



**Escola Politècnica Superior
de Castelldefels**

UNIVERSITAT POLITÈCNICA DE CATALUNYA

TREBALL DE FI DE CARRERA

TÍTOL DEL TFC: Implementació d'una solució domòtica basada en sensors (II)

TITULACIÓ: Enginyeria Tècnica de Telecomunicació, especialitat Sistemes de Telecomunicació

**AUTORS: Adrià Dasquens Carulla
Anna Ortiz Mazarico**

**DIRECTORS: Lluís Casals Ibáñez
Rafael Vidal Ferré**

DATA: 2 de febrer de 2010

Títol: Implementació d'una solució domòtica basada en sensors (II)

Autors: Adrià Dasquens Carulla
Anna Ortiz Mazarico

Director: Lluís Casals Ibáñez
Rafael Vidal Ferré

Data: 2 de febrer de 2010

Resum

Aquest projecte parteix d'un treball final de carrera anterior, dedicat a la configuració d'un node sensor i un altre controlador mitjançant l'acoblament d'uns mòduls d'infraroig. Un cop establerta la configuració inicial, aconseguia transmetre una ordre al sensor.

L'objectiu del present TFC és la implementació d'una xarxa de nodes sensors, utilitzant un sensor dissenyat pel grup de recerca de xarxes sense fils de la UPC i el disseny i realització dels mòduls utilitzats en el projecte anterior, l'emissor i receptor d'infraroig, així com altres mòduls necessaris per a la interacció amb l'usuari com seria un teclat.

Per tal d'establir la comunicació entre els diferents nodes encara que no tinguin comunicació directa entre ells, s'ha creat un protocol de reenviament de missatges.

El disseny d'aquesta xarxa s'ha utilitzat per tal d'obtenir els valors de temperatura dels nodes sensors. A fi de mostrar de forma visual l'evolució d'aquesta variable, s'ha creat una interfície web on es pugen les dades via FTP, i on es pot visualitzar aquesta gràfica a través de qualsevol navegador web.

Title: Deployment of a domotic solution based on sensors (II)

Author: Adrià Dasquens Carulla
Anna Ortiz Mazarico

Director: Lluís Casals Ibáñez
Rafael Vidal Ferré

Date: February, 2nd 2010

Overview

The project departs from a final project done before, dedicated to the configuration of a sensor node and another driver with the coupling of some infrared modules. Once established the initial setup, it was able to transmit an order to the sensor.

The aim of this TFC is deploying a network of sensor nodes using a sensor designed by the research group of wireless networks of UPC and the design and implementation of modules used in the previous project, the sender and receiver infrared, and other modules necessary for the interaction with the user like a keyboard.

To establish communication between different nodes if they have no direct communication between them, have created a protocol for message forwarding.

The design of this network was used to obtain the values of temperature sensor nodes. To visually show the evolution of this variable, we have created a web interface where the data are uploaded via FTP, and where you can see this graphically through any Web browser.

AGRAÏMENTS

Volem agrair als nostres directors Lluís Casals i Rafael Vidal tot el suport que ens han donat al llarg d'aquests mesos, i per la seva implicació i esforç dedicat en la realització d'aquest TFC.

També agrair a les nostres famílies i amistats la seva ajuda i comprensió.

ÍNDIX

CAPÍTOL 1. INTRODUCCIÓ	1
1.1. Origen.....	1
1.2. Objectius.....	2
CAPÍTOL 2. REQUERIMENTS	5
CAPÍTOL 3. DISSENY DEL HARDWARE	7
3.1. Característiques generals	7
3.1.1. Tipus de connectors	7
3.1.2. Pins.....	9
3.1.3. Alimentació.....	9
3.2. Software utilitzat	10
3.3. Emissor d'infrarojos	12
3.4. Receptor d'infrarojos.....	15
3.5. Placa de connexions.....	17
3.6. Teclat.....	18
CAPÍTOL 4. OBTENCIÓ I REPRESENTACIÓ DE LA TEMPERATURA	25
4.1. Esquema	25
4.2. Obtenció de la temperatura	26
4.2.1. ADC	26
4.2.2. Càlculs previs.....	27
4.2.3. Calibració.....	28
4.2.4. Codi utilitzat.....	31
4.3. Node <i>gateway</i> : comunicació port sèrie-pc.....	32
4.3.1. Enviament de dades	32
4.3.2. Recepció de dades	35
4.4. Casos particulars de funcionament.....	36
4.4.1. Un node.....	36
4.4.2. Diversos nodes	37
4.5. Tractament de les dades	44
4.5.1. Diagrama.....	44
4.5.2. Codi C#.....	45
4.5.3. Interfície web	46
CAPÍTOL 5. CONSUM	51

5.1. Agilent N6705A.....	51
5.2. Mesures.....	52
5.2.1. Transmissor.....	55
5.2.2. Receptor.....	56
5.2.3. Conclusions de les verificacions	57
CAPÍTOL 6. CONCLUSIONS	59
6.1. Línies futures.....	59
BIBLIOGRAFIA I REFERÈNCIES	61
ANNEX	63

CAPÍTOL 1. INTRODUCCIÓ

En els darrers temps s'ha evolucionat molt ràpidament en l'àrea de les xarxes de sensors sense fils. Aquestes disposen d'unes característiques que les fan molt atractives, com poden ser l'absència de cablejat, i una progressiva reducció de les mides dels seus components, així com del seu consum i del seu cost econòmic.

Tot això fa que ofereixin una gran versatilitat i facilitat d'integració en qualsevol entorn.

Existeix una certa diversitat de tecnologies que comparteixen les característiques abans esmentades i que tenen diferents graus d'especialització (per ex. la domòtica o les xarxes industrials) . D'entre elles es pot destacar l'estàndard Zigbee (IEEE802.15.4) [1], una de les tecnologies més utilitzades i que compta amb més consens.

Es caracteritza pel seu abast (com a màxim 100 m), i la baixa velocitat de transmissió de dades, unes característiques suficients per a moltes aplicacions i que van acompanyades d'un consum molt baix. Aquestes especificacions s'ajusten a les necessitats de la xarxa que es vol implementar en aquest projecte, una xarxa de sensors amb aplicacions domòtiques. Per aquest motiu, IEEE802.15.4 serà la tecnologia ràdio utilitzada en aquest projecte.

1.1. Origen

L'origen d'aquest treball fi de carrera (TFC) és un TFC anterior de títol: "Implementació d'una solució domòtica basada en sensors", d'Eric Peris [2].

Aquest treball va ser una primera aproximació a un sistema domòtic basat en una xarxa de sensors, formada per uns nodes es poden configurar fàcilment a través d'un comandament a distància, que utilitza el protocol d'infrarojos RC5, conjuntament amb el protocol radio IEEE802.15.4.

Això es va aconseguir mitjançant la utilització del dispositiu CC2430 de Texas Instruments [3] [4], que fa ús de la tecnologia ràdio 802.15.4. Un d'aquests dispositius es va configurar com a controlador i l'altre com a node sensor.

Per tal de configurar aquests dispositius se'ls va afegir una interfície de comunicació per infrarojos. Al controlador se li va acoblar un emissor, i al node sensor, un receptor.

Finalment, en el citat TFC també s'introduïa l'entorn de programació que s'utilitza i l'estructura de la pila de software TIMAC que proporciona el fabricant del dispositiu.

Les limitacions d'aquesta primera aproximació són evidents. Es van dissenyar l'emissor i receptor d'infrarojos, però la implementació es va fer en una placa de

proves *protoboard*. Els sensors es van configurar manualment mitjançant una petita modificació en el codi, ja que no es disposava d'un comandament implementat físicament per poder-los configurar.

1.2. Objectius

En la figura 1.1 es mostra l'escenari en el que es pretén desenvolupar la xarxa de sensors i els elements que n'hauran de formar part, així com la seva distribució. Prenent aquesta figura com a referència, a continuació es van explicant els objectius que es volen assolir.

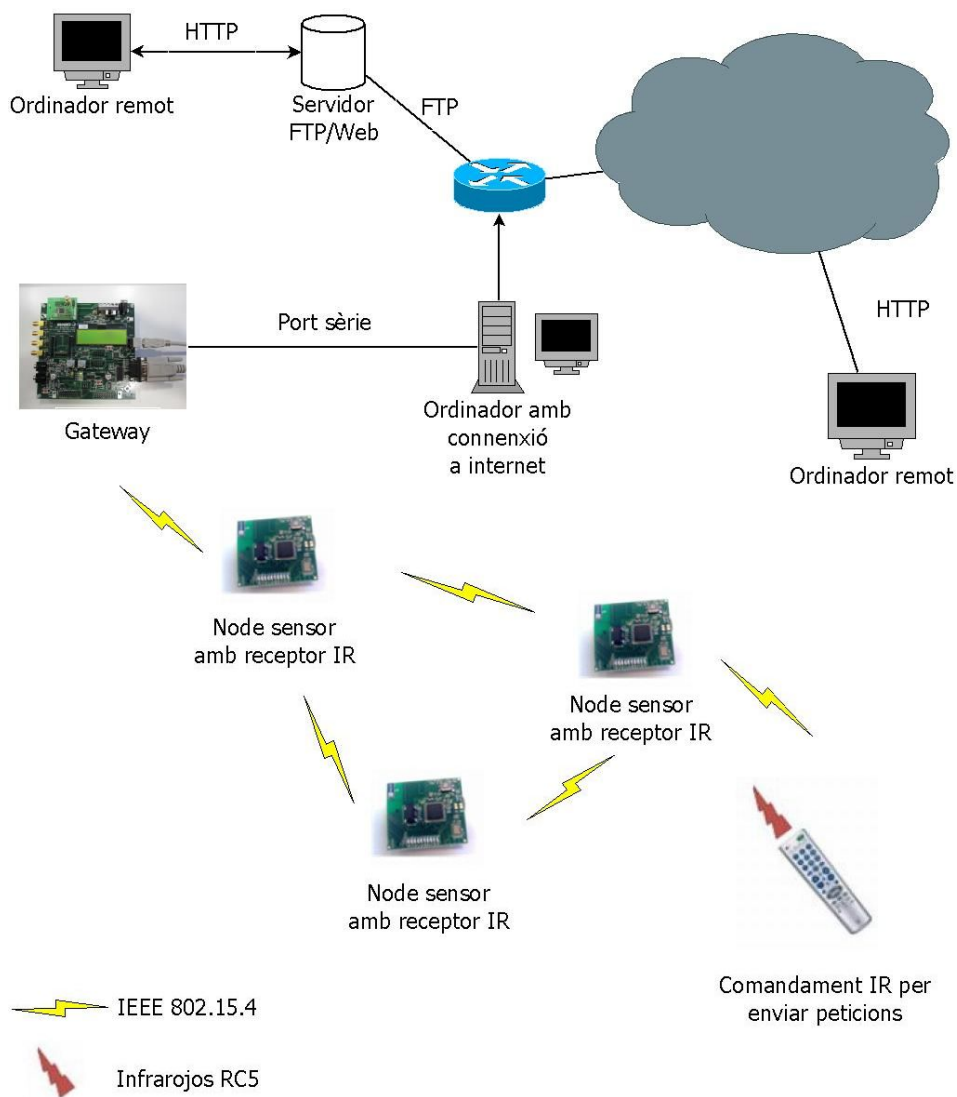


Fig. 1.1 Representació de l'escenari

Un dels objectius del present TFC és la implementació del comandament i dels nodes sensors, per tal que funcionin de manera autònoma en un disseny compacte, i de reduïdes dimensions. Per això s'han de dissenyar els circuits anteriorment esmentats en una placa de circuit imprès.

També serà necessari el disseny d'altres plaques com la que ha de contenir el teclat, que anirà acoblat al sensor i a l'emissor d'infrarojos. Aquesta servirà per a facilitar la interacció amb l'usuari.

Un aspecte directament relacionat a l'autonomia dels nodes és el consum. Per tal d'allargar al màxim la vida útil d'aquests, a l'hora d'escollir els components s'ha de tenir en compte que consumeixin el mínim d'energia i configurar els sensors en mode baix consum.

Un altre dels objectius és la utilització d'un node dissenyat pel Grup de recerca de Xarxes Sense Fils de la UPC, que utilitza el mateix xip. Aquest node conté una pila de 3V que fa que aquest pugui ser independent de la placa de desenvolupament.

El software que es desenvolupava al treball fi de carrera precedent, s'ha d'utilitzar com a referència, i ampliar-lo per tal d'acollir les noves funcionalitats. Aquestes passaran per recollir informació dels sensors i interactuar amb ells.

Com que s'utilitzarà el mateix xip, el codi es podrà re aprofitar, a la vegada que es podrà desenvolupar-ne de nou amb el mateix entorn.

Les possibilitats del software quedaran acotades en funció del hardware que es pugui implementar. En aquest sentit es pensa en un disseny prou flexible per poder adaptar diversos mòduls. Concretament es pensa incorporar un mòdul de mesura i control del pas del corrent elèctric i, els citats mòduls d'infraroig per a controlar la configuració. A les possibilitats que obren aquests mòduls cal afegir la mesura de la temperatura, gràcies a un sensor que incorpora el xip utilitzat en aquest projecte

Un altre objectiu és que els nodes s'han de comunicar entre ells de manera que no sigui necessària una connexió de tipus estrella (un node central al que tots els altres estan connectats). Per aquest motiu s'optarà per una solució totalment distribuïda. Això implica que hi haurà d'haver un protocol de reenviament de trames entre els nodes de manera que la informació arribi sense cap problema al destinatari, sigui quin sigui el node d'origen i el node destí, i encara que hagi de passar per diversos nodes intermedis.

També s'haurà de poder visualitzar les dades recollides des nodes sensors i interactuar amb ells a través d'una interfície web. Així s'hi podrà accedir des de qualsevol plataforma, aportant molta més flexibilitat per mostrar la informació i interactuar amb el sistema. Caldrà fer accessible aquesta interfície web per xarxa, ja sigui la LAN de la casa on està instal·lada la xarxa de sensors o des d'Internet. Això motivarà la presència d'un node especial, el *gateway*, que connectarà la xarxa de sensors a un ordinador que farà d'intermediari entre la xarxa de sensors i el servidor web.

El present document es dividirà en sis capítols diferents. En aquest capítol s'ha fet una breu introducció del projecte així com una descripció dels objectius establerts. En el segon capítol es descriuran els requeriments del sistema per tal de complir els objectius esmentats anteriorment. Seguidament, al capítol tres, es descriurà tot el procés de disseny del hardware. Això inclou des de la selecció de components al software utilitzat per a fer el disseny de les plaques. En el quart capítol s'explicarà el procés d'obtenció de la temperatura dels nodes sensors i tot el procés necessari per els seu enviament fins a fer possible la visualització dels valors obtinguts en una gràfica a través d'un servidor web. En el capítol cinc s'analitzarà el consum dels sensors i es compararà amb els valors teòrics que proporciona el fabricant, Texas Instruments. Finalment el treball conclourà amb un apartat amb les conclusions i unes línies futures de treball per a continuar aquest projecte.

CAPÍTOL 2. REQUERIMENTS

Un cop definits els objectius d'aquest TFC, en aquest apartat es detallen els requeriments que haurà de tenir el sistema mostrat en la figura 1.1.

A continuació es mostren les funcions dels elements més destacats de la xarxa:

- Comandament: enviar per infrarojos les comandes als nodes sensors.
- Node sensor: respondre les peticions de l'usuari i reenviament de missatges.
- *Gateway*: actuar d'enllaç entre la xarxa de sensors i el PC.
- Servidor FTP/Web: allotjar les dades recollides pels sensors i mostrar-les a través d'una interfície web.

Els requeriments són els següents:

- L'usuari haurà de poder interactuar directament amb els nodes per ajustar-hi diversos paràmetres (com per exemple l'Identificador, les funcions a realitzar, etc) i per a donar-li ordres.
- Per tal d'establir aquesta comunicació entre l'usuari i el sensor, s'ha de dissenyar el comandament a distància. Aquest ha d'incloure un teclat numèric complet per poder adjudicar els identificadors (Ids) a cada node, i quatre tecles de funció. Les funcions bàsiques que es volen realitzar són les següents: Afegir Id a un node, modificar Id, esborrar Id i actuar s'obre l'element (per exemple si un node esta acoblat a un tristor, mitjançant aquest botó es podrà encendre i apagar l'aparell connectat). El teclat ha de tenir alguna tecla buida, o algun mètode per a que si es necessita ampliar el número de botons, sigui possible.
- Els nodes han de ser autònoms, és a dir, s'han de poder alimentar mitjançant una pila per tal d'ubicar-los als llocs on es desitgi.
- Per aquesta raó, el consum és un aspecte primordial a tenir en compte. Tot el disseny (ajustaments dels paràmetres dels aparells, selecció de components) ha d'estar orientat a la mínima despesa energètica del sistema, per poder aconseguir la màxima autonomia dels nodes.
- El nombre de nodes de la xarxa, ha de ser flexible. La xarxa ha de permetre a l'usuari poder afegir i treure nodes sense cap tipus de configuració addicional, i el sistema d'encaminament, ha de ser capaç de fer arribar cada paquet de dades al seu destí encara que hi hagi

modificacions a la xarxa (aparició de nous nodes, o caiguda de nodes actius anteriorment).

- Cada node sensor ha de tenir la possibilitat d'acoblar algun mòdul addicional. Aquests mòduls han de desenvolupar tasques com, agafar dades d'un teclat extern, enviar i rebre dades per infraroig per tal que l'usuari pugui adreçar-se i configurar els sensors, obrir o tancar un circuit elèctric amb elements connectats a aquest (per exemple activar o desactivar un endoll) , o un mesurador de corrent per mesurar el consum d'uns determinats aparells.
- Per això els circuits emissor i receptor que s'havien provat en una placa *protoboard* ara s'ha de dissenyar el circuit imprès en una placa física per implementar aquests circuits. A més d'aquests últims, també s'han de dissenyar el circuit imprès del teclat i una placa de connexions que s'ocuparà d'unir els diferents mòduls.
- Els nodes es comunicaran entre ells per tal d'enviar-se missatges. Han de poder identificar el destinatari del missatge i si no va adreçat a ell, l'ha de poder reenviar fins que la informació arribi al seu destí. Per això s'ha de crear un protocol de reenviament de missatges.
- El sistema serà accessible per xarxa local o des d'Internet. Per això caldrà que un node faci la funció de *gateway*. Aquest node estarà connectat a un ordinador per cable, i a la vegada amb la resta dels nodes de la xarxa a través de la interfície ràdio.
- S'ha de desenvolupar un entorn web on l'usuari pugui veure la informació de la xarxa i poder-hi dur a terme accions.
- L'usuari ha de poder interactuar amb la xarxa de la manera més senzilla i intuïtiva possible, a través d'un comandament a distància o d'una interfície Web.
- Les funcions que hauria de desenvolupar la xarxa serien:
 - Mesurar el corrent a través d'un mòdul de consum.
 - Activar o desactivar el corrent de l'endoll al qual estigui el node connectat.
 - Mesura instantània de la temperatura d'un sol node.
 - Mesura instantània de la temperatura de tots els nodes.
 - Mesura periòdica de la temperatura.
 - Generar gràfiques i estadístiques.

CAPÍTOL 3. DISSENY DEL HARDWARE

En aquest capítol es vol mostrar com s'han dissenyat les diferents plaques que formen el comandament i els nodes sensors així com algunes particularitats d'aquestes.

Aquesta tasca s'ha dut a terme mitjançant el programa Proteus, que es divideix en dos programes diferents però relacionats entre sí.

Els circuits realitzats són l'emissor d'infrarojos, el receptor d'infrarojos, el teclat i una placa de connexions per tal d'unir mitjançant connectors els diferents elements que formen tant el comandament com els diferents nodes sensors.

3.1. Característiques generals

Com que un dels objectius principals d'aquest projecte és la optimització del consum, s'han escollit tots els components amb el criteri de consum mínim d'energia.

Un altre objectiu ha sigut la mida de les plaques. S'han situat els components i les pistes de tal manera que ocupin el mínim d'espai possible i es pugui aconseguir un dispositiu de dimensions reduïdes tant en el cas del comandament com del node sensor.

Per tal que tots els circuits puguin ser col·locats a qualsevol encapsulat, o subjectats, s'ha afegit a cadascun dels *layouts* uns forats als seus extrems de manera que serveixin de subjecció en un futur.

Per prevenir problemes futurs amb les pistes, s'ha evitat al màxim que aquestes tinguin angles rectes a les seves traçades, ja que és molt més convenient que les corbes siguin el més arrodonides possible.

També és important que les pistes siguin d'una amplada superior a la predefinida, ja que d'aquesta manera es minimitza l'error en les connexions i en el procés de soldat dels components. En alguns trams no ha estat possible degut a que varis trams de les pistes passen entre les potes dels connectors.

A continuació es detallen els tipus de connectors i les característiques més elementals d'aquests.

3.1.1. Tipus de connectors

Per connectar els diferents mòduls o plaques, són necessaris uns connectors determinats.

Per unir la placa de connexions amb les altres s'ha escollit el connector PEC10DAAN (del fabricant Sullins Connector Solutions). Aquest connector consta de 20 *pins*, situats en 2 files de 10 *pins* cadascuna, amb una separació de 2.54 mm entre *pins*. La mida del connector compleix amb el compromís de mida i versatilitat. Interessa que el connector sigui relativament petit de manera que la mida de les plaques no sigui molt gran, però a la vegada la separació entre els pins del connector ha de ser prou gran com per poder-hi passar una pista entre *pin* i *pin* per tal de facilitar el disseny de les plaques.

A l'extrem del connector femella hi ha un cable que és el que fa d'unió entre aquestes com es mostra en la figura següent:

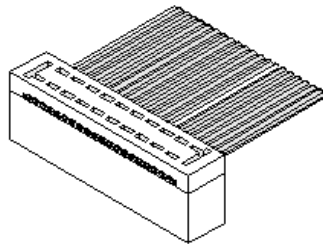


Fig. 3.1 Connector de 10x2 femella (2.54 mm)

Els mòduls on hi ha muntat el xip microcontrolador CC2430 també tenen un connector de 20 *pins* situats en 2 files de 10 *pins* cadascuna a la part inferior de la placa, però la separació entre *pins* és de 1.27 mm. Per tal de connectar el circuit del xip CC2430 amb la placa de connexions i/o altres mòduls, es realitzarà mitjançant un cable, que per un extrem tindrà el connector amb l'espaiat de 2.54 mm i per l'altra banda serà de 1.27 mm i farà la funció de convertidor de connectors.

A la figura següent es mostra el connector mascle de 10x2 amb un espaiat entre pins de 1.27mm que connecta el sensor amb la placa de connexions.

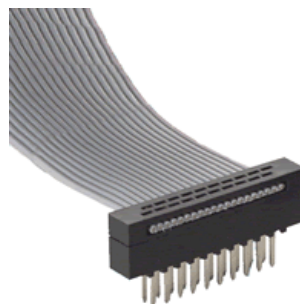


Fig. 3.2 Connector de 10x2 mascle (1.27mm)

Hi ha altres connectors als mòduls de l'emissor d'infrarojos, el receptor i el teclat que s'utilitzen per a l'alimentació d'aquestes.

Són un connector de 3 *pins* i un altre de 2 *pins*, amb una separació entre aquests de 2.54 mm que serveixen per a escollir si l'alimentació de cadascuna li ve de la pila de 3v del sensor o bé d'una alimentació externa.

3.1.2. Pins

Per tal de connectar el sensor (el mòdul basat en el xip CC2430) amb les diferents plaques, hi ha un connector a la part inferior d'aquest, a través del qual es poden transmetre i rebre les ordres pertinents.

En la següent imatge es mostra l'esquema dels pins utilitzats:

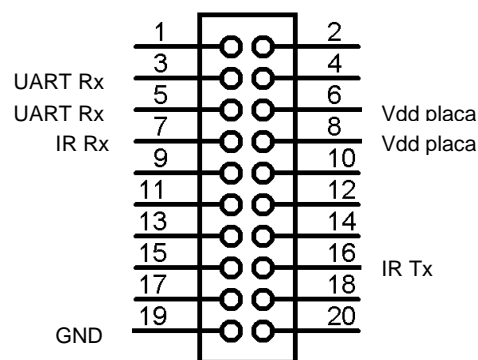


Fig. 3.3 Pins del connector

No tots els pins disponibles s'utilitzen ni en el comandament ni en els nodes sensors però s'ha considerat necessari establir connexió amb cadascun d'ells per a preveure possibles ampliacions de les funcions del sistema.

3.1.3. Alimentació

Pel que fa a l'alimentació de les plaques, s'ha contemplat la possibilitat que fossin alimentades de dues formes diferents. Una és a través de la pila de 3V que conté el sensor i l'altra és mitjançant l'alimentació externa que pot proporcionar una pila i que es connectaria al connector de 2x1 (J2) que hi ha en un extrem de cada placa.

Per tal que les dues alternatives siguin possibles al mateix temps, s'ha col·locat un connector de 3x1 (J4) de manera que pels extrems arribin les dues alimentacions i mitjançant un pont es pugui escollir de forma fàcil quina utilitzar. Pel pin central d'aquest sortiran els 3V que alimentaran tot el circuit, com s'observa en la imatge següent.

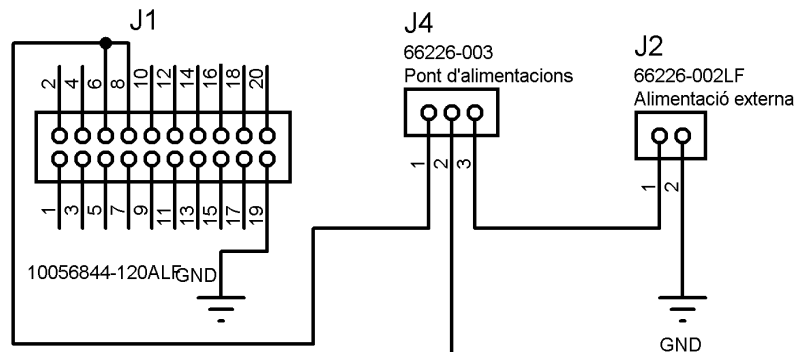


Fig. 3.4 Esquema dels connectors

3.2. Software utilitzat

L'ISIS, una part de l'aplicació Proteus, s'utilitza per a dissenyar els circuits, escollir els components adequats i crear les connexions pertinents entre aquests. Aquest programa també permet la simulació de circuits elèctrics.

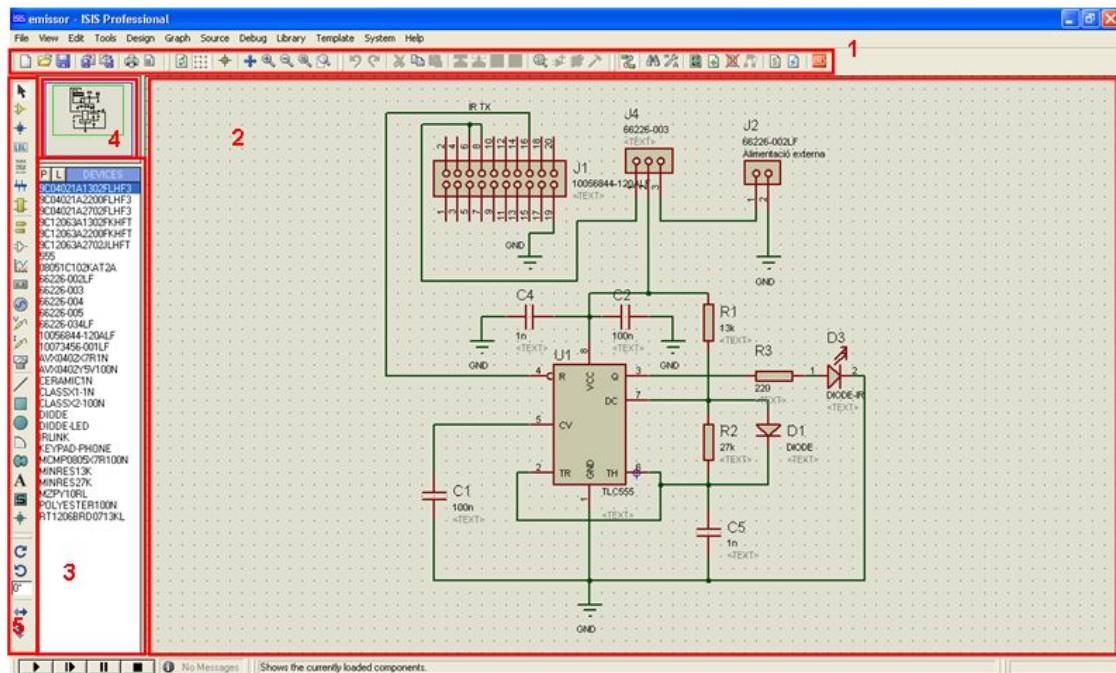


Fig. 3.5 Pantalla principal de l'ISIS

Les àrees principals d'aquest programa són:

1. Barra d'eines bàsiques. Es mostren les opcions més bàsiques del programa, obrir i tancar arxius, guardar canvis, ajustar el zoom...
2. Àrea de disseny del circuit. En aquesta es col·loquen tots els components i s'uneixen segons l'esquema del circuit que es vulgui dissenyar.

3. Llistat de components: Aquí surten tots els components que s'utilitzen en el projecte actual. D'aquesta manera és més fàcil de localitzar-ne algun si es vol tornar a posar. Els 2 botons de la part superior, són per obrir el catàleg de components (que al seleccionar-los s'afegiran a la llista) i les llibreries de components.
4. Aquest visualitzador mostra una vista general del tot el disseny obert. És molt útil quan s'amplia una zona en concret del circuit, per situar quina part del disseny s'està visualitzant. També mostra en detall el component quan és seleccionat de la llista de components (3).
5. Barra d'eines de disseny. A aquesta zona hi ha ubicats els botons d'edició i configuració de circuits. Amb aquests botons podem editar la posició i la configuració dels components i del circuit. també permet dissenyar busos i cables, afegir etiquetes i totes les opcions més bàsiques d'edició i disseny del circuit i dels seus components.

L'ARES, l'altra part que s'ha fet servir de l'aplicació Proteus, utilitza els circuits i els components de l'ISIS per al disseny final del *layout* de la placa de circuit imprès. Amb aquest programa es dissenya la placa, i ja es pot apreciar com quedarà físicament el circuit amb les seves dimensions correctes. Es pot escollir l'encapsulat de cada component, el tipus de pistes que els uniran, capes, forats, dibuixar plans de massa entre moltes altres opcions.

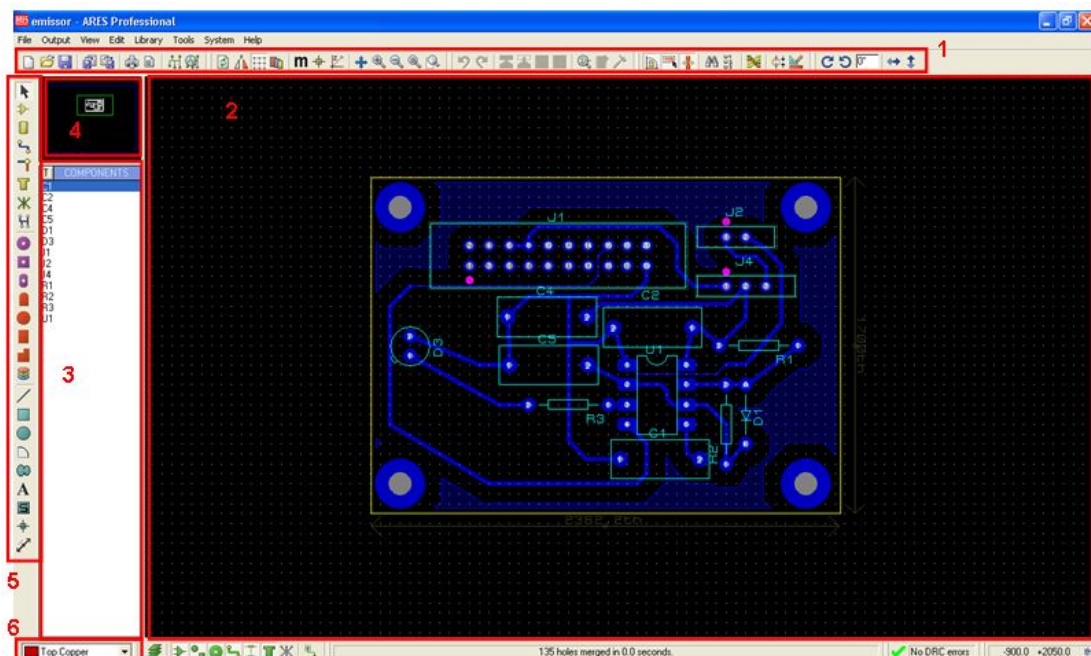


Fig. 3.6 Pantalla principal de l'ARES

1. Barra d'eines bàsiques, igual que a l'ISIS.
2. Àrea de disseny de la placa. Al situar els components, el programa marca amb unes línies on s'ha de connectar cada extrem de cada component, per evitar errors. Aquestes relacions de connexions, són les que s'han configurat quan s'ha dissenyat el circuit amb el ISIS. També es col·loca cada component on interessa, es dissenyen les pistes que els uneixen i el disseny global de la placa.
3. Igual que a l'ISIS, és la llista dels components seleccionats que figuraran al disseny.
4. Visualitzador del circuit. Desenvolupa les mateixes funcions que les explicades anteriorment per l'ISIS.
5. Igual que a l'ISIS, és la barra d'eines del circuit, però en aquest cas no hi ha les opcions de disseny del circuit, hi ha les opcions de disseny de la placa.
6. Selecció de la capa. Aquesta opció és molt important, ja que ajusta a quina capa de la placa va cada pista, component... Les plaques dissenyades són totes d'una sola capa excepte la placa de connexions que és de dues.

3.3. Emissor d'IR

Per al disseny de la placa que conté l'emissor d'IR s'ha utilitzat l'esquema següent:

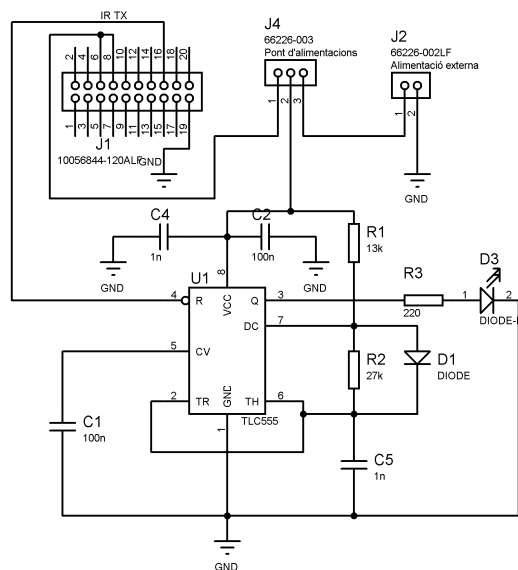


Fig. 3.7 Esquema de l'emissor

Els elements que formen aquest esquema són: un TLC555 [5], un diode emissor d'IR, quatre condensadors i tres resistències.

De tota la varietat que existeix de TLC555 s'ha escollit el model TLC555CP.

Els models es mostren a la taula següent:

Taula. 3.1 Taula comparativa dels TLC555.

AVAILABLE OPTIONS [†]							
PACKAGED DEVICES							
T _A	V _{DD} RANGE	SMALL OUTLINE (D) [‡]	SSOP (DB) [‡]	CHIP CARRIER (FK)	CERAMIC DIP (JG)	PLASTIC DIP (P)	TSSOP (PW) [‡]
0°C to 70°C	2 V to 15 V	TLC555CD	TLC555CDB	—	—	TLC555CP	TLC555CPW
-40°C to 85°C	3 V to 15 V	TLC555ID	—	—	—	TLC555IP	—
-40°C to 125°C	5 V to 15 V	TLC555QD	—	—	—	—	—
-55°C to 125°C	5 V to 15 V	TLC555MD	—	TLC555MFK	TLC555MJG	TLC555MP	—

Un dels motius pels que s'ha escollit aquest és pel tipus d'encapsulat, que és de plàstic (DIP) i té 8 potes. Un altre és el marge de V_{dd} que va d'entre 2 i 15 V, que comprèn el valor al qual s'alimentarà que és de 3V. El rang de temperatures d'aquest va entre 0 i 70°, i com que es tracta d'una aplicació domèstica interior, és suficient.

Una altra característica del TLC555CP és el seu baix consum. Si s'alimenta a 5v consumeix només 1mW.

Serà configurat en mode astable per tal que oscil·li a la freqüència i cicle de treball que el sistema requereix.

El diode emissor d'IR escollit és el LTE-4802. Té el mateix encapsulat que un diode Led qualsevol. Es caracteritza perquè el pic d'emissió de la longitud d'ona és de 940nm.

Els condensadors i resistències utilitzats han sorgit dels càlculs realitzats en el projecte previ a aquest [1].

Els valors resultants són els següents:

R1 = 13 kΩ
 R2 = 27 kΩ
 R3 = 220 Ω
 C1 = 100 nF
 C2 = 100 nF
 C3 = 1 nF
 C4 = 1 nF

S'ha inclòs un condensador de desacoblament de contínua (C2), per tal de reduir els corrents paràsits als pins del TLC555, per la qual cosa s'ha situat tant a prop com ha sigut possible d'aquest.

El disseny final de la placa de l'emissor d'IR és la que es mostra en la figura següent:

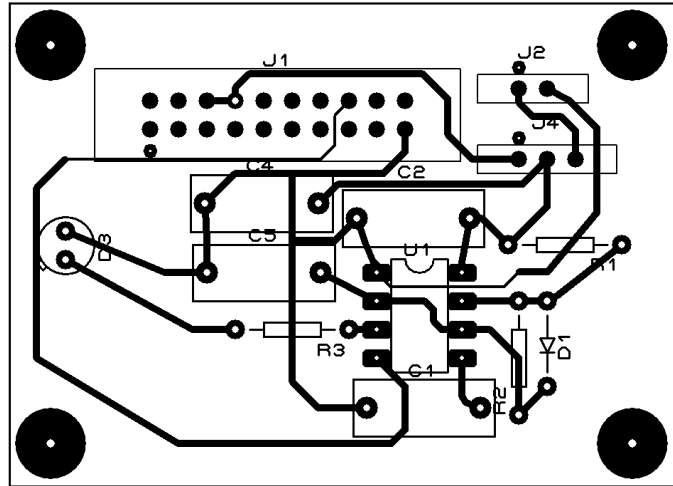


Fig. 3.8 Placa de l'emissor

S'observa com el diode emissor d'IR (D3) està ubicat en un dels extrems de la placa per tal que tingui visió directa amb el receptor per a establir la comunicació entre aquests.

En la següent figura es mostra la placa de l'emissor un cop fabricada.

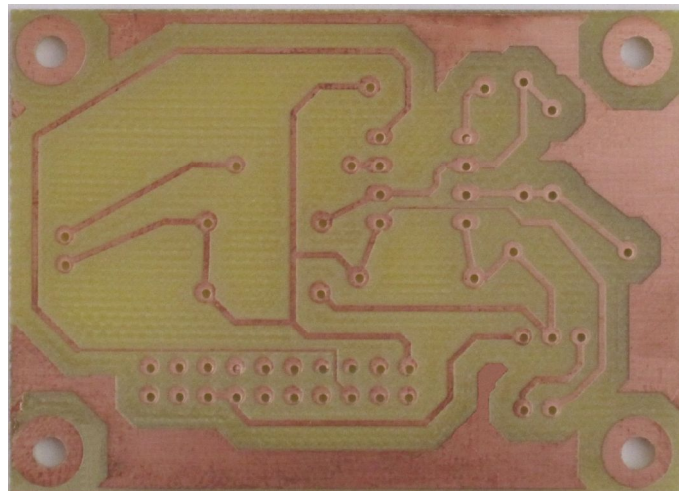


Fig. 3.9 Emissor

3.4. Receptor d'IR

L'esquema del receptor d'infrarojos és el següent:

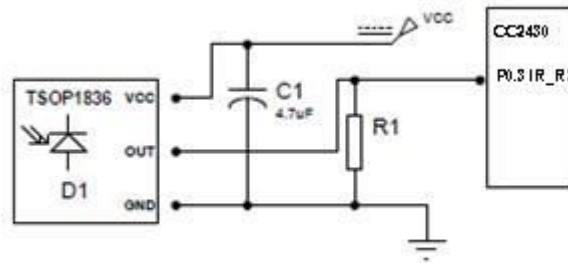


Fig. 3.10 Esquema del receptor

Inicialment, el xip receptor d'infraroig elegit va ser el TSOP 1836SB3V ja que com s'observa a les especificacions és aquest el model més adequat per la seva f_o . Aquest paràmetre fa referència a la freqüència de la portadora del senyal d'infraroig que es vol rebre.

Taula. 3.2 Tipus de TSOP

Type	f_o	Type	f_o
TSOP1830SB3V	30 kHz	TSOP1833SB3V	33 kHz
TSOP1836SB3V	36 kHz	TSOP1837SB3V	36.7 kHz
TSOP1838SB3V	38 kHz	TSOP1840SB3V	40 kHz
TSOP1856SB3V	56 kHz		

Per tal que el TSOP1836 tingui una bona recepció del senyal, al moment de dissenyar la placa, aquest ha d'estar en un dels seus extrems, orientat cap a l'exterior.

Com s'ha comentat anteriorment, el model escollit a l'inici del disseny, actualment ja no està a la venda, per tant s'ha hagut d'escollir un model nou, que complís les mateixes especificacions que aquest.

En aquest cas, s'ha optat pel TSOP34836 [7].

Aquest té unes característiques molt semblants al TSOP1836, però com que és una versió més nova, conté millores significatives.

- El seu consum és molt baix, consumeix 10 mW a una temperatura ambient inferior a 85°C.
- És compatible amb tots els formats comuns de dades de l'IR, en aplicacions de comandament a distància.
- L'interval de voltatges en el qual treballa és entre 2.5 i 5.5V.
- Major abast de transmissió, fins a 45 m.

- Dissenyats per a suprimir polsos de sortida falsos degut al soroll o alteracions de la senyal.

Els valors dels components que formen el disseny són:

$R1 = 5.6 \text{ k}\Omega$

$C1 = 4.7 \text{ nF}$

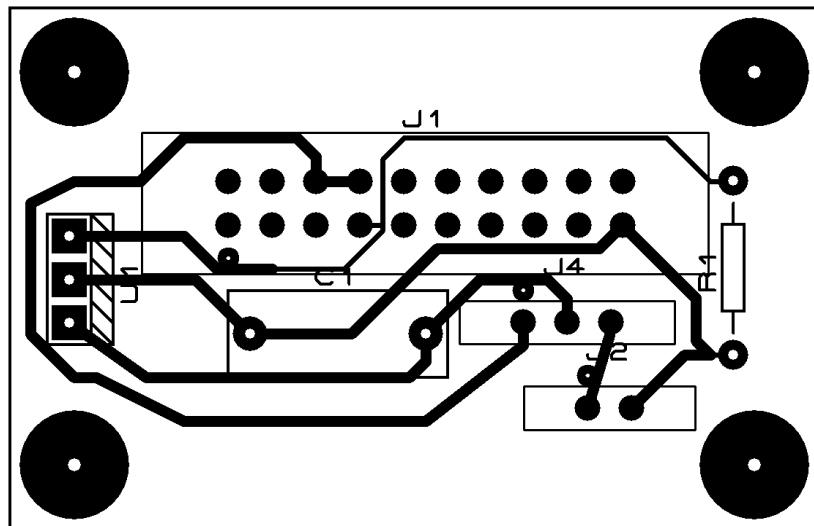


Fig. 3.11 Esquema del *layout* del receptor

En la següent figura es mostra la placa del receptor un cop fabricada.

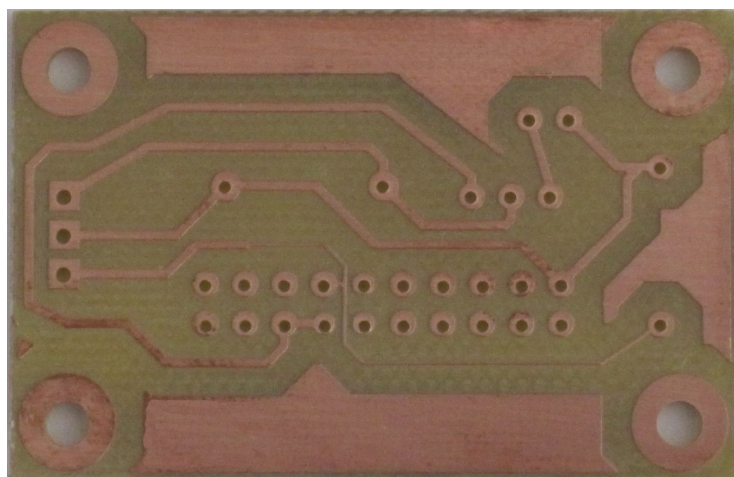


Fig. 3.12 Receptor

3.5. Placa de connexions

La finalitat de la placa de connexions és connectar totes les plaques del dispositiu, i també poder-les desconnectar quan no siguin necessàries.

Aquesta consta de 4 connectors 10x2 tipus PEC10DAAN (detallat anteriorment a l'apartat tipus de connectors)

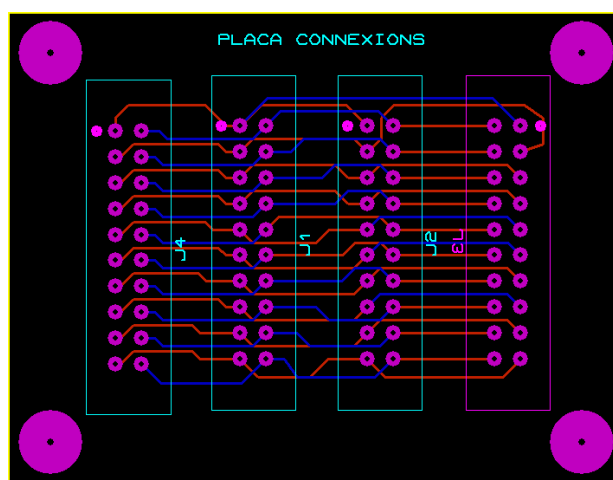


Fig. 3.13 *Layout* de la placa de connexions

A la figura anterior, es mostra el *layout* de la placa de connexions. Per realitzar aquest disseny, és necessari que la placa consti de 2 capes, ja que sinó seria impossible degut a que es creuarien les pistes.

Aquestes capes es poden veure clarament diferenciades amb els colors blau i vermell. Per a realitzar la figura, s'han tret els plans de massa per tal que sigui més fàcil de poder distingir els components i les pistes.

El connector de la dreta (J3) s'ha col·locat a una capa diferent dels altres 3, i al revés. Aquest connector està per una banda de la placa, i els altres 3 estan per l'altra. La raó és que el mòdul del xip CC2430 estarà connectat a la placa de connexions per una banda, i així quedaran, a l'altre costat, les connexions més accessibles pels perifèrics.

Les dimensions de la placa són: 46.23 mm (alt) x 59.69 (ample)

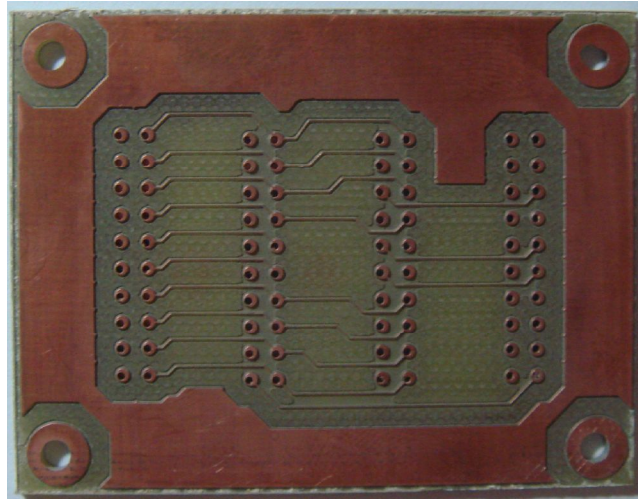


Fig. 3.14 Placa de connexions

3.6. Disseny del teclat

Al moment de dissenyar el teclat s'ha considerat que com a mínim són necessàries les tecles numèriques (0 a 9) i 4 botons addicionals corresponents a les funcions mínimes establertes: modificar, col·locar i eliminar ID i modificar objecte (per exemple, encendre o apagar una bombeta).

Degut a que en un teclat numèric, per defecte, les seves sortides són una matriu, implica que en el millor dels casos seria necessari un bus de 8 línies per tal de poder connectar el teclat al sensor.

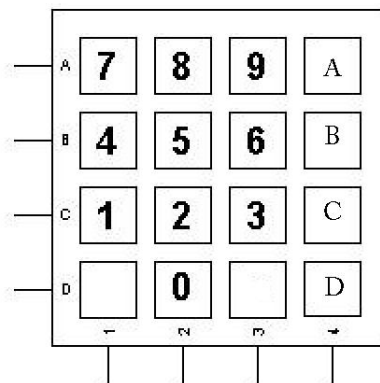


Fig. 3.15 Dibuix del teclat

Com que aquest nombre és molt elevat tenint en compte les entrades disponibles al node sensor, s'ha d'utilitzar algun mètode per tal d'ocupar el menor nombre de línies possibles.

Una possible solució contemplada és utilitzar dues funcions per tecla. S'implementaria de manera equivalent a la tecla majúscules (*shift*) del teclat de

l'ordinador. Cada tecla tindria 2 funcions, i seleccionariem una o l'altra depenent si abans s'ha premut la tecla *shift* o no. Aquesta solució no seria molt vàlida, ja que com a mínim necessitem les 10 tecles numèriques, cosa que aquest sistema com a molt aconseguiria reduir una o dues línies.

Una altra solució seria assignar, igual que al cas anterior, 2 o 3 funcions per cada tecla, com utilitzen els telèfons mòbils per escriure caràcters. El mètode consistiria en pitjar la tecla 1, 2 o 3 vegades seguides, seleccionariem una opció o una altra. Amb aquest mètode es podria aconseguir una reducció elevada de tecles, amb el corresponent estalvi de línies de bus.

Aquests 2 mètodes, sobretot el segon, poden reduir bastant l'ocupació de busos, però no són pràctics d'utilitzar en l'entorn que estan dissenyats, ja que es tracta d'un comandament, d'ús senzill, i sense cap pantalla LCD, per tan utilitzar aquests mètodes seria molt poc pràctic.

Degut a que els mètodes per reduir tecles amb la finalitat de reduir el número de línies no corresponen amb els requisits pel qual es vol dissenyat el teclat, la solució vàlida que es pot implementar és utilitzar el teclat sencer com es mostra a la figura anterior, sense cap mètode per simplificar tecles, i afegir algun element entremig que multiplexi, o codifiqui les tecles de manera que es redueixi de manera significativa el nombre de línies necessàries per interconnectar la de la matriu del teclat al sistema.

Per això s'han contemplat tres possibilitats:

1) Codificador de teclats matricials de 4x4 (EDE1144) [8]

Aquest és un circuit integrat de 18 pins que està dissenyat per a codificar les entrades d'una matriu de teclat de 4x4 per a un microcontrolador o microprocessador.

Entre les seves característiques més importants cal destacar que disposa de 4 sortides en paral·lel i una en sèrie per a l'obtenció del codi del botó pitjat, per tal d'escollir l'opció que més convingui en cada cas.

També conté d'una sortida d'un to de 1KHz que si alimenta un brunzidor, pot fer la funció de feedback, és a dir, pot donar la confirmació que s'ha premut una tecla.

Un altre avantatge d'aquest IC és que només escaneja les files i columnes per a determinar quina tecla s'ha pitjat un cop s'ha pitjat, és a dir, que no està contínuament actiu.

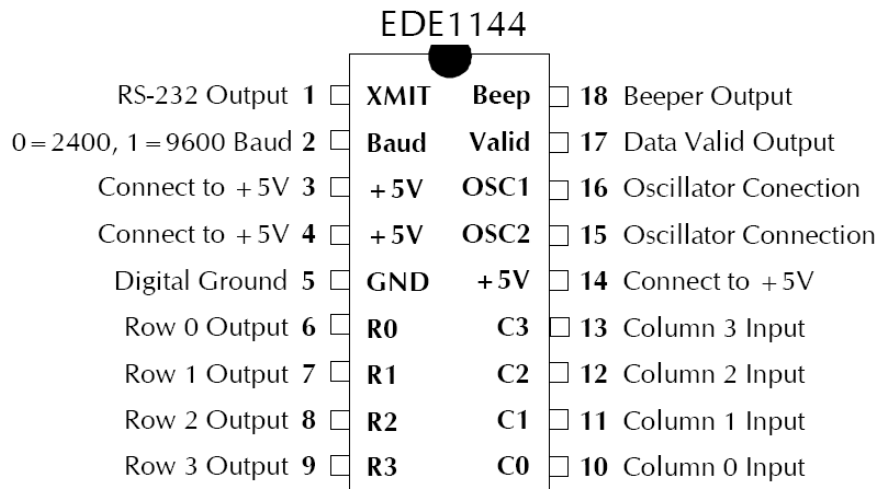


Fig. 3.16 Esquema dels ports del EDE1144

El cicle que segueix és el següent:

- 1- Detecta que s'ha premut una tecla.
- 2- Inicia un cicle per escanejar les tecles.
- 3- Les sortides en paral·lel reben el valor que correspon a la tecla.
- 4- S'activa el senyal de Data Valid que indica que hi ha un valor vàlid en les sortides en paral·lel.
- 5- S'activa la sortida del to d'1KHz.
- 6- Transmissió de les dades per la sortida en sèrie.

Aquest cicle es repeteix cada 50 ms. Si la tecla es manté premuda durant un període superior al retard entre cicle i cicle (660ms), aquest neteja les sortides i repeteix el cicle 5 vegades per segon fins que es deixa de prémer la tecla.

Inconvenient: Necessita una alimentació mínima de 4.5V.

2) Codificador de 16 tecles (MM74C922) [9]

Aquest és un circuit integrat CMOS que codifica 16 línies d'un teclat matricial en 4 bits.

Quan no hi ha cap tecla premuda, les entrades de les files activen el flag i les columnes de sortida treuen un zero lògic. En canvi, quan es prem una tecla la fila i la columna on està situada la tecla dins de la matriu 4x4, atura el comptador, es desactiva el flag i es bloquegen les altres entrades per tal que no hi hagi una confusió. El codi resultant és la combinació del valor del comptador i el valor descodificat de les entrades de les columnes.

Quan aquest procés ha acabat, s'activa el flag de Data Available que significa que s'ha rebut una nova dada. Es pot controlar la velocitat d'escaneig de les tecles i el període d'obtenció de les tecles modificant el condensador d'oscil·lació.

No s'escaneja novament el teclat per veure si s'ha premut una altra tecla fins que s'ha transmès la tecla anterior i s'han esborrat els comptadors. Aquest IC disposa de 4 sortides en paral·lel per a l'obtenció de la tecla premuda.

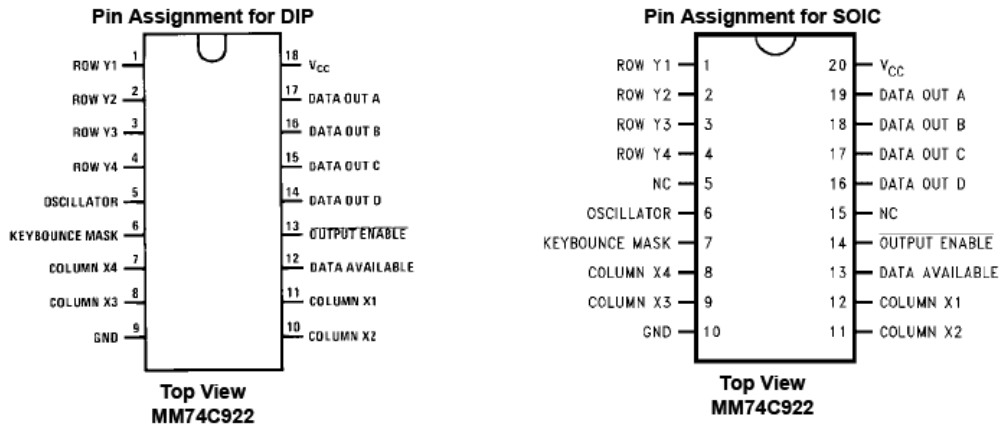


Fig. 3.17 Esquema dels ports del MM74C922

Inconvenient: Per a obtenir la sortida en un port sèrie, es necessita un convertidor de paral·lel a sèrie.

Al tractar-se d'un convertidor de 4 ports en paral·lel a un de sèrie s'ha trobat que tots els models estan obsolets i els que existeixen són de com a mínim, 8 entrades en paral·lel.

3) Programació d'un PIC16F628 [10]

És un microcontrolador de 18 pins d'arquitectura Harvard, que significa que les dades i la memòria circulen per busos diferents.

Contenen memòries internes, Flash, RAM i EEPROM. Tots els registres especials de funcions són emmagatzemats en aquestes memòries, això fa que es puguin programar amb les funcions que es desitgin.

Mitjançant aquestes funcions es pot aconseguir que tingui sortides en paral·lel i també en sèrie, i també es pot fer que hi hagi una sortida d'un to per tal d'alimentar un brunzidor, que anunciarà cada vegada que s'ha pitjat una tecla.

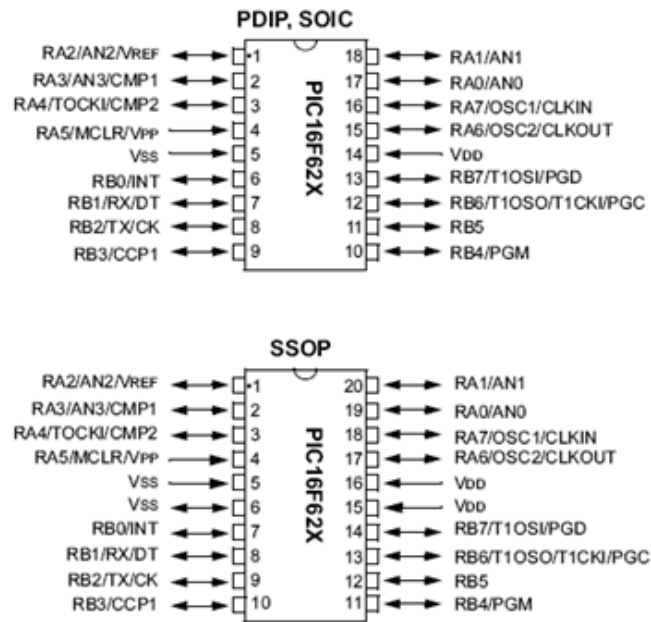


Fig. 3.18 Esquema dels ports del PIC16F628

Inconvenient: S'han de programar les funcions per tal que actuï com a codificador i doni les sortides en el format desitjat, la qual cosa sempre pot comportar més problemes que no pas escollir un codificador ja programat.

Després de sospesar les diferents alternatives s'ha escollit l'EDE1144 ja que disposa d'una sortida en sèrie que evita haver d'utilitzar un convertidor paral·lel/sèrie, i també s'ha descartat la utilització del PIC16F628 ja que la programació de qualsevol xip és una font més de possibles errors en el sistema.

Amb la utilització del EDE1144 el disseny amb Proteus de la placa és la següent:

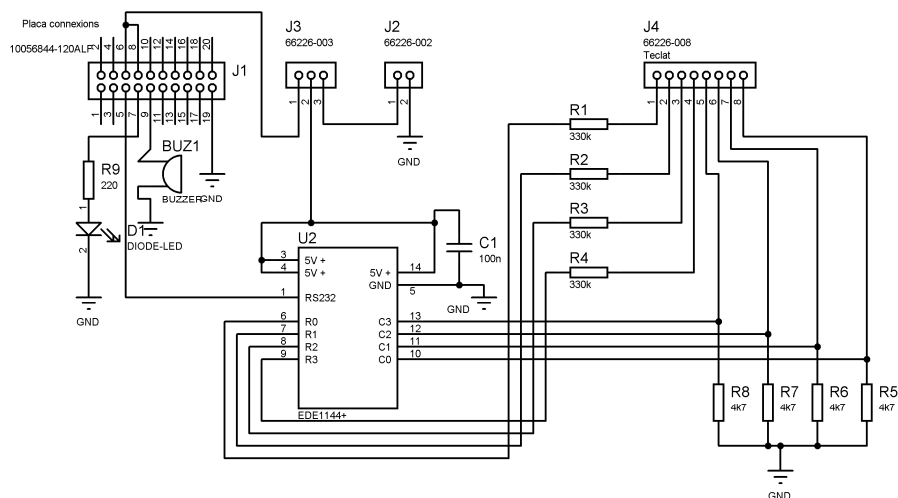


Fig. 3.19 Esquema del teclat

L'esquema de la configuració ha estat extret del *datasheet* del codificador EDE1144[8], que és la figura U2 de la imatge anterior.

L'alimentació com en la resta de plaques li ve donada a través del connector 3x1 (J3) on es pot escollir si s'agafa del sensor o bé d'una pila externa.

Al connector J4 hi ha els 8 busos de sortida de dades del teclat, que aquestes van connectades al codificador de la següent manera:

A les entrades de dades R0 a R3, es connecten els 4 primers pins del connector J4, que corresponen a les files del teclat, a través de resistències de 330 Ω . La raó de l'existència d'aquestes és per a evitar un possible curtcircuit durant el cicle d'escaneig de les files i columnes en cas que hi hagi més d'una tecla premuda al mateix moment.

Les dades de les columnes del teclat entraran al codificador per les entrades C0 a C3. Aquestes provenen dels últims 4 pins del connector J4. Entre aquestes i massa també hi ha connectades unes resistències de 4.7 K Ω (R5 a R8), que serveixen com a fites per tal que el valor de les entrades no oscil·li.

La sortida de dades en sèrie es fa pel port RS232 del EDE1144 (U2) que va a parar al pin 5 del connector que correspon a la RX UART.

En aquesta configuració també s'hi ha inclòs un condensador de desacoblament (C1).

Per tal de tenir la certesa que una ordre s'ha emès amb èxit, s'ha inclòs al disseny un brunzidor i un led per tal que donin una senyal sonora i lluminosa.

Aquests estan connectats al node sensor a través de les entrades UART.

En el disseny de la placa s'han tingut en compte les mides del teclat per tal que quedi inclòs en la mateixa.

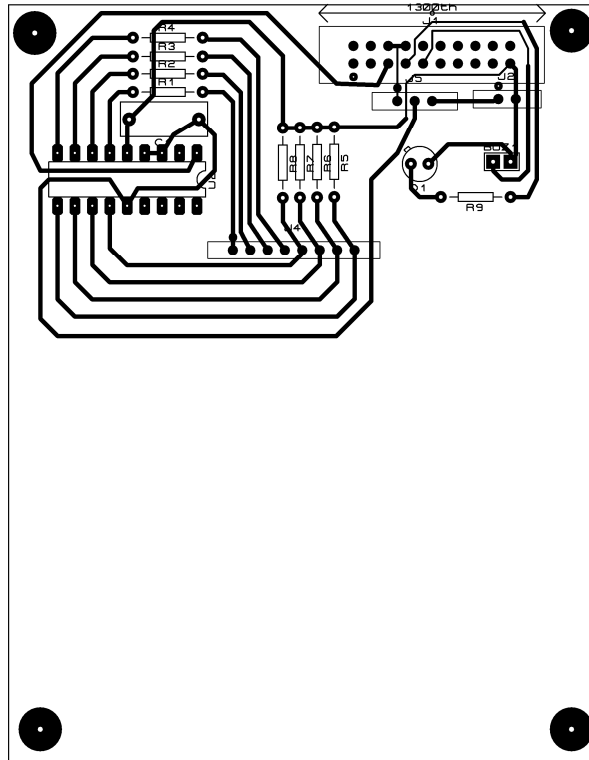


Fig. 3.20 *Layout del teclat*

En la següent figura es mostra la placa un cop fabricada.

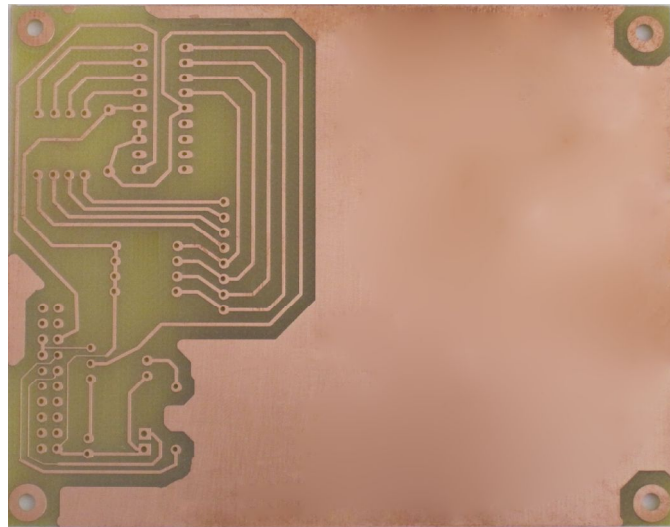


Fig. 3.21 *Teclat*

En un inici estava planejat també el disseny d'un sensor de corrent aprofitant un disseny ja existent, però degut a problemes amb l'obtenció de la informació necessària, no ha sigut possible la realització d'aquest mòdul

CAPÍTOL 4. OBTENCIÓ I REPRESENTACIÓ DE LA TEMPERATURA

Una de les característiques del CC2430 que es vol aprofitar en aquest projecte (veure capítol 2) és que conté un sensor de temperatura integrat que es pot utilitzar per a obtenir-ne dades.

En aquest cas les dades obtingudes són mostrades a través d'una interfície web confeccionant un gràfic lineal on es mostra de forma visual l'evolució de la temperatura al llarg del temps mesurada en un o més sensors.

4.1. Esquema

En l'esquema que hi ha a la figura següent es mostren els elements que intervenen en el procés d'obtenció de la temperatura i de visualització dels valors en una gràfica.

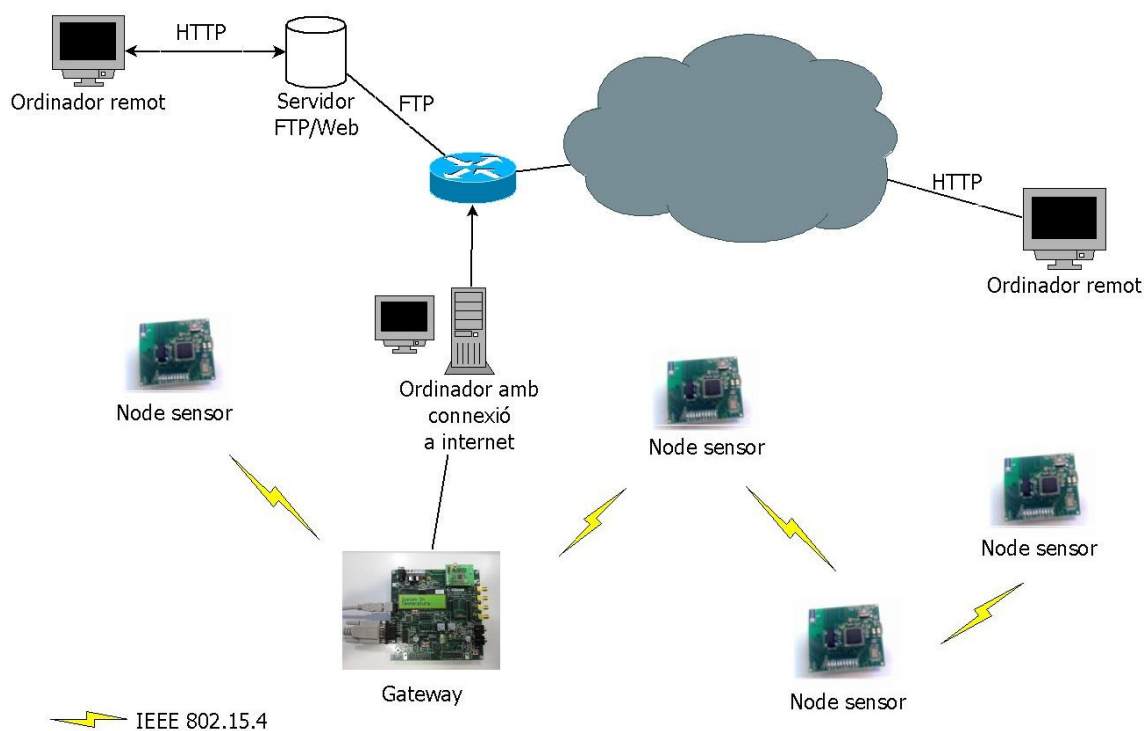


Fig. 4.1 Escenari de la xarxa de sensors

El sistema consta d'un node *gateway* format per un PC connectat a un node basat en el mòdul de Texas Instruments, que s'encarrega de recollir les dades

dels sensors. La transmissió de dades entre aquests dos es realitza mitjançant el port sèrie de què disposa la placa de desenvolupament SmartRF04EB a la que està acoblat al mòdul de Texas Instruments.

El node *gateway* processa i envia les dades per FTP a un servidor Web, per tal que puguin ser vistes mitjançant la connexió a un navegador.

4.2. Obtenció de la temperatura

Per a l'obtenció de la temperatura mitjançant el CC2430, primer és necessari configurar el ADC i fer una sèrie de càlculs per tal que el valor obtingut sigui el més fidel possible a la realitat.

4.2.1. ADC

La sortida del sensor de temperatura del xip es pot seleccionar com una entrada de l'ADC (convertor analògic-digital) per a realitzar les mesures de la temperatura.

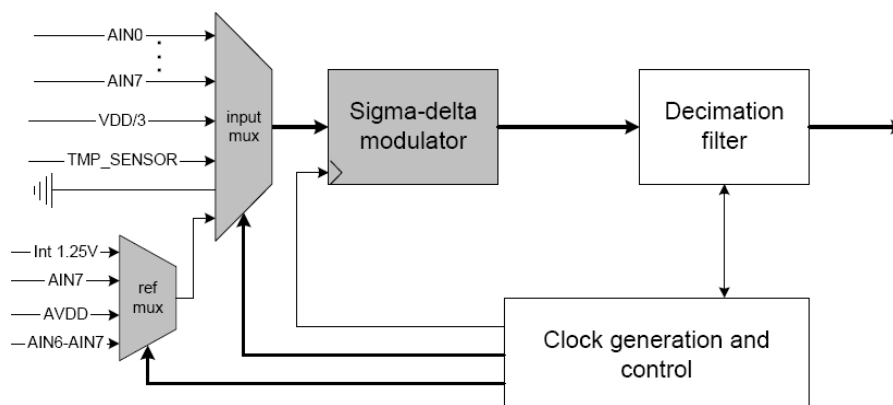


Fig. 4.2 Esquema de l'ADC

Per tal que l'ADC funcioni com a tal, s'han de configurar tres registres ADCCON1, ADCCON2 i ADCCON3 que a la vegada en reportaran també l'estat de la conversió.

En l'ADCCON1 (ADC control 1) els bits més importants a controlar són: EOC i ST.

El ADCCON1.EOC indica el final de la conversió posant-se a 1. Quan el valor s'ha llegit, es reseteja aquest valor i el ADCCON1.ST indica l'inici de la conversió.

Per tal de configurar l'entrada del sensor de temperatura s'ha de configurar el registre ADCCON2 (ADC control 2) que controla com es realitza la seqüència de conversions.

Així tenim que ADCCON2.SREF s'utilitza per a seleccionar el voltatge de referència. Aquest només es pot canviar mentre no hi ha cap conversió.

El bit de ADCCON2.SDIV selecciona la resolució i la freqüència de mostreig, que és el temps necessari per a completar una conversió. Aquest bit tampoc es pot canviar mentre una conversió s'està duent a terme.

Finalment tenim que amb el bits de ADCCON2.SCH es selecciona la seqüència de canal.

El registre ADCCON3 controla el número del canal, el voltatge de referència i també la taxa de conversió per una conversió extra.

El resultat de la conversió es mostra en complement a dos. Així es tindrà que per una entrada analògica igual a VREF (voltatge de referència), el valor de la conversió digital tindrà el mateix signe que aquest.

Aquest valor s'obté quan el bit ADCCON1.EOC és 1, i el resultat es col·loca tant en ADCH com ADCL. Serà quan es llegeixin els bits de ADCCON2.SCH quan es determini en quin dels dos ha d'anar.

4.2.2. Càlculs previs

Per tal que les dades obtingudes siguin reals, primer cal configurar el sensor amb els valors adequats.

Taula 4.1. Paràmetres dels sensors 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]

Aquesta taula mostra el voltatge (en mV) de sortida del sensor quan aquest estigui a 0°C. També s'hi pot trobar l'increment (en mV) per cada grau Celsius. Per tant, el voltatge de sortida serà:

$$V_{\text{sortida}} [\text{mV}] = V_{0^{\circ}\text{C}} [\text{mV}] + \text{Coef. de temp.} [\text{mV}/^{\circ}\text{C}] \cdot \text{Temp} [^{\circ}\text{C}] \quad (4.1)$$

I la temperatura, substituint els valors de la taula anterior, serà:

$$Temp [^{\circ}C] = \frac{V_{sortida} [mV] - 743 mV}{2.45 [mV/^{\circ}C]} \quad (4.2)$$

Tal com s'explica anteriorment, per a mesurar el voltatge de sortida del sensor de temperatura, aquest s'ha de seleccionar com a entrada de l'ADC.

Es fa ajustant els següents bits del ADC control 2:

`ADCCON2.SCH[3:0] = 1110`

Mitjançant aquests bits s'indica que s'ha seleccionat l'entrada del sensor de temperatura.

Es suposa que:

- S'utilitza el voltatge de 1.25V interns de referència:
`ADCCON2.SREF[1:0] = 00`
- L'ADC és configurat per una resolució de 12 bits.
`ADCCON2.SDIV[1:0] = 11`
 - Implica que el valor màxim serà de 2047 (2^{11}) ja que el valor es dóna en forma de complement a 2

la fórmula serà:

$$V_{sortida} [mV] = ValorADC \cdot \frac{1250 [mV]}{2047} \quad (4.3)$$

El *valor ADC* s'obtindrà a la secció MSB de `ADCH:ADCL`

4.2.3. Calibració

Segons les especificacions, l'error màxim de temperatura ha de ser de $\pm 2^{\circ}C$. 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 4.1.

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.

Com més mesures es realitzin, a la mateixa o a diferents temperatures, menys error es cometrà a l'estimar el valor d'*offset*.

Si es realitzen diferents mesures de temperatura per trobar diferents valors d'*offset* es farà la mitjana de tots per tal d'aplicar-ho a la fórmula de l'equació 4.4.

$$V_{offset} = V_{25^{\circ}\text{C}}[\text{mV}] - (V_{0^{\circ}\text{C}}[\text{mV}] + \text{Coef.de temp.}[\text{mV}/^{\circ}\text{C}] \cdot \text{Temp}[^{\circ}\text{C}]) \quad (4.4)$$

Per tal de calcular pràcticament el valor de l'offset es mesura la temperatura ambient de la sala mitjançant un termòmetre: es mesuren 22.6°C

$$V_{sortida}[\text{mV}] = \text{ValorADC} \cdot \frac{1250[\text{mV}]}{2047} = 21320 \cdot \frac{1250[\text{mV}]}{2047} = 813.18 \text{ mV} \quad (4.5)$$

$$\begin{aligned} V_{offset} &= V_{25^{\circ}\text{C}}[\text{mV}] - (V_{0^{\circ}\text{C}}[\text{mV}] + \text{Coef.de temp.}[\text{mV}/^{\circ}\text{C}] \cdot \text{Temp}[^{\circ}\text{C}]) \\ &= 813.18 - (743 + 2.45 \cdot 22.6) = 15.01 \text{ mV} \end{aligned} \quad (4.6)$$

S'aplica la fórmula de l'equació (4.4) i s'obté: $V_{offset} = 15.01 \text{ mV}$ Això implica que per obtenir una mesura més exacta de la temperatura, s'ha de sumar aquest valor al voltatge de sortida, de manera que la temperatura s'obindrà de la següent manera:

$$\text{Temp}_{\text{Real}}[^{\circ}\text{C}] = \frac{V_{sortida} - 743 \text{ mV} + V_{offset}}{2.45} = \frac{V_{sortida} - 768.75}{2.45} \quad (4.7)$$

Al següent gràfic es pot veure el voltatge de sortida tenint en comte la correcció d'offset (línia vermella) i sense tenir-la en comte (línia blava).

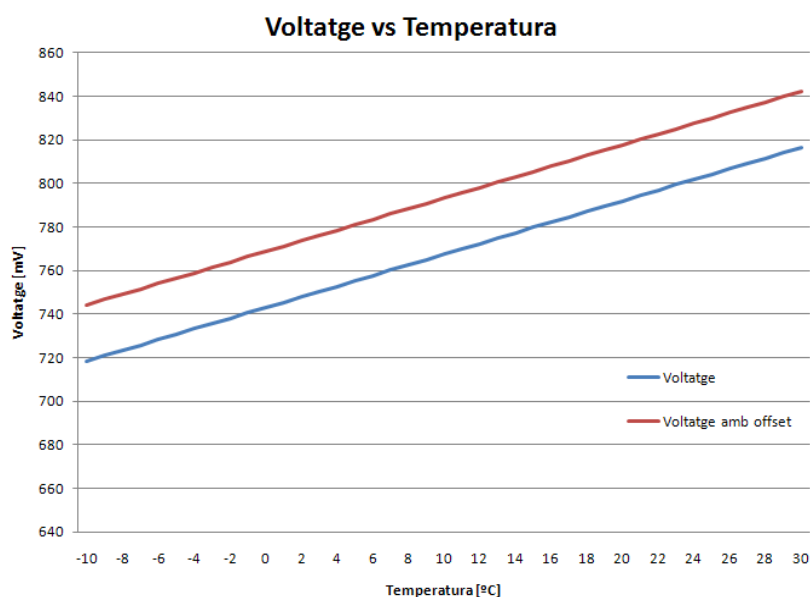


Fig. 4.3 Gràfic de Voltatge vs temperatura

A la següent taula es poden veure els paràmetres del sensor de temperatura integrat al CC2430.

Taula. 4.2 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 $+40^{\circ}\text{C}$		0.840		V	Value is estimated
Output voltage at $+80^{\circ}\text{C}$		0.939		V	Value is estimated
Temperature coefficient		2.45		$\text{mV}/^{\circ}\text{C}$	Fitted from -20°C to $+80^{\circ}\text{C}$ on estimated values.
Absolute error in calculated temperature		-8		$^{\circ}\text{C}$	From -20°C to $+80^{\circ}\text{C}$ when assuming best fit for absolute accuracy on estimated values: 0.743V at 0°C and $2.45\text{mV}/^{\circ}\text{C}$.
Error in calculated temperature, calibrated	-2	0	2	$^{\circ}\text{C}$	From -20°C to $+80^{\circ}\text{C}$ when using $2.45\text{mV}/^{\circ}\text{C}$, 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	

Una altra característica que es pot apreciar en aquesta taula és el consum. Quan està habilitat el sensor de temperatura, el CC2430 consumeix $280\ \mu\text{A}$ més que en condicions normals.

4.2.4. Codi utilitzat

En aquest apartat s'explica el codi de programa utilitzat per tal de recollir les dades que pren el sensor de temperatura del CC2430. Aquest programa s'ha desenvolupat mitjançant l'aplicació IAR Embedded Workbench, que és un entorn de programació, depuració i eina per carregar el codi màquina, que suporta diversos xip microcontroladors/processadors, entre els quals hi ha el xip CC2430.

Primer que tot i un cop realitzats els càlculs de calibració esmentats anteriorment es defineixen les variables següents:

La constant que correspon a la divisió (1250/2047), l'offset teòric que esmenta en la taula del *datasheet*, l'offset que s'ha calculat amb aquest sensor i la suma d'aquests. Finalment també es defineix el coeficient de temperatura.

```
#define CONSTA 0.61065
#define OFFSET_DATASHEET 743
#define OFFSET_MEASURED_AT_23_DEGREES_CELCIUS 15.03
#define OFFSET (OFFSET_DATASHEET +
OFFSET_MEASURED_AT_23_DEGREES_CELCIUS)
#define TEMP_COEFF 2.45
```

En l'arxiu *hal_adc.h* i *hal_adc.c* s'ubica la funció que es realitza: *HalAdcCheckVdd*. Aquesta funció el que fa és configurar el ADC control 3 per tal que tingui els paràmetres anteriorment esmentats, i agafa el valor en mV de l'ADC.

Per tal de passar aquest valor a graus Cèlsius un cop ha obtingut el valor de l'ADC, el desplaça 4 bits, per tal que la mostra contingui els 12 bits més significatius. En les següents operacions s'ocupa de multiplicar el valor per la constant i restar-li l'offset que s'introdueix, tal com mostra (4.7).

```
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);
}
```

4.3. Node *gateway*: comunicació port sèrie - PC

Tal i com es mostra en l'esquema de la figura 4.1, el node *gateway* està format per un PC connectat a un node sensor amb el mòdul de Texas Instruments. La comunicació entre aquest node i el PC es realitza mitjançant una comunicació sèrie RS232, mentre que la comunicació d'aquest node amb la resta de sensors que formen la xarxa es fa via ràdio.

En els apartats que venen a continuació s'expliquen els detalls per a la seva realització.



Fig. 4.4 Escenari de la comunicació *gateway*-pc

4.3.1. Enviament de dades

Un cop s'ha obtingut el valor de la temperatura, cal transferir aquesta dada a un PC extern per tal de poder-la tractar.

Per això s'aprofita el port sèrie que conté la placa SmartRF04EB, en la que es basa el *gateway*.

El port del CC2430 pel qual es vol enviar la informació s'ha de configurar com UART. Consultant la taula que conté la relació entre els pins i cada port, es pot veure com per a utilitzar el port 0, els pins pels quals es pot rebre i enviar són el 0.2 i el 0.3 respectivament.

Taula. 4.3 Relació entre pins i ports del CC2430

Periphery / Function	P0								P1								P2				
	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	4	3	2	1	0
ADC	A7	A6	A5	A4	A3	A2	A1	A0													
USART0 SPI			C	SS	M0	MI															
Alt. 2											M	MI	C	SS							
USART0 UART			RT	CT	TX	RX															
Alt. 2											TX	RX	RT	CT							
USART1 SPI			MI	M0	C	SS															
Alt. 2									MI	M0	C	SS									
USART1 UART			RX	TX	RT	CT															
Alt. 2									RX	TX	RT	CT									

Per tant, al moment de configurar el port, s'ha d'especificar aquest condició. Això es realitza a l'arxiu `hal_uart.c`.

Dins d'aquest hi ha totes les funcions relacionades amb el port sèrie, encara que només se n'utilitzen dues per tal d'enviar les dades: *HalUARTOpen* i *HalUARTWrite*.

La primera s'utilitza per a obrir el port, i és aquí on s'ha d'especificar quins pins són els que s'utilitzen per enviar i rebre. A continuació es mostra l'encapçalament de la funció i com es seleccionen els pins.

P0SEL és el registre de la funció de selecció del port 0.

P0DIR és el registre de direcció del port 0.

```
uint8 HalUARTOpen( uint8 port, halUARTCfg_t *config )
{
[...]
```

```
P0SEL |= (1 << 3) | (1 << 2); // Selecciona P0.2 i P0.3
com Tx i Rx UART
P0DIR = (P0DIR & 0x06) | 0x04; // Selecciona el P0.2 com
entrada i el P0.3 com a sortida

[...]
```

La següent funció s'utilitza per a enviar les dades. Se li passen com a paràmetres el port pel qual es vol enviar, un vector que conté els bytes de dades i la longitud d'aquest vector.

```
uint16 HalUARTWrite( uint8 port, uint8 *buf, uint16 len )
{
[...]
```

En el `msa_main.c` es configura el port 0 amb els paràmetres desitjats i s'obra aquest port amb la configuració que s'ha elegit.

En aquest cas, la configuració que s'ha escollit és la següent:

```
halUARTCfg_t uartConfig;
    uartConfig.configured           = TRUE;
    uartConfig.baudRate             = HAL_UART_BR_38400;
    uartConfig.flowControl          = TRUE;
    uartConfig.flowControlThreshold = 48;
    uartConfig.rx.maxBufSize        = 64;
    uartConfig.tx.maxBufSize        = 64;
    uartConfig.idleTimeout          = 6;
    uartConfig.intEnable            = TRUE;
    uartConfig.callBackFunc         = MT_UartProcessRxData;

HalUARTOpen(0, &uartConfig);
```

Per tal que les dades es prenguin cada un interval determinat, es configura un temporitzador a l'arxiu `msa.c`, que genera un event un cop acabat aquest temps.

```
osal_start_timerEx(MSA_TaskId, MSA_PROVATEMP, 60e3);
```

L'event següent mesura la temperatura i la desa a la variable `valor_temp`. Un cop es té aquest valor, es multiplica per 100 ja que el valor que obté la variable `valor_temp` conté decimals, i per a l'enviament d'aquestes dades és millor que sigui una variable de tipus enter.

El valor de la multiplicació es guarda a la variable `provaprova`. Amb aquesta operació s'obté un enter de quatre dígit.

Posteriorment es transforma aquesta variable `provaprova` en un vector del tipus `uint8`, és a dir, un enter de 8 bits. Cada `uint8` conté un byte.

Per tal que es pugui representar un valor de dos dígit i dos més darrere la coma, és necessari enviar dos bytes. Aquests es guarden al vector `provaprova2`.

Quan es tenen els valors d'aquests dos bytes emmagatzemats al vector `vectorprova`, s'envien mitjançant la funció `HalUARTWrite`.

En la funció següent es detallen les operacions esmentades anteriorment.

```
uint16 MSA_ProcessEvent(uint8 taskId, uint16 events)
{
    [...]
    if (events & MSA_PROVATEMP) {
```

```

    HalLcdWriteString("Temperatura",2);
    valor_temp=HalAdcCheckVdd();
    provaprova=(int)((valor_temp)*100);
    provaprova2=(uint8 *) &prova;
    vectorprova[0]=*prova;
    vectorprova[1]=*(prova+1);
    HalUARTWrite (0,vectorprova, 2);

    return events ^ MSA_PROVATEMP;
}
[...]
```

4.3.2. Recepció de dades

Per a la recepció de dades al PC, s'ha creat un programa amb C# que s'ocupa de configurar els ports d'aquest amb els mateixos paràmetres que s'han establert al moment d'enviar.

Aquest programa s'executa de manera permanent i resta a l'espera de rebre dades a través del port sèrie i un cop les ha rebut, les processa, i torna a romandre a l'espera.

S'ha configurat el port de la següent manera:

```

SerialPort sp = new SerialPort(); //S'inicialitza un port
sèrie //anomenat sp

sp.PortName = "COM1"; //Es configura que el port sèrie
sigui el COM1
sp.BaudRate = int.Parse("38400"); //Es posea el mateix
BaudRate que //l'utilitzat pel xip

sp.Parity = (Parity)Enum.Parse(typeof(Parity), "None");
//Paritat //desactivada

sp.StopBits = (StopBits)Enum.Parse(typeof(StopBits),"One");
// Un bit //de parada

sp.DataBits = int.Parse("8"); // bits de dades
sp.Handshake = (Handshake)Enum.Parse(typeof(Handshake),
"None"); //Handshake desactivat

//s'obre el port sèrie
sp.Open()
//es configura el port per recepció
sp.RtsEnable = true;
sp.DtrEnable = true;
```

El *timeout* s'ha desactivat, ja que ens interessa que el programa estigui esperant que li arribin les dades del port sèrie el temps que faci falta, d'aquesta manera és el sensor qui decideix la freqüència d'actualització de les dades al gràfic sense necessitat de modificar el codi de l'ordinador.

El programa, té un *buffer* de recepció de 4 bytes. D'aquests quatre, dos s'utilitzen per emmagatzemar el valor de la temperatura i dos més per a emmagatzemar l'adreça del sensor que li està enviant la dada. Quan el *buffer* s'omple, significa que ja li han arribat les dades necessàries i procedeix a tractar-les.

Aquest tractament de dades s'explica en l'apartat 4.4.1.

4.4. Casos particulars de funcionament

En aquest apartat es contemplen dos tipus de xarxa per a l'obtenció de la temperatura: una xarxa formada per un node i una xarxa formada per un conjunt de nodes dispersos.

4.4.1. Un node

La xarxa formada per un node consisteix en un mòdul sensor Texas Instruments acoblat a la placa de desenvolupament SmartRF04EB, la qual està connectada a través del port sèrie al PC.

La configuració de la xarxa es mostra en la figura següent:

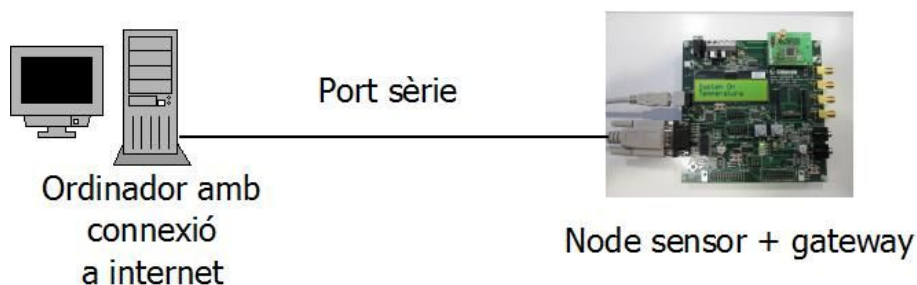


Fig. 4.5 Xarxa d'un sol node

En donar-li la ordre pertinent, aquest recull el valor de la temperatura i l'envia a través del port sèrie al PC per al seu posterior tractament.

Se li poden enviar dos tipus d'ordres, per tal que envii només una mostra de temperatura, o bé perquè envii mostres de forma continuada amb un interval de temps determinat.

4.4.2. Diversos nodes

La xarxa formada per diversos nodes consisteix en un node *gateway* connectat al PC per port sèrie i un número indeterminat de nodes repartits de manera que tenen comunicació entre ells i amb com a mínim un node que té comunicació amb el *gateway*.

Aquesta configuració es mostra en la següent imatge:

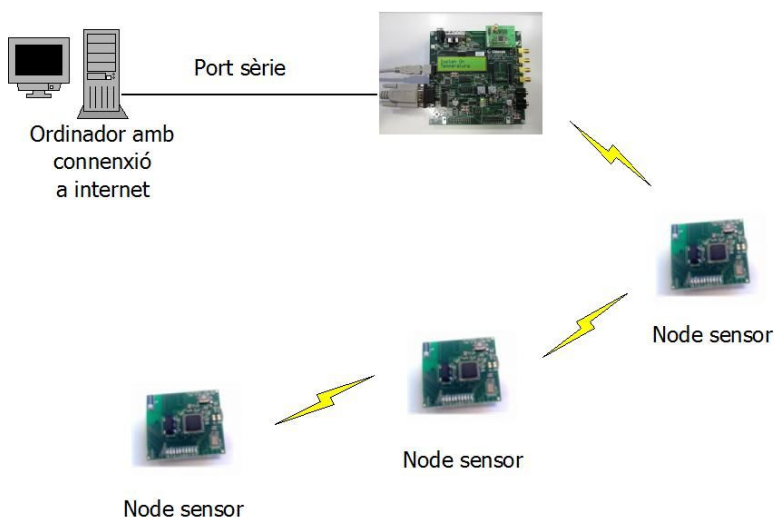


Fig. 4.6 Xarxa amb diversos nodes

Aquest escenari es correspondria, per exemple, amb un conjunt de nodes repartits entre les diferents estances d'una llar. Tots es poden comunicar amb tots, encara que no sigui directament, formant una xarxa de sensors multisalt. Per aconseguir això s'utilitza un protocol de reenviament de missatges entre nodes que s'encarrega de processar i si cal reenviar un missatge per a que aquest arribi al seu destí.

D'aquesta manera es pot fer que el *gateway* rebi els valors de les dades que desitgi, del node que desitgi, encara que no hi tingui enllaç directe amb ell.

4.4.2.1. Protocol de reenviament

El protocol de reenviament emprat en aquest TFC ha estat implementat per l'estudiant Eric Peris (veure Annex 1). És un protocol d'inundació. Cada node reenvia tots els missatges que rep per a facilitar que arribin el seu destí. El destí per la seva banda processa el missatge i pren les accions oportunes.

Per a fer una inundació el més eficaç possible s'han implementat una sèrie de mecanismes per a diferenciar els missatges entre ells, evitant reenviar missatges que ja han estat reenviats abans.

Aquest protocol s'ha utilitzat de base per a construir el mecanisme d'enviament de missatges entre diferents nodes sensors. S'han adaptat diverses funcions i s'han creat nous tipus de missatge per tal d'obtenir les dades desitjades de cadascun d'ells.

Per entendre com això s'aconsegueix, primer s'explicarà el format dels missatges i després el seu processat. Finalment es mostrarà la seva implementació en software.

4.4.2.2. *Format del missatge*

Per tal de complir els requeriments esmentats en el capítol 2, s'ha adaptat la capçalera del missatge dissenyat en el protocol de reenviament per tal que es puguin demanar diferents tipus de dades, i també s'ha ampliat la longitud de manera que hi hagi espai suficient per a aquestes.

S'hi han afegit dos bytes més als 9 que es contemplaven en el protocol, que formen una longitud total del missatge d'11 bytes.

El missatge creat té el següent format:

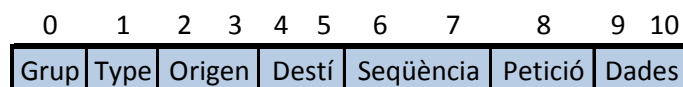


Fig. 4.7 Format dels missatges

Cada byte correspon a:

- **Grup:** És el PAN ID
- **Type:** Tipus de missatge.
- **Origen:** Adreça d'origen del missatge.
- **Destino:** Adreça destí del missatge.
- **Seqüència:** Número de seqüència.
- **Petició:** Tipus d'ordre i resposta.
- **Dades:** Dades a enviar.

El primer byte correspon al camp del grup. Com que es poden obtenir diferents dades dels sensors, així com també se'ls pot enviar diferents ordres a cadascun d'ells, és necessari que hi hagi un camp de grup, per tal que es puguin crear diferents xarxes dins de la xarxa de sensors.

El byte que correspon al tipus de missatge s'utilitza per diferenciar dos tipus que es poden rebre: tipus MSG o RESTART.

Aquest camp es defineix dins del protocol de reenviament de missatges.

Quan aquest byte és MSG, significa que és un missatge de dades. En canvi, quan aquest byte és RESTART, significa que el node s'ha reiniciat, i per tant, cal prendre una sèrie de mesures que s'explicaran a continuació.

Els camps d'origen i destí contenen les adreces del node origen i del destinatari del missatge. Cadascuna ocupa dos bytes.

Cada missatge pot anar adreçat a un node en concret o bé a tots els nodes a la vegada. Per aquest últim cas s'ha definit la següent adreça:

```
#define MSA_DEV_SHORT_ADDR_BROADCAST 0xFFFF
```

El número de seqüència és un valor que identifica cada missatge. S'utilitza per a comprovar que els missatges que rep un sensor son nous, no són repetits. D'aquesta manera s'evita una possible saturació de la xarxa.

El byte de petició és el que serveix per identificar el tipus de dades que se li estan demanant o bé que envia un sensor com a resposta.

Aquest és un dels nous camps que s'han inclòs en el format de missatge que contenia el protocol de reenviament.

En el cas que es vulgui obtenir la temperatura d'un sensor, el camp de petició pot ser de dos tipus:

```
#define TREQ 0x03
#define TRSPS 0x04
```

El primer s'utilitza quan es demana la temperatura a un sensor. Quan el destinatari del missatge identifiqui aquest byte, sabrà que se li està demanant el valor de la temperatura.

En el moment de contestar el missatge, en el byte de petició hi haurà un TRSPS. D'aquesta manera, quan el *gateway* rebí aquest missatge, sabrà que porta dades de temperatura enviades per un node sensor.

Aquesta diferenciació de tipus, permet que un mateix missatge es pugui enviar a més d'un node, sense enviar peticions individuals a cada node que pugui saturar el sistema.

Finalment, els dos últims bytes del missatge, que s'han inclòs en aquest format, s'utilitzen per a enviar les dades que se li han demanat al sensor.

4.4.2.3. *Processat dels missatges*

El processat és el conjunt d'accions que duu a terme un node quan rep un missatge.

Tant en el cas del *gateway* com en el de qualsevol altre sensor, el procés és el que s'il·lustra en la figura següent:

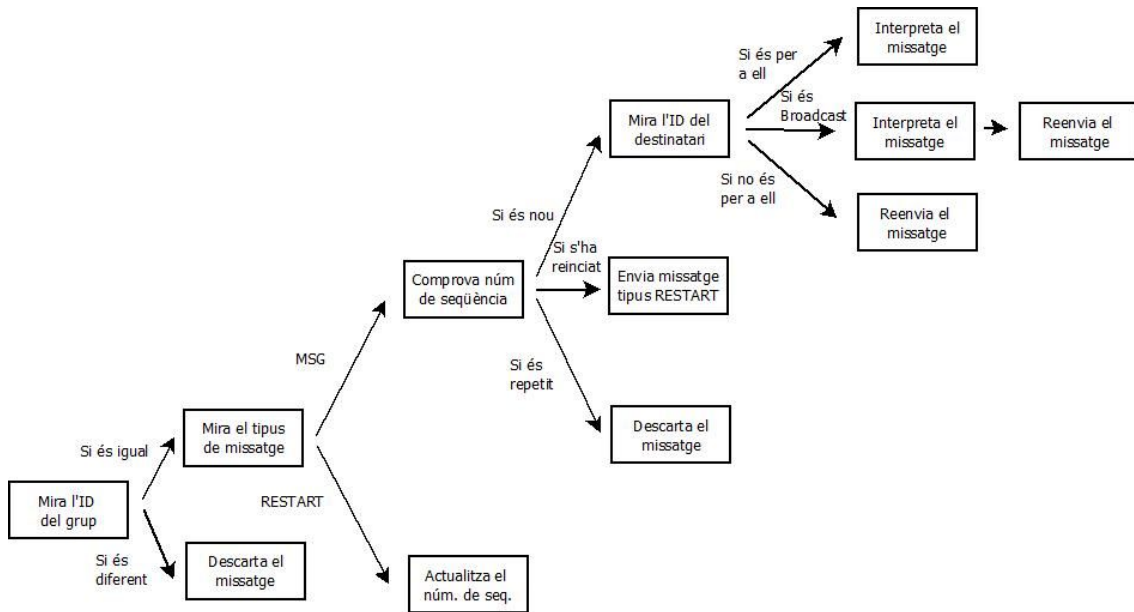


Fig. 4.8 Diagrama de processat dels missatges

Aquest protocol el primer que fa és verificar si el missatge rebut anava destinat al grup al qual pertany.

Si és així, comprova el tipus de missatge. Si és del tipus RESTART, desa el número de seqüència que li han enviat, i continua enviant missatges de tipus MSG amb el número de seqüència actualitzat.

En el cas que el missatge sigui de tipus MSG, comprova el número de seqüència. Això ho fa mitjançant els criteris descrits en el protocol de reenviament de missatges (Annex 1).

Si detecta que no és un missatge nou, pot ser degut a dos casos: que el node que envia el missatge s'hagi reiniciat o bé que rebi un missatge repetit.

En cas que el node s'hagi reiniciat, el node que ho ha detectat, li envia un missatge de tipus RESTART. Aquest missatge conté l'últim número de seqüència que li havia enviat a aquest. En rebre un missatge d'aquest tipus, cada sensor actualitza el número de seqüència per tal de continuar enviant missatges correctes.

Si el missatge és repetit, el node descarta el missatge per tal de no saturar la xarxa.

Si el número de seqüència rebut és correcte, passa a mirar-ne el destinatari. Si el missatge no va adreçat al propi node, reenvia el missatge rebut a la resta de nodes.

En el cas que el missatge contingui l'adreça de *Broadcast*, el sensor interpretarà el missatge i a més, el reenviarà a la resta de nodes per tal que es propagui per tota la xarxa.

Si el missatge va adreçat al propi node, aquest n'interpreta el contingut.

4.4.2.4. Implementació del software:

En aquest apartat, es mostrarà el codi implementat tant per al *gateway* com per a la resta de nodes que formen la xarxa.

Està basat en el protocol de reenviament de missatges, però amb l'adaptació de les diverses funcions als requeriments esmentats.

Per al node *gateway*, fragment de el codi a destacar és el següent:

Quan el gateway ha d'enviar una ordre als nodes, genera un esdeveniment (MSA_SEND_PRUEBA2 o MSA_SEND_PRUEBA3) que serà processat en la part de codi següent:

```
if ((events & MSA_SEND_PRUEBA2) || (events &
MSA_SEND_PRUEBA3))
{
    if(events & MSA_SEND_PRUEBA2)
        addr_destino=5;
    else
        addr_destino=MSA_DEV_SHORT_ADRR_BROADCAST;

        /* Group */
        msg_prueba[0] = GROUP;
        /* Type */
        msg_prueba[1] = MSG;
        /* Addr destino */
        msg_prueba[2] = LO_UINT16( addr_destino );
        msg_prueba[3] = HI_UINT16( addr_destino );
        /* Addr origen */
        msg_prueba[4] = LO_UINT16( msa_DevShortAddr );
        msg_prueba[5] = HI_UINT16( msa_DevShortAddr );
        /* Seqnum */
        msg_prueba[6] = LO_UINT16( sequence );
        msg_prueba[7] = HI_UINT16( sequence );
        /* Temperatura */
        msg_prueba[8] = TREQ;
        msg_prueba[9] = BYTE_EMPTY;
        msg_prueba[10] = BYTE_EMPTY;
```

```

sequence++;

if (sequence++ == 2*seq_umbral)
{
sequence = seq_umbral;
}

if(get_radio() == RX_OFF)
{
set_radio(RX_ON);
macRxOn();
}

ultimos_rx[msa_DevShortAddr-1].seqnum = sequence;
MSA_McpsDataReq((uint8*)msg_prueba,
                MSA_PACKET_LENGTH,
                TRUE,
                MAC_SHORT_ADDR_BROADCAST);

if (events & MSA_SEND_PRUEBA2)
return (events ^ MSA_SEND_PRUEBA2);
else
return (events ^ MSA_SEND_PRUEBA3);

```

En aquest codi, mira quin esdeveniment s'ha generat per tal de determinar l'adreça del destinatari, un sensor particular (per exemple, l'últim sensor), o bé a tots els sensors.

Posteriorment, es construeix el missatge a enviar amb les adreces corresponents i el byte de petició a TREQ. S'augmenta el número de seqüència de cara a futurs enviaments de missatges i finalment envia el missatge a tots els nodes.

Si el missatge és una resposta a una petició de temperatura, primer que tot comprova el número de seqüència amb els passos que s'esmenten en l'Annex 1, comprova que estigui adreçat a ell, i en cas afirmatiu, comprova que el byte de petició sigui TRSPS, i obté el valor de la temperatura així com el de l'adreça origen del sensor i envia aquests dos valors a través del port sèrie RS232 cap al PC.

```

if(sequence_ok)
{
ultimos_rx[addr_origen-1].seqnum = sequence_tmp;

if(addr_destino == msa_DevShortAddr)
{
tipus=pData->dataInd.msdu.p[8];

switch (tipus)
{
case TRSPS:

```

```

        vectorprova[0]=pData->dataInd.msdu.p[9];
        vectorprova[1]=pData->dataInd.msdu.p[10];
        vectorprova[2]=pData->dataInd.msdu.p[4];
        vectorprova[3]=pData->dataInd.msdu.p[5];
        HalUARTWrite (0,vectorprova, 4);

        break;
    }
}

```

Pel que fa als sensors, el codi utilitzat a l'hora de processar un missatge rebut és el següent:

```

if(sequence_ok)
{
    ultimos_rx[addr_origen-1].seqnum = sequence_tmp;

    if ((addr_destino ==
msa_DevShortAddr) || (addr_destino==MSA_DEV_SHORT_ADRR_BROADC
AST))
    {
        tipus=pData->dataInd.msdu.p[8];
        switch (tipus)
        {
            case TREQ:
                valor_temp=HalAdcCheckVdd();
                provaprova=(int)((valor_temp)*100);
                provaprova2=(uint8 *) &provaaprova;
                vectorprova[0]=*provaaprova2;
                vectorprova[1]=*(provaaprova2+1);
                prova_seq=sequence_tmp+1;

                msg_prueba[0] = pData->dataInd.msdu.p[0];
                msg_prueba[1] = pData->dataInd.msdu.p[1];
                msg_prueba[2] = pData->dataInd.msdu.p[4];
                msg_prueba[3] = pData->dataInd.msdu.p[5];
                msg_prueba[4] = LO_UINT16( msa_DevShortAddr );
                msg_prueba[5] = HI_UINT16( msa_DevShortAddr );
                msg_prueba[6] = LO_UINT16( prova_seq );
                msg_prueba[7] = HI_UINT16( prova_seq );
                msg_prueba[8] = TRSPS;
                msg_prueba[9] = vectorprova[0];
                msg_prueba[10] = vectorprova[1];

                MSA_McpsDataReq((uint8*)msg_prueba,
                                MSA_PACKET_LENGTH,
                                TRUE,
                                addr_origen);

            break;
        }
    }
}

```

En aquest cas, comprova si el missatge va adreçat a ell. Si és així, comprova si es tracta d'una petició de temperatura i obté el valor de la temperatura del sensor per retornar-la a l'adreça origen de la petició. En cas que aquest missatge no sigui per a ell, el reenvia per tal que pugui arribar al seu destí.

4.5. Tractament de les dades

Un cop s'han rebut les dades al PC, aquestes s'han de processar per tal d'obtenir els valors de la temperatura i també els números d'identificador del sensor que li està enviant les dades, per a posteriorment, confeccionar un gràfic que mostri de forma visual el progrés d'aquesta variable.

4.5.1. Diagrama

En el següent diagrama es mostra tot el procés que es segueix, des de l'adquisició de les dades fins al tractament final per tal de mostrar-les de forma visual en una gràfica.

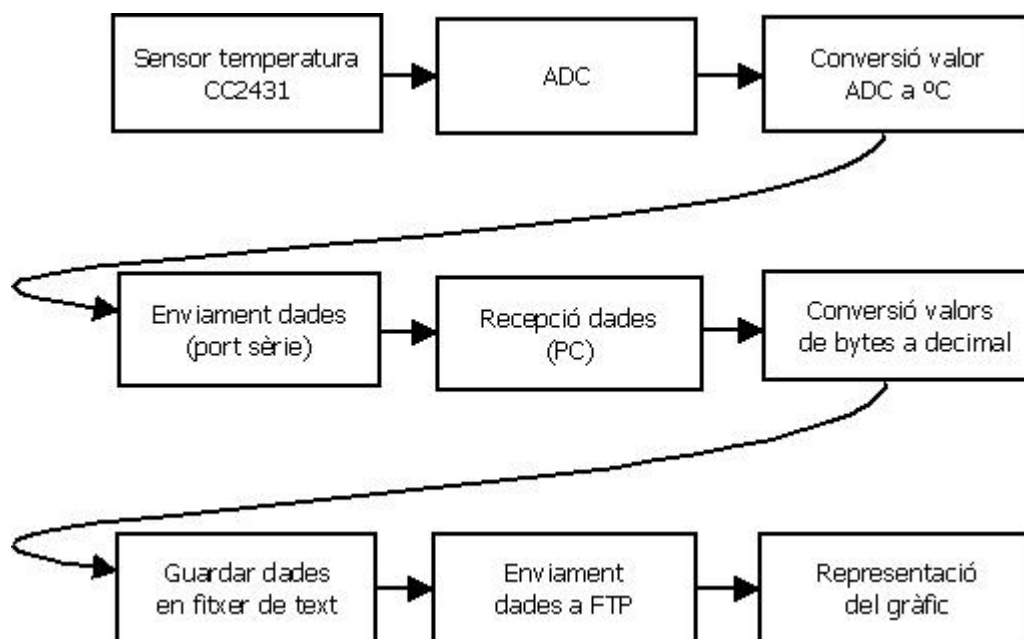


Fig. 4.9 Diagrama de flux del tractament de dades

4.5.2. Codi C#

El programa que s'ha fet amb C#, és el que s'executa a l'ordinador que s'encarrega de rebre les dades del sensor i actualitzar el gràfic de la pàgina web.

El programa rep pel port sèrie les dades que provenen del sensor.

Agafa la part més significativa de cadascun dels 2 bytes que li arriben del sensor (deprecia la part menys significativa) i la converteix a un valor decimal.

Un cop ja s'ha obtingut la temperatura, l'ordinador obté l'hora actual, i guarda el valor de la temperatura, el dia i l'hora pertanyent a l'instant de la temperatura en un arxiu de text. El programa, verifica si l'arxiu de text està creat o no. Si no ho està el crea, i si ho està, comprova si hi ha dades escrites. Si n'hi hagués, escriuria els valors després de les dades que ja hi ha escrites, d'aquesta manera evita que es destrueixin dades que puguin haver-hi al fitxer, i destruir els valors anteriors. Si el fitxer estigués en blanc, guardaria les dades al seu inici.

Les dades es guarden en el següent format: Data i hora ; Temperatura.

A continuació es mostra un exemple d'unes línies del fitxer de dades:

```
1/16/2010 5:57:30 PM;26.23
1/16/2010 6:07:33 PM;25.99
1/16/2010 6:17:36 PM;25.93
1/16/2010 6:27:38 PM;25.64
```

Un cop les dades ja estan en el format correcte perquè l'aplicació web les pugui interpretar i mostrar correctament, s'ha d'actualitzar el fitxer de dades al servidor FTP per tal que la visualització del gràfic sigui amb les dades actualitzades. El que farà el programa, serà executar un fitxer .bat que s'encarregarà d'actualitzar les dades. Des del codi C# s'executa un fitxer de la següent manera:

```
Process.Start("hdh.bat");
```

Aquesta línia de codi, executa el fitxer hdh.bat.

L'ordinador que rep les dades s'ha d'encarregar, un cop adaptades al format correcte, d'enviar-les al servidor FTP per tal que es pugui actualitzar el gràfic automàticament.

S'ha aconseguit mitjançant 2 arxius:

- **hdh.bat:** Aquest fitxer conté el següent codi: `ftp -s:ftp.txt`
Aquesta línia de codi, fa que s'executi una consola de FTP amb els paràmetres que contenen a l'arxiu que precedeix la comanda `ftp -s:`

(en aquest cas, el fitxer ftp.txt). Aquest arxiu és executat des del programa en C# cada cop que té una dada nova.

- **ftp.txt:** l'arxiu *.txt (en aquest cas anomenat ftp.txt) conté les dades de sessió, i la ubicació dels arxius que s'han d'enviar al servidor FTP.

Codi de l'arxiu ftp.txt:

```
open ***** (Direcció del servidor FTP)
*****      (Nom d'usuari)
*****      (Paraula de pas)
bin         (Transferència de fitxer binari)
lcd C:\Documents and Settings\Admin\Escritorio (Directori local)
cd /htdocs/amline (Directori del servidor FTP)
put amline_data.txt (Arxiu que s'envia al servidor FTP)
bye        (Es tanca la sessió)
```

4.5.3. Interfície WEB

Un cop les dades estan en el format desitjat, ja es poden pujar al servidor FTP, de manera que s'actualitzi el gràfic automàticament.

Per mostrar els valors al gràfic s'ha escollit el software *Line & Area* d'*amCharts.com* (<http://www.amcharts.com/line>). 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, tampany...). 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 on 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.
- **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.

S'ha generat 2 tipus de gràfics, un que mostra la temperatura d'un sol sensor, i l'altre que mostra la temperatura dels 4 sensors utilitzats en el sistema (aquest número és ampliable).

El gràfic de tots els sensors junts presenta l'avantatge que amb un sol gràfic es pot visualitzar la informació de temperatura de tots els sensors, però l'inconvenient que ens obliga a mostrear tots els sensors a la vegada per tal de no obtenir cap valor d'abscisses nul.

4.5.3.1. Gràfic de temperatura d'un sol sensor

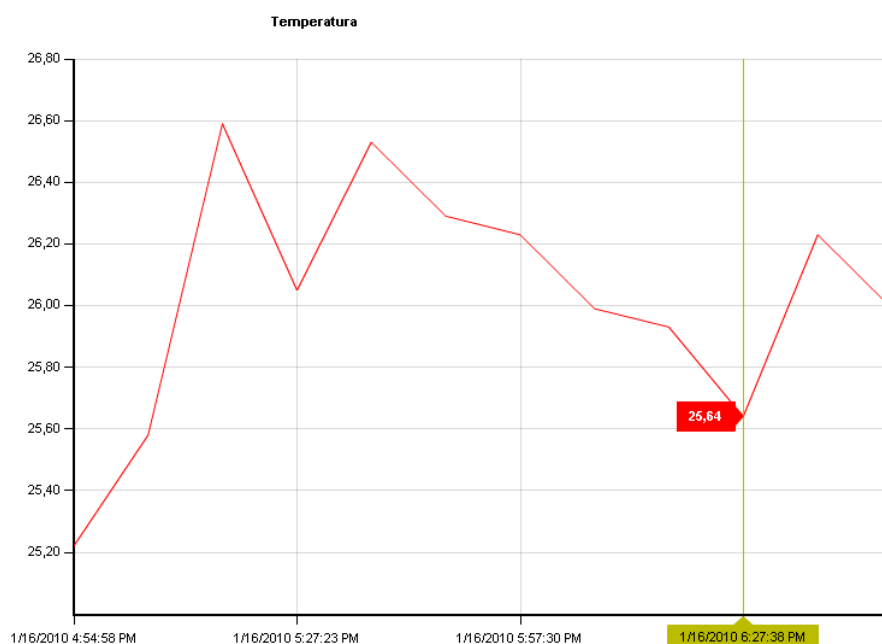


Fig. 4.10 Gràfic de temperatura de la pàgina web

A la imatge anterior, es pot veure el gràfic amb unes. Al posar el cursor sobre d'un punt, es mostren els valors de temperatura i instant de temps.

També es pot ampliar una part del gràfic seleccionant amb el ratolí el tros concret. aquesta opció pot ser molt útil per apreciar algun punt en concret quan hi hagi un nombre elevat de dades al gràfic.

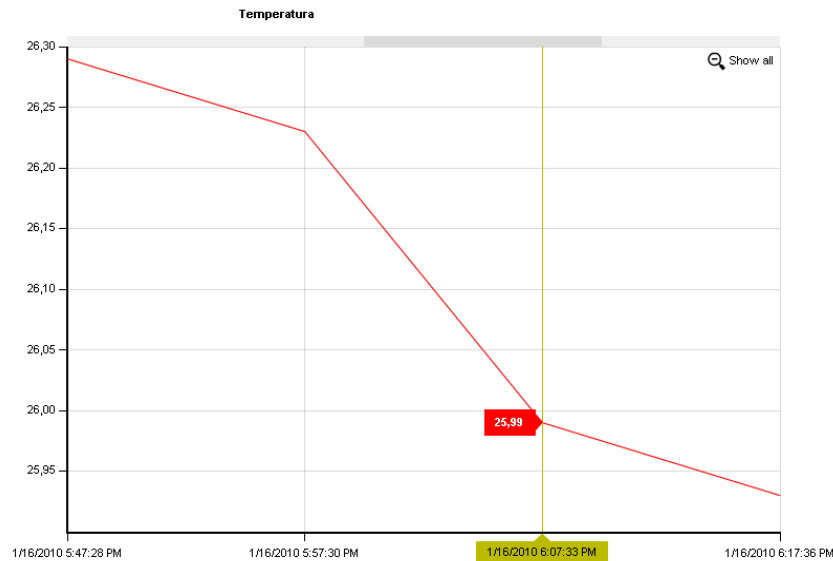


Fig. 4.11 Ampliació d'una part del gràfic

A la figura anterior, es pot veure una part del gràfic ampliat, i a la part superior del gràfic apareix una barra pel seu desplaçament a la totalitat de les dades. Per tornar el gràfic a la seva posició inicial, es clica el botó *Show all* que apareix a l'extrem superior dret del gràfic, i tornen a quedar totes les mostres visibles.

4.5.3.2. Gràfic de temperatura de tots els sensors

Aquest gràfic, té les mateixes característiques que l'anterior referent al zoom i al desplaçament, però afegeix la llegenda i l'opció de seleccionar un o varis gràfics, i també de poder activar o desactivar mitjançant la llegenda les bombolles amb el valor.

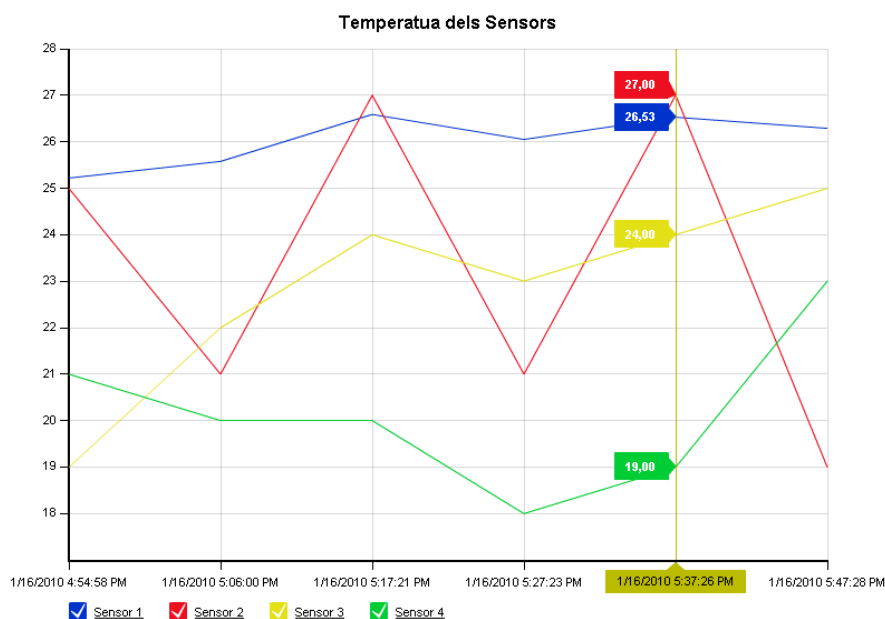


Fig. 4.12 Ampliació d'una part del gràfic

A la llegenda, ubicada a la part inferior del gràfic, si es clica a la icona de cada sensor, s'activa o es desactiva la línia del gràfic, i si es clica el text de la llegenda, s'activa o es desactiva l'etiqueta del valor del sensor en qüestió.

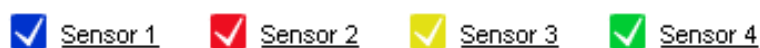


Fig. 4.13 Llegenda

L'arxiu de dades del gràfic dels 4 sensors, segueix el mateix format que el d'un sol sensor:

Data i hora ; Temperatura Sensor 1 ; Temp sensor 2 ... ; ...Temp sensor N

A continuació es mostra un exemple d'unes línies del fitxer de dades:

```

1/16/2010 4:54:58 PM;25.22;25;19;21
1/16/2010 5:06:00 PM;25.58;21;22;20
1/16/2010 5:17:21 PM;26.59;27;24;20
1/16/2010 5:27:23 PM;26.05;21;23;18
1/16/2010 5:37:26 PM;26.53;27;24;19
1/16/2010 5:47:28 PM;26.29;19;25;23
    
```

Els gràfics es troben ubicats a les següents pàgines web:

<http://temperatura.zobyhost.com/amline.html>
<http://temperatura.zobyhost.com/graf/grafic2.html>

CAPÍTOL 5. CONSUM

Aconseguir una solució domòtica de baix consum és un dels objectius esmentats en el punt 1.1. Per aquest motiu, en aquest apartat es comparen els valors teòrics de consum dels nodes utilitzats amb els valors reals. Es tracta de validar els valors de referència del fabricant.

Aquesta validació es realitzarà mitjançant l'analitzador de potència Agilent N6705A.

5.1. Agilent N6705A

L'analitzador de potència Agilent N6705A és un aparell de mesura que permet obtenir els mateixos valors de tensió o corrent que s'obtenen mitjançant un oscil·loscopi convencional, però que a més té la capacitat de mesurar el consum de potència de diferents dispositius.



Fig. 5.1 Agilent N6705A

Disposa de quatre mòduls de potència independents que permeten mesurar amb més o menys precisió aquests valors. En l'analitzador utilitzat, es disposa de dos mòduls.

En la imatge següent es poden observar les diferents preses de cada mòdul.



Fig. 5.2 Fonts d'alimentació

Una altra característica d'aquest analitzador és que disposa de port USB per a l'emmagatzematge de les captures realitzades.

5.2. Mesures

Per a realitzar les verificacions, s'ha utilitzat el mòdul 2, que és del tipus Agilent N6774A, considerat pel fabricant com un model de font d'alimentació d'alt rendiment.

Les característiques d'aquest mòdul són les següents:

Taula. 5.1 Característiques del mòdul 2.

Paràmetre a mesurar	Valor màxim
Tensió	35 V
Corrent	8.5 mA
Potència	300 W

Abans de mesurar la potència de cada sensor, cal configurar dos paràmetres: la tensió i la corrent.

En aquest cas, s'escullen els valors següents:

- Tensió: 3 V
- Corrent: 200 mA

Això es fa prement el botó *Settings*, mitjançant el qual s'accedeix al menú de configuració. Un cop allà es seleccionen els valors mitjançant el teclat numèric.

Un cop establerts aquests valors, s'activa la font de tensió escollida, en aquest cas, el mòdul 2, prement el botó ON, situat damunt de les preses d'alimentació com mostra el requadre de la figura 5.2.

Per tal de connectar els sensors a l'analitzador s'ha fet mitjançant cables banana-banana. Aquests amb unes pinces a l'extrem acoblades al sensor han permès prendre les diferents mesures.

Degut a les reduïdes dimensions del sensor i a la dificultat de connectar les dues pinces en aquest, al moment de prendre les mesures es van soldar uns cables als pols positiu i negatiu per tal de facilitar aquesta connexió.

En la figura següent es mostra aquesta configuració:

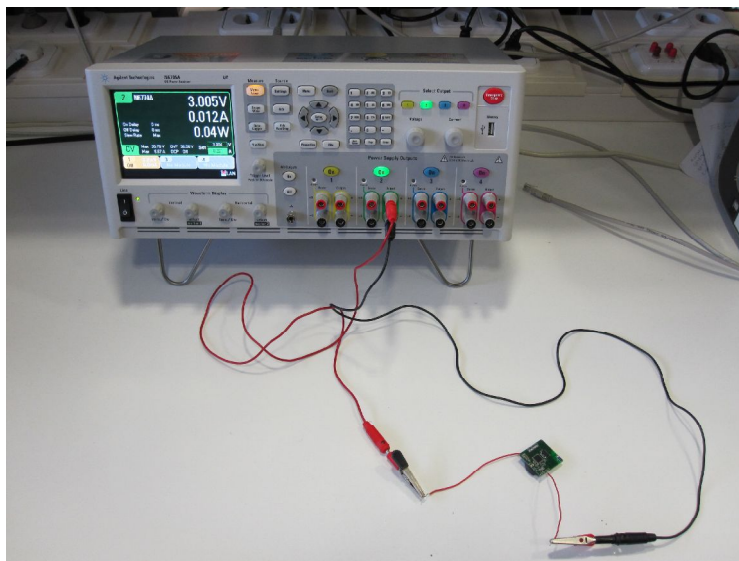


Fig. 5.3 Connexió analitzador-sensor

En el *datasheet* del CC2431 [4] s'especifiquen les dades de consum següents:

Taula. 5.2 Dades de consum teòriques

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.

Les dades de la taula anterior serviran per a comparar els valors teòrics que especifica el *datasheet* amb els valors mesurats amb l'analitzador.

Per tal de realitzar aquestes mesures s'ha tingut en compte l'escenari següent:

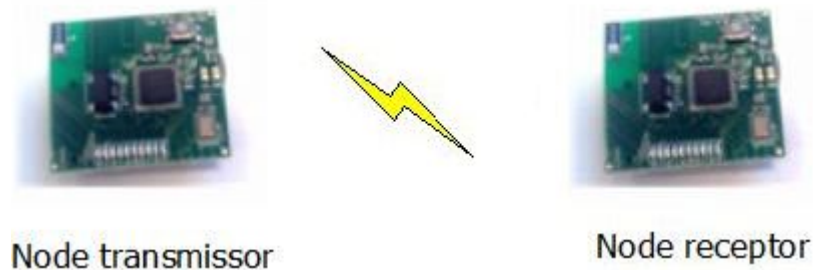


Fig. 5.4 Escenari de proves

Tal com s'aprecia en la figura anterior, es crea una xarxa formada per 2 nodes: un node actua de transmissor i envia missatges periòdicament (un cada segon) a un node receptor.

Tots els nodes estan configurats en Power Mode 1, és a dir, tenen un consum mínim de 296 μA , quan es troben a l'espera de rebre un missatge

En els següents apartats es detallen les mesures realitzades depenent del tipus de node que es tracti: transmissor o receptor.

5.2.1. Transmissor

En aquest apartat es mesura el consum del node transmissor quan està enviant missatges al node receptor.

En la següent figura es mostra el resultat de la mesura:

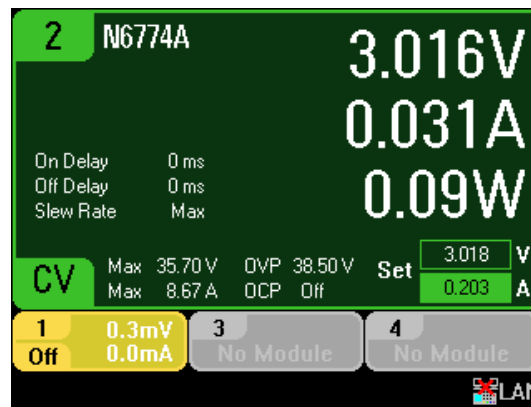


Fig. 5.5 Captura del consum del transmissor

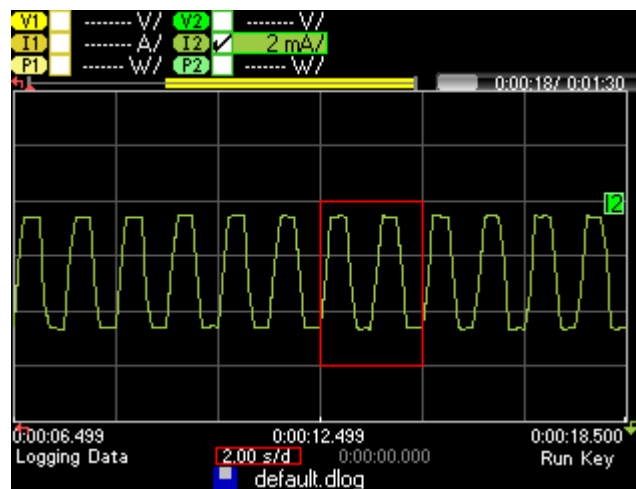


Fig. 5.6 Captura del consum del transmissor

La figura 5.6 mostra el consum del node emissor. Es pot apreciar que l'escala temporal seleccionada és 2 segons per divisió, i al requadre (de 2 segons) hi ha justament 2 períodes, això implica que la freqüència d'enviament de missatges és d'1 Hz.

El consum del transmissor és de 31 mA. Quan un node està transmetent, el seu consum teòric es correspon amb el MCU Active and TX Mode de la taula 5.2, que és de 24.7 mA.

5.2.2. Receptor

En el cas del receptor, la mesura s'ha pres mentre el node estava rebent els missatges que se li enviaven.

En la figura següent es mostra els valors mesurats:

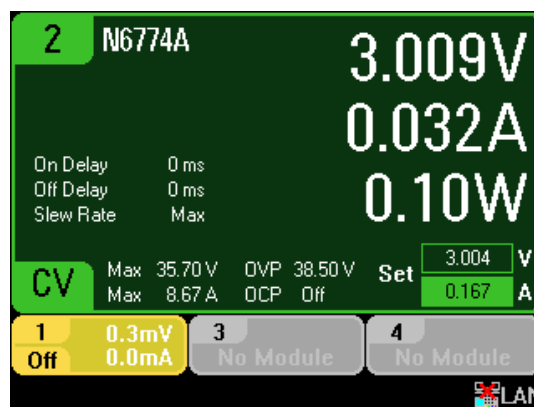


Fig. 5.7 Captura del consum del receptor

El resultat de la mesura indica que el receptor consumeix 32 mA, mentre que el valor teòric que s'especifica en la taula 5.2, quan el node està rebent, és a dir, en MCU Active and RX Mode és de 27 mA.

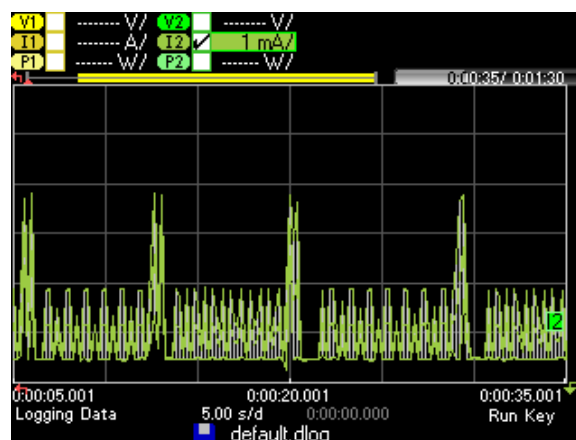


Fig. 5.8 Captura del consum del receptor

La figura 5.8 correspon al consum del receptor. Es pot apreciar clarament quan el dispositiu està en espera, i l'augment de consum quan aquest rep un paquet.

5.2.3. Conclusions de les verificacions

En aquest apartat s'extreuen les conclusions dels resultats de les mesures anteriors.

En el cas del transmissor, es pot comprovar com el valor real és un 26% major que el valor teòric.

Aquesta diferència és deguda a què en el node sensor, s'inclouen altres elements a part del circuit integrat CC2430, com poden ser els díodes LED, que n'augmenten el consum.

Pel que fa al receptor, la diferència entre el valor real i el teòric no és tant gran, és de l'ordre del 18%.

També s'aprecia un increment del consum, i és degut al mateix factor que en cas del transmissor.

De l'observació de la figura 5.8 se'n pot concloure que el consum del node en estat de recepció és més del doble que en estat de repòs. Així, l'autonomia del node, dependrà molt del tràfic que circuli a través d'ell.

CAPÍTOL 6. CONCLUSIONS

En aquest treball fi de carrera s'ha configurat una xarxa de nodes sensors sense fils que s'ha aplicat a l'obtenció de la temperatura.

S'han dissenyat i fabricat els mòduls addicionals que s'acoblen al node: receptor d'infraroig, emissor d'infraroig i teclat. Així com una placa per tal de poder connectat els dispositius al node sensor.

També s'ha dissenyat un comandament per tal de poder configurar i inicialitzar els diferents nodes, així com fer peticions i actuar sobre la xarxa.

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.

El disseny de la xarxa està totalment obert i no és gens estàtic, ja que es pot ampliar i disminuir sense cap mena de problema el nombre de nodes de la xarxa. Els mòduls també estan preparats per a possibles ampliacions de funcionalitats.

Les dades de temperatura que mesuri el sensor, són mostrades a través d'una interfície web.

S'ha adaptat un protocol de reenviament de missatges per tal que la informació arribi a tots els nodes, encara que no hi hagi comunicació directa entre tots.

Aquest protocol és suficientment robust com per descartar els missatges duplicats, evitar reenviaments innecessaris i el bloqueig de la xarxa.

Els objectius de disseny i fabricació dels mòduls addicionals dels nodes, s'ha complert tot i que no s'ha pogut verificar el seu funcionament.

Els sensors contesten a totes les peticions que se'ls demanen, i reenvien els missatges correctament, per tant compleixen amb tots els requisits que estan descrits a l'escenari (figura 1.1)

6.1. Línies futures

Com s'ha pogut apreciar a la memòria, l'àmbit d'aquest treball fi de carrera és força ampli, cobrint des de la realització de hardware com a la programació amb diferents entorns propis de sensors i PCs. Tot plegat ha fet que hagin quedat temes pendents i d'altres a millorar.

Això fa que aquest projecte pugui ser continuat per molts dels seus aspectes, i a continuació es comenten els més rellevants.

- **Interfície web:** Per tal de fer el sistema molt més proper i fàcil de cara l'usuari, seria interessant, desenvolupar una interfície web, per poder interactuar al sistema remotament, tan per visualitzar les dades, com per actuar sobre el propi sistema. Les peticions les continuaria realitzant el node que fa la funció de *gateway* que continuaria connectat a un ordinador per port sèrie però l'entorn seria molt més amigable.
- **Posta en marxa del hardware:** Tot el hardware que s'ha dissenyat a aquest projecte, seria interessant posar-lo en marxa, per comprovar si realment és fàcil per l'usuari interactuar amb el sistema, o és preferible fer alguna modificació o ampliació de les possibilitats (per exemple, la botonera del comandament). També seria convenient desenvolupar un encapsulat pel hardware dissenyat.
- **Estudi profund del consum:** A aquest projecte s'ha contemplat sempre la prioritat d'estalvi energètic davant de qualsevol altra, sobretot a l'hora de fer funcionar els modes dels xips, al triar els components... Seria interessant fer una estimació de la vida útil d'una bateria en un node integrat a una xarxa i funcionant. També es podria estudiar el cost energètic que tindria si la interfície d'infraroig estigués encesa sempre.
- **Completar el hardware amb un interruptor d'estat sòlid i un xip de mesura del consum:** Degut a la versatilitat del sistema, es poden acoblar diferents mòduls als sensors. Uns importants de dissenyar, serien el xip de mesura de corrent, que donaria la informació del consum dels aparells connectats a l'endoll on es trobaria el sensor, i un interruptor d'estat sòlid que podria obrir o tancar un circuit, això faria que el sistema podria apagar o encendre tots els aparells connectats a l'endoll.
- **Diferents tipus de peticions:** A aquest projecte, bàsicament s'ha treballat amb les peticions i respostes de la temperatura que mesuren els sensors, però el sistema ha de suportar tot tipus de missatges i respostes, a mesura que es vagi incorporant nou hardware (per exemple interruptors d'estat sòlid) i funcionalitats addicionals.
- **Longitud flexible del camp de dades:** Seguint el punt anterior, caldria un nou camp, a la capçalera, amb la longitud del camp de dades, ja que depenent de les dades que transporti, seran necessaris més o menys bits de dades. Això faria el sistema molt més flexible.
- **Optimització de codi:** S'ha dissenyat el codi de manera que compleixi els requisits desitjats i faci les funcions demanades, però la declaració de variables, crides a funcions, tipus de variables utilitzades... podrien canviar-se per tal d'obtenir un cost computacional més baix, donat que aquest xip no disposa d'una gran potència de processat.

BIBLIOGRAFIA I REFERÈNCIES

- [1] Estàndard IEEE 802.15.4:
<http://standards.ieee.org/getieee802/download/802.15.4a-2007.pdf>
- [2] Peris Bespín, E; *Implementación de una solución domótica basada en sensores*. Juny de 2009
<http://upcommons.upc.edu/pfc/bitstream/2099.1/8290/1/memoria.pdf>
- [3] CC2430: <http://focus.ti.com/lit/ds/symlink/cc2430.pdf>
- [4] CC2431: <http://focus.ti.com/docs/prod/folders/print/cc2431.html>
- [5] TLC555: <http://focus.ti.com/docs/prod/folders/print/tlc555.html>
- [6] TSOP1836:
<http://www.alldatasheet.com/datasheet-pdf/pdf/26603/VISAY/TSOP1836.html>
- [7] TSOP34836: <http://www.farnell.com/datasheets/30492.pdf>
- [8] EDE1144: <http://www.jameco.com/Jameco/Products/ProdDS/171969.pdf>
- [9] MM74C922:
<http://www.ucm.es/info/electron/laboratorio/componentes/MM74C922.pdf>
- [10] PIC16F627A:
<http://ww1.microchip.com/downloads/en/DeviceDoc/40044G.pdf>
- Giribert Peraire, J ; Madrid Villar, M. *Xarxa cel·lular de 4G basada en IPv6: desenvolupament d'un demostrador (II)*
http://upcommons.upc.edu/pfc/bitstream/2099.1/6906/7/TFC_FINAL.pdf
- Design Note DN102: <http://focus.ti.com/lit/an/swra101a/swra101a.pdf>
- Design Note DN112: <http://focus.ti.com/lit/an/swra222b/swra222b.pdf>



**Escola Politècnica Superior
de Castelldefels**

UNIVERSITAT POLITÈCNICA DE CATALUNYA

ANNEXOS

TÍTOL DEL TFC: Implementació d'una solució domòtica basada en sensors (II)

TITULACIÓ: Enginyeria Tècnica de Telecomunicació, especialitat Sistemes de Telecomunicació

**AUTORS: Adrià Dasquens Carulla
Anna Ortiz Mazarico**

**DIRECTORS: Lluís Casals Ibáñez
Rafael Vidal Ferré**

DATA: 2 de febrer de 2010

ANNEX I SISTEMA DE DIFUSIÓN PARA CC2430N

Aquest annex és un document creat per l'estudiant Eric Peris. Explica i mostra el format dels missatges que s'envien els nodes i el protocol de reenviament que utilitzen.



Escola Politècnica Superior
de Castelldefels

UNIVERSITAT POLITÈCNICA DE CATALUNYA

Sistema de difusión para CC2430N

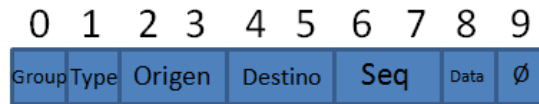
INDEX

1. CAPÍTULO 1. FORMATO DEL MENSAJE Y CRITERIO DE DIFUSIÓN	69
1.1. Formato del mensaje	69
1.2. Criterio de difusión	69
2. CAPÍTULO 2. LOLLIPOP	71
2.1. Introducción	71
2.1.1. Ejemplo	71
2.2. Abstracción del código	72
2.3. Código implementado	74

CAPÍTULO 1. FORMATO DEL MENSAJE Y CRITERIO DE DIFUSIÓN

Formato del mensaje

Los mensajes tienen una longitud de 10 bytes. El formato es el siguiente:



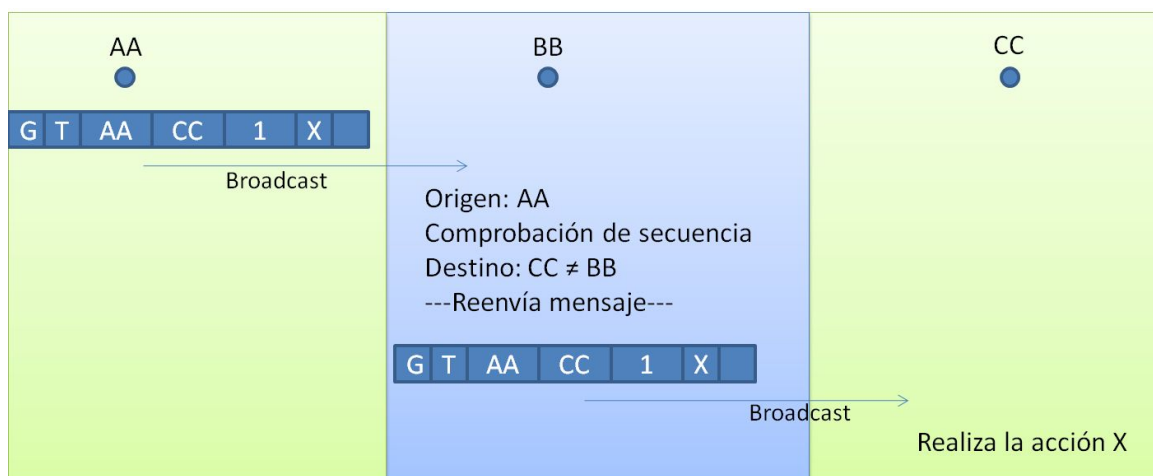
A saber:

- **Group:** es el PAN ID
- **Type:** tipo de mensaje (MSG o RESTART). Más información en el apartado 2.2.
- **Origen:** Dirección de origen del mensaje.
- **Destino:** Dirección de destino del mensaje.
- **Seq:** número de secuencia.
- **Data:** byte de datos
- **Byte 9:** actualmente no utilizado.

Criterio de difusión

El sistema está pensado para enviar un mensaje de un nodo a otro sin que estos tengan cobertura directa. Es decir, es necesaria la existencia de uno o varios nodos que reenvíen el mensaje.

En la siguiente imagen se observa un esquema simplificado del funcionamiento del sistema. Sean AA, BB y CC tres nodos, y el color de fondo su zona de cobertura. De forma que AA y CC no tienen contacto directo.



El mensaje es reenviado por el nodo B puesto que ese mensaje no iba dirigido a él. Además, se utiliza un número de secuencia que ayuda a determinar la

antigüedad del mensaje. De forma que se evita reenviar mensajes repetidos entre nodos.

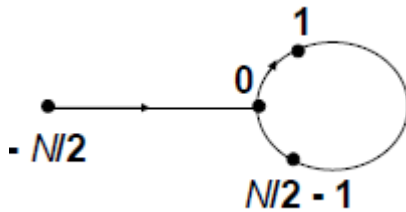
El modo elegido para tratar los números de secuencia ha sido el del tipo Lollipop.

CAPÍTULO 2. LOLLIPOP

Introducción

Los dispositivos transmiten junto con cada mensaje un número de secuencia que se va incrementando conforme se van enviando los mensajes.

Existen dos regímenes diferentes:



De $-N/2$ a 0: régimen de arranque
De 0 a $N/2$: régimen permanente

Una vez dentro del régimen permanente, para considerar cuándo un número de secuencia es más nuevo que el anterior, se mirará la distancia entre las dos secuencias y se comparará con la mitad del recorrido del régimen permanente ($N/4$).

Sea "a" la última secuencia guardada del nodo "A" y sea "b" la secuencia recibida del mismo nodo.

Si $a > b$ y $|b - a| > N/4$ implica que la secuencia recibida "b" es más nueva que la secuencia guardada "a".

Para el caso contrario se considera que si $a < b$ y $|b - a| < N/4$ la secuencia recibida "b" es más nueva que la secuencia guardada "a".

Ejemplo

Este sistema se diseñó básicamente por un motivo:

Supongamos:

$$"-N/2" = 0$$

$$"0" = 500$$

$$"N/2 - 1" = 999$$

Y que los dispositivos llevan tiempo en marcha y enviándose mensajes.

El dispositivo A transmite un mensaje con la secuencia 554.

El dispositivo B coge el valor recibido (554) y lo compara con el que había recibido por última vez (553). Como esta secuencia es mayor que la anterior, se interpreta el mensaje leyéndose los datos que contiene.

Ahora supongamos que A se reinicia inesperadamente.

El dispositivo A transmite un mensaje con la secuencia 1.

B coge el valor recibido (1) y lo compara con el que había recibido por última vez (554).

Como detecta que A está en régimen de arranque. B le envía un mensaje a A con el último número de secuencia que tenía guardado. Y elimina el paquete que acababa de recibir.

A recibe el mensaje y establece su valor de secuencia como $554+1$ (555). Acto seguido, se ha establecido que para este caso concreto se vuelva a enviar el mensaje con el número de secuencia correcto.

B coge el valor recibido (555) y lo compara con el que había recibido por última vez (554).

Como esta secuencia es mayor que la anterior, se interpreta el mensaje leyéndose los datos que contiene.

Abstracción del código

El código está implementado de forma que para evitar valores negativos:

$$-N/2 = 0$$

$$0 = \text{seq_umbral} = 500$$

$$N/2-1 = 2*\text{seq_umbral}-1 = 999$$

$$N/4 = \text{seq_umbral}/2 = 255$$

De 0 a 499 (incluido) se considera arranque. De 500 a 999 (incluido) se considera régimen permanente.

Para empezar, existen dos tipos (TYPE) de mensajes:

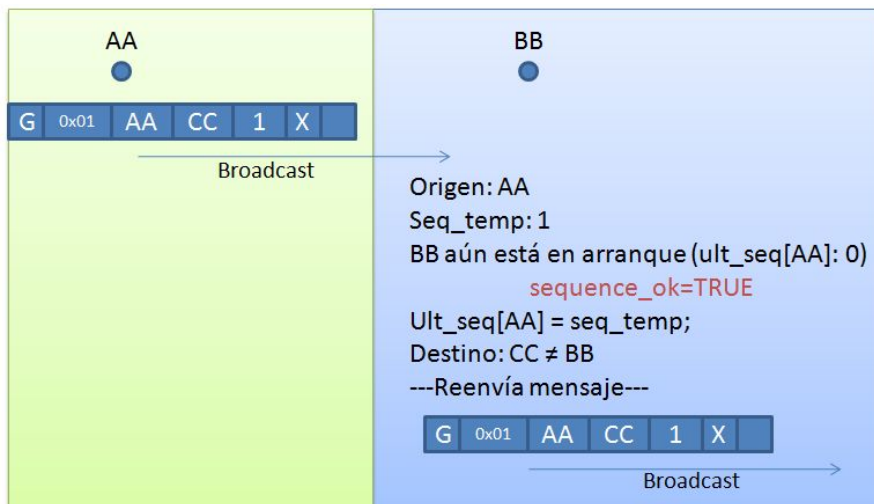
- MSG (0x01):** indica que se está enviando un mensaje (con el nodo de origen, nodo destino, secuencia, datos, etc...).
- RESTART (0x02):** indica al receptor que su número de secuencia es incorrecto y deberá tener en cuenta la secuencia que le envíen.

Además, el mensaje solo se interpreta si la secuencia es correcta. Cosa que se indica con el booleano: *sequence_ok*.

Al recibir un mensaje de tipo *MSG* (broadcast), lo primero que se hace es obtener la secuencia recibida.

Se observan los 4 casos que pueden pasar:

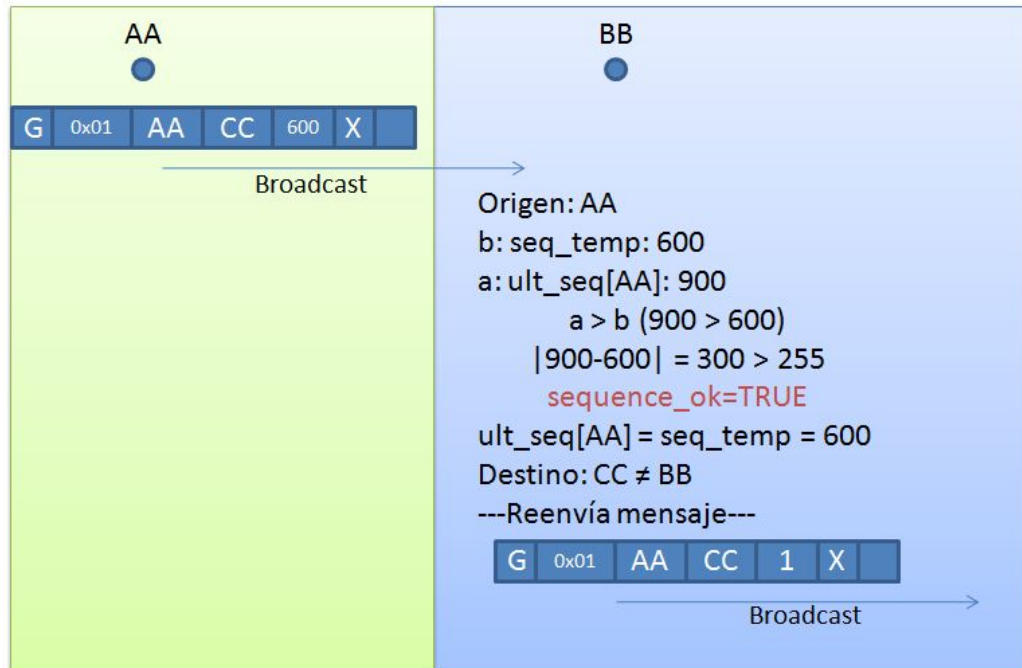
- 1) *BB* se encuentra en régimen de arranque. Da igual la secuencia que reciba



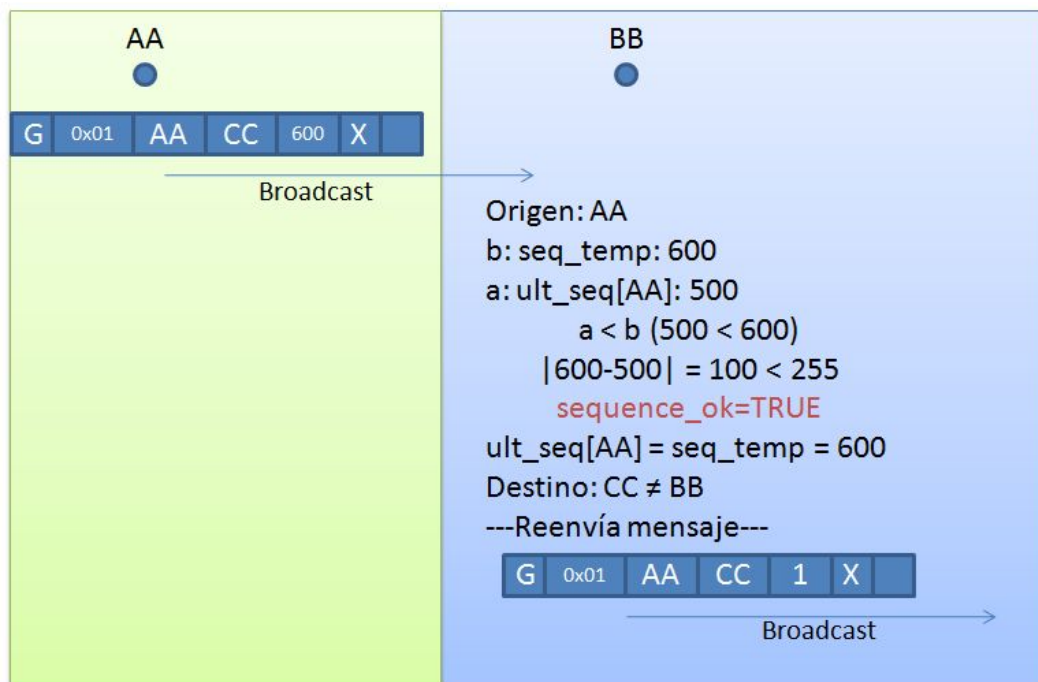
de AA, ya que simplemente se guardará la secuencia que se le esté enviando.

2) Ambos en régimen permanente. Sea “a” la última secuencia guardada del nodo “AA” y sea “b” la secuencia recibida del mismo nodo.

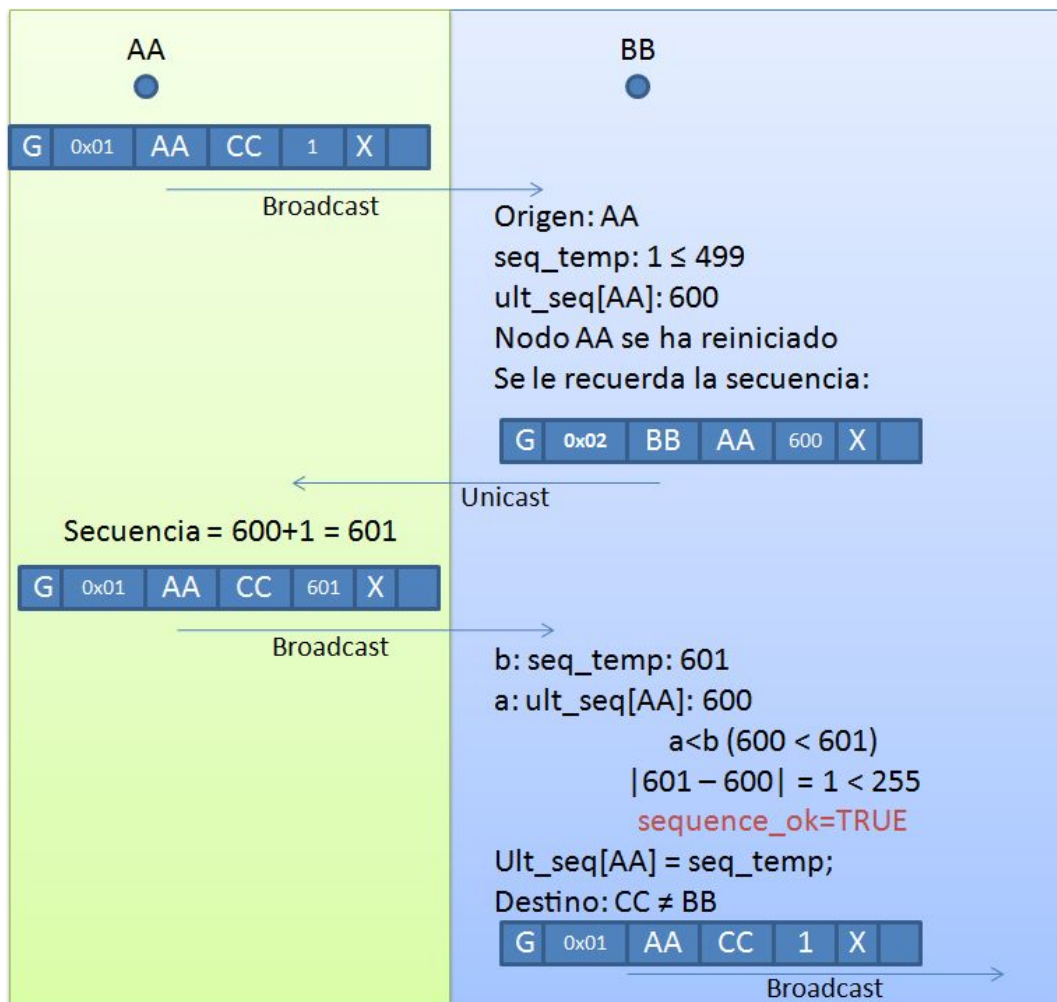
a) Si $a > b$ y $|b - a| > N/4$: la secuencia recibida **b** es más nueva que la secuencia guardada **a**.



b) Si $a < b$ y $|b - a| < N/4$: la secuencia recibida **b** es más nueva que la secuencia guardada **a**.



- 3) El nodo *AA* se acaba de reiniciar. El nodo *BB* le envía un mensaje unicast recordándole la secuencia por la que iba el nodo *AA* antes de reiniciarse.



- 4) Si ninguno de los casos anteriores se cumple, quiere decir que la secuencia es incorrecta. Y por tanto el mensaje es descartado, de forma que la variable "*sequence_ok*" no se activa en ningún momento.

Código implementado

Variables a comentar:

ultimos_rx[addr_origen-1].seqnum: Guarda la última secuencia recibida del nodo "addr_origen".

sequence: es el valor de secuencia que envía el transmisor en cada mensaje.

sequence_tmp: es el valor de secuencia que se obtiene tras leer un mensaje.

msg_prueba[i]: es el vector con el mensaje que se va enviar.

En "msa.c":

```

if (events & SYS_EVENT_MSG)
{
    [...]
    switch ( *pMsg )
    {
        [...]
        case MAC_MCPS_DATA_IND:
        [...]
            if(type == MSG) //recibimos un mensaje de tipo MSG
            {
//obtenemos del mensaje recibido la dirección de destino, origen y numero de secuencia.
                addr_destino = BUILD_UINT16(pData->dataInd.msdu.p[2], pData->dataInd.msdu.p[3]);
                addr_origen = BUILD_UINT16(pData->dataInd.msdu.p[4], pData->dataInd.msdu.p[5]);
                sequence_tmp = BUILD_UINT16(pData->dataInd.msdu.p[6], pData->dataInd.msdu.p[7]);

//Comprobamos la secuencia recibida según los dos casos posibles:
                /*Caso 1: El nodo se acaba de reiniciar y le indicamos la secuencia por la que iba
                antes de resetearse*/
                if((sequence_tmp < seq_umbral) && (ultimos_rx[addr_origen-1].seqnum >= seq_umbral))
                {
                    /* Group */
                    msg_prueba[0] = pData->dataInd.msdu.p[0];
                    /* Type */
                    msg_prueba[1] = RESTART; //el mensaje a enviar será del tipo RESTART
                    /* Addr destino */ //la dirección de destino pasa a ser la dirección de origen
                    del mensaje recibido
                    msg_prueba[2] = pData->dataInd.msdu.p[4];
                    msg_prueba[3] = pData->dataInd.msdu.p[5];
                    /* Addr origen */ //incluimos el nodo que está avisando del error
                    msg_prueba[4] = LO_UINT16( msa_DevShortAddr );
                    msg_prueba[5] = HI_UINT16( msa_DevShortAddr );
                    /* Seqnum */ //añadimos el ultimo valor de secuencia conocido por este nodo
                    msg_prueba[6] = LO_UINT16( ultimos_rx[addr_origen-1].seqnum );
                    msg_prueba[7] = HI_UINT16( ultimos_rx[addr_origen-1].seqnum );

                    //Enviamos el mensaje al dispositivo que se acaba de resetear.
                    MSA_McpsDataReq((uint8*)msg_prueba,
                                    MSA_PACKET_LENGTH,
                                    TRUE,
                                    addr_origen);
                }

//Caso 2: el nodo se encuentra en régimen permanente
            else
            {
                /*Si el ultimo valor que se tenia guardado pertenecia al arranque, usamos el
                nuevo*/
                if(ultimos_rx[addr_origen-1].seqnum < seq_umbral)
                    sequence_ok=TRUE;
                /*Si a>b y |b-a|>N/4 => b mas nuevo que a*/
                else if((ultimos_rx[addr_origen-1].seqnum > sequence_tmp) && (abs(sequence_tmp -
                    ultimos_rx[addr_origen-1].seqnum) > seq_umbral/2))
                    sequence_ok=TRUE;
                /*Si a<b y |b-a|<N/4 => b mas nuevo que a*/
                else if((ultimos_rx[addr_origen-1].seqnum < sequence_tmp) && ( abs(sequence_tmp
                    - ultimos_rx[addr_origen-1].seqnum) < seq_umbral/2))
                    sequence_ok=TRUE;
                /*Si no ocurre nada, significa que la secuencia es incorrecta (directamente
                salta a dspues del if(sequence_ok))*/
            }
        }
    }
}

```

