

Xarxa de sensors sense fils (WSN) per a la detecció d'incendis

Josep Joan Nieto Cuevas
Enginyeria Tècnica de Telecomunicació - Telemàtica

Consultor: Jordi Bécares Ferrés

Lliurament: 12 de juny de 2012

Resum

El present treball de final de carrera té com objecte aprofundir en el disseny i en la implementació d'una xarxa de sensors sense fils. Aquestes xarxes, a diferència de les xarxes convencionals, estan compostes de nodes o estacions amb recursos limitats que transporten informació sobre paràmetres físics, en lloc de dades, veu i serveis com succeeix en una LAN o WLAN tradicional.

Basades en l'estàndard IEEE.802.15.4, es caracteritzen per la seva senzillesa i per ser molt econòmiques, permetent la connectivitat sense fils en aplicacions i entorns que no necessiten una gran potència de transmissió. Les principals característiques són la facilitat de desplegament dels components, la transferència fiable de dades, l'operació en un rang de curt abast, el baix cost i la llarga durada de les bateries dels dispositius. Tot controlat mitjançant un protocol simple i flexible.

Un terme estretament relacionat amb l'evolució de les *Wireless Sensor Networks* és el de ciutat intel·ligent. Aquests tipus de projectes es fonamenten en la utilització de components estàndards de baix cost, on es pot identificar fàcilment una estructura pròpia de les WSN. En aquestes xarxes de transmissió de dades metropolitanes, són molt importants els nodes de recollida de valors ambientals o sensors que permeten als nodes centrals, amb la informació subministrada, prendre les decisions 'intel·ligents'.

La detecció d'incendis requereix d'una gran fiabilitat en els protocols de transmissió així com en els mètodes de detecció. Un sistema eficient de sensors permetrà subministrar a l'usuari, de manera clara i intel·ligible, la informació necessària que li permeti prendre les decisions adients en una situació de risc.

Paraules clau: *TinyOS*, *nesC*, sistema encastat, ciutat intel·ligent, detecció d'incendis, mota, sensor, xarxa de sensors sense fils, WSN.

Àrea TFC: Sistemes encastats

Índex

Introducció	1
Justificació	1
Descripció	2
Objectius	3
Objectius generals	4
Objectius específics	4
Abast	5
Enfocament	6
Planificació	7
Calendari general de projecte	7
Calendari de la PAC 1	10
Calendari de la PAC 2	10
Calendari de la PAC 3	12
Calendari de la PAC 4	13
Descripció de riscos	14
Recursos emprats	16
Recursos de Maquinari	16
Recursos de Programari	17
Productes obtinguts	18
Descripció dels altres capítols	19
Antecedents	20
Estat de l'art	21
Ciutats intel·ligents	21
Xarxes d'abast personal (<i>Personal Area Network</i>)	24
Estudi de mercat	26
Sistemes de detecció d'incendis	28
Descripció funcional	29
Sistema total	29
Detecció d'incendis	30
Fiabilitat en la transmissió	31
Gestió dels nodes	33
Nodes remots	33

Node base	34
Aplicació de l'usuari.....	34
Funcionalitats addicionals	35
Descripció detallada	35
Plataforma de desenvolupament	35
Node remot - FireRemote.....	36
Node local - FireBase.....	43
Aplicació d'usuari - FireClient	44
Viabilitat tècnica.....	50
Valoració econòmica.....	51
Conclusions	52
Revisió dels objectius.....	52
Proposta de millores.....	55
Autoavaluació i conclusions finals	56
Glossari	57
Bibliografia.....	59
Publicacions.....	59
Enllaços electrònics.....	60
Annexos.....	i
Execució i compilació	i
Fórmules de conversió	ii
Consideracions inicials	ii
Tensió (<i>Counts</i> a volts).....	ii
Lluminositat (<i>Counts</i> a lux)	ii
Temperatura (<i>Counts</i> a graus centígrads)	ii
Cobertura (<i>RSSI</i> a <i>dBm</i>).....	ii
Sensors addicionals.....	.iii
Problemes trobats	iv
Entorn de desenvolupament	iv
FireBase	iv
FireRemote.....	v
FireClientvii

II-lustracions

II-lustració 1: Estàndard IEEE 802.15.4, topologies d'estrella, arbre i malla	2
II-lustració 2: Calendari general inicial del projecte	7
II-lustració 3: Calendari general actualitzat del projecte	8
II-lustració 4: Diagrama de Gantt del projecte	9
II-lustració 5: Calendari de la PAC 1	10
II-lustració 6: Planificació ajustada de la PAC2	11
II-lustració 7: Diagrama de Gantt actualitzat de la PAC 3	13
II-lustració 8: Calendari de la PAC 4	13
II-lustració 9: Passarel·la entre diferents tipus de xarxa	26
II-lustració 10: Diagrama general del projecte SISVIA.....	28
II-lustració 11 Estructura del sistema	29
II-lustració 12: Fases en el desenvolupament d'un incendi.....	30
II-lustració 13: <i>ZigBee</i> ® BER / SNR	32
II-lustració 14: IEEE 802.15.4 i <i>ZigBee</i> ®	32
II-lustració 15: Esquema de connexions del mòdul <i>Zigbit 24 A2</i>	37
II-lustració 16: Esquema de connexió dels sensors.....	38
II-lustració 17: Esquema del connector d'expansió de les motes.....	38
II-lustració 18: Esquema de components de <i>FireRemote</i>	39
II-lustració 19: Motes i sensors del projecte	42
II-lustració 20: Connexió al port d'expansió	42
II-lustració 21: Esquema actual de <i>FireBase</i>	43
II-lustració 22: Components de <i>FireBase</i>	43
II-lustració 23: Entorn de desenvolupament de Java	44
II-lustració 24: Esquema del funcionament de l'aplicació.....	45
II-lustració 25: Interfície d'usuari de <i>FireClient</i>	46
II-lustració 26: Pantalla de configuració de funcionament.....	47
II-lustració 27: Registre de missatges rebuts.....	48
II-lustració 28: Pantalla de test del sistema	49

Introducció

El treball de final de carrera que presentem a continuació, va tenir com a objectiu principal el desenvolupament d'un sistema de detecció i alarma d'incendi que, amb la màxima fiabilitat possible, acomplís amb les restriccions imposades per un baix pressupost i un reduït consum.

Justificació

A l'hora de fer un diagnòstic acurat sobre una situació de risc, d'incendi en el nostre cas, és important disposar dels millors mètodes de detecció. Tecnològicament, hi ha una gran disponibilitat de sensors que permeten avaluar alguns dels paràmetres presents en un incident d'aquest tipus amb un elevat grau de precisió. Entre ells, podem distingir detectors de temperatura, detectors de fums, detectors de monòxid de carboni¹, etc.

El problema més rellevant que presenten aquests tipus clàssics de sensors és, a banda del cost del propi sensor, el cost de la instal·lació i la complexitat del desplegament. A més, un cop instal·lats, ens trobem amb la limitació de que són poc flexibles, ja que, per la seva naturalesa, són fixes i no es poden canviar d'ubicació sense un desemborsament addicional, generalment elevat.

Creiem que, donat que en les xarxes sense fils no és necessari un medi físic de propagació, ja que fan servir ones electromagnètiques de baixa potència per tal d'intercanviar dades entre dispositius, aquest tipus de comunicacions ens ofereixen les condicions de llibertat d'utilització i de flexibilitat que requereix el nostre projecte.

És per això que hem treballat en el desenvolupament i en la implementació d'una xarxa sensorial sense fils (*Wireless Sensor Network*) de baix cost basada en l'estàndard IEEE.802.15.4².

¹ Per exemple: http://www.ziton.com/conv_detectors.html

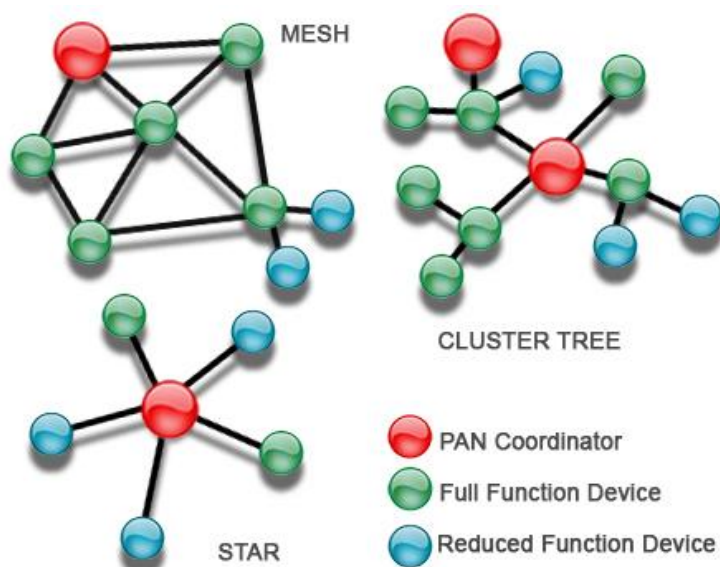
² <http://standards.ieee.org/getieee802/download/802.15.4-2003.pdf>
<http://standards.ieee.org/getieee802/download/802.15.4-2006.pdf>

Aquestes xarxes, conegudes com *LR-WPAN* (*Low-Rate Wireless Personal Area Network*), es caracteritzen per la seva senzillesa i per ser molt econòmiques, permetent la connectivitat sense fils en aplicacions i entorns que no necessiten una gran potència de transmissió. Les principals característiques d'una *LR-WPAN* són la facilitat d'instal·lació, la transferència fiable de dades, l'operació de curt abast, el baix cost i la llarga durada de les bateries. Tot controlat mitjançant un protocol simple i flexible.

Descripció

La xarxa de sensors està coordinada per un node principal (*Personal Area Network Coordinator*), que s'encarrega de rebre les dades de la resta de dispositius (que recolliran i transmetran els valors dels paràmetres pels que han estat programats) i transmetre-les al PC de l'usuari.

En la il·lustració següent, podem veure les topologies de xarxa més comuns utilitzades en l'estàndard IEEE 802.15.4. La topologia d'estrella permet que cada sensor funcioni de manera independent, comunicant-se únicament amb el node central que concentra la informació que és mostrada en l'aplicació d'usuari.



Il·lustració 1: Estàndard IEEE 802.15.4, topologies d'estrella, arbre i malla³

³ http://www.icpdas.com/products/GSM_GPRS/zigbee/images/zigbee_topology.jpg

Com a dispositiu autònom, cada node es compon d'un microcontrolador, d'una bateria, d'un mòdul de comunicacions i d'un o més sensors. Per tal de garantir un correcte funcionament del conjunt i la monitorització contínua de l'entorn a observar, cada part haurà de ser supervisada, de manera que es generi una alarma de sistema en cas de detectar anomalies en el sistema.

Tant la topologia, com la independència dels nodes, ens han permès desenvolupar i articular la xarxa de sensors com un sistema distribuït al voltant de l'estàndard IEEE.802.15.4, donant al sistema flexibilitat i escalabilitat, i afegint la possibilitat d'incorporar, sempre que les diferents implementacions dels fabricants siguin compatibles, un gran nombre de nodes heterogenis. Per exemple, aquesta arquitectura permetria que el node principal estès basat en un *dongle* USB compatible amb 802.15.4⁴ i sensors addicionals de tipus *Netduino*⁵ o *Arduino*⁶ amb un *shield Arduino XBee 802.15.4 OnChip*⁷ amb detectors addicionals de CO⁸, fum⁹, etc.

A més, en el nostre cas, la xarxa de sensors està supervisada i personalitzada mitjançant una interfície gràfica d'usuari que permet tant respondre a les alertes recollides pels nodes remots en temps real com configurar els paràmetres generals i particulars de funcionament del sistema.

Objectius

Amb la realització d'aquest treball de fi de carrera es pretenia adquirir les capacitats necessàries per tal de crear, manipular i analitzar una xarxa de sensors sense fils (WSN), concretament un sistema de detecció i d'alarma d'incendis, i assolir els següents objectius generals i específics:

⁴ <http://www.advanticsys.com/shop/asud1000-p-25.html?language=es>

⁵ <http://www.netduino.com>

⁶ <http://www.arduino.cc/es/>

⁷ Aquesta és la versió compatible amb el nostre sistema, els mòduls de les Series 2 i superiors de *XBee* compleixen amb l'especificació ZigBee enlloc de amb 802.15.4.

⁸ <http://english.hwsensor.com/ewebeditor/uploadfile/20110124131043217.pdf>

⁹ http://www.sharp-world.com/products/device/lineup/selection/pdf/opto201201_e.pdf
http://www.sparkfun.com/datasheets/Sensors/gp2y1010au_e.pdf

Objectius generals

- Saber plantejar i resoldre un projecte basat en una xarxa de sensors sense fils.
- Comprendre els conceptes tecnològics que intervenen en una xarxa de sensors sense fils, i en el seu desplegament.
- Implementar un sistema de sensors que ens permetin assolir els objectius especificats en el document de requeriments.
- Conèixer l'estructura de diferents tipus de sistemes operatius i llenguatges de programació utilitzats en els sistemes encastats (*TinyOS*¹⁰, *nesC*¹¹, etc.).
- Aprendre les habilitats necessàries per treballar en un sistema sense fils de manera que es tregui el màxim profit pel que es refereix a mobilitat i flexibilitat.
- Conèixer el funcionament dels diferents tipus de sensors.
- Controlar els sensors de manera remota a través de comunicacions sense fils.
- Conèixer de manera pràctica el funcionament de diferents tipus de microcontroladors.
- Entendre i saber interpretar les dades recollides pels sensors, així com presentar-les de manera intel·ligible als usuaris del sistema.

Objectius específics

- Detectar el risc d'incendi amb la màxima fiabilitat possible basant-nos en les dades recollides pels sensors i amb els dispositius disponibles.
- Dissenyar un sistema escalable.
- Dissenyar una interfície d'usuari gràfica senzilla i útil que permeti interpretar les dades de manera ràpida.
- Implementar els mecanismes de control necessaris per tal de garantir la comunicació amb els nodes.

¹⁰ <http://www.TinyOS.net/>

¹¹ <http://nescc.sourceforge.net/>

- Notificar als usuaris de manera inequívoca de les situacions de risc, tant en els sensors com en la consola de control.
- Implementar un sistema de diagnòstic que permetin conèixer l'estat de la xarxa i dels seus components.

Abast

Es considera dintre de l'abast d'aquest projecte:

- Estudi de sistemes encastats i de la seva metodologia de disseny.
- Entendre diferents aspectes dels sistemes distribuïts i de les xarxes de sensors sense fils, així com dels protocols de comunicació emprats.
- Estudiar protocols per a la transmissió de gran quantitat de dades i el procés de coordinació d'una xarxa.
- Analitzar els requeriments d'una aplicació que necessita resposta en temps real.
- Estudi de diferents sistemes d'alta disponibilitat, així com dels mecanismes necessaris per tal de garantir-la (*WatchDog*, etc.).
- Instal·lació i configuració del programari necessari per desenvolupar el projecte.
- Estudi dels mecanismes de programació propis de les motes i de sistemes operatius particulars.
- Utilitzar diferents fonts d'informació com a referència.
- Recopilació, cerca i tractament d'informació sobre els diferents sistemes de detecció, mesura i processament de les alertes.
- Creuar informació generada pels diversos mètodes i saber analitzar les dades rebudes dels sensors per tal de fer un diagnòstic acurat.
- Analitzar, categoritzar i tipificar els elements i missatges rebuts per tal de que siguin comprensibles per l'usuari.
- Redacció de la memòria del projecte.
- Realització de la presentació virtual.
- Respondre a les preguntes del tribunal sobre el projecte.

Enfocament

En l'enfocament que hem fet servir per a desenvolupar el projecte, podem distingir dues etapes. Una primera, basada en el concepte més clàssic de la reenginyeria de processos i una segona més propera a les tecnologies àgils.

La primera fase es correspon amb la planificació i la definició del pla de treball, on es van definir les funcionalitats bàsiques, es van fixar les tasques principals i les fites internes i externes del projecte. Les fites internes venien marcades, com veurem en l'apartat següent, per les diferents dates de lliurament, mentre que les externes venien donades per l'inici i fi del quadrimestre.

Pel desenvolupament dels objectius principals, es va seguir el mètode de cicle de vida en cascada¹², ja que les característiques del projecte s'adaptaven perfectament. En primer lloc, els requeriments estaven ben definits, així que era possible tenir una idea força aproximada de com seria el producte final des d'un principi. També coneixíem els elements que havien de formar part del sistema, el que implicava que el disseny també venia condicionat per la plataforma i pel hardware disponible. A més, un cop lliurat, no es preveien modificacions en el codi.

Per tal de poder controlar l'evolució del projecte, cada tasca es va dividir en subtasques amb un abast més reduït, acompanyades del corresponent pla de proves, seguint el model iteratiu per tal de solucionar els problemes que anéssim trobant. Aquest plantejament ens va permetre tenir una visió acurada del progrés del projecte així com una detecció de riscos més detallada.

La segona part del desenvolupament va anat evolucionant a mida que progressava el coneixement de la plataforma i de l'entorn de treball. D'aquesta manera es va decidir mantenir els objectius fixos definits en el document de requeriments però modificar els objectius addicionals. Per exemple, en la planificació inicial es va plantejar la possibilitat d'emmagatzemar les dades rebudes en una base de dades, tasca que finalment es va substituir per afegir nous sensors a les motes a través del port d'expansió.

¹² Sommerville, I., (2005). *Ingeniería del software*. Madrid: Pearson Addison-Wesley, p.62.

Planificació

Dins de l'apartat d'organització del projecte trobarem el calendari del projecte, les fites més rellevants i d'identificació de riscos.

Calendari general de projecte

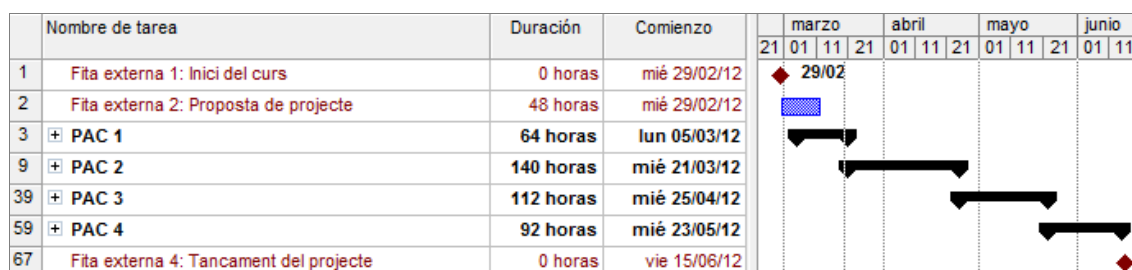
El projecte es va estructurar en quatre fases o PACs que es corresponien amb grups d'activitats:

PAC 1	Elaboració del pla de treball del projecte
PAC 2	Preparació de l'entorn de treball i primera versió del codi
PAC 3	Versió definitiva del codi
PAC 4	Revisió de la memòria del projecte i de la presentació virtual

El total d'hores dedicades al projecte va ser de 408 (sense tenir en compte la proposta). Per tal de fer aquest càlcul, es va fer servir el programari MS Project, amb un calendari de treball de 4 hores de dedicació diària i de 7 dies a la setmana (un total de 102 dies). Es van especificar com a festius, per motius personals, els dies:

- 22 de març
- 10 de maig

En les il·lustracions següents, podem veure el diagrama de Gantt de tot el projecte així com els diagrames detallats de cada grup d'activitats. El diagrama de Gantt inicial del calendari general del projecte era:



Il·lustració 2: Calendari general inicial del projecte

Finalment, la previsió inicial s'ha mantingut, tot i que s'ha modificat la distribució de les tasques, així com les dates de lliurament de les fites marcades per les PAC 2 i 3, corresponents amb els lliuraments del codi. El termini pel primer lliurament es va desplaçar del 24 al 26 d'abril, i el segon lliurament del 22 al 26 de maig. Aquestes modificacions les podem veure en la Il·lustració 3:



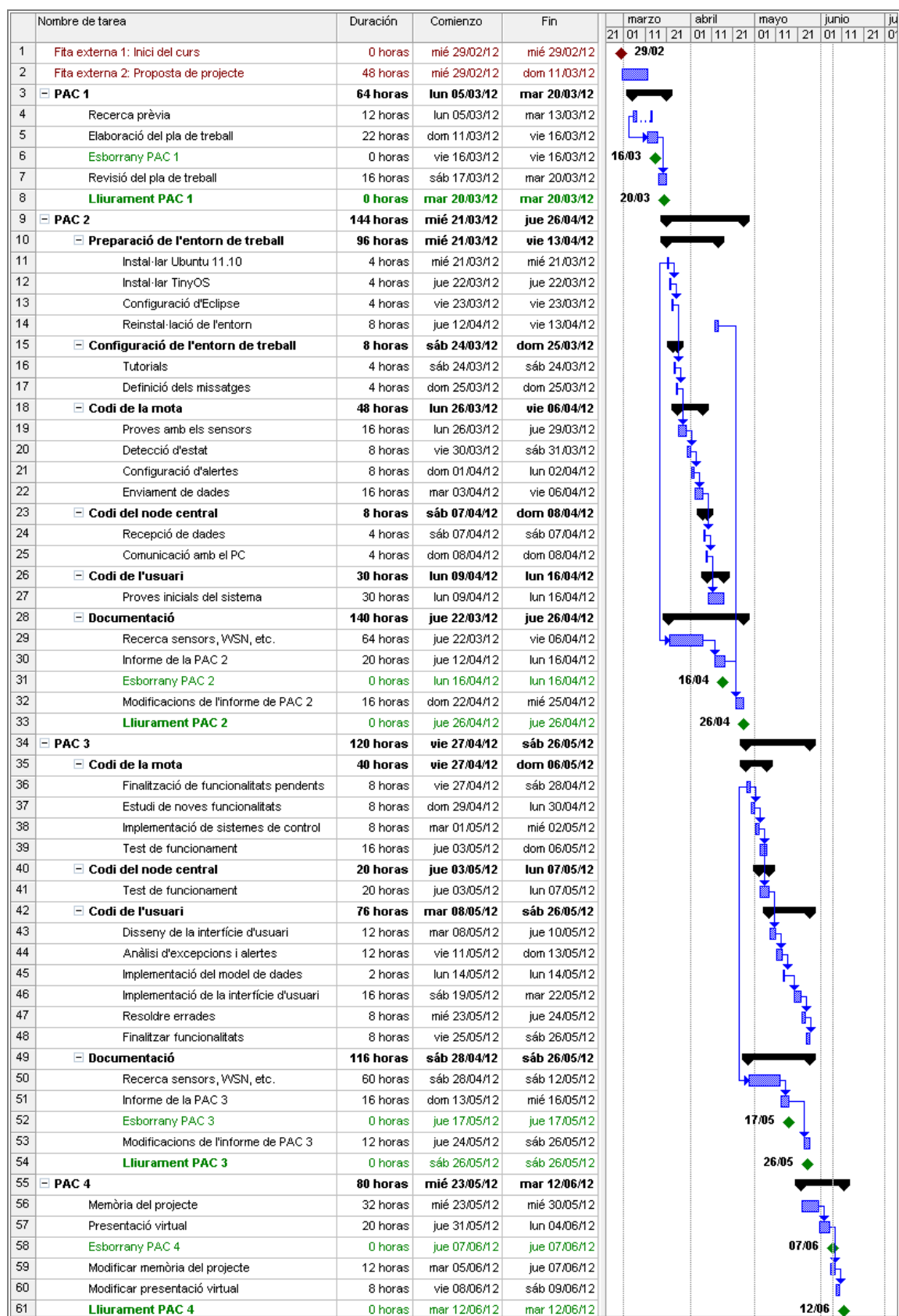
Il·lustració 3: Calendari general actualitzat del projecte

Les fites principals del projecte, ordenades per data, van quedar com podem veure en aquesta taula:

Data	Descripció
29 de febrer de 2012	Fita externa 1: Inici del curs
11 de març de 2012	Fita externa 2: Proposta del projecte
20 de març de 2012	Lliurament PAC 1 (pla de treball)
26 d'abril de 2012	Lliurament PAC 2 (primera versió del codi)
26 de maig de 2012	Lliurament PAC 3 (versió definitiva del codi)
12 de juny de 2012	Lliurament de la memòria del projecte
14 de juny de 2012	Lliurament de la presentació virtual
15 de juny de 2012	Fita externa 3: Tancament del projecte

Com ja hem comentat en apartats anteriors, hem considerat com fites externes al projecte, l'inici i final de curs i la proposta, ja que no influeixen directament en el desenvolupament del mateix.

En la pàgina següent podem trobar el diagrama de Gantt complet del projecte.



II-lustració 4: Diagrama de Gantt del projecte

Calendari de la PAC 1

La PAC 1, que comprenia la recerca prèvia d'informació i la redacció del pla de treball, tenia com objectiu principal situar-nos en el projecte. Un cop llegit el document de requeriments, es va fer una primera recerca d'informació per tal de poder fer una estimació aproximada de quin havia de ser el pla de treball. Era necessari trobar informació bàsica sobre conceptes com *WSN*, topologia, microprocessadors, etc. També va ser interessant fer una petita recerca sobre sistemes alternatius, estat actual, etc. La temporització d'aquesta primera activitat va quedar com es pot veure en la II-lustració 5:

	Nombre de tarea	Duración	Comienzo	febrero			marzo			abril	
				01	11	21	01	11	21	01	11
3	PAC 1	64 horas	lun 05/03/12								
4	Recerca prèvia	12 horas	lun 05/03/12								
5	Elaboració del pla de treball	22 horas	dom 11/03/12								
6	Esborrany PAC 1	0 horas	vie 16/03/12								
7	Revisió del pla de treball	16 horas	sáb 17/03/12								
8	Lliurament PAC 1	0 horas	mar 20/03/12								

II-lustració 5: Calendari de la PAC 1

En el pla de treball es va adjuntar una planificació basada en estimacions sobre les diferents tasques. Com veurem en els apartats següents, aquestes estimacions van sofrir modificacions per diversos motius.

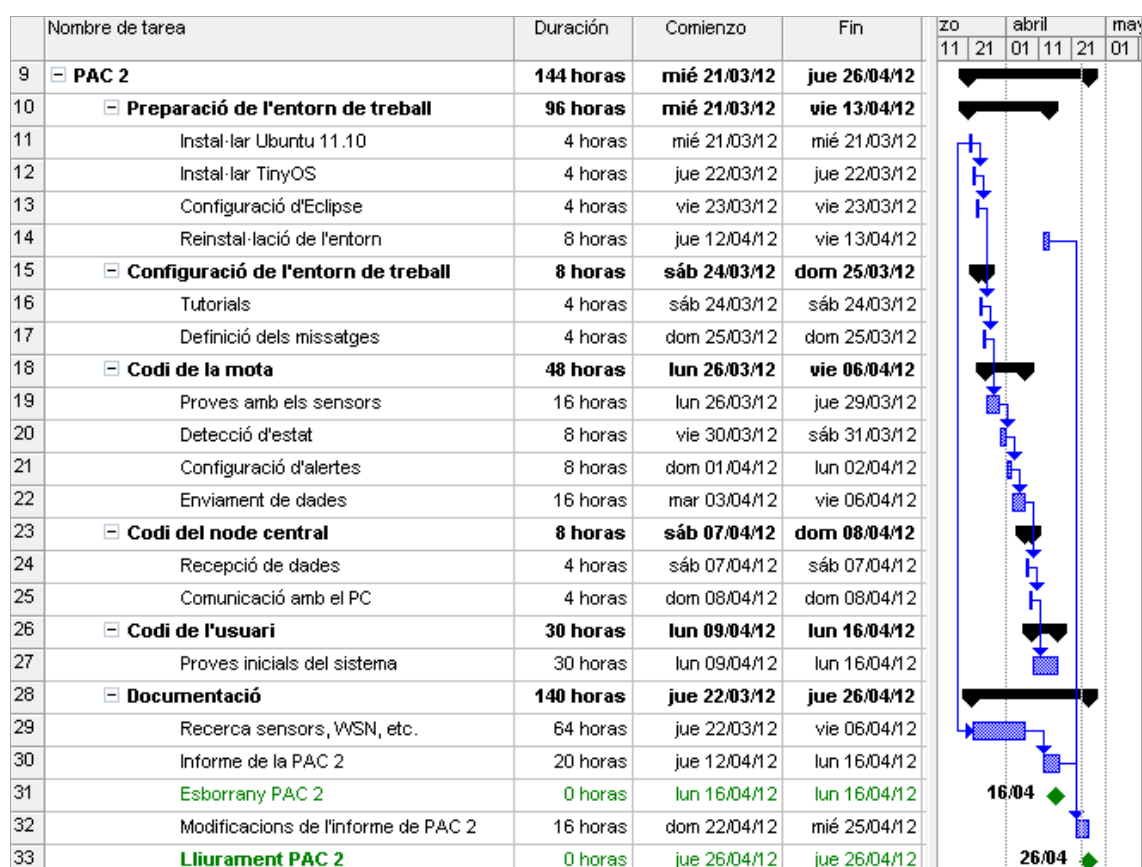
Calendari de la PAC 2

La segona PAC tenia com fita final el lliurament d'una primera versió del codi del sistema. En la relació d'activitats estava inclosa la instal·lació de l'entorn de desenvolupament, tant del sistema operatiu (*Ubuntu 11.10*) com d'*Eclipse* i dels *plugins* necessaris (*Yeti*), així com la compilació i execució del codi dels *tutorials* per tal de verificar que l'entorn estava correctament configurat. Es va incloure un esbós dels components de l'aplicació, que es van separar en consola d'usuari (*FireClient*), mòdul de comunicacions (*FireBase*) i sensors remots (*FireRemote*).

Respecte a la planificació inclosa en el pla de treball, es van produir alguns canvis significatius:

- Va haver un increment d'hores dedicades al codi de les motes i una disminució de la dedicació al codi de l'aplicació d'usuari, que va passar al següent lliurament.
- Les hores de test es van incrementar.
- Es va descartar el desenvolupament del model de dades i la implementació de l'emmagatzemament de les dades rebudes en una base de dades a canvi de dedicar més temps a recerca sobre la possibilitat d'expandir les motes afegint nous sensors a la plataforma actual.
- Es va eliminar de l'apartat de documentació la confecció de la memòria de projecte, deixant aquest punt per més endavant.
- Va ser necessari reinstal·lar el sistema, per un funcionament erroni de la plataforma.
- El lliurament de la PAC 2 es va ajornar, passant del dimarts 24 al dijous 26 d'abril.

En la imatge següent podem veure el diagrama de Gantt un cop ajustat:



II-lustració 6: Planificació ajustada de la PAC2

Si comparem el temps previst i el dedicat realment, per grups de tasques:

Grup de tasques	Previst	Real
Preparació de l'entorn de treball	12	20
Configuració de l'entorn de treball	12	8
Codi de la mota	20	48
Codi del node base	4	8
Codi de l'usuari	50	30
Documentació ¹³	108	140

Calendari de la PAC 3

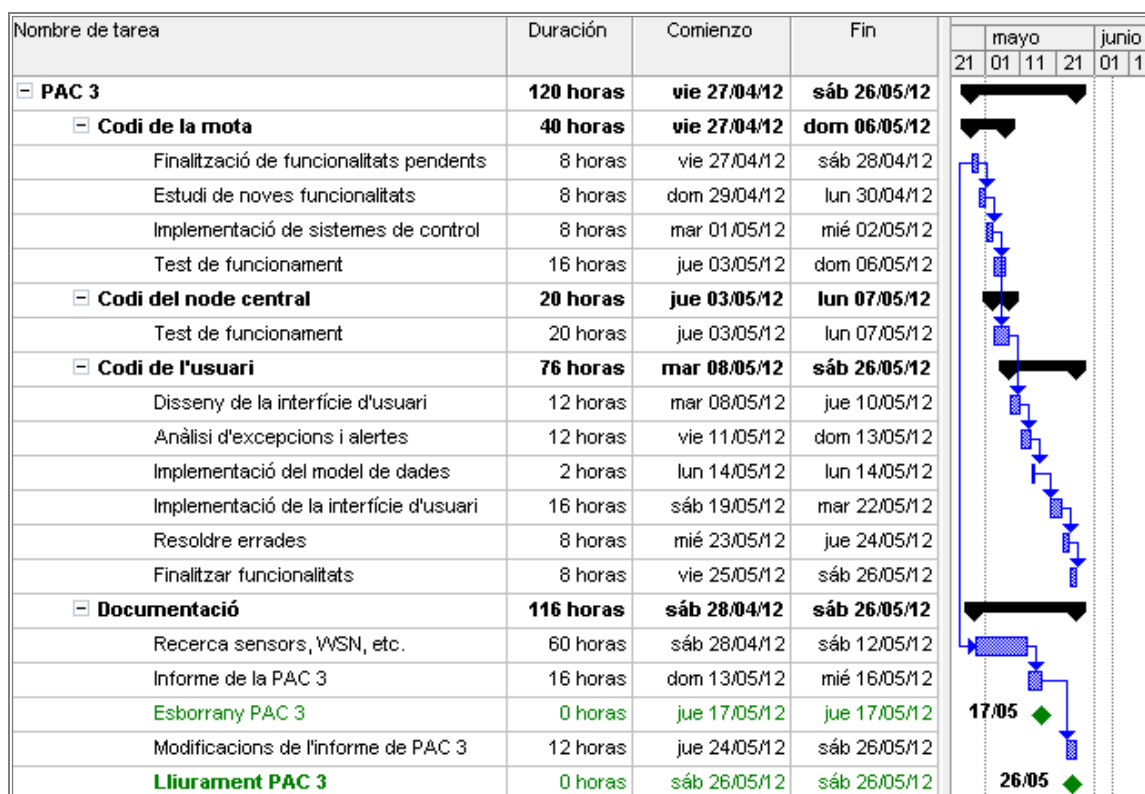
La PAC 3 heretava les modificacions que s'havien fet en el lliurament anterior. Aquest fet es reflectia en un increment important en les hores dedicades a la interfície de l'usuari. Comparant el temps previst i el dedicat, per grups de tasques:

Grup de tasques	Previst	Real
Codi dels nodes remots	48	40
Codi del node base	20	20
Codi de l'aplicació d'usuari	28	76
Documentació	108	116

En el diagrama de la pàgina següent podem veure les dates ajustades, on també es pot observar un increment en el temps de proves del node central així com en el desenvolupament de les funcionalitats addicionals. A més, es va ampliar el termini del lliurament en quatre dies.

El canvi en la data de lliurament, del 22 al 26 de maig, ens va permetre finalitzar algunes de les proves iniciades amb el mòdul d'expansió i els sensors addicionals.

¹³ En l'apartat de documentació s'inclou la recerca sobre l'entorn i els llenguatges de programació necessaris per dur a terme les tasques del lliurament.



Il·lustració 7: Diagrama de Gantt actualitzat de la PAC 3

Calendari de la PAC 4

El calendari de la PAC 4 va sofrir modificacions, ja que incloïa la redacció final de la memòria del projecte i la presentació virtual.

Com veiem al diagrama següent, un cop lliurada la presentació, es va donar per tancat el projecte a l'espera de les preguntes del tribunal d'avaluació.



Il·lustració 8: Calendari de la PAC 4

Descripció de riscos

En l'àmbit del projecte, varem considerat com a un risc, qualsevol eventualitat que pogués interferir el correcte desenvolupament del projecte. Els principals riscos detectats van ser els que es relacionen a continuació:

Risc	Problemes de disponibilitat
-------------	-----------------------------

Descripció	Problemes per trobar temps per desenvolupar les diferents activitats del projecte, ja sigui per càrrega de feina o per malaltia.
-------------------	--

Impacte	Durada de les tasques.
----------------	------------------------

Probabilitat	Mitja.
---------------------	--------

Acció	Recuperar el temps, si és possible, en dates posteriors.
--------------	--

Risc	Avaries de l'equipament informàtic
-------------	------------------------------------

Descripció	Avaria en el maquinari de la plataforma de desenvolupament.
-------------------	---

Impacte	Durada de les tasques.
----------------	------------------------

Probabilitat	Baixa.
---------------------	--------

Acció	Reparar l'avaria o, en cas de llarga durada, reinstal·lar el programari a un altre equip. Es pot realitzar una acció preventiva que consistiria en tenir un altre equip de recanvi.
--------------	---

Cal establir un sistema de còpies de seguretat per tal d'evitar la pèrdua d'informació.

Risc	Avaries en els sensors
-------------	------------------------

Descripció	Els dispositius són delicats i una manipulació incorrecta pot fer malbé el seu funcionament.
-------------------	--

Impacte	Impossibilitat d'obtenir els resultats requerits.
----------------	---

Probabilitat	Mitja.
---------------------	--------

Acció	Reduir l'àmbit de l'estudi o substituir els dispositius espatllats per uns altres.
--------------	--

Risc	Desconeixement de la metodologia en projectes <i>WSN</i>
Descripció	Errors en la planificació de les tasques per desconeixement de l'abast real de un projecte <i>WSN</i> .
Impacte	Durada de les tasques.
Probabilitat	Mitja.
Acció	Dedicar més temps per finalitzar les tasques.

Risc	Desconeixement de l'entorn de desenvolupament
Descripció	Desconeixement de l'entorn de desenvolupament, més pràctica en entorns Windows, encara que amb coneixements del llenguatge C.
Impacte	Desenvolupament feixuc i poc operatiu, que endarrereix la resta de tasques.
Probabilitat	Mitja.
Acció	Major dedicació.

Risc	Manca de components compatibles amb el sistema
Descripció	En cas de desenvolupar funcionalitats addicionals, dificultat per trobat components electrònics compatibles amb el sistema. Darrerament, l'oferta de mòduls d'ampliació basats en AVR està sent reemplaçada per d'altres orientats a plataformes basades en <i>Arduino</i> . Una diferència important és l'alimentació dels circuits 3,3V d'AVR per 5V d' <i>Arduino</i> .
Impacte	Limitació a l'hora d'ampliar el sistema.
Probabilitat	Baixa.
Acció	Cercar els components adients, es tracta d'un sistema de codi obert i components estàndards, com ara el microprocessador <i>Atmega1281</i> .

Risc	Errors en la interpretació de les dades rebudes
Descripció	Conclusions errònies donades per errors en les lectures dels sensors o en la seva interpretació.
Impacte	Impossibilitat d'arribar a conclusions encertades i a un diagnòstic acurat del risc.
Probabilitat	Mitja.
Acció	Comparar amb altres resultats, establir plans de proves fiables.

Recursos emprats

En aquest apartat detallem els recursos de maquinari i programari que han estat utilitzats en el desenvolupament del projecte:

Recursos de Maquinari

- Dues motes COU 1_2 24 A2 amb l'integrat *ZigBit™ 2.4 GHz Wireless* (ATZB-24-A2/B0)¹⁴ d'*Atmel Corporation*, que incorporen:
 - Microprocessador Atm1281 d'*Atmel Corporation*¹⁵
 - transceptor AT86RF230 d'*Atmel Corporation*¹⁶
 - Sensor de llum PDV-P9003-1 d'*Advanced Photonic, Inc.*¹⁷
 - Sensor efecte Hall BU52001GUL de *Rohm Semiconductor*¹⁸
 - Sensor de tensió d'alimentació (pont divisor de tensió)
 - Sensor de temperatura MCP9700 de *Microchip Technology Inc.*¹⁹

¹⁴ <http://www.atmel.com/Images/doc8226.pdf>

¹⁵ http://www.atmel.com/dyn/resources/prod_documents/doc2549.pdf

¹⁶ <http://www.atmel.com/Images/doc5131.pdf>

¹⁷ http://www.advancedphotonix.com/ap_products/pdfs/PDV-P9003-1.pdf

¹⁸ <http://www.rohm.com/products/databook/sensor/pdf/bu52001gul-e.pdf>

¹⁹ <http://ww1.microchip.com/downloads/en/DeviceDoc/21942e.pdf>

Comunicació sèrie (USART) pel port USB

- Port d'expansió per connectar els sensors addicionals, com ara:
 - Flame sensor de *DFRobotics*²⁰
 - Smoke sensor (MQ2) de *Seed Studio*²¹
 - Buzzers de *DFRobotics*²² i *Seed Studio*²³
 - Water sensor de *Seed Studio*²⁴
 - Touch sensor de *Seed Studio*²⁵
- CPU AMD Athlon(tm) 64 X2 Dual Core Processor 4400+ 2,3 GHz
- 4 GB de memòria RAM
- Disc dur Western Digital® WD2500JS® 250 GB, ATA, 7200 rpm
- Targeta gràfica ATI® RADEON XPRESS 200 Series 128 MB DDR
- Monitor HP® L1950 19" 1280 x 1024
- Connexió a Internet amb ADSL 10 Mbps
- Disc extern LaCie de 160 GB per a còpies de seguretat

Recursos de Programari

El programari que s'han fet servir en el desenvolupament del projecte, la confecció de la memòria i el disseny de la presentació és el següent:

- Microsoft® Windows® 7 amb Service Pack 1.
- Sistema operatiu *Ubuntu* 11.10.
- *TinyOS* 2.1.1.

²⁰ http://www.dfrobot.com/index.php?route=product/product&path=36_58&product_id=195

²¹ [http://www.seeedstudio.com/wiki/Electronic_brick_-_Smoke_sensor\(MQ2\)](http://www.seeedstudio.com/wiki/Electronic_brick_-_Smoke_sensor(MQ2))

²² http://www.dfrobot.com/index.php?route=product/product&filter_name=buzzer&product_id=84

²³ [http://www.seeedstudio.com/wiki/Electronic_brick_-_Buzzer\(A/D\)](http://www.seeedstudio.com/wiki/Electronic_brick_-_Buzzer(A/D))

²⁴ http://www.seeedstudio.com/wiki/Electronic_Brick_%E2%80%93_Water_Sensor

²⁵ http://www.seeedstudio.com/wiki/Electronic_brick-Touch_Sensor_module

- Compilador *nesC* v1.3.3.
- *Eclipse Galileo* i *Yeti 2 - TinyOS 2 Plugin for Eclipse*, edició del codi.
- *meshprog* versió 0.1.2, programació de les motes.
- VMware Workstation 8
- Microsoft® Office Word® 2007. Redacció dels diferents documents del projecte: informes de les PACs, memòria, etc.
- Microsoft® Office Project® 2003. Diagrames de Gantt del pla de projecte.
- Microsoft® Office Visio® 2003. Diagrames, planificació, etc.
- Microsoft® Office PowerPoint® 2007. Preparació de la presentació virtual.
- Microsoft® Office Excel® 2007. Càlculs i taules addicionals.
- Adobe® Photoshop® CS4. Edició d'imatges.
- Adobe® Premiere® CS4. Edició de vídeo.

Productes obtinguts

Una vegada finalitzat el projecte, s'han obtingut els productes següents:

- Un model o disseny d'un sistema conforme amb les especificacions inicials del document de proposta.
- El disseny d'un sistema basat en *WSN* capaç de detectar riscos d'incendi.
- Uns esquemes de connexió de les motes amb els circuits dels sensors addicionals (flama, fums, etc.).
- El codi font i els executables de les aplicacions dels sensors.
- El codi font i els executables del node de comunicacions.
- El codi font i els executables de l'aplicació d'usuari.
- La memòria del projecte.
- La presentació virtual del projecte.

Descripció dels altres capítols

En els apartats següents desenvoluparem els temes introduïts fins ara, veurem quines fites s'han assolit i que resta per fer per tal d'obtenir un producte comercialment viable.

En el capítol '*Antecedents*' veurem quina ha estat l'evolució tecnològica dels components que ens han permès dissenyar aquest projecte. També veurem quins són alguns dels usos que s'estan donant a les xarxes de sensors i com influeixen en la millora dels entorn on es fan servir.

En l'apartat de '*Descripció funcional*' explicarem com i a on s'ha desenvolupat cadascun dels requeriments especificats en el document inicial del projecte, com s'han estructurat els components i quina relació hi ha entre ells.

A continuació veurem, en la '*Descripció detallada*', com s'han assolit els objectius anteriors, i quins recursos tecnològics, tant de software com de hardware, s'han emprats per aconseguir-ho.

En '*Viabilitat tècnica*' hem fet una reflexió sobre l'estat actual dels productes obtinguts i sobre les modificacions que caldria fer per obtenir unes versions comercials a partir dels prototipus desenvolupats en el projecte.

En l'apartat dedicat a la '*Valoració econòmica*' hem mirat de valorar quin hauria de ser el cost d'una instal·lació d'aquest tipus, tenint en compte diferents opcions a l'hora de fer els pressupostos.

També hem dedicat un capítol, '*Conclusions*', per fer un resum dels objectius assolits i de les nostres impressions sobre el resultat i sobre el camí recorregut.

Finalment, tant en el '*Glossari*' com en la '*Bibliografia*' hem volgut deixar constància de les referències i termes emprats en la redacció dels diferents documents confeccionats durant el quadrimestre.

A més, en els annexos hem inclòs informació que hem cregut d'utilitat, com ara els problemes que hem anat trobat al llarg del projecte.

Antecedents

Un dels primers antecedents de les modernes xarxes de sensors el podem trobar en el *Sound Surveillance System (SOSUS)*²⁶. Es tractava d'una xarxa de boies instal·lada en la costa dels Estats Units durant la Guerra Freda amb la missió de detectar, mitjançant sensors de so (*hydrophones*), la presència de submarins enemics. Actualment, aquesta xarxa és utilitzada per la *National Oceanographic and Atmospheric Administration (NOAA)* per tal d'enregistrar incidents oceànics, com ara moviments sísmics o activitat animal²⁷.

També durant la Guerra Freda es va desenvolupar un sistema de control aeri amb l'objectiu de defensar els Estats Units i Canadà dels atacs aeris. Basat inicialment en radars, ha evolucionat per incloure aeròstats i avions de tipus *Airborne Warning and Control System (AWACS)* a mode de sensors, i és utilitzat a més en el control i detecció de tràfic de drogues.

A començament dels anys 80, en la *Defense Advanced Research Projects Agency (DARPA)* es va iniciar el projecte *Distributed Sensor Networks (DSN)* que és considerat el predecessor de les xarxes actuals. Els components tecnològics de DSN s'identificaven en un document de 1978²⁸ i ja s'inclouen sensors acústics i protocols de comunicacions que permetien compartir els resultats.

A finals dels anys 90, la companyia *Dust Inc., Berkeley, CA*, va iniciar en la Universitat de Califòrnia el projecte de recerca *Dust*²⁹. En agost de 2001 es va desplegar en aquesta mateixa universitat, com un experiment en una presentació, una xarxa de sensors de 800 nodes per tal de demostrar la potència i flexibilitat del sistema.

²⁶ <http://www.globalsecurity.org/intell/systems/sosus.htm>

²⁷ C. E. Nishimura and D. M. Conlon, "IUSS dual use: Monitoring whales and earthquakes using SOSUS" *Mar. Technol. Soc. J.*, vol. 27, no. 4, pp. 13–21, 1994.

²⁸ *Proceedings of the Distributed Sensor Nets Workshop*. Pittsburgh, PA: Dept. Comput. Sci., Carnegie Mellon Univ., 1978

²⁹ J. M. Kahn, R. H. Katz, and K. S. J. Pister, "Mobile networking for smart dust." in *Proc. ACM/IEEE Int. Conf. Mobile Computing and Networking (MobiCom)*, 1999, pp. 271–278.

Estat de l'art

Les xarxes sense fils són cada vegada més presents en tots els aspectes de les nostres tasques diàries. Les xarxes WSN són un tipus especials de xarxes sense fils que, a diferència de les xarxes convencionals, estan compostades de nodes o estacions amb recursos limitats. A més, les dades que es transporten es corresponen amb mesures de paràmetres físics, en lloc de dades, veu i serveis com en una LAN o WLAN tradicionals.

Ciutats intel·ligents

Un terme estretament relacionat amb les WSN és el de ciutat intel·ligent. Aquests tipus de projectes es fonamenten en la utilització de components estàndards de baix cost, on es pot identificar fàcilment una estructura pròpia de les *Wireless Sensor Networks*. En aquestes xarxes de transmissió de dades metropolitanes, són molt importants els nodes de recollida de valors ambientals o sensors que permeten als nodes centrals, amb la informació subministrada, prendre les decisions 'intel·ligents'.

Si fem cas de l'estudi "*Smart cities – Ranking of European medium-sized cities*"³⁰, les ciutats es poden definir i classificar com més o menys intel·ligents basant-se en sis paràmetres principals: economia, mobilitat, entorn, població, estil de vida i governament. Un resum, molt encertat, el podem trobar en següent cita de "*Smart cities in Europe*"³¹:

"Podem considerar una ciutat com 'intel·ligent' quan les inversions en capital humà i social, i en infraestructures de comunicació tradicionals (transport) i modernes (ICT), fomenten un desenvolupament econòmic sostenible i una elevada qualitat de vida, amb una sàvia gestió dels recursos naturals, a través d'un govern participatiu."

³⁰ Giffinger, R., & Pichler-Milanović, N. (2007). *Smart cities: Ranking of European medium-sized cities*. Vienna: Centre of Regional Science, Vienna University of Technology.

³¹ Caragliu, A., Del, B. C., & Nijkamp, P. (2009). *Smart cities in Europe*. Amsterdam: Vrije Universiteit, Faculty of Economics and Business Administration.

Així doncs, podem trobar algunes ciutats que ja destaquen, o que tenen en marxa projectes per tal de potenciar alguns dels indicadors esmentats anteriorment. Per exemple:

- Eficiència i gestió energètica: Sant Cugat, Málaga, Amsterdam, ...
- Entorns de negoci i economia del coneixement: Luxemburg, Dubai, Malta, Kochi, ...
- Transport i mobilitat urbana: Singapur, Brisbane, Estocolm, Maastricht, ...
- e-Govern i participació ciutadana: Tampere, Turku, Albuquerque, ...
- Medi ambient: Copenhaguen, Vancouver, Melbourne, Montpel·lier, ...
- Urbanisme: Masdar, Sondgo, ...
- Turisme i activitat cultural: París, Londres, Salzburg, Bruixes, Sidney, Zurich, ...
- Sanitat: París, Granada, ...

Molt sovint els projectes de ciutat intel·ligent venen de la mà d'un *partner* o soci tecnològic principal. Per exemple; en el cas de Sant Cugat³², l'ajuntament, juntament amb *Indra*³³, van presentar el seu model de *Smart City* amb l'objectiu de reduir la despesa pública a través de tecnologia sostenible.

El projecte de Sant Cugat està centrat en la reducció de les factures de l'ajuntament mitjançant una millor gestió dels recursos, oferint serveis de valor afegit. Ha tingut com a resultat la reducció de fins a un 20% la despesa d'aigua gràcies a la instal·lació de sensors que recullen informació d'humitat de sòl i capacitat de camp, i que, complementats amb una estació meteorològica i informació de les característiques de la vegetació, subministren al sistema central les dades necessàries per tal de determinar i enviar la dosi de reg justa, mínima i necessària a aplicar per un dia segons en les condicions meteorològiques concretes d'aquell dia.

Aquest projecte s'ha dotat d'un sistema de comunicacions via radio freqüència VHF, ubicat a la torre de comunicacions municipal de la Floresta, que possibilita l'enllaç per distribuir la senyal de control de reg intel·ligent, facilitant la comunicació directa d'anada i de retorn.

³² Sant Cugat, ciutat intel·ligent, <http://smartcity.santcugat.cat/>

³³ <http://www.indracompany.com/pais/espana>

En el cas de Màlaga, es tracta de Endesa³⁴ i IBM³⁵, amb l'objectiu d'assolir un estalvi energètic del 20% i una reducció en les emissions de CO₂ de 6000 tones, fomentant l'ús de les energies renovables³⁶.

En el projecte que es desenvolupa a Santander³⁷, *Internet of Things*, es fa servir el protocol 802.15.4 per tal de connectar els nodes o sensors *IoT*. D'aquesta manera, és possible monitoritzar amb un baix cost indrets i serveis, com els autobusos urbans, els aparcaments públics, les platges, la recollida i tractament d'escombraries, etc. Aquesta és probablement la xarxa de sensors sense fils més important del món. Fins a juny de 2012 s'havien instal·lat 1100 dispositius, 400 per detectar espais d'aparcament i 700 per a mesurar paràmetres ambientals, com soroll, CO (monòxid de carboni) i temperatura. Al finalitzar el desplegament, es preveu que s'hauran instal·lat més de 12000 sensors distribuïts per tota la ciutat³⁸.

En la ciutat d'Oulu (Finlàndia) es fa servir NFC (*Near Field Communication*) en diversos projectes³⁹: *smartParking*, bitllets d'autobús de l'operador *Koskijalat*, tiquets del teatre de la ciutat, informació sobre les obres de teatre, etc.

En nombroses ciutats d'Holanda, Suïssa, Dinamarca, Àustria o Alemanya es fan servir models PAYT (*Pay As You Throw*) en la recollida d'escombraries, de manera que els usuaris paguen en funció dels residus que generen. Per a això s'instal·len sensors RFID a les bosses o en els contenidors que poden ser llegits en remot o pels camions de recollida.

En el *Mobile World Congress* (MWC) de Barcelona, en el mes de febrer de 2012, Telefónica i els Bombers de la Generalitat de Catalunya van presentar *@textil*⁴⁰, un

³⁴ http://www.endesa.com/es/conoceendesa/lineasnegocio/principalesproyectos/Paginas/Malaga_SmartCity.aspx

³⁵ http://www.ibm.com/smarterplanet/es/es/smarter_cities/cities/index.html

³⁶ <http://www.smartcitymalaga.es/>

³⁷ <http://smartsantander.eu>

³⁸ <http://www.blogenergiasostenible.com/santander-ciudad-inteligente/>

³⁹ <http://ttuki.vtt.fi/smarttouch/www/?info=case>

⁴⁰ http://www.telefonica.com/es/descargas/mwc/np_2012_02_29_2.pdf

projecte que incorpora sensors en els vestits dels bombers que poden detectar fins a 9 tipus de gasos perillosos i notificar, mitjançant un protocol de tipus M2M (màquina a màquina) basat en *Arduino*⁴¹, tant al professional com a la central.

Valladolid i Palència, que presenten un ambiciós projecte comú, *Smart City Vyp*⁴², inclou dintre d'aquest, el concepte d'*Smart Grid*⁴³. En aquests tipus de xarxes elèctriques, es substitueixen els comptadors tradicionals per *Smart meters*⁴⁴, o comptadors intel·ligents, que ofereixen com avantatges la lectura remota del consum, la facturació sobre dades reals, les modificacions de tarifes en temps real, el suport 24x7, la detecció i resolució proactiva d'incidents, etc.

Xarxes d'abast personal (*Personal Area Network*)

Les xarxes de curs abast PAN (*Personal Area Network*) són especialment adients per desenvolupar WSN.

La tecnologia UWB (*ultra-wide-band*) es pot fer servir, en casos d'emergència, per a localitzar persones atrapades en lloc de difícil accés⁴⁵. Per les seves característiques, és possible utilitzar UWB en l'àmbit domèstic, per a fer *streaming* i *broadcast* de continguts MPEG-2 o MPEG-2-HD de vídeo a varies TV, per a interconnectar càmeres de fotos i vídeo, per reemplaçar la necessitat de cables USB o IEEE 1394 *FireWire*. *Bluetooth* (IEEE 802.15.1), *ZigBee* (IEEE 802.15.4), i *Ultra Wide Band* (UWB/IEEE 802.15.3a) es fan servir en comunicacions sense fils en l'àmbit de l'automobilisme.

*Z-Wave*⁴⁶ està orientat a automatitzar l'interior de les cases. Es tracta d'un sistema de radio dissenyat per *ZenSys AS*⁴⁷ i que opera en la banda dels 868.42 MHz en Europa i 908.42 MHz en U.S, pel que comparteix rang amb alguns sistemes de telefonia sense

⁴¹ <http://www.arduino.cc/>

⁴² <http://www.smartcity-vyp.com>

⁴³ <http://www.smartgrids.eu/>

⁴⁴ <http://www.metersandmore.com/>

⁴⁵ <http://crabbe-consulting.com/files/spie2006twr.pdf>

⁴⁶ <http://www.z-wave.com/modules/ZwaveStart/>

⁴⁷ <http://www.zen-sys.com>

fils i d'altres electrodomèstics, sense interferir en la banda de 2,4 GHz característica de les WLAN. El disseny és modular, de manera que pot créixer a mida que creixen les necessitats. Les seves característiques principals són baix cost, baix consum, fiabilitat, facilitat d'instal·lació, no necessita manteniment de la xarxa i pot interactuar amb altres dispositius que comparteixin l'especificació. Va ser dissenyat per un nombre de nodes entre 20 i 200, que es comuniquen entre ells a intervals de 5 a 15 minuts amb uns missatges de mida variable entre 4 i 6 bytes. Dins d'aquest mateix segment de mercat, es poden trobar altres productes de xarxes sense fils com *Insteon*⁴⁸, *ZigBee*⁴⁹, o d'altres del tipus *Powerline Networks* com ara *X10*⁵⁰, *UPB*⁵¹, *LonWorks*⁵², *HomePlug*⁵³, *Intellon*⁵⁴, o *CEBus*.

En la imatge següent podem veure una comparativa entre diferents protocols de xarxes utilitzades en PAN (*Personal Area Network*):

Standard	Bluetooth IEEE 802.15.1	ZigBee IEEE 802.15.4	UWB IEEE 802.15.3a	Wi-Fi IEEE 802.11a/b/g
Freq. band	• 2.4 Ghz & 2.5 Ghz (ver 1.2)	• 2.4 Ghz	• 3.1-10.6 Ghz	• 2.4 Ghz (b/g) & 5 Ghz (a)
Network	• P2P	• Mesh	• P2P	• P2P
Modulation technique	• Frequency Hopping Spread Spectrum (FHSS)	• Direct Sequence Spread Spectrum (DSSS)	• Orthogonal Frequency Division Multiplexing (OFDM) or Direct-Sequence UWB (DS-UWB)	• OFDM or DSSS with Complementary Code Keying (CCK)
Maximum network speed	• 1 Mbps (ver 1.0) • 3 Mbps (ver 1.2) • 12 Mbps (ver 2.0)	• 250 Kbps	• 50-100 Mbps (480 Mbps within short ranges expected).	• 54 Mbps (802.11a) • 11 Mbps (802.11b) • 54 Mbps (802.11g)
Network range	• Up to 100 meters, depending on radio class (effective 10 meters).	• Up to 70 meters (effective 20 meters).	• Up to 20 meters (effective 10 meters).	• Up to 100 meters (effective 50 meters).
Main usage	• Voice applications. • Eliminating short-distance cabling.	• Sensors/control applications. • Grand-scale automation. • Remote control.	• Multimedia applications. • Healthcare applications.	• Office and home networks. • WLAN. • Replace Ethernet cables.
Strong points	• Dominating PAN tech. • In vehicles today. • Easy synchronization of mobile devices. • Frequency hopping tolerant to harsh environments.	• Static network. • Control/sensor. • Many devices/nodes. • Small data packets. • Low duty cycle. • Low power.	• Easy and cheap to build. • Consume very little power. • Provides high bandwidth. • Broad spectrum of frequencies (robustness).	• Dominating WLAN tech. • Know-how.
Weak points	• Interference with WiFi. • Consume medium power.	• Low bandwidth.	• Short range. • Interference.	• Traditionally consume high power.
Automotive usage (potential)	• Portable devices. • Diagnostics tools. • Real-time communications. • Device connectivity.	• In-vehicle communications. • Mobile/static sensor networks.	• Robust vehicle communications. • High bandwidth communications.	• Inter-vehicle communications. • Vehicle-to-vehicle. • Vehicle-to-roadside.

Taula comparativa entre diferents protocols emprats en xarxes sense fils⁵⁵:

⁴⁸ <http://www.insteon.net/pdf/insteoncompared.pdf>

⁴⁹ ZigBee Alliance (www.zigbee.org)

⁵⁰ <http://www.x10.com>

⁵¹ Universal Powerline Bus, dissenyat per PCS (Powerline Control Systems, Inc., www.pcslighting.com)

⁵² Desenvolupat en 1988 per Echelon, Inc., (www.echelon.com)

⁵³ HomePlug Powerline Alliance (www.homeplug.org)

⁵⁴ <http://www.intellon.com>

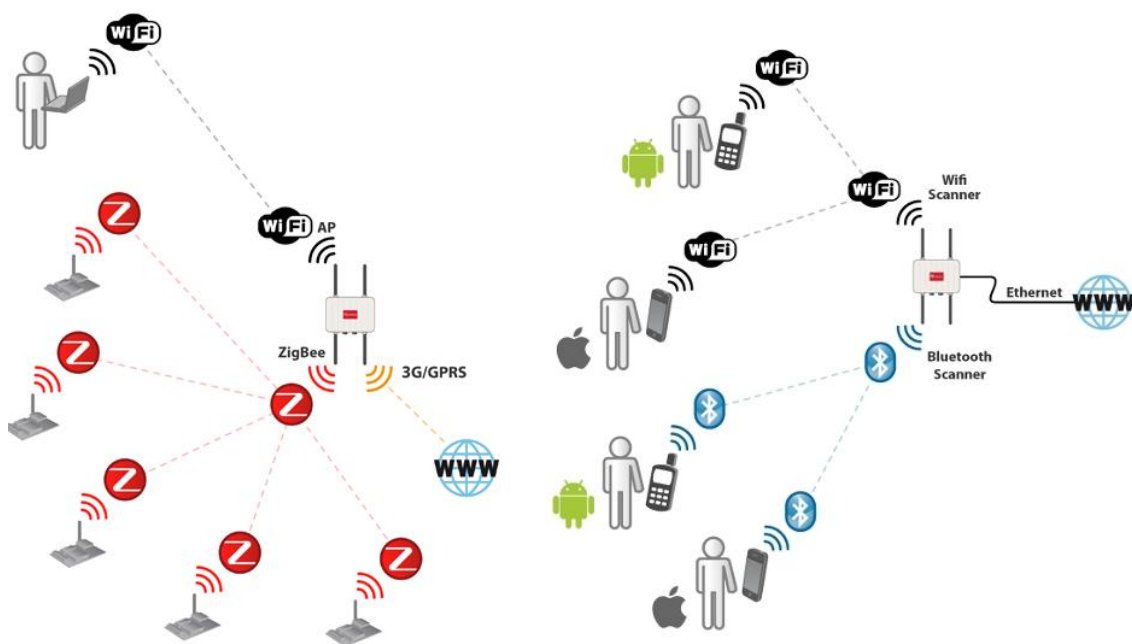
⁵⁵ http://ant.comm.ccu.edu.tw/course/97 ITS/1_HW1/0.Wireless%20Automotive%20Communications.pdf

Estudi de mercat

Les WSN constitueixen un producte de mercat que ha tingut una ràpida expansió. Un dels motius que creiem que ha influït directament en aquest fet és que, per les seves característiques, el desplegament es pot dur a terme de manera graduals, repartint la inversió necessària en diferents períodes temporals.

Recordem que es tracta de sistemes flexibles i escalables, amb una alta resistència a fallades i amb un baix consum i cost unitari dels sensors. També existeix una quantitat considerable d'eines de desenvolupament amb llicència *Open Source*.

A més, cada vegada és més senzill integrar aquest tipus de xarxes en una xarxa corporativa d'empresa. Per exemple *Libelium* ofereixen productes com *Meshlium Xtreme*⁵⁶ que, com veiem en la il·lustració següent, tenen la capacitat d'actuar com a passarel·la entre diferents tipus de xarxa.



Il·lustració 9: Passarel·la entre diferents tipus de xarxa

Podem trobar un gran nombre d'empreses dedicades a la comercialització de solucions basades en WSN:

⁵⁶ <http://www.libelium.com/products/meshlium/models>

En Austràlia, *Csiro*⁵⁷, especialitzada en sensors exteriors aplicats a les granges de grans extensions i a la mesura de paràmetres mediambientals.

En Alemanya *Dresden Elektronik*⁵⁸ comercialitza dispositius basats en ZigBee® i aplicacions per *traffic engineering*.

En Suïssa *Shockfish SA* desenvolupa els dispositius *Tinynode*⁵⁹ amb un estret contacte amb l'*Ecole Polytechnique Fédérale de Lausanne (EPFL)*⁶⁰.

En Espanya *Dexma Sensors, SL*, que va iniciar les seves activitats a principis de 2007, amb el suport de la Universitat Politècnica de Catalunya (UPC), La Salle i El Govern català en l'àmbit de CIDEM|COPCA ACCIÓ.

També en Espanya *Libelium Comunicaciones Distribuidas S.L.*, que dissenya i fabrica hardware modular per a la implementació de WSN, i *Wireless Sensor Networks Valencia S.L.*⁶¹, especialitzada en el disseny, desenvolupament i implantació d'aplicacions basades en WSN i amb origen en l'Institut ITACA de la *Universidad Politécnica de Valencia*.

Per contra, en Estats Units *Crossbow*⁶², que desenvolupava solucions hardware (com per exemple les motes MICA2 (868/916 MHz), Micaz (2.4 GHz), IMOTE2, telos i telosb) i software per a WSN i que va col·laborar amb la Universitat de Berkeley en el desenvolupament de *TinyOS*, ha reorientat el negoci i ja no ofereix aquest tipus de productes. El mateix ha passat amb l'empresa *Moteiv Corporation*, que fundada per Joseph Polastre, antic doctorand d'un grup de treball de la Universitat de Berkeley i que va desenvolupar les motes *Tmote Sky* i *Tmote Invent*, ha canviat el nom per *Sentilla Corporation*⁶³ i ha focalitzat la seva activitat en virtualització, *private cloud*, etc.

⁵⁷ <http://www.sensornets.csiro.au>

⁵⁸ <http://www.dresden-elektronik.de>

⁵⁹ <http://www.tinynode.com>

⁶⁰ <http://www.epfl.ch/>

⁶¹ <http://www.wsnaval.com/>

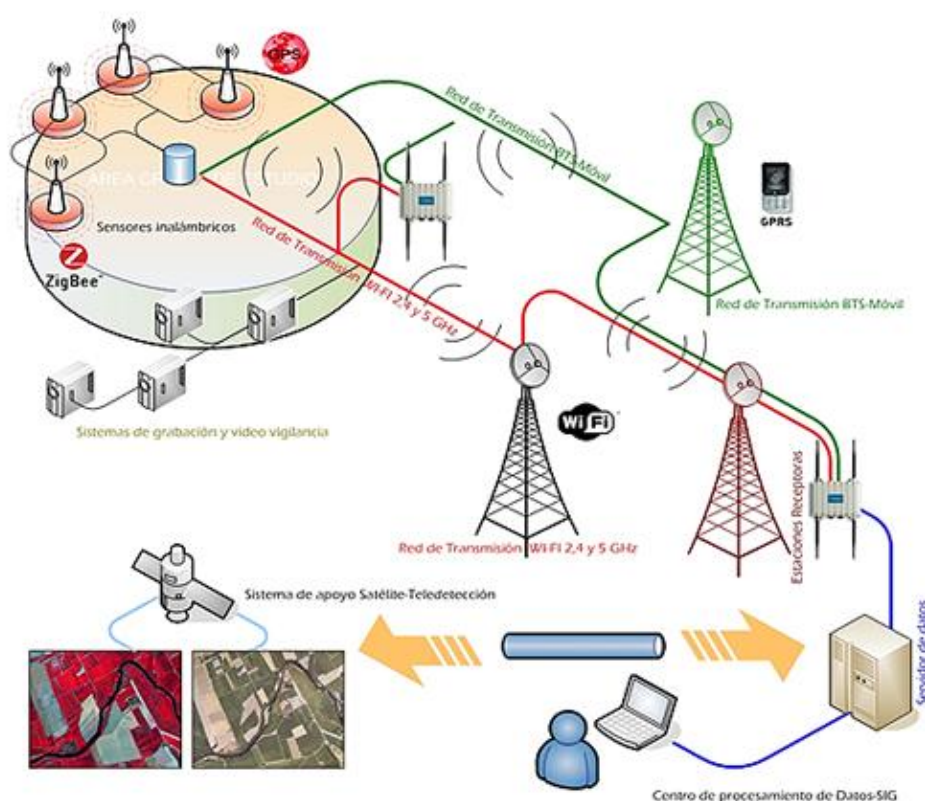
⁶² <http://www.xbow.com/>

⁶³ <http://www.sentilla.com/>

Sistemes de detecció d'incendis

Pel que fa a la detecció d'incendis, forestals en aquest cas, *DIMAP-FactorLink*⁶⁴, sota la denominació '*SISVIA Vigilancia y Seguimiento Ambiental*' ofereixen de manera conjunta productes per a la protecció del medi ambient. Concretament, han implementat un sistema de detecció d'incendis en una àrea de 210 hectàrees en les Comunitats Autònomes de Galícia i Astúries⁶⁵ cobertes amb 90 dispositius. Aquest projecte utilitza com a sensors la línia *WaspMote* de *Libelium*.

Aquesta empresa ofereix també com a opció una placa de gasos⁶⁶, que conté els sensors electrònics necessàries per a detectar una gran varietat d'emissions i que s'integra de manera senzilla amb les motes. En la imatge següent podem veure un diagrama general del sistema complet:



Il·lustració 10: Diagrama general del projecte SISVIA⁶⁷

⁶⁴ <http://www.dimap.es/index.html>

⁶⁵ http://www.libelium.com/wireless_sensor_networks_to_detec_forest_fires/

⁶⁶ <http://www.libelium.com/products/waspmote/sensors>

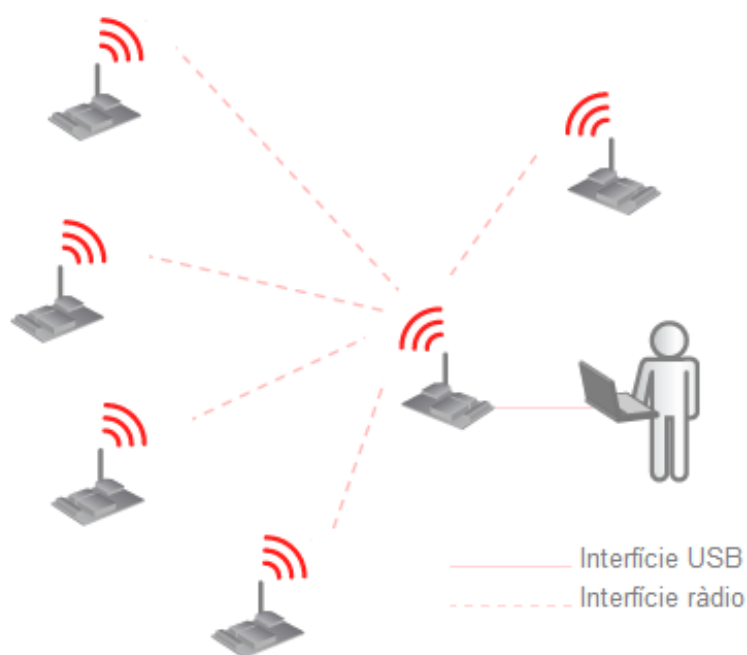
⁶⁷ http://www.libelium.com/libelium-images/waspmote_antiincendios/environmental_infra_big.png

Descripció funcional

Com ja hem comentat anteriorment, el disseny de la solució estava condicionat per l'arquitectura triada, que estructurava el nostre projecte com una aplicació distribuïda. Així doncs, el sistema es va dividir en tres blocs principals, diferenciant entre els mòduls corresponents als sensors, encarregat de recollir les lectures ambientals, les comunicacions, amb la funció principal de fer arribar les dades al centre de control, i la interfície d'usuari, que permetia analitzar i gestionar les dades rebudes.

Sistema total

Dins de la descripció del sistema total podem veure com les *motes* o sensors són la part més important del projecte, i estan basades en el mòdul *Atmel ATZB-24-A2*⁶⁸, *ZigBit*TM que té un factor de forma ultra compacte, i que es caracteritza per un baix consum i per un sistema de comunicacions d'alta sensibilitat que compleix amb els estàndards 2.4 GHz IEEE 802.15.4/*ZigBee*[®]⁶⁹. L'esquema general del sistema es pot veure en la Il·lustració 11:



Il·lustració 11 Estructura del sistema

⁶⁸ <http://www.atmel.com/Images/doc8226.pdf>

⁶⁹ <http://www.zigbee.org/>

Com que els nodes, a més de sensors tenen microcontroladors amb la capacitat de processar informació, varem poder triar si enviàvem les lectures del sensor de manera periòdica, per tal de mantenir un històric, o bé si enviàvem només les alarmes, comparant el llindar d'alerta en el mateix node. Seguint les indicacions del document de requeriments, optarem per fer les dues coses, de manera que les lectures periòdiques van servir com a confirmació de les alertes.

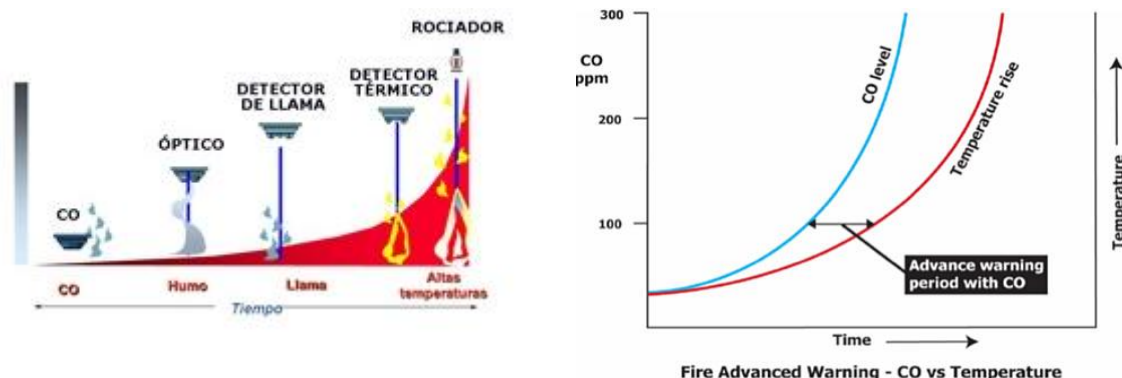
La tasca principal del mòdul central o de comunicacions era la d'actuar de mitjancer entre els sensor i l'usuari de l'aplicació.

Finalment, el mòdul d'usuari permet accedir de manera senzilla a les dades recollides pels sensors així com a la configuració del sistema.

Detecció d'incendis

La detecció d'una alarma d'incendi en aquest projecte s'ha basat en l'augment de la temperatura en l'entorn monitoritzat. Tot i així, s'han desenvolupat, com a funcionalitats addicionals, proves amb nous sensors connectats a les motes a través del port d'expansió, mirant d'obtenir un millor temps de resposta en la detecció del risc.

Per exemple, en els següents gràfics es mostren els avantatges de fer servir un detector de CO (monòxid de carboni) envers d'un sensor tèrmic.



Il·lustració 12: Fases en el desenvolupament d'un incendi⁷⁰

⁷⁰ <http://www.casmar.es/noticiasDetalle.asp?id=44&id2=1>

http://www.conspect-controls.com/images/insets_asme_coalgen-1.jpg

Com podem veure, en una fase inicial, malgrat que en el punt d'origen es poden generar flames amb una temperatura de més de 500°C, la temperatura de la resta del recinte només estarà lleugerament per sobre de la temperatura normal.

En el nostre sistema, quan un dels sensors detecta que el valor llegit supera el llindar establert, configurat mitjançant l'aplicació d'usuari, s'envia una alerta de manera automàtica al centre de control a través de la xarxa sense fils.

Al mateix temps, es notifica de manera visual i acústica⁷¹, en el node local. Per tal de validar les alertes, es recull el valor de la temperatura dels nodes a intervals regulars definits per l'usuari. Els diferents nodes també contemplen l'activació manual del sistema d'alarma mitjançant un botó o polsador d'emergència simulat amb el botó USR de les motes.

L'aplicació d'usuari mostra el tipus d'alarma, manual o automàtica, i l'hora, sensor i temperatura detectada. Un cop rebuda la notificació, l'usuari té la possibilitat de confirmar la recepció de l'alarma i activar la senyalització en el dispositiu remot.

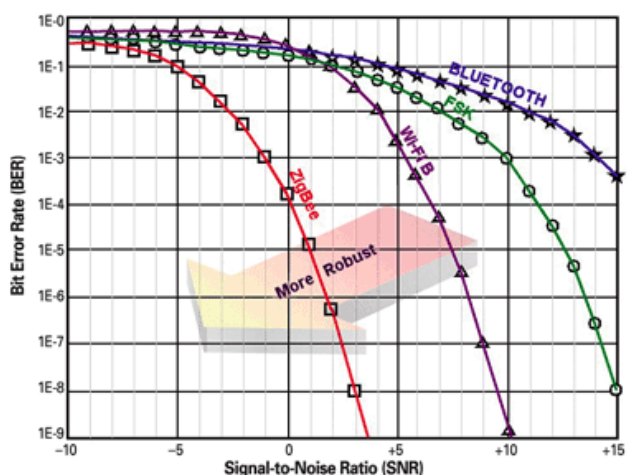
Fiabilitat en la transmissió

Com que és molt important que no es perdi cap alarma d'incendi, s'havia de garantir la recepció mitjançant confirmació. Per tal de comprovar si un node està actiu, es fa servir un *heartbeat*, o senyal de control, basat en la transmissió de l'estat de les bateries del node. Aquest mateix senyal de control s'ha fet servir per obtenir la potència del senyal de transmissió entre els nodes remots i el node base.

Com ja hem comentat anteriorment, les motes que hem utilitzat per a la detecció utilitzen com a mòdul de comunicacions l'integrat *ZigBit 2.4 GHz Wireless Module (ATmega1281/AT86RF230)*, que compleix amb les especificacions IEEE 802.15.4 i *ZigBee®*. D'aquesta manera, el mateix protocol garanteix la fiabilitat de les comunicacions amb un menor rati d'error en les comunicacions.

⁷¹ Actualment amb un *buzzer*, encara que una millor opció podria ser un dispositiu piezoelèctric amb un nivell de sortida més elevat.

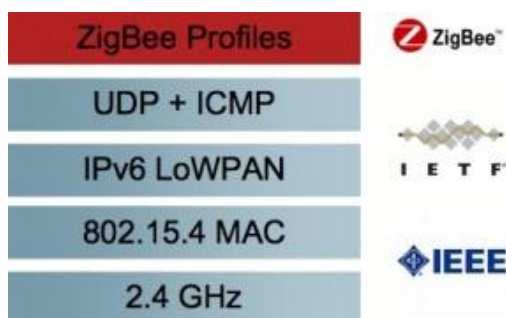
Tot i que el mòdul *ZigBit* dóna suport tant al protocol 802.15.4 com a *ZigBee*, les llibreries de *TinyOS* només treballen sobre 802.15.4, pel que ens hem centrat en aquest darrer. En la il·lustració següent podem veure una comparativa de la fiabilitat en la transmissió de dades de diferents protocols emprats per les xarxes sense fils.



Il·lustració 13: *ZigBee*® BER / SNR⁷²

ZigBee® fa servir la banda lliure de 2,4 GHz i està protegida contra interferències al basar-se en la norma IEEE 802.15.4. A més, la xarxa té la capacitat de crear-se i auto reparar-se en temps real. Pel que fa al xifrat, es fa servir AES (*Advanced Encryption Standard*) de 128 bits.

De manera esquemàtica, la relació entre IEEE 802.15.4 i *ZigBee*® es pot veure en el gràfic següent:



Il·lustració 14: IEEE 802.15.4 i *ZigBee*®⁷³

⁷² <http://www.rcgroups.com/forums/attachment.php?attachmentid=3257054>

Gestió dels nodes

Pel que fa als nodes, podem distingir entre els dispositius de captació de dades i l'encarregat de fer de pont amb l'equip de l'usuari.

En tots dos s'ha fet servir un temporitzador de tipus *WatchDog*⁷⁴ per tal de garantir que els nodes no es queden fora de servei i que, en cas de succeir, tinguin la capacitat de recuperar automàticament la configuració i tornar a ser operatius. El *WDT (Enhanced Watchdog Timer)* és implementat directament pel microprocessador en que es basen els dispositius, l'*Atmega1281*⁷⁵, i s'executa de manera independent a la resta del sistema, garantint la detecció de funcionaments erronis i el reinici automàtic del sistema en cas de ser necessari.

Nodes remots

Els nodes remots són els encarregats de recollir les dades ambientals que ens permetran determinar si s'ha produït una situació d'alarma. Envien periòdicament, amb uns intervals configurats per l'usuari, les lectures dels sensors, entre ells l'estat de les bateries i, en cas de detectar un estat crític, proper a l'esgotament, es genera una alarma. També es produeix una alarma, generada per l'aplicació d'usuari, si es perd la comunicació amb un node. Les seves funcionalitats principals són:

- Habilitar / deshabilitar els sensors mitjançant la detecció de pics magnètics pel sensor d'efecte *Hall*⁷⁶.
- Llegir els valors dels diferents sensors i modificar l'estat dels *leds* per notificar visualment als usuaris.
- Enviar les dades recollides per ràdio al node base.
- Enviar un *heartbeat* al node base amb l'estat de les bateries i, en cas de detectar un estat crític, proper a l'esgotament, generar una alarma.
- Rebre els paràmetres de configuració enviats pel node base i modificar el funcionament del node remot (interval de lectura, llindar, etc.).

⁷³ <http://zachshelby.files.wordpress.com/2009/02/zigbee-ip.jpg?w=288>

⁷⁴ <http://www.atmel.com/Images/doc2551.pdf>

⁷⁵ <http://www.atmel.com/devices/ATMEGA1281.aspx?tab=documents>

⁷⁶ http://dialnet.unirioja.es/servlet/fichero_articulo?codigo=3694136

Node base

El paper del node base és servir d'intermediari entre els nodes remots i l'aplicació d'usuari, recollint els missatges de les motes que arriben per ràdio i enviant-los al PC pel port USB i enviant les comandes rebudes del PC pel port USB a les motes per ràdio. En aquest node dispositiu no es fan tractament de la informació, simplement s'encaminen les dades cap al receptor adient.

També s'ha proveït un sistema per comprovar si existeix cobertura en la ubicació del sensor, a partir del valor de RSSI (*Receive Signal Strength Indication*) que hem inclòs en els missatges d'estat de la bateria. La potència rebuda es transforma en *dBm* per l'aplicació d'usuari.

Aplicació de l'usuari

L'aplicació d'usuari s'ha dissenyat de manera que sigui senzilla d'utilitzar però que permeti controlar i configurar el sistema. Es compon de la interfície gràfica que interactua amb l'usuari i de la lògica necessària per interpretar les dades rebudes dels sensors. Les funcionalitats principals són:

- Mostrar en pantalla els valors de les lectures recollides, el tipus d'alarma (manual o automàtica), l'hora i el sensor que ha generat les dades.
- Convertir els valors rebuts dels sensors en dades fàcilment interpretables per l'usuari: graus centígrads, lux, *dBm*, etc.
- Mostrar l'estat dels sensors: habilitat / deshabilitat, cobertura, etc.
- Definir els paràmetres de funcionament, intervals de mostreig, llindars d'alarma dels sensors, etc.
- Generar una alarma si es perd la comunicació amb algun dels nodes per un interval superior a l'especificat.
- Avisar de manera clara en cas d'alarma i donar la possibilitat de confirmar o descartar l'avís. Es considera que s'ha produït una alarma en cas de que les lectures estiguin per sobre del valor mínim establert per l'usuari.
- Proveir d'un sistema de prova que permeti validar que el sistema està funcionant dintre dels paràmetres considerat correctes.

Funcionalitats addicionals

Entre les funcionalitats addicionals desenvolupades podem destacar:

- Monitoritzar la lluminositat.
- Notificació acústica de l'alarma.
- Informar de la potència rebuda en la prova de cobertura.
- Incloure altres sensors com ara detectors de fum, flama, aigua, etc.
- Interfície gràfica d'usuari.

Descripció detallada

En aquest apartat veurem quins han estat els recursos tecnològics emprats per implementar el sistema descrit anteriorment.

Plataforma de desenvolupament

El desenvolupament dels diferents mòduls s'ha dut a terme sota entorn Linux (Ubuntu 11.10 *Oneiric Ocelot*)⁷⁷ de 32 bits.

Per a les motes, tant pel node base com pel nodes remots, hem fet servir TinyOS⁷⁸, (distribució *Oneiric*) desenvolupat per la Universitat de *Berkeley* en cooperació amb *Intel Research* i *Crossbow Technology* i dissenyat expressament per a gestió de xarxes de dispositius sense fils.

Aquest entorn inclou el compilador *nesC*⁷⁹ (*network embedded systems C*) basat en components i orientat a events. Per tal d'enviar el codi binari a les motes, hem fet servir la utilitat *meshprog*⁸⁰ versió 0.1.2.

⁷⁷ <http://releases.ubuntu.com/11.10/>

⁷⁸ <http://www.TinyOS.net/>

⁷⁹ <http://nesc.sourceforge.net/>

⁸⁰ http://eimtcollab.uoc.edu/softcou24_12/meshprog-0.1.2.tar.gz

Les motes del projecte utilitzen el microprocessador Atmel ATZB-24-A2 (*ZigBit™ 2.4 GHz Wireless Modules*)⁸¹. Per tal de poder accedir a les característiques tant del processador com dels sensors que inclou aquest mòdul, hem fet servir l'estructura definida per l'*Institut Tecnològic de Zurich*⁸².

Com a IDE de desenvolupament hem utilitzat la versió 3.5.2 d'*Eclipse (Galileo SR2)*⁸³ amb el *plugin Yeti2 (ETHZ)*⁸⁴, que permet integrar *TinyOS* en *Eclipse*. La interfície d'usuari s'ha implementat en Java, utilitzant les llibreries *SWT*⁸⁵ (*Standard Widget Toolkit*) i *WindowBuilder Pro*⁸⁶.

Node remot - FireRemote

El primer dubte que se'ns va plantejar a l'hora de començar el desenvolupament d'aquest mòdul va ser on ubicàvem el tractament dels missatges. *TinyOS* fa servir *Active Messages*, un mecanisme que permet associar un identificador a cada missatge i adreçar-lo al destí corresponent.

Així doncs, el primer pas en l'anàlisi consistia en saber si dissenyàvem un sistema on cada component tractés els seus missatges i funcionés de manera independent, o si es centralitzava el tractament en un component principal, definint interfícies que ens permetessin disparar *events* i mètodes per modificar el comportament de la resta de components.

Finalment vàrem decidir deixar que cada component tractés els seus missatges funcionant de manera independent en l'enviament i la recepció de missatges però fent que la comunicació entre mòduls, quan era necessària, es fes a través del component principal (App de tipus *FireRemoteC*) mitjançant interfícies i events.

⁸¹ <http://www.atmel.com/Images/doc8226.pdf>

⁸² http://www.ethz.ch/index_EN

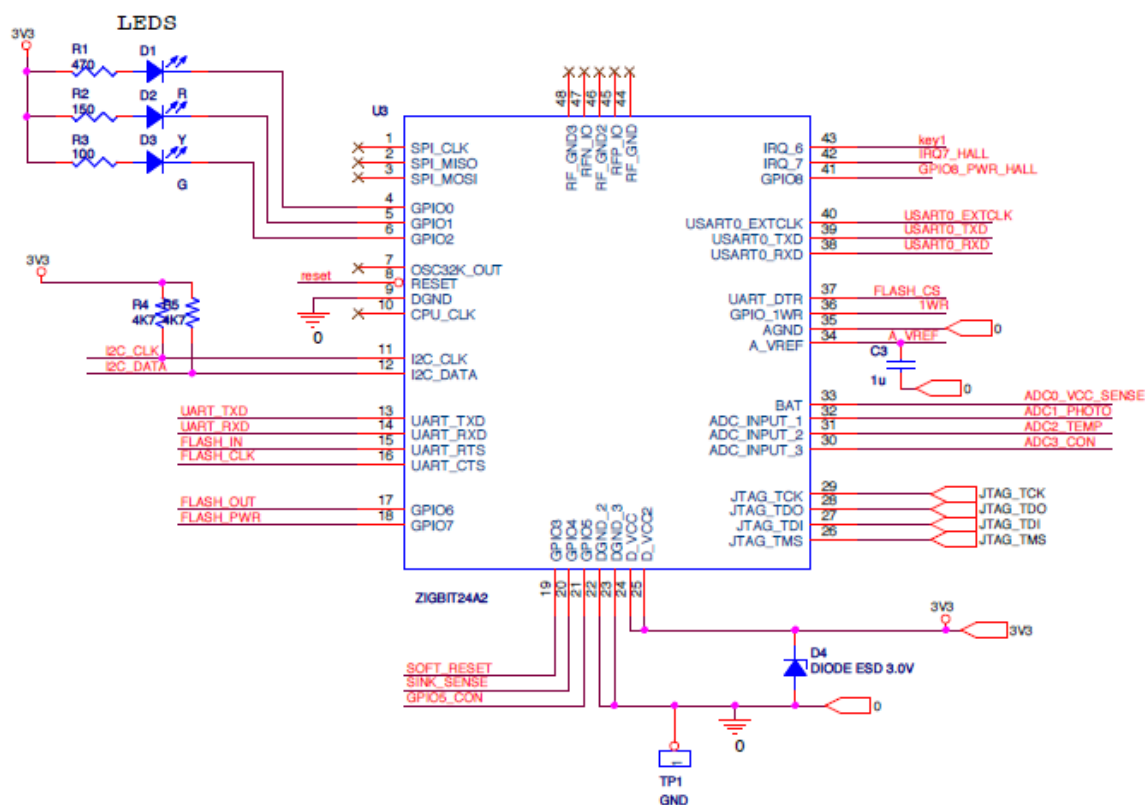
⁸³ <http://www.eclipse.org/galileo/>

⁸⁴ <http://tos-ide.ethz.ch/wiki/index.php>

⁸⁵ <http://www.eclipse.org/swt/>

⁸⁶ <http://dl.google.com/eclipse/inst/d2wbpro/latest/3.5>

Un cop decidit el model, el següent pas va ser veure quines connexions físiques necessitàvem accedir. Per això, vàrem fer servir la informació de l'esquemàtic de la mota⁸⁷. Així vam veure que, per exemple, per llegir els pics magnètics del sensor d'efecte Hall necessitem tenir en compte els pins 41 i 42 del mòdul *Zigbit*, com podem veure en la il·lustració següent:



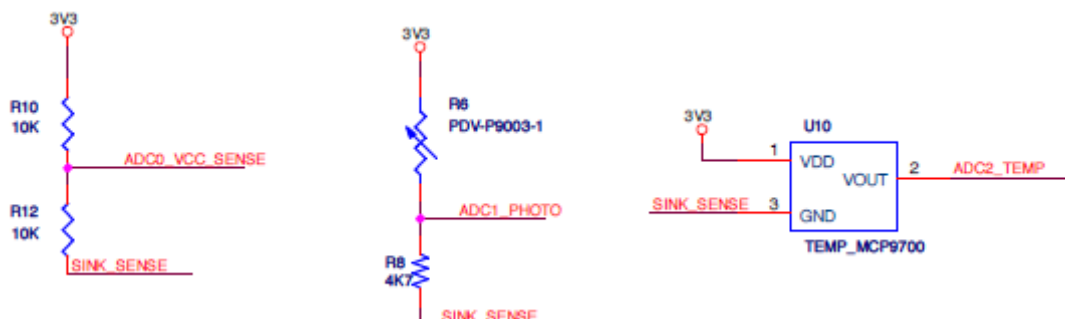
Il·lustració 15: Esquema de connexions del mòdul *Zigbit* 24 A2

Aquests pins es corresponen amb el Port E3 (alimentació) i E7 (interrupció per tal de detectar que s'ha polsat) del model de la plataforma cou24 de *TinyOS*.

De la mateixa manera, podem veure que l'estat de la bateria fa servir el pin 33 i els sensors de lluminositat i temperatura els 32 i 31 respectivament. Aquests es corresponen amb els canals `ATM128_ADC_SINGL_ADC0`, `ATM128_ADC_SINGL_ADC1` i `ATM128_ADC_SINGL_ADC2` respectivament utilitzats pel component `AdcReadClientC` de *TinyOS*.

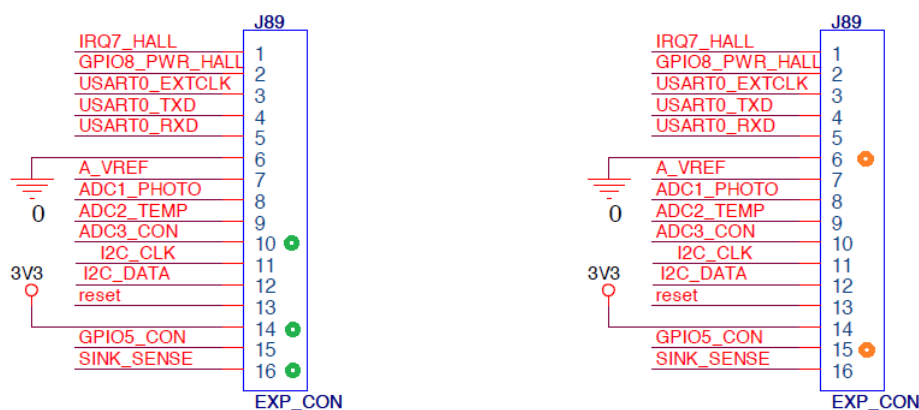
⁸⁷ http://eimtcollab.uoc.edu/softcou24_12/mota_cou_1_2_24A2_2011.pdf

En la il·lustració següent podem veure els esquemes dels sensors de l'estat de la bateria, d'il·luminació i de temperatura:



Il·lustració 16: Esquema de connexió dels sensors

Els sensors de Vcc, il·luminació i temperatura tenen connectat GND a SINK_SENSE, corresponent al pin 20 del microcontrolador, el que ens permet habilitar-los (posant un 0 en aquest pin) i deshabilitar-los conjuntament a través del port G1⁸⁸. També hem afegit un sensor extern i un dispositiu extern d'alarma (un *buzzer*). Per tal d'integrar-los en el nostre sistema, els hem connectat a la placa de la mota fent servir el connector d'expansió. En la imatge següent podem veure l'esquema corresponent:



Il·lustració 17: Esquema del connector d'expansió de les motes

Com veiem en la Il·lustració 17, per al sensor extern hem fet servir els pins 16 (SINK_SENSE) per a GND, 14 (alimentació) i 10 (ADC3_CON) per llegir les dades. El *buzzer* l'hem connectat a GPIO5_CON (pin 15) i a GND (pin 6).

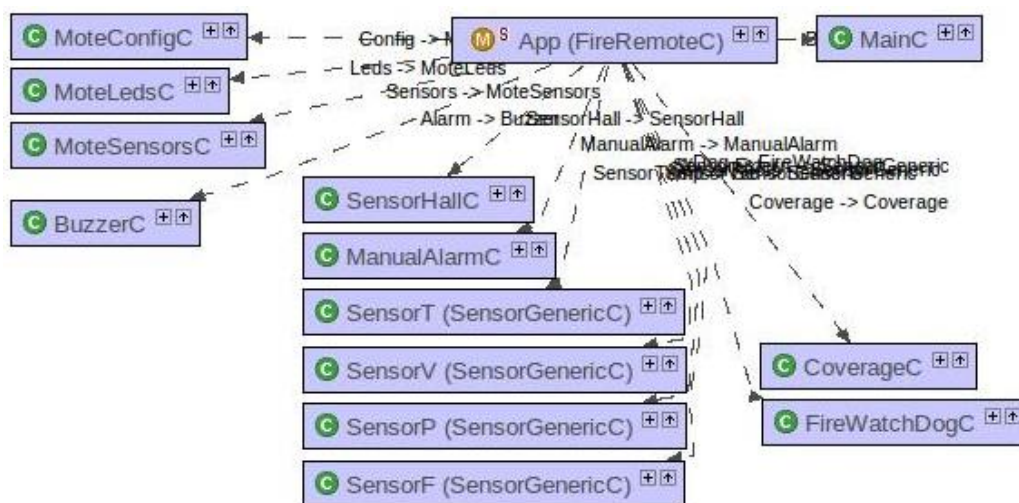
⁸⁸ L'equivalència ports i pins està a: http://disco.ethz.ch/theses/hs08/mb900_TinyOS.pdf

Com podem veure en l'esquema de Zigbit 24 A2 de la Il·lustració 15 en la pàgina 37, ADC3_CON es correspon a ADC_INPUT_3 (pin 30), el canal ATM128_ADC_SINGL_ADC3 utilitzat pel component *AdcReadClientC* de *TinyOS*, i GPIO5_CON amb GPIO5 (pin 21) que es correspon amb el port G2 del model de la plataforma cou24 de *TinyOS*.

A continuació es va definir el codi d'activació dels *leds* per tal que l'usuari pogués conèixer l'estat del node segons els colors actius:

- El *led* verd s'ha reservat per al senyal d'operació, de manera que el parpelleig cada 2 segons, indica que els sensors estan activats i el funcionament és correcte. Un interval de 5 segons indica que els sensors estan apagats.
- El vermell s'ha reservat per a les alarmes, tant manuals com automàtiques. Un funcionament intermitent indica que s'ha produït una alarma pendent de confirmar. Un cop confirmada, es fixa (i adicionalment s'activa la connexió del *buzzer* que emet un senyal acústic). Si expira el temps de confirmació, s'anul·la automàticament.
- El taronja indica que hi ha recepció / transmissió de dades per ràdio.

Un cop fet el disseny, l'esquema de l'aplicació corresponent als nodes remots i la interacció entre els diferent mòduls ha quedat com es mostra en la següent il·lustració:



Il·lustració 18: Esquema de components de *FireRemote*

A continuació fem una petita descripció de les funcionalitats de cadascun dels mòduls:

- *FireWatchDog* s'encarrega de que la mota no es quedi penjada. Hem aprofitat la senyalització de funcionament de la mota per tal d'anar resetejant el comptador i no permetre que el sistema es reiniciï si el funcionament és correcte.
- *CoverageC*. Mesura els valors de *rss*, *lqi* i *power* del missatge enviat per l'aplicació, omple els camps corresponents i el torna a la mota base. Segons els *flags* activats en el *make*, obtindrem uns resultats o altres.
- *SensorT*, *SensorP*, *SensorV* i *SensorF* estan basats en *SensorGenericC*. Cadascun d'ells està connectat a un dels canals ADC i obté les lectures del sensor corresponent.
- *ManualAlarmC* gestiona el botó d'usuari i envia una alarma manual a l'aplicació d'usuari si no hi ha una alarma activa i la cancel·la si ja està activa (d'aquesta manera cancel·lem el *buzzer*).
- *SensorHallC* detecta els pics magnètics i senyala un event que és capturat per App per activar o desactivar els sensors.
- *MoteSensorsC* encapsula la gestió dels sensors, modificant el valor de *PortG1* per habilitar / deshabilitar els sensor de manera global amb *SINK_SENSE*.
- *BuzzerC* implementa els mètodes per activar o tancar el *buzzer*.
- *MoteLedsC* gestiona els codis de colors assignats als estats de la mota. El led verd indica si els sensors estan activats (parpelleig cada 2 segons) o desactivats (cada 5 segons). El groc el fem servir per a notificacions: missatge enviat / rebut (parpelleig ràpid) bateria baixa (parpelleig cada 2 segons) o fora de cobertura (fixe). El led vermell per a les alarmes: alarma detectada o botó d'usuari polsat (parpelleig ràpid) enviat a l'aplicació d'usuari (cada 2 segons) o alarma confirmada (fixe).
- *MoteConfigC*, demana la configuració a l'aplicació d'usuari i, un cop rebuda, amb els mètodes corresponents de cada component de sensor assigna els valors (interval de lectura i límits).
- *App (FireRemoteC)* És el punt d'entrada a l'aplicació. El primer que es fa és demanar la configuració a l'aplicació d'usuari. Un cop rebuda, es configuren els sensors i s'inicien els sensors. Un cop estabilitzats, comencen els cicles de recollida de dades.

Així doncs, hem vist que el component *FireRemoteC* controla les operacions internes, i les notificacions a l'usuari a través dels *leds*. Per tal d'accedir als sensors, hem creat el component *SensorGenericC* i la interfície *SensorInterface* que ens permet 'instanciar' un component per a cada sensor personalitzat amb paràmetres.

De la mateixa manera, hem creat un component i una interfície per a les alarmes manuals, pels *leds*, pel *buzzer*, per a la configuració de la mota, per a l'activació dels sensors, per a la cobertura i per al temporitzador *Watch Dog*. En aquest darrer hem establert un temps de *timeout* de 8 segons, temps en el que si no s'anul·la el compte enrere, amb un senyal produït per un funcionament correcte del dispositiu, aquest es reinicia.

En el prototipus desenvolupat s'han fet servir els identificadors 0 i 1 respectivament per a les motes base i per a la remota.

Els sensors externs que hem fet servir són de tipus '*brick*', i es componen d'una placa amb el sensor i una interfície per connectar-lo directament a la mota. Hem fet servir pins de tipus '*solderless headers*' que no és necessari soldar, molt adients per a fer proves i prototipus.

Els sensors que hem provat han estat:

- Flame sensor de DFRobotics⁸⁹
- Smoke sensor (MQ2) de Seed Studio⁹⁰
- Buzzers de DFRobotics⁹¹ i Seed Studio⁹²
- Water sensor de Seed Studio⁹³
- Touch sensor de Seed Studio⁹⁴

⁸⁹ http://www.dfrobot.com/index.php?route=product/product&path=36_58&product_id=195

⁹⁰ [http://www.seeedstudio.com/wiki/Electronic_brick_-_Smoke_sensor\(MQ2\)](http://www.seeedstudio.com/wiki/Electronic_brick_-_Smoke_sensor(MQ2))

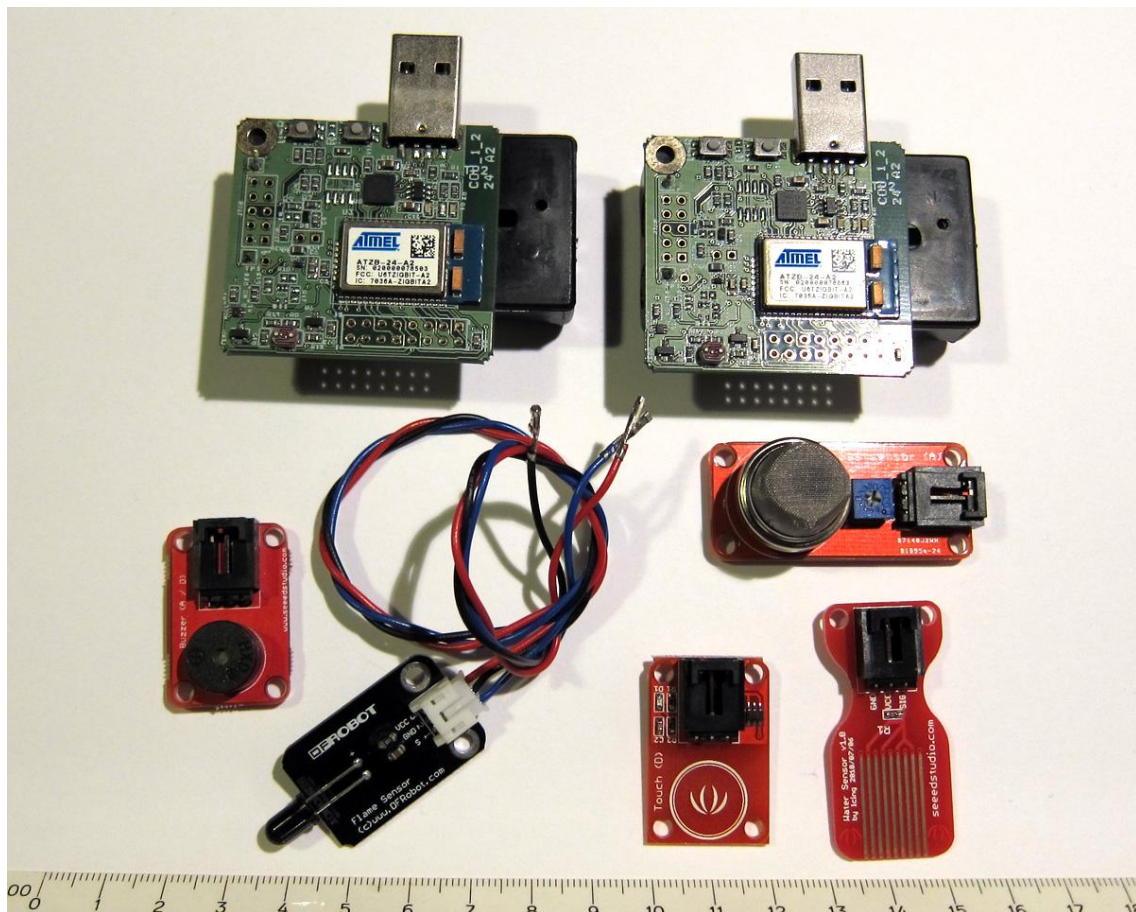
⁹¹ http://www.dfrobot.com/index.php?route=product/product&filter_name=buzzer&product_id=84

⁹² [http://www.seeedstudio.com/wiki/Electronic_brick_-_Buzzer\(A/D\)](http://www.seeedstudio.com/wiki/Electronic_brick_-_Buzzer(A/D))

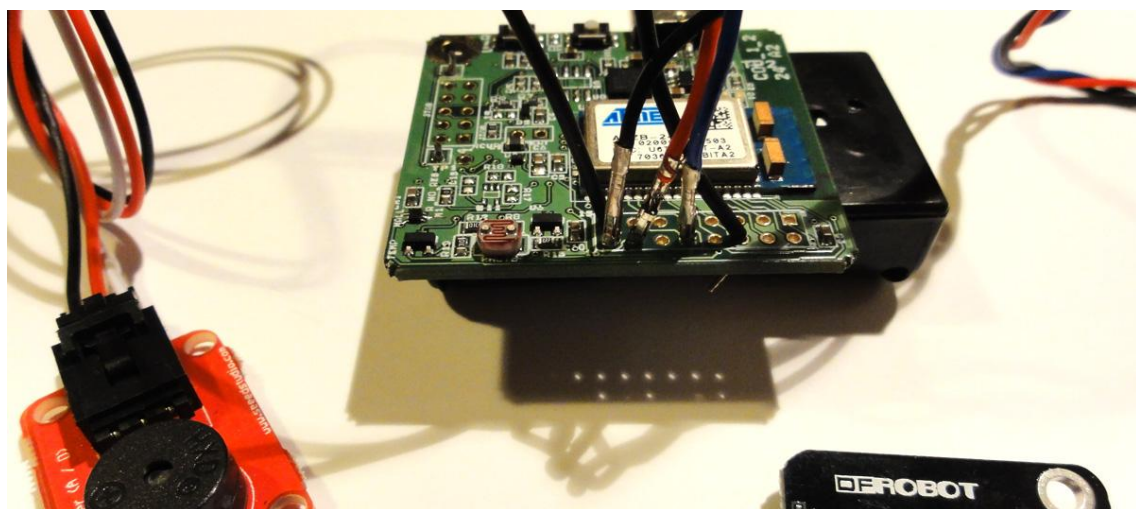
⁹³ http://www.seeedstudio.com/wiki/Electronic_Brick_%E2%80%93_Water_Sensor

⁹⁴ http://www.seeedstudio.com/wiki/Electronic_brick-Touch_Sensor_module

En les imatges següents podem veure els dispositius emprats i la connexió al port d'expansió de les notes:



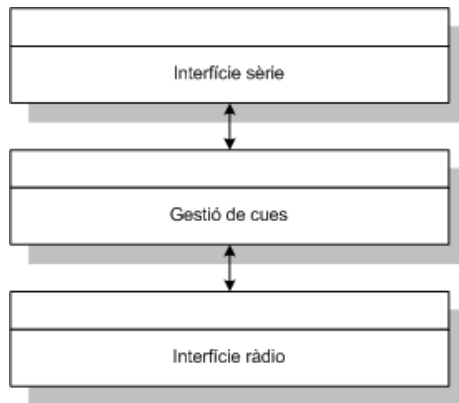
Il·lustració 19: Motes i sensors del projecte



Il·lustració 20: Connexió al port d'expansió

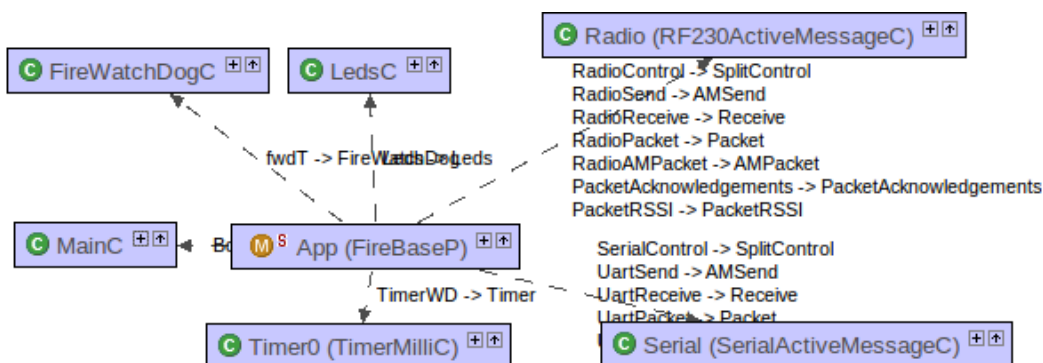
Node local - FireBase

La funció d'aquest mòdul és la de fer de pont entre el port sèrie del PC i la ràdio de la mota, incorporant una cua d'entrada i una altra de sortida per tal d'emmagatzemar els missatges en cas de produir-se pics en la càrrega de les transmissions.



Il·lustració 21: Esquema actual de *FireBase*

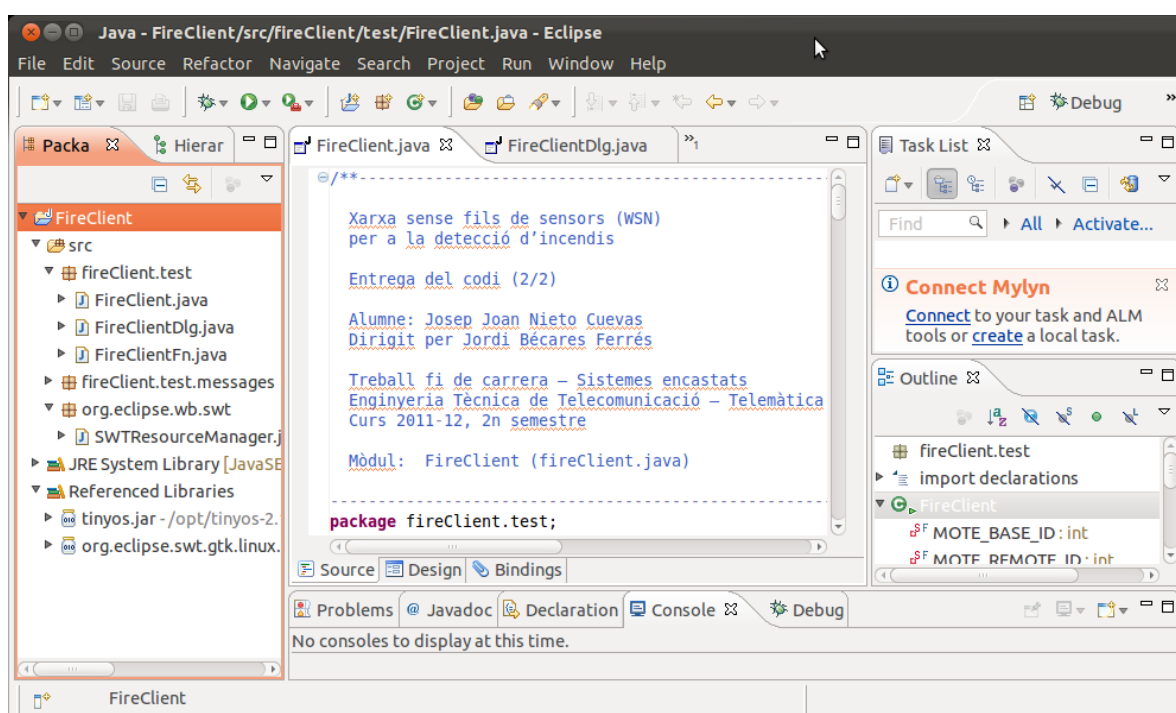
El codi del node local està basat en *BaseStation*, que forma part de la distribució de *TinyOS*. Es van fer fet però, algunes modificacions. En primer lloc, es van ampliar les cues d'entrada i sortida. També es va afegir un component *WatchDog* per comprovar que la mota no es queda penjada i que sempre està disponible (*FireWatchDogC*). Aquesta modificació va estar particularment important per tal d'alleujar els problemes de saturació detectats en els diferents plans de proves. A més, per tal de registrar la cobertura, en aquest mòdul es mesura el valor *RSSI* en els missatges de tipus *heartbeat* i s'afegeix als valors de mesura de la bateria enviats pels nodes remots. En la imatge següent es pot veure un esquema dels components utilitzats en *FireBase*:



Il·lustració 22: Components de *FireBase*

Aplicació d'usuari - FireClient

L'aplicació d'usuari ha estat desenvolupada en Java, i costa de dos *packages*. El primer conté l'aplicació pròpiament dita i el segon la definició dels missatges per tal d'intercanviar informació amb les motes. Com podem veure en la il·lustració següent, per a dissenyar la interfície gràfica hem fet servir SWT, que juntament amb la llibreria de *TinyOS* (*TinyOS.jar*), són les dues referències externes del projecte.



Il·lustració 23: Entorn de desenvolupament de Java

L'aplicació es compon de tres fitxers:

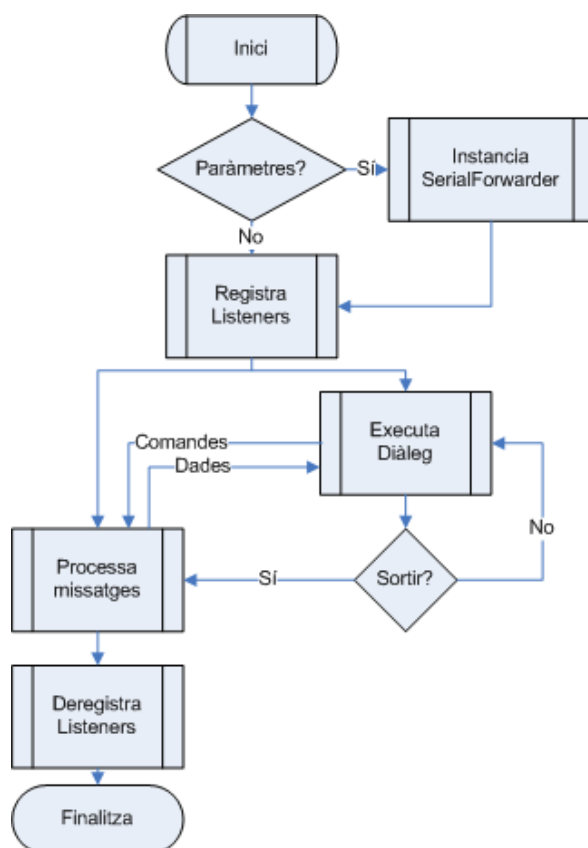
- *FireClient.java*. Conté la lògica de l'aplicació, i s'encarrega de la comunicació amb les motes a través de la utilitat subministrada per *TinyOS*, *SerialForwarder*.
- *FireClientDlg.java*. Gestiona la interacció amb l'usuari i envia les peticions de comunicació amb les motes a *FireClient*.
- *FireClientFn.java*. En aquest mòdul hem concentrat les funcions de càlcul necessàries per convertir les unitats de mesura retornades pels sensors de les motes a valors estàndards. Per exemple, transforma '*counts*' a volts, lux, graus centígrads, etc. i a l'inrevés.

Per comoditat, es van incloure inclòs les dues llibreries externes i les classes de l'aplicació dins d'un únic fitxer .jar. Així doncs, podem executar de la manera següent:

```
java -jar fireClient.jar  
java -jar fireClient.jar ttyUSB0 9002
```

En el primer cas necessitem tenir *SerialForwarder* en funcionament, en el segon, la mateixa aplicació el crida amb els paràmetres indicats (port sèrie i port de l'aplicació).

En la classe principal es fa un *implements* de la classe *MessageListener*, definida per *TinyOS*, que ens permet tenir accés als missatges enviats per les motes. També s'inclou el modificador *extends TimerTask*, per definir tasques periòdiques, com ara controlar si es perd el senyal de *heartbeat*, o com havíem previst per més endavant, per comprovar l'estat del node base. L'esquema de funcionament de l'aplicació és:



Il·lustració 24: Esquema del funcionament de l'aplicació

La primera carpeta de la interfície gràfica d'usuari, que podem veure a continuació, mostra un resum de l'estat actual del sistema i permet acceptar o rebutjar les alarmes (la llista del desplegable conté les darreres 10 alarmes sense confirmar).

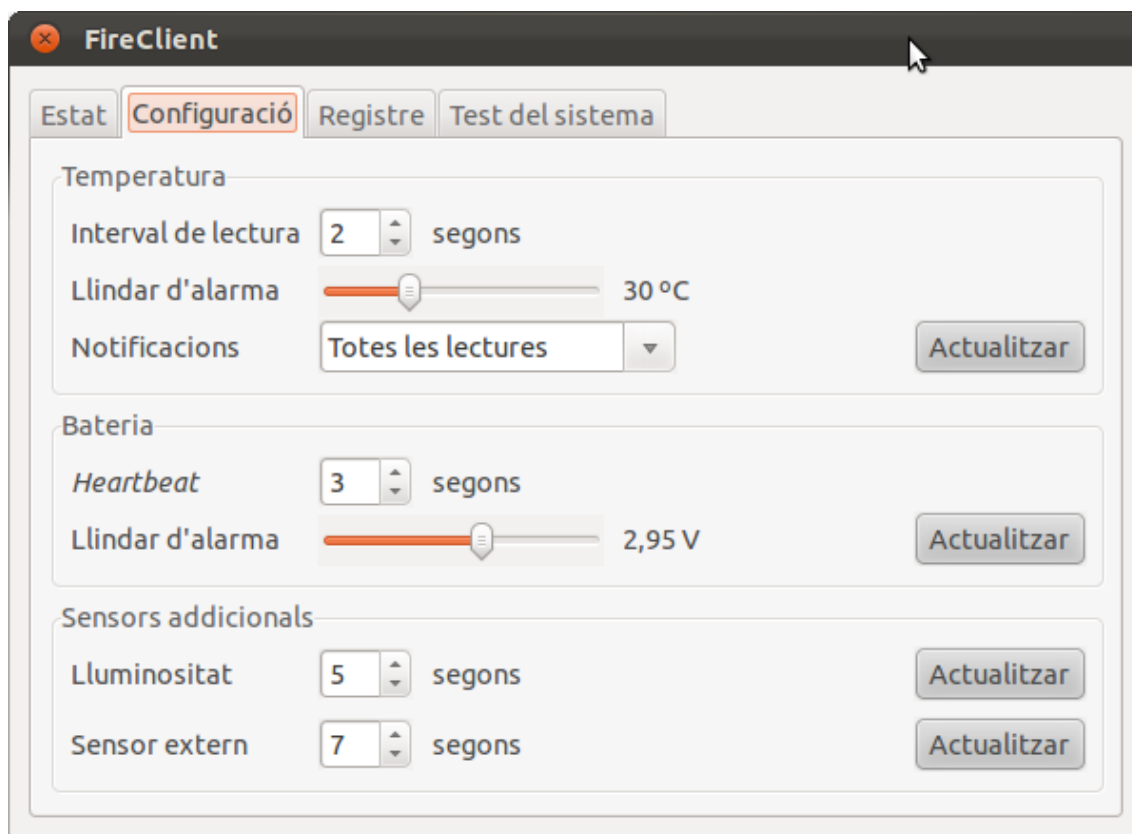


Il·lustració 25: Interfície d'usuari de *FireClient*

Aquesta pantalla s'actualitza a mida que arriben els missatges, pel que es normal que inicialment no apareguin valors. Les lectures d'estat, excepte les dels sensors externs, estan en les unitats corresponents: graus centígrads, lux, volts i dBm. En els annexos podem trobar les fórmules que hem fet servir per a les conversions.

En el cas de la cobertura, el resultat pot diferir depenent de l'opció triada, ja que en la lectura automàtica es mesura el valor en la mota local, a partir dels missatges rebuts, mentre que en el test que podem trobar en la darrera carpeta, '*Test del sistema*', es mesura en la mota remota, a partir d'un missatge enviat des de la local.

Mitjançant els controls que trobem en la segona carpeta, se'ns permet configurar el comportament dels sensors. En la imatge següent podem veure la pantalla de configuració de l'aplicació:



Il·lustració 26: Pantalla de configuració de funcionament

Hem tingut en compte les següents restriccions:

Intervals de lectura:

- Temperatura: d'1 a 60 segons.
- Bateria, *heartbeat*: d'1 a 5 segons.
- Lluminositat: d'1 a 60 segons.
- Sensor extern: d'1 a 60 segons

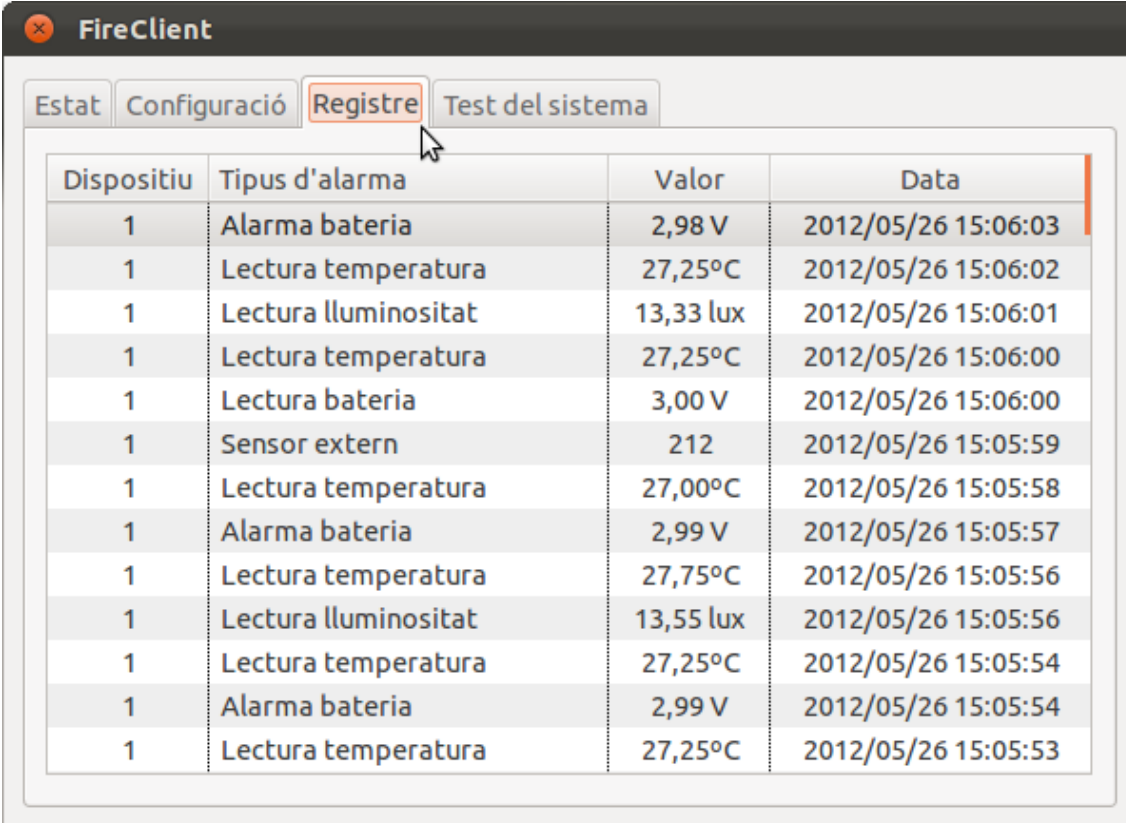
Llindars d'alarma:

- Temperatura: de 0 a 100 °C
- Bateria: 0 a 5 Volts.

En la configuració corresponent a la temperatura podem triar si volem rebre totes les lectures o només les alarmes. En el primer cas, es fa una diferenciació entre valors normals i els que han ultrapassat el llindar.

Pel *heartbeat* no podem triar rebre només les alarmes, ja que el fem servir com a senyal de que la mota està activa. Un llindar considerat normal per a aquest tipus de sistemes està al voltant dels 60°C. Nosaltres per a fer les proves hem estimat uns valors al voltant dels 30°C.

En la tercera carpeta que podem veure a continuació, tenim un registre amb els darrers 100 missatges rebuts:



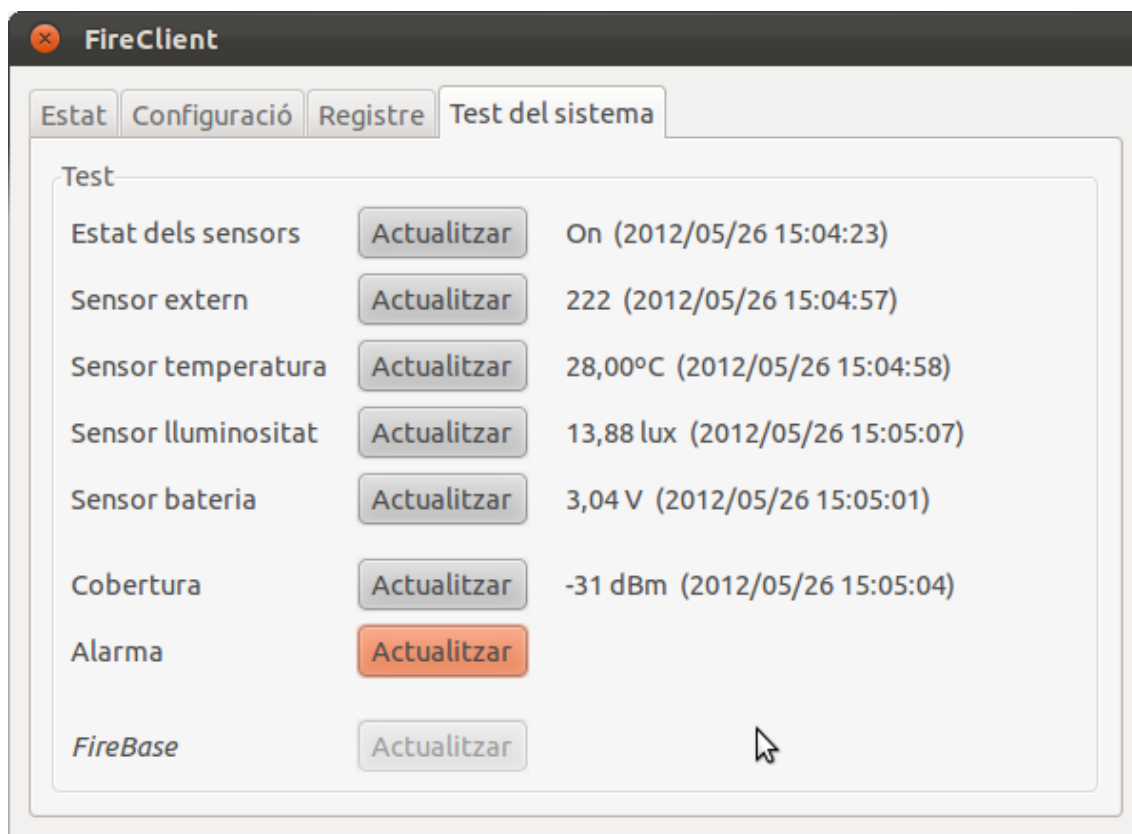
The screenshot shows the FireClient application interface. At the top, there are four tabs: 'Estat', 'Configuració', 'Registre', and 'Test del sistema'. The 'Registre' tab is selected and highlighted. Below the tabs is a table with four columns: 'Dispositiu', 'Tipus d'alarma', 'Valor', and 'Data'. The table contains 14 rows of data, each representing a received message. The data is as follows:

Dispositiu	Tipus d'alarma	Valor	Data
1	Alarma bateria	2,98 V	2012/05/26 15:06:03
1	Lectura temperatura	27,25°C	2012/05/26 15:06:02
1	Lectura lluminositat	13,33 lux	2012/05/26 15:06:01
1	Lectura temperatura	27,25°C	2012/05/26 15:06:00
1	Lectura bateria	3,00 V	2012/05/26 15:06:00
1	Sensor extern	212	2012/05/26 15:05:59
1	Lectura temperatura	27,00°C	2012/05/26 15:05:58
1	Alarma bateria	2,99 V	2012/05/26 15:05:57
1	Lectura temperatura	27,75°C	2012/05/26 15:05:56
1	Lectura lluminositat	13,55 lux	2012/05/26 15:05:56
1	Lectura temperatura	27,25°C	2012/05/26 15:05:54
1	Alarma bateria	2,99 V	2012/05/26 15:05:54
1	Lectura temperatura	27,25°C	2012/05/26 15:05:53

Il·lustració 27: Registre de missatges rebuts

Les dades que podem veure es corresponen a l'identificador de la mota, una descripció del tipus de missatge (lectura / alarma / pèrdua de senyal) el valor llegit pels sensor en les unitats de la magnitud corresponent i la data d'arribada del missatge. Aquesta data és local a la màquina de l'usuari.

Finalment, en la quarta carpeta, una sèrie de botons que ens permet provar de manera manual algunes de les opcions de l'aplicació. Com podem veure en la Il·lustració 28 el test de connectivitat amb la mota base està deshabilitat en aquesta versió.



Il·lustració 28: Pantalla de test del sistema

En la resta d'opcions el funcionament és similar: s'envia un missatge a la mota (en aquest cas a la mota 1) demanant l'estat dels sensors del paràmetre que volem llegir. En el missatge de retorn trobem el valor, que és interpretat seguint el procediment habitual pels missatges. Un identificador afegit a les dades del missatge de retorn ens indica que es tracta d'una sol·licitud manual i que hem de mostrar el resultat en aquesta pantalla.

L'opció de cobertura a diferència de la lectura automàtica obté el valor de *RSSI* en la mota remota enlloc de en la mota local, i torna els valors obtinguts en el missatge corresponent.

El botó d'alarma ens permet simular una alarma (led i *buzzer*) en la mota remota.

Viabilitat tècnica

En aquests moments creiem que el projecte es troba en la fase de prototip. Pel tal de que sigui comercialment viable caldria fer algunes modificacions, algunes d'elles importants.

En primer lloc, a nivell d'aplicació, caldria resoldre el problema del reconeixement de missatges (ACK). Donat que es tracta d'un tipus d'aplicació en el que és crític que no es perdin missatges, no és acceptable que les cues d'entrada / sortida del mòdul base es saturin i el dispositiu estigui fora de connexió durant uns segons. Per això, creiem que s'ha de reescriure el codi de la mota base per mirar de solucionar la incidència.

D'altra banda, el projecte s'ha desenvolupat tenint en compte únicament una mota base i un altra remota, pel que errades en els sensors de mesura del dispositiu de recollida dels paràmetres ambientals no són fàcilment comprovables. La situació ideal passaria per tenir una instal·lació múltiple de sensors remots, per exemple disposats formant un triangle, de manera que les lectures comuns ens servissin per validar les alarmes.

També creiem que caldria dotar als dispositius remots d'algun tipus de blindatge. No oblidem que segons les especificacions operatives del mòdul ATZB-24-A2 (*Bit™ 2.4 GHz Wireless Modules*)⁹⁵ el paràmetre *Operating Temperature Range* està entre -20°C i +70°C i que *Operating Relative Humidity Range* ha de ser inferior al 80%. En un incendi és fàcil superar aquest rang de temperatures, pel que es podria donar la situació de que la mota deixés de ser operativa sense notificar a l'usuari. A més, en cas de proveir d'un sistema automàtic d'extinció, per exemple per aspersió, caldria protegir les motes de l'aigua.

Els sensors addicionals que hem afegit són útils per millorar en el diagnòstic de la situació de risc. Tot i així, per fer les proves hem fet servir aparells calibrats, segons els corresponents *datasheets*, per operar en un entorn a 25°C i amb una alimentació de 5V, pel que caldria estudiar la fiabilitat a temperatures elevades i 3,3V d'alimentació.

⁹⁵ <http://www.atmel.com/Images/doc8226.pdf>, apartat 3.2

Valoració econòmica

Per a la valoració econòmica hem fet servir diverses fonts, i hem valorat el desenvolupament del prototipus, descartant el temps dedicat a la documentació. Segons el BOE del 18 de marzo de 2009, on es registra y publica el *XVI Convenio colectivo estatal de empresas de consultoría*⁹⁶ el preu / hora mínim regulat pel desenvolupament del projecte oscil·la entre 10 i 20 € hora. Per a les motes hem fet una estimació, basant-nos en informació sobre el preu dels components d'uns 30 €. A continuació podem trobar un resum de la valoració del desenvolupament del prototipus.

Descripció	Proveïdor	Preu unitari	Quantitat	Total
Anàlisi dels sistema		20,00 €	40	800,00 €
Desenvolupament FireRemote		15,00 €	88	1.320,00 €
Desenvolupament FireBase		15,00 €	28	420,00 €
Desenvolupament FireClient		15,00 €	106	1.590,00 €
Desplegament del sistema		10,00 €	8	80,00 €
Subtotal				4.210,00 €
Mota	UOC	30,00 €	2	60,00 €
PC HP Compaq 6200 Pro SFF	HP	825,00 €	1	825,00 €
DFRobot Fame Sensor	cooking-hacks	3,00 €	1	3,00 €
EB Smoke Sensor (MQ2)	cooking-hacks	6,90 €	1	6,90 €
EB Buzzer	cooking-hacks	4,00 €	1	4,00 €
EB Water Sensor	cooking-hacks	2,50 €	1	2,50 €
EB Touch Sensor Module	cooking-hacks	6,00 €	1	6,00 €
Subtotal				907,40 €
Total final				5.117,40 €

Aquest pressupost pot variar segons el rang de preus dels dispositius que es facin servir. Per exemple, l'empresa *Advanticsys* comercialitza una gamma bastant completa de motes⁹⁷ basades en la *TelosB* i compatibles amb TinyOS 2.x i ContikiOS⁹⁸. Els preus oscil·len entre els 65 € del model UD1000 (*802.15.4 compliant USB dongle*) sense sensors i els 85 € del model XM1000 amb sensors de temperatura,

⁹⁶ <http://www.boe.es/boe/dias/2009/04/04/pdfs/BOE-A-2009-5688.pdf>

⁹⁷ <http://www.advanticsys.com/shop/802154-mote-modules-c-3.html>

⁹⁸ <http://www.contiki-os.org/>

humitat i lluminositat. A més, ofereix una sèrie de *sensor boards*⁹⁹, com el model AR1000 amb sensors addicionals de CO, CO2 i pols (fum) amb un cost de 162 €. *SOWNet Technologies*¹⁰⁰ ofereix productes en la línia *G-Node* també compatibles amb *TinyOS*, com el *G-Node G301* per 68 €, amb sensors addicionals en la placa *GColta++ sensor board* per 48,50 €. En *Sparkfun*¹⁰¹ podem trobar la placa *ATmega128RFA1 Development Board* per 44 €.

Conclusions

En l'apartat de conclusions hem inclòs, a més, la revisió dels objectius, les propostes de millora i l'autoavaluació.

Revisió dels objectius

Per tal de portar un control acurat dels objectius, vàrem decidir fer una taula amb els requeriments i l'estat de cadascun d'ells. Aquest resum, que podem veure a continuació amb els estats finals del projecte, també ens va servir per tenir una idea clara dels objectius que s'havien d'acomplir i en quin component del sistema s'havia d'implementar.

1. Detectar incendis

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

⁹⁹ <http://www.advanticsys.com/shop/802154-sensor-boards-c-5.html>

¹⁰⁰ <http://www.sownet.nl/index.php/en/products/gnode>

¹⁰¹ <http://www.sparkfun.com/products/9734>

2. Notificar automàticament l'alarma de forma remota.

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

3. Notificar automàticament l'alarma de forma local.

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

4. Sistema manual per activar l'alarma

- a) El node ha de proveir d'un botó d'emergència per tal d'activar l'alarma de forma manual **Implementat. Simulat amb el botó USR.**

5. Transmissió d'alarmes sense pèrdues

- a) No es pot perdre cap alarma d'incendi **Implementat, encara que vàrem tenir alguns problemes amb el reconeixement dels missatges.**
- b) La comunicació s'ha de fer amb ACK **Igual que l'apartat anterior.**

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

- a) L'aplicació ha de mostrar per pantalla quan hi ha hagut una alarma. **Implementat**
- b) Ha de mostrar: **Implementat, l'hora que es mostra és local a l'aplicació.**
 - 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.
- c) L'usuari ha de poder confirmar la recepció **Implementat**

7. Monitoritzar la bateria dels nodes de la xarxa.

- a) Cada node enviarà la seva bateria de forma periòdica cada L segons **Implementat**
- b) L ha de ser configurable des de l'aplicació **Implementat**

- c) Quan el node detecti que té la bateria crítica, enviarà una alarma amb el seu estat **Implementat**
- d) 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. **Implementat**

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

- a) Sensor apagat. De forma periòdica pot encendre un led, conforme el node pot funcionar correctament. **Implementat**
- b) No alarma. De forma periòdica pot encendre un led, conforme el node esta funcionant correctament. **Implementat**
- c) Alarma detectada **Implementat**
- d) Alarma rebuda en estació central **Implementat**
- e) Alarma reconeguda **Implementat**
- f) Nivell bateria baix **Implementat**
- g) Fora de cobertura **Implementat**

9. Sistema de protecció de caigudes. Els nodes no es poden quedar "penjats"

- a) El node ha de fer servir el *WatchDog* per evitar quedar-se penjat. **Implementat**
- b) Ha de recuperar la configuració de forma automàtica **Implementat**

10. Sistema de *debug* de l'aplicació.

- a) El usuari ha de poder veure les temperatures dels diferents sensors. **Implementat**
- b) El usuari pot demanar rebre les dades cada M segons. M ha de ser configurable. **Implementat**

11. Prova de cobertura

- a) Proporcionar un sistema per comprovar si hi ha cobertura amb el node base. **Implementat**

12. Sistema d'activació

- a) El sensor no estarà sempre encès. Implementat, afegida activació remota per ràdio, encara que en aquesta versió no l'hem afegit a l'aplicació d'usuari.
- 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

- a) Ha de ser simple Implementat
- b) Autoinstal·lable i autoexecutable Implementat, paquet .jar
- c) Contenir un menú d'ajuda (help amb les comandes) Manual d'usuari, al ser entorn gràfic l'usuari no necessita comandes

Funcionalitats extres

- a) Monitoritzar la lluminositat Implementat
- b) Integrar un *buzzer* per notificar de forma acústica l'alarma Implementat
- c) Afegir sensors externs Implementat
- d) En la prova de cobertura informar de la potència (dB) rebuda. Implementat
- e) Interfície gràfica per l'aplicació del PC Implementat, amb java i SWT.

Proposta de millores

Les propostes de millora al prototipus desenvolupat ja les hem avançat en apartats anteriors, i consistirien principalment en la revisió del sistema de reconeixement de missatges per part de la mota local i en la possibilitat de rebre / enviar missatges a més d'una mota remota per part de l'aplicació de client.

També afegiríem, com en el plantejament inicial, la possibilitat d'emmagatzemar les dades rebudes en una base de dades relacional.

Autoavaluació i conclusions finals

Com a darrer punt de les conclusions hem deixat l'autoavaluació. Un cop finalitzat el projecte, podem dir que creiem que hem assolit els objectius generals que plantejaven al començament del mateix.

En aquests moments tenim una idea bastant més acurada de en que consisteix una xarxa de sensors sense fils, així com de quines són les possibilitats i les limitacions dels dispositius.

Un dels desavantatges plantejats inicialment, que en un principi es va inclús identificar com un risc, finalment no ha estat tan determinant. Es tracta de l'entorn de desenvolupament, que no era el que utilitzem habitualment. No podem sinó que valorar positivament el fet d'haver-nos vist 'obligats' a treballar en Linux, ja que ha estat una experiència molt favorable.

On ens quedem amb una pitjor impressió és en el tema de les comunicacions. Aquí el resultat ha estat pitjor de l'esperat, ja que les pèrdues de paquets han estat més importats de les previstes inicialment. En un sistema crític, com és la detecció d'incendis, on està en perill la integritat física de persones i la pèrdua de béns materials importants, no creiem que sigui acceptable que el sistema no es comporti amb una fiabilitat absoluta, pel que creiem que encara ens ha faltat insistir en aquest punt.

La possibilitat que ens ha ofert el sistema d'ampliar el rang de sensors a utilitzar a través del port d'expansió i de la capa d'abstracció del hardware (HAL, *Hardware Abstraction Layer*) que fa *TinyOS* ha simplificat moltíssim l'ampliació. Falta això sí, calibrar els dispositius per tal d'adaptar-se a les condicions de treball i a les característiques de les motes.

Per finalitzar, dir que independentment del resultat final, creiem que ha estat una experiència molt positiva i que l'elecció d'àrea per aquest treball de final de carrera no podia haver estat més encertada.

Glossari

ADC (*Analog-to-Digital Converter*) Dispositiu que té la capacitat de convertir senyals analògiques en senyals digitals per tal de facilitar el seu processament, i fer el senyal menys sensible al soroll i altres interferències.

Cicle de vida en cascada Metodologia de desenvolupament de programari en la que es defineixen unes etapes ordenades i tancades, de forma que per iniciar una nova etapa ha d'estar finalitzada completament l'anterior.

Diagrama de Gantt Diagrama on es mostra el temps de dedicació previst per a un conjunt de tasques i les relacions existents entre elles.

Sensor d'efecte Hall Sensor que s'utilitza per mesurar pics magnètics gràcies a l'aplicació de l'efecte *Hall* o afectació de camps magnètics perpendiculars.

I2C (*Inter-Integrated Circuit*) Bus de comunicacions dissenyat per *Phillips*, amb velocitats que oscil·len entre els 100 Kbits/s i els 3,4 Mbit/s, utilitzat per a comunicar microcontroladors i perifèrics. Fa servir una línia pel rellotge i una altra per a les dades.

IEEE 802.15.4 protocol estàndard de comunicacions que defineix el nivell físic i l'accés al medi de xarxes sense fils de tipus PAN (Personal Area Network).

SPI (*Serial Peripheral Interface*) Estàndard de comunicacions utilitzat per al bescanvi d'informació entre circuits integrats. funciona amb transferència d'informació controlada per rellotge. Té 4 línies de transmissió: *data in* (MISO), *data out* (MOSI), *clock* (SCLK) i *selecció del xip* (SS).

Memòria EEPROM (*Electrically-Erasable Programmable Read-Only Memory*) Memòria de només lectura que pot ser esborrada elèctricament i tornada a programar.

Microcontrolador Sistema digital format per una unitat central, memòria ROM, una RAM petita i alguns dispositius d'entrada i / o sortida.

Mote Acrònim de *remote sensing*. Es tracta de dispositius de mida reduïda, amb capacitat de procés i equipats amb sensors que funcionen de manera autònoma, però que poden establir comunicacions sense fils amb d'altres dispositius similars o iguals a través d'un protocol comú.

NesC (*Network Embedded Systems C*) és un dialecte del llenguatge de programació C optimitzat per aprofitar les limitacions de memòria i processament dels dispositius que conformen les xarxes de sensors.

Sistema encastat sistema computacional de baix cost específicament dissenyat per a realitzar una tasca concreta on tots els seus components estan integrats.

Smart dust en anglès, pols intel·ligent, xarxa sense fils de motes.

TinyOS 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 ells.

Transceptor Aparell que és la combinació d'un transmissor i un receptor de ràdio aprofitant per fer-ho de circuits comuns als dos processos.

USART (*Universal Asynchronous Receiver-Transmitter*) Integrat que controla els ports i dispositius sèrie. Les seves funcions principals són el control de les interrupcions dels dispositius connectats al port sèrie i convertir les dades en format paral·lel del bus de sistema, a dades en format sèrie.

WDT (*Watch Dog Timer*) sistema basat en un temporitzador independent del microprocessador que té com a funció reiniciar un dispositiu en cas de mal funcionament.

WSN tipus especials de xarxes sense fils que, a diferència de les xarxes convencionals, estan compostes de nodes o estacions amb recursos limitats (motes). Les dades que es transporten es corresponen amb mesures de paràmetres físics, en lloc de dades, veu i serveis com en una LAN o WLAN tradicionals.

Bibliografia

Publicacions

Atorino, Javier I., Bortolín, Luciano P., Rodríguez, Ezequiel, Farías, Rubén O., & Eduardo E., Rodríguez. (2009). “*Experimentos con un sensor de efecto Hall*”. Instituto Politécnico Nacional, Latin American Physics Education Network.

Chee-Yee, C. S. S., & Kumar, S. P. (August 01, 2003). “*Sensor networks: Evolution, opportunities, and challenges*”. Proceedings of the IEEE, 91, 8, 1247-1256.

EWSN 2012, Picco, G. P., & Heinzelman, W. (2012). “*Wireless sensor networks: 9th European conference, EWSN 2012, Trento, Italy, February 15-17, 2012 : proceedings*”. Berlin: Springer.

Faludi, R. (2010). “*Building wireless sensor networks: With ZigBee, XBee, Arduino, and processing*”. Farnham: O'Reilly Media.

Iyengar, S. S. (2011). “*Fundamentals of sensor network programming: Applications and Technology*”. Hoboken, N.J: John Wiley & Sons.

Levis, P., & Gay, D. E. (2009). “*TinyOS programming*”. Cambridge: Cambridge University Press.

Pressman, R. S. (2010). “*Ingeniería del software: Un enfoque práctico*”. México [etc.: MacGraw-Hill.

Suhonen, J. (2012). “*Low-power wireless sensor networks: Protocols, services and applications*”. New York: Springer.

Otis, B., & Rabaey, J. M. (2007). “*Ultra-low power wireless technologies for sensor networks*”. New York: Springer.

Enllaços electrònics

Arp@:Embedded Systems Lab@Home.

[en línia]. <http://cv.uoc.edu/app/mediawiki14/>

[data de consulta: 12/06/2012].

AT86RF230 2,4 GHz Zigbee

[en línia]. http://www.atmel.com/dyn/resources/prod_documents/doc5131.pdf

[data de consulta: 12/06/2012].

ATmega1281.

[en línia]. <http://www.atmel.com/devices/atmega1281.aspx>

[data de consulta: 12/06/2012].

ATMEL Zigbit 2,4 GHz ATZB-24-A2

[en línia]. http://www.atmel.com/dyn/resources/prod_documents/doc8226.pdf

[data de consulta: 12/06/2012].

DFRobot

[en línia]. <http://www.dfrobot.com/index.php?route=product/category&path=36>

[data de consulta: 12/06/2012].

nesC: A Programming Language for Deeply Networked Systems.

[en línia]. <http://nesc.sourceforge.net/>

[data de consulta: 12/06/2012].

TinyOS Home Page.

[en línia]. <http://www.TinyOS.net/>

[data de consulta: 12/06/2012].

Zigbee Alliance

[en línia]. <http://www.zigbee.org/Home.aspx>

[data de consulta: 12/06/2012].

Annexos

Execució i compilació

Hem creat, en cada carpeta amb codi de les motes, un fitxer makefile que ens permet compilar i crear els fitxers binaris que executaran els dispositius. El codi binari de *FireBase* i *FireRemote* l'enviem a les motes amb *meshprog*:

```
cd FireBase/src
make cou24 install,0
meshprog -t/dev/ttyUSB0 -f./build/cou24/main.srec.out-0'

cd FireRemote/src
make cou24 install,1
meshprog -t/dev/ttyUSB1 -f./build/cou24/main.srec.out-1'
```

L'aplicació de l'usuari, *FireClient* l'executem directament com un .jar de Java després d'haver iniciat *SerialForwarder* o amb paràmetres sense iniciar *SerialForwarder*:

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

```
java -jar fireClient.jar
java -jar fireClient.jar ttyUSB0 9002
```

Fórmules de conversió

A continuació fem un resum de les fórmules que hem fet servir per interpretar els valors rebuts dels sensors extretes a partir dels corresponents *datasheets*.

Consideracions inicials

L'ADC de les motes té una resolució de 10 bits, el que ens permetrà tenir 2^{10} 1024 nivells de resolució. Com que sabem que la tensió de referència és de 2560mV, tindrem que el factor de conversió és de 2,5mV.

$$Tensió = \frac{V_{Ref}}{1024} \cdot Counts_{ADC} = \frac{2560mV}{1024} \cdot Counts_{ADC} = 2,5mV \cdot Counts_{ADC}$$

Tensió (Counts a volts)

$$Bateria = 2,5mV \cdot Counts_ADC0 \cdot 2$$

Lluminositat (Counts a lux)

$$Lux = \frac{2,5 \cdot Counts_ADC1}{Bateria/2} \cdot 100$$

Temperatura (Counts a graus centígrads)

$$Temperatura = \frac{2,5 \cdot Counts_ADC2 - 0,50}{10}$$

Cobertura (RSSI a dBm)

$$RSSI_BASE_VAL = 91$$

$$RSSI_dBm = PHY_ED_LEVEL - RSSI_BASE_VAL$$

$$PRF = RSSI_BASE_VAL + 3 \cdot (RSSI - 1)$$

Sensors addicionals

En els sensors addicionals no hem fet cap conversió d'unitats. En part perquè hem utilitzat els mateixos connectors per a tipus diferents de sensors però també perquè els dispositius que hem fet servir, segons els fulls d'especificacions, estaven calibrats en la majoria per funcionar a 25°C i una tensió d'alimentació de 5V. Així doncs, els resultats han estat més aviat experimentals.

Per exemple, en el detector de flama, obteníem unes lectures en l'ADC3 de 900 davant de la flama generada per un misto, mentre que en repòs els valors estaven al voltant de 200.

El sensor de pluja o aigua sí acceptava directament valors d'alimentació de 3,3V i el seu funcionament era mol senzill. En presència d'aigua, les línies conductores de la superfície actuaven com una resistència de l'ordre de k Ω , bastant més petita que la resistència de 1M Ω interna del component, connectant l'alimentació a GND i obtenint a la sortida un senyal analògic de baix voltatge. En cas de no estar mullat, la sortida és el valor de l'alimentació. Aquest sensor és útil, per exemple, per detectar, en el cas de sistemes automàtic d'extinció per aspersion, si està en funcionament.

En el cas del detector de fums els resultats no han estat tan concloents, en part perquè el sensor necessita un temps relativament llarg d'estabilització i la sortida depèn de factors com el gra d'humitat ambiental. Tot i així, sí que vàrem poder detectar, encara que sense quantificar-ho, una reacció davant del fum produït per una cigarreta.

Problemes trobats

Durant les diferents fases de desenvolupament, hem trobat diversos problemes que han fet endarrerir la planificació inicial. Dos dels més rellevants han estat relacionats amb l'entorn de desenvolupament i un altre amb el reconeixement de missatges (ACK) per les notes.

Entorn de desenvolupament

Pèrdua de comunicació amb les notes

Es va perdre la comunicació amb les notes. Després de cercar informació, vàrem veure que es tractava d'un bug documentat¹⁰² de la versió d'Ubuntu (11.10) i d'un dels drivers (*Braille*) que s'instal·la per defecte. Es va resoldre desinstal·lant el paquet de software corresponent.

```
sudo apt-get remove brltty.
```

Error generant els fitxer de missatges per a Java

El segon problema estava relacionat amb la utilitat *mig* i la versió del compilador nesC. El codi Java generat per *mig* era erroni, amb salts de línia i camps desconeguts que el feien inservible. Es va solucionar actualitzant la versió del compilador a la 1.3.3¹⁰³.

FireBase

Saturació de missatges

Aquest ha estat un dels problemes més importants del desenvolupament. Tot i haver triat fer servir modificar *BaseStation*, amb la seguretat que ens donava el fet d'estar distribuïda amb *TinyOS*, finalment l'elecció no ha estat encertada. Si tornéssim a començar el projecte, segurament optaríem per un desenvolupament complet d'aquest mòdul. En un principi pensàvem que es tractava d'un problema de les modificacions, o de la mida de les cues, però no ha estat així. Pel que hem vist, es tracta d'un problema del xip RF230 del que hem trobat nombroses referències¹⁰⁴.

¹⁰² <https://bugs.launchpad.net/ubuntu/+source/brltty/+bug/874181>

¹⁰³ http://TinyOS.stanford.edu/TinyOS/dists/ubuntu/pool/main/n/nesc/nesc_1.3.3-20110821_i386.deb

¹⁰⁴ <http://mail.millennium.berkeley.edu/pipermail/TinyOS-help/2011-February/049671.html>

El problema principal venia donat pel fet de que, encara que el missatge que arribava pel por sèrie s'enviava correctament per ràdio, a l'aplicació no es senyalava com a SUCCESS. Això generava un reintent que es podia convertir en un bucle sense fi, amb la saturació de les cues, etc.

Si fèiem servir *PacketLink*, el problema encara s'agreujava, ja que el nombre de reiteracions era encara més gran, sense arribar a aconseguir un ACK positiu:

```
call PacketLink.setRetries(msg, radioRetries);
call PacketLink.setRetryDelay(msg, radioDelay);
...
if (!call PacketLink.wasDelivered(msg)) // sempre és fals...
```

El mateix problema per *PacketAcknowledgements*:

```
call PacketAcknowledgements.requestAck(msg);

if (!call PacketAcknowledgements.wasAked(msg)); // sempre és fals...
```

Tampoc no podíem fer servir el flag RF230_HARDWARE_ACK sense eliminar el component *RadioSnoop*.

Finalment, tot i que no s'ha solucionat completament, hem aconseguit alleujar el problema amb la implementació del *WatchDog* que reinicia la mota si detecta un bucle i incloent els *flags* següents en el *make*:

```
CFLAGS+=-DSOFTWAREACK_TIMEOUT=1000
CFLAGS+=-DTASKLET_IS_TASK
```

FireRemote

Saturació de missatges

Degut al problema amb la saturació de *FireBase*, també teníem saturació de missatges en el node remot.

<http://www.mail-archive.com/TinyOS-help@millennium.berkeley.edu/msg36624.html>

<http://www.mail-archive.com/TinyOS-help@millennium.berkeley.edu/msg19509.html>

Ack dels missatges

Per les mateixes causes que en *FireBase*, hem tingut problemes per a obtenir resposta correcta dels mètodes *wasDelivered* de *PacketLink* i *wasAked* de *PacketAcknowledgements*.

Sensor de temperatura

Un cop s'assoleix un pic de temperatura (per exemple apropant la mota a una bombeta), al restablir-se la situació inicial, la mesura retornada pel sensor està aproximadament 2 °C per sobre de la registrada per un termòmetre de referència.

Leds

El sistema de notificació a l'usuari a través dels leds és limitat i és possible el solapament de les diferents alarmes.

Error al llegir el valor del PortG1

No hem aconseguir llegir l'estat correcte del PortG1. La comanda següent no torna el valor real:

```
enabled = call SinkPowerSensors.get()
```

L'hem solucionat amb una variable que conté l'estat dels sensors, tot i que no és el més correcte. Relacionat amb aquest punt, alguna vegada hem obtingut lectures errònies dels sensors ja que, encara que el port estigui deshabilitat, si accedim als canals ADC obtenim dades, encara que incorrectes (205°C, 5V, etc..).

Error en missatges amb diferent mida

En un principi es va pensar en fer servir un missatge curts, només amb id i codi d'operació, per a les sol·licituds de lectura dels sensors i una altres més complerts per tornar els valors. Aquest plantejament tenia com a resultat que la mota entrés en un mode indeterminat i finalment es reiniciava per acció del *watch dog*. Es va solucionar utilitzant missatges de la mateixa mida per la sol·licitud i per la resposta.

FireClient

Error d'accés a *thread* incorrecte

En accedir al diàleg per mostrar dades des de `FireClient.java` es produïa un error d'accés. L'hem resolt executant el codi de `FireClientDlg` al que volíem accedir de manera asíncrona:

```
Display.getDefault().asyncExec(new Runnable()
{
    public void run()
```

Error de *null pointer* a l'inici

Si la mota base ja estava rebent missatges en iniciar l'aplicació es podia produir un error al tractar d'accedir a un component de `FireClientDlg` que encara no està creat. Hem creat un *flag* de sincronització per no mostrar els missatges abans de que la interfície estigui totalment creada.

Error amb *SerialForwarder*

Si l'aplicació es crida amb paràmetres s'instancia la classe de `SerialForwarder`. Si es produeix un error en `BuildSource`, som capaços de capturar l'error de `PrintStreamMessenger` dins d'un *array* de bytes, però no tenim la possibilitat de mostrar el resultat ja que finalitzar l'execució.

```
phoenix = BuildSource.makePhoenix("sf@localhost:" + appPort,
    new PrintStreamMessenger(ps));
errString = ba.toString("UTF8");
```

Error en `module_path`: «`pixmap`»

Apareixia un warning a l'executar l'aplicació gràfica. Per resoldre'l s'instal·len els temes de `Gtk` amb la comanda:

```
sudo apt-get install gtk2-engines-pixbuf
```