

SISTEMAS BASADOS EN MICROPROCESADOR

TRABAJO FINAL DE PRÁCTICAS

Departamento de Tecnología Electrónica

11 de julio de 2025

Índice

1. Introducción	1
2. Relación de datos (Variables)	1
3. Descripción de módulos funcionales	2

1. Introducción

El presente documento detalla la arquitectura de software del proyecto de la asignatura mediante un microcontrolador MSP430. El objetivo de esta documentación es proporcionar una referencia técnica completa que facilite la depuración y la legibilidad del sistema.

El análisis se divide en dos secciones principales:

- **Relación de datos:** Una descripción de todas las variables de memoria utilizadas en el programa.
- **Descripción de Módulos Funcionales:** Un análisis detallado de la interfaz y el propósito de cada función, tanto en C como en Ensamblador.

2. Relación de datos (Variables)

La siguiente tabla presenta una relación alfabética de todas las variables de memoria (globales, locales y estáticas) definidas en el proyecto.

Cuadro 1: Relación de variables del programa

Variable	Propósito y significado de Valores	Módulo de definición	Función(es) de escritura	Función(es) de lectura
char1, char2	Variables locales para almacenar los caracteres ASCII de las decenas y unidades de un número.	main.c (local en CambiaLCD)	CambiaLCD	CambiaLCD
contador	Índice estático para recorrer el <code>mensaje</code> de bienvenida. Se reinicia a 0 al final del mensaje.	main.c (estática en AnimaIco)	AnimaIco	AnimaIco
Diferencia	Almacena el tiempo medido de pulsación (escalado) para ser mostrado y comparado.	main.c (local en main)	main	main
estado_pulsador	Almacena el valor de <code>puls()</code> para decidir la acción a tomar en el bucle principal.	main.c (local en main)	main	main
i	Variable de bucle para generar pausas bloqueantes por software (busy-waiting).	main.c (local en pausa)	pausa, pausa_debounce	pausa, pausa_debounce
mensaje	Array de caracteres (string) con el texto de bienvenida. Almacenado en memoria de programa (ROM).	main.c (global)	N/A (Inicialización en compilación)	AnimaIco
objetivo	Almacena el número pseudoaleatorio (10-99) que el usuario debe igualar.	main.c (local en main)	main	main

Cuadro 1 – continuación

Variable	Propósito y significado de valores	Módulo de definición	Función(es) de escritura	Función(es) de lectura
<code>p</code>	Puntero estático al carácter actual del <code>mensaje</code> de bienvenida.	<code>main.c</code> (estática en <code>AnimaIco</code>)	N/A (Inicializado una vez)	<code>AnimaIco</code>
<code>Resultado</code>	Almacena la diferencia final entre <code>Diferencia</code> y <code>objetivo</code> .	<code>main.c</code> (local en <code>main</code>)	<code>main</code>	<code>main</code>
<code>SystemTimer</code>	Contador de 32 bits que constituye la base de tiempo del sistema. Es incrementado por una ISR.	<code>syst.asm</code> (<code>.bss</code> , global)	<code>systA2ISR</code>	<code>systTim</code>
<code>systPeriodo</code>	Almacena el valor del periodo del Timer A2 para programar la siguiente interrupción.	<code>syst.asm</code> (<code>.bss</code> , global)	<code>systIni</code>	<code>systA2ISR</code>
<code>Tab14Seg</code>	Tabla de búsqueda en ROM para convertir ASCII a los bits de un display de 14 segmentos.	<code>lcd.asm</code> (Sección de datos)	N/A (Definida en compilación)	<code>lcd2aseg</code>
<code>tfinal</code>	Almacena el timestamp (<code>systTim</code>) al momento de soltar el pulsador SW2.	<code>main.c</code> (local en <code>main</code>)	<code>main</code>	<code>main</code>
<code>tiempo_inicial</code>	Almacena el timestamp al inicio de <code>pausa_milisegundos</code> para medir el tiempo transcurrido.	<code>main.c</code> (local en <code>pausa_...</code>)	<code>pausa_milisegundos</code>	<code>pausa_milisegundos</code>
<code>tinicial</code>	Almacena el timestamp (<code>systTim</code>) al momento de presionar SW2 para iniciar la medición.	<code>main.c</code> (local en <code>main</code>)	<code>main</code>	<code>main</code>
<code>TpulsacionS2</code>	Variable local para capturar y devolver el valor actual de <code>systTim</code> .	<code>main.c</code> (local en <code>Midetsw2</code>)	<code>Midetsw2</code>	<code>Midetsw2</code>
<code>variable_local</code>	Variable cuya dirección de memoria se usa como fuente para la generación de números aleatorios.	<code>main.c</code> (local en <code>GeneraNum</code>)	<code>GeneraNum</code>	<code>GeneraNum</code> (vía <code>'&'</code>)

3. Descripción de módulos funcionales

La siguiente tabla detalla la interfaz de programación de cada una de las funciones utilizadas en el proyecto. Se especifica su propósito, ubicación, parámetros de E/S y las relaciones de llamadas entre ellas.

Cuadro 2: Relación de funciones del programa

Función	Propósito	Módulo	Parámetros entrada	Parámetros salida	Llamada por
AnimaIco	Muestra un carácter del mensaje de bienvenida y avanza la posición (efecto scroll).	main.c	void	void	main
CambiaLCD	Descompone un número y lo muestra en una posición específica del LCD.	main.c	long numero: Valor a mostrar. int estado: Posición ('1, 2, 3').	void	main
csIniLf	(ASM) Inicializa el Clock System para usar el oscilador de baja frecuencia (LFXT).	cs.asm	void	void	main
GeneraNum	Crea un número pseudoaleatorio entre 10 y 99.	main.c	void	long: Número aleatorio.	main
lcd2aseg	(ASM) Convierte dos caracteres ASCII a códigos de 14 segmentos.	lcd.asm	R12, R13: Caracteres ASCII.	R12, R13: Códigos de 16-bits. (-1 si error).	Pintar, lcdLPut
lcdClear	(ASM) Borra los 6 dígitos principales del LCD.	lcd.asm	void	void	AnimaIco
lcdClearAll	(ASM) Borra toda la memoria del LCD, incluyendo iconos.	lcd.asm	void	void	main
lcdClearSW	(ASM) Borra los dígitos de resultado y diferencia, dejando visible el objetivo.	lcd.asm	void	void	main
lcdIni	(ASM) Inicializa el controlador del LCD.	lcd.asm	void	void	main
lcdLPut	(ASM) Muestra un carácter y desplaza los demás a la izquierda (efecto scroll).	lcd.asm	R12: Carácter ASCII.	void	AnimaIco

Cuadro 2 – continuación

Función	Propósito	Módulo	Parámetros entrada	Parámetros salida	Llamada por
main	Inicialización del sistema y los bucles de bienvenida y de juego.	main.c	void	int: 0 (Nunca retorna).	(Sistema)
Midetsw2	Captura y devuelve un timestamp del contador del sistema.	main.c	void	long: Valor del timer.	main
P1ISR	(ISR) Maneja la interrupción del Puerto 1 y actualiza R12 con el estado del pulsador.	syst.asm	(Hardware)	Efecto secundario en R12 .	Hardware
Pintar	(ASM) Pinta dos códigos de 14 segmentos en una posición del LCD.	lcd.asm	R12, R13: Códigos. R14: Posición.	void	CambiaLCD
pausa	Generan pausas bloqueantes de corta y media duración.	main.c	void	void	main
pausa_milisegundos	Genera una pausa precisa en milisegundos usando el timer del sistema.	main.c	unsigned int ms: Duración.	void	main
puls	(ASM) Habilita interrupciones del P1 y retorna el estado del último pulsador accionado.	syst.asm	void	int (en R12): '1, 2, 3'.	main
pulsIni	(ASM) Inicializa los pines de los pulsadores SW1 y SW2.	syst.asm	void	void	main
systA2ISR	(ISR) Maneja la interrupción del Timer A2, incrementando el contador SystemTimer .	syst.asm	(Hardware)	Efecto secundario en SystemTimer .	Hardware

Cuadro 2 – continuación

Función	Propósito	Módulo	Parámetros entrada	Parámetros salida	Llamada por
<code>systIni</code>	(ASM) Inicializa el Timer A2, su interrupción, y las interrupciones globales.	<code>syst.asm</code>	R12 : Periodo del timer.	int (No usado).	main
<code>systTim</code>	(ASM) Devuelve de forma segura el valor del contador <code>SystemTimer</code> .	<code>syst.asm</code>	void	unsigned long (en R12 , R13).	(Varias)