

Estación meteorológica de bajo consumo

María José Rodríguez Feal

Ingeniería Técnica de Informática de Sistemas

Consultor: Jordi Bécares Ferrés

11 de Junio de 2013

A mi padre

RESUMEN

En este proyecto de final de carrera se ha desarrollado un sistema empotrado sobre la placa LPCxpresso LPC1769. Mediante el uso de sensores se obtienen parámetros físicos ambientales que son transmitidos regularmente a Internet vía Wi-Fi por medio del módulo de comunicación inalámbrica Wifly RN-XV, facilitando así, conocer en todo momento las condiciones atmosféricas del lugar a analizar u observar. La estación permite ser configurada a través de una aplicación web adaptándola a las necesidades reales.

Un punto esencial ha sido el ahorro de energía, dotando al sistema de un modo de reposo cuando no hay operaciones que realizar.

Para el desarrollo de la aplicación se ha utilizado el sistema operativo en tiempo real FreeRTOS diseñado para sistemas embebidos con limitados recursos.

Palabras clave: sistemas empotrados, microcontroladores, sensores, red de sensores inalámbricas, LPCxpresso LPC1769, Wifly RN-XV, FreeRTOS

El Trabajo Final de Carrera pertenece al área de Sistemas Empotrados de los estudios de Ingeniería Técnica de Informática de Sistemas de la Universitat Oberta de Catalunya.

ÍNDICE DE CONTENIDOS

1. Introducción

1.1	Descripción.....	7
1.2	Objetivos.....	8
1.3	Enfoque y método seguido.....	9
1.4	Planificación del proyecto.....	9
1.5	Recursos empleados.....	11
1.6	Productos obtenidos.....	15
1.7	Descripción del resto de capítulos.....	15

2. Descripción funcional

2.1	Sistema total.....	16
2.1.1	Diagrama de bloques de la aplicación.....	16
2.1.2	Comunicación e interacción de los diferentes objetos del sistema..	17
2.2	Diseño de la interfaz de usuario.....	20
2.2.1	Interfaz de configuración.....	20
2.2.2	Interfaz de resultados.....	22
2.2.3	Funcionamiento de la aplicación WEB.....	22
2.3	Diseño de la aplicación empotrada.....	24
2.4	Esquema de conexiones.....	25

3.	Descripción Aplicación Sistema Empotrado	
3.1	Aplicación principal.....	26
3.2	Librería de controladores de dispositivos.....	33
3.2.1	Módulo wifly.h.....	34
3.2.2	Módulo log.h.....	37
4.	Aplicaciones prácticas	
4.1	Climatización.....	38
4.2	Sistema de riego.....	40
4.3	Alarma de incendios.....	41
5.	Viabilidad técnica.....	43
6.	Valoración económica.....	44
7.	Conclusiones.....	45
8.	Glosario.....	46
9.	Bibliografía.....	47

ÍNDICE DE FIGURAS

Figura 1: Planificación inicial	8
Figura 2: Planificación final	8
Figura 3: LPC1769 LPCXpresso board	11
Figura 4: Módulo WiFly RN-XV	11
Figura 5: Módulo CP2102 Adaptador USB a 3v3 TTL	12
Figura 6: Sensor de temperatura LM35	12
Figura 7: Fotorresistencia LDR.....	13
Figura 8: Leds, botón, zumbador.....	14
Figura 9: Componentes varios.....	14
Figura 10: Producto obtenido.....	15
Figura 11: Diagrama de bloques del sistema total.....	16
Figura 12: Diagrama de bloques del sistema empotrado.....	17
Figura 13: Conexiones UART.....	18
Figura 14: Conexiones ADC.....	19
Figura 15: Interfaz de configuración	20
Figura 16: Resultados de la lectura de los sensores	22
Figura 17: Diagrama de bloques de la interfaz Web.....	23
Figura 18: Diseño de la aplicación empotrada.....	24
Figura 19: Esquema de conexiones del SE.....	25
Figura 20: Función main.....	26
Figura 21: Diagrama de flujo del control de la configuración.....	27
Figura 22: Lectura y envío de los valores de los sensores de forma periódica.....	28
Figura 23: Lectura periódica e instantánea de los sensores.....	29
Figura 24: Filtro RC	29
Figura 25: Diagrama de flujo del control de conexión.....	30
Figura 26: Diseño del módulo wifly.....	36
Figura 27: Sistema log.....	37
Figura 28: Esquema de conexión de la calefacción.....	38
Figura 29: Diagrama de flujo del sistema de calefacción.....	39
Figura 30: Diagrama de flujo del sistema de riego.....	40
Figura 31: Esquema de conexiones del sistema de alarma.....	41
Figura 32: Diagrama de flujo del sistema de alarma.....	42

1. Introducción

Los sistemas empotrados y las redes de sensores inalámbricas formadas por pequeños dispositivos y provistas por algún tipo de sensor (temperatura, luminosidad, etc.) abarcan una amplia variedad de aplicaciones, como la domótica, control de tráfico o control ambiental, y permiten integrar funcionalidades que anteriormente eran independientes unas de otras.

En este sentido se ha desarrollado una aplicación práctica con capacidad de comunicación Wi-Fi, que a partir de un conjunto de sensores adquiere regularmente información física del entorno.

1.1. Descripción

El proyecto consiste en el desarrollo de una estación meteorológica basada en un microcontrolador, un módulo de comunicaciones Wi-Fi y un conjunto de sensores.

En concreto, la estación deberá realizar mediciones periódicas de temperatura y luminosidad, y transmitir las a través de la red inalámbrica con el fin de poder observar en tiempo real las condiciones atmosféricas del lugar de interés. Asimismo, se desarrollará una interfaz Web que permitirá controlar algunas variables del sistema como la frecuencia de las mediciones.

Dada la variedad de aplicaciones que ofrecen este tipo de sistemas, se mostrarán algunos ejemplos prácticos de estas aplicaciones, como podrían ser el control de la climatización de una estancia o un sistema de alarma de incendios.

Un aspecto crítico de los sistemas embebidos es el consumo de energía. Muchas aplicaciones empotradas pasan la mayor parte del tiempo inactivas a la espera de que suceda algo. Se diseñará el sistema en modo de bajo consumo durante estas esperas para no desperdiciar energía.

1.2 Objetivos

Se listan los objetivos y una breve descripción a continuación:

- **Mediciones periódicas de los sensores**

Se dispone de los sensores analógicos: sensor de temperatura y sensor de luminosidad. El sistema deberá realizar lecturas periódicas de estos sensores.

- **Reporte de las mediciones a la web**

Los resultados de las mediciones de los sensores se enviarán vía Wi-Fi a un servidor web accesible desde un navegador.

- **Interacción del usuario**

Se facilitará la configuración del sistema a través de una interfaz web. Se permitirá modificar la velocidad de las mediciones, activar o desactivar sensores, ingresar umbrales de temperatura y luminosidad, y cambiar la frecuencia de actualización de los datos de configuración.

- **Lectura instantánea de los sensores**

Mediante un pulsador el sistema efectuará una lectura instantánea de los sensores y el resultado se enviará a la web.

- **Reinicio ante fallos**

Dispondrá de un mecanismo que, ante un fallo de conexión o bloqueo de la placa, reiniciará el sistema.

- **Bajo consumo**

Se colocará el procesador en modo de bajo consumo cuando esté inactivo.

- **Simulación de aplicaciones**

Se utilizarán leds y un zumbador para simular las aplicaciones que ofrece el sistema: control de riego, alarma de incendios y climatización de una estancia.

- **Reset manual**

Mediante un botón se forzará el reinicio de la ejecución de la aplicación.

1.3. Enfoque y método seguido

Para el desarrollo de la aplicación se ha seguido la técnica de diseño modular, combinando el modelo de diseño ascendente o “bottom-up” y el modelo descendente o “top-down”.

Inicialmente se ha seguido un diseño “bottom-up”. Se han implementado en algunos casos, o utilizado en otros, los módulos de bajo nivel: los controladores de dispositivos, y el resto de módulos se han desarrollado para acomodarlos a los diseños previos.

El desarrollo de la aplicación principal ha seguido un diseño “top-down”, donde el flujo de control del programa se ha fraccionado en módulos y éstos se han dividido en submódulos, de forma jerárquica.

La programación modular ha permitido descomponer el programa en trozos más pequeños, manejables, fáciles de probar y modificar.

1.4. Planificación del proyecto

Dados los limitados conocimientos al inicio del curso sobre los sistemas empotrados y lo difícil que resulta plantearse un proyecto partiendo de esta situación y una pequeña fase de formación posterior, la intención al elaborar la planificación inicial era cumplir los requisitos de la propuesta facilitada en la asignatura. Estos requisitos se pueden ver en el siguiente diagrama de Gantt:

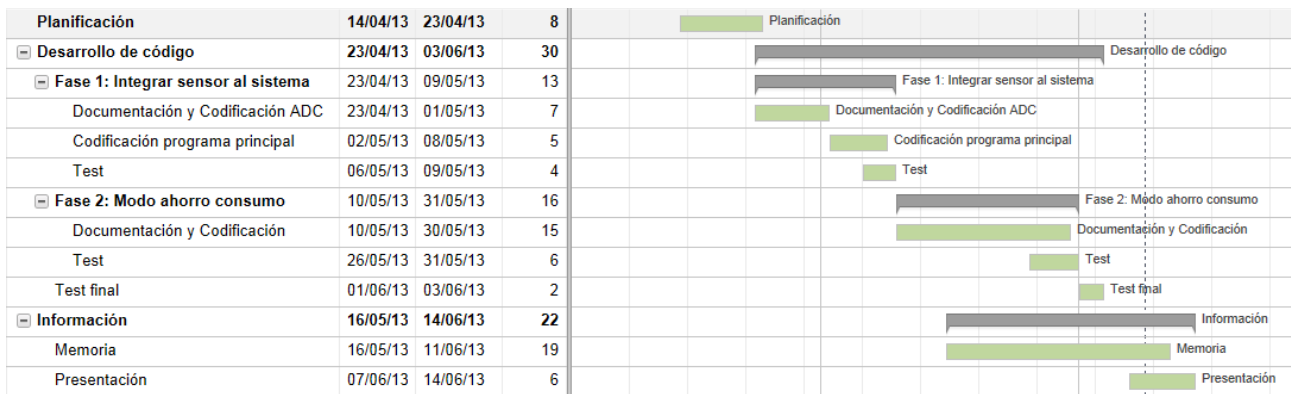


Figura 1: Planificación inicial

Una vez cumplido los requisitos mínimos, se han ampliado los objetivos añadiendo nuevas funcionalidades al sistema. La planificación final se puede ver en el siguiente diagrama:

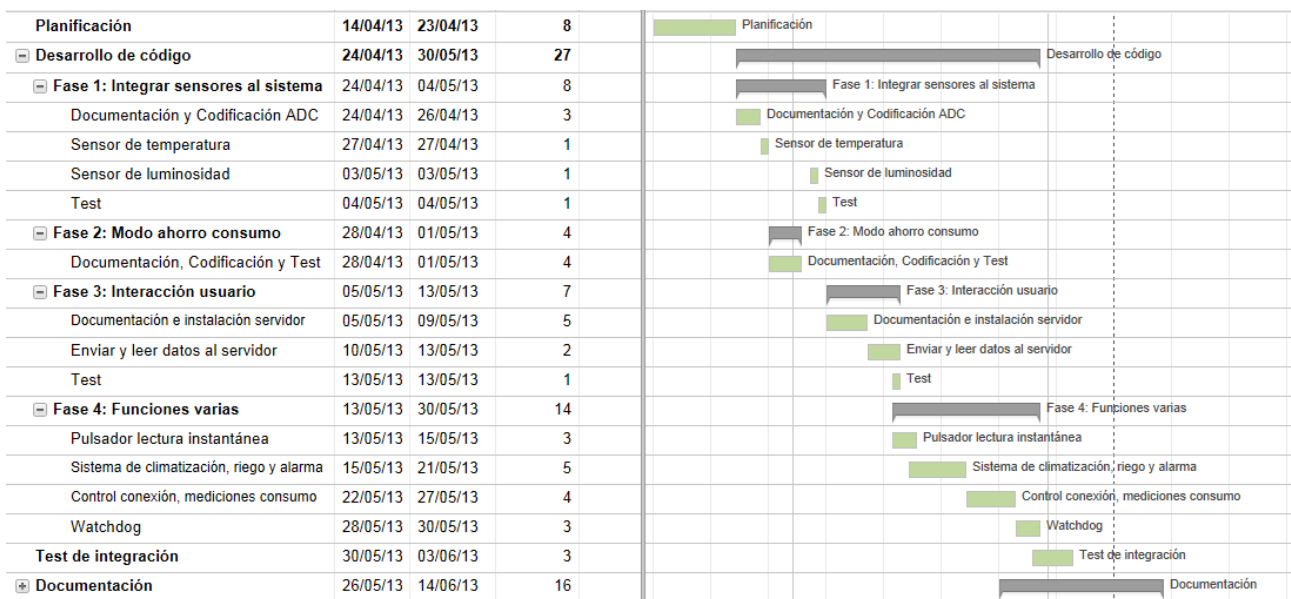


Figura 2: Planificación final

Las diferencias entre la planificación inicial y la final se han debido principalmente a dos motivos:

- Encontrar la documentación necesaria para llevar a cabo los requisitos antes del tiempo previsto.
- Utilizar y/o modificar el software del fabricante para el manejo de los controladores, como el ADC, en lugar de implementarlos desde cero ha ayudado a cumplir los requisitos iniciales con mayor rapidez.

1.5. Recursos empleados

Los recursos empleados en el desarrollo del proyecto se detallan a continuación, distinguiendo entre recursos hardware y software.

Recursos hardware

- **LPC1769 LPCXpresso**

Placa principal del sistema formada por la placa del microcontrolador Cortex-M3 combinada con un programador/depurador o JTAG. Incluye, entre otros, 64kB SRAM, 512 kB Flash, 4 UART, 3 I2C, SPI, ADC de 12 bits de resolución y 8 canales, DAC de 10 bits, 4 timers de propósito general y 70 GPIO.

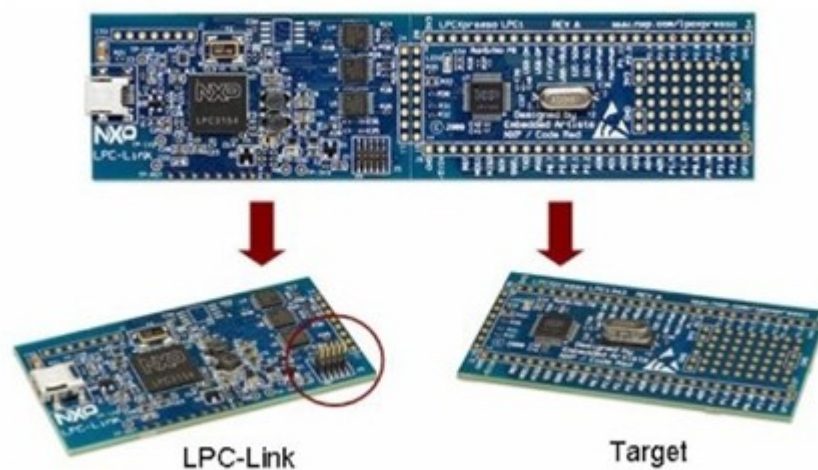


Figura 3: LPC1769 LPCXpresso board

- **RN-XV WiFly 802.11 b/g**

Módulo de comunicaciones inalámbricas con microcontrolador de 32 bits, consumo de 4uA en modo *sleep*, 38mA en modo activo, que requiere para una configuración básica cuatro pines: 3v3, RX, TX y GND para crear una conexión Wi-Fi.

- **Fotorresistencia LDR**

Sensor de luminosidad consistente en una resistencia que varía su valor en función de la luz recibida. A mayor luz, menor resistencia. Se debe tener en cuenta el tiempo de respuesta de la resistencia cuando cambia la luz que recibe, que además es diferente de luz a oscuridad que de oscuridad a luz. Para utilizarla en la placa se debe hacer un divisor de tensión.



*Figura 7:
Fotorresistencia LDR*

- **Leds, botones, zumbador**



Figura 8: Leds, botón, zumbador

- **Elementos varios**

Multímetro, placa de inserción, regulador de tensión 3v3 y 5V, resistencias, condensador, diodo, pilas, cables.

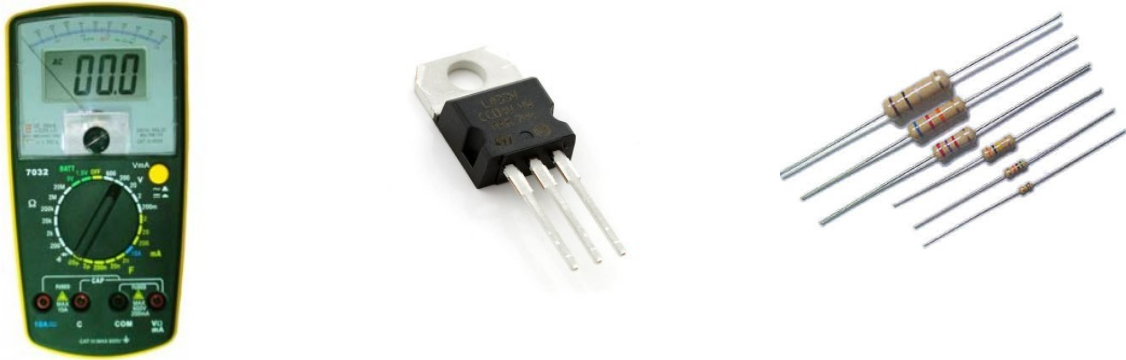


Figura 9: Componentes varios

Recursos software

- Entorno de desarrollo LPCXpresso IDE basado en Eclipse
- Sistema operativo en tiempo real FreeRTOS
- Lenguaje de programación C
- Terminal de comunicaciones TeraTerm
- Aplicación web con PHP

1.6. Productos obtenidos

Tras la conclusión del proyecto se han obtenido principalmente dos productos:

- Sistema empotrado formado por la placa LPC, módulo WiFly, sensores, leds, zumbador y botones, y la aplicación FreeRTOS.

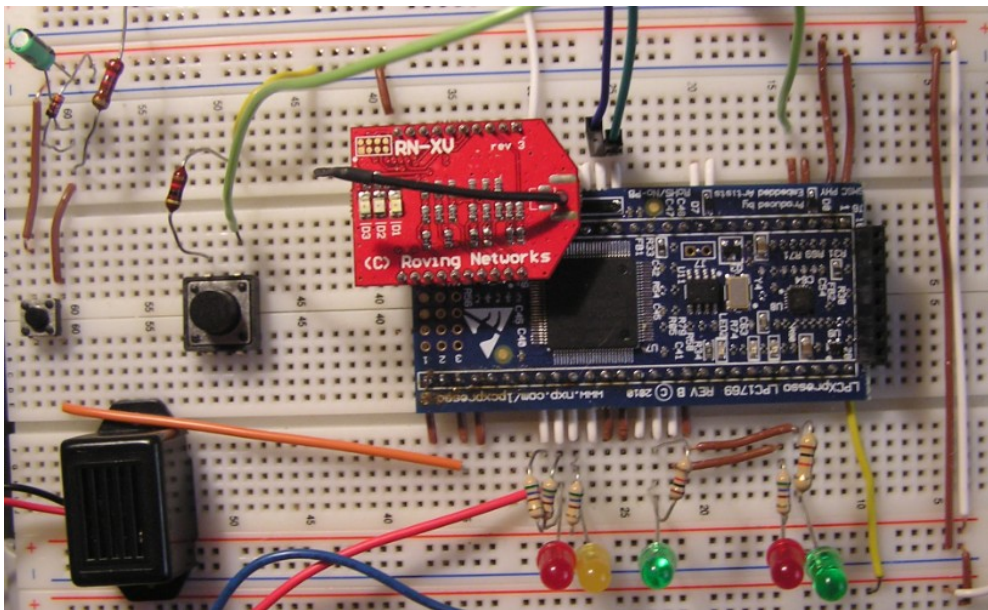


Figura 10: Producto obtenido

- Interfaz de usuario o aplicación web para la comunicación con el sistema, tanto para ver los resultados de las mediciones como para modificar los datos de configuración.

1.7. Breve descripción del resto de capítulos de la memoria

En el siguiente capítulo se describe el funcionamiento y diseño de las partes que componen el proyecto. En el capítulo 3 se detallan los aspectos técnicos de la aplicación en el sistema empotrado. En el capítulo 4 se presentan algunas aplicaciones prácticas. En el capítulo 5 y 6 se analiza la viabilidad técnica y valoración económica. Por último, en el capítulo 7 se extraen las conclusiones.

2. Descripción funcional

2.1. Sistema total

A continuación se describe el funcionamiento básico del sistema y como interaccionan los diferentes objetos que lo forman.

2.1.1 Diagrama de bloques de la aplicación

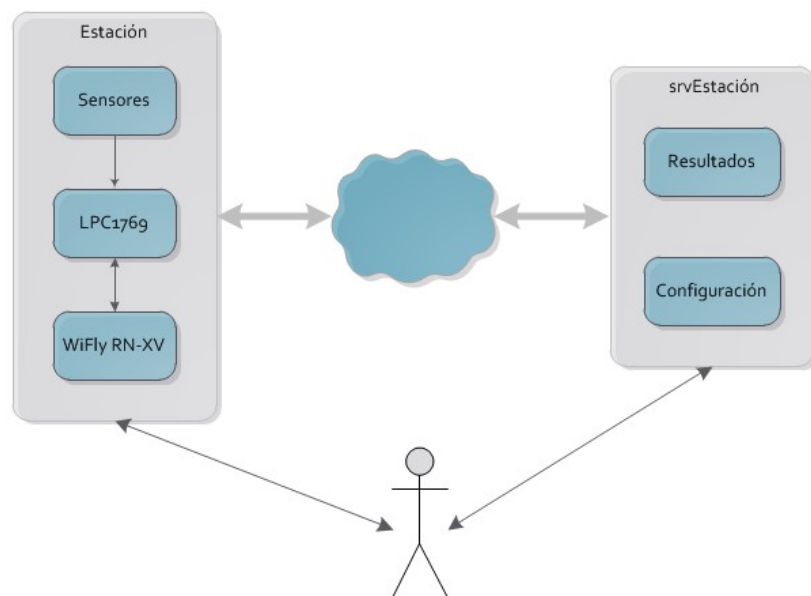


Figura 11: Diagrama de bloques del sistema

Como se puede observar en la figura 11 el sistema comprende dos aplicaciones con las que el usuario se puede comunicar, bien directamente, bien a través de un navegador.

- **Sistema empotrado o Estación**

La interacción entre el usuario y el sistema es directa y se efectúa mediante periféricos de entrada o salida:

- Leds de información de las diferentes funcionalidades, de aviso de alarma, de aviso de fallo del sistema, etc.
- Botón de reset, botón de medición instantánea

- **Aplicación WEB o srvEstación**

El usuario se comunica con la aplicación a través de un navegador web. Esta aplicación permite:

- Visualizar las mediciones de los sensores
- Configurar la estación según sus necesidades

2.1.2 Comunicación e interacción de los diferentes objetos del sistema

Como se aprecia en el diagrama de bloques de la figura anterior, la comunicación entre los dos componentes principales del sistema es **inalámbrica**. Esta comunicación se realiza por medio del módulo WiFly, el cual debe asociarse a un punto de acceso de la red local para establecer la conexión a Internet. Una vez establecida la conexión Wi-Fi, la aplicación del sistema empujado realiza una petición HTTP utilizando el método GET para enviar o solicitar datos de la aplicación web.

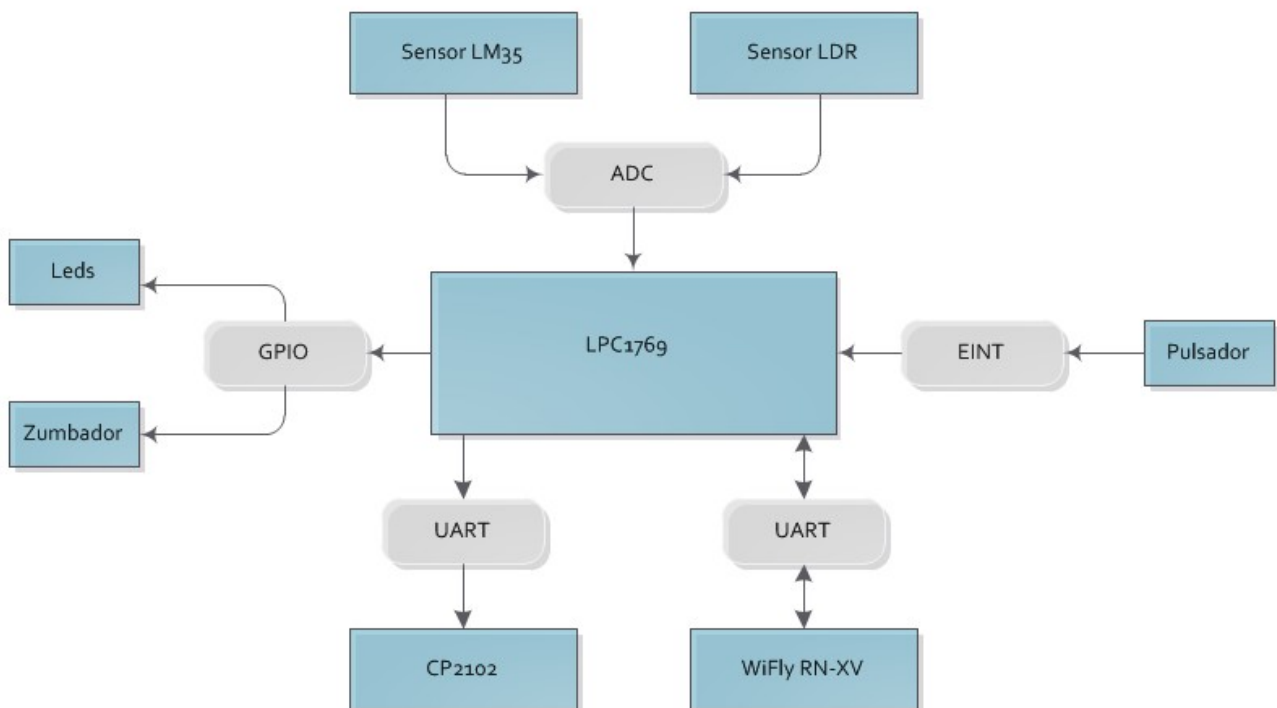


Figura 12: Diagrama de bloques del sistema empujado

La comunicación de los diferentes dispositivos del sistema empotrado se realiza **por cable** a través de los puertos y periféricos internos disponibles en la placa LPC. La estructura de la conexión se muestra en el diagrama anterior.

Se describen a continuación los elementos principales del sistema:

UART – Universal Asynchronous Receiver-Transmitter

La UART convierte la información del formato paralelo al serie y añade información adicional (bits de paridad, parada...) para establecer un protocolo de comunicación, normalmente utiliza la interfaz RS-232.

El LPC1769 dispone de cuatro UART que controlan los puertos serie.

Se han utilizado dos UART, una para la comunicación con el módulo WiFly y la otra para la comunicación con el terminal TeraTerm, que ha sido muy útil para obtener información y para la depuración del código. El esquema de conexiones de ambas UART se muestra en la siguiente figura:

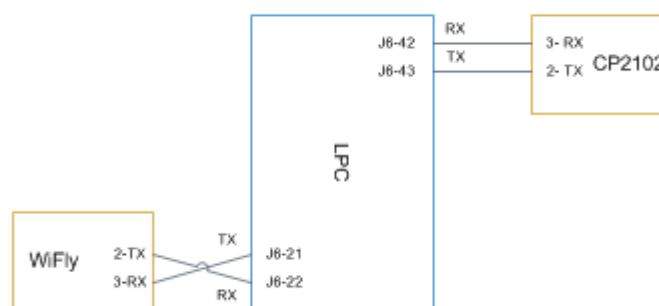


Figura 13: Conexiones UART

En la comunicación serie se utiliza la línea TX para transmitir y la línea RX para recibir, de ahí la conexión WiFly – LPC mostrada en la figura. Por el contrario, en el dispositivo CP2102, el pin 2, TX, corresponde con la transmisión del equipo a conectar, lo mismo ocurre con el pin 3, RX.

ADC – Analog-to-digital converter

Convierte la señal de entrada analógica en una señal digital. El LPC1769 dispone de un ADC de 12 bits de resolución y 8 pines de entrada. Se han utilizado dos pines para conectar los sensores de temperatura, canal AD0.0, y luminosidad, canal AD0.1, como se muestra de forma simplificada en el siguiente esquema:

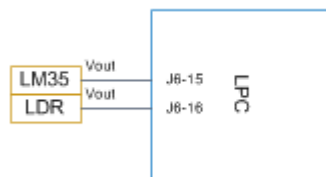


Figura 14: Conexiones ADC

GPIO – General Purpose Input/Output

Son pines digitales de uso general agrupados en puertos que se pueden configurar como entrada o como salida, obteniendo una entrada lógica o generando una salida lógica.

Se han utilizado varios GPIO conectados a diferentes leds y a un zumbador para simular los estados de sistema.

EINT – External Interrupt

Genera una interrupción externa y puede utilizarse para despertar el procesador de los modos de bajo consumo: Sleep, Deep-sleep o Power-down. La placa LPC dispone de cuatro entradas EINT.

Se ha utilizado para conectar el botón y generar una interrupción cuando éste es pulsado, con la finalidad de obtener una medición instantánea de los sensores.

2.2. Diseño de la interfaz de usuario

Se ha creado una aplicación WEB desde la cual el usuario puede interactuar con el sistema. La aplicación consiste en dos páginas: la página de configuración y la página de resultados.

2.2.1 Interfaz de configuración

En la página de configuración, mediante un formulario, se pueden establecer los datos de configuración del sistema. En la siguiente figura se puede ver el formulario.

The image shows a web-based configuration form with the following sections and fields:

- Climatización**
 - Activo
 - Temperatura Día Temperatura Noche
- Alarma**
 - Activo
 - Temperatura alarma
- Riego**
 - Activo
 - Luminosidad Día Luminosidad noche
 - Duracion riego
- Periodos de muestreo**
 - Configuración Temperatura Luminosidad
- Sensores activos**
 - Temperatura Luminosidad

At the bottom of the form is a button labeled "Enviar consulta".

Figura 15: Interfaz de configuración

La descripción de los diferentes parámetros se muestra en la siguiente tabla.

Datos de configuración	
Periodo de Configuración	Frecuencia a la que se accederá a esta misma página para actualizar los datos
Periodo de Temperatura	Periodicidad a la que se obtendrá el valor de temperatura y será enviado a la página de resultados
Periodo de Luminosidad	Periodicidad a la que se obtendrá el valor de luminosidad y será enviado a la página de resultados
Temperatura Activo	Permite activar o desactivar la lectura y envío de la temperatura
Luminosidad Activo	Permite activar o desactivar la lectura y envío de la luminosidad
Climatización	
Temperatura Día	Umbral de temperatura deseado durante el día
Temperatura Noche	Umbral de temperatura deseado durante la noche
Alarma	
Temperatura Alarma	Umbral de temperatura a partir del cual hay riesgo de incendio
Riego	
Luminosidad Día	Umbral de luminosidad a partir del cual se considera que es de día. También usado para la hora deseada de riego, normalmente el amanecer
Luminosidad Noche	Umbral de luminosidad por debajo del cual se considera que es de noche
Duración riego	Tiempo deseado de duración del riego diario

Los parámetros: Luminosidad Día y Luminosidad Noche se utilizan tanto en el sistema de riego como en el de climatización.

2.2.2 Interfaz de resultados

En la página de resultados se mostrarán los valores recibidos de los sensores así como la fecha en la que se obtuvieron, como se muestra en la siguiente figura.

Temperatura: 21.11 C	03-06-2013 15:02:01
Luminosidad: 47.71 lx	03-06-2013 15:01:57

Figura 16: Resultados de la lectura de los sensores

Los datos recibidos se almacenan en un archivo, de esta forma, si un sensor está desactivado se mostrará el último valor almacenado y la fecha en la que se hizo la medición.

2.2.3 Funcionamiento de la aplicación WEB

El funcionamiento de la aplicación es el siguiente: el usuario accede a la página de configuración, userConfig.php, para establecer los parámetros de configuración deseados. El módulo config.php lee los datos y los almacena en el archivo config.txt. Desde el sistema se obtiene la configuración accediendo al módulo config.php, enviando una petición HTTP con la siguiente estructura:

GET config.php HTTP/1.1 “nombre del servidor”

Por otro lado, el sistema actualizará la página de resultados, estacion.php, enviando una petición HTTP mediante el método GET. En este caso existen tres posibilidades, dependiendo de los datos que se quieran enviar, pudiendo enviar solamente la temperatura, la luminosidad o ambas.

GET estacion.php?tmp=X&lum=Y HTTP/1.1 “nombre del servidor”

GET estacion.php?tmp=X HTTP/1.1 “nombre del servidor”

GET estacion.php?lum=X HTTP/1.1 “nombre del servidor”

La página estación.php se encargará de interpretar la petición, mostrar los valores de temperatura y luminosidad, y de almacenarlos en el archivo resultado.txt. Si alguno de los parámetros no se ha enviado, obtendrá el último valor medido desde el archivo resultado.txt y lo mostrará en la interfaz de resultados.

En el siguiente diagrama de bloques se representa este comportamiento.

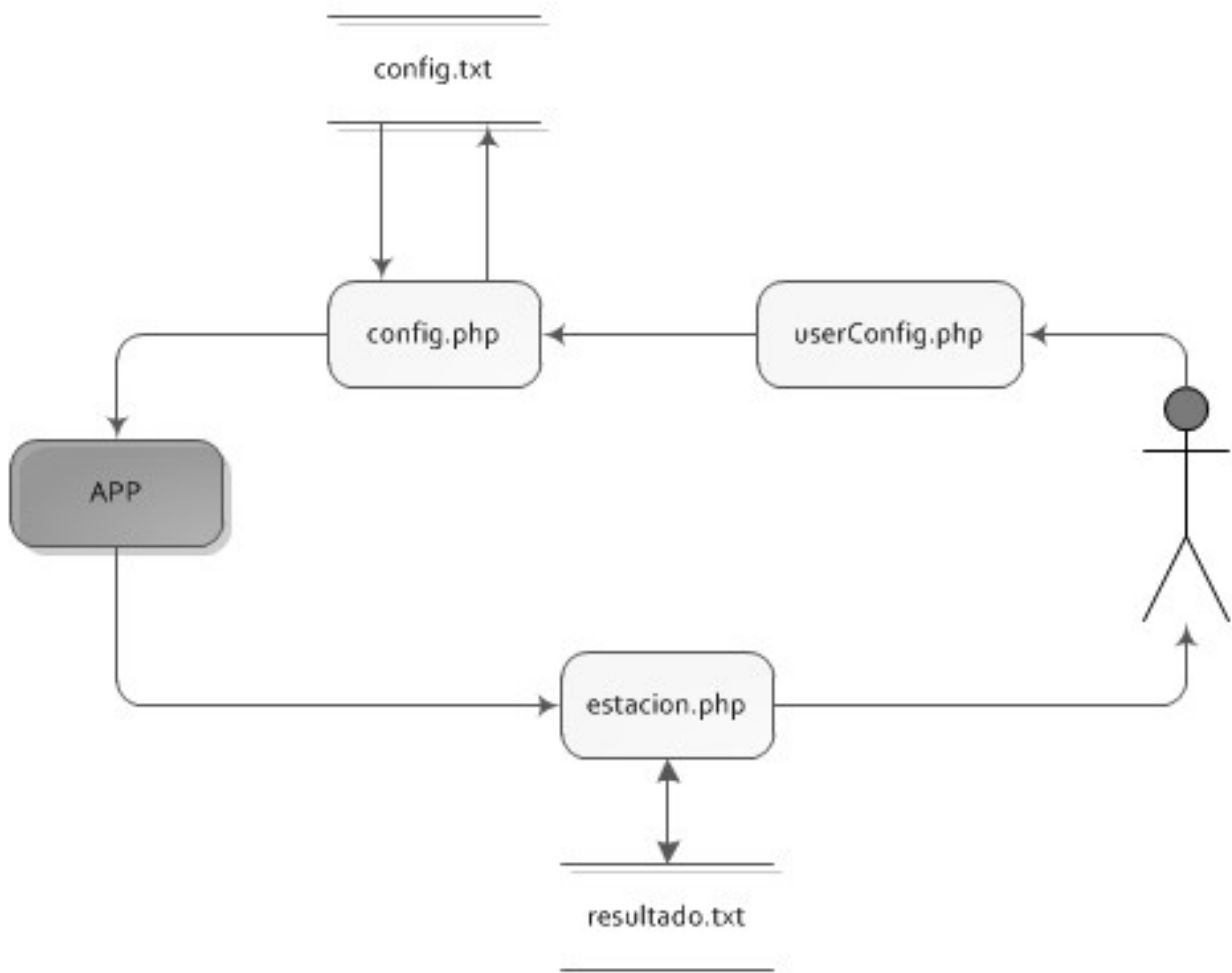


Figura 17: Diagrama de bloques de la interfaz Web

2.3. Diseño de la aplicación empotrada

Como se aprecia en la imagen siguiente, la aplicación desarrollada en el sistema empotrado está dividida en dos grandes módulos. Por un lado, la librería con los controladores de los dispositivos hardware y por otro lado, la aplicación principal. Cada módulo a su vez se descompone en submódulos logrando de esta forma eliminar interdependencias entre ellos y facilitando la reusabilidad.

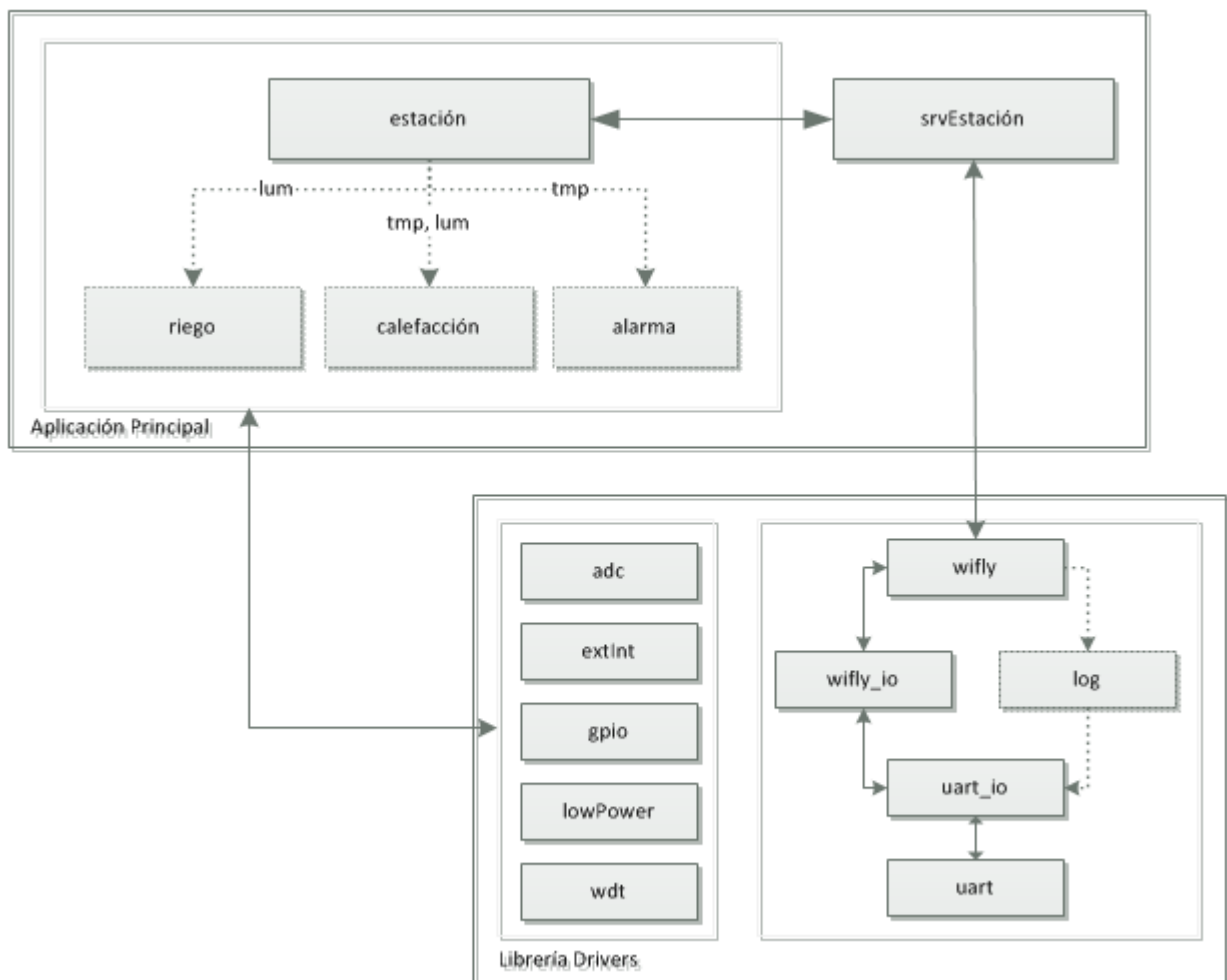


Figura 18: Diseño de la aplicación empotrada

2.4. Esquema de conexiones

La placa LPCXpresso LPC1769 (LPC-Link y microcontrolador) aunque se alimenta con un voltaje de 5v dispone de un regulador de tensión que suministra 3v3 al microcontrolador LPC1769. Uno de los criterios para la elección de los dispositivos externos ha sido este voltaje de trabajo.

En la fase de desarrollo y depuración, la placa estará conectada al puerto USB del ordenador y los dispositivos externos al pin J6-28 del microcontrolador que proporciona un voltaje de 3v3. Una vez el producto terminado, los dos componentes de la placa se separan y el microcontrolador podrá ser alimentado a través del pin J6-28. De esta forma, una fuente de alimentación de 3v3 alimentará todo el sistema.

El esquema de conexiones de los componentes principales se muestra en la siguiente figura.

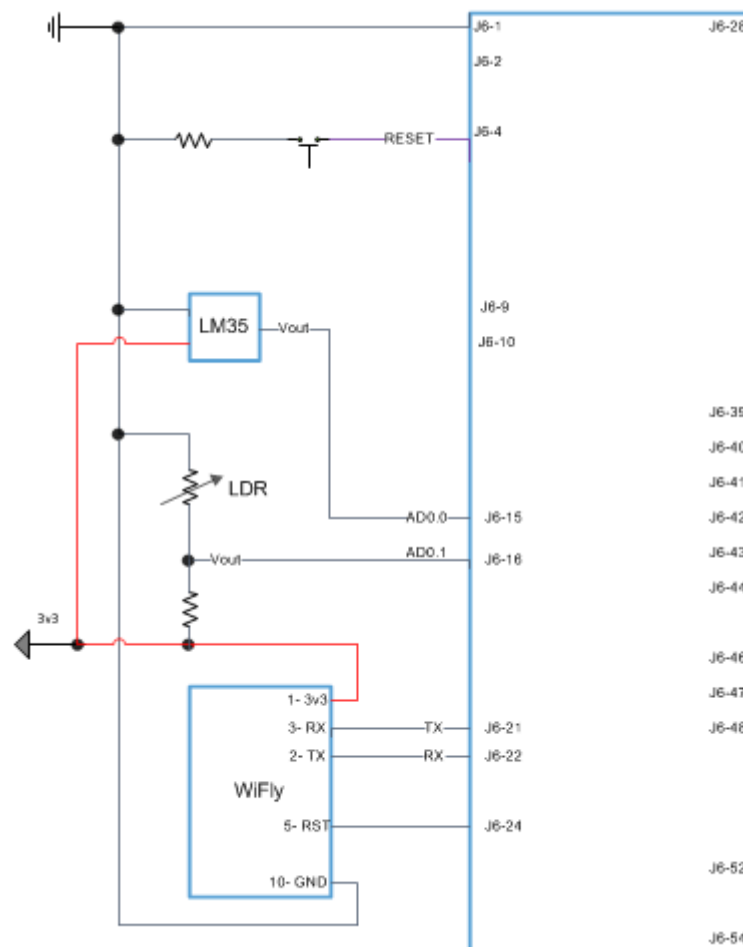


Figura 19: Esquema de conexiones del SE

3. Descripción Aplicación Sistema Empotrado

Tal como se ha visto en el capítulo 3, la aplicación del sistema empotrado está formada por dos módulos: la librería de drivers y la aplicación principal. En los siguientes apartados se describen de forma detallada estos módulos.

3.1. Aplicación Principal

La ejecución de la aplicación comienza con la inicialización de los componentes hardware: uart, wifly, adc, gpio, eint1, watchdog; y los componentes software: tareas, colas, semáforos. A continuación el planificador gestiona las tareas para que lleven a cabo su función. Estas funciones se describen a continuación.

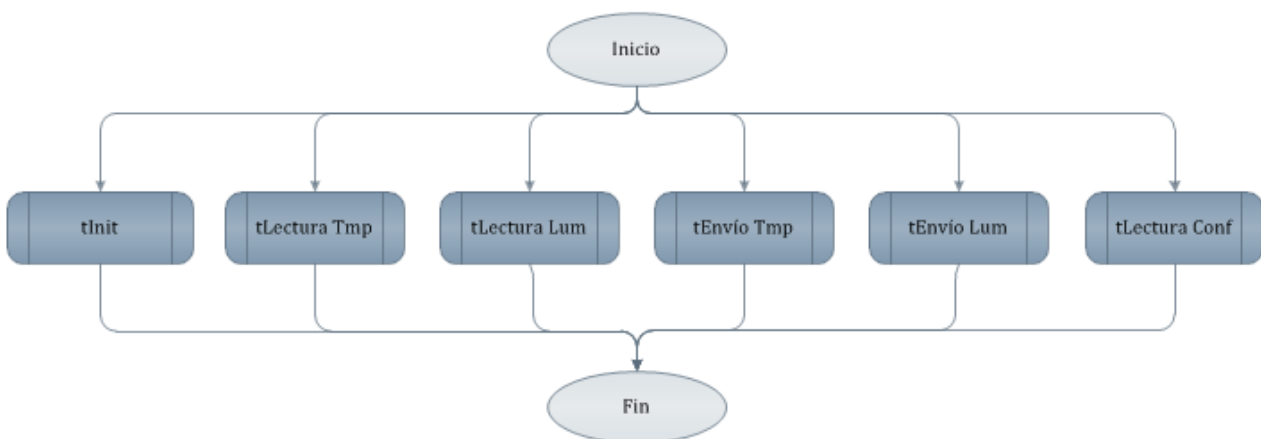


Figura 20: Función main

Lectura de los datos de configuración de la Web y actualización de la configuración del sistema

La tarea que realiza esta función es tLecturaConf. Accede a la Web para obtener los datos de configuración introducidos por el usuario y se ocupa de activar o desactivar las tareas de lectura de los sensores en función de los datos obtenidos, tal como se muestra en el siguiente diagrama de flujo.

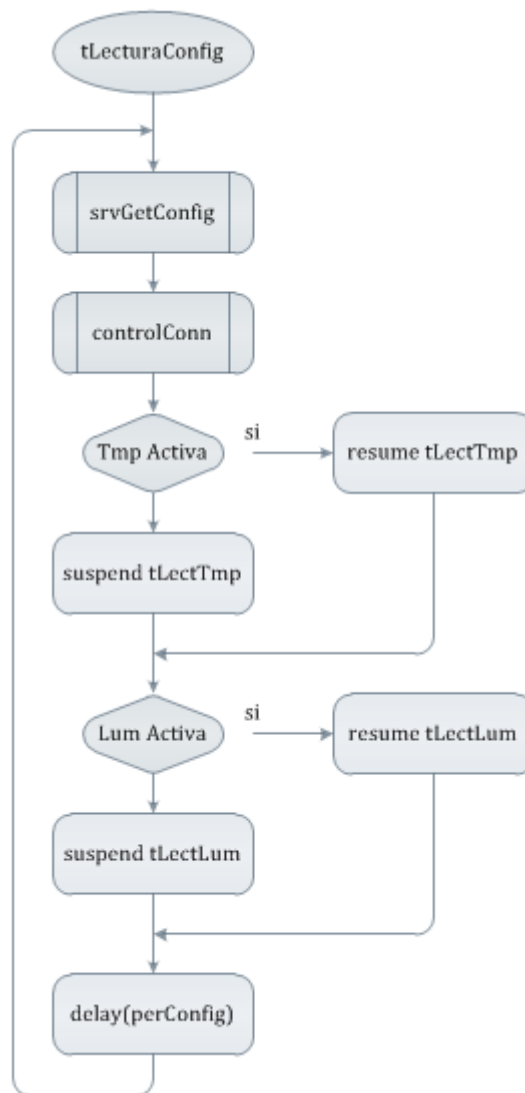


Figura 21: Diagrama de flujo del control de la configuración

Otros datos de configuración como la frecuencia de actualización de los datos, y la frecuencia de lectura y envío de los valores de los sensores se restablecen con los nuevos valores al llamar a la función *delay*.

Esta tarea, al igual que el resto de tareas que acceden a la Web, llama a la función *controlConn* que comprueba si la conexión es correcta.

Lectura de los sensores y envío a la Web

Esta función se puede llevar a cabo de forma periódica o mediante una interrupción hardware, en este caso se realiza una lectura instantánea.

- **Lectura periódica**

La lectura de los sensores de temperatura y luminosidad se efectúa de forma independiente, de esta forma, el usuario puede desactivar uno, los dos o ninguno de los sensores, o asignarles una frecuencia de lectura individual. En cualquier caso, la implementación es similar en ambos sensores y consiste en ejecutar de forma periódica las siguientes operaciones:

- Leer el valor del sensor y guardarlo en una cola
- Leer el valor de la cola y enviarlo a la Web

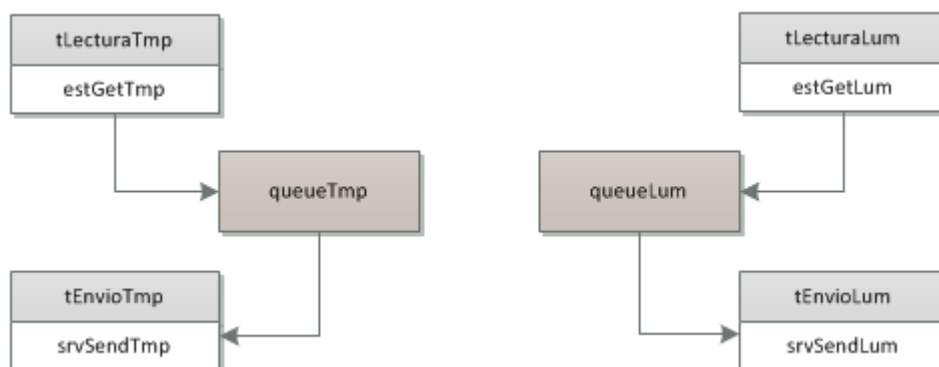


Figura 22: Lectura y envío de los valores de los sensores de forma periódica

- **Lectura instantánea**

La lectura instantánea se realiza mediante un pulsador conectado al pin EINT1, que genera una interrupción cuando detecta un flanco en la entrada. El manejador de la interrupción envía un *flag* a una cola para avisar que se ha pulsado el botón. La tarea *vTaskLecturaInstantanea* permanece bloqueada hasta que la cola recibe el aviso. En ese momento, *vTaskLecturaInstantanea* lee los sensores y envía los valores a las colas de temperatura y luminosidad, tal como se aprecia en la siguiente figura, a la que se le ha

añadido la lectura periódica para mostrar la interacción de los dos tipos de lectura.

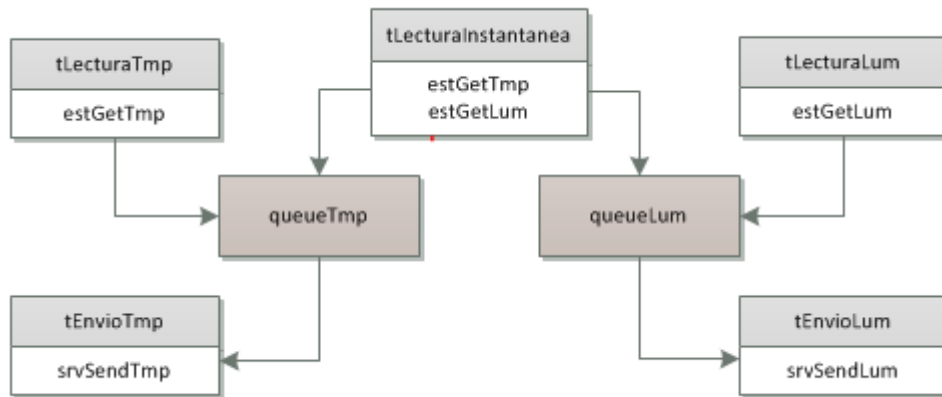


Figura 23: Lectura periódica e instantánea de los sensores

El botón es un elemento mecánico que al pulsarlo produce un efecto llamado rebote, lo que se traduce en varias interrupciones consecutivas en pocos milisegundos. Para suavizar este efecto se ha construido un circuito eléctrico con un filtro RC como se muestra en la siguiente figura.

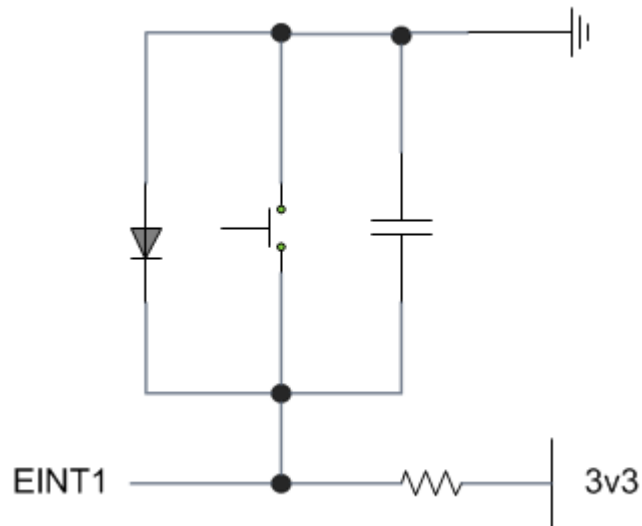


Figura 24: Filtro RC

Control de la conexión

Todas las tareas que necesitan acceso a Internet llaman a la función *controlConexión*. En esta función se comprueba si la conexión está funcionando y en caso negativo se resetea el módulo WiFly. El diagrama de flujo del funcionamiento del control de la conexión se presenta en la figura 25.

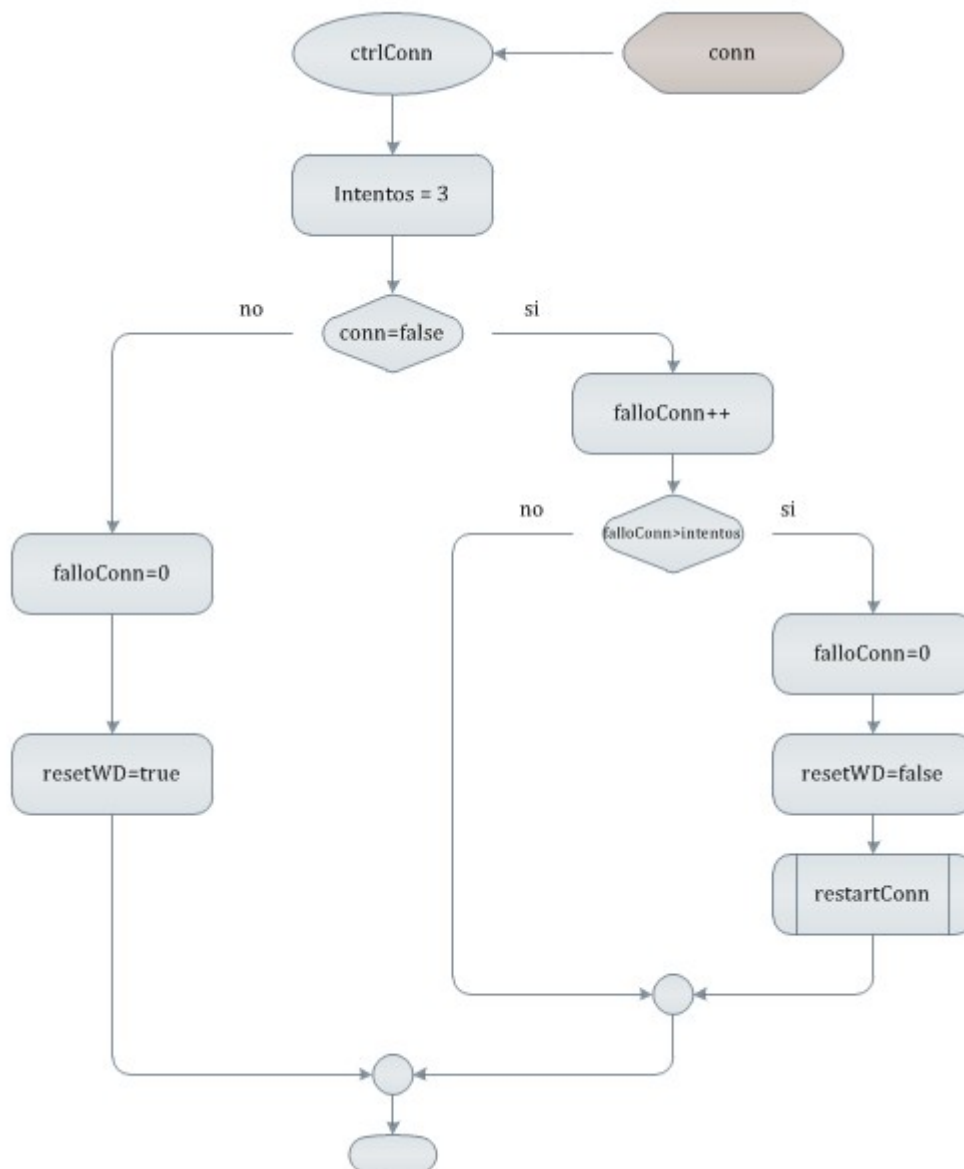


Figura 25: Diagrama de flujo del control de conexión

Se estima que se ha perdido la conexión después de cuatro intentos consecutivos con resultados erróneos. Después de resetear el módulo WiFly, la aplicación sigue su ejecución normal, de forma que, si la conexión no se ha restablecido, se detectará el fallo nuevamente, otra vez se reseteará el módulo WiFly, y así sucesivamente. Si finalmente la conexión no se restablece actuará el *watchdog* como se explica en el siguiente apartado.

Watchdog

Provoca un reset de la placa en el caso de que ésta se haya bloqueado o en el caso de que no haya sido posible restablecer la conexión. La tarea `vTaskWD` actualizará el contador del *watchdog* a su valor de inicio antes de que provoque el reset. Si el sistema se bloquea o no hay conexión a Internet, el contador del *watchdog* llegará a 0 y reiniciará el sistema.

Una función añadida dependiente del *watchdog* consiste en comprobar si cuando la aplicación se inicia, ha sido a causa de un fallo (reset provocado por el WD) o se trata de un reinicio normal. Un led rojo parpadeante durante unos segundos indicará que la aplicación ha sido reiniciada por el WD. Y un led verde indica que el reinicio es correcto.

Ahorro de energía

El microcontrolador LPC1769 está equipado con cuatro modos de bajo consumo que dan flexibilidad al diseño de las aplicaciones obteniendo mayor duración de la batería. Estos modos son: *sleep*, *deep-sleep*, *power-down* y *deep power-down*.

En este trabajo se ha implementado el modo *sleep*. En este modo, se suspende la ejecución de las instrucciones hasta que o bien se produce un reinicio u ocurre una interrupción. Los periféricos siguen funcionando y pueden generar interrupciones para hacer que el procesador reanude su ejecución.

Otros métodos para reducir el consumo de energía son:

- Reducir la frecuencia del reloj de los periféricos´
- Si no se necesita, desactivar el circuito BOD: reinicio de la placa si la alimentación no es estable.

- Desactivar los periféricos no usados. Por defecto, muchos de los periféricos del microcontrolador están activados produciendo un consumo de energía innecesario.
- Desconectar el divisor de reloj del USB desde PLL0 y conectarlo a PLL1
- Reducir consumo de los pines: configurar los pines como GPIO, desactivar los pull-ups internos, configurarlos como salidas y ponerlos a '0'.

Cuando no hay ninguna tarea en estado READY, el planificador ejecuta la tarea *idle*, por tanto, esta tarea, es el lugar idóneo para situar al procesador en modo de bajo consumo. Para añadir funcionalidad a la tarea *idle*, ésta llama a la función *idle hook* que tiene varios usos, entre ellos, situar el procesador en modo de bajo consumo, obteniendo así un método automático de reducción de energía cuando no hay tareas preparadas para entrar en ejecución.

El inconveniente de este método es que la tarea *idle* llama periódicamente a la función *idle hook*, provocando que el procesador salga y entre del modo *sleep* repetidamente. La periodicidad a la que la tarea *idle* llama a la función *idle hook* depende de la frecuencia a la que se generan los *ticks*: interrupción periódica usada por el planificador de tareas. Esta frecuencia es configurable. Se ha reducido a 100 Hz, así las tareas tendrán un tiempo de ejecución de 10 milisegundos antes de ser interrumpidas por una tarea de mayor prioridad, y se producirán 100 *ticks* por segundo.

Se han realizado mediciones de consumo utilizando los métodos mencionados en esta sección, obteniendo un consumo de 96mA en modo de ejecución normal y 75mA en modo de ejecución *sleep*, con alimentación de la placa de 3.3V.

Botón Hard-Reset

Se ha implementado un botón que conectado al pin $\overline{\text{RESET}}$ reinicia el microcontrolador.

3.2. Librería de controladores de dispositivos

En este apartado se detallan los módulos de la librería que contiene los controladores de dispositivos, sus funciones principales y la interfaz de comunicación.

uart.h	
Controla los puertos serie UART0, UART1 y UART3	
Funciones	
UARTInit	
Parámetros de entrada	Puerto UART
	Baud rate o velocidad de transmisión
Configura el puerto UART: establece la velocidad de transmisión, selecciona el reloj, configura los bits de sincronización (longitud, paridad, parada), configura los pines con la función de TX y RX e instala el controlador de interrupción.	
UARTSend	
Parámetros de entrada	Puerto UART
	Mensaje a enviar
Envía el mensaje por el puerto serie	

adc.h	
Transforma el voltaje recibido por cualquiera de los canales de entrada del ADC en un número binario de 12 bits	
Funciones	
ADCInit	
Parámetros de entrada	Frecuencia de reloj del ADC
Inicializa los ocho canales del ADC, habilita el reloj y establece su frecuencia	
ADCRead	
Parámetros	Número del canal
Lee el canal, realiza la conversión analógica-digital y devuelve el resultado	

gpio.h	
Control de los pines de entrada-salida de propósito general GPIO	

Funciones	
gpioSetDir	Selecciona la dirección del pin
gpioSetValue	Fija un 1 en el pin
gpioClear Value	Fija un 0 en el pin
gpioInvert	Invierte el valor del pin

extInt.h	
Controlador de las interrupciones externas EINT0, EINT1, EINT2 y EINT3	
Funciones	
EINT1Init	
Inicializa el pin EINT1 para que detecte una transición de un 1 a un 0 o flanco de bajada, e instala el controlador de interrupción	

wdt.h	
Control del Watchdog. Mecanismo que reinicia el sistema si su contador llega a 0	
Funciones	
WDTInit	
Parámetros de entrada	Tiempo antes de que el WD reinicie el sistema si no ha sido refrescado
Inicializa el timer del WD e instala el controlador de interrupción	
WDTFeed	
Refresca el timer del WD	

3.2.1. Módulo wifly.h

Este driver juega un papel importante en el sistema ya que tiene la tarea de controlar el módulo de comunicaciones WiFly encargado de proveer conectividad Wi-Fi.

El dispositivo WiFly puede estar en dos modos: comando o datos. En el modo comando se pueden establecer o consultar los parámetros de configuración. Y en el modo datos, puede aceptar o iniciar conexiones.

Los comandos se clasifican en cinco categorías:

- **Set:** Cambia la configuración. Los cambios realizados son permanentes después de grabar.
- **Get:** Obtiene la información almacenada permanentemente.
- **Status:** Obtiene información de estado como tipo de autenticación, rssi, tiempo encendido, etc.
- **Action:** Permite acciones como salir del modo comando, cerrar una conexión, etc.
- **File_IO:** Permite acciones de manejo de archivos como grabar, listar o cargar.

Algunas de estas funciones han sido desarrolladas, las principales se describen a continuación.

- **WiflyInit**
Inicializa el puerto serie conectado al módulo.
- **WiflyReset**
Reinicia por hardware el módulo aplicando un pulso de al menos 160us.
- **WiflyReboot**
Reinicia por software el módulo. El reinicio conlleva la pérdida de datos no almacenados.
- **WiflyConnect**
Asocia el módulo al identificador de red. Dependiendo del modo de seguridad de la red, será necesario indicar el *password*.
- **WiflyConnectTCP**
Configura el módulo para realizar conexiones TCP. Necesita los parámetros: nombre del servidor, puerto e IP.
- **WiflyGetRssi**
Obtiene la calidad de la señal. Cuanto más próxima a 0 mejor calidad. Por debajo de -80 se pueden producir cortes de conexión.

- **WiflySend**

Envía un comando o datos al puerto serie y comprueba que el envío es correcto. Es necesario indicar el tiempo esperado para realizar la operación. Dependiendo de la operación, puede tardar más o menos en realizarla, pudiendo variar entre milisegundos y en algunos casos segundos, como establecer una conexión o reiniciar el módulo.

Siguiendo un enfoque modular, se ha creado el módulo `wifly_io.h` para eliminar la dependencia de `wifly.h` con el puerto serie utilizado. De esta forma, `wifly.h` no necesita conocer el módulo `uart.h` y por otro lado, los módulos que hagan uso de los servicios de `wifly.h` tampoco necesitan saber a qué puerto está conectado el dispositivo, como en el caso de `srvEstacion.h`. Este comportamiento se muestra en la figura 26.

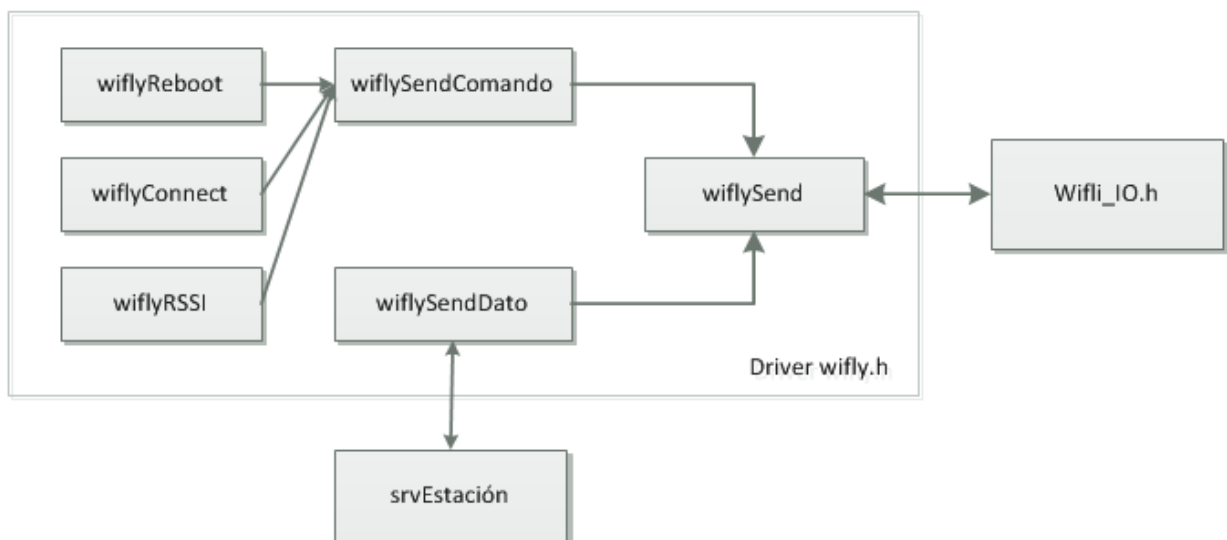


Figura 26: Diseño del módulo `wifly`

El módulo **wifly_io** implementa básicamente dos funciones:

uartSend. Envío de datos a través del puerto serie.

uartScanf. Obtención de la respuesta.

3.2.2. Módulo log.h

Este módulo ha sido de gran utilidad en las fases de desarrollo y pruebas para obtener información de las acciones de la aplicación y para la depuración de código. Es un módulo intermedio en la jerarquía que permite abstraer los módulos superiores del puerto utilizado para enviar información. Mediante un terminal de comunicaciones se puede hacer un seguimiento de las operaciones que realiza la aplicación.

Básicamente las funciones que implementa son:

- **logInit**. Inicializa el puerto serie
- **log**. Recibe como parámetro un texto y lo envía al puerto serie.

Un ejemplo de funcionamiento se muestra en la siguiente imagen.

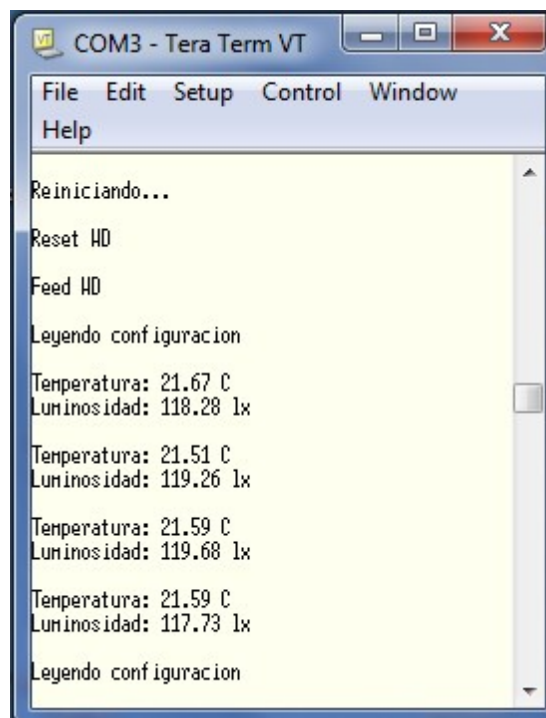


Figura 27: Sistema log

4. Aplicaciones prácticas

La monitorización de parámetros meteorológicos ofrece una amplia gama de aplicaciones. En este capítulo se ofrecen algunas de estas aplicaciones a partir de los sensores implementados y teniendo en cuenta que con sensores apropiados se pueden desarrollar aplicaciones para necesidades específicas.

4.1. Climatización

Esta aplicación gestiona la calefacción de una estancia. A partir de un umbral óptimo de temperatura que puede ser diferente para el día y para la noche, se pretende encender o apagar la calefacción para alcanzar ese umbral.

La calefacción se encenderá si la temperatura actual obtenida del sensor de temperatura es inferior al umbral $- 1$ y se apagará si es superior a umbral $+ 1$. En el rango [umbral-1, umbral+1] la calefacción mantiene su estado, y este rango es el considerado como temperatura idónea.

Los diferentes estados serán simulados mediante leds: un led rojo indica que la calefacción está encendida y un led verde que la temperatura en la estancia es la deseada.

En la siguiente figura se puede observar el esquema de conexiones:

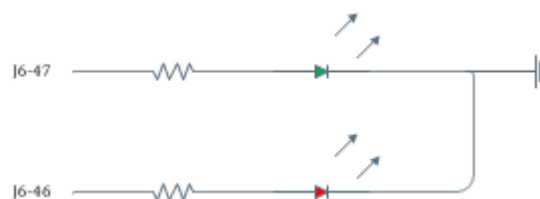


Figura 28: Esquema de conexión de la calefacción

Y el diagrama de flujo:

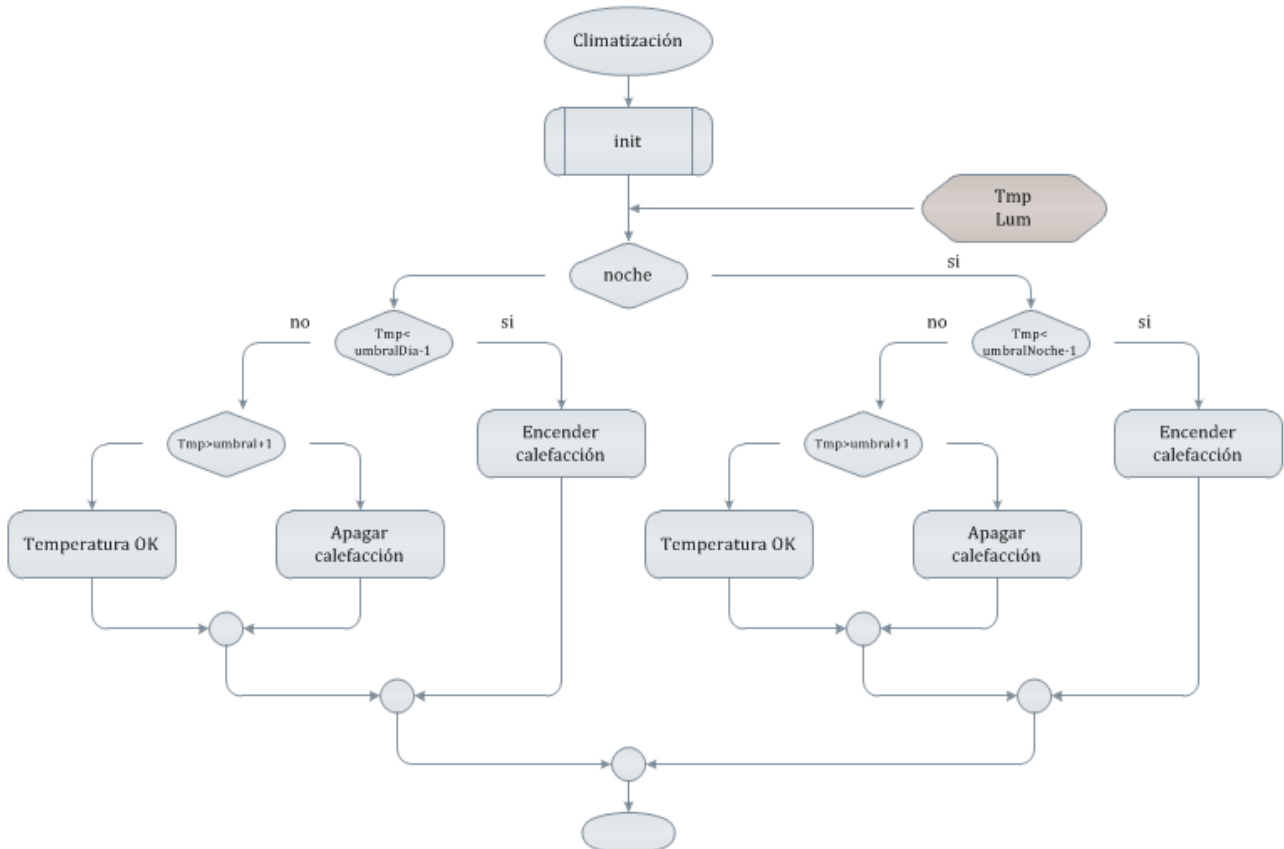


Figura 29: Diagrama de flujo del sistema de calefacción

La función recibe del módulo *estacion* los valores de temperatura y luminosidad.

4.2. Sistema de riego

Se trata de regar un campo, jardín, invernadero, etc. que requiera un riego automático diario basado en la temporización, es decir, en el tiempo de duración del riego.

La fotorresistencia (LDR) se utiliza para detectar la noche, por lo que debe estar colocada fuera del alcance de cualquier lámpara y apuntar al cielo. El riego se realizará por el día, preferiblemente al amanecer, a esa hora hace menos calor y por tanto se pierde menos agua por evaporación. Aun así el momento del día en el que se efectuará el riego es configurable, como lo es también la duración del riego y la medida de luminosidad por debajo de la cual se considera que es de noche.

El riego se simula mediante un led verde que permanecerá encendido mientras se esté regando. Este led se conectará a un puerto de entrada-salida (GPIO) de la placa LPC1769 con su resistencia de protección.

El diagrama de flujo que representa esta aplicación es el siguiente:

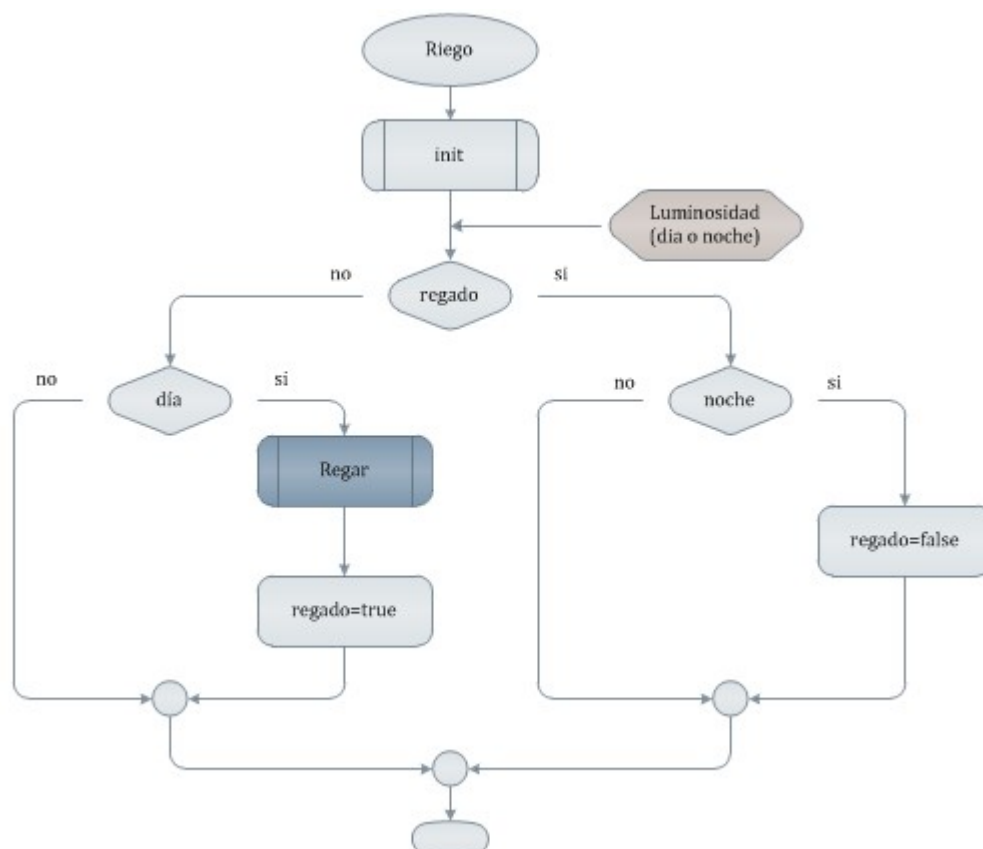


Figura 30: Diagrama de flujo del sistema de riego

4.3. Alarma de incendios

Esta aplicación consiste en activar una alarma cuando el sensor de temperatura alcance el umbral de riesgo de incendio. Un rango de temperaturas inferior al umbral avisarán de “prealarma” mediante un led amarillo y una vez alcanzado el umbral, el parpadeo de un led rojo y un zumbador indicarán la situación de riesgo de incendio.

El esquema de conexiones y el diagrama de flujo se representan en las siguientes figuras:

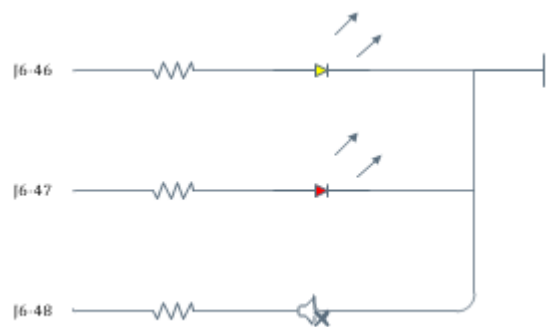


Figura 31: Esquema de conexiones del sistema de alarma

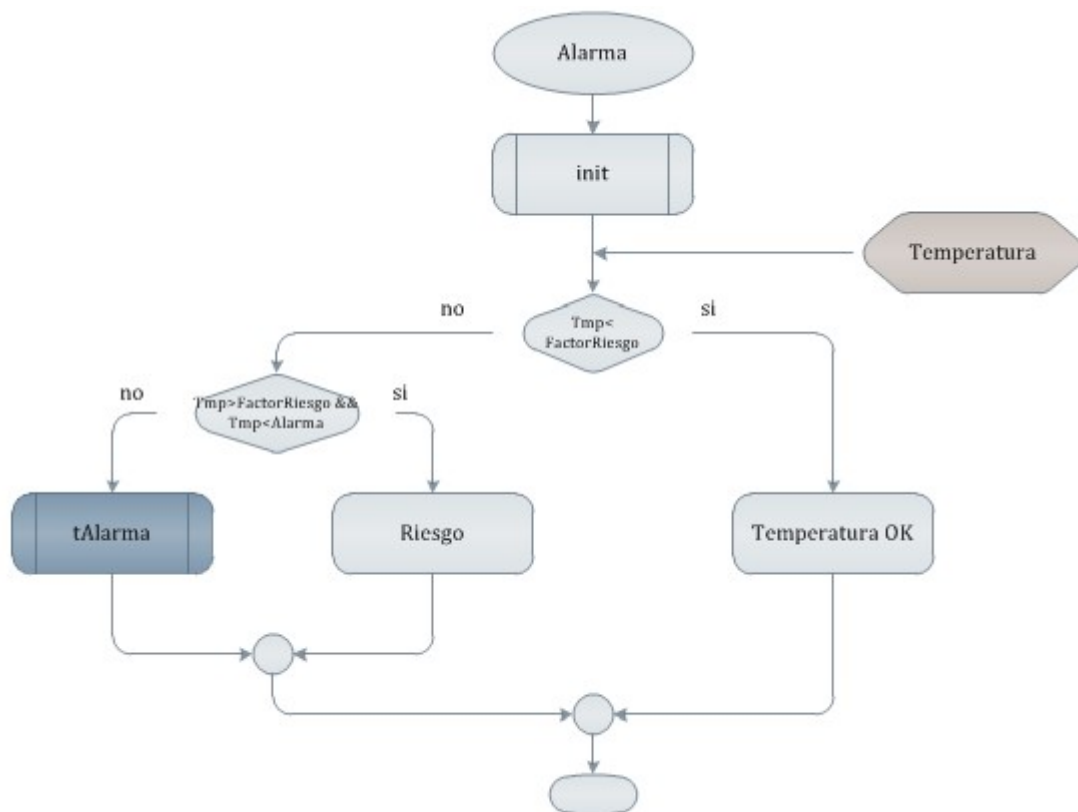


Figura 32: Diagrama de flujo del sistema de alarma

La función recibe la temperatura actual del módulo *estación*. Esta función gestiona la tarea tAlarma, suspendiéndola y reanudándola en función de la temperatura. La tarea tAlarma se ocupa del parpadeo del led rojo y del zumbador mientras la temperatura está por encima del umbral de riesgo de incendio.

5. Viabilidad técnica

Para el análisis de la viabilidad técnica del proyecto se debe tener en cuenta tres factores: los dispositivos que disponíamos inicialmente, microcontrolador y módulo WiFly, los objetivos propuestos y los dispositivos adquiridos para cumplir los objetivos, esencialmente los sensores. Dado que los sensores son compatibles con el microcontrolador y a partir de este conjunto se ha podido desarrollar un sistema que cumple con los objetivos, se puede concluir que el proyecto es viable.

Ahora bien, no se ha desarrollado un producto terminado, sino un objeto diseñado para demostrar un conjunto de funciones o aplicaciones. Dista mucho de ser un producto final. Desde aspectos como el montaje, alimentación o sensores adecuados, son muchos los puntos a mejorar.

Como punto fuerte destacar la escalabilidad. La placa LPC dispone de numerosos periféricos para otros dispositivos. Con facilidad se pueden añadir nuevos sensores y crear aplicaciones atendiendo a los requisitos de los usuarios.

6. Valoración económica

A continuación se presenta una valoración orientativa del coste del proyecto, incluyendo los materiales empleados, desarrollo e instalación.

Concepto	Cantidad	Precio unitario	Total
Placa LPCXpresso LPC1769	1	30,00 €	30,00 €
Módulo Wifly RN-XV	1	32,00 €	32,00 €
CP2102	1	3,00 €	3,00 €
Sensor LM35	1	2,00 €	2,00 €
Sensor LDR	1	2,00 €	2,00 €
Material electrónico	1	20,00 €	20,00 €
Horas de desarrollo	170	12,00 €	2.040,00 €
Instalación	20	12,00 €	240,00 €
		TOTAL	2.369,00 €

Se debe considerar que el proyecto se ha realizado en una fase de formación que ha requerido mucho tiempo de documentación, por tanto la valoración no se puede comparar con un proyecto realizado por un profesional.

7. Conclusiones

Los requisitos mínimos consistían en:

- Adquisición de datos de los sensores (ADC)
- Estudio del consumo energético (multímetro, modo IDLE)
- Envío periódico de los datos a un servidor local
- Conexión a Internet vía Wi-Fi

Se han podido cumplir los requisitos y se han añadido algunas funcionalidades como se aprecia en la planificación final.

Entre las mejoras se propone añadir más sensores para que la estación pueda registrar otras variables meteorológicas como la humedad relativa o velocidad y dirección del viento, y realizar una interfaz de usuario más amigable y completa.

A modo de autoevaluación, este TFC me ha dado la oportunidad de iniciarme en el mundo de los sistemas empotrados y redes de sensores inalámbricos, conocer el funcionamiento básico de los microcontroladores, varias formas de comunicación con otros periféricos, destacando la comunicación inalámbrica.

También me ha dado la oportunidad de utilizar un sistema operativo en tiempo real basado en la planificación de tareas y las diferentes formas de comunicación y sincronismo mediante el uso de colas o semáforos.

8. Glosario

ADC. Conversión analógica digital.

Baudrate. Número de unidades de señal por segundo.

GPIO. Entrada-salida de propósito general.

Microcontrolador. Circuito integrado programable, capaz de ejecutar las órdenes grabadas en su memoria.

Planificador o scheduler. Componente funcional que reparte el tiempo disponible de un microprocesador entre los procesos preparados para su ejecución.

Servidor. Nodo que provee servicios a nodos clientes.

Sistema empotrado. Sistema de computación diseñado para realizar una o algunas pocas funciones dedicadas, frecuentemente en un sistema de computación en tiempo real.

RSSI. Indicador de fuerza de señal de recepción. Escala para medir el nivel de potencia de una señal inalámbrica.

UART. Transmisor/Receptor asíncrono universal. Controla los puertos y dispositivos serie.

Wi-Fi. Mecanismo de conexión de dispositivos electrónicos de forma inalámbrica.

9. Bibliografía

Libros y manuales de usuario:

- Using the FreeRTOS Real Time Kernel. NXP LPC17xx Edition. Richard Barry
- LPC17xx User manual
- Using the LPC1700 power modes
- User Manual and Command References. Wifly GSX/EZX

Recursos Web:

- Web de Sistemas Empotrados de la UOC. <http://cv.uoc.edu/app/mediawiki14/wiki/IniciCortexM3>
- Pagina oficial del sistema operativo FreeRTOS, <http://www.freertos.org/>
- Ejemplos de proyectos LPCXpresso, <http://www.nxp.com/lpcxpresso-support>
- Foro de LPCXpresso - NXP, <http://knowledgebase.nxp.com/>
- Soporte Code Red, <http://support.code-red-tech.com/CodeRedWiki/WikiHome>
- Datasheet TMP35, <http://www.ti.com/lit/ds/symlink/lm35.pdf>
- Datasheet LDR, http://www.biltek.tubitak.gov.tr/gelisim/elektronik/dosyalar/40/LDR_NSL19_M51.pdf
- Manual de PHP, <http://php.net/manual/es/index.php>
- Ansi C, <http://c.conclase.net/librerias>