

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

CUESTIONES:

- C1. Dibuja el organigrama básico del proceso de desarrollo de programas (compilación o ensamblado, etc..). (0.5 puntos).
- C2. Enumera y explica los pasos y tareas que hay que realizar para una correcta modularización de los programas. (0.5 puntos).
- C3. Enumera los diferentes espacios de memoria que gestiona el compilador de lenguaje C del 8051, indica para que se emplea cada uno y explica porque son necesarios en la definición de variables y punteros. (0.5 puntos).
- C4. Enumerar y comentar los diferentes tipos de interfaces conectados a sus buses de datos y direcciones que puede tener un sistema microcomputador de control industrial (como están compuestos y que finalidad tienen). (0.5 puntos).
- C5. Comenta la manera de codificar en lenguaje C una interrupción y como lleva a cabo el código generado por el compilador de C KEIL la gestión de las interrupciones (0.5 puntos).

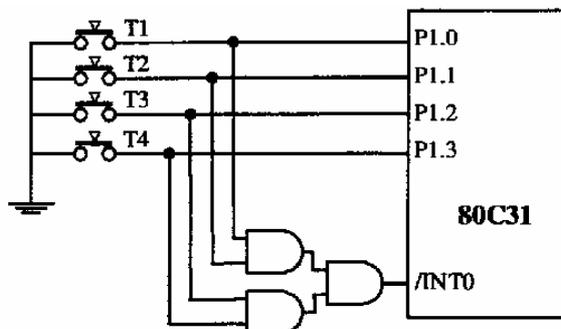
PROBLEMAS:

P1. (2 puntos) Escribir el código en C correspondiente a la función de conversión de código binario a código ASCII. El código ASCII del número '0' es 0x30 y el de la letra 'A' es el 0x41. La función devuelve un byte con el código ASCII correspondiente al byte en código binario que toma como parámetro:

```
unsigned char bcd_to_ascii(unsigned char cod_bcd);  
// cod_bcd va de 0x00 a 0x0F
```

P2.

a) Traducir el código ensamblador a C escribiendo código únicamente dentro de las cajas.



```

;*****
; Rutina de vectorización
;*****
ORG 0H
    LJMP inicio
ORG 03H
    LJMP RSI_INT0
;*****
; Rutina de inicio.
; (habilita interrupciones y /INT0 flanco descendente)
;*****
Inicio:
    SETB PX0          ;Prioridad alta para /INT0 (Registro E)
    SETB EX0          ;Habilitación de /INT0 (Registro E)
    SETB EA           ;Habilitación general (Registro E)
    LJMP Principal
;*****
; Programa Principal
;*****
ORG 0300H
Principal:
    SJMP Principal    ;Bucle infinito sin propósito definido
;*****
; Rutina de RSI de /INT0
;*****
RSI_INT0:
    JNB P1.0,Tecla_1    ;¿Es la tecla T1?
    JNB P1.1,Tecla_2    ;¿Es la tecla T2?
    JNB P1.2,Tecla_3    ;¿Es la tecla T3?
    SJMP Tecla_4        ;ninguna, es tecla T4
Tecla_1: MOV B,#01H      ;Pone 01H en B
    SJMP Salir          ;Ir a Salir
Tecla_2: MOV B,#02H      ;Pone 02H en B
    SJMP Salir          ;Ir a Salir
Tecla_3: MOV B,#03H      ;Pone 03H en B
    SJMP Salir          ;Ir a Salir
Tecla_4: MOV B,#04H      ;Pone 04H en B
Salir:   CLR IE0         ;Se pone a cero para que /INT0
                    ; pueda interrumpir de nuevo.
    RETI                ;Final de interrupción

```

El programa consiste en un tratamiento de las teclas mediante interrupción. Como variable compartida entre programa principal y rutina de interrupción se emplea el registro B. Utilizar como variable compartida la variable global “tecla”. (2 puntos)

```

// Atención del teclado
// Ficheros "include"

// Declaraciones de constantes y puertos de E/S

// Variables globales

// Rutina de inicialización
void Inicio (void) {

}
// Rutina de servicio de /INT0
void RSI_int0(void) interrupt [] using [] {

}
// programa principal
void main(void) {

}

```

Registro IP						
(MSB)						(LSB)
---	PPC†	PT2*	PS	PT1	PX1	PX0
Bit	Comentario					
PX0	Bit de prioridad de /INT0					
PT0	Bit de prioridad del Timer 0					
PX1	Bit de prioridad de /INT1					
PT1	Bit de prioridad del Timer 1					
PS	Bit de prioridad del puerto serie					
PT2	Bit de prioridad del Timer 2					
PPC	Bit de prioridad de la PCA					
-	Bit reservado					

b) Ampliar el programa principal (función main del apartado anterior) de modo que convierta la tecla a un número comprendido entre 1 y 3 y lo muestre en un display en formato BCD conectado al puerto P1, de forma que el bit 0 del puerto es el bit de menos peso y el bit 4 del puerto es el bit de más peso en código BCD. (1.5 puntos)

P3. (2 puntos) En una placa de desarrollo basada en el 8051 con 8K de ROM se pretende colocar una zona de RAM de 8Kbytes compartida con los mapas de memoria externa de datos y de código a continuación de la memoria ROM. Las señales de control de la memoria RAM son /CS, /RD y /WR y las de la ROM /CE y /OE.

- a) Dibujar el mapa de memoria resultante del sistema.
- b) Diseñar el circuito de control de los chips de RAM y ROM, emplear todos los bits de direcciones para realizar la decodificación.
- c) Diseñar el circuito de control para la decodificación de un dispositivo periférico con un registro de control/estado. Colocarlo en la dirección 0x8000, no es necesario emplear todos los bits del bus de direcciones.