Índice del documento

Departamento de Ingeniería Telemática y Electrónica

Universidad Politécnica de Madrid

2022-23

Sistemas Basados en Microprocesador

Integración y desarrollo de

una aplicación:

Sintonizador de Frecuencia Modulada (FM)

Alumno:

A:…………………………..

B:…………………………..

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 |
| USART |  |
| I2C |  |
| GPIO |  |
| LCD |  |
| SPI |  |
| PWM |  |
| ADC |  |
| RS232 |  |
| BR | Baud Rate |
|  |  |
|  |  |

## 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. |

Arreglo Joystick respecto B2\_P6 55 minutos + 15 test + 0 Debug

Arreglo LCD respecto B2\_P4 20 mintuos + 10 test + 15 Debug

## 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

En este punto debe explicar el funcionamiento de cada uno de los módulos que haya desarrollado.

* LCD: Módulo que se encarga de toda la gestión interna con el *display* LCD, desde la inicialización de este, hasta el dibujado de caracteres en este. Toda la comunicación entre el LCD y el microcontrolador se lleva a cabo a través del protocolo de comunicación SPI. El módulo esta desarrollado bajo la premisa, de ser un modulo de salida, es decir, no adquiere datos, simplemente representa lo que buenamente haya sido programado. Para comunicarse con el exterior, este modulo hace uso de una cola de mensajes. El mensaje que se manda por esta cola contiene 4 parámetros: “Data\_L1” y “Data\_L2” son ambos un array de tipo *char* de 32 de tamaño, en estos parámetros se envían las cadenas de caracteres (*string literals*) que se quiere mostrar, tanto en la primera como en la segunda línea del *display*; “Init\_L1” e “Init\_L2” son valores enteros sin signo, de 8 bits de resolución (uint8\_t) que simbolizan el offset, dentro de cada línea, a partir del cual se empezará a escribir. El usuario final de este módulo solo debe tener en cuenta, que si la cadena de caracteres a representar en una línea, más el offset correspondiente que se le haya asignado, sobrepasa el tamaño máximo de cada línea, la representación va a resultar inexacta, llegándose a mezclar los diferentes caracteres entre sí. En este caso, se debería acortar la cadena a representar, reducir el offset introducido, o ambas cuestiones a la vez.
* Joystick:

## 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.

Para realizar todas las pruebas mencionadas más adelante, se deberán seguir estos pasos:

1. Incluir en Keil lo siguientes archivos:
   1. El módulo bajo estudio, añadir el “.c” (eg: pwm.h + pwm.c).
   2. El módulo test cuyo nombre empieza con el nombre a estudiar seguido de “\_test “, añadir el “.c” (eg: pwm\_test.h + pwm \_test.c).
   3. Algunos test requieren de incluir también el modulo del LCD, en aquellos en los que así se requiere, viene especificado EXPRESAMENTE en un comentario, en su correspondiente “.h”
2. Añadir en el “main.c” la cabecera del módulo del test (eg: #include “pwm\_test.h”) tanto el modulo bajo estudio, como en caso de necesitarlo, el módulo del lcd, se gestionan dentro del test. Se hace esto, bajo la premisa de que el usuario final que vaya a realizar los tests, tenga que preocuparse lo mínimo posible.
3. Añadir en el “main.c” y entre las órdenes “osKernelInitialize()” y “osKernelStart()”, el inicio del hilo correspondiente del test (eg: Init\_Th\_pwm\_test()). Tanto el hilo del modulo bajo estudio, como en caso de necesitarlo, el modulo del lcd, se inician dentro del test. Se hace esto, bajo la premisa de que el usuario final que vaya a realizar los tests, tenga que preocuparse lo mínimo posible.

* LCD: en este test, se mira el correcto funcionamiento del LCD al completo. Se prueba el uso de diferentes offsets, se comprueba que al reescribir frases cortas sobre frases largas, estas ultimas se borran por completo.