



Universitat de Vic
**Escola Politècnica
Superior**

Treball Final de Grau

Control de la temperatura i el llum mitjançant dispositius sense fils

Ramon Altarriba Roca

Grau en Enginyeria Mecatrònica

Director: Jaume Cabaní Masip

Vic, gener de 2015

Agraïments

Vull agrair al meu tutor de projecte, Jaume Cabaní Masip, l'ajuda, les facilitats i els recursos que m'ha proporcionat durant tota la realització d'aquest Treball Final de Grau.

També vull mostrar el meu agraïment a Moisès Plana i Sergi Sánchez per la seva inestimable ajuda, sense la qual l'elaboració d'aquest Treball Final de Grau no hauria estat possible.

I també vull donar les gràcies a la meva família, i molt especialment a la meva dona, per la paciència, els ànims, l'ajut i el suport incondicional que m'han donat durant tots aquests anys d'estudi.

Resum Treball Final de Grau

Grau en Enginyeria Mecatrònica

Títol: Control de la temperatura i el llum mitjançant dispositius sense fils.

Paraules clau: Comunicació sense fils, Xbee, Arduino, programació PHP, entorn Xampp, MySQL, Internet de les coses i sensors.

Autor: Ramon Altarriba Roca.

Direcció: Jaume Cabaní Masip.

Data: Gener de 2015.

Resum

El concepte Internet de les Coses va néixer fa molt poc, cap a l'any 2010, i es refereix a la connexió d'objectes a Internet, de manera que puguin interactuar en xarxa entre ells o amb les persones per transmetre o rebre informació, i realitzar, així, les accions programades en funció de les dades disponibles.

Des de llavors la gran majoria d'institucions, tant públiques com privades, s'han interessat en aquest concepte, ja que pot suposar un augment molt important de la qualitat de vida de les persones, un estalvi de recursos i beneficis econòmics.

D'aquí neix aquest Treball Final de Grau, ja que des de la Fundació Eduard Soler de Ripoll tenen el desig d'introduir-se en aquest camp.

L'objectiu principal d'aquest Treball Final de Grau és crear una xarxa sense fils que pugui controlar la temperatura i la il·luminació de l'edifici C de la Fundació Eduard Soler i que, a més, es pugui connectar a Internet.

Una de les conclusions a què he arribat gràcies a aquest treball és que la utilització d'eines de baix cost (Arduino, Xbee, MySQL, PHP...) permet introduir-se en l'apassionant món de l'Internet de les Coses i, a més, aconseguir molt bons resultats.

Summary final project

Degree in mechatronics engineering

Title: Control of temperature and light through wireless devices.

Keywords: Wireless Communication, XBee, Arduino, PHP, Xampp, MySQL, Internet of Things, sensors.

Author: Ramon Roca Altarriba.

Director: Jaume Cabaní Masip.

Date: January 2015.

Summary

The concept Internet of Things has recently appeared, more specifically by 2010. It makes reference to the connection between objects/ concepts within the Internet. The point is that they can interact with each other and with people by transmitting or receiving information as well as performing programmed actions based on available data.

From then onwards, the vast majority of public and private institutions have become highly interested in this concept, as it might lead to a huge increase in people's quality of life, bringing not only a resources saving, but also economic benefits.

This is the origin of this Final Project since Fundació Eduard Soler from Ripoll is willing to enter and explore this field.

The main objective of this Final Project is to create a wireless network which controls the temperature and lighting of Building C from the Fundació Eduard Soler while connecting it to the Internet.

One of the conclusions that allowed me to come to this point/ project was the fact that by using low cost tools (Arduino, XBee, MySQL, PHP, ...) we can enter the exciting world of the Internet of Things and more achieving very good results.

Índex

Índex.....	5
Índex taules	7
Índex figures	7
Índex equacions.....	7
Apartat 1 : Motivació, objectius i abast	8
1.1 Motivació.....	8
1.2 Objectius.....	8
1.3 Abast.....	8
Apartat 2 : Estudi i elecció dels diferents components de la xarxa	9
2.1 Protocol de comunicació	9
2.1.1 Introducció al protocol Zigbee	10
2.1.2 Característiques tècniques de Zigbee	10
2.1.3 Topologies de xarxa	11
2.2 Elecció del mòdul sense fils.....	12
2.2.1 Mòduls Xbee de Digi	12
2.3 Elecció del microcontrolador.....	14
2.4 Elecció del software	16
2.4.1 IDE d'Arduino	16
2.4.2 X-CTU	16
2.4.3 XAMPP.....	16
2.4.4 MySQL	17
2.4.5 PHP.....	17
2.5 Elecció dels components de la placa	18
2.5.1 Sensor de temperatura.....	18
2.5.2 Sensor de llum	20
2.5.3 Relé.....	21
2.5.4 Optoacobrador.....	21
2.5.5 Regulador de tensió	21
Apartat 3 : Proves prèvies	22
3.1 Arduino	22
3.2 Xbee	22
3.2.1 Material necessari	22
3.2.2 Revelacions i indicacions	23

3.2.3	Funcionament i configuració	24
3.2.4	Conversors A / D	25
3.2.5	Reset.....	26
3.3	Arduino + Xbee	27
3.3.1	Prova Mode transparent (AP).....	27
3.3.2	Introducció al Mode API	27
3.3.3	Prova Mode API	28
Apartat 4 : Implantació de la xarxa		30
4.1	Disseny de la xarxa	30
4.1.1	Alimentació dels mòduls.....	31
4.1.2	Valors de temperatura i nivell d'il·luminació objectiu	33
4.1.3	Condicions d'activació del ventilador.....	34
4.1.4	Definició de les zones	34
4.2	Configuració dels mòduls Xbee	35
4.3	Disseny de les plaques de circuit imprès	36
4.4	Codi Arduino	37
4.4.1	Llibreries.....	37
4.4.2	Variables	37
4.4.3	void setup()	37
4.4.4	Funcions.....	38
4.4.5	void loop ()	39
4.5	Disseny de la base de dades i el control remot	40
4.5.1	Base de dades	40
4.5.2	Control remot	41
4.6	Disseny de les caixes per a cada mòdul Xbee	44
4.7	Pressupost del material	46
4.7.1	Material per programar.....	46
4.7.2	Mòdul Coordinador.....	46
4.7.3	Mòdul Sensor	47
4.7.4	Mòdul Relé	47
4.7.5	Total	47
Apartat 5 : Conclusions		48
Apartat 6 : Possibles millores		49
Apartat 7 : Bibliografia i Webgrafia		50

Índex taules

Taula 2-1. Comparativa dels protocols de comunicació sense fils	9
Taula 2-2. Característiques del mòdul Xbee Sèrie2	13
Taula 4-1. Consum mòdul Xbee en mode normal i en mode <i>Sleep</i>	31
Taula 4-2. Configuració mòduls Xbee	35
Taula 4-3. Pressupost del material necessari per programar mòduls Xbee	46
Taula 4-4. Pressupost del material necessari per al mòdul coordinador	46
Taula 4-5. Pressupost del material necessari per al mòdul sensor	47
Taula 4-6. Pressupost del material necessari per al mòdul relé	47
Taula 4-7. Pressupost total del material necessari	47

Índex figures

Figura 2-1. Tipus de mòduls Xbee	12
Figura 2-2. Arduino Uno	14
Figura 2-3. Arduino Ethernet	14
Figura 2-4. Xbee Shield.....	15
Figura 2-5. Relació Temperatura-Tensió.....	18
Figura 2-6. Circuit amb amplificador operacional	19
Figura 2-7. Divisor de tensió LDR	20
Figura 3-1. Xbee Explorer USB i Breakout Board	23
Figura 3-2. Muntatge dels leds indicadors.....	23
Figura 3-3. Connexions mínimes per al mòdul Xbee.....	24
Figura 3-4. Tipus de trames API.....	28
Figura 3-5. Prova mode API.....	29
Figura 4-1. Calibratge fotoresistència LDR amb luxímetre	34
Figura 4-2. Disseny de la caixa per al mòdul amb sensors	44
Figura 4-3. Disseny de la caixa del coordinador.....	44
Figura 4-4. Disseny de la caixa del coordinador modificada	45
Figura 4-5. Caixes fabricades abans i després de tenyir-les	45

Índex equacions

Equació 2-1. Tensió sortida amplificador operacional	19
Equació 2-2. Tensió sortida amplificador operacional(2).....	19
Equació 2-3. Divisor de tensió.....	21
Equació 4-1. Consum del mòdul	33

Apartat 1 : Motivació, objectius i abast

1.1 Motivació

La motivació d'aquest Treball Final de Grau neix de la voluntat de la Fundació Eduard Soler de Ripoll de reduir el seu impacte ambiental i els costos econòmics relacionats amb el consum energètic, així com d'augmentar el benestar dels ocupants dels edificis propietat d'aquesta fundació.

1.2 Objectius

Aquest Treball Final de Grau té com a objectius principals dissenyar i implantar una xarxa sense fils de sensors i actuadors sense fils per tal de poder optimitzar el consum energètic, així com augmentar el benestar dels ocupants de l'edifici C de la Fundació Eduard Soler. També, però, en té d'altres:

- Crear una xarxa de baix cost econòmic i amb un manteniment mínim.
- Disposar d'un control remot de la xarxa.
- Utilitzar programari lliure per evitar haver de pagar llicències.
- Disposar d'un registre històric de les lectures dels sensors.

A més, aquesta xarxa de sensors sense fils ha de servir com a embrió per a una posterior automatització de l'edifici.

1.3 Abast

A causa de les dimensions de l'edifici, s'ha decidit delimitar la xarxa sense fils de sensors i actuadors al taller. Les limitacions de material i de pressupost han comportat que el projecte es concreti en:




- Un mòdul amb sensors de temperatura i llum i amb un relé que ha d'actuar sobre els ventiladors.
- Un mòdul amb un relé que ha d'actuar sobre la caldera.
- Un mòdul amb relé que ha d'actuar sobre els llums.
- Un mòdul coordinador de la xarxa.

Apartat 2 : Estudi i elecció dels diferents components de la xarxa

2.1 Protocol de comunicació

A causa de les característiques del treball, s'ha decidit utilitzar una xarxa de sensors sense fils, perquè permet cobrir grans distàncies amb un cost més reduït i una major facilitat d'instal·lació.

Per al pas següent, escollir quin protocol de comunicació cal emprar, s'ha realitzat una comparativa entre els protocols sense fils més coneguts: WiFi, Bluetooth i ZigBee.

			
	WiFi	Bluetooth	ZigBee
Velocitat	<50 Mbps	1 Mb	<250 Kbps
Nombre nodes	32	8	255 / 65535
Duració bateria	Hores	Dies	Anys
Consum transmissió	400 mA	40 mA	30 mA
Consum repòs	20 mA	0.2 mA	<3 µA
Cost econòmic	Alt	Mig	Baix
Configuració	Complexa	Complexa	Simple
Aplicacions	Internet en edificis	Informàtica i mòbils	Domòtica

Taula 2-1. Comparativa dels protocols de comunicació sense fils

S'ha decidit usar el protocol ZigBee perquè, tenint en compte les necessitats del projecte, és el més idoni, ja que:

- Permet una gran quantitat de nodes.
- Té un consum molt baix i, per tant, les bateries tenen una vida llarga.
- Té un cost econòmic reduït.
- No és necessària una gran velocitat de transmissió.

2.1.1 Introducció al protocol Zigbee

En el desenvolupament d'aquest Treball Final de Grau s'usa Zigbee com a protocol de comunicacions per a la implementació dels sensors sense fils que es veurà més endavant.

Des del punt de vista del hardware, es pot dir que Zigbee ha estat acuradament optimitzat per al baix cost a gran escala, perquè té poques parts analògiques i utilitza circuits digitals sempre que és possible.

Des del punt de vista tècnic, és un estàndard que defineix un conjunt de protocols de comunicació de baixa velocitat de dades. I és que està basat en dispositius sense fils que operen a 868 MHz, 915 MHz i 2.4 GHz.

La tècnica de modulació és Direct Sequence Spread Spectrum (DSSS), que té un màxim rang de velocitat d'uns 250 Kbit / s.

Per tant, en resum es pot catalogar Zigbee com un protocol:

- Basat en l'estàndard 802.15.4.
- Amb aplicacions WPAN (Wireless Personal Area Network).
- De baix cost.
- Amb baix consum.
- Amb una instal·lació fàcil i barata.
- Amb xarxes flexibles i extensibles.

2.1.2 Característiques tècniques de Zigbee

Zigbee defineix els següents tipus de dispositius.

Zigbee Coordinator: És el més complet. És l'administrador de la xarxa i n'hi ha un per xarxa.

Zigbee Router: S'encarrega d'interconnectar els dispositius mitjançant tècniques d'encaminament i adreçament.

Zigbee End Device: És un element passiu de la xarxa, ja que no transmet informació de manera autònoma; simplement disposa de la funcionalitat mínima per ser capaç de respondre a peticions de dispositius superiors (coordinador o router). Gràcies a això i a la poca memòria que necessita, és molt econòmic. A més, la seva bateria pot durar anys, ja que es passa adormit la major part del temps.

2.1.3 Topologies de xarxa

Zigbee permet tres topologies de xarxa:

Punt a Punt: La forma més senzilla de la xarxa és amb dos nodes, un dels quals ha de ser un coordinador.

Estrella: En aquest cas el coordinador és el centre de la xarxa i és el que es connecta en cercle amb els altres dispositius (End Devices). Per tant, tots els missatges han de passar pel coordinador.

Mesh: La configuració compta amb nodes router i amb un de coordinador. Es tracta d'una topologia no jeràrquica, ja que qualsevol dispositiu pot interactuar amb qualsevol altre.

Clúster Tree: En aquest disseny els routers formen una columna vertebral amb els dispositius finals, que estan agrupats entorn dels routers.

2.2 Elecció del mòdul sense fils

Una vegada establert el protocol a utilitzar, el següent pas ha estat triar el mòdul Zigbee més adequat, perquè hi ha una àmplia varietat d'aquest tipus de dispositius al mercat.

Un cop realitzada l'anàlisi sota els criteris de cost, senzillesa de la programació, extensió en el mercat i informació disponibles, s'ha decidit utilitzar els mòduls Xbee de Digi.

2.2.1 Mòduls Xbee de Digi

Per portar a terme la propera acció, decidir quins dels mòduls Xbee i quina antena cal emprar, s'han analitzat els mòduls:

Xbee Sèrie1.

Xbee Sèrie2.

Xbee Pro.

I també les següents antenes:

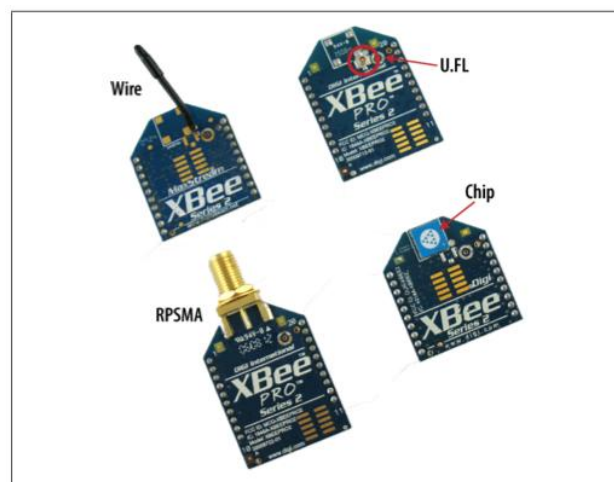


Figura 2-1. Tipus de mòduls Xbee

Una vegada fet l'estudi, s'ha decidit que el mòdul i l'antena que s'ajusten més a les necessitats del projecte són el mòdul Xbee Sèrie2 amb l'antena Wire, ja que els Sèrie1 només permeten la configuració de xarxa Punt a Punt i, en aquest cas, es necessita un altre tipus de configuració de xarxa.

En aquest punt cal puntualitzar que amb els Xbee Pro, tot i que permeten crear tots els tipus de xarxes i disposen d'un rang d'abast molt gran (1600 m), el consum s'incrementa significativament (295 mA). I en aquest projecte cal fer prevaldre un menor consum a un major abast.

Les característiques del mòdul escollit són les següents:

Dispositiu Xbee:	Xbee 2mW Wire antena, Serie2.
Rang:	120 m outdoor – 40 m indoor.
Rang alimentació:	2.1 – 3.6 V.
Rang entrades analògiques:	0 – 1.2 V.
Consum transmissió i recepció dades	40 mA.
Consum mode Sleep:	1 μ A.
Freqüència:	2.4 GHz.
Protocol:	Zigbee Mesh.
Potència Tx:	2mW.
Dades:	250 kbps.
Antena:	Cable.

Taula 2-2. Característiques del mòdul Xbee Sèrie2

2.3 Elecció del microcontrolador

Una vegada escollits el protocol de comunicació i els mòduls a emprar, ha calgut decidir quin microcontrolador s'utilitza per ser el cervell de la xarxa i per servir de pont de connexió entre el coordinador i el PC.

Els requisits que s'han tingut en compte per fer l'elecció han estat: compatibilitat amb els mòduls Xbee, programari lliure, baix cost, capacitat de disposar d'un entorn amigable i facilitat de programació, opció de múltiples entrades i sortides analògiques i digitals, i possibilitat d'aplicacions remotes.

S'han escollit dues plaques de la família Arduino:

Arduino Uno: Per fer les proves prèvies al muntatge final.



Figura 2-2. Arduino Uno

Arduino Ethernet: Per realitzar el muntatge final, ja que permet aplicacions remotes. En aquest cas s'ha tingut en compte la necessitat d'utilitzar un programador amb connexió USB, ja que la placa Arduino Ethernet no disposa de connexió USB per carregar els programes del PC al microcontrolador.



Figura 2-3. Arduino Ethernet

Les dues plaques de la família Arduino escollides utilitzen el microcontrolador ATMEL AVR Atmega 328, amb diverses entrades i sortides, tant digitals com analògiques, i amb connexió USB. A més, disposen d'un entorn de desenvolupament propi i el llenguatge de programació del microcontrolador està basat en C / C ++ .

En aquest punt cal esmentar que, a l'utilitzar plaques Arduino, se soluciona el problema de l'acoblament dels mòduls Xbee al PC per poder llegir i processar les dades, ja que les plaques Arduino permeten l'acoblament de la Xbee Shield, que possibilita la comunicació dels mòduls Xbee amb el microcontrolador d'Arduino.

La placa que s'ha triat és la Xbee Shield, perquè s'ha prioritzat el cost econòmic per sobre de la comoditat, ja que aquesta placa comporta que, cada cop que es vol carregar un codi al microcontrolador, s'ha de separar físicament la Xbee Shield de la placa Arduino, i un cop acabada la càrrega s'ha de tornar a connectar.



Figura 2-4. Xbee Shield

2.4 Elecció del software

2.4.1 IDE d'Arduino

L'elecció del software a emprar per programar el microcontrolador ha estat molt fàcil, ja que, a l'usar plaques de la família Arduino, estava molt clar que s'havia d'utilitzar l'entorn de desenvolupament gratuït IDE d'Arduino, que permet realitzar programes perquè siguin carregats al microcontrolador via USB. En aquest cas s'ha optat per la versió 1.5.8.

2.4.2 X-CTU

Si bé és cert que és possible utilitzar el Hyperterminal de Windows per configurar un mòdul Xbee, s'ha optat per usar el programa anomenat X-CTU, que permet realitzar aquestes operacions de manera més natural, fàcil i ràpida. De fet, X-CTU és un programa desenvolupat per Digi que permet configurar els seus mòduls.

En l'elaboració d'aquest treball de grau s'han utilitzat dues versions del programa esmentat:

Versió 5.2.7.5: Per a la configuració dels mòduls durant les proves prèvies al muntatge final.

Versió: 6.1.0: Per a la configuració final dels mòduls.

2.4.3 XAMPP

Un dels objectius del Treball Final de Grau és disposar d'un control remot de la xarxa de sensors, i per això ha estat necessari buscar una plataforma que permeti fer les proves necessàries i la posterior implantació.

S'ha decidit utilitzar el servidor independent de plataforma Xampp, ja que compleix amb l'objectiu comentat anteriorment i, a més, és un programari d'accés lliure. Un altre avantatge molt important que ha decantat la balança cap aquest servidor independent de plataforma és que disposa de la base de dades MySQL i intèrpret pel llenguatge PHP.

2.4.4 MySQL

Per tal de poder complir amb el principal objectiu d'aquest treball, és a dir, optimitzar al màxim el consum energètic de l'edifici i que el manteniment de la xarxa sigui mínim, ha estat necessari crear un calendari on, en funció del dia i l'hora, es puguin definir els objectius de temperatura i llum que es volen aconseguir. Per exemple, no té cap sentit que s'activi el sistema de calefacció si el dia en qüestió és festiu, encara que la temperatura llegida per la xarxa de sensors així ho indiqui.

Per tant, per no haver d'estar cada dia configurant el codi del microcontrolador, s'ha decidit crear una base de dades amb el sistema de gestió MySQL.

2.4.5 PHP

Per complir amb els objectius de control remot, optimització del consum energètic i manteniment mínim de la xarxa, ha estat necessari poder comunicar la base de dades amb el microcontrolador. És a dir, ha calgut saber el dia i l'hora en cada moment per conèixer quin valor de temperatura i llum objectius defineix la base de dades. Un cop fet això s'han hagut de llegir aquests valors i integrar-los dins del codi del microcontrolador. I, per últim, enviar les lectures de la xarxa de sensors a la base de dades per crear un registre històric d'aquestes lectures. Per realitzar aquestes tasques s'ha utilitzat el llenguatge PHP, que permet integrar les consultes SQL amb el codi necessari per connectar la base de dades amb el del micorcontrolador.

2.5 Elecció dels components de la placa

2.5.1 Sensor de temperatura

Tenint en compte les necessitats del projecte, s'ha escollit el sensor de temperatura TMP20 per diferents motius:

- El rang d'alimentació, ja que es pot alimentar amb el mateix rang que els mòduls Xbee.
- El poc espai que ocupa.
- El baix consum.

Tot i que el punt dèbil d'aquest sensor és la sensibilitat, s'ha cregut que per a la nostra aplicació aquesta no era crítica.

Descripció

El TMP20 és un sensor que opera entre $-55\text{ }^{\circ}\text{C}$ a $130\text{ }^{\circ}\text{C}$, amb una precisió de $2,5\text{ }^{\circ}\text{C}$ dins d'aquest rang. La tensió d'alimentació oscil·la entre els 1.8 V i els 5.5 V , i la funció de transferència lineal té un pendent de $-11.77\text{ mV / }^{\circ}\text{C}$.

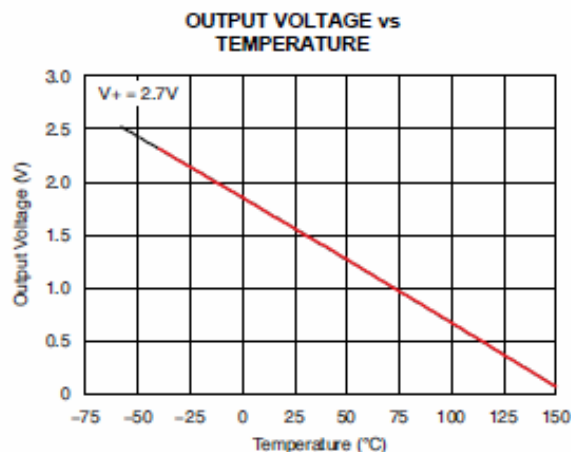


Figura 2-5. Relació Temperatura-Tensió

Muntatge

Per aconseguir lectures vàlides amb el sensor, el primer que s'ha fet ha estat agafar el nostre fons d'escala. S'ha pres -15°C com la temperatura mínima de què es podrà agafar la lectura. Si s'observa la gràfica que relaciona temperatura amb voltatge, es veu que el -15°C correspon a 2 V . Per tant, tenint

en compte que el rang de les entrades analògiques dels mòduls Xbee va de 0 a 1.2 V i el de tensió del sensor va de 0 a 2V, s'ha necessitat utilitzar un factor de conversió per adaptar aquests valors:

$$1.2 / 2 = 0.6$$

Per tant, s'agafa aquesta relació:

$$V_{out} = V_{sensor} \times \frac{60}{100}$$

Per aconseguir-la s'ha utilitzat un amplificador operacional de baix consum, concretament el TLV2401, amb la topologia d'amplificador diferencial:

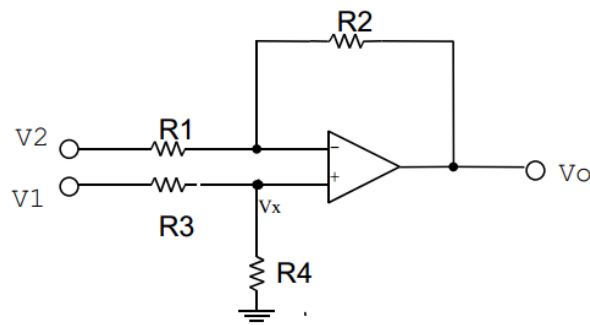


Figura 2-6. Circuit amb amplificador operacional

Usant aquesta topologia la tensió de sortida és:

$$V_o = \left(1 + \frac{R_2}{R_1}\right) \left(\frac{\frac{R_4}{R_3}}{\left(1 + \frac{R_2}{R_3}\right)}\right) V_1 - \frac{R_2}{R_1} V_2$$

Equació 2-1. Tensió sortida amplificador operacional

Però s'ha decidit que:

$$R_4 = R_2$$

$$R_3 = R_1$$

Així, la tensió de sortida és:

$$V_o = \frac{R_2}{R_1} (V_1 - V_2)$$

Equació 2-2. Tensió sortida amplificador operacional(2)

Cal tenir en compte que $V_2 = 0$.

Per tant¹:

- $R_2 = 60 \Omega$
- $R_1 = 100 \Omega$
- $R_3 = 100 \Omega$
- $R_4 = 60 \Omega$

2.5.2 Sensor de llum

Com a sensor de llum s'ha optat per una fotoresistència LDR VT900, ja que presenta les següents característiques que el fan adient al projecte:

- Té un temps de resposta curt.
- Varia el seu valor d'acord amb la intensitat de la llum.
- És analògic i de baix consum.

Muntatge

Per poder integrar la LDR al circuit s'ha decidit utilitzar un divisor de tensió en configuració *pull-down*. Això s'ha fet així perquè a més llum, més voltatge.

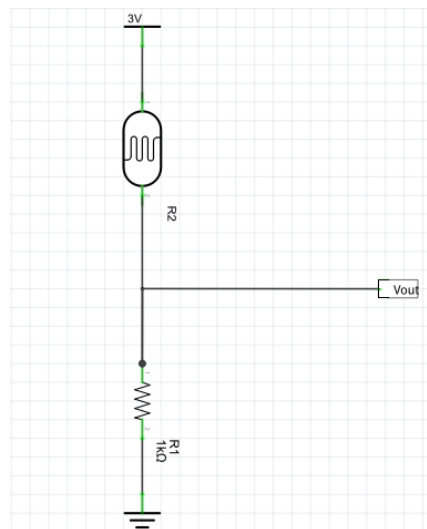


Figura 2-7. Divisor de tensió LDR

S'ha optat per agafar el valor de 1K com a fons d'escala.

Si s'aplica la fórmula del divisor de tensió:

¹ Es pot consultar l'esquema elèctric a l'Annex 1.

$$V_{out} = \frac{R_2}{R_1 + R_2} V_{in}$$

Equació 2-3. Divisor de tensió

La tensió màxima serà d'1.5 V (màxim nivell d'il·luminació). Per tant, s'ha seguit el mateix mètode que un detector de temperatura per adaptar-ho al rang del mòdul Xbee².

2.5.3 Relé

Per actuar sobre els llums, la caldera i el ventilador cal utilitzar un relé. L'escollit ha estat un d'estàtic, perquè respecte als electromecànics presenta una vida molt més llarga, ja que no disposa de cap peça mòbil que es desgasti ni es poden formar arcs elèctrics en els contactes.

S'ha triat el relé 34.81-8240 de Finder.

2.5.4 Optoacobrador

A causa de la sensibilitat dels mòduls Xbee i per evitar possibles danys en la resta del circuit, s'ha decidit emprar un optoacobrador, concretament el 6N139. Això ha permès aïllar elèctricament la resta del circuit, ja que la part de control del circuit podria sofrir algun dany al voler controlar elements que funcionen a 220 V.

2.5.5 Regulador de tensió

Un dels problemes que s'ha hagut de resoldre ha estat que, a l'utilitzar relés per activar elements que funcionen a 220 V, es necessitava alimentar el relé a 5 V, mentre que la resta del circuit (sensors, mòdul Xbee,...) funciona a 3 V. A més, és molt més fàcil trobar una font d'alimentació a 5 V (USB) que a 3 V.

Per tant, per quan sigui necessari utilitzar el relé, s'ha decidit alimentar tot el circuit a 5 V conjuntament amb el regulador de tensió. Per tal de poder portar a terme aquesta tasca s'ha escollit el regulador de tensió LM317AEMP³.

² Es pot consultar l'esquema elèctric a l'Annex 1.

³ Com que el rang de V_{out} és molt ampli, es pot consultar el càlcul de les R per aconseguir els 3 V a la sortida del regulador a l'Annex 1.

Apartat 3 : Proves prèvies

Per afrontar i donar resposta als objectius marcats, cal disposar d'uns coneixements en la utilització dels diferents entorns que s'han escollit per realitzar aquest treball de grau. Per això s'ha decidit portar a terme un seguit d'estudis i proves prèvies.

3.1 Arduino

Tot i disposar de coneixements en la programació en C i que el llenguatge que utilitza IDE Arduino per programar el microcontrolador està basat en C, entre d'altres, s'ha cregut necessari realitzar alguna prova per familiaritzar-nos completament amb aquest món.

Una de les proves que s'ha fet ha estat:

- Encendre o apagar un determinat nombre de leds en funció de la variació de la temperatura ambient respecte a una temperatura de referència prèviament definida⁴.

3.2 Xbee

3.2.1 Material necessari

A part de disposar dels mòduls Xbee Sèrie2 i del programa de configuració X-CTU instal·lat, han calgut altres elements per configurar i treballar amb els mòduls X-CTU. Així, per configurar els mòduls Xbee mitjançant X-CTU, s'ha hagut d'adquirir un Xbee Explorer USB. A més, a causa de la incompatibilitat dels mòduls Xbee amb les protoboards, ha estat necessari obtenir Breakout Board.

Amb tot el material a punt, el pas següent ha estat muntar i soldar les connexions per poder-hi col·locar el mòdul Xbee i així poder-lo ubicar a la protoboard⁵.

⁴ Es pot consultar el codi complet i comentat a l'Annex 2.

⁵ Es pot consultar el procediment i el material necessari a l'Annex 2.

3.2.3 Funcionament i configuració

Com ja s'ha explicat en apartats anteriors, els mòduls escollits són els Xbee Sèrie2.

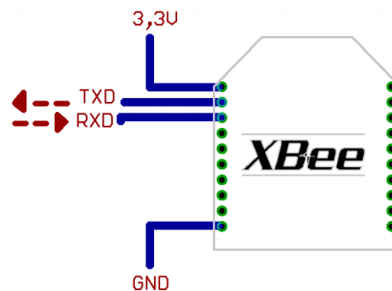


Figura 3-3. Connexions mínimes per al mòdul Xbee

El mòdul requereix una alimentació de 2.8 a 3.4 V, la connexió a terra i les línies de transmissió de dades per mitjà de l'UART (TXD i RXD) per comunicar-se amb un microcontrolador.

Els modes d'operació del mòdul Xbee són els següents:

Mode recepció / transmissió.

Es troba en un d'aquests modes quan arriba al mòdul algun paquet RF per l'antena (mode recepció) o quan es demana informació serial al buffer del pin 3, que després serà transmesa (mode transmissió).

Mode de baix consum.

Fa possible que el mòdul es trobi en un estat de baix consum quan no està en ús. La configuració del mode *Sleep* es realitza mitjançant ordres. Per defecte, el mode *Sleep* està deshabilitat (mòdul en estat de repòs / recepció).

Mitjançant el pin d'hivernació *Sleep_RQ* (pin 9) en estat alt, el mòdul acaba qualsevol transmissió, recepció o procediment d'associació i entra en mode repòs i després en *Sleep*. Hi ha un altre mode *Sleep* a part del controlat pel pin, el mode *Sleep* cíclic, que consisteix a despertar el mòdul cada cert període perquè comprovi si hi ha dades per enviar.

Mode de comandament.

Amb aquest és possible configurar paràmetres del mòdul mitjançant comandes AT. Per aconseguir-ho s'ha d'emprar un programari com el X-CTU de Digi.

Mode transparent (AT)

En el mode transparent tot el que ingressa pel pin 3 (DIN) es guarda en un buffer d'entrada i es transmet, i el que ingressa com a paquet RF es guarda en una memòria intermèdia de sortida i s'envia pel pin 2 (Dout) immediatament o després del pas d'un temps (paràmetre RO). Aquesta manera es destina a la comunicació punt a punt, on no cal cap tipus de control.

Mode d'operació API

Quan es troba en aquest estat, el mòdul empaqueta en trames tota la informació que entra i surt. Aquestes defineixen operacions i esdeveniments del mòdul. Entre les opcions que permet l'API hi ha:

- Transmetre informació a múltiples destinataris, sense entrar a la manera d'ordres.
- Rebre l'estat d'èxit / fallada de cada paquet RF transmès.
- Identificar l'adreça d'origen de cada paquet rebut.

Mode IDLE

Es diu que si el mòdul no està en cap dels estats esmentats anteriorment es troba en estat IDLE.

3.2.4 Conversors A / D

Per mostrejar senyals els mòduls Xbee estan equipats amb convertidors A / D de 10 bits.

Una altra característica important és el temps de mostreig, que es pot ajustar fent servir el comandament IR i les unitats en ms. Per exemple, per demanar un temps de mostreig de 10 ms programarem ATIR = 0x0A. Un altre tret destacat és el nombre de mostres que es volen prendre abans d'enviar dades. Això es pot canviar amb l'ordre IT, tenint en compte que es poden emmagatzemar 93 bytes com a màxim i que cada mostra ocupa 2 bytes. Així doncs, es poden

arribar a prendre 43 mostres abans d'enviar. El temps màxim de mostreig és de 1 mostra / ms.

3.2.5 Reset

Un dels principals problemes a què s'ha hagut de fer front durant la realització d'aquest Treball Final de Grau ha estat realitzar un reset als mòduls Xbee, ja que es poden quedar bloquejats.

Després de moltes hores de proves, s'ha trobat la manera de restablir els valors de fàbrica dels mòduls per tal de poder-los tornar a utilitzar⁶.

⁶ Es pot consultar el mètode que cal seguir a l'Annex 2.

3.3 Arduino + Xbee

Amb l'objectiu de conèixer els diferents modes dels mòduls Xbee i la integració d'aquests amb les plaques Arduino, s'han portat a terme diverses proves.

3.3.1 Prova Mode transparent (AP)

Com ja s'ha explicat, el mode transparent serveix per les comunicacions punt a punt.

Per tal de conèixer millor el mode AP i *Sleep* s'ha fet les següents proves:

- Encendre i apagar un led remotament.
- Fer una lectura remota de la temperatura.
- Realitzar una lectura de la temperatura remotament cada 10 minuts, conèixer la lectura del sensor, transformar-ho a volts i graus i mostrar-ho tot per Serial Port⁷.

3.3.2 Introducció al Mode API

Aquest mode ha permès treballar aprofitant al màxim el treball de xarxes amb els dispositius Xbee Sèrie2.

Els protocols de xarxes utilitzen, generalment en la capa d'enllaç de dades, l'encapsulament de la informació en trames. Així, cada protocol té estandarditzades les diferents parts d'aquestes trames. Per exemple, el protocol TCP / IP divideix les seves trames en un preàmbul de 7 bytes, un delimitador de trama d'1 byte, les adreces físiques (MAC) de destinació i origen de 6 bytes cada una, la longitud de la trama de 2 bytes, la informació, que pot estar entre 46 i 1.500 bytes, i finalment l'ús de 4 bytes per l'algorisme de verificació de la recepció completa de la informació.

En el protocol Zigbee les trames API són diverses, ja que cadascuna té una funció especial d'acord amb l'aplicació que es vol realitzar.

⁷ Es pot consultar la configuració dels mòduls i el codi Arduino a l'Annex 2.

API Frame Names	API ID
AT Command	0x08
AT Command - Queue Parameter Value	0x09
ZigBee Transmit Request	0x10
Explicit Addressing ZigBee Command Frame	0x11
Remote Command Request	0x17
Create Source Route	0x21
AT Command Response	0x88
Modem Status	0x8A
ZigBee Transmit Status	0x8B
ZigBee Receive Packet (AO=0)	0x90
ZigBee Explicit Rx Indicator (AO=1)	0x91
ZigBee IO Data Sample Rx Indicator	0x92
XBee Sensor Read Indicator (AO=0)	0x94
Node Identification Indicator (AO=0)	0x95
Remote Command Response	0x97
Over-the-Air Firmware Update Status	0xA0
Route Record Indicator	0xA1

Figura 3-4. Tipus de trames API

Per treballar en xarxes amb mode API, cal que com a mínim el coordinador sigui API.

Per introduir-nos en aquest camp, s'han realitzat les següents pràctiques⁸:

- Anàlisi d'una trama API rebuda.
- Muntatge d'una petita xarxa sense fils en la que hi haurà un coordinador API i un End Device, on aquest tindrà un temps de mostreig establert i enviarà l'estat de les seves entrades i sortides al coordinador.

3.3.3 Prova Mode API

Aquest és el mode que s'ha escollit, ja que és el que permet realitzar xarxes Mesh. S'han portat a terme dues pràctiques per tal de conèixer el Mode API dels mòduls Xbee⁹:

⁸ Es poden consultar aquestes pràctiques a l'Annex 2.

⁹ Es pot consultar la configuració dels mòduls i el codi Arduino a l'Annex 2.

- Crear una xarxa sense fils que permeti saber la lectura d'un sensor simulat per un potenciòmetre connectat a un mòdul Xbee, que s'enviïn les lectures al mòdul Xbee que s'ha definit com a coordinador, que l'Arduino tingui accés a aquestes dades i que, en funció dels valors obtinguts, s'envii una ordre d'encendre o apagar un led al mòdul coordinador.
- Afegir una pantalla LCD per visualitzar les lectures i, en funció de la lectura, fer que el coordinador enviï l'ordre d'activació d'un led col·locat en el mòdul que realitza les lectures del potenciòmetre. A més, s'ha col·locat un divisor de tensió en el potenciòmetre per limitar el voltatge al rang de les entrades del mòdul Xbee.

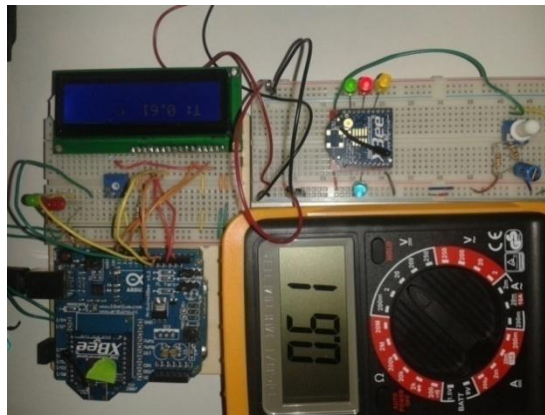


Figura 3-5. Prova mode API

Apartat 4 : Implantació de la xarxa

Un cop decidit el hardware, el software i les configuracions a utilitzar, els següents passos han estat:

- Disseny de la xarxa.
- Configuració dels mòduls Xbee.
- Disseny de les plaques per cada mòdul Xbee.
- Codi Arduino.
- Disseny de les base de dades i el control remot.
- Disseny de les caixes per cada mòdul Xbee.

4.1 Disseny de la xarxa

S'ha decidit dissenyar una xarxa sense fils que sigui capaç de:

- Saber la temperatura i el nivell d' il·luminació actual del taller.
- Engregar i parar la caldera, els llums i el ventilador.
- Accedir a una base de dades que, en funció del dia i l'hora, tingui una temperatura i un nivell d'il·luminació objectiu.
- Engregar o apagar la caldera i/o els llums si la temperatura i/o el nivell d'il·luminació actual estan per sota dels respectius valors objectiu definits en la base de dades
- Engregar o parar el ventilador si la temperatura actual i l'objectiu estan dins d'uns determinats valors.
- Disposar d'un mode manual que remotament pugui engregar o parar els llums, la caldera o el ventilador.
- Permetre que es pugui ampliar fàcilment a altres zones de l'edifici.

4.1.1 Alimentació dels mòduls

L'alimentació de la xarxa no s'ha considerat problemàtica, ja que no s'utilitzen bateries perquè, com que cada mòdul actua sobre un aparell elèctric (ventilador, caldera o llums), s'ha considerat que sempre hi haurà un punt fix d'alimentació de la xarxa elèctrica de l'edifici pròxim al mòdul.

Malgrat tot, s'ha decidit fer un petit estudi del consum dels mòduls i de la vida de les bateries per si més endavant es vol desplegar la xarxa. En aquest cas, s'haurien de tenir en compte les següents premisses:

- El mòdul coordinador sempre estarà alimentat per la xarxa elèctrica.
- Els mòduls amb relé (els que activen o desactivem llums, caldera...) sempre estaran alimentats per la xarxa elèctrica.
- Els mòduls amb els sensors són els que aniran alimentats amb bateries i estaran repartits per diferents punts de l'edifici.

Per tant, els únics mòduls alimentats amb bateries seran els que tinguin sensors. Cal tenir en compte que en aquests mòduls, per tal d'augmentar al màxim l'autonomia de les bateries, caldrà eliminar tots els components que no serveixin, tals com el relé i l'optoacobrador. I és que aquests mòduls no tindran la funció d'actuar sobre cap element i, per tant, al no haver d'anar alimentats a 5 V, el regulador de tensió i els leds indicadors també s'eliminaran.

S'ha realitzat el següent estudi de consum del mòdul amb els sensors:

ACCIÓ/COMPONENT	CONSUM MODE NORMAL	CONSUM MODE SLEEP
Transmissió dades	40000 μ A	1 μ A
Recepció dades	40000 μ A	1 μ A
3 Amplificadors	2.64 μ A	2.64 μ A
Sensor Temperatura	4 μ A	4 μ A
Sensor Llum	3000 μ A	500 μ A
TOTAL	83006.6 μA	508.6 μA

Taula 4-1. Consum mòdul Xbee en mode normal i en mode *Sleep*

Un cop conegut el consum, és on entra en joc la importància del mode *Sleep*.

Per això s'ha fet la següent suposició:

A causa de les característiques de la nostra xarxa i de les dades que es volen conèixer, s'ha decidit que amb una lectura cada 20 minuts és suficient.

La configuració necessària del mòdul Xbee, doncs, seria la següent:

- A la carpeta *Sleep Modes*, seleccionar l'opció *SM* i posar-hi el valor 4, que activa el *Cyclic Sleep*.
- A la carpeta *Sleep Modes*, seleccionar l'opció *ST*, en què s'especifica el temps d'inactivitat abans d'entrar en el mode *Sleep*, el rang de valors acceptats (hexadecimal), que és de $0X1 - 0XFFFE \times 1mS$, i col·locar-hi el valor 1388 (5 segons).
- A la carpeta *Sleep Modes*, seleccionar l'opció *SP*, en què s'especifica el temps que el mòdul romandrà dormint i el rang de valors acceptats (hexadecimal), que és de $0X20 - 0XAF0 \times 10mS$. Com es pot observar, el valor màxim permès és en decimal $2800 \times 10mS = 28000$, és a dir, 28 segons. Llavors caldria col·locar-hi el valor màxim $0X3E8$, que en decimal és 1000, $1000 \times 10 mS = 10000 mS$.
- A la carpeta *Sleep Modes*, seleccionar l'opció *SN*, en què s'especifiquen els cicles en què el mòdul ha d'estar dormint. Aquesta opció serveix per ampliar el temps màxim que el mòdul pot estar en mode *Sleep*. El rang de valors acceptats (hexadecimal) és de $0X1 - 0XFFFF$. Cal col·locar-hi, doncs, $0X78$, que en decimal és 120.
- A la carpeta *Sleep Modes*, seleccionar l'opció *SO*, col·locar el valor $0X2$ i el mòdul estarà en mode actiu durant el temps especificat en *ST*. Escollir després l'opció $0X04$, que permet aconseguir un temps major de *Sleep*, ja que en aquest cas el temps total serà $SP \times SN$, és a dir, $10000 \times 120 = 1200000 mS$, que són 1200 segons (20 minuts).

A part de la configuració del *Sleep Modes*, cal tenir en compte un altre paràmetre: el temps de mostreig (IR) dins de la carpeta "I/O Sampling". S'ha decidit col·locar-hi 1388 (5 segons).

Amb totes aquestes dades s'ha pogut realitzar un càlcul aproximat del consum.

$$\text{Consum} = \frac{T_{cn} \cdot I_{cn} + T_{cd} \cdot I_{cd}}{T_{cn} + T_{cd}} \quad \text{Equació 4-1. Consum del mòdul}$$

T_{cn} = Temps consum normal → 5 segons

I_{cn} = Intensitat consum normal → 83 mA

T_{cd} = Temps consum adormit → 1200 segons

I_{cd} = Intensitat consum adormit → 0.5 mA

$$\text{Consum} = \frac{5 \cdot 83 + 1200 \cdot 0.5}{5 + 1200} = 0.84 \text{ mA}$$

Si s'utilitza una pila de 1000mAh, la durada d'aquesta serà de 1.190,5 hores, és a dir, uns 50 dies.

4.1.2 Valors de temperatura i nivell d'il·luminació objectiu

Valor de temperatura objectiu

Per definir el valor referència per la temperatura s'ha tingut en compte el RD 486, en el qual s'especifiquen els valors de referència i diferència entre la temporada de tardor-hivern i la de primavera-estiu per als dies i torns laborals:

- Temperatura referència tardor-hivern: 20°C
- Temperatura referència primavera-estiu: 23°C

Per als dissabtes, diumenges i dies festius s'ha definit el següent valor de temperatura objectiu¹⁰:

- Temperatura referència tardor-hivern: 12°C
- Temperatura referència primavera-estiu: 12°C

Valor de nivell d'il·luminació objectiu

Per definir el valor referència per al nivell d'il·luminació també s'ha tingut present el RD 486, en què s'especifiquen els nivells d'il·luminació per a diferents activitats que es poden desenvolupar en l'entorn laboral. A causa de les característiques de la instal·lació on s'ha d'implementar la xarxa, el RD 486 determina un nivell d'il·luminació de 200 lux.

¹⁰ S'ha decidit aquesta temperatura objectiu per evitar possibles problemes de condensació i/o glaçades.

Per tant, per als dies i torns laborables s'ha definit el següent valor objectiu mínim:

- 200 lux.

I per als dies o torns festius s'ha definit el següent valor :

- 0 lux.

Per poder saber a quin voltatge corresponen aquests 200 lux, s'ha fet una prova amb un luxímetre calibrat:



Figura 4-1. Calibratge fotoresistència LDR amb luxímetre

S'ha decidit que 200 lux equivalen a 0.84 V del nostre sensor, però degut que la il·luminació del taller compleix amb els valors marcats en la normativa vigent s'ha establert 0.90 V com a valor objectiu.

4.1.3 Condicions d'activació del ventilador

Les condicions que s'han definit per decidir si cal o no activar el ventilador són:

- Remotament, utilitzant el mode manual.
- Si la temperatura actual és superior a 30 °C i la temperatura objectiu és superior a 12 s'activa el ventilador. Si no, es para.

4.1.4 Definició de les zones

El disseny de la xarxa s'ha centrat en el taller. Tot i així, s'ha decidit crear un camp denominat zona per si en un futur es decideix ampliar la xarxa a altres àrees, com per exemple a les oficines. Perquè aquesta ampliació sigui més fàcil, en el nostre cas només hi ha una zona, que és la 1, que correspon al taller, i s'ha decidit que per defecte el codi carregat a l'Arduino enviï aquesta zona.

4.2 Configuració dels mòduls Xbee

Per complir amb el disseny explicat, s'han utilitzat i configurat els quatre mòduls disponibles de la següent manera¹¹:

Adreça Mac	Configuració	Funció	NI
0013A200 40B3CEB1	Coordinador API	Coordinador	COORDINADOR
0013A200 40A59D8B	End Device API	Sensors + Ventilador	NI0101
0013A200 40A59DAE	End Device API	Caldera	RI0101
0013A200 40BF9972	End Device API	Llums	RI0102

Taula 4-2. Configuració mòduls Xbee

Per poder configurar els mòduls s'ha utilitzat el software X-CTU.

Com ja s'ha explicat, l'autonomia dels mòduls no es considera crítica, ja que poden estar endollats a la xarxa elèctrica de l'edifici mitjançant una font d'alimentació de 5 V. Per aquesta raó no s'ha programat el mode *Sleep*, ja que sempre estan desperts. Només s'ha configurat el mode *Sleep* del mòdul NI0101, de manera que cada 30 segons es desperti i enviï les lectures dels sensors. Això s'ha fet, bàsicament, per mostrar el funcionament d'aquesta funció.

¹¹ Es pot consultar la configuració completa a l'Annex 3.

4.3 Disseny de les plaques de circuit imprès

Tot i que segons el disseny de la xarxa que s'ha definit es necessiten dues plaques diferents:

- Placa per al mòdul amb els dos sensors i relè.
- Placa per als dos mòduls sense sensors i relè.

Al final s'ha optat per crear-ne una de sola i, en funció de quin mòdul hi hagi d'anar, muntar-hi els sensors o no.

A més, tot i que no és necessari per aquest treball, s'ha decidit que a part de l'alimentació de 5 V, la placa també es pugui alimentar amb una bateria de 3 V per a futures aplicacions¹².

Tot i que la configuració dels mòduls permet controlar el nivell de bateria mitjançant l'opció V+, a causa de les incongruències trobades durant les proves portades a terme, s'ha decidit llegir l'estat de la bateria com una entrada analògica més. Per això ha estat necessari adaptar la tensió d'alimentació del mòdul que va de 0 a 3 V al rang de les entrades analògiques del mòdul (0 a 1.2 V). Per fer aquesta adaptació s'ha utilitzat el mateix procediment i amplificador operacional que en els sensors¹³.

Pel que fa a la placa del coordinador, ja que en un futur pròxim pot patir àmplies modificacions, s'ha decidit realitzar-la amb una placa de "topos"¹⁴.

¹² Es poden consultar els esquemes elèctrics i el disseny de la placa a l'Annex 3.

¹³ Es pot consultar l'esquema elèctric a l'Annex 3.

¹⁴ Ídem 13.

4.4 Codi Arduino

En aquest apartat es comenten les parts del codi¹⁵ carregat al microcontrolador de l'Arduino que s'han considerat més rellevants:

4.4.1 Llibreries

S'han inclòs les llibreries necessàries: la de comunicació sèrie, la de la pantalla LCD, la *SPI*, que serveix perquè els pins 10,11,12 i 13 s'utilitzin per comunicar-se amb el *W5100* en l'*Ethernet Shield*; la llibreria *Ethernet*, que permet crear programes que es connecten a Internet, i la *avr/wdt*, que serveix per poder utilitzar el *watchdog*, que permet reiniciar si sorgeix algun imprevist.

4.4.2 Variables

S'han inicialitzat les variables. A la variable *mac[]* tipus *byte* s'hi ha definit l'adreça *Mac* (ve definida de fàbrica) de l'*Arduino Ethernet* que s'utilitza, i a la variable *ip[]*, també de tipus *byte*, s'hi ha definit l'adreça *ip* que utilitzarà l'Arduino. Per trobar-la s'ha agafat l'adreça que té assignada el nostre ordinador i se l'hi ha sumat 10, d'aquesta manera s'ha assegurat que comparteixen la mateixa màscara de subxarxa i la mateixa porta d'enllaç, però amb direccions diferents. Per últim, a la variable *server*, del tipus *char*, s'hi ha definit l'adreça *ip* del nostre ordinador.

A causa del fet que la placa *Arduino Ethernet* té molt poques sortides i entrades digitals, ha calgut utilitzar quatre entrades analògiques com a digitals. Per això s'han declarat com a sortides digitals les 14, 15, 16 i 17¹⁶.

4.4.3 void setup()

S'han configurat les connexions i s'ha activat el *watchdog*.

¹⁵ Es pot consultar el codi complet a l'Annex 3.

¹⁶ Només es pot fer si no tenim cap entrada analògica configurada com a tal, ja que llavors automàticament es configuren totes en analògiques.

4.4.4 Funcions

S'han definit les funcions que serveixen per mostrar els missatges a LCD, com la *void setRemoteState (char a, char b, char c, char value)*, que servirà per activar o desactivar la caldera, el llum o el ventilador. Depenent de les condicions que es compleixin es cridarà aquesta funció, on *a*, *b* i *c* són els tres últims valors de l'adreça SL del mòdul Xbee que controla l'element que volem manipular, i la variable *value* si és 04 vol dir que es desactiva i si és 05 que s'activa.

També s'ha especificat la funció *void httpRequest()*, que és cridada quan es vol establir connexió amb la base de dades per enviar i rebre dades. Si s'estableix connexió mitjançant el port 80, es fa un requeriment HTTP i s'envia la zona, la temperatura actual en la variable lectura i el llum actual en la variable llum a calendari/calendari.php (per defecte ja va directe a htdocs). Cal tenir en compte que al realitzar el GET és necessari col·locar les variables després del nom del *script* utilitzant el caràcter *?* i que llavors cal separar cada variable pel caràcter *&*. HTTP/1.1 informa del tipus d'arxiu.

En aquesta funció també s'actualitza el comptador (milis), que serveix per definir els períodes de connexió. Malgrat tot, si no es pot establir la connexió, s'envia un missatge d'error.

Pel que fa a la funció *String GetValue*¹⁷, també s'ha definit, ja que és necessària, perquè les dades que arriben des de la base de dades ho fan en un *String*, que s'ha de poder dividir per guardar cada dada a la variable correcta. Cal recordar que les dades arriben totes juntes, però que s'ha definit un separador *#*, i que es coneix l'ordre en què estan situades (posició 0, temperatura; posició 1, llum; posició 2, ordre ventilador; posició 3, ordre caldera, i posició 4, ordre llum). Aquesta funció s'aprofita d'això per poder dividir l'*String*.

¹⁷ Aquesta funció s'ha extret de fòrums de programació d'Internet (consultar la webgrafia).

4.4.5 void loop ()

Primer s'ha definit el procediment a seguir per arribar als bytes de la trama API que ens interessen (bytes 15 i 16, valors de temperatura; bytes 17 i 18, valors de llum, i bytes 19 i 20, valors d'estat de la bateria), i llavors s'han tractat els valors per tal d'obtenir ° C en el cas de la temperatura i voltatge quant al llum i l'estat de bateria.

A continuació, s'han emmagatzemat les dades a la variable *valors_web*. Mitjançant *lastIndexOf* es troba la posició de \n (salt de línia). Si aquest és major de 0 s'emmagatzema una part de les dades utilitzant un *substring* a la variable *valors_web* els valors.

Llavors s'ha creat una variable *String* per cada paràmetre que es vol guardar i, mitjançant la funció *GetValue*, s'ha agafat la part de *String* enviada en PHP que ens interessa a cada paràmetre. També s'ha realitzat un *cast* a *int* o *float* en funció de les necessitats.

Per últim, en funció dels paràmetres d'activació o desactivació explicats a l'apartat 4.1, que correspon al disseny de la xarxa, s'envien les ordres a cada mòdul utilitzant la funció *setRemoteState* i s'estableixen els intervals de connexió.

4.5 Disseny de la base de dades i el control remot

4.5.1 Base de dades

Com ja s'ha explicat, s'ha utilitzat el sistema de gestió MySQL i s'han creat les següents taules¹⁸:

TAULA CALENDARI:

En aquesta s'han definit els valors de temperatura i llum objectius en funció d'un període de dies, torn i zona.

TAULA FESTIUS:

En aquesta s'han establert els dies que són festius segons el calendar laboral.

TAULA LECTURES:

En aquest cas s'ha decidit guardar els valors de la temperatura i el llum que hi ha al taller, a més del dia, l'hora i la zona.

TAULA ORDRE:

Aquesta s'ha creat per poder donar les ordres d'activació o desactivació del ventilador, la caldera i el llum manualment.

Així, si el valor és:

- 0: es desactiva.
- 1: s'activa.
- 3: funciona automàticament.

TAULA TORNS:

En aquesta s'han definit els torns de l'empresa en funció de l'hora d'inici i de finalització de cada torn.

¹⁸ Es pot consultar l'estructura de la base de dades a l'Annex 3.

4.5.2 Control remot

El primer pas que s'ha seguit ha estat crear dos arxius PHP que serviran per realitzar la connexió i fer les consultes a la base de dades.

Aquests dos arxius¹⁹ són:

- Connexió
- Calendari

A l'arxiu *connexio.php* s'han especificat els paràmetres per realitzar la connexió. Al *calendari.php* es fa una crida al *connexio.php* per portar a terme la connexió i es realitza la consulta per determinar la temperatura i el nivell d'il·luminació objectiu en funció del dia i l'hora i per enviar les dades de temperatura i nivell d'il·luminació actual a la taula de lectures.

Tot i que es podria haver fet tot el codi en un sol arxiu, s'ha decidit realitzar-ho en dos perquè, si mai és necessari canviar el paràmetre de connexió, sigui més còmode.

L'arxiu *connexio.php* s'ha fet de la següent manera:

- S'ha creat la variable *conn*, que és on s'obre una connexió amb el servidor MySQL especificat. En el nostre cas és el servidor *localhost*, amb l'usuari *root* i sense contrasenya.
- Seguidament s'ha especificat la base de dades que es vol utilitzar, que és l'*arduino_calendari*, mitjançant l'ordre *mysql_select_db*.

Pel que fa a l'arxiu *calendari.php*, els passos a seguir han estat els següents:

- S'ha inclòs l'arxiu connexió.
- S'han creat les variables *hora_actual*, *dia_actual* i *dia_setmana* i s'hi ha guardat els valors d'hora, dia i dia de la setmana en format any-mes-dia.
- La funció *date* dona format a la data i l'hora actuals.
 - *H*: Format hora amb zeros inicials (00 a 23).
 - *i*: Minuts amb zeros inicials (00 a 59).
 - *s*: Segons amb zeros inicials (00 a 59).
 - *Y*: Representació completa de l'any en quatre dígit (per exemple, 2014).
 - *m*: Representació numèrica del mes amb zeros inicials (01 a 12).

¹⁹ Els *scripts* complets es poden consultar a l'Annex 3.

- *d*: Dia del mes amb dos dígit amb zeros inicials (01 a 31).
- *D*: Representació textual del dia de la setmana amb tres lletres (Mon fins Sun).
- S'ha inicialitzat la variable *valors* com un *string* buit.
- Per poder realitzar la cerca, s'ha definit la condició que la variable *dia_setmana* ha de ser diferent a dissabte (Sat) i diumenge (Sun), ja que com a norma general no treballa ningú el cap de setmana. Si la variable *dia_setmana* no compleix la condició, s'ha decidit retornar en valors igual a 12 per la temperatura i a 0 en llum objectius.
- A continuació s'ha mirat si la variable *dia_actual* es trobava dins de la taula de festius.
- També s'ha fet una consulta a la base de dades definida a l'arxiu connexió, on es vol comptar el nombre de vegades que apareix la variable *dia_actual* a la columna *data* de la taula de festius.
- Llavors s'ha comprovat el resultat de la consulta. Si el resultat és 0 vol dir que no hi és, i si el resultat és diferent a 0 vol dir que apareix i, per tant, cal retornar a 12 pel que fa a la temperatura i a 0 pel llum objectiu.
 - *rs*: És la variable on es guarda el resultat de la consulta.
 - *0*: Indica el nombre de files del resultat de la consulta.
 - *0*: Indica el nombre de camp que s'està recuperant.
- Una vegada feta aquesta comprovació, cal saber a quin torn pertany la variable *hora_actual*. Per això s'ha fet una altra consulta. En aquesta s'ha seleccionat *id_torn* de la taula de torns, en què *hora_inici* sigui més petita o igual que *hora_actual* i en què *hora_final* sigui més gran o igual a *hora_actual*. La variable *conn* fa referència a la definida a l'arxiu connexió que estableix la connexió.
- El resultat s'ha guardat a la variable *torn*.
- Un cop descobert que el *dia_actual* no és un dissabte, un diumenge o un festiu, i el torn a què pertany *hora_actual*, el següent pas ha estat anar a la taula del calendari per trobar els valors de temperatura objectiu i llum objectiu. Per fer això s'ha realitzat una consulta, on s'han seleccionat els

valors de temperatura i llum de la taula del calendari, on el torn és igual a la variable *torn*, la zona és igual al valor de *zona* que li passem des del programa Arduino mitjançant el *GET* i la *data_inici* és més petita o igual a *dia_actual* i *data_fi* és més gran o igual a *dia_actual*.

- Llavors s'han guardat els valors a temperatura i llum. A la variable *temperatura*, el valor de la fila 0 i camp 0 de la consulta, i a *llum* el valor de la fila 0 i camp 1. I aquests s'han desat a la variable *valors*. Cal dir que s'ha d'utilitzar el separador # per indicar quan s'acaba cada valor.
- Per saber si es vol actuar manualment sobre el ventilador, els llums o la caldera, s'ha seguit un procediment semblant. Per afegir aquestes tres variables a la variable *valors* mantenint els valors anteriors s'ha utilitzat `.=`.
- Finalment s'ha creat la inserció dels valors de les variables *dia_actual*, *hora_actual*, *zona*, *lectura* i *llum* a les columnes *dia*, *hora*, *id_zona*, *lectura* i *llum* de la taula de lectures.

Un cop elaborats els dos arxius s'han col·locat dins la carpeta calendari i aquesta s'ha ubicat dins la carpeta htdocs de Xampp.

4.6 Disseny de les caixes per a cada mòdul Xbee

Per tal d'aconseguir una bona implantació s'han dissenyat²⁰ i fabricat tres models de caixes per tal de poder col·locar-hi els mòduls:

- Caixa per al mòdul amb els sensors i relé.
- Caixa per al mòdul amb relé.
- Caixa per al mòdul coordinador.

En el disseny de les tres caixes s'ha realitzat una tapa inferior i una de superior que s'uneixen mitjançant cargols avellanats de M4 x 35. El disseny dels dos models per als mòduls és exactament el mateix, amb l'única diferència que la caixa del mòdul amb els sensors disposa d'una obertura per a la resistència LDR .

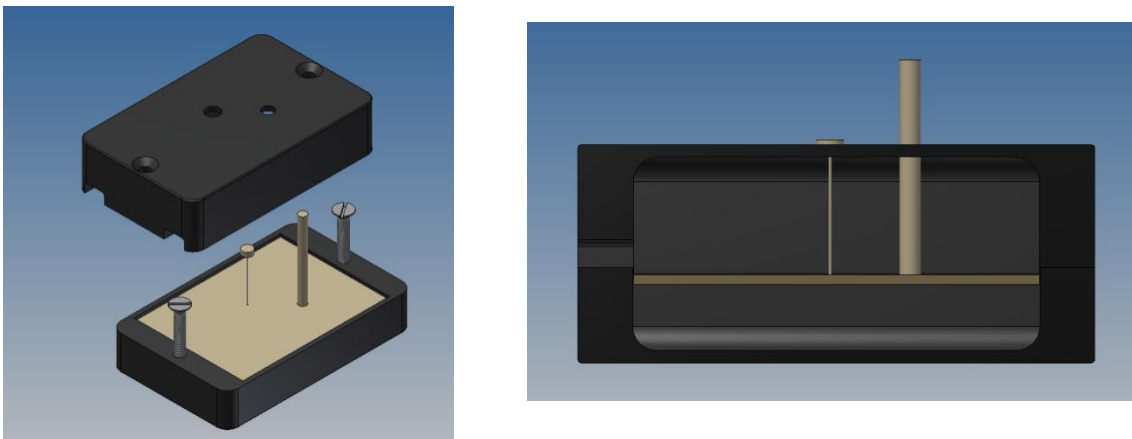


Figura 4-2. Disseny de la caixa per al mòdul amb sensors

Caixa per al mòdul coordinador:

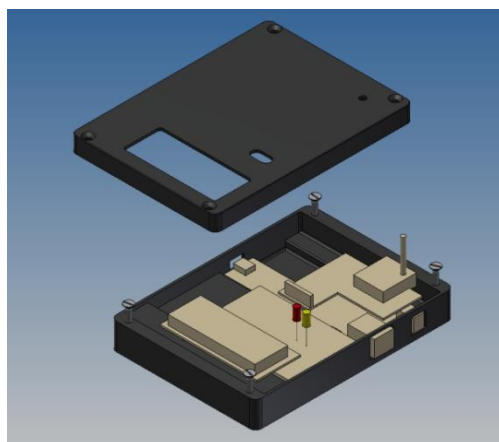


Figura 4-3. Disseny de la caixa del coordinador

²⁰ Es poden consultar els plànols a l'Annex 3.

Una vegada fet el disseny amb el programa Inventor, s'han guardat en format .stp. Abans d'enviar-les a fabricació, però, s'ha realitzat una modificació a la tapa inferior del coordinador per tal de reduir costos:

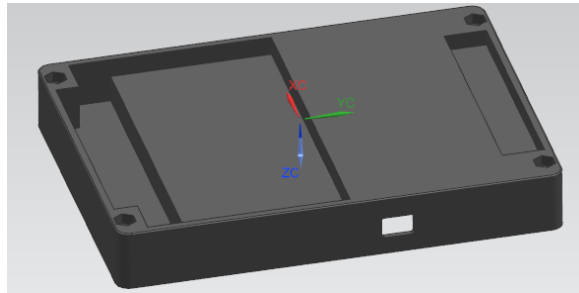


Figura 4-4. Disseny de la caixa del coordinador modificada

Un cop fets els canvis, s'han enviat a l'empresa Initial perquè les fabriqui mitjançant una impressora 3D i, quan s'han rebut les peces, s'han tenyit²¹ per tal d'aconseguir un acabat millor.



Figura 4-5. Caixes fabricades abans i després de tenyir-les

²¹ Es pot consultar el procés de tenyit a l'Annex 3.

4.7 Pressupost del material

Tenint en compte el material utilitzat, s'ha realitzat un pressupost del cost d'aquest²²:

4.7.1 Material per programar

QUANTITAT	MATERIAL	PREU
1	Xbee Explorer V12	25 €
1	Arduino USB2Serial	10 €
TOTAL		35 €

Taula 4-3. Pressupost del material necessari per programar mòduls Xbee

4.7.2 Mòdul Coordinador

QUANTITAT	MATERIAL	PREU
1	Mòdul Xbee Serie2	24 €
1	Xbee Shield	20 €
1	Arduino Ethernet	40 €
1	Pantalla LCD GDM1602K	20 €
1	Resistències, leds	2 €
1	Placa	2 €
1	Caixa	130 €
TOTAL		238 €

Taula 4-4. Pressupost del material necessari per al mòdul coordinador

²² Els preus són sense IVA.

4.7.3 Mòdul Sensor

QUANTITAT	MATERIAL	PREU
1	Mòdul Xbee Serie2	24 €
1	Optoacobrador 6N139	0.98 €
1	Sensor TMP20	0.94 €
1	Sensor LDR VT900	0.70 €
1	Relé estàtic	10 €
1	Regulador de tensió LM317AEMP	1.44 €
3	Amplificador operacional TLV2401	3.15 €
1	Resistències, leds	2 €
1	Placa	2 €
1	Caixa	50 €
TOTAL		95.21 €

Taula 4-5. Pressupost del material necessari per al mòdul sensor

4.7.4 Mòdul Relé

QUANTITAT	MATERIAL	PREU
1	Mòdul Xbee Serie2	24 €
1	Optoacobrador 6N139	0.98 €
1	Relé estàtic	10 €
1	Regulador de tensió LM317AEMP	1.44 €
1	Amplificador operacional TLV2401	1.05 €
1	Resistències, leds	2 €
1	Placa	2 €
1	Caixa	50 €
TOTAL		91.47 €

Taula 4-6. Pressupost del material necessari per al mòdul relé

4.7.5 Total

QUANTITAT	MATERIAL	PREU
1	Material per programar	35 €
1	Mòdul Coordinador	238 €
1	Mòdul Sensor	95.21 €
2	Mòdul Relé	91.47 €
TOTAL		459.68 €

Taula 4-7. Pressupost total del material necessari

Apartat 5 : Conclusions

El projecte ha complert les expectatives plantejades al principi, ja que s'ha aconseguit crear una xarxa sense fils que pot servir com a punt de partida per a una futura automatització de l'edifici. A més, s'ha utilitzat programari lliure i s'ha aconseguit dotar la xarxa d'un control remot.

L'únic objectiu que potser a primera vista pot semblar que no s'hagi assolit és el de crear una xarxa de baix cost econòmic, però cal tenir en compte que el que fa augmentar més el preu són les caixes, perquè s'ha decidit fabricar-les a un nivell molt alt de qualitat i estan fetes a mida

A nivell personal, la realització d'aquest Treball Final de Grau m'ha permès consolidar i posar en pràctica coneixements adquirits durant aquests anys d'estudi.

A més, m'ha possibilitat introduir-me en camps fins ara desconeguts per mi, com per exemple:

- L'Entorn Arduino.
- El protocol Zigbee.
- Els mòduls Xbee.
- Les possibilitats de la comunicació i el control des de xarxes locals.
- La plataforma Xampp.
- El sistema de gestió de base MySql.
- El llenguatge PHP.
- La impressió 3D.

I, per últim, m'ha permès descobrir un camp que m'apassiona, que crec que té un recorregut molt gran i que en un futur molt pròxim formarà part de les nostres vides: l'Internet de les Coses.

Apartat 6 : Possibles millores

Tot i que com s'ha comentat en l'apartat anterior crec que s'han complert els principals objectius, sí que és cert que a mesura que el projecte anava arribant a la seva fi s'han detectat possibles millores que, ja sigui per falta de temps o per la seva complexitat, no s'han pogut realitzar.

Aquestes possibles millores serien:

- Realitzar un pàgina web on es pugui integrar la base de dades, per tal de disposar d'un entorn més amigable per interactuar amb la xarxa.
- Portar a terme un estudi acurat de l'abast dels mòduls Xbee i la incorporació de mòduls configurats com a routers a la xarxa.
- Buscar un procés de fabricació més econòmic per realitzar les caixes dels mòduls.
- Estudiar la manera de facilitar la incorporació de nous mòduls a la xarxa sense haver de modificar cada cop el codi de programació, utilitzant l'EEPROM de l'Arduino i la característica NI dels mòduls Xbee.

Apartat 7 : Bibliografia i Webgrafia

Llibres consultats:

FALUDI, Robert, (2010), *Building Wireless Sensor Networks*. Sebastopol. O'Reilly Media Inc, 2011.

FITZGERALD, Scott et al., (2012), *Arduino Projects Book*. Torí. Arduino LLC, 2013.

Product Manual v1.x.4x - ZigBee Protocol, 2008.

Pàgines web consultades:

ARDUINO

<http://arduino.cc> (Última consulta: 20 de desembre de 2014)

<http://unicarlos.com/ARDUINO/Arduino%20+%20XBee.pdf> (Última consulta: 6 de setembre de 2014)

<http://stackoverflow.com/questions/9072320/split-string-into-string-array> (Última consulta: 3 de gener de 2015)

COMPONENTS DE LA PLACA

<http://es.farnell.com> (Última consulta: 4 de gener de 2015)

INFORMACIÓ GENÈRICA

<http://es.wikipedia.org> (Última consulta: 4 de gener de 2015)

PHP:

<http://php.net/> (Última consulta: 2 de gener de 2015)

XBEE

<http://www.andresduarte.com/arduino-y-xbee> (Última consulta: 18 de juny de 2014)

<http://www.digi.com> (Última consulta: 5 d'agost de 2014)

<http://geekytheory.com/> (Última consulta: 4 d'octubre de 2014)