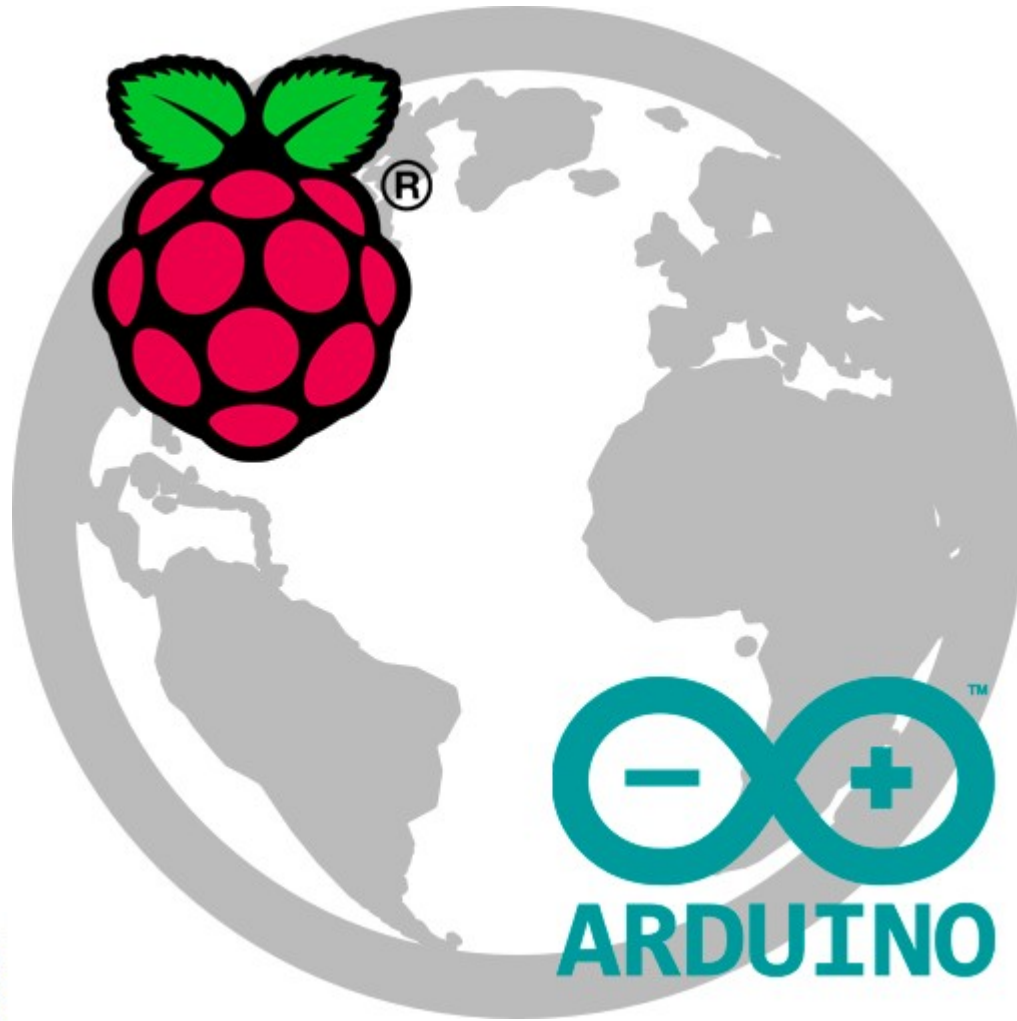


Arduino i Raspberry Pi



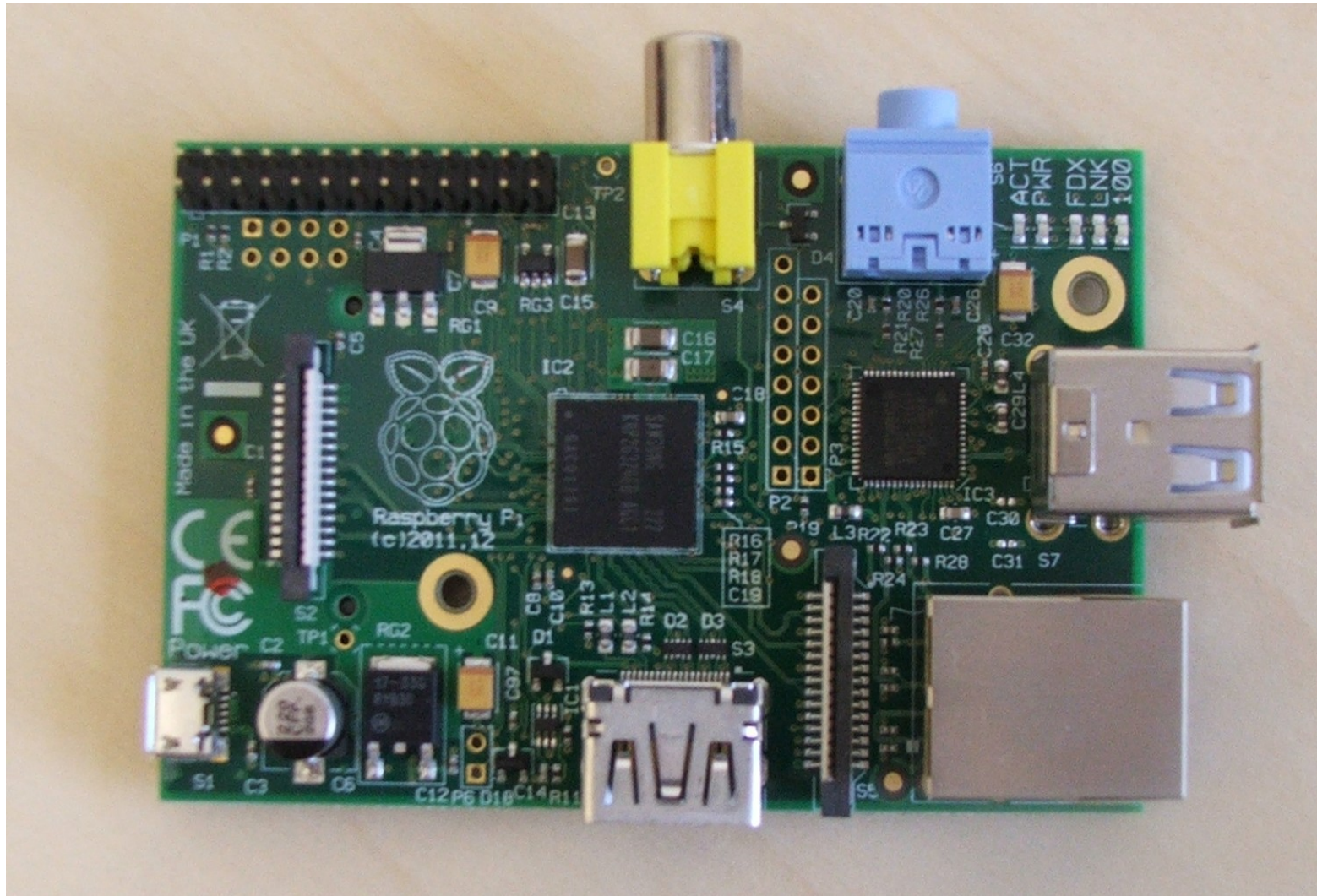
Telecos.cat
enginyers de telecomunicació,
electrònica i multimèdia-audiovisual

**22, 24, 29 de novembre
i 1 de desembre 2016**



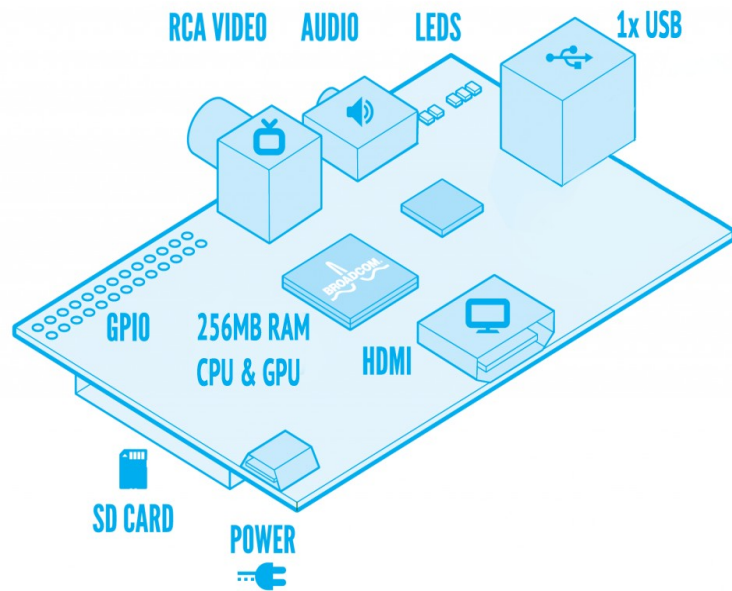
JESUÏTES El Clot
Escola del Clot

Què és la Raspberry Pi ?

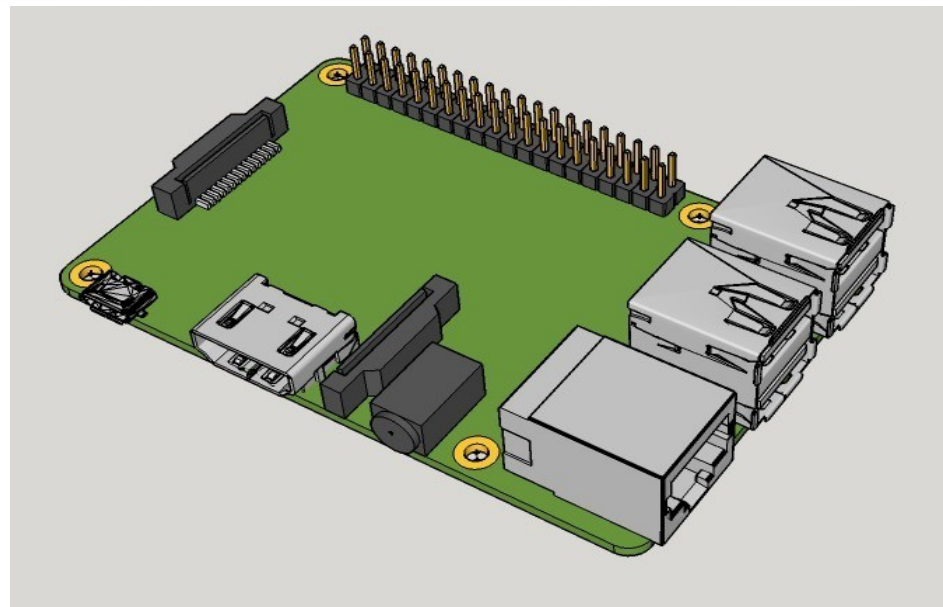
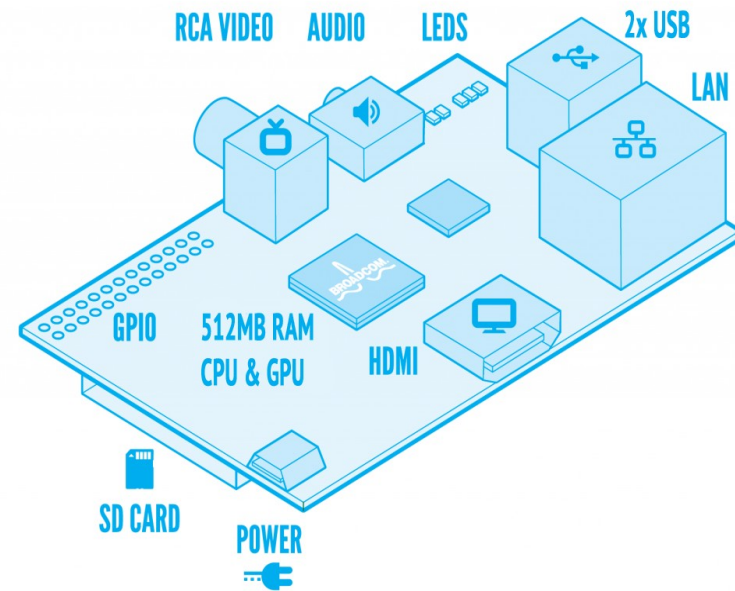


Raspberry Pi

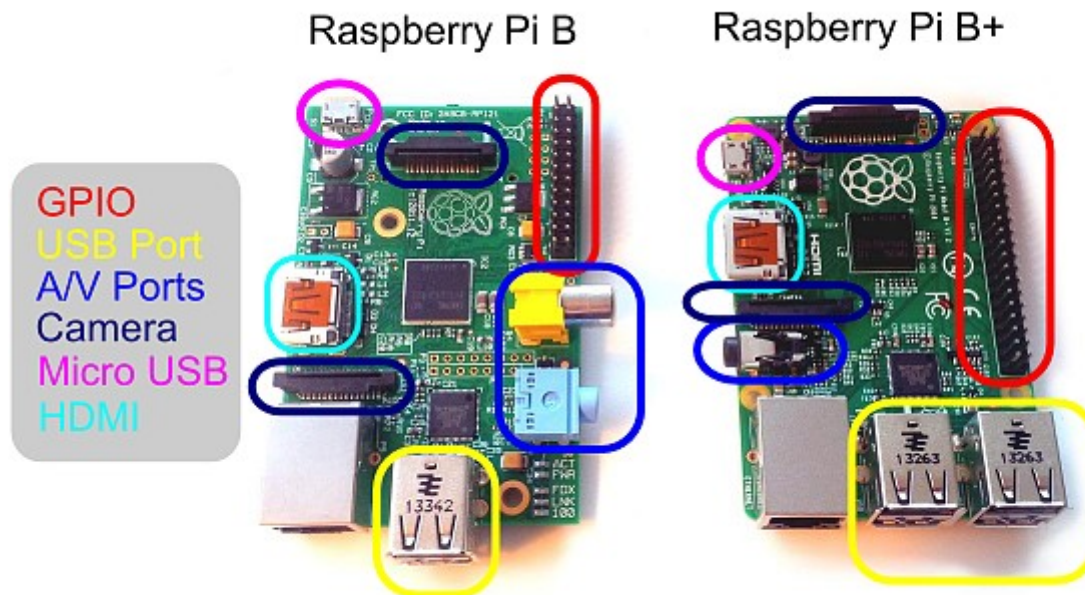
RASPBERRY PI MODEL A - \$25



RASPBERRY PI MODEL B - \$35

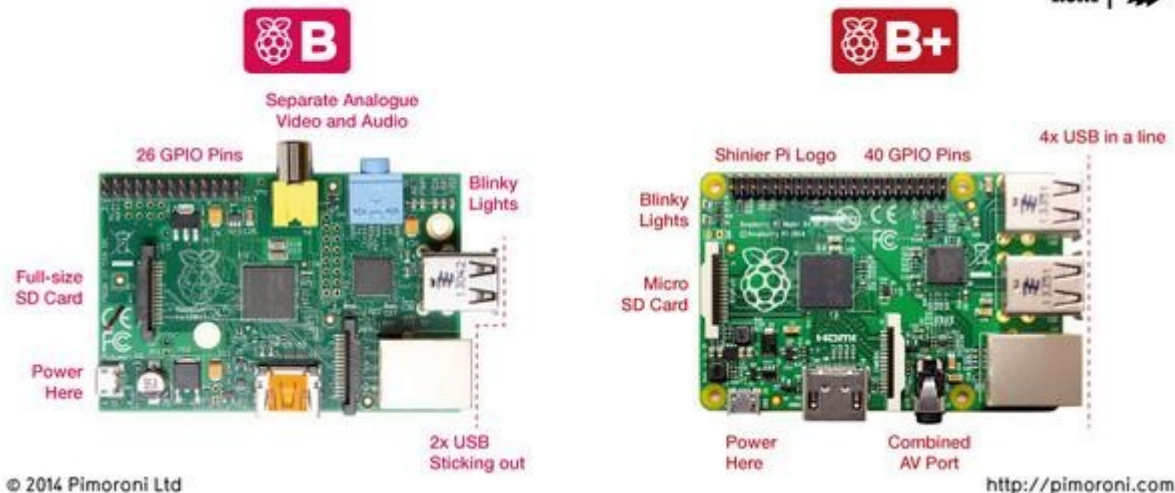


Raspberry Pi



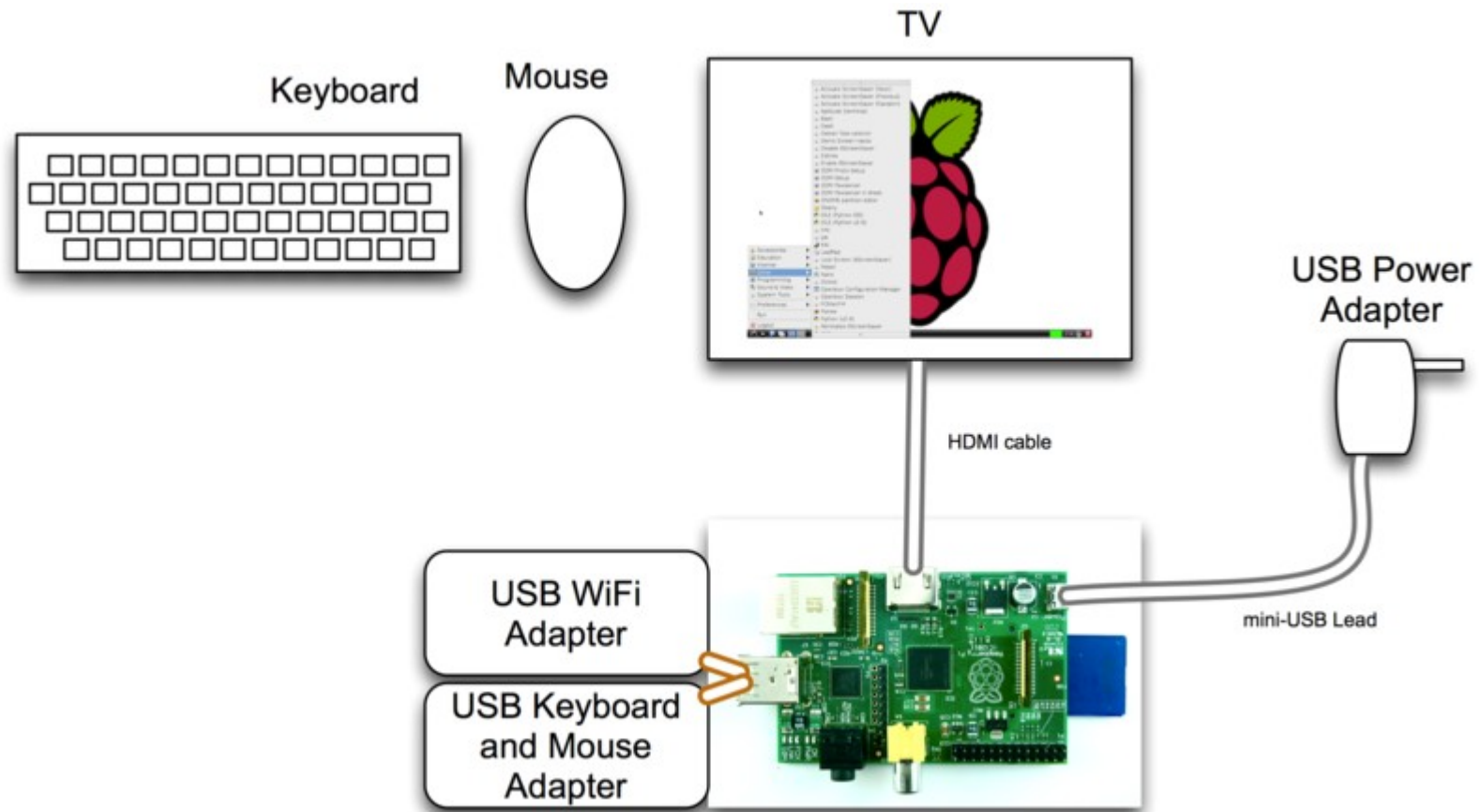
<https://helloacm.com/review-raspberry-pi-b-plus-model/>

Raspberry Pi: Which model do I have?



<http://misapuntedesde.com/post.php?id=431>

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



Altre maquinari

- Ports GPIO, sèrie, SPI, I2C i JTAG
- Ve amb un connector soldat (P1) de 26 pines i un altre de 8 pines (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

GPIO Numbers

Raspberry Pi B
Rev 1 P1 GPIO Header

Pin No.		
3.3V	1 2	5V
GPIO0	3 4	5V
GPIO1	5 6	GND
GPIO4	7 8	GPIO14
GND	9 10	GPIO15
GPIO17	11 12	GPIO18
GPIO21	13 14	GND
GPIO22	15 16	GPIO23
3.3V	17 18	GPIO24
GPIO10	19 20	GND
GPIO9	21 22	GPIO25
GPIO11	23 24	GPIO8
GND	25 26	GPIO7

Raspberry Pi A/B
Rev 2 P1 GPIO Header

Pin No.		
3.3V	1 2	5V
GPIO2	3 4	5V
GPIO3	5 6	GND
GPIO4	7 8	GPIO14
GND	9 10	GPIO15
GPIO17	11 12	GPIO18
GPIO27	13 14	GND
GPIO22	15 16	GPIO23
3.3V	17 18	GPIO24
GPIO10	19 20	GND
GPIO9	21 22	GPIO25
GPIO11	23 24	GPIO8
GND	25 26	GPIO7

Raspberry Pi B+
B+ J8 GPIO Header

Pin No.		
3.3V	1 2	5V
GPIO2	3 4	5V
GPIO3	5 6	GND
GPIO4	7 8	GPIO14
GND	9 10	GPIO15
GPIO17	11 12	GPIO18
GPIO27	13 14	GND
GPIO22	15 16	GPIO23
3.3V	17 18	GPIO24
GPIO10	19 20	GND
GPIO9	21 22	GPIO25
GPIO11	23 24	GPIO8
GND	25 26	GPIO7
DNC	27 28	DNC
GPIO5	29 30	GND
GPIO6	31 32	GPIO12
GPIO13	33 34	GND
GPIO19	35 36	GPIO16
GPIO26	37 38	GPIO20
GND	39 40	GPIO21

Key

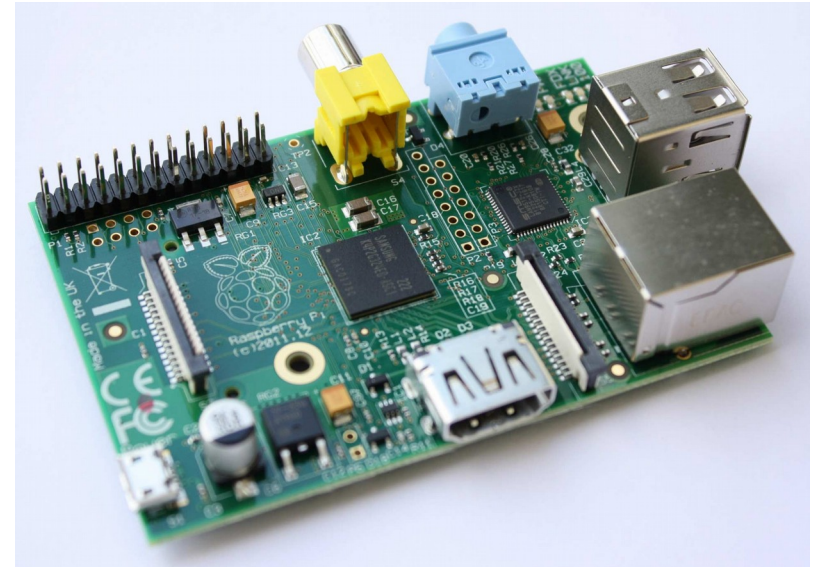
Power +	UART
GND	SPI
I²C	GPIO

Raspberry Pi Model B

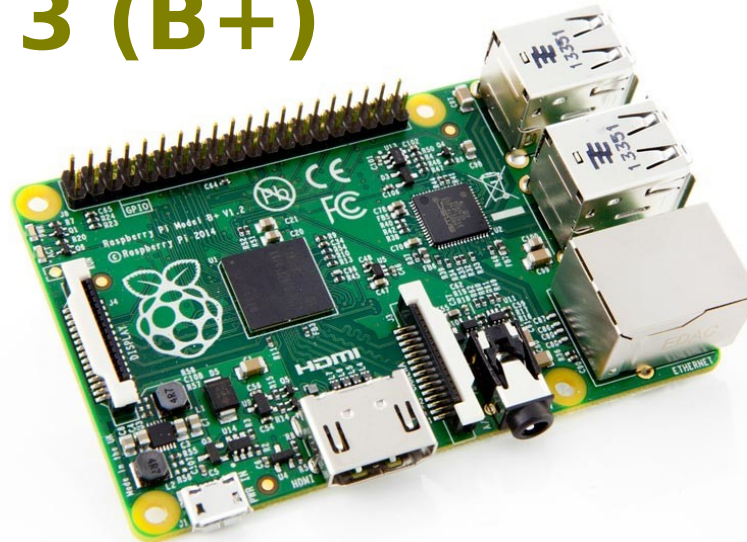
Rev 1



Rev 2



Rev 3 (B+)



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



Arduino i Raspberry Pi

Configuració de xarxa

WiFi : <https://www.raspberrypi.org/documentation/configuration/wireless/wireless-cli.md>

Ethernet : <http://hectorgarciaperez.com/2013/05/01/configurar-interfaces-de-red-en-debian-6/>

eth0 amb IP estàtica

```
# interfaces(5) file used by ifup(8) and ifdown(8)

# Please note that this file is written to be used with dhcpcd
# For static IP, consult /etc/dhcpcd.conf and 'man dhcpcd.conf'

# Include files from /etc/network/interfaces.d:
source-directory /etc/network/interfaces.d

auto lo
iface lo inet loopback

#eth0 amb DHCP
#allow-hotplug eth0
#iface eth0 inet dhcp

#eth0 amb IP estàtica
allow-hotplug eth0
iface eth0 inet static
    address 192.168.1.2
    netmask 255.255.255.0
    network 192.168.1.0
    broadcast 192.168.1.255
    gateway 192.168.1.1
    dns-nameservers 80.58.0.33 80.58.32.97

#WIFI DHCP
allow-hotplug wlan0
iface wlan0 inet dhcp
    wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf
```

eth0 amb IP dinàmica

```
# interfaces(5) file used by ifup(8) and ifdown(8)

# Please note that this file is written to be used with dhcpcd
# For static IP, consult /etc/dhcpcd.conf and 'man dhcpcd.conf'

# Include files from /etc/network/interfaces.d:
source-directory /etc/network/interfaces.d

auto lo
iface lo inet loopback

#eth0 amb DHCP
allow-hotplug eth0
iface eth0 inet dhcp

#eth0 amb IP estàtica
#allow-hotplug eth0
#iface eth0 inet static
#    address 192.168.1.2
#    netmask 255.255.255.0
#    network 192.168.1.0
#    broadcast 192.168.1.255
#    gateway 192.168.1.1
#    dns-nameservers 80.58.0.33 80.58.32.97

#WIFI DHCP
allow-hotplug wlan0
iface wlan0 inet dhcp
    wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf
```

/etc/network/interfaces



Arduino i Raspberry Pi

Configuració de xarxa

Si modifiqueu **wlan0** a /etc/network/interfaces i no voleu reiniciar l'equip, executeu:

```
ifdown wlan0  
ifup wlan0
```

Si modifiqueu **eth0** a /etc/network/interfaces i no voleu reiniciar l'equip, executeu:

```
ifdown eth0  
ifup eth0
```

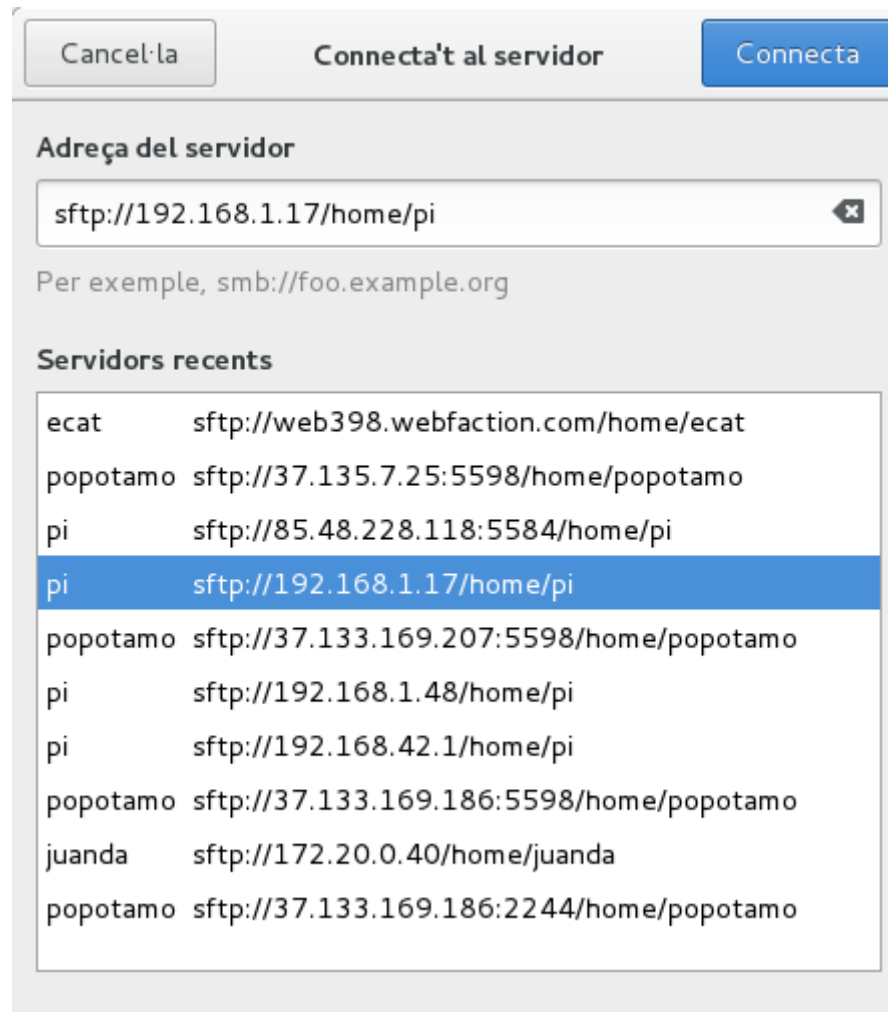
/etc/wpa_supplicant/wpa_supplicant.conf

```
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev  
update_config=1  
#country=GB  
  
network={  
    ssid="MOVISTAR_2840"  
    psk="TFTMX3YTUFRC4NAFJUJJ"  
}  
  
network={  
    ssid="Telecos.cat"  
    psk="Fourier17"  
    key_mgmt=WPA-PSK  
}  
  
network={  
    ssid="IoT-eCat"  
    psk="clotClot"  
    key_mgmt=WPA-PSK  
}
```



Arduino i Raspberry Pi

Connexió remota emprant el navegador d'arxius





Arduino i Raspberry Pi

ifconfig

```

pi@raspberrypi:~ $ ifconfig
eth0      Link encap:Ethernet  HWaddr b8:27:eb:fd:d0:b5
          inet addr:192.168.1.17  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::b88e:e7c9:8c24:45d7/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1140 errors:0 dropped:7 overruns:0 frame:0
          TX packets:206 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:77745 (75.9 KiB)  TX bytes:28255 (27.5 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:226 errors:0 dropped:0 overruns:0 frame:0
          TX packets:226 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:20208 (19.7 KiB)  TX bytes:20208 (19.7 KiB)

wlan0     Link encap:Ethernet  HWaddr b8:27:eb:a8:85:e0
          inet addr:192.168.42.1  Bcast:192.168.42.255  Mask:255.255.255.0
          inet6 addr: fe80::4504:2549:cd46:698/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:246 errors:0 dropped:168 overruns:0 frame:0
          TX packets:85 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:39016 (38.1 KiB)  TX bytes:17059 (16.6 KiB)

pi@raspberrypi:~ $ █

```



Arduino i Raspberry Pi

Porta d'enllaç i DNS

```
pi@raspberrypi:~ $ route -n
Kernel IP routing table
Destination      Gateway          Genmask          Flags Metric Ref    Use Iface
0.0.0.0          192.168.1.1     0.0.0.0          UG    202    0      0 eth0
169.254.0.0      0.0.0.0         255.255.0.0      U     303    0      0 wlan0
192.168.1.0      0.0.0.0         255.255.255.0    U     202    0      0 eth0
192.168.42.0     0.0.0.0         255.255.255.0    U      0      0      0 wlan0
pi@raspberrypi:~ $ cat /etc/resolv.conf
# Generated by resolvconf
nameserver 62.81.29.254
nameserver 62.81.16.213
pi@raspberrypi:~ $ █
```




Arduino i Raspberry Pi

SSH

ssh -X pi@192.168.1.17

```
jordi@debianJB:~$ ssh -X pi@192.168.1.17
pi@192.168.1.17's password:
```

```
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 Nov 30 10:09:47 2016 from 192.168.1.12
```

```
pi@raspberrypi:~$ █
```

L'opció -X ens permet poder executar remotament programes amb interfície gràfica d'usuari.

En sistemes operatius Windows existeix el programa putty (<http://www.putty.org/>)



Arduino i Raspberry Pi

Accés SSH sense contrasenya

<https://docs.webfaction.com/user-guide/access.html>

Genereu una clau al vostre ordinador (si no existeix `~/.ssh/id_rsa.pub`):

- 1- Obriu una sessió de terminal.
- 2- Creeu la carpeta `~/.ssh`, en cas de que no existeixi. (`mkdir -p $HOME/.ssh`)
- 3- Aneu al directori `~/.ssh` (`cd ~/.ssh` i premeu Enter)
- 4- Genereu les vostres claus (`ssh-keygen -t rsa`)
- 5- Premeu Enter per a contestar totes les preguntes per defecte.

Desplegueu la clau a la Raspberry Pi

- 1- Copieu la clau a la Raspberry Pi. Entreu `scp ~/.ssh/id_rsa.pub pi@192.168.1.17:temp_id_rsa_key.pub` (canvieu 192.168.1.17 per la IP de la vostra Raspberry Pi) i premeu Enter.
- Entreu la contrasenya quan us ho demani (la contrasenya per defecte és raspberry).

```
jordi@debianJB:~$ scp ~/.ssh/id_rsa.pub pi@192.168.1.17:temp_id_rsa_key.pub
pi@192.168.1.17's password:
id_rsa.pub                                100% 396      0.4KB/s   00:00
jordi@debianJB:~$ █
```



Arduino i Raspberry Pi

Accés SSH sense contrasenya

Desplegueu la clau a la Raspberry Pi

2- Obriu una sessió SSH a la vostra Raspberry Pi des del vostre ordinador. Entreu **ssh pi@192.168.1.17** (canvieu 192.168.1.17 per la IP de la vostra Raspberry Pi) i premeu Enter. Entreu la contrasenya quan us ho demani (la contrasenya per defecte és **raspberrypi**).

```
jordi@debianJB:~$ ssh pi@192.168.1.17
pi@192.168.1.17's password:
```

```
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 Nov 30 10:18:28 2016 from 192.168.1.12
```

```
pi@raspberrypi:~$ █
```

3- Creeu la carpeta `~/ .ssh`, en cas de que no existeixi. (`mkdir -p $HOME/.ssh`)

4- Afegiu la vostra clau a l'arxiu `authorized_keys` . Entreu `cat ~/temp_id_rsa_key.pub >> ~/.ssh/authorized_keys` i premeu Enter.

```
pi@raspberrypi:~$ cat ~/temp_id_rsa_key.pub >> ~/.ssh/authorized_keys
```



Arduino i Raspberry Pi

Accés SSH sense contrasenya

Desplegueu la clau a la Raspberry Pi

5- Esborreu l'arxiu amb la clau temporal. Feu `rm temp_id_rsa_key.pub` i premeu Enter.

```
pi@raspberrypi:~ $ rm temp_id_rsa_key.pub
```

6- Protegiu l'arxiu de claus SSH. Entreu-hi `chmod 600 ~/.ssh/authorized_keys` i premeu Enter.

```
pi@raspberrypi:~ $ chmod 600 ~/.ssh/authorized keys
```

7- Protegiu el directori SSH. Entreu-hi `chmod 700 ~/.ssh` i premeu Enter.

```
pi@raspberrypi:~ $ chmod 700 ~/.ssh
```

8- Protegiu el vostre directori d'usuari. Entreu-hi `chmod go-w $HOME` i premeu Enter.

```
pi@raspberrypi:~ $ chmod go-w $HOME
```

9- Tanqueu la sessió SSH.



Arduino i Raspberry Pi

Accés SSH sense contrasenya

Desplegueu la clau a la Raspberry Pi

```
pi@raspberrypi:~ $ cat ~/temp_id_rsa_key.pub >> ~/.ssh/authorized_keys
pi@raspberrypi:~ $ rm temp_id_rsa_key.pub
pi@raspberrypi:~ $ chmod 600 ~/.ssh/authorized_keys
pi@raspberrypi:~ $ chmod 700 ~/.ssh
pi@raspberrypi:~ $ chmod go-w $HOME
pi@raspberrypi:~ $ exit
logout
Connection to 192.168.1.17 closed.
jordi@debianJB:~$
```

**Verifiqueu que ja podeu accedir mitjançant SSH sense que us
pregunti la contrasenya**

```
jordi@debianJB:~$ ssh pi@192.168.1.17

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.

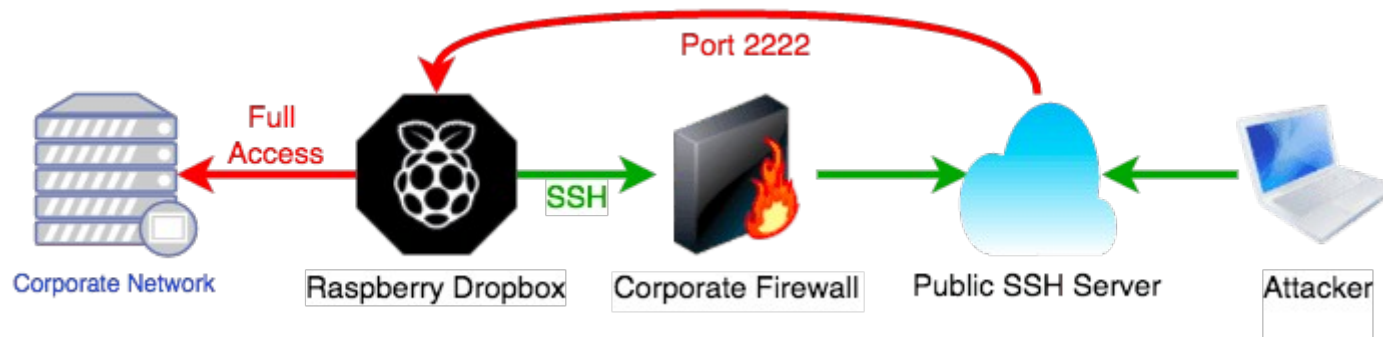
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Nov 30 11:03:54 2016 from 192.168.1.12
pi@raspberrypi:~ $
```



Arduino i Raspberry Pi

Túnel SSH invers

<http://jerrygamblin.com/2016/04/23/persistent-reverse-ssh-tunnels-on-a-raspberrypi/>





Arduino i Raspberry Pi

Túnel SSH invers

```
jordi@debianJB: ~
Fitxer  Edita  Visualitza  Cerca  Terminal  Ajuda

pi@raspberrypi:~ $ ssh -R 12345:localhost:22 jordi@192.168.1.12
The authenticity of host '192.168.1.12 (192.168.1.12)' can't be established.
ECDSA key fingerprint is 01:eb:89:96:25:8b:48:22:e9:a7:ed:5e:d4:98:c4:ac.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.1.12' (ECDSA) to the list of known hosts.
jordi@192.168.1.12's password:

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.
jordi@debianJB:~$
```

```
pi@raspberrypi: ~
Fitxer  Edita  Visualitza  Cerca  Terminal  Ajuda

jordi@debianJB:~$ ssh -p 12345 pi@127.0.0.1
The authenticity of host '[127.0.0.1]:12345 ([127.0.0.1]:12345)' can't be established.
ECDSA key fingerprint is db:79:95:3e:5d:97:96:cd:ce:57:64:96:63:82:15:e2.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '[127.0.0.1]:12345' (ECDSA) to the list of known hosts.

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 Nov 30 23:46:52 2016 from 192.168.1.12
pi@raspberrypi:~ $
```



Arduino i Raspberry Pi

autoSSH

Si el túnel deixa de funcionar (acostuma a ocórrer en xarxes molt ocupades o d'escasa qualitat, com el 3G), es pot utilitzar el paquet autossh en lloc de l'ssh per a establir la connexió que s'encarregarà de mantenir el túnel obert reiniciant automàticament la connexió.

```
pi@raspberrypi:~ $ autossh -M 65500 -o ServerAliveInterval=20 -R 19994:localhost:22 ecat@web398.webfaction.com
```

L'autossh no funciona com s'espera si al connectar la Raspberry Pi no hi ha connexió a Internet. Per a evitar això, primer verifiquem la connectivitat fent un ping.

```
jordi@debianJB:~$ ssh ecat@web398.webfaction.com
ecat@web398.webfaction.com's password:
Last login: Wed Nov 30 23:21:27 2016 from 62.83.215.143
[ecat@web398 ~]$ ssh -p 19994 pi@localhost
pi@localhost's password:
```

```
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: Thu Dec 1 00:24:02 2016 from localhost
```

```
pi@raspberrypi:~ $ █
```




Arduino i Raspberry Pi

Verificació de la connectivitat

```
pi@raspberrypi:~ $ ping -c 1 www.binefa.cat
PING binefa.cat (37.58.75.228) 56(84) bytes of data.
64 bytes from web398.webfaction.com (37.58.75.228): icmp_seq=1 ttl=52 time=49.2 ms

--- binefa.cat ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 49.200/49.200/49.200/0.000 ms
pi@raspberrypi:~ $ echo $?
0
pi@raspberrypi:~ $ ping -c 1 www.jordibinefa.cat
ping: unknown host www.jordibinefa.cat
pi@raspberrypi:~ $ echo $?
2
pi@raspberrypi:~ $ ping -c 1 192.168.1.222
PING 192.168.1.222 (192.168.1.222) 56(84) bytes of data.
From 192.168.1.17 icmp_seq=1 Destination Host Unreachable

--- 192.168.1.222 ping statistics ---
1 packets transmitted, 0 received, +1 errors, 100% packet loss, time 0ms

pi@raspberrypi:~ $ echo $?
1
pi@raspberrypi:~ $ ping -c 1 192.168.1.1
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.
64 bytes from 192.168.1.1: icmp_seq=1 ttl=64 time=1.01 ms

--- 192.168.1.1 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 1.019/1.019/1.019/0.000 ms
pi@raspberrypi:~ $ echo $?
0
pi@raspberrypi:~ $ █
```



Arduino i Raspberry Pi

Ordres consecutives emprant &&

```
pi@raspberrypi:~ $ ping -c 1 192.168.1.222 && ls
PING 192.168.1.222 (192.168.1.222) 56(84) bytes of data.
From 192.168.1.17 icmp_seq=1 Destination Host Unreachable

--- 192.168.1.222 ping statistics ---
1 packets transmitted, 0 received, +1 errors, 100% packet loss, time 0ms

pi@raspberrypi:~ $ ping -c 1 192.168.1.1 && ls
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.
64 bytes from 192.168.1.1: icmp_seq=1 ttl=64 time=1.01 ms

--- 192.168.1.1 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 1.019/1.019/1.019/0.000 ms
codis      Documents Music      Pictures  pwdVnc.txt  sketchbook  tunelSSH.txt
Desktop Downloads oldconffiles Public  python_games  Templates  Videos
pi@raspberrypi:~ $ ping -c 1 www.jordibinefa.cat && ls
ping: unknown host www.jordibinefa.cat
pi@raspberrypi:~ $ ping -c 1 www.binefa.cat && ls
PING binefa.cat (37.58.75.228) 56(84) bytes of data.
64 bytes from web398.webfaction.com (37.58.75.228): icmp_seq=1 ttl=52 time=48.6 ms

--- binefa.cat ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 48.642/48.642/48.642/0.000 ms
codis      Documents Music      Pictures  pwdVnc.txt  sketchbook  tunelSSH.txt
Desktop Downloads oldconffiles Public  python_games  Templates  Videos
pi@raspberrypi:~ $ █
```

En cas de que la primera ordre tingui èxit, llavors executa la següent.



Arduino i Raspberry Pi

Processos actius i grep

```

pi@raspberrypi:~ $ ps aux | grep autossh
pi          2586  0.0  0.2  4772  2032 pts/0    S+   12:18   0:00 grep --color=auto autossh
pi@raspberrypi:~ $ ps aux | grep autossh | grep -v grep
pi@raspberrypi:~ $ if ! ps aux | grep autossh > /dev/null; then echo "No hi ha cap procés"; else echo "Hi ha algun procés"; fi
Hi ha algun procés
pi@raspberrypi:~ $ if ! ps aux | grep autossh | grep -v grep > /dev/null; then echo "No hi ha cap procés"; else echo "Hi ha algun pro
cés"; fi
No hi ha cap procés
pi@raspberrypi:~ $ if ! ps aux | grep autossh > /dev/null; then echo "No hi ha cap procés"; fi
pi@raspberrypi:~ $ if ! ps aux | grep autossh | grep -v grep > /dev/null; then echo "No hi ha cap procés"; fi
No hi ha cap procés
pi@raspberrypi:~ $ █

```



Arduino i Raspberry Pi

crontab

/etc/crontab

```
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# m h dom mon dow user  command
17 * * * * root    cd / && run-parts --report /etc/cron.hourly
25 6 * * * root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
47 6 * * 7 root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
52 6 1 * * root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )
*/1 * * * * root    ping -c 1 www.binefa.cat && if ! ps aux | grep autossh | grep -v grep > /dev/nl
#
```

```
*/1 * * * * root    ping -c 1 www.binefa.cat
&& if ! ps aux | grep autossh | grep -v grep >
/dev/null; then su pi -c 'autossh -f -nNT -M 65500
-o ServerAliveInterval=20 -R 19994:localhost:22
ecat@web398.webfaction.com'; fi
```




Arduino i Raspberry Pi

Persistència de les accions remotes screen (1/2)

<http://www.tecmint.com/screen-command-examples-to-manage-linux-terminals/>

```
pi@raspberrypi:~ $ screen -ls
No Sockets found in /var/run/screen/S-pi.
```

```
pi@raspberrypi:~ $ screen
```

```
pi@raspberrypi:~ $ ping 192.168.1.1
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.
64 bytes from 192.168.1.1: icmp_seq=1 ttl=64 time=2.07 ms
64 bytes from 192.168.1.1: icmp_seq=2 ttl=64 time=0.919 ms
64 bytes from 192.168.1.1: icmp_seq=3 ttl=64 time=1.17 ms
64 bytes from 192.168.1.1: icmp_seq=4 ttl=64 time=0.960 ms
64 bytes from 192.168.1.1: icmp_seq=5 ttl=64 time=0.934 ms
64 bytes from 192.168.1.1: icmp_seq=6 ttl=64 time=1.33 ms
```

Per a sortir temporalment d'screen premeu **Ctrl A + D**

```
[detached from 2427.pts-0.raspberrypi]
pi@raspberrypi:~ $
```



Arduino i Raspberry Pi

Persistència de les accions remotes screen (2/2)

<http://www.tecmint.com/screen-command-examples-to-manage-linux-terminals/>

```
[detached from 2427.pts-0.raspberrypi]
pi@raspberrypi:~ $ exit
logout
Connection to 192.168.1.17 closed.
jordi@debianJB:~$ ssh pi@192.168.1.17
```

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 Nov 30 11:55:07 2016 from 192.168.1.12

```
pi@raspberrypi:~ $ screen -r
```

```
64 bytes from 192.168.1.1: icmp_seq=205 ttl=64 time=0.883 ms
64 bytes from 192.168.1.1: icmp_seq=206 ttl=64 time=0.881 ms
64 bytes from 192.168.1.1: icmp_seq=207 ttl=64 time=0.969 ms
64 bytes from 192.168.1.1: icmp_seq=208 ttl=64 time=0.873 ms
64 bytes from 192.168.1.1: icmp_seq=209 ttl=64 time=1.10 ms
64 bytes from 192.168.1.1: icmp_seq=210 ttl=64 time=0.927 ms
64 bytes from 192.168.1.1: icmp_seq=211 ttl=64 time=0.906 ms
64 bytes from 192.168.1.1: icmp_seq=212 ttl=64 time=0.860 ms
```

```
[screen is terminating]
pi@raspberrypi:~ $
```

Per a cancel·lar el ping podem prémer Ctrl C i per a sortir de screen fem exit



Arduino i Raspberry Pi

Publicació de la IP pública (1/2)

← ⓘ | binefa.cat/php/svr/

37.133.169.186
dc nov 30 13:00:02 CET 2016

```
popotamo@popotamo:~$ cat /etc/crontab
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# m h dom mon dow user  command
17 * * * * root    cd / && run-parts --report /etc/cron.hourly
25 6 * * * root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
47 6 * * 7 root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
52 6 1 * * root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )
#
*/5 * * * * popotamo /home/popotamo/bin/svr.sh
#
*/1 * * * * popotamo /home/popotamo/bin/checkAutoSSH.sh
popotamo@popotamo:~$ █
```



Arduino i Raspberry Pi

Publicació de la IP pública (2/2)

```
popotamo@popotamo:~$ cat /home/popotamo/bin/svr.sh
#!/bin/bash
```

```
# rm ip.php
wget http://www.binefa.cat/php/ip/ip.php
echo "<BR>" >> ip.php
date >> ip.php
```

```
scp ip.php ecat@web398.webfaction.com:/home/ecat/webapps/www_binefa_cat_php/svr/index.php
```

```
rm ip.php
popotamo@popotamo:~$ █
```

```
< ui_ctrlLeds01.py x ecat.cpp x index.htm x index.php x
1 <META HTTP-EQUIV="refresh" CONTENT="0;URL=ip.php">
2
```

```
ecat.cpp x index.htm x index.php x ip.php x >
1 <?php
2 echo $_SERVER['REMOTE_ADDR'];
3 ?>
4
```

```
ecat.cpp x index.htm x index.php x ip.php x
1 <HTML>
2 <HEAD>
3 <TITLE>Your Internet IP</TITLE>
4 </HEAD>
5 <BODY>
6 <?php
7 echo $_SERVER['REMOTE_ADDR'];
8 ?>
9 </BODY>
10 </HTML>
11
```



Arduino i Raspberry Pi

Raspberry Pi com a punt d'accés (1/4)

<http://elinux.org/RPI-Wireless-Hotspot>

Instal·lació de paquets:

sudo apt-get install hostapd udhcpd

```
pi@raspberrypi:~ $ cat /etc/udhcpd.conf
# Sample udhcpd configuration file (/etc/udhcpd.conf)

# The start and end of the IP lease block

start          192.168.42.20    #default: 192.168.0.20
end            192.168.42.254  #default: 192.168.0.254

# The interface that udhcpd will use

interface      wlan0           #default: eth0

#Examples
#opt    dns      192.168.10.2 192.168.10.10
opt     dns      8.8.8.8 4.2.2.2
option  subnet   255.255.255.0
#opt    router   192.168.10.2
opt     router   192.168.42.1
#opt    wins     192.168.10.10
#option dns      129.219.13.81  # appened to above DNS servers for a total of 3
#option domain   local
option  lease     864000        # 10 days of seconds
```




Arduino i Raspberry Pi

Raspberry Pi com a punt d'accés (2/4)

```
pi@raspberrypi:~ $ cat /etc/default/udhcpd
# Comment the following line to enable
#DHCPD_ENABLED="no"

# Options to pass to busybox' udhcpd.
#
# -S      Log to syslog
# -f      run in foreground

DHCPD_OPTS="-S"
pi@raspberrypi:~ $ █
```

```
pi@raspberrypi:~ $ cat /etc/network/interfaces
# interfaces(5) file used by ifup(8) and ifdown(8)

# Please note that this file is written to be used with dhcpcd
# For static IP, consult /etc/dhcpcd.conf and 'man dhcpcd.conf'

# Include files from /etc/network/interfaces.d:
source-directory /etc/network/interfaces.d

auto lo
iface lo inet loopback

iface eth0 inet manual

allow-hotplug wlan0
#iface wlan0 inet manual
#    wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf

iface wlan0 inet static
    address 192.168.42.1
    netmask 255.255.255.0

allow-hotplug wlan1
iface wlan1 inet manual
    wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf

up iptables-restore < /etc/iptables.ipv4.nat
```



Arduino i Raspberry Pi

Raspberry Pi com a punt d'accés (3/4)

```
pi@raspberrypi:~ $ cat /etc/hostapd/hostapd.conf
```

```
interface=wlan0
driver=nl80211
ssid=pvSL
hw_mode=g
channel=10
macaddr_acl=0
auth_algs=1
ignore_broadcast_ssid=0
wpa=2
wpa_passphrase=fanalBolet0
wpa_key_mgmt=WPA-PSK
#wpa_pairwise=TKIP           # You better do not use this weak encryption (only used by old client devices)
rsn_pairwise=CCMP
ieee80211n=1                 # 802.11n support
wmm_enabled=1               # QoS support
ht_capab=[HT40][SHORT-GI-20][DSSS_CCK-40]
pi@raspberrypi:~ $ █
```

```
pi@raspberrypi:~ $ cat /etc/default/hostapd
# Defaults for hostapd initscript
#
# See /usr/share/doc/hostapd/README.Debian for information about
# methods of managing hostapd.
#
# Uncomment and set DAEMON_CONF to the absolute path of a hostap
# file and hostapd will be started during system boot. An exampl
# file can be found at /usr/share/doc/hostapd/examples/hostapd.c
#
DAEMON_CONF="/etc/hostapd/hostapd.conf"

# Additional daemon options to be appended to hostapd command:-
#       -d      show more debug messages (-dd for even more)
#       -K      include key data in debug messages
#       -t      include timestamps in some debug messages
#
# Note that -B (daemon mode) and -P (pidfile) options are automa
# configured by the init.d script and must not be added to DAEMC
#
#DAEMON_OPTS=""
pi@raspberrypi:~ $ █
```



Arduino i Raspberry Pi

Raspberry Pi com a punt d'accés (4/4)

A l'arxiu `/etc/sysctl.conf`

```
# Uncomment the next line to enable packet forwarding for IPv4
net.ipv4.ip_forward=1
```

```
pi@raspberrypi:~ $ cat /etc/iptables.ipv4.nat
# Generated by iptables-save v1.4.21 on Sat Oct  8 22:18:23 2016
*filter
:INPUT ACCEPT [36:2032]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [41:6275]
-A FORWARD -i eth0 -o wlan0 -m state --state RELATED,ESTABLISHED -j ACCEPT
-A FORWARD -i wlan0 -o eth0 -j ACCEPT
COMMIT
# Completed on Sat Oct  8 22:18:23 2016
# Generated by iptables-save v1.4.21 on Sat Oct  8 22:18:23 2016
*nat
:PREROUTING ACCEPT [0:0]
:INPUT ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
:POSTROUTING ACCEPT [0:0]
-A POSTROUTING -o eth0 -j MASQUERADE
COMMIT
# Completed on Sat Oct  8 22:18:23 2016
pi@raspberrypi:~ $ █
```

Per a reiniciar els serveis:

```
sudo update-rc.d hostapd enable
sudo update-rc.d udhcpd enable
```



Arduino i Raspberry Pi

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>

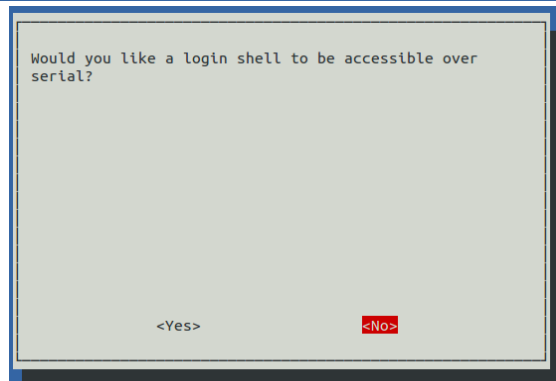
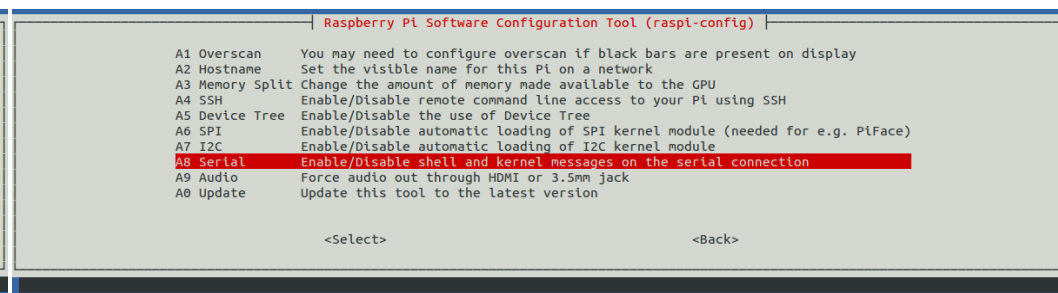
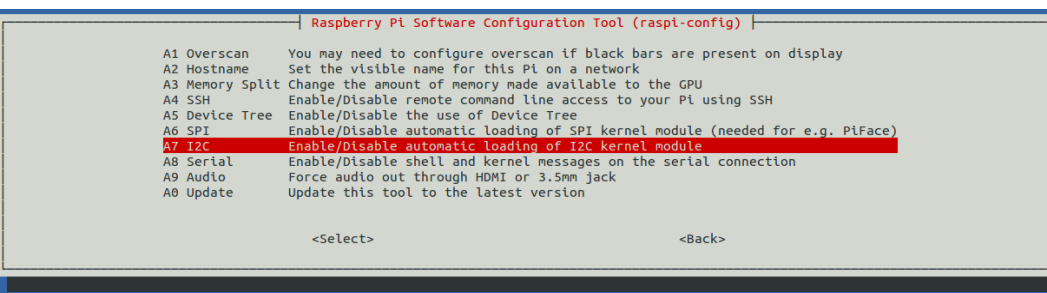
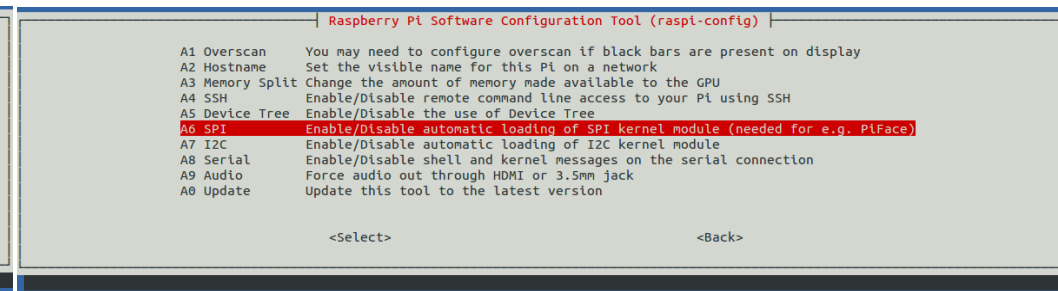
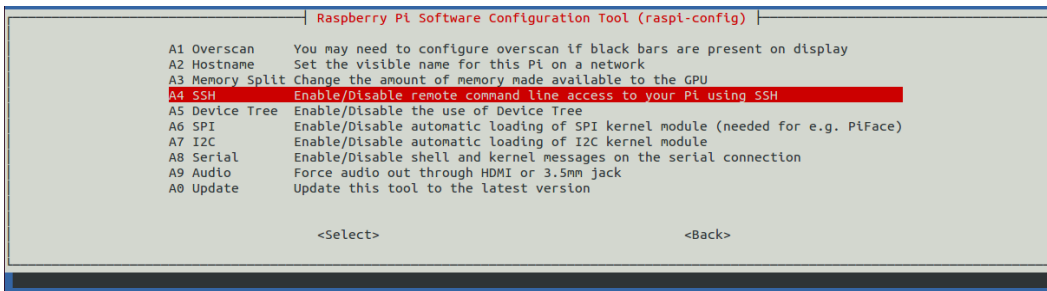
<Cancel>



Arduino i Raspberry Pi

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>



Arduino i Raspberry Pi

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
```



Arduino i Raspberry Pi

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 ~ $
```



Arduino i Raspberry Pi

Configuració de l'I2C

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

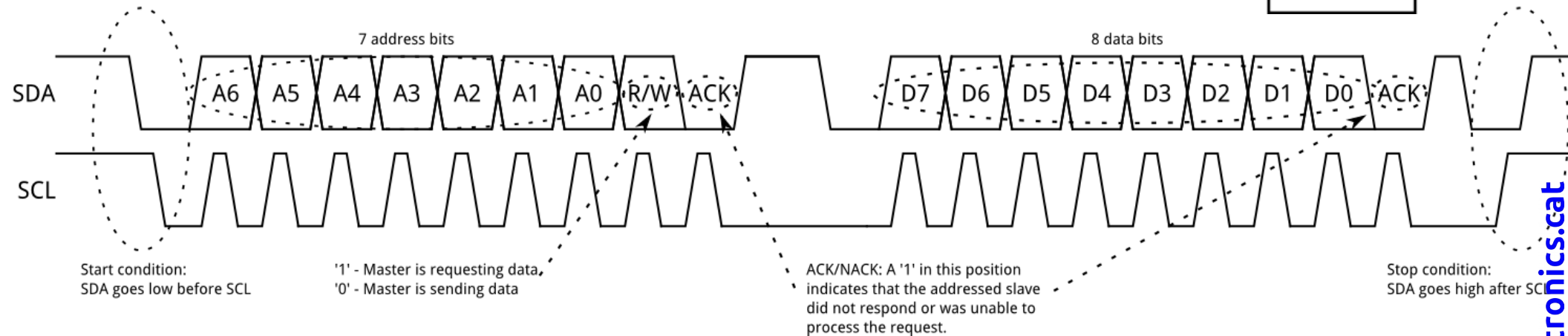
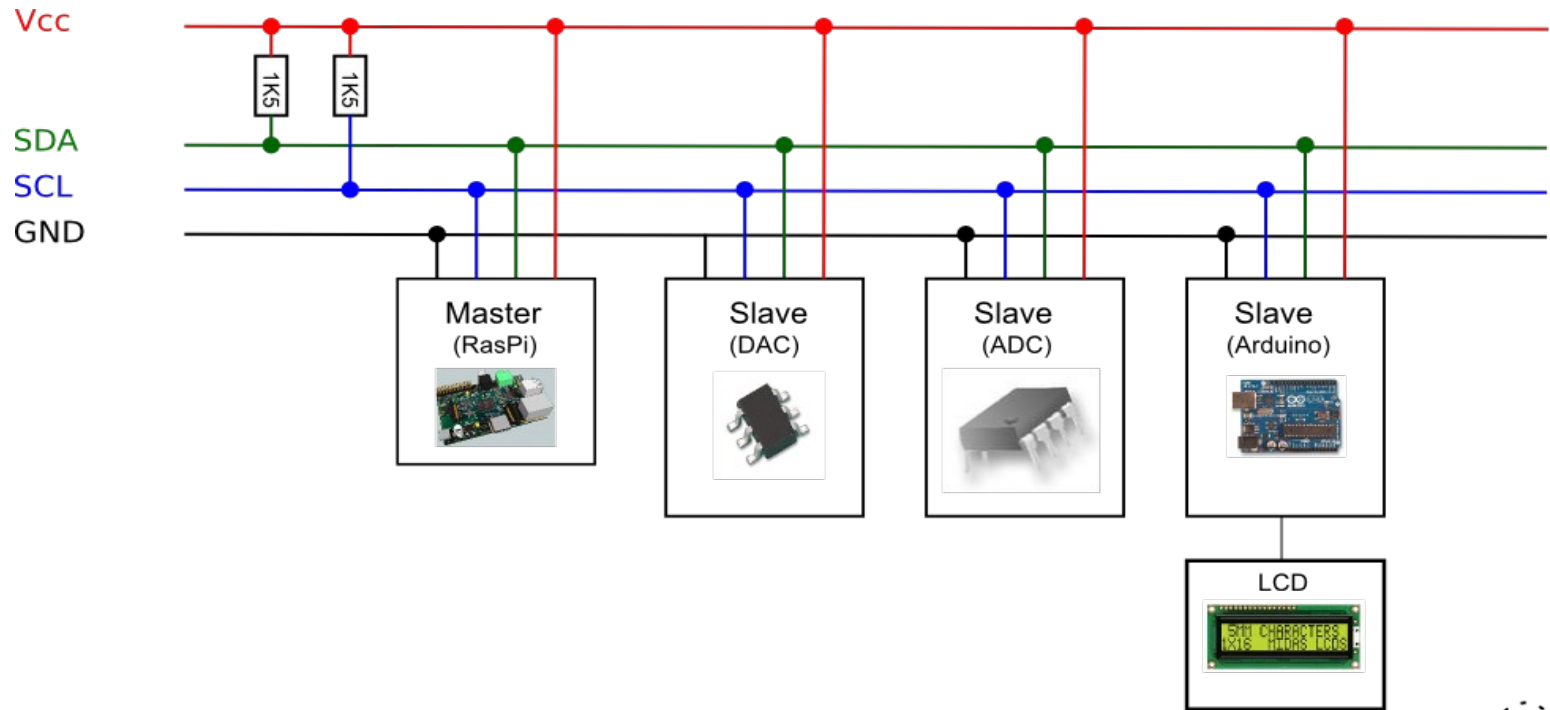
```
pi@raspberrypi:~ $ sudo i2cdetect -y 1
    0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
10:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
20:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
30:  --  --  --  --  --  --  --  --  38 39  --  --  --  --  --  --
40:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
50:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
60:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
70:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
pi@raspberrypi:~ $
```

```
sudo i2cset -y 1 0x38 0xA5
sudo i2cset -y 1 0x39 0xF0
sudo i2cset -y 1 0x39 0xF
```



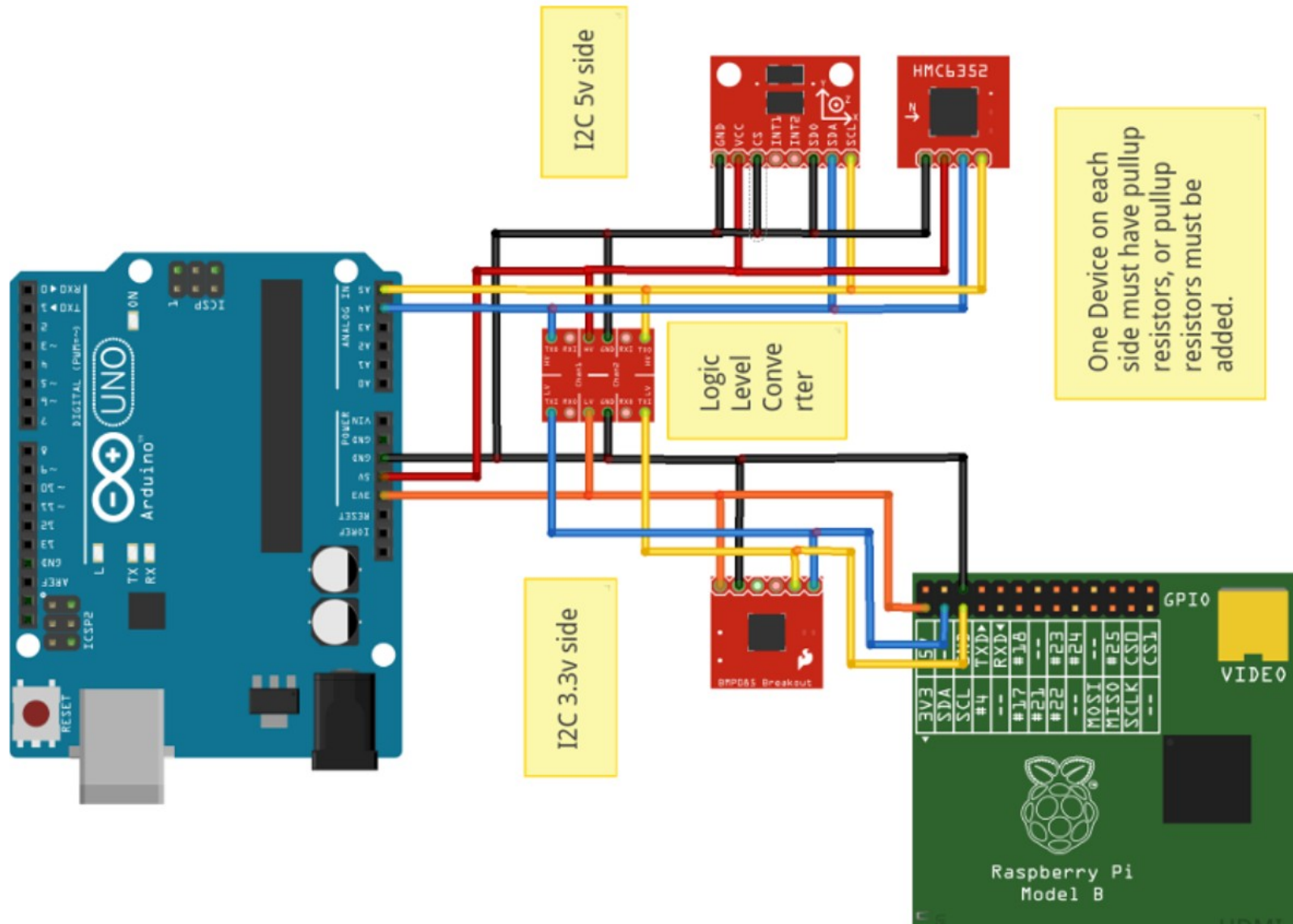
Arduino i Raspberry Pi

I2C



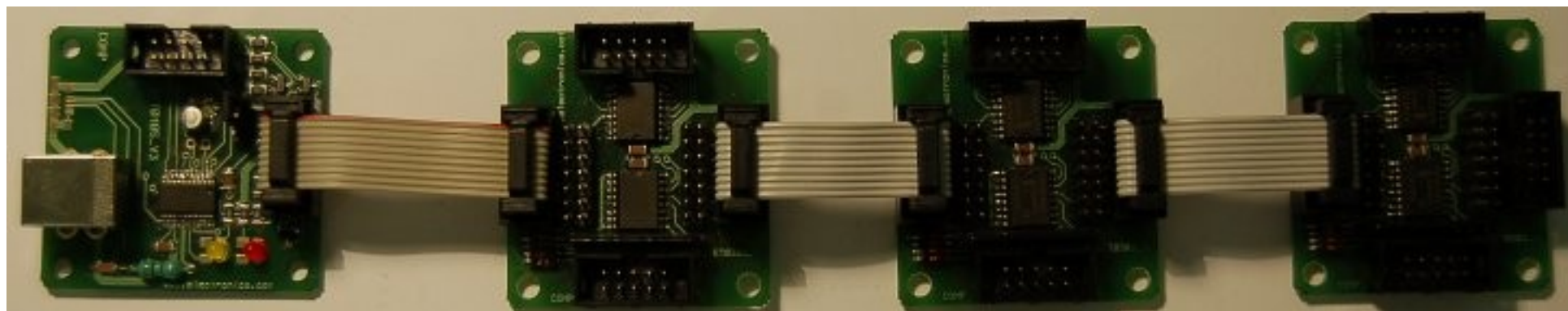
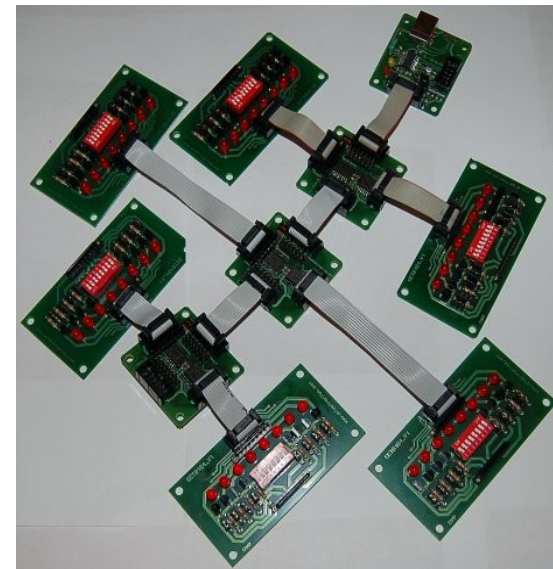
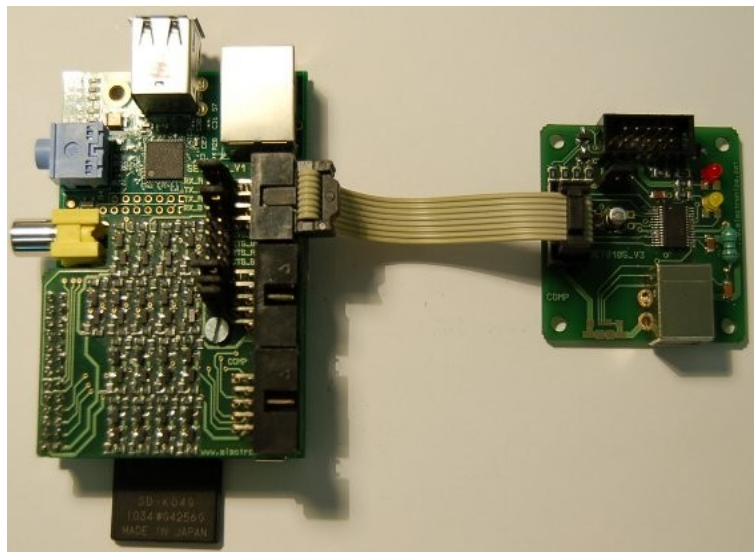
Arduino i Raspberry Pi

I2C



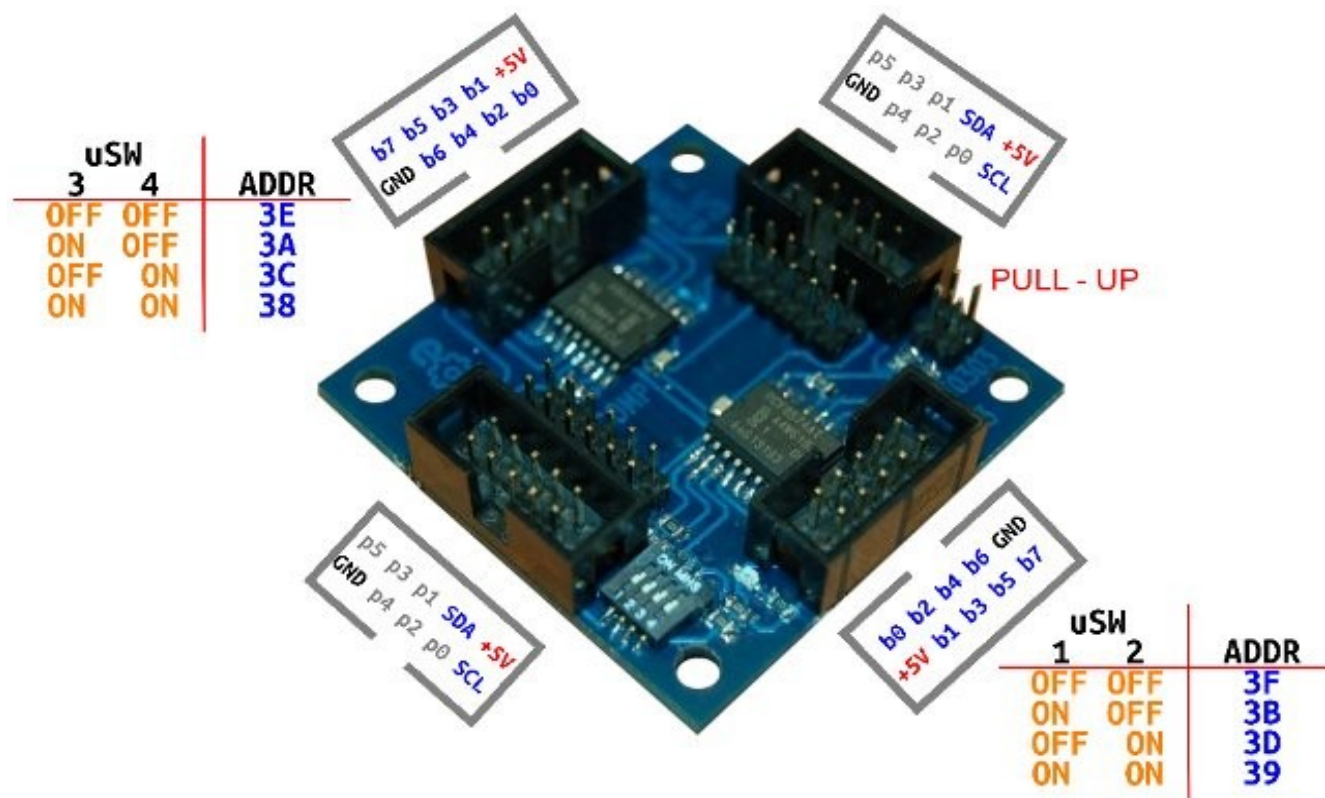
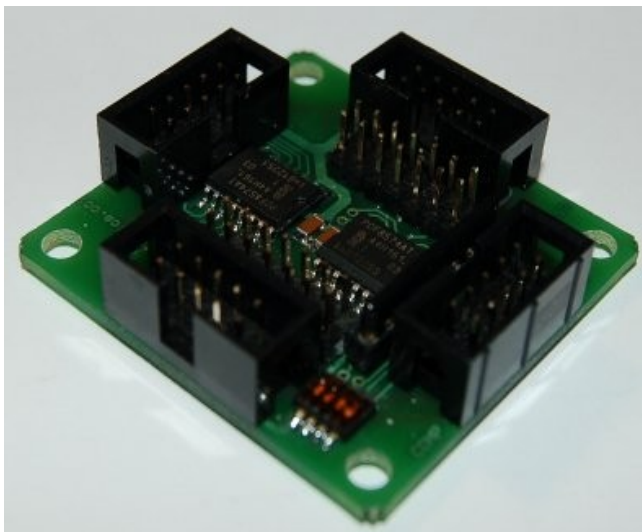
Arduino i Raspberry Pi

I2C



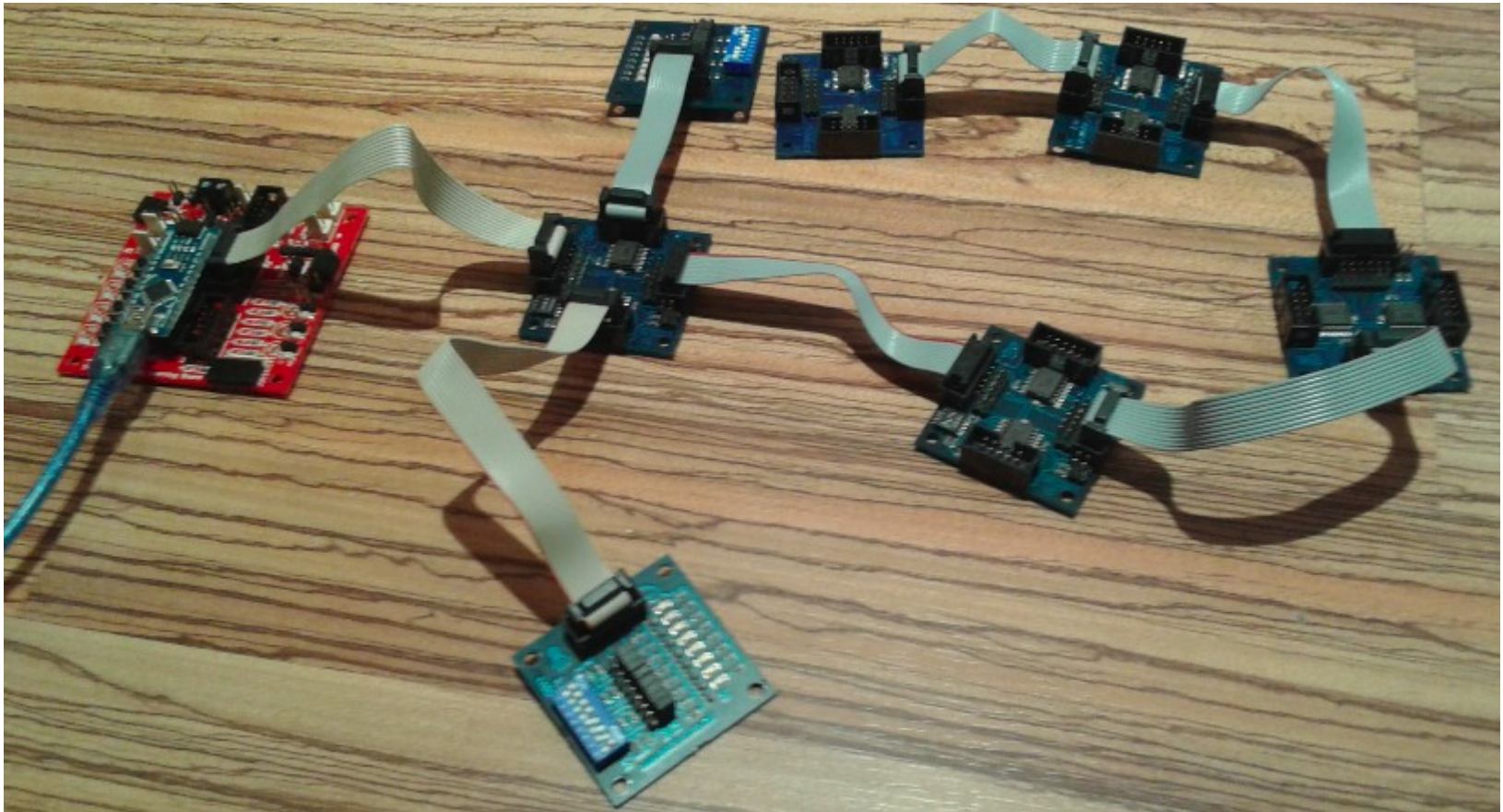
Arduino i Raspberry Pi

I2C



Arduino i Raspberry Pi

I2C



Codis I2C



Arduino i Raspberry Pi

I2C

I2cScanner.ino

```
void loop()
{
  byte error, address;
  int nDevices;

  Serial.println("Scanning...");

  nDevices = 0;
  for(address = 1; address < 127; address++ )
  {
    // The i2c_scanner uses the return value of
    // the Write.endTransmission to see if
    // a device did acknowledge to the address.
    Wire.beginTransmission(address);
    error = Wire.endTransmission();

    if (error == 0)
    {
      Serial.print("I2C device found at address 0x");
      if (address<16)
        Serial.print("0");
      Serial.print(address,HEX);
      Serial.println(" !");

      nDevices++;
    }
  }
}
```



Arduino i Raspberry Pi

I2C

I2cScanner.ino

```
/dev/ttyUSB0
done
Scanning...
I2C device found at address 0x20 !
I2C device found at address 0x21 !
I2C device found at address 0x22 !
I2C device found at address 0x23 !
I2C device found at address 0x24 !
I2C device found at address 0x25 !
I2C device found at address 0x26 !
I2C device found at address 0x27 !
I2C device found at address 0x38 !
I2C device found at address 0x39 !
done
```

☒ Desplaçament automàtic Ambdós NL & CR 9600 baud



Arduino i Raspberry Pi

I2C

[// http://playground.arduino.cc/Code/I2CPortExpander8574](http://playground.arduino.cc/Code/I2CPortExpander8574)

```
#include <Wire.h>
// 8574 Address range is 0x20-0x27
// 8574A Address range is 0x38-0x3F
// 9555 Address range is 0x20-0x27 (same as 8574, bummer)

#define INAddr 0x20
#define OUTAddr 0x21

void setup()
{
  pinMode(12, INPUT); // to read /INT
  pinMode(13, OUTPUT); // to show we are working
  Wire.begin();
  expanderSetInput(INAddr, 0xFF);
}
```

```
void loop(){
  static byte data = 0x01;

  expanderWrite(OUTAddr, (byte)data);
  data <=< 1;
  if(!data)
    data = 0x01;
  delay(200);
}
```

```
void loop(){
  static byte data = 0x01;

  expanderWrite(OUTAddr, data);
  expanderWrite(INAddr, ~data);
  data <=< 1;
  if(!data)
    data = 0x01;
  delay(200);
}
```

```
byte expanderRead(int i2caddr) {
  int _data = -1;
  Wire.requestFrom(i2caddr, 1);
  if(Wire.available()) {
    _data = Wire.read();
  }
  return _data;
}

void expanderWrite(int i2caddr, byte data)
{
  Wire.beginTransmission(i2caddr);
  Wire.write(data);
  Wire.endTransmission();
}
```

```
void loop(){
  byte data = expanderRead(INAddr);

  expanderWrite(OUTAddr, data);
}
```




Arduino i Raspberry Pi

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
```



Arduino i Raspberry Pi

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'



Arduino i Raspberry Pi

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
```



Arduino i Raspberry Pi

Càrrega del protocol Firmata a l'Arduino

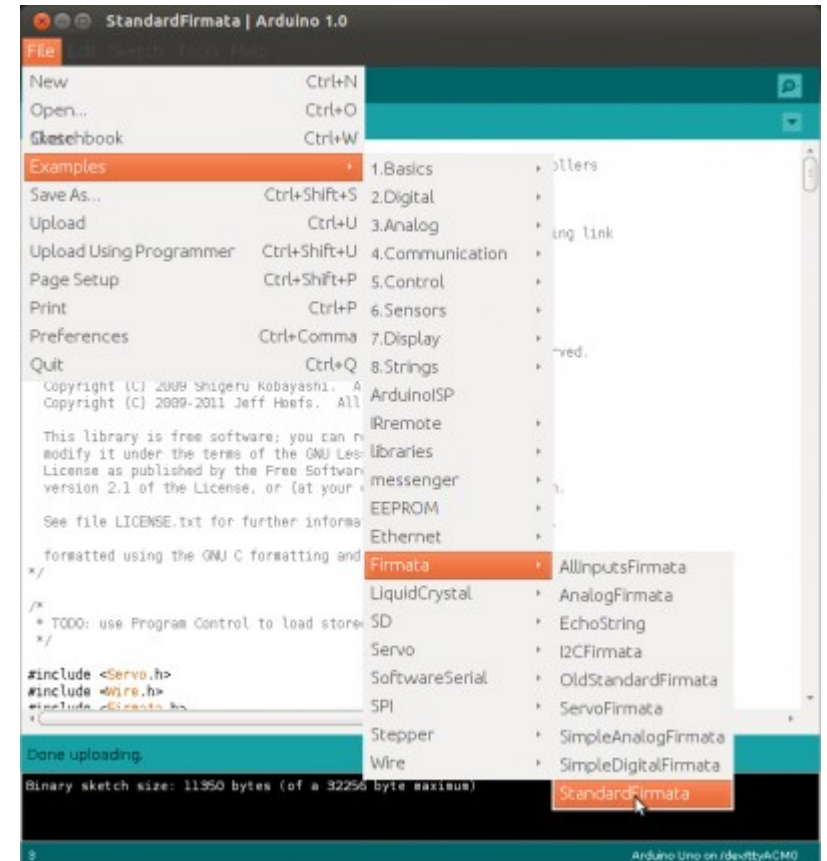
Connecteu la Raspberry Pi a la placa nano-eCat (la que conté l'Arduino Nano) mitjançant el cable USB-A a miniUSB.

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.





Arduino i Raspberry Pi

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

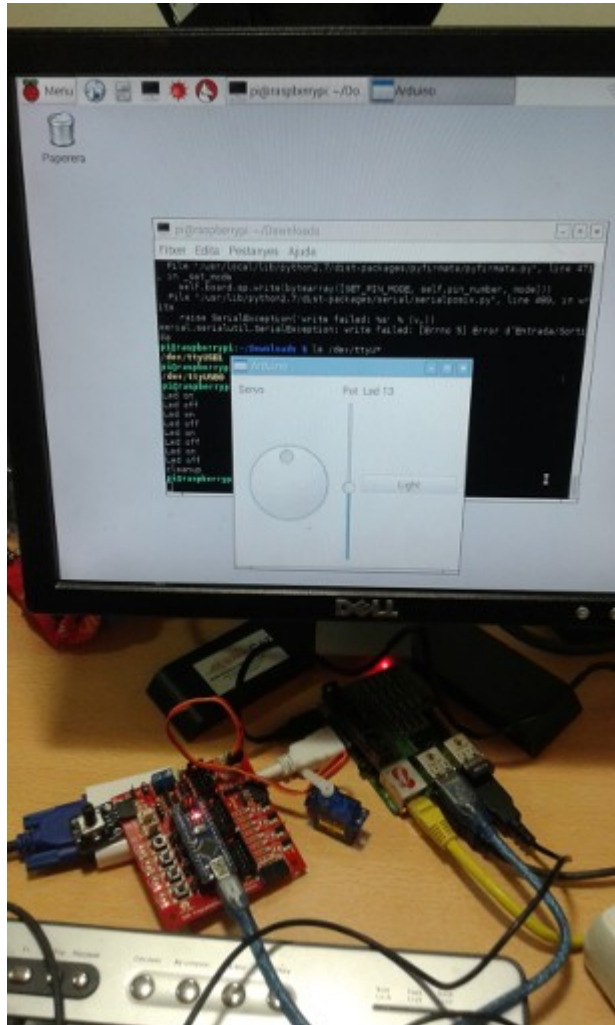
- ☒ Led Bit 7
- ☐ Led Bit 6
- ☒ Led Bit 5
- ☒ Led Bit 4
- ☐ uSw Bit 3
- ☐ uSw Bit 2
- ☐ uSw Bit 1
- ☐ uSw Bit 0

Surt



Arduino i Raspberry Pi

Raspberry Pi amb pyfirmata
i Arduino amb firmata estàndard



Codi per a la Raspberry Pi ([qtgui03.py](#))

Cal tenir instal·lada la biblioteca pyfirmata:

sudo pip install pyfirmata

Més informació : <http://binefa.cat/blog/?p=102>
Vídeo: <https://youtu.be/2kSSPvfXqug>



Arduino i Raspberry Pi

Protocol Firmata



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

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



Arduino i Raspberry Pi

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>



Arduino i Raspberry Pi

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



Arduino i Raspberry Pi

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
```



Arduino i Raspberry Pi

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>



Arduino i Raspberry Pi

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;

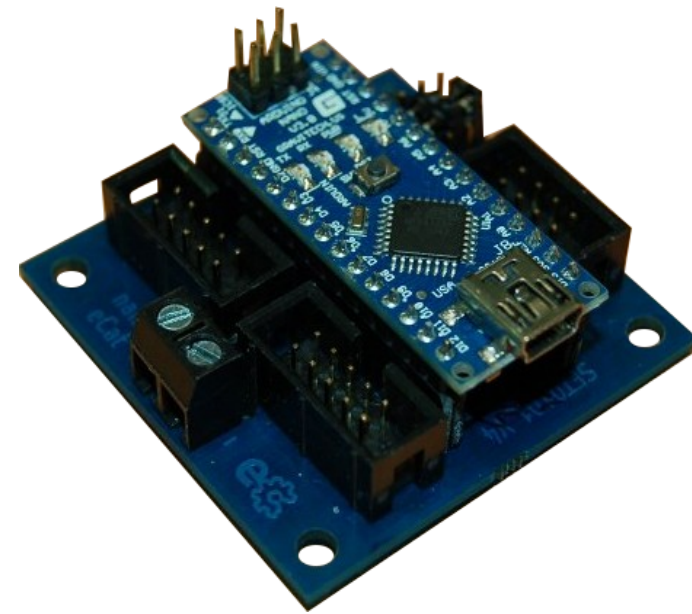
  if (Serial.available()) {
    byteLlegit = Serial.read();
    Serial.write(byteLlegit);
    vIterpretaByteLlegit(byteLlegit);
  }
  if( bCanviP2B0() ){
    delay(100); // Per a evitar rebots
    vTrametValor(ecat.nPinP2B0);
  }
}
```

eCat Maquinari lliure amb Arduino Nano

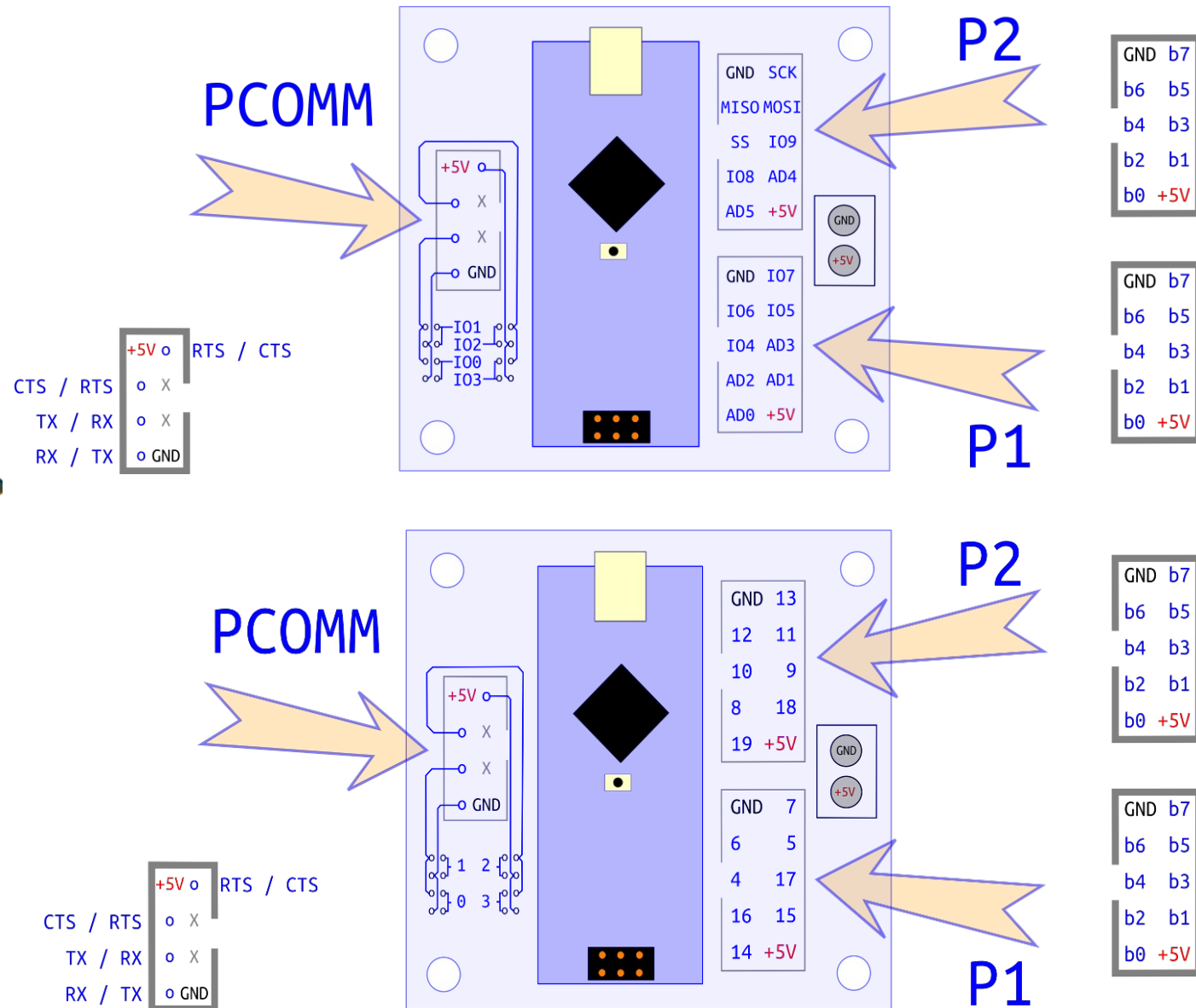
Plaques amb sistema de bus lliure eCat

nano-eCat

Placa 0101 v4



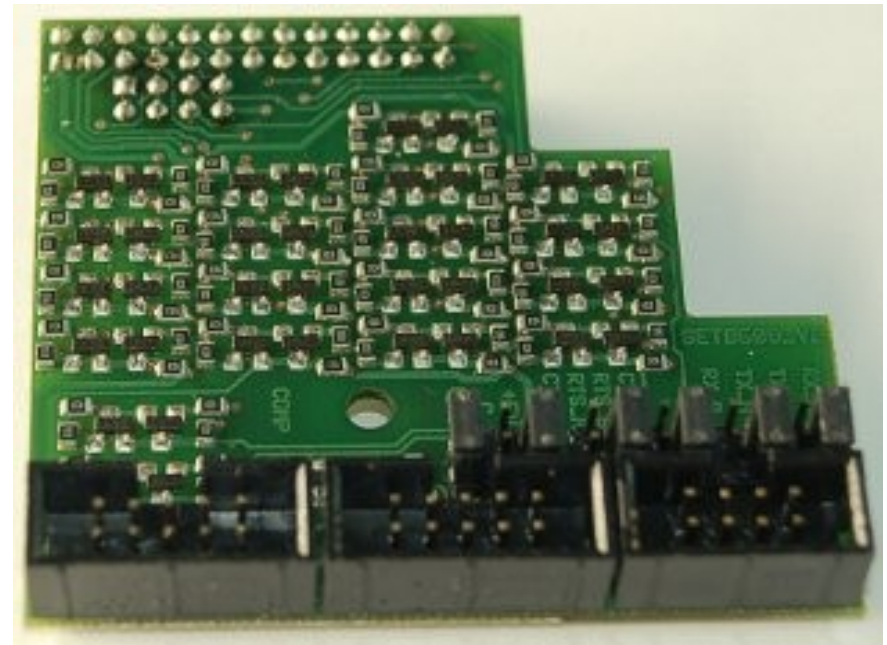
Placa convertidora de l'Arduino Nano al sistema de bus lliure eCat System





Maquinari

3Bpi - Pi de les Tres Branques Placa traductora de nivells per a Raspberry Pi Placa set05_08_v2





Arduino i Raspberry Pi

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

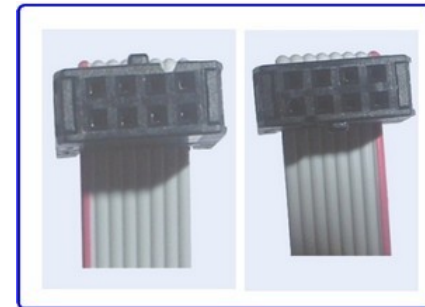
TX / RX

RX / TX

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

CTS / RTS

Cable pla
4x2 a 4x2
(0311)

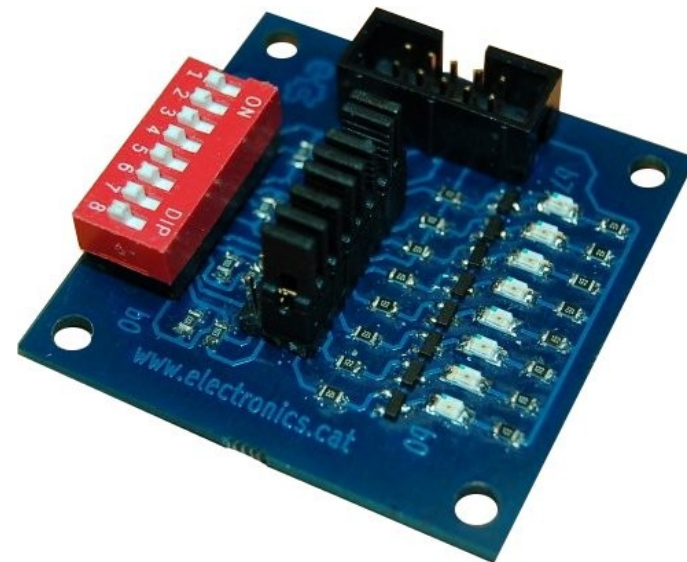
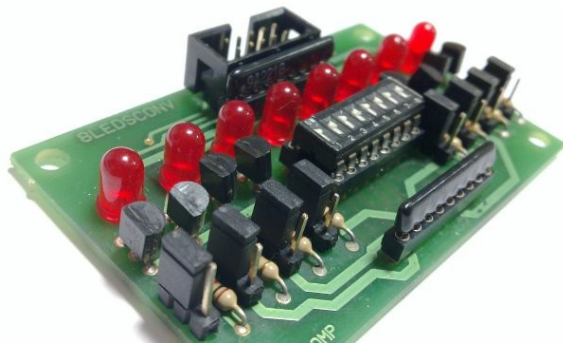
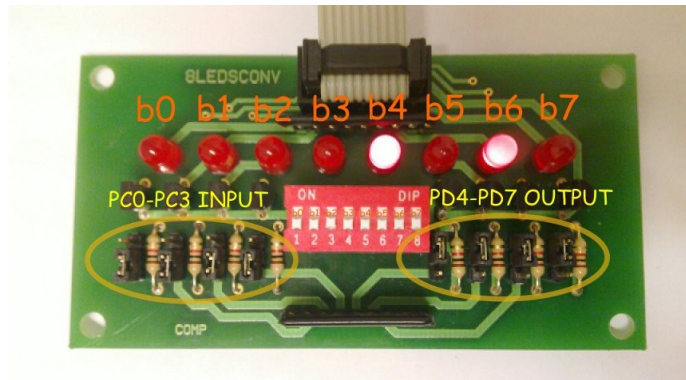
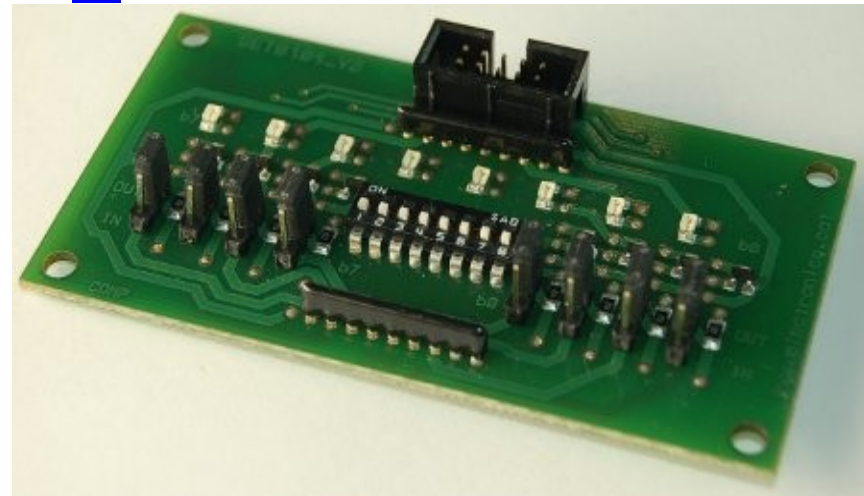
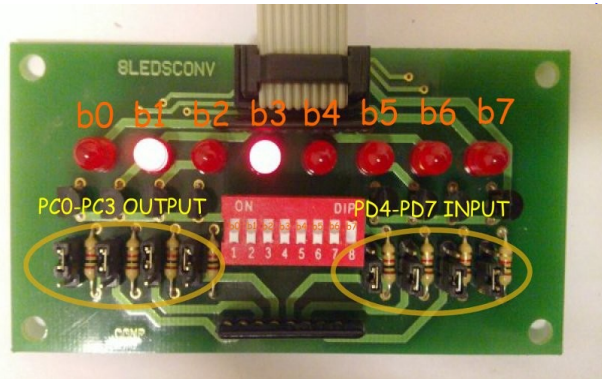


eCat System : Dues tipologies de BUS, dades (8 bits) i comunicacions



Arduino i Raspberry Pi

Leds i microrruptors - Placa set01_04





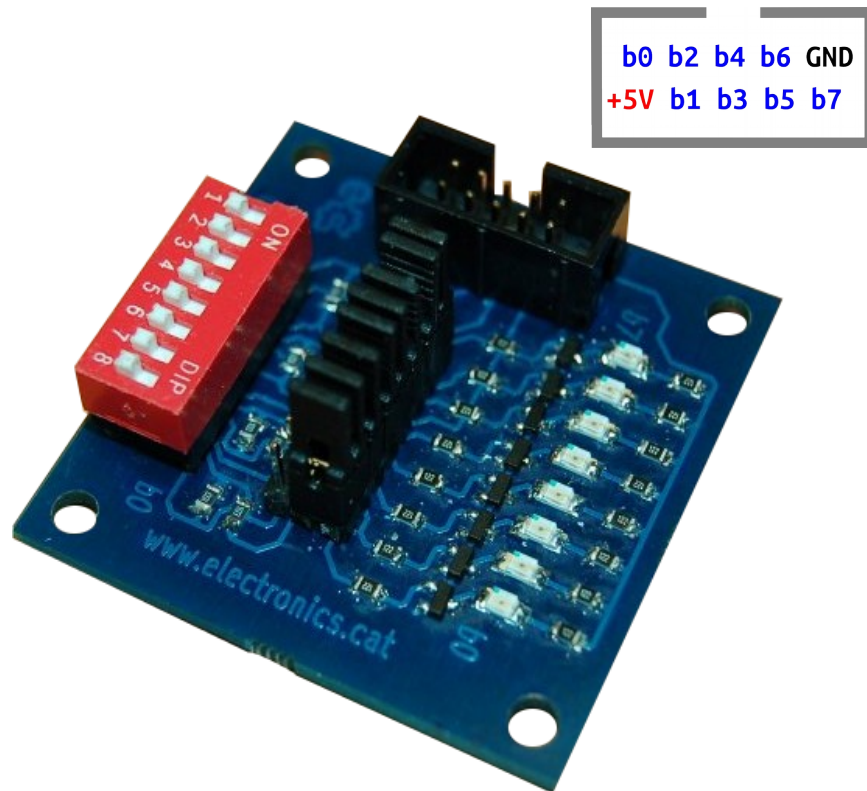
Arduino i Raspberry Pi

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).

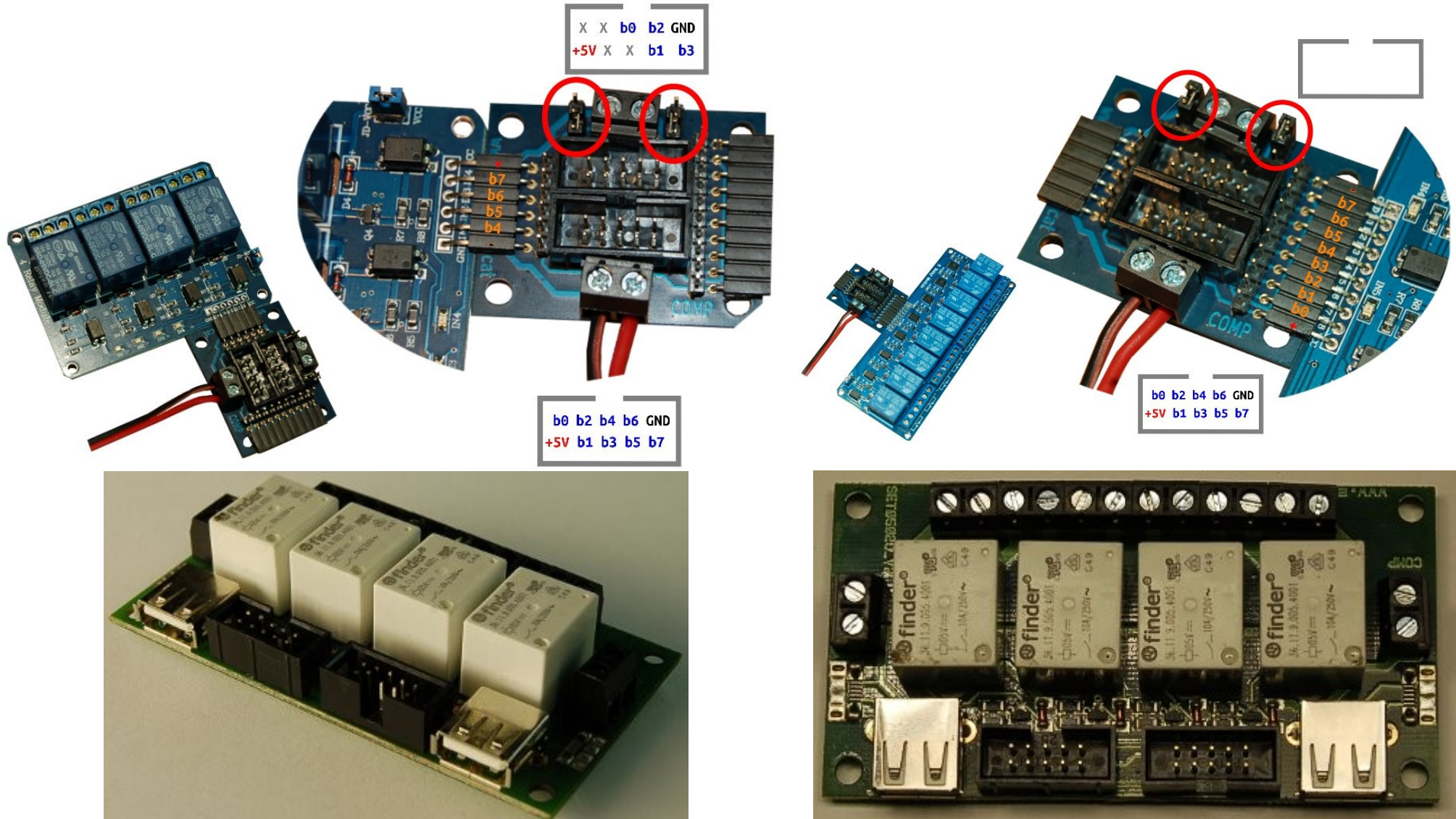
Les entrades es fan per microrruptor (ON : zero lògic, OFF : u lògic).

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



Maquinari

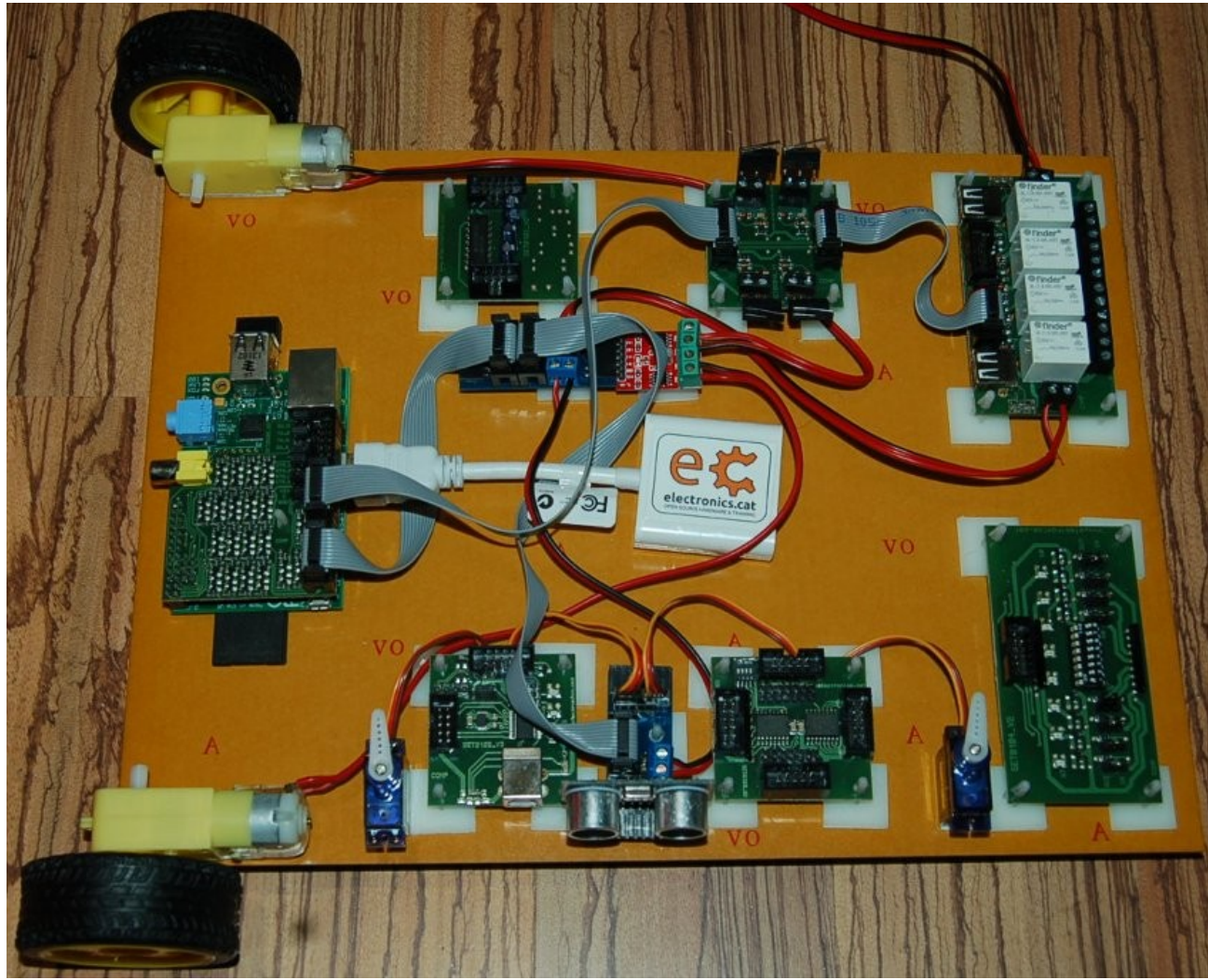
4 Relés - Placa set05_02



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



Arduino i Raspberry Pi



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

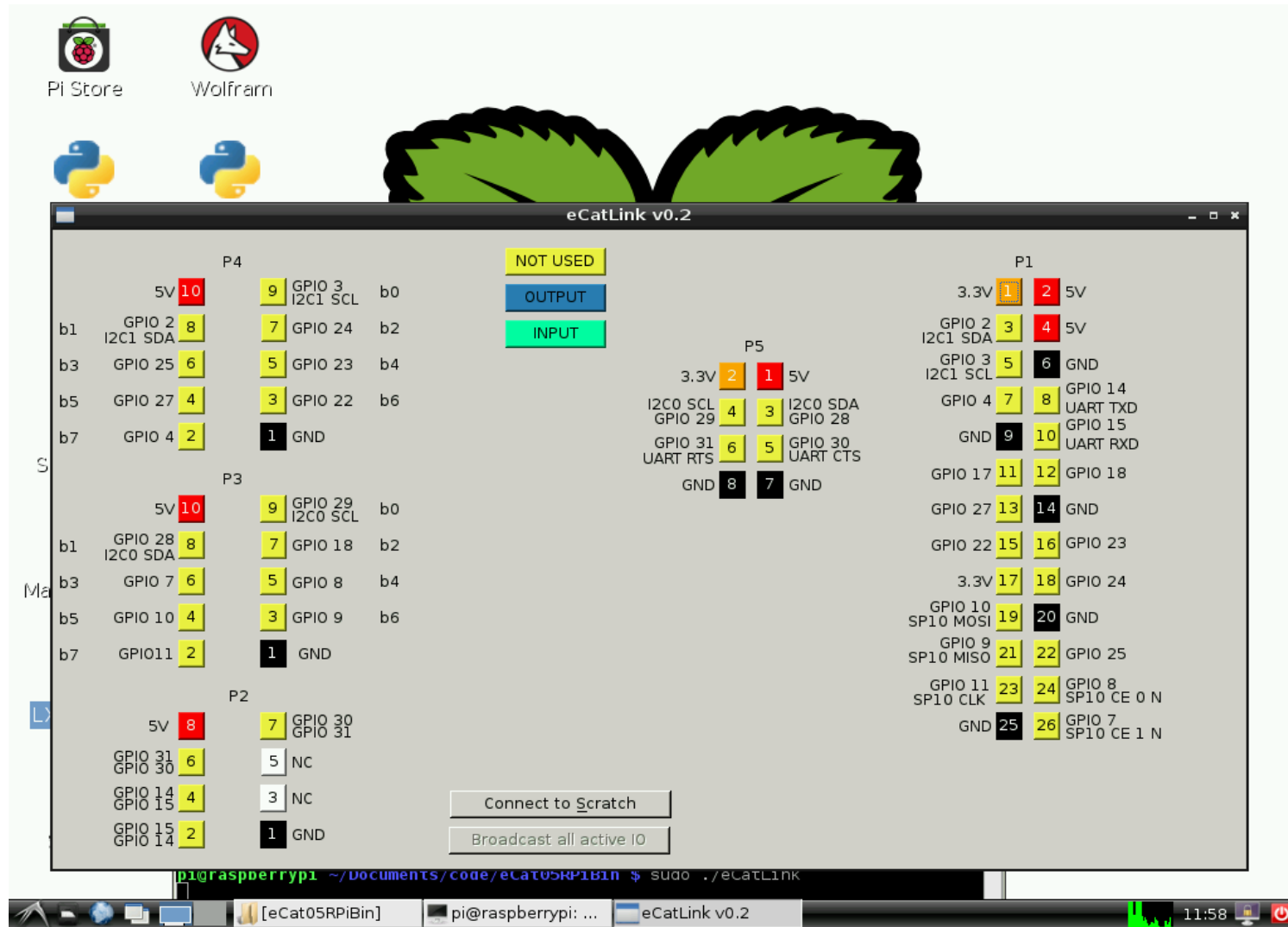
Accès a la GPIO emprant Qt



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

Accès a la GPIO emprant Qt eCatLink

76



<http://www.binefa.cat/php/raspberryPi/qt/eCat05b.tar.gz>

Accès a la GPIO emprant Qt

77

eCatLink

eCatLink v0.2

P4

5V	10	<input checked="" type="checkbox"/>	9	GPIO 3 I2C1 SCL	b0
GPIO 2 I2C1 SDA	8	<input checked="" type="checkbox"/>	7	GPIO 24	b2
GPIO 25	6	<input checked="" type="checkbox"/>	5	GPIO 23	b4
GPIO 27	4	<input type="checkbox"/>	3	GPIO 22	b6
GPIO 4	2	<input type="checkbox"/>	1	GND	b7

P3

5V	10	<input type="checkbox"/>	9	GPIO 29 I2C0 SCL	b0
GPIO 28 I2C0 SDA	8	<input type="checkbox"/>	7	GPIO 18	b2
GPIO 7	6	<input type="checkbox"/>	5	GPIO 8	b4
GPIO 10	4	<input type="checkbox"/>	3	GPIO 9	b6
GPIO11	2	<input type="checkbox"/>	1	GND	b7

P2

5V	8	7	GPIO 30 GPIO 31
GPIO 31 GPIO 30	6	5	NC
GPIO 14 GPIO 15	4	3	NC
GPIO 15 GPIO 14	2	1	GND

P5

3.3V	2	1	5V
I2C0 SCL GPIO 29	4	3	I2C0 SDA GPIO 28
GPIO 31 UART RTS	6	5	GPIO 30 UART CTS
GND	8	7	GND

P1

3.3V	1	2	5V
GPIO 2 I2C1 SDA	3	4	5V
GPIO 3 I2C1 SCL	5	6	GND
GPIO 4	7	8	GPIO 14 UART TXD
GND	9	10	GPIO 15 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 SPI0 MOSI	19	20	GND
GPIO 9 SPI0 MISO	21	22	GPIO 25
GPIO 11 SPI0 CLK	23	24	GPIO 8 SPI0 CE 0 N
GND	25	26	GPIO 7 SPI0 CE 1 N

NOT USED

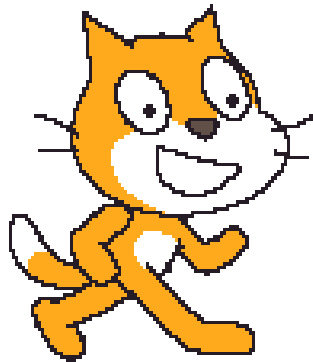
OUTPUT

INPUT

Connect to Scratch

Broadcast all active IO

SCRATCH

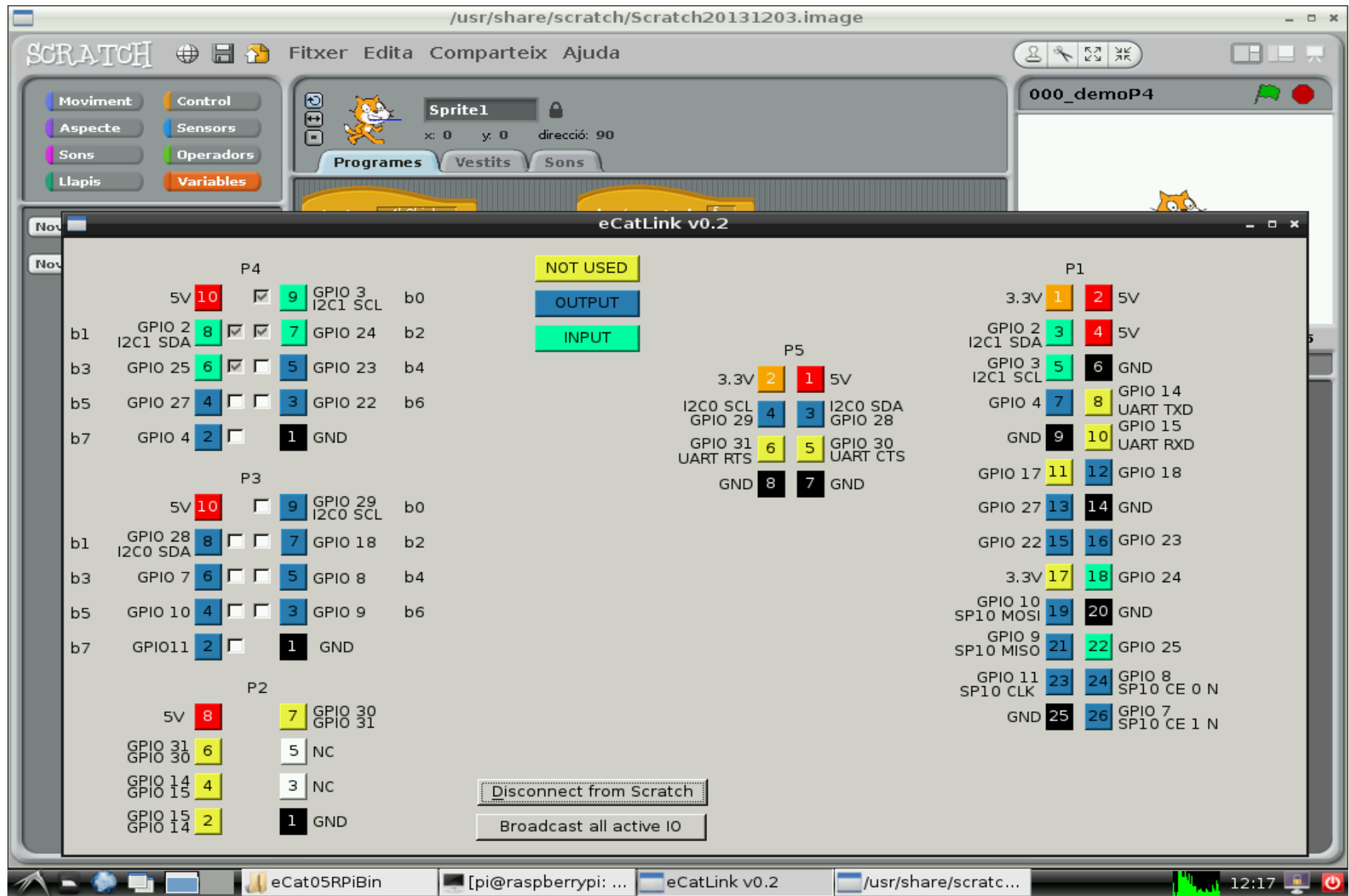


+



Maquinari

Fent ús de P4 - eCatLink



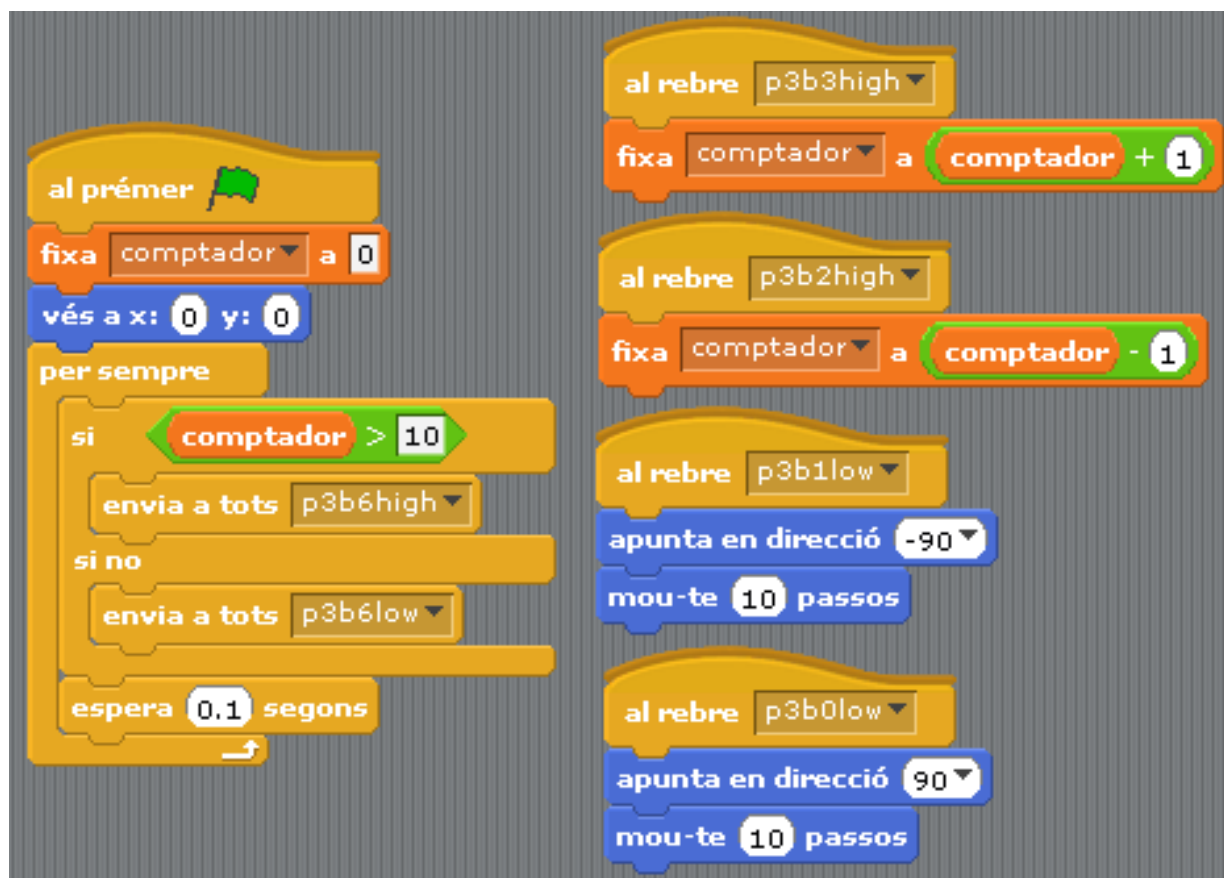
Maquinari

eCatLink + Scratch



Maquinari

eCatLink + Scratch



http://www.binefa.cat/php/raspberryPi/scratch/002_cmpt.sb

Maquinari

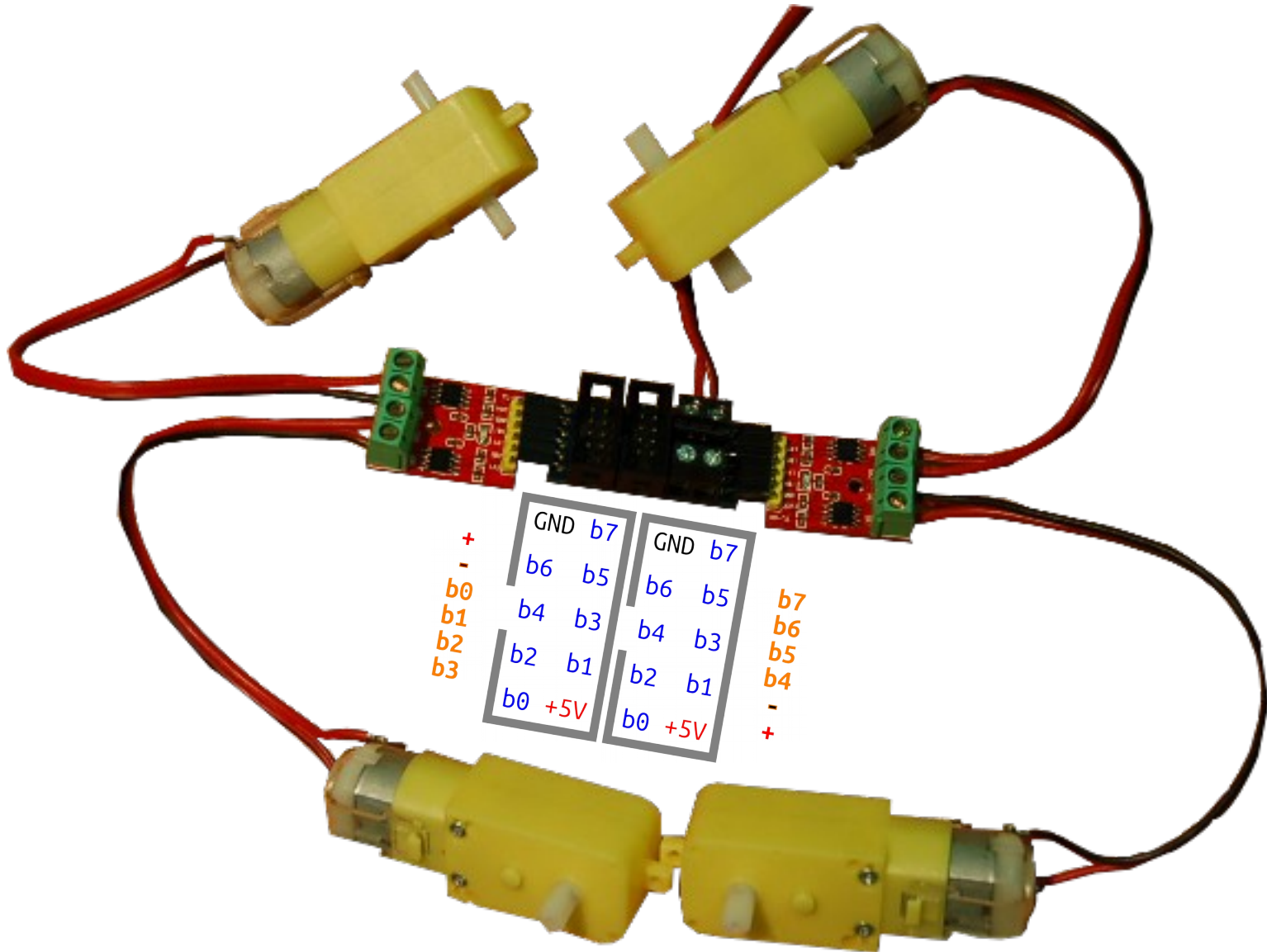
eCatLink + Scratch





Maquinari

Motors emprant mòdul en H



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

87

```
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 - p4b2 i n4b3

89

```
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(24, GPIO.OUT)
11 GPIO.setup(25, GPIO.OUT)
12 p = GPIO.PWM(24,50)      # 50Hz
13 q = GPIO.PWM(25,50)      # 50Hz
14 p.start(7.5)
15 q.start(7.5)
16 try:
17     while True:
18         p.ChangeDutyCycle(7.5) # Neutral
19         q.ChangeDutyCycle(7.5) # Neutral
20         sleep(1)
21     # p.ChangeDutyCycle(12.5) # 180 graus
22     p.ChangeDutyCycle(10)     # 180 graus
23     q.ChangeDutyCycle(10)     # 180 graus
24     sleep(1)
25     # p.ChangeDutyCycle(2.5) # 0 graus
26     p.ChangeDutyCycle(4)      # 0 graus
27     q.ChangeDutyCycle(4)      # 0 graus
28     sleep(1)
29 except KeyboardInterrupt:
30     GPIO.cleanup()
```

<http://www.binefa.cat/php/raspberryPi/python/pyServos.py>

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

Lectura de sensor d'ultrasons



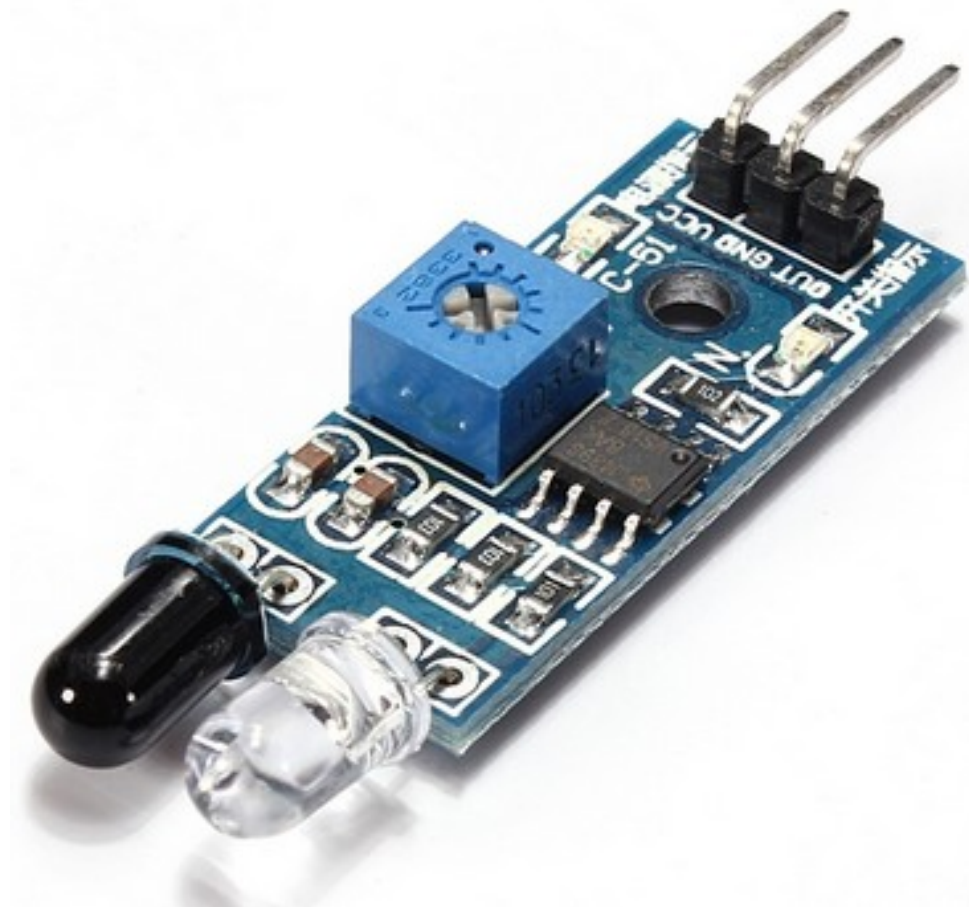
Lectura de sensor d'ultrasons

```
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
```

<http://www.binefa.cat/php/raspberryPi/python/pyUltrasons.py>

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

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>

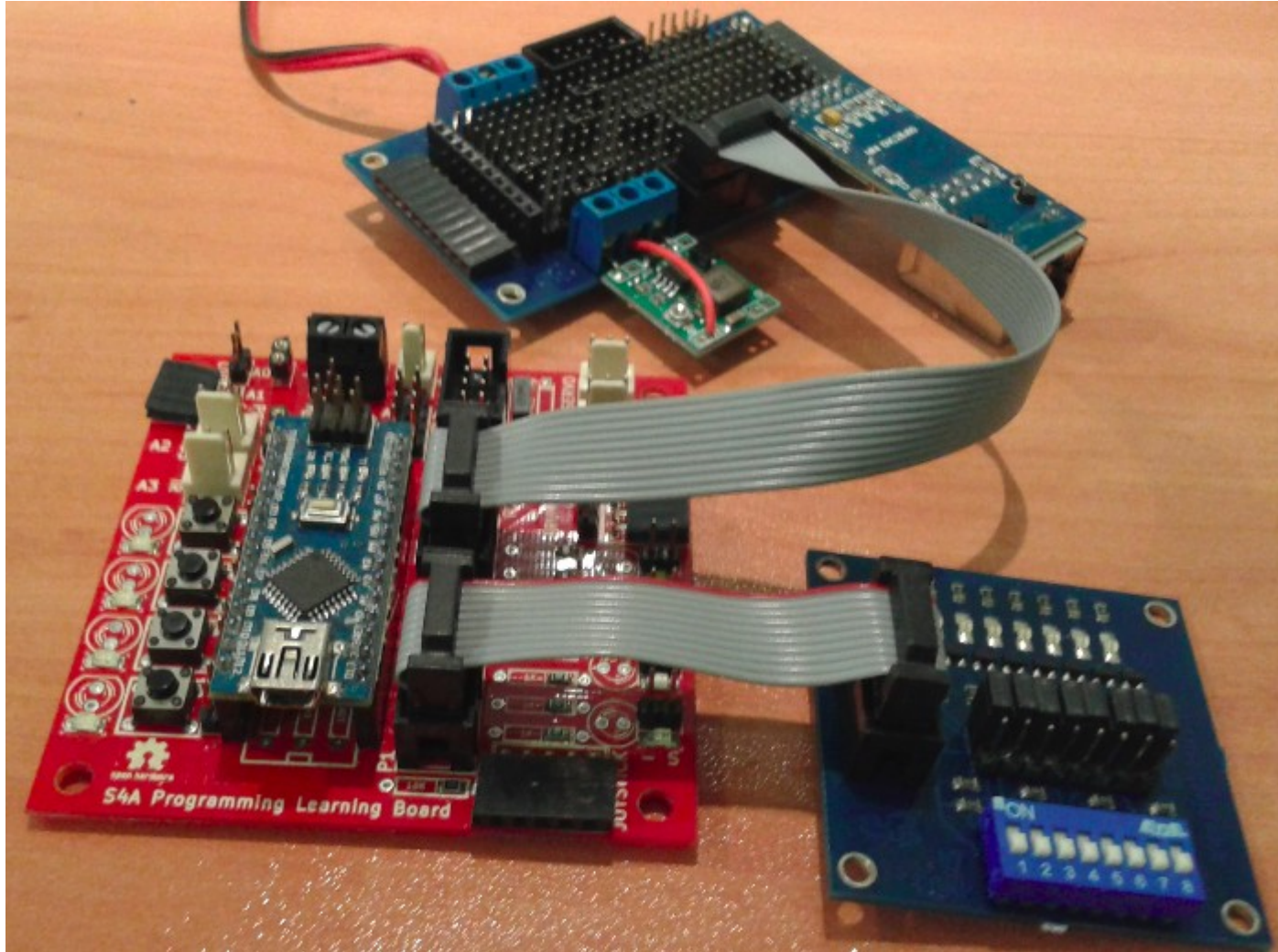
<https://dl.dropboxusercontent.com/u/65254823/oshw20140319/py04b.py>

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



Arduino i Raspberry Pi

Ethernet



Codi per a Arduino UdpServer01b



Arduino i Raspberry Pi

Ethernet – Mòdul ENC28J60

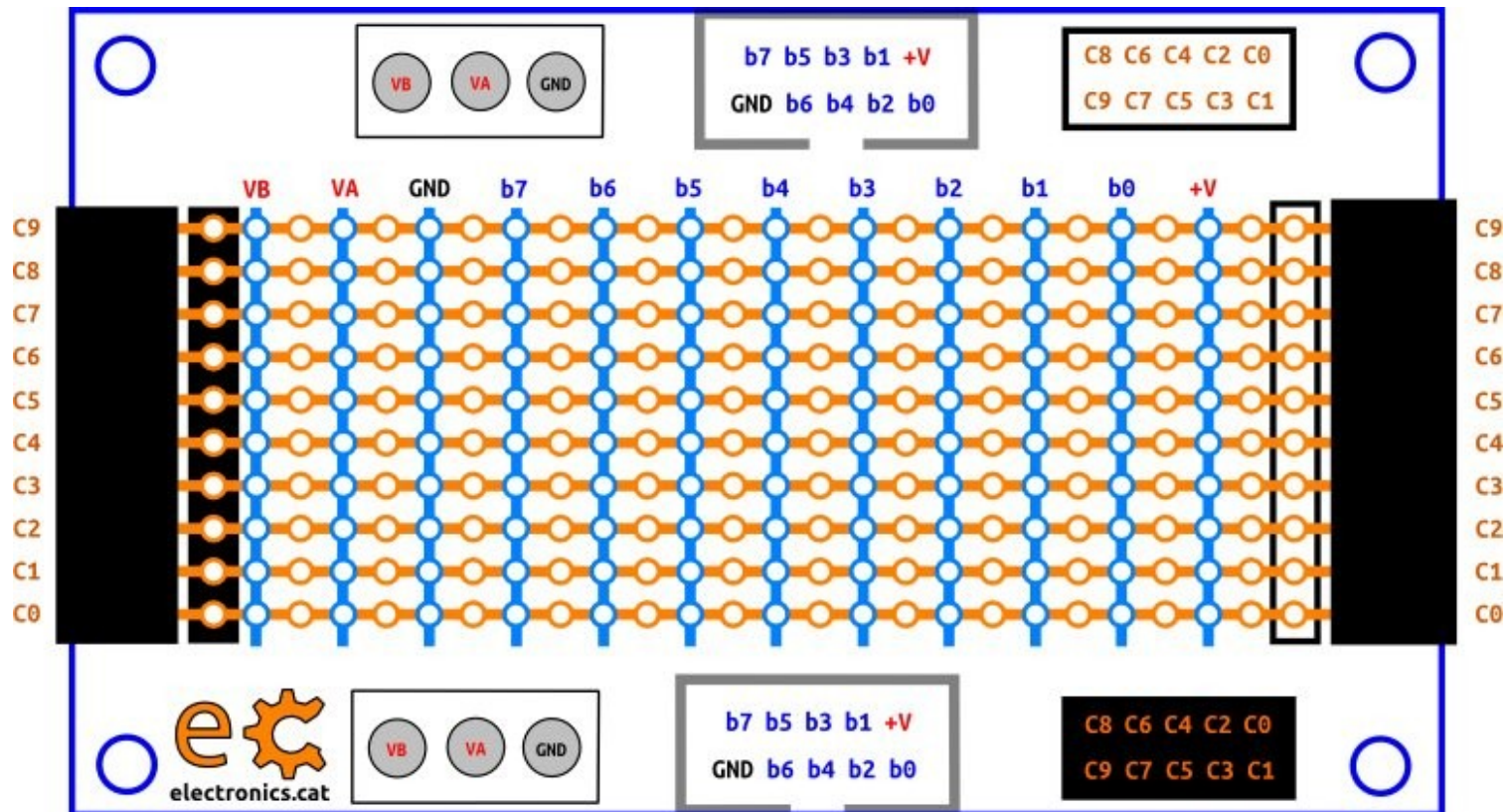


Biblioteca UIPEthernet per Arduino i ENC28J60



Arduino i Raspberry Pi

Matriu

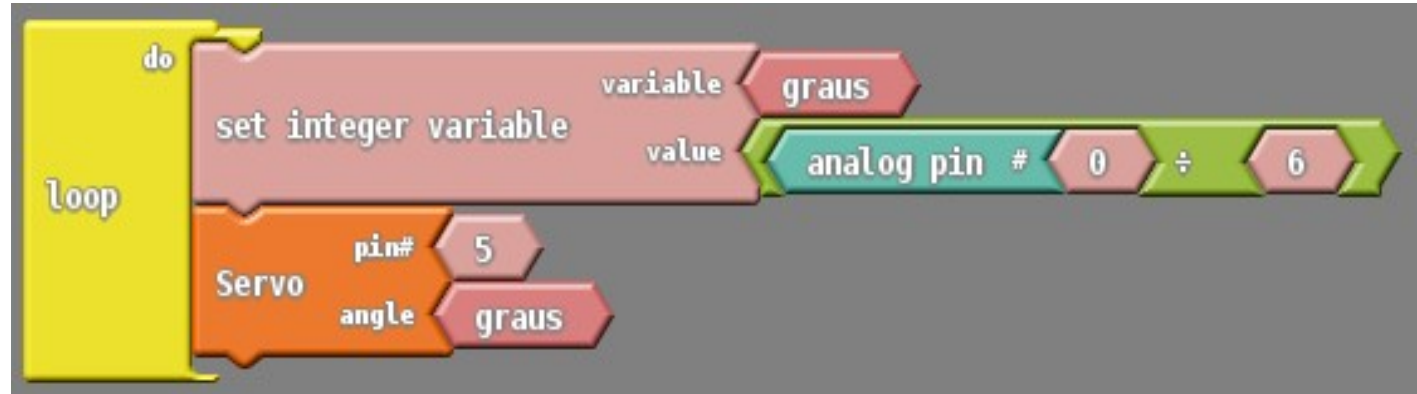
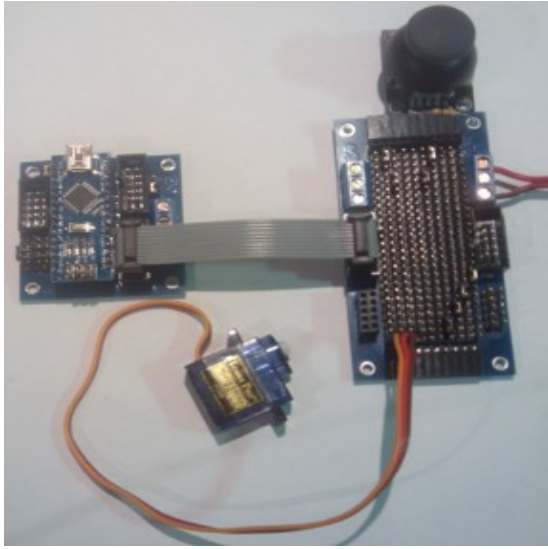


<http://electronics.cat/php/common/index.php?lang=ca&page=517>

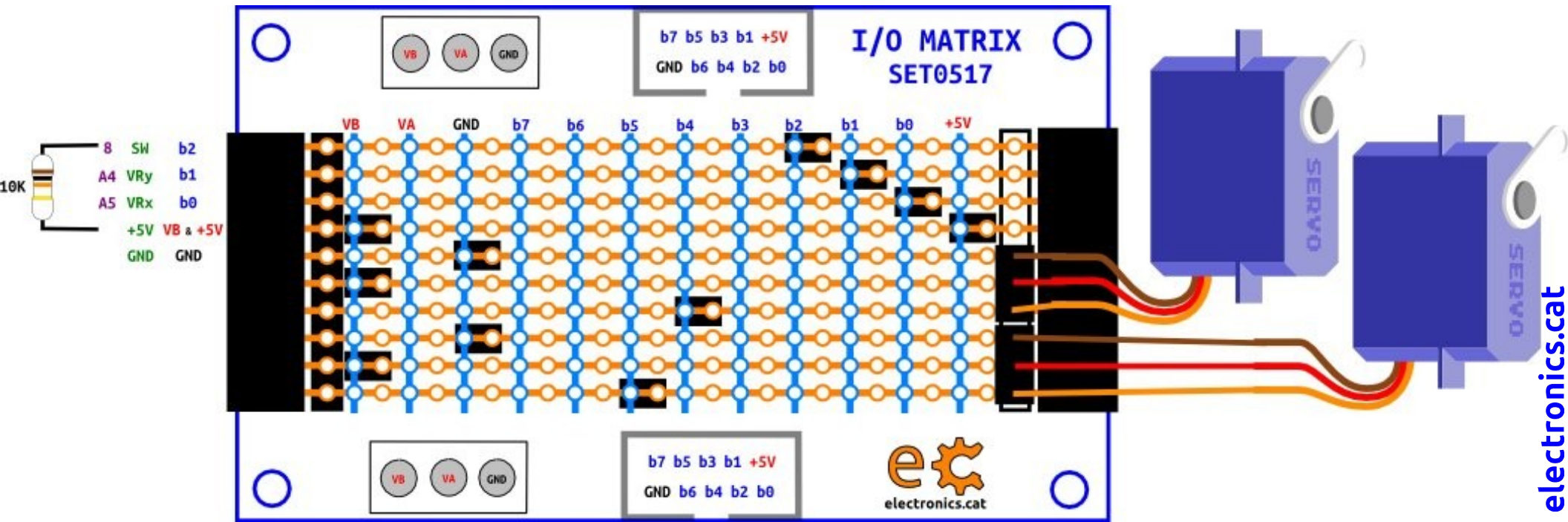


Arduino i Raspberry Pi

Exemple d'ús de la matriu



<http://www.binefa.cat/php/arduino/ardublock/pr04joystickServo.abp>





Arduino i Raspberry Pi

Matriu

Connexió a la matriu inversa (pàgina 100 del pdf)

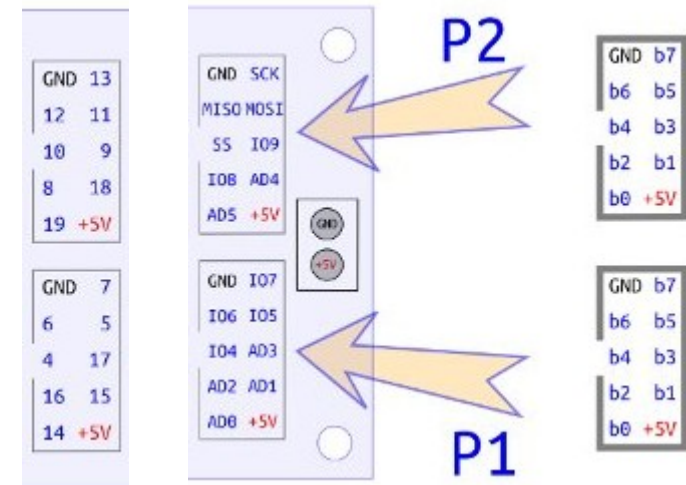
C8	C6	c4	C2	C0
C9	C7	C5	C3	C1

Connexió de l'ENC28J60

CLK	WOL	SI	CS	VCC
INT	S0	SCK	RST	GND

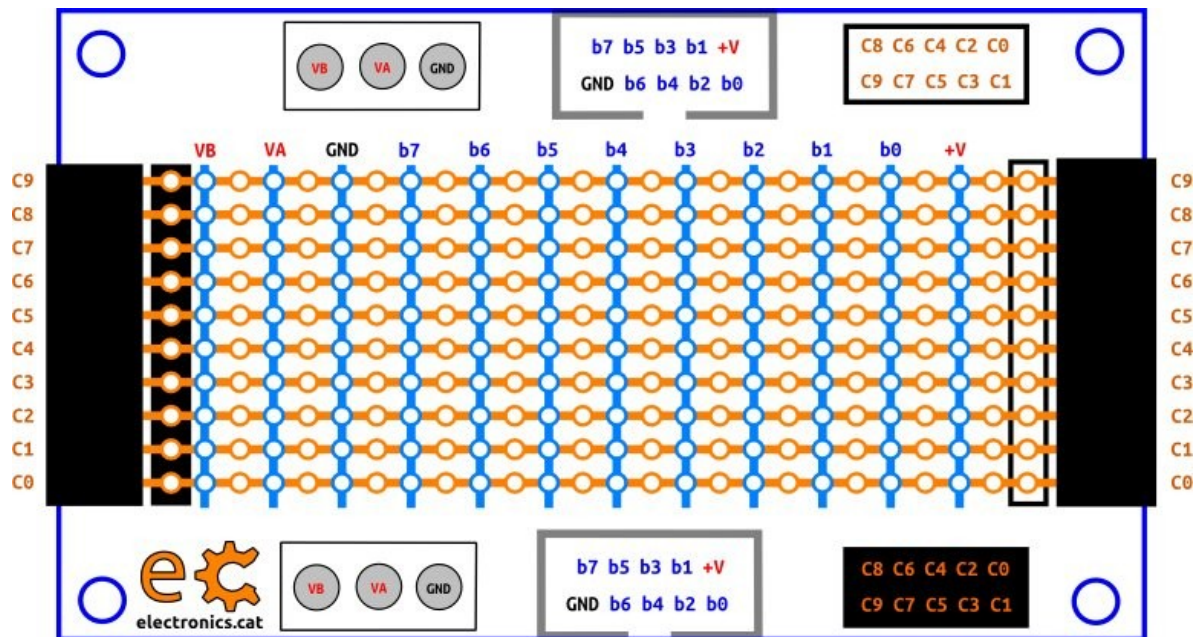
Connexió bus ecat

N	NC	b5	b4	3V3
NC	b6	b7	+5V	GND



Per pota de connector ecat

1	CK	NC
2	INT	NC
3	WOL	NC
4	S0	b6
5	SI	b5
6	SCK	b7
7	CS	b4
8	RST	+5V
9	VCC	3V3
10	GND	





Arduino i Raspberry Pi

Ethernet – Recepció de datagrames UDP

```
#include <UIPEthernet.h>
#include <ecat.h>

EthernetUDP udp;
Ecat ecat;

void setup() {
  Serial.begin(9600);

  uint8_t mac[6] = {0x00,0x01,0x02,0x03,0x04,0x05};

  Ethernet.begin(mac,IPAddress(192,168,1,108));
  //Ethernet.begin(mac,IPAddress(172,20,1,168));

  int success = udp.begin(5000);

  Serial.print("initialize: ");
  Serial.println(success ? "success" : "failed");
  ecat.setupHighNibbleP1(OUTPUT);
}
```

```
void loop() {
  //check for new udp-packet:
  int size = udp.parsePacket();
  if (size > 0) {
    do
    {
      char* msg = (char*)malloc(size+1);
      int len = udp.read(msg,size+1);
      msg[len]=0;
      Serial.print("received: ");
      Serial.print(msg);
      if(msg[0]=='0'){
        digitalWrite(ecat.nPinP1B7, LOW);
      }else{
        if(msg[0]=='1'){
          digitalWrite(ecat.nPinP1B7, HIGH);
        }
      }
      free(msg);
    }
    while ((size = udp.available())>0);
    //finish reading this packet:
    udp.flush();
    Serial.println("");
  }
```




Arduino i Raspberry Pi

Ethernet – Enviament de datagrames UDP

```

int success;
do
{
    Serial.print("remote ip: ");
    Serial.println(udp.remoteIP());
    Serial.print("remote port: ");
    Serial.println(udp.remotePort());
    //send new packet back to ip/port of client. This also
    //configures the current connection to ignore packets from
    //other clients!
    success = udp.beginPacket(udp.remoteIP(),udp.remotePort());
    Serial.print("beginPacket: ");
    Serial.println(success ? "success" : "failed");
    //beginPacket fails if remote ethaddr is unknown. In this case an
    //arp-request is send out first and beginPacket succeeds as soon
    //the arp-response is received.
}
while (!success);

success = udp.println("hello world from arduino");

Serial.print("bytes written: ");
Serial.println(success);

success = udp.endPacket();

Serial.print("endPacket: ");
Serial.println(success ? "success" : "failed");

udp.stop();
//restart with new connection to receive packets from other clients
Serial.print("restart connection: ");
Serial.println (udp.begin(5000) ? "success" : "failed");
}
}

```

Codi per a Arduino UdpServer01b



Arduino i Raspberry Pi

Ethernet – Enviament de datagrames UDP

```
$ ./udpOn.py 192.168.1.108
```

```
#!/usr/bin/python

import socket
import sys

szServer = str(sys.argv[1])
nUdpPort = 5000

client_socket = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
client_socket.sendto("1", (szServer, nUdpPort))
client_socket.close()
```

```
$ ./udpOff.py 192.168.1.108
```

```
#!/usr/bin/python

import socket
import sys

szServer = str(sys.argv[1])
nUdpPort = 5000

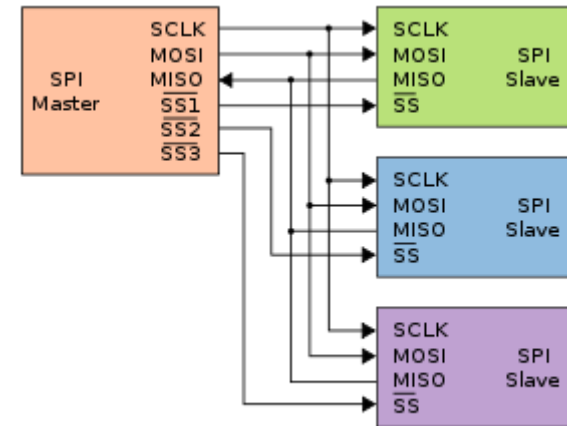
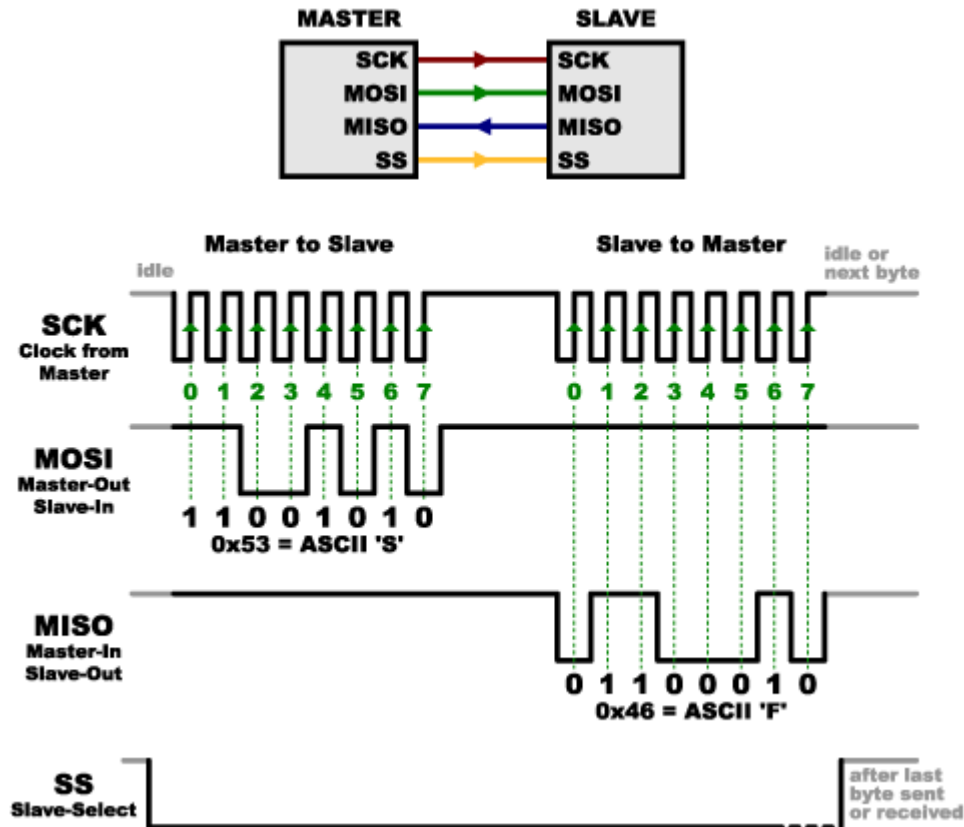
client_socket = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
client_socket.sendto("0", (szServer, nUdpPort))
client_socket.close()
```

Codis font dels codis d'enviament de datagrames UDP en Python



Arduino i Raspberry Pi

SPI

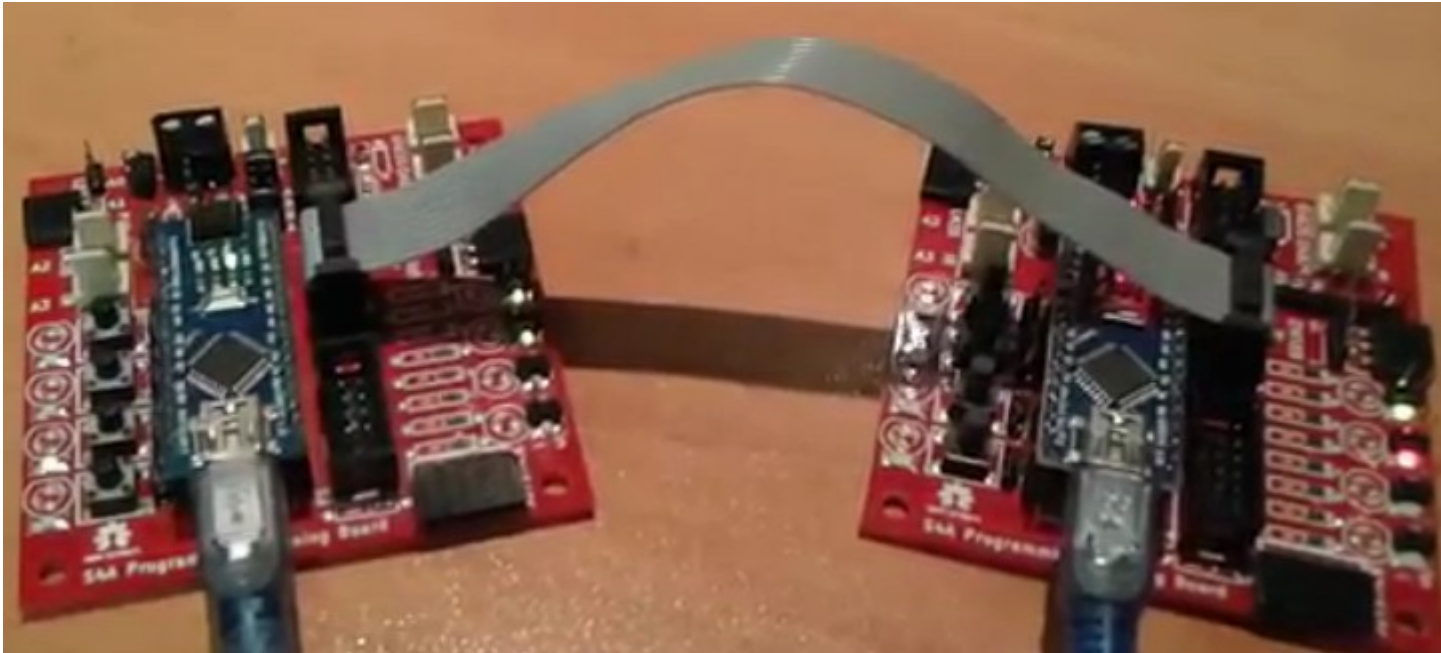


Explicació del bus SPI



Arduino i Raspberry Pi

SPI



Arduino's SPI master code

Arduino's SPI slave code

Vídeo de funcionament



Arduino i Raspberry Pi

SPI

```

Serial.println("Master Initialized");
}
// The loop() function runs continuously after setup
void loop() {
  // Master button pressed?
  if (!digitalRead(btn)) {
    // Yes
    Serial.println("Master Button Pressed.");
    // Select and wait for slave.
    digitalWrite(SS, LOW);
    Serial.println("***Slave Selected.");
    delay(20);
    // Send cmdBtn
    SPI.transfer(cmdBtn);
    Serial.println("cmdBtn Sent.");
    // Wait for slave.
    delay(20);
    // Get slave response.
    byte rx = SPI.transfer(255);
    // Acknowledged?
    if (rx == cmdBtn) {
      // Yes.
      Serial.println("Slave acknowledged cmdBtn.");
    }
    // Command not recognized?
  }
}

```

Arduino's SPI master code

Arduino's SPI slave code

```

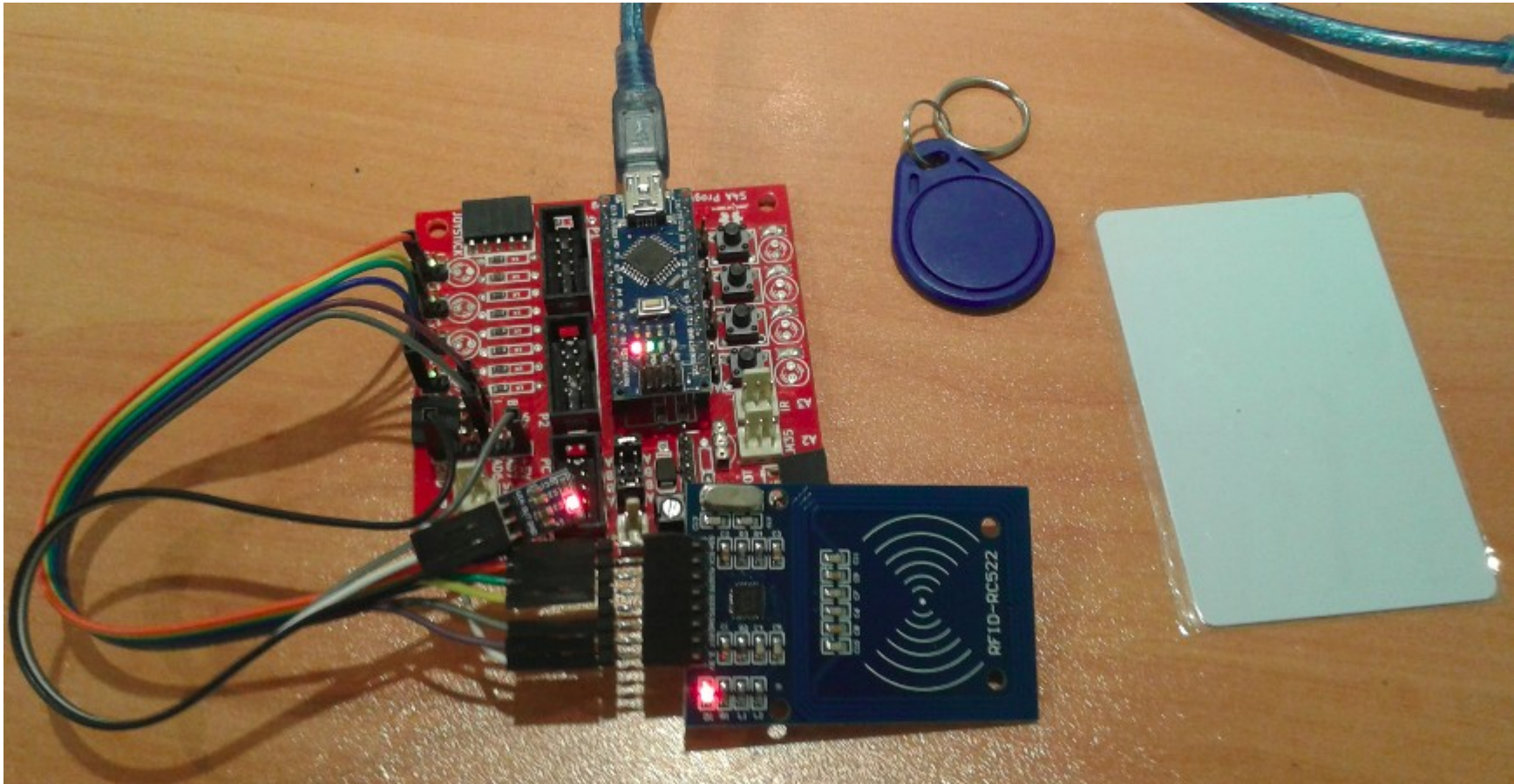
void loop() {
  // Slave Enabled?
  if (!digitalRead(SS)) {
    // Yes, first time?
    if (SSlast != LOW) {
      // Yes, take MISO pin.
      pinMode(MISO, OUTPUT);
      Serial.println("***Slave Enabled.");
      // Write -1 slave response code and receive master command code
      byte rx = SPI.transfer(255);
      Serial.println("Initial -1 slave response code sent");
      Serial.println("rx:" + String(rx) + ".");
      // cmdBtn?
      if (rx == cmdBtn) {
        // Acknowledge cmdBtn.
        byte rx = SPI.transfer(cmdBtn);
        Serial.println("cmdBtn Acknowledged.");
        Serial.println("rx:" + String(rx) + ".");
        // Toggle LED State
        ledState = !ledState;
        digitalWrite(led, ledState);
      }
      // cmdLEDState?
      else if (rx == cmdLEDState) {
        // Acknowledge cmdLEDState.
        byte rx = SPI.transfer(cmdLEDState);
        Serial.println("cmdLEDState Acknowledged.");
        Serial.println("rx:" + String(rx) + ".");
        rx = SPI.transfer(ledState);
        Serial.println("ledState:" + String(ledState) + " Sent.");
        Serial.println("rx:" + String(rx) + ".");
      }
    }
  }
}

```




Arduino i Raspberry Pi

SPI



Com connectar el mòdul RFID-RC522 a Arduino



Arduino i Raspberry Pi

SPI

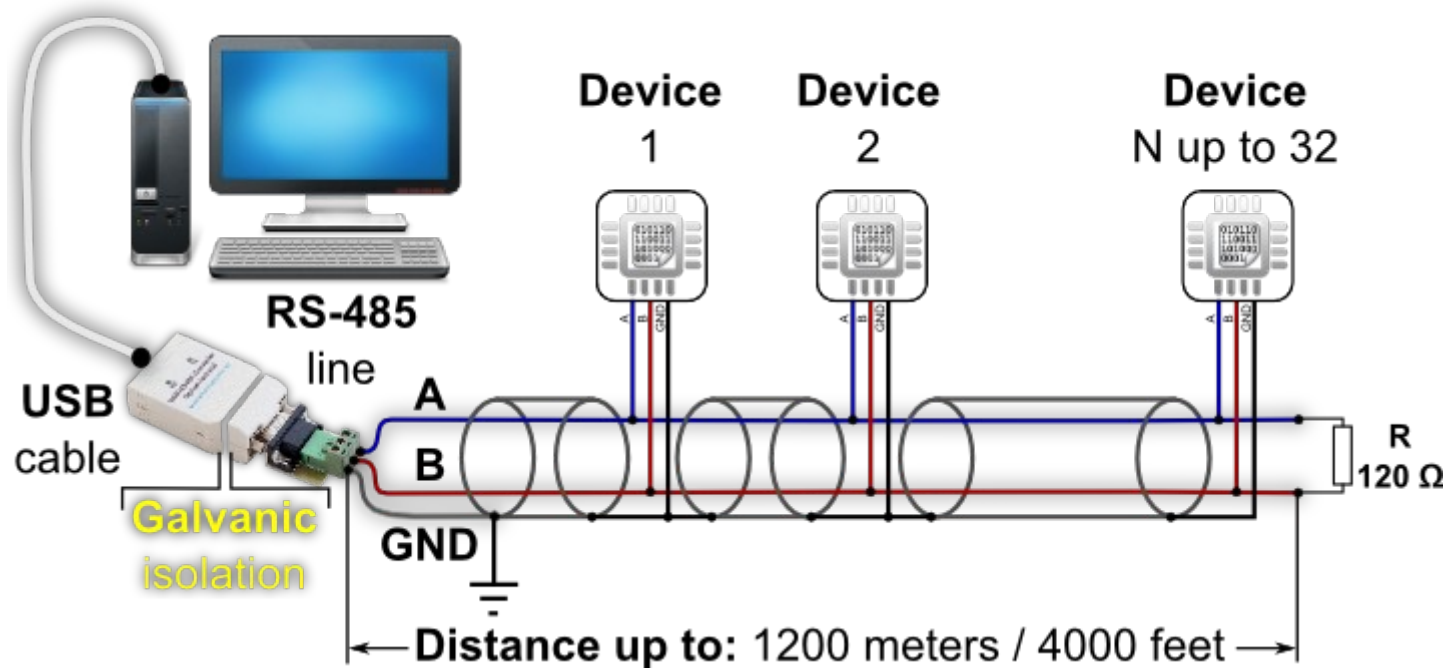
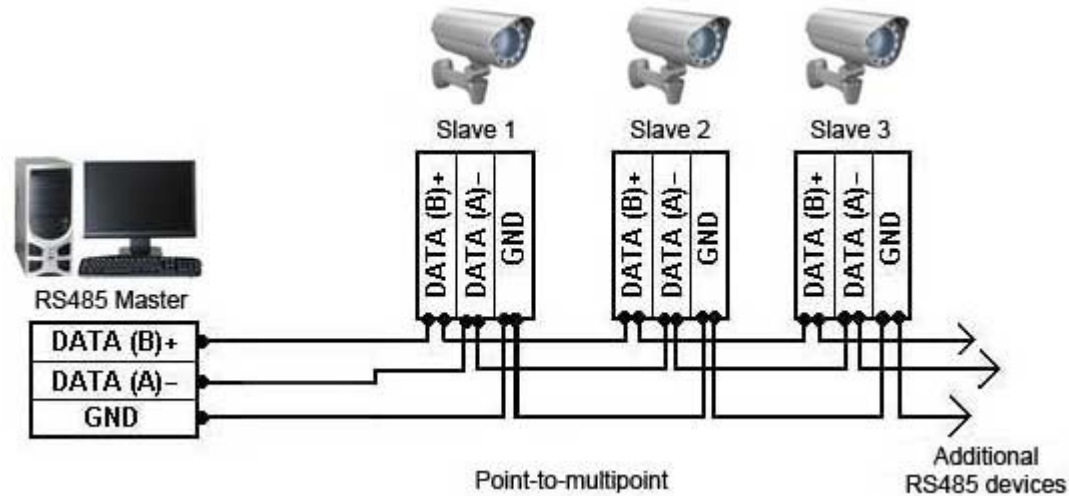
```
/dev/ttyUSB0
| Envia

This code scan the MIFARE Classsic NUID.
Using the following key: FF FF FF FF FF FFPICC type: MIFARE 1KB
A new card has been detected.
The NUID tag is:
In hex: 35 07 10 22
In dec: 53 07 16 34
PICC type: MIFARE 1KB
A new card has been detected.
The NUID tag is:
In hex: F1 45 25 61
In dec: 241 69 37 97

☒ Desplaçament automàtic  Ambdós NL & CR  9600 baud
```

Arduino i Raspberry Pi

RS485



Arduino i Raspberry Pi

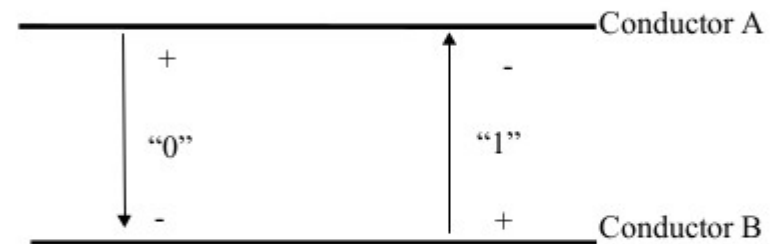
RS422

Comunicación Serie RS-422

Descripción

La RS-422 trabaja en forma diferencial con las líneas que transmite y recibe, el circuito tiene solo dos hilos sin que exista una línea de masa común. Los unos y ceros lógicos se establecen en función de la diferencia de tensión ambos conductores del circuito.

Especificaciones		RS422
Modo de Operación		DIFERENCIAL
Número de dispositivos		1 EMISOR 10 RECEPTORES
Máxima longitud del cable		1200 metros
Máxima velocidad de transmisión		10 Mb/s
Rango de trabajo		+/-10V
Rango de señal	Alto	+/-6V
	Bajo	+/-2V
Sensibilidad de entrada receptor		+/-200mV
Resistencia de entrada receptor		>=4K



$$\text{"0"} \quad 1,5 \text{ V} < \Delta V_{AB} < 5 \text{ V}$$

$$\text{"1"} \quad -1,5 \text{ V} > \Delta V_{AB} > -5 \text{ V}$$



Arduino i Raspberry Pi

RS485

Comunicación Serie RS-485 (I)

Descripción

La RS-485 es una leve modificación de la RS-422, redefiniendo características eléctricas para asegurar un nivel de tensión adecuado a la máxima carga, incrementándose el número de dispositivos.

En una red de dispositivos sobre una simple línea, es necesario direccionar uno en particular. Esto se puede realizar simplemente utilizando caracteres ASCII, constituyendo comandos de identificación del dispositivo y que este a su vez responde con los datos. Esto es un esquema básico de protocolo de comunicación denominado comúnmente maestro/esclavo (Master/Slave).

Especificaciones		RS485
Modo de Operación		DIFERENCIAL
Número de dispositivos		32 EMISORES 32 RECEPTORES
Máxima longitud del cable		1200 metros
Máxima velocidad de transmisión		10 Mb/s
Rango de trabajo		-7V a +12V
Rango de señal	Alto	+/-6V
	Bajo	+/-1,5V
Sensibilidad de entrada receptor		+/-200mV
Resistencia de entrada receptor		>=12K



Arduino i Raspberry Pi

RS485

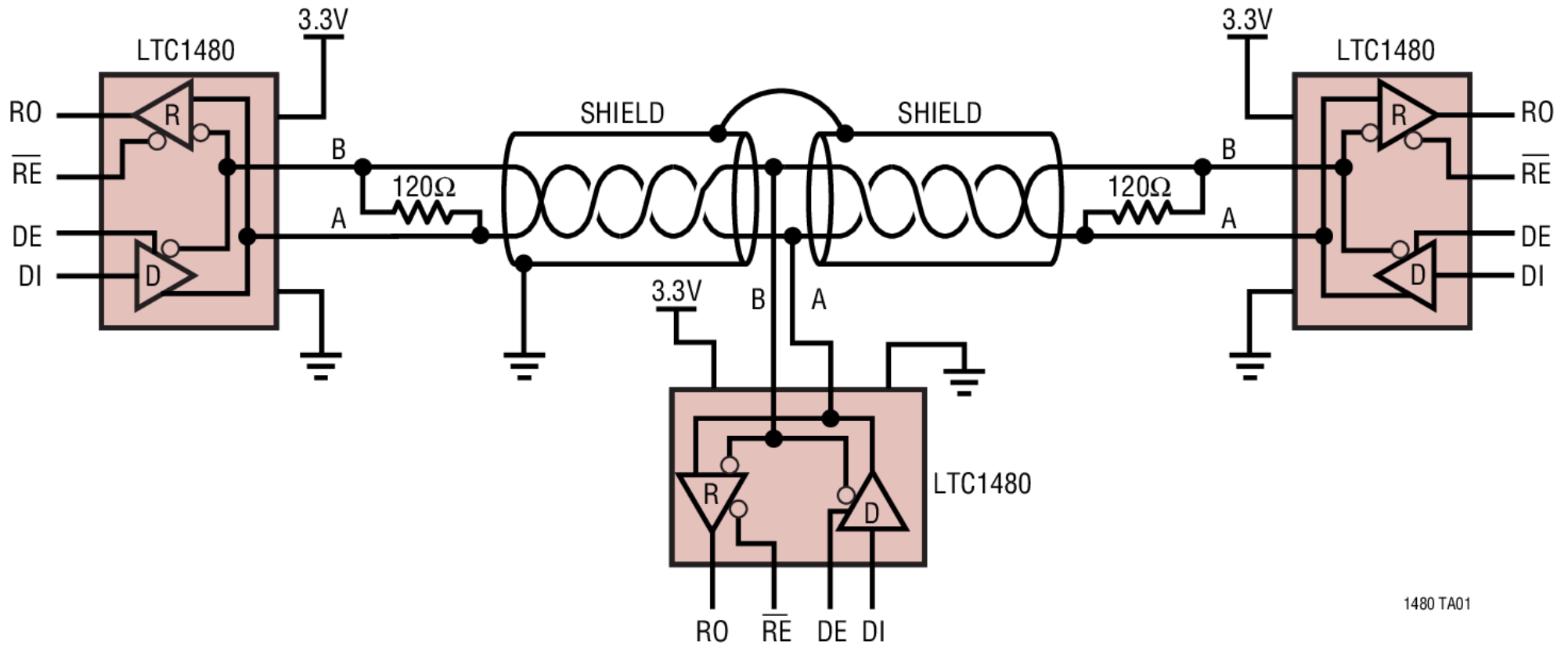
Comunicación Serie en Tensión – Comparativa

Especificaciones		RS232	RS423	RS422	RS485
Modo de Operación		NO DIFERENCIAL	NO DIFERENCIAL	DIFERENCIAL	DIFERENCIAL
Número de dispositivos		1 EMISOR 1 RECEPTOR	1 EMISOR 10 RECEPTORES	1 EMISOR 10 RECEPTORES	32 EMISORES 32 RECEPTORES
Máxima longitud del cable		15 metros	1200 metros	1200 metros	1200 metros
Máxima velocidad de transmisión		19,2 Kb/s	100 Kb/s	10 Mb/s	10 Mb/s
Rango de trabajo		+/-25V	+/-6V	+/-10V	-7V a +12V
Rango de señal	Alto	+/-15V	+/-6V	+/-6V	+/-6V
	Bajo	+/-3V	+/-3,6V	+/-2V	+/-1,5V
Sensibilidad de entrada receptor		+/-3V	+/-200mV	+/-200mV	+/-200mV
Resistencia de entrada receptor		7K	4K	$\geq 4K$	$\geq 12K$

Arduino i Raspberry Pi

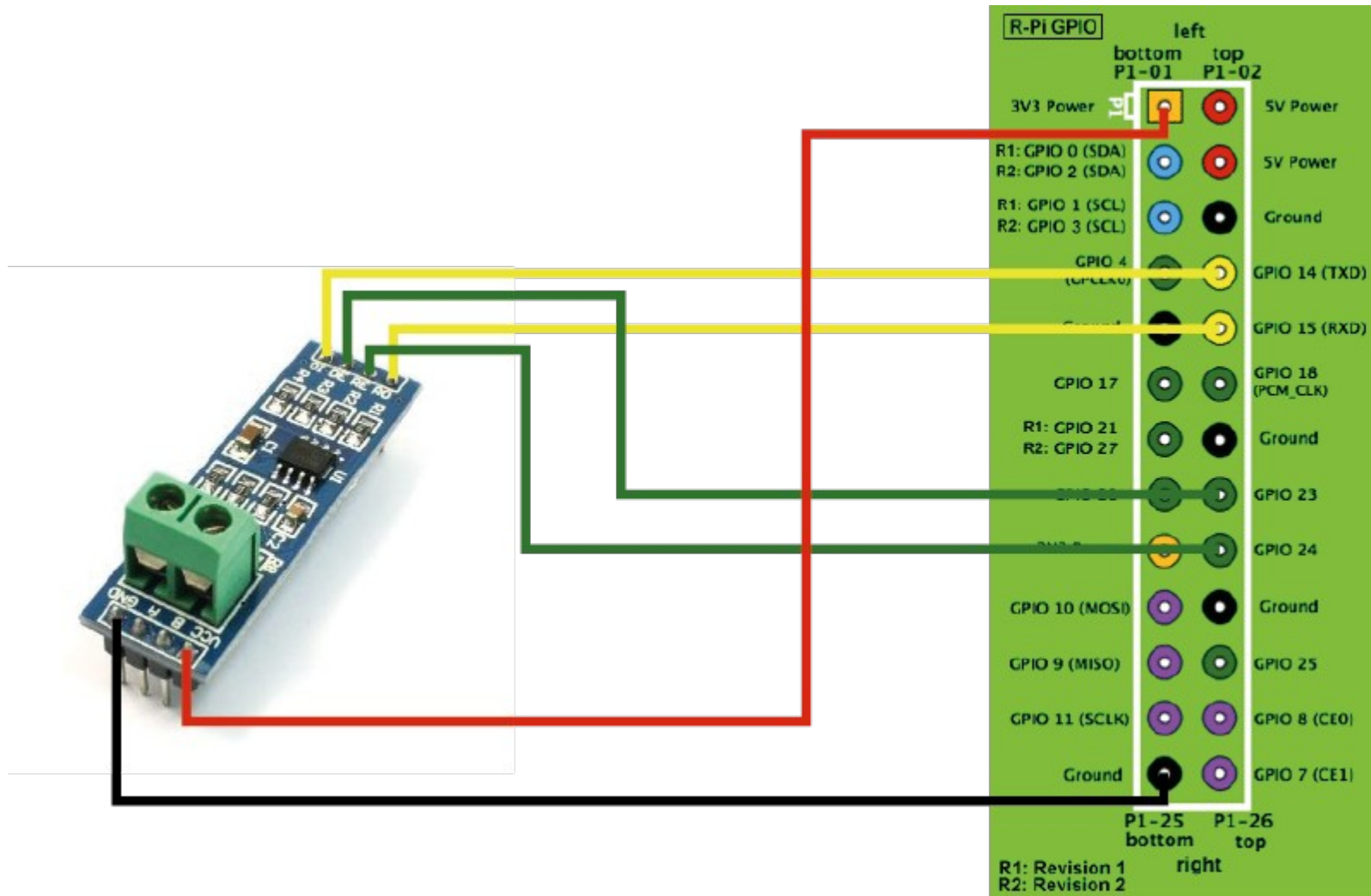
RS485

3.3V RS485 Network



1480 TA01

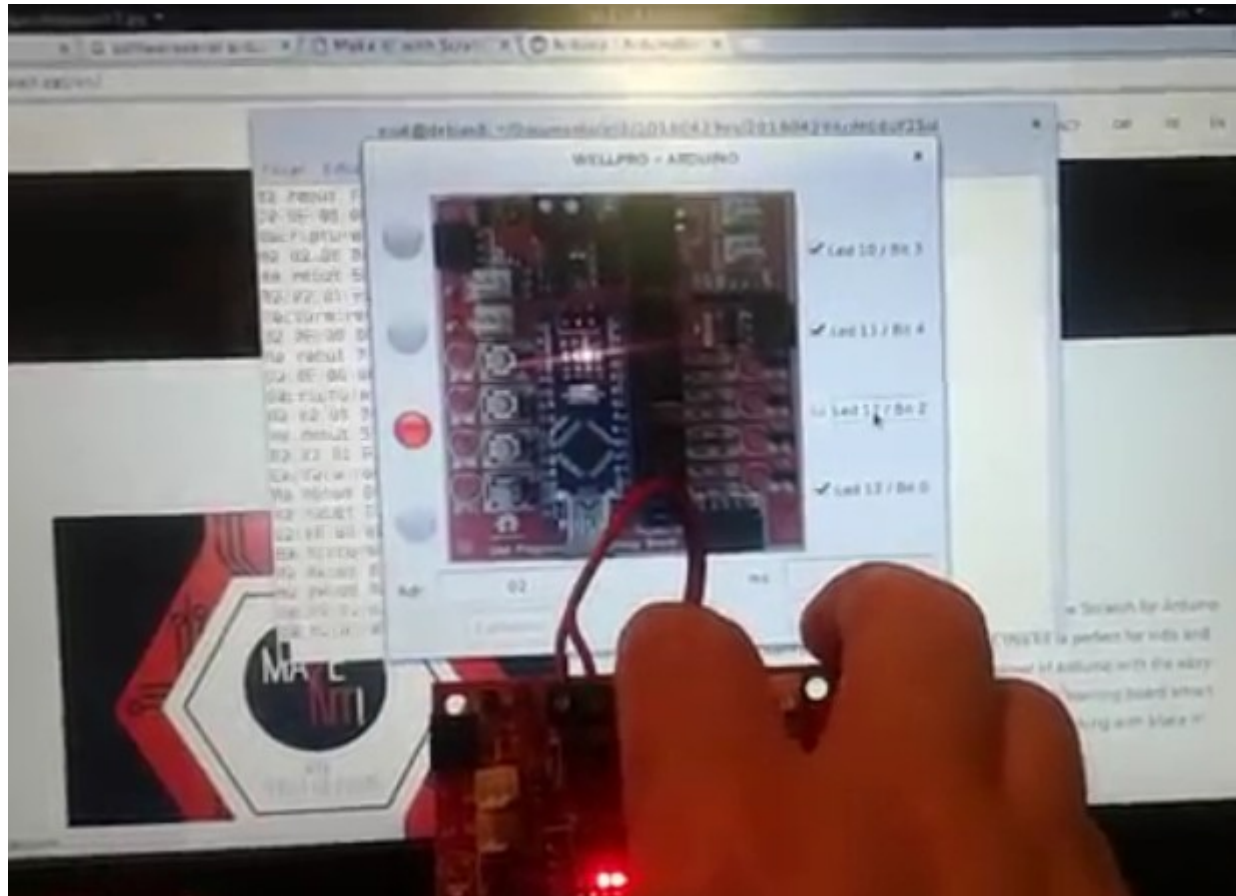
electronics.cat





Arduino i Raspberry Pi

RS485 -ModBusRTU

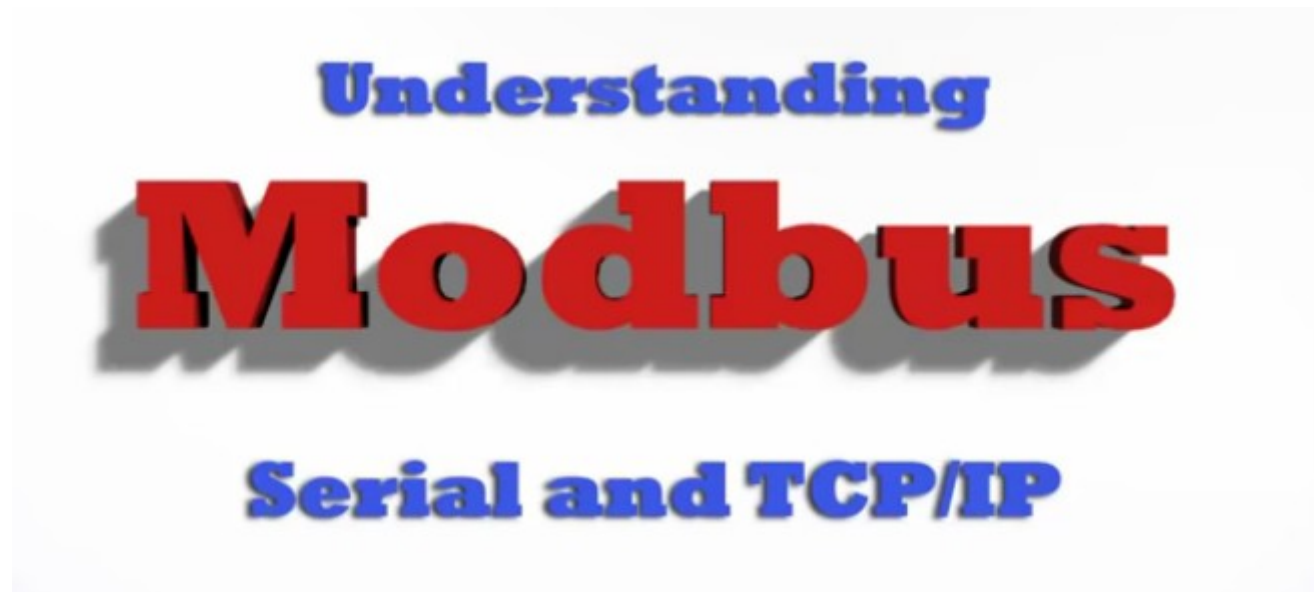


ModBus RTU sobre RS485 emprant Arduino i pyQt
 Modbus sobre USB emprant Arduino i pyQt
 Modbus RTU sobre un mòdul industrial



Arduino i Raspberry Pi

RS485 -ModBusRTU

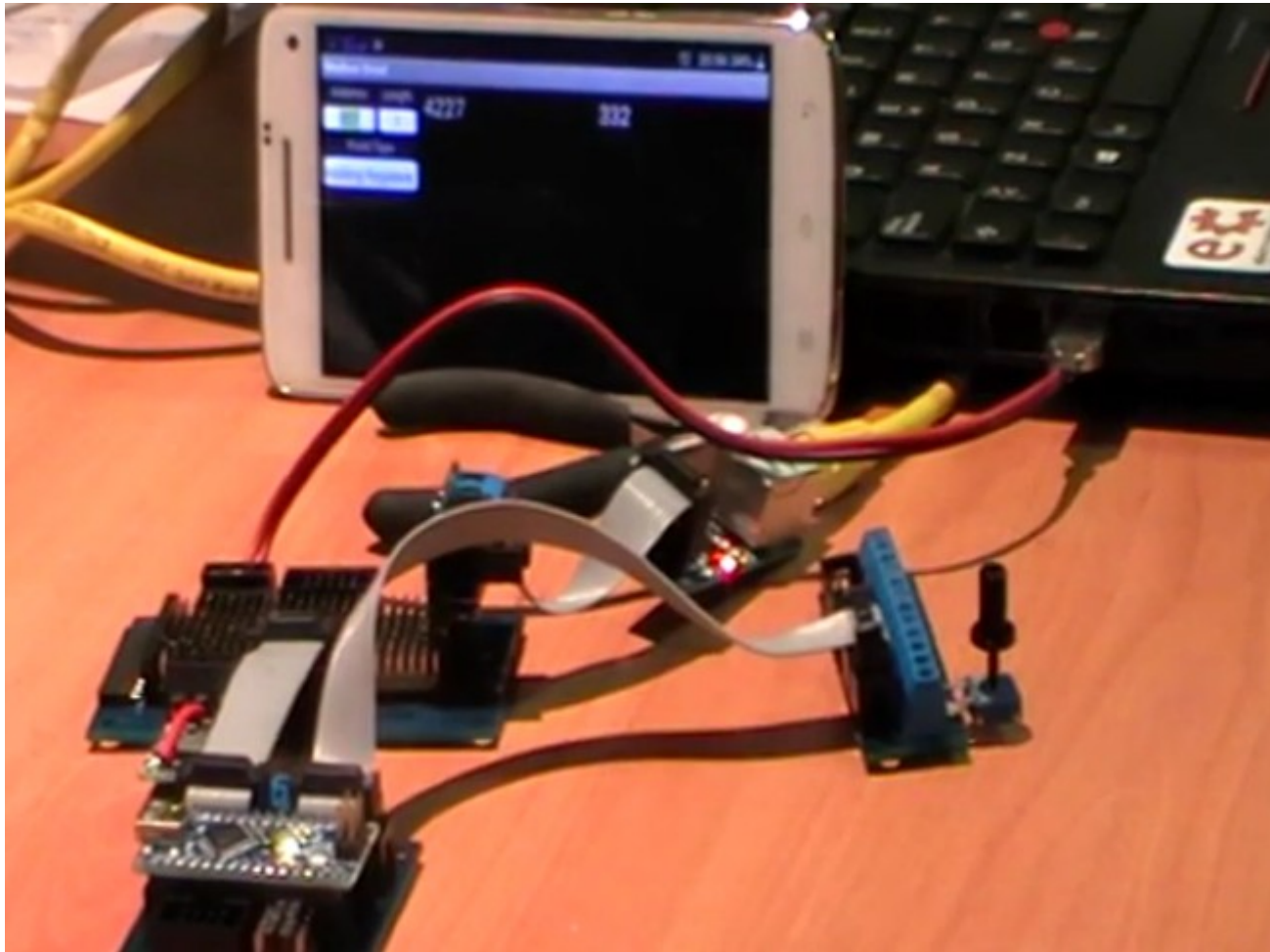


Vídeo explicatiu del ModBusRTU i el ModBusTCP



Arduino i Raspberry Pi

ModBusTCP



<http://www.binefa.cat/php/doc/modbus/>



Arduino i Raspberry Pi

ESP8266

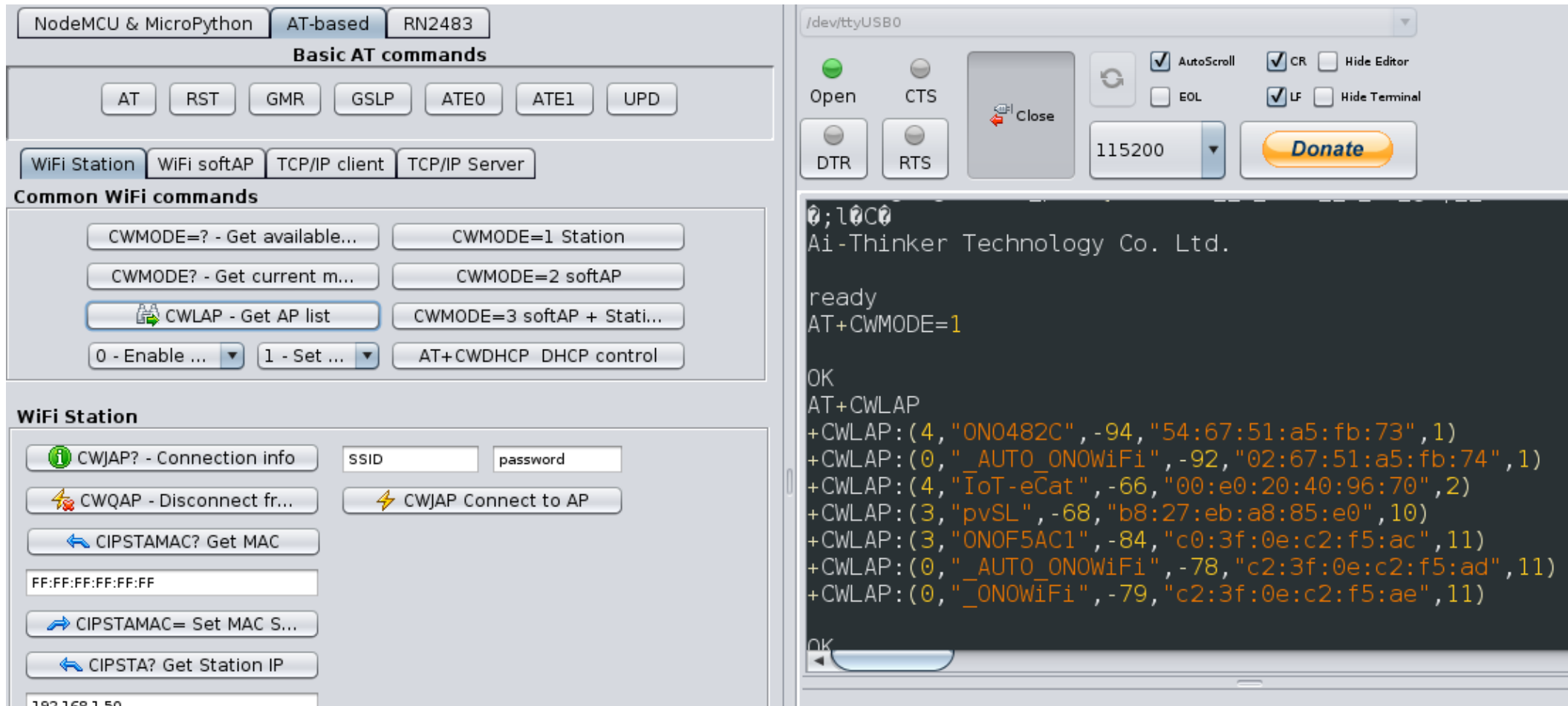
Programa per a carregar microprogramari a l'ESP8266
Binari amb ordres AT per a l'ESP8266

```
jordi@debianJB:~/Documents/electronics.cat/presentacions/20161129_telecosCat/codis/ESP8266$ python esptool.py  
--port /dev/ttyUSB0 write_flash 0x000000 v0.9.5.2\_ AT_Firmware.bin  
Connecting...  
Erasing flash...  
Wrote 520192 bytes at 0x00000000 in 50.0 seconds (83.2 kbit/s)...  
  
Leaving...  
jordi@debianJB:~/Documents/electronics.cat/presentacions/20161129_telecosCat/codis/ESP8266$
```



Arduino i Raspberry Pi

ESP8266



The screenshot displays the ESPlorer IDE interface. On the left, the 'Basic AT commands' section includes buttons for AT, RST, GMR, GSLP, ATE0, ATE1, and UPD. Below this, the 'Common WiFi commands' section features buttons for CWMODE=? (Get available...), CWMODE=1 Station, CWMODE=? (Get current m...), CWMODE=2 softAP, CWLAP - Get AP list, CWMODE=3 softAP + Stati..., 0 - Enable ..., 1 - Set ..., and AT+CWDHCP DHCP control. The 'WiFi Station' section includes buttons for CWJAP? - Connection info, CWQAP - Disconnect fr..., CIPSTAMAC? Get MAC, CIPSTAMAC= Set MAC S..., and CIPSTA? Get Station IP. On the right, the terminal window shows the output of the AT+CWMODE=1 command, displaying the list of available WiFi networks and their signal strengths.

Programa ESPlorer

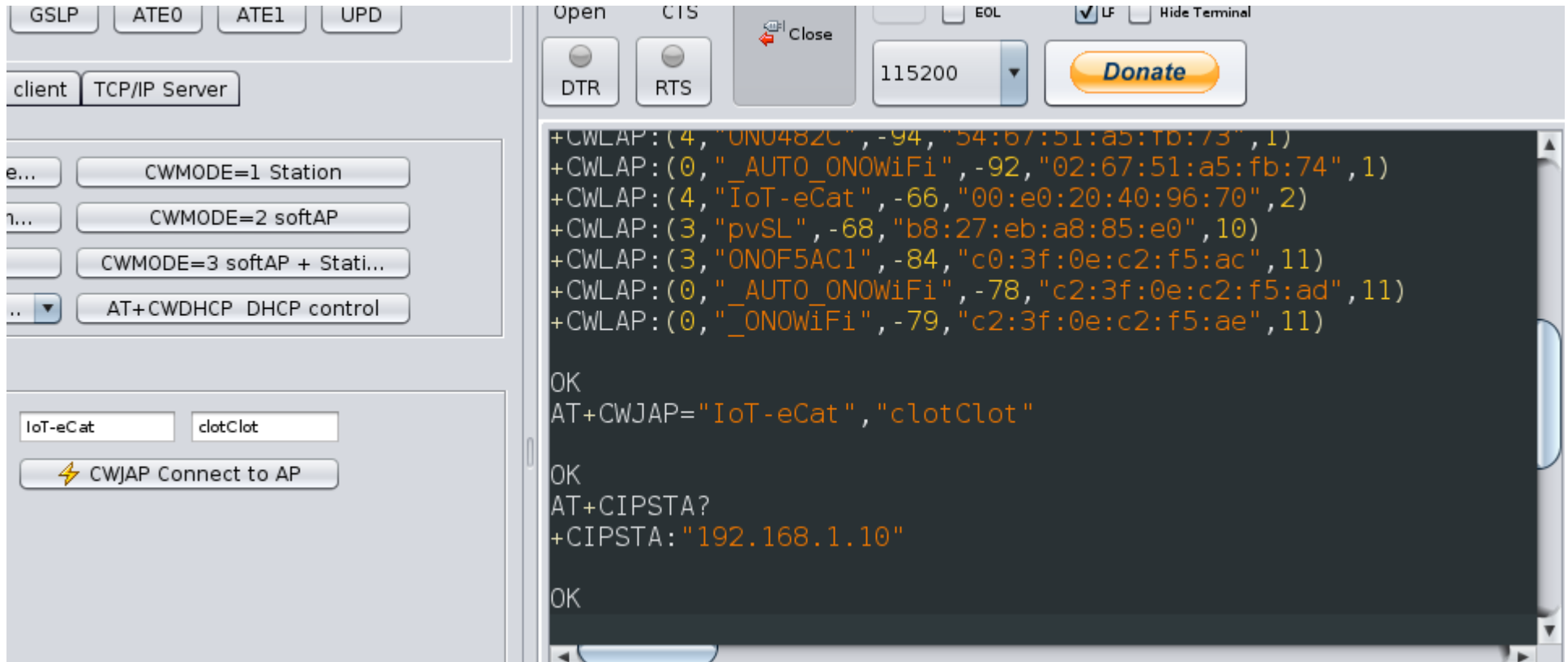
\$ java -jar ESPlorer.jar





Arduino i Raspberry Pi

ESP8266



Llistat d'ordres AT





Arduino i Raspberry Pi

ESP8266

Microprogramari `nodemcu_integer_0.9.6-dev_20150704.bin`

```
jordi@debianJB:~/Documents/electronics.cat/presentacions/20161129_telecosCat/ESP8266$  
./esptool.py --port /dev/ttyUSB0 write_flash 0x000000 nodemcu_integer_0.9.6-dev_20150704.bin  
Connecting...  
Erasing flash...  
Wrote 450560 bytes at 0x00000000 in 43.9 seconds (82.1 kbit/s)...  
Leaving...  
jordi@debianJB:~/Documents/electronics.cat/presentacions/20161129_telecosCat/ESP8266$
```





Arduino i Raspberry Pi

ESP8266

```

/dev/ttyUSB0
[Open] [CTS] [Close] [Refresh] [AutoScroll] [CR] [Hide Editor]
[DTR] [RTS] [9600] [Donate]
NodeMCU 0.9.6 build 20150704 powered by Lua
lua: cannot open init.lua
>
=wifi.getmode()
=wifi.getmode()
1
>
wifi.setmode(wifi.STATION)
wifi.setmode(wifi.STATION)
>
wifi.sta.config("IoT-eCat","clotClot")
wifi.sta.config("IoT-eCat","clotClot")
>
=wifi.sta.getip()
=wifi.sta.getip()
192.168.1.11    255.255.255.0    192.168.1.1
>

```





Arduino i Raspberry Pi

ESP8266

```
> w([==[end]==]),
> w([==[end]==]);
> file.close();
> dofile("clientWeb.lua");
nil
> file.remove("closeTcp.lua");
> file.open("closeTcp.lua","w+");
> w = file.writeline
> w([==[srv.close(srv)==]);
> w([==[srv=nil]==]);
> file.close();
> dofile("closeTcp.lua");
>
dofile("clientWeb.lua")
dofile("clientWeb.lua")
nil
```



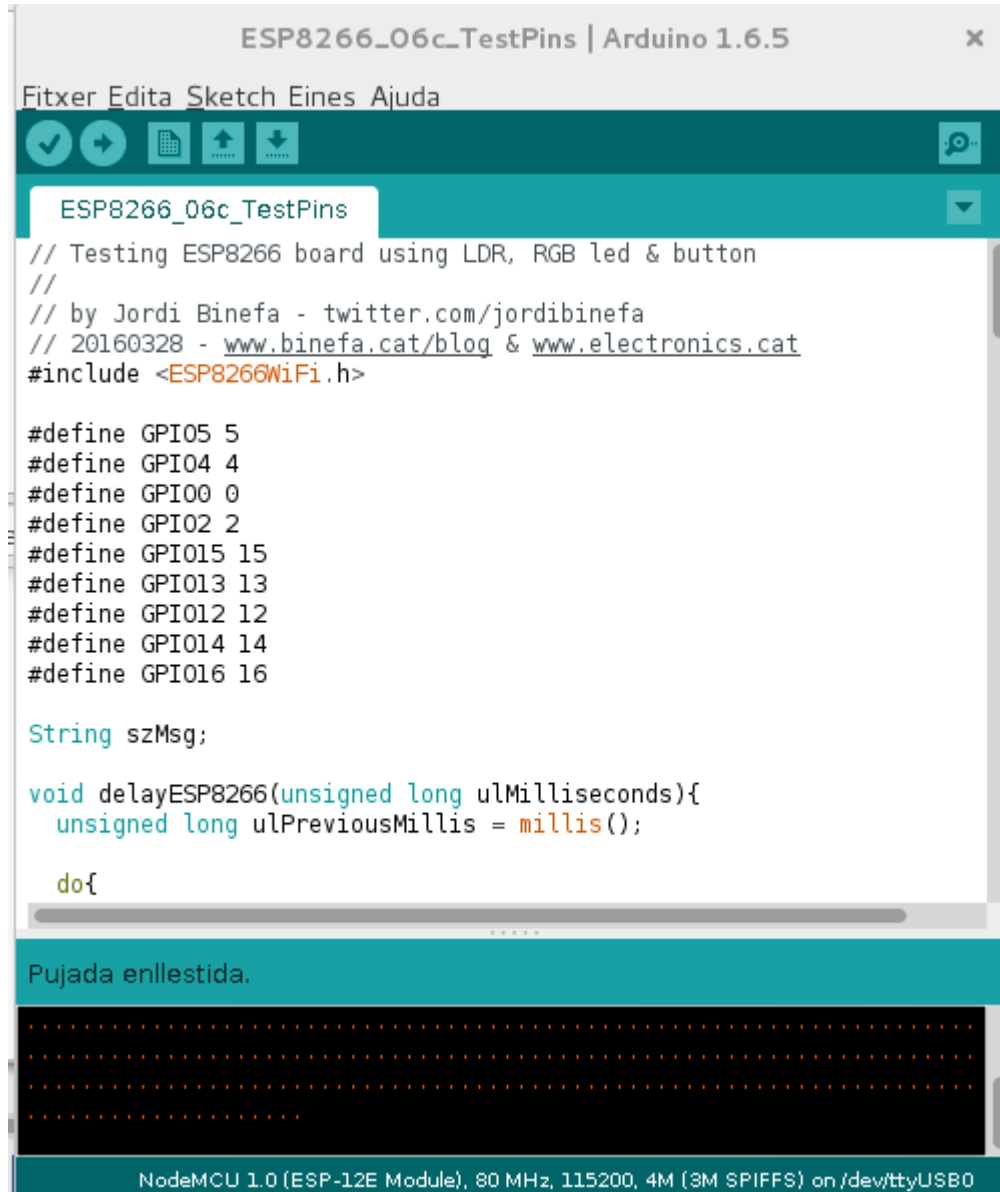
Client web en Lua

Per tancar la connexió TCP en Lua



Arduino i Raspberry Pi

ESP8266



```

ESP8266_06c_TestPins | Arduino 1.6.5
Fitxer Edita Sketch Eines Ajuda

ESP8266_06c_TestPins
// Testing ESP8266 board using LDR, RGB led & button
//
// by Jordi Binefa - twitter.com/jordibinefa
// 20160328 - www.binefa.cat/blog & www.electronics.cat
#include <ESP8266WiFi.h>

#define GPIO5 5
#define GPIO4 4
#define GPIO0 0
#define GPIO2 2
#define GPIO15 15
#define GPIO13 13
#define GPIO12 12
#define GPIO14 14
#define GPIO16 16

String szMsg;

void delayESP8266(unsigned long ulMilliseconds){
  unsigned long ulPreviousMillis = millis();

  do{

```

Pujada enllestida.

NodeMCU 1.0 (ESP-12E Module), 80 MHz, 115200, 4M (3M SPIFFS) on /dev/ttyUSB0

Microprogramari de verificació





Arduino i Raspberry Pi

ESP8266

/dev/ttyUSB0

Envia

UDPClientMAC02 | Arduino 1.6.5

Fitxer Edita Sketch Eines Ajuda

UDPClientMAC02

```

Serial.println();
Serial.println();
Serial.print("Connecting to ");
Serial.println(ssid);

WiFi.begin(ssid, password);

while (WiFi.status() != WL_CONNECTED) {
  delay(500);
  Serial.print(".");
}

Serial.println("");
Serial.println("WiFi connected");
Serial.println("IP address: ");
Serial.println(WiFi.localIP());

// Udp.beginPacket("192.168.1.105", 888);
// Udp.beginPacket("192.168.1.13", 6000);
// Udp.beginPacket("192.168.1.52", 6000);
Udp.beginPacket("192.168.1.12", 6000);
        
```

UDPdIlg

Text :

IP servidor :

Port Tx : <-> Port Rx :

Tramet

```

00004) 5c:cf:7f:c0:84:07
00005) 5c:cf:7f:c0:84:07
00006) 5c:cf:7f:c0:84:07
00007) 5c:cf:7f:c0:84:07
00008) 5c:cf:7f:c0:84:07
00009) 5c:cf:7f:c0:84:07
00010) 5c:cf:7f:c0:84:07
00011) 5c:cf:7f:c0:84:07
00012) 5c:cf:7f:c0:84:07
00013) 5c:cf:7f:c0:84:07
00014) 5c:cf:7f:c0:84:07
00015) 5c:cf:7f:c0:84:07
00016) 5c:cf:7f:c0:84:07
        
```





Arduino i Raspberry Pi

ESP8266

A partir de la pàgina 18 del document [ethernetTcpUdp_00.pdf](#)

Torn de preguntes ...



... i sessió pràctica.



Arduino i Raspberry Pi

Presentació descarregable a : <http://binefa.cat/blog>

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://www.electronics.cat>

<http://www.makeit.cat>

Moltes gràcies per la vostra atenció