

20/05/2017

HC-06

Arduino Bluetooth Module

Miquel Àngel Román Colm

Taula de continguts

HC-06. Que és?	2
Premisses del circuit	3
Muntatge	5
Coses a tenir en compte	5
Creació del circuit	6
Muntatge lògic. Software.	8
Premisses del sistema	10
Els sketch.....	14
BT_RECEIVER.....	14
BT_SENDER	17
Altres dades d'interès.....	19
Llistat de comandes AT per al mòdul Bluetooth.....	19
Emprant la placa com a interfície per configurar el mòdul Bluetooth	20

HC-06. Que és?

HC-06 és un mòdul electrònic que té com a funcionalitat convertir les senyals Bluetooth a senyals de tipus Serial (TTL).

Algunes de les propietats més destacables són:

- No consumeix molta energia
- És barat
- No necessita antena
- Programable mitjançant comandes AT
- Funciona amb voltatges baixos

L'HC-06 és una altra versió d'un altre mòdul de connexió Bluetooth l'HC-05.

Aquesta versió és més simple ja que en comptes de tenir 6 pins, en té 4.

VCC	GND	RX	TX
Voltatge d'alimentació. 3.6V cap a 6V	<i>Ground.</i> Connexió a terra.	Pin receptor	Pin Transmissor

Tant els pins RX com TX treballen a 3,3 V i en alguns casos seria necessari emprar un divisor de voltatge.

Premisses del circuit

En aquest cas; com que treballarem amb una *Arduino Nano* les especificacions són les següents:

Technical specs

Microcontroller	ATmega328
Architecture	AVR
Operating Voltage	5 V
Flash Memory	32 KB of which 2 KB used by bootloader
SRAM	2 KB
Clock Speed	16 MHz
Analog I/O Pins	8
EEPROM	1 KB
DC Current per I/O Pins	40 mA (I/O Pins)
Input Voltage	7-12 V
Digital I/O Pins	22
PWM Output	6
Power Consumption	19 mA
PCB Size	18 x 45 mm
Weight	7 g
Product Code	A000005

Figura1: Especificacions de la placa Nano extretes de arduino.cc

Pel que podem observar:

Operating Voltage	5 V
-------------------	-----

Figura 2: La placa opera amb voltatges de 5V, i els pins RX i TX del mòdul només treballen amb 3,3V.

Anem a esbrinar quina seria la resistència necessària a afegir per no fer mal bé el mòdul.

Per la llei d'Ohm:

$$I = \frac{V}{R}$$

La intensitat que tenen els I/O pins de la placa és la següent:

DC Current per I/O Pins	40 mA (I/O Pins)
-------------------------	------------------

Figura 3: Amperatge de la placa en els pins I/O

Per tant ens queda la següent equació:

$$I = 40\text{mA} = 40 \cdot 10^{-3} \text{ A}; \quad V = 3,3\text{V} \quad i \quad R = R. \text{ (Valor desconegut)}$$

$$40 \cdot 10^{-3} \times R = 3,3$$

$$R = 82,5 \, \Omega$$

Aquest valor és tan petit que és despreciable.

Per tant no fa falta afegir cap circuit extra ni divisors de voltatge per els exemples següents.

Muntatge

Coses a tenir en compte

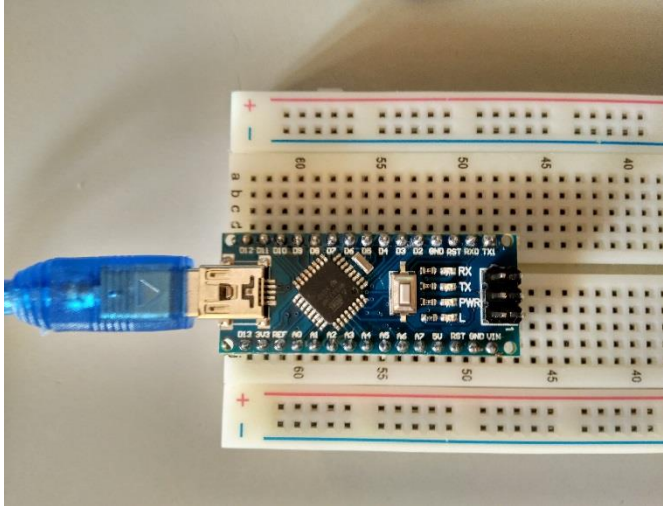
El següent muntatge és el corresponent per poder emprar el *built-in serial* de la placa com a port de transmissió entre el mòdul i la Arduino.

La manera de fer ús del *built-in serial* és emprant l'objecte **Serial** i els seus mètodes corresponents que ens ofereix l'IDE de la Arduino.

Els codis d'exemple que figuren després d'aquest apartat de muntatge s'haurien d'adaptar amb la llibreria *SoftwareSerial* en cas de que no es volgués manipular el mòdul amb el *built-in serial*.

Creació del circuit

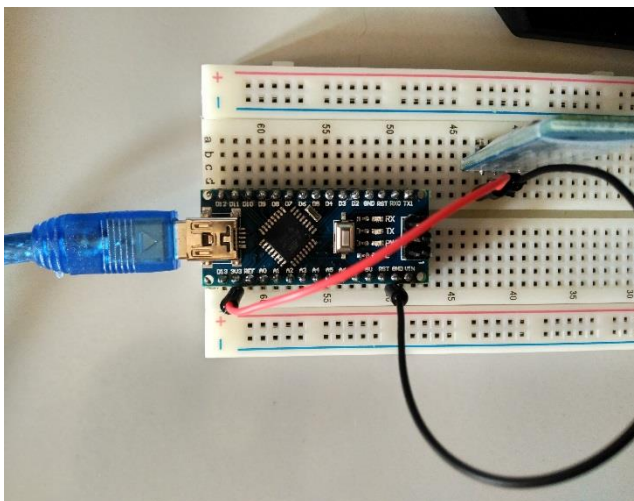
Com bé s'ha explicat abans, no fa falta afegir cap divisor de voltatge; llavors procedim:



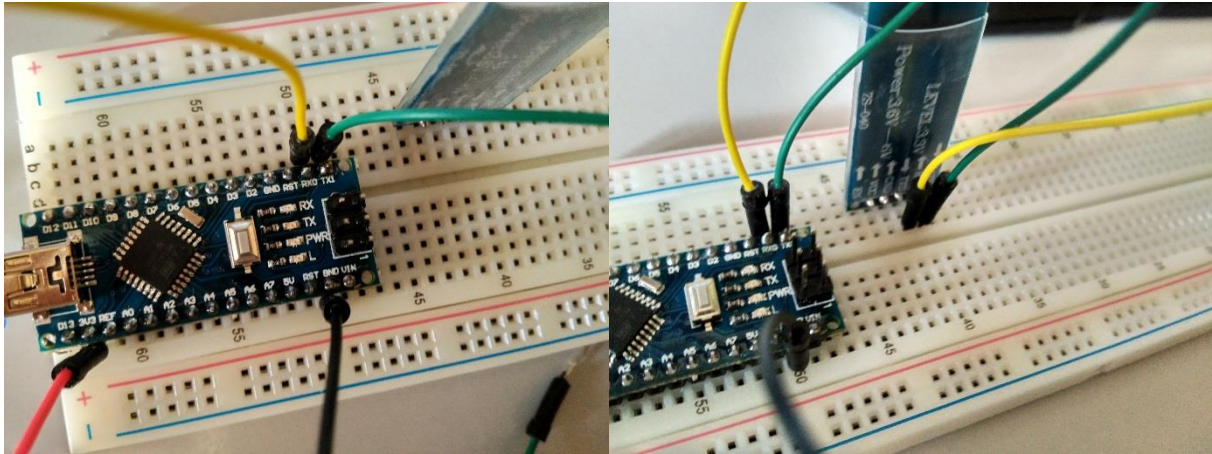
Muntam la placa a la *protoboard*



Muntam el mòdul a la *protoboard*

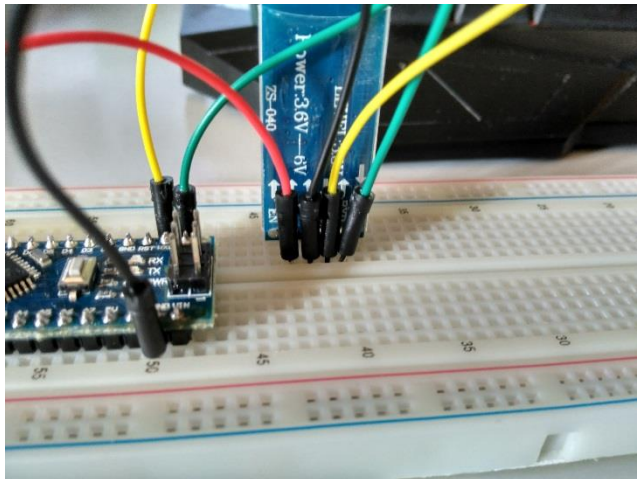


Conectam el pin VCC del mòdul al pin 3V3 de la *Arduino Nano* (cable vermell) i es conecta un dels pins GND de la placa amb el GND corresponent del mòdul (cable negre).



PINS		
CABLE	ARDUINO NANO	HC-06
GROC	RX	TX
VERD	TX	RX

D'aquesta manera la placa rebrà (Arduino RX) l'output (BT TX) del mòdul i el podrà mostrar per pantalla, fent ús del monitor sèrie de l'IDE.



Així és com hauria de quedar al final segons les especificacions donades.

Muntatge lògic. Software.

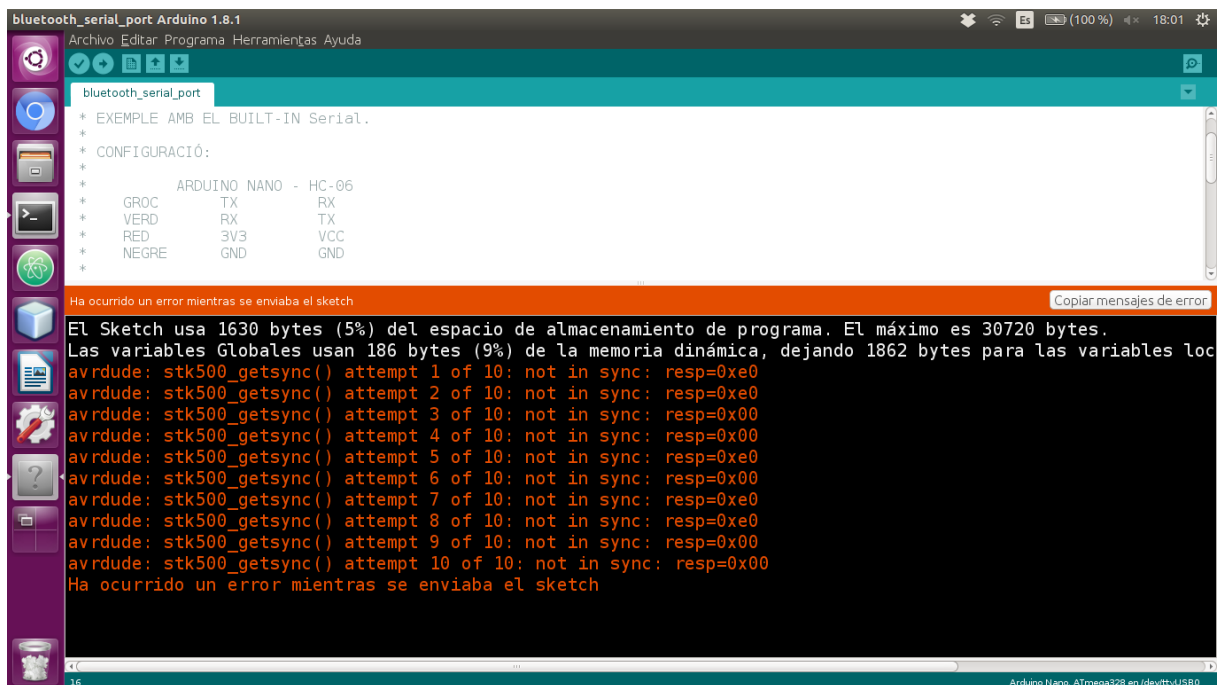
Principalment tindrem dos sketch molt bàsics però realment il·lustratius.

Són dos sketch en els quals es mostren dos programes molt simples on es realitzen enviaments i recepcions de dades. El procediment és sempre el mateix: Escriure/Llegir en el *serial-port*.

Abans de mostrar els sketch i els corresponents resultats; s'explicarà un error molt comú.

Errors

El que qualsevol faria ara, seria compilar el codi i intentar carregar-lo a la placa. Però obtindria el següent resultat:



Això es deu a que l'IDE no sap on ha de carregar el programa. El *built-in serial port* correspon tant a la connexió USB com als pins RX/TX de la placa els quals actualment estan connectats; i per això podem emprar el mateix objecte *Serial* tant per mostrar per pantalla com per escriure al mòdul de Bluetooth si hem realitzat les connexions tal com s'ha especificat abans.

Per tant, aquest error ens sortirà sempre que intentem carregar el codi a la placa i els pins RX/TX estiguin essent emprats i si el mòdul (independentment del mòdul que sigui) estigués connectat a través d'altres pins amb la llibreria *SoftwareSerial* això no ens passaria.

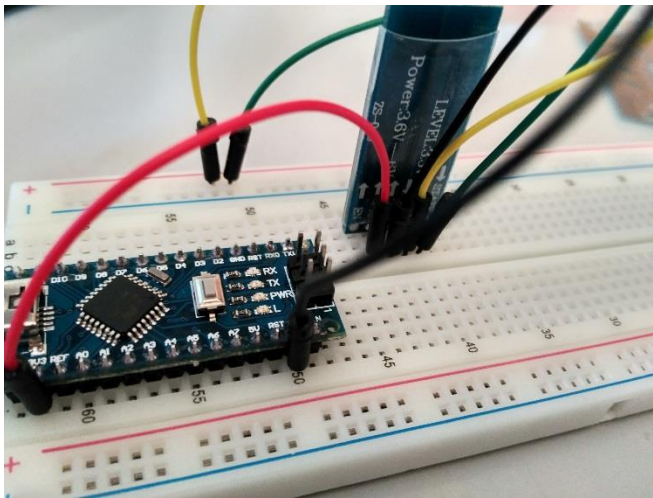
La solució és realment senzilla:

Desconectar el mòdul de Bluetooth de la alimentació

o bé

Desconectar els cables que connecten els RX's i TX's

En aquest cas vaig triar la segona opció.



```
bluetooth_serial_port Arduino 1.8.1
Archivo Editar Programa Herramientas Ayuda

bluetooth_serial_port

* EJEMPLO CON EL BUILT-IN Serial.
* CONFIGURACIÓ:
*
* ARDUINO NANO - HC-06
* GROB TX RX
* VERD RX TX
* RED 3V3 VCC
* NEGRE GND GND
*
*****/
void setup() {
  // Inicialitzam el Built-In Serial.
  Serial.begin(9600);
}
void loop() {
  // Mentre hi hagi bytes per llegir, els llegim
  if(Serial.available() > 0){
    while(Serial.available() > 0){
      Serial.print((char) Serial.read());
    }
  }
}

Subido
El Sketch usa 1630 bytes (5%) del espacio de almacenamiento de programa. El máximo es 30720 bytes.
Las variables Globales usan 186 bytes (9%) de la memoria dinámica, dejando 1862 bytes para las variables loc

16 Arduino Nano, ATmega328 en /dev/ttyUSB0
```

El problema resolts.

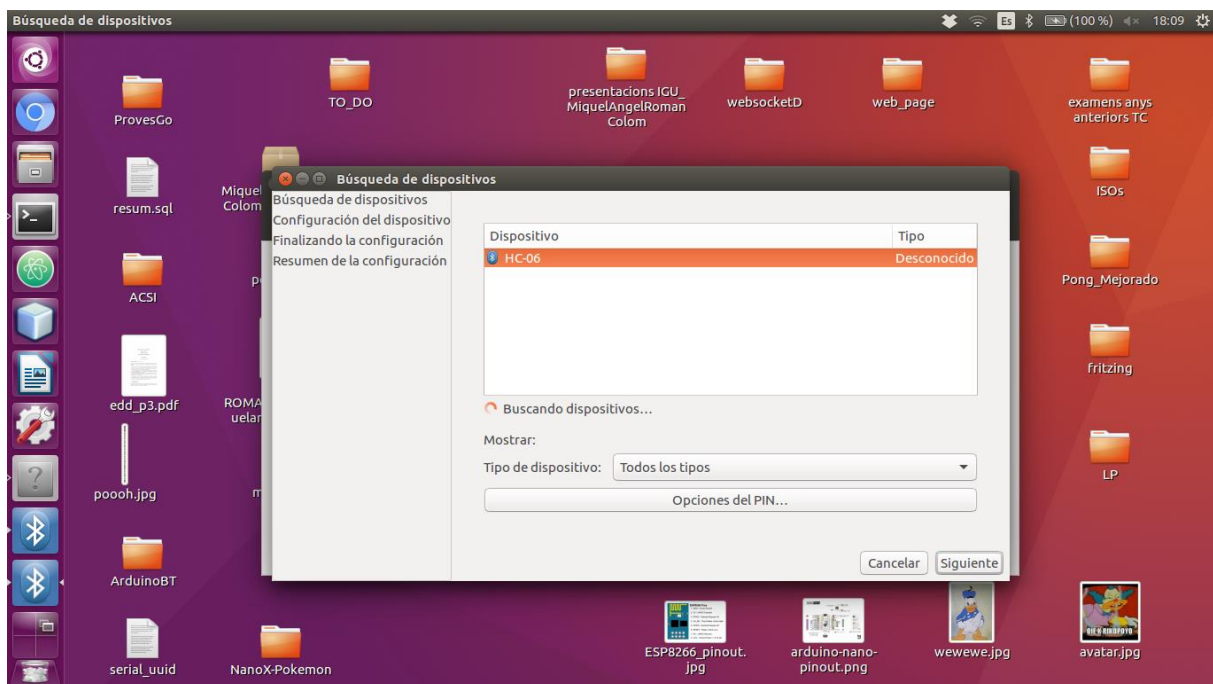
Una vegada carregat el codi a la placa s'han de refer les connexions tal i com estaven.

Premisses del sistema

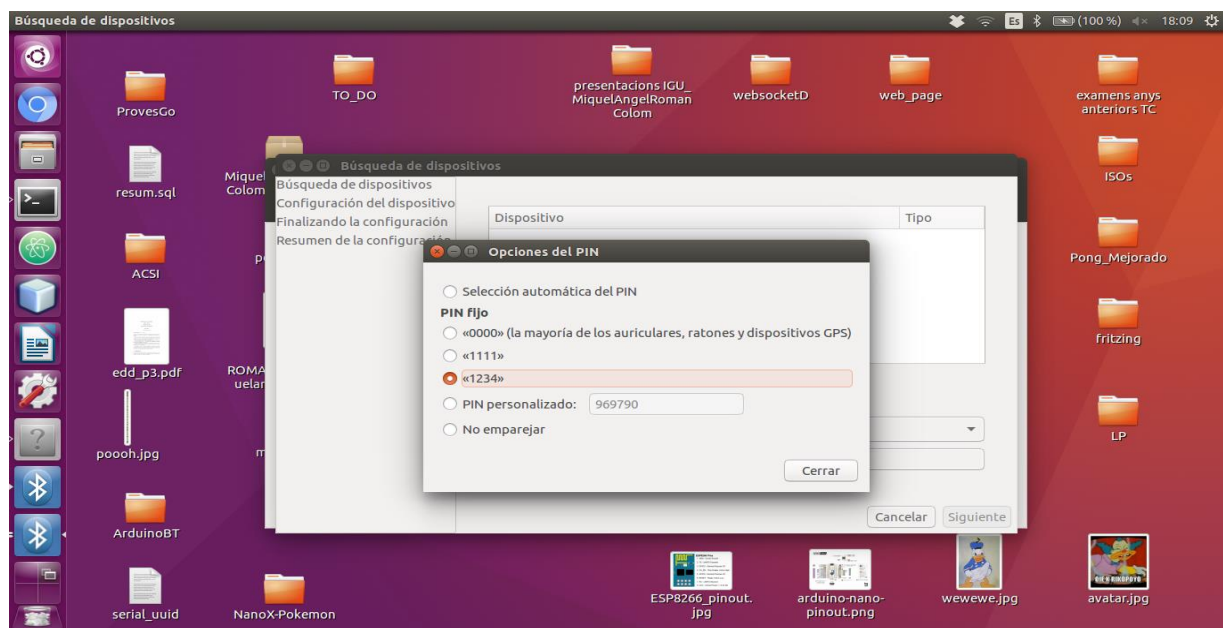
Abans de poder fer funcionar qualsevol cosa serà necessari emparellar els dos dispositius: el mòdul amb el dispositiu amb el qual volem que es comuniqui.

Com que el procediment sempre és el mateix, es demostrarà mitjançant un sistema *Ubuntu 16.04 LTS* amb un mòdul Bluetooth intern.

Trobar el dispositiu que volem emparellar. El mòdul Bluetooth necessita estar en marxa. (ON)



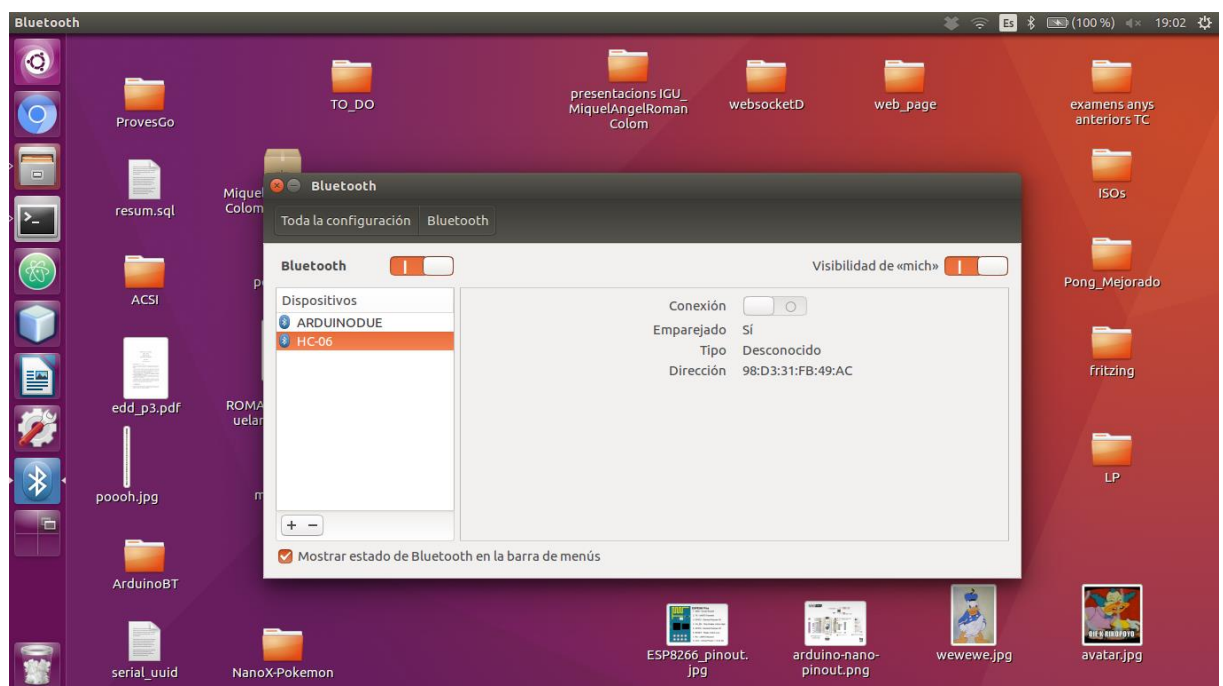
Introduir el codi PIN corresponent. En aquest cas 1234.

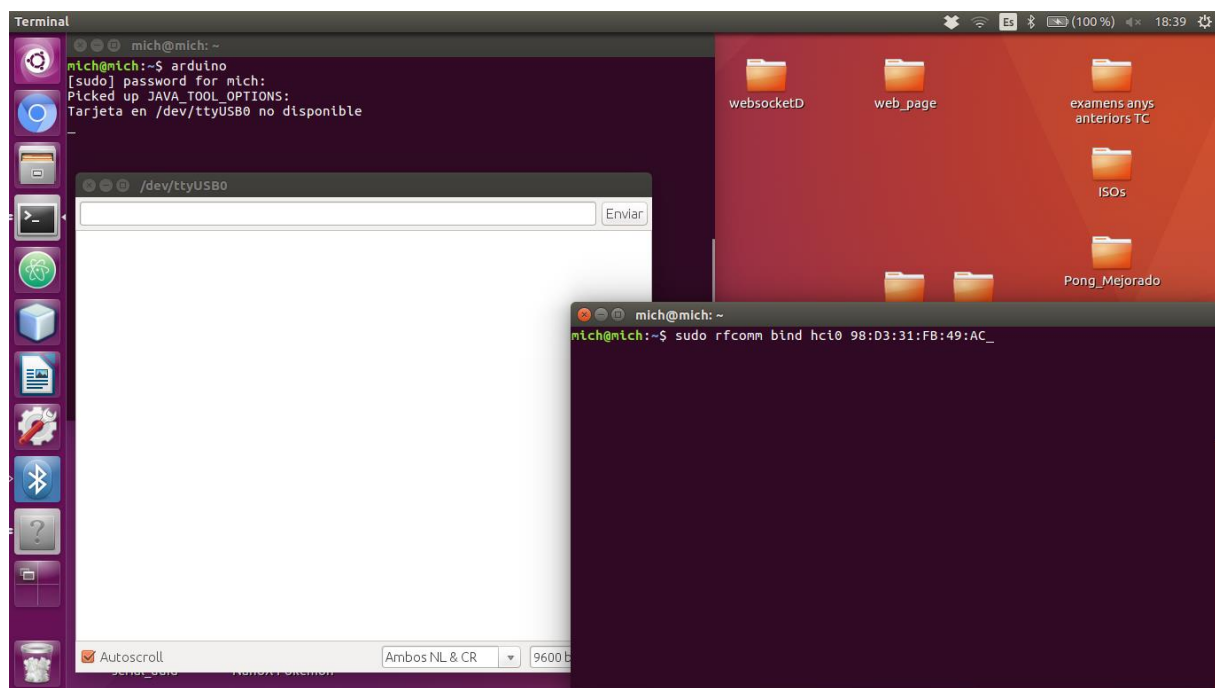


Un cop emparellats els dispositius, necessitem obrir el port sèrie de la següent manera:

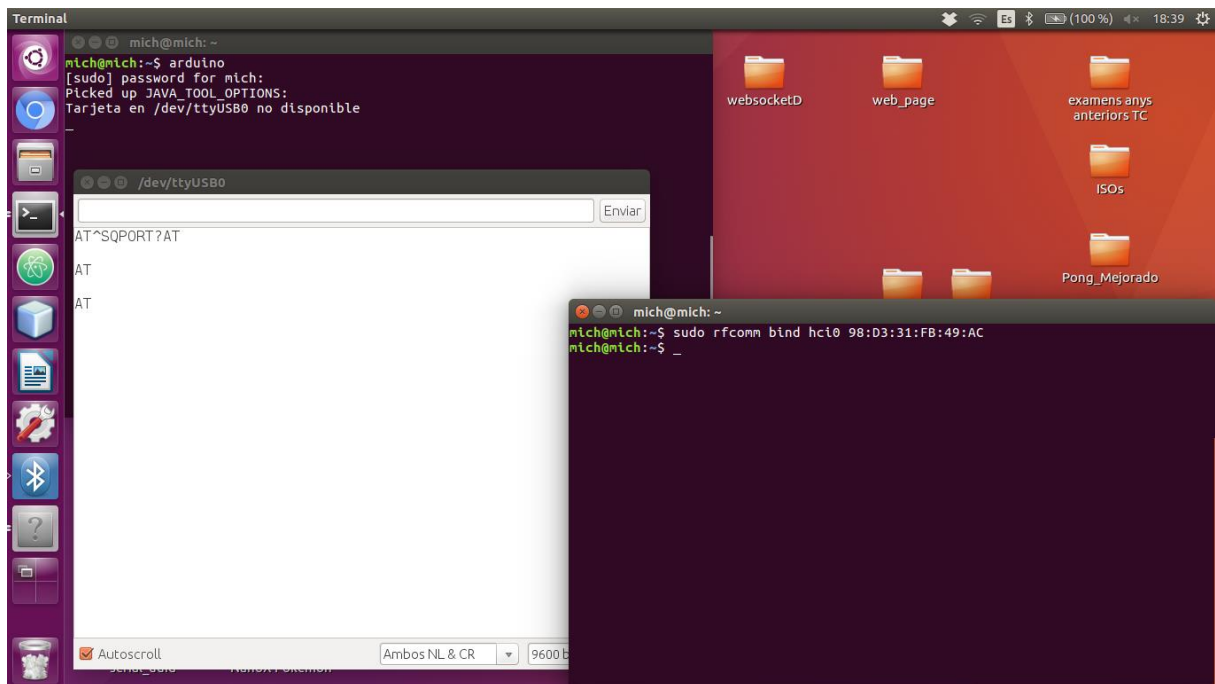
```
Sudo rfcomm bind [disp._bluetooth_sistema] [adreça_MAC_HC-06]
```

La adreça MAC pot ésser consultada en qualsevol moment.



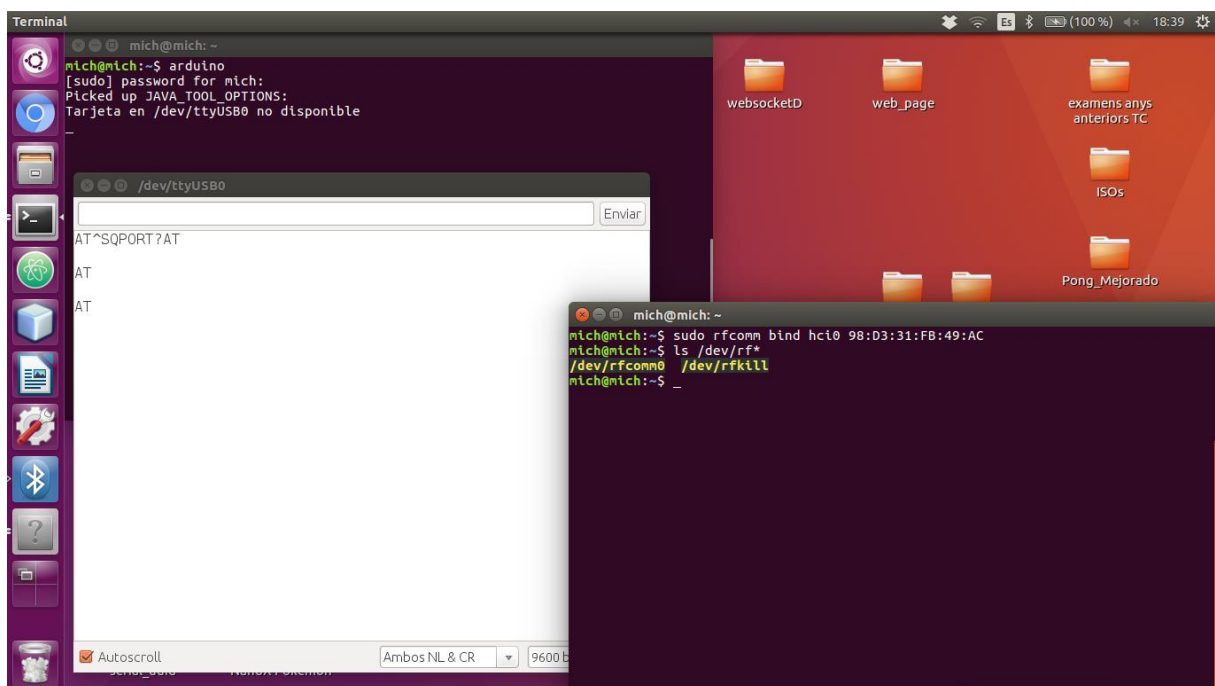


En la següent imatge podem veure que el mòdul ja ha rebut informació mentre s'obria el port sèrie.



Apreciem que s'ha creat un nou "arxiu", *rfcommO*, al subdirectori */dev/*.

Aquest arxiu és on s'escriurà i d'on es llegirà per poder tenir comunicació entre els dos dispositius.



Els sketch

BT_RECEIVER

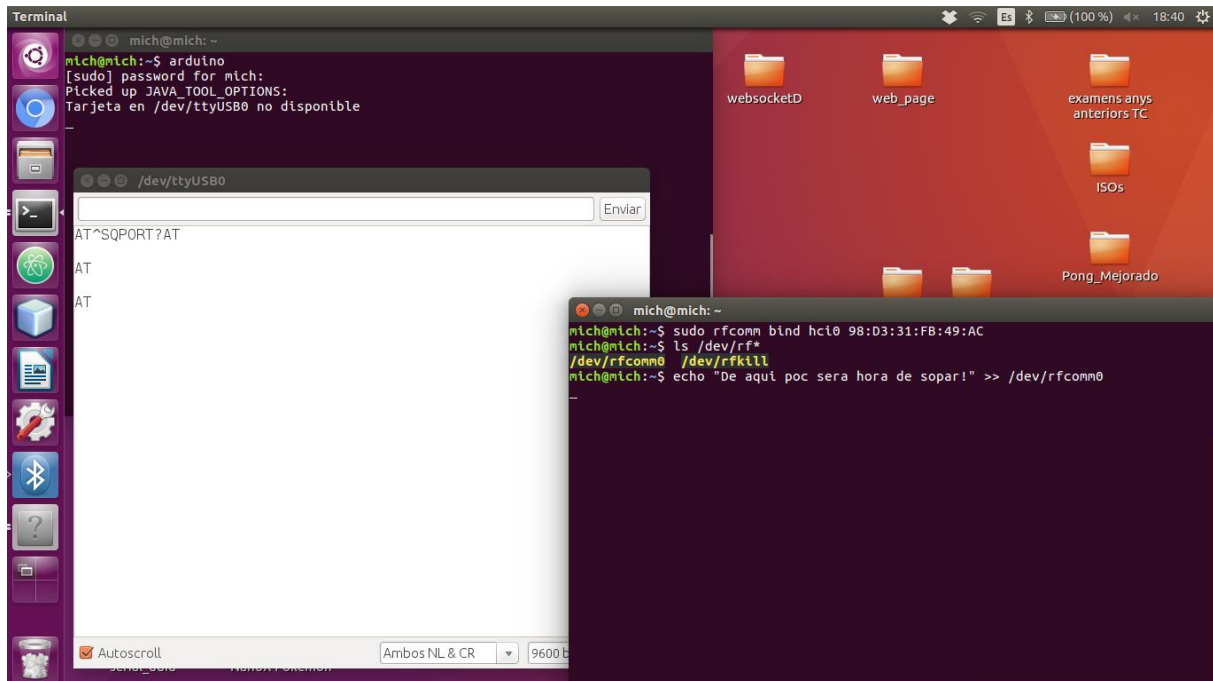
Un senzill programa que revisa que hi hagi contingut al port sèrie cada dos segons i que en cas de que sí que hi hagi alguna cosa, la llegeix i la mostra per el monitor sèrie de l'Arduino IDE.

```
/* *****  
 *  
 *  EXEMPLE AMB EL BUILT-IN Serial.  
 *  
 *  CONFIGURACIÓ:  
 *  
 *          ARDUINO NANO - HC-06  
 *  GROC      TX      RX  
 *  VERD      RX      TX  
 *  RED       3V3     VCC  
 *  NEGRE     GND     GND  
 *  
 * ***** */  
void setup() {  
  // Inicialitzam el Built-In Serial.  
  Serial.begin(9600);  
}  
void loop() {  
  // Mentre hi hagi bytes per llegir, els llegim  
  if(Serial.available() > 0){  
    while(Serial.available() > 0){  
      Serial.print((char) Serial.read());  
    }  
    Serial.println(); Serial.println();  
  }  
  delay(2000);  
}
```

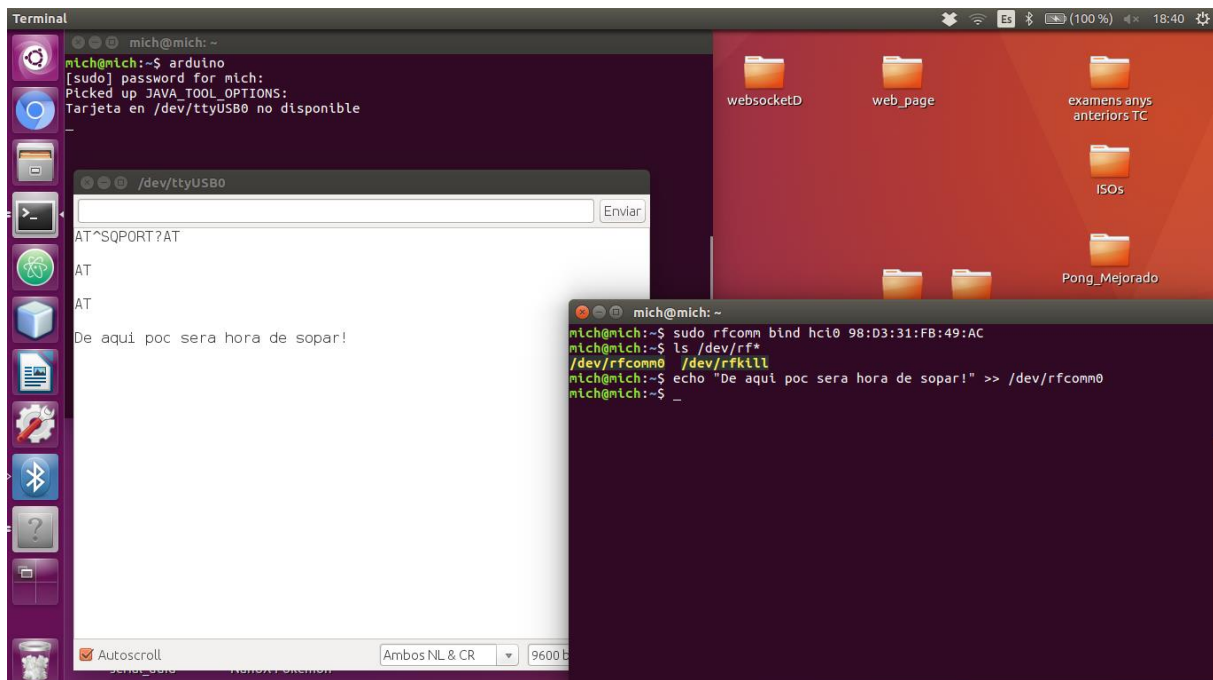
Ara que tenim el programa en marxa només falta escriure alguna cosa al port sèrie i veure la comunicació!

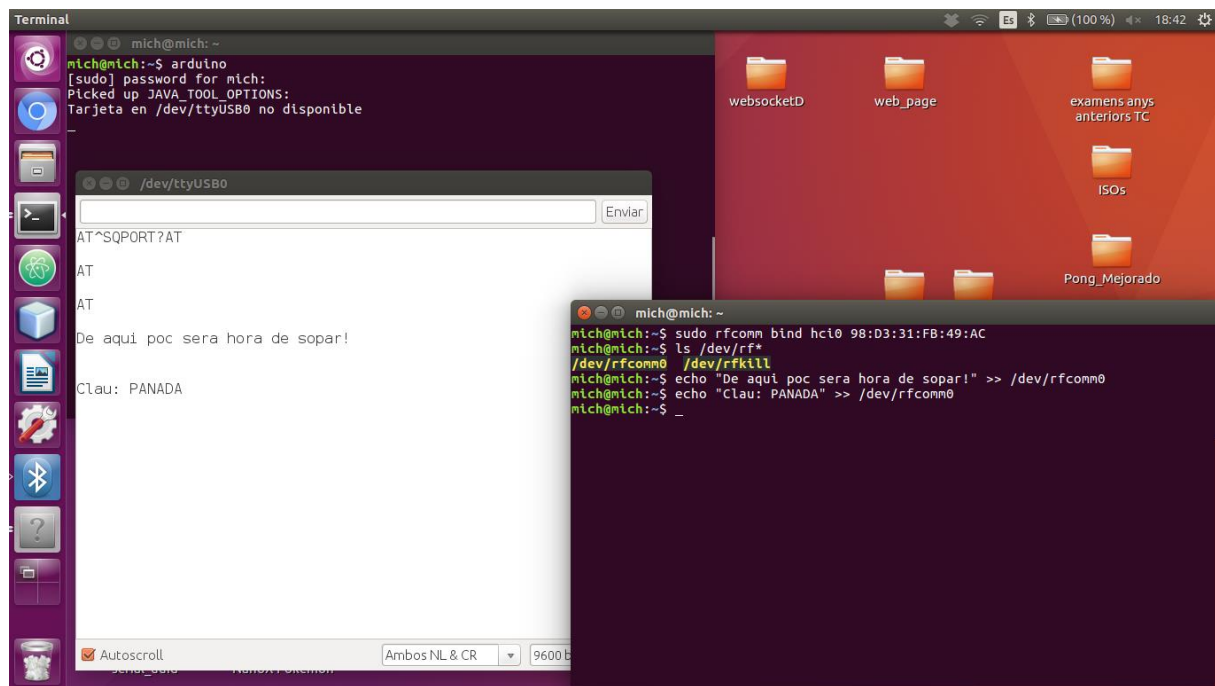
Escrivim alguna cosa a l'arxiu emprant la comana "echo" i l'operador de concatenació a l'arxiu.

echo "missatge" >> /dev/rfcomm0



I ara la Arduino reb el missatge, aquest és imprès al monitor sèrie de la placa.





BT_SENDER

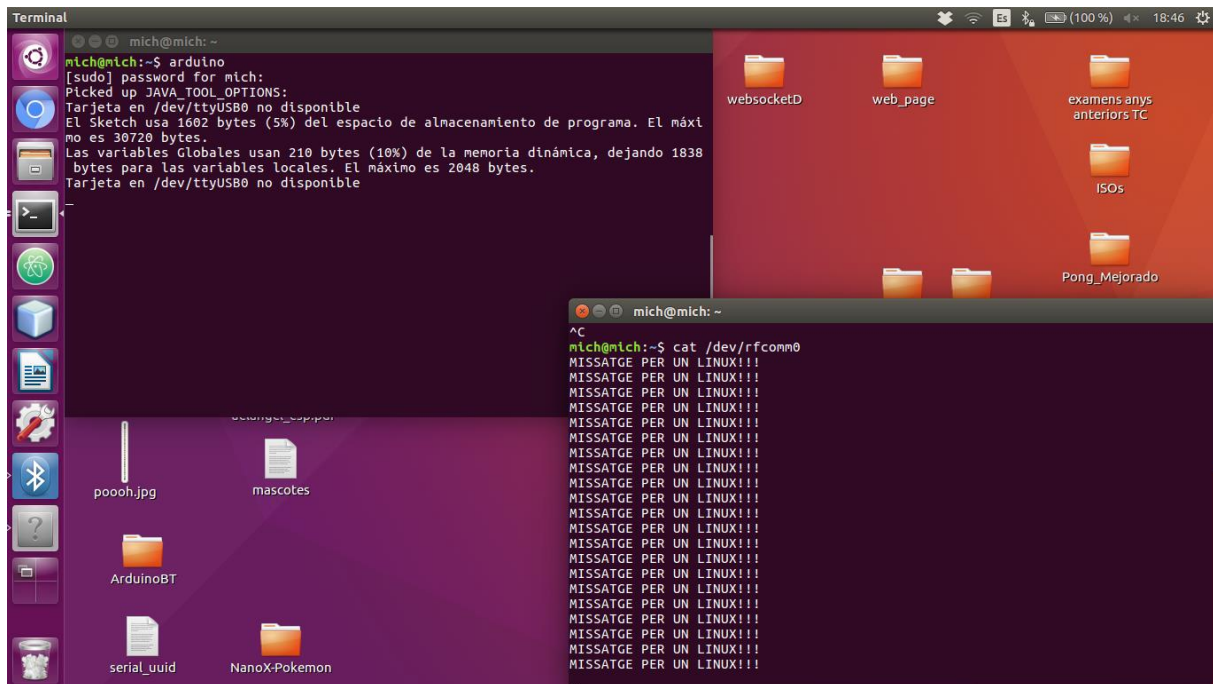
Un senzill programa que envia, en bucle, missatges al port sèrie.

```
/* *****  
 *  
 *  EXEMPLE AMB EL BUILT-IN Serial.  
 *  
 *  CONFIGURACIÓ:  
 *  
 *          ARDUINO NANO - HC-06  
 *  GROC      TX      RX  
 *  VERD      RX      TX  
 *  RED       3V3     VCC  
 *  NEGRE     GND     GND  
 *  
 ***** */  
void setup() {  
  // Inicialitzam el Built-In Serial.  
  Serial.begin(9600);  
}  
void loop() {  
  // Enviam el missatge desitjat  
  Serial.println("MISSATGE PER UN LINUX!!!");  
  delay(2000);  
}
```

Com que aquest sketch només realitza enviament de dades, l'output només es veurà al sistema *Ubuntu*.

Emprem la comana *cat* per visualitzar el contingut del port sèrie:

cat /dev/rfcomm0



Altres dades d'interès

LES COMANDES AT NO DUEN CAP SEQÜENCIA D'ACABAMENT (\n\r)

Llistat de comandes AT per al mòdul Bluetooth

N'existeixen 3 més, però les més interessants són les següents:

- AT
Serveix per comprovar la connectivitat. Resposta: OK
- AT+BAUD[1-9/ABC]
Serveix per canviar els baudis del mòdul.

1-----	1200
2-----	2400
3-----	4800
4-----	9600 (Default)
5-----	19200
6-----	38400
7-----	57600
8-----	115200
9-----	230400
A-----	460800
B-----	921600
C-----	1382400

Taula de baud ratios del HC-06.

Exemple: AT+BAUD3
Resposta: OK4800

- AT+NAME[nou_nom]
Serveix per canviar el nom visible del mòdul.
Exemple: AT+NAMEArduinoNano
Resposta: OKsetname
- AT+PIN[xxxx]
Serveix per canviar el codi PIN del mòdul.
Exemple: AT+PIN3234
Resposta: OKsetpin
- AT+VERSION
Mostra la versió del *firmware* del mòdul.

Emprant la placa com a interfície per configurar el mòdul Bluetooth

Apart dels dos sketch comentats anteriorment se'n ha programat un altre que permet configurar el mòdul Bluetooth a través del monitor sèrie i la entrada de teclat des de l'ordenador.

La idea del Sketch és realment senzilla:

Aïllar el mòdul del *built-in serial port* fent ús de la llibreria *SoftwareSerial*. Col·locarem l'HC-06 a altres pins de la placa que no siguin els RX i TX i ho indicarem al constructor del objecte **SoftwareSerial**.

Tot aquest procediment està suficientment il·lustrat i comentat a l'sketch BT_CONFIG.ino en el subdirectori BT_CONFIG/ adjunt a aquesta documentació.

Llavors només ens falta llegir del monitor sèrie les comanes, ***SENSE \r\n ni \n ni res***, i pasarl la comana llegida a l'HC-06 mitjançant el mètode .print() aplicable al objecte instanciat de la classe *SoftwareSerial*.

Com que aquest sketch és un poc més llarg que els altres no s'incluran captures del codi i es demana al lector que revisi el codi adjunt ja que està documentat.