

# Maquinari lliure amb Ubuntu

Comunicació entre Ubuntu i Arduino  
emprant plaques d'electronics.cat

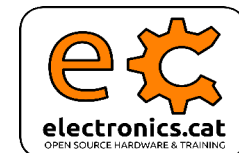


9 / maig / 2015

*Jordi Binefa i Martínez*

Responsable d'R+D+i a

Professor de cicles formatius a



JESUÏTES El Clot  
Escola del Clot



# Maquinari lliure amb Ubuntu

## Comunicacions sèrie - minicom

Permisos d'usuari per a emprar el port sèrie :

```
sudo usermod -a -G dialout $USER
```

(Cal que sortiu i torneu a entrar a la sessió per a que els permisos siguin efectius)

Instal·lació del programa de comunicacions minicom :

```
sudo aptitude install minicom
```

Ús del minicom :

```
minicom -b 9600 -o -D /dev/ttyUSB0
```

Per a sortir del minicom : Prémer les tecles 'Ctrl' i 'A' simulatàniament i sense deixar anar, prémer 'Z', després l'opció 'X' i seleccionar 'Yes'



# Maquinari lliure amb Ubuntu

## Comunicacions sèrie - minicom

Llistat de dispositius connectats al port USB :

**lsusb**

Dues sortides típiques quan es connecten Arduino Nano a l'USB :

```
Bus 003 Device 002: ID 0403:6001 Future Technology  
Devices International, Ltd FT232 USB-Serial (UART) IC
```

```
Bus 003 Device 003: ID 1a86:7523 QinHeng Electronics  
HL-340 USB-Serial adapter
```

Llistat de dispositius connectats al port USB :

**ls /dev/ttyU\* -ls**

```
0 crw-rw---- 1 root dialout 188, 0 mai 7 16:52 /dev/ttyUSB0  
0 crw-rw---- 1 root dialout 188, 1 mai 7 16:53 /dev/ttyUSB1
```

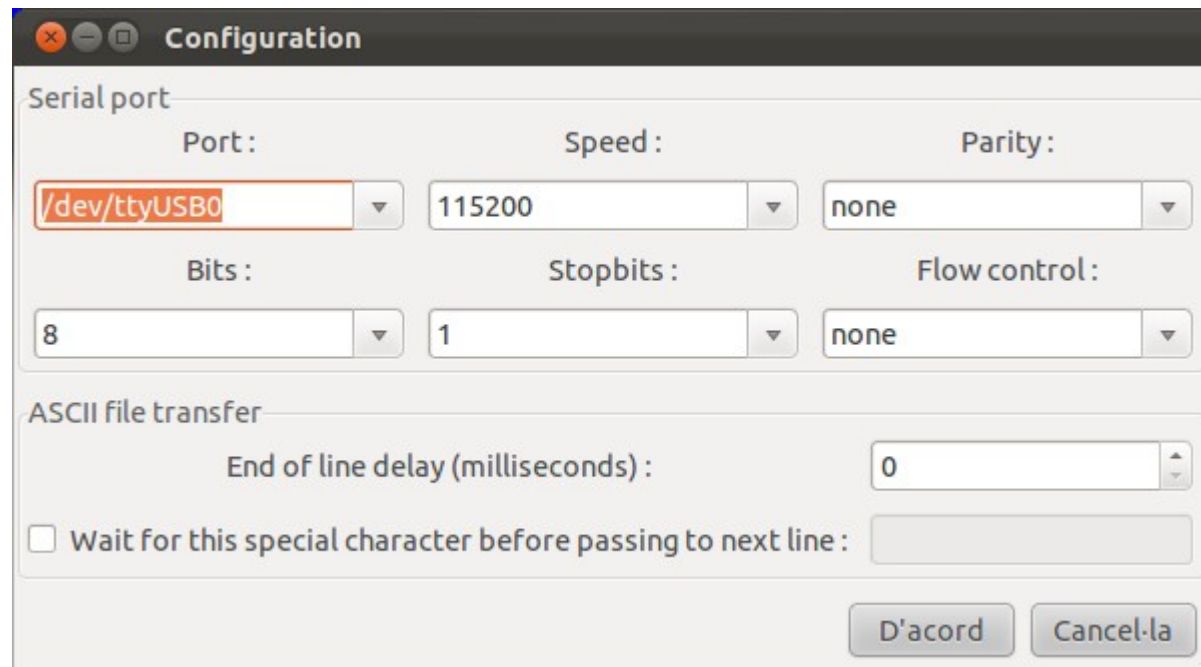


# Maquinari lliure amb Ubuntu

## Comunicacions sèrie - GtkTerm

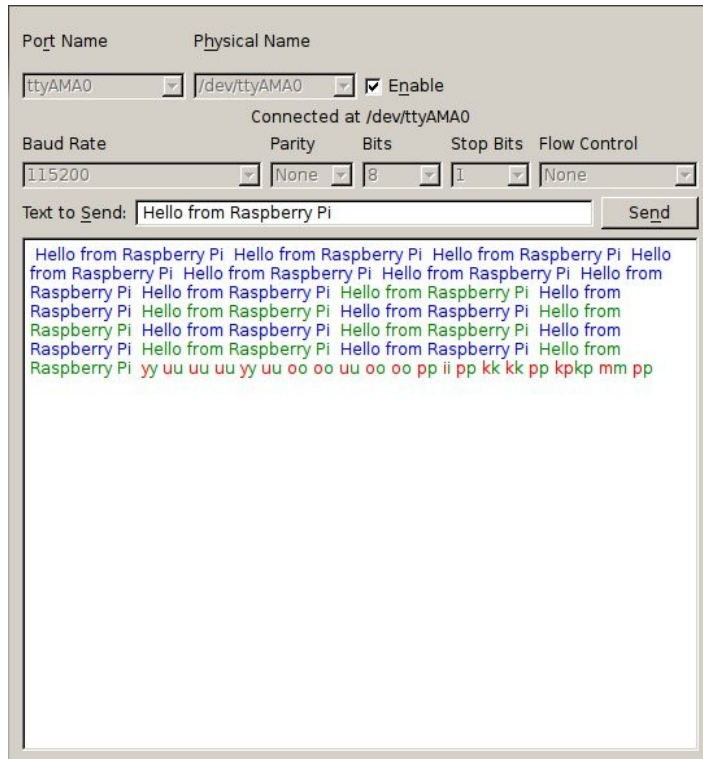
Instal·lació del programa de comunicacions GtkTerm :

```
sudo aptitude install gtkterm
```



# Maquinari lliure amb Ubuntu

## Comunicacions sèrie - qtTerm



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



# Maquinari lliure amb Ubuntu

## Instal·lació de més programes

Instal·lació de paquets relacionats amb Python :

```
sudo aptitude install python-dev python-serial  
python-netifaces python-pip pyqt4-dev-tools
```

Instal·lació de biblioteques de Python complementàries :

```
sudo pip install pyfirmata
```

Instal·lació d'Arduino :

```
sudo aptitude install arduino
```

Instal·lació de biblioteques complementàries per l'Arduino (**ecat.h**) :

```
wget http://www.electronics.cat/doc/arduinoLib/ecat.zip
```

Descomprimiu ecat.zip i moveu la carpeta ecat generada a la carpeta de biblioteques d'Arduino

```
sudo mv ecat /usr/share/arduino/libraries
```



# Maquinari lliure amb Ubuntu

## Instal·lació de més biblioteques per a l'IDE d'Arduino

Instal·lació de la biblioteca complementària per l'Arduino **ecat.h** :

```
wget http://www.electronics.cat/doc/arduinoLib/ecat.zip
```

Descomprimiu **ecat.zip** i moveu la carpeta ecat generada a la carpeta de biblioteques d'Arduino

```
sudo mv ecat /usr/share/arduino/libraries
```

Instal·lació de la biblioteca complementària per l'Arduino **UIPEthernet.h** :

```
wget http://www.electronics.cat/doc/arduinoLib/UIPEthernet.tar.gz
```

Descomprimiu **UIPEthernet.tar.gz** i moveu la carpeta **UIPEthernet** generada a la carpeta de biblioteques d'Arduino :

```
sudo mv UIPEthernet /usr/share/arduino/libraries
```

Instal·lació de la biblioteca complementària per l'Arduino **SCoop.h** :

```
wget http://www.binefa.cat/php/arduino/ardublock/SCoop.tar.gz
```

Descomprimiu **SCoop.tar.gz** i moveu la carpeta **SCoop** generada a la carpeta de biblioteques d'Arduino :

```
sudo mv SCoop /usr/share/arduino/libraries
```



# Maquinari lliure amb Ubuntu

## Instal·lació d'ArduBlock

Des del navegador web descarregueu-vos **ardublock-beta-20140828** :  
<http://sourceforge.net/projects/ardublock/files/ardublock-beta-20140828.jar/download>

A les versions de l'IDE d'Arduino prèvies a la versió 1.6.1 :

Copieu **ardublock-beta-20140828.jar** a : ~/sketchbook/tools/ArduBlockTool/tool

A la versió de l'IDE d'Arduino 1.6.1 o superior :

Copieu **ardublock-beta-20140828.jar** a : ~/Arduino/tools/ArduBlockTool/tool





# Maquinari lliure amb Ubuntu

## Instal·lació de l'Scratch for Arduino (S4A)

Paquet de l'**S4A** per a Debian / Ubuntu

Es pot instal·lar fent servir el gestor de paquets per defecte o escrivint al terminal :

```
sudo dpkg -i S4A16.deb
```



# Maquinari lliure amb Ubuntu

## Càrrega del protocol Firmata a l'Arduino

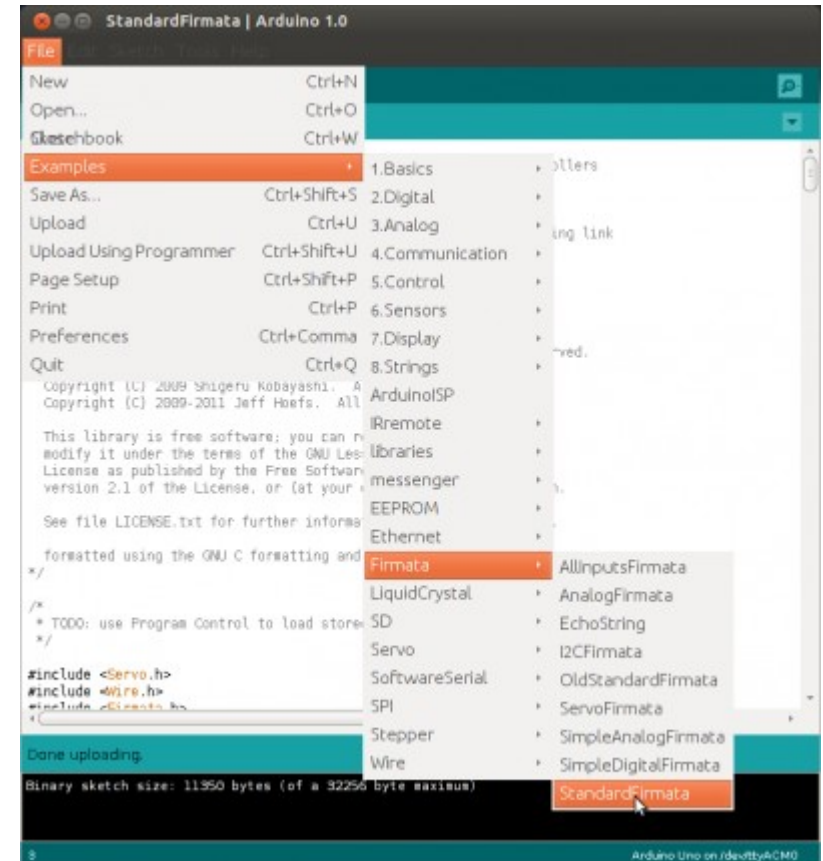
Connecteu la Raspberry Pi a la placa nano-eCat (la que conté l'Arduino Nano) mitjançant el cable USB-A a miniUSB.

A la Raspberry Pi aneu a Menú / Electrònica / Arduino IDE.

A l'Arduino IDE aneu a Fitxer / Exemples / Firmata / Standard Firmata.

Un cop l'Arduino IDE té carregat el programa Standard Firmata, premeu a la icona de la fletxa que apunta a la dreta (quan passeu el ratolí per sobre hi sortirà el text "Puja" a la seva dreta).

Un cop hi surti el text "Pujada enllestida" de color negre sobre fons verd a la part de sota de l'IDE, ja està preparat l'Arduino Nano per establir comunicacions seguint el protocol Firmata.





# Maquinari lliure amb Ubuntu

## Prova del protocol Firmata entre l'Arduino i Ubuntu

Al terminal d'Ubuntu us podeu descarregar el codi :

```
wget http://www.electronics.cat/doc/arduino/exFirmata03P2.py.tar.gz
```

Descomprimir-ho :

```
tar -xvzf exFirmata03P2.py.tar.gz
```

Executar-ho :

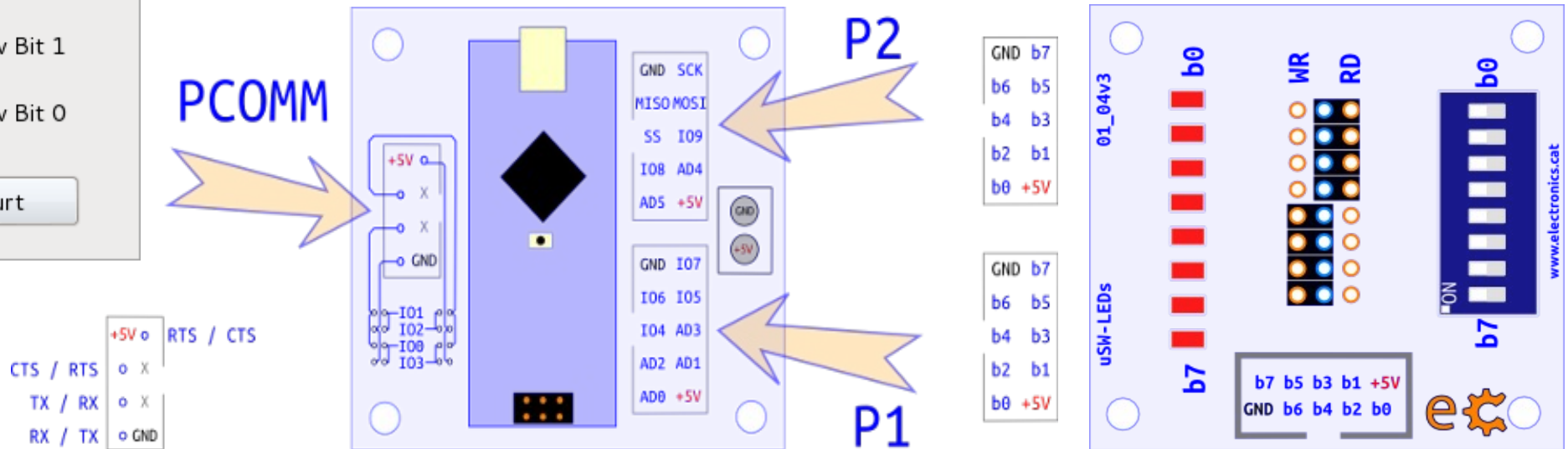
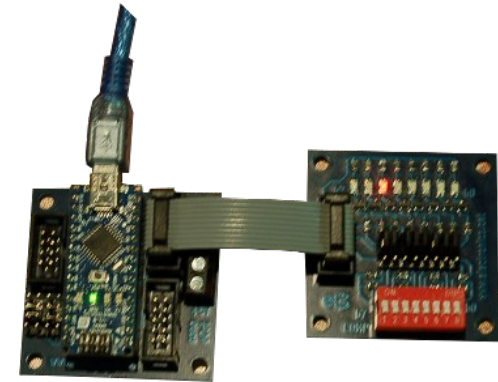
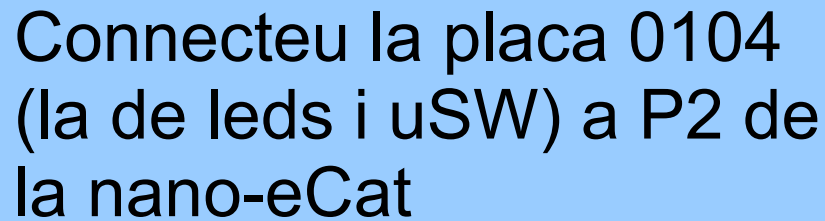
```
./exFirmata03P2.py
```

Us sortirà la finestra de la dreta amb diverses caselles de selecció. Seleccioneu i deseccioneu la casella corresponent a Led Bit 7 i observeu com canvia d'estat (il·luminat / apagat) el led integrat a l'Arduino.

Checkbox
<input checked="" type="checkbox"/> Led Bit 7
<input type="checkbox"/> Led Bit 6
<input checked="" type="checkbox"/> Led Bit 5
<input checked="" type="checkbox"/> Led Bit 4
<input type="checkbox"/> uSw Bit 3
<input type="checkbox"/> uSw Bit 2
<input type="checkbox"/> uSw Bit 1
<input type="checkbox"/> uSw Bit 0

Surt

# Prova del protocol Firmata entre l'Arduino i Ubuntu





# Maquinari lliure amb Ubuntu

**BUS** : Sistema digital que transfereix dades entre dispositius electrònics

GND	b7
b6	b5
b4	b3
b2	b1
b0	+5V

Cable pla  
5x2 a 5x2  
(0104A)



CTS / RTS

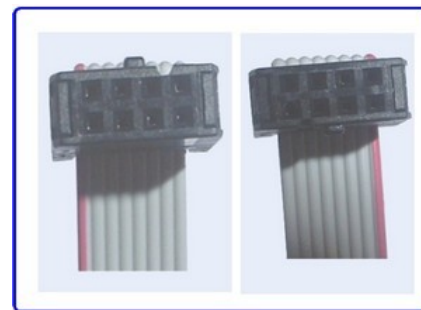
TX / RX

RX / TX

+5V	o
o	X
o	X
o	GND

CTS / RTS

Cable pla  
4x2 a 4x2  
(0311)



**eCat System** : Dues tipologies de BUS, dades (8 bits) i comunicacions

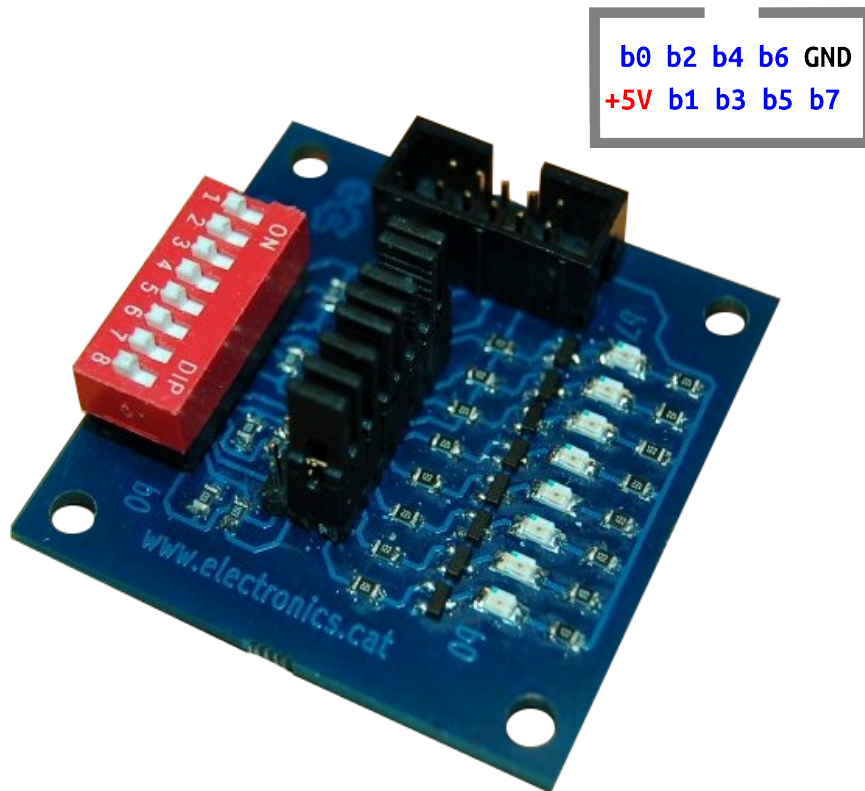


# Maquinari lliure amb Ubuntu

## Plaques amb sistema de bus lliure eCat

### uSW – LEDs

01\_04 v3



**Placa configurable d'entrades / sortides.**

**La configuració de cada bit es fa mitjançant un pont (jumper).**

**Les entrades es fan per microrruptor (ON : zero lògic, OFF : u lògic).**

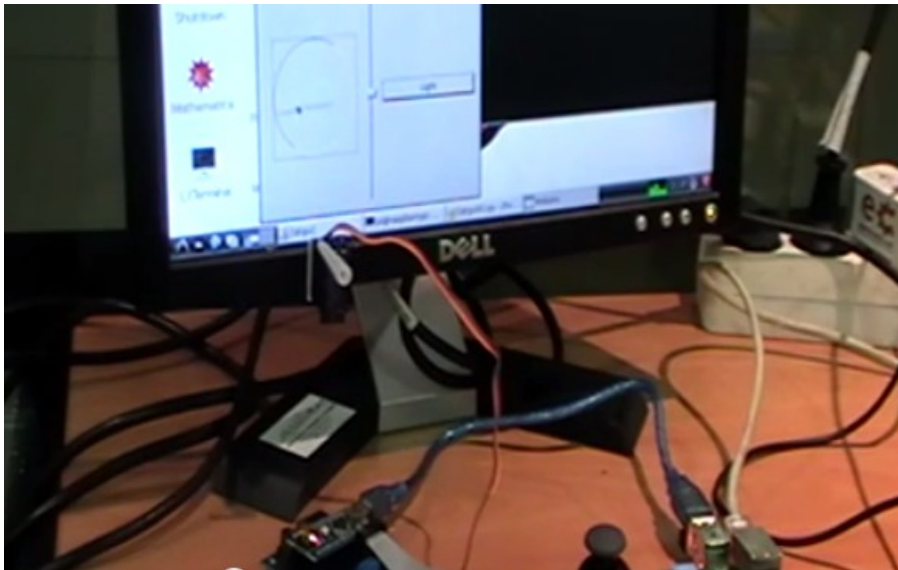
**Les sortides es visualitzen mitjançant leds (apagat : zero lògic, encès : u lògic).**





# Maquinari lliure amb Ubuntu

## Prova del protocol Firmata entre l'Arduino i Ubuntu

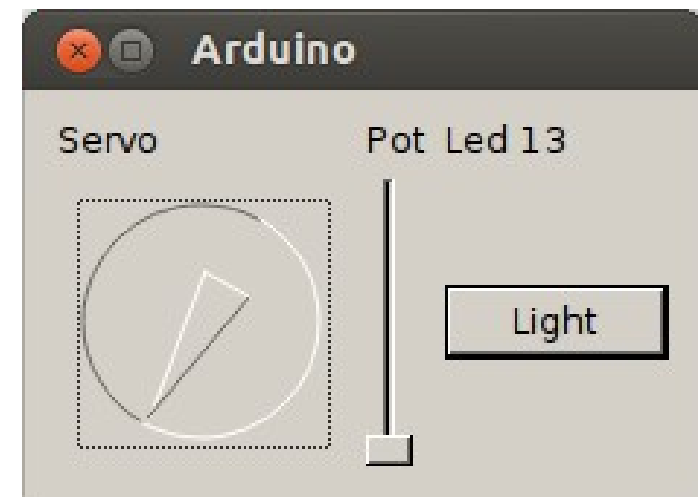
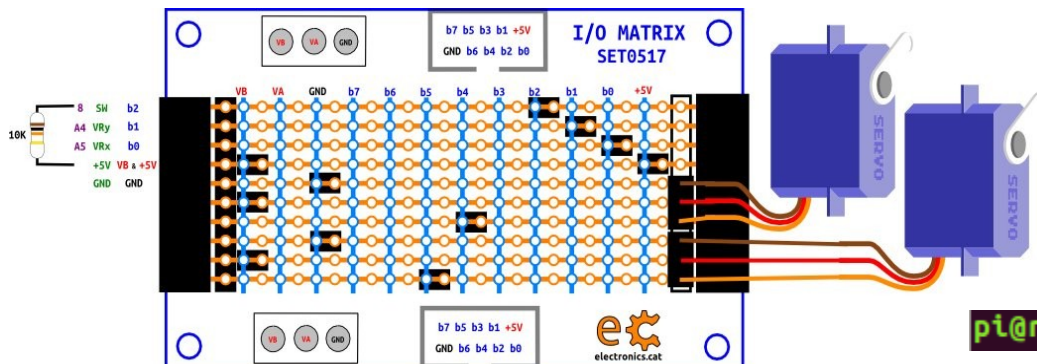


Vídeo :

<https://youtu.be/2kSSPvfXqug>

Més detallat :

<http://binefa.cat/blog/?p=102>



```
pi@raspberrypi ~/Documents/codis/firmata $ ./qtgui02.py
```

<http://www.binefa.cat/php/raspberryPi/python/pyQt/qtgui02.py.tar.gz>



# Maquinari lliure amb Ubuntu

## Protocol Firmata



<http://firmata.org/wiki/V2.3ProtocolDetails>

Vídeo de com funciona firmata : <https://youtu.be/l4jFQIoY9ZY>





# Maquinari lliure amb Ubuntu

## Ús de pyFirmata

### Usage

Basic usage:

```
>>> from pyfirmata import Arduino, util
>>> board = Arduino('/dev/tty.usbserial-A6008rIF')
>>> board.digital[13].write(1)
```

To use analog ports, it is probably handy to start an iterator thread. Otherwise the board will keep sending data to your serial, until it overflows:

```
>>> it = util.Iterator(board)
>>> it.start()
>>> board.analog[0].enable_reporting()
>>> board.analog[0].read()
0.661440304938
```

: . Eg. `a:0:i` for analog 0 as input or `d:3:p` for digital pin 3 as pwm.:

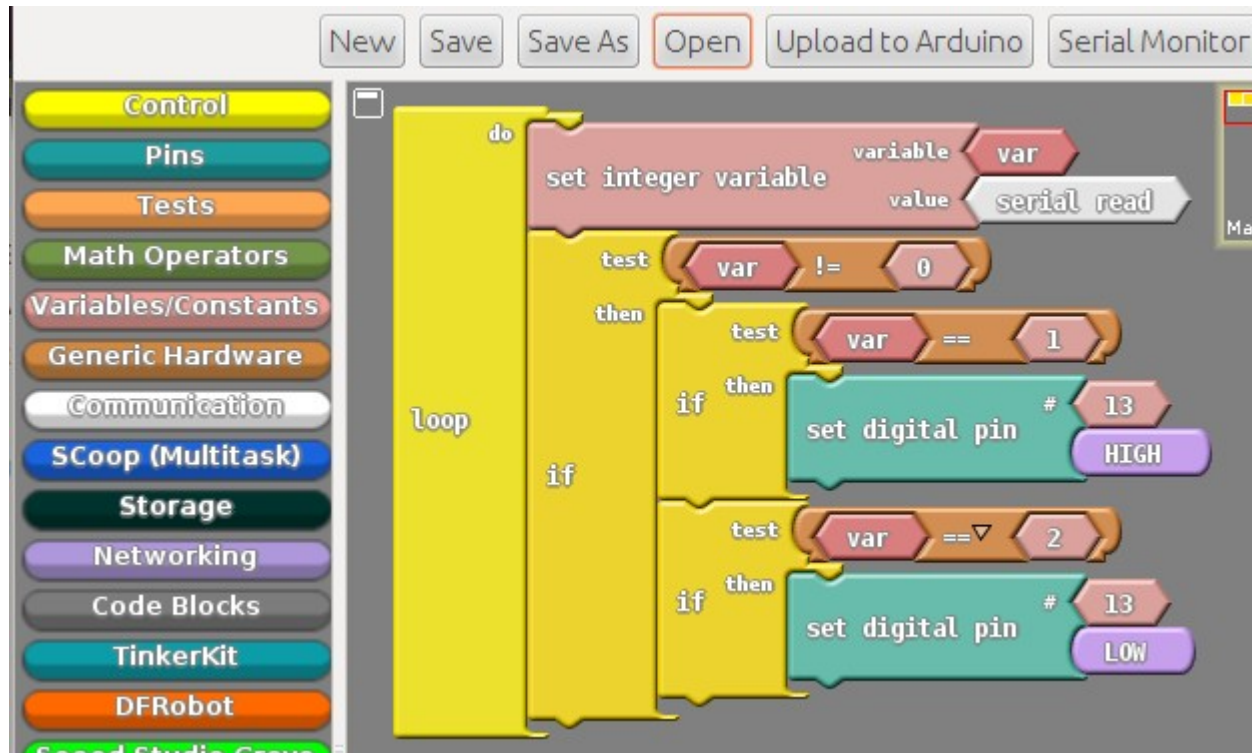
```
>>> analog_0 = board.get_pin('a:0:i')
>>> analog_0.read()
0.661440304938
>>> pin3 = board.get_pin('d:3:p')
>>> pin3.write(0.6)
```

<https://github.com/tino/pyFirmata>



# Maquinari lliure amb Ubuntu

## Ús d'ArduBlock



Projecte ArduBlock : <http://blog.ardublock.com/>

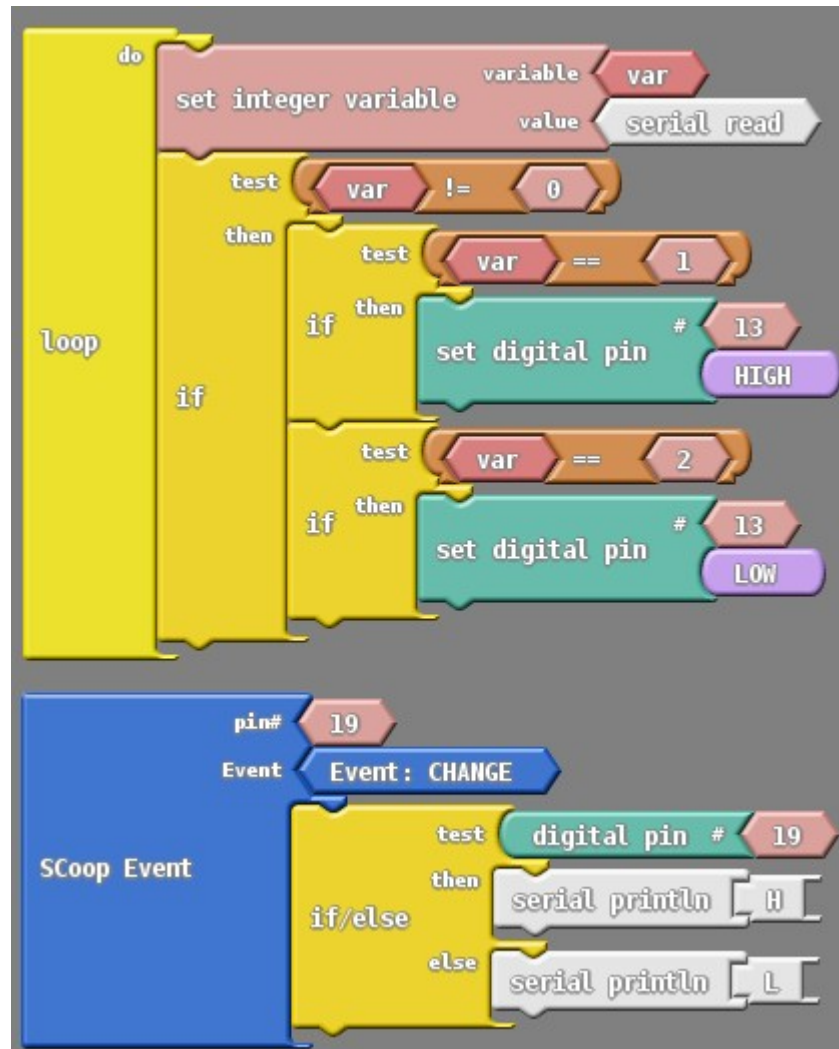
La lectura de dades pel port sèrie de la versió actual d'ArduBlock es limita a la lectura de números enters. En cas de no rebre res l'ArduBlock interpreta que està llegint un zero.

<http://www.electronics.cat/doc/arduino/ardublock/abCom00.abp>



# Maquinari lliure amb Ubuntu

## Ús d'ArduBlock



<http://www.electronics.cat/doc/arduino/ardublock/abCom01.abp>

<http://binefa.cat/blog/?p=94> i <http://binefa.cat/blog/?p=96>



# Maquinari lliure amb Ubuntu

## Ús d'ArduBlock

Baixeu-vos aquests dos codis :

```
wget http://www.electronics.cat/doc/arduino/ardublock/abCom00.abp
```

```
wget http://www.electronics.cat/doc/arduino/ardublock/abCom01.abp
```

Un cop està en marxa l'Arduino IDE aneu a Eines / ArduBlock

Un cop esteu dins d'ArduBlock premeu el botó **Open** i obriu l'arxiu **abCom00.abp** que està a la ruta `/home/pi/Documents/codis/ardublock`

Premeu el botó **Upload to Arduino**

Premeu el botó **Serial Monitor** per a interactuar amb l'Arduino



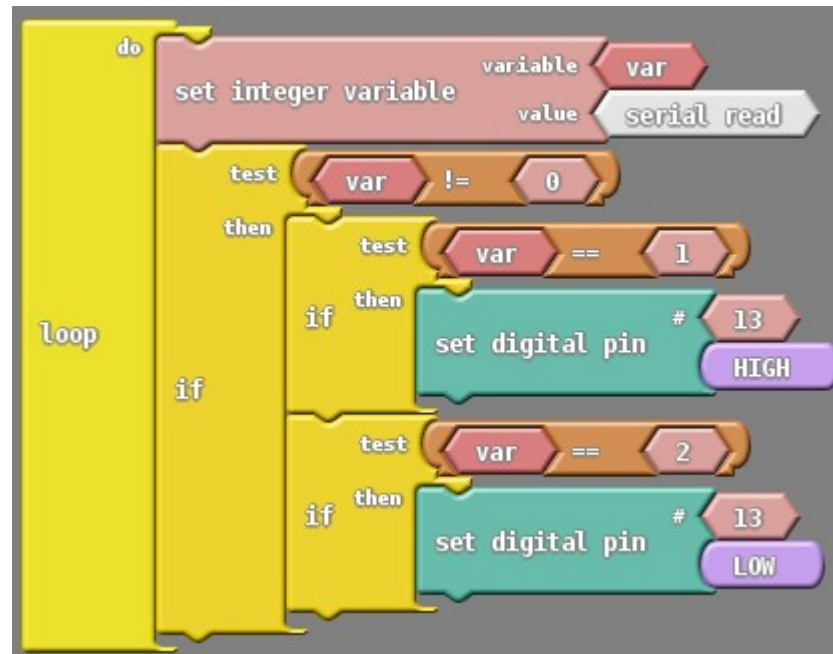


# Maquinari lliure amb Ubuntu

## Ús d'ArduBlock

Amb el **Serial Monitor** en marxa escriviu a l'esquerra del botó **Envia** el número **1** i premeu el botó **Envia**. Observeu com s'il·lumina el led propi (13) de l'Arduino.

Amb el **Serial Monitor** en marxa escriviu a l'esquerra del botó **Envia** el número **2** i premeu el botó **Envia**. Observeu com s'apaga el led propi (13) de l'Arduino.





# Maquinari lliure amb Ubuntu

## Ús d'ArduBlock

Ara tanqueu el **Serial Monitor** i prement el botó **Open** obriu l'arxiu **abCom01.abp** que està a la ruta `/home/pi/Documents/codis/ardublock`

Premeu el botó **Upload to Arduino**

Premeu el botó **Serial Monitor** per a tornar a interactuar amb l'Arduino

Quan envieu un 1 o un 2 es comporta igual que abans

Quan canvieu d'estat el microrruptor corresponent al bit 0 de la placa 0104 rebreu **H** (High : nivell alt) o **L** (Low : nivell baix)





# Maquinari lliure amb Ubuntu

## Ús del minicom

Aprofitant que està carregat l'Arduino amb un programa que empra comunicacions (trametent un 1 s'il·lumina un led, trametent un 2 s'apaga i movent el microrruptor l'Arduino tramet a la Raspberry Pi una H o una L), farem ús del programa minicom per a establir les comunicacions.

És important assegurar-se de que cap altre programa està fent ús del canal sèrie. Ni l'Arduino IDE ni l'ArduBlock.

```
minicom -b 9600 -o -D /dev/ttyUSB0
```

```
Welcome to minicom 2.6.1

OPTIONS: I18n
Compiled on Apr 28 2012, 19:24:31.
Port /dev/ttyUSB0

Press CTRL-A Z for help on special keys

L
H
```

Prement 1 s'il·lumina un led, trametent un 2 s'apaga (quan escrivim no hi surt res pel minicom perquè l'Arduino no fa eco del que li trametem) i movent el microrruptor l'Arduino tramet a la Raspberry Pi una H o una L. Per sortir-ne : Ctrl A + Z, X i Yes





# Maquinari lliure amb Ubuntu

## Ús del port sèrie des de Python

```
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
```

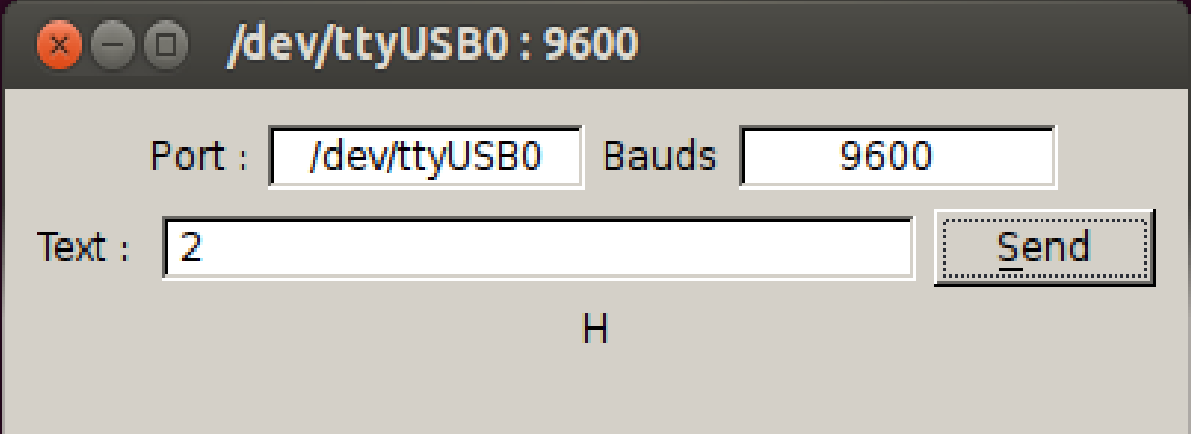




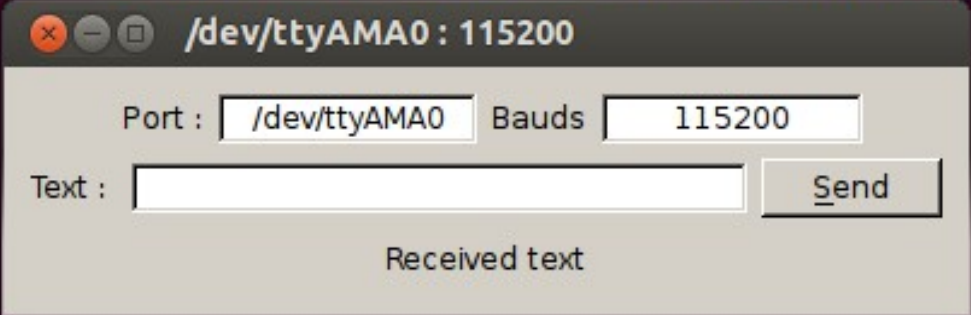
# Maquinari lliure amb Ubuntu

## Ús d'un programa amb interfície gràfica

```
pi@raspberrypi ~ $ cd Documents/codis/tty01/  
pi@raspberrypi ~/Documents/codis/tty01 $ ./tty01.py  
L  
H  
[ ]
```



```
pi@raspberrypi ~/Documents/codis/tty01 $ ./tty01.py /dev/ttyAMA0 115200  
[ ]
```



<http://www.binefa.cat/php/RPi/tty01.py.zip>



# Maquinari lliure amb Ubuntu

## Codi d'Arduino

```
/*  
serialWrRd01.ino  
  
http://www.binefa.cat/php/arduino/serialWrRd01.zip  
*/  
  
void setup() {  
  // Velocitat de la comunicacio en bauds  
  Serial.begin(9600);  
}  
  
void loop() {  
  byte byteLlegit;  
  /* Si hi ha quelcom a llegir: */  
  if (Serial.available()) {  
    /* llegeix el byte mes recent */  
    byteLlegit = Serial.read();  
    /* Retorna el valor llegit. Es a dir, fa eco.*/  
    Serial.write(byteLlegit);  
  }  
}
```

<http://www.binefa.cat/php/arduino/serialWrRd01.zip>

```
/*
serialWrRd02.ino
http://www.binefa.cat/php/arduino/serialWrRd02.zip
*/
#include <ecat.h>
Ecat ecat;

void setup() {
    // Els 4 bits (nibble) baixos de P2 son d'entrada
    ecat.setupNibbleMode(NIBBLE_L_P2, INPUT);
    // Els 4 bits (nibble) alts de P2 son de sortida
    ecat.setupNibbleMode(NIBBLE_H_P2, OUTPUT);
    // Velocitat de la comunicacio en bauds
    Serial.begin(9600);
}

void loop() {
    byte byteLlegit;
    /* Si hi ha quelcom a llegir: */
    if (Serial.available()) {
        /* llegeix el byte mes recent */
        byteLlegit = Serial.read();
        /* Retorna el valor llegit. Es a dir, fa eco.*/
        Serial.write(byteLlegit);
        if(byteLlegit == '1'){
            digitalWrite(ecat.nPinP2B7, HIGH);
        }
        if(byteLlegit == '2'){
            digitalWrite(ecat.nPinP2B7, LOW);
        }
    }
}
```

<http://www.binefa.cat/php/arduino/serialWrRd03.zip>

```
/*
serialWrRd03.ino
http://www.binefa.cat/php/arduino/serialWrRd03.zip
*/
#include <ecat.h>
Ecat ecat;

void setup() {
  ecat.setupNibbleMode(NIBBLE_L_P2, INPUT);
  ecat.setupNibbleMode(NIBBLE_H_P2, OUTPUT);
  Serial.begin(9600);
}

void vIterpretaByteLlegit(byte by){
  if(by == '1')
    digitalWrite(ecat.nPinP2B7, HIGH);
  if(by == '2')
    digitalWrite(ecat.nPinP2B7, LOW);
}

bool bCanviP2B0(){
  static bool bNivellP2B0 = digitalRead(ecat.nPinP2B0);

  if(bNivellP2B0 != digitalRead(ecat.nPinP2B0)){
    bNivellP2B0 = digitalRead(ecat.nPinP2B0);
    return true;
  }
  return false;
}

void vTrametValor(byte byQuinPin){
  if(digitalRead(ecat.nPinP2B0))
    Serial.write('H');
  else
    Serial.write('L');
}

void loop() {
  byte byteLlegit;

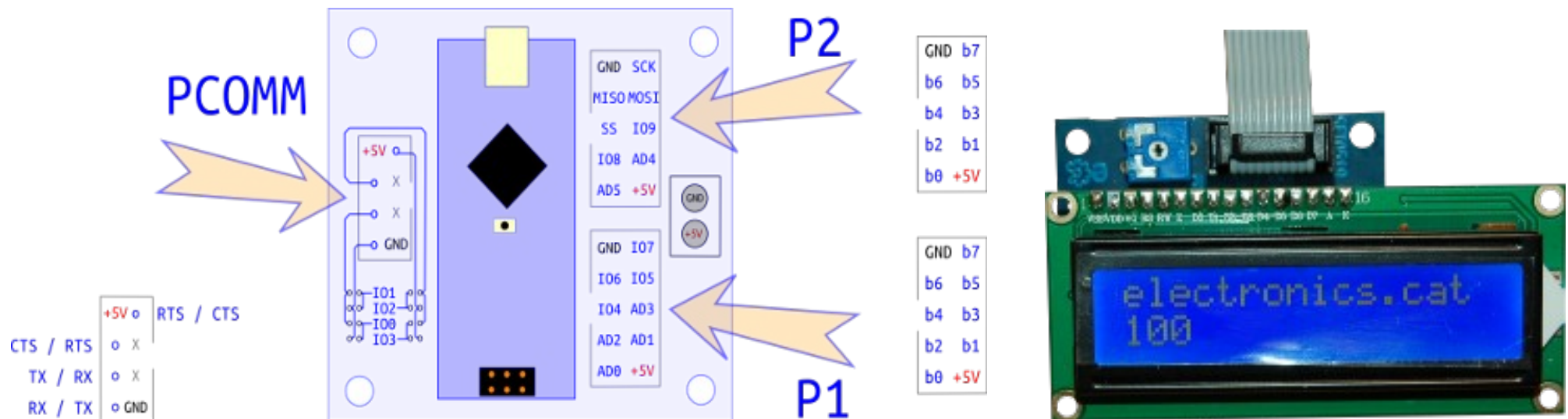
  if (Serial.available()) {
    byteLlegit = Serial.read();
    Serial.write(byteLlegit);
    vIterpretaByteLlegit(byteLlegit);
  }
  if( bCanviP2B0() ){
    delay(100); // Per a evitar rebots
    vTrametValor(ecat.nPinP2B0);
  }
}
```



# Maquinari lliure amb Ubuntu

## LCD

Afegiu el mòdul de l'LCD a P1 de la nano-eCat :



<http://www.binefa.cat/php/arduino/serialWrRd04LcdP1.zip>



# Maquinari lliure amb Ubuntu

## Codi d'Arduino

```

/*
serialWrRd04LcdP1.ino
http://www.binefa.cat/php/arduino/serialWrRd04LcdP1.zip
*/
#include <ecat.h>
#include <LiquidCrystal.h>

Ecat ecat;
lcdP1_init();

void setup() {
  ecat.setupNibbleMode(NIBBLE_L_P2, INPUT);
  ecat.setupNibbleMode(NIBBLE_H_P2, OUTPUT);
  lcdP1.begin(16, 2);
  lcdP1.print("ecat.h - v"); lcdP1.print(ecat.fVersion());
  lcdP1.setCursor(0, 1);
  lcdP1.print("electronics.cat |");
  Serial.begin(9600);
}

void vInterpretaByteLlegit(byte by){
  lcdP1.setCursor(0, 1);
  lcdP1.print("P2B7 = ");

  if(by == '1'){
    digitalWrite(ecat.nPinP2B7, HIGH);
    lcdP1.print("ON");
  }
  if(by == '2'){
    digitalWrite(ecat.nPinP2B7, LOW);
    lcdP1.print("OFF");
  }
}

```

```

bool bCanviP2B0(){
  static bool bNivellP2B0 = digitalRead(ecat.nPinP2B0);

  if(bNivellP2B0 != digitalRead(ecat.nPinP2B0)){
    bNivellP2B0 = digitalRead(ecat.nPinP2B0);
    return true;
  }
  return false;
}

void vTrametValor(byte byQuinPin){
  lcdP1.setCursor(0, 0);
  lcdP1.print("P2B0 = ");
  if(digitalRead(ecat.nPinP2B0)){
    Serial.write('H');
    lcdP1.print("OFF");
  }else{
    Serial.write('L');
    lcdP1.print("ON");
  }
}

void loop() {
  byte byteLlegit;

  if (Serial.available()) {
    byteLlegit = Serial.read();
    Serial.write(byteLlegit);
    vInterpretaByteLlegit(byteLlegit);
  }
  if( bCanviP2B0() ){
    delay(100); // Per a evitar rebots
    vTrametValor(ecat.nPinP2B0);
  }
}

```

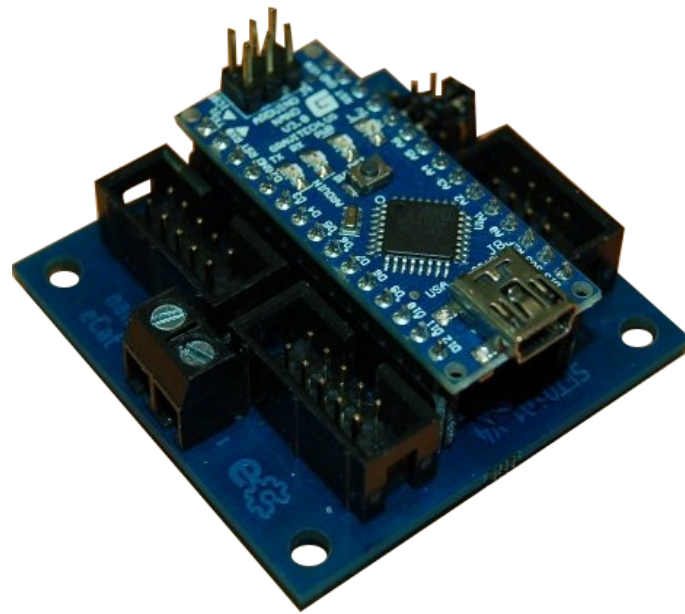
<http://www.binefa.cat/php/arduino/serialWrRd04LcdP1.zip>

# Maquinari lliure amb Ubuntu

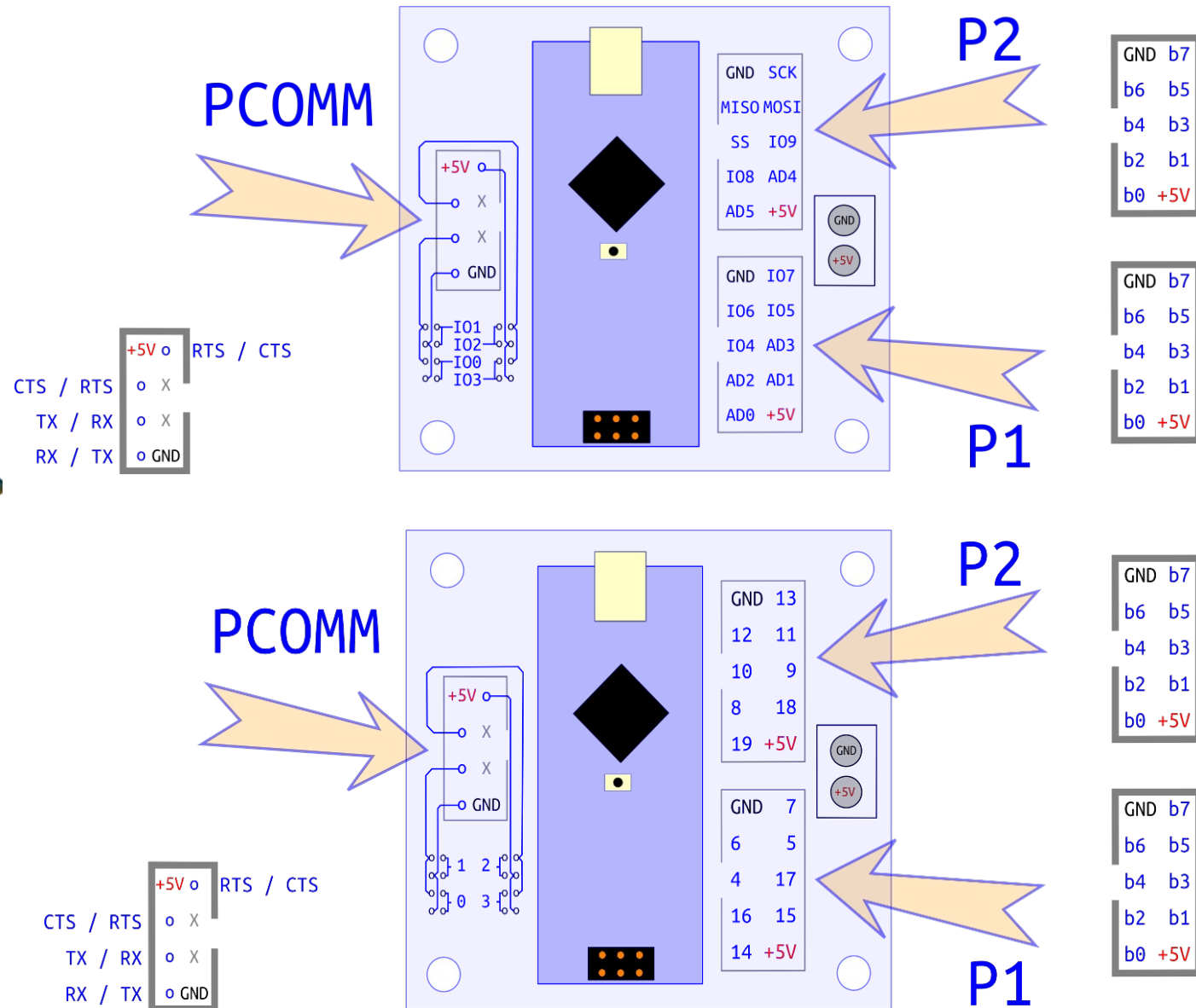
## Plaques amb sistema de bus lliure eCat

### nano-eCat

Placa 0101 v4



Placa convertidora de  
l'Arduino Nano al  
sistema de bus lliure  
eCat System



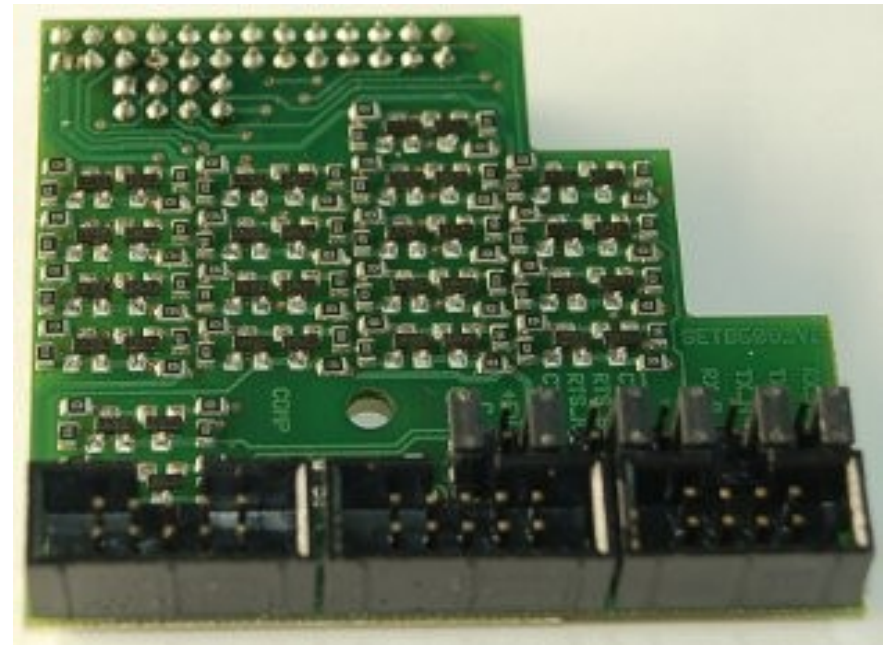




# Maquinari lliure amb Ubuntu

3Bpi - Pi de les Tres Branques

Placa traductora de nivells per a Raspberry Pi  
Placa set05\_08\_v2

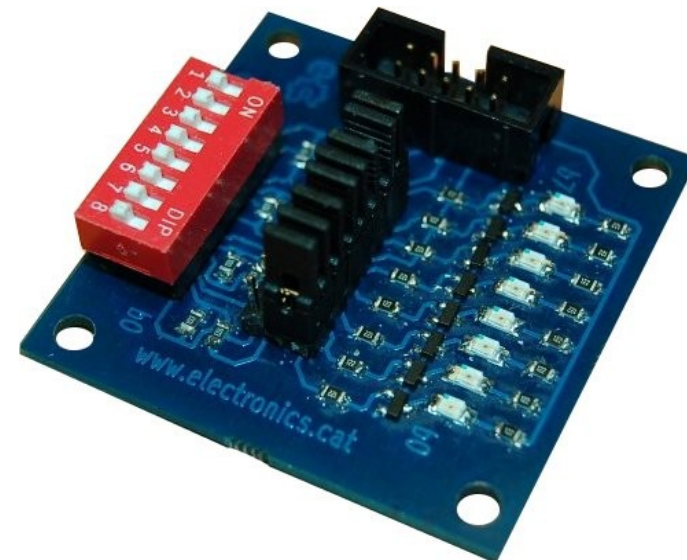
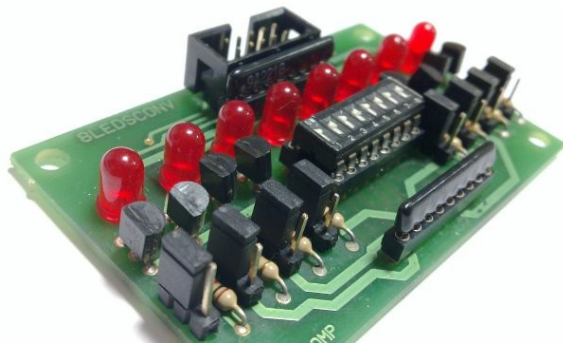
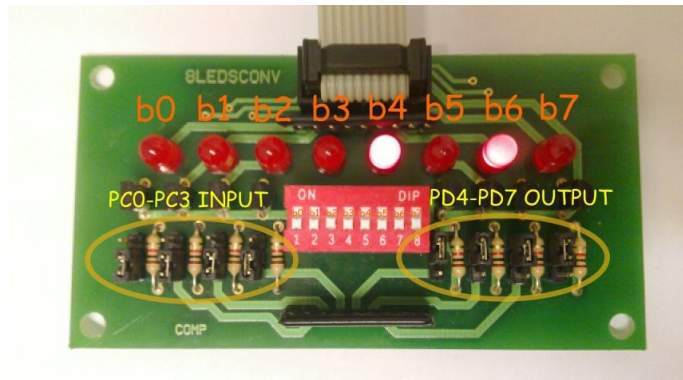
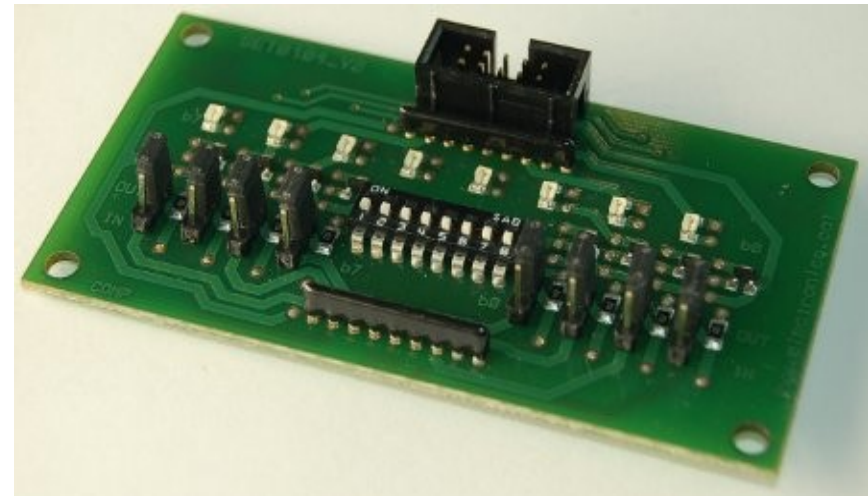
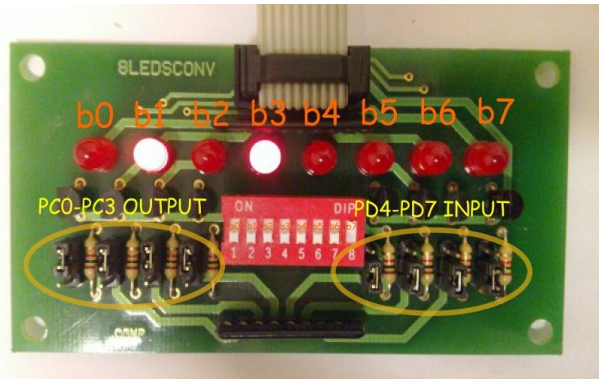






# Maquinari lliure amb Ubuntu

## Leds i microrruptors - Placa set01\_04



<http://electronics.cat/php/common/index.php?lang=ca&page=104>

# e⚙ Maquinari lliure amb Ubuntu

## Concepte de relé

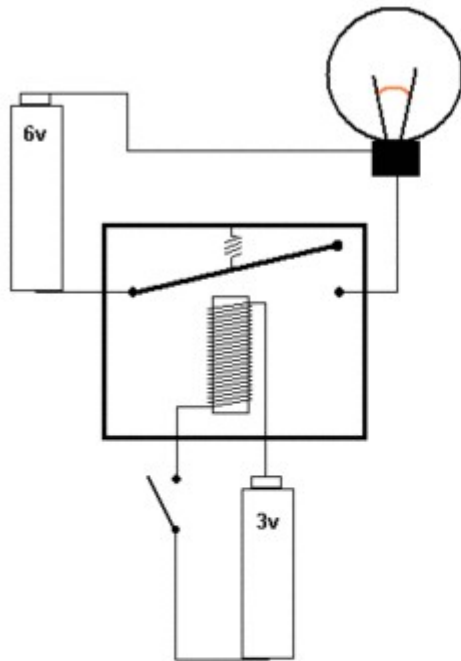


Figure 1: Relay off

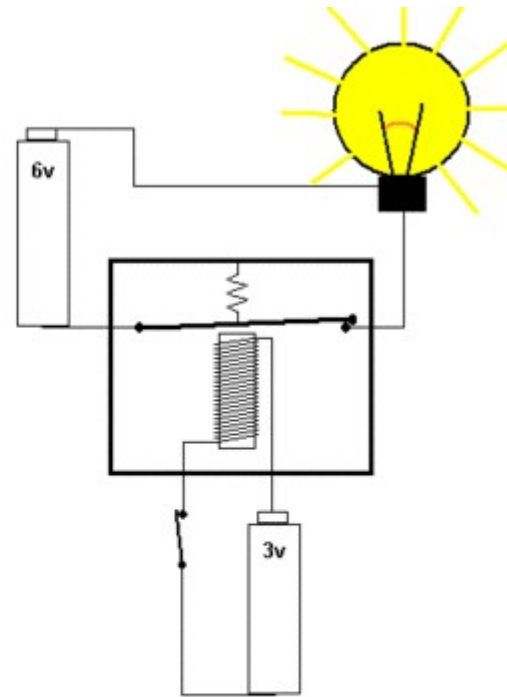
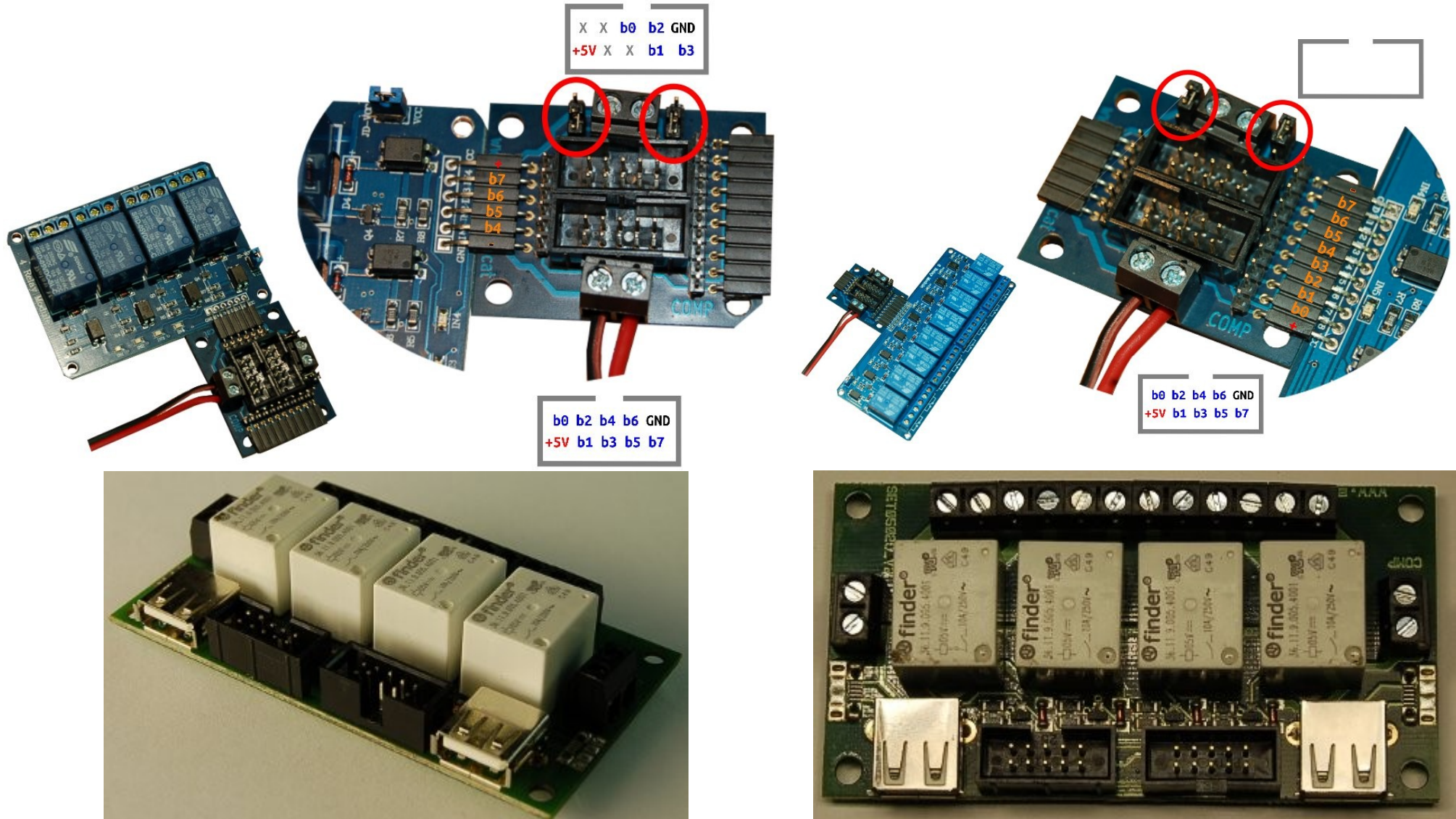


Figure 2: Relay on

# e⚙ Maquinari lliure amb Ubuntu

## 4 Relés - Placa set05\_02



<http://electronics.cat/php/common/index.php?lang=ca&page=502>





# Maquinari lliure amb Ubuntu

## Connexió a un mòdul Bluetooth



```
sudo aptitude install bluetooth bluez-utils blueman
cat /etc/bluetooth/rfcomm.conf
```

```
rfcomm1 {
    bind yes;
    device 98:D3:31:40:1F:2B;
    channel 1;
    comment "Connection to Bluetooth serial module";
}
```

<http://www.uugear.com/portfolio/bluetooth-communication-between-raspberry-pi-and-arduino/>

[http://www.thinkwiki.org/wiki/How\\_to\\_setup\\_Bluetooth](http://www.thinkwiki.org/wiki/How_to_setup_Bluetooth)

# Torn de preguntes ...



... i sessió pràctica.

# COMIAT

Presentació descarregable a :

[https://dl.dropboxusercontent.com/u/65254823/oshw20150509/20150509\\_00.pdf](https://dl.dropboxusercontent.com/u/65254823/oshw20150509/20150509_00.pdf)

<http://binefa.cat/blog/>

Correu electrònic de contacte : [jordibinefa@electronics.cat](mailto:jordibinefa@electronics.cat)

Twitter



<https://twitter.com/electronicscat>

<https://twitter.com/JordiBinefa>

LinkedIn

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

Plaques aviat disponibles a :

<http://www.electronics.cat>

**Moltes gràcies per la vostra assistència**