

- 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 desea diseñar con un microprocesador T80 un sistema que transmita en forma inalámbrica los datos de un sensor de la siguiente manera: mientras el dato no varíe es retransmitido de forma periódica; cuando el dato se modifica se reduce considerablemente el intervalo entre transmisiones y en cada retransmisión se duplica hasta volver a régimen como se ejemplifica en la figura.

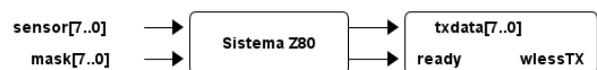


Para esto se cuenta con un bloque timer y su correspondiente controlador de interrupciones para interrumpir al procesador trabajando en modo 2. En régimen, los datos se deben retransmitir con período **128T**. Un dato nuevo debe transmitirse inmediatamente después de detectado el cambio, y se retransmite **1T** después, luego **2T**, **4T** y así sucesivamente hasta llegar a **128T**, a partir de ahí no se incrementa más el período de retransmisión. Para evitar una avalancha de transmisiones en caso que el sensor cambie continuamente, cada vez que se detecta un cambio y se transmite un nuevo valor debe pasar un tiempo **1T** antes de una nueva transmisión.

Durante la inicialización se debe leer el valor de una máscara desde la entrada **mask[7..0]** (cambios en **mask[]** posteriores a la inicialización deben ignorarse). Para detectar cambios en el dato leído, solamente se deben considerar los bits de **sensor[]** en los que dicha máscara vale 1, sin embargo cuando se detecta un cambio se debe transmitir el dato entero.

El programa principal debe ser un loop infinito en que se lee el sensor para detectar cambios respecto a la lectura anterior, y se atiende a otras tareas invocando la subrutina **atiendo_otros()** (que se supondrá implementada). En caso de detectar un cambio el programa principal debe realizar la primera transmisión del nuevo valor y reprogramar el timer para arrancar la secuencia de interrupciones a intervalos crecientes. En las sucesivas interrupciones se debe reprogramar el timer duplicando el intervalo hasta llegar al máximo de **128T**. Para tener en cuenta el mínimo de **1T** antes de una nueva transmisión, no se volverá a leer la entrada hasta que se produzca la primera retransmisión por la rutina de interrupción.

La transmisión inalámbrica se resuelve con la interfaz **wlessTX**, que trasmite el dato disponible en su entrada **txdata[]** cada vez que recibe un pulso en **ready**.



En la inicialización el sistema debe comenzar transmitiendo el primer dato leído como si fuera un dato nuevo.

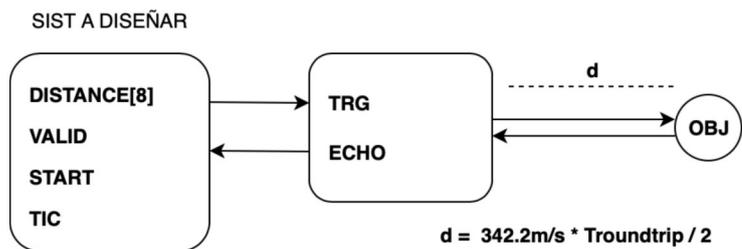
Datos: $f_{clk} = 4\text{MHz}$, $T = 1\text{ms} = 2^5 \cdot 125 \cdot 1/f_{clk}$

Se pide:

- Puertos y periféricos necesarios para las funcionalidades solicitadas. **Ya está implementada** la memoria con **32KB 32KB de ROM y 32KB de RAM** y están **ocupados por otros dispositivos** el intervalo **0x00-0xBF** del espacio de E/S y **los dos primeros lugares en la tabla** de interrupciones. Se debe configurar la interrupción de timer como la de mayor prioridad, para no retrasar las retransmisiones de datos debido a otras interrupciones.
- Inicialización del sistema y directivas de reserva de memoria. Las tareas y dispositivos no descriptos deben inicializarse invocando la subrutina **init_otros()** que se supone implementada.
- Programa principal.
- Rutina de atención a la Interrupción del timer.

PROBLEMA 2

En el marco de un sistema de asistencia de estacionamiento para un automóvil se necesita diseñar un módulo basado en Z80 que permita utilizar un sensor externo de distancia por ultrasonido. El principio de funcionamiento del sensor se basa en medir el tiempo transcurrido entre la emisión de un tren de pulsos de ultrasonido y la posterior detección del eco que se produce cuando estos pulsos rebotan en el obstáculo próximo al sensor. La distancia al obstáculo se puede calcular de forma simple a partir de la velocidad del sonido y el tiempo medido entre emisión y eco.



El sensor disponible funciona de la siguiente forma. Al detectar un flanco de subida en su entrada **TRG**, el sensor comienza a emitir el tren de pulsos hacia afuera y simultáneamente lleva su salida **ECHO** a 0. En el momento en que el sensor detecte el eco provocado por el objeto próximo, la salida **ECHO** sube a 1 inmediatamente y permanece así hasta el próximo flanco en **TRG**.

El sistema a diseñar será responsable de comandar el sensor y medir el tiempo hasta la llegada del eco. En un principio, el sistema está en estado de reposo a la espera de un flanco de subida en la entrada **START**. Una vez detectado el flanco de **START**, activará el sensor con un flanco en **TRG** y comenzará a medir el tiempo ($T_{\text{ROUNDTRIP}}$) hasta la llegada de **ECHO**. Habiendo detectado la activación de **ECHO**, se calculará la distancia (en centímetros) hasta el objeto a partir del tiempo $T_{\text{ROUNDTRIP}}$ medido para luego reportar su valor en la salida **DISTANCE[8]**. La salida **VALID** debe estar en 1 para señalar que los datos en **DISTANCE[8]** son válidos, por lo que debe ser 0 mientras se realiza una medida. Notar que en ausencia de un objeto próximo el sensor nunca recibirá un eco, y habrá que manejar este caso detectando el transcurso un tiempo T_{TIMEOUT} sin eco. En el caso en que no se detecte el eco antes de T_{TIMEOUT} , el sistema debe reportar la constante especial **0xFF** en **DISTANCE[8]**.

Para medir el tiempo, se utilizará una interrupción **modo 1** asociada a la señal periódica **TIC** externa. **TIC** tiene un período $T_{\text{TIC}} = 116 \text{ us} = 4 \text{ cm} / (343.2 \text{ m/s})$. Notar que en 116 us el sonido habrá recorrido una distancia de 4cm, por lo que es admisible que el pulso de **START** pueda darse en cualquier momento con respecto a **TIC** (no necesita estar sincronizado) ya que el error será despreciable para la aplicación.

Notas:

- El rango de distancias válido para el sensor es tal que d vale entre 0 y 4m.
- $T_{\text{TIMEOUT}} = 250 * 116\text{us} = 29\text{ms}$

Se pide:

- a) hardware completo del sistema (memorias, puertos, decodificación)
- b) rutina de atención a interrupción **TIC** responsable de medir tiempo con resolución T_{TIC} y variables para comunicación con programa principal.
- c) programa principal encargado de esperar **START**, manejar el sensor, avisar mediante variables a la rutina de atención a interrupción que se está midiendo y reportar medición de distancia.
- d) inicialización completa del sistema (directivas de memoria, interrupción, variables, etc).