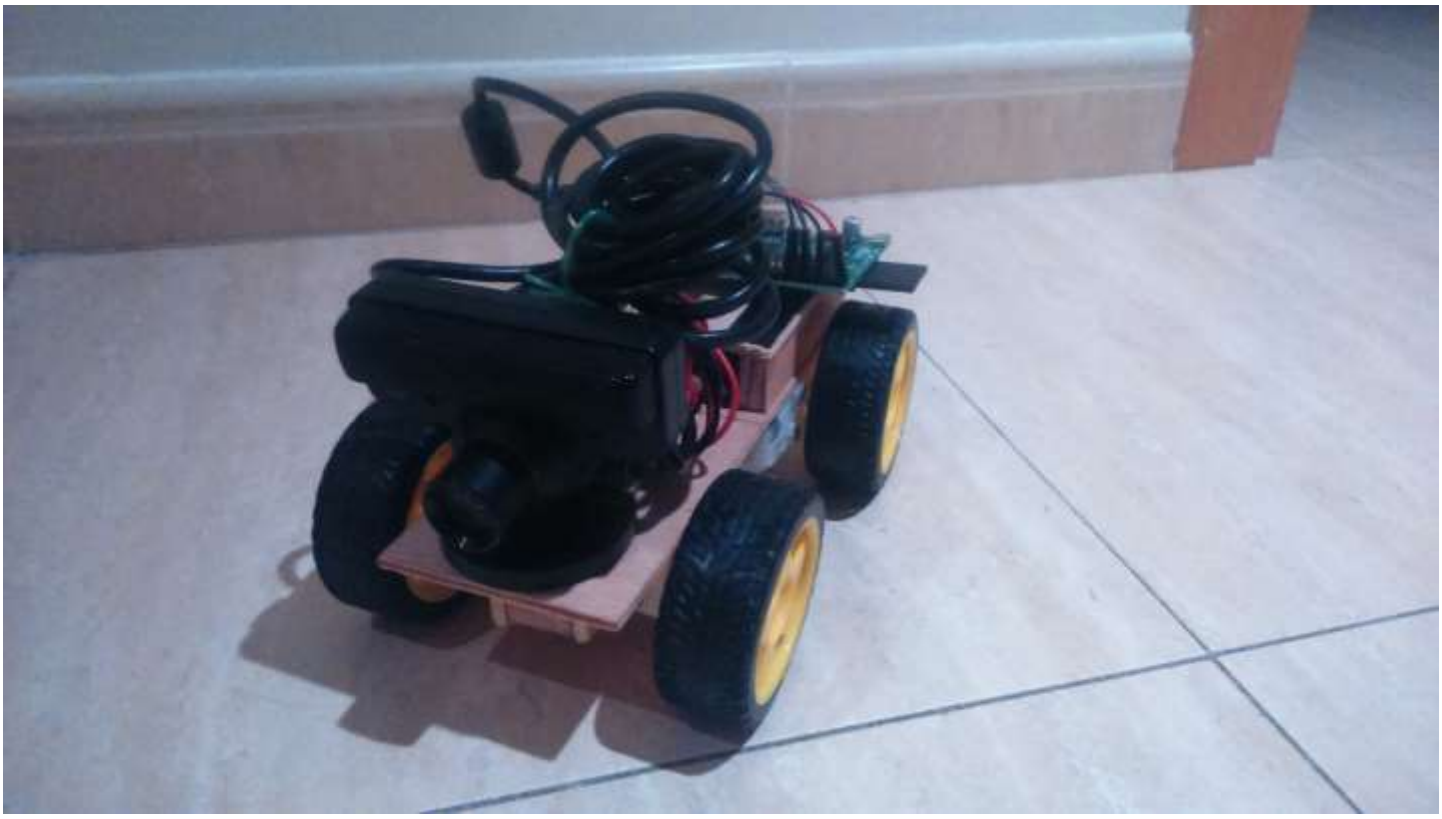


Programació d'un vehicle telecomandat per WiFi amb la Raspberry Pi

Data: 19 de novembre de 2014

Índex

Introducció 4
1.- Cos teòric 5
1.1.- Materials 5
1.2.- Raspberry Pi 5
1.2.1.- Hardware 6
1.2.1.2.- Pins GPIO 7
1.2.2.- Software 8
1.2.2.1.- Sistemes operatius 9
1.2.3.- Especificacions tècniques 10
1.3.- Python 11
1.3.1.- Una mica d'història 11
1.3.2.- Característiques 11
1.3.3.- Elements bàsics de programació 12
1.4.- PHP 13
1.4.1.- Que és PHP? 13
1.4.2.- Com funciona? 14
1.4.3.- Elements bàsics de programació 15
1.4.4.- Que es necessita per començar? 16
1.5.- Robòtica 16
1.5.1.- Classificació dels robots 16
1.6.- Connexió SSH 17
2.- Cos pràctic 18
2.1.- Muntatge del cotxe 18
2.2.- Instal·lant el Sistema Operatiu Raspbian 20
2.2.1.- Preparant la tarja SD 22
2.3.- Connectant la Raspberry Pi a través de SSH 25
2.4.- Instal·lant i configurant un servidor web a la Raspberry Pi 29
2.5.- Instal·lant i configurant NO-IP a la Raspberry Pi 33
2.6.- Controlant el GPIO de la Raspberry Pi amb Python 35
2.7.- Accedint al GPIO de la Raspberry Pi des de PHP 36
2.7.- Instal·lant i configurant la càmera 40
2.8.- Aplicació web per controlar el GPIO 36

3.- Conclusions 41
4.- Agraïments 42
5.- Webgrafia 43

Introducció

Per què aquest i no un altre?

El treball de recerca porta moltes hores de treball i molta feina, per tant val la pena triar un tema amb el qual et sentis còmode i t'interessi, per tal que t'obligui a cercar informació, t'aporti nous coneixements i et permeti un enriquiment personal.

Al ser estudiants de la branca tecnològica del batxillerat, vam pensar que el nostre treball hauria d'estar relacionat amb el que estàvem estudiant, ja que en teoria és el que ens agrada.

Ens vam decidir per un projecte de robòtica desprès de moltes xerrades i de compartir punts de vista i opinions entre el nostre tutor i nosaltres. Dic nosaltres ja que aquest treball ha estat realitzat entre dos persones, la raó d'aquesta decisió és que el projecte es podia separar en 2 parts ben diferenciades, la part del muntatge, que té més a veure amb l'electrònica, i la part de programació, que té més a veure amb la informàtica.

És cert que no sabíem ni per on començar a treballar en el cotxe, però gràcies al nostre tutor, que ens va orientar i ens va mostrar el camí que hauríem de seguir, vam iniciar el projecte.

Ens vam decidir per utilitzar una Raspberry Pi com a "cervell" del cotxe robot, ja que es tracta d'un ordinador petit, econòmic, i amb un potencial enorme, el qual ens va fascinar i vam voler investigar-ho a fons.

Estructura

He estructurat el treball en els següents apartats: el primer apartat és un cos teòric, on trobem una conceptualització bàsica i essencial per a poder endinsar-se al tema i poder obtenir un determinat coneixement sobre la robòtica, la programació bàsica, i els llenguatges de programació utilitzats en aquest projecte.

El segon apartat fa referència a la part pràctica, on expliquem el nostre disseny i la programació que hem realitzat per tal que el cotxe funcioni correctament.

Limitacions

Les limitacions que vam tenir a l'hora de realitzar el nostre projecte van ser:

- Va haver limitacions importants en el tema d'horaris per poder combinar el nostre temps lliure sense classes amb el del nostre tutor.
- En segon lloc, una limitació molt gran que vam tenir, i que ha endarrerit molt el nostre treball, ha estat que nosaltres precisàvem d'estar a l'institut per poder fer el muntatge del cotxe, i com que aquest any el treball de recerca s'entrega abans, vam haver d'esforçar-nos per tal detenir-lo acabat per la data assignada.

1.- Cos teòric

1.1.- Materials

Per realitzar la part pràctica de programació d'aquest projecte hem hagut d'utilitzar o de fabricar aquests materials o dispositius:

- Raspberry Pi
  - Carregador de móvil (*smartphone*)
  - Cable ethernet per tenir internet
- Driver (muntatge explicat al treball del meu company)
- Camara USB
- Bateria lipo
- Xassís cotxe (muntatge explicat al treball del meu company)

1.2.- Raspberry Pi

La Raspberry Pi és un ordinador en una placa reduïda de baix cost desenvolupada per la Fundació Raspberry Pi al Regne Unit, amb l'objectiu d'estimular l'ensenyament informàtic a les escoles.

Es tracta d'un aparell de maquinari lliure, el qual vol dir que les especificacions i els diagrames del qual són lliures d'accés per a tothom, igual que el seu programari.

Porta un *"System-on-a-chip"*, que conté un processador central (CPU) ARM1176JZF-S a 700MHz, el qual porta unes configuracions de *"Turbo"* que et permeten fer *overlock* de fins a 1GHz, un processador gràfic (GPU) VideoCore IV i 512MB de RAM. Al no portar cap disc dur, s'utilitza una tarja Micro SD com a tal, en la que carregarem el sistema operatiu i el farem funcionar.

A la Raspberry Pi podrem treballar amb sistemes operatius amb base Linux i modificats pel seu correcte funcionament en el microordinador. Encara que suporta més llenguatges de programació, el més utilitzat és Python.

1.2.1.- Hardware

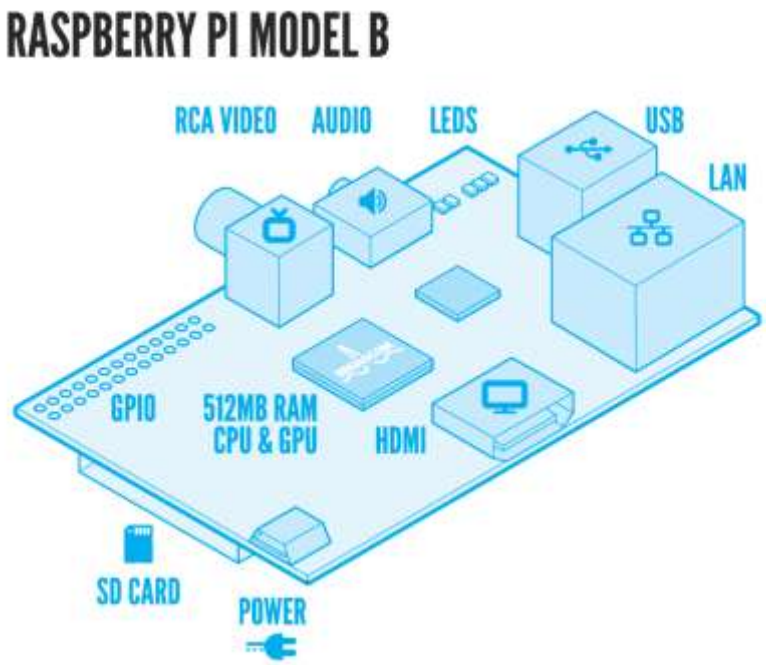
Hi ha dos models de Raspberry Pi, el model A i el model B, cadascun amb unes característiques diferents de l'altre.

El model B, el de més "alta gama", compta amb 512MB de RAM, 2 ports USB, un adaptador Ethernet, un adaptador per HDMI, un connector RCA i els ports GPIO. Tot això equival a la pràctica a un ordinador, amb unes capacitats gràfiques similars a una XBOX de Microsoft, amb la possibilitat de reproduir vídeos a 1080p.

Els ports GPIO ens permeten fer tot tipus "de interfaces" hardware, com el control d'altres dispositius, encendre i apagar de llums, control domòtic, etc.

El model A, a diferència del B, compta amb 256MB de RAM, 1 sol port USB i no té cap adaptador Ethernet, per tant per connectar-te a internet necessites alguna antena Wi-Fi per via USB o un adaptador USB-Ethernet.

Fa poc es va posar a la venta el nou model B+, amb més ports USB i un processador i una RAM millorades.



1.2.1.2.- Pins GPIO

El nom de GPIO ve de **G**eneral **P**urpose **I**nput **O**utput, i són pins genèrics que poden ser d'entrada o sortida i el seu comportament pot ser programat a través d'un software.

Els pins GPIO no tenen cap propòsit definit, sinó que l'usuari se'l dona depenent el dispositiu que els porti, i de les seves necessitats. Normalment s'utilitza per afegir línies de control digital de diversos accessoris per ampliar la funció del dispositiu que els porti.

Característiques:

- Poden ser configurats com a entrada o com a sortida. Hi ha un registre per cada pin que es diu DDR (**D**ata **D**irection **R**egister), el qual si s'escriu un 1, s'estableix com a output, i si s'escriu un 0 com a input.
- Els pins poden ser activats o desactivats.
- Els valors d'entrada es poden llegir (per lo general el nivell alt té un valor 1 i el nivel baix de 0).
- Els valors de sortida poden ser de lectura o d'escriptura.

Els GPIO a la Raspberry Pi

A través dels GPIO es poden encendre i apagar elements electrònics, respondre a events i prendre una varietat de mesures. Això és el que s'anomena *"interfacing"*.

Els GPIO poden ser entrades i sortides, les entrades a un ordinador podrien ser un teclat, i les sortides uns altaveus. Amb els GPIO podrem controlar un altre tipus de coses connectades a l'ordinador.

Tipus de senyals en components electrònics:

- Senyal digital: per exemple quan premem un botó, l'ordinador no sap la força, només sap que l'has premut, per tant representa el seu estat, si està premut un 1 i si no ho està un 0.
- Senyal analògica: és una senyal que varia de forma contínua al llarg del temps. La majoria de les senyals que representen magnituds físiques (temperatura, lluminositat, humitat, etc) són senyals analògiques. Les senyals analògiques poden prendre tots els valors possibles d'un interval. A la Raspberry Pi si volem introduir senyals analògiques s'hauran de passar a digitals.

Cada pin GPIO pot ser utilitzat com a entrada o com a sortida. El número de pins que té la Raspberry és de 26 (2 x 13) i tenen varies utilitats, alimentació 3v3 y 5v, I2C, i més usos de propòsit general.

Aquests pins no són *"plug and play"* i s'ha de tenir cura per evitar connecions incorrectes. Per exemple els pins 3V3 no toleren 5V, i no hi ha protecció contra sobre-voltatge, per tant es podria cremar la Raspberry simplement connectant un pin a 5V.

Aquí hi ha una taula amb els pins de a capçalera P1:

BCM2835	Conector GPIO P1		BCM2835
3.3v	1	2	5v
I2C SDA GPIO 2	3	4	5v
I2C SCL GPIO 3	5	6	GND
GPIO 4 GPCLK	7	8	UART TXD GPIO 14
GND	9	10	UART RXD GPIO 15
GPIO 17	11	12	PWM GPIO 18
GPIO 27	13	14	GND
GPIO 22	15	16	GPIO 23
3.3v	17	18	GPIO 24
SPI MOSI GPIO 10	19	20	GND
SPI MISO GPIO 9	21	22	GPIO 25
SPI SCLK GPIO 11	23	24	SPI CEO GPIO 8
GND	26	26	SPI CE1 GPIO 7
	Revisión 2		

1.2.2.- Software

La Raspberry Pi funciona amb sistemes operatius basats en el nucli Linux, tals com Raspbian, PiDora, etc. El Raspbian, és un sistema operatiu basat en Debian i modificat per que el seu us sigui el correcte amb el microordinador.

També executa els sistemes operatius ARM, ja que els seus requisits son molt baixos, i la Raspberry Pi els compleix sense cap problema.

1.2.2.1.- Sistemes operatius

Hi ha una llista de sistemes operatius complerts que han estat programats per que funcionin amb la Raspberry Pi:

- AROS
- Linux
    - Android
    - Arch Linux ARM
    - Debian Whezzy Soft-Float
    - Firefox OS
    - Gentoo Linux
    - Google Chromium OS
    - Kali Linux
    - Open webOS
    - PiBang Linux
    - Pidora
    - QtonPi
    - **Raspbian**
    - Slackware ARM
- Plan 9 from Bell Labs
- RISC OS 5
- Unix
    - FreeBSD
    - NetBSD

He utilitzat Raspbian com a sistema, ja que és el més pràctic i fàcil d'utilitzar.

1.2.3.- Especificacions tècniques

Model A	Model B	Model B+
Preu:	32 €	35 €
SoC:	Broadcom BCM2835 (CPU + GPU + DSP + SDRAM)	
CPU:	ARM1176JZF-S a 700 MHz	
GPU:	Broadcom VideoCore IV, OpenGL ES 2.0, descodificador 1080p30 H.264	
Memòria (SDRAM):	256 MiB (compartits amb la GPU)	512 MiB (compartits amb la GPU)
PortsUSB 2.0:	1	2 (vía hub USB integrat) 4
Sortida de vídeo:	Connector RCA, HDMI	
Sortida d'àudio:	connector de vídeo compost, connector 3,5 mm jack, HDMI	connector jack de 3,5 mm i 4 pols per àudio i vídeo, HDMI (rev 1.3 & 1.4)
Emmagatzematge integrat:	SD / MMC / ranura per a SDIO	microSD
Connectivitat de xarxa:	Cap	10/100 Ethernet (RJ45)
Perifèrics de baix nivell:	24 pins GPIO, SPI, I²C, UART	40 × GPIO
Rellotge de temps real:	Cap	
Consum energètic:	300 mA (1.5 W)	700 mA, (3.5 W) 600 mA (3.0 W)
Font d'alimentació:	5 V via Micro-USB o GPIO header	
Dimensions:	85,60mm × 53,98mm × 17 mm (3,370 × 2,125 polsades)	85 mm × 56 mm × 17 mm

Sistemes operatius suportats:	Debian GNU/Linux, Fedora, Arch Linux
Sistemes operatius no suportats:	RISC OS (shared source)

1.3.- Python

Python és un llenguatge de programació interpretat, multiparadigma (soporta la programació orientada a objectes) i multiplataforma d'alt nivell, el qual vol dir que la seva sintaxi és clara i senzilla d'aprendre i amb molt de potencial, alguns diuen que programar en Python és gairebé com escriure en anglès.

És de codi obert, el qual et permet fer modificacions al teu gust i personalitzar-lo tot el que vulguis.

1.3.1.- Una mica d'història

Python va ser creat a finals dels vuitanta per Guido van Rossum (ex-treballador de Google i actual treballador de Dropbox), el qual continua sent qui escull el camí que ha de seguir el llenguatge de programació.

El seu llançament va ser al 1991 amb la versió 0.9.0, la qual no incorporava totes les funcions que té ara. Durant tot aquest temps des de la seva primera aparició en públic, s'han llançat moltes noves versions amb noves actualitzacions i funcions, fins la versió actual, la 3.1.

1.3.2.- Característiques

Aquestes són les característiques del llenguatge:

- **Propòsit general.** Es poden crear tot tipus de programes amb Python, no ha estat creat específicament per a la web com PHP o per aplicacions d'ordinador, abasta un gran número d'utilitats.

- **Multi plataforma.** Es va desenvolupar per Unix, encara que actualment el pots instal·lar i utilitzar en tot tipus de plataformes, Linux, Winows i Mac.

- **Interpretat.** Vol dir que no s'ha de compilar el codi abans de la seva execució, encara que realment si que es compila, però d'una manera transparent (no s'ha d'utilitzar cap compilador, quan s'executa per primera vegada es generen uns bytecodes que es desen al sistema i serveixen per futures execucions).

- **Funcions i llibreries.** Disposa de moltes funcions incorporades pel tractament de strings, números, arxius, etc. A més, existeixen moltes llibreries que ens

permeten fer coses com programar finestres, sistemes de xarxa, crear fitxers en .zip, etc.

- **Sintaxi clara.** La sintaxi de Python és de les millors sintaxis que hi ha en els llenguatges de programació, ja que no fan falta claus, ni res per l'estil, només fa falta agrupar els codis amb el tabulador. Per tant tots els programadors adapten la mateixa manera de programar, i tothom pot entendre un codi que no ha programat.

1.3.3.- Elements bàsics de programació

Alguns elements bàsics de programació que ens permetran programar en Python són:

- **Definir variables.** Una variable pot ser una cadena de text, un número, etc. I es defineixen d'aquesta manera:
  variable = valor_de_la_variable
  Exemples:
  - Número: *edat = 16*
  - Cadena de text: *cadena = "Hola"*
  - Booleans: *major_edat = false* | *menor_edat = true*

- **Operadors aritmètics.**

Símbol	Significat	Exemple	Resultat
+	Suma	*x = 10 + 2*	x = 12
-	Resta	*x = 10 - 2*	x = 8
*	Multiplicació	*x = 10 * 2*	x = 20
**	Exponent	*x = 10 ** 2*	x = 100
/	Divisió	*x = 10 / 2*	x = 5
//	Divisió entera	*x = 12.5 / 2*	x = 6.0
%	Mòdul	*x = 27 % 4*	x = 3

- **Operadors lògics.**

Símbol	Utilitat
&& / AND	S'utilitza per verificar que dues o més variables compleixin una norma, normalment en una condició *"if/else"*.
|| / OR	S'utilitza per verificar que alguna de les dues o més variables compleix una norma, normalment en una condició *"if/else"*.

Exemple:
if $variable_a = 2 and $variable_b = 4:
#farà una acció si les dues variables compleixen la seva norma (que variable_a sigui 2 i que variable_b sigui 4)
else:
#farà una altra

1.4.- PHP

PHP és un llenguatge de programació de la part del servidor, el qual vol dir que el codi fa el procés de la petició de l'usuari interpretant un *"script"* de la part del servidor web per tal de generar una pàgina HTML dinàmica.

És un llenguatge de programació flexible, potent i d'alt rendiment, el qual ha permès a múltiples pàgines web amb molta demanda de tràfic com FaceBook, Twitter, Google (fa uns anys, ja que ara utilitza Python), etc, utilitzar-ho com a tecnologia del servidor.

La majoria de espais web d'internet amb contingut dinàmic i connexions amb bases de dades utilitzen PHP com a llenguatge de part del servidor, ja que gairebé tots els servidors web, sistemes operatius i plataformes poden interpretar-ho de manera correcta.

La semblança que té la sintaxi de PHP amb altres llenguatges de programació com C, C++ o Perl, ha permès que molts de programadors puguin aprendre d'una manera ràpida i desenvolupar sistemes complexes en poc temps.

1.4.1.- Que és PHP?

PHP és un llenguatge de programació que busca brindar la possibilitat als creadors de pàgines web de dissenyar webs dinàmiques d'una forma senzilla i ràpida, encara que PHP té unes possibilitats molt més grans que fer una pàgina web, com les que veurem en aquest treball.

Actualment totes les pàgines són actualitzades d'una forma freqüent, de manera que si hagéssim de fer-ho per fitxers HTML, estaríem tot el dia pujant i actualitzant els fitxers del servidor per FTP. En canvi, en una pàgina web dinàmica això no passa, ja que la informació del lloc web està emmagatzemada en una base de dades. Cada cop que carreguem la pàgina, el codi PHP s'encarrega de cercar el contingut a la base de dades i mostrar-ho a través d'HTML, i cada cop que volguéssim pujar informació a la base de dades, només ens faria falta introduint-la fent servir un formulari.

Podem posar un exemple d'un bloc de notícies:

- **Estàtic:** el redactor/a o redactors/es d'aquest bloc, haurien de redactar les notícies en un programa com Word, per després amb un programa de disseny dissenyar la nova pàgina amb la notícia més recent. Un cop dissenyat, hauria de pujar el fitxer fent servir un client FTP al seu servidor.

- **Dinàmica:** els redactors/es del bloc, tindrien accés a un tauler d'administració, en el qual hi hauria un formulari per redactar la notícia, que en el moment que s'envïi, s'afegiria a la base de dades, i per tant, la pròxima visita al bloc, ja tindria aquesta notícia en primera plana.

PHP ens permet crear sistemes per tal de facilitar les nostres tasques o de fer-les automatitzades.

1.4.2.- Com funciona?

Per funcionar PHP necessita el seu motor de scripting (script engine) que executa les parts de codi abans que el servidor envií la resposta a l'usuari.

Actualment s'ha arribat a la versió 5.6 del motor PHP, el qual funciona pràcticament en tots els servidors de la xarxa.

Quan un usuari sol·licita una pàgina PHP, el motor executa el codi que està en aquesta pàgina. Mentre es realitza l'execució, el codi dóna unes informacions en format HTML. Finalment s'envia l'arxiu (completament en format html) a l'usuari.

Si intenteu visualitzar la font d'una pàgina en PHP podeu comprovar que no apareix cap línia del cod. L'avantatge és precisament aquest, cap usuari extern, exclòs el webmaster, pot accedir al codi i modificar-lo. Per a l'usuari extern, la pàgina en PHP és exactament igual que una pàgina qualsevol en HTML, encara que existeix la injecció SQL (iSQL), però això és un altre tema més complicat.

1.4.3.- Elements bàsics de programació

Alguns elements bàsics de programació que ens permetran programar en PHP són:

- **Definir variables.** Una variable pot ser una cadena de text, un número, etc. I es defineixen d'aquesta manera:
  $variable = valor_de_la_variable;
  Exemples:
  - Número: *$edat = 16;*
  - Cadena de text: *$cadena = "Hola";*
  - Booleans: *$major_edat = false;* | *$menor_edat = true;*

- **Operadors Aritmètics.**

Símbol	Significat	Exemple	Resultat
+	Suma	*x = 10 + 2*	x = 12
-	Resta	*x = 10 - 2*	x = 8
*	Multiplicació	*x = 10 * 2*	x = 20
/	Divisió	*x = 10 / 2*	x = 5

- **Operadors lògics.**

Símbol	Utilitat
&& / AND	S'utilitza per verificar que dues o més variables compleixin una norma, normalment en una condició *"if/else"*.
|| / OR	S'utilitza per verificar que alguna de les dues o més variables compleix una norma, normalment en una condició *"if/else"*.

Exemple:

```
if($variable_a == 2 && $variable_b == 4){
    //farà una acció si les dues variables compleixen la seva norma (que
variable_a sigui 2 i que variable_b sigui 4)
}else{
    //farà una altra
}
```

1.4.4.- Que es necessita per començar?

Per començar a programar en PHP necessites:

1- Un editor de text, pot ser el Bloc de Notes, encara que es recomana utilitzar algun especial que resalti la sintaxi, com Notepad++ (gratuït) o Sublime Text 3 (amb llicència).
2- Necessites tenir un servidor web instal·lat, el més utilitzat és l'Apache (gratuït), un intèrpret PHP i si vols, una base de dades.

A partir d'aquí, ja tens el necessari per programar i fer aplicacions web amb PHP.

1.5.- Robòtica

La robòtica és la branca de la tecnologia que estudia el disseny i la construcció de màquines capaces d'exercir tasques repetitives, en les quals es necessita una gran precisió, són perilloses per l'ésser humà o són irrealitzables sense la intervenció d'una màquina.

1.5.1.- Classificació dels robots

Els robots es poden classificar segona la seva cronologia, o segons la seva arquitectura.

Segons la seva cronologia

- **1a generació.** Anomenats "manipuladors". Són multifuncionals amb un sistema de control manual, de seqüència fixa o de seqüència variable.

- **2a generació.** Se'ls anomena "robots d'aprenentatge". Repeteixen els moviments que ha realitzat prèviament un operador humà.

- **3a generació.** Són robots amb control per sensors.

- **4a generació.** Anomenats "robots intel·ligents". Similars als anteriors, però a més envia informació sobre l'estat del procés.

Segons la seva arquitectura

L'arquitectura d'un robot, que es tracta del seu tipus de configuració general, pot ser metamòrfica. El grau de metamorfisme d'un robot és la capacitat que té un robot de canviar la seva configuració a través d'ell mateix. La subdivisió dels robots en base a la seva arquitectura és la següent:

- **Poliarticulats.** Robots amb forma i configuració molt diversa, els quals tenen un determinat grau de "llibertat" que no és molt gran. En aquest grup es troben robots industrials, cartesians.

- **Mòbils.** Robots amb gran capacitat de moviment, amb una plataforma i un motor que el faci rodar i moure's. Es mouen per telecomandament o a través de la informació que reben dels seus sensors.

- **Androides.** Intenten reproduir total o parcialment el comportament cinemàtic de l'ésser humà. Actualment estan molt poc evolucionats i estan destinats a l'estudi. Un dels principals problemes és la locomoció bípeda.

- **Zoomòrfics.** Imiten a diversos éssers vius, i s'agrupen en caminadors i no caminadors. Els no caminadors estan molt poc evolucionats, mentre que els caminadors estan en un continu estudi pel que podrien ser veritables vehicles terrens, pilotats o autònoms, que podrien superar superfícies molt accidentades.

- **Híbrids.** La seva estructura és una combinació de dues o més estructures citades anteriorment.

  Poden considerar-se híbrids els robots formats per una juxtaposició entre un carro i un braç. També es troben en la mateixa situació els robots antropomorfes, que és el cas dels robots personals.

1.6.- Connexió SSH

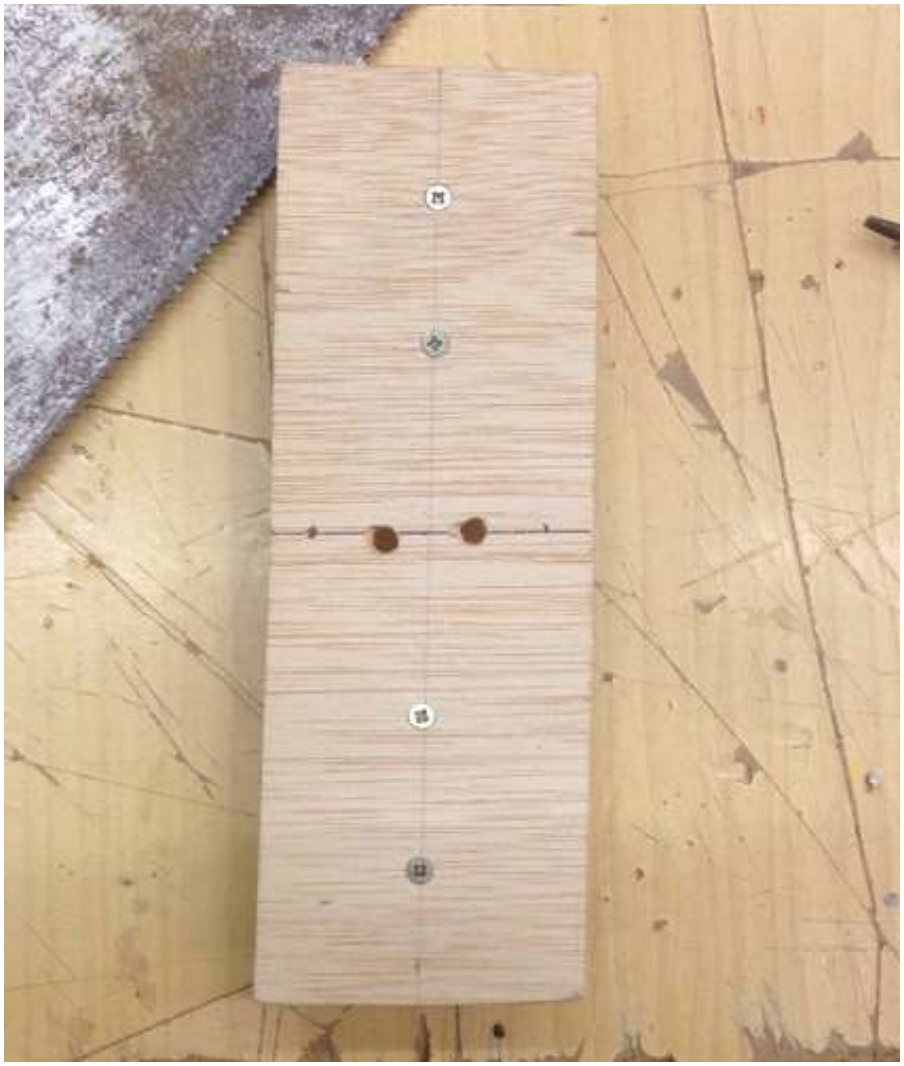
Per tal de accedir a la Raspberry Pi sense necessitar d'un cable HDMI, he hagut d'utilitzar un client SSH.

SSH (**S**ecure **SH**ell) és un protocol que ens permet connectar-nos a un equip de manera remota a través d'una xarxa LAN/WAN. Permet gestionar per complert l'ordinador, de manera que tot el que executem es darà sobre aquest ordinador, i es veurà en el que estem localment. Per exemple, si a través de l'ordinador local (amb el que estem treballant) reproduïm una cançó per SSH, el so sortirà per l'ordinador remot (al que estem connectats).

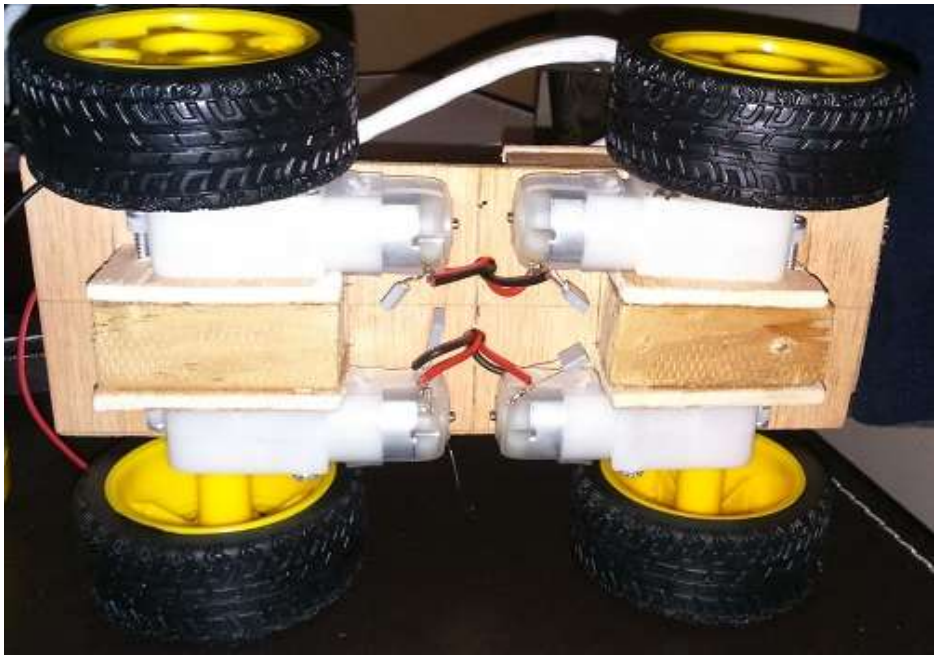
2.- Cos pràctic

2.1.- Muntatge del cotxe

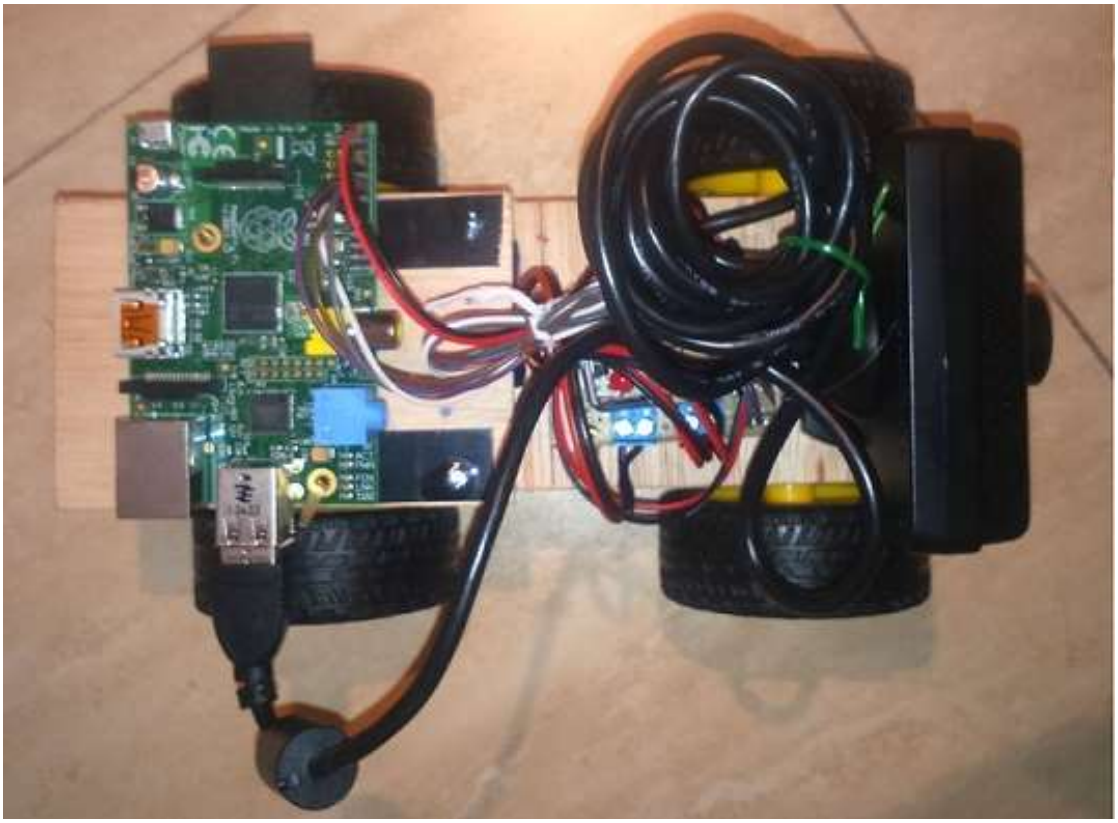
El primer pas que vam haver de realitzar va ser la construcció del xassís del cotxe, mitjançant fustes i cargols. En aquest pas, el cotxe es veia així:

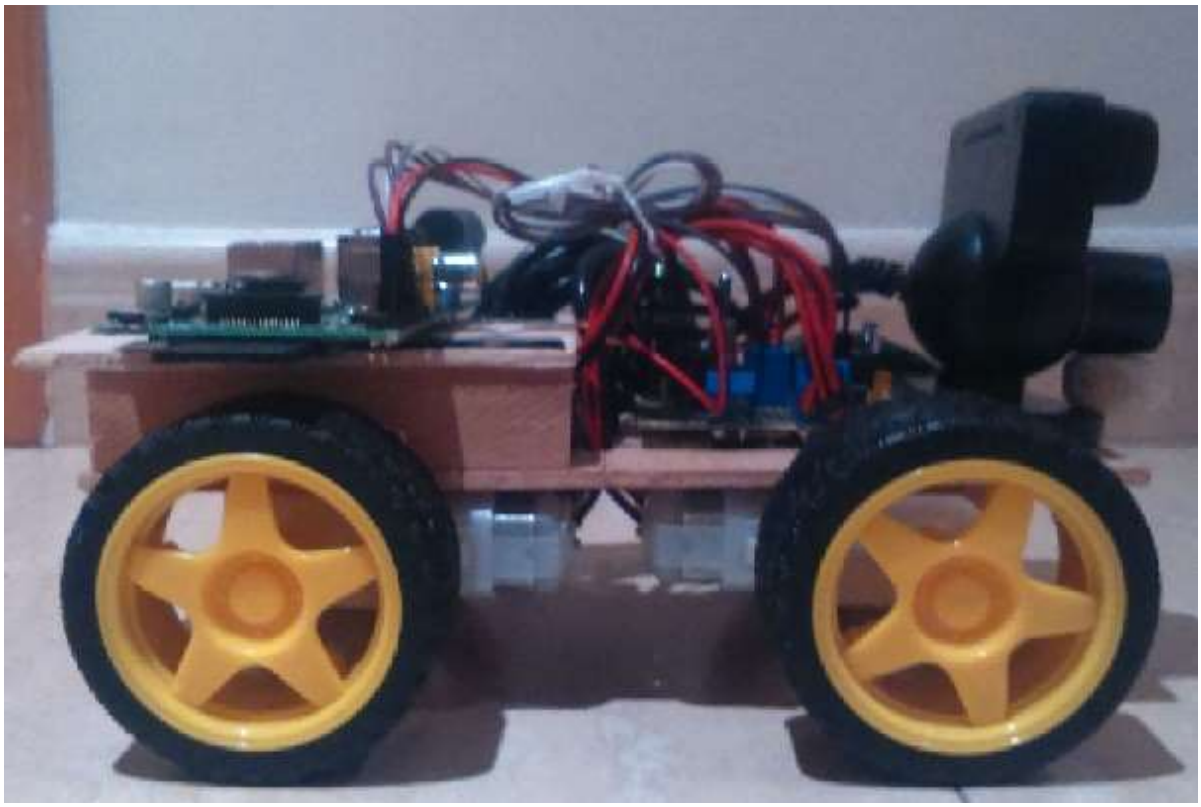


El següent pas va ser col·locar els motors i les rodes, i va quedar així:



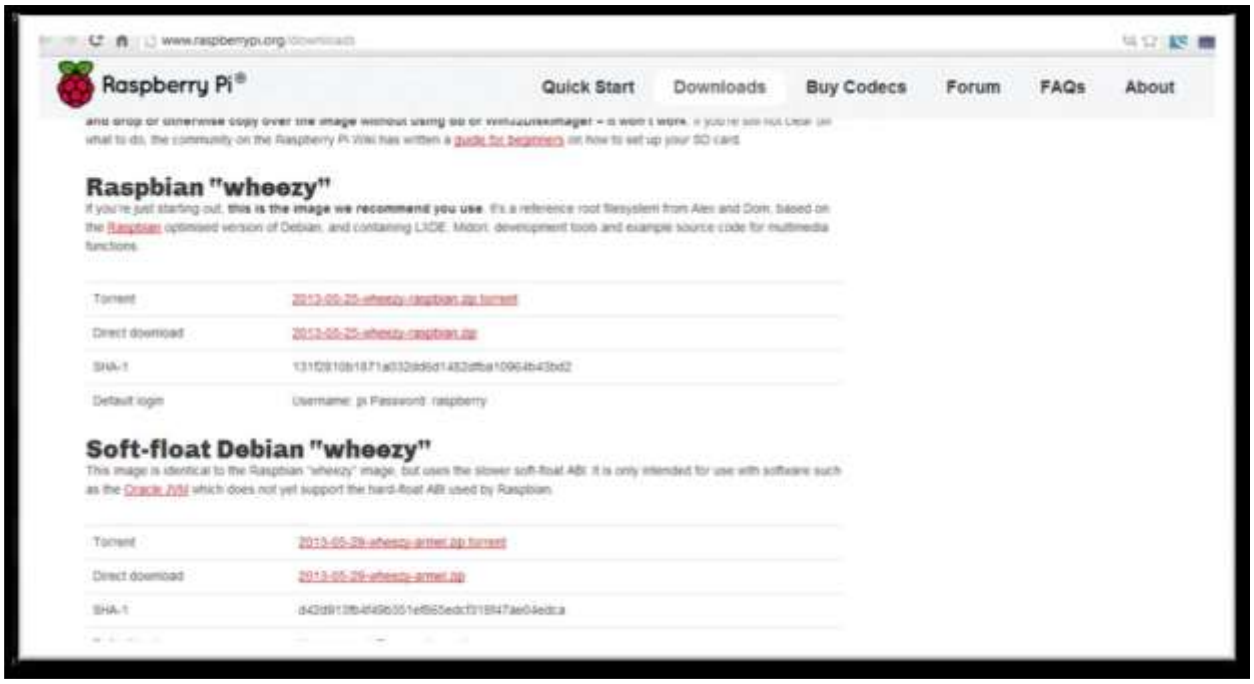
A continuació vam haver de posar el driver, la bateria lipo i la Raspberry Pi al cotxe, per fer això també vam haver de construir un tipus de plataforma per tal que quedés tot més ordenat. Finalment, el cotxe ha quedat així:

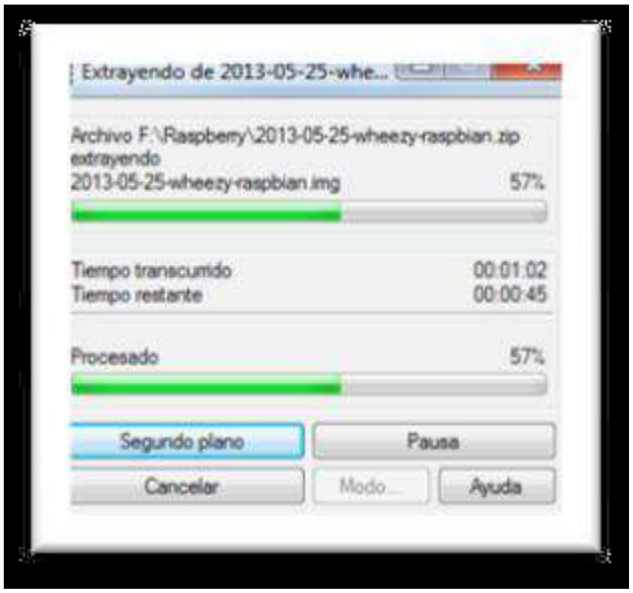




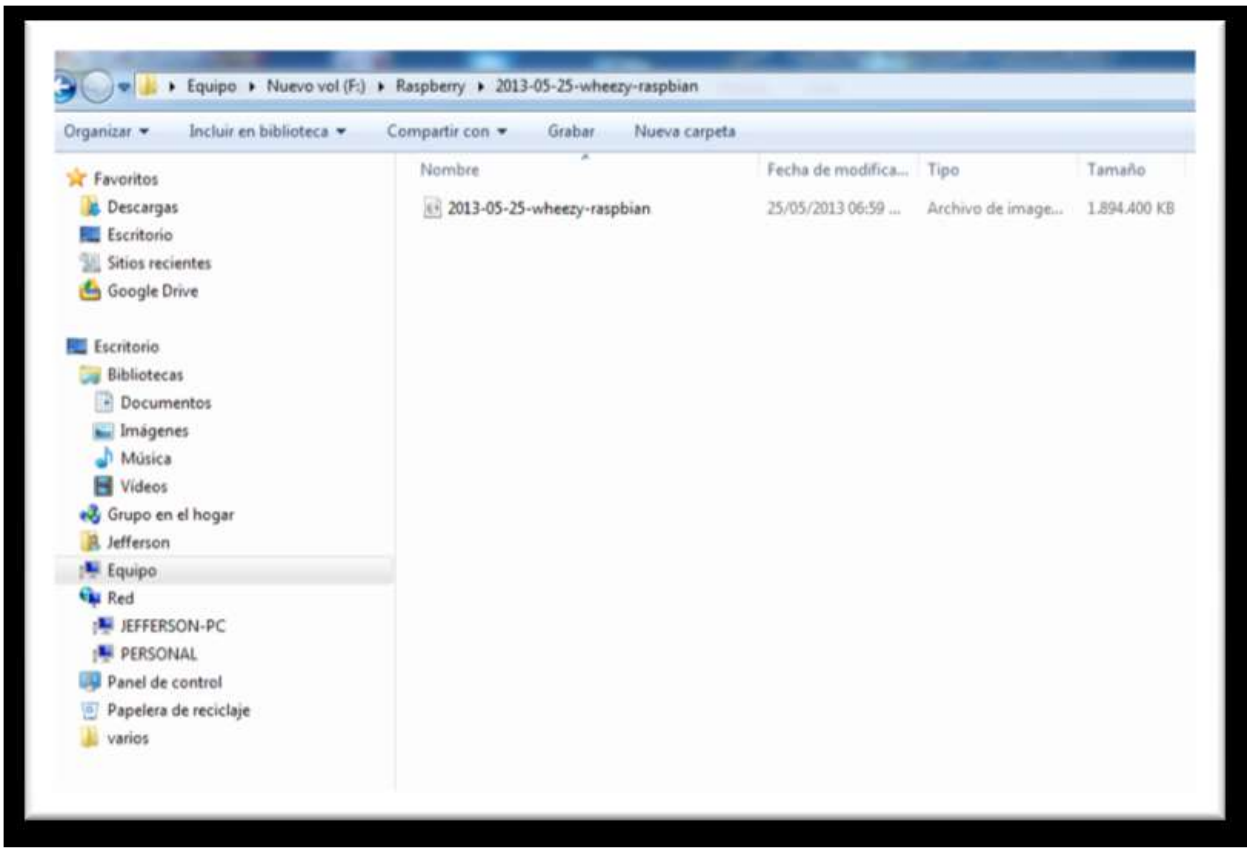
2.2.- Instal·lant el Sistema Operatiu Raspbian

La primera cosa que hem de fer amb la Raspberry Pi, és instal·lar el Sistema Operatiu a la tarja SD. He escollit Raspbian, i per descarregar-ho s'ha d'anar a la pàgina de descàrregues de Raspberry (http://www.raspberrypi.org/downloads).





Un cop descarregat, el descomprimim per tal d'obtenir la imatge del Sistema Operatiu.

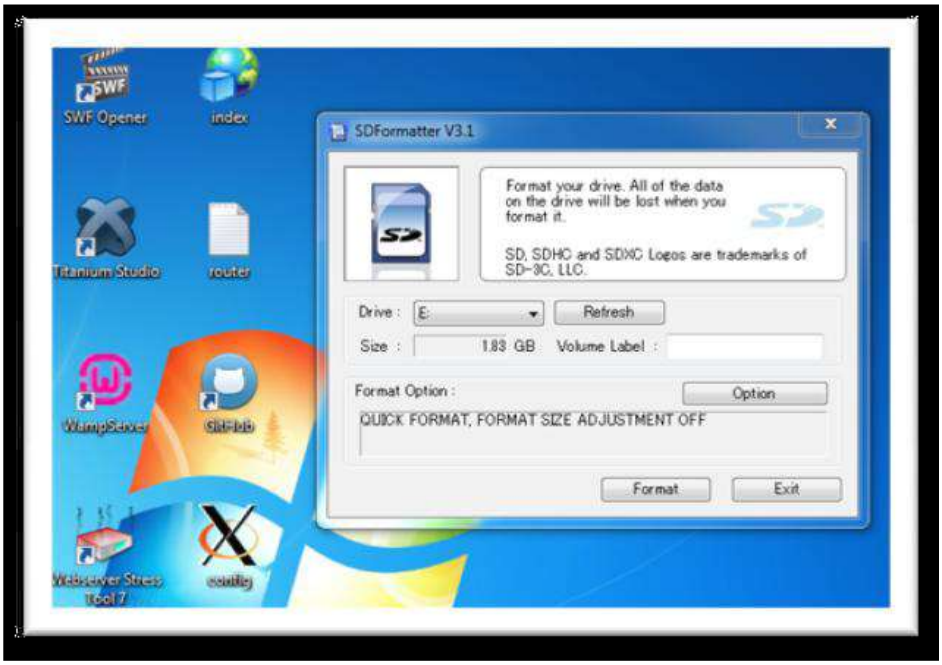


2.2.1.- Preparant la tarja SD

Per que la tarja SD es converteixi en un disc dur per que arranqui el Sistema Operatiu hem de seguir aquests passos:

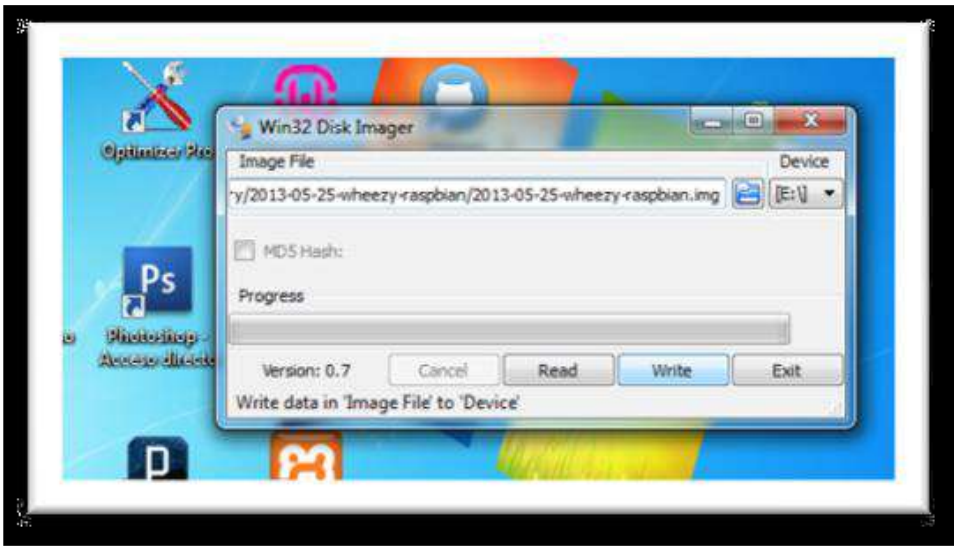
1- Descarregar SD Fromatter
(https://www.sdcard.org/downloads/formatter_3/)

2- Formatar la memòria amb SD Formatter:



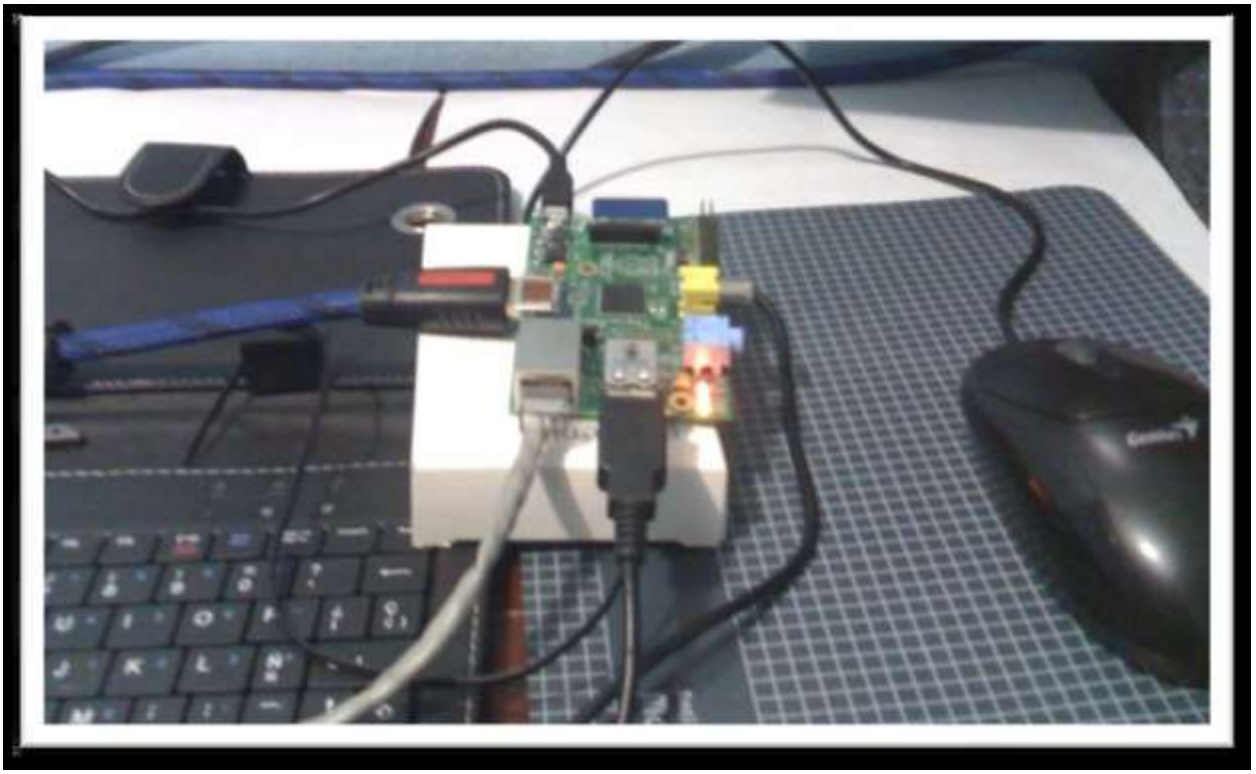
3- Descarregar i instal·lar Win32DiskImager
(http://sourceforge.net/projects/win32diskimager/)

4- Carregar la imatge de Raspbian i li donem a Write



Trigarà uns 10 minuts en gravar la imatge.

Quan hagi acabat, retirem la tarja i la fiquem a la Raspberry Pi. Connectem un cable ethernet, un carregador de 2 Ampers, teclat i ratolí i la connectem a un monitor amb HDMI. Ens ha de quedar així:



Quan s'encengui la Raspberry, primer carregarà els divers i els controladors (la pantalla negra amb lletres blanques) i després ens sortirà la configuració inicial:

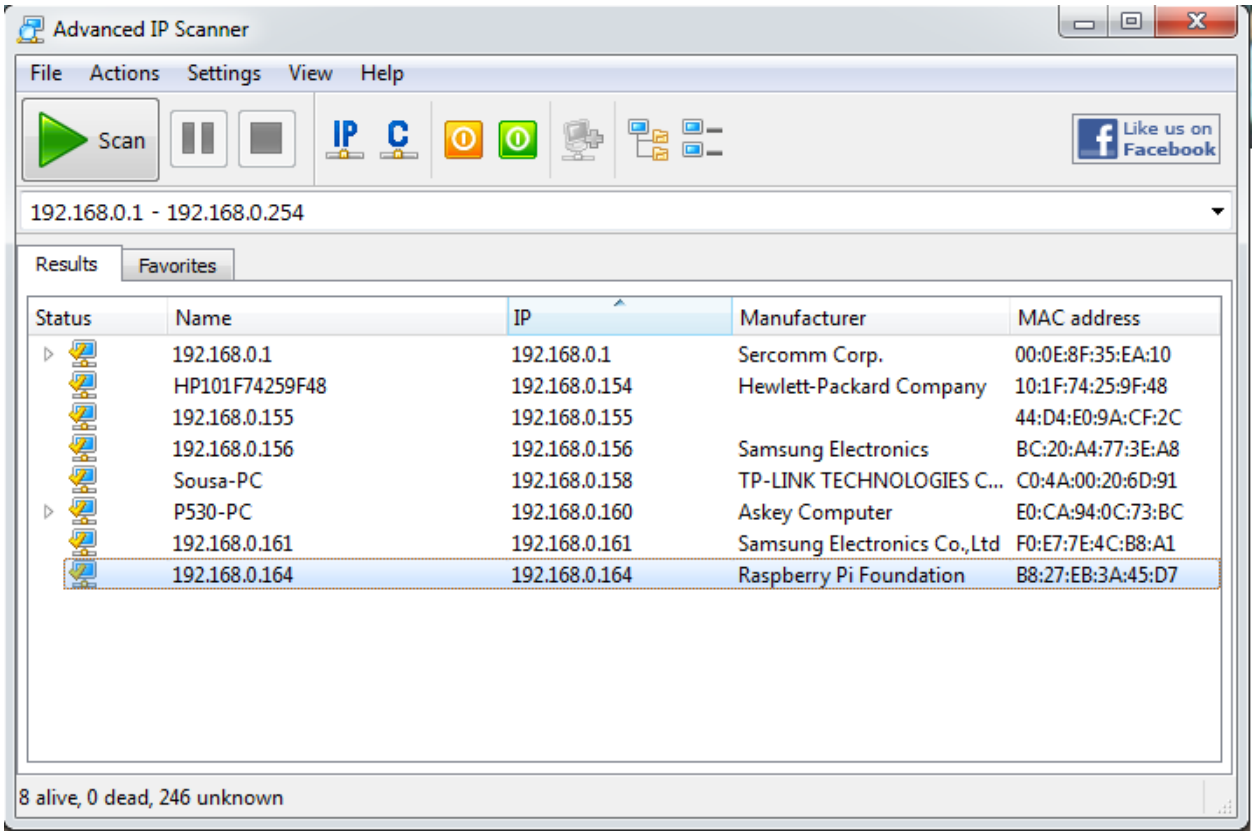
Donarem a la primera opció (*Expand Filesystem*), la qual redimensionarà la memòria SD per tal de donar-nos més espai per arxius i aplicacions.

Ara anem a la tercera opció (*Enable Boot to Desktop*), que farà que s'encengui la interfase gràfica de l'escriptori cada cop que encenem el Raspbian.

2.3.- Connectant la Raspberry Pi a través de SSH

Poder controlar la Raspberry Pi a través de SSH ens facilita la feina al moment d'instal·lar i configurar aplicacions, sense necessitat de tenir connectat una TV o monitor amb l'entrada HDMI.

Per fer això primer hem d'identificar la IP local que té assignada la Raspberry Pi, i ho podem fer utilitzant un petit programa anomenat Advance IP Scan (http://www.advanced-ip-scanner.com/es).



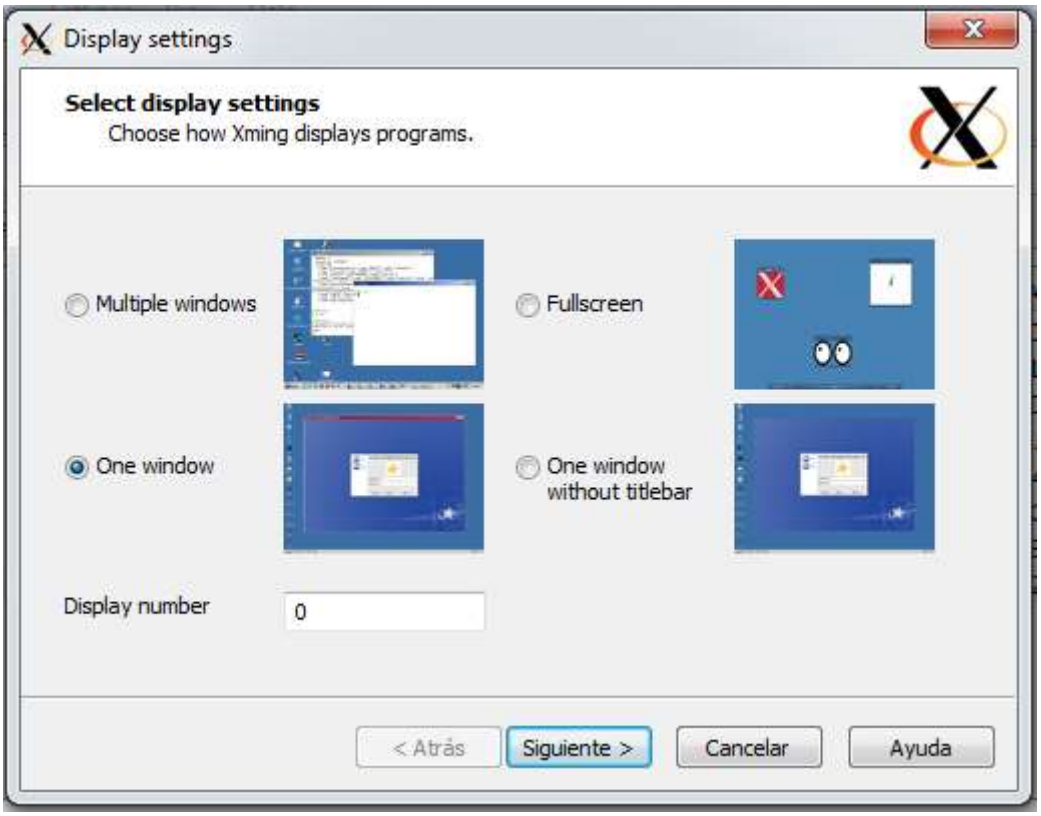
Com es pot veure a la imatge, la IP assignada a la Raspberry és la 192.168.0.164

Ara descarregarem 2 programes, PuTTY, que ens servirà per establir una connexió amb la Raspberry a través de SSH per consola (no l'utilitzarem directament, però l'Xming el necessita per fer la seva tasca), i el programa Xming, que ens servirà per fer una connexió SSH amb la Raspberry amb l'entorn gràfic.

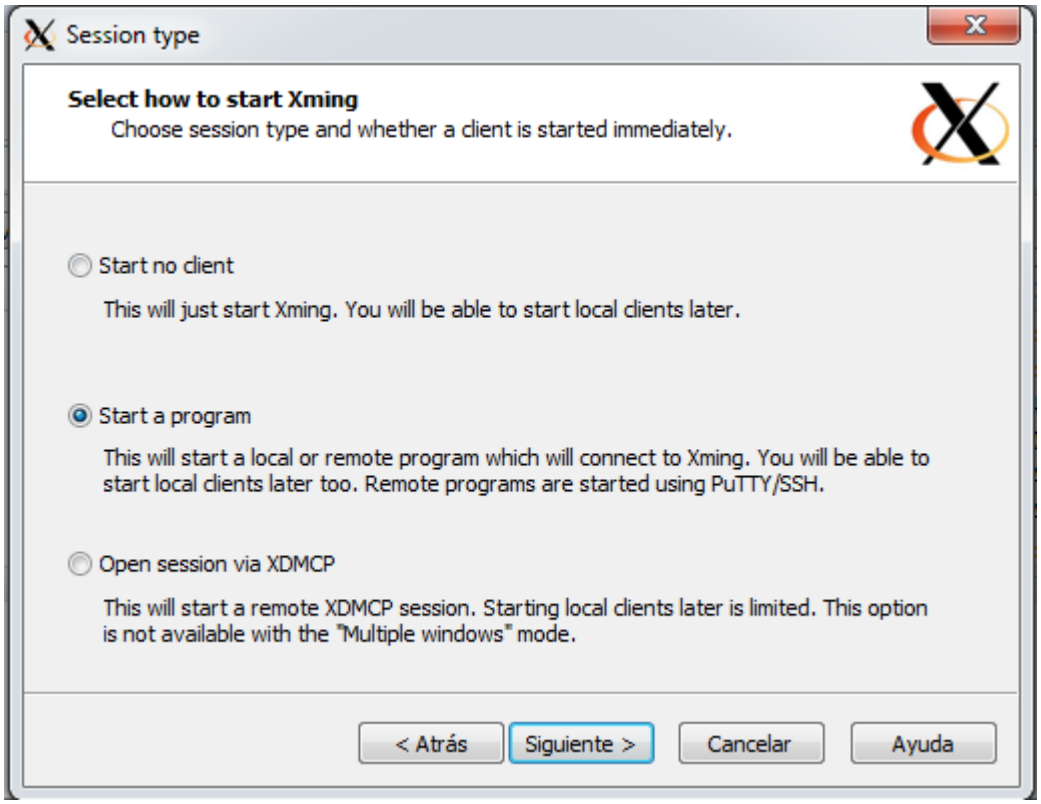
PuTTY: http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html

Xming: http://sourceforge.net/projects/xming/files/latest/download

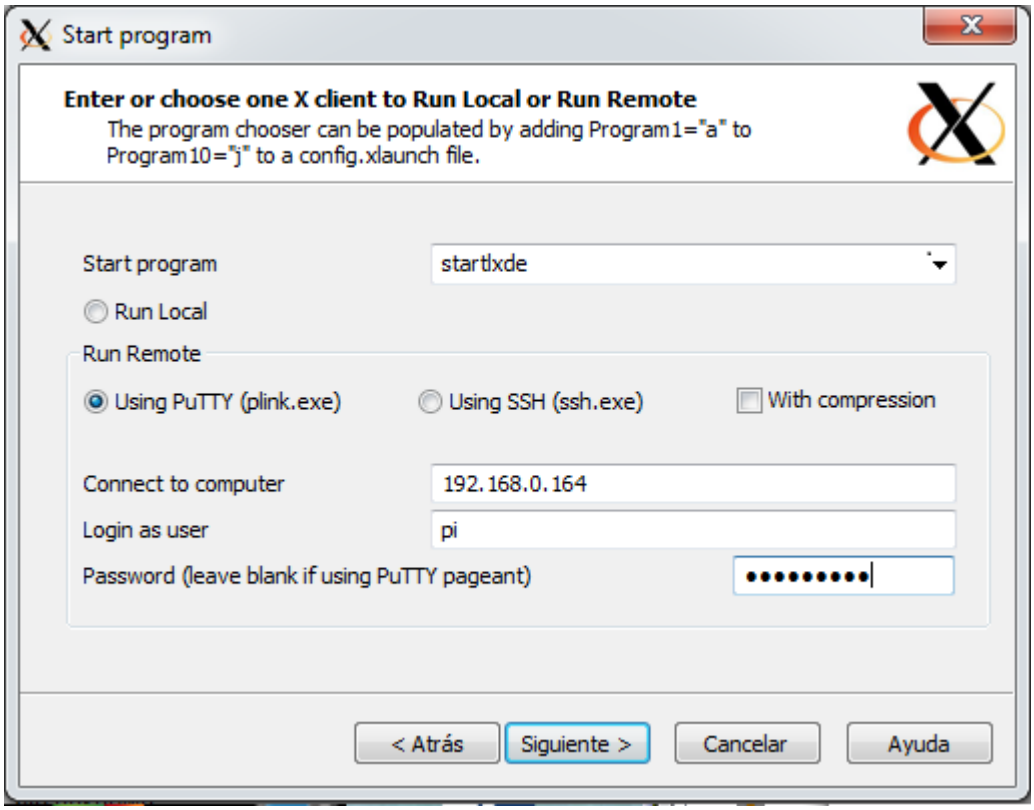
Un cop hem instal·lat els dos programes, obrim l'Xlauncher. Per tal de connectar-nos a la Raspberry Pi, haurem de configurar el programa seguint aquests passos:



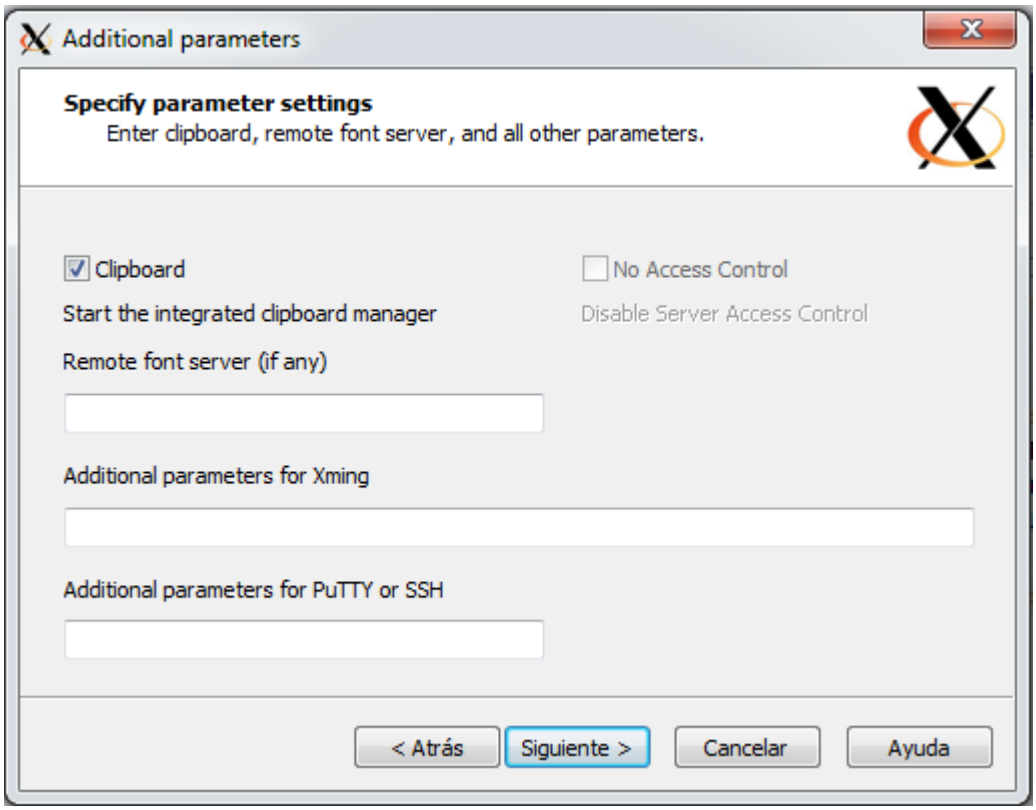
Seleccionem l'opció *"One window"*.



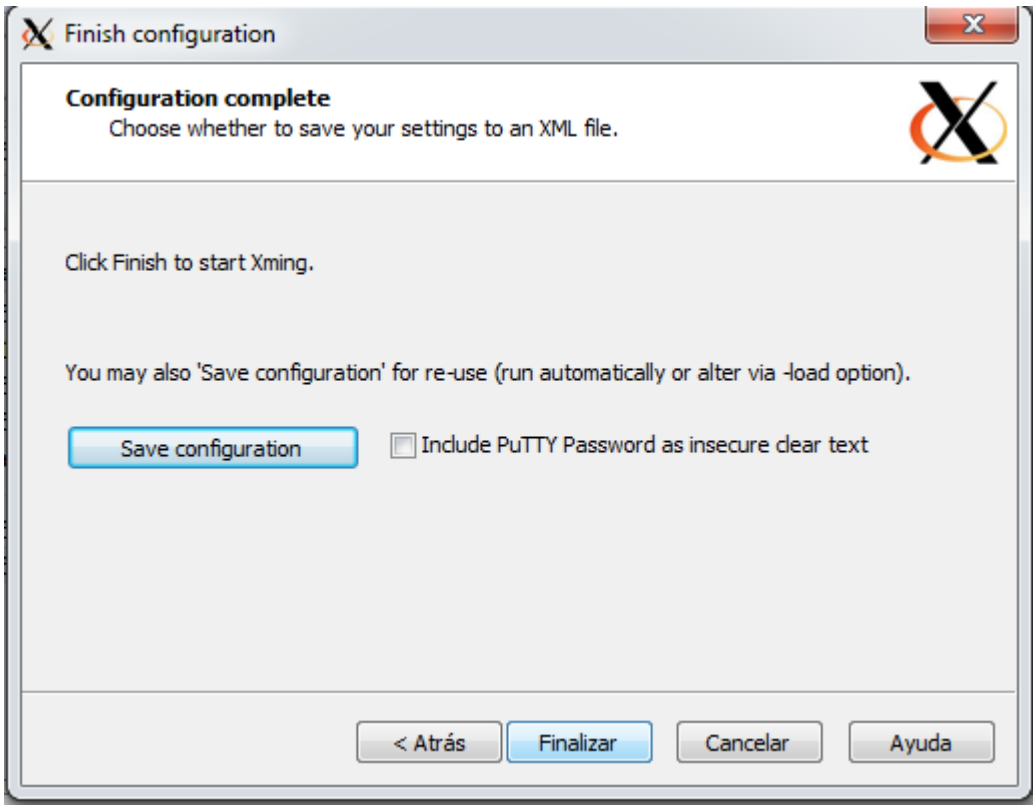
Seleccionem l'opció *"Start a Program"*.



En aquesta part escriurem *"startlxde"* al costat de *"Start program"*, seleccionarem *"Using PuTTY"* i emplenarem les dades de sota, a *"Connect to computer"* escriurem la IP de la Raspberry, a *"Login as user"* escriurem "pi" i a *"Password"* escriurem "raspberry".



Això ho deixarem com està.



Aquí si volem podem desar la configuració, encara que no és necessari, i donem a "Finalizar", per connectar-nos a la Raspberry Pi.

2.4.- Instal·lant i configurant un servidor web a la Raspberry Pi

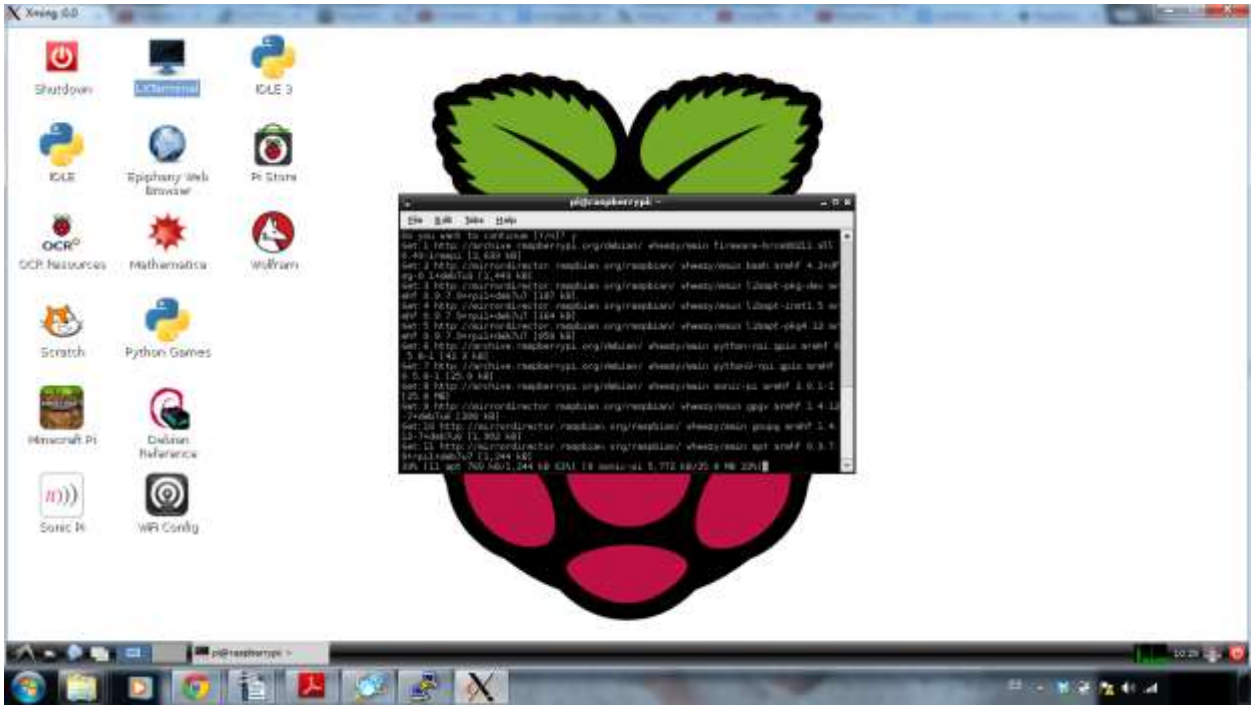
Ara instal·larem un servidor APACHE, amb PHP i MySQL, utilitzant la connexió remota SSH. Haurem d'executar la terminal (si utilitzem PuTTY no farà falta).

Primer escriurem:

```
sudo apt-get update
```

```
sudo apt-get upgrade
```

Això per actualitzar les aplicacions de la Raspberry. *"Sudo"* serveix per indicar que dones permís d'administrador, i *"apt-get"* és una ordre que serveix per instal·lar aplicacions, actualitzar el sistema, etc. Depèn del que escriguis desprès.

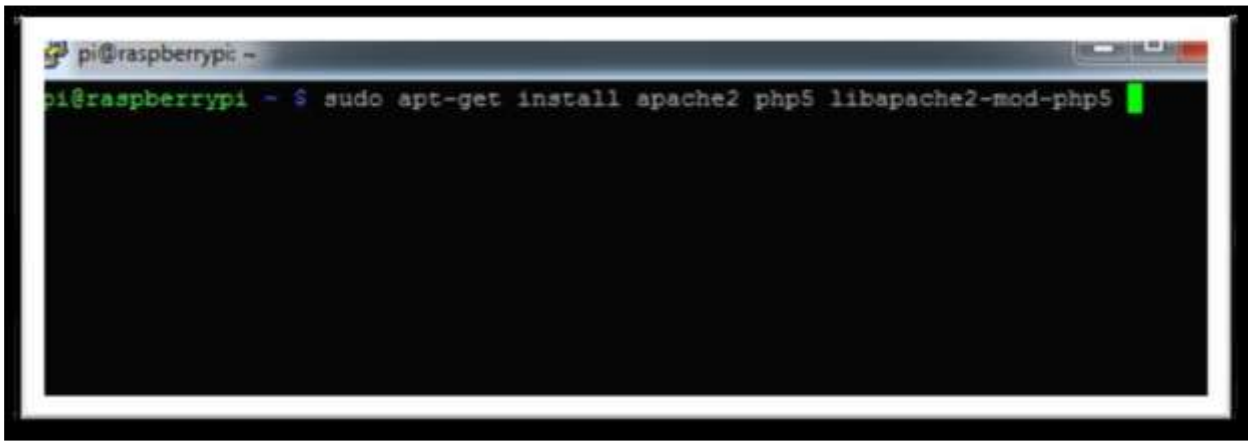


Veurem això, com es pot observar podrem veure l'escriptori de la Raspberry Pi de manera remota amb la connexió SSH.

Després d'això:

Instal·lar Apache i PHP

```
sudo apt-get install apache2 php5 libapache2-mod-php5
```



Reiniciem l'Apache

```
sudo /etc/init.d/apache2 restart
```

Atorgament de permisos

Els directoris típicament utilitzats per un servidor web en Linux es situen en ***/var/www***, y l'usuari típic per aquest entorn és ***www-data***. Ara crearem el grup i l'usuari estàndard pel nostre servidor, al mateix temps que atorgarem els permisos pertinents i afegirem al nostre usuari per defecte (pi) a aquest grup. D'aquesta forma no serà necessari que l'usuari root sigui sempre qui pugui operar en ***/var/www***.

Canviem l'usuari i el grup al directori

```
Sudo chown www-data:www-data /var/www/
```

Donem els permisos a la carpeta www/

```
Sudo chmod 775 /var/www/
```

Afegim l'usuari pi al grup www-data

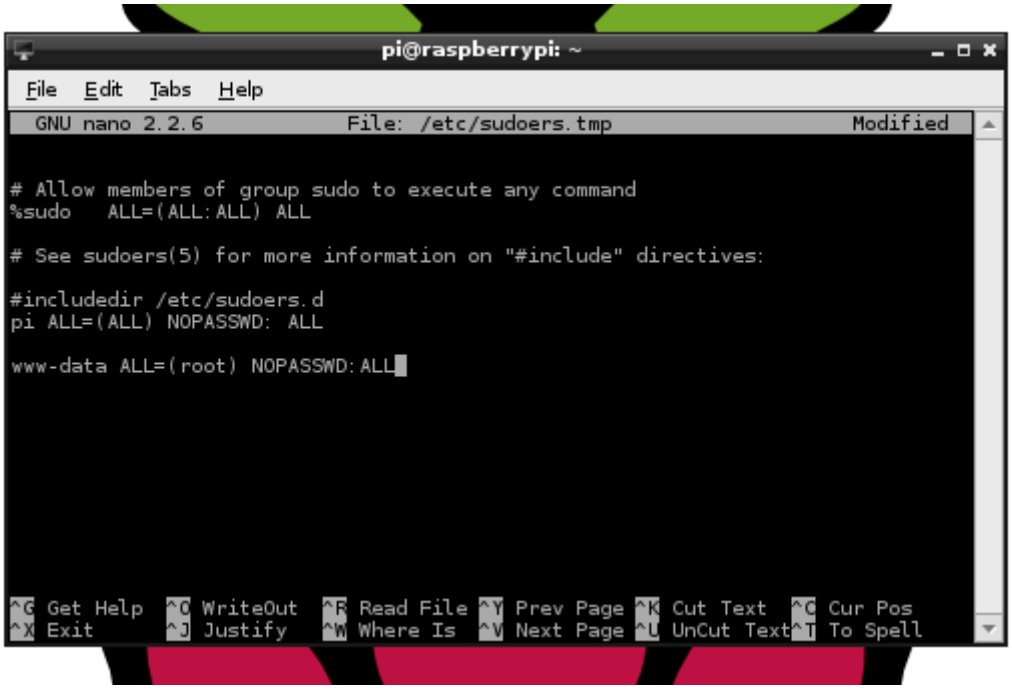
```
Sudo usermod -a -G www-data pi
```

```
Sudo visudo
```

Afegim el següent codi al final:

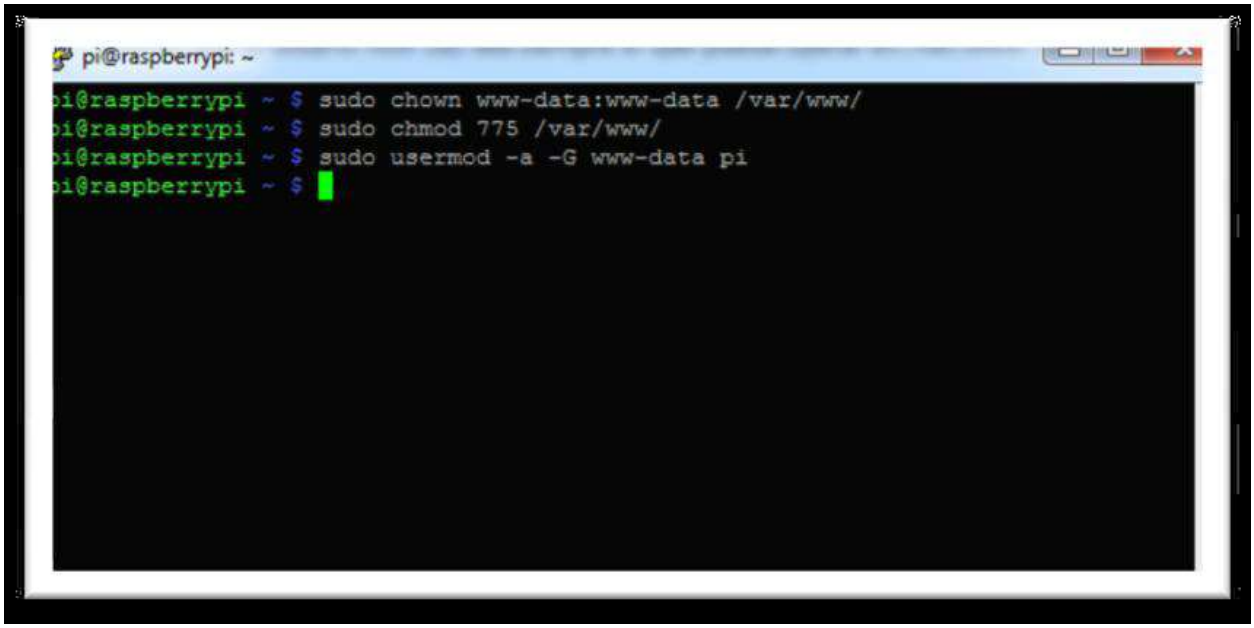
```
www-data ALL=(root) NOPASSWD:ALL
```

Donem a Ctrl+X, premem "Y" i donem a Enter. (A la foto de sota podeu veure un exemple de com ha de quedar).

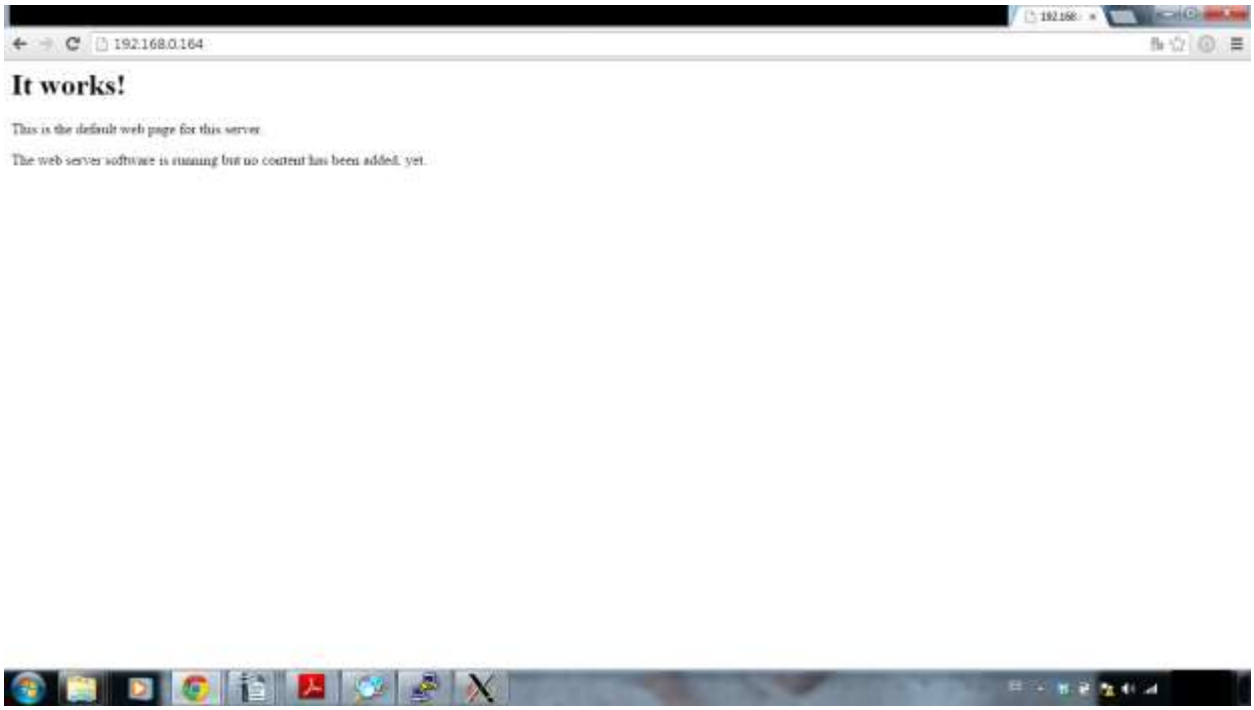


Reiniciem el servidor

```
Sudo /etc/init.d/apache2 restart
```



Col·loquem a un navegador web la IP de la nostra Raspberry Pi, en aquest exemple seria: http://192.168.0.164/



Amb aquest missatge comprovem que tot funciona, encara que crearem un fitxer PHP per confirmar-ho i per veure com funciona.

Creant un arxiu de prova, "Hola món" en PHP

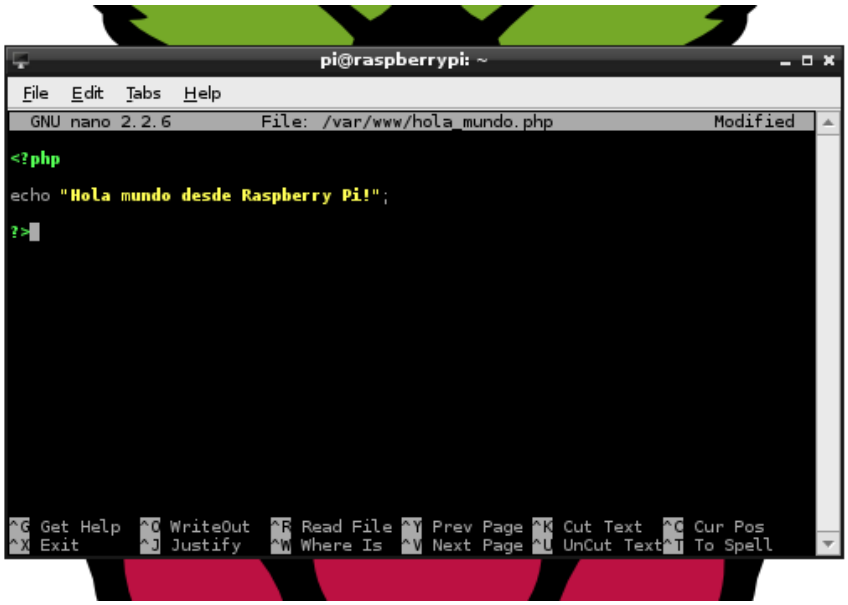
```
Sudo nano /var/www/hola_mundo.php
```

L'ordre "nano" serveix per crear o editar fitxers.

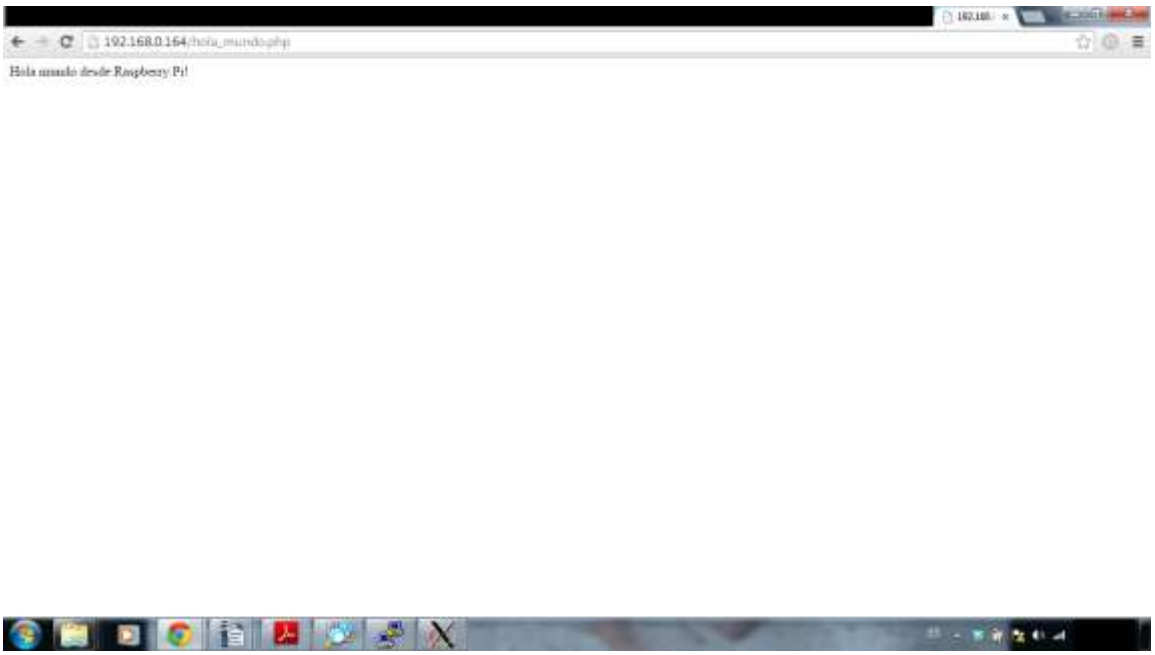
Escriurem el codi del fitxer:

```
<?php
echo "Hola mundo desde Raspberry Pi!";
?>
```

Utilitzem "<?php" i "?>" per indicar que entre aquestes dues etiquetes hi ha un codi PHP. La funció "echo" serveix per escriure a la web un missatge, en aquest cas serà "Hola mundo desde Raspberry Pi!".



Entrem a la web de la Raspberry Pi, ara afegint /hola_mundo.php al final de l'adreça, quedaria així: http://192.168.0.164/hola_mundo.php



Veiem que tot funciona correctament.

2.5.- Instal·lant i configurant NO-IP a la Raspberry Pi

Aquesta aplicació és bastant important, ja que ens permetrà controlar la Raspberry i accedir a la pàgina web amb la que donarem les ordres al cotxe des de qualsevol lloc sense haver d'estar connectat a la mateixa xarxa. És a dir, converteix la nostra IP Privada en una adreça pública.

Hem de crear un compte a http://noip.com/ i ens hem de recordar de les dades, ja que després les haurem d'utilitzar.

Instal·lant el client NO-IP a la Raspberry

Creant el directori pel programa client

```
Sudo mkdir /home/pi/noip
```

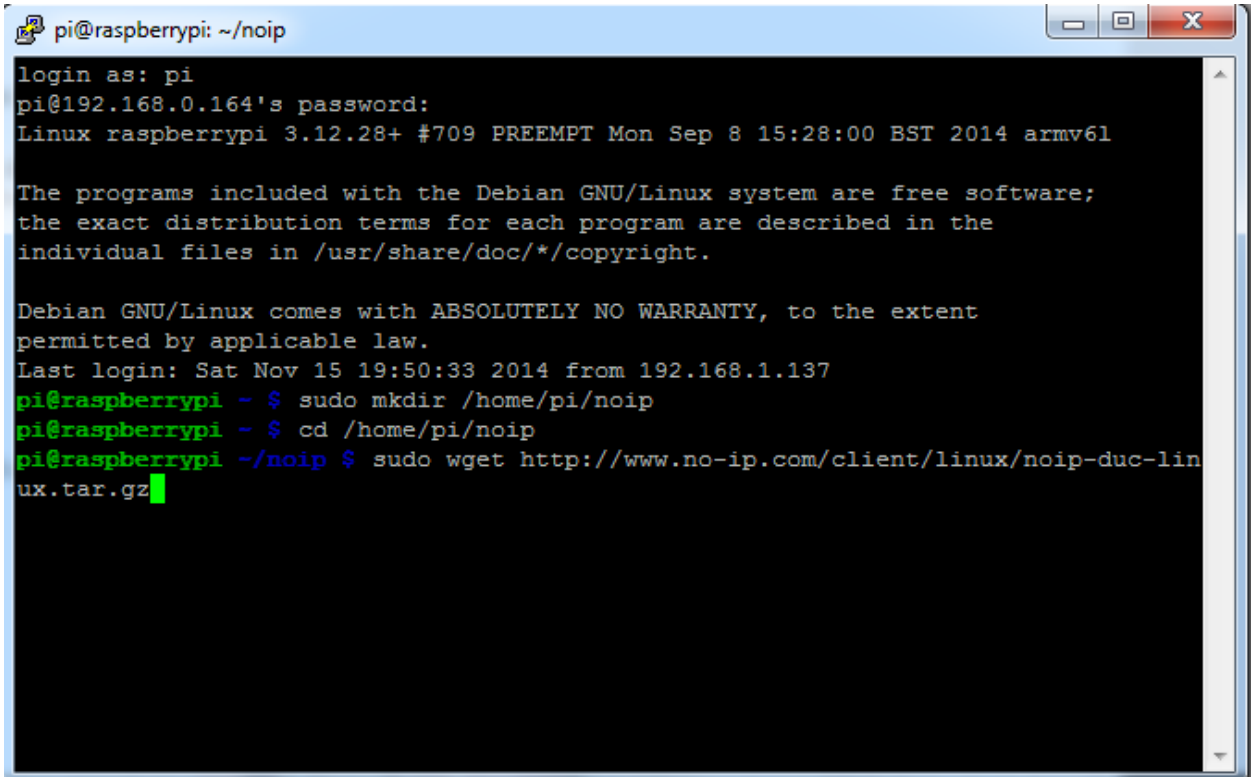
L'ordre "mkdir" s'utilitza per crear directoris.

Entrem dins el directori creat

```
cd /home/pi/noip
```

Descarreguem el programa client

```
Sudo wget http://www.no-ip.com/client/linux/noip-duc-linux.tar.gz
```



Extraurem el fitxer

```
Sudo tar vzxf noip-duc-linux.tar.gz
```

Es descomprimirà una carpeta, entrem dins aquesta

```
Cd noip-2.1.9-1
```

Instal·lem el programa client

```
Sudo make
Sudo make install
```

Haurem d'anar afegint les nostres dades per la instal·lació, com es mostra a la imatge.

```
pi@raspberrypi: ~/noip/noip-2.1.9-1
le]
noip2.c: In function 'hosts':
noip2.c:1838:20: warning: variable 'y' set but not used [-Wunused-but-set-variab
le]
pi@raspberrypi ~/noip/noip-2.1.9-1 $ sudo make install
if [ ! -d /usr/local/bin ]; then mkdir -p /usr/local/bin;fi
if [ ! -d /usr/local/etc ]; then mkdir -p /usr/local/etc;fi
cp noip2 /usr/local/bin/noip2
/usr/local/bin/noip2 -C -c /tmp/no-ip2.conf

Auto configuration for Linux client of no-ip.com.

Please enter the login/email string for no-ip.com  alexalarcondrs@gmail.com
Please enter the password for user 'alexalarcondrs@gmail.com'  *********

Only one host [alexsoussa.ddns.net] is registered to this account.
It will be used.
Please enter an update interval:[30]  30
Do you wish to run something at successful update?[N] (y/N)  n

New configuration file '/tmp/no-ip2.conf' created.

mv /tmp/no-ip2.conf /usr/local/etc/no-ip2.conf
pi@raspberrypi ~/noip/noip-2.1.9-1 $
```

2.6.- Controlant el GPIO de la Raspberry Pi amb Python

Raspbian ve amb una llibreria Python incorporada la qual ens permet treballar amb els pins GPIO de la nostra Raspberry. Un exemple de com controlar un motor amb aquesta llibreria i dos pins GPIO:

```
1  import RPi.GPIO as GPIO #importem la llibreria que ens permet controlar els GPIO
2
3  GPIO.setup(17,GPIO.OUT) #indiquem que utilitzarem el pin 17 com a sortida
4  GPIO.setup(18,GPIO.OUT) #indiquem que utilitzarem el pin 18 com a sortida
5
6  GPIO.output(17,True) #donem un valor d'1 al pin 17
7  GPIO.output(18,False)  #donem un valor de 0 al pin 18
```

A la imatge s'explica que fa cada ordre.

2.7.- Accedint al GPIO de la Raspberry Pi des de PHP

A PHP tenim una funció anomenada ***exec***, la qual executa un programa extern. Un exemple d'ús:

```
<?php
echo exec('programa');
?>
```

Ara nosaltres programarem un codi per que executi el fitxer de l'exemple de l'apartat anterior, que l'hem anomenat "exemple.py":

```
1  <?php
2  echo "exec sudo python exemple.py";
3  ?>
```

Quan executem el fitxer PHP, aquest executarà el fitxer Python que li hem dit.

2.8.- Aplicació web per controlar el GPIO

L'aplicació web que he programat té aquest fitxers:

Carpeta css → Conté arxius d'estils

Carpeta images → Conté imatges per la web

Carpeta js → Conté el script de jQuery 1.9

Carpeta py → Conté els arxius que faran moure el cotxe en Python

Index.html → Aparença bàsica de la pàgina, i organització dels botons

Procesa.php → Arxiu que és cridat per l'index.html i que crida als arxius Python

De totes aquestes carpetes i arxius, explicarem els arxius de dins la carpeta ***py*** i l'arxiu ***pocesa.php***, ja que els altres no ens aportaran cap informació que ens sigui d'utilitat al projecte.

Procesa.php →

```
1   <?php
2   $valor_estado=$_POST['valor_estado'];
3   switch ($valor_estado) {
4       case 1:
5           exec('sudo python /var/www/control/py/avanza.py');
6           break;
7       case 2:
8           exec('sudo python /var/www/control/py/izquierda.py');
9           break;
10      case 3:
11          exec('sudo python /var/www/control/py/derecha.py');
12          break;
13      case 4:
14          exec('sudo python /var/www/control/py/retrocede.py');
15          break;
16      case 5:
17          exec('sudo python /var/www/control/py/para.py');
18          break;
19      default:
20          exec('sudo python /var/www/control/py/para.py');
21          break;
22  }
23  ?>
24
```

A la línia 2 rebem la variable ***valor_estado***, la qual ens ve donada per l'index.html i ens diu quin botó hem premut per tal de que doni l'ordre correcte, per exemple, si hem premut el botó per anar endavant, la variable ***valor_estado*** serà 1, i per tant cridarem l'arxiu avanza.py, el qual tindrà el seu codi per que avanci el cotxe.

Les connexions dels motors amb el GPIO són així:

	Gris (Pin 18)	Lila (Pin 17)	Blanc (Pin 23)	Marró (Pin 22)
endavant	0	1	0	1
endarrere	1	0	1	0
dreta	0	1	0	0
esquerra	0	0	1	0
	motor esquerre		motor dret	

Per tant, amb les programacions dels fitxers Python hem de fer que cadascun doni el valor corresponent a cada pin GPIO, per exemple, el fitxer ***avanza.py***, haurà de donar un valor de 0 als pins 18 i 23, i un valor d'1 als pins 17 i 22.

Avanza.py →

```
1   import RPi.GPIO as GPIO #importem la llibreria que ens permet controlar els GPIO
2
3   GPIO.setmode(GPIO.BCM)
4   GPIO.setwarnings(False) #desactivem les possibles alarmes que poden sortir
5   GPIO.setup(17,GPIO.OUT)#Motor1
6   GPIO.setup(18,GPIO.OUT)#Motor1
7
8   GPIO.setup(22,GPIO.OUT)#motor2
9   GPIO.setup(23,GPIO.OUT)#Motor2
10
11  GPIO.output(17,True) #donem un valor d'1 al pin 17
12  GPIO.output(18,False) #donem un valor de 0 al pin 18
13
14  GPIO.output(22,False) #donem un valor de 0 al pin 22
15  GPIO.output(23,True) #donem un valor d'1 al pin 23
```

Explicació de funcions que no estan explicades a la imatge o que no queden clares:

La funció ***GPIO.setmode(GPIO.BCM)***, serveix per indicar que els noms dels pins GPIO que utilitzarem són els basats en el BCM, que a la imatge que he posat per explicar els pins GPIO a la part de teoria, són els noms que estan als extrems.

Amb aquestes instruccions el cotxe anirà cap endavant.

Derecha.py →

```
1   import RPi.GPIO as GPIO
2
3   GPIO.setmode(GPIO.BCM)
4   GPIO.setwarnings(False)
5   GPIO.setup(17,GPIO.OUT)#Motor1
6   GPIO.setup(18,GPIO.OUT)
7
8   GPIO.setup(22,GPIO.OUT)#motor2
9   GPIO.setup(23,GPIO.OUT)
10
11  GPIO.output(17,True)
12  GPIO.output(18,False)
13
14  GPIO.output(22,False)
15  GPIO.output(23,False)
```

Com es pot veure, només canvia els ***GPIO.output*** de cada pin, que donen un valor o un altre, els quals els tenim a la taula d'abans. Tot lo altre és idèntic en tots els fitxers.

Izquierda.py →

```
1   import RPi.GPIO as GPIO
2
3   GPIO.setmode(GPIO.BCM)
4   GPIO.setwarnings(False)
5   GPIO.setup(17,GPIO.OUT)#Motor1
6   GPIO.setup(18,GPIO.OUT)
7
8   GPIO.setup(22,GPIO.OUT)#motor2
9   GPIO.setup(23,GPIO.OUT)
10
11  GPIO.output(17,False)
12  GPIO.output(18,False)
13
14  GPIO.output(22,False)
15  GPIO.output(23,True)
```

Retrocede.py →

```
1   import RPi.GPIO as GPIO
2
3   GPIO.setmode(GPIO.BCM)
4   GPIO.setwarnings(False)
5   GPIO.setup(17,GPIO.OUT)#Motor1
6   GPIO.setup(18,GPIO.OUT)
7
8   GPIO.setup(22,GPIO.OUT)#motor2
9   GPIO.setup(23,GPIO.OUT)
10
11  GPIO.output(17,False)
12  GPIO.output(18,True)
13
14  GPIO.output(22,True)
15  GPIO.output(23,False)
```

Para.py →

```
1   import RPi.GPIO as GPIO
2
3   GPIO.setmode(GPIO.BCM)
4   GPIO.setwarnings(False)
5   GPIO.setup(17,GPIO.OUT)#Motor1
6   GPIO.setup(18,GPIO.OUT)
7
8   GPIO.setup(22,GPIO.OUT)#motor2
9   GPIO.setup(23,GPIO.OUT)
10
11  GPIO.output(17,False)
12  GPIO.output(18,False)
13
14  GPIO.output(22,False)
15  GPIO.output(23,False)
```

Amb aquest codi, el cotxe pararà.

2.7.- Instal·lant i configurant la càmera

Per poder tenir un *streaming* amb una càmera connectada per USB, necessitarem instal·lar la llibreria MJPG-streamer.

Per fer-ho, tornem a connectar-nos per SSH a la Raspberry, i executem aquesta ordre:

Sudo wget http://sourceforge.net/projects/mjpg-streamer/files/latest/download

Després, descomprimim el fitxer descarregat

Sudo tar xzvf mjpg-streamer.tar.gz

Instal·lem més llibreries

Sudo apt-get install lobjpeg8-dev

Sudo apt-get install imagemagick

Entrem al directori de MJPG-streamer que hem descomprimit abans

Cd mjpg-streamer/mjpg-streamer

Instal·lem el contingut

Sudo make

Creem un directori per el nostre *streaming* i el configurem

Sudo mkdir /tmp/stream

Raspistill -w 640 -h 480 -q 5 -o /tmp/stream/pic.jpg -tl 1 -t 999999 -th 0:0:0

Amb aquesta última ordre estem iniciant un streaming amb uns paràmetres per defecte, per exemple, hem assignat la resolució del vídeo a 640x480px, a una qualitat 5. Aquets paràmetres s'han de posar per defecte, no es poden variar.

Ara si entrem a la web de la nostra Raspberry Pi afegint ":8080", podrem veure el nostre *streaming*.

Per poder veure aquest *streaming* juntament amb la nostra web de control del cotxe, haurem d'afegir aquesta etiqueta al nostre arxiu ***index.html***.

```
7        <img src="/?action=stream" />
```

3.- Conclusions

Aquest treball ens ha servit per introduir-nos una mica a la robòtica, i per aprendre una mica més sobre programació. No ha estat un treball fàcil, però amb esforç hem aconseguit el que ens vam proposar des d'un principi, la construcció d'un cotxe telecomandat per WiFi.

A partir d'aquest treball he après:

- Coneixements més profunds sobre la robòtica i l'electrònica.
- Coneixements més profunds sobre la programació en PHP i Python i sobre els Sistemes Operatius Linux.
- Com funciona i de que es tracta la Raspberry Pi, i quins usos li podem donar.

4.- Agraïments

Primer de tot, aquest treball no hagués estat possible sense el nostre tutor. Ens ha guiat i donat un cop de mà sempre que ens ha fet falta, ens ha animat i valorat la feina que anàvem fent.

També agrair a l'Institut, on hem pogut realitzar el muntatge i totes les probes del nostre projecte.

5.- Webgrafia

http://www.raspberrypi.org – Ult. Visita: 5/11/2014

http://commandcat.blogspot.com.es/2013/02/raspberry-pi-como-crear-una-sd-de.html - Ult. Visita: 5/11/2014

http://elinux.org/RPi_Low-level_peripherals - Ult. Visita: 5/11/2014

http://www.cyberhades.com/2013/01/18/raspberry-pi-interactuando-con-gpio-desde-la-linea-de-comandos/ - Ult. Visita: 9/11/2014

http://www.esdebian.org/wiki/lista-comandos-gnulinux-i - Ult. Visita: 9/11/2014

https://www.noip.com - Ult. Visita: 11/11/2014

http://raspberryparatorpes.net - Ult. Visita: 5/11/2014

http://raspberrypihelp.net/tutorials/29-raspberry-pi-no-ip-tutorial - Ult. Visita: 11/11/2014

http://openmicros.org/index.php/articles/94-ciseco-product-documentation/raspberry-pi/217-getting-started-with-raspberry-pi-gpio-and-python - Ult. Visita: 5/11/2014

https://www.youtube.com/watch?v=JJXzlCK4vnY

http://www.ea3um.com/Carta.html - Ult. Visita: 5/11/2014

http://www.akoranga.org/ict4d/2013/04/iniaciacion-al-uso-de-raspberry-pi-en-educacion-i/ - Ult. Visita: 5/11/2014