Índice del documento

Departamento de Ingeniería Telemática y Electrónica

Universidad Politécnica de Madrid

2023-24

Sistemas Basados en Microprocesador

Integración y desarrollo de

una aplicación:

Controlador de una válvula termostática

Alumno:

A: Sara Jiménez Muñoz

B: Antonio Castellana Hajjar

Puesto Nº: x

[1 Objetivos de la PRÁCTICA 2](#_Toc88129291)

[1.1 Resumen de los objetivos de la práctica realizada 2](#_Toc88129292)

[1.2 Acrónimos utilizados 2](#_Toc88129293)

[1.3 Tiempo empleado en la realización de la práctica. 2](#_Toc88129294)

[1.4 Bibliografía utilizada 2](#_Toc88129295)

[1.5 Autoevaluación. 2](#_Toc88129296)

[2 RECURSOS UTILIZADOS DEL MICROCONTROLADOR 3](#_Toc88129297)

[2.1 Diagrama de bloques hardware del sistema. 3](#_Toc88129298)

[2.2 Cálculos realizados y justificación de la solución adoptada. 3](#_Toc88129299)

[3 SOFTWARE 4](#_Toc88129300)

[3.1 Descripción de cada uno de los módulos del sistema 4](#_Toc88129301)

[3.2 Descripción global del funcionamiento de la aplicación. Descripción del autómata con el comportamiento del software (si procede) 4](#_Toc88129302)

[3.3 Descripción de las rutinas más significativas que ha implementado. 4](#_Toc88129303)

[4 DEPURACION Y TEST 5](#_Toc88129304)

[4.1 Pruebas realizadas. 5](#_Toc88129305)

# Objetivos de la PRÁCTICA

T

## Resumen de los objetivos de la práctica realizada

Se deben enumerar los objetivos que ha alcanzado al realizar la práctica. Enumérelos de forma precisa y sencilla.

## Acrónimos utilizados

Identifique los acrónimos usados en su documento.

|  |  |
| --- | --- |
| UART | Universal Asynchronous Receiver Transmitter |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

## Tiempo empleado en la realización de la práctica.

Debe realizar una descripción sencilla del tiempo que ha dedicado a la realización de las actividades relacionadas con la práctica.

|  |  |
| --- | --- |
|  | **[Tiempo empleado para realizer la práctica]:** El tiempo total empleado ha sido de x horas. |

## Bibliografía utilizada

1. Hoja de catalogo xxxx, libro, manual de usuario, etc

## Autoevaluación.

Autoevalúese comprobando los objetivos de aprendizaje indicados en la guía de la asignatura. Compruebe si supera los objetivos de adquisición obligatoria. Si ha encontrado dificultades en la realización de la práctica indicelo.

# RECURSOS UTILIZADOS DEL MICROCONTROLADOR

## Diagrama de bloques hardware del sistema.

Este apartado debe contener un diagrama de bloques donde se identifiquen claramente los elementos utilizados en el diseño o ejercicio. Debe elaborar una figura que muestre todos los elementos utilizados de la tarjeta NUCLEO STM32F429ZI y su interconexión con los elementos externos (tarjeta de aplicaciones, sensores, etc.).

## Cálculos realizados y justificación de la solución adoptada.

En este punto debe describir como ha configurado cada uno de los recursos del microcontrolador, los cálculos que haya realizado y los valores programados en los registros más significativos.

# SOFTWARE

## Descripción de cada uno de los módulos del sistema

**3.1.1 Módulo joy.c y joy.h**

El módulo joy gestiona las pulsaciones del joystick, identificando la dirección (UP, DOWN, LEFT, RIGHT, MIDDLE) y la duración (corta o larga). La información se envía mediante una cola de mensajes al sistema.

**Funcionamiento**

1. **Inicialización**:
   * **init\_Th\_joy**: Crea el hilo del joystick y una cola de mensajes (id\_MsgQueue\_joy).
   * **init\_joy**: Configura los pines GPIO como interrupciones para las direcciones del joystick.
2. **Hilo del joystick (Th\_joy)**:
   * Procesa las interrupciones y clasifica la pulsación:
     + **Antirrebote (REB)**: Usa un temporizador de 50 ms (tim\_reb) para eliminar falsos positivos.
     + **Duración (CHECK)**: Usa un temporizador periódico (tim\_pul) para medir si la pulsación es corta (<1s) o larga (≥1s).
   * Envía un mensaje a la cola con la dirección y duración detectadas.
3. **Interrupciones**:
   * **EXTI15\_10\_IRQHandler**: Detecta la pulsación y llama a HAL\_GPIO\_EXTI\_Callback.
   * **HAL\_GPIO\_EXTI\_Callback**: Determina la dirección usando comprobarPines y activa el hilo con la bandera IRQ.
4. **Temporizadores**:
   * **tim\_reb**: Antirrebote (50 ms).
   * **tim\_pul**: Mide duración de la pulsación.
5. **Estructuras**

* **MSGQUEUE\_OBJ\_JOY**:
  + dir\_t dir: Dirección detectada.
  + uint8\_t dur: 0 (corta), 1 (larga).

**3.1.2 Módulo hora.c y hora.h**

Este módulo implementa un reloj digital que actualiza y muestra la hora en formato **HH:MM:SS**. Utiliza un hilo (Th\_hora) y un temporizador periódico de 1 segundo para gestionar el conteo de segundos, minutos y horas.

**Funcionamiento**

1. **Inicialización**:
   * **Init\_Th\_hora**: Crea el hilo Th\_hora, encargado de manejar la lógica del reloj.
   * **timer\_seg**: Temporizador periódico que genera una interrupción cada segundo y activa el hilo mediante la bandera SEG.
2. **Hilo del reloj (Th\_hora)**:
   * **Inicio del reloj**: Configura la pantalla LCD para mostrar el tiempo.
   * **Ciclo principal**: Espera la bandera SEG para ejecutar la lógica del reloj:
     + Incrementa los **segundos**, habilitando un incremento de minutos cuando llega a 59.
     + Incrementa los **minutos**, habilitando un incremento de horas cuando llega a 59.
     + Incrementa las **horas**, reiniciándolas a 0 al alcanzar 23.
     + Resetea el reloj si se activa la señal hab\_reset.
   * **Actualización del display**: Muestra la hora actual en la pantalla LCD mediante dataToBuffer.
3. **Estructuras y Variables**

* **Variables globales**:
  + seg, min, hor: Almacenan los valores actuales de segundos, minutos y horas.
  + hab\_min, hab\_hor, hab\_reset: Señales para controlar el fin de cada unidad de tiempo y reiniciar el reloj.
* **Temporizador (timer\_seg)**:
  + Genera una interrupción cada 1 segundo para avanzar el tiempo.
* **Pantalla LCD**:
  + linea1 y linea2: Buffers para el texto que se muestra en el display.

**3.1.3 Módulo ace.c y ace.h**

El módulo **ace** gestiona la comunicación I2C con un acelerómetro y un sensor de temperatura, midiendo las aceleraciones en los ejes X, Y, Z, y la temperatura cada segundo. Los datos se envían mediante una cola de mensajes al sistema.

**Funcionamiento**

1. **Inicialización:**

* **init\_Th\_ace**:
  + Inicializa el periférico I2C mediante la función init\_i2c.
  + Crea el hilo principal para la gestión del acelerómetro (id\_Th\_ace).
  + Crea una cola de mensajes para enviar los datos (id\_MsgQueue\_ace).
* **init\_i2c**:  
  Configura y prepara el periférico I2C para la comunicación:
  + Define un callback para notificaciones (i2c\_Callback).
  + Inicializa el controlador I2C (Initialize).
  + Configura el modo de operación, velocidad estándar y limpia posibles errores en el bus.
  + Establece la dirección del esclavo (0x68).

**2. Hilo del Acelerómetro (Th\_ace):**

Realiza las siguientes tareas en bucle:

1. **Configuración inicial del dispositivo**:
   * Escribe valores en registros específicos para activar el sensor (PWR\_MGMT\_1) y configurar el rango del acelerómetro (ACCEL\_CONFIG).
2. **Adquisición de datos**:
   * Solicita y lee las aceleraciones en los ejes X, Y, Z, y los datos de temperatura desde los registros del esclavo I2C.
3. **Procesamiento de datos**:
   * Convierte los datos crudos del sensor a valores de aceleración física (g) y temperatura en grados Celsius.
4. **Temporización**:
   * Usa un temporizador (tim\_1seg) para asegurarse de realizar las mediciones cada segundo.
5. **Comunicación**:
   * Envía los datos procesados a través de la cola de mensajes (id\_MsgQueue\_ace).

**3. Interrupciones:**

* **Callback de I2C (i2c\_Callback)**:
  + Notifica al hilo del acelerómetro cuando una transacción I2C se completa, usando banderas (I2C).
* **Callback del temporizador (tim\_1seg\_Callback)**:
  + Notifica al hilo para que realice una nueva medición después de un segundo, usando banderas (TIM).

**4. Temporizadores:**

* **tim\_1seg**:
  + Temporizador que asegura que las mediciones se realicen cada segundo.

**5. Estructuras**

* **MSGQUEUE\_OBJ\_ACE**:  
  Almacena los datos a enviar a través de la cola de mensajes:
  + **ox, oy, oz**: Aceleraciones en los ejes X, Y y Z en unidades de g.
  + **temp**: Temperatura en grados Celsius.

## Descripción global del funcionamiento de la aplicación. Descripción del autómata con el comportamiento del software (si procede)

En este punto debe explicar el funcionamiento completo de su aplicación. Debe aportar detalles de todos los elementos que forman la aplicación y de cómo interaccionan entre sí. Puede utilizar autómatas para describir el funcionamiento del software.

## Descripción de las rutinas más significativas que ha implementado.

En este punto debe enumerar y describir la funcionalidad de las rutinas más importantes que ha implementado.

# DEPURACION Y TEST

## Pruebas realizadas.

Descripción del método de prueba utilizado para comprobar que las rutinas funcionan adecuadamente. Resultados de los tests. Indicación explicita de si el test es correcto o incorrecto. En este punto debe hace especial hincapié en definir:

1. Cuál es el objetivo de la prueba, indicando los módulos implicados
2. Cuál es el proyecto de Keil que permite realizar la prueba
3. Cuáles son las condiciones de entrada que permiten ejecutar la prueba
4. Cuáles son los resultados que se esperan y cuáles son los realmente obtenidos.