

## Resumen

Este proyecto se basa en la programación y diseño de un sistema que monitorice la temperatura de diversos puntos que pueden estar alejados entre sí y se visualicen dichas temperaturas en un punto central con conexión inalámbrica entre sensores y pantalla.

Primero de todo ha sido necesario diseñar la estructura del sistema y definir las especificaciones, como el número de sensores, la manera en que se organizará el intercambio de información y cómo el usuario podrá manejar el sistema.

Para que la unidad central se pueda comunicar con todos los sensores sin que aparezcan interferencias se ha multiplexado el tiempo. Es decir, se ha dividido el ciclo de funcionamiento del sistema en diversos espacios de tiempo en el cual en cada uno de ellos se comunicará uno de los sensores con la unidad central.

Posteriormente se han analizado los requisitos de ambas unidades y se ha definido el medio de transmisión de información como ondas de radiofrecuencia. También se han definido el microcontrolador a utilizar y los módulos que contendrán dichas unidades. Con los módulos definidos se han escogido la marca y el modelo de cada componente. PIC18F4520 como microcontrolador, DS18B20 como sensor de temperatura, LCD1602 como pantalla y nRF24L01 como módulo de radiofrecuencia. La elección se ha basado principalmente por su alta disponibilidad en la universidad.

Con el sistema totalmente definido ha sido necesario programar ambas unidades de modo que la primera lea la temperatura y la envíe a la unidad central para que la procese cuando ésta le indique, guarde las máximas y mínimas de cada sensor y las visualice por la pantalla. Para la programación de ambas unidades ha sido necesario utilizar subrutinas previamente creadas para los diversos componentes del sistema.

Es conveniente recalcar que durante la programación han surgido diversos contratiempos como el que la pantalla compartiese puertos con el módulo de radiofrecuencia y haya sido necesario multiplexar la señal y deshabilitar cada componente cuando el otro es el que está siendo usado. Otro inconveniente importante ha sido el de los botones, éstos utilizan unos registros por defecto que ya eran utilizados por la pantalla y han sido necesario utilizar otros y conectarlos mediante cables. Además ha sido necesario controlar los rebotes de dichos botones.

Por último se han calculado las prestaciones finales del sistema, éstas son el consumo, el alcance, el rango de temperaturas, la alimentación, el coste y el impacto medioambiental.



# Índice

<b>RESUMEN</b>	<b>1</b>
<b>ÍNDICE</b>	<b>3</b>
<b>1. OBJETIVOS</b>	<b>5</b>
<b>2. FASES DEL PROYECTO</b>	<b>7</b>
<b>3. ESPECIFICACIONES DEL SISTEMA</b>	<b>9</b>
3.1. Especificaciones de hardware .....	9
3.2. Ciclo de funcionamiento.....	9
3.3. Interfaz de usuario .....	10
<b>4. ESTRUCTURA DEL SISTEMA: HARDWARE Y SOFTWARE</b>	<b>11</b>
4.1. Unidad sensor .....	11
4.2. Unidad central .....	11
4.3. Módulos que integran cada unidad.....	12
4.4. Multiplexado en el tiempo y sincronización.....	12
4.5. Reducción del consumo.....	13
<b>5. ELECCIÓN DE COMPONENTES</b>	<b>14</b>
5.1. PIC18F4520 .....	14
5.2. El sensor de temperatura DS18B20 .....	14
5.3. La pantalla LCD1602 .....	15
5.4. El módulo de radiofrecuencia nRF24L01 RF Board.....	16
5.5. Placa de demostración con PIC18F4520 .....	16
<b>6. PIC18F4520</b>	<b>19</b>
6.1. Características generales .....	19
6.2. Sistema y herramientas de desarrollo .....	20
6.3. Bibliotecas de subrutinas .....	20
<b>7. EL SENSOR DE TEMPERATURA DS18B20</b>	<b>22</b>
7.1. Características generales .....	22
7.2. Medida y conversión de la temperatura.....	22
<b>8. EL MÓDULO DE RADIOFRECUENCIA NRF24L01 RF BOARD</b>	<b>24</b>
8.1. Características generales .....	24
8.2. El bus SPI.....	25

8.3. Comunicación entre unidades.....	26
<b>9. PANTALLA LCD1602</b> .....	<b>28</b>
9.1. Características generales.....	28
<b>10. ESTRUCTURA DEL SOFTWARE DE LA APLICACIÓN</b> .....	<b>30</b>
10.1. Software del nodo sensor.....	30
10.2. Software del nodo central.....	33
10.2.1. Multiplexado de señales.....	37
<b>11. PRESTACIONES DEL SISTEMA FINAL</b> .....	<b>39</b>
11.1. Alimentación.....	39
11.2. Alcance.....	39
11.3. Consumo.....	40
11.4. Rango de temperatura.....	41
11.5. Costes.....	41
11.6. Impacto medioambiental.....	42
<b>12. GUÍA DE USUARIO DEL SISTEMA</b> .....	<b>44</b>
<b>CONCLUSIONES</b> .....	<b>46</b>
<b>AGRADECIMIENTOS</b> .....	<b>48</b>
<b>BIBLIOGRAFIA</b> .....	<b>50</b>

# 1. Objetivos

El objetivo principal del trabajo es la realización del proyecto de final de grado de la universidad. El proyecto consiste en el diseño y la creación de una aplicación real basada en microcontroladores. La aplicación escogida tiene que ser capaz de captar la temperatura de diversos puntos y visualizarlos de manera centralizada. Para ello será necesario desarrollar diversos programas para los diferentes microcontroladores y estudiar las prestaciones finales del sistema.

Una vez conseguido el objetivo principal, se intentará minimizar el consumo y el impacto medioambiental de la aplicación.

Otro objetivo es el de aplicar de manera práctica los conocimientos previamente adquiridos en la asignatura de Electrónica y de manera más específica en la asignatura optativa de "Desarrollo de aplicaciones basadas en microcontroladores". Todo esto además de aprender nuevos conocimientos teóricos de este ámbito.



## 2. Fases del proyecto

Los pasos que se han realizado para desarrollar el proyecto y cumplir los objetivos mencionados en el apartado anterior son:

- Escoger la aplicación a desarrollar.
- Definir las especificaciones que tendrá el proyecto.
- Desarrollar la estructura del sistema.
- Escoger microcontrolador, el PIC18F4520.
- Escoger los módulos necesarios que cumplan con las especificaciones previamente definidas. DS18B20 como sensor de temperatura, nRF24L01 como módulo de radiofrecuencia y LCD1602 como pantalla visualizador.
- Programación de los microcontroladores que harán de sensor y captarán la temperatura.
- Programación del microcontrolador que hará de central. Almacenará y mostrará la información.
- Sincronización de los microcontroladores en el tiempo a modo de comunicar todos los microcontroladores sensores con el microcontrolador central y que éste los reconozca.
- Estudio del alcance, consumo, alimentación, coste, impacto medioambiental y rango de temperaturas del sistema final.





## 3. Especificaciones del sistema

### 3.1. Especificaciones de hardware

La finalidad del proyecto es el diseño de un aparato para medir las variaciones de temperatura en función del tiempo y del lugar. El lugar a estudiar es importante ya que implica unos datos que son necesarios posteriormente, como el rango de temperatura, la cantidad de puntos necesarios, la precisión de las medidas o la distancia mínima a la que tienen que ser capaz de comunicarse las unidades.

La especificación más importante del sistema es el número de puntos de los que se requiera la temperatura. En el caso de este sistema el número escogido es 10. Este número puede variar pero a la hora de hacer las pruebas el número escogido ha sido éste. Cada uno de ellos los denominaremos sensores y se les asociará un número que hará que la unidad central pueda relacionar cada temperatura con el punto del que proviene.

El rango de temperatura de los puntos que queremos estudiar es importante a la hora de elegir el sensor adecuado. En nuestro caso, la aplicación tiene que poder realizar medidas entre los  $-20\text{ }^{\circ}\text{C}$  y los  $+60\text{ }^{\circ}\text{C}$ .

En cuanto a la precisión y a la forma de las medidas, la pantalla tiene que mostrar el valor de la temperatura con mínimo 2 decimales y el número asociado al sensor del cual ha sido tomada esa temperatura.

La distancia mínima a la que los microcontroladores tendrán que poder comunicarse será de 40 metros. Buscando que el sistema no se aleje mas allá de un mismo edificio.

### 3.2. Ciclo de funcionamiento

El ciclo de funcionamiento del sistema será de un segundo, una décima de segundo por cada sensor. Durante este segundo la unidad central irá enviando una señal de activación a cada sensor y recibirá la temperatura del sensor activado. Por lo tanto los sensores tendrán que ir sincronizados a la unidad central y estar preparados para enviar la información en la décima de segundo en la que la unidad central lo requiera. Con esto reduciremos el consumo una décima parte manteniendo durante el resto del tiempo los sensores dormidos, es decir, en bajo consumo.

La medición de temperatura será también de una vez por segundo, mantendrá la información hasta que la envíe a la unidad central y luego la sobrescribirá con la nueva temperatura antes de entrar en modo dormido.

La unidad central almacenará las temperaturas de todos los sensores del ciclo en el que se encuentre, no guardará temperaturas de ciclos pasados para evitar el colapso de memoria.

### **3.3. Interfaz de usuario**

El sistema requiere de una pantalla en la cual se pueda visualizar tanto letras como números. Tiene que ser capaz de albergar un número mínimo de dígitos determinado donde se contenga tanto el sensor del cual proviene la temperatura, el tipo de temperatura ya sea actual, mínima o máxima y dicha temperatura.

Además de la pantalla, será necesario disponer de varios botones para que el usuario pueda cambiar el sensor del cual se está mostrando la temperatura y el tipo de temperatura del sensor mostrado.

## 4. Estructura del sistema: hardware y software

Después del estudio de las diversas posibilidades de diseño del sistema es necesario disponer de unas unidades satélite de captación de temperatura y una unidad central.

### 4.1. Unidad sensor

Dichas unidades se colocarán en los puntos que se requiera conocer la temperatura. Éstas tendrán que captarla y poder transmitirla a la unidad central sin interferirse mutuamente. Esta transmisión tendrá que ser inalámbrica.

### 4.2. Unidad central

Esta unidad será el centro del sistema y tiene que ser capaz de recibir y almacenar la información que envíen todas las unidades sensor. Además será necesario que pueda identificar qué sensor le envía cada temperatura y que almacene la temperatura máxima y mínima de cada sensor. Por último se le requerirá mostrar cualquiera de las temperaturas a petición del usuario del sistema.

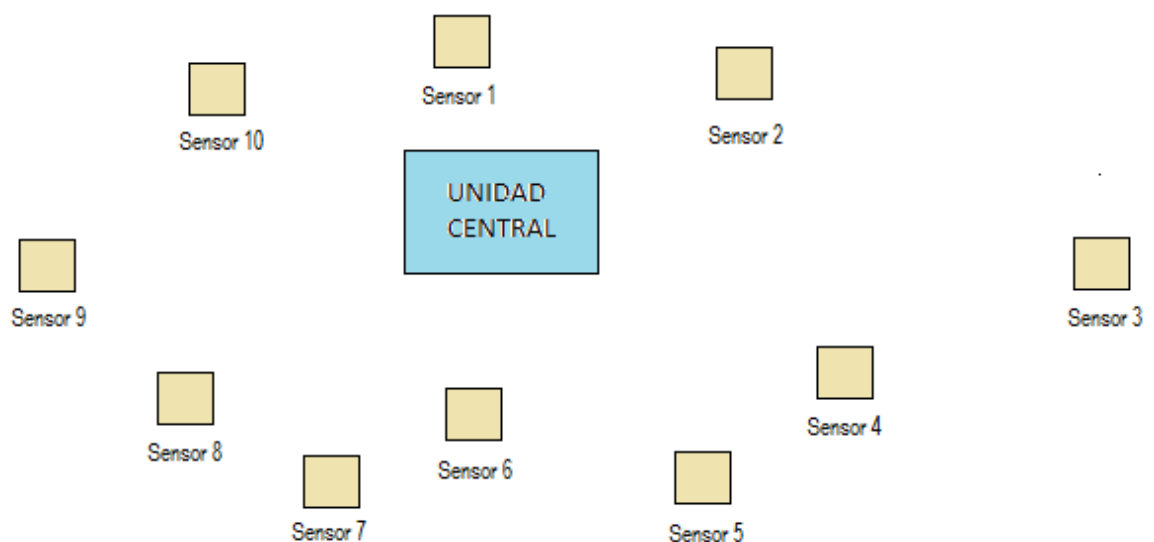


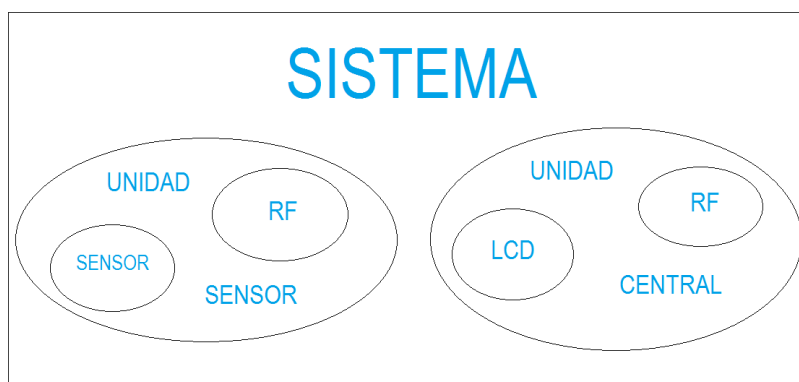
Figura 4.1. Diagrama de la estructura básica del sistema

### 4.3. Módulos que integran cada unidad

La unidad sensor requiere para su función un módulo de captación de temperatura que transforme la señal de temperatura en una señal digital que el microcontrolador pueda interpretar. Además es necesario un módulo de transmisión de información por radiofrecuencia para comunicarse con otros microcontroladores.

La unidad central precisa del mismo módulo de radiofrecuencia para comunicarse y requiere de un visualizador, en este caso una pantalla LCD, para mostrar la información. También requiere como hemos dicho antes, de al menos 2 botones para que el usuario pueda manejar la información almacenada.

Figura 4.2. Esquema básico de la estructura del sistema



### 4.4. Multiplexado en el tiempo y sincronización

Como ya se ha introducido al inicio de este apartado, el sistema consta de muchos sensores que se tienen que comunicar con la unidad central a través de la misma frecuencia. Por tanto, es necesario sincronizar todos los sensores para evitar interferencias entre las mismas. Para ello, se divide el espacio de tiempo de cada ciclo de funcionamiento de un segundo en el número de sensores del sistema. Por tanto, se le dedica a cada sensor una parte de ese espacio para que se comuniquen con la unidad central. Es decir, si el ciclo de funcionamiento de la unidad central en este caso es de un segundo, se le dedica una décima de segundo a cada sensor.

## 4.5. Reducción del consumo

Al dedicar una parte de cada ciclo de funcionamiento a cada sensor, provocas que no sea necesario mantener los módulos de radiofrecuencia encendidos todo el rato. Fuera del espacio de tiempo dedicado a cada sensor lo único que harán los sensores será captar la siguiente temperatura y enviarla a la unidad central y mantenerse inactivos el resto del tiempo con el fin de disminuir en gran medida su consumo de energía.

## 5. Elección de componentes

Con las especificaciones definidas en apartados anteriores se han escogido los componentes necesarios para el desarrollo de la aplicación.

### 5.1. PIC18F4520

El microcontrolador escogido para el desarrollo de la aplicación es el PIC18F4520 de Microchip. Tiene disponibles los periféricos necesarios para todas las unidades y ha sido escogido por su disponibilidad en el departamento de electrónica de la universidad. Tiene también un simulador y emuladores útiles a la hora de probar la aplicación en el proceso de programación.

Otra razón es el conocimiento obtenido sobre este microcontrolador en especial en la asignatura optativa de "Desarrollo de aplicaciones basadas en microcontroladores".

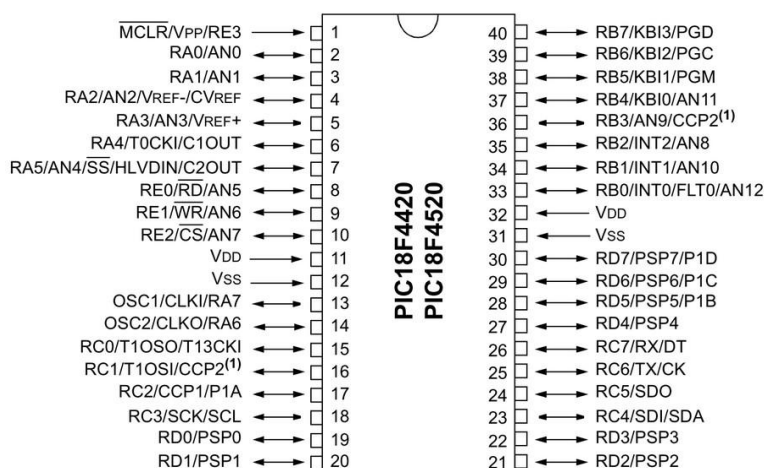


Figura 5.1. PIC18F4520 y sus pines

### 5.2. El sensor de temperatura DS18B20

El modulo escogido para captar la temperatura y transmitirla al microcontrolador es el DS18B20. También está disponible y puede conectarse al microcontrolador escogido.

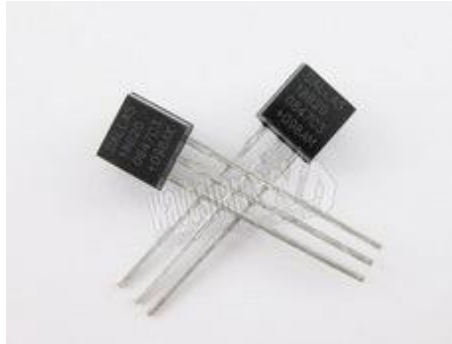


Figura 5.2. Sensor de temperatura DS18B20

### 5.3. La pantalla LCD1602

Dentro de las diversas pantallas que se pueden utilizar junto al microcontrolador como la 7 segmentos, la LCD en color, la LCD de 2 colores y una pantalla de ordenador, la escogida ha sido la de 2 colores. Ésta es la LCD1602. Tiene más dígitos que la de 7 segmentos, la cual no dispone de dígitos suficientes. La de colores no es necesaria para la función requerida y la pantalla de ordenador es demasiado grande para visualizar la información utilizada en la aplicación. Ésta además contiene los dígitos necesarios para el desarrollo de la aplicación.



Figura 5.3. Pantalla LCD1602 utilizada en el proyecto

## 5.4. El módulo de radiofrecuencia nRF24L01 RF Board

Para transmitir la información entre unidades, se ha escogido la transmisión mediante ondas electromagnéticas. Dentro de las diferentes ondas, el módulo que contiene el nRF24L01 de Norton utiliza ondas de radio y además de estar disponible cumple con los requisitos necesarios.

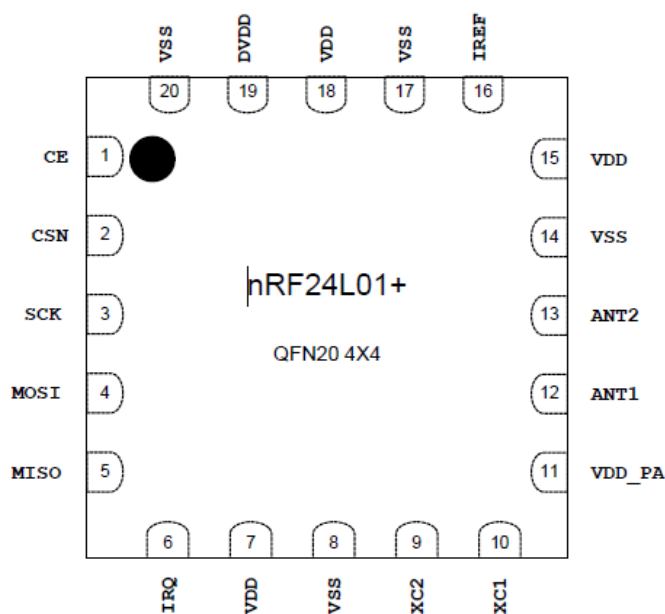


Figura 5.4. nRF24L01 y sus pines

## 5.5. Placa de demostración con PIC18F4520

La placa de demostración que puede albergar tanto el microcontrolador como los diversos periféricos utilizados en la aplicación y que está disponible es la Open18F4520 mostrada en la figura 5.5.

Esta placa dispone de ciertos periféricos de entrada, como pulsadores, elementos de salida como pueden ser la LCD, varios LEDs o el sensor de temperatura y además *jumpers* (puentes que conectan diferentes salidas de la placa). Éstos pueden ser de simple puenteo de los periféricos o de configuración, como el de selección de oscilador de placa o externo o el de selección de tensión a 5V o 3,3V. Éste se ha ajustado en todo momento a 3,3V para no exceder la tensión máxima de alimentación de la pantalla LCD o del módulo de radiofrecuencia.



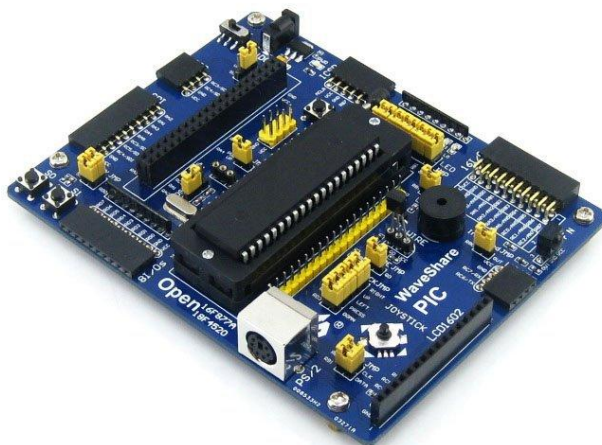


Figura 5.5. Placa de demostración Open18F4520



## 6. PIC18F4520

### 6.1. Características generales

El microcontrolador del proyecto es el PIC18F4520 de la familia Microchip. Éste tiene diversas características de importancia para el desarrollo de la aplicación.

En primer lugar, su alimentación base puede ser de 4'2 a 5'5 V y tiene un rango de temperaturas operativo que va desde -40°C a +85°C.

Para crear el ciclo de funcionamiento del sistema se utilizará uno de los diversos *timers* que posee en microcontrolador. Éste entra en interrupción cíclicamente cada espacio de tiempo. El *timer* en cuestión tiene un ciclo variable que se utilizará tanto para la unidad sensor como para la unidad central en ciclos de 0,1 segundos y que estos utilizarán para controlar el tiempo de sus ciclos de funcionamiento. Cada ciclo el microcontrolador sensor tendrá que comunicar la información sobre la temperatura y el microcontrolador central tendrá que recibir la información de uno de los sensores.

Para controlar las interrupciones utiliza tres bits, uno para permitir su desarrollo, otro para indicar cuando ocurre el evento asociado a la interrupción y un tercero para indicar la prioridad de la interrupción.

A la hora de ejecutar las instrucciones es necesaria una señal de reloj. Para ello, el microcontrolador contiene una señal de entrada de reloj externa. Dispone también de un oscilador interno que genera dos señales de reloj distintas que pueden ser utilizadas como señal de reloj del microcontrolador.

Durante el desarrollo de la aplicación, será necesario deshabilitar las señales de reloj para mantener el microcontrolador dormido (sin ejecutar instrucciones) y bajar su consumo en gran medida. Para ello el microcontrolador dispone de un estado llamado *sleep* al cual se puede acceder mediante una instrucción.

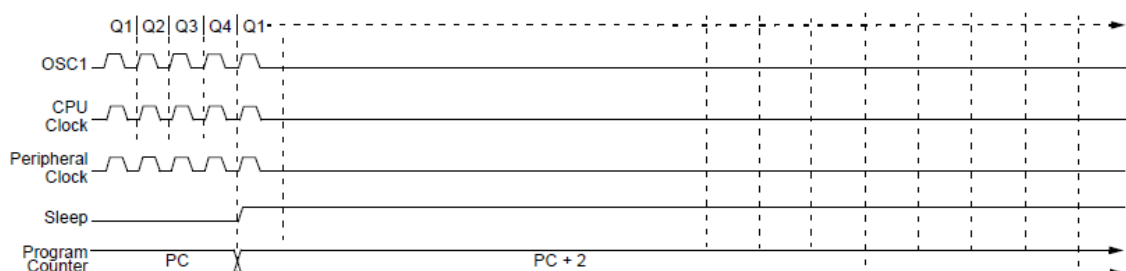


Figura 6.1. Transición de las señales de reloj de modo operativo a modo sleep

Otro aspecto importante es el de los periféricos disponibles. Tiene la posibilidad de conectarle gran variedad de componentes, entre ellos los necesarios para el desarrollo de la aplicación. Para controlar estos periféricos y determinar si son de entrada o de salida, la familia de microcontroladores PIC utiliza para cada periférico tres registros. Del puerto A, por ejemplo, utilizan el TRISA como registro de configuración entrada/salida (0 para salida y 1 para entrada), el LATA como registro para generar salidas y el PORTA como registro de lectura del puerto.

Para evitar el colapso del microcontrolador o que éste entre en bucles infinitos dispone también de diversas señales de reset que lo reinician.

En cuanto a la memoria, en la tabla 6.1 se pueden observar los diferentes tamaños de las diferentes memorias del microcontrolador.

Tabla 6.1 Memoria del PIC18F4520

	Memoria de programa		Data Memory	
	Flash ( <i>bytes</i> )	<i>Single-Word Instructions</i>	SRAM ( <i>bytes</i> )	EEPROM ( <i>bytes</i> )
<b>PIC18F4520</b>	32K	16384	1536	256

## 6.2. Sistema y herramientas de desarrollo

El fabricante del microcontrolador, proporciona también un programa que tiene funciones de editor, compilador, ensamblador, simulador, programador y *debugger* (detección y corrección de errores). Allí se puede escoger el microcontrolador que se está utilizando y facilita la labor de programación de la aplicación. Allí se puede programar en lenguaje C tu aplicación y el programa lo convierte a ensamblador y detecta los errores en el código.

## 6.3. Bibliotecas de subrutinas

A la hora de programar la aplicación, se han utilizado diversos archivos que contienen subrutinas previamente programadas que facilitan al usuario programador el desarrollo de la aplicación. También se han añadido nuevas funciones para el mejor desarrollo de la aplicación.

Estas subrutinas contienen funciones que se refieren tanto a elementos del microcontrolador como a los *timers* o a las interrupciones. También hay subrutinas referidas a cada uno de los periféricos utilizados en el sistema.

Dichas subrutinas realizan funciones como las de inicialización de los periféricos, captación de la temperatura en el caso del sensor, enviar o recibir mensajes por el módulo de radiofrecuencia o mostrar por la pantalla un elemento en el caso de la pantalla LCD.

## 7. El sensor de temperatura DS18B20

### 7.1. Características generales

El sensor DS18B20 proporciona al chip de nueve a doce bits dependiendo del rango de temperatura que estudies. La información la transmite a través de un único bus. Tiene un rango de temperaturas operativo de  $-55^{\circ}\text{C}$  a  $+125^{\circ}\text{C}$  y una precisión de  $\pm 0,5^{\circ}\text{C}$  entre el rango de  $-10^{\circ}\text{C}$  a  $+85^{\circ}\text{C}$ .

Cada sensor DS18B20 tiene un único número de serie de 64 bits, que permite trabajar a varios sensores en el mismo bus de comunicación. Por eso, es fácil controlar desde el mismo microcontrolador una red de sensores a lo largo de un gran área. Aplicaciones como los sistemas de control de temperaturas en casas, maquinaria o equipamiento y sistemas de monitorización y control se aprovechan de este hecho.

Otras características del sensor influyentes en el proyecto son que no necesita componentes externos, cada bus requiere un único pin para comunicarse, la tensión de alimentación va de 3 a 5,5 V, su tiempo de toma de temperatura es de 750 ms máximo y tiene unos sensores de alarma no volátiles.

### 7.2. Medida y conversión de la temperatura

El sensor de temperatura no mide la temperatura constantemente, únicamente lo hace cuando el microcontrolador da la orden. El microcontrolador es de formato digital, por tanto es necesario convertir la medida rápidamente a digital. El tiempo de conversión analógico-digital depende de la resolución que quieras obtener según la tabla siguiente:

Tabla 7.1 Relación entre la resolución del sensor i el tiempo máximo de conversión

Resolución	Máximo tiempo de conversión
9 bits	93,75ms
10 bits	187,5ms
11 bits	375ms
12 bits	750ms

La memoria del sensor se organiza en una memoria volátil de tipo SRAM que se le llama *scratchpad* y contiene 9 *bytes* y una memoria no volátil de tipo EEPROM que contiene 3 *bytes*. Los 3 *bytes* de la EEPROM se copian automáticamente a los *bytes* homónimos de la memoria SRAM. La organización, nombre y función de ambas memorias están expuestos en la figura 7.1.

byte	SRAM		EEPROM
byte 0	LSB de la temperatura		
byte 1	MSB de la temperatura		
byte 2	Alarma de temperatura alta (TH)	↔	Alarma de temperatura alta (TH)
byte 3	Alarma de temperatura baja (TL)	↔	Alarma de temperatura baja (TL)
byte 4	Registro de configuración	↔	Registro de configuración
byte 5	byte reservado 1		
byte 6	byte reservado 2		
byte 7	byte reservado 3		
byte 8	CRC		

Figura 7.1. Estructura de las memorias del sensor DS18B20

Los dos primeros constituyen la información de la variable temperatura expresada en 16 bits. Los dos siguientes guardan el valor de la temperatura máxima y mínima que puede medir el sensor sin que el sensor genere una alarma. El registro de configuración guarda la precisión de la medida, que puede ser configurada de 9 a 12 bits. Como se puede ver también contiene 3 *bytes* reservados para el uso interno del sensor. El último de sus *bytes* es un código que ayuda a la detección de errores del sensor.

La palabra de 12 bits que suministra el sensor codifica en binario complemento a 2 la temperatura expresada en °C. Los 12 bits se reparten en: 1 bit de signo, 7 de parte entera y 4 de parte decimal.

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
LS BYTE	$2^3$	$2^2$	$2^1$	$2^0$	$2^{-1}$	$2^{-2}$	$2^{-3}$	$2^{-4}$
	BIT 15	BIT 14	BIT 13	BIT 12	BIT 11	BIT 10	BIT 9	BIT 8
MS BYTE	S	S	S	S	S	$2^6$	$2^5$	$2^4$

Figura 7.1. Estructura de la variable al medir la temperatura

## 8. El módulo de radiofrecuencia nRF24L01 RF Board

### 8.1. Características generales

El nRF24L01 es un chip radio transceptor que está diseñado para trabajar en la banda ISM (*Industrial, Scientific and Medical*) a 2.4 – 2.5 GHz (banda de frecuencias reservada para finalidades no comerciales). El módulo consiste en un sintetizador de frecuencia, un amplificador de potencia, un oscilador de cristal, un modulador, un demodulador y un sistema de *Enhanced ShockBurst*.

El módulo nRF24L01 RF Board tiene cinco modos de trabajo. Los más importantes son los de envío de datos (estado de transmisión TX) en el cual envía la información hasta que recibe un mensaje de que el envío ha sido recibido o deja de intentarlo o el de recepción de datos (estado de recepción RX) en el cual busca señales entrantes. Dispone también de un modo de bajo consumo (*Power Down*) que se utiliza para configurar los componentes o cargar los bytes de datos. Por último, tiene dos modos de Standby (I y II), uno para el receptor y otro para el transmisor, en los cuales se encuentra cuando éstos no están transmitiendo o recibiendo datos.

El módulo consta de un sistema de *Auto Acknowledgement* que comunica al emisor con el receptor de manera que ambos saben el estado de la transmisión de datos. Es decir, el emisor sabe si el receptor ha recibido bien el mensaje y el receptor sabe si el emisor le ha enviado ése mensaje.

La potencia de radiofrecuencia, los canales de frecuencia y el protocolo de configuración son fácilmente programables mediante la interfaz SPI. Por otro lado, el consumo del módulo es bajo, oscilando entre los 9mA (a una potencia de salida de -6dBm) y los 1,3mA (en el modo de trabajo RX, que en valor medio es el de mayor consumo). Además, un gran ahorro en el consumo es factible gracias al trabajo en los modos de *Power Down*, *Standby-I* y *Standby-II*.

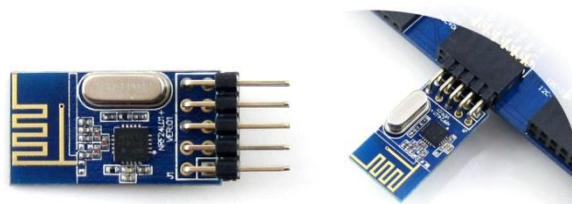


Figura 8.1. El módulo de radiofrecuencia nRF24L01



Otras características básicas del transceptor nRF24L01 son:

- Sistema Automático de Acuse de Recibo o *Auto Acknowledgement* sin intervención del microcontrolador gracias al sistema *Enhanced Shockburst*. Como se verá próximamente, esto permite que el dispositivo TX o transmisor reciba automáticamente un comprobante del dispositivo RX o receptor al recibir un paquete de datos.
- Sistema de identificación y verificación del paquete de datos mediante una dirección (de entre 3 y 5 bytes) y un código CRC (si se activa, de 1 o 2 bytes).
- Sistema múltiple de recepción de datos con 6 *Pipes* de llegada.
- Tasa de transmisión de datos por aire (RF) de 1 o 2 Mbps.
- Tasa de transmisión SPI de entre 0 y 10 Mbps.
- 125 canales de RF.
- Chip de 20 pines con un potencial de alimentación requerido de entre 1.9 y 3.6 V.
- Rango de temperaturas operativo de -40°C a +85°C y de almacenamiento de -40°C a +125°C.

## 8.2. El bus SPI

El módulo de radiofrecuencia utilizado en el proyecto se comunica con el microcontrolador mediante un bus especial, el SPI (*Serial Peripheral Interface*). Esta comunicación permite al microcontrolador cargar una configuración determinada en el módulo de radiofrecuencia, activar y desactivar el envío y recepción de datos, y recoger en cada momento información del proceso para poder controlarlo.

Las características principales del bus SPI son:

- Un canal de comunicación en serie y síncrono, es decir, que paralelamente al canal de datos, existe una señal de reloj que marca los tiempos de funcionamiento.
- Un protocolo de comunicación que trabaja en modo *full dúplex*, es decir, que no tiene un solo canal de datos unidireccional, sino que tiene uno de entrada y uno de salida.
- Está diseñado para comunicar dos o más componentes, pero siempre habrá uno que sea de mayor relevancia que el resto. La característica más importante del maestro es

generar la señal de reloj a la frecuencia adecuada y la de habilitar y deshabilitar los diferentes dispositivos.

- Tiene un mínimo de tres líneas, aunque suelen ser cuatro. Una señal SCLK (Serial Clock) de reloj, una señal MOSI (Master Output, Slave Input) de salida para el componente de mayor relevancia, una señal MISO (Master Input, Slave Output) de entrada para éste y una señal SS (Slave Select) de habilitación para cada uno de los componentes esclavos.
- El proceso de transferencia de datos se inicia cuando el dispositivo maestro activa la señal SS que habilita el dispositivo esclavo e inicia la oscilación del reloj.
- La transferencia de datos se realiza bit a bit. A cada señal del reloj se envía y se recibe un bit.

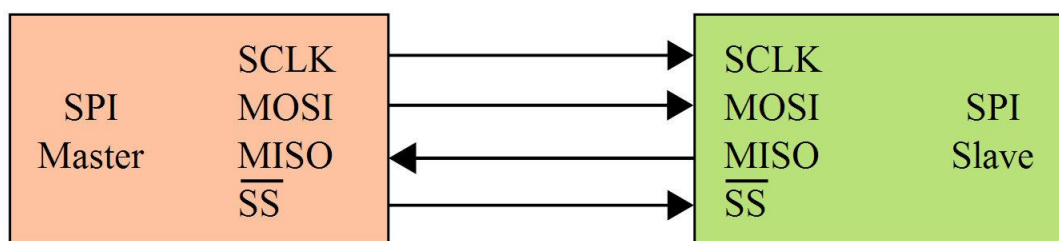


Figura 8.2. Estructura del bus SPI entre maestro y esclavo

### 8.3. Comunicación entre unidades

En el sistema usado en el proyecto, es necesaria una comunicación entre las unidades. Esta comunicación sirve tanto para transmitir la información como para identificar el sensor que está transmitiendo y para marcar los tiempos del ciclo.

Cada unidad debe configurarse para transmitir o recibir datos. El mensaje es enviado por el configurado como emisor y lo recibe el configurado como receptor. El emisor recibe la comprobación de que el mensaje ha sido enviado con éxito y el receptor comprueba que el mensaje recibido ha sido recibido correctamente y tiene diversos mecanismos para saber si el mensaje es el correcto. Entre estos mecanismos se encuentran algunos obligatorios como el de CRC y alguno que se puede deshabilitar como el de checksum.

Los microcontroladores, internamente, se comunican de forma que si el mensaje no le llega al receptor el emisor lo vuelve a enviar hasta un número máximo de retransmisiones variable entre 0 y 15.

Tras la comprobación mencionada, la unidad central guarda la información recibida en la memoria FIFO (First Input, First Output) y posteriormente en la memoria RAM.

## 9. Pantalla LCD1602

### 9.1. Características generales

El modo de visualización de la temperatura escogido es mediante la pantalla LCD1602 de 16 caracteres por 2 líneas. Tiene 16 pines de conexión y es obligatorio utilizar 3'3V de tensión de alimentación. Puede mostrar letras, números y caracteres en blanco con un fondo azul. Puede operar en modo de 8 bits o de 4 para ahorrar pines del microcontrolador. Su rango de temperatura mientras está en funcionamiento es de 0 a +50 °C.

Se puede variar el contraste de la pantalla y así la nitidez de los valores mostrados mediante una tensión suministrada por el tercer pin, referido como V0, que puede variar entre el valor de la tensión de alimentación. Todo esto y otras características de la pantalla LCD se pueden ver en la figura 9.1.

PIN NO.	SYMBOL	DESCRIPTION	FUNCTION
1	VSS	Ground	0V (GND)
2	VCC	Power Supply For Logic Circuit	+3.3V
3	V0	LCD Contrast Adjustment	Recommends 10K Potentiometer between V0 and GND
4	RS	Instruction/Data Register Selection	RS = 0 : Instruction Register RS = 1 : Data Register
5	R/W	Read/Write Selection	R/W = 0 : Register Write R/W = 1 : Register Read
6	E	Enable Signal	Active when High-level turns to Low-level
7	D0	Data Input/Output Lines	8 BIT: D0-D7
8	D1		
9	D2		
10	D3		
11	D4		
12	D5		
13	D6		
14	D7		
15	A	Supply Voltage For LED+	+3.3V
16	K	Supply Voltage For LED-	0V (GND)

Figura 9.1. Descripción y funciones de los pines de la pantalla LCD

Como podemos ver, existe un pin que configura si la información que se está utilizando son datos o instrucciones, otro que configura si se está leyendo o escribiendo y una señal de habilitación que será útil y de la cual se hablará más adelante. También se puede

comprobar que tiene ocho pines de datos bidireccionales y corresponden a los pines desde el D0 al D7 del microcontrolador.

Para su configuración se han utilizado diversas subrutinas que configuran estos pines de la manera que requiere el sistema. Estas subrutinas han sido definidas utilizando la señal de reloj de 4 MHz que contiene el microcontrolador. En caso de verse alterada la señal de reloj, es necesario reconfigurar dichas subrutinas.

## 10. Estructura del software de la aplicación

Para el desarrollo de la aplicación, es necesaria la programación de los dos microcontroladores necesarios, el sensor y el central. Para ello se dispone de la herramienta de desarrollo mencionado anteriormente que convierte el lenguaje C en lenguaje para el microcontrolador (lenguaje máquina).

El código de programación tanto del nodo sensor como del central se divide en las siguientes partes:

- Incluir subrutinas necesarias para el código
- Iniciación de variables
- Código principal
- Interrupciones

A continuación, se exponen los diversos programas, el desarrollo de éstos están expuestos junto con las subrutinas utilizadas en los anexos. También se incluyen las subrutinas utilizadas, muchas de ellas obtenidas de otros proyectos y alguna desarrollada para esta aplicación.

### 10.1. Software del nodo sensor

En el caso del nodo sensor, las subrutinas necesarias son las que tienen que ver con funciones del sensor DS18B20 y del módulo de radiofrecuencia basado en el chip nRF24L01. Además incluye un archivo con subrutinas que tienen que ver con el bus 1wire que utiliza el sensor de temperatura y es un bus especial.

Las variables que utiliza la unidad sensor son:

- direccionX: variable compuesta por cinco bytes que constituyen la dirección de cada sensor donde X es el número del sensor.
- direccion transmision: variable también de cinco bytes que corresponde a la dirección del nodo central.
- T: variable de nueve bytes donde se guarda toda la información obtenida del sensor DS18B20.

- temperatura: variable de dos bytes donde se guardan los dos primeros bytes de la variable T que corresponden al valor de la temperatura obtenida por el sensor.
- informe: variable que corresponde al informe que devuelve la unidad de RF al enviar o recibir la información. En ella se puede ver si el envío ha tenido éxito.
- activación: variable diferente en cada sensor que se usa para comprobar si el sensor en cuestión es el que debe enviar la temperatura.
- i: contador que se utiliza para contabilizar el tiempo el valor del cual aumenta cada vez que el programa entra en la interrupción.

La secuencia de funciones y la función que realiza cada una en el nodo sensor se divide en las que forman parte del programa principal y las que se encuentran en la secuencia de interrupción.

Dentro del programa principal se configuran los registros TRISB a 0x00 y TRISD a 0xFF de manera que requiere la unidad y se inicializan las interrupciones junto con el timer0 de manera que éste se ejecute cada décima de segundo.

Las funciones referentes al sensor DS18B20 que se ejecutan a continuación en un bucle infinito durante todo el sistema son las siguientes:

- Detect Slave Device(): comprueba la existencia del sensor en el microcontrolador en cuestión y envía un 1 en caso afirmativo y un 0 en caso contrario.
- DS18B20 convert temperature(): realiza la conversión de temperatura interiormente sin enviarla ni almacenarla en ninguna variable.
- DS18B20 read scratchpad(T): lee la conversión de temperatura realizada en la función anterior y la guarda en la variable de entrada, en este caso la variable utilizada es T.

Una vez guardada la temperatura se cogen los dos primeros bytes de la variable T que corresponden al valor de la temperatura.

Por último se iguala el registro OSCCONbits.IDLEN a 0 que deja al microcontrolador en modo dormido a la espera que una interrupción lo despierte.

En el caso de la interrupción, la secuencia de instrucciones que ejecuta el programa se divide en la recepción del mensaje de activación por parte de la unidad central y el envío de la temperatura a modo de respuesta. Esta interrupción se ejecuta cada décima de segundo

pero el programa asociado únicamente se desarrolla una vez cada segundo. Esto es posible gracias a la variable *i* que hace de contador de tiempo.

Las funciones que componen la recepción del mensaje de activación son las siguientes:

- SPI\_Start(0b10): inicializa el canal SPI por el cual se transmite la información. La variable de entrada corresponde a la frecuencia que utilizará el canal.
- nRF24L01\_Ports\_Start(): inicializa el módulo de RF dentro del microcontrolador.
- Start\_RX\_Mode\_nRF24L01(0b11, 0b1000000, 0, 0b11, 1, 1, direccion1, direccion2, 0x00, 0x00, 0x00, 0x00, 1, 1, 0, 0, 0, 0, 0): configura el módulo de RF como receptor. Las variables de entrada de principio a fin corresponden al tamaño de las direcciones que puede ser de 3 a 5 bytes, frecuencia utilizada que va de 0x00 a 0xFF, velocidad de envío que puede ser de 1 Mbps o de 2 Mbps, potencia emisora que va de -18 dBm a 0dBm, habilitación del amplificador que disminuye la influencia del ruido en la transmisión, existencia de 1 o 2 bytes de CRC que comprueba la veracidad de los datos enviados, direcciones de los seis pipes de entrada de datos y el tamaño de los datos a recibir en cada uno de los pipes.
- Receive\_Data\_RX\_Mode\_nRF24L01(1, 100, 1, activacion): recepción del mensaje que devuelve un byte que indica si el envío ha sido correcto. Las variables de entrada corresponden a la habilitación del byte de *checksum*, el tiempo de espera máximo, el tamaño del mensaje a recibir y la variable donde se guardará el mensaje recibido.
- Finish\_nRF24L01\_Operation(): desactiva el módulo de RF al acabar la transmisión.
- Finish\_SPI\_Operation(): concluye la secuencia de instrucciones desactivando el canal SPI por el cual se ha transmitido la información.

Una vez recibido el mensaje de activación se comprueba si el sensor en cuestión es el indicado y en caso afirmativo se procede al envío de la temperatura obtenida. Ésta se ejecuta mediante las instrucciones siguientes:

- SPI\_Start(0b10) y nRF24L01\_Ports\_Start(): se inicializa tanto el canal SPI como el módulo de RF al igual que en la secuencia de recepción.
- Start\_TX\_Mode\_nRF24L01(0b11, 0b1000000, 0, 0b11, 1, 1, 0b0000, 10, 1, 2): configura el módulo de RF como receptor. Las variables de entrada de principio a fin corresponden al tamaño de las direcciones que puede ser de 3 a 5 bytes,



frecuencia utilizada que va de 0x00 a 0xFF, velocidad de envío que puede ser de 1 Mbps o de 2 Mbps, potencia emisora que va de -18 dBm a 0dBm, habilitación del amplificador que disminuye la influencia del ruido en la transmisión, existencia de 1 2 bytes de CRC que comprueba la veracidad de los datos enviados, tiempo entre retransmisiones que va de 0b0000 a 0b1111, cantidad de retransmisiones a enviar en caso de fallo que va de 0 a 15, habilitación del byte de *checksum* y tamaño del mensaje a enviar.

- Send Data TX Mode nRF24L01(1, 0b11, direccion transmision, 2, temperatura): envío del mensaje. Las variables de entrada corresponden a la habilitación del byte de *checksum*, tamaño de la dirección a la que enviar el mensaje que puede ser de 3 a 5 bytes, dirección a la que enviar el mensaje, tamaño del mensaje a enviar y mensaje a enviar.
- Check Data Sent TX Mode nRF24L01(): función que comprueba el envío del mensaje y devuelve un byte con el éxito o no del mensaje y el número de retransmisiones utilizadas para el envío.
- Finish nRF24L01 Operation() y Finish SPI Operation(): se concluye el envío igual que en el caso de la recepción, cerrando el módulo de RF y el canal SPI.

Por último se desactiva el bit de la interrupción para evitar que ésta entre en un bucle.

## 10.2. Software del nodo central

En el caso del nodo central, las subrutinas necesarias son las que tienen que ver con funciones de la pantalla LCD1602 y también del módulo de radiofrecuencia anterior.

Las variables que utiliza la unidad central son:

- direccionenvio: variable compuesta por cinco bytes que constituyen la dirección del sensor al cual va a realizar el envío de la activación.
- direccionX: variable también de cinco bytes que constituyen la dirección de cada sensor donde X es el numero del sensor.
- direccion transmision: variable también de cinco bytes que corresponde a la dirección del nodo central.
- T: variable de dos bytes donde se guarda la temperatura recibida de la unidad sensor.

- buffer: variable de doce bytes donde se descodifica la información referente a la temperatura para poder ser mostrada por la pantalla.
- informe e informe2: variables que corresponden a los informes que devuelven las unidades de RF al enviar o recibir la información. En ella se puede ver si el envío ha tenido éxito y se utilizará para conocer si el sensor en cuestión está activo o no.
- activación: variable que indica el sensor al que se dirige la unidad central.
- encendido: variable de diez bytes donde se guarda el estado de cada uno de los sensores.
- act: variable compuesta por diez bytes donde contiene todas las variables de activación de todos los sensores.
- s, c, j: contadores que se utilizan para contabilizar el tiempo, la temperatura que ha de mostrar la pantalla o el sensor de la temperatura mostrada.
- s00 y s11: variables que guardan el valor anterior de los botones utilizados en el sistema.
- cambio: variable que contempla un cambio en la información y que ejecuta el programa asociado a la pantalla.
- sensor: diez estructuras constituidas por tres variables de doce bytes donde se guardan las variables temperatura, temperatura máxima y temperatura mínima de cada sensor.

La secuencia de instrucciones y la función que realiza cada una en el nodo sensor se divide en las que forman parte del programa principal y las que se encuentran en la secuencia de interrupción.

Dentro del programa principal se inicializan las variables temperatura mínima y máxima a los valores extremos para su correcto funcionamiento; se configuran los registros ADCON0bits.ADON, TRISCbits.TRISC4, TRISCbits.TRISC5 y TRISCbits.TRISC6 a 0 para el correcto funcionamiento de la pantalla; el registro TRISB y TRISD a 0xFF para los botones y la pantalla y se inicializan las interrupciones junto con el timer0 de manera que éste se ejecute cada décima de segundo. Se ejecuta también la función InitLCD() que inicializa la pantalla y la activa.

A continuación se inicia un bucle infinito que ocurre únicamente cada vez que la unidad recibe un mensaje por RF. En este bucle se realizan diversas instrucciones que se pueden agrupar de la siguiente manera:

- strcpy(sensor[c].T,T): copia la variable temperatura recibida en la variable que le corresponde de la estructura sensor.
- Se compara la variable de entrada con las variables mínima y máxima del sensor en cuestión y se actualiza el valor si es necesario. Para esta función ha sido necesario estudiar el signo de las variables para estudiarlas por separado debido a la complejidad de la comparación.
- Ayudado por las variables direccionX se actualiza la variable direccionenvio para la próxima comunicación por RF.
- Se ejecuta el código asociado a la pantalla, primero de todo se limpia la pantalla con la función ClearLCD(), se escriben las palabras sensor y el número del sensor con la ayuda del contador j, para escribir cualquier cosa en la pantalla es necesario utilizar la función BusyXLCD(). Por último se escribe la temperatura que se requiera con la ayuda del contador s. Para poder escribirla por la pantalla es necesario transformarla de la variable T a la variable buffer mediante la función DS18B20\_decode\_temp(sensor[j].T,buffer). Destacar que la temperatura viene acompañada por los signos T en caso de que sea una temperatura actual, por los signos Tm en caso de una temperatura mínima y de TM en caso de temperaturas máximas. Por último, en caso que el sensor este apagado en lugar de las temperaturas se muestra la palabra "Apagado". Para escribir en la pantalla se usan las funciones putsXLCD, putsXLCD y putcXLCD dependiendo de la naturaleza de la variable a mostrar.
- Se iguala la variable cambio a 0 a la espera de una nueva comunicación de la unidad.

En el caso de la interrupción la secuencia de instrucciones que ejecuta el programa de esta unidad es más compleja debido al mayor número de necesidades de esta unidad.

Primero de todo se configura el LATAbits.LATA5 a 0 para evitar interferencias entre la pantalla y el módulo de RF. A continuación se actualiza la variable activación para cada comunicación y se pasa al envío y recepción del módulo de RF.

En esta unidad la primera parte de la comunicación es la de transmisión del mensaje de activación al sensor que se requiera, para ello se utilizan las funciones especificadas en la

unidad sensor con algún parámetro de entrada modificado. A continuación se vuelven a exponer dichas funciones:

- SPI\_Start(0b10) y nRF24L01\_Ports\_Start(): se inicializa tanto el canal SPI como el módulo de RF al igual que en la secuencia de la unidad sensor.
- Start\_TX\_Mode\_nRF24L01(0b11, 0b1000000, 0, 0b11, 1, 1, 0b0000, 10, 1, 1): configura el módulo de RF como receptor. Las variables de entrada de principio a fin corresponden al tamaño de las direcciones que puede ser de 3 a 5 bytes, frecuencia utilizada que va de 0x00 a 0xFF, velocidad de envío que puede ser de 1 Mbps o de 2 Mbps, potencia emisora que va de -18 dBm a 0dBm, habilitación del amplificador que disminuye la influencia del ruido en la transmisión, existencia de 1 o 2 bytes de CRC que comprueba la veracidad de los datos enviados, tiempo entre retransmisiones que va de 0b0000 a 0b1111, cantidad de retransmisiones a enviar en caso de fallo que va de 0 a 15, habilitación del byte de *checksum* y tamaño del mensaje a enviar.
- Send\_Data\_TX\_Mode\_nRF24L01(1, 0b11, direccionenvio, 1, activacion): envío del mensaje. Las variables de entrada corresponden a la habilitación del byte de *checksum*, tamaño de la dirección a la que enviar el mensaje que puede ser de 3 a 5 bytes, dirección a la que enviar el mensaje, tamaño del mensaje a enviar y mensaje a enviar.
- Check\_Data\_Sent\_TX\_Mode\_nRF24L01(): función que comprueba el envío del mensaje y devuelve un byte con el éxito o no del mensaje y el número de retransmisiones utilizadas para el envío, esto se guarda en la variable informe2.
- Finish\_nRF24L01\_Operation() y Finish\_SPI\_Operation(): se concluye el envío al igual que en el caso de la unidad sensor, cerrando el módulo de RF y el canal SPI.

Una vez enviado el mensaje de activación se pone a esperar la respuesta de dicho sensor y se dispone a recibir dicha temperatura. Ésta recepción se ejecuta similar a la de la unidad sensor mediante las instrucciones siguientes:

- SPI\_Start(0b10) y nRF24L01\_Ports\_Start(): se inicializa tanto el canal SPI como el módulo de RF al igual que en la secuencia de envío.
- Start\_RX\_Mode\_nRF24L01(0b11, 0b1000000, 0, 0b11, 1, 1, direccion1, direccion2, 0x00, 0x00, 0x00, 0x00, 1, 2, 2, 0, 0, 0, 0): configura el módulo de RF como receptor. Las variables de entrada de principio a fin corresponden al tamaño de las direcciones que puede ser de 3 a 5 bytes, frecuencia utilizada que va de 0x00 a 0xFF, velocidad de envío que puede ser de 1 Mbps o de 2 Mbps, potencia emisora

que va de -18 dBm a 0dBm, habilitación del amplificador que disminuye la influencia del ruido en la transmisión, existencia de 1 o 2 bytes de CRC que comprueba la veracidad de los datos enviados, direcciones de los seis pipes de entrada de datos y el tamaño de los datos a recibir en cada uno de los pipes.

- Receive Data RX Mode nRF24L01(1, 1, 2, T): recepción del mensaje que devuelve un byte que indica si el envío ha sido correcto. Las variables de entrada corresponden a la habilitación del byte de *checksum*, el tiempo de espera máximo, el tamaño del mensaje a recibir y la variable donde se guardará el mensaje recibido.

- Finish nRF24L01 Operation() y Finish SPI Operation(): se concluye la recepción al igual que en el caso del envío, cerrando el módulo de RF y el canal SPI.

Una vez recibida la temperatura se leen los registros asociados a los botones y con la ayuda de las variables s00 y s11 se estudia si se han pulsado dichos botones. En caso afirmativo se modifican los contadores j y s.

El siguiente paso del código es estudiar si el sensor correspondiente a la comunicación estaba encendido o apagado. Para ello se escoge la variable informe2 correspondiente al envío del mensaje de activación. En esta variable está incluido el éxito del envío y se puede observar si el sensor está o no activado, por tanto si el envío ha sido correcto se considera que el sensor está encendido y si no ha tendido éxito se considera que el sensor se encuentra apagado. Entonces se actualiza el byte del sensor correspondiente de la variable encendido como convenga.

A cada ciclo de sensores, es decir, cada diez interrupciones, y una vez realizado la comunicación con todos los sensores se activa la variable de cambio para leer el código del programa principal y se vuelve a configurar los registros de la pantalla y el registro LATAbits.LATA5 a 1.

Por último se desactiva el bit de la interrupción para evitar que ésta entre en un bucle.

### 10.2.1. Multiplexado de señales

La configuración del nodo central se ha encontrado un problema que no ocurre en el nodo sensor. Éste es que existen diversos pines que son utilizados tanto por la pantalla LCD como por el módulo de radiofrecuencia. Estos pines son los 4 y 5 del puerto C. Para evitar que al comunicarse con uno de ellos el otro se vea afectado, es necesario definir bien cuando estás comunicándote con uno y cuando con el otro.

En el caso de la interferencia del módulo de radiofrecuencia con la LCD, se ha visto que la LCD solo funciona cuando en el pin 6 del puerto C efectúa un flanco de bajada. Por tanto, como el módulo de radiofrecuencia solo altera el pin 4 y 5 éste no interferirá con la LCD.

En el caso de la interferencia de la LCD con el modulo de radiofrecuencia, se ha visto que el módulo de radiofrecuencia se habilita con el pin 5 del puerto A, por lo tanto es necesario habilitarlo cuando se quiera utilizar el módulo de radiofrecuencia y deshabilitarlo al acabar.

Por último, destacar que de los dos, el pin 4 es de entrada para la LCD y de salida para el módulo de radiofrecuencia. Para ello, se ha comprobado que no existan conflictos eléctricos a la hora de utilizarlos en ambos sentidos. Esto no pasa debido a que el bit de habilitación del módulo de radiofrecuencia lo deja en alta impedancia.

### **10.3.Mensajes de radiofrecuencia utilizados**

Dentro de la aplicación, los microcontroladores se comunican mediante radiofrecuencia como se ha explicado anteriormente. Los dos tipos de mensajes que se intercambian son el mensaje de activación de la unidad central a cada uno de los sensores y el mensaje con la variable que contiene la temperatura que va de la unidad sensor a la unidad central.

La variable activación que envía la unidad central varía cada décima de segundo para que el sensor que la identifique sea cada vez diferente. Esta variable será de un byte.

En cuanto a la variable temperatura, ésta consta de dos bytes que se actualizarán cada segundo que contiene la información referente a la temperatura del sensor en cuestión. La figura 7.1 muestra el formato y la estructura de esta variable.

## 11. Prestaciones del sistema final

En este apartado se exponen las prestaciones del sistema que se han conseguido una vez finalizada la aplicación.

### 11.1. Alimentación

La tensión de alimentación que requieren los diferentes componentes del sistema son los especificados en la tabla 11.1.

Tabla 11.1 Alimentaciones de los diferentes componentes del sistema

Componente	Alimentación (V)
PIC18F4520	4,2 a 5,5
DS18B20	3 a 5,5
LCD1602	3,3
nRF24L01	1,9 a 3,6

Como se puede ver en la tabla 11.1 el rango de temperaturas es diverso, por lo tanto será necesario utilizar las diferentes temperaturas que ofrece la Open18F4520. Estas son de 5 V y de 3,3 V.

### 11.2. Alcance

El alcance del sistema final puede variar en función de gran un gran número de factores, muchos de ellos difíciles de calcular como por ejemplo la existencia de obstáculos. Es por eso que los fabricantes no proporcionan el valor de este alcance. Ellos proporcionan el valor de la potencia emisora y el valor de la sensibilidad de el receptor, ambas en dBm.

Ésta potencia no es fija y se puede variar dependiendo del sistema en cuestión. Ésta puede variar desde 0 dBm (la más grande) hasta -18dBm (la más pequeña). Este valor se fija a la hora de configurar vía software tanto las unidades transmisora como receptora.

Cuanto mayor alcance mayor potencia emisiva, por lo tanto, mayor consumo. Por lo tanto dependiendo de la aplicación específica que cada usuario le dé al sistema tendrá que configurar una potencia u otra.

Un aspecto importante a la hora de configurar el alcance y la potencia emisora es que existe una legislación vigente respecto al tiempo máximo al que puedes estar emitiendo en una frecuencia libre en función del alcance de tu transmisión.

A la hora de aproximar el valor del alcance en función de la potencia emisora se han realizado unas pruebas sin obstáculos intermedios utilizando las potencias emisoras máxima y mínima configurables. Los resultados obtenidos son los expresados en la tabla

Tabla 11.2 Alcance del sistema en función de la potencia emisora

11.2.

Potencia emisora (dBm)	Alcance (m)
0	3,5
-18	54

Por tanto y para cumplir con las especificaciones iniciales, se ha escogido la potencia emisora de -18 dBm.

### 11.3. Consumo

Como ya se ha expuesto anteriormente el consumo del sistema variará en función de la aplicación del sistema y de la potencia emisora escogida. Como ya se ha dicho antes, se ha diseñado el sistema minimizando el consumo de tal forma que las unidades no estén en continuo funcionamiento.

A pesar de todo, dependiendo de la distancia de las unidades en la aplicación real a utilizar el usuario elegirá una potencia que satisfaga sus necesidades, por tanto el consumo del sistema siempre será variable.



## 11.4. Rango de temperatura

A continuación se exponen los rangos de temperatura operativos de los diferentes componentes que conforman el sistema:

Tabla 11.3 Rango de temperaturas operativas del sistema

Componente	T. mínima (°C)	T. máxima (°C)
PIC18F4520	-40	+85
DS18B20	-55	+125
LCD1602	0	+50
nRF24L01	-40	+85

Como se puede ver en la tabla 11.3, el rango de temperaturas de la unidad sensor es de -40 a 85 °C. Esto implica que no se podrá aprovechar al máximo el rango del sensor DS18B20. No obstante, se cumple con creces el intervalo impuesto al comienzo del proyecto.

En el caso de la unidad master o central, el rango de temperatura operativo lo impone la pantalla LCD y es de 0 a 50°C. Como esta unidad no está diseñada para medir la temperatura, no supone ningún inconveniente a la hora que el sistema necesite medir temperaturas fuera de su rango.

## 11.5. Costes

MATERIAL Y SOFTWARE			
ITEM	Cantidad	Precio Unidad (€/u)	Coste Total Item (€)
<b>Pack Open 18F4520 Package B</b> (Placa, PIC18F4520, módulo nRF24L01, LCD, sensor DS18B20, cables de conexión, periféricos)	4	73,75	295
<b>Programador MPLAB ICD 2</b> (Programador, cable USB)	1	147,13	147,13
<b>Software MPLAB y compilador C18</b>	1	0	0
<b>TOTAL</b>			442,13

TIEMPO DE TRABAJO			
	Horas	Coste (€/h)	Coste Total (€)
<b>Trabajo dedicado al estudio previo</b>	75	10	750
<b>Trabajo dedicado al desarrollo del proyecto</b>	225	10	2250
<b>TOTAL</b>			3000

<b>CONSUMO DE RECURSOS</b>			
	Horas	Coste (/Wh)	Coste Total (€)
<b>Consumo electricidad del PC y placas (100 W)</b>	225	0,14	31,5
<b>Consumo electricidad del espacio de trabajo (100 W)</b>	225	0,14	31,5
		<b>TOTAL</b>	63

<b>COSTE TOTAL DEL PROYECTO</b>	
	Coste (€)
<b>Material y software</b>	442,13
<b>Tiempo de trabajo</b>	3000
<b>Consumo de recursos</b>	63
<b>TOTAL</b>	3505,13

## 11.6. Impacto medioambiental

Respecto al impacto medioambiental causado por el hardware como futuro residuo, una vez termina la vida útil del producto que debe ser lo más larga posible, las opciones que pueden ser consideradas son el reciclaje o su desecho en gestores de residuos especializados en residuos electrónicos.

En cuanto al reciclaje de cada una de las partes de una unidad, su reciclaje es altamente inviable, ya que la reutilización correspondería un coste mayor que adquirir nuevos componentes. Por lo tanto, la mejor opción sería la reutilización de cada unidad en su conjunto para otras funciones.

Por último, aunque se realicen diferentes fases de reciclado, será necesario acabar desechando los componentes de forma correcta. Esto se realiza en centros de gestión de residuos de aparatos eléctricos y electrónicos (RAEE).

Uno de los aspectos más importantes del impacto medioambiental del proyecto es también la influencia de las transmisiones por radiofrecuencia en las personas y en los otros elementos electrónicos cercanos a los transmisores. Para ello, es necesario conocer la frecuencia de trabajo, la distancia a la que se encontrarán los elementos de las antenas y la potencia de las mismas.

La frecuencia de trabajo del módulo de radiofrecuencia nRF24L01 es de entre 2.4 y 2.5 GHz (enmarcada dentro de la Alta Frecuencia de entre 3 MHz y 3 GHz). A potencias muy altas, estos espectros pueden causar en un individuo dolores, quemaduras, pérdidas de visión, o muchos otros daños. Para conocer el impacto en los individuos es necesario controlar dos factores adicionales, la potencia del campo electromagnético aplicado y el tiempo de exposición.

La potencia de salida del nRF24L01 corresponde como máximo a 0dBm, lo que corresponde a 1mW según la fórmula de la figura 11.1.

$$P_{dBm} = 10 \cdot \log_{10} P_{mW}$$

---

Figura 11.1. Fórmula para calcular la potencia  
emisor

Para determinar por tanto la incidencia del campo generado por el nRF24L01 en el organismo, se tiene en consideración el parámetro SAR (*specific absorption rate*). Según el Organismo Mundial de la Salud, dentro del rango de frecuencias de entre 1 MHz y 10 GHz, es necesario un SAR de por lo menos 4W/Kg para producir efectos adversos en la salud de gravedad, y según el *Safety Code 6* del departamento de sanidad de Canadá, para un largo periodo de exposición (por ejemplo de un operario que trabaja durante horas), no es admisible una densidad de potencia mayor de 0.4W/Kg para evitar cualquier tipo de lesión. Suponiendo un peso alto para una persona, como 100 Kg, y el último valor expuesto, la potencia máxima que puede soportar es de 24 W. Por lo tanto y sabiendo que toda la potencia generada por la antena es de 1 mW, se puede llegar a la conclusión de que el uso del sistema no supone ningún daño a las personas.

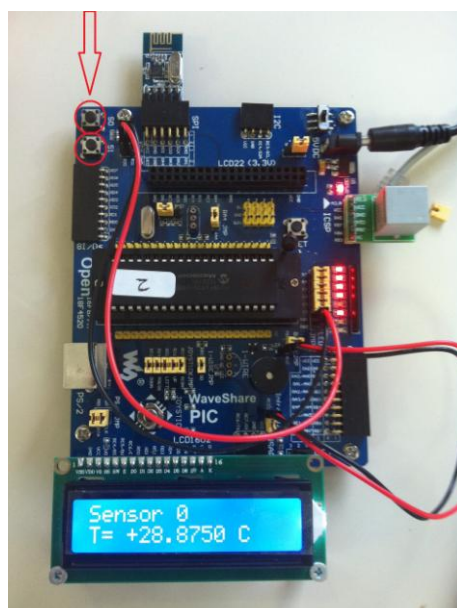
## 12. Guía de usuario del sistema

En la pantalla, el usuario siempre podrá ver la temperatura actual, máxima o mínima de uno de los sensores y el número de sensor del que proviene.

Para la correcta utilización del sistema, el usuario dispone de 2 botones con los que manejar la información. Los botones varían el sensor del cual se están mostrando las temperaturas y las temperaturas que se están mostrando. Los sensores están clasificados del 1 al 10 y el botón superior (S1) variará el sensor del cual se está mostrando la temperatura y el inferior (S0) variará la temperatura que se está mostrando, ya sea la actual, la máxima o la mínima. En la figura 12.1 se pueden observar los botones a utilizar.

Destacar que debido a que los registros que utilizan los botones por defecto estaban previamente utilizados por la pantalla, se han cambiado por los bits 0 y 1 del puerto B con ayuda de *jumpers*.

Figura 12.1 Unidad central con señalización en los botones



Cabe destacar que los puertos que utilizan los botones por defecto ya son utilizados por la pantalla. Por lo tanto se han tenido que modificar para no causar interferencias, los utilizados han sido los puertos RB0 y RB1.

A la hora de mostrar la temperatura actual, ésta quedara mostrada como  $T$ , en el caso de las temperaturas máximas y mínimas del sensor en cuestión se mostrarán como  $T_M$  y  $T_m$  respectivamente como puede verse en la figura 12.2.

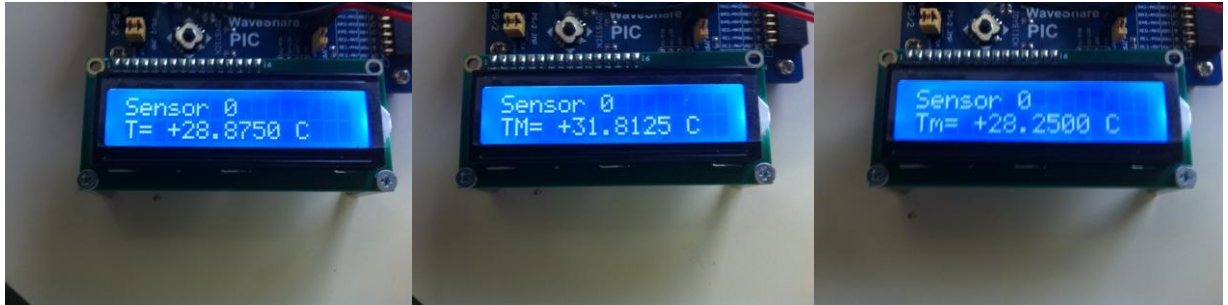


Figura 12.2 Pantalla mostrando la temperatura actual, máxima y mínima de un sensor

Para conocer cuántos sensores están encendidos y enviando las temperaturas, se ha programado que los sensores de los cuales no se reciba ninguna temperatura se muestre en ellos la palabra "Apagado" como se muestra en la figura 12.3.

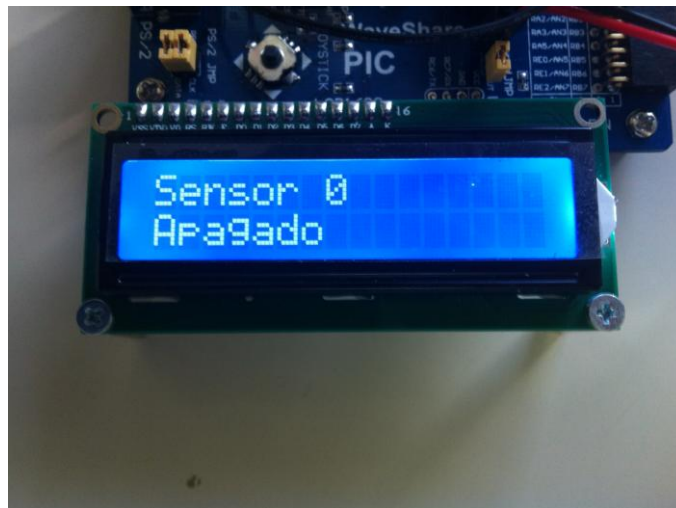


Figura 12.3 Pantalla mostrando un sensor apagado

## Conclusiones

La conclusión principal del proyecto es que ha sido posible realizar el sistema y que cumpliese con las especificaciones previamente definidas.

Cabe destacar también la gran utilidad de las subrutinas que han sido proporcionadas para el desarrollo de la aplicación. Éstas contenían funciones de todos los módulos y del bus que utiliza el sensor de temperatura y sin ellas el proyecto no se podría haber conseguido.

El punto más dramático del proyecto ha sido la interferencia de señales entre el módulo de radiofrecuencia y la pantalla LCD. Para ello, ha sido necesario estudiar a fondo ambos módulos para encontrar la señal de inhabilitación de cada componente cuando el otro está siendo usado. También ha sido importante estudiar las implicaciones eléctricas de esta inhabilitación debido a que cada uno de los módulos utilizaba la señal en sentido contrario.

Otra conclusión importante que se ha observado es que el rango de temperaturas a estudiar es inferior al que puede medir el sensor debido a que alguno de los componentes no puede soportar intervalos de temperatura tan grandes. También destacar que la unidad visualizador tiene un rango todavía más pequeño pero que esto no interfiere con el rango a estudiar porque la unidad visualizador no mide ninguna temperatura.

Como observación final destacar que las pruebas han sido realizadas con tres sensores pero con los espacios definidos para el número de sensores definidos en las especificaciones.



## Agradecimientos

Por último me gustaría agradecer principalmente a mi tutor Emili Lupón por su aguante ante mi falta de concentración en muchos momentos y su ayuda en los momentos importantes. También a Manuel Moreno por su gran ayuda ante los muchos problemas que se me han planteado a lo largo del proyecto.

No me quiero olvidar de Martín Rolando por sus consejos y ayudas en los momentos finales y de Mar Font por su gran ayuda y apoyo durante estos meses.





## Bibliografía

- MICROCHIP TECHNOLOGY INC. *PIC18F2420 / 2520 / 4420 / 4520 Data Sheet*, 2008.
- UPC DEPARTAMENTO DE ING. ELECTRÓNICA. *DS18B20 Digital Thermometer*, 2014.
- MAXIM INTEGRATED. *DS18B20 Programmable Resolution 1-Wire Digital Thermometer*.
- SASHAVALLI MANIYAR Microchip Technology Inc. *1-Wire® Communication with PIC® Microcontroller*.
- UPC DEPARTAMENTO DE ING. ELECTRÓNICA. *LCD1602 Board*.
- UPC BARCELONATECH. *LCD1602 Lib. User Manual*
- NORTON. *nRF24L01 Product Specification*. 2008.
- PABLO SANZ ETSEIB UPC. *UPC\_nRF24L01 User Manual*. 2014.

## Bibliografía complementaria

- AITOR ZULOAGA IZAGUIRRE, ARMANDO ASTARLOA CUÉLLAR. *Sistemas de procesamiento digital*, 2008.
- DOGAN IBAHIM. *Advanced PIC Microcontroller Projects in C*, 2011.
- JAVIER MARTÍNEZ PÉREZ, MARIANO BARRÓN RUIZ. *Prácticas con microcontroladores [de 8 bits]*, 1993.