

Control d'incendis basat en tecnologia WSN

Autor: Javier Díaz Espejo
Enginyeria Tècnica en Informàtica de Sistemes

Consultor: Jordi Bécares Ferrés

12 de Juny de 2012

Agraïments

Aquest projecte representa la culminació d'una etapa de la meua vida, la qual va començar fa ja uns quants anys enrere, i que he pogut portar a terme gràcies al suport de la meua família i parella, que han sigut capaços de comprendre tot el temps que he necessitat dedicar per a poder arribar a l'objectiu que representa aquest treball. Per això, dedico aquest projecte als meus pares, a les meves germanes, a la meua recent nascuda neboda i a la meua parella que ha estat amb tot moment al meu costat.

Resum

El projecte tracta de desenvolupar un sistema de detecció d'incendis, utilitzant la tecnologia de sensors sense fils (Wireless Sensors Network). La utilització d'aquests petits dispositius ens introdueix en el món dels sistemes encastats, un àrea en ple desenvolupament industrial, i que la trobem en l'actualitat incorporada en molts productes de consum de la nostra vida quotidiana.

Els sensors WSN són petits dispositius de recursos limitats, coneguts com a "motes". Estan integrats bàsicament per un micró-controlador, un xip encarregat de la comunicació via ràdio i una interfície de comunicació via sèrie. Com a elements opcionals les motes incorporen diferents tipus de sensors: lectura de bateria, temperatura, magnetisme, de pressió atmosfèrica, de moviment...

El sistema a desenvolupar consta d'una xarxa de sensors de tipologia en estrella, en la qual existeix un sensor que realitza la funció d'estació base i que està connectat amb un ordinador. La resta de sensors es comuniquen únicament via ràdio amb l'estació base. La utilització d'un ordinador personal serveix per a l'execució d'una aplicació que gestiona el sistema de sensors i que realitza la funció d'interfície per a que l'usuari interactuï amb el sistema.

El desenvolupament del sistema basat amb WSN permet obtenir un conjunt d'avantatges respecte a un sistema de detecció d'incendis tradicional. L'absència d'instal·lació de cablatge comporta un estalvi econòmic molt important, i en certes aplicacions ens serveix per salvar barreres físiques que en cas contrari són insalvables; el baix cost econòmic dels sensors és un afegit. Per tant, tenim un sistema segur, econòmic i ràpid d'instal·lar, característiques que fan molt atractiva la implementació del sistema.

Paraules clau: seguretat, sistemes encastats, WSN, motes, robustesa, fiabilitat, costos, optimització, monitorització, futur.

Àrea del projecte: Sistemes encastats.

Abstract

The project seeks to develop a fire detection system using wireless sensor technology (Wireless Sensor Network.) The use of these small devices will enter the world of embedded systems, a booming industrial area, and that are now incorporated in many consumer products of daily life.

WSN sensors are small devices with limited resources, known as "motes". They are made primarily by a micro-controller, a chip in charge of communication and radio communication via a serial interface. Optional elements as the motes incorporate different types of sensors: reading, battery temperature, magnetism, atmospheric pressure, movement...

The developed system consists of a sensor network in a star-type, in which there is a sensor that performs the function of base station and is connected with a computer. The other sensors are connected only via radio base station. Using a personal computer used for running an application that manages the sensor system and performs the function interface that the user interacts with the system.

The development of WSN-based system allows to obtain a set of advantages over a traditional fire detection system. The absence of cabling installation involves a very significant cost savings, and in some applications we used to save physical barriers that are insurmountable otherwise, the low cost of sensors is added. Therefore, we have a safe, economical and quick to install, very attractive features that make the system implementation.

Keywords: security, embedded systems, WSN, specks, robustness, reliability, cost, optimization, monitoring, future.

Project Area: Embedded Systems.

Índex de continguts

Capítol 1. Introducció.....	1
1.1 Justificació.....	2
1.2 Descripció.....	2
1.3 Objectius	3
1.4 Enfocament.....	5
1.5 Planificació.....	5
1.6 Recursos emprats.....	9
1.7 Productes obtinguts.....	9
1.8 Breu descripció dels altres capítols.....	10
Capítol 2. Antecedents.....	11
2.1 Estat de l'art.....	11
2.1.1 Definició de sistemes encastats.....	11
2.1.2 Components d'un sistema encastat.....	11
2.1.3 Arquitectura bàsica d'un sistema encastat.....	12
2.1.4 Característiques dels sistemes encastats.....	15
2.1.6 Exemples d'us de sistemes encastats.....	16
2.1.7 Concepte de família.....	18
2.1.8 Xarxes de sensors sense fils.....	19
2.1.8.1 Motes o nodes.....	19
2.1.8.2 Comunicacions.....	21
2.1.8.3 Exemples de WSN.....	22
2.1.9 TinyOS.....	23
2.1.9.1 Característiques de TinyOS.....	23
2.1.9.2 Llenguatge de programació nesC.....	25
2.2 Estudi de mercat.....	26
Capítol 3. Descripció funcional.....	29
3.1 Sistema total.....	29
3.1.1 Estat inicial del sensor.....	31
3.1.2 Tipus d'alarmes que envia un sensor.....	31
3.1.3 Fiabilitat del programari del sensor.....	32
3.1.4 Funcionalitats de l'estació base.....	32
3.1.5 Funcionalitats de l'aplicació PC.....	32
3.1.6 Identificador del sensor.....	33
3.1.7 Comunicacions assegurades en la transmissió d'alarmes.....	33
3.2 PC.....	33
3.3 Mota.....	35
3.4 Connexions de perifèrics en les motes (Buzzer)	36
Capítol 4. Descripció detallada.....	38
4.1 Estructures de missatges de la xarxa de sensors.....	38
4.2 Gestió de ADC per un estalvi de bateria.....	40
4.3 Comunicació segura en l'enviament d'alarmes i altres.....	42
Capítol 5. Bugs detectats en el sistema desenvolupat.....	45
Capítol 6. Viabilitat tècnica.....	46

Capítol 7. Valoració econòmica.....	47
Capítol 8. Conclusions.....	48
8.1 Conclusions.....	48
8.2 Propostes de millora.....	48
8.3 Auto-avaluació.....	49
Capítol 9. Glossari.....	50
Capítol 10. Bibliografia.....	51
Capítol 11. Annexos.....	52
11.1 Connexió del buzzer a la mota (driver).....	52
11.2 Manual d'usuari.....	53
11.3 Manual d'instal·lació.....	54
11.4 Diagrames de connexions de la mota COU24.....	57
11.5 Característiques dels sensors de la mota COU24.....	58
11.6 Característiques elèctriques del mòdul Atmel ATZB-24-A2.....	60

Índex de figures

Figura 1: Diagrama de la xarxa del sistema.....	1
Figura 2: Diagrama de blocs del mòdul ATZB-24-A2.....	1
Figura 3: Diagrama de Gantt.....	8
Figura 4: Exemple de sistema encastat.....	11
Figura 5: Arquitectura general d'un sistema encastat.....	14
Figura 6: Relació d'eficiència amb diferents característiques.....	16
Figura 7: Característiques necessàries per assegurar una qualitat.....	16
Figura 8: Ús de sistemes encastats en l'àrea aeroespacial.....	16
Figura 9: Ús de sistemes encastats en electrodomèstics.....	17
Figura 10: Sistemes encastats que integra un automòbil.....	17
Figura 11: Evolució de vendes d'unitats de dispositius mòbils.....	18
Figura 12: Tipologies comunes en WSN.....	19
Figura 13: Exemple de components que integren una mota.....	20
Figura 14: Components principals de la mota ATZB-24-A2.....	20
Figura 15: Distribució física real dels components de la mota COU24.....	20
Figura 16: Capes del protocol ZigBee (IEEE 802.15.4).....	22
Figura 17: Diagrames de radiació de l'antena de la mota COU24.....	22
Figura 18: Model de components de TinyOS i la seva interacció.....	24
Figura 19: Exemple de connexió de components d'una aplicació.....	25
Figura 20: Procés de compilació d'una aplicació en TinyOS.....	26
Figura 21: Creixement previst de dispositius sense fils.....	28
Figura 22: Modes de connexió dels diferents components del sistema.....	29
Figura 23: Diagrama de casos d'ús del sistema complet.....	30
Figura 24: Contingut del fitxer "Valors.conf".....	33
Figura 25: Diagrama UML de l'aplicació PC.....	34
Figura 26: Pantalla principal de l'aplicació PC.....	34
Figura 27: Pantalla de l'opció de monitorització de temperatures.....	35
Figura 28: Pantalla de l'opció de monitorització de bateries.....	35
Figura 29: Diagrama de blocs dels components del programari d'un sensor.....	35
Figura 30: Diagrama de flux del programari d'un sensor.....	36
Figura 31: Identificació dels pins del mòdul ZigBit ATZB-24-A2.....	37
Figura 32: Connector de propòsit general de la mota COU24.....	37
Figura 33: Diagrama de flux per la realització de lectures dels sensors.....	42
Figura 34: Diagrama de flux d'enviament de missatges d'alarmes del sensor.....	43
Figura 35: Diagrames de flux dels diferents missatges enviats per la base.....	44
Figura 36: Driver per la connexió del buzzer a la mota.....	52
Figura 37: Mostra del contingut de l'arxiu de configuració (manual).....	55
Figura 38: Serial Forwarder.....	55
Figura 39: Exemple d'execució des de terminal.....	56
Figura 40: Pantalla principal de l'aplicació PC (manual).....	56
Figura 41: Pantalla de monitorització de temperatures (manual).....	56
Figura 42: Pantalla de monitorització de bateries i cobertura (manual).....	56
Figura 43: Diagrama de connexions del microcontrolador de la mota COU24.....	57
Figura 44: Identificació de pins dels connectors de la mota.....	57

Índex de taules

Taula 1: Planificació tasca número 1 del diagrama de Gantt.....	6
Taula 2: Planificació tasca número 2 del diagrama de Gantt.....	6
Taula 3: Planificació tasca número 3 del diagrama de Gantt.....	6
Taula 4: Planificació tasca número 4 del diagrama de Gantt.....	6
Taula 5: Planificació tasca número 6 i 7 del diagrama de Gantt.....	6
Taula 6: Planificació real tasca número 1 del diagrama de Gantt.....	7
Taula 7: Planificació real tasca número 6 i 7 del diagrama de Gantt.....	7
Taula 8: Comparació dels estàndards de comunicació sense fils.....	21
Taula 9: Estructura de directoris de TinyOS.....	24
Taula 10: Relació dels pins del mòdul ZigBit i el microcontrolador.....	37
Taula 11: Estructura del missatge dels valors de temperatura.....	38
Taula 12: Estructura del missatge dels valors de bateria.....	38
Taula 13: Estructura de missatge per enviar els valors del sistema.....	39
Taula 14: Estructura de missatge per l'enviament d'alarmes.....	40
Taula 15: Composició de les signatures de la interfície Resource.....	41
Taula 16: Composició de les signatures de la interfície PacketLink.....	43
Taula 17: Costos de materials per al desenvolupament del sistema.....	47
Taula 18: Costos de desenvolupament de codi del sistema.....	47
Taula 19: Costos d'instal·lació del sistema.....	47
Taula 20: Costos de manteniment dels sistema.....	47
Taula 21: Característiques del sensor de llum de la mota COU24.....	58
Taula 22: Valors absoluts del sensor de temperatura de la mota COU24.....	58
Taula 23: Valors absoluts del sensor d'efecte Hall de la mota COU24.....	59
Taula 24: Valors màxims absoluts de la mota ATZB-24-A2.....	60
Taula 25: Valors obtinguts en proves.....	60
Taula 26: Característiques del xip de ràdio.....	60
Taula 27: Rang de valors del mòdul d'interfície.....	60
Taula 28: Característiques del microcontrolador Atmega1281.....	61

Capítol 1. Introducció

El sistema de detecció d'incendis forma una xarxa de tipologia en estrella de sensors, tal com es mostra en la figura 1, dels quals un d'ells realitza la funció d'estació base i té la funció de recollir tota la informació de la resta de sensors, per transferir-la a l'aplicació de PC, com de transferir les dades de l'aplicació de PC cap als sensors, tot això per mitjà del cable sèrie connectat de l'estació base al PC.

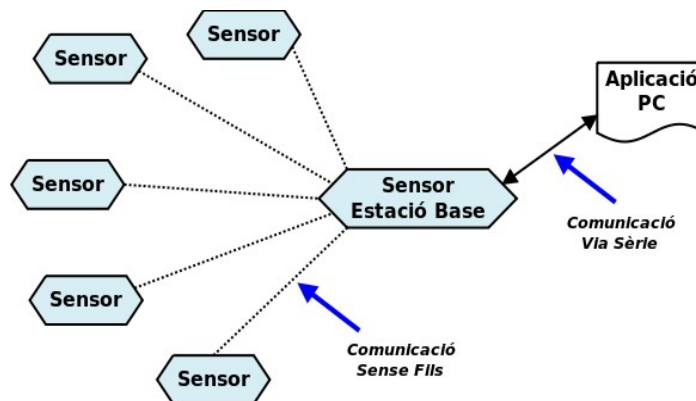


Figura 1: Diagrama de la xarxa del sistema.

El sensor utilitzat o *mota* en el projecte és “ZigBit™ 2.4GHz Wireless Modules” model ATZB-24-A2 pertanyent a la família “AVR” i la plataforma “COU24”. Les *mot*es incorporen quatre sensors: de magnetisme, de temperatura, de bateria i de llum. Disposen de tres leds: roig, verd i taronja, d'un polsador d'usuari i un altre de *reset*. El model de microcontrolador és el “ATmega1281” i el model del xip de ràdio el “AT86RF230”. En la figura 2 es mostra el diagrama de blocs del mòdul de la *mota* utilitzada.

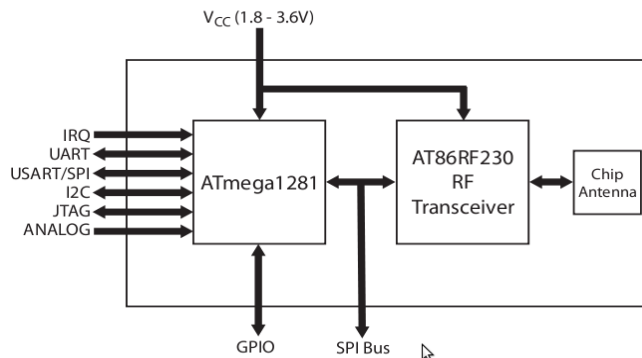


Figura 2: Diagrama de blocs del mòdul ATZB-24-A2.

El projecte consta del desenvolupament del codi dels sensors, del codi de l'estació base i de l'aplicació de PC de l'usuari. Pel desenvolupament del codi dels sensors i de l'estació base s'utilitza el llenguatge de programació “nesC” amb el suport del sistema operatiu “TinyOS”, el qual ens facilita un nombre de components estàndards alhora de desenvolupar el codi. Per l'aplicació d'usuari de PC s'utilitza el llenguatge de programació “Java”.

Per tant, l'abast del projecte és el desenvolupament d'un sistema de detecció d'incendis totalment operatiu en la realitat, però d'unes dimensions petites, ja que la tipologia de la xarxa no permet grans distàncies de

comunicació i per les característiques de les *motes* utilitzades. Aquestes, tenen com a característica que la freqüència de treball es de 2.4GHz, i això provoca que les comunicacions entre un sensor i la base són possibles solament entre una desena de metres sense barreres físiques.

1.1 Justificació

La realització d'aquest projecte, be donada per la implementació d'un sistema de detecció d'incendis, amb la característica que es pugui instal·lar en certs ambients que continguin barreres arquitectòniques insalvables. L'existència de barreres físiques, provoca que un sistema tradicional no sigui el més adient a causa de la necessitat de realització d'obres notòries per la instal·lació del cable i quan es parla d'obres hi ha certs llocs que presenten restriccions, per exemple, museus i edificis monumentals. Un altre punt important, és el cost econòmic del sistema en global, ja que, respecte a un de tradicional la inversió és molt més baixa i, per tant, un sistema a l'abast de moltes més empreses o institucions.

1.2 Descripció

El sistema a desenvolupar proporciona una xarxa de sensors de tipologia en estrella, mostrat en la figura 1 de la pàgina anterior, els quals realitzen un mostratge cada certs temps establert de la temperatura ambient del lloc on cada sensor està ubicat, i es comprova la temperatura llegida respecte a un llindar d'alarma, en cas de superar-lo el sensor envia una alarma d'incendi a la base. Totes les lectures de temperatura dels sensors, a petició del usuari a través de l'aplicació de PC, es poden enviar a l'aplicació per a que l'usuari les visualitzi.

Els sensors també envien cada cert temps el seu valor de bateria a l'aplicació de PC, a través de la base, els quals serveixen per controlar la cobertura dels sensors des de l'aplicació i per a que l'usuari les pugui visualitzar. Cada vegada que un sensor envia una lectura de bateria fa una comprovació respecte a un llindar d'alarma establert, i si es inferior envia una alarma de bateria a l'aplicació.

Els sensor disposen d'un polsador d'usuari, el qual fa disparar l'alarma d'incendi de forma manual i envia l'avís a l'aplicació. Tots els estats diferents que pot estar un sensor són indicats localment en cada sensor a través dels seus tres leds que disposa i, cadascun d'ells disposa d'un mètode per apagar i encendre el sensor. Tots els enviaments d'alarma dels sensors tenen aplicats un sistema de seguretat que fa que no es perdi cap alarma.

El node que realitza la funció d'estació base, ahora és un sensor i, per tant, té les mateixes funcionalitats que la resta de sensors, l'únic canvi es que la seva comunicació es fa via sèrie a través del cable que té connectat amb el PC on resideix l'aplicació d'usuari. En el moment que s'engega un sensor o l'aplicació d'usuari el dispositiu s'auto-configura amb el valors per defecte que té el sistema. Els sensors en cas de quedar-se penjats tenen un sistema que fa que es reiniciïn.

1.3 Objectius

A continuació es detallen uns objectius bàsics i seguidament uns objectius extrems que s'han marcat per al desenvolupament del sistema:

Objectius bàsics:

1. Detectar incendis.

- Detecció per temperatura. Quan la temperatura supera un llindar (TEMP_ALARM) el node ha d'enviar l'alarma.
- El nivell d'alarma s'ha de poder configurar des de l'aplicació de l'usuari.
- El sistema agafarà una dada del sensor cada N segons per tal de comprovar el llindar.
- N ha de ser configurable per l'usuari.

2. Notificar automàticament l'alarma de forma remota.

- Un cop es detectada l'alarma aquesta s'ha d'enviar de forma automàtica.
- La transmissió de dades serà a través d'una comunicació sense fils.

3. Notificar automàticament l'alarma de forma local.

- S'ha d'informar de l'alarma de forma local, mitjançant un senyal visual (led/s).

4. Sistema manual per activar l'alarma.

- El node ha de proveir d'un botó d'emergència per tal d'activar l'alarma de forma manual.

5. Transmissió d'alarmes sense pèrdues.

- No es pot perdre cap alarma d'incendi.
- La comunicació s'ha de fer amb ACK.

6. Reconeixement d'alarmes des de l'aplicació d'usuari.

- L'aplicació ha de mostrar per pantalla quan hi ha hagut una alarma.
- Ha de mostrar:
 - i. Tipus d'alarma (manual o automàtica).
 - ii. Quin sensor ha detectat l'alarma i quina temperatura detecta.
 - iii. Hora en que s'ha produït.
- L'usuari ha de poder confirmar la recepció d'alarma.

7. Monitoritzar la bateria dels nodes de la xarxa.

- Cada node enviarà la seva bateria de forma periòdica cada L segons.
- L ha de ser configurable des de l'aplicació.
- Quan el node detecti que té la bateria crítica, és a dir per sota de un llindar (BAT_LVL_ALARM), enviarà una alarma notificant del seu estat.
- Aquest missatge es farà servir per controlar que el node funciona correctament. Si el PC fa més de L segons que no rep un missatge del node, mostrarà una alarma per pantalla.

8. Mostrar els diferents estats del sistema en el node.

- Sensor apagat. De forma periòdica pot encendre un led, conforme el node pot funcionar correctament.

- No alarma. De forma periòdica pot encendre un led, conforme el node esta funcionant correctament.
 - Alarma detectada.
 - Alarma rebuda en estació central.
 - Alarma reconeguda.
 - Nivell bateria baix.
 - Fora de cobertura.
9. Sistema de protecció de caigudes. Els nodes no es poden quedar “penjats” (WatchDog).
- El node ha de fer servir el WatchDog per evitar quedar-se penjat.
 - Ha de recuperar la configuració de forma automàtica.
10. Sistema de debug de l'aplicació.
- El usuari ha de poder veure les temperatures dels diferents sensors.
 - El usuari pot demanar rebre les dades cada M segons. M ha de ser configurable.
11. Prova de cobertura.
- Proporcionar un sistema per comprovar si on s'instal·la el node hi ha cobertura amb el node base.
12. Sistema d'activació.
- El sensor no estarà sempre encès. Per tal d'iniciar el sensor, es farà ús del sensor d'efecte Hall. Per tal d'encendre el sensor s'haurà de passar un imant 2 vegades seguides per sobre d'aquest. Per apagar-lo s'haurà de passar 4 vegades seguides per sobre.
13. Interfície d'usuari.
- Ha de ser simple.
 - Auto-instal·lable i auto-executable :
 - i. Script d'instal·lació, ...
 - ii. Contingui totes les llibreries necessàries.
 - iii. Manual d'instal·lació.
 - Contenir un menú d'ajuda.

Objectius extres:

14. Control de lluminositat (control d'obertura de la caixa d'un sensor).
15. Integrar un buzzer per notificar de forma acústica l'alarma.
16. En la prova de cobertura informar de la potència (dB) rebuda.
17. Interfície gràfica per l'aplicació del PC.

Per a portar a terme aquest objectius, els classificarem segons la importància bàsica de funcionament. Els primers que durem a terme seran els dels sensors de temperatura i bateria, llegint els seus valors corresponents, seguidament l'enviament per ràdio d'aquests valors i finalment l'aplicació dels llindars d'alarma. Un cop realitzat l'objectiu anterior, passarem a el desenvolupament del pulsador i del sensor d'efecte Hall, seguidament del controls de leds i el sistema de protecció de caigudes com a últim. Els quatre últims objectius (objectius extres) els considerem com a secundaris, per tant ens centrarem en la resta d'objectius amb l'ordre esmentat anteriorment, per tindre un sistema mínim funcionant. A partir d'aquí

desenvoluparem l'aplicació de PC en mode text i comprovarem el funcionament del sistema bàsic desenvolupat. Un cop tot funcioni correctament, acabarem de desenvolupar la resta d'objectius.

1.4 Enfocament

Per al desenvolupament complet del projecte s'ha establert la següent metodologia:

Es defineixen dos fases per a la realització del projecte, la primera etapa desenvoluparà una versió del projecte bàsica, la qual no implementarà els quatre últims objectius de l'apartat anterior, i una segona etapa la qual implementarà tots els objectius definits.

Primera etapa:

- Llegir i comprendre les pàgines de l'assignatura de la *wiki* de la UOC.
- Preparació del entorn informàtic del equip (instal·lació de Ubuntu 10.04 i altres).
- Instal·lació del sistema operatiu "TinyOS" en l'entorn preparat anteriorment.
- Llegir i comprendre els *tutorials* de la pàgina oficial de "TinyOS".
- Primer contacte amb les motes; comprovar el funcionament de les motes i del sistema, programant l'exemple facilitat per la pàgina de l'assignatura i realitzar diverses proves.
- Realitzar la proposta del projecte.
- Realitzar la planificació del projecte.
- Desenvolupament del codi de la primera etapa del projecte.
- Inici de la memòria del projecte.
- Proves i correccions del codi desenvolupat en la primera etapa.

Segona etapa:

- Desenvolupament del codi de la segona etapa del projecte.
- Proves i correccions del codi desenvolupat en la segona etapa.
- Inici de la presentació del projecte.
- Entrega de la memòria del projecte.
- Presentació del projecte.

1.5 Planificació

Amb ajuda del programari "OpenProj" s'ha realitzat una planificació al inici de curs per a portar amb èxit el desenvolupament del projecte, i un altra al final de curs representant el temps real dedicat; d'acord amb les dates oficials de l'assignatura i, segons les tasques enumerades en l'apartat anterior sobre la metodologia seguida en el desenvolupament. Després es detallen els punts que han fet provocar la variació de la planificació inicial. A continuació es mostren les taules de planificació.

Taules de planificació previstes en el inici de curs:

Documentació	29/02/12 08:00	14/06/12 17:00
Proposta	29/02/12 08:00	7/03/12 17:00
Planificació	5/03/12 08:00	20/03/12 17:00
Memòria TFC	10/04/12 08:00	12/06/12 17:00
Presentació TFC	7/06/12 08:00	14/06/12 17:00

Taula 1: Planificació tasca número 1 del diagrama de Gantt.

Preparació entorn	29/02/12 08:00	7/03/12 17:00
Llegir pàgina assignatura	29/02/12 08:00	2/03/12 17:00
Preparació entorn informàtic	29/02/12 08:00	5/03/12 17:00
Instal·lació sistema TinyOS	5/03/12 08:00	7/03/12 17:00

Taula 2: Planificació tasca número 2 del diagrama de Gantt.

Estudi del sistema TinyOS	2/03/12 08:00	20/03/12 17:00
Llegir tutorials TinyOS	2/03/12 08:00	20/03/12 17:00
Buscar dubtes en FAQ's i en la xarxa	2/03/12 08:00	20/03/12 17:00

Taula 3: Planificació tasca número 3 del diagrama de Gantt.

Proves prèvies	7/03/12 08:00	20/03/12 17:00
Test d'aplicacions d'exemples	7/03/12 08:00	12/03/12 17:00
Estudi aplicació exemple assignatura	7/03/12 08:00	20/03/12 17:00
Resolució de problemes	9/03/12 08:00	20/03/12 17:00

Taula 4: Planificació tasca número 4 del diagrama de Gantt.

Desenvolupament de codi	20/03/12 08:00	22/05/12 17:00
Primera etapa	20/03/12 08:00	24/04/12 17:00
Estructures bàsiques	20/03/12 08:00	2/04/12 17:00
Comunicacions	2/04/12 08:00	9/04/12 17:00
Acabament	9/04/12 08:00	16/04/12 17:00
Proves	16/04/12 08:00	24/04/12 17:00
Segona etapa	24/04/12 08:00	22/05/12 17:00
Implementació part d'objectius secundaris	24/04/12 08:00	7/05/12 17:00
Creació entorn gràfic aplicació PC	7/05/12 08:00	14/05/12 17:00
Proves	14/05/12 08:00	22/05/12 17:00

Taula 5: Planificació tasca número 6 i 7 del diagrama de Gantt.

Taules de planificació reals a final de curs:

Es mostren solament les que han estat modificades i les files de les taules que han estat revisades marcades en vermell.

Documentació	29/02/12 08:00	14/06/12 17:00
Proposta	29/02/12 08:00	7/03/12 17:00
Planificació	5/03/12 08:00	20/03/12 17:00
Memòria TFC	20/04/12 08:00	12/06/12 17:00
Presentació TFC	7/06/12 08:00	14/06/12 17:00

Taula 6: Planificació real tasca número 1 del diagrama de Gantt.

Desenvolupament de codi	20/03/12 08:00	25/05/12 17:00
Primera etapa	20/03/12 08:00	24/04/12 17:00
Estructures bàsiques	20/03/12 08:00	2/04/12 17:00
Comunicacions	2/04/12 08:00	21/04/12 17:00
Acabament	9/04/12 08:00	23/04/12 17:00
Proves	21/04/12 08:00	24/04/12 17:00
Segona etapa	24/04/12 08:00	25/05/12 17:00
Implementació part d'objectius secundaris	24/04/12 08:00	7/05/12 17:00
Creació entorn gràfic aplicació PC	7/05/12 08:00	14/05/12 17:00
Proves	14/05/12 08:00	25/05/12 17:00

Taula 7: Planificació real tasca número 6 i 7 del diagrama de Gantt.

Punts que han fet variar la planificació durant el desenvolupament del projecte:

- En el desenvolupament de codi, durant la primera etapa la implementació de les comunicacions segures va requerir un nombre d'hores molt superior al previst. Això, va provocar que en el punt de documentació, el començament d'inici de la memòria es va retardar bastants dies i, un nombre inferior de dies per a la realització de proves de codi.
- En la segona etapa de desenvolupament de codi, l'ampliació de la data límit d'entrega de codi, es va aprofitar per la correcció d'errors detectats en la primera entrega i implementar adequadament la gestió d'estalvi de bateria amb un control del ADC molt més rigorós.

Diagrama de Gantt d'inici de curs:

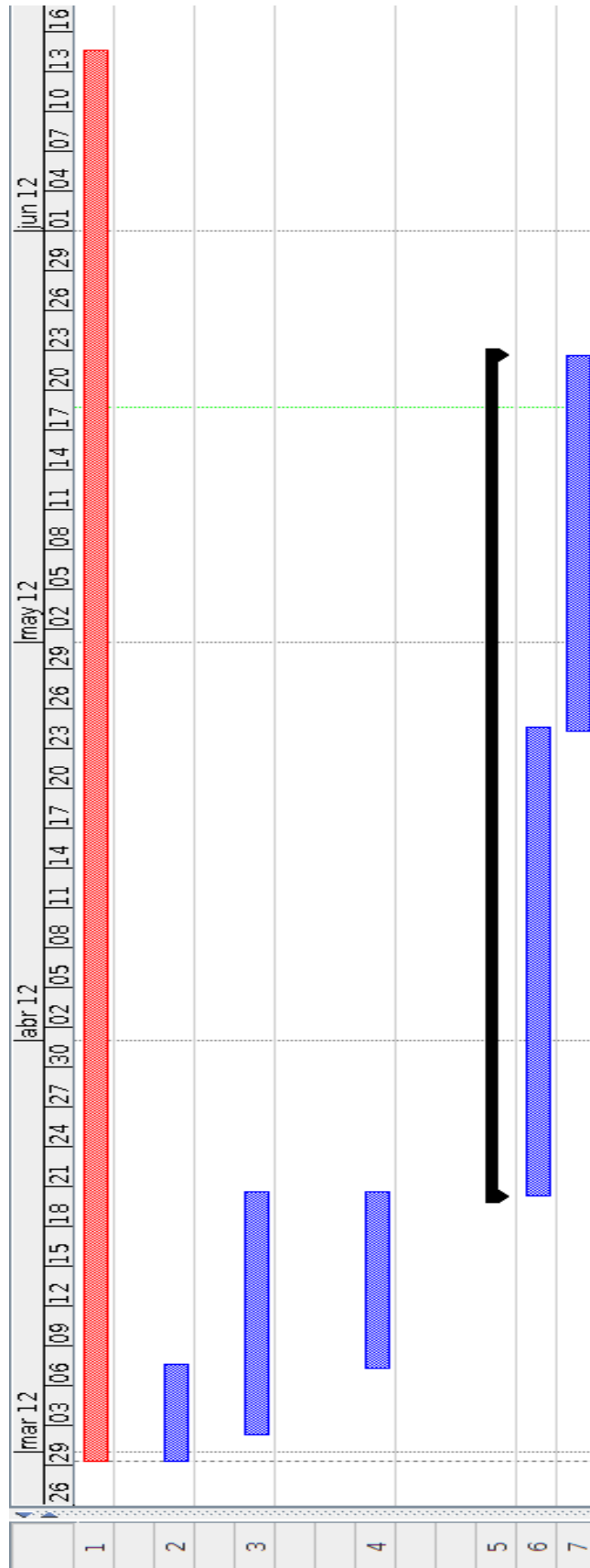


Figura 3: Diagrama de Gantt.

1.6 Recursos emprats

Llistat dels recursos emprats per al desenvolupament del projecte:

- Ordinador personal portàtil HP Pavilion dv6 amb 4Gb de RAM.
- Sistema operatiu Ubuntu 10.04 de 32 bits.
- Sistema operatiu TinyOS 2.1.1.
- Dos motes de la plataforma COU24 marca Atmel i model ATZB-24-A2.
- Pàgina web de l'assignatura de la *wiki* de la UOC.
- Pàgina web oficial del sistema operatiu TinyOS.
- Tutorials inclosos en el sistema operatiu TinyOS.
- Llenguatge de programació nesC.
- Llenguatge de programació JAVA.
- Paquet d'ofimàtica LibreOffice.
- Programari de disseny de diagrames Dia.
- Entorn de programació Eclipse Ganymede preparat per a llenguatge nesC i Java.
- Plugin NESCDDT per a l'Eclipse per al llenguatge de programació nesC.
- Programari de gestió de projectes OpenProj.

1.7 Productes obtinguts

Al final del desenvolupament del projecte s'obtenen tres tipus de producte:

Codi Sensor:

Codi funcional desenvolupat en llenguatge “nesC” per implementar en una *mota*, la qual funcionarà com a sensor remot, i que compleix una sèrie de requisits establerts en els objectius detallats en el capítol 1 d'aquesta memòria.

Codi Estació Base:

Codi funcional desenvolupat en llenguatge “nesC” per implementar en una *mota*, la qual funcionarà com a estació base i que compleix una sèrie de requisits establerts en els objectius detallats en el capítol 1 d'aquesta memòria.

Aplicació d'usuari:

Aplicació desenvolupada en llenguatge “java” per executar-la en l'ordinador personal, on la *mota* amb funcionalitat d'estació base esta connectada físicament a través del port *USB* de l'ordinador. L'aplicació es desenvolupa en dues versions, la primera en mode text, la qual l'execució es realitza a través de la consola de comandes de l'ordinador i, una segona en mode gràfic. L'aplicació és la interfície que disposa l'usuari pel control i maneig de tota la xarxa de sensors i, per tant, del sistema de detecció d'incendis.

La instal·lació del programari desenvolupat, tant en les *motes* com en un ordinador personal, dóna com a resultat el sistema de detecció d'incendis proposat i desenvolupat en aquest projecte.

1.8 Breu descripció dels altres capítols

En el següent capítol, el número dos, es realitza un breu repàs sobre l'evolució i estat dels sistemes encastrats en l'actualitat, com la tecnologia actual que es disposa en el mercat. Es defineix també els conceptes de *motes*, famílies i plataformes existents fins a l'actualitat, els protocols de comunicació de les diferents plataformes... I per últim, es realitza un petit estudi sobre el mercat i el sistema desenvolupat en aquest projecte.

En el capítol tres, es realitza la descripció del disseny emprat pel sistema total com per les diferents parts que el compon amb ajuda de diferents diagrames de blocs, incloent-hi, les decisions preses durant el disseny. En el capítol quatre, es realitza la descripció tècnica detallada dels punts descrits en el capítol anterior, s'inclouen diferents tipus de diagrames com les estructures dels missatges de comunicació desenvolupats per al sistema.

Sobre la viabilitat del sistema es parla en el capítol 6, on es detallen els punts dèbils i forts del sistema desenvolupat. En el següent capítol, el set, es realitza un estudi sobre la valoració econòmica, la qual inclou un pressupost desglossat dels diferents punts per la posta en marxa i el manteniment posteriori del sistema.

El capítol vuit descriu les diferents conclusions extretes durant la realització del projecte de final de carrera, els objectius aconseguits, una sèrie de propostes de millores i una auto-avaluació. Als següents capítols s'inclouen el glossari, la bibliografia i els annexos.

Capítol 2. Antecedents

En aquest capítol es realitza un breu repàs sobre l'evolució i estat dels sistemes encastats en l'actualitat, com la tecnologia actual que es disposa en el mercat. Es defineix també els conceptes de *motes*, famílies i plataformes existents fins a l'actualitat, els protocols de comunicació de les diferents plataformes... I per últim, es realitza un petit estudi sobre el mercat i el sistema desenvolupat en aquest projecte.

2.1 Estat de l'art

2.1.1 Definició de sistemes encastats

Del mòdul “Introducció als sistemes encastats” de l'assignatura trobem la següent definició¹:

“Un sistema encastat el podem definir com un sistema informàtic d'ús específic, que és encapsulat totalment pel dispositiu que controla. Els sistemes encastats constitueixen un sistema computacional fruit de la combinació de maquinari i programari. Aquesta combinació té com a missió dur a terme una funcionalitat o un conjunt de funcionalitats determinades. Es diuen *encastats* perquè normalment formen part d'un sistema complet o amb funcionalitats més generals.”

Per tant, un sistema encastat no està dissenyat per ser programat per un usuari final, com passaria amb un PC, si no que l'única opció que té l'usuari final és com a molt la seva configuració. El sistema encastat pot realitzar una o diverses tasques concretes pel qual ha estat dissenyat. El desenvolupament d'un sistema encastat és condicionat sempre per la seva robustesa i eficiència, i amb el condicionant que cap dels seus usuaris final no sigui conscient de la seva existència. En la següent figura es mostra un exemple de sistema encastat.



Figura 4: Exemple de sistema encastat.

2.1.2 Components d'un sistema encastat

En un sistema encastat a nivell general es pot distingir les següents parts²:

- En la part central es troba el microprocessador, microcontrolador, DSP, etc. Es a dir, la CPU o unitat que aporta capacitat de càlcul al sistema, amb la possibilitat d'incloure memòria interna o externa, una CPU amb arquitectura específica segons requisits.

¹ En l'apartat de recursos de l'assignatura de TFC es poden trobar els diferents mòduls.

² Informació extreta de la *wikipedia*: http://es.wikipedia.org/wiki/Sistemas_empotrados

- Les comunicacions adquireixen gran importància en els sistemes encastats. L'habitual és que el sistema pugui comunicar-se mitjançant interfícies estàndards de cable o sense fils. Així, un sistema encastat normalment incorporarà ports de comunicació del tipus RS-232, RS-485, SPI, I²C, CAN, USB, IP, Wi-Fi, GSM, GPRS, DSRC, etc.
- El subsistema de presentació tipus sol ser una pantalla gràfica, tàctil, LCD, alfanumèric, etc.
- S'anomenen actuadors als possibles elements electrònics que el sistema s'encarrega de controlar. Pot ser un motor elèctric, un commutador tipus relé etc. El més habitual pot ser una sortida de senyal PWM per al control de la velocitat en motors de corrent contínua.
- El mòdul de E/S analògiques i digitals se solen utilitzar per a digitalitzar senyals analògiques procedents de sensors, activar díodes LED, reconèixer l'estat obert/tancat d'un commutador o polsador, etc.
- El mòdul de rellotge és l'encarregat de generar els diferents senyals de rellotge a partir d'un únic oscil·lador principal. El tipus d'oscil·lador és important per varis aspectes: per la freqüència necessària, per l'estabilitat necessària i pel consum de corrent requerit. L'oscil·lador amb millors característiques pel que fa a estabilitat i cost són els basats en ressonador de cristall de quars, mentre que els que requereixen menor consum són els RC. Mitjançant sistemes PLL s'obtenen altres freqüències amb la mateixa estabilitat que l'oscil·lador patró.
- El mòdul d'energia s'encarrega de generar les diferents tensions i corrents necessàries per alimentar els diferents circuits del sistema encastat. Usualment es treballa amb un rang de possibles tensions d'entrada que mitjançant convertidors AC/DC o DC/DC s'obtenen les diferents tensions necessàries per alimentar els diversos components actius del circuit.
- A demés dels convertidors AC/DC i DC/DC, altres mòduls típics, filtres, circuits integrats supervisors d'alimentació, etc.
- El consum d'energia pot ser determinant en el desenvolupament d'alguns sistemes encastats que necessàriament s'alimenten amb bateries, amb que el temps d'ús del sistema encastat sol ser la duració de la carrega de les bateries.

2.1.3 Arquitectura bàsica d'un sistema encastat

Un sistema encastat posseeix una arquitectura de la qual es pot distingir certs elements bàsics, els quals es detallen a continuació³:

Microprocessador, microcomputador, microcontrolador, DSP, etc.

És l'encarregat de realitzar les operacions de càlcul principals del sistema. Executa codi per a realitzar una determinada tasca i dirigeix el funcionament dels demés elements que l'envolten, en mode de director d'una orquestra.

Memòria

En aquesta resideix el codi dels programes que el sistema pot executar així com les dades. La seva característica principal és que té de tindre un accés de lectura i escriptura el més ràpid possible per

3 Informació extreta del mateix recurs de l'apartat anterior.

a que el microprocessador no perdi temps en tasques que no siguin merament de càlcul. Al ser volàtil el sistema requereix d'un suport a on s'emmagatzemi les dades inclús sense disposar d'alimentació o energia.

Memòria cau

Memòria més ràpida que la principal en al que s'emmagatzema les dades i el codi accedit estalviar temps ja que no serà necessari anar a la memòria principal si la dada o l'instrucció es troba en la cau. El seu alt preu fa que tingui una grandària molt inferior (8-512 KB) amb respecte a la memòria principal (8-256 MB). En l'interior del xip del microprocessador es troba una petita cau (L1), però normalment es té una major en un altre xip de la placa mare (L2).

Disc dur

Permet que la informació sigui no volàtil i a demés pot aconseguir capacitats molt elevades. A diferència de la memòria que és d'estat sòlid aquest sol ser magnètic. Però, la seva excessiva grandària a vegades el fa inviable per als sistemes encastats, amb la qual cosa es requereixen solucions com unitats d'estat sòlid. Un altre problema que presenten els dispositius magnètics, es que alhora d'integrar-los en sistemes encastats, es que porten parts mòbils, el que els fa inviables per a entorns a on estaran exposats a certes condicions de vibracions. Existeix en el mercat varies solucions d'aquesta classe (DiskOnChip, CompactFlash, IDE Flash Drive, etc.) amb capacitats suficients per la majoria de sistemes encastats (des de 2 MB fins a varis GB).

BIOS-ROM

BIOS (Basic Input & Output System, sistema bàsic d'entrada i sortida) és codi que es necessari per a inicialitzar la computadora i per a posar en comunicació els distints elements de la placa mare. La ROM (Read Only Memory, memòria de solament lectura no volàtil) és un xip a on es troba el codi BIOS.

CMOS-RAM

És un xip de memòria de lectura i escriptura alimentat amb una pila a on s'emmagatzema el tipus i ubicació dels dispositius connectats a la placa mare (disc dur, ports d'entrada i sortida, etc.) A demés conté un rellotge en permanent funcionament que ofereix al sistema la data i l'hora.

ChipSet

Xip que s'encarrega de controlar les interrupcions dirigides al microprocessador, l'accés directe a memòria (DMA) i al bus ISA, a demés d'oferir temporitzadors, etc. És freqüent trobar la CMOS-RAM i el rellotge de temps real en l'interior del ChipSet.

Entrades al sistema

Poden existir ports per a ratolí, teclat, video en format digital, comunicacions sèrie o paral·lel, etc.

Sortides del sistema

Ports de vídeo per a monitor o televisió, pantalles de cristall líquid, altaveus, comunicacions sèrie o paral·lel, etc.

Ranures de expansió per a targetes de tasques específiques

Per a funcions que poden no venir incorporades en la placa mare, com poden ser més ports de comunicacions, accés a la xarxa de computadors via LAN (Local Area Network, xarxa d'àrea local) o

via xarxa telefònica: bàsica, RDSI (Xarxa digital de serveis integrats), ADSL (Asynchronous Digital Subscriber Loop, Llaç digital asíncron de l'abonat), Cable mòdem, etc. Un PC estàndard sol tindre moltes més ranures d'expansió que un sistema encastat. Les ranures d'expansió estan associades a diferents tipus de bus: VESA, ISA, PCI, NLX (ISA+PCI), etc.

En la figura següent es mostra l'arquitectura general d'un sistema encastat:

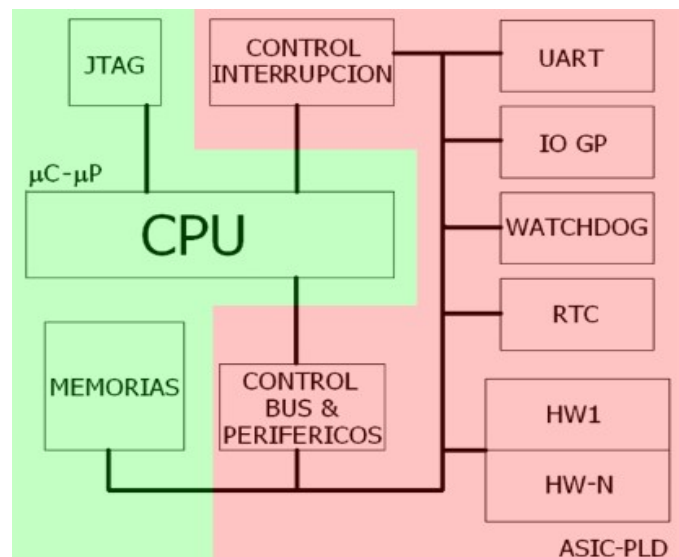


Figura 5: Arquitectura general d'un sistema encastat.

Avui dia, existeixen en el mercat fabricants que integren un microprocessador i els elements controladors dels dispositius fonamentals d'entrada i sortida en un mateix xip, pensant en les necessitats dels sistemes encastats (baix cost, petita grandària, entrades i sortides específiques, etc.). La seva capacitat de procés sol ser inferior als processadors de propòsit general, però, compleixen amb el seu comès ja que els sistemes a on s'ubiquen no requereixen molta potència. Els principals fabricants són STMicroelectronics (família de xips STPC), AMD (família Geode), Motorola (família ColdFire) i Intel.

Pel que fa als sistemes operatius necessaris per a que un sistema basat en microprocessador pugui funcionar i executar programes, solen ser específics per als sistemes encastats. Així, ens trobem amb sistemes operatius de baix requisits de memòria, possibilitat d'execució d'aplicacions de temps real, modulars (inclusió solament dels elements necessaris del sistema operatiu per al sistema encastat concret), etc. El més coneguts en l'actualitat són Windows CE, QNX i VxWorks de WindRiver. En aquest projecte s'utilitza un nou sistema operatiu anomenat TinyOS, per a xarxes de sensors o motors, desenvolupat per la universitat de Berkeley (Califòrnia, EUA) i que avui dia té un recolzament notori.

2.1.4 Característiques dels sistemes encastats

Les principals característiques que ofereixen els sistemes encastats són:

Concurrència:

- Els components del sistema controlat o monitoritzat funcionen simultàniament.
- El sistema de control té que atendre'l i generar les accions de control o visualització de forma simultània.
- Un computador executa les seves accions de forma seqüencial, donant rapidesa i per tant, es pot fer que el computador executi les seves accions de forma aparentment simultània.
- Computadors multiprocessador o sistemes amb varies computadores.

Fiabilitat i seguretat:

- Una fallada en un sistema de control pot fer que el sistema controlat es comporti de forma perillosa o amb resultats negatius econòmics.
- És important assegurar que si el sistema de control falla el faci de forma que el sistema controlat quedi en un estat segur, per tant, hi ha que tindre en compte les possibles fallades o excepcions en el disseny.

Eficiència:

- Una gran part dels sistemes de control deuen respondre amb gran rapidesa als canvis en el sistema controlat.

Interacció amb dispositius físics:

- Els sistemes encastats interaccionen amb el seu entorn mitjançant diversos tipus de dispositius que normalment no són convencionals (teclats, impressores,...): convertidors A/D i D/A, PWM, entrades i sortides digitals paral·leles i sèrie, etc. (interfícies amb sensors, actuadors, perifèrics especials,...).
- Els components del programari que controlen el funcionament d'aquests dispositius (controladors, "drivers") són, en general, dependents del sistema concret.

Robustesa:

- Embarcats en sistemes amb moviment o que puguin ser transportats, subjectats a vibracions i inclús impactes (cotxes, robots, instrumentació portàtil,...).
- No sempre treballen en condicions òptimes de temperatura, humitat, neteja.
- Factor de protecció IP: IP65
 - - Primer dígit: protecció davant l'entrada de sòlids (pols).
 - - Segon dígit: protecció davant l'entrada de líquids.

Baix consum:

- Molts d'aquests sistemes estan alimentats amb bateries o piles, per tant, menor consum dona com a resultat major autonomia.
- En molts casos necessitats de baix voltatge (3V).

Baix pes:

- Característica d'agrair en sistemes portàtils.
- No depèn únicament del computador embarcat i de la seva perifèria, si no també de l'alimentació (bateries) o dels sensors i actuadors.

Baix preu:

- Aplicable a electrònica de consum i altres dispositius amb mercats molt competitius (per exemple la telefonia mòbil).

Petites dimensions:

- Les dimensions d'un sistema encastat no depenen solament de si mateix si no també del espai disponible en el sistema que controla i/o monitoritza.
- Característica a tindre molt en compte pels problemes que comporta.

En les següents figures es mostren la relació de diferents característiques respecte a l'eficiència (figura 6) i l'exigència (figura 7) que es demana a un sistema encastat.

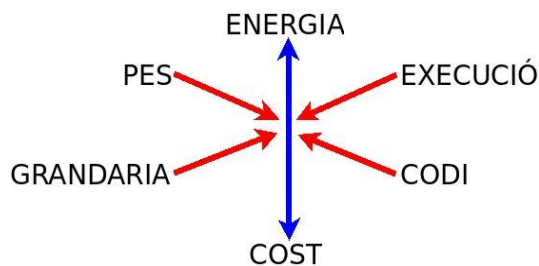


Figura 6: Relació d'eficiència amb diferents característiques.



Figura 7: Característiques necessàries per assegurar una qualitat.

2.1.6 Exemples d'us de sistemes encastats

L'ús de sistemes encastats en l'actualitat està en plena expansió, ja que cada dia són molts més aparells electrònics els quals els incorporen. Tant a nivell de productes de consum com són els electrodomèstics, els cotxes, els telèfons mòbils, etc. Com a nivell de les diferents àrees: indústria civil, aeroespacial (figura 8), militar, geologia, agricultura, etc.



Figura 8: Ús de sistemes encastats en l'àrea aeroespacial.

En l'àmbit dels productes de consum, els sistemes encastats són els principals actors encarregats de realitzar les funcions específiques de cada electrodomèstic concret, són exemples els següents mostrats en la figura 9, els quals amb molta probabilitat resideixen en el nostre entorn domèstic.

Tasques Específiques



Figura 9: Ús de sistemes encastrats en electrodomèstics.

En el cas concret dels automòbils, és l'exemple més clar d'ús de sistemes encastrats, existeix un gran nombre de sistemes encastrats, els quals cadascun d'ells porta a terme una funcionalitat concreta, tal com s'especifica en la figura 10.

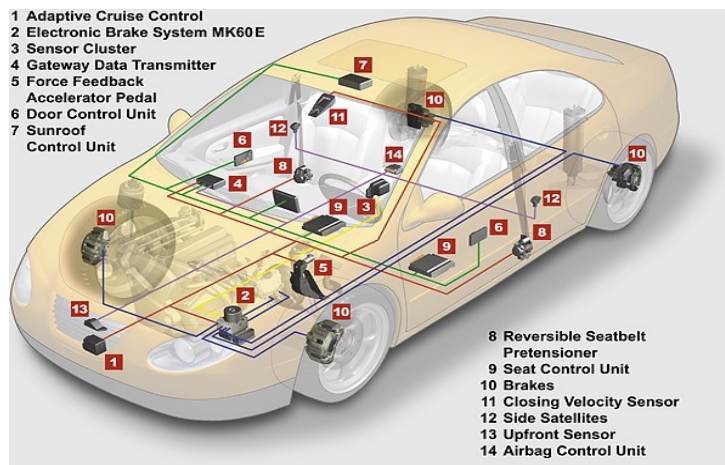


Figura 10: Sistemes encastrats que integra un automòbil.

Els dispositius mòbils són un dels grans culpables de l'augment exponencial de l'ús de sistemes encastrats, ja que els últims anys les seves vendes i la seva tecnologia ha experimentat un creixement imparabile, el qual encara a llarg termini no mostra símptomes d'un alentiment. La figura 11 mostra una gràfica d'aquest comportament.

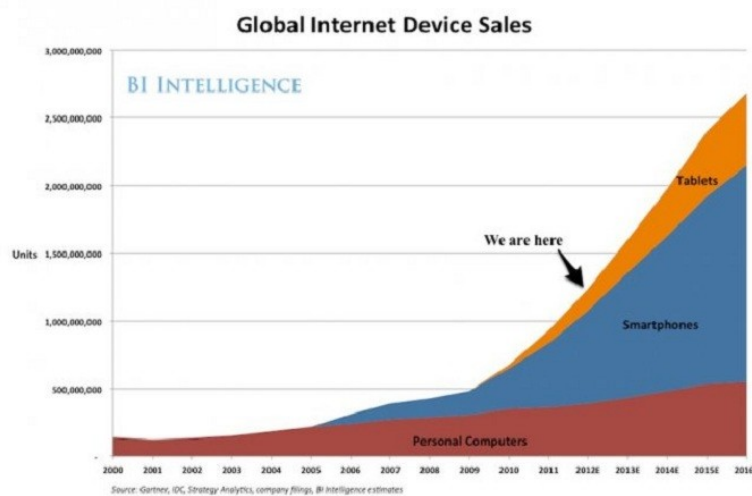


Figura 11: Evolució de vendes d'unitats de dispositius mòbils.

Per tant, com es pot comprovar l'evolució dels sistemes encastats està experimentant un augment significatiu i, està previst que durant els pròxims anys continuarà experimentant un creixement exponencial.

2.1.7 Concepte de família

En l'àrea de sistemes encastats existeix el concepte de *família*⁴, el qual vol dir que una família engloba tots els components que estan dissenyats sota un mateix patró i que comparteixen un conjunt de propietats. Així, en el context d'un sistema encastat, podem trobar components que s'engloben dins l'àmbit d'una família i que, per tant, comparteixen propietats.

En l'àmbit dels sistemes encastats hi ha un conjunt de famílies que han anat evolucionant a mesura que se n'estenia l'ús. Les més representatives són:

- L'arquitectura ARM de l'empresa ARM.
- L'arquitectura AVR d'Atmel.
- L'arquitectura PIC de Microchip.
- L'arquitectura MSP430 de Texas Instruments.
- L'arquitectura 8051 d'Intel.
- L'arquitectura Z80 de Zilog.
- L'arquitectura 65816 de Western Design Center.
- L'arquitectura SuperH d'Hitachi.
- L'arquitectura ETRAX CRIS de Axis Communications.
- L'arquitectura Power Architecture (pròpiament dita PowerPC).
- L'arquitectura eSi-RISC d'EnSilica's.

Una família és constituïda per múltiples microcontroladors que segueixen les especificacions bàsiques de la família, com són el format de les instruccions o l'arquitectura del processador. Una família no deixa de ser

⁴ Definició i dades extrets del mòdul "Introducció als sistemes encastats" de l'àrea de recursos de l'assignatura.

una especificació i un disseny d'un nucli i un conjunt de funcionalitats que forma el que s'anomena microcontroller unit (MCU) o core.

2.1.8 Xarxes de sensors sense fils

Les xarxes de sensors sense fils, també anomenades WSN (Wireless Sensors Network), es troben englobades en el món dels sistemes encastats. Es pot definir com una gran quantitat de petits dispositius autònoms i distribuïts físicament, anomenats nodes de sensors, instal·lats al voltant d'un fenomen per a ser monitoritzat, amb la possibilitat d'emmagatzemar i comunicar dades en una xarxa sense fils. Les seves característiques són:

- Facilitat de desplegament.
- No s'utilitza infraestructura de xarxa:
 - Encaminament entre nodes sense visió directa amb comunicacions multi-salt.
- Topologia dinàmica.
 - Nodes autoconfigurables, tolerància a fallades.
- Utilització de "broadcast".
- Ultra baix consum.
 - Funcionament amb piles o bateries, llarga autonomia.
- Molt baix cost.
- Petita grandària.
- Operació sense manteniment durant varis mesos o anys.

Les xarxes de sensors sense fils es constitueixen mitjançant els següents elements i les quals poden tindre una tipologia comuna com les mostrades en la figura 12:

- Sensors: Prenen del medi la informació i la transformen en senyals elèctrics.
- Nodes (Motes): Prenen les dades dels sensors i envien la informació a l'estació base.
- Porta d'enllaç: Elements per la connexió de la xarxa de sensors a una xarxa de dades.
- Estació base: Recol·lector de dades basat en un ordinador comú o sistema encastat.
- Xarxa sense fils: Típicament basada en l'estàndard 802.15.4 *ZigBee*.

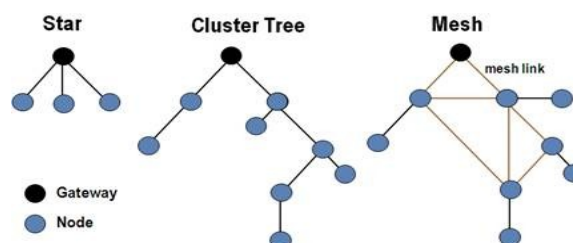


Figura 12: Tipologies comunes en WSN.

2.1.8.1 Motes o nodes

Les motes són petits dispositius, les quals integren una sèrie d'elements com un microcontrolador, un xip de comunicació via ràdio, un port de comunicació sèrie, convertidors A/D, varis sensors, etc. En la figura 13 es

mostra un exemple dels diferents components que pot tindre una mota. Algunes de les seves característiques són: la memòria té una capacitat d'uns pocs KB, la velocitat del processador oscil·la normalment en uns quants Mhz, les ones de ràdio tenen un abast d'unes desenes de metres, incorporen un porta-piles normalment per dues bateries tipus AA o AAA, etc.

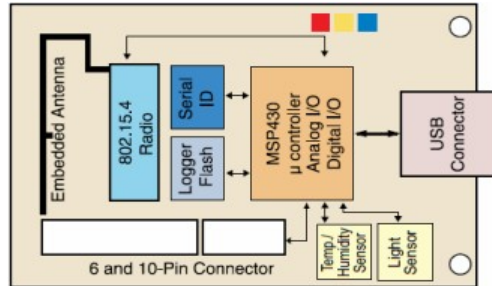


Figura 13: Exemple de components que integren una mota.

Les motes s'agrupen en plataformes, que són construccions de sensors concrets, on cada plataforma utilitza un microcontrolador i un xip de ràdio específic amb diversos sensors. Existeixen diverses plataformes: Telosb, Mica2, MicaZ, Tmote Sky, Iris, Imote2, Epic Mote, Xbee, etc. En el desenvolupament d'aquest projecte s'ha utilitzat dues motes de la plataforma COU24, model ZigBit™ 2.4 GHz Wireless Modules ATZB-24-A2, la qual té el diagrama de blocs de la figura 2.

Els tipus de sensors i els components principals que disposa l'usuari per interactuar amb la mota COU24, són els mostrats en la figura 14.

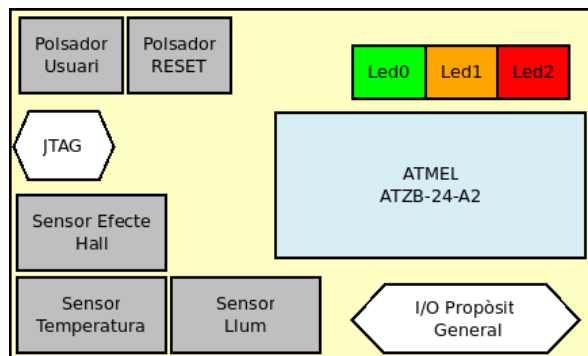


Figura 14: Components principals de la mota COU24.

El seu aspecte real amb la identificació física dels diferents components es mostren en la figura 15.

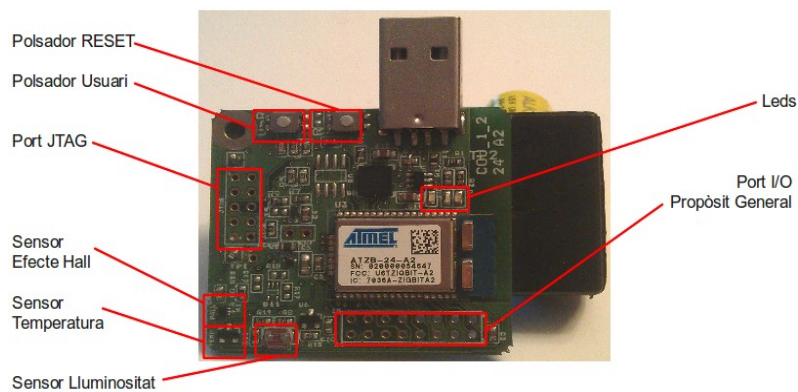


Figura 15: Distribució física real dels components de la mota COU24.

2.1.8.2 Comunicacions

Les comunicacions en les xarxes de sensors sense fils és una part molt important, així, les tecnologies sense fils estàndards i propietàries per a sensors sense fils més conegudes són per a les xarxes LAN el IEEE 802.11b (“WiFi”), per a les xarxes PAN, el protocol IEEE 802.15.1 (Bluetooth IEEE, 2002) i IEEE 802.15.4 (ZigBee IEEE, 2003). Utilitzen les bandes ISM (Instrumentation, Scientific and Medical radio bands), 902–928 Mhz (EEUU), 868 – 870 MHz (Europa), 433.05–434.79 MHz (EEUU i Europa) i 314 – 316 Mhz (Japó) i la banda de GHz de 2.400 – 2.4835 GHz (universalment acceptada).

En la taula 8 es mostra una comparació de les tecnologies sense fils: WiFi, Bluetooth i Zigbee .

	<i>WiFi (IEEE 802.11g)</i>	<i>Bluetooth (IEEE 802.15.1)</i>	<i>ZigBee (IEEE 802.15.4)</i>
<i>Ràdio</i>	DSSS, (direct sequence spread spectrum)	FHSS, (frequency hopping spread spectrum)	DSSS, (direct sequence spread spectrum)
<i>Velocitat</i>	54Mbps	1Mbps	250kbps
<i>Núm. de nodes por màster</i>	32	7	64.000
<i>Latència</i>	Up to 3s	Up to 10s	30ms
<i>Tipus de dades</i>	Vídeo, àudio, gràfics, pel·lícules, fitxers	Àudio, gràfics, pel·lícules, fitxers	Petits paquets de dades
<i>Abast (m)</i>	100	10 (v1.1)	70 100
<i>Expansió</i>	Roaming	no	si
<i>Duració bateria</i>	12 i 48 hores	1 setmana	100 – 1000 dies
<i>Cost \$</i>	9	9	9
<i>Complexitat</i>	Complex	Molt complex	Senzill
<i>Aplicació principal</i>	WLAN	WPAN	Control i monitorització
<i>Memòria necessària</i>	1MB+	250KB+	4KB-32KB
<i>Paràmetres més importants</i>	Velocitat i flexibilitat	Cost i perfils d'aplicació	Fiabilitat, baix consum i baix cos

Taula 8: Comparació dels estàndards de comunicació sense fils.

Les WSN actuals estan basades en l'estàndard IEEE 802.15.4. ZigBee amb el protocol subjacent 802.15.4, que és de tipus més general que WISA, però amb un rendiment de comunicacions menor. Inclou multi-salt, el que implica que un missatge pot utilitzar varis salts en les ones de ràdio per arribar al seu destí. Els nodes no tenen assignats intervals específics de temps, si no que han de competir per accedir al canal. Això, permet l'accés de més usuaris al medi sense fils, però introdueix incertesa en el sistema, ja que la demora i el consum d'energia augmenten quan un node està esperant el seu torn. A demés, els nodes intermedis desconeixen el moment en que poden ser sol·licitats per encaminar paquets per altres. Per tant, és aconsellable disposar de nodes intermedis. ZigBee resulta ideal per aplicacions de monitorització d'actius. Quan el nombre de nodes a connectar és molt alt, la solució són xarxes de més d'un nivell amb distintes tecnologies (xarxes híbrides). En la figura 16 es mostra les capes del protocol ZigBee (IEEE 802.15.4).

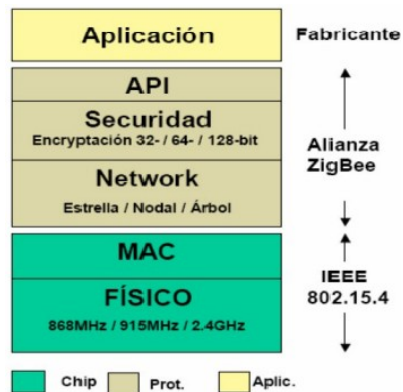


Figura 16: Capes del protocol ZigBee (IEEE 802.15.4).

La mota utilitzada COU24 per al desenvolupament del projecte treballa amb el protocol ZigBee, el qual és una aliança sense ànim de lucre, de 25 empreses, la major part d'aquestes fabricants de semiconductors, amb l'objectiu d'afavorir el desenvolupament i implementació d'una tecnologia sense fils de baix cost. La figura 17 mostra els diagrames horitzontal i vertical de radiació de l'antena incorporada en la mota utilitzada.

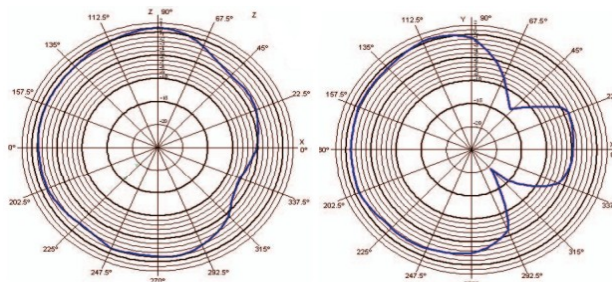


Figura 17: Diagrames de radiació de l'antena de la mota COU24.

2.1.8.3 Exemples de WSN

L'ús de les xarxes de sensors sense fils tenen una gran varietat d'aplicacions i, amb el pas del temps s'en van incrementant de forma exponencial, recordem que s'engloben en l'àrea dels sistemes encastats. A continuació es defineix una llista de possibles aplicacions:

- Domòtica i inmòtica.
 - Obtenció de mesures de consum d'aigua, electricitat, etc. Per a la seva optimització.
 - Alarmes i seguretat.
 - Creació d'entorns que s'adapten dinàmicament a les activitats dels usuaris.
- Monitorització industrial.
 - Nodes WSN en edificis, infraestructura, equipament, hivernacles, etc.
 - Per motius de seguretat, prevenció de riscos laborals, o eficiència en l'ús de recursos.
 - Permet obtindre informació de l'ús o comportament dels actius per optimitzar el seu cost i cicle de vida.
- Salut.
 - Health@Home: monitorització de pacients en la llar mitjançant elements vestibles no intrusius.

- Supervisió de gent gran, persones amb necessitats especials, etc.
- Protecció d'infraestructures.
 - Distribució elèctrica.
 - Canalitzacions d'aigua i canonades per a detecció de pèrdues.
- Recollida de dades de quadrilles d'operaris/manteniment.
- Monitorització d'habitats/ecologia.
- Monitorització de fonts de contaminació, agents químics o biològics.
- Monitorització de tràfic.
- Altres: localització/posicionament, multimèdia, etc.

2.1.9 TinyOS

TinyOS⁵ és un sistema operatiu de codi obert basat en components per a xarxes de sensors sense fils. Està escrit en el llenguatge de programació *nesC* com un conjunt de tasques i processos que col·laboren entre si. Està dissenyat per incorporar novetats ràpidament i per a funcionar sota les importants restriccions de memòria que es donen en les xarxes de sensors. TinyOS està desenvolupat per un consorci alliberat per la Universitat de Califòrnia en Berkeley (EUA) en cooperació amb Intel Research.

2.1.9.1 Característiques de TinyOS

Algunes de les seves característiques més importants són:

- Suporta diferents plataformes de sensors.
- Es pot aplicar a altres sistemes encastats.
- Conté nombroses aplicacions pre-construïdes.
- Disposa d'eines que faciliten el desenvolupament.
- El llenguatge de TinyOS es *nesC*, un metallenguatge que deriva de *C*.
- És un projecte de codi obert.
- Disseny basat en respondre a a les característiques i necessitats de les xarxes de sensors.
 - Reducció grandària de memòria.
 - Baix consum d'energia.
 - Operacions de concurrència intensiva.
 - Diversitat en dissenys i usos.
 - Operacions robustes.
- El *kernel* té una estructura de dos nivells de planificació.
 - Esdeveniments:
 - Pensats per a realitzar un procés petit.
 - Per exemple: interrupcions del *timer*, interrupcions de final de conversió A/D, etc.
 - Poden interrompre les tasques que s'estan executant.

5 Definició extreta de la pàgina web <http://es.wikipedia.org/wiki/TinyOS>.

- Tasques:
 - Estan pensades per realitzar una quantitat major de processament.
 - No són crítiques en el temps.
 - Per exemple: calcular la mitjana en una taula.
 - Les tasques s'executen en la seva totalitat.
 - En tasques de llarga duració, la sol·licitud d'iniciar una tasca, i l'acabament d'ella són funcions separades (característica pròpia de la programació orientada a components).

Les aplicacions de TinyOS són construïdes per components, un component proveeix i utilitza interfícies, aquestes interfícies són l'únic punt d'accés al component. Un component està compost d'un espai de memòria i un conjunt de tasques, els quals són: gestor de comandes, gestor d'esdeveniments, un *frame*⁶ de grandària fixa i estàticament assignat, en el qual, es representa l'estat intern del component i un bloc amb tasques simples. En la figura 18 es mostra el model gràfic d'un component, en el qual es mostra la seva estructura.

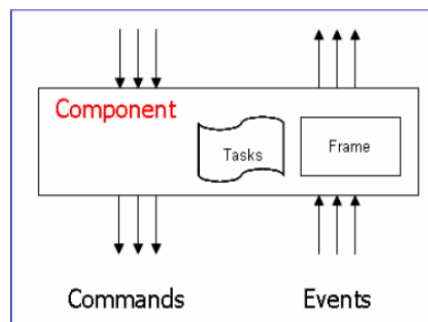


Figura 18: Model de components de TinyOS i la seva interacció.

A continuació, es mostra una taula (taula 9) on s'especifica l'estructura de directoris que componen TinyOS.

Directorio	Descripció
/tos/interfaces	Conté totes les interfícies que són proporcionades pels components primitius i per les aplicacions d'exemple.
/tos/lib	Conté llibreries per resoldre determinats problemes.
/tos/system	Conté tots els components primitius que proporciona TinyOS.
/tos/types	Conté els tipus que s'utilitzen en els primitius de TinyOS.
/tos/platforms	Conté els fitxers necessaris per l'execució en les diverses plataformes.
/tos/sensorboards	Conté els fitxers que són específics dels sensors de cada plataforma.

Taula 9: Estructura de directoris de TinyOS.

6 Referència a un "bloc" que proporciona el context en el qual s'executa el programari i s'emmagatzema les variables.

2.1.9.2 Llenguatge de programació *nesC*

El llenguatge *nesC* (**n**etwork **e**MBEDDED **s**ystems **C**) és un metallenguatge de programació basat en *C*, orientat a sistemes encastats que incorporen el maneig de la xarxa. A més, suporta un model de programació que integra el maneig de comunicacions, les concurrències que provoquen les tasques i esdeveniments, i la capacitat de reaccionar en front a successos que puguin succeir en els ambients a on s'exerceix (per exemple, mostreig).

També, realitza optimitzacions en la compilació del programa, detectant possibles carreres de dades que poden succeir producte de modificacions concurrents a un mateix estat, dintre del procés d'execució de l'aplicació. A més, simplifica el desenvolupament d'aplicacions, redueix la grandària del codi i, elimina moltes fonts potencials d'errors.

Les característiques que ofereix *nesC* són les següents:

- Separació entre la construcció i la composició. Hi ha dos tipus de components en *nesC*: mòduls i configuracions. Els mòduls proveeixen el codi de l'aplicació, implementant una o més interfícies. Aquestes interfícies són els únics punts d'accés al component. Les configuracions són utilitzades per unir els components entre si, connectant les interfícies que alguns components proveeixen amb les interfícies que altres utilitzen.
- Interfícies bidireccionals: las interfícies són els accessos als components, que contenen comandes i esdeveniments, les quals són les que implementen les funcions. El proveïdor d'una interfície implementa les comandes, mentre el que les utilitza implementa els esdeveniments.
- Unió estàtica de components, via les seves interfícies. Això, augmenta l'eficiència en temps d'execució, incrementa la robustesa del disseny, i permet una millor anàlisi del programari.
- El llenguatge presenta eines d'optimització en la generació de codi. Un exemple d'això és el detector de "carreres de dades", en temps de compilació.

A mode d'exemple, la figura 19, mostra l'estructura de components d'una aplicació i la connexió entre els diferents components emprats.

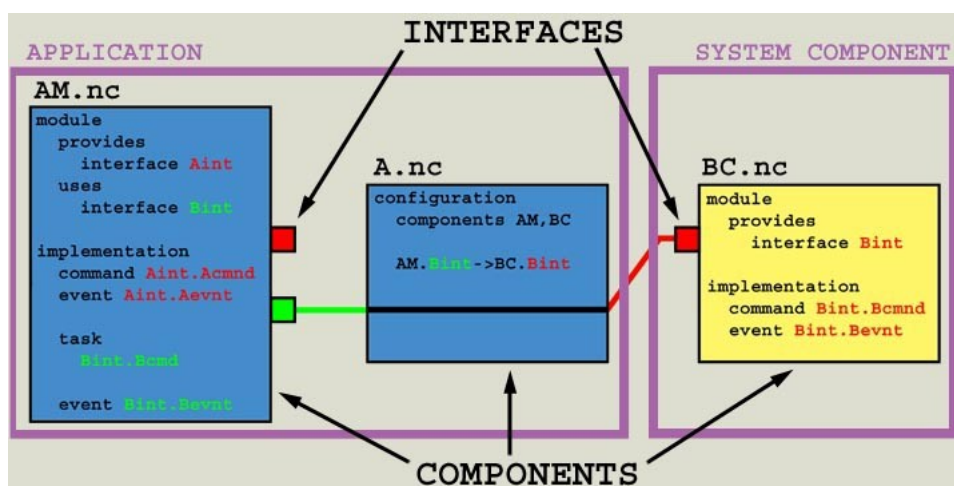


Figura 19: Exemple de connexió de components d'una aplicació.

Per acabar aquest apartat, un cop es vist per sobre el sistema operatiu TinyOS i el seu llenguatge de programació *nesC*, es mostra un diagrama del procés de compilació d'una aplicació (figura 20):

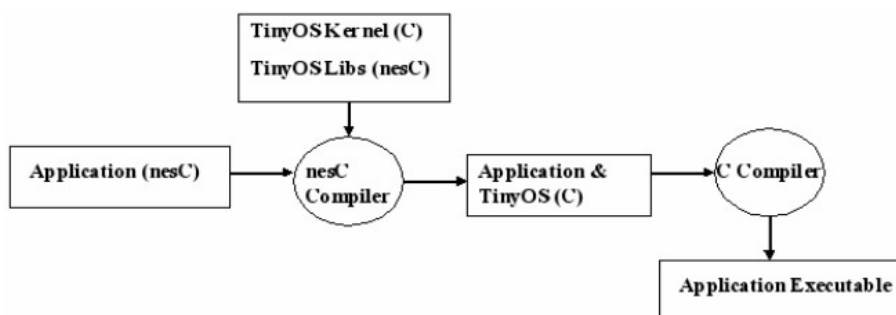


Figura 20: Procés de compilació d'una aplicació en TinyOS.

2.2 Estudi de mercat

En l'actualitat, en el mercat existeixen un nombre indeterminat de sensors detectors d'incendi, els quals posseeixen diferents característiques de detecció, com els sensors de fum. Entre altres projectes, el govern d'Aragó posseeix un programa d'investigació, en el qual es troba un projecte de detecció d'incendis que s'aplica pel control d'incendis forestal⁷ i, a més del control de temperatura afegeix altres paràmetres com el d'humitat. Per tant, l'aplicació desenvolupada es troba en un mercat en el qual ja existeixen una varietat d'altres aplicacions similars.

Des de 1998 la WSN, quan el MIT la va catalogar com una de les 21 tecnologies més influents en el segle XXI, va començar el seu desenvolupament en l'àmbit de la investigació. Això va provocar el desenvolupament de protocols per a cada una de les capes del model OSI, donat que en els dispositius d'aquest tipus de xarxa el consum energètic és un punt fonamental i els protocols TCP/IP no eren vàlids per aquest requisit. Aquest desenvolupaments encara avui dia es porten a terme i són moltes les universitats arreu del món que tenen projectes destinats a la investigació de les xarxes de sensors sense fils.

El món de les xarxes de sensors sense fils actualment està experimentant un creixement imparable. Actualment les WSN es troben en una etapa de desenvolupament i millora continua, així les funcions encara són limitades. Segons alguns experts, per donar un impuls més gran a aquestes xarxes sense fils, els nodes no poden estar lligats a la utilització de bateries i fonts d'alimentació.

Per això, l'existència d'aquests problemes es tenen que resoldre per a que les xarxes de sensors sense fils siguin elements quotidians al voltant nostre i aconseguir l'eficàcia, auto-organitzativa robustesa i diversitats adequades. Les motes són components de recursos molt limitats: tenen una velocitat de processador limitada, una capacitat d'emmagatzematge limitada, així com un ample de banda de comunicació limitat. A més, la vida operativa es determina per la seva capacitat de conservar l'energia (no malgastar la bateria). Totes aquestes limitacions requereixen nous dissenys de maquinari, noves arquitectures de xarxa, nous

⁷ El projecte es pot consultar en línia: <http://www.aragoninvestiga.org/Sistema-de-sensores-inalambricos-para-prevenir-incendios/>

algorismes, noves arquitectures de programari, noves heurístiques, mantenint al mateix temps que la seva implantació sigui de baix cost. Tota aquesta problemàtica s'està abordant i intentant solucionar en diversos centres tecnològics, entre els que podem destacar:

- Centre d'investigació de Intel (Intel Research Laboratory).
- Centre d'investigació de Palo Alto.
- Universitat de Berkeley, Califòrnia (EUA).
- Universitat de Los Angeles, Califòrnia (EUA) . Centre per a malles de sensors encastats (CENS).
- Universitat de Ohio (EUA).
- Universitat de Virginia (EUA).
- Swiss Federal Institute of Technology – EPFL (Suïssa).

Sobre els fabricants d'aquests tipus de sensors s'han trobat varis i, la gran majoria es troba als Estats Units (EUA), en Europa existeix el fabricant Shockfish resident en Suïssa (<http://www.shockfish.com>) i les representacions dels fabricants més importants del món, com són:

- AVID Wireless (<http://www.avidwireless.com>).
- AVID Wireless (<http://www.avidwireless.com>).
- DUST Networks (<http://www.dust-inc.com/flash-index.shtml>).
- Jennic (<http://www.jennic.com>).
- Millennial Net (<http://www.millennial.net>).
- Sensicast (<http://www.sensicast.com>).
- Sensoria (<http://www.sensoria.com>).
- CrossBow (<http://www.xbow.com>).
- Intel (<http://www.intel.com>).

La tecnologia WSN té unes perspectives molt prometedores en el mercat mundial i és un dels motius pel qual es considera un àrea molt atractiva pel seu estudi i desenvolupament. Algunes de les aplicacions sense fils que s'estan començant a realitzar abasten pràcticament tots els camps: biomedicina (monitorització de nivells de la sang en animals), agricultura de precisió (climatització en temperatura i humitat d'hivernacles), medi ambient (monitorització de nivells de contaminació), vulcanologia (monitorització de zones sísmiques), indústria (monitorització de desgast de la maquinaria) i, entre altres, la seguretat (aplicacions militars).

En l'actualitat, empreses com Emerson i Honeywell han posat en marxa dos grans productes WSN, amb l'antiga versió de la norma de les comunicacions sense fils HART7.1, aquesta última norma adoptada per ISA100. Però, la gran majoria presenten limitacions en els sensors de temperatura, de pressió, de flux i amb una potencia de transmissió una mica inferior.

Enquestes realitzades a empreses de l'indústria, mostra que el 40% dels enquestats planeja utilitzar les solucions sense fils per a controlar la qualitat de les màquines de producció. Segons una investigació de

l'empresa ONWorld de 2011, el mercat mundial de les xarxes de sensors sense fils (WSN) en l'àrea de sistemes i serveis augmentarà a prop de 46 mil milions de dòlars americans respecte als 500 milions de dòlars americans actuals. Les xarxes de sensors sense fils acaben de començar el seu desenvolupament en el món de les empreses, i per això aquestes tenen que aprofitar les oportunitats augmentant les seves inversions i promovent el conjunt del sector. Per tant, l'ús de dispositius sense fils, tal com es mostra en la figura 21, els pròxims anys experimentarà un creixement exponencial.

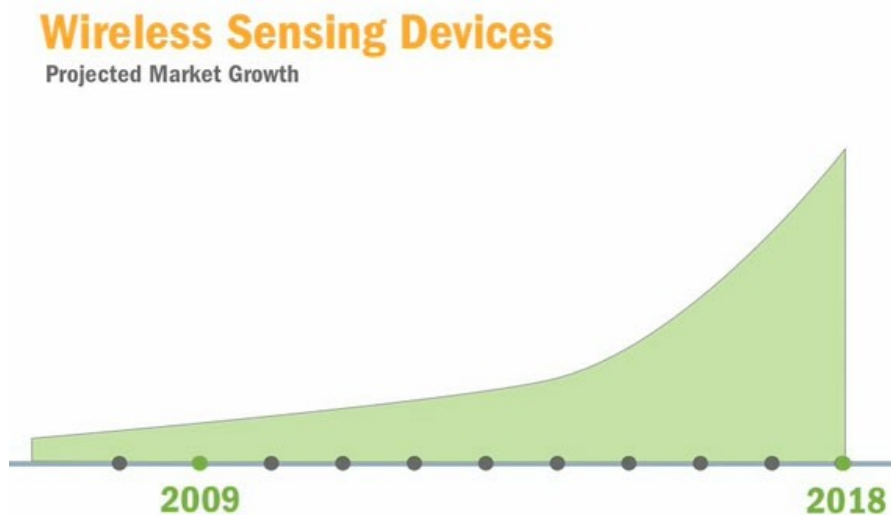


Figura 21: Creixement previst de dispositius sense fils.

Per concloure aquest apartat, l'aplicació desenvolupada en aquest projecte va dirigida a organitzacions petites, les quals tinguin una necessitat d'un sistema de protecció d'incendis per a cobrir petits espais. Aquestes organitzacions valoraran la simplicitat d'instal·lació del sistema com el manteniment i, sobretot el cost total.

Capítol 3. Descripció funcional

En els següents apartats es descriu les funcionalitats i característiques del sistema implementat, com les decisions de disseny preses i, una sèrie de diagrames de disseny dels diferents components que formen part del sistema.

3.1 Sistema total

El sistema desenvolupat està compost per dues motes (plataforma COU24) i un ordinador personal. Una de les motes s'estableix com a sensor detector remot i l'altra com a estació base. L'estació base a part de realitzar la mateixa funció de detecció que el sensor remot, té la funció de comunicar en els dos sentits possibles, el dos nodes de la xarxa sense fils amb l'aplicació PC que controla el sistema. En la següent figura es mostra els diferents modes de connexió dels components del sistema.

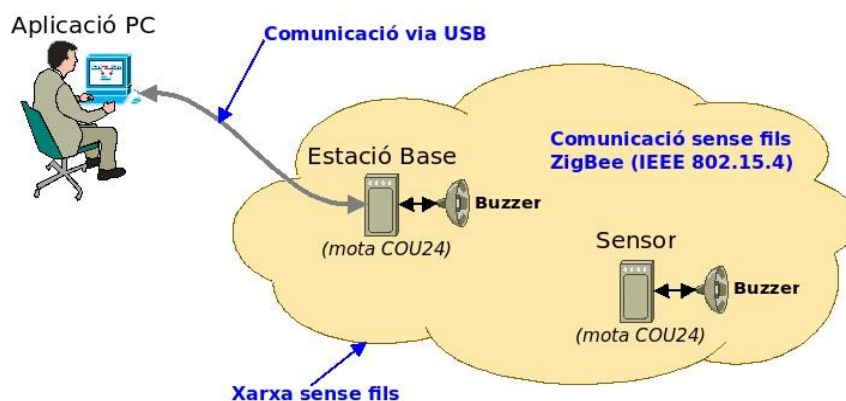


Figura 22: Modes de connexió dels diferents components del sistema.

Tal i com s'aprecia en la figura anterior, el tipus de xarxa sense fils del sistema és de tipologia en estrella. L'aplicació PC implementa una sèrie d'opcions per interactuar amb la xarxa de sensors, les quals detallarem a continuació:

- Enviament dels diferents valors de configuració del sistema (llindars i temps de mostratge).
- Monitorització dels valors de bateries de tots els nodes.
- Monitorització dels valors de la intensitat de senyal rebuda per l'estació base pel node remot.
- Comprovació de la cobertura del node remot.
- Monitorització de temperatures de tots els nodes.
- Confirmació de qualsevol alarma rebuda de forma transparent per l'usuari.
- Confirmació d'alarmes d'incendi per part de l'usuari.
- Reiniciar un sensor que presenti qualsevol alarma o que es vulgui encendre de forma remota.
- Configuració automàtica dels sensors amb els valors del sistema en cas de posta en marxa o de inicialització del programari del sensor per bloqueig.
- Configuració automàtica de tota la xarxa de sensors amb els valors del sistema quan es posa en marxa l'estació base o l'aplicació PC.
- Emmagatzematge de les diferents alarmes que pot rebre el sistema.

- Donar de baixa un sensor que es vulgui treure del sistema per no tindre'l en compte.
- Ajuda d'usuari de les diferents opcions de l'aplicació PC.

Per part dels dos nodes que fan de sensors (estació base i sensor remot), ofereixen les següents característiques:

- Activació i desactivació del sensor a través del sensor de magnetisme.
- Indicació dels diferents estats del sensor a través dels leds.
- Activació manual de l'alarma d'incendi mitjançant el polsador d'usuari del sensor.
- Lectura de la temperatura ambient, en intervals de temps configurables per l'usuari, per la comprovació del llindar d'alarma d'incendi establert.
- Enviament de les lectures de temperatures en intervals de temps definits per l'usuari.
- Lectura de la bateria del node, en intervals de temps configurables per l'usuari, per la comprovació del llindar d'alarma de bateria establert i l'enviament a l'estació base.
- Lectura de la lluminositat del node, en intervals de temps prefixats pel sistema, per la comprovació del llindar d'alarma de caixa oberta establert (també fixat pel sistema).
- Enviament de les alarmes amb espera de confirmació de recepció per part de l'aplicació PC, en cas de no rebre'l, es realitza un altre enviament de l'alarma cada cert temps.
- Comprovació de la cobertura respecte de l'estació base amb la indicació corresponent a través dels leds (l'estació base aquesta característica no la incorpora per motius obvis).

Seguidament es mostra el diagrama de casos d'ús del sistema:

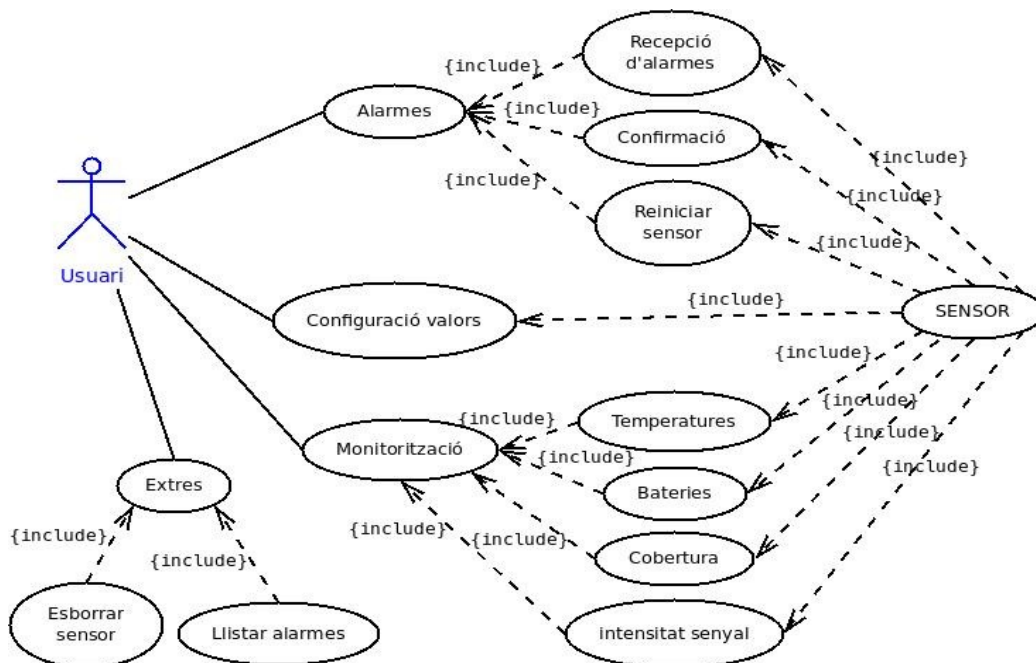


Figura 23: Diagrama de casos d'ús del sistema complet.

En els següents apartats, es detalla el funcionament del sistema respecte a una sèrie de punts basats en les qüestions més importants que es poden plantejar sobre el sistema desenvolupat.

3.1.1 Estat inicial del sensor

Cada sensor quan se li dóna tensió mostrarà el led de color verd intermitent, indicant que esta apagat, l'única cosa que el sensor fa és enviar un missatge de configuració a la base indicant-li que acaba de iniciar-se i que vol rebre els valors per defecte que hi ha al sistema. Aquesta petició de configuració la repetirà cada segon, fins a rebre resposta de la base i no enviar cap més. Per encendre el sensor es té que fer dues passades amb un imant pel sensor d'efecte Hall, el led verd s'apaga i el roig fa intermitent. Per apagar-ho farem quatre passades amb l'imant. En l'apartat "Mota" de més endavant, s'indiquen tots els estats que pot marcar un sensor i quins són.

Un cop que s'encén el sensor aquest comença a fer lectures de temperatura i comprovar-les amb el llindar, a fer lectures del sensor de llum per controlar el desmuntatge del sensor i també, es comença a fer lectures de bateria per comprovar-les amb el llindar i enviar-les per ràdio a la base. Cada cop que envia una lectura de bateria la base li respon amb un tren de missatges de bateria (20) dels quals el sensor conta els que ha rebut i comprova la intensitat de senyal rebuda, amb aquests paràmetres decideix si indica amb cobertura (rebre tres missatges amb senyal superior a -57dbm) o no.

3.1.2 Tipus d'alarmes que envia un sensor

Les alarmes d'incendi poden ser de dos tipus, manuals o automàtiques, les primeres són generades amb el polsador per part de l'usuari, i les automàtiques per les lectures de temperatura que superin un llindar establert. Quan salta qualsevol de les dos s'envia l'alarma a la base i els leds ho indicaran, juntament s'activa una sirena connectada al sensor (buzzer). Les alarmes d'incendi s'envien a l'aplicació fins que es confirma la seva recepció, a partir d'aquest moment el sensor es queda a la espera de la confirmació manual de l'usuari. L'usuari a través de l'aplicació de PC rep l'alarma i la pot confirmar quan vulgui (un cop l'usuari la confirma la sirena s'atura). Tots aquest passos es visualitzen en el sensor a través dels leds. Un cop confirmada, l'usuari pot reiniciar el sensor per a que torni a estar funcionant com si l'acabés d'encendre.

Les alarmes de bateria són com les d'incendi, aquestes s'envien a la base fins rebre confirmació d'aquesta, però per part de l'usuari no ha de reconèixer res, simplement veu l'alarma per pantalla. El sensor indicarà aquest estat a través dels seus leds. L'usuari, per tant, té que canviar les bateries al sensor, o canviar el llindar d'alarma i apagar/encendre el sensor de nou.

L'alarma de comprovació de caixa ens serveix per controlar desmuntatges de sensors no autoritzats, aquesta alarma es visualitza en el sensor amb els tres leds encesos. Al encendre un sensor, cada 60 segons es comprova la lluminositat de la caixa del sensor, en cas de superar un llindar establert s'activa l'alarma (el llindar esta configurat per a que la llum dels leds no el superi). L'enviament de l'alarma a l'estació base es repeteix fins a rebre confirmació de la recepció per part de la base. Un cop que s'ha disparat l'alarma s'ha de reiniciar el sensor per a que l'alarma torni a funcionar. Així, durant la instal·lació dels sensors que tindran les caixes obertes saltaran les alarmes, per tant, un cop estiguin ubicats en les seves caixes s'hauran de reiniciar.

Comentar que la visualització de les alarmes en el sensor tenen prioritat respecte a la comprovació de cobertura. També dir que a través de l'aplicació de PC l'usuari amb l'opció de reiniciar el sensor, pot encendre un sensor apagat sense tenir que anar físicament al punt on es troba, això es pot aplicar al cas d'alarma de bateria, quan es canvia el llindar d'activació per no canviar les bateries i, al cas d'alarma de caixa oberta, totes dues alarmes es poden reiniciar amb aquesta opció del PC.

3.1.3 Fiabilitat del programari del sensor

Tant al sensor com a l'estació base s'ha implementat un watchdog, en el sensor de 1 segon i en la base de 2 segons ja que té més càrrega de treball. Aquestos depenen del component "Estat" que és l'encarregat de visualitzar l'estat del sensor a través dels leds, dintre del disparador del timer que controla el canvi d'estat dels leds es va reiniciant el watchdog. En cas de bloqueig del programa del sensor fa que es reiniciï i per tant estarà en mode apagat (led verd intermitent). Això ens serveix per controlar els sensors que tinguin algun problema, si no un sensor en mal estat, es podria estar reiniciant varies vegades i no ens assabentaríem si es poses en mode "on" automàticament, ja que perdre un cop la cobertura podria ser per altres motius (interferències).

3.1.4 Funcionalitats de l'estació base

L'estació base és el mateix que un sensor però es comunica amb l'aplicació PC a través de cable (sèrie), i té una funcionalitat més que és la de rebre i enviar els missatges via ràdio de la resta de sensors cap a l'aplicació PC. Per tant, a partir de l'identificador de la mota a la qual va dirigit el missatge la base sap cap a on té que dirigir el missatge. L'estació base té la peculiaritat que quan s'engega demana el missatge de auto-configuració i quan el rep l'utilitza per ella però a demés ho envia a la resta de la xarxa de sensors.

3.1.5 Funcionalitats de l'aplicació PC

L'aplicació PC es bastant intuïtiva, ja que conté una sèrie de botons marcats amb les diferents opcions, entre aquestes es troba una ajuda, on s'explica la funció una mica més detallada. Comentar que al engegar l'aplicació per primer cop, aquesta carregarà els valors que es troben en l'arxiu de configuració "Valors.conf" en el sistema i, realitza l'enviament d'un missatge de configuració a tota la xarxa amb aquests valors. Així, alhora de donar alimentació a l'estació base o engegar l'aplicació PC, s'envia la configuració amb els valors que tingui l'aplicació en aquell moment a la resta de sensors de la xarxa, en cas contrari, al donar alimentació a qualsevol sensor solament se l'envia la configuració a aquell sensor en concret.

La cobertura també es comprova en l'aplicació PC, tal com diu els requeriments, així l'aplicació quan rep per primera vegada un missatge de bateria d'un sensor nou, l'emmagatzema i comença a controlar la seva cobertura. En cas que per algun motiu un sensor es vulgui donar de baixa, o sigui treure'l de la xarxa, si l'apaguem sense fer res més hi haurà el problema que l'aplicació PC continuarà comprovant la seva cobertura i quan es monitoritzi les bateries es mostrarà sempre el missatge de fora de cobertura d'aquest sensor, que sabem que l'hem tret de la xarxa. Per evitar això, es disposa de l'opció "Esborrar sensor del

sistema” la qual li podem indicar a l'aplicació PC que volem esborrar del sistema un sensor concret indicant-li el seu identificador.

3.1.6 Identificador del sensor

Sobre els identificadors que es tenen que utilitzar, és el número “1” per l'estació base i els successius fins al número “63” com a màxim. El primer sensor amb el “2”, el tercer amb el “3”... En aquest cas concret, per realitzar les proves el sensor s'ha identificat com número “2”. Comentar que en el desenvolupament de l'aplicació PC s'ha optimitzat per l'estació base i tres sensors (identificadors “2”, “3” i “4”), per tant, segons la necessitat de cada client pel nombre de sensors que pugui arribar a utilitzar, es té que ampliar el nombre de sensors previstos per l'aplicació PC.

3.1.7 Comunicacions assegurades en la transmissió d'alarmes

S'ha implementat dos nivells d'acks alhora d'enviar totes les alarmes dels sensors. Un primer nivell es fa mitjançant el mateix missatge d'alarma que l'aplicació retorna utilitzant una variable d'aquest missatge per indicar el tipus de reconeixement, el segon nivell es assegurar l'enviament des de l'estació base al sensor. Aquest segon nivell d'ack, en l'estació base també està implementat per l'enviament dels missatges de configuració i els de bateria utilitzats per la cobertura. Així, si l'aplicació PC es penja el sensor no rebrà cap confirmació i enviarà l'alarma cada cert temps fins que l'aplicació PC la processi correctament, en cas de penjar-se la base passaria el mateix.

3.2 PC

Per l'aplicació PC s'ha desenvolupat un programari en mode gràfic amb totes les funcionalitats descrites en l'apartat anterior. S'ha utilitzat el llenguatge de programació JAVA, per la creació de les diferents classes que componen el programari, la figura 25 de la següent pàgina mostra el diagrama UML del programari.

L'aplicació PC càrrega els valors per defecte del sistema des d'un fitxer de configuració (Valors.conf) en el moment del seu inici. El fitxer es troba en la mateixa carpeta a on es troben les classes compilades de l'aplicació (/bin), juntament amb un fitxer *script* per a executar l'aplicació des d'un terminal del sistema. A continuació, la figura 24 mostra el contingut del fitxer de configuració. Aquest es pot modificar amb els valors que l'usuari vulgui, respectant el format del nombre decimals, ja que si no l'aplicació PC mostrarà un missatge d'error en la càrrega del fitxer i l'aplicació no s'iniciarà.



```
#Connexió (ip:port):
localhost:9002
#N (segons):
12
#L (segons):
5
#Llindar Temperatura entre 1 i 100°C(format XX.XX):
50.00
#Llindar Bateria entre 1 i 3.5V (format XX.XX):
2.5
#FI
```

Figura 24: Contingut del fitxer "Valors.conf".

Diagrama UML de l'aplicació:

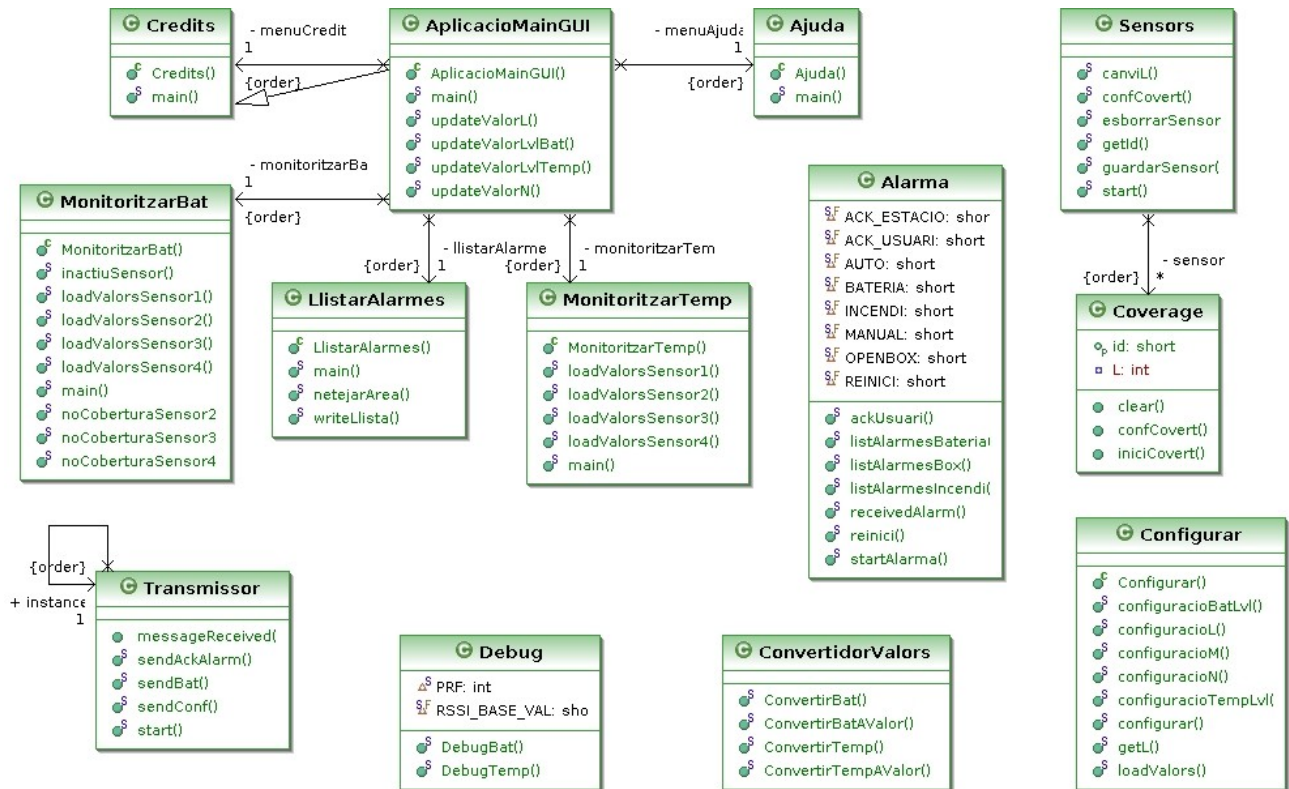


Figura 25: Diagrama UML de l'aplicació PC.

En el contingut del fitxer es pot observar com la primera línia fa referència als paràmetres de connexió que l'aplicació utilitzarà per connectar-se amb el *SerialForwarder*⁸ i les següents són els valors del sistema que es carregaran en la xarxa de sensors. Un cop iniciada l'aplicació PC, es mostra la pantalla principal tal com es pot veure en la figura 26.

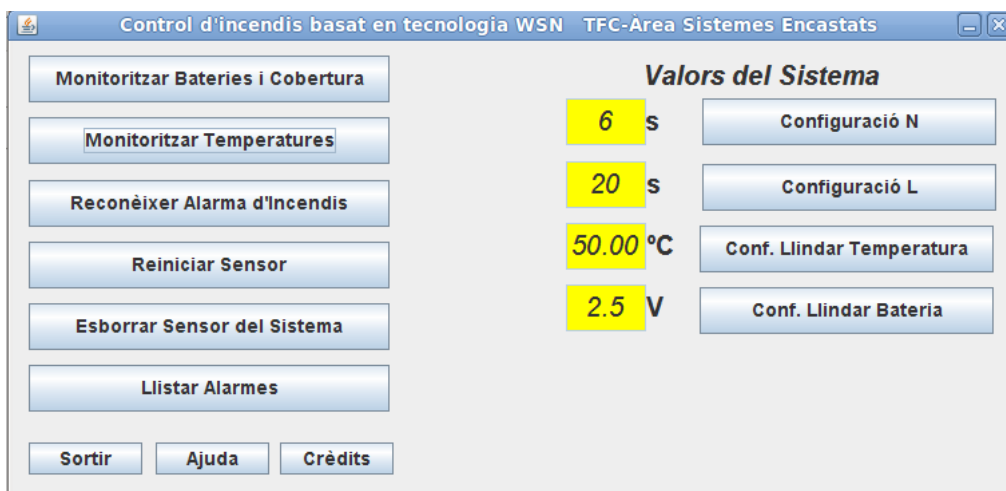


Figura 26: Pantalla principal de l'aplicació PC.

8 Més informació sobre *SerialForwarder* en l'annex.

Les següents figures mostren les pantalles de les opcions de monitorització de temperatures i bateries.

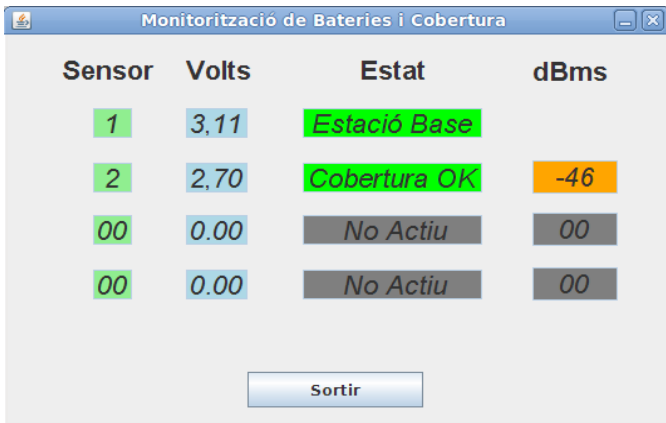


Figura 28: Pantalla de l'opció de monitorització de bateries.

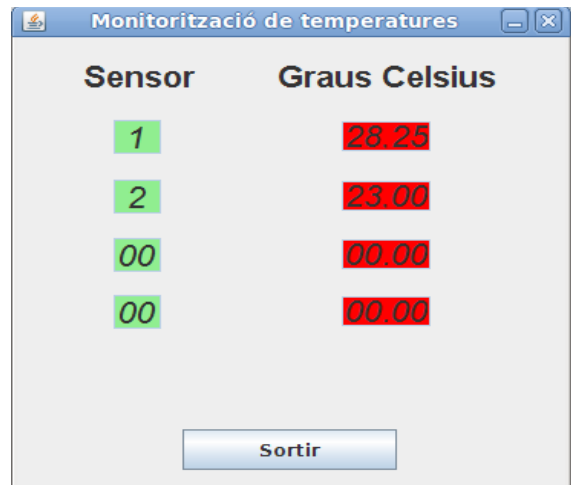


Figura 27: Pantalla de l'opció de monitorització de temperatures.

3.3 Mota

El programari de les motes s'ha desenvolupat amb el llenguatge de programació nesC, el qual està suportat pel sistema operatiu TinyOS, pensat per a xarxes de sensors. Les característiques més destacades que s'han portat a terme han estat la programació amb baixa cohesió de les diferents funcionalitats, la optimització de les bateries i la seguretat en les comunicacions. Tot seguit es mostra el diagrama, figura 29, dels diferents components que integren el programari d'un sensor.

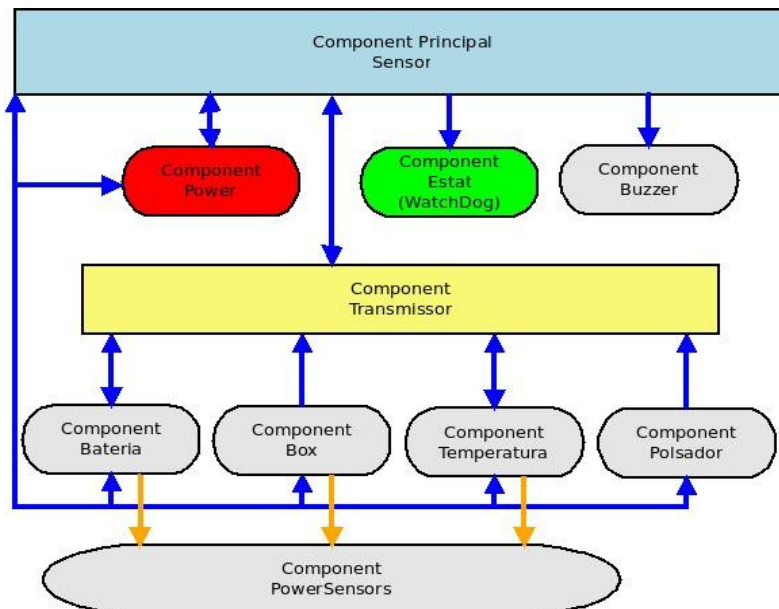


Figura 29: Diagrama de blocs dels components del programari d'un sensor.

La interfície "Estat" és l'encarregada de mostrar els diferents estats del sensor a través dels leds de la mota, els quals podem definir els següents:

- Apagat: led **verd** intermitent.
- Encès: led **roig** intermitent.

- Alarma incendi: led **roig** encès.
- Alarma incendi rebuda per estació: led **roig** encès i **taronja** intermitent.
- Alarma incendi reconeguda per usuari: led **roig** encès i **taronja** encès.
- Alarma bateria: led **verd** encès.
- Fora de cobertura: els tres leds intermitents.
- Alarma caixa oberta: la freqüència dels leds s'incrementa al doble (en els cassos dels estats que tenen algun led intermitent).

Per comprendre millor com es relacionen els diferents components del programari, es mostra un diagrama de flux d'un sensor:

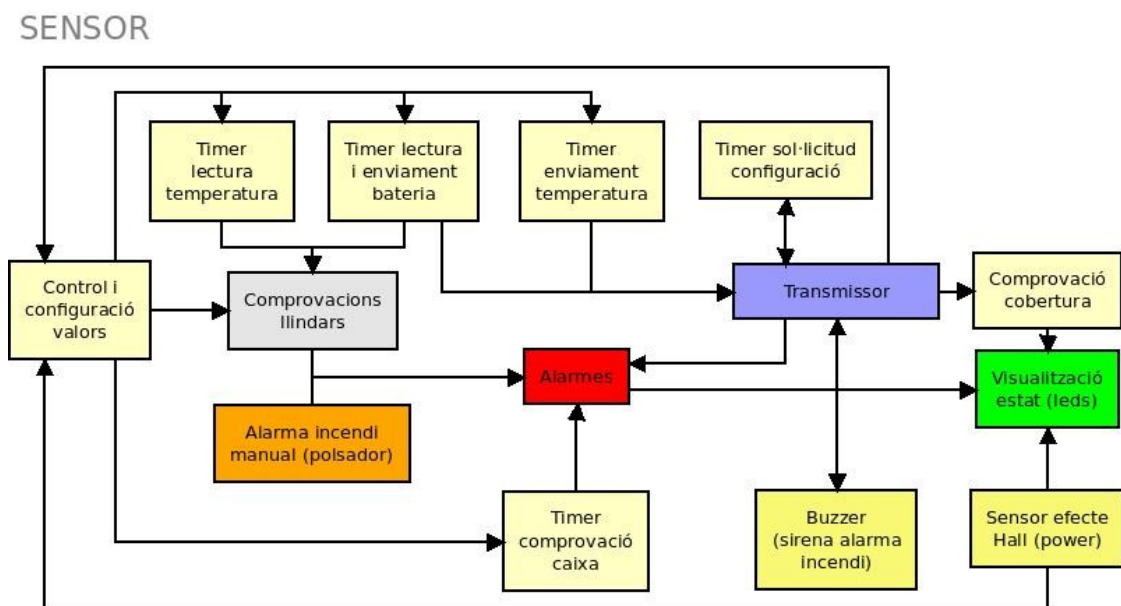


Figura 30: Diagrama de flux del programari d'un sensor.

3.4 Connexions de perifèrics en les motes (Buzzer)

El sistema ha esta implementat amb sirenes d'alarma per a cada sensor, per tant en la mota s'utilitza un port de propòsit general disponible en el connector per tal efecte. Primer de tot tindrem que identificar quin és el port que tenim disponible en el connector de propòsit general.

Segons els següents esquemes (figures 31 i 32) es pot observar que l'únic port que tenim disponible per a usos generals és el GPIO5_CON, que correspon al *pin* número 15. Ara es té que relacionar aquest port del mòdul ZigBit de la mota, que segons la figura 34 correspon al *pin* 21 del mòdul, amb els ports que disposa el microcontrolador (Atmega 1281) intern del mòdul. La taula 15 mostra aquesta relació, de la qual es pot veure que el port GPIO5 (*pin* 21) correspon al port PG2 (port G, *pin* 2).

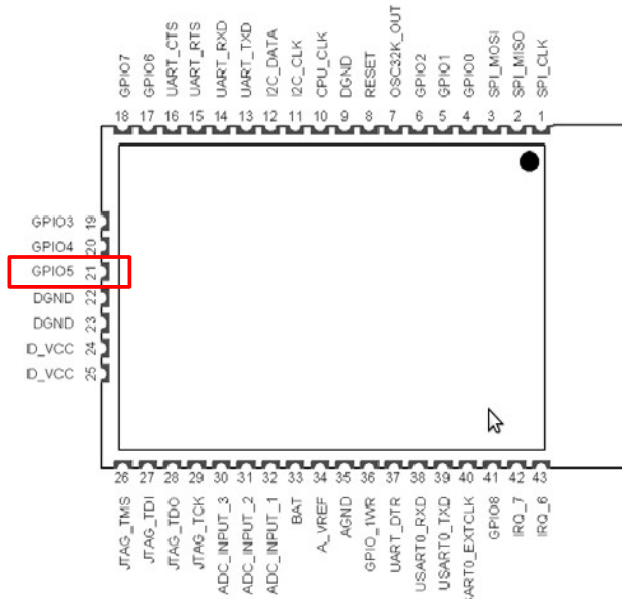


Figura 31: Identificació dels pins del mòdul ZigBit ATZB-24-A2.

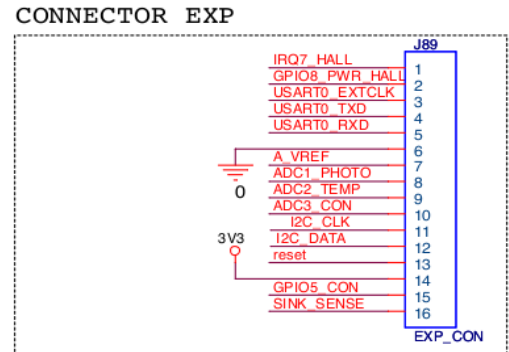


Figura 32: Connector de propòsit general de la mota COU24.

Relació dels pins del mòdul ZigBit amb els ports del microcontrolador Atmega 1281:

ATmega 1281 Port	ZigBit Pin	ATmega 1281 Port	ZigBit Pin
PB0	-	PF0	33
PB1	1	PF1	32
PB2	3	PF2	31
PB3	2	PF3	30
PB4	-	PF4	29
PB5	4	PF5	26
PB6	5	PF6	28
PB7	6	PF7	27
PD0	11	PG0	19
PD1	12	PG1	20
PD2	13	PG2	21
PD3	14	PG3	7
PD4	15	PG4	-
PD5	16	PG5	36
PD6	17	PG6	-
PD7	18	PG7	-
PE0	38		
PE1	39		
PE2	40		
PE3	41		
PE4	37		
PE5	-		
PE6	43		
PE7	42		

Taula 10: Relació dels pins del mòdul ZigBit i el microcontrolador.

Comentar que el *buzzer* no es connectarà directament a la mota, si no que farà falta un *driver*⁹. Aquest estarà controlat per la mota pel seu port corresponent i, una font d'alimentació externa subministrarà la corrent necessària al *buzzer*.

9 A l'annex es detalla un possible esquema d'un *driver* i com és connectaria a la mota.

Capítol 4. Descripció detallada

En aquest apartat aprofundirem en els aspectes més importants de la implementació de codi del sistema. Es realitza una descripció de les estructures de missatges que s'intercanvien en la xarxa de sensors, una explicació en detall sobre com s'ha implementat les comunicacions segures entre motes i, de com s'ha realitzat una gestió eficient de bateria en la utilització del convertidor ADC de les motes.

4.1 Estructures de missatges de la xarxa de sensors

Per la implementació de la comunicacions de la xarxa de sensors, s'han creat quatre estructures de missatges pròpies del sistema, per a l'intercanvi dels valors llegits pels sensors de les motes i per al control dels diferents estats que poden succeir en un sensor. Les estructures emprades amb els seus camps descrits són els mostrats en les taules següents:

Estructura utilitzada per enviar les dades de la temperatura de la mota:

	<i>Tipus</i>	<i>Camp</i>	<i>Descripció</i>
nx_struct TempMsg	nx_uint8_t	moteld	Identificador de la mota.
	nx_uint16_t	countsTemp	Valor llegit del sensor de temperatura.
	nx_uint16_t	vrefmilivolts	Valor de referència que utilitza el sensor per la lectura del ADC.

Taula 11: Estructura del missatge dels valors de temperatura.

El missatge solament té com a objectiu l'enviament del valor llegit de temperatura.

Estructura utilitzada per enviar les dades de la bateria de la mota:

	<i>Tipus</i>	<i>Camp</i>	<i>Descripció</i>
nx_struct BatMsg	nx_uint8_t	moteld	Identificador de la mota.
	nx_uint16_t	countsBat	Valor llegit de la bateria de la mota.
	nx_uint16_t	vrefmilivolts	Valor de referència que utilitza el sensor per la lectura del ADC.
	nx_uint8_t	rss	Valor que guarda la intensitat del senyal rebut d'aquest missatge.

Taula 12: Estructura del missatge dels valors de bateria.

Aquesta estructura ens serveix pel control de cobertura dels sensors per part de l'aplicació PC, ja que se sap que cada cert temps determinat es té que rebre un missatge, juntament amb la informació de la intensitat de senyal, rssi¹⁰, rebuda per l'estació base en la recepció del missatge.

¹⁰ El terme rssi significa "Received Signal Strength Indicator".

Estructura utilitzada per enviar les dades de configuració a la mota¹¹:

	<i>Tipus</i>	<i>Camp</i>	<i>Descripció</i>
nx_struct ConfMsg	nx_uint8_t	moteld	Identificador de la mota.
	nx_uint32_t	L	Variable que configura el temps de lectura de la bateria.
	nx_uint32_t	N	Variable que configura el temps de lectura de la temperatura.
	nx_uint32_t	M	Variable que configura el temps d'enviament de les lectures de temperatura (per defecte 0, desactivat).
	nx_uint16_t	BAT_LVL_ALARM	Llindar d'alarma de la bateria.
	nx_uint16_t	TEMP_ALARM	Llindar d'alarma de la temperatura.

Taula 13: Estructura de missatge per enviar els valors del sistema.

En l'estructura del missatge solament és obligatori el camp de l'identificador de la mota, ja que un cop rebut el missatge per un sensor aquest solament càrrega els valors que siguin diferents a zero, excepte el del camp "M", que en cas de valdre "0" atura el temporitzador d'enviament de la temperatura si aquest està en marxa.

Per al control de la monitorització de temperatures, s'ha implementat de forma que el temporitzador que envia la temperatura del sensor estigui aturat mentre que l'aplicació PC no el requereixi. Per realitzar aquest control es fa servir l'enviament del missatge de configuració amb el camp "M", el qual si és un valor diferent a zero posarà en marxa el temporitzador d'enviament de temperatura amb el valor indicat, en cas contrari, si val zero i el temporitzador està en marxa l'aturarà. Amb això es contribueix a l'estalvi de bateria.

El missatge s'envia en els dos sentits, de l'aplicació PC a les motes i viceversa. Això es degut perquè en els sensors esta implementat la auto-configuració, que vol dir que en el moment que arranquen demanen a l'aplicació PC que l'envii la configuració. Per tant, al engegar-se un sensor envia aquest tipus de missatge amb tots els valors buits menys el camp "moteld" a l'aplicació PC, quan el rep l'aplicació PC li retorna aquest mateix missatge però amb els valors del sistema establerts. Recordar que quan l'aplicació PC s'inicia o l'estació base demana la configuració automàtica, l'aplicació PC envia aquest missatge amb l'identificador de la mota "1", fent que s'envii a tota la resta de sensors de la xarxa. La resta de motes accepten qualsevol missatge amb el seu propi identificador o amb l'identificador "1".

¹¹ L'enviament també es realitza de la mota a l'aplicació PC per l'auto-configuració.

Estructura utilitzada per enviar les dades d'alarmes de la mota¹²:

	Tipus	Camp	Descripció
nx_struct AlarmMsg	nx_uint8_t	moteld	Identificador de la mota.
	nx_uint8_t	classe	Indica el tipus d'alarma (incendi, bateria o caixa).
	nx_uint16_t	countsTemp	Valor llegit del sensor de temperatura (en cas d'alarma d'incendi).
	nx_uint16_t	vrefmilivolts	Valor de referència que utilitza el sensor per la lectura del ADC (en cas d'alarma d'incendi).
	nx_uint8_t	type	Mode en cas d'alarma d'incendi (auto o manual).
	nx_uint8_t	typeAck	Tipus de reconeixement d'alarma (estació, usuari o reiniciar).

Taula 14: Estructura de missatge per l'enviament d'alarmes.

Els camps “countsTemp”, “vrefmilivolts” i “type” són utilitzats exclusivament per les alarmes d'incendi, la resta de camps per totes les alarmes. Un cop que salta qualsevol alarma el sensor envia aquest missatge a l'aplicació PC, aquesta automàticament retorna el mateix missatge al sensor sense variar el camp “typeAck” (que per defecte s'envia amb el valor d'estació), d'aquesta manera el sensor rep confirmació que l'alarma ha estat rebuda per l'aplicació PC. En cas d'alarma d'incendi, l'usuari en qualsevol moment pot reconèixer l'alarma des de l'aplicació PC, la qual envia aquesta estructura de missatge amb l'identificador del sensor, la classe d'alarma com a incendi i el camp “typeAck” amb el valor d'usuari. De la mateixa manera l'usuari des de l'aplicació PC pot reiniciar un sensor per esborrar-li qualsevol alarma que tingui activa o simplement canviar el seu estat a “ON”, amb l'enviament del missatge com si reconegués una alarma d'incendi però el camp “typeAck” amb el valor reiniciar.

4.2 Gestió de ADC per un estalvi de bateria

La utilització de la interfície “Resource” del sistema TinyOS, bé donat per la gestió d'energia del ADC de la mota, ja que una bona pràctica per un estalvi adequat és encendre els sensors solament quan es necessiten per realitzar lectures i, la resta de temps tenen que estar desactivats per evitar un malbaratament de l'ús de la pila.

El problema sorgeix en quan a les lectures dels diferents sensors, que es realitzen en diferents components del codi de la mota i, en intervals de temps asíncrons entre els components. Aquest fet presenta el problema que no se sap si quan un component té de fer una lectura i té d'activar els sensors, hi ha un altre realitzant la mateixa operació i, per tant, es pot donar en un mateix moment operacions d'activació i desactivació dels sensors, produint lectures errònies.

Aquest problema té la solució amb la creació d'un component únic i específic que s'encarrega de l'activació i desactivació dels sensors. En el sistema desenvolupat s'ha creat el component “PowerSensors” per la realització d'aquesta tasca. Per tant, el component és un recurs compartit per part dels components que

¹² L'enviament també es realitza de l'aplicació PC a la mota pel reconeixement d'alarmes o reiniciar el sensor.

realitzen lectures dels sensors. La gestió d'aquest recurs és implementada amb ajuda de la interfície “Resource”, que actua com a un semàfor indicant segons ordre de peticions, l'ús del component compartit per evitar accessos concurrents al mateix.

Les comandes i esdeveniments que disposa la interfície “Resource” són les indicades en la taula següent:

<i>Tipus</i>	<i>Signatura</i>	<i>Retorn</i>	<i>Descripció</i>
Comanda	<i>request()</i>	<i>error_t</i>	Petició per l'accés al recurs compartit.
Comanda	<i>immediateRequest()</i>	<i>error_t</i>	Petició immediata per l'accés al recurs compartit.
Esdeveniment	<i>granted()</i>	void	Esdeveniment que ens indica que es pot accedir al recurs amb garantia de ser l'únic.
Comanda	<i>release()</i>	<i>error_t</i>	Alliberament del recurs.
Comanda	<i>isOwner()</i>	bool	Per comprovar si l'usuari actual és el que té l'accés.

Taula 15: Composició de les signatures de la interfície Resource.

Amb la utilització d'aquesta interfície s'ha d'utilitzar un component del sistema que ens faci d'àrbitre per gestionar l'ordre en els accessos al recurs. En aquesta implementació s'utilitza el component “FdfsArbiterC” del sistema, que és un àrbitre amb la política del que primer entra, primer se'l serveix (First-Come, First-Serve).

Per als components que utilitzen el recurs compartit, o sigui el “TempC”, “BatC” i “BoxC” que són els que realitzen lectures dels sensors, cada cop que vulguin realitzar una lectura procedeixen de la forma que indica el diagrama de flux (figura 33) següent:

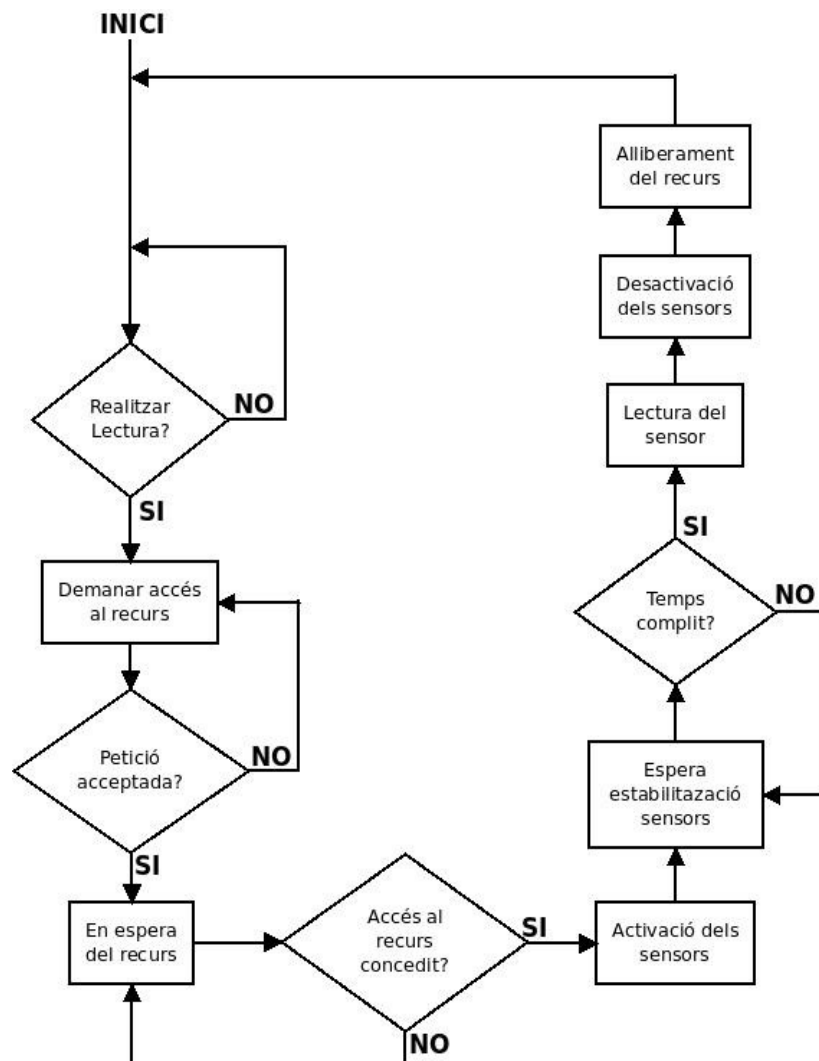


Figura 33: Diagrama de flux per la realització de lectures dels sensors.

4.3 Comunicació segura en l'enviament d'alarmes i altres

Per al sistema desenvolupat en aquest projecte, detecció d'incendis, és molt important l'assegurament de la recepció de l'enviament de certs missatges, com el d'alarmes d'incendi, que per la seva importància no es pot perdre cap.

Per part dels sensors qualsevol alarma activa el seu temporitzador corresponent que passat un cert temps, si no s'ha rebut la confirmació de l'alarma torna a enviar la mateixa alarma. Per tant, aquest tipus de reconeixement d'alarmes dels sensors esta implementat de forma manual.

Per part de l'estació base, el retorn dels reconeixements d'alarmes, l'enviament dels missatges de configuració i l'enviament del tren de missatges utilitzat pel sensor per la comprovació de la cobertura, s'utilitza la interfície "PacketLink" que ens ofereix els reconeixements automàtics dels missatges. Aquesta interfície a diferència de la "PacketAcknowledgement", ens proporciona uns controls més exhaustius, com

el nombre d'intents de re-enviament del missatge en cas de no rebre confirmació i la configuració de l'interval de temps (*delay*) que hi ha entre els re-enviaments.

Aquest control més específic dels reconeixements automàtics en l'estació base, bé donat per evitar la saturació del xip de ràdio, ja que el nombre de sensors pot ser superior a 1 i, un tràfic molt intens d'enviament de missatges pot originar que no es rebin cap missatge d'altres sensors i arribar a saturar l'estació base.

Les comandes que disposa la interfície "PacketLink" són les indicades en la taula següent:

Tipus	Signatura	Retorn	Descripció
Comanda	<i>setRetries(message_t *msg, uint16_t maxRetries)</i>	<i>void</i>	Estableix el nombre d'intents en cas de no rebre confirmació.
Comanda	<i>setRetryDelay(message_t *msg, uint16_t retryDelay)</i>	<i>void</i>	Estableix l'interval de temps entre cada re-intent.
Comanda	<i>getRetries(message_t *msg)</i>	<i>uint16_t</i>	Retorna el valor establert de re-intents.
Comanda	<i>getRetryDelay(message_t *msg)</i>	<i>uint16_t</i>	Retorna el valor establert de l'interval de temps dels re-enviaments.
Comanda	<i>wasDelivered(message_t *msg)</i>	<i>bool</i>	Mostra si s'ha rebut confirmació de recepció en l'enviament d'un missatge.

Taula 16: Composició de les signatures de la interfície PacketLink.

En les següents figures es mostren els diagrames de flux de l'enviament de missatges d'alarma per part del sensor (figura 34) i dels missatges enviats per ràdio per part de l'estació base (figura 35):

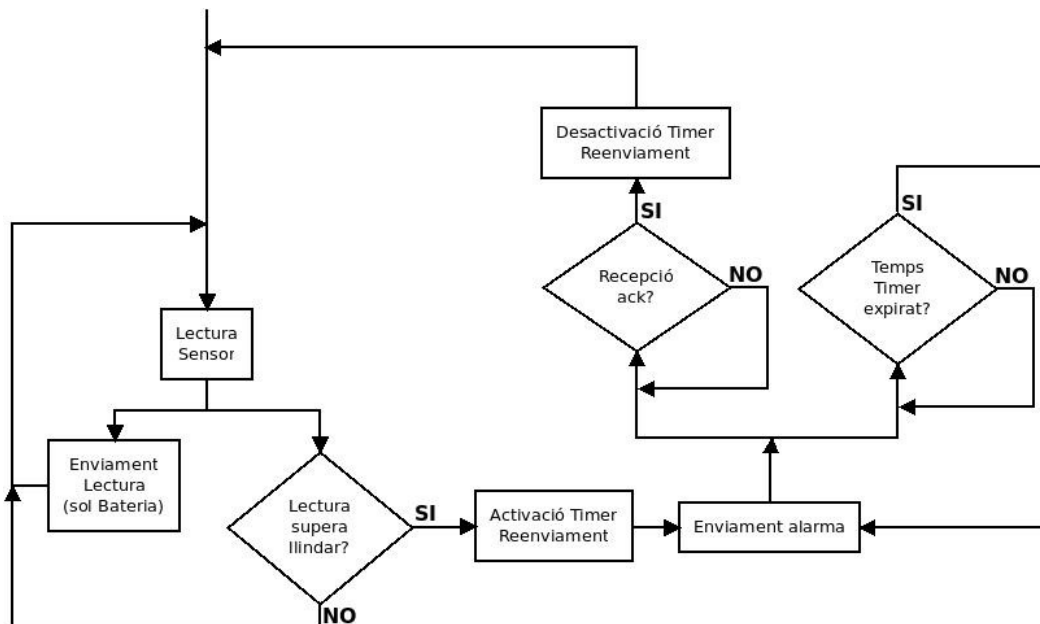


Figura 34: Diagrama de flux d'enviament de missatges d'alarmes del sensor.

En la figura 34 es veu clarament com la implementació dels reconeixements per part del sensor, en l'enviament d'alarmes, es realitza de forma manual a través d'un temporitzador.

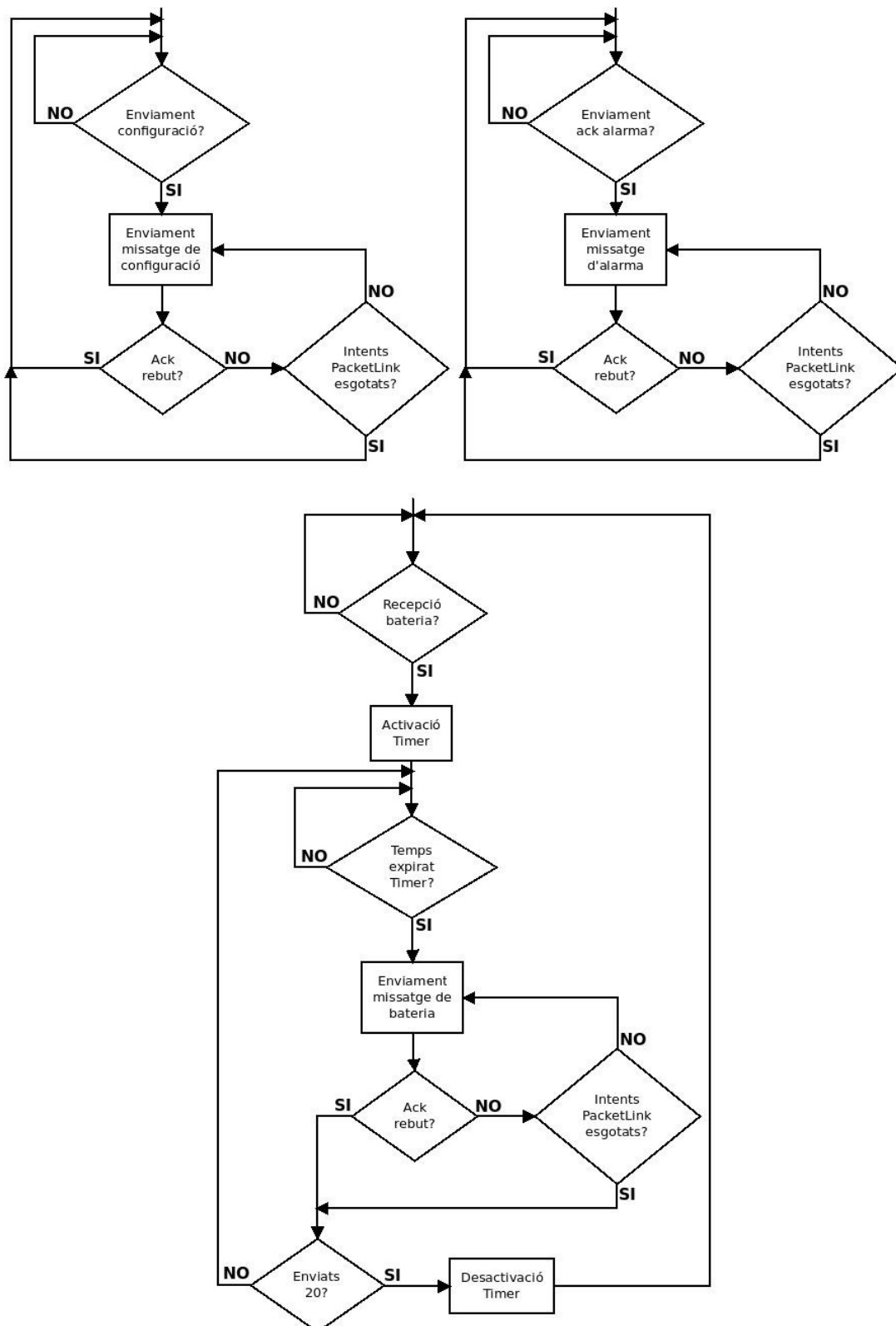


Figura 35: Diagrames de flux dels diferents missatges enviats per la base.

En aquest cas, els reconeixements, es fan amb ajuda de la interfície “PacketLink”. La interfície, gestiona els acks de forma automàtica i amb un nombre limitat d'intents en cas de no rebre confirmació en els enviaments realitzats.

Capítol 5. *Bugs* detectats en el sistema desenvolupat

Existeixen alguns errors en diferents llocs del sistema desenvolupat, es citen a continuació:

La indicació de cobertura en el mateix sensor ha sigut el que ha costat més desenvolupar, i més errors ha donat. Per tant, quan engeguem un sensor i el posem en marxa (led roig intermitent) es normal que passat L segons indiqui durant uns segons fora de cobertura, després en el 90% del temps la indicació es correcta, encara que també es normal que en algun moment no hagi rebut els missatges adequats de la base, i també pot indicar alguns instants fora de cobertura.

Per part de la base no es massa recomanable exigir-li la lectura de les bateries cada segon i alhora monitoritzar les temperatures també cada segon dels sensors. Ja que la base té la feina de respondre a cada missatge de bateria rebut amb un tren de missatges (20) els quals ocupen un cert temps en la seva transmissió. Això pot provocar que si està massa carregat de missatges a enviar pot perdre missatges que rep i, per tant, en la monitorització de les lectures de bateria l'aplicació no el rebrà i pot indicar fora de cobertura del sensor al qual pertany el missatge perdut. Com a conclusió el sistema funcionarà amb més fiabilitat amb uns temps de lectures no massa baixos. La implementació de cues de missatges en l'estació base milloraria aquest aspecte, però queda pendent per un desenvolupament futur.

Comentar també, que quan es rep una alarma l'aplicació respon amb un *ack*, si aquest *ack* no arriba al sensor de l'alarma aquest torna a enviar-la (cada 10s la d'incendi, 30s la de caixa i 30s la de bateria). El *ack* de l'aplicació en la base està implementat amb un segon *ack* que en cas de no rebre confirmació fa fins a 20 re-intents amb un cert "delay" entre ells (implementat amb ajuda de la interfície "PacketLink"), es pot donar que algun cop no rebi cap d'ells i per tant l'alarma es tornarà a enviar al sistema i aquest la tornarà a mostrar. Aquest mateix problema, que no arribi bé la petició, es pot tindre quan enviem alguna configuració, reconeixem alguna alarma o reiniciem algun sensor, igualment es pot repetir l'acció des del menú principal de l'aplicació. S'ha de dir que en la majoria dels cassos amb el nombre d'intents establert el funcionament es correcte. Per últim, comentar que la recepció de les alarmes per l'estació base està assegurat al 100%.

Sobre l'accionament de l'alarma d'incendi manual, polsador, dir que si es pitja només arrancar el sensor i aquest encara no ha fet cap lectura de temperatura perquè no ha passat N temps, el valor enviat de temperatura de l'alarma és de -50°C. Per tant, es suposa que quan es faci la instal·lació de la xarxa de sensors, es faran proves de lectures tant de bateria com de temperatures, per posar en marxa el sistema abans d'instal·lar qualsevol sensor sense comprovar el seu funcionament.

Per la monitorització de temperatures el valor que s'ha d'introduir té que ser superior al valor de la variable del sistema N, en cas contrari, s'envia la mateixa lectura de temperatura fins que passat N temps es llegeix un nou valor.

Capítol 6. Viabilitat tècnica

La viabilitat tècnica d'aquest projecte es positiva, ja que el cost d'implementació i la facilitat d'instal·lació són un dels punts més destacats. Tot això, junt amb unes altres característiques, fa que aquest sistema de detecció d'incendis sigui molt interessant per a certes organitzacions. No obstant, s'ha de dir que el sistema no s'ha pogut comprovar amb un nombre de sensors superior a u en la xarxa de sensors, per tant no tenim cap seguretat que hi hagin nous *bugs* no detectats. Per tant, en principi tindria una viabilitat per a organitzacions que volguessin fer una instal·lació d'una xarxa molt petita, en quan a nombre de sensors; afegint que el tipus de xarxa dels sensors (estrella) també representa una limitació en aquest aspecte. A continuació es mostra dos llistats, un indicant-hi els punts forts del sistema implementat i, l'altre els punts dèbils.

Punts forts:

- Cost d'implementació.
- Instal·lació molt ràpida i econòmica.
- Recomanat per a instal·lacions a on el cablatge pugui ser un problema.
- Robustesa del sistema dels sensors.
- Fiabilitat en el funcionament.
- Assegurament en l'enviament de les alarmes al 100%.
- Facilitat d'ús.
- Protecció contra desmuntatges no autoritzats dels sensors.

Punts dèbils:

- Falta de proves amb un nombre de sensors alt, possibles *bugs* no detectats.
- Tipologia de la xarxa (estrella) amb limitacions, altres tipologies són més recomanables.
- Les comunicacions de la xarxa dels sensors no estan xifrades.
- Ús del mateix canal de RF pels sensors.
- Potència d'emissió fixa dels sensors sense la possibilitat de variació per l'estalvi de bateries.
- Memòria de l'aplicació PC volàtil, respecte als sensors registrats i a les alarmes.

Capítol 7. Valoració econòmica

El pressupost establert per al desenvolupament i instal·lació d'aquest sistema, consta dels següents punts:

Material	Unitats	Cost/Unitat (€)	Total (€)
Ordinador	1	500	500
Sistema operatiu (Linux)	1	0	0
Sistema operatiu (TinyOS)	1	0	0
Motes plataforma COU24	2	50	100
Encapsulat motes	2	12	24
Buzzer	1	18	18
Diversos components electrònics per la fabricació del "driver" pel control del Buzzer	1	12	12
Polsadors exteriors pel disparo manual d'alarma d'incendi	2	9	18
Cost total			672

Taula 17: Costos de materials per al desenvolupament del sistema.

Desenvolupament	Hores	Cost/Hores (€)	Total (€)
Disseny codi de sensor	70	30	2100
Disseny codi d'estació base	20	30	600
Disseny d'aplicació PC	40	30	1200
Cost total			3900

Taula 18: Costos de desenvolupament de codi del sistema.

Instal·lació	Hores	Cost/Hores (€)	Total (€)
Muntatge i instal·lació de sensor	0,5	18	9
Muntatge i instal·lació d'estació base	0,5	18	9
Instal·lació de programari	0,25	18	4,5
Posta en marxa i proves	1	18	18
Cost total			40,5

Taula 19: Costos d'instal·lació del sistema.

Manteniment	Unitats	Cost/Unitat (€)	Total (€)
Canvi de bateries	2	0,9	1,8
Comprovació periòdica del sistema	1	40	40
Actualització de programari	1	50	50
Cost total			93,6

Taula 20: Costos de manteniment dels sistema.

Cost global de desenvolupament del sistema: **4704,3€**

Capítol 8. Conclusions

En aquest apartat es detallen tots els objectius aconseguits, que estan definits al principi del projecte, una sèrie de propostes de millora del sistema i, una descripció dels coneixements que s'han adquirit durant el desenvolupament d'aquest projecte.

8.1 Conclusions

Els objectius aconseguits respecte als definits al principi del projecte, han estat tots. Així la llista d'objectius que estan implementats en el sistema és la següent (no es mostren els detalls de cada objectiu, ja que són els mateixos que els d'inici de projecte):

Objectius bàsics:

1. Detectar incendis. (#implementat)
2. Notificar automàticament l'alarma de forma remota. (#implementat)
3. Notificar automàticament l'alarma de forma local. (#implementat)
4. Sistema manual per activar l'alarma. (#implementat)
5. Transmissió d'alarmes sense pèrdues. (#implementat)
6. Reconeixement d'alarmes des de l'aplicació d'usuari. (#implementat)
7. Monitoritzar la bateria dels nodes de la xarxa. (#implementat)
8. Mostrar els diferents estats del sistema en el node. (#implementat)
9. Sistema de protecció de caigudes. Els nodes no es poden quedar "penjats" (WatchDog). (#implementat)
10. Sistema de debug de l'aplicació. (#implementat)
11. Prova de cobertura. (#implementat)
12. Sistema d'activació. (#implementat)
13. Interfície d'usuari. (#implementat)

Objectius extres:

14. Control de lluminositat (control d'obertura de la caixa d'un sensor). (#implementat)
15. Integrar un buzzer per notificar de forma acústica l'alarma. (#implementat)
16. En la prova de cobertura informar de la potència (dB) rebuda. (#implementat)
17. Interfície gràfica per l'aplicació del PC. (#implementat)

8.2 Propostes de millora

Una sèrie de millores per al sistema desenvolupat podrien ser les següents:

Sobre el registre d'emmagatzematge d'alarmes de l'aplicació PC:

Aquest registre és volàtil, o sigui que aquest s'esborra en cas de sortir de l'aplicació del PC. La connexió a una base de dades seria una solució interessant, ja que permetria emmagatzemar a part de les alarmes valors de bateries, temperatures,... Això permetria la inserció en l'aplicació PC d'una opció per realitzar gràfics temporals de les diferents variables.

Tipologia de la xarxa:

La tipologia de xarxa desenvolupada en la xarxa de sensors del sistema és d'estrella. Aquest tipus de xarxes tenen algunes limitacions que fan que no siguin les més recomanables, per exemple: la caiguda per fallada d'un sol node provoca l'aturada de la xarxa, les distàncies entre nodes de la xarxa són limitades, ja que qualsevol sensor té de tindre cobertura des del node que fa d'estació base, això limita la xarxa a unes desenes de metres. Una xarxa de tipologia en malla, amb cada node amb funció d'encaminament (multi-salt), possibilita xarxes de grandària molt més grans i, són més robustes en cas de caiguda d'algun node.

Desconnexió de la visualització de l'estat del sensor a través dels leds:

L'estalvi de bateria en els sensors és un dels objectius més importants de les xarxes de sensors sense fils. Un sensor a través dels seus leds mostra els diferents estats possibles, indefinidament en el temps. Això provoca un malbaratament de l'energia de la bateria, ja que durant períodes de temps llargs aquesta visualització no té cap utilitat pràctica. La implementació d'un sistema que actives la visualització de l'estat del sensor solament en cas d'alarmes o a petició de l'usuari durant un cert temps, milloraria molt el temps de vida de la bateria.

Xifratge de les comunicacions dels sensors:

Assegurament de que tercers sense autorització puguin obtindre informació de l'organització.

Canal de RF del sensor i control de potència d'emissió:

Possibilitat de canvi de canal d'emissió cada cert temps automàticament, per motius d'eficiència radioelèctrica com per seguretat. Control automàtic de potència d'emissió per contribuir a un estalvi de bateria.

8.3 Auto-avaluació

Durant el desenvolupament d'aquest projecte s'han adquirit diverses competències noves. Una d'elles és l'aprenentatge d'una distribució adequada del temps en la realització d'un treball, que em permeti assolir uns objectius establerts en el desenvolupament d'un projecte. En la següent llista es mostren la resta de competències adquirides:

- La introducció en el món dels sistemes encastats, que fins ara estava envoltat d'ells però no havia prestat atenció. Conèixer l'estat actual d'aquest sistemes, m'ha fet veure que és un àrea en plena expansió i, que presenta un futur molt atractiu a nivell professional.
- La capacitat d'aprenentatge de nous conceptes, com llenguatges de programació o sistemes operatius, a través de documentació en anglès.
- En qualsevol projecte, aprendre a fixar objectius reals, o sigui que puguin ser portats a terme amb la tecnologia actual.
- Coneixement de les xarxes de sensors sense fils com una tecnologia puntera en plena expansió i, que en l'actualitat s'estan implementant en moltes àrees.

Capítol 9. Glossari

sistemes encastats *m* És un sistema de computació dissenyat per portar a terme una a varies funcions específiques pel qual ha estat dissenyat exclusivament, normalment s'utilitza en sistemes de computació en temps real. Són exemples un mòdem de ADSL, un taxímetre, equips de medicina d'un hospital,...
en embedded systems

microcontrolador *m* Microprocessador altament integrat dissenyat específicament per al seu ús en sistemes encastats. Els microcontroladors solen incloure un processador integrat, memòria (una petita quantitat de memòria RAM, ROM o ambdós), i altres perifèrics en el mateix xip. Els exemples més comuns són PIC de Microchip, el 8051, 80196 d'Intel i una sèrie de Motorola 68HCxx.
en microcontroller

mota *f* També es coneix com a node i, és un dispositiu petit el qual normalment posseeix una memòria molt reduïda, un processador amb una freqüència de rellotge baixa, un mòdul de ràdio amb una cobertura d'unes desenes de metres i, s'alimenten a través de bateries o piles. Una agrupació en xarxa de motes forma una xarxa de sensors sense fils.
en mote

xarxa de sensors sense fils *f* Xarxa que es compon per una sèrie de nodes o motes, els quals es comuniquen entre ells per via ràdio.
en wireless sensors network
sigla WSN

sistema operatiu *m* En el context dels sistemes encastats, element de programari que fa possible el funcionament en multi-tasca distribuïnt els recursos disponibles entre les diferents tasques que cal dur a terme. Un sistema operatiu consisteix típicament en un conjunt de crides al sistema, un planificador de tasques, una rutina de servei d'interrupcions, un comptador/temporitzador de sistema i un conjunt de controladors dels dispositius perifèrics.
en operative system (OS)

sistema operatiu en temps real *m* Sistema operatiu dissenyat específicament per al seu ús en sistemes de temps real. S'utilitzen en els sistemes en què és imprescindible complir unes especificacions temporals determinades. Per exemple, garantir uns temps màxims en la resolució d'un càlcul o en la resposta a un esdeveniment extern.
en real-time operative system
sigla RTOS

tasca *f* Qualsevol càlcul individual, conjunt de càlculs, la lògica de la presa de decisions, intercanvi d'informació o combinació d'elles que ha de dur a terme en temps d'execució el programari. És l'abstracció central dels RTOS. Quan s'utilitza un RTOS, cada tasca ha de mantenir la seva pròpia còpia del punter d'instruccions de la CPU i els registres de propòsit general. La diferència principal amb els processos és que les tasques comparteixen un espai de memòria comuna. Per tant, cal anar amb compte per a evitar sobreescriure codi, dades i piles d'altres tasques.
en task

Capítol 10. Bibliografia

Vilajosana I. (2011). *Introducció als sistemes encastats* (mòdul assignatura: TFC Àrea Sistemes Encastats). Barcelona: FUOC.

López J., Morell A. (2011). *El maquinari: arquitectura i components* (mòdul assignatura: TFC Àrea Sistemes Encastats). Barcelona: FUOC.

Hernández F., Prades J. (2011). *El programari* (mòdul assignatura: TFC Àrea Sistemes Encastats). Barcelona: FUOC.

Gómez J. (2011). *Sistemes operatius encastats* (mòdul assignatura: TFC Àrea Sistemes Encastats). Barcelona: FUOC.

Vilajosana I. (2011). *Simulació i test* (mòdul assignatura: TFC Àrea Sistemes Encastats). Barcelona: FUOC.

Levis P. (2006). *TinyOS Programming*. Stanford, EUA: Stanford University.

Levis P. (2006). *TinyOS Enhancement Proposal* (TEP). Stanford, EUA: Stanford University.

TinyOS (2010). *TinyOS Documentation Wiki*. [en línia]. http://docs.tinyos.net/tinywiki/index.php/Main_Page [data de consulta: 02/03/12].

Viquipèdia UOC (2012). *Inici24*. [en línia]. <http://cv.uoc.edu/app/mediawiki14/wiki/Inici24> [data de consulta: 28/02/12].

Viquipèdia (2012). *Sistemas empotrados*. [en línia]. http://es.wikipedia.org/wiki/Sistemas_empotrados [data de consulta: 20/04/12].

Wikispaces (2012). *Sistema embebido*. [en línia]. <http://g11empresas.wikispaces.com/Sistema+embebido> [data de consulta: 24/04/12].

UPV (2010). *Desarrollo de aplicaciones basadas en WSN*. [en línia]. <http://riunet.upv.es/bitstream/handle/10251/8592/PFC%20-%20DESARROLLO%20DE%20APLICACIONES%20BASADAS%20EN%20WSN.pdf> [data de consulta: 02/05/12].

Fundación Deusto (2007). *Experiencias en la aplicación de Wireless Sensor*. [en línia]. <http://docencia.etsit.urjc.es/moodle/mod/resource/view.php?id=3760> [data de consulta: 02/05/12].

Source Juice (2010). *Red inalámbrica de sensores (WSN) la tecnología para escanear*. [en línia]. <http://www.sourcejuice.com/1332031/2010/04/28/Red-inal%C3%A1mbrica-sensores-WSN-tecnolog%C3%ADa-escanear/es/> [data de consulta: 02/05/12].

borrowBITS (2012). *Instalación de NS2 para redes de sensores inalámbricos*. [en línia]. <http://borrowbits.com/tag/wsn/> [data de consulta: 05/05/12].

Proyecto RIMSI(2007). *Qué es WSN*. [en línia]. http://www.motas.es/rimsi/index.php?option=com_content&task=view&id=2&Itemid=3 [data de consulta: 05/05/12].

Universidad Carlos III (2011). *WIRELESS SENSOR NETWORKS: Estado del Arte e Investigación*. [en línia]. http://arcos.inf.uc3m.es/~sescolar/index_files/presentacion/wsn.pdf [data de consulta: 07/05/12].

Universidad de Granada (2007). *Redes de Sensores Inalámbricos*. [en línia]. http://atc.ugr.es/~aprieto/TIC_socio_sanitario/A11_4_05_Redos_sensores.pdf [data de consulta: 07/05/12].

UTFSM Chile (2010). *Redes de Sensores Inalámbricos*. [en línia]. http://profesores.elo.utfsm.cl/~tarredondo/info/networks/Presentacion_sensores.pdf [data de consulta: 07/05/12].

Capítol 11. Annexos

11.1 Connexió del *buzzer* a la mota (*driver*)

El sistema desenvolupat inclou un *buzzer* (sirena) connectat a cada mota que fa de sensor. Com que el port que s'utilitza del connector de propòsit general de la mota té unes certes limitacions, ofereix una tensió d'uns 3V i un corrent màxim de 40mA. Amb aquests paràmetres, fa que es necessiti un driver (controlador) per tal d'activar o desactivar un *buzzer* que normalment s'alimenta amb una tensió superior i, per tant, absorbeix un corrent major. Una possible circuit electrònic que faci de *driver* per al *buzzer* pot ser el mostrat en la figura següent:

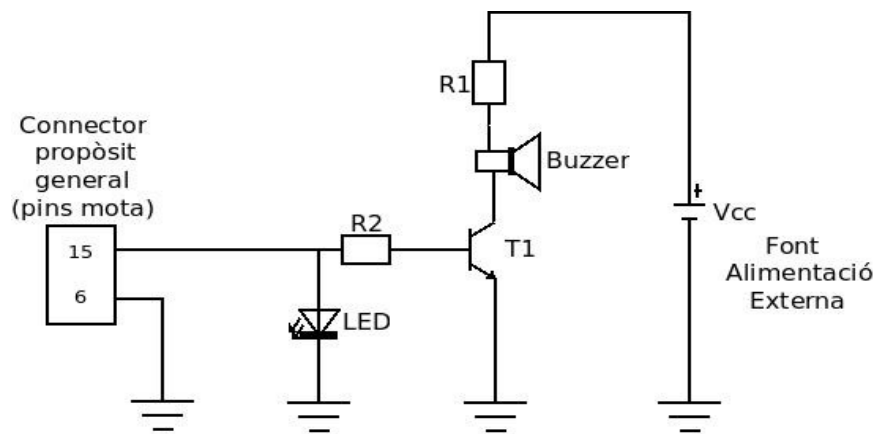


Figura 36: Driver per la connexió del buzzer a la mota.

Igualment, s'ha de comentar que al mercat hi ha la possibilitat de trobar algun *buzzer* que ja disposi d'una electrònica integrada, la qual faci la mateixa funció que aquest circuit mostrat.

11.2 Manual d'usuari

Les diferents opcions que disposa l'aplicació PC són explicades al menú d'ajuda que té inclòs, el seu contingut és el següent:

[Configuració N]: Estableix el temps de la variable N, la qual indica l'interval temps en el que es llegeix la temperatura i es comprova el llindar d'alarma establert. El valor mínim que es pot introduir és a partir d'un segon.

[Configuració L]: Estableix el temps de la variable L, la qual indica l'interval temps en el que es llegeix la bateria, es comprova el llindar d'alarma establert i s'envia a l'estació base. El valor mínim que es pot introduir és a partir d'un segon.

[Conf. Llindar Temperatura]: Estableix el llindar d'activació de l'alarma de temperatura. El valor introduït té que estar comprès entre 1°C i 100°C, amb possibilitat de dos decimals.

[Conf. Llindar Bateria]: Estableix el llindar d'activació de l'alarma de bateria. El valor introduït té que estar comprès entre 1V i 3.5V, amb possibilitat de dos decimals.

[Monitoritzar Bateries i Cobertura]: Permet llistar per pantalla les dades de les bateries que es va rebre de tots els sensors que estan actius, en el cas que algun sensor registrat en el sistema, no enviï les seves dades de bateria es mostrarà un missatge de fora de cobertura del mateix. Per finalitzar el llistat s'ha de prémer la tecla 'Enter'.

[Monitoritzar Temperatures]: S'activa en els sensors l'enviament de les lectures de temperatures llegides, i s'en van llistant per pantalla. Per finalitzar el llistat s'ha de prémer la tecla 'Enter'.

[Reconèixer Alarma d'Incendis]: Un cop s'ha rebut una alarma d'incendi, en cas que l'usuari no la reconegui a través del mateix missatge, tancant l'avís pel botó de la finestra, la pot reconèixer més tard a través d'aquesta opció. El sensor canviarà d'estat i, si és el cas, el buzzer connectat al sensor s'aturarà. Introduir l'identificador del sensor a reconèixer l'alarma.

[Reiniciar Sensor]: L'usuari un cop controlat l'incendi, pot reiniciar el sensor afectat i tornar-ho a deixar operatiu. S'ha d'introduir l'identificador del sensor el qual ha enviat l'alarma.

[Esborrar Sensor del Sistema]: En cas de voler deixar fora de servei un sensor, per modificacions de la instal·lació, es pot esborrar l'identificador del sensor del sistema perquè no el tingui en compte alhora de comprovar la cobertura. S'ha d'introduir l'identificador del sensor el qual es vol esborrar del sistema.

[Llistar Alarmes]: Permet llistar les alarmes d'incendi, bateria i caixa detectades pel sistema.

11.3 Manual d'instal·lació

Sistema de detecció d'incendis basat en tecnologia WSN

Versió: 1.0

Manual d'instal·lació del sistema

Requeriments del sistema:

El sistema requereix tindre instal·lat en el PC, en el qual s'executarà l'aplicació i connectarà l'estació base, el sistema operatiu TinyOS. Més informació en: <http://www.tinyos.net>.

Pas previ:

Descompressió del arxiu zip en un directori qualsevol del disc dur del PC, es crearà una carpeta anomenada Codi_Final. Les referències a les diferents ubicacions s'entendran a partir del directori a on s'ha ubicat la carpeta descomprimida.

Instal·lació del programari d'un sensor:

El programari es troba en /Codi_Final/Sensor.

Per compilar:

```
make cou24
```

S'estableix en la mota l'adreça i l'identificador amb el valor "2" pel primer sensor i així successivament:

```
tos-set-symbols build/cou24/main.srec build/cou24/main.srec.2 TOS_NODE_ID=2 ActiveMessageAddressC__addr=2
```

Per programar la mota:

```
meshprog -t/dev/ttyUSB0 -f./build/cou24/main.srec.2
```

Instal·lació del programari de l'estació base:

El programari es troba en /Codi_Final/Estacio_Base.

Per compilar:

```
make cou24
```

S'estableix en la mota l'adreça i l'identificador amb el valor "1":

```
tos-set-symbols build/cou24/main.srec build/cou24/main.srec.1 TOS_NODE_ID=1 ActiveMessageAddressC__addr=1
```

Per programar la mota:

```
meshprog -t/dev/ttyUSB0 -f./build/cou24/main.srec.1
```

Execució de l'aplicació PC:

El programari es troba en /Codi_Final/Aplicacio_PC/bin.

Primer de tot es té que executar l'aplicació de TinyOs "Serial Forwarder", amb els paràmetres del port concret on estigui connectada la mota (estació base) i el port TCP que utilitza l'aplicació per la seva connexió amb altres aplicacions. Per tant obrim un terminal i com a exemple (el server port per defecte és 9002):

```
java net.tinyos.sf.SerialForwarder -comm serial@/dev/ttyUSB0:19200
```

Abans d'executar l'aplicació PC es pot veure i/o modificar si es vol el fitxer "Valors.conf", ubicat en la mateixa carpeta. Aquest fitxer conté els valors de les diferents variables que el sistema carregarà en el moment d'iniciar-se i, per tant enviarà a tota la xarxa de sensors. En cas que un valor sigui erroni l'aplicació mostrarà un missatge d'error i no s'iniciarà (el nombres decimals es tenen que escriure amb el punt i no amb la coma). En la figura següent es mostra el contingut d'aquest fitxer.



```
*PAC3 x  Valors.conf x
#Connexió (ip:port):
localhost:9002
#N (segons):
12
#L (segons):
5
#Llindar Temperatura entre 1 i 100°C(format XX.XX):
50.00
#Llindar Bateria entre 1 i 3.5V (format XX.XX):
2.5
#FI
```

Figura 37: Mostra del contingut de l'arxiu de configuració (manual).

La línia "localhost:9002" ens indica la ip de la màquina on s'està executant l'aplicació servidor "Serial Forwarder" juntament amb el port de connexió . En el exemple s'executa sobre la mateixa màquina i en el port per defecte 9002.

Primer s'executa l'aplicació "Serial Forwarder", la següent imatge mostra el resultat:

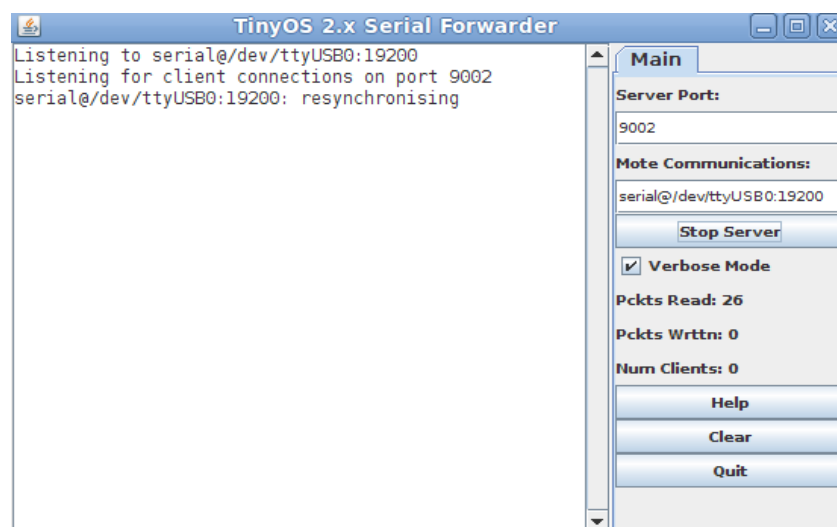


Figura 38: Serial Forwarder.

Seguidament executar l'aplicació. Obrir un terminal del sistema operatiu i ubicar-se dintre de la carpeta anterior, executar:

```
./run.sh (alternativa: java AplicacioMainGUI)
```

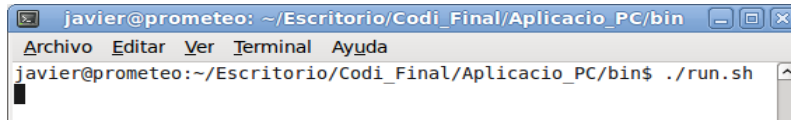


Figura 39: Exemple d'execució des de terminal.

I s'ens obrirà la pantalla principal de l'aplicació PC, mostrada en la següent imatge:

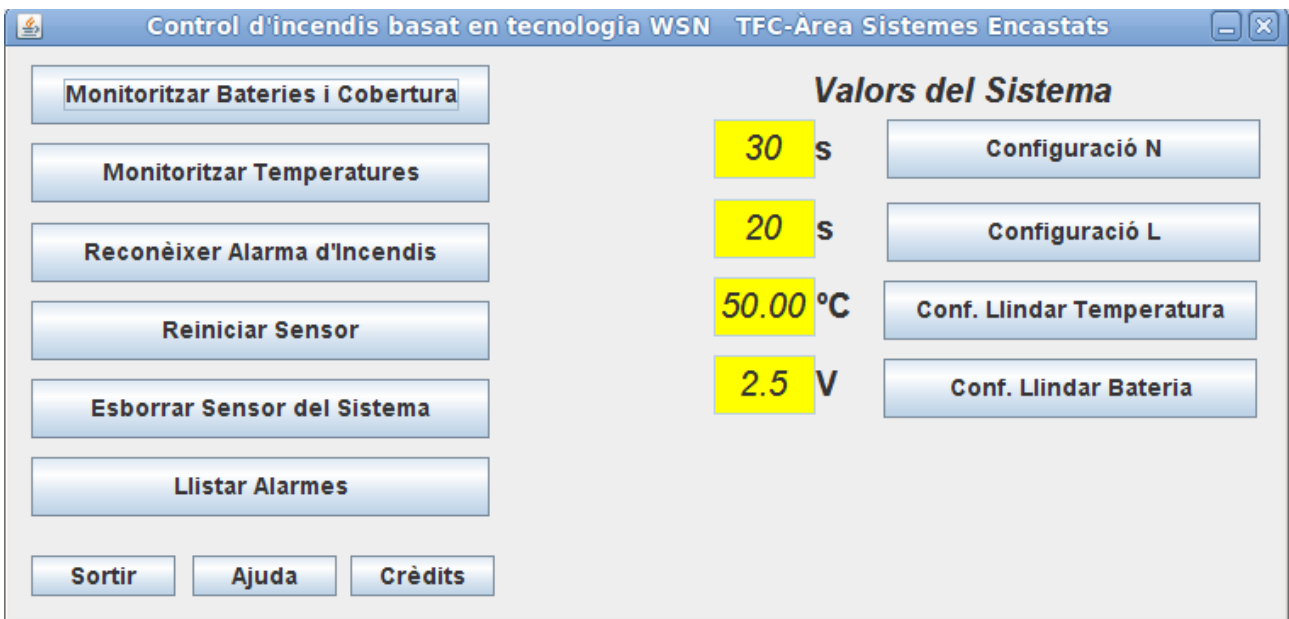


Figura 40: Pantalla principal de l'aplicació PC (manual).

Recordar que l'aplicació disposa d'una opció que ens mostra l'ajuda d'usuari de les diferents opcions existents. Les diferents pantalles que ens mostra l'aplicació per algunes opcions són:

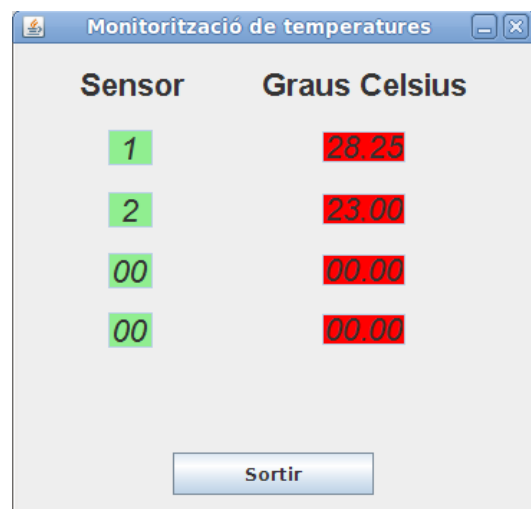
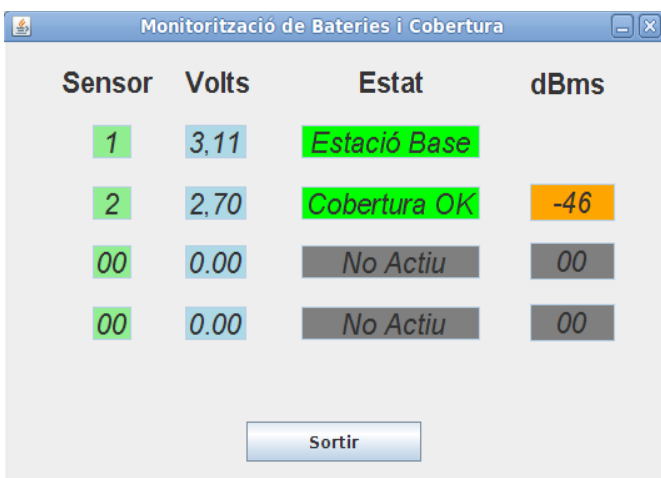


Figura 42: Pantalla de monitorització de bateries i cobertura (manual). Figura 41: Pantalla de monitorització de temperatures (manual).

11.4 Diagrames de connexions de la mota COU24

Microcontrolador:

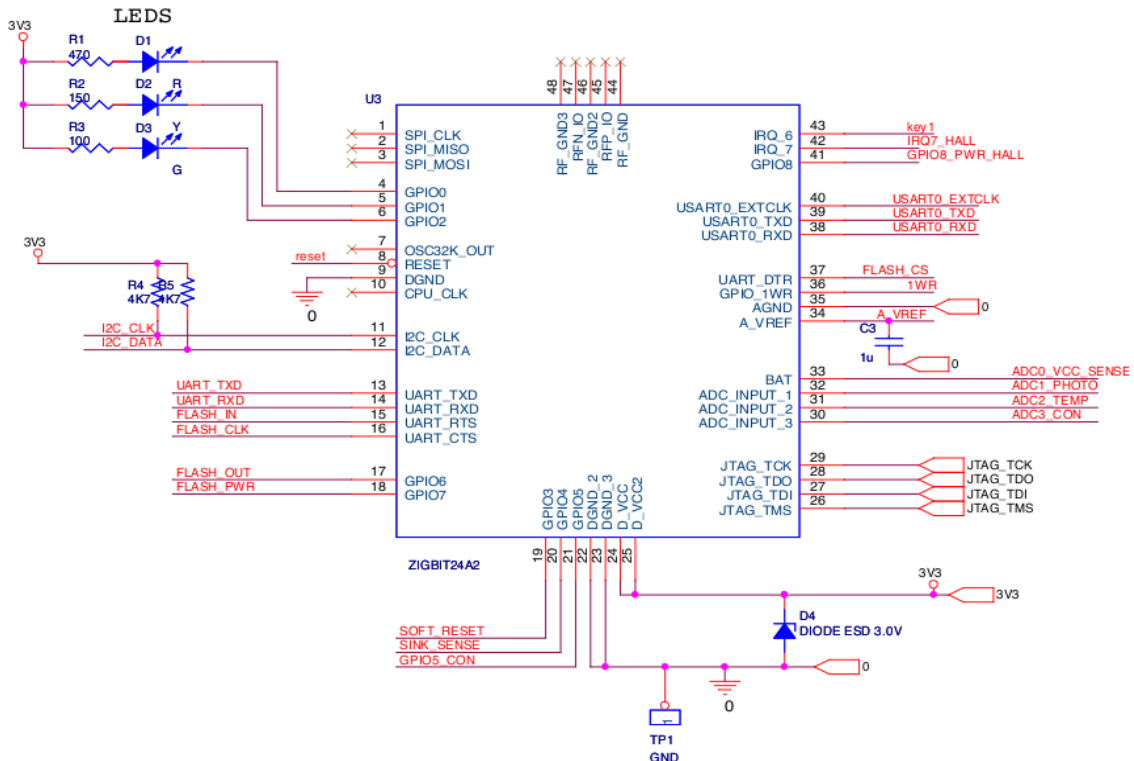


Figura 43: Diagrama de connexions del microcontrolador de la mota COU24.

Connectors:

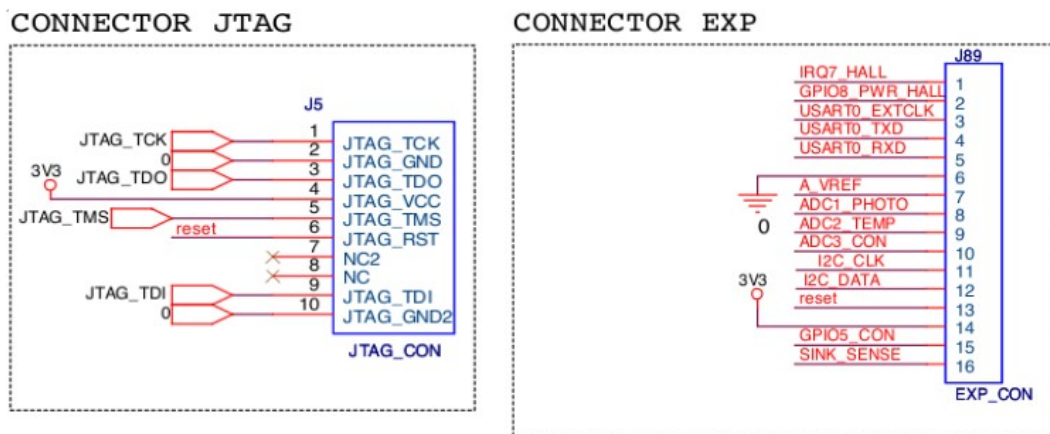


Figura 44: Identificació de pins dels connectors de la mota.

11.5 Característiques dels sensors de la mota COU24

Sensor de llum:

FEATURES

- Visible light response
- Sintered construction
- Low cost

DESCRIPTION

The **PDV-P9003-1** are (CdS), Photoconductive photocells designed to sense light from 400 to 700 nm. These light dependent resistors are available in a wide range of resistance values. They're packaged in a two leaded plastic-coated ceramic header.

APPLICATIONS

- Camera exposure
- Shutter controls
- Night light Controls

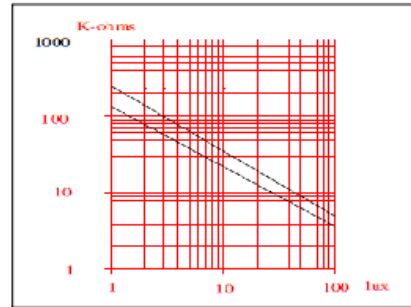


ABSOLUTE MAXIMUM RATING (TA= 23°C UNLESS OTHERWISE NOTED)

SYMBOL	PARAMETER	MIN	MAX	UNITS
V _{pk}	Applied Voltage		150	V
P _{d Δp/Δt}	Continuous Power Dissipation		125	mW/°C
T _O	Operating and Storage Temperature	-25	+75	°C
T _S	Soldering Temperature*		+260	°C

* 0.200 inch from base for 3 seconds with heat sink.

CELL RESISTANCE VS. ILLUMINANCE



ELECTRO-OPTICAL CHARACTERISTICS RATING (TA= 23°C UNLESS OTHERWISE NOTED)

SYMBOL	CHARACTERISTIC	TEST CONDITIONS	MIN	TYP	MAX	UNITS
R _D	Dark Resistance	After 10 sec. @ 10 Lux @ 2856 °K	1			MΩ
R _I	Illuminated Resistance	10 Lux @ 2856 °K	23		33	KΩ
S	Sensitivity	LOG(R100)-LOG(R10)** LOG(E100)-LOG(E10)***		0.85		Ω/Lux
λ _{range}	Spectral Application Range	Flooded	400		700	nm
λ _{peak}	Spectral Application Range	Flooded		570		nm
t _r	Rise Time	10 Lux @ 2856 °K		60		ms
T _f	Fall Time	After 10 Lux @ 2856 °K		25		ms

**R100, R10: cell resistances at 100 Lux and 10 Lux at 2856 °K respectively .

***E100, E10: luminances at 100 Lux and 10 Lux 2856 °K respectively.

Taula 21: Característiques del sensor de llum de la mota COU24.

Sensor de temperatura:

1.0 ELECTRICAL CHARACTERISTICS

Absolute Maximum Ratings †

V_{DD}:..... 6.0V
 Storage temperature: -65°C to +150°C
 Ambient Temp. with Power Applied:.. -40°C to +150°C
 Output Current ±30 mA
 Junction Temperature (T_J): 150°C
 ESD Protection On All Pins (HBM:MM):(4 kV:200V)
 Latch-Up Current at Each Pin: ±200 mA

Taula 22: Valors absoluts del sensor de temperatura de la mota COU24.

Sensor d'efecte Hall:

BU52011HFV (Ta=25°C)

Parameter	Symbol	Ratings	Unit
Power Supply Voltage	V _{DD}	-0.1 ~ +4.5 ^{※9}	V
Output Current	I _{OUT}	±0.5	mA
Power Dissipation	P _d	536 ^{※10}	mW
Operating Temperature Range	T _{opr}	-40 ~ +85	°C
Storage Temperature Range	T _{stg}	-40 ~ +125	°C

※9. Not to exceed P_d

※10. Reduced by 5.36mW for each increase in Ta of 1°C over 25°C
(mounted on 70mm × 70 mm × 1.6mm Glass-epoxy PCB)

Taula 23: Valors absoluts del sensor d'efecte Hall de la mota COU24.

11.6 Característiques elèctriques del mòdul Atmel ATZB-24-A2

Rang de valors màxims i mínims absoluts:

Parameters	Min	Max
Voltage on any pin, except RESET with respect to Ground	-0.5V	VCC + 0.5V
DC Current per I/O Pin		40 mA
DC Current DVCC and DGND pins		200 mA
Input RF Level		+10 dBm

Taula 24: Valors màxims absoluts de la mota ATZB-24-A2.

Valors obtinguts en proves amb condicions de V_{cc} de 3V i temperatura ambient 25°C:

Parameters	Range	Unit
Supply Voltage, V_{CC}	1.8 to 3.6	V
Current Consumption: RX mode	19	mA
Current Consumption: TX mode ⁽¹⁾	18	mA
Current Consumption: Radio is turned off, MCU is active 50% of the time ⁽¹⁾	14	mA
Current Consumption: Power-save mode ⁽¹⁾	6	μ A

Taula 25: Valors obtinguts en proves.

Característiques del xip de ràdio:

Parameters	Condition	Range	Unit
Frequency Band		2.4000 to 2.4835	GHz
Numbers of Channels		16	
Channel Spacing		5	MHz
Transmitter Output Power	Adjusted in 16 steps	-17 to +3	dBm
Receiver Sensitivity	PER = 1%		
On-Air Data Rate		250	kbps
TX Output/ RX Input Nominal Impedance	For balanced output	100	Ω

Taula 26: Característiques del xip de ràdio.

Característiques del mòdul d'interfícies:

Parameters	Condition	Range	Unit
UART Maximum Baud Rate		38.4	kbps
ADC Resolution/ Conversion Time	In single conversion mode	10/200	Bits/ μ s
ADC Input Resistance		>1	M Ω
ADC Reference Voltage (VREF)		1.0 to $V_{CC} - 3$	V
ADC Input Voltage		0 - VREF	V
I ² C Maximum Clock		222	kHz
GPIO Output Voltage (High/Low)	-10/ 5 mA	2.3/ 0.5	V
Real Time Oscillator Frequency		32.768	kHz

Taula 27: Rang de valors del mòdul d'interfície.

Característiques del microcontrolador Atmega1281V:

Parameters	Condition	Range	Unit
On-chip Flash Memory size		128K	bytes
On-chip RAM size		8K	bytes
On-chip EEPROM size		4K	bytes
Operation Frequency		4	MHz

Taula 28: Característiques del microcontrolador Atmega1281.