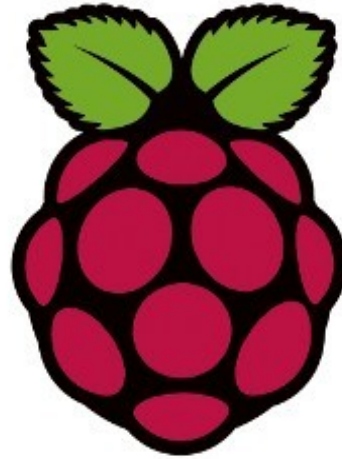


Comunicació entre Raspberry Pi i Arduino emprant plaques d'electronics.cat



Jordi Binefa i Martínez

Responsable d'R+D+i a

Professor de cicles formatius a



JESUÏTES El Clot
Escola del Clot

11 / abril / 2015



Comunicació entre Raspberry Pi i Arduino

Configuració de la Raspberry Pi

```
pi@raspberrypi ~ $ sudo raspi-config
```

Raspberry Pi Software Configuration Tool (raspi-config)

- | | |
|----------------------------------|---|
| 1 Expand Filesystem | Ensures that all of the SD card storage is available to the OS |
| 2 Change User Password | Change password for the default user (pi) |
| 3 Enable Boot to Desktop/Scratch | Choose whether to boot into a desktop environment, Scratch, or the command-line |
| 4 Internationalisation Options | Set up language and regional settings to match your location |
| 5 Enable Camera | Enable this Pi to work with the Raspberry Pi Camera |
| 6 Add to Rastrack | Add this Pi to the online Raspberry Pi Map (Rastrack) |
| 7 Overclock | Configure overclocking for your Pi |
| 8 Advanced Options | Configure advanced settings |
| 9 About raspi-config | Information about this configuration tool |

<Select>

<Finish>

Chose boot option

Console Text console, requiring login (default)
 Desktop Log in as user 'pi' at the graphical desktop
 Scratch Start the Scratch programming environment upon boot

<Ok>

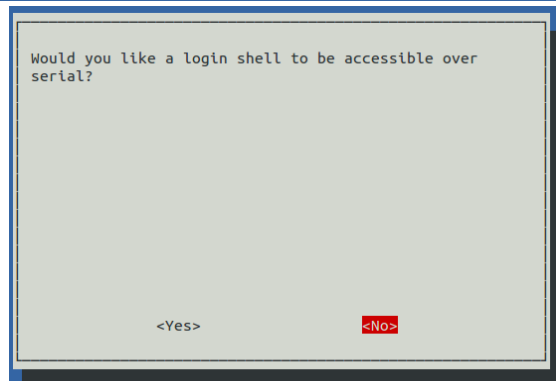
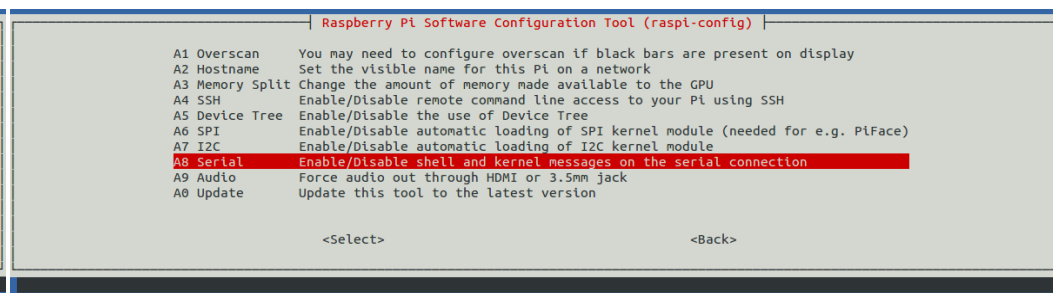
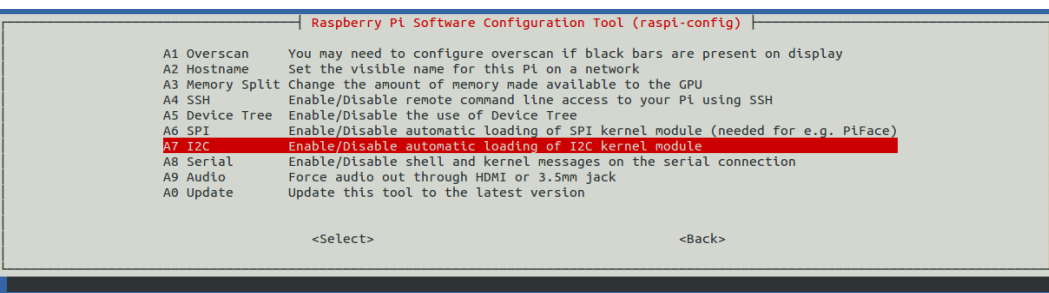
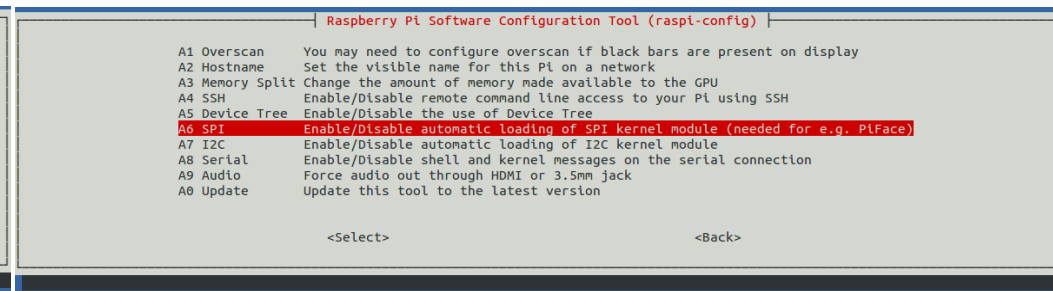
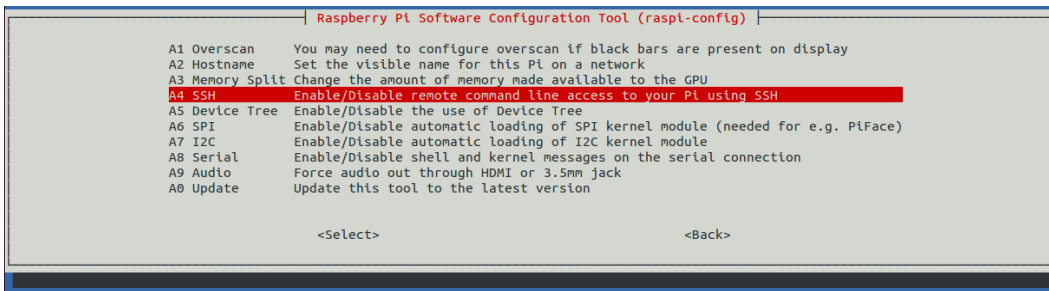
<Cancel>



Comunicació entre Raspberry Pi i Arduino

Configuració de la Raspberry Pi

```
pi@raspberrypi ~ $ sudo raspi-config
```



Configuració del port sèrie per a versions més antigues de Raspbian :

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



Comunicació entre Raspberry Pi i Arduino

Accés remot mitjançant SSH

```
jordi@eCat: ~  
jordi@eCat:~$ ssh pi@192.168.1.37  
pi@192.168.1.37's password: 
```

```
jordi@eCat:~$ ssh pi@192.168.1.37  
pi@192.168.1.37's password:  
Linux raspberrypi 3.18.7+ #755 PREEMPT Thu Feb 12 17:14:31 GMT 2015 armv6l  
  
The programs included with the Debian GNU/Linux system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.  
Last login: Wed Apr  8 13:03:17 2015 from 192.168.1.5  
pi@raspberrypi ~ $
```

Per accedir amb opció d'executar remotament programes amb interfície gràfica (GUI) :

```
ssh -X pi@192.168.1.37
```



Comunicació entre Raspberry Pi i Arduino

Configuració de l'I2C

```
pi@raspberrypi ~ $ sudo nano /etc/modules
```

```
pi@raspberrypi ~ $ sudo nano /etc/modules
pi@raspberrypi ~ $ cat /etc/modprobe.d/raspi-blacklist.conf
pi@raspberrypi ~ $
```

```
pi@raspberrypi ~ $ cat /etc/modules
# /etc/modules: kernel modules to load at boot time.
#
# This file contains the names of kernel modules that should be
# at boot time, one per line. Lines beginning with "#" are ignored.
# Parameters can be specified after the module name.

snd-bcm2835
i2c-bcm2708
i2c-dev
```

```
pi@raspberrypi ~ $ uname -a
Linux raspberrypi 3.18.7+ #755 PREEMPT Thu Feb 12 17:14:31 GMT
pi@raspberrypi ~ $ tail /boot/config.txt
#config_hdmi_boost=4

# uncomment for composite PAL
#sdtv_mode=2

#uncomment to overclock the arm. 700 MHz is the default.
#arm_freq=800

dtparam=spi=on
dtparam=i2c_arm=on
pi@raspberrypi ~ $
```

```
pi@raspberrypi ~ $ tail /boot/config.txt
#config_hdmi_boost=4
```

```
# uncomment for composite PAL
#sdtv_mode=2
```

```
#uncomment to overclock the arm. 700 MHz is the default.
#arm_freq=800
```

```
dtparam=spi=on
```

```
dtparam=i2c_arm=on
```

```
pi@raspberrypi ~ $ sudo nano /boot/config.txt
```

```
pi@raspberrypi ~ $ tail /boot/config.txt
```

```
# uncomment for composite PAL
#sdtv_mode=2
```

```
#uncomment to overclock the arm. 700 MHz is the default.
#arm_freq=800
```

```
dtparam=spi=on
```

```
dtparam=i2c1=on
```

```
dtparam=i2c_arm=on
```

```
pi@raspberrypi ~ $
```



Comunicació entre Raspberry Pi i Arduino

Configuració de l'I2C

```
pi@raspberrypi ~ $ sudo aptitude install i2c-tools python-smbus
```

```
pi@raspberrypi ~ $ sudo i2cdetect -y 1
   0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
10:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
20:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
30:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
40:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
50:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
60:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
70:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
```



Comunicació entre Raspberry Pi i Arduino

Configuració de l'accés sèrie

Instal·lació del minicom, programa de comunicacions amb el port sèrie :

```
pi@raspberrypi ~ $ sudo aptitude install minicom
```

Assignació de permisos a l'usuari normal (més enllà del primari o root) :

```
pi@raspberrypi ~ $ sudo usermod -a -G dialout pi
```

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

Execució del minicom pel terminal :

```
minicom -b 9600 -o -D /dev/ttyAMA0
```

```
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'



Comunicació entre Raspberry Pi i Arduino

Instal·lació de més programes

Instal·leu biblioteques complementàries per a interactuar amb Python, l'IDE d'Arduino i les biblioteques complementàries d'electronics.cat :

```
pi@raspberrypi ~ $ sudo aptitude install python-dev python-rpi.gpio python-serial  
python-netifaces python-pip
```

```
pi@raspberrypi ~ $ sudo pip install pyfirmata
```

```
pi@raspberrypi ~/downloads $ sudo aptitude install arduino
```

```
pi@raspberrypi ~/downloads $ 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

```
pi@raspberrypi ~/downloads $ sudo mv ecat /usr/share/arduino/libraries
```

Descarregueu el paquet de l'Scratch For Arduino a la carpeta /home/pi/downloads i instal·leu-lo

```
pi@raspberrypi ~ $ cd downloads/  
pi@raspberrypi ~/downloads $ sudo dpkg -i S4A15_RP.deb
```




Comunicació entre Raspberry Pi i Arduino

Imatge de la Raspberry Pi descarregable

Us podeu descarregar la imatge ja preparada de la Raspberry prement sobre la icona o sobre l'enllaç :



https://mega.co.nz/#!1xFVEJKK!zear_LuHkAKryys4D_wsj8_jehya_7MZ1Ath84BF--4

L'arxiu descarregat és **raspbianEcat00.img.zip** i ocupa 1,27GB. Un cop descomprimit l'arxiu **raspbianEcat00.img** ocupa 7,9GB. L'ocupació real de la imatge és de menys de 5GB. La primera opció del configurador raspi-config és per a expandir la imatge a la mida màxima de la vostra targeta SD concreta.

```
pi@raspberrypi ~ $ sudo raspi-config
```



Comunicació entre Raspberry Pi i Arduino

Càrrega del protocol Firmata a l'Arduino

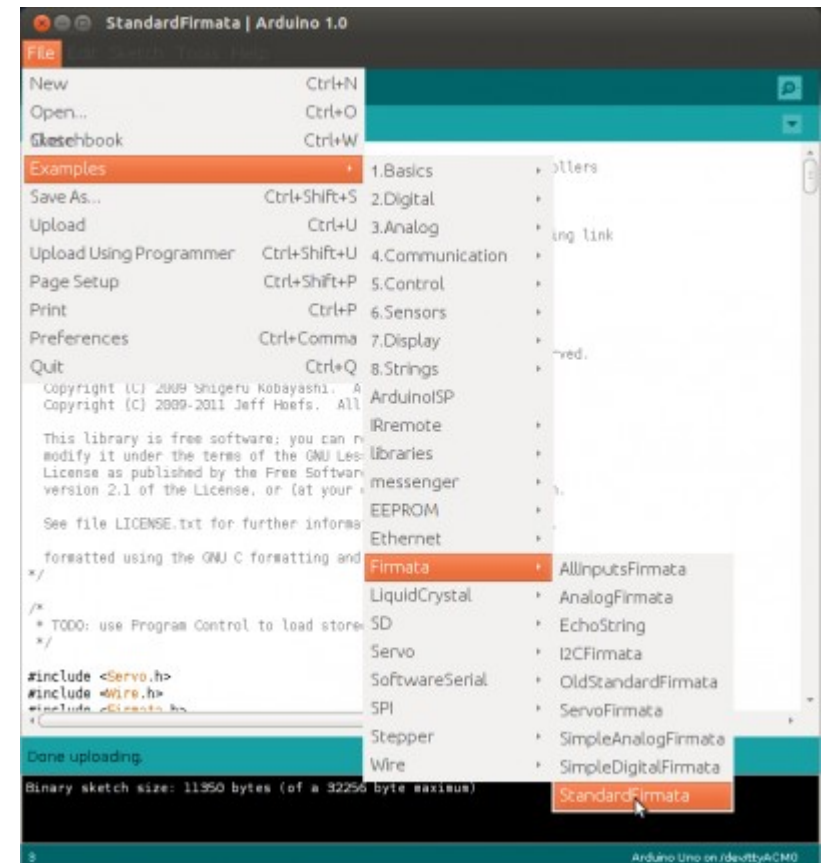
Connecteu la Raspberry Pi a la placa nano-eCat (la que conté l'Arduino Nano) mitjançant el cable USB-A a miniUSB.

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.





Comunicació entre Raspberry Pi i Arduino

Prova del protocol Firmata entre l'Arduino i la Raspberry Pi

A la Raspberry Pi aneu a Menú / Accessoris / LXTerminal i escriviu el següent :

```
pi@raspberrypi ~ $ cd Documents/codis/firmata/pyQt/
pi@raspberrypi ~/Documents/codis/firmata/pyQt $ ls -ls
total 24
8 -rwxr-xr-x 1 pi pi 4122 des  7 21:10 exFirmata03P1.py
4 -rwxr-xr-x 1 pi pi 3970 des  7 20:57 exFirmata03P2.py
4 -rw-r----- 1 pi pi 3255 abr  9 13:31 led_uSw.ui
4 -rw-r--r-- 1 pi pi 3954 des  5 16:25 ui_led_uSw.py
4 -rw-r--r-- 1 pi pi 3058 abr  9 13:33 ui_led_uSw.pyc
pi@raspberrypi ~/Documents/codis/firmata/pyQt $ ./exFirmata03P2.py
```

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

CheckBox

- ☒ Led Bit 7
- ☐ Led Bit 6
- ☒ Led Bit 5
- ☒ Led Bit 4
- ☐ uSw Bit 3
- ☐ uSw Bit 2
- ☐ uSw Bit 1
- ☐ uSw Bit 0

Surt



Comunicació entre Raspberry Pi i Arduino

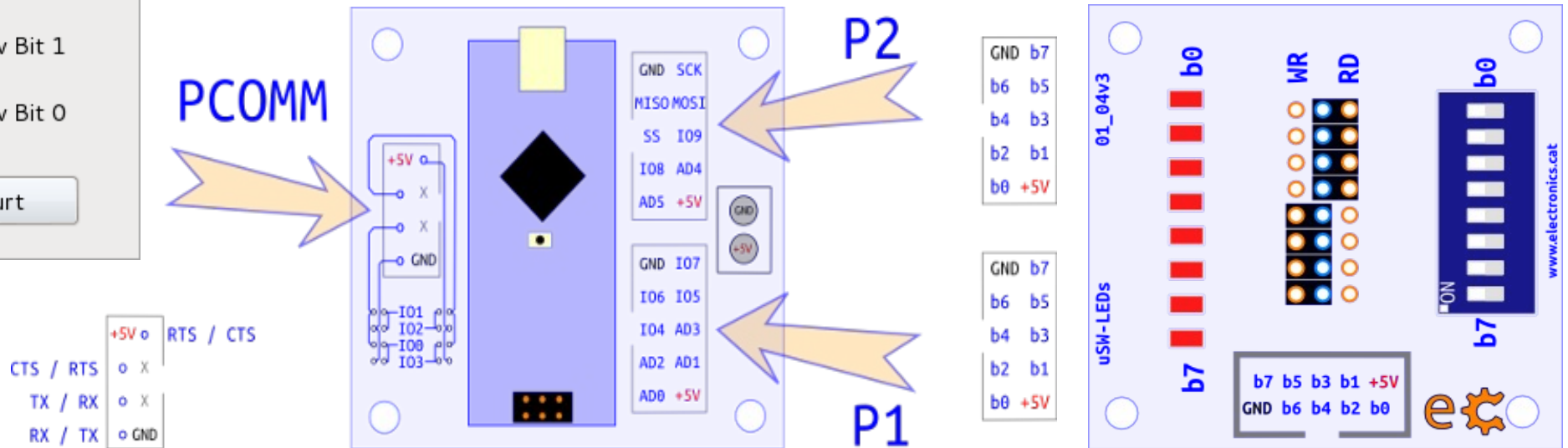
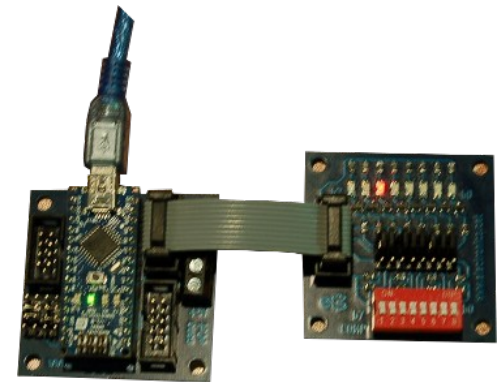
Prova del protocol Firmata entre l'Arduino i la Raspberry Pi

CheckBox

☒ Led Bit 7
☐ Led Bit 6
☒ Led Bit 5
☒ Led Bit 4
☐ uSw Bit 3
☐ uSw Bit 2
☐ uSw Bit 1
☐ uSw Bit 0

Surt

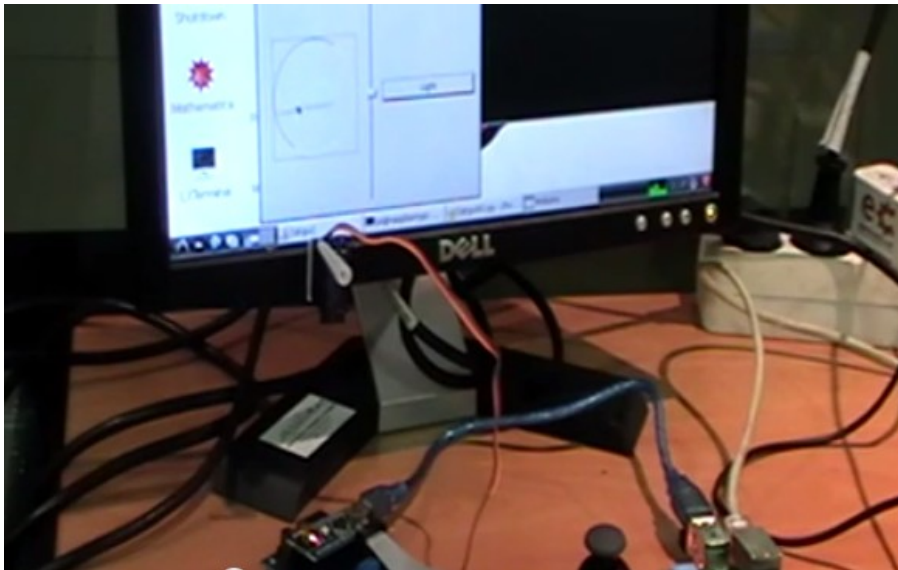
Connecteu la placa 0104
(la de leds i uSW) a P2 de
la nano-eCat





Comunicació entre Raspberry Pi i Arduino

Prova del protocol Firmata entre l'Arduino i la Raspberry Pi

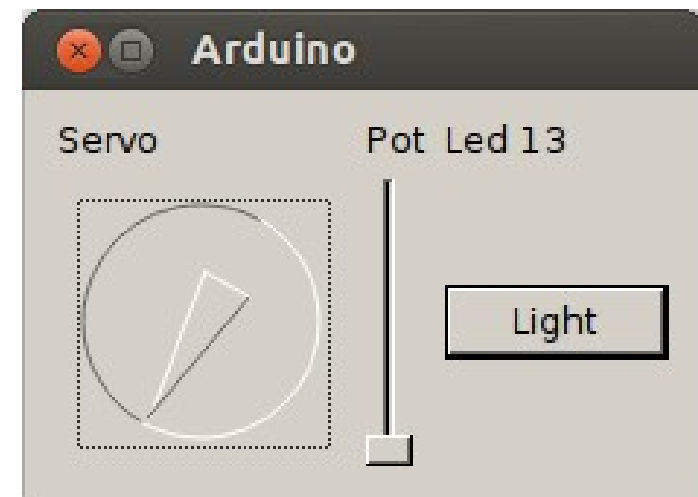
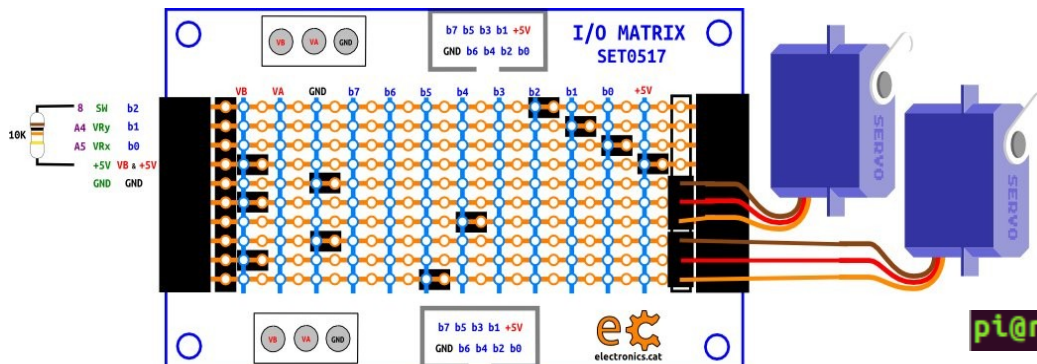


Vídeo :

<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>



Comunicació entre Raspberry Pi i Arduino

Protocol Firmata



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

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



Comunicació entre Raspberry Pi i Arduino

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.:

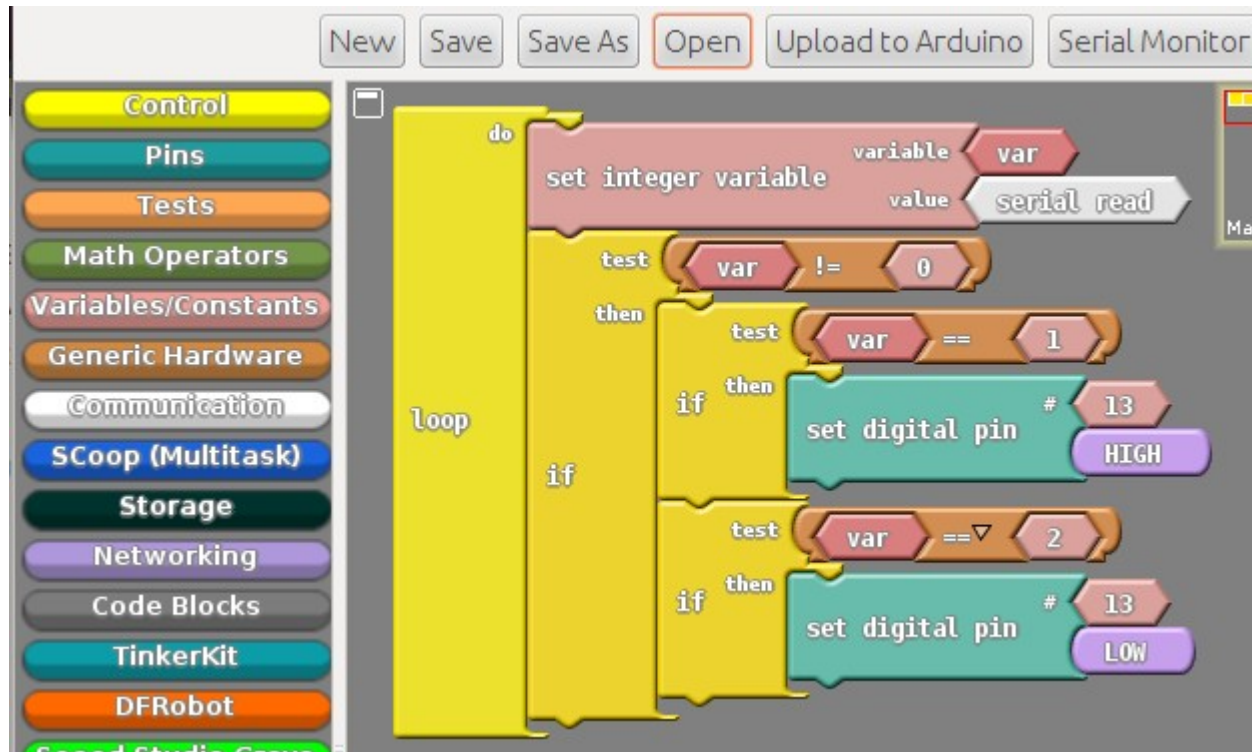
```
>>> analog_0 = board.get_pin('a:0:i')
>>> analog_0.read()
0.661440304938
>>> pin3 = board.get_pin('d:3:p')
>>> pin3.write(0.6)
```

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



Comunicació entre Raspberry Pi i Arduino

Ús d'ArduBlock



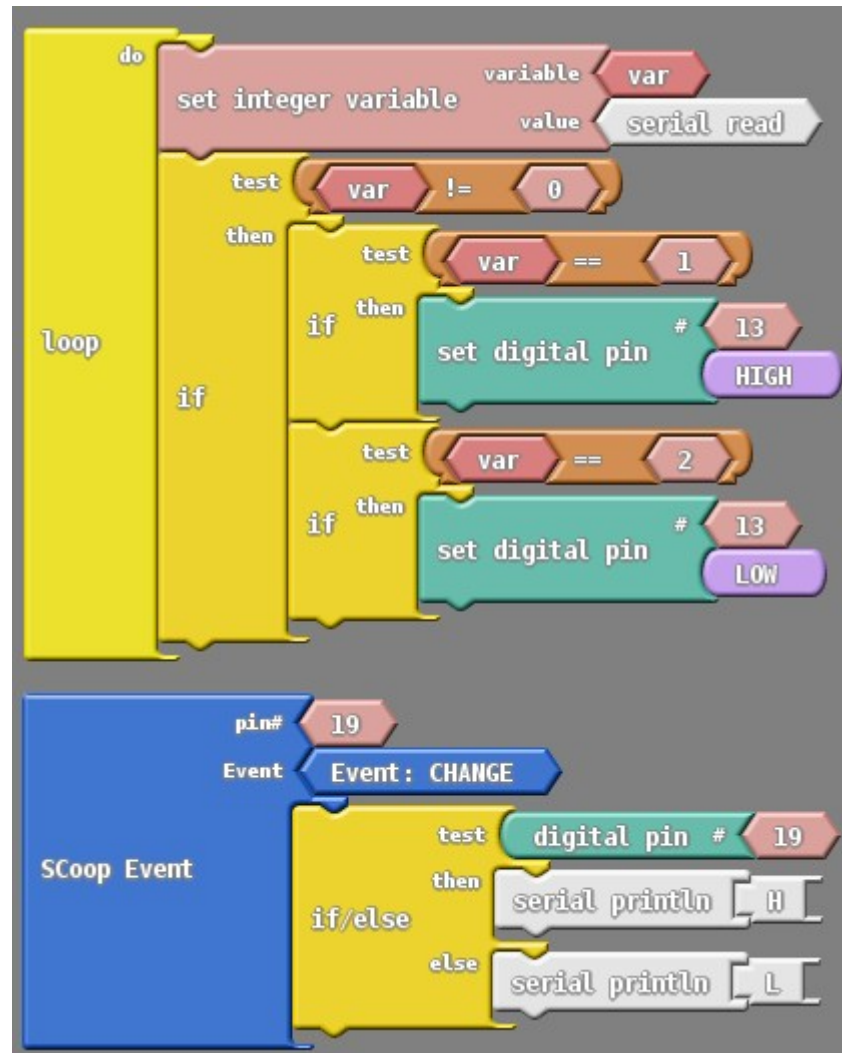
Projecte ArduBlock : <http://blog.ardublock.com/>

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.



Comunicació entre Raspberry Pi i Arduino

Ús d'ArduBlock



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



Comunicació entre Raspberry Pi i Arduino Ús d'ArduBlock

L'ArduBlock alenteix molt la velocitat (minuts) d'execució de l'Arduino IDE.

Per activar l'ArduBlock a la Raspberry Pi amb la imatge **raspbianEcat00.img** s'ha de copiar la carpeta **tools** de **/home/pi/sketchbook2** a **/home/pi/sketchbook**

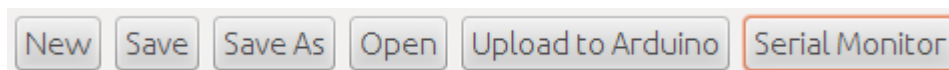
Per a tornar a treballar amb normalitat amb l'Arduino IDE haureu d'esborrar la carpeta **tools** de **/home/pi/sketchbook**

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 **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 interactuar amb l'Arduino



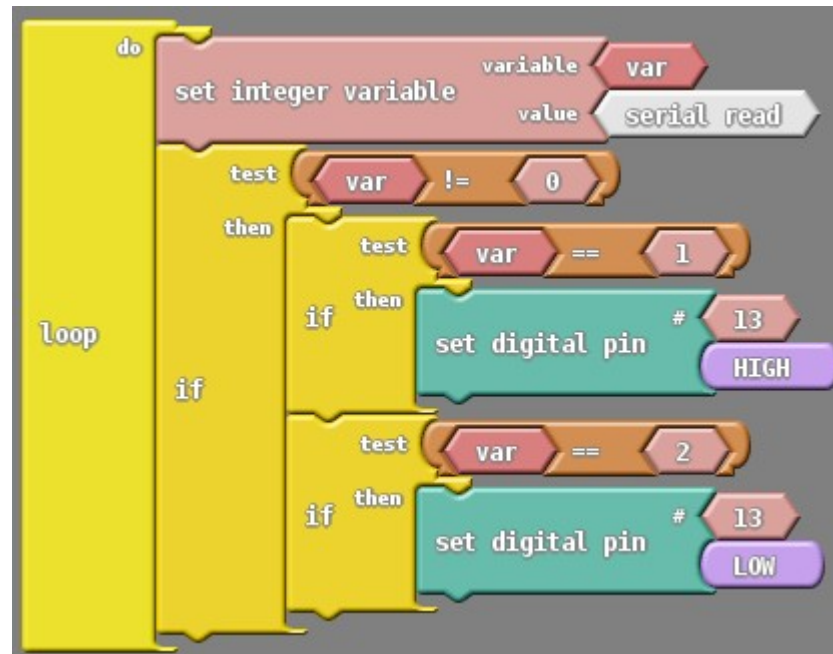


Comunicació entre Raspberry Pi i Arduino

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.





Comunicació entre Raspberry Pi i Arduino

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)



Un cop hem acabat les proves, tanquem l'ArduBlock i l'Arduino IDE i esborrem la carpeta **tools** del directori `/home/pi/sketchbook`



Comunicació entre Raspberry Pi i Arduino

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.

```
pi@raspberrypi ~ $ 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



Comunicació entre Raspberry Pi i Arduino

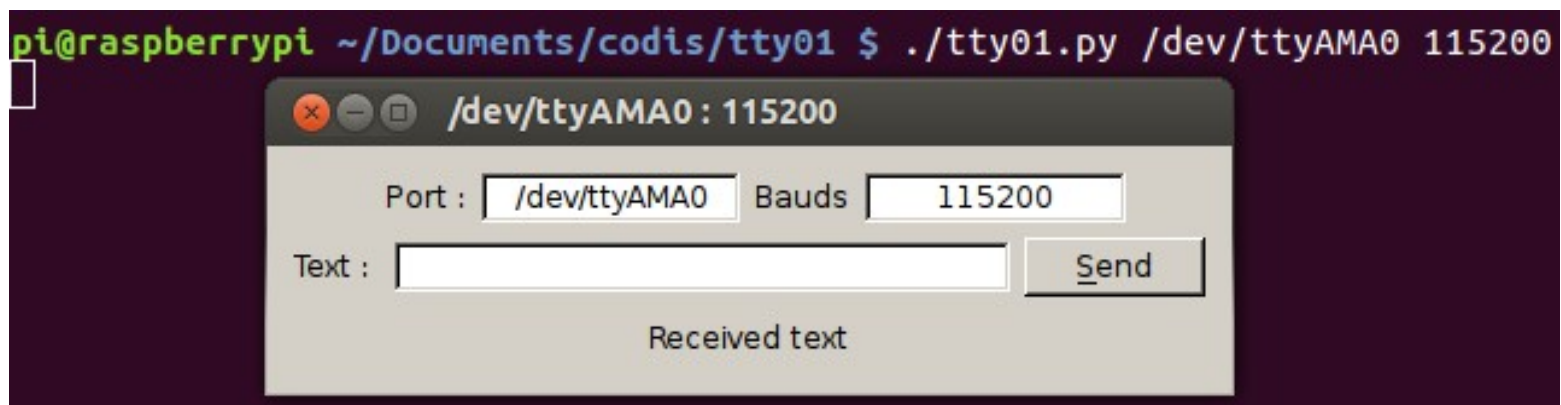
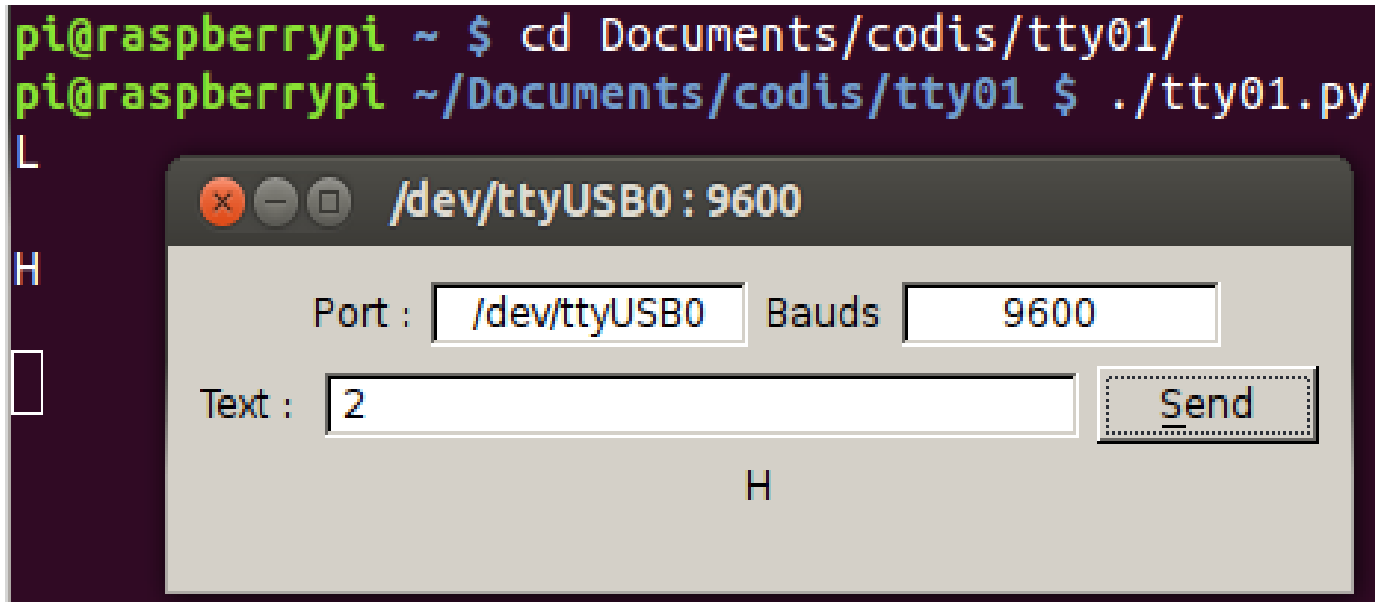
Ús del port sèrie des de Python

```
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
```



Comunicació entre Raspberry Pi i Arduino

Ús d'un programa amb interfície gràfica



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



Comunicació entre Raspberry Pi i Arduino

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');
}

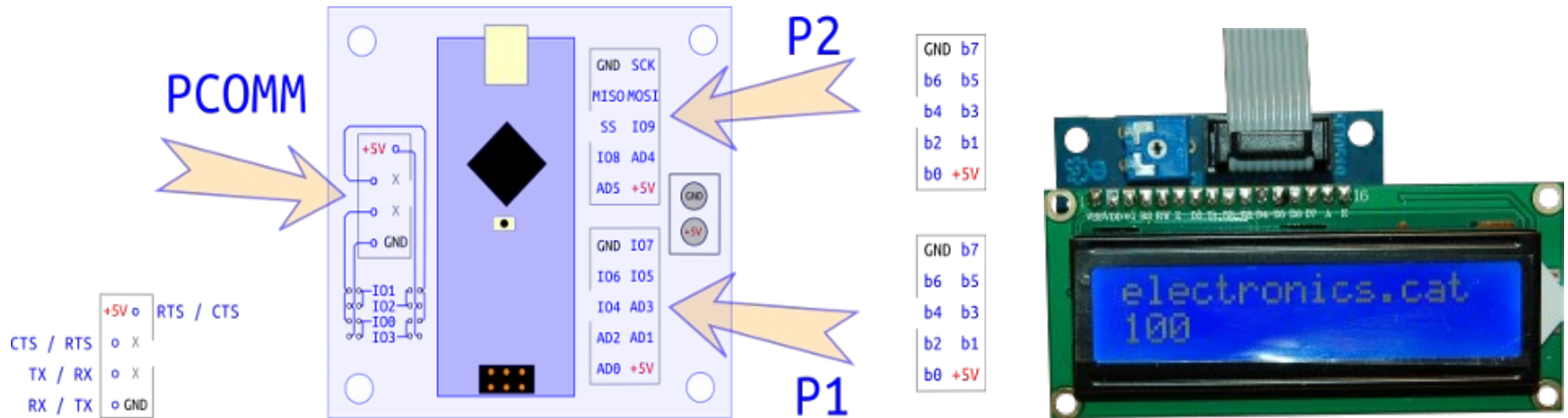
void loop() {
  byte byteLlegit;

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



Comunicació entre Raspberry Pi i Arduino LCD

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



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



Comunicació entre Raspberry Pi i Arduino

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 vIterpretaByteLlegit(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);
    vIterpretaByteLlegit(byteLlegit);
  }
  if( bCanviP2B0() ){
    delay(100); // Per a evitar rebots
    vTrametValor(ecat.nPinP2B0);
  }
}

```

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

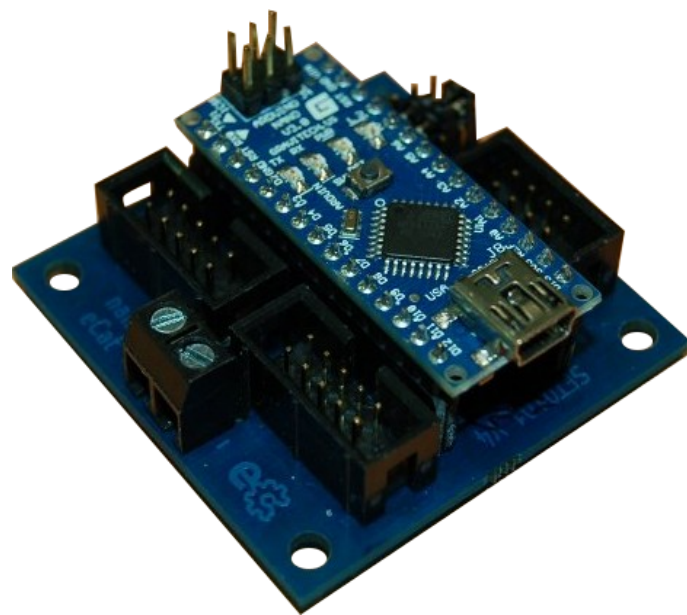


Maquinari lliure amb Arduino Nano

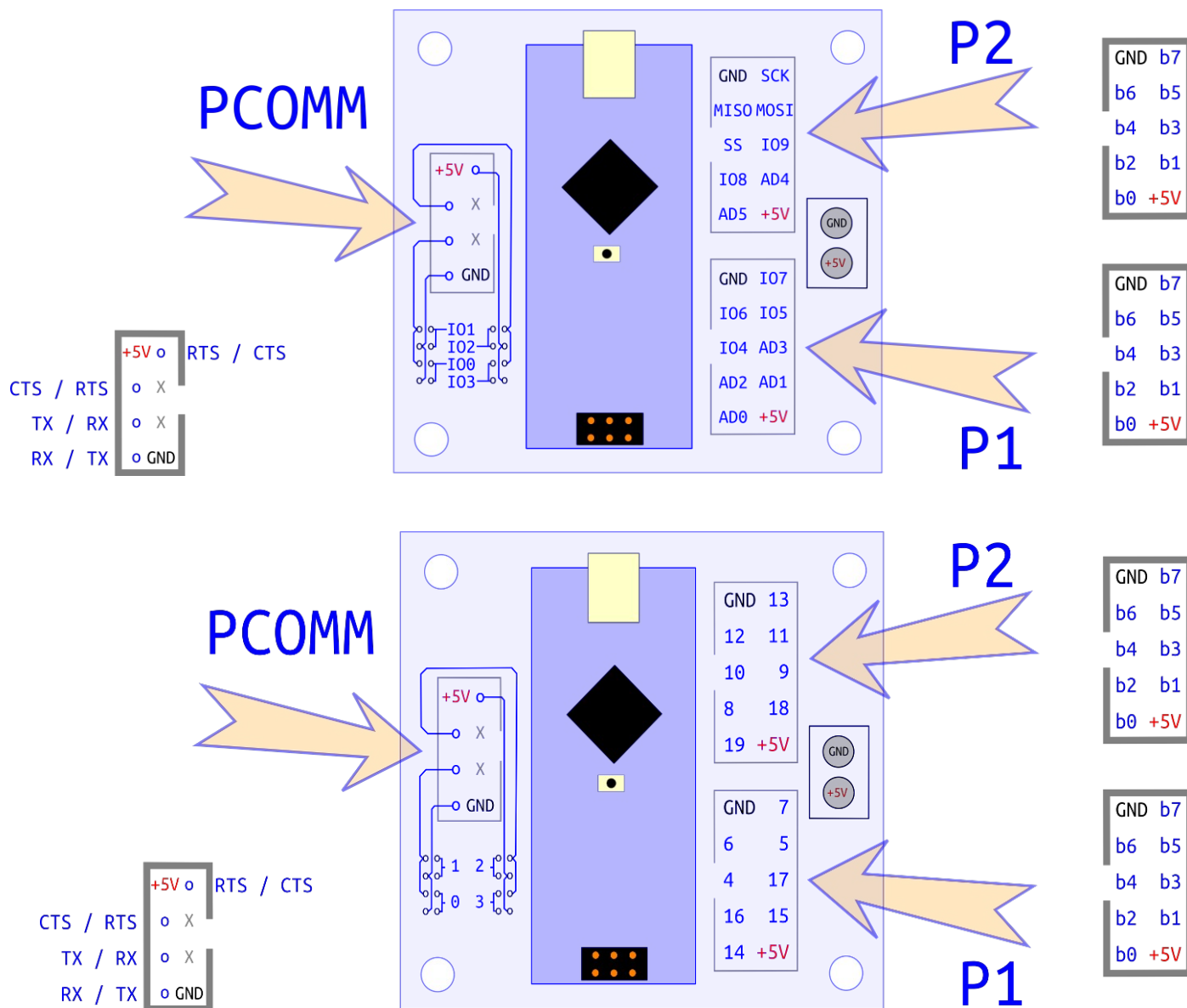
Plaques amb sistema de bus lliure eCat

nano-eCat

Placa 0101 v4



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

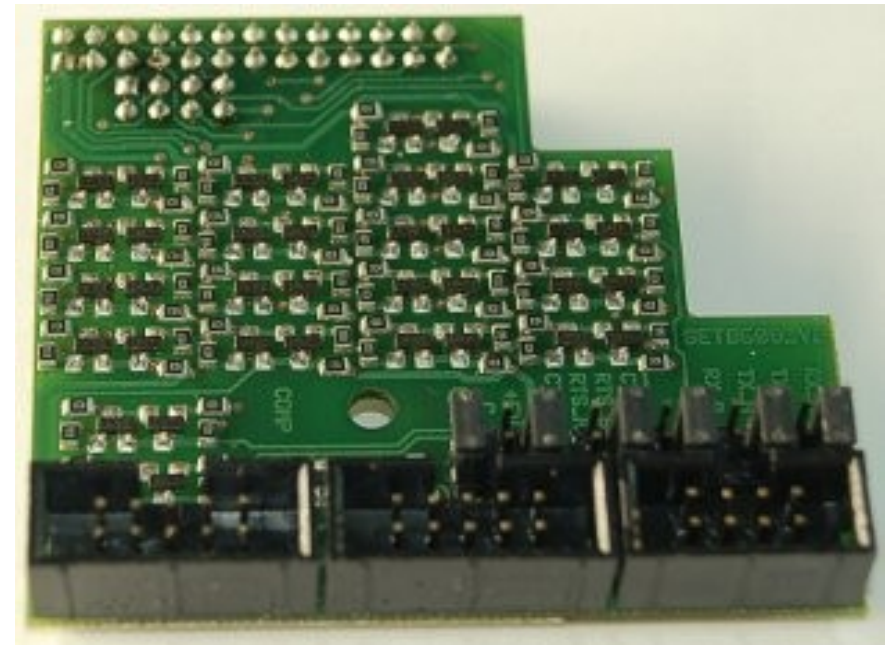




Maquinari

3Bpi - Pi de les Tres Branques

Placa traductora de nivells per a Raspberry Pi
Placa set05_08_v2

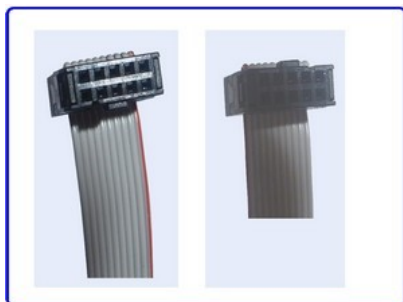


Maquinari

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)



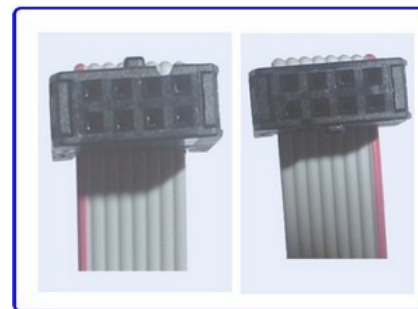
CTS / RTS

TX / RX

RX / TX

+5V	o	CTS / RTS
o	X	
o	X	
o	GND	

Cable pla
4x2 a 4x2
(0311)

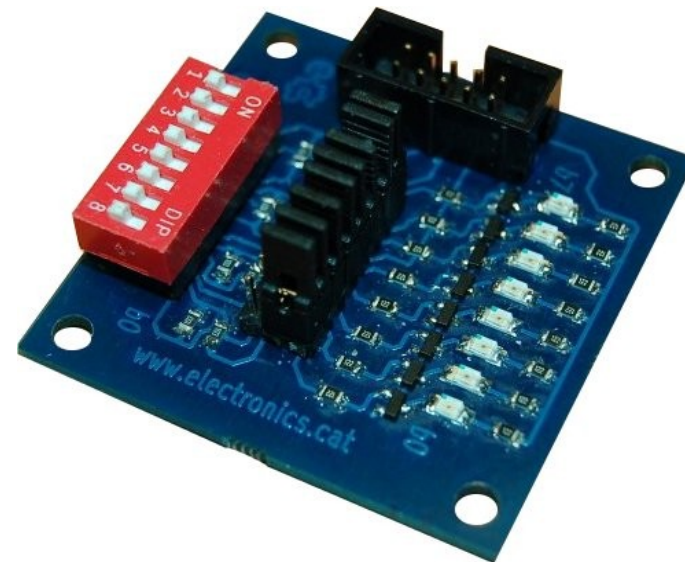
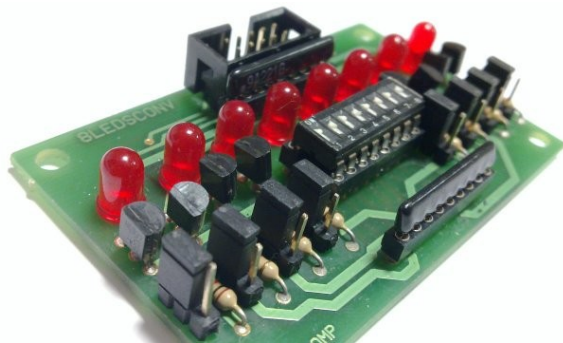
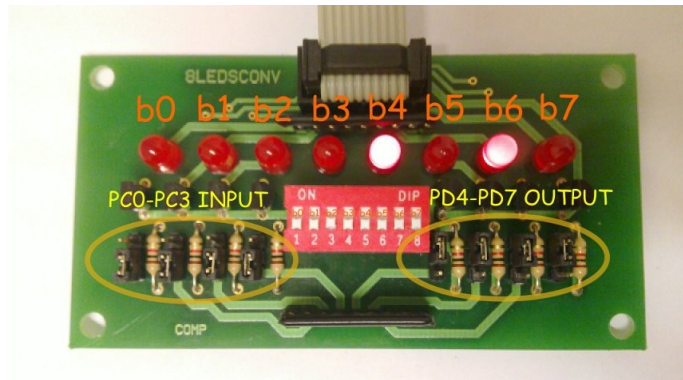
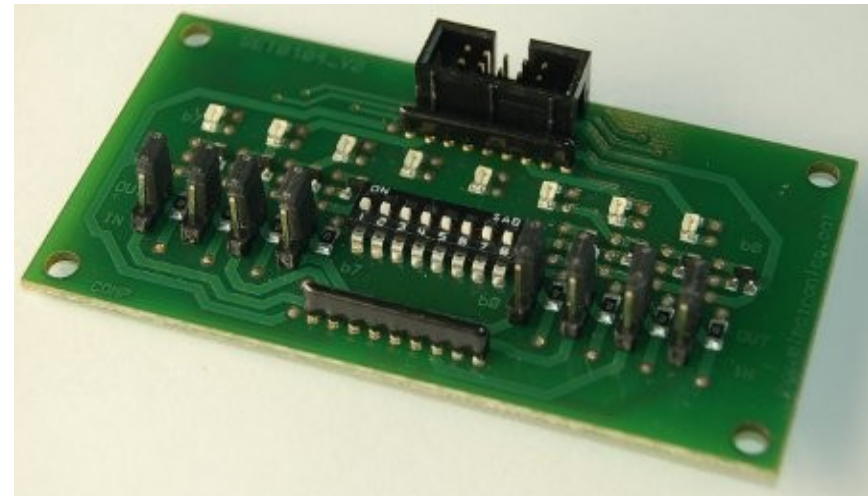
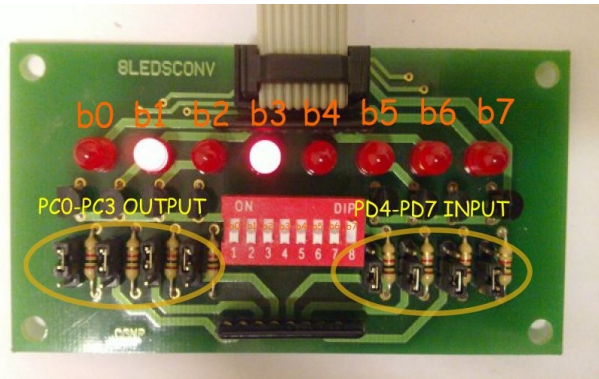


eCat System : Dues tipologies de BUS, dades (8 bits) i comunicacions



Maquinari

Leds i microrruptors - Placa set01_04



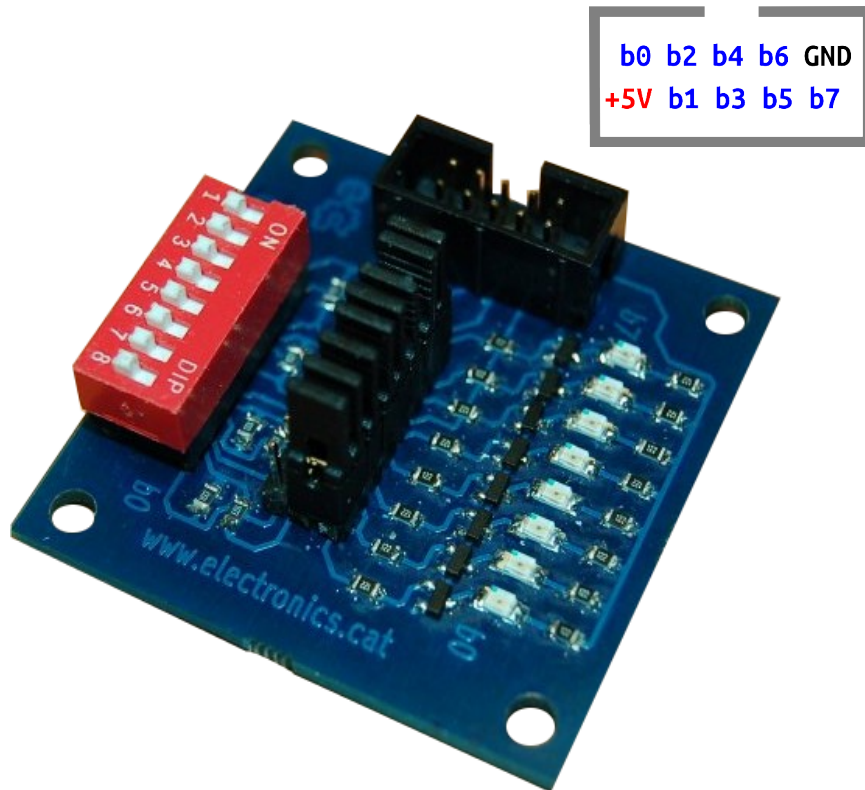
<http://electronics.cat/php/common/index.php?lang=ca&page=104>



Maquinari lliure amb Arduino Nano 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

Concepte de relé

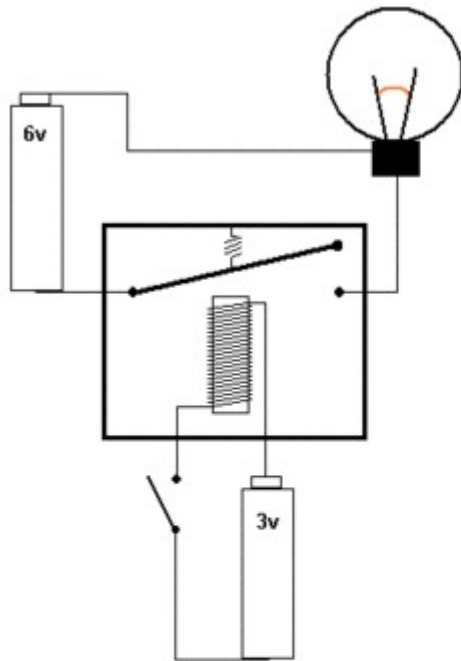


Figure 1: Relay off

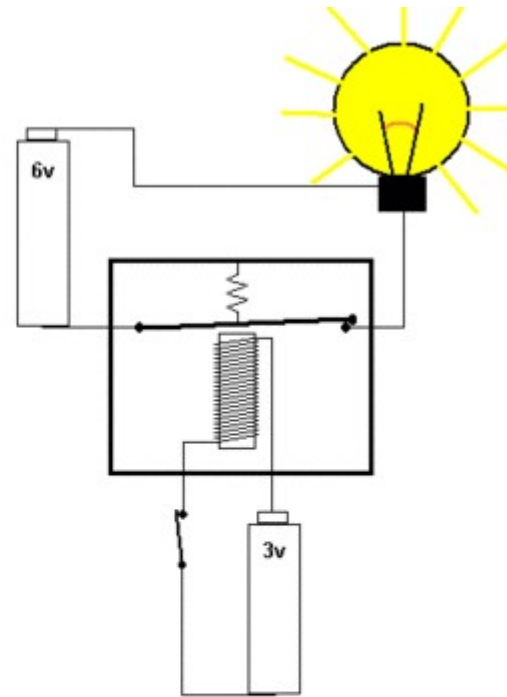
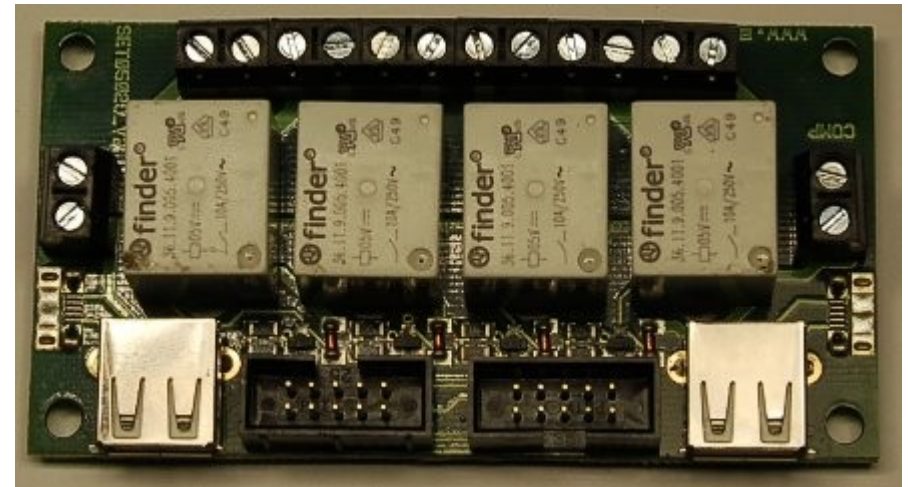
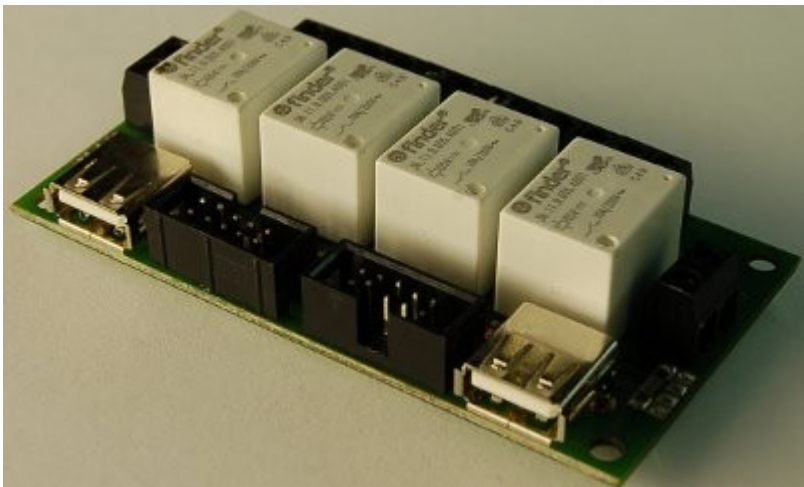
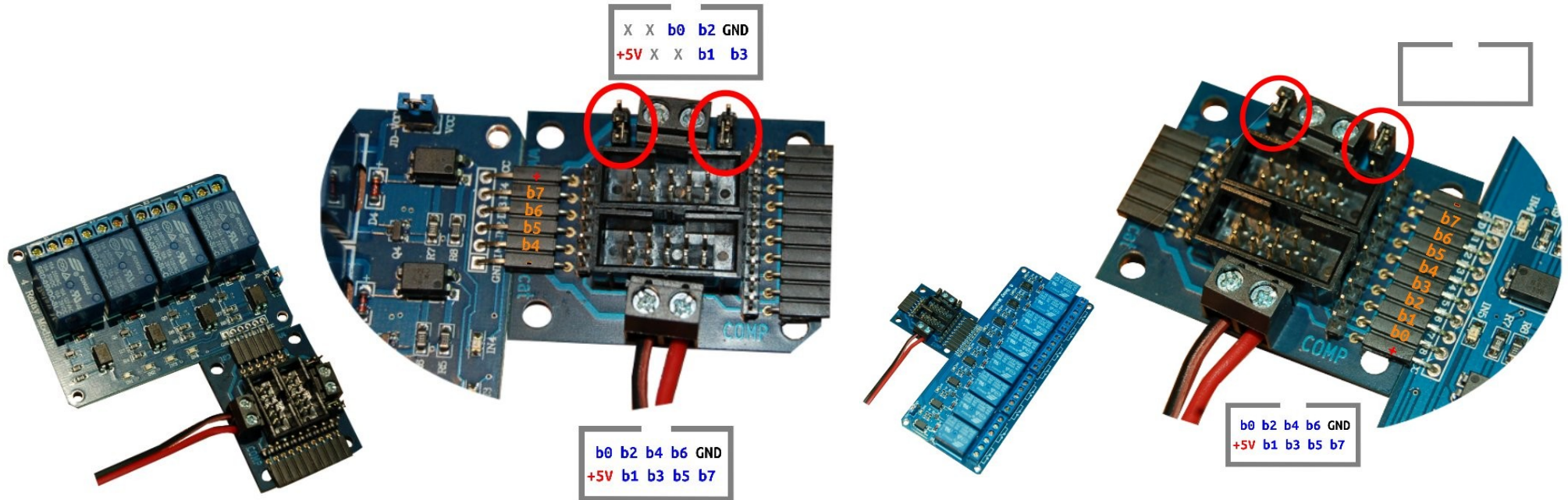


Figure 2: Relay on



Maquinari

4 Relés - Placa set05_02



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



Comunicacions sèrie

GtkTerm

The image shows a 'Configuration' dialog box for a serial port. It has a title bar with standard window controls and the text 'Configuration'. The dialog 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 currently unchecked. At the bottom right of the dialog 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) :

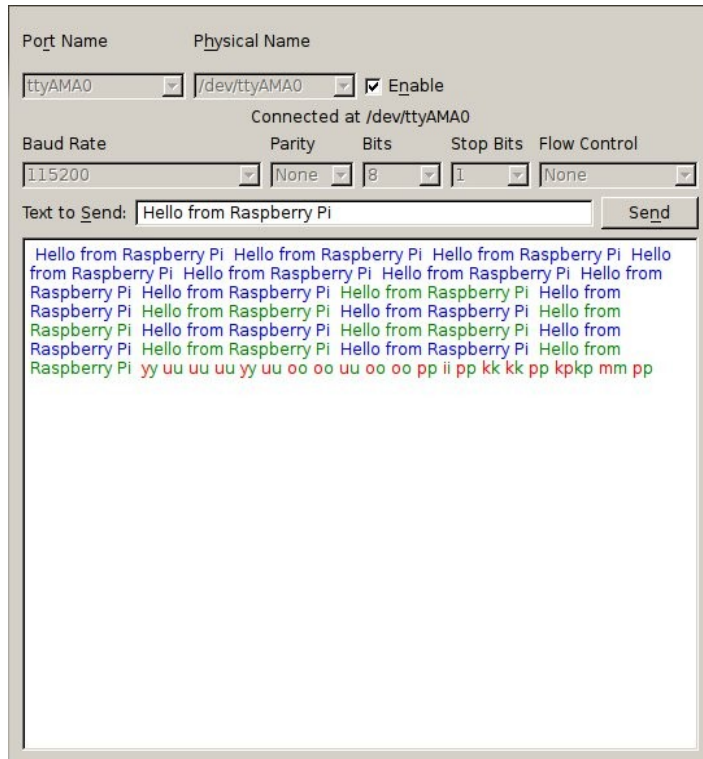
☐ Wait for this special character before passing to next line :

D'acord Cancel·la



Comunicacions sèrie

qtTerm



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

Torn de preguntes ...



... i sessió pràctica.

COMIAT

Presentació descarregable a :

https://dl.dropboxusercontent.com/u/65254823/oshw20151104/20150411_02.pdf

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

Correu electrònic de contacte : 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