Escola Politècnica Superior de Castelldefels
UNIVERSITAT POLITÈCNICA DE CATALUNYA

TREBALL DE FI DE CARRERA

TÍTOL DEL TFC: Monitoratge ambiental d'entorns aïllats utilitzant xarxes de sensors

TITULACIÓ: Enginyeria Tècnica de Telecomunicació, especialitat Sistemes de Telecomunicació

AUTOR: Àngel Piqueras Moya

DIRECTORS: Lluís Casals Ibáñez
Rafael Vidal Ferré

DATA: 14 d'Octubre de 2010

Títol: Monitoratge ambiental d'entorns aïllats utilitzant xarxes de sensors

Autor: Àngel Piqueras Moya

Director: Lluís Casals Ibáñez; Rafael Vidal Ferré

Data: 14 d'Octubre de 2010

Resum

Aquest projecte té com a idea poder fer el monitoratge d'entorns aïllats o de difícil accés.

Aquest monitoratge el podem dividir en dues fases. La primera col·locar una xarxa de sensors realitzant tasques d'obtenció de dades en aquests entorns aïllats i la segona el disseny d'una eina capaç de recol·lectar aquestes dades.

Es vol realitzar una eina econòmica i que sigui capaç de superar les restriccions del canal de comunicacions, ja que les zones que es volen monitorar poden no tenir cobertura mòbil.

És per això que s'implementa una xarxa de sensors oportunística. Mitjançant el pas periòdic de l'usuari per la zona que conté els nodes sensors es poden recollir les dades que després es podran visualitzar a un ordinador.

Per al disseny de la xarxa s'han utilitzat els nodes CC2430N, realitzats pel Grup de Recerca de Xarxes sense Fils (WNG, Wireless Networks Group) de la Universitat Politècnica de Catalunya (UPC). Aquests nodes estan basats en el xip CC2430 de Texas Instruments i és per això que s'utilitza la pila de protocols TIMAC proporcionada pel mateix fabricant per programar els nodes.

L'eina recol·lectora de dades està constituïda per un node de les mateixes característiques que els que formen la xarxa, i per un circuït d'adaptació que permet, mitjançant el port sèrie, transmetre les dades al PC.

Un cop processades les dades al ordinador es poden visualitzar en una gràfica dissenyada amb el software que proporciona amCharts.

Title: Environmental monitoring of isolated zones using sensor networks

Author: Àngel Piqueras Moya

Director: Lluís Casals Ibáñez; Rafael Vidal Ferré

Date: October 14th, 2010

Overview

This project is based in the idea of do the monitoring of isolated zones using sensor networks.

This monitoring can be divided into two phases. The first is place sensor networks doing data collection tasks in these isolated zones. The second is design a tool capable of collecting this data.

The objective is to make an economic tool that is capable of overcoming the restrictions of the communication channel. That's because the zones that want to be monitored may not have mobile coverage.

That's why we implement an opportunistic sensor network. With the periodical pass of the user by the zones that contain the sensor nodes it's possible to collect the data and visualize it in a computer.

For the design of the network has been used CC2430N nodes built by Grup de Recerca de Xarxes sense Fils (WNG, Wireless Networks Group) of the UPC university. These nodes are based on CC2430 chip of Texas Instruments.

The collector tool is a CC2430N too. It has also been built an adaptation circuit that allows sending the data to a computer with the serial port.

After processing the data in the computer it can be visualized in a chart designed by the software provided by amCharts.

ÍNDEX

CAPÍTOL 1. INTRODUCCIÓ 9

1.1 Escenari 10

1.2 Esquema 11

1.3 Objectius 12

CAPÍTOL 2. ESTÀNDARD IEEE 802.15.4 13

2.1 Breu descripció 13

2.2 Dispositius i topologies 14

2.3 Capa física 15

2.4 Capa de control d'accés al medi 16

2.5 Funcionament 17

2.6 Mecanismes de robustesa 17

CAPÍTOL 3. HARDWARE 19

3.1 Xip CC2430 de Texas Instruments 19
3.1.1 Característiques generals 19
3.1.2 Especificacions del CC2430 20

3.2 Kit de desenvolupament 22
3.2.1 CC2430EM 22
3.2.2 SmartRF04EB 23
3.2.3 SOC_BB 24
3.2.4 SOC_DEM 25

3.3 CC2430N i connector TFSD de SAMTEC 25

CAPITOL 4. ENTORN DE PROGRAMACIÓ 27

4.1 IAR Embeeded WorkBench 27

4.2 TIMAC 29

4.3 Precompilador 30

CAPÍTOL 5. CONSUM 31

5.1 Modes d'energia 31

5.2 Consums del microcontrolador 31

5.3 Dispositiu treballant a baix consum 32

5.4 Tasques sense estalvi d'energia 33

CAPITOL 6. PARÀMETRES A MESURAR 35

6.1 Temperatura 35
6.1.1 Especificacions 35
6.1.2 Paràmetres 36
6.1.3 Càlculs prèvis 36
6.1.4 Calibració 37
6.1.5 Codi per configurar el sensor de temperatura intern 38

6.2 Humitat 38
6.2.1 Especificacions 38
6.2.2 Connexió de sortida del sensor 39
6.2.3 Instal·lació del sensor 39
6.2.4 Recol·lecció de les dades 40

6.3 Nivell de Bateria 41
6.3.1 Obtenció del nivell de bateria 41

CAPITOL 7. TRACTAMENT DE LES DADES I PROTOCOL D'ENVIAMENT 43

7.1 Tractament de les dades al node sensor 43

7.2 Comunicació entre nodes. Protocol d'enviament 45

7.3 Tractament de les dades al node gateway 47
7.3.1 Configuració del UART 50

7.4 Circuit d'adaptació per transmetre les dades al PC 50

7.5 Recepció de les dades al PC i creació de les gràfiques 51

CAPÍTOL 8. CONCLUSIONS I LINIES FUTURES 55

8.1 Conclusions 55

8.2 Línies futures 55

BIBLIOGRAFIA 57

CAPÍTOL 1. INTRODUCCIÓ

Entenem com xarxa de sensors oportunística aquella xarxa de sensors on no hi ha sempre una visió, un camí, en definitiva, una comunicació establerta entre emissor i receptor. Cal dir que solen aplicar-se a entorns amb un canal de comunicació inestable i és per això que els nodes d'aquestes xarxes són capaços de guardar les dades a enviar fins que trobin un mitjà per fer-ho.

A l'actualitat hi ha una forta investigació i constant evolució en el món de les xarxes de sensors sense fils. Al mercat trobem dispositius de mida reduïda, que tenen un baix consum d'energia i que a més d'això el seu cost és assequible. És per tot això que és una forta motivació el realitzar aquest treball final de carrera (TFC) per tal d'implementar un model de comunicació amb forta embranzida i acceptació.

Més concretament, el que es desitja en aquest projecte, és poder monitorar una zona aïllada mitjançant uns nodes sensors. Com a zona aïllada podem pensar per exemple en zones de muntanya o en zones de conreu. En les zones de muntanya el forestal seria l'encarregat de recollir les dades que poden servir per la previsió d'incendis o ara bé, en les zones de conreu, el pagès seria capaç de recol·lectar unes dades que li servirien per extreure el màxim rendiment dels seus cultius.

Els sensors dissenyats en aquest TFC seran capaços de mesurar la temperatura i humitat de l'ambient, i la bateria que els hi queda. Aquestes dades es van emmagatzemant fins que es troba un connexió que fa possible la transmissió. L'eina recol·lectora, al igual que els sensors de la xarxa entre sí, es comuniquen mitjançant l'estàndard IEEE 802.15.4.

És molt important el consum dels nodes que formen la xarxa de sensat. L'estàndard de comunicació emprat ha estat creat, entre d'altres coses, per aquest motiu, ser capaç de realitzar una transmissió de dades sense tenir un alt consum. En la part de disseny dels nodes també s'ha tingut en compte aquest factor i per això s'han dissenyat per ser capaços d'estalviar el màxim d'energia possible i així allargar el seu temps de vida.

Les dades obtingudes pel seguit de nodes sensors es recol·lecten mitjançant un node del mateix tipus al que anomenarem node gateway. Aquest node s'encarrega de demanar les dades als nodes sensors, mitjançant la interfície ràdio les rep, i per últim les envia a un ordinador mitjançant el port sèrie.

Un cop es tenen les dades al ordinador es realitzen les gràfiques que posteriorment seran enviades a un servidor, així es permet l'accés a qualsevol usuari

Tot el citat anteriorment es programarà sobre el hardware CC2430N realitzat pel Grup de Recerca de Xarxes sense Fils (WNG, Wireless Networks Group) de la Universitat Politècnica de Catalunya (UPC).

1.1 Escenari

A la Figura 1.1 s'observa l'escenari que s'ha realitzat en aquest TFC:

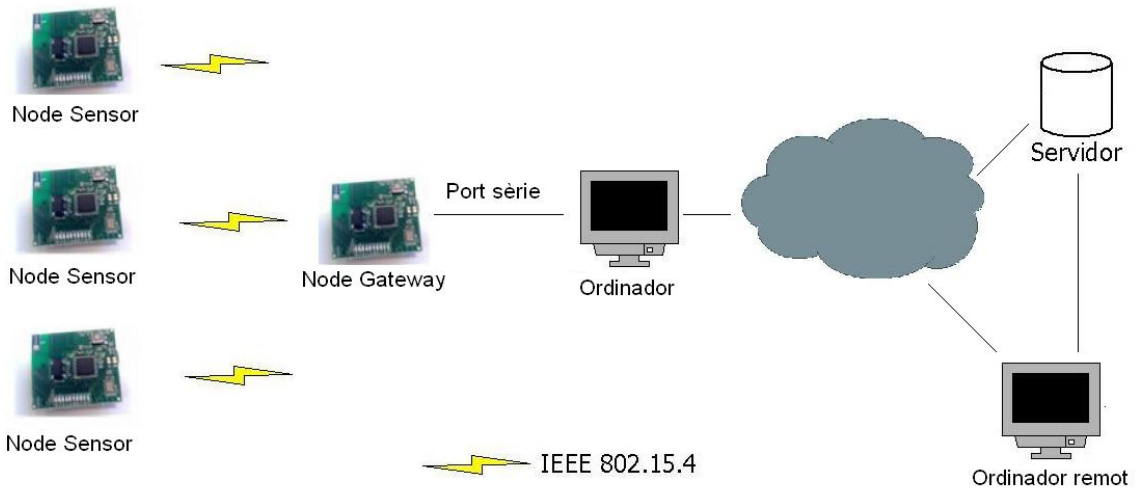


Figura 1.1 Escenari

Amb aquest escenari com a referència cal dir que només s'ha dissenyat el codi que permet la comunicació del node gateway amb un únic node sensor. Es planteja l'escenari amb els nodes sensors molt separats entre ells, col·locats en zones aïllades, sense visibilitat amb cap altre sensor.

En aquest TFC no es contempla que en una zona de cobertura hi hagi més d'un sensor, és a dir, el node sensor no té cap altre sensor del qual rebre dades, i el node gateway, sempre que estableixi comunicació, només ho farà amb un sol node sensor.

A la Figura 1.1 s'observa, d'esquerra a dreta:

- Xarxa de nodes sensors: Nodes encarregats d'obtenir i emmagatzemar les dades de temperatura i humitat ambient així com la bateria que els hi queda. L'obtenció d'aquestes dades la fan periòdicament amb un interval d'un minut entre mesura i mesura.

- Node gateway: Node encarregat de rebre, mitjançant l'estàndard 802.15.4, les dades obtingudes per la xarxa de nodes sensors

- Enviament de les dades a un ordinador: El node gateway, mitjançant el port sèrie, envia tot el que rep dels nodes sensors a un ordinador. En aquest ordinador és on es generen les gràfiques perquè l'usuari sigui capaç de visualitzar les dades obtingudes en el sensat.

- Distribució de les dades: Es pugen les gràfiques obtingudes a un servidor web per tal de que qualsevol usuari les pugui visualitzar.

Totes aquestes etapes es descriuran amb major detall en els capítols posteriors.

1.2 Esquema

En aquest apartat es mostra l'esquema de connexions entre els diferents membres que formen l'escenari. Seguint la Figura 1.1 es descriuran les connexions entre els diferents blocs d'esquerra a dreta, és a dir, començant per la xarxa de sensat fins arribar al ordinador. Cada bloc i les seves connexions són detallades en capítols posteriors.

El primer que es troba és el node sensor que porta connectat el sensor d'humitat extern:

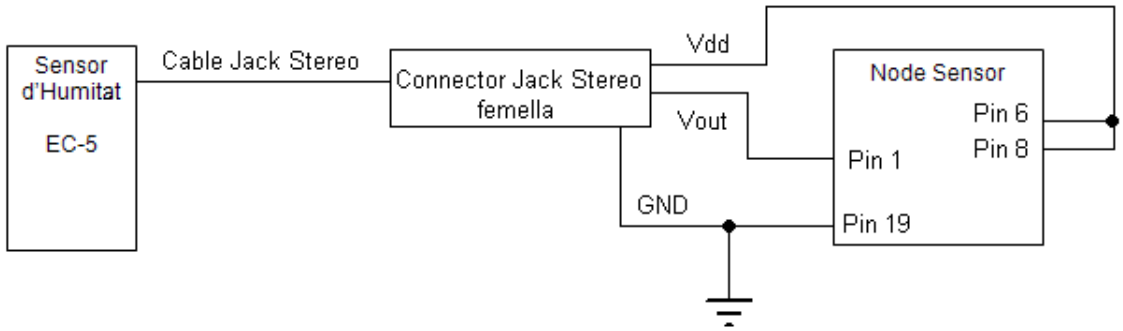


Figura 1.2 Connexionat entre els sensor d'humitat extern EC-5 i el node sensor

El següent pas és la connexió entre el node sensor i el node gateway. Com s'observa a la figura 1.1 es comuniquen mitjançant la interfície ràdio.

Per últim trobem el node gateway connectat a un ordinador mitjançant el port sèrie. L'esquema seria el següent:

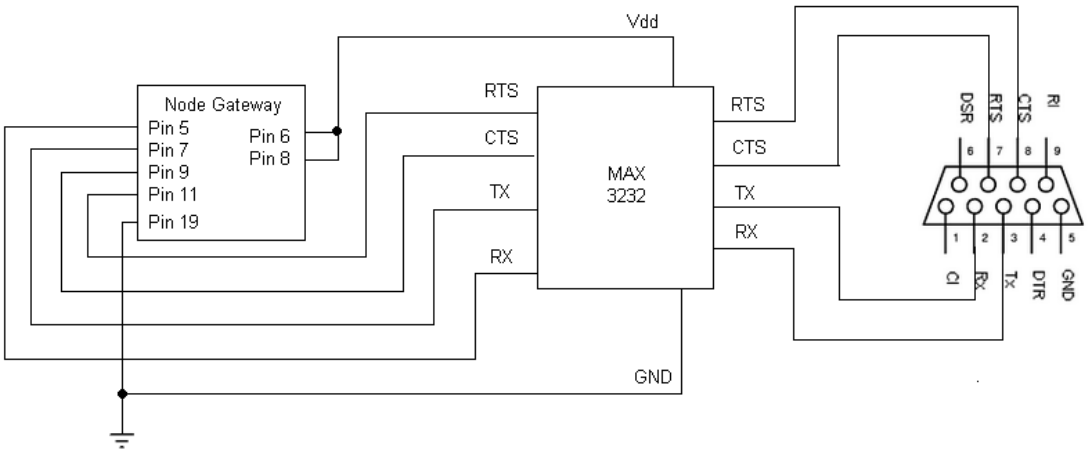


Figura 1.3 Esquema de connexions entre el node gateway i l'adaptador femella RS-232

El cable que connecta la connexió RS-232 femella que observem a la Figura 1.3 amb l'ordinador recol·lector de dades és directe.

1.3 Objectius

En aquest apartat es volen descriure els objectius que es volen assolir amb la realització d'aquest TFC:

- Realitzar una xarxa capaç de monitorar entorns aïllats que assoleixi tres aspectes fonamentals: baix cost, òptima transmissió, mínim consum i fàcil us per part de l'usuari.

- Autonomia tant dels nodes sensors com del node gateway. Tots han de funcionar mitjançant l'alimentació d'una pila i han de ser capaços de mesurar el nivell de bateria que el hi queda per tal saber quan necessiten un canvi.

- Capacitat per obtenir les dades de temperatura amb el sensor integrat al xip.

- Capacitat per obtenir les dades d'humitat mitjançant la utilització d'un sensor d'humitat extern.

- Utilitzar l'estàndard 802.15.4 per transmetre les dades entre els nodes.

- Capacitat d'optimitzar el consum dels nodes mitjançant els modes de treball que ens ofereixen els chips. El disseny dels nodes sensors anirà enfocat cap al mínim consum de recursos.

- Capacitat per transmetre les dades des de el node gateway a un ordinador mitjançant el port sèrie.

- Realitzar una aplicació senzilla al ordinador receptor que sigui capaç de processar les dades enviades pel node gateway.

- Mostrar les dades al ordinador d'una forma gràfica i visual.

- Penjar les gràfiques finals a un servidor per tal de donar accés a qualsevol usuari des de un ordinador remot.

Aquesta memòria es dividirà en 8 capítols. Aquest primer és introductori, el segon ens parla del estàndard IEEE 802.15.4, el tercer del hardware, el quart de l'entorn del programació, el cinquè ens detalla el consum del hardware, el sisè capítol ens descriu els paràmetres a mesurar i el setè explica com es tracten i envien les dades. L'últim capítol el formen les conclusions i línies futures.

CAPÍTOL 2. ESTÀNDARD IEEE 802.15.4

En aquest capítol es descriurà l'estàndard que es fa servir en la comunicació entre els nodes que formen la xarxa de sensors d'aquest TFC. Es farà una introducció sobre la historia i característiques de l'estàndard IEEE 802.15.4, es descriuran els dispositius i topologies i els mecanismes de robustesa.
En cada apartat es detallarà quina de les opcions que ens ofereix l'estàndard s'ha escollit per al disseny d'aquest TFC.

2.1 Breu descripció

L'estàndard IEEE 802.15.4[1] es va crear per ser utilitzat en xarxes de sensors. Ja es disposaven d'altres estàndards com el IEEE 802.11 (WIFI) o el IEEE 802.15.1 (Bluetooth), però aquests consumien massa energia i tenien massa ample de banda en relació al que necessita una xarxa de sensors. Finalment, l'estàndard definit, va quedar amb les següents característiques, que són del tot suficients per les comunicacions en una xarxa de sensors:

Taula 2.1. Característiques principals de l'estàndard IEEE 802.15.4

Estàndard	**Taxa de transferència**	**Consum**	**Avantatges**	**Aplicacions**
IEE 802.15.4	250 Kbps	1,8 mA transmetent i 5,1 µA en repòs	Gran durada de la bateria i baix cost	Sensors, aplicacions per control remot, etc...

Va ser a l'Octubre de 2003 quan es van definir les especificacions per l'estàndard 802.15.4. Es van definir la capa física i la capa d'accés al medi per xarxes LR-WPAN (*Low Rate Wireless Personal Area Network)* i després un conglomerat de companyies anomenat ZigBee Alliance, basant-se en les capes ja definides va completar l'estàndard definint les capes de xarxa, aplicació i seguretat.

El resultat és un estàndard que permet comunicacions de fins a 75 metres. És a dir, està pensat per a comunicacions de curt abast i de baix consum ja que, el que es pretén és que un sensor equipat amb aquesta tecnologia sigui capaç de mantenir-se operatiu durant com a mínim 6 mesos i com a màxim 2 anys únicament alimentant-se amb una pila de botó de 3.3 volts.

Per últim cal dir que es poden utilitzar diferents bandes de transmissió. La més utilitzada és la banda de 2,4 GHz a 250 Kbps ja que és la que es pot utilitzar en tot el món. També es pot transmetre a 40 Kbps en la banda de 868 MHz si ens trobem a Europa i si estem als Estats Units es pot fer servir la banda de 915 MHz a 20 Kbps.

Es pot afirmar que la millor tecnologia per comunicar els nodes a una xarxa de sensors és l'estàndard IEEE 802.15.4

2.2 Dispositius i topologies

Dintre de l'estàndard 802.15.4 podem definir dos tipus de dispositius diferenciats per la seva funció:

- FFD (*Full Function Device*) són dispositius capaços d'organitzar i coordinar l'accés al medi d'altres dispositius. Aquests dispositius són utilitzats gairebé sempre com a nodes centrals i solen necessitar un consum més alt d'energia.

- RFD (*Reduced Function Device*) són dispositius més simples que els anteriors i consumeixen un nivell més baix d'energia. La seva fabricació és més simple i són capaços de realitzar menys funcions que els FFD.

En aquest TFC s'utilitzen els dos tipus de dispositius:
Tots els nodes que formen la xarxa de sensat són RFD, la seva operativa té com objectiu l'estalvi d'energia. En canvi el node gateway és el que s'encarrega de coordinar la recol·lecció de dades, fa unes funcions diferents als nodes sensors i no es té tant en compte aquest estalvi d'energia sinó la correcta recepció de les dades, és per això que podem afirmar que el node gateway és un FFD.

Segons l'aplicació que es vol realitzar es fa servir una topologia o una altra, a continuació definirem els tres tipus que es troben:

- Topologia estrella: és la tipologia en la que trobem un node FFD central al qual es connecten altres nodes. El node central és el coordinador, el que controla les transmissions entre nodes i permet l'accés de nous dispositius a la xarxa.

Figura 2.1 Topologia estrella

- Topologia *peer to peer o mesh*: és la topologia en malla, en aquesta també trobem un node central però que no és tant rellevant com en la topologia estrella. En la topologia mesh qualsevol node es pot comunicar amb un altre si està en el mateix rang de cobertura o bé utilitzant altres nodes per arribar-hi. En aquesta topologia es pot donar el cas que hi hagi més d'un salt entre el node origen i el destí, per tant calen protocols d'encaminament.

Figura 2.2 Topologia peer to peer o mesh

En el disseny del nostre TFC s'utilitza una tipologia mesh.
El node gateway és el que fa la funció de node coordinador ja que es el que recull les mostres però no té tanta importància com té el coordinador de una xarxa amb tipologia estrella. Sempre que hi hagi un canal els nodes es podran comunicar entre ells. En aquest TFC la comunicació d'un node sensor només es possible amb el coordinador, ja que no s'ha contemplat que puguin haver 2 o més nodes sensors en el mateix rang de cobertura.

2.3 Capa física

La capa física (PHY) de l'estàndard IEEE 802.15.4 és l'encarregada de controlar el transceptor de radiofreqüència activant-lo i desactivant-lo quan cal, tria els canals dels 27 que disposa, controla el senyal a enviar i el consum energètic d'aquest, detecta la qualitat de l'enllaç i també detecta la portadora. La PHY pot operar en 3 rangs de freqüència, totes dintre de la banda ISM (*Industrial, Scientific and Medical*) i per totes elles s'utilitza DSSS (*Direct Sequence Spread Spectrum*).

En la taula 2.2 mostrada a continuació s'observen els diferents paràmetres que posseeix la capa física.

Taula 2.2. Característiques principals de la capa física segons la banda de freqüència.

Banda de freqüència (MHz)	**Paràmetres de Spread**		**Paràmetres de Dades**		
	Tassa de chips (kchips/s)	Modulació	Tassa de bits (Kbps)	Tassa de símbols (ksimbols/s)	Símbols
868-868.6	300	BPSK	20	20	2
902-928	600	BPSK	40	40	2
2400-2483.5	2000	O-QPSK	250	62.5	16

A la Figura 2.3 Espectres del senyal radio s'observen els espectres dels senyals de totes les bandes de freqüència.

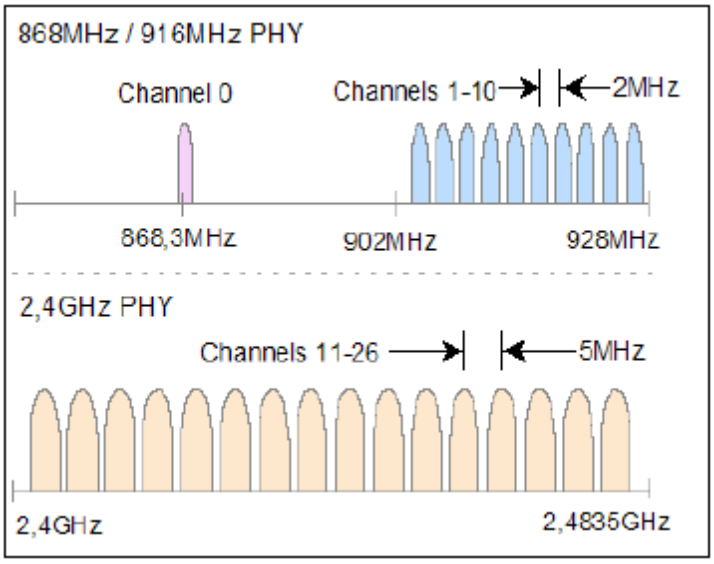


Figura 2.3 Espectres del senyal radio

En aquest projecte es fa servir la banda de 2,4 GHz, aquesta banda disposa de 16 canals de 2MHz d'ample de banda i separats entre ells per 5MHz, així s'evita la interferència entre canals. En la Figura 2.4 s'observa l'espectre d'aquest senyal.

2.4 Capa de control d'accés al medi

La capa de control d'accés al medi o capa MAC és la que ens proporciona eines per generar i controlar els *beacons*, per accedir als canals, per gestionar els *time slots*, actua en el moment d'associació i dissociació a xarxes i envia les trames de reconeixement ACK.

És una MAC caracteritzada per la seva senzillesa en comparació a altres estàndards més coneguts com poden ser el Bluetooth i és capaç de generar 4 tipus de trama diferent; la trama de beacon, la de dades, la de reconeixement ACK i trames de comandes. A continuació es detalla la trama de dades:

A la Figura 2.4 es mostra la trama de dades i posteriorment s'analitza.

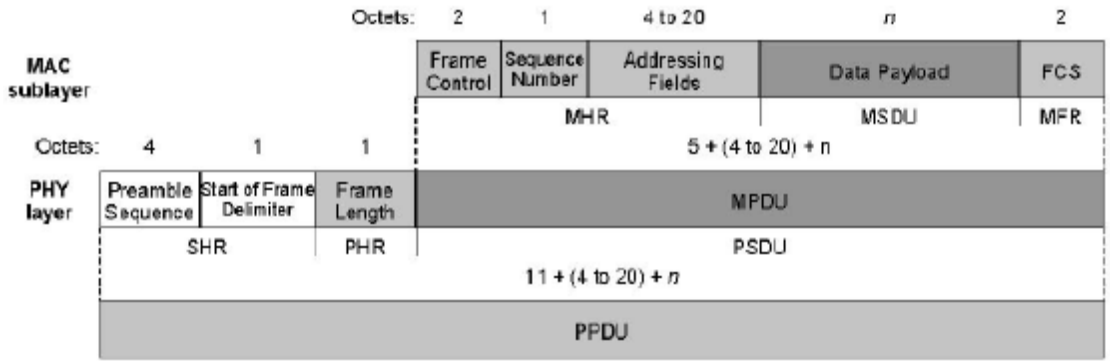


Figura 2.4 Trama de dades

Observant la Figura 2.4 de dalt a baix el que es veu primer són les dades a enviar (*Data Playload*), que en la capa MAC es nomenen MSDU (MAC *Service Data Unit*). Precedent la MSDU hi ha la MHR (MAC *Header*), aquesta és la capçalera on trobem el control de trama, el número de seqüència i l'adreçament. Posterior a la MSDU hi ha la MFR (MAC *Footer*) on s'afegeixen bits de FCS (*Frame Check Sequence*) que donen una seguretat major en la transmissió.

Tot aquest conjunt descrit forma la MPDU (MAC *Packet Data Unit*) que a la capa física s'envia amb el nom de PSDU (PHY *Service Data Unit*). Precedent la PSDU es troba la SHR (*Synchronization header*) on hi ha els 4 bytes de preàmbul i el byte que delimita l'inici de la trama i també la PHR (PHY *Header*) on s'indica la mida total de la trama.

La PPDU de la capa física ens mostra els bytes totals en la següent equació tenint en compte que *n* són els bytes de dades:

$$Bytes\ totals = 11 + (4\ a\ 20) +\ n$$

2.5 Funcionament

La capa MAC té dos modes de funcionament que es descriuran a continuació:

- Transmissió en mode *beacon*: Aquest mode va orientat a xarxes on hi ha un node que treballa com a coordinador. Aquest sincronitza als altres dispositius de nivell inferior mitjançant un *beacon* o també conegut com senyal balisa. Això permet que els dispositius puguin estar en repòs ja que només inicien l'activitat al rebre un d'aquests beacons.

- Transmissió en mode *beaconless*: Aquest mode va orientat a xarxes que no necessiten coordinador i que per tant, sempre que trobin el canal lliure podran transmetre.

En aquest TFC s'utilitza una transmissió en mode *beaconless*, tot i que s'envia un missatge de petició de mostres, aquest no es considera *beacon*, és simplement un missatge per iniciar la comunicació, no hi ha cap mena de sincronisme, sempre que hi hagi canal lliure es podrà establir la comunicació.

2.6 Mecanismes de robustesa

L'estàndard IEEE 802.15.4 està pensat per ser utilitzat en xarxes amb canals de comunicació plens d'obstacles i moltes vegades els nodes sensors no tenen un camí predefinit per establir la comunicació. És per això que es necessiten mecanismes de robustesa que assegurin una correcta transmissió. En aquest estàndard trobem els següents mecanismes:

- CSMA-CA: És un protocol de control de xarxes de baix nivell que permet que múltiples estacions utilitzin un mateix medi de transmissió. Cada equip anuncia la seva intenció de transmetre abans de fer-ho per tal d'evitar col·lisions entre els paquets de dades. D'aquesta forma, la resta d'equips de la xarxa sabran quan hi ha col·lisions i en lloc de transmetre la trama quan el mitjà està lliure, s'espera un temps aleatori addicional curt i solament si, després d'aquest curt interval el mitjà segueix lliure, es procedeix a la transmissió reduint la probabilitat de col·lisions en el canal.

- Trames de reconeixement (ACK): Paquets que ens confirmen que les dades han arribat al receptor.

- Verificació de dades (CRC): mitjançant un polinomi de grau 16 s'obté la redundància que permet comparar el que es calcula amb el paquet rebut i saber si aquest ha arribat sense danys.

- Restriccions de consum: Fa servir modes de transmissió que ajuden al baix consum dels nodes que formen la xarxa, cosa indispensable ja que funciones amb bateries finites.

CAPÍTOL 3. HARDWARE

En aquest capítol es detallarà el hardware fet servir per al disseny de la xarxa.

3.1 Xip CC2430 de Texas Instruments

El CC2430[2] és un sistema integrat a un xip. És un SOC (*System-On-Chip*) preparat per treballar amb el protocol IEEE 802.15.4, i cal dir que és molt utilitzat ja que el seu cost és considerablement baix. Les principals característiques que el diferencien dels seus antecessors és la major potència del microcontrolador i una major memòria. Ja està en el mercat una versió posterior, el CC2431, que posseeix la singularitat, respecte als altres, de disposar d'un sistema de localització capaç de determinar la posició del node respecte als altres integrants de la xarxa.

En els següents apartats es descriuran les característiques principals del CC2430.

3.1.1 Característiques generals

- Microcontrolador 8051: És un MCU (*Micro Controller Unit*) que dona un alt rendiment encara que té un baix consum. És programable mitjançant el software de IAR Workbench.

- Memòria: Conté fins a 128 KB de memòria flash no volàtil i 4 KB de memòria de dades. En el pitjor dels casos fa 1000 cicles d'escriptura o esborrat. Permet blocar codi per tal de que no pugui ser esborrat i té 4096 bytes de SRAM interna per retenir dades.

- Encriptació: Suporta l'encriptació/desencriptació AES[1] (*Advanced Encryption Standard*).

- En quant als perifèrics disposem de:

    - Potent controlador DMA que permet un ràpid moviment de la memòria.
    - Vuit canals amb resolució de 8 a 14 bits per al conversor analògic digital.
    - *Watchdog* per reiniciar el dispositiu en cas de fallida.
    - Cristall oscil·lant a 32.768 kHz.
    - 4 timers, un de 16 bits, dos de 8 bits i un utilitzant la capa MAC.

[1] L'AES és un esquema de xifrat per blocs usat en primer terme pel govern dels Estats Units però que al 2006 es va convertir en estàndard i avui en dia és un dels més utilitzats en criptografia simètrica, que és aquella en la que s'utilitza la mateixa clau per xifrar i desxifrar el missatge enviat, és a dir, emissor i receptor s'han posat abans d'acord en la clau a utilitzar.

  - 2 USARTs(*Universal Synchronous Asynchronous Receiver-Transmitter*) programables que poden treballar com UART encarregada de controlar el port sèrie o com a bus SPI, estàndard en la transmissió de dades en circuits integrats.
  - 21 pins configurables.
  - Generador de nombres aleatoris.

- Baix consum: Compta amb 4 modes d'energia, comentats posteriorment, que permeten un control total del consum de potència i fins i tot mantindre el dispositiu en mode *sleep* fins que una interrupció externa o interna el faci treballar.

- Suport de hardware en la capa MAC per al IEEE 802.15.4

Per al disseny de la xarxa de sensat d'aquest TFC s'han de tenir en compte diverses característiques generals del xip:

- En quant a la memòria de dades:
  La xarxa de sensat va emmagatzemant en la memòria de dades les mesures obtingudes de temperatura, humitat i bateria. Com el xip porta programades diferents utilitats que utilitzen aquesta memòria de dades no es tenen els 4KB disponibles. Per al disseny de la xarxa de sensat es disposen de 1,4 KB de memòria de dades utilitzable. Sabent això el protocol de sensat s'ha dissenyat utilitzant el màxim d'espai disponible.

- En quant als perifèrics:
  Per el disseny de la xarxa de sensors d'aquest TFC s'han fet servir 3 canals (un per cada paràmetre a mesurar) del conversor analògic digital. Tots tres s'han dissenyat amb una resolució de 12 bits.
  S'ha fet servir un dels USARTs en mode UART per transmetre les dades mitjançant el port sèrie.
  I per últim s'han fet servir *timers* per anar cridant les tasques a executar en cada moment.

- En quant als modes d'energia es detallen al capítol 5.

3.1.2 Especificacions del CC2430

A la Taula 3.1 s'observen els valors màxims del dispositiu.
En quant a l'alimentació del node observem que el mode escollit per aquest TFC, una pila de boto de 3.3 volts, és suficient. També s'ha de tenir en compte que el voltatge que podem llegir a qualsevol dels pins del node pot ser com a màxim de 3.6 volts. Aquesta característica s'ha de tenir en compte si es vol connectar qualsevol dispositiu extern del qual es vulgui llegir un voltatge de sortida. En el nostre cas s'ha de connectar el sensor d'humitat, que com es veurà al capítol 6, entra dintre del marge acceptat pel node.

Taula 3.1 Valors màxims del dispositiu.

Parameter	Min	Max	Units
Supply voltage	–0.3	3.6	V
Voltage on any digital pin	–0.3	VDD+0.3, max 3.6	V
Voltage on the 1.8V pins (pin no. 22, 25-40 and 42)	–0.3	2.0	V
Input RF level		10	dBm
Storage temperature range	–50	150	°C
Reflow soldering temperature		260	°C

A la Taula 3.2 veiem el consum del microcontrolador sense tenir en compte el consum dels perifèrics.
La part del consum del dispositiu, important per al disseny de la xarxa de sensors, es detallarà al capítol 5.

Taula 3.2 Consums del MCU (*Micro Controller Unit*)

Current Consumption					
MCU Active Mode, static		492		µA	Digital regulator on, High Speed RCOSC running. No radio, crystals, or peripherals.
MCU Active Mode, dynamic		210		µA/MHz	Digital regulator on, High Speed RCOSC running. No radio, crystals, or peripherals.
MCU Active Mode, highest speed		7.0		mA	MCU running at full speed (32MHz), 32MHz XOSC running. No peripherals.
MCU Active and RX Mode		27		mA	MCU running at full speed (32MHz), 32MHz XOSC running, radio in RX mode. No peripherals.
MCU Active and TX Mode, 0dBm		24.7		mA	MCU running at full speed (32MHz), 32MHz XOSC running, radio in TX mode. No peripherals.
Power mode 1		296		µA	Digital regulator on, High Speed RCOSC and crystal oscillator off. 32.768kHz XOSC, POR and ST active. RAM retention.
Power mode 2		0.9		µA	Digital regulator off, High Speed RCOSC and crystal oscillator off. 32.768kHz XOSC, POR and ST active. RAM retention.
Power mode 3		0.6		µA	No clocks. RAM retention. Power On Reset (POR) active.

A la Taula 3.3 s'observen els consums que provoquen els diferents perifèrics al ser utilitzats.
Al capítol 5 on es detallarà el consum del microcontrolador s'haurà de tenir en compte que en el nostre disseny s'utilitza 1 timer, 1 USART i l'ADC, per tant, al consum total haurem de sumar els 10 µA que consumeix el timer quan s'habilita, els 0.9 mA que consumeix l'ADC cada cop que pren una mesura i els 12 µA que consumirà el node gateway quan faci la transmissió de dades al PC mitjançant el port sèrie.

Taula 3.3 Consums dels perifèrics

Peripheral Current Consumption					Adds to the figures above if the peripheral unit is activated
Timer 1		10		µA/MHz	When enabled
Timer 2		10		µA/MHz	When enabled
Timer 3		10		µA/MHz	When enabled
Timer 4		10		µA/MHz	When enabled
Sleep Timer		0.5		µA	Including low-power RC oscillator or 32.768kHz XOSC
AES		50		µA/MHz	When encrypting/decrypting
ADC		0.9		mA	When converting
USART1 / USART2		12		µA/MHz	For each USART in use. Not including current for driving I/O pins.
DMA		30		µA/MHz	When operating, not including current for memory access
Flash write		3		mA	

3.2 Kit de desenvolupament

En els següents apartats es descriuran els diferents elements que formen el Kit de desenvolupament de *Texas Instruments* amb el qual s'ha dissenyat la xarxa de sensors.

3.2.1 CC2430EM

Aquest component, que es pot observar a la Figura 3.1, és simplement un dispositiu format pel xip CC2430 i una interfície radio compatible amb l'estàndard 802.15.4 utilitzat en la transmissió i recepció de missatges.

Figura 3.1 CC2430EM (*Evaluation Mode)[3]*

A la Taula 3.4 s'observen les propietats en transmissió i recepció del transceptor ràdio.
Aquest treballa a la freqüència de 2.4 GHz, el seu rang de potencia de transmissió oscil·la entre els -24 i els 0 dbm, i en recepció presenta una sensibilitat de -90 dbm.

Taula 3.4 Paràmetres del transceptor de RF

Parameter	Min	Typ	Max	Unit	Condition/Note
Receiver sensitivity		-90		dBm	PER = 1%, as specified by [1] Measured in 50 Ω single endedly through a balun. [1] requires –85 dBm

Parameter	Min	Typ	Max	Unit	Condition/Note
Nominal output power	-3	0		dBm	Delivered to a single ended 50 Ω load through a balun. [1] requires minimum –3 dBm
Programmable output power range		24		dB	The output power is programmable in 8 steps from approximately –24 to 0 dBm.

3.2.2 SmartRF04EB

El SmartRF04EB és el dispositiu que ens permet comunicar un ordinador i el CC2430. És un suport de desenvolupament que permet depurar el codi i verificar el seu correcte funcionament, i disposa d'una sèrie de hardware que ens permet afegir aplicacions al CC2430 sense necessitat de construir-les. El SmartRF04EB ens aporta perifèrics com polsadors, visualitzadors LCD, jocs de LED's, interfície USB, interfície RS-232, entrada jack stereo per so o micròfon, connectors SMA i pins que faciliten l'accés als pins del xip.

Per connectar el dispositiu al ordinador es fa servir un cable USB, aquest cable és el que proporciona l'alimentació de 3.3 volts. El xip va acoblat directament com s'observa a la part superior esquerra de la Figura 3.2.

Figura 3.2 SmartRF04EB amb CC2430EM acoblat.[3]

A la Figura 3.3 observem els perifèrics detallats.

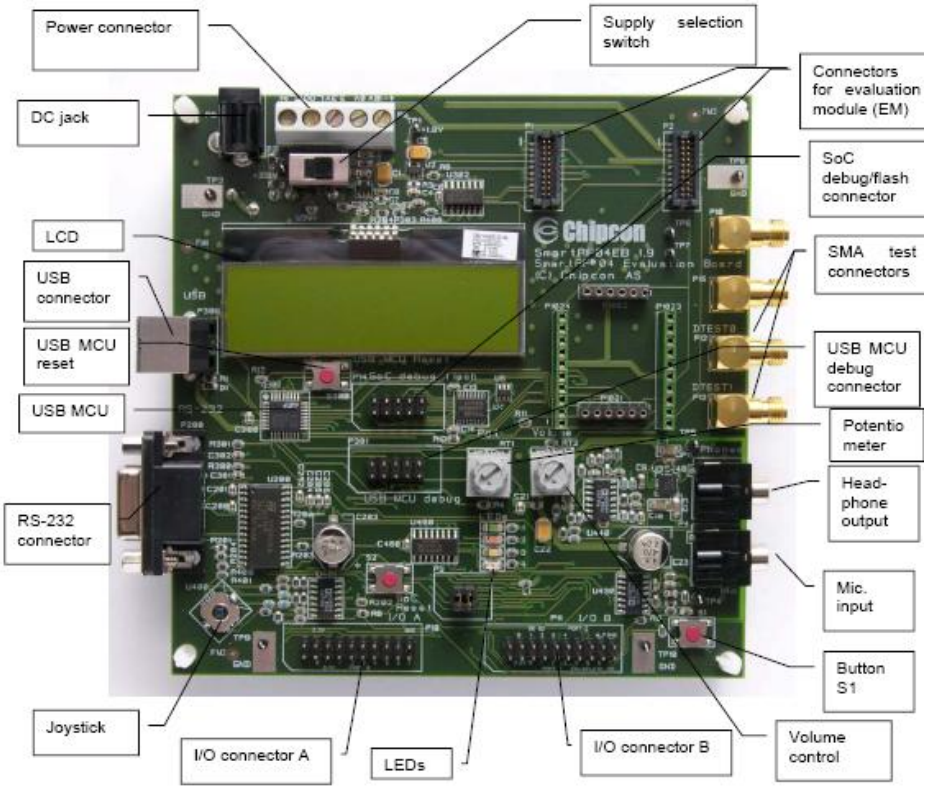


Figura 3.3 SmartRF04EB amb les parts detallades[3]

3.2.3 SOC_BB

El SOC_BB (Battery Board) té com a funcionalitat alimentar el SoC sense que hagi d'estar connectat a un ordinador. Alimenta al SoC mitjançant unes piles que porta a la part inferior. Cal dir que té menys funcionalitats que el SmartRF04EB.

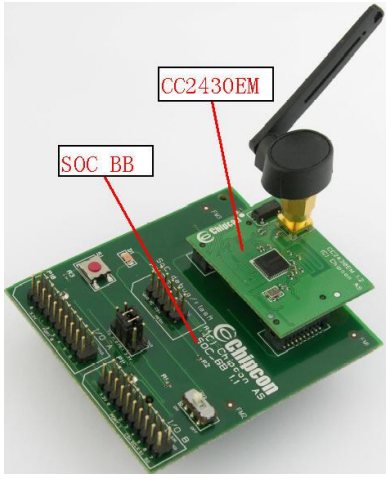


Figura 3.4 SoC_BB (Battery Board) [3]

3.2.4 SOC_DEM

El SOC_DEM és un adaptador per poder acoblar al SmartRF04EB un SOC_BB. D'aquesta forma es pot programar o depurar directament des de el SOC_BB.

En aquest TFC és imprescindible l' utilització del SoC_DEM ja que tot el codi s'ha programat sobre uns nodes fabricats directament pel Grup de Recerca de Xarxes sense Fils (WNG, Wireless Networks Group) de la Universitat Politècnica de Catalunya (UPC) que no són debugables directament sobre l'SmartRF04EB.

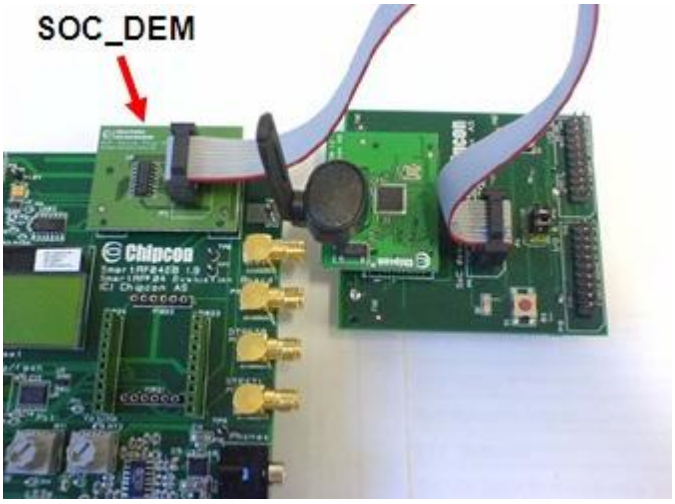


Figura 3.5 SoC_DEM connectat al SmartRF04EB [3]

3.3 CC2430N i connector TFSD de SAMTEC

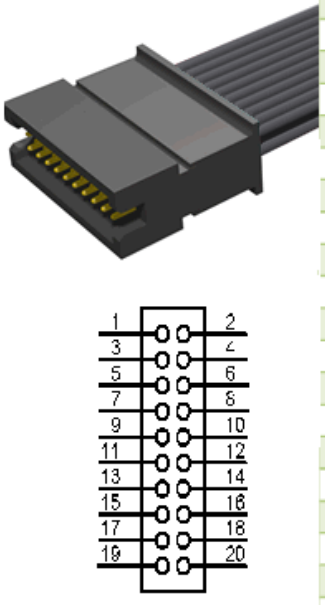
Tot el codi dissenyat amb el kit de *Texas Instruments* es programa finalment sobre els CC2430N. Aquests són els nodes fabricats pel Grup de Recerca de Xarxes sense Fils (WNG, Wireless Networks Group) de la Universitat Politècnica de Catalunya (UPC).
Tenen les mateixes funcionalitats que els CC2430EM, i tot i que a la figura 3.6 no és tan fàcil d'apreciar la interfície ràdio com ho era a la Figura 3.1 del CC2430EM, els CC2430N també disposen d'aquesta interfície compatible amb l'estàndard 802.15.4.
La diferència rellevant és el cost més reduït, paràmetre important en el disseny de la xarxa.
Els nodes CC2430N van alimentats amb una pila de botó de 3.3 volts col·locada a la part posterior del node.

Figura 3.6 CC2430N

El connector que s'ha fet servir en aquest TFC per connectar el node CC2430N amb els altres components de la xarxa és el TFSD-10-30-G-03.00-S de SAMTEC.

L'aspecte del connector és el que s'observa a la Figura 3.7. En un extrem del connector trobem la connexió mascle que encaixa amb la femella que trobem a la part posterior del CC2430N.
Per l'altre extrem del connector trobem 20 cables pertanyents als 20 pins configurables que disposem. Tots els cables del connector van numerats.

Per relacionar cada pin amb la seva funció es fa servir la taula de la Figura 3.7. Aquesta taula és el mapeig de pins que ens indica la relació entre les funcions dels pins originals del CC2430EM amb els del CC2430N. Per exemple, si volem fer servir el pin 3 del port 0, en el CC2430EM s'hauria de fer servir el pin0_3, però en canvi al CC2430N s'haurà de fer servir el cable del connector de SAMTEC numerat amb un 7.



CC2430EM	CC2430N
P0_0	1
P0_1	3
P0_2	5
P0_3	7
P0_4	9
P0_5	11
P0_6	13
P0_7	15
P1_3	4
P1_4	14
P1_5	16
P1_6	18
P1_7	20
P2_0	2
P2_1	10
P2_2	12
VDD PLACA	6
VDD PLACA	8
RESET	17
0V (GND)	19

Figura 3.7 Connector de SAMTEC i mapeig amb els del connector del CC2430N

CAPITOL 4. Entorn de programació

En aquest capítol es vol descriure tot el relacionat amb l'entorn de programació: Software utilitat, pila de protocols, i el precompilador, part essencial per al correcte funcionament del codi.

4.1 IAR Embeeded WorkBench

El software utilitzat per programar els nodes és l'IAR Embeeded Workbench 7.50A. És un entorn de programació que programa en llenguatge C i té una aparença senzilla i intuïtiva si l'usuari ja ha utilitzat prèviament altres programes de desenvolupament de codi.
El programa té dos modes d'ús: el mode de programació i el mode de depuració.

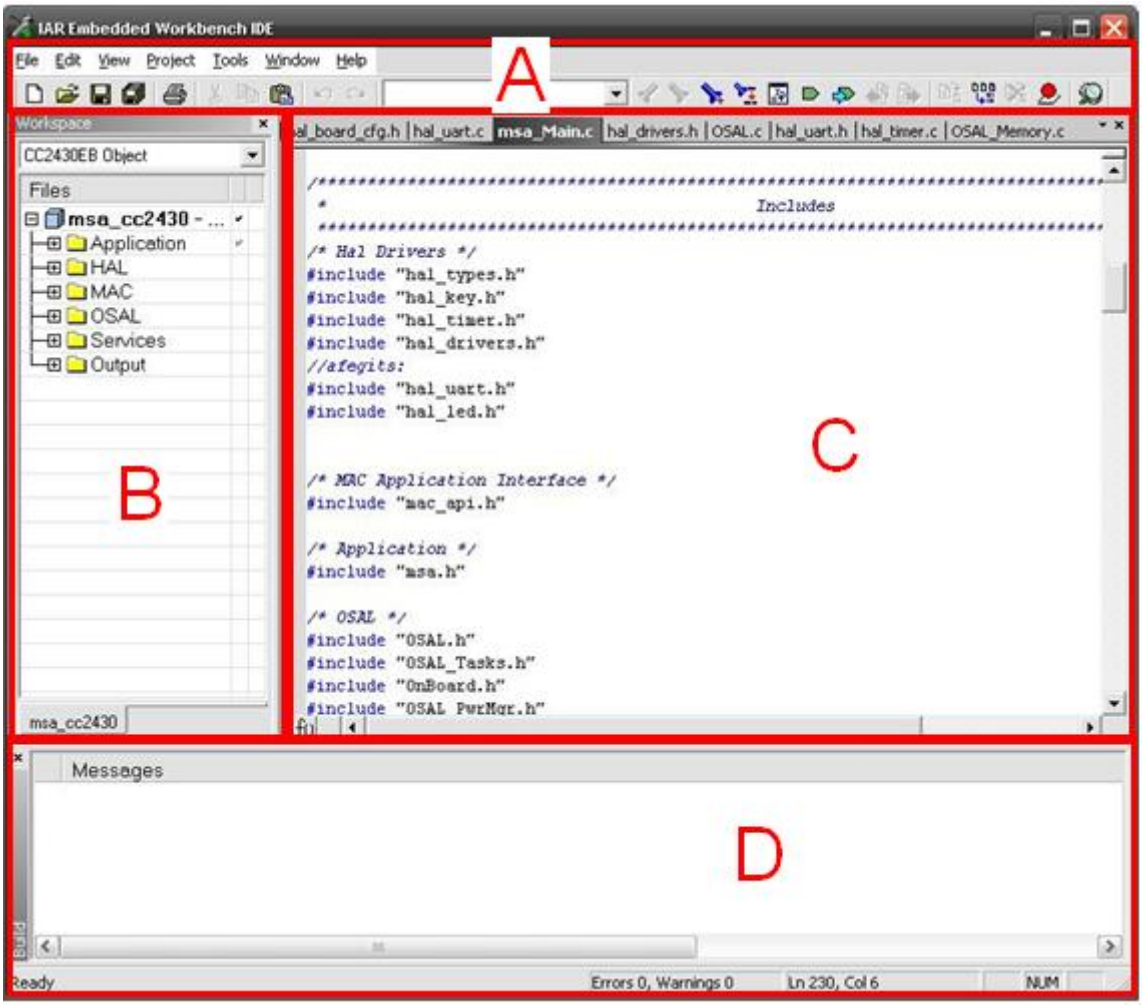


Figura 4.1 Pantalla principal de programació

En el cas d'estar programant podem distingir entre les 4 àrees marcades a la Figura 4.1 amb les lletres A,B,C i D:

En el cas d'estar programant podem distingir entre les 4 àrees marcades a la Figura 4.1 amb les lletres A,B,C i D:

- A: En aquesta àrea podem navegar pels diferents menús. Podem canviar les configuracions o bé crear, obrir o guardar projectes. Amés, en aquesta barra, disposem d'icones que faciliten l'ús de l'entorn de programació.

- B: Aquí es troba l'àrea de treball o *workspace*. En aquest espai podem accedir a tots el arxius que formen part del projecte.

- C: Aquí es on s'escriu el codi. A la part superior s'observen una sèrie de pestanyes que faciliten la navegació entre els arxius del projecte que estiguin oberts.

- D: Aquesta zona és la finestra d'errors. Si es compila el projecte els errors de codi són mostrats en aquesta finestra, únicament fent doble clic sobre l'error ja ens porta automàticament a la línia de codi afectada.

En el cas d'estar depurant el codi, podem obrir diferents finestres en funció de la informació que vulguem obtenir. A la Figura 4.2 es mostren les finestres més utilitzades en el desenvolupament d'aquest projecte.

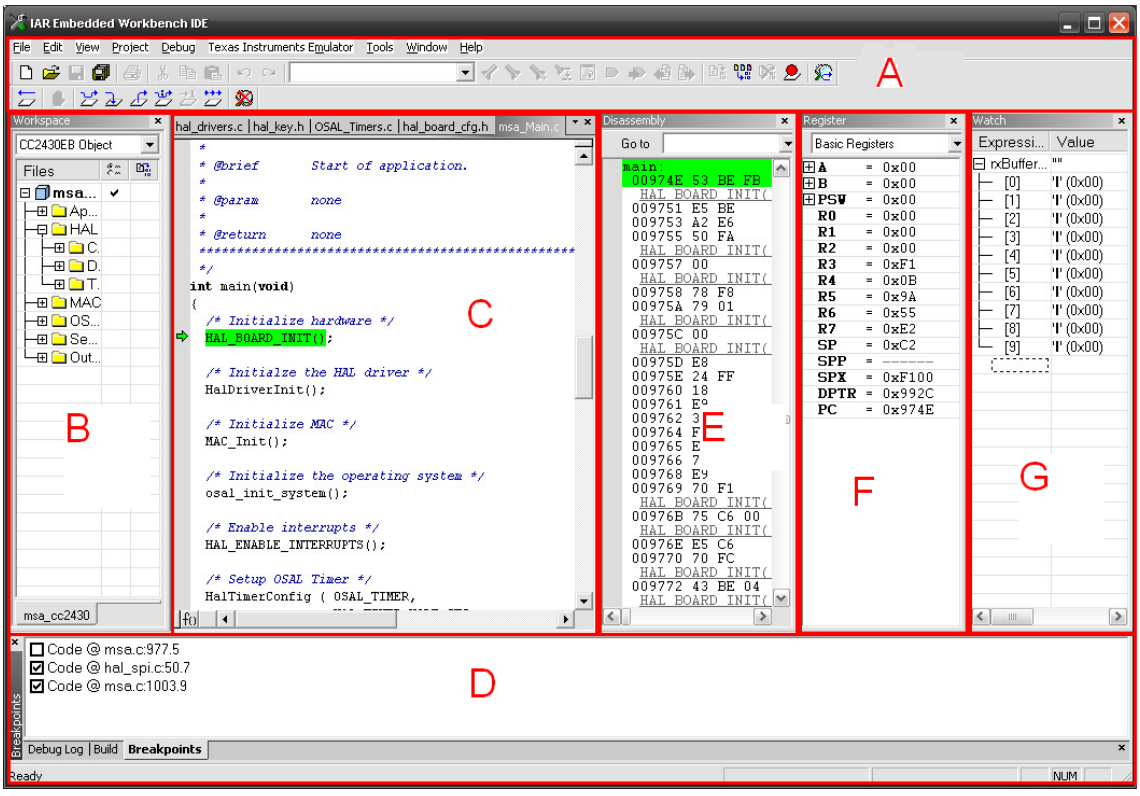


Figura 4.2 Pantalla principal de depuració de codi

- A: La funció es la mateixa que en el mode anterior. La única diferencia és que ara s'utilitzen els botons que serveixen per gestionar el mode de depuració. Podem fer parades, executar pas a pas, o executar sense pauses.

- B: Aquesta finestra no canvia respecte el mode anterior.

- C: Es segueix mostrant el codi programat. La marca verda que s'observa indica la propera funció que s'executarà.

- D: Igual que al mode anterior s'observen els errors. La novetat per el mode de depuració és que es poden observar les línies amb punt de parada.

- E: S'observen l'execució del codi en mode ensamblador.

- F: Mostra els valors dels registres.

- G: Mostra els valors de les variables que es desitgin.

4.2 TIMAC

Les sigles TIMAC volen dir Texas Instruments Medium Access Control.
TIMAC es una pila de protocols que s'utilitza per dispositius amb l'estàndard IEE 802.15.4.

Com a qualsevol pila de protocols, TIMAC està composta per diferents capes que permeten organitzar l'aplicació. A cada capa li correspon un conjunt de funcions.

Les 4 capes de TIMAC són:

- MAC: controla el medi físic del SoC. És una capa que està molt protegida. Es poden observar les capçaleres de les funcions però no el seu funcionament.

- HAL (Hardware Abstract Layer): la capa HAL és una abstracció del hardware, és a dir, permet controlar el hardware de forma més funcional, no s'ha de conèixer el cablejat de connexions, ni que pin correspon a cada acció. Amb aquesta capa es pot controlar, mitjançant simples funcions, l'ADC, els leds, la pantalla LCD, el joystick, els modes d'energia, el UART i l'enviament de missatges.

- OSAL (Operating System Abstraction Layer): Aquesta capa la podríem anomenar com el petit sistema operatiu integrat al nostre SoC. És la capa que s'encarrega de prioritzar i organitzar les diferents tasques a realitzar.

- Application: es la capa més alta de totes, utilitza totes les capes anteriorment citades. És en aquesta capa on es programa la gran majoria de codi, com el seu nom indica és la capa on gestionarem l'organització de la rutina pensada per al nostre dispositiu.

4.3 Precompilador

El *precompilador* tracta una part del codi que s'està desenvolupant. Al precompilador es defineixen constants i símbols que es tenen en compte a l'hora de compilar. Aquestes definicions tenen a veure amb l'activació de perifèrics, tipus de dispositiu, adreçament i estalvi d'energia entre d'altres.

Per programar els nodes que formen part de la xarxa de sensors el *precompilador* conté els següents elements:

CC2430N→ Indica que es vol compilar sobre un node C2430N

NODE_SENSOR→ Indica que s'ha d'incloure només la part de codi que correspon al node sensor.

POWER_SAVING→ Indica estalvi d'energia activat (capítol 5)

NAL_ADDR=0x02 → Adreça origen del node

GROUP=0x04 → Grup origen del node

HAL_ADC=TRUE → Indica que s'ha d'habilitar l'ADC

Per programar el node gateway:

CC2430N→ Indica que es vol compilar sobre un node C2430N

GATEWAY→ Indica que s'ha d'incloure només la part de codi que correspon al node gateway.

POWER_SAVING→ Indica estalvi d'energia activat (capítol 5)

NAL_ADDR=0x01 → Adreça origen del node

GROUP=0x04 → Grup origen del node

HAL_UART=TRUE → Indica que s'ha d'habilitar l'UART

HAL_UART_ISR = TRUE → Habilita les interrupcions del UART

CAPÍTOL 5. CONSUM

En aquest capítol és detallarà un dels aspectes importants en el disseny de la xarxa de sensors: el consum.

A la xarxa de sensors dissenyada en aquest TFC distingim entre els nodes sensors i el node gateway o coordinador. Tots els paràmetres que es descriuran en aquest capítol només són vàlids pels nodes que formen la xarxa de sensat, ja que pel node gateway, com no hi ha restriccions en quant a l'alimentació, no es té en compte el consum.

5.1 Modes d'energia

El CC2430 té 4 modes d'energia [4] nomenats PM0, PM1, PM2 i PM3 sent aquest últim el mode de menys consum energètic. Les principals característiques són:

- **PM0 o *Active Mode*:** És el mode funcional, aquest mode ho manté tot treballant: regulador de voltatge, CPU, perifèrics, oscil·ladors i RF.

- **PM1:** Mode parcialment funcional utilitzat quan es vol reactivar el dispositiu en un temps curt . Des de aquest mode d'energia es triguen 2μs per tornar a l'*Active Mode*. En aquest mode es desactiven els oscil·ladors d'alta freqüència i es mantenen encesos els oscil·ladors de 32kHz i el regulador de voltatge.

- **PM2:** Mode que consumeix menys que l'anterior i sol ser utilitzat quan es vol despertar mitjançant el *sleep timer*. En aquest mode es manté l'oscil·lador de 32kHz però es desactiva el regulador de voltatge. Per tornar a l'*Active Mode* des de el PM2 es triguen 54μs.

- **PM3 :** Mode que consumeix menys energia. Aquest mode és activat quan el que es vol és despertar el dispositiu mitjançant una interrupció externa. Tot es manté desactivat excepte el reset. Per tornar a l'*Active Mode* des de el PM3 es triguen 54μs.

5.2 Consums del microcontrolador

A la taula 5.1 podem observar els consums d'energia del microcontrolador segons la tasca que està realitzant. Aquests consums són sense tenir en compte la suma que provoca l'ús dels perifèrics (Taula 3.3).

Taula 5.1 Consums del MCU (*Micro Controller Unit*)

Current Consumption					
MCU Active Mode, static		492		µA	Digital regulator on, High Speed RCOSC running. No radio, crystals, or peripherals.
MCU Active Mode, dynamic		210		µA/MHz	Digital regulator on, High Speed RCOSC running. No radio, crystals, or peripherals.
MCU Active Mode, highest speed		7.0		mA	MCU running at full speed (32MHz), 32MHz XOSC running. No peripherals.
MCU Active and RX Mode		27		mA	MCU running at full speed (32MHz), 32MHz XOSC running, radio in RX mode. No peripherals.
MCU Active and TX Mode, 0dBm		24.7		mA	MCU running at full speed (32MHz), 32MHz XOSC running, radio in TX mode. No peripherals.
Power mode 1		296		µA	Digital regulator on, High Speed RCOSC and crystal oscillator off. 32.768kHz XOSC, POR and ST active. RAM retention.
Power mode 2		0.9		µA	Digital regulator off, High Speed RCOSC and crystal oscillator off. 32.768kHz XOSC, POR and ST active. RAM retention.
Power mode 3		0.6		µA	No clocks. RAM retention. Power On Reset (POR) active.

Els nodes que formen la xarxa de sensat utilitzen l'estalvi d'energia que ens proporciona el software programat al CC2430. És per això que són capaços d'assolir qualsevol dels *power modes* que s'han detallat a l'apartat anterior.

Segons el disseny de la nostra xarxa hem de tenir en compte els següents consums:

- Quan el node sensor està rebent o transmetent. És aquí on es produeixen els màxims consums d'energia, 27mA i 24.7mA respectivament.

- Quan el node sensor no està ni transmetent ni rebent missatges de dades però si que està obtenint-les es troba al PM1, ja que és l'estat de menys consum que manté el regulador de voltatge activat, dispositiu necessari per prendre correctament les mesures de voltatges externs mitjançant l'ADC. Durant la realització d'aquestes tasques el microcontrolador consumeix 296µA

- Quan el node està totalment inactiu s'activa el mode d'energia PM2, quan s'ha de retornar al mode actiu s'activa la interrupció del *sleep timer.* Durant aquest període es consumeix 0.9µA.

5.3 Dispositiu treballant a baix consum

Els dispositius aconsegueixen entrar al mode d'energia PM2 quan la capa OSAL no realitza cap tasca que necessiti tenir activa la interfície ràdio ni el regulador de voltatge. Aquests períodes de màxim estalvi es troben quan el sensor no està ni rebent, ni transmetent ni mesurant dades.

En aquests períodes d'estalvi s'ha de tenir en compte que la ràdio té que està activa si es vol ser capaç de rebre el missatge de petició de mostres. Si en

aquests períodes, la capa OSAL, detectés que la ràdio està encesa mai es podria entrar al mode PM2.

És per això que s'ha dissenyat una rutina que encén i apaga la ràdio periòdicament. La manté encesa durant 100ms i l'apaga durant 900ms.
Amb això aconseguim que en la tasca d'esperar el missatge de petició, el node estigui durant només el 10% del temps consumint el màxim d'energia.

La rutina d'encendre i apagar la ràdio s'inicia en posar en marxa el dispositiu i sempre està activa. Aquesta tasca s'implementa amb el codi següent:

```
if (events & MSA_RADIO_ON)
  {
    set_radio(RX_ON);
    osal_start_timerEx(MSA_TaskId, MSA_RADIO_OFF, 100);
    return events ^ MSA_RADIO_ON;
  }

if (events & MSA_RADIO_OFF)
  {
    set_radio(RX_OFF);
    osal_start_timerEx(MSA_TaskId, MSA_RADIO_ON, 900);
    return events ^ MSA_RADIO_ON;
  }
```

5.4 Tasques sense estalvi d'energia

La capa que s'encarrega d'activar i controlar l'estalvi d'energia és la capa OSAL. És aquesta capa la que executa les funcions de la capa HAL que permeten canviar de mode d'energia. La capa OSAL canvia de mode d'energia en funció de les necessitats que impliqui la tasca que està executant.

Per tal de que el dispositiu sigui capaç d'habilitar els modes d'energia automàticament s'ha de configurar les següents línies de codi:

- En el *precompilador* s'ha de definir, com es veu a l'apartat 4.3, el símbol POWER SAVING.

- En el *main* del programa s'ha d'escriure la comanda `osal_pwrmgr_device(PWRMGR_BATTERY);`. Aquesta comanda s'executa al iniciar el dispositiu i habilita l'estalvi d'energia.

Per últim, s'ha de tenir en compte que hi han tasques que no poden estalviar energia. Aquestes tasques són les que utilitzen la ràdio. No es pot permetre el canvi d'energia a un mode de baix consum quan s'utilitza la ràdio ja que qualsevol dels modes que estalvia energia no contempla que la ràdio s'activi.

És per això que en les tasques de rebre i enviar missatges, i en la tasca que encén la ràdio per veure si el coordinador envia el missatge de petició de

mostres, s'ha d'escriure la següent línia de codi que inhabilita el canvi a modes de baix consum:

```
osal_pwrmgr_task_state(PWRMGR_HOLD);
```

CAPITOL 6. PARÀMETRES A MESURAR

6.1 Temperatura

En aquest capítol es descriurà el sensor de temperatura integrat al CC2430 i com calcular la temperatura.

6.1.1 Especificacions

Primer de tot es mostra en la següent taula els paràmetres del sensor intern de temperatura [5] del CC2430.

Taula 6.1 Paràmetres del sensor de temperatura

Parameter	Min	Typ	Max	Unit	Condition/Note
Output voltage at –40°C		0.648		V	Value is estimated
Output voltage at 0°C		0.743		V	Value is estimated
Output voltage at +40°C		0.840		V	Value is estimated
Output voltage at +80°C		0.939		V	Value is estimated
Temperature coefficient		2.45		mV/°C	Fitted from –20°C to +80°C on estimated values.
Absolute error in calculated temperature		–8		°C	From –20°C to +80°C when assuming best fit for absolute accuracy on estimated values: 0.743V at 0°C and 2.45mV / °C.
Error in calculated temperature, calibrated	-2	0	2	°C	From –20°C to +80°C when using 2.45mV / °C, after 1-point calibration at room temperature. Values are estimated. Indicated min/max with 1-point calibration is based on simulated values for typical process parameters
Current consumption increase when enabled		280		µA	

Com s'observa a la taula 6.1 el sensor de temperatura intern mesura entre -40 i +80 ºC, especificacions que assoleixen els objectius de l'escenari dissenyat per aquest projecte.

El sensor de temperatura integrat està connectat directament al conversor analògic-digital. Si es configura l'ADC per que ens proporcioni el voltatge donat pel sensor de temperatura podem calcular fàcilment la temperatura ambient.
A la Figura 6.1 s'observa el descrit anteriorment:

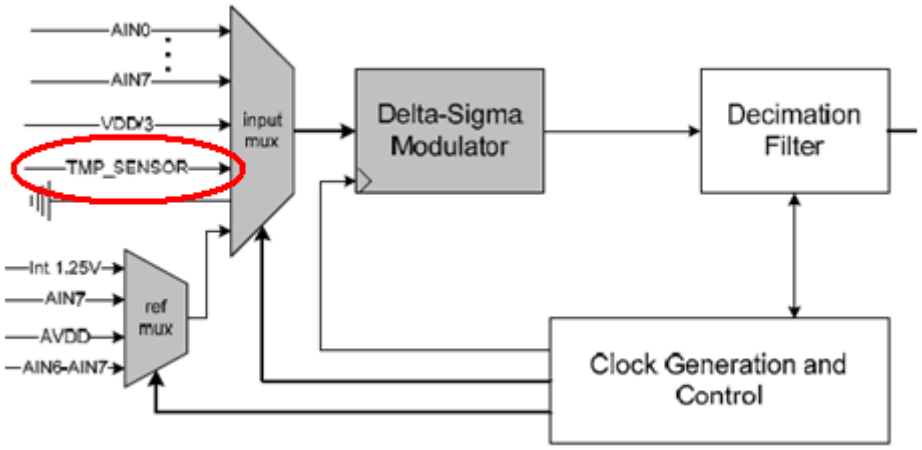


Figura 6.1 Diagrama de blocs del conversor analògic digital

6.1.2 Paràmetres

A la Taula 6.2 observem els paràmetres necessaris per poder calcular la temperatura després de conèixer el voltatge que proporciona el conversor analògic digital. Observem el voltatge obtingut a 0ºC i el coeficient de temperatura.

Taula 6.2 Paràmetres del sensor de temperatura

	Typical Output Voltage [mV] @ 0°C	Typical Temperature Coefficient [mV/°C]	Datasheet Revisions (from where the numbers are taken)
CC1110/CC1111	755	2.54	SWRS033D [1]
CC2510/CC2511	750	2.43	SWRS055D [2]
CC2430/CC2431	743	2.45	SWRS036F [3]

6.1.3 Càlculs prèvis

A la Figura 6.2 observem el voltatge de sortida en funció de la temperatura que el sensor detecti.

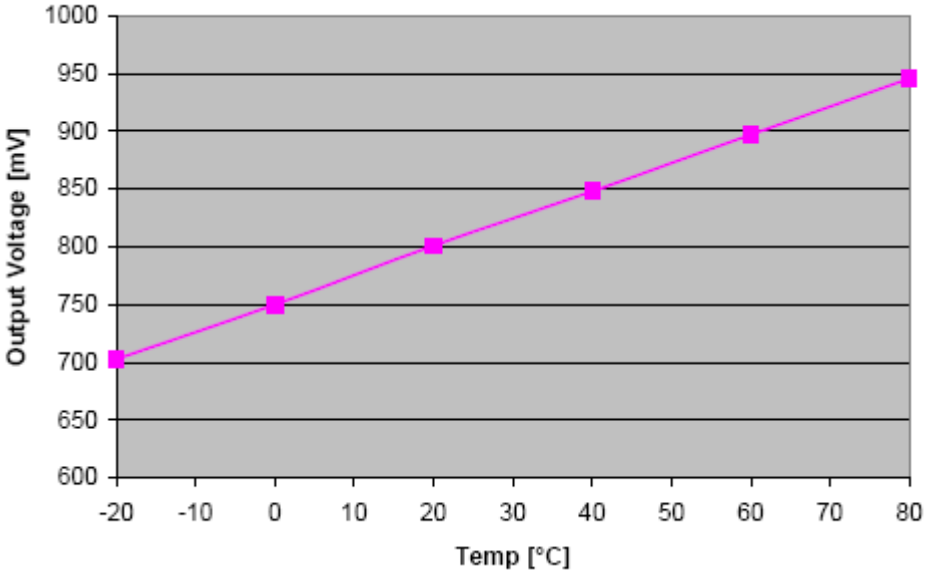


Figura 6.2 Voltatge de sortida en funció de la temperatura

D'aquesta gràfica, i fent servir els paràmetres anteriorment definits a la taula 6.2, s'obté la següent equació:

Vout [mV] = Coef. Temperatura [mV/(ºC)] · Temperatura ambient [ºC]+ Vout a 0º [ºC] **(6.1)**

De l'equació 6.1, el voltatge de sortida (Vout) és el que es llegeix mitjançant l'ADC, el coeficient de Temperatura i el voltatge de sortida a 0ºC són dades conegudes del datasheet, per tant deduïm:

Temperatura ambient [ºC] = (Vout [mV] - Vout a 0º [ºC]) / Coef. Temperatura [mV/(ºC)] **(6.2)**

Per llegir el voltatge retornat per l'ADC s'ha configurat un canal amb 12bits de resolució i un voltatge de referència intern de 1.25 volts, per tant:

Vout [mV] = *Valor ADC* * 1250 [mV] / 2047 **(6.3)**

6.1.4 Calibració

Segons les especificacions, l'error màxim de temperatura ha de ser de ±2ºC.
Per poder garantir-ho, és necessari establir un punt de calibració per tal d'ajustar al màxim la mesura, ja que la sortida de l'ADC podria tenir alguna desviació (*offset*) respecte al valor teòric de la temperatura de l'equació 6.1.

La manera més senzilla és fer una mesura de temperatura en una habitació en la qual es conegui amb exactitud la temperatura ambient. D'aquesta manera podrem veure la diferència entre el valor real, i el valor teòric. L'equació que ens determinarà l'*offset* exacte és:

VOffset [mV] = Vmesurat [mV] – (Vout a 0ºC [mV] + = Coef. Temp. [mV/(ºC)] · Temp. [ºC] **(6.4)**

Mesurant la temperatura al laboratori s'obté un valor de 30.6ºC. Amb aquesta mesura i l'equació 6.4 obtenim que l'*offset* que s'ha d'afegir a la mesura final per tal de que aquesta sigui el més exacta possible és de 16.03mV.

6.1.5 Codi per configurar el sensor de temperatura intern

Primer de tot cal definir les variables que permeten convertir el voltatge retornat per l'ADC en graus centígrads.

```
#define CONSTA 0.61065
#define OFFSET_DATASHEET 743
#define OFFSET_MEASURED_AT_30_DEGREES_CELCIUS 16.03
#define OFFSET (OFFSET_DATASHEET +
OFFSET_MEASURED_AT_30_DEGREES_CELCIUS)
#define TEMP_COEFF 2.45
```

A l'arxiu hal_adc.c s'ubica la funció que s'encarrega de llegir la temperatura:

```
float HalAdcCheckVdd (){
unsigned int value;
ADCIF = 0;
ADCCON3   =   (HAL_ADC_REF_125V   |   HAL_ADC_DEC_256   |
HAL_ADC_CHN_TEMP);
while ( !ADCIF );
value = ADCL;
value |= ((unsigned int) ADCH) << 8;
value >>= 4;
outputVoltage = value * CONSTA;
return ((outputVoltage - OFFSET) / TEMP_COEFF);
```

6.2 Humitat

En aquest apartat es detallarà el sensor extern que mesura el nivell d'humitat del sòl i com el node sensor recol·lecta les dades. El sensor utilitzat en aquest projecte és el EC-5 de *Decagon Devices*[6].

6.2.1 Especificacions

Ara es detallaran les principals especificacions del sensor d'humitat:

- Temps de mesura: 10 milisegons
- Sensibilitat: com a mínim 0.03 m^3/m^3 fins a 8 dS/m
- Alimentació: 2.5 volts – 3.6 volts @ 10mA
- Sortida: 10-40 % del voltatge d'entrada
- Rang d'operació: -40 a 60 ºC
- Dimensions: 8.9cm x 1.8cm x 0.7cm
- Longitud del cable de sortida: 5 metres

- Connector de sortida: jack de 3.5mm tot i que es poden separar els cables si el *datalogger* no té l'entrada adient.

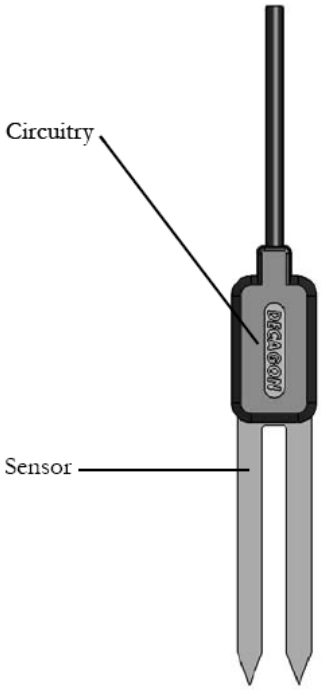


Figura 6.3 EC-5 de *Decagon Devices*

6.2.2 Connexió de sortida del sensor

Si es disposa d'un *datalogger* de *Decagon Devices* només cal connectar el cable de sortida que es detalla a la Figura 6.4

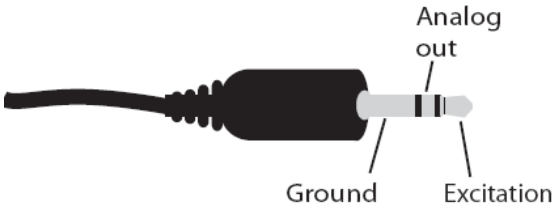


Figura 6.4 Stereo Jack de 3.5 mm (cable de sortida del sensor d'humitat)

Si no es disposa d'un *datalogger* amb aquest tipus de connexió el que s'ha de fer és adaptar una femella de jack stereo com s'explicarà a l'apartat 6.2.4.

6.2.3 Instal·lació del sensor

Aquest sensor mesura directament la proporció d'aigua que troba al sòl i així extrau una mesura d'humitat relativa, és per això que cal inserir el sensor en una zona on no trobem materials al subsòl que puguin malmetre la mesura. Objectes com branques o pedres que es trobessin entre les potes del sensor afectarien la qualitat de la mesura, i qualsevol objecte metàl·lic podria fer variar el camp magnètic del sensor fent que el valor donat no fos el correcte.

Coneixent això es procedeix a inserir el sensor al subsòl com s'observa a la Figura 6.5. El sensor ha estat dissenyat per una fàcil inserció al terra, si el sòl és massa dur no s'ha de forçar sinó que s'ha de fer un petit forat per poder introduir-hi el sensor.

La posició ideal per minimitzar pèrdues és la de la Figura 6.5; perpendicular a la superfície.

Si per qualsevol causa es volgués treure el sensor del terra el que mai es pot fer és estirar el cable ja que es podrien trencar les connexions.

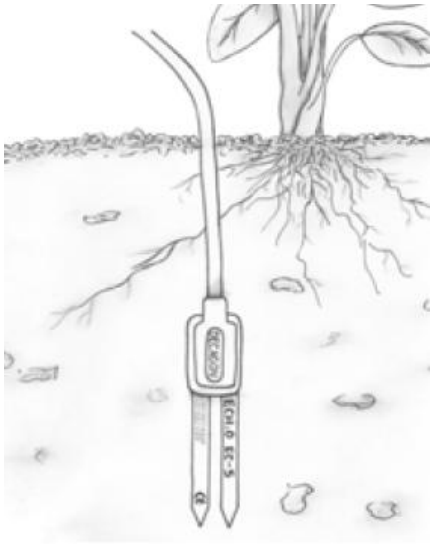


Figura 6.5 Sensor d'humitat EC-5 inserit al subsòl

6.2.4 Recol·lecció de les dades

Per tal de que el node sensor sigui capaç de llegir el voltatge que dona els sensor d'humitat s'ha d'acoblar la sortida jack stereo als pins del node, tal i com s'explica a l'apartat 1.2.

Cal tenir en compte que per poder llegir correctament el voltatge de sortida del sensor d'humitat s'ha de poder alimentar aquest com a mínim amb 2.5 volts, s'ha de poder suportar un voltatge de fins a 3.3 volts al pin del node amb el qual es vol llegir, i s'ha de poder configurar una resolució de 12 bits. El CC2430 suporta aquestes limitacions.

Es necessita un connector jack femella per tal d'adaptar la sortida del EC-5 al node, a la Figura 6.6 es pot observar a quina part del connector aniran soldades l'alimentació, la terra i quina serà la part que ens proporcionarà el senyal de sortida.

Figura 6.6 Connector femella jack stereo

La part vermella correspon a l'alimentació, la part blanca es terra, i la part negra és de on es llegeix el voltatge de sortida. Per tal d'assegurar les connexions cal mesurar amb un multímetre abans de soldar.

Un cop acoblat el connector femella al SoC només cal programar el node perquè llegeixi del canal d'entrada de l'ADC corresponent. Com es veu a l'apartat 1.2, la senyal de sortida del EC-5 es connecta al pin 0 del port 0. La línia de codi que ens permet llegir per aquest canal amb una resolució de 12 bits és:

```
Humitat = HalAdcRead (HAL_ADC_CHN_AIN0, HAL_ADC_DEC_256);
```

HalAdcRead és la funció de la capa HAL encarregada de llegir de l'ADC segons el canal i resolució configurats.
HAL_ADC_CHN_AIN0 és el canal corresponent al pin 0 del port 0.
HAL_ADC_DEC_256 la resolució de 12 bits.

6.3 Nivell de Bateria

En aquest apartat es descriu com es llegeix el valor de la bateria que té el node.

6.3.1 Obtenció del nivell de bateria

Els nodes que formen part de la xarxa de sensat són autònoms. Estan alimentats per una pila de botó de 3.3 volts. El disseny de la xarxa de sensat ha optimitzat molt el consum dels nodes, tot i això les bateries són finites i per tal de saber quan s'han de canviar és interessant mesurar el seu estat.

Per saber l'energia restant que els hi queda a les bateries s'ha de configurar l'ADC perquè llegeixi del canal que conté el voltatge que proporciona l'alimentació del node. Si sabem que 3.3 volts és el 100% de bateria, només s'ha d'obtenir el voltatge que dona la pila i aplicar un senzill factor de conversió.

La línia de codi que ens proporciona aquesta lectura és:

```
Battery = HalAdcRead(HAL_ADC_CHN_VDD3,HAL_ADC_DEC_256);
```

HalAdcRead és la funció de la capa HAL encarregada de llegir de l'ADC basant-se en el canal i resolució configurats.
HAL_ADC_CHN_VDD és el canal corresponent al voltatge d'alimentació.
HAL_ADC_DEC_256 la resolució de 12 bits.

CAPITOL 7. TRACTAMENT DE LES DADES I PROTOCOL D'ENVIAMENT

En aquest capítol es descriurà el procés d'obtenció i enviament de les mesures de temperatura, humitat i bateria. Seguint la Figura 7.1 es veurà com es van processant les dades des del node sensor fins l'ordinador.

Tot el codi descrit en aquest capítol es troba a l'arxiu principal msa.c.

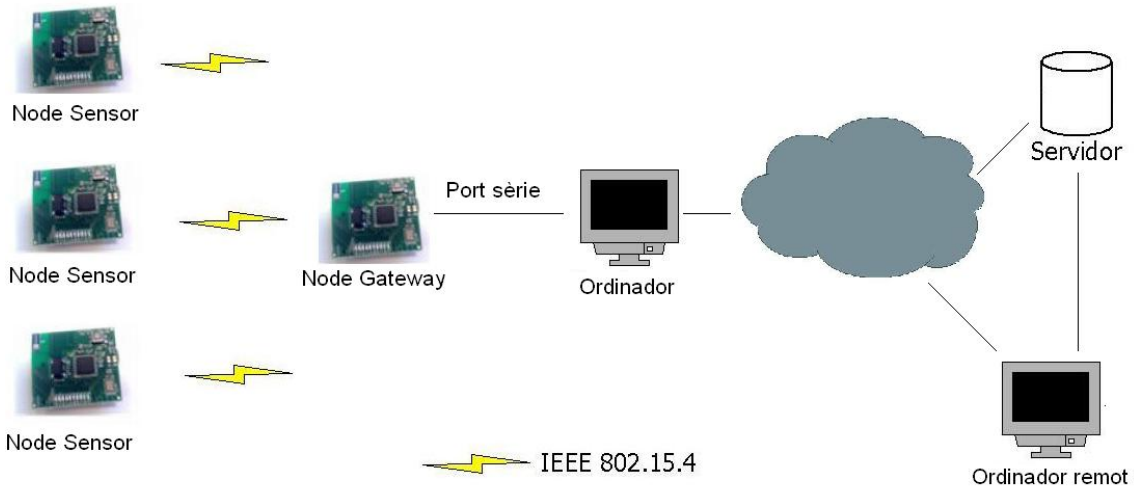


Figura 7.1 Escenari

7.1 Tractament de les dades al node sensor

El node sensor és l'encarregat d'obtenir les dades de temperatura, humitat i bateria.

Aquest procés d'obtenció de dades el realitza la tasca anomenada MSA_SENSAT, el codi referent a aquesta tasca és:

```
/****************SENSAT*********************/
     /* Condició per sobreescriure correctament*/
      if (nou == 990){
        nou = 0;
        vell = 5;
      }

      Temperatura = HalAdcCheckVdd();
      Temperatura = Temperatura *100;
      Temp = (int)Temperatura;
      CastTemp = (uint8 *) &Temp;
      VSend[nou] = *CastTemp;
      VSend[nou +1] = *(CastTemp+1);
```

```
Humitat=HalAdcRead (HAL_ADC_CHANNEL_1,HAL_ADC_DEC_256);
  Hum = (int)Humitat;
  CastHum = (uint8 *) &Hum;
  VSend[nou+2] = *CastHum;
  VSend[nou +3] = *(CastHum+1);


 Battery=HalAdcRead(HAL_ADC_CHANNEL_3,HAL_ADC_DEC_256);
  Batteryfloat = (float) Battery;
  Batteryfloat = Batteryfloat*100/3300;
  Batteryint = (int) Batteryfloat;
  VSend[nou+4] = Batteryint;

  nou = nou+5;

   /* Sensat sobreescrivint*/

  if (vell!=0){

      vell = vell+5;
      HalLedBlink(HAL_LED_ALL, 2, 50, TIME_LED_ON);

 }
  osal_start_timerEx( MSA_TaskId, MSA_SENSAT, 60000);
```

Com s'observa al codi es disposa de 2 punters (vell i nou) que són els encarregats de recorre el bloc de mostres Vsend de 990 bytes de tamany. Aquests 990 bytes són el màxim tamany que ens permet utilitzar la memòria de dades.

Són aquests 2 punters els que ens permetran sobreescriure les dades.

Primer es fa una primera volta de sensat on només va augmentant el punter anomenat nou, aquest punter és el que marca la posició de memòria del bloc de mostres on s'ha escrit l'última mesura obtinguda. Quan el punter nou arriba al valor 990, és a dir, s'ha omplert el bloc de mostres, és quan comença a augmentar el punter anomenat vell.
Aquest augment del segon punter serveix per marcar la posició on es troba la dada més antiga ja que les dades noves entrants aniran sobreescrivint les primeres posicions del bloc de mostres.

Segons el disseny del protocol de sensat, cada mesura de dades són 5 bytes: 2 bytes pel valor de la temperatura, 2 bytes pel valor de la humitat i 1 byte pel valor de la bateria. És per això que els punters augmenten el seu valor de 5 en 5.

Tant pel valor de temperatura com d'humitat calen 2 bytes per tal de poder codificar la mostra, és necessari perquè la precisió que es vol aconseguir enviar no es viable amb els 256 valors possibles que dona 1 sol byte.

Per exemple, si el valor obtingut de temperatura és 25.34 ºC, es multiplicarà aquesta variable tipus float per 100 i es passarà a tipus int, convertint el valor en 2534. Aquest valor no es pot codificar en 1 byte, és aquest el motiu pel qual es fan servir 2 bytes per enviar tant la mesura de temperatura com d'humitat.

Pel valor de la bateria només cal 1 byte ja que el aquest valor s'expressarà en tant per cent amb nombres enters del 0 fins al 100, per tant, amb un byte que ofereix 256 possibilitats de valor assolim les necessitats.

Com s'observa al codi la rutina de sensat es va cridant cada 60000ms, és a dir, es prenen mostres cada minut. És per això que les dades es començaran a sobreescriure quan passin 3 hores i 18 minuts: 5bytes/minut per emplenar un bloc de mostres de 990 bytes

7.2 Comunicació entre nodes. Protocol d'enviament

Per comunicar els nodes sensors amb el node gateway s'utilitza l'estàndard IEE 802.15.4. La capa NAL aprofita els serveis del IEEE 802.15.4. Aquesta capa ens permet controlar la interfície ràdio i ens permet enviar i rebre missatges.

La capa NAL únicament envia vectors de tipus byte, i té capacitat per enviar blocs de com a màxim 92 bytes.

En el disseny del protocol d'enviament entre nodes s'aprofitaran aquests 92 bytes màxims de dades fent missatges del tipus:

Figura 7.2 Missatge de resposta de mostres

El primer byte serveix per identificar amb un número de l'1 al 11 quina porció del bloc de mostres és la que s'està enviant, ja que sabent que el nostre bloc de mostres és de 990 bytes, si enviem aquest tipus de missatges, es podran enviar un total d'11.

El segon byte, que tindrà valor 0 o 1, segons si és o no l'últim paquet, serveix per identificar quantes porcions del bloc de mostres s'han enviat. Així després al rebre-les sabem si la transmissió de porcions ha sigut correcte.

La tasca anomenada MSA_ENVIANT és la que s'encarrega de formar els missatges de resposta de mostres i configura els camps de control del a capçalera.

```
    if (events & MSA_ENVIANT)
  {
```

```
    for (int k=0; k<90; k+=5){
      if(vell==990){
        vell=0;
      }
      if (vell==nou){
        break;
      }
      Tamany[0]=id;
      Tamany[1]=ult;
      Tamany[k+2] = VSend [vell];
      Tamany[k+3] = VSend [vell+1];
      Tamany[k+4] = VSend [vell+2];
      Tamany[k+5] = VSend [vell+3];
      Tamany[k+6] = VSend [vell+4];
      vell+=5;
    }
    if (vell!=nou){
    NAL_Send(MSG,0xFF,(uint8*)Tamany,sizeof(Tamany));
    HalLedBlink(HAL_LED_ALL, 1, 50, TIME_LED_ON);
    id++;
    osal_start_timerEx(MSA_TaskId, MSA_ENVIANT,1000);
    }
      if(vell==nou){
         ult = 1;
         Tamany [0]=id;
         Tamany [1] = ult;
         HalLedBlink(HAL_LED_ALL, 1, 50, TIME_LED_ON);
         NAL_Send(MSG,0xFF,(uint8*)Tamany,sizeof(Tamany));
         for(int k=0; k<990; k++){
         VSend[k]=0;
         }
         vell=0;
         nou=0;

      }

    return events ^ MSA_ENVIANT;
  }
```

Com s'observa al codi es van enviant, amb la comanda NAL_Send, missatges de 92 bytes. La condició que ens fa sortir del bucle de generació de missatges és si vell=nou, és a dir, si els 2 punters es troben a la mateixa posició del bloc de mostres indicant que s'ha recorregut sencer.
Quan aquesta condició és compleix, l'últim missatge de resposta de mostres enviat es marca amb el segon byte a 1, sabent així que és la última porció del bloc de mostres.

L'enviament de missatges per part del node sensor es produeix quan el node gateway li envia el missatge de petició de mostres, la tasca de rebre missatges s'anomena SYS_EVENT_MSG. Aquesta tasca s'implementa amb el codi següent:

```
if (events & SYS_EVENT_MSG)
  {
    pMsg = osal_msg_receive(MSA_TaskId);

    switch ( *pMsg )
    {
    case MSG_RECEIVED:
      pData = (nalCbackEvent_t *)pMsg;

      dato0=pData->data[0];

      if(dato0==33)
      {
        HalLedSet (HAL_LED_ALL, HAL_LED_MODE_OFF);
       osal_start_timerEx(MSA_TaskId, MSA_ENVIANT, 1000);//
Inicialitzem l'event de transmissió de paquets.
      }
      break;

    default:
      break;
    }

    osal_msg_deallocate((uint8 *) pMsg);

    return events ^ SYS_EVENT_MSG;
  }
```

Com s'observa al codi, quan el node rep un missatge de petició de mostres crida a l'event MSA_ENVIANT anteriorment descrit.

7.3 Tractament de les dades al node gateway

El node gateway és l'encarregat d'iniciar la comunicació enviant el missatge de petició de mostres. Aquest missatge és un sol byte amb valor 33 i s'observa a la Figura 7.1.

Figura 7.1 missatge de petició de mostres

Aquesta acció es realitza periòdicament des de que s'inicia el dispositiu. S'envia un missatge de petició de mostres cada 50ms. Aquesta tasca s'implementa amb el codi següent:

```
if (events & MSA_PETICIO)
  {
    uint8 envio[1] = {33}
```

```
    set_radio(RX_ON);
    msa_State = MSA_SEND_STATE;
    NAL_Send(MSG,0xFF,(uint8*)envio,sizeof(envio));
    osal_start_timerEx( MSA_TaskId, MSA_PETICIO, 50);
  }
```

Com s'observa al codi, mitjançant la comanda NAL_Send, el valor 33 que està esperant el node sensor per iniciar la comunicació és enviat cada 50ms.

Per altra banda el node gateway és l'encarregat de rebre els missatges de resposta de mostres enviats pel node sensor. El node gateway no fa cap processament de dades, simplement guarda les dades rebudes en un bloc de mostres del mateix tamany que el del node sensor anomenat VSendR, augmenta la variable paquetsrebuts segons els paquets que rebi i iguala el valor de la variable ultimpaquet amb el de l'identificador de l'últim missatge rebut.
La rebuda de dades i la seva col·locació al vector VSendR s'observa en aquest codi:

```
if (events & SYS_EVENT_MSG)
  {
    pMsg = osal_msg_receive(MSA_TaskId);

    switch ( *pMsg )
    {
    case MSG_RECEIVED:
      pData = (nalCbackEvent_t *)pMsg;

      dato0=pData->data[0];
      dato1=pData->data[1];
      Source=pData->SrcAddress;

      if (dato0 == 1){
            HalLedSet (HAL_LED_ALL, HAL_LED_MODE_OFF);
            HalLedBlink(HAL_LED_4, 1, 50, TIME_LED_ON);
            paquetsrebuts++;
            origen=(int)Source;
            if (dato1==1){
              ultimpaquet=1;
           osal_start_timerEx(MSA_TaskId, MSA_UART, 1000);
            }
            for (int w=0; w<90; w++){
              VSendR [w]= pData->data[w+2];
            }
      }
```

En el codi anterior només es visualitza la rebuda del primer missatge o porció del bloc de mostres. Es fa el mateix per les 11 porcions.

Quan es reben tots els missatges es procedeix a enviar-los al PC mitjançant el port sèrie. Hi han 2 tasques a executar pel UART:

La primera envia un vector de 3 bytes anomenat Control, que conté l'ID del node del qual s'han rebut les dades, el nombre de paquets rebuts i el valor identificatiu de l'últim paquet. El codi d'aquest enviament és:

```
 if (events & MSA_UART)
{
  control[0]=paquetsrebuts;
  control[1]=ultimpaquet;
  control[2]=origen;
  HalUARTWrite(0,control,3);

  osal_start_timerEx( MSA_TaskId, MSA_UART2, 1000);
  return events ^ MSA_UART;
}
```

Un cop s'ha enviat aquest bloc de control es crida a la tasca MSA_UART2, que és l'encarregada d'enviar el bloc de mostres rebut. No envia els 990 bytes de cop sinó que els va dividint en paquets de 90 bytes. El codi que realitza aquesta funció és:

```
 if (events & MSA_UART2)
{
 for(i=1; i<=paquetsrebuts; i++){
  for (int e=0; e<90; e++){
  uart[e]=VSendR[e+k];
  }
  HalUARTWrite(0,uart,90);
  k += 90;
 }
 return events ^ MSA_UART2;
}
```

Com s'observa al codi s'envien tants paquets de 90 bytes com missatges de resposta de mostres s'han rebut.

Les tasques que envien les mostres al PC mitjançant el port sèrie són activades quan arriba l'últim missatge de mostres. S'ha de contemplar la possibilitat de que aquest missatge no arribi. És per això que s'ha dissenyat una tasca que s'encarrega de cridar a la rutina que envia les dades pel port sèrie si després d'un segon no s'ha rebut cap missatge identificat com a últim. Aquesta tasca està implementada en el codi següent:

```
 if (events & MSA_TIMEOUT)
{
   if(ultimpaquet == 0){
     osal_start_timerEx(MSA_TaskId, MSA_UART, 1000);
   }
}
```

7.3.1 Configuració del UART

Per poder enviar els paquets pel UART[7] prèviament s'ha de configurar. En aquest apartat es vol fer una breu descripció de com fer-ho.

Simplement s'ha de definir en el *main* del programa els paràmetres que es desitgin i obrir el port mitjançant la comanda HalUARTOpen:

```
halUARTCfg_t uartConfig;

uartConfig.configured           = TRUE;
uartConfig.baudRate             = HAL_UART_BR_38400;
uartConfig.flowControl          = TRUE;
uartConfig.flowControlThreshold = 120;
uartConfig.rx.maxBufSize        = 120;
uartConfig.tx.maxBufSize        = 120;
uartConfig.idleTimeout          = 6;
uartConfig.intEnable             = TRUE;
uartConfig.callBackFunc         = MT_UartProcessRxData;

  HalUARTOpen(0, &uartConfig);
```

La comanda que s'utilitza per escriure dades pel port sèrie és:

```
HalUARTWrite(port,vector a enviar, longitud);
```

7.4 Circuit d'adaptació per transmetre les dades al PC

El nostre node sensor transmet en nivells lògics TTL (entre 0 i 5 volts) i en canvi, al nostre PC han d'arribar nivells lògics de tensió RS232 (entre -15 i 15 volts). És per això que s'ha implementat el circuit que es pot observar a la figura 7.3. El xip MAX3232[8] fa la funció conversora de TTL a RS232.

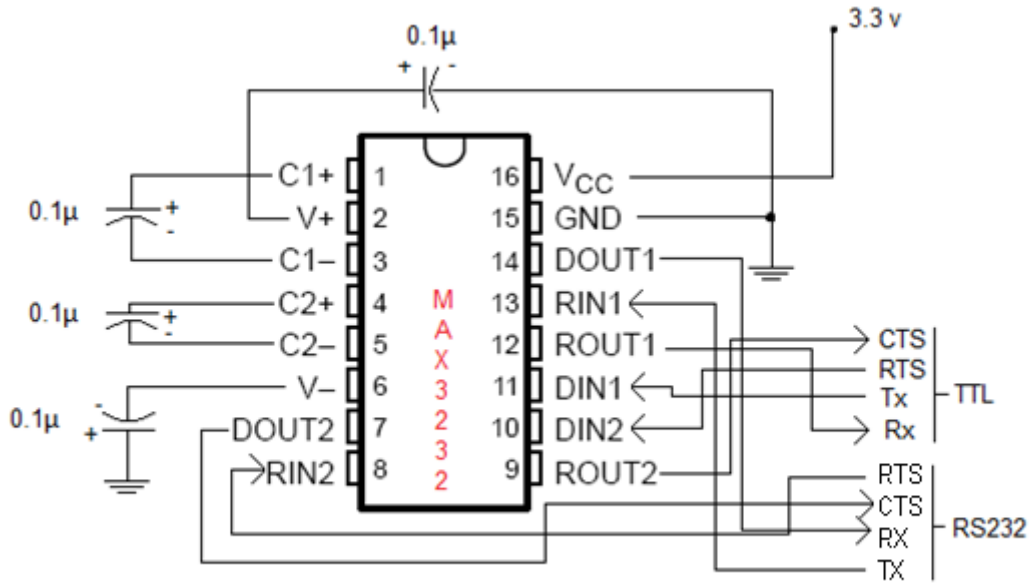


Figura 7.3 Circuit d'adaptació de TTL a RS232

A la sortida del MAX3232 es fa un creuament. CTS i RTS es creuen entre si i RX i TX entre si. Així mitjançant un cable sèrie directe podem connectar amb un PC.
El circuit d'adaptació soldat s'observa a la figura 7.4

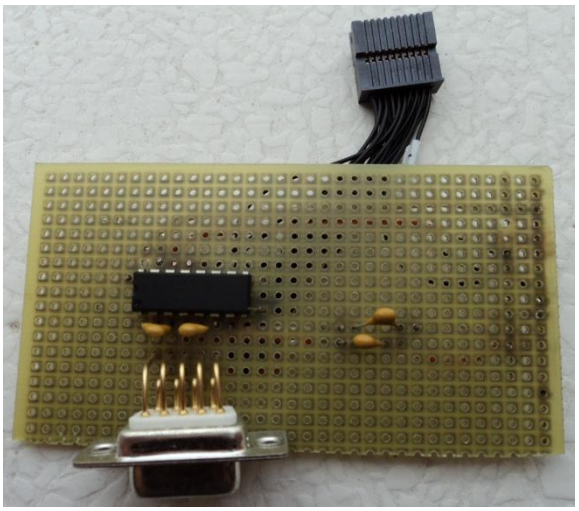


Figura 7.4 Circuit d'adaptació real.

7.5 Recepció de les dades al PC i creació de les gràfiques

Un cop rebudes les dades al PC mitjançant el port sèrie, aquestes es processen mitjançant una petita aplicació de consola realitzada amb el Microsoft Visual Studio 2008 i programada en C#.

Aquesta aplicació llegeix els paquets rebuts. Només mostra per pantalla, com s'observa a la Figura 7.5, el paquet de control. Així l'usuari sap de quin node provenen les dades i quants paquets s'han rebut correctament.

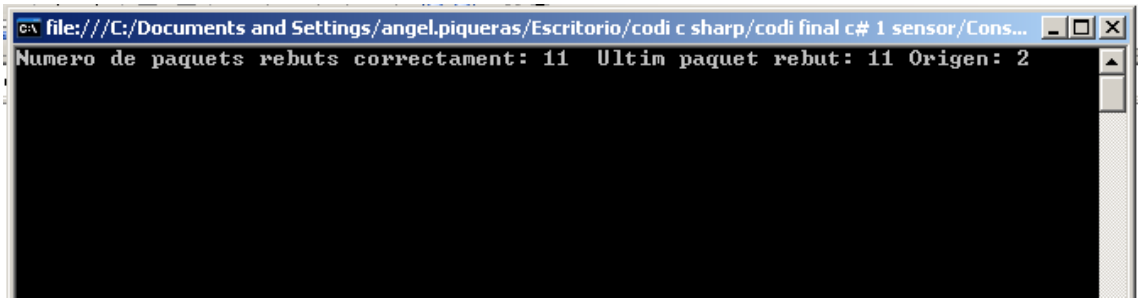


Figura 7.5 Consola al rebre el paquet de control

Amb la resta de les dades el que fa és omplir un fitxer del tipus .txt. Les dades es guarden amb el format: Data + Temperatura + Humitat + Bateria. Un exemple de línia d'aquest fitxer és:

10/3/2010 11:24:09 AM;27.64;99.17242;69

Per mostrar els valors al gràfic s'ha escollit el software *Line & Area* d'*amCharts.com[9].* Està format de varis arxius, i ofereix un alt nivell de personalització.

A continuació es comenten els arxius més importants, són els que s'han modificat per aconseguir el resultat desitjat.

- **amline.html:** És l'arxiu principal, i és el que visualitzem a la plana web. Hi ha els paràmetres principals del gràfic (nom, tamany...). També s'hi troba el format de les dades que ha de mostrar, el directori de l'arxiu de dades i d'opcions.

- **amline_data.txt:** És l'arxiu un es troben les dades que seran mostrades al gràfic. En aquest cas s'utilitza aquest fitxer perquè a l'arxiu amline.html s'ha configurat que llegeixi les dades en format .csv (separat per comes) de l'arxiu .txt. Aquesta opció s'ha modificat, ja que per defecte es llegeixen les dades en format .html i llavors el fitxer de les dades seria el amline_data.xml.
  L'arxiu .txt del qual es llegeixen les dades és el que s'ha anat omplint amb l'aplicació en C#

- **amline_seetings.xml:** A aquest fitxer s'hi poden trobar tots els paràmetres de configuració del gràfic: colors, llegendes, posició, possibilitat d'amagar o mostrar una o vàries línies... D'aquest fitxer, s'han modificat alguns aspectes bàsics de l'aparença del gràfic, el zoom, refresc automàtic de dades, etiquetes, llegenda i el format de les dades.

Una gràfica d'exemple és la que observem a la figura 7.6:

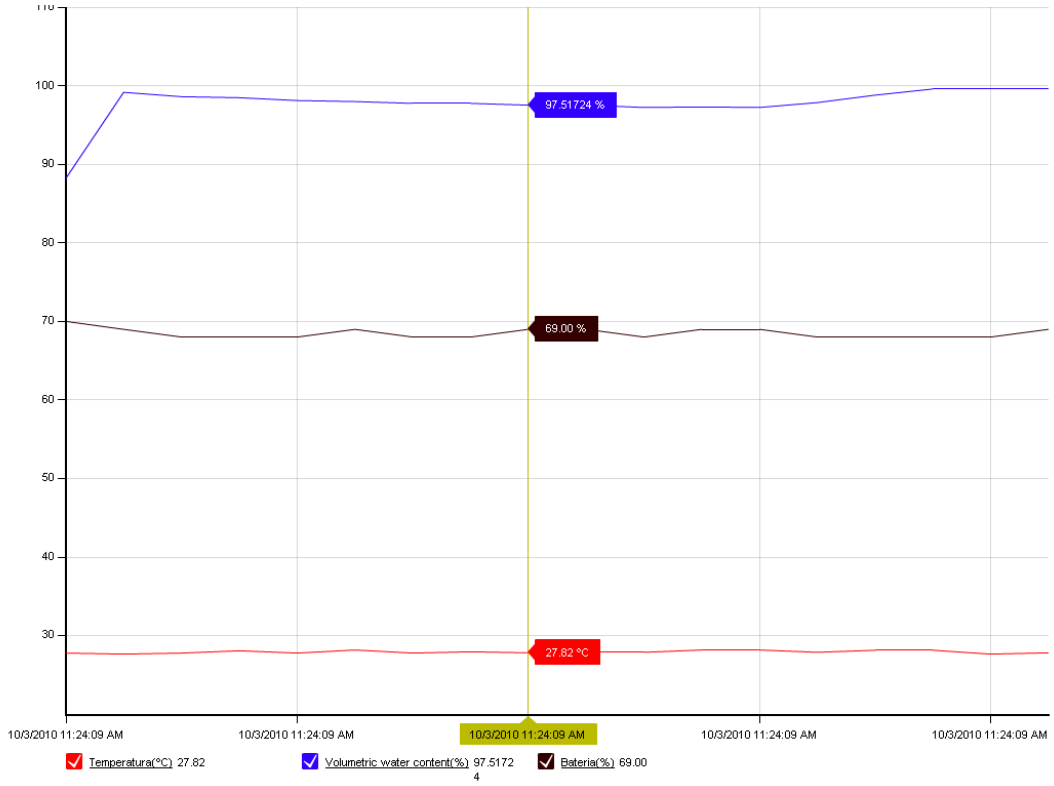


Figura 7.6 Gràfica que conté els valors de temperatura, humitat i bateria

CAPÍTOL 8. CONCLUSIONS I LINIES FUTURES

8.1 Conclusions

En aquest treball fi de carrera s'ha realitzat una xarxa de sensors oportunística que recull les dades de temperatura, humitat i bateria.

S'ha realitzat una adaptació per acoblar un sensor d'humitat extern.

S'ha realitzat un circuit d'adaptació conversor de TTL a RS-232 per poder enviar les dades recollides pel node CC2430N.

Es mostren les dades a través d'una interfície web.

Per tal de minimitzar el consum d'aquests nodes, s'ha escollit els components i la configuració dels nodes amb la finalitat que el consum global sigui mínim.

S'ha utilitzat satisfactòriament la capa NAL per controlar la interfície ràdio.

S'ha dissenyat i implementat protocol d'enviament de missatges entre el node sensor i el gateway que funciona correctament.

S'ha assolit l'escenari plantejat a la Figura 1.1

8.2 Línies futures

Aquest projecte podria ser la primera part d'una llarga sèrie de projectes ja que es podrien millorar diverses qüestions.

- Optimització del codi: Eliminar les parts d'aplicació del CC2430 que no es facin servir mai per tal de guanyar en espai de memòria.

- Acabat del Hardware: Crear interfícies conjuntes que englobin tots els circuits d'acoblament i els nodes per tal de facilitar una comercialització.

- Mesura del temps exacte: Es podria millorar molt l'eina realitzada en aquest projecte si es fos capaç de calcular la hora exacta mitjançant els timers interns del CC2430. Es va intentar però no s'ha assolit.

- Estudi profund del consum: S'ha dissenyat aquest projecte minimitzant al màxim els consums d'energia. Es podria fer un estudi per determinar la vida útil exacte de les bateries.

- Implementació d'un *scan* pel node gateway: Es podria realitzar una funció que fos capaç de retornar al ordinador quins nodes estan dintre de la cobertura del coordinador i que l'usuari pogués escollir de quin dels sensors del seu rang de cobertura vol rebre les dades.

- Protocol de reenviament: Utilitzar el protocol de reenviament dissenyat per Eric Peris Bespín[10] per tal de poder ampliar la xarxa de sensat ja que encara que el node gateway només tingues accés a un sol node, aquest podria haver rebut totes les dades dels demés.

- Treball de camp: Escollir escenaris per tal de provar veritablement el funcionament de l'eina realitzada en aquest TFC.

BIBLIOGRAFIA

[1] Estàndard IEEE 802.15.4:
http://standards.ieee.org/getieee802/download/802.15.4a-2007.pdf

[2] CC2430 Datasheet:
http://focus.ti.com/lit/ds/swrs036f/swrs036f.pdf

[3] Revisió 1.7 del Development Kit de Texas Instruments.
http://www.ti.com/litv/pdf/swru133

[4] Modes d'energia del CC2430:
http://focus.ti.com/lit/an/swra162b/swra162b.pdf

[5] Sensor de temperatura intern del CC2430
http://focus.ti.com/general/docs/lit/getliterature.tsp?literatureNumber=swra101a&fileType=pdf

[6] Sensor d'humitat EC-5 de Decagon Devices:
http://www.decagon.com/pdfs/manuals/EC-20_EC-10_EC-5_v9.pdf

[7] Com configurar el UART, DN112:
focus.ti.com/lit/an/swra222b/swra222b.pdf

[8] Datasheet MAX3232:
http://www.datasheetcatalog.com/datasheets_pdf/M/A/X/3/MAX3232.shtml

[9] Material descarregable per fer les gràfiques:
http://amcharts.com/download

[10] Peris Bespín, E; *Sistema de difusión para CC2430*