

APELLIDOS : _____
NOMBRE: _____ D.N.I.: _____

- LEER ATENTAMENTE EL ENUNCIADO.
- RAZONAR CADA UNA DE LAS DECISIONES ADOPTADAS TANTO EN EL HW COMO EN EL SW.
- CUIDAR LA PRESENTACIÓN EVITANDO ESQUEMAS ININTELIGIBLES, LETRA INDESCIFRABLE Y EXPRESIONES MALSONANTES.
- UTILIZAR PARA EL DISEÑO LOS CRITERIOS DE INGENIERÍA (MÁXIMA EFICIENCIA, MÍNIMO COSTO).
- UTILIZAR COMENTARIOS EN LA ELABORACIÓN DEL CÓDIGO SW.

CUESTIONES:

C1. Comentar como los microcontroladores PIC tienen distribuidas las zonas de código y de memoria RAM interna (0.5 puntos).

C2. Indicar como se programan las patillas de los microcontroladores PIC para que actúen como entradas o como salidas. (0.5 puntos).

C3. Enumera, comenta y pon un ejemplo de las diferentes topologías de REDES DE ÁREA LOCAL. (0.5 puntos).

C4. Enumerar los diferentes sistemas de interconexión (cableado) en RED LOCAL y compararlos indicando sus ventajas e inconvenientes. (0.5 puntos).

PROBLEMAS:

P1. (5 puntos) Disponemos de un sistema de domótica distribuido constituido por un nodo central y varios nodos esclavos basados en 8051. Se pretende realizar el programa de control de los nodos esclavos en ensamblador. Cada nodo esclavo dispone de los siguientes elementos:

- Una salida RS485C para comunicar con nodo central. Esta salida se implementa mediante un chip adaptador de niveles eléctricos. Se emplea la patilla P1.6 del microcontrolador como señal de dirección de los datos.
- Tres salidas, que activarán tres leds (rojo en P1.3, amarillo en P1.4 y verde en P1.5).
- Tres entradas en las que hay conectados tres interruptores (en P1.0, P1.1 y P1.2).

El **protocolo de comunicación** consiste en lo siguiente: El maestro envía una orden a cada uno de los esclavos y cada nodo esclavo responde una vez realizada la orden con un paquete de confirmación de orden.

El **formato de paquete** es el siguiente: un byte de comienzo, un byte que indica el código del destinatario, otro que indica el código del remitente y un byte de comando y otro opcional de datos. Se finaliza con un byte de fin. Nuestro sistema recibirá paquetes de este tipo:

BYTE de Comienzo (0AH)	Destinatario (Binario)	Remitente (Binario)	BYTE de Instrucción (ASCII Mayúscula)	N Datos Opcionales (ASCII)	BYTE de Fin (0DH)	$N \in \{0,1\}$
------------------------	------------------------	---------------------	---------------------------------------	----------------------------	-------------------	-----------------

Y transmitirá paquetes con este formato:

BYTE de Comienzo (0AH)	Destinatario (Binario)	Remitente (Binario)	BYTE de Instrucción (minúscula)	byte de Datos (ASCII)	BYTE de Fin (0DH)	$M \in \{0,1\}$
------------------------	------------------------	---------------------	---------------------------------	-----------------------	-------------------	-----------------

De forma que se puede saber si un paquete es de envío o de respuesta porque su byte de instrucción está en mayúscula (petición u orden) o en minúscula (reconocimiento o ACK de orden). El programa atiende la comunicación y para cada comando recibido realiza las siguientes tareas:

Comando	Valor ASCII	Tarea	Bytes recibidos	Bytes Transmitidos
'Q'	51H	enciende el led rojo	5	5
'A'	41H	apaga el led rojo	5	5
'W'	57H	enciende el led amarillo	5	5
'S'	53H	apaga el led amarillo	5	5
'E'	45H	enciende el led verde	5	5
'D'	44H	apaga el led verde	5	5
'T'	54H	Envía el estado de los tres terminales de entrada (pulsadores), como un número en código ASCII comprendido entre 0 y 3.	5	7
'C'	43H	Hace eco del siguiente byte recibido	6	6

Emplear la interrupción del puerto serie para recibir los comandos de la interfaz de comunicación, recordar que para cada byte recibido y/o transmitido se genera una interrupción una vez programado el registro de control de interrupciones.

Los datos de los sensores se muestrean y se envían únicamente cuando se reciba la petición o se pulse el pulsador de envío. Usar el Timer1 en modo autorrecarga como generador de baudios.

El programa podría comenzar así:

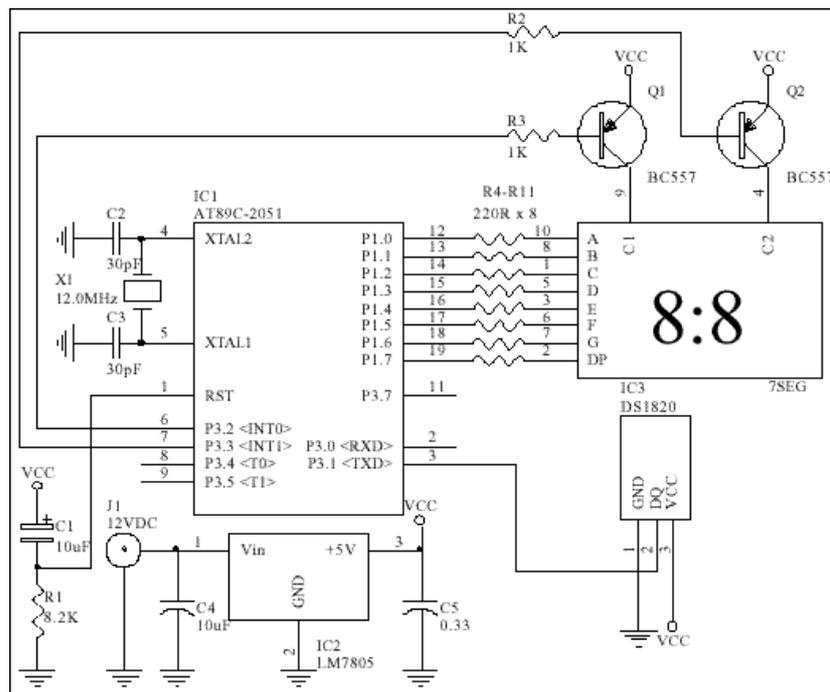
```

;*****
; Programa para la comunicación del bus RS-485
;*****
; Declaración de constantes y de variables
;*****
ID_MOD      EQU    #02H ;Número de identificación de este módulo
ID_Master   EQU    #00H ;Número de identificación del maestro

Orden_in    EQU    10H  ;Variable que indica el n° de byte recibido
Haz_Eco     EQU    13H  ;Variable que indica la acción de eco
Dato_Eco    EQU    14H  ;Variable que contendrá el dato enviado por el
maestro
;*****
; Rutina de Vectorización
;*****
ORG    0H
LJMP   Inicio
ORG    023H      ; Vectorización del puerto de comunicación serie
LJMP   RSI_RS
.....

```

P2. Se dispone de un sistema basado en el 8051 con una frecuencia de reloj de 12 MHz, equipado con un sensor de temperatura y dos displays como muestra la figura. El objetivo de este diseño es realizar la lectura del sensor en formato serie y mostrarla en el display.



- (1 punto) Realizar el diagrama de bloques del programa teniendo en cuenta que la lectura del sensor la realiza una rutina llamada lee_sensor que ya está codificada en ensamblador y devuelve los dos dígitos en formato BCD empaquetado en el acumulador. El programa principal consistirá en un bucle que realiza la lectura del sensor y muestra el resultado en pantalla.
- (2 puntos) Realizar una rutina en lenguaje ensamblador del microcontrolador 8051 basándose en el diagrama de bloques anterior que escriba en el puerto 1 y el puerto 3 y actualice el display con una frecuencia de refresco de 50Hz.