- Nombre y CI en cada hoja - Numere las hojas - Indique el total de hojas en la primera - Utilice solo un lado de las hojas	- Incluya un solo problema por hoja - **Sea prolijo** - **Aprobación:** mínimo UN problema

PROBLEMA 1

Se desea implementar un dispositivo que reciba datos de un sensor en la rueda de un automóvil. El dispositivo funciona además como velocímetro digital y está basado en un Z80.
En cada vuelta de rueda el sensor adquiere un dato y lo pone a disposición del dispositivo a diseñar. Este debe leer el dato y además contar
la cantidad de pulsos generados al girar las ruedas durante 1 segundo, convertirlo a Km/h y mostrar la velocidad en la salida **DISPLAY[8]**.

Cada vez que queda un dato disponible el sensor genera un pulso corto en la señal **P** y pone el valor sensado en la señal **sensor_in[8]**. El pulso en P deberá usarse para generar una interrupción en la que se lea el valor de **sensor_in** y se lleve la cuenta de las vueltas de rueda. Además cada valor leído debe ser promediado con el anterior y escrito en la salida **sensor_out[8]**.
La velocidad deberá ser calculada una vez por segundo y cada vez que es calculada deberá actualizarse su valor en **DISPLAY**.

Para generar las interrupciones se dispone de un CTC. Se utilizará un canal como contador con constante de recarga = 1 para generar una interrupción con cada flanco creciente de **P** y los canales que sea necesario para generar una interrupción periódica cada un segundo.

Para la conversión de No. de pulsos a Km/h se invocará a la subrutina **pulsosakph**, que se supone implementada, que recibe en el acumulador el Nº de pulsos a convertir y devuelve la velocidad en una variable de memoria **varkph**. Esta subrutina preserva todos los registros que utiliza. La subrutina **pulsosakph** utiliza el contenido de una variable en memoria **RODADO**, que sirve para seleccionar el diámetro de la rueda usada y que **se deberá inicializar** al prender el sistema. El valor a cargar en **RODADO** deberá ser leído del puerto de entrada **port_RODADO** de 8 bits (que también debe ser implementado) durante la inicialización.

a) Implementar todo el hardware necesario. El Z80 deberá trabajar en **modo 2** de interrupciones. La mitad más baja del espacio de E/S deberá dejarse libre para futuras ampliaciones.

b) Implementar las rutinas de atención a interrupción del CTC que implementan las funciones de lectura y promediado del sensor y de velocímetro.

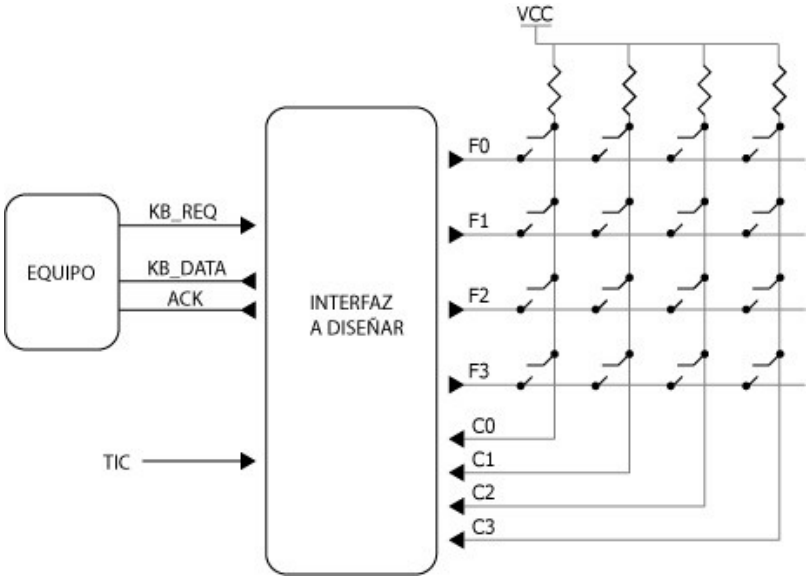
c) Escribir el código de inicialización que realice todo lo necesario a partir de un reset para la funcionalidad del velocímetro. Debe terminar con un salto a la dirección **ppal** donde comienza el programa principal del sistema que realiza otras funciones. Incluir además las directivas de reserva de memoria necesarias para todas las variables utilizadas.

Notas:
fck = 1MHz, 1.000.000 = 16 x 250 x 250

PROBLEMA 2

Un teclado de 4 x 4 teclas está construido con 4 conductores horizontales (líneas de fila) y 4 verticales (líneas de columna). Cada tecla es un interruptor que conecta una línea de columna con una línea de fila. Las líneas de fila son entradas al teclado y las líneas de columna son salidas. Las líneas de columna están conectadas con un pull-up a Vcc, de manera que si ninguna tecla está apretada todas las salidas están en nivel alto.
Si se presiona una tecla (fila i, columna j) el interruptor conecta la fila i a la columna j por lo que en la línea de salida j se verá el nivel presente en la entrada de fila i.
Si se escribe 1110 en las líneas de entrada y se lee las salidas de columna se determina el estado de las teclas de la primera fila (F0 en el dibujo). Repitiendo la operación para las otras filas se puede hace un barrido de todo el teclado.

Se quiere diseñar, utilizando un Z80, una interfaz entre un teclado de 4x4 teclas y un equipo que necesita saber si hay alguna tecla presionada.



Se utiliza una interrupción periódica para interrogar al teclado y comunicar al programa principal el estado del mismo.
En cada interrupción se hace un barrido completo del teclado.

Cuando el equipo necesite saber el estado del teclado, dará un pulso a 1 de corta duración en la señal KB_REQ y esperará recibir un flanco de subida en su entrada ACK. En el momento del flanco el resultado del último barrido deberá ser presentado en la entrada KB_DATA de 8 bits.

Se supondrá que nunca están presionadas dos teclas en forma simultánea.

Se pide:

a) Todo el hardware del sistema. Para generar la interrupción periódica se dispone de una señal TIC con una onda cuadrada.

b) Subrutina scan() que realice un barrido del teclado hasta encontrar la tecla presionada (si la hubiera) y que devuelva en el acumulador un código que identifica la tecla presionada como se explica a continuación. El nibble alto (A7..A4) debe indicar la fila con un 0 en la posición correspondiente. En forma análoga (A3..A0) indica la columna. P. ej. A = 1110 1011 B indica que está presionada la tecla en la fila 0, columna 2. A = 1111 1111 B indica que ninguna tecla está presionada.

c) Rutina de atención a la interrupción que invoque a la subrutina scan(). El resultado del barrido se almacenará en la variable KBD_VAL (código de la tecla presionada).

d) Programa a ejecutar luego de un reset que inicialice el sistema y reporte el código almacenado en KBD_VAL cuando el equipo lo solicite.