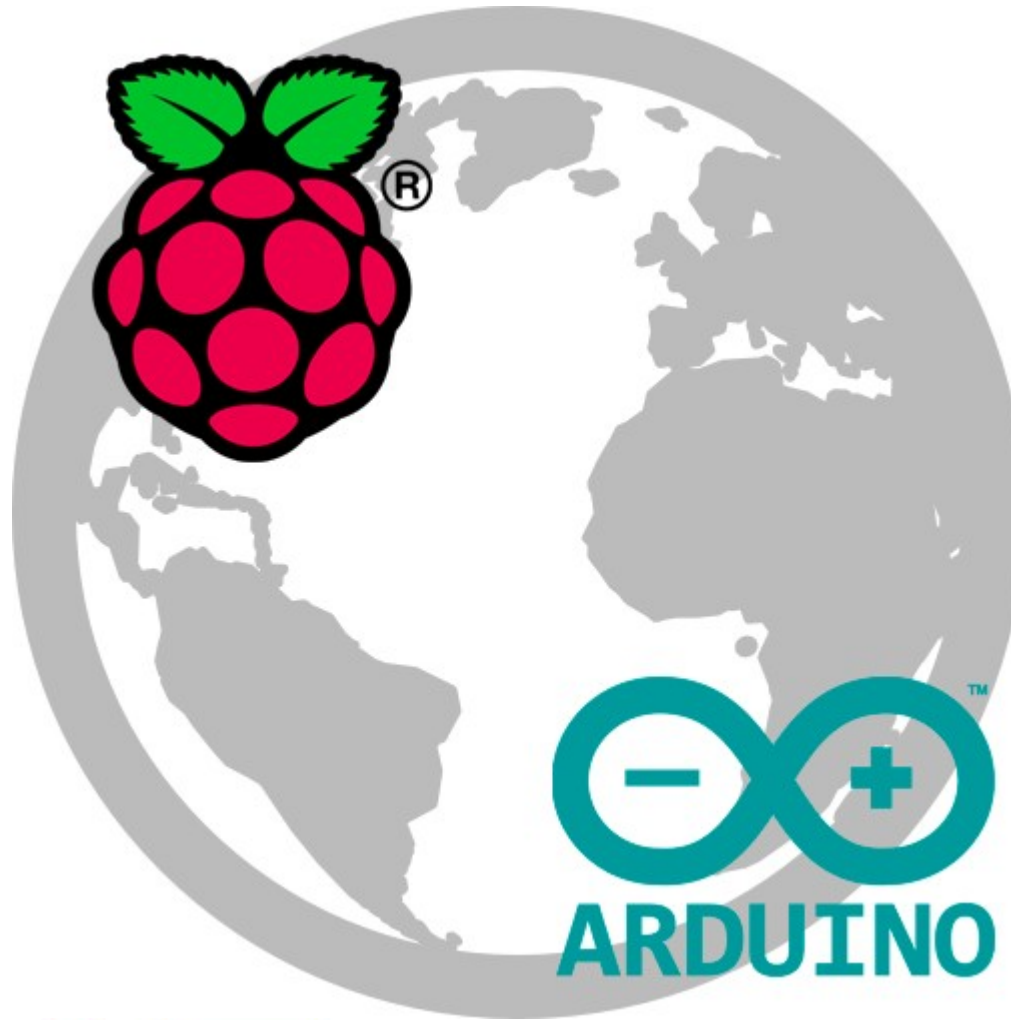


# Arduino i Raspberry Pi



Formació  
EN  
ENGINYERIA

**Engineers**  
Industrials de Catalunya

**21 de febrer 2018**



**JESUÏTES** El Clot  
Escola del Clot

# Arduino i Raspberry Pi

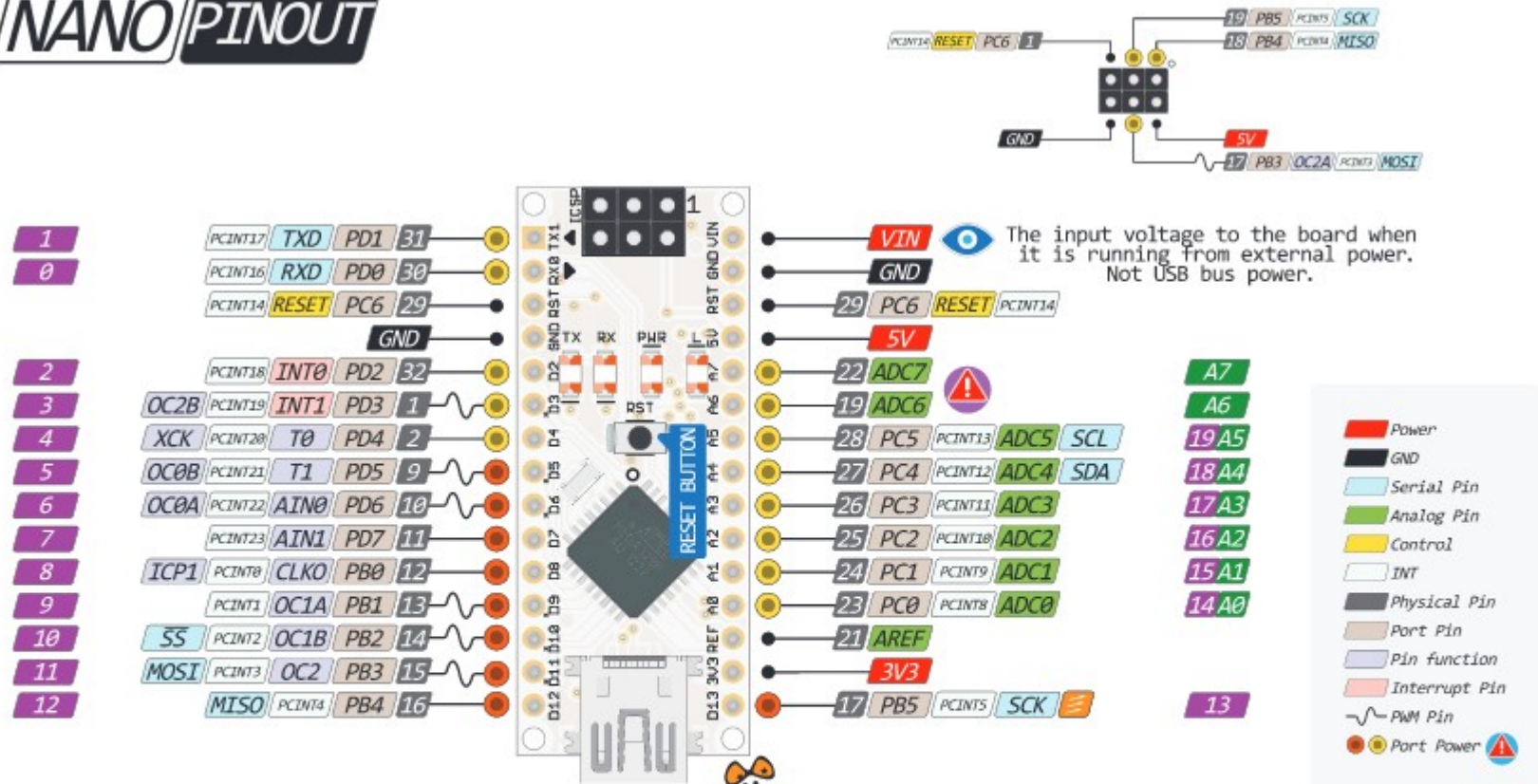
## Part d'Arduino (matí)

1. El microcontrolador Arduino
2. Sensors i actuadors. Exemples pràctics
3. Maneres de programar un Arduino:
  - \* Per blocs
  - \* Des de l'IDE d'Arduino
4. Connexió de perifèrics a l'Arduino:  
entrades i sortides digitals, SPI, I2C, UART
5. Comunicació entre l'Arduino i l'ordinador
6. Bluetooth i RS485

# El microcontrolador Arduino

## L'Arduino Nano

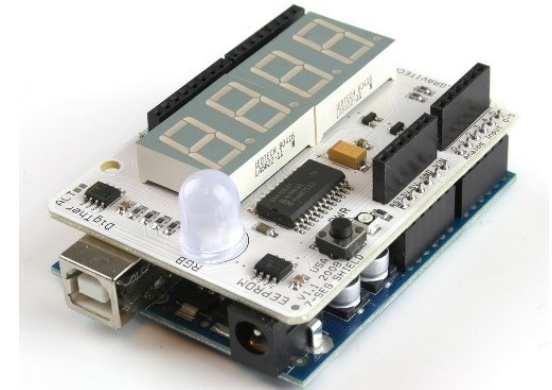
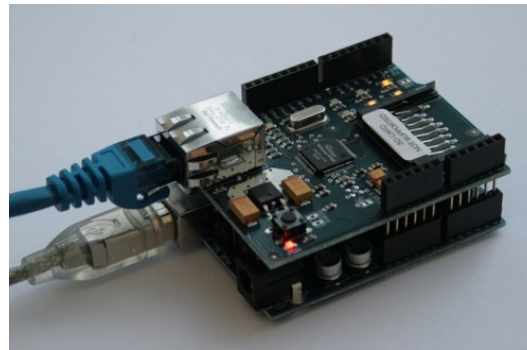
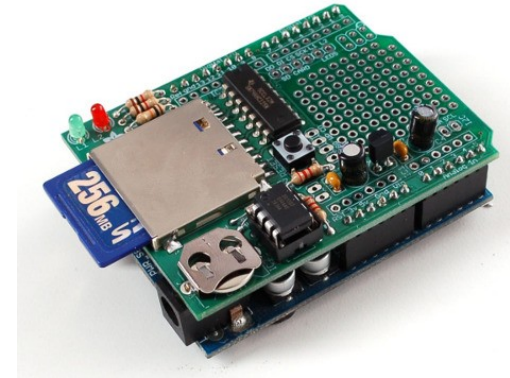
NANO PINOUT





# El microcontrolador Arduino

## Plaques complementàries ( shield / hat )



La majoria de plaques complementàries venen amb biblioteques i arxius de capçalera que acceleren el procés de funcionament.

<http://arduino.cc/en/pmwiki.php?n=Main/ArduinoShields>





# El microcontrolador Arduino

## Interacció amb el món físic

### Sensors



**Lectura**  
**d'informació**

### Actuadors

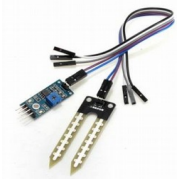


**Escriptura**  
**d'informació**

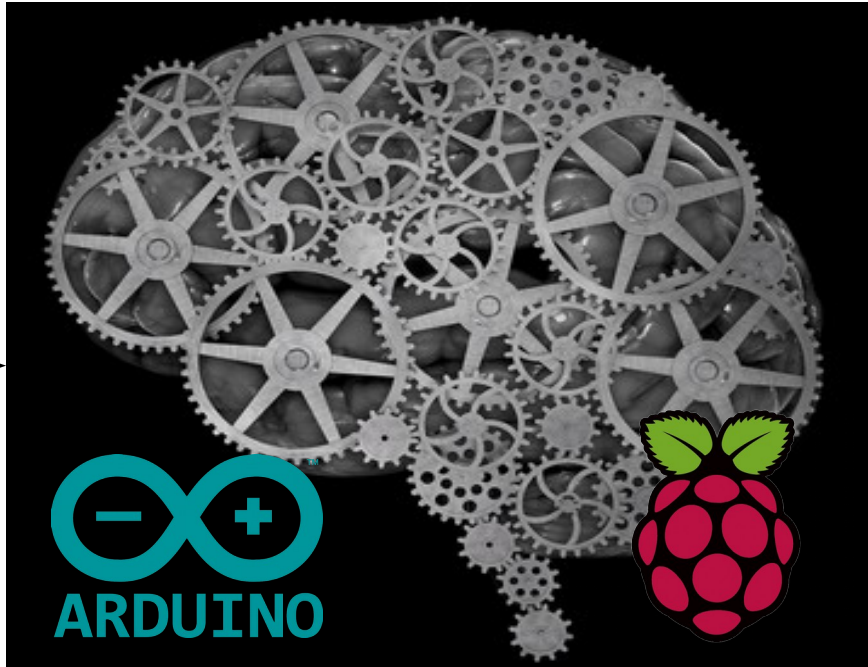


# El microcontrolador Arduino

## Automatització del món físic



**Sensors**



**Actuadors**



**Sortida**  
de sensors  
Escriptura -->

**Entrada**  
d'informació  
--> Lectura

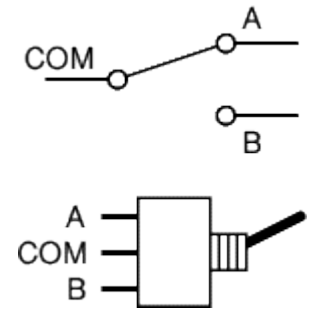
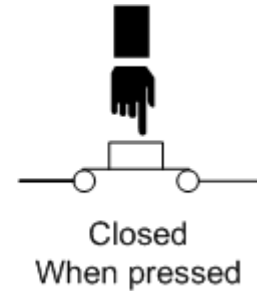
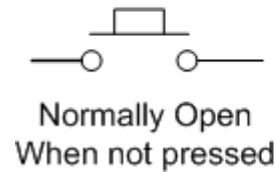
**Sortida**  
d'informació  
Escriptura -->

**Entrada**  
d'actuadors  
--> Lectura



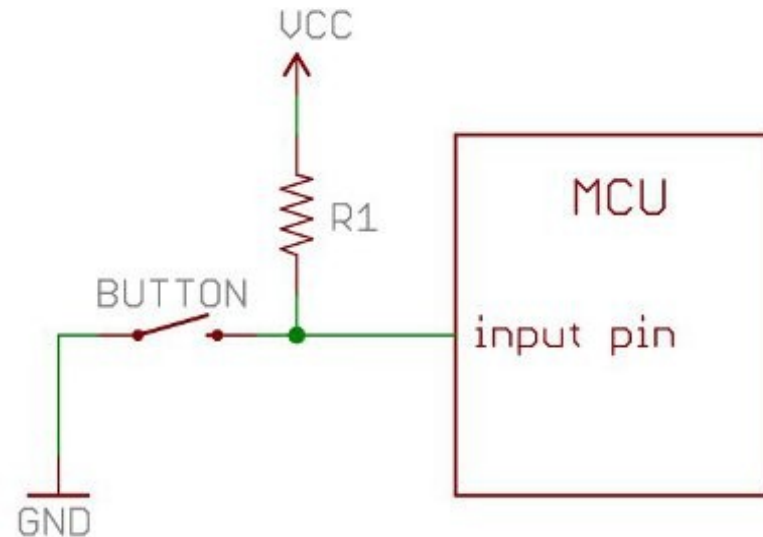
# El microcontrolador Arduino

## Lectura de l'estat d'un sensor digital



**Cert : 1.8v, 3.3V,  
5V**  
**Fals : 0v**

**True / False**  
**High / Low**

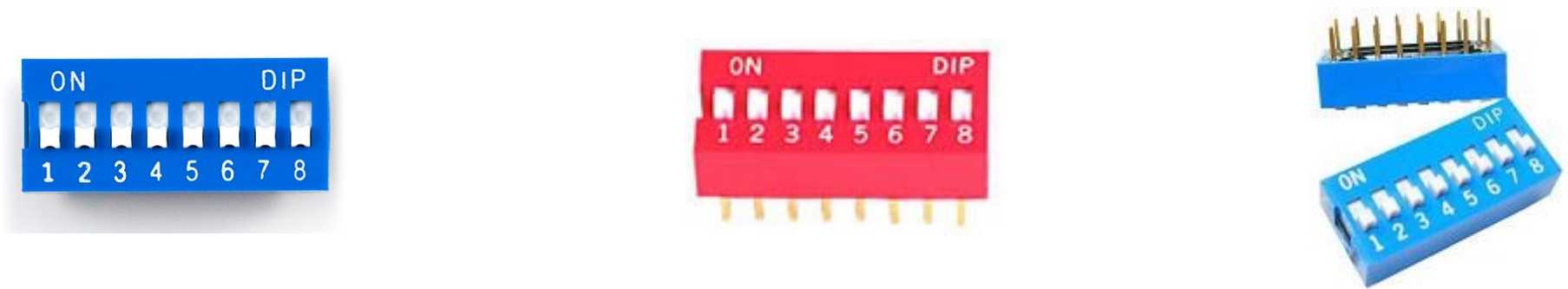




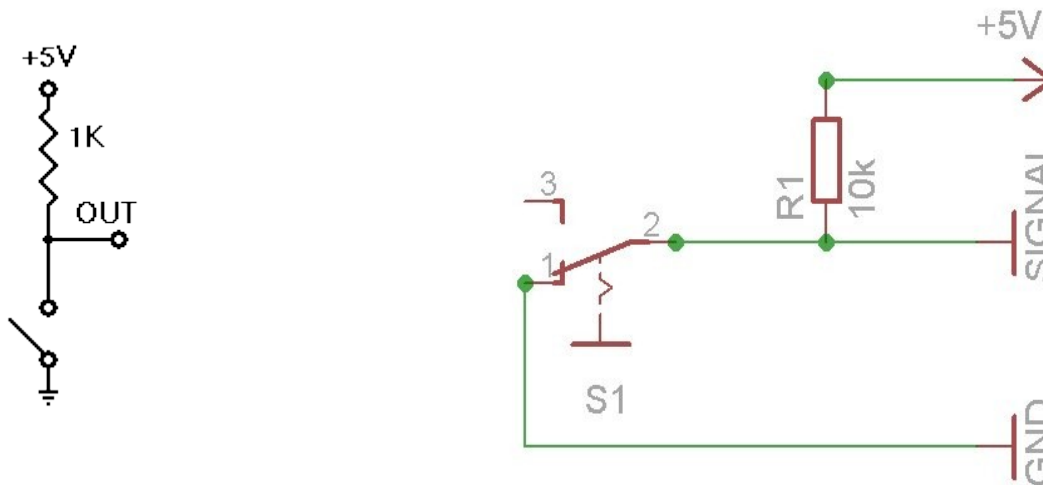
# El microcontrolador Arduino

## Lectura de l'estat d'un sensor digital

**uSW** : Microrruptor (dip-switch, microswitch)



**Pull-up** : Connexió d'un resistor a alimentació per assegurar un nivell lògic a una entrada digital

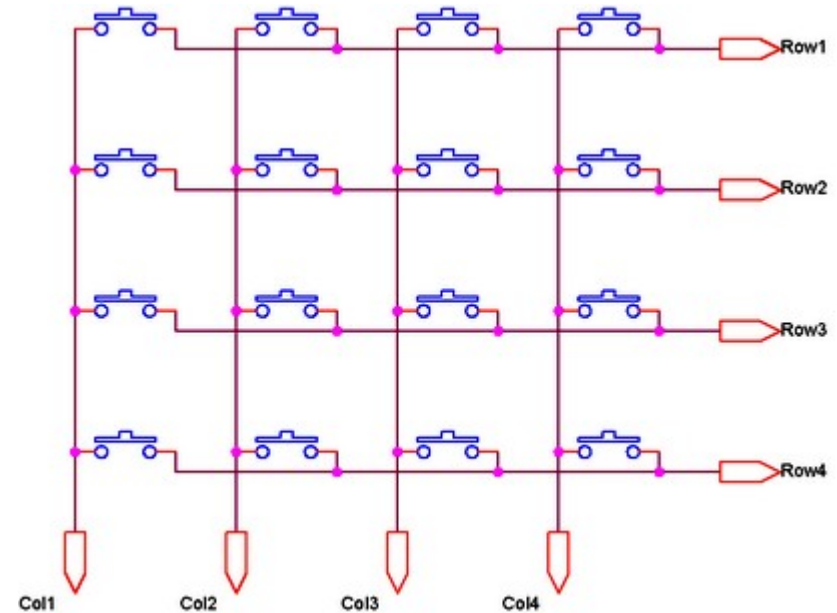






# El microcontrolador Arduino

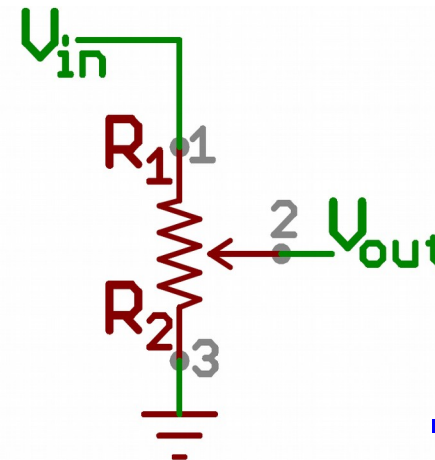
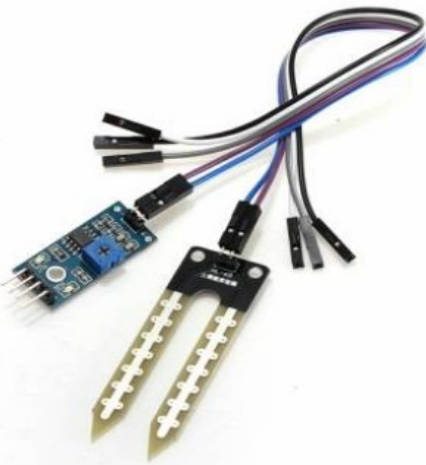
## Lectura de la informació d'un teclat





# El microcontrolador Arduino

## Lectura d'informació analògica



**0..3,3V**

**0..5V**

**0..10V**

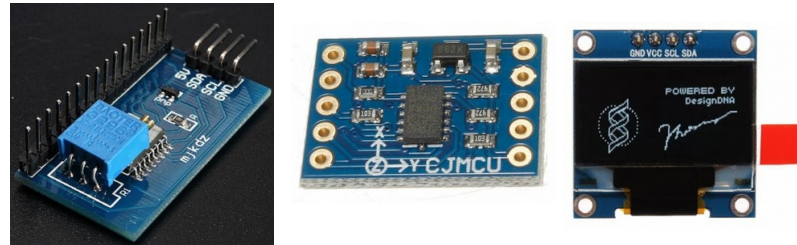
**-10..10V**



# El microcontrolador Arduino

## Protocols I2C (TWI), SPI i 1-Wire

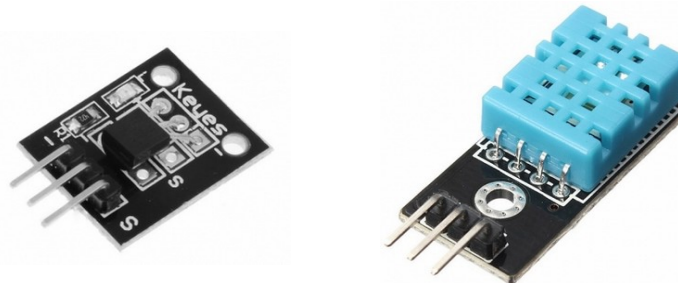
**I2C : Inter-Integrated Circuit**



**SPI : Serial Peripheral Interface Bus**



**1-Wire**

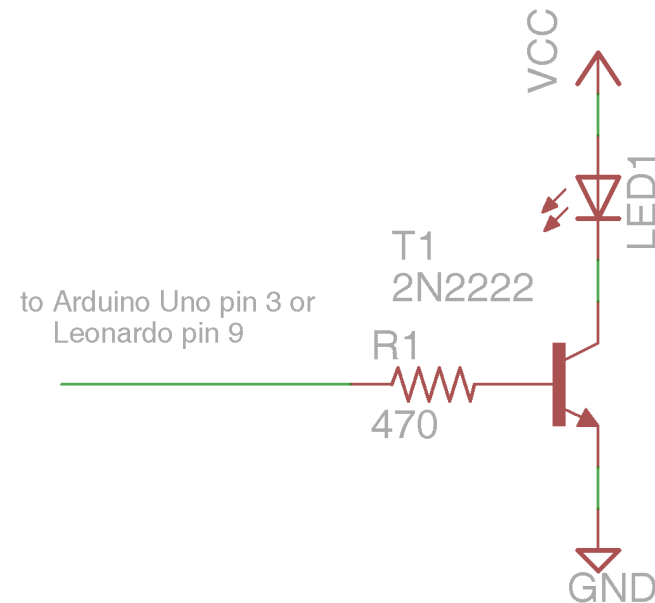
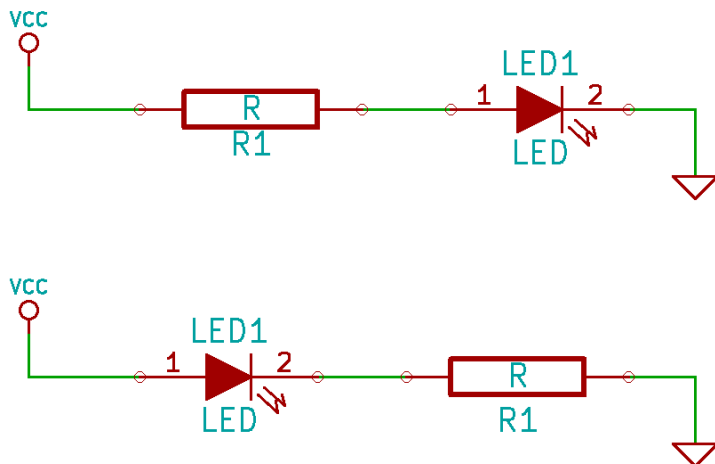
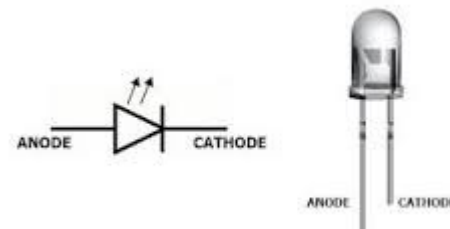
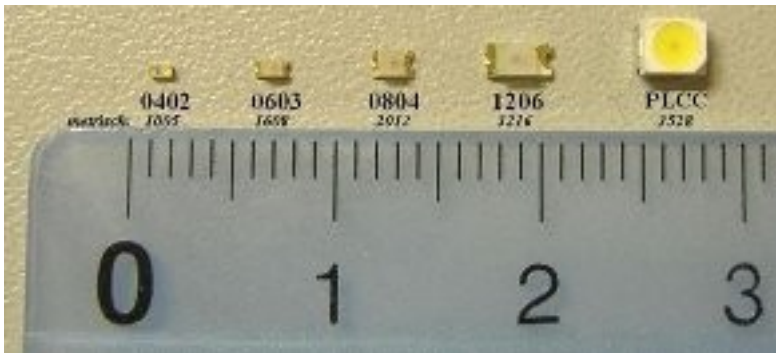




# El microcontrolador Arduino

## Espectura d'estat a un actuator digital

### LED : Díode emissor de llum



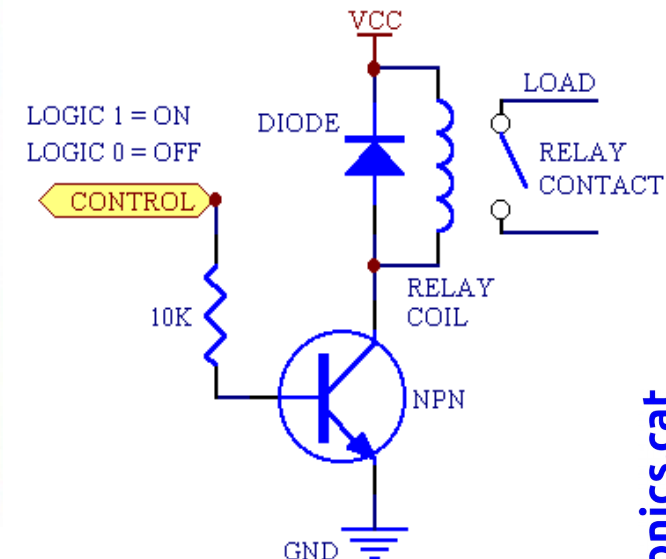
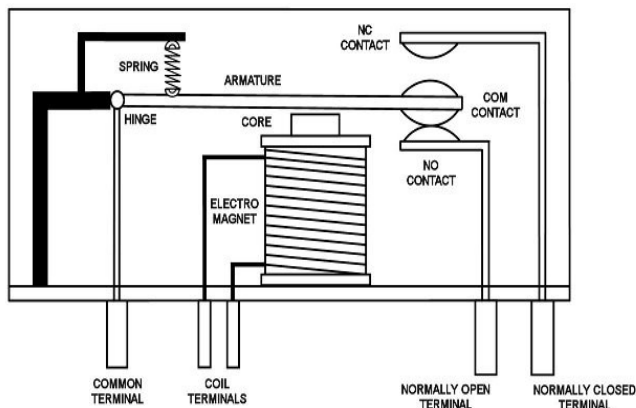
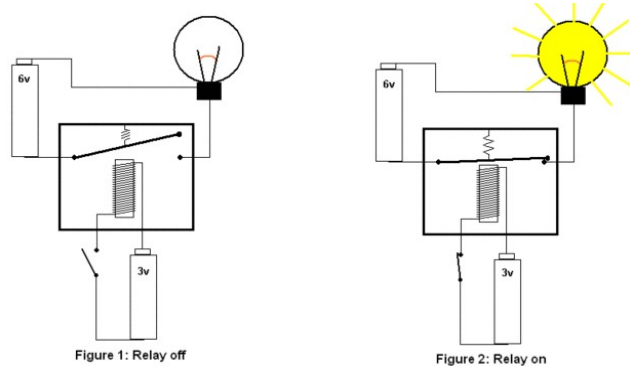




# El microcontrolador Arduino

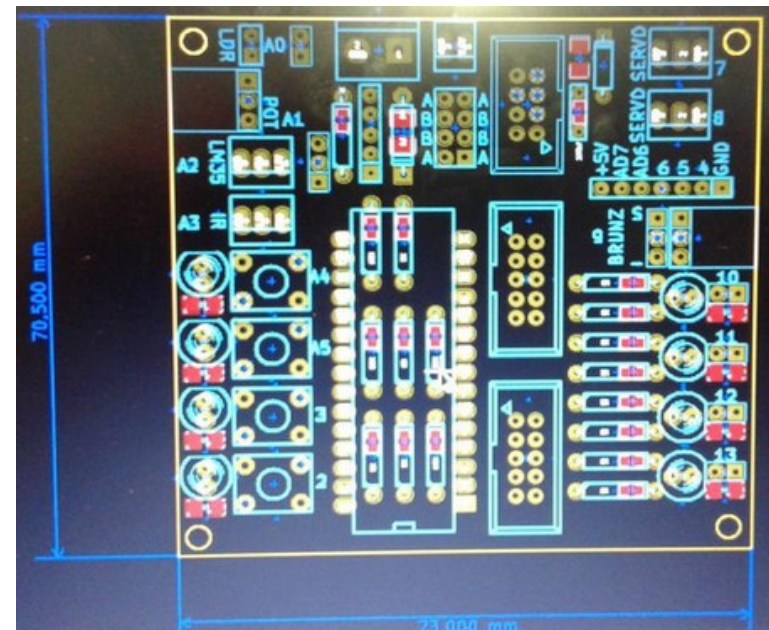
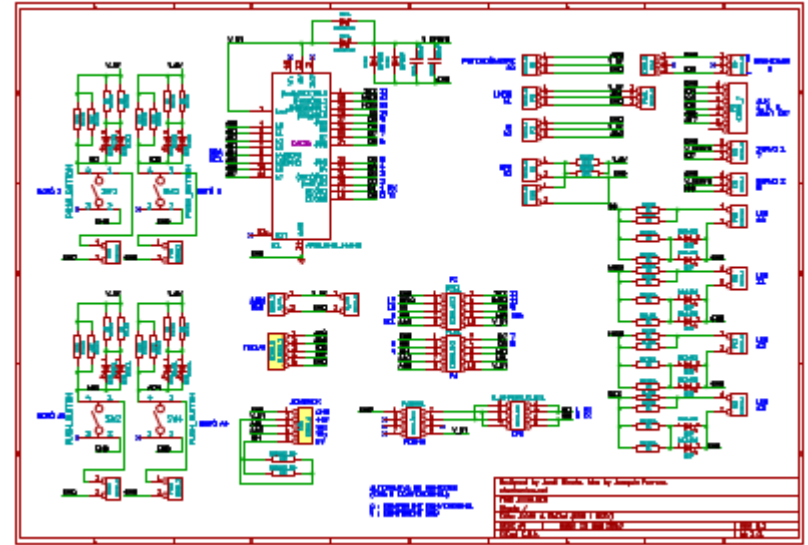
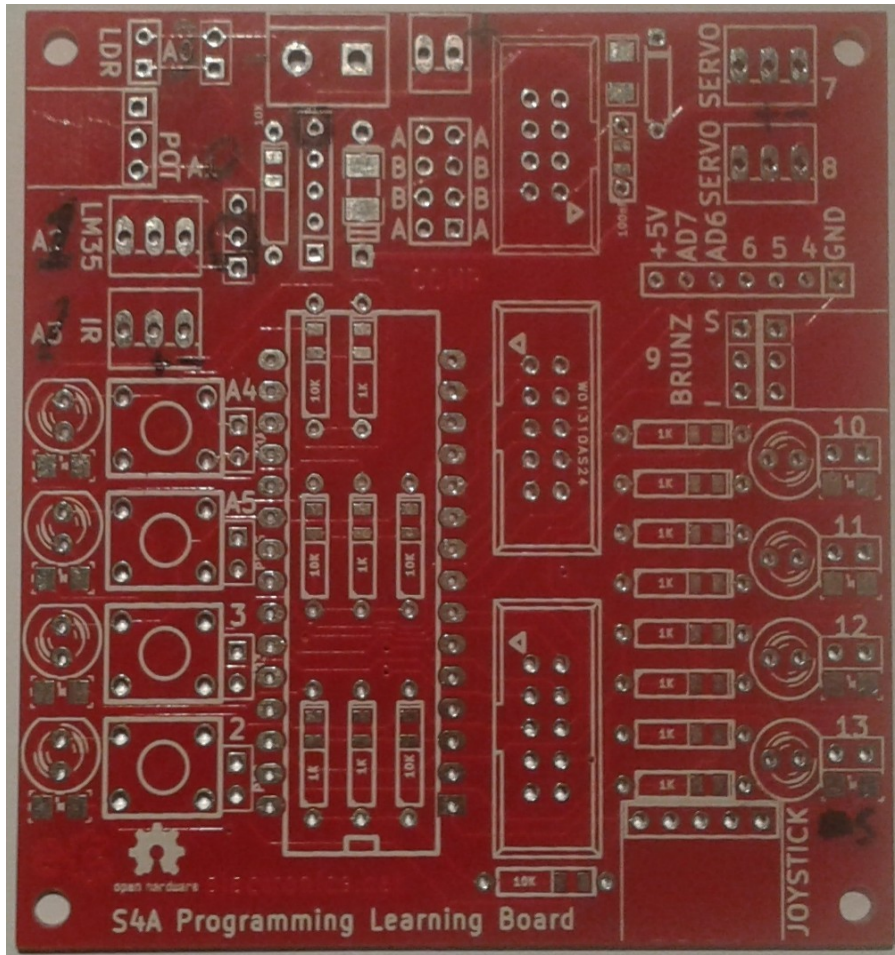
## Esriptura d'estat a un actuator digital

**Relé** : Sistema electromecànic que modifica l'estat d'un commutador. Amb una tensió de control petita s'activa un electroimant podent controlar tensions molt superiors.



# El microcontrolador Arduino

## La placa S4A Programming Learning Board



Placa desenvolupada amb l'entorn de disseny integrat de programari lliure

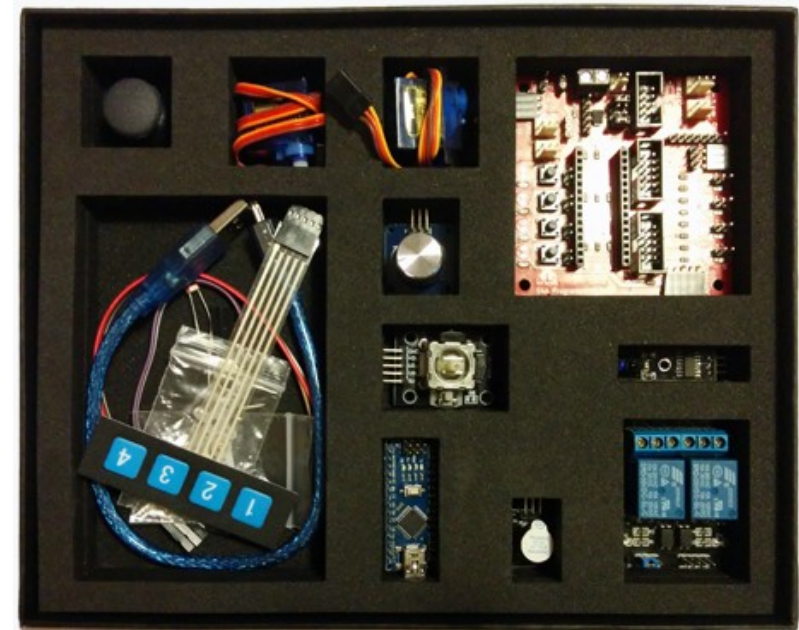






# El microcontrolador Arduino

## Make It!

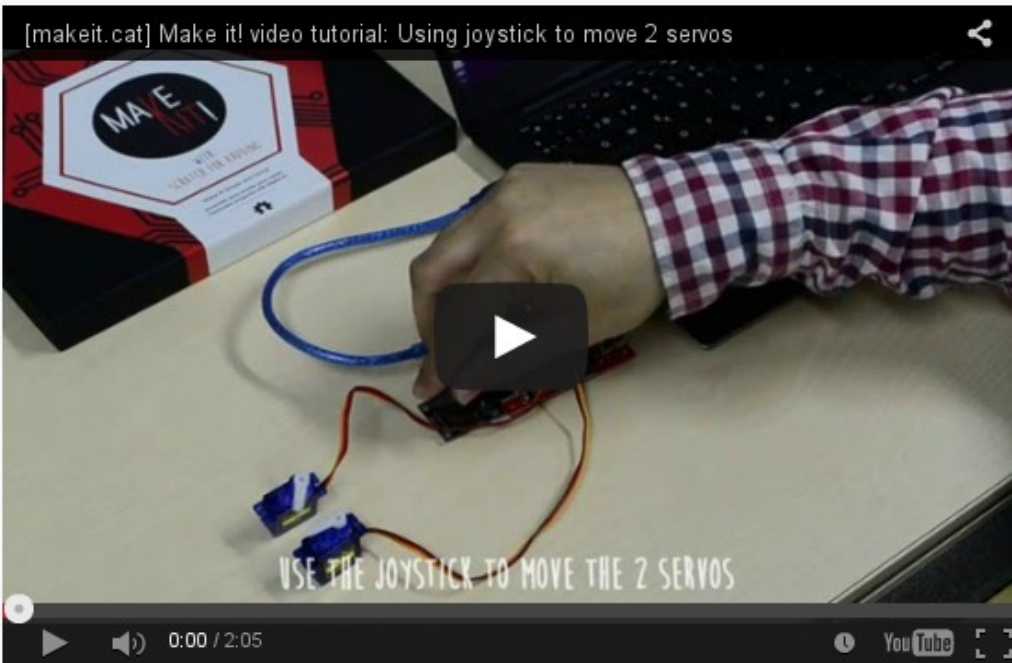


<http://www.makeit.cat>



# El microcontrolador Arduino

## Make It!



Tutorial *Make it!*: Utilitzant un joystick per moure 2 servos

Com moure 2 servos utilitzant un joystick, el *Make it!* i el programa Scratch for Arduino.

<http://makeit.cat/cat/>



# El microcontrolador Arduino

## Entorn de l'S4A

Based on Scratch from the MIT Media Lab

Fitxer Edita Ajuda

Arduíno1

x: 0 y: 0 direcció: 90

Programes Vestits Sons

al prémer

per sempre

si ¿sensor Digital2 premut?

digital 13 apagat

si no digital 13 encès

si ¿sensor Digital3 premut?

digital 12 apagat

si no digital 12 encès

si valor del sensor Analog5 < 15

digital 11 encès

si no digital 11 apagat

si valor del sensor Analog4 < 15

digital 10 encès

si no digital 10 apagat

al prémer

per sempre

si no ¿sensor Digital2 premut?

analògic 9 valor 255

si no analògic 9 valor 0

pcbs4A\_01

Arduíno1  
port: USB0

Analog0 628  
Analog1 522  
Analog2 519  
Analog3 114  
Analog4 1023  
Analog5 1023  
Digital2 true  
Digital3 true

Arduíno Duemilanove

Nova animació:

Arduíno...

Escenari

x: -636 y: -87

Codis d'exemple d'aquesta presentació per a l'S4A

[www.s4a.cat](http://www.s4a.cat)

# El microcontrolador Arduino

## Microprogramari de l'S4A

El microprogramari és el programari del maquinari

*Català*

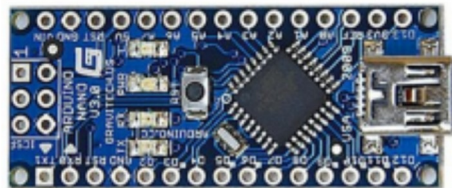
**Programari**  
**Maquinari**  
**Microprogramari**

*Anglès*

**Software**  
**Hardware**  
**Firmware**



Arduino Nano



```

S4AFirmware16 | Arduino 1.6.5
Eitxer Edita Sketch Eines Ajuda

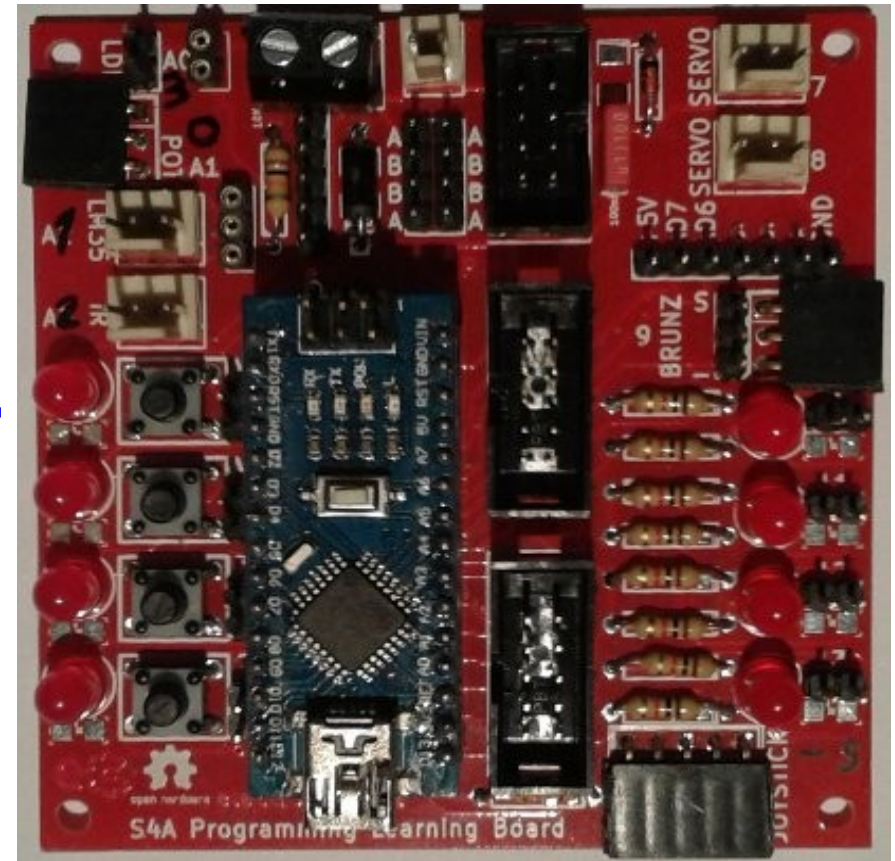
S4AFirmware16
// NEW IN VERSION 1.6c (by Jorge Gomez):
// Fixed variable type in pin structure: pin.state should be bool
// Optimized speed of execution while receiving data from S4A

// NEW IN VERSION 1.6b (by Jorge Gomez):
// Added new structure arduinoPins to hold the pins info
// - This makes the code easier to read and modify (IMH)
// - Allows to change the type of pin more easily to me
// - Eliminates the need of having to deal with different pin types
// - By using an enum to hold all the possible output pins
// Changed all functions using old style pin access: conPin to arduinoPins
// Fixed possible overflow every 70 minutes (2e32 us) in millis()
// Some minor coding style fixes
  
```

Microprogramari per l'Arduino per funcionar amb l'S4A :  
<http://vps34736.ovh.net/S4A/S4AFirmware16.ino>

# S4A Programming Learning Board

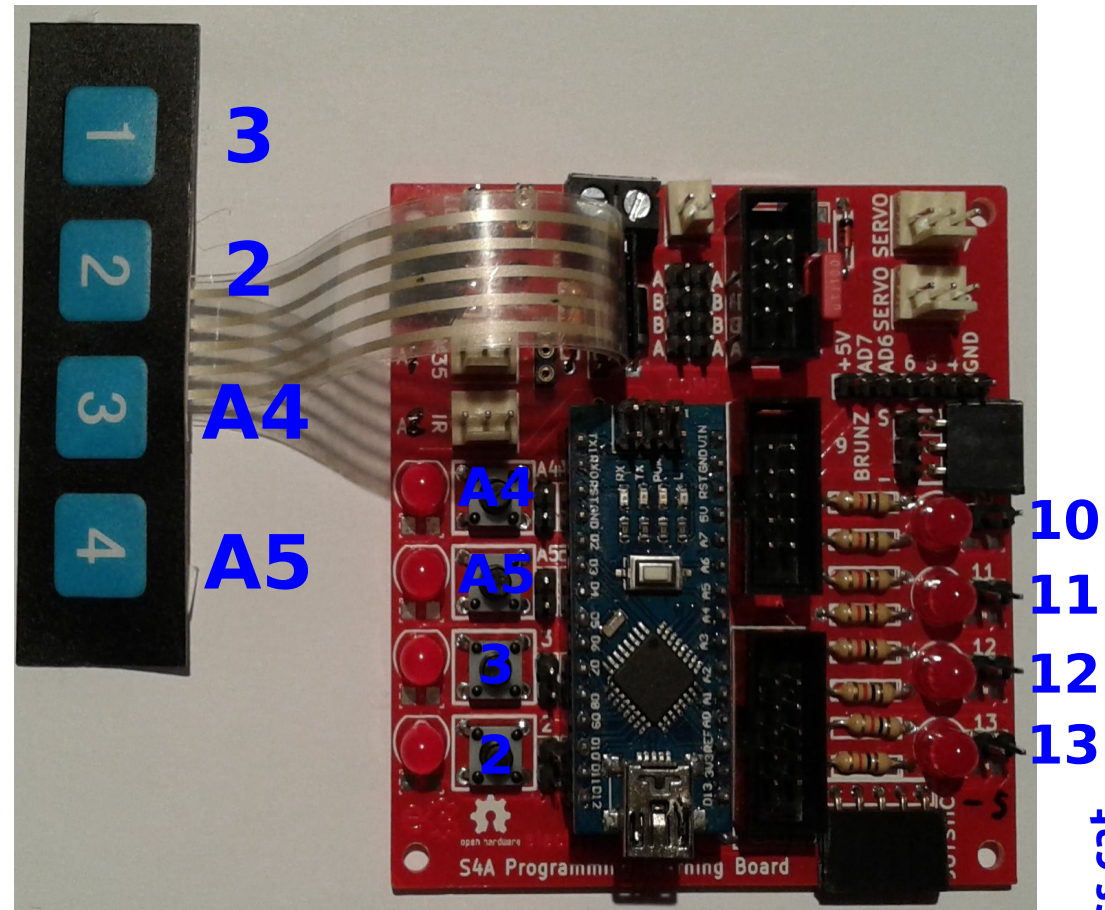
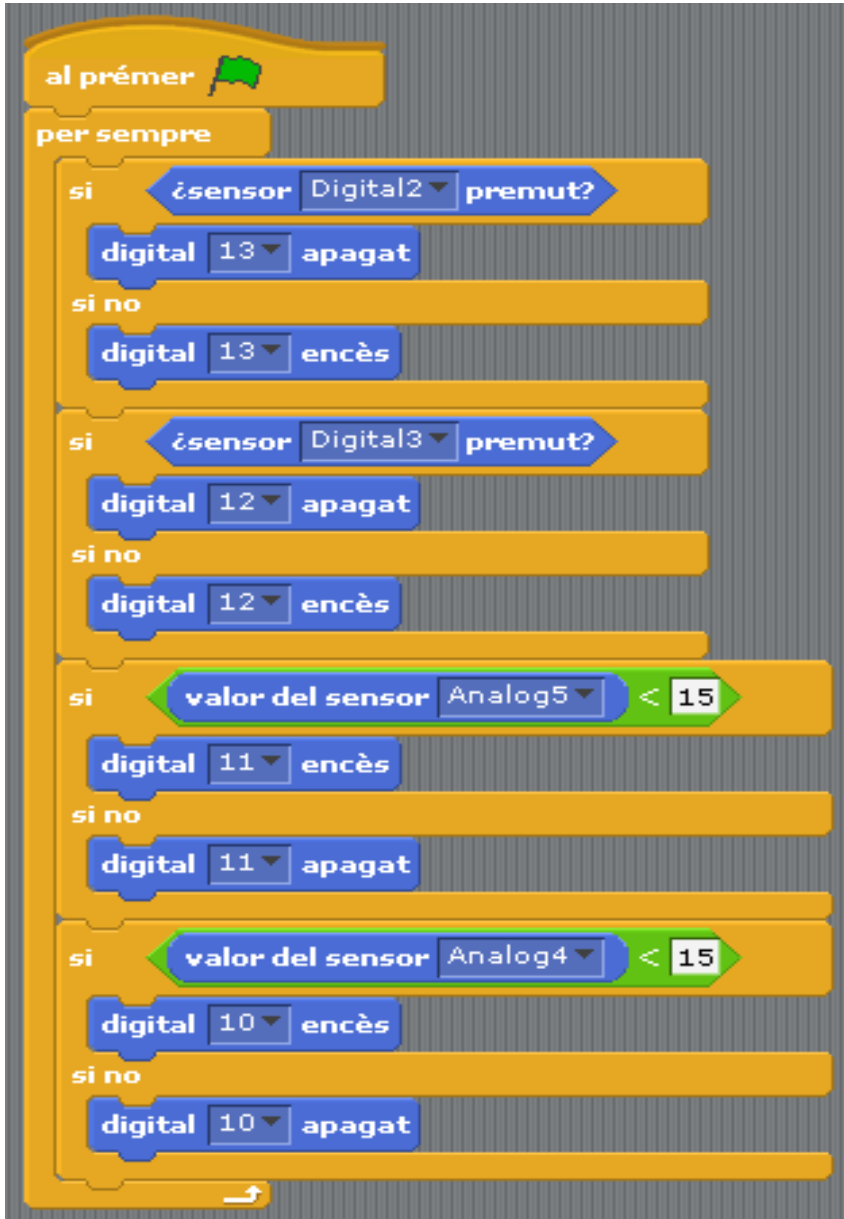
## Botons i leds



Codis d'exemple d'aquesta presentació per a l'S4A

# S4A Programming Learning Board

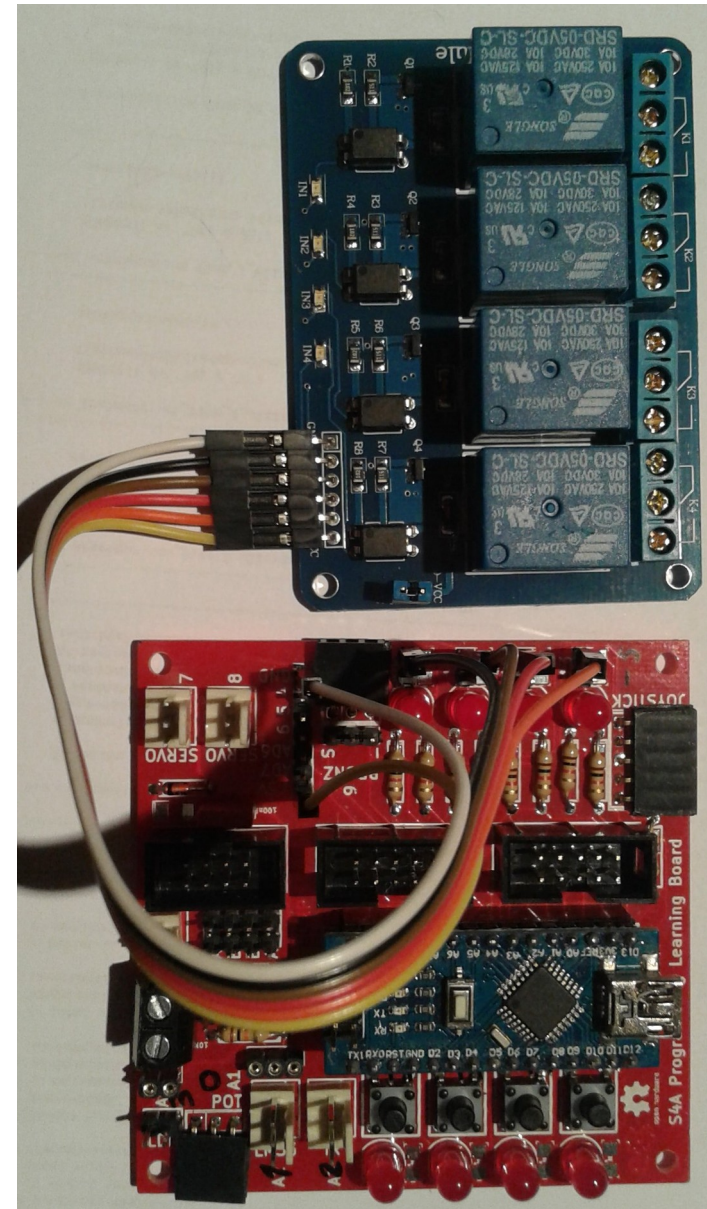
## Teclat de membrana i leds





# e⚙ S4A Progamming Learning Board

## Botons, leds i relés



# e⚙ S4A Progamming Learning Board

## Concepte de relé

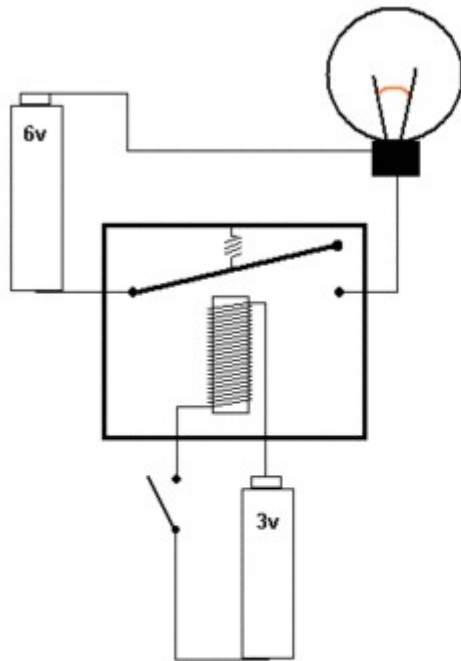


Figure 1: Relay off

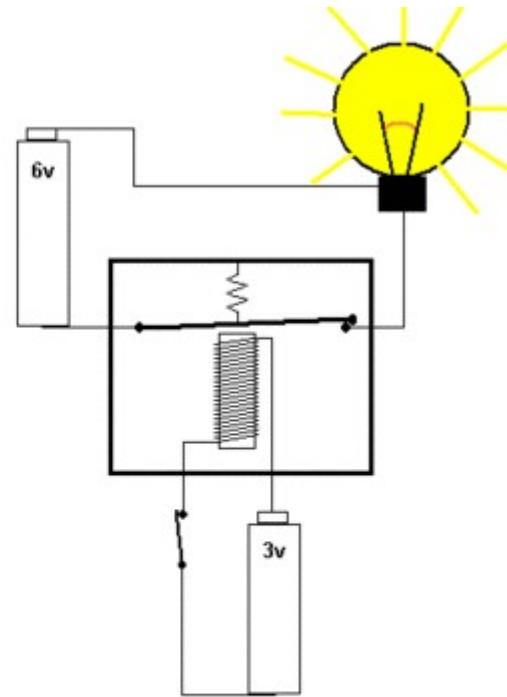
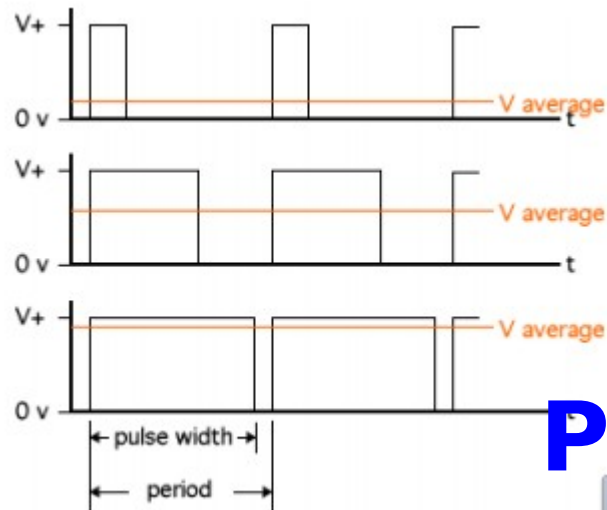


Figure 2: Relay on



# S4A Programming Learning Board

## Botons i brunzidor



A4

A5

3

2

PWM

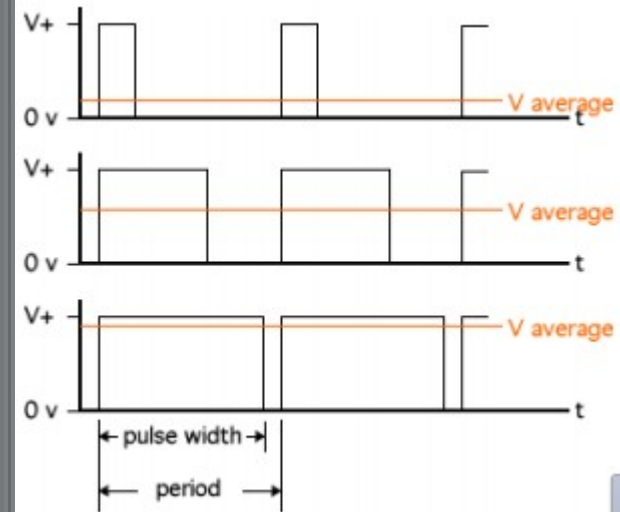
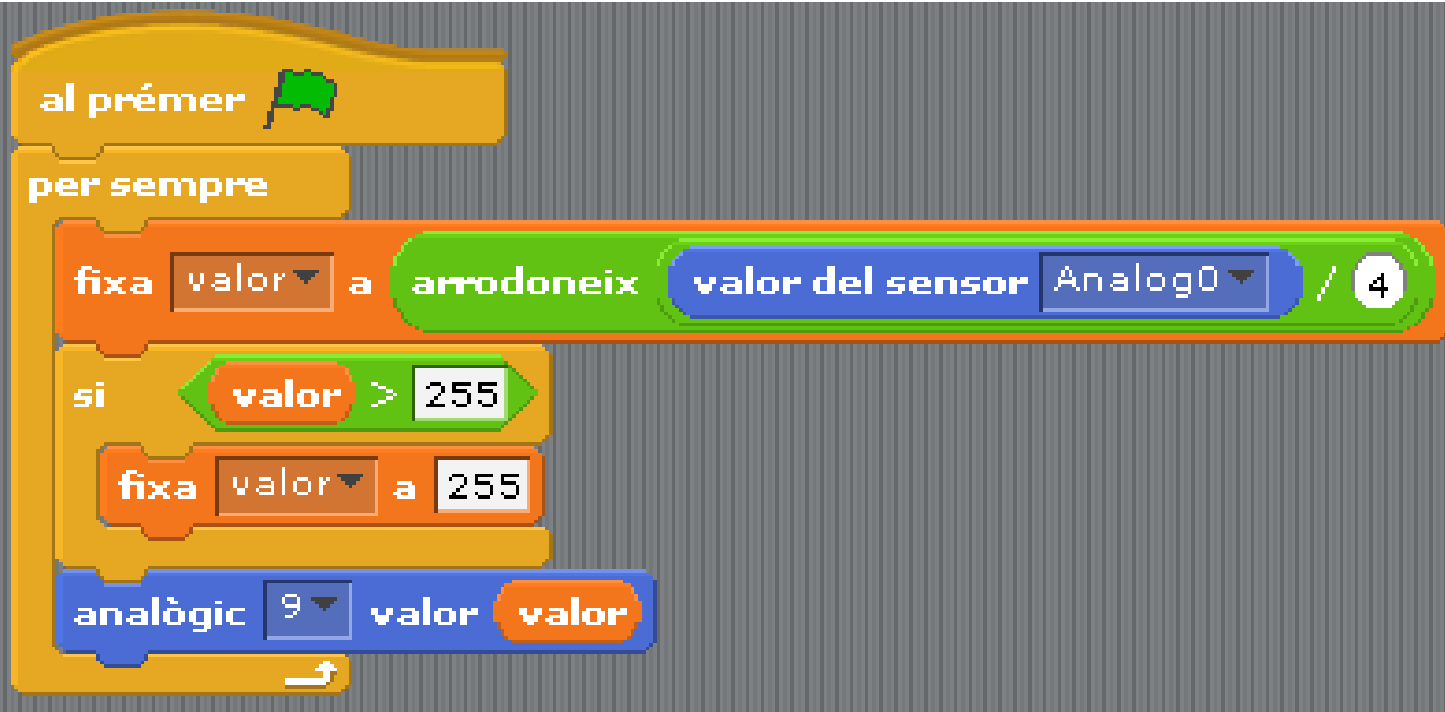


9

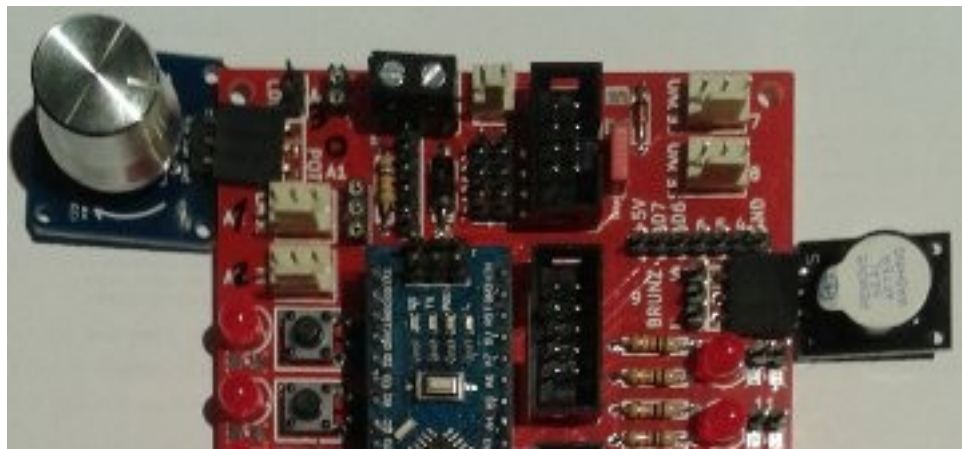


# S4A Programming Learning Board

## Potenciòmetre i brunzidor



A0

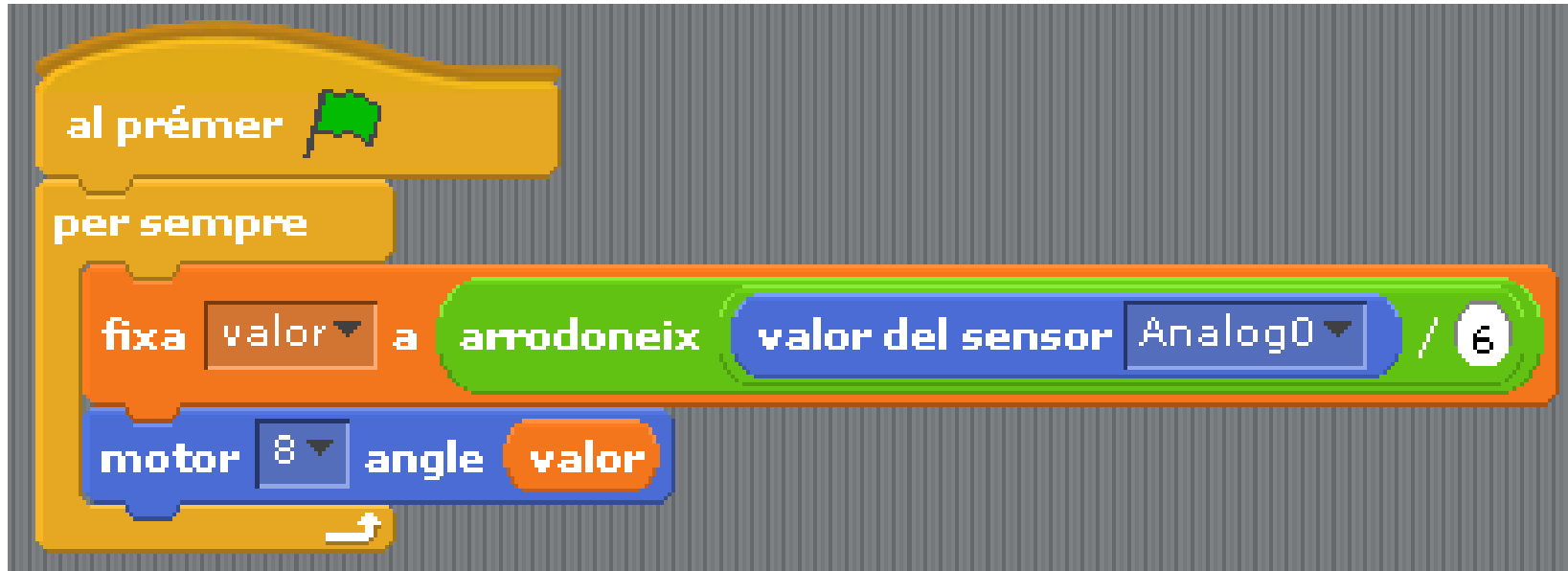


9

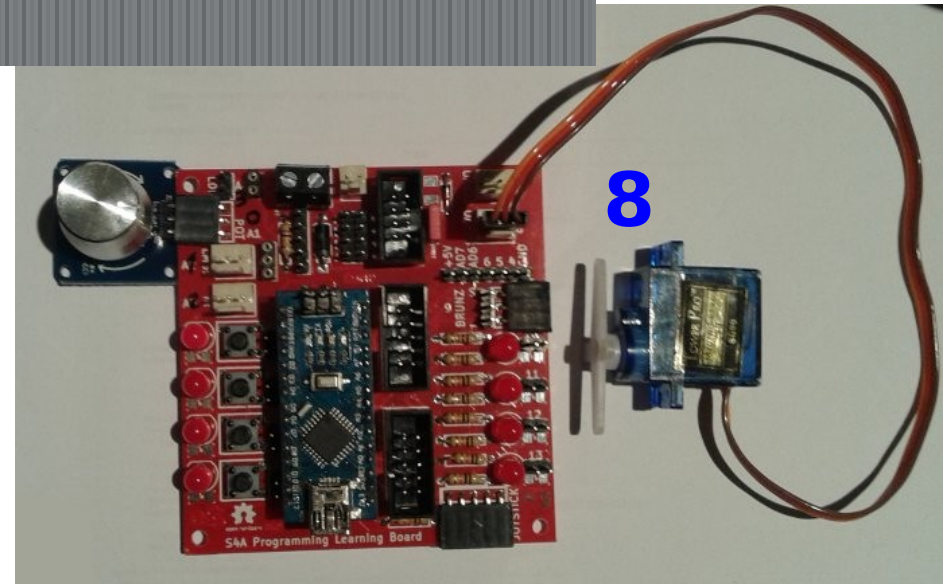


# S4A Programming Learning Board

## Potenciòmetre i servomotor



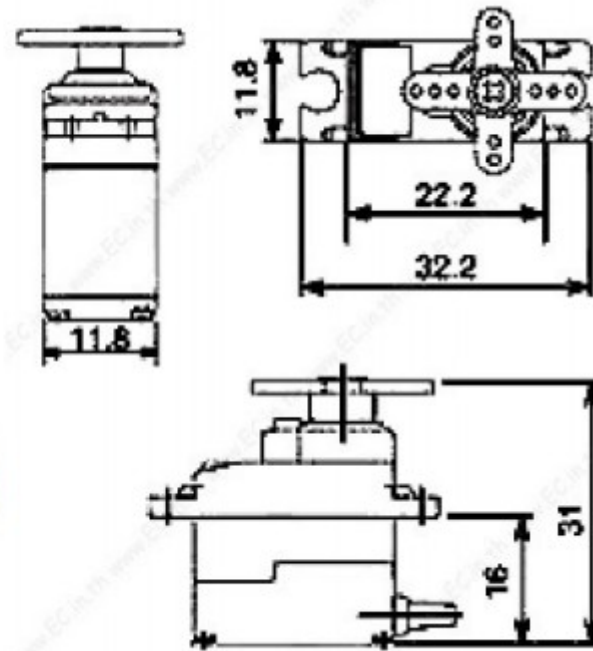
A0





# e⚙ S4A Progamming Learning Board

## Servo



### Specifications


- Weight: 9 g
- Dimension: 22.2 x 11.8 x 31 mm approx.
- Stall torque: 1.8 kgf·cm
- Operating speed: 0.1 s/60 degree
- Operating voltage: 4.8 V (~5V)
- Dead band width: 10  $\mu$ s
- Temperature range: 0 °C – 55 °C

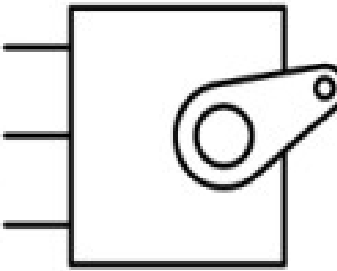
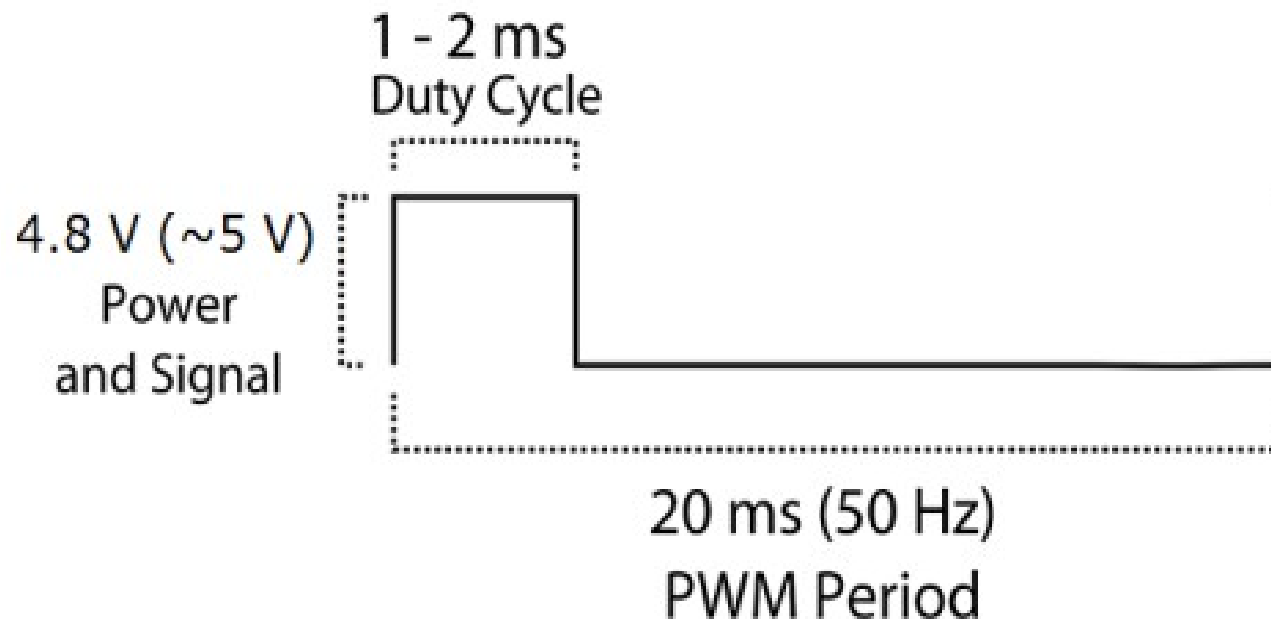
<http://datasheet.sparkgo.com.br/SG90Servo.pdf>



# S4A Progamming Learning Board

## Servo

PWM=Orange (  )  
 Vcc = Red ( + )  
 Ground=Brown ( - )

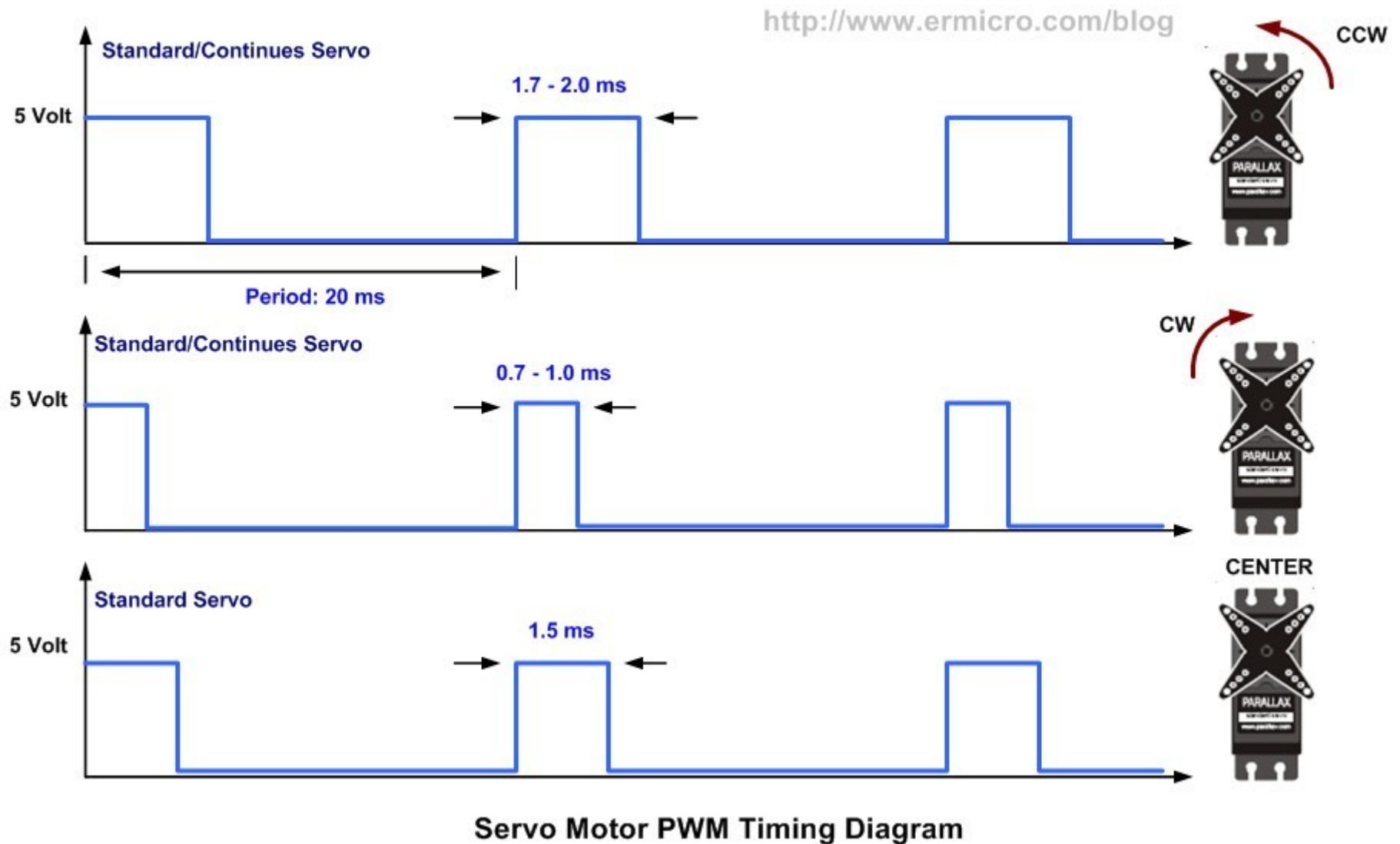



Position "0" (1.5 ms pulse) is middle, "90" (~2 ms pulse) is all the way to the right, "-90" (~1 ms pulse) is all the way to the left.

<http://datasheet.sparkgo.com.br/SG90Servo.pdf>

# eS4A Progamming Learning Board

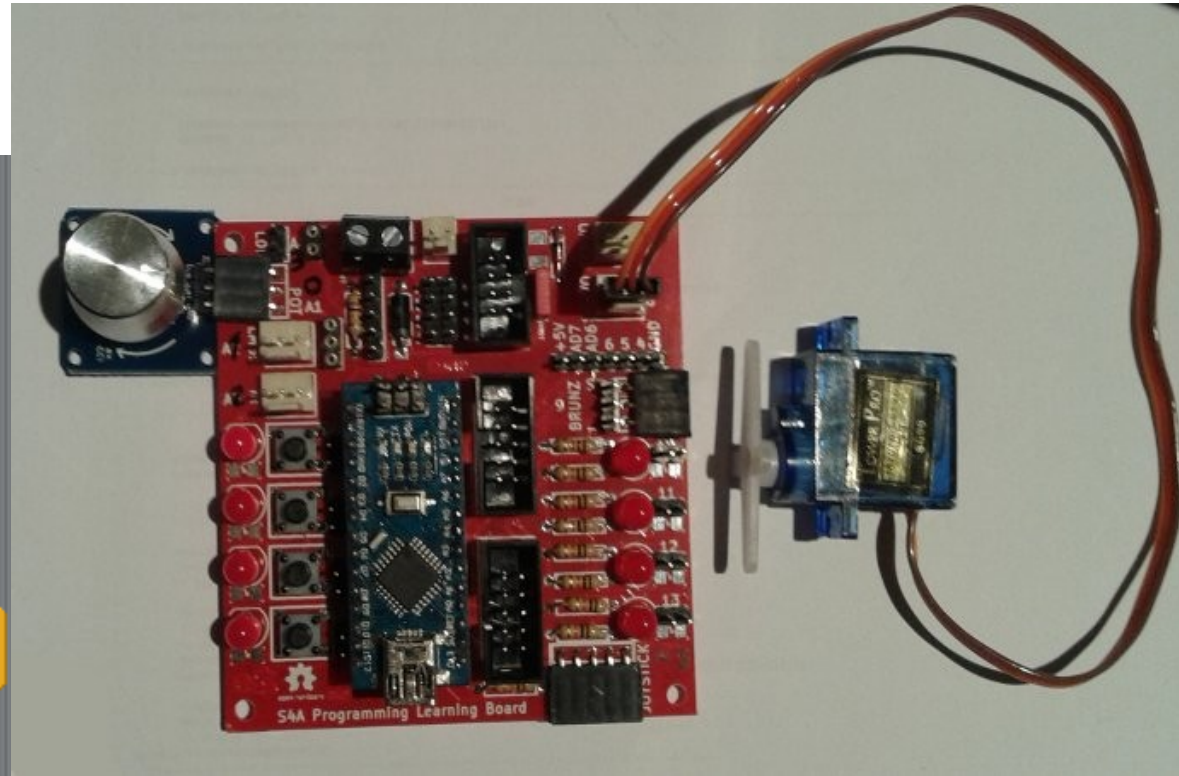
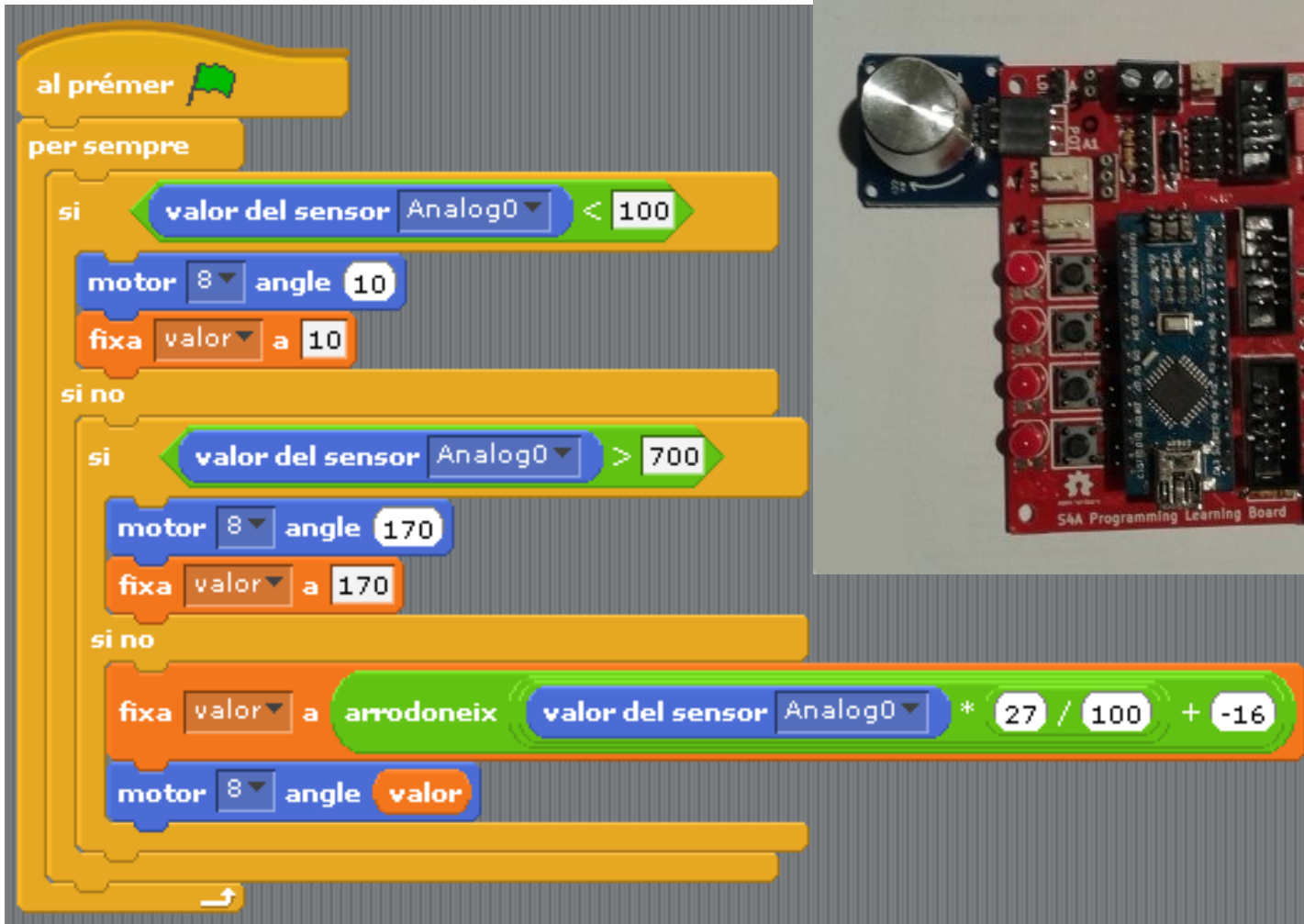
## Servo



[http://www.ermicro.com/blog/wp-content/uploads/2009/02/servo\\_01.jpg](http://www.ermicro.com/blog/wp-content/uploads/2009/02/servo_01.jpg)

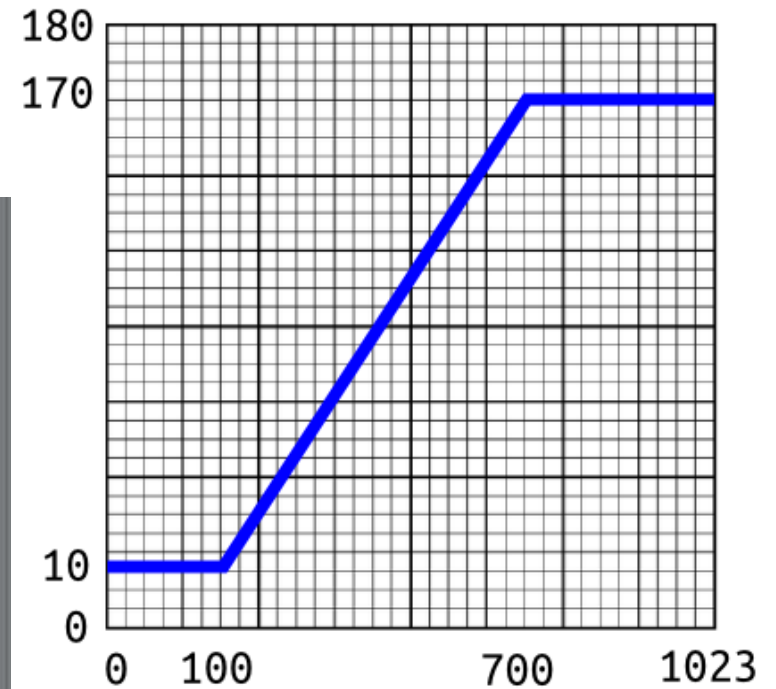
# S4A Programming Learning Board

## Potenciòmetre i servomotor



# S4A Progamming Learning Board

## Potenciòmetre i servomotor

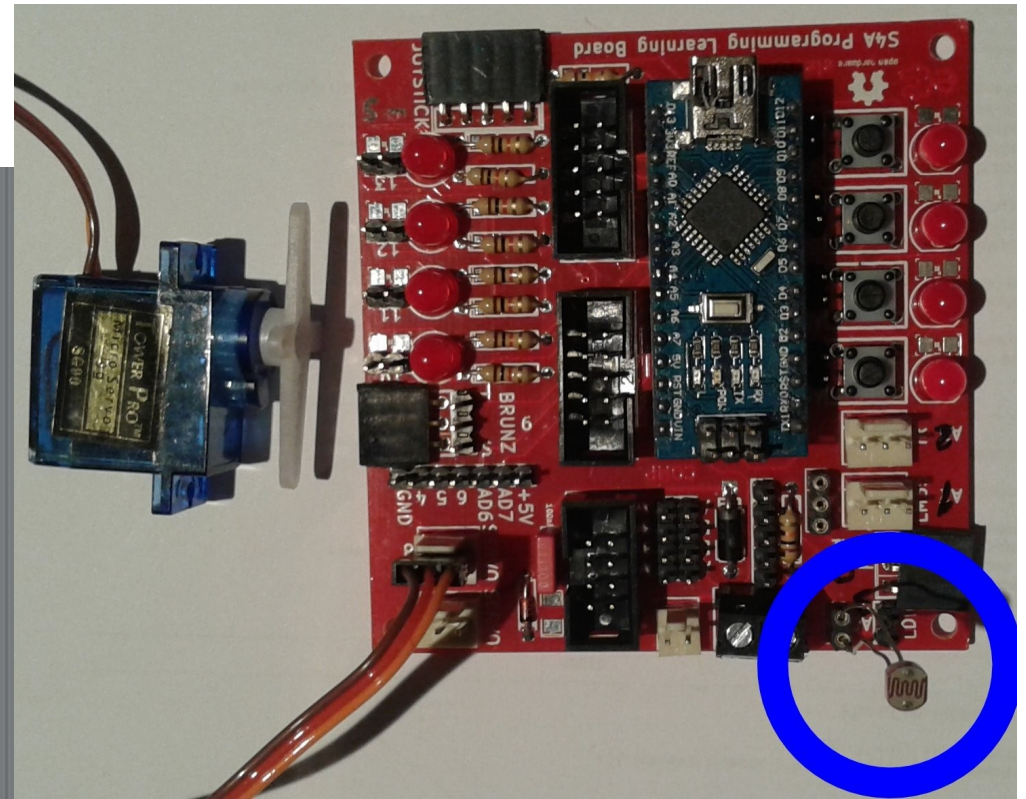
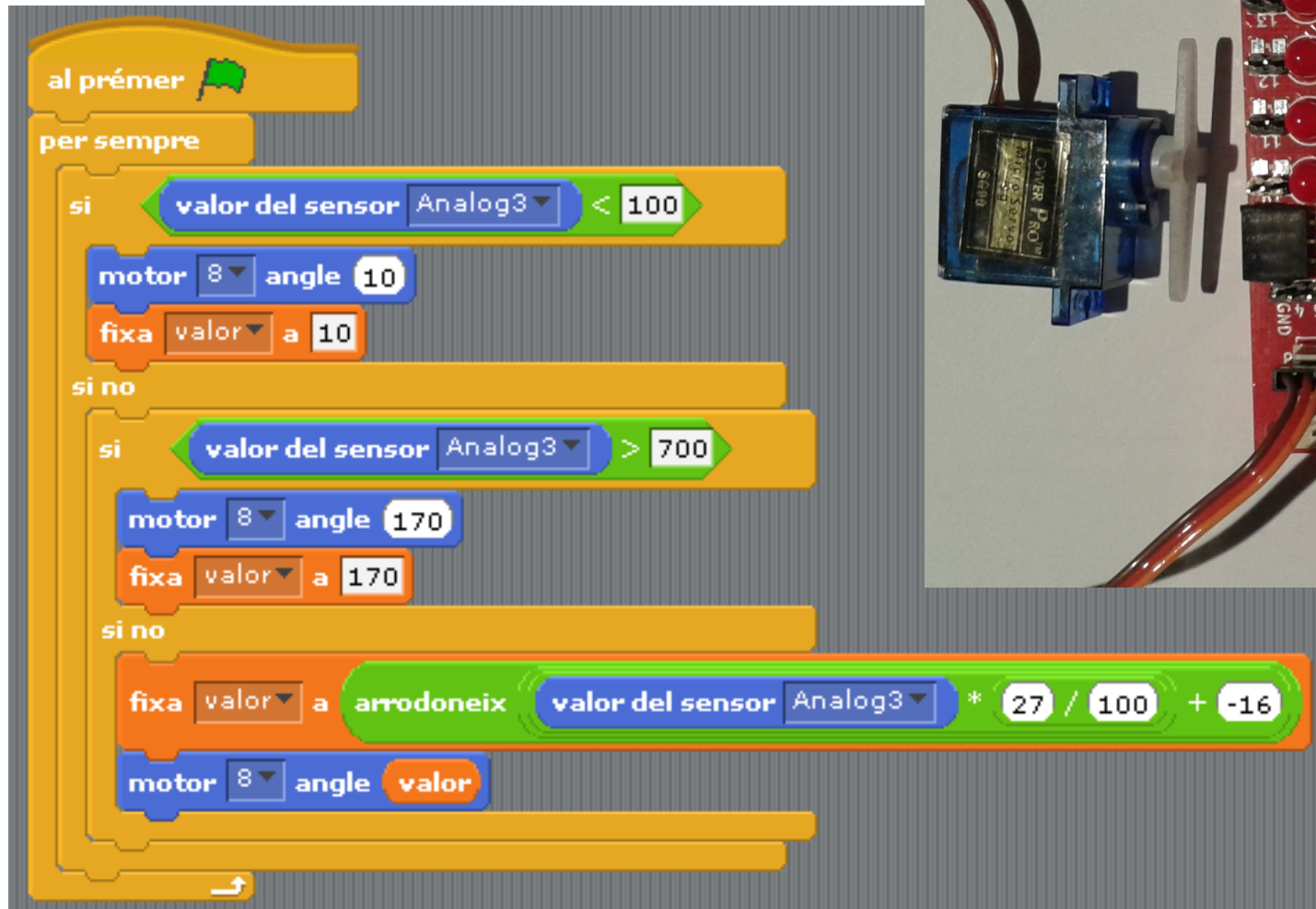


$$\text{graus} = \text{Analog0} \frac{27}{100} - 16$$



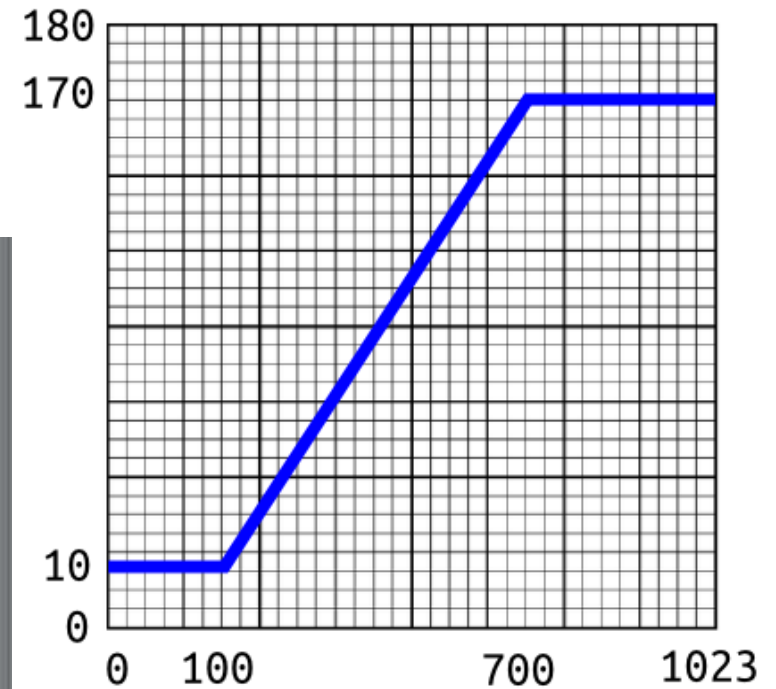
# S4A Programming Learning Board

## LDR i servomotor



# S4A Progamming Learning Board

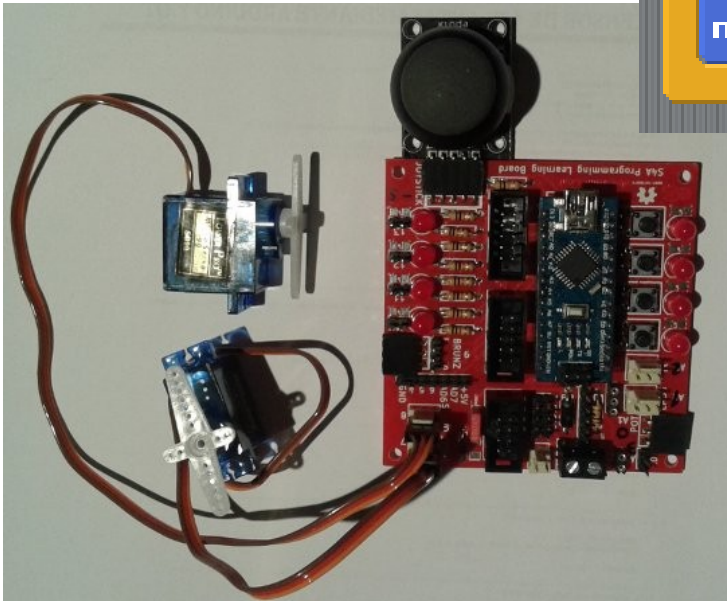
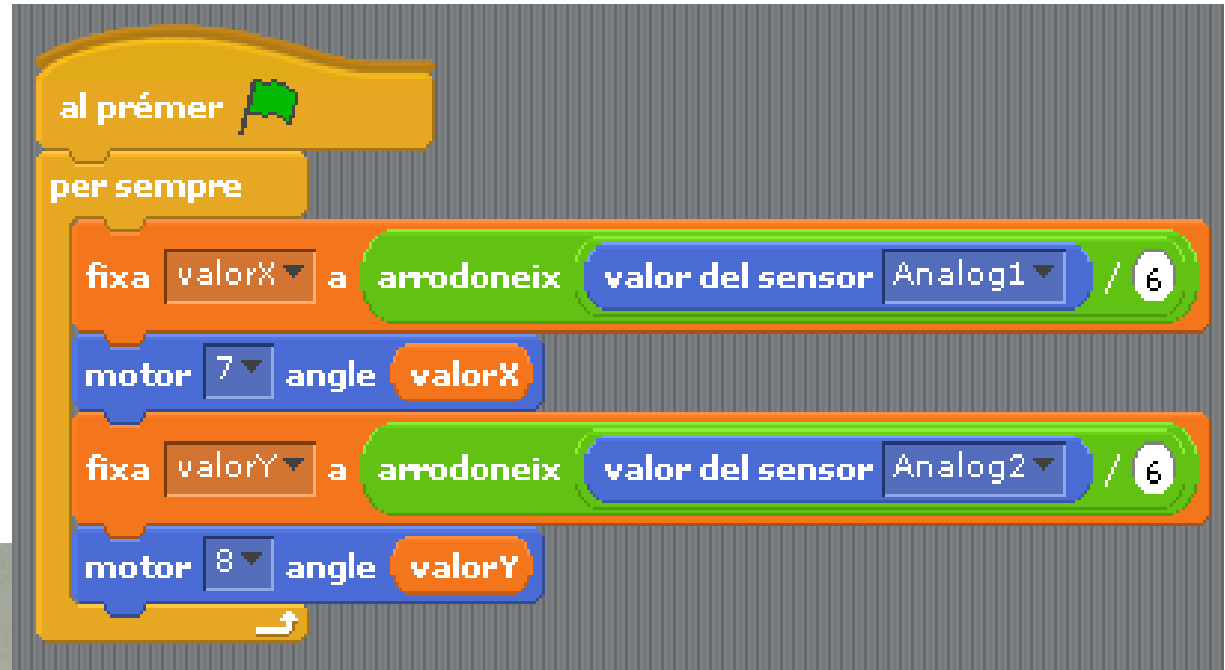
## LDR i servomotor



$$\text{graus} = \text{Analog0} \frac{27}{100} - 16$$

# eS4A Programming Learning Board

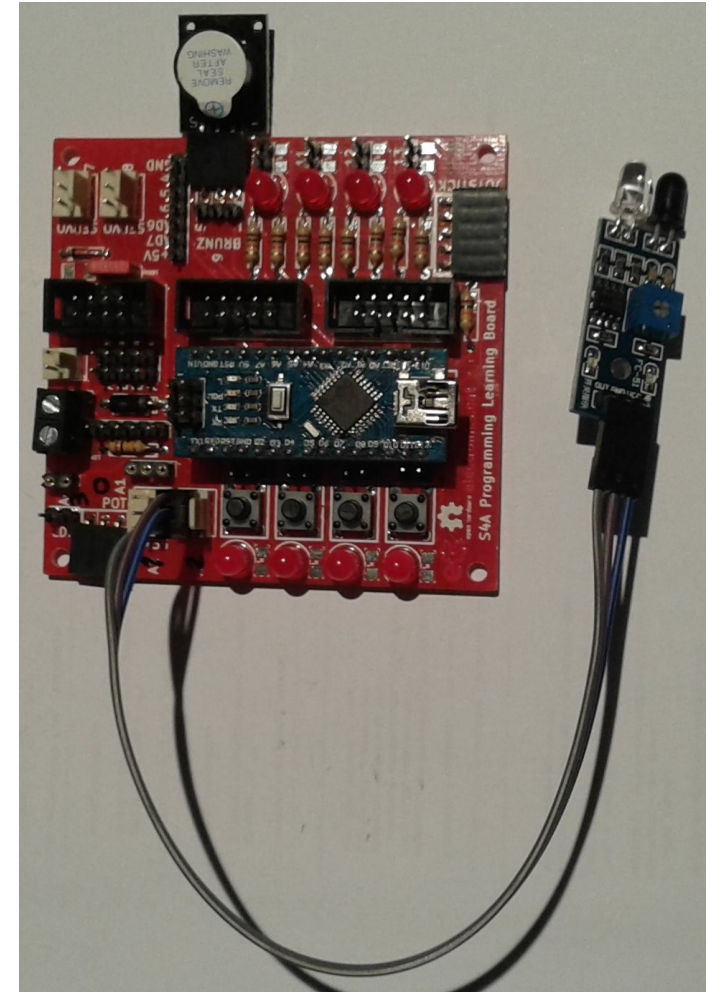
## Joystick i servomotors



Per a fer servir el joystick desconnecteu el sensor de temperatura i el sensor infraroig

# S4A Programming Learning Board

## Sensor d'infrarojos i brunzidor

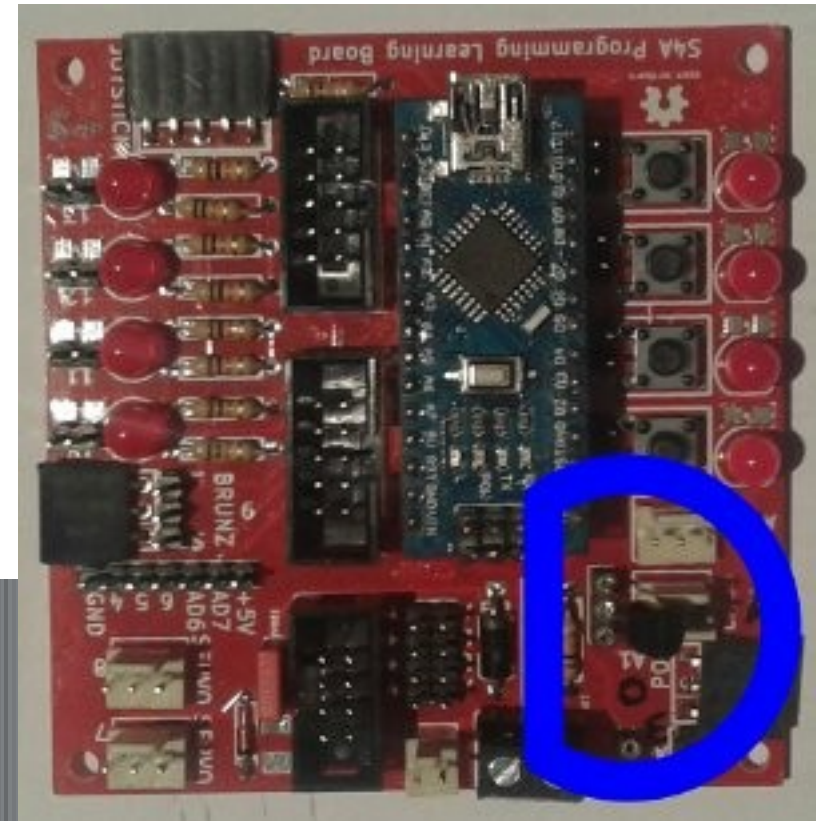


Per a fer servir el sensor infraroig desconnecteu el joystick



# S4A Progamming Learning Board

## Sensor de temperatura LM35



Per a fer servir el sensor de temperatura desconnecteu el joystick

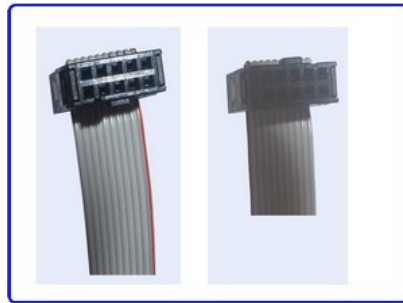


# El microcontrolador Arduino

**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)

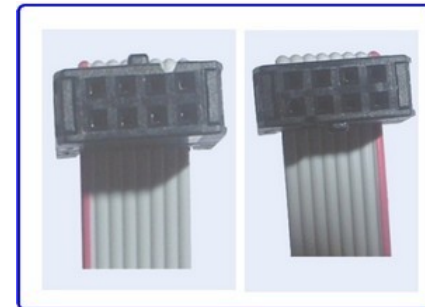


CTS / RTS

TX / RX

RX / TX

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



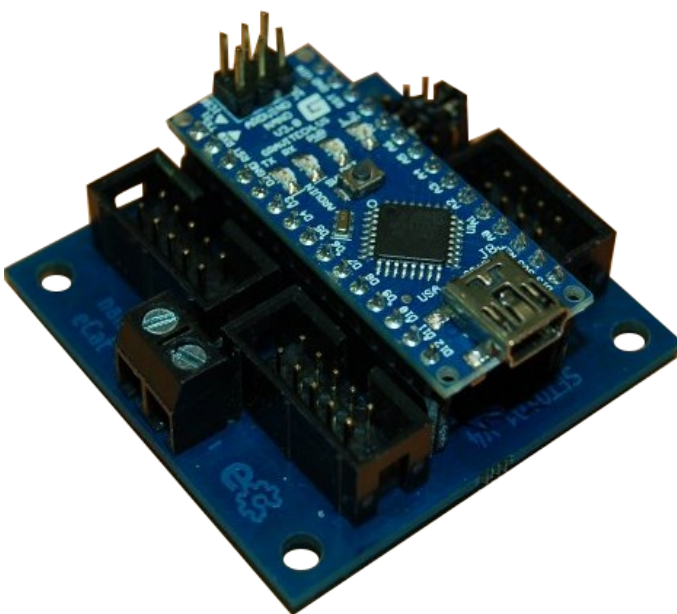
**eCat System** : Dues tipologies de BUS, dades (8 bits) i comunicacions



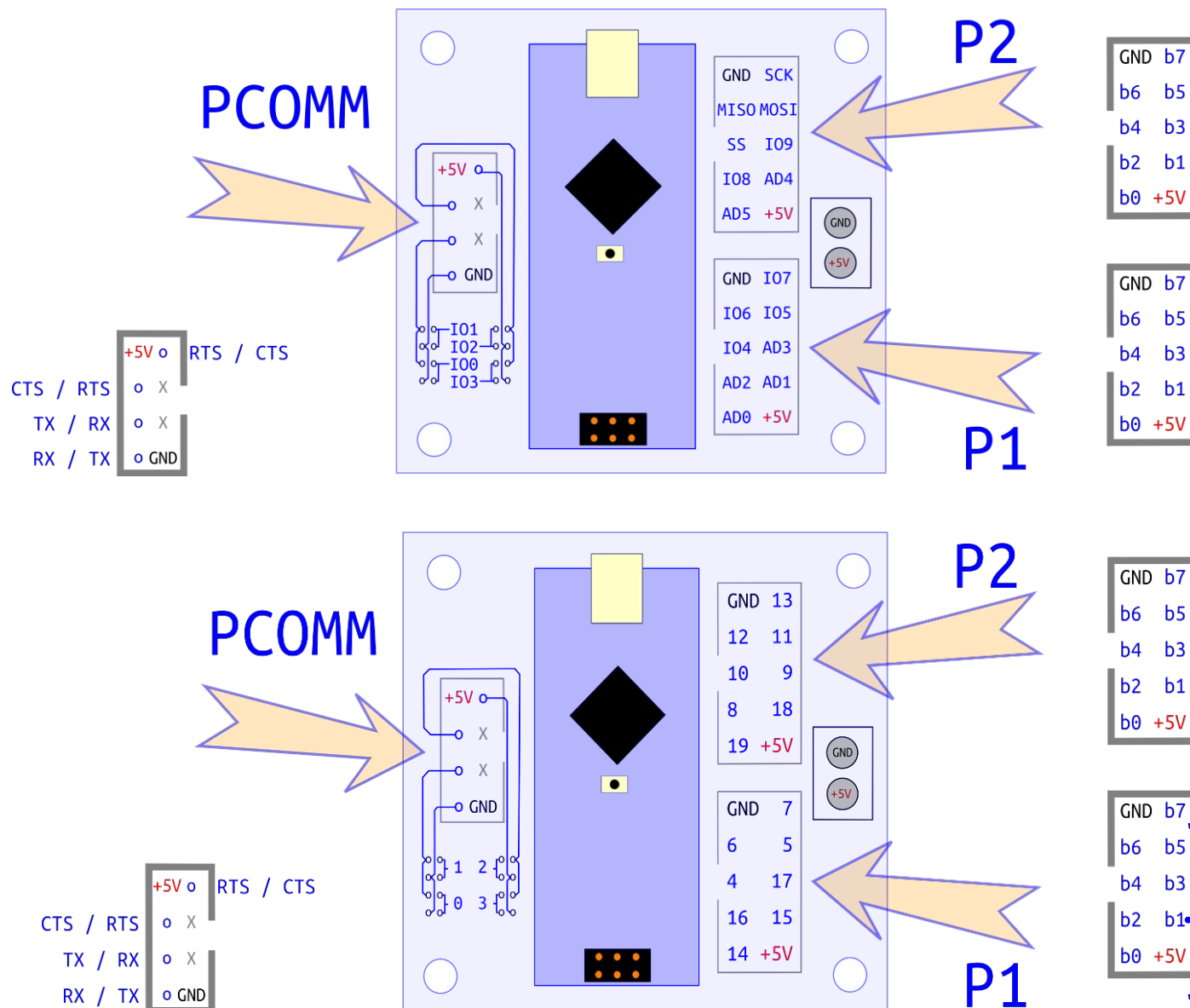
# El microcontrolador Arduino

## nano-eCat

Placa 0101 v4

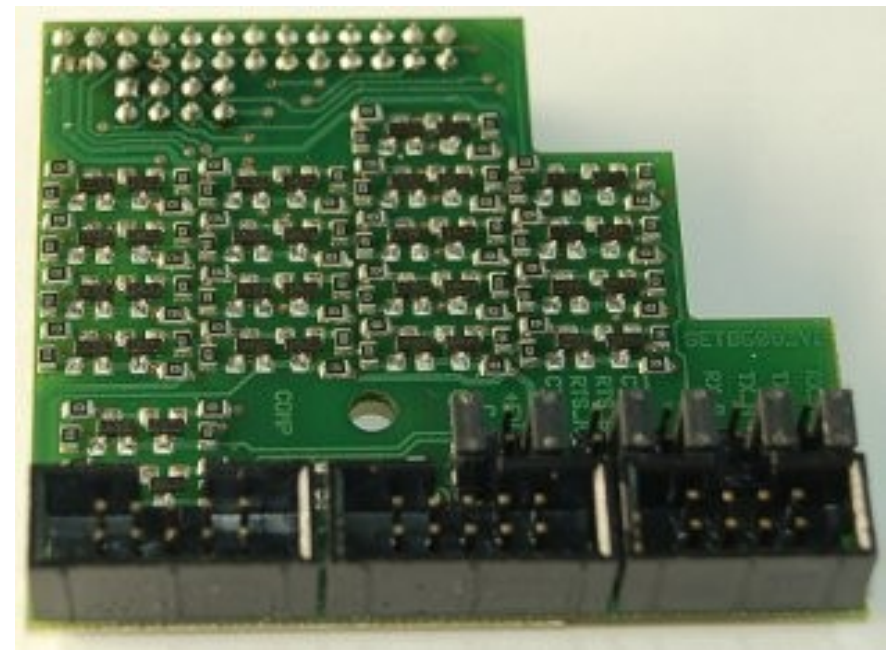


Placa convertidora de  
l'Arduino Nano al  
sistema de bus lliure  
eCat System



# El microcontrolador Arduino

## Nivells de tensió



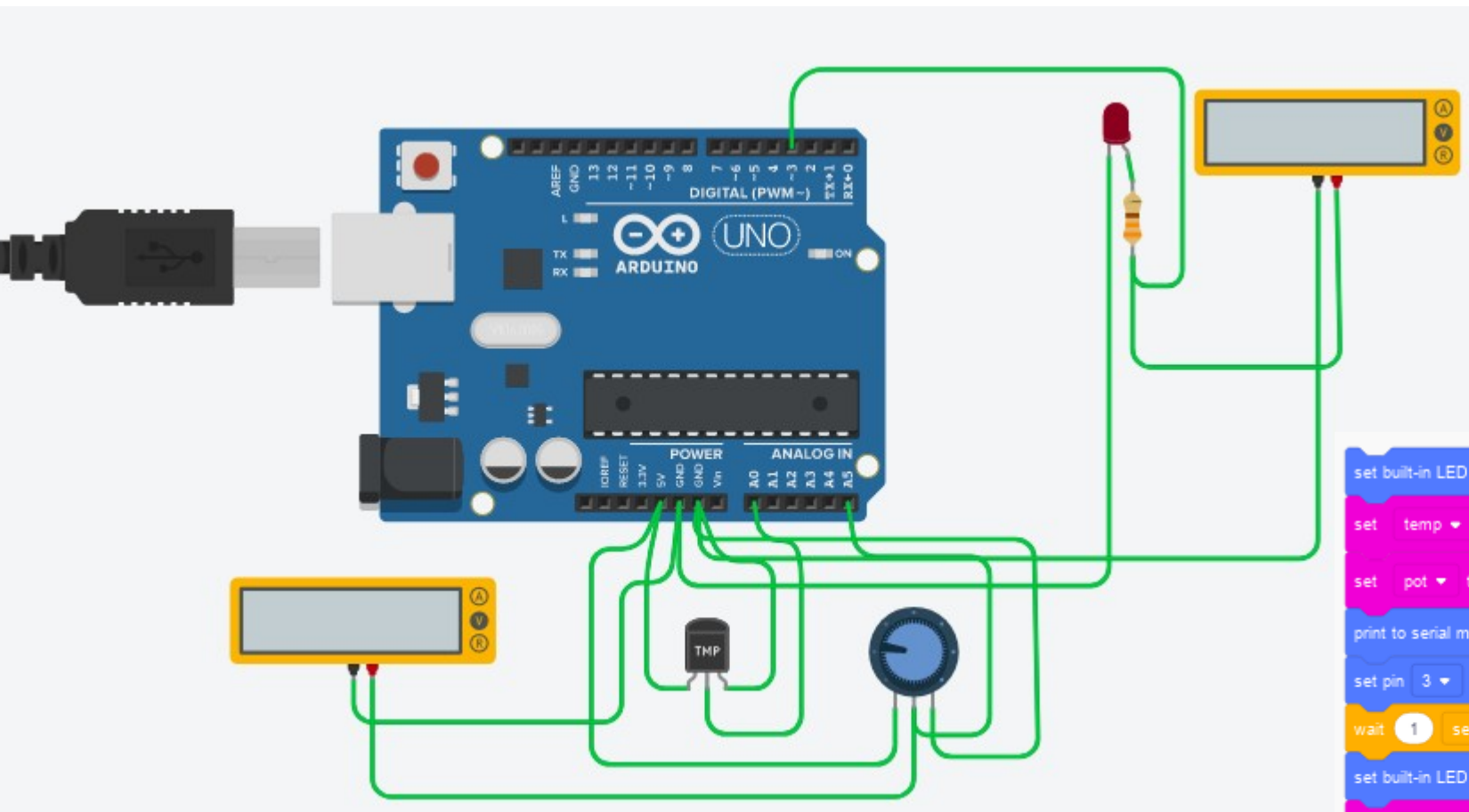
<http://electronics.cat/php/common/index.php?lang=ca&page=508>





# Maneres de programar un Arduino

## Simulació per blocs a TinkerCad



```

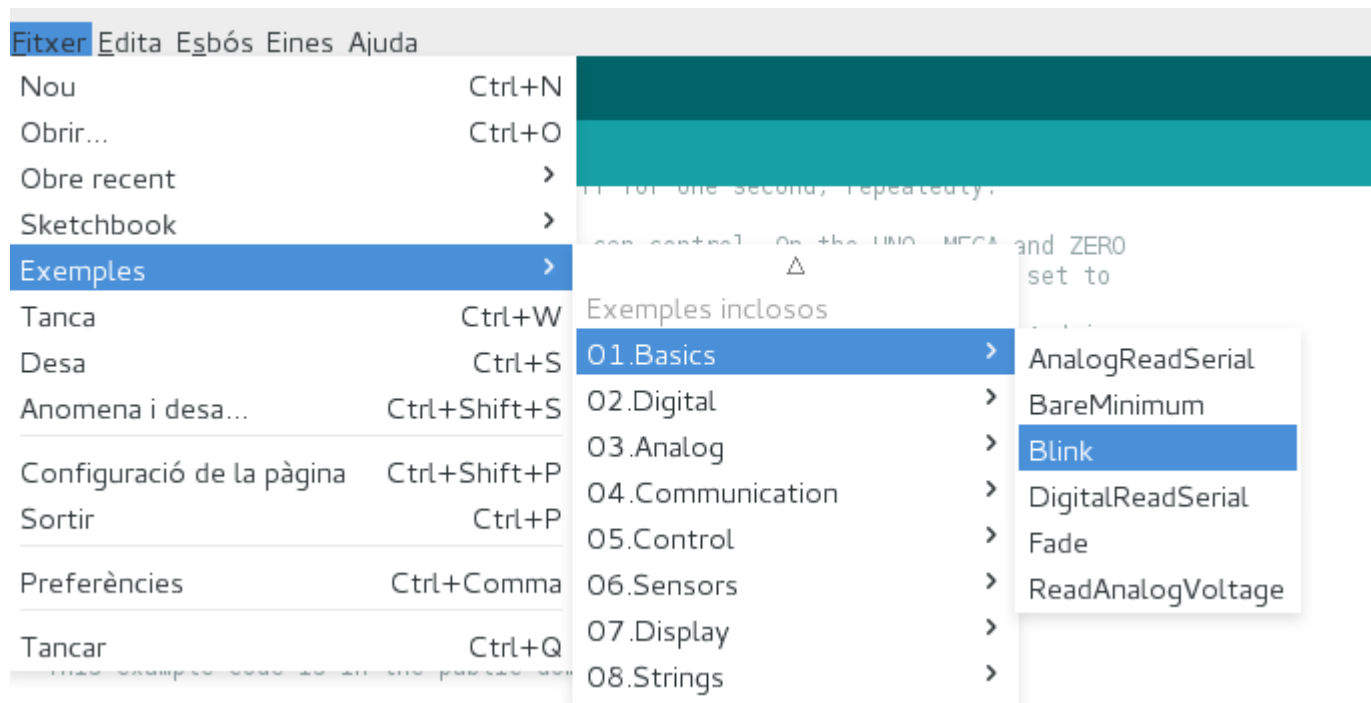
set built-in LED to HIGH
set temp to read analog pin A0 / 4
set pot to read analog pin A5 / 4
print to serial monitor temp with newline
set pin 3 to pot
wait 1 secs
set built-in LED to LOW
set temp to read analog pin A0 / 4
set pot to read analog pin A5 / 4
print to serial monitor temp with newline
wait 1 secs
set pin 3 to pot
  
```

<https://www.tinkercad.com/>



# Maneres de programar un Arduino

## Des de l'IDE d'Arduino





# Connexió de perifèrics a l'Arduino

## Entrades i sortides digitals

```
// basic_05_flanc.ino
#define BOT0 2
#define TEMPS_REBOTS_EN_ms 20

void setup(){
    Serial.begin(9600);
    pinMode(BOT0,INPUT);
    Serial.println("Quants cops he premut ...");
}

void loop(){
    static bool bPrevi = HIGH;
    bool bAra = digitalRead(BOT0);
    static int nCmpt = 0;

    if(bAra != bPrevi){
        // Hi ha canvi
        delay(TEMPS_REBOTS_EN_ms);
        if(!bAra){ // if (bAra == LOW){
            // Serial.println("Flanc de baixada detectat.");
            Serial.print("He premut ");
            nCmpt++;
            Serial.print(nCmpt);
            Serial.println(" cops");
        }
        bPrevi = bAra;
    }
}
```

Exemples d'entrades / sortides digital i flancs

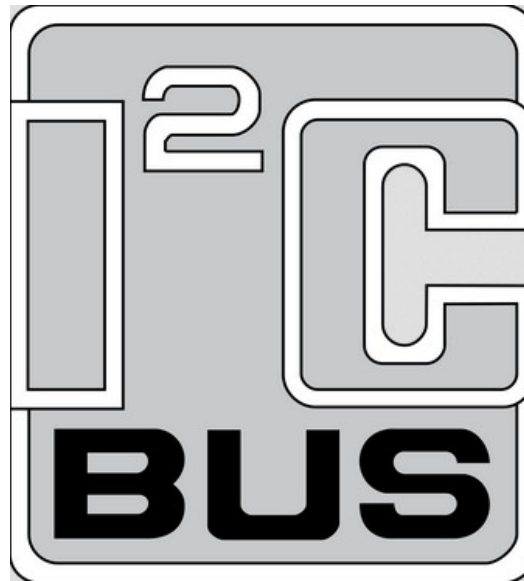


# Connexió de perifèrics a l'Arduino

## SPI, I2C, UART

### SPI

Serial Peripheral Interface



### UART Bus

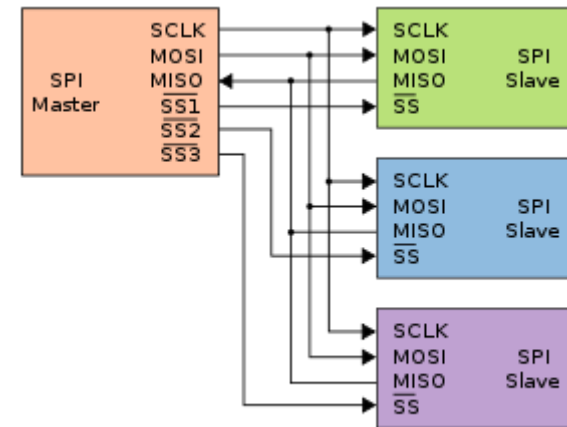
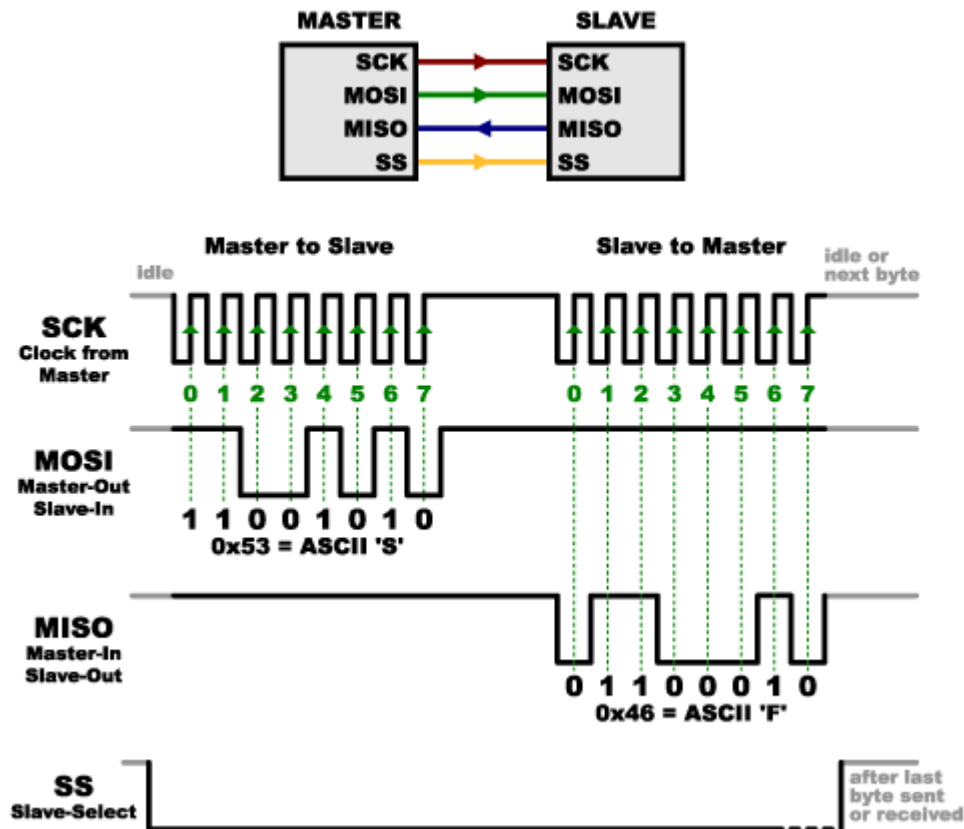
Universal asynchronous receiver-transmitter





# Connexió de perifèrics a l'Arduino

## SPI

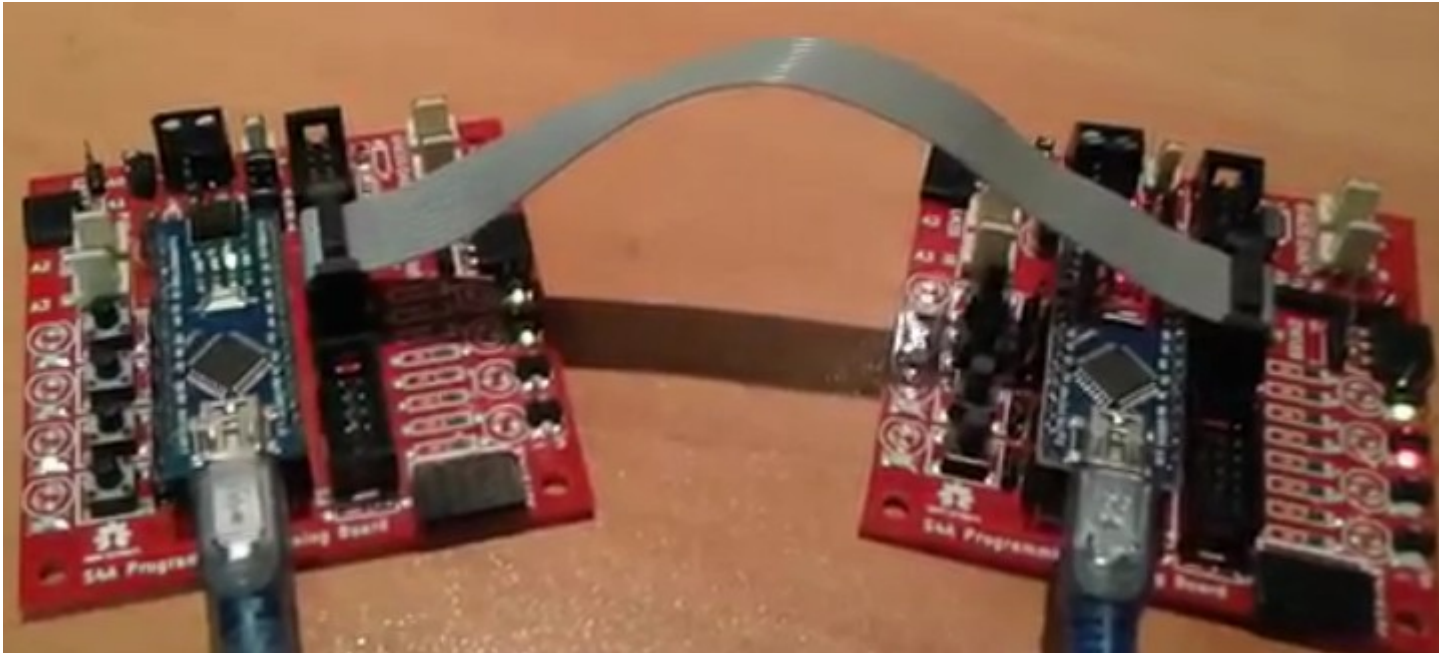


Explicació del bus SPI



# Connexió de perifèrics a l'Arduino

## SPI



Codis d'Arduino per al mestre i l'esclau

Vídeo de funcionament



# Connexió de perifèrics a l'Arduino

## 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?
```

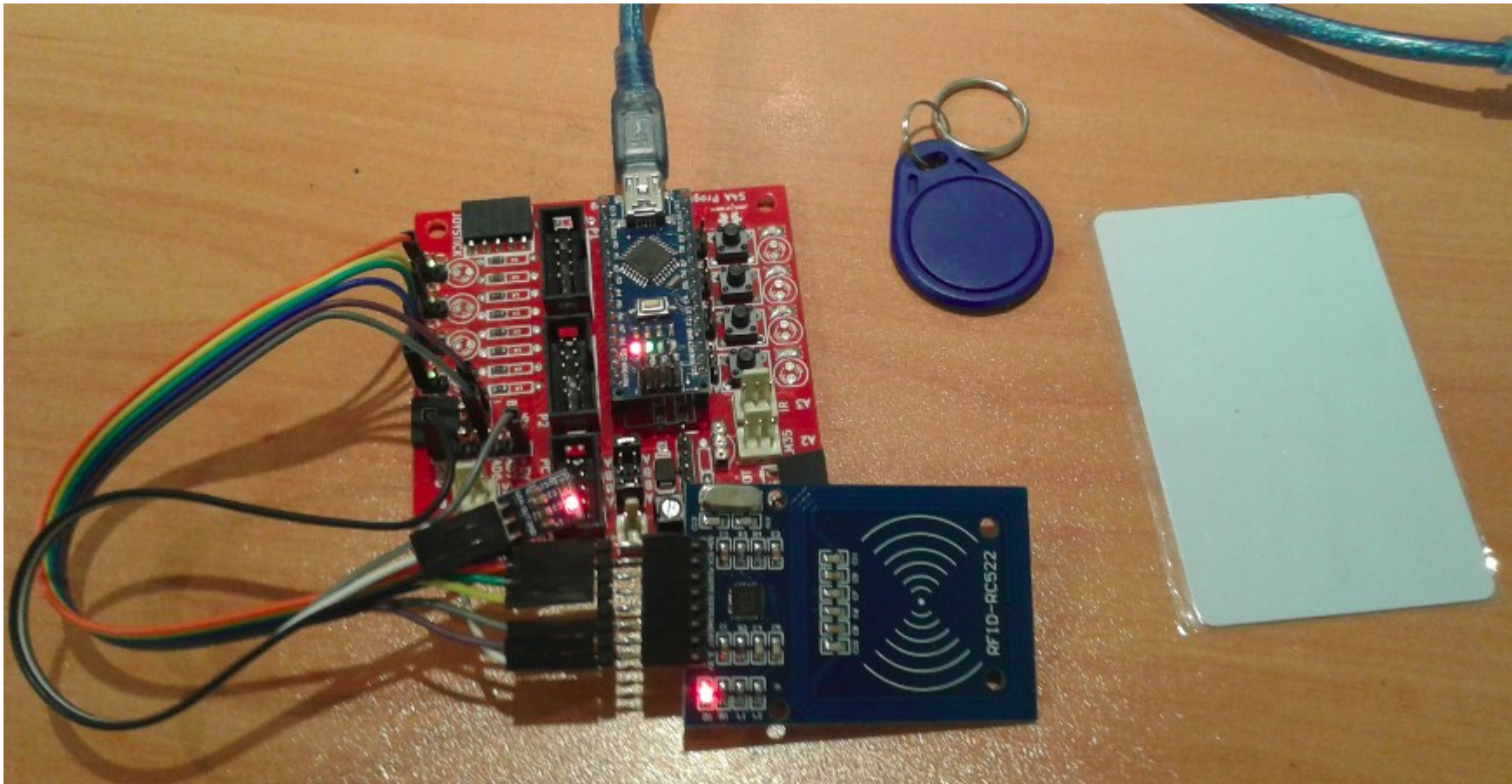
```
void loop() {
  // Slave Enabled?
  if (!digitalRead(SS)) {
    // Yes, first time?
    if (SSlast != LOW) {
      // Yes, take MISO pin.
      pinMode(MISO, OUTPUT);
      Serial.println("***Slave Enabled.");
      // Write -1 slave response code and receive master command code
      byte rx = SPI.transfer(255);
      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) + ".");
      }
    }
  }
}
```

Codis d'Arduino per al mestre i l'esclau  
Vídeo de funcionament



# Connexió de perifèrics a l'Arduino

## SPI



Com connectar el mòdul RFID-RC522 a Arduino





# Connexió de perifèrics a l'Arduino

## SPI

```
/dev/ttyUSB0
| Envia

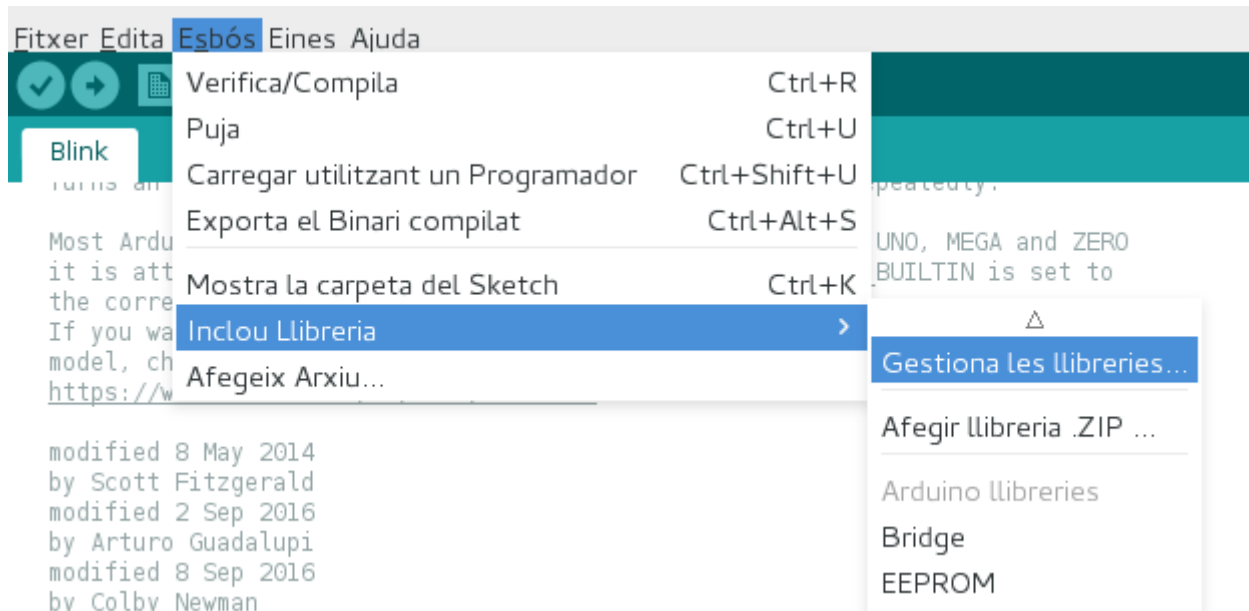
This code scan the MIFARE Classsic NUID.
Using the following key: FF FF FF FF FF FFPICC type: MIFARE 1KB
A new card has been detected.
The NUID tag is:
In hex: 35 07 10 22
In dec: 53 07 16 34
PICC type: MIFARE 1KB
A new card has been detected.
The NUID tag is:
In hex: F1 45 25 61
In dec: 241 69 37 97

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



# Connexió de perifèrics a l'Arduino

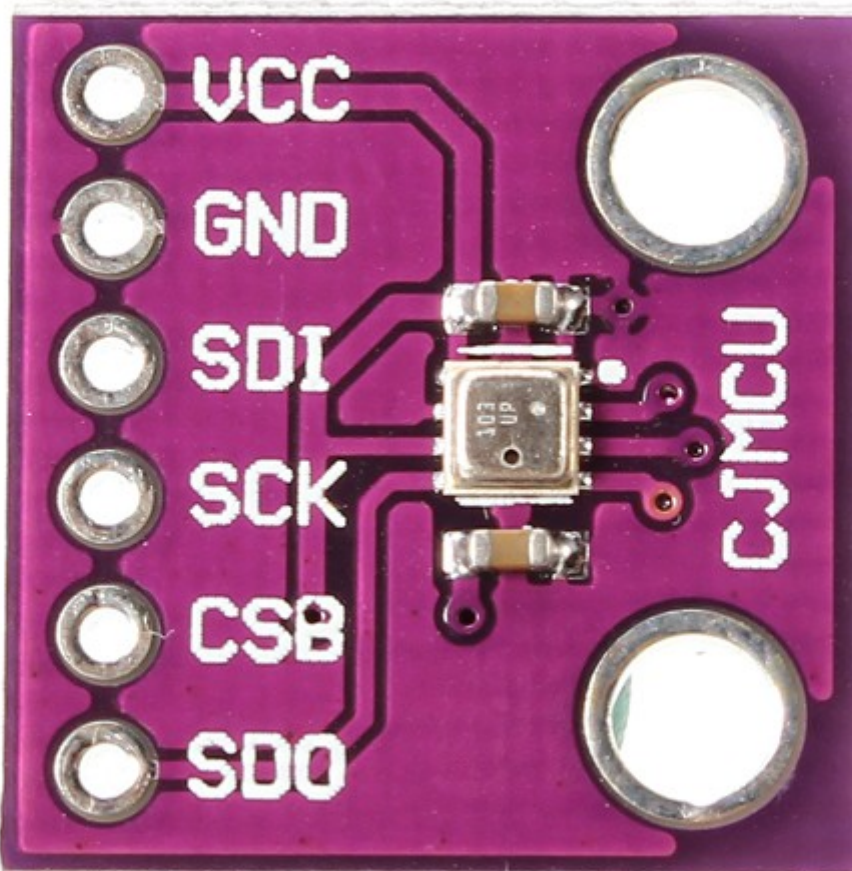
## Afegint biblioteques a l'IDE d'Arduino





# Connexió de perifèrics a l'Arduino

## BME 280 funcionant amb SPI



```
Mode SPI:
bme280.parameter.communication = 2;

VCC VCC
GND GND
SCL SCK      (13)
SDA MOSI     (11)
CSB CS       (10) // bme280.parameter.SPI_cs = 2; <-- Canvieu el 2 per un 10
SDO MISO     (12)
```

P2

GND	SCK
MISO	MOSI
CS	x
x	A4
A5	+5V

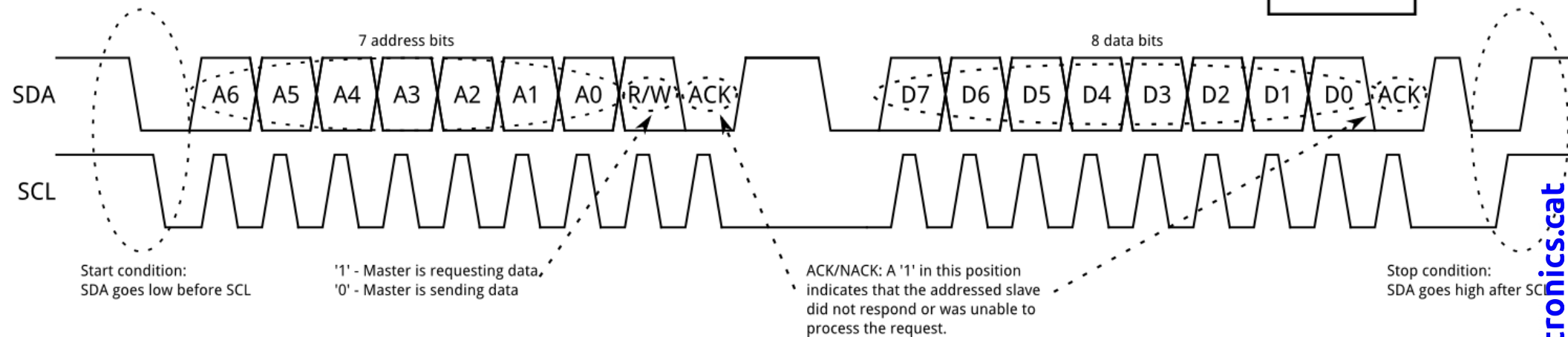
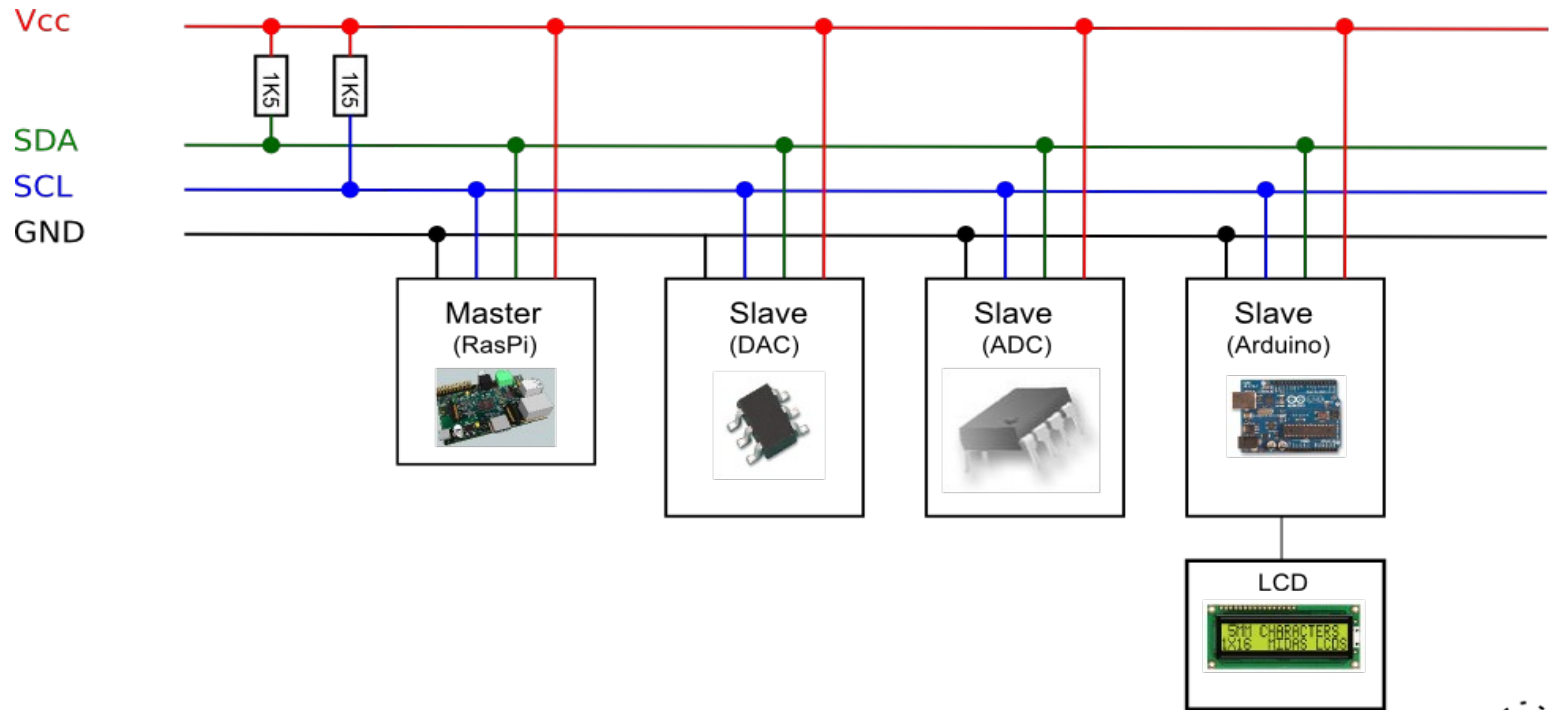
P2

GND	13
12	11
10	9
8	18
19	+5V



# Connexió de perifèrics a l'Arduino

## I2C

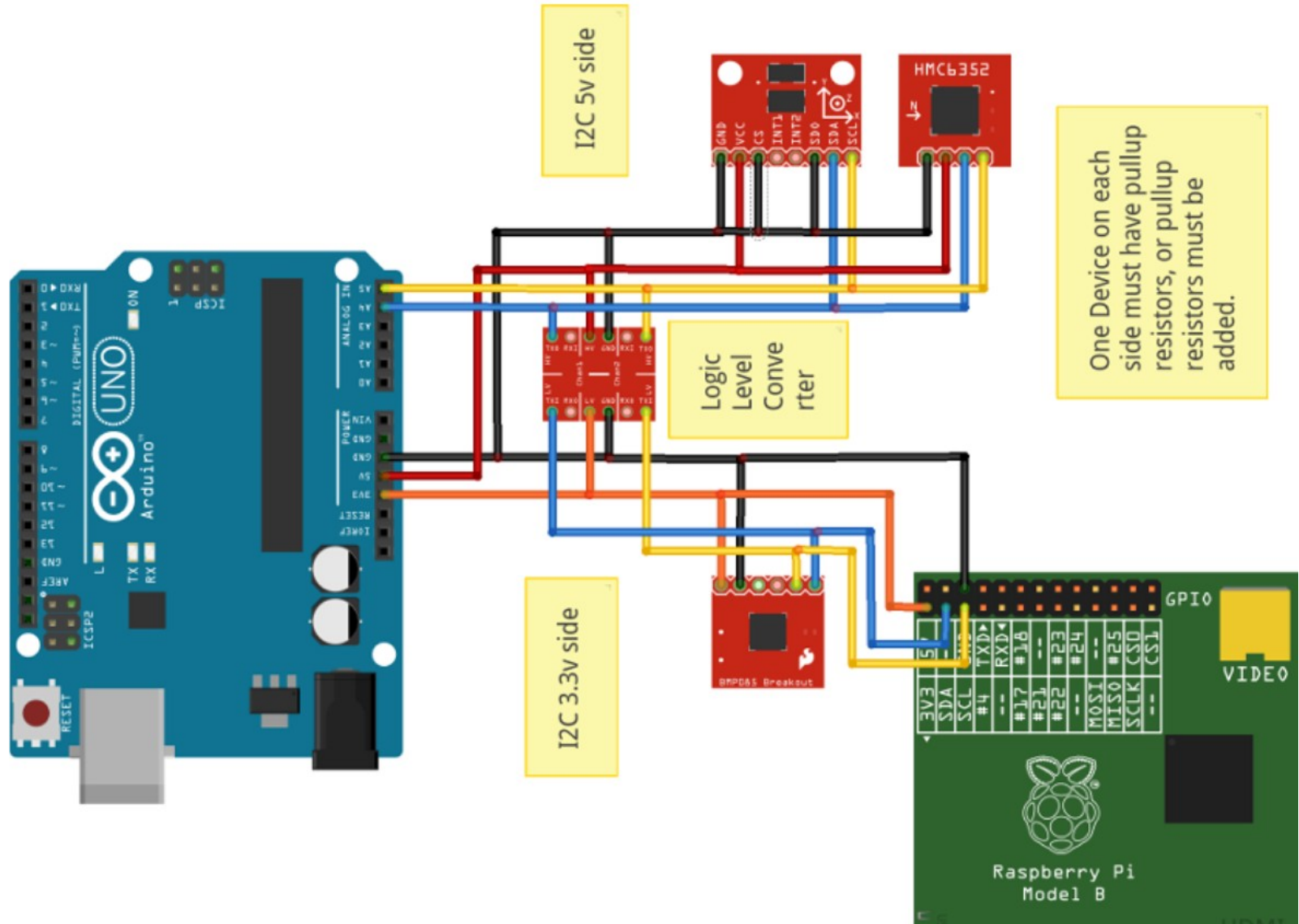






# Connexió de perifèrics a l'Arduino

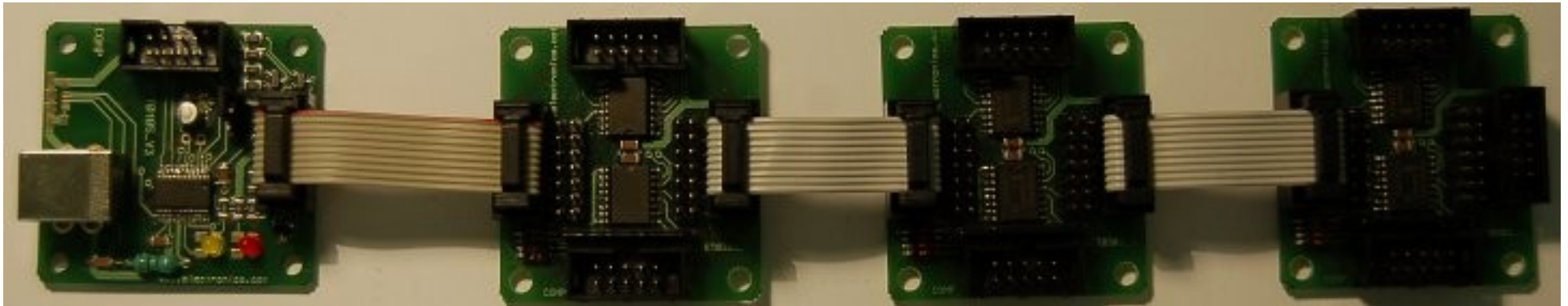
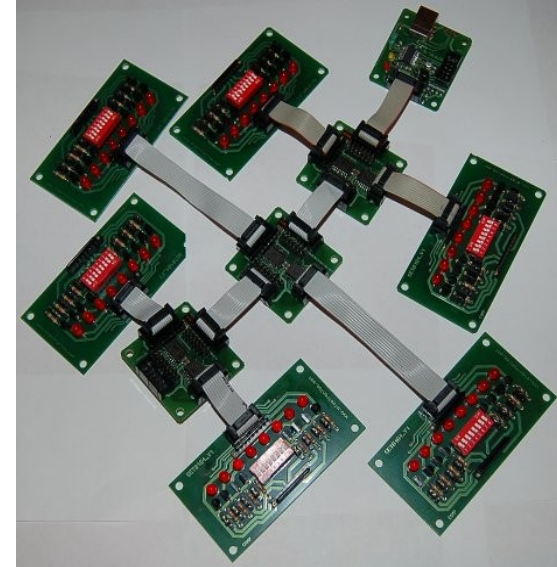
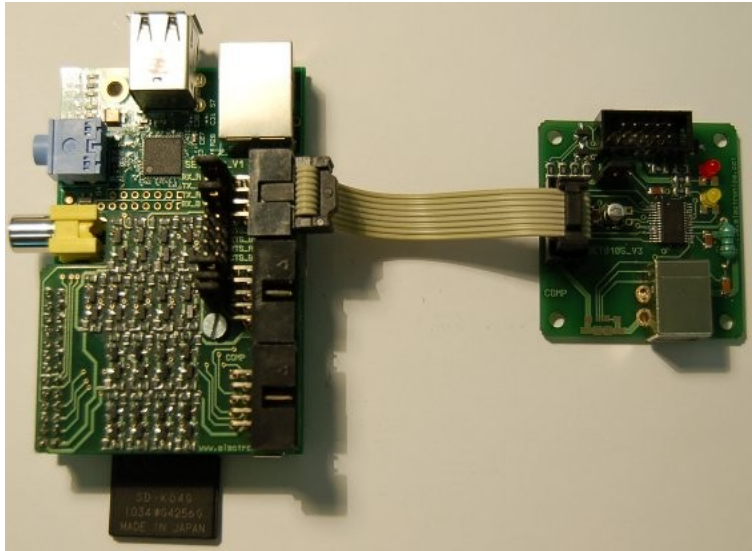
## I2C





# Connexió de perifèrics a l'Arduino

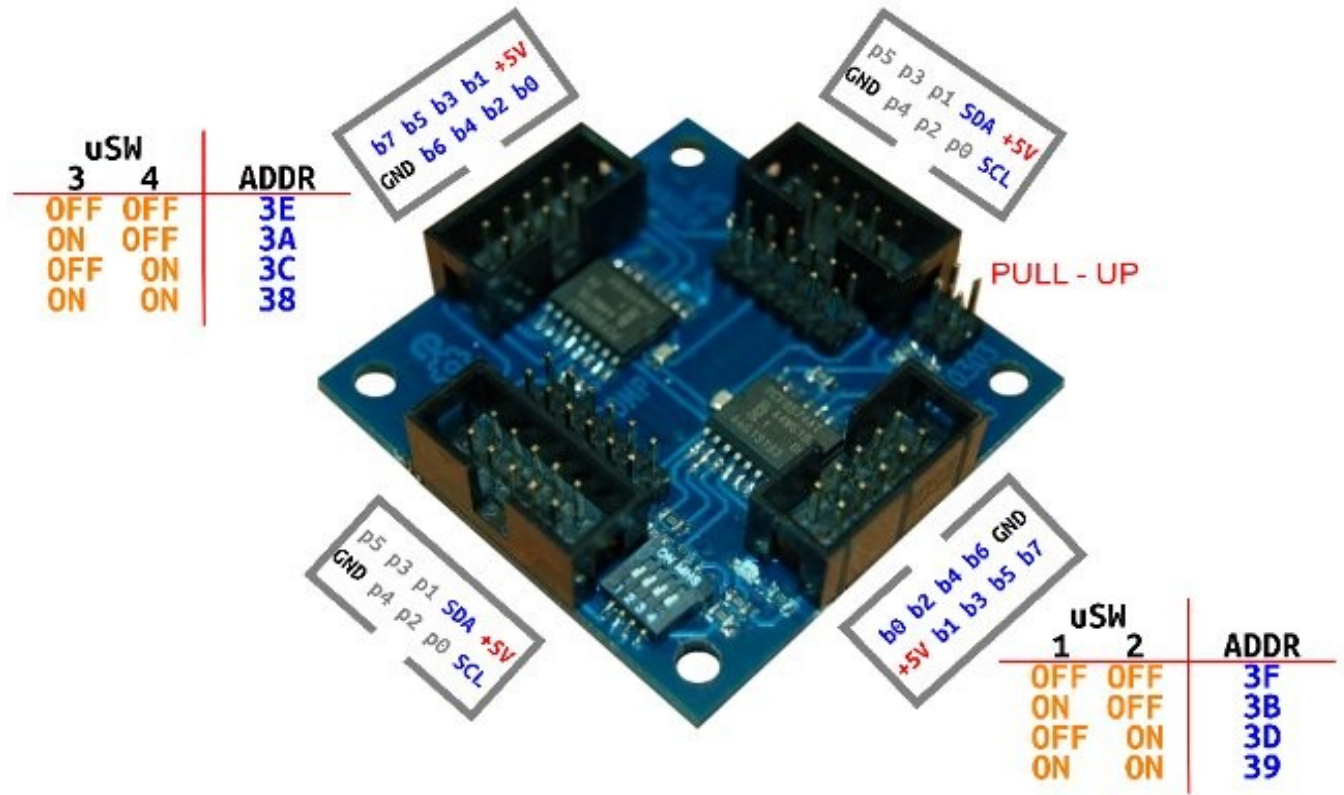
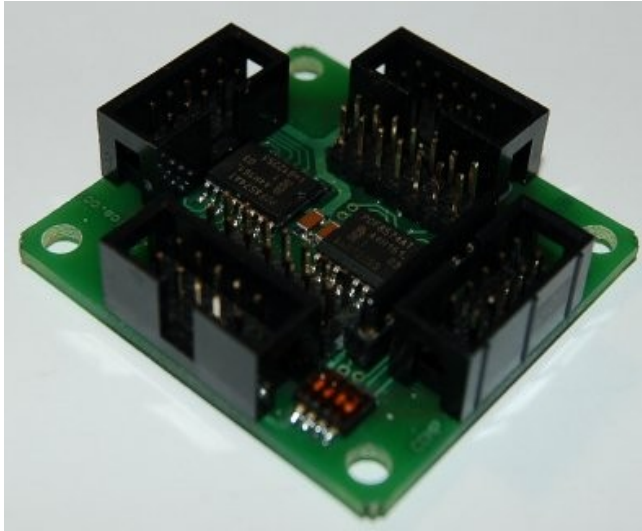
## I2C





# Connexió de perifèrics a l'Arduino

## I2C

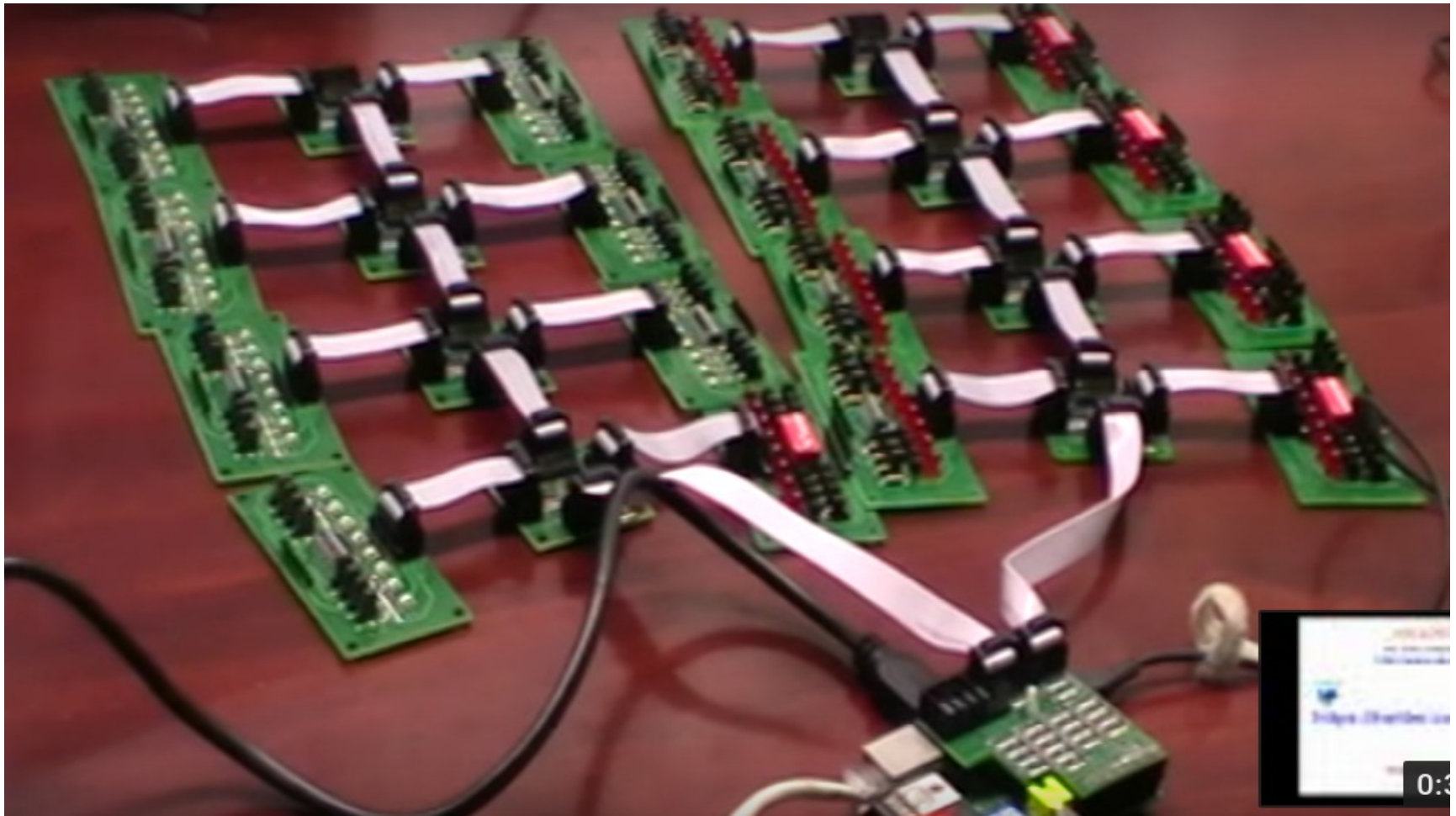






# Connexió de perifèrics a l'Arduino

## I2C

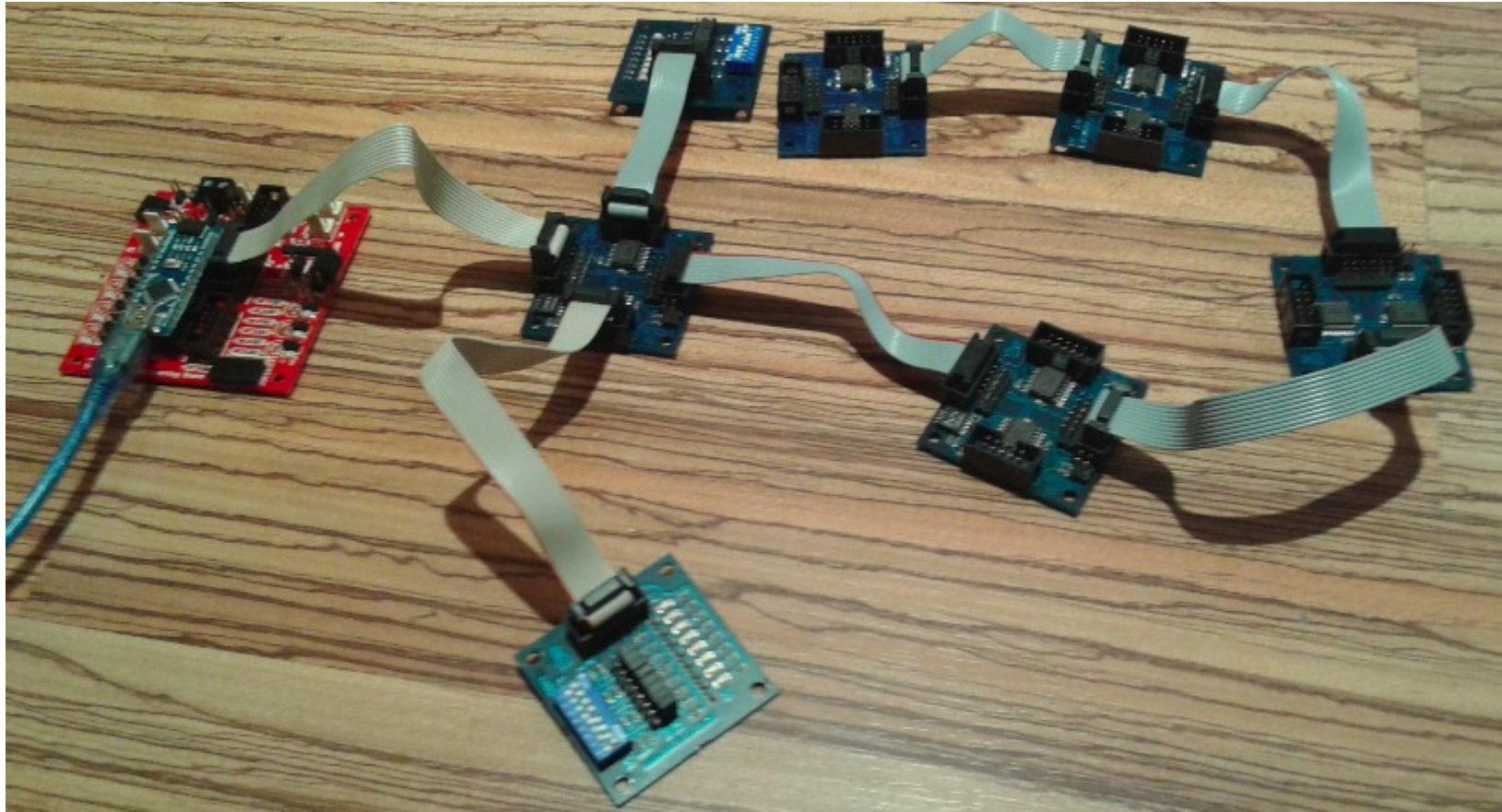


<http://electronics.cat/php/common/index.php?lang=ca&page=508>



# Connexió de perifèrics a l'Arduino

## I2C



Codis I2C



# Connexió de perifèrics a l'Arduino

## 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++;
    }
  }
}
```



# Connexió de perifèrics a l'Arduino

## 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



# Connexió de perifèrics a l'Arduino

## 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);
}
```

Codis I2C



# Connexió de perifèrics a l'Arduino

## BME 280 funcionant amb I2C

P2

+-----+	
GND  SCK	
MISO MOSI	
CS    x	
x     A4	
A5    +5V	
+-----+	

P2

+-----+	
GND  13	
12   11	
10   9	
8    18	
19   +5V	
+-----+	

Mode I2C:

```
bme280.parameter.communication = 0;
```

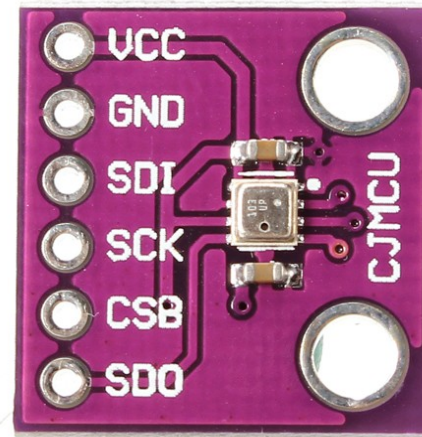
Recordeu de fer servir un scanner I2C per saber l'adreça (sovint és 0x76)

```
bme280.parameter.I2CAddress = 0x77; <-- Canvieu el 0x77 per 0x76 o l'adreça I2C trobada.
```

A l'Arduino Nano:

SDA: A4

SCL: A5



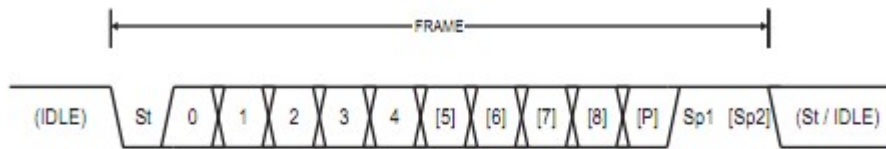
[https://binefa.cat/training/s4a/i2c\\_scanner.zip](https://binefa.cat/training/s4a/i2c_scanner.zip)





# Connexió de perifèrics a l'Arduino

## UART



**St** Start bit, always low.

**(n)** Data bits (0 to 8).

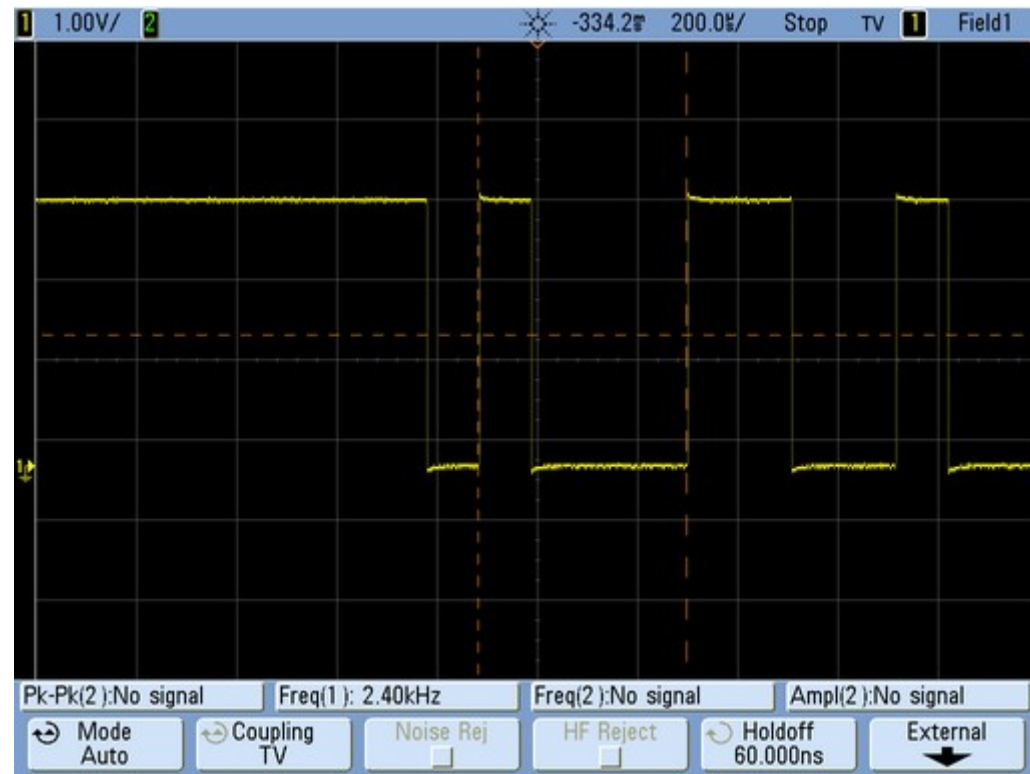
**P** Parity bit. Can be odd or even.

**Sp** Stop bit, always high.

**IDLE** No transfers on the communication line (RxDn or TxDn). An IDLE line must be high.

UART Settings

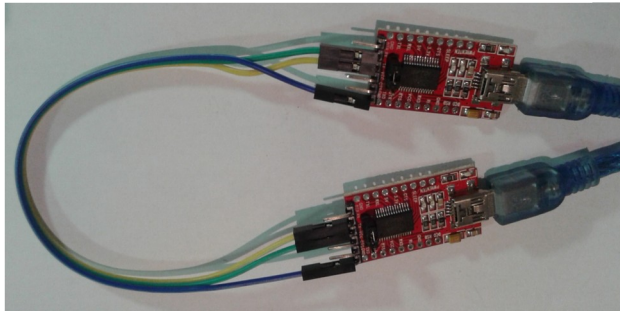
Baudrate	Data Bits	Parity	Stop Bits
<input type="radio"/> 300	<input type="radio"/> 5	<input checked="" type="radio"/> none	<input checked="" type="radio"/> 1
<input type="radio"/> 1200	<input type="radio"/> 6	<input type="radio"/> even	<input type="radio"/> 1.5
<input type="radio"/> 2400	<input checked="" type="radio"/> 7	<input type="radio"/> odd	<input type="radio"/> 2
<input type="radio"/> 4800	<input type="radio"/> 8		
<input checked="" type="radio"/> 9600	Software Flow Control		
<input type="radio"/> 19200	<input checked="" type="radio"/> on <input type="radio"/> off		
<input type="radio"/> 38400	Xon/Xoff karakter		
<input type="radio"/> 57600	Xon Char (HEX) 11 (0x11)		
<input type="radio"/> 115200	Xoff Char (HEX) 13 (0x13)		





# Connexió de perifèrics a l'Arduino

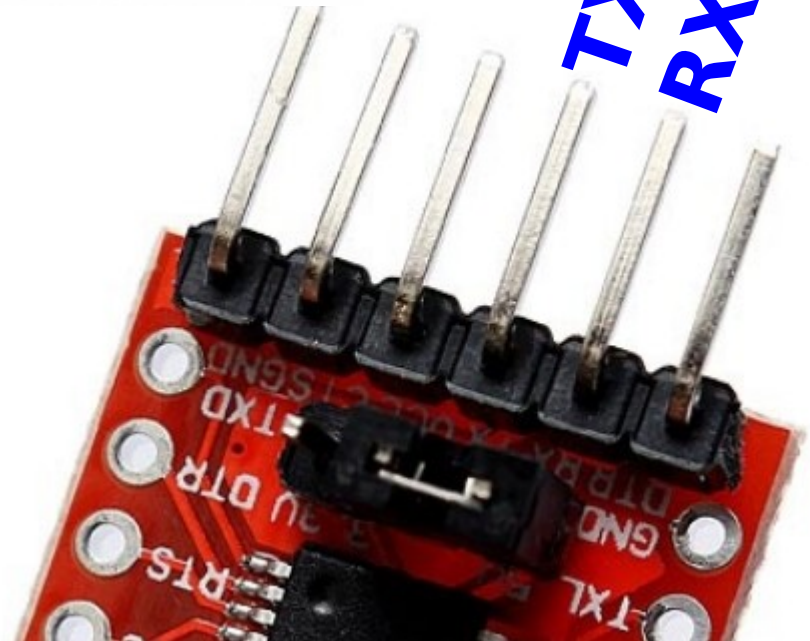
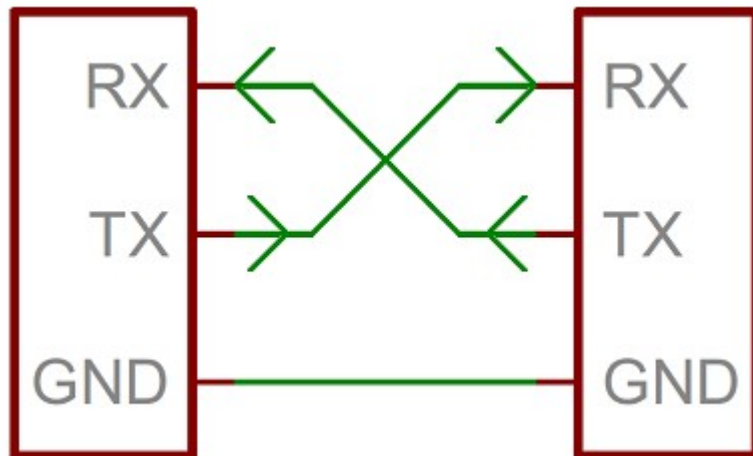
## UART



El pont pot estar  
a 3,3V o a 5V

**GND**

**TX**  
**RX**





# Connexió de perifèrics a l'Arduino

## UART virtual / Biblioteca SoftwareSerial

```
#include <SoftwareSerial.h>

SoftwareSerial mySerial(4, 5); // RX, TX

void setup()
{
    // Open serial communications and wait for port to open:
    Serial.begin(115200);
    while (!Serial) {
        ; // wait for serial port to connect. Needed for Leonardo only
    }

    // set the data rate for the SoftwareSerial port
    mySerial.begin(9600);
}

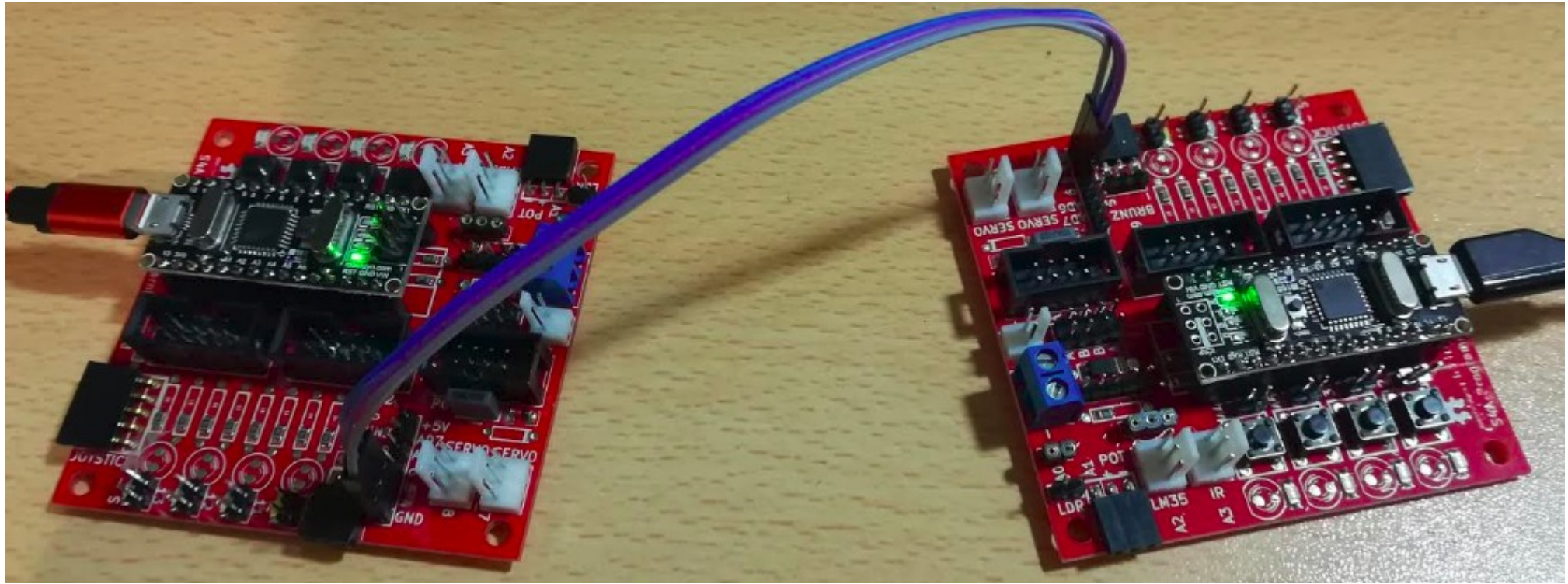
void loop() // run over and over
{
    if (mySerial.available())
        Serial.write(mySerial.read());
    if (Serial.available())
        mySerial.write(Serial.read());
}
```

<https://binefa.cat/training/s4a/virtualSerial/virtualSerial.ino>



# Connexió de perifèrics a l'Arduino

## UART virtual / Biblioteca SoftwareSerial



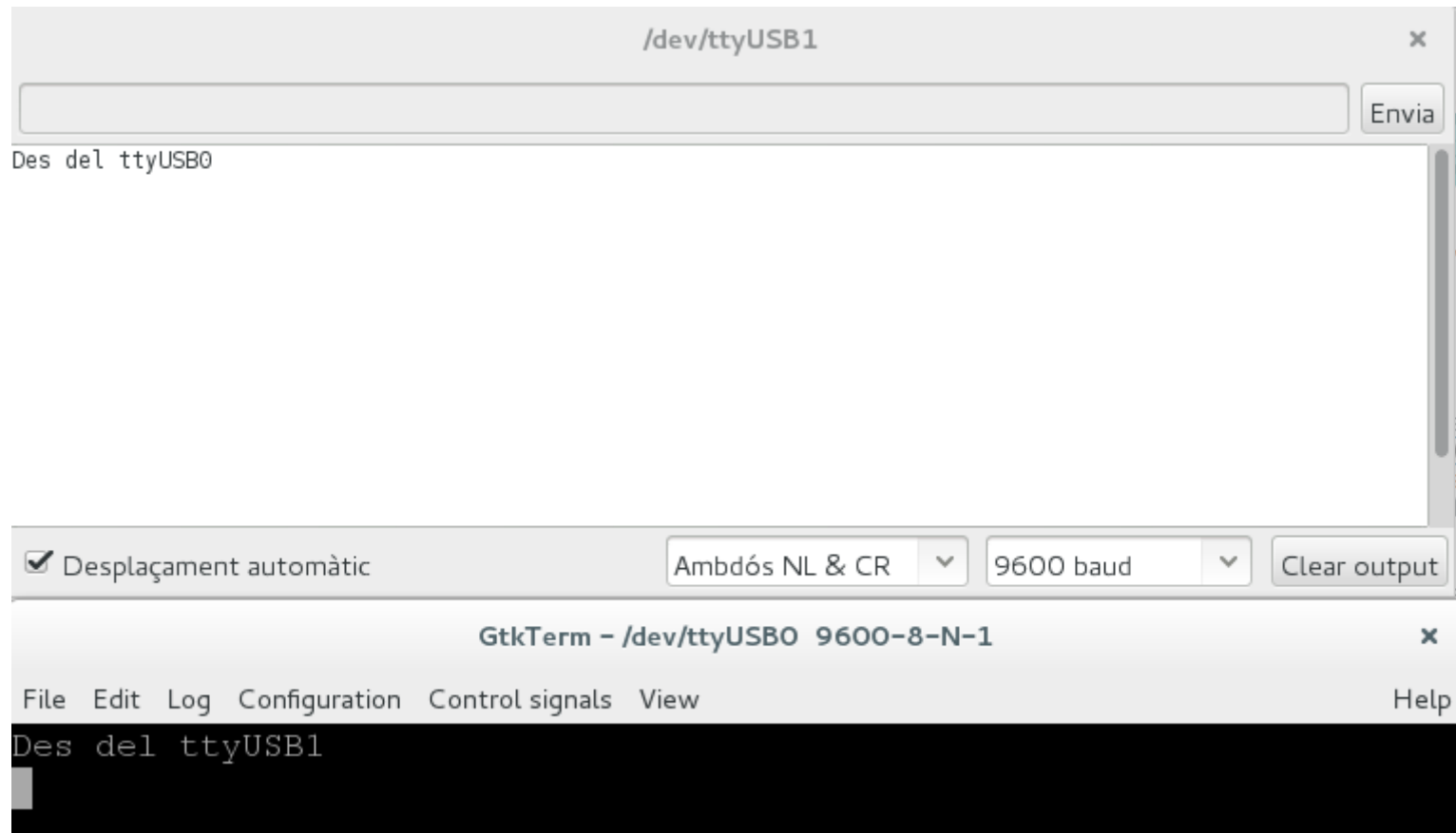
<https://binefa.cat/training/s4a/virtualSerial/virtualSerial.ino>





# Connexió de perifèrics a l'Arduino

## UART virtual / Biblioteca SoftwareSerial



<https://binefa.cat/training/s4a/virtualSerial/virtualSerial.ino>



# e⚙️ Comunicació entre l'Arduino i l'ordinador

## Càrrega del protocol Firmata a l'Arduino

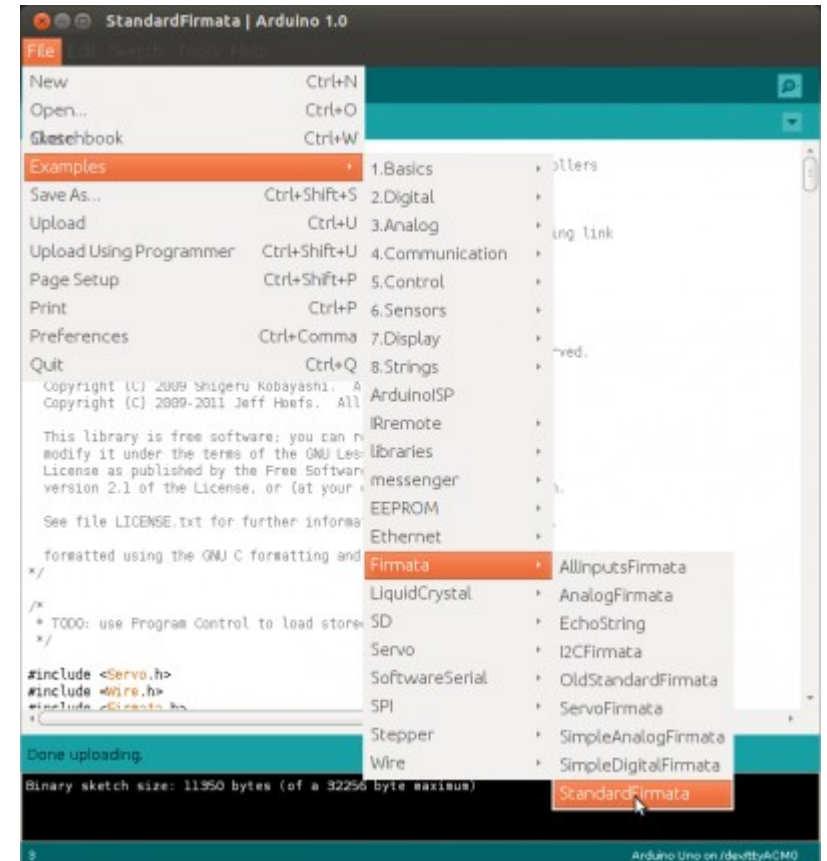
Connecteu la Raspberry Pi a la placa nano-eCat (la que conté l'Arduino Nano) mitjançant el cable USB-A a miniUSB.

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 l'Arduino i l'ordinador

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

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

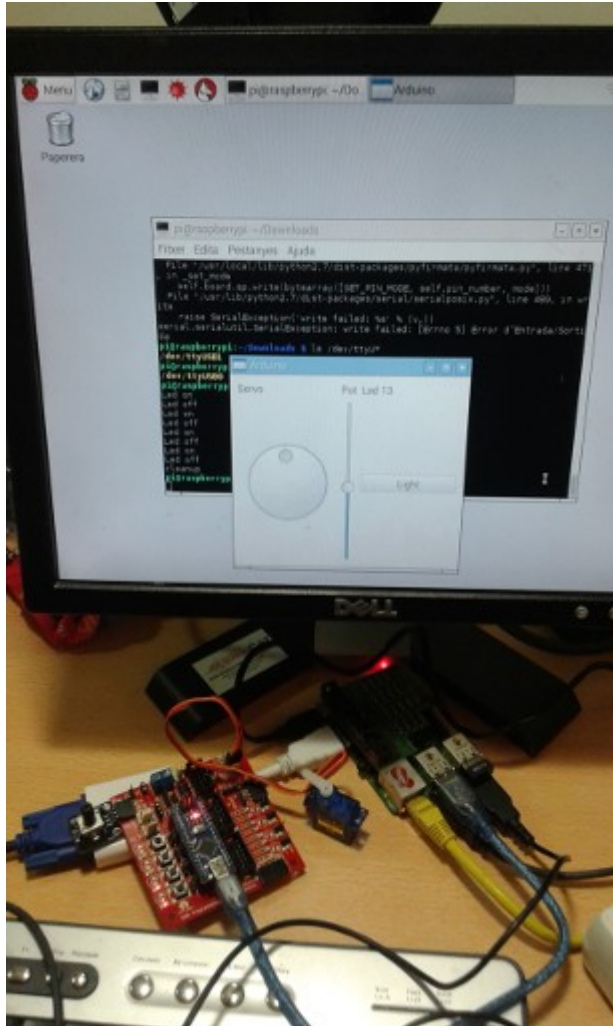
```
pi@raspberrypi ~ $ cd Documents/codis/firmata/pyQt/
pi@raspberrypi ~/Documents/codis/firmata/pyQt $ ls -ls
total 24
8 -rwxr-xr-x 1 pi pi 4122 des  7 21:10 exFirmata03P1.py
4 -rwxr-xr-x 1 pi pi 3970 des  7 20:57 exFirmata03P2.py
4 -rw-r----- 1 pi pi 3255 abr  9 13:31 led_uSw.ui
4 -rw-r--r-- 1 pi pi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



# Comunicació entre l'Arduino i l'ordinador

## 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>



# Comunicació entre l'Arduino i l'ordinador

## Protocol Firmata estàndard



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

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



# Comunicació entre l'Arduino i l'ordinador

## 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>

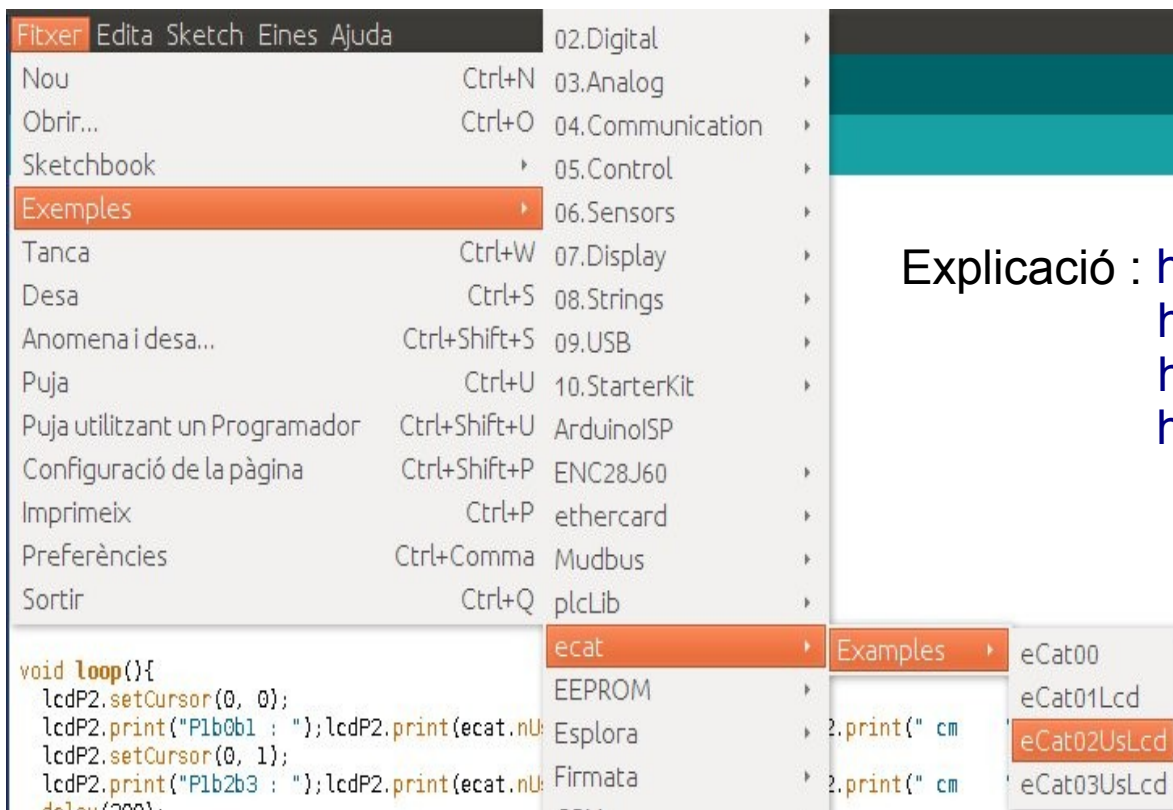




# Tecnologies creatives a l'aula

## Biblioteca **ecat.h** per a l'IDE d'Arduino

<http://www.electronics.cat/doc/arduinoLib/ecat.zip>

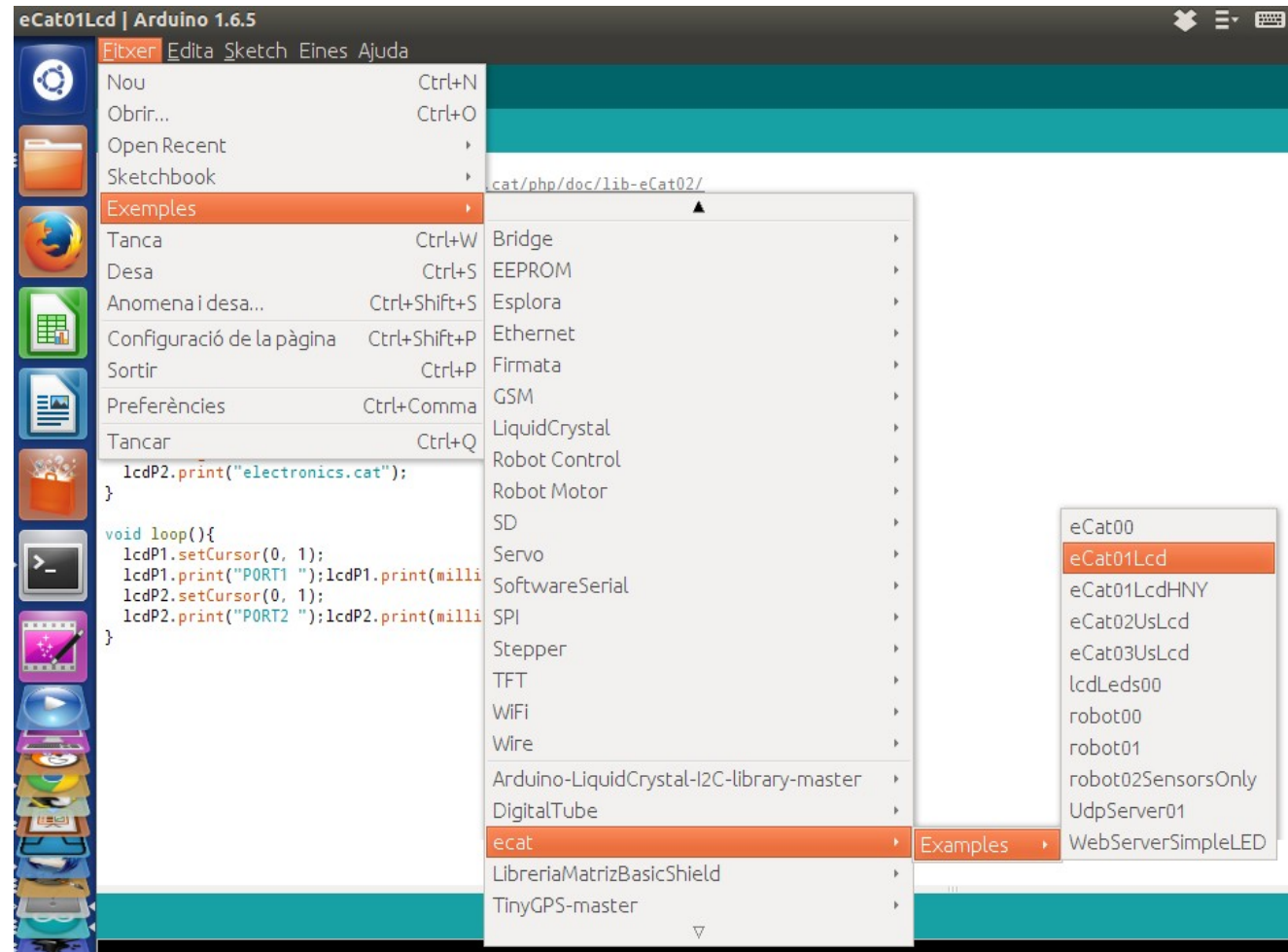
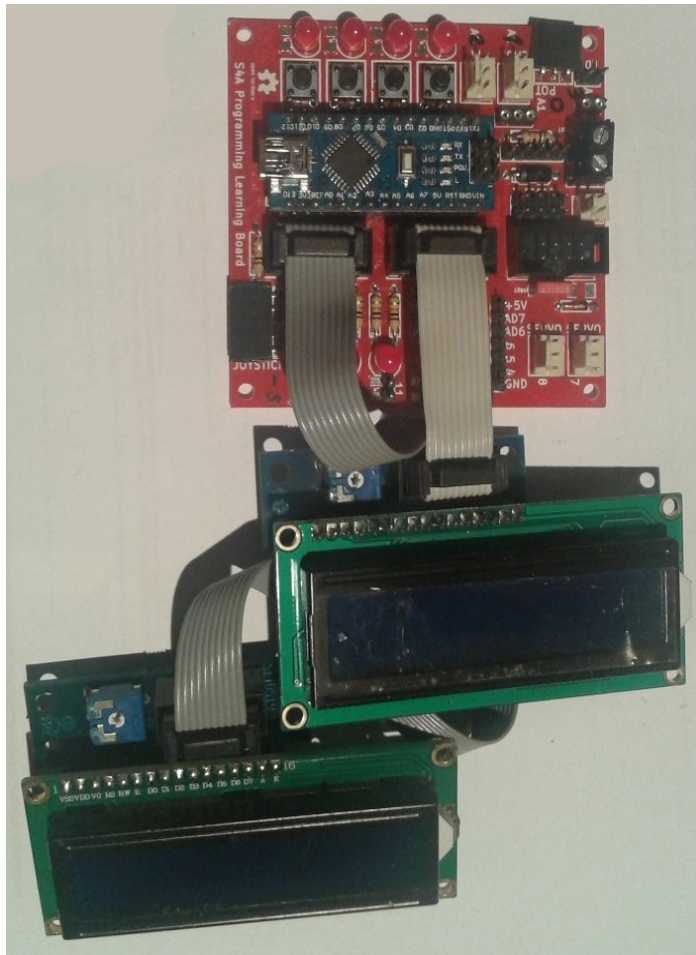


Explicació : <http://www.binefa.cat/php/doc/lib-eCat01/>  
<http://www.binefa.cat/php/doc/lib-eCat02/>  
<http://www.binefa.cat/php/doc/lib-eCat03/>  
<http://binefa.cat/blog/?p=140>



# S4A Programming Learning Board

## Ús de la biblioteca ecat.h





# Tecnologies creatives a l'aula

## Biblioteca **ecat.h** per a l'IDE d'Arduino

<http://www.electronics.cat/doc/arduinoLib/ecat.zip>

Explicació : <http://www.binefa.cat/php/doc/lib-eCat01/>  
<http://www.binefa.cat/php/doc/lib-eCat02/>  
<http://www.binefa.cat/php/doc/lib-eCat03/>  
<http://binefa.cat/blog/?p=140>

```

1  #include <ecat.h>
2
3  #define ROBOT_ATURAT      0
4  #define ROBOT_ENDAVANT    1
5  #define ROBOT_ENDARRERA   2
6  #define ROBOT_DRETA       3
7  #define ROBOT_ESQUERRA    4
8
9  String szMissatge;
10 Ecat ecat;
11 boolean bConnectat;
12 int nG,nA,nB,nEstatActual;
13
14 void setup(){
15     ecat.setupNibbleMode(NIBBLE_H_P1,OUTPUT);
16     ecat.vUltrasonicSensorP1b0b1_init();
17     pinMode(ecat.nPinP1B2,INPUT);
18     pinMode(ecat.nPinP1B3,INPUT);
19     pinMode(ecat.nPinP2B7,OUTPUT);
20     pinMode(ecat.nPinP2B6,INPUT);
21     pinMode(ecat.nPinP2B5,INPUT);
22     pinMode(ecat.nPinP2B4,INPUT);
23     ecat.setupNibbleMode(NIBBLE_L_P2,INPUT);
24     Serial.begin(9600);
25     nG = 150;
26     nA = nB = 30;
27     bConnectat = false;
28     nEstatActual = ROBOT_ATURAT;
29 }
30

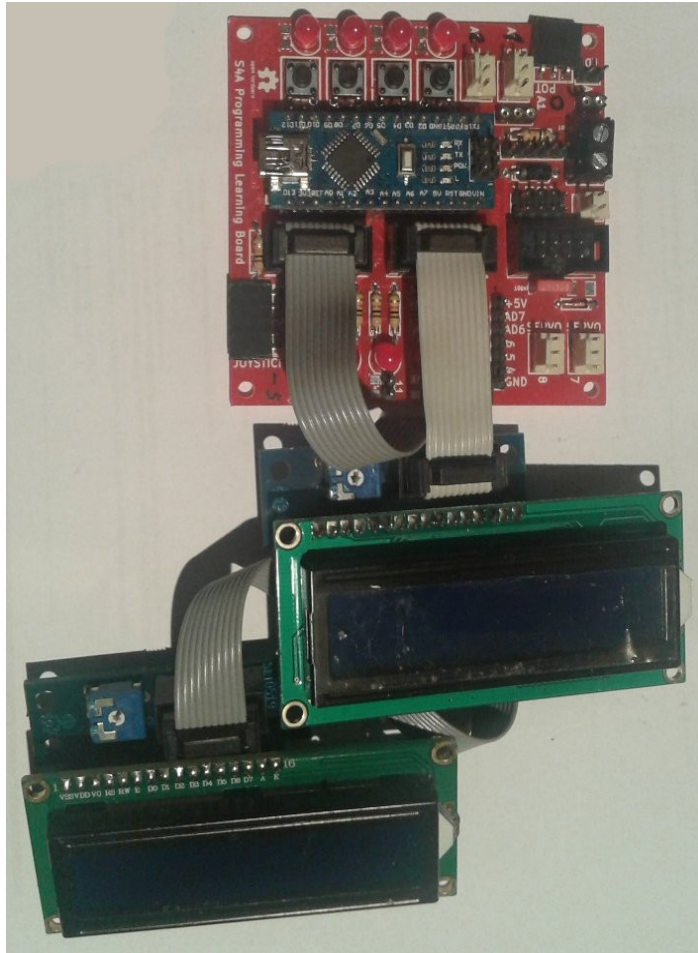
```

<http://electronics.cat/downloads/code/robot08.ino>



# S4A Programming Learning Board

## Ús de la biblioteca ecat.h



```

/*
Connection schematic at http://www.binefa.cat/php/doc/lib-eCat02/
*/

#include <ecat.h>
#include <LiquidCrystal.h>

Ecat ecat;
lcdP1_init();
lcdP2_init();

void setup(){
  lcdP1.begin(16, 2);
  lcdP1.print("ecat.h - v");lcdP1.print(ecat.fVersion());
  lcdP2.begin(16, 2);
  lcdP2.print("electronics.cat");
}

void loop(){
  lcdP1.setCursor(0, 1);
  lcdP1.print("PORT1 ");lcdP1.print(millis()/1000);lcdP1.print(" seconds");
  lcdP2.setCursor(0, 1);
  lcdP2.print("PORT2 ");lcdP2.print(millis()/1000);lcdP2.print(" seconds");
}

```





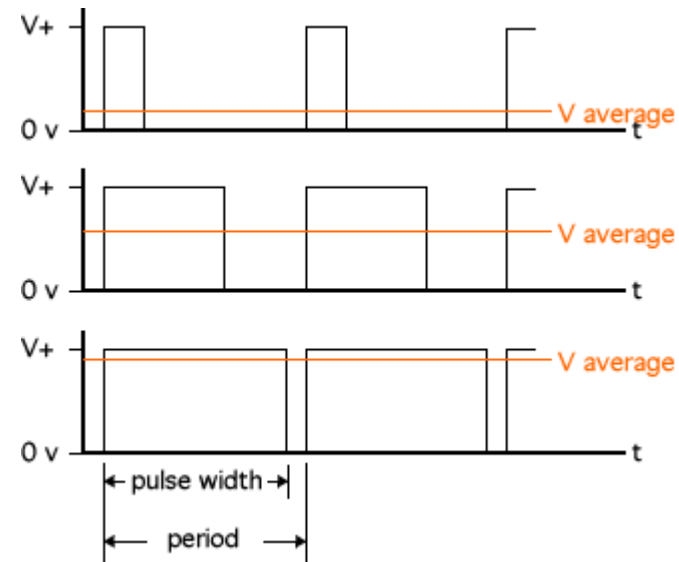
# Tecnologies creatives a l'aula

## Modulació per amplada de pols PWM

```

31 int nPwmA(){
32     int nAux = nG + nA;
33     if(nAux > 255)
34         nAux = 255;
35     return nAux;
36 }
37
38 int nPwmB(){
39     int nAux = nG + nB;
40     if(nAux > 255)
41         nAux = 255;
42     return nAux;
43 }
44
45 void vRobotAturat(){
46     nEstatActual = ROBOT_ATURAT;
47     ecat.vWriteHighNibbleP1(0x00);
48 }
49
50 void vRobotEndavant(){
51     //ecat.vWriteHighNibbleP1(B00000110);
52     nEstatActual = ROBOT_ENDAVANT;
53     digitalWrite(ecat.nPinP1B4, LOW);
54     analogWrite(ecat.nPinP1B5, nPwmA());
55     analogWrite(ecat.nPinP1B6, nPwmB());
56     digitalWrite(ecat.nPinP1B7, LOW);
57 }

```





# Tecnologies creatives a l'aula

## Preparant la comunicació Bluetooth

```
#include <ecat.h>

Ecat ecat;
boolean bP2B0;

void setup(){
  pinMode(ecat.nPinP2B7, OUTPUT);
  pinMode(ecat.nPinP2B0, INPUT);
  Serial.begin(9600);
  bP2B0 = digitalRead(ecat.nPinP2B0);
}

void loop(){
  String szMsg;

  while(Serial.available()){
    delay(3);
    char c = Serial.read();
    szMsg += c;
  }
  if(szMsg == "n"){
    digitalWrite(ecat.nPinP2B7, HIGH);
  }
  if(szMsg == "f"){
    digitalWrite(ecat.nPinP2B7, LOW);
  }
  if(bP2B0 != digitalRead(ecat.nPinP2B0)){
    bP2B0 = digitalRead(ecat.nPinP2B0);
    if(bP2B0){
      Serial.println("P2B0 HIGH");
    }else{
      Serial.println("P2B0 LOW");
    }
  }
  szMsg = "";
}
```



<http://electronics.cat/doc/hc06/bluetooth04.ino>

# Tecnologies creatives a l'aula

## Com trobar pel terminal el número ID del Bluetooth

```
$ hcitool scan
Scanning ...
    98:D3:31:30:2C:0D    HC-06
```





# Tecnologies creatives a l'aula

## Preparant la comunicació Bluetooth

```
#include <ecat.h>

Ecat ecat;
boolean bP2B0;

void setup(){
  pinMode(ecat.nPinP2B7, OUTPUT);
  pinMode(ecat.nPinP2B0, INPUT);
  Serial.begin(9600);
  bP2B0 = digitalRead(ecat.nPinP2B0);
}
```



# Tecnologies creatives a l'aula

## Preparant la comunicació Bluetooth

```
void loop(){
  String szMsg;

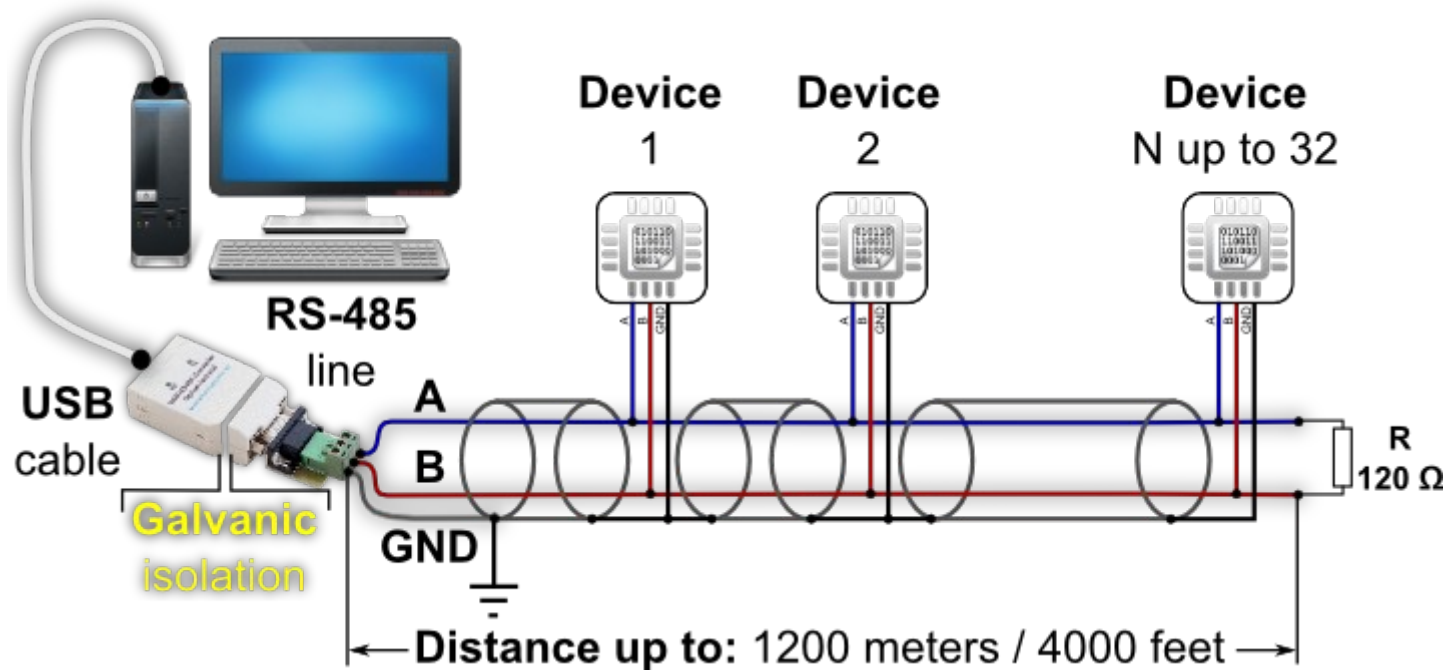
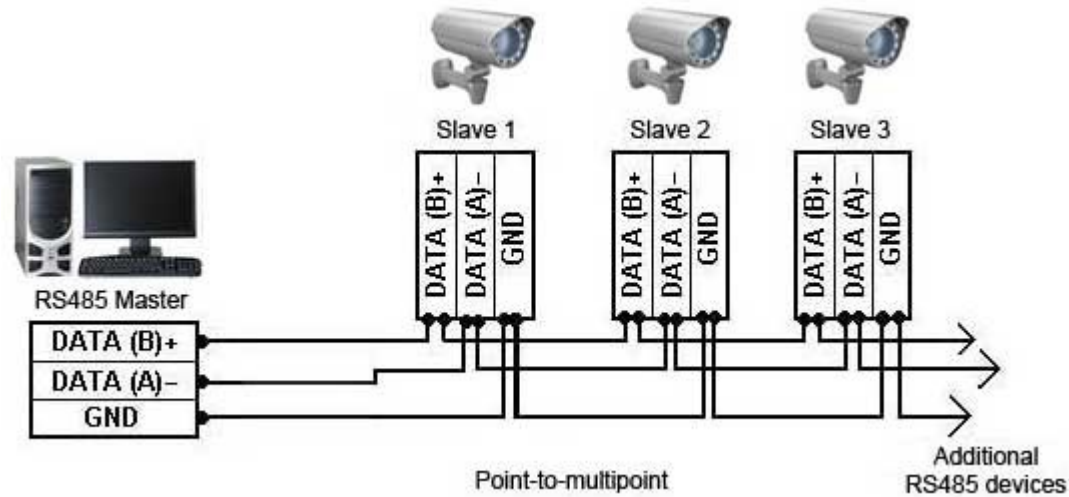
  while(Serial.available()){
    delay(3);
    char c = Serial.read();
    szMsg += c;
  }
  if(szMsg == "n"){
    digitalWrite(ecat.nPinP2B7,HIGH);
  }
  if(szMsg == "f"){
    digitalWrite(ecat.nPinP2B7,LOW);
  }
  if(bP2B0 != digitalRead(ecat.nPinP2B0)){
    bP2B0 = digitalRead(ecat.nPinP2B0);
    if(bP2B0){
      Serial.println("P2B0 HIGH");
    }else{
      Serial.println("P2B0 LOW");
    }
  }
  szMsg = "";
}
```

<http://electronics.cat/doc/hc06/bluetooth04.ino>





# RS485

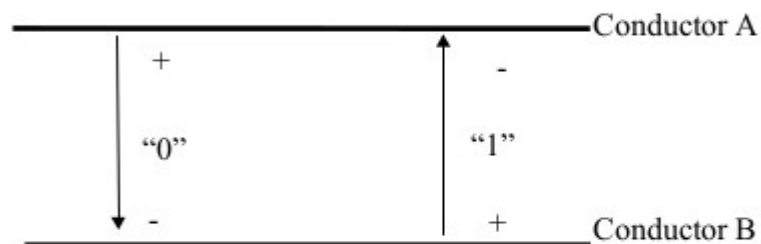


## Comunicación Serie RS-422

### 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



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

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



# 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



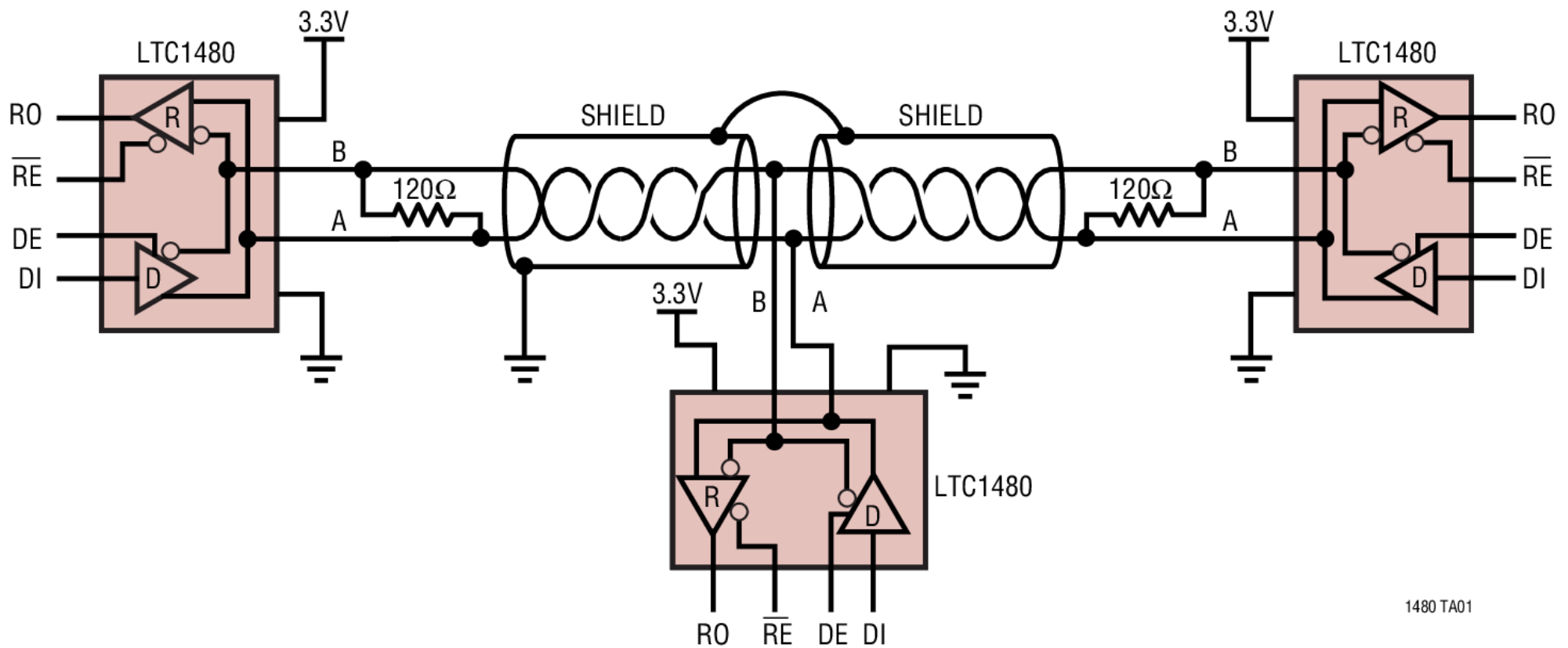
# RS485

## *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	≥4K	≥12K

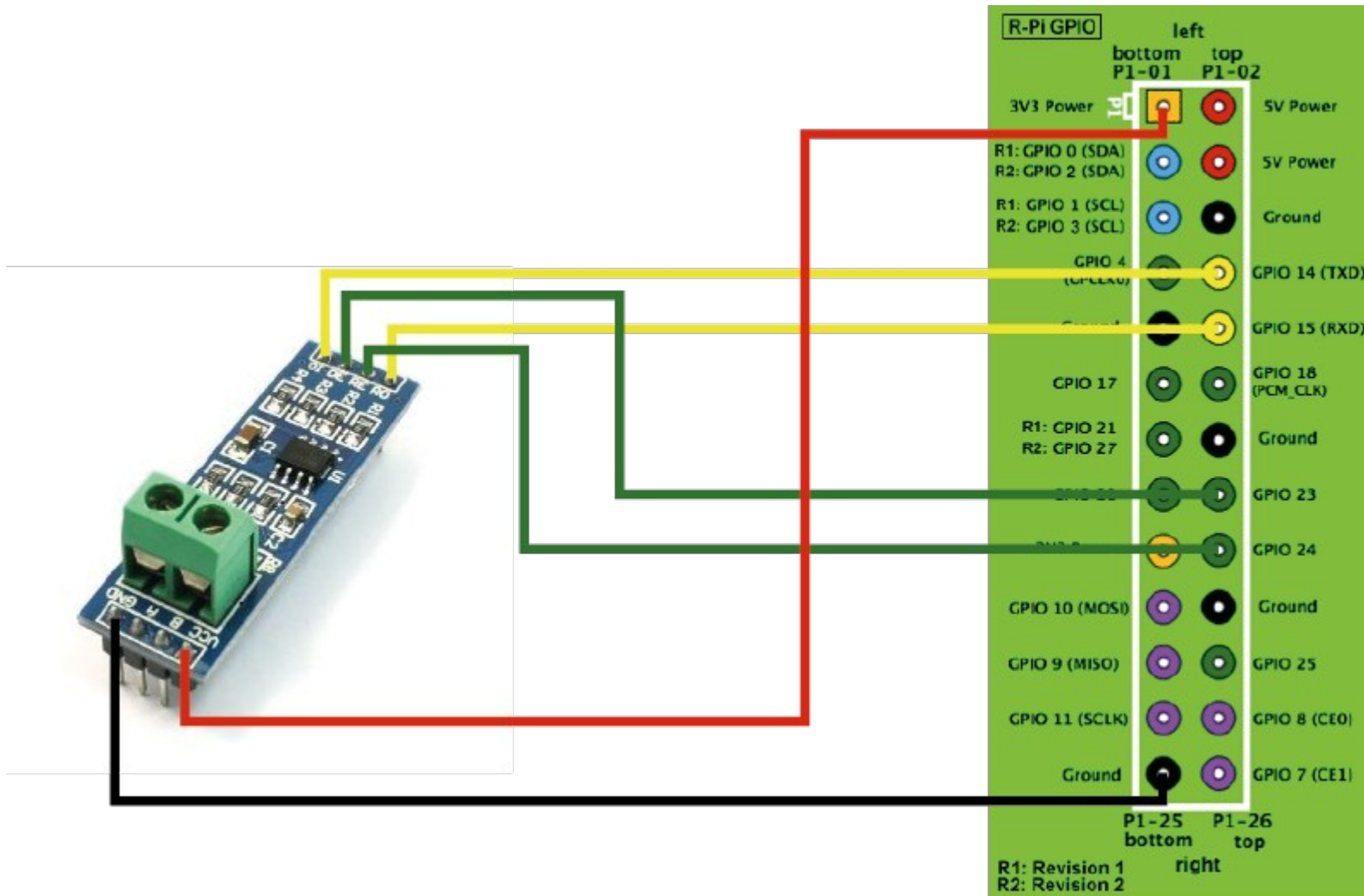
# RS485

## 3.3V RS485 Network



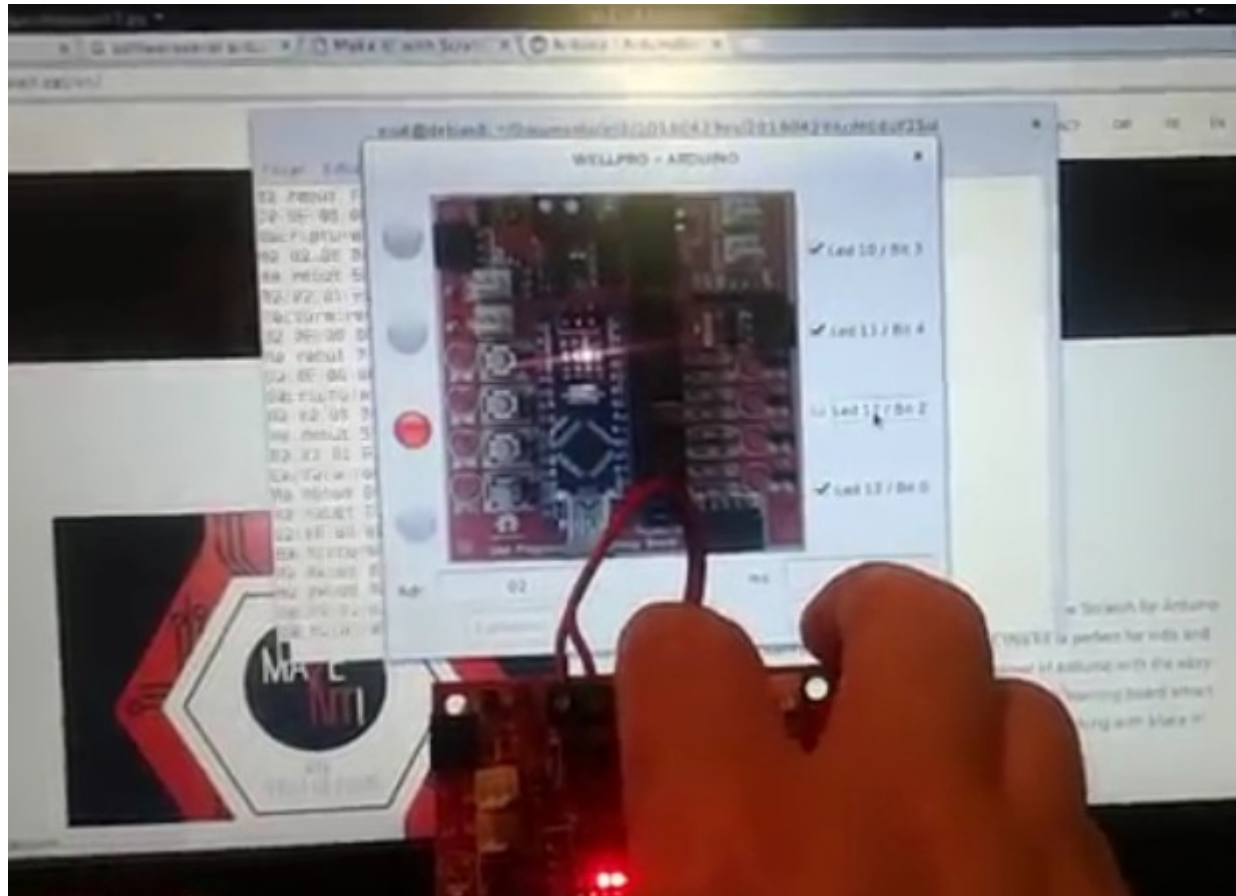


**electronics.cat**



# RS485

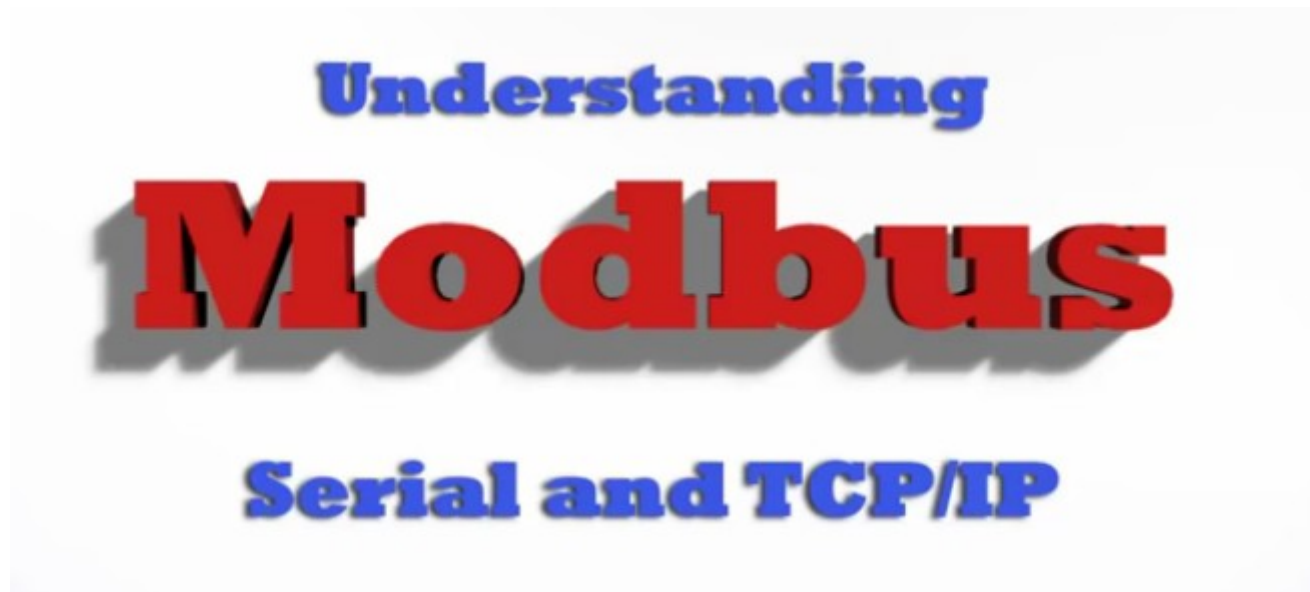
## ModBus RTU



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

# RS485

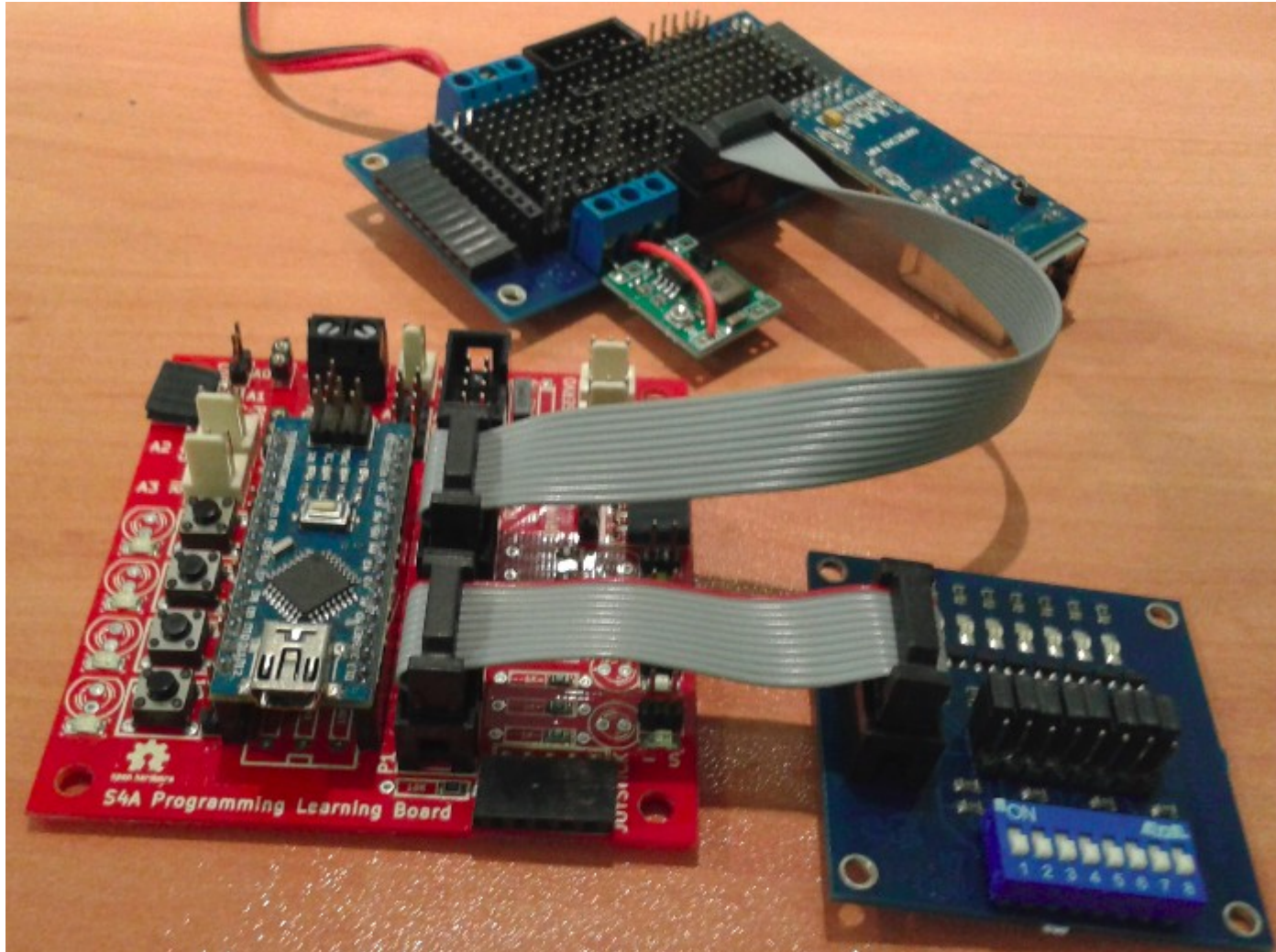
## ModBus RTU



Vídeo explicatiu del ModBusRTU i el ModBusTCP

# Ethernet

## Ethernet - Mòdul ENC28J60 Un altre ús del bus SPI

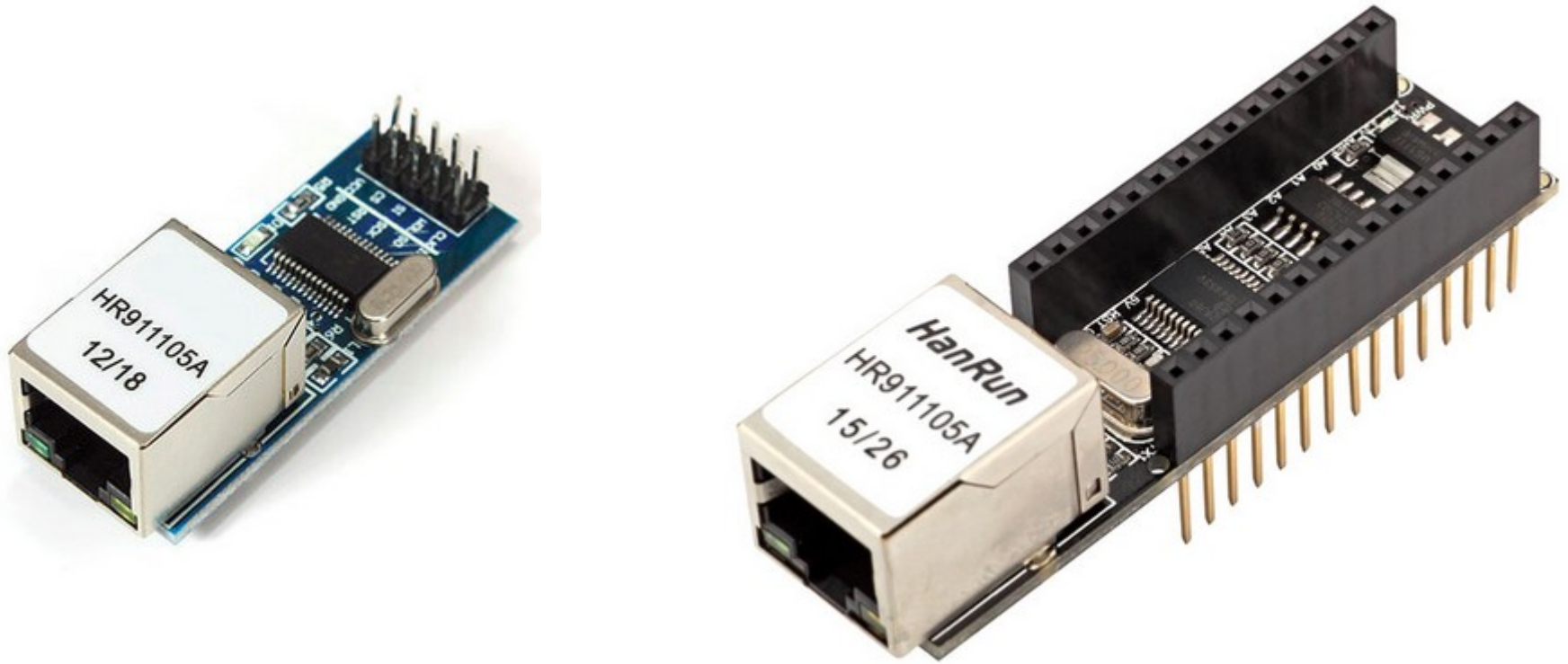


Codi per a Arduino UdpServer01b



# Ethernet

## Ethernet - Mòdul ENC28J60



Biblioteca UIPEthernet per Arduino i ENC28J60





# Ethernet

## Ethernet - Mòdul ENC28J60

Connexió a la matriu inversa (pàgina 100 del pdf)

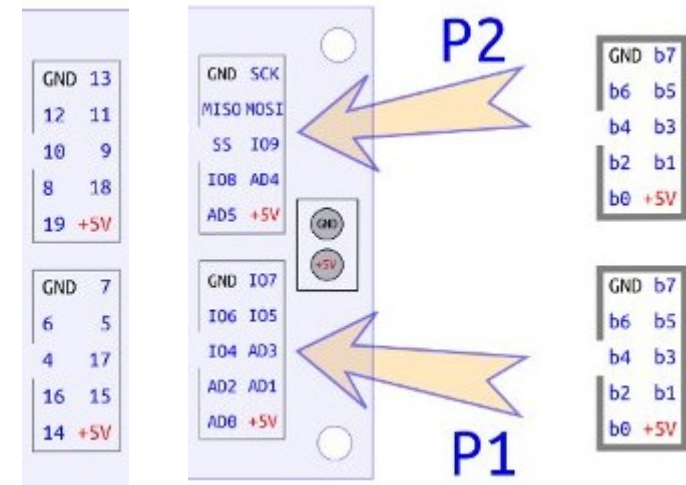
C8	C6	c4	C2	C0
C9	C7	C5	C3	C1

Connexió de l'ENC28J60

CLK	WOL	SI	CS	VCC
INT	S0	SCK	RST	GND

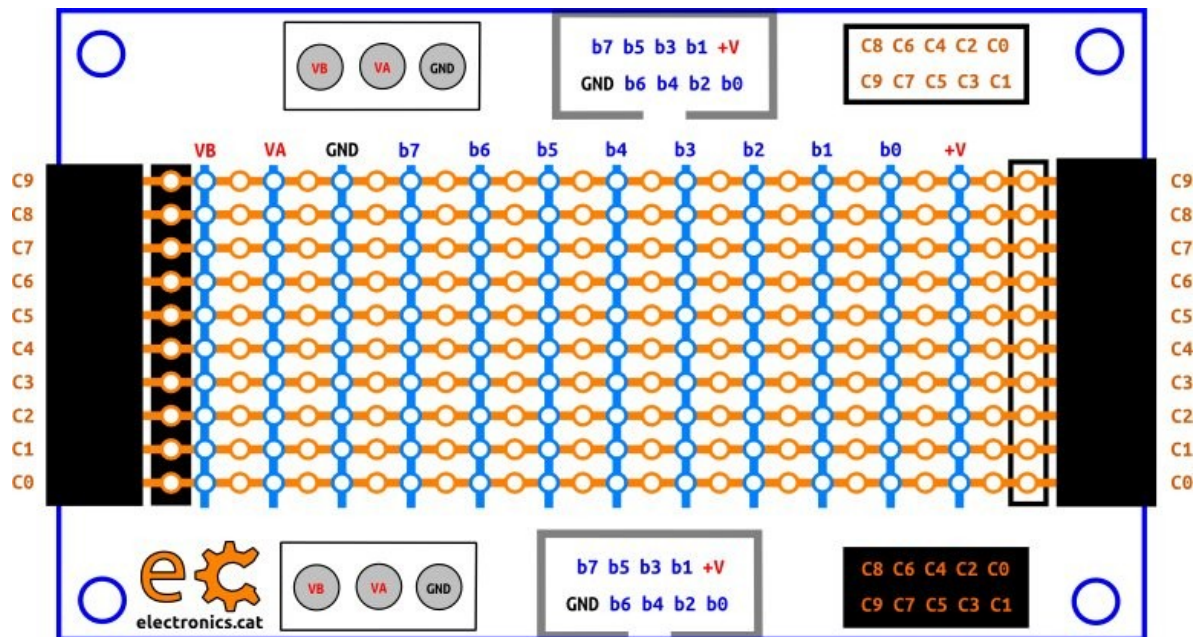
Connexió bus ecat

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	





# Ethernet

## Ethernet - Mòdul ENC28J60

### 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("");
  }
}
```

Codi per a Arduino UdpServer01b



# Ethernet

## Ethernet – Mòdul ENC28J60

### 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



# Ethernet

## Enviament de datagrames UDP en Python

```
$ ./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

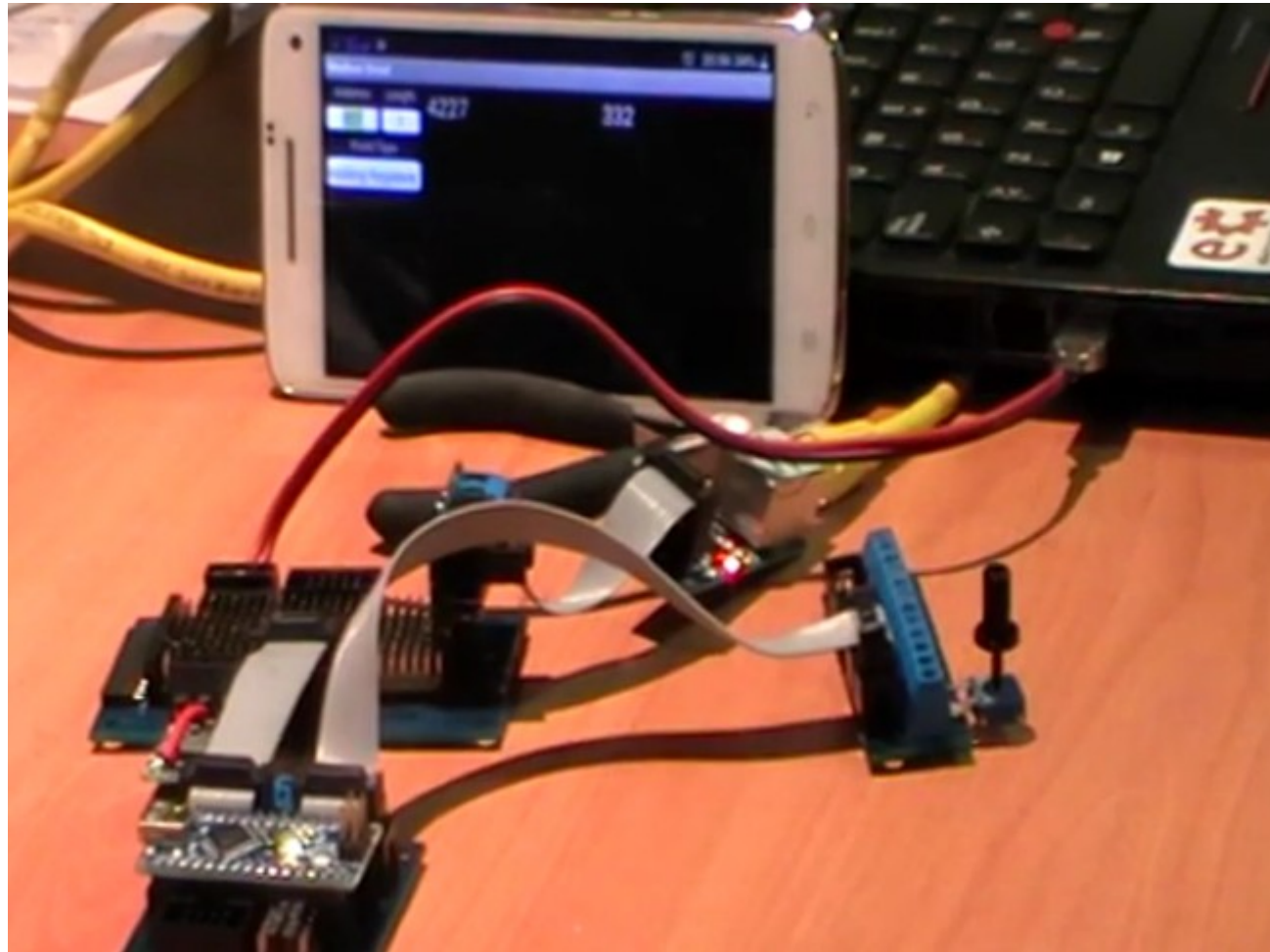
import socket
import sys

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

# Ethernet ModBus - TCP



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

Vídeo: <https://www.youtube.com/embed/j-1f-dV7ohQ>





# LoRaMAC

## Encriptació AES

**/dev/ttyUSB0 @9600**

String: SX1276 at Arduino

Hexadecimal: 5358313237362061742041726475696E6F    CRC: 6791

☒ Tx    ☐ Without CRC    ☒ Read bytes as characters  
☐ Rx    ☐ Add CRC    ☒ Encrypt AES at uC (Tx)  
☐ Freq    ☐ CRC at uC    ☒ Decrypt AES at uC (Rx)

Send

```

202020205278206D6F64650D0A ( Rx mode )
E0
202020205278206D6F64650D0A (: 32) (Rx mode )
363208115358313237362061742045535038323636 (6: 17) (SX1276 at ESP8266)
08115358313237362061742041726475696E6F
08115358313237362061742041726475696E6F
08115358313237362061742041726475696E6F
08115358313237362061742041726475696E6F
08115358313237362061742041726475696E6F
08115358313237362061742041726475696E6F
C8
202020205278206D6F64650D0A (: 32) (Rx mode )
373208115358313237362061742045535038323636 (7: 17) (SX1276 at ESP8266)
383208115358313237362061742045535038323636 (8: 17) (SX1276 at ESP8266)
08115358313237362061742041726475696E6F
08115358313237362061742041726475696E6F
08115358313237362061742041726475696E6F
    
```

ecat@debian8:~/Documents/lora/python/lora09\$ ./lora09.py

**/dev/ttyUSB1 @9600**

String: SX1276 at ESP8266

Hexadecimal: 5358313237362061742045535038323636    CRC: 0861

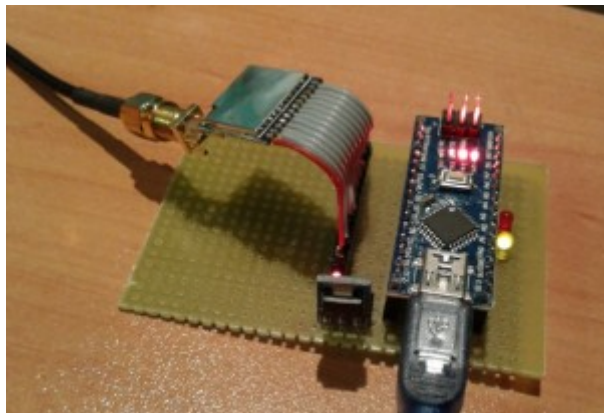
☐ Tx    ☒ Without CRC    ☒ Read bytes as characters  
☒ Rx    ☐ Add CRC    ☒ Encrypt AES at uC (Tx)  
☐ Freq    ☐ CRC at uC    ☒ Decrypt AES at uC (Rx)

Send

```

E0
202020205278206D6F64650D0A ( Rx mode )
08115358313237362061742045535038323636
08115358313237362061742045535038323636
C8
202020205278206D6F64650D0A (: 32) (Rx mode )
3132081153 (1: 17) (S)
58313237362061742041726475696E6F (X: 55) (6 at Arduino)
333208115358313237 (3: 17) (SX127)
362061742041726475696E6F (6: 116) ( Arduino)
353208115358313237362061742041726475696E6F (5: 17) (SX1276 at Arduino)
08115358313237362061742045535038323636
08115358313237362061742045535038323636
C8
202020205278206D6F64650D0A (: 32) (Rx mode )
383208115358313237362061742041726475696E6F (8: 17) (SX1276 at Arduino)
    
```

an8:~/Documents/lora/python/lora09\$ ./lora09.py /dev/ttyUSB1



<http://binefa.cat/blog/?p=327>

# LoRaWAN

## Arduino MKR WAN 1300



Ús i configuració: <http://tinkerman.cat/arduino-mkr-wan-1300/>



# Altres plaques programables des de l'IDE d'Arduino

## Família d'Arduinos i Teensy



Arduino Uno



Arduino Leonardo



Arduino Due



Arduino Yún



Arduino Tre



Arduino Micro



Arduino Robot



Arduino Esplora



Arduino Mega ADK



Arduino Ethernet



Arduino Mega 2560



Arduino Mini



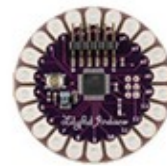
LilyPad Arduino USB



LilyPad Arduino Simple



LilyPad Arduino SimpleSnap



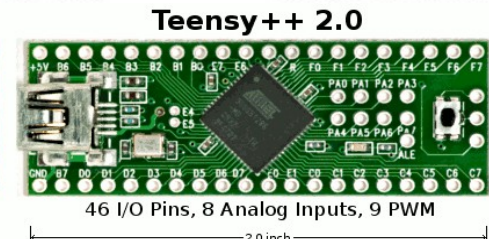
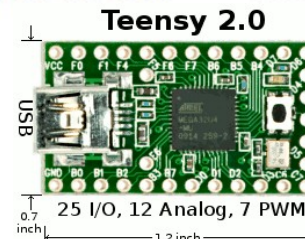
LilyPad Arduino



Arduino Nano



Arduino Pro Mini

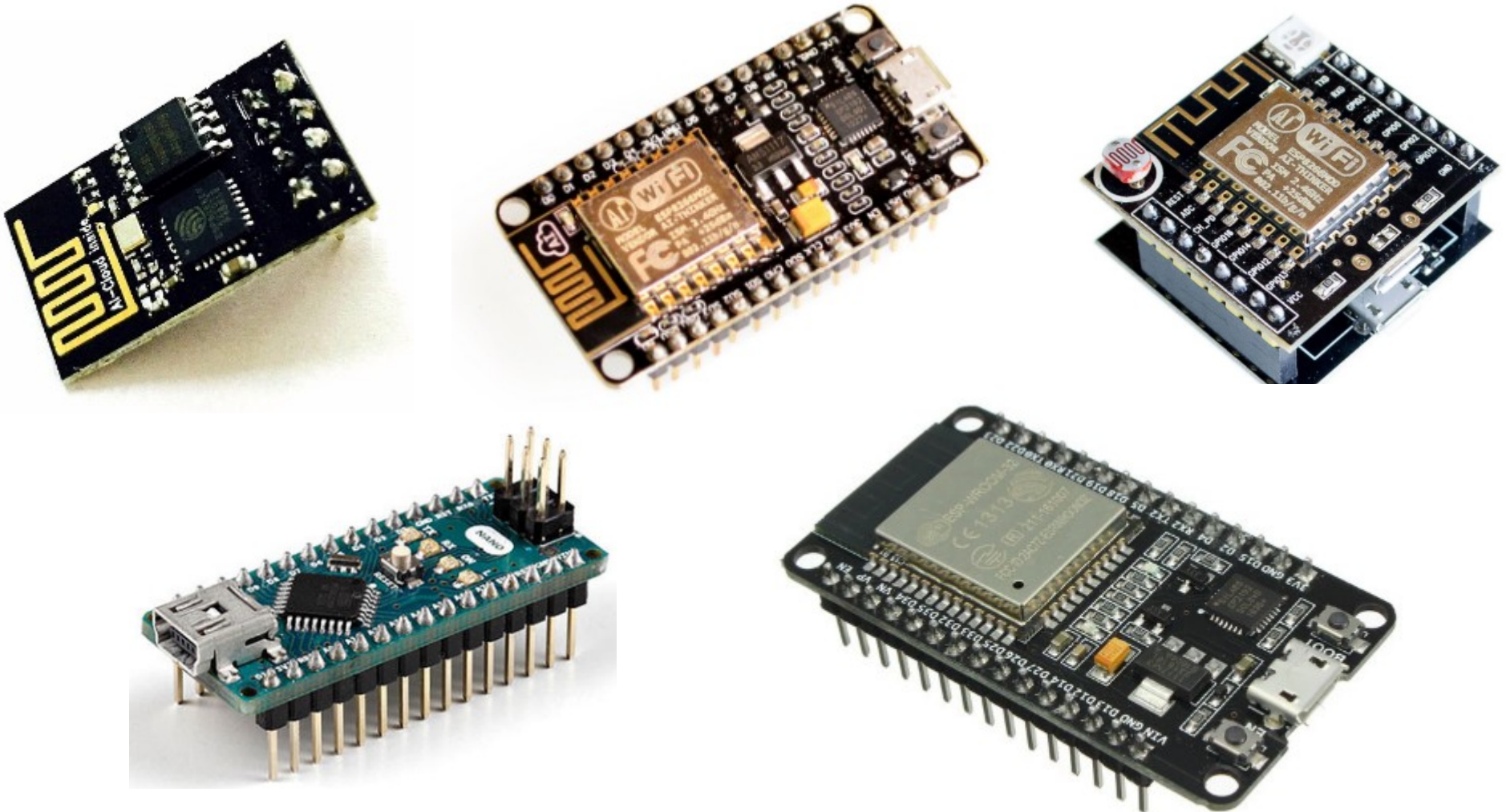






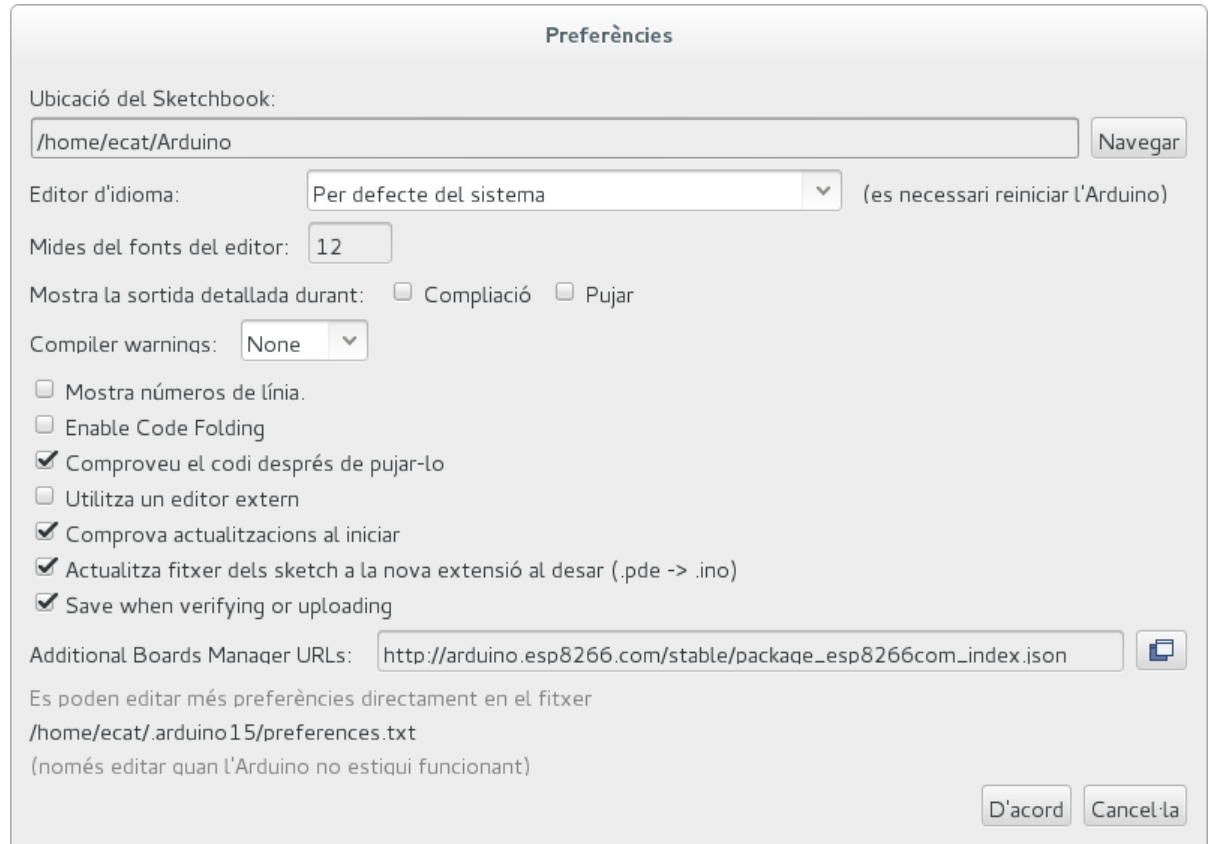
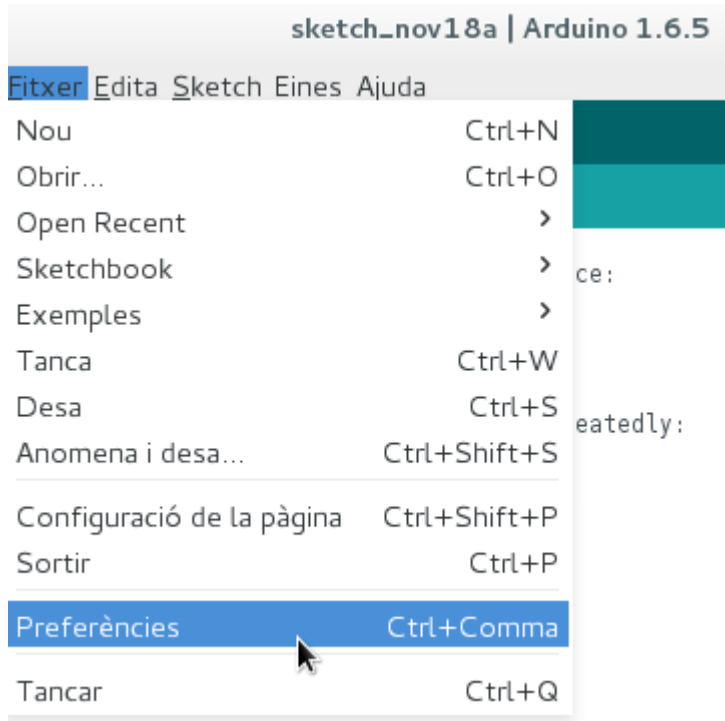
# Altres plaques programables des de l'IDE d'Arduino

**ESP8266 / ESP32 / STM32 / ...**





# ESP8266



Entreu aquesta adreça a "Additional Boards Manager URLs":  
[http://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](http://arduino.esp8266.com/stable/package_esp8266com_index.json)

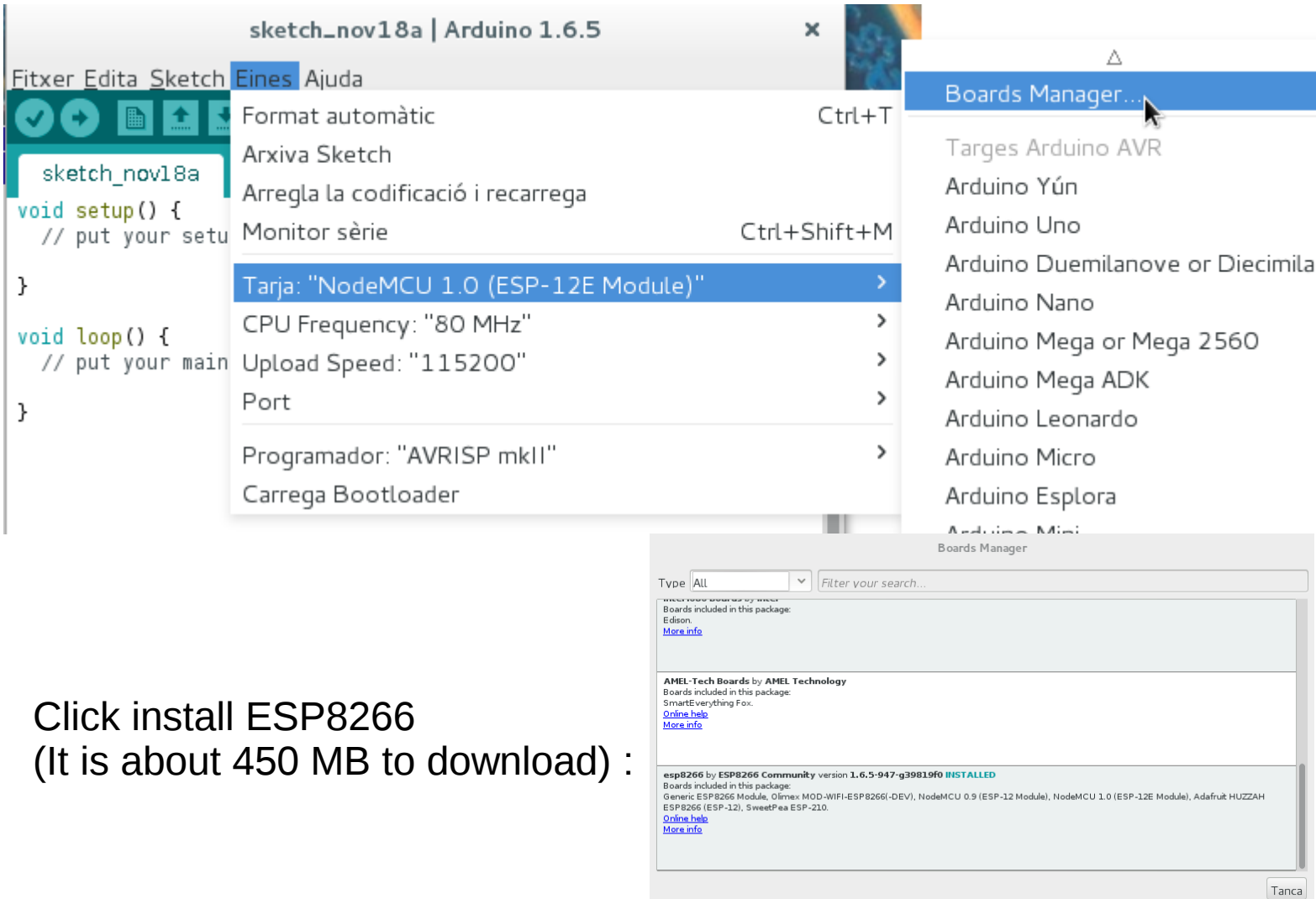






# ESP8266

Tools → Board: "current" → Boards Manager ...



Click install ESP8266  
(It is about 450 MB to download) :

Una altra manera de fer la instal·lació de l'IDE d'Arduino per a ESP8266:  
<https://libraries.io/github/adafruit/ESP8266-Arduino>





# ESP8266

## witty

```

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{
        *****
    } while( (millis() - ulPreviousMillis) < ulMilliseconds );
}

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

Microprogramari de verificació





# ESP8266

## witty

**/dev/ttyUSB0**

Envia

```

sent to UDP server
00010) 5c:cf:7f:c0:84:07
sent to UDP server
00011) 5c:cf:7f:c0:84:07
sent to UDP server
00012) 5c:cf:7f:c0:84:07
sent to UDP server
00013) 5c:cf:7f:c0:84:07
sent to UDP server
00014) 5c:cf:7f:c0:84:07
sent to UDP server
00015) 5c:cf:7f:c0:84:07
sent to UDP server
00016) 5c:cf:7f:c0:84:07
sent to UDP server

```

☒ Desplaçament automàtic

per la xarxa

webfa...

a un servi...

pyUdp

qt

qtPyU

s2a\_fr

scratc

Scrato

**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);

```

**UDPdlg**

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

```

Microprogramari client UDP





# ESP8266

## Trametent trames UDP amb l'ESP8266

UDPClientMAC02 | Arduino 1.6.5

Fitxer Edita Sketch Eines Ajuda

UDPClientMAC02

```
#include <ESP8266WiFi.h>
// #include <WiFiUDP.h>
#include <WiFiUDP.h>

#ifdef ESP8266
extern "C" {
#include "user_interface.h"
}
#endif

const char* ssid = "IoT-eCat";
const char* password = "clotClot";

// A UDP instance to let us send and receive packets over UDP
WiFiUDP Udp;

void setup() {
  Serial.begin(115200);
  delay(10);

  // We start by connecting to a WiFi network
  ....
}
```

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

```
void loop() {
  // char temp[20];      // buffer for for
  char temp[28];      // buffer for forma

  vGetMac(temp);

  Udp.write(temp);
  Udp.endPacket();

  Serial.print(temp);
  Serial.println(" sent to UDP server");

  delay(5000);
}
```





# MQTT



MQTT és una forma de comunicar dispositius d'Internet de les Coses (IdC / IoT) entre si. És un protocol lleuger i molt versàtil que es pot fer servir des d'un Arduino, una Raspberry Pi, un PC multinucli fins als serveis d'Amazon Web Services.

<https://www.baldengineer.com/mqtt-introduction.html>

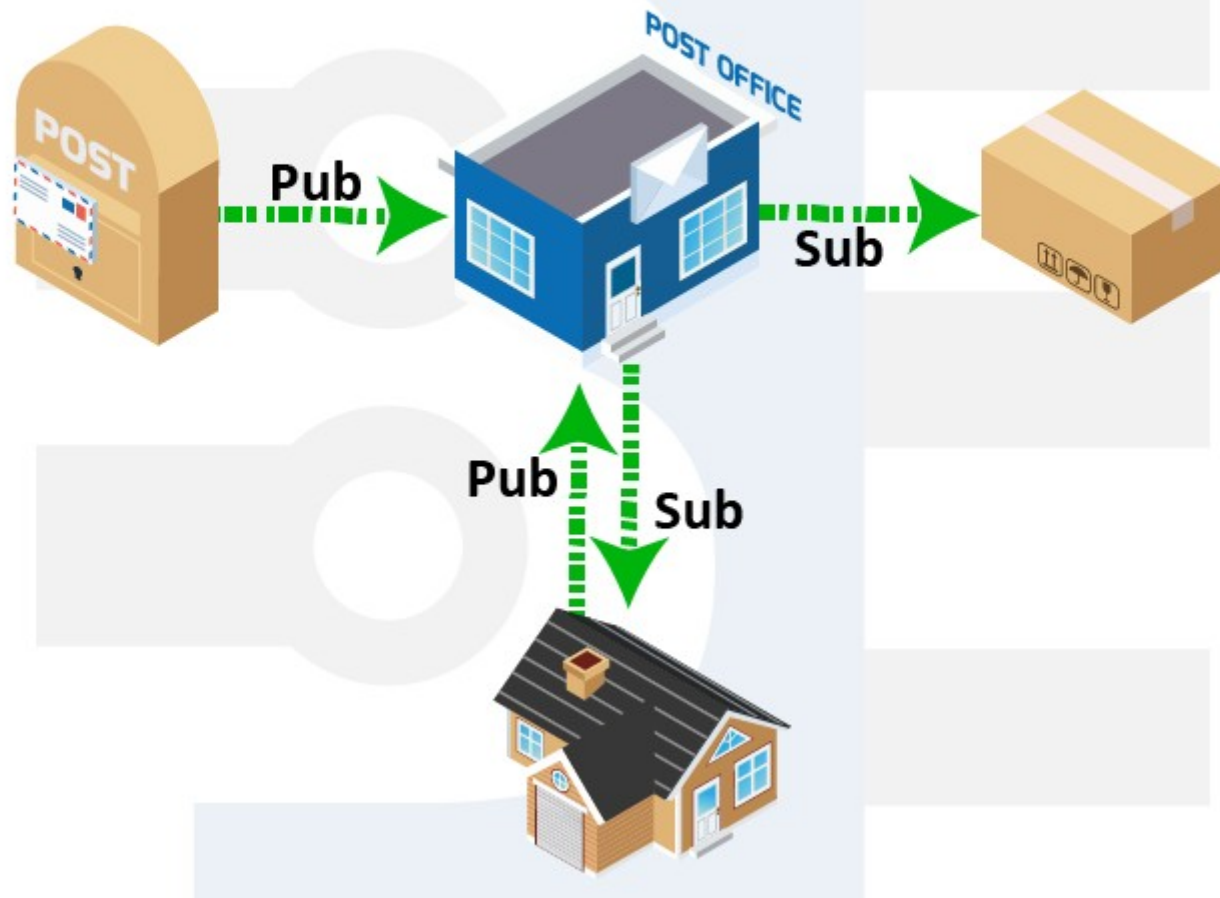
[http://binefa.cat/php/dam/m09uf3/20170224/2017024\\_mqtt\\_00.pdf](http://binefa.cat/php/dam/m09uf3/20170224/2017024_mqtt_00.pdf)



# MQTT

## Analogia amb el servei postal

mqtt: //broker/topic/message





# MQTT

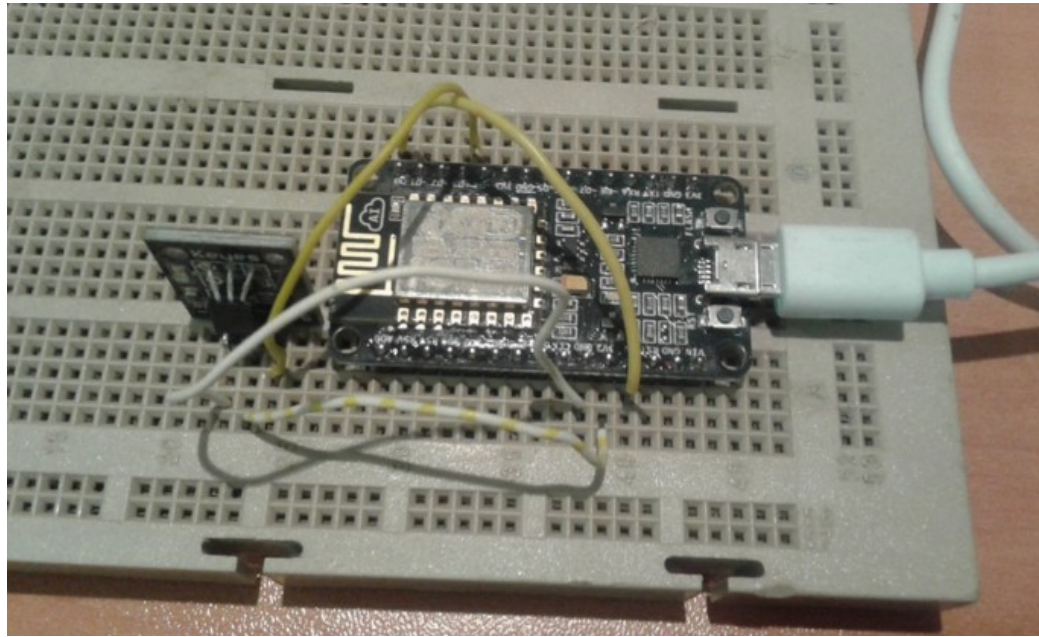
## Perquè no fer servir HTTP (o REST)

HTTP requereix de múltiples accions POST per a distribuir un missatge a més d'un client. L'objectiu del sistema de missatgeria intermediària és que l'intermediari distribueixi el missatge, tan sols als clients interessats. La funcionalitat del MQTT és trametre d'un a molts.



# MQTT

## ESP8266 sobre l'IDE d'Arduino - MQTT



```

Fitxer  Edita  Vi
Client mosqsub/15509-debian8: Entering deep sleep mode for 3 seconds...
Client mosqsub/15509-debian8: Connecting to IoT-eCat
24.94
Client mosqsub/15509-debian8: WiFi connected
25.00
Client mosqsub/15509-debian8: IP address:
24.94 192.168.1.12
Client mosqsub/15509-debian8: Attempting MQTT connection...connected
Client mosqsub/15509-debian8: Requesting DS18B20 temperature...
25.00 Sending temperature: 24.69
Client mosqsub/15509-debian8: Closing MQTT connection...
Client mosqsub/15509-debian8: Closing WiFi connection...
Client mosqsub/15509-debian8: Entering deep sleep mode for 3 seconds...
Client mosqsub/15509-debian8: Connecting to IoT-eCat
25.00
Client mosqsub/15509-debian8: ...
25.06
^C
acat@debian8:~$ mosquitto_sub -d -t sensors/test/temperature
Client mosqsub/15509-debian8 sending CONNECT
Client mosqsub/15509-debian8 received CONNACK
Client mosqsub/15509-debian8 sending SUBSCRIBE (Mid: 1, Topic:
Client mosqsub/15509-debian8 received SUBACK
Subscribed (mid: 1): 0
Client mosqsub/15509-debian8 received PUBLISH (d0, q0, r0, m0,
24.81
Client mosqsub/15509-debian8 received PUBLISH (d0, q0, r0, m0,
24.75
Client mosqsub/15509-debian8 received PUBLISH (d0, q0, r0, m0,
24.69
  
```

[https://binefa.cat/telecos/ESP8266/esp8266\\_03.pdf](https://binefa.cat/telecos/ESP8266/esp8266_03.pdf)

<http://www.jerome-bernard.com/blog/2015/10/04/wifi-temperature-sensor-with-nodemcu-esp8266/>

# Torn de preguntes ...



# ... i sessió pràctica.



# Arduino i Raspberry Pi

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

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

twitter



<https://twitter.com/JordiBinefa>

<https://twitter.com/electronicscat>



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

<https://wiki.binefa.cat>

<http://www.electronics.cat>

<http://www.makeit.cat>

**Moltes gràcies per la vostra atenció**