

# **Sistema d'alarma i monitorització ambiental amb gestió remota per a entorns crítics**

**Enginyeria Tècnica de Telecomunicació, especialitat Telemàtica**

**Estudiant**

**Sergio Gómez Espín**

**Consultor**

**Jordi Bécares Ferrés**

14 de gener de 2014

## Resum

Aquest treball descriu el disseny i desenvolupament d'un sistema d'alarma i monitorització ambiental amb gestió remota per a entorns crítics o d'alta disponibilitat, format per un dispositiu integrat de control amb connexió Wi-Fi i una aplicació servidor amb interfície web.

El dispositiu integrat de control està basat en una placa LPCXPRESSO LPC1769, a la qual se li ha incorporat un sensor digital de temperatura TMP102 de Texas Instruments, un bronzidor amb oscil·lador incorporat per fer funcions d'alarma sonora, un convertidor UART-USB CP2102 que permet visualitzar missatges de *log* a través de consola i un mòdul WiFly RN-XV que dota al sistema de connexió a Internet a través de Wi-Fi. Per al desenvolupament de l'aplicació s'han utilitzat les eines proporcionades pel sistema operatiu en temps real FreeRTOS.

El dispositiu obté periòdicament les mesures de temperatura i les processa d'acord amb els paràmetres de configuració prèviament establerts per l'usuari. El sistema d'alarma consisteix en l'enviament, per part del dispositiu, d'un correu electrònic d'alerta que s'envia de manera autònoma quan les temperatures obtingudes estan fora dels marges de seguretat, utilitzant el servidor SMTP que l'usuari hagi configurat. El dispositiu també disposa d'una alarma sonora que es pot habilitar per configuració.

El sistema es complementa amb una aplicació servidor desenvolupada en PHP, que rep les dades de temperatura del dispositiu i li proporciona els paràmetres de configuració del sistema, tot a través d'una comunicació via *socket* TCP entre els dos extrems. A través d'una interfície web, l'usuari pot configurar el dispositiu de manera remota i visualitzar les dades de temperatura en temps real.

**Paraules clau:** sistemes encastats, WSN, LPC1769, WiFly RN-XV, TMP102, I2C, CP2102, FreeRTOS, Socket TCP, SMTP.

Aquest treball de final de carrera està emmarcat dins de l'àrea de Sistemes Encastats dels estudis d'Enginyeria Tècnica de Telecomunicació, especialitat Telemàtica, de la Universitat Oberta de Catalunya.

# Índex de continguts

<b>1. Introducció</b> .....	7
1.1. Justificació .....	7
1.2. Descripció .....	8
1.3. Objectius.....	8
1.4. Enfocament i mètode seguit.....	9
1.5. Planificació .....	9
1.6. Recursos emprats .....	11
1.6.1. Recursos de maquinari .....	11
1.6.2. Recursos de programari.....	13
1.7. Productes obtinguts.....	14
1.8. Descripció dels altres capítols de la memòria .....	15
<b>2. Antecedents</b> .....	16
2.1. Estat de l'art.....	16
2.1.1. Plataformes <i>hardware</i> .....	16
2.1.2. Sensors de temperatura .....	18
2.1.3. Connectivitat sense fils.....	19
2.2. Estudi de mercat .....	20
<b>3. Descripció funcional</b> .....	21
3.1. Sistema d'alarma i monitorització ambiental amb gestió remota .....	21
3.1.1. Diagrama de blocs .....	21
3.1.2. Comunicació entre els diferents elements del sistema.....	23
3.2. Disseny de l'aplicació desenvolupada al dispositiu integrat de control .....	24
3.2.1. Diagrama de blocs .....	24
3.3. Disseny de l'aplicació servidor i interfície d'usuari .....	25
3.3.1. Diagrama de blocs .....	25
3.3.2. Servidor i base de dades.....	26
3.3.3. Interfície d'usuari .....	26
<b>4. Descripció detallada</b> .....	29
4.1. Aplicació desenvolupada al dispositiu integrat de control.....	29
4.1.1. Diagrama de blocs .....	29
4.1.2. Llibreria de controladors .....	30
4.1.3. Aplicació i tasques.....	32
4.2. Detall de les implementacions.....	34
4.2.1. Configuració i gestió remota del sistema .....	34
4.2.2. Obtenció i enviament de mesures de temperatura .....	35

4.2.3. Sistema d'alarma.....	36
4.2.4. Gestió de pèrdua de connexió .....	39
4.2.5. Sistema de <i>log</i> .....	40
4.3. Detall de la comunicació entre els elements del sistema .....	42
4.3.1. Comunicació amb el servidor. <i>Socket TCP</i> .....	42
4.3.2. Alertes per correu electrònic. Protocol SMTP .....	45
4.4. Aplicació servidor i interfície d'usuari .....	46
4.4.1. Diagrama de blocs .....	46
4.4.2. Servidor TCP .....	46
4.4.3. Base de dades .....	48
4.4.4. Interfície web .....	49
4.5. Esquema de connexions .....	50
<b>5. Viabilitat tècnica .....</b>	<b>51</b>
<b>6. Valoració econòmica .....</b>	<b>52</b>
<b>7. Conclusions .....</b>	<b>53</b>
7.1. Conclusions del projecte .....	53
7.2. Proposta de millores.....	53
7.3. Autoavaluació .....	54
<b>8. Glossari .....</b>	<b>55</b>
<b>9. Bibliografia.....</b>	<b>57</b>
<b>10. Annexos .....</b>	<b>58</b>
10.1. Execució, compilació i càrrega de l'aplicació encastada .....	58
10.2. Instal·lació i configuració de l'aplicació servidor .....	59

## Índex de figures

Figura 1: Planificació inicial del projecte .....	10
Figura 2: Planificació final del projecte.....	10
Figura 3: Placa LPCXpresso LPC1769 .....	11
Figura 4: Mòdul WiFly RN-XV de Roving Networks.....	11
Figura 5: Convertidor UART-USB CP2102 .....	12
Figura 6: Sensor TMP102 de Texas Instruments .....	12
Figura 7: Brunzidor Velleman SV3 .....	12
Figura 8: Transformador 220VAC-USB .....	13
Figura 9: Cable Barrel-USB.....	13
Figura 10: Connector Barrel .....	13
Figura 11: Material electrònic divers .....	13
Figura 12: Dispositiu integrat de control amb connexió Wi-Fi.....	14
Figura 13: Captures de pantalla de la interfície web d'usuari .....	15
Figura 14: Comparativa de plataformes hardware.....	17
Figura 15: Comparativa de sensors de temperatura.....	18
Figura 16: Comparativa de tecnologies de comunicació sense fils .....	19
Figura 17: Comparativa de solucions d'alarma i monitorització.....	20
Figura 18: HWg-STE. Termòmetre Ethernet amb alertes per e-mail.....	20
Figura 19: HWg-Ares12. Termòmetre GSM/GPRS amb alertes per e-mail i SMS.....	20
Figura 20: Diagrama de blocs de la solució proposada .....	21
Figura 21: Diagrama de blocs simplificat de l'aplicació desenvolupada a l'LPC .....	24
Figura 22: Diagrama de blocs simplificat de l'aplicació servidor i interfície d'usuari.....	25
Figura 23: Captura de pantalla del web de visualització de temperatures .....	27
Figura 24: Captura de pantalla del web de configuració.....	27
Figura 25: Paràmetres configurables per l'usuari des de l'entorn web .....	28
Figura 26: Diagrama de blocs detallat de l'aplicació desenvolupada a l'LPC.....	30
Figura 27: Diagrama de flux de la inicialització del dispositiu integrat de control .....	33
Figura 28: Diagrama de flux de la funcionalitat de configuració i gestió remota del sistema .....	35
Figura 29: Diagrama de flux de l'obtenció i enviament de les dades de temperatura .....	36
Figura 30: Diagrama de flux de la funcionalitat d'alarma .....	38
Figura 31: Correu electrònic d'alarma rebut des d'un dispositiu mòbil .....	39
Figura 32: Correu electrònic d'informació rebut des d'un dispositiu mòbil.....	39
Figura 33: Diagrama de flux de la funcionalitat de gestió de la pèrdua de connexió .....	40
Figura 34: Missatges de <i>log</i> en temps real accessibles des d'una aplicació terminal .....	41
Figura 35: Detall de segments TCP en la comunicació entre l'LPC i el servidor .....	43
Figura 36: Captura d'un segment TCP enviat a través del socket.....	44
Figura 37: Detall de la comunicació amb el servidor SMTP .....	45
Figura 38: Diagrama de blocs detallat de l'aplicació servidor i interfície d'usuari .....	46

Figura 39: Comandes que accepta el servidor TCP .....	47
Figura 40: Dades rebudes pel servidor a través del <i>socket</i> TCP.....	48
Figura 41: Estructura de les taules de la base de dades .....	48
Figura 42: Contingut de la taula <i>temperatures</i> .....	49
Figura 43: Contingut de la taula <i>config</i> .....	49
Figura 44: Esquema de connexions del dispositiu integrat de control .....	50

# 1. Introducció

En aquest treball es descriu el disseny i desenvolupament d'un sistema d'alarma i monitorització ambiental amb gestió remota orientat per al seu ús en entorns crítics o d'alta disponibilitat, tals com sales de servidors.

En aquest capítol, es descriuen els motius que han justificat l'elecció d'aquest treball, es realitza una breu descripció tècnica i funcional del sistema dissenyat, s'exposen els objectius, s'analitza la planificació seguida per l'assoliment de les fites proposades i es relacionen els recursos de maquinari i programari emprats en el desenvolupament de la solució.

## 1.1. Justificació

Aquest projecte sorgeix com a proposta de solució davant la necessitat actual d'un entorn d'alta disponibilitat. A continuació s'exposen els detalls.

La meua empresa disposa d'una sala de servidors que concentra tots els seus elements centrals de comunicació (servidors, unitats de *backup*, centraleta telefònica, concentradors, encaminadors...), refrigerada per un únic aparell d'aire condicionat, sense redundància, que manté la sala a una temperatura de 22°C.

Existeixen precedents recents d'avaría de l'aparell de refrigeració, que han ocasionat que la temperatura de la sala s'elevi de forma molt considerable, posant en perill els dispositius electrònics i el funcionament dels sistemes de comunicació de l'oficina. En aquests casos, l'avaría no va ser detectada fins dies després, quan el responsable de sistemes va accedir a la sala de servidors per realitzar altres tasques, ja que no es disposa de cap sistema d'avís en cas de fallada del sistema de refrigeració.

L'objectiu d'aquest projecte és donar solució a aquesta problemàtica, dotant d'un sistema d'alarma i monitorització ambiental a la sala de servidors, que permeti al responsable de sistemes ser alertat per correu electrònic quan els paràmetres de temperatura estiguin fora dels nivells de seguretat establerts, i visualitzar les dades de temperatura en temps real, de manera senzilla, a través d'un entorn web.

D'aquesta manera, el responsable dels sistemes de comunicació de l'empresa, podrà controlar les temperatures i ser alertat en cas de fallada del sistema de refrigeració, encara que no estigui físicament a l'oficina.

Tot i que el sistema a desenvolupar es centrarà en donar solució a aquest cas concret, es dissenyarà de tal manera que els productes i el sistema obtinguts puguin funcionar en qualsevol entorn crític o d'alta disponibilitat de característiques similars.

## 1.2. Descripció

Per tal de donar solució al problema plantejat, s'ha dissenyat un **dispositiu integrat de control amb connexió Wi-Fi** basat en una placa LPCXpresso LPC1769, a la qual se li ha incorporat un sensor de temperatura, una alarma sonora, un mòdul CP2102 que permet visualitzar missatges de *log* a través de consola i un mòdul WiFly RN-XV que dota al sistema de connexió a Internet.

Aquest dispositiu, que està ubicat físicament a l'espai que es vol monitoritzar, és l'encarregat d'obtenir periòdicament les mesures de temperatura i processar-les, comprovant que estan dins dels límits de seguretat prèviament definits per l'usuari. En cas que les temperatures obtingudes estiguin fora dels marges establerts, el dispositiu envia de manera autònoma un correu electrònic d'alerta a l'usuari i activa una alarma sonora, en cas que estigui habilitada per configuració. Un cop normalitzada la situació, s'informa a l'usuari també a través de correu electrònic i es desactiva l'alarma sonora.

El sistema es complementa amb una **aplicació servidor** que rep les dades de temperatura del dispositiu de control i li proporciona els paràmetres de configuració del sistema, tot a través d'una comunicació via *socket* TCP. A través d'una **interfície web**, l'usuari pot configurar el dispositiu de manera remota i visualitzar les dades de temperatura en temps real.

## 1.3. Objectius

Els objectius del sistema a desenvolupar són els següents:

- **Obtenció de dades de temperatura**  
El sistema ha d'obtenir, de manera periòdica, les dades de temperatura de l'espai on està instal·lat a través del sensor implementat a l'LPC.
- **Enviament de les dades mesurades a un servidor d'Internet**  
Les dades obtingudes pel sensor s'han d'enviar, en temps real, a un servidor d'Internet, a través d'una comunicació via *socket* TCP.
- **Enviament d'alertes via correu electrònic**  
El sistema ha d'enviar, de manera autònoma, una alerta per correu electrònic quan les mesures obtingudes a través del sensor estiguin fora dels nivells de seguretat prèviament establerts per l'usuari.



Adicionalment, també ha d'enviar un correu d'informació quan les mesures tornin a estar dins dels marges de seguretat.

- **Alarma sonora**

L'usuari ha de poder activar una funcionalitat mitjançant la qual el sistema emeti una alarma sonora quan les mesures obtingudes pel sensor estiguin fora dels nivells de seguretat.

- **Consulta de les dades mesurades via web**

L'usuari ha de poder visualitzar les dades de temperatura obtingudes pel sensor de manera senzilla, a través d'un entorn web.

- **Configuració del sistema via web**

L'usuari ha de poder configurar, en qualsevol moment, els paràmetres de configuració del sistema (temperatura de seguretat màxima i mínima, freqüència d'obtenció de mesures, dades del servidor SMTP i estat de l'alarma sonora) de manera senzilla, a través d'un entorn web.

#### **1.4. Enfocament i mètode seguit**

Per tal d'afrontar amb èxit del projecte, en primer lloc, ha estat fonamental realitzar una fase d'aprenentatge per tal d'entrar en contacte amb la tecnologia a utilitzar. En aquesta fase s'han adquirit nocions bàsiques del funcionament del maquinari i del programari que s'ha fet servir durant les següents etapes del projecte.

Un cop finalitzada la fase d'aprenentatge, s'ha realitzat una planificació del projecte, on s'han definit de manera detallada les tasques necessàries a desenvolupar per tal d'assolir els objectius proposats.

Cada tasca a realitzar ha representat una fita, i no s'ha passat a la següent fins que s'ha assegurat el correcte funcionament de l'actual, mitjançant el procés de proves corresponent. Aquesta metodologia ha facilitat la detecció i correcció dels errors que han anat sorgint durant l'etapa de desenvolupament.

#### **1.5. Planificació**

A continuació es presenta la planificació temporal de les tasques, comparant la que es va realitzar a l'inici del projecte, amb la planificació que finalment s'ha seguit.

## Planificació Inicial

Cronograma de tasques que es va realitzar a l'inici del projecte:

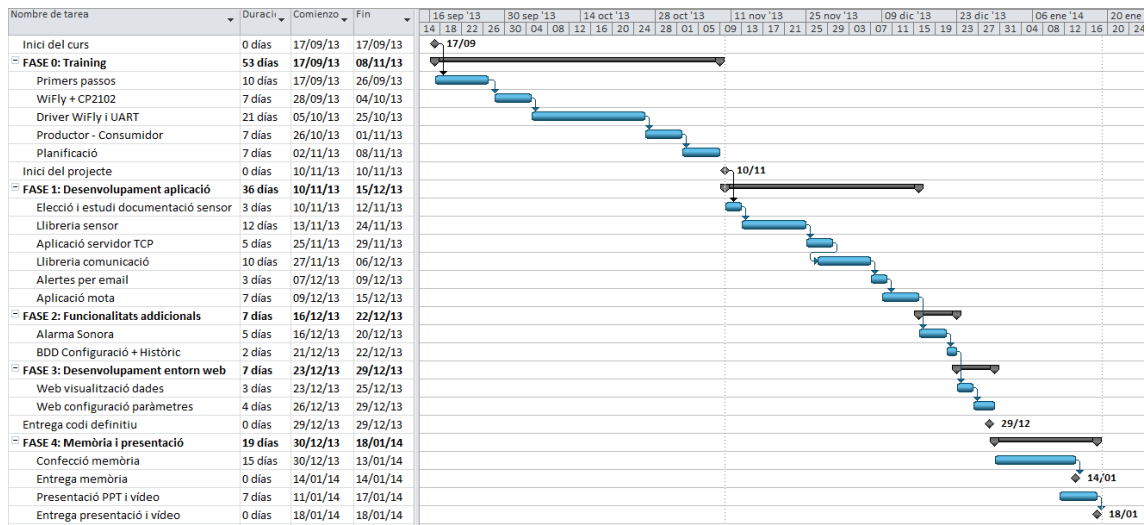


Figura 1: Planificació inicial del projecte

## Planificació final

Cronograma de tasques que finalment s'ha seguit:

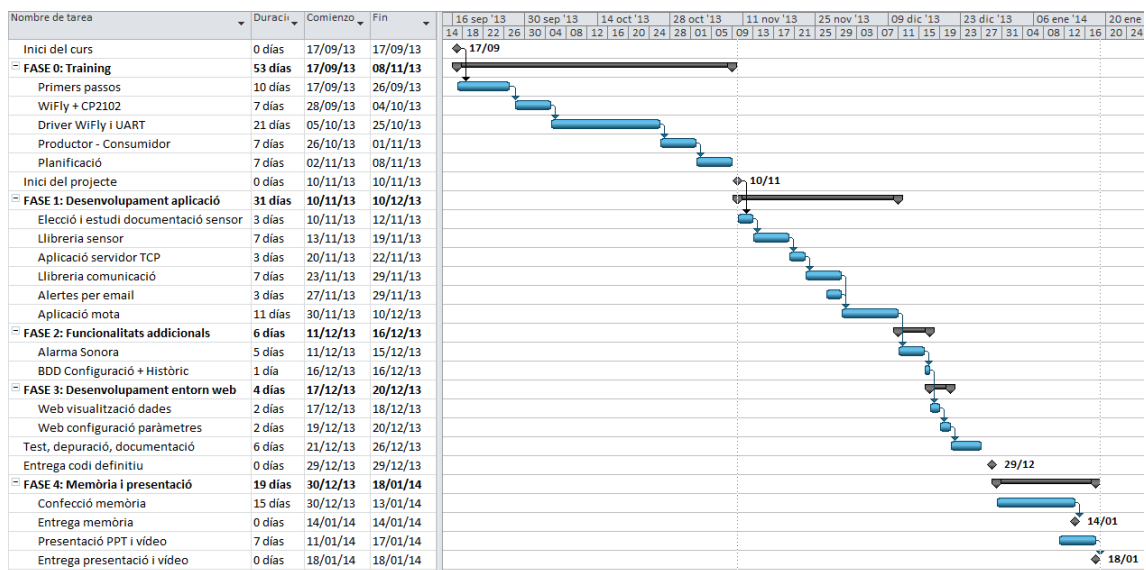


Figura 2: Planificació final del projecte

Pel que fa a la primera fase, tant el desenvolupament de les llibreries, com el de l'aplicació servidor, han requerit menys temps del previst. La diferència més gran es troba en el desenvolupament de la llibreria de comunicació amb el sensor, que va passar de 12 a 7 dies, donat que es va poder aprofitar i adaptar una llibreria d'exemple proporcionada pel fabricant NXP. Tanmateix, part del temps avançat, va ser emprat en el desenvolupament de l'aplicació a la mota, que va requerir més temps del planificat inicialment, passant de 7 a 11 dies.

En total, l'assoliment de la primera fase va requerir 5 dies menys de l'inicialment previst. Un dels motius és que, en general, durant el desenvolupament no han existit problemes greus que hagin alterat la planificació, donat que la majoria de complicacions que han anat sorgint s'han pogut detectar i solucionar ràpidament gràcies al bagatge adquirit durant la fase d'aprenentatge.

La segona fase s'ha realitzat d'acord amb la planificació i, en la tercera fase, el temps requerit també va ser menor a l'inicialment previst. Els dies que es van poder avançar, van possibilitar dedicar 6 dies a final del desenvolupament per realitzar proves, depuració i documentació del codi.

## 1.6. Recursos emprats

En aquest apartat es presenten els diferents recursos que s'han utilitzat per desenvolupar el sistema, diferenciant entre recursos de maquinari i de programari.

### 1.6.1. Recursos de maquinari

A continuació es relacionen els recursos de maquinari emprats:

- **Placa LCPXpresso LPC1769**

Placa que incorpora un microcontrolador ARM Cortex-M3 de NXP.



Figura 3: Placa LCPXpresso LPC1769

- **Mòdul WiFly RN-XV**

Mòdul Wi-Fi 802.11b/g de baix consum.



Figura 4: Mòdul WiFly RN-XV de Roving Networks

- **Convertidor CP2102**

Convertidor UART a USB i viceversa.



Figura 5: Convertidor UART-USB CP2102

- **Sensor de temperatura TMP102 de Texas Instruments**

Sensor digital de temperatura de baix consum, amb 12 bits de resolució, precisió de  $\pm 0.5^{\circ}\text{C}$  i comunicació a través d'interfície I2C/TWSI.



Figura 6: Sensor TMP102 de Texas Instruments

- **Brunzidor Velleman SV3**

Brunzidor de to continu amb oscil·lador incorporat.



Figura 7: Brunzidor Velleman SV3

- **Components per a l'alimentació del sistema**

Transformador 220VAC-5VDC amb connector USB, cable d'alimentació Barrel-USB i connector d'alimentació tipus Barrel compatible amb plaques de prototipat.



Figura 8: Transformador  
220VAC-USB



Figura 9: Cable Barrel-USB



Figura 10: Connector Barrel

#### - **Material electrònic divers**

Plaques de prototipat, adaptador DIP per a XBee, cables de prototipat M/M i M/F, pins sense soldadura, resistències, multímetre...



Figura 11: Material electrònic divers

### 1.6.2. Recursos de programari

A continuació es relacionen els recursos de programari emprats per al desenvolupament de l'aplicació al dispositiu integrat de control, i els utilitzats per al desenvolupament de l'aplicació servidor i la interfície d'usuari.

- **Per al desenvolupament de l'aplicació al dispositiu integrat de control:**
  - o Entorn de desenvolupament LPCXpresso v5.2.6
  - o Sistema operatiu en temps real FreeRTOS
  - o Llenguatge de programació C
  - o Aplicació terminal TeraTerm
- **Per al desenvolupament de l'aplicació servidor i interfície d'usuari:**
  - o Entorn de desenvolupament Aptana Studio 3 (IDE PHP)
  - o Llenguatge de programació PHP
  - o Base de dades MySQL

## 1.7. Productes obtinguts

Tal com s'ha avançat anteriorment, la solució desenvolupada està formada per dos productes:

### Dispositiu integrat de control amb connexió Wi-Fi

El dispositiu està basat en una placa LPCXpresso LPC1769, a la qual se li ha incorporat un sensor de temperatura TMP102 de Texas Instruments, un bronzidor amb oscil·lador incorporat per fer funcions d'alarma sonora, un convertidor UART-USB CP2102 que permet visualitzar missatges de *log* a través de consola i un mòdul WiFly RN-XV que dota al sistema de connexió a Internet a través de Wi-Fi.

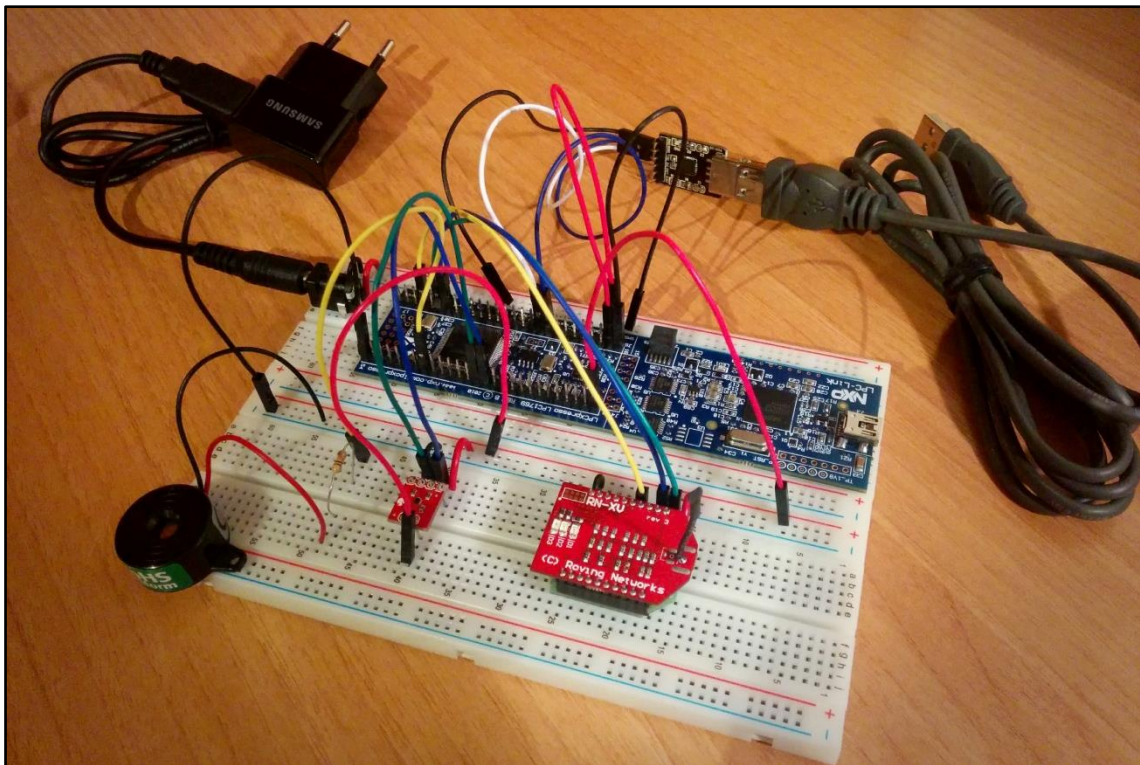


Figura 12: Dispositiu integrat de control amb connexió Wi-Fi

### Aplicació servidor i interfície d'usuari

L'aplicació està formada per un **servidor TCP** que rep les mesures del dispositiu de control i li proporciona els paràmetres de configuració del sistema, una **base de dades** on es desen les dades rebudes i els paràmetres de configuració, i una **interfície d'usuari basada en web** que permet configurar el dispositiu remotament de manera senzilla i visualitzar les mesures obtingudes en temps real.

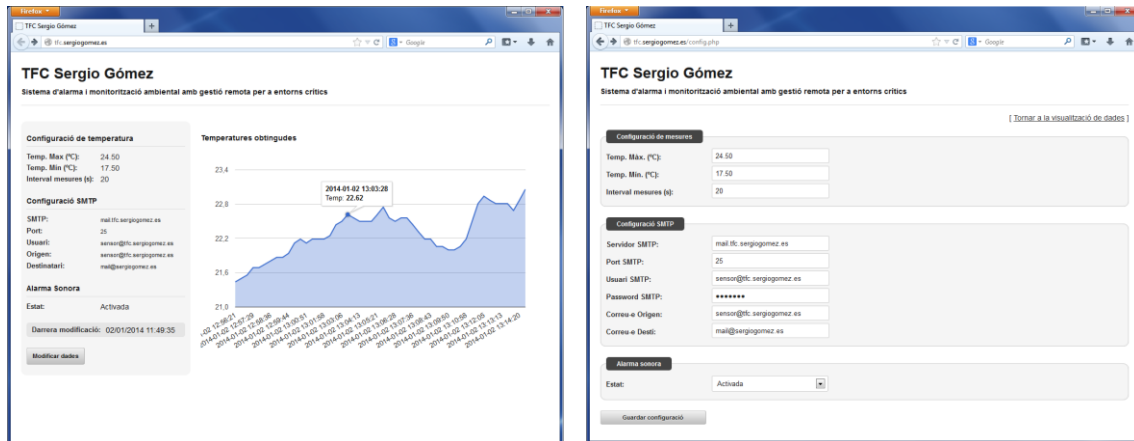


Figura 13: Captures de pantalla de la interfície web d'usuari

## 1.8. Descripció dels altres capítols de la memòria

En el següent capítol de la memòria s'analitzen i es comparen les tecnologies utilitzades en el desenvolupament de la solució dissenyada, amb altres tecnologies i dispositius disponibles al mercat. En el tercer capítol, es descriu el sistema desenvolupat des d'un punt de vista funcional per, més tard i ja en el quart capítol, explicar de manera tècnica i detallada cadascuna de les implementacions desenvolupades.

En els capítols posteriors, s'analitza la viabilitat tècnica de la solució proposada i els costos associats al desenvolupament. Finalment, es presenten les conclusions del projecte i una proposta de futures millores de la solució dissenyada.



## 2. Antecedents

Un sistema encastat és un sistema informàtic dissenyat per realitzar una funció específica de manera dedicada. A diferència d'un ordinador, l'usuari final pot configurar el dispositiu encastat, però no pot modificar la funcionalitat per la qual ha estat creat.

Una de les possibles aplicacions d'aquests sistemes és la crear xarxes de sensors sense fils, formades per dispositius encastats orientats a monitoritzar paràmetres ambientals dins d'uns marges de seguretat i enviar les dades obtingudes a través de la xarxa.

### 2.1. Estat de l'art

En aquest apartat s'analitzen diferents plataformes orientades al desenvolupament de xarxes de sensors, així com diversos sensors de temperatura i solucions de connectivitat sense fils, per tal de realitzar una breu comparativa amb la tecnologia existent actualment al mercat i la utilitzada en el desenvolupament del sistema dissenyat.

#### 2.1.1. Plataformes *hardware*

En els darrers anys ha anat creixent el nombre de plataformes *hardware* orientades al desenvolupament de xarxes de sensors sense fils. Un exemple és l'aparició del popular projecte Arduino l'any 2005, una plataforma lliure inicialment dissenyada amb propòsits educatius, però que actualment ja disposa de més de 20 tipus de plaques al mercat per cobrir un gran ventall de necessitats.

Més recentment, l'any 2010, va aparèixer Waspote, una plataforma creada a Saragossa i orientada específicament per a projectes de comunicació sense fils de molt baix consum. Un dels fets diferencials d'aquest projecte, és que el mateix fabricant ofereix els perifèrics de connectivitat per a qualsevol protocol (ZigBee, WiFi, Bluetooth, 3G, GSM/GRPS...) ja preparats per a treballar amb la placa.

A continuació es presenta una taula comparativa entre la placa LPC1769 utilitzada en aquest projecte, i les plaques Arduino Due (la primera placa Arduino amb arquitectura de 32 bits, apareguda a l'octubre de 2012) i Waspote.



	NXP LPC1769	Arduino Arduino Due	Libelium Waspote
<b>Processador</b>	ARM Cortex-M3 100MHz	ARM Cortex-M3 84MHz	ATmega1281 14MHz
<b>Flash</b>	512KB	512KB	128KB
<b>RAM</b>	64KB SRAM	96KB SRAM	8KB SRAM
<b>Consum</b>	Active <sup>1</sup> : 67mA Sleep: 2mA Deep Sleep: 240uA Power-down: 31uA	Active <sup>2</sup> : 76.47mA Sleep: 2,5uA-45.9mA Wait: 26.6uA	Active: 15mA Sleep: 55uA Deep Sleep: 55uA Hibernate: 0.06uA
<b>Interfícies</b>	4xUART 3xI2C 2xSSP 2xCAN 1xI2S  SPI, PWM, ADC, DAC, JTAG, USB Ethernet	4xUART 2xI2C 2xCAN 1xI2S  SPI, PWM, ADC, DAC, JTAG, USB Ethernet	2xUART 1xI2C  SPI, PWM, USB
<b>Preu</b>	30€	39€	90€

Figura 14: Comparativa de plataformes hardware

Tal com es pot veure a la taula, la placa LPC1769 utilitzada en el projecte i la placa Arduino Due tenen característiques molt similars, donat que ambdues incorporen el mateix tipus de processador: un Cortex-M3 d'ARM. Tot i així, la placa de NXP disposa d'un processador amb una velocitat de rellotge sensiblement superior, amb menor consum en mode actiu i a un cost més econòmic.

Tanmateix, la diferència principal entre ambdues plaques la trobem a nivell de programari. En aquest aspecte, Arduino utilitza un llenguatge de desenvolupament propi de molt alt nivell, dissenyat amb propòsits educatius i orientat a introduir als desenvolupadors en el món de la programació. Amb la placa LPC1769, en canvi, el programador ha de controlar tots els aspectes del desenvolupament mitjançant la utilització de les eines proporcionades pel sistema operatiu en temps real utilitzat.

Finalment, pel que fa la placa Waspote, les prestacions que ofereix són més limitades, fet que també fa que tingui un consum notablement inferior i que sigui una opció indicada per realitzar tasques molt concretes que no requereixin grans recursos.

<sup>1</sup> Consum amb el rellotge a 120MHz i execució de codi des de la memòria Flash.

<sup>2</sup> Consum amb el rellotge a 84MHz i execució de codi des de la memòria Flash.

## 2.1.2. Sensors de temperatura

Per tal de realitzar l'elecció del sensor de temperatura que inclou el dispositiu integrat de control, es van analitzar diferents sensors disponibles al mercat, comparant les seves prestacions i els aspectes de compatibilitat amb la placa LPC1769 utilitzada.

A continuació es mostra una taula amb els sensors analitzats:

	Analog Devices TMP36	Texas Instruments TMP102	Sensirion SHT15 <sup>3</sup>	Maxim Integrated DS18B20
<b>Sortida</b>	Analògica	Digital	Digital	Digital
<b>Comunicació</b>	ADC	I2C/TWSI	TWSI No I2C	1-Wire
<b>Precisió a 25°C</b>	±1°C	±0.5°C	±0.3°C	±0.5°C
<b>Precisió en rang típic</b>	±3°C	±0.5°C	±1°C	±0.5°C
<b>Consum màxim</b>	Actiu: 50uA Idle: 0,5uA	Actiu: 10uA Idle: 1uA	Actiu: 1mA Idle: 1,5uA	Actiu: 1,5mA Idle: 1uA
<b>Alimentació</b>	2.7V – 5.5V	1.4V – 3.6 V	2.4V – 5.5V	3.0V – 5.5V
<b>Rang</b>	-40°C a +125°C	-40°C a +125°C	-40°C a 124°C	-55°C a +125°C
<b>Preu</b>	1,90€	6,00€	33,95€	4,90€

Figura 15: Comparativa de sensors de temperatura

Tots els sensors analitzats treballen en el mateix rang de temperatures i són compatibles amb la placa LPC1769 pel que fa a l'alimentació (3.3V). En canvi, sí que es troben diferències significatives pel que fa al protocol utilitzat en la comunicació, el consum i la precisió. Els motius per optar pel model TMP102 de Texas Instruments van ser els següents:

- **Prestacions:** És el sensor que ofereix menor marge d'error (±0.5°C) en un rang més ampli de temperatures (de -25°C a 85°C), i el que té el menor consum en mode actiu i dels més baixos en mode inactiu.
- **Comunicació:** Tot i que en la taula s'analitzen tres sensors digitals, i dos d'ells amb interfície *Two-Wire Serial Interface*, el TMP102 és l'únic totalment compatible amb el protocol I2C, fet que facilita enormement el desenvolupament del controlador de comunicació amb la placa.

<sup>3</sup> El model SHT15 també obté dades d'humitat relativa.

### 2.1.3. Connectivitat sense fils

El dispositiu integrat de control requereix de connexió a Internet per tal de poder comunicar-se amb l'aplicació servidor desenvolupada i amb un servidor SMTP, a través de qual es realitzaran els enviaments dels correus electrònics.

Com s'ha avançat en la introducció, i s'entrarà en detall en capítols posteriors, tota la comunicació es realitza a través de *socket* TCP, on el volum de dades enviades i rebudes pel dispositiu és molt reduït. Per exemple, la mida total de les trames enviades i rebudes en una comunicació per a l'enviament d'una dada de temperatura és inferior a 700 bytes.

Amb aquestes necessitats, a continuació es fa una breu comparativa entre la tecnologia Wi-Fi 802.11g utilitzada en el projecte, mitjançant el mòdul WiFly RN-XV, i altres tecnologies de comunicació mòbil que també es podrien haver implementat:

	Wi-Fi / 802.11g	GPRS	3G / UMTS	3.5G / HSPA
<b>Velocitat de transmissió</b>	54 Mbps	115 Kbps	2 Mbps	14 Mbps
<b>Disponibilitat de mòduls</b>	Alta	Baixa	Molt baixa	Molt baixa
<b>Preu orientatiu dels mòduls</b>	30 €	60-80 €	150-200 €	150-200 €

Figura 16: Comparativa de tecnologies de comunicació sense fils

Tal com es pot veure, la solució basada en Wi-Fi 802.11g és la que ofereix majors prestacions i també un preu més reduït. L'inconvenient d'aquesta tecnologia és que es requereix d'una infraestructura prèvia (un punt d'accés que ofereixi connectivitat Wi-Fi) en l'espai on es vol instal·lar el dispositiu. En el cas que ens ocupa això no és un aspecte crític, donat que el dispositiu està dissenyat per a ser instal·lar en entorns crítics, com sales de servidors, on habitualment ja es disposa d'aquest tipus de connectivitat.

En cas de voler dotar de major autonomia al dispositiu, es podria utilitzar un protocol de comunicació mòbil, que només requerís d'una targeta SIM per a disposar de connexió a Internet. En aquest cas, donat el reduït volum de dades, seria suficient amb un mòdul de connectivitat GPRS.

## 2.2. Estudi de mercat

Fent una recerca per Internet, s'han trobat diverses solucions comercials similars a la desenvolupada en aquest treball. A continuació s'analitzen dos d'aquests dispositius (HWg-STE i HWg-Ares12) i es comparen amb la solució dissenyada:

	Solució dissenyada	HWg-STE	HWg-Ares12
<b>Connectivitat</b>	Wi-Fi 802.11g	Ethernet	GSM/GPRS
<b>Configuració i visualització</b>	Aplicació web a Internet	Aplicació web encastada	Aplicació web encastada
<b>Alertes</b>	E-mail i alarma sonora	E-mail	E-mail i SMS
<b>Precisió</b>	$\pm 0.5^{\circ}\text{C}$	$\pm 0.5^{\circ}\text{C}$	$\pm 0.5^{\circ}\text{C}$
<b>Preu</b>	-	155 €	549 €

Figura 17: Comparativa de solucions d'alarma i monitorització

Les solucions comercials analitzades disposen de característiques similars a la solució dissenyada en aquest treball, com és el sistema d'alertes basat en correu electrònic, o el fet de disposar d'un entorn web per a la configuració i visualització de dades.

Les principals diferències es troben en la connectivitat i en el fet de que l'aplicació web estigui encastada en el propi dispositiu, en lloc d'estar a Internet. D'una banda, això comporta l'avantatge de no requerir una aplicació externa per funcionar, però també afegeix limitacions al sistema: és necessari configurar l'encaminador per permetre tràfic entrant si es vol accedir a l'entorn web des d'Internet i l'espai es limita (no permet disposar d'un històric molt gran).

A continuació es mostren unes imatges dels productes analitzats:



Figura 18: HWg-STE. Termòmetre Ethernet amb alertes per e-mail



Figura 19: HWg-Ares12. Termòmetre GSM/GPRS amb alertes per e-mail i SMS

### 3. Descripció funcional

En aquest capítol es presenta el disseny de la solució proposada i es descriu des d'un punt de vista funcional. En primer lloc, es presenta el sistema complet, per posteriorment explicar el disseny de l'aplicació desenvolupada al dispositiu integrat de control i de l'aplicació servidor.

#### 3.1. Sistema d'alarma i monitorització ambiental amb gestió remota

La solució dissenyada està formada per diferents elements que es comuniquen entre ells. En aquest apartat es descriu el funcionament total del sistema i la comunicació entre els diferents elements que el conformen.

##### 3.1.1. Diagrama de blocs

A continuació es presenta el diagrama de blocs del sistema:

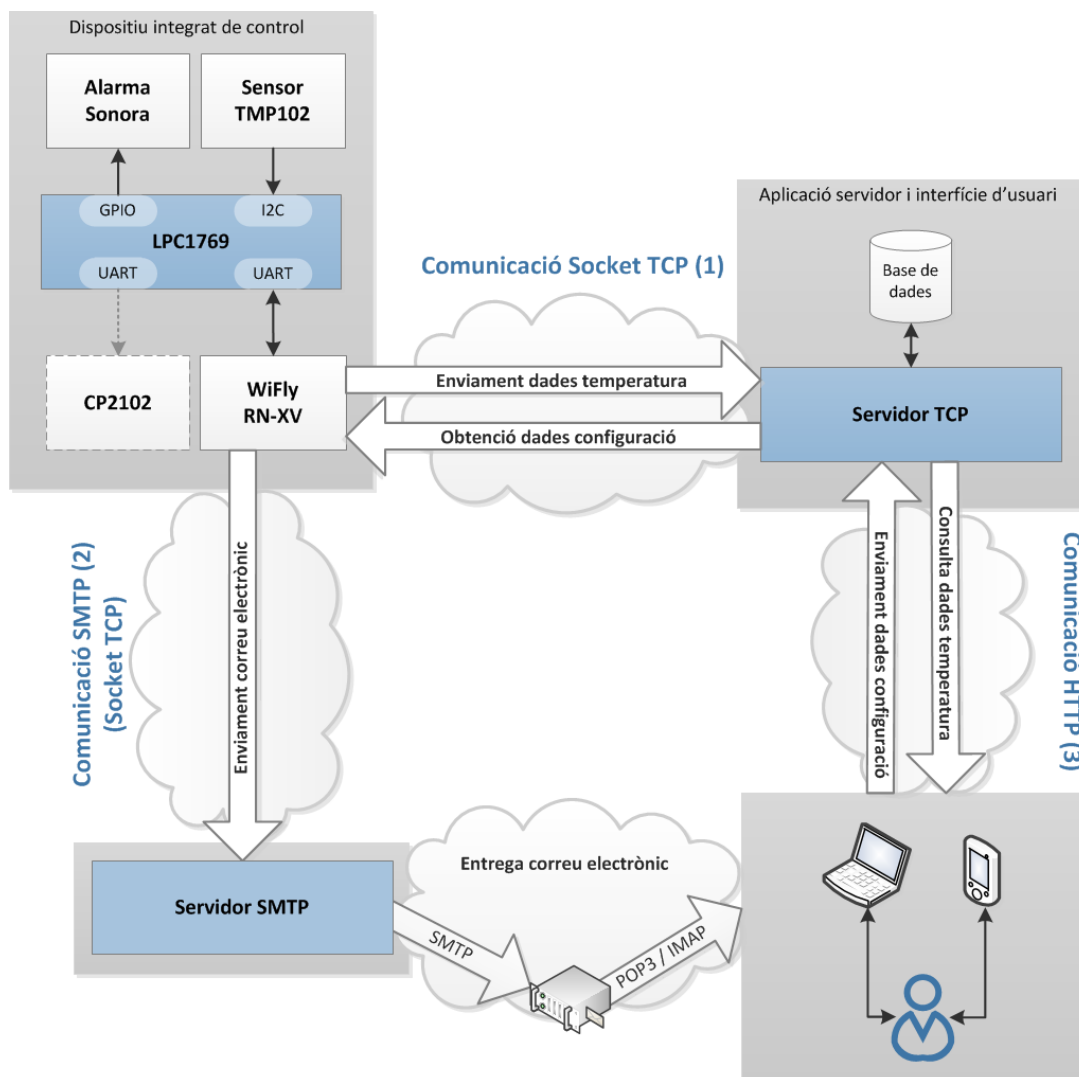


Figura 20: Diagrama de blocs de la solució proposada

### **Dispositiu integrat de control amb connexió Wi-Fi**

Està format per la placa LPC1769, el sensor de temperatura, el bronzidor, el convertidor CP2102 i el mòdul WiFly RN-XV que dota al sistema de connexió a Internet a través de Wi-Fi. Aquesta part del sistema està ubicada físicament a l'espai que es vol monitoritzar, i és l'encarregada d'obtenir periòdicament les mesures de temperatura, de gestionar l'estat d'alarma, d'enviar les mesures obtingudes al servidor perquè puguin ser consultades per l'usuari i de mantenir actualitzats els paràmetres de configuració del sistema proporcionats pel servidor (temperatura de seguretat màxima i mínima, freqüència d'obtenció de mesures, dades del servidor SMTP i estat de l'alarma sonora).

Quan el dispositiu detecta que les temperatures obtingudes estan fora dels marges de seguretat establerts, envia de manera autònoma un correu electrònic d'alerta a l'adreça que l'usuari hagi especificat i, al mateix temps, activa l'alarma sonora en cas que estigui habilitada per configuració. Un cop normalitzada la situació, s'informa a l'usuari també a través de correu electrònic i es desactiva l'alarma sonora.

El mòdul WiFly incorporat permet la comunicació amb l'aplicació servidor i amb el servidor SMTP a través d'una connexió *socket* TCP, tal com es detallarà en el punt següent.

El dispositiu també incorpora un mòdul CP2102 amb connexió USB, que ofereix a l'usuari la possibilitat de veure per consola, a través del seu ordinador, possibles missatges d'error que poden ser d'utilitat per detectar i solucionar problemes en el funcionament del sistema.

Finament, s'ha dotat al dispositiu d'un sistema d'alimentació que permet connectar-lo a la xarxa elèctrica a través d'un transformador incorporat. La decisió d'optar per aquest tipus d'alimentació ve donada pels propis requeriments de la solució, dissenyada per ser instal·lada en entorns crítics o d'alta disponibilitat, com sales de servidors, que en la major part dels casos disposen de sistemes d'alimentació ininterrompuda. Tot i així, donat que el transformador 220VAC – USB és extraïble, l'usuari també té la possibilitat de connectar el dispositiu a una bateria amb connexió USB.

### **Aplicació servidor i interfície d'usuari**

Està format per un servidor TCP que atén les peticions del dispositiu integrat de control, una base de dades on s'emmagatzemen les mesures de temperatura rebudes i els paràmetres de configuració del sistema, i una interfície web que permet a l'usuari configurar el dispositiu de manera remota i visualitzar les dades de temperatura en temps real.

L'aplicació dissenyada és multi-plataforma i està preparada per funcionar en la majoria de servidors, sota entorns Windows o Linux.

### 3.1.2. Comunicació entre els diferents elements del sistema

A continuació es descriu la comunicació entre els diferents elements del sistema:

- **Comunicació entre el dispositiu integrat de control i el servidor TCP (1)**

La comunicació es realitza utilitzant el protocol TCP i, més concretament, a través d'un *socket* orientat a connexió que s'estableix entre els dos extrems, seguint un model client-servidor. A través del *socket*, el dispositiu integrat de control envia les dades de temperatura obtingudes pel sensor, i obté les dades de configuració del sistema que el servidor li proporciona.

La decisió d'optar per un protocol de transport orientat a connexió, en lloc d'un protocol no orientat a connexió (com UDP), ve donada per la necessitat de garantir que totes les dades transmises arriben correctament al destí. El fet d'utilitzar TCP proporciona fiabilitat a l'aplicació, garantint el lliurament de tota la informació en el mateix ordre en què ha estat transmesa. Aquest aspecte és essencial pel que fa a l'enviament dels paràmetres de configuració.

Pel que fa a la transmissió de les dades de temperatura, es va considerar la possibilitat de realitzar l'enviament a través de datagrames, utilitzant un *socket* no orientat a connexió amb protocol UDP. Tot i així, aquest model de comunicació no oferia cap avantatge en aquest escenari, on el volum de dades a transmetre és mínim, i per contra, es perdia fiabilitat en el lliurament de la informació.

- **Comunicació entre el dispositiu integrat de control i el servidor SMTP (2)**

La comunicació també es realitza a través d'un *socket* TCP que s'estableix entre els dos extrems, però en aquest cas s'utilitza el protocol SMTP a la capa d'aplicació per a la transmissió de les dades.

Mitjançant el *socket*, el dispositiu integrat de control envia les comandes SMTP necessàries al servidor (comandes d'autenticació, remitent, destinatari, missatge...), per tal d'enviar un correu electrònic al destinatari que l'usuari hagi configurat.

Aquest disseny permet que el dispositiu integrat de control pugui enviar el correu electrònic de manera autònoma, utilitzant qualsevol servidor SMTP d'Internet, i possibilita que el correu d'alerta s'envii de manera satisfactòria encara que el servidor TCP estigui caigut. A més, el servidor SMTP utilitzat per l'enviament, així com els paràmetres de connexió del mateix (usuari, contrasenya, port...), són configurables per l'usuari a través de la interfície web.

### - Comunicació entre el servidor TCP i l'usuari (3)

Donat que el servidor incorpora una aplicació web, la comunicació entre el servidor i l'usuari es realitza a través de protocol HTTP. D'aquesta manera, l'usuari pot interactuar amb el servidor de manera senzilla, fent servir qualsevol dispositiu que incorpori un navegador web (ordinador portàtil, *smartphone*, tauleta...).

## 3.2. Disseny de l'aplicació desenvolupada al dispositiu integrat de control

Tal com s'ha indicat anteriorment, el dispositiu integrat de control està basat en una placa LPC1769 a la qual s'han connectat diferents perifèrics que possibiliten realitzar les funcionalitats descrites. En aquest apartat es descriu el disseny de l'aplicació desenvolupada a l'LPC per tal que aquest pugui interactuar amb els perifèrics connectats.

### 3.2.1. Diagrama de blocs

A continuació es presenta el diagrama de blocs del dispositiu integrat de control:

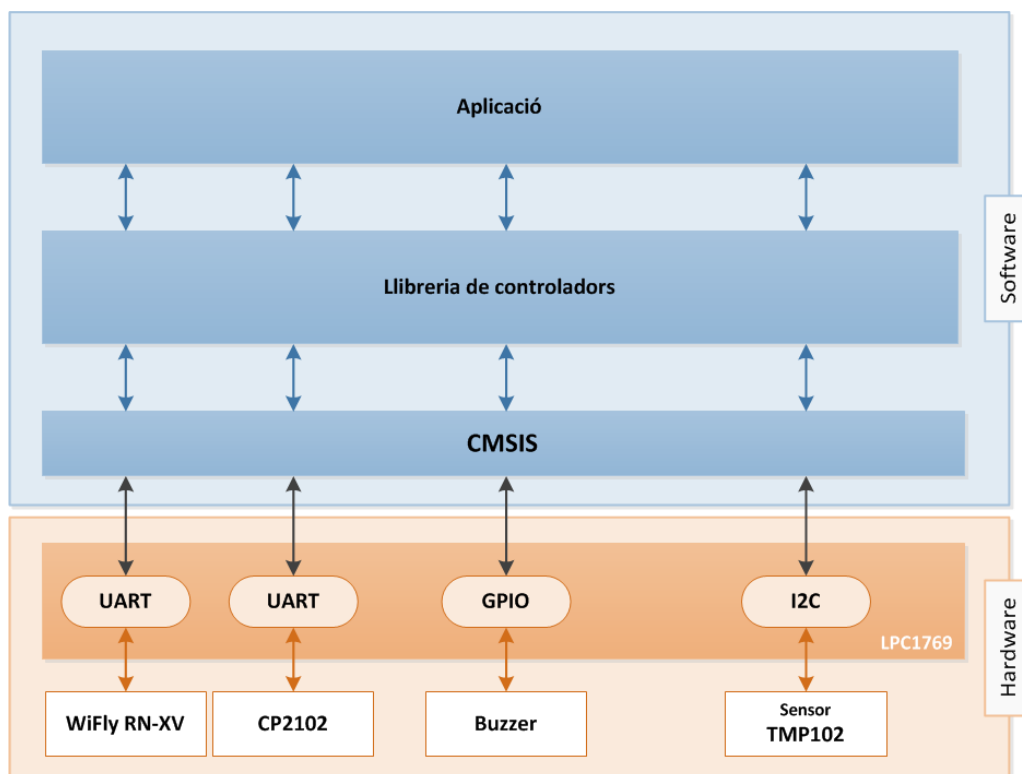


Figura 21: Diagrama de blocs simplificat de l'aplicació desenvolupada a l'LPC

Tal com s'observa al diagrama, d'una banda es disposa de l'LPC i dels diferents perifèrics connectats a les corresponents interfícies de la placa i, per l'altra, el programari necessari per fer funcionar el dispositiu.



L'aplicació desenvolupada està formada per tres grans capes:

- **Llibreria CMSIS:** És una llibreria proporcionada pel mateix fabricant NXP, que ofereix una capa d'abstracció de *hardware* de baix nivell per als processadors de la sèrie Cortex-M.
- **Llibreria de controladors:** Inclou les llibreries necessàries per accedir a les diferents interfícies (UART, I2C i GPIO), els controladors desenvolupats per interactuar amb els dispositius connectats a l'LPC (WiFly, CP2102, bronzidor i sensor), i la llibreria de comunicació amb l'aplicació servidor desenvolupada.
- **Aplicació:** És l'aplicació pròpiament dita, on estan definides i implementades les diferents funcionalitats del sistema com a tasques independents que interactuen amb els controladors desenvolupats a la llibreria.

Tant pel desenvolupament dels controladors, com de l'aplicació, s'han utilitzat els recursos que proporciona el sistema operatiu en temps real **FreeRTOS**. En el cas de la llibreria de controladors, s'han utilitzat *mútex* que garanteixen el correcte accés a les diferents interfícies, i per al desenvolupament de l'aplicació s'han utilitzat tasques, cues, prioritats i altres recursos que han possibilitat el funcionament de l'aplicació tal com s'havia dissenyat.

### 3.3. Disseny de l'aplicació servidor i interfície d'usuari

En aquest apartat es descriu el disseny de l'aplicació formada pel servidor TCP, la base de dades i la interfície d'usuari basada en web.

#### 3.3.1. Diagrama de blocs

A continuació es presenta el diagrama de blocs de l'aplicació:

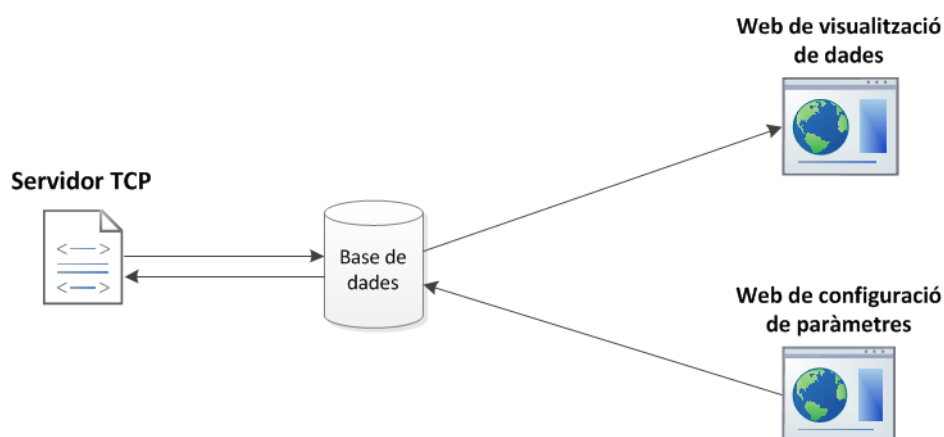


Figura 22: Diagrama de blocs simplificat de l'aplicació servidor i interfície d'usuari

### 3.3.2. Servidor i base de dades

Tal com s'ha explicat anteriorment, l'aplicació servidor atén les peticions del dispositiu integrat de control a través d'una comunicació *socket* TCP, seguint un model client-servidor. L'aplicació està connectada a una base de dades on s'emmagatzemen, d'una banda, les dades de temperatura rebudes per a la seva consulta per part de l'usuari i, d'altra banda, els paràmetres de configuració del sistema que el servidor proporciona al dispositiu integrat de control.

El desenvolupament de l'aplicació servidor s'ha realitzat utilitzant el llenguatge de programació PHP, i la base de dades utilitza el sistema de gestió MySQL. A continuació s'exposen els motius d'aquestes decisions:

- Tant PHP com MySQL són multi-plataforma, el que permet que l'aplicació dissenyada pugui ser utilitzada en la majoria de servidors, amb independència de si utilitzen un sistema operatiu basat en Windows o Linux.
- Permeten ser desplegats utilitzant una infraestructura LAMP, sense cap cost en concepte de llicències de programari per a l'usuari final.
- Donat que l'aplicació servidor i la interfície d'usuari estan desenvolupats en el mateix llenguatge de programació, es redueixen els requisits de programari del servidor físic on s'executa l'aplicació.

### 3.3.3. Interfície d'usuari

S'ha desenvolupat una interfície web que permet a l'usuari final interactuar remotament amb el dispositiu integrat de control a través de l'aplicació servidor, i visualitzar en temps real les dades de temperatura obtingudes.

La interfície d'usuari ha estat desenvolupada en PHP i HTML, i està formada per un web de visualització de dades i un web de configuració de paràmetres.

#### **Web de visualització de dades**

A través del web de visualització de dades, es mostren les dades de temperatura emmagatzemades a la base de dades MySQL. D'aquesta manera, l'usuari pot visualitzar a través d'un gràfic interactiu que s'actualitza en temps real, les dades de temperatura obtingudes pel dispositiu integrat de control.

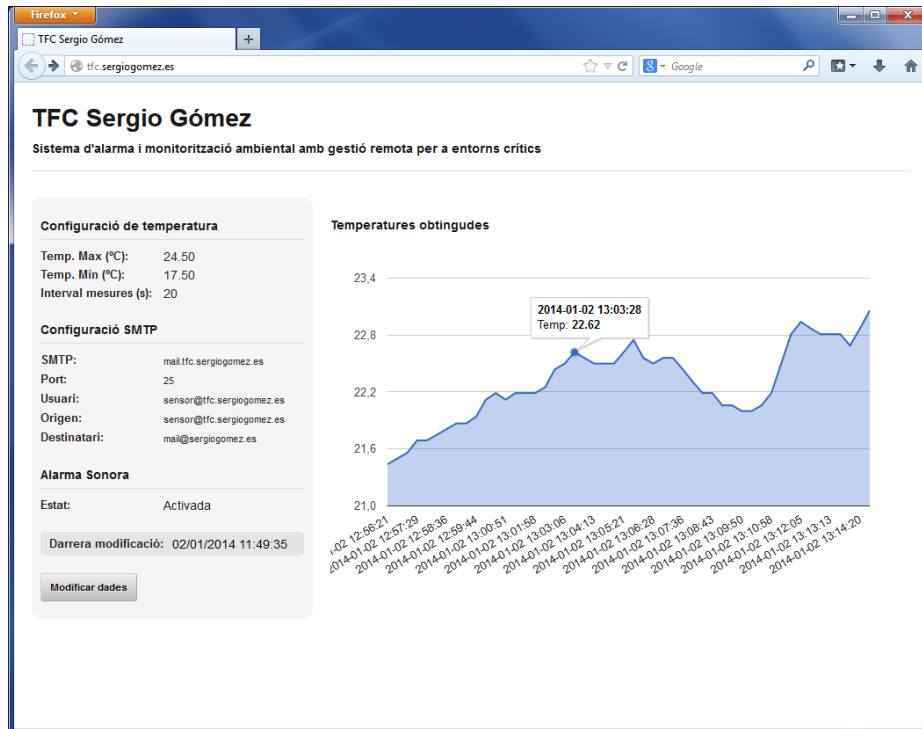


Figura 23: Captura de pantalla del web de visualització de temperatures

### Web de configuració de paràmetres

El web de configuració de paràmetres permet a l'usuari modificar les dades de configuració del sistema emmagatzemades a la base de dades, que el servidor proporciona al dispositiu integrat de control.

Figura 24: Captura de pantalla del web de configuració

Els paràmetres que l'usuari pot modificar a través del web de configuració són les següents:

Configuració de mesures	
<b>Temperatura màxima</b>	Temperatura màxima de seguretat, en graus.
<b>Temperatura mínima</b>	Temperatura mínima de seguretat, en graus.
<b>Interval de mesures</b>	Interval de temps, en segons, en el que el dispositiu integrat de control obté una nova mesura de temperatura.
Dades de configuració del servidor SMTP	
<b>Servidor</b>	Nom del host del servidor SMTP.
<b>Port</b>	Port del servidor SMTP, habitualment 25 o 587.
<b>Usuari</b>	Nom d'usuari per a l'autenticació amb el servidor.
<b>Contrasenya</b>	Contrasenya per a l'autenticació amb el servidor.
<b>Correu d'origen</b>	Adreça de correu que figurarà com a remitent de l'enviament.
<b>Correu destí</b>	Correu electrònic on s'enviaran els missatges d'alarma.
Alarma sonora	
<b>Estat de l'alarma</b>	Activada o desactivada. Indica al dispositiu integrat de control si ha d'activar l'alarma en sonora en de que les mesures estiguin fora dels marges de seguretat. També permet desconnectar l'alarma sonora si està sonant.

Figura 25: Paràmetres configurables per l'usuari des de l'entorn web

## **4. Descripció detallada**

En aquest capítol es descriu el sistema dissenyat de forma tècnica i detallada. En primer lloc, s'explica el funcionament de l'aplicació desenvolupada al dispositiu integrat de control, per a posteriorment descriure amb detall cadascuna de les funcionalitats implementades.

Seguidament es detallen els aspectes de comunicació entre els diferents elements del sistema, analitzant les diferents fases de la comunicació i el detall de les trames enviades, per a continuació descriure els aspectes tècnics de l'aplicació servidor, la base de dades i l'aplicació web desenvolupada.

Finalment, s'adjunta un esquema de connexions entre l'LPC i els diferents components utilitzats.

### **4.1. Aplicació desenvolupada al dispositiu integrat de control**

En aquest apartat es descriu el funcionament de l'aplicació desenvolupada al dispositiu integrat de control, detallant els diferents controladors que conformen la llibreria creada i les tasques que possibiliten el funcionament de l'aplicació

#### **4.1.1. Diagrama de blocs**

En el següent diagrama, es mostra el detall de les llibreries i tasques creades, i la seva comunicació entre elles.

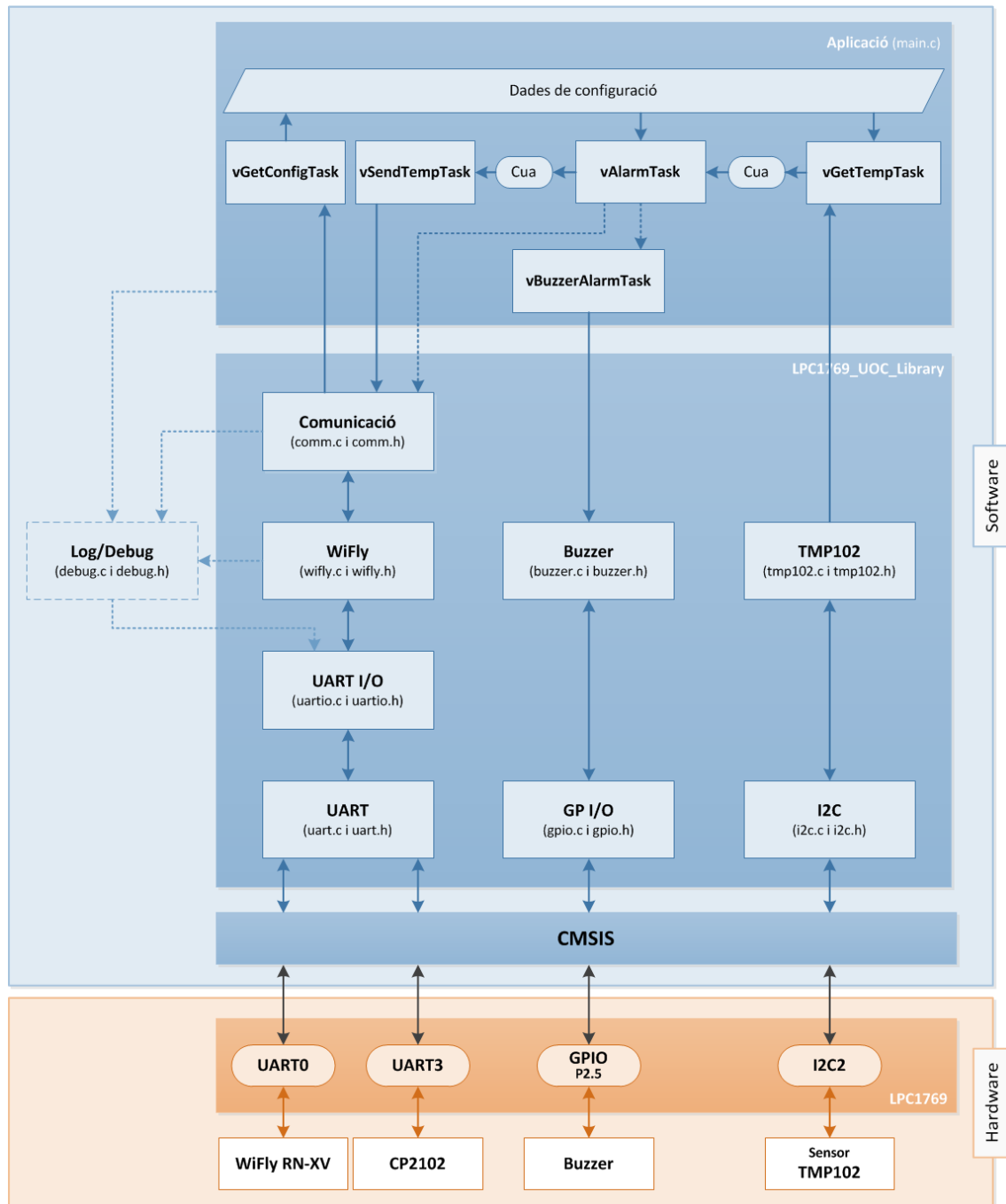


Figura 26: Diagrama de blocs detallat de l'aplicació desenvolupada a l'LPC

#### 4.1.2. Llibreria de controladors

La llibreria de controladors s'ha desenvolupat seguint un disseny modular i per capes, en el que cada controlador implementat té unes tasques acotades i realitza les seves pròpies funcionalitats. Els controladors de la capa més baixa (UART, GP I/O i I2C) interactuen directament amb el maquinari a través de la llibreria CMSIS. A partir d'aquí, les llibreries de les capes superiors utilitzen les funcions implementades en els controladors de la capa més baixa,

per incorporar funcions que permetin interactuar amb els diferents dispositius connectats de manera més senzilla per al programador.

A continuació es descriuen les funcionalitats de cadascuna de les llibreries i controladors desenvolupats i que integren l'aplicació:

- **UART:** Permet interactuar amb la interfície UART de la placa. S'ha agafat com a base la llibreria proporcionada pel fabricant, que incorpora funcions d'inicialització i escriptura a través dels ports UART (0, 1 i 3) i se li han afegit noves funcions que permeten també obtenir dades per qualsevol d'aquests tres ports.
- **UART I/O:** Incorpora funcions que permeten llegir i escriure pel port UART especificat de manera més senzilla i còmode per al programador.
- **WiFly:** Controlador del mòdul WiFly RN-XV. Incorpora funcions que permeten interactuar amb el dispositiu a través de la interfície UART, tals com connexió a una xarxa sense fils, obtenció i modificació de paràmetres del mòdul, connexió via *socket* TCP a un *host* remot, enviament i recepció de dades a través del *socket* i control d'errors de connectivitat, entre d'altres.
- **Comunicació:** Inclou les funcions per a la comunicació amb el servidor TCP desenvolupat (enviament de temperatures, obtenció de paràmetres de configuració...) i per a l'enviament de correus electrònics a través d'un servidor SMTP.
- **GP I/O:** Permet interactuar amb els pins GP I/O de la placa, incorporant funcions per a la seva inicialització, configuració de direcció, i establiment i obtenció de l'estat de cada pin (HIGH o LOW).
- **Buzzer:** Incorpora funcions que permeten activar i desactivar l'alarma sonora de manera còmode per al programador, utilitzant les funcions desenvolupades a la llibreria GP I/O.
- **I2C:** Permet interactuar amb la interfície I2C de la placa. Aquesta llibreria és proporcionada pel fabricant, i s'ha modificat per tal d'adaptar-la a la versió de CMSIS utilitzada en el projecte (CMSISv2p00), així com també s'han corregut alguns errors que incorporava la versió original proporcionada.
- **TMP102:** Controlador del sensor de temperatura TMP102. Incorpora funcions que permeten inicialitzar i obtenir les mesures de temperatura del sensor a través de la interfície I2C de la placa, utilitzant les funcions de la llibreria I2C.

- **Log/Debug:** Aquesta llibreria és utilitzada per altres llibreries i per l'aplicació principal, per tal d'imprimir, a través del mòdul CP2102, missatges d'informació i/o error que poden ser visualitzats per l'usuari amb una aplicació terminal instal·lada en el seu ordinador. La llibreria apareix en línia discontinua en el diagrama, donat que implementa una funcionalitat d'ús opcional.

#### 4.1.3. Aplicació i tasques

L'aplicació s'ha desenvolupat utilitzant els recursos que proporciona el sistema operatiu en temps real FreeRTOS, tals com *mútxs*, tasques, cues i prioritats.

Les diferents funcionalitats s'han implementat com a tasques, que realitzen funcionalitats independents. El planificador incorporat en el sistema operatiu és l'encarregat d'executar les diferents tasques al processador, d'acord amb els temps i les prioritats definides.

Les tasques que conformen l'aplicació desenvolupada són les següents:

- **vGetTempTask:** Obté les mesures de temperatura del sensor de manera periòdica, segons l'interval de temps definit per l'usuari, i les posa a una cua a l'espera de que siguin processades.
- **vAlarmTask:** Obté les dades de temperatura de la cua, les processa i les posa a una segona cua, a l'espera que siguin enviades al servidor. És l'encarregada de comprovar si les temperatures estan dins o fora del marge de seguretat definit per l'usuari, d'enviar les notificacions per correu electrònic i d'activar o desactivar l'alarma sonora.
- **vBuzzerAlarmTask:** S'encarrega de fer funcionar l'alarma sonora. La tasca és activada o desactivada per vAlarmTask quan la situació ho requereix.
- **vSendTempTask:** Obté les dades de temperatura de la cua, un cop han estat processades, i les envia al servidor.
- **vGetConfigTask:** És l'encarregada de mantenir el sistema actualitzat. Comprova de manera periòdica si el servidor disposa de configuració més actualitzada i, en cas afirmatiu, actualitza els paràmetres de configuració del sistema amb els proporcionats pel servidor.



### Inicialització del dispositiu

Un cop s'encén el dispositiu, el primer que fa l'aplicació és inicialitzar els diferents perifèrics (brunzidor, mòdul CP2102, sensor TMP102 i mòdul WiFly RN-XV). Posteriorment es prova d'associar el mòdul WiFly a la xarxa sense fils segons els paràmetres incorporats a la memòria flash de l'LPC i, si la connexió té èxit, es configura el mòdul WiFly per primer cop, amb els paràmetres de connexió del servidor TCP també guardats a la memòria flash del dispositiu. Finalment, es creen les cues, les diferents tasques descrites anteriorment i s'inicialitza el planificador.

A partir d'aquest moment és el planificador qui controla el funcionament de l'aplicació, gestionant l'execució de les diferents tasques.

En aquest sentit, segons el disseny realitzat, el primer que fa el dispositiu és connectar-se al servidor TCP i obtenir tots els paràmetres de configuració del sistema poder començar a funcionar.

A continuació es mostra el diagrama de flux de la inicialització del dispositiu:

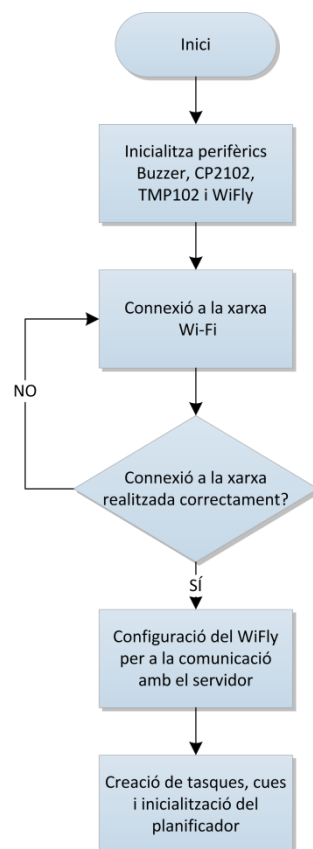


Figura 27: Diagrama de flux de la inicialització del dispositiu integrat de control

## 4.2. Detall de les implementacions

En aquest apartat es descriuen amb detall les diferents funcionalitats implementades sobre el dispositiu integrat de control.

### 4.2.1. Configuració i gestió remota del sistema

El dispositiu integrat de control és l'encarregat de mantenir actualitzats els paràmetres de configuració del sistema. Per a fer-ho, l'aplicació comprova periòdicament si el servidor disposa de nova configuració i, en cas afirmatiu, actualitza les seves dades amb les que el servidor li proporciona.

Per tal de comprovar si es disposa de configuració més actualitzada, s'ha dissenyat un sistema basat en segell de temps (o *timestamp*), on les dades de configuració van acompanyades d'una marca o segell que indica el moment exacte (data, hora, minuts i segons) en què han estat actualitzades. Concretament, com a segell de temps s'ha fet servir el "Temps Unix" o temps POSIX, que és un nombre enter que indica el nombre de segons transcorreguts des de les 00:00:00 de l'1 de gener de 1970, UTC.

D'aquesta manera, el dispositiu demana de manera periòdica al servidor el valor *timestamp* de la darrera actualització dels paràmetres de configuració i compara el valor obtingut amb el valor *timestamp* del que disposa l'LPC. Si el valor obtingut és superior, i per tant, el servidor disposa de configuració més recent, l'aplicació demana al servidor tots els paràmetres de configuració i els guarda a l'LPC, junt amb el nou segell de temps.

Amb aquest disseny, s'aconsegueix optimitzar el temps destinat a la tasca `vGetConfigTask`, donat que es garanteix que el procés d'obtenció de nous paràmetres de configuració només es realitza en cas que realment aquests s'hagin modificat.

D'altra banda, l'interval de temps en què l'aplicació comprova si hi ha nova configuració disponible és el mateix que el definit per l'usuari per a l'obtenció i enviament de les mesures de temperatura, sempre que aquest sigui inferior a 60 segons, que és el temps màxim en què l'aplicació comprova si el servidor disposa de nova configuració.

D'aquesta manera, s'aconsegueix un doble objectiu: d'una banda, es garanteix que en el moment d'obtenir una nova dada de temperatura del sensor, l'aplicació sempre disposarà de configuració actualitzada i podrà processa-la d'acord amb els marges de temperatura definits, i d'altra banda, que la configuració enviada per l'usuari a través de l'entorn web serà efectiva a l'LPC en un temps inferior a 1 minut.

A continuació es mostra el diagrama de flux corresponent al procediment d'actualització dels paràmetres de configuració per part del dispositiu integrat de control:

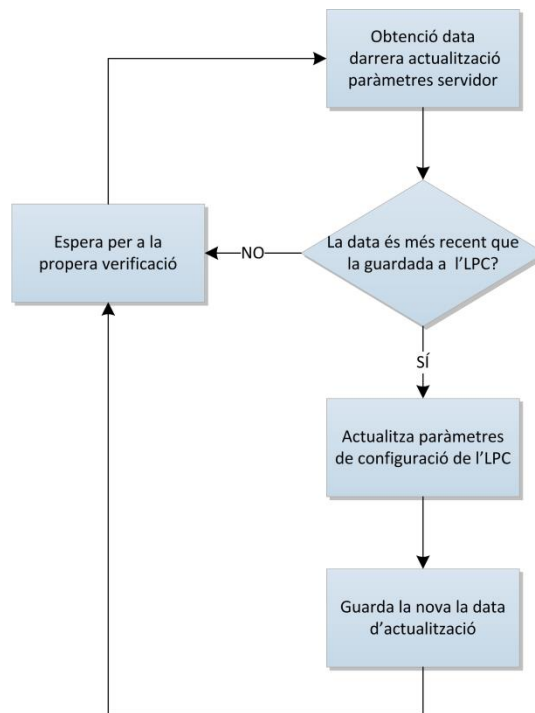


Figura 28: Diagrama de flux de la funcionalitat de configuració i gestió remota del sistema

#### 4.2.2. Obtenció i enviament de mesures de temperatura

La tasca `vGetTempTask` obté de manera periòdica, segons l'interval de temps configurat per l'usuari, les dades de temperatura del sensor digital i les posa a una cua a l'espera que siguin processades pel sistema d'alarma. Un cop les dades són processades, la tasca `vSendTempTask` les envia de manera immediata al servidor a través del *socket* TCP.

##### Obtenció de temperatures del sensor

L'obtenció de les dades de temperatura, es realitza a través de les funcions desenvolupades a la llibreria TMP102, que es comuniquen amb el sensor a través de la interfície I2C de l'LPC, utilitzant la pròpia llibreria I2C proporcionada pel fabricant NXP.

Amb l'ús d'aquesta llibreria, només és necessari conèixer l'adreça sèrie del sensor un cop connectat (*serial bus address*), per tal de poder accedir al seu registre intern i poder llegir la dada de temperatura en cada moment. En el cas del sensor TMP102 emprat, aquesta adreça està formada per 7 bits que es determinen segons si el pin ADD0 està alimentat o no, més un bit que indica si l'operació és de lectura o escriptura.

Un cop determinada l'adreça del sensor, s'accedeix al seu registre intern per tal de llegir la dada de temperatura obtinguda. En el cas del TMP102, el registre de temperatura està format per 12 bits que caldrà convertir a graus segons les especificacions del *datasheet*.

A continuació es mostra el diagrama de flux del procediment d'obtenció i enviament de les dades de temperatura:

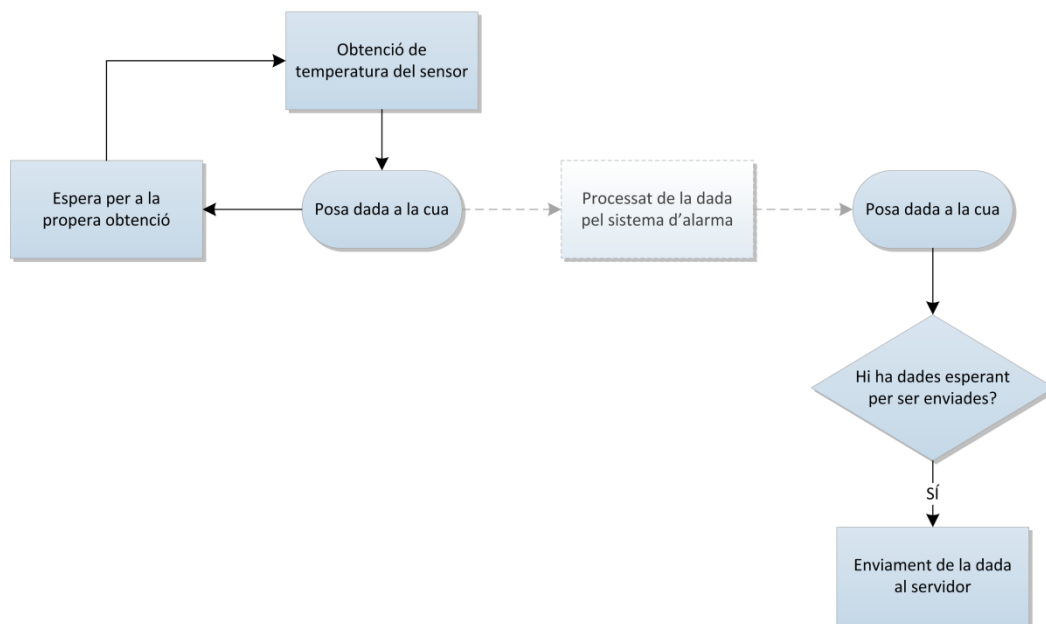


Figura 29: Diagrama de flux de l'obtenció i enviament de les dades de temperatura

### 4.2.3. Sistema d'alarma

Les dades de temperatura obtingudes a través del sensor són processades abans de ser enviades al servidor. En aquesta fase de processat, realitzada per la tasca `vAlarmTask`, es determinen les accions d'alerta a realitzar en funció de si la temperatura obtinguda està dins o fora dels marges de seguretat establerts per l'usuari.

Segons el disseny realitzat, el sistema pot estar en dos estats:

- **Estat d'alarma ACTIVAT:** les temperatures estan fora dels límits de seguretat.
- **Estat d'alarma DESACTIVAT:** les temperatures estan dins dels límits de seguretat.

L'aplicació s'inicia amb l'estat d'alarma desactivat. En el moment que s'obté una temperatura fora dels marges de seguretat, s'activa l'estat d'alarma. En aquest sentit, per tal de tornar a l'estat d'alarma desactivat, és necessari que la temperatura obtinguda estigui  $0.5^{\circ}\text{C}$  per sobre del límit mínim de seguretat i  $0.5^{\circ}\text{C}$  per sota del límit màxim.

D'aquesta manera, donat que les accions d'alerta a l'usuari es realitzen quan es canvia d'un estat a l'altre, s'evita que el sistema envii alertes d'activació/desactivació de forma continuada quan la temperatura oscil·la en un dels límits.

Tal com s'ha indicat en altres apartats, el sistema disposa de dos tipus d'alertes per a l'usuari: alerta per correu electrònic i alarma sonora.

#### **Alerta per correu electrònic**

L'aplicació envia un correu electrònic d'alerta a l'usuari quan s'activa l'estat d'alarma. En el correu, que s'envia a l'adreça que l'usuari hagi especificat a través del web de configuració, s'indica la temperatura que ha provocat que el sistema entri en l'estat d'alarma.

De la mateixa manera, un cop es torna a l'estat d'alarma desactivat, l'aplicació envia un altre correu electrònic a l'usuari per informar que la situació s'ha normalitzat

#### **Alarma sonora**

A diferència de d'alerta per correu electrònic, l'alarma sonora és una funcionalitat que l'usuari pot habilitar o deshabilitar a través del web de configuració de paràmetres.

En cas que la funcionalitat estigui habilitada, el dispositiu emetrà un so intermitent mentre l'estat d'alarma estigui activat, i s'apagarà quan es torni a l'estat desactivat. Tot i així, també s'ha previst que l'usuari pugui canviar la configuració de l'alarma sonora, i que el canvi sigui efectiu en qualsevol moment de l'execució de l'aplicació. D'aquesta manera, l'usuari té la possibilitat d'apagar l'alarma sonora mitjançant l'entorn web de configuració, quan aquesta estigui sonant.

Per tal de posar en marxa l'alarma sonora, la tasca `vAlarmTask` activa `vBuzzerAlarmTask`, que és la tasca encarregada d'interactuar amb la llibreria `Buzzer`, i crida a les funcions corresponents per posar a HIGH i a LOW de manera repetida i continuada, el pin on està connectat el bronzidor.

A continuació es mostra el diagrama de flux corresponent a la funcionalitat d'alarma desenvolupada.

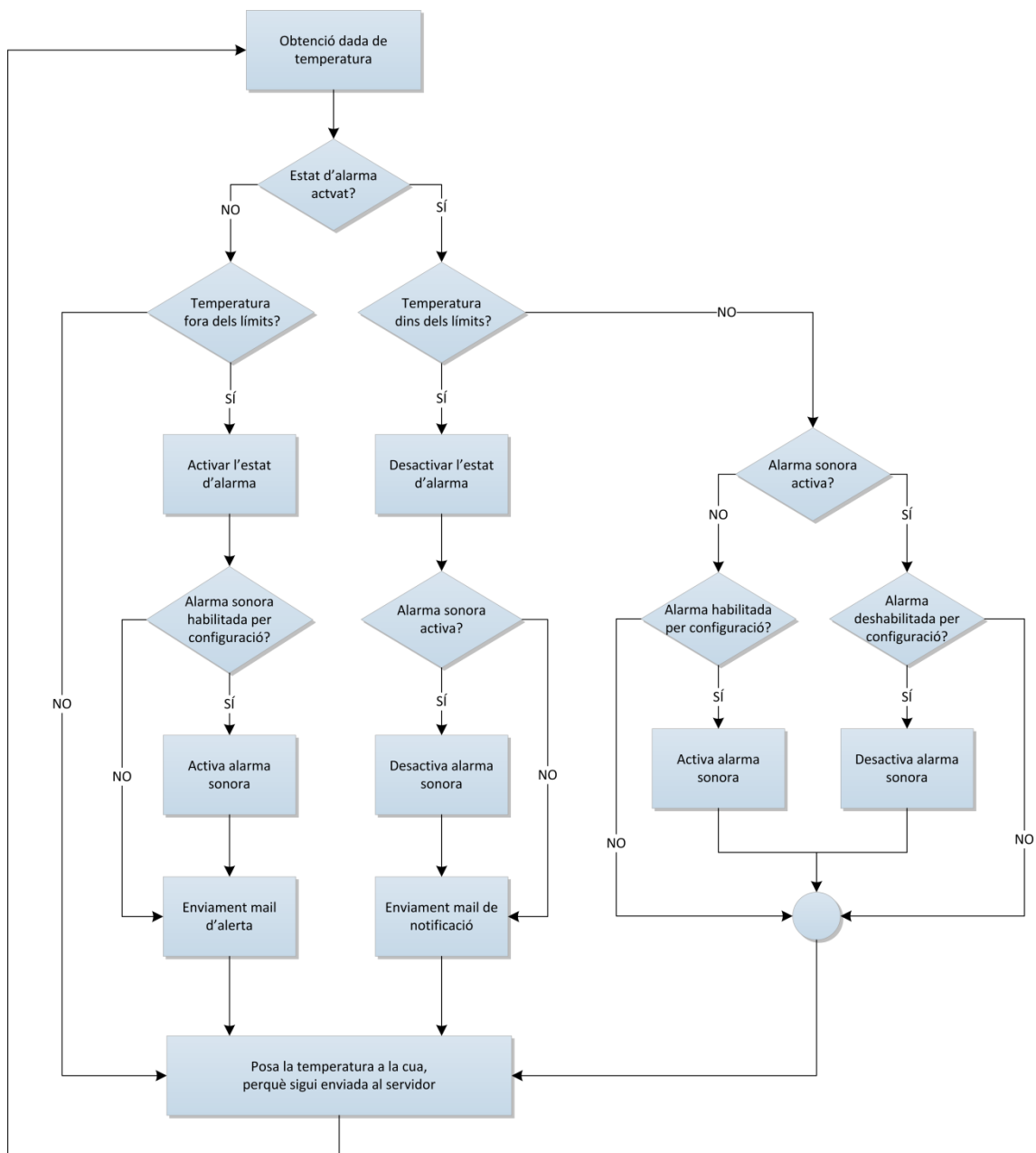


Figura 30: Diagrama de flux de la funcionalitat d'alarma

A continuació es mostra un exemple dels correus electrònics rebuts per l'usuari, des d'un dispositiu mòbil, per a una configuració de temperatura màxima de 23.5°C. Com es pot veure, el correu d'alarma s'envia quan la temperatura obtinguda és de 23.56°C i el correu de restabliment quan la temperatura és inferior a 23°C (0.5°C per sota de la temperatura màxima establerta).



Figura 31: Correu electrònic d'alarma rebut des d'un dispositiu mòbil

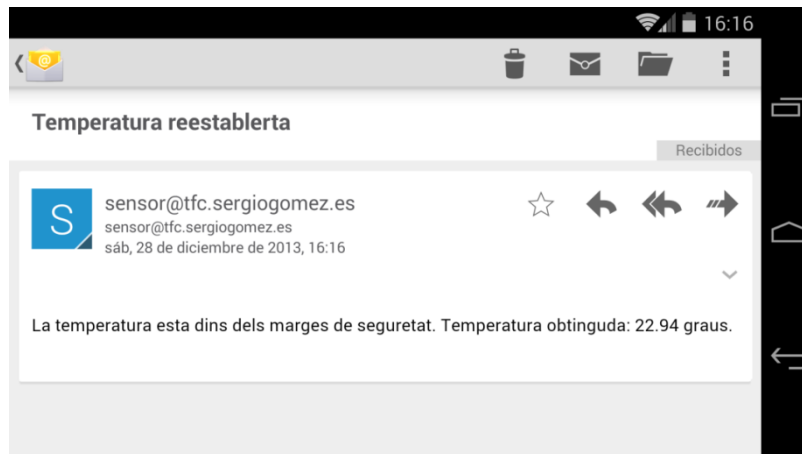


Figura 32: Correu electrònic d'informació rebut des d'un dispositiu mòbil

#### 4.2.4. Gestió de pèrdua de connexió

Per tal de prevenir problemes derivats de possibles pèrdues de connexió, s'ha implementat una funcionalitat mitjançant la qual es procedeix a reiniciar el mòdul WiFly en cas de detectar errors de connectivitat.

Quan no es completa una comunicació de manera satisfactòria (per exemple, no es rep la confirmació de temperatura rebuda pel part del servidor, o no es pot obtenir el segell de temps de la configuració), la llibreria corresponent reporta un error de connectivitat mitjançant una funció implementada al controlador del mòdul WiFly.

Un cop s'arriba al límit màxim d'errors permesos (per defecte, cinc), el controlador del WiFly procedeix a realitzar un *reset per hardware* al dispositiu.

A continuació es mostra el diagrama de flux de la funció que és cridada cada cop que es reporta un error de connectivitat:

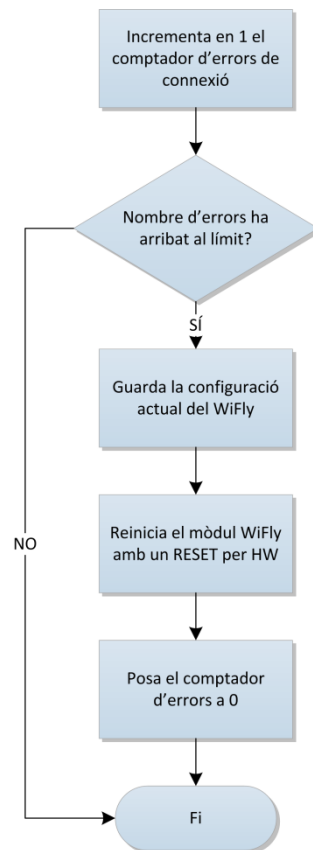


Figura 33: Diagrama de flux de la funcionalitat de gestió de la pèrdua de connexió

#### 4.2.5. Sistema de *log*

En dispositiu integrat de control incorpora un sistema de *log*, que ofereix a l'usuari la possibilitat de veure en temps real, a través del seu ordinador, missatges d'error i/o informació que poden ser d'utilitat per detectar i solucionar problemes en el funcionament del sistema.

El controlador del mòdul WiFly, la llibreria de comunicació i l'aplicació principal, utilitzen les funcions implementades a la llibreria Log/Debug per tal d'enviar al mòdul CP2102 els diferents missatges d'error i/o informació. L'usuari pot visualitzar aquests missatges connectant el seu ordinador al mòdul CP2102 a través d'USB, i utilitzant una aplicació terminal.

En l'aplicació final, el sistema de *log* està configurat per mostrar els següents missatges:

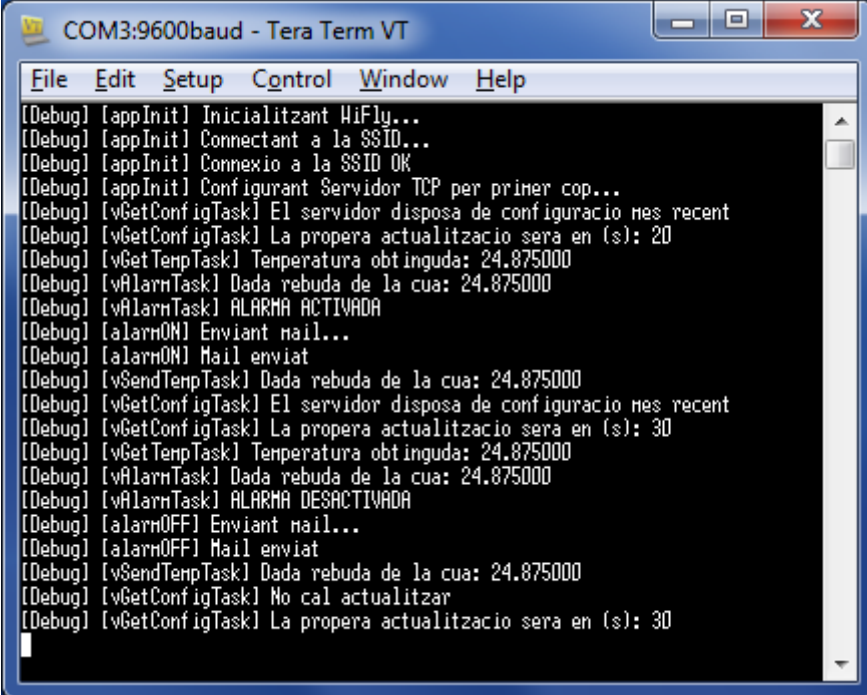
- **Missatges d'error de la llibreria de comunicació:** Problemes de connectivitat a través del *socket* TCP durant l'obtenció de dades de temperatura, obtenció de la data



de la darrera actualització, obtenció dels paràmetres de configuració o connexió amb el servidor SMTP.

- **Missatges d'informació i error de l'aplicació principal:** Connexió a la xarxa Wi-Fi, mesures de temperatura obtingudes, obtenció de dades de configuració, activació i desactivació de l'estat d'alarma...
- **Missatges de reinici del mòdul WiFly:** Informa quan el nombre d'errors de connectivitat ha arribat al màxim i s'ha procedit al reinici del mòdul WiFly.

A continuació es mostra una captura de pantalla dels missatges de *log* visualitzats a través de l'aplicació terminal *Tera Term* per a Windows:



```
COM3:9600baud - Tera Term VT
File Edit Setup Control Window Help
[Debug] [appInit] Inicialitzant WiFly...
[Debug] [appInit] Connectant a la SSID...
[Debug] [appInit] Connexio a la SSID OK
[Debug] [appInit] Configurant Servidor TCP per primer cop...
[Debug] [vGetConfigTask] El servidor disposa de configuracio mes recent
[Debug] [vGetConfigTask] La propera actualitzacio sera en (s): 20
[Debug] [vGetTempTask] Temperatura obtinguda: 24.875000
[Debug] [vAlarmTask] Dada rebuda de la cua: 24.875000
[Debug] [vAlarmTask] ALARMA ACTIVADA
[Debug] [alarmON] Enviant mail...
[Debug] [alarmON] Mail enviat
[Debug] [vSendTempTask] Dada rebuda de la cua: 24.875000
[Debug] [vGetConfigTask] El servidor disposa de configuracio mes recent
[Debug] [vGetConfigTask] La propera actualitzacio sera en (s): 30
[Debug] [vGetTempTask] Temperatura obtinguda: 24.875000
[Debug] [vAlarmTask] Dada rebuda de la cua: 24.875000
[Debug] [vAlarmTask] ALARMA DESACTIVADA
[Debug] [alarmOFF] Enviant mail...
[Debug] [alarmOFF] Mail enviat
[Debug] [vSendTempTask] Dada rebuda de la cua: 24.875000
[Debug] [vGetConfigTask] No cal actualitzar
[Debug] [vGetConfigTask] La propera actualitzacio sera en (s): 30
```

Figura 34: Missatges de *log* en temps real accessibles des d'una aplicació terminal

### 4.3. Detall de la comunicació entre els elements del sistema

En aquest apartat es detallen els aspectes de comunicació entre els diferents elements del sistema, explicant les diferents fases de la comunicació i analitzant les trames enviades.

#### 4.3.1. Comunicació amb el servidor. *Socket* TCP.

Tal com s'ha explicat en el capítol anterior, la comunicació entre el dispositiu integrat de control i el servidor es realitza a través d'un *socket* orientat a connexió (TCP).

Les fases de comunicació entre els dos extrems, són les següents:

- **Establiment de connexió:** L'establiment de la connexió es realitza mitjançant el procediment de negociació *three-way handshake*, que és l'utilitzat pel protocol TCP. En aquest cas, el dispositiu integrat de control realitza l'obertura activa de la connexió, enviant un segment amb el bit SYN activat al servidor TCP.
- **Salutació:** Un cop s'ha establert la connexió, el servidor envia el missatge "\*\*\*HELLO\*\*" a través del *socket* al dispositiu integrat de control. D'aquesta manera, l'aplicació desenvolupada a l'LPC sap que el *socket* s'ha creat correctament i que el servidor està preparat per rebre dades.
- **Intercanvi de dades:** En aquesta fase es produeix l'intercanvi de dades a través del *socket*. El dispositiu integrat de control pot enviar comandes al servidor per sol·licitar dades de configuració, o dades de temperatura. En el primer cas, el servidor enviarà el valor del paràmetre demanat i, en el segon, respondrà amb el missatge "\*\*DATAOK\*\*", per confirmar que ha rebut la temperatura correctament.
- **Finalització de connexió:** Un cop finalitzat l'intercanvi de dades, el dispositiu integrat de control envia el missatge "QUIT" a través del *socket*, per indicar al servidor que finalitzi la connexió. Al rebre el missatge, el servidor tanca el *socket* i envia un segment amb el bit RST activat a l'LPC, que dóna per finalitzada la comunicació enviant un segment amb el bit FIN activat.

A continuació es mostra el detall de segments TCP enviats i rebuts durant una comunicació entre el dispositiu integrat de control i el servidor, on l'LPC demana la data de la darrera actualització dels paràmetres de configuració al servidor.

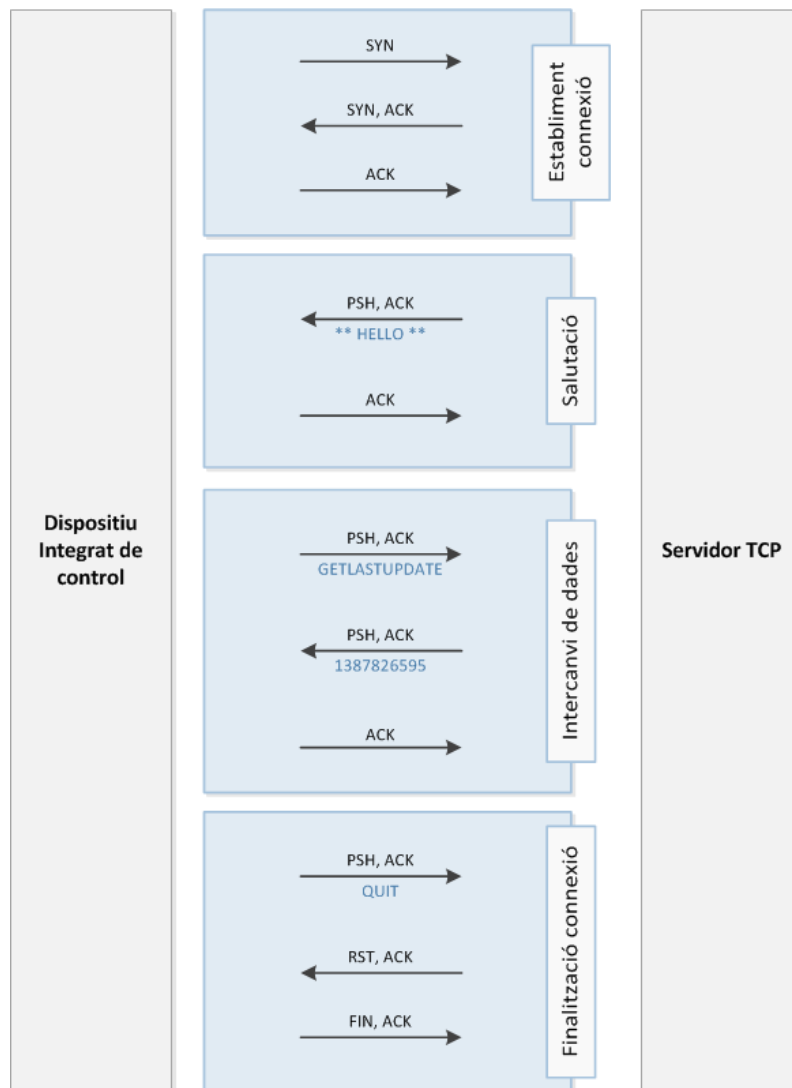


Figura 35: Detall de segments TCP en la comunicació entre l'LPC i el servidor

Tal com es pot veure en el diagrama, una característica destacada de la comunicació és que els segments TCP enviats a través del *socket* tenen el bit de control PSH activat, que invoca la funció *push* pròpia del protocol TCP.

Aquesta funció crea un segment TCP amb les dades que es volen enviar pel *socket* de manera immediata i activa el bit de control PSH del segment. Aquest bit indica al receptor que ha de lliurar a l'aplicació les dades disponibles al segment també de manera immediata, sense esperar a completar-les amb dades addicionals. D'aquesta manera, les dades s'envien i es reben sempre de manera immediata, sense esperar a omplir el segment TCP segons la mida màxima negociada (MSS).

En la imatge següent, es mostra la captura d'un segment TCP, corresponent a l'enviament d'una mesura de temperatura al servidor, a través del *socket*.

```

⊞ Frame 87: 61 bytes on wire (488 bits), 61 bytes captured (488 bits) on interface 0
⊞ Ethernet II, Src: RovingNe_80:37:07 (00:06:66:80:37:07), Dst: DellPcba_42:71:52 (f0:1f:af:42:71:52)
⊞ Internet Protocol Version 4, Src: 192.168.1.19 (192.168.1.19), Dst: 192.168.1.18 (192.168.1.18)
⊞ Transmission Control Protocol, Src Port: 42962 (42962), Dst Port: x11 (6001), Seq: 1, Ack: 10, Len: 7
  Source port: 42962 (42962)
  Destination port: x11 (6001)
  [Stream index: 10]
  Sequence number: 1 (relative sequence number)
  [Next sequence number: 8 (relative sequence number)]
  Acknowledgment number: 10 (relative ack number)
  Header length: 20 bytes
  ⊞ Flags: 0x018 (PSH, ACK)
  window size value: 2048
  [Calculated window size: 2048]
  [window size scaling factor: -2 (no window scaling used)]
  ⊞ Checksum: 0x318e [validation disabled]
  ⊞ [SEQ/ACK analysis]
    TCP segment data (7 bytes)
0000 f0 1f af 42 71 52 00 06 66 80 37 07 08 00 45 00 ...BqR.. f.7...E.
0010 00 2f 00 45 00 00 ff 06 38 0e c0 a8 01 13 c0 a8 ../.E... 8.....
0020 01 12 a7 d2 17 71 00 00 1a f8 f5 08 80 0a 50 18 .....q... ..P.
0030 08 00 31 8e 00 00 32 34 2e 31 39 0d 0a ...1...24 .19..

```

Figura 36: Captura d'un segment TCP enviat a través del socket

Com es veu a la captura, la dada de temperatura apareix com a “TCP segment data”, donat que no s'està utilitzant cap protocol de capa d'aplicació per a l'enviament de les dades.

El detall dels camps del segment capturat és el següent:

	Camp	Mida	Descripció
Capçalera (20 bytes)	Port d'origen	16 bits	Port local usat pel WiFly (42962).
	Port destinació	16 bits	Port del servidor TCP (6001).
	Núm. de seqüència	32 bits	Número de seqüència.
	Número ACK	32 bits	Número de reconeixement.
	Longitud capçalera	4 bits	Longitud indicada en paraules de 32 bits. Donat que el valor és 5, indica que la longitud és de 20 bytes.
	Reservat	4 bits	Camp reservat inicialitzat amb zeros.
	Control	8 bits	En els missatges enviats a través del <i>socket</i> , estan activats els bits de control PSH i ACK. (0x018).
	Finestra	16 bits	Finestra anunciada per l'emissor.
	<i>Checksum</i>	16 bits	Camp usat per detectar errors.
	<i>Urgent pointer</i>	16 bits	Bit de control URG no actiu.
	Opcions TCP	0 bits	El segment no té opcions TCP addicionals.
Payload	Missatge enviat pel <i>socket</i>	7 bytes	En aquest cas, el missatge enviat pel <i>socket</i> és una dada de temperatura (24.19).

### 4.3.2. Alertes per correu electrònic. Protocol SMTP.

L'enviament dels missatges de correu electrònic des del dispositiu integrat de control, també es realitza a través d'un *socket* TCP que s'estableix entre l'LPC i el servidor SMTP. En aquest cas, s'utilitza el protocol SMTP a la capa d'aplicació per a l'intercanvi de dades entre els extrems.

Mitjançant el *socket*, el dispositiu integrat de control envia les comandes necessàries -definides pel protocol SMTP- al servidor, per tal d'enviar un correu electrònic al destinatari que l'usuari hagi configurat.

La funció implementada a la llibreria de comunicació per a l'enviament del correu, s'ha dissenyat assumint que el servidor SMTP requereix autenticació (SMTP AUTH) i, per tant, requereix de l'enviament dels paràmetres d'autenticació codificats en base64.

Els missatges intercanviats entre l'LPC i el servidor SMTP a través del *socket* són els següents:

```
EHLO
250-vl-2546.vps.dns-servicios.com
250-AUTH=LOGIN CRAM-MD5 PLAIN
250-AUTH LOGIN CRAM-MD5 PLAIN
250-STARTTLS
250-PIPELINING
250 8BITMIME

AUTH LOGIN
334 VXNlcm5hbWU6 (Missatge "Username:" codificat en Base64)
c2Vuc29yQHRmYy5zZXJnaW9nb21lei5lcw== (Nom d'usuari codificat en Base64)
334 UGFzc3dvcmQ6 (Missatge "Password:" codificat en Base64)
bHBjMTc2OQ== (Contrasenya codificada en Base64)
235 go ahead

MAIL FROM:sensor@tfc.sergiogomez.es
250 ok

RCPT TO:mail@sergiogomez.es
250 ok

DATA
354 go ahead

Subject: Alarma de temperatura
La temperatura esta fora dels marges de seguretat. Temperatura obtinguda: XX graus.
.
250 ok 1385038188 qp 12260

quit
221 vl-2546.vps.dns-servicios.com
```

Figura 37: Detall de la comunicació amb el servidor SMTP

## 4.4. Aplicació servidor i interfície d'usuari

En aquest apartat es descriu de manera detallada el funcionament de l'aplicació formada pel servidor TCP, la base de dades i la interfície d'usuari basada en web.

### 4.4.1. Diagrama de blocs

El diagrama de blocs detallat de l'aplicació és el següent:

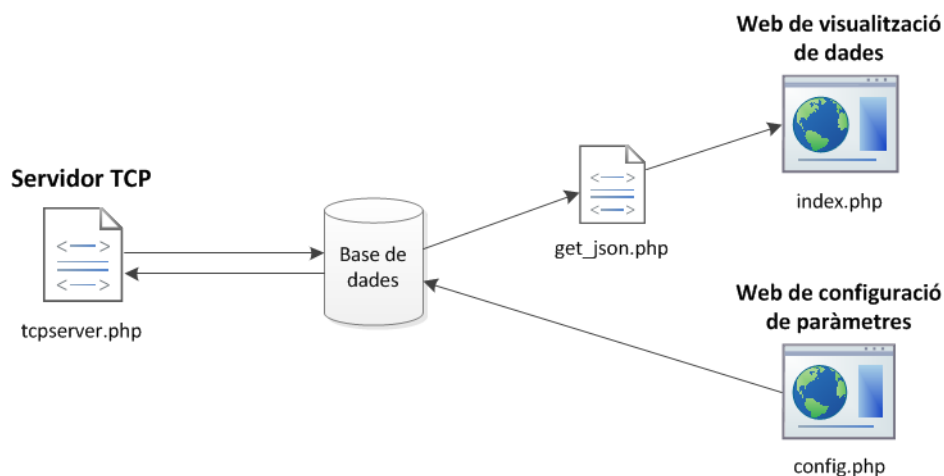


Figura 38: Diagrama de blocs detallat de l'aplicació servidor i interfície d'usuari

### 4.4.2. Servidor TCP

Tal com s'ha explicat en el capítol anterior, el servidor TCP s'ha desenvolupat en PHP, que al ser un llenguatge multi-plataforma permet l'execució de l'aplicació sota la majoria de sistemes operatius.

Una vegada s'inicialitza l'aplicació servidor, aquesta es queda en mode *listen*, esperant peticions de connexió a través del port TCP que s'hagi configurat (per defecte, el 6001). A l'arribar una petició de connexió, l'aplicació crea el *socket* i envia el missatge "\*\*\*HELLO\*\*" a l'altre extrem a través del *socket* creat.

En aquest punt, el servidor queda a l'espera de les peticions que li arribin de l'altre extrem. Les dades rebudes pel *socket* poden ser de tres tipus:

- **Comanda sol·licitant un paràmetre de configuració:** el servidor farà una consulta a la base de dades per tal d'obtenir el valor del paràmetre demanat i enviarà un segment TCP amb aquest valor a l'altre extrem a través del *socket*.

- **Dada de temperatura:** el servidor guardarà la dada de temperatura rebuda, junt amb la data i hora de recepció, a la base de dades. Per confirmar la correcta recepció, enviarà un segment TCP amb el missatge “\*DATAOK\*” a l'altre extrem a través del *socket*.
- **Comanda d'actuació sobre el servidor:** en cas de rebre la comanda “QUIT”, el servidor tancarà el *socket* i tornarà al mode *listen*, esperant noves peticions de connexió a través del port TCP configurat.

En la següent taula es descriuen les comandes que accepta el servidor TCP:

Comandes que accepta el servidor TCP	
<b>GETTEMPMAX</b>	Retorna la temperatura màxima de seguretat, en graus.
<b>GETTEMPMIN</b>	Retorna la temperatura mínima de seguretat, en graus.
<b>GETTEMPINTERVAL</b>	Retorna l'interval de temps, en segons, en el que el dispositiu integrat de control obté una nova mesura de temperatura.
<b>GETMAILSERVER</b>	Retorna el nom del host del servidor SMTP.
<b>GETMAILPORT</b>	Retorna el port del servidor SMTP.
<b>GETMAILUSER</b>	Retorna el nom d'usuari per a l'autenticació amb el servidor.
<b>GETMAILPASS</b>	Retorna la contrasenya per a l'autenticació amb el servidor.
<b>GETMAILFROM</b>	Retorna l'adreça de correu que figurarà com a remitent de l'enviament.
<b>GETMAILTO</b>	Retorna el correu electrònic on s'enviaran els missatges d'alarma.
<b>GETSOUNDALARM</b>	Retorna la configuració de la funcionalitat de l'alarma sonora. Retorna 0 en cas d'estar desactivada i 1 en cas d'estar activada.
<b>GETLASTUPDATE</b>	Retorna el valor <i>timestamp</i> de la darrera actualització de la configuració per part de l'usuari.
<b>QUIT</b>	Tanca el <i>socket</i> i torna al mode <i>listen</i> .
<b>SHUTDOWN</b>	Apaga el servidor

Figura 39: Comandes que accepta el servidor TCP

A continuació es mostra una captura de pantalla del servidor, en mode consola, amb els missatges rebuts a través del *socket*:

```

root@vl-2546:~
**** SERVIDOR PHP ****
IP: tfc.sergiogomez.es
Port: 6001
*****
Escoltant peticions...
Acceptada connexio de 81.61.145.25
Rebut: GETLASTUPDATE
Finalitzada comunicacio.

Acceptada connexio de 81.61.145.25
Rebut: GETTEMPMAX
Rebut: GETTEMPMIN
Rebut: GETTEMPINTERVAL
Rebut: GETMAILSERVER
Rebut: GETMAILPORT
Rebut: GETMAILUSER
Rebut: GETMAILPASS
Rebut: GETMAILFROM
Rebut: GETMAILTO
Rebut: GETSOUNDALARM
Rebut: GETLASTUPDATE
Finalitzada comunicacio.

Acceptada connexio de 81.61.145.25
Rebut: 24.87
Finalitzada comunicacio.

Acceptada connexio de 81.61.145.25
Rebut: GETLASTUPDATE
Finalitzada comunicacio.

```

Figura 40: Dades rebudes pel servidor a través del socket TCP

#### 4.4.3. Base de dades

La base de dades MySQL està formada per dues taules senzilles:

- **Taula *temperatures*:** S'emmagatzemen les dades de temperatura rebudes pel servidor, junt amb la data i hora de recepció. Les temperatures es guarden un camp de tipus *float* i les dates en un camp de tipus *datetime*.
- **Taula *config*:** S'emmagatzemen els paràmetres de configuració del sistema modificables per l'usuari a través del web de configuració, junt amb el segell de temps de la darrera modificació. La taula té dos camps: *param* i *value*, de tipus *varchar*, on es guarda els paràmetre amb el seu valor corresponent.

L'estructura de les taules és la següent:

Table Name	Field Name	Field Type
tfc.temperatures	time	datetime
	temp	float
tfc.config	param	varchar(255)
	value	varchar(255)

Figura 41: Estructura de les taules de la base de dades



A continuació es mostra el contingut de les dues taules, durant el funcionament del sistema:

time ▲	temp
2013-12-22 15:35:06	23.75
2013-12-22 15:35:28	23.87
2013-12-22 15:36:18	23.87
2013-12-22 15:36:38	23.87
2013-12-22 15:37:16	23.87
2013-12-22 15:37:36	23.87
2013-12-22 15:37:56	24
2013-12-22 15:38:16	24.06
2013-12-22 15:38:36	24
2013-12-22 15:38:56	24
2013-12-22 15:39:16	24

Figura 42: Contingut de la taula *temperatures*

param	value
tempMax	24.50
tempMin	17.50
tempInterval	20
mailServer	mail.tfc.sergiogomez.es
mailPort	25
mailUser	sensor@tfc.sergiogomez.es
mailPass	lpc1769
mailFrom	sensor@tfc.sergiogomez.es
mailTo	mail@sergiogomez.es
soundAlarm	0
lastUpdate	1387826595

Figura 43: Contingut de la taula *config*

#### 4.4.4. Interfície web

La interfície d'usuari està formada per dos webs: el visualització de dades i el de configuració de paràmetres.

##### Web de visualització de dades

El web de visualització mostra les dades de temperatura en temps real, per mitjà d'un gràfic generat a través de l'API de *Google Chart*. El gràfic s'actualitza cada 5 segons, amb les dades proporcionades pel fitxer *get\_json.php*, encarregat d'obtenir el contingut de la taula *temperatures* mitjançant les comandes MySQL pertinents, i de retornar-lo en format JSON cada cop que és cridat.

##### Web de visualització de dades

El web de visualització de dades està format per un formulari que permet modificar els paràmetres de configuració del sistema guardats a la base de dades. Cada cop que l'usuari envia una nova configuració a través del formulari, l'arxiu *config.php* actualitza el valor dels paràmetres de configuració de la taula *config* amb els proporcionats per l'usuari, i actualitza el camp *lastUpdate* amb el valor *timestamp* de la data en que s'ha produït l'actualització.

## 4.5. Esquema de connexions

A continuació es mostra l'esquema de connexions entre la placa LPC1769 i els diferents components utilitzats:

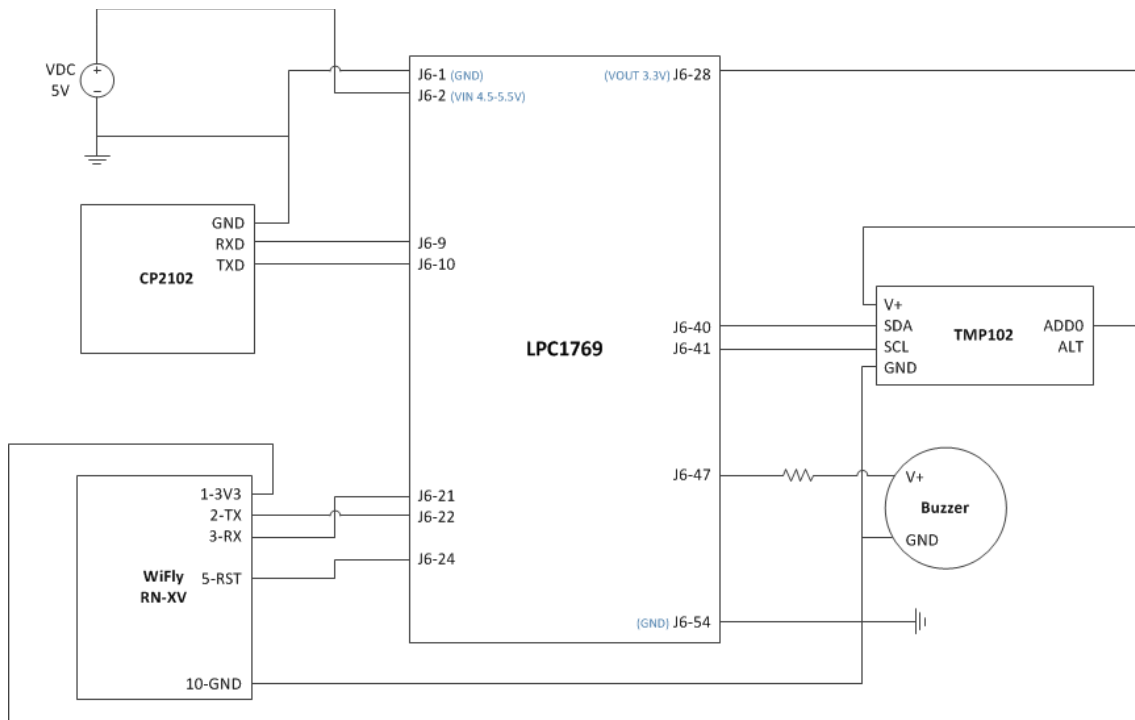


Figura 44: Esquema de connexions del dispositiu integrat de control

## 5. Viabilitat tècnica

El sistema dissenyat ofereix una solució suficientment completa funcionalment, precisa en l'obtenció de mesures i amb un cost en materials raonable, fets que possibiliten que el prototip desenvolupat serveixi de base com a punt de partida per al disseny d'una solució comercial.

A continuació s'analitzen els punts forts i els punts febles de la solució dissenyada:

### Punts forts

- És una solució completa funcionalment: ofereix alarmes basades en correu electrònic i so, el dispositiu està dotat de connexió Wi-Fi, l'usuari pot configurar el sistema i visualitzar dades obtingudes fàcilment des d'un dispositiu amb navegador web, incorpora un sistema de *log* per a la detecció d'errors...
- Donat el disseny modular de l'aplicació desenvolupada i les possibilitats de l'LPC, es podrien incorporar nous sensors de manera relativament senzilla.
- L'aplicació servidor pot ser instal·lada en un servidor privat de l'usuari sense cap cost en llicències de programari, o pot ser oferida com a servei basat en web seguint el paradigma *Internet Of Things*.

### Punts febles

- Per modificar els paràmetres de configuració de la connexió Wi-Fi (SSID, contrasenya i tipus d'autenticació) i del servidor TCP (adreça i port) cal fer un *flash* a la placa, donat que actualment aquests valors no són modificables per l'usuari.
- L'aplicació servidor desenvolupada només pot atendre peticions d'un dispositiu al mateix temps.
- El dispositiu integrat de control no està optimitzat per al seu funcionament amb bateries.

## 6. Valoració econòmica

En aquest apartat es realitza una valoració econòmica del projecte, detallant els costos associats als materials necessaris per a la construcció de la solució dissenyada, així com les hores de desenvolupament emprades.

### Despeses de material i desenvolupament:

Concepte	Unitats	Import Unitari	Import Total
Placa LPCXpresso LPC1769	1	20,00 €	20,00 €
WiFly RN-XV	1	31,95 €	31,95 €
Convertidor UART-USB CP2102	1	17,25 €	17,25 €
Adaptador DIP per a XBee	1	2,50 €	2,50 €
Sensor de temperatura TMP102	1	5,00 €	5,00 €
Brunzidor Velleman SV3	1	1,93 €	1,93 €
Transformador 220VAC-USB Samsung	1	18,95 €	18,95 €
Connector Barrel	1	0,65 €	0,65 €
Cable Barrel-USB 5.5mm	1	1,95 €	1,95 €
Pins sense soldadura (Pack 10 Unitats)	1	0,15 €	0,15 €
Set de cables M/F (Pack 10 Unitats)	1	2,50 €	2,50 €
Set de Cables M/M (Pack 10 Unitats)	1	2,50 €	2,50 €
Placa de prototipat 16x50cm	2	9,00 €	18,00 €
Hores de desenvolupament Fases 1, 2 i 3	150	30,00 €	4.500,00 €
<b>Despeses d'enviament de material</b>			<b>14,95 €</b>
		Subtotal	4.638,28 €
		IVA (21%)	974,04 €
		<b>TOTAL</b>	<b>5.612,32 €</b>

Els costos del material necessari per a la construcció del dispositiu integrat de control sumen **138,20 € + IVA**. Pel que fa a les hores de desenvolupament, s'ha fet una estimació en hores i cost, que ascendeix a 4.500 € + IVA.

D'altra banda, per al funcionament de l'aplicació servidor a Internet, s'ha contractat un Servidor Virtual Privat amb CentOS de característiques bàsiques (1GB de RAM, 25GB d'espai en disc i 1000GB de transferència), amb un **cost mensual de 9,30 € + IVA**.

## 7. Conclusions

En aquest apartat s'exposen les conclusions del projecte un cop finalitzat, es presenta una proposta de futures millores de la solució dissenyada i es realitza una autoavaluació dels coneixements adquirits.

### 7.1. Conclusions del projecte

La solució dissenyada i desenvolupada compleix amb tots els objectius que es van definir a l'inici d'aquest projecte. Per tant, el *sistema d'alarma i monitorització ambiental amb gestió remota per a entorns crítics* desenvolupat, és capaç de realitzar totes les funcionalitats inicialment previstes de manera satisfactòria.

La realització d'una planificació acurada i detallada, la divisió del treball a realitzar en tasques clarament definides i delimitades, la fase d'aprenentatge inicial i la metodologia de desenvolupament seguida basada en fites, han estat fonamentals per poder assolir els objectius plantejats en el temps previst.

### 7.2. Proposta de millores

Tot i que el sistema desenvolupat compleix amb els objectius plantejats, aquest disposa de diversos punts febles, tal com s'ha comentat anteriorment. Per tant, la proposta de millores que es realitza a continuació, contempla la correcció de les debilitats detectades, així com la incorporació de noves funcionalitats:

- Incorporar una funcionalitat que permeti a l'usuari final modificar els paràmetres de configuració de la connexió Wi-Fi (SSID, contrasenya i tipus d'autenticació) i del servidor TCP (adreça i port), accedint al dispositiu a través d'una aplicació terminal, mitjançant un ordinador connectat al CP2102.
- Optimitzar el consum del dispositiu per al seu funcionament amb bateries, fent servir els diferents modes de funcionament que el processador ofereix: *Sleep*, *Deep Sleep* i *Power-down*.
- Implementar un sistema de reinici del dispositiu complet en cas de bloqueig, fent servir la funcionalitat *Watchdog*.
- Possibilitar que el nivell de detall dels missatges del sistema de *log* sigui configurable per l'usuari, des de l'entorn web de configuració.

### 7.3. Autoavaluació

L'elecció de l'àrea de Sistemes Encastats per realitzar el meu treball de final de carrera, va estar motivada totalment per una inquietud personal. Vaig considerar que desenvolupar el TFC dins d'aquest àrea, era l'oportunitat ideal per introduir-me en un àmbit de coneixement que em cridava especialment l'atenció i de poder desenvolupar algun projecte útil i interessant.

Avui, quatre mesos després de l'inici de l'assignatura, puc dir que estic plenament satisfet amb l'elecció que vaig prendre en el seu dia. Partia d'uns coneixements nuls en el món de la programació de sistemes encastats i, amb els coneixements adquirits durant el curs, s'ha pogut desenvolupar un prototip totalment funcional, que compleix amb tots els objectius previstos, i que dóna solució a una necessitat existent.

Ha estat interessant veure com, a mida que s'anava avançant en el desenvolupament del projecte i s'anaven adquirint nous coneixements, també creixien el nombre d'idees que anaven sorgint per a implementar en el projecte. En aquest sentit, donada la limitació de temps de què es disposava, en moltes ocasions ha estat difícil cenyir-se a la planificació prevista i deixar les noves idees que anaven sorgint com a millores per implementar en un futur.

En definitiva, considero que ha estat un projecte molt enriquidor i que m'ha portat uns coneixements que serviran com a base per al desenvolupament de nous projectes dins d'aquest àmbit.

## 8. Glossari

**HTTP** *Hypertext Transfer Protocol*. Protocol de la capa d'aplicació utilitzat per a les transaccions WWW.

**I2C** *Inter-Integrated Circuit*. Bus de comunicacions en sèrie dissenyat per Philips semiconductors (actualment NXP) que utilitza dues línies per a la transmissió de la informació: una per a les dades i una altra per al senyal de rellotge.

**Idle Mode** en què el processador està inactiu.

**JSON** *JavaScript Object Notation*. Format lleuger per a l'intercanvi de dades.

**JTAG** *Joint Test Action Group*. Terme per designar la norma IEEE 1149.1, per al testeig de circuits electrònics.

**LAMP** Acrònim utilitzat per a descriure un sistema d'infraestructura que utilitza Linux com a sistema operatiu, Apache com a servidor web, MySQL com a gestor de bases de dades i PHP, Perl o Python com a llenguatges de programació.

**Log** Registre d'esdeveniments què succeeixen en una aplicació quan està funcionant.

**Model client-servidor** Model per a representar aplicacions locals i distribuïdes en què es defineix una part servidora (que és qui proporciona els serveis) a la qual accedeix una part client (que és qui fa les peticions de servei).

**MSS** Defineix la longitud màxima de dades que el TCP enviarà. S'especifica durant l'establiment de la connexió.

**Mútex** Indicadors binaris utilitzats usualment per a protegir recursos compartits d'accessos simultanis durant l'execució de seccions críiques de codi.

**PHP** Llenguatge de programació del costat servidor (*server-side*) dissenyat per a desenvolupament web, encara que també és utilitzat com a llenguatge de programació d'ús general.

**Planificador** Element de programari integrat en el nucli del sistema operatiu responsable de decidir quina tasca ha d'utilitzar el processador en un moment determinat.

**SMTP** *Simple Mail Transfer Protocol*. Protocol de la capa d'aplicació utilitzat per a transferir missatges de correu electrònic. Estàndard d'Internet.

**Socket** Un *socket* o connector és un punt d'accés d'un flux de comunicació entre processos a través d'una xarxa.

**SRAM** Memòria RAM estàtica.

**TCP** *Transmission Control Protocol*. Protocol de transport orientat a la connexió amb control de flux i errors, que garanteix el lliurament de tota la informació en el mateix ordre en què ha estat transmesa,.

**TWI/TWSI** *Two Wire Interface* o *Two-Wire Serial Interface*. Terme que utilitzen algunes marques per a referir-se a I2C. En alguns casos, l'ús del terme TWI/TWSI indica que la interfície no és totalment compatible amb les especificacions I2C.

**UART** *Universal Asynchronous Receiver-Transmitter*. Component *hardware* encarregat de controlar els ports i dispositius sèrie.

**UDP** *User Datagram Protocol*. Protocol de transport no orientat a la connexió, que no proporciona cap tipus de control d'errors ni de flux, tot i que utilitza mecanismes de detecció d'errors.



## 9. Bibliografia

**Barry, Richard** (2010). *Using the FreeRTOS Real Time Kernel – NXP LCP17xx Edition* (2a edició).

**Gómez Cama, José Maria; Hernández Ramírez, Francisco; López Vicario, José; Morell Pérez Antonio; Prades García, Juan Daniel; Vilajosana Guillén, Ignasi; Vilajosana Guillén, Xavier** (2011). *Sistemes encastats* (1a edició).

**Íñigo Griera Jordi; Barceló Ordinas, José Maria; Lorente Viejo, Silvia; Marquès i Puig, Joan Manuel; Martí Escalé, Ramón; Peig Olivé, Enric; Perramon Tornil, Xavier** (2007). *Protocols i aplicacions Internet* (2a edició).

### Recursos Web:

- FreeRTOS  
<http://www.FreeRTOS.org>
- LPCWare Forums  
<http://www.lpcware.com/forum>
- Sparkfun  
<https://www.sparkfun.com/>
- Wiki Embedded Systems Lab (UOC)  
<http://cv.uoc.edu/webapps/xwiki/wiki/matembeddedsystems slabhome/view/Material/IniciCortexM3>

### Datasheets dels components utilitzats:

- *LPC17xx User manual. UM10360.*  
[http://www.nxp.com/documents/user\\_manual/UM10360.pdf](http://www.nxp.com/documents/user_manual/UM10360.pdf)
- TMP102 de Texas Instruments  
<http://www.sparkfun.com/datasheets/Sensors/Temperature/tmp102.pdf>
- WiFly RN-XV de Roving Networks  
<http://wsn.uoc.edu/xvilajosana/arpa/WiFly-RN-XV-DS.pdf>

## 10. Annexos

### 10.1. Execució, compilació i càrrega de l'aplicació encastada

#### Instal·lació del IDE LPCXpresso

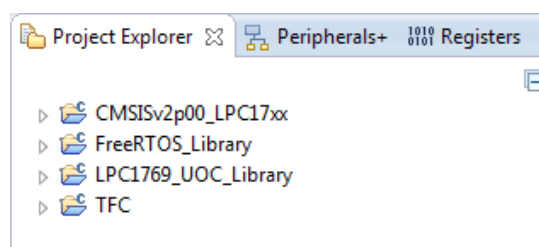
En primer lloc, cal descarregar i instal·lar l'IDE LPCXpresso. L'aplicació i la guia per a realitzar la instal·lació es poden trobar en: <http://www.lpcware.com/lpcxpresso/download>

#### Importació del projecte

Un cop instal·lat l'IDE, cal importar el projecte a l'entorn de desenvolupament. Una manera senzilla de fer-ho és la següent:

- Crear una carpeta *workspace* en el sistema. Normalment ja és crea quan s'inicia l'LPCXpresso.
- Copiar el contingut de la carpeta "*workspace*" proporcionada, en la carpeta acabada de crear.
- Obrir l'LPCXpresso.
- Anar a *File* → *Import*
  - *General* → *Existing projects into Workspace*
    - En la finestra "*Import*" seleccionar la carpeta corresponent.

Un cop realitzada la importació, l'estructura de carpetes del projecte hauria de ser la següent:



#### Configuració de l'aplicació



Abans de realitzar la càrrega de l'aplicació a l'LPC, cal configurar els paràmetres de la xarxa Wi-Fi a la que es connectarà el dispositiu i les dades del servidor TCP, modificant les següents línies del fitxer *main.c*:

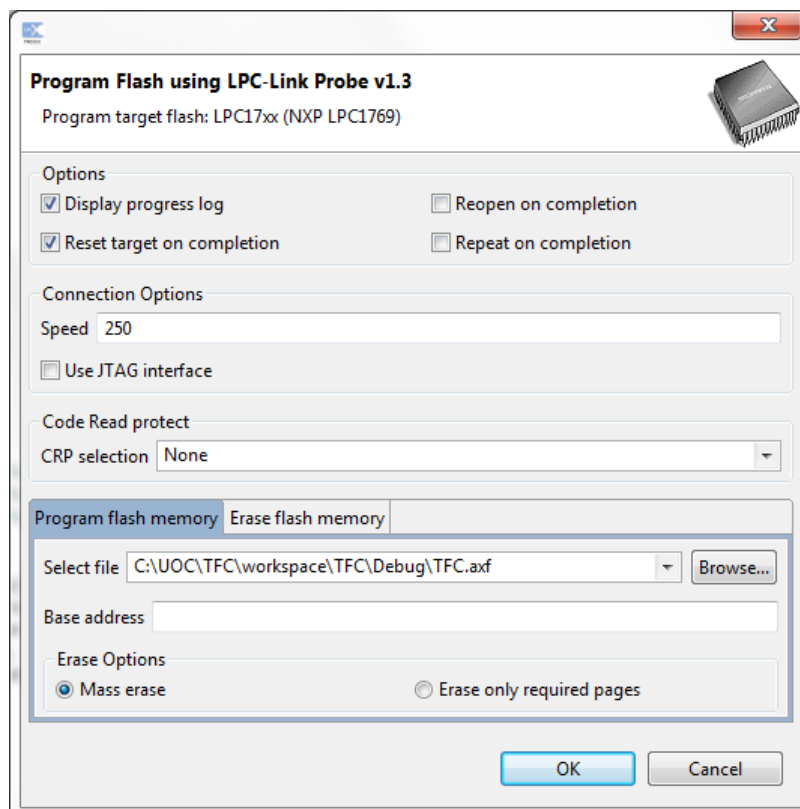
```
/* Dades configuració AP */
static char* WLAN_SSID = "";           //SSID de la xarxa
static char* WLAN_PWD = "";           //Contrasenya de la xarxa
static uint8_t WLAN_AUTH = WPA;       //Tipus d'autenticació de la xarxa: OPEN / WEP / WPA

/*Dades Servidor TCP*/
static char *ServerHost = "";         //Host Name or IP
static int ServerPort = 6001;         //Port
static uint8_t HostType = HOSTDNS;    //HOSTDNS for Name or HOSTIP for IP
```

## Compilació i càrrega

Per tal de realitzar la compilació i càrrega de l'aplicació a l'LPC cal seguir els següents passos:

- Connectar la placa LPC a l'ordinador a través d'USB.
- Obrir la carpeta TFC del projecte
- Realitzar la compilació, fent clic a la icona 
- Realitzar el flash del programa a la placa, fent clic a la icona 
- Un cop a la finestra "Program Flash" només cal seleccionar el fitxer .axf corresponent i fer clic al botó OK, per realitzar el flash a la placa.



## 10.2. Instal·lació i configuració de l'aplicació servidor

Els requisits de programari per a la instal·lació i execució de l'aplicació servidor són: intèrpret PHP, servidor web Apache (o un altre que suporti PHP) i base de dades MySQL.

### Servidor TCP

En primer lloc, cal indicar les dades del servidor on funcionarà l'aplicació, modificant les següents línies del fitxer **tcpserver.php**:

```
$address = 'tfc.sergiogomez.es'; //Host o IP del servidor  
$port = 6001; //Port d'escolta del servidor
```

És important tenir present que s'ha d'especificar **el host o la IP** que utilitzarà el servidor per escoltar les connexions. Així, si especifiquem un host, el servidor no atendrà les peticions realitzades pel client si aquest es connecta per IP.

Per tal de posar en funcionament el servidor, només cal executar-lo amb la comanda corresponent segons el sistema operatiu utilitzat:

- Execució del servidor en entorns Linux: ***/usr/bin/php -q /ruta/tcpserver.php***
- Execució del servidor en entorns Windows: ***php.exe -q c:\ruta\tcpserver.php***

### **Base de dades**

L'arxiu ***DB/tfc.sql*** proporcionat, inclou les sentències necessàries per crear la base de dades i l'estructura de taules, així com també conté una configuració base del sistema.

La importació es pot realitzar mitjançant la següent comanda:

- Importació en entorns Linux: ***mysql -u usuari -p < ruta/tfc.sql***
- Importació entorns Windows: ***mysql.exe -u usuari -p < c:\ruta\tfc.sql***

Finalment, caldrà indicar al servidor les dades d'autenticació de la base de dades modificant el fitxer ***mysql.php***.

### **Aplicació web**

Per tal d'accedir a l'entorn web, només cal cridar al fitxer ***index.php*** des d'un navegador.