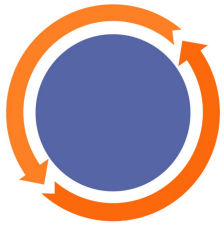
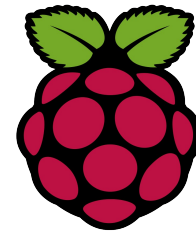
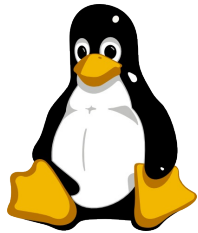
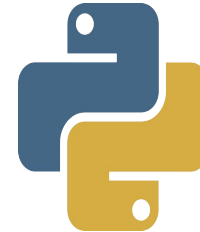


Maquinari lliure controlat amb GNU/Linux



INSTITUT JOAN D'ÀUSTRIA
JEDAI
Jornada d'Empreses, Docents i Alumnat d'Informàtica



Jordi Binefa i Martínez

Responsable d'R+D+i a

Professor de cicles formatius a

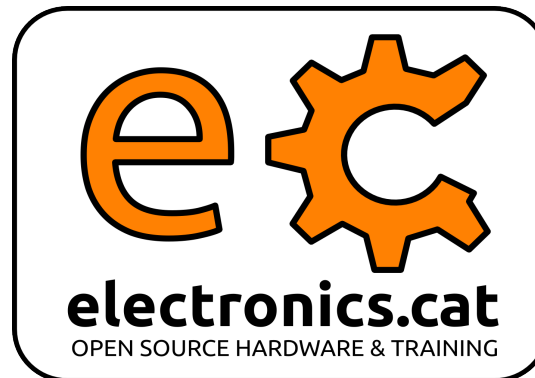


19 / març / 2014

Maquinari lliure controlat amb GNU/Linux

Índex

- Introducció
- Què és la Raspberry Pi?
- Dispositius d'entrada i sortida
- Maquinari
- Python
- Comunicacions I2C i sèrie
- Torn de preguntes



Maquinari lliure controlat amb GNU/Linux

Introducció

Sessió pràctica de maquinari lliure

Introducció

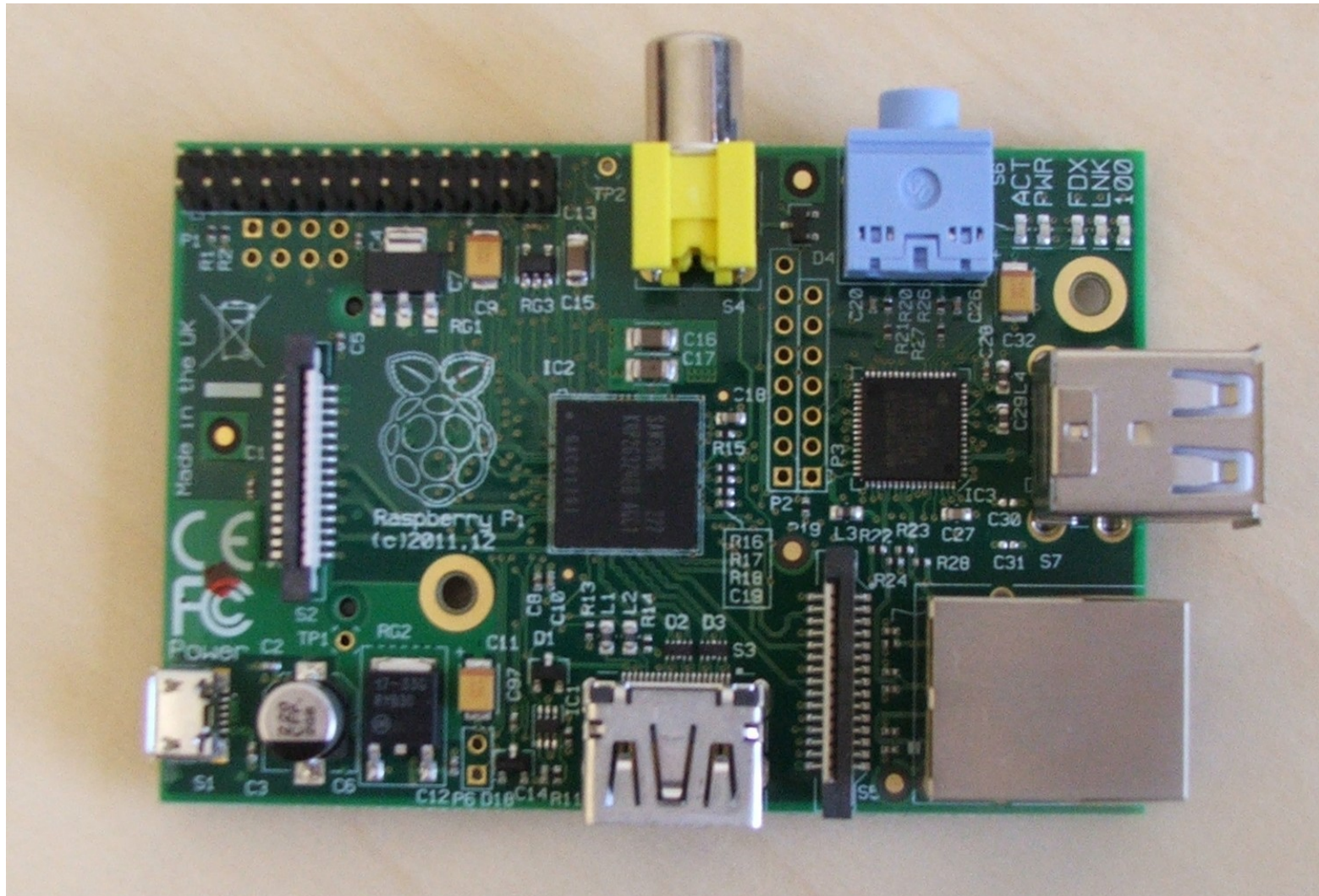
Definició de maquinari lliure :

És un maquinari en que el disseny està a disposició del públic per a que qualsevol ho pugui estudiar, modificar, distribuir, fabricar i vendre.

open hardware

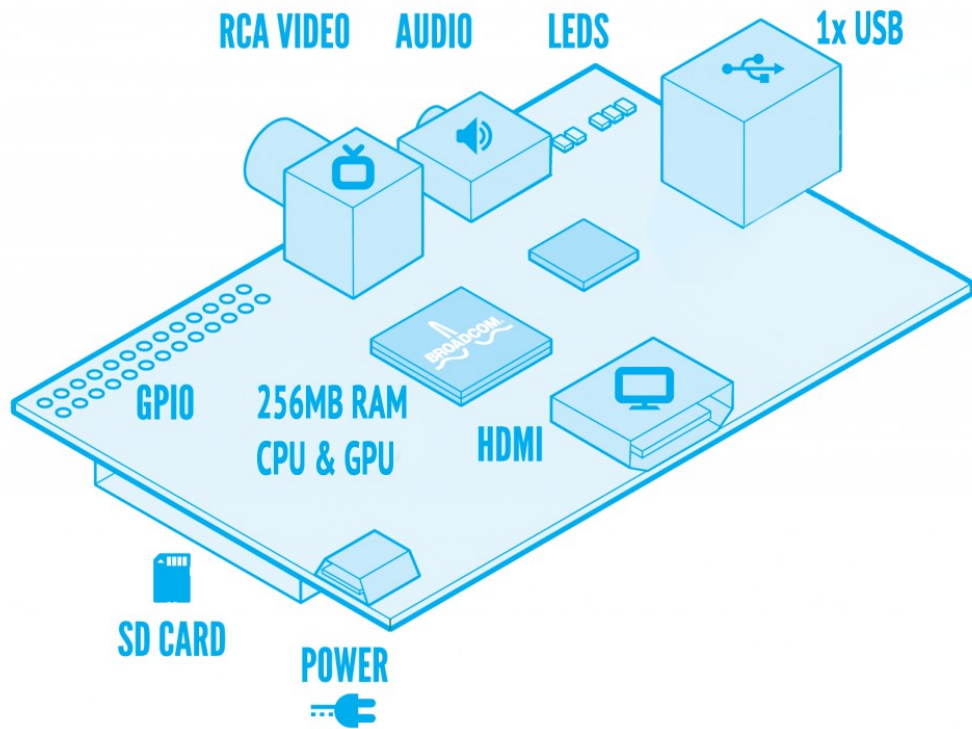
Font : <http://freedomdefined.org/OSHW>

Què és la Raspberry Pi ?

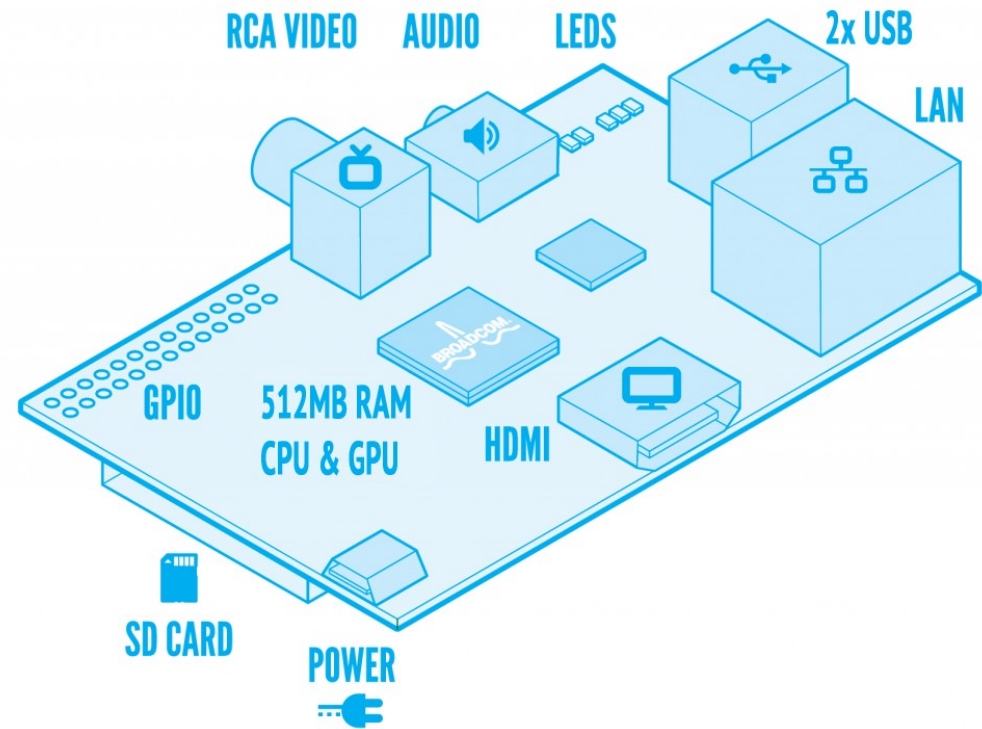


Raspberry Pi

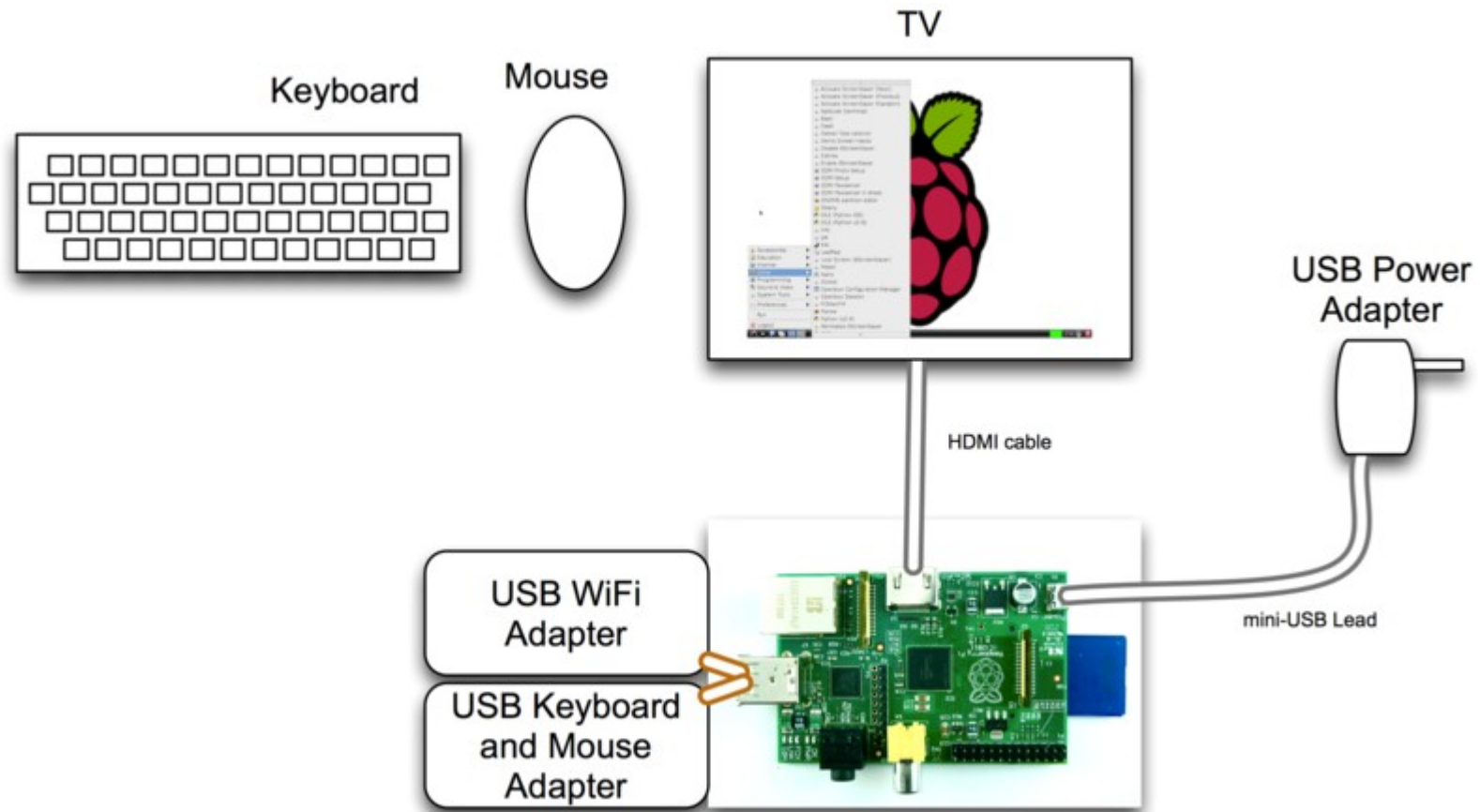
RASPBERRY PI MODEL A - \$25



RASPBERRY PI MODEL B - \$35



Raspberry Pi



Raspberry Pi

- Ordinador amb la mida d'una targeta de crèdit
- CPU: Broadcom BCM2835 SOC
- 700MHz ARM 11 amb coma flotant
- Videocore 4 GPU capaç de reproduir BluRay amb qualitat de video 1080p30 emprant H.264 a 40Mbits/s
- OpenGL ES2.0 i OpenVG
- Targeta SD (es pot emprar USB un cop està funcionant)
- **Model A: 256MB RAM, 1 port USB**
- **Model B: 512MB RAM, 2 ports USB, Ethernet**

Raspberry Pi

- Sortides de vídeo compost i HDMI
- So per HDMI i connector jack d'àudio de 3.5mm; també es pot emprar un micròfon USB com a entrada
- Connexió a un port GPIO (entrades/sortides de propòsit general a **3,3V**)
- Alimentat a 5V per micro USB (2.5W/3.5W. Es poden emprar bateries, per exemple 4 piles AA cells)
- No té rellotge de temps real RTC (ho obté de la xarxa)
- Memòria no expandible

Raspberry Pi

- PVD : US\$25 (Model A) / US\$35 (Model B)
- La placa ve sola: típicament s'afegeix un monitor per HDMI, una targeta SD, teclat i ratolí USB i una font d'alimentació de 5V per microUSB
- Accés lliure als esquemes del maquinari

Dispositius d'entrada – Ratolí i teclat

- Es poden connectar ratolins i teclats USB
- Si calen més connexions USB es pot emprar un concentrador amb alimentació auxiliar



Dispositius de sortida i pantalla tàctil

- Sortida de vídeo HDMI i vídeo compost
- Es poden fer servir connectors DVI o VGA amb un adaptor
- Les pantalles tàctils HDMI funcionen si emulen un ratolí USB
- Chalkboard Electronics té una pantalla tàctil compatible de 10 polsades amb una interfície d'HDMI a LVDS



Altres maquinari

- Ports GPIO, sèrie, SPI, I2C i JTAG
- Ve amb un connector soldat (P1) de 26 pins i un altre de 8 pins (P5. Tan sols està a la segona revisió)
- Treballa internament a 3,3V. Si es vol fer compatible amb nivells TTL (5V) cal un convertidor de nivells
- MIPI CSI-2 (Camera Serial Interface) al connector S5
- DSI (Display Serial Interface) al connector S2 per a LCD (no hi ha drivers actualment)
- La segona revisió té alguns canvis menors respecte a l'original

Connector GPIO

Versió 1 de la Raspberry Pi

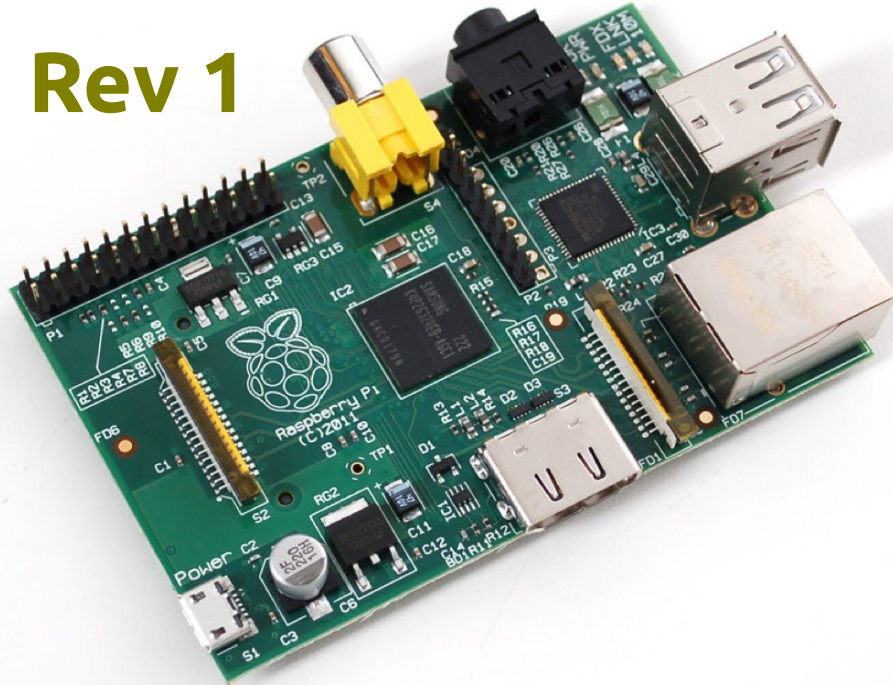
3.3V	1	2	5V
I2C0 SDA	3	4	DNC
I2C0 SCL	5	6	GROUND
GPIO4	7	8	UART TXD
DNC	9	10	UART RXD
GPIO 17	11	12	GPIO 18
GPIO 21	13	14	DNC
GPIO 22	15	16	GPIO 23
DNC	17	18	GPIO 24
SP10 MOSI	19	20	DNC
SP10 MISO	21	22	GPIO 25
SP10 SCLK	23	24	SP10 CE0 N
DNC	25	26	SP10 CE1 N

Versió 2 de la Raspberry Pi

3.3V	1	2	5V
I2C1 SDA	3	4	5V
I2C1 SCL	5	6	GROUND
GPIO4	7	8	UART TXD
GROUND		10	UART RXD
GPIO 17	11	12	GPIO 18
GPIO 27	13	14	GROUND
GPIO 22	15	16	GPIO 23
3.3V	17	18	GPIO 24
SP10 MOSI	19	20	GROUND
SP10 MISO	21	22	GPIO 25
SP10 SCLK	23	24	SP10 CE0 N
GROUND	25	26	SP10 CE1 N

Raspberry Pi Model B

Rev 1



Rev 2

Comparació d'arquitectures i famílies ARM

Arquitectura	Família
ARMv1	ARM1
ARMv2	ARM2, ARM3
ARMv3	ARM6, ARM7
ARMv4	StrongARM, ARM7TDMI, ARM9TDMI
ARMv5	ARM7EJ, ARM9E, ARM10E, XScale
ARMv6	ARM11, ARM Cortex-M
ARMv7	ARM Cortex-A, ARM Cortex-M, ARM Cortex-R
ARMv8	Not available yet. They will be 64 bits processors

ARMv5 Platforms

	Processor	RAM	NAND	Ethernet	USB	SATA
OLinuXino	Freescale i.MX233 454MHz	64MB		10/100	2	
OpenRD	Marvell Kirkwood 1.2GHz	512MB	512MB	Gigabit	7	2
Pogoplug Series 4	Marvell Kirkwood 800MHz	128MB	128MB	Gigabit	3	1
Pogoplug v2 (Pink/Gray)	Marvell Kirkwood 1.2GHz	256MB	128MB	Gigabit	4	
Seagate DockStar	Marvell Kirkwood 1.2GHz	128MB	256MB	Gigabit	4	
Seagate GoFlex Home	Marvell Kirkwood 1.2GHz	128MB	256MB	Gigabit	1	1
Seagate GoFlex Net	Marvell Kirkwood 1.2GHz	128MB	256MB	Gigabit	1	2
SheevaPlug	Marvell Kirkwood 1.2GHz	512MB	512MB	Gigabit	1	
TonidoPlug	Marvell Kirkwood 1.2GHz	512MB	512MB	Gigabit	1	
TonidoPlug 2	Marvell Armada 310 800MHz	512MB	512MB	Gigabit	1	1
ZyXEL NSA320	Marvell Kirkwood 1.2Ghz	512MB	128MB	Gigabit	3	2

ARMv6 Platforms

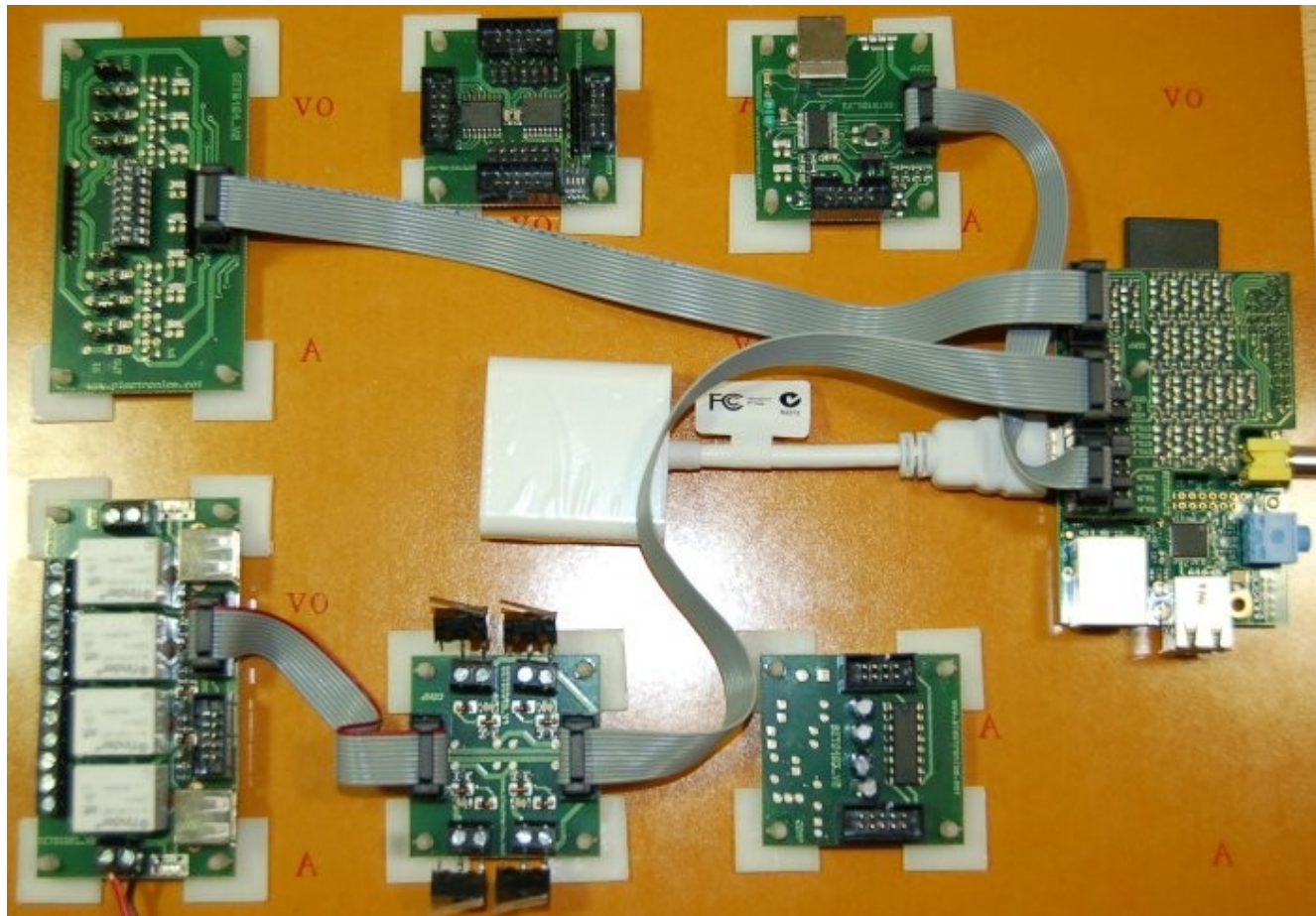
	Processor	RAM	NAND	Ethernet	USB	SATA
Pogoplug Pro/Video/v3	PLX 7820 700MHz Dual-core	128MB	128MB	Gigabit	4	1
Raspberry Pi	Broadcom BCM2835 700MHz	256MB		10/100	2 (100mA max)	

ARMv7 Platforms

	Processor	RAM	Ethernet	SD	USB	Wireless
BeagleBoard	TI OMAP 3530 720MHz	256MB	10/100	Full SD	1	
BeagleBoard-xM	TI DM3730 1GHz	512MB	10/100	Micro SD	4	
BeagleBone	TI AM3358 720Mhz	256MB	10/100	Micro SD	2	
CuBox	Marvell Armada 510 800MHz	1024MB	Gigabit	Micro SD	2	
Gumstix Overo	TI OMAP 35xx 600/720MHz	512MB	10/100	Micro SD	Exp	B/G, Bluetooth v2.0 + EDR
IGEP v2	TI DM3730 1GHz	512MB	10/100	Micro SD	2	B/G, Bluetooth v2.0 + EDR
Mele A100	Allwinner A10 1.0Ghz	512MB	10/100	Full SD	3	B/G/N
Nitrogen6X	Freescale i.MX6 Quad 1GHz	1024MB	Gigabit	Micro SD	2	B/G/N
PandaBoard	TI OMAP 4430 1GHz Dual-core	1024MB	10/100	Full SD	2	B/G/N, Bluetooth v2.1 + EDR
TrimSlice	NVIDIA Tegra 2, 1GHz Dual-core	1024MB	Gigabit	Full and Micro SD	4	Optional B/G/N, Optional Bluetooth

Maquinari

Maquinari

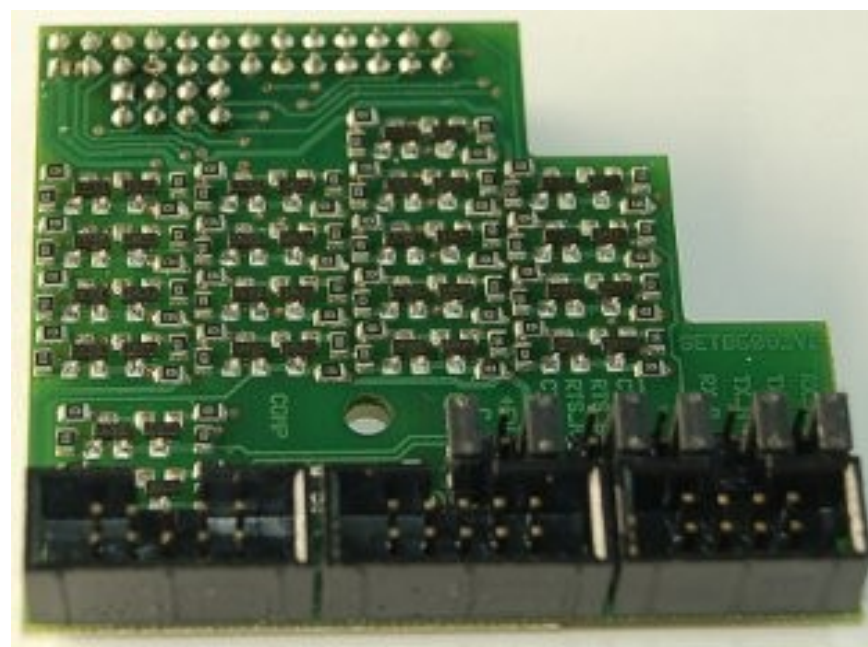


Conjunt d'una Raspberry Pi amb plaques de <http://www.electronics.cat/>

Maquinari

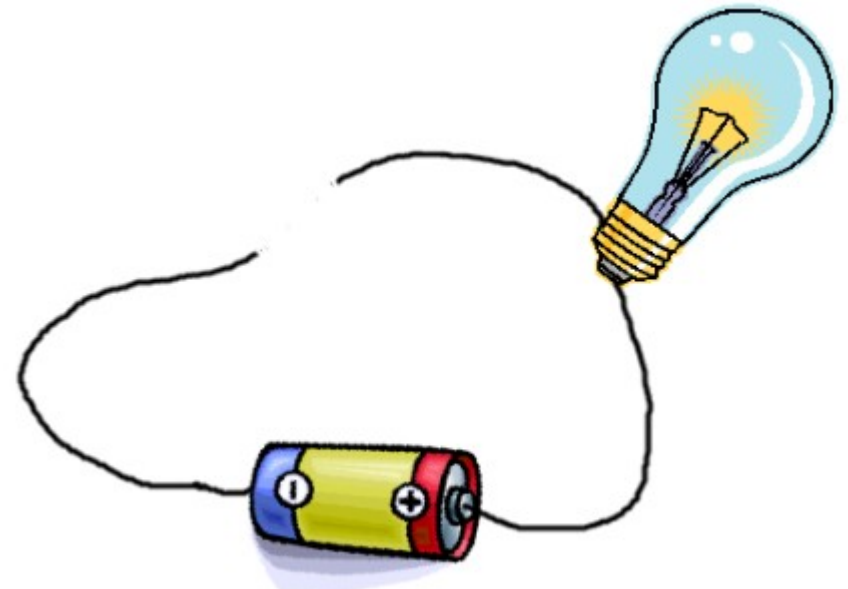
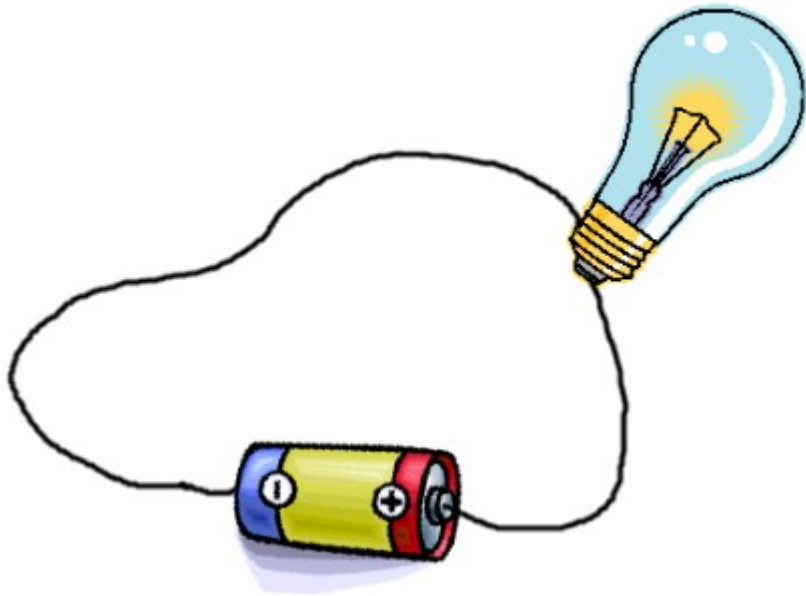
3Bpi - Pi de les Tres Branques

Placa traductora de nivells per a Raspberry Pi
Placa set05_08_v2



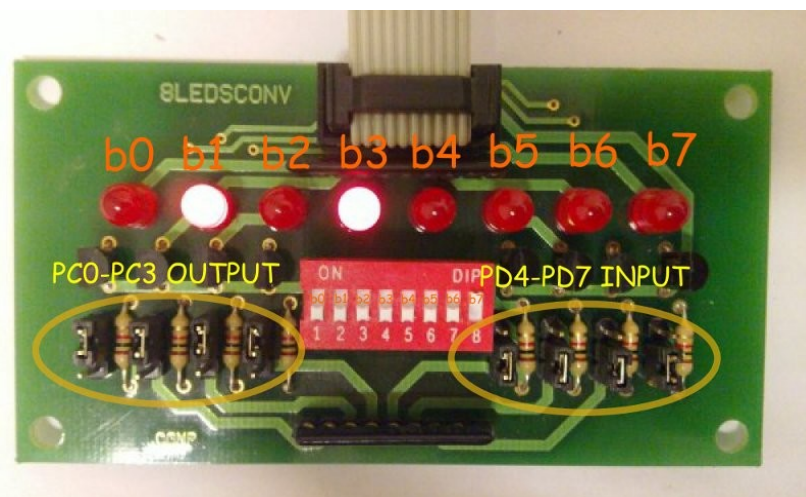
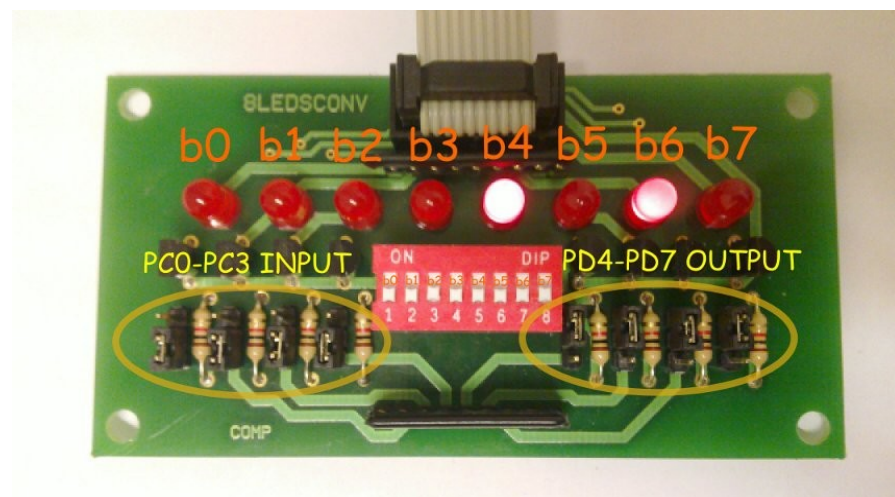
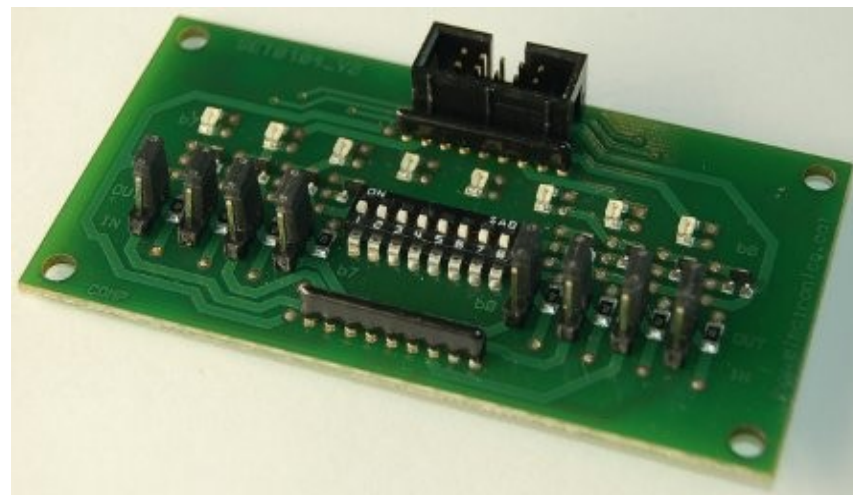
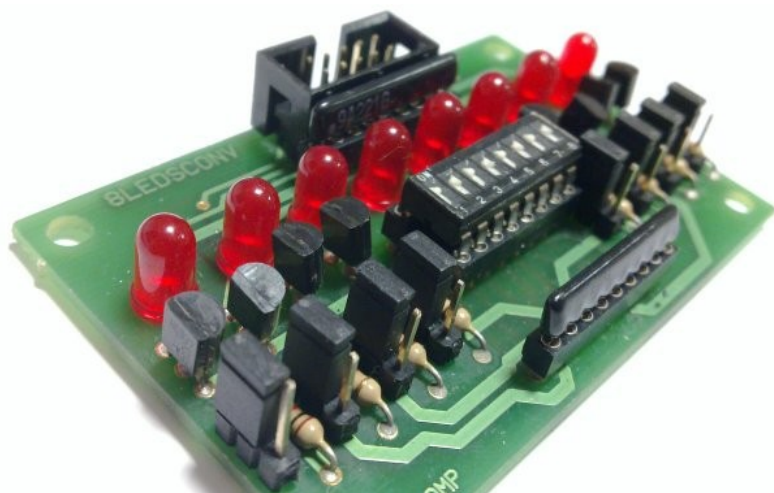
Maquinari

Concepte d'interruptor



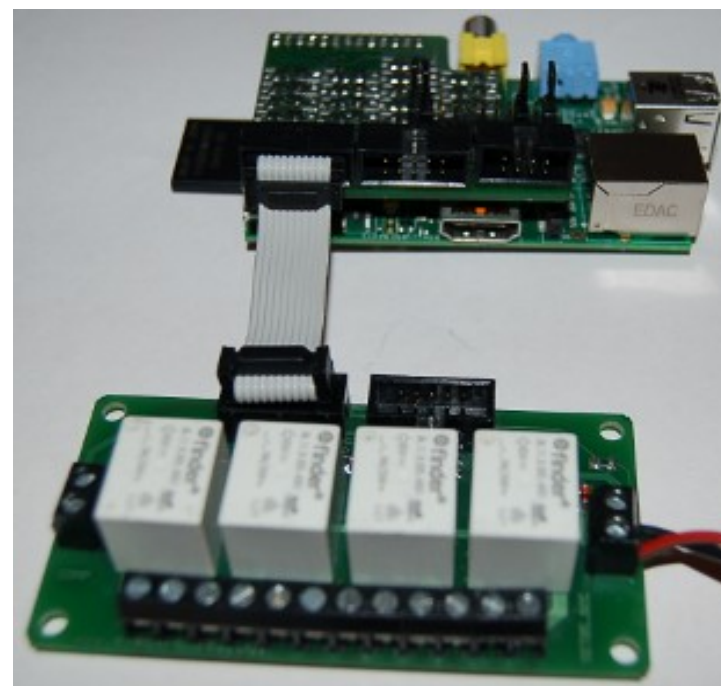
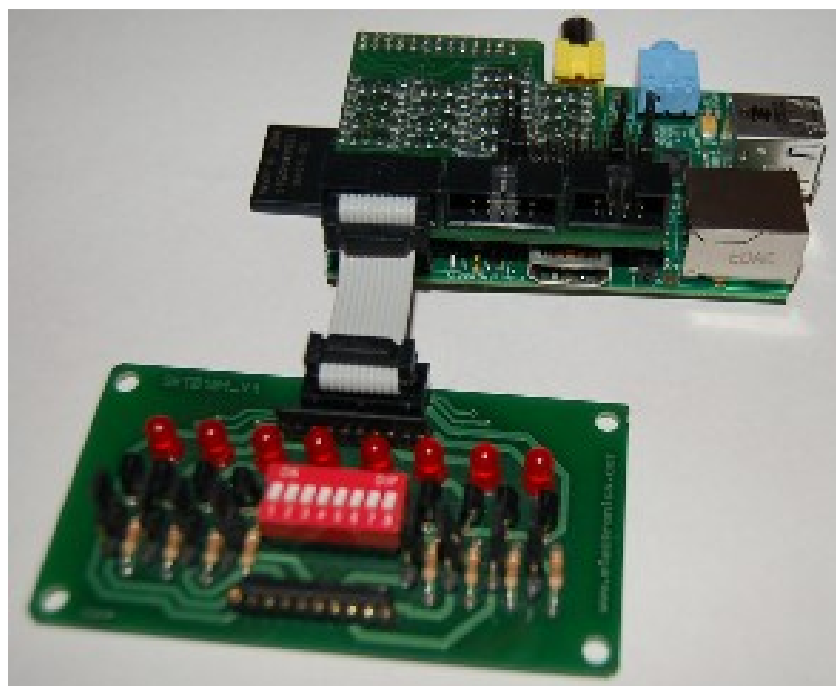
Maquinari

Leds i microrruptors - Placa set01_04



Maquinari

Raspberry Pi + 3Bpi + entrades/sortides



Vídeo explicatiu : <http://youtu.be/bUjgEGbktFM>

Maquinari

Concepte de relé

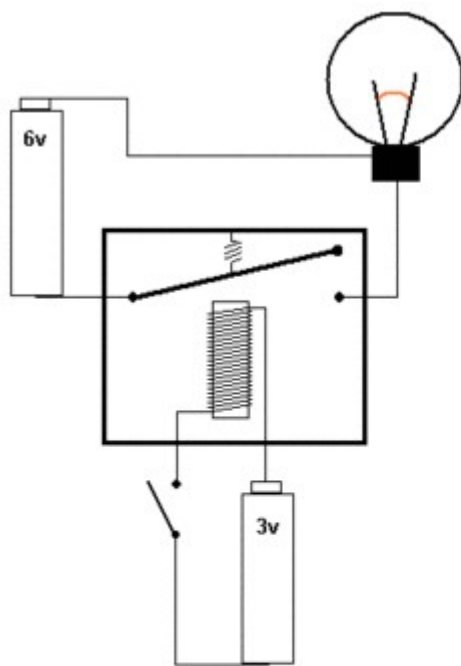


Figure 1: Relay off

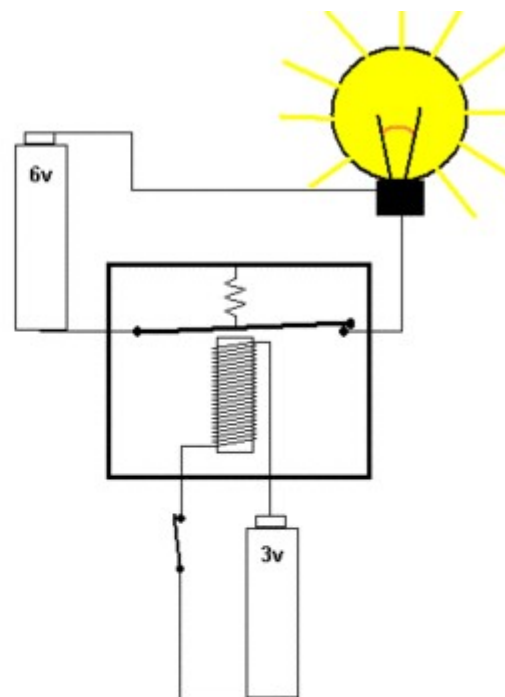
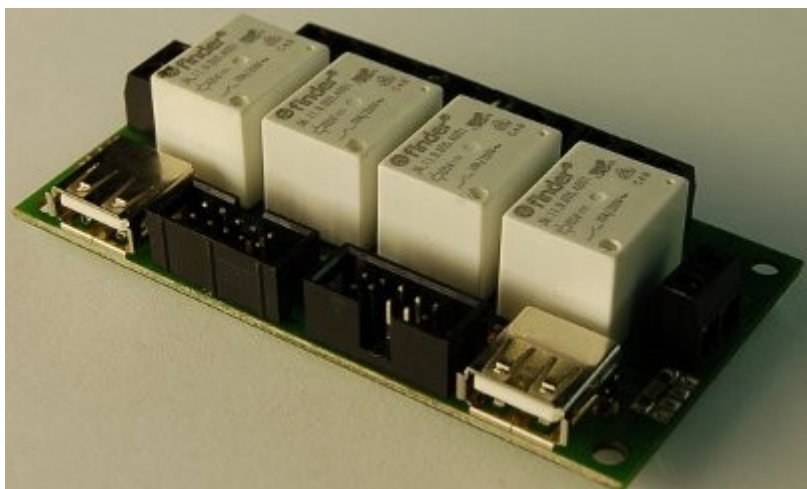


Figure 2: Relay on

Maquinari

4 Relés - Placa set05_02



Targeta SD per la demostració

Podeu baixar-vos la imatge de la targeta SD (2014-03-12-eCat.img de 4GB) des de :

<http://ves.cat/jpEE>

<https://mega.co.nz/#!/B0dFkQAQ!8Ro7Hka05tvvXGRRJBrJSh6euuVas5rIKKWKLp7ZP1I>

- En cas de que la vostra targeta estigui a /dev/sdb podeu copiar la imatge amb un dd:
sudo dd if=2014-03-12-eCat.img of=/dev/sdb bs=1M



Accès a la GPIO emprant Qt

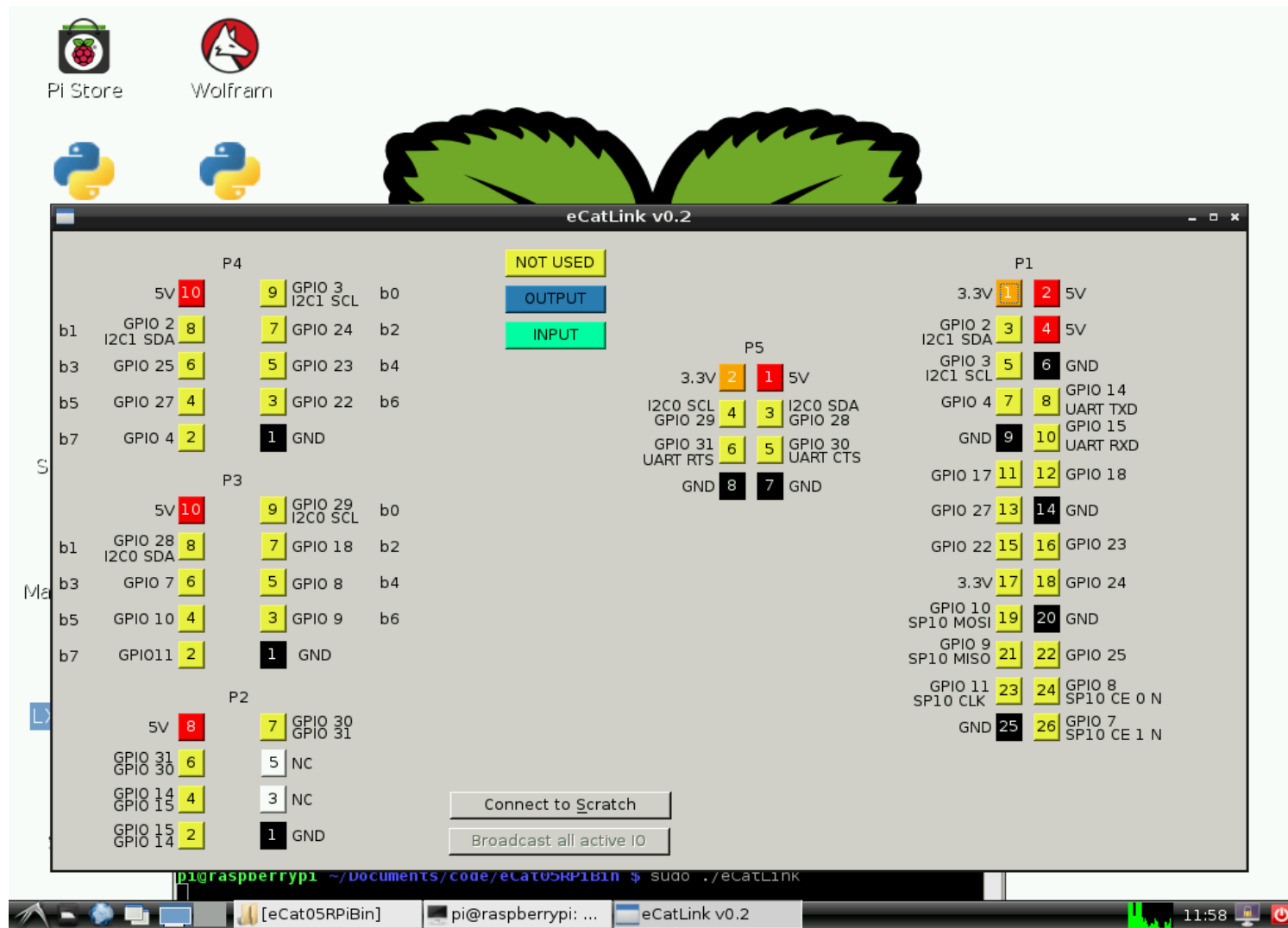


**Code less.
Create more.
Deploy everywhere.**

<http://qt-project.org/>

Accès a la GPIO emprant Qt

eCatLink



Accès a la GPIO emprant Qt eCatLink

eCatLink v0.2

P4

5V **10** ☒ **9** GPIO 3 I2C1 SCL b0

b1 GPIO 2 I2C1 SDA **8** ☒ ☒ **7** GPIO 24 b2

b3 GPIO 25 **6** ☒ ☐ **5** GPIO 23 b4

b5 GPIO 27 **4** ☐ ☐ **3** GPIO 22 b6

b7 GPIO 4 **2** ☐ **1** GND

NOT USED

OUTPUT

INPUT

P3

5V **10** ☐ **9** GPIO 29 I2C0 SCL b0

b1 GPIO 28 I2C0 SDA **8** ☐ ☐ **7** GPIO 18 b2

b3 GPIO 7 **6** ☐ ☐ **5** GPIO 8 b4

b5 GPIO 10 **4** ☐ ☐ **3** GPIO 9 b6

b7 GPIO11 **2** ☐ **1** GND

P2

5V **8** **7** GPIO 30 GPIO 31

GPIO 31 **6** **5** NC

GPIO 14 **4** **3** NC

GPIO 15 **2** **1** GND

P5

3.3V **2** **1** 5V

I2C0 SCL **4** **3** I2C0 SDA

GPIO 29 **4** **3** GPIO 28

GPIO 31 **6** **5** GPIO 30

UART RTS **6** **5** UART CTS

GND **8** **7** GND

P1

3.3V **1** **2** 5V

GPIO 2 **3** **4** 5V

I2C1 SDA **3** **4** 5V

GPIO 3 **5** **6** GND

I2C1 SCL **5** **6** GND

GPIO 4 **7** **8** GPIO 14

UART TXD **7** **8** GPIO 15

GPIO 15 **9** **10** UART RXD

GPIO 17 **11** **12** GPIO 18

GPIO 27 **13** **14** GND

GPIO 22 **15** **16** GPIO 23

3.3V **17** **18** GPIO 24

GPIO 10 **19** **20** GND

SP10 MOSI **19** **20** GND

GPIO 9 **21** **22** GPIO 25

SP10 MISO **21** **22** GPIO 25

GPIO 11 **23** **24** GPIO 8

SP10 CLK **23** **24** SP10 CE 0 N

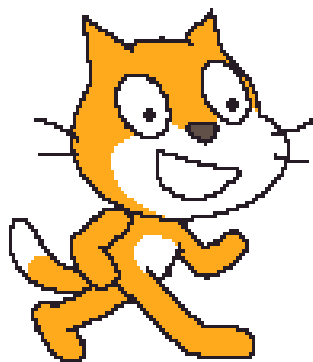
GND **25** **26** GPIO 7

SP10 CE 1 N **25** **26** SP10 CE 1 N

Connect to Scratch

Broadcast all active IO

SCRATCH

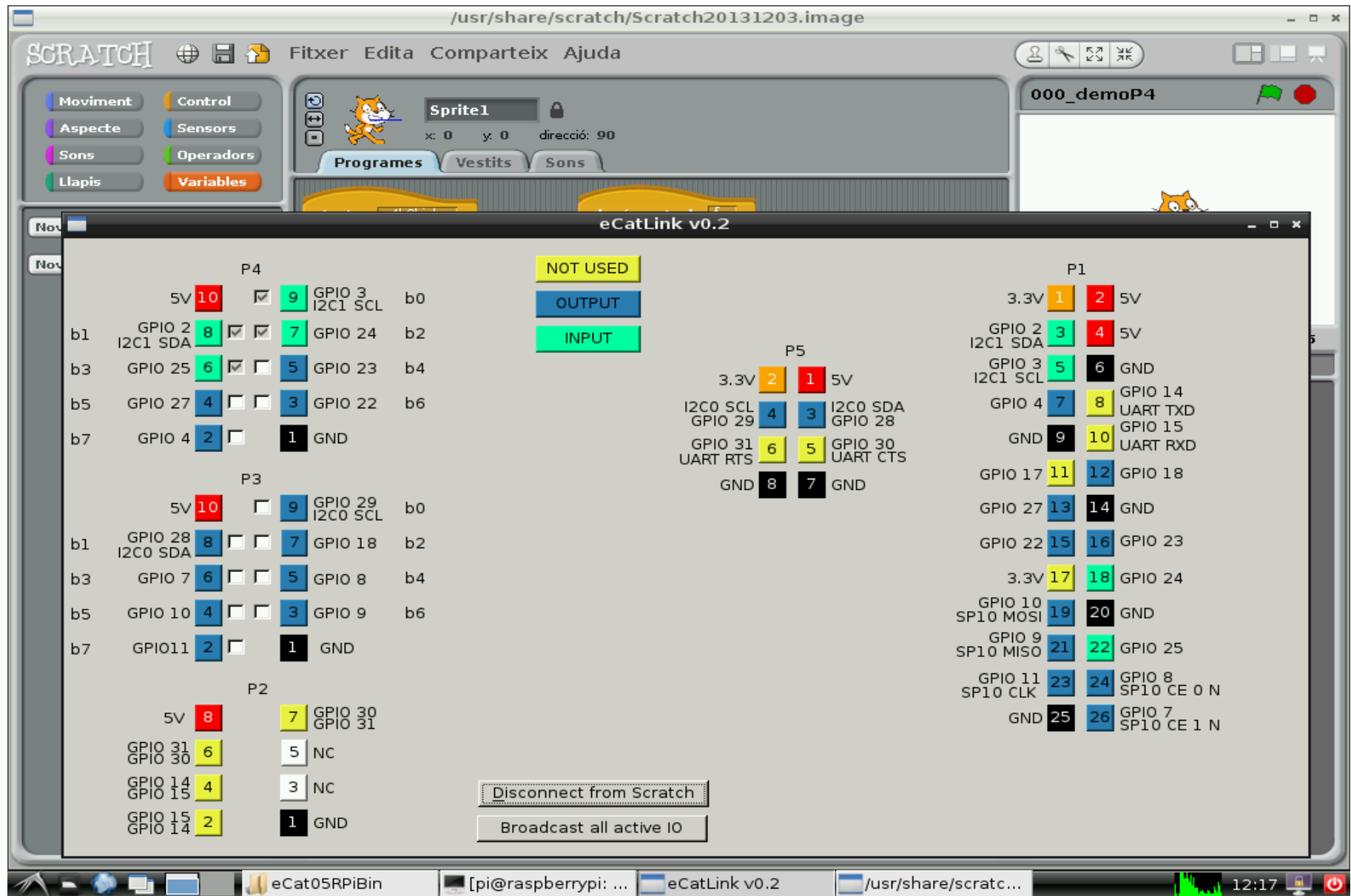


+



Maquinari

Fent ús de P4 - eCatLink



Maquinari

eCatLink



Accès a la GPIO emprant Python



Instal·lació del mòdul RPi.GPIO

\$sudo apt-get update

\$sudo apt-get install python-dev

\$sudo apt-get install python-rpi.gpio

\$sudo aptitude install python-serial

\$sudo aptitude install python-netifaces

\$sudo usermod -a -G dialout pi

<http://code.google.com/p/raspberry-gpio-python/wiki/BasicUsage>

Rpi.GPIO llegint entrades de P3

```
1  #!/usr/bin/env python
2
3  from time import sleep
4  import os
5  import RPi.GPIO as GPIO
6
7  GPIO.setmode(GPIO.BCM)
8  GPIO.setup(29, GPIO.IN)
9  GPIO.setup(28, GPIO.IN)
10 GPIO.setup(18, GPIO.IN)
11 GPIO.setup(7, GPIO.IN)
12
13 while True:
14     if ( GPIO.input(29) == False ):
15         print "GPIO 29 pressed"
16     if ( GPIO.input(28) == False ):
17         print "GPIO 28 pressed"
18     if ( GPIO.input(18) == False ):
19         print "GPIO 18 pressed"
20     if ( GPIO.input(7) == False ):
21         print "GPIO 7 pressed"
22     sleep(0.1);
23
```

Rpi.GPIO escrivint sortides a P3

```
1  #!/usr/bin/env python
2
3  from time import sleep
4  import os
5  import RPi.GPIO as GPIO
6
7  GPIO.setwarnings(False)
8
9  GPIO.setmode(GPIO.BCM)
10 GPIO.setup(8, GPIO.OUT)
11 GPIO.setup(10, GPIO.OUT)
12 GPIO.setup(9, GPIO.OUT)
13 GPIO.setup(11, GPIO.OUT)
14
15 GPIO.output(8,0)
16 GPIO.output(10,0)
17 GPIO.output(9,0)
18 GPIO.output(11,0)
19
```

```
19
20 while True:
21     GPIO.output(8,1)
22     sleep(0.5);
23     GPIO.output(8,0)
24     sleep(0.5);
25     GPIO.output(10,1)
26     sleep(0.5);
27     GPIO.output(10,0)
28     sleep(0.5);
29     GPIO.output(9,1)
30     sleep(0.5);
31     GPIO.output(9,0)
32     sleep(0.5);
33     GPIO.output(11,1)
34     sleep(0.5);
35     GPIO.output(11,0)
36     sleep(0.5);
```

Control d'un servo – bit7 de P4



Control d'un servo – bit7 de P4

```
1  #!/usr/bin/env python
2
3  from time import sleep
4  import os
5  import RPi.GPIO as GPIO
6
7  GPIO.setwarnings(False)
8
9  GPIO.setmode(GPIO.BCM)
10 GPIO.setup(4, GPIO.OUT)
11 p = GPIO.PWM(4,50)          # 50Hz
12 p.start(7.5)
13 try:
14     while True:
15         p.ChangeDutyCycle(7.5) # Neutral
16         sleep(1)
17         p.ChangeDutyCycle(12.5) # 180°
18         sleep(1)
19         p.ChangeDutyCycle(2.5) # 0°
20         sleep(1)
21
22 except KeyboardInterrupt:
23     GPIO.cleanup()
24
```

<https://dl.dropboxusercontent.com/u/65254823/oshw20140319/py02.py>

Adaptació de codi basat en exemples de <http://www.theraspberrypiguy.com/>

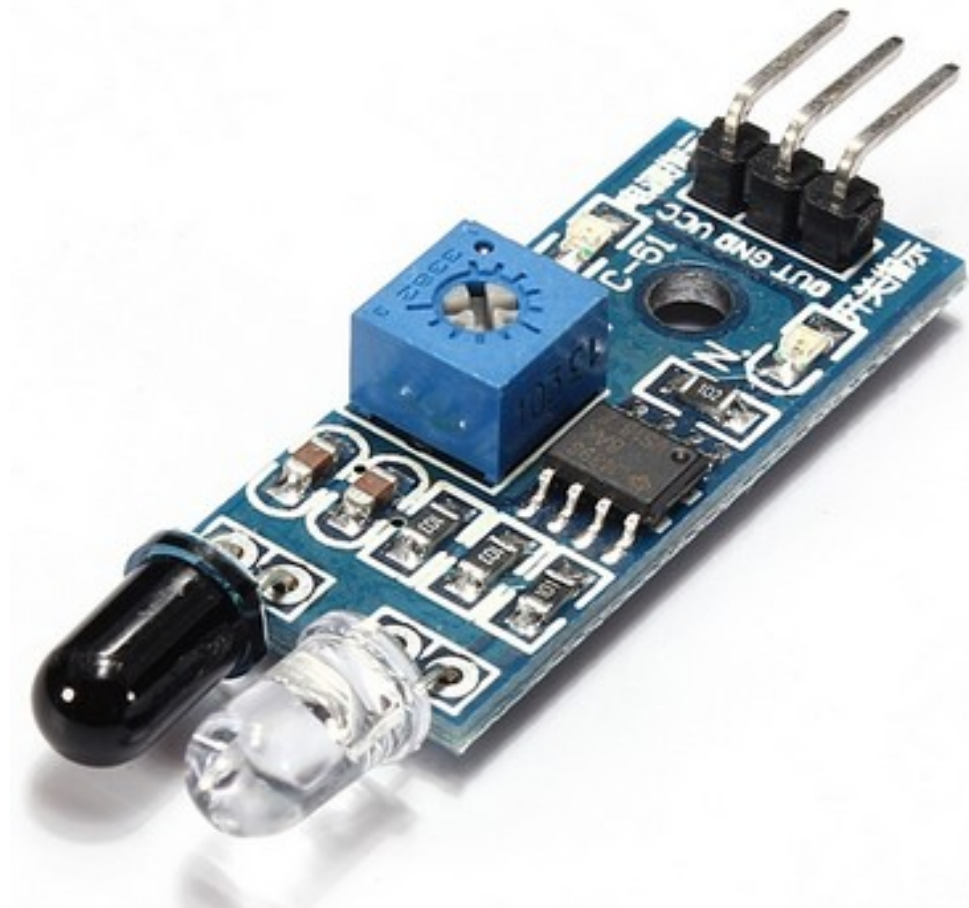
Lectura de sensor d'ultrasons



Lectura de sensor d'ultrasons

```
1  #!/usr/bin/python
2
3  def reading(sensor):
4      import time
5      import RPi.GPIO as GPIO
6
7      GPIO.setwarnings(False)
8      GPIO.setmode(GPIO.BCM)
9
10     if sensor == 0:
11         GPIO.setup(22,GPIO.OUT)
12         GPIO.setup(27,GPIO.IN)
13         GPIO.output(22, GPIO.LOW)
14         time.sleep(0.3)
15
16         GPIO.output(22, True)
17         time.sleep(0.00001)
18         GPIO.output(22, False)
19
20         while GPIO.input(27) == 0:
21             signaloff = time.time()
22         while GPIO.input(27) == 1:
23             signalon = time.time()
24
25         timepassed = signalon - signaloff
26         distance = timepassed * 17000
27
28         GPIO.cleanup()
29         return distance
30
31     else:
32         print "Incorrect usonic() function variable."
33
34 print reading(0)
35
```

Lectura de sensor infrarrojo



Lectura de sensor infrarroig

```
1  #!/usr/bin/env python
2
3  from time import sleep
4  import os
5  import RPi.GPIO as GPIO
6
7  GPIO.setmode(GPIO.BCM)
8  GPIO.setup(23, GPIO.IN)
9
10 while True:
11     if ( GPIO.input(23) == True ):
12         print "Tapat"
13         sleep(0.1);
14
```

<https://dl.dropboxusercontent.com/u/65254823/oshw20140319/py05.py>

Adaptació de codi basat en exemples de <http://www.theraspberrypiguy.com/>

Motor pas a pas



Motor pas a pas

```

1  #!/usr/bin/python
2  import RPi.GPIO as GPIO
3  import time
4
5  GPIO.setmode(GPIO.BCM)
6  ControlPin = [3,2,24,25]
7
8  for pin in ControlPin:
9      GPIO.setup(pin,GPIO.OUT)
10     GPIO.output(pin,0)
11
12  seq = [ [0,0,0,1],
13          [0,0,1,1],
14          [0,0,1,0],
15          [0,1,1,0],
16          [0,1,0,0],
17          [1,1,0,0],
18          [1,0,0,0],
19          [1,0,0,1] ]
20
21  for i in range(512):
22      for halfstep in range(8):
23          for pin in range(4):
24              GPIO.output(ControlPin[pin],seq[halfstep][pin])
25              time.sleep(0.001)
26
12  seq = [ [1,0,0,0],
13          [1,1,0,0],
14          [0,1,0,0],
15          [0,1,1,0],
16          [0,0,1,0],
17          [0,0,1,1],
18          [0,0,0,1],
19          [1,0,0,1] ]
20

```

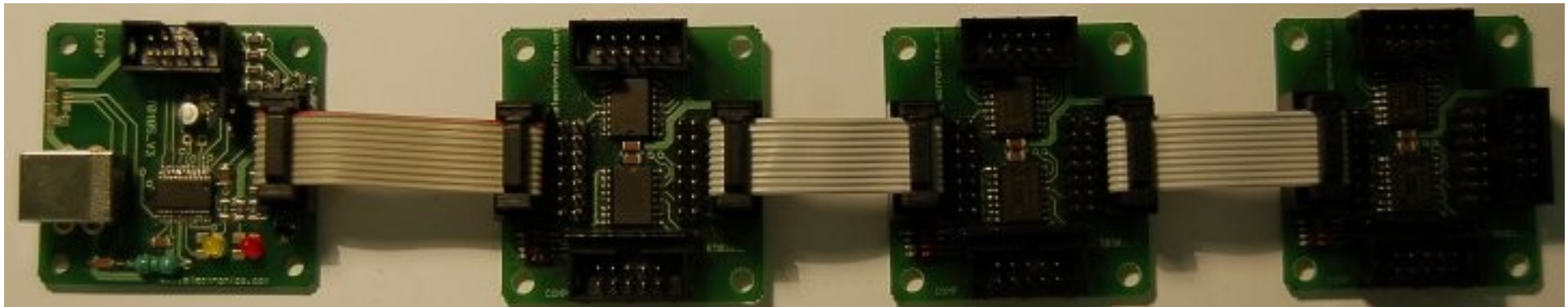
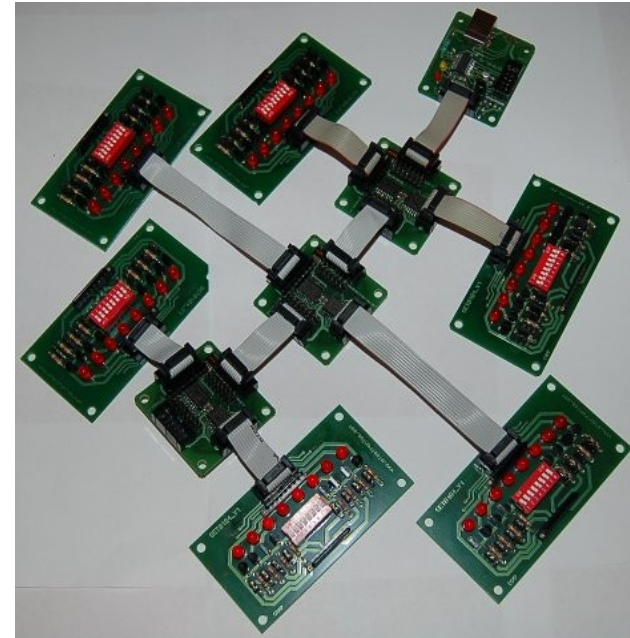
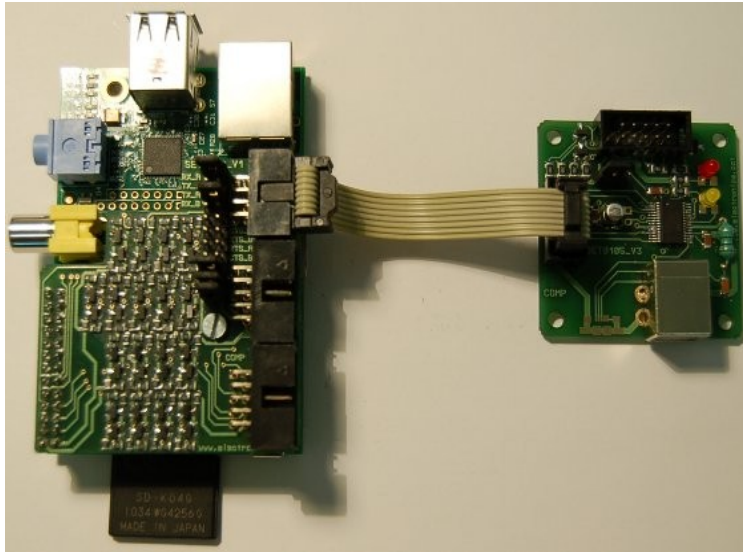
<https://dl.dropboxusercontent.com/u/65254823/oshw20140319/py04.py>

<https://dl.dropboxusercontent.com/u/65254823/oshw20140319/py04b.py>

Adaptació de codi basat en exemples de <http://www.theraspberrypiguy.com/>

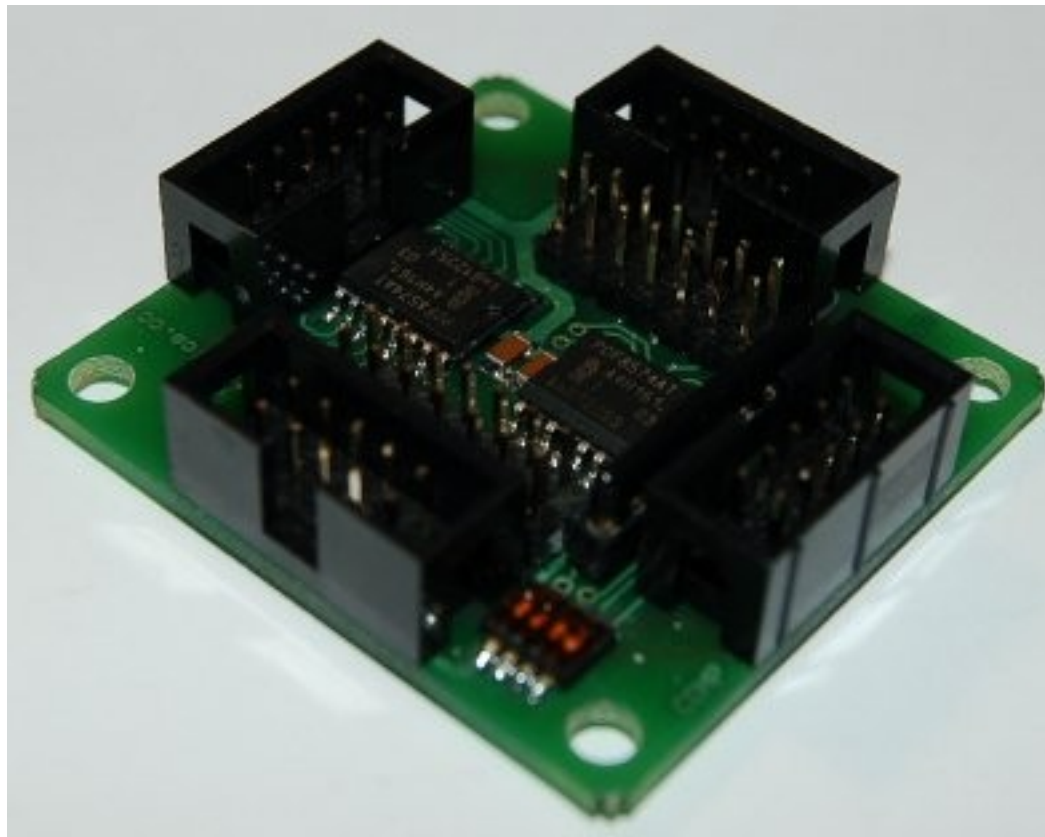
Comunicaciones I2C

Comunicaciones USB/TTL-I2C Placa set01_05



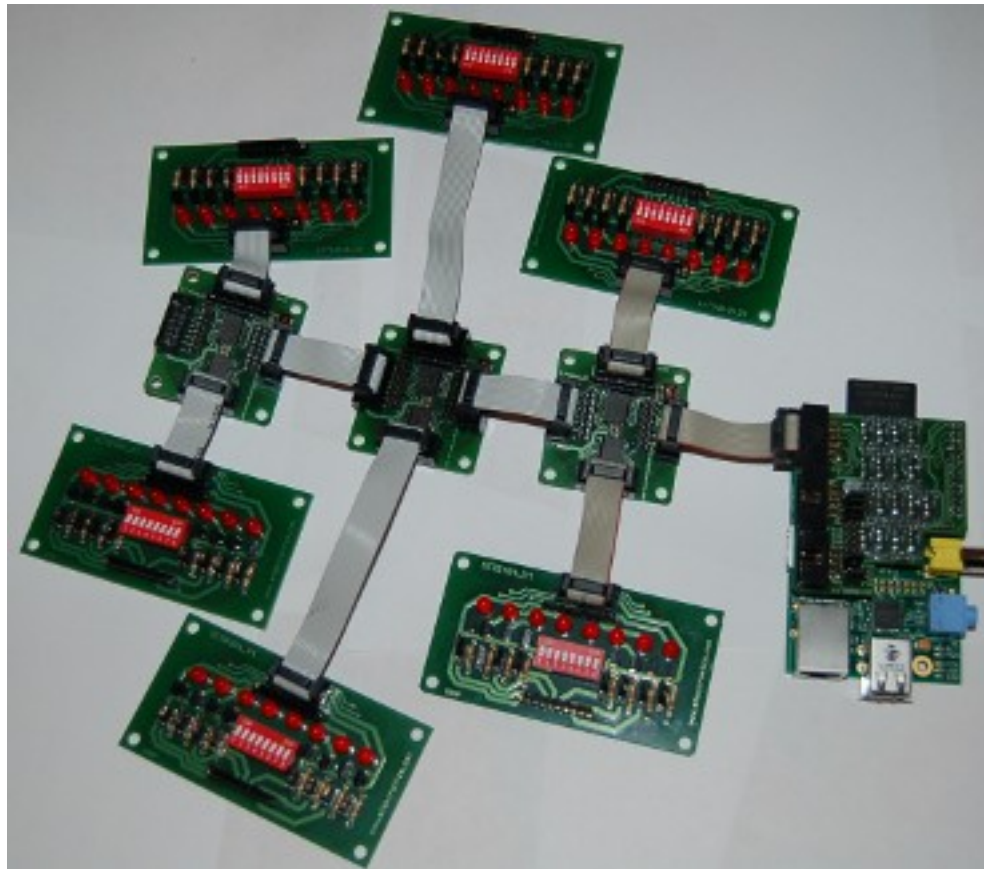
Comunicacions I2C

Expansor TWI (I2C) d'entrades/sortides digitals - set03_03



Maquinari

Raspberry Pi + 3Bpi + TWI (I2C)



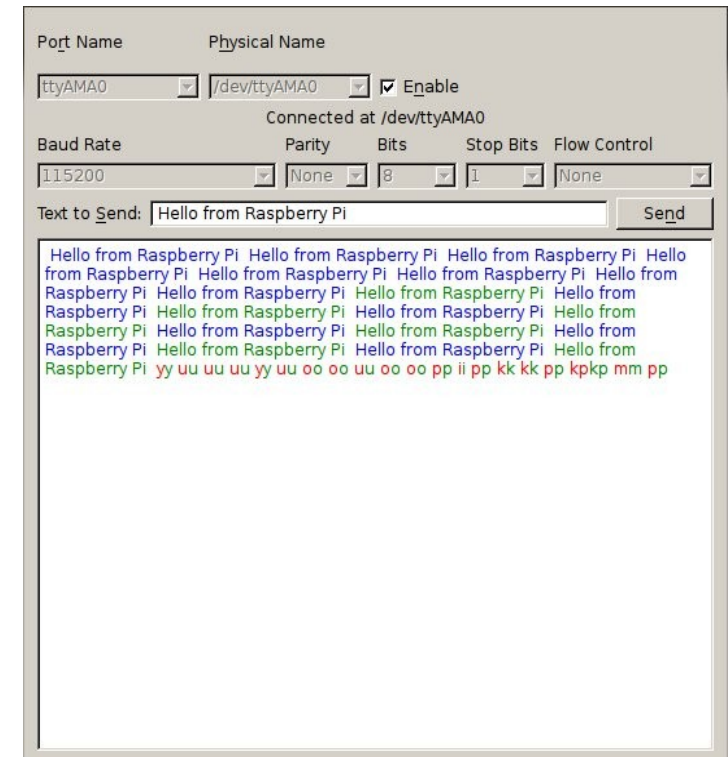
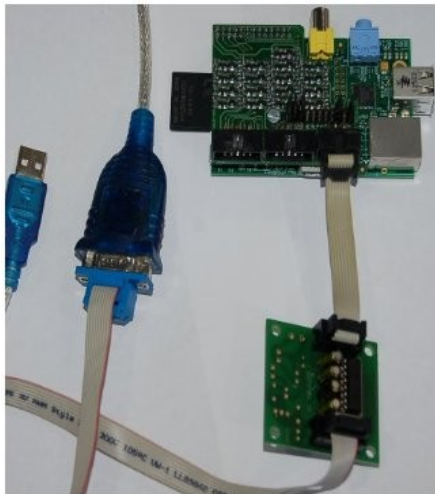
Vídeo explicatiu : <http://youtu.be/Was5QJDAGKU>

Codi de detecció d'elements TWI connectats : <http://www.electronics.cat/code/3Bpi/twi/twiDetect.c>

Codi per a que un led il·luminat doni dues voltes : <http://www.electronics.cat/code/3Bpi/twi/twi01.c>

Comunicacions sèrie

Raspberry Pi + 3Bpi + Comunicació sèrie

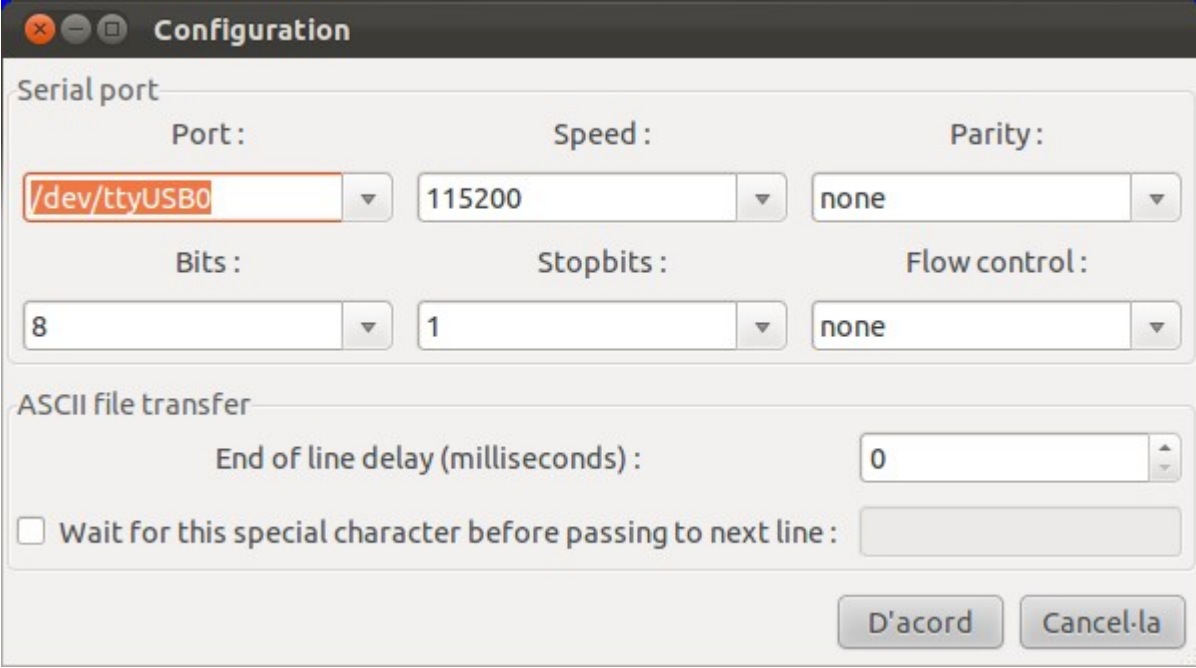


Explicació de com fer la instal·lació del qtTerm a la Raspberry Pi :

<http://electronics.cat/code/3Bpi/rs232/setRs232up.html>

Comunicacions sèrie

GtkTerm



The image shows a 'Configuration' dialog box for a serial port. It is divided into two main sections: 'Serial port' and 'ASCII file transfer'. The 'Serial port' section contains six dropdown menus arranged in two rows. The first row has 'Port:' (set to '/dev/ttyUSB0'), 'Speed:' (set to '115200'), and 'Parity:' (set to 'none'). The second row has 'Bits:' (set to '8'), 'Stopbits:' (set to '1'), and 'Flow control:' (set to 'none'). The 'ASCII file transfer' section has a label 'ASCII file transfer' and a text input field 'End of line delay (milliseconds):' with the value '0'. Below this is a checkbox 'Wait for this special character before passing to next line:' which is unchecked, followed by an empty text input field. At the bottom right are two buttons: 'D'acord' and 'Cancel·la'.

Configuration

Serial port

Port :

Speed :

Parity :

Bits :

Stopbits :

Flow control :

ASCII file transfer

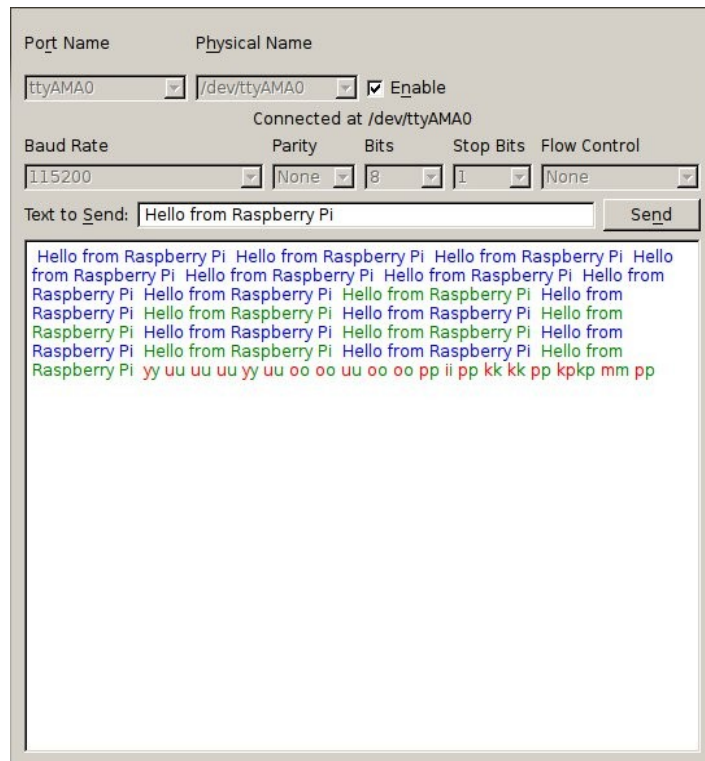
End of line delay (milliseconds) :

☐ Wait for this special character before passing to next line :

D'acord Cancel·la

Comunicacions sèrie

qtTerm



<http://code.google.com/p/qtterm/>

Port sèrie a la Raspberry Pi

```
1  #!/usr/bin/python
2
3  import serial
4
5  print "Python serial port sample"
6
7  rpiSerialPort = '/dev/ttyAMA0'
8  ser = serial.Serial(rpiSerialPort, 115200)
9
10 ser.write("\n\rFrom Raspberry Pi\n\r")
11 ser.close()
12
```

Exemple de maquinari lliure emprant programari lliure

Trepant / fresadora - qtCnc

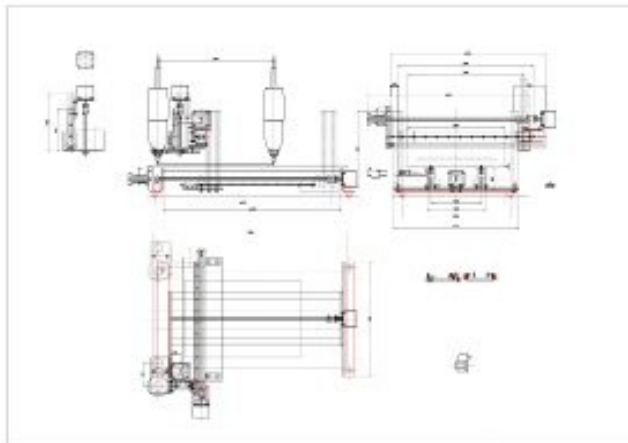
- Programari allotjat a sourceforge.net :
(<http://sourceforge.net/projects/qtCnc/files/>)



Home 

Name ▾	Modified ▾	Size ▾	Downloads ▾
 Desktop Software	2011-12-21		
 FirmwareAtmega	2011-12-21		

Totals: 2 Items



Maquinari lliure a la formació formal (CFGGS-DAM)

DAM2

4 M7: Desenvolupament d'interfícies UF1. Disseny i implementació d'interfícies: 79 hores UF2. Preparació i distribució d'aplicacions: 20 hores	3 M10: Sistemes ERP UF1. Sistemes ERP-CRM. Implantació: 33 hores UF2. Sistemes ERP-CRM. Explotació i adequació: 33 hores	4 M8: Programació multimèdia i dispositius mòbils UF1. Desenvolupament d'aplicacions per mòbils: 45 hores UF2. Programació multimèdia: 24 hores UF3. Desenvolupament de jocs per mòbils: 30 hores	4 Mòdul 6: Accés a dades UF1. Persistència en fitxers: 24 hores UF2. Persistència en BDR-BDOR-BDOO: 27+33 hores UF3. Persistència en BD natives XML: 24 hores UF4. Components d'accés a dades: 24 hores	5 M03. Programació II UF4. Programació orientada a objectes. Fonaments: 30+5h UF5. POO. Llibries de classes fonamentals: 30+10h UF7. Aplicacions distribuïdes i computació en el núvol: 18h UF6. POO. Introducció a la persistència en BD: 29h	3 M05. Entorns de desenvolupament UF1. Desenvolupament de programari: 20h UF2. Optimització del programari: 20h	3 M9: Programació de serveis i processos UF1. Seguretat i criptografia: 25 hores UF2. Processos i flus: 37 hores UF3. Sòcols i serveis: 37 hores
---	---	--	--	---	--	---

































JESUÏTES El Clot
 Escola del Clot

C/València, 680 08027 Barcelona
 93.351.60.11 info.clot@fje.edu

Torn de preguntes ...



... i sessió pràctica.

COMIAT

Presentació descarregable a : <http://ves.cat/jvGs>

Correu electrònic de contacte : jordibinefa@electronics.cat

twitter



<https://twitter.com/JordiBinefa>

<https://twitter.com/electronicscat>



<http://es.linkedin.com/pub/jordi-binefa/13/717/90b>

Plaques aviat disponibles a :

<http://suministros-industriales-barcelona.com>

<http://www.faelectronic.com>

<http://www.electronics.cat>

Moltes gràcies per la vostra assistència