



Universidad  
de Huelva

Escuela Politécnica Superior  
Universidad de Huelva

TERCER CURSO. ELECTRÓNICA DIGITAL

# PRÁCTICAS DE PROGRAMACIÓN CON EL 8051

Manuel Sánchez Raya  
Versión 1.0  
20 de Abril de 2000

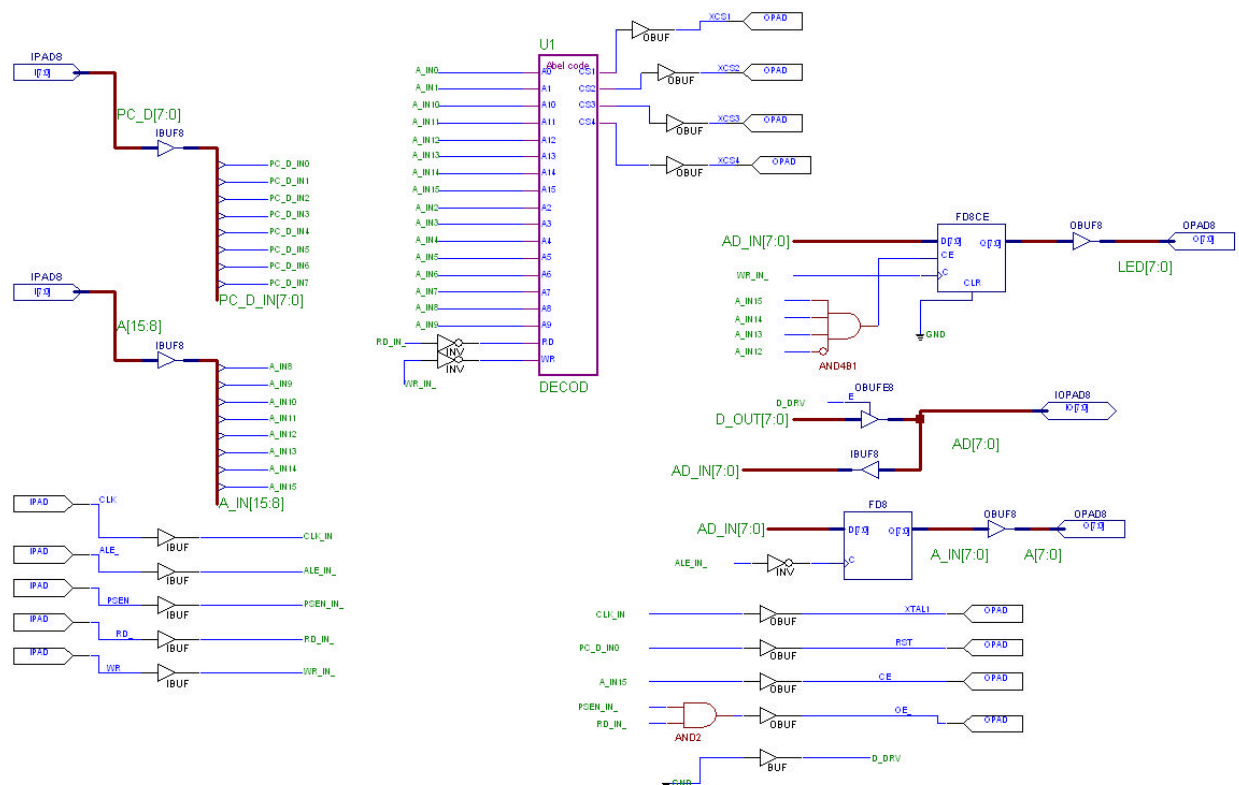
# Práctica 1: INTRODUCCIÓN A LA PROGRAMACIÓN EN ENSABLADOR

## OBJETIVOS:

- Visión general del codiseño FPGA/?C.
- Diseño de periféricos para ?C 8031.
- Manejo del entorno “? Visión/51”.
- Manejo del debuguer “dScope”.
- Creación de un programa en ensamblador.
- Comprensión del funcionamiento de la placa LED-XESS.

## INTRODUCCIÓN:

Esta práctica consiste en generar el fichero de configuración de la FPGA usando el entorno de desarrollo con FPGA de XILINX M2 y la generación de un programa que se ejecute sobre el 8031 cuyo fin sea efectuar una cuenta y su visualización sobre el display que incorpora la placa.



Departamento de Ingeniería Electrónica	Project: XESS-LED
Escuela Politécnica Superior	Sheet: nuevo
Campus La Rabida	Date: 1/8/98
Date Last Modified: 5/1/0	

La placa FPGA XESS 1.4 se conectará sobre una placa de desarrollo (LED) que incorpora dos puertos de salida (solo escritura) un puerto de E/S (lectura/escritura), tres pulsadores: dos conectados a las interrupciones externas y uno a la patilla de reset. Un zumbador piezoeléctrico, un led, una interfaz serie RS-232 para comunicar el ?C al PC y un display LCD.

Recurso	Puerto	Acceso
Zumbador	P3.5 / T1	Escritura
Pulsador 0	P3.2 / INT0	Lectura
Pulsador 1	P3.3/ INT1	Lectura
LED 0	P3.4 / T0	Escritura

La interfaz ?C/PC se ha diseñado de tal forma que los puertos se encuentran colocados en la memoria externa de datos, por lo que para acceder a ellos habrá que emplear instrucciones MOVX y el puntero de datos DPTR. La tabla siguiente muestra la dirección de estos puertos.

Dirección	Puerto	/RD	/WR	/CS1	/CS2	/CS3	/CS4
EXXXH	LED 7SEG	1	0	1	1	1	1
FFF0H	XP0(lec.)	0	1	0	1	1	1
FFF0H	XP0(esc.)	1	0	1	0	1	1
FFF1H	XP1(solo esc.)	1	0	1	1	0	1
FFF2H	LCD datos (lec/esc)	X	X	1	1	1	0
FFF3H	LCD control (lec/esc)	X	X	1	1	1	0

La conexión del display interno de la placa XESS al puerto E000H tiene lugar de la siguiente forma: dp a f b g e c d, desde el bits más significativo al menos significativo, donde los segmentos son los siguientes: a superior, b el de la izq más arriba, c el de la izq. abajo, d el inferior, e el de la derecha abajo, f el de la derecha arriba y g el central; dp es el punto decimal. Por ejemplo, si queremos encender el 0 tendremos que escribir el dato 77H al puerto E000H, el 12H para el 1, etc.

El código fuente del decodificador de direcciones es el siguiente:

---

Equations

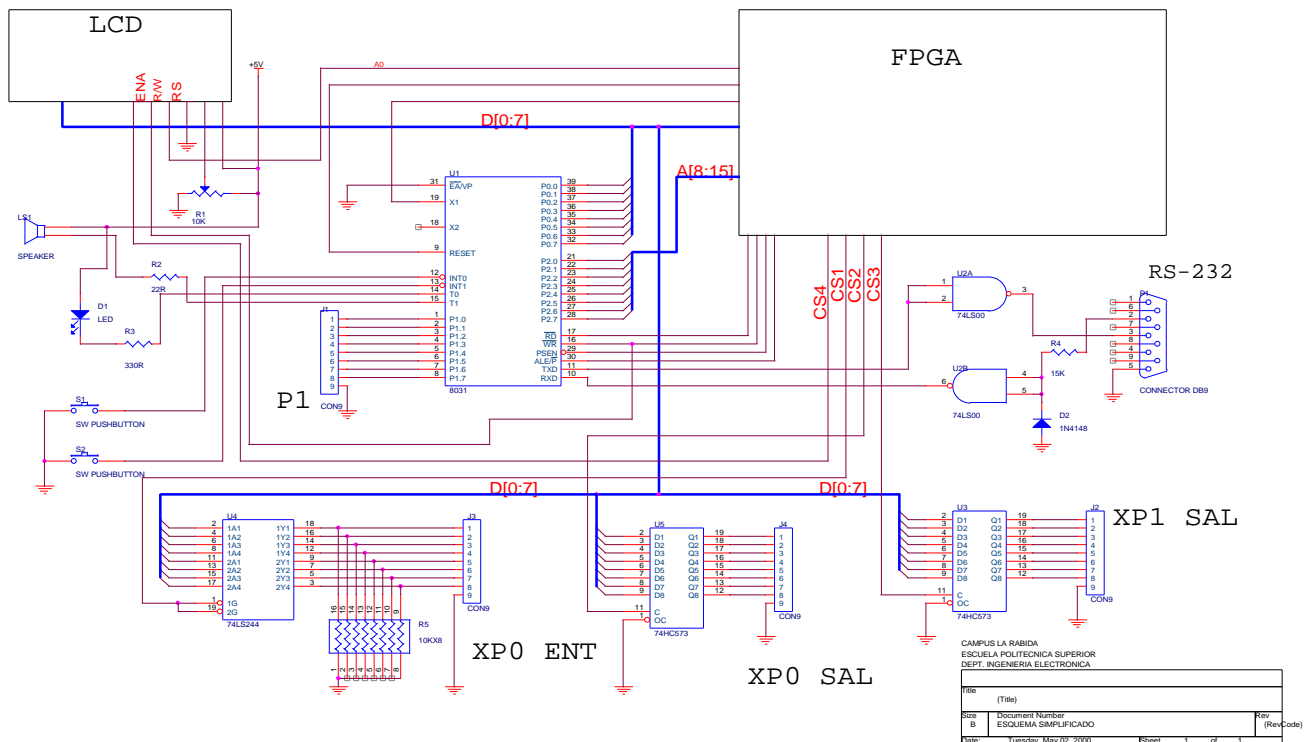
```

Truth_table ([A , RD, WR] -> [ CS1 ,  CS2 ,  CS3 ,  CS4 ])
    [ ^HFFF0, OFF, ON ] -> [ OFF,  ON,  ON,  ON ];
    [ ^HFFF0, ON , OFF] -> [  ON, OFF,  ON,  ON ];
    [ ^HFFF1, ON , OFF] -> [  ON,  ON, OFF,  ON ];
    [ ^HFFF2,  X ,  X ] -> [  ON,  ON,  ON, OFF ];
    [ ^HFFF3,  X ,  X ] -> [  ON,  ON,  ON, OFF ];

end decod

```









---



### ESPECIFICACIONES:

El programa se realizará en ensamblador. Consiste en definir una variable y una rutina de conversión BCD -> 7 segmentos. El programa principal inicializará la variable e ira incrementándola hasta llegar a 0FH, momento en el que pasará a valer cero tras su incremento, volviendo a realizar la secuencia. Para poder observar la cuenta se empleara un pequeño bucle de retardo y se programara para generar un retardo de un segundo aproximadamente.

### PROCESO OPERATIVO:

-   Editar el código fuente en ensamblador correspondiente: PRA1.A51.
-   Compilar el archivo anterior mediante la sentencia:  
*ASM51 PRA1.ASM*
-   Simular el diseño mediante la aplicación “dScope”.
-   Por último se procederá a la descarga del fichero de programación del ?C (extensión HEX) mediante la aplicación XSLOAD. Puesto que la configuración ha sido guardada en memoria FLASH sobre CPLD solo será necesario grabarla una sola vez.

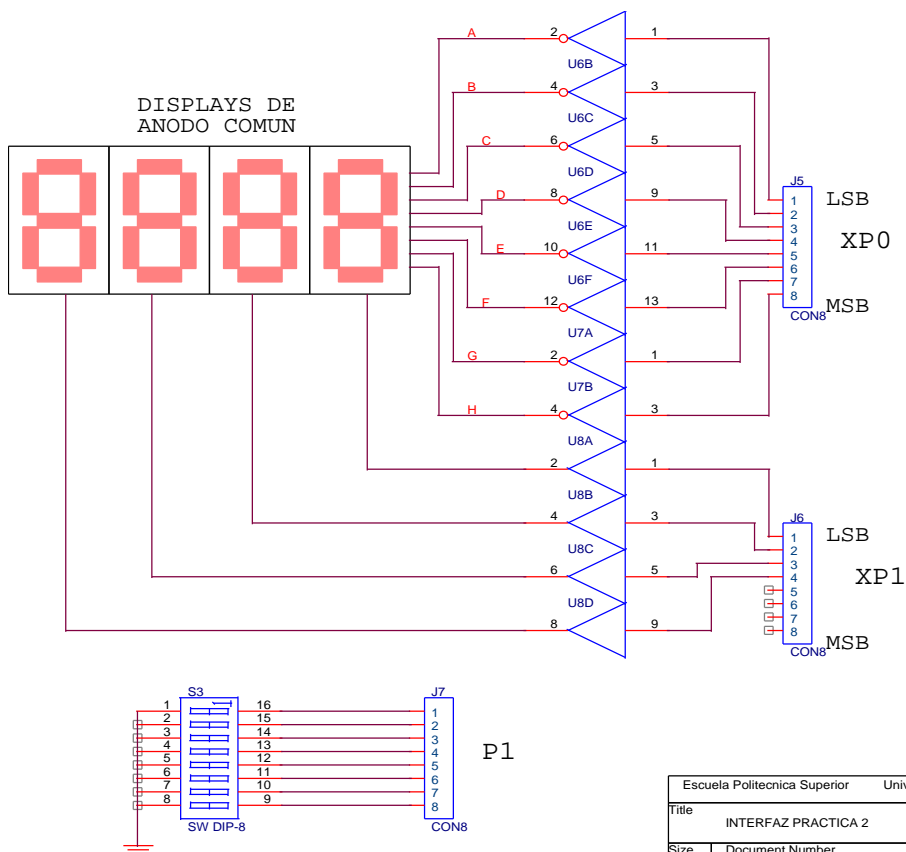
# Práctica 2 : INTRODUCCIÓN A LA PROGRAMACIÓN EN C.

### OBJETIVOS:

- ? ? Manejo de la documentación de los fabricantes de microcontroladores (?C).
- ? ? Iniciación en el uso de la tarjeta de ampliación LED-XESS.
- ? ? Configuración y uso de los timers.
- ? ? Manejo de los puertos.
- ? ? Control de varios displays con una única salida mediante multiplexión temporal.

## INTRODUCCIÓN:

En la presente práctica se va a hacer uso de una tarjeta de aplicación para el microcontrolador 8031 cuyo esquema se representaba en la práctica anterior. En dicha tarjeta el Puerto XP0 está conectado a un conjunto de cuatro displays de 7 segmentos LED y el puerto interno P1 del microcontrolador a ocho microinterruptores. Además, el puerto XP1 se utiliza para el control de cuatro displays y por último el Puerto 3 se emplea para las entradas de dos pulsadores externos en las patillas /INT0 e /INT1.



Escuela Politecnica Superior		Universidad de Huelva	
Title INTERFAZ PRACTICA 2			
Size A	Document Number ESQUEMA		Rev 001
Date:	Monday, May 15, 2000	Sheet	2 of 2

Para encender un display se colocará en los ocho bits del puerto XP0 el código en siete segmentos correspondiente al dígito que se quiere representar, de forma que para encender el segmento se coloca un cero. El bit más significativo está conectado al punto decimal y después la conexión se realiza a los segmentos g f e d c b y el menos significativo al segmento a. Por los

cuatro bits de menor peso del puerto XP1 se pondrá a cero el bit correspondiente al display que se desea activar.

Activando cíclicamente cada display durante un tiempo determinado (por ejemplo 250 ?S) se tendrá la sensación de que todos están encendidos al mismo tiempo.

#### ESPECIFICACIONES:

Realizar un programa en C para el microcontrolador 8031 tal que el funcionamiento del sistema sea el siguiente, teniendo en cuenta los posibles rebotes de los pulsadores:

- ✂✂Cada pulsación de S0(patilla INT0) hará que se lea el contenido del puerto P1 y se visualice el número en decimal en los tres displays de la derecha (el cuarto permanecerá apagado).
- ✂✂Las pulsaciones consecutivas de S1(patilla INT1) harán que el número visualizado se incremente o no cada segundo.
- ✂✂La implementación se realizará empleando dos métodos: **Polling** o muestreo sobre las patillas donde se conectan los pulsadores en el puerto 3, P3.2(S0) y P3.3(S1); **Interrupciones**, programando adecuadamente el registro de control de interrupciones IE.  
¿Qué sucede con los rebotes ? y ¿cómo solucionarlo?

#### PROCESO OPERATIVO:

- ✂✂Crear el proyecto que contendrá la práctica usando el entorno “?Visión/51”
- ✂✂Editar el modulo fuente del programa en C.
- ✂✂Compilar el módulo del programa.
- ✂✂Realizar el montaje del proyecto.
- ✂✂Simular el diseño mediante la aplicación “dScope”.
- ✂✂Una vez corregidos todos los errores posibles mediante la simulación se volcará el código binario (archivo .HEX) sobre la placa XESS mediante la aplicación XSLOAD.

## Práctica 3 : PROGRAMACIÓN MODULAR Y MANEJO DEL PUERTO SERIE.

### OBJETIVOS:

- Manejo de la documentación de los fabricantes de microcontroladores (?C).
- Uso de las comunicaciones internas y externas.
- Comunicación con un PC mediante el puerto serie.
- Creación de un diseño con varios módulos.

### INTRODUCCIÓN:

En la presente práctica se va a hacer uso de la tarjeta de aplicación para el microcontrolador LED-XESS cuyo esquema se representa en la siguiente figura:

<figura de practica 3>

En el esquema anterior puede observarse como los pines RxD y TxD del microcontrolador se encuentran conectados, a través del correspondiente interface, a un conector de 9 pines que permitirá establecer una comunicación serie con un PC.

Por otro lado, se ha conectado un altavoz al pin 7 del puerto 3, mediante el cual podrá oírse la señal que se está generando.

Para la transmisión serie se usará el timer 1 en modo 2 como generador de baudios. El valor a cargar en TH1 vendrá dado por la fórmula que se representa a continuación.

$$TH1 = 256 - \frac{Frecuencia\ del\ oscilador\ (Hz)}{384 \cdot Velocidad\ de\ transmisión\ (baudios)} 2^{SMOD}$$

### ESPECIFICACIONES:

En la presente práctica se implementará un programa que genere una onda cuadrada de frecuencia variable entre 20 y 9999 Hz, la cual será aplicada al altavoz. La frecuencia de la onda a generar se indicará desde el PC a través de un puerto de comunicación serie RS-232 configurado a una velocidad de 2400 baudios.

El funcionamiento será el siguiente:

Cada vez que se pulse S2 aparecerá en el monitor del PC el mensaje:

**Introduzca la frecuencia deseada [20-9999 Hz]**

Si se introduce una frecuencia fuera del intervalo se visualizará:

**¡Frecuencia fuera de rango!**

**Introduzca otra frecuencia**

En caso de que la frecuencia introducida esté comprendida dentro del rango especificado, se visualizará el mensaje:

**Ha seleccionado una frecuencia de [frecuencia elegida] Hz.**

Acto seguido se presentará en los cuatro displays de la tarjeta de aplicación la frecuencia seleccionada expresada en Hz, se generará la onda y se aplicará al altavoz para que pueda ser oída.

#### PROCESO OPERATIVO:

- Editar el código correspondiente a la práctica que estará compuesto por dos módulos P3\_a.c y P3\_b.c.

**P3\_a.c** será el módulo principal que constará de:

- Definición de variables globales.
- Rutinas de atención a las interrupciones.
- Programa principal.

**P3\_b.c** incluirá:

- Una función para leer la frecuencia a través del puerto serie.
- Una función que determine el dígito a representar en cada display, partiendo de la frecuencia introducida.
- Compilar los archivos anteriores mediante las sentencias:  
*C51 P3\_a.c DEBUG*  
*C51 P3\_b.c DEBUG*
- Realizar el montaje del proyecto mediante:  
*L51 P3\_a.obj, P3\_b.obj to P3*
- Simular el diseño mediante la aplicación “dScope”.
- Una vez corregidos todos los errores posibles mediante la simulación se generará el fichero en formato INTEL-HEX mediante:  
*OH51 P3*
- Por último se procederá a la descarga del fichero de configuración de la FPGA (extensión BIT) y de programación del ?C (extensión HEX) simultáneamente mediante la aplicación XSLOAD.

La figura siguiente ilustra gráficamente la secuencia de pasos a seguir:

<figura>



## Práctica 4. TRABAJO CON LIBRERIAS Y CONTROL DE DISPLAY LCD.

### OBJETIVOS:

- Manejo de documentación de fabricantes de ?C.
- Manejo del entorno integrado de programación en C.
- Obtención de código modularizado y funciones C para utilización en programas modulares.

### INTRODUCCIÓN:

La práctica que se presenta consiste en la elaboración de una librería de funciones que facilite el acceso SW a una unidad LCD paralelo. La unidad LCD que se va a utilizar en este caso es la HD 44780 de HITACHI.

A continuación se presenta la definición de los módulos que deberá poseer la librería:

Definición de bits de control : E, RS, RW

- void dato\_display(char dato)
- void control\_display(char dato)
- void init\_display(void)
- void borrar\_display(void)
- void locate(short int fila, short int col)
- char putchar(char c)
- void scroll\_x1\_y1(short int x1, short int y1, char \*p) *OPCIONAL*

### ESPECIFICACIONES:

Se presentan las especificaciones mínimas de cada uno de los módulos que se incluyen en la librería de funciones para HD44780:

- La librería de funciones se almacenará en un fichero de texto denominado “HD44780.c” y permitirá su inclusión en otros programas mediante la directiva #include.
- void dato\_display(char dato): Esta función envía un dato al display según la información adjunta en el anexo A.
- void control\_display(char dato): esta función envía un dato de control al display según la información adjunta en el anexo A.
- void init\_display(void): esta función envía al display la información necesaria para configurarlo de la siguiente manera:
  - Selección del interfaz en modo de control de 8 líneas.
  - Visualización en 2 filas.
  - Fuentes de caracteres de 5x10 puntos.
  - Incrementa el puntero de dirección después de cada operación de escritura/lectura.
  - Deja encendido el visualizador.
  - Pone activo el cursor.

- void borrar\_display(void): mediante esta función se borra el contenido del display y se vuelve el cursor a la posición cero.
- void locate(short int fila, short int columna): esta función pone el cursor en la coordenadas fila, columna del display.
- char putchar(char c): a través de putchar se envía un carácter ASCII al display.
- void scroll\_x1\_x2(short int x1, short int x2, short int fila, char \*p) : con esta función se realizará un “scroll” del texto almacenado en \*p apareciendo el primer carácter en la posición (fila, x1) y desapareciendo el último carácter en la posición (fila, x2).

Para la comprobación de las funciones implementadas se desarrollará un programa en “C” que hará uso de todas las funciones desarrolladas y *aquellas que por iniciativa propia se quieran generar*. El programa se denominará “PRA4.c” e incluirá la biblioteca de rutinas diseñada.

Para la elaboración de funciones de control del display se deberá consultar la información adjunta en el Anexo A.

#### PROCESO OPERATIVO:

- Editar el código correspondiente a la práctica que estará compuesto por la librería y por un módulo de test: PRA4.c y HD44780.c.

**PRA4.c** será el módulo principal que constará de:

- Definición de variables globales.
- Rutinas de atención a las interrupciones.
- Programa principal.

**HD44780.c** incluirá las funciones de librería anteriormente definidas.

- Compilar los archivos anteriores mediante las sentencias:  
C51 PRA4.c DEBUG  
C51 HD4478.c DEBUG
- Realizar el montaje del proyecto mediante:  
L51 HD44780.obj, PRA4.obj to P4
- Simular el diseño mediante la aplicación “dScope”.
- Una vez corregidos todos los errores posibles mediante la simulación se generará el fichero en formato INTEL-HEX mediante:  
OH51 P4
- Por último se procederá a la descarga del fichero de configuración de la FPGA (extensión BIT) y de programación del ?C (extensión HEX) simultáneamente mediante la aplicación XSLOAD.