

DISSENY D'UN SISTEMA SENSE FILS PER LA MONITORITZACIÓ DE SENSORS



Gerard Moreno Aragonès

Grau en Enginyeria de Tecnologies i Serveis de
Telecomunicació

Disseny de sistemes electrònics

Consultor: Carlos Gonzalo Moreno Soriano

Professor responsable: Germán Cobo Rodríguez

11 de Gener de 2021



Aquesta obra està subjecta a una llicència de
[Reconeixement-NoComercial-SenseObraDerivada](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)
[3.0 Espanya de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

FITXA DEL TREBALL FINAL

| | |
|--|---|
| Títol del treball: | <i>Disseny d'un sistema sense fils per a la monitorització de sensors</i> |
| Nom de l'autor: | <i>Gerard Moreno Aragonès</i> |
| Nom del consultor/a: | <i>Carlos Gonzalo Moreno Soriano</i> |
| Nom del PRA: | <i>Germán Cobos Rodríguez</i> |
| Data de lliurament (mm/aaaa): | <i>01/2021</i> |
| Titulació o programa: | <i>Grau en Enginyeria de Tecnologies i Serveis de Telecomunicació</i> |
| Àrea del Treball Final: | <i>Disseny de sistemes electrònics</i> |
| Idioma del treball: | <i>Català</i> |
| Paraules clau | <i>ZigBee, Sensors, Monitorització</i> |
| Resum del Treball (màxim 250 paraules): <i>Amb la finalitat, context d'aplicació, metodologia, resultats i conclusions del treball</i> | |
| <p>Actualment, la indústria està implementant noves tecnologies en la digitalització del seus processos. En els darrers anys, les plantes químiques estan desenvolupant sistemes interconnectats, per tenir un control més exhaustiu de la productivitat, seguretat i qualitat. No obstant, hi ha molts punts en què el control és encara 100% manual, és a dir, l'operari llegeix físicament els diferents paràmetres de la màquina de manera subjectiva. Així doncs, la finalitat del projecte és aplicar un sistema WSN que permeti aconseguir una digitalització d'aquests valors.</p> <p>A nivell metodològic, el disseny es planteja en tres fases ben definides. La primera tracta de dissenyar els dispositius locals que es situen a prop de l'equip, que mesuren les variables i les transmeten. La segona fase tracta el disseny del dispositiu central, que rep i processa les dades, per emmagatzemar-les i mostrar-les en una pantalla. En l'última fase es mostra la construcció del prototip utilitzant els dissenys esmentats anteriorment.</p> | |

Un cop finalitzat el procés de disseny del sistema, es pot dir que els resultats obtinguts són la digitalització i la centralització en un sol dispositiu, de les dades captades per els sensors. A més a més, la visualització des d'aquest dispositiu central, mitjançant una pantalla, permet accedir a aquestes dades de manera senzilla.

Finalment, la conclusió que es pot extreure, és que el sistema ZigBee és ideal per a aquesta aplicació, ja que permet una gran capacitat de connexió de sensors, gràcies als 65.000 dispositius que accepten les seves xarxes.

Abstract (in English, 250 words or less):

Currently, the industry is implementing new technologies in processes digitalization. During the last years, chemical plants have been applying interconnected systems in order to exhaustively control key parameters such as productivity, safety and quality. Unfortunately, there are many processes where the control is still 100% manual, which means the operator reads physically the spectrum of machinery parameters in a subjective way. Therefore, the aim of the project is to apply a WSN system, which will allow a digitalization of these values.

In the methodology, the design is posed in three very clear phases. The first is to design local devices located near the machine. These devices measure the variables and transmit them. The second phase is concerned with the design of the central device, which receives and processes data in order to store and display it on a screen. The last phase exhibits the manufacture of the prototype, using the design mentioned above.

Once the design processes have been completed, the result can be defined as the digitalization and centralization all into a single device of the data captured by the sensors. Additionally, analyzing the data from this central device, through a screen, allows an easy access to these data generated.

In light of above, the conclusion of the project is that ZigBee system perfectly fits for this application, allowing a large capacity of sensors connection, due to the 65.000 devices that accept their networks.

ÍNDEX

| | |
|---|----|
| 1. INTRODUCCIÓ | 1 |
| 1.1. Context i justificació del treball..... | 1 |
| 1.2. Objectius del treball..... | 2 |
| 1.3. Enfocament i mètode seguit..... | 3 |
| 1.4. Planificació del treball | 6 |
| 1.5. Breu sumari de productes obtinguts | 6 |
| 1.6. Breu descripció dels altres capítols de la memòria..... | 7 |
| 2. STATE OF THE ART | 8 |
| 2.1. WSN | 8 |
| 2.2. ZIGBEE | 9 |
| 2.2.1. Accés al medi | 9 |
| 2.2.2. Modulació O-QPSK..... | 10 |
| 2.2.3. Tècniques de modulació Zigbee..... | 10 |
| 2.2.4. Banda de transmissió Zigbee | 11 |
| 2.2.5. Altres capes OSI del sistema Zigbee | 12 |
| 2.2.6. Topologia de xarxa..... | 12 |
| 2.2.7. Prestacions del sistema ZigBee..... | 13 |
| 2.3. Comparativa amb altres tecnologies..... | 13 |
| 3. DISPOSITIUS I PROGRAMARI | 15 |
| 3.1. Hardware | 15 |
| 3.1.1. Dispositius Xbee de DIGI | 15 |
| 3.1.2. Termoresistència PT-100..... | 20 |
| 3.1.3. 28090 Mini Liquid Level Sensor de Parallax..... | 21 |
| 3.1.4. PIC18F6622..... | 22 |
| 3.1.5. LCD EAT162G-3NLED | 23 |
| 3.1.6. Polsadors 8500 Series Subminiature Pushbutton Switches | 23 |
| 3.2. Software..... | 23 |
| 3.2.1. TINA-TI-SPICE (Texas Instruments)..... | 23 |
| 3.2.2. XCTU de DIGI | 24 |
| 3.2.3. PIC C Compiler | 25 |
| 3.2.4. Proteus 8..... | 26 |
| 4. DESENVOLUPAMENT DEL PROJECTE | 27 |
| 4.1. 1a FASE: Disseny dels dispositius locals..... | 27 |
| 4.1.1. Dispositiu de temperatura..... | 27 |
| 4.1.1.1. Sensor de temperatura | 27 |

| | | |
|-----------|---|-----------|
| 4.1.1.2. | Circuit d'adaptació | 28 |
| 4.1.1.3. | Connexió i configuració del dispositiu Xbee | 31 |
| 4.1.1.4. | Disseny de la font d'alimentació | 32 |
| 4.1.2. | Dispositiu de nivell | 37 |
| 4.1.2.1. | Sensor de nivell..... | 37 |
| 4.1.2.2. | Connexió i configuració del dispositiu Xbee..... | 38 |
| 4.1.2.3. | Disseny de la font d'alimentació | 39 |
| 4.2. | 2a FASE: Disseny del dispositiu central | 41 |
| 4.2.1. | Microcontrolador | 44 |
| 4.2.2. | Pantalla LCD..... | 46 |
| 4.2.3. | Polsadors | 47 |
| 4.2.4. | Mòdul XBee | 47 |
| 4.2.5. | Codi..... | 50 |
| 4.2.6. | Connexió i simulació..... | 54 |
| 4.2.7. | Font d'alimentació..... | 56 |
| 5. | FUTURES AMPLIACIONS | 58 |
| 5.1. | Mòdul Bluetooth i aplicació mòbil. | 58 |
| 5.2. | Ampliació de la xarxa..... | 62 |
| 5.3. | Memòria externa..... | 63 |
| 6. | CONCLUSIONS | 65 |
| 7. | GLOSSARI..... | 67 |
| 8. | BIBLIOGRAFIA I WEBGRAFIA..... | 68 |
| 9. | ANNEXOS..... | 71 |

1. INTRODUCCIÓ

1.1. Context i justificació del treball

El projecte en què es basa el treball de final de grau s'emmarca en el sector de la indústria química, però amb poques modificacions podria aplicar-se a molts altres tipus d'indústria. Concretament, el projecte es centra en els processos de producció, en el control i monitorització de variables principals i secundaries com ara temperatures, caudals, pressions, nivells, entre d'altres, i també sobre diferents materials, com ara aigua, oli, etilè, propà, etc.

Actualment, els processos de producció en indústries químiques es controlen principalment mitjançant sistemes de control distribuïts però centralitzats en sales de control, on a través de pantalles i teclats es poden visualitzar i modificar les diferents variables. Aquests sistemes estan connectats als sensors que capten les variables mitjançant cables, què si es tenen en compte els milers de variables que ha de controlar qualsevol unitat de producció, poden arribar a ser molts.

Altres variables de control però, encara es monitoritzen de manera local, és a dir, es llegeixen els paràmetres in situ per part d'un operari, que els anota en un paper que quedarà arxivat a mode de registre. Aquestes acostumen a ser variables secundaries, com ara temperatures i nivells d'oli de lubricació de compressors, pressions i caudals de bombes de sistemes auxiliars, etc.

Segons un estudi de D.velop del 2018, la digitalització en la indústria química està sent molt lenta en comparació amb altres sectors, com es pot comprovar en la següent figura.

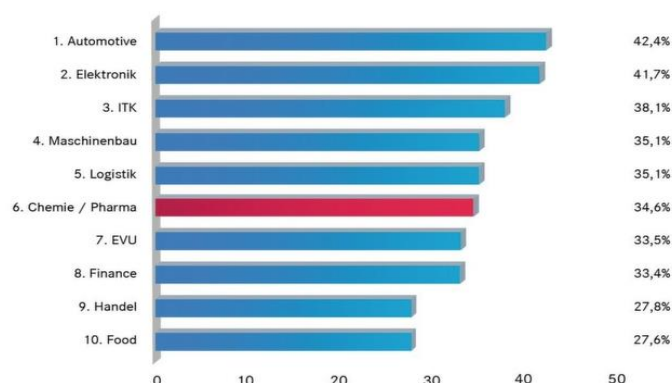


Figura 1. Digitalització de diferents sectors.¹

¹ <https://www.infoplcn.net/plus-plus/mercado/item/105056-procesos-inversion-mediocre-digitalizacion>

Aquestes variables no digitalitzades també tenen la seva importància, i de fet, han de servir per alguna cosa, més enllà de visualitzar l'estat del sistema en aquell precís moment. Les dades que s'obtenen al llarg del temps amb el procés en marxa, es poden utilitzar per a millorar-lo, o per a optimitzar el manteniment de les màquines i d'aquesta manera poder evitar averies.

Així doncs, el projecte pretén crear un prototip per al control del nivell i la temperatura d'oli del càrter d'un compressor d'etilè, que permeti al operari obtenir les dades de períodes de temps relativament llargs, de manera fàcil i informatitzada, i a més, implementant un sistema sense cablejats per evitar saturar més les plantes químiques. Aquest prototip busca la modernització del sector, informatitzant totes les dades per tal de poder treballar fàcilment amb elles, eliminar la subjectivitat de l'operari i facilitar la seva feina, però també permet eliminar el paper i els grans arxius que es necessiten per tal de poder emmagatzemar els registres d'aquestes variables.

1.2. Objectius del treball

- ❖ Digitalitzar les variables de procés que encara no ho estan.
- ❖ Centralitzar aquestes variables en un sol dispositiu.
- ❖ Facilitar l'accés a les dades d'aquestes variables de forma remota.

Objectius específics.

- Dissenyar i construir dos dispositius locals que captin i envien les dades de les variables de nivell i temperatura.
- Dissenyar i construir un dispositiu central que recull les dades de diversos dispositius locals, emmagatzemar-les i permetre la seva visualització mitjançant un display o a través d'un dispositiu mòbil.
- Dissenyar i crear una aplicació mòbil per obtenir les dades que envia el dispositiu central.

1.3. Enfocament i mètode seguit

El treball es divideix en tres fases ben diferenciades, on s'ha tingut en compte la càrrega de treball de cadascuna per tal de poder planificar bé la feina en cadascuna d'aquestes.

- Primera fase.

En aquesta primera part es buscarà tota la informació referent a la tecnologia ZigBee, freqüència de treball, modulació, codificació i es justificarà l'ús d'aquesta enfront altres similars. També s'exposarà la informació disponible sobre el sistema XBee que s'utilitzarà, així com tot l'entorn de software i hardware disponible per al desenvolupament del disseny.

Seguidament es seleccionaran els dispositius XBee que es faran servir i es començarà el disseny dels dispositius locals que són els encarregats de mesurar les variables de nivell i temperatura i transmetre les dades digitalitzades cap al dispositiu central.

Per al disseny, es consideren els sensors que s'utilitzaran en el moment de muntar el prototip, així com els circuits d'adaptació de les mesures per tal de què, el dispositiu XBee pugui transmetre les dades. En aquest apartat també es prepararà la font d'alimentació per a cadascun dels mòduls, així com les configuracions del mòduls XBee.

Finalment, mitjançant eines de simulació, es provaran tots els sistemes per comprovar el seu funcionament.

La primera fase finalitzarà amb l'entrega de la PAC 2.

- Segona fase.

Aquí la feina es centra en dissenyar el dispositiu central, encarregat de rebre les dades dels dos dispositius de la fase anterior. En primer lloc, es seleccionarà el microcontrolador que s'utilitzarà, tenint en compte les necessitats del dispositiu i també es triaran el display, el teclat i el mòdul Bluetooth. Per aquest últim, també es farà una recerca d'informació sobre el protocol Bluetooth i d'aquesta manera conèixer bé el tipus de tecnologia.

Un cop tots els elements s'han seleccionat, s'implementarà el programa de control del microcontrolador. Aquest programa permetrà actuar sobre el mòdul XBee per tal de comunicar amb un sensor o l'altre i triarà el destí

de les dades, ja sigui per pantalla o a través del mòdul Bluetooth. Dins del programa del microcontrolador estarà inclòs el sistema de menús del teclat i pantalla LCD, així com tot el procés d'emmagatzematge de les dades en la memòria interna.

Seguidament, es dissenyaran les fonts d'alimentació per a cada mòdul, així com la configuració del mòdul XBee i s'acabarà aquesta fase amb la simulació dels circuits, del programa de control i amb l'entrega de la PAC 3.

No obstant, en aquesta fase es preveu un punt de decisió en el projecte. Aquest punt de decisió altera la ruta de disseny del dispositiu, escurçant l'abast del projecte. En funció de les dates del diagrama de Gantt, es preveu deixar el mòdul Bluetooth com a una ampliació per al futur, tot i que la recerca del mòdul i de les especificacions del protocol es duran a terme igualment.

- Tercera fase.

En la última part del projecte, es treballarà en l'aplicació per a dispositiu mòbil basat en el sistema operatiu Android, que connectarà amb el dispositiu central i permetrà dur a terme les mateixes funcions que amb la pantalla i el teclat del dispositiu central.

Concretament, es definirà una estructura visual intuïtiva i fàcil de fer funcionar, així com la implementació de les funcions necessàries per tal de complir amb un dels objectius del projecte.

Es té en compte que si en la fase dos es decideix eliminar el mòdul Bluetooth del prototip, aquesta part quedarà directament eliminada del projecte i només quedarà mencionada en la memòria final, sense entrar en cap mena de detalls ni disseny.

En la tercera fase, s'inclou la fabricació del prototip. Per a la seva realització, es farà una recerca dels dispositius i materials necessaris, per tal de obtenir un pressupost ajustat del que costaria i es faria la comanda. A continuació, es generaran les PCB dels tres dispositius, es carregarà el programa al microcontrolador i es realitzarà el muntatge de tots els elements. S'aplicaran les configuracions dels mòduls XBee esmentades en les dos fases anteriors i es realitzaran les proves de funcionament per assegurar el correcte funcionament dels dispositius. Per a aquestes

proves, es construirà un petit dipòsit de metacrilat, on s'instal·laran els dos sensors i es provocaran variacions de temperatura i de nivell.

En aquesta fase també es preveu un punt de decisió en funció del compliment del pla de treball. Es tractaria de no realitzar el muntatge físic dels dispositius, i només mencionar un pressupost aproximat del que costaria la seva construcció.

La tercera i última fase conclou amb la redacció de la memòria basada amb cadascun dels documents produïts fins llavors, més tota la documentació extreta d'aquesta. També s'ha previst un temps per a la gravació i la realització del vídeo de presentació del treball.

1.4. Planificació del treball

| Fases | Tasques | Duració | Inici | Final |
|-----------------------|---|---------|------------|------------|
| Planificació | PAC 1 | 13 dies | 17/09/2020 | 05/10/2020 |
| Sistema Xbee | | 4 dies | 28/09/2020 | 01/10/2020 |
| | Informació i comparativa | 2 dies | 28/09/2020 | 29/09/2020 |
| | Especificacions i mòduls | 1 dia | 30/09/2020 | 30/09/2020 |
| | Redacció document | 1 dia | 01/10/2020 | 01/10/2020 |
| Dispositius finals | | 20 dies | 06/10/2020 | 02/11/2020 |
| | Sensors | 5 dies | 06/10/2020 | 12/10/2020 |
| | Adaptació i conversio A/D | 4 dies | 13/10/2020 | 16/10/2020 |
| | Font alimentació | 4 dies | 19/10/2020 | 22/10/2020 |
| | Simulació | 3 dies | 23/10/2020 | 27/10/2020 |
| | Configuració Xbee | 1 dia | 28/10/2020 | 28/10/2020 |
| | PAC2 | 3 dies | 29/10/2020 | 02/11/2020 |
| Dispositiu central | | 25 dies | 03/11/2020 | 07/12/2020 |
| | MCU | 2 dies | 03/11/2020 | 04/11/2020 |
| | Display i teclat | 1 dia | 05/11/2020 | 05/11/2020 |
| | Informació i caracteristiques Bluetooth | 2 dies | 06/11/2020 | 09/11/2020 |
| | Mòdul Bluetooth | 2 dies | 10/11/2020 | 11/11/2020 |
| | Programa de control | 6 dies | 12/11/2020 | 19/11/2020 |
| | Configuració Xbee | 1 dia | 20/11/2020 | 20/11/2020 |
| | Font alimentació | 5 dies | 23/11/2020 | 27/11/2020 |
| | Simulació | 3 dies | 30/11/2020 | 02/12/2020 |
| | PAC3 | 2 dies | 03/12/2020 | 07/12/2020 |
| Aplicació mòbil | | 8 dies | 08/12/2020 | 17/12/2020 |
| | Eina desenvolupament | 1 dia | 08/12/2020 | 08/12/2020 |
| | Estructura d'interfície d'usuari | 1 dia | 09/12/2020 | 09/12/2020 |
| | Funcions | 5 dies | 10/12/2020 | 16/12/2020 |
| | Proves | 1 dia | 17/12/2020 | 17/12/2020 |
| Muntatge del prototip | | 12 dies | 18/12/2020 | 04/01/2021 |
| | Pressupost | 1 dia | 18/12/2020 | 18/12/2020 |
| | Compra dels elements | 1 dia | 21/12/2020 | 21/12/2020 |
| | PCB | 3 dies | 22/12/2020 | 24/12/2020 |
| | Configurar MCU | 1 dia | 25/12/2020 | 25/12/2020 |
| | Configurar Xbee | 1 dia | 28/12/2020 | 28/12/2020 |
| | Soldar | 2 dies | 29/12/2020 | 30/12/2020 |
| | Maqueta | 2 dies | 31/12/2020 | 01/01/2021 |
| | Proves | 1 dia | 04/01/2021 | 04/01/2021 |
| Memòria (PAC4) | | 5 dies | 05/01/2021 | 11/01/2021 |
| | Redacció document | 3 dies | 05/01/2021 | 07/01/2021 |
| | Video presentació | 2 dies | 08/01/2021 | 11/01/2021 |

1.5. Breu sumari de productes obtinguts

Cap.

1.6. Breu descripció dels altres capítols de la memòria

En el segon capítol es tracta l'estat de la qüestió pel que fa a les xarxes de sensors sense fils. Tot seguit s'analitza tècnicament el sistema ZigBee fent referència a totes les capes del seu model OSI. S'acaba el capítol comentant les seves possibles topologies de xarxa, les seves prestacions i una comparació respecte tecnologies semblants.

En el tercer capítol s'exposen tots els elements utilitzats en el projecte, tant pel que fa al hardware com al software, amb una breu descripció.

El quart capítol és el més important, ja que es tracta el disseny i construcció del sistema. En aquest punt es diferencien perfectament les tres etapes planificades en el treball. La primera fase que correspon al disseny dels dispositius locals d'obtenció de les dades de temperatura i de nivell. La segona que tracta del disseny del dispositiu central. I, per últim la tercera fase, on s'explica la construcció del prototip.

En el cinquè capítol es comenten breument les futures ampliacions que es poden aplicar al sistema dissenyat, per tal de poder ampliar les seves prestacions i versatilitat.

En el darrer capítol es tracten les conclusions del projecte.

2. STATE OF THE ART

2.1. WSN

Les Wireless Sensor Network són un grup de sensors sense fils que serveixen per monitoritzar les condicions de l'entorn. En els últims anys, l'interès en aquests tipus de sistemes a crescut per part de la comunitat de recerca. International Electrotechnical Commission, en una recent publicació, informa que la instal·lació d'aquests dispositius sense fils es va incrementar en un 553% entre els anys 2011 i 2016. Entre aquests, el 39% dels dispositius s'utilitzen per a noves aplicacions de WSN.

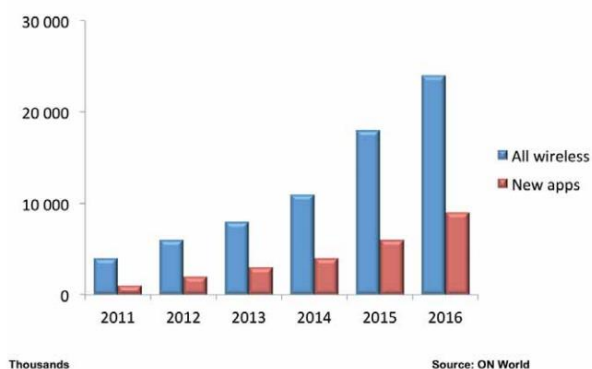


Figura 2. Evolució de noves aplicacions amb dispositius sense fils.²

Les WSN permeten explorar múltiples aplicacions per monitoritzar diverses variables des d'un lloc remot. Les aplicacions més comuns on s'utilitzen aquestes xarxes són la domòtica, la salut, el medi ambient, la seguretat, l'agricultura, entre d'altres, com la indústria que és el sector que ocupa aquest projecte.

Les xarxes es formen mitjançant nodes distribuïts al voltant d'una zona per intercanviar informació sobre mesures físiques com la temperatura, la pressió, el nivell, el caudal, etc. Així, els principals components d'aquests nodes són els sensors de mesura, els transmissors/receptors de dades, un petit microcontrolador que incorpora un convertidor A/D per processar mínimament les dades, i una petita font d'energia, ja que una de les principals característiques és el seu baix consum.

Com es veurà més endavant, les velocitats d'aquests sistemes no són gaire elevades, degut a la poca amplada de banda que utilitzen. Per això s'acostumen a utilitzar en aplicacions on la latència de les dades no és un impediment. Una

² <https://www.iec.ch/whitepaper/pdf/iecWP-internetofthings-LR-en.pdf>

de les característiques principals és la comunicació bidireccional, ja que aquests sistemes aconseguixen rebre informació dels sensors però també enviar-los ordres.

Els nodes, que poden arribar a ser milers, acaben concentrant les dades en un dispositiu més potent que actua d'enllaç cap a xarxes externes on s'emmagatzemen o s'utilitzen.

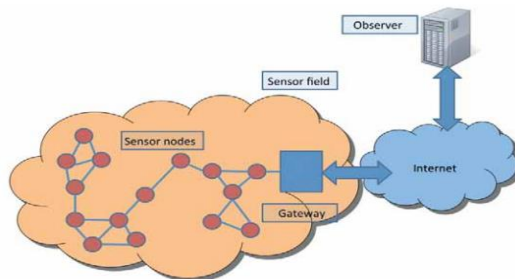
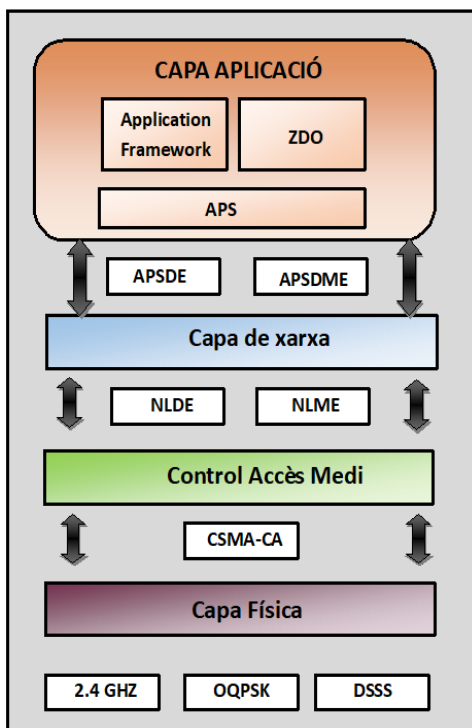


Figura 3. Xarxa WSN. ³

2.2. ZIGBEE



ZigBee és una tecnologia que s'inclou dins del grup de les WSN i es regeix per l'estàndard IEEE 802.15.4, que defineix les dos primers capes del model OSI, la capa física i la capa d'accés i control del medi. Les capes restants, xarxa i aplicació, venen definides per l'estàndard desenvolupat per la ZigBee Alliance.

En la figura, es mostra la pila OSI que utilitza ZigBee per comunicar entre els seus dispositius.

Tot seguit es comenten cadascuna d'elles.

Figura 4. Arquitectura del protocol ZigBee. Creació pròpia.

2.2.1. Accés al medi

Aquest estàndard utilitza la codificació per detecció de portadora amb prevenció de col·lisions (CSMA-CA). Aquesta mecanisme d'accés al medi permet reduir la

³ <https://www.iec.ch/whitepaper/pdf/iecWP-internetofthings-LR-en.pdf>

probabilitat de col·lisió de les transmissions, fent que el dispositiu que vol transmetre, escolti la xarxa abans d'enviar qualsevol paquet de dades. Quan el dispositiu té la confirmació que el canal està lliure envia la trama i espera la confirmació de rebut des de l'altre dispositiu. Si no la rep, vol dir que s'ha produït una col·lisió entre dos dispositius que tenien el canal lliure, aleshores, espera una temps determinat per a retransmetre la trama una altre vegada. Aquesta tècnica, a diferència de la CSMA-CD (Codificació per detecció de portadora amb detecció de col·lisions), si segueix sense rebre la confirmació de rebut de l'altre dispositiu, va augmentant exponencialment el temps d'espera per a retransmetre.

2.2.2. Modulació O-QPSK.

ZigBee, mitjançant l'estàndard 802.15.4, utilitza la modulació O-QPSK (Offset Quadrature Phase-Shift Keying), una variant de la modulació QPSK, molt ineficient espectralment per a comunicacions digitals, ja que la potència del senyal surt de la banda assignada.

La modulació utilitza un pols de Nyquist amb un factor roll-off que limita l'amplada de banda del senyal modulad, però aquest fet produeix un augment de la sensibilitat a les no-linealitats dels amplificadors, ja que el senyal modulad deixa de tenir una envolupant constant. Per pal·liar aquest efecte no desitjat, es desplaça el component en quadratura mig temps de símbol respecte el component en fase, evitant canvis bruscos en l'envolupant.

2.2.3. Tècniques de modulació Zigbee

L'estàndard 802.15.4 utilitza la tècnica DSSS (Direct Sequence Spread Spectrum) que permet xifrar les dades que s'envien al medi, reduir la detectabilitat de la transmissió i augmentar la robustesa davant les interferències en el canal utilitzant seqüències pseudoaleatòries. Aquestes seqüències són binàries, de gran velocitat i tenen la característica de semblar aleatòries i infinites per part d'observadors externs, però per el transmissor i el receptor no.

Les seqüències s'utilitzen per a modular els diferents senyals i distribuir la seva potència dins d'una amplada de banda més gran, a raó de L_c (Relació d'eixamplament), reduint la densitat de potencia espectral.

La transmissió només pot ser desxifrada pel receptor que coneix la seqüència, mentre que els demés receptors ho perceben com a soroll.

2.2.4. Banda de transmissió Zigbee

Les bandes de transmissió d'aquest estàndard són les bandes ISM (Industrial, Scientific, Medical), els quals no necessiten llicència per a operar si es respecten els nivells de potència de transmissió. Aquestes bandes són:

- 2.4 GHz a tot el món.
- 915 MHz a Amèrica.
- 868 MHz a Europa.

La banda que s'estudiarà serà la de 2.4 GHz, ja que els dispositius que es seleccionaran utilitzen aquesta banda i permeten una velocitat de transmissió més elevada.

Aquesta banda està dividida en 16 canals (del 11 al 26), de 2 MHz d'amplada de banda cadascun i 5 MHz de separació entre portadores

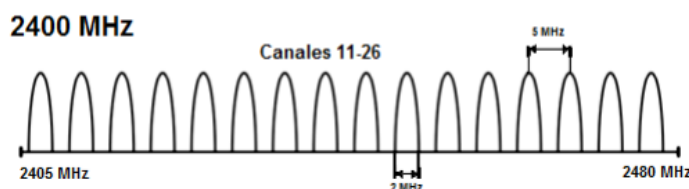


Figura 5. Distribució de canals del protocol ZigBee dins de la banda 2.4 GHz. ⁴

Segons la normativa espanyola, en l'Ordre ETU/1033/2017, del 25 d'octubre, per la que s'aprova el quadre nacional d'atribució de freqüències i seguint les recomanacions de la Unió Europea (ETSI-EN-300-328), imposa una potència isotròpica radiada equivalent (PIRE) màxima de 100 mW (20 dBm) per a les transmissions en aquesta banda i per a xarxes sense fils de comunicacions electròniques en àrees locals.

⁴ (<http://repositorio.unicauca.edu.co:8080/bitstream/handle/123456789/1834/EVALUACION%20Y%20ANALISIS%20DEL%20DESEMPE%20c3%91O%20A%20NIVEL%20FISICO%20DE%20UN%20SISTEMA%20DE%20COMUNICACI%20c3%93N%20DE%20DATOS%20PUNTO%20A.pdf?sequence=1&isAllowed=y->).

2.2.5. Altres capes OSI del sistema Zigbee

Segons Zigbee Alliance (2015), la capa de xarxa proporciona, mitjançant dues entitats, les funcionalitats necessàries per al bon funcionament entre la capa d'aplicació i l'estàndard 802.15.4. Una de les entitats és NLDE, que es l'encarregada de proporcionar el servei de transmissió de dades. L'altre és NLME, ofereix el servei de gestió de la xarxa mitjançant una base de dades d'objectes d'aquesta.

La capa d'aplicació inclou Application Framework, ZigBee Device Object (ZDO) i Application support Sub-Layer (APS). La primera permet definir fins a 254 tipus d'objectes d'aplicació diferents per a cada ZDO. Aquests ZDO representen classes de funcionalitats que fan d'interfície entre els objectes d'aplicació, esmentats anteriorment i la APS. Finalment, la subcapa APS fa d'enllaç amb la capa de xarxa mitjançant dues entitats, equivalents a les NLDE i NLME, que són APSDE i APSME respectivament.

2.2.6. Topologia de xarxa.

Segons Dávila (2017), ZigBee pot definir 3 tipus de dispositius:

- Coordinador: és el dispositiu que inicia, escull els paràmetres claus i controla la xarxa i manté la resta de dispositius dins d'aquesta.
- Router: és un dispositiu que permet ampliar la xarxa i es capaç de encaminar les dades dins d'ella i cap a dispositius que no només siguin coordinadors.
- End Device: tota la resta de dispositius de la xarxa que es comuniquen amb el coordinador o, si existeixen, amb els routers.

Les topologies de xarxa, és a dir. les diferents maneres en què s'associen els dispositius són:

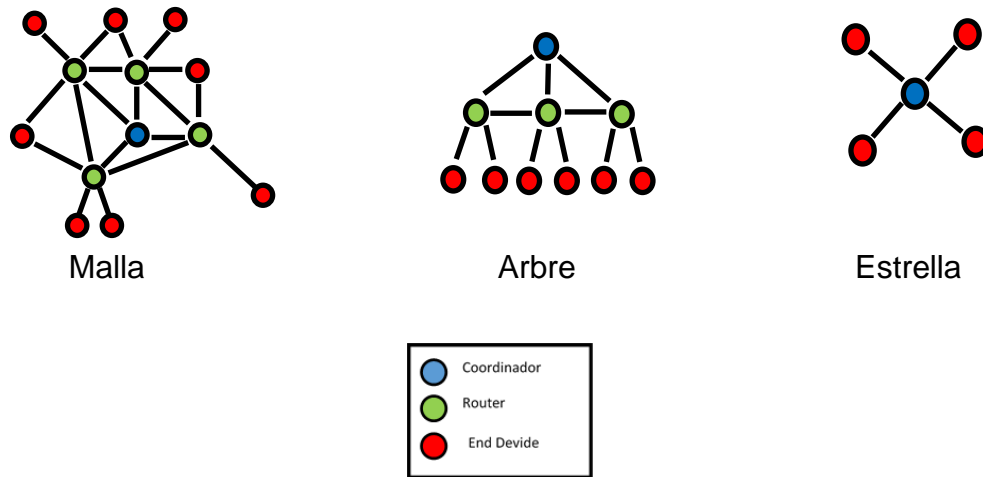


Figura 6. Topologies de xarxa ZigBee. Creació pròpia.

2.2.7. Prestacions del sistema ZigBee.

Segons Zigbee Alliance (2020), les característiques del sistema ZigBee són:

- Protocol de xarxa: ZigBee Pro 2015.
- Nombre màxim de dispositius: 65000.
- Tecnologia de transmissió: IEEE 802.15.4.
- Banda: 2.4 GHz, amb 16 canals de 2 MHz cadascun.
- Taxa de transmissió: fins a 250 kbps.
- Xifrat: AES-128.
- Distància de transmissió: 300 m amb visió directe i entre 75-100 m en interiors.
- Baix consum d'energia.

2.3. Comparativa amb altres tecnologies

Algunes de les tecnologies que també s'inclouen en les WSN són Z-WAVE, LoRa, entre d'altres. En la següent taula s'ofereix una comparativa entre els tres sistemes segons Lora Alliance (2015) i Z-WAVE (2020).

| | ZigBee | Z-WAVE | LoRa |
|-------------------|---------------|--------------|------------|
| Tecnologia | IEEE.802.15.4 | ITU-T G.9959 | LoRaWan |
| Nº de dispositius | 65000 | 232 | 255 |
| Banda | 2.4 GHz | 868 MHz | 868 MHz |
| Distància | 300 m | 100 m | 10 – 20 km |
| Taxa | 250 kbps | 100 kbps | 50 kbps |
| Consum | Baix | Baix | Molt baix |
| Preu | 20 -30 € | 8-10 € | 10 – 20 € |

Taula 1. Comparativa entre xarxes WSN. Creació pròpia.

3. DISPOSITIUS I PROGRAMARI

3.1. Hardware

3.1.1. Dispositius Xbee de DIGI

Els dispositius bàsics de XBee són els mòduls RF. Existeixen 3 sèries de mòduls, la sèrie 1, la sèrie 2 i la sèrie 3. També es poden diferenciar per a cada sèrie, entre el dispositiu normal i el dispositiu PRO, essent aquest últim més potent però més car. En aquest projecte s'utilitzarà la sèrie S2C, ja que la 1 està gairebé obsoleta i la 3 és molt nova i més cara.



Figura 7. Mòdul RF XBee S2C.⁵

Aquests mòduls es basen en ZigBee i permeten dos tipus de comunicació entre ells, el mode AT i el mode API.

MODE AT

És el mode per defecte en els mòduls RF i és semblant a una comunicació de port serial. El dispositiu forma la trama amb les dades que arriben al pin D_{IN} per a enviar-les a un altre dispositiu. El dispositiu emmagatzema en un buffer les dades que ha de transmetre fins que no rep més paràmetres, rep el màxim permès a la trama (100 bytes) o rep una comanda de configuració.

MODE API

És un mode alternatiu al mode AT que ofereix moltes més prestacions. S'utilitza per gestionar xarxes grans i permet la comunicació amb diferents dispositius. Aquest mode utilitza trames estructurades que permeten dirigir els paquets en un ordre determinat. Permet enviar dades a diferents dispositius o esbrinar de quin dispositiu arriben les dades. Les trames són creades per un microcontrolador extern per tal de gestionar la comunicació.

⁵ <https://components101.com/wireless/xbee-s2c-module-pinout-datasheet>

Segons DIGI (2020), la trama API té aquesta forma:

| Start delimiter | Length | | Frame data | | | | | | | | Checksum |
|-----------------|--------|-----|----------------|------|---|---|---|---|-----|---|-------------|
| | | | Frame type | Data | | | | | | | |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | ... | n | n+1 |
| 0x7E | MSB | LSB | API frame type | Data | | | | | | | Single byte |

Figura 8. Trama API dels dispositius XBee.

- Delimitador inicial: permet als dispositius detectar un paquet nou i sempre és 0x7E.
- Longitud: indica la longitud de les dades de la trama sense incloure el delimitador, la longitud i el checksum.
- Tipus de trama: es tracta d'un identificador de tipus de trama API.
- Dades: conté les dades a transmetre i la seva forma depèn del identificador vist anteriorment. Per a les trames de transmissió i recepció de dades inclou la adreça de destinació o origen respectivament.
- Checksum: és una suma hash de verificació que permet comprovar la integritat de les dades. Els dispositius no processen les trames que no compleixen aquesta suma i descarten les dades de la trama.

Les principals especificacions dels mòduls utilitzats són:

| | |
|----------------------------|-----------------|
| Taxa de dades | 250 kbps |
| Rang interiors | 60 m |
| Rang exteriors | 1200 m |
| Potència de transmissió | +5 / +8 dBm |
| Sensibilitat de recepció | -100 / -102 dBm |
| Freqüència | 2.4 GHz |
| Entrades ADC | 4 (10 bits) |
| Entrades/Sortides Digitals | 15 |
| Tensió d'alimentació | 2.1 – 3.6 V |

Taula 2. Especificacions mòdul XBee S2C. ⁶

⁶ https://www.mouser.es/datasheet/2/111/digiinternational_10192016_s2c-xbeedigimesh-24-1163821.pdf

El mòdul XBee S2C té aquesta disposició amb 20 pins.

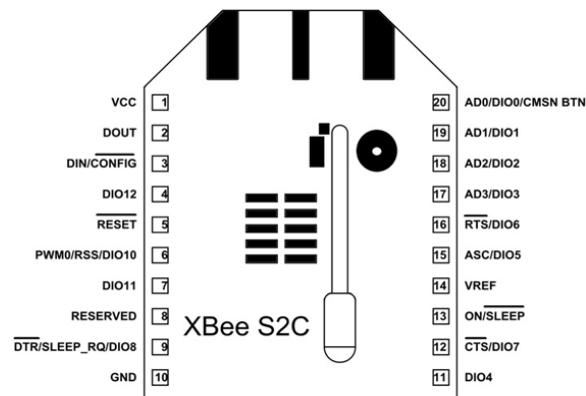


Figura 9. Distribució de pins en el mòdul XBee S2C.⁷

Els pins més importants per a l'aplicació són:

- Vcc: és l'alimentació del transmissor i requereix 3.3 V.
- GND: és la referència a massa del dispositiu.
- AD0: es tracta d'una entrada analògica de dades al transmissor que inclou el convertidor analògic-digital i s'utilitzarà per a la sortida del circuit d'adaptació de la termoresistència.

Els paràmetres més importants a configurar, mitjançant el software XCTU, en els mòduls XBee es divideixen en diferents grups i són:

- Xarxa.
 - PAN ID: es l'identificador de xarxa i ha d'estar dins del rang de valors que ens indica el mateix mòdul. El valor ha de ser el mateix per a tots els mòduls de la xarxa.
 - Coordinador Enable: s'utilitza per definir el mòdul XBee que farà la funció de coordinador. Només s'ha d'activar en un dels mòduls RF de la xarxa.

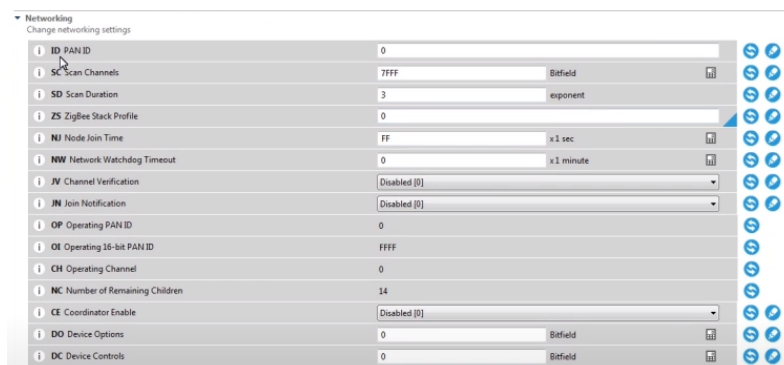


Figura 10. Configuració de xarxa dels mòduls XBee.

⁷ <https://components101.com/wireless/xbee-s2c-module-pinout-datasheet>

- Direccionament.
 - o Serial Number High/Low: es tracta d'un valor únic per a cada mòdul.
 - o Destination Address High/Low: permet indicar amb quin altres mòduls s'ha de comunicar. El valor per defecte és 0 i indica que només es comunicarà amb el coordinador. En cas que es vulgui que comuniqui amb algun altre mòdul en concret, s'ha de incloure en aquest paràmetre els serial number high/low respectivament.

En cas que es necessiti enviar en broadcast (múltiples mòduls), el valor high s'ha de deixar en 0 i el low ha de tenir el valor FFFF.

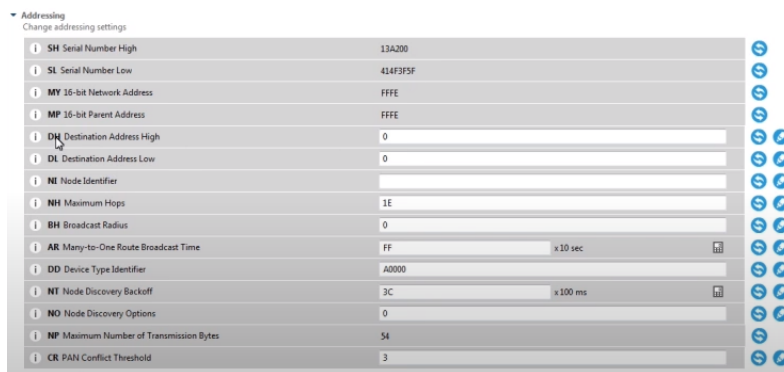


Figura 11. Configuració del direccionalment dels mòduls XBee.

- Interfície serial.
 - o API Enable: per defecte es troba en mode AT, però es pot modificar a mode API per poder treballar amb comandes API mitjançant la creació de trames.



Figura 12. Configuració d'interfície serial.

- Modes Sleep.
 - o Sleep Mode: per defecte es troba desactivat i es la configuració per a un router. Si s'activa Cyclic Sleep, el mòdul queda configurat com un End Device i dorm segons el paràmetre de temps que li apliquem.
 - o Cycling Sleep Period: paràmetre de temps que dormirà l'End Device.

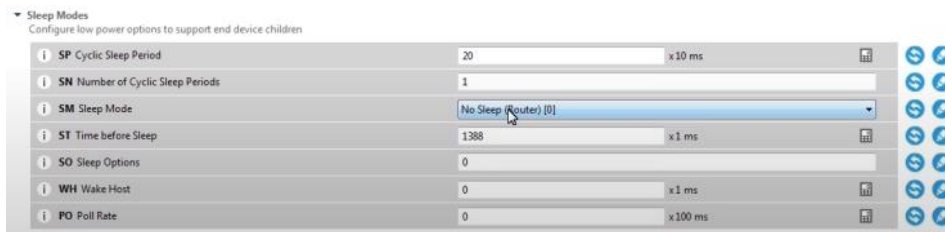


Figura 13. Configuració mode Sleep.

- Entrades/Sortides.
 - o AD0/DIO0: permet seleccionar la funció del pin, en aquest cas, entre la entrada/sortida digital 0 o l'entrada analògica al convertidor digital 0.

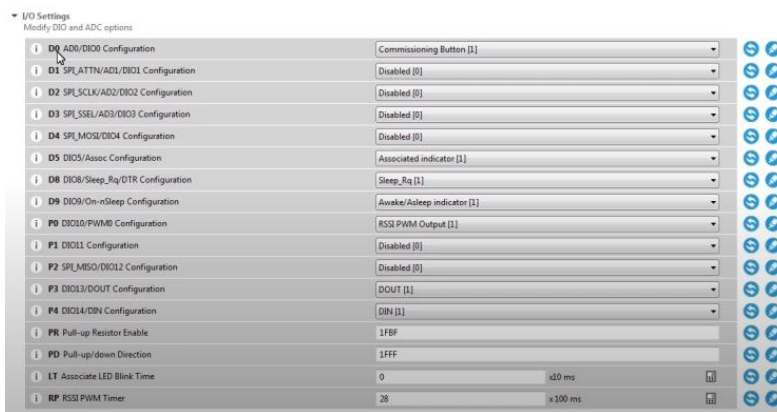


Figura 14. Configuració d'entrades i sortides.

L'altre dispositiu necessari és un XBee-USB, fabricat per Gravitech. Es tracta d'un adaptador per als mòduls RF de XBee que permet configurar-los i accedir a les dades a través de la connexió serial (USB 1.0 i 2.0) de la que també disposen.



Figura 15. Mòdul adaptador USB per a XBee.⁸

El dispositiu és compatible amb tots els mòduls XBee i no necessita alimentació externa. Per realitzar les configuracions es treballa, com s'ha vist, amb el software XCTU de Digi des de un PC. A més, aquest programa permet l'accés a les dades que rep i envia el mòdul.

⁸ https://www.mouser.es/datasheet/2/595/XBee-USB_Manual-221002.pdf

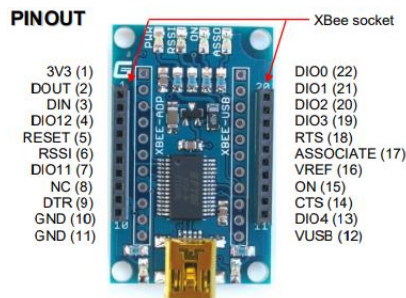


Figura 16. Distribució dels pins de l'adaptador.⁸

Com s'aprecia en la figura anterior, el mòdul s'incrusta en els pins de l'adaptador i mitjançant el port USB es connecta al PC. Una altra de les utilitats d'aquest dispositiu és que permet adaptar el mòdul XBee a una protoboard, ja que per mesures aquests no són compatibles.

3.1.2. Termoresistència PT-100

El fabricant Wika és molt conegut i utilitzat a les indústries en general. La termoresistència que s'utilitzarà és el model TF45, amb les següents característiques:

- Classe A
- Rang de mesura: -50 fins a 200 °C.
- 4 cables de connexió.
- Protecció IP67.
- Longitud del cable: 2m.
- Cable de silicona.



Figura 17. Termoresistència PT100, model TF45 de Wika.⁹

⁹ https://media.automation24.com/datasheet/es/DS_TE6715_es_es_28573.pdf

3.1.3. 28090 Mini Liquid Level Sensor de Parallax.



Figura 18. Sensor de nivell de líquid mini de Parallax.¹⁰

Les característiques del sensor són:

- Voltatge d'operació: 2-5 V.
- Sortida analògica.
- Profunditat detectable: 4.8 cm.

La placa de connexió inclou 3 pins:

| Nº Pin | Símbol | Descripció |
|--------|--------|-------------------|
| 1 | AOUT | Sortida analògica |
| 2 | GND | Massa |
| 3 | Vcc | Alimentació |

Taula 3. Nomenclatura dels pins del sensor.¹¹

La fulla de dades del sensor inclou una taula de equivalència entre nivell i sortida analògica:

| Nivell (cm) | V _{out} (V) |
|-------------|----------------------|
| 0 | 0 |
| 0.5 | 1.3 |
| 1 | 1.53 |
| 1.5 | 1.62 |
| 2 | 1.69 |
| 2.5 | 1.74 |
| 3 | 1.77 |
| 3.5 | 1.81 |
| 4 | 1.84 |
| 4.5 | 1.86 |
| 4.8 | 1.88 |

Taula 4. Relació profunditat/sortida analògica.¹¹

¹⁰ <https://www.mouser.es/new/parallax/parallax-28090-sensor/>

¹¹ <https://www.mouser.es/pdfDocs/28090-Mini-Liquid-Level-Sensor-User-Manual.pdf>

3.1.4. PIC18F6622



Figura 19. PIC18F6622.¹²

Aquest microcontrolador de 8 bits amb arquitectura PIC i AVR fabricat per Microchip s'utilitza majoritàriament en el sector de l'automoció, concretament en els sistemes d'interfície humana, per això és ideal per a l'aplicació que s'està dissenyant. La temperatura de funcionament també és una de les característiques que interessen en el disseny, ja que pot estar entre -40°C i 150°C, cosa que la fan adequada per a la indústria química on es troba a la intempèrie i sovint a prop de canonades amb temperatures de més de 100°C.

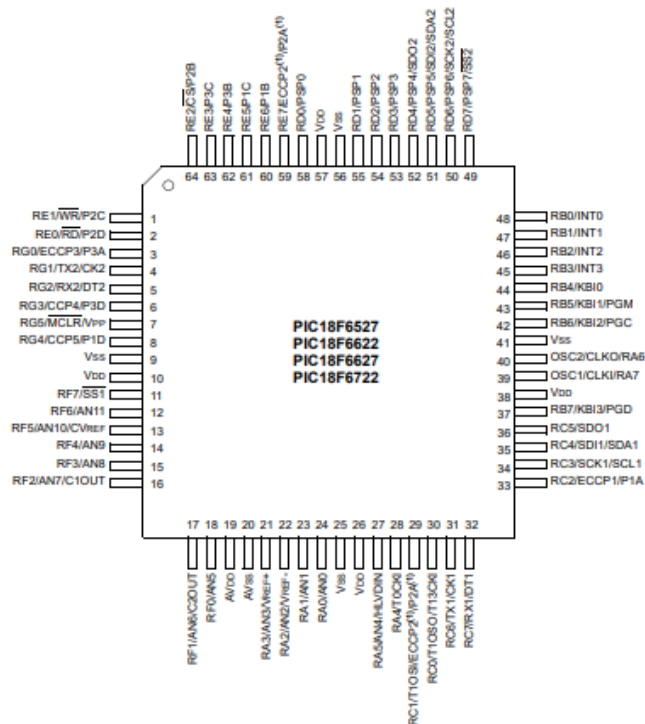


Figura 20. Distribució dels pins en el PIC18F6622.¹³

¹² <https://www.mouser.es/new/microchip/microchip-automotive-8bit-mcus/>

¹³ <https://www.mouser.es/datasheet/2/268/39646c-17052.pdf>

3.1.5. LCD EAT162G-3NLED



Figura 21. LCD¹⁴

3.1.6. Polsadors 8500 Series Subminiature Pushbutton Switches



Figura 22. Polsadors.

3.2. Software

3.2.1. TINA-TI-SPICE (Texas Instruments)

Aquest programa s'utilitza per a dissenyar i simular circuits reals. Pertany a Texas Instruments i és una versió gratuïta per a estudiants que es pot trobar a la seva web. La interfície és molt senzilla i molt intuïtiva que et permet construir circuits fàcilment.

El software inclou molts tipus de components, des dels bàsics com ara resistències, condensadors, fonts d'alimentació..., fins a dispositius específics de la gamma que ofereix aquest fabricant de components, com ara amplificadors, transistors, díodes,... També permet crear blocs personalitzats o afegir llibreries amb altres components.

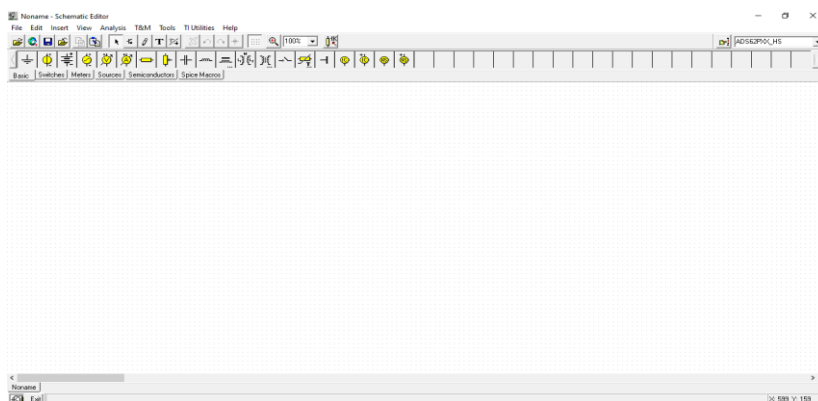


Figura 23. Interfície del software TINA-TI-SPICE.

¹⁴ <https://www.mouser.es/datasheet/2/127/t162g-3-553439.pdf>

L'aplicació és capaç de fer anàlisis en corrent continu i en corrent alterna, crear gràfics sobre paràmetres variables i analitzar punts en aquests gràfics.

3.2.2. XCTU de DIGI

Aquest programa és gratuït i distribuït per DIGI que es pot descarregar des de la seva web. Es tracta d'una interfície d'usuari per a programar i provar els diferents mòduls RF. La part superior esquerra de la pantalla principal ens permet afegir mòduls XBee connectats per USB. També permet descobrir mòduls RF remotament, però només si ja hi ha algun dispositiu local connectat. En la part superior dreta hi ha dos grups de funcions, el més centrat que s'utilitza per a configurar el propi programa i el que es situa més a la dreta que ens permet gestionar la xarxa de diferents maneres.

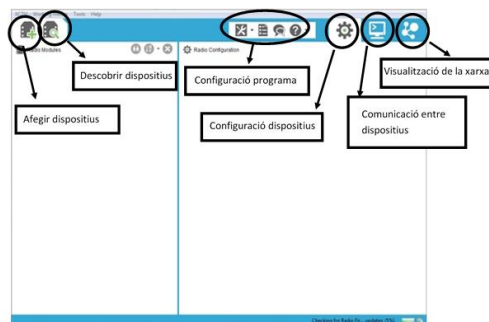


Figura 24. Interfície del software XCTU.

D'aquest últim grup, la primera serveix per modificar i veure la configuració de cadascun dels mòduls XBee, un cop s'han afegit a la xarxa.

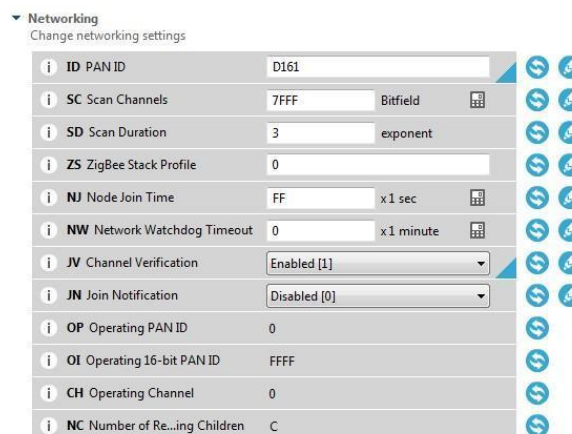


Figura 25. Mode configuració.

La segona per comunicar-se amb els diferents mòduls de la xarxa, mitjançant comandes API o AT.

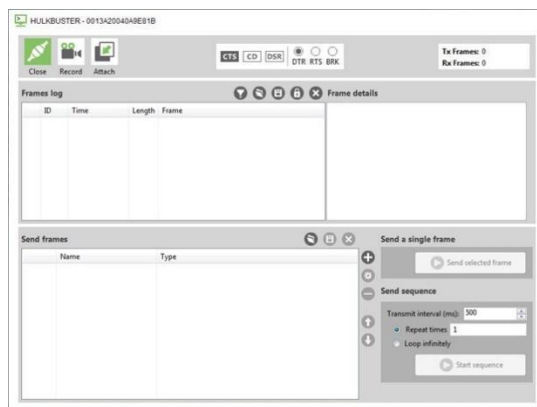


Figura 26. Mode comunicació.

La última per a visualitzar la xarxa amb la interconnexió dels diferents dispositius.

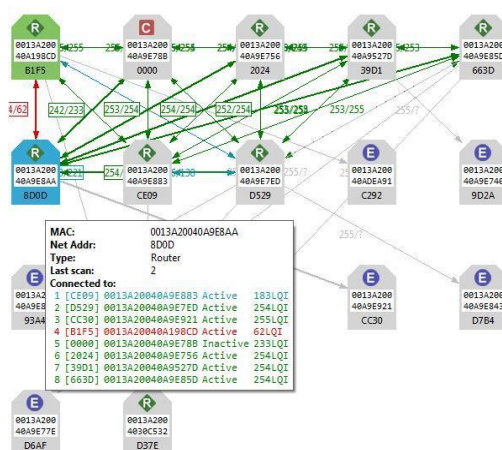


Figura 27. Mode xarxa.

3.2.3. PIC C Compiler

Per programar el codi que s'utilitzarà per a crear les funcionalitats que es necessiten per a l'aplicació, es fa servir PIC C Compiler.

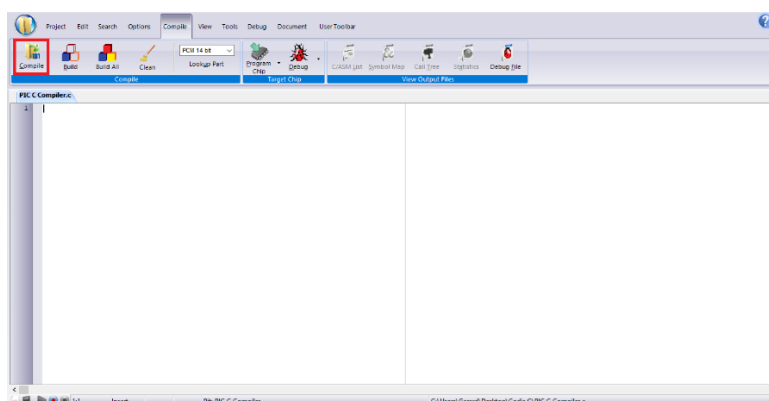


Figura 28. Entorn PIC C Compiler.

El programa permet crear l'arxiu on es troba el conjunt d'instruccions en llenguatge C. Aquest programa inclou un conjunt de llibreries i funcions predeterminades que facilita molt la feina al programador. Un cop creat l'arxiu es compila i el programa genera un informe amb els errors i advertències que té el codi. Per a funcionar correctament, es necessita que no tingui errors i que les advertències estiguin controlades. Un cop compilat es genera l'arxiu '.hex' que és el que es gravarà dins del microcontrolador.

3.2.4. Proteus 8

L'entorn de disseny Proteus 8 conté una gran quantitat de models de microcontroladors, concretament tota la gamma PIC de Microchip i molts altres elements que permeten dissenyar i simular els circuits.

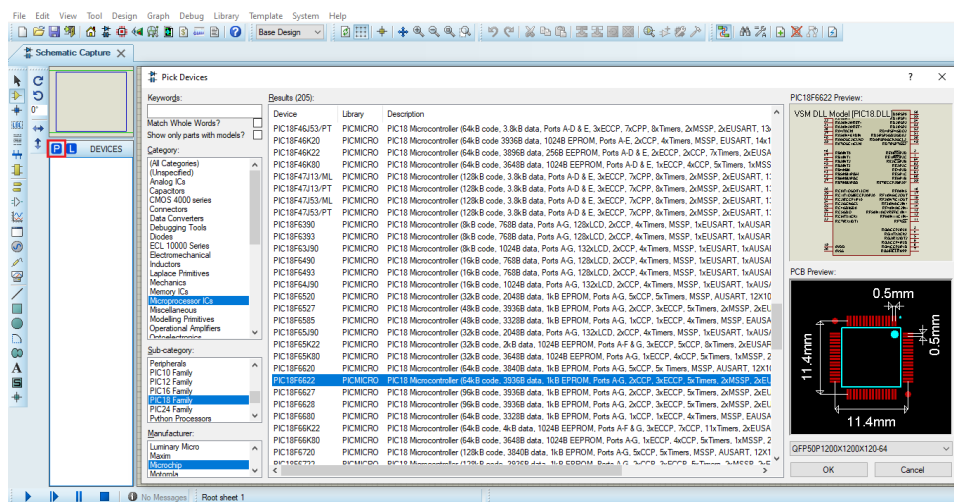


Figura 29. Selecció de MCU a Proteus 8.

El programa permet, un cop creat, copiar l'arxiu '.hex' generat després de compilar el codi C creat.

4. DESENVOLUPAMENT DEL PROJECTE

4.1.1a FASE: Disseny dels dispositius locals.

4.1.1. Dispositiu de temperatura.

4.1.1.1. Sensor de temperatura

Els sensor de temperatura més utilitzats en la indústria química són els termopars (tipus J i K) i les termoresistències (PT100 o PT1000). Els termopars s'utilitzen per a mesurar temperatures altes, fins a 1700°C, mentre que les termoresistències es fan servir per mesurar temperatures baixes o mitges, entre -200°C i 600°C.

Segons Wilka (2016), els termopars estan formats per dos filaments fets de diferents metalls i units en el punt de mesura. Quan aquest punt es calenta, és possible mesurar una tensió en els extrems dels filaments, proporcional a la temperatura en què es troba el punt de mesura. Aquest fenomen, anomenat efecte Seebeck, es produeix degut a la diferència de densitat d'electrons entre els dos materials i a la diferència de temperatura entre el punt de mesura i els extrems. Els instruments construïts mitjançant termopars gaudeixen de molta precisió, però s'hi ha d'incloure un circuit de compensació, degut a que l'extrem dels filaments es troben a temperatura ambient.

Segons García (2019), les termoresistències estan formades bàsicament per un element resistiu on la seva resistència canvia en funció de la temperatura en que es troba, cosa que el fa molt atractiu per la seva simplicitat i la seva precisió. Existeixen diferents classes de termoresistències que depenen de la mida de la resistència del dispositiu, però també es poden diferenciar segons la seva construcció interna, les bobinades i les de pel·lícula. La resistència està connectada mitjançant dos filaments que normalment estan fets de níquel, tungstè, coure o platí, essent aquest últim el més utilitzat. Tal i com fa referència a l'article d'Adelantado (2017), la classificació de les termoresistències, segueixen la normativa IEC / DIN 60751. Aquestes ofereixen una gran estabilitat i una alta linealitat, a més de una bona precisió segons el tipus que s'esculli. No obstant, aquest tipus de sensor necessita una font d'energia per tal de mesurar aquesta resistència variable i aquesta mesura pot ser difícil d'obtenir.

El dispositiu que s'està dissenyant s'utilitzarà per a mesurar la temperatura de l'oli d'un dipòsit amb un rang entre 20°C i 70°C, ja que aquesta és la temperatura

normal d'operació. No obstant, es preveu un rang de mesura entre 0 i 100 °C, per aquest motiu s'utilitzarà una termoresistència.

Els dos models que es plantegen són el PT100 i el PT1000. La principal diferència és que el primer té una resistència de 100 Ω a la temperatura de 0 °C, mentre que el segon és de 1000 Ω. Degut a aquesta diferència de resistència, el PT100 és una mica menys precís que el PT1000.

El PT1000, com que té una resistència més elevada, permet un consum més reduït que el PT100, i això interessa per tal de dissenyar un dispositiu més autònom.

Els PT100 són àmpliament utilitzats en la indústria i compatibles amb altres elements de instrumentació. S'acostumen a utilitzar en aplicacions amb rangs petits com el cas que ens ocupa, per aquest motiu s'utilitzarà aquest dispositiu.

4.1.1.2. Circuit d'adaptació

L'objectiu d'aquest circuit és mesurar la resistència mitjançant la variació de la caiguda de tensió en la termoresistència. El mètode utilitzat per aconseguir-ho és un pont de Wheatstone, ja que si s'utilitza un simple divisor de tensió, s'obté un error degut a la component constant que apareix als seus extrems.

Aquest circuit està format per dos branques simètriques amb quatre resistències iguals 2 a 2, quan la termoresistència està a 0 °C. Les dues resistències de la part superior han de ser suficientment grans per a reduir la intensitat de corrent del pont, però suficientment petites com per evitar que la tensió de sortida sigui massa petita per mesurar-la.

La tensió utilitzada per al pont de Wheatstone serà de 12 V. El circuit està en equilibri quan la termoresistència es troba a 0 °C que equivalen a 100 Ω. En aquest moment la sortida, és a dir, la caiguda de tensió entre les branques del pont és nul·la.

Segons la normativa IEC / DIN 60751 la resistència del PT100 es calcula mitjançant la següent expressió, amb un error <0.1 °C:

$$R_T = R_{0^\circ\text{C}} \cdot (1 + \alpha \cdot T) \quad 15$$

¹⁵ https://webstore.iec.ch/preview/info_iec60751%7Bed2.0%7Db.pdf

On α és el coeficient per a les PT100 amb un valor:

$$\alpha = 3.851 \cdot 10^{-3} \text{ } ^\circ\text{C}^{-1}$$

Així doncs, la resistència de la PT100 a $100 \text{ } ^\circ\text{C}$ és:

$$R_{100^\circ\text{C}} = 100 \cdot (1 + 3.851 \cdot 10^{-3} \cdot 100) = 138.51 \text{ } \Omega$$

La caiguda de tensió en la branca de la termoresistència a $100 \text{ } ^\circ\text{C}$ és:

$$I_1 = \frac{12 \text{ V}}{2338.5 \text{ } \Omega} = 5.13 \text{ mA}$$

$$V_1 = I_1 \cdot R_{PT100} = 5.13 \cdot 10^{-3} \cdot 138.5 = 710.5 \text{ mV}$$

En canvi, la caiguda en l'altre branca és:

$$I_2 = \frac{12 \text{ V}}{2300 \text{ } \Omega} = 5.22 \text{ mA}$$

$$V_2 = I_2 \cdot R_3 = 5.22 \cdot 10^{-3} \cdot 100 = 522 \text{ mV}$$

Així la diferència entre les dues branques és:

$$V = V_1 - V_2 = 710.5 - 522 = 188.5 \text{ mV}$$

S'observa el circuit amb un valor de $138.5 \text{ } \Omega$ a la PT100:

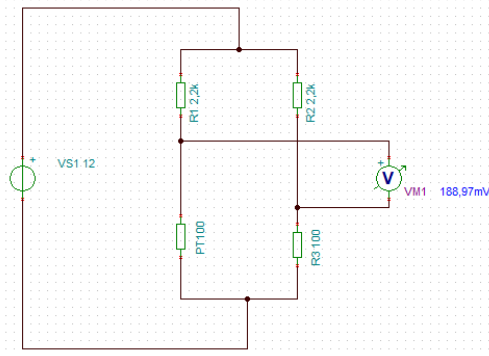


Figura 30. Pont de Wheatstone en desequilibri. Creació pròpia.

En aquest punt, s'obté una caiguda de tensió en la sortida directament proporcional al valor de la termoresistència. Es pot observar la evolució de la resistència en la PT100, dins del rang d'operació ($0 - 100 \text{ } ^\circ\text{C}$), en vers la tensió de sortida.

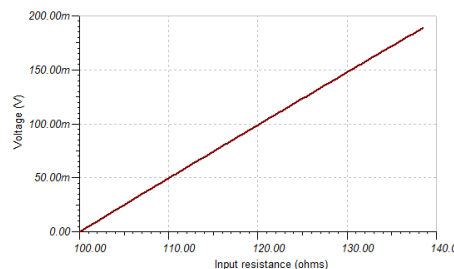


Figura 31. Relació PT100/Vout.

El valor de tensió a la sortida és massa petit per a ser utilitzat per el transmissor, per això s'utilitza un amplificador d'instrumentació en la següent etapa. Aquest tipus de configuració mitjançant amplificadors operacionals permet obtenir en la sortida, la diferència entre les seves entrades i amplificada per el guany necessari.

El dispositiu de transmissió requereix una tensió entre 0 i 3.3 V en la seva entrada, així que el guany estarà definit per la següent relació:

| T (°C) | V _{in} (mV) | V _{out} (V) |
|--------|----------------------|----------------------|
| 0 | 0 | 0 |
| 100 | 188.5 | 3.3 |

Taula 5. Rang de tensió de sortida. Creació pròpia.

Així doncs, el guany del amplificador ha de ser:

$$G = \frac{3.3}{0.1885} = 17.5$$

L'expressió que permet obtenir la tensió a la sortida d'un amplificador d'instrumentació és:

$$V_{out} = (V_2 - V_1) \cdot \left(1 + \frac{2 \cdot R_1}{R_g}\right) \cdot \left(\frac{R_3}{R_2}\right)$$

Essent el guany:

$$G = \left(1 + \frac{2 \cdot R_1}{R_g}\right) \cdot \left(\frac{R_3}{R_2}\right)$$

Amb les resistències disponibles s'obté el següent guany:

$$V_{out} = (V_2 - V_1) \cdot \left(1 + \frac{2 \cdot 2200}{1000}\right) \cdot \left(\frac{56000}{18300}\right) = (V_2 - V_1) \cdot 16.5$$

S'utilitza l'amplificador operacional uA471 de Texas Instruments àmpliament conegut, econòmic, molt utilitzat i alimentat amb ± 15 V.

Així que quan la temperatura estigui a 100 °C, la sortida cap al transmissor és:

$$V_{out} = 188.5 \cdot 10^{-3} \cdot 16.5 = 3.11 \text{ V}$$

S'observa en la imatge el circuit dissenyat i la simulació de la sortida quan la termoresistència PT100 està a 100 °C.

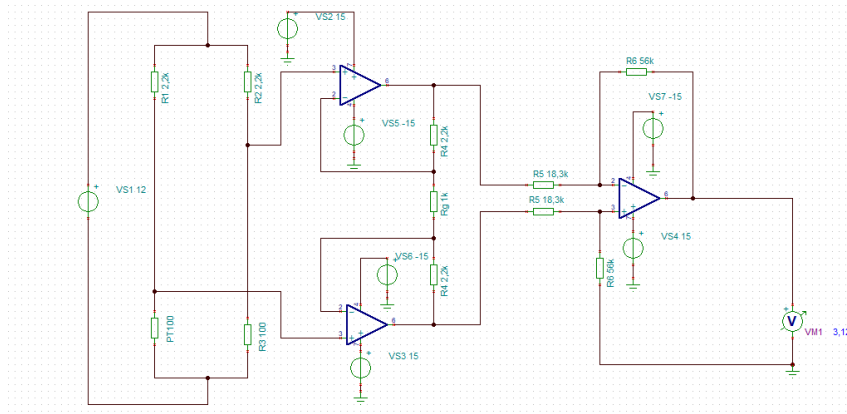


Figura 32. Connexió pont de Wheatstone i amplificador d'instrumentació. Creació pròpia.

Es pot comprovar amb la relació de tensió sortida vers la resistència PT100, com la sortida del sensor de temperatura és totalment lineal:

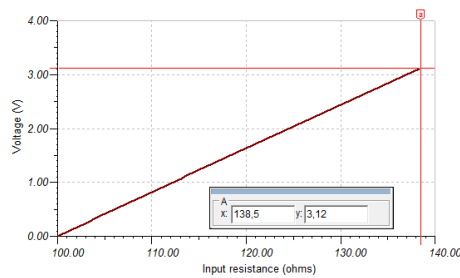


Figura 33. Relació $V_{out}/PT100$ un cop adaptat el circuit. Creació pròpia.

4.1.1.3. Connexió i configuració del dispositiu Xbee

El dispositiu local de temperatura queda:

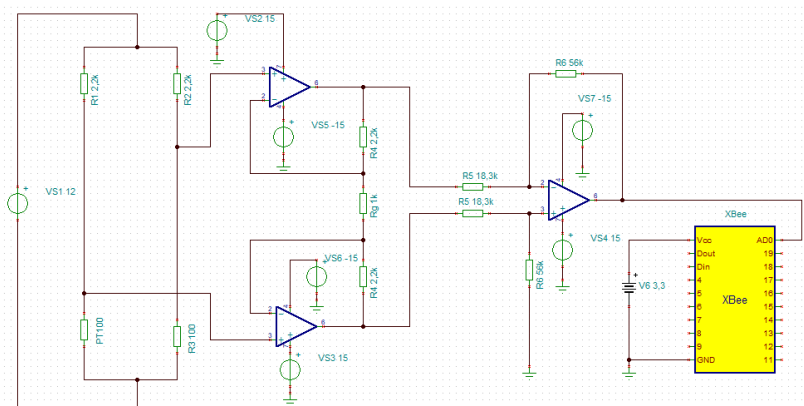


Figura 34. Connexió del sensor de temperatura al XBee.

Els principals paràmetres del mòdul es configuren de la següent manera:

- PANID: 2020
- Coordinator: disable
- Destination High: 0
- Destination Low: 0
- API: Transparent Mode
- Sleep Mode: enable
- Cycle sleep period: valor per defecte. S'ajustarà un cop dissenyat tot el sistema.

4.1.1.4. Disseny de la font d'alimentació

Les tensions i corrents necessàries en aquest dispositiu són:

- 12 V: per a la termoresistència PT100.

Si es calcula la resistència total a 100°C del pont de Wheatstone:

$$R_T = \frac{(R_1 + R_3) \cdot (R_2 + R_{PT100})}{R_1 + R_2 + R_3 + R_{PT100}} = \frac{2300 \cdot 2338}{4600} = 1159 \Omega$$

La intensitat màxima dins del rang de la termoresistència que pot necessitar és:

$$I_{max} = \frac{V}{R_T} = \frac{12}{1159} = 10.35 \text{ mA}$$

- 3.3 V: per el mòdul XBee.

Segons el datasheet del mòdul RF de Xbee, l'amperatge màxim del mòdul és 33 mA en la transmissió.

- 15 V: per a l'alimentació dels AO's.

Segons el datasheet del amplificador uA741, l'amperatge de funcionament és 1.7 mA.

La font d'alimentació ha d'oferir un mínim de 15 V i la suma de tots els consums de corrent:

$$I_T = 10.35 + 33 + (3 \cdot 1.7) = 48.45 \text{ mA}$$

En el mercat existeixen multitud de transformadors de 220Vac a una tensió de 15 Vdc que és la que es necessita en aquest dispositiu. En un distribuïdor de confiança s'ha trobat el model VSK-S3-15U que s'ajusta perfectament a les necessitats exigides.



Figura 35. Transformador 220Vac-15Vdc¹⁶



| MODEL | output voltage | output current | output power | ripple and noise ¹ | efficiency |
|-------------|----------------|----------------|--------------|-------------------------------|------------|
| | (Vdc) | max (A) | max (W) | typ (mVp-p) | typ (%) |
| VSK-S3-3R3U | 3.3 | 0.7 | 2.3 | 30 | 63 |
| VSK-S3-5U | 5 | 0.6 | 3 | 30 | 72 |
| VSK-S3-9U | 9 | 0.33 | 3 | 30 | 74 |
| VSK-S3-12U | 12 | 0.25 | 3 | 30 | 76 |
| VSK-S3-15U | 15 | 0.2 | 3 | 30 | 76 |
| VSK-S3-24U | 24 | 0.125 | 3 | 30 | 78 |

Notes: 1. Ripple and noise measured at 20 MHz bandwidth

Figura 36. Especificacions VSK-S3-15U.¹⁶

El model ve encapsulat i llest per a la inserció a una PCB. S'escull el model de 15 V i 200 mA, ja que és el més pròxim a les necessitats de voltatge i l'amperatge que ofereix és suficientment gran.

El fabricant recomana utilitzar alguns components addicionals per protegir els circuits que es connectin al transformador. S'observa la simulació de la connexió del transformador:

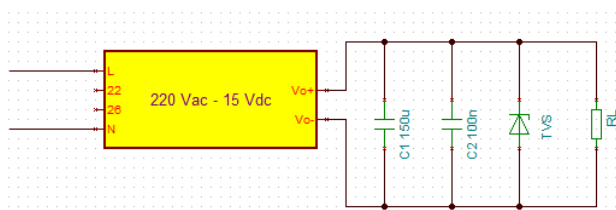


Figura 37. Connexió recomanada pel fabricant.¹⁶

Mitjançant el disseny d'un divisor de tensió amb tres resistències s'aconseguiran les diferents tensions necessàries per al circuit. Aplicant l'expressió :

$$V_{R_n} = V_{in} \cdot \frac{R_n}{\sum_{i=1}^3 R_i}$$

Les diferents resistències corresponen a:

$$3 = 15 \cdot \frac{R_1}{R_1 + R_2 + R_3}$$

¹⁶ https://www.mouser.es/datasheet/2/670/vsk_s3-1313305.pdf

$$12 = 15 \cdot \frac{R_2 + R_3}{R_1 + R_2 + R_3}$$

$$3.3 = 15 \cdot \frac{R_3}{R_1 + R_2 + R_3}$$

Solucionant el sistema, s'obté:

$$R_1 = 0.909 \cdot R_3 \quad ; \quad R_2 = 2.64 \cdot R_3$$

Forçant una valor comercial a la resistència R_3 , s'aconsegueix el divisor amb les tensions necessàries:

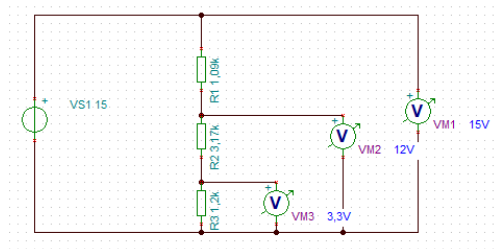


Figura 38. Divisor de tensió. Creació pròpia.

4.1.1.5. Implementació en protoboard.

S'implementa el circuit en una protoboard el disseny per al sensor de temperatura i se'n comprova el funcionament mitjançant la temperatura ambient, aigua freda (amb gel) i aigua calenta.

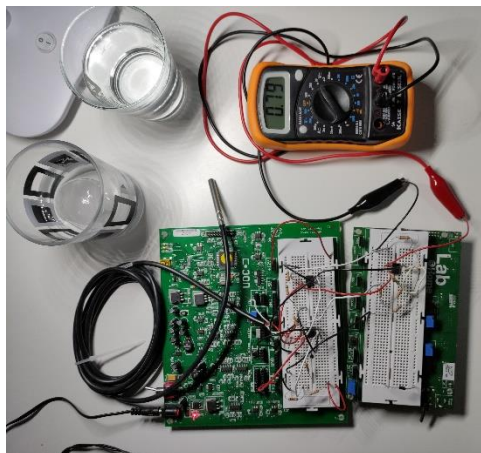


Figura 39. Temperatura ambient (24 °C).



Figura 40. Aigua amb gel.

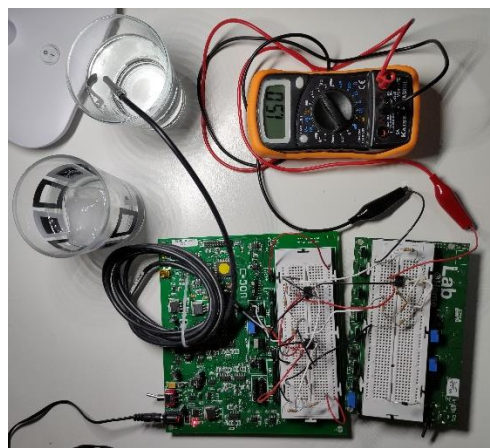


Figura 41. Aigua calenta

Les temperatures equivalents utilitzant els voltatges són:

$$T_{ambient} = \frac{0.79 V}{3.3 V} \cdot 100 \text{ }^{\circ}\text{C} = 23.94 \text{ }^{\circ}\text{C}$$

$$T_{freda} = \frac{0.33 V}{3.3 V} \cdot 100 \text{ }^{\circ}\text{C} = 10 \text{ }^{\circ}\text{C}$$

$$T_{calenta} = \frac{1.5 V}{3.3 V} \cdot 100 \text{ }^{\circ}\text{C} = 45.4 \text{ }^{\circ}\text{C}$$

Un cop comprovat el bon funcionament, el circuit s'implementa sobre una placa de baquelita, i els circuits es formen mitjançant soldadura d'estany. Per tal de dissenyar la forma dels circuits sobre la placa, es té en compte la mida de la placa i dels elements. S'ha de tenir en compte que els camins dels circuits no s'han de creuar mai, i per tal de poder soldar millor, és preferible que estiguin com més distanciats millor. La millor opció segueix essent la implementació dels circuits en PCB's, però degut al temps i el cost econòmic que implica enviar a imprimir en empreses dedicades a això s'ha decidit utilitzar l'opció de la

soldadura. També és possible, mitjançant productes químics, crear PCB's casolanes encara que l'acabat no és el mateix.

En la construcció del prototip també s'ha obviat la compra dels transformadors i la implementació dels circuits associats a les fonts d'alimentació. S'utilitzen les plaques de desenvolupament de la UOC que s'han utilitzat en assignatures anteriors.

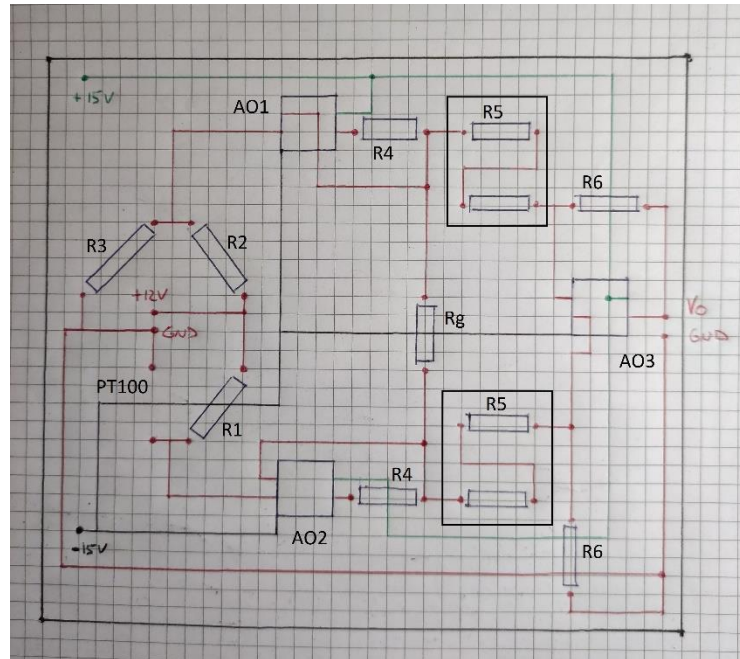


Figura 42. Disseny del circuit a soldar.

Un cop soldat el circuit, s'obté:

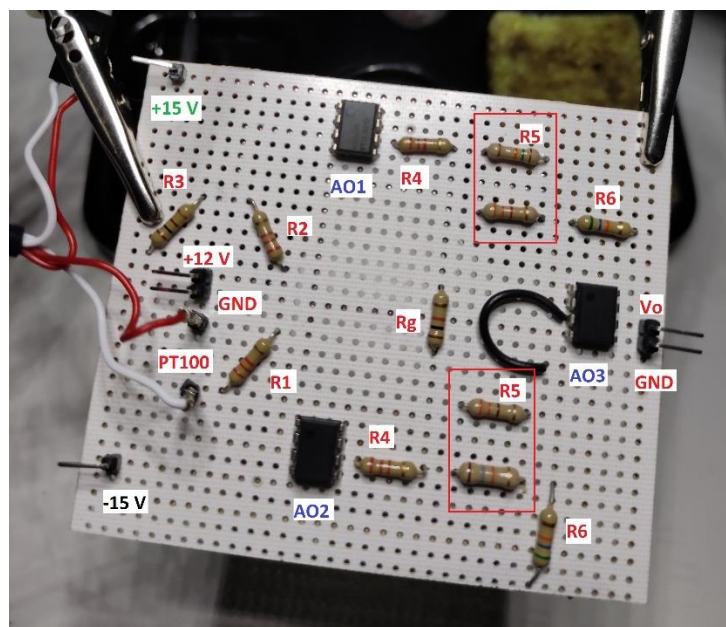


Figura 43a. Placa soldada (part superior).

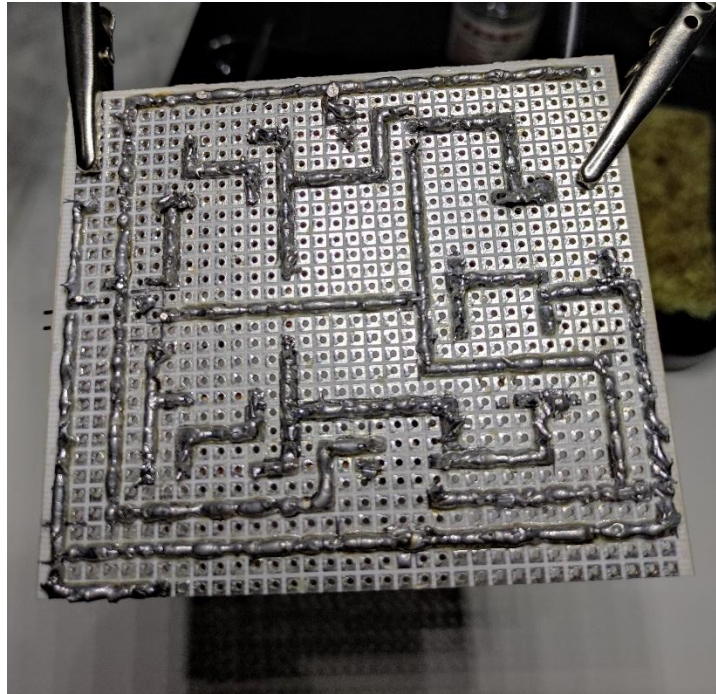


Figura 43b. Placa soldada (part inferior).

4.1.2. Dispositiu de nivell

4.1.2.1. Sensor de nivell

El sensor de nivell escollit permet obtenir una sortida relacionada amb el nivell de líquid que detecta. La detecció es produeix mitjançant la conductivitat del líquid que actua com una resistència molt gran quan no hi ha nivell i que es va reduint a mesura que augmenta el nivell i permeten que circuli corrent. Aquest corrent alimenta la base d'un transistor que amplifica el corrent entre el emissor i el col·lector del transistor. Mitjançant una càrrega de sortida permet mesurar la tensió a la sortida del dispositiu en funció de la quantitat de corrent que permet circular el líquid. Amb més nivell de líquid detectat, més corrent circula i més tensió es genera a la sortida.

S'ha escollit aquest tipus de sensor perquè, a més de detectar si hi ha nivell de líquid o no, permet detectar un cert rang i poder configurar diferents estats de nivell.

El model escollit permet una tensió d'alimentació entre 2 i 5 V i una profunditat detectable de 4.8 cm mitjançant una sortida analògica que permet connectar directament amb un convertidor analògic-digital.

Segons el fabricant, la relació nivell i tensió de sortida és:

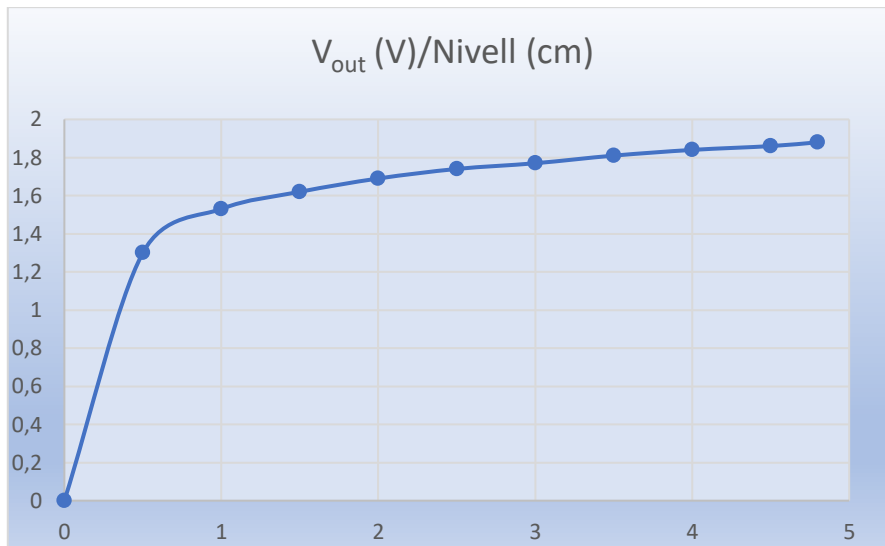


Figura 44. Relació Profunditat/Vout. Creació pròpia.

Una representació del sensor de nivell podria ser aquesta:

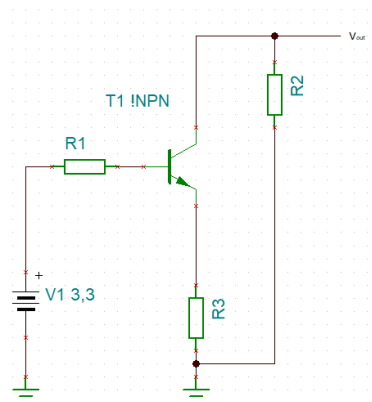


Figura 45. Model equivalent del sensor de nivell. Creació pròpia.

Una de els avantatges que té aquest sensor és que no necessita circuit d'adaptació.

4.1.2.2. Connexió i configuració del dispositiu Xbee

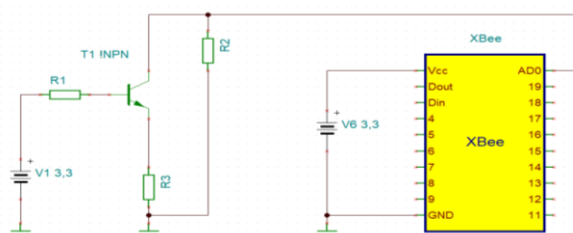


Figura 46. Connexió del sensor de nivell al Xbee. Creació pròpia.

Els principals paràmetres del mòdul es configuren de la següent manera:

- PANID: 2020
- Coordinator: disable
- Destination High: 0
- Destination Low: 0
- API: Transparent Mode
- Sleep Mode: enable
- Cycle sleep period: valor per defecte. S'ajustarà un cop dissenyat tot el sistema.


4.1.2.3. Disseny de la font d'alimentació

Les tensions necessàries en aquest dispositiu són:

- 3.3 V: per al mòdul RF XBee.
- 3.3 V: per al sensor de nivell.

El corrent màxim del XBee, com s'ha vist en el dispositiu de temperatura, és de 33 mA. Per al sensor de nivell, no apareix el consum de corrent en les especificacions, per tant, s'escollirà un transformador amb marge suficient.

En aquest cas s'ha escollit un transformador de la mateixa sèrie que el dispositiu de temperatura. El model VSK-S3-12U proporciona 12 V i un corrent màxim de 250 mA, suficient per a les necessitats del circuit a alimentar.



| MODEL | output voltage (Vdc) | output current max (A) | output power max (W) | ripple and noise ¹ typ (mVp-p) | efficiency typ (%) |
|-------------|-------------------------|------------------------------|----------------------------|---|--------------------------|
| VSK-S3-3R3U | 3.3 | 0.7 | 2.3 | 30 | 63 |
| VSK-S3-5U | 5 | 0.6 | 3 | 30 | 72 |
| VSK-S3-9U | 9 | 0.33 | 3 | 30 | 74 |
| VSK-S3-12U | 12 | 0.25 | 3 | 30 | 76 |
| VSK-S3-15U | 15 | 0.2 | 3 | 30 | 76 |
| VSK-S3-24U | 24 | 0.125 | 3 | 30 | 78 |

Notes: 1. Ripple and noise measured at 20 MHz bandwidth

Figura 47. Especificacions transformador 220Vac-12Vdc.¹⁷

La connexió recomanada per el fabricant és:

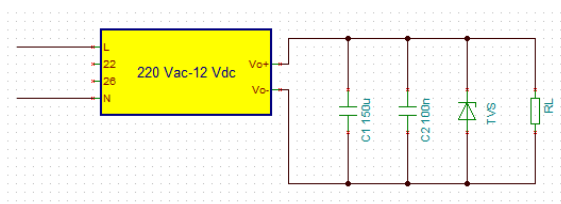


Figura 48. Connexió recomanada del transformador.¹⁷

¹⁷ https://www.mouser.es/datasheet/2/670/vsk_s3-1313305.pdf

Un cop obtinguts els 12 V de tensió contínua, es procedeix a dissenyar el divisor de tensió que permetrà aconseguir els voltatges dels components del dispositiu.

$$3.3 = 12 \cdot \frac{R_2}{R_1 + R_2}$$

Donant un valor comercial de resistència a R_1 , s'obté el circuit:

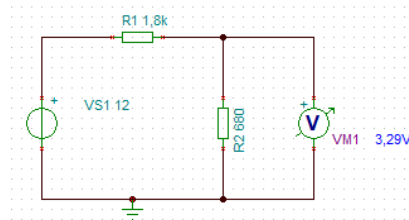


Figura 49. Divisor de tensió. Creació pròpia.

4.1.2.4. Implementació en protoboard.

S'implementa el circuit i se'n comprova el funcionament.

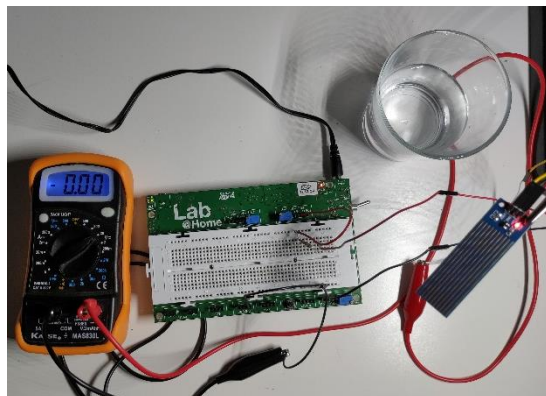


Figura 50. Nivell nul.

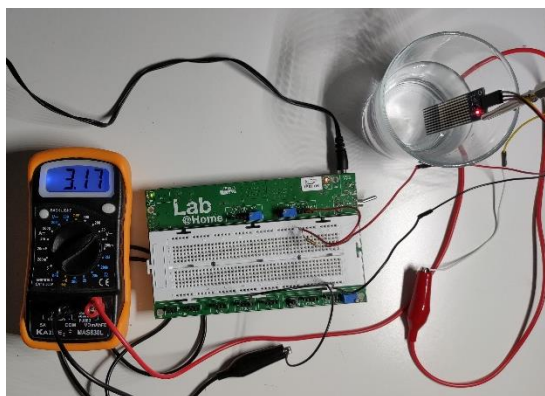


Figura 51. Nivell mig.

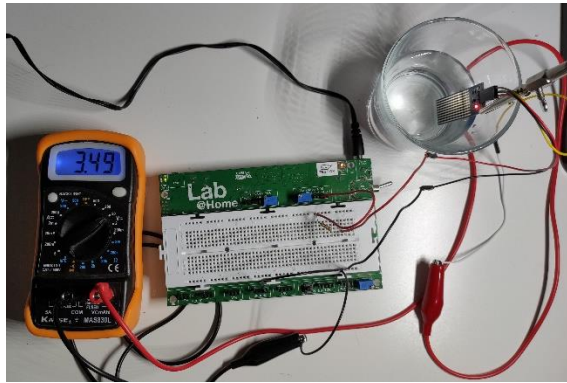


Figura 52. Nivell alt

Si es comparen les mesures obtingudes amb la taula 4 del apartat 3.1.3, es pot observar que la sortida analògica obtinguda no és molt coherent amb el que ens diu el seu full d'especificacions. Aquest fet demostra la poca precisió del sensor escollit.

4.2. 2a FASE: Disseny del dispositiu central

Per a l'aplicació que s'està dissenyant, és necessari un dispositiu que integri les funcions necessàries per a realitzar les tasques que s'esperen del sistema. Els sistemes integrats poden ser de propòsit general o de propòsit específic amb arquitectures idèntiques, és en la selecció dels diferents elements on es diferencien l'un de l'altre.

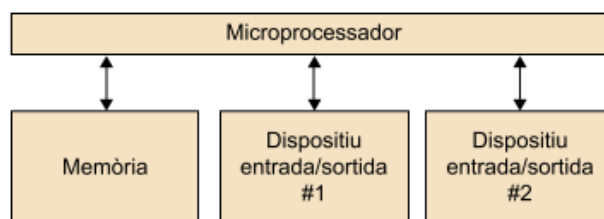


Figura 53. Estructura d'un sistema integrat.¹⁸

¹⁸ Mòdul 5. Electrònica Digital. UOC

Com s'observa en la imatge anterior, el dispositiu central estarà format per les següents parts, el microprocessador, la memòria i els perifèrics. Seguidament, s'anomenaran i es comentaran les principals característiques que permeten escollir el tipus dispositiu integrat:

- Tipus de processador.

En la categoria de microprocessadors, els sistemes de propòsit general són els més potents i utilitzen les architectures CISC, RISC, SISC, etc.

Per als de propòsit específic s'utilitzen architectures ARM, però també es poden trobar architectures RISC i SISC.

Una altre categoria de processadors són els DSP (Digital Signal Processor) que s'utilitzen en alguns sistemes específics que requereixen una gran capacitat de computació, ja que permeten realitzar operacions aritmètiques molt complexes.

Els microcontroladors (MCU) són un altre tipus de processador utilitzat àmpliament en els sistemes específics, amb architectures des de 8 fins a 32 bits.

Per últim, es troben les FPGA (Field Programmable Gate Array) que són un conjunt de portes lògiques que es poden programar formant un circuit lògic per a que realitzi una funció específica.

- Memòria.

Existeixen dos tipus d'arquitectures de memòria per a aquests dispositius. L'arquitectura Von Neumann (figura 50, esquerra) emmagatzema totes les dades i el codi de programa en una mateixa memòria i només necessita una connexió amb el processador. En canvi, l'arquitectura Harvard (figura 50, dreta) guarda el codi del programa separat de les dades, però necessita dos connexions amb el processador.

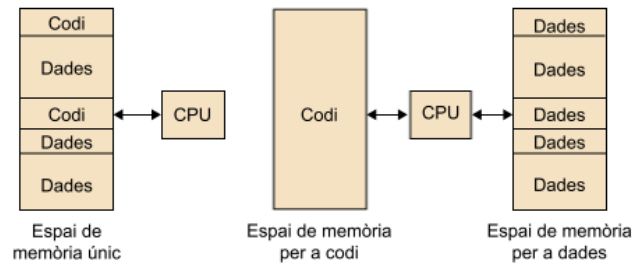


Figura 54. Arquitectures de memòries.¹⁹

També es poden distingir dos tipus de memòries:

- ROM: originalment és un tipus de memòria només de lectura que s'utilitza per a guardar el programa d'instruccions i les dades constants. En l'actualitat, existeixen un conjunt de memòries ROM que permeten un nombre limitat de reescriptures. Les EPROM permeten ser esborrades però només amb llum ultraviolada. Les EEPROM poden ser programades y esborrades mitjançant enregistrator y un PC. Les memòries FLASH són molt semblants a les EEPROM però amb més prestacions.
- RAM: és un tipus de memòria volàtil, ja que s'esborren si són desconnectades de la font d'alimentació. Aquest tipus permet la reescriptura i s'utilitza per a emmagatzemar les variables del programa de control i les dades que s'utilitzen en l'execució.
- Pins E/S: són els pins on es connecten els perifèrics. S'utilitzen per interaccionar amb el microcontrolador, ja sigui a través de comunicacions, mitjançant pantalles i teclats, o sensors i actuadors. Aquests pins es poden configurar com a entrades o sortides i també com a digitals o analògiques mitjançant convertidors A/D.

Un sistema integrat necessita utilitzar altres elements per tal de funcionar correctament. S'anomenen i es comenten els més importants:

- Rellotge: es tracta de una ona quadrada d'alta freqüència que permet mantenir el temps d'execució del programa de control del microcontrolador. Es poden utilitzar components externs per tal de fer més precís aquest temps d'execució.
- Ports de comunicació: són pins del microcontrolador que permeten comunicar-se amb dispositius externs, com ara mòduls RF, PC's, altres

¹⁹ Mòdul 5. Electrònica Digital. UOC

microcontroladors, etc. Les comunicacions es poden dur a terme mitjançant diferents tipus de protocols, com ara UART, USART, SPI, I²C, etc.

- Llenguatge de programació: normalment tots els microcontroladors poden ser programats mitjançant el codi universal C, però també es pot utilitzar el llenguatge màquina en alguns tipus de dispositius. El llenguatge C permet fer més senzilla la creació d'ordres complexos per a fer funcionar el microcontrolador sense haver de programar amb el llenguatge màquina, que és més difícil de comprendre per els humans.

4.2.1. Microcontrolador

El dispositiu escollit per al control del sistema és un microcontrolador de 8 bits del fabricant Microchip, concretament el model PIC18F6622. Les seves característiques són:

| Device | Program Memory | | Data Memory | | I/O | 10-Bit A/D (ch) | CCP/ ECCP (PWM) | MSSP | | EUSART | Comparators | Timers 8/16-Bit | External Bus | |
|------------|----------------|----------------------------|--------------|----------------|-----|-----------------|-----------------|------|--------------------------|--------|-------------|-----------------|--------------|---|
| | Flash (bytes) | # Single-Word Instructions | SRAM (bytes) | EEPROM (bytes) | | | | SPI | Master I ² C™ | | | | | |
| PIC18F6527 | 48K | 24576 | 3936 | 1024 | 54 | 12 | 2/3 | 2 | Y | Y | 2 | 2 | 2/3 | N |
| PIC18F6622 | 64K | 32768 | 3936 | 1024 | 54 | 12 | 2/3 | 2 | Y | Y | 2 | 2 | 2/3 | N |
| PIC18F6627 | 96K | 49152 | 3936 | 1024 | 54 | 12 | 2/3 | 2 | Y | Y | 2 | 2 | 2/3 | N |
| PIC18F6722 | 128K | 65536 | 3936 | 1024 | 54 | 12 | 2/3 | 2 | Y | Y | 2 | 2 | 2/3 | N |
| PIC18F8527 | 48K | 24576 | 3936 | 1024 | 70 | 16 | 2/3 | 2 | Y | Y | 2 | 2 | 2/3 | Y |
| PIC18F8622 | 64K | 32768 | 3936 | 1024 | 70 | 16 | 2/3 | 2 | Y | Y | 2 | 2 | 2/3 | Y |
| PIC18F8627 | 96K | 49152 | 3936 | 1024 | 70 | 16 | 2/3 | 2 | Y | Y | 2 | 2 | 2/3 | Y |
| PIC18F8722 | 128K | 65536 | 3936 | 1024 | 70 | 16 | 2/3 | 2 | Y | Y | 2 | 2 | 2/3 | Y |

Figura 55. Característiques família PIC18F6XXX.²⁰

Aquest model s'ha escollit per la seva memòria EEPROM de 1024 bytes, que permetrà emmagatzemar una gran quantitat de mostres rebudes des dels sensors. Com s'ha vist en el dispositiu XBee, les mostres analògiques són de 10 bits, això equival a 2 bytes per mostra. Tenint en compte la capacitat de la memòria i que el sistema està format per dos sensors, les mostres totals que es poden emmagatzemar són:

$$N_{Totals} = \frac{1024 \text{ bytes}}{2 \text{ bytes/mostra}} = 512 \text{ mostres}$$

$$N_{temp} = N_{Niv} = \frac{512 \text{ mostres}}{2 \text{ sensors}} = 256 \text{ mostres/sensor}$$

²⁰ <https://www.mouser.es/datasheet/2/268/39646c-17052.pdf>

Com es pot veure en el codi, també s'ha tingut en compte guardar 2 bytes per cada sensor amb la finalitat de emmagatzemar la posició de la memòria. D'aquesta manera, s'implementa una memòria rotativa i s'utilitzen totes les posicions de la memòria, allargant-ne la seva vida útil.

El microcontrolador inclou un oscil·lador intern fins a 32 MHz, que es pot ampliar fins a 40 MHz amb la instal·lació d'un cristall extern. Aquest model de PIC és ideal, ja que té la possibilitat de connectar dos ports serials mitjançant UART, un per al mòdul XBee i l'altre per un mòdul Bluetooth o un altre microcontrolador. El nombre de pins E/S no és rellevant per al projecte, així que s'han agafat el mínim per tal de reduir costos.

L'alimentació del microcontrolador pot ser entre 2 i 5.5 V, per tant, s'alimentarà a 3.3 com el mòdul XBee i d'aquesta manera no caldrà adaptar les comunicacions serials entre ells.

Transmissió Serial UART.

En les transmissions es necessita una sincronització entre el transmissor i el receptor, ja sigui en la pròpia comunicació o en els propis dispositius. Per tal de comunicar el mòdul XBee i el microcontrolador s'utilitzarà una connexió serial. En el cas de la transmissió serial UART (Universal Asincron Receptor Transmissor), la sincronització s'efectua en els propis microcontroladors i mitjançant una marca d'inici i una altre de final.

La trama d'aquest protocol de transmissió és:

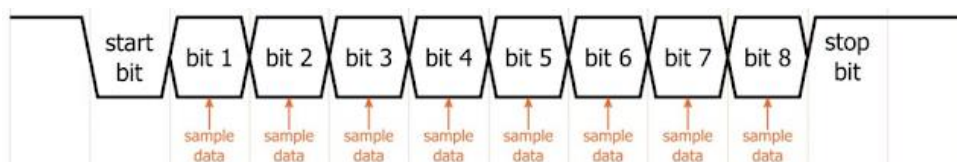


Figura 56. Trama UART.²¹

Com s'observa, la trama comença sempre amb un 0 lògic i amb una duració de 1 bit, i s'acaba amb un 1 lògic. Entre aquests dos bits, s'envien els bits d'informació a transmetre. Els bits d'informació poden ser 8 o 9, segons si s'utilitza un bit de paritat per detectar errors en la recepció. El protocol pot utilitzar

²¹ <http://programaciondeavr.blogspot.com/2018/12/usart-o-uart.html>

diferents estàndards de comunicació serial com a medi físic i que es diferencien per els voltatges dels '1' i '0' lògics que utilitzen. Aquests estàndards són RS-232, RS-422 i RS-485, essent el primer de tots el més utilitzat. En el dispositiu no s'utilitza cap d'aquests ja que els dos dispositius tenen la mateixa alimentació i la connexió és directe.

El sincronisme s'aconsegueix mitjançant els bits d'inici i final i el temps de bit, que ha de coincidir en els dos dispositius que s'estan comunicant. Aquest temps es calcula en bauds mitjançant la següent expressió:

$$R_b = \frac{1}{T_{bit}}$$

Les velocitats més utilitzades van entre 4800 bauds i els 57600 bauds, encara que poden ser inferiors o superiors segons els dispositius utilitzats. Per defecte, qualsevol dispositiu està configurat a 9600 bauds, que equival a un temps de bit de:

$$T_{bit} = \frac{1}{R_b} = \frac{1}{9600} = 0.1 \text{ ms}$$

La comunicació mitjançant aquest protocol pot ser simplex si només transmet un dels dos dispositius, full-duplex quan els dos dispositius envien i reben dades al mateix moment o half-duplex si els dispositius esperen a transmetre un cop acabi l'altre. Això és possible, ja que utilitzen línies diferents per a transmetre i rebre.

Segons Suárez, et al., (2009), en l'article "Interfaz programable de comunicació serie asincrónica en FPGA", els avantatges d'aquest protocol són:

- Simplicitat per implementar una programació.
- Degut al asincronisme utilitzat, és més ràpid verificar les dades.
- No necessita molta precisió de temps de bit per a funcionar.

Els inconvenients són:

- És un protocol molt ineficient respecte a la capacitat del canal.
- La velocitat de transmissió és baixa.

4.2.2. Pantalla LCD

La pantalla LCD permetrà visualitzar els menús de l'aplicació i mostrar els valors dels sensors remots, tant en temps real com emmagatzemats en memòria.

L'alimentació de la pantalla és de 5V amb un corrent típic de 1.8mA. Aquest model permet la visualització de 2 files de 16 caràcters cadascuna.

S'ha considerat aquesta pantalla per tal de reduir costs en la fabricació del prototip. En un model final es podria incloure una pantalla de 4 files per 20 caràcters cadascuna, que facilitaria la lectura per part de l'usuari.

4.2.3. Polsadors

La navegació a través dels menús de l'aplicació es realitzarà mitjançant 4 polsadors connectats als pins RA0, RA1, RA2 i RA3 del PIC. Les funcions de cadascun seran de pujar, baixar, entrar i sortir. La relació de funcions amb els pins és:

| Funció | Pin |
|--------|-----|
| Pujar | RA1 |
| Baixar | RA0 |
| Entrar | RA2 |
| Sortir | RA3 |

Taula 6. Funcions associades als pins del PIC.

4.2.4. Mòdul XBee

El mòdul XBee del dispositiu central es configurarà de la mateixa manera que els dispositius locals. L'única diferència roman en la configuració en mode API, que permetrà enviar ordres als sensors i aquests ens retornaran els valors mesurats. Els dispositius XBee configurats en mode API no poden dormir per tal de reduir el seu consum, sempre estan escoltant la xarxa per si algun dispositiu End Device envia alguna dada.

Les comandes AT són ordres que els mòduls XBee interpreten per a modificar paràmetres de configuració, canviar els nivells de les sortides digitals o forçar l'enviament de l'estat dels pins d'entrada, entre molts altres. Aquest últim interessa en especial, ja que permet demanar el valor actual que està mesurant el sensor remot.

Per poder comunicar amb aquest dispositius, el microcontrolador envia mitjançant la comunicació serial UART, una trama específica al mòdul RF i que permet que el dispositiu remot ens contesti.

La trama API enviada per a la sol·licitud de la comanda AT és la següent:

| Byte | Valor | Descripció | |
|------|-------|--|---|
| 0 | 0x7E | Byte delimitador | |
| 1 | 0x00 | Llargària del missatge (número de bytes des del byte 3 al byte 18) | |
| 2 | 0x10 | | |
| 3 | 0x17 | Sol·licitud de comanda AT | |
| 4 | 0x01 | Número seqüència del paquet | |
| 5 | 0x01 | Direcció de 64 bits del destinatari (equival al número de sèrie del dispositiu) Es configura per a cadascun dels dispositius. | |
| 6 | 0x13 | | |
| 7 | 0xA2 | | |
| 8 | 0x00 | | |
| 9 | 0x40 | | |
| 10 | 0x8B | | |
| 11 | 0x78 | | |
| 12 | 0x4E | | |
| 13 | 0xFF | | Direcció de la xarxa de destí. Sempre es configura com a 0xFFFFE |
| 14 | 0xFE | | |
| 15 | 0x00 | | Opció de la comanda AT, en aquest cas cap. |
| 16 | 0x49 | | Comanda AT per a forçar una lectura dels pins del XBee remot. Correspon a la comanda 'IS' en hexadecimal. |
| 17 | 0x53 | | |
| 18 | 0x00 | Valor paràmetre de la comanada AT. | |
| 19 | 0x08 | Checksum. | |

Taula 7. Trama API de sol·licitud de comanda AT.

Un cop el dispositiu remot ha rebut el missatge, enviarà la resposta dins d'una trama específica amb el valor mesurat al XBee receptor. Aquest l'enviarà mitjançant la comunicació UART al microcontrolador.

La trama AT de resposta és la següent:

| Byte | Valor | Descripció | |
|------|-------|--|--|
| 0 | 0x7E | Byte delimitador | |
| 1 | 0x00 | Llargària del missatge (des del byte 3 al byte 18) | |
| 2 | 0x11 | | |
| 3 | 0x97 | Resposta de comanda AT | |
| 4 | 0x01 | Número seqüència del paquet | |
| 5 | 0x01 | Direcció de 64 bits del emissor (equival al número de sèrie del dispositiu) Es configurarà per a cadascun dels dispositius. | |
| 6 | 0x13 | | |
| 7 | 0xA2 | | |
| 8 | 0x00 | | |
| 9 | 0x40 | | |
| 10 | 0x8B | | |
| 11 | 0x78 | | |
| 12 | 0x4E | | |
| 13 | 0xFF | | Direcció de la xarxa de destí. Sempre es configura com a 0xFFFFE |
| 14 | 0xFE | | |
| 15 | 0x49 | Comanda AT per a forçar una lectura dels pins del XBee remot. Correspon a la comanda 'IS' en hexadecimal. | |
| 16 | 0x53 | | |
| 17 | 0x00 | Codi d'error. | |
| 18 | 0x01 | Valor del sensor. | |
| 19 | 0xD1 | | |
| 20 | 0x33 | Checksum. | |

Taula 8. Trama API de resposta de comanda AT.

El checksum és una suma de comprovació que permet identificar errors en la trama ràpidament. Si un mòdul Xbee rep una trama amb un error en el checksum, aquesta serà eliminada. La manera de calcular-la és sumant els valor de cadascun dels bytes de la trama, exceptuant el delimitador i la llargària. Posteriorment s'agafa el byte de menys pes i es resta a el byte FF.

S'observa el càlcul per a les trames que s'utilitzaran:

- Per a la sol·licitud:

$$17 + 01 + 13 + A2 + 40 + 8B + 78 + 4E + FF + FE + 49 + 53 \\ = 4F7$$

$$FF - F7 = 08$$

- Per a la resposta:

$$97 + 01 + 13 + A2 + 12 + 34 + 56 + 78 + FF + FE + 49 + 53 + 01 \\ + D1 = 5CC$$

$$FF - CC = 33$$

Per comprovar si es correcta una trama API rebuda, només cal sumar tots els bytes de la trama, exceptuant el delimitador i la llargària però incloent-hi el checksum. El byte de menys pes resultant sempre ha de ser FF.

Seguidament s'apliquen els paràmetres esmentats en aquest apartat per al dispositiu central i en els apartats 4.1.1.3 i 4.1.2.2 per al transmissor de temperatura i nivell. Un cop aplicat en els respectius mòduls XBee, l'aspecte de la xarxa mitjançant el programa XCTU és:

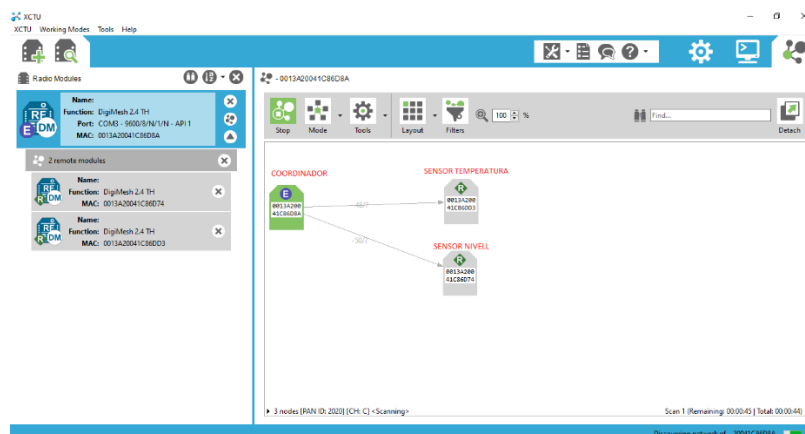


Figura 57. Captura de la xarxa amb XCTU.

4.2.5. Codi

La programació del microcontrolador s'ha realitzat amb llenguatge C. L'estructura bàsica és un menú principal amb diferents menús secundaris que permet visualitzar tota la informació dels dispositius. La navegació a través

d'aquests s'efectua mitjançant els polsadors implementats com a entrades digitals per pujar o baixar la selecció, entrar i sortir de cada menú.

L'estructura dels menús és la següent:

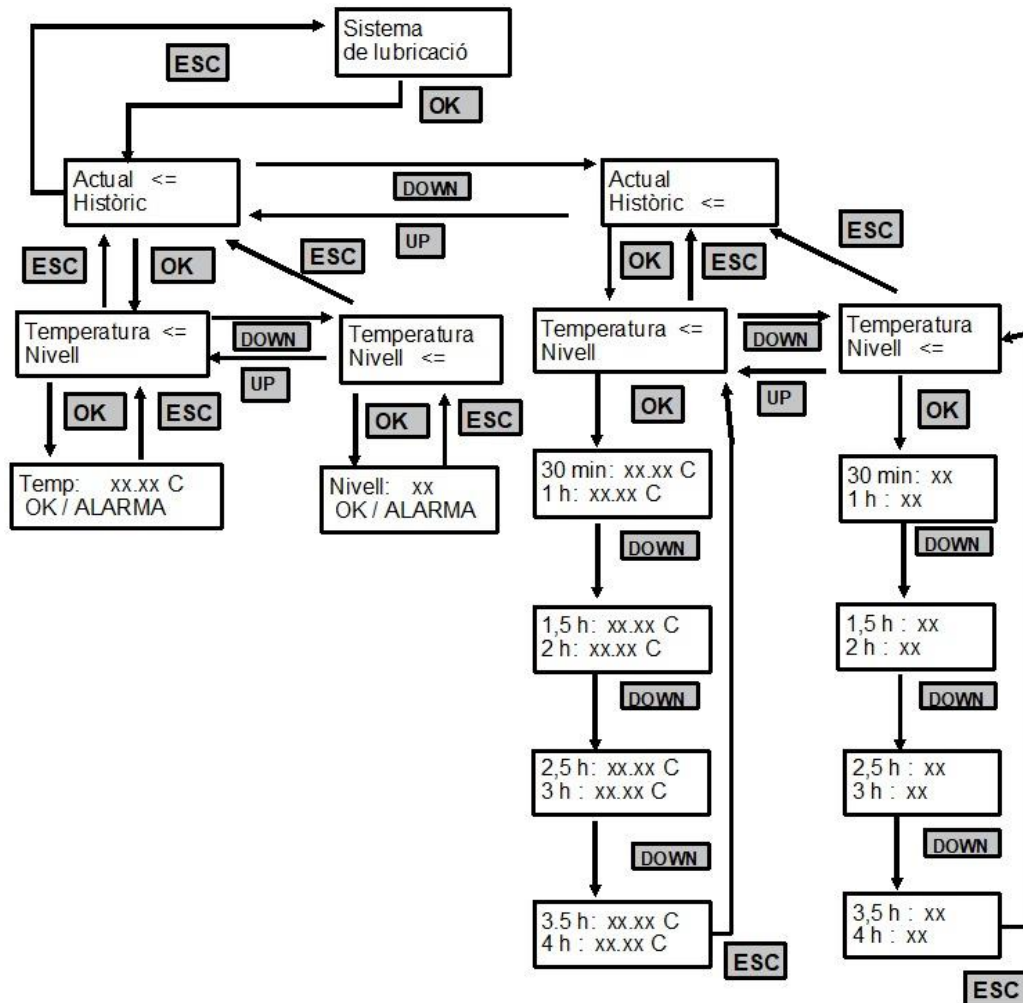


Figura 58. Estructura de menús. Creació pròpia.

El programa de control és un bucle infinit, que genera interrupcions sempre que es rep alguna dada a través del port de comunicació UART R_{x1}, situat en el pin 32 del PIC. La interrupció es pot produir per una petició de l'usuari o per un enviament programat del dispositiu remot. En el primer cas, el PIC envia les dades de petició al XBee a través del port de comunicació UART T_{x1}, situat en el pin 31 del PIC. En aquests cas la dada únicament es mostra per pantalla en temps real. En canvi, si la mostra és enviada automàticament per el dispositiu remot, el programa l'emmagatzema en la memòria i modifica el número de posició de la memòria.

Quan l'usuari vol recuperar les mostres de temps anteriors, el programa recupera fins a 8 mostres anteriors, corresponents a les 4 últimes hores, mitjançant el número de posició de la memòria. Les dades són mostrades per pantalla de manera correlativa.

La memòria està dividida en dues parts. S'utilitza la primera meitat de la memòria (2-511 bytes) per als valors de temperatura, i la segona (514 a 1023) per als valors de nivell. Els bytes 0, 1, 512 i 513 s'utilitzen per a la posició de memòria esmentada anteriorment i per a cadascun dels sensors.

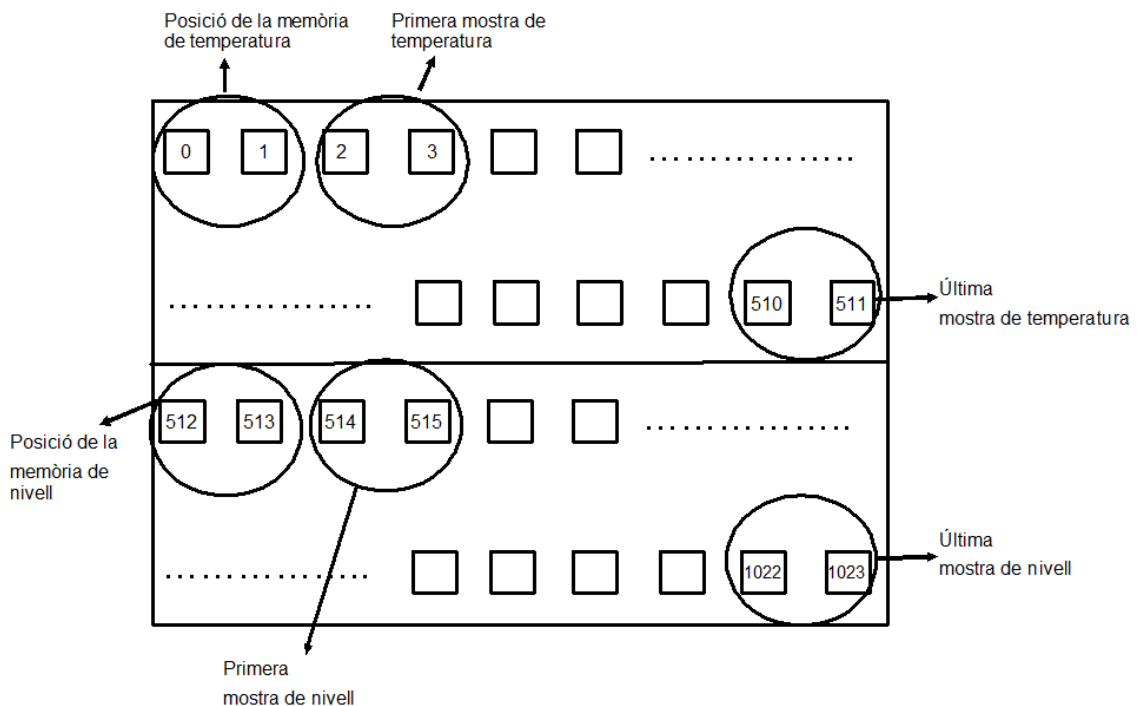


Figura 59. Estructura de la memòria. Creació pròpia.

Les mostres enviades per els sensors ocupen 2 bytes de memòria, per això s'haurà de dividir abans d'emmagatzemar i unir cada cop que les vulguem recuperar.

Per a imprimir els valors dels sensors per pantalla, s'han de convertir de hexadecimal a voltatge, i de voltatge al valor real de la mesura.

Per a la temperatura, es coneix que el convertidor A/D del XBee és de 10 bits ($2^{10}=1024$), i que el rang de voltatge en la sortida del sensor és de 0 a 3.3 volts.

Per tant, l'expressió de conversió és:

$$T_V = \frac{\text{mostra} \cdot 3.3}{1024}$$

El sensor de temperatura s'ha configurat un rang de 0 a 100°C, per tant, l'expressió que converteix la mostra en el valor de temperatura és:

$$T_{\text{C}} = \frac{T_V \cdot 100}{3.3}$$

Per al nivell, la primera expressió que converteix la mostra a voltatge és idèntica:

$$N_V = \frac{\text{mostra} \cdot 3.3}{1024}$$

Com que el voltatge de sortida del sensor no és lineal amb el nivell, s'ha programat mitjançant un percentatge en funció de la taula de relació voltatge-nivell.

El mode de visualització dels valors dels sensors en temps real, inclou un missatge que permet identificar ràpidament al usuari si el valor rebut és correcte o està fora dels valors preestablerts. En el cas de la temperatura, si el valor rebut és més baix de 25°C o superior a 75°C, s'imprimeix un missatge "ALARMA" en la segona fila de la pantalla (figura 60, esquerra i dreta). Si el valor és dins d'aquest interval, en canvi, es mostra un missatge de verificació "OK" (figura 60, centre).

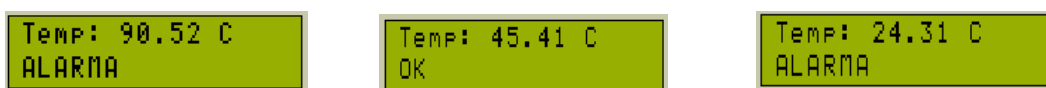


Figura 60. Visualització de la temperatura en la pantalla LCD.

El mateix ocorre en el cas del nivell, el límit establert es situa en el 25% o menys per al missatge "ALARMA" i superior a 25% per al missatge "OK".



Figura 61. Visualització del nivell en la pantalla LCD.

4.2.6. Connexió i simulació

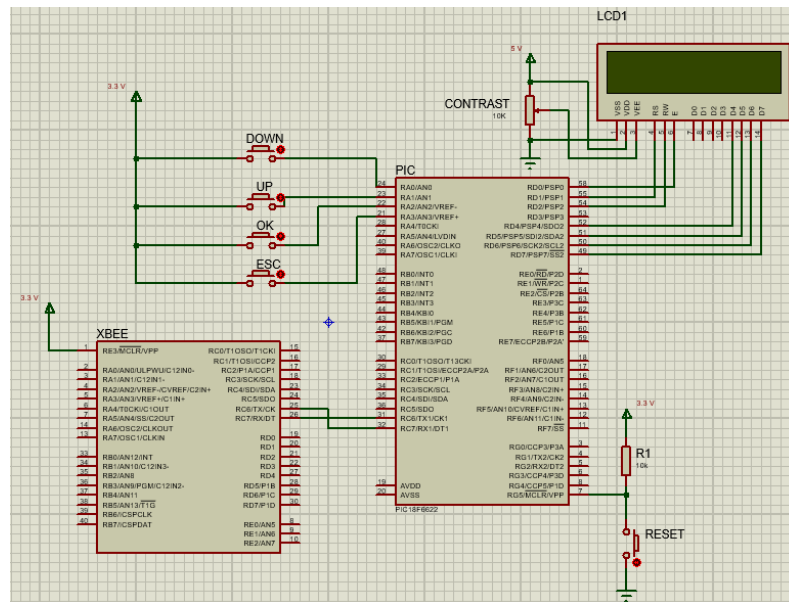


Figura 62. Connexió elements dispositiu central.

La connexió dels diferents elements s'observa en la figura 62. Els voltatges necessaris per al funcionament són de 3.3 V per al XBee i el PIC, i de 5 V per a la pantalla LCD. Aquesta última està connectada al PIC mitjançant 4 pins de dades (D4, D5, D6 i D7) per on s'envia la informació a visualitzar. També es connecten 3 pins més (RS, RW i E) que s'utilitzen per al control de la comunicació entre el PIC i la pantalla. La seva connexió elèctrica, com es pot veure, inclou una resistència variable que permet regular el contrast i visualitzar millor el contingut.

El PIC, apart de les connexions que ja s'han anomenat per a la pantalla LCD i els polsadors, es connecta als 3.3 V a través del pin MCLR. Quan aquest pin està amb tensió, alimenta el microcontrolador i es troba en mode operatiu, però quan es desconnecta provoca el mode inactiu del microcontrolador. El pin també s'utilitza en el moment de programar el PIC.

El XBee es connectarà als 3.3 V i els pins de transmissió i recepció serials amb el microcontrolador.

Per tal de simular el dispositiu XBee s'ha utilitzat un PIC16F887, ja que no existeix en el programa de simulació utilitzat, i d'ara en endavant s'anomenarà XBee. El XBee, actua igual que el mòdul RF real, rep la trama de sol·licitud a través del pin R_x (equivalent al pin D_{in}) procedent del pin T_{X1} del microcontrolador. Un cop processat envia el valor mesurat mitjançant la trama

de resposta a través del pin Tx (equivalent al pin D_{out}) cap al pin Rx₁ del microcontrolador. Per tal de generar una mostra automàtica i poder simular l'enregistrament i posterior recuperació, s'han utilitzat 2 polsadors que generen una mostra de temperatura i una mostra de nivell, respectivament.

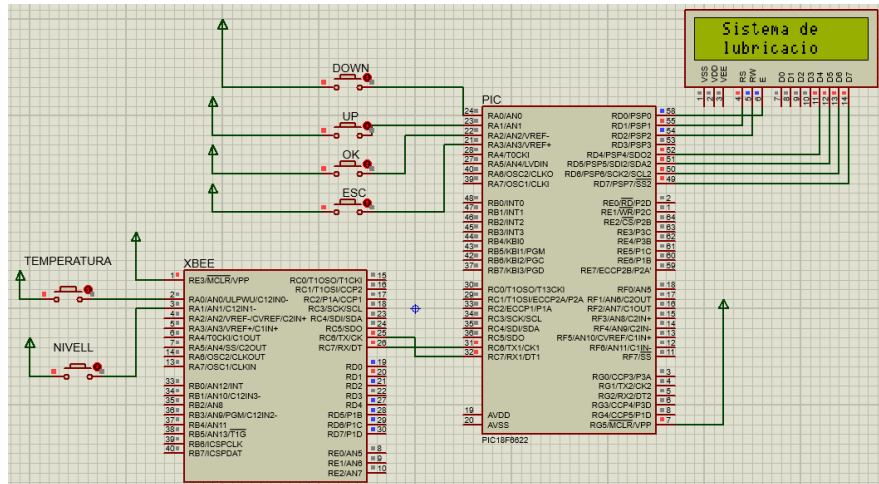


Figura 63. Sistema en repòs.

Des de l'estat del sistema en repòs, polsant 3 vegades 'OK', es demana al sistema que ens mostri el valor actual del sensor de temperatura.

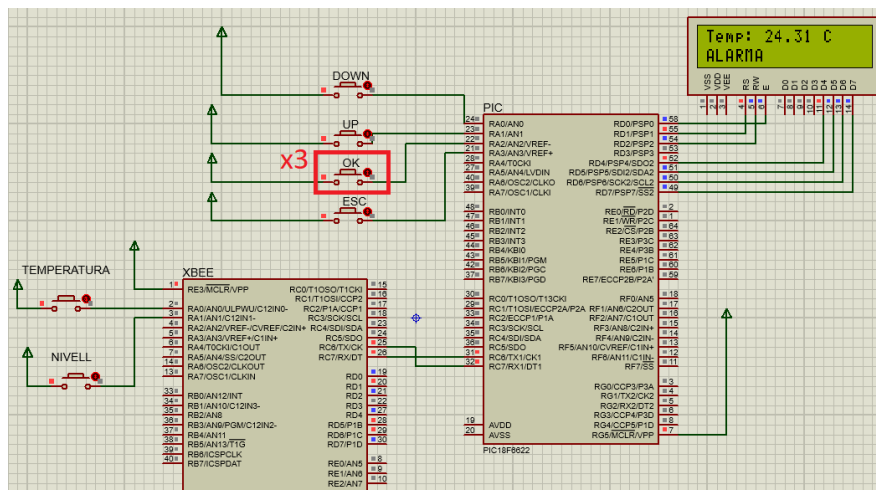


Figura 64. Visualització de la temperatura en temps real.

Es generen 3 mostres de temperatura des dels posadors del XBee, la última diferent de les dos anteriors. Des de l'estat del sistema en repòs, polsant la següent combinació: 'OK' + 'DOWN' + 'OK' + 'OK', es demana al sistema que ens mostri els valors de temperatura emmagatzemats.

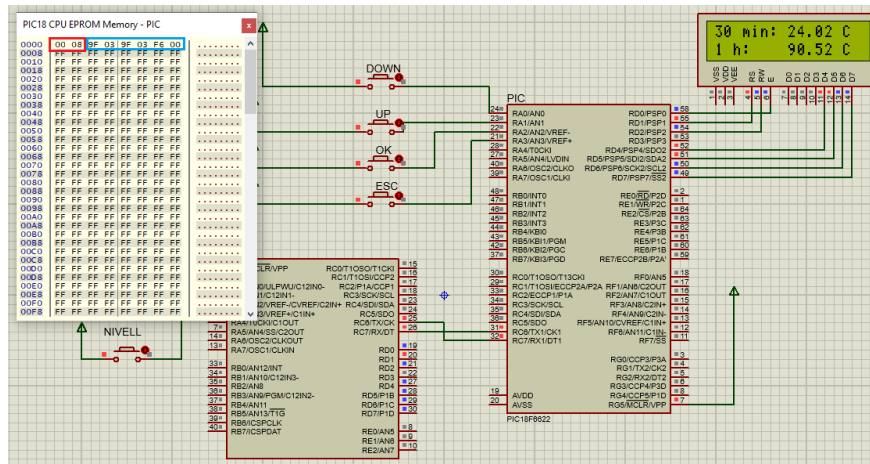


Figura 65. Visualització de la temperatura en la memòria.

S'observa en la figura 65, com la memòria de posició està preparada per a rebre una nova dada (8), i que les posicions de la 2 a la 7 estan ocupades per les mostres enviades pel XBee, essent la última diferent. El PIC llegeix la memòria en ordre invers a l'ordre que s'ha escrit. Com que la memòria per defecte conté el valor màxim (FF), s'ha definit que el PIC descarti aquest valor i mostri en pantalla '----' que simbolitza que la mostra és buida.

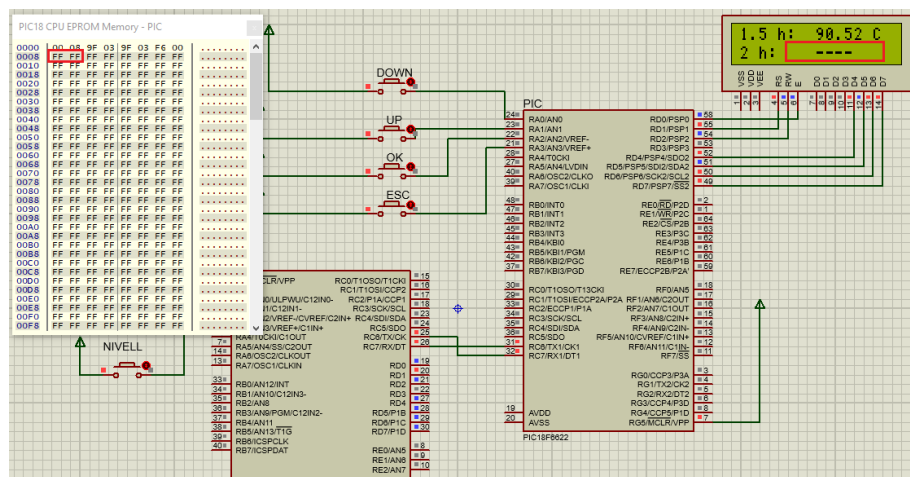


Figura 66. Visualització de memòria buida.




4.2.7. Font d'alimentació

En aquest dispositiu es necessitaran les següent tensions:

- 3.3 V: per al PIC18F6622.
- 5 V: per al LCD.
- 3.3 V: per al mòdul XBee.

El màxim consumidor de corrent és el mòdul XBee quan està enviant trames amb 33 mA, ja que tant la pantalla LCD (1.8 mA) com el PIC (6.8 μ A) tenen consums molt baixos.

En aquest dispositiu s'utilitzarà un altre model de la sèrie de transformadors que ja s'han utilitzat per als altres dispositius. Concretament el VSK-S3-9U:

RoHS   

| MODEL | output voltage (Vdc) | output current max (A) | output power max (W) | ripple and noise ¹ typ (mVp-p) | efficiency typ (%) |
|-------------|-------------------------|------------------------------|----------------------------|---|--------------------------|
| VSK-S3-3R3U | 3.3 | 0.7 | 2.3 | 30 | 63 |
| VSK-S3-5U | 5 | 0.6 | 3 | 30 | 72 |
| VSK-S3-9U | 9 | 0.33 | 3 | 30 | 74 |
| VSK-S3-12U | 12 | 0.25 | 3 | 30 | 76 |
| VSK-S3-15U | 15 | 0.2 | 3 | 30 | 76 |
| VSK-S3-24U | 24 | 0.125 | 3 | 30 | 78 |

Notes: 1. Ripple and noise measured at 20 MHz bandwidth

Figura 67. Especificacions transformador VSK-S3-9U.²²

El model s'implementarà segons les recomanacions del fabricant, com s'ha vist en els dispositius dissenyats anteriorment.

S'escull aquest model que subministra 9 V a la sortida, per tal d'utilitzar seguidament un regulador de tensió. Aquest regulador de tensió permet obtenir una tensió de 5 V molt estable i necessària per a l'alimentació dels components del dispositiu central. Es complementa amb un parell de condensadors, un a l'entrada de 100 μ F i l'altre a la sortida de 10 μ F.

S'inclou un interruptor que permetrà encendre i apagar el dispositiu, i un LED que ens indica si el dispositiu està encès o no, complementat amb una resistència de 330 Ω per tal de limitar el corrent que circula per aquest. Un cop s'obtenen els 5 V per a la pantalla LCD, es dissenya mitjançant dues resistències un divisor de tensió per tal d'alimentar el mòdul XBee i el PIC.

El disseny final de la font d'alimentació és:

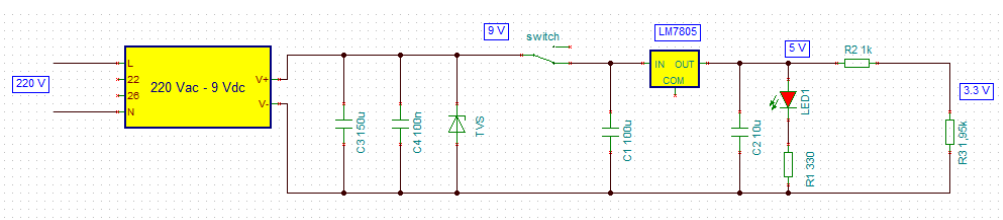


Figura 68. Disseny de la font d'alimentació del dispositiu central.

²² https://www.mouser.es/datasheet/2/670/vsk_s3-1313305.pdf

5. FUTURES AMPLIACIONS

En aquest apartat s'inclouen algunes petites modificacions que es podrien implementar, per tal de fer el dispositiu encara més versàtil i adquirir més prestacions. Algunes d'aquestes ja s'havien contemplat al inici del projecte, però per motius que ja s'havien tingut en compte en el anàlisi de riscos, no s'han pogut dur a terme. Es tracta del mòdul Bluetooth i de l'aplicació mòbil. La codificació del programa del PIC s'ha allargat més del previst, degut a la poca experiència en codificació de transmissions UART. D'aquesta manera, s'ha prioritzat el funcionament principal del sistema en front a les prestacions superiors que s'havien previst.

5.1. Mòdul Bluetooth i aplicació mòbil.

Segons Cardozo (2003), els inicis del sistema de transmissió sense fils Bluetooth es troben al 1994 quan Ericsson Mobile Telecommunications va començar a treballar amb senyals de radio de baixa potència i baix cost entre dispositius, atraient altres companyies importants. Així, es pot dir que és una tecnologia molt madura i àmpliament implementada arreu.

Els propòsits d'aquesta tecnologia són:

- Facilitar comunicacions entre dispositius.
- Eliminar els cables i connectors.
- Crear petites xarxes entre dispositius personals.

Bluetooth utilitza la banda de transmissió ISM de 2.4 GHz i utilitza una topologia mestre-esclau de fins a 8 dispositius simultanis (1 mestre i 7 esclaus) formant una piconet. La banda d'operació es divideix en canals de 1 MHz i utilitza la modulació GFSK (Gaussian Frequency Shift Keying).

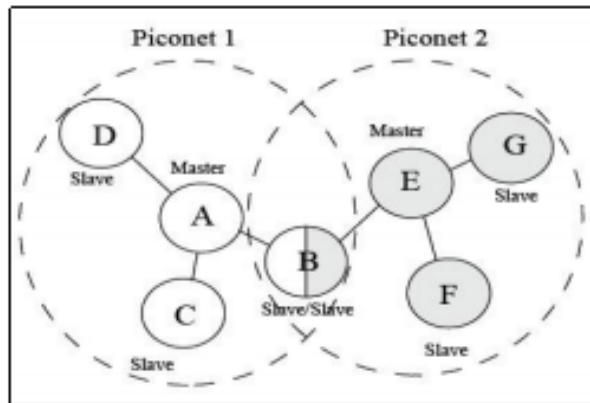


Figura 69. Dues piconet.²³

La figura 69 mostra una scatternet, ja que un mateix dispositiu es pot trobar en dues piconet diferents però només pot estar actiu en una de les dues.

El protocol utilitza la tècnica de codificació d'espectre eixamplat per salts de freqüència (FHSS, Frequency Hopping Spread Spectrum) que permet obtenir una senyal molt robusta i resistent a les interferències electromagnètiques. Els salts de freqüència d'una a l'altre permeten que una transmissió només pugui ser desxifrada per el receptor correcte, ja que aquests salts segueixen un patró pseudoaleatori. El temps T_C fa referència a la ranura de temps que està transmeten a la mateixa freqüència portadora, anomenada dwell time.

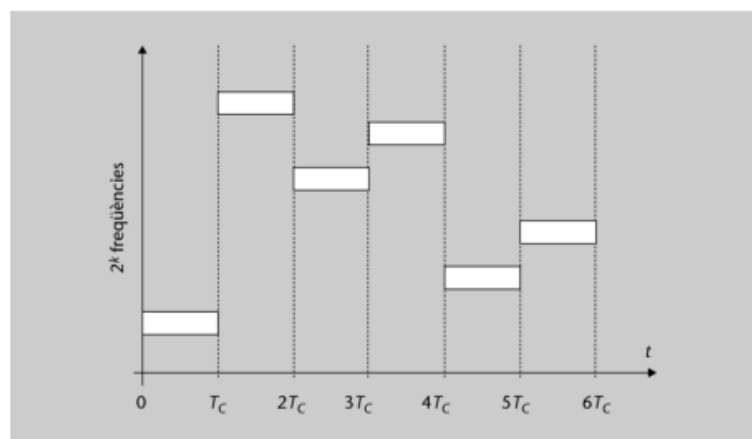


Figura 70. Distribució temporal dels salts de freqüència.²⁴

²³ <http://bibing.us.es/proyectos/abreproy/11972/fichero/Cap%C3%ADtulo+2+-+Bluetooth.pdf>

²⁴ Mòdul 4. Codificació i Modulacions Avançades. UOC.

La pila de protocols que utilitza el sistema Bluetooth es pot observar resumida en la imatge.

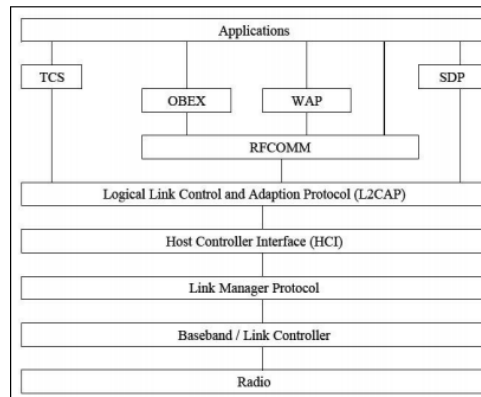


Figura 71. Model OSI per a Bluetooth.²³

La capa radio defineix les característiques del transceptor del dispositiu a 2.4 GHz. La capa de control d'enllaç s'encarrega de controlar l'accés als canals de dos maneres diferents, síncron orientat a connexió (SCO) y asíncron sense connexió (ACL), el primer utilitzat per a àudio en temps real i el segon per a paquets de dades.

LMP és la capa encarregada de l'establiment i configuració del enllaç entre els dispositius, també és la responsable de la seguretat de la piconet. La capa HCI és la interfície amb el mòdul de ràdio. L2CAP adapta els protocols de les capes inferiors. SDP és l'encarregat de buscar els serveis i les característiques d'aquests. RFCOMM transporta les dades a les capes superiors i és el responsable de la possibilitat de realitzar múltiples connexions simultànies. TCS és una especificació exclusiva per al control de trucades y dades entre dispositius.

El mòdul necessari per a implementar connexió Bluetooth al dispositiu central seria un amb la capacitat d'esclau, ja que la funció de mestre la duria a terme el dispositiu mòbil. En el mercat existeixen multitud de mòduls amb aquesta funció, però s'ha pensat amb el model HC-06 ideal per a una connexió fàcil entre els dispositiu central i el mòbil mitjançant microcontroladors.

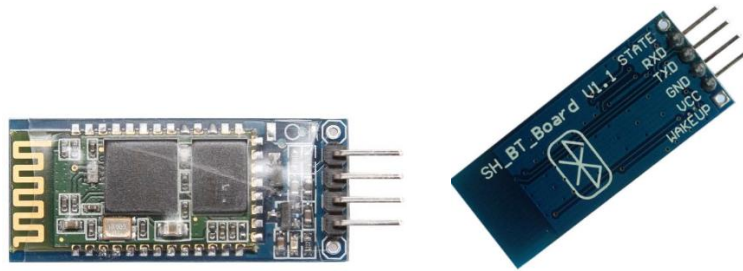


Figura 72. Mòdul HC-06.²⁵

Un cop implementat aquest mòdul, s'ha de dissenyar una aplicació que utilitzant el Bluetooth d'un dispositiu mòbil, com ara un mòbil o una tauleta, permeti la comunicació amb el dispositiu central. Actualment la creació de aplicacions mòbils quasi està al abast de tothom, gràcies a programes senzills i molt intuïtius que corren sobre sistemes Android, alguns gratuïts i d'altres més potents de pagament.

Per a aquest projecte s'havia pensat amb l'aplicació online i gratuïta MIT App Inventor. Aquesta web permet crear el disseny de l'aplicació mitjançant botons, menús desplegable i estructures de pantalles.



Figura 73. Disseny interfície de l'aplicació. Creació pròpia.²⁶

La mateixa aplicació et permet dissenyar les funcions mitjançant una estructura de sentències molt intuïtives. Es capaç de crear la navegació dins de l'aplicació,

²⁵ <https://store.prometec.net/producto/modulo-bluetooth-hc-06/>

²⁶ <https://appinventor.mit.edu/>

incloure la connexió Bluetooth amb el dispositiu a enllaçar, i fins i tot emmagatzemar les dades en el dispositiu mòbil.

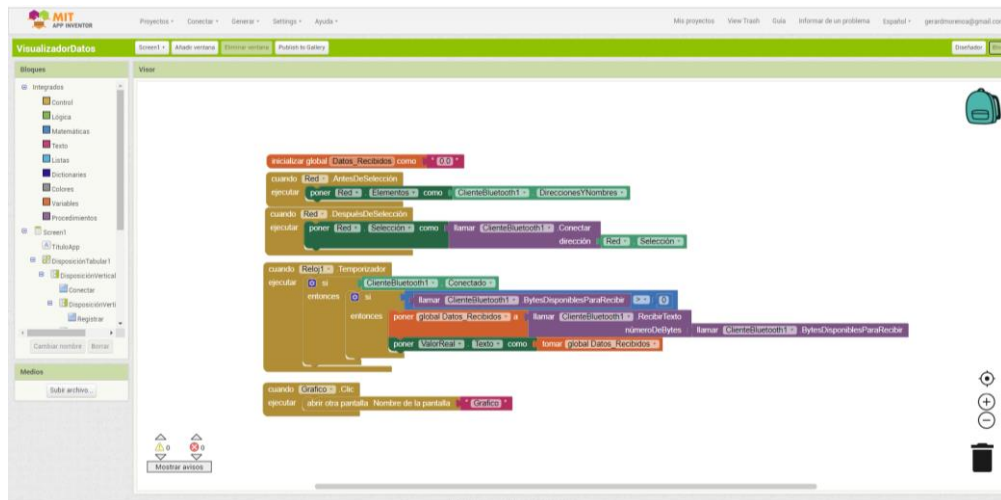


Figura 74. Disseny funcionalitats de l'aplicació. Creació pròpia.²⁶

5.2. Ampliació de la xarxa.

El sistema dissenyat contempla la monitorització de dues variables d'un mateix sistema d'un únic compressor d'etilè. Les plantes químiques estan formades per desenes d'equips amb dos o tres sistemes cadascun i diverses variables a controlar dins d'aquests. Es pot considerar que les plantes químiques poden arribar a controlar milers de variables. El sistema, que es tracta en aquest projecte, té la capacitat per a poder ampliar-se gràcies a la capacitat del protocol ZigBee i del mòduls XBee tractats. Com s'ha vist, ZigBee permet fins a més de 65000 nodes interconnectats en una xarxa.

Els dispositius locals es configurarien com a End Device, els dispositius centrals com a Routers i s'hauria de dissenyar un dispositiu que inclouria el Coordinador on es concentrarien totes les dades de la planta. Cada End Device podria incloure diferents senyals analògics i digitals d'un dels sistemes d'un equip de la planta i tots els sistemes quedarien agrupats en Routers. Els Routers seguirien tenint la capacitat de visualització, mitjançant una pantalla, de totes les variables dels sistemes associats al equip.

5.3. Memòria externa.

Aquesta modificació aniria estretament lligada a l'anterior, és a dir, els dispositius centrals governats per els XBee configurats com a Routers, controlarien algunes desenes de variables i la memòria necessària per a emmagatzemar les diferents mostres d'aquestes variables quedaria molt limitada.

La possibilitat d'instal·lar una memòria EEPROM externa seria una solució a aquest inconvenient. La comunicació amb aquesta memòria s'efectuaria mitjançant transmissió serial I²C (Inter-Integrated Circuit). Es tracta d'una comunicació bidireccional que permet la interacció entre diferents dispositius connectats a un bus amb una estructura formada per un mestre i diferents esclaus. Segons Balseca (2009) aquest sistema de comunicació va ser desenvolupat per Philips Semiconductors a principis dels anys 80.

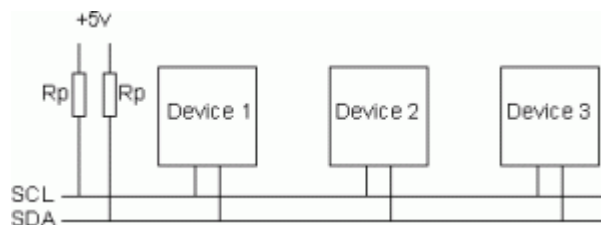


Figura 75. Connexió I²C.²⁷

I²C utilitza dos fils, un anomenat SDA per a les dades i un altre anomenat SCL per al rellotge que sincronitza totes les dades. Les dues línies es connecten a tots els dispositius i necessiten dues resistències de polarització R_p a través d'una alimentació de 5V. El dispositiu mestre és l'encarregat de controlar la línia del rellotge SCL i és l'únic que pot iniciar una transferència.

El PIC18F6622 seleccionat per al dispositiu central ja disposa de dos ports serials d'aquest tipus i d'aquesta manera es podria solucionar i poder ampliar la memòria per donar cabuda a l'augment de variables emmagatzemades.

Actualment, existeixen moltes memòries EEPROM i de moltes capacitats. A més el sistema I²C permet incloure múltiples memòries en paral·lel al llarg del bus.

²⁷ <https://www.diarioelectronico hoy.com/blog/introduccion-al-i2c-bus>

Un exemple de Microchip és la memòria AT24CM02, que multiplica per 256 la capacitat d'emmagatzematge respecte el PIC18F6622 escollit per al disseny.

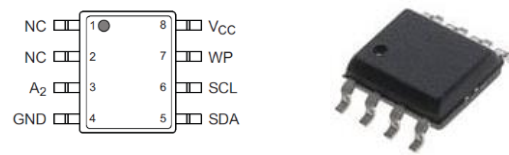


Figura 76. Memòria externa EEPROM AT24CM02.²⁸

²⁸ <https://www.mouser.es/datasheet/2/268/Atmel-8828-SEEPROM-AT24CM02-Datasheet-1398087.pdf>

6. CONCLUSIONS

Un cop finalitzat el disseny del sistema sense fils de monitorització de sensors, es pot concloure que ZigBee permet crear xarxes de sensors a distàncies curtes, però suficientment grans per a les plantes de la indústria química. El nombre de dispositius interconnectats és el punt fort d'aquesta tecnologia, els 65.000 dispositius que permet connectar el fan ideal per als milers de variables que s'han de controlar en les plantes de producció. El seu punt dèbil és el preu dels transmissors, que es situa entre els més cars de les WSN i encareix una mica la seva implementació. No obstant, és molt més econòmica i fàcil d'instal·lar que xarxes més potents com ara el WiFi o la tecnologia mòbil.

El sensor de nivell utilitzat per a aquest prototip és inadequat degut a la seva mida i, sobretot, a la seva precisió. El sector utilitza altres instruments més precisos, però a la vegada més cars. El projecte tracta simplement de comprovar la viabilitat del sistema i una aproximació de les seves funcionalitats, minimitzant recursos.

El disseny del dispositiu de mesura de temperatura utilitzat tampoc és gaire acurat, ja que es poden trobar en el mercat específic per a plantes químiques, models molt més desenvolupats i molt més precisos. Aquests models però, són molt més cars i s'ha prioritzat poder construir un prototip més econòmic.

Els objectius principals del treball es poden considerar assolits, ja que el sistema dissenyat és capaç de digitalitzar variables físiques, centralitzar-les en un sol dispositiu i accedir a les dades des d'un punt allunyat del equip mesurat. No obstant, els objectius específics s'han vist alterats degut a un increment de temps invertit en la codificació del programa de processament i visualització de les dades. El retard en aquesta codificació ve donat per la falta d'experiència en la programació en transmissions UART, ja que aquestes són de 8 bits i les dades enviades pels transmissors són de 10 bits i això implica que s'han de dividir en dos bytes cada cop que es rep una dada. Aquest increment de temps s'ha traduït amb l'eliminació del mòdul Bluetooth i l'aplicació per a mòbil dins del projecte i en conseqüència no s'han pogut assolir la totalitat dels objectius específics que s'havien definit prèviament.

En el moment de la construcció del hardware també s'han tingut problemes, ja que finalment i per falta de temps, no s'ha pogut acabar implementant els dispositius. En el moment de connectar els dispositius XBee amb el sensor, el valor analògic que es rebia era incorrecte. Aquest fet pot ser degut a una connexió incorrecte de les referències de voltatge del transmissors, fent que el convertidor A/D no sigui capaç d'obtenir el valor real mesurat pel sensor.

Pel que fa al circuit d'adaptació, finalment es va decidir implementar el circuit d'adaptació de la termoresistència mitjançant soldadura. Degut a la gran precisió necessària a l'hora de soldar, la soldadura no era del tot acurada i això també podria afectar a la qualitat del senyal obtingut i provocar problemes en el convertidor A/D.

Aquest treball m'ha servit per obtenir experiència en el desenvolupament de projectes, ja que aquesta era nul·la per part meua. El fet de viure tots els passos des de la planificació, passant pel disseny i la construcció, fins arribar a aquestes conclusions, ajuda a veure els errors comesos i fer una valoració crítica del treball realitzat. Segurament, per aquesta falta d'experiència, s'ha pecat d'ambicions i poc realista amb el temps necessaris per a cadascuna de les tasques. En futurs projectes, això m'ajudarà a ajustar millor les tasques a realitzar i el temps necessari per dur-les a terme.

El treball també m'ha servit per adquirir habilitats noves en el sector de la electrònica, com ara soldadura de circuits, codificar programes amb una finalitat real, la càrrega de programes dins de microcontroladors, aprendre a utilitzar transmissors RF, entre d'altres. Tot i que, al haver hagut d'aprendre en tant poc temps i amb un solapament amb la redacció d'aquest treball, els resultats són poc acurats.

7. GLOSSARI

Android: conjunt de programari, sistema operatiu i aplicacions per a dispositius mòbils.

API: mode de transmissió basat en trames editables entre dispositius XBee.

AT: mode de transmissió automàtic entre dispositius XBee.

EEPROM: memòria ROM programable i esborrable elèctricament.

LCD: tipus de pantalla de visualització de dades, basat en cristall líquid.

LoRa: protocol de transmissió sense fils.

Model OSI: protocol de referència de sistemes oberts.

PCB: placa de circuit imprès.

PIC: circuits integrats programables fabricats per Microchip Technology Inc.

Protoboard: placa de proves per a construir prototips de circuits electrònics.

PT100: tipus de termoresistència utilitzada per a mesurar temperatura mitjançant contacte.

RF: abreviació de radiofreqüència.

WSN: xarxa de sensors sense fils.

XBee: mòdul RF de transmissió/recepció del protocol ZigBee de DIGI Inc.

ZigBee: protocol de transmissió sense fils.

Z-Wave: protocol de transmissió sense fils.

8. BIBLIOGRAFIA I WEBGRAFIA

- ADELANTADO, G. (2017). Las termorresistencias y su representación según las normas IEC y ASTM. Instrumentos WIKA S.A.U. Extret de: <https://www.bloginstrumentacion.com/productos/termorresistencias-su-representacin-segn-las-normas-iec-astm/>
- AMARILLO, M.O.; TRUJILLO, L.C. (2014) Simulación de redes de sensores inalámbricos: Un modelo energético a nivel de Nodo-Sensor bajo las especificaciones IEEE 802.15.4 y Zigbee. Pontificia Universidad Javeriana, Departamento de Electrónica, Bogotá, DC, Colombia. Extret de: <https://hemeroteca.unad.edu.co/index.php/publicaciones-e-investigacion/article/view/1430/1749>
- ANAYA, L.M.; CORDOBA R.A. (2012). Evaluación y análisis del desempeño a nivel físico de un sistema de comunicación de datos punto a punto en 2.4 GHz basado en ZIGBEE. Facultad de Ingeniería Electrónica y de Telecomunicaciones. Popayán. Extret de: <http://repositorio.unicauca.edu.co:8080/bitstream/handle/123456789/1834/EVALUACION%20Y%20ANALISIS%20DEL%20DESEMPE%c3%91O%20A%20NIVEL%20FISICO%20DE%20UN%20SISTEMA%20DE%20COMUNICACION%20DE%20DATOS%20PUNTO%20A.pdf?sequence=1&isAllowed=y->
- BALSECA, R.A.; SINCHIGUANO, M.A. (2009). Integración de sistemas de comunicación digital aplicado al control de ruta en una unidad de transporte urbano. Escuela Superior Politécnica de Chimborazo. Extret de: <http://dspace.espoch.edu.ec/bitstream/123456789/167/1/38T00158.pdf>
- BOE (2017). Orden ETU/1033/2017, de 25 de octubre, por la que se aprueba el cuadro nacional de atribución de frecuencias. Extret de: <https://www.boe.es/buscar/pdf/2017/BOE-A-2017-12318-consolidado.pdf>
- CABRERA, M.; REY, F.; TARRÉS, F. (2012). Sistemes de comunicació I. Universitat Oberta de Catalunya.
- CABRERA, M.; REY, F.; TARRÉS, F. (2014). Codificació i modulacions avançades. Universitat Oberta de Catalunya.
- CARDOZO, C.A.; MASTRASCUSA J.M. (2003). Tecnología Bluetooth: conexión sin cables. Facultad de Ingeniería de Sistemas de Cartagena de Indias. Extret de: <https://repositorio.utb.edu.co/bitstream/handle/20.500.12585/992/0024207.pdf?sequence=1>
- COMPONENTS101. (2018) XBee S2C – RF Module. Extret de: <https://components101.com/wireless/xbee-s2c-module-pinout-datasheet>

DÁVILA, M.A.; PÉREZ, J.F.; MORENO, J.E. (2017). Diseño de una Red Inalámbrica con Tecnología Zigbee para la Implementación de un Sistema Domótico. REVISTA DE CIENCIA E INGENIERÍA DEL INSTITUTO TECNOLÓGICO SUPERIOR DE COATZACOALCOS. Extret de: https://www.researchgate.net/profile/Maria_Davila5/publication/311207799_Diseño_de_una_red_inalambrica_tipo_ZigBee_para_la_implementacion_de_un_sistema_domotico/links/58b7a11492851c471d47abbe/Diseño-de-una-red-inalambrica-tipo-ZigBee-para-la-implementacion-de-un-sistema-domotico.pdf

DÍAZ, B. (2015). UART Universal Asynchronous Receiver Transmitter. Universidad Carlos III de Madrid. Extret de: https://e-archivo.uc3m.es/bitstream/handle/10016/23730/TFG_Borja_Diaz_Mulas.pdf?sequence=1&isAllowed=y

DIGI (2020) DIGI XBEE® S2C DIGIMESH® 2.4. Embedded RF modules deliver wireless connectivity using the innovative and easy-to-deploy DigiMesh protocol. Extret de: https://www.mouser.es/datasheet/2/111/digiinternational_10192016_s2c-xbeedigimesh-24-1163821.pdf

DIGI (2020) XBee®/XBee-PRO S2C Zigbee® RF Module. Extret de: <https://www.digi.com/resources/documentation/digidocs/pdfs/90002002.pdf>

DIGI International Inc. (2018). *XCTU overview*. Extret de: https://www.digi.com/resources/documentation/digidocs/90001458-13/default.htm#concept/c_xctu_layout.htm%3FTocPath%3DXCTU%2520overview%7C_0

GARCÍA, L. (2019). Medida de temperatura en entornos Industriales. Universidad Carlos III de Madrid. Escuela Politécnica Superior de Leganés. Extret de: https://e-archivo.uc3m.es/bitstream/handle/10016/30474/TFG_Lucia_Garcia_Jimenez.pdf?sequence=1&isAllowed=y

GÓMEZ, E. (2018). Cómo conectar una PT100 ¿ Qué es una RTD?. Extret de: <https://www.rinconingenieril.es/como-conectar-una-pt100/>

IEC (2008). International Standard. Industrial platinum resistance thermometers and platinum temperature sensors. INTERNATIONAL ELECTROTECHNICAL COMMISSION. ICS 17.200.20. Extret de: https://webstore.iec.ch/preview/info_iec60751%7Bed2.0%7Db.pdf

LORA ALLIANCE (2015). Lorawan. What is it?. A technical overview of Lora and Lorawan. Technical Marketing Workgroup 1.0. Extret de: <https://lora-alliance.org/sites/default/files/2018-04/what-is-lorawan.pdf>

MARCO, M.J.; VILAPLANA, J. (2010). Introducció a la programació. Universitat Oberta de Catalunya.

- MÁRQUEZ, J.M. (2020). Desarrollo e implementación de una red de sensores para monitorización basada en ZigBee. Escuela Técnica Superior de Ingeniería. Sevilla. Extret de: https://repositorio.uoyola.es/bitstream/handle/20.500.12412/2260/TFG_JMM_ARQUEZ.pdf?sequence=1&isAllowed=y
- MARTINEZ, J.A.; BARÁ, M. (2014). Electrónica digital. Universitat Oberta de Catalunya.
- SERNA, S. (2009). Especificación de perfil zigbee para monitoreo y control de plantas industriales Tecno Lógicas, núm. 23, pp. 167-185. Instituto Tecnológico Metropolitano Medellín, Colombia. Extret de: <https://www.redalyc.org/pdf/3442/344234316010.pdf>
- SUÁREZ, A., et al (2009). Interfaz programable de comunicación serie asincrónica en FPGA, Revista Ingeniería Electrónica, Automática y Comunicaciones, Vol. 30, Num 1. Extret de: <https://rielac.cujae.edu.cu/index.php/rieac/article/view/42/26>
- TEXAS INSTRUMENTS (2015). µA741 General-Purpose Operational Amplifiers. Extret de: https://www.electronicoscaldas.com/datasheet/uA741_TI.pdf
- VIQUIPÈDIA (2016) ETSI EN 300 328. Extret de: https://ca.wikipedia.org/wiki/ETSI_EN_300_328
- WIKA (2010). Termómetro OEM de inserción con cable Modelo TF45. Instrumentación de temperatura eléctrica. Barcelona. Extret de: https://media.automation24.com/datasheet/es/DS_TE6715_es_es_28573.pdf
- WIKA (2016). El uso de termopares. Información técnica. Barcelona. Extret de: https://www.wikapolska.pl/upload/DS_IN0023_es_es_62452.pdf
- WIRELESS SENSOR NETWORKS PROJECT TEAM (2014). Internet of Things: Wireless Sensor Networks. International Electrotechnical Commission. Switzerland. Extret de: <https://www.iec.ch/whitepaper/pdf/iecWP-internetofthings-LR-en.pdf>
- ZIGBEE (2015). ZigBee Specification. Extret de: <https://zigbeealliance.org/wp-content/uploads/2019/11/docs-05-3474-21-0csg-zigbee-specification.pdf>
- ZIGBEE ALLIANCE (2020). The full-Stack solution interlacing all your smart devices. <https://zigbeealliance.org/solution/zigbee/>
- Z-WAVE (2020). About Z-Wave Technology. https://z-wavealliance.org/about_z-wave_technology/
- Z-WAVE (2020). Smarter Safer. <https://www.z-wave.com/learn>

9. ANNEXOS.

```
1 #include <18f6622.h>
2 #uses xt,nowdt
3 #use delay(clock=4000000) // Relloige intern de 4 MHz
4 #use rs232(baud=9600,xmit=PIN_06,rcv=PIN_07,stream=PIC, parity=N, bits=8) // Canal de transmissió UART
5 #include <lcd.c> // Libreria funcions per a LCD
6
7 int Data_H,Data_L,Data_H2,Data_L2,Data_H3,Data_L3,pos_H_Lev,pos_L_Lev,pos_H_Temp,pos_L_Temp,device,data;
8 int count=0,force=0;
9 float tempC,level,tempCl,level1;
10 int16 DataLong,DataLong2,DataLong3,posLong;
11 int ok,A,B,C;
12
13 float calc_To_Print(int16 data_Long,int device);
14
15 #int_RDA
16
17 RDA_isr() // Execució cada cop que es rebin dades del XBee
18 {
19     data=getc();
20     if(data==0x7E){ // El byte delimitador ens indica que és una trama correcta
21         count=1;
22         while(count<5){ // Obvia els bytes que no interessen
23             data=getc();
24             count++;
25         }
26         data=getc();
27         count++;
28         if(data==0x01){ // // Llegeix la direcció del
29             device=1; // //
30         } // //
31         if(data==0x02){ // dispositiu que envia per seleccionar
32
33             device=2; // // la memòria i guardar el valor
34         } // //
35         while(count<20){ // Obvia els bytes que no interessen
36             data=getc();
37             count++;
38         }
39         Data_L=data; // // Llegeix els dos bytes de la mostra per separat
40         Data_H=getc(); // //
41
42         DataLong=makel6(Data_L,Data_H); // Genera la dada de 16 bits (2 bytes) enviada per XBee
43
44         if(force==0){ // Mostra no forçada que es guarda a la memòria
45
46             if(device==1){ // // TEMPERATURA
47
48                 pos_H_Temp=READ_EEPROM(0); // // Llegeix la última posició
49                 pos_L_Temp=READ_EEPROM(1); // //
50                 posLong=makel6(pos_H_Temp,pos_L_Temp); // // de la memòria
51
52                 if(posLong==512){ // La memòria torna al principi
53                     posLong=2;
54                 }
55                 WRITE_EEPROM(posLong,Data_H); // //
56                 posLong++; // // Guarda la mostra
57                 WRITE_EEPROM(posLong,Data_L); // //
58                 posLong++; // //
59
60                 pos_H_Temp=make8(posLong,1); // //
61                 pos_L_Temp=make8(posLong,0); // // Actualitza la posició
62                 WRITE_EEPROM(0,pos_H_Temp); // //
63
64                 WRITE_EEPROM(1,pos_L_Temp); // // de la memòria
65             }
66             if(device==2){ // // NIVELL
67
68                 pos_H_Lev=READ_EEPROM(512); // // Llegeix la última posició
69                 pos_L_Lev=READ_EEPROM(513); // //
70                 posLong=makel6(pos_H_Lev,pos_L_Lev); // // de la memòria
71
72                 if(posLong==1024){ // La memòria torna al principi
73                     posLong=514;
74                 }
75                 WRITE_EEPROM(posLong,Data_H); // //
76                 posLong++; // // Guarda la mostra
77                 WRITE_EEPROM(posLong,Data_L); // //
78                 posLong++; // //
79
80                 pos_H_Lev=make8(posLong,1); // //
81                 pos_L_Lev=make8(posLong,0); // // Actualitza la posició
82                 WRITE_EEPROM(512,pos_H_Lev); // //
83                 WRITE_EEPROM(513,pos_L_Lev); // // de la memòria
84             }
85             data=getc(); // // Llegeix el checksum
86         }
87         device=0; // // Reinicia els parametres
88         count=0; // //
89         force=0; // //
90     }
91 }
92 void main() // Programa principal
93 {
94     set_tris_A(0xFF); // Configura el port A com entrada (polsadors)
95
96     lcd_init(); // S'inicialitza la pantalla LCD
97
98     enable_interrupts(INT_RDA); // Habilita Interrupcions i
99     enable_interrupts(GLOBAL); // escolta el canal UART per rebre les dades
100
101     while(1) // Bucle d'execució infinita
102     {
103         printf(lcd_putc,"%f"); // //
104         lcd_gotoxy(3,1); // // Pantalla
105         printf(lcd_putc,"Sistema de"); // // Principal
106         lcd_gotoxy(3,2); // //
107         printf(lcd_putc,"lubricacio"); // //
108         delay_ms(200); // //
109         if(!input(pin_A2)==0){ // Entrar
110             ok=1;
111             A=2;
112             delay_ms(500);
113         }
114         while(ok==1){ // Menu principal
115
116             if(!input(pin_A1)==0){ // Baixa cursor
117                 A=2;
118                 delay_ms(200);
119             }
120             if(!input(pin_A0)==0){ // Puja cursor
121                 A=1;
122                 delay_ms(200);
123             }
124             if(!input(pin_A3)==0){ // Sortir
```

```

125     ok=0;
126     delay_ms(500);
127 }
128 if(A==2){
129     printf(lcd_putc,"%f"); // Menu Principal
130     lcd_gotoxy(1,1); // Menu Principal
131     printf(lcd_putc,"Actual <="); // Primera
132     lcd_gotoxy(1,2); // Opció
133     printf(lcd_putc,"Historic"); //
134     delay_ms(200); //
135 }
136 if(!input(pin_A2)==0){ // Entrar
137     B=1;
138     A=2;
139     delay_ms(500);
140 }
141 while(B==1){ // Menu secundari
142     if(!input(pin_A1)==0){ // Baixa cursor
143         A=2;
144         delay_ms(200);
145     }
146     if(!input(pin_A0)==0){ // Puja cursor
147         A=1;
148         delay_ms(200);
149     }
150     if(!input(pin_A3)==0){ // Sortir
151         B=0;
152         A=2;
153         delay_ms(500);
154     }
155 }
156 if(A==2){ //

```

```

156     printf(lcd_putc,"%f"); // Menu Secundari
157     lcd_gotoxy(1,1); // Menu Secundari
158     printf(lcd_putc,"Temperatura <="); // Primera
159     lcd_gotoxy(1,2); // Opció
160     printf(lcd_putc,"Nivell"); //
161     delay_ms(200); //
162 }
163 if(!input(pin_A2)==0){ // Entrar
164     C=1;
165     delay_ms(500);
166 }
167 while(C==1){ // Visualització Temperatura
168     {
169         putc(0x7E); // Demana dada de temperatura
170         putc(0x00); // Es configura en funció del dispositiu
171         putc(0x10);
172         putc(0x17);
173         putc(0x01);
174         putc(0x01);
175         putc(0x13);
176         putc(0x22);
177         putc(0x00);
178         putc(0x40);
179         putc(0x0B);
180         putc(0x78);
181         putc(0x4E);
182         putc(0xFF);
183         putc(0xFE);
184         putc(0x00);
185         putc(0x49);
186     }

```

```

187     putc(0x53);
188     putc(0x00);
189     putc(0x08);
190 }
191 force=1; // Marca la interrupció com a forçada
192 tempC=calc_To_Print(DataLong,1); // Calcula el valor a mostrar
193 printf(lcd_putc,"%f"); // Mostra
194 lcd_gotoxy(1,1); // Dada
195 printf(lcd_putc,"Temp: %f C",tempC); // Temperatura
196 if(tempC>70 || tempC<25){
197     lcd_gotoxy(1,2); // Mostra missatge d'alarma
198     printf(lcd_putc,"ALARMA"); // Si és superior a 70°C o és inferior a 25°C
199     delay_ms(200);
200 }
201 else{
202     lcd_gotoxy(1,2); // Mostra missatge OK
203     printf(lcd_putc,"OK");
204     delay_ms(200);
205 }
206 if(!input(pin_A3)==0){ // Sortir
207     C=0;
208     DataLong=0;
209     delay_ms(500);
210 }
211 }
212 }
213 }
214 }
215 if(A==1){ //
216     printf(lcd_putc,"%f"); // Menu Secundari
217     lcd_gotoxy(1,1); // Segona

```

```

218     printf(lcd_putc,"Temperatura"); // Opció
219     lcd_gotoxy(1,2); //
220     printf(lcd_putc,"Nivell <="); //
221     delay_ms(200); //
222 }
223 if(!input(pin_A2)==0){ // Entrar
224     C=1;
225     delay_ms(500);
226 }
227 while(C==1){ // Visualització Nivell
228     {
229         putc(0x7E); // Demana dada de nivell
230         putc(0x00); // Es configura en funció del dispositiu
231         putc(0x10);
232         putc(0x17);
233         putc(0x01);
234         putc(0x02);
235         putc(0x13);
236         putc(0x22);
237         putc(0x00);
238         putc(0x40);
239         putc(0x0B);
240         putc(0x78);
241         putc(0x4E);
242         putc(0xFF);
243         putc(0xFE);
244         putc(0x00);
245         putc(0x49);
246         putc(0x53);
247         putc(0x00);
248     }

```

```

249         putc(0x08);
250
251         // Marca la interrupció com a forçada
252
253         level=calc_To_Print(DataLong,2);           // Calcula el valor a mostrar
254
255         printf(lcd_putc,"%f");                    // Mostra
256         lcd_gotoxy(1,1);                          // Dada
257         printf(lcd_putc,"Nivell: %f ",level);     // Nivell
258         delay_ms(200);                            //
259
260         if(level<=25){                            // Mostra missatge d'alarma
261             lcd_gotoxy(1,2);                      // Si el nivell és inferior al 25%
262             printf(lcd_putc,"ALARMA");           //
263             delay_ms(200);                        //
264         }
265         else{
266             lcd_gotoxy(1,2);                      // Mostra el missatge OK
267             printf(lcd_putc,"OK");               //
268             delay_ms(200);                        //
269         }
270         if(!input(pin_A3)==0){                   // Sortir
271             C=0;
272             DataLong=0;
273             delay_ms(500);
274         }
275     }
276 }
277
278
279 if(A==1){ //

```

```

280     printf(lcd_putc,"%f"); //
281     lcd_gotoxy(1,1);      // Menu Principal
282     printf(lcd_putc,"Actual"); // Segona
283     lcd_gotoxy(1,2);     // Opció
284     printf(lcd_putc,"Historic <="); //
285     delay_ms(200);       //
286     if(!input(pin_A2)==0){ // Entrar
287         B=1;
288         A=2;
289         delay_ms(500);
290     }
291     while(B==1){
292         if(!input(pin_A1)==0){ // Pujar
293             A=2;
294             delay_ms(200);
295         }
296         if(!input(pin_A0)==0){ // Baixar
297             A=1;
298             delay_ms(200);
299         }
300         if(!input(pin_A3)==0){ // Sortir
301             B=0;
302             A=1;
303             delay_ms(500);
304         }
305         if(A==2){ //
306             printf(lcd_putc,"%f"); // Menú Secundari
307             lcd_gotoxy(1,1);      //
308             printf(lcd_putc,"Temperatura <="); // Primera
309             lcd_gotoxy(1,2);     //
310             printf(lcd_putc,"Nivell"); // Opció

```

```

311         delay_ms(200); //
312
313         if(!input(pin_A2)==0){ // Entrar
314             C=1;
315             A=0;
316             delay_ms(500);
317         }
318         while(C==1){ // TEMPERATURA
319
320             pos_H_Temp=READ_EEPROM(0); // Llegeix la posició
321             pos_L_Temp=READ_EEPROM(1); //
322             posLong=make16(pos_H_Temp,pos_L_Temp); // de la memòria
323
324             if(posLong==2){ // Torna la memòria al final
325                 posLong=512; // per seguir llegint dades
326             }
327             Data_H2=READ_EEPROM(posLong-2); // Llegeix 2 bytes
328             Data_L2=READ_EEPROM(posLong-1); //
329
330             posLong=posLong-2; // Posiciona el següent byte
331
332             if(posLong==2){ // Torna la memòria al final
333                 posLong=512; // per seguir llegint dades
334             }
335
336             Data_H3=READ_EEPROM(posLong-2); // Llegeix 2 bytes
337             Data_L3=READ_EEPROM(posLong-1); //
338             posLong=posLong-2; // Posiciona el següent byte
339         }else{
340             Data_H3=READ_EEPROM(posLong-2); // Llegeix 2 bytes
341             Data_L3=READ_EEPROM(posLong-1); //

```

```

342         posLong=posLong-2; // Posiciona el següent byte
343     }
344     DataLong2=make16(Data_L2,Data_H2); // Genera la mostra de 10 bits
345     tempC=calc_To_Print(DataLong2,1); // Calcula el valor de la primera mostra a visualitzar
346     DataLong3=make16(Data_L3,Data_H3); // Genera la mostra de 10 bits
347     tempC1=calc_To_Print(DataLong3,1); // Calcula el valor de la segona mostra a visualitzar
348     if(tempC<100){ //
349         printf(lcd_putc,"%f"); //
350         lcd_gotoxy(1,1); // Visualització
351         printf(lcd_putc,"30 min: ----"); //
352         lcd_gotoxy(1,2); // si la memòria és buida
353         printf(lcd_putc,"1 h: ----"); //
354         delay_ms(200); //
355     }
356     else{
357         if(tempC1>100){
358             printf(lcd_putc,"%f"); //
359             lcd_gotoxy(1,1); // Visualització
360             printf(lcd_putc,"30 min: %f C",tempC); //
361             lcd_gotoxy(1,2); // si només la segona mostra és buida
362             printf(lcd_putc,"1 h: ----"); //
363             delay_ms(200); //
364         }
365         else{
366             printf(lcd_putc,"%f"); // Visualització
367             lcd_gotoxy(1,1); //
368             printf(lcd_putc,"30 min: %f C",tempC); // de les dos últimes mostres
369             lcd_gotoxy(1,2); //
370             printf(lcd_putc,"1 h: %f C",tempC1); // de temperatura
371             delay_ms(200); //
372         }

```

```

373     }
374     while (A==0) { // Flag
375         if (!input(pin_A0)==0) { // Espera polsador per avançar
376             A=1; // Espera polsador per avançar
377             delay_ms(200); // Espera polsador per avançar
378         }
379     }
380     A=0; // Reinicia flag
381
382     if (posLong==2) { // Torna la memòria al final
383         posLong=512; // per seguir llegint dades
384     }
385     Data_H2=READ_EEPROM(posLong-2); // Llegeix 2 bytes
386     Data_L2=READ_EEPROM(posLong-1); //
387
388     posLong=posLong-2; // Posiciona el següent byte
389
390     if (posLong==2) { // Torna la memòria al final
391         posLong=512; // per seguir llegint dades
392     }
393     Data_H3=READ_EEPROM(posLong-2); // Llegeix 2 bytes
394     Data_L3=READ_EEPROM(posLong-1); //
395     posLong=posLong-2; // Posiciona el següent byte
396 } else {
397     Data_H3=READ_EEPROM(posLong-2); // Llegeix 2 bytes
398     Data_L3=READ_EEPROM(posLong-1); //
399
400     posLong=posLong-2; // Posiciona el següent byte
401 }
402 DataLong2=make16(Data_L2,Data_H2); // Genera la mostra de 16 bits
403 tempC=calc_To_Print(DataLong2,1); // Calcula el valor de la primera mostra a visualitzar

```

```

404 DataLong3=make16(Data_L3,Data_H3); // Genera la mostra de 16 bits
405 tempC1=calc_To_Print(DataLong3,1); // Calcula el valor de la segona mostra a visualitzar
406 if (tempC<100) {
407     printf(lcd_putc,"%f"); //
408     lcd_gotoxy(1,1); // Visualització
409     printf(lcd_putc,"1.5:  ---"); //
410     lcd_gotoxy(1,2); // si la memòria és buida
411     printf(lcd_putc,"2 h:  ---"); //
412     delay_ms(200); //
413 }
414 else {
415     if (tempC1>100) {
416         printf(lcd_putc,"%f"); //
417         lcd_gotoxy(1,1); // Visualització
418         printf(lcd_putc,"1.5 h: %f C",tempC); //
419         lcd_gotoxy(1,2); // si només la segona mostra és buida
420         printf(lcd_putc,"2 h:  ---"); //
421         delay_ms(200); //
422     }
423     else {
424         printf(lcd_putc,"%f"); // Visualització
425         lcd_gotoxy(1,1); //
426         printf(lcd_putc,"1.5 h: %f C",tempC); // de la 3era i 4a últimes mostres
427         lcd_gotoxy(1,2); //
428         printf(lcd_putc,"2 h: %f C",tempC1); // de temperatura
429         delay_ms(200); //
430     }
431 }
432 while (A==0) { // Flag
433     if (!input(pin_A0)==0) { // Espera polsador per avançar
434         A=1; // Espera polsador per avançar

```

```

435         delay_ms(200); //
436     }
437     A=0; // Reinicia flag
438
439     if (posLong==2) { // Torna la memòria al final
440         posLong=512; // per seguir llegint dades
441     }
442     Data_H2=READ_EEPROM(posLong-2); // Llegeix 2 bytes
443     Data_L2=READ_EEPROM(posLong-1); //
444
445     posLong=posLong-2; // Posiciona el següent byte
446
447     if (posLong==2) { // Torna la memòria al final
448         posLong=512; // per seguir llegint dades
449     }
450     Data_H3=READ_EEPROM(posLong-2); // Llegeix 2 bytes
451     Data_L3=READ_EEPROM(posLong-1); //
452     posLong=posLong-2; // Posiciona el següent byte
453 } else {
454     Data_H3=READ_EEPROM(posLong-2); // Llegeix 2 bytes
455     Data_L3=READ_EEPROM(posLong-1); //
456
457     posLong=posLong-2; // Posiciona el següent byte
458
459     DataLong2=make16(Data_L2,Data_H2); // Genera la mostra de 16 bits
460     tempC=calc_To_Print(DataLong2,1); // Calcula el valor de la primera mostra a visualitzar
461     DataLong3=make16(Data_L3,Data_H3); // Genera la mostra de 16 bits
462     tempC1=calc_To_Print(DataLong3,1); // Calcula el valor de la segona mostra a visualitzar
463     if (tempC<100) {
464         printf(lcd_putc,"%f"); //

```

```

465         lcd_gotoxy(1,1); // Visualització
466         printf(lcd_putc,"2.5 h:  ---"); // si la memòria és buida
467         lcd_gotoxy(1,2); //
468         printf(lcd_putc,"3 h:  ---"); //
469         delay_ms(200); //
470     }
471     else {
472         if (tempC1>100) {
473             printf(lcd_putc,"%f"); //
474             lcd_gotoxy(1,1); // Visualització
475             printf(lcd_putc,"2.5 h: %f C",tempC); //
476             lcd_gotoxy(1,2); // si només la segona mostra és buida
477             printf(lcd_putc,"3 h:  ---"); //
478             delay_ms(200); //
479         }
480         else {
481             printf(lcd_putc,"%f"); // Visualització
482             lcd_gotoxy(1,1); //
483             printf(lcd_putc,"2.5 h: %f C",tempC); // de les 5a i 6a últimes mostres
484             lcd_gotoxy(1,2); //
485             printf(lcd_putc,"3 h: %f C",tempC1); // de temperatura
486             delay_ms(200); //
487         }
488     }
489 }
490 while (A==0) { // Flag
491     if (!input(pin_A0)==0) { // Espera polsador per avançar
492         A=1; // Espera polsador per avançar
493         delay_ms(200); //
494     }
495     A=0; // Reinicia flag

```



```

497
498     if(posLong==2){ // Torna la memòria al final
499         posLong=512; // per seguir llegint dades
500     }
501     Data_H2=READ_EEPROM(posLong-2); // Llegeix 2 bytes
502     Data_L2=READ_EEPROM(posLong-1); //
503
504     posLong=posLong-2; // Posiciona el següent byte
505
506     if(posLong==2){ // Torna la memòria al final
507         posLong=512; // per seguir llegint dades
508     }
509     Data_H3=READ_EEPROM(posLong-2); // Llegeix 2 bytes
510     Data_L3=READ_EEPROM(posLong-1); //
511     posLong=posLong-2; // Posiciona el següent byte
512 }else{
513     Data_H3=READ_EEPROM(posLong-2); // Llegeix 2 bytes
514     Data_L3=READ_EEPROM(posLong-1); //
515
516     posLong=posLong-2; // Posiciona el següent byte
517 }
518 DataLong2=make16(Data_L2,Data_H2); // Genera la mostra de 10 bits
519 tempC=calc_To_Print(DataLong2,1); // Calcula el valor de la primera mostra a visualitzar
520 DataLong3=make16(Data_L3,Data_H3); // Genera la mostra de 10 bits
521 tempC1=calc_To_Print(DataLong3,1); // Calcula el valor de la segona mostra a visualitzar
522 if(tempC>100){ //
523     printf(lcd_putc,"%f"); //
524     lcd_gotoxy(1,1); // Visualització
525     printf(lcd_putc,"3.5 h: ----"); //
526     lcd_gotoxy(1,2); // si la memòria és buida
527     printf(lcd_putc,"4 h: ----"); //

```

```

528     delay_ms(200); //
529 }
530 else{
531     if(tempC1>100){ //
532         printf(lcd_putc,"%f"); // Visualització
533         lcd_gotoxy(1,1); //
534         printf(lcd_putc,"3.5 h: %f C",tempC); //
535         lcd_gotoxy(1,2); // si només la segona mostra és buida
536         printf(lcd_putc,"4 h: ----"); //
537         delay_ms(200); //
538     }
539     else{
540         printf(lcd_putc,"%f"); // Visualització
541         lcd_gotoxy(1,1); //
542         printf(lcd_putc,"3.5 h: %f C",tempC); // de les 7a i 8a últimes mostres
543         lcd_gotoxy(1,2); //
544         printf(lcd_putc,"4 h: %f C",tempC1); // de temperatura
545         delay_ms(200); //
546     }
547 }
548 while(A==0){ //
549     if(!input(pin_A3)==0){ // Sortir
550         A=2; //
551         C=0; //
552         delay_ms(500); //
553     }
554 }
555 }
556 if(A==1){
557     printf(lcd_putc,"%f"); // Menú Secundari
558 }

```

```

559     lcd_gotoxy(1,1); //
560     printf(lcd_putc,"Temperatura"); // Segona
561     lcd_gotoxy(1,2); //
562     printf(lcd_putc,"Nivell <="); // Opció
563     delay_ms(200); //
564
565     if(!input(pin_A2)==0){ // Entrar
566         C=1; //
567         A=0; //
568
569         delay_ms(500); //
570     }
571     while(C==1){ // NIVELL
572
573         pos_H_Lev=READ_EEPROM(512); // Llegeix la posició
574         pos_L_Lev=READ_EEPROM(513); //
575         posLong=make16(pos_H_Lev,pos_L_Lev); // de la memòria
576
577         if(posLong==514){ // Torna la memòria al final
578             posLong=1024; // per seguir llegint dades
579         }
580         Data_H2=READ_EEPROM(posLong-2); // Llegeix 2 bytes
581         Data_L2=READ_EEPROM(posLong-1); //
582
583         posLong=posLong-2; // Posiciona el següent byte
584
585         if(posLong==514){ // Torna la memòria al final
586             posLong=1024; // per seguir llegint dades
587         }
588         Data_H3=READ_EEPROM(posLong-2); // Llegeix 2 bytes
589         Data_L3=READ_EEPROM(posLong-1); //

```

```

590         posLong=posLong-2; //Posiciona el següent byte
591     }
592 }else{
593     Data_H3=READ_EEPROM(posLong-2); // Llegeix 2 bytes
594     Data_L3=READ_EEPROM(posLong-1); //
595
596     posLong=posLong-2; //Posiciona el següent byte
597 }
598 }
599
600 DataLong2=make16(Data_L2,Data_H2); // Genera la mostra de 10 bits
601 level=calc_To_Print(DataLong2,2); // Calcula el valor de la primera mostra a visualitzar
602 DataLong3=make16(Data_L3,Data_H3); // Genera la mostra de 10 bits
603 level1=calc_To_Print(DataLong3,2); // Calcula el valor de la primera mostra a visualitzar
604
605 if(level>100){ //
606     printf(lcd_putc,"%f"); //
607     lcd_gotoxy(1,1); // Visualització
608     printf(lcd_putc,"30 min: ----"); //
609     lcd_gotoxy(1,2); // si la memòria és buida
610     printf(lcd_putc,"1 h: ----"); //
611     delay_ms(200); //
612 }
613 else{
614     if(level1>100){ //
615         printf(lcd_putc,"%f"); //
616         lcd_gotoxy(1,1); // Visualització
617         printf(lcd_putc,"30 min: %f ",level); //
618         lcd_gotoxy(1,2); // si només la segona mostra és buida
619         printf(lcd_putc,"1 h: ----"); //
620         delay_ms(200); //

```

```

621     }
622     else{
623         printf(lcd_putc, "\f"); // Visualització
624         lcd_gotoxy(1,1); // Visualització
625         printf(lcd_putc, "30 min: %f ",level); // de les dues últimes mostres
626         lcd_gotoxy(1,2); // de les dues últimes mostres
627         printf(lcd_putc, "1 h: %f ",level); // de nivell
628         delay_ms(200); // de nivell
629     }
630 }
631 while (A==0) { // Flag
632     if (!input(pin_A0)==0) { // Espera polsador per avançar
633         A=1; // Espera polsador per avançar
634         delay_ms(200); // Espera polsador per avançar
635     }
636 }
637 A=0; // Reinicia flag
638 }
639 if (posLong==514) { // Torna la memòria al final
640     posLong=1024; // per seguir llegint dades
641 }
642 Data_H2=READ_EEPROM(posLong-2); // Llegeix 2 bytes
643 Data_L2=READ_EEPROM(posLong-1); // Llegeix 2 bytes
644 }
645 posLong=posLong-2; // Posiciona el següent byte
646 }
647 if (posLong==514) { // Torna la memòria al final
648     posLong=1024; // per seguir llegint dades
649 }
650 Data_H3=READ_EEPROM(posLong-2); // Llegeix 2 bytes
651 Data_L3=READ_EEPROM(posLong-1); // Llegeix 2 bytes
652 }
653 posLong=posLong-2; // Posiciona el següent byte
654 }
655 } else{
656     Data_H3=READ_EEPROM(posLong-2); // Llegeix 2 bytes
657     Data_L3=READ_EEPROM(posLong-1); // Llegeix 2 bytes
658 }
659 posLong=posLong-2; // Posiciona el següent byte
660 }
661 }
662 DataLong2=makel6(Data_L2,Data_H2); // Genera la mostra de 10 bits
663 level=calc_To_Print(DataLong2,2); // Calcula el valor de la primera mostra a visualitzar
664 DataLong3=makel6(Data_L3,Data_H3); // Genera la mostra de 10 bits
665 level1=calc_To_Print(DataLong3,2); // Calcula el valor de la primera mostra a visualitzar
666 }
667 if (level>100) {
668     printf(lcd_putc, "\f"); // Visualització
669     lcd_gotoxy(1,1); // Visualització
670     printf(lcd_putc, "1.5 h: ----"); // si la memòria és buida
671     lcd_gotoxy(1,2); // si la memòria és buida
672     printf(lcd_putc, "2 h: ----"); // si la memòria és buida
673     delay_ms(200); // si la memòria és buida
674 }
675 } else{
676     if (level>100) {
677         printf(lcd_putc, "\f"); // Visualització
678         lcd_gotoxy(1,1); // Visualització
679         printf(lcd_putc, "1.5 h: %f ",level); // si només la segona mostra és buida
680         lcd_gotoxy(1,2); // si només la segona mostra és buida
681         printf(lcd_putc, "2 h: ----"); // si només la segona mostra és buida
682         delay_ms(200); // si només la segona mostra és buida
683     }
684 } else{
685     printf(lcd_putc, "\f"); // Visualització
686     lcd_gotoxy(1,1); // Visualització
687     printf(lcd_putc, "1.5 h: %f ",level); // de les dues últimes mostres
688     lcd_gotoxy(1,2); // de les dues últimes mostres
689     printf(lcd_putc, "2 h: %f ",level); // de nivell
690     delay_ms(200); // de nivell
691 }
692 }
693 while (A==0) { // Flag
694     if (!input(pin_A0)==0) { // Espera polsador per avançar
695         A=1; // Espera polsador per avançar
696         delay_ms(200); // Espera polsador per avançar
697     }
698 }
699 A=0; // Reinicia flag
700 }
701 if (posLong==514) { // Torna la memòria al final
702     posLong=1024; // per seguir llegint dades
703 }
704 Data_H2=READ_EEPROM(posLong-2); // Llegeix 2 bytes
705 Data_L2=READ_EEPROM(posLong-1); // Llegeix 2 bytes
706 }
707 posLong=posLong-2; // Posiciona el següent byte
708 }
709 if (posLong==514) { // Torna la memòria al final
710     posLong=1024; // per seguir llegint dades
711 }
712 Data_H3=READ_EEPROM(posLong-2); // Llegeix 2 bytes
713 Data_L3=READ_EEPROM(posLong-1); // Llegeix 2 bytes
714 }
715 posLong=posLong-2; // Posiciona el següent byte
716 }
717 } else{
718     Data_H3=READ_EEPROM(posLong-2); // Llegeix 2 bytes
719     Data_L3=READ_EEPROM(posLong-1); // Llegeix 2 bytes
720 }
721 posLong=posLong-2; // Posiciona el següent byte
722 }
723 }
724 DataLong2=makel6(Data_L2,Data_H2); // Genera la mostra de 10 bits
725 level=calc_To_Print(DataLong2,2); // Calcula el valor de la primera mostra a visualitzar
726 DataLong3=makel6(Data_L3,Data_H3); // Genera la mostra de 10 bits
727 level1=calc_To_Print(DataLong3,2); // Calcula el valor de la primera mostra a visualitzar
728 }
729 if (level>100) {
730     printf(lcd_putc, "\f"); // Visualització
731     lcd_gotoxy(1,1); // Visualització
732     printf(lcd_putc, "2.5 h: ----"); // si la memòria és buida
733     lcd_gotoxy(1,2); // si la memòria és buida
734     printf(lcd_putc, "3 h: ----"); // si la memòria és buida
735     delay_ms(200); // si la memòria és buida
736 }
737 } else{
738     if (level>100) {
739         printf(lcd_putc, "\f"); // Visualització
740         lcd_gotoxy(1,1); // Visualització
741         printf(lcd_putc, "2.5 h: %f ",level); // si només la segona mostra és buida
742         lcd_gotoxy(1,2); // si només la segona mostra és buida
743         printf(lcd_putc, "3 h: ----"); // si només la segona mostra és buida
744         delay_ms(200); // si només la segona mostra és buida
745     }
746 } else{
747     printf(lcd_putc, "\f"); // Visualització
748     lcd_gotoxy(1,1); // Visualització
749     printf(lcd_putc, "2.5 h: %f ",level); // de les dues últimes mostres
750     lcd_gotoxy(1,2); // de les dues últimes mostres
751     printf(lcd_putc, "3 h: %f ",level); // de nivell
752     delay_ms(200); // de nivell
753 }
754 }

```

```

745     }
746     else{
747         printf(lcd_putc, "\f"); // Visualització
748         lcd_gotoxy(1,1);
749         printf(lcd_putc, "2.5 h: %f ", level); // de les dues últimes mostres
750         lcd_gotoxy(1,2);
751         printf(lcd_putc, "3 h: %f ", level); // de nivell
752         delay_ms(200);
753     }
754 }
755 while (A==0) { // Flag
756     if (!input(pin_A0)==0) { // Espera pulsador per avançar
757         A=1;
758         delay_ms(200);
759     }
760 }
761 A=0; // Reinicia flag
762
763 if (posLong==514) { // Torna la memòria al final
764     posLong=1024; // per seguir llegint dades
765 }
766 Data_H2=READ_EEFROM(posLong-2); // Llegeix 2 bytes
767 Data_L2=READ_EEFROM(posLong-1); //
768
769 posLong=posLong-2; // Posiciona el següent byte
770
771 if (posLong==514) { // Torna la memòria al final
772     posLong=1024; // per seguir llegint dades
773 }
774 Data_H3=READ_EEFROM(posLong-2); // Llegeix 2 bytes
775 Data_L3=READ_EEFROM(posLong-1); //
776
777 posLong=posLong-2; // Posiciona el següent byte
778
779 } else {
780     Data_H3=READ_EEFROM(posLong-2); // Llegeix 2 bytes
781     Data_L3=READ_EEFROM(posLong-1); //
782
783     posLong=posLong-2; // Posiciona el següent byte
784
785 }
786 DataLong2=make16(Data_L2, Data_H2); // Genera la mostra de 10 bits
787 level=calc_To_Print(DataLong2); // Calcula el valor de la primera mostra a visualitzar
788 DataLong3=make16(Data_L3, Data_H3); // Genera la mostra de 10 bits
789 level=calc_To_Print(DataLong3); // Calcula el valor de la primera mostra a visualitzar
790
791 if (level>100) {
792     printf(lcd_putc, "\f"); // Visualització
793     lcd_gotoxy(1,1);
794     printf(lcd_putc, "3.5 h: ----"); // si la memòria és buida
795     lcd_gotoxy(1,2);
796     printf(lcd_putc, "4 h: ----"); //
797     delay_ms(200);
798 }
799 else {
800     if (level>100) {
801         printf(lcd_putc, "\f"); // Visualització
802         lcd_gotoxy(1,1);
803         printf(lcd_putc, "3.5 h: %f ", level); // si només la segona mostra és buida
804         lcd_gotoxy(1,2);
805         printf(lcd_putc, "4 h: ----"); //
806         delay_ms(200);
807     }
808     else {
809         printf(lcd_putc, "\f"); // Visualització
810         lcd_gotoxy(1,1);
811         printf(lcd_putc, "3.5 h: %f ", level); // de les dues últimes mostres
812         lcd_gotoxy(1,2);
813         printf(lcd_putc, "4 h: %f ", level); // de nivell
814         delay_ms(200);
815     }
816 }
817 while (A==0) { // Sortir
818     if (!input(pin_A3)==0) { //
819         A=1; //
820         C=0; //
821         delay_ms(500); //
822     }
823 }
824 }
825 }
826 }
827 }
828 }
829 }
830 }
831
832 float calc_To_Print(int16 data_Long, int device) {
833     float value, l_V, t_V;
834     if (device==1) {
835         t_V=(data_Long*3.3)/1024; // La mostra rebuda es passa a voltatge
836         value=(t_V*100)/3.3; // La mostra rebuda es passa a temperatura
837     }
838 }
839
840 if (device==2) {
841     l_V=(data_Long*3.3)/1024; // La mostra rebuda es passa a voltatge
842     if (l_V<3.3) { //
843         value=100; // Retorna el percentatge de nivell
844         if (l_V<1.91) { //
845             value=75; //
846             if (l_V<1.74) { // 100, 75, 50, 25, 0
847                 value=50; //
848                 if (l_V<1.53) { //
849                     value=25; //
850                     if (l_V<1.3) { //
851                         value=0; //
852                     }
853                 }
854             }
855         }
856     } else { // Retorna un valor fora del rang
857         value=150; //
858     }
859 }
860 return value;

```