

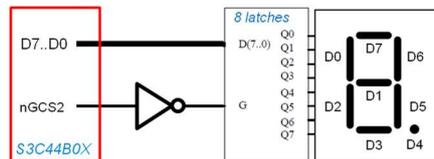


EXAMEN DE PROGRAMACIÓN DE SISTEMAS Y DISPOSITIVOS

CURSO 2024-25, SEGUNDA CONVOCATORIA, 19 DE JUNIO DE 2025

En la realización de los ejercicios se considerarán implementadas y, por tanto, podrán usarse todas las funciones y macros públicas incluidas en el BSP desarrollado en los laboratorios. No podrá usarse ninguna función perteneciente a bibliotecas estándar de C. Si fuera necesaria alguna función auxiliar adicional, deberá ser codificada explícitamente. Se usará únicamente aritmética entera.

1. **(0.25 puntos)** Dado que en la placa de prototipado está desocupado el banco 2 de memoria del microcontrolador S3C44B0X, se conecta un nuevo *display* 7 segmentos al bus de datos del SoC como se muestra en la figura. Para que un programador pueda visualizar números en él, defina una nueva macro SEGS2 como nemotécnico de la dirección de memoria en la que se encuentra mapeado el nuevo *display*.



2. **(0.25 puntos)** Se desea añadir un nuevo led a la placa de prototipado similar a los ya existentes. Indique a qué pin y de qué puerto lo conectaría, así como los cambios necesarios a realizar en la función `port_init()` para que esto sea posible.
3. **(0.25 puntos)** Indique los cambios necesarios a realizar en la función `uart0_init()`, para que la velocidad transmisión del interfaz sea 2400 baudios y los datos se transmitan con paridad impar (*odd parity*). Asuma que la frecuencia del reloj del sistema es 64 MHz.
4. **(0.25 puntos)** Para evitar que el *RTC* dispare interrupciones periódicas existen 2 alternativas: deshabilitar en el controlador de interrupciones las interrupciones de este dispositivo o deshabilitar en el propio dispositivo la generación de *ticks*. Indique en cada caso la sentencia/sentencias C necesarias para hacerlo.
5. **(0.25 puntos)** Indique el retardo en ms que hace la llamada a esta función. Asuma que la frecuencia del reloj del sistema es 64 MHz.

```
void timer3_unknown_delay( void )
{
    TCFG0 = (TCFG0 & ~(0xff << 8)) | (127 << 8);
    TCFG1 = (TCFG1 & ~(0xf << 12)) | (2 << 12);
    TCNTB3 = 18750;
    TCON = (TCON & ~(0xf << 16)) | (1 << 17);
    TCON = (TCON & ~(0xf << 16)) | (1 << 16);
    while( !TCNTO3 );
    while( TCNTO3 );
}
```

6. **(0.25 puntos)** Indique el número real que representa de la siguiente constante suponiendo codificada en punto fijo Q24.8

```
const int32 foo = 174752;
```

7. (0.25 puntos) Se desea que una tarea periódica cuyo prototipo es `void Task(void)` se ejecute cada 50 ms bajo un kernel de planificación no expropiativo como el utilizado en los laboratorios. El *tick* de reloj deberá generarse a una frecuencia de 100 Hz. Complete los puntos suspensivos del siguiente código para que esto sea posible.

```
create_task( ..., ... );
timer0_open_tick( scheduler, ... );
```

8. (0.25 puntos) En una aplicación bajo $\mu\text{C}/\text{OS-II}$ una tarea, `Task1`, desea enviar de uno en uno una serie de números pares consecutivos a otra tarea, `Task2`. Para ello, debe usar un buzón de mensajes. Se pide:

- Indicar la sentencia/sentencias C necesarias para declarar el buzón.
- Indicar la sentencia/sentencias C necesarias para crear el buzón.
- Complete los puntos suspensivos del siguiente código.

```
void Task1( void *id )
{
    int32 num;

    num = 0;
    while( 1 ) {
        ...
        num = (num + 1) % 100;
        OSTimeDly( 5 );
    }
}

void Task2( void *id )
{
    ...
    OSTimeDly( 5 );
}
```

9. (2 puntos) Basándose en la función `iis_playWawFile` del BSP, codifique en C una función que permita reproducir, a través del controlador de bus IIS del microcontrolador S3C44B0X, un fragmento de un fichero *wav* almacenado en memoria en el formato habitual: alternando las muestras del canal izquierdo y del derecho a razón de 16 bits/muestra y 16000 muestras/segundo. El instante de inicio y final del fragmento a reproducir se indican relativas al comienzo del sonido en centésimas de segundo. Se valorará que la función haga los correspondientes chequeos de consistencia de los argumentos (que el instante de inicio sea anterior al final y que ambos instantes sean anteriores a la duración total del sonido). El prototipo de la función será:

```
void iis_playCutWawFile( int16 *wav, uint32 from, uint32 to );
```