

"SISTEMA DE DESARROLLO PARA APLICACIONES BASADAS EN μ CONTROLADOR"

Fernando Quero Sanz, Rafael Embid Romero, Marta Arguedas Lafuente,
Javier Cuevas Domingo, José Luis Vela Pérez, Gonzalo Morales Ojeda

Grupo de Sistemas Digitales
(ESCUELA UNIVERSITARIA POLITÉCNICA LA ALMUNIA DE D^a GODINA)

e-mail: fernando@eupla.unizar.es

Resumen

Los equipos de prácticas de distintas especialidades tienen un precio muy elevado y son muy dispares entre sí, no ajustándose completamente al programa de las diferentes asignaturas que pueden abarcar. Se plantea interesante en nuestro Grupo de Trabajo desarrollar un "SISTEMA DE DESARROLLO PARA APLICACIONES BASADAS EN μ CONTROLADOR", de forma que fuese lo más polivalente posible.

En este sistema se incorporan todos los recursos característicos en el diseño de sistemas con microprocesadores tales como; la típica expansión de buses de datos y direcciones "mapeando" todos los dispositivos direccionados por el microprocesador, la incorporación de dispositivos I2C que con tan solo dos líneas permiten ser gestionados por la CPU, comunicaciones RS-232 y RS-485 que permiten la posibilidad de crear sistemas distribuidos. De forma paralela se introducen todos los bloques combinacionales y secuenciales que anteriormente se implementaban con componentes discretos de la serie TTL 74xx o CMOS 40xx en PLD's y FPGA's programados con lenguajes descriptivos de hardware como VHDL.

1 Introducción

En el terreno industrial los μ C de 8 bits son empleados frecuentemente en aplicaciones de mediana complejidad, por su facilidad de manejo y excelente nivel de prestaciones. Existen en el mercado numerosos fabricantes que ofertan familias de μ C orientadas algunas de ellas a aplicaciones industriales muy específicas; de entre todas ellas hemos optado para abordar este proyecto la familia MCS-51, por su gran difusión industrial, excelente relación calidad/precio y gran número de fabricantes que lo desarrollan, que de forma continuada introducen nuevas versiones con interesantes innovaciones en cuanto a arquitectura, sin alterar la compatibilidad en cuanto al software con el estándar de la familia.

2 Especificación general del sistema

Se trata de realizar el diseño de una arquitectura específica

para esta familia de μ C, enfocada a la docencia de aplicaciones basadas en microprocesador. El principal objetivo es implementar un entorno expandible a un número ilimitado de aplicaciones, de forma que se cubra el contenido práctico de asignaturas afines a esta disciplina en carreras Universitarias de Ingeniería en la especialidad Electrónica e Informática de Sistemas.

El desarrollo del proyecto comprende las siguientes fases:

- Diseño de una arquitectura base.
- Diseño de un programa gestor sobre PC.
- Diseño de un kernel de un tiempo real.
- Diseño de aplicaciones.

Las fases del diseño siguen las pautas de cualquier diseño industrial y son planteadas a nivel docente, en asignaturas tales como Diseño de Sistemas basados en microprocesador e Instrumentación y Microprocesadores. En este apartado se hace uso de las diferentes herramientas de desarrollo tales como, simuladores, emuladores, analizadores lógicos, etc; empleadas en el diseño de sistemas con microprocesador, según el organigrama presentado en la Figura 1.

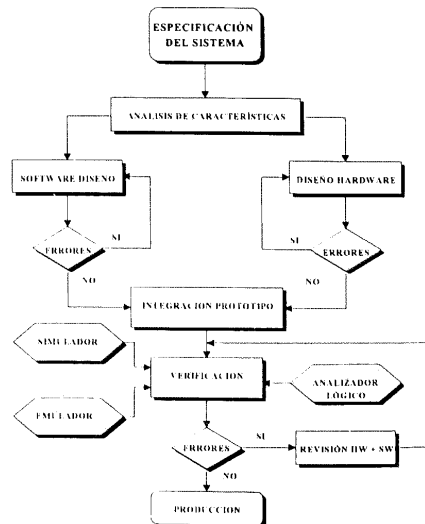


Figura 1

En la Figura 2 se presenta los diferentes bloques en los que se ha dividido el desarrollo del sistema.

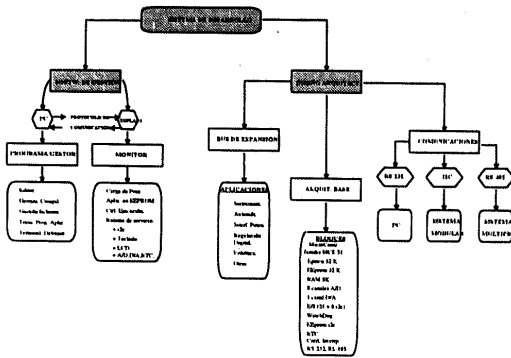


Figura 2

2.1 Especificación de la arquitectura base

El núcleo del presente proyecto reside en el diseño de una arquitectura con los elementos que componen el sistema mínimo del μC y elementos cuya incorporación potencie el conocimiento de diversos dispositivos fundamentales en diseño de sistemas con microprocesadores. El sistema cuyo diagrama de bloques se presenta en la Figura 3 dispone de un bus de expansión de forma que se puedan controlar diversas tarjetas de aplicación, consiguiéndose un sistema modular y muy flexible.

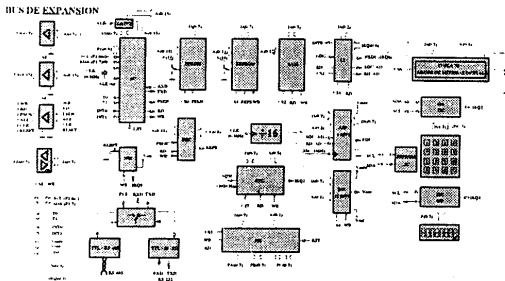


Figura 3

Otro bloque necesario es un interface serie RS-232 de comunicación hacia el PC, el cual nos permitirá gestionar la transmisión de programas de aplicación y posterior depuración de estos.

La filosofía con la que se ha abordado este diseño permite el crecimiento del sistema sin más que ir conectando varias tarjetas entre sí mediante un interface RS-485, de forma que éstas trabajen en red consiguiendo un entorno distribuido.

Así pues dividiremos el desarrollo de la placa base en dos apartados; por un lado el desarrollo Hardware y por otro el Software de gestión de la tarjeta.

2.1.1 Características hardware

- μC de la familia MCS-51 a 16 MHz, pudiéndose

conectar desde un 8031 hasta un 80652, el cual posee líneas de control para bus I²C.

- 32 KB de memoria EPROM para programa monitor de gestión de la tarjeta.
- 32 KB de memoria E²PROM en la cual se almacenará el programa de aplicación.
- 8 KB de memoria RAM.
- 8 canales A/D de 8 bits. 1 canal D/A de 12 bits.
- 24 líneas de E/S digitales de carácter general.
- 8 líneas de E/S digitales controladas en I²C.
- 1 Kbit de memoria E²PROM serie controlada en I²C.
- Controlador de interrupciones, disponiendo de 7 líneas de interrupción.
- Teclado Hexadecimal y pantalla LCD alfanumérica de 2x16 caracteres.
- WatchDog de programa y de alimentación.
- Interface serie RS-232 para comunicación con PC
- y RS-485 para comunicaciones multiprocesador.
- Reloj en tiempo Real.
- Bus de expansión.

En la Figura 4 se presenta una vista del Sistema de desarrollo EUPLA'51.

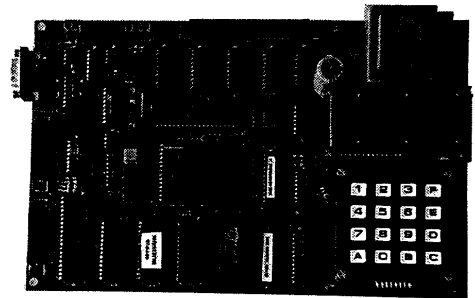


Figura 4

2.1.2 Características software

El alumno realizará los programas de aplicación en ensamblador o lenguaje C específicos de esta familia de μC , y el fichero objeto deberá ser formateado según el formato hexadecimal de Intel. Estos programas deberán ser enviados vía serie al sistema.

La placa base dispondrá de un programa monitor con las siguientes funciones de comunicación con el PC:

- Recepción de programas de aplicación desde PC.
- Protocolo de comunicación para control de errores en la transmisión.
- Ejecución paso a paso y transmisión de registros y variables al PC, para depuración en el desarrollo de programas.

El monitor estará contenido en el bloque de memoria EPROM y respecto al Sistema de desarrollo contendrá las siguientes funciones:

- Cargador de programas de aplicación recibidos vía

serie desde el PC sobre el bloque de memoria EEPROM.

- Ejecución de programas de aplicación desde EEPROM.
- Control de ejecución de programas paso a paso.
- Funciones de servicio propias de la placa base y funciones de alguna aplicación específica., por ejemplo funciones de tratamiento del convertidor A/D, D/A, RTC, etc..

En el desarrollo de un programa el alumno deberá de seguir una secuencia de operaciones en el Sistema, haciendo uso de las siguientes funciones del monitor anteriormente descritas (recepción, ejecución, depuración, etc.); toda esta evolución en el desarrollo de un programa podrá ser almacenada en un banco de la memoria EEPROM (no volátil) con los tiempos concretos en los que el alumno ha accedido a cada una de estas fases.

El profesor puede solicitar desde PC al Sistema mediante un determinado comando, cual ha sido la actividad del alumno en el desarrollo de un programa, consiguiéndose evaluar el trabajo con un mayor criterio.

2.2 Especificación del programa gestor del PC

Se trata de un entorno de desarrollo el cual integrará las siguientes herramientas:

- Editor de textos.
- Herramientas de ensamblado y compilación.
- Gestión de ficheros.
- Terminal de recepción con la tarjeta base.
- Transmisión de ficheros *.HEX a la tarjeta base.

3 Aplicaciones

La placa base está dotada de diversos bloques enfocados al control de parámetros tanto de tipo analógico como digital; sin embargo puede ser interesante la incorporación de tarjetas que conectadas en el bus de expansión, potencien dicho sistema en determinadas áreas de entre las que podemos destacar las siguientes:

• INSTRUMENTACIÓN:

Circuitos acondicionadores enfocados a la medida de parámetros analógicos con salida de control analógica actuando sobre un determinado entorno físico.

• SERVOMECANISMOS, REGULACIÓN DIGITAL:

Control analógico o digital (posición, velocidad) mediante controladores de motores de tipo universal (LM629, HCTL-1100) o circuitos de interface específicos para el control de motores específicos de C.C., sin escobillas y paso-paso.

• ROBÓTICA:

Manipulación de un brazo de robot y guiado con detección de obstáculos (Ultrasonidos, etc.).

• AUTOMÁTICA:

Control de cualquier tipo de automatismo a partir de

variables analógicas o digitales.

• SISTEMAS DISTRIBUIDOS:

Se puede abordar cualquiera de las Áreas anteriormente citadas, en entornos de grandes dimensiones o Sistemas de elevada complejidad, estableciendo una red de placas μ Controladoras en las que cada una de ellas se encargará bien de una determinada zona de dicho entorno o de una fracción de variables de dicho Sistema.

4 Estudio del sistema por bloques

Se dividirá la arquitectura en bloques, siendo éstos los siguientes:

- Sistema mínimo 8031 en modo microprocesador, sistema de memoria y lógica de selección.
- Sistema de conversión Analógica- Digital, Digital-Analógica y Controlador de Interrupciones.
- Dispositivos I^2C : Teclado, Puertos de E/S y memoria SEEPROM.
- Interfaces de Comunicaciones RS-232 y RS-485.
- Otros dispositivos: LCD, RTC (Reloj en Tiempo Real) y PPI (Interface de Puertos Programable).
- Bus de expansión.

4.1 Sistema mínimo 8031 en modo microprocesador, sistema de memoria y lógica de selección.

El sistema mínimo necesario para que funcione el μC en modo microprocesador se presenta en la Figura 5.

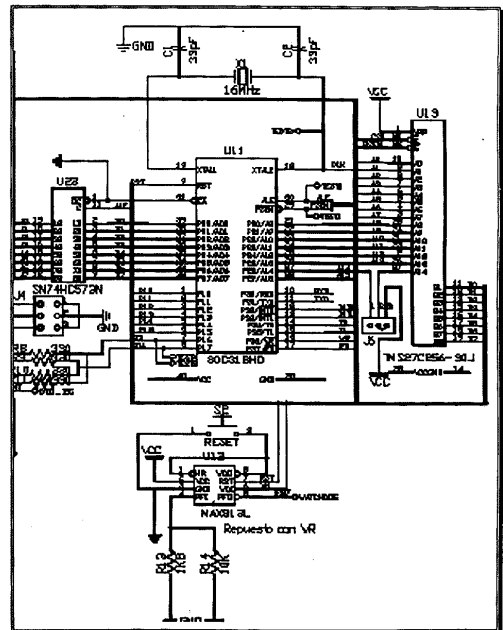


Figura 5

Lo compone el propio microcontrolador (U11), un latch (U23) para expandir el bus de direcciones ya que la parte baja A(0-7) está multiplexada con el bus de datos y la memoria de programa (U13) realizándose el ciclo de Fetch (búsqueda de código de operación) mediante la línea PSEN. Además del sistema mínimo se incorporan en este bloque los siguientes dispositivos:

- Circuito Watch-Dog (Perro Guardián) (U12), generador de reset, supervisor de programa y generador de alimentación.
- Decodificador de direcciones para lógica de selección (U6).
- Pulsadores (S2-S6), leds indicadores de estado (D1-D5) y altavoz (SP1).
- Buffer de corriente (U14) conectado al puerto P1 del μC para excitación de los leds indicadores y altavoz.

4.1.1 Leds indicadores, pulsadores y altavoz

En este bloque Se incluyen dispositivos tales como los pulsadores, el buffer de corriente, los leds y el altavoz.

El buffer de corriente ULN2803 (U14) esta conectado al puerto P1 del microcontrolador para la excitación de los leds indicadores de estado y del altavoz.

Con estos componentes se visualizará el estado actual de proceso del sistema y la posibilidad de actuar sobre él haciendo selecciones de los pulsadores S2-S6, también asociados a líneas de puerto del μC .

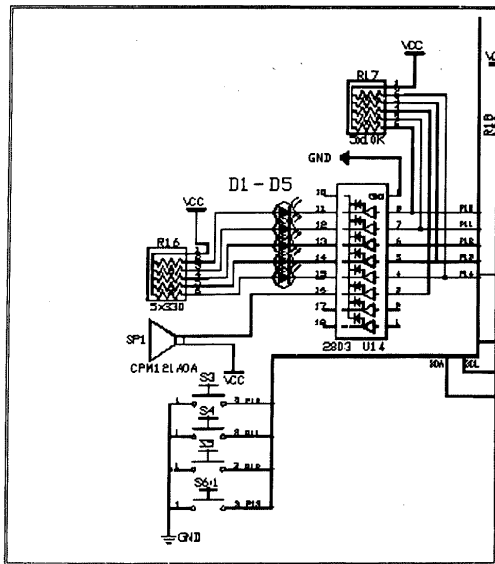


Figura 6

4.1.2 Circuito de Watch-Dog

Se hace necesario incorporar un bloque que genere el

pulso de Reset en el arranque del sistema, pudiendo solucionarse de una forma sencilla mediante una red RC; sin embargo se ha optado por realzar este bloque incorporando un circuito de Watch-Dog (WD) con objeto de asegurar la correcta ejecución del programa de aplicación ante un evento espúreo tal como una interferencia por ruido eléctrico o un fallo de alimentación.

4.1.2.1 Supervisión sobre la ejecución de programa

En ambientes industriales las aplicaciones basada en lógica programada están expuestas a fuertes interferencias por ruido eléctrico, pudiendo producirse una alteración puntual de los registros internos del $\mu\text{P}/\mu\text{C}$; por ejemplo si se produjese un "glitch" (pico de tensión por ruido) que afectara al contador de programa, el μC podría realizar ciclos de Fetch sobre direcciones de memoria de programa que no corresponden a una instrucción quedando el sistema completamente bloqueado, siendo necesario un reset manual.

El Watch-Dog realiza un control del reset del μC respecto a la correcta ejecución del programa, de tal forma que de forma periódica habrá que generar por programa pulsos de refresco sobre el WD que aseguren la no activación de la línea Reset. El circuito empleado es el MAX-813 de Maxim, el cual mantiene el estado de reposo del reset durante 1,6 seg., siendo necesario una reactivación por programa para asegurarse un nuevo intervalo de tiempo de ejecución, caso de que el programa no se ejecutara de forma correcta, este refresco de temporización del reset no se produciría y a los 1,6 seg el sistema se resetearía de forma automática.

El control de refresco del MAX-813 se realiza a través de la línea WDI; el diseño ésta se ha conectado a la línea de escritura WR del μC de tal forma que un ciclo genérico de acceso de escritura asegura un nuevo intervalo de mantenimiento del reset. A la hora de elaborar un programa hay que tener en cuenta este aspecto, si éste contiene una secuencia continua de instrucciones de acceso de escritura sobre dispositivos periféricos, el refresco del WD queda asegurado; sin embargo si esto no es así habrá que generar pulsos de escritura a una dirección no ocupada por un dispositivo en diferentes tramos del programa. En este diseño el rango de direcciones 2500H-7FFFH corresponde al bus de expansión, pudiendo elegirse una dirección en la que no sea conectada una aplicación, por ejemplo la 3000H.

A continuación se presenta un programa que no direcciona dispositivos externos, con lo que habrá que introducir instrucciones de refresco del WD.

```

.....
;Programa: ROTA.A51
;
; PROGRAMA EJEMPLO DE ROTACION DE UN BIT POR EL
; PUERTO P1
; Este programa realiza rotación a izquierda y derecha de un bit
; con carry, con retardo y salida por el puerto P1.
; Se hace necesario generar refresco de un circuito Watch-dog

```

```

;externo (MAX-813)
;..... $MOD51
$LIST
NAME ROTA
=====
;Macro: WD
;Genera un pulso WR para refresco del Watch-Dog
;.....
WD      MACRO
        MOV     DPTR,#3000H
        MOVX    @DPTR,A
        ENDM
;.....
;DEFINICION DEL AREA DE STACK
;.....
STACK   SEGMENT IDATA
        RSEG STACK
        DS 10
;.....
;.....
ROT SEGMENT CODE
        RSEG ROT
;.....
CSEG AT 0

        SETB    PSW.7
        MOV     A,#00
ROTAI:   RLC     A
        JB      PSW.7,ROTAD

        MOV     P1,A
        CALL    RETARDO
        JMP     ROTAI
ROTAD:   RRC     A
        MOV     P1,A
        JB      PSW.7,ROTAI
        CALL    RETARDO
        JMP     ROTAD

;Para realizar el retardo empleamos el Timer 0;en Modo 0.
RETARDO: MOV     TMOD,#00
        CLR     TF0
        MOV     TH0,#80H
        MOV     TL0,#00H
        SETB    TR0
PULSO_WD: WD
        JNB     TF0,PULSO_WD
        RET
END
    
```

el sistema EUPLA'51 esta línea es conectada al nivel de interrupción 5 del controlador de interrupciones (WATCHDOG).

Para evitar este problema puede monitorizarse el estado de la alimentación y una vez detectada una bajada de tensión específica, el sistema puede realizar un almacenamiento en memoria no volátil (EEPROM) del estado del proceso, activando además una bandera indicativa para el modo de arranque del programa, de tal forma que el reset por fallo de alimentación el programa realice un test de esta bandera y caso de estar activada cargar desde la EEPROM el estado de ejecución del proceso.

4.1.3 Mapa de memoria. lógica de selección

La lógica de selección se implementa en una PLD (U6) y se realiza distinguiendo los siguientes tipos de acceso; ciclo de Fetch, ciclo de acceso de lectura y ciclo de acceso de escritura. De esta forma se consigue un alto aprovechamiento en cuanto a la capacidad de direccionamiento del µC.

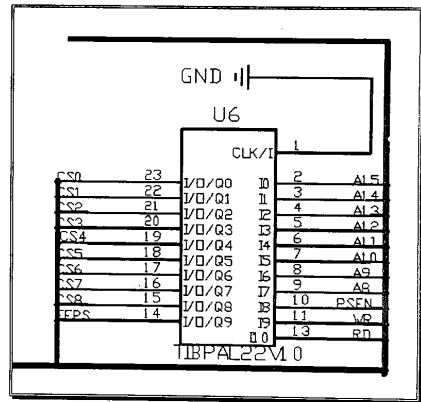


Figura 7

4.1.2.2 Supervisión de la alimentación

Hay sistemas en los que un fallo de alimentación puede producir trastornos importantes en un determinado proceso, por ejemplo considérese un sistema de control de embotellado de líquido, en el que interviene como entradas un sensor de caudal (encoder) y un sensor de presencia de recipiente (barrera infrarroja), y como salidas control de una electroválvula de paso de líquido y control de movimiento de la cadena de recipientes. El sistema una vez detectado un recipiente activa la electroválvula y mediante el sensor de caudal controla el llenado de líquido; si en esta fase se produjera un corte de alimentación, en el restablecimiento de la misma el sistema comenzaría un nuevo llenado habiendo perdido la información del estado del proceso.

El circuito MAX-813 incorpora un entrada de tensión de referencia (PFI) de tal forma que si disminuye por debajo de 1,25 V activa a nivel bajo una línea de salida (PFO) que podría ser conectada a una línea de interrupción. En

4.1.3.1 Ciclo de fetch

La capacidad de direccionamiento en cuanto a memoria de programa es de 64Kb, y este acceso se puede realizar sobre la memoria EPROM (U13) y sobre la memoria EEPROM (U15), direccionados sobre la primera y segunda mitad de la máxima capacidad de direccionamiento respectivamente. El acceso a memoria de programa se realiza a través de la línea PSEN del µC activa a nivel bajo. El mapa de memoria de programa se presenta a continuación en las tablas Tabla1 y Tabla2.

CSX	Dispositivo	Direc. Inicial	Direc. Final	Tamaño (bytes)	PSE N	RD	WR
CS0'	EPROM (U13)	0000 H	7FFF H	32K	0	1	1
CS1'	EEPROM (U15)	8000 H	FFFF H	32K	0	1	1

Tabla 1

En el reset el µC ejecutará el código de la memoria

EPROM, siendo éste el programa monitor de la tarjeta (MON_V7.HEX) encargado de la gestión de comunicación con el PC para recepción de programas de aplicación, arranque del programa de aplicación y rutinas de control de dispositivos.

La memoria EEPROM se destina a contener los programas de aplicación cargados vía serie desde el PC, hay que tener en cuenta que el código a la hora de realizar el linkado habrá que reubicarlo en la dirección base 8000H.

EEPS (PLD)	E ² PROM 32K	FFFF H
	PROG. APLICACION	8000 H
PSEN (μC)	EPROM 32K	7FFF H
	PROGRAMA MONITOR	0000 H

Tabla 2

-Lógica de Selección : Implementada en U6 (PAL 22V10)

- (23) CS0' = A15' * PSEN'
 (14) EEPS' = PSEN' + PSEN RD' (Ciclo de Fetch en EEPROM)
 (22) CS1' = A15
 * N° de Pin de la PLD 22V10

4.1.3.2 Ciclos de acceso de lectura y escritura

Los ciclos de acceso de lectura y escritura se realizarán sobre memoria EEPROM para almacenamiento del programa de aplicación y variables, sobre RAM para almacenamiento de variables externas al μC y sobre el resto de dispositivos que incorpora la arquitectura. A continuación se presenta en la Tabla 3 el mapa de direcciones y las líneas de habilitación CSx de los diferentes dispositivos.

CSX	Dispositivo	Direc. Inicial	Direc. Final	Tamaño (bytes)	PSE N	RD	WR
CS2'	RAM (U13)	0000 H	1FFF H	8 K	1	X	X
CS3'	A/D (U22)	2000 H	2007 H	8	1	X	X
CS4'	D/A (U21)	2100 H	2100 H	1	1	1	X
CS4'	C. DE INT (U20)	2100 H	2100 H	1	1	X	1
CS5'	8255 (U5)	2200 H	2203 H	4	1	X	X
CS6	LCD (LCD)	2300 H	2303 H	4	1	X	X
CS7'	RTC (U17)	2400 H	243F H	64	1	X	X
CS8'	BUS DE EXP.	2500 H	7FFF H	23.295	1	X	X
CS1'	EEPROM (U15)	8000 H	FFFF H	32 K	1	X	X

Tabla 3

4.1.3.3 Decodificador y lógica de selección

La lógica de selección de los diferentes dispositivos que integran la arquitectura se realiza mediante una PLD tipo PAL22V10 (U6).

A continuación se presenta una tabla en la que se incluyen todas las variables lógicas de entrada y salida que intervienen en la lógica de habilitación para los diferentes tipos de acceso vistos anteriormente.

A partir de la Tabla 3 se obtiene la lógica combinacional para todas las variables de salida realizada a partir de U6, programada mediante el fichero EUPLA1.JED en formato JEDEC. A continuación se presenta la lógica combinacional para cada variable de salida y entre paréntesis el número de pin de U6 donde es implementada.

- (14) EEPS' = PSEN' + PSEN * RD'
 (23) CS0' = A15' * PSEN'
 (22) CS1' = A15
 (20) CS3' = A15' * A14' * A13' * A12' * A11' * A10' * A9' * A8' * PSEN
 (19) CS4' = A15' * A14' * A13' * A12' * A11' * A10' * A9' * A8' * PSEN
 (18) CS5' = A15' * A14' * A13' * A12' * A11' * A10' * A9' * A8' * PSEN
 (17) CS6' = A15' * A14' * A13' * A12' * A11' * A10' * A9' * A8' * PSEN
 (16) CS7' = A15' * A14' * A13' * A12' * A11' * A10' * A9' * A8' * PSEN
 (15) CS8' = A15' * A14' * A13' * A12' * A11' * A10' * A9' * A8' * PSEN

TIPO DE CELDA=10(Lógica combinacional activa a nivel Bajo)
 TIPO DE CELDA=11(Lógica combinacional activa a nivel Alto)

4.2 Sistema de conversión analógica-digital, digital-analógica y controlador de interrupciones

4.2.1 Sistema de conversión analógica-digital

El sistema de conversión A/D está compuesto por el convertor A/D ADC0808 (U22) de National Semiconductors, la lógica de interface con el μC implementada en la PLD PAL 22V10 (U20) y un divisor de frecuencia por 16 realizado con el contador 7493AN (U18) con objeto de suministrar la señal de reloj necesaria para el convertor a partir de la frecuencia de oscilador (16 Mhz).

El convertor A/D ADC0808 dispone de 8 canales analógicos de entrada (IN0-IN7) también disponibles en K8 junto con una línea VREF+ que permite aplicar una tensión de referencia externa al convertor.

Las 8 entradas están multiplexadas, teniendo que escoger con anterioridad que entrada se quiere muestrear. Esto es un inconveniente ya que el tiempo de cambiar de entrada y muestreo es distinto al de solo muestrear. Este dispositivo, tiene una velocidad de conversión de 100μs, que equivale a una frecuencia de 10khz, una frecuencia no excesivamente elevada, pero valida para

muestrear señales de 5Khz.

Debido a que el ADC0808 no dispone de línea de habilitación de chip (#CS) se hace necesario para poder ubicarlo dentro de un rango de direcciones, establecer una lógica combinacional respecto a los accesos de lectura y escritura del mismo.

Las líneas de orden de disparo STCNV (Comienzo de Conversión) y LE (Selección de canal) estarán ligadas a un acceso de escritura y la línea OE a un acceso de lectura, estas líneas son activas a nivel alto.

La línea de Fin de Conversión, EOC, también es activa a nivel alto, sin embargo si queremos ligarla al sistema de interrupciones del μC , el cual trabaja con niveles de activación por nivel bajo o flanco de bajada, se hace conveniente el invertirla (NEOC).

En la Figura 8 se presenta el conexionado del ADC.

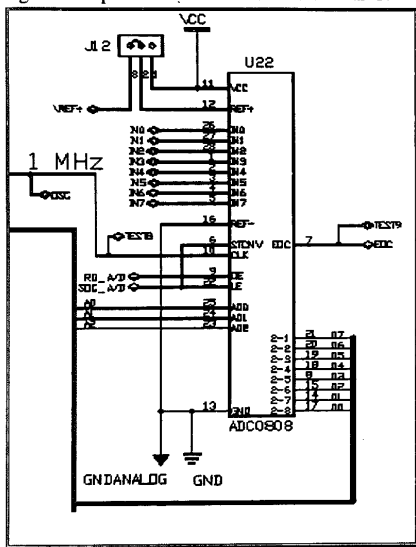


Figura 8

Toda esta lógica combinacional (Figura 9) de interface con el convertor A/D, se implementa en la PLD U20, además de sus funciones como controlador de interrupciones con prioridad.

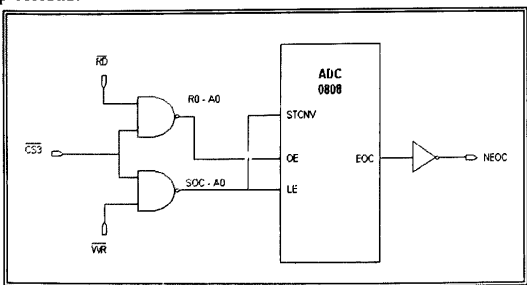


Figura 9

A continuación se exponen en la Figura 10 aquellos bloques de U20 que afectan al módulo de conversión A/D, tanto a

nivel de lógica combinacional como sistema de interrupciones.

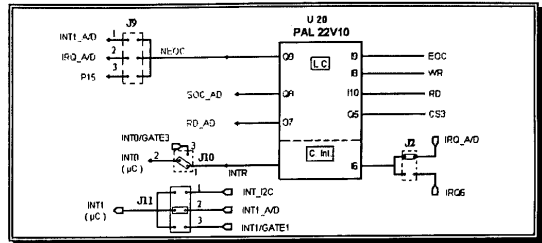


Figura 10

Este dispositivo tiene como objeto el muestreo de señales analógicas y convertirlas en señales digitales procesables por el microcontrolador. El ADC esta mapeado en memoria ocupando 8 bytes, uno por cada canal digital..

En un acceso de escritura, sobre el ADC se realizan dos fases; se selecciona el canal por el cual se quiere muestrear y además se comienza la conversión. La selección del canal depende de la dirección BASE+n a la que se acceda, siendo n el numero de canal de entrada y BASE=0x2000.

En un acceso de lectura, se lee el dato muestreado. Es indiferente sobre que puerto se realice el acceso, ya que se leerá el dato del canal anteriormente fijado.

En la practica, en la realización de un muestreo, hay que realizar un acceso de escritura (para empezar la conversión y seleccionar canal), y otro de lectura(para recoger el dato muestreado), por cada muestra.

El final de la conversión indica la posibilidad de realizar la transferencia del dato ya convertido, y ésta se puede comprobar, por interrupción o por encuesta (polling).

Para realizar transferencias del ADC0808 a nivel de interrupción los Jumpers J2 y J10 se cablearán como en el circuito de la Figura 10; J9 se configurará en función del tipo de transferencia, presentando las siguientes posibilidades.

- **Posición 1: Máxima prioridad de transferencia**
Transferencia por petición de interrupción directa al μC , a través de INT1, para ello J11 debe configurarse tal y como aparece en la Figura 10.
- **Posición 2: Transferencia por petición de interrupción a través del Controlador de Interrupciones (Nivel 6)**
A nivel de μC , el tratamiento de las peticiones del Controlador de Interrupciones se realiza a través de INT0.
- **Posición 3: Transferencia por encuesta a través del estado de P1.5.**
P1.5 = 0 -> FIN DE CONVERSION (Dato disponible).

La lógica combinacional de interface del ADC0808 con el μC es la siguiente:

- (14) NEOC = EOC'
- (15) SOC_AD = WR' * CS3'
- (16) RD_AD = RD' * CS3'

El convertor AD esta instalado en el rango de direcciones 2000-2007H correspondientes a IN0-IN7 respectivamente.

4.2.2 Sistema de conversión digital analógico

Este sistema gira alrededor del convertor digital analógico (DAC) MAX530 (U21) de Maxim (Figura 11), siendo un convertor de 12 bits lo que implica un doble acceso de escritura a la hora de enclavar el código binario correspondiente a un determinado nivel de salida, ya que el bus de datos del μC es de 8 bits.

El DAC tiene una velocidad de conversión de 50 nS; incluye una tensión de referencia interna de 2.048 V con lo que se podrían obtener unos valores de 0 a 4.096 V o de ± 2.048 .

El DAC está direccionado en la dirección 2100H compartiendo dirección de interrupción ya que en el DAC se escribirán datos y en el controlador de interrupciones solamente se leerán.

Este dispositivo se encuentra mapeado en memoria, teniendo un único byte de direccionamiento. Los 12 bits que tiene de resolución se deben acceder de una forma diferente a la habitual, ya que se encuentran multiplexados, teniéndose que acceder según los bits A0 y A1. estos bits seleccionaran a que líneas corresponden los bits d0,d1, d2,d3.

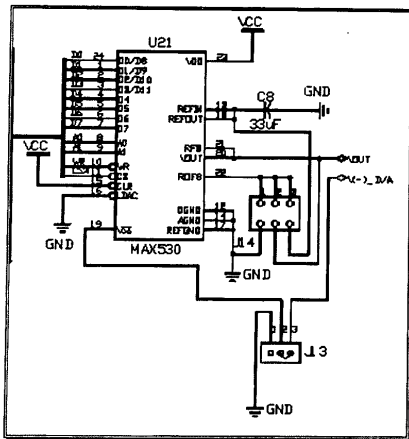


Figura 11

4.2.3 Controlador de interrupciones

El sistema dispone de un controlador de interrupciones con prioridad (Figura 12), el cual se ha considerado interesante incorporar realizando en este aspecto las prestaciones del propio μC . Se dispone de 8 niveles de interrupción (Trazado, IRQ0-IRQ6), de los que 5 pueden ser empleados a nivel interno por la propia arquitectura (IRQ2-IRQ6); la interrupción de trazado se consigue mediante la activación a nivel bajo de la línea P1.3 del μC pudiendo de esta forma generarse una interrupción hardware controlada por programa, esto puede ser muy útil a la hora de realizar un control de ejecución paso a

paso de un programa el cual se desea depurar. La rutina de interrupción asociada tendrá como misión transmitir vía serie los valores actuales de los registros fundamentales y aquellas variables de memoria seleccionadas en la fase de depuración.

Hay que tener presente que siempre que se incorpore el controlador de interrupciones en una determinada aplicación, el estado de la línea P1.3 debe de ser nivel alto, ya que si esto no es así (P1.3=0), se enmascarará cualquier evolución en el estado de las líneas de petición de interrupción IRQ0 a IRQ6.

Tal cual se ha comentado con anterioridad, el sistema dispone de un controlador de interrupciones (CI) de 8 niveles activas por flanco de bajada, donde Trazado, IRQ0 e IRQ6 corresponden a los niveles de interrupción de más alta y más baja prioridad respectivamente. Además los niveles IRQ2 a IRQ6 pueden ser empleados por diversos dispositivos internos o disponibles en el conector K8 para otras aplicaciones. El sistema de interrupciones se realiza sobre la línea INTO del μC , de tal forma que si se activa un determinado nivel de interrupción del controlador CI, realmente el μC lo asocia a la rutina de atención de INTO, debiendo prever en la misma instrucciones de lectura sobre el CI para capturar el número de interrupción.

La identificación del nivel de interrupción se realiza con tres bits, siendo cableados a los bits menos significativos del bus de datos d(0-2). A continuación se muestra una tabla de los niveles de interrupción y su correspondiente número asociado.

El controlador de interrupciones esta situado en la dirección 2100H compartiendo dirección con el DAC, debido a que en el CI solamente vamos a leer y en el DAC solamente vamos a poder escribir.

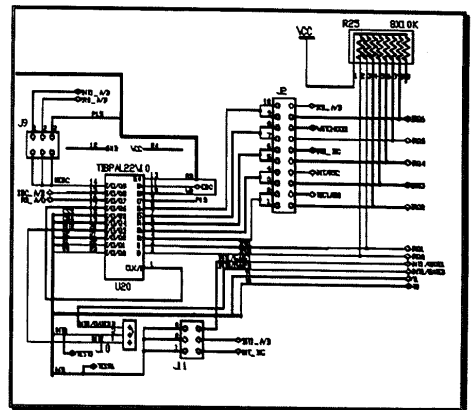


Figura 12

La lógica combinacional correspondiente al controlador de interrupciones con prioridad y la lógica combinacional de interface con el convertidor A/D se implementa en una PAL (U20), programada mediante el fichero EUPLA2.JED en formato JEDEC. A continuación se presenta la lógica combinacional para cada variable de salida y entre paréntesis el número de

pin de U20 donde es implementada.

NIVEL	INTERNO	EXTERNO	NÚMERO: d2 d1 d0
0	Trazado P1.3	-	0 0 0
1	-	IRQ0	0 0 1
2	-	IRQ1	0 1 0
3	TECLADO	IRQ2	0 1 1
4	RTC	IRQ3	1 0 0
5	I2C	IRQ4	1 0 1
6	WATCH-DOG	IRQ5	1 1 0
7	ADC	IRQ6	1 1 1

Tabla 4

- $D0' = 11'13'15'17 + 12'13'15'17 + 14'15'17 + 16'17$
- $D1' = 12'13'16'17 + 14'16'17 + 15'16'17$
- $D2' = 14'15'16'17$
- $INTR = 17'16'15'14'13'12'11 * 18$
- $INTR' = 17'16'15'14'13'12'11 * 18$
- $RD_AD = RD' * CS3'$
- $SOC_AD = WR' * CS3'$
- $NEOC = EOC'$

TIPO DE CELDA=00 registro salida activa baja
 TIPO DE CELDA=10 registro salida activa alta
 TIPO DE CELDA=01 salida combinacional activa baja
 TIPO DE CELDA=11 salida combinacional activa alta

4.3 Dispositivos I²C: Teclado, Puertos de E/S y memoria SEEPR0M

4.3.1 Puertos de E/S y memoria SEEPR0M

El sistema incorpora dispositivos en bus I2C; dos C.I. PCF8574 (U8, U19) con 8 líneas de entrada/salida direccionables a nivel de bit y una memoria SEEPR0M (U7) de 512 bytes Figuras 13 y 14.

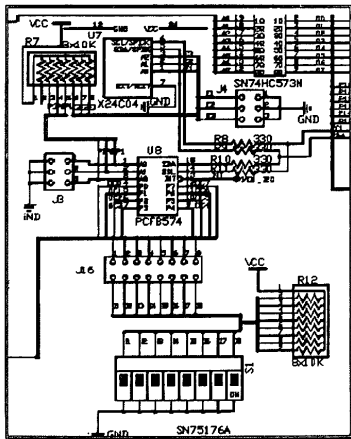


Figura 13

U8 se emplea como líneas de entrada/salida de carácter general disponibles en el conector K2 y a través de los jumpers J16 se conectan a 8 interruptores (S1) con objeto de asignar un nivel lógico a cada línea del puerto; una posible aplicación sería la de asignar mediante S1 una dirección de módulo cuando la tarjeta trabaje en red formando un sistema distribuido multiprocesador. U19 actúa como interfase de una matriz de pulsadores hexadecimal, realizando a través del mismo el rastreo de teclado.

Una de las ventajas que presenta el C.I. PCF8574 para esta aplicación es que cuando el estado de una línea de puerto configurada como entrada varía su estado, la línea INT del mismo pasa a nivel Bajo; si ésta es conectada al controlador de interrupciones (TECLADO, nivel 2), puede realizarse una atención del teclado únicamente cuando sea pulsada una tecla.

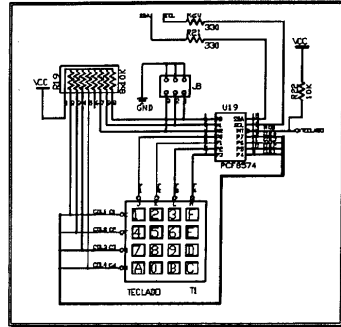


Figura 14

En el caso de U8 la línea de INT (I/O1_I2C) se conecta en el conector K4 a la línea de interrupción del bus I2C; una de las ventajas del bus I2C reside en que las líneas INT de diferentes dispositivos pueden conectarse al ser del tipo colector abierto, pudiendo tener una línea común de petición de interrupción I2C. La línea I/O1_I2C se conecta a través del jumper J1 al controlador de interrupciones (IRQ_I2C, nivel 4) o a través de J11 a la interrupción INT1 del μC .

El C.I. U7 puede ser empleada como memoria no volátil para almacenamiento de variables, una posible aplicación sería el que emulara una tarjeta chip con información para control de accesos, etc.

Los jumpers J3, J4 y J8 permiten asignar la dirección de esclavo a los dispositivos I2C U8, U7 y U19 respectivamente.

4.4 LCD de 2 líneas x 16 caracteres

Se ha incorporado como visualizador el LCD (Liquid Crystal Display) LM016L de Hitachi cuyo controlador interno es el LSI HD44780. Las características principales de este controlador son:

- Interface programable de 8 o 4 bits.
- RAM de almacenamiento de hasta 80 caracteres, DD RAM (RAM de Datos del Display).
- Generador de caracteres en ROM (CG ROM), con 192 caracteres pudiendo definirse otros ocho en RAM.
- 160 caracteres de 5 x 7 puntos.
- 32 caracteres de 5 x 10 puntos.
- Acceso desde programa tanto a la Ram de caracteres como a la ROM (generador de caracteres).
- Amplio rango de funciones: Borra Display, Retorno de cursor, Display ON/OFF, Desplazamiento de cursor, etc.
- Circuito de reset interno.
- Control de contraste.

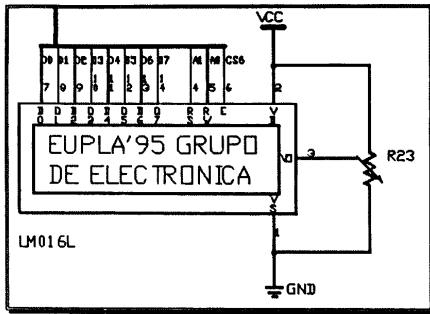


Figura 15

- Registro de control de configuración de puertos y modo de operación.
- 3 posibles modos de operación: E/S básica, E/S controlada y Bus bidireccional.

Dentro del mapa de memoria de la arquitectura este dispositivo se ha ubicado a partir de la dirección base 2200H ocupando a partir de aquí cuatro bytes según la Tabla 5

DIRECCIÓN	A1	A0	OPERACIÓN
2200 H	0	0	PUERTO A
2201 H	0	1	PUERTO B
2202 H	1	0	PUERTO C
2203 H	1	1	REGISTRO DE CONTROL

Tabla 5

4.5 RTC (RELOJ EN TIEMPO REAL)

El C.I. U17 es el DS-1286 de Dallas Semiconductor Figura 16, se trata de un reloj en tiempo real (RTC) cuyas características son las siguientes:

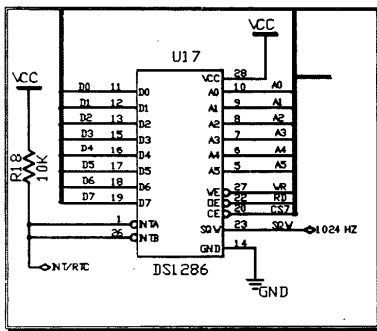


Figura 16

- Temporización de centésimas de segundo, segundos, minutos, horas, días, días del mes, meses y años.
- Funciones de alarma con dos líneas de interrupción de salida en colector abierto.
- Interrupciones de salida programables.
- Generación de una señal cuadrada de salida de 1024 Hz.
- Todos los registros pueden ser direccionados de forma aleatoria.
- Pila de litio integrada manteniendo el RTC en funcionamiento durante 10 años.
- Bit interno de bajo consumo para ahorro de energía.
- 50 bytes de memoria RAM no volátil.

El RTC está direccionado en el intervalo de direcciones 2400 H a 243F H, en total 14 bytes de funciones de reloj-calendario y 50 bytes de memoria RAM.

4.6 PPI (INTERFACE DE PUERTOS PROGRAMABLE)

Como puertos de Entrada/Salida se ha añadido un 82C55 (U5), cuyas características principales son:

- 3 Puertos de configurables como entrada o salida : A, B y C.
- El puerto C conforma dos puertos independientes de 4 bits, pudiendo programarse de forma separada.

4.7 Interfaces de comunicación RS-232 y RS-485

Los circuitos integrados U9 y U10 constituyen los interfaces de comunicación serie según normas RS-485 y RS-232 respectivamente. Para el interface RS-232 se ha elegido el MAX-232 de Maxim (U9) que incorpora ya que permite adaptar los niveles TTL de las líneas TXD y RXD de μC a los niveles lógicos según norma RS-232 con una única alimentación a +5 V.

El interface RS-485 permite interconectar hasta 32 emisores-receptores a una distancia máxima de 1200 m mediante un par diferencial (2 hilos); se ha incorporado como interface el C.I. 75176 de Texas Instruments consiguiendo así el poder establecer un sistema distribuido multiprocesador. Para habilitar la transmisión en modo RS-485 la línea P1.4 del μC debe activarse a nivel alto.

Para poder seleccionar uno u otro interface se ha añadido un 4053 (U25), se trata de un conmutador analógico triple, de tal forma que en función del nivel de P1.5, (Tabla 6) las líneas TXD y RXD del μC son cableadas al interface de comunicación serie seleccionado.

INTERFACE SERIE	ESTADO P1.5
RS-485	0
RS-232	1

Tabla 6

4.8 Bus de expansión

Se ha incorporado un conector (K1) con objeto de poder expandir el sistema conectando nuevos dispositivos, el bus de expansión incorpora las siguientes líneas:

- Línea de reset.
- Líneas de control, interrupciones, timers, bus de direcciones y datos del μC .
- Líneas I2C.
- Líneas del controlador de interrupciones IRQ0-IRQ3.
- Canales analógicos IN0-IN3 y línea de tensión de referencia externa (Vref+) del ADC0808.
- Línea digital/analógica Vout de U21 (MAX-530).
- Alimentación.

Todas las líneas del μC así como la de reset están

amplificadas mediante buffers con objeto de no cargar en exceso al μC , el bus de datos incorpora un buffer bidireccional (UI, Transceiver).

Se ha asignado una dirección base (2500 H) para habilitación del bus de expansión, disponiendo a partir de ésta de 23.296 bytes para direccionar aplicaciones.

5 KERNEL DE TIEMPO REAL

Se ha implementado un kernel de tiempo real para la familia de microcontroladores MCS-51. Pudiendo realizarse aplicaciones tanto en lenguaje C como en ensamblador.

La ejecución en serie de programas provoca que en algún momento los dispositivos de E/S o el procesador estén desocupados, incluso cuando la corriente de trabajo de entrada no esté nunca vacía. Un cambio para esto sería asignar algún otro trabajo al procesador y a los dispositivos de E/S cuando de otra manera estarían desocupados. Al ser asignados a componentes infrautilizados amplios surtidos de trabajo pendiente, se puede proporcionar que se ejecuten los programas pendientes concurrentemente.

Se usan los S.O. en tiempo real en entornos donde se deben aceptar y procesar en tiempo breve y sin tiempos muertos un gran número de sucesos, en su mayoría externos al sistema de computadora. Ejemplos de tales aplicaciones serían control industrial, equipamiento telefónico conmutado, control de vuelo y simulaciones en tiempo real.

Proporcionar tiempos rápidos de respuesta y así hacer frente a los tiempos muertos de planificación es un objetivo primario de los sistemas en tiempo real.

4.-BIBLIOGRAFÍA.

❖ MANUALES DE REFERENCIA

- *"8-BIT EMBEDDED CONTROLLERS"*, Intel Corp. 1990.
- *"MICROCONTROLLER COMPONENT 80515 "*, Siemens Corp. 1985.
- *"EIGHT-BIT 80C51 EMBEDDED PROCESSORS"*, AMD Corp. 1990.
- *"MICROCONTROLLER USER'S GUIDE"*, Signetics Corp. 1989.
- *"MICROCONTROLLER HANDBOOK"*, Intel Corp. 1984.
- *"SOFT MICROCONTROLLER"*, Dallas Semiconductor. 1993.
- *"TIMEKEEPING & NV RAM DATA BOOK"*. 1994-1995.
- *"PROGRAMMABLE LOGIC DEVICES (PLD)"*, Philips Semiconductor 1991.
- *"C51 DEVELOPERS KIT"*, Franklin Software, Inc USA. 1991
- *"EMUL51 HANDBOOK"*, Nohau Corporation 1993
- *"EXPRO-60 HANDBOOK"*, Sunshine Device Programmer.
- *"ASSEMBLER X8051"*. 2500 AD Software
- *"I2C BUS SPECIFICATION"*. Philips. 1991.
- *"I2C SPECIFICATIONS SIGNETICS"*, Signetics linear products. 1992.

❖ TEXTOS

- **YERELAN-AHLUWALIA,** *"PROGRAMING AND INTERFACING THE 8051 MICROCONTROLLER"*, Addison- Wesley Company 1995
- **D.PARET.,** *"EL BUS I2C. DE LA TEORÍA A LA PRACTICA"*. Paraninfo 1995.
- **GONZALEZ VAZQUEZ.,** *"INTRODUCCION A LOS MICROCONTROLADORES"*, Mc Graw Hill 1992
- **MARTINEZ-BARRON.,** *"PRACTICAS CON MICROCONTROLADORES DE 8 BITS"* Mc Graw Hill 1993
- **GARCÍA GUERRA, A.,** *"SISTEMAS DIGITALES. INGENIERÍA DE LOS MICROPROCESADORES"*. E.T.S. Ingenieros de Telecomunicación. Madrid 1986.
- **GARCÍA PIE, A.,** *"MICROPROCESADORES. ESTRUCTURA Y PROGRAMACIÓN"*. Rede. Barcelona 1987.
- **LANCETA, A Y PECIÑA, L,** *"MICROCONTROLADORES INDUSTRIALES MCS-51"*. Edebé. Barcelona 1994.
- **MANO, M.M.,** *"COMPUTER SYTEM ARCHITECTURE"*. Prentice-Hall. New Jersey 1976.
- **MCS-51,** *"USER'S MANUAL"*. Intel Corporation. Santa Clara 1985.
- **MCS-85,** *"USER'S MANUAL"*. Intel Corporation. Santa Clara 1978.
- **MUNDO ELECTRÓNICO,** *"INTERCONEXIÓN DE PERIFÉRICOS A MICROPROCESADORES"*. Marcombo. Barcelona 1987.
- **SHORT, K.L.,** *"MICROPROCESADORES Y LÓGICA PROGRAMADA"*. Gustavo Gili. Barcelona 1985.
- **TANENBAUM, A.S.,** *"ORGANIZACIÓN ESTRUCTURADA DE COMPUTADORES"*. Prentice-Hall. New Jersey 1976.