# WSN aplicadas a la gestión inteligente de la calefacción.

**Joan Sierra Moreno**

"Enginyeria Tècnica de Telecomunicació, especialitat en Telemàtica"

**Consultor**

**Jordi Becares**

Data Lliurament

11.06.2013

*Seis años de resistencia y apoyo continuo.*

*Sin ti no lo habría conseguido.*

*Miriam*

*Dos luceros en el horizonte.*

*Víctor y Aina*

# Agradecimientos

*En especial quisiera agradecer a toda mi familia que me ha apoyado incondicionalmente durante estos años intensos.*

*A los compañeros de viaje que hemos compartido conocimiento y sufrimientos.*

*A los tutores que se han esforzado en ayudarnos a pesar de nuestra torpeza en algunos momentos.*

*A los que creen que compartir no es robar y que el mundo no se acabará compartiendo.*

*Y sobre todo a la red que extiende sus brazos desinteresadamente en muchos casos para beneficio de todos y en otros en benefecio de muchos.*

# Resumen

El presente proyecto final de carrera (TFC) tiene como objetivo poner en práctica parte de los conocimientos adquiridos durante el estudio de las diferentes disciplinas que forman parte de la ***"Enginyeria Tècnica de Telecomunicació, especialitat en Telemàtica".***

El proyecto tiene su ámbito dentro de los Sistemas Embebidos e intenta crear una aplicación válida tomando provecho de los conocimientos adquiridos.

El proyecto desarrollado consiste en un Sistema Embebido que estará formado por una red de sensores con la finalidad de dar un uso más racional y eficiente al sistema de calefacción de una casa aislada de tres plantas, gestionando directamente la activación de la caldera y de los circuitos de agua de calefacción para cada una de las plantas de forma individualizada.

# Índice de Contenidos

# Índice de Figuras

# 1. Introducción

## 1.1. Justificación

El presente proyecto final de carrera nace de la necesidad de optimizar el uso de la calefacción en una casa aislada de tres plantas. En la actualidad el sistema de calefacción esta dotado de tres circuitos independientes de agua de calefacción, uno por planta, equipados de un sistema de apertura/cierre manual y, un único sensor de temperatura ubicado en la planta intermedia.

La propuesta del sistema a desarrollar consiste en la instalación de tres sensores de temperatura, uno por planta, que enviarán la información en tiempo real a un servidor WEB. Una unidad central leerá los valores de temperatura de las estancias así como las temperaturas de consigna de uso de la calefacción del servidor WEB. La unidad central se encargará de procesar la información y abrir o cerrar el circuito de calefacción pertinente mediante la activación de un relé que será el encargado de accionar las electroválvulas que abren o cierran el circuito de agua de calefacción correspondiente. Adicionalmente, la unidad central comandará la señal para encender o apagar la caldera.

## 1.2. Descripción

El sistema estará formado por una red de sensores inalámbricos (WSN, Wireless Sensor Network) compuesto cada uno de ellos por una placa LPCXpresso1769, un módulo WiFly RN-171, un sensor de temperatura y una batería. De esta manera y con el encapsulado correcto los sensores serán totalmente movibles y autónomos.

La red WSN enviará los datos a traveś de un punto de acceso (AP) hasta un servidor WEB utilizando comandos HTTP encapsulados bajo los protocolos de comunicación TCP/IP estandard.

El servidor WEB realizará adicionalmente las funciones de módulo de usuario a través de un portal WEB que permitirá monitorizar los datos en tiempo real así como programar diferentes temperaturas de consigna según la franja horaria para cada una de las plantas de la vivienda independientemente.

El servidor WEB queda fuera del ámbito del presente TFC. Las funciones de visualización de la temperatura serán simuladas con una versión modificada de la aplicación arp@Lab ubicada en un servidor de Google Apps y accesible públicamente en la URL http://tfcjoansierra.appspot.com Las funciones de programación de temperaturas de consigna y franjas horarias serán simuladas con la aplicación modificada, la que dispondrá de dos versiones diferentes con dos temperaturas de consigna diferentes que nos servirán para comprobar el correcto funcionamiento de la aplicación creada para el sistema encastado dentro de lo que denominaremos Unidad Centra de Control (UCC).

La UCC se encargará de recibir los valores monitorizados de los diferentes sensores así como las temperaturas de consigna. Con esta información, la UCC tendrá la función de interpretar los datos recibidos y activar/desactivar la calderá así como los diferentes actuadores que comandarán las electroválvulas para cada uno de los circuitos de agua de calefacción de cada planta.

Con esta definición, el ámbito del proyecto queda limitado a la **toma de contacto con los sistemas encastados , la programación de sensores y el control de dispositivos**

adicionales, en concreto la red de sensores de temperatura y el control de la caldera y las electroválvulas que están accionadas por los actuadores o rele.

Una vez sentadas éstas bases, la ampliación de la red de sensores con otro tipo de detectores como sensores de humedad, movimiento, gas, humo, motores de persianas, etc, no tiene que representar ninguna complicación adicional. De esta manera se consigue iniciar una aproximación al entorno de los sistemas encastados en el ámbito de la domótica.

En el siguiente diagrama de bloques (figura1) se puede apreciar el sistema al completo de una manera simplificada Tres sensores de temperatura, uno por cada planta, que se conectan a un servidor WEB a través de internet, que a su vez hace las funciones de interfaz de usuario. El corazón del sistema está formado por la UCC que recibe la información del servidor y decide si actúa o no sobre la caldera y las electroválvulas

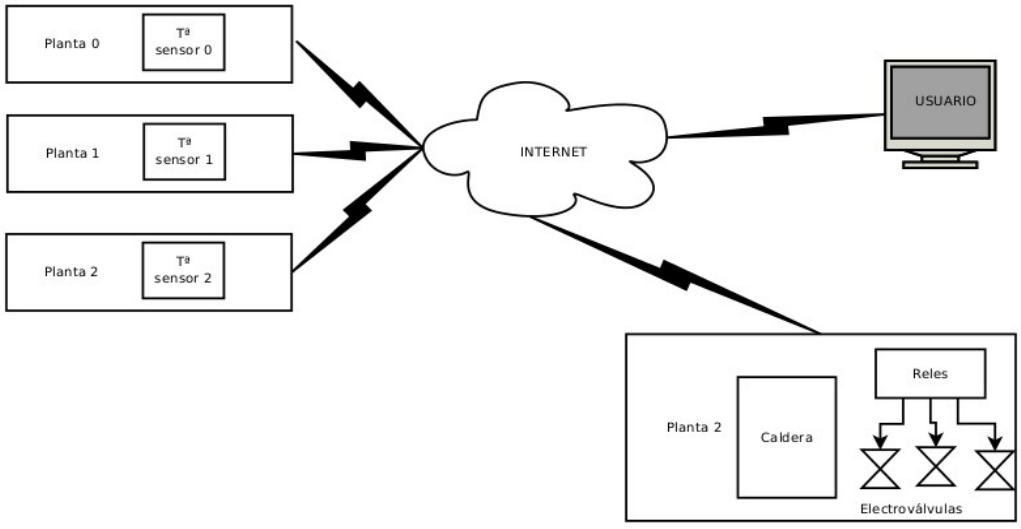


Figura 1

## 1.3 Objetivos

Los diferentes objetivos del sistema tienen como misión lograr la independización del sistema de calefacción para cada una de las tres plantas de la casa.

- Lectura del valor de temperatura ambiente en cada planta.
- Muestreo del valor de la temperatura en cada planta con una frecuencia determinada.
- Envío de las lecturas muestreadas en tiempo real a un servidor.
- Visualización de los valores registrados en tiempo real a través de la aplicación Arp@Lab modificada.
- Registrar temperatura de consigna según franja horaria para cada planta en la UCC.
- Evaluación de la temperatura en tiempo real respecto a la temperatura de consigna para la franja horaria en curso..
- Activación/Desactivación de la caldera.
- Accionamiento del relé asociado a la electroválvula que necesitamos abrir/cerrar según la planta a controlar.

## 1.4. Enfoque y método

El proyecto está desarrollado con un enfoque plenamente didáctico. Para la realización con éxito de los objetivos propuestos ha sido necesario un periodo de aprendizaje que se ha llevado a cabo con el seguimiento de las 'Proves d'Avaluació Continua' o PAC's.

Las tres primeras PAC's han servido para familiarizarme con el Entorno de Desarrollo Integrado (IDE), basado en ECLIPSE, el hardware compuesto por la placa LPC1769, el módulo WiFly y el conversor Serie-USB, además de crear una pequeña aplicación que interaccionase con una aplicación WEB creada por la UOC llamada Arp@Lab.

El resto del proyecto ha seguido un método mucho más experimental y productivo hasta seleccionar y encontrar el material necesario para el desarrollo definitivo.

A tener en cuenta especialmente es la utilización del concepto Open Source o Código Abierto. El proyecto se presenta en formato Open Source y se beneficia también de otros códigos de otros autores bien como librerías o bien como fuente de estudio para conseguir los objetivos fijados.

## 1.5 Planificación

El proyecto de ha dividido en tres fases bien definidas: aprendizaje, definición y desarrollo.

Durante la fase de aprendizaje se ha tomado contacto con el entorno de trabajo así como de los dispositivos necesarios. Durante esta fase he sido capaz de hacerme una idea de las capacidades y oportunidades que nos ofrece el equipo disponible.

Gracias a esta fase de aprendizaje y, conociendo las limitaciones en cuanto a tiempo y recursos disponibles, he podido definir un proyecto atractivo y a su vez con valor añadido.

Una vez el proyecto definido, ha llegado la hora de ponerlo a la práctica durante el desarrollo. Adjunto diagrama de Gantt (figura 2) del proyecto que a su vez he dividido en pequeñas tareas para asegurar el exito del trabajo.

Diagrama de Gantt

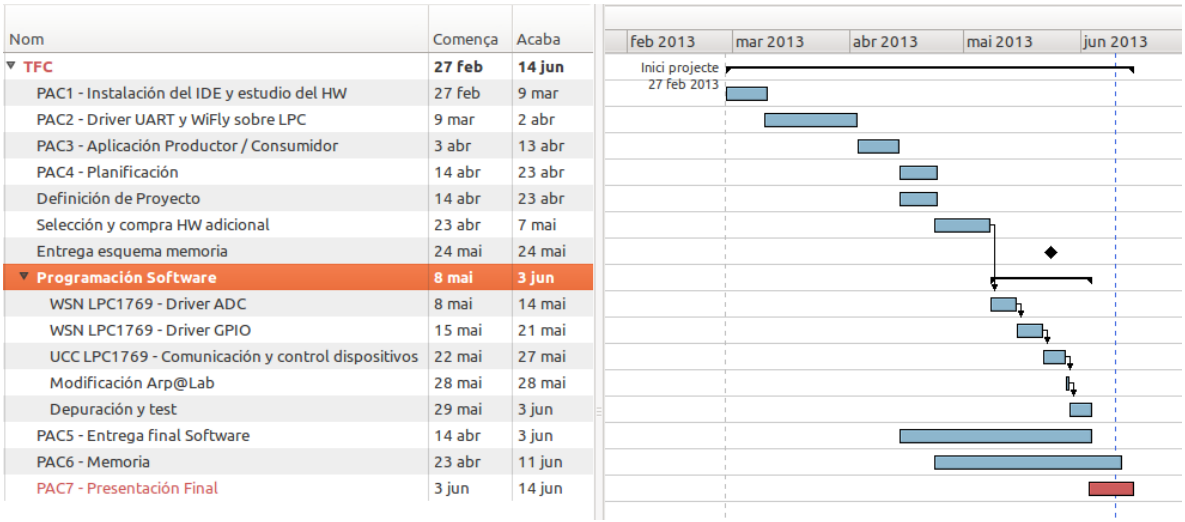


Figura 2

## 1.6 Recursos empleados

Para el desarrollo del proyecto han sido necesario diferentes dispositivos y herramientas informáticas. A continuación pasa a su descripción comenzando por el soporte informático utilizado.

Todo el trabajo se ha desarrollado en un ordenador portátil corriendo bajo un sistema operativo de código abierto. En el caso que nos ocupa la elección ha sido una distribución Linux muy conocida, Ubuntu 12.10 Quantal Quetzal, de la que existe gran

cantidad de recursos e información disponible en internet en caso de que se presenten problemas.

Los programas de soporte utilizados, todos ellos de código abierto ha excepción de Ghrome, han sido:

- **LibreOffice 3.6.2.2:** editor de textos, hoja de cálculo y gestor de presentaciones.
- **The GIMP 2.8**: edición de imágenes.
- **Dia-gnome 0.97.2**: editor de diagramas estructurado.
- **Planner 0.14.6**: gestor de proyectos.
- **Google Chrome 25.0.1364.160 Ubuntu 12.10**: navegador web.
- **Ksnapshot 0.8.2:** capturador de imágenes de pantalla.

Los programas específicos para la creación del proyecto han sido por un lado el Entorno Integral de Desarrollo y un terminal para visualizar los mensajes creados para la depuración del código.

- **LPCXpresso 5.1.2_2065:** Entorno de Desarrollo Integral basado en Eclipse. Utilizado para la programación y grabación del microcontrolador LPC1769.

- **Eclipse Java EE IDE for Web Developers JUNO SR2:** Entorno de Desarrollo Integral con conectores adicionales para creación de aplicaciones WEB en Java y Google App.

- **Cutecom 0.22.0**: terminal de comunicación puerto serie.

Para la programación de la placa LPC1769, ha sido necesario el uso de un sistema operativo en tiempo real: FreeRTOS. Además para su programación ha sido necesario el conocimiento de lenguaje C estandard. Adicionalmente, como lenguaje de programación ha sido necesario Java para poder modificar la aplicación servidor Arp@Lab.

Como componentes complementarios a continuación el listado (figura 3) de todo el material que ha sido necesario en algún momento del desarrollo:

- Placa LPC1769.
- Adaptador USB-Serie CP2102.
- Módulo wifi WiFly RN-171.
- Cable USB.
- Optoacoplador 3.3v a 5v.
- Circuito amplificador de 3.3v a 5v.
- Rele de 5v / 230v
- Placa para construcción de circuitos prototipos (protoboard).
- Cables para protoboard macho-macho y hembra-macho.
- Resistencias fijas.

- Resistencia variable.
- Sensor de temperatura TMP36.
- Leds de luz roja.
- Bombilla 230V con portalámparas.
- Polímetro.
- Soldador y estaño.

Figura 3

## 1.7 Productos obtenidos

El producto obtenido está formado por una placa prototipo con varias funciones integradas y un servidor WEB con un mínimo de funcionalidades que permiten la verificación del funcionamiento del sistema.

La placa prototipo está formada por la placa LPC1769, un sensor de temperatura TMP36, un ḿodulo WiFly RN-171 , un conversor USB-Serie, un optoacoplador conectado a un rele que actuará sobre una bombilla de 230v.

El objetivo de la placa prototipo es el de simular el funcionamiento de todo el sistema descrito con anterioridad. De esta manera podemos evaluar la viabilidad del proyecto antes de crear productos completamente funcionales según la descripción del proyecto.

Las diferentes funciones a verificar con la placa prototipo son:

- **Red de Sensores inalámbricos y envío de datos via WiFi**

Dentro del mismo prototipo podemos simular las funciones de la red de sensores de temperatura y su conexión a una red WiFi a través de un Punto de Acceso (AP) para el envía de datos a un sistema externo, en este caso un servidor WEB.

- **Recepción de datos via WiFi**

También podemos simular la recepción de datos desde un sistema externo lo que nos va a poder permitir en un futuro interactuar con el sistema encastado desde el exterior, aumentando considerablemente el valor añadido y funcionalidad del sistema.

- **Actuación sobre dispositivos externos**

Otro de los módulos que nos permite verificar la placa prototipo es la actuación sobre dispositivos externos. En este caso ha sido necesario realizar una adaptación del voltaje de salida de la placa LPC1769 de 3.3v a 5v para poder actuar sobre el rele que necesita de esta diferencia de potencial para activar el conmutador del circuito de alta de 230v.

- **Visualización de la información a través de una aplicación WEB**

El servidor WEB consiste en la adapación de una aplicación existente facilitada por la UOC: Arp@Lab, que puede encontrarse en la siguiente URL: http://arpalab.appspot.com. Esta aplicación tiene como funcionalidad monitorizar los diferentes parámetros de una red de sensores inalámbricos.

La adaptación realizada de la aplicación ha sido ubicada en los servidores de Google App y es de acceso público en la URL: http://tfcjoansierra.appspot.com.

La modificación ha consistido en presentar conjuntamente con el gráfico que monitoriza los parámetros de la red de sensores, unos campos de texto que nos muestran en todo momento **el último valor leído de temperatura,** la **temperatura de consigna** para ese instante, y el **estado de la calefacción y la electroválvula (ON / OFF)**.

Otro de los cambios introducidos en la aplicación original, ha sido el retorno del valor de consigna tras una consulta GET HTTP a la siguiente URL: http://tfcjoansierra.appspot.com/rest/feed. Anteriormente esta consulta daba como retorno un valor numérico aleatorio. Con esta pequeña adaptación se puede variar el valor de consigna de la calefacción desde fuera del sistema embebido.

Se han creado dos verisones distintas de la aplicación. La versión 2 con temperatura de consigna de 35ºC y, la versión 3 con temperatura de consigna de 20ºC. La versión que se encuentra 'on line' sólo puede cambiarse desde el panel de administración de Google App al cual tan sólo yo tengo acceso. Suficiente para demostrar el correcto funcionamiento de la UCC y poder visualizar la activación y/o desactivación de los dispositivos externos conectados.

La modificación de la aplicación del servidor no incluye ningún parámetro que permita al usuario interactuar directamente con el sistema embebido de forma directa, aunque este es el siguiente paso del desarrollo fuera del ámbito del presente Proyecto Final de Carrera (TFC).

## 1.8 Descripción de capítulos

En el siguiente capítulo haré una pequeña descripción de lo que entendemos por sistema encastado o embebido. Analizaré la situación actual del mercado así como las diferentes alternativas a la escogida para este proyecto.

También analizaré las diferentes soluciones dómoticas que existen en el mercado para la gerstión de la calefacción de una manera inteligente y por tanto eficiente.

En el capítulo 3 realizaré una descripción funcional de todo el sistema tal y como quedará tras la finalización del proyecto.

El capítulo 4 nos permitirá entender con todo detalle y profundidas las soluciones técnicas adoptadas así como su justificación e implementación.

La viabilidad final del proyecto será abordada en el capitulo 5 para posteriormente, en el capítulo 6 poder realizar una valoración económica estimada del proyecto en su fase prototipo así como en una posible comercialización en serie.

Las conclusiónes y la autoevaluación serán abordadas en el capítulo 7 que dará paso a la bibliografía y recursos utilizados en el capítulo 8.

# 2. Antecedentes

## 2.1 Estado del arte

La definición exacta de 'Sistema Embebido' es bastante compleja debido a la flexibilidad que presentan éstos dispositivos. Consultanto la Wikipedia obtenemos la siguiente definición: *"un **sistema embebido** o empotrado es un sistema de computación diseñado para realizar una o algunas pocas funciones dedicadas frecuentemente en un sistema de computación en tiempo real."*.

Si analizamos la definición podemos encontrar los términos claves que nos permiten definir con garantías un sistema embebido. Las tres expresiones claves que yo destacaría son: **sistema de computación, funciones dedicadas y tiempo real.**

A diferencia de los sistemas de computación que estamos habituados a utilizar, como podría ser el ordenador portátil que utilizo para redactar este texto, que está diseñado con un proposito más general, los sistemas embebidos, aún siendo sistemas de computación, están desarrollados para realizar unas tareas muy específicas que requieren una respuesta en tiempo real habitualmente.

Están formados por una CPU, memoria, buses de datos y puertos de entrada y salida en su versión más simplificada. En la actualidad los sistemas embebidos incorporan dentro del mismo sistema dispositivos como conversores ADC, DAC, controladores Ethernet, PWM y una gran cantidad de diferentes protocolos de comunicación como UART, I2C, SSP, CAN, PWM que permiten abrirse al mundo exterior al sistema para interaccionar con un numeroso conjunto de dispositivos como sensores, redes, motores, iluminación, sonido, camaras y un largo etcétera que día a día se va ampliando.

Las características del sistema embebido vienen marcadas principalmente por la CPU integrada y su velocidad de funcionamiento. La CPU nos marcará que instrucciones tendremos que utilizar y su velocidad nos permitirá decidir para que proyectos serán viables el uso de determinados sistemas embebidos.

A continuación en la figura 4, podemos observar la comparativa de las principales características del sistema utilizado frente a uno de los sistemas embebidos más utilizado a nivel aficionado en el mercado actual.

| Caraterística | LPC1769 | Arduino Due |
|---|---|---|
| **CPU** | Cortex M3 32 bits | ARM AT91SAM3X8E 32 bits |
| **Frecuencia reloj** | 120 MHz | 84 MHz |
| **Memoria RAM** | 64 Kb | 96 Kb |
| **Memoria Flash** | 512 Kb | 512 Kb |
| **Voltaje alimentación** | 3.3 V | 3.3 V |
| **ADC** | 8 canales | 12 |
| **GPIO** | 70 | 54 |
| **USB** | SI | SI |
| **JTAG** | SI | SI |

Figura 4

La oferta disponible de microcontroladores en el mercado es extensa y no es el objeto de este TFC justificar la elección de un sistema frente a otro.

Una de las características adicionales del sistema utilizado para el proyecto es que es posible utilizar un sistema operativo y librerías especificas para el microcontrolador de código abierto y, a la vez es posible crear aplicaciones propietarias dentro del mismo sistema.

El sistema utilizado obtiene provecho de que práctimente en todos los hogares o industrias disponemos en la actualidad de una infraestructura wifi que nos permite el acceso a internet y consiguientemente al servidor utilizado como interfaz de usuario. Pero también habría sido del todo viable la utilización de una conexión 3G.

Un ejemplo práctico del beneficio de la tecnología 3G respecto a la WIFI sería la creación de una pequeña estación meteorológica que monitorizase las horas de insolación así como la lluvia caida en un terreno agrícola aislado sin acceso a coberura WIFI, para activar el sistema de riego en los campos solamente en los casos necesarios de alta temperatura y en condiciones de no lluvia. La utilización de la cobertura 3G permitiría el acceso a los datos de forma remota sin necesidad de disponer un acceso a una línea ADSL y permitiría la activación o desactivación del sistema de riego a distancia. Este sistema debería estar apoyado de un pequeño sistema de placa solar mas batería para convertirlo en un sistema totalmente autónomo.

Los precios de las tarifas de datos y la cobertura 3G en la actualidad, hacen viables sistemas embebidos en lugares remotos de una manera que hasta hace muy poco tiempo no eran imaginables para la pequeña empresa o el autónomo a unos costes asequibles.

## 2.2 Estudio de mercado

Las soluciones al control de sistemas de calefacción a distancia encontradas en el mercado actual son amplias y con diferentes niveles de funcionalidades. A continuación enumeraré y haré una pequeña descripción de la solución más sencilla encontrada así como la más compleja en cuanto a funcionalidades que aporta.

El producto más sencillo es un módulo que funciona con tecnología GSM y que se conecta directamente a un rele de 230V que activa o desactiva la caldera a través de la recepción de mensajes SMS desde un móvil.

Un ejemplo de un producto de este tipo sería **GSMClimDirect** con un precio aproximado en el mercado de unos 215 €.


Figura 5

El sistema permite únicamente el encendido y/o apagado de la alimentación de la caldera. No resulta ser un sistema muy versatil aunque sí económico. Resulta interesante para una segunda residencia sin buscar muchas más pretensiones o funcionalidades.

Un producto muy interesante que sería algo parecido al presente proyecto TFC pero en fase de industrialización es el **DT-1400 Pantalla Táctil TYDOM 4000** complementado con el **Pack TYBOX 4250** con un precio de compra de todo el sistema de aproximadamente **1400€.**


Figura 6

El sistema está formado por un módulo principal que a su vez es el interfaz de comunicación con el usuario y, dos módulos que actúan como sensores de temperatura que envían la información via radifrecuencia a los actuadores que están conectados a los circuitos de calefacción.

El sistema permite la integración de diferentes módulos que son controlados por el módulo principal que incorpora una pantalla táctil. Éste sería el equivalente a la UCC del proyecto dotada de una pantalla táctil y más funcionalidades. El pack adicional sería el equivalente a la red de sensores WSN y al actuador del UCC.

Este sistema no dispone de acceso a través de una apicación WEB ni dispositivo móvil como un SmartPhone o Tablet PC al contrario que el presente proyecto, que sí está planificado fuera del ambito del TFC al ser necesario más tiempo y recursos para su desarrollo.

# 3. Descripción funcional

## 3.1 Sistema inteligente de control de la calefacción

El sistema está formado por una Red de Sensores Inalámbricos (WSN) dotados sensores de temperatura, una Unidad Central de Control (UCC) y una aplicación de Monitorización y Control por el Usuario (MCU) ubicada en un servidor remoto.

La WSN envía los datos de temperatura a la MCU a través de internet. LA UCC recibe la temperatura de consigna de la MCU a través de internet y toma la decisión de encender o apagar la caldera y abrir o cerrar la electroválvula del circuito de agua caliente correspondiente.

En la figura 7 se puede observar el diseño para una casa con tres plantas independientes y un único sistema de calefacción con tres circuitos de agua de calefacción diferentes.

Todo el sistema está basado en el uso de una red wifi para obtener el acceso a internet y por lo tanto al MCU.

Los parámetros de consigna los introduce el usuario a través del MCU. Cada planta tiene sus propios parámetros de consigna, temperatura y hora.

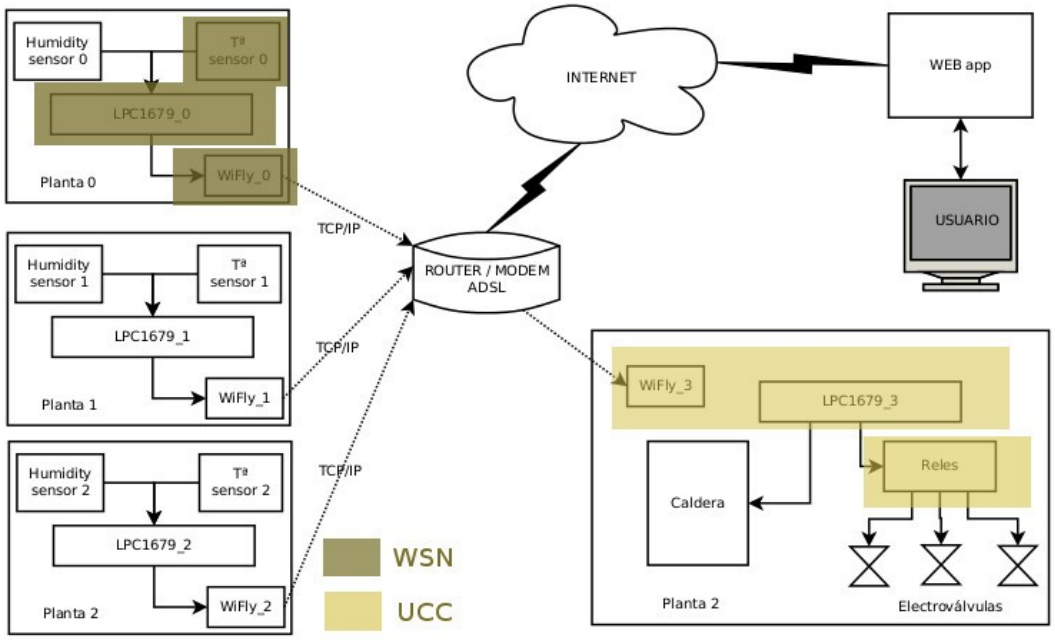


Figura 7

## 3.2 Red de Sensores Inalámbricos de temperatura (WSN)

La Red de Sensores inalámbricos es la encargada de tomar la lectura de la temperatura ambiente de la sala con una frecuencia determinada. La lectura obtenida tiene que ser lo mas real posible, estando esta libre de posible ruido eléctrico que introduzca errores en la lectura. Para ello se ha introducido un algoritmo que intenta minimizar dentro de lo posible los errores de lectura.

Una vez obtenido un valor de temperatura fiable, éste será enviado hasta la MCU utilizando internet como medio. Para ello, el sensor encastado dispone de un módulo de acceso a internet.

Adicionalmente, el sistema estará alimentado con baterias y encapsulado de manera que quede protegido de golpes y caidas.

El sistema, se convierte de esta manera en un sensor de temperatura totalmente transportable y autónomo dentro del rango de uso para el que está diseñado, es decir, dentro de una de las plantas de la casa, ya que el sistema controla única y exclusivamente esa planta especificamente.

En la figura 7 se puede observar en color verde oscuro los elementos que forman la WSN: microcontrolador, sensor de temperatura y módulo Wifi.

## 3.2 Unidad Central de Control (UCC)

El objetivo final del sistema es el de racionalizar el uso de la calefacción calentando aquellas plantas que queremos únicamente y a la temperatura que deseamos. La UCC es la encargada de actuar sobre los diferentes elementos físicos que componen el sistema de calefacción. Será el encargado de activar y desactivar la caldera, así como de abrir y cerrar los tres circuitos de agua caliente de calefacción de una manera independiente entre ellos.

Para poder realizar éstas funciones, la UCC esta formada por los elementos que podemos ver de color amarillo en la figura 7: microcontrolador, módulo Wifi y reles conectados a la caldera y a las electroválvulas que abren y cierran los diferentes circuitos de agua de calefacción.

En una etapa más avanzada de desarrollo, fuera ya del TFC, está planificado la conexión de una pantalla táctil sobre la UCC que servirá como interfaz de usuario para programar las diferentes temperaturas de consigna así como para modificar determinados parámetros del sistema, como podría ser la frecuencia de recogida de los valores de temperatura o de envio de datos a la MCU.

Más adelante, en el apartado 7.1 ampliación y opciones de mejora, se describirá con más detalle que tipo de pantalla se tiene previsto implementar.

## 3.4 Monitorización y Control por el Usuario (MCU)

Una parte fundamental en cualquier sistema es la interacción con el usuario. En el presente proyecto, inicialmente se había considerado la opción de crear una aplicación WEB para la monitorización de la red de sensores, así como para la programación de las diferentes temperaturas de consigna y horarios para cada una de las plantas.

Ésta aplicación estaba previsto de ser creada bajo el framework de Google App GWT al igual que la aplicación facilitada por la UOC Arp@Lab.

Lamentablemente, después de evaluar el tiempo necesario para su implementación, se descartó como parte del TFC. La aplicación completa será desarrollada fuera del ámbito del proyecto bajo el framework de programación Ruby on Rails. Los detalles de la aplicación serán comentados en el capítulo 7.

Aún sabiendo que no era viable la creación completa de la aplicación WEB, era necesario crear algún tipo de interfaz de usuario para poder visualizar lo que está ocurriendo en el sistema encastado de forma remota. La opción escogida ha sido la de modificar el código fuente de la aplicación Arp@Lab para incluir unas funcionalidades mínimas que permitan por un lado, visualizar el estado del sistema en tiempo real y, por otro, verificar el correcto funcionamiento de todo el sistema.

La aplicación modificada ha sido alojada en los servidores de Google App y es de acceso público en la siguiente URL: http://tfcjoansierra.appsppot.com.

Para poder comprobar que la UCC actúa correctamente sobre los rele encendiendo y apagando el sistema, se han creado dos versiones diferentes de la versión modificada de Arp@Lab.

La versión 3 (figura 6) con una temperatura de consigna de 20ºC y la versión 2 (figura 7) con un temperatura de consigna de 35ºC.

# 4. Descripción detallada

## 4.1 Dispositivos físicos utilzados (Hardware)

El objetivo de este capítulo es conocer los diferentes componentes HW utilizados así como la funcionalidad de cada uno de ellos.

### LPCXpresso LPC1769

Es la placa que incluye el microcontrolador LPC1769 con los diferentes dispositivos y puertos de comunicación descritos en el apartado 2.1. Más información disponible en la WEB del fabricante.

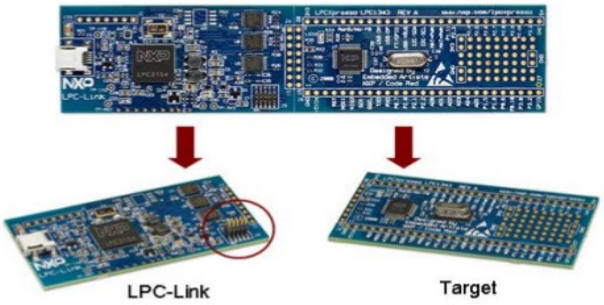


Figura 8

### CP2102 Conversor UART-USB

Dispositivo utilizado como salida de los mensajes para realizar la depuración del código. Adicionalmente he utilizado el PIN3 Vout 5V para alimentar al optoacoplador sin necesidad de utilizar el conversor de niveles en el prototipo creado.

Figura 9

| PIN1 | PIN2 | PIN3 | PIN4 | PIN5 | PIN6 |
|---|---|---|---|---|---|
| Reset | Vout 3V3 | Vout 5V | TxD | RxD | GRND |

### Módulo WiFly RN-171

Módulo de acceso a redes wifi con conexión UART y firmware precargado que permite la rápida creación de aplicaciones y conexión a redes inalámbricas.

Figura 10

### Sensor de temperatura TMP36

Sensor analógico de temperatura con un rango de medida entre -40ºC y 125ºC.

Figura 11

### Placa optoacoplador ILD213T

Permite aislar el microcontrolador de una tensión superior a la soportada. Es conveniente su uso ya que las salidas digitales del LPC1769 son de 3V3 mientras que el array de reles utilizados necesita 5V para su actuación. Dispone de canales acoplados en el mismo circuito.

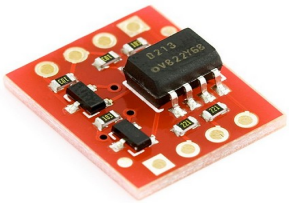

Figura 12

### Conversor de niveles 8 canales TXB0108

Adaptador utilizado para convertir las señales de salida digitales de 3V3 a 5V. En el caso del prototipo no ha sido necesario su uso ya que se ha utilizado la salida de 5V proporcionada por el CP2102. En el modelo productivo será necesario su uso para dotar al sistema de autonomía ya que el CP2102 es sólo una herramienta de desarrollo.

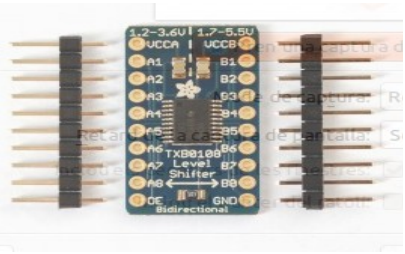


Figura 13

### Rele de cuatro canales 5v / 230V

Actuador utilizado para activar las electroválvulas.

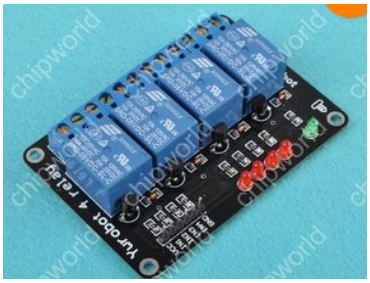

Figura 14

## Descripción placa prototipo montada

Con todos los componentes y el proyecto bien definido se procede al montaje del producto prototipo. En la figura 15 podemos ver su esquema y conexionado entre los diferentes elementos que forman la placa prototipo. A mencionar que el montaje se ha realizado sobre una placa de prototipo. Algunos componentes no han sido descritos en este apartado, como son resistencias, leds y cables, por considerarse de menor interés.

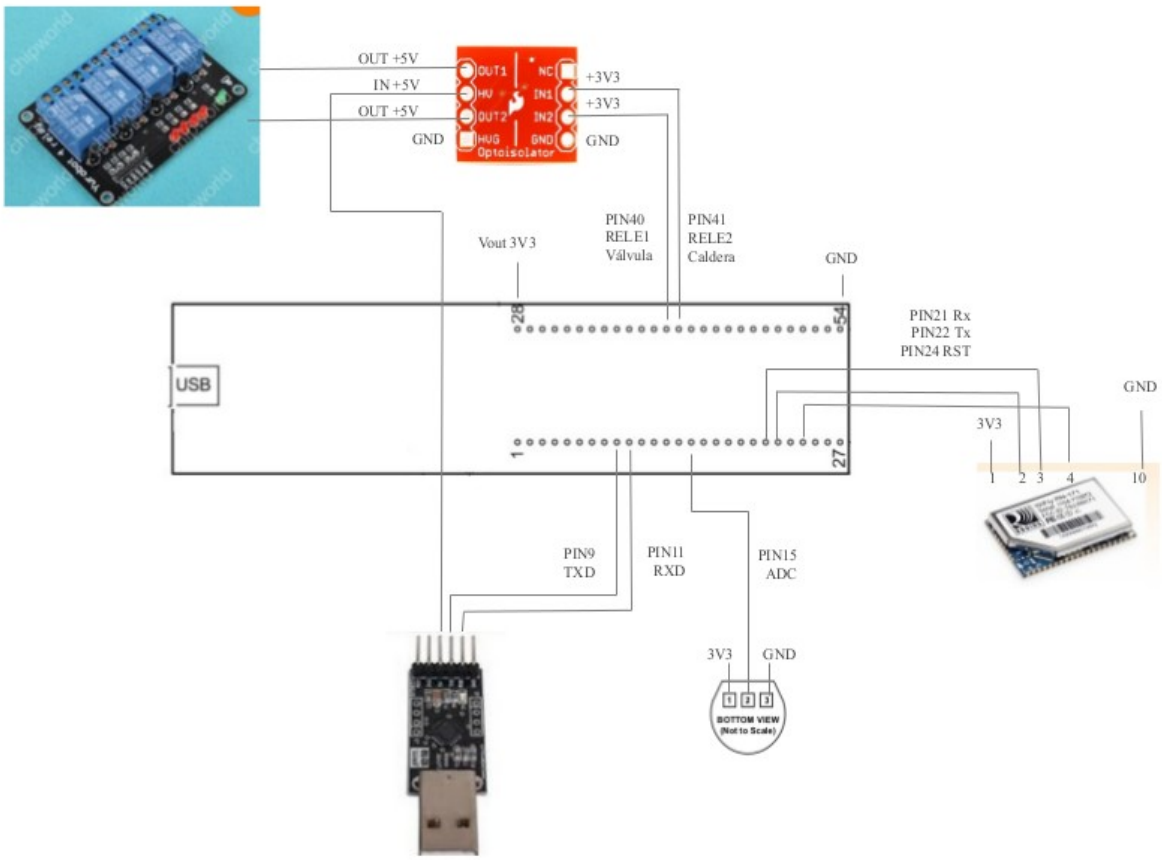


Figura 15

El nucleo del sistema es la placa LPCXpresso con el microcontrolador LPC1769 que está dotado de una CPU Cortex M3 a 120 Mhz como se ha descrito en capítulos anteriores. La placa dispones de dos filas de 27 pin de conexión que dotan a la placa de la capacidad de comunicarse que el exterior.

En esta versión prototipo, la alimentación al sistema se ha realizado a través de la conexión USB disponible en el módulo JTAG de la placa. En una versión productiva se deberá dotar al sistema de una batería o alimentación de red adapatada a los 3V3 necesarios para su funcionamiento.

Cuando la placa es alimentada a través del puerto USB, el PIN 28 se comporta como Vout con una tensión de salida de 3V3. He utilizado este PIN para dotar a la protoboard de la alimentación necesaria. Para completar el circuito de alimentación conectamos el PIN 54 al borne negativo de la protoboard ya que este PIN tiene como función GRND.

La conexión de la placa LPCXpresso con el conversor Serie-USB se ha realizado a través del puerto UART número 3. Para ello se han conectado los PIN 9 y PIN 10 a los PIN 4 y PIN 5 del CP2102 respectivamente, siendo éstos los canales de transmisión y recpción de datos en el mismo orden.

La conexión de la placa LPCXpresso con el módulo WiFly RN-171 se ha realiza también a través de un puerto de comunicación UART, pero en esto caso el puerto UART número 0. Los PIN 21 y PIN 22 de la placa LPCXpresso son los canales de recepción y transmisión de datos respectivamente. Éstos se han conectado a los PIN 3 y PIN 2 respectivamente del módulo WiFly RN-171. En este caso, adicionalmente, hemos configurado un puerto GPIO como salida para poder realizar un RESET al módulo WiFly RN-171 via software. Para ello el PIN 24 del LPC ha sido conectado al PIN 4 del WiFly.

El módulo WiFly ha sido dotado de 3V3 de tensión utilizando los PIN 1 (Vin) y PIN 10 (GRND) directamente desde la placa protoboard, que ha su vez esta alimentada con los 3V3 de tensión del LPC.

El sensor de temperatura TMP36 requiere la conexión de un puerto de entrada analógico al LPC. Para ello se ha configurado el PIN 15 como ADC y se ha conectado a la patilla central del sensor de temperatura, que proporciona un valor de tensión variable y lineal dependiendo de la temperatura ambiente. El sensor es dotado de tensión directamente desde la protoboard con 3V3.

Finalmente se han configurado los PIN 40 y PIN 41 GPIO de la placa LPCXpresso como salidas digitales. Ambos PINs están conectados a los rele o actuadores en la versión productiva, pero en este prototipo el PIN 40 que comandará la caldera estará directamente conectado a un LED rojo en la protoboard.

El rele necesita de una tensión de 5V para poder actuar y, la tensión de salida de los puertos digitales de la placa LPCXpresso es de tan sólo 3V3. Inicialmente se había optado por el uso de un conversor de niveles de 8 canales (figura 13), pero en la versión prototipo he decidido utilizar la salida de 5V proporcionada por el conversor CP2102.

He considerado del todo necesario aislar a la placa LPC y por tanto al microcontrolador LPC1769 de una posible entrada de tensión de 5V que podría llegar a dañar el sistema. Para ello he utilizado un optoacoplador de 2 canales. La entrada estará conectada al PIN 41 que ofrece una salida de 3V3. La parte del circuito de alta tensión (5V) estará alimentado directamente desde el CP2102. El optoacoplador actúa en este caso como interruptor sobre la tensión de salida de 5V; cuando la entrada 3V3 esta en baja, la salida de 5V tambien está en baja, pero cuando la entrada de 3V3 está en alta, la salida se activa automáticamente. De esta manera eliminamos todo riesgo de sobretensión para el

microcontrolador por un posible error de manipulación o configuración de los puertos GPIO.

De esta manera el rele es activado cuando el PIN 41 está en alta, independientemente que la placa tan sólo puede aportar 3V3 de tensión de salida y, desactivado cuando está en baja.

Como comentado anteriormente, en una versión productiva será necesario utilizar el conversor de niveles para subir la tensión de los 3V3 a los 5V de una forma autónoma sin depender del CP2102, que únicamente se utiliza como herramienta de depuración.

El prototipo montado ha resultado totalmente funcional y no ha presentado problemas de fiabilidad. A destacar unicamente algunos problemas de contacto con la conexión de los PIN del módulo WiFly debido al diferente espaciado entre los PIN respecto al estandard más utilizado habitualmente. Una solución para la versión productiva pasa por la compra de una adaptador existente en el mercado. Lamentablemente, en el momento de la compra de los componentes para realizar el prototipo, el adaptador se encontraba sin existencias y no ha sido posible su utilización.

## 4.2 Aplicaciones programadas (Software)

La aplicación desarrollada en el sistema empotrado ha sido programada en lenguaje C estandard bajo el sistema operativo de código abierto FreeRTOS utilizando un Entorno Integral de Desarrollo basado en el aclamado Eclipse, LPCXpresso.

FreeRTOS es un sistema operativo en tiempo real diseñado para sistemas que necesitan de una alta eficiencia y calidad. Es compatible con más de 33 tipos diferente de arquitectura de microcontroladores y soporta numerosas herramientas de desarrollo.

Al ser un sistema en tiempo real, la aplicación principal contine un bucle *main()* en el que habitualmente se inicializa el sistema, se crean las diferentes tareas y se lanza el gestor de tareas o vTaskStartScheduler.

Las tareas son como funciones que realizan una actividad muy especifica y pueden coexistir tantas como sea necesario. El gestor de tareas se encarga de cambiar el estado de cada tarea de los posibles que puede estar, ya que únicamente una tarea puede estar en funcionamiento simultaneamente, sólo disponemos de un microprocesador.

El gestor de tareas utiliza un sistema de prioridades para saber que tarea se tiene que ejecutar en cada momento. La prioridad más baja es el 0 que tiene una función especial y es utilizada por una tarea llamada 'idle'. Esta tarea asegura que en el caso de que ninguna otra tarea esté disponible para su ejecución, siempre existirá una para ser ejecutada por el procesador. Habitualmente la prioridad mas baja utilizada es la 1, pero esta puede ser un número como 100 o incluso 200.

Los estados posibles de una tarea son:

- Ready: esperando ejecución.
- Suspended: en suspensión, detenida.
- Running: en ejecución.
- Blocked: esperando algún recurso o un evento.

Tan sólo las tareas en estado 'ready' pueden entrar en ejecución y por lo tanto consideradas por el gestor de tareas. Siempre que exista una tarea de prioridad superior se ejecutará preferentemente sobre las demás de prioridad inferior. Si existen dos tareas con la misma prioridad, el gestor de tareas repartirá diferentes 'slots' de tiempo para cada una de ellas.

Una tarea en estado 'blocked' pasará a 'ready' cuando se haya producido el evento que esperaba o se haya liberado el recurso que necesitaba y estaba ocupado.

A continuación, en la figura 16, se pueden apreciar los diferentes estados y sus posibles transiciones.

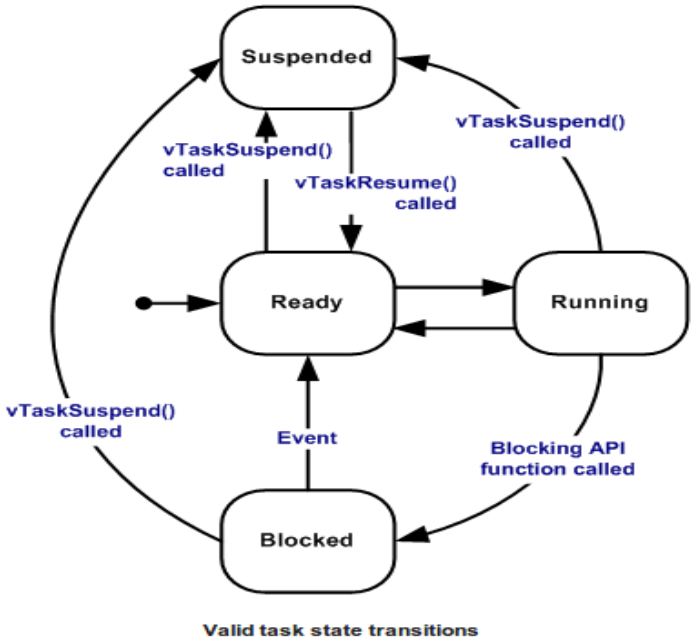


Figura 16

He comentado que una tarea puede estar esperando la liberación de recursos. Como podemos tener muchas tareas ejecutandose y compartiendo recursos, tenemos que asegurar de alguna manera que estos recursos se comparten de una forma segura. Para ello FreeRTOS nos permite el uso de semáforos.

También es posible compartir datos entre tareas y, una herramiento muy potente que nos facilita FreeRTOS es el uso de colas de diferentes tipos.

Esta pequeña exposición de FreeRTOS nos sirve para entender como se ha creado la la estructura de la aplicación para el sistema embebido de este TFC. Se han creado tres tareas diferentes que describiré con detalle a continuación.

Éstas tareas necesitan utilizar recursos externos o internos y, para ello también ha sido necesaria la creación de diferentes 'drivers' o controladores que permitan su uso.

En la figura 17 podemos ver el diagrama de bloques correspondiente al software programado en el sistema desarrollado.

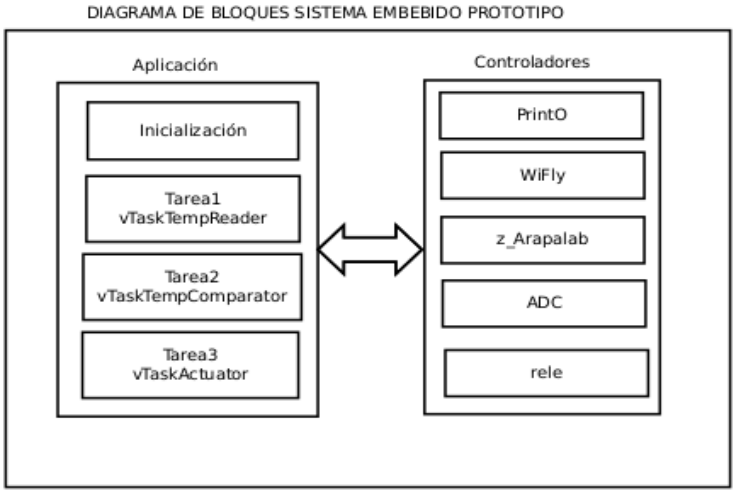


Figura 17

Como se puede apreciar, los controladores son funciones que están disponibles para su uso por parte de la aplicación. A continuación describiré brevemente las funcionalidades de cada controlador. El código fuente está comentado adecuadamente para profundizar más en elfuncionamiento de las funciones en caso necesario.

**printo**

La función de este controlador es la de enviar mensajes de texto a través de un puerto UART para poder ser visualizados por un terminal con conexión serie. Es de vital importancia para el depurado de codigo, ya que con este controlador enviamos aquellos mensajes que nos serán de utilidad para verificar el correcto funcionamiento del codigo programada. En definitiva es una herramienta de depuración.

Consta únicamente de tres funciones:

- ***uint32_t PrintoSetup (int uart, int baud)***

Incializa el puerto UART 3.

- ***uint32_t printOut (int uart, char *outString)***

  Envia una cadena al puerto UART definido.
- ***void printOutTextandNum ( uint8_t port , char *Texte , signed int Num )***

  Envia un texto y un número al puerto UART definido.

**wiflyDriver**

Este controlador nos permite el uso del módulo de comunicación WiFly RN-171. Su programación se ha realizado comunicando a traves del puerto UART 0 y tenemos disponibles las siguientes funciones:

- ***void WiflyEmptyBuffers ()***

  Vacia los buffers de lectura del puerto UART.

- ***void WiflyDelayMs(int ms)***

Genera un retraso especificado en milisegundos.

- ***void WiflyReset()***

Reinicia el dispositivo

- ***uint32_t WiflyGetAnswer(char *String)***

Obtiene la respuesta del puerto UART.

- ***uint32_t WiflyInit(int uart, int baud)***

Inicializa el dispositivo.

- ***uint32_t WiflyCommandMode(int uart)***

Entra en modo consola.

- ***uint32_t WiflySendCommand (int uart, char *command)***

Envía un comando al módulo.

- ***uint32_t WiflySendCommand2 (int uart, char *command)***

Envía un comando al módulo (diferente retorno de línea).

- ***uint32_t WiflyJoinOpenNet ( char *bssid)***

Conecta el módulo a una red wifi sin protección.

- ***uint32_t WiflySendPing (char *ip)***

Envía un ping a la ip indicada.

- ***uint32_t WiflyGetIp()***

Obtiene la IP asignada al módulo por la red wifi.

- ***uint32_t WiflyGetChannel()***

Obtiene el canal al que está conectado el módulo.

- ***uint32_t WiflyGetRssi()***

Obtiene la intensidad de señal recibida del punto de acceso.

- ***uint32_t WiflyGetTime()***

Obtiene el tiempo desde el arranque del dispositivo.

- ***uint32_t WiflyGetTxPower()***

Obtiene la potencia de transmisión del módulo.

- ***uint32_t WiflySetTxPower(uint32_t power)***

Fija la potencia de transmisión del módulo.

**arpalab**

Arpalab es la aplicación web utilizada para visualizar los datos de una red de sensores. Con este controlador podemos fijar los parametros necesarios para el envío y recepción de datos a la aplicación.

- ***uint32_t ArpalabInit(char *dns, int port)***

Inicializa el módulo con el DNS y puerto de conexión de la aplicación.

- ***long ArpalabGetData()***

Envía un comando GET HTTP para obtener un dato de la aplicación.

- ***uint32_t ArpalabSendDataSensor( uint32_t data)***

Envía el valor de la temperatura a la aplicación.

- ***uint32_t ArpalabSendDataUCC( uint32_t data)***

Envía el estado de la caldera y la electroválvula a la aplicación.

**adc**

Con este controlador obtenemos un valor digital del sensor de temperatura. Las diferentes funciones programadas son:

- ***int ADC_ReadRawSample(int channel)***

Lee y convierte a digital 100 valores de tensión del sensor de temperatura TMP36. Inlcuye un pequeño algoritmo para intentar minimizar el posible ruido.

- ***void ADC_stopChannel(int channel)***

Para el canal ADC.

- ***void ADC_StartChannel(int channel)***

Inicializa el canal ADC

- ***void ADC_init(void)***

Inicializa el conversor ADC.

- ***void ADC_read(uint8_t channel_num)***

Lee el valor unitario de un canal.

- ***int ADC_isStarted()***

Verifica que el conversos ADC se ha inicializado previamente.

**rele**

Este controlador tiene como función configurar los puertos GPIO en puerto de salidas digitales. Tiene únicamente tres funciones:

- ***void rele_init()***

Inicializa los puertos GPIO y los configura como salida digital.

- ***void rele_on(int rele);***

Pone en alta el puerto indicado.

- ***void rele_off(int rele)***

Pone en baja el puerto indicado.

## La aplicación

Durante la fase de inicialización de la aplicación realizamos las siguientes funciones descritas en el diagrama de bloques de la figura 18.

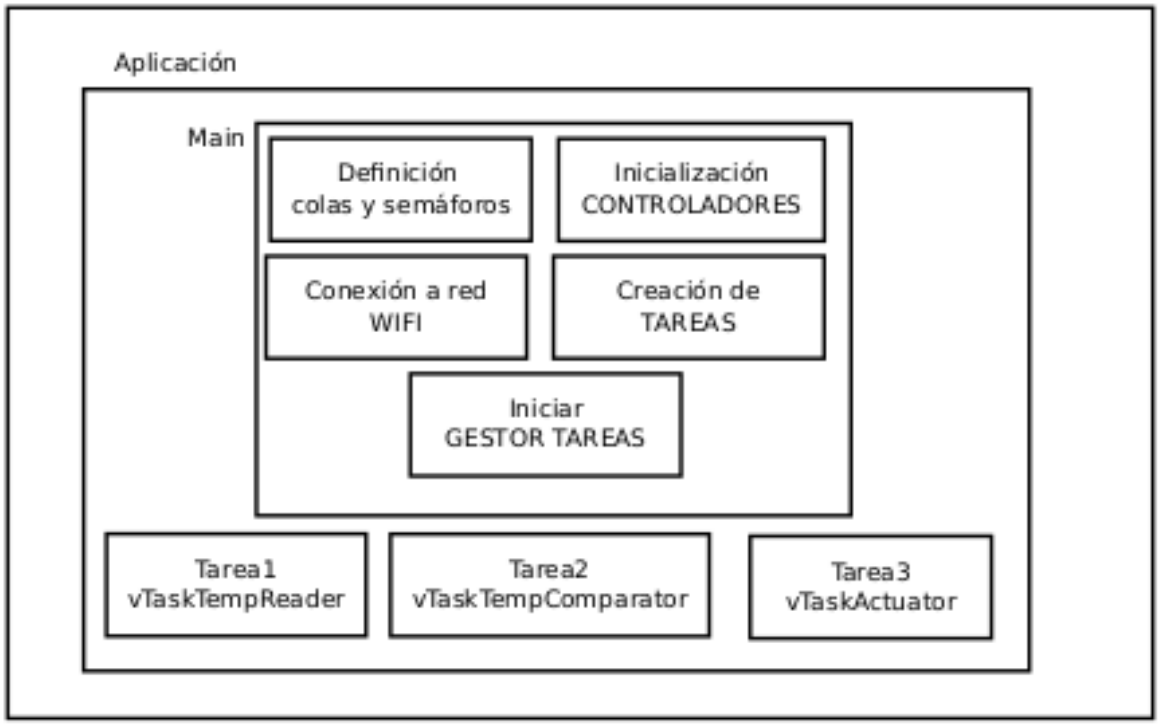


Figura 18

Las tareas tienen una parte de código que se ejecuta la primera vez únicamente y, la parte mas importante de la actividad se ejecuta en un bucle sin fin. A continuación paso a describir cada una de las tres tareas a través de sus organigrámas.

### VtaskTempReader

La finalidad de esta tarea es la de obtener el valor digital de la tensión entregada por el sensor de temperatura y convertirlo a un valoe de temperatura en grados centígrados. Adicionalmente realiza dos funciones más; enviar el dato obtenido a la 'cola2' para que sea tratado por otra tarea y, enviar el mismo data a la aplicación web de monitorización MCU ubicada en la URL: http://tfcjoansierra.appspot.com. En la figura 19 podemos ver el diagrama de flujo de la tarea programda.

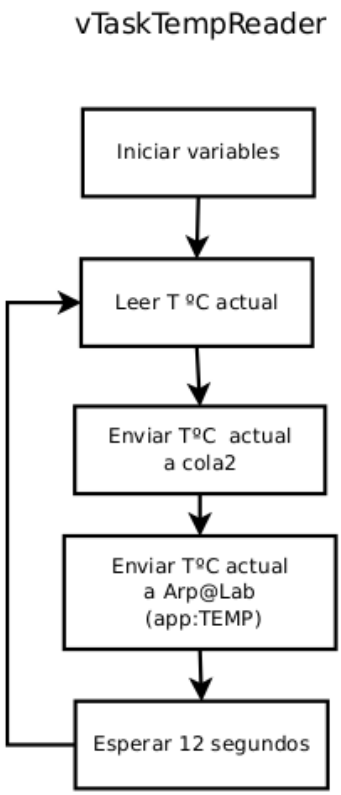


Figura 19

### vTaskTempComparator

Esta tarea recoge el valor de temperatura de la 'cola2' y lo compara con el dato que pide a la aplicación web http://tfcjoansierra.appspot.com. Del resultado de la comparación crea un código de 4 bits que coloca en la 'cola1' para que la siguiente tarea pueda recogerlo.

Este código indica si se debe activar o no la caldera y la electroválvula. Mas detalles de la codificación utilizada se puede encontrar en el código fuente. En la figura 20 podemos observar el diagrama de flujo de la tarea.

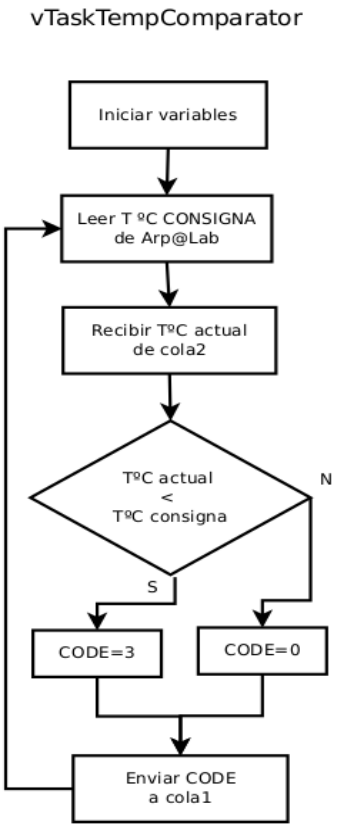


Figura 20

### vtaskActuator

Esta tarea recibe el codigo a través de la 'cola1' y activa o desactiva en consecuencia los puerto conectados a los rele. Adicionalmente envía el estado de la caldera y la válvula a la aplicación WEB http://tfcjoansierra.appspot.com. A continuación diagrama de flujo de la tarea en la figura 21.

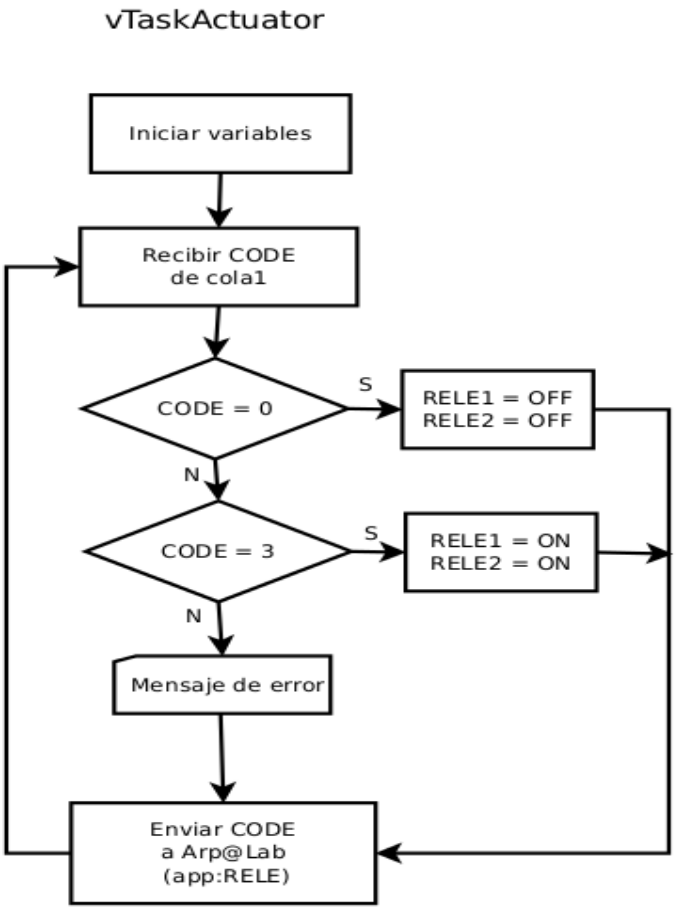


Figura 21

## 4.3 Interfaz de usuario

Como comentado anteriormente, la idea inicial del proyecto era la de crear una aplicación web totalmente funcional que permitiese no sólo monitorizar la temperatura, sino que también la programación de las diferentes temperaturas de consigna en función de la franja horaria para cada una de las plantas.

Lamentablemente está aplicación ha quedado fuera del ámbito del presente TFC. Como alternativa he optado por modificar la aplicación Arp@Lab facilitada por la UOC y ubicada en la URL: http://arpalab.appspot.com.

Las modificaciones han consistido en incluir unos campos de texto que faciliten información sobre el último valor de temperatura recibido, el valor de la temperatura de consigna y, el estado de la caldera y la válvula.

Para ello ha sido necesario realizar las siguientes modificaciones en el código fuente del lado clliente de la aplicación GWT Arp@Lab:

- **MainWindow.ui.xml:** adición de las etiquetas necesarias.
- **MainWindow.java:** adición del método ***updateLabels()***
- **OpenWSNChart.java***:* adción de la llamada al método anterior dentro del método ***getDataFromServers()*** en el caso ***OnSuccess().***

Y en la parte del servidor para poder enviar la temperatura de consigna al sistema embembed he tenido que modificar el método GetRandom de la clase GetRandom.

En la figura 22 podemos ver una captura de pantalla de la aplicación modificada y funcionando.

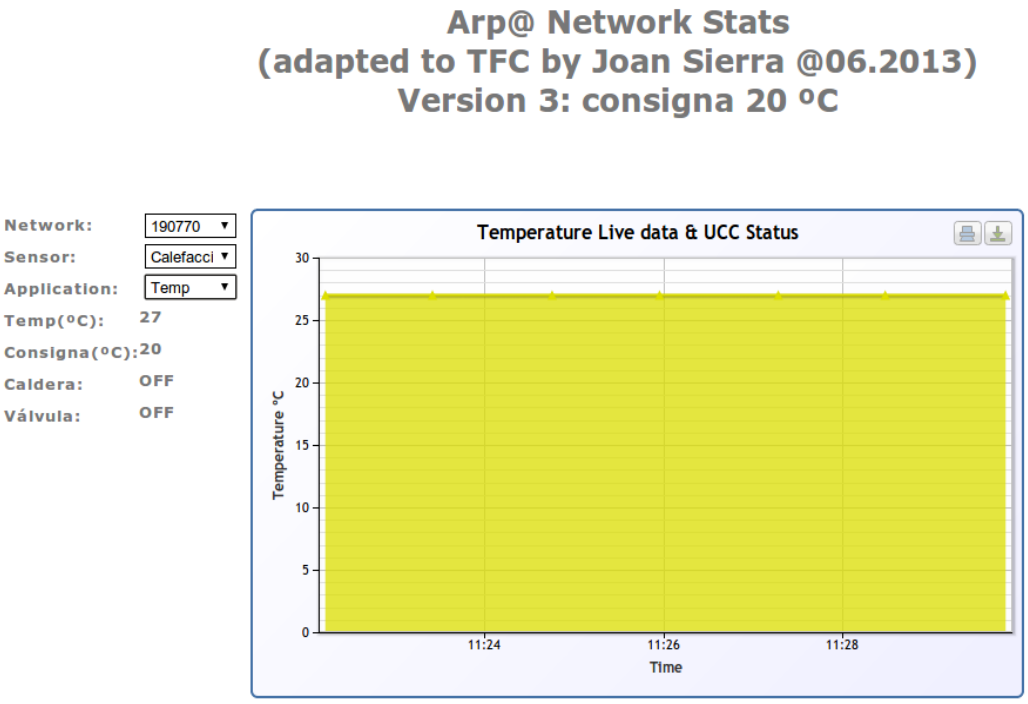


Figura 22

Si seleccionamos del desplegable 'Application' la aplicación 'rele', lo que obtendremos en el gráfico será el histórico del código utilizado para actuar sobre la caldera y las electroválvulas (Figura 23).

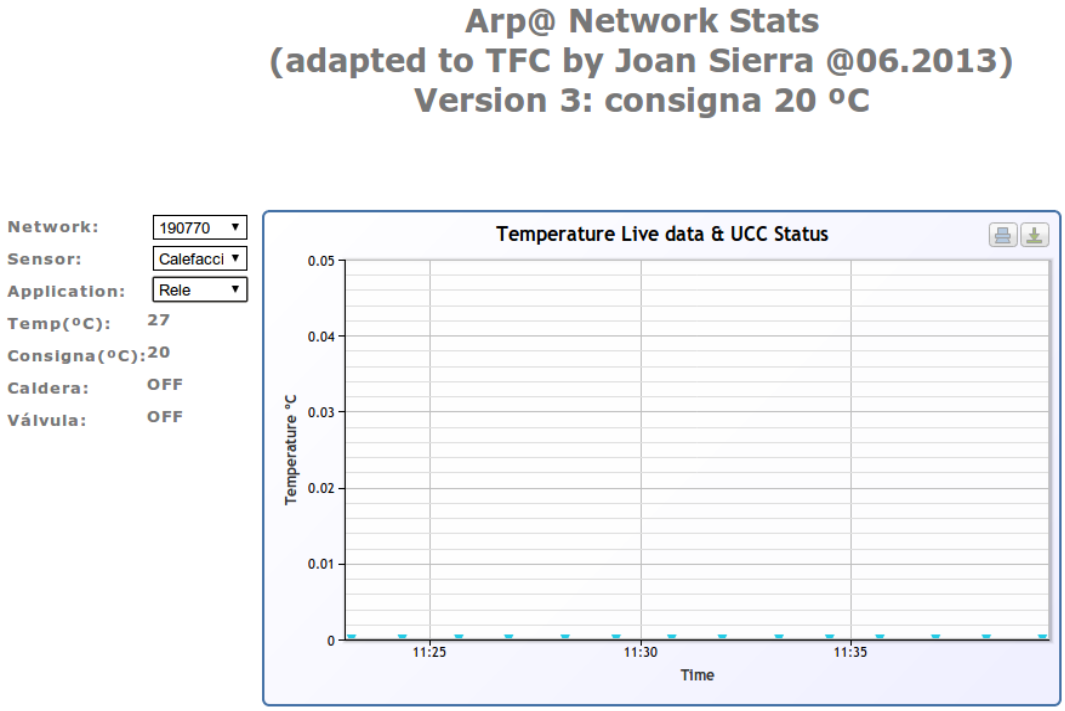


Figura 23

Como se aprecia en la imagen, en este caso se trata de la versión 3 que tiene asignada una temperatura de consigna de 20ºC.

He creado otra versión con una temperatura de consigna de 35ºC con el objetivo de poder verificar el correcto funcionamiento de la aplicación. En la figura 24 se puede apreciar la versión 2 de la aplicación con un temperatura de consigna de 35ºC como ha producido la activación de caldera y válvula.

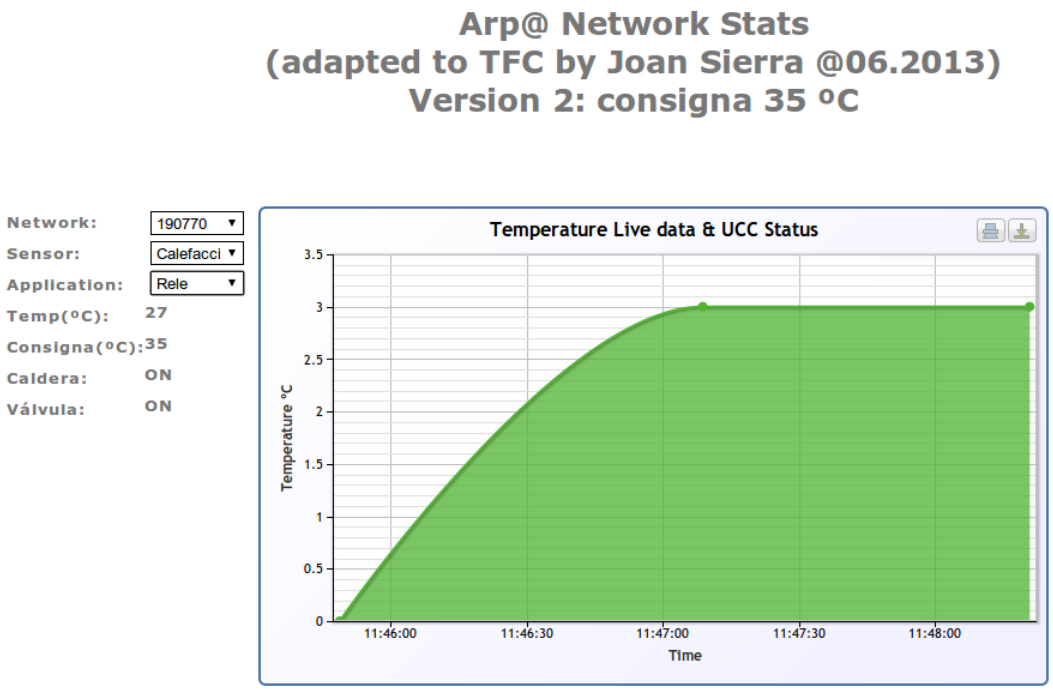


Figura 24

# 5. Viabilidad técnica

El proyecto tal y como está presentado no creo que sea viable desde un punto de vista comercial. Sus objetivos son muy específicos y este motivo lo hace poco atractivo para el mercado.

De todas maneras, el sistema presentado es la base para crear un sistema domótico de control para casas o industrias. La integración de diferentes sensores en el sistema encastado acompañado de una batería dotaría al sistema de una autonomía y flexibilidad extraordinaria poniendo a disposición números recursos de monitorización (temperatura, humedad, sensores de humo, gas, presencia, cámaras, micrófonos, etc).

Los sistemas que actuasen como red de sensores (WSN) deberían estar equipados de una carcasa adecuada y todo el sistema estaría basado en el acceso a una red wifi.

Por otro lado, el sistema que actúa sobre las electroválvulas y la caldera es un ejemplo de aplicación a partir de una de las variables leidas. Las aplicaciones se podrían extrapolar de diferentes maneras a los diferentes sensores que necesitemos.

Lo ideal sería acompañar a la red WSN con una aplicación WEB que permitiera el control y actuación sobre las variables monitorizadas y adicionalmente recopilar la información en una base de datos para tener de esta manera un histórico de los registros. Inicialmente, esta aplicación limitada a la gestión de la calefacción se habia planificado en este TFC, pero debido a falta de tiempo ha tenido que ser suprimida.

A pesar de todo, el sistema presentado es técnicamente viable siendo capaz de monitorizar la temperatura y de activar caldera y vávula en función de unos valores de consigna con unos recursos mínimos y una tecnología puntera.

Como puntos fuertes del sistema podemos destacar el uso de un microcontrolador potente y versatil con tecnología puntera y la posibilidad de acceder al sistema remotamente.

Como puntos débiles comentar que la elección del sensor de temperatura no ha sido correcta y deberá cambiarse para la versión productiva. El margen de error de 2ºC y el amplio margen de temperatura que puede registrar lo hace poco recomendable para la aplicación diseñada.

Otro de los puntos débiles del sistema es la falta de transparencia de su funcionamiento. Sería altamente recomendable la instalación de una pantalla que permita obtener la información deseada sin necesidad de acceder remotamente con la plaicación web y poder conocer en todo momento cual es el estado del sistema.

# 6. Valoración económica

Con el objetivo de poder comparar el sistema desarrollado con otros equipos en el mercado, a continuación adjunto dos tablas con el desglose del coste total del desarrollo del proyecto prototipo y con el coste teórico del sistema para la definición inicial del proyecto.

**Prototipo**

| Descripción | Cantidad | Precio Unitario (€) | Precio Total (€) |
|---|---|---|---|
| LPCXPresso LPC1769 | 1 | 20.00 | 20.00 |
| CP2102 | 1 | 11.00 | 11.00 |
| WiFly RN-171 | 1 | 26.00 | 26.00 |
| TMP36 | 1 | 2.30 | 2.30 |
| Placa optoacoplador ILD213T | 1 | 4.96 | 4.96 |
| Conversor de Niveles 8 canales TXB0108 | 1 | 8.35 | 8.35 |
| Rele 4 canales 5V/230V | 1 | 4.50 | 4.50 |
| Protoboard | 1 | 13.99 | 13.99 |
| Horas desarrollo y programación | 250 | 15.00 | 3750.00 |
| | | **COSTE TOTAL** | **3841.10 Euros** |

Figura 25

**Sistema completo**

| Descripción | Cantidad | Precio Unitario (€) | Precio Total (€) |
|---|---|---|---|
| LPCXPresso LPC1769 | 4 | 20.00 | 80.00 |
| CP2102 | 1 | 11.00 | 11.00 |
| WiFly RN-171 | 4 | 26.00 | 104.00 |
| TMP36 | 3 | 2.30 | 6.90 |
| Placa optoacoplador ILD213T | 1 | 4.96 | 4.96 |
| Conversor de Niveles 8 canales TXB0108 | 1 | 8.35 | 8.35 |
| Rele 4 canales 5V/230V | 1 | 4.50 | 4.50 |
| Cajas protectoras | 4 | 20.00 | 80.00 |
| Baterias | 3 | 11.00 | 33.00 |
| | | **COSTE TOTAL** | **332.71 Euros** |

Figura 26

# 7. Conclusiones

## 7.1 Ampliación y opciones de mejoras

La mejora más inmediata a introducir es la creación de una aplicación web completa que permita la monitorización y la configuración de algunos de los parámetros del sistema encastado de una forma remota.

Es del todo necesario la inclusión de una pantalla que permita la interacción con el dispositivo localmente sin necesidad de hacerlo a través de la aplicación web. Para ello estoy valorando la incorporación de una pantalla tactil en color de 2.8 “. En concreto el modelo ILI9325 con puerto de comunicación SPI que permitiría la conexión al LPC lo podemos ver en la figura 27. De esta manera el sistema se volvería mucho más versatil y funcional.


Figura 27

Adicionalmente, como mejora sobre los sensores, se podría incorporar una pequeña pantalla LCD que mostrase la temperatura actual, la consigna y el estado de la caldera. Una opción sería la pantalla LCD con fondo azul que se puede ver en la figura 28.


Figura 28

### 7.2 Autoevaluación

Gracias a este proyecto me he introducido de lleno en un tema apasionante y de total actualidad como es el de los sistemas encastados. He aprendido su funcionamiento interno, en parte debido a la especial naturaleza de los sistemas operativos en tiempo real, en parte a la gran cantidad de documentación existente en internet.

Creo que los objetivos del TFC han sido alcanzados, poniendo en práctica diferentes disciplinas estudiadas durante estos últimos años: programación C, POO (java), planificación, diseño de circuitos, exposición de contenido, conocimiento redes, uso de compiladores, etc..

Valoro muy positivamente esta experiencia de programación, que adicionalmente me ha hecho descubrir Java y GWT y abrirme a la programación de aplicaciones web. Adicionalmente he descubierto Ruby on Rails que finalmente será el framework que he decidido utilizar para la aplicación WEB.

# 8. Bibliografía


- "Using the FreeRTOS Real Time Kernel - A Practical Guide NXP LPC17xx Edition". Richard Barry.
- "FreeRTOS Reference Manual". Richard Barry.
- "Using.freertos.lpc17xx.summary". Richard Barry.

## Enlaces Web


- http://cv.uoc.edu/app/mediawiki14/wiki/IniciCortexM3
- http://www.freertos.org

- http://www.embeddedartists.com/products/lpcxpresso/lpc1769_xpr.php
- http://www.lpcware.com/lpcxpresso
- http://es.wikipedia.org/wiki/Sistema_embebido
- https://developers.google.com/web-toolkit/?hl=ca
- http://arpalab.appspot.com
- http://tfcjoansierra.appspot.com
- http://www.eclipse.org/
- http://www.rovingnetworks.com/products/RN171
- http://www.analog.com/en/mems-sensors/digital-temperature-sensors/tmp36/products/product.html
- http://www.deltadore.com/spain/es/catalogo-domotica/control-mandos/pantalles/panel-control/tydom-4000.html
- http://myworld.ebay.com/chipworld/
- http://www.bricogeek.com/shop/
- http://www.google.coom