

UNIVERSITAT POLITÈCNICA DE MANRESA

Departament de Disseny i Programació de Sistemes Electrònics


Treball de fi de Grau:

SMART LIGHTING

Realitzat per Cristian Subirana Fidone per assolir el Grau
d'Enginyeria de Sistemes TIC

Sota la direcció de:
Dr.Eng Jordi Bonet Dalmau

Autorització del Treball Final de Grau



Escola Politècnica Superior
d'Enginyeria de Manresa
UNIVERSITAT POLITÈCNICA DE CATALUNYA

NORMATIVA DEL TFG DE L'EPSEM
Emplenar per l'estudiant/a

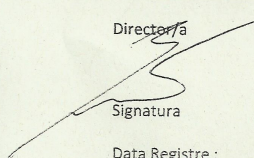
ANNEX 1.- AUTORIZACIÓ DE LA MATRÍCULA DEL TFG EN MODALITAT A o B

ESTUDIANT/A: CRISTIAN SUBIRANA FIDONE Núm. identificatiu 39399778-6
 GRAU EN: ENGINYERIA EN SISTEMES TIC
 DIRECTOR/A DEL TFG: JORDI BONET DALMAU
 DEPARTAMENT: DIPSE
 REALITZACIÓ EN ANGLÈS: Sí No
 MODALITAT: A B

TÍTOL: SMART LIGHTING

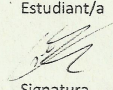
DESCRIPCIÓ:
Realització d'una maqueta sobre il·luminació intel·ligent controlada remotament amb la capacitat de regular-ne la seva potència.

SOL·LICITUD PER A SER AVALUAT DE LES COMPETÈNCIES GÈNERIQUES EN NIVELL 3 (si no estan assolides)	Sí	No
1.- Emprenedoria i innovació	<input type="checkbox"/>	<input checked="" type="checkbox"/>
2.- Sostenibilitat i compromís social	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.- Tercera Llengua	<input type="checkbox"/>	<input checked="" type="checkbox"/>
4.- Comunicació eficaç oral i escrita	<input checked="" type="checkbox"/>	<input type="checkbox"/>
5.- Ús solvent dels recursos d'informació	<input type="checkbox"/>	<input checked="" type="checkbox"/>
6.- Apreneatge autònom	<input checked="" type="checkbox"/>	<input type="checkbox"/>
7.- Treball en equip	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Director/a

Signatura

Co-director/a
(si s'escau)

Signatura

Estudiant/a

Signatura

Data Registre : _____ (validesa: un any des de la data del registre)

3 Exemplars: Director/a / Secretaria de l'Escola / Estudiant/a per incorporar al TFG

* Normativa Transitòria fins que l'aplicatiu PRISMA permeti la gestió telemàtica del TFG
 (Document aprovat per la Comissió Permanent de 5 de juliol de 2012, i per la Junta de Centre de 12 de juliol de 2012)

Figura 1: Autorització TFG

Abstract

This project deals with the monitoring and control of a lighting system to be able to keep track in real time what it is doing and what its current operation. The aim of this project is to manage and control the consumption and conditions of the system maintaining a proper functioning from a set of elements.

The system consists of a prototype of three lamps, which represent the complete system. This lamps communicates between them and finally a web system in order to appreciate this operation and have a control over lamps .

To reduce their consumption system has reduced his state of rest to 50 of power capacity and when the system detected a body works at 100 power capacity. Unlike the other systems, this system don't works with full power capacity along the day.

The main issues addressed in this work are:

- Communication between nodes or lampposts and coordinators.
- Implementation of a web system in order to control and monitor the lighting system.
- Create a system of effective detection lights to see if someone has been detected.

Donar les gràcies a tot el departament de Disseny i Programació de Sistemes Electrònics per la seva dedicació i l'ajuda que m'han presentat al llarg del Grau.

Agraïments

Primerament vull donar les gràcies al meu tutor de treball de fi de grau, el Dr. Jordi Bonet Dalmau, per la seva dedicació, tracte i ajut que m'ha ofert alhora d'enfocar el projecte, així com resoldre dubtes i problemes crítics.

A més de mostrar un gran interès per resoldre els problemes, buscar informació sobre el treball, m'ha orientat i proposat diferents punts de vista quan m'he trobat encallat.

Agraeixo l'ajuda a en Joan Martínez i a l'Albert Babí del laboratori del departament de Disseny i Programació de Sistemes Electrònics per deixar-me material per al projecte, a més d'ajudar-me a entendre el funcionament d'aquest material.

Donar gràcies a Ferran Sarrió de l'empresa SpaceBits per l'ajuda en la resolució de problemes relacionats amb servidors i bases de dades.

Finalment mostrar el meu agraïment al meu tutor de pràctiques, Jordi Mitjana per ajudar-m'he a entendre molts conceptes relacionats amb el projecte així com el tracte que m'ha ofert.

Resum

En aquest document es presenten totes les eines necessàries per a desenvolupar un sistema amb la capacitat de controlar i monitorar un sistema d'il·luminació intel·ligent a partir d'elements capaços de comunicar-se entre ells sense cablejat, i tenint un control i visió de les dades via servei web.

El principal objectiu del projecte presentat, és oferir la capacitat d'administrar i tenir un control sobre un sistema d'il·luminació destinat per a entitats que els interessi reduir costos energètics així com tenir control sobre el sistema d'il·luminació.

S'ha realitzat un prototip a petita escala del sistema complet d'il·luminació a fi de demostrar el seu funcionament així com presentar-lo com una alternativa d'estalvi energètic.

El prototip realitzat no presenta totes les característiques del sistema final o posterior a aquest projecte, ja que el sistema final, previst com a continuació de la maqueta, per dimensions, cost i per alimentació elèctrica d'aquest, han fet impossible aplicar-les al prototip. Tot i així al projecte és el pas prèvi de cara al resultat final del sistema.

Al llarg de projecte s'ha hagut de realitzar una important recerca de llenguatge de programació no coneguts o poc utilitzats com són PHP, HTML, LaTeX, CSS i Javascript. Aquest aprenentatge s'ha fet de manera autònoma, ja que en els temps en què ens trobem, podem trobar tot allò que necessitem per internet.

Paraules clau: Smart Lighting, il·luminació intel·ligent, comunicació sense fils, estalvi energètic.

Índex

0.2	Introducció	11
1	Situació actual	12
1.1	Introducció	13
1.2	Entorn de treball del sistema	13
1.3	Requisits del sistema	14
1.4	Impacte ambiental	15
1.5	Sistemes semblants al proposat	16
2	Estructura del sistema	18
2.1	Introducció	19
2.2	Descripció conceptual	19
2.3	Bloc I. Sistema de Comunicació	20
2.3.1	Introducció	20
2.3.2	Zigbee	20
2.3.3	Sèries Xbee	23
2.3.4	Modes de funcionament d'un Xbee	24
2.4	Bloc II. Hardware	31
2.4.1	Introducció	31
2.4.2	Arduino Uno R3	31
2.4.3	Arduino Shield	32
2.4.4	Sensor PIR	33
2.4.5	USB Explorer	34
2.4.6	LED's i resistències	35
2.4.7	Muntatge del Hardware	36
2.5	Bloc III. Sistema Web	37
2.5.1	Introducció	37
2.5.2	Base de dades	37
2.5.3	Interfície web	39

3	Configuració i Programació del sistema d'il·luminació	40
3.1	Introducció	41
3.2	Configuració del Sistema de comunicació utilitzant el programa X-CTU	41
3.2.1	Configuració feta per a End Device AT, Coordinador API	51
3.3	Configuració hardware	54
3.3.1	Programació programa principal carregat al ordinador.	61
3.3.2	Programació del programa carregat als nodes	69
3.4	Configuració del Sistema Web	76
3.4.1	Programació de la interfície web	76
3.4.2	Programació de la Base de dades	84
4	Conclusions	87
5	Annex	90

0.2 Introducció

La gran majoria d'estudiants d'avui dia trien projectes de fi de grau molt relacionats amb temes que es tracten al llarg del curs o bé trien projectes proposats per la mateixa universitat. S'hauria d'animar als estudiants a triar temes relacionats amb possibles aplicacions del grau en un futur, o bé, temes que aportin a l'estudiant uns coneixements que durant el grau no han vist, però que estan relacionats amb el grau.

D'aquesta manera l'alumne es forma amb més coneixements, tindrà més experiència en crear i desenvolupar projectes que hagi pensat per ell sol que havent triat un tema pròpi de l'escola, potenciant aspectes com l'organització, l'emprenedoria i la innovació.

Ens trobem en una situació global on els estudiants hauran de plantejar mètodes i sistemes capaços de reduir costos, estalviar la quantitat més gran de recursos possible, crear d'acord amb l'estalvi energètic.

Aquests plantejaments que s'hauran de preparar els estudiants són deguts en gran part a la demanda d'energia i recursos que segueix creixent cada any, i que si no se'n prenen un seguit de mesures per estalviar energèticament, podem trobar-nos amb un esgotament de recursos o que la demanda sobrepassi el que som capaços de produir

A partir dels plantejaments i preocupacions anteriors va sorgir la idea de dur a terme aquest projecte per tal de poder proposar una solució a entitats, ajuntaments i organitzacions motivades per l'estalvi energètic.

Actualment l'enllumenat públic de l'estat espanyol suposa aproximadament un 12 per cent de l'energia total generada, aproximadament 5.370 GWh/any a Espanya, contant des de 2010 amb 7.965.000 punts de llum en tot l'estat [1], dels quals la gran majoria es basa en sistemes antiquats i poc eficients.

Capítol 1

Situació actual

1.1 Introducció

En aquesta part es tractaran temes com l'entorn on treballa el sistema a dissenyar, els requisits necessaris, l'impacte que comportarà dit projecte en l'ambient i sistemes ja creats per a satisfer necessitats molt semblants a les d'aquest projecte.

1.2 Entorn de treball del sistema

Hem de considerar que aquest projecte s'ha orientat a treballar a l'exterior, pel que refereix al sistema final o a continuació d'aquest projecte, concretament en aquest projecte s'ha enfocat a parcs i jardins, el qual són llocs amb un reduït volum de persones passejant-hi respecte a un carrer normal i corrent.

A més a més, l'estalvi energètic s'apreciarà millor en una zona poc visitada, ja que el sistema en estat de repòs o que no detecta ningú, funciona al 50 per cent de capacitat, mentre que en detectar una persona, funciona a ple rendiment, elevant-ne així el seu consum.

S'ha de tenir en compte que el sensor de presència ha de tenir una sensibilitat tal que no s'encengui en detectar animals o fulles, ja que això podria generar un augment en el consum esperat. Un altre aspecte a tenir en compte són els danys que poden ocasionar les persones o el mateix temps, per tant es tindrà en compte que l'estructura que protegirà el sistema sigui robusta i de fàcil instal·lació.

Perquè el sistema arribi a ocupar tota una zona, es necessitarà una xarxa de nodes o dispositius finals, el qual s'encarregaran de captar les dades dels sensors de presència, actuar en cas de detecció i comunicació del seu estat al coordinador. El coordinador tindrà la tasca de gestionar les dades rebudes de cada node a més de comunicar les dades gestionades, amb el sistema web. Aquest sistema web mostrarà en temps real l'estat de cada node, permetrà actuar sobre cada node i oferirà dades sobre el consum del sistema en global.

L'entorn en que es trobarà el prototip, no tindrem en compte tots els fets anteriors, simplement els tindrem en compte per a la continuació d'aquest projecte. El prototip realitzat funciona en espais tancats com per exemple una aula, on no hi considerem cap adversitat que pugui afectar negativament el prototip.

1.3 Requisits del sistema

El sistema el podem dividir en tres blocs, el primer fa referència a la comunicació entre elements, el segon al hardware necessari i el tercer a la part del sistema web.

En el primer bloc s'ha considerat prioritzar els sistemes de comunicació per radiofreqüència per sobre de sistemes de comunicació per fils, ja que estalviàrem el cost del cablejat i d'haver de perforar el terreny per fer passar tots els fils.

S'ha de tenir en compte que la comunicació entre nodes pot ser a una distància destacable, per tant, el dispositiu a utilitzar ha de ser capaç de mantenir una transmissió de dades eficient i ràpida entre punts de comunicació. També hem de considerar que en tot sistema de comunicació per radiofreqüència hi ha d'haver seguretat, ja que podrien interferir els paquets que contenen informació i manipular-los, per tant el dispositiu a utilitzar ha de permetre xifrar o codificar els missatges.

En el segon bloc hi trobarem tot el referent als components físics i actuadors, el qual hauran de ser el més compactes possible a fi d'ocupar el menor espai possible. Aquests components seran els encarregats de la interacció amb els usuaris i els que se n'encarregaran de processar els missatges per tal de ser enviats pel sistema de comunicació.

L'últim bloc estarà format pel sistema web, el qual tractarà els missatges rebuts i els presentarà adequadament en la web permetent així la visió de les dades per part de l'administrador o l'usuari encarregat de tractar les dades, com també haurà de poder generar missatges.

Aquest sistema web haurà de tenir seguretat davant d'usuaris no desitjats que vulguin accedir al monitoratge i control de les dades, on només l'administrador o usuari encarregat de la web podrà veure.

Tots aquests requisits s'esperen que es compleixin en el prototip, en què, en la part d'explicació del hardware utilitzat i en la configuració del sistema, es comentaran els requisits que s'han aconseguit realitzar.

1.4 Impacte ambiental

Si considerem els impactes que pot generar un sistema d'il·luminació ens trobem hi han diversos que mereixen ser considerats, el qual poden tenir un impacte negatiu amb l'ambient que envolta el sistema.

Podem considerar molèsties referents a la visió provocades per l'enlluernament d'adaptació, que poden acabar generant fatiga mental, o estrès. Tambè alteracions en el son, trastorns en el rellotge biològic de les persones, canvis de comportaments, etc.

Cal tenir en compte que la il·luminació pot atraure insectes, alterar la natura que l'envolta així com els animals i les plantes.

Per tant tot aquest seguit de factors s'han de tenir en compte alhora de dissenyar un sistema eficaç i respectuós amb l'ambient que l'envolta per tal de minimitzar el màxim possibles els seus efectes negatius.

Els impactes ambientals esmentats fan referència al sistema final ha realitzar. En el prototip aquests aspectes no s'han considerat, ja que, com s'ha esmentat anteriorment l'ambient en que es trobarà aquest prototip serà molt semblant a l'ambient de dins d'una aula.

1.5 Sistemes semblants al proposat

En el mercat relacionat amb els sistemes d'il·luminació intel·ligent ens trobem que moltes empreses s'han interessat en sistemes d'aquest tipus amb l'objectiu de tenir un control més eficaç per tal de generar un estalvi energètic.

Per exemple podem trobar empreses com Grupo Schröder [2] amb la gamma de solucions de control intel·ligent Owlet, el qual es presenta com un sistema capaç de reduir fins a un 80 per cent el consum a partir d'un sistema basat en sensors de moviment i de velocitat, juntament amb tecnologies modernes que permeten tenir-ne un control i visió del seu estat.

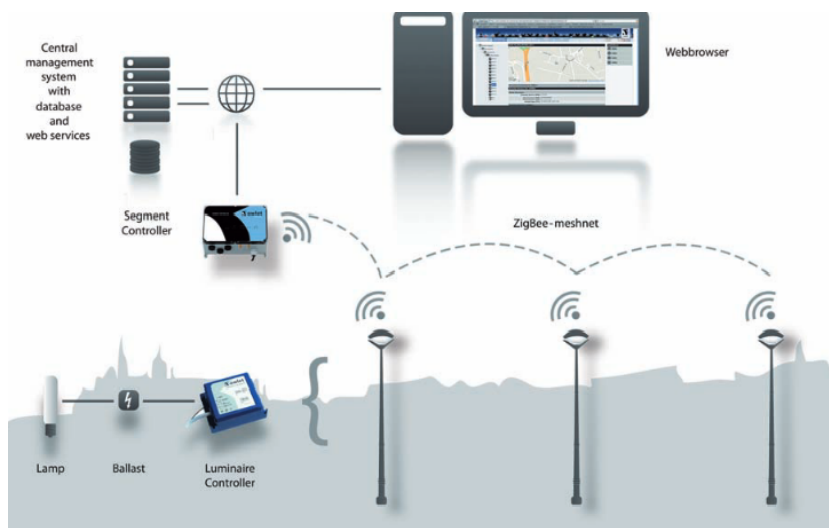


Figura 1.1: Owlet

Troblem d'altres empreses com Luix [3], en que el seu sistema es basa en uns fanals que representen un node a la xarxa amb un ordinador encastat encarregat de gestionar aquests nodes amb tecnologia 3G permetent així la transmissió de dades entre nodes.

Per últim esmentar l'empresa Philips [4], amb el sistema Lumimotion, capaç d'adaptar-se a qualsevol entorn, permetent regular la il·luminació de les fanals a partir d'una gestió a distància, el qual es basa en les condicions meteorològiques, hora del dia, volum de persones al voltant entre altres aspectes. Es basa en sensors de presència i òptics juntament amb una comunicació sense fils.



Figura 1.2: Philips

Capítol 2

Estructura del sistema

2.1 Introducció

En aquesta secció es tractaran específicament tots els materials emprats en el projecte de cada bloc, el qual en l'apartat anterior de 'Requisits del sistema' sen'ha fet un breu resum de quin tipus d'elements haurien de ser els escollits. Es farà una explicació de cada element així com el motiu pel qual ha estat triat, la seva configuració i la seva funcionalitat. Aquestes explicacions es realitzaran dins dels següents blocs de continuació.

2.2 Descripció conceptual

El sistema estarà format per un seguit de nodes o "punts finals" que generaran els missatges a transmetre amb informació del seu estat. Aquests nodes per radiofreqüència es comunicaran amb el node coordinador, l'encarregat d'administrar i gestionar l'informació rebuda dels "punts finals", també es comunicarà amb el sistema web per tal de donar-li les dades dels nodes. El sistema web tindrà la funció de tractar les dades rebudes del coordinador i les mostrarà en una interfície web el qual podrà interactuar amb els "punts finals" per mitjà del coordinador.

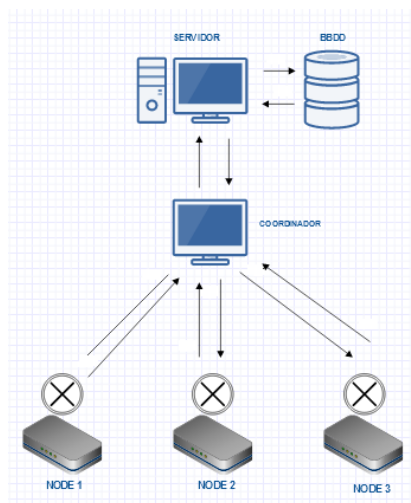


Figura 2.1: Esquema del sistema

2.3 Bloc I. Sistema de Comunicació

2.3.1 Introducció

Aquest bloc tractarà del dispositiu utilitzat per a la comunicació del sistema entre dispositius. També es tractarà les sèries que podem trobar al mercat i els modes disponibles de funcionament.

2.3.2 Zigbee

El sistema encarregat de la comunicació el formen un conjunt de Xbee's, el qual són mòduls integrats que permeten la comunicació entre nodes i dispositius. Aquests dispositius utilitzen el protocol de xarxa anomenat Zigbee basat en l'estàndard IEEE 802.15.4 el qual permet crear xarxes FAST POINT-TO-POINT (punt a multipunt) o bé, per a xarxes PEER-TO-PEER (punt a punt). Aquests mòduls resulten molt útils per a sistemes en què pot haver-hi un volum de dades gran, amb baixa latència i una sincronització de comunicació predicable.

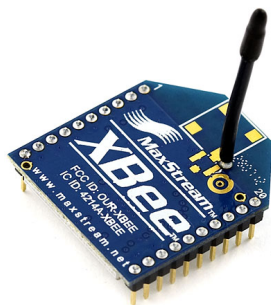


Figura 2.2: Imatge mòdul Xbee

A més a més, aquests mòduls Xbee són molt compactes, permeten una comunicació a distàncies considerables de fins a 200 metres, tenen un consum molt reduït, poden funcionar durant dos anys amb un parell de piles normals. A continuació es mostra la comparativa entre aquest mòduls de comunicació i altres mètodes de comunicació com Wifi o Bluetooth:

Market Name	ZigBee®	---	Wi-Fi™	Bluetooth™
Standard	802.15.4	GSM/GPRS CDMA/1xRTT	802.11b	802.15.1
Application Focus	Monitoring & Control	Wide Area Voice & Data	Web, Email, Video	Cable Replacement
System Resources	4KB - 32KB	16MB+	1MB+	250KB+
Battery Life (days)	100 - 1,000+	1-7	.5 - 5	1 - 7
Network Size	Unlimited (2 ⁶⁴)	1	32	7
Maximum Data Rate (KB/s)	20 - 250	64 - 128+	11,000+	720
Transmission Range (meters)	1 - 100+	1,000+	1 - 100	1 - 10+
Success Metrics	Reliability, Power, Cost	Reach, Quality	Speed, Flexibility	Cost, Convenience

Figura 2.3: Comparació entre tecnologies

Les característiques de les xarxes Zigbee són:

1. Velocitat de transmissió entre 25 i 250 kbps.
2. Protocol asíncron, half duplex i estandarditzat, permetent que productes de diferents fabricants puguin funcionar junts.
3. Poden estar formats des de dos dispositius fins a cents.
4. Treballen a una freqüència de 2.4 GHz i també de 868 MHz i 915 MHz.
5. La xarxa pot canviar de canal de forma dinàmica en cas que apareguin interferències.
6. És un protocol segur, ja que es pot implementar encriptació i autenticació.

Considerant les característiques anteriors podem apreciar que aquests tipus de mòduls s'adapten perfectament a les nostres necessitats, ja que, utilitza mòduls compactes (Xbee). Té un consum molt baix en comparació amb la seva competència, permet la comunicació a distància a més d'oferir una comunicació segura mitjançant la possibilitat d'encriptar els paquets. Les característiques esmentades són un recull de les més importants que s'han considerat per al projecte, ja que n'hi han moltes més.

Els mòduls Xbee es basen en el protocol Zigbee. Aquest protocol el qual està basat en el protocol IEEE 802.15.4, té com a objectiu assegurar comunicacions segures entre nodes, assegurar un consum baix a partir de tenir un volum de dades baix i maximitzar la vida útil de les seves bateries.

L'estàndard en què es basa el Zigbee, el IEEE 802.15.4 el qual defineix el nivell físic i el control d'accés al medi de les xarxes sense fils d'àrea personal, pretén assegurar un volum baix de dades, reduir el cost de la comunicació, adaptació per ser utilitzat en temps real mitjançant slots o paquets en un temps garantit, evasió de col·lisions per CSMA/CA i pretén oferir comunicacions segures entre nodes.

2.3.3 Sèries Xbee

El mòdul Xbee pot ser de 6 sèries possibles, el qual es mostren a continuació amb la seva característica més important:

1. XBee Series 1 (Xee 802.15.4) - Són fàcils de fer servir, ja que no necessiten ser configurats.
2. XBee Znet 2.5 (Reconegut com Series 2) - Retirats del mercat. Han de ser configurats abans de ser utilitzats.
3. ZB (actual mòdul Series2) - Es basen en el mòdul Znet 2.5, però amb un firmware millorat.
4. 2B(més actual que el mòdul Series2) - Presenten millores en el hardware respecte dels de sèrie 2 anteriors.
5. 900MHz - Tècnicament no són una sèrie el qual poden funcionar amb diferents firmware, només poden funcionar amb el firmware de Digi-Mesh i el firmware Point-to-Multipoint.
6. XSC - Bàsicament són mòduls de 900MHz, però redueixen la velocitat de dades a causa de la distància que són capaços d'operar.

La sèrie que s'utilitzarà, la sèrie 2, té les següents característiques:

1. 3.3V @ 40mA
2. 250kbps Max data rate
3. 2mW output (+3dBm)
4. 400ft (120m) range
5. Built-in antenna
6. Fully FCC certified
7. 6 10-bit ADC input pins
8. 8 digital IO pins
9. 128-bit encryption
10. Local or over-air configuration
11. AT or API command set

2.3.4 Modes de funcionament d'un Xbee

Mode de Rebre/Transmetre

Mode de Baix consum(sleep mode)

Mode de comanda

Mode transparent (AT)

Mode d'operació (API)

Mode IDLE

Mode de connexió transparent

Primerament mostrar el diagrama d'estats en què es pot trobar un xbee

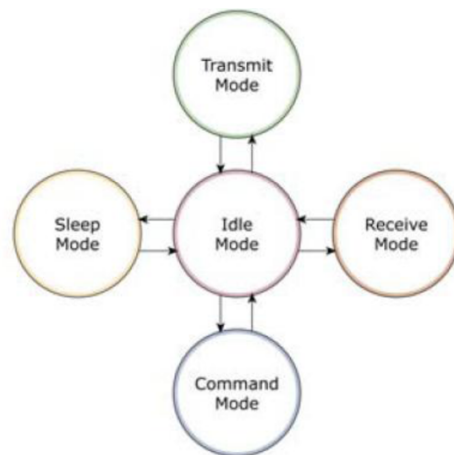


Figura 2.4: Estats

Mode de Rebre/Transmetre

Ens trobem en aquest mòdul quan en aquest li arriba algun paquet RF a través de l'antena (mode receive) o quan envia informació serial al buffer del pin 3 (UART Data in) que després serà transmesa (mode Transmit).

L'informació transmesa pot ser Directe o Indirecte. En el mode directe la informació s'envia immediatament a la direcció de destí. En el mode indirecte la informació és retinguda durant un període de temps i és enviada només quan la direcció de destí li demana.

A més, és possible enviar informació per Unicast o Broadcast

Mode de Baix consum

El mode de baix consum fa possible que el mòdul RF entri en un mode de baix consum d'energia quan no se'n fa ús.

Per a poder entrar en mode de repòs, s'ha de complir una de les següents condicions:

- Sleep-RQ (pin 9) està en nivell alt i el mòdul està en el pin sleep mode (SM=1,2 o 5).
- El mòdul està en repòs (no hi ha transmissió ni recepció de dades) per la quantitat de temps definit per ST (Time Before Sleep). [ST només està activat quan SM=4,5].

La configuració dels cicles de son es realitzen amb la comanda SM. Per defecte, els modes de son estan deshabilitats (SM=0) mantenint el mòdul en estat de repòs.

Dins d'aquest mode hi trobem modes secundaris com el mode de repòs per cicles

A continuació es presenta una taula-resum, considerant els modes d'operació, consums i voltatges i condicions per entrar en mode de baix consum.

Modo	Consumo alimentación			Modo Sleep	Modo Wake -up
	2.8 – 3 V	3.2 V	3.4V		
SM=1	<3 uA	32 uA	255 uA	Sleep RQ	Sleep RQ
SM=2	<35 uA	48 uA	170 uA	Sleep RQ	Sleep RQ
SM=3	(reservado)			(reservado)	(reservado)
SM=4	<34 uA	49 uA	240 uA	Comando ST	Comando SP
SM=5	<34 uA	49 uA	240 uA	Comando ST	Sleep RQ

Figura 2.5: Taula consums

Mode de comanda

En aquest mode es permet ingressar comandes AT (AT fa referència a comandes transparents, que no requereixen un format específic) al mòdul Xbee, per a configurar, ajustar o modificar paràmetres. Permet ajustar paràmetres com la direcció pròpia o la de destí. Així com el seu mode d'operació entre altres coses. Per a poder ingressar les comandes AT és necessari utilitzar el programa X-CTU.

Per ingressar a aquest mode s'han d'esperar un temps donat per la comanda GT que equival a 1000ms, tot seguit escriure +++ i esperar un altre cop el temps GT. Com a resposta el mòdul donarà un OK. El mòdul Xbee ve per defecte amb una velocitat de 9600bps. En cas de no poder ingressar al mode de comandes, és possible que sigui degut a la diferència de velocitats entre el mòdul i la interfície amb la qual es comunica via serial.

A continuació veiem l'exemple de la sintaxi d'una comanda AT.

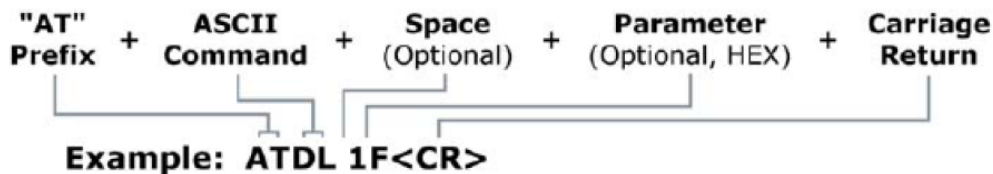


Figura 2.6: Comanda AT

Per a sortir d'aquest mode s'ha d'escriure ATCN, o el pròpi mòdul al cap d'un temps determinat per CT (Command Mode Timeout) en sortirà automàticament.

Mode transparent

En aquest mode tot el que arriba pel pin 3 (Data in) és guardat en el buffer d'entrada i després transmès i tot el que rep com a paquet RF, és guardat en el buffer de sortida i tot seguit enviat pel pin 2 (Data out). Aquest mode ve per defecte en els mòduls Xbee.

Aquest mode està destinat principalment a les comunicacions punt a punt, on no és necessari cap tipus de control. També s'utilitza per substituir alguna connexió serial per cable. En aquest mode la informació és rebuda pel pin 3 del mòdul Xbee, i és guardada en el buffer d'entrada. Depenent de com es configuri la comanda RO, es pot transmetre la informació tot just arribi un caracter (RO = 0), o després d'un temps donat sense rebre cap caràcter serial pel pin 3. En aquest moment, es pren el que tingui el buffer d'entrada, s'empaqueta, és a dir, s'integra a un paquet RF i es transmet.

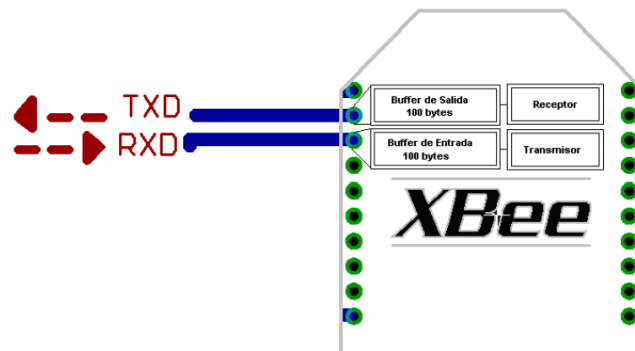


Figura 2.7: Transmissió - Recepció

Mode d'operació API

Aquest mode és més complex, però permet l'ús de frames amb capçaleres que assegurin l'entrega de dades, a l'estil TCP. Expandeix el nivell en el qual l'aplicació del client, pot interactuar amb les capacitats de xarxa del mòdul.

Quan el mòdul xbee és troba en aquest mode, tota la informació que entra i surt, és empaquetada en frames, que defineixen operacions i events dintre del mòdul.

Així doncs un frame de transmissió d'informació (informació rebuda pel pin 3 o DIN) inclou:

- Frame d'informació RF transmesa.
- Frame de comandes (equivalent a comandes AT).

Mentre que un frame de recepció d'informació inclou:

- Frame d'informació RF rebuda.
- Comanda de resposta
- Notificacions d'events com reset.

Aquest API, proveeix alternatives per a la configuració del mòdul i adreçament de l'informació en la capa d'aplicació del client. Un client pot enviar informació al mòdul Xbee. Aquestes dades seran contingudes en un frame el qual la seva capçalera tindrà informació útil referent al mòdul.

Aquesta informació a més es podrà configurar, això és, en lloc d'estar fent servir el mode de comandes per modificar les direccions, l'API ho realitza automàticament. El mòdul així enviarà paquets de dades continguts en frames a altres mòduls de destí, amb informació a les seves respectives aplicacions, contenint paquets d'estat, així com l'origen, RSSI (potència del senyal de recepció) i informació de la càrrega útil dels paquets rebuts.

Entre les opcions que permet la API, es tenen:

- Transmetre informació a múltiples destinataris, sense entrar al mode de comandes.
- Rebre estat d'èxit o fallida de cada paquet RF transmès.
- Identificar la direcció d'origen de cada paquet rebut.

Mode IDLE

Quan el mòdul no es troba en cap dels altres modes, es troba en aquest. És a dir, si no es troba ni transmetent ni rebent, ni estalviant energia ni en el mode de comandes, llavors es diu que es troba en un estat que es diu IDLE (ideal), el qual podem apreciar a la imatge 2.4, mostrada anteriorment .

Mode de connexió transparent

El mode transparent ve per defecte i és la més senzilla forma de configurar el mòdem. Bàsicament tot el que passa pel port UART (DIN, pin 3), és enviat al mòdul desitjat, i el que arriba al mòdul, és enviat de nou pel mateix port UART (DOUT, pin2). Existeixen bàsicament 4 tipus de connexió transparent. La diferència principal, es basa en el nombre de nodes o punts d'accés, i la forma en què aquest interactua'n entre si.

Podem trobar els següents tipus:

- Punt a punt
- Punt a Multipunt
- Broadcast
- Cable Virtual I/O

2.4 Bloc II. Hardware

2.4.1 Introducció

En aquest apartat es tractaran tots els dispositius físics utilitzats en el projecte, així com un breu resum de cada un, fen èmfasi en les seves característiques principals.

2.4.2 Arduino Uno R3

Aquest dispositiu triat, es basa en una placa amb un microcontrolador i un entorn de desenvolupament, dissenyada per a treballar en projectes multidisciplinaris. El hardware està basat en el chip ATmega328p. Té 14 entrades/sortides digitals, 6 entrades analògiques. L'Arduino permetrà processar les dades captades pels Xbee's i un cop processades es pendran unes accions depenen d'aquestes dades i tornaran a ser enviades al mateix xbee per què executi les noves ordres.

S'ha triat aquest dispositiu pel fet que al llarg del grau l'hem fet servir habitualment, i per tant, en tenim experiència en el seu ús, a més de ser un dispositiu compacte i capaç de comunicar-se amb els dispositius Xbee's mitjançant un Arduino Shield.

Summary

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz

Figura 2.8: Especificacions Arduino Uno R3

2.4.3 Arduino Shield

Aquest dispositiu permet a l'Arduino comunicar-se de forma sense fils utilitzant el protocol Zigbee.

Pot ser utilitzat com a substitut de sèrie/usb o es pot tenir en mode de comanda i configurar-lo per a una varietat d'emissions. Conté tots els pins pròpis de l'arduino però amb l'afegit que permet connectar-hi el xbee. A més a més, conté un 'jumper' que pot estar en dos estats , USB o Micro.

El Arduino Uno té una sola UART o USART hardware (port sèrie), que normalment s'utilitza per a la programació o per la comunicació amb el monitor sèrie. Aquest port sèrie només es pot utilitzar per comunicar-se amb un dispositiu a la vegada, per a què no tinguin problemes de contenció de bus o que en carregar el programa pugui ser enviat a qualsevol dispositiu connectat a la UART hardware del Arduino.

Així per evitar qualsevol problema que pugui sorgir de la comunicació de la Xbee a la UART hardware del Arduino, s'utilitza la biblioteca 'Software Serie', connectant els pins RX i TX del Xbee a un parell de pins digitals de l'arduino.

Amb el 'jumper' en la posició Xbee (És a dir, el pin en la posició Micro), el pin DOUT del mòdul Xbee està connectat al pin RX del microcontrolador, i DIN està connectat al pin TX. (DOUT fa referència a la sortida de dades i DIN a entrada de dades).

Cal tenir en compte, que en aquesta posició el microcontrolador només serà capaç de rebre dades procedents del Xbee. En la posició d'USB, no podrà rebre dades procedents del Xbee, només podrà comunicar-se amb el dispositiu connectat via USB.

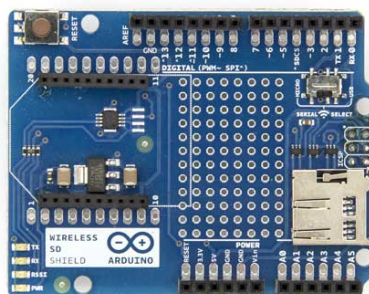


Figura 2.9: Shield Xbee-Arduino

2.4.4 Sensor PIR

Un sensor PIR (Passive Infra Red sensor) mesura la llum infraroja irradiada dels objectes que té al voltant. Cal considerar que tots els objectes amb una temperatura per sobre del zero absolut emeten calor en forma de radiació infraroja.

Aquests sensors s'utilitzen per a detectar el moviment de persones i/o animals. El sensor triat es capaç de detectar a una distància aproximadament d'uns 6 metres. A més, en aquest dispositiu se li pot ajustar el temps abans que es dispari la detecció (2-4 segons), es pot ajustar la seva sensibilitat de detecció. Treballa a un voltatge de 5V-16V, i ofereix un senyal de sortida de 3,3V. El rang de detecció arriba a 7 metres (En forma de con de 120 graus).

En la següent imatge podem veure que a la part inferior del sensor hi ha reguladors. Aquest permeten regular la sensibilitat i l'altre del temps que tardarà a disparar-se en detectar moviment.

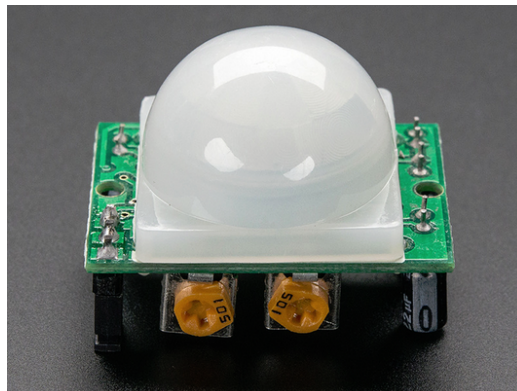


Figura 2.10: Sensor PIR

2.4.5 USB Explorer

Aquest dispositiu funciona per a totes les sèries de Xbee. Consisteix en una petita base adaptada per a la connexió del Xbee el qual inclou una connexió mini USB que permet connectar el Xbee amb qualsevol dispositiu amb entrada USB. En aquest projecte el dispositiu amb entrada USB serà un ordinador amb la funció de gestionar les dades rebudes de tots els nodes o punts finals. El Xbee connectat en aquest suport, anomenat USB Explorer, farà la funció de coordinador, ja que té la única funció de formar la xarxa i és el responsable d'establir el canal de comunicació entre nodes, i de la PAN ID, el qual identifica la xarxa. Un cop establerts aquest paràmetres el coordinador pot constituir una xarxa de comunicació amb totes els altres nodes que s'hi han afegit.

Un cop formada la xarxa el Coordinador fa les funcions de 'Router' i s'encarrega d'encaminar els paquets i ser destinatari o origen d'informació.

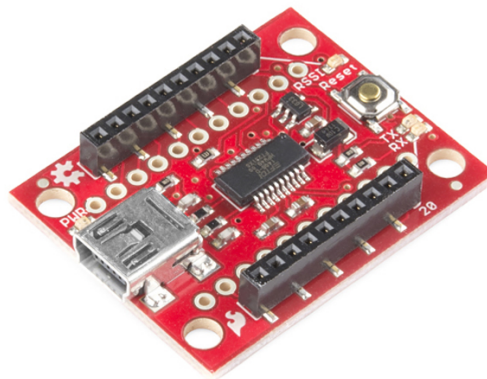


Figura 2.11: USB Explorer

2.4.6 LED's i resistències

Mitjançant leds bàsics i resistències de 270 ohms simularem un fanal. Cal tenir en compte que aquests components permeten la regulació de l'intensitat de llum emesa, permeten així a l'usuari apreciar quan un fanal funciona a ple rendiment o al 50 per cent de la seva capacitat. En la continuació d'aquest projecte es necessitarà uns altres components encarregats de la il·luminació, degut a que l'intensitat de llum esperada serà molt més alta.



Figura 2.12: Resistències de 270 Ohms

2.4.7 Muntatge del Hardware

A continuació es mostra una imatge referent a com seria el muntatge final fet en la part del hardware. Aquesta imatge s'ha fet a partir del programa Fritzing, el qual s'especialitza en la creació de PCB, plaques electròniques i muntatges de hardware [6]. Cal tenir en compte que tots els arduinos que es mostren estan connectats a l'ordinador per tal de ser alimentats. Els sensors negres que apareixen en l'imatge representen els sensors PIR.

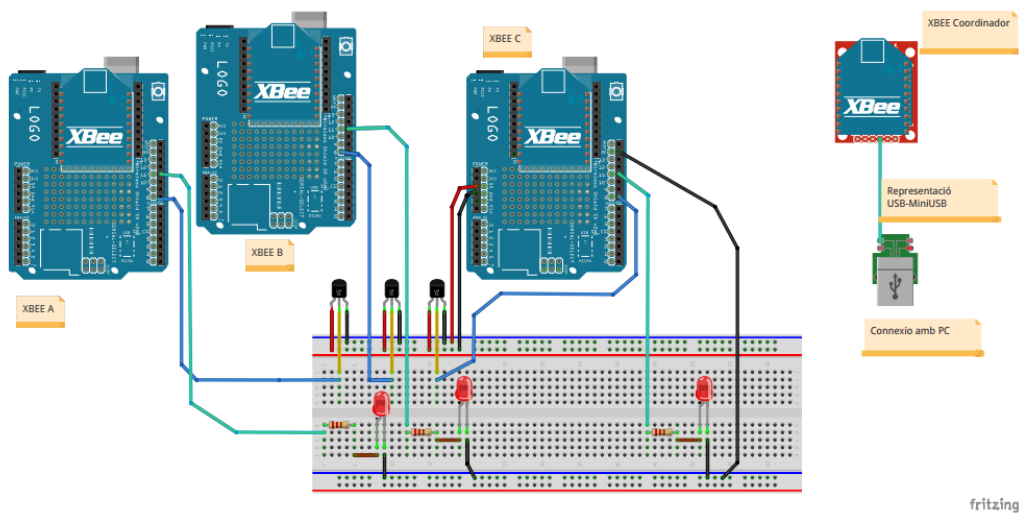


Figura 2.13: Muntatge Hardware

2.5 Bloc III. Sistema Web

2.5.1 Introducció

En aquest bloc es mostra les parts en què es divideix el sistema web, la base de dades i la interfície web, així com una breu explicació de cadascuna.

2.5.2 Base de dades

La base de dades s'encarrega de guardar totes les dades relacionades amb un tema, el qual poden ser posats en categories i ordenats com desitgem. Entre els elements que formen la base de dades, aquests poden compartir relacions entre ells. Per a gestionar la base de dades s'ha fet servir el gestor de base de dades anomenat MySQL, que ens permet programar amb el llenguatge de base de dades SQL de manera còmoda i intuïtiva.

La base de dades està formada per taules, el qual una taula, vindria a ser una col·lecció de dades organitzades en files i columnes. Gràficament, s'assemblen molt a fulls de càlcul de programes de l'estil de l'Excel. Les dades s'organitzen en columnes, el qual també s'anomenen camps, que contenen un tipus d'informació en concret.

Quan creem una base de dades hem de tenir en compte el tipus de dades que hi volem posar, és a dir, si un camp fa referència a un tipus de valor numèric o de text per exemple.

Podem trobar tots aquests tipus de dades en el gestor esmentat:

1. Numèric
2. Data i temps
3. Cadena de caràcters
4. Espacials

Numèric

MySQL es capaç de suportar tots els tipus de valors numèrics estàndards de SQL. Aquests tipus inclouen les dades numèriques exactes com per exemple INTEGER, SMALLINT, DECIMAL, i NUMERIC, tals com els tipus de dades numèriques referents a les aproximacions FLOAT, REAL, I DOUBLE PRECISION. Cal considerar que MySql utilitza diverses abreviatures referents als tipus de dades en general com per exemple el tipus INT el qual es representa com INTEGER, o DEC i FIXED el qual són sinònims de DECIMAL. Podem trobar més informació d'abreviacions a [9].

Data i temps

Els tipus que representen les dates i el temps són DATE, TIME, DATETIME, TIMESTAMP, i YEAR. Cada tipus temporal esmentat té un rang de valors vàlids com per exemple el zero.

Cadena de caràcters

Els tipus possibles per aquests són CHAR, VARCHAR, BINARY, VARBINARY, BLOB, TEXT, ENUM, i SET.

Espacials

Els tipus que corresponen a aquest apartat son GEOMETRY, POINT, LINESTRING, POLYGON, MULTIPOINT, MULTILINESTRING, MULTIPOLYGON, GEOMETRYCOLLECTION. Cal tenir en compte que tots aquests tipus corresponen a la classe OpenGIS.

2.5.3 Interfície web

La interfície web del projecte la formen tots els elements de la pàgina el qual l'usuari hi pot interactuar, realitzant accions sobre la web que està visitant. Totes aquestes accions sobre la web han de ser capaces d'oferir els serveis que l'usuari espera trobar.

Aquest entorn web haurà de poder interactuar amb la base de dades, ja que, el que es pretén mostrar principalment és l'estat dels fanals, així com poder actuar sobre aquestes també. Per tant, la interfície web haurà de llegir i escriure a la base de dades segons les accions de l'usuari.

No es pretén carregar la pàgina web estèticament, pel fet que l'objectiu no és cridar l'atenció de cap sector, sinó fer un disseny compacte, ordenat, i poc carregat, a més, que sobrecarregar la web no és l'objectiu d'aquest projecte.

La interfície no serà d'àmbit públic, ja que, es podrà interactuar amb els fanals i qualsevol sense permís en podria fer un mal ús. Per això la pàgina web serà protegida mitjançant contrasenya que només es facilitarà als administradors de la web que s'encarregaran de controlar i monitorar el sistema. Aquest sistema de seguretat s'ha basat en l'ús d'un plugin pròpi de Wordpress, el qual no es basa en cap tipus de protocol de seguretat. En l'apartat de configuració de l'interfície web es s'explicarà la seguretat establerta per aquesta interfície.

Capítol 3

Configuració i Programació del sistema d'il·luminació

3.1 Introducció

En aquest apartat s'explicarà la configuració i programació feta dels tres blocs esmentats anteriorment, amb l'objectiu de generar un sistema que pugui satisfer totes les necessitats plantejades al llarg del projecte. També es mostrarà el codi generat, juntament amb la seva explicació.

3.2 Configuració del Sistema de comunicació utilitzant el programa X-CTU

L'aplicació que s'ha escollit per a la configuració dels Xbee's ha estat la plataforma X-CTU de la companyia Digi (<http://www.digi.com/>). Existeixen altres plataformes que permeten la configuració d'aquests dispositius, com per exemple, Multisenso.

Els motius per als quals s'ha triat X-CTU ha estat que és una plataforma gratuïta, capaç de funcionar tant a sistemes operatius com Windows, MacOS, inclús a Linux si es disposa de l'aplicació Wine per executar-la.

Un altre motiu és que permet una visió molt gràfica per a la configuració de xarxes i arquitectures, que de cara a l'usuari resulta ser més senzill i entenedor. A més aquesta plataforma permet construir trames en format API de manera molt senzilla gràcies a les seves eines que en permeten la seva generació.

Per últim motiu, té integrat l'opció de fer servir l'aplicació del 'núvol', és a dir, ens permet treballar amb un dispositiu allà on siguem, encara que estiguem a una distància molt allunyada.

Primerament hem de connectar els Xbee's a l'ordinador mitjançant una 'Shield' per tal de tenir establerta la connexió Xbee-pc. Un cop connectat el Xbee, podem obrir el X-ctu, cal tenir en compte que tota la configuració referent a x-ctu fa referència a la seva última versió presentada a www.digi.com a versió 6.1.2.

Dins del programa x-ctu i ja amb els Xbee connectats haurem de polsar al símbol de l'esquerre a dalt de tot del programa on apareix com un Xbee amb un símbol '+', el qual s'encarrega de trobar els Xbee connectats.

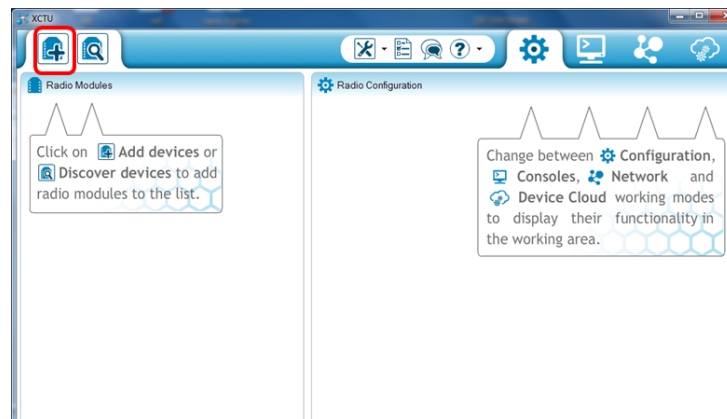


Figura 3.1: Detecció del Xbee - X-ctu

Podrem veure que el programa ha detectat el Xbee en aparèixer la següent pantalla.

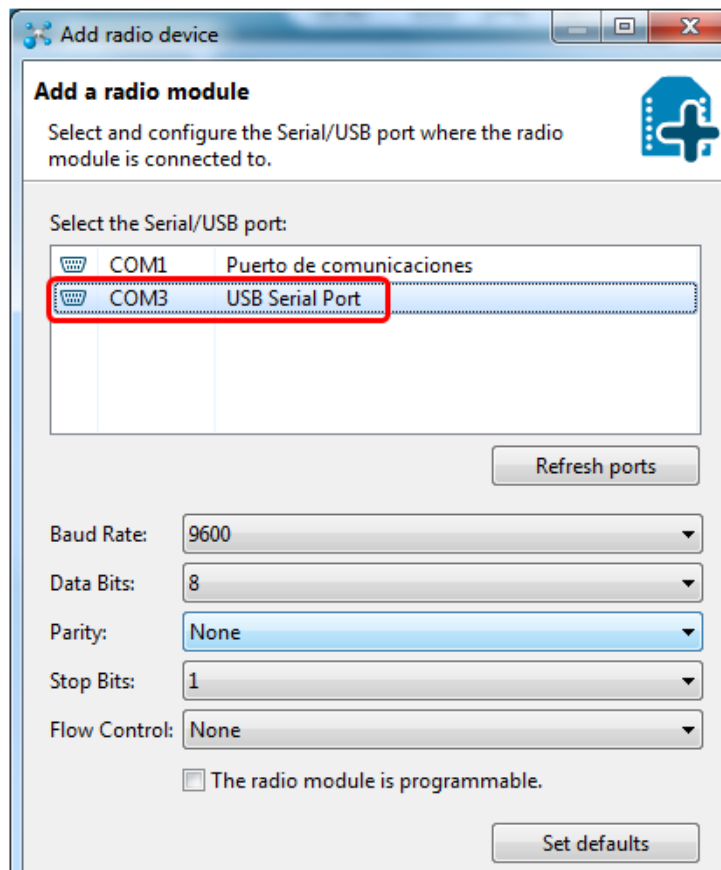


Figura 3.2: Comprovació que s'ha trobat el xbee - X-ctu

Tot seguit en l'apartat Radio Modules ens apareixeran els Xbee's connectats, el qual al clicar-hi podem veure'n la configuració que ja porten i així també els podrem configurar.



Figura 3.3: Radio Modules - X-ctu

Abans de configurar els xbee's es recomanable restaurar a als valors de fàbrica els Xbee's, ja que d'aquesta manera ens assegurem que cap paràmetre d'una configuració anterior ens afecti.

Per a fer un reset el Xbee de fàbrica haurem de polsar dins de 'Radio Configuration' el símbol d'una fàbrica. Tot seguit hem d'establir quin tipus de comunicació hi haurà entre dispositius. Haurem d'anar al símbol en forma de xip amb una fletxa sobre aquest (Update firmware), on definirem els tipus de Xbee que tenim i la funció que volem que faci, com End Device en mode AT, o Coordinador o bé Router, seguidament de la versió de firmware.

Un cop carregada ja la versió de firmware que hem triat, podem veure que la configuració bàsica dels dispositius ha canviat. A partir d'aquí podem procedir a configurar els paràmetres del firmware.

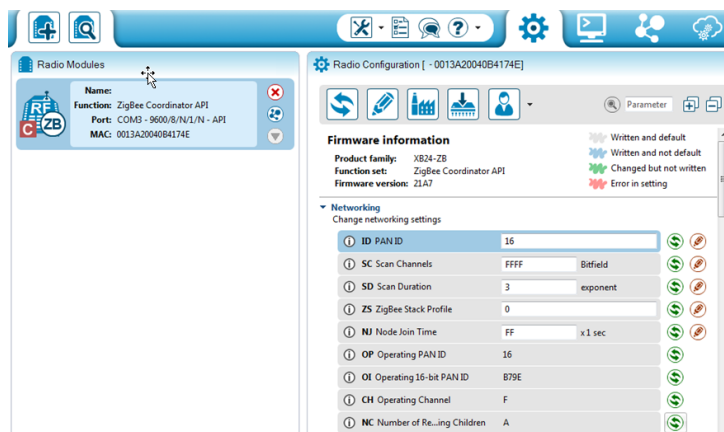


Figura 3.4: Pantalla configuració xbee - X-ctu

En la imatge presentada podem apreciar la icona d'una fàbrica, el qual ens permet resetejar a valors de fàbrica el Xbee. També podem veure que podem modificar els seus paràmetres com la 'PAN ID'.

Abans de configurar res hem de tenir en compte certs aspectes que esmentarem.

S'ha pensat que explicar part de la configuració dels xbee mitjançant imatges faria menys pesat d'entendre'n el seu funcionament, ja que, el funcionament d'aquests dispositius s'ha considerat important degut a que s'encarregen de la comunicació de tot el sistema generat.

Els tres principals opcions que modificarem seran la 'PAN ID', 'MY address' i la 'destination address'.

Hi ha diversos nivells de xarxes Xbee. Primer hi ha el canal o channel. Aquest controla l'ample de freqüència amb el que el Xbee es comunica. La majoria de Xbee's treballa a 2.4GHz 802.15.4 d'ample, i el canal calibra la freqüència d'operació dintre aquest ample. Generalment es pot deixar el paràmetre de canal sol, o si més no, assegurar-se de què cada Xbee que s'espera tenir en la mateixa xarxa treballi en el mateix canal.

El següent nivell de la xarxa Xbee és la 'Personal Area Network ID' (PAN ID). La xarxa ID està formada per valors hexadecimals entre 0 i 0xFFFF. El Xbee només pot comunicar-se amb un altre si aquest té la mateixa PAN ID. Hi ha 65536 possibles ID's.

Finalment la MY i la 'destination address'. Cada Xbee en una xarxa hauria de ser assignat a una adreça de 16 bits (des de 0 a 0xFFFF), el qual fa referència a la 'MY address' o adreça font. L'altre paràmetre, la adreça destí, determina a quina adreça, un Xbee pot enviar-li dades. Per exemple, si el Xbee 1 té l'adreça 0x1234, i un altre Xbee 2 té la mateixa adreça, llavors el Xbee 2 pot enviar dades al Xbee 1. En cas de tenir diferents adreces ja no podrien enviar-se dades.

Per tant, dins del x-ctu haurem d'anar a 'Configuration tab' que té la icona d'un engranatge, omplir els paràmetres explicats anteriorment, el CH (channel), l'ID (PAN ID), DH (Part alta adreça destí), DL (Part baixa adreça destí) i la MY (Adreça font).

Cal tenir en compte que dos xbee's poden tenir la mateixa MY, pero llavors els dos rebran les mateixes dades, vindria a ser com un broadcasted.

En el cas dels End-Device si volem configurar-los, necessitem establir un temps de sleep elevat. Per tal d'introduir les comandes correctament, cada cop que volem modificar els seus paràmetres és necessari fer un reset al mòdem (connectant el pin de Reset 5 a GND, que és el 10 o el GND del Arduino).

Cal tenir en compte que la comunicació des de el coordinador als nodes, ha de seguir una estructura en concret, el qual es mostrarà en l'apartat de configuració del sistema. Aquest comunicació estarà en format hexadecimal, on el coordinador dins del missatge o paquet que enviï haurà d'introduir l'a-

dreça del destinatari, així com el pròpi missatge, entre d'altres paràmetres que es mostraran més endavant. Podrem apreciar en l'imatge 3.5 i 3.6 l'estructura que ha de tenir aquests paquets o trames per part del coordinador.

Si volem provar la comunicació entre dos end-device no ens trobarem amb cap dificultat, ja que un cop configurats els dos Xbee en mode End-device AT podran comunicar-se directament entre ells, sense formatar el missatge.

En el cas de voler enviar un paquet en mode API, ja hem de seguir unes certes pautes a seguir per a comunicar-nos amb un altre Xbee.

La següent imatge ens ofereix una idea general de l'estructura que hauria de tenir.

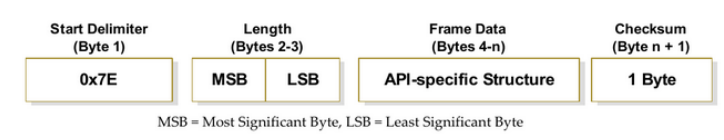


Figura 3.5: Estructura trama

Paquet en mode API:

1. **El Byte 1** - Sempre 0x7E
2. **Bytes 2-3** - Indica la llargada del paquet, calculat a partir dels bits entre el byte menys significatiu del tamany i el byte del checksum.
3. **Bytes 4-n** - Aquí és on pot variar la estructura del paquet, depenent de la funció API que es vulgui cridar s'utilitzara una o altre estructura.
4. **Byte n+1** - La suma de la verificació de tots els bytes continguts en el paquet. Es fa servir per a verificar la integritat dels missatges.

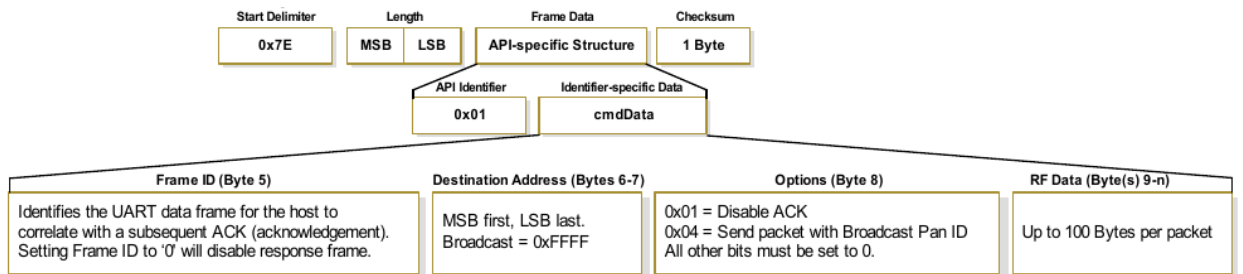


Figura 3.6: Estructura trama

A continuació es mostraran els principals paràmetres que hauriem de modificar del xbee per a poder establir una correcta comunicació entre ells, tenint en compte que hem posat tots els Xbee's amb la mateixa 'PAN ID'.

En la següent imatge podem veure que li fixem una adreça destí fixa el qual ell dirigirà tots els paquets generats.

▼ Addressing
Change addressing settings

SH Serial Number High	13A200
SL Serial Number Low	40B4174E
MY 16-bit Network Address	0
DH Destination Address High	13A200
DL Destination Address Low	408B6061
NI Node Identifier	
NH Maximum Hops	30
BH Broadcast Radius	0
AR Many-to-One ...oadcast Time	FF x10 sec
DD Device Type Identifier	30000
NT Node Discovery Backoff	3C x100 ms
NO Node Discovery Options	3

Figura 3.7: Paràmetre destinació xbee - X-ctu

Si volem transmetre un paquet a un altre Xbee utilitzant l'adreça de 16 bits:

1. **Byte 4** - Identificador de l'API, aquest codi identifica el tipus de comanda que volem executar, per aquest cas el codi 0x01 correspon a 'Transmit Request: 16-Bit Address' o sol·licitud de transmissió a una adreça de 16 bits.
2. **Byte 5** - Correspon a un nombre seqüencial del paquet. Per a cada enviament es rep el ACK, el qual ens servirà més endavant per verificar que un paquet que hem enviat l'ha rebut el destinatari.
3. **Byte 6,7** - Corresponen a la direcció del Xbee destinatari, es col·loca primer el byte més significatiu i després el menys significatiu.
4. **Byte 8** - Correspon les opcions alhora d'enviar, si establim el bit 0x01 en 1 no s'esperarà un ACK, mentre que si habilitem el bit 0x04 s'enviarà el missatge amb broadcast per totes les xarxes d'àrea personal (PAN ID).
5. **Byte 9** - Dades a enviar (Màxim 100 bytes). El últim byte, la suma de verificació, s'utilitza per garantir que el paquet s'ha transmès de manera íntegra, el qual el calculem restant a 0xFF la suma truncada a 8 bits de tots els bytes entre la mida del paquet i el checksum, per a l'exemple anterior el checksum seria: 0xC3

3.2.1 Configuració feta per a End Device AT, Coordinador API

En el projecte es disposa de quatre dispositius principals, un Xbee que s'encarrega d'administrar les dades que li arriben del Arduino i tot seguit actualitza aquestes dades a la web. Aquest dispositiu s'ha configurat com a coordinador en mode API. Dit dispositiu és coordinador, ja que és necessari per a formar la xarxa de comunicació. S'encarrega d'establir el canal de comunicacions i de la PAN ID per tota la xarxa. Un cop fet això, es poden unir a ell dispositius 'Routers' i 'end devices'.

Un cop formada la xarxa el coordinador fa funcions molt similars al dispositiu Router, és a dir, participar en l'encaminament de paquets i ser l'origen i/o destí d'informació. Els motius pel qual s'ha fet en mode API i no en mode AT és a causa del fet que es necessita transmetre l'informació rebuda a un destinatari sense haver d'entrar al mode de comandes, necessitem saber si cada paquet RF transmès s'ha enviat amb èxit o no. A més a més, necessitem poder identificar la direcció d'origen de cada paquet rebut per tal de poder-ho gestionar correctament.

Els altres tres dispositius s'ha decidit que fossin end device, ja que no fa falta que s'encarreguin de encaminar paquets. Aquests nodes seran els encarregats d'enviar les dades de l'estat de les fanals cada vegada que es produeixi un esdeveniment. El motiu pel qual s'ha decidit que es configurassin en mode AT ha estat que aquest mode és ideal per a comunicacions punt a punt (end device – coordinador), ja que no es requereix cap tipus de control, a més de tenir un consum relativament baix comparat amb els altres modes de funcionament.

Els paràmetres principals modificats han estat:

Xbee A:

Adreça: 0013A200 408B66E1

PANID: 16

Mode: Transparent (AT) End-device

Destination address: 0013A200 40B4174E

Xbee B:

Adreça: 0013A200 408B6061

PANID: 16

Mode: Transparent (AT) End-device

Destination address: 0013A200 40B4174E

Xbee C:

Adreça: 0013A200 408B6088

PANID: 16

Mode: Transparent (AT) End-device

Destination address: 0013A200 40B4174E

Xbee Coordinador:

Adreça: 0013A200 40B4174E

PANID: 16

Mode: Coordinador (API)

Cal tenir en compte que en tots els dispositius el paràmetre dins de l'apartat de configuració, Sleep Modes del X-ctu, anomenat SM (Sleep Mode) ha estat desactivat, ja que no volem que els nostres dispositius entrin en un estat de repòs o que s'adormin, es vol que estiguin operatius sempre, ja que pot rebre un paquet provinent del Arduino en qualsevol moment, a més que es vol que la comunicació sigui el més ràpid possible.

Pel que fa a la part de Hardware, hem de tenir en compte que hem hagut d'utilitzar certs pins dels xbee. Partint d'aquesta llista de pins:

Table 1-02. Pin Assignments for the XBee and XBee-PRO Modules
(Low-asserted signals are distinguished with a horizontal line above signal name.)

Pin #	Name	Direction	Description
1	VCC	-	Power supply
2	DOUT	Output	UART Data Out
3	DIN / CONFIG	Input	UART Data In
4	DO8*	Output	Digital Output 8
5	<u>RESET</u>	Input	Module Reset (reset pulse must be at least 200 ns)
6	PWM0 / RSSI	Output	PWM Output 0 / RX Signal Strength Indicator
7	PWM1	Output	PWM Output 1
8	[reserved]	-	Do not connect
9	<u>DTR</u> / SLEEP_RQ / DI8	Input	Pin Sleep Control Line or Digital Input 8
10	GND	-	Ground
11	AD4 / DIO4	Either	Analog Input 4 or Digital I/O 4
12	<u>CTS</u> / DIO7	Either	Clear-to-Send Flow Control or Digital I/O 7
13	ON / SLEEP	Output	Module Status Indicator
14	VREF	Input	Voltage Reference for A/D Inputs
15	<u>Associate</u> / AD5 / DIO5	Either	Associated Indicator, Analog Input 5 or Digital I/O 5
16	<u>RTS</u> / AD6 / DIO6	Either	Request-to-Send Flow Control, Analog Input 6 or Digital I/O 6
17	AD3 / DIO3	Either	Analog Input 3 or Digital I/O 3
18	AD2 / DIO2	Either	Analog Input 2 or Digital I/O 2
19	AD1 / DIO1	Either	Analog Input 1 or Digital I/O 1
20	AD0 / DIO0	Either	Analog Input 0 or Digital I/O 0

Figura 3.8: Pins

Normalment, en intentar carregar la configuració al Xbee, es demana fer un reset al dispositiu físicament, per a fer-ho hem de posar a 0V el pin 5 del Xbee.

3.3 Configuració hardware

Primerament mostrar una imatge referent a la comunicació entre nodes. On es pot apreciar com s'ha representat el sistema. El node A i C representen nodes finals o inicials, depenent d'on decidim començar. El sistema presentat fa referència a un sistema no-tancat, on el node inicial i el final estan molt separats entre ells, sense tenir cap comunicació entre ells. En cas de fer una maqueta més gran o implementar el sistema a escala real, es podria fer un sistema de detecció tancat, mitjançant sensors PIR, ja que així aquests sensors podrien ocupar una zona sencera, on podrien detectar.

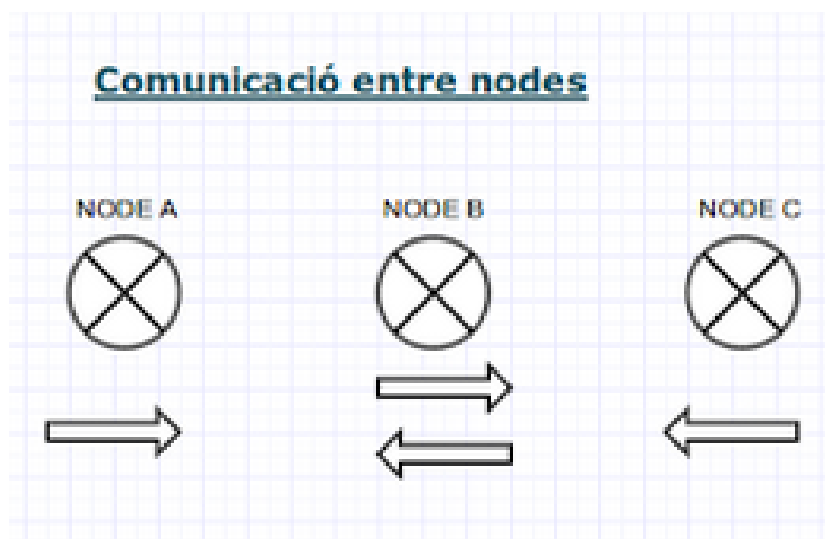


Figura 3.9: Comunicació entre nodes

Els components hardware ha configurar són l'arduino i un ordinador. Els Arduinos s'encarregaran de processar totes les dades provinents dels sensors així com gestionar-les. En l'ordinador haurà un programa encarregat de gestionar les dades procedents dels Arduinos, el qual rebrà gràcies a la connexió d'un Xbee coordinador.

La connexió Xbee coordinador amb l'ordinador es farà a partir d'un cable USB-MiniUSB. En l'ordinador haurà un programa creat amb Python que s'encarregarà de gestionar les dades provinents dels nodes o punts finals i els gestionarà.

El programa generat en Python gestionarà les dades a partir d'un seguit de màquines d'estat. Cada node o punt final es basa en una màquina d'estats per tal de controlar totalment els estats del dispositiu. Finalment hi haurà una màquina d'estats en el coordinador el qual s'encarregarà de gestionar les dades dels nodes.

Tot seguit es mostrarà una descripció dels estats en què un node pot estar i quines accions s'hauran de dur a terme. En aquesta descripció s'esmenten les accions que el sistema duu a terme, referents a la imatge de la màquina d'estats que es presenta seguidament. Cal comentar que la funció del timer que s'ha programat fa la funció com de temporitzador que es dispara al arribar a un cert valor del comptador que es va incrementant al llarg del temps.

1. Inicialment el node es troba en repòs, en què el Timer (TMR_n) està apagat així com el led de el dit node (LED_n). Cal tenir en compte que 'n' representa el número de node.
2. Passat l'estat de repòs s'entra a l'Estat '0' o estat principal. Només avança ara a l'Estat 1 en cas de detectar moviment (M_n) o de rebre una ordre provinent del coordinador (Ordre ON n).
3. Si es detecta moviment (M_n), s'activa el timer (TMR_n ON), el comptador del Timer (T_n) es reinicia posant-lo a 0, encén el Led del node en què ens trobem (ON LED_n) i envia una notificació al coordinador per informar-lo que ha encès (TX ON LED_n).
4. Si es rep una Ordre del coordinador (Ordre ON n), s'activa el timer (TMR_n ON), el comptador del Timer (T_n) es reinicia posant-lo a 0, i finalment encén el led del node (ON LED n)
5. Si s'han finalitzat un d'aquests dos esdeveniments, passarem a l'estat 1.
6. En l'estat 1 hi trobem els mateixos esdeveniments anteriors, però amb diferents accions. En cas de detectar moviment reiniciem el comptador del timer i notifiquem al coordinador que ha d'encendre el següent node respecte del qual ens trobem (TX ON LED_{n+1}) Si rebem una ordre del coordinador, en aquest cas simplement reiniciarem el comptador del timer.

7. En cas que no s'hagin produït els esdeveniments esmentats anteriorment si el comptador del timer ha arribat al seu màxim, es retorna a l'estat inicial o estat 0, en què s'apaga el timer (OFF TMR_n), s'apaga el corresponent led (LED_n) i s'envia una notificació al coordinador tal com el led s'ha apagat (TX info LED n OFF)

Per a donar èmfasi en aquesta descripció es mostraran les màquines d'estat dels nodes, per a tindre una visió més gràfica del seu funcionament.

Màquina d'estats del Node A:

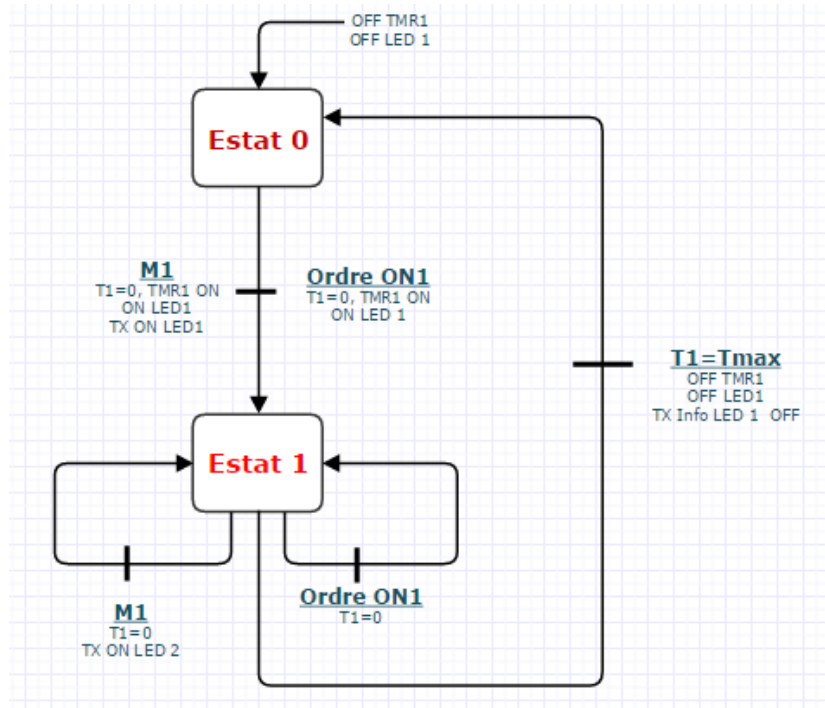


Figura 3.10: Màquina d'estats node A

Màquina d'estats del Node B:

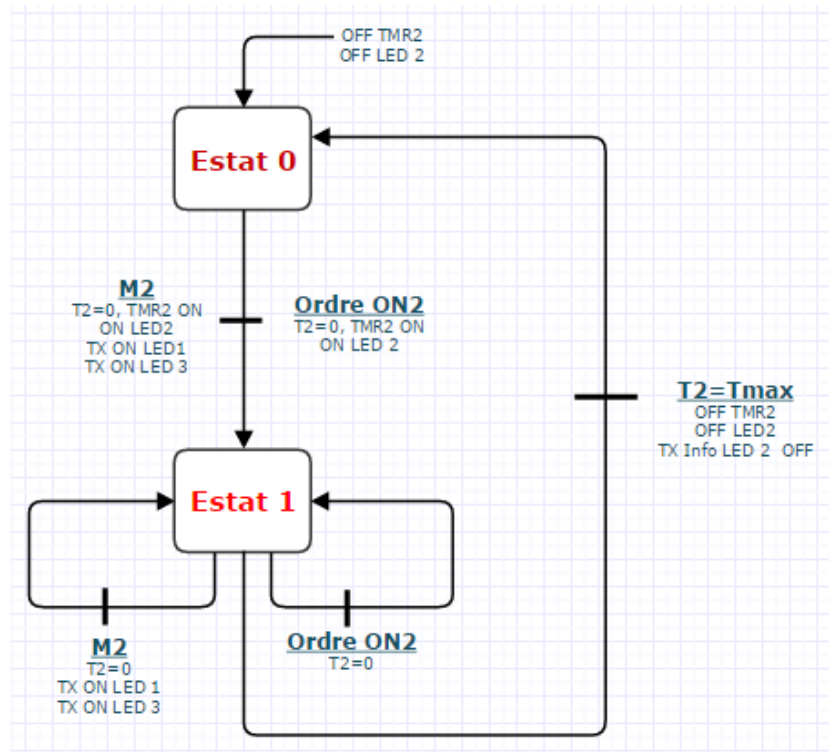


Figura 3.11: Màquina d'estats node B

En aquesta màquina d'estats es pot apreciar que varien les accions a dur a terme pel node. Ja que el Node B ens trobem que hi ha diferència respecte del Node A o C, pot rebre esdeveniments provinents de qualsevol banda, tant del node A com el node C. Per tant, si aquest node detecta moviment haurà de notificar-ne al node A i C que s'encenguin, a diferència dels nodes A i C que només notificaven en una direcció.

Màquina d'estats del Node C:

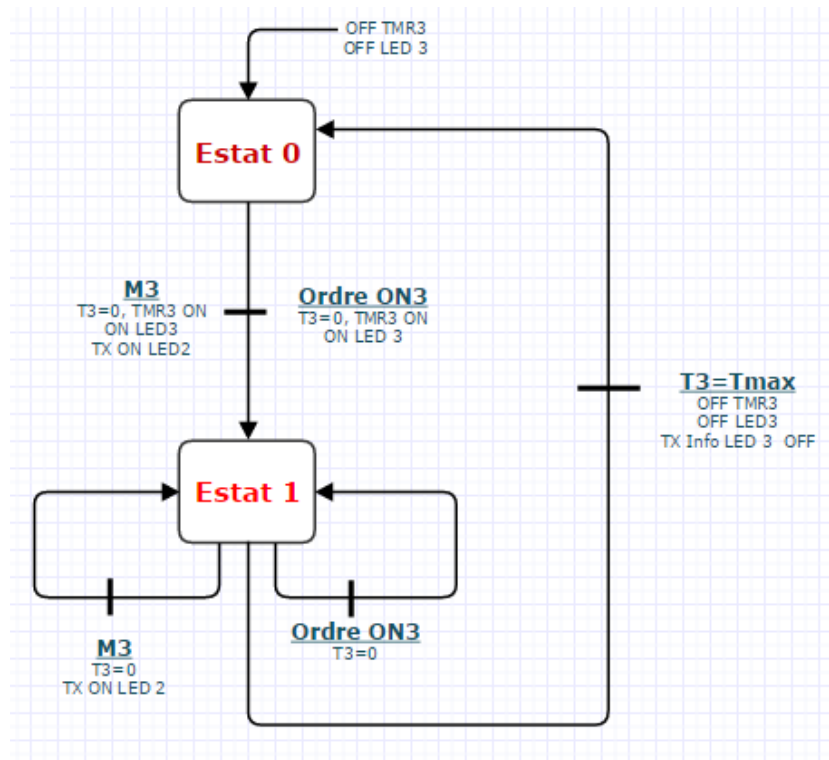


Figura 3.12: Màquina d'estats node C

Finalment presentar la màquina d'estats del coordinador, que s'encarregarà de gestionar les dades provinents dels punts finals.

A continuació s'expliquen els estats en què el coordinador pot estar. Cal

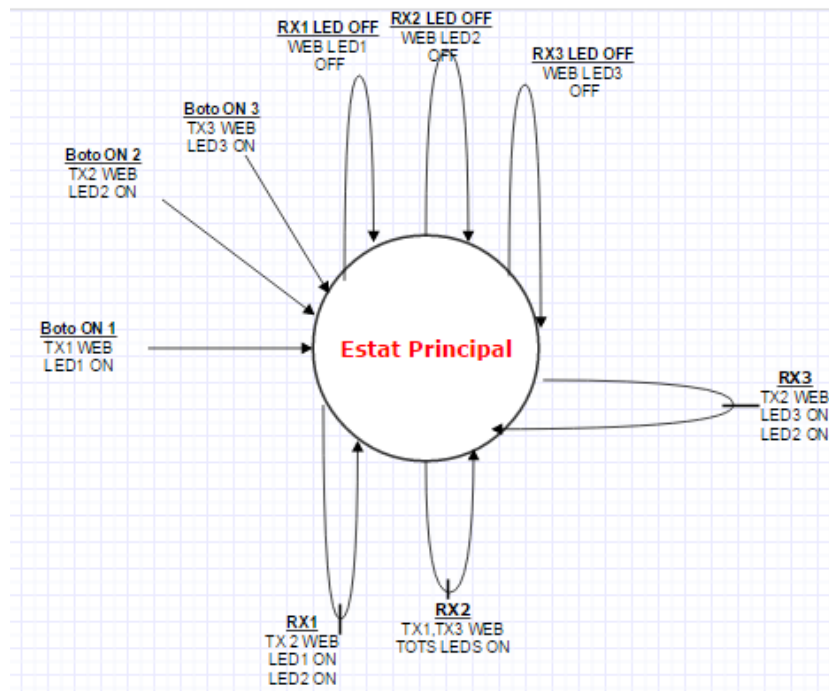


Figura 3.13: Màquina d'estats Coordinador

tenir en compte que 'n' fa referència al número de node.

1. Event Boto ON n. Representa els botons de les llums situats en la web, en què en ser pulsats encenen el llum de la web i el led del sistema hardware.
2. Event Rx(n) LED OFF. S'apaga el led n situat en la web.
3. Event RX(n). Fa referència als missatges que rep el coordinador per part dels nodes finals. Per exemple si rep la notificació provinent del node A del sistema hardware que està encès, el coordinador comunica al següent node que s'ha d'encendre, en aquest cas el node B, així doncs s'encendran el led del Node A i el del B. En el cas de rebre del Node B, el qual està al mig del sistema d'il·luminació, aquest notificarà al coordinador que s'han d'encendre els leds del Node A i C, per tant com a resultat tots els nodes tindran el led encès.

3.3.1 Programació programa principal carregat al ordinador.

Tot el codi presentat a continuació ha estat comentat per a explicar-ne el seu funcionament. Per arrancar el programa hem d'escriure en un terminal obert des de la carpeta on tenim tot el codi:

```
$ python main.py/dev/ttyUSB0
```

On main.py es el nom del programa carregat al ordinador, el qual a continuació s'explicarà el seu funcionament.

Importació de totes les llibreries necessàries, així com els altres fitxers creats que s'en faran ús en aquest programa.

```
#Llibreries creades: xbee, trama  
#Llibreries ja creades: serial, sys, binascii, time,  
MySQLdb, datetime  
  
import xbee  
import trama  
import serial, sys  
import binascii  
from time import sleep  
  
import MySQLdb  
from datetime import date, datetime, timedelta
```

Inicialització del sistema, es creen les variables ha utilitzar amb el contingut necessàri.

Primerament s'envia el missatge 'Running' per a notificar-nos que el programa ha començat. Seguidament s'en extreu de la comanda esmentada anteriorment (`python main.py/dev/ttyUSB0`) la ubicació del port sèrie, que en aquest cas es `/dev/ttyUSB0`.

Aquesta ubicació fa referència a on hem connectat al coordinador que és un xbee, el qual aquest està connectat com un USB gràcies al USB Explorer que ens connecta el Xbee amb el ordinador.

Seguidament s'en guarda aquesta direcció per quan ens haguem de comunicar, ja que necessitem saber per on s'enviaran les dades cap a altres nodes.

Guardem les adreces dels xbee's en variables per a saber exactament a quin Xbee ens hem de comunicar. Seguidament inicialitzem els valors de la pàgina web, el qual aquesta revisa els valors en que ha d'estar en la Base de Dades.

En la Base de dades definim el estat de cada node, així com el valor del fanal. Hem de tenir en compte que aquest valors no s'afegeixen d'immediat a la base de dades, i per tant hem d'afegir un petit temps entre comanda a la base de dades així pot introduir els valors sense problemes.

```
#Principi programa principal anomenat __main__

if __name__ == '__main__':
    try:
        print "Running"
        ser=sys.argv[-1]
        xbee = xbee.XBee(ser)
        trama =trama.Trama()
        addr1="13a200408b66e1" #Node B
        addr2="13a200408b6061" #NODE A
        addr3="13a200408b6088" #Node C
        prova=trama.Basedades(1,1111,2222,"OFF","OFF",0)
        sleep(0.5)
        prova=trama.Basedades(2,3333,4444,"OFF","OFF",0)
        sleep(0.5)
        prova=trama.Basedades(3,5555,6666,"OFF","OFF",0)
        sleep(0.5)
        a=0
        b=0
        c=0
```


Dins del 'while' de continuació trobaríem el nucli del nostre programa el qual aquest 'while(True)' representa un bucle infinit que és produirà sempre. Dins d'aquest bucle guardem el missatge rebut pel Xbee, només en cas d'haver rebut.

Seguidament guardem en una variable anomenada 'missi' la lectura feta de la base de dades, en la que mirem si una variable de la base de dades ens indica que hem de llegir, ja que s'ha polsat un fanal del sistema web per encendre-la en el sistema.

En cas que hàgim de llegir, mirem una segona variable de la base de dades que ens informa de quin node es tracta. Un cop sabem què hem de llegir, el node que s'ha de produir l'acció, ens comuniquem amb el Xbee adequat per encendre el fanal físic, el qual ha estat polsat en la web.

Cal tenir en compte que en polsar per exemple el fanal A en el sistema web, s'encén el fanal A de la web i el fanal físic A. Finalment guardem a la base de dades el valor que ens informava si hem de tornar a llegir, però amb un valor que indiqui que no hem de tornar a llegir.

```
while (True) :  
  
    missatge = xbee.Receive()  
    missi=trama.Basedades2()  
  
    if missi[0]==1:  
        if missi[1]==1:  
            xbee.SendStr('1', int(addr2,16))  
            provaZ=trama.Basedades3(0)  
        elif missi[1]==2:  
            xbee.SendStr('2', int(addr1,16))  
            provaZ=trama.Basedades3(0)  
        elif missi[1]==3:  
            xbee.SendStr('3', int(addr3,16))  
            provaZ=trama.Basedades3(0)
```

En cas que la variable 'missi' ens informi que no hem de llegir, comprovem si hem rebut algun missatge dels punts finals o nodes.

Mirem si el missatge rebut és provinent del node A, B o C. Per a qualsevol cas el procediment és el mateix, hem de notificar els nodes corresponents el qual en la màquina d'estats del coordinador es dona una idea gràfica. A més, s'actualitza la base de dades amb el nou estat dels nodes afectats.

```
elif missatge:
    # Es fa un print del missatge en format
    # llegible
    content = missatge[14:-1].decode('utf-8', 'ignore')
    print("Missatge_RF_rebut:_" + content)

if content=='A':

    a=a+1
    xbee.SendStr('2', int(addr1,16))#
        adreça del xbee 1 - node B
    provaD=trama.Basedades(2,1111,2222,
        "ON", "ON", a)

elif content=='B':
    b=b+1
    c=c+1
    xbee.SendStr('1', int(addr2,16))
    xbee.SendStr('3', int(addr3,16))
    provaD=trama.Basedades(1,1111,2222,
        "ON", "ON", b)
    sleep(0.5)
    provaC=trama.Basedades(3,5555,6666,
        "ON", "ON", c)
elif content=='C':
    c=c+1
    xbee.SendStr('2', int(addr1,16))#
        adreça del xbee 1 - node B
    provaC=trama.Basedades(3,5555,6666,
        "ON", "ON", c)
```

En el cas que el missatge rebut provinent dels punts finals notifiqui que el node s'ha apagat, sempre n'hem d'actualitzar l'estat del corresponent node a la base de dades.

```
elif content=="A_OFF\r\n":
    provaD=trama.Basedades(2,3333,4444,
        "OFF", "ON", a)

elif content=="B_OFF\r\n":
    provaD=trama.Basedades(1,1111,2222,
        "OFF", "ON", b)

elif content=="C_OFF\r\n":
    provaD=trama.Basedades(3,5555,6666,
        "OFF", "ON", c)
```

Finalment mirem si el missatge rebut és una confirmació provinent del punt final o node, informant que ha rebut el missatge o ordre anterior del coordinador. És a dir, per exemple si el coordinador, envia l'ordre al node A de que s'encengui, aquest respon al coordinador la confirmació que ha rebut aquest missatge correctament i ha executat l'acció demanada.

S'actualitzarà la base de dades amb l'estat actual dependent de quin node vingui la confirmació de recepció.

```
elif content=="ACK_A\r\n" :
    a=a+1
    provaD=trama . Basedades (2 ,1111 ,2222 ,
        "ON" , "ON" , a)

elif content=="ACK_B\r\n" :
    b=b+1
    provaD=trama . Basedades (1 ,3333 ,4444 ,
        "ON" , "ON" , b)

elif content=="ACK_C\r\n" :
    c=c+1
    provaD=trama . Basedades (3 ,5555 ,6666 ,
        "ON" , "ON" , c)

else :

    print ("Trama descartada")
```

En aquest últim tros del codi tractem tots els possibles errors en què es pot trobar el nostre programa. Per exemple, en el cas que desconnectem el sistema de cop, podem generar errors no desitjats, com estats no correctes. Així doncs amb aquestes excepcions podem prevenir aquests possibles efectes.

```
except serial.SerialException:
    print "ERROR: \_No\_device\_connected\_or\_wrong\_
        device"
    sys.exit()
except KeyboardInterrupt:
    print '\n\rEXIT: \_KeyboardInterrupt \_
    sys.exit()
except OSError:
    print "ERROR: \_No\_device\_connected\_or\_wrong\_
        device"
    sys.exit()
except IOError:
    print "ERROR: \_USB\_has\_disconnected"
    sys.exit()
```

3.3.2 Programació del programa carregat als nodes

A continuació es presenta el codi generat en cada un dels nodes, el qual és diferent per a cada node, ja que els missatges a transmetre no són els mateixos per cada un d'ells ni les accions a dur a terme.

El codi generat és carregat al Arduino de cada node, el qual s'encarrega de gestionar les accions a dur a terme. Per a carregar el codi al Arduino s'ha utilitzat un compilador de codi utilitzat al llarg del grau anomenat Makefile, que permet compilar el codi manera ràpida i senzilla. El codi referent del Makefile, es mostrarà en l'Annex juntament amb tot el codi del projecte. La comanda utilitzada per compilar el codi ha estat:

```
$ make xbee
```

Seguit de la següent comanda, utilitzada per a carregar el codi compilat al corresponent arduino connectat al ordinador.

```
$ make xbee-transfer
```

Es presenta el codi d'un dels nodes, en concret del node C, ja que només s'en vol mostrar el funcionament en general. En l'annex s'hi adjuntar-ha el codi referent a cada node.

Es mostra la funció inicial del programa, encarregada d'arrencar el funcionament del node.

Inicialitzem les variables necessàries i informem al coordinador que inicialment el node té la bombeta funcionant a mitja potència, que a la web s'interpreta amb una bombeta apagada. Acte seguit cridem la funció que ens executarà la funció principal, sempre que tinguem el programa en funcionament, ja que aquesta funció l'executarà sempre gràcies al 'while(true)'

```
int main(void) {  
  
    DDRB =0b00100000 ;  
    setup_pwm_tmr2(11);  
    setup_timer0();  
    setup_ADC(5,5,16);  
    serial_init();  
  
    put_message("C_OFF\r\n");  
    xbeesleep();  
    while(true) {  
        loop();  
    }  
    return 0;  
}
```

La següent funció s'executarà sempre que el programa estigui en funcionament. Primerament inicialitzem les variables a utilitzar en aquesta funció.

```
void loop(void){//estat loop
  uint8_t x, val;

  val = PINB & _BV(PB0); //Pin 8 arduino per lectura PIR
  uint8_t value=read8_ADC();
  start_ADC();

  /*Al arrancar el programa estat=1*/
```


En el següent codi podem apreciar que segueix l'esquema de la màquina d'estats del node C presentat anteriorment. Inicialment, al arrancarr el programa sempre entrarem en el 'case 1:' ja que s'ha definit que 'estat=1' al iniciar el programa. Tot seguit comprovem si el PIR ha detectat moviment, si ha detectat reiniciem el timer, posem a potència màxima la bombeta i preparem la màquina d'estats per a que avanci al següent estat.

En cas que el PIR no hagi detectat, comprovem si hem rebut ordre del coordinador, tot seguit, mirem si les dades que ha enviat el coordinador van dirigides al node en què ens trobem. Si aquestes comprovacions s'han complert, contestem al coordinador amb la confirmació de la recepció del missatge, reiniciem el timer i activem la bombeta a màxima potència, finalment actualitzem la màquina d'estats perquè pugui avançar al següent estat.

```

switch ( estat )
{
  case 1: /* Estat Espera*/

    if ( val > 0 ) { // Si detecta
      w = 1;
      TIFR1 |= 1 << OCF1A;
      xbeewake ();
      estat = 2;
    }

    else if ( serial_can_read () ) {
      q [ y ] = serial_get ();
      if ( q [ 0 ] == ' 3 ' ) {
        estat = 2;
        _delay_ms ( 1000 );
        put_message ( "ACK_C\r\n" );
        TIFR1 |= 1 << OCF1A;
        xbeewake ();
      }
    }
  break ;
}

```

En aquest estat podem veure que també segueix l'esquema de la màquina d'estats del node C presentada anteriorment.

Primerament mirem si el PIR ha detectat moviment, si detecta, reiniciem el timer i enviem al coordinador el missatge assegurant que la bombeta ha

encès. Perquè el node no envii contínuament missatges al coordinador per a dir-li que ha detectat s'ha utilitzat la variable 'w' en què aconseguim que el node només envii un cop al coordinador si ha detectat.

La següent comprovació, en cas de no haver detectat res per part del PIR, és comprovar si hem rebut alguna ordre del coordinador. En cas d'haver rebut comprovem que missatge rebut ha estat dirigit correctament al node en què ens trobem. Responem al coordinador amb la confirmació de què hem rebut el missatge correctament finalment reiniciem el timer.

La ultima comprovació, es duu a terme si el temps del timer ha arribat al seu màxim i el PIR no ha detectat, ni hem rebut res provinent del coordinador.

En aquest cas apaguem el timer i enviem al coordinador que la bombeta funciona en baix rendiment.

```
case 2:

    if (val > 0) {
        if (w == 1) {
            TIFR1 |= 1 << OCF1A;
            serial_put('C');
            w = 0;
            estat = 2;
        }
    }

    else if (serial_can_read()) {
        q[y] = serial_get();
        w = 1;
        if (q[0] == '3') {
            _delay_ms(1000);
            put_message("ACK_C\r\n");
            TIFR1 |= 1 << OCF1A;
            estat = 2;
        }
    }

    else if (((TIFR1) & 1 << OCF1A) ) { // Si el flag esta a
        1, ha arribat al top value
        if (c_TMR1 == 6) {
            c_TMR1 = 0;
            estat = 1; // Estat Slave
            xbeesleep();
            put_message("C_OFF\r\n");
        }
        c_TMR1++;
        w = 1;
        TIFR1 |= 1 << OCF1A;
    }
    break;
}
```

Cal tenir en compte que el codi presentat en aquest apartat no és el total de codi fet. La resta de codi i llibreries necessàries es presentaran en l'annex. L'objectiu d'haver presentat aquest codi és mostrar el funcionament general del sistema sense entrar en especificacions concretes.

3.4 Configuració del Sistema Web

3.4.1 Programació de la interfície web

Els arxius generats per a l'interfície estan guardats dins d'un domini, en concret dins de la carpeta 'httpdocs', on hi ha guardat l'estructura de la nostra pagina web, les imatges utilitzades i el codi font generat. Aquest domini s'ha comprat per un any a una empresa que es dedica al allotjament de dominis. Aquesta empresa s'encarrega en aquest cas de allotjar el domini.

Tota la web compleix amb el protocol HTTP, però cal tenir en compte que les dades entre el servidor web i el navegador web no van encriptades, per tant, si algú intentés interceptar la comunicació entre servidor i navegador, no hi trobaria molts problemes.

El plugin utilitzat per controlar l'accés al sistema web, l'ofereix la pròpia plataforma de Wordpress, en que aquest plugin anomenat 'Protected with password' es simplement un formulari POST, el qual comprova si la contrasenya introduïda coincideix amb la que hi ha guardada. Aquesta contrasenya, el creador del plugin no en dona detalls d'on es guarda, però segurament és guardada en una Cookie.

La primera pàgina generada anomenada 'On treballem' mostra la ubicació d'on s'ha implantat el sistema de manera fictícia, mitjançant un mapa pròpi de google, on podem apreciar les coordenades geogràfiques d'on es troba dit sistema.

Per a generar aquesta pàgina s'ha utilitzat el següent codi:

```
<h2><span style="color: #008000;"><strong>Parc de les  
Homilies d'Organyà </strong></span></h2>  
[wpgmza id="1"]
```

En codi presentat s'ha encapsulat el codi que genera el mapa amb la ubicació en '[wpgmza id="1"]'. Per a encapsular el codi s'ha utilitzat un plugin pròpi de Wordpress orientat a la utilització de mapes anomenat 'Google XML Sitemaps'.

La pàgina principal generada s'anomena 'Estat del Sistema' on hi podem trobar la representació de les bombetes del sistema, una taula que conté les vegades que s'han encés les llums i durant quant de temps, l'estat actual de les bombetes, permet la interacció amb les bombetes poden encendre-les.

A continuació es presentarà el codi necessari per a desenvolupar l'interfície web així com l'explicació necessària per a entendre'n el seu funcionament. Primerament s'ha hagut de generar un script en Javascript el qual hem guardat en una pàgina anomenada 'web.js', per tal que refresqui cada un cert temps el contingut de la pàgina, sense haver de refrescar la pàgina web al complet. D'aquesta manera l'usuari no pot apreciar com la pàgina es recarrega sempre. Simplement s'actualitzen les dades i les bombetes. Aquest script, fa recarregar una altra pàgina cada segon, que hem anomenat 'refresc.php' el qual conté el contingut principal de la pàgina web així com els components a refrescar.

És a dir, el contingut global de la pàgina web que conté l'estructura i les crides a altres pàgines és el 'Estat del sistema'. El que s'encarrega de carregar el contingut de la pàgina web s'anomena 'web.js', i finalment el que conté el contingut és el 'refresc.php'. Tot seguit es presenta el codi de la pàgina 'Estat del sistema', les pàgines 'refresc.php' i 'web.js' s'adjuntaran en l'annex, ja que es pretén entendre el funcionament del sistema web, sense entrar en detall.

Seguidament del script generat es crea un 'div' el qual encapsula un conjunt de dades com en una taula invisible per a l'usuari, permetent així poder moure un conjunt de dades de cop. En aquest div i encapsularem la taula amb les dades dels fanals i les mateixes bombetes. Dins d'aquest div comencem posant la primera bombeta que centrarem a una posició en concret juntament amb un botó a sota d'aquesta que permetrà a l'usuari encendre aquesta bombeta en el sistema.

Perquè en polsar el botó s'encengui la bombeta és necessari que aquest botó generi una acció. Aquesta acció fa referència a connectar-se a la base de dades i actualitzar el valor referent a l'estat de la bombeta amb la identificació corresponent per tal d'encendre la bombeta. Primerament ens hem de connectar a la base de dades, seleccionar quina taula conte les dades referents a l'acció a generar i una vegada trobada la taula s'hi genera una comanda que actualitzi les dades de la taula. En cas d'una connexió errònia amb la base de dades se'n donarà constància. El resultat d'aquesta connexió es guardarà en una variable que utilitzarem més endavant.

El següent tros de codi fa referència a la generació d'una bombeta amb el seu respectiu botó. Les altres dues bombetes s'en presentarà el codi al l'annex ja que el codi és molt semblant només canviant el nom de les variables utilitzades. El funcionament d'aquest node és el mateix per a la resta de nodes.

```
<script type="text/javascript" src="/wordpress_d/web.js
"></script>

<div id="refresh">
<div style="position: absolute; top: 500px; left: 328px">
<body>
<form name="form" method="post">
<input type="submit" name="button1" value="ON_Node_1"
/>
</form>
</body>
</div>

<?php

if(isset($_POST['button1'])) {
$linkz = mysql_connect('subiranatfg.com', 'subirana_tfg
', 'H6bwy34!');
    or die('No se pudo conectar: ' . mysql_error());
    mysql_select_db('dades_farola') or die('No se pudo
seleccionar la base de datos');
$queryz = 'UPDATE_dades_client SET_estat_client=1,
id_client=1_WHERE_1';

$resultz = mysql_query($queryz) or die('Consulta
fallida: ' . mysql_error());
}
?>
```


El següent codi fa referència a la taula que conté les dades referents a l'estat dels nodes, així com l'estat de les bombetes i el temps que han estat enceses des que el sistema ha arrancat. Primerament connectem amb la base de dades, en seleccionem la taula que desitgem fer les consultes i finalment guardem en unes variables el contingut d'aquestes consultes. La consulta a realitzar ha de ser de lectura de la base de dades, ja que volem saber l'estat de les bombetes. Tot seguit carreguem a la taula aquests valors llegits de la base de dades. Com que el contingut de la pàgina es refresca cada segon, podem veure les dades actualitzades cada segon també, ja que la càrrega del valor de l'estat de les bombetes el fa contínuament mentre la web està activa.

```

<?php
$linkp = mysql_connect( 'subiranatfg.com' , 'subirana_tfg
    ' , 'H6bwy34!' )
    or die( 'No_s_e_pudo_c_onectar:_ ' . mysql_error() );

    mysql_select_db( 'dades_farola' ) or die( 'No_s_e_pudo_
        seleccionar_la_base_de_datos' );
$queryp = 'SELECT_*_FROM_dades_WHERE_id=1';
$querypb = 'SELECT_*_FROM_dades_WHERE_id=2';
$querypc = 'SELECT_*_FROM_dades_WHERE_id=3';
$resultp = mysql_query( $queryp ) or die( 'Consulta_
    fallida:_ ' . mysql_error() );
$resultpb = mysql_query( $querypb ) or die( 'Consulta_
    fallida:_ ' . mysql_error() );
$resultpc = mysql_query( $querypc ) or die( 'Consulta_
    fallida:_ ' . mysql_error() );
$rsrp = mysql_fetch_array( $resultp , MYSQL_BOTH );
$rsrpb = mysql_fetch_array( $resultpb , MYSQL_BOTH );
$rsrpc = mysql_fetch_array( $resultpc , MYSQL_BOTH );
?>

<table border="1">
<tr>
<td></td>
<td><strong>NODE 1</strong></td>
<td><strong>NODE 2</strong></td>
<td><strong>NODE 3</strong></td>

```

```
</tr>
<tr>
<td><strong>Vegades ON</strong></td>
<td><?php echo $rspb[5]; ?></td>
<td><?php echo $rsp[5]; ?></td>
<td><?php echo $rspc[5]; ?></td>
</tr>
<tr>
<td><strong>Temps ON</strong></td>
<td><?php echo $rspb[5]*26.975; ?></td>
<td><?php echo $rsp[5]*26.975; ?></td>
<td><?php echo $rspc[5]*26.975; ?></td>
</tr>
</table>
```

Establím primer la connexió amb la base de dades, seguidament de la selecció de la taula on farem les consultes. En aquest cas la consulta a fer serà de lectura el qual guardarem en una variable per a més endavant mirar-ne el contingut.

Comprovem que el contingut d'aquesta variable fa referència a què l'estat de la bombeta és 'ON', està encesa, i per tant hem de mostrar una bombeta que representi que aquesta està encesa.

En cas que el contingut de la variable sigui 'OFF', estigui apagada la bombeta, s'ha de mostrar una bombeta que representi que aquesta està apagada. Finalment, al final del programa hem de tancar la connexió feta amb la base de dades.

```
<?php
// Conectando, seleccionando la base de datos
$link = mysql_connect('subiranatfg.com', 'subirana_tfg',
    , 'H6bwy34!');
    or die('No se pudo conectar: ' . mysql_error());
//echo 'Connected successfully'. "\n";
mysql_select_db('dades_farola') or die('No se pudo
    seleccionar la base de datos');

//NODE 1
// Realizar una consulta MySQL
$query = 'SELECT * FROM dades WHERE id=1';

$result = mysql_query($query) or die('Consulta fallida:
    ' . mysql_error());

$rs = mysql_fetch_array($result, MYSQL_BOTH);

if ($rs[3]==ON) {
    $comptador_node_1++;
?>
```

```

<div style="position: absolute; top: 350px; left: 650px">
<p><strong>Node 2</strong></p>
<a href="http://www.subiranatfg.com/wordpress_d/wp-
content/uploads/2015/08/BOmbillaon.png"></a>
</div>

<?php
}

if ($rs[3]==OFF) {
?>

<div style="position: absolute; top: 350px; left: 650px">
<p><strong>Node 2</strong></p>
<a href="http://www.subiranatfg.com/wordpress_d/wp-
content/uploads/2015/08/bombilla_off.jpg"></a>
</div>

<?php
}

mysql_close($link);
?>

```

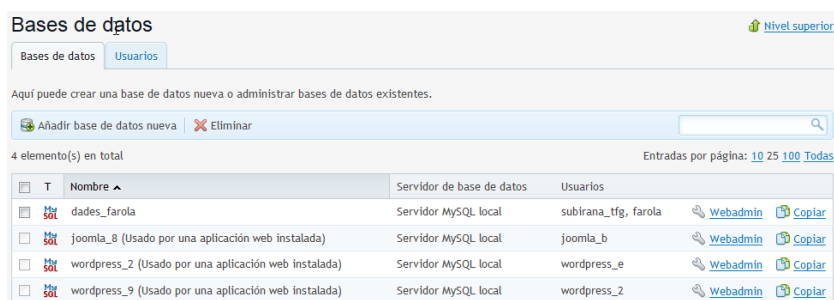
3.4.2 Programació de la Base de dades

La base de dades creada s'ha anomenat 'dades farola' el qual conté dues taules, la taula 'dades' i 'dades client'. En la taula 'dades' trobem tota la informació referent a l'estat dels fanals físics, en la taula de 'dades client' hi trobem la informació referent a l'estat dels fanals de la web. El motiu principal per al qual s'ha dividit en dues taules és a fi d'evitar possibles problemes d'escriptura/lectura en la base de dades, ja que la connexió i consulta a la base de dades no és immediata, ens podem trobar amb retards.

Aquests retards poden presentar un estat d'un fanal que no és correcte. És a dir, per exemple, ens podem trobar que si tenim només una taula referent a l'estat dels fanals, tant físiques com de la web, en polsar la bombeta de la web, aquesta escriurà a la base de dades segons ha encès, però en un instant molt proper, el fanal físic s'ha apagat i també escriu a la base de dades, però en aquest cas dóna constància que s'ha apagat. Ens podem trobar que a causa d'algun retard no puguem controlar l'estat final del fanal.

Primerament hem de crear la base de dades on crearem les taules. Per a crear les base de dades s'ha fet a partir del sistema de gestió de base de dades anomenat MySQL, el qual permet crear base de dades de manera senzilla i molt gràfica.

En la imatge podem apreciar les base de dades utilitzades, entre elles es troba la base de dades anomenada 'dades farola'. La resta de taules que podem veure són utilitzades per al disseny de la interfície i els plugins utilitzats.



The screenshot shows the MySQL database management interface. At the top, there's a header 'Bases de datos' with a 'Nivel superior' link. Below it, there are tabs for 'Bases de datos' and 'Usuarios'. A message states: 'Aquí puede crear una base de datos nueva o administrar bases de datos existentes.' There are buttons for 'Añadir base de datos nueva' and 'Eliminar'. Below this, there's a search bar and a table listing databases. The table has columns for 'Nombre', 'Servidor de base de datos', and 'Usuarios'. There are also icons for 'Webadmin' and 'Copiar' for each database.

	Nombre	Servidor de base de datos	Usuarios		
<input type="checkbox"/>	dades_farola	Servidor MySQL local	subirana_tfg, farola	Webadmin	Copiar
<input type="checkbox"/>	joomla_8 (Usado por una aplicación web instalada)	Servidor MySQL local	joomla_b	Webadmin	Copiar
<input type="checkbox"/>	wordpress_2 (Usado por una aplicación web instalada)	Servidor MySQL local	wordpress_e	Webadmin	Copiar
<input type="checkbox"/>	wordpress_9 (Usado por una aplicación web instalada)	Servidor MySQL local	wordpress_2	Webadmin	Copiar

Figura 3.14: Bases de dades

+ Options			id	locX	locY	estat	estatsistema	vegades
<input type="checkbox"/>	Edit Copy Delete	1	1111	2222	OFF	ON	16	
<input type="checkbox"/>	Edit Copy Delete	2	3333	4444	ON	OFF	0	
<input type="checkbox"/>	Edit Copy Delete	3	5555	6666	ON	ON	9	

Figura 3.15: Creació de una taula

Acte seguit podem crear les taules on hi guardarem un conjunt d'informació. Per a crear la taula anomenada 'dades' ho hem fet de la següent manera:

```
CREATE TABLE IF NOT EXISTS 'dades' (
  'id' int(2) NOT NULL,
  'locX' int(5) NOT NULL,
  'locY' int(5) NOT NULL,
  'estat' varchar(5) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

INSERT INTO dades id, locX, locY, estat VALUES 1, 1234,
2345, 'ON';
```

+ Options			id	locX	locY	estat	estatsistema	vegades
<input type="checkbox"/>	Edit Copy Delete	1	1111	2222	OFF	ON	16	
<input type="checkbox"/>	Edit Copy Delete	2	3333	4444	ON	OFF	0	
<input type="checkbox"/>	Edit Copy Delete	3	5555	6666	ON	ON	9	

Figura 3.16: Taula dades

En aquesta taula es pot apreciar quins seran els valors que hi guardarem. Cal tenir en compte que els paràmetres 'estatsistema' i 'vegades' han estat inserits un cop la taula ja era creada. Aquests valors hem definit que no puguin tindre valors nuls, ja que la identificació, localització i estat de sistema s'ha pensat que sempre tindran certs tipus d'estat. És a dir, la identificació només podrà contenir la identificació dels nodes, que referenciem numèricament. En

la localització ens hi adrecem només a partir de coordenades numèriques i finalment en l'estat només hi han dos estats possibles, 'ON' o 'OFF'.

Seguidament de la creació de la taula hi inserim un conjunt de valors per mostrar la consulta que es fa servir en cas de voler introduir dades a la taula.

Per a actualitzar les dades de la base de dades, fem servir la consulta 'UPDATE', a continuació en mostrem un exemple:

```
UPDATE dades SET locX=0001, locY=0002, estat='OFF'  
WHERE id=1;
```

Per a seleccionar dades de la taula per tal de llegir-les ho fem mitjançant la següent estructura de comanda:

```
SELECT estat from dades_client WHERE id=1;
```

Capítol 4

Conclusions

En aquest projecte s'han pogut assolir la gran majoria dels objectius presentats en un començament. S'ha aconseguit crear un sistema capaç de monitorar i gestionar eficaçment la il·luminació, per tal de poder tindre control sobre un conjunt de fanals, així com poder consultar el estat de cada un d'aquest fanals.

Remarcar la importància d'aquest tipus de projecte en un futur proper, on es preveu un augment considerable de sistemes capaços de generar un estalvi energètic ja que la societat cada cop presenta més demanda de recursos, i aquests són més i més escassos. Si no es proposen sistemes capaços de presentar solucions a aquesta demanda en augment ens podem arribar a trobar en una situació en que la falta d'energia no pugui ser coberta per l'energia que produïm.

Cal tenir en compte que aquest projecte ha estat l'inici d'un projecte molt més llarg, degut a que presenta una continuació on els fanals hauran de ser de tamany real amb les especificacions necessàries, s'hauran de crear nous prototips adaptats als fanals de tamany real, el sistema haurà de permetre gestionar un volum gran de nodes així com presentar una interfície web més atractiva per a l'usuari i amb més funcionalitats com poder afegir o treure fanals d'un sistema triat,

A nivell d'objectius personals, s'han aconseguit tots els plantejats inicialment, ja que s'ha après nous llenguatges de programació fins ara desconeguts, a més d'entendre la necessitat de promoure i crear sistemes relacionats amb l'estalvi energètic.

Bibliografia

- [1] IDAE, <http://www.idae.es/index.php/id.644/relmenu.429/mod.pags/mem.detalle> Setembre, 2015
- [2] Owlet, <http://www.schreder.com> Setembre, 2015
- [3] Luix, <http://www.iluminacionluix.com/> Setembre, 2015
- [4] Lumimotion, http://www.lighting.philips.es/application_areas/lighting_control/outdoor_lumimotion.wpd Setembre, 2015
- [5] MySQL, <http://dev.mysql.com/doc/refman/5.5/en/index.html> Setembre, 2015
- [6] Fritzing, <http://fritzing.org/home/> Setembre, 2015
- [7] Manual PHP, <https://secure.php.net/manual/es/> Setembre, 2015
- [8] PHP i MySQL, http://www.w3schools.com/php/php_mysql_update.asp Setembre, 2015
- [9] Manual SQL, <http://www.tutorialspoint.com/mysql/mysql-update-query.htm> Setembre, 2015
- [10] Especificacions Arduino, <https://www.arduino.cc/en/Main/ArduinoBoardUno#techspecs> Setembre, 2015
- [11] Introducció a l'ús de Shield, <https://learn.sparkfun.com/tutorials/xbee-shield-hookup-guide> Setembre, 2015
- [12] Especificacions Xbee, https://cdn.sparkfun.com/datasheets/Wireless/Zigbee/ds_xbeezbmodules.pdf Setembre, 2015
- [13] Introducció al Javascript, <http://librosweb.es/libro/javascript/> Setembre, 2015

- [14] Manual d'ajuda de Javascript, <http://www.w3schools.com/js/> Setembro, 2015
- [15] Manual Latex, https://es.wikibooks.org/wiki/Manual_de_LaTeX Setembro, 2015
- [16] Ajuda Latex, <http://nokyotsu.com/latex/curso.html> Setembro, 2015

Capítol 5

Annex

A continuació es presentarà tot el codi utilitzat en el projecte.

- main.py - Programa principal

```
import xbee
import trama
import serial, sys
import binascii
from time import sleep

import MySQLdb
from datetime import date, datetime, timedelta

if __name__ == '__main__':
    try:
        print "Running"
        ser=sys.argv[-1]
        xbee = xbee.XBee(ser)
        trama =trama.Trama()
        #AdreÀ§a Node A
        addr1="13a200408b66e1" #Node B
        #AdreÀ§a Node B
        addr2="13a200408b6061" #NODE A
        addr3="13a200408b6088" #Node C xbee3
        prova=trama.Basedades(1,1111,2222,"OFF","OFF",0)
        sleep(0.5)
        prova=trama.Basedades(2,3333,4444,"OFF","OFF",0)
        sleep(0.5)
        prova=trama.Basedades(3,5555,6666,"OFF","OFF",0)
        sleep(0.5)

        a=0
        b=0
        c=0
```

```

while(True):

    missatge = xbee.Receive()
    missi=trama.Basedades2()

    if missi[0]==1:
        if missi[1]==1:
            xbee.SendStr('1', int(addr2,16))
            provaZ=trama.Basedades3(0)
        elif missi[1]==2:
            xbee.SendStr('2', int(addr1,16))
            provaZ=trama.Basedades3(0)
        elif missi[1]==3:
            xbee.SendStr('3', int(addr3,16))
            provaZ=trama.Basedades3(0)

    elif missatge:
        content = missatge[14:-1].decode('utf-8', 'ignore')
        print("Missatge_RF_rebut:_" + content)
        if content=='A':
            a=a+1
            xbee.SendStr('2', int(addr1,16))#
                adreça del xbee 1 - node B
            provaD=trama.Basedades(2,1111,2222,
                "ON", "ON", a)
                elif content=='B':
            b=b+1
            c=c+1
            xbee.SendStr('1', int(addr2,16))
            xbee.SendStr('3', int(addr3,16))
            provaD=trama.Basedades(1,1111,2222,
                "ON", "ON", b)
            sleep(0.5)
            provaC=trama.Basedades(3,5555,6666,
                "ON", "ON", c)

```

```

elif content=='C':
    c=c+1
    xbee.SendStr('2', int(addr1,16))#
        adreÃ§a del xbee 1 - node B
    provaC=trama.Basedades(3,5555,6666,
        "ON", "ON", c)

elif content=="A_OFF\r\n":
    provaD=trama.Basedades(2,3333,4444,
        "OFF", "ON", a)

elif content=="B_OFF\r\n":
    provaD=trama.Basedades(1,1111,2222,
        "OFF", "ON", b)

elif content=="C_OFF\r\n":
    provaD=trama.Basedades(3,5555,6666,
        "OFF", "ON", c)

elif content=="ACK_A\r\n":
    a=a+1
    provaD=trama.Basedades(2,1111,2222,
        "ON", "ON", a)

elif content=="ACK_B\r\n":
    b=b+1
    provaD=trama.Basedades(1,3333,4444,
        "ON", "ON", b)

elif content=="ACK_C\r\n":
    c=c+1
    provaD=trama.Basedades(3,5555,6666,
        "ON", "ON", c)

```

```
        else:

            print("Trama descartada")

except serial.SerialException:
    print "ERROR: No device connected or wrong device"
    sys.exit()
except KeyboardInterrupt:
    print '\n\rEXIT: KeyboardInterrupt'
    sys.exit()
except OSError:
    print "ERROR: No device connected or wrong device"
    sys.exit()
except IOError:
    print "ERROR: USB has disconnected"
    sys.exit()
```

- trama.py - Programa creat per a llegir i escriure de la base de dades

```
import time
import MySQLdb
from datetime import date, datetime, timedelta

class Trama():
    """
    Classe Trama.
    """
    def Cadena(self, missatge):

        self.trama = missatge.split(";")
        print(" Això es la trama split")
        print self.trama
        trama = self.trama
        return trama

    def Basedades(self, iden, localx, localy, estatl,
        estatsist, vegad):

        sql_in = """UPDATE dades SET locX='%s ', locY='%s ',
        estat='%s ', estatsistema='%s ', vegades
        ='%s ' WHERE id='%s ';""" %(localx, localy,
        estatl, estatsist, vegad, iden)

        try:
            cursor.execute(sql_in)
            db.commit()
```



```

except :
    db.rollback ()
    cursor.close ()
    db.close ()

db = MySQLdb.connect ("subiranatfg.com", "
    subirana_tfg", "H6bwy34!", "dades_farola")
cursor = db.cursor ()
sql_sel = """SELECT_*_from_dades"""

try :
    cursor.execute (sql_sel)

except :
    db.rollback ()
    cursor.close ()
    db.close ()

def Basedades2 (self):

    db = MySQLdb.connect ("subiranatfg.com", "
        subirana_tfg", "H6bwy34!", "dades_farola")
    cursor = db.cursor ()
    sql_sel = """SELECT_*_from_dades_client"""

    try :
        cursor.execute (sql_sel)
        result = cursor.fetchall ()

        for element in result:

            return element [0], element [1]

```

```
except:
    db.rollback()
    cursor.close()
    db.close()

def Basedades3(self ,num):

    db = MySQLdb.connect("subiranatfg.com", "
        subirana_tfg", "H6bwy34!", "dades_farola")
    cursor = db.cursor()

    sql_in = """UPDATE_dades_client_SET_
        estat_client=%s' _WHERE_1;""" %(num)

    try:
        cursor.execute(sql_in)
        db.commit()
    except:
        db.rollback()
        cursor.close()
        db.close()
```

- xbee.py - Programa utilitzar per formatar els missatges entre Xbee's

```
import serial
from collections import deque

class XBee():
    """
    Classe XBee
    """
    RxBuff = bytearray()
    RxMessages = deque()

    def __init__(self, serialport, baudrate=9600):
        """
        Inicialitza el port s'rie corresponent.

        *param** serialport **: port s'rie
        *param** baudrate **: baudrate
        """
        self.serial = serial.Serial(port=serialport,
                                    baudrate=baudrate)

    def Receive(self):
        """
        Rep dades del port s'rie i omple el buffer
        amb missatges correctes.
        Retorna el segon quart missatge de la cua si
        existeix.

        *return*: Missatge
        *rtype*: byte object bytearray
        """
        remaining = self.serial.inWaiting()
        while remaining:
            chunk = self.serial.read(remaining)
            remaining -= len(chunk)
            self.RxBuff.extend(chunk)

        msgs = self.RxBuff.split(bytes(b'\x7E'))
```

```

for msg in msgs[:-1]:
    self.Validate(msg)

self.RxBuff = (bytearray() if self.Validate(
    msgs[-1]) else msgs[-1])

if self.RxMessages:
    return self.RxMessages.popleft()
else:
    return None

def Validate(self, msg):
    """
    ..:::Analitza els bytes dels objectes bytearray per
    ..verificar que el missatge estÃ formatat segons un
    missatge XBee.
    ..:::I retorna True o False, segons la validesa del
    missatge.

    ..:::*param**msg*: Un missatge XBee entrant.
    ..:::*type msg*: bytearray
    ..:::*return*: True o False
    ..:::*rtype*: bool
    ..:::
    """
    # 9 bytes Ãs la mÃnima longitud per ser un
    # frame Rx.
    # LSB, MSB, Type, Source Address(2), RSSI,
    # Options, 1 byte data, checksum
    if (len(msg) - msg.count(bytes(b'0x7D')) < 9:
        return False

    # Tots els bytes del missatge han de ser '
    # unescaped' abans de validar el contingut
    frame = self.Unescape(msg)

    LSB = frame[1]

```

```

# Frame (menys checksum) ha de contenir com a
# mÃnim la longitud igual al LSB
if LSB > (len(frame[2:]) - 1):
    return False

# Valida checksum
if (sum(frame[2:3+LSB]) & 0xFF) != 0xFF:
    return False

#print("Rx: " + self.Format(bytearray(b'\x7E')
    + msg))
self.RxMessages.append(frame)
return True

def SendStr(self , msg, addr=0x000000000000FFFFFFFE,
options=0x01, frameid=0x00):

    return self.Send(msg.encode('utf-8'), addr,
        options, frameid)

def Send(self , msg, addr=0x000000000000FFFFFFFE,
options=0x01, frameid=0x00):

    if not msg:
        return 0

```

```

# Es formata el frame segons les directius XBee
hexs = '7E_00_{:02X}_10_{:02X}_00_{:02X}_{:02X}
_{:02X}_{:02X}_{:02X}_{:02X}_{:02X}_FF_FE_00
_00'.format(
    len(msg) + 14,          # LSB (longitud)
    frameid,
    (addr & 0xFF000000000000) >> 48, #
        Adre a de dest  / a partir del 13
    (addr & 0x00FF0000000000) >> 40,
    (addr & 0x0000FF00000000) >> 32,
    (addr & 0x000000FF000000) >> 24,
    (addr & 0x00000000FF0000) >> 16,
    (addr & 0x0000000000FF00) >> 8,
    (addr & 0x000000000000FF)      #
        Destination address low byte
)
frame=bytearray.fromhex(hexs)

# Afegeix el contingut del missatge
frame.extend(msg)

# Es calcula el byte de checksum
frame.append(0xFF - (sum(frame[3:]) & 0xFF))

# Escape any bytes containing reserved
characters
#frame = self.Escape(frame)

#print("Tx: " + self.Format(frame))
return self.serial.write(frame)

def Unescape(self, msg):
    if msg[-1] == 0x7D:
        return None

```

```

    out = bytearray()
    skip = False
    for i in range(len(msg)):
        if skip:
            skip = False
            continue

        if msg[i] == 0x7D:
            out.append(msg[i+1] ^ 0x20)
            skip = True
        else:
            out.append(msg[i])

    return out

def Escape(self, msg):
    """
    Faun 'Escape' dels carÀcters reservats abans
    d'enviar el missatge XBee.

    *param**msg*: Un byte o objecte bytearray
    que contÀ el missatge original a ser enviat.
    *return**escaped*: Un objecte bytearray per
    a ser enviat per XBee en mode API.
    """
    escaped = bytearray()
    # Bytes reservats en XBee mode API
    reserved = bytearray(b"\x7E\x7D\x11\x13")

```

```
        escaped.append(msg[0])
    for m in msg[1:]:
        if m in reserved:
            escaped.append(0x7D)
            escaped.append(m ^ 0x20)
        else:
            escaped.append(m)

    return escaped

def Format(self, msg):

    return "␣".join("{:02x}".format(b) for b in msg
)
```


- Makefile - Programa utilitzat per a carregar els programes dins de l'arduino.

```
CC=avr-gcc
CPPFLAGS=-DF_CPU=16000000UL
CFLAGS=-std=c99 -Os -mmcu=atmega328p -fshort-enums -
    llibpbn
LDFLAGS=-std=c99 -mmcu=atmega328p

vpath lib% libpbn

xbee: lcd.o twiAVR.o pwm_tmr2.o tmr0.o -lpbn
xbee.o: lcd.h pwm_tmr2.h tmr0.h utils.h

prova_lcd: lcd.o twiAVR.o -lpbn
prova_lcd.o: lcd.h

lcd.o: twiAVR.o

%.hex: %
    avr-objcopy -Oihex $< $@

%-transfer: %.hex
    avrdude -c arduino -p atmega328p -P /dev/
        ttyACM0 -U $<

.PHONY: clean
.PHONY: picocom
clean:
    \rm -f *~ *.o *.s *.hex xbee

picocom:
    picocom /dev/ttyACM0 -b9600
```

- XBEE.C - Programa carregat a cada arduino per a funcionar com a node.

```
#include <pbn.h>
#include <math.h>
#include <inttypes.h>
#include <stdbool.h>
#include <stdint.h>
#include <util/delay.h>
#include <avr/io.h>
#include <stdio.h>
#include <stdlib.h>
#include <avr/interrupt.h>

#include "tmr0.h"
#include "pwm_tmr2.h"
typedef uint8_t taula [64];
taula t,q;
//ADMUX bits
#define refs1 0b10000000
#define refs0 0b01000000
#define adlar 0b00100000
#define mux3 0b00001000
#define mux2 0b00000100
#define mux1 0b00000010
#define mux0 0b00000001
//Multiplexor input MUX3|MUX2|MUX1|MUX0
#define adc0 0b0000
#define adc1 0b0001
#define adc2 0b0010
#define adc3 0b0011
#define adc4 0b0100
#define adc5 0b0101
#define adc6 0b0110
#define adc7 0b0111
#define adc8 0b1000 //temperature sensor
#define adc1_1V 0b1110 //14
#define adcgnd 0b1111 //15
```

```

//Voltage Reference REFS1|REFS0
#define vAREF 0b00000000
#define v5V 0b01000000
#define v1_1V 0b11000000

//ADCSRA bits
#define aden 0b10000000
#define adsc 0b01000000
#define adps2 0b00000100
#define adps1 0b00000010
#define adps0 0b00000001
// prescaler ADSP2|ADSP1|ADSP0
#define pre2 0b001
#define pre4 0b010
#define pre8 0b011
#define pre16 0b100
#define pre32 0b101
#define pre64 0b110
#define pre128 0b111

//DIDR0 combinations
//to enable 2 inputs DIDR0=adc5d&adc4d
#define adc5d 0b00011111
#define adc4d 0b00101111
#define adc3d 0b00110111
#define adc2d 0b00111011
#define adc1d 0b00111101
#define adc0d 0b00111110
#define adcx d 0b00111111
int y=0;
int estat=1;//Estat "Espera"
int mom=0;
int w=1;
uint8_t c_TMR1=0;

void setup_ADC(uint8_t adc_input ,uint8_t v_ref ,uint8_t
adc_pre){
uint8_t adc ,adcd ,vref ,pre ;

```

```

//adc_input (0-5 (default=5),8 TÂa, 14 1.1V, 15 GND
  switch (adc_input){
    case 0 :adc=adc0;adcd=adc0d; break;
    case 1 :adc=adc1;adcd=adc1d; break;
    case 2 :adc=adc2;adcd=adc2d; break;
    case 3 :adc=adc3;adcd=adc3d; break;
    case 4 :adc=adc4;adcd=adc4d; break;
    case 5 :adc=adc5;adcd=adc5d; break;
    case 8 :adc=adc8;adcd=adcxd; break;
    case 14:adc=adc1_1V;adcd=adcxd; break;
    case 15:adc=adcgnd;adcd=adcxd; break;
    default : adc=adc5; adcd=adc5d; break;
  }

//vref 0 (AREF), 1(1.1V), default=5 (5V)
  switch (v_ref){
    case 0:vref=vAREF; break;
    case 1:vref=v1_1V; break;
    case 5:vref=v5V; break;
    default : vref=v5V; break;
  }

//adc_pre 2,4,8, default=16,32,64,128
  switch (adc_pre){
    case 2:pre=pre2; break;
    case 4:pre=pre4; break;
    case 8:pre=pre8; break;
    case 16:pre=pre16; break;
    case 32:pre=pre32; break;
    case 64:pre=pre64; break;
    case 128:pre=pre128; break;
    default : pre=pre16; break;
  }

  ADMUX = vref|adc|adlar;
  ADCSRA = aden|pre;
  DIDR0 = adcd;
}

```

```

uint8_t read8_ADC() {
    while (bit_is_set(ADCSRA, ADSC));
    return ADCH;
}

void start_ADC() {
    ADCSRA |= (1 << ADSC);
}

static void put_message(taula c) {
    for (int i=0; c[i] != '\0'; i++) {
        serial_put(c[i]);
    }
}

void xbeewake(void) {
    //put_message("Xbee woke up\r\n\0");
    //PORTB |= _BV(PORTB5);
    set_pwm_tmr2(255);
}

void xbeesleep(void) {
    //put_message("Xbee slept\r\n\0");
    //PORTB &= ~_BV(PORTB5);
    set_pwm_tmr2(50);
}

```

```

void setup_timer0(void) {
    //Mode 4 - CTC

    /* ((T1*16MHz)/prescaler)-1=OCR1A
       T1=3s; Prescaler=1024;
       —>OCR1A=46874

       Aquest timer esta programat per a saltar
       aproximadament cada 3 segons

    */
    TCCR1A=0x00; //B7 i B6 OC1A Set on compare match
    TCCR1B=0b00001101; //B4=WGM13 i B3=WGM12 per tenir
        mode ctc, B2,B1,B0 per definir prescaler de 1024
    OCR1A=65000; //Top value
    //TIMSK1=0b00000000;
    //TCNT1=0; //Comptador que es comparat amb el OCR1A(
        top value)
    // TIFR1 &= 0 << (OCF1A); //Flag, si val 1 el timer ha
        trobat el top value, el posema 0 per reiniciar
        el flag
}

void desactiva_timer0(void) {
    TCCR1B=0b00000000; //No hi ha clock
}

void loop(void) { //estat loop
    uint8_t x, val;

```

```

val = PINB & _BV(PB0); //Pin 8 arduino per lectura PIR
uint8_t value=read8_ADC();
start_ADC();

/*Al arrancar el programa estat=1*/

switch (estat)
{
  case 1: /*Estat Espera*/

    if(val>0){ //Si detecta
      w=1;
      TIFR1 |= 1<<OCF1A;
      xbeewake(); //Il·luminacio max
      estat=2;
      //serial_put('B');

    }
    else if(serial_can_read()){
      q[y]=serial_get();
      if(q[0]== '3'){
        estat=2;
        _delay_ms(1000);
        put_message("ACK_C\r\n");
        TIFR1 |= 1<<OCF1A;
        xbeewake();

      }
    }

  break;
}

```

```

case 2: /* Estat ON*/

    if (val > 0) {

        if (w == 1) {
            TIFR1 |= 1 << OCF1A;
            serial_put('C');
            w = 0;
            estat = 2;
        }
        //      serial_put('A'); // Notifica al coord
    }
    // else if (val <= 0) {
        // w = 1;
        // }

    else if (serial_can_read()) {
        q[y] = serial_get();
        w = 1;
        if (q[0] == '3') {
            _delay_ms(1000);
            put_message("ACK_C\r\n");
            TIFR1 |= 1 << OCF1A;
            estat = 2;
        }
    }

    else if (((TIFR1) & 1 << OCF1A) ) { // Si el flag esta a
        1, ha arribat al top value
    }

```



```
        if(c_TMR1==6){
            c_TMR1=0;
            estat=1;//Estat Slave
            xbeesleep();
            put_message("C_OFF\r\n");
        }
        c_TMR1++;
        w=1;
        TIFR1 |= 1<<OCF1A;

    }

    break;

}

}

int main(void){

    DDRB =0b00100000;
    setup_pwm_tmr2(11);
    setup_timer0();//Borra flag timer
    setup_ADC(5,5,16);
    serial_init();

    put_message("C_OFF\r\n");
    //_delay_ms(3000);//Temps que veurem el LED encès
    xbeesleep();
    while(true){
        loop();
    }
    return 0;
}
```

- web.js - Programa utilitzat a la web per a refrescar la pàgina.

```
function autoRefresh_div()
{
    jQuery("#refresh").load("http://www.
        subiranatfg.com/wordpress_d/
        refresc.php");// a function which
        will load data from other file
        after x seconds

}
setInterval('autoRefresh_div()', 1000); //
    refresh div after 5 secs
```

- refresc.php - Programa que conté tot el contingut de la web

```
<div id="refresh">
<div style="position: absolute; top: 500px; left: 328px">
<body>
<form name="form" method="post">
<input type="submit" name="button1" value="ON_Node_1"
  />
</form>
</body>
</div>

<?php

if(isset($_POST['button1'])) {
$linkz = mysql_connect('subiranatfg.com', 'subirana_tfg', 'H6bwy34!');
  or die('No se pudo conectar: ' . mysql_error());
  mysql_select_db('dades_farola') or die('No se pudo seleccionar la base de datos');
$queryz = 'UPDATE dades_client SET estat_client=1, id_client=1 WHERE 1';

$resultz = mysql_query($queryz) or die('Consulta fallida: ' . mysql_error());
}
?>

<div style="position: absolute; top: 500px; left: 628px">
<body>
<form name="form" method="post">
<input type="submit" name="button2" value="ON_Node_2"
  />
</form>
</body>
</div>
```

```

<?php

if(isset($_POST['button2'])) {
$linkz2 = mysql_connect('subiranatfg.com', '
    subirana_tfg', 'H6bwy34!')
    or die('No se pudo conectar: ' . mysql_error());
    mysql_select_db('dades_farola') or die('No se pudo
        seleccionar la base de datos');
$queryz2 = 'UPDATE dades_client SET estat_client=1,
    id_client=2 WHERE 1';

$resultz2 = mysql_query($queryz2) or die('Consulta
    fallida: ' . mysql_error());

}
?>

<div style="position: absolute; top: 500px; left: 928px">
<body>
<form name="form" method="post">
<input type="submit" name="button3" value="ON Node 3"
    />
</form>
</body>
</div>

<?php

if(isset($_POST['button3'])) {
$linkz3 = mysql_connect('subiranatfg.com', '
    subirana_tfg', 'H6bwy34!')
    or die('No se pudo conectar: ' . mysql_error());
    mysql_select_db('dades_farola') or die('No se pudo
        seleccionar la base de datos');
$queryz3 = 'UPDATE dades_client SET estat_client=1,
    id_client=3 WHERE 1';

$resultz3 = mysql_query($queryz3) or die('Consulta
    fallida: ' . mysql_error());

```

```

}
?>
<?php
$linkp = mysql_connect('subiranatfg.com', 'subirana_tfg', 'H6bwy34!');
        or die('No se pudo conectar:_' . mysql_error());

        mysql_select_db('dades_farola') or die('No se pudo seleccionar la base de datos');
$queryp = 'SELECT_*_FROM_dades_WHERE_id=1';
$querypb = 'SELECT_*_FROM_dades_WHERE_id=2';
$querypc = 'SELECT_*_FROM_dades_WHERE_id=3';
$resultp = mysql_query($queryp) or die('Consulta fallida:_' . mysql_error());
$resultpb = mysql_query($querypb) or die('Consulta fallida:_' . mysql_error());
$resultpc = mysql_query($querypc) or die('Consulta fallida:_' . mysql_error());
$rsrp = mysql_fetch_array($resultp, MYSQL_BOTH);
$rsrpb = mysql_fetch_array($resultpb, MYSQL_BOTH);
$rsrpc = mysql_fetch_array($resultpc, MYSQL_BOTH);
?>

<table border="1">
<tr>
<td></td>
<td><strong>NODE 1</strong></td>
<td><strong>NODE 2</strong></td>
<td><strong>NODE 3</strong></td>
</tr>
<tr>
<td><strong>Vegades ON</strong></td>
<td><?php echo $rsrpb[5]; ?></td>
<td><?php echo $rsrp[5]; ?></td>
<td><?php echo $rsrpc[5]; ?></td>
</tr>
<tr>

```

```

<td><strong>Temps ON</strong></td>
<td><?php echo $rsqb[5]*26.975; ?></td>
<td><?php echo $rsp[5]*26.975; ?></td>
<td><?php echo $rspc[5]*26.975; ?></td>
</tr>
</table>

<?php

$link = mysql_connect( 'subiranatfg.com', 'subirana_tfg',
    , 'H6bwy34!' )
    or die( 'No se pudo conectar: ' . mysql_error() );

mysql_select_db( 'dades_farola' ) or die( 'No se pudo
    seleccionar la base de datos' );

//NODE 1

$query = 'SELECT * FROM dades WHERE id=1';

$result = mysql_query( $query ) or die( 'Consulta fallida:
    ' . mysql_error() );

$rs = mysql_fetch_array( $result , MYSQL_BOTH );

if ( $rs[3]==ON ) {
    $comptador_node_1++;
?>
<div style="position: absolute; top: 350px; left: 650px">
<p><strong>Node 2</strong></p>
<a href="http://www.subiranatfg.com/wordpress/wp-content/uploads/2015/08/BOMBillaon.png"></a>
</div>

```

```

<?php
}

if ( $rs[3]==OFF) {
?>

<div style=" position : absolute ; top : 350px ; left : 650px ">
<p><strong>Node 2</strong></p>
<a href=" http://www.subiranatfg.com/wordpress_d/wp-
content/uploads/2015/08/bombilla_off.jpg"></a>
</div>

<?php
}

mysql_close( $link );
?>

<?php

$linkB = mysql_connect( 'subiranatfg.com' , 'subirana_tfg
' , 'H6bwy34!' )
or die( 'No se pudo conectar : ' . mysql_error() );

mysql_select_db( 'dades_farola' ) or die( 'No se pudo
seleccionar la base de datos ' );

//NODE_2

$queryB = 'SELECT * FROM dades WHERE id=2';
$resultB = mysql_query( $queryB ) or die( 'Consulta
fallida : ' . mysql_error() );

```

```

$rsB = mysql_fetch_array($resultB , MYSQL_BOTH);
if ($rsB[3]==ON) {
?>
<div style="position: absolute; top: 350px; left: 350px">
<p><strong>Node 1</strong></p>
<a href="http://www.subiranatfg.com/wordpress_d/wp-
content/uploads/2015/08/BOmbillaon.png"></a>
</div>

<?php
}
if ($rsB[3]==OFF) {
?>

<div style="position: absolute; top: 350px; left: 350px">
<p><strong>Node 1</strong></p>
<a href="http://www.subiranatfg.com/wordpress_d/wp-
content/uploads/2015/08/bombilla_off.jpg"></a>
</div>

<?php
}

mysql_close($linkB);
?>

<?php

```



```

$linkC = mysql_connect('subiranatfg.com', 'subirana_tfg', 'H6bwy34!');
        or die('No se pudo conectar: ' . mysql_error());

mysql_select_db('dades_farola') or die('No se pudo seleccionar la base de datos');

//NODE 3

$queryC = 'SELECT * FROM dades WHERE id=3';
$resultC = mysql_query($queryC) or die('Consulta fallida: ' . mysql_error());

$rsC = mysql_fetch_array($resultC, MYSQL_BOTH);
if ($rsC[3]==ON) {
?>
<div style="position: absolute; top: 350px; left: 950px">
<p><strong>Node 3</strong></p>
<a href="http://www.subiranatfg.com/wordpress_d/wp-content/uploads/2015/08/BOmbillaon.png"></a>
</div>

<?php
}
if ($rsC[3]==OFF) {
?>

<div style="position: absolute; top: 350px; left: 950px">
<p><strong>Node 3</strong></p>

```

```
<a href="http://www.subiranatfg.com/wordpress_d/wp-
content/uploads/2015/08/bombilla_off.jpg"></a>
</div>

<?php
}

mysql_close($linkC);
?>
</div>
_
```

- tmr0.c - Programa on inicialitzem timer 0

```
#include <inttypes.h>
#include <avr/io.h>
#include "tmr0.h"

void setup_tmr0(uint8_t ocr0a, uint16_t tmr0_pre){
    TCCR0A=0x02; //CTC mode: TOP=OCRA
    TIMSK0=0x02; //Bit 1 â OCIE0A: Timer/Counter0 Output
               Compare Match A Interrupt Enable
    OCR0A=ocr0a; //TOP value

    //tmr0_pre 1, default=8,64,256,1024
    //TMR0=prescaler*(ocr0a+1)*T_clk
    switch (tmr0_pre){
        case 1:TCCR0B=1;break;
        case 8:TCCR0B=2;break;
        case 64:TCCR0B=3;break;
        case 256:TCCR0B=4;break;
        case 1024:TCCR0B=5;break;
        default:TCCR0B=2;break;
    }
}
```

- pwm_tmr2.c - Programa que permet regular la potència del led a partir del PWM

```
#include <stdio.h>
#include <stdint.h>
#include <avr/io.h>
#include "pwm_tmr2.h"

void setup_pwm_tmr2(uint8_t pwm_out){
    // pwm output values:3,11
    switch (pwm_out){
        case 3:
            DDRD |= (1 << DDD3);
            TCCR2A=0b00100011; //OC2B(b5-b4): Fast PWM(b1-b0)
            break;
        case 11:
            DDRB |= (1 << DDB3);
            TCCR2A=0b10000011; //OC2A(b7-b6) : Fast PWM(b1-b0)
            break;
        default:
            DDRB |= (1 << DDB3);
            TCCR2A=0b10000011; //OC2A(b7-b6) : Fast PWM(b1-b0)
            break;
    }
    TCCR2B=0b00000001; //prescaling (b2-b1-b0, freq=clk/256
        = 62500, TOP=0xFF(b3)
}

void set_pwm_tmr2(uint8_t value){
    OCR2A=value; //OC2A:
    OCR2B=value; //OC2B:
}
```

