

Treball Final de Grau

*Control de la temperatura i el llum
mitjançant dispositius sense fils*

ANNEXOS

Ramon Altarriba Roca

Grau en Enginyeria Mecatrònica

Director: Jaume Cabaní Masip

Vic, gener de 2015

Índex

Índex.....	2
Índex Figures	4
Índex equacions.....	6
Annex 1 . Estudi i elecció dels diferents components de la xarxa	7
1.1 Elecció components de la placa	7
1.1.1 Esquema elèctric sensor temperatura.....	7
1.1.2 Esquema elèctric sensor llum	7
1.1.3 Esquema elèctric sensor bateria	8
1.1.4. Configuració regulador de tensió	8
Annex 2 . Proves prèvies	10
2.1 Arduino	10
2.2 Xbee	13
2.2.1 Material necessari	13
2.2.2 Reset.....	14
2.3 Arduino + Xbee	17
2.3.1 Comunicació punt a punt.....	17
2.3.2 Estudi trama API	28
2.3.3 Introducció xarxa mode API	31
2.3.4 Lectura potenciòmetre mitjançant trames API.....	36
2.3.5 Lectura potenciòmetre mitjançant trames API i activar Led	44
2.4 Configuració Arduino Ethernet.....	49
Annex 3 . Implantació Xarxa.....	51
3.1 Configuració mòduls Xbee.....	51
3.1.1 Configuració mòdul NI0101 (Sensors + relé ventilador).....	51
3.1.2 Configuració mòdul RI0101 (Relé caldera)	54
3.1.3 Configuració mòdul RI0102 (Relé llum).....	57
3.1.4 Configuració mòdul COORDINADOR	61
3.2 Codi Arduino	64
3.3 Disseny de les plaques de circuit imprès	72
3.4 Disseny base de dades i control remot.....	75
3.4.1 Disseny base de dades	75
3.4.2 connexio.php.....	78
3.4.3 calendari.php.....	78

3.5 Disseny de les caixes	80
3.5.1 Plànol tapa inferior comú.....	80
3.5.2 Plànol tapa superior mòdul sensor	81
3.5.3 Plànol tapa superior mòdul relé.....	82
3.5.4 Plànol tapa inferior mòdul coordinador.....	83
3.5.5 Plànol tapa superior mòdul coordinador.....	84
3.5.6 Tenyit de les caixes.....	85

Índex Figures

Figura 1-1. Esquema elèctric sensor temperatura.....	7
Figura 1-2. Esquema elèctric sensor llum	7
Figura 1-3. Esquema elèctric sensor bateria	8
Figura 1-4. Relació Voltatge sortida - valors Resistències	8
Figura 2-1. Fragment codi Arduino.....	10
Figura 2-2. Fragment codi Arduino.....	10
Figura 2-3. Fragment codi Arduino.....	10
Figura 2-4. Fragment codi Arduino.....	11
Figura 2-5. Fragment codi Arduino.....	11
Figura 2-6. Fragment codi Arduino.....	11
Figura 2-7. Fragment codi Arduino.....	12
Figura 2-8. Material necessari per adaptar mòdul Xbee al protoboard.....	13
Figura 2-9. Captura pantalla del procediment a seguir per fer un Reset	14
Figura 2-10. Captura pantalla del procediment a seguir per fer un Reset	15
Figura 2-11. Realització Reset manualment	15
Figura 2-12. Captura pantalla del procediment a seguir per fer un Reset	16
Figura 2-13. Captura pantalla configuració mòdul Xbee Punt a Punt.....	17
Figura 2-14. Captura pantalla configuració mòdul Xbee Punt a Punt.....	18
Figura 2-15. Captura pantalla configuració mòdul Xbee Punt a Punt.....	19
Figura 2-16. Captura pantalla configuració mòdul Xbee Punt a Punt.....	20
Figura 2-17. Captura pantalla configuració mòdul Xbee Punt a Punt.....	21
Figura 2-18. Captura pantalla configuració mòdul Xbee Punt a Punt.....	23
Figura 2-19. Captura pantalla configuració mòdul Xbee Punt a Punt.....	25
Figura 2-20. Captura pantalla codi Arduino prova Punt a Punt	26
Figura 2-21. Captura de pantalla resultat final prova Punt a Punt.....	27
Figura 2-22. Captura pantalla Trama API a analitzar	28
Figura 2-23. Captura pantalla configuració mòdul Xbee xarxa API	31
Figura 2-24. Captura pantalla configuració mòdul Xbee xarxa API	32
Figura 2-25. Captura pantalla configuració mòdul Xbee xarxa API	33
Figura 2-26. Captura pantalla configuració mòdul Xbee xarxa API	34
Figura 2-27. Captura pantalla configuració mòdul Xbee xarxa API	35
Figura 2-28. Muntatge circuit assaig lectura potenciòmetre	37

Figura 2-29. Captura pantalla config. mòdul Xbee lectura potenciòmetre	38
Figura 2-30. Captura pantalla config. mòdul Xbee lectura potenciòmetre	39
Figura 2-31. Captura pantalla config. mòdul Xbee lectura potenciòmetre	40
Figura 2-32. Captura pantalla config. mòdul Xbee lectura potenciòmetre	41
Figura 2-33. Muntatge del circuit de la prova Lectura potenciòmetre mitjançant trames API i activar Led	46
Figura 2-34. Captura pantalla configuració Arduino Ethernet.....	49
Figura 2-35. Captura pantalla configuració Arduino Ethernet.....	49
Figura 2-36. Captura pantalla configuració Arduino Ethernet.....	50
Figura 3-1. Captura pantalla configuració mòdul NI 0101	51
Figura 3-2. Captura pantalla configuració mòdul NI 0101	51
Figura 3-3. Captura pantalla configuració mòdul NI 0101	52
Figura 3-4. Captura pantalla configuració mòdul NI 0101	52
Figura 3-5. Captura pantalla configuració mòdul NI 0101	52
Figura 3-6. Captura pantalla configuració mòdul NI 0101	53
Figura 3-7. Captura pantalla configuració mòdul NI 0101	53
Figura 3-8. Captura pantalla configuració mòdul NI 0101	54
Figura 3-9. Captura pantalla configuració mòdul RI 0101	54
Figura 3-10. Captura pantalla configuració mòdul RI 0101	55
Figura 3-11. Captura pantalla configuració mòdul RI 0101	55
Figura 3-12. Captura pantalla configuració mòdul RI 0101	55
Figura 3-13. Captura pantalla configuració mòdul RI 0101	56
Figura 3-14. Captura pantalla configuració mòdul RI 0101	56
Figura 3-15. Captura pantalla configuració mòdul RI 0101	56
Figura 3-16. Captura pantalla configuració mòdul RI 0101	57
Figura 3-17. Captura pantalla configuració mòdul RI 0102	57
Figura 3-18. Captura pantalla configuració mòdul RI 0102	58
Figura 3-19. Captura pantalla configuració mòdul RI 0102	58
Figura 3-20. Captura pantalla configuració mòdul RI 0102	59
Figura 3-21. Captura pantalla configuració mòdul RI 0102	59
Figura 3-22. Captura pantalla configuració mòdul RI 0102	59
Figura 3-23. Captura pantalla configuració mòdul RI 0102	60
Figura 3-24. Captura pantalla configuració mòdul RI 0102	60
Figura 3-25. Captura pantalla configuració mòdul Coordinador	61

Figura 3-26. Captura pantalla configuració mòdul Coordinador	61
Figura 3-27. Captura pantalla configuració mòdul Coordinador	62
Figura 3-28. Captura pantalla configuració mòdul Coordinador	62
Figura 3-29. Captura pantalla configuració mòdul Coordinador	63
Figura 3-30. Captura pantalla configuració mòdul Coordinador	63
Figura 3-31. Esquema elèctric de la placa de circuit imprès	72
Figura 3-32. Disseny placa de circuit imprès.....	73
Figura 3-33. Esquema elèctric del mòdul coordinador	74
Figura 3-34. Captura pantalla estructura taula Calendari.....	75
Figura 3-35. Captura pantalla taula Calendari.....	75
Figura 3-36. Captura pantalla estructura taula Festius.....	75
Figura 3-37. Captura pantalla taula Festius.....	76
Figura 3-38. Captura pantalla estructura taula Lectures	76
Figura 3-39. Captura pantalla taula Lectures	76
Figura 3-40. Captura pantalla estructura taula Ordres	77
Figura 3-41. Captura pantalla taula Ordres	77
Figura 3-42. Captura pantalla estructura taula Torns	77
Figura 3-43. Captura pantalla taula Torns	77
Figura 3-44. Plànol tapa inferior comú mòduls	80
Figura 3-45. Plànol tapa superior mòdul sensor.....	81
Figura 3-46. Plànol tapa superior mòdul relé.....	82
Figura 3-47. Plànol tapa inferior mòdul coordinador.....	83
Figura 3-48. Plànol tapa superior mòdul coordinador.....	84
Figura 3-49. Material necessari tenyit peces	85
Figura 3-50. Dissolució colorant	86
Figura 3-51. Ebullició de les peces.....	86
Figura 3-52. Assecat de les peces	87

Índex equacions

1-1. Càlcul tensió de sortida	9
-------------------------------------	---

Annex 1 . Estudi i elecció dels diferents components de la xarxa

1.1 Elecció components de la placa

1.1.1 Esquema elèctric sensor temperatura

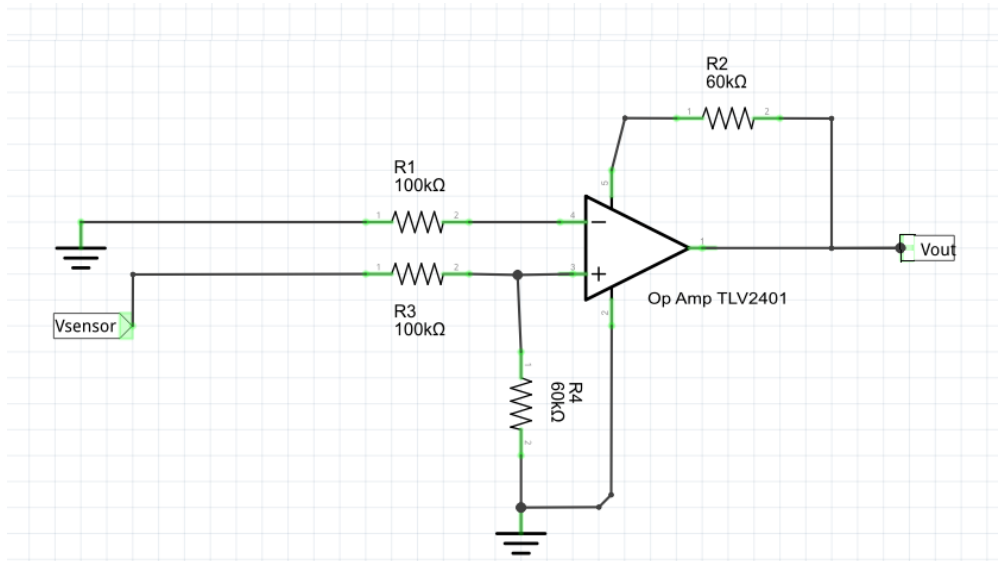


Figura 1-1. Esquema elèctric sensor temperatura

1.1.2 Esquema elèctric sensor llum

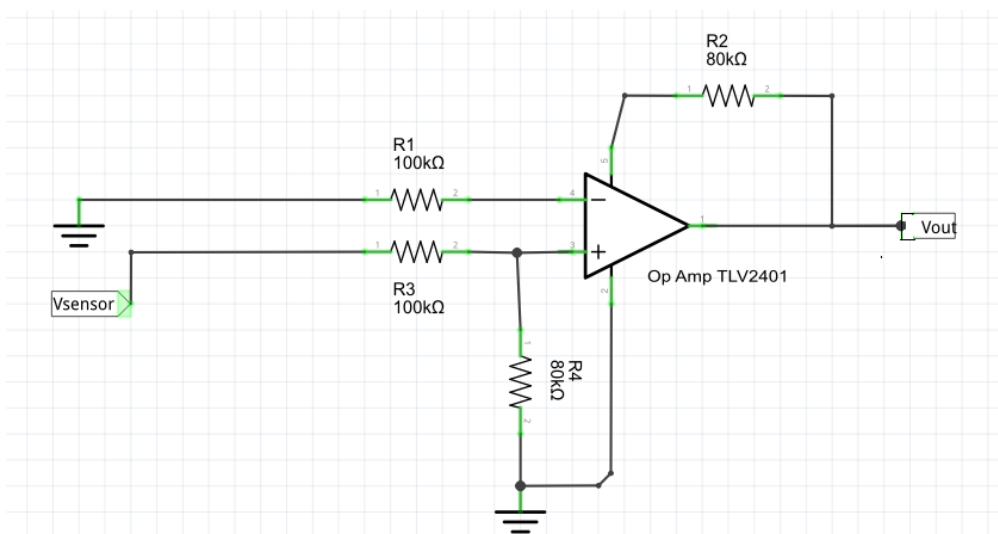


Figura 1-2. Esquema elèctric sensor llum

1.1.3 Esquema elèctric sensor bateria

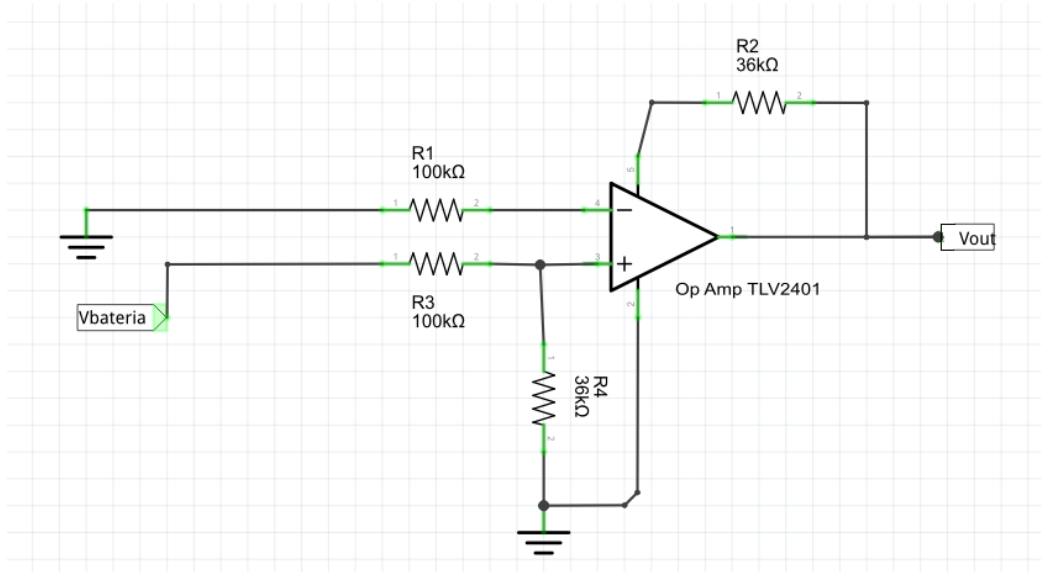
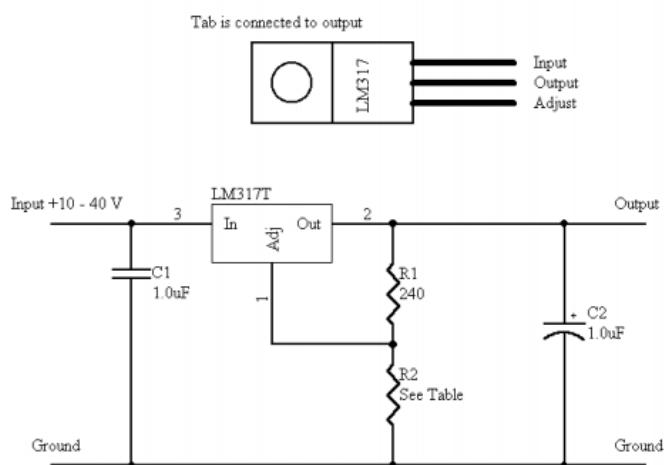


Figura 1-3. Esquema elèctric sensor bateria

1.1.4. Configuració regulador de tensió



R2, ohms	Voltage
680	4.79
750	5.16
820	5.52
910	5.99
1000	6.46
1100	6.98
1200	7.50
1300	8.02
1500	9.06
1600	9.58
1800	10.63
2000	11.67
2200	12.71

Figura 1-4. Relació Voltatge sortida - valors Resistències

$$V_o = 1,25 \left(\frac{R_2}{R_1} \right) I_{Ad} * R_2$$

1-1. Càlcul tensió de sortida

La I_{Ad} no es té en compte.

$$2.96 = 1,25 (330 / 240).$$

Per tant, les resistències són:

$$R_1 = 240 \Omega$$

$$R_2 = 330 \Omega$$

Annex 2 . Proves prèvies

2.1 Arduino

Una de les proves que s'ha fet és encendre o apagar un determinat nombre de leds en funció de la variació de la temperatura ambient respecte a una temperatura de referència prèviament definida.

A continuació es comenta el codi:

S'han definit dues constants per definir l'entrada del detector de temperatura i la temperatura de referència

```
const int sensorPin = A0;
const float TempLimit = 20.0;
```

Figura 2-1. Fragment codi Arduino

En la configuració s'ha utilitzat l'ordre *Serial.begin ()*. Aquest comandament inicialitza una connexió entre l'Arduino i l'ordinador; d'aquesta manera es poden veure els valors de l'entrada analògica a la pantalla del PC. L'argument 9600 és la velocitat a la qual es comunica l'Arduino, 9600 bits per segon. Quan s'obre aquest monitor, cal verificar que la ràtio de transmissió (*baud rate*) és de 9600. A més, s'han configurat els Pins 2, 3 i 4 com a sortides i apagats.

```
void setup() {
    Serial.begin(9600);
    for(int pinNumero = 2; pinNumero<5; pinNumero++){
        pinMode(pinNumero, OUTPUT);
        digitalWrite(pinNumero, LOW);
    }
}
```

Figura 2-2. Fragment codi Arduino

En el bucle, s'ha d'utilitzar una variable local anomenada *sensorVal* per emmagatzemar els valors llegits pel sensor. Per recollir les dades obtingudes del sensor, s'ha utilitzat la funció *analogRead ()*, que té com a argument el Pin d'entrada del sensor. Aquest valor, el qual està comprès entre el 0 i el 1023, és una representació del voltatge.

```
void loop() {
    int sensorVal = analogRead(sensorPin);
```

Figura 2-3. Fragment codi Arduino

La funció `Serial.print ()` envia la informació de l'Arduino al PC que està connectat. Es pot veure aquesta informació al monitor sèrie. Si a la funció `Serial.print ()` se li passa com a argument un text entre cometes, mostrarà aquest text com a sortida. Si per contra, se li passa com a argument una variable, mostrarà el valor contingut en aquesta variable.

```
Serial.print("Valor Sensor: ");
Serial.print(sensorVal);
```

Figura 2-4. Fragment codi Arduino

Per tal de poder saber la tensió real en què es troba el sensor, s'ha realitzat la següent operació. Com que la tensió està compresa entre 0 i 5 volts, es divideix `sensorVal` entre 1024.0 i es multiplica per 5.0. El nou número representa el voltatge del Pin.

```
float voltatge = (sensorVal/1024.0) * 5.0;
```

Figura 2-5. Fragment codi Arduino

S'ha examinat el full d'especificacions (*datasheet*) del sensor, i s'ha trobat informació sobre el rang de la tensió de sortida. El *datasheet* d'aquest sensor explica com per a cada variació de 10 mV en aquest sensor equival a un canvi de temperatura de 1 grau Celsius. També informa que el sensor pot llegir temperatures per sota dels 0 graus. Per aquesta raó, s'ha necessitat crear una compensació per a valors per sota de la congelació (0 graus).

Una vegada s'ha obtingut el voltatge se li ha restat 0.5 i s'ha multiplicat per 100, i el resultat és la temperatura precisa en graus Celsius. S'ha emmagatzemat aquest nou valor en la variable de tipus *float* anomenada *temperaturaActual*.

```
float temperaturaActual = (voltatge - .5) * 100;
Serial.println(temperaturaActual);
```

Figura 2-6. Fragment codi Arduino

El següent pas ha estat mostrar el valor real de la temperatura pel monitor sèrie i llavors, amb els bucles *if()...else*, s'activa un led quan la temperatura actual augmenta dos o més graus i menys de quatre respecte a la temperatura de referència; s'activen dos leds quan la temperatura actual augmenta quatre o més graus i menys de sis respecte a la temperatura de referència; s'activen tres leds quan la temperatura actual augmenta sis o més graus respecte a la temperatura de referència.

```
    if(temperaturaActual < TempLimit){
        digitalWrite(2, LOW);
        digitalWrite(3, LOW);
        digitalWrite(4, LOW);
    }
    else if(temperaturaActual >= TempLimit+2 &&
temperaturaActual < TempLimit+4){
        digitalWrite(2, HIGH);
        digitalWrite(3, LOW);
        digitalWrite(4, LOW);
    }
    else if(temperaturaActual >= TempLimit+4 &&
temperaturaActual < TempLimit+6){
        digitalWrite(2, HIGH);
        digitalWrite(3, HIGH);
        digitalWrite(4, LOW);
    }
    else if(temperaturaActual >= TempLimit+6){
        digitalWrite(2, HIGH);
        digitalWrite(3, HIGH);
        digitalWrite(4, HIGH);
    }
    delay(1);
}
```

Figura 2-7. Fragment codi Arduino

2.2 Xbee

2.2.1 Material necessari

Com ja s'ha explicat en la memòria, un dels problemes que ha calgut solucionar ha estat el fet que els mòduls Xbee no es puguin connectar directament al protoboard, ja que la separació entre les potes del mòdul Xbee és de 2mm i no de 2.54 mm, que és el pas estàndard del protoboard.

Per solucionar aquest contratemps ha calgut adquirir:

- Breakout Board.
- Connectors Xbee pas 2 mm.
- Tira de pins estàndard pas 2.54 mm.
- Soldador d'estany.
- Estany.

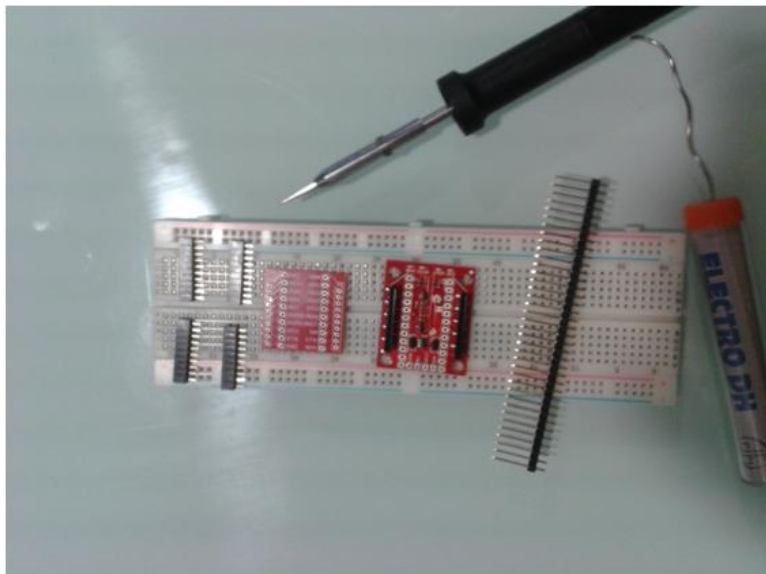


Figura 2-8. Material necessari per adaptar mòdul Xbee al protoboard

2.2.2 Reset

Un dels problemes que s'ha solucionat ha estat el de reconfigurar els mòduls Xbee als paràmetres de fàbrica.

Els passos que s'han seguit són:

- Tancar el missatge d'avís que informa que no reconeix el mòdul.
- Anar a *Add devices*.
- Canviar el *baud rate of bootloader* a 38400.
- Clicar a *Finish*.
- Clicar a *Recovery*.
- Seleccionar *XB24-ZB Zigbee Coordinator API 21A7*.

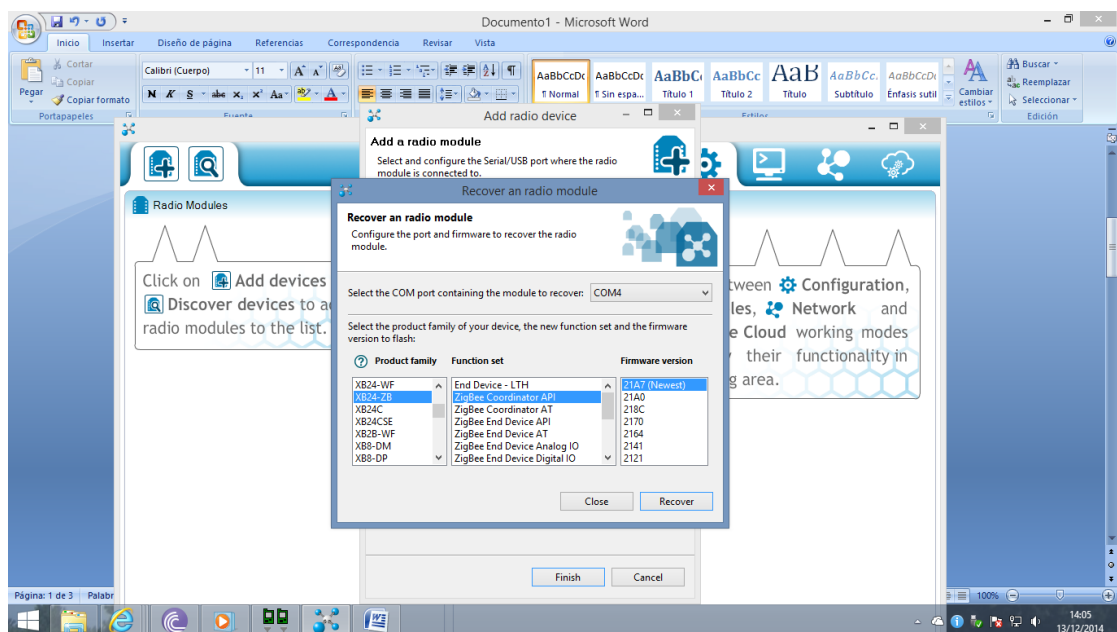


Figura 2-9. Captura pantalla del procediment a seguir per fer un Reset

- Clicar a *Recover*.

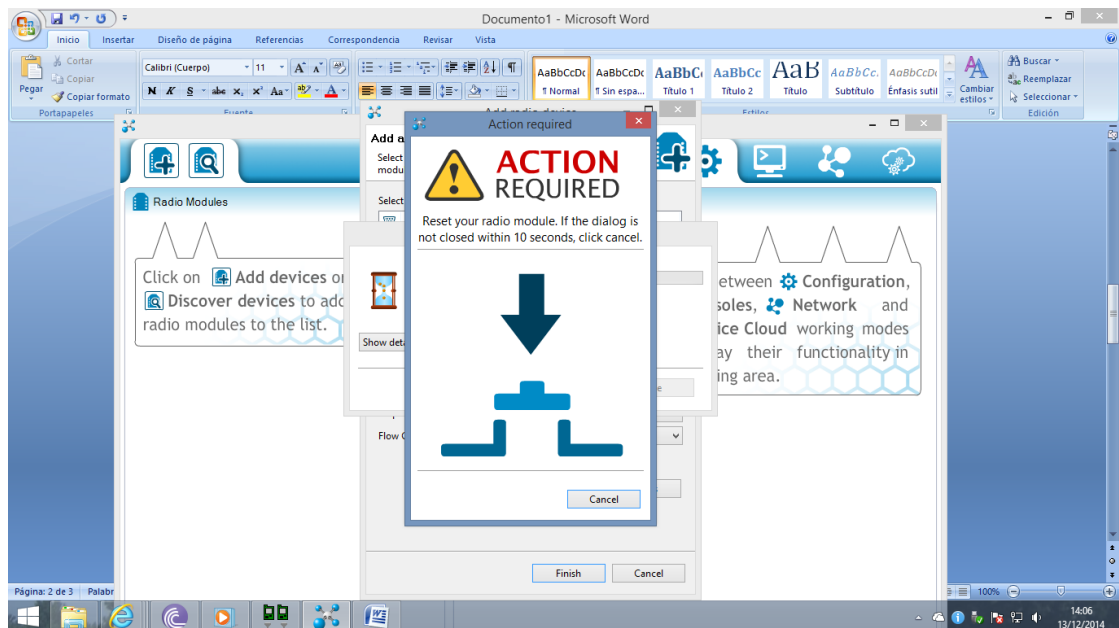


Figura 2-10. Captura pantalla del procediment a seguir per fer un Reset

- Fer un Reset amb el mòdul connectat tal com indica la imatge (fent un pont entre pin 5 i 10) durant 30 segons com a mínim.

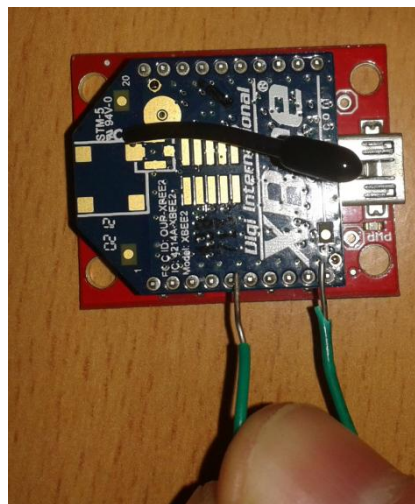


Figura 2-11. Realització Reset manualment

- Automàticament es reinicialitzarà el mòdul.

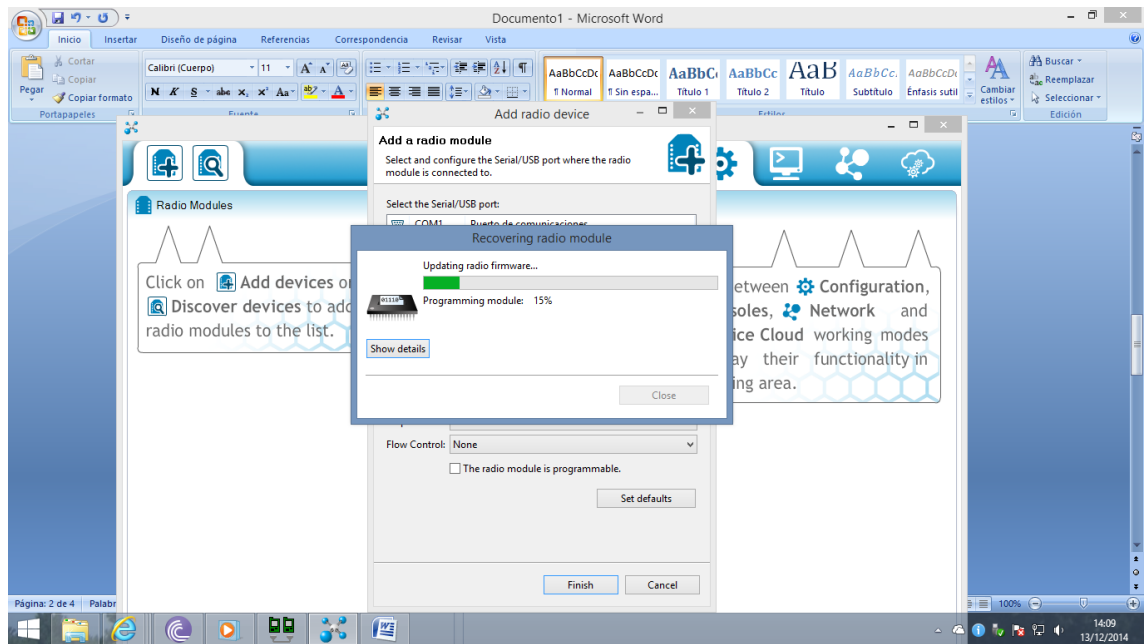


Figura 2-12. Captura pantalla del procediment a seguir per fer un Reset

- Anar a *Add devices*.
- Canviar el *baud rate of bootloader* a 9600.
- Clicar a *Finish*.
- Apareix la configuració de fàbrica, i ja es pot tornar a treballar amb el mòdul.

2.3 Arduino + Xbee

2.3.1 Comunicació punt a punt

Configuració del port

- Executar el programa X-CTU .

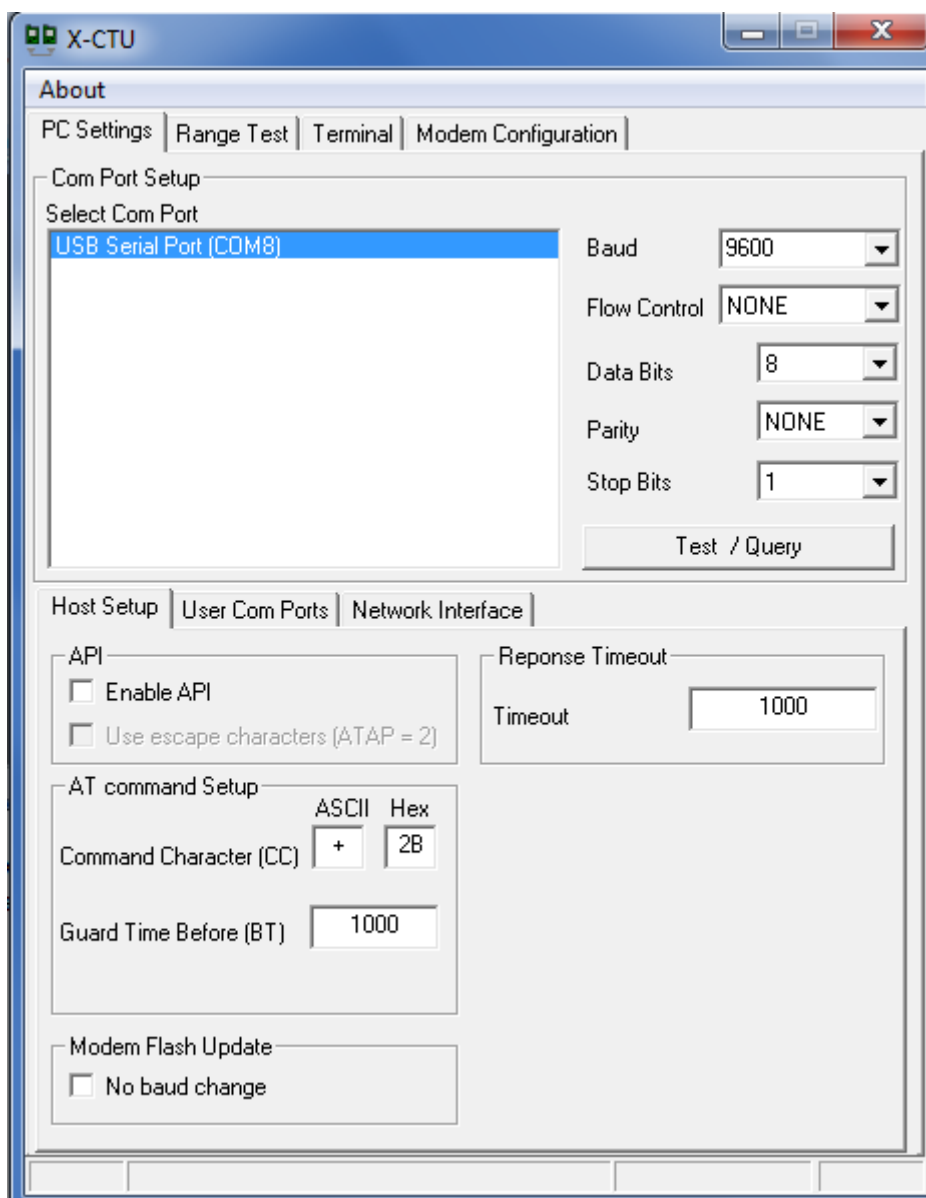


Figura 2-13. Captura pantalla configuració mòdul Xbee Punt a Punt

- Seleccionar el port, en el nostre cas el COM8, i el *Test/Query*, que ha de donar com a resposta *Communication with modem. OK*.

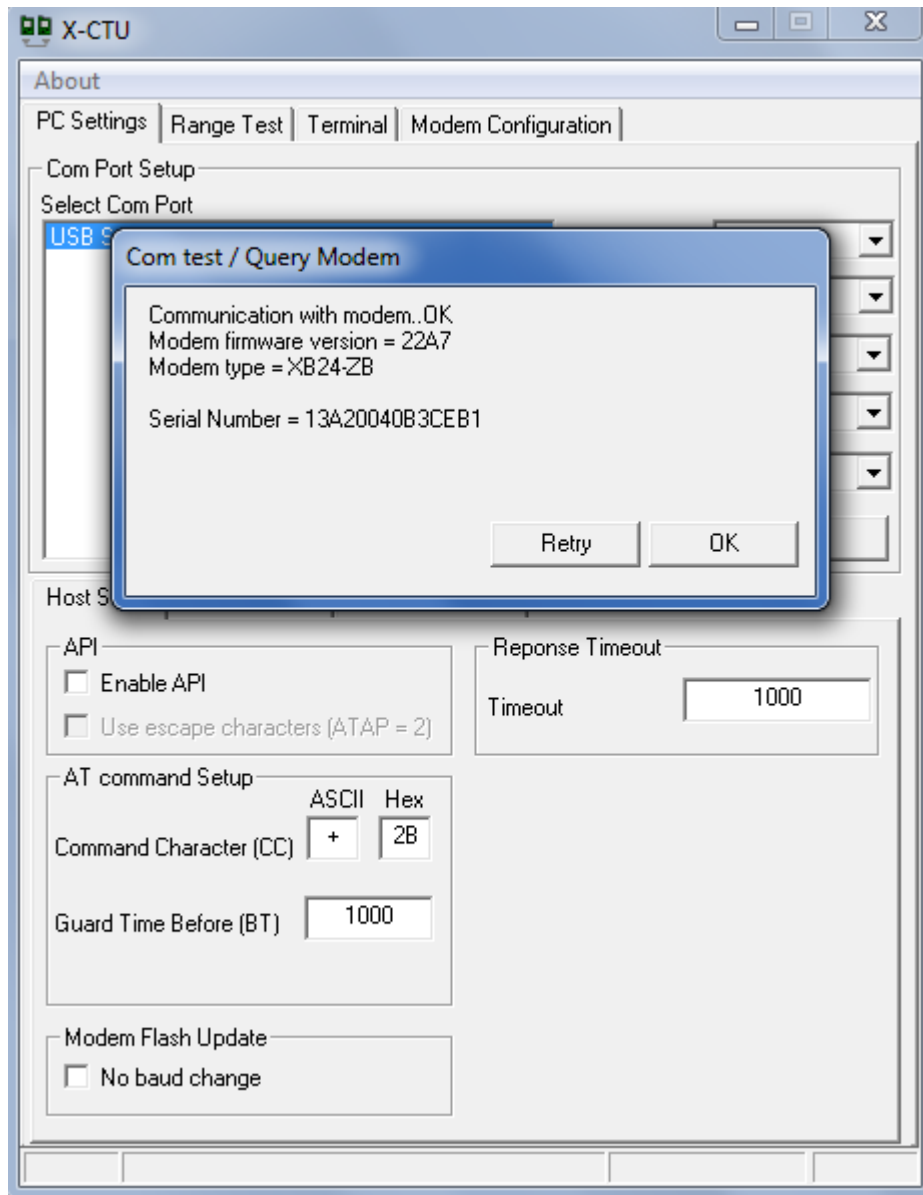


Figura 2-14. Captura pantalla configuració mòdul Xbee Punt a Punt

Configuració del coordinador

- Seleccionar la pestanya *Modem Configuration*.
- Seleccionar l'opció *Read*.

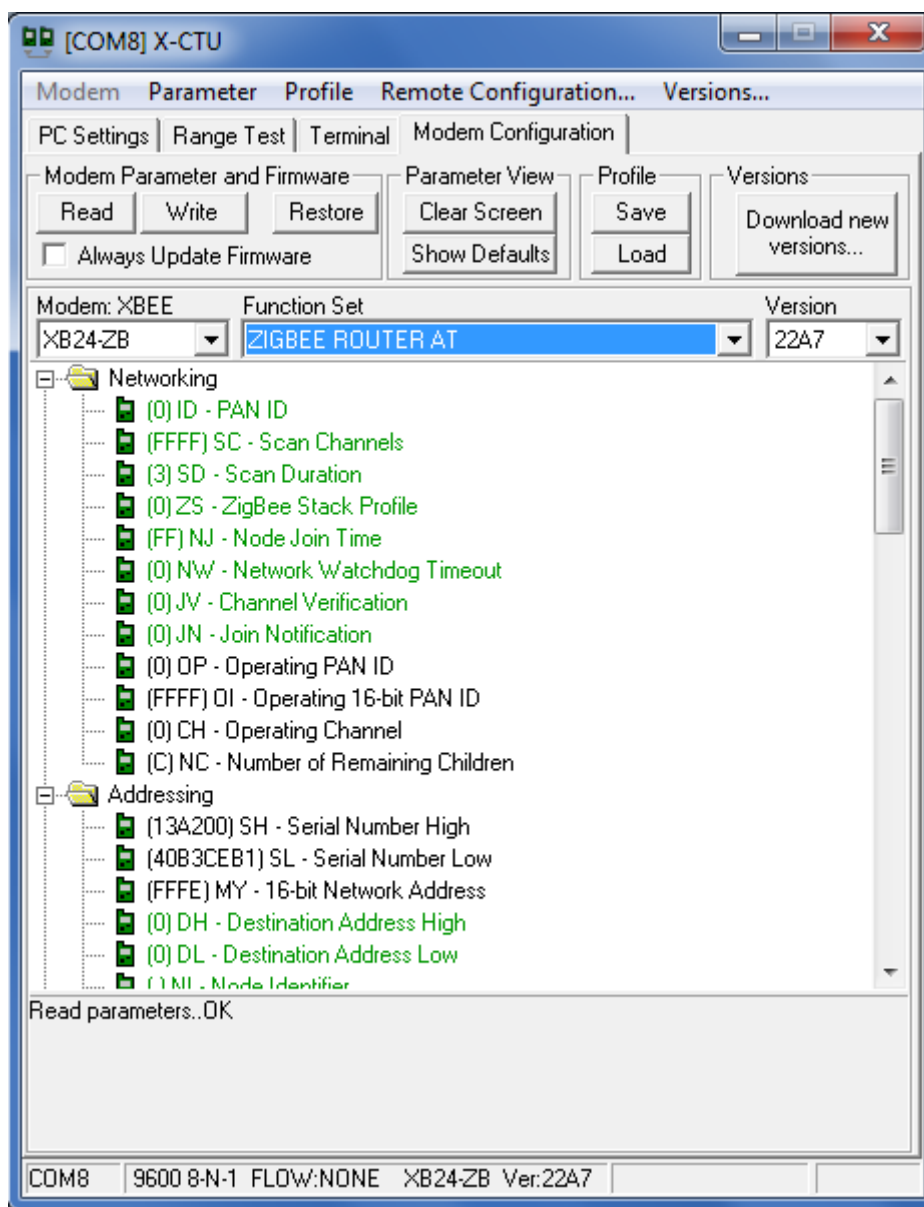


Figura 2-15. Captura pantalla configuració mòdul Xbee Punt a Punt

- En l'opció *Function Set*, seleccionar *ZIGBEE COORDINADOR AT*.
- En la carpeta *Networking* seleccionar *ID-PAN ID* i establir un número entre el rang de valors permesos. El *PAN ID* és un número d'identificació

que ha de ser el mateix per tots els Xbee de la xarxa; a la nostra xarxa serà el 17.

- En la carpeta *Addressing*, en l'opció *DH-Destination Address High*, establir el valor SH de l'altre Xbee, que en el nostre cas és 0013A200.
- A l'opció *DL-Destination Address Low*, establir el valor SL de l'altre Xbee, que en el nostre cas és 40C04C95.

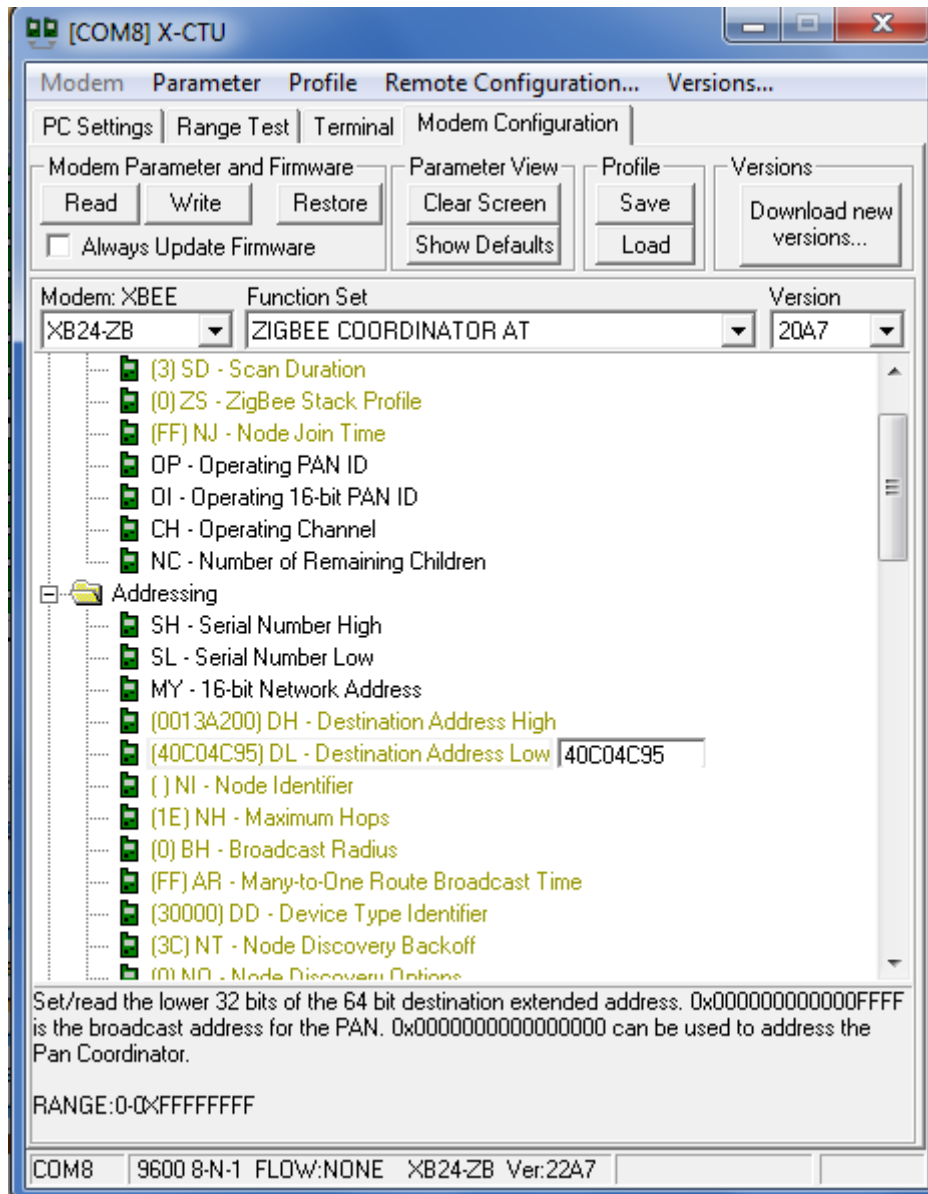


Figura 2-16. Captura pantalla configuració mòdul Xbee Punt a Punt

- Seleccionar el botó *Write* i esperar que acabi la configuració.

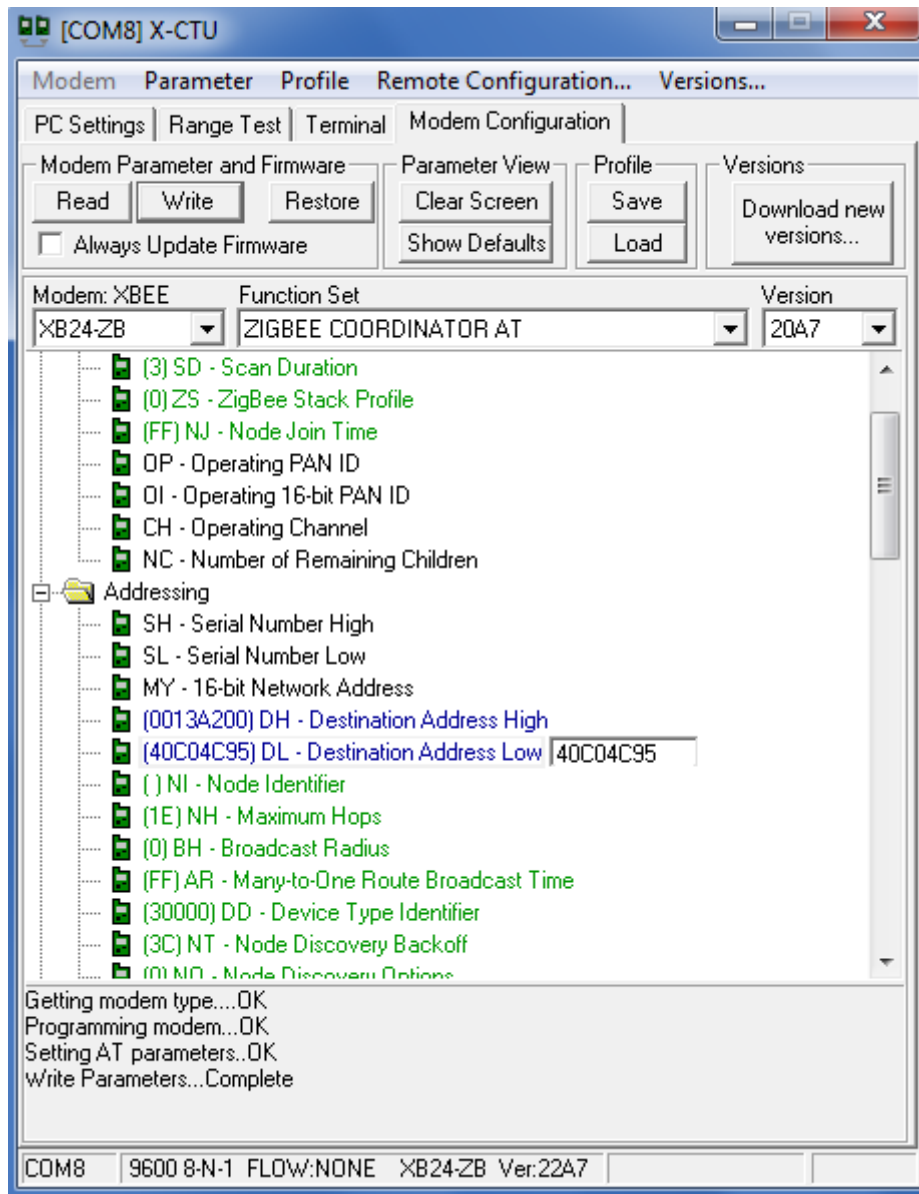


Figura 2-17. Captura pantalla configuració mòdul Xbee Punt a Punt

- Tancar el programa X-CTU i desconnectar l'explorador i el mòdul Xbee.

Configuració End Device

- Repetir els primers passos, fins el que seleccionàvem l'opció *Read*.
- En l'opció *Function Set*, hi ha dues possibilitats en funció de si es vol que sigui Router o Terminal; en el primer cas, *ZIGBEE ROUTER AT*; en el segon, *ZIGBEE END DEVICE AT*. Cal escollir la segona opció, és a dir, que treballi com a End Device.
- En la carpeta *Networking*, seleccionar *ID-PAN ID* i col·locar el mateix valor escollit com a ID de la xarxa en el coordinador; en el nostre cas, 17.
- En la carpeta *Addressing*, en l'opció *DH-Destination Address High*, establir el valor SH de l'altre Xbee (coordinador), que en el nostre cas és 0013A200.
- En l'opció *DL-Destination Address Low*, establir el valor SL de l'altre Xbee (coordinador), que en el nostre cas és 40B3CEB1.
- Seleccionar el botó *Write* i esperar que acabi la configuració.

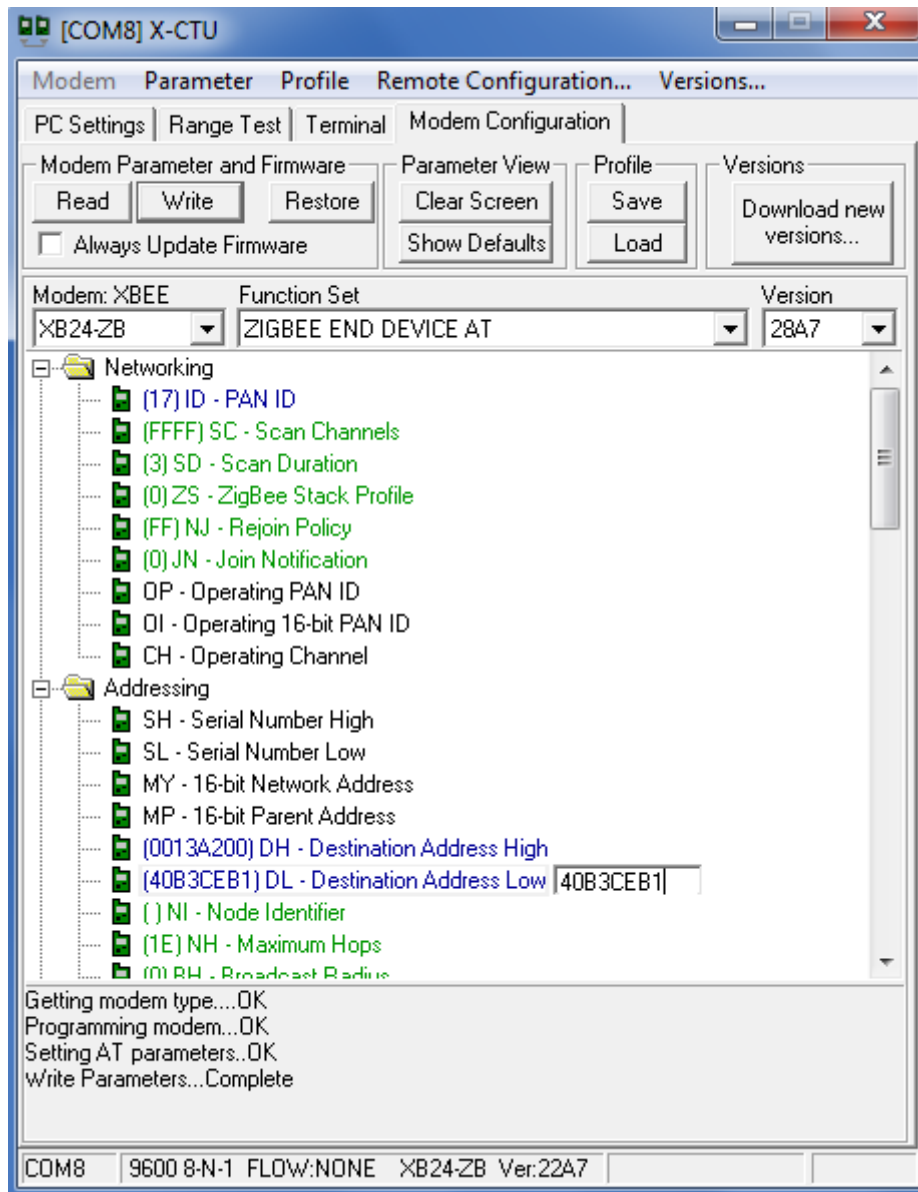


Figura 2-18. Captura pantalla configuració mòdul Xbee Punt a Punt

- Tancar el programa X-CTU i desconnectar l'explorador i el mòdul Xbee.

Configuració mode *Sleep*:

Per poder configurar el mòdul on hi ha el sensor de temperatura en mode *Sleep*, el que es vol aconseguir per tal de poder augmentar al màxim l'autonomia del node, és que només faci una lectura cada 10 minuts. Per assolir aquest objectiu s'han seguit els següents passos:

- En la carpeta *Sleep Modes*, seleccionar l'opció *SM* i col·locar-hi el valor 4; això activa el Cyclic Sleep.
- En la carpeta *Sleep Modes*, seleccionar l'opció *ST*, en què s'especifica el temps d'inactivitat abans d'entrar en el mode *Sleep*. El rang de valors acceptats (hexadecimal) és de $0X1 - 0XFFFE \times 1\text{mS}$, i cal col·locar el valor 190.
- En la carpeta *Sleep Modes*, seleccionar l'opció *SP*, en què s'especifica el temps que el mòdul romandrà dormint. El rang de valors acceptats (hexadecimal) és de $0X20 - 0XAF0 \times 10\text{ mS}$. Com es pot observar, el valor màxim permès és en decimal $2800 \times 10\text{ mS} = 28000$, és a dir, 28 segons. Cal col·locar el valor màxim $0X3E8$, que en decimal és 1000, $1000 \times 10\text{ mS} = 10000\text{ mS}$.
- En la carpeta *Sleep Modes*, seleccionar l'opció *SN*, en què s'especifiquen els cicles en què el mòdul ha d'estar adormit. Aquesta opció serveix per ampliar el temps màxim que el mòdul pot estar en mode *Sleep*. El rang de valors acceptats (hexadecimal) és de $0X1 - 0XFFFF$. Cal col·locar $0X3C$, que en decimal és 60.
- En la carpeta *Sleep Modes*, seleccionar l'opció *SO*. Cal col·locar el valor $0X2$, el mòdul estarà en mode actiu durant el temps especificat en *ST*, i escollir l'opció $0X04$, que permet aconseguir un temps major de *Sleep*, ja que en aquest cas el temps total serà $SP \times SN$, és a dir, $10000 \times 60 = 60000\text{ mS}$, que són 600 segons (10 minuts).

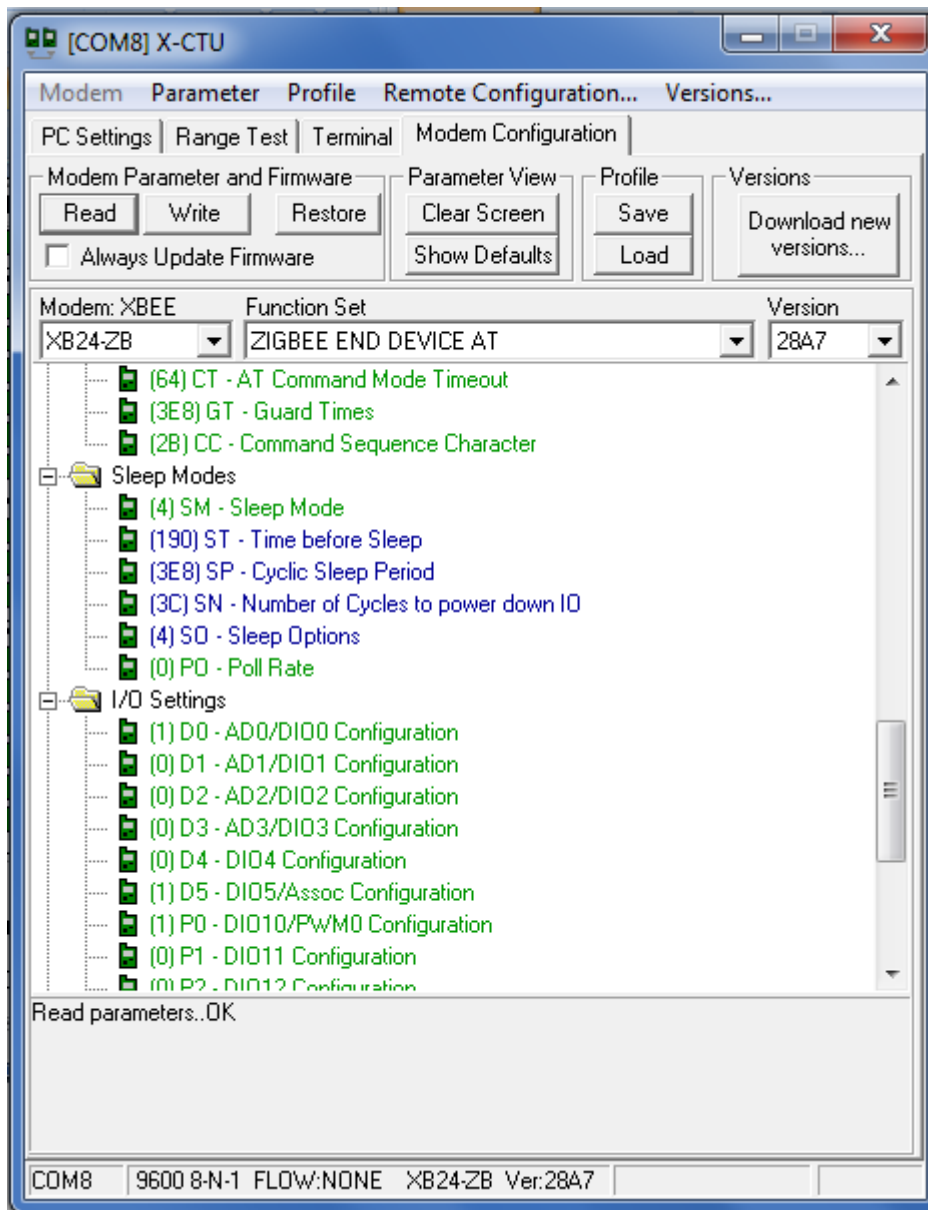
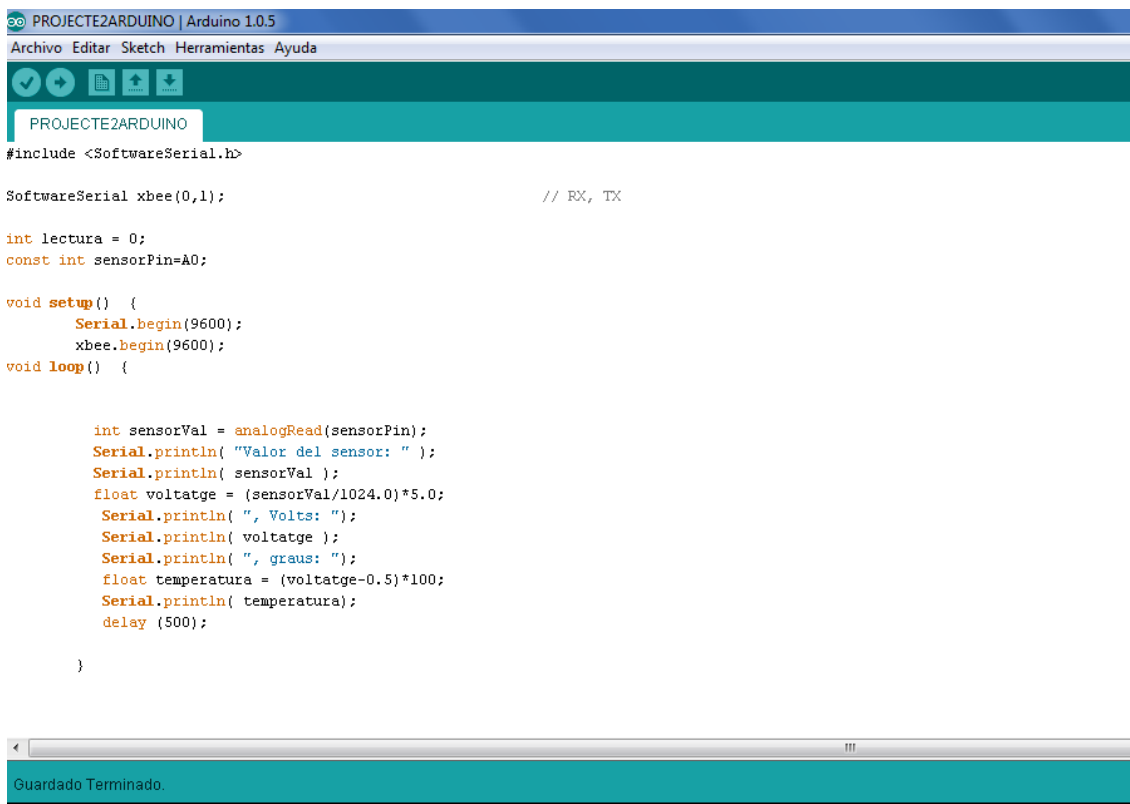


Figura 2-19. Captura pantalla configuració mòdul Xbee Punt a Punt

Codi Arduino



```
PROJECTE2ARDUINO | Arduino 1.0.5
Archivo Editar Sketch Herramientas Ayuda
PROJECTE2ARDUINO
#include <SoftwareSerial.h>

SoftwareSerial xbee(0,1); // RX, TX

int lectura = 0;
const int sensorPin=A0;

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

void loop() {

  int sensorVal = analogRead(sensorPin);
  Serial.println( "Valor del sensor: " );
  Serial.println( sensorVal );
  float voltatge = (sensorVal/1024.0)*5.0;
  Serial.println( " , Volts: " );
  Serial.println( voltatge );
  Serial.println( " , graus: " );
  float temperatura = (voltatge-0.5)*100;
  Serial.println( temperatura );
  delay (500);

}
```

Guardado Terminado.

Figura 2-20. Captura pantalla codi Arduino prova Punt a Punt

Resultat final

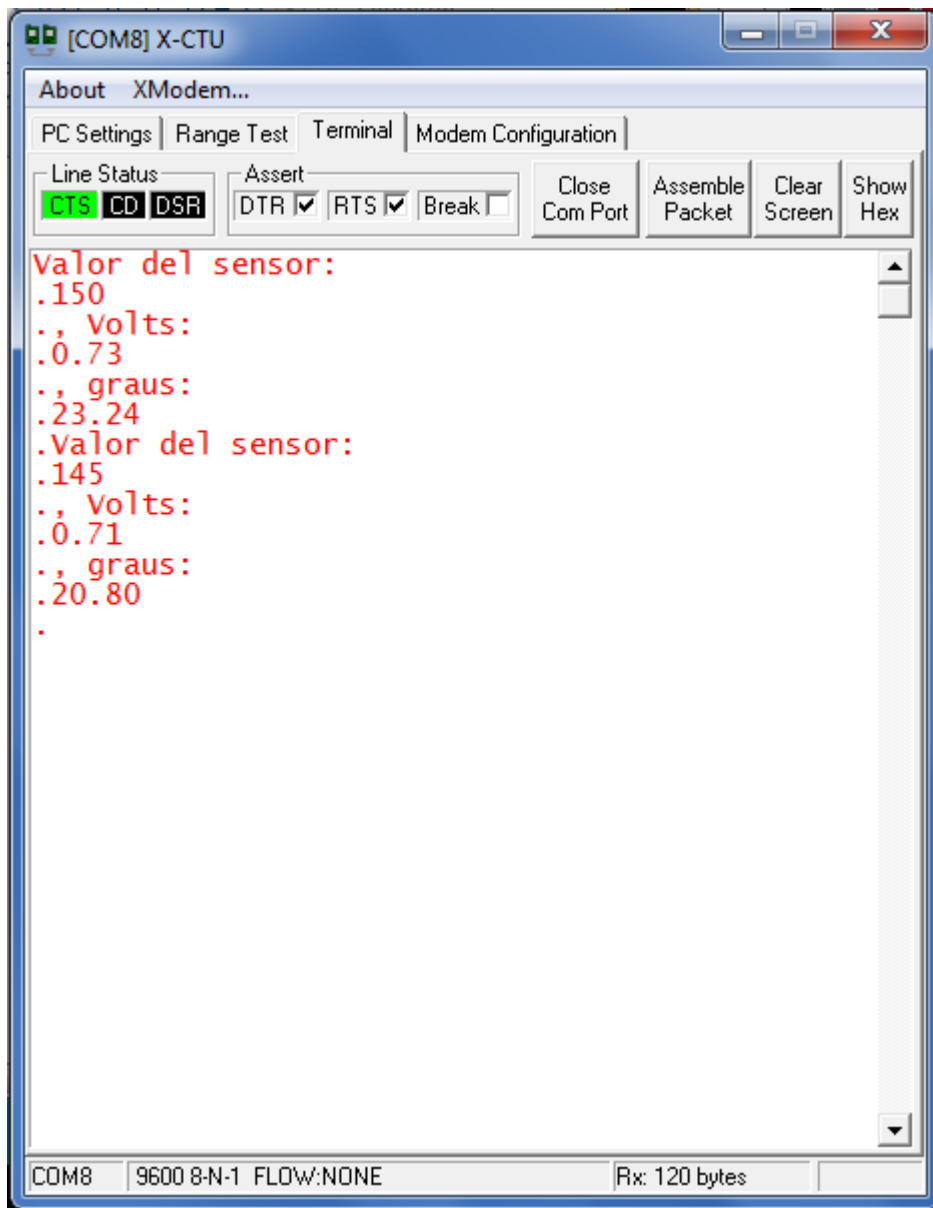


Figura 2-21. Captura de pantalla resultat final prova Punt a Punt

2.3.2 Estudi trama API

En aquest exercici s'ha realitzat l'estudi complet d'una trama API.

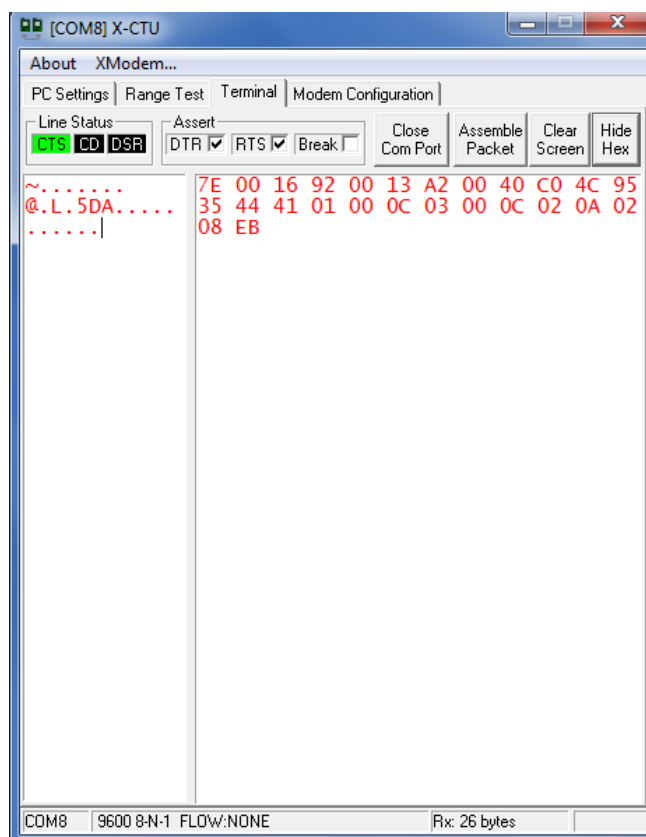


Figura 2-22. Captura pantalla Trama API a analitzar

Com es pot observar, la trama està en hexadecimal.

Tot seguit es passa a comentar la trama rebuda:

7E 00 16 92 00 13 A2 00 40 C0 4C 95 35 44 41 01 00 0C 03 00 0C 02 0A 02 08 EB

7E és el delimitador de les trames API, qualsevol trama API ha de començar per 7E.

00 16 són els dos bytes que representen la longitud de la trama, això vol dir que a partir d'aquí el número de dades és 0X16, és a dir, 22 en decimal, sense comptar l'últim byte, que és l'algoritme de verificació que la informació ha arribat completa (*Checksum*).

92 és el tipus de trama API; en aquest cas és la recepció de mostres de I/O ZIGBEE.

00 13 A2 00 40 C0 4C 95 són la direcció del mòdul font, és a dir, *End Device*, que envia la informació *SH* i *SL*, respectivament.

35 44 és la direcció curta de 16 bits del mòdul font assignada pel coordinador.

41 és el **ACK** o *acknowledgement* que ha de tornar el receptor per indicar que ha arribat la trama; ha de ser un número diferent a zero.

01 és el número de mostres sol·licitades; en aquest cas, una sola mostra de totes les I/O configurades.

00 0C són les entrades / sortides configurades. Recordem que les digitals D2 i D3 les tenim com a sortida digital alta i entrada digital respectivament; llavors, 00 0C ha de ser conseqüent amb això:

0000 0000 0000 1100 = 00 0C. Podem observar com correspon amb D2 i D3.

03 són entrades analògiques configurades. Recordem que les analògiques eren D0 i D1, llavors 03 ha de ser conseqüent amb això:

0000 0011 = 03. Podem observar que correspon amb D0 i D1.

00 0C representa l'estat de les I/O digitals configurades, és a dir, D2 i D3, en aquest cas:

0000 0000 0000 1100 = 00 0C. Podem observar que la sortida D2 dona un nivell alt i que l'entrada D3 està rebent un nivell alt.

02 0A és el valor de l'entrada analògica D0, si ho passem a decimal és 522. Partint que el voltatge de referència en el mòdul Xbee Serie2 és intern i té un valor de 1.2 volts i que el mòdul ADC és de 10 bits (1023), es pot dir que el voltatge d'entrada és de:

Valor senyal analògic = Resolució x valor digital llegit

$(1,2 / 1023) \times 522 = 0.61 \text{ V.}$

02 08 és el valor de l'entrada analògica D1, si ho passem a decimal és 520. Partint del fet que el voltatge de referència en el mòdul Xbee Serie2 és intern i té un valor de 1.2 Volts i que el mòdul ADC és de 10 bits (1023), es pot dir que el voltatge d'entrada és de:

Valor senyal analògic = Resolució x valor digital llegit

$(1,2 / 1023) \times 520 = 0.60 \text{ V.}$

EB és el *Checksum*. L'algoritme utilitzat pel protocol ZIGBEE simplement suma tots els bytes de la trama partint des del tipus de trama i llavors el byte menys significatiu del resultat el resta de FF. Comprovem el *Checksum*:

$$92 + 00 + 13 + A2 + 00 + 40 + C0 + 4C + 95 + 35 + 44 + 41 + 01 + 00 + 0C + 03 + 00 + 0C + 02 + 0A + 02 + 08 = 414.$$

Agafem 14 i llavors: *Checksum* = FF – 14 = **EB**

2.3.3 Introducció xarxa mode API

Per introduir-se en aquest camp, s'ha realitzat una petita pràctica en la que hi haurà un coordinador API i un End Device, on aquest tindrà un temps de mostreig establert i enviarà l'estat de les seves entrades i sortides al coordinador.

Els passos que s'han seguit són els següents:

Configuració Coordinador

- Executar el programa X-CTU i configurar un mòdul Xbee com a Coordinador API.

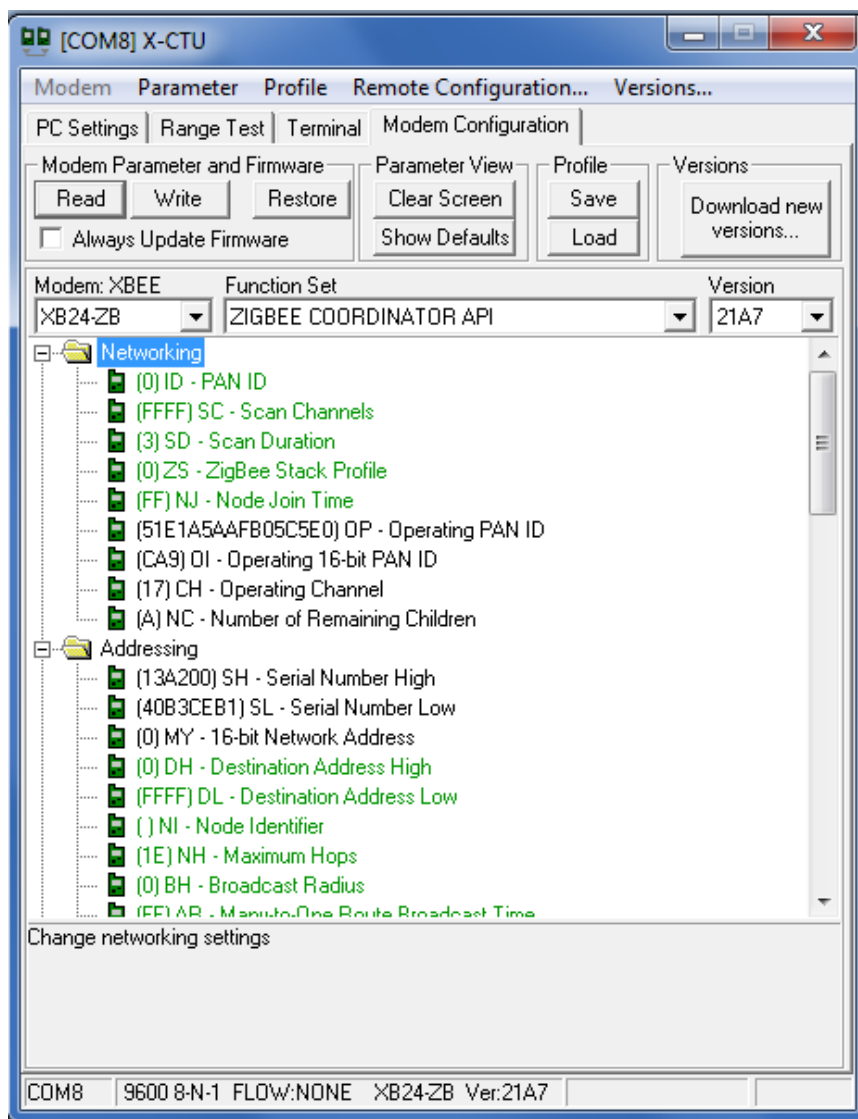


Figura 2-23. Captura pantalla configuració mòdul Xbee xarxa API

- El Coordinador ha escollit les següents configuracions automàticament:
 - *Operating PAN ID (OP)*: 51E1A5AAFB05C5E0.
 - *Operating 16-bit PAN ID (OI)*: CA9.
 - *Operating Chanel (CH)*: 17, ha escollit aquest canal com el més net.
 - A més, dins de la carpeta *Addressing* en els apartats *SH* i *SL* hi ha el número de sèrie que identifica el mòdul com a únic a nivell mundial, és a dir, no existeix cap altre mòdul amb aquesta identificació.

Configuració End Device

Per poder configurar l'altre mòdul correctament cal tenir en compte que el coordinador estigui treballant.

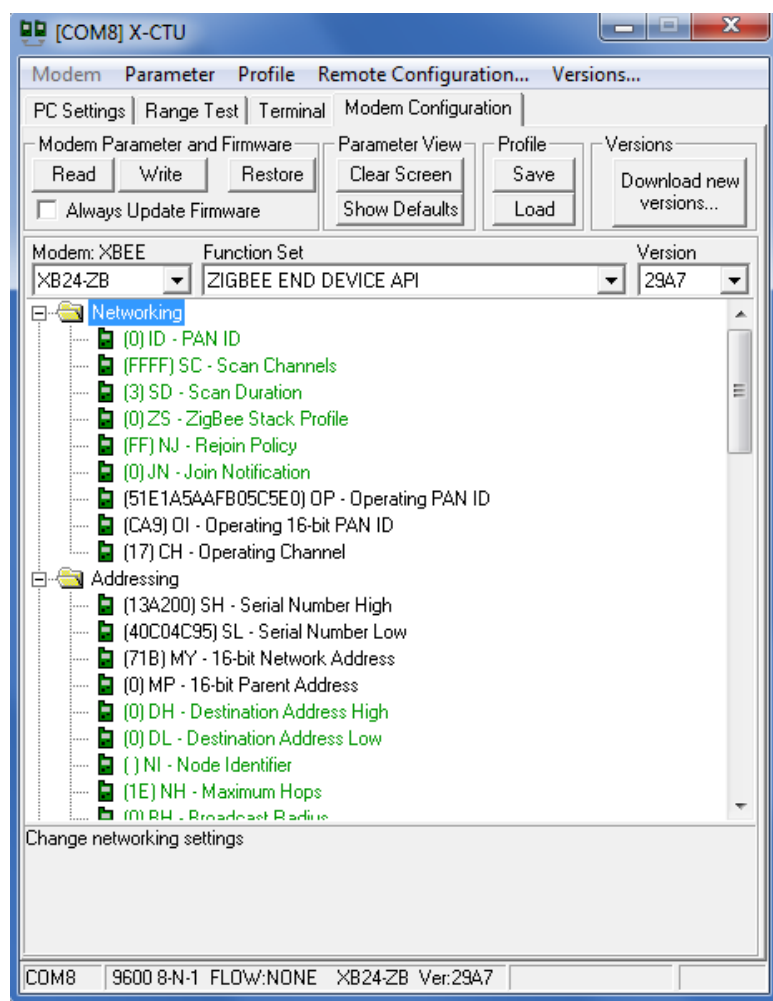


Figura 2-24. Captura pantalla configuració mòdul Xbee xarxa API

- El primer que es pot observar és que automàticament s'ha associat a les mateixes *OP*, *OI* i *CH* que havia definit el coordinador i que ha proporcionat una adreça de 16 bits a l'apartat *MY*:
 - *Operating PAN ID (OP)*: 51E1A5AAFB05C5E0.
 - *Operating 16-bit PAN ID (OI)*: CA9.
 - *Operating Chanel (CH)*: 17.
 - *16-bit Network Address (MY)*: 71B.
- Dins la carpeta *Addressing* les *DH* i *DL* vénen en blanc de fàbrica; en aquests dos punts cal posar la *SH* i *SL* del coordinador.

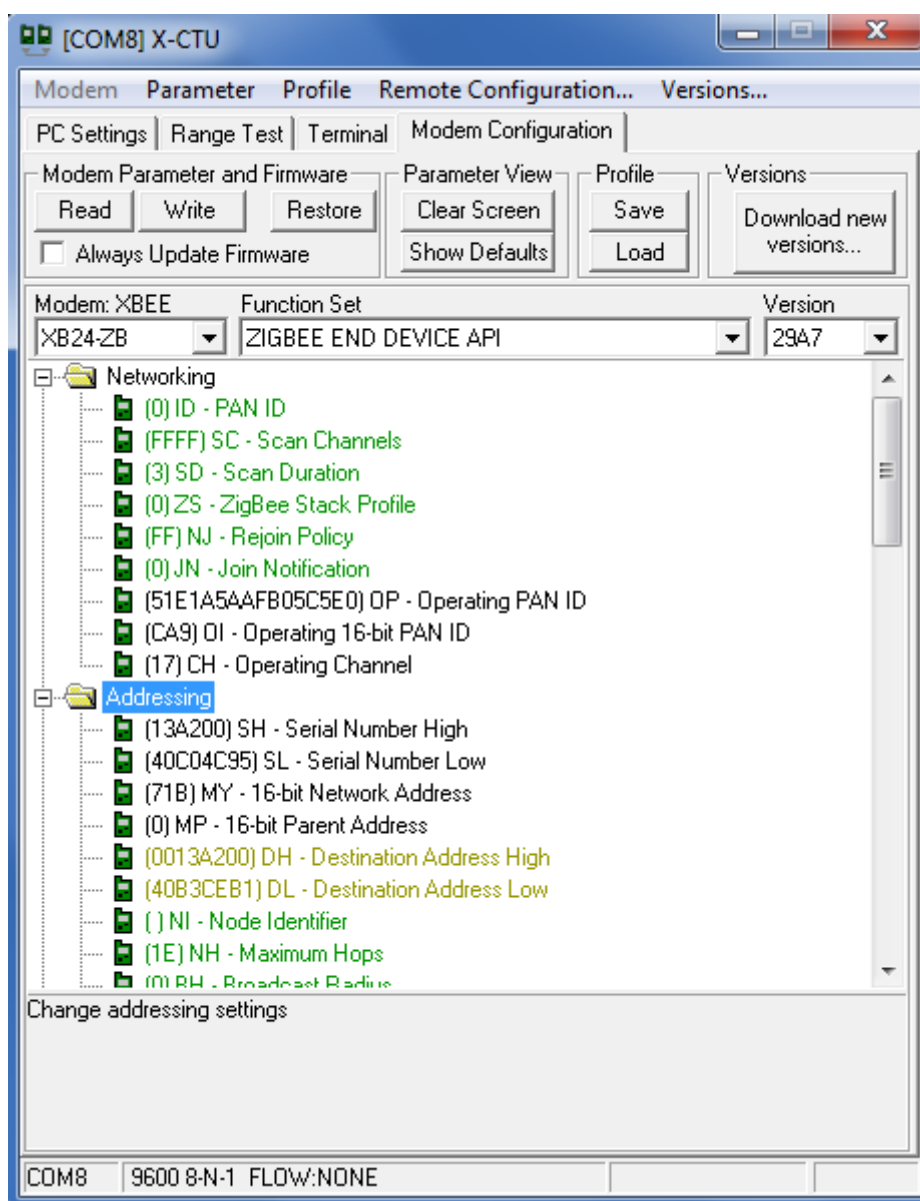


Figura 2-25. Captura pantalla configuració mòdul Xbee xarxa API

- Definir les entrades / sortides i el temps de mostreig de la següent manera:
 - D0 ubicada en el pin 20: entrada analògica.
 - D1 ubicada en el pin 19: entrada analògica.
 - D2 ubicada en el pin 18: sortida digital en alt.
 - D3 ubicada en el pin 17: entrada digital.
 - Temps de mostreig de 15 segons (0X3A98).

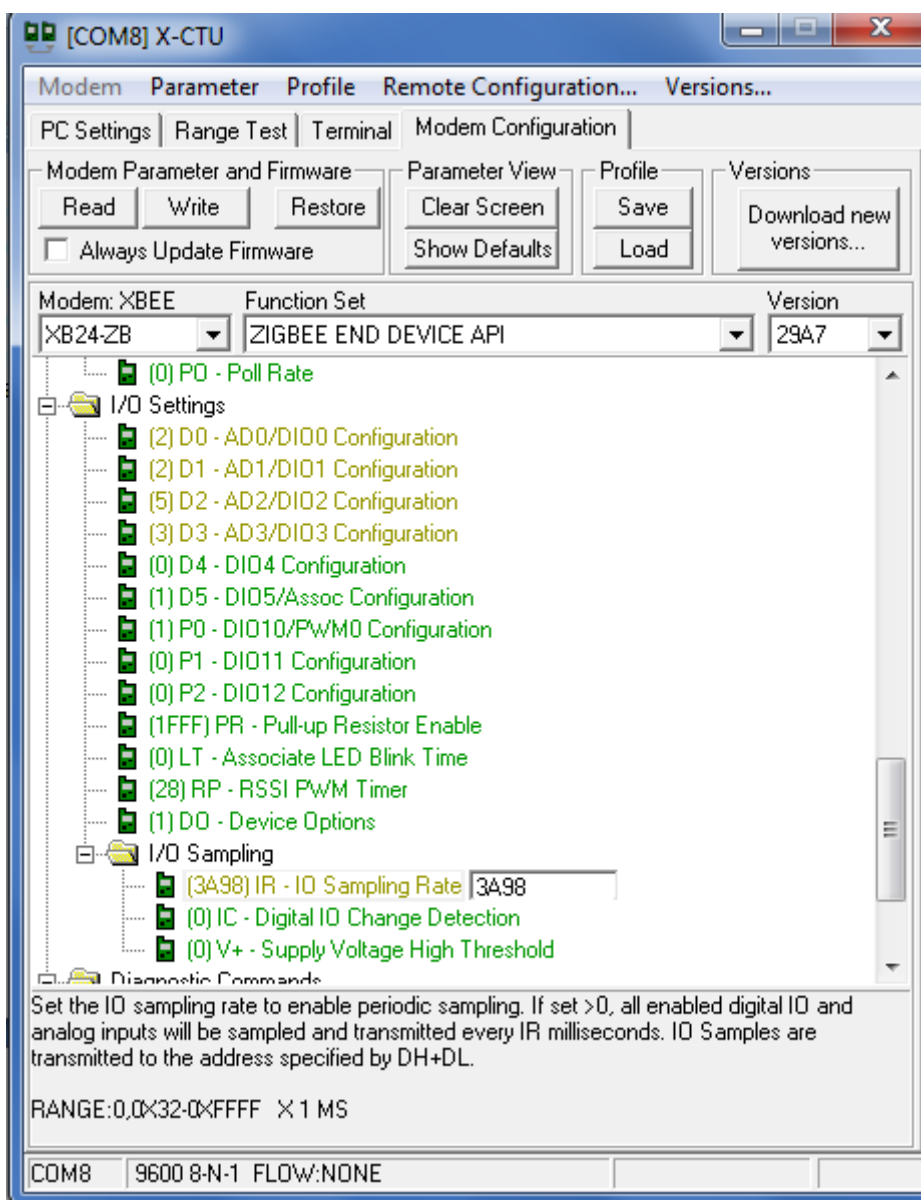


Figura 2-26. Captura pantalla configuració mòdul Xbee xarxa API

- A partir d'aquest moment cada 15 segons es rebran trames API amb l'estat de les I/O configurades, i es poden observar en la finestra terminal del X-CTU.

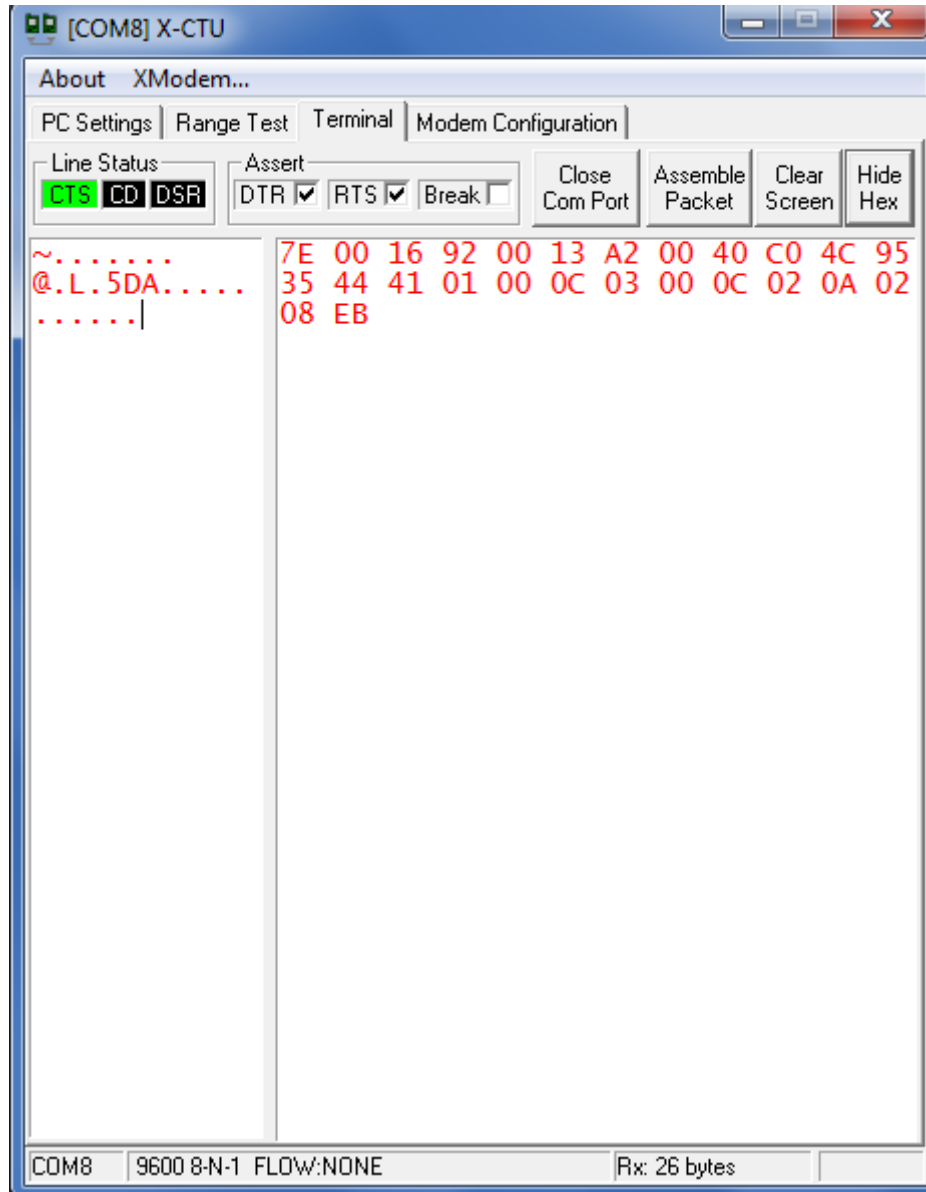


Figura 2-27. Captura pantalla configuració mòdul Xbee xarxa API

2.3.4 Lectura potenciòmetre mitjançant trames API

Explicació prova

En aquest assaig s'ha utilitzat un Arduino amb el coordinador Xbee connectats entre ells amb una Shield, on arribarà la lectura del mòdul Xbee que té el potenciòmetre.

El microcontrolador està programat per tal de poder saber quin voltatge hi ha a la sortida del potenciòmetre.

Per poder fer això el primer que cal saber és la llargada de la trama API on s'envien les dades i on hi ha la lectura del pin D0. Per poder saber això cal saber que la longitud d'aquesta trama és de 23 bytes, i que la lectura de l'entrada analògica està en els bytes 18 i 19 (comptant sempre començant per 0).

Per poder transformar la lectura de l'entrada D0 a volts cal saber i aplicar la següent informació:

Rang lectura pin analògics: 0-1.2 V.

10 bits = $2^{10} = 1023$

S'aplica la següent fórmula:

Voltatge potenciòmetre = $(1.2 / 1023) * \text{lectura D0}$;

A més, també cal tenir en compte que dins de la trama API la lectura de D0 està en dos bytes diferents i consecutius; per tant, primer cal desxifrar aquest valor. El primer byte és la part alta de la lectura i el segon byte n'és la part baixa:

Lectura D0 = $\text{bytealt} * 256 + \text{bytebaix}$

Per últim, s'utilitzarà el port sèrie per visualitzar els resultats.

A més, per interactuar més amb el microcontrolador s'ha decidit que en funció del valor de la lectura de D0 el led connectat a la sortida 6 del microcontrolador faci una seqüència o una altra.

Muntatge circuit

Cal col·locar un potenciòmetre que simularà les lectures d'un sensor; aquest potenciòmetre va acompanyat d'un condensador de 100microF per evitar "interferències".

El potenciòmetre anirà connectat a l'entrada D0 del mòdul Xbee.

A més, s'han de col·locar els tres leds indicadors tal com s'explica a l'apartat Revelacions i Indicacions de la memòria.

En la placa de proves de l' Arduino cal connectar un led a la sortida 6 de l'Arduino.

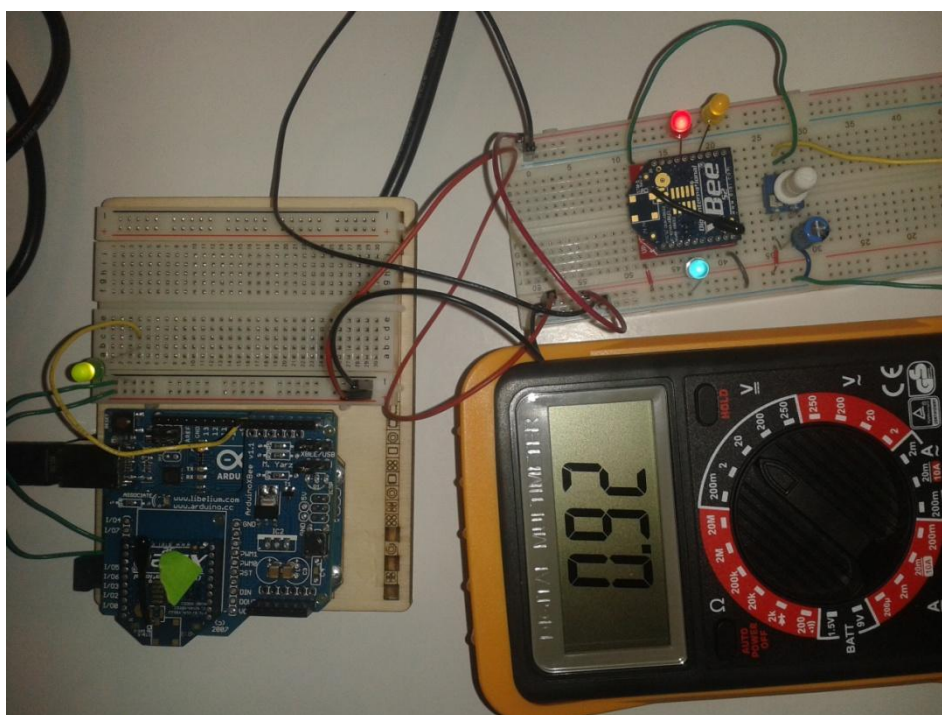


Figura 2-28. Muntatge circuit assaig lectura potenciòmetre

Configuració Coordinador:

- Activar la configuració API:

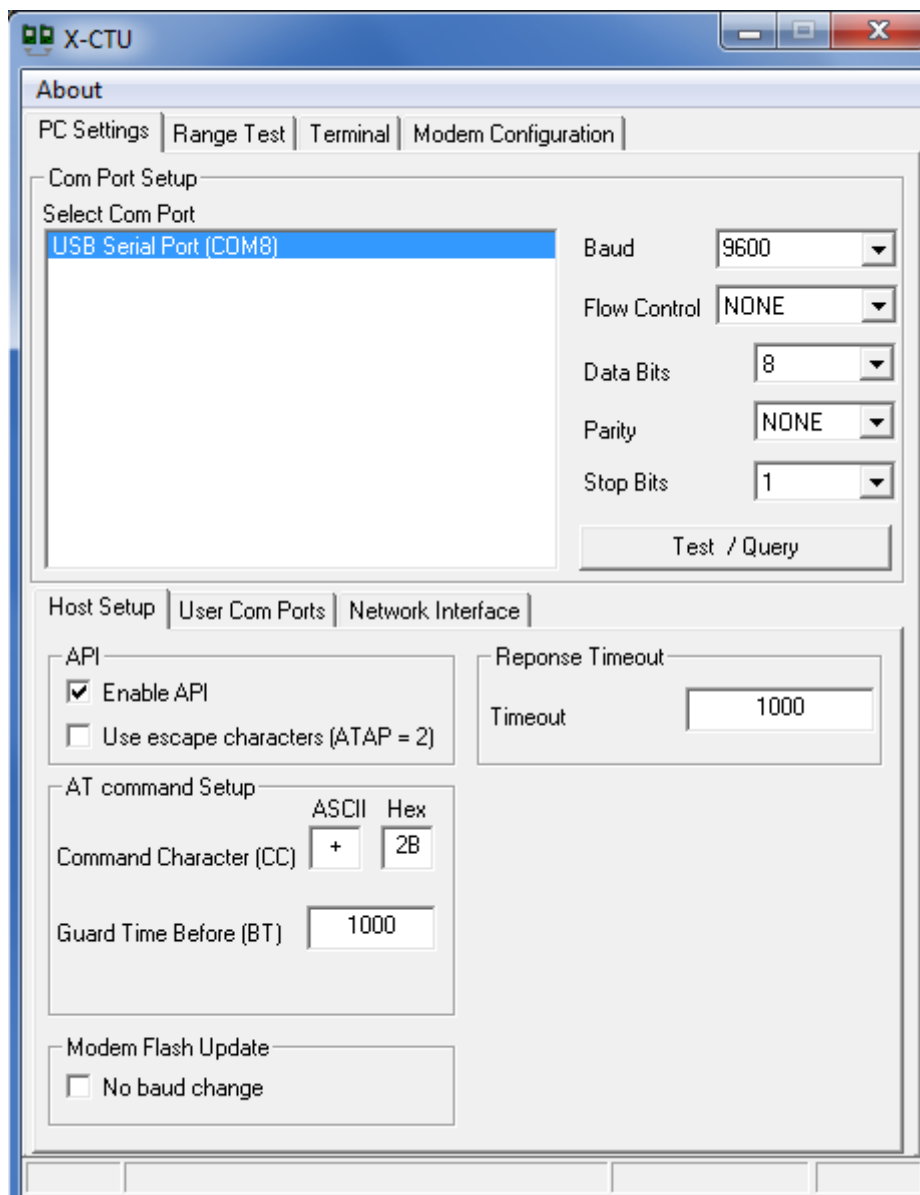


Figura 2-29. Captura pantalla config. mòdul Xbee lectura potenciòmetre

- Anar a *Modem Configuration* i seleccionar *Read*:

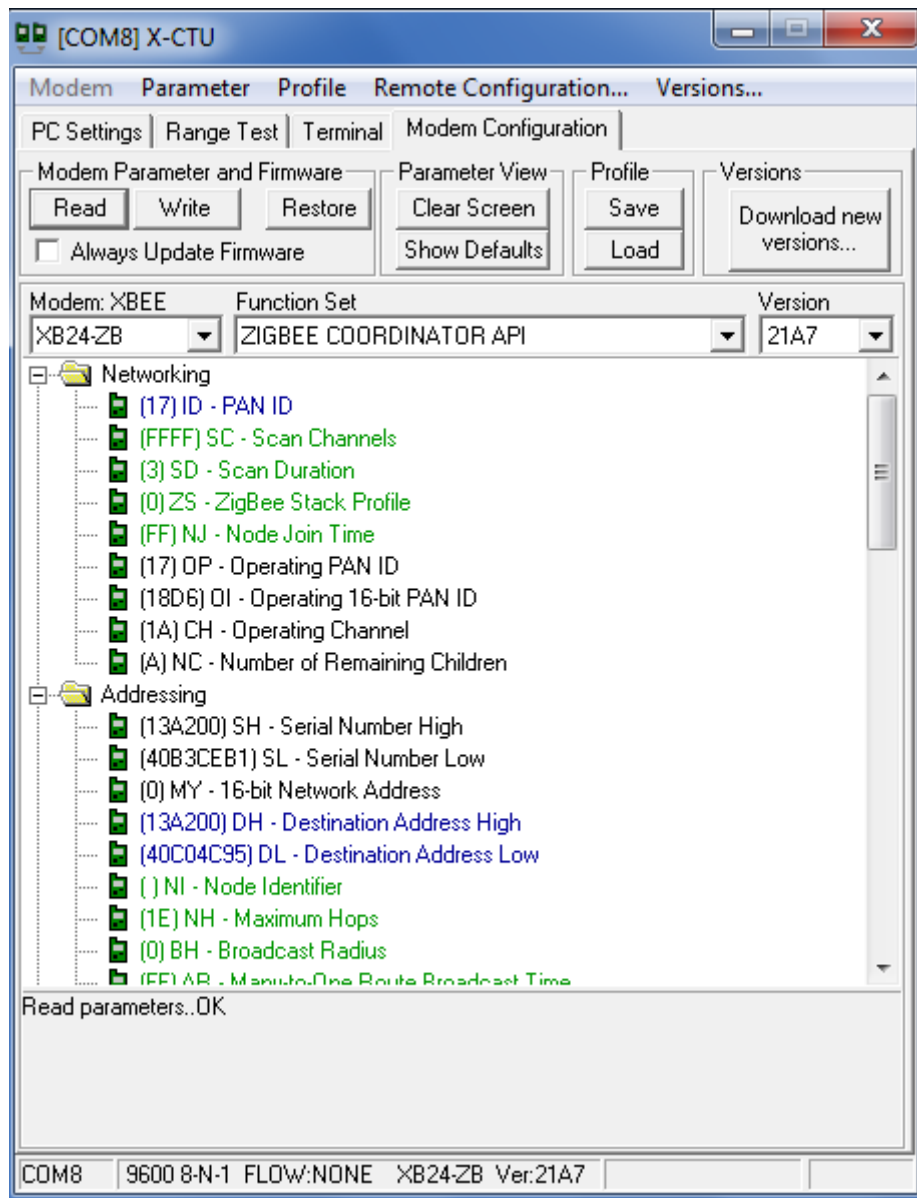


Figura 2-30. Captura pantalla config. mòdul Xbee lectura potenciómetre

- Definir *PAN ID* com a 17.
- Es pot observar com el mòdul automàticament ens defineix els següents paràmetres:
 - *Operating PAN ID (OP)*: 17.
 - *Operating 16-bit PAN ID (OI)*: 18D6.
 - *Operating Chanel (CH)*: 1A ha escollit aquest canal com el més net.

- Seleccionar *Write* i guardar els paràmetres.

Configuració Router

- Desactivar la configuració API.
- Anar a *Modem Configuration* i seleccionar *Read*.

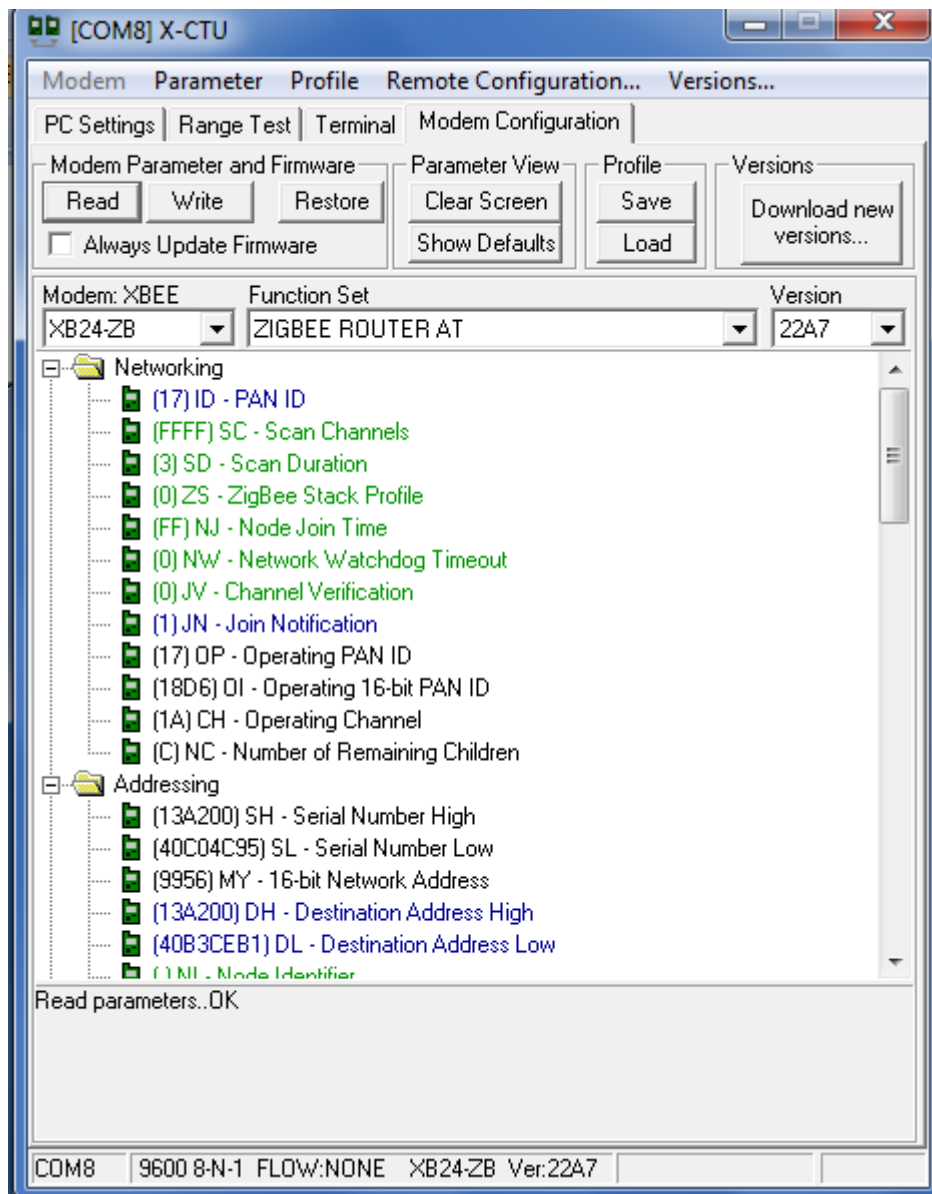


Figura 2-31. Captura pantalla config. mòdul Xbee lectura potenciómetre

- Definir *PAN ID* com a 17 i seleccionar *Write*.

- Es pot observar com el mòdul automàticament es defineix amb els paràmetres del coordinador:
 - *Operating PAN ID (OP)*: 17.
 - *Operating 16-bit PAN ID (OI)*: 18D6.
 - *Operating Chanel (CH)*: 1A.
- Dins la carpeta *Addressing* les *DH* i *DL* vénen en blanc per defecte; en aquest dos punts cal posar la *SH* i *SL* del coordinador.
- Anar a la carpeta *I/O Settings* i configurar *I/O D0* com a analògica:

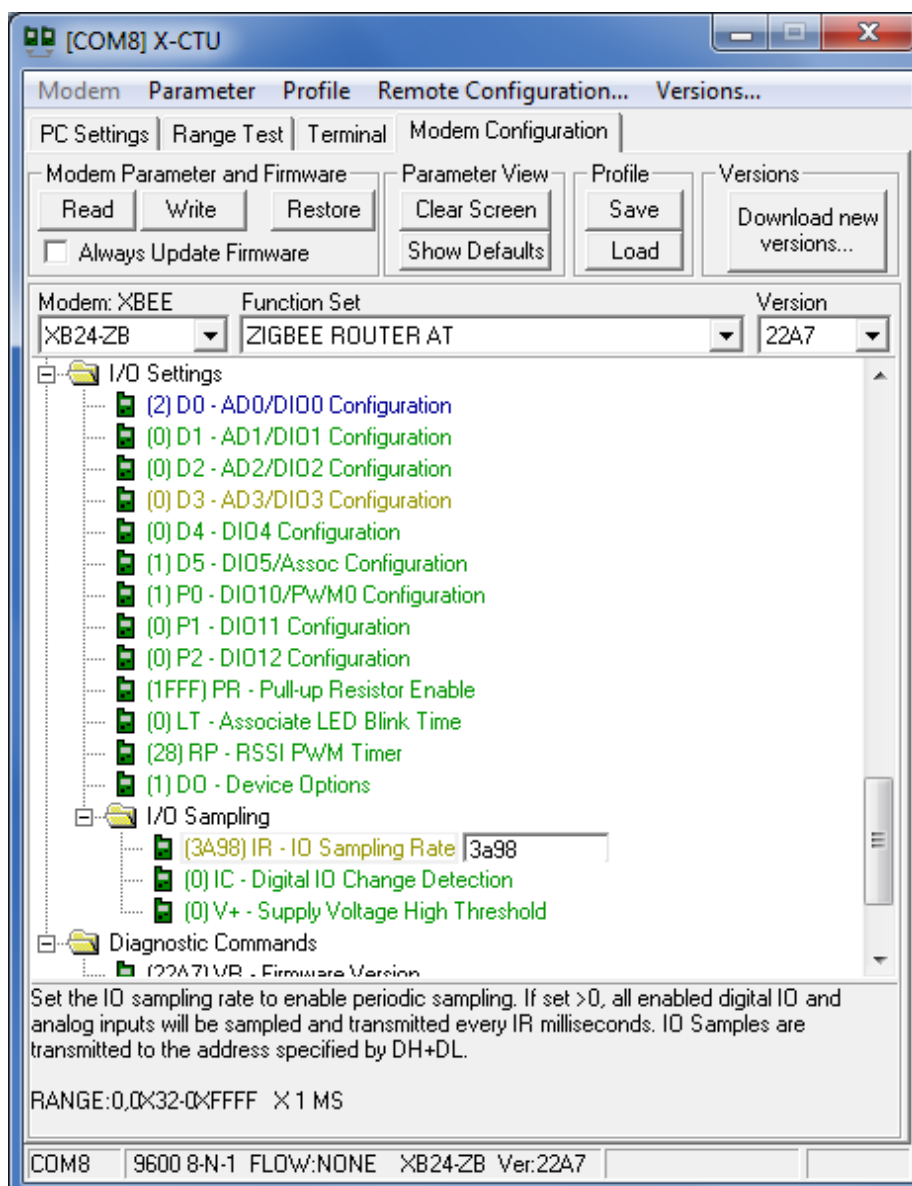


Figura 2-32. Captura pantalla config. mòdul Xbee lectura potenciómetre

- Anar a la subcarpeta *I/O Sampling* dins de la carpeta *I/O Settings* i configurar el paràmetre *I/O Sampling Rate* a 3A98, que significa que el

mòdul farà una lectura de l'estat de les seves entrades i sortides cada 15 segons.

- Seleccionar *Write* i guardar els paràmetres.

Programa Arduino

```
#define VERSION "1.02"
#include <SoftwareSerial.h>
SoftwareSerial Xbee(0,1); // RX, TX

int LED = 6;
int analogValue = 0;
float analogValuePot = 0;

void setup(){
  pinMode(LED,OUTPUT);
  Serial.begin(9600);
  Xbee.begin(9600);
}

void loop(){
  if (Serial.available() >= 23){
    if (Serial.read() == 0x7E){
      for (int i = 0; i<20; i++){
        byte discard = Serial.read();
      }
      int analogHigh = Serial.read();
      int analogLow = Serial.read();
      analogValue = analogLow + analogHigh*256;
      analogValuePot = (1.2 / 1023) * analogValue;
    }
  }
  if (analogValuePot > 0 && analogValuePot <= 1){
    digitalWrite (LED, HIGH);
```

```
}  
if (analogValuePot > 1 ){  
  digitalWrite (LED, LOW);  
}  
}
```

**CAL RECORDAR QUE S'HA DE RETIRAR EL SHIELD ABANS DE CARREGAR
EL PROGRAMA A L'ARDUINO.**

2.3.5 Lectura potenciòmetre mitjançant trames API i activar Led

Explicació prova

En aquest assaig s'ha utilitzat un Arduino amb el coordinador Xbee connectats entre ells amb una Shield, on arribarà la lectura del mòdul Xbee que té el potenciòmetre.

El microcontrolador està programat per tal de poder saber quin voltatge hi ha a la sortida del potenciòmetre (igual que en l'exemple anterior).

Aquest valor es mostrarà per la pantalla LCD.

Si el voltatge és un valor entre 0 i 1 s'encendrà el led de la sortida 6 i s'apagarà el led de la sortida 7 de la placa de l'Arduino i s'enviarà l'ordre d'activació del led de la sortida D3 del mòdul Xbee on hi ha el potenciòmetre.

Si el voltatge és un valor superior a 1 s'encendrà el led de la sortida 7 i s'apagarà el led de la sortida 6 de la placa de l'Arduino i s'enviarà l'ordre de desactivació del led de la sortida D3 del mòdul Xbee on hi ha el potenciòmetre.

Per poder fer això s'ha creat la següent funció en el codi de l'Arduino, que permet enviar l'ordre d'activació o desactivació del led:

```
void setRemoteState (char value){           //Activar o desactivar sortida
Serial.write(0x7E);                         // Inici trama.
Serial.write((byte)0x00);                   // Longitud trama
Serial.write(0x10);                         // Longitud trama
Serial.write(0x17);                         // Trama API
Serial.write((byte)0x00);                   // ACK en aquest cas 0 pq no volem resposta
Serial.write((byte)0x00);                   // SH mòdul destí sempre és igual en tots els mòduls
Serial.write(0x13);                         // SH mòdul destí sempre és igual en tots els mòduls
Serial.write(0xA2);                         // SH mòdul destí sempre és igual en tots els mòduls
Serial.write((byte)0x00);                   // SH mòdul destí sempre és igual en tots els mòduls
Serial.write(0x40);                         // SL mòdul destí sempre comença per 40
Serial.write(0xC0);                         // SL mòdul destí ja diferent per cada mòdul
Serial.write(0x4C);                         // SL mòdul destí ja diferent per cada mòdul
Serial.write(0x95);                         // SL mòdul destí ja diferent per cada mòdul
Serial.write(0xFF);                         // Adreça curta 16 bits com que és desconeguda FFEE
```

```
Serial.write(0xFE); // Adreça curta 16 bits com que és desconeguda FFEE ja que cada cop
                    // que es desconnecta i connecta el coordinador n'assigna una
Serial.write(0x02); // Paràmetre perquè s'apliquin els canvis,
Serial.write(0x44); // Especifiquem la sortida D en hexadecimal segons ASCII
Serial.write(0x33); // Especifiquem la sortida 3 en hexadecimal segons ASCII
Serial.write(value); // Si volem activar 05 o desactivar 04.
long sum = 0x17 + 0x13 + 0xA2 + 0x40 + 0xC0 +0x4C +0x95 + 0xFF + 0xFE +
0x02 + 0x44 + 0x33 + value;
// suma per calcular el checksum, suma tots els bytes de la trama començant des del tipus de trama API fins l'últim byte de
//la trama.
Serial.write(0xFF - (sum & 0xFF)); // S'agafa el byte de menys pes i es resta de FF
}
```

Muntatge circuit

Cal col·locar un potenciòmetre que simularà les lectures d'un sensor; aquest potenciòmetre va acompanyat d'un condensador de 100microF per evitar "interferències".

S'ha de muntar un divisor de tensió a l'entrada del potenciòmetre per tal d'adequar el rang del voltatge d'entrada al potenciòmetre al rang de les entrades analògiques del mòdul Xbee (0 a 1.2 V).

El potenciòmetre anirà connectat a l'entrada D0 del mòdul Xbee.

Cal col·locar un led a la sortida D3 del mòdul Xbee que llegeix el potenciòmetre.

A més, també s'han de col·locar els tres leds indicadors, tal com s'explica a l'apartat Revelacions i Indicacions de la memòria.

En la placa de proves de l' Arduino cal connectar un led a la sortida 6 i un altre a la 7 de l'Arduino, i la pantalla LCD.

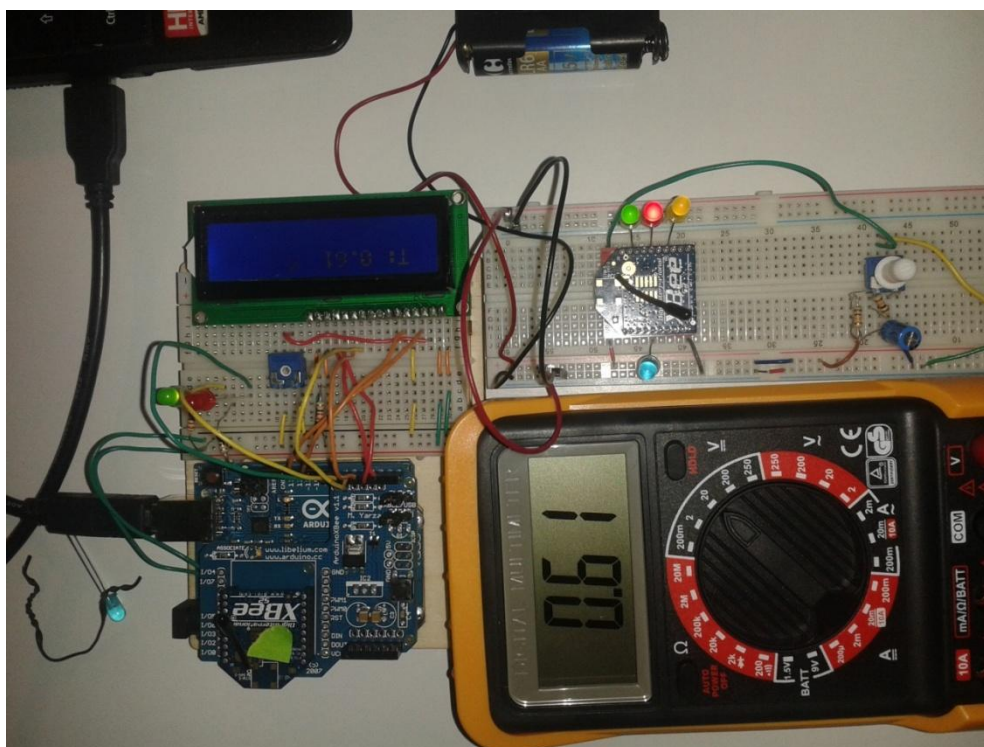


Figura 2-33. Muntatge del circuit de la prova Lectura potenciòmetre mitjançant trames API i activar Led

Configuració mòduls Xbee

La configuració dels mòduls és exactament igual que en la pràctica anterior. L'únic canvi és que cal pensar a configurar la sortida D3 del Router com a Digital Low.

Codi Arduino

```
#define VERSION "1.02"
#include <SoftwareSerial.h>
#include <LiquidCrystal.h>
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
SoftwareSerial Xbee(0,1);           // RX, TX

int LED = 6;
int LED2= 7;
int analogValue = 0;
float analogValuePot = 0;

void setup(){
  pinMode(LED,OUTPUT);
  pinMode(LED2,OUTPUT);
  Serial.begin(9600);
  Xbee.begin(9600);
}

//FUNCIONS

void setRemoteState (char value){
  Serial.write(0x7E);
  Serial.write(0x00);
  Serial.write(0x10);
  Serial.write(0x17);
  Serial.write(0x00);
  Serial.write(0x00);
  Serial.write(0x13);
  Serial.write(0xA2);
  Serial.write(0x00);
  Serial.write(0x40);
  Serial.write(0xC0);
  Serial.write(0x4C);
  Serial.write(0x95);
  Serial.write(0xFF);
}
```

```

Serial.write(0xFE);
Serial.write(0x02);
Serial.write(0x44);
Serial.write(0x33);
Serial.write(value);
long sum = 0x17 + 0x13 + 0xA2 + 0x40 + 0xC0 + 0x4C + 0x95 + 0xFF + 0xFE +
0x02 + 0x44 + 0x33 + value;
Serial.write(0xFF - (sum & 0xFF));
}

void setPantalla (){
  lcd.begin(16, 2);
  lcd.setCursor(1, 0);
  lcd.print("V:");
  lcd.setCursor(4, 0);
  lcd.print(analogValuePot);
  lcd.setCursor(10, 0);
  lcd.print("V");
}

void loop(){
  if (Serial.available() >= 23){
    if (Serial.read() == 0x7E){
      for (int i = 0; i<20; i++){
        byte discard = Serial.read();
      }
      int analogHigh = Serial.read();
      int analogLow = Serial.read();
      analogValue = analogLow + analogHigh*256;
      analogValuePot = (1.2 / 1023) * analogValue;
    }
  }
  if (analogValuePot > 0 && analogValuePot <= 1){
    digitalWrite (LED, HIGH);
    digitalWrite (LED2, LOW);
    setRemoteState(0x05);
    setPantalla();
  }
  if (analogValuePot > 1 ){
    digitalWrite (LED2, HIGH);
    digitalWrite (LED, LOW);
    setRemoteState(0x04);
    setPantalla();
  }
}

```


2.4 Configuració Arduino Ethernet

Els passos que s'han seguit per configurar l'Arduino Ethernet han estat els següents:

- Executar l'ordre *cmd* a la pestanya *Iniciar* de Windows.

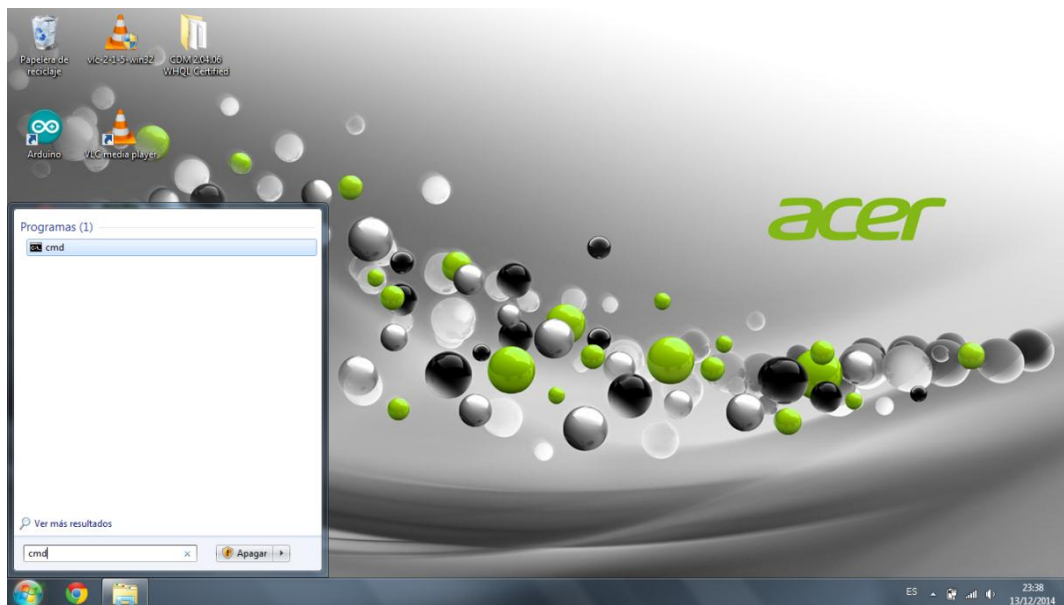


Figura 2-34. Captura pantalla configuració Arduino Ethernet

- Donar l'ordre *ipconfig* per saber l'adreça *ip* de l'ordinador.

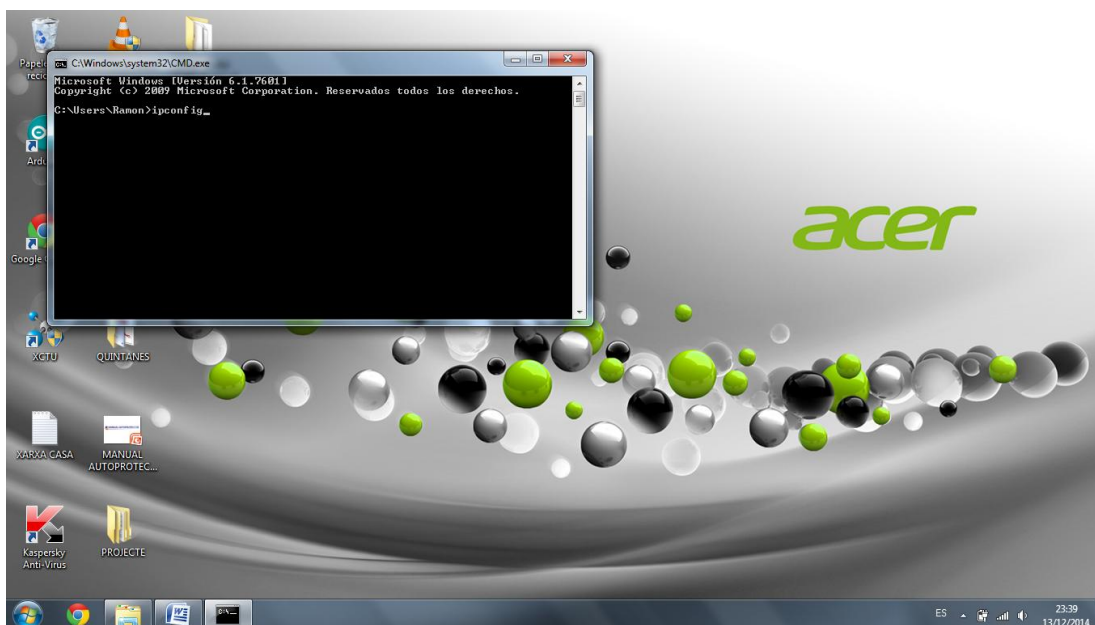


Figura 2-35. Captura pantalla configuració Arduino Ethernet

- Apareix l'adreça *ip* del PC, que serà l'adreça del servidor.

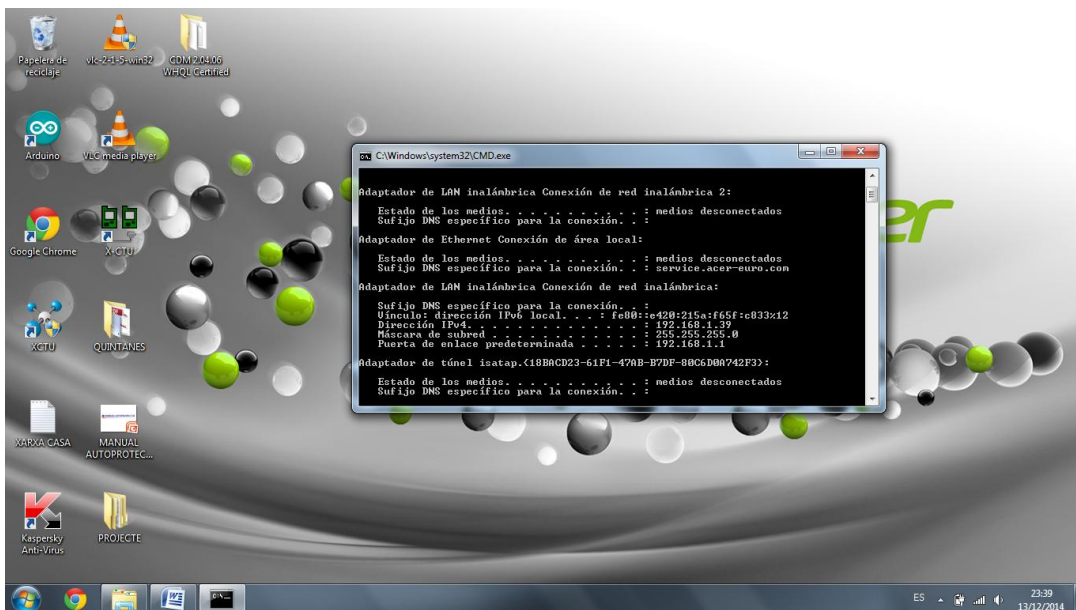


Figura 2-36. Captura pantalla configuraci n Arduino Ethernet

- Per definir l'adreça *ip* de l'Arduino cal sumar 10 a l'adreça del PC; d'aquesta manera s'intenta no utilitzar cap adreça que el Router ja hagi assignat a un altre equip.

Annex 3 . Implantació Xarxa

3.1 Configuració mòduls Xbee

3.1.1 Configuració mòdul NI0101 (Sensors + relé ventilador)

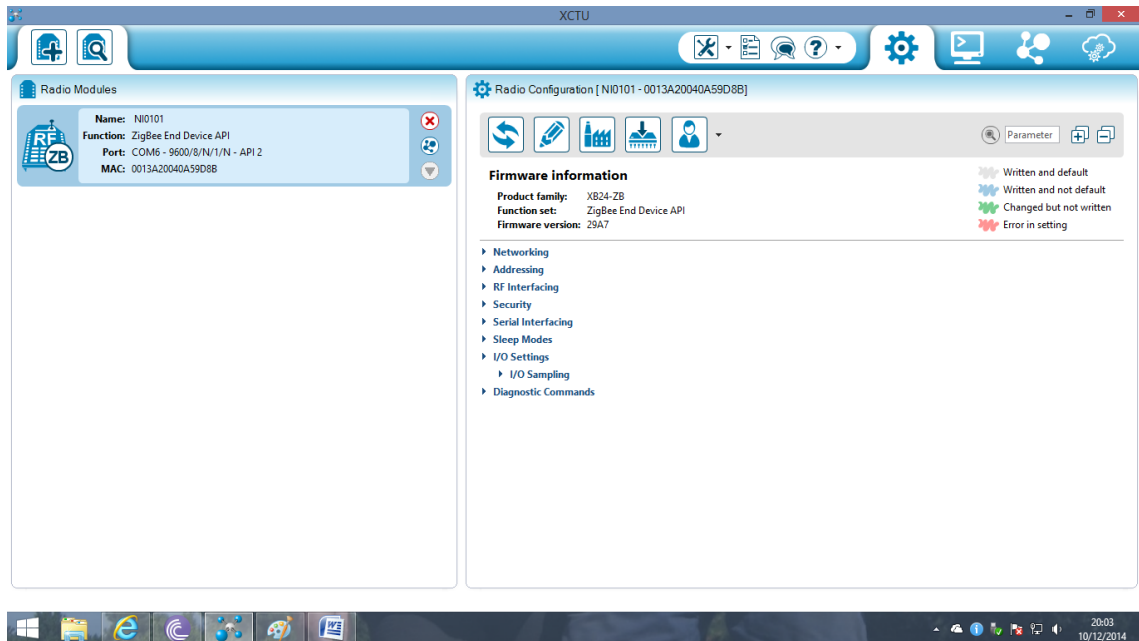


Figura 3-1. Captura pantalla configuració mòdul NI 0101

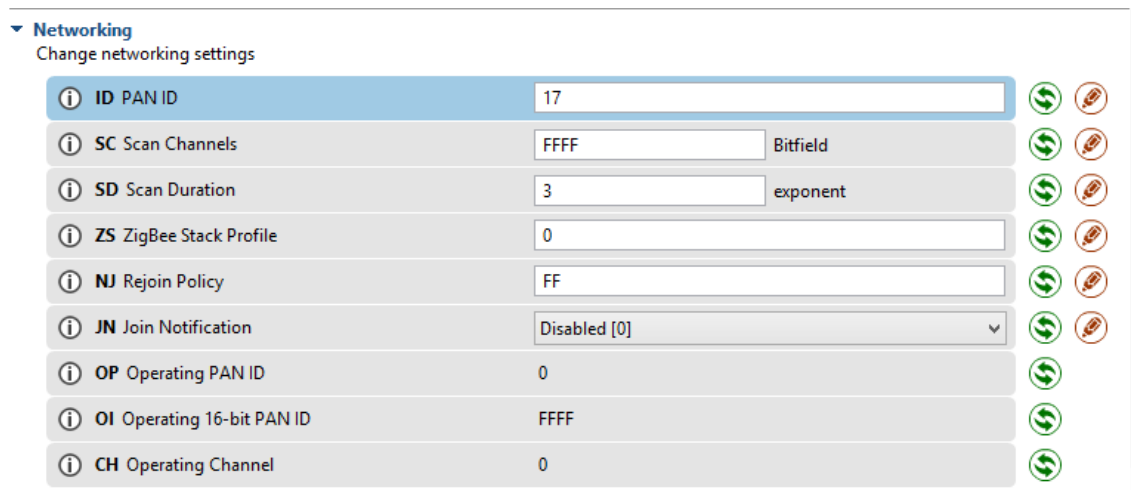


Figura 3-2. Captura pantalla configuració mòdul NI 0101

▼ **Addressing**

Change addressing settings

SH Serial Number High	13A200		
SL Serial Number Low	40A59D8B		
MY 16-bit Network Address	FFFE		
MP 16-bit Parent Address	FFFE		
DH Destination Address High	13A200		
DL Destination Address Low	40B3CEB1		
NI Node Identifier	NI0101		
NH Maximum Hops	30		
BH Broadcast Radius	0		
DD Device Type Identifier	30000		
NT Node Discovery Backoff	3C x 100 ms		
NO Node Discovery Options	0		
NP Maximum Number of API Transmission Bytes	FF		
CR PAN Conflict Threshold	3		

Figura 3-3. Captura pantalla configuració mòdul NI 0101

▼ **RF Interfacing**

Change RF interface options

PL Power Level	Highest [4]		
PM Power Mode	Boost Mode Enabled [1]		
PP Power at PL4	3		

▼ **Security**

Change security parameters

EE Encryption Enable	Disabled [0]		
EO Encryption Options	0 Bitfield		
KY Encryption Key			

Figura 3-4. Captura pantalla configuració mòdul NI 0101

▼ **Serial Interfacing**

Change modem interfacing options

BD Baud Rate	9600 [3]		
NB Parity	No Parity [0]		
SB Stop Bits	One stop bit [0]		
D7 DIO7 Configuration	CTS flow control [1]		
D6 DIO6 Configuration	Disable [0]		
AP API Enable	2 API enabled (... escaping [2])		
AO API Output Mode	Native [0]		

Figura 3-5. Captura pantalla configuració mòdul NI 0101

▼ Sleep Modes

Configure low power options for end devices













SM Sleep Mode	Cyclic Sleep [4]	 
ST Time before Sleep	1388 x 1 ms	 
SP Cyclic Sleep Period	5DC x 10 ms	 
SN Number of Cycles to power down IO	2	 
SO Sleep Options	4	 
PO Poll Rate	0	 

Figura 3-6. Captura pantalla configuració mòdul NI 0101

▼ I/O Settings

Modify DIO and ADC options



























D0 AD0/DIO0 Configuration	ADC [2]	 
D1 AD1/DIO1 Configuration	ADC [2]	 
D2 AD2/DIO2 Configuration	ADC [2]	 
D3 AD3/DIO3 Configuration	Disabled [0]	 
D4 DIO4 Configuration	Disabled [0]	 
D5 DIO5/Assoc Configuration	Associated indicator [1]	 
P0 DIO10/PWM0 Configuration	Disabled [0]	 
P1 DIO11 Configuration	Disabled [0]	 
P2 DIO12 Configuration	Digital Out, Low [4]	 
PR Pull-up Resistor Enable	1FFF	 
LT Associate LED Blink Time	0 x10 ms	 
RP RSSI PWM Timer	28 x 100 ms	 
DO Device Options	1 Bitfield	 

Figura 3-7. Captura pantalla configuració mòdul NI 0101

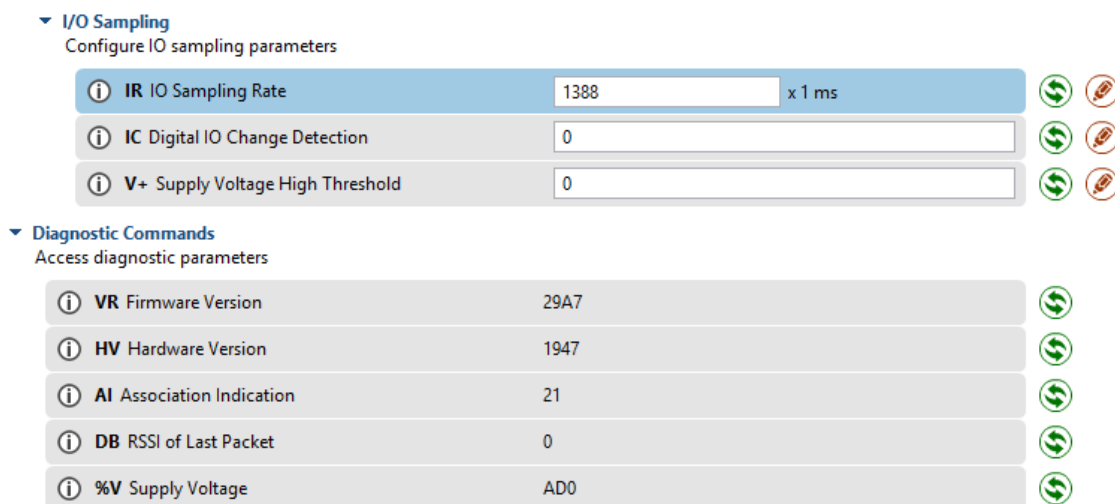


Figura 3-8. Captura pantalla configuració mòdul NI 0101

3.1.2 Configuració mòdul RI0101 (Relé caldera)

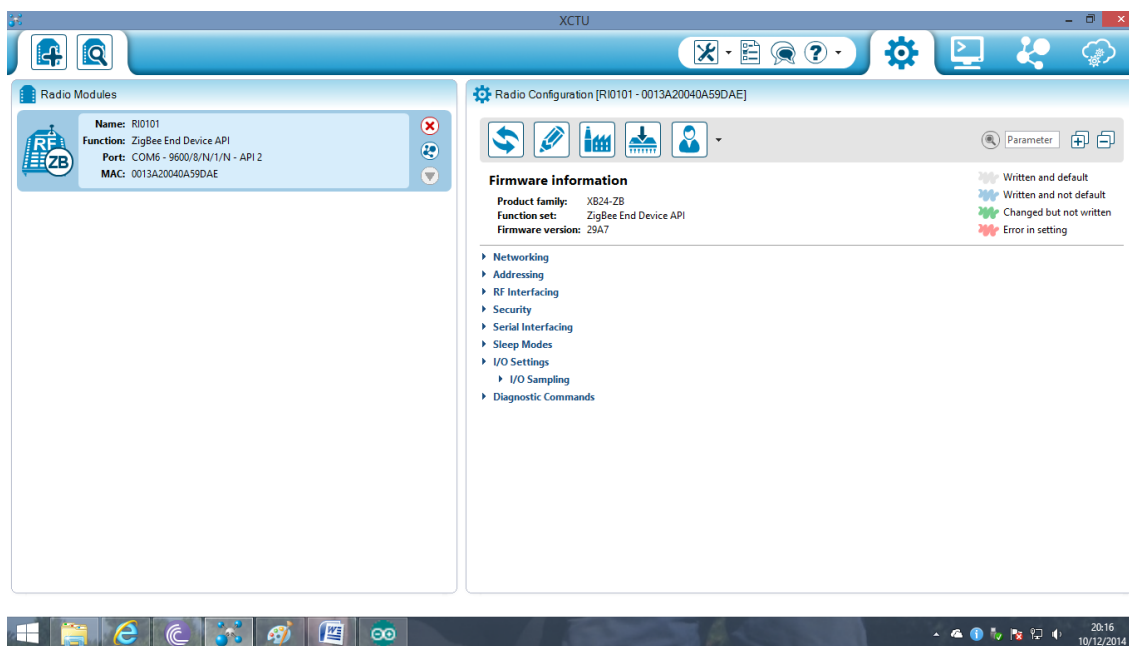


Figura 3-9. Captura pantalla configuració mòdul RI 0101

▼ **Networking**

Change networking settings














ID PAN ID	17	 
SC Scan Channels	FFFF Bitfield	 
SD Scan Duration	3 exponent	 
ZS ZigBee Stack Profile	0	 
NJ Rejoin Policy	FF	 
JN Join Notification	Disabled [0]	 
OP Operating PAN ID	0	
OI Operating 16-bit PAN ID	FFFF	
CH Operating Channel	0	

Figura 3-10. Captura pantalla configuració mòdul RI 0101

▼ **Addressing**

Change addressing settings





























SH Serial Number High	13A200	
SL Serial Number Low	40A59D8B	
MY 16-bit Network Address	FFFE	
MP 16-bit Parent Address	FFFE	
DH Destination Address High	13A200	 
DL Destination Address Low	40B3CEB1	 
NI Node Identifier	NI0101	 
NH Maximum Hops	30	 
BH Broadcast Radius	0	 
DD Device Type Identifier	30000	 
NT Node Discovery Backoff	3C x 100 ms	 
NO Node Discovery Options	0	 
NP Maximum Number of API Transmission Bytes	FF	
CR PAN Conflict Threshold	3	 

Figura 3-11. Captura pantalla configuració mòdul RI 0101

▼ **RF Interfacing**

Change RF interface options

PL Power Level	Highest [4]	 
PM Power Mode	Boost Mode Enabled [1]	 
PP Power at PL4	3	

▼ **Security**

Change security parameters







EE Encryption Enable	Disabled [0]	 
EO Encryption Options	0 Bitfield	 
KY Encryption Key		 

Figura 3-12. Captura pantalla configuració mòdul RI 0101

▼ **Serial Interfacing**

Change modem interfacing options















BD Baud Rate	9600 [3]	 
NB Parity	No Parity [0]	 
SB Stop Bits	One stop bit [0]	 
D7 DIO7 Configuration	CTS flow control [1]	 
D6 DIO6 Configuration	Disable [0]	 
AP API Enable	2 API enabled (... escaping (2)	 
AO API Output Mode	Native [0]	 

Figura 3-13. Captura pantalla configuració mòdul RI 0101

▼ **Sleep Modes**

Configure low power options for end devices













SM Sleep Mode	Cyclic Sleep [4]	 
ST Time before Sleep	1388 x 1 ms	 
SP Cyclic Sleep Period	5DC x 10 ms	 
SN Number of Cycles to power down IO	2	 
SO Sleep Options	4	 
PO Poll Rate	0	 

Figura 3-14. Captura pantalla configuració mòdul RI 0101

▼ **I/O Settings**

Modify DIO and ADC options



























D0 AD0/DIO0 Configuration	Disabled [0]	 
D1 AD1/DIO1 Configuration	Disabled [0]	 
D2 AD2/DIO2 Configuration	Disabled [0]	 
D3 AD3/DIO3 Configuration	Disabled [0]	 
D4 DIO4 Configuration	Disabled [0]	 
D5 DIO5/Assoc Configuration	Associated indicator [1]	 
P0 DIO10/PWM0 Configuration	RSSI PWM Output [1]	 
P1 DIO11 Configuration	Disabled [0]	 
P2 DIO12 Configuration	Digital Out, Low [4]	 
PR Pull-up Resistor Enable	1FFF	 
LT Associate LED Blink Time	0 x10 ms	 
RP RSSI PWM Timer	28 x 100 ms	 
DO Device Options	1 Bitfield	 

Figura 3-15. Captura pantalla configuració mòdul RI 0101

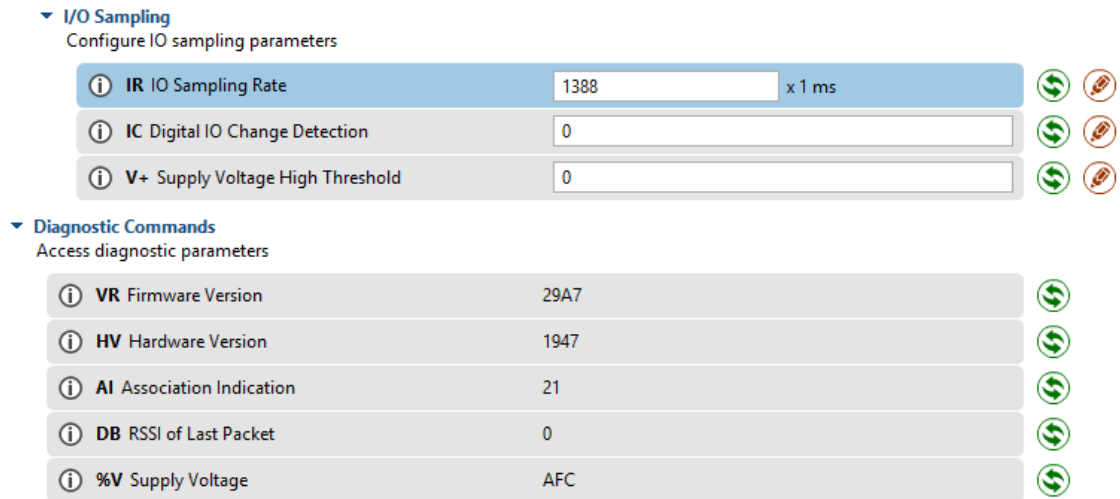


Figura 3-16. Captura pantalla configuració mòdul RI 0101

3.1.3 Configuració mòdul RI0102 (Relé llum)

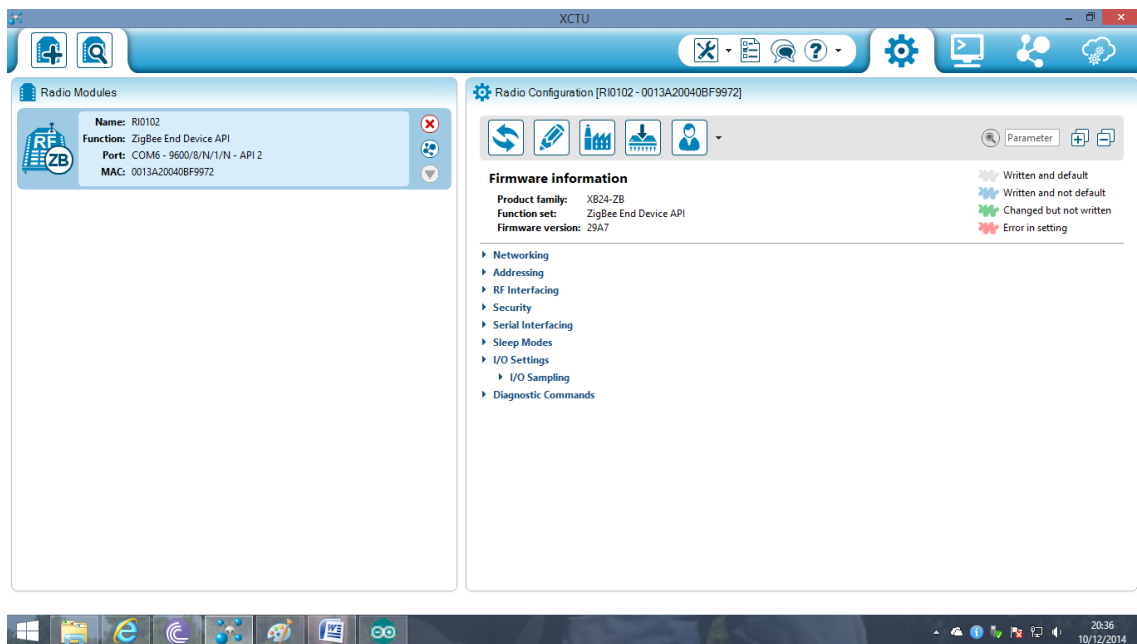


Figura 3-17. Captura pantalla configuració mòdul RI 0102

▼ **Networking**

Change networking settings

ID PAN ID	17	 
SC Scan Channels	FFFF Bitfield	 
SD Scan Duration	3 exponent	 
ZS ZigBee Stack Profile	0	 
NJ Rejoin Policy	FF	 
JN Join Notification	Disabled [0]	 
OP Operating PAN ID	0	
OI Operating 16-bit PAN ID	FFFF	
CH Operating Channel	0	

Figura 3-18. Captura pantalla configuració mòdul RI 0102

▼ **Addressing**

Change addressing settings
























SH Serial Number High	13A200	
SL Serial Number Low	40BF9972	
MY 16-bit Network Address	FFFE	
MP 16-bit Parent Address	FFFE	
DH Destination Address High	13A200	 
DL Destination Address Low	40B3CEB1	 
NI Node Identifier	RI0102	 
NH Maximum Hops	30	 
BH Broadcast Radius	0	 
DD Device Type Identifier	30000	 
NT Node Discovery Backoff	3C x 100 ms	 
NO Node Discovery Options	0	 
NP Maximum Number of API Transmission Bytes	FF	
CR PAN Conflict Threshold	3	 

Figura 3-19. Captura pantalla configuració mòdul RI 0102

▼ **RF Interfacing**
Change RF interface options

PL Power Level	Highest [4]		
PM Power Mode	Boost Mode Enabled [1]		
PP Power at PL4	3		

▼ **Security**
Change security parameters

EE Encryption Enable	Disabled [0]		
EO Encryption Options	0 Bitfield		
KY Encryption Key			

Figura 3-20. Captura pantalla configuració mòdul RI 0102

▼ **Serial Interfacing**
Change modem interfacing options

BD Baud Rate	9600 [3]		
NB Parity	No Parity [0]		
SB Stop Bits	One stop bit [0]		
D7 DIO7 Configuration	CTS flow control [1]		
D6 DIO6 Configuration	Disable [0]		
AP API Enable	2 API enabled (... escaping [2])		
AO API Output Mode	Native [0]		

Figura 3-21. Captura pantalla configuració mòdul RI 0102

▼ **Sleep Modes**
Configure low power options for end devices

SM Sleep Mode	Cyclic Sleep [4]		
ST Time before Sleep	1388 x 1 ms		
SP Cyclic Sleep Period	5DC x 10 ms		
SN Number of Cycles to power down IO	2		
SO Sleep Options	4		
PO Poll Rate	0		

Figura 3-22. Captura pantalla configuració mòdul RI 0102

▼ **I/O Settings**
Modify DIO and ADC options

































D0 AD0/DIO0 Configuration	Disabled [0]	 
D1 AD1/DIO1 Configuration	Disabled [0]	 
D2 AD2/DIO2 Configuration	Disabled [0]	 
D3 AD3/DIO3 Configuration	Disabled [0]	 
D4 DIO4 Configuration	Disabled [0]	 
D5 DIO5/Assoc Configuration	Associated indicator [1]	 
P0 DIO10/PWM0 Configuration	RSSI PWM Output [1]	 
P1 DIO11 Configuration	Disabled [0]	 
P2 DIO12 Configuration	Digital Out, Low [4]	 
PR Pull-up Resistor Enable	1FFF	 
LT Associate LED Blink Time	0 x10 ms	 
RP RSSI PWM Timer	28 x 100 ms	 
DO Device Options	1 Bitfield	 

Figura 3-23. Captura pantalla configuració mòdul RI 0102

▼ **I/O Sampling**
Configure IO sampling parameters

IR IO Sampling Rate	1388 x 1 ms	 
IC Digital IO Change Detection	0	 
V+ Supply Voltage High Threshold	0	 

▼ **Diagnostic Commands**
Access diagnostic parameters






VR Firmware Version	29A7	
HV Hardware Version	1947	
AI Association Indication	21	
DB RSSI of Last Packet	0	
%V Supply Voltage	AFC	

Figura 3-24. Captura pantalla configuració mòdul RI 0102

3.1.4 Configuració mòdul COORDINADOR

▼ **Networking**
Change networking settings















ID PAN ID	17	 
SC Scan Channels	FFFF Bitfield	 
SD Scan Duration	3 exponent	 
ZS ZigBee Stack Profile	0	 
NJ Node Join Time	FF x 1 sec	 
OP Operating PAN ID	17	
OI Operating 16-bit PAN ID	B360	
CH Operating Channel	14	
NC Number of Remaining Children	A	

Figura 3-25. Captura pantalla configuració mòdul Coordinador

▼ **Addressing**
Change addressing settings

























SH Serial Number High	13A200	
SL Serial Number Low	40B3CEB1	
MY 16-bit Network Address	0	
DH Destination Address High	FFFF	 
DL Destination Address Low	FFFF	 
NI Node Identifier	coordinator	 
NH Maximum Hops	1E	 
BH Broadcast Radius	0	 
AR Many-to-One Route Broadcast Time	FF x 10 sec	 
DD Device Type Identifier	30000	 
NT Node Discovery Backoff	3C x 100 ms	 
NO Node Discovery Options	3	 
NP Maximum Number of API Transmission Bytes	FF	
CR PAN Conflict Threshold	3	 

Figura 3-26. Captura pantalla configuració mòdul Coordinador

▼ **RF Interfacing**
Change RF interface options

PL Power Level	Highest [4]		
PM Power Mode	Boost Mode Enabled [1]		
PP Power at PL4	3		

▼ **Security**
Change security parameters

EE Encryption Enable	Disabled [0]		
EO Encryption Options	0 Bitfield		
KY Encryption Key			
NK Network Encryption Key			

Figura 3-27. Captura pantalla configuració mòdul Coordinador

▼ **Serial Interfacing**
Change modem interfacing options

BD Baud Rate	9600 [3]		
NB Parity	No Parity [0]		
SB Stop Bits	One stop bit [0]		
D7 DIO7 Configuration	CTS flow control [1]		
D6 DIO6 Configuration	Disable [0]		
AP API Enable	2 API enabled (... escaping (2)		
AO API Output Mode	Native [0]		

▼ **Sleep Modes**
Configure low power options to support end device children

SP Cyclic Sleep Period	20 x 10 ms		
SN Number of Cyclic Sleep Periods	1		

Figura 3-28. Captura pantalla configuració mòdul Coordinador

▼ **I/O Settings**
Modify DIO and ADC options

ⓘ	D0 AD0/DIO0 Configuration	Disabled [0]	↻	🚫
ⓘ	D1 AD1/DIO1 Configuration	Disabled [0]	↻	🚫
ⓘ	D2 AD2/DIO2 Configuration	Disabled [0]	↻	🚫
ⓘ	D3 AD3/DIO3 Configuration	Disabled [0]	↻	🚫
ⓘ	D4 DIO4 Configuration	Disabled [0]	↻	🚫
ⓘ	D5 DIO5/Assoc Configuration	Associated indicator [1]	↻	🚫
ⓘ	P0 DIO10/PWM0 Configuration	Disabled [0]	↻	🚫
ⓘ	P1 DIO11 Configuration	Disabled [0]	↻	🚫
ⓘ	P2 DIO12 Configuration	Disabled [0]	↻	🚫
ⓘ	PR Pull-up Resistor Enable	1FFF	↻	🚫
ⓘ	LT Associate LED Blink Time	0 x10 ms	↻	🚫
ⓘ	RP RSSI PWM Timer	28 x 100 ms	↻	🚫
ⓘ	DO Device Options	1 Bitfield	↻	🚫

Figura 3-29. Captura pantalla configuració mòdul Coordinador

▼ **I/O Sampling**
Configure IO sampling parameters

ⓘ	IR IO Sampling Rate	0 x 1 ms	↻	🚫
ⓘ	IC Digital IO Change Detection	0	↻	🚫
ⓘ	V+ Supply Voltage High Threshold	0	↻	🚫

▼ **Diagnostic Commands**
Access diagnostic parameters

ⓘ	VR Firmware Version	21A7	↻
ⓘ	HV Hardware Version	194B	↻
ⓘ	AI Association Indication	0	↻
ⓘ	DB RSSI of Last Packet	0	↻
ⓘ	%V Supply Voltage	ADB	↻

Figura 3-30. Captura pantalla configuració mòdul Coordinador

3.2 Codi Arduino

```
#define VERSION "1.01"
#include <SoftwareSerial.h>
#include <LiquidCrystal.h>
#include <SPI.h>
#include <Ethernet.h>
#include <avr/wdt.h>

EthernetClient client;
SoftwareSerial xbee(0,1);

//VARIABLES

byte mac[] = {0xDE,0xAD,0xBE,0xEF,0xFE,0xED};
byte ip[] = {192,168,1,49};
char server[] = "192.168.1.39";
LiquidCrystal lcd(7, 6, 5, 4, 3, 2);
int LEDVENTILADOR = 8;
int LEDCALDERA= 9;
int LEDLLUM= 14;
int LEDAUTOVENTILADOR= 15;
int LEDAUTOCALDERA= 16;
int LEDAUTOLLUM= 17;
int analogHigh = 0;
int analogLow =0;
int analogHigh2 = 0;
int analogLow2 =0;
int analogHigh3 = 0;
int analogLow3 =0;
int analogValueT = 0;
float analogValueTem = 0;
float temperatura_actual = 0;
int analogValueB = 0;
float analogValueBV = 0;
int analogValueLL = 0;
float llum_actual = 0;
int ventilador = 0;
int llums = 0;
int caldera = 0;
int b0=0;
int b1=0;
int b2=0;
int b3=0;
float temperatura_objectiu = 0;
```



```
float llum_objectiu = 0;
int ordre_ventilador = 0;
int ordre_caldera = 0;
int ordre_llum = 0;
float temperatura_actual2 = 100;
float llum_actual2 = 100;
unsigned long lastConnectionTime = 0;
boolean lastConnected = false;
const unsigned long postingInterval = 5*1000;
int x=0;
char lf=10;

void setup(){
  wdt_disable();
  pinMode(LEDVENTILADOR,OUTPUT);
  pinMode(LEDALDERA,OUTPUT);
  pinMode(LEDLLUM,OUTPUT);
  pinMode(LEDAUTOCALDERA,OUTPUT);
  pinMode(LEDAUTOVENTILADOR,OUTPUT);
  pinMode(LEDAUTOLLUM,OUTPUT);
  Serial.begin(9600);
  xbee.begin(9600);
  Serial.begin(9600);
  delay(1000);
  Ethernet.begin(mac, ip);
  wdt_enable(WDTO_8S);
  setRemoteState(0xC0, 0x4C, 0x95, 0x04);
  setRemoteState(0xA5, 0x9D, 0x8B, 0x04);
  setRemoteState(0xBF, 0x99, 0x72, 0x04);
}
```

//FUNCIO ENVIAMENT TRAMA ACTIVACIO DESACTIVACIO RELE

```
void setRemoteState (char a, char b, char c, char value){
  Serial.write(0x7E);
  Serial.write((byte)0x00);
  Serial.write(0x10);
  Serial.write(0x17);
  Serial.write((byte)0x00);
  Serial.write((byte)0x00);
  Serial.write(0x13);
  Serial.write(0xA2);
  Serial.write((byte)0x00);
  Serial.write(0x40);
  Serial.write(a);
  Serial.write(b);
```

```

Serial.write(c);
Serial.write(0xFF);
Serial.write(0xFE);
Serial.write(0x02);
Serial.write(0x50);
Serial.write(0x32);
Serial.write(value);
long sum = 0x17 + 0x13 + 0xA2 + 0x40 + a + b + c + 0xFF + 0xFE + 0x02 + 0x50
+ 0x32 + value;

```

```

Serial.write(0xFF - (sum & 0xFF));
}

```

//FUNCIÓ LCD

```

void setPantalla (int s){
  lcd.begin(16, 2);
  lcd.setCursor(1, 0);
  lcd.print("T:");
  lcd.setCursor(6, 0);
  lcd.print(temperatura_actual);
  lcd.setCursor(12, 0);
  lcd.print("C");
  lcd.setCursor(1, 1);
  lcd.print("LL:");
  lcd.setCursor(6, 1);
  lcd.print(llum_actual);
  lcd.setCursor(12, 1);
  lcd.print("V");

  temperatura_actual2 = temperatura_actual;
  llum_actual2 = llum_actual;
}

```

```

void setPantalla2(char k, char e, char b, char s){
  lcd.begin(16, 2);
  lcd.setCursor(1, 0);
  lcd.print(k);
  lcd.setCursor(4, 0);
  lcd.print(e);
  lcd.setCursor(7, 0);
  lcd.print(b);
  lcd.setCursor(10, 0);
  lcd.print(s);
  lcd.setCursor(1, 1);
  lcd.print("BATERIA BAIXA");
}

```

```

}

void httpRequest()
{
  if (client.connect(server, 80)) {
    Serial.println("connecting...");
    client.println("GET
/calendari/calendari.php?zona=1&lectura="+(String)temperatura_actual2+"&llum="
+(String)llum_actual2+" HTTP/1.1");
    client.println("Host: 192.168.1.39");
    client.println("User-Agent: arduino-ethernet");
    client.println("Connection: close");
    client.println();
    lastConnectionTime = millis();
  }
  else {
    Serial.println("connection failed");
    Serial.println("disconnecting.");
    client.stop();
  }
}

```

```

String getValue(String data, char separator, int index)
{
  int found = 0;
  int strIndex[] = {0, -1 };
  int maxIndex = data.length()-1;
  for(int i=0; i<=maxIndex && found<=index; i++){
    if(data.charAt(i)==separator || i==maxIndex){
      found++;
      strIndex[0] = strIndex[1]+1;
      strIndex[1] = (i == maxIndex) ? i+1 : i;
    }
  }
  return found>index ? data.substring(strIndex[0], strIndex[1]) : "";
}

```

```

void loop(){

String valors_web = "";
if (Serial.available() >= 23){
  if (Serial.read() == 0x7E){
    for (int y=0; y<2; y++){
      byte discard = Serial.read();
    }
    if (Serial.read() == 0x92){

```

```

delay (100);
for (int y=0; y<5 ; y++){
    byte discard = Serial.read();
}
b0 = Serial.read();
b1 = Serial.read();
b2 = Serial.read();
b3 = Serial.read();
for (int i = 0; i<9; i++){
    byte discard = Serial.read();
}
analogHigh = Serial.read();
analogLow = Serial.read();
analogValueT = analogLow +(analogHigh*256);
analogValueTem = (1.2 / 1023) * analogValueT;
temperatura_actual = (analogValueTem -1.2)/-0.01177;
analogHigh2 = Serial.read();
analogLow2 = Serial.read();
analogValueB = analogLow2 +(analogHigh2*256);
analogValueBV = (1.2 / 1023) * analogValueB;
analogHigh3 = Serial.read();
analogLow3= Serial.read();
analogValueLL = analogLow3 + (analogHigh3*256);
llum_actual = (1.2 / 1023) * analogValueLL;
}
}
}

```

```

while(client.connected() && !client.available()) delay(1);
while (client.available()) {
    char c = client.read();
    valors_web= valors_web + c;
}
int responseBodyIndex = valors_web.lastIndexOf('\n');
if (responseBodyIndex > 0) {
    valors_web = valors_web.substring(responseBodyIndex);
    //Serial.println(valors_web);
    String temperatura = getValue(valors_web, '#', 0);
    String llum = getValue(valors_web, '#', 1);
    temperatura_objectiu = temperatura.toFloat();
    llum_objectiu = llum.toFloat();
    String aux_ventilador = getValue(valors_web, '#', 2);
    ordre_ventilador = aux_ventilador.toInt();
    String aux_caldera = getValue(valors_web, '#', 3);
    ordre_caldera = aux_caldera.toInt();
    String aux_llum = getValue(valors_web, '#', 4);
}

```

```

ordre_llum = aux_llum.toInt();
Serial.println((String)llum_actual+" < "+(String)llum_objectiu);
if (ordre_llum==1) {
  llums = 1;
  setRemoteState(0xBF, 0x99, 0x72, 0x05);
  digitalWrite (LEDLLUM, HIGH);
  digitalWrite (LEDAUTOLLUM, HIGH);
}
if (ordre_llum==0) {
  llums = 1;
  setRemoteState(0xBF, 0x99, 0x72, 0x04);
  digitalWrite (LEDLLUM, LOW);
  digitalWrite (LEDAUTOLLUM, HIGH);
}
if (ordre_llum==3){
  llums = 0;
  digitalWrite (LEDAUTOLLUM, LOW);
}
if (ordre_ventilador == 1){
  ventilador = 1;
  setRemoteState(0xA5, 0x9D, 0x8B, 0x05);
  digitalWrite (LEDVENTILADOR, HIGH);
  digitalWrite (LEDAUTOVENTILADOR, HIGH);
}
if (ordre_ventilador == 0){
  ventilador = 1;
  setRemoteState(0xA5, 0x9D, 0x8B, 0x04);
  digitalWrite (LEDVENTILADOR, LOW);
  digitalWrite (LEDAUTOVENTILADOR, HIGH);
}
if (ordre_ventilador == 3){
  ventilador = 0;
  digitalWrite (LEDAUTOVENTILADOR, LOW);
}
if (ordre_caldera==1){
  caldera = 1;
  setRemoteState(0xC0, 0x4C, 0x95, 0x05);
  digitalWrite (LEDCALDERA, HIGH);
  digitalWrite (LEDAUTOCALDERA, HIGH);
}
if (ordre_caldera==0){
  caldera = 1;
  setRemoteState(0xC0, 0x4C, 0x95, 0x04);
  digitalWrite (LEDCALDERA, LOW);
  digitalWrite (LEDAUTOCALDERA, HIGH);
}

```

```

    if (ordre_caldera==3){
        caldera = 0;
        digitalWrite (LEDAUTOCALDERA, LOW);
    }
    if ((llum_actual2 < llum_objectiu) && (llums==0)){
        setRemoteState(0xBF, 0x99, 0x72, 0x05);
        digitalWrite (LEDLLUM, HIGH);
    }
    if ((llum_actual2 >= llum_objectiu) && (llums==0)){
        setRemoteState(0xBF, 0x99, 0x72, 0x04);
        digitalWrite (LEDLLUM, LOW);
    }
    if(temperatura_actual2 < 100){
        if (((temperatura_actual2 > 30) && (temperatura_objectiu > 12)) &&
(ventilador==0)){
            setRemoteState(0xA5, 0x9D, 0x8B, 0x05);
            digitalWrite (LEDVENTILADOR, HIGH);
        }
    }
    if (((temperatura_actual2 < 30) || (temperatura_objectiu == 12)) && (ventilador
== 0)){
        setRemoteState(0xA5, 0x9D, 0x8B, 0x04);
        digitalWrite (LEDVENTILADOR, LOW);
    }
    if ((temperatura_actual2 < temperatura_objectiu) && (caldera == 0)){
        setRemoteState(0xC0, 0x4C, 0x95, 0x05);
        digitalWrite (LEDCALDERA, HIGH);
    }
    if ((temperatura_actual2 >= temperatura_objectiu) && (caldera == 0)){
        setRemoteState(0xC0, 0x4C, 0x95, 0x04);
        digitalWrite (LEDCALDERA, LOW);
    }
    wdt_reset();
}
if (!client.connected() && lastConnected) {
    Serial.println();
    Serial.println("disconnecting.");
    client.stop();
}
if(!client.connected() && (millis() - lastConnectionTime > postingInterval)) {
    httpRequest();
    valors_web = "";
}
lastConnected = client.connected();
valors_web = "";

```

```
if (analogValueBV < 2.2){  
    setPantalla2(b0, b1, b2, b3);  
}  
if ((b0==64) && (b1==165) && (b2==157) && (b3==139)){  
    setPantalla(1);  
}  
}
```

3.3 Disseny de les plaques de circuit imprès

S'ha dissenyat el següent esquema elèctric per a la placa:

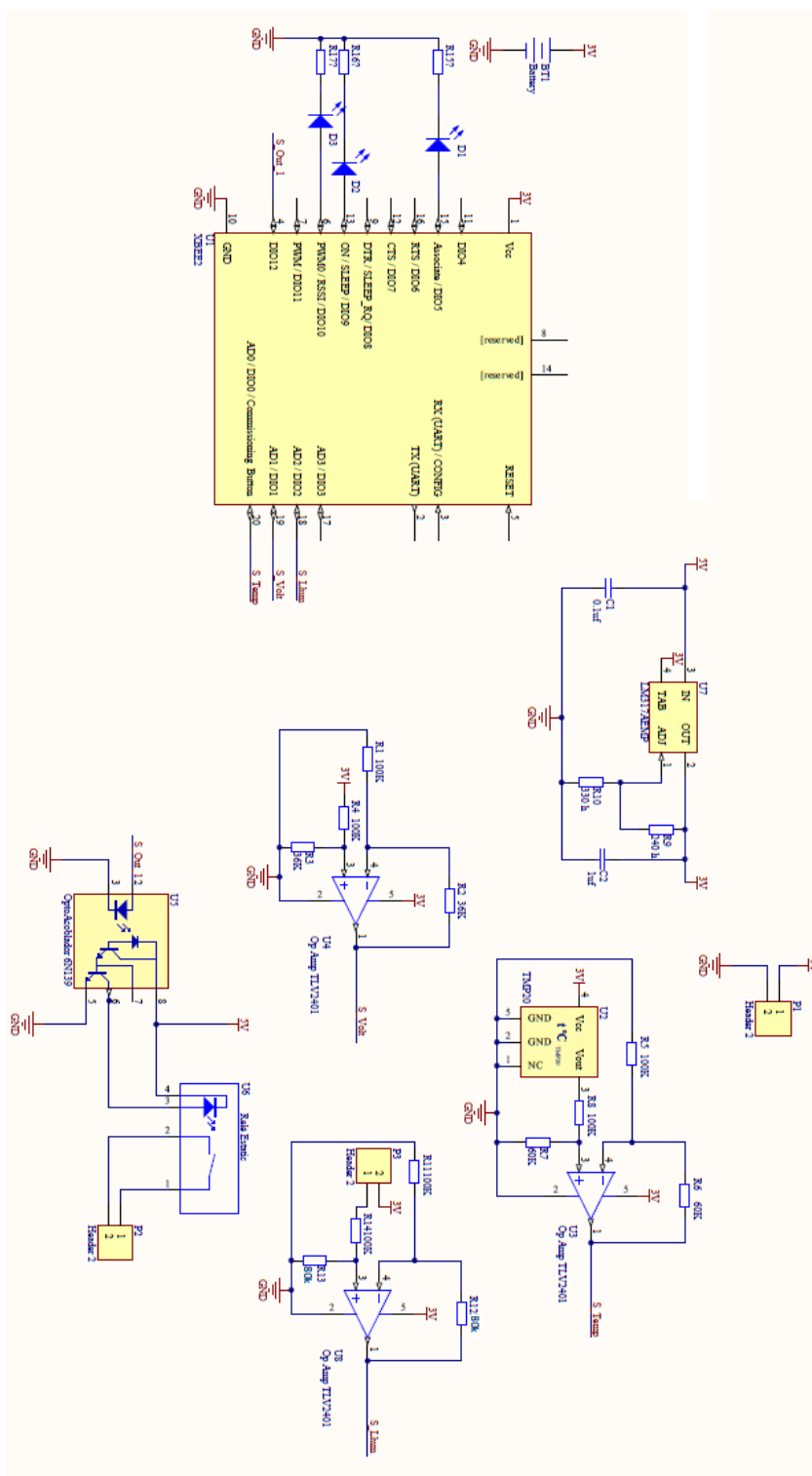


Figura 3-31. Esquema elèctric de la placa de circuit imprès

Disseny de la placa:

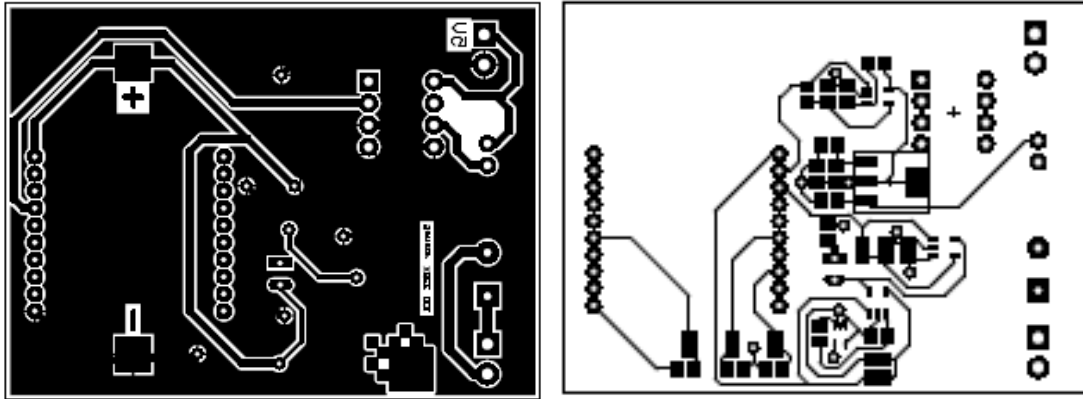


Figura 3-32. Disseny placa de circuit imprès

Esquema elèctric per al mòdul coordinador:

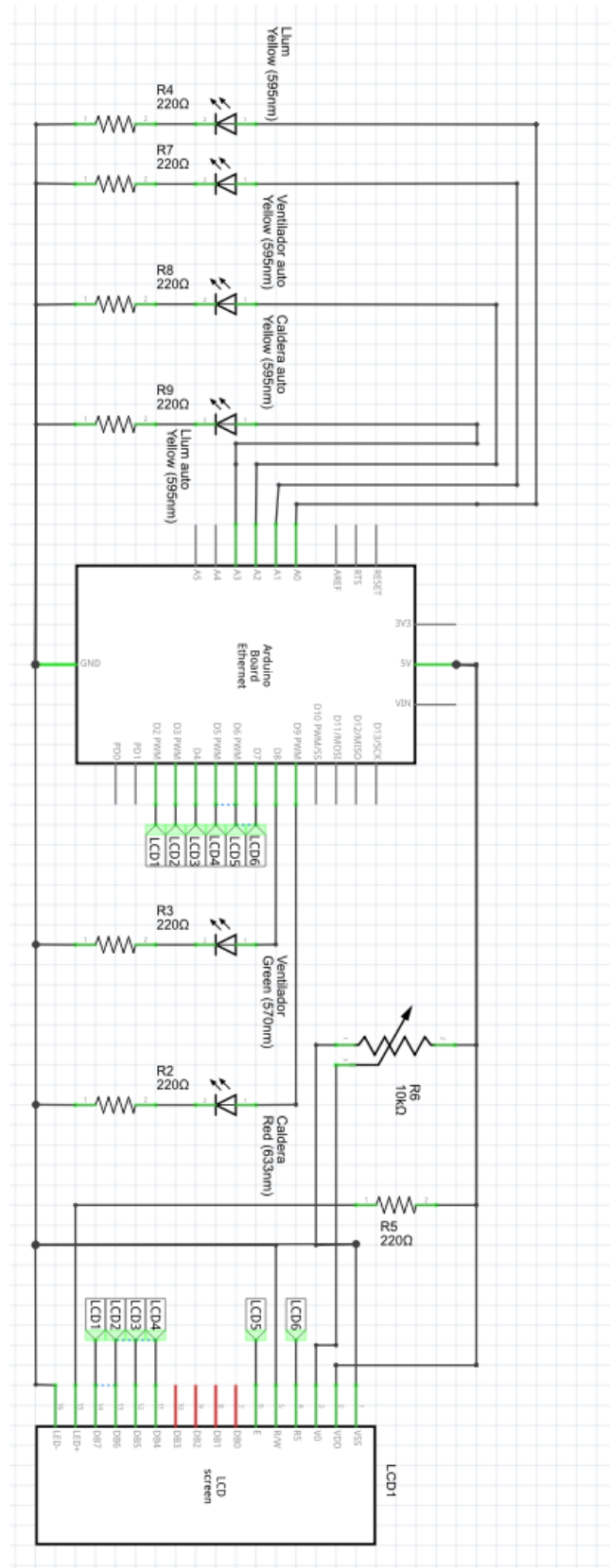


Figura 3-33. Esquema elèctric del mòdul coordinador

3.4 Disseny base de dades i control remot

3.4.1 Disseny base de dades

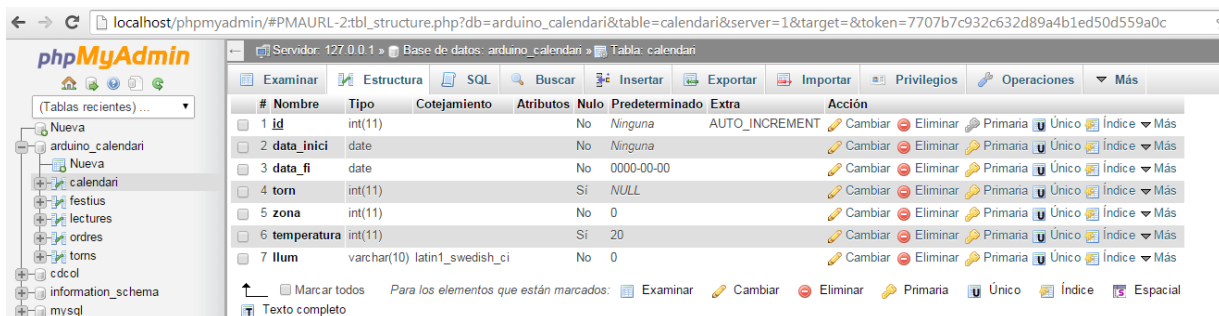


Figura 3-34. Captura pantalla estructura taula Calendari

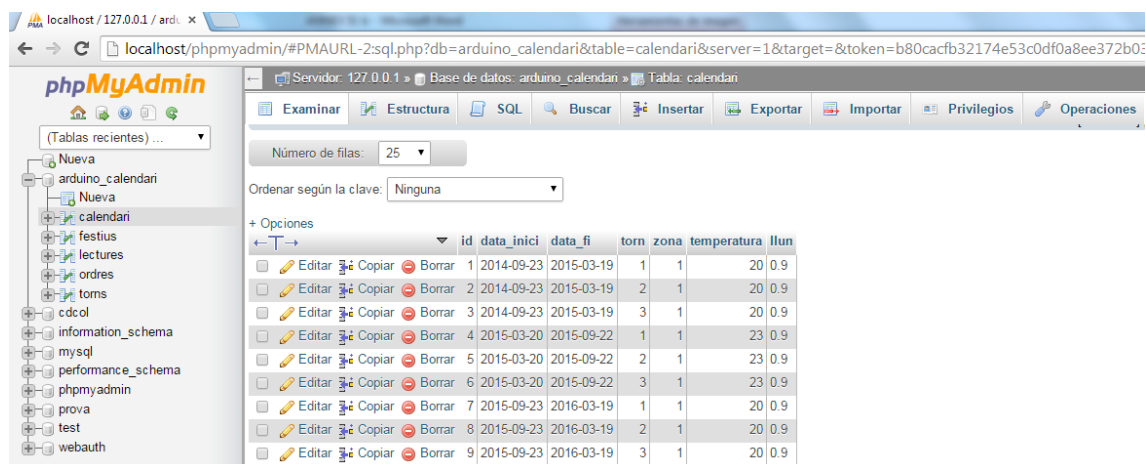


Figura 3-35. Captura pantalla taula Calendari

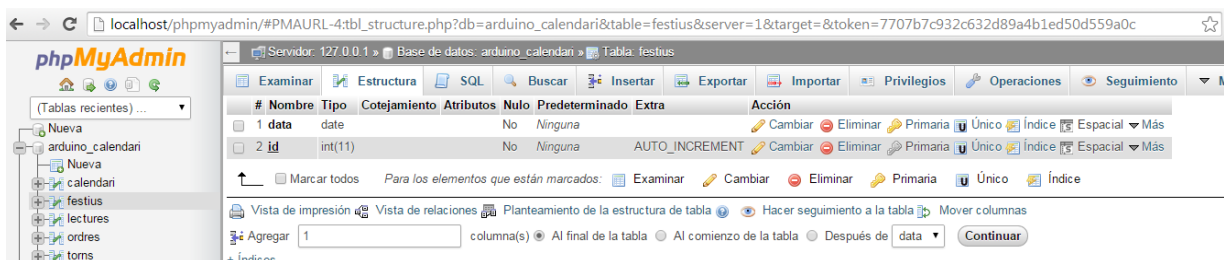


Figura 3-36. Captura pantalla estructura taula Festius

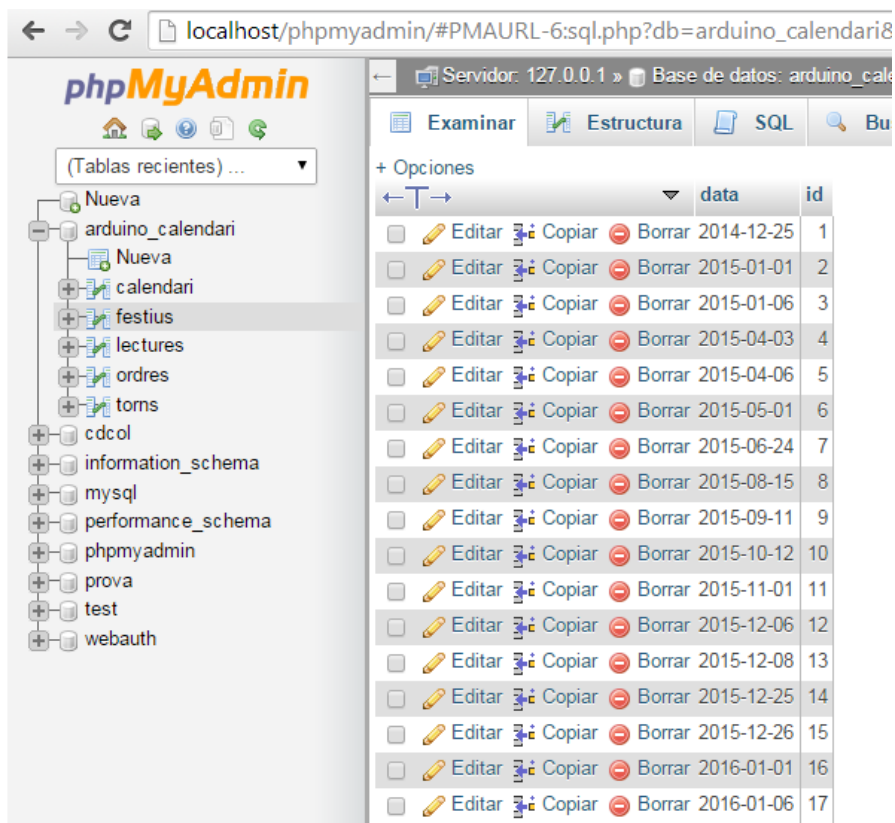


Figura 3-37. Captura pantalla taula Festius

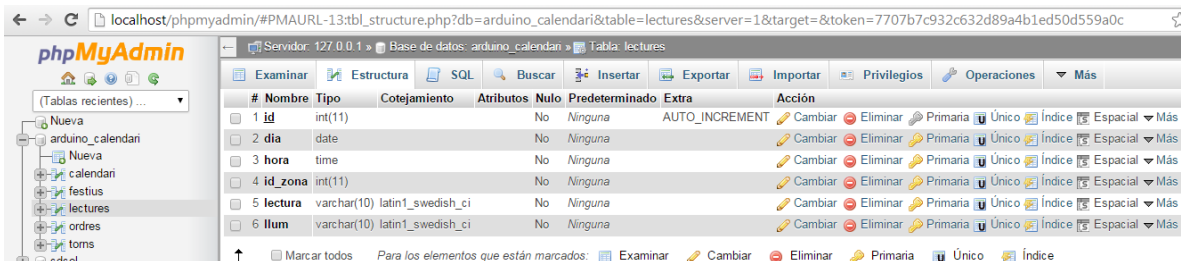


Figura 3-38. Captura pantalla estructura taula Lectures

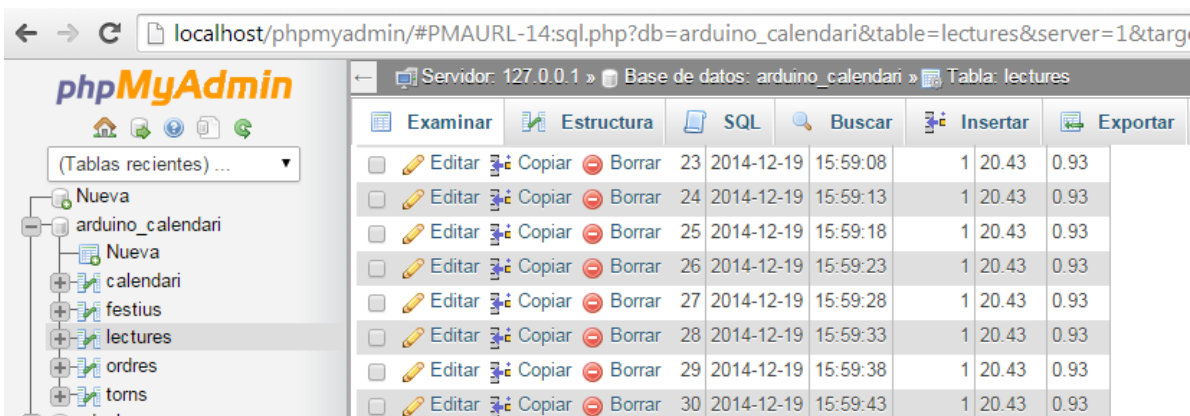


Figura 3-39. Captura pantalla taula Lectures

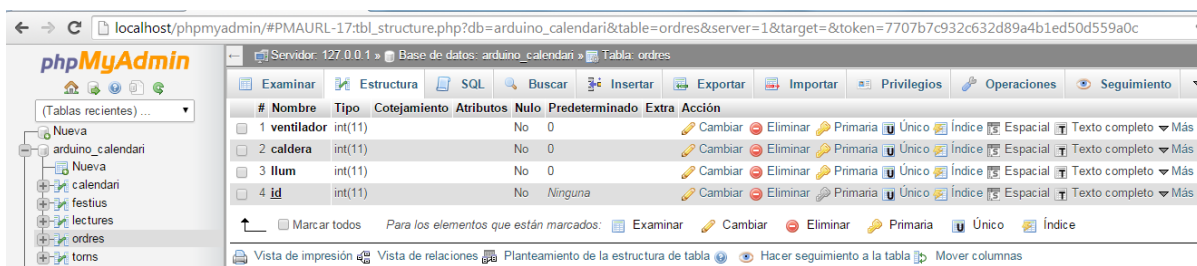


Figura 3-40. Captura pantalla estructura taula Ordres

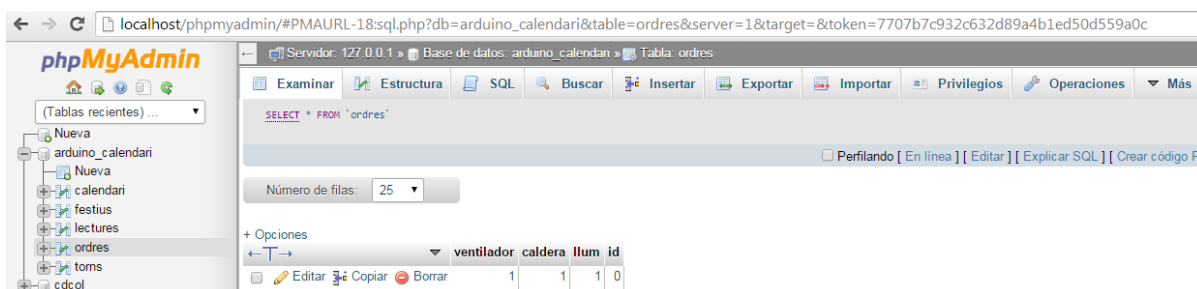


Figura 3-41. Captura pantalla taula Ordres

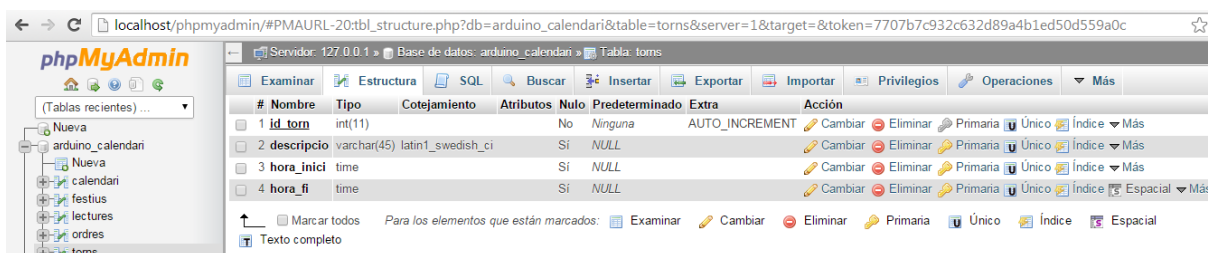


Figura 3-42. Captura pantalla estructura taula Torns

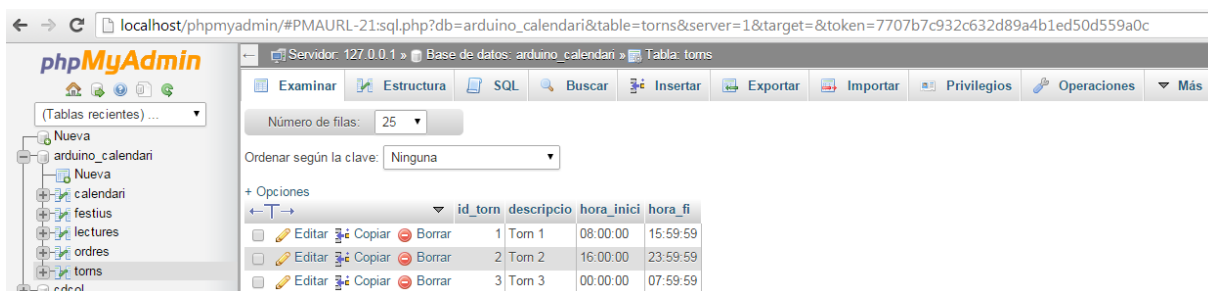


Figura 3-43. Captura pantalla taula Torns

3.4.2 connexio.php

```
<?php

$conn = mysql_connect ("localhost", "root", "");
mysql_select_db("arduino_calendari", $conn);

?>
```

3.4.3 calendari.php

```
<?php

include "connexio.php";

$hora_actual = date("H:i:s");
$dia_actual = date("Y-m-d");
$dia_setmana = date("D");
$valors = "";

if (($dia_setmana != "Sat") && ($dia_setmana != "Sun"))
{
    $rs = mysql_query("SELECT COUNT(*) FROM festius WHERE data =
    ".$dia_actual."");
    if (mysql_result($rs, 0, 0) == 0)
    {
        $rs = mysql_query("SELECT id_torn FROM torns WHERE hora_inici
        <= ".$hora_actual." AND hora_fi >= ".$hora_actual."", $conn);
        $torn = mysql_result($rs, 0, 0);

        $rs = mysql_query("SELECT temperatura,llum FROM calendari
        WHERE torn = ".$torn." AND zona = ".$_GET["zona"]." AND
        data_inici <= ".$dia_actual." AND data_fi >= ".$dia_actual."",
        $conn);
        $temperatura = mysql_result($rs, 0, 0);
        $llum = mysql_result($rs, 0, 1);

        $valors = $temperatura."#".$llum."#";
    }
    else
        $valors = "12#0#";
}
else
    $valors = "12#0#";
```

```
$rs = mysql_query("SELECT ventilador,caldera,llum FROM ordres", $conn);

$ordre_ventilador = mysql_result($rs, 0, 0);
$ordre_caldera = mysql_result($rs, 0, 1);
$ordre_llum = mysql_result($rs, 0, 2);
$valors .= $ordre_ventilador."#".$ordre_caldera."#".$ordre_llum;

echo $valors;

mysql_query("INSERT INTO lectures(dia, hora, id_zona, lectura, llum) VALUES
('.$dia_actual.','.$hora_actual.', ".$_GET["zona"].", ".$_GET["lectura"].",
'.$_GET["llum"].'");

?>
```

3.5 Disseny de les caixes

3.5.1 Plànol tapa inferior comú

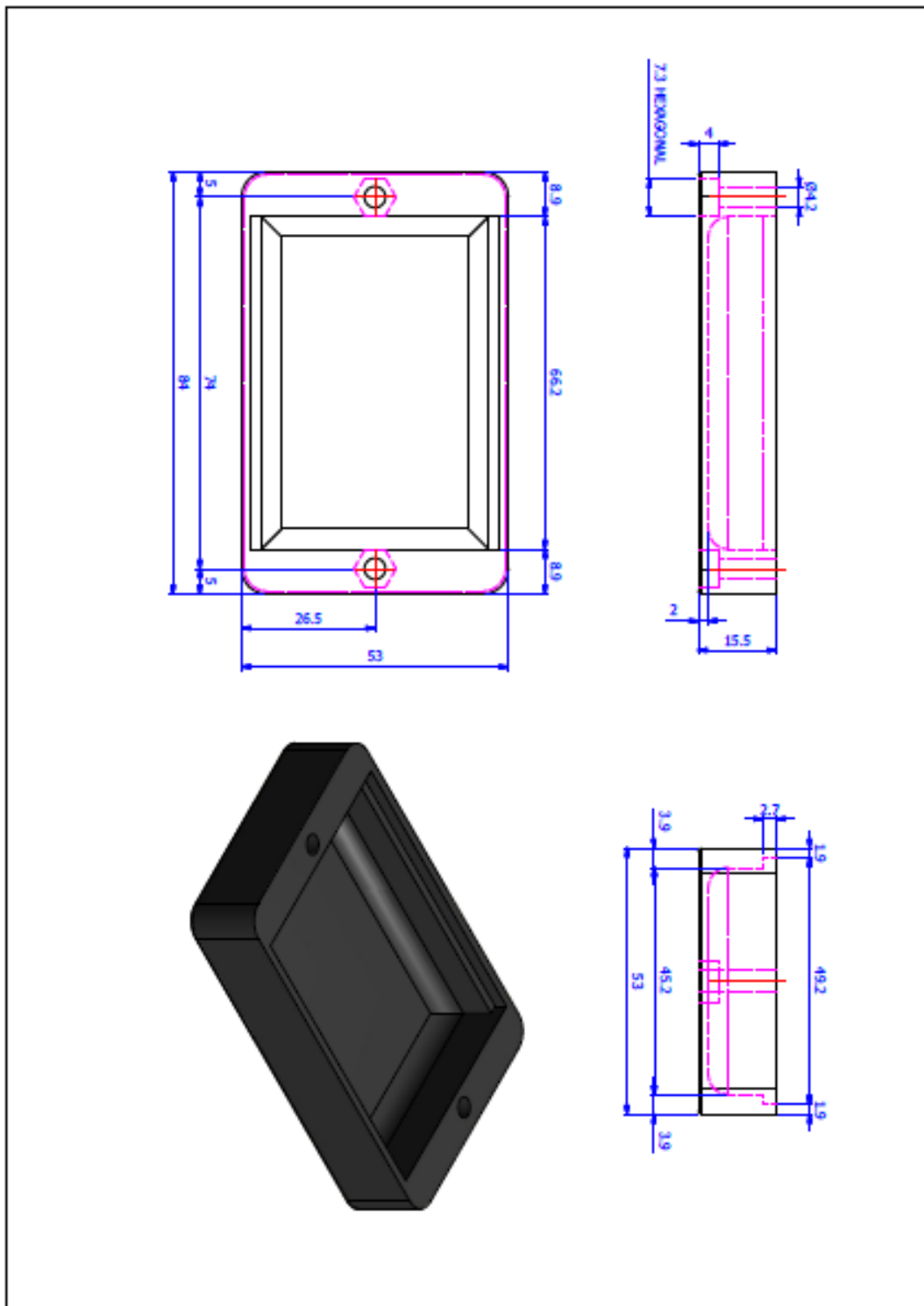


Figura 3-44. Plànol tapa inferior comú mòdul

3.5.2 Plànol tapa superior mòdul sensor

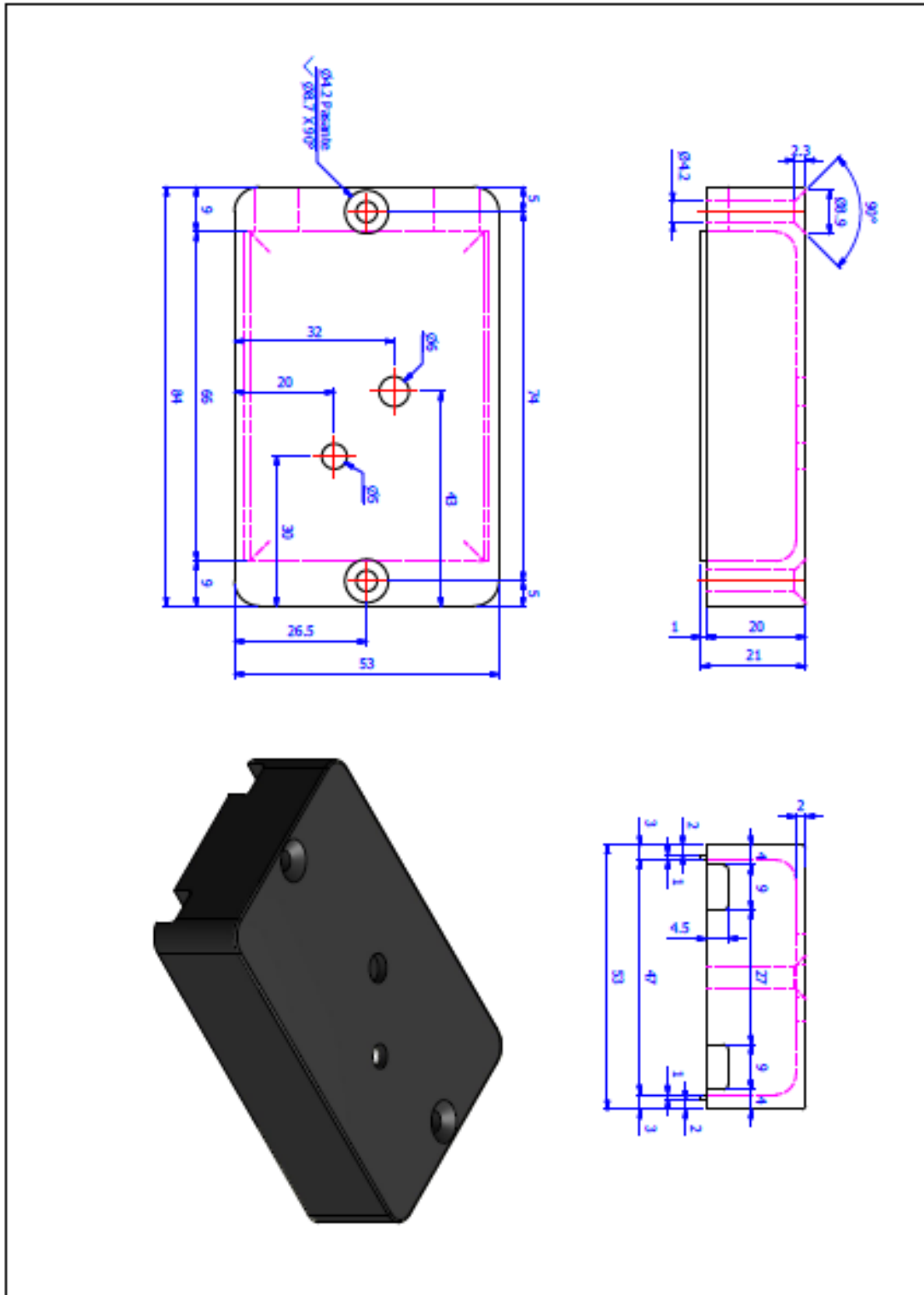


Figura 3-45. Plànol tapa superior mòdul sensor

3.5.3 Plànol tapa superior mòdul relé

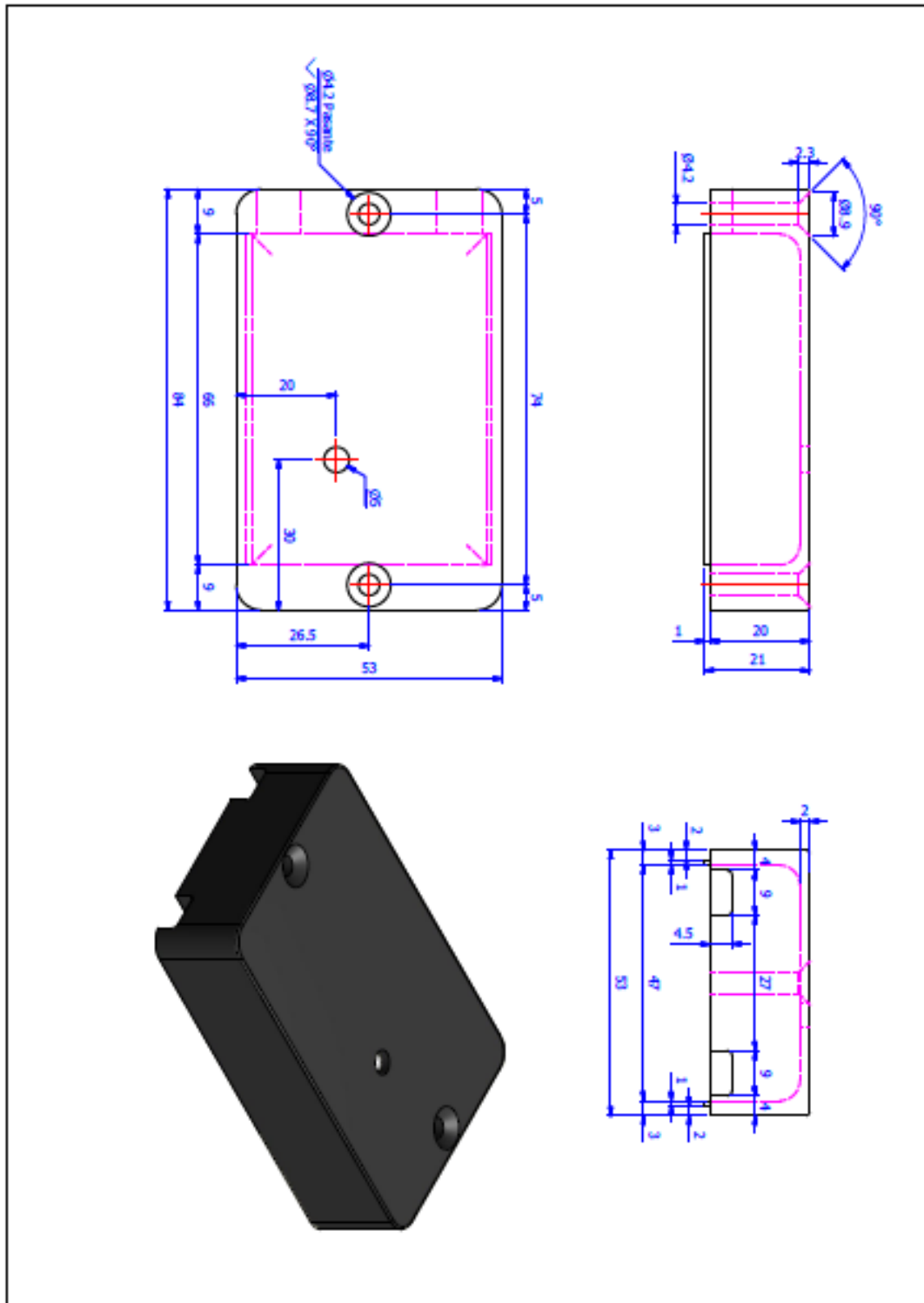


Figura 3-46. Plànol tapa superior mòdul relé

3.5.4 Plànol tapa inferior mòdul coordinador

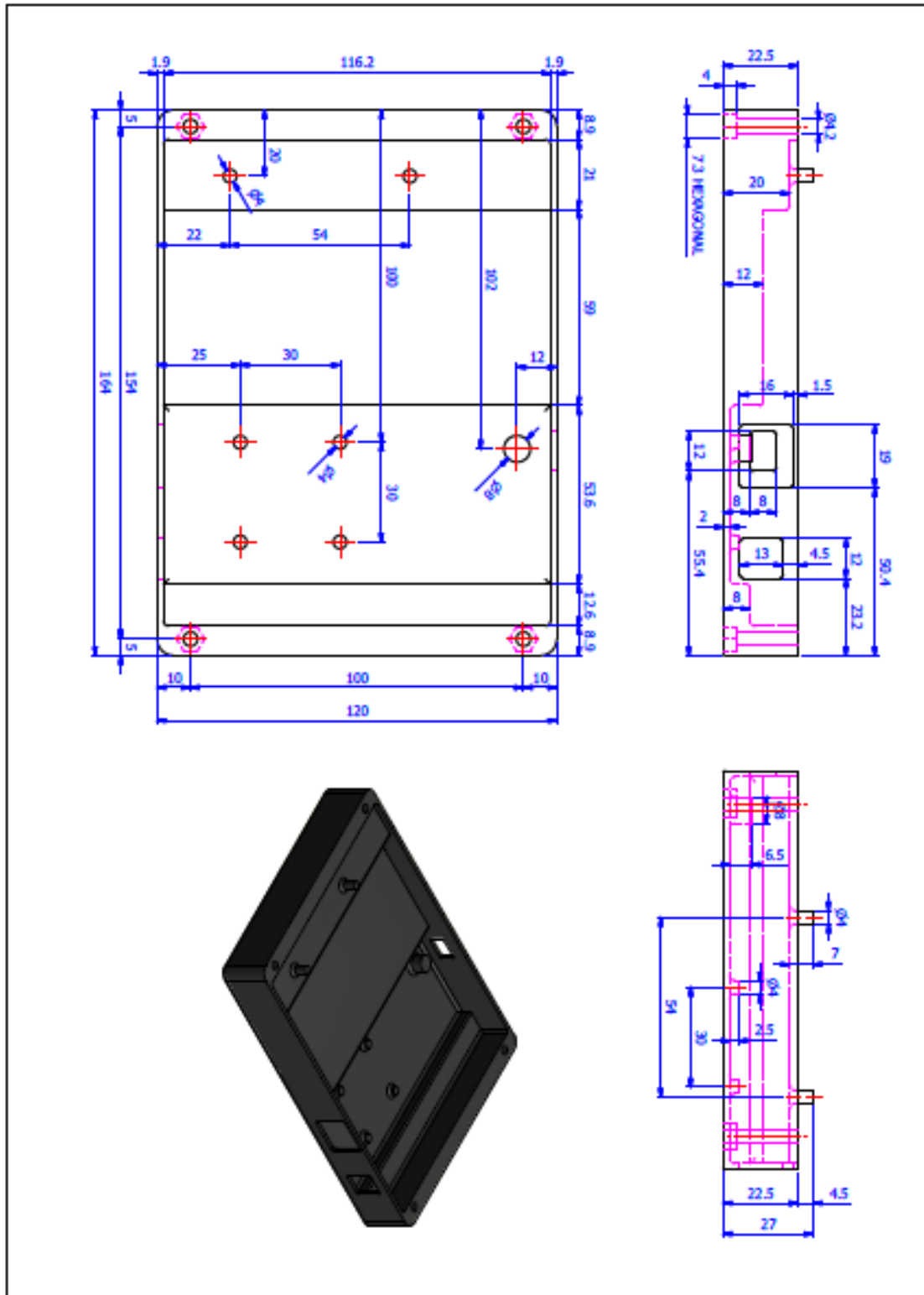


Figura 3-47. Plànol tapa inferior mòdul coordinador

3.5.5 Plànol tapa superior mòdul coordinador

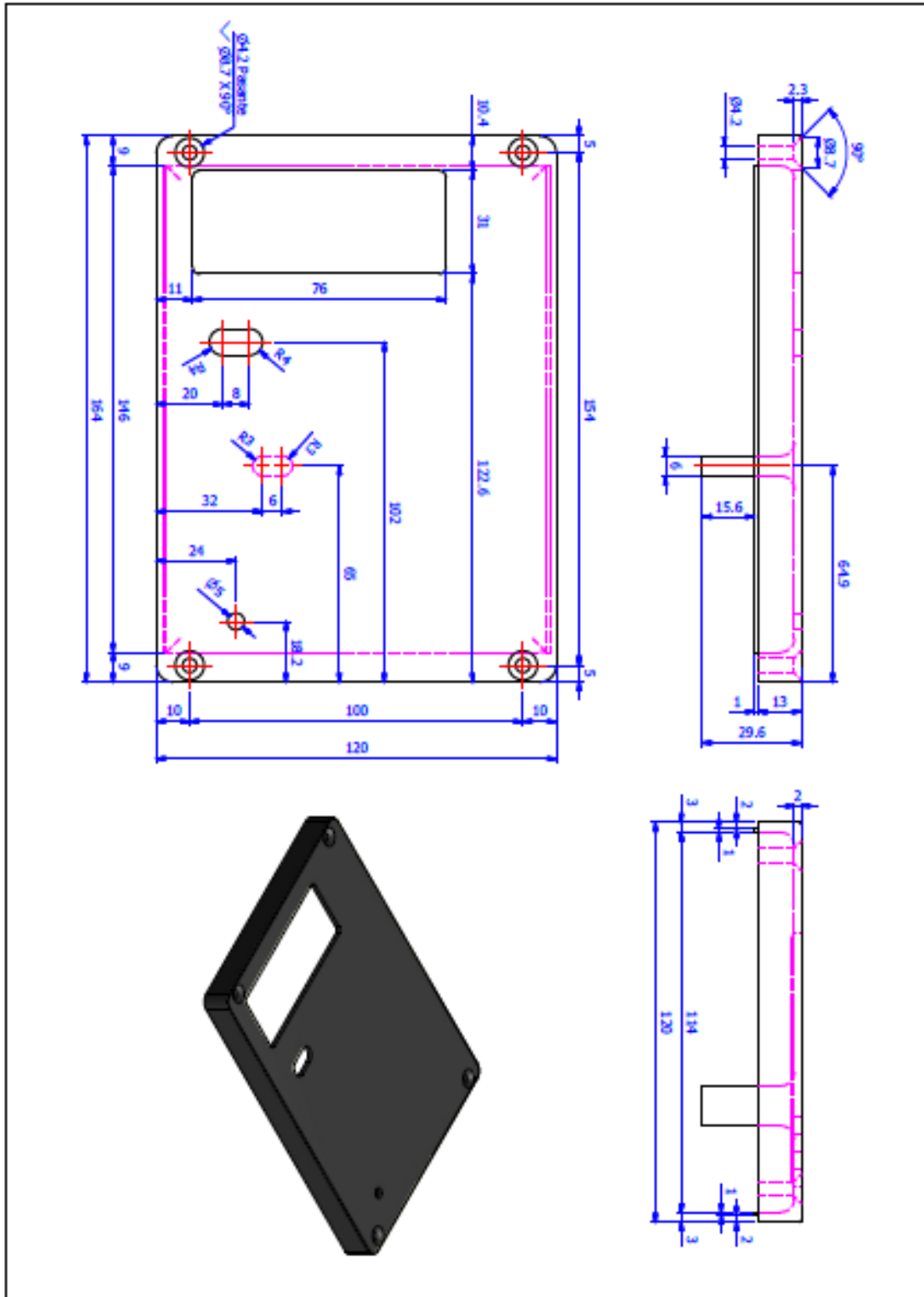


Figura 3-48. Plànol tapa superior mòdul coordinador

3.5.6 Tenyit de les caixes

Per tal d'obtenir un millor acabat s'ha decidit tenyir les peces mitjançant un colorant dispers en pols de color gris Foron de la casa Clariant.

Les proporcions que s'han utilitzat són:

- 1 litre d'aigua destil·lada i un gram de colorant en pols per cada 100 gr. de material a tenyir.

Com que les quatre caixes tenen un pes de 230 gr., s'han utilitzat dos litres d'aigua destil·lada i 2 gr. de colorant en pols.



Figura 3-49. Material necessari tenyit peces

El procediment que s'ha seguit és:

- Remullar en aigua destil·lada les caixes a tenyir durant 30 minuts.
- Posar a escalfar dos litres d'aigua destil·lada en una olla d'acer inoxidable (no pot ser d'alumini).

- Quan l'aigua està a punt d'arrancar a bullir, agafar un got d'aigua de l'olla i dissoldre el colorant.



Figura 3-50. Dissolució colorant

- Tirar el colorant dissolt a l'olla i remenar fins que l'aigua arranqui a bullir.
- Tirar les peces a dins de l'olla i deixar-les-hi durant 20 minuts.



Figura 3-51. Ebullició de les peces

- Treure les peces i netejar-les amb abundant aigua.
- Deixar-les assecar.



Figura 3-52. Assecat de les peces

Cal tenir en compte que tota l'aigua que pugui contenir colorant s'ha de guardar i portar a la deixalleria.