

RECEPCIÓN Y TRANSMISIÓN DE DATOS MEDIANTE UART

Sincronización de los dispositivos (encuesta o interrupción)

- **Pulsador S4:** Se gestiona por encuesta
- **Pulsadores S3, S6:** Se gestionan por interrupción
- **Timer T9:** Se gestiona por encuesta
- **Timers T7, T5:** Se gestionan por interrupción

Funciones que llama el programa principal

- **inic_oscilator** (en oscilator.c)
 - Inicializa y configura el módulo oscilator
- **inic_leds** (GPIO.c)
 - Define e inicializa los puertos correspondientes a los leds
- **inic_pulsadores** (GPIO.c)
 - Define e inicializa los puertos correspondientes a los pulsadores
- **inic_LCD** (LCD.c)
 - inicializa el display LCD estableciendo los estados iniciales para data y pines de control e inicializa la secuencia de inicialización de la LCD
- **copiar_FLASH_RAM** (memoria.c)
 - Recibe una cadena de texto y asigna sus valores a la variable Ventana_LCD la cual se usa para mandar datos al display
- **line_1** (LCD.c)
 - Establece el cursor en la primera línea del display
- **puts_lcd** (LCD.c)
 - Recibe la variable Ventana_LCD y despliega su información en el display LCD
- **line_2** (LCD.c)
 - Establece el cursor en la segunda línea del display
- **inic_crono** (timers.c)
 - Inicializa las variables de cronometro y cambia el valor del flag_inic_crono a 1
- **inic_Timer7** (timers.c)
 - Inicializa y configura el Timer 7
- **inic_Timer5** (timers.c)
 - Inicializa y configura el Timer 5
- **inic_CN** (CN.c)
 - Inicializa el módulo CN
- **Inic_UART2** (UART_RS232.c)
 - Inicializa el módulo UART2, para recibir y transmitir información.
- **crono** (timers.c)
 - Realiza el conteo de las variables del cronometro, en cada cambio de valor de una de estas variables se realiza la conversión de tiempo de número entero a ASCII con la función conversión_tiempo de modo que la variable Ventana_LCD actualiza su contenido.

- **comprobar_inic_crono** (timers.c)
 - Evalúa el cambio del flag_inic_crono y realiza la conversión del tiempo de las variables del cronometro, esto se realiza para que estas operaciones no se hagan directamente en la rutina de atención

Otras funciones importantes

- **lcd_cmd** (en LCD.c)
 - Se encarga de enviar los comandos necesarios para el funcionamiento de la pantalla LCD.
- **lcd_data** (en LCD.c)
 - Subrutina encargada de imprimir un carácter en la pantalla LCD.
- **Delay_ms** (en timers.c)
 - Función que espera el tiempo pasado por parámetro en milisegundos.
- **Delay_us** (en timers.c)
 - Función que espera el tiempo pasado por parámetro en microsegundos.
- **conversion_tiempo** (en utilidades.c)
 - Función que, dados dos dígitos, devuelve estos en forma de caracteres.

Rutinas de atención

- **Rutina del módulo CN** (en CN.c): `_CNInterrupt()`
 - Comprueba el pulsador S3 y si está pulsado para o pone en marcha el timer 7.
 - Comprueba el pulsador S6 y si está pulsado reinicia el cronometro y para el timer 7.
- **Rutina del timer T7** (en timers.c): `_T7Interrupt()`
 - Simplemente actualiza la variable mili cada 10ms e inhabilita el flag de interrupción.
- **Rutina del timer T5** (en timers.c): `_T5Interrupt()`
 - Se implementa la maquina de estados para que envíe datos y comandos al LCD cada 2,5ms.
- **Rutina de la UART2** (en UART2_RS232.c): `_U2RXInterrupt()`
 - Evalúa el registro recepción de la UART2, si el carácter recibido es "p" o "P" para el cronometro, si es "i" o "I" lo inicializa y si es "c" o "C" lo pone en marcha.
- **Rutina de la UART2** (en UART2_RS232.c): `_U2TXInterrupt()`
 - Contiene una máquina de estados que enviará los datos y comandos necesarios a una velocidad 4800 baudios mediante la UART2 (10bits) y se verán reflejados en el terminal del Tera Term.

FUNCIONAMIENTO GENERAL

El funcionamiento para el timer y LCD son los mismos que para las prácticas anteriores. Esta práctica se diferencia en que esta vez se usa la UART2 para enviar y recibir los datos para visualizar y manipular el cronometro.

La UART2 recogerá los datos recibidos mediante Tera Term y evaluará el carácter recibido mediante la rutina de atención de `_U2RXInterrupt()` con el que se modificará el comportamiento del cronómetro.

También se enviarán datos a través de la UART2 de lo que se encargará la rutina de atención de `_U2TXInterrupt()`. Se ha determinado que la mejor frecuencia para enviar los datos es 4800

baudios, ya que con menos frecuencia perderíamos datos a la hora de refrescar y con más, estaríamos desperdiciando recursos. En total, se enviarán $(3+16+2+16) \times 10 \text{ bits} = 37 \times 10 \text{ bits} = 370 \text{ bits}$ a una velocidad de transmisión de 22 us/bit .