

Meteo – estació autònoma amb sensors

Jordi Fenés Castell

“Enginyeria Tècnica en Telecomunicació, especialitat Telemàtica” (E.T.T.T.)

Jordi Bécares Ferrés

Consultor

11 de Juny de 2013

Copyright © 2013 Jordi Fenés Castell

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.

A copy of the license is included in the section entitled "GNU Free Documentation License".

Dedicatòria i agraïments:

Dedico aquest projecte a totes les persones interessades en la meteorologia, en la informació mesurable i analitzable que es pot obtenir d'aquesta ciència, que és de molta utilitat a un gran nombre de persones en el seu dia a dia.

En primer terme, vull agrair al consultor Jordi Bécares el seu seguiment de l'evolució del projecte, i sobre tot els ànims rebuts, els seus comentaris, observacions i recomanacions. Tot plegat, ha estat de gran ajuda i orientació per tal de tirar endavant el desenvolupament del projecte.

També vull agrair tot el suport rebut per part de familiars, amics i coneguts als qui he explicat el meu projecte de final de carrera. Sense el seu puntal, la seva presència i ànims rebuts, en els moments en que no tot funciona com un vol, de segur, que a hores d'ara estaria perdut, en el decurs d'aquesta empresa.

Per tant, moltes gràcies a tots !

Resum:

Aquest treball, emmarcat dins de l'àrea de sistemes encastats, té per objectiu, la construcció d'una **estació meteorològica de baix consum** que reporta periòdicament les dades de lectures preses, a un servidor i es poden consultar. Per que el dispositiu sigui autònom quant a energia, es dota amb sistemes de baix consum (es posa el μC^1 en mode 'Idle') i per altra banda es fa autosuficient, al afegir un panell solar i una bateria.

La part central del disseny es basa en un sistema microcontrolador LPC1769² que funciona amb un processador ARM-Cortex³ M3 de 120 MHz i ofereix un elevat nombre de ports I/O⁴. Té un cost reduït i es pot programar fàcilment amb un entorn de desenvolupament i llibreries, facilitat per el mateix fabricant.

El nostre microcontrolador està connectat i governa, a un mòdul WiFly⁵ i un mòdul de sensors. El primer, permet la connexió a la xarxa via Wi-Fi⁶, i el segon pren les lectures dels sensors que te incorporats. D'aquesta forma, el nostre dispositiu, cada interval de temps establert, recull la informació generada pels diferents sensors, li dona el format convenient i la fa arribar a un servidor web (local o a Internet) per tal d'emmagatzemar-la en una Base de Dades. Posteriorment, aquesta informació pot ser tractada per qualsevol altra aplicació o reenviada a webs meteorològiques.

Per tal de facilitar la configuració del sistema, es permet una connexió remota al dispositiu, que emprant uns menús específics es poden fer els canvis oportuns de : temps de presa de lectures, altres modes de treball o reinici del sistema.

¹ Acrònim de la paraula Microcontrolador

² Dispositiu fabricat per "NXP Semiconductors N.V." per treballar amb sistemes encastats.

³ Unitat central de procés (CPU) produïda per ARM : <http://www.arm.com>

⁴ De l'anglès "Input / Output", fa referència a ports d'entrada – sortida.

⁵ Placa electrònica amb CPU, Wi-Fi, port de comunicació i pins de funcions

⁶ De l'anglès, es una marca, i fa referència al mecanisme de connexió sense fils de dispositius electrònics

Índex de continguts:

1.	Introducció.....	7
1.1.	Justificació.....	7
1.2.	Descripció del projecte.....	8
1.3.	Objectius del projecte.....	9
1.4.	Enfocament i mètode seguit.....	11
1.5.	Planificació del projecte.....	12
1.6.	Recursos emprats.....	14
1.6.1.	Maquinari.....	14
1.6.2.	Programari.....	17
1.7.	Productes obtinguts.....	19
1.8.	Descripció dels altres capítols.....	21
2.	Antecedents.....	21
2.1.	Estat de l'art.....	22
2.2.	Estudi de mercat.....	27
3.	Descripció funcional.....	28
3.1.	Sistema de l'estació meteorològica.....	29
3.1.1.	Diagrama de blocs.....	30
3.1.2.	Interacció amb la xarxa.....	31
3.1.3.	Interacció dels diferents objectes del sistema.....	32
3.2.	Disseny interfície d'usuari.....	33
3.3.	Disseny aplicació al sistema encastat.....	34
4.	Descripció detallada del sistema.....	35
4.1.	Creació de l'estació meteorològica.....	36
4.1.1.	Implementar controladors al sistema LPC1769.....	36
4.1.2.	Construcció del prototip de l'aplicació.....	39
4.1.3.	Creació de l'aplicació del sistema.....	41
4.2.	Creació servidor Web amb Base de Dades.....	44
4.3.	Oferir connexió de configuració remota.....	45
5.	Viabilitat tècnica.....	46
6.	Valoració econòmica.....	48
7.	Conclusions.....	49
7.1.	Exposició de les conclusions.....	50
7.2.	Proposta de millores.....	50
7.3.	Autoavaluació.....	51
8.	Glossari.....	51
9.	Bibliografia.....	53
10.	Annexos.....	54
10.1.	Manual d'usuari.....	54
10.2.	Execució, compilació i càrrega.....	57

Índex de figures:

<i>Figura 1 : Escenari de treball.....</i>	9
<i>Figura 2 : Planificació inicial.....</i>	12
<i>Figura 3 : Planificació final.....</i>	13
<i>Figura 4 : microcontrolador LPC1769.....</i>	15
<i>Figura 5 : Mòdul WiFly.....</i>	15
<i>Figura 6 : Adaptador USB - sèrie CP2102.....</i>	15
<i>Figura 7 : Mòdul de sensors (USB Weather V3).....</i>	16
<i>Figura 8 : Placa solar amb bateria recarregable.....</i>	16
<i>Figura 9 : Logo del LPCXpresso.....</i>	17
<i>Figura 10 : Logo de FreeRTOS.....</i>	17
<i>Figura 11 : Logo de Microsoft Visual Studio 2008.....</i>	18
<i>Figura 12 : Logo de Windows 7.....</i>	18
<i>Figura 13 : Estació meteorològica.....</i>	19
<i>Figura 14 : Entrada manual de dades.....</i>	20
<i>Figura 15 : Enviament de dades correcte.....</i>	20
<i>Figura 16 : Taula interès vehicles amb sistemes encastats.....</i>	22
<i>Figura 17 : Placa de Arduino Duemilanove.....</i>	23
<i>Figura 18 : Placa de Raspberry pi Model B.....</i>	24
<i>Figura 19 : Placa LPC1769 i components.....</i>	25
<i>Figura 20 : Taula comparació tecnologia sense fils.....</i>	26
<i>Figura 21 : Estacions meteorològiques d'us particular i professional.....</i>	27
<i>Figura 22 : Diagrama de blocs de l'estació meteorològica.....</i>	30
<i>Figura 23 : Dades emmagatzemades a la DB.....</i>	33
<i>Figura 24 : Transicions d'estat de tasques a FreeRTOS.....</i>	34
<i>Figura 25 : Combinació FreeRTOS i Llibreria maquinari.....</i>	35
<i>Figura 26 : Diagrama de blocs d'aplicació de test.....</i>	37
<i>Figura 27 : Diagrama de flux d'aplicació de test.....</i>	38
<i>Figura 28: Pantalla configuració mòdul de sensors.....</i>	39
<i>Figura 29 : Diagrama de blocs aplicació Station.....</i>	41
<i>Figura 30 : Diagrama de flux aplicació Station.....</i>	43
<i>Figura 31 : Configuració del Servidor Web.....</i>	44
<i>Figura 32 : Pantalla configuració remota aplicació.....</i>	45
<i>Figura 33 : Taula de consums placa LPC1769.....</i>	46
<i>Figura 34 : Taula dades tècniques - requeriments mòdul sensors.....</i>	47
<i>Figura 35 : Taula de costos associats al projecte.....</i>	49

1. Introducció.

En aquest treball final de carrera de l'àrea de sistemes encastats, s'exposa la construcció d'una **estació meteorològica de baix consum**, que reporta periòdicament les lectures preses a uns sensors, cap a un servidor web, que les emmagatzema i en permet la consulta. D'altra banda, per millorar l'autonomia energètica del dispositiu, s'implementa sistemes de baix consum (es posa el μ C en mode 'Idle') i es fa autosuficient, al afegir un panell solar i una bateria.

Per a la seva realització, es farà servir el microcontrolador LPC1769, connectat a un mòdul WiFly i a un mòdul de sensors. El dispositiu, serà programat en llenguatge C, amb el software proporcionat pel mateix fabricant NXP⁷, concretament, emprarem el IDE⁸ LPCXpresso i la llibreria CMSIS v2. I això, es combina amb el O.S.⁹ FreeRTOS, que farà de nucli central, per a la gestió dels processos del sistema desenvolupat.

De primer, es desenvolupa una llibreria, amb un driver de Debug/Log¹⁰ del sistema, per poder controlar el passos que seguirà l'aplicació, i posteriorment, altres drivers, per tal de controlar els recursos i perifèrics o mòduls de que es compona el nostre sistema. Un cop finalitzada la llibreria de drivers, s'utilitza totes les funcions necessàries d'aquesta, per tal de crear una aplicació que s'ajusti als requisits, explicats al inici de l'apartat.

Com que el dispositiu que es crea, es tracta d'un sistema que genera informació, aquesta serà emmagatzemada i es podrà consultar. Per tant, també es desenvolupa un Web connectat a una Base de Dades, que realitza aquesta funció, amb la informació rebuda del nostre dispositiu, via connexió Wi-Fi.

L'abast de coneixements necessari, per a poder realitzar aquest treball, és el d'electrònica bàsica, el llenguatge C de programació, el funcionament i creació de pàgines Web, nocions de Bases de dades i també coneixements bàsics de sistemes encastats. En quant al termini per a la seva realització, el límit temporal per realitzar aquest projecte, ha estat aproximadament d'uns 3 mesos a partir del dia 28/02/2013.

1.1. Justificació.

Principalment, s'ha escollit aquest treball final de carrera, perquè hem considerat que era força interessant el procediment emprat i la solució obtinguda. A més, els dispositius i sistemes emprats, son molt representatius de l'entorn de sistemes encastats. Una motivació extra, ha estat el interès general o de moltes persones, que hi ha per les dades meteorològiques.

⁷ NXP Semiconductors N.V., companyia fundada per PHILIPS : <http://www.nxp.com>

⁸ De l'anglès "Integrated Development Environment", Entorn de desenvolupament integrat

⁹ De l'anglès "Operating System", Sistema Operatiu

¹⁰ Procediment que permet analitzar, depurar i registrar els passos d'una aplicació informàtica

Si fem un estudi del mercat actual, sobre estacions meteorològiques, podem veure que està inundat per dispositius d'aquest tipus. La majoria dels de baix cost, a pesar de que ofereixen bastant informació, són de fiabilitat dubtosa, funcionen només amb piles, alguns ofereixen connexió via cablejat, solen estar suportats només pel mateix fabricant i solen emprar components electrònics obsolets o de difícil localització. Per altra banda, els de cost superior, encara que ofereixen millors prestacions, més qualitat i connectivitat, els components materials i electrònics només els sol proporcionar el fabricant. En la majoria dels dos casos anteriors, es tracta d'un producte tancat, sense possibilitats de creixement o ampliació per part de l'usuari final.

Amb aquest TFC¹¹, es vol mitigar i superar les mancances detectades en els sistemes que estan actualment en el mercat. Pel que es farà servir components electrònics actuals, de fàcil localització en el mercat i amb molta informació sobre el seu funcionament i prestacions. D'aquesta forma, el dispositiu final, disposarà d'informació actual i no fa servir material obsolet o vinculat només a un únic fabricant. També, al dotar al sistema d'autosuficiència, amb una placa solar i una bateria, s'elimina el problema d'anar controlant l'estat de les piles, amb les que solen funcionar la majoria.

Per altra banda, s'ofereix la programació en obert, amb llicència GPL¹² de GNU, és a dir, es dona lliurement, el codi desenvolupat per al funcionament de tot el sistema amb el dispositiu. D'aquesta forma, l'usuari final del dispositiu, pot millorar, ampliar o adaptar les funcionalitats d'aquest, segons les seves necessitats i tornar a difondre la seva versió particularitzada. Això, sol ser una pràctica molt estesa, per fer créixer i millorar un desenvolupament inicial sobre un dispositiu.

Tot l'anterior, fa que aquest projecte, sigui d'interès general o important per a moltes persones, ja que elimina les restriccions detectades, en els productes actuals que hi ha en el mercat, i obre una nova via d'implicació i interacció, amb l'usuari final del dispositiu creat.

1.2. Descripció del projecte.

Els requisits d'aquests projecte, en principi, són els propis d'un sistema encastat i els definits per a l'opció triada de **estació meteorològica de baix consum**. Per tant, en el projecte, es faran servir microcontroladors programables i elements electrònics controlats també per programació. Com que un dels requisits és que sigui de baix consum, s'ha d'aprofitar els modes 'Idle'¹³ de que disposen els microcontroladors emprats.

El funcionament que s'ha d'obtenir en el projecte triat, és relativament senzill, s'ha de recollir la informació generada per uns sensors, de diferents valors meteorològics (temperatura, humitat, lluminositat, pressió Atmosfèrica ...), i enviar-la a un servidor Web local o a Internet. Degut a que aquesta informació estarà en base a un varem, possibles unitats per defecte, i es rebrà amb cert

¹¹ Treball Final de Carrera

¹² De l'anglès "General Public License" de GNU (GNU is Not Unix), Llicència General Pública de GNU

¹³ Mode en que s'atura el processador i per tant no consumeix, en hi ha de diferents nivells

format, caldrà recuperar-la convenientment, processar-la per tenir-la en referència al Sistema Internacional i així tenir-la a punt per ser enviada.

Per altra banda, el fet de rebre-la periòdicament, també ens permet poder realitzar un monitor a temps real sobre aquesta informació rebuda. I també, com que la informació serà enviada a un servidor Web, cal mirar d'emmagatzemar aquestes dades rebudes, per que puguin ser consultades, o per si s'ha de fer un tractament estadístic o evolutiu d'elles. D'aquesta forma, s'amplia en valor d'aquesta informació, al poder-ne fer un estudi del que s'hagi recopilat.

Per tal de clarificar l'escenari de treball, en que es trobarà el dispositiu final creat, posem una imatge força aclaridora, sobre l'entorn i connectivitat del sistema.

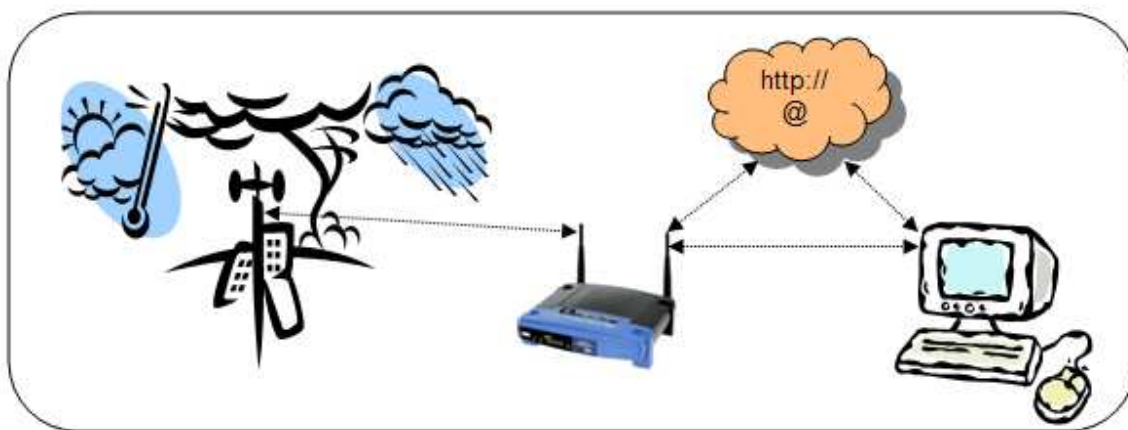


Figura 1 : Escenari de treball

1.3. Objectius del projecte.

L'objectiu principal del treball final de carrera (TFC) és el d'aplicar i materialitzar els coneixements adquirits en el decurs de la carrera. En aquest cas concret, s'ha escollit el projecte de "Estació meteorològica de baix consum", on es faran servir microcontroladors (μC), sensors, i altres dispositius (com el WiFly), tots ells de baix consum i controlats per programació. Per tant, cal revisar i conèixer a fons les particularitats tècniques d'aquests aparells, per poder-los acoblar i fer que treballin de forma conjunta, mitjançant una programació que hem de desenvolupar. En particular, s'ha de fer que el sistema treballi en mode de baix consum, ja que es una peculiaritat dels sistemes encastats, i també, serveix per estalviar consum d'energia al sistema. Per altra banda, ha de ser possible conèixer, consultar o tenir la possibilitat de tractar les dades recollides pel nou dispositiu creat. Així doncs, cal recollir les dades generades pels sensors, tractar-les de forma convenient, i enviar la informació pertinent a un servidor Web (local o a Internet) per poder-li fer un tractament posterior.

Degut a que l'anterior engloba moltes coses, de diferent temàtica, podem desglossar en objectius del projecte, les parts mes concretes i d'abast inferior. Així, els objectius principals del nostre sistema, associats a aquest projecte en concret, són els de la llista següent :

- Modelar un sistema μ C per controlar l'estació meteorològica.
- Dotar al sistema d'un mètode de depuració (Debug/Log).
- Dotar de connectivitat sense fils al sistema.
- Recollir la informació generada per els diferents sensors.
- Rebre de forma sincronitzada la informació al μ C.
- Enviar al Web les dades rebudes, en format correcte.
- Poder consultar o visualitzar les dades rebudes al Web.
- Dotar de funcionament en mode baix consum al sistema.

Per altra banda, per tal de millorar les prestacions del sistema definit i donar-li més funcionalitat, decidim afegir altres objectius, que son viables amb el sistema fet servir. Per tant, com ha objectius secundaris o que hem decidit afegir, als que s'han de fer com a mínim en aquest projecte, tenim la següent relació :

- Dotar al sistema d'un mecanisme de 'reset'¹⁴ manual.
- Dotar d'autosuficiència energètica al sistema.
- Dotar al sistema d'un mètode de configuració remota via IP
- Dotar de seguretat davant falles al sistema.

De forma resumida, en objectius principals, s'exposa que s'ha de dur a terme la programació del sistema encastat, per tal de controlar el conjunt de mòduls i dispositius, que conformen el sistema de la nostra solució. Inicialment, es desenvolupa un driver a la llibreria, que permet veure per una sortida, els missatges que posem al sistema, per tal de poder fer una depuració, control i diagnosi del que anem fent. Seguidament, es fa servir un mòdul de comunicacions sense fils, connectat al sistema, per que aquest es pugui comunicar amb l'exterior. Així, es desenvolupa un altre driver, per tal de gestionar i controlar aquest mòdul, que ens servirà per poder comunicar amb el servidor Web. Després, es posa al sistema, un mòdul que gestiona varis sensors i que envia les dades pertinents, cada x segons. Per tant, es crea un altre driver que gestiona i controla aquest mòdul, i permet programar la freqüència d'enviament i format, de les lectures efectuades. Per tal de sincronitzar l'enviament de les dades, amb la disponibilitat del sistema per a la recepció d'aquesta informació. S'implementarà un altre driver, que atura els processos del sistema i el posa en mode espera (baix consum) fins rebre un senyal. Al rebre'l, el sistema es reactiva, i processa les dades rebudes, que venen en un format concret. Per tant, s'implementa un altre driver que formata i

¹⁴ Procediment físic o per programa, mitjançant el qual es fa reiniciar el sistema.

envia la informació segons els requeriments del servidor web. I com que la informació rebuda al servidor web, a d'estar disponible per consultes o tractament, s'emmagatzema e una DB¹⁵.

I per als objectius secundaris, de primer s'implementa un botó que facilita el reinici i sincronització del sistema. Després farem que el sistema s'alimenti d'una placa solar i una bateria, que permet una llarga autonomia. Posteriorment, s'implementa un sistema de configuració remot, per poder modificar el valor de la freqüència de lectures, per poder canviar a un altre mode de baix consum dels que disposa el sistema, i per últim poder fer un reinici remot, que posi i sincronitzi a l'estat inicial el sistema. Finalment, s'afegeix al sistema un codi de seguretat, que a part de despertar al sistema si no arriba cap lectura, controla el errors, tant d'enviament com de recepció de dades, fent un reinici o un apagat del sistema segons les opcions programades.

1.4. Enfocament i mètode seguit.

Per tal d'arribar a bon port amb aquest projecte, a nivell personal, s'ha enfocat de forma que hi hem dedicat totes les hores que ens ha estat possible. En quant als recursos de coneixements, des de bon començament, s'ha emprat la 'wiki' de sistemes encastats, també s'ha seguit un procediment d'informació i aprenentatge bastant lògic. És a dir, primer sempre hem buscat informació, recollit documentació i exemples de codi, per aprendre el funcionament del que havíem de fer o emprar. Així doncs, abans que res, ja teníem una sèrie de carpetes organitzades amb tot el que havíem trobat per Internet, i a mesura que avançàvem en el projecte, anàvem afegint coses segons necessitats, si no les teníem. D'aquesta forma, sempre hem disposat com a mínim d'una part important d'informació tècnica, documentació i exemples de codi, sobre les diferents parts del sistema que hem desenvolupat.

Per altra banda, no sempre hem trobat el que buscàvem i hem recorregut a fòrums especialitzats en sistemes encastats, que tractaven el tema que ens interessava. Així, aquest procediment, s'ha convertit en una altra font important d'informació, sobre tot, per als temes mes complexos. Podem destacar en aquest cas, el fòrum de NXP per al microcontrolador LPC1769 i el fòrum de Sparkfun¹⁶ per al mòdul de sensors que hem emprat en el projecte.

En quant al desenvolupament del nostre programari, tant per la llibreria com les aplicacions fetes, a part de fer servir els recursos de Debug del IDE, i el driver Debug/Log que hem creat, hem emprat tot sovint, la clàssica metodologia de prova – error. És a dir, desenvolupàvem una funció nova i provàvem el seu funcionament en una aplicació de test que havíem creat expressament per aquest motiu. Alguna vegada, quan la funció a generar era similar a una altra, hem copiat i modificat el codi pertinent sense fer la seva comprovació, ja que era bastant obvi que havia de funcionar correctament. De totes formes, aquest procediment, ens ha donat alguna sorpresa, al no valorar

¹⁵ De l'anglès "Data Base", Base de Dades.

¹⁶ Companyia americana que subministra material electrònic i principalment de sistemes encastats.

correctament el que volíem fer. També, hem de reconèixer que el procediment definit, de fer unes PAC's inicials, per tal de construir una base de treball, ens ha estat molt útil. D'aquesta forma, hem adquirit uns coneixements bàsics i una mica d'experiència sobre el funcionament del sistema encastat, que es fa servir en el projecte.

Finalment, per a l'elaboració de la memòria, ens ha ajudat força el que havíem fet en les PAC's anteriors, ja que algunes coses que es demanaven, es poden aprofitar per algun apartat d'aquesta memòria. De fet, en el nostre cas, com que hem fet algunes variacions en quant a objectius i tasques realitzades, hem tingut que modificar i adaptar-ho a la nova situació, en que hem desenvolupat aquest projecte.

1.5. Planificació del projecte.

Inicialment hem desglossat el projecte en tres fases: una d'iniciació i comunicacions bàsiques del sistema, una altra de comunicacions avançades i tractament d'informació, i l'última de generació de documents i presentació. En principi, la fase 1 correspon a les quatre primeres PAC's, la fase 2 a la PAC5, i la fase 3 a les PAC's 6 i 7. Per tant, hem definit el següent cronograma :

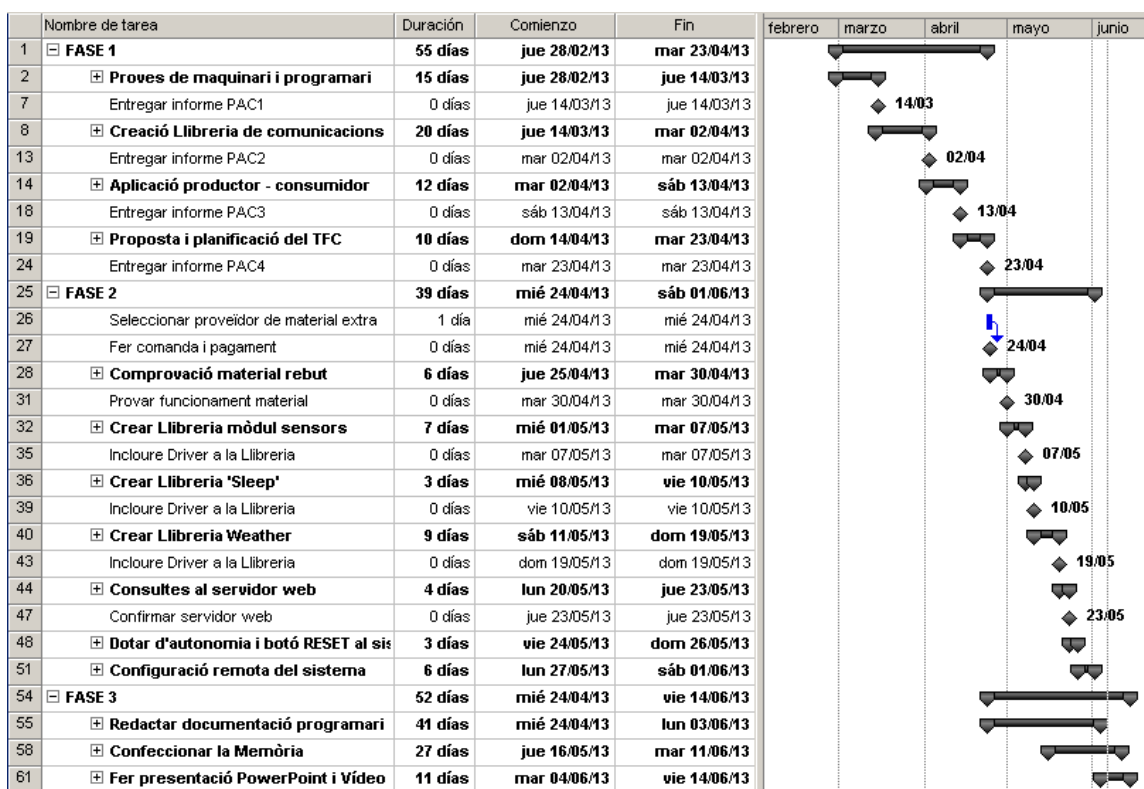


Figura 2 : Planificació inicial

Degut a modificacions d'abast als objectius definits inicialment, l'annexió d'un nou objectiu, i algun canvi per problemes sobre el funcionament esperat, la planificació final que s'ha seguit, ha variat.

Posem el diagrama de Gantt, amb la nova planificació, que finalment hem seguit :

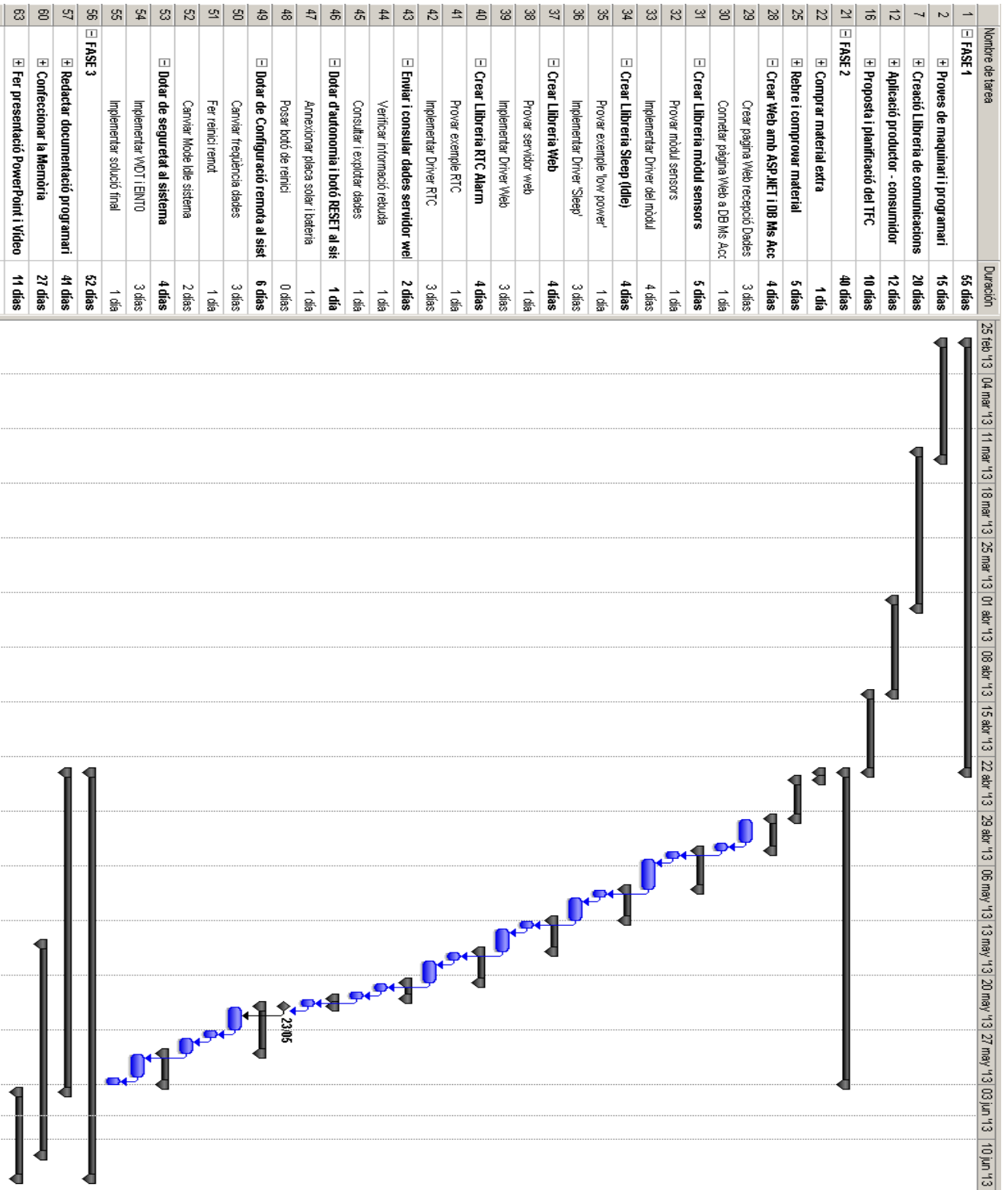


Figura 3 : Planificació final

Mirant el cronograma anterior, es pot observar que només han variat les tasques involucrades en la fase 2. Els motius d'aquests canvis, son deguts primer a que el servidor web que havíem triat no funcionava tal com esperàvem, estava classificat com obsolet. Segon, hem decidit emprar els altres modes 'Idle' de que disposa el microcontrolador fet servir en el projecte. I tercer i últim, a que hem volgut dotar d'una seguretat extra davant falles al sistema.

En la primera situació, això ens ha obligat a desenvolupar una pàgina Web, connectada a una Base de Dades, que funciona amb un servidor de pàgines ASP.NET¹⁷. Per tant, hem tingut que dissenyar i adaptar el seu funcionament a les necessitats del projecte. Més endavant, hem tingut que verificar el seu funcionament, i rectificar alguna cosa, quan ja teníem el projecte avançat.

En el segon cas, al voler emprar els diferents modes 'Idle' del sistema, hem tingut que crear alguna funció nova per al WiFly, variar el procediment de lectura de les dades dels sensors, i generar un driver nou que proporcionés un senyal al sistema per despertar-lo. De fet, també ens ha obligat a fer alguns retocs en diverses funcions d'altres drivers per que fessin el que volíem.

I per últim, al dotar de seguretat davant falles al sistema, hem tingut que fer alguns canvis a les funcions dels drivers de la llibreria que teníem. També ens ha fet replantejar i modificar el sistema de Debug/Log que havíem definit inicialment per al projecte, però com que això pertany a la fase 1, no ho hem afegit. Per altra banda, degut a tot l'anterior, hem tingut que fer alguns canvis a l'aplicació final, per tal de que funcionés amb conseqüència.

1.6. Recursos emprats.

En aquest treball final de carrera, per tal de construir el dispositiu que hem seleccionat, podem veure clarament diferenciats dos tipus de recursos, que s'han fet servir. El primer, correspon al maquinari seleccionat i utilitzat per a la part electrònica del projecte, i el segon, correspon al programari fet servir per al desenvolupament d'una aplicació, que fes funcionar al maquinari de la forma desitjada. En els apartats següents en fem una breu descripció :

1.6.1. Maquinari.

- **Microcontrolador LPC1769**

Inicialment va ser dissenyat com sistema d'avaluació per als circuits integrats del fabricant NXP amb un processador ARM-Cortex M3. Posteriorment, amb la col·laboració de Embedded Artists i Code Red Technologies, va evolucionar i s'ha convertit en un bon sistema encastat, que permet l'avaluar i provar nous dissenys de circuits que funcionen amb aquest microcontrolador. Pertany a la família de plaques LPCXpresso, que es caracteritzen per portar afegida la unitat LPC-LINK, que

¹⁷ Tecnologia de Microsoft per processar pàgines web, emprada en els seus servidors i eines de desenvolupament.

es pot separar, i que facilita la càrrega d'aplicacions al sistema, mitjançant un connector USB. També, es possible emprar un altre connector JTAG¹⁸ que hi ha a la mateixa placa, per tal d'efectuar la càrrega d'aplicacions.



Figura 4 : microcontrolador LPC1769

- **Mòdul WiFly RN-XV (xarxa Wi-Fi)**

Permet la connexió a una xarxa sense fils, ja que disposa d'un Wi-Fi, i a més, un port UART¹⁹ i altres pins amb funcionalitat diversa. De fet, es tracta d'un microcontrolador amb el seu propi 'Firmware'²⁰, i funciona amb un seguit de comandes de configuració, que permeten operar amb ell. Un cop se li ha ajustat una configuració, l'interessant d'aquest mòdul, és que funciona de mode transparent entre la part Wi-Fi i el seu port UART. És a dir, el que s'envia per un costat, surt per l'altre i a la inversa, per tant, ens fa de pont, per poder accedir a la xarxa.



Figura 5 : Mòdul WiFly

- **Adaptador USB – sèrie CP2102**

Serveix per poder veure els missatges de Debug/Log del nostre desenvolupament, mitjançant un programa d'emulació de terminal. Per entorns Windows, cal instal·lar un driver, que genera un port UART VCP (Virtual COM Port) sobre la connexió USB, que es emprat per el programa terminal. En l'altre extrem, a part dels pins de UART, en té d'altres per donar alimentació a 3,3 V ó a 5 V.



Figura 6 : Adaptador USB - sèrie CP2102

¹⁸ De l'anglès "Join Test Action Group", norma IEEE 1149.1, per testejar circuits electrònics

¹⁹ De l'anglès "Universal Asynchronous Receiver-Transmitter", conegut per Port COM

²⁰ Mot anglès, és una aplicació que treballa a baix nivell i sol ser actualitzada per fabricant

- **Mòdul de sensors (USB Weather V3)**

Pren les lectures dels sensors que porta incorporats i les envia per port USB o uns pins UART de que disposa, cada X temps, posat a la seva configuració. Al igual que el WiFly, es tracta d'un microcontrolador amb el seu propi 'Firmware', però en aquest cas, disposa d'un menú de configuració. En ell, es defineix la mètrica del mode de treball, el format de les dades, l'alçaria on estarà ubicat i el temps de presa de lectura, entre d'altres més que hi ha.

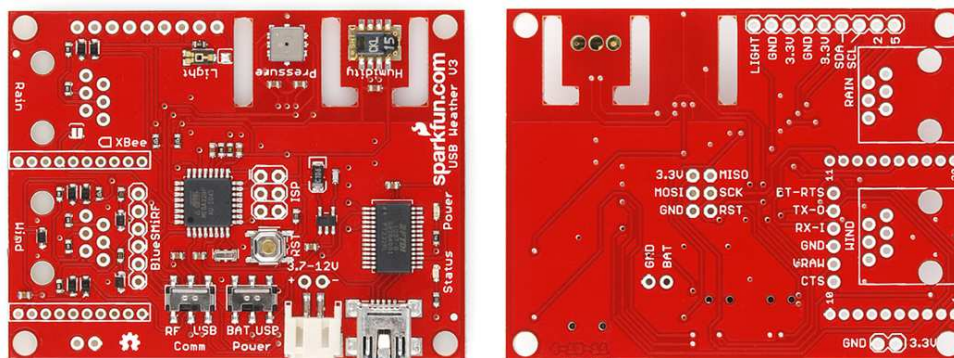


Figura 7 : Mòdul de sensors (USB Weather V3)

- **Placa solar amb bateria recarregable**

La bateria, subministra l'energia necessària per fer funcionar el nostre dispositiu, proporciona 5 V i fins a 4000 mAh de corrent. I la placa solar, en horari diürn, arriba fins als 1000 mAh de corrent generada, per tant, l'energia que no es consumeix en aquesta franja solar, es destina a carregar la bateria, per tal de poder funcionar en horari nocturn, i durant molt de temps.



Figura 8 : Placa solar amb bateria recarregable

- **Material electrònic divers**

Per connectar tot això, hem fet servir una 'protoboard'²¹, cables pont de diferents mides, connectors específics per als mòduls en que ha fet falta, i pins de coure estanyat. Per últim, també s'ha fet servir un soldador d'estany, per posar els pins i assegurar els connectors.

²¹ Mot anglès, placa per muntar prototips, també coneguda com "Breadboard"

1.6.2. Programari.

- **LPCXpresso V5.0.14 basat en Eclipse 4.2.1**

Es una plataforma de desenvolupament o IDE, subministrada per NXP, i està dissenyada per treballar amb diferents microcontroladors LPC d'arquitectura ARM. De fet, l'adaptació i ajustament sobre Eclipse, d'aquest programari, l'ha fet l'empresa Code Red. Es tracta d'una eina fonamental en el nostre projecte, ja que ens permet escriure el codi de l'aplicació, compilar-lo i carregar-lo directament, via USB, a la nostra placa. A part de correcció d'errors en l'escriptura, també ens proporciona, un sistema de depuració del codi executat, amb breakpoints i la consulta de valors en temps d'execució. Per altra banda, ofereix tota una sèrie d'utilitats (incorporació de programes, exportació en format zip, ...) que son de gran valor afegit i que hem emprat sovint.



Figura 9 : Logo del LPCXpresso

- **FreeRTOS**

Es tracta d'un sistema operatiu (O.S.) per a sistemes encastats, que treballa a temps real. Es troba distribuït amb llicència GPL, amb una exclusió, que permet que l'usuari final pugui tenir el codi tancat, és a dir, fa que l'usuari final pugui crear aplicacions propietàries. Està dissenyat de forma que sigui simple en l'ús i generi aplicacions petites en quant a mida de l'executable. Principalment està desenvolupat en llenguatge C, però, també inclou algunes rutines en ensamblador, depenent de l'arquitectura que es faci servir. Proporciona funcions o mètodes, per a la gestió, programació i control, de tasques, cues, semàfors, suspensió o bloquejos, prioritats d'execució i temporitzadors. La seva principal virtut, respecte a altres sistemes, és que ofereix un sistema compacte amb una gran velocitat d'execució. A més, es troba recolzat per molts dels grans fabricants de maquinari de sistemes encastats.



Figura 10 : Logo de FreeRTOS

- **Aplicació de terminal “TeraTerm”**

Bàsicament, es un terminal gràfic per a comunicacions sèrie, del mateix estil que Hyperterminal de Windows o altres de similars. És distribuït de forma ‘open source’²² sota la llicència BSD²³ i es pot trobar fàcilment a Internet. Per a la seva construcció, s’ha fet servir la distribució gratuïta de Microsoft Visual C++ 8.0. Un dels grans avantatges que te aquesta aplicació, respecte les altres, és la gran quantitat d’opcions i configuracions possibles que te.

- **Microsoft Visual Studio 2008**

Es tracta de la versió d’estudiant d’aquest producte de Microsoft, aconseguida la llicència per mitjà del registre d’estudiant a la UOC. S’ha fet servir exclusivament, per crear i provar, les pàgines ASP.NET i la connexió a una Base de Dades Microsoft Access que també hem creat. De fet, es podria haver emprat la distribució gratuïta, però aquesta, no dona facilitats per a la connexió a Bases de Dades, i porta molta funcionalitat retallada. Per que funcionin les pàgines generades, només cal posar-les al directori de treball del servidor Web IIS (Internet Information Services) de Microsoft. En el nostre cas, hem triat una altra opció, emprar el servidor gratuït de pàgines fetes amb ASP.NET, que es diu “UltiDev Cassini 2.0”²⁴.



Figura 11 : Logo de Microsoft Visual Studio 2008

- **Microsoft Windows 7**

En el PC que hem instal·lat tot el programari, hi tenim per sistema operatiu (O.S.) el Microsoft Windows 7. Al ser un sistema de gran difusió, fa que tot el programari que hem fet servir, hagi estat fortament adaptat i comprovat per funcionar correctament en aquest entorn de treball. Per altra banda, aquesta versió, ofereix moltes millores respecte a les anteriors del mateix fabricant i ens ha permès la instal·lació i configuració del programari fet servir en el projecte, amb molta facilitat.



Figura 12 : Logo de Windows 7

²² De l'anglès, codi obert, significa que es distribueix tant l'executable com el codi original

²³ Variant de llicències de programari lliure : <http://tssh2.sourceforge.jp/manual/en/about/copyright.html>

²⁴ Es pot trobar a la URL : <http://ultidev.com/products/cassini/>

1.7. Productes obtinguts.

Al tancament de l'execució d'aquest projecte, els lliurables o productes obtinguts, han estat en principi dos, clarament identificables i ven diferents. Primer, tenim la estació meteorològica, com a compliment bàsic dels requisits del projecte, i segon, les pàgines Web amb la Base de Dades connectada, amb les que ha de treballar l'estació meteorològica. Cada un, ha estat desenvolupat amb la seva pròpia tecnologia i forma de funcionar, adaptada a les necessitats del projecte. La comunicació entre els dos productes, es fa via xarxa Wi-Fi, aprofitant la connectivitat de que hem dotat a la estació meteorològica. Aquest mateix fet, ens ha permès poder realitzar una configuració remota del dispositiu, emprant una sessió Telnet²⁵ al port 2000 de la IP del mòdul WiFly.

- **Estació meteorològica**

L'aspecte més notable d'aquest producte, es la seva senzillesa, tant en la seva construcció, com en el seu funcionament, amb uns resultats més que òptims. Aquest fet, permet que pugui ser ampliada la seva funcionalitat o que si puguin afegir altres components al dispositiu, amb molta facilitat. La part central del sistema, és el microcontrolador LPC1769, que interacciona i controla, dos mòduls electrònics que te connectats. La comunicació entre ells i el microcontrolador, s'ha realitzat per mitjà dels diferents ports sèrie que tenen cada un dels dispositius implicats. Com que cada un dels mòduls te el seu propi 'Firmware', això ha implicat, el desenvolupament de tot un seguit de funcions, per cada un, que faciliten el treball de programació dins de l'aplicació que governa l'estació. Donat que en aquest projecte, la fita temporal marcada era molt justa, només ha donat temps a fer el muntatge d'un prototip funcional en una 'protoboard', i més endavant ja es farà una versió muntada i soldada en una placa.

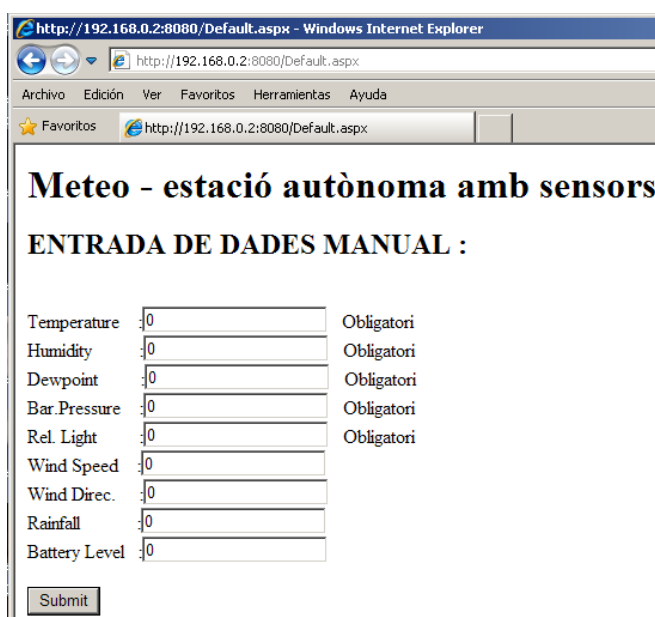


Figura 13 : Estació meteorològica

²⁵ Protocol de comunicacions TCP/IP, molt senzill, similar a una sessió de Terminal.

- **Web amb Base de Dades**

Aquest es una altre des productes obtinguts, i necessari per acomplir el requisits del projecte. Hem triat l'opció de fer la construcció amb ASP.NET, degut a la funcionalitat que ens ofería i de que ja disposàvem de les eines necessàries. De totes formes, hem tingut que cercar informació per aprendre i aplicar la funcionalitat que volíem aconseguir. De primer, hem generat una pàgina web que permetés la introducció manual de les dades que ens donarà l'estació meteorològica, i les guardés a la Base de Dades, tal com es pot veure a la imatge adjunta.

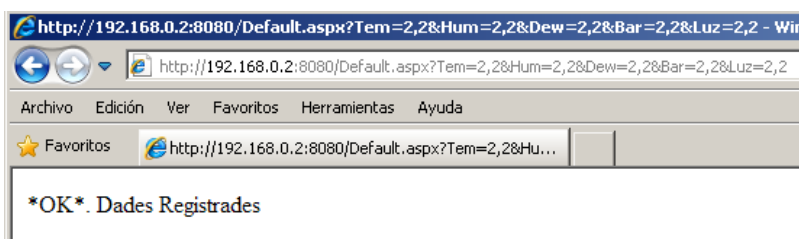


The screenshot shows a web browser window with the address bar displaying `http://192.168.0.2:8080/Default.aspx`. The page title is "Meteo - estació autònoma amb sensors". Below the title, the heading "ENTRADA DE DADES MANUAL :" is displayed. The form contains several input fields, each with a label and a "0" value, and a "Submit" button at the bottom. The fields are:

Temperature	:0	Obligatori
Humidity	:0	Obligatori
Dewpoint	:0	Obligatori
Bar.Pressure	:0	Obligatori
Rel. Light	:0	Obligatori
Wind Speed	:0	
Wind Direc.	:0	
Rainfall	:0	
Battery Level	:0	

Figura 14 : Entrada manual de dades

Posteriorment, hem modificat la funcionalitat de la mateixa pàgina, aprofitant les facilitats que ens ofería l'eina de programació, per tal de que processés dades que venien com a paràmetres. En aquest punt, hem tingut en compte, que les dades poden tenir un format curt o ampliat, segons els sensors que tingui funcionant el mòdul de sensors. Per altra banda, el separador de punt decimal que porten les dades dels sensors, està en format anglès, pel que també hem fet les correccions necessàries, per obtenir la informació correcta. També, hem posat un sistema que controla el valor dels paràmetres enviats i si l'enviament està formatat correctament, donant per resposta, la convenient pàgina de OK o No OK. Posem una imatge exemple del que surt, en cas correcte :



The screenshot shows a web browser window with the address bar displaying `http://192.168.0.2:8080/Default.aspx?Tem=2,2&Hum=2,2&Dew=2,2&Bar=2,2&Luz=2,2 - Win`. The page content displays the message: ***OK*. Dades Registrades**

Figura 15 : Enviament de dades correcte

1.8. Descripció dels altres capítols.

Fins aquí, s'ha fet una introducció als diferents aspectes que engloba aquest projecte, amb les justificacions pertinents en quant a objectius, decisions preses, planificació, recursos i productes obtinguts. Les explicacions tècniques donades, s'han fet forma generalista i sense aprofundir en la complexitat dels detalls que incorpora aquest projecte.

En els propers apartats, de primer, es farà un estudi de la tecnologia actual comparable a la que hem fet servir, fent un valoració de les seves possibilitats sobre aquest projecte. Posteriorment, presentarem de forma general el disseny emprat en el nostre cas, i desgranarem en detall cada una de les parts en que està composta la solució que hem triat en aquest projecte. Entrarem també en un estudi de la seva viabilitat tècnica i farem una valoració econòmica, dels costos materials i de temps dedicat per assolir la totalitat del projecte. Finalment exposarem unes conclusions a que hem arribat en el decurs del projecte, també comentarem unes millores que creiem que son factibles d'implementar i farem una autoavaluació sobre el projecte.

2. Antecedents.

En el decurs de la història, tenim una presència temporal, gairebé igual entre els microcontroladors i els microprocessadors. Inicialment, els microcontroladors eren emprats per a petits automatismes electrònics o per donar suport als microprocessadors i alliberar-los de tasques rutinàries. A mesura que tot això ha anat evolucionant, les prestacions dels microcontroladors, també s'han ampliat fortament. Les primeres aplicacions que trobem d'aquestos, com aproximació a un sistema encastat, seria els automatismes electrònics programables, posteriorment algunes calculadores científiques i finalment les agendes electròniques, que van generar les PDA²⁶ amb pantalla.

Aquests últims anys, s'ha viscut una gran revolució, amb l'aparició dels sistemes SoC²⁷, que combinats entre ells, a donat peu a una gran difusió dels sistemes encastats, com les PDA, el GPS²⁸, també els telèfons mòbils i les càmeres fotogràfiques digitals. Això, ha generat un interès creixent per aquest tipus d'aparells, si bé, la frontera amb els microprocessadors, ha quedat cada cop més difosa, degut al gran augment de prestacions i funcionalitat oferta.

Els dispositius anteriors, ofereixen una solució tancada, i en el nostre cas, no ens serveixen, per tant, emprarem els que han evolucionat dels automatismes electrònics programables, que son els mes semblants. Aquestos, també han millorat en prestacions, tecnologia emprada i han facilitat molt la programació. En l'actualitat, aquestos, solen ser emprats en laboratoris, vehicles, automatismes, militar, robòtica, seguretat, control remot, desenvolupament industria, etc.

²⁶ De l'anglès "Personal Digital Assitant", agenda electrònica de butxaca.

²⁷ De l'anglès "System-on-Chip", fa referència a posar tot un sistema sencer en un sol chip.

²⁸ De l'anglès "Global Position System", sistema de posicionament global basat en satèl·lits.

2.1. Estat de l'art.

Actualment, ens trobem en un moment de molta difusió i utilització de sistemes encastats, si bé, amb certs matisos. Per exemple, en el cas dels telèfons mòbils, càmeres fotogràfiques digitals, GPS, tauletes digitals, o llocs de venda, que més que un sistema encastat, els podríem considerar un petit ordinador, amb unes prestacions fixades. En aquest sentit, s'ha de reconèixer que Linux, sempre ha estat un pioner en ser utilitzat per aquest tipus d'aparells, amb més prestacions que un sistema encastat clàssic. De fet, gairebé en tots aquests aparells, es sol parlar de quin sistema operatiu (O.S.) porten per funcionar, més que de versió de Firmware. Encara que aquest programari pugui ser de lliure distribució o potser tancat (com el Windows 8, Symbian, IOS, ...), sol ser el fabricant qui en proporciona versions fiables, i adaptades correctament al seu dispositiu. Per altra banda, gràcies a Linux i Android, molts d'aquests aparells tenen un baix preu que es molt bo per al consumidor final.

Un camp en el que actualment està jugant un paper molt important, els sistemes encastats, es el de l'automoció. S'està aplicant des del GPS a bord, passant per controls de seguretat, consums i funcionament del motor, vehicles intel·ligents o personalitzats, fins al control remot del vehicle. De fet, la demanda de vehicles amb més prestacions d'aquest tipus, va en augment, i es preveu un fort creixement en aquest sentit. Adjuntem una taula d'aplicacions / prestacions – interessats, segons "SURGE-Transport"²⁹.

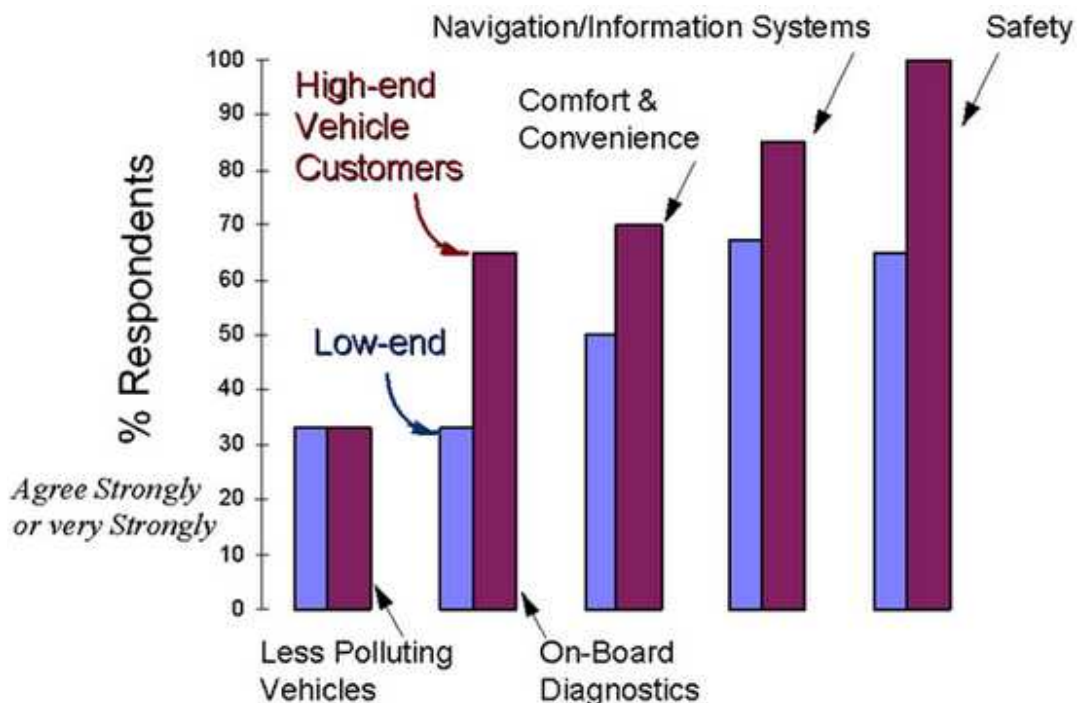


Figura 16 : Taula interès vehicles amb sistemes encastats

²⁹ Es pot trobar a la URL : <http://cordis.europa.eu/esprit/src/omi-trpt.htm>

En quant a dispositius, del mateix estil que el emprat en el projecte, tenim el clàssic “Arduino” i recentment el ‘Raspberry pi’. Aquest últim, degut a un gran interès per les seves prestacions i opcions, en molt poc temps, va superar el milió d’unitats venudes. És interessant, perquè te la mida d’una targeta de crèdit i el seu cost es a partir de 25\$ més despeses d’enviament. Aquests dos dispositius i el que hem emprat, tenen circuits accessibles a nivell electrònic, disposen de connectors genèrics per annexionar nou maquinari, i fan servir una connexió USB per donar accés fàcil i còmode a l’entorn de programació. Tots ells, tenen un preu força raonable, i degut a les prestacions i opcions que tenen, en aquests últims anys ha crescut amb rapidesa la seva popularitat.

En aquest sentit, val la pena que posem les seves característiques principals, per tenir una millor idea del que ofereix cada un d’ells, i així en podem fer una valoració correcta, en quant avantatges i inconvenients per al projecte.

- **Arduino Duemilanove, característiques tècniques :**

- Microprocessador ATmega168 a 16 MHz
- Tensió operativa 5 V
- Tensió d’entrada (recommended) 7-12 V
- Tensió d’entrada (limits) 6-20 V
- Digital I/O pins 14 (of which 6 provide PWM output)
- Entrades analògiques 6
- Corrent DC per I/O pin 40 mA
- Corrent DC per 3,3V pin 50 mA
- Memòria Flash (per codi) 16 KB of which 2 KB used by bootloader
- SRAM 1 KB
- EEPROM 512 bytes



Figura 17 : Placa de Arduino Duemilanove

- **Raspberry pi Model B, característiques tècniques :**

- Microprocessador 700 MHz ARM1176JZF-S core (ARM11 family)
- GPU Broadcom Videocore IV, OpenGL ES 2.0, 1080p30 h.264/MPEG
- SoC Broadcom BCM2835 (CPU, GPU, DSP, and SDRAM)
- Memòria (SDRAM) 256 Megabytes (shared with GPU)
- Ports USB 2.0 2 (via integrated USB hub)
- Adaptador xarxa 10/100 Ethernet (RJ45)
- Low-level perifèrics 8xGPIO, UART, I²C bus, SPI bus with 2 ch.sel., +3,3V, +5V, GND
- Onboard storage SD, MMC, SDIO card slot
- Power ratings 500 mA (2,5 W) 700 mA (3,5 W)
- Power source 5 volts via MicroUSB or GPIO header
- Sortida vídeo Composite RCA, HDMI
- Sortida àudio 3,5 mm jack, HDMI

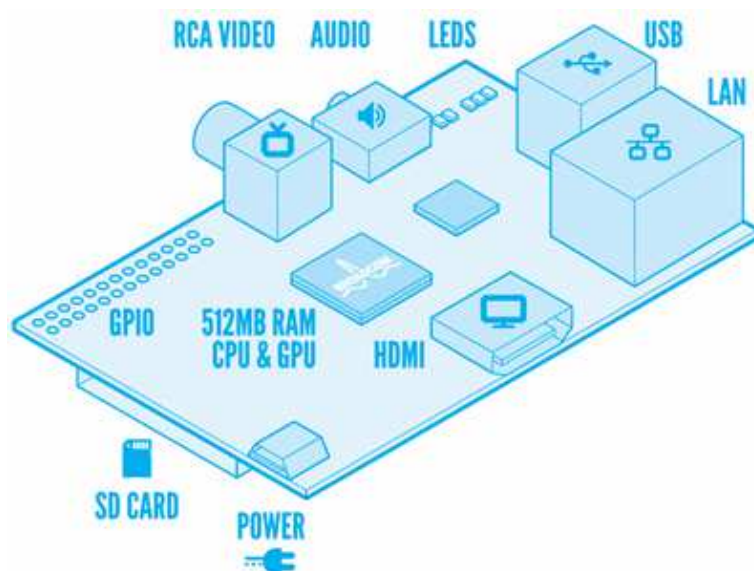


Figura 18 : Placa de Raspberry pi Model B

- **LPC1769 de NXP, característiques tècniques :**

- Microprocessador ARM-Cortex M3 core with speed up to 120 MHz
- Memòria Flash 512 kB
- Memòria RAM 64 kB
- Xarxa Ethernet 10/100 Ethernet MAC
- Ports USB 2.0 1 USB 2.0 Host, 1 USB 2.0 Device, 1 USB 2.0 OTG
- Alimentació 5 V via MicroUSB ó connector J6-1 i J6-2
- Low-level perifèrics 2xCAN, 4xUART, 3xI2C, 2xSSP, 1xI2S

- Altres dispositius 4 32-bit counter / timers, 1 Watchdog, PWM / Motor, Motor control PWM with support for 3 phase, Quadrature encoder interface, 12-bit ADC multiplexing 8 pins, Single 10-bit DAC output, 4 power modes : Sleep, DeepSleep, PowerDown, and DeepPowerDown.

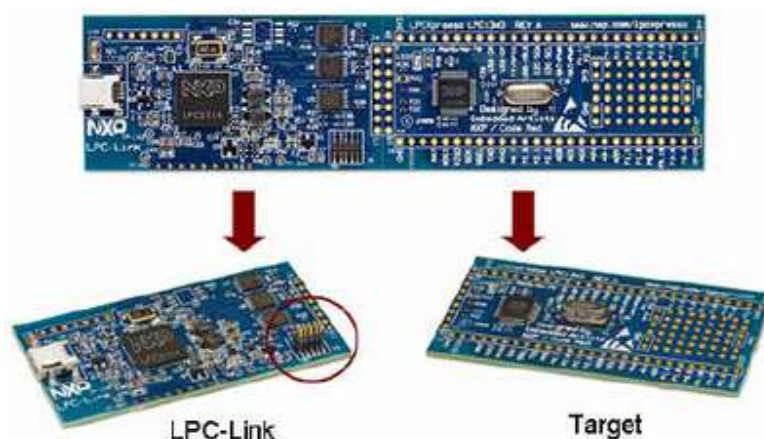


Figura 19 : Placa LPC1769 i components

Mirant les característiques tècniques que hem posat, podem apreciar que hi ha importants diferències en quant a velocitat CPU, memòria disponible, consums, pins per a dispositius i connexions disponibles. Podríem dir, que cada una d'elles, en certa forma, està enfocada a un tipus de solució concreta. En aquest cas, tenim per exemple la 'Raspberry pi' que es comporta més aviat com un petit ordinador. La placa Arduino es farà servir més aviat per temes de molt baix consum i pocs requeriments de programació, encara que disposa d'un mòdul d'ampliació, que li permet controlar molts dispositius. I per últim, la més versàtil i entremig de les dos anteriors, podem trobar la placa LPC1769, que donat l'ús dels ports UART i dels modes 'Idle' que fem en el projecte, es la ideal. Així doncs, tal com hem fet, sempre s'ha de mirar primer les necessitats del projecte i després cercar entre les possibles candidates, la que s'ajusta més al que s'ha definit.

En quant al tema de comunicació via TCP/IP³⁰, només porta de base un connector Ethernet la 'Raspberry pi', la LPC1769 te els pins i s'hauria de muntar el connector, i en la placa Arduino s'ha d'emprar un mòdul d'expansió. Però, cap de les tres plaques, incorpora un sistema Wi-Fi de connexió a la xarxa, que faciliti la mobilitat del dispositiu. De totes formes, en el mercat, hi ha altres sistemes de connexió sense fils, que faciliten la mobilitat, com pot ser el Bluetooth³¹, el ZigBee³², la UWB³³ o el 3G³⁴. Totes elles disposen de formats electrònics, que podrien ser emprats en aquest

³⁰ De l'anglès "Transport Control Protocol / Internet Protocol", protocol de comunicacions de la torre OSI.

³¹ Tecnologia de connexió sense fils, de curt abast, desenvolupada inicialment per NOKIA.

³² Altre tipus de connexió sense fils, de curt abast, baix consum i xarxa en malla.

³³ De l'anglès "Ultra-Wide-Band", banda ultra-ampla, nova tecnologia de connexió sense fils i curt abast.

³⁴ De l'anglès "3 Generation", fa referència a la telefonia mòbil de tercera generació, bona cobertura.

projecte. En aquest sentit, caldrà doncs, fer una valoració d'elles, segons el que ens poden oferir per dotar del millor sistema de comunicació sense fils, al nostre projecte. Per tant, fem una taula comparativa de les principals característiques que les defineixen, com a mòduls que es poden acoblar al nostre sistema i que ens poden interessar.

Especificacions	Bluetooth	ZigBee	UWB	Wi-Fi	3G
Bit-Rate	1 Mb/s	250 kb/s	110 Mb/s	54 Mb/s	3,2 Mb/s
Abast	10-100 m	10-100 m	10-100 m	100-300 m	Variable
Latència	200 ms	300 ms	200 ms	150 ms	200 ms
Seguretat	Baixa	Mitja	Alta	Alta	Alta
Consum	Mitjà	Baix	Baix	Mitjà	Alt
Cost	Baix	Baix	Alt	Baix	Alt
Estàndard	802.15.1	802.15.4	82.15.3a	802.11 a/b/g	UMTS
Freqüència	2,4 GHz	868/915 MHz i 2,4 GHz	3,1 -10,6 GHz	2,4 i 5 GHz	1,8 - 2,5 GHz
Potència Màx.	10 dBm	- 25 dBm	- 41,3 dBm	15 – 20 dBm	15 – 35 dBm
Canals	79	1/10; 16	15	14	No afecta
Facilitat TCP/IP	Mitja	Mitja	Baixa	Alta	Alta
Presència de Mòduls en el Mercat	Alta	Alta	Baixa	Alta	Baixa

Figura 20 : Taula comparació tecnologia sense fils

Si mirem la taula anterior, podem observar que per tal de tenir les millors prestacions, amb un consum moderat i baix cost, la millor opció de totes és el Wi-Fi. Destaca principalment, per oferir una bona velocitat de comunicació, un molt bon abast en comparació amb els altres, una bona seguretat de connexió, les seves facilitats en emprar TCP/IP i la seva gran presència o facilitat de trobar en el mercat.

En quant al mòdul de sensors, la decisió ha estat ràpida i senzilla, ja que només hi havia dues opcions possibles. La primera, era escollir els sensors individualment i desenvolupar un driver de comunicació per cada un, si no eren del mateix estil. I la segona, escollir un mòdul de sensors, que ja tenia el tema de preses de lectures i enviament de dades solucionat amb un port UART.

En resum, les opcions escollides de la placa LPC1769 de NXP i el mòdul WiFly, que es fan servir en aquest projecte, són els idonis per realitzar aquest TFC. Per altra banda, en quant al mòdul de sensors, també s'ha triat, tenint en compte les facilitats que ofereix, vers les altres opcions que hi ha en el mercat.

2.2. Estudi de mercat.

Al fer un estudi del mercat actual, sobre estacions meteorològiques, podem veure que està inundat per una immensa quantitat de dispositius d'aquest tipus. Només cal posar el text "estació meteorològica", tant en idioma castellà, com en idioma anglès, en un cercador d'Internet, i surten milions de pàgines trobades. D'entrada, es pot veure que el mercat va dirigit a dos tipus d'usuari diferent, un s'enfoca a l'usuari particular, per al seu ús personal o casolà, i l'altre al usuari professional. Dins de cada un d'ells, podem trobar un gran ventall de productes, amb diferents prestacions, qualitat i preu final de venda. Per altra banda, en hi ha de tota forma, mida, color, ampliables o evolutius, instal·lables a diversos llocs, amb cables o Wi-Fi, i amb piles o bateries o autosuficients. Un exemple de les dos categories principals trobades, es la imatge següent :



Figura 21 : Estacions meteorològiques d'ús particular i professional

En l'actualitat, el mercat de productes dirigit a l'ús personal o casolà, s'ha vist fortament alterat, per la irrupció massiva de productes de fabricació China. Això, ha provocat una gran baixada de preu en aquesta categoria, encara que es manté el nivell de prestacions ofertes, però, la qualitat del producte ha baixat una mica, i han aparegut noves solucions que no existien abans. Això, vol dir que el consumidor final, té més tria, amb millors possibilitats de trobar la que realment li interessa. De totes formes, totes les opcions, són solucions tancades, que només manté el fabricant.

En quant a companyies fabricants d'aquests productes, totes en tenen de les dos categories, i tenim noms molt consolidats en aquest sector, com : “La Crosse Technology”, “Oregon Scientific”, “Davis Instruments”, Columbia Weather Systems”, “skyview”, i un llarg etc. Entre les d'ús personal trobem un varem de preus, que va dels 15€, fins als 300€, i per les de tipus professional, hi ha preus que oscil·len entre 85€, fins als 6000€. Degut a aquest fet, vista la qualitat que donen, junt a les opcions que porten i el seu preu, creiem que la millor classificació, hauria de ser entre baix cost, i cost superior, que es la que farem servir a partir d'ara.

La majoria de les de baix cost, a pesar de que ofereixen bastant informació, tenen una fiabilitat dubtosa, solen funcionar només amb piles, algunes ofereixen connexió via cablejat, només estan suportades pel mateix fabricant i solen emprar components electrònics obsolets o de difícil localització. Les de cost superior, per altra banda, ofereixen millors prestacions, més qualitat, millor connectivitat, més autonomia, i els components materials i electrònics només els sol proporcionar el fabricant. En tots els casos, es tracta d'un producte tancat, sense possibilitats de creixement o ampliació per part de l'usuari final.

Segons aquest estudi, podem veure, que no hi ha cap solució oferta en codi obert, que permeti que l'usuari final el pugui millorar o adaptar a les seves necessitats. Per altra banda, el material emprat, només es pot aconseguir al mateix fabricant, si bé, algun components electrònics es pot trobar en altres llocs.

3. Descripció funcional.

El funcionament del sistema, el podríem comparar als processos tradicionals, fets manualment, i seguits en una estació meteorològica clàssica. Es a dir, hi ha una sèrie d'aparells que mesuren uns valors, en base a un escala establerta, de tant en tant, es pren nota d'aquests valors, i s'anoten en un registre horari per dies. Posteriorment, aquestes anotacions es solen guardar o es transmeten a un centre de dades, per al seu estudi, valoració o seguiment.

El nostre sistema, fa el mateix, però, substituint tots els passos manuals, per processos electrònics digitals, mitigant els possibles errors humans comesos en la seva execució. D'aquesta forma, es millora la qualitat dels valors presos en cada mesura i s'automatitza el procés, deixant només la feina de supervisió o possible modificació del funcionament. Per altra banda, el fet de tenir un espai temporal variable entre les diferents preses i molt precís, fa que de les dades, es pugui fer un correcte anàlisis evolutiu. En quant al registre de les dades, hem decidit emprar una Base de Dades, ja que dona més versatilitat al sistema i facilita el tractament d'aquesta informació, mitjançant eines informàtiques. Referent també a les dades obtingudes, aquestes s'han tingut que adaptar al format correcte o requeriments, dels diferents processos que les han fet servir, fins que no han arribat a la Base de Dades. D'aquesta forma, tenim la interacció del sistema encastat amb un servidor Web, que es a qui realment posarà la informació a la Base de Dades.

Dins del sistema encastat, hem carregat una aplicació o 'Firmware' que està desenvolupada amb les eines de programari, que hem exposat anteriorment. Aquesta aplicació, utilitza les funcions dels drivers que hem creat al inici del projecte, per millorar la codificació emprada, i optimitzar les tasques que ha de realitzar. I els drivers construïts, s'han posat en una Llibreria per millorar-ne la seva portabilitat, codificació i facilitar-ne l'ús.

3.1. Sistema de l'estació meteorològica.

En el sistema desenvolupat, el primer que veiem, es que el sistema encastat, actuarà de client del servidor Web, que te la Base de Dades. Tota la informació de les lectures preses als sensors, es trobarà emmagatzemada en aquesta DB. El Servidor Web, un cop rep la informació, primer la recupera de la seva entrada, després verifica que hi hagi les dades correctes, també mira si te el format correcte, i si està tot correcte, la processa cap a la DB. Un cop finalitzats els passos anteriors, dona una resposta de OK, o No OK, al client, per tal d'informar-lo del resultat obtingut.

El funcionament del sistema encastat, es totalment diferent, ja que es des del punt de vista client i te altres tasques específiques. En l'aplicació que el governa, primer trobem un procés d'iniciació dels diferents mòduls que componen el sistema, seguit de l'enviament de les primeres dades de lectura inicials, i l'entrada en una tasca amb mode baix consum. Posteriorment el mòdul de sensors efectua una presa de dades, segons el temps de freqüència que te guardat, les envia a la placa LPC1769, i això genera una IRQ³⁵, que desperta a la CPU. A les hores, degut a un control de codi de la IRQ, s'efectua la recuperació de les dades, amb el format correcte i es passa a la tasca d'enviament d'aquestes dades. Aquí, es formata la informació de forma que la entengui el servidor Web, i se li envia, mentrestant, el sistema es queda per un temps, a l'espera de la resposta del servidor. Un cop rebuda o no, es fan les accions pertinents segons sigui OK, No OK, ó resposta no rebuda, i es passa el control a la tasca inicial de recepció de lectura. Al entrar en aquesta tasca, el primer que es fa, es posar la placa LPC1769 en mode baix consum i espera recepció de la presa de dades als sensors.

Mentre el sistema està en mode baix consum, s'ha permès efectuar una connexió remota al sistema encastat, per tal de modificar la configuració de funcionament de l'aplicació. I s'ha desenvolupat en el sistema, unes funcions específiques que ho fan possible.

Com es pot veure, aquest funcionament es repetitiu en el temps i només depèn de la recepció de les dades dels sensors. En aquest sentit, s'ha desenvolupat una variant, que dona solució als modes de baix consum, que no reconeixen la IRQ de recepció de dades. Es a dir, s'ha posat un control de temps a la LPC1769, que força la lectura de dades cada interval programat i no depèn del que tingui programat el mòdul de sensors. Això a millorat les prestacions del sistema que hem desenvolupat i ha fet possible reduir encara més el consum d'energia de tot el sistema.

³⁵ De l'anglès "Interrupt Request", vol dir petició de interrupció, pot ser de maquinari o de programari.

3.1.1. Diagrama de blocs.

En el següent diagrama, es pot apreciar tot el sistema en conjunt, surt tant la composició de la xarxa i el servidor remot, com la composició del sistema encastat, amb els diferents mòduls o plaques amb que està comunicat :

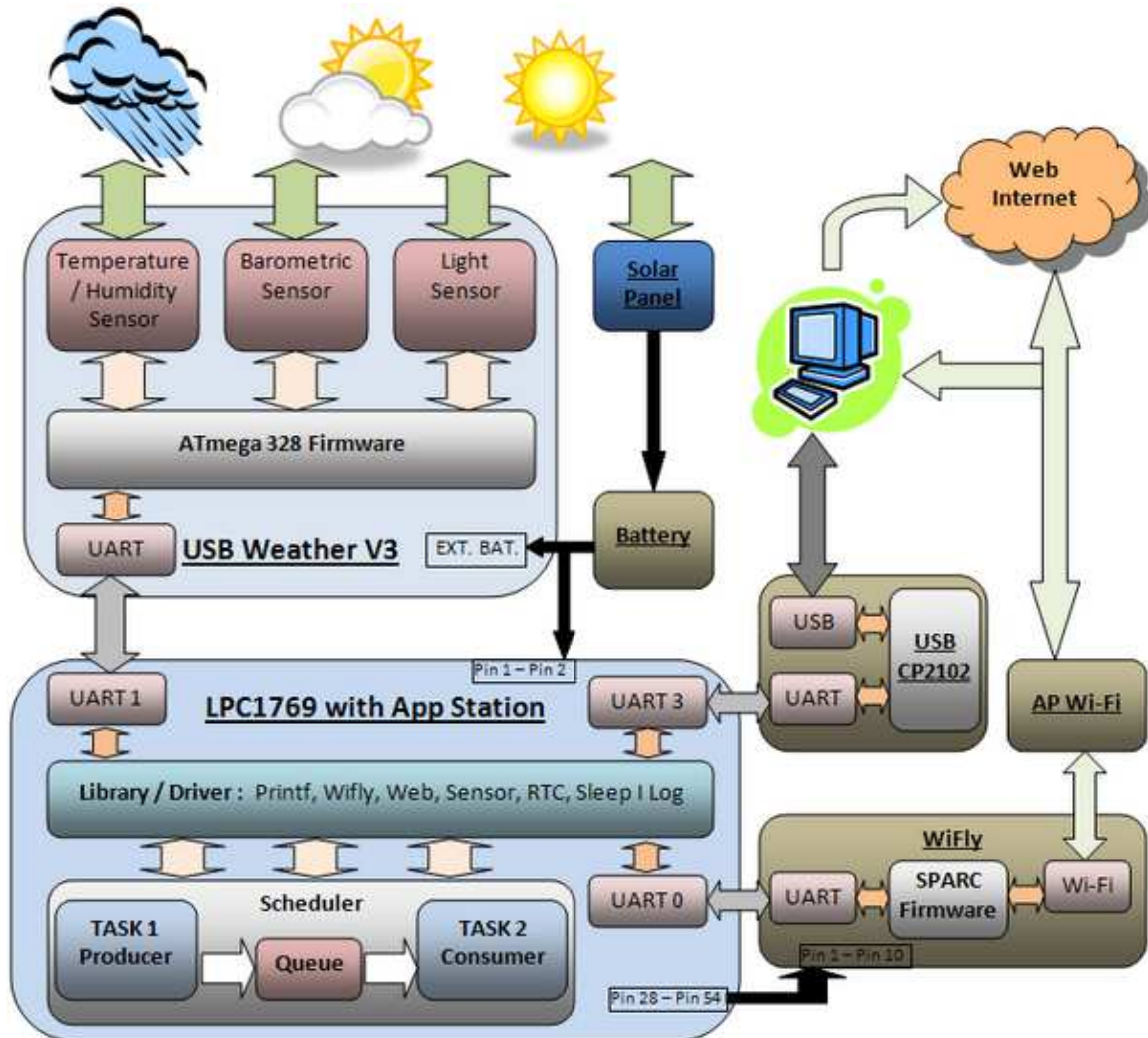


Figura 22 : Diagrama de blocs de l'estació meteorològica

En el diagrama anterior, es pot observar, que tenim diferenciades les següents parts :

Placa LPC1769 : fa córrer l'aplicació 'Station', amb un planificador que te varies tasques i empra la llibreria desenvolupada. Es comunica fent servir els ports UART, fa servir el port 0 amb el WiFly, el port 1 amb el mòdul de sensors "USB Weather V3" i el port 3 amb el CP2102 per Debug / Log. La seva font d'alimentació es una bateria que subministra 5V als pins 1 i 2 de la placa.

WiFly : connectat al port UART0, rep i transmet els missatges de / vers la LPC1769, cap un AP Wi-Fi al es connecta sense fils. S'alimenta dels pins 28 i 54 de la placa LPC1769

Placa de sensors “USB Weather V3” : fa córrer un ‘Firmware’ particular, configurat de forma que efectua lectures dels sensors incorporat, cada X segons, i ho envia pel seu port UART. La seva font d'alimentació es la mateixa bateria anterior, però li subministra 5V via EXT.BAT.

Battery : subministra 5V, a la placa LPC1769 pels pins 1 i 2, i al mòdul de sensors .

Solar Panel : en horari diürn, carrega la bateria i també fa de font d'alimentació del sistema.

USB CP2102 : connectat al UART3, envia els missatges Debug / Log a un port USB d'un PC.

AP Wi-Fi : permet la connexió del WiFly, a la xarxa local i a Internet

PC local : es poden veure els missatges de Debug / Log amb una consola, i pot tenir un servidor Web, emmagatzemar dades, consultar-les i/o reenviar-les a Internet

Requadre pins i fletxes negres : Representa cables i pins d'alimentació.

3.1.2. Interacció amb la xarxa.

L'estació meteorològica dissenyada, fa servir un adaptador Wi-Fi, concretament el WiFly RN-XV, per realitzar les comunicacions amb la xarxa. Aquest dispositiu o mòdul, està certificat per treballar amb xarxes IEEE 802.11 b/g a 2,4 GHz. Inclou mètodes d'accés a xarxes sense fils obertes o amb autenticació de seguretat, i també pot funcionar en mode ad-hoc³⁶ o també canviant el ‘Firmware’, en mode AP³⁷. En el nostre cas, fem servir el mètode client, amb la configuració de seguretat, que tenim definida al nostre router AP.

Un cop s'inicia el sistema, el primer que es fa és un ‘reset’ del mòdul WiFly, i mitjançant les comandes pertinents, se li configura una IP fixa i els paràmetres de connexió al router AP que tenim. Aquest fet, permet que el sistema encastat, tingui accés a la xarxa local i a Internet, emprant la connexió del router AP, al que s'ha associat. Un cop ha efectuat la connexió Wi-Fi, es guarda la configuració, perquè les properes vegades, ho faci de forma automàtica. D'aquesta forma, els reinicis del sistema i connexió a la xarxa triguen molt poc temps, cosa que ja ens interessa, per al funcionament eficient del sistema.

El sistema encastat, com que actua de client, es qui sempre iniciarà la comunicació amb el servidor Web, cada cop que tingui dades per enviar-li. A les hores, es realitza una sessió de connexió al servidor Web, les dades anteriors, es posen com una petició HTTP³⁸ amb paràmetres i s'envien per la connexió realitzada. Un cop han estat processades per el servidor, espera durant un temps la seva resposta, amb el mateix protocol que s'ha enviat. Si arriba la resposta OK continua correctament, i si arriba No OK o no hi ha resposta, controla l'error per a prendre les mesures que s'han programat.

³⁶ Mode de connexió sense fils descentralitzat, senzill, on tots els dispositius estan en igualtat de condicions.

³⁷ De l'anglès “Access Point”, punt d'accés, fa referència a un router Wi-Fi.

³⁸ De l'anglès “Hypertext Transfer Protocol”, protocol de transferència de hipertext, estàndard d'Internet

Per altra banda, com que el temps emprat en l'enviament i recepció de la resposta del servidor es molt curt, i el sistema està principalment en mode 'Idle', en aquest període, permetem una connexió remota al sistema. Com que el WiFly actua de mode transparent, fem una connexió 'Telnet' a la IP que hem fixat al WiFly, i aquest reenvia la petició a la placa LPC1769, on hi ha el sistema que hem desenvolupat. D'aquesta forma, activem una altra senyal IRQ, que es controla a l'aplicació creada i redirigeix el funcionament de la aplicació cap una espècie de servidor, de terminal de comandes. I per facilitar la configuració remota, hem generat un menú que s'envia a la sessió remota i té les opcions disponibles al nostre sistema. Així, amb la sessió 'Telnet' iniciada, només cal enviar l'opció del menú que volem, i el sistema actuarà en conseqüència, un cop tanquem la sessió amb l'opció corresponent del menú.

3.1.3. Interacció dels diferents objectes del sistema.

Anteriorment, ja hem indicat que el microcontrolador LPC1769 ofereix un gran nombre de sistemes d'interconnexió amb altres dispositius, que no tornarem a detallar en aquest punt. De tot el que té, per al nostre sistema de funcionament, ens interessa només, els 4 ports UART de que disposa i els 70 ports GPIO³⁹ que té per a configurar. De fet, en la nostra solució, fem servir només 3 ports UART i 2 ports GPIO, els primers per a comunicacions, i els segons per a control de dispositius.

La comunicació amb el WiFly, es realitza per mitjà del port UART0 del microcontrolador LPC1769, ja que el WiFly també disposa d'un port UART. Donat que el WiFly porta un 'Firmware' que gestiona el seu funcionament i connexions, ofereix un mode comandes de configuració. D'aquesta forma, l'aplicació creada i que funciona a la placa LPC1769 accedeix a les comandes de funcionament del WiFly, per realitzar les configuracions i comandes necessàries. Una característica particular, a tenir en compte, d'aquests ports UART emprats, es que treballen amb un nivell lògic de 3,3 V. De fet, a pesar de tenir diferent voltatge d'alimentació la placa LPC1769, internament i en les seves entrades i sortides, treballa amb un voltatge de 3,3 V, que es estàndard per a molts sistemes encastats. Per altra banda, el WiFly disposa d'un pin, per tal d'executar un 'reset' al seu sistema, i hem fet servir un port GPIO de la placa LPC1769, per realitzar aquesta funció, al iniciar la connexió de comunicació amb el WiFly.

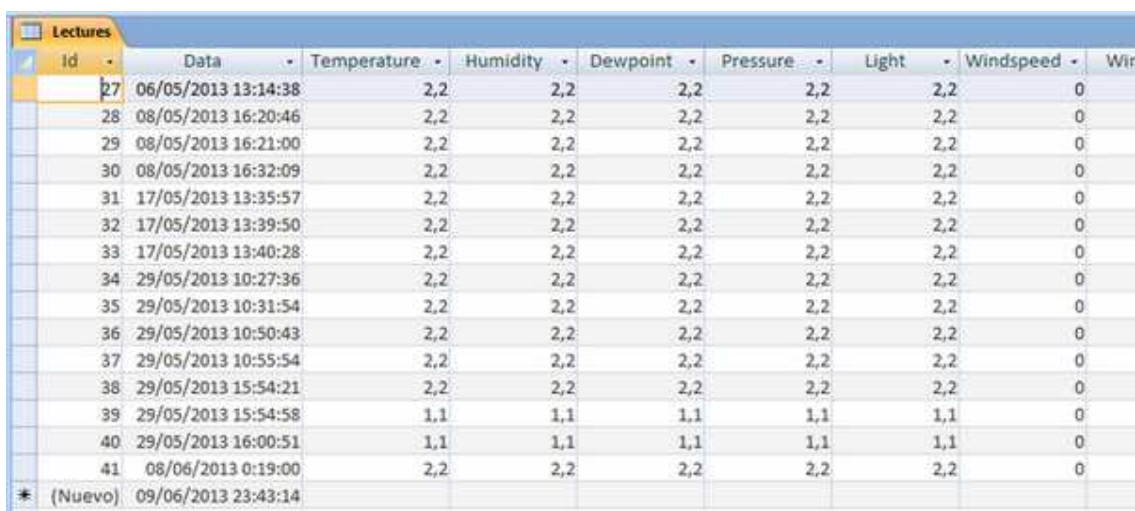
Amb el mòdul de sensors, tenim un cas semblant al del WiFly, aquest mòdul també disposa d'un port UART, d'igual especificacions, que hem emprat en seva la comunicació amb el port UART1 de la placa LPC1769. I com abans, el mòdul de sensors disposa d'un 'Firmware' que el gestiona, i també té un pin per fer 'reset', la diferència amb el WiFly, rau en que aquest sistema ofereix un menú d'opcions, en lloc d'un sistema de comandes. Per tant, l'aplicació creada, inicia el mòdul, però, treballa i configura el sistema, emprant les opcions del menú que tenim disponibles. Un cop fet això, el mòdul de sensors, envia pel port UART, cada X segons, la lectura dels sensors.

³⁹ De l'anglès "General Purpose Input / Output", fa referència a pins d'entrada sortida de propòsit general.

I per últim, tenim el adaptador USB – sèrie CP2102, que també ofereix un port UART que treballa a nivell lògic de 3,3 V. Aquí, hem fet la connexió amb el port UART3 de la placa LPC1769, que només serà de sortida, per els missatges que genera l'aplicació que hem carregat al sistema encastat. Així, al PC connectat a la part USB, només hem d'executar una aplicació Terminal, configurada correctament, per tal de veure els missatges.

3.2. Disseny interfície d'usuari.

En quant al servidor Web, no hem posat una pàgina que fes una consulta a la Base de Dades, ja que ho deixem per una implementació més endavant. Per contra, com que la DB emprada es el Microsoft Access, només cal obrir la taula corresponent i mirar les dades emmagatzemades que hi ha, i si es fa refrescar, es poden anar veient les noves que arriben. En aquest senti, ho deixem obert a qualsevol possible millora, i posem una imatge de dades de prova generades :



Id	Data	Temperature	Humidity	Dewpoint	Pressure	Light	Windspeed	Wir
27	06/05/2013 13:14:38	2,2	2,2	2,2	2,2	2,2	0	
28	08/05/2013 16:20:46	2,2	2,2	2,2	2,2	2,2	0	
29	08/05/2013 16:21:00	2,2	2,2	2,2	2,2	2,2	0	
30	08/05/2013 16:32:09	2,2	2,2	2,2	2,2	2,2	0	
31	17/05/2013 13:35:57	2,2	2,2	2,2	2,2	2,2	0	
32	17/05/2013 13:39:50	2,2	2,2	2,2	2,2	2,2	0	
33	17/05/2013 13:40:28	2,2	2,2	2,2	2,2	2,2	0	
34	29/05/2013 10:27:36	2,2	2,2	2,2	2,2	2,2	0	
35	29/05/2013 10:31:54	2,2	2,2	2,2	2,2	2,2	0	
36	29/05/2013 10:50:43	2,2	2,2	2,2	2,2	2,2	0	
37	29/05/2013 10:55:54	2,2	2,2	2,2	2,2	2,2	0	
38	29/05/2013 15:54:21	2,2	2,2	2,2	2,2	2,2	0	
39	29/05/2013 15:54:58	1,1	1,1	1,1	1,1	1,1	0	
40	29/05/2013 16:00:51	1,1	1,1	1,1	1,1	1,1	0	
41	08/06/2013 0:19:00	2,2	2,2	2,2	2,2	2,2	0	
*(Nuevo)	09/06/2013 23:43:14							

Figura 23 : Dades emmagatzemades a la DB

Per a la interfície de la connexió remota, que es pot efectuar quan el sistema es troba en mode de baix consum, hem seguit la mateixa tècnica aplicada al mòdul de sensors. És a dir, com que el mòdul de sensors funciona amb un menú d'opcions, hem generat també un menú d'opcions, per tal de configurar el funcionament de l'aplicació del sistema encastat. Això, ens ha permès poder fer un enllaç, de forma transparent, cap a la configuració del menú del mòdul de sensors. Dit d'altra forma, hem posat el menú del mòdul de sensors, com una opció del menú principal de configuració del nostre sistema. Per altra banda, hem gestionat els canvis en aquests menús, de forma que la nostra aplicació reconegui els paràmetres que han canviat. Així, un cop es surt de les pantalles dels menús de configuració, primer tanquem la connexió remota i posteriorment apliquem les accions seleccionades en el menú. L'únic inconvenient d'aquesta connexió remota, es que només funciona en un mode de baix consum, ja que els altres modes tallen les comunicacions dels ports UART, i de moment, no es pot solucionar.

3.3. Disseny aplicació al sistema encastat.

Per fer el disseny del nostre desenvolupament, hem aprofitat moltes de les funcionalitats que ens ofereix el O.S. FreeRTOS en quant a control de les tasques del sistema. Hem emprat molts dels recursos que hi ha disponibles, com cues, semàfors, suspensió o bloquejos, prioritats d'execució i temporitzadors. De fet, totes aquests recursos, ja els hem fet servir en la creació dels drivers, que hem posat a la nostra Llibreria, per a poder realitzar el desenvolupament de l'aplicació carregada al sistema encastat que tenim.

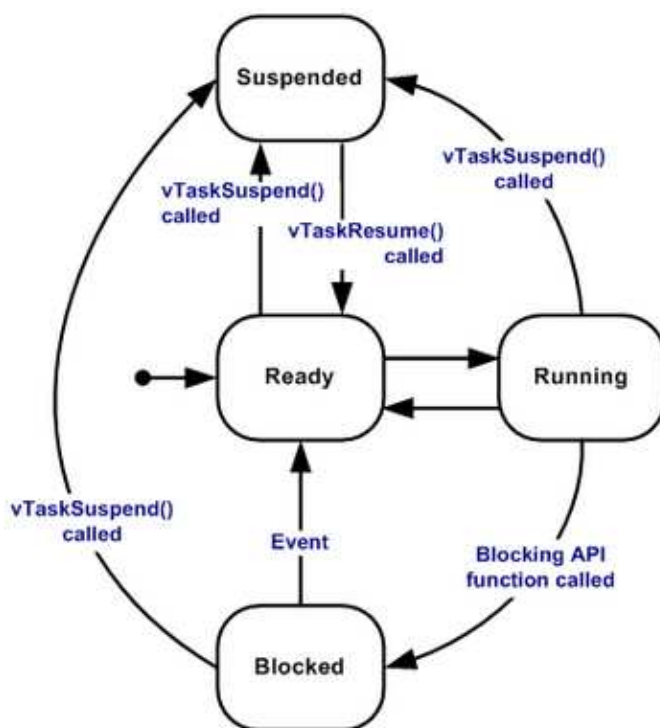


Figura 24 : Transicions d'estat de tasques a FreeRTOS

Per altra banda, com que la solució ha de funcionar en el microcontrolador LPC1769, hem emprat la Llibreria CMSIS v2, que es tracta d'una capa d'abstracció del maquinari i també ofereix funcionalitat per activar i desactivar opcions. És a dir, hem combinat les dues coses, el O.S. FreeRTOS i la Llibreria CMSIS v2, d'aquesta forma, obtenim una aplicació compacta, controlable i que ens ofereix molta funcionalitat. Així, tenim la part d'entorn de treball, control i gestió general de l'aplicació, que funciona sota FreeRTOS, i la part d'iniciació, configuració o parametrització del maquinari, que fa servir la Llibreria CMSIS v2.

El fet anterior, defineix i condiciona, la forma en que s'ha dissenyat l'aplicació del sistema encastat. Per una costat, tenim tots els requeriments del programari en quant a recursos del sistema i forma de treballar. I per l'altre, tots els requeriments del maquinari, per ser inicialitzat i configurat de forma correcta, per tal de que funcioni de la forma esperada i permeti controlar els dispositius que se li poden connectar. Una imatge que ho resumeix, pot ser la següent :

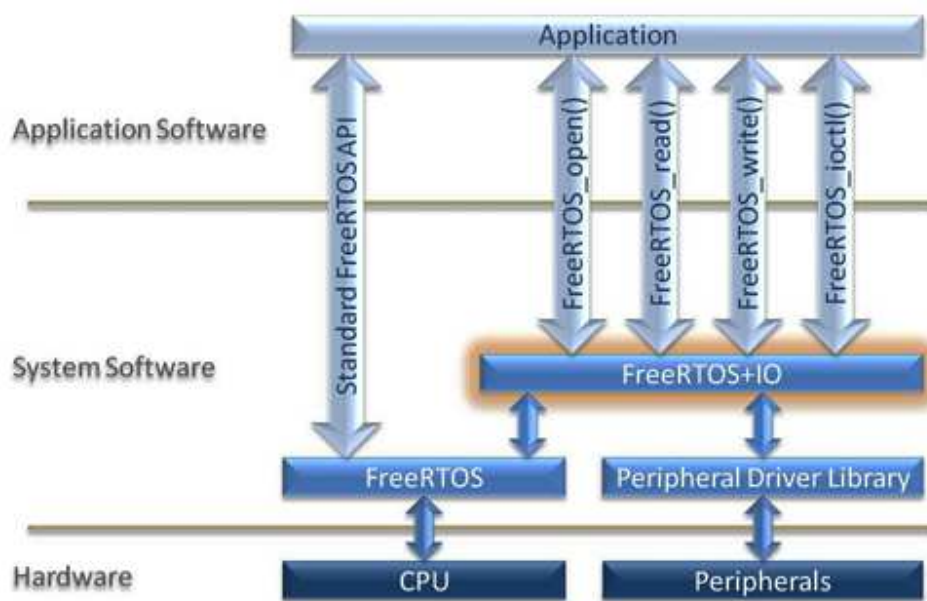


Figura 25 : Combinació FreeRTOS i Llibreria maquinari

Aquí, podem veure clarament diferenciada, la part principal que treballa amb la CPU, que seria el programari, però, es comunica amb la Llibreria, per treballar amb el maquinari. Normalment, això sol ser una cosa habitual, en tots els O.S., però la diferència, es troba en que en el nostre cas, es el fabricant del producte qui ens ofereix la Llibreria. Per tant, això permet que l'entorn de FreeRTOS⁴⁰ pugui funcionar amb molts sistemes encastats diferents.

Com que la Llibreria CMSIS⁴¹ v2, ofereix un accés a molt baix nivell, hem creat una altra Llibreria amb els drivers, dels dispositius que fa servir la nostra aplicació. I per tal de no interferir en el funcionament de l'entorn FreeRTOS, hem combinat les dues parts, en el disseny d'aquesta nova Llibreria. Per tant, ens fa de pont entre el programari i el maquinari, de forma que ens facilita l'accés i control dels dispositius, dins de l'entorn en que han de treballar.

4. Descripció detallada del sistema.

Fins ara, hem vist a trets generals, tot el procés, parts generades, requeriments tècnics i disseny, que comporta la creació de la nostra aplicació de sistemes encastats. En els propers apartats, entrarem en el detall, de tota la part tècnica del programari, i com interactua amb el maquinari que intervé en aquest projecte. D'aquesta forma, podrem entendre millor el seu funcionament, i podrem relacionar o aclarir punts, que han estat explicats una mica per sobre, en apartats anteriors. Donat que es tracta d'una part eminentment tècnica, farem servir mots o expressions, pròpies d'aquesta entorn especialitzat en sistemes encastats.

⁴⁰ De l'anglès "Free Real Time Operating System", Sistema operatiu de temps real lliure.

⁴¹ De l'anglès "Microcontroller Software Interface Standard", interfície estàndard de programari de microcontroladors.

4.1. Creació de l'estació meteorològica.

En aquest apartat, detallarem tots els processos, interaccions i programari creat o que intervé, en la creació de l'aplicació de l'estació meteorològica. Degut a la composició del projecte, podem veure que tenim dos parts clarament diferenciades, i aquesta es la part que correspon al sistema encastat, on hem fet el desenvolupament de programari. Per altra banda, com que s'ha seguit un procés evolutiu, fins arribar finalment a la construcció de l'aplicació del sistema encastat, exposem el que hem anat creant, fins arribar a aquest punt. Per tant, els apartats següents, detallen tot el procediment emprat en aquest projecte, en quant a programari

4.1.1. Implementar controladors al sistema LPC1769.

Com hem indicat en la part de disseny, tenim un entorn de programació, que es compon de dos parts, la del control del maquinari i la del control del programari. La primera es correspon a la Llibreria CMSIS v2, i la segona al O.S. FreeRTOS, que treballen conjuntament. Degut al baix nivell d'accés que ens ofereix la primera, i poca implicació que té en els processos que es donen en la segona, inicialment, s'ha creat una nova llibreria, amb els drivers dels dispositius que tenim al inici. Els passos següents, per realitzar aquesta implementació, els exposem a continuació.

Inicialment, s'ha incorporat un driver extern, que gestiona i controla els ports UART de la placa LPC1769 i ens facilita la comunicació amb els dispositius que també en tenen i es connecten al sistema. Aquí, podem identificar inicialment, el WiFly i el adaptador USB – sèrie CP2102, que són emprats per comunicació amb la xarxa i Debug/Log, respectivament. A pesar de que ja era operatiu el driver, tal com estava, hem afegit i modificat algunes funcions, per tal de millorar les seves prestacions i que s'ajustés al funcionament i rendiment, que volíem obtenir.

En el següent pas, s'ha fet la construcció d'un driver que ens donés la funcionalitat de la funció 'printf' del llenguatge C, però, emprant com a sortida, qualsevol dels ports UART del sistema. Així, aquest driver, fa una capa sobre el driver anterior, per tal d'enviar missatges de text per els ports UART de que disposem. Com que això es una característica funcional de les comunicacions per ports UART, ens serveix per comunicar amb qualsevol dispositiu que en disposi. En aquest punt, a nivell bàsic, ja hem pogut enviar missatges de Debug/Log per al CP2102, i emprar el WiFly de forma rudimentària. De fet, aquest driver, ha estat fonamental per poder fer el desenvolupament dels altres que s'han creat a la Llibreria d'utilitats.

Amb l'anterior driver finalitzat, s'ha procedit a fer un driver per gestionar i controlar el WiFly que tenim connectat al sistema, via port UART. Com que aquest dispositiu permet entrar a un mode comandes per configurar-lo i emprar-lo, hem creat tot un seguit de funcions que ho fan. De primer, cal enviar uns caràcters específics per entrar al mode comandes, i posteriorment, ja es poden fer servir les comandes específiques que surten en un manual d'utilització. Degut a que aquest

funcionament, es interactiu, és a dir, hi ha una resposta per cada comanda enviada, hem tingut que llegir la resposta que ens donava, en cada cas. Per tant, hem implementat un sistema de recuperació d'informació dels ports UART, de que disposa la placa LPC1769, amb un temps d'espera màxim per llegir la resposta. D'aquesta forma, hem creat totes les funcions, i combinat algunes d'elles, tenint en compte el funcionament anterior, per tal d'obtenir un driver que ens donés solució a les necessitats de comunicació. Per altra banda, com que una de les particularitats del WiFly, es que realitza connexions transparents, és a dir, un cop connectat a un servidor, el que rep del servidor ho reenvia pel seu port UART, i a l'inversa, el que rep pel port UART, ho reenvia al servidor, s'han implementat unes funcions que aprofiten aquest fet. Així, aquestes funcions, fan comunicar la placa LPC1769 amb qualsevol servidor de la xarxa o a l'inversa, la xarxa amb un servei que doni la aplicació de sistemes encastats, que hi ha a la placa LPC1769.

En un pas posterior, donat que el funcionament de les aplicacions FreeRTOS (per tasques a temps real) poden interferir en l'execució de les funcions creades en els drivers, s'ha implementat un sistema de suspensió o bloqueig de tasques, durant l'execució del codi de les funcions, als drivers necessaris. Per tal de comprovar tot el funcionament, que ofereixen aquests drivers, s'han fet varies aplicacions de test, que verifiquen les funcions creades. En aquest cas, arribats a aquest punt, posem el diagrama de blocs, d'un exemple generat :

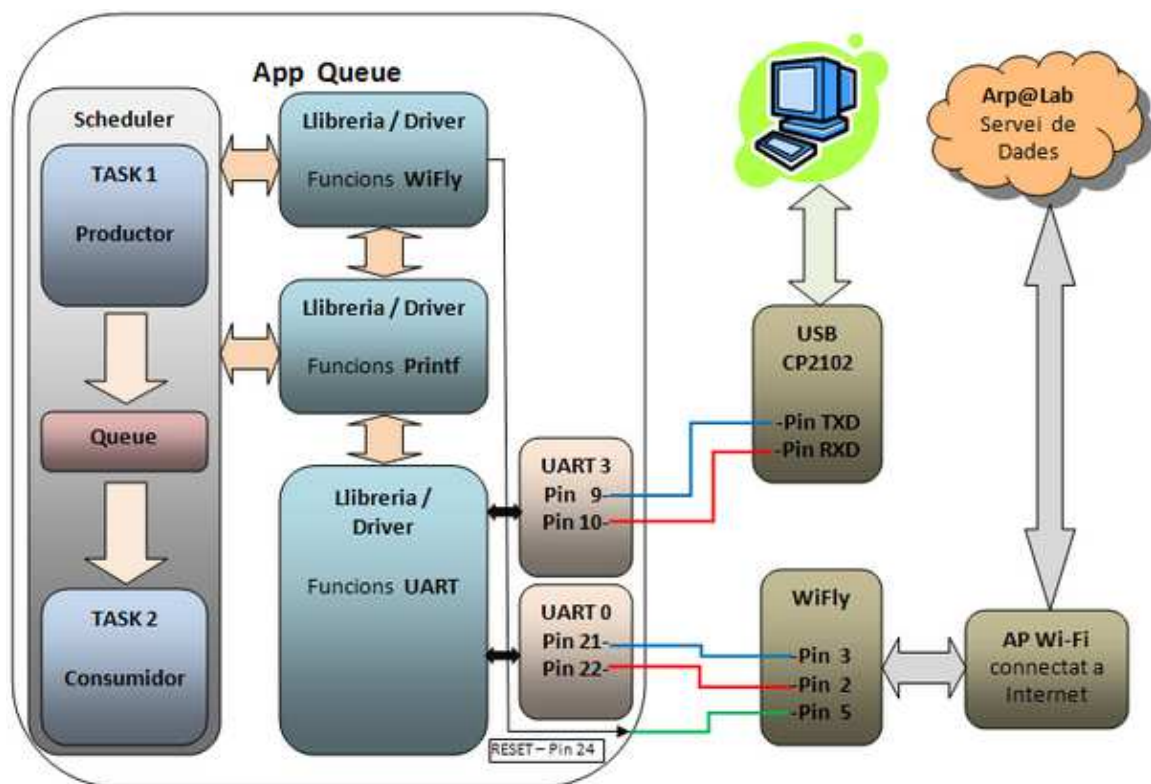


Figura 26 : Diagrama de blocs d'aplicació de test

Donat que s'ha tingut que verificar la comunicació amb un servidor d'Internet, amb una aplicació que es va demanar en una PAC, s'ha creat un driver específic per aquest fet. El nou driver, fa una capa sobre el driver del WiFly, i treballa amb les seves funcions, per aconseguir la funcionalitat necessària. Per a la seva creació, també s'han tingut en compte els requeriments que ja complien els altres drivers, en quant a temps d'espera de resposta, i suspensió o bloqueig de tasques. En aquest cas, s'ha tingut que recuperar una dada d'Internet, operar amb ella i uns valors del WiFly, i tornar-la a enviar al servidor d'Internet. Per tant, com que hi ha dues operacions diferents, s'han generat dues funcions ajustades al cas de cada una d'elles, i com que es pot donar un error de comunicació o de dades, s'ha posat unes funcions de controls d'errors, en aquesta situació.

En aquest punt, també s'ha fet una aplicació de test, per tal de comprovar tot el funcionament, que ofereixen aquests drivers. A més, un requisit d'aquesta aplicació, es que ha de funcionar en dues tasques i fent servir el sistema de bloquejos de FreeRTOS. I s'han creat varies aplicacions, emprant diferents mètodes o sistemes disponibles a l'entorn de FreeRTOS. Com que en el punt anterior hem posat un diagrama de blocs, aquí farem servir un diagrama de flux de l'aplicació, per tal de veure els processos que succeeixen :

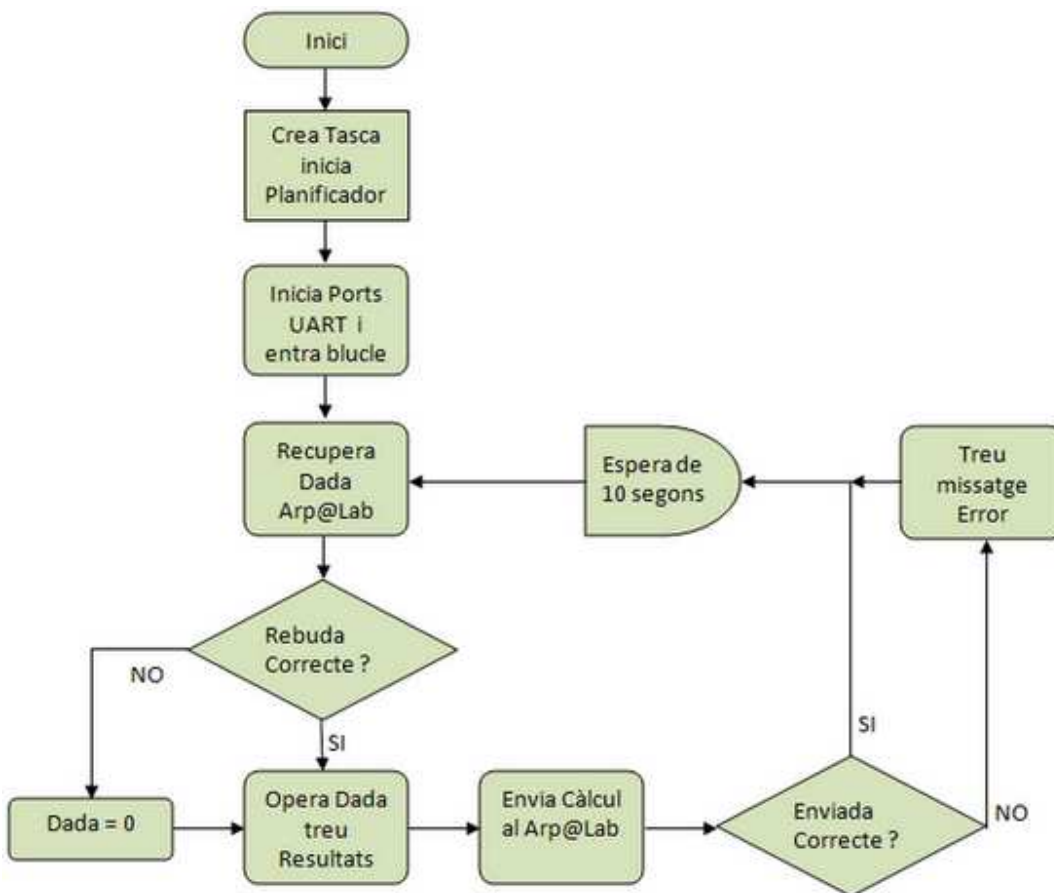


Figura 27 : Diagrama de flux d'aplicació de test

4.1.2. Construcció del prototip de l'aplicació.

Un cop s'ha verificat el funcionament de tots els drivers anteriors, s'ha iniciat la segona fase del projecte. Primer s'ha seleccionat el projecte a desenvolupar, que ha estat "estació meteorològica de baix consum", i posteriorment s'ha adquirit el material necessari per realitzar-la. En el nostre cas, s'ha adquirit un mòdul de sensors, de la companyia 'Sparkfun', que es comercialitza amb el nom "USB Weather V3". Encara que hi posi USB, ofereix un port UART, que inicialment està destinat a un mòdul WiFly, però com que ja en tenim un al sistema, no ens cal posar-lo aquí, i fem servir aquest port UART, prèvia selecció en un commutador que hi ha per aquests casos.

En un primer moment, desenvolupem el driver del mòdul de sensors, fent una capa al driver 'printf', i tenint en compte tot el que hem après amb el driver del WiFly, ja que aquest, també porta un 'Firmware' que el gestiona. I també, com els altres drivers, li apliquem el requisits de temps d'espera de resposta, i suspensió o bloqueig de tasques. Com que aquest mòdul disposa d'un menú d'opcions, al que s'hi accedeix enviant també uns caràcters específics (Control-Z), adaptem les funcions creades a aquesta forma de treballar. Per altra banda, cada cop que es surt del menú d'opcions, o es fa un 'reset' del sistema del mòdul de sensors, aquest envia pel seu port UART una lectura de les mesures dels sensors, i això, genera una IRQ al mode de baix consum 'Sleep'. Aprofitant el fet anterior, adaptem les funcions creades, per tal de recuperar la informació que s'envia per port UART del mòdul. Per altra banda, com que el sistema de menú de configuració, que té aquest mòdul de sensors, és molt senzill i pràctic, desenvolupem una funció que hi interactua, fent de servidor 'Telnet' cap a un port UART. Això, ens ha permès emprar aquesta funció, per poder configurar remotament el mòdul de sensors, recuperant els canvis en unes variables concretes.

```
*****
* HELLO ... INICI SESSIO REMOTA HEATHER V3 *
*****

SparkFun USB Weather Board V3 firmware version 1.1
free RAM: 181 bytes

1. Data format (CSV)
2. General units (SI)
3. Sample rate (10)
4. Pressure units (mbar)
5. Pressure type (absolute)
6. Station altitude (695 meters)
7. Baud rate (9600 baud)
8. Zero rain counter
9. Weather Meters attached (yes)
X. Exit (don't save changes to EEPROM)
S. Save (save changes to EEPROM)
```

Figura 28: Pantalla configuració mòdul de sensors

Quan hem acabat el driver anterior, ens hem adonat de que tots els drivers desenvolupats, tenien unes necessitats de Debug/Log i unes funcions comunes a tots ells. Per tant, hem creat un driver

'logutil' que satisfà les necessitats dels altres drivers i centralitza les funcions d'ús comú, en un sol lloc, per reutilitzar codi, i millorar el funcionament. D'aquesta forma, aquest driver, es relaciona amb tots els que fan servir, poc o molt, les funcions que te implementades.

En següent pas, ha estat la incorporació d'un driver, que ens proporciona la funcionalitat dels modes 'Idle' de que disposa la placa LPC1769. I en els exemples avançats de NXP, per a les plaques LPC17xx, hem trobat un codi que hem pogut aprofitar i adaptar als nostres requeriments. Per tant, hem afegit un driver 'Sleep', amb els canvis i modificacions necessaris, a la Llibreria d'utilitats que ja tenim.

Tal com hem fet en l'apartat anterior, també hem generat unes aplicacions de test, per verificar el funcionament dels drivers anteriors. Aquí, ens hem adonat de que si fallava la recepció de dades del mòdul de sensors, el sistema no es reactivaria mai del mode de baix consum, en el que deixem el sistema, a l'espera de noves dades. Per tant, hem desenvolupat un driver 'rtc' que fa d'alarma i desperta el sistema, si uns segons després del temps límit, no han arribat les dades, que fan funcionar el procés de recuperació i enviament de lectures. Igual que abans, hem trobat a exemples FreeRTOS per plaques LPC17xx, un codi que hem pogut aprofitar i adaptar a les nostres necessitats. A part de modificar aquest driver, hi hem afegit una funció, per tal de que treballés amb un comptador de segons, tal com ho fa el mòdul de sensors. Donat que aquesta alarma, funciona amb tots els modes de baix consum, de que disposa la placa LPC1769, ens ha permès enfocar una altra forma de fer les lectures als sensors. D'aquesta forma, hem pogut fer servir els altres modes 'Idle' que te el sistema, i emprar l'alarma de temps que hem implementat, per despertar a la placa LPC1769. I com que per aquests modes 'Idle', s'ha de reiniciar els rellotges dels ports UART, hem aprofitat el fet de reiniciar els dispositius, per forçar una lectura al mòdul de sensors. Així, tenim una altra forma de treballar, amb l'aplicació final, que redueix més el consum del sistema.

Per altra banda, també hem implementat un driver 'extint', que controla un senyal extern, genera una IRQ i reactiva el sistema. La raó de crear aquest driver, ha estat de que amb els altres modes 'Idle' de la placa LPC1769, el fet de rebre una informació per port UART, no reactiva el sistema. Per tant, no es pot realitzar una connexió remota, en aquests modes 'Idle', i això podria ser una possible solució per poder fer-ho. Aquest driver funciona correctament amb tot els modes de baix consum i hem pogut verificar que reactiva el sistema. Però, malauradament, el WiFly no disposa d'un pin que faci aquesta funció.

I per últim, hem creat també un driver 'wdt' (Watchdog⁴²), perquè ens proporciona una altra forma de controlar el sistema davant de falles, però no hem pogut acabar d'afinar el seu funcionament.

Ara que ja tenim tota la Llibreria de drivers complerta, hem passat a la construcció de l'aplicació del sistema, que exposem en el següent apartat.

⁴² De l'anglès, literalment significa gos guardià, es sol emprar per controlar funcionament d'aplicacions.

4.1.3. Creació de l'aplicació del sistema.

Un cop finalitzada la llibreria d'utilitats, hem començat la creació de l'aplicació del sistema, pròpiament dita 'Station'. Aquesta aplicació, a part dels recursos esmentats de FreeRTOS i CMSIS, fa servir sobre tot, la Llibreria de drivers que hem creat en els apartats anteriors. Així, tenim una relació molt estreta, entre l'aplicació i la llibreria, per el que posem el diagrama de blocs, de com es la seva interacció i dependència :

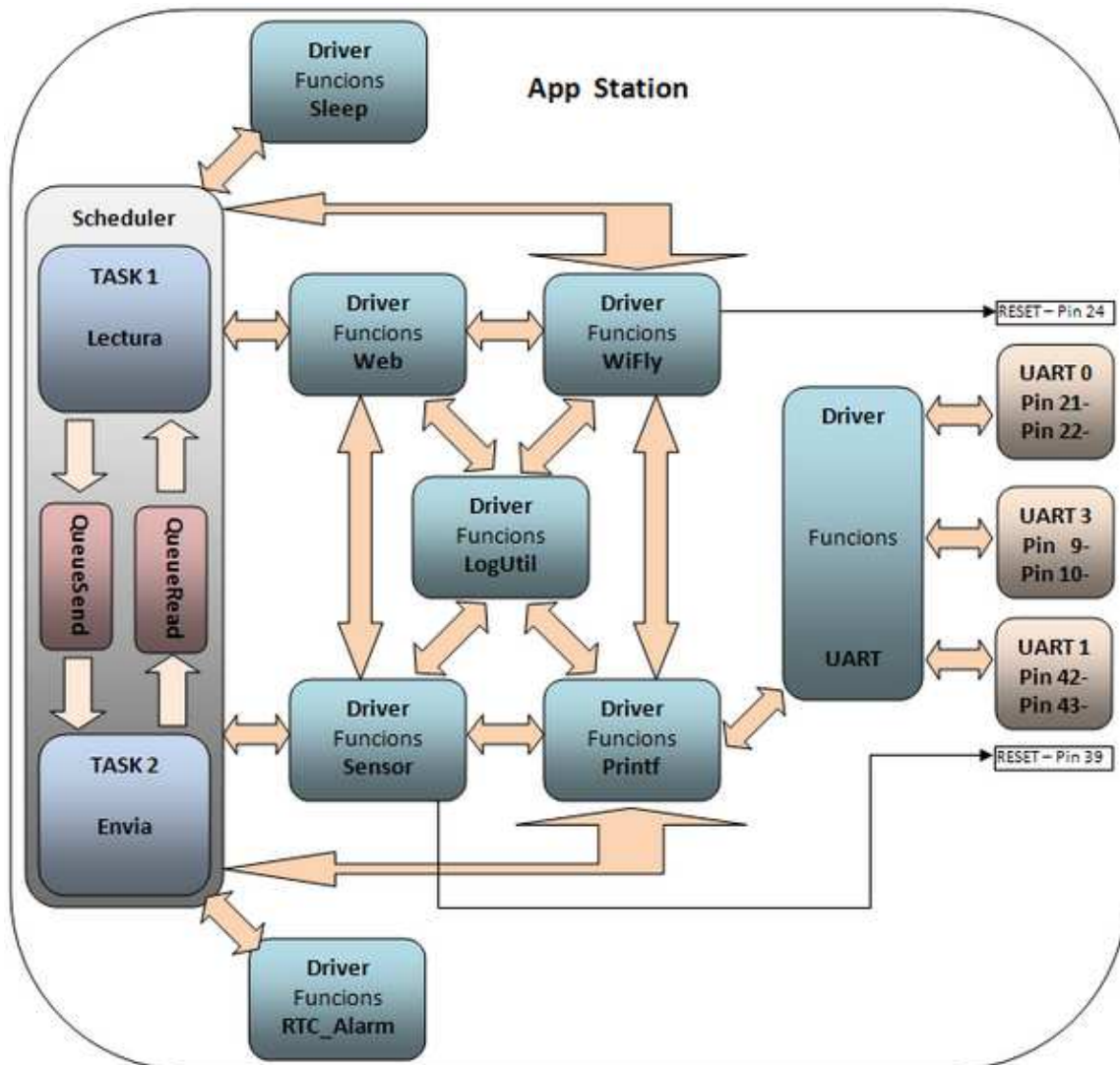


Figura 29 : Diagrama de blocs aplicació Station

En la figura anterior, podem veure de forma resumida, totes les relacions entre els drivers de la llibreria, que hem exposat en els apartats anteriors, i amb el nucli central de l'aplicació creada. I es pot veure que gràcies a emprar capes, no fa falta enllaçar el driver UART amb l'aplicació. Per altra banda, hem posat els pins GPIO, que fan servir els drivers, per forçar una 'reset' als corresponents mòduls. També temin les tasques internes de l'aplicació i el detall dels ports UART emprats.

Per a la creació de l'aplicació, primer hem inclòs tots els drivers necessaris, per tal de poder accedir a les funcions que hem creat. Després, hem definit totes les variables d'ús particular en l'aplicació, les funcions internes de l'aplicació, i les funcions de les tasques emprades, per obtenir la funcionalitat esperada. Per defecte, fem que el sistema utilitzi el mode 'Idle' de Sleep.

Al inici, hem creat dos cues, una per cada tasca que hem creat posteriorment, i hem engegat el planificador de FreeRTOS. S'ha de tenir en compte, que hem fet que les cues bloquegin les tasques, fins a rebre una dada d'activació. Com que a igual prioritat, l'última tasca es la primera en iniciar-se, hem posat al inici d'aquesta tasca, les funcions d'iniciació dels perifèrics emprats, es a dir, el WiFly, el mòdul de sensors, el adaptador CP2102 i el rellotge d'alarma. Al iniciar el WiFly, hem fixat la seva configuració d'accés a la xarxa, i al iniciar el rellotge, hem recollit la configuració de lapse de temps del mòdul de sensors i configurat la alarma. Això, i iniciar el mòdul de sensors, fa que es realitzi una lectura inicial, pel que acte seguit, hem saltat a la tasca d'enviament de dades, enviant una dada de correcte a la seva cua.

Un cop a finalitzat l'enviament de dades, controlem el resultat si ha esta correcte o no, per saber si falla la xarxa o el servidor. El resultat el retornem per cua a l'altra tasca de lectura i així s'activa, i el primer que fa, es controlar si hi ha hagut un error d'enviament, per incrementar un comptador global. Després s'activa el mode 'Idle' que està programat segons la configuració, i l'aplicació es queda a l'espera que una IRQ, que desperti al sistema.

Un cop despertat, primer es mira amb quina IRQ ha estat, si es per rebre dades pel port UART del mòdul de sensors, realitza una lectura de les dades, i si es pel port UART del Wifly, salta a la funció de servei 'Telnet' de configuració. En el cas de la que IRQ sigui per alarma de temps, comprova si el mode 'Idle' es el corresponent, si es així, reinicia els rellotges i els perifèrics, sinó no fa res. Després es comprova si amb el servei 'Telnet' s'ha sol·licitat un reinici del sistema, i actua en conseqüència. En el següent pas, es verifica si es superat el comptador global d'errors del sistema i fa un reinici del sistema, o el posa en mode molt baix consum, segons el que s'ha triat a la configuració per defecte. En el pas següent, es posa a zero el indicadors de IRQ, i es verifica si s'ha fet una lectura de les dades del mòdul de sensors. Si hi ha dades, es salta a la tasca d'enviar, i en cas contrari, s'acumula un error, i es força un 'reset' del mòdul de sensors. Es torna a verificar si hi ha lectura de dades, si es que si es salta a la tasca d'enviar, i si no en hi ha, treu missatge d'error, i es posa una dada de falla a la cua i es salta a la tasca d'enviar.

El primer que fa la tasca d'enviar, es comprovar la dada que rep, per saber si ha d'enviar les dades o controlar l'error que li arriba. Si ha d'enviar les dades, ho fa i comprova que s'ha fet correctament, en cas contrari acumula error i treu missatge. Si ha de controla l'error, només treu el missatge de que no pot enviar dades, i retorna per la cua correctament a la tasca de lectura.

Per veure de forma més clara tot el que hem exposat fins ara, adjuntem un diagrama de flux de l'aplicació:

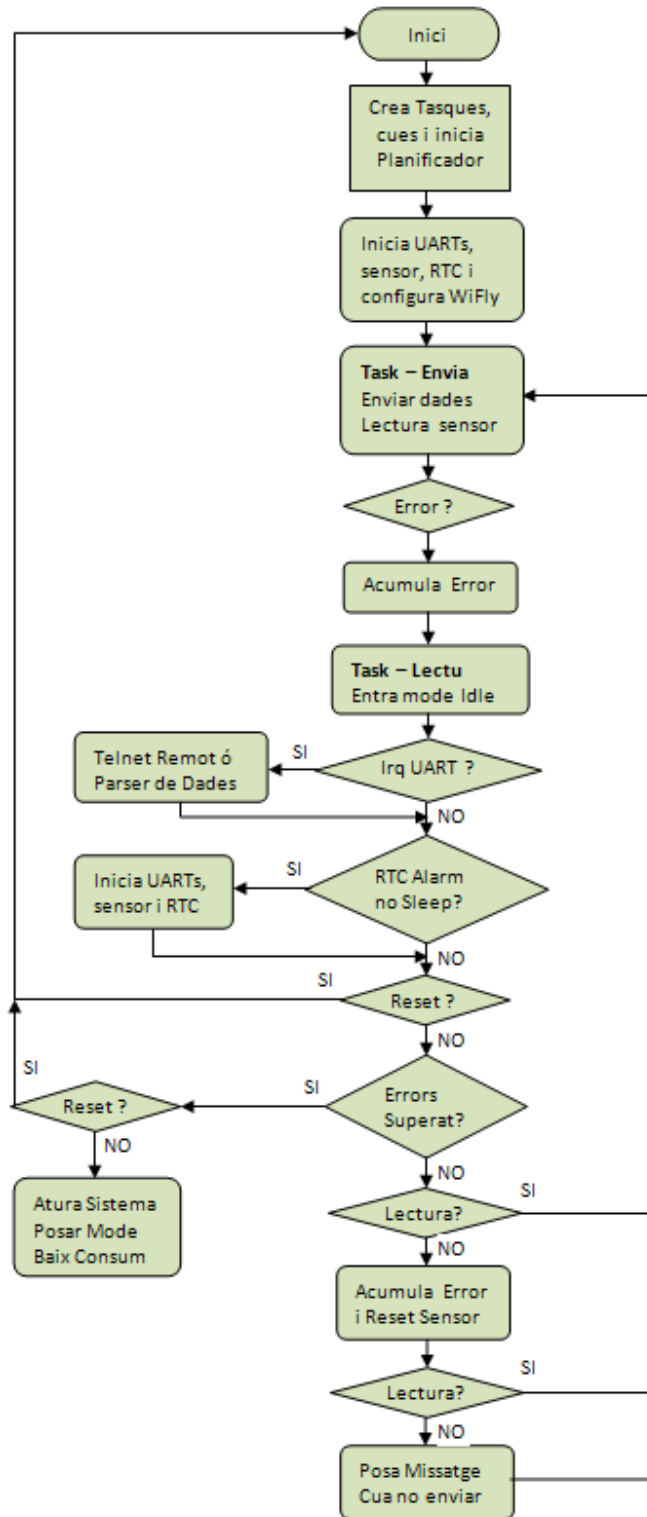


Figura 30 : Diagrama de flux aplicació Station

Es pot observar en el diagrama de flux, que el procés, si no hi ha errors, funciona en un bucle, tal com sol ser una aplicació feta en FreeRTOS.

4.2. Creació servidor Web amb Base de Dades.

En quant al servidor Web, hem emprat la tecnologia de Microsoft ASP.NET per crear una pàgina web que ofereixi la funcionalitat que ens fa falta en el projecte. Així, després només cal posar aquesta pàgina dins del servidor Web IIS⁴³, com a pàgina per defecte a servir, per a que funcioni de forma prevista. En el nostre cas, hem triat una altra opció, emprar el servidor gratuït de pàgines fetes amb ASP.NET, que es diu “UltiDev Cassini 2.0”. En aquest servidor, només cal fer una petita configuració sobre el sistema, per tal de donar el servei, tal com es pot veure en la imatge adjunta :

Figura 31 : Configuració del Servidor Web

Es pot observar, que la ubicació per defecte, que s'ha triat per funcionar el servei de la pàgina Web, connectada a la Base de Dades, ha de ser : “C:\WUHU”, que ha d'incloure la DB en la base, i en una sots carpeta les parts de la pàgina Web.

El primer pas, per a la creació de la pàgina Web, ha estat definir els camps de la DB a la que es connectarà la pàgina, i crear la DB segons les especificacions donades. Després, s'ha creat una pàgina ASP.NET, que permetés l'entra manual de dades en aquesta DB, emprés una funció per guardar les dades a la DB, i al final s'ha verificat el seu funcionament. Com que la tecnologia emprada, es caracteritza per oferir mètodes i funcions que controlen tots els passos, en el procés de servir o interactuar amb una pàgina Web, hem aprofitat aquest fet. Per tant, seguidament, s'ha creat una funció que controlés la petició de la pàgina Web, i que actués segons la nostra definició d'operació. En aquest punt, primer s'ha verificat si hi havia paràmetres en la petició efectuada, si no

⁴³ De l'anglès “Internet Information Services”, Serveis d'Internet Information Server.

el hi ha, es mostra la pàgina d'entrada manual de dades, definida abans. En cas contrari, si hi ha paràmetres, es revisa que hi hagi com a mínim uns d'obligatoris, si no hi son, es mostra pàgina d'error de petició. Si hi son, es recull la informació que porten aquests paràmetres, es formata correctament per si el separador decimal no es correcte, i s'introdueix la informació a la DB. I al final, es retorna la corresponent pàgina de OK, per al funcionament correcte.

Com que no hem posat una pàgina Web que fes una consulta a la Base de Dades, donat que la DB emprada es de Microsoft Access, només cal obrir la taula corresponent i mirar les dades emmagatzemades que hi ha. A més, com que no es bloqueja la DB, per al seu ús, es pot tenir oberta sense cap problema, i si es fa refrescar, es poden anar veient les noves que arriben. Per altra banda, degut a les eines i utilitats que ofereix el Microsoft Access, aquestes dades, poden ser reutilitzades amb molta facilitat.

4.3. Oferir connexió de configuració remota.

Com que hem vist que l'aplicació generada, estarà la major part del seu temps en mode 'Idle', hem decidit aprofitar-ho, per oferir servei de connexió de configuració remota de l'aplicació. En aquest sentit, ja sabem que el fet de rebre informació per un port UART, desperta la placa LPC1769, per tant, permet efectuar una connexió remota, emprant el WiFly com a pont de comunicacions. Però, això només funciona per al mode Sleep, els altres modes de que disposa la placa LPC1769, tallen les connexions dels ports UART. Així, hem posat a l'aplicació el mode 'Idle' de Sleep com a per defecte, i si es canvia, es perdrà la possibilitat de rectificar-lo, fins que es faci un 'reset' del sistema.

En un primer moment, hem generat un menú d'opcions, del mateix estil que el que fa servir el mòdul de sensors, per tal de configurar el funcionament de l'aplicació del sistema encastat, emprant la IP i port 2000 del WiFly. I hem creat una funció que ofereixi el servei d'aquest menú, seguint el mateix procediment, que s'ha emprat en la funció de 'Telnet' remot, creada al driver del mòdul de sensors. Una imatge del que es pot veure, al iniciar la connexió, és :

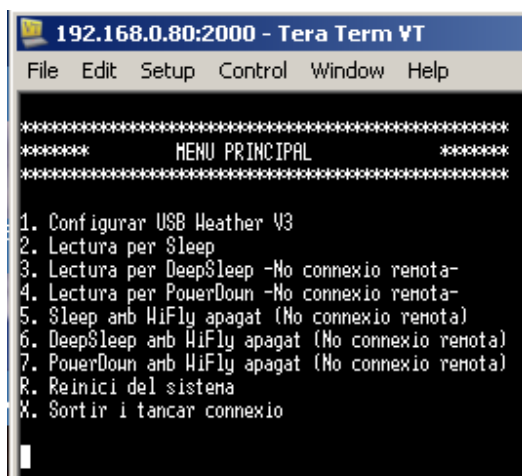


Figura 32 : Pantalla configuració remota aplicació

D'entrada, tenim que hem posat l'accés al menú del mòdul de sensors, com una opció del menú principal de configuració del nostre sistema. Per altra banda, hem posat unes variables al sistema, que indiquen l'opció escollida de configuració. Al sortir d'aquest menú, primer es força el tancament de la connexió, i després s'utilitzen aquests valors, per realitzar els canvis de configuració pertinents, en el funcionament de l'aplicació dins de la tasca de lectura.

Donat que la configuració remota només funciona en el mode 'Idle' de Sleep, hem intentat millorar aquesta funcionalitat, afegint un driver, que controlés una 'EINT0' (interrupció externa, port 0). Amb això, és possible despertar al sistema i seguir oferint la possibilitat de realitzar una connexió remota de configuració, per als altres modes 'Idle'. Malauradament, el WiFly no genera un senyal que indiqui que es vol efectuar una connexió remota, i per tant, no hem pogut emprar aquesta funció dins del sistema. Esperem que en noves versions del 'Firmware' del WiFly hi hagi aquesta funció, per poder-la emprar, i ampliar la funcionalitat del nostre sistema.

Per altra banda, al veure que no podíem emprar el WiFly, hem decidit ampliar la funcionalitat del sistema, afegint l'opció de posar-lo en mode 'Idle'. D'aquesta forma, encara aconseguim un consum més baix en el nostre sistema, i ampliem l'oferta dels modes de treball disponibles

5. Viabilitat tècnica.

Com que volem desenvolupar una petita estació meteorològica amb sensors, ens cal veure que pot fer el material que ja tenim, quines limitacions podem tenir i que ens cal implementar. Primer, ens cal tenir en compte els consums que ja te associats la placa LPC1769, de forma normal i amb els diferents modes 'Idle' de treball de que disposa :

mBed Power Supply consumption @ 5.00V			
Type	Draw (mA)	Consumption (W)	% Normal
Normal mode (while(1){} loop)	138.5 mA	0.6925 W	100%
All Peripherals On (PHY on, ETH in)	200.6 mA	1.103 W	144%
128 MHz, All Peripherals On (PHY on, ETH in)	220.6 mA	1.003 W	159%
All Peripherals Off (PHY off)	101.9 mA	0.5095 W	73.5%
40MHz, All Peripherals Off (PHY off)	81.6 mA	0.408 W	58.9%
Brown Out Disabled (Normal)	138.2 mA	0.691 W	99.7%
Sleep	127.0 mA	0.635 W	91.6%
Deep Sleep	96.5 mA	0.4825 W	69.7%
Power Down	96.3 mA	0.4815 W	69.5%
Deep Power Down	91.3 mA	0.4564 W	65.9%
Deep Power Down (PHY off)	57.7 mA	0.2885 W	41.6%

Figura 33 : Taula de consums placa LPC1769

Això, es una cosa molt important a tenir en compte, ja que segons la imatge anterior, de la URL : <http://mbed.org/users/no2chem/notebook/mbed-power-controlconsumption/> tenim un consum del sistema bastant elevat. A més, encara hem d'afegir al consum total del sistema, el consum del WiFly i el consum del mòdul de sensors emprat. Per altra banda, per tal d'activar els ports i les funcions del hardware de la placa LPC1769, s'utilitza com a guia, el document "UM10360.pdf" de NXP, que està en sintonia amb les opcions i funcions de la Llibreria CMSIS v2.

D'entrada, veiem que el mòdul de sensors "USB Weather V3" ha de complir una sèrie de coses per tal de ser compatible amb l'entorn que tenim. Per tant, fem una taula amb les dades tècniques del mòdul de sensors, els valors o opcions oferts, i si ens serveix o no al nostre sistema.

Especificacions tècniques - Requeriments	Valors / Opcions	Compliment
Disposar d'alimentació a 5 V ó 3,3 V	USB ó 3,7 – 12 V	SI
Tenir baix consum en mode Operatiu	15 mA	SI
Tenir port UART serial TTL per comunicació dades	1 TTL 3,3 V	SI
Poder configurar port UART a diferents velocitats	300 – 115200 bps	SI
Disposar d'opció de reinici ('reset') per senyal Low	Port ISP	SI
Disposar d'opcions de configuració	Menú	SI
Poder canviar freqüència de Lectures	Màx. 4294966 s.	SI
Tenir dades Lectures en altres mètriques	ENG i S.I.	SI
Disposar de format de dades de lectures compacte	CVS, ANSI i LCD	SI , CVS
Tenir en compte alçada sensors per pressió atmosfèrica	Entrada alçada	SI
Configurar pressió atmosfèrica absoluta o relativa	Opció Menú	SI
Emprar mode 'Idle' entre lectures	No Operatiu	NO
Poder ampliar funcions amb altres sensors meteorològics	WIND i RAIN	SI
Poder iniciar comptador de pluja	Opció Menú	SI
Tenir en compte nous sensors a dades Lectures	Short i Long	SI

Figura 34 : Taula dades tècniques - requeriments mòdul sensors

Si observem amb detall la taula anterior, podem veure que només hi ha una opció que no es compleix, però donat el baix consum del mòdul de sensors, en comparació amb la resta del sistema, això gairebé no té importància. Per altra banda, al poder emprar un port UART, ens va perfecte per comunicar el mòdul de sensors amb la placa LPC1769, que farà de nucli central.

Un cop s'ha configurat el mòdul de sensors, amb les característiques que ens interessa, aquest, efectua una lectura de les mesures dels sensors cada X segons i les envia per el seu port UART, si no hi ha cap alteració en el funcionament. Per altra banda, cada cop que s'inicia o es fa un 'reset' del mòdul, la primera operació que es realitza, es fer un lectura de les mesures, i enviar-la pel port UART. I el mateix passa, si entrem al menú de configuració del mòdul de sensors, al sortir, la primera operació que fa, es una lectura i enviament de dades.

Els fets anteriors, ens ha permès veure, diferents formes de treballar amb les lectures d'aquest mòdul de sensors. La normal, es llegir les dades cada X segons que son enviades, i que genera una IRQ al port UART. La segona, aprofitar el fet d'entrar al menú, per recuperar algun valor de la seva configuració, i recuperar les dades de la lectura feta al sortir del menú. I finalment, recuperar les dades de lectura dels sensors, al forçar un 'reset' del mòdul de sensors.

Per tant, tot l'anterior, combinat amb els diferents modes de 'Idle' que te la placa LPC1769, ens permet funcionar de diferents formes, amb diferents consums totals del sistema. I així, hem fet treballar al sistema amb els modes : Sleep, DeepSleep i PowerDown. Hem descartat el darrer possible, que és DeepPowerDown, ja que al despertar, efectua un 'reset' de la placa LPC1769 i conseqüentment de tot el sistema.

Finalment, fem una valoració de tot el que ens aporta el mòdul de sensors, comparant-lo amb uns quants sensors individuals. D'entrada, cada sensor, individualment te un sistema de comunicació particular, si bé, sol ser I2C o CAN, caldria desenvolupar un driver per cada un. Per altra banda, tenim certs límits en el nombre de connexions d'aquest tipus, per el que ens limitaria el nombre de sensors possibles al sistema. També, ens augmentaria la complexitat de gestió i sincronització de les lectures, i hauríem de fer els càlculs pertinents per a la mètrica corresponent. A més, a pesar de ser més econòmics, tindríem que implementar els circuits electrònics necessaris, per adaptar-los al funcionament de la placa LPC1769. Tot això, ens fa valorar molt positivament l'elecció presa d'emprar el mòdul de sensors "USB Weather V3".

L'única pega del mòdul de sensors, es que te preu elevat, en comparació dels components individuals dels sensors emprats.

6. Valoració econòmica.

En aquest apartat, farem una valoració econòmica de tots els costos associats al desenvolupament d'aquest projecte. Tindrem en compte tot el material emprat, la mà d'obra de desenvolupament i posarem el cost de lloguer d'algun maquinari i programari. Això últim, es degut a que no podem imputar al projecte el cost total d'aquestes eines, i per altra banda, ens es impossible valorar la part d'import d'amortització que li correspondria en cada cas. Encara que no hem pogut fer el muntatge final del format definitiu, imputem al projecte, tot el material adquirit fer poder fer-ho.

Producte	Unitats	Preu unitari	Import productes
Placa LPC1769	1	20,00 €	20,00 €
WiFly RN-XV	1	35,00 €	35,00 €
USB Weather V3 mòdul de sensors	1	99,50 €	99,50 €
Sòcol adaptador DIP per a xBee	1	3,40 €	3,40 €
Placa Solar i Bateria	1	65,00 €	65,00 €
Barra contes 40 pins	2	2,05 €	4,10 €
Protoboard groga de 150x50	2	5,86 €	11,72 €
Joc de cables per a protoboard	2	3,25 €	6,50 €
Joc de cables rígids colors	1	9,20 €	9,20 €
Interruptor ON/OFF	1	0,95 €	0,95 €
Connector alimentació amb cable 2,54 mm	1	1,45 €	1,45 €
Joc de tornillos per plaques electròniques	1	2,69 €	2,69 €
Base alimentació placa	1	0,32 €	0,32 €
Cable alimentació base - USB	1	0,84 €	0,84 €
Peus de plastic subjecció placa electrònica	1	1,12 €	1,12 €
Placa de topos 2,54 mm	1	5,16 €	5,16 €
Caixa estanca per circuits electrònics	1	11,75 €	11,75 €
Flexi light tube conductor de llum	1	13,25 €	13,25 €
Tramesa de material	4	7,32 €	29,28 €
Hores desenvolupador al projecte FASE 2	120	30,00 €	3.600,00 €
Lloguer Ordinador estàndard	30	3,50 €	105,00 €
Lloguer eines desenvolupament VS2008	30	2,25 €	67,50 €
Lloguer instrumental : multimetre, soldador ...	30	0,55 €	16,50 €
		SUMA	4.110,23 €
		Import I.V.A.	863,15 €
		TOTAL	4.973,38 €

Figura 35 : Taula de costos associats al projecte

Per al preu / hora del desenvolupador, s'ha realitzat una estimació, en base al preu mig que es troba en el mercat actualment. Per altra banda, s'ha prorratejat els costos de tramesa, per poder tenir-los en cost – unitat. Per altra banda, cal fer notar, que alguns preus de productes electrònics, dels proveïdors emprats, podrien sortir més econòmics, en alguns altres.

7. Conclusions.

En aquest apartat, un cop ja hem finalitzat tots els punt anteriors, fem una valoració dels aspectes interessants, que hem anat trobant en el decurs del projecte. També, fem una proposta de possibles millores que es podrien realitzar en el projecte, per tal d'assolir un nivell superior. I finalment, fem una reflexió sobre el projecte, des del punt de vista d'autoavaluació, de la feina realitzada i de tot plegat.

7.1. Exposició de les conclusions.

Ara que ja hem finalitzat el projecte, podem concloure que s'ha seguit una planificació molt adequada als objectius del projecte. Encara que hem assolit tots els objectius de forma satisfactòria, la planificació temporal ha estat molt justa en quant a dates d'entrega.

A pesar de que no hem pogut fer el muntatge final de l'estació meteorològica, tal com teníem previst, el prototip fet amb una protoboard, funciona perfectament, i ens ha permès aconseguir tots els objectius del projecte. D'aquesta forma, hem aconseguit tots els objectius de la planificació inicial, i gairebé tots, els objectius de la nova planificació ampliada. Hem observat, que la nova fita que ens hem imposat, a pesar de dedicar-hi molt temps i esforç, no la podríem assolir per falta de més temps, i de possibilitats tècniques, que no ens oferien el material emprat.

Per altra banda, hem après molt sobre el món dels sistemes encastats, i de com funcionen les solucions que s'ofereixen en diferents dispositius, com pot ser el WiFly, el mòdul de sensors i la nostra pròpia aplicació. Això, ens ha obert els ulls, sobre com treballen els sistemes encastats, que cada dia tenen més aplicacions i rellevància en el món actual.

També, hem après a trobar informació sobre sistemes encastats en fòrums especialitzats en aquest tema, ja que pels cercadors d'Internet és complicat de trobar alguna cosa aprofitable. I hem vist que hi ha molta bona col·laboració entre els diferents participants d'aquests fòrums, pel que el resultat ha estat molt satisfactori i profitós.

Finalment, tot el desenvolupament del projecte, ens ha fet esforçar, en trobar les solucions més optimes per cada repte que es presentava. D'aquesta forma, hem anat aprenent cada dia una mica més sobre el funcionament i característiques particulars del material emprat, i de com interactua amb la programació que hem desenvolupat. Així, al tancament del projecte, hem pogut realitzar una aplicació de sistemes encastats, que engloba gairebé tota la feina realitzada i coneixements adquirits en el decurs del projecte.

7.2. Proposta de millores.

Un dels punts que es podria millorar, degut a la limitació de connexió remota a un sol mode de treball 'Idle' que té el sistema desenvolupat, seria buscar la forma de poder generar una interrupció externa 'EINT0'. Això permetria a l'aplicació, poder-la controlar i actuar en conseqüència, per tal d'admetre connexions remotes de configuració.

Referent al mateix tema, també caldria controlar el canvi de segons quines opcions al menú del mòdul de sensors, per tal d'evitar, que aquest es quedi no operatiu. Això, es degut a que tenim fixats uns paràmetres de treball, que si es modifiquen, no estan controlats per la aplicació que hem desenvolupat. De fet, bloquejant les opcions, es resol el problema, però es millor controlar-ho.

Per altra banda, hem desenvolupat un driver 'wdt' de Watchdog, que no hem pogut aplicar, ja que ens ha faltat afinar el control temporal que porta. Aquesta, també es una altra millora que es podria afegir a l'aplicació desenvolupada, per tal d'oferir més seguretat davant falles.

I per últim, es podria afegir els sensors de vent i pluja, al mòdul de sensors, ja que l'opció que hem escollit es el model bàsic, i així, es podria ampliar la quantitat de dades preses. En aquest cas, no cal desenvolupar res més, ja que el mòdul de sensors ho te controlat, i la nostra aplicació, incloent el servidor Web, també.

7.3. Autoavaluació.

En general, la feina realitzada, ha estat força satisfactòria donat el ajustat marge de temps de que disposàvem. De totes formes, ens queda la sensació de que hi ha alguns aspectes de sistemes encastats en que valdria la pena aprofundir una mica més, però, donades les limitacions, ha estat un mètode adequat. Ens ha generat el propòsit d'aplicar les millores al que tenim desenvolupat, i de finalitzar el muntatge final del sistema.

A més a més, el projecte ens ha fet canviar la visió que teníem dels sistemes encastats, i també, ens ha fet treballar de forma constant, per assolir unes fites. En conjunt, hem après unes tècniques d'organització i gestió del temps, hem repassat moltes coses apreses i hem après conceptes i idees noves.

8. Glossari.

3G : De l'anglès "3 Generation", referència telefonia mòbil de tercera generació, bona cobertura.

µC : Acrònim de la paraula Microcontrolador.

Ad-hoc : Mode de connexió sense fils descentralitzat, senzill, on tots els dispositius estan en igualtat de condicions.

AP : De l'anglès "Access Point", punt d'accés, fa referència a un router Wi-Fi.

ARM-Cortex : Unitat central de procés (CPU) produïda per ARM : <http://www.arm.com>

ASP.NET : Tecnologia de Microsoft per processar pàgines web, emprada en els seus servidors i eines de desenvolupament.

Bluetooth : Tecnologia connexió sense fils, de curt abast, desenvolupada inicialment per NOKIA.

BSD : Variant de llicències de programari lliure, que es pot trobar a la següent URL : <http://tssh2.sourceforge.jp/manual/en/about/copyright.html>

CMSIS : De l'anglès "Microcontroller Software Interface Standard", interfície estàndard de programari de microcontroladors.

DB : De l'anglès "Data Base", Base de Dades.

Debug/Log : Procediment que permet analitzar, depurar i registrar els passos d'una aplicació informàtica.

Firmware : Mot anglès, aplicació que treballa a baix nivell i sol ser actualitzada per fabricant.

FreeRTOS : De l'anglès "Free Real Time Operating System", Sistema operatiu de temps real lliure.

GPIO : De l'anglès "General Purpose Input / Output", fa referència a pins d'entrada sortida de propòsit general.

GPL : De l'anglès "General Public License" de GNU (GNU is Not Unix), Llicència General Pública de GNU.

GPS : De l'anglès "Global Position System", sistema de posicionament global basat en satèl·lits.

HTTP : De l'anglès "Hypertext Transfer Protocol", protocol de transferència de hipertext, estàndard d'Internet.

I/O : De l'anglès "Input / Output", fa referència a ports d'entrada – sortida.

IDE : De l'anglès "Integrated Development Environment", Entorn de desenvolupament integrat.

Idle : Mode en que s'atura el processador i per tant no consumeix, en hi ha de diferents nivells.

IIS : De l'anglès "Internet Information Services", Serveis d'Internet Information Server.

IRQ : De l'anglès "Interrupt Request", petició de interrupció, pot ser de maquinari o de programari.

JTAG : De l'anglès "Join Test Action Group", norma IEEE 1149.1, per testejar circuits electrònics.

Microcontrolador LPC1769 : Dispositiu fabricat per "NXP Semiconductors N.V." per treballar amb sistemes encastats.

NXP : NXP Semiconductors N.V., companyia fundada per PHILIPS, a la URL : <http://www.nxp.com>

O.S. : De l'anglès "Operating System", Sistema Operatiu.

Open source : De l'anglès, codi obert, significa que es distribueix tant l'executable com el codi original.

PDA : De l'anglès "Personal Digital Assitant", agenda electrònica de butxaca.

Protoboard : Mot anglès, placa per muntar prototips, també coneguda com "Breadboard".

Reset : Procediment físic o per programa, mitjançant el qual es fa reiniciar el sistema.

SoC : De l'anglès "System-on-Chip", fa referència a posar tot un sistema sencer en un sol chip.

Sparkfun : Companyia U.S.A. que subministra material electrònic i de sistemes encastats.

TCP/IP : De l'anglès "Transport Control Protocol / Internet Protocol", protocol de comunicacions de la torre OSI.

Telnet : Protocol de comunicacions TCP/IP, molt senzill, similar a una sessió de Terminal.

TFC : Acrònim de Treball Final de Carrera.

UART : De l'anglès "Universal Asynchronous Receiver-Transmitter", conegut per Port COM.

UWB : De l'anglès "Ultra-Wide-Band", banda ultra-ampla, nou tipus connexió sense fils i curt abast.

Watchdog : De l'anglès, literalment significa gos guardià, es sol emprar per controlar funcionament d'aplicacions.

Wi-Fi : De l'anglès, es una marca, i referència mecanisme connexió sense fils de dispositius.

WiFiFly : Placa electrònica amb CPU, Wi-Fi, port de comunicació i pins de funcions.

ZigBee : Altre tipus de connexió sense fils, de curt abast, baix consum i xarxa en malla

9. Bibliografia.

- [1] **Wiki Sistemes Encastats**, (15/05/2013). Dedicada a la introducció del TFC en de sistemes encastats. url : <http://cv.uoc.edu/app/mediawiki14/wiki/IniciCortexM3>
- [2] **FreeRTOS**, (19/05/2013). Web oficial del sistema operatiu FreeRTOS, conté exemples i manuals on-line. url : <http://www.freertos.org/>
- [3] **NXP**, (10/05/2013). Web oficial del fabricant NXP, conté material divers, IDE, exemples, documentació i altra informació. url : <http://www.nxp.com/>
- [4] **Fòrum NXP**, (01/06/2013). Fòrum del fabricant NXP pels seus productes, amb exemples de codi, FAQ, etc. url : <http://knowledgebase.nxp.com/>
- [5] **WiFLy**, (03/05/2013). Web oficial del fabricant Roving Networks, amb els detalls del adaptador Wi-Fi emprat. url : <http://www.rovingnetworks.com/products/RN171XV>
- [6] **Adaptador USB – sèrie CP2102**, (18/04/2013). Web oficial del fabricant Silabs, conté útils i informació. url : <http://www.silabs.com/products/mcu/Pages/USBtoUARTBridgeVCPDrivers.aspx>
- [7] **Sparkfun**, (18/05/2013). Web oficial del distribuïdor i fabricant Sparkfun, amb venda on-line, informació productes, i fòrum. url : <https://www.sparkfun.com/products/10586>
- [8] **Code Red**, (16/05/2013). Web oficial del fabricant de programari Code Red, conté informació i FAQs molt útils. url : <http://www.support.code-red-tech.com/CodeRedWiki>
- [9] **Wiki tutorial FreeRTOS**, (20/05/2013). Wiki dedicada al sistema operatiu FreeRTOS, ofereix codi d'exmple. url : http://www.socialledge.com/sjsu/index.php?title=FreeRTOS_Tutorial
- [10] **Components electrònics**, (01/05/2013). Web de venda on-line de Diotronic, ofereix una ampli catèleg de productes. url : <http://www.diotronic.com/>
- [11] **Components de Sistemes encastats**, (02/05/2013). Web de venda on-line de BricoGeek, poca varietat de productes. url : <http://www.bricogeek.com/shop/>
- [12] **Components de Sistemes encastats**, (02/05/2013). Web de venda on-line de Watterott amb molts productes. url : <http://www.watterott.com/>
- [13] **Components de Sistemes encastats**, (02/05/2013). Web de venda on-line de cooking-hacks amb molts productes i poc stock. url : <http://www.cooking-hacks.com/index.php/>
- [14] **Centre meteorològic**, (25/05/2013). Web oficial de Wunderground, recollida de dades i informació Apps meteorològiques. url : <http://www.wunderground.com/weatherstation/setup.asp>

10. Annexos.

10.1. Manual d'usuari.

Per posar en marxa l'estació meteorològica, el primer que s'ha de fer, es instal·lar el servidor de pàgines Web "UltiDev Cassini 2.0" de la URL : <http://ultidev.com/products/cassini/> . Posteriorment, descomprimir el fitxer ZIP "WUHU.zip", a la unitat C:\, de forma que surti la carpeta C:\WUHU.

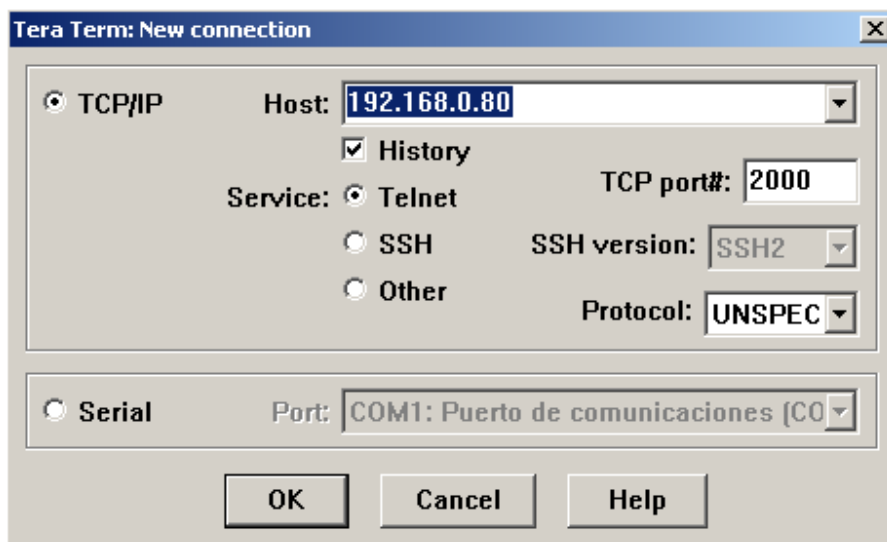
El pas següent, es tracta de configurar el servei web amb el programa instal·lat "Cassini Webserver Explorer". Un cop iniciat, cal posar una configuració con la de la imatge adjunta :

Un cop es guarda la configuració, el servei ja està en marxa i podem comprova l'accés, posant al navegador la IP del nostre PC, per el port 8080. Si tot funciona correctament, ens ha de sortir una pantalla com la següent :

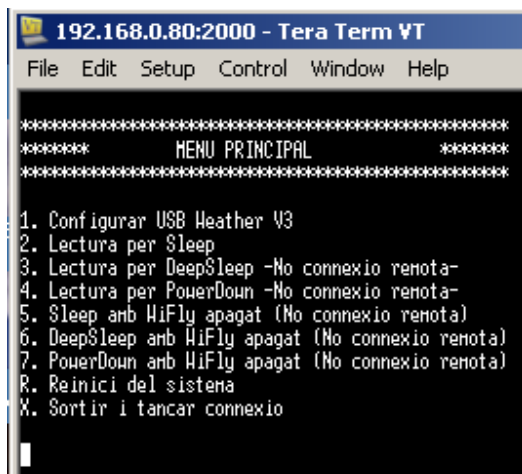
Posteriorment, ja podem donar corrent al sistema encastat, per que engegui l'aplicació 'Satation' que hi tenim carregada. I aquesta, va enviant dades al servidor Web, que les guarda a la DB. Per consultar la informació rebuda, només cal obrir la base de dades Ms Access de la carpeta del Web, que és : C:\WUHU\BBDD1.mdb i al obrir la taula que hi ha, veurem una informació com :

Id	Data	Temperature	Humidity	Dewpoint	Pressure	Light	Windspeed	Wir
27	06/05/2013 13:14:38	2,2	2,2	2,2	2,2	2,2	0	
28	08/05/2013 16:20:46	2,2	2,2	2,2	2,2	2,2	0	
29	08/05/2013 16:21:00	2,2	2,2	2,2	2,2	2,2	0	
30	08/05/2013 16:32:09	2,2	2,2	2,2	2,2	2,2	0	
31	17/05/2013 13:35:57	2,2	2,2	2,2	2,2	2,2	0	
32	17/05/2013 13:39:50	2,2	2,2	2,2	2,2	2,2	0	
33	17/05/2013 13:40:28	2,2	2,2	2,2	2,2	2,2	0	
34	29/05/2013 10:27:36	2,2	2,2	2,2	2,2	2,2	0	
35	29/05/2013 10:31:54	2,2	2,2	2,2	2,2	2,2	0	
36	29/05/2013 10:50:43	2,2	2,2	2,2	2,2	2,2	0	
37	29/05/2013 10:55:54	2,2	2,2	2,2	2,2	2,2	0	
38	29/05/2013 15:54:21	2,2	2,2	2,2	2,2	2,2	0	
39	29/05/2013 15:54:58	1,1	1,1	1,1	1,1	1,1	0	
40	29/05/2013 16:00:51	1,1	1,1	1,1	1,1	1,1	0	
41	08/06/2013 0:19:00	2,2	2,2	2,2	2,2	2,2	0	
*(Nuevo)	09/06/2013 23:43:14							

Si volem modificar la configuració de funcionament de l'estació meteorològica, només tenim que obrir una sessió de connexió TCP/IP, amb el programa 'TeraTerm', per exemple, les dades a posar, en el nostre cas, serien :




I un cop ens connectem de forma remota, al WiFly, aquest desperta a la aplicació de sistemes encastats 'Station', i ens mostra una pantalla con la següent :



```
192.168.0.80:2000 - Tera Term VT
File Edit Setup Control Window Help
*****
***** MENU PRINCIPAL *****
*****
1. Configurar USB Weather V3
2. Lectura per Sleep
3. Lectura per DeepSleep -No connexio remota-
4. Lectura per PowerDown -No connexio remota-
5. Sleep amb HiFly apagat (No connexio remota)
6. DeepSleep amb HiFly apagat (No connexio remota)
7. PowerDown amb HiFly apagat (No connexio remota)
R. Reinici del sistema
X. Sortir i tancar connexio
```

Si volem modificar la configuració de temps de les lectures, hem d'escollir la primera opció de la llista anterior. A les hores, ens surt un altre menú com el següent :



```
192.168.0.80:2000 - Tera Term VT
*****
* HELLO ... INICI SESSIO REMOTA HEATHER V3 *
*****
SparkFun USB Weather Board V3 firmware version 1.1
free RAM: 181 bytes
1. Data format (CSV)
2. General units (SI)
3. Sample rate (10)
4. Pressure units (mbar)
5. Pressure type (absolute)
6. Station altitude (695 meters)
7. Baud rate (9600 baud)
8. Zero rain counter
9. Weather Meters attached (yes)
X. Exit (don't save changes to EEPROM)
S. Save (save changes to EEPROM)
```

Hem de tenir clar, que en cada pantalla, després de posar el número d'opció, sempre hem de pulsar entrada, per tal d'enviar les dades. Per altra banda, aquí només tenim activa l'opció de canviar 'Sample rate', es a dir temps entre lectures.

Finalment, cal sortir dels menús, per que comencin a funcionar les opcions modificades, i en el cas de canviar el mode de 'Idle', si es vol fer un canvi posterior, caldrà reiniciar manualment el sistema encastat.

10.2. Execució, compilació i càrrega.

Això no representa cap problema si es té instal·lat el IDE LPCXpresso al sistema. Com que es similar al funcionament del IDE Eclipse, ja que es tracta d'una variant, emprà el mateix procediment. Per altra banda, te incorporades funcionalitats específiques per treballar amb l'entorn de sistemes encastats de les plaques LPCxxxx del fabricant NXP.

Només cal llegir 5 minuts el manual de funcionament del LPCXpresso "Starting Guide", per veure com funciona tot. Com que la nostra aplicació no te cap 'bug' de compilació, no ha de donar cap problema, seguint els passos indicats el l'anterior manual.