

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

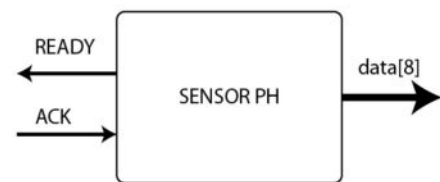
- Incluya un solo problema por hoja
- **Sea prolijo**
- **Aprobación:** mínimo UN problema

## PROBLEMA 1

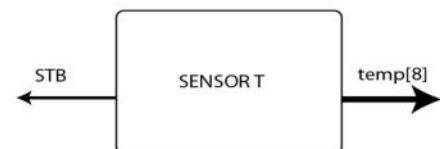
En la cuenca del río Santa Lucía se ha instalado un sistema de boyas para controlar la calidad del agua. Cada boya dispone de dos sensores, uno de PH y otro de Temperatura, un módem para comunicarse a través de la red de telefonía celular y un sistema basado en un **microprocesador T80** funcionando en modo 2 de interrupciones para gestionar todo el funcionamiento.

Los sensores se comunican con el microprocesador de forma asíncrona, usando interrupciones. El funcionamiento de cada uno se describe a continuación.

El sensor de PH codifica el valor sentido en 2 bytes, cuando tiene un nuevo valor pronto para comunicar, pone el primer byte en su salida **data(8)** y sube a 1 su pata **READY**. Espera un pulso a 0 en su entrada **ACK** y con el flanco de subida de **ACK** pone el segundo byte en la salida **data(8)**. La comunicación termina cuando recibe un segundo pulso en **ACK** y baja su pata **READY**. La pata **READY** se mantendrá baja hasta la próxima comunicación.



Por otro lado, el sensor de Temperatura codifica el valor sentido en 1 byte. Cuando tiene un nuevo dato pronto, lo coloca en su salida **temp(8)** y da un pulso a 1 en su salida **STB**.



El microprocesador gestiona las interrupciones y debe guardar los datos de los sensores en memoria RAM: los datos de PH a partir de la dirección **data\_PH** y los de temperatura a partir de la dirección **data\_T**.

Cuando el sistema haya recogido 64 datos de PH (128 bytes), o 128 datos de Temperatura (128 bytes), debe llamar a la subrutina **Transmisión** que se supondrá implementada. Esta subrutina utiliza el módem, y transmitirá 128 datos a partir de la dirección almacenada en el registro HL.

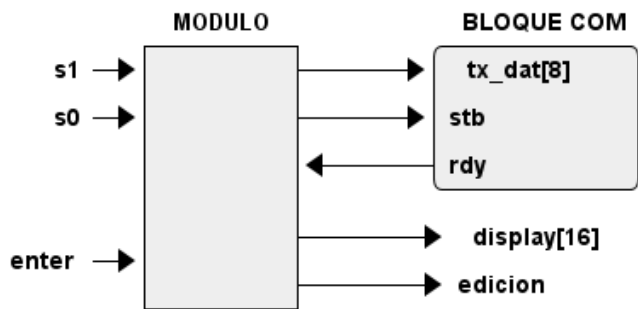
El sistema con microprocesador dispone de 32K de ROM y 32K de RAM. Se pide:

- Todo el HW del sistema con excepción de la memoria y la conexión de datos al módem que se supondrán ya implementados (microprocesador, sensores y controladores de interrupciones y sus conexiones). No puede usarse el rango 0x00-0x3F en el espacio de E/S, ya que está ocupado por el módem. Debe incluirse tres controladores de interrupciones. El módem se conecta a uno de ellos mediante las señales **IRQ\_modem** e **INTA\_n\_modem**. El orden de prioridades de interrupción debe ser: Modem. Sensor de Temperatura, Sensor de PH.
- Inicialización de todo el sistema: constantes, variables y lo necesario para que funcionen las interrupciones. El módem funciona por interrupciones utilizando el primer lugar en la tabla de interrupciones, la subrutina se encuentra en la dirección 0x3500.
- Subrutina de atención a la interrupción del sensor de PH. Tener en cuenta que en cada interrupción se deben leer los dos bytes que componen el dato.
- Subrutina de atención a la interrupción del sensor de Temperatura.

(El sistema ya cuenta con la subrutina Transmisión y la rutina de atención a la interrupción del módem cargadas en ROM.)

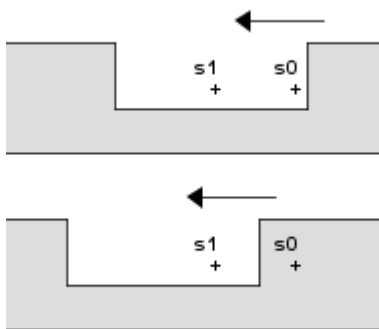
## PROBLEMA 2

Se debe diseñar un módulo periférico de un sistema de control industrial. El módulo debe permitir ajustar el valor deseado en una magnitud a controlar. Cuenta con un pulsador **enter** que genera un pulso a 0 cada vez que es presionado y una perilla giratoria equipada con dos sensores **s1** y **s0**. Cuenta además con una salida **display[16]** para mostrar al usuario el valor que se está seleccionando, una salida **edicion** para activar un indicador luminoso y un bloque de comunicación que envía el valor seleccionado al sistema de control central.



El sistema comienza en modo reposo con la magnitud fijada en **VAL\_INI**.

Un flanco de bajada en la entrada **enter** lo pone en modo edición. A partir de ese momento el giro de la perilla se usa para para incrementar o decrementar el valor seleccionado.



Para sensar la posición de la perilla se utiliza un disco ranurado adosado a su eje y dos sensores **s1** y **s0**. La salida de cada sensor toma el valor cero cuando está frente a una ranura y uno frente a un diente. Los sensores están ubicados a una distancia igual a media ranura de modo que el valor de **s1** en el momento de un flanco de subida de **s0** indica el sentido de giro de la perilla. Si la perilla gira en sentido positivo como en la figura, **s1** valdrá 0 (cambio de  $s_1s_0=00$  a  $s_1s_0=01$ ). Si gira en sentido contrario (negativo) valdrá 1 (cambio de  $s_1s_0=10$  a  $s_1s_0=11$ ).

Con cada flanco de subida de **s0** se modifica el valor seleccionado en +/- 1, dependiendo del sentido de giro, y se actualiza la salida **display** en forma acorde.

Con un nuevo flanco de bajada en **enter** se debe volver a modo reposo y transmitir los dos bytes del valor seleccionado en ese momento hacia el controlador central, comenzando por el byte menos significativo. Estando en modo reposo, no debe modificarse el valor seleccionado.

El bloque de comunicación con el controlador central indica con **rdy** = 1 que está pronto para recibir un byte. Cuando recibe un pulso de corta duración en **stb** pone **rdy** a cero y comienza a transmitir el dato presente en **tx\_dat[8]**. Cuando termina la transmisión lleva nuevamente **rdy** a uno.

A los efectos del problema se podrá suponer que el valor que alcanza la magnitud es siempre representable en complemento a 2 de 16 bits (no hay overflow).

La solución debe organizarse en un programa principal y una interrupción en modo 1:

- Interrupción por el flanco de subida de **s0** que incremente o decremente el valor seleccionado cuando corresponda y actualice el valor de **display** en forma acorde.
- El programa principal debe encargarse de la entrada **enter**, la salida **edicion** y la transmisión de datos hacia el controlador central.

Se dispone de un Z80 tradicional con bus triestado, memorias y lógica.

Se pide:

- a) Todo el hardware necesario.
- b) Todo el software que incluye:
  - i. El lazo infinito del programa principal.
  - ii. La rutina de atención a interrupción.
  - iii. La inicialización del sistema. Debe indicarse la ubicación en memoria de todo el código, tablas y variables utilizadas mediante directivas al ensamblador.