

Sistema Contra Incendis

Jaume Roca Badia
Consultor: Jordi Bécares Ferrés
Gener 2012

Continguts

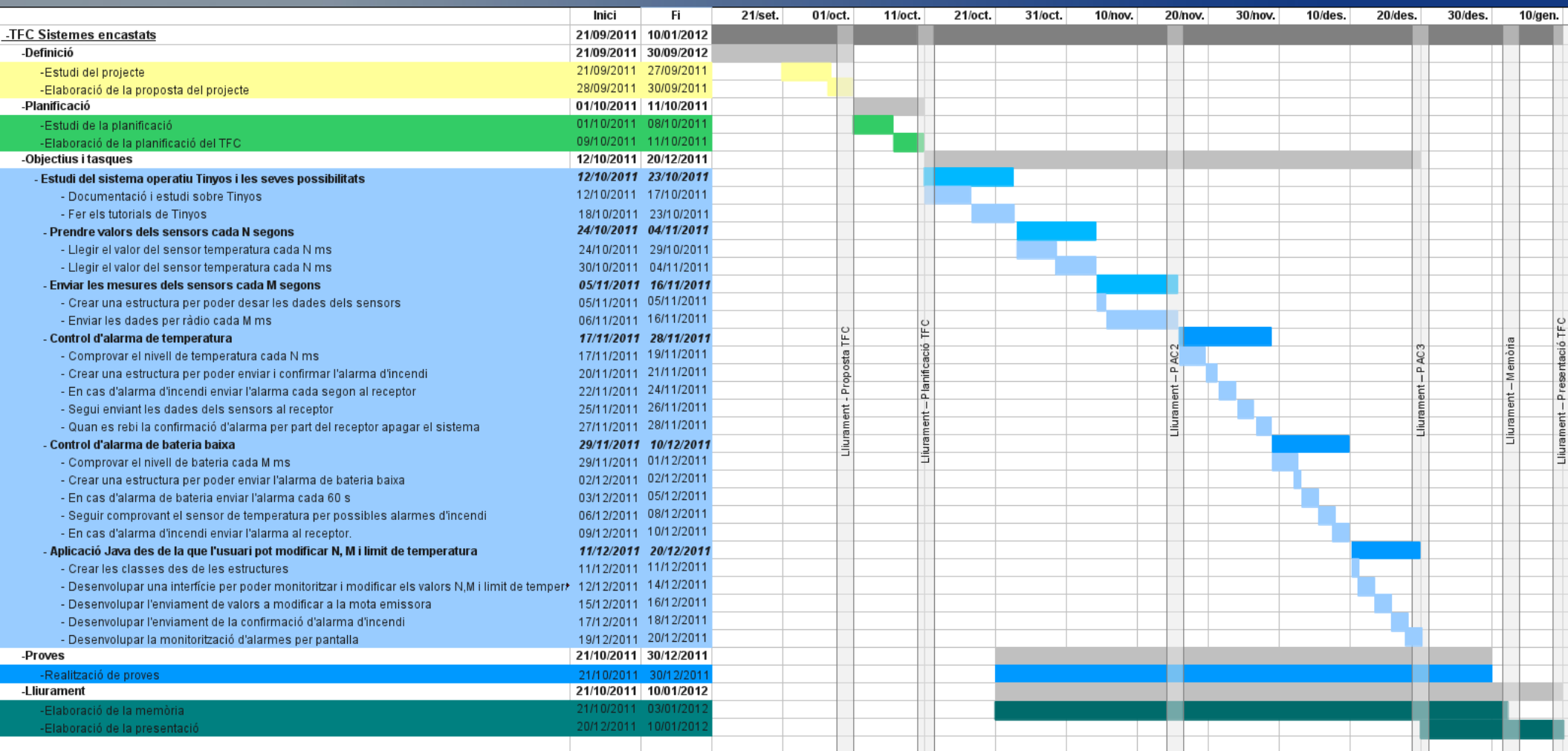
1. Introducció
2. Antecedents
3. Descripció funcional
4. Descripció detallada
5. Viabilitat tècnica
6. Conclusions

1. Introducció

En aquest projecte s'ha procedit a desenvolupar un Sistema Contra Incendis que compleix els objectius següents:

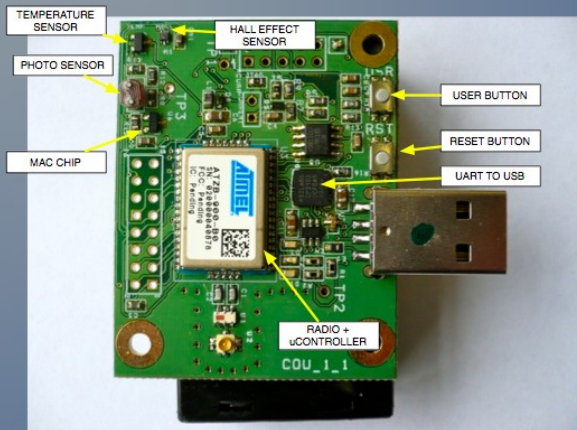
- Prendre mesures de temperatura i de bateria (per saber l'autonomia de la mota) cada N ms.
- Enviar les mesures al receptor cada M ms per monitoritzar-les.
- Si la temperatura supera un nivell s'envia l'alarma d'incendi cada 1s fins que l'usuari envii un missatge de confirmació de l'alarma
- Si la bateria baixa d'un nivell establert fixat en el codi, la mota envia una alarma cada 60s de nivell de bateria baix.
- L'usuari pot modificar N , M i el limit de temperatura des de l'aplicació receptora.

Planificació, diagrama de Gantt



Recursos de maquinari

Dues motes cou_1_2 24A2



Cable USB mascle/femella



Un PC

PC	
Processador	Intel Core i7 2600K
Dissipador	Prolimatech Black Series Megahalems
Ventiladores	Noiseblocker BlackSilentPRO PLPS 120x120x25 Noiseblocker BlackSilentPRO PLPS 120x120x25
Placa Base	Asus Sabertooth P67 B3
Targeta de Vídeo	MSI GeForce GTX580 Lightning 1536MB GDDR5 - VGA
RAM	Corsair Vengeance DDR3 1600 8GB CL9
Fuente	Seasonic X-760W Gold Modular 760W
Caixa	Silverstone Raven 3 RV03B
SSD	Crucial M4 128GB SATA3 MAESTRO Disco Duro SSD
HD	WD Caviar Black 1TB SATA3 64MB MAESTRO
Gravadora	Regrabadoras DVD Serial ATA ASUS DRW-24B3ST
Ctrl. Vent.	Scythe Kaze Master Ace 5.25" Frontal Multifunción

Piles AA



Recursos de programari

Arch Linux x86_64: Sistema Operatiu del PC

TinyOS: Sistema Operatiu per les motes

Eclipse amb Yeti2: Editor de programació

Meshprog: Aplicació per carregar els programes compilats a les motes.

SerialForwarder: Aplicació que ens permet comunicar-nos des del PC amb la mota.

NesC: Llenguatge de programació utilitzat per programar l'aplicació de la mota.

UMLet: Aplicació amb la que s'han fet diferents diagrames UML de l'aplicació.

Productes obtinguts

- Programari mota emissora

Aplicació *FireProtectionSystem*: Desenvolupada en NesC amb el sistema operatiu TinyOS, gestiona el sistema contra incendis en la mota emissora, controla les possibles alarmes de temperatura i bateria baixa així com enviar els valors dels sensors a la mota receptora.

- Programari PC receptor

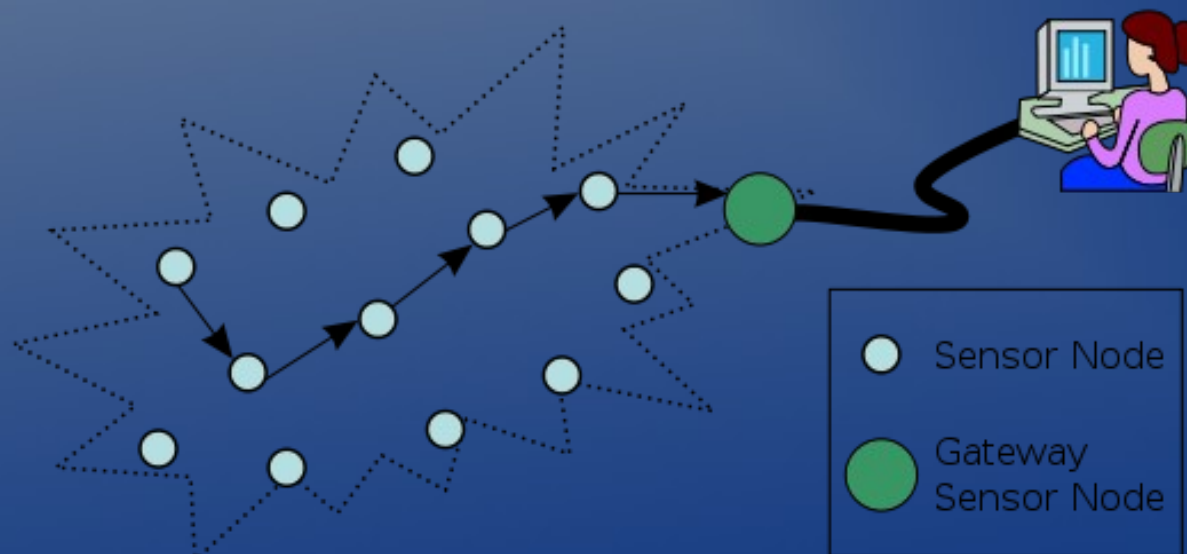
Aplicació *FireProtectionSystemJava*: Aquesta aplicació està desenvolupada amb Java i gràcies als components de TinyOS per Java MotIF i MIG i amb l'ajuda de l'aplicació SerialForwarder és capaç de comunicar-se amb la mota connectada al PC per poder assolir els objectius exposats.

2. Antecedents

Estat de l'art

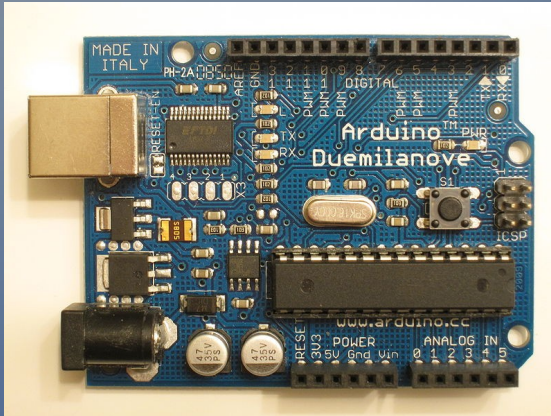
- Xarxes de sensors sense fils

Una xarxa de sensors sense fils (*WSN wireless sensor network*) consisteix a distribuir nodes autònoms que disposen d'una sèrie de sensors de baix consum amb els que es poden monitoritzar diferents condicions físiques o mediambientals, (temperatura, so, vibració, pressió de l'aire, etc..). Aquestes dades s'envien de forma cooperativa mitjançant la xarxa al node principal que es l'encarregat de rebre-les i comunicar-les normalment a un PC.



- Motes

Una mota és un node d'una xarxa de sensors sense fil que és capaç de realitzar algun tipus de processament, recopilació de la informació sensorial i comunicar-se amb altres nodes connectats a la xarxa. Ex:



Atmel AVR Atmega328



Btnode



Epic Mote

- **Sistemes operatius per a motes**

Són sistemes operatius dissenyats per funcionar sota les restriccions de les xarxes de sensors sense fils com son el baix consum, poca memòria,...etc. Hi ha varietat de sistemes operatius per a motes:

Contiki, Erika Enterprise, Nano-RK, TinyOS, LiteOS.

- **Estàndards de comunicació per xarxes de sensors sense fils**

WirelessHART: És una tecnologia de xarxa de tipus malla que opera a la banda de ràdio ISM (Indústria, Científica i Mèdica) de 2,4GHz.

ISA100.11a: Té per objectiu proporcionar comunicació sense fils fiable i segura per aplicacions de monitorització no crítica, alerta i control de supervisió.

IEEE 1451: És un grup de normes de caràcter universal i obert per estandarditzar sensors intel·ligents tenint en compte tres característiques fonamentals: Integració de la intel·ligència propera al punt de mesura, capacitat de còmput i interfícies digitals estàndards per la transferència de dades.

ZigBee: És el nom de l'especificació d'un conjunt de protocols d'alt nivell de comunicacions sense fils per la seva utilització en radiodifusió digital de baix consum, basada en l'estàndard IEEE 802.15.4 de xarxes sense fils d'àrea personal (WPAN).

IEEE 802.15.14: És un estàndard que defineix el nivell físic i el control d'accés al medi de xarxes sense fils d'àrea personal amb tasses baixes de transmissió de dades. El propòsit d'aquest estàndard és definir els nivells de xarxa bàsic per donar servei a un tipus específic de xarxa sense fils d'àrea personal centrada en l'habilitació de comunicació entre dispositius de baix cost i baixa velocitat de transmissió.

Estudi de mercat

Hi ha moltes empreses que es dediquen a donar seguretat als edificis per exemple:

<http://www.aguilera.es/> <http://www.icima.es>

El sistema desenvolupat en aquest projecte es cert que funciona però per poder-lo portar al mercat falta madurar-lo bastant en el sentit de poder instal·lar-lo en un edifici complert amb el que això implica, un desplegament d'una xarxa important així com unes comunicacions fiables entre els diferents nodes, també s'hauria de fer més robust a les col·lisions que produeix el medi.

Per tant cal destacar que el sistema encara que funciona està encara en fase de millora i maduració.

3. Descripció funcional

FireProtectionSystem (mota emissora):

S'encarrega d'assolir els objectius següents:

- Prendre mesures de temperatura i de bateria (per saber l'autonomia de la mota) cada N ms.
- Enviar les mesures al receptor cada M ms per monitoritzar-les.
- Si la temperatura supera un nivell s'envia l'alarma d'incendi cada 1s.
- Si la bateria baixa d'un nivell establert fixat en el codi, la mota envia una alarma cada 60s de nivell de bateria baix.

FireProtectionSystemJava (PC):

S'encarrega d'assolir els objectius següents:

- L'usuari pot modificar N, M i el limit de temperatura des de l'aplicació receptora.
- Enviar el missatge de confirmació d'alarma d'incendi

Diagrama de seqüencia del sistema

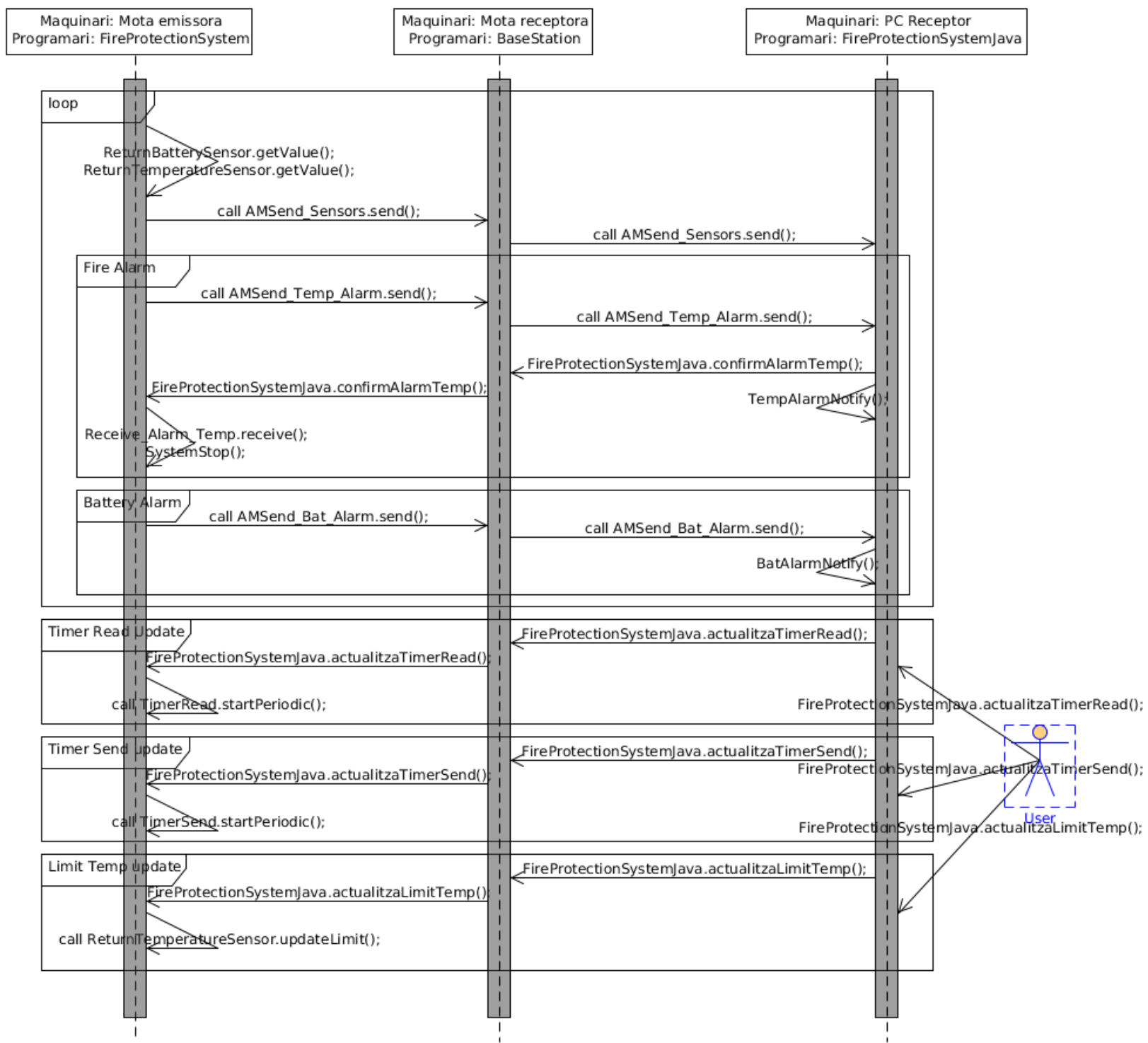


Diagrama de blocs del sistema

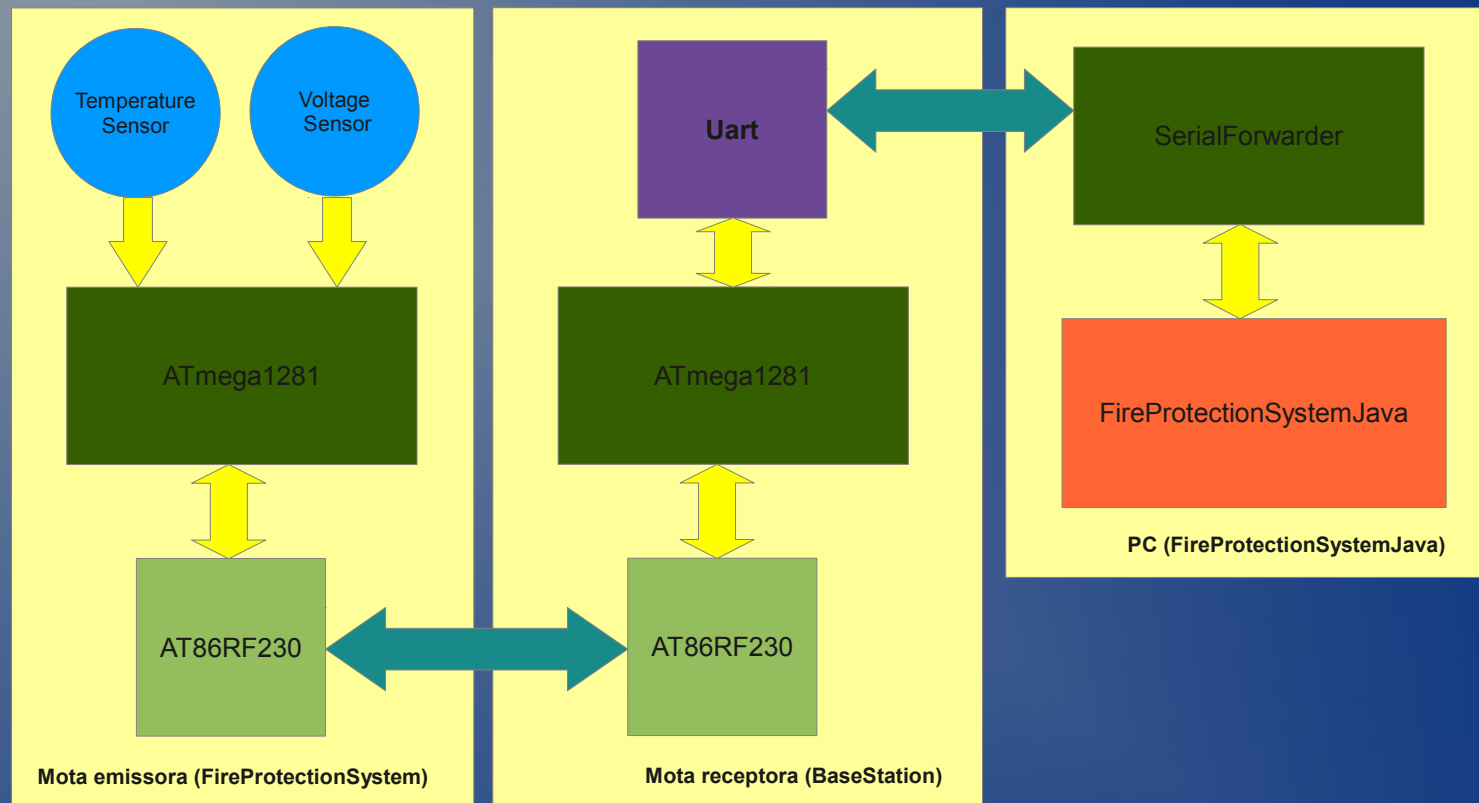


Diagrama d'activitats, FireProtectionSystem

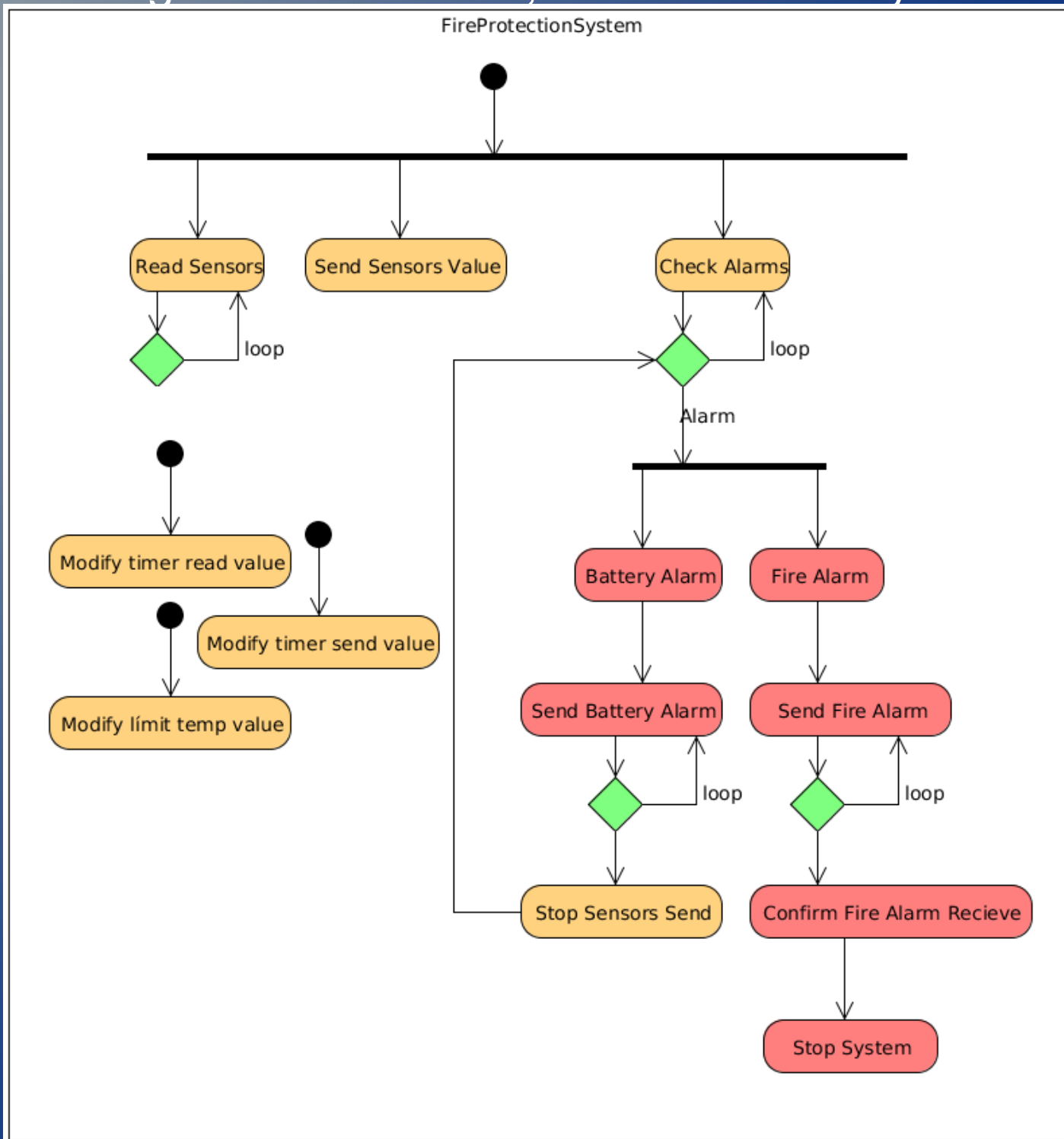
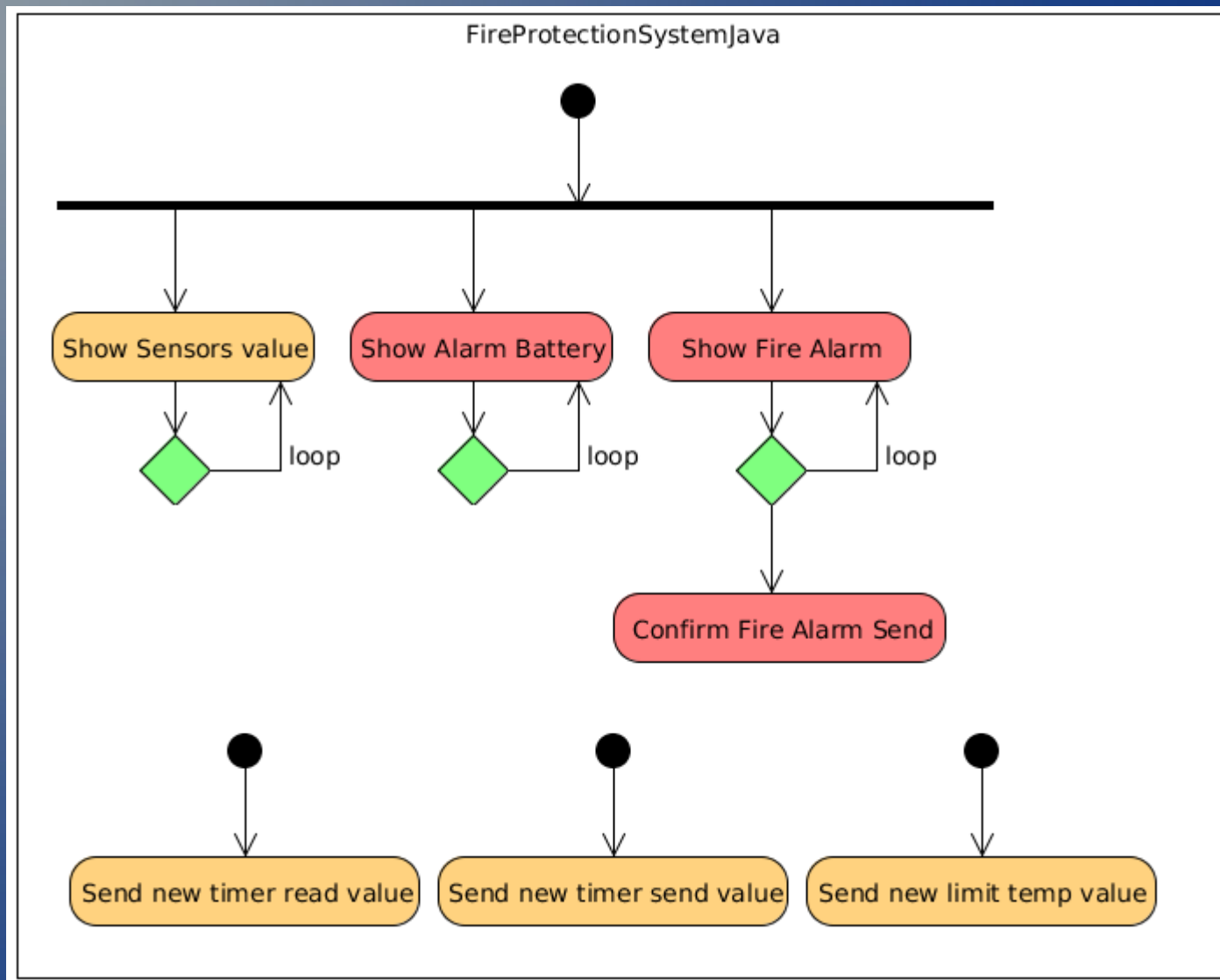


Diagrama d'activitats, FireProtectionSystemJava



4. Descripció detallada

FireProtectionSystem:

- TimerRead: Per cada iteració d'aquest timer i si l'estat de l'aplicació es READY o READ_TEMP es farà una lectura dels sensors i a més es comprovarà si s'ha produït alguna alarma d'incendi o de bateria.
- TimerSend: Per cada iteració s'envia un missatge a la mota receptora amb els valors dels sensors de temperatura i de bateria.
- TimerSendBatAlarm: Per cada iteració si es produeix l'alarma de bateria baixa s'envia un missatge a la mota receptora indicant-la.
- TimerSendTempAlarm: Per cada iteració si es produeix l'alarma d'incendi s'envia un missatge a la mota receptora indicant-la.
- Receive_Timer_Read.receive: Quan es rep el nou valor del timer TimerRead es procedeix a modificar-lo.
- Receive_Timer_Send.receive: Quan es rep el nou valor del timer TimerSend es procedeix a modificar-lo.
- Receive_Limit_Temp.receive: Quan es rep el nou valor del límit de temperatura es procedeix a modificar-lo.
- Receive_Alarm_Temp.receive: Quan es rep el missatge de confirmació d'alarma d'incendi es procedeix a parar el sistema.

Component principal de l'aplicació

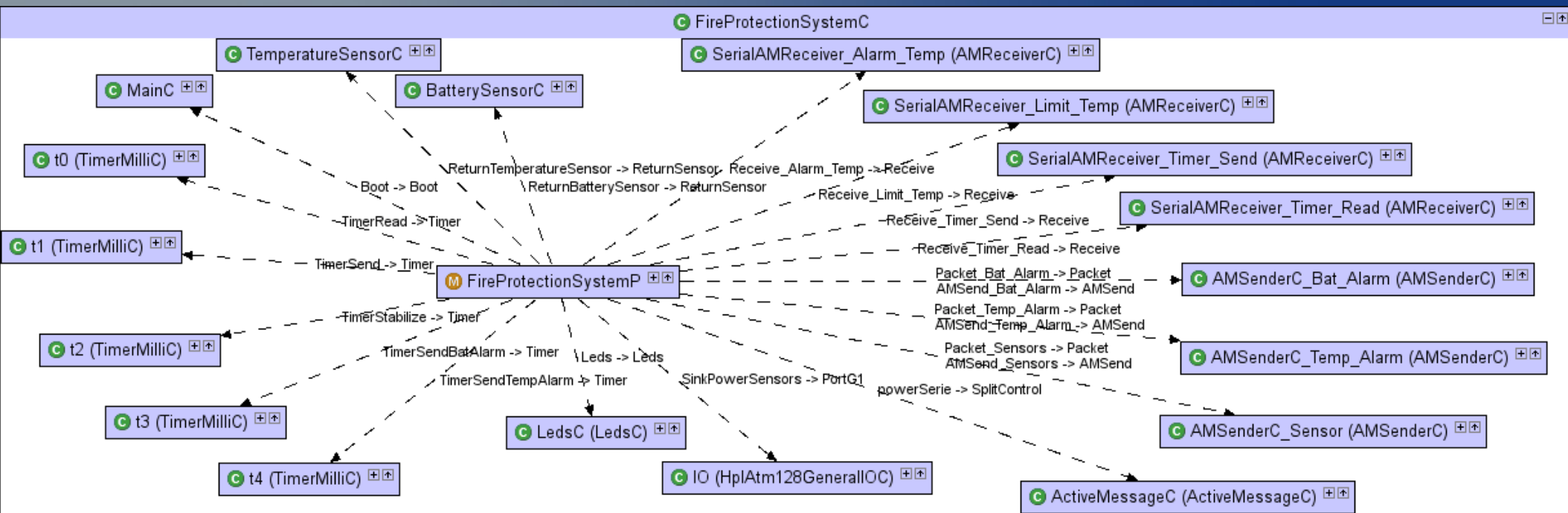


Diagrama de connexions

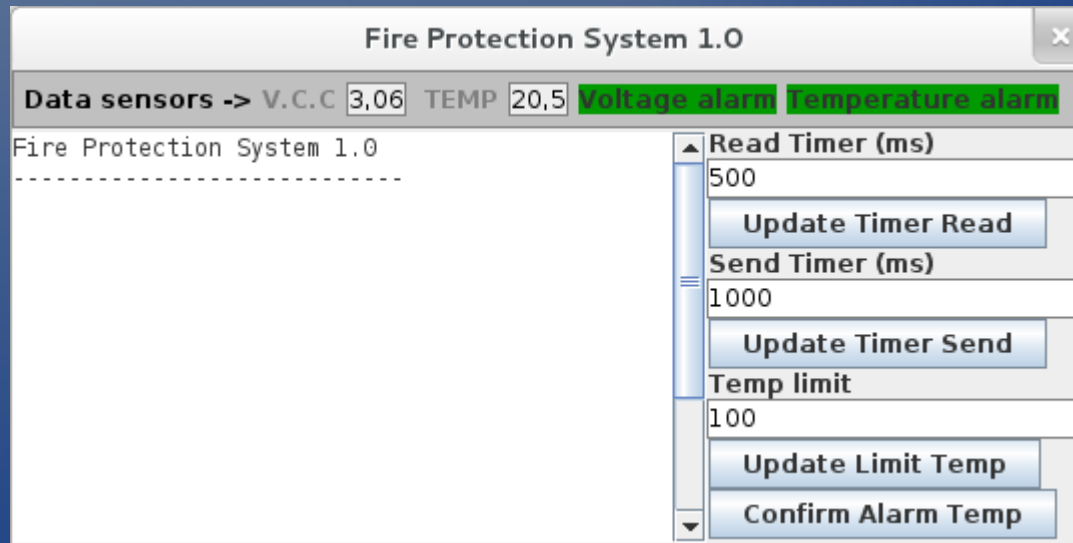
Descripció dels components

- FireProtectionSystemP: Component on hi ha la gestió principal de l'aplicació.
- MainC: Fem servir aquest component perquè ens proporciona la interfície Boot necessària per iniciar l'aplicació.
- TimerMilliC: Necessari per poder crear els bucles d'iteració (timers) amb els que llegirem els valors dels sensors, enviarem els valors dels sensors, sincronitzarem l'encesa dels sensors i enviarem les alarmes de bateria baixa i d'incendi.
- LedsC: Necessari per poder interactuar amb els leds de que disposen les motes, encendre i parar els leds.
- HplAtm128GeneralIOC: Necessari per poder encendre i parar els sensors.
- ActiveMessageC: Necessari per poder encendre i parar la ràdio.
- AMSenderC: Necessari per poder enviar missatges per ràdio.
- AMReceiverC: Necessari per poder rebre missatges per ràdio.
- BatterySensorC: Component desenvolupat per nosaltres amb el que agafem el valor del sensor de bateria i controlem si hi ha una alarma de bateria.
- TemperatureSensorC: Component desenvolupat per nosaltres amb el que agafem el valor del sensor de temperatura i controlem si hi ha una alarma d'incendi.

FireProtectionSystemJava

Ens ajudem de les eines següents:

- **SerialForwarder**: Aplicació que ens permet comunicar-nos des del PC amb la mota.
- **MoteIF**: Llibreria proporcionada per TinyOS amb la que podem rebre i enviar missatges a la mota connectada al **socket** que ha creat l'aplicació **SerialForwarder**.
- **mig**: (*Message Interface Generator*) llibreria proporcionada per TinyOS amb la que es poden crear classes Java a partir de les estructures definides en NesC.



5. Viabilitat tècnica

Aquest projecte ara per ara no es podria comercialitzar, faltarien una sèrie d'objectius:

- Fer el sistema de comunicació sense fils més robust a les col·lisions produïdes pel medi, implementant un sistema d'acks.
- Modificar el codi per poder monitoritzar més d'una mota a la vegada.
- Desenvolupar un sistema d'interconnexió entre múltiples nodes, perquè per exemple es pogués monitoritzar diferents sales d'una planta d'un edifici.
- **Punts forts:** Les eines utilitzades pel seu desenvolupament, Java i TinyOS ja que actualment són projectes mantinguts i en fase emergent, així ens assegurem en un futur que aquest projecte no estigui desfasat.
- **Punts febles:** El punts febles serien tots els objectius que falta per implementar.

6. Conclusions

- He estudiat una nova àrea, Sistemes Encastats i xarxes de sensors sense fils.
- He après el funcionament de TinyOS, NesC, SerialForwardes, meshprog.
- Projecte encara en fase de desenvolupament i millora.
- En aquesta àrea es poden fer infinitat de projectes.
- La organització i planificació són molt importants