

- Nombre y CI en cada hoja
- Numere las hojas, indique el total en la primera
- Utilice sólo un lado de las hojas

- Incluya un solo problema por hoja
- **Sea prolijo**
- **Aprobación:** mínimo UN problema

PROBLEMA 1

Se debe diseñar un sistema basado en Z80 para controlar la posición de un disco giratorio. Dos sensores en posición fija permiten determinar la posición del disco. Uno detecta el pasaje de dientes y ranuras presentes en el borde del disco, otro detecta cuándo se encuentra frente a la posición origen un orificio realizado en el disco a esos efectos. El sistema tiene las siguientes entradas y salidas:

- **girar:** (in) pulsador para iniciar y detener el giro como se indica más adelante (1 = activo).
- **ranura:** (in) sensor que detecta el pasaje de las ranuras (1 = ranura, 0 = diente).
- **origen:** (in) sensor de posición origen (1 = el orificio está frente a la posición origen).
- **motor:** (out) encendido del motor (1 = encendido).
- **pos:** (out) 8 bits, cantidad de ranuras que han pasado desde la posición de origen.
- **tic:** (in) señal periódica que se usará para interrumpir al procesador.

El efecto producido por **girar** depende de por cuánto tiempo se mantiene presionado el pulsador. Para medir la duración de los pulsos se utiliza la interrupción periódica **tic**. El pulsador se lee en la interrupción y debe funcionar como sigue:

- Si se presiona el pulsador (**girar** = 1) el motor debe comenzar a girar.
- Si el pulsador se suelta antes de la quinta interrupción, entonces se detiene el giro.
- Si el botón se mantiene presionado hasta la quinta interrupción, entonces el disco debe seguir girando hasta que se detecte un flanco de subida en la señal girar.

Todas las acciones de encender y apagar el motor se toman en la interrupción siguiente a que sucede el cambio en las entradas. El sistema debe ser capaz de detectar un flanco de subida en **girar** incluso cuando el botón se suelta y se vuelve a presionar entre dos interrupciones consecutivas.

De esta forma para girar el disco durante un tiempo prolongado no es necesario mantener presionado el pulsador durante todo el tiempo.

Cuando el sistema se enciende se debe invocar a la subrutina **buscar_origen** que avanza el disco hasta que se active el sensor **origen**, apaga el motor y retorna. A partir de ahí debe responder a **girar** como se describió más arriba.

Además, el sistema debe llevar la cuenta de la posición. Durante la inicialización, mientras la posición es desconocida la salida **pos** debe valer 0xFF. Cuando se retorna de **buscar_origen** debe hacerse 0. Luego debe incrementarse con cada flanco de subida en **ranura**, volviendo a 0 cada vez que se detecta la posición origen.

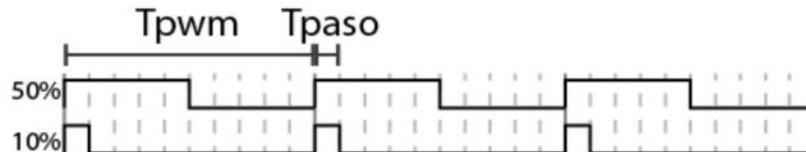
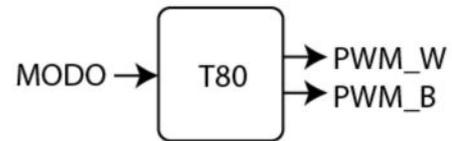
El software debe organizarse en base a la interrupción periódica que lee el pulsador y maneja el motor, y de un programa principal que inicializa todo, busca la posición origen y luego queda permanentemente monitoreando la posición y actualizando la salida **pos**.

Se pide:

- a) Todo el hardware.
- b) Rutina de atención a la interrupción.
- c) Inicialización y lazo de programa principal. Incluir las directivas necesarias para indicar la ubicación en memoria de todo el código y de todas las variables que se utilicen. La subrutina **buscar_origen** se supondrá implementada.

PROBLEMA 2

Se desea hacer un control para luces led Blancas y Azules mediante un sistema basado en un T80 corriendo a 50MHz. La intensidad o brillo de los leds se controla mediante una señal PWM de 1280us de período ($T_{pwm} = 10 \times T_{paso}$). Si se desea que uno de los leds este al 50%, la señal deberá estar en 1 durante la mitad del período (5 T_{paso}) y en 0 durante el resto del tiempo (5 T_{paso}). Si se desea que la intensidad sea 10%, se deberá poner en 1 la salida durante un 10% del período T_{pwm} y en 0 el resto del T_{pwm} .



El sistema debe de permitir elegir entre varios modos de funcionamiento. Cuando se presiona el botón de selección (se produce un flanco en la entrada Modo), el sistema deberá de pasar al modo siguiente especificado en la tabla.

Modo	Leds Blancos	Leds Azules
Apagado	0%	0%
Cálido	60%	0%
Neutro	60%	20%
Frío	50%	40%

Para la base de tiempos, debe de utilizarse un timer programado para que genere una interrupción con un período $T_{paso} = T_{pwm}/10$.

El sistema deberá de funcionar en modo 2 de interrupciones. Durante la inicialización, se debe de invocar la subrutina INI_OTRAS_INT, la cual inicializa otros dos controladores existentes en el sistema. Esta rutina utiliza el primer y segundo lugar de la tabla y espera encontrarla a partir de la dirección 0x9000.

El software debe de organizarse en:

- una tabla en ROM con los modos y valores de los ciclos de trabajo de las señales PWM;
- un programa principal que inicializa todos los puertos y periféricos, inicializa todo lo necesario para trabajar en modo 2 de interrupciones y atiende los pedidos de cambio de modo. Cuando hay un cambio de modo se recomienza la generación de la señal PWM descartando cualquier periodo que estuviera incompleto.
- una rutina de atención a interrupciones que maneja las salidas PWM_i de acuerdo al modo seleccionado y momento en que esta dentro del período T_{pwm} .

Se pide:

a) hardware completo del sistema (puertos, periféricos). Se supone que se cuenta con 32K de ROM y 32K de RAM.

b) reservas de memoria (variables, tabla de modos, tabla de interrupciones) y programa principal. Luego de un reset el sistema arranca en modo Calido

c) rutina de atención de interrupciones.

NOTA: $T_{paso} = 128 \text{ us} = 1/50\text{E}6 \cdot 2^5 \cdot 200$