

Evaluación del Laboratorio Número 2

Sistemas con Microprocesadores y Microcontroladores

29/09/2021

Instrucciones

- La presente evaluación es individual y podrá utilizar todo el material escrito del que disponga.
- La misma consta de 1 ejercicio en total y la duración será de 120 minutos.
- En cada ejercicio se indica el tiempo estimado para resolverlo y el puntaje que otorga el mismo.
- En ningún caso el puntaje máximo podrá superar los 100 puntos.
- La evaluación se desarrollará en el entorno de Google Classroom.
- La evaluación deberá ser desarrollada en la maquina virtual otorgada por la cátedra y deberán enviar el/ los archivos como adjuntos a la tarea de Google Classroom.
- Se deberá completar los datos (nombre, apellido y DNI) del encabezado en cada código fuente.
- No se admitirán otros archivos o formatos a menos que sean explícitamente solicitados.
- No se permitirá intercambiar ni prestar ningún tipo de elementos o información entre los estudiantes.
- La legibilidad del código, la cantidad de comentarios y la buena presentación serán tenidas en cuenta para la calificación final.

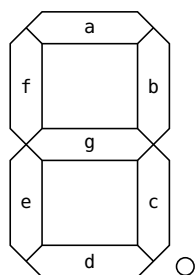
Ejercicios para resolver

1) 120 Minutos - 110 Puntos (20, 35, 30, 25)

Se quiere mostrar el día y el mes actual en los displays 7 segmentos del poncho de la placa EDU-CIAA. El día está guardado en la dirección de memoria "día", mientras que el mes está guardado en la dirección de memoria "mes". Ambos datos ocupan un byte y están almacenados en binario natural.

Tenga presente que la solución a este problema debe funcionar para cualquier fecha del año, no sólo para el caso particular planteado en los ejemplos. Para desarrollar la solución usted deberá completar los siguientes pasos:

- a) Transforme en una subrutina transparente **segmentos** el código desarrollado para el laboratorio 1 que convierte un número en su representación de siete segmentos. La nueva subrutina deberá recibir en **R0** el valor BCD a convertir y devuelve en el mismo registro su correspondiente representación en siete segmentos de acuerdo a la conexión que se muestra en la siguiente figura.



b7	b6	b5	b4	b3	b2	b1	b0
	g	f	e	d	c	b	a

- b) Escriba una subrutina transparente **conversion** que recibe en **R0** la dirección de memoria donde está almacenado un número binario de un byte menor o igual que 99. La subrutina guarda, como bytes, el valor de las decenas en la dirección de memoria almacenada en **R1** y el valor de las unidades en la dirección siguiente.

Dato	Resultado
(numero) = 29	(destino) = 2
	(destino + 1) = 9

- c) Escriba una subrutina transparente **fecha** que reciba en el registro **R0** la dirección de memoria donde se encuentra almacenada la fecha actual. Esta fecha se almacena como el día en un byte y el mes en el byte siguiente. Esta subrutina debe almacenar la correspondiente representación en siete segmentos de los cuatro dígitos a partir de la dirección de memoria apuntada por **R1**, según el siguiente ejemplo. Esta subrutina debe **obligatoriamente** utilizar las dos subrutinas previamente desarrolladas.

Fecha	Resultado
(dia) = 29	(destino) = 0x5B (Segmentos correspondientes al 2)
(mes) = 9	(destino + 1) = 0x67 (Segmentos correspondientes al 9)
	(destino + 2) = 0x3F (Segmentos correspondientes al 0)
	(destino + 3) = 0x67 (Segmentos correspondientes al 9)

Escriba también un programa principal que pruebe la subrutina desarrollada.

- d) Se proporciona la subrutina transparente **mostrar** que permite mostrar un dígito en uno de los displays siete segmentos del poncho correspondiente a la placa EDU-CIAA. Para ello, antes de llamar a la subrutina, debe almacenar en **R0** el mapa de bits de siete segmentos correspondiente al número que desea mostrar, y en **R1** debe almacenar el valor 0x08, 0x04, 0x02 o 0x01 según el display que desea utilizar. Modifique el programa principal del apartado anterior para mostrar en los cuatro displays de siete segmentos el día y fecha actual, por ejemplo "2 9 0 9". Para poder lograr esto, deberá encender un display por vez, esperando el tiempo necesario antes de apagarlo y encender el siguiente. Para esto se proporciona también una subrutina **demora** que no requiere parámetros.