

# **Sistemas Basados en Microprocesador**

## **Bloque 3**

---

**Integración y desarrollo de una aplicación:  
Controlador de una válvula termostática**

---

## **Realización del diseño (IMPORTANTE)**

Para la realización de este bloque de la asignatura (Bloque 3) se dispone de 7 sesiones presenciales más una sesión de evaluación. La evaluación se realizará en la **última sesión presencial (miércoles, 20 de diciembre)**, si bien puede realizarse con anterioridad, previa solicitud al profesor de su grupo de laboratorio y siempre que se haya entregado tanto los proyectos como la documentación requerida con anterioridad.

Para afrontar con garantías el diseño, éste debe dividirse en una serie de hitos que permitan ir comprobando el correcto funcionamiento de cada uno de los módulos que se vayan desarrollando.

Se deberán realizar entregas en Moodle con el progreso del diseño. Deben entregarse proyectos completos que demuestren el correcto funcionamiento de cada uno de los módulos que se van desarrollando.

El calendario de entrega de módulos verificados es el siguiente:

	Fecha	Módulos
<b>E1</b>	Martes, 28/11/2023, 21:00	JOYSTICK, HORA
<b>E2</b>	Martes, 5/12/2023, 21:00	TEMPERATURA, LCD
<b>E3</b>	Martes, 12/12/2023, 21:00	COM-PC, PWM OUTPUT, POT_1/POT_2
<b>E4</b>	Martes, 19/12/2023, 21:00	Proyecto FINAL y memoria técnica

	10	13/11/2023	B3 (T)		B2			10	17/11/2023
	11	20/11/2023	B3		B3			11	24/11/2023
dic	12	27/11/2023	Examen B2	E1	B3			12	01/12/2023
	13	04/12/2023	B3	E2	06-dic	07-dic	08-dic	13	08/12/2023
	14	11/12/2023	B3	E3	B3			14	15/12/2023
	15	18/12/2023	B3	E4	Examen B3			15	22/12/2023

Adicionalmente a la implementación del diseño es necesario tener en cuenta que se debe entregar una memoria técnica al finalizar el diseño, siguiendo el modelo que se encuentra en Moodle ("*B3\_Plantilla\_Memoria.docx*").

La **memoria**, junto al proyecto de **Keil µVision de la aplicación final**, deberán subirse a Moodle antes del **martes, 19 de diciembre de 2023, a las 21:00 horas.**

## Introducción y objetivos del diseño

El objetivo de este diseño es desarrollar una aplicación completa basada en la tarjeta NUCLEO STM32F429Zi a partir de unas especificaciones generales. Para poder realizar este diseño se necesita conocer cada uno de los elementos abordados (hardware y software) en los bloques anteriores de la asignatura. Además, los estudiantes deberán analizar el funcionamiento y utilizar de forma autónoma algunos elementos no estudiados hasta el momento e incluirlos en la aplicación propuesta.

Como se comprobará más adelante, la especificación del diseño es muy abierta para que se puedan tomar decisiones durante el desarrollo del mismo. Todas aquellas cuestiones que no estén claramente especificadas pueden implementarse del modo que se considere más oportuno, aplicando criterios basados en la simplicidad del diseño y la usabilidad del sistema.

Al igual que en prácticas anteriores, se propondrán algunos ejercicios opcionales que podrán realizarse por parte de los alumnos en función del tiempo disponible y que tendrán influencia sobre la calificación final de la asignatura.

## Descripción del sistema

El diseño que se va a desarrollar es un controlador de una válvula inteligente termostática, utilizadas habitualmente en la calefacción de una vivienda para regular automáticamente la temperatura de los radiadores.

Este sistema se encargará de monitorizar de manera continua la temperatura próxima al radiador, de compararla con una temperatura de referencia y actuar sobre el motor de la válvula para abrir/cerrar proporcionalmente el paso de agua caliente.

Los elementos básicos que se utilizarán para esta aplicación son los siguientes:

- Tarjeta NUCLEO STM32F429Zi.
- Tarjeta mbed App Board.

La figura 1 muestra un esquema simplificado del sistema.

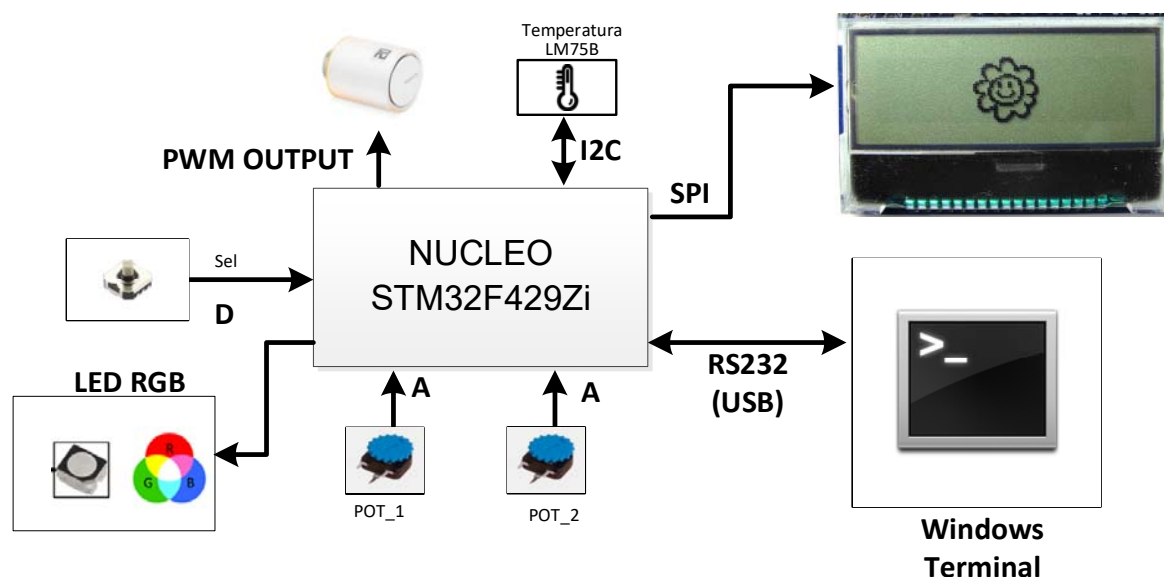


Figura 1

## **Especificaciones**

El funcionamiento del sistema debe responder a las siguientes especificaciones:

- a) Es obligatorio el uso del sistema operativo CMSIS RTOSv2 (RTX). La aplicación se construirá en torno a varios hilos/timers que tendrán que dar soporte a una o varias funcionalidades del sistema.
- b) En todo momento el sistema debe mantener la hora con una resolución de un segundo.
- c) Entradas al sistema:
  - a. Sensor de temperatura, proporcionará medidas de manera periódica.
  - b. Joystick, para poder interactuar con el sistema, el usuario utilizará las diferentes posiciones del joystick. La gestión del joystick se realizará por interrupciones, eliminando los posibles rebotes que éste origine, distinguiendo entre pulsaciones cortas y largas.
  - c. Potenciómetro POT\_1 y POT\_2 cuya funcionalidad se detallará más adelante.
  - d. UART para recibir comandos de programación y depuración.
- d) Salidas del sistema:
  - a. LCD. El usuario podrá conocer el modo del sistema a través de la información proporcionada por el LCD
  - b. LED RGB que mostrará diferentes estados de la electroválvula.
  - c. Señal PWM. Permitirá gobernar el motor que regula el paso de agua caliente al radiador según su ciclo de trabajo. Se visualizará en el osciloscopio/analizador lógico y su frecuencia será 1kHz.
  - d. UART para responder a los comandos de programación y depuración.
- e) El sistema tendrá cuatro modos de trabajo: REPOSO, ACTIVO, TEST, PROGRAMACIÓN/DEPURACIÓN.

Se podrá cambiar de modo en cualquier momento mediante una **pulsación larga** (se considera una pulsación larga aquella que dura más de 1 segundo) del botón CENTER del joystick. El sistema, tras un reset, arrancará en modo REPOSO.

- **Modo REPOSO.**

En este modo se presentará en el LCD el mensaje "SBM 2023" y la hora tal y como se indica en la Figura 2. Tras un *reset* el reloj marcará las 00:00:00. En este modo sólo se atenderá a una pulsación larga del botón CENTER del *joystick*, que hará que el sistema pase a modo ACTIVO.

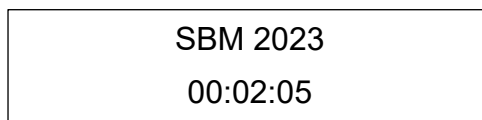


Figura 2. LCD en modo REPOSO

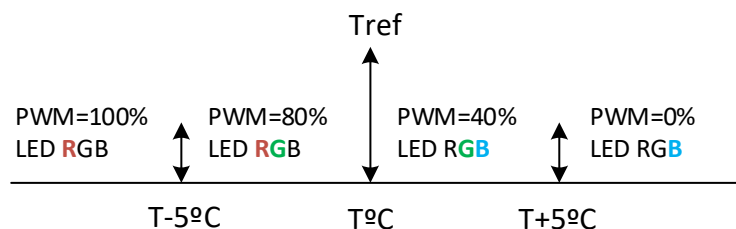
- **Modo ACTIVO.**

En este modo se estará leyendo la temperatura del sensor LM75 cada segundo ( $T_m$ ). Esta temperatura se comparará con la temperatura de referencia ( $T_r$ ) del sistema, que por defecto será de  $25^\circ\text{C}$ , siendo la resolución de éstas  $0.5^\circ\text{C}$ .

Se realizarán las siguientes acciones en función de la diferencia entre ambas temperaturas:

**Tabla 1. Intervalos de control de la electroválvula**

$x = T_r - T_m$	LED RGB	D(%) Señal motor PWM
$+5^\circ\text{C} \leq x$	ROJO	100% (3,3 V)
$0^\circ\text{C} \leq x < +5^\circ\text{C}$	VERDE+ ROJO	80%
$-5^\circ\text{C} < x < +0^\circ\text{C}$	VERDE+ AZUL	40%
$x \leq -5^\circ\text{C}$	AZUL	0% (0 V)



Cuando el sistema se encuentre en el modo ACTIVO, en el LCD debe representarse una información similar a la de la siguiente figura:

ACT---00:02:05---

$T_m:21.5^\circ-T_r:22.5^\circ-D:80\%$

**Figura 3. LCD en modo ACTIVO**

Adicionalmente el sistema debe mantener, cuando se encuentre en este modo, una tabla de 10 medidas almacenadas en un buffer circular. Cada entrada debe mantener en ASCII la siguiente información (hora, temperatura medida, temperatura de referencia y ciclo de trabajo de la señal PWM):

**HH:MM:SS-- $T_m:XX.X^\circ$ -- $T_r:YY.Y^\circ$ --D:ZZ%**

Desde el modo ACTIVO podrá pasarse en cualquier momento al modo TEST pulsando la posición CENTER del joystick.

- **Modo TEST.**

El objetivo de este modo es comprobar el modo ACTIVO, pero en este caso la temperatura medida y la de referencia se van a obtener del potenciómetro 1 y 2 de la tarjeta de aplicaciones, respectivamente. El margen de temperaturas obtenidas en ambos potenciómetros será de  $5.0^\circ\text{C}$  a  $30^\circ\text{C}$  utilizando el máximo giro disponible. En este modo no debe actualizarse el buffer circular, pero sí se debe actualizar el led RGB y generar la señal PWM.

Cuando el sistema se encuentre en el modo TEST, en el LCD debe representarse una información similar a la de la siguiente figura:

TEST---00:02:05---  
 Tm:21.5°-Tr:22.5°-D:80%

Figura 4. LCD en modo TEST

Desde el modo TEST podrá pasarse en cualquier momento al modo PROGRAMACIÓN/DEPURACIÓN pulsando la posición CENTER del joystick.

- **Modo PROGRAMACIÓN/DEPURACIÓN.**

En este modo se programará la hora del sistema y la temperatura de referencia. La modificación de estos valores se realizará utilizando el joystick (pulsaciones cortas), presentando los cambios realizados en el LCD.

---P&D---  
 H: 00:02:05 ---Tr: 22.5°

Figura 5. LCD en modo PROGRAMACIÓN/DEPURACIÓN

Utilizando los gestos UP/DOWN/LEFT/RIGHT se modificarán los valores y para hacerlos efectivos se realizará una pulsación corta en el gesto CENTER. Se deberán validar los valores introducidos para la hora y para la temperatura de referencia (5 a 30°C).

Desde el modo PROGRAMACIÓN/DEPURACIÓN podrá pasarse en cualquier momento al modo REPOSO realizando una pulsación LARGA del gesto CENTER del joystick.

El sistema, además de responder a las peticiones del joystick, se controlará desde un PC a través de un canal serie de comunicaciones RS232. El formato de la comunicación RS232 será 115200 baudios, 8 bits de datos, 1 bit de stop y sin paridad. Es obligatorio que las comunicaciones serie se gestionen por **interrupción/eventos**.

Para el envío de estas tramas/comandos se recomienda el uso del programa **Teraterm**. Dentro de este programa se pueden utilizar macros para enviar tramas, ficheros de texto, en los que se definen cada uno de los bytes que componen dicha trama. En Moodle se encuentra un ejemplo de este tipo de ficheros.

El formato de las tramas de comunicación será el indicado en la Tabla 2.

Tabla 2. Formato de las tramas de comunicación

SOH CMD LEN Payload EOT		
Campo	Byte	Descripción
SOH	0x01	Todas las tramas comienzan con este carácter (Start of Head)
CMD	0xXX	Comando
LEN	0xXX	Longitud <b>TOTAL</b> de la trama, incluidos SOH y EOT.
Payload		Información del comando, puede ser de cualquier longitud e irá codificada en ASCII
EOT	0xFE	Fin de la trama

Las tramas que no estén bien formadas, no respetando el criterio indicado en la tabla anterior, deberán ser ignoradas por el sistema.

El PC podrá enviar diferentes comandos al equipo, especificados en la Tabla 3.

Tabla 3. Comandos del PC al sistema<sup>1</sup>

Comando / CMD	Ejemplo	Comentario
Puesta en hora [0x20]	0x01 <b>0x20</b> 0x0C "15:20:15" 0xFE	Establece la hora a las 15:20:15. Si el equipo acepta la hora, éste devolverá la misma trama con el valor CMD en complemento a uno [0xDF]
Establece el valor de la temperatura de referencia [0x25]	0x01 <b>0x25</b> 0x08 "22.5" 0xFE	Establece el valor de la temperatura de referencia en 22.5°C. El equipo contesta con la misma trama y el valor CMD en complemento a uno [0xDA]
Todas las Medidas [0x55]	0x01 <b>0x55</b> 0x04 0xFE	Solicita todas las medidas almacenadas en el buffer circular. El equipo responde con tantas tramas del tipo "Medida" como medidas tenga almacenadas.
Borrar medidas [0x60]	0x01 <b>0x60</b> 0x04 0xFE	Borra las medidas almacenadas en el buffer circular. Cuando se ha borrado la memoria el equipo responde con la misma trama con el valor CMD en complemento a uno [0x9F]

El sistema podrá responder con las tramas especificadas en la Tabla 4.

Tabla 4. Respuestas del sistema al PC

Respuestas / CMD	Ejemplo	Comentario
Respuesta al comando <i>Puesta en hora</i> [0x20]	0x01 <b>0xDF</b> 0x0C "15:20:15" 0xFE	
Respuesta al comando <i>Establece el valor de la temperatura de referencia</i>	0x01 <b>0xDA</b> 0x08 "22.5" 0xFE	
Medida [0xAF]	0x01 <b>0xAF</b> 0x25 "HH:MM:SS--Tm:XX:X--Tr:YY:Y--D:ZZ%" 0xFE	Devuelve el valor de una medida almacenada.
Respuesta al comando <i>Borrar medidas</i> [0x60]	0x01 <b>0x9F</b> 0x04 0xFE	

<sup>1</sup> Los datos entre comillas ( " ") en esta tabla están representados en ASCII. Las comillas ( " ") no forman parte de la trama.

Estas especificaciones se deben implementar de manera obligatoria para superar el diseño. Todos aquellos detalles que no estén descritos previamente pueden ser implementados a criterio del estudiante aplicando fundamentalmente principios de usabilidad para la aplicación.

La calificación máxima que puede obtenerse implementando las opciones básicas de estas especificaciones es de **8 puntos**. Puede obtenerse una calificación superior implementando alguna de las mejoras que se indican en un apartado posterior.

## Conexiones del sistema

Como puede comprobarse del análisis de las especificaciones del sistema, el mismo estará compuesto por tres elementos diferentes: 1) la tarjeta NUCLEO STM32F429Zi, que incorpora el microcontrolador; 2) la tarjeta *mbed app board*, que proporciona el LCD, el joystick, el LED RGB, el sensor de temperatura, y los potenciómetros; y 3) el PC el cual transmite/recibe tramas serie del sistema.

Las conexiones entre estos elementos deben realizarse OBLIGATORIAMENTE siguiendo las indicaciones proporcionadas en la Tabla 5.

Tabla 5. Conexiones entre la *mbed app board* y la tarjeta NUCLEO STM32F429Zi

		Pin mbed app board	STM32F429	NUCLEO STM32F429Zi
<b>VCC</b>	+ 3.3V	DIP40	+ 3.3V	CN8.7 / +3.3V
<b>GND</b>	GND	DIP1	GND	CN8.11 / GND
<b>LCD (SPI)</b>	MOSI	DIP5	PA7	CN7.14 / D11
	SCK	DIP7	PA5	CN7.10 / D13
	CS	DIP11	PD14	CN7.16 / D10
	A0	DIP8	PF13	CN10.2 / D7
	RESET	DIP6	PA6	CN7.12 / D12
<b>Joystick</b>	Down	DIP12	PE12	CN10.26 / D39
	Left	DIP13	PE14	CN10.28 / D38
	Center	DIP14	PE15	CN10.30 / D37
	Up	DIP15	PB10	CN10.32 / D36
	Right	DIP16	PB11	CN10.34 / D35
<b>Temperature (I2C)</b>	SCL	DIP27	PB8	CN7.2 / D15
	SDA	DIP28	PB9	CN7.4 / D14
<b>LED RGB</b>	R	DIP23	PD13	CN10.19 / D28
	G	DIP24	PD12	CN10.21 / D29
	B	DIP25	PD11	CN10.23 / D30
<b>POT_1</b>		DIP19	PA3	CN9.1 / A0
<b>POT_2</b>		DIP20	PC0	CN9.3 / A1
<b>PWM OUTPUT</b>			PE9	CN10.4 / D6



## Módulos software del sistema

El software del sistema debe desarrollarse de manera modular, de forma que cada uno de los módulos que componen el mismo se encargue de la gestión de un aspecto determinado (tarea, periférico, sensor, funcionalidad, etc). Además de los módulos específicos, deberá desarrollarse un módulo principal que se encargará de la gestión y sincronización de todos ellos. Cada uno de los módulos específicos debe ser autocontenido y tendrá únicamente interacción con el módulo principal, sin posibilidad de ninguna comunicación directa entre módulos específicos que no esté controlada por el módulo principal.

El software correspondiente a cada uno de los módulos debe estar recogido en un fichero que contenga el código (.c) y un fichero de cabecera (.h).

**Se proporciona en Moodle un proyecto con un ejemplo de un módulo de salida (LEDs de la tarjeta NUCLEO) y el programa de test que verifica su funcionamiento.**

Para el desarrollo de la versión básica del sistema con las especificaciones obligatorias se proponen los siguientes módulos, cuya estructura se muestra en la Figura 6.

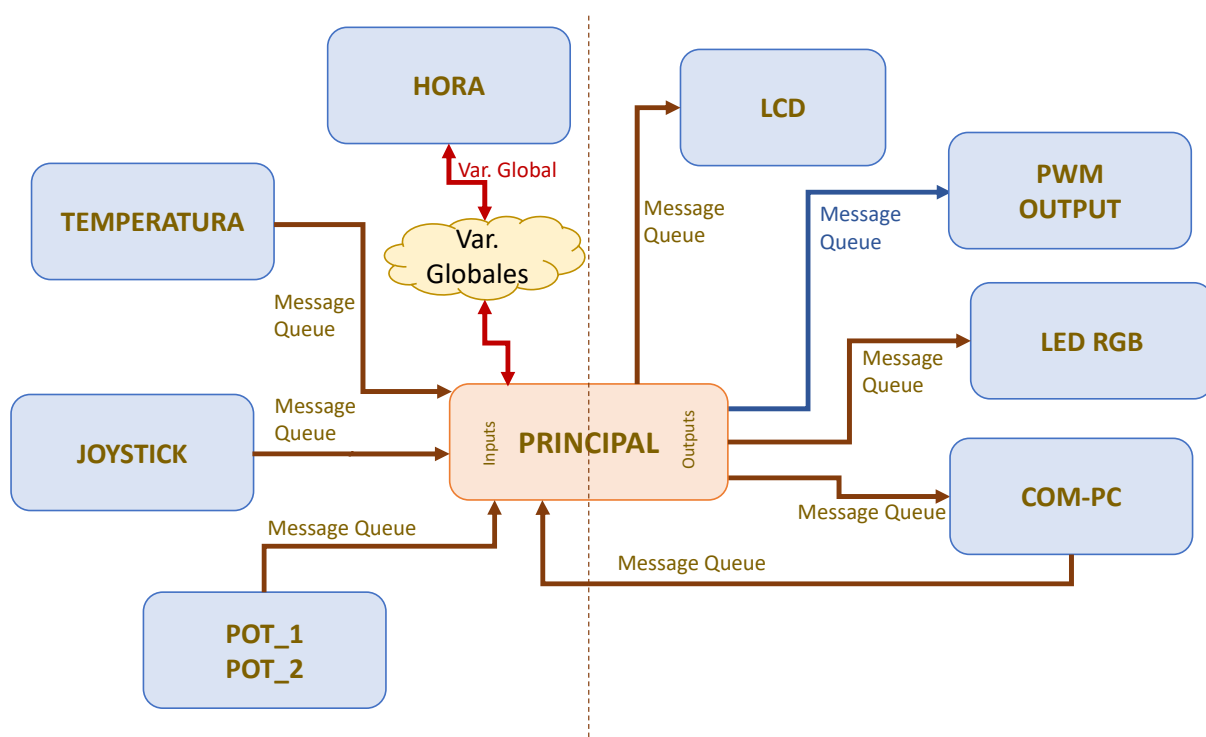


Figura 6. Estructura de los módulos que componen la aplicación

<b>Módulo HORA</b>	Módulo encargado de la gestión del reloj del sistema
<b>Entrada</b>	Ninguna
<b>Salida</b>	Método de sincronización: Variables Globales Actualiza las variables globales “horas”, “minutos” y “segundos”
<b>Ficheros</b>	clock.c y clock.h

<b>Módulo TEMPERATURA</b>	Módulo encargado de la lectura de la temperatura proporcionada por el sensor LM75B conectado al bus I2C
<b>Entrada</b>	Ninguna
<b>Salida</b>	Método de sincronización: Message Queue Envía un mensaje con la temperatura medida cada segundo
<b>Ficheros</b>	temp.c y temp.h (código de ejemplo en <a href="#">mbed</a> )

<b>Módulo JOYSTICK</b>	Módulo encargado de detectar e identificar una pulsación en el joystick
<b>Entrada</b>	Ninguna
<b>Salida</b>	Método de sincronización: Message Queue Cada vez que se detecta una pulsación se envía un mensaje a una cola de mensajes con información de la tecla pulsada y si la pulsación ha sido normal o larga
<b>Ficheros</b>	joystick.c y joystick.h

<b>Módulo POT_1/POT_2</b>	Módulo encargado de leer los valores de temperatura para el modo de test.
<b>Entrada</b>	Ninguna
<b>Salida</b>	Método de sincronización: Message Queue Cada vez que se detecta un cambio de valor se envía un mensaje a una cola de mensajes con información de los nuevos valores seleccionados
<b>Ficheros</b>	pot.c y pot.h (código de ejemplo conversión A/D en Moodle)

<b>Módulo LCD</b>	Módulo encargado de representación de información por el LCD conectado al bus SPI
<b>Entrada</b>	Método de sincronización: Message Queue Thread leyendo de cola de mensajes con la información que debe representarse en el LCD y la línea en que debe representarse
<b>Salida</b>	Ninguna
<b>Ficheros</b>	lcd.c y lcd.h

<b>Módulo OUTPUT</b>	<b>PWM</b>	Módulo encargado de generar una señal PWM
<b>Entrada</b>		Método de sincronización: Message Queue Thread a la espera de cola de mensajes que indique que hay que modificar el ciclo de trabajo de la señal que se aplica a la termoválvula
<b>Salida</b>		Ninguna
<b>Ficheros</b>		pwm.c y pwm.h

<b>Módulo COM-PC</b>		Módulo encargado de la comunicación con el PC a través de la línea serie integrada en el USB
<b>Entrada</b>		Método de sincronización: Message Queue Thread leyendo de una cola de mensajes con la información que se debe enviar al PC (Teraterm)
<b>Salida</b>		Método de sincronización: Message Queue Cada vez que se reciba una trama correcta desde el pc se enviará al módulo principal la información relevante para proporcionar la funcionalidad demandada
<b>Ficheros</b>		com.c y com.h

<b>Módulo PRINCIPAL</b>		Módulo principal del sistema que se encarga de coordinar todos los demás. Es el módulo que decide las acciones a tomar en función del modo del sistema y de la información que reciba del resto de los módulos
<b>Entradas</b>		Métodos de sincronización: Message Queue y Variables Globales Todas las salidas del resto de los módulos
<b>Salidas</b>		Métodos de sincronización: Message Queue y Thread Flags Todas las entradas al resto de los módulos
<b>Ficheros</b>		principal.c y principal.h

## **Mejoras del sistema (opcionales)**

Con el fin de obtener un sistema con más prestaciones y de esta forma poder optar a una mejor calificación del trabajo, cada estudiante puede implementar, de manera opcional, una o varias de las mejoras que se describen a continuación. Para ello será necesario que se consulte con su profesor la forma de abordarlas.

La implementación de estas mejoras puede suponer modificaciones tanto en el código como en la estructura de los módulos ya desarrollados, por lo que se sugiere encarecidamente que se realice una copia de seguridad del sistema con las especificaciones básicas antes de acometer las posibles mejoras.

Cada una de las mejoras implementadas correctamente puede suponer un incremento máximo de la calificación del valor indicado.

También existe la posibilidad de que el estudiante proponga una mejora al margen de las que se detallan, siempre que se consulte antes con su profesor.

Mejoras propuestas:

- Gestión de la señal PWM mejorada. Será proporcional a la diferencia de temperaturas. Además, la iluminación del led RGB tendrá una intensidad variable en función de la diferencia. (+1 punto).
- Programación de diferentes intervalos horarios (al menos dos) con temperaturas de referencia diferentes. (+1 punto)
- Modos de funcionamiento avanzados:
  - ✓ Boost. Permite calentar a máxima temperatura la estancia en un intervalo de 30 minutos. Se enviará un nuevo comando en el modo ACTIVO. (+1 punto)
  - ✓ Detección de ventana abierta, Estando en el modo ACTIVO si se detecta una temperatura menor a un umbral programable durante 30 minutos, se cierra la electroválvula (D=0%). Se enviará un nuevo comando en el modo ACTIVO. (+1 punto)