

Universidad de Costa Rica

Microprocesadores

Tarea 05

Yoel Moya Carmona - B75262

17 de Noviembre de 2023

Índice

1. Descripción del diseño	2
1.1. Configuración de Hardware	2
1.1.1. Output Compare	2
1.1.2. Pantalla LCD	2
1.2. Programa Principal	2
1.3. Tarea LCD	2
1.4. Tarea Send LCD	3
1.5. Led Testigo	3
1.6. Tarea Conversión	3
1.7. Sub Rutina BIN BCD MUXP	3
1.8. Sub Rutina BCD 7 Segmentos	3
1.9. Tarea Pantalla MUX	3
1.10. Tarea TCM	4
1.11. Rutina de Atención a Interrupciones	4
1.12. Tabla de Timers	4
2. Anexos	6
2.1. Configuración de Hardware	6
2.1.1. Output Compare	6
2.1.2. Pantalla LCD	7
2.2. Programa Principal	8
2.3. Tarea LCD	9
2.4. Tarea Send LCD	10
2.5. Led Testigo	12
2.6. Tarea Conversión	13
2.7. Sub Rutina BIN BCD MUXP	14
2.8. Sub Rutina BCD 7 Segmentos	15
2.9. Tarea Pantalla MUX	16
2.10. Tarea TCM	18
2.11. Rutina de Atención a Interrupciones	19

1. Descripción del diseño

Se dividió en secciones todo lo descrito, con todos los diagramas de flujo de cada tarea o maquina de estados en la sección de 2

1.1. Configuración de Hardware

Los diagramas de la Configuración del hardware de la tarea 5 unicamente, se encuentran en la sección 2.1

1.1.1. Output Compare

El diagrama de la interrupción se encuentra en la sección 2.1.1

Se utilizó un preescalador de 1, el cual se determina con \$00, y se le suma 480 para poder obtener los 20uS de interrupción deseados:

$$TC4 = \frac{Clk_{BUS} \cdot T_{OC}}{PRS} = \frac{24 \times 10^6 \cdot 20 \times 10^{-6}}{1} = 480$$

1.1.2. Pantalla LCD

El diagrama de flujo de la inicialización de la pantalla LCD se encuentra en la seccion 2.1.2

Este inicia habilitando el puerto K como salida, iniciando los timers de la maquina de tiempos de las bases que contienen los contadores que se necesitan durante la inicialización y durante el llamado a la tarea **Send_LCD**. Se carga el primer estado de dicha tarea, se borran todas las banderas, ya que en este punto no se necesita ninguna, y se cargan el puntero de LCD y el índice Y con la dirección de la tabla de los comandos de la configuración.

Seguidamente en un ciclo se carga el dato a enviar, se revisa si es fin de tabla, y se decide si enviar el comando de clear, o si todavía quedan datos por enviar. La tarea **Send_LCD** se llama cíclicamente para que esta pueda avanzar a través de todos sus estados, y una vez que se envían todos los datos, se esperan 2mS y se continua con el programa.

1.2. Programa Principal

El diagrama de flujo del programa principal se encuentra en la sección 2.2.

Acá no se presenta todo el programa principal sino únicamente las cosas que se consideraron importantes para comprender el funcionamiento de la tarea 5. Inicialmente se cargan todos los timers de la maquina de tiempos, se borran todas las banderas, se inicializa el stack y se habilitan las interrupciones.

Se carga la variable de brillo para que esta sea inicializada en algún valor, se carga el mensaje de bienvenida y se envía a la pantalla LCD borrando la bandera **ARRAY_OK** para que este mensaje aparezca en el *power-up reset* cuando se corra el programa. Seguido de esto se entra al despachador de tareas, el cual únicamente la **Tarea_LCD** se accede condicionalmente con el borrado de la bandera anteriormente mencionada. Esta bandera, luego de entrar al despachador de tareas, puede ser borrada únicamente por la **Tarea_TCM**.

1.3. Tarea LCD

Los diagramas de flujo de esta tarea se encuentran en la sección 2.3.

Esta tarea posee únicamente dos estados, los cuales el primero envía comandos de configuración a la pantalla, así como la selección de la linea en la cual se pretende escribir. Utiliza la bandera **Second_Line** y RS para tomar estas decisiones. Los punteros **Msg_L1** y **L2** determinan la dirección de la tabla de datos que va para la linea 1 y linea 2. Estos son pasados al siguiente estado para poder ser utilizados.

El segundo estado de esta maquina, es el que se encarga de enviar los datos a la pantalla hasta que se encuentre el final de la tabla. Se utiliza la bandera **Second_Line** para determinar si el final de la tabla es únicamente de la linea 1 o del mensaje completo.

Similar a la configuración de hardware, la tarea **Send_LCD** se debe llamar repetidamente para que esta evolucione.

1.4. Tarea Send LCD

Los diagramas de flujo de esta tarea se encuentran en la sección 2.4.

Esta tarea se encarga de implementar el protocolo estroboscópico que utiliza la pantalla LCD para comunicarse con el microprocesador. Esta tarea se encarga de, ubicar los datos entre los bits 2 y 5 del puerto K, determinar según la bandera **RS** si es un dato o un comando, y enviar un pulso de **EN** al bit 1 por un tiempo de 260uS. El fin de este tiempo determina el fin del primer pulso para enviar la parte alta de los datos, donde seguidamente se ubica la parte baja de ellos en el puerto K, se determina el tipo (dato o comando con **RS**) y se envía un segundo pulso de la misma longitud de tiempo por **EN**. El estado 1 se encarga de enviar la parte alta, y el estado 2 se encarga de evaluar cuando termino de enviarse la parte alta para enviar la parte baja.

El estado 3 y 4 se encargan de evaluar cuando termino de enviarse la parte baja, para luego esperar el tiempo de ejecución del dato o comando enviado respectivamente, lo que culmina en el envío de 1 byte de datos. El tiempo de ejecución es de 40uS.

1.5. Led Testigo

EL diagrama de flujo de esta tarea se encuentra en la sección 2.5.

Se determina en los tres bits menos significativos de la parte alta del puerto P, cual esta encendido, y se apaga. se enciende el siguiente de manera incremental (primero 4, luego 5, ultimo 6, y luego 4, ...) donde rotan los tres colores del led tricolor cada vez que el cronometro de un segundo asignado a esta tarea sea cero.

1.6. Tarea Conversión

El diagrama de flujo de esta tarea se encuentra en la sección 2.6.

La tarea Conversión, toma los valores que se asignen a las variables **BIN1** y **BIN2**; y los traduce a los códigos de la tabla de 7 segmentos para la pantalla multiplexada. Este utiliza dos subrutinas, las cuales calculan el offset de la tabla, y lo asignan en valores BCD a otras dos variables utilizando el algoritmo XS3, pasarlas de BCD a las variables de las pantallas de 7 segmentos respectivamente.

1.7. Sub Rutina BIN BCD MUXP

El diagrama de flujo de esta tarea se encuentra en la sección 2.7.

Se implementa el algoritmo XS3, optimizando el inicio de este al ignorar las comprobaciones para los primeros dos desplazamientos. Se utiliza el stack como variable temporal para el operando que se encuentra en progreso de ser desplazado.

1.8. Sub Rutina BCD 7 Segmentos

El diagrama de flujo de esta tarea se encuentra en la sección 2.8.

Se cargan en ambos índices la dirección de la tabla de datos de 7 segmentos, y la dirección de **BIN1**, con el objetivo de reutilizar el mismo código para extraer los datos y guardarlos en las variables utilizadas por la pantalla de 7 segmentos, realizando el cambio en el índice de **BIN1** por **BIN2**.

1.9. Tarea Pantalla MUX

El diagrama de flujo de esta tarea se encuentra en la sección 2.9.

Esta tarea posee dos estados, donde busca realizar cambios únicamente cuando el tiempo determinado para cada pantalla de 7 segmentos haya finalizado. Cuando esto sucede, determina por una variable la siguiente pantalla que debe encenderse, y pone en el puerto B el dato que esta desea mostrar. Utiliza la parte baja del puerto P como

habilitador para cada una de las pantallas, y el bit 1 del puerto J como habilitador de los 7 leds (Ya que el puerto B esta conectado a todos).

Según la variable brillo, se determina que porcentaje del tiempo reservado para cada uno de los elementos se utilizara como tiempo de encendido, lo que resulta en una modificación del brillo de los datos presentados en pantalla y en los Leds. Este trabajo se realiza por el estado 2 de la maquina.

1.10. Tarea TCM

El diagrama de flujo de esta tarea se encuentra en la sección 2.10.

Esta tarea se encarga de asignar los valores de los timers de minutos y segundos a las variables BIN2 y BIN1 respectivamente, cada vez que se entre a cualquier estado de esta maquina. EL timer de minutos se descuenta de manera manual, y el timer de segundos se asigna con un offset de 1, el cual inicia contando en el valor establecido por el usuario, y luego se asigna con un valor de 60. Se carga el valor de este timer en el acumulador A, y se le resta 1 antes de enviarlo a la variable binaria respectiva, lo que resulta en la aparición del valor cero en la pantalla de 7 segmentos. Esto porque no pasa suficiente tiempo en cero para ser apreciado por el usuario.

Esta tarea se encarga de administrar el mensaje de bienvenida y el mensaje transitorio, asignando a los punteros para los mensajes respectivos y borrando la bandera que determina la ejecución de la **Tarea_LCD** en el despachador de tareas.

Con un push botton en PH0, se inicia el descuento de los timers y el envío del mensaje transitorio. Cuando los times lleguen a cero, se devuelve al mensaje de bienvenida y se restablecen los timers en el tiempo original. Estas acciones suceden en los estados 1 y 2 respectivamente.

1.11. Rutina de Atención a Interrupciones

El diagrama de flujo de esta tarea se encuentra en la sección 2.11.

Como se tuvo que agregar código a esta rutina de atención, se presenta el diagrama de flujo completo. Se establecieron (Como se puede ver en la sección 1.12) una tabla de base T que maneja timers sensibles al error y un timer de base 20us. Una tabla de base 20us que maneja el timer de 1mS, que genera la base de 1mS y así sucesivamente.

El restablecimiento de la interrupción de output compare se realiza al inicio de esta rutina, lo que minimiza los errores de tiempo.

1.12. Tabla de Timers

Org \$1500

Tabla_Timers_BaseT:

```
Timer20uS      ds 1      ;Timer 20uS con base tiempo de interrupcion
Timer40uS      ds 1
Timer260uS     ds 1
Fin_BaseT      db $FF
```

Tabla_Timers_Base20uS:

```
Timer1mS       ds 1      ;Timer 1mS para generar la base tiempo 1mS
Counter_Ticks  ds 1
```

Fin_Base20uS: db \$FF

Tabla_Timers_Base1mS:

```
Timer10mS      ds 1      ;Timer para generar la base de tiempo 10 mS
```

```

Timer2mS      ds 1
TimerDigito   ds 1      ;Timer de digito de pantalla MUX
Timer_RebPB   ds 1      ;Timer supresion rebotes Leer PB
Timer_RebTCL  ds 1      ;Timer de supresion de rebotes teclado

Fin_Base1mS   dB $FF

Tabla_Timers_Base10mS:

Timer100mS    ds 1      ;Timer para generar la base de tiempo de 100 mS
Timer_SHP     ds 1      ;Timer para short press

Fin_Base10ms  dB $FF

Tabla_Timers_Base100mS:

Timer1S       ds 1      ;Timer para generar la base de tiempo de 1 Seg.

Fin_Base100mS dB $FF

Tabla_Timers_Base1S:

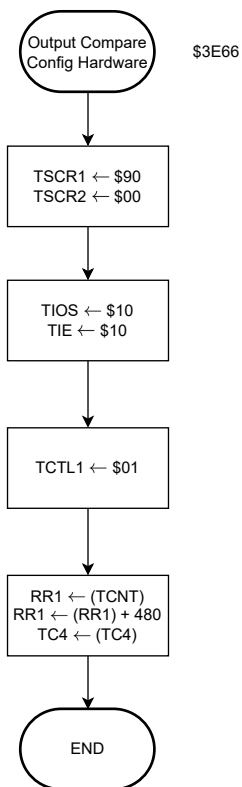
Timer_LED_Testigo ds 1 ;Timer para parpadeo de led testigo
Timer_LP        ds 1 ;Timer para long press
SegundosTCM     ds 1
Fin_Base1S      dB $FF

```

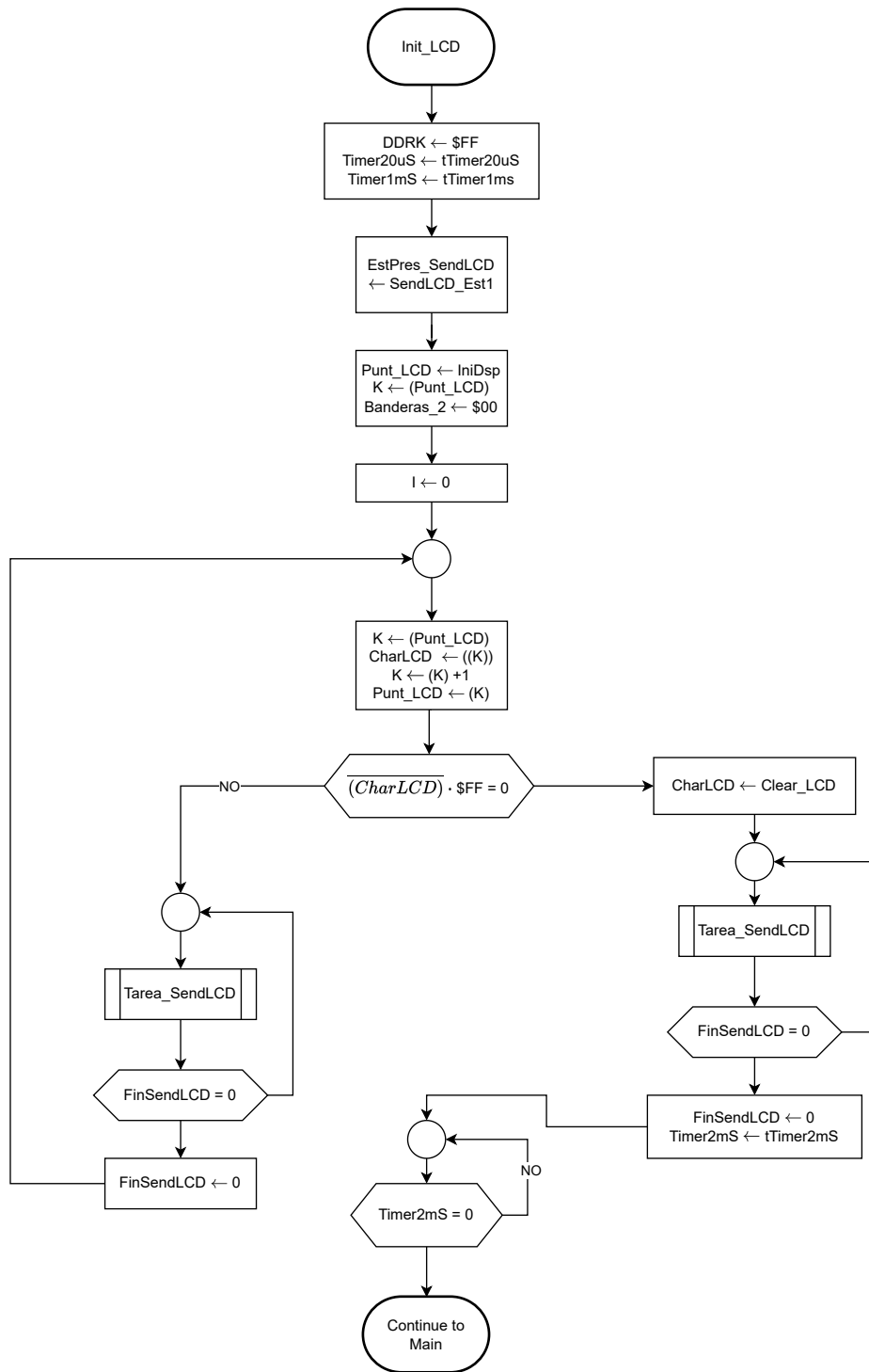
2. Anexos

2.1. Configuración de Hardware

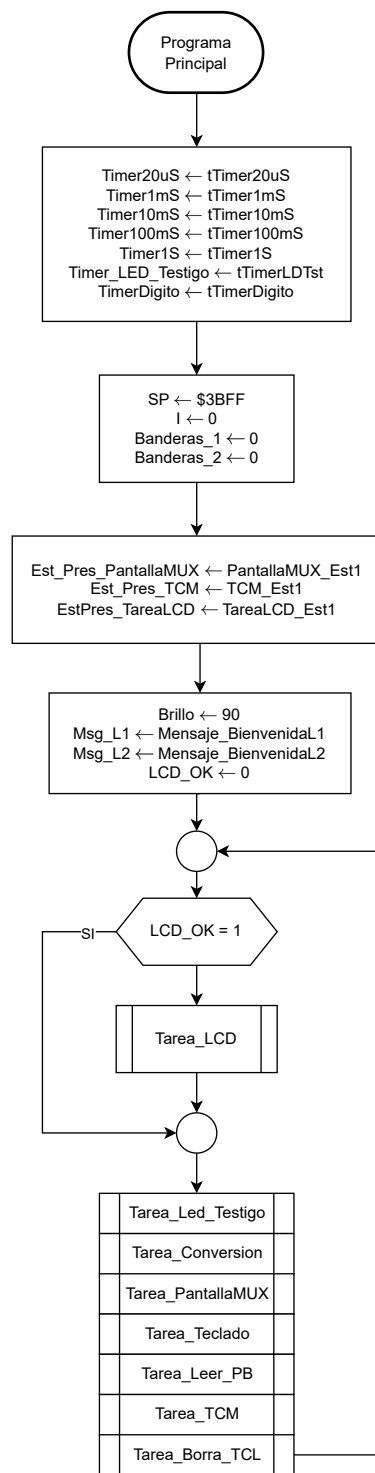
2.1.1. Output Compare



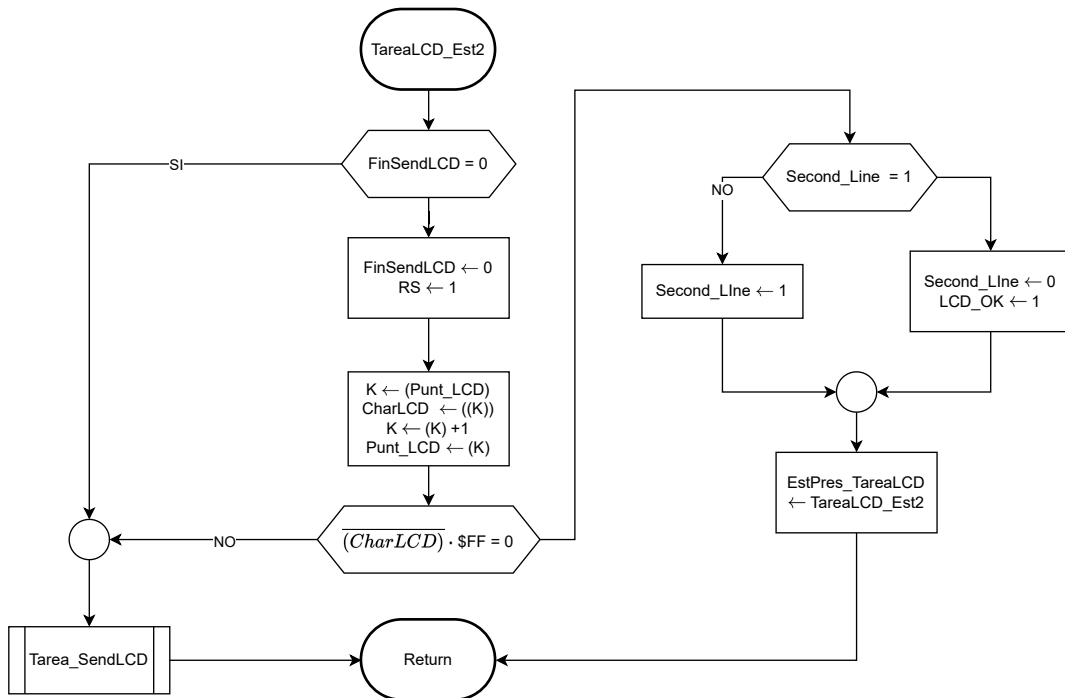
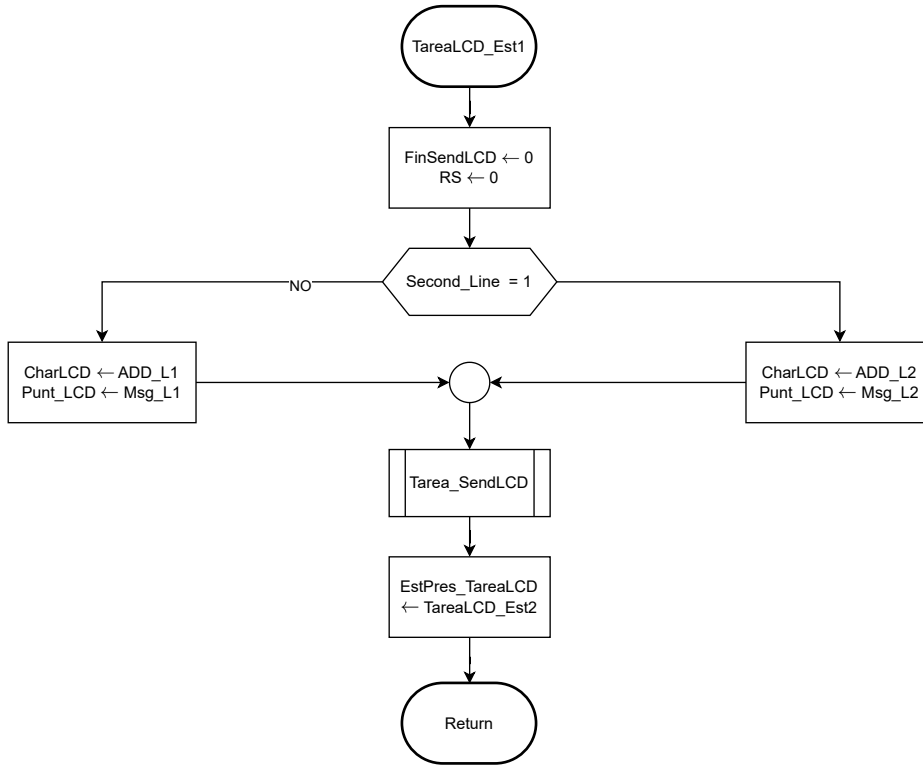
2.1.2. Pantalla LCD



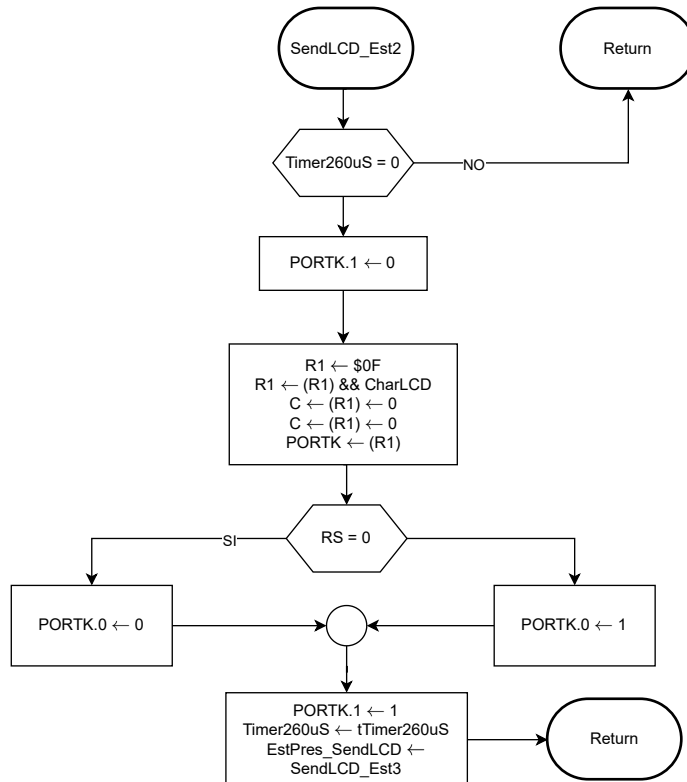
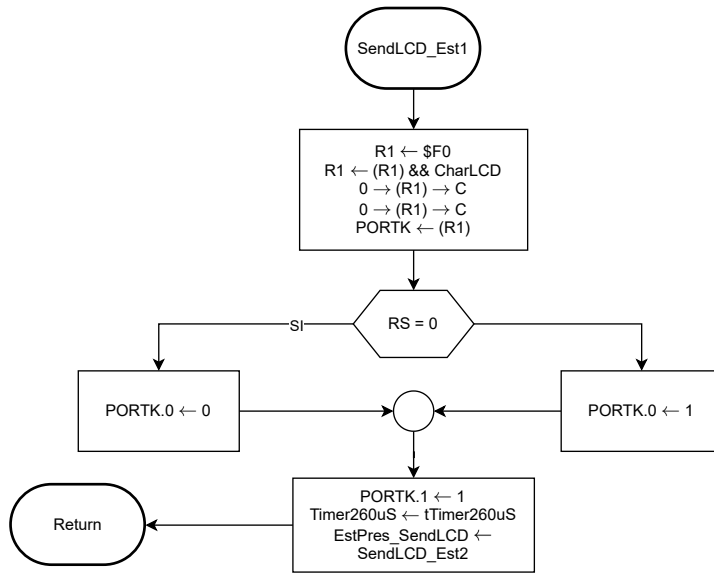
2.2. Programa Principal

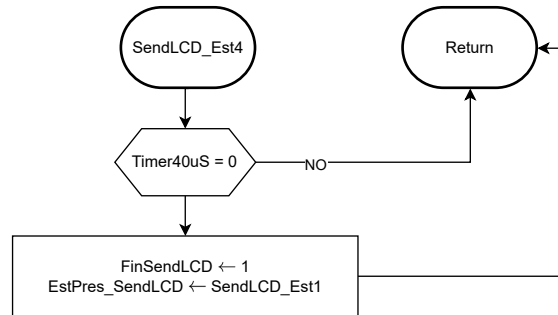
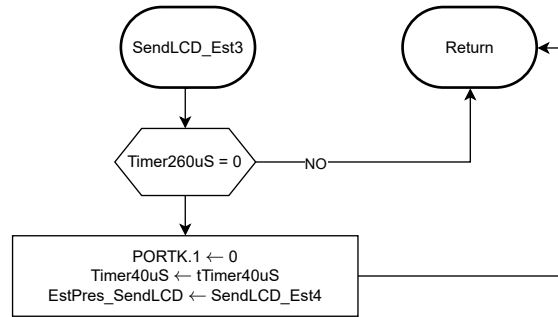


2.3. Tarea LCD

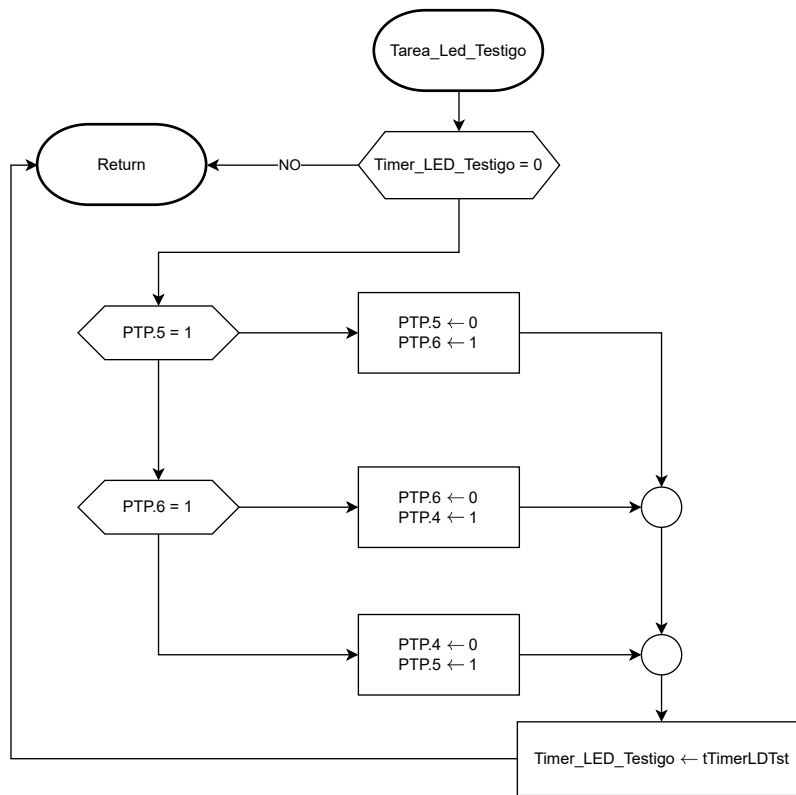


2.4. Tarea Send LCD

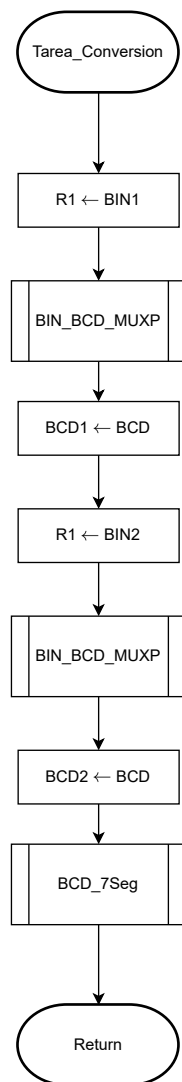




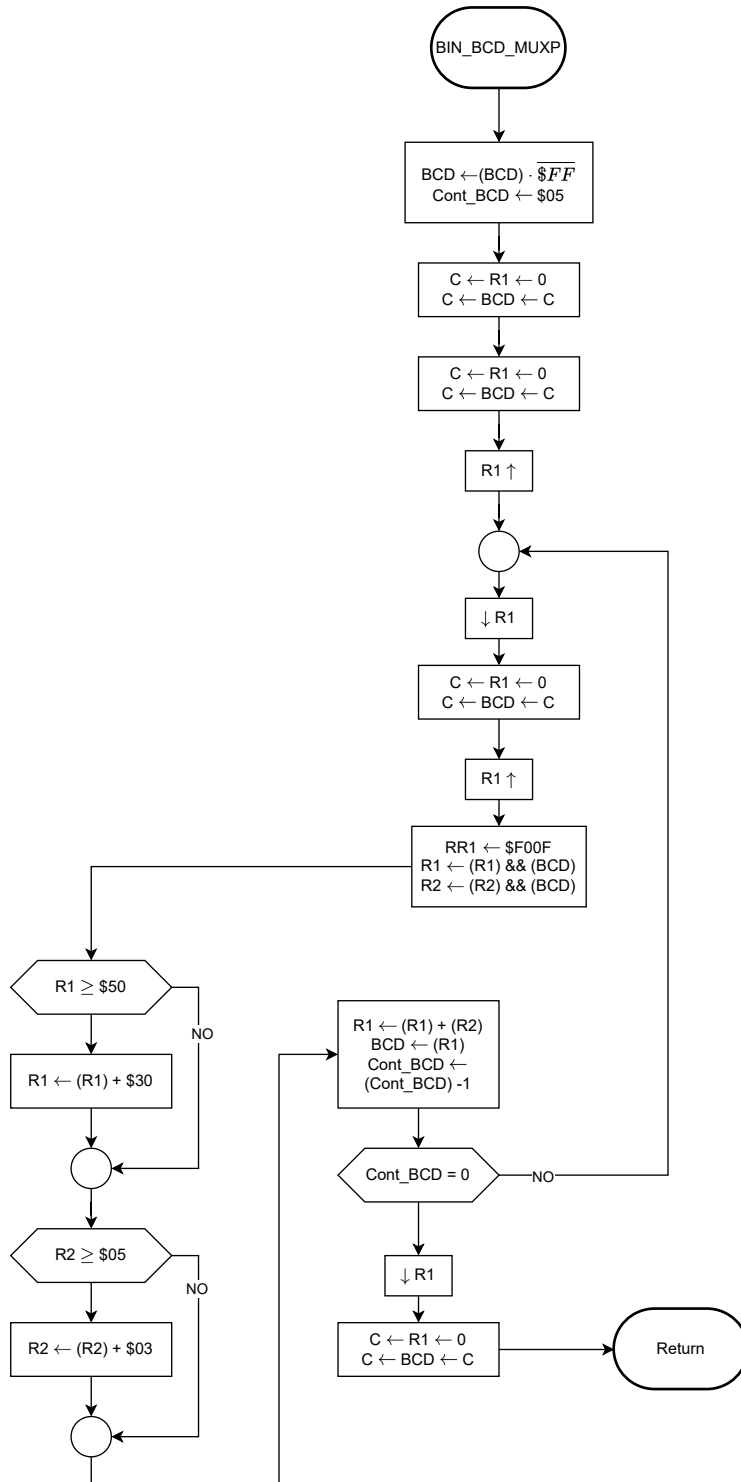
2.5. Led Testigo



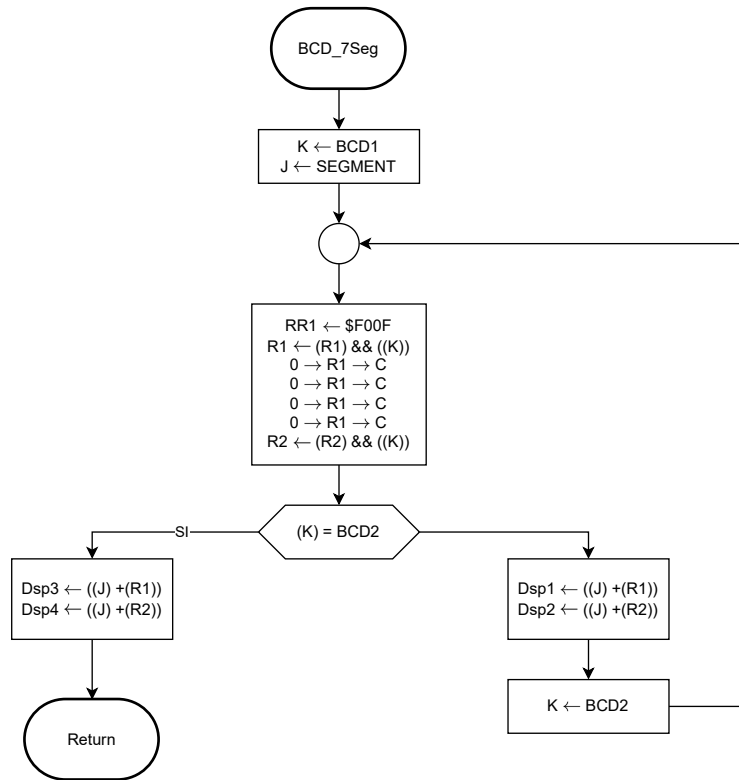
2.6. Tarea Conversión



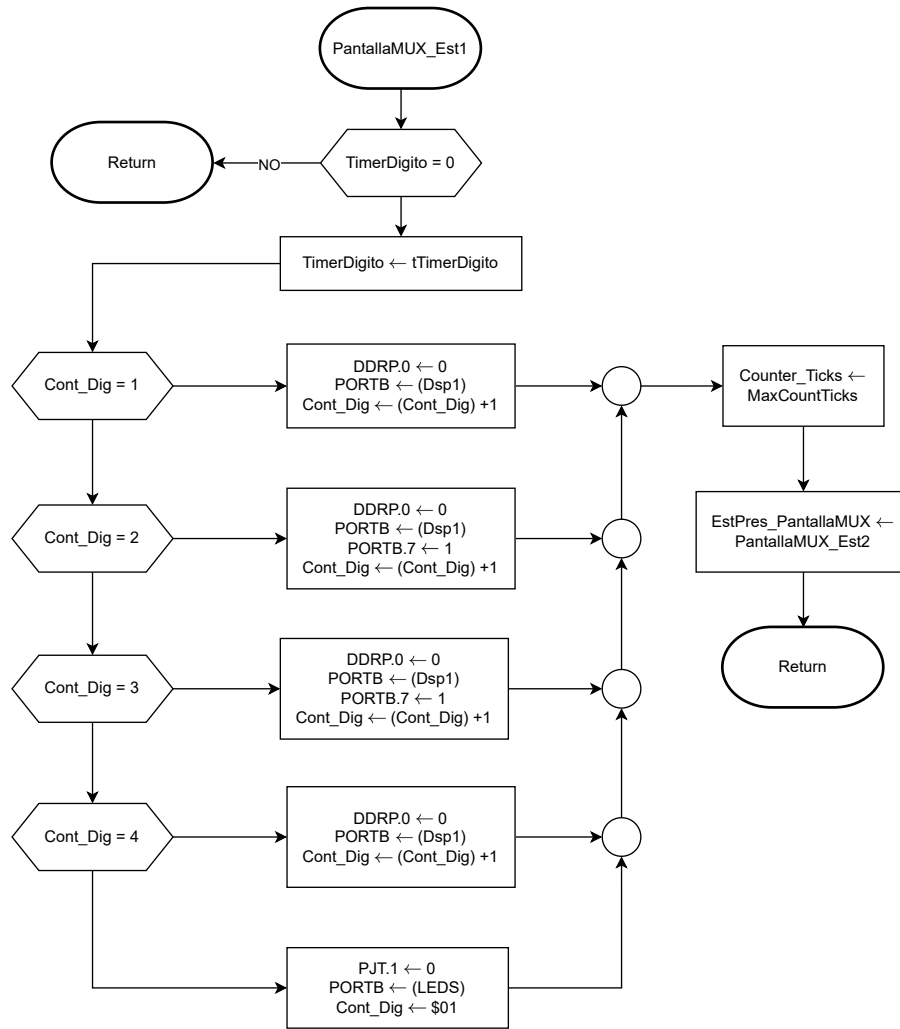
2.7. Sub Rutina BIN BCD MUXP

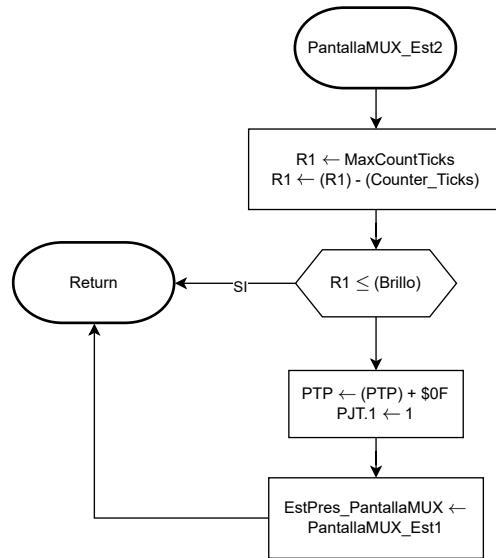


2.8. Sub Rutina BCD 7 Segmentos

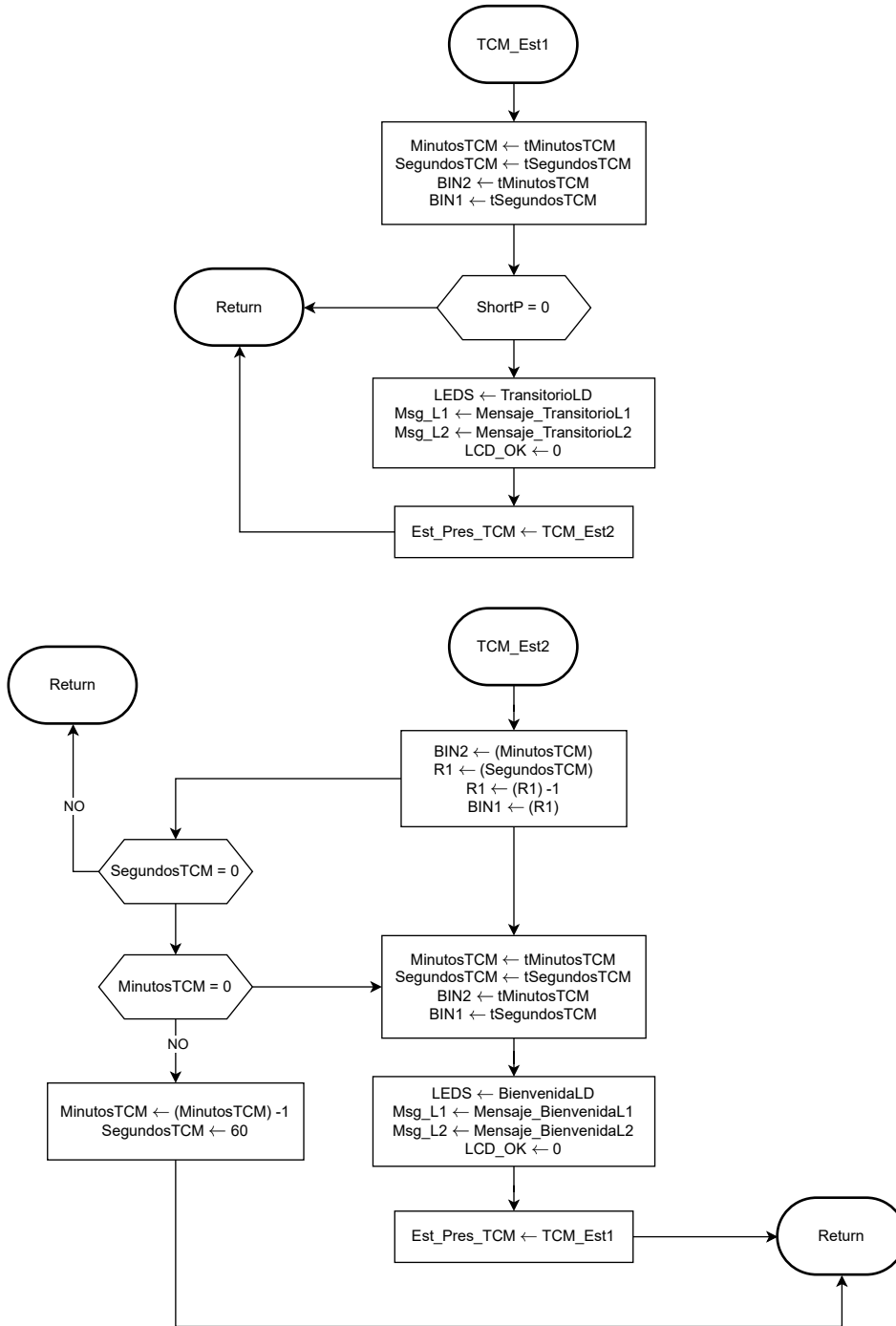


2.9. Tarea Pantalla MUX





2.10. Tarea TCM



2.11. Rutina de Atención a Interrupciones

