

Sistemas Empotrados: Trabajo Tutelado

Gonzalo Gilvalde Blanco

April 2024

1 Introducción

Este documento fue creado por la necesidad de proporcionar una guía más detallada que los comentarios dentro del código, los cuales pueden resultar confusos. Esta guía se divide en dos secciones principales: guía de usuario y explicación por bloques del código.

2 Guía de Usuario

2.1 Inicio

Una vez que la placa esté conectada a la corriente, se observará que el LED rojo se enciende y que el LCD muestra 00:00.

2.2 Modificación de Segundos

El LED rojo encendido indica que estamos en la etapa de modificación de los segundos.

Ambos botones, sw1 y sw2, están disponibles. Para fines de esta guía, referiremos a sw1 como el botón derecho y a sw2 como el izquierdo.

Al mantener pulsado el botón derecho, los segundos aumentarán de uno en uno con un pequeño retraso de aproximadamente 1 segundo. Si deseas aumentar solo un segundo, espera 1 segundo y luego verás cómo se refleja en el LCD. Puedes mantener el botón pulsado para continuar aumentando los segundos.

Al mantener pulsado el botón izquierdo, los segundos disminuirán de uno en uno, con un funcionamiento similar al aumento.

Para cambiar al estado de modificación de minutos, hay dos opciones: mientras mantienes pulsado el botón derecho, mantén también el izquierdo, o viceversa. Sin embargo, esta forma de cambiar de estado conlleva un problema, el cual será abordado junto a todos los errores en la sección 2.6.

2.3 Modificación de Minutos

Al cambiar de estado, notarás que el LED rojo se apaga y el LED verde se enciende, indicando que el cambio de fase se ha realizado correctamente.

Es importante levantar los dedos rápidamente después de este cambio de estado, ya que mantener los botones pulsados podría provocar otro cambio no deseado.

El proceso para aumentar y disminuir los minutos es idéntico al de los segundos: mantén pulsado uno de los botones para aumentar o disminuir los minutos, y cambia de fase manteniendo pulsados ambos botones.

2.4 Cuenta Regresiva

Al cambiar de estado, notarás que los LEDs alternan entre encendido y apagado, y los segundos disminuyen automáticamente. Si no observas este comportamiento, ni el cambio en los LEDs ni la disminución en el LCD, hay dos posibilidades:

1. La cuenta regresiva se ha pausado al mantener pulsados los botones, por lo que debes reanudarla presionando el botón izquierdo.
2. El cambio de fase no se ha realizado correctamente, así que verifica si los minutos aumentan o disminuyen al mantener pulsado uno de los botones.

Durante esta etapa, puedes pausar y reanudar la cuenta regresiva. Para pausarla, basta con pulsar brevemente el botón derecho. Para reanudarla, utiliza el botón izquierdo de la misma manera.

Para pasar a la última etapa, debes esperar a que el contador llegue a 00:00.

2.5 Final del Contador

Al finalizar el contador, tienes dos opciones:

1. Presionar el botón derecho para reiniciar el contador desde el tiempo establecido al finalizar la modificación de los minutos.
2. Presionar el botón izquierdo para volver al punto 2.2 y modificar los segundos, y luego los minutos.

2.6 Errores

Durante todo el proceso, pueden surgir algunos problemas que se abordarán en este punto:

1. Para pasar a la siguiente etapa después de modificar los segundos o los minutos, es necesario presionar el botón contrario y esperar un breve período, lo que podría resultar en un segundo o un minuto adicional o menos. Es una cuestión de precisión que debería corregirse.
2. Durante el cambio de etapa de segundos a minutos o de minutos al contador, es posible que mantengas presionados los botones por demasiado tiempo, lo que provocaría un cambio no deseado de fase. Si esto sucede en el cambio de segundos a minutos, el contador pasará directamente a la cuenta regresiva.

3. Durante la ejecución del contador, pausar y reanudar puede hacer que el tiempo avance más rápido de lo normal. Este comportamiento podría no ser necesariamente un error, ya que podría deberse a la compensación del tiempo transcurrido al pausar y reanudar.
4. Al reiniciar el contador desde el tiempo establecido originalmente, siempre comienza en estado pausado, lo cual podría ser confuso para algunos usuarios, aunque no es un error en sí mismo.

3 Código

3.1 Introducción

Esta sección tiene como objetivo aclarar ciertos aspectos del código que podrían resultar confusos.

3.2 Macros

Dentro del código, se emplean varias macros, principalmente para el manejo de los LEDs y los botones.

3.2.1 LEDs

Las macros y constantes relacionadas con los LEDs son las siguientes:

- `LED_GREEN_PIN` y `LED_RED_PIN`: Constantes que representan los pines de los LEDs.
- Las cuatro macros siguientes se utilizan para encender o apagar los LEDs correspondientes.
- `LED_TOGGLE`: Función para cambiar el estado de encendido/apagado de los LEDs durante la cuenta regresiva.

3.2.2 Botones

En cuanto a los botones:

- Las constantes representan los pines de los botones.
- Las funciones se encargan de verificar si los botones están siendo presionados o no.

3.2.3 Incrementar y Decrementar el Contador

Aunque estas funciones no son macros, merecen una mención especial dentro del código.

Inicialmente, eran macros que realizaban la misma tarea que la versión final del código: aumentar el contador y, si alcanzaba 60, reiniciarlo a 00; o disminuir el contador y, si llegaba a -1, ajustarlo a 59.

Las macros originales eran las siguientes:

```
#define INC_COUNTER(counter) do {  
    (counter) = ((counter) + 1) % (MINUTES_MAX + 1);  
} while(0)  
  
#define DEC_COUNTER(counter) do {  
    if ((counter) == 0) (counter) = MINUTES_MAX + 1;  
    (counter) = ((counter) - 1) & 0x3F;  
} while(0)
```

Aunque funcionaban correctamente, por razones personales decidí reimplementar estas funciones utilizando ensamblador, ya que encontré que su traducción era bastante sencilla y fue un tema de trabajos anteriores.

El código resultante quedó así:

```
#define MINUTES_MAX 59  
  
void INC_COUNTER(int *counter) {  
    asm volatile (  
        "ldr r1, [%0]\n"  
        "add r1, r1, #1\n"  
        "cmp r1, %1\n"  
        "ble .done\n"  
        "movs r1, #0\n"  
        ".done:\n"  
        "str r1, [%0]\n"  
        :  
        : "r" (counter), "i" (MINUTES_MAX)  
        : "r1"  
    );  
}  
  
void DEC_COUNTER(int *counter) {  
    asm volatile (  
        "ldr r1, [%0]\n"  
        "cmp r1, #0\n"  
        "ble .reset\n"  
        "sub r1, r1, #1\n"  
        "b .mask\n"
```

```

        ".reset:\n"
        "mov r1, %1\n"
        ".mask:\n"
        "mov r2, #0x3F\n"
        "and r1, r1, r2\n"
        "str r1, [%0]\n"
        :
        : "r" (counter), "i" (MINUTES_MAX)
        : "r1", "r2"
    );
}

```

4 Lógica general del programa

El código puede resultar un tanto confuso a primera vista, por lo que en lugar de mostrar el código directamente, explicaré la lógica detrás del mismo.

El programa comienza inicializando el LCD, los botones y los LEDs, así como configurando las interrupciones de los botones. El tiempo se establece en el LCD como 00:00 y luego entra en un bucle infinito.

Se utiliza una variable global llamada **estado**, inicializada en 0. Mientras **estado** sea 0, el programa está en modo de modificación de segundos, invocando a las funciones **INC_COUNTER** o **DEC_COUNTER** según el botón presionado.

Si ambos botones se mantienen presionados, lo que indica un cambio de estado, se incrementa la variable **estado** en 1, lo que lleva a modificar los minutos. El proceso es similar al de los segundos.

Una vez se presionan ambos botones, se guardan las variables relacionadas con segundos y minutos en otras variables para poder reiniciar el contador en el futuro. Antes de cambiar **estado** a 2, se llama a la función **confpit()**, que desactiva las interrupciones de los botones y activa el PIT (Programmable Interval Timer), tanto el contador como las interrupciones. En este punto, el programa se traslada a la función de interrupciones **PIT_IRQ_HANDLER(void)**.

Inicialmente, esta función simplemente decrementa los segundos o los minutos según corresponda, actuando como un cronómetro regresivo.

Cuando se presiona el botón de pausa, se desactivan las interrupciones del PIT y se vuelven a activar las de los botones. En el estado 2, solo es posible reanudar la cuenta o volver a llamar a **confpit()**. Esta decisión, aunque compleja, se tomó debido a problemas para utilizar **PIT_StopTimer** para detener el temporizador desde ese lado.

Cuando el contador finaliza, es decir, cuando **estado** alcanza el valor 3, se activan las interrupciones de los botones y se desactivan las del PIT. Esto nos lleva a dos opciones: reiniciar el contador con los minutos y segundos previamente guardados, o volver a modificar los segundos y minutos.

Para facilitar la comprensión, se presenta un diagrama de flujo que muestra el estado a estado del programa.

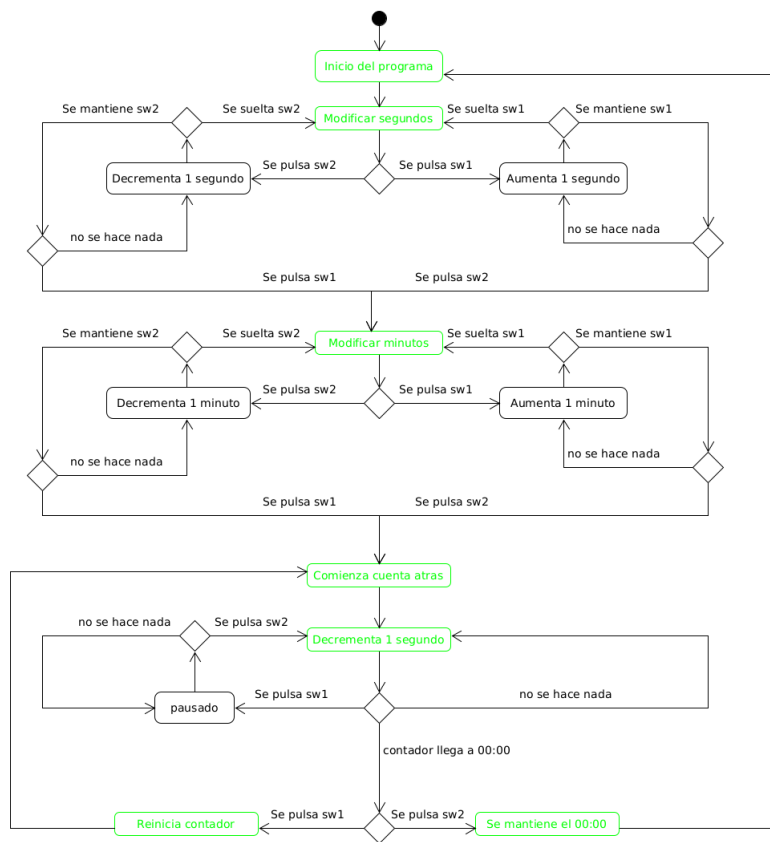


Figure 1: Diagrama de flujo del programa.

Gracias por leer hasta aquí y disculpa por no condensar todo esto en el código y tener estos errores arreglados, por la falta de tiempo por todas las asignaturas y el tfg no puedo trabajar tanto en este trabajo tutelado como me gustaría. Espero que esta explicación te haya sido útil.