

300617



UNIVERSIDAD LA SALLE 3

ESCUELA DE INGENIERIA
INCORPORADA A LA U.N.A.M.

2ej

MICROCONTROLADORES:

***“DESARROLLO DE UN SISTEMA DE
ADQUISICION DE DATOS CON BASE EN UN
MICROCONTROLADOR (8751)***

TESIS PROFESIONAL

QUE PARA OBTENER EL TITULO DE:

INGENIERO MECANICO ELECTRICISTA

P R E S E N T A N :

MARCELA ALDAPE ALONSO
JOSE RAMON GARCIA ALVAREZ
MIGUEL ANGEL MOLINERO TORRES
FAUSTINO ALBERTO RIVAS ORTIZ
ALEJANDRO SANCHEZ SABBAGH

ASESOR DE TESIS; ING. GUILLERMO ARANDA PEREZ

MEXICO, D. F.

1994

**TESIS CON
FALLA DE ORIGEN**



UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso

DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis está protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.



LA SALLE

A los Pasantes Señores:

Marcela Aldape Alonso
José Ramón García Álvarez
Miguel Ángel Molinero Torres
Faustino Alberto Rivas Ortiz
Alejandro Sánchez Sabbagh

En atención a su solicitud relativa, me es grato transcribir a Ud. a continuación, el tema que aprobado por esta Dirección, propuso como Asesor de Tesis el Ing. Guillermo Aranda Pérez, para que lo desarrolle como tesis en su Examen Profesional de Ingeniero Mecánico Electricista con área principal en Ingeniería Electrónica.

"MICROCONTROLADORES: DESARROLLO DE UN SISTEMA DE ADQUISICION DE DATOS CON BASE EN UN MICROCONTROLADOR (8751)"


con el siguiente índice:

CAPITULO I	INTRODUCCION DESCRIPCION GENERAL Y ESPECIFICACIONES DEL SISTEMA DE ADQUISICION DE DATOS
CAPITULO II	EL MICROCONTROLADOR 8751
CAPITULO III	DISEÑO DEL SISTEMA
CAPITULO IV	INTERFAZ HOMBRE-MAQUINA
	CONCLUSIONES
	APENDICES
	BIBLIOGRAFIA

Ruego a Ud., tomar debida nota de que en cumplimiento de lo especificado en la Ley de Profesiones, deberá prestar Servicio Social como requisito indispensable para sustentar Examen Profesional, así como de la disposición de la Dirección General de Servicios Escolares, en el sentido de que se imprima en lugar visible de los ejemplares de la tesis, el título del trabajo realizado.

A T E N T A M E N T E
"INDIVISA MANENT"
ESCUELA DE INGENIERIA
México, D.F., a 9 de Mayo de 1994


ING. GUILLERMO ARANDA PEREZ
ASESOR DE TESIS


ING. EDUARDO BARRERA MONTOYA
D I R E C T O R

UNIVERSIDAD LA SALLE

BENJAMIN FRANKLIN 47, TEL. 516-99-60 MEXICO 06140, D.F.

INDICE

INDICE

PAGINA

INTRODUCCION.....	i
-------------------	---

CAPITULO 1

DESCRIPCION GENERAL Y ESPECIFICACIONES DEL SISTEMA DE ADQUISICION DE DATOS.

1.1) Antecedentes.....	2
1.2) Sistema de adquisición propuesto.....	6
1.3) Diagrama de bloques general.....	7
1.4) Descripción de la operación.....	8

CAPITULO 2

EL MICROCONTROLADOR 8751

2.1) Selección del microcontrolador.....	11
2.2) Arquitectura interna.....	16
2.3) Registros internos.....	21
2.4) Memoria ROM y RAM.....	31
2.5) Manejo de interrupciones vectoriales.....	32
2.6) Comunicación serial.....	34
2.7) Juego de instrucciones.....	35

CAPITULO 3

DISEÑO DEL SISTEMA

3.1)	Especificaciones del sistema.....	46
3.1.1)	Descripción del sistema.....	46
3.1.2)	Diagrama de bloques específico.....	48
3.2)	Requerimientos de hardware.....	48
3.2.1)	Configuración del 8751.....	48
3.2.2)	Convertidor analógico digital de 8 bits con 8 canales multiplexados ADC0808.....	54
3.2.3)	Transmisión serial RS-232C.....	60
3.2.4)	Buffer 74LS244.....	63
3.2.5)	Selección del cristal de cuarzo.....	65
3.3)	Desarrollo de software.....	71

CAPITULO 4

INTERFAZ HOMBRE-MAQUINA

4.1)	Especificaciones.....	96
4.2)	Descripción de la operación.....	97

CONCLUSIONES.....	102
--------------------------	------------

APENDICE 1

Programa final.....	105
---------------------	-----

APENDICE 2

Diagrama final del circuito.....	115
----------------------------------	-----

APENDICE 3

Simulador AVSIM51.....	117
------------------------	-----

BIBLIOGRAFIA.....	128
--------------------------	------------

INTRODUCCION

INTRODUCCION

El propósito de este trabajo es el diseño de un Sistema de Adquisición de Datos con base en un Microcontrolador 8751. Dicho sistema consiste en muestrear un número determinado de variables físicas que intervienen en un proceso. En otras palabras, se podrán leer diferentes variables tanto analógicas como digitales de algún proceso y desplegar dichas lecturas de manera gráfica en el monitor de una computadora personal.

Para llevar a cabo el desarrollo de esta propuesta se utilizan las características del microcontrolador como núcleo de este sistema, por sus diversas ventajas y su facilidad de manejo. Como requisito indispensable se debe contar con un Convertidor Analógico Digital (ADC), pues es necesario convertir los valores de las variables analógicas en valores digitales.

El microcontrolador se encargará de administrar los datos que provienen del proceso y de igual forma mantener la comunicación con la computadora personal a través de su puerto RS-232C, con el objeto de que el operador de la computadora pueda recibir dicha información y manipularla en base a ciertos requerimientos.

Este trabajo se encuentra organizado de la siguiente manera:

En el capítulo 1 se plantean antecedentes y problemáticas que se presentaron para elaborar el sistema de adquisición de datos, así como una descripción general del proyecto a desarrollar junto con un diagrama esquemático general.

El capítulo 2 consta de una descripción del microcontrolador 8751, incluyendo su arquitectura, diagramas de bloques y de conexiones, sus capacidades, así como sus registros internos y juego de instrucciones.

En el capítulo 3 se lleva a cabo el planteamiento específico y el desarrollo del proyecto, tanto en sus justificaciones como en especificaciones y requerimientos de "hardware" y "software". En este mismo capítulo se lleva a cabo la elaboración del programa requerido para hacer trabajar al microcontrolador como es necesario para cumplir con los planteamientos, mismo que se muestra en su totalidad en el Apéndice 1 del presente trabajo.

En el capítulo 4 se habla sobre las características y manejo de la interfaz hombre-máquina, es decir, el planteamiento sobre como funcionaría la parte del usuario en la computadora y su relación con la adquisición de datos.

El diagrama final del circuito se encuentra en el Apéndice 2.

Como herramienta de apoyo para el diseño de este proyecto se emplea el simulador AVSIM51, el cual es un paquete de software que se utiliza en una computadora para la simulación de la ejecución de un programa. Este paquete de software cuenta con un compilador que hace la conversión pertinente al lenguaje utilizado por el simulador, lo que hace posible que el programa se escriba en cualquier editor de texto. En el Apéndice 3 se hace una breve descripción de la operación y funcionamiento de dicho simulador.

Finalmente se encuentran las conclusiones.

CAPITULO 1

DESCRIPCION GENERAL Y ESPECIFICACIONES DEL SISTEMA DE ADQUISICION DE DATOS

1.1 ANTECEDENTES

COMPUTADORAS EN CONTROL DE PROCESOS

Las computadoras se han convertido en fundamentales para los procesos industriales. Tal es el caso de las computadoras empleadas como controladores programables que actúan como controladores de proceso, o como las computadoras para los sistemas de supervisión y adquisición de datos (DAS, "Data Acquisition System").

Este avance tecnológico ha podido brindar una gran velocidad en cuanto a captura de información, evaluación y almacenamiento de información, tal es el caso de los sistemas de adquisición de datos que han hecho posible monitorear datos en forma continua, automática y poder realizar cálculos sobre los mismos. Todo esto da como resultado una gran eficiencia en control, excelentes rangos de producción y muchos otros beneficios.

Actualmente, si se desea supervisar un proceso, se requiere de un sistema de adquisición de datos. Este sistema está formado por una computadora que almacena y despliega la información del proceso, convertidores A/D (Analógico/Digital) y acondicionadores de señal.

A diferencia de un controlador, un sistema de adquisición sólo recibe información del proceso y no la envía.

Un sistema de adquisición moderno está basado en una computadora y con frecuencia se diseña y construye en una sola tarjeta de circuito impreso. Normalmente, una tarjeta de este tipo, aparte de tener entradas para las variables del proceso también tiene salidas hacia él, lo que le permite controlar lazos pequeños.

En un sistema de este tipo el elemento principal es el microprocesador ó microcontrolador, el cual realiza todas las operaciones con los datos en el momento de la ejecución. El programa del controlador está contenido en la memoria permanente, que generalmente es la memoria de estado sólido y únicamente de lectura (ROM, Read Only Memory). Los almacenamientos temporales que se llevan a cabo durante los cálculos y otras funciones del programa utilizan la memoria de acceso aleatorio (RAM, Random Access Memory). En esta memoria RAM se pueden tanto escribir como leer los datos, la información procesada permanece en ésta siempre y cuando no se suspenda el suministro de energía eléctrica. Este tipo de sistema se diseña para manejar señales de entrada y salida estándar (4-20 mA, 0-10 V, etc).

Las computadoras personales se pueden utilizar como sistemas de supervisión y/o control de procesos. En general, una computadora personal para esta aplicación necesita "hardware" especial en forma de tarjetas de Entrada/Salida (E/S) y "software" consistente en programas para el manejo de E/S de datos y los modos de operación del mismo.

Para adquirir el valor de una variable analógica se requiere de un convertidor A/D.

El dato a la salida de un convertidor A/D es la representación binaria en formato de n-bits del valor analógico. Al tener una resolución finita, el valor analógico que se representa mediante los 2^n valores posibles pueden tener un error de conversión. Por ejemplo al utilizar un convertidor A/D de 8 bits se tendrá un error de conversión mayor que al utilizar uno de 16 bits, ya que la resolución del primero es de $1/2^8$ y la del segundo es de $1/2^{16}$.

El sistema de adquisición por computadora toma únicamente muestreos periódicos del valor de la variable, ocasionando que se ignore el valor o cualquier variación de la misma entre muestreos. Para que dicho sistema funcione correctamente, se deben suponer ciertas condiciones sobre las variaciones entre muestreos.

Existe un número máximo de muestreos por unidad de tiempo en cualquier sistema y está definido por el tiempo que se requiere para tomar una muestra (el tiempo de conversión A/D según sea el caso), más el tiempo que se requiere para procesar la información.

También existe un número mínimo de muestreos por unidad de tiempo en cualquier proceso que depende de la naturaleza con que la variable muestreada varía con respecto al tiempo. Los muestreos deben ser tomados a una frecuencia grande para que la señal se pueda reconstruir lo mejor posible. Existen serias consecuencias al muestrear a una frecuencia demasiado baja ya que, en el caso de un controlador, será incapaz de corregir las variaciones que fueron omitidas.

Cuando una computadora está dedicada a la supervisión y/o control, no se puede utilizar para realizar otras tareas, pero se puede ligar a una red de trabajo para obtener comunicación con otras computadoras y terminales remotas a través de un puerto de comunicaciones. Una red de área local (LAN, Local Area Network) se utiliza para enlazar una red de computadoras de manera que todos los datos y los comandos de operación se puedan intercambiar desde cualquier estación de trabajo. Cada computadora opera de acuerdo a su propio programa y realiza sus obligaciones asignadas. El "software" que opera la red es residente, de manera que permite que todas las computadoras conectadas hagan y reciban llamadas para mantener actualizado el sistema de supervisión y/o control.

Las computadoras "Mainframe" generalmente significan facilidad de cómputo a gran escala, requieren de una instalación especial, dan atención a muchos usuarios y tienen alta velocidad de cómputo. Tales computadoras se emplean para la supervisión total de los procesos de una planta y para realizar las evaluaciones financieras, estudios de ingeniería, control de inventarios y algunas otras actividades de la administración.

Todo sistema de supervisión tiene alarmas. La alarma es una señal que se genera para notificar que un parámetro ha excedido sus límites preestablecidos. A menudo, una alarma opera dando una señal tanto auditiva como visual para atraer la atención hacia dicha situación. Inclusive, la computadora debe tener un grupo de operaciones programadas para ejecutarse bajo una condición de alarma.

Existen muchas ventajas al utilizar una computadora como un sistema de supervisión. Por ejemplo, una computadora puede manejar una gran cantidad de variables, los datos son menos susceptibles a sufrir errores a causa de la inducción del ruido, etc.

Es importante recalcar la importancia que las computadoras han ganado en el terreno de la adquisición de datos y supervisión de procesos, ya que son un elemento indispensable para la implementación de dichos sistemas. Así mismo se debe recordar que éste es el primer paso para implantar un sistema de control.

El mencionado sistema es de gran beneficio tanto para el usuario final como para el mismo proceso, ya que se disminuye el tiempo empleado para la colección de datos, así como el tiempo de respuesta para la corrección de alguna desviación detectada.

Este proyecto se basa única y exclusivamente en un sistema de supervisión, no se considera la posibilidad de ejercer acciones de control.

1.2 SISTEMA DE ADQUISICION PROPUESTO

La presente tesis propone un sistema de adquisición de datos con base en un microcontrolador 8751 y una computadora personal. Las características de este sistema son las siguientes:

- 1) Ocho canales de entrada de señales analógicas (0 - 5 volts).
- 2) Hasta 16 canales digitales de entrada.
- 3) La comunicación con la computadora será a través del puerto serial RS-232C.
- 4) Periodo de Muestreo de los canales analógicos y/o digitales configurable desde 1 hasta 15 segundos.

El presente trabajo no está diseñado para alguna aplicación específica, por lo cual no se considera ninguna etapa de acondicionamiento de señales, ya que esta dependerá de las señales que lo requirieran.

1.3 DIAGRAMA DE BLOQUES GENERAL

El diagrama de bloques del sistema se observa en el siguiente diagrama :

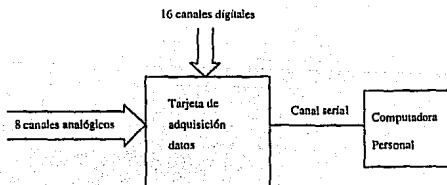


Figura 1.1 Diagrama de Bloques General.

La tarjeta de adquisición de datos es un elemento basado en un microcontrolador que permite el muestreo y la manipulación de variables tanto analógicas como digitales. También se encarga de la comunicación con una computadora personal en un esquema maestro-esclavo, donde el maestro es la computadora y el esclavo es la tarjeta de adquisición.

1.4 DESCRIPCIÓN DE LA OPERACION

El funcionamiento del sistema es el siguiente:

- 1) La tarjeta de adquisición muestrea las señales analógicas y digitales (lógicas) a un periodo de muestreo constante configurable desde la computadora personal. El periodo de muestreo puede ser desde 1 hasta 15 segundos para todos los canales de entrada. La operación de muestreo se realizará en forma continua hasta que la computadora indique, a través de un mensaje, otra operación.
- 2) La tarjeta estará esperando (mediante interrupción) algún mensaje de la computadora personal. Al recibir el mensaje, lo decodificará y realizará la operación indicada. Los mensajes y las operaciones que la computadora puede enviar a la tarjeta de adquisición de datos pueden ser:

Mensaje	Función
4CH	Envío de datos a la PC
5XH	Cambio de período de muestreo
47H	Sólo lectura de variables lógicas
3XH	Muestreo X líneas analógicas
6XH	Muestreo X líneas analógicas y lógicas

- 3) La computadora recibe los datos muestreados y los presenta al usuario en forma de gráficas de tendencia en el caso de variables analógicas, y en el caso de variables lógicas en forma de gráfica de barras. El operador podrá seleccionar una de las gráficas de tendencia para ocupar toda la pantalla (zoom). La información que se presente en la pantalla se podrá almacenar en algún archivo.
- 4) La computadora podrá enviar cualquier mensaje en cualquier momento .
- 5) La tarjeta no podrá comunicarse con la computadora por sí sola. Sólo podrá responder a los mensajes que la computadora le envíe.

CAPITULO 2

EL MICROCONTROLADOR 8751.

2.1 SELECCION DEL MICROCONTROLADOR.

Para la elaboración de este proyecto es posible utilizar varios microcontroladores, por lo cual se analizarán algunos de ellos, así como las herramientas de desarrollo disponibles y finalmente la selección del mismo.

A continuación, se presenta un breve resumen de las características principales de algunos de los modelos de tres de las compañías más importantes en el mercado de los microcontroladores, como son: Motorola, Texas Instruments e Intel.

2.1.1 Análisis Comparativo.

a) Microcontroladores Motorola.

Motorola es una de las empresas pioneras en el campo de los microcontroladores de 8 bits, éstos están constituidos por una gran variedad de modelos y familias.

Con el objeto de realizar un análisis comparativo entre las diferentes marcas de microcontroladores se selecciona la familia 68HC11, ya que este circuito reúne las características necesarias para esta aplicación.

El 68HC11, es un circuito que cuenta con una ALU de 8 bits y gran versatilidad para el programador. Tiene una velocidad máxima de reloj de 2 MHz, y su estructura estática permite el uso de frecuencias extremadamente bajas. Este circuito puede tener en forma interna hasta 8 KB de memoria ROM, 512 Bytes de EPROM y 256 Bytes de memoria RAM. Además cuenta con los siguientes circuitos periféricos:

- Una interfaz de comunicaciones asíncrona.
- Contadores de 16 bits.
- Interrupciones de tiempo real.
- Convertidor analógico/digital con resolución de 8 bits.

b) Microcontroladores Texas Instruments.

Los microcontroladores de Texas Instruments cuentan con un CPU, memoria (ROM, RAM, EPROM), puertos de entrada/salida, contadores e interrupciones en un solo circuito.

Para efecto del análisis comparativo se selecciona la familia TMS7000, la cual cuenta con las siguientes características:

- Puertos mapeados en memoria para facilitar el direccionamiento.
- Ocho formas de direccionamiento.
- Dos interrupciones externas.
- Amplio margen de frecuencias y voltajes de operación.
+ 2.5 a 6 Volts, + 0.8 a 6.5 MHz
- Arquitectura de registro a registro.

c) Microcontroladores Intel.

La familia MCS-51 es la seleccionada ya que cuenta con un juego de instrucciones amplio, así como con un número suficiente de temporizadores y fuentes de interrupción. Entre sus principales características se encuentran las siguientes:

- CPU de 8 bits.
- Circuitería de oscilador y reloj en el mismo circuito.
- Líneas de entrada/salida bidireccionales.
- Contadores de 16 bits.
- Fuentes de interrupción con 2 niveles de prioridad.
- Puerto serial "full duplex".

En la tabla que se muestra a continuación se muestran los resultados del análisis comparativo entre los tres microcontroladores antes mencionados.

CARACTERISTICAS	68HC711D3	TMS77C82	8751
MEMORIA DE PROGRAMA (EPROM) INTERNA	4 KB	8 K	4 K
MEMORIA DE DATOS RAM INTERNA	196 B	256 B	128 B
LINEAS DE E/S	30	24	36
MANEJO DE MEMORIA EXTERNA	SI	SI	SI
BUS DE DATOS Y DIRECCIONES MULTIPLEXADO	SI	SI	SI
CONTADORES DE TIEMPO Y EVENTOS	3	3	2
INTERRUPCIONES EXTERNAS	1	2	2

2.1.2 Herramientas de Desarrollo.

Este es un punto muy importante para el desarrollo de la aplicación, ya que a través de estas herramientas (Software y Hardware), se facilita la generación de código y se permite una eficiente detección de errores.

Con la utilización de estas herramientas se reduce el tiempo de desarrollo ya que con ellas es posible realizar simulaciones en tiempo real de las actividades del microcontrolador y hacer las correcciones pertinentes, evitando así la posible reprogramación.

En este caso en particular, se cuenta con los siguientes productos:

- AVLINK-51. Es un ensamblador de la marca Avocet, que compila el código en ensamblador convirtiéndolo en código de máquina. Este producto facilita la programación, ya que además de permitir que el programa sea escrito en ensamblador, cuenta con un gran número de directivas para la simplificación del mismo.

- AVSIM51. Es un programa para PC's compatibles, que simula instrucción por instrucción un programa. Permite al usuario observar el contenido de los registros de CPU en todo momento.

Ambas herramientas están diseñadas para la familia MCS-51 de intel.

2.1.3 Selección.

Como se puede observar en la tabla comparativa los tres microcontroladores cuentan con características similares y suficientes para la elaboración de este proyecto. Debido a lo anterior, se llega a la conclusión que el punto determinante para la selección de este dispositivo serán las herramientas de desarrollo.

Dado que sólo se cuenta con herramientas de desarrollo para el Intel 8751, se eligirá este microcontrolador para el desarrollo del presente trabajo.

2.2 ARQUITECTURA INTERNA.

El 8751 es un microcontrolador de 8 bits de la familia 8051 de tecnología CMOS o NMOS, sus aplicaciones son variadas en la industria. Opera con 5 Volts de C.D. con un consumo máximo de corriente de 20 mA.

El 8751 se caracteriza por tener una memoria EPROM de 4 KBytes y 128 Bytes de memoria RAM interna. Tiene 4 puertos bidireccionales de 8 bits cada uno.

A continuación se muestran sus Diagramas de Bloques interno y de Conexiones:

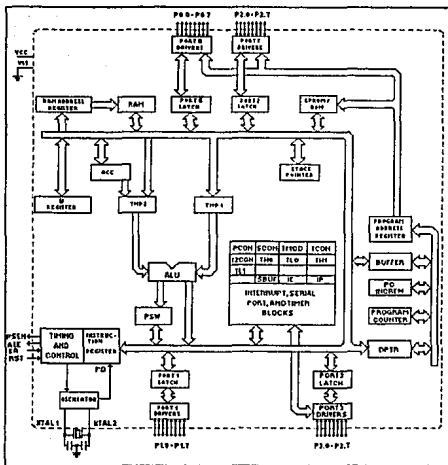


Figura 2.1 Diagrama de Bloques interno.

P1.0	1	40	VCC
P1.1	2	39	P0.0 / AD0
P1.2	3	38	P0.1 / AD1
P1.3	4	37	P0.2 / AD2
P1.4	5	36	P0.3 / AD3
P1.5	6	35	P0.4 / AD4
P1.6	7	34	P0.5 / AD5
P1.7	8	33	P0.6 / AD6
RST	9	32	P0.7 / AD7
RXD / P3.0	10	31	EA / VPP
TXD / P3.1	11	30	ALE / PROG
INT0 / P3.2	12	29	PSEN
INT1 / P3.3	13	28	P2.7 / A15
TO / P3.4	14	27	P2.6 / A14
TI / P3.5	15	26	P2.5 / A13
WR / P3.6	16	25	P2.4 / A12
RD / P3.7	17	24	P2.3 / A11
XTAL2	18	23	P2.2 / A10
XTAL1	19	22	P2.1 / A9
VSS	20	21	P2.0 / A8

Figura 2.2 Diagrama de Conexiones.

Donde:

P0.0 - P0.7 Es el puerto 0. Cuando se accesa a memoria externa, es el bus de datos multiplexado con la parte baja del bus de direcciones.

P1.0 - P1.7 Es el puerto 1.

P2.0 - P2.7 Es el puerto 2. Cuando se accesa memoria externa, es la parte alta del bus de direcciones.

P3.0 - P3.7 Es el puerto 3, el cual también se utiliza para producir señales de control de dispositivos externos como son :

RXD Puerto serie de entrada. P3.0

TXD Puerto serie de salida. P3.1

$\overline{\text{INT0}}$ Interrupción Externa 0. P3.2

$\overline{\text{INT1}}$ Interrupción Externa 1. P3.3

T0 Entrada externa Temporizador 0. P3.4

T1 Entrada externa Temporizador 1. P3.5

$\overline{\text{WR}}$ Habilitador de escritura para memoria externa de datos. P3.6

$\overline{\text{RD}}$ Habilitador de lectura para memoria externa de datos. P3.7

RST Reset.

XTAL 2 Cristal 2.

XTAL 1 Cristal 1.

VSS Tierra. Referencia 0 Volts.

VCC Alimentación.

$\overline{\text{EA}} / \text{VPP}$ Habilitador de acceso a memoria ROM externa (External Access Enable).

1 Indica que sólo existe ROM interna.

0 Indica que las instrucciones se deben buscar en la ROM externa.

VPP, recibe el Voltaje de Programación durante la programación de la EPROM.

$\overline{\text{ALE}} / \text{PROG}$ Habilitador del registro para conservar la parte baja de la dirección durante el acceso a memoria externa (Address Latch Enable).

$\overline{\text{PROG}}$, Recibe los pulsos durante la programación de la EPROM.

PSEN Habilitador de lecturas para memoria de programa externa
(Program Store Enable).

Con los temporizadores se pueden generar tiempos internos o contar eventos externos, lo cual es de gran utilidad en varias aplicaciones.

2.3 REGISTROS INTERNOS.

El 8751 tiene direcciones separadas para la memoria de programa y la memoria de datos. La memoria de programa puede ser de hasta 64 KB (Kilo-Bytes) de longitud. Los primeros 4 KB residen en el circuito. La memoria de datos es de 128 bytes en forma interna, pero puede crecer hasta 64 KB en forma externa. Además de esto, el circuito cuenta con una serie de registros especiales que sirven para realizar operaciones, manejo de puertos, etc. Algunos de estos registros especiales (SFR) son:

ACUMULADOR. Es un registro de propósito general y por su frecuencia de intervención es considerado el más importante. También se le conoce como registro A.

REGISTRO B. Este registro se encuentra especializado en las operaciones de multiplicación y división, pero puede ser utilizado como registro de propósito general.

APUNTADOR DE PILA (Stack Pointer). Es un registro de 8 bits utilizado para apuntar la última dirección de la pila, se incrementa y/o decrementa durante la ejecución de las instrucciones PUSH, POP, ACALL, LCALL, RET, y RETI.

APUNTADOR DE DATOS (Data Pointer). La función de este registro es la de retener una dirección de 16 bits (DPTR). Está formado por dos registros de 8 bits cada uno (DPH y DPL).

PUERTOS 0 A 3. Los registros P0, P1, P2 y P3 fijan los valores de los puertos 0, 1, 2 y 3 respectivamente.

BUFFER DE PUERTO SERIAL. Se compone de un registro de transmisión y uno de recepción aparentemente separados pero físicamente son el mismo.

REGISTRO DE LOS TEMPORIZADORES. Es un registro de 16 bits T0 (TH0 y TL0) y T1 (TH1 y TL1), que pueden actuar como temporizadores o contadores de 16 bits.

PSW. Contiene información sobre el status del microcontrolador.

CY	AC	F0	RS1	RS0	OV	-	P
----	----	----	-----	-----	----	---	---

CY = PSW.7 Bandera del acarreo

AC = PSW.6 Bandera del acarreo auxiliar

F0 = PSW.5 Bandera 0, disponible para propósito general

RS1 = PSW.4 Bit 1 selector del banco de registros

RS0 = PSW.3 Bit 0 selector del banco de registros

OV = PSW.2 Bandera de sobreflujo

- = PSW.1 Bandera de propósito general

P = PSW.0 Bandera de paridad

REGISTROS DE CONTROL

Mediante la activación de los bits de los siguientes registros, es posible configurar las diferentes formas de funcionamiento de los temporizadores / contadores e interrupciones.

IE : Habilitador de Interrupciones (direccionable por bit)

EA	-	-	ES	ET1	EX1	ET0	EX0
----	---	---	----	-----	-----	-----	-----

EA = IE.7 Habilita o deshabilita todas las interrupciones, es decir, si EA=0 todas las interrupciones serán ignoradas, si EA=1 cada interrupción podrá ser habilitada o deshabilitada individualmente.

- IE.6 No implementada.

- IE.5 No implementada.

ES IE.4 Habilita la interrupción del puerto serial.

ET1 IE.3 Habilita la interrupción por sobreflujo del temporizador 1.

EX1 IE.2 Habilita la interrupción externa 1.

ET0 IE.1 Habilita la interrupción por sobreflujo del temporizador 0.

EX0 IE.0 Habilita la Interrupción externa 0.

IP : Prioridad de Interrupciones (direccionable por bit).



- IP.7 No implementada.

- IP.6 No implementada.

- IP.5 No implementada.

PS IP.4 Define el nivel de prioridad del puerto serial.

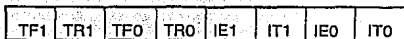
PT1 IP.3 Define el nivel de prioridad del temporizador 1.

PX1 IP.2 Define el nivel de prioridad de la interrupción externa 1.

PT0 IP.1 Define el nivel de prioridad del temporizador 0.

PX0 IP.0 Define el nivel de prioridad de la interrupción externa 0.

TCON : Registro de Control de Temporizadores, Contadores e Interrupciones
(direccionable por bit).



TF1 TCON.7 Bandera de sobreflujo del temporizador 1. Se enciende cuando el temporizador pasa de FFFF a 0000.

TR1 TCON.6 Bit de Encendido / Apagado del temporizador 1.

TF0 TCON.5 Bandera de sobreflujo del temporizador 0. Se enciende cuando el temporizador pasa de FFFF a 0000.

TR0 TCON.4 Bit de encendido / apagado del temporizador 0.

IE1 TCON.3 Bandera de activación para la interrupción externa 1. Se activa cuando se detecta una transición en la línea de la interrupción 1. Se apaga cuando se procesa la interrupción.

IT1 TCON.2 Control de tipo de la interrupción 1.

IE0 TCON.1 Bandera de activación para la interrupción externa 0. Se activa cuando se detecta una transición en la línea de la interrupción 0. Se apaga cuando se procesa la interrupción.

IT0 TCON.0 Control de tipo de la interrupción 0.

SCON : Registro de Control de Energía. No es direccionable por bit.

SMOD	-	-	-	GF1	GF0	PD	IDL
------	---	---	---	-----	-----	----	-----

SMOD Bit para doblar el "baud-rate" (velocidad de transmisión de datos). Si se utiliza el temporizador 1 para generar el "baud-rate" y SMOD=1, el baud rate será duplicado al utilizar el puerto serial en los modos 1, 2 o 3.

- No implementada.

- No implementada.

- No implementada.

- GF1 Bandera de propósito general
- GF0 Bandera de propósito general
- PD Bit para apagado. Solo en la versión CMOS.
- IDL Bit para modo Inactivo. Solo en la versión CMOS.
- PCON : Registro de Control del Puerto Serial.

SM0	SM1	SM2	REN	TB8	RB8	TI	RI
-----	-----	-----	-----	-----	-----	----	----

SM0 SCON.7 Especifican el modo de operación del puerto serie.

SM1 SCON.6

SM0	SM1	MOD0	ESPECIFICACION	BAUD RATE
0	0	0	Registro de Corrimiento	f. osc. / 12
0	1	1	UART 8 bits	variable
1	0	2	UART 9 bits	f. osc. / 32 ó / 64
1	1	3	UART 9 bits	variable

SM2	SCON.5	Habilita la comunicación del tipo multiprocesador utilizada en los Modos 2 y 3. En estos modos, si SM2 = 1, RI no se activa si el noveno dato recibido (RB8) es 0. En modo 1, RI no se activa si no se recibe un bit de stop. En el modo 0, SM2 será 0.
REN	SCON.4	Establece la recepción serie, cuando REN = 0 se desactiva la recepción.
TB8	SCON.3	Almacena el noveno bit que será transmitido en los modos 2 y 3.
RB8	SCON.2	Es el Noveno bit que fue recibido en los modos 2 y 3. En el modo 1, si SM2 = 0, RB8 es el bit de paro recibido. En modo 0, RB8 no se utiliza.
TI	SCON.1	Bandera de interrupción de la transmisión.
RI	SCON.0	Bandera de interrupción de la recepción.

TMOD: Registro de Configuración de Temporizadores y Contadores. Es direccionable por bit.

GATE	C / T	M1	M0	GATE	C / T	M1	M0
------	-------	----	----	------	-------	----	----

TEMPORIZADOR 1

TEMPORIZADOR 0

GATE Cuando TRx (en TCON) está encendido y GATE=1 el temporizador/contador "x" correrá mientras el pin INT"x" se mantenga activado (control por hardware).

Cuando GATE=0, el temporizador/contador "x" correrá mientras el bit TR"x" se encuentre encendido (control por software).

C / \bar{T} Selector entre Temporizador o Contador

M1 Bit selector de modo

M0 Bit selector de modo

M1	M0	MODO DE OPERACION
0	0	0 Temporizador de 13 bits.
0	1	1 Temporizador/Contador de 16 bits.
1	0	2 Temporizador/Contador de 8 bits autorecargable
1	1	3 Temporizador 0. TL0 es un Temporizador/Contador de 8 bits controlado por los bits de control estándar del temporizador 0. TH0 es un Temporizador/Contador de 8 bits controlado por los bits de control del temporizador 1. 3 Temporizador 1. El Temporizador/Contador 1 no se utiliza.

2.4 MEMORIA ROM Y RAM.

La memoria del microcontrolador 8751 se puede dividir en tres partes fundamentales:

a) **Memoria de Programa.** Es en donde se encuentran todas las instrucciones que van a ser ejecutadas por el microcontrolador, es decir, el programa de trabajo. Dependiendo de la versión del mismo es la cantidad de memoria de programa interna con la que cuenta. Cuando se requiere trabajar con mayor capacidad de memoria de la disponible internamente, se cuenta con la memoria de programa externa, la cual se selecciona mediante la activación de la señal PSEN. El máximo espacio de memoria de programa (interna más externa) que se puede acceder es de 64 KB.

b) Memoria RAM Interna. Esta conformada por 128 bytes (00H a 7FH) y en la parte baja de esta memoria se encuentran cuatro bancos de siete registros cada uno R0 - R7. El microcontrolador 8751 cuenta con Registros de Funciones Especiales (SFR), los cuales ocupan las localidades de memoria de 80H hasta F8H también direccionables como localidades de memoria. Si es necesaria una mayor cantidad de memoria se podrá añadir hasta un total de 64 KB de memoria RAM externa, cuyas localidades de memoria empezarán en la 0000H.

c) Memoria de Datos Externa. Se activa mediante el encendido de las señales RD y WR, durante la lectura o escritura de datos respectivamente. En este espacio el microcontrolador toma todos los valores que se encuentran en memoria como datos, es decir, el micro no puede ejecutar ninguna instrucción que se encuentre aquí almacenada. Puede direccionar hasta 64 KB de memoria de datos.

2.5 MANEJO DE INTERRUPCIONES VECTORIALES.

Existen cinco posibles fuentes de interrupción:

1) Interrupciones Externas (2). Las interrupciones externas INT0 e INT1, se pueden activar por nivel o transición, dependiendo del contenido del registro de función especial (SFR) TCON.

2) Interrupciones de los Temporizadores (2). Estas interrupciones se generan al existir un desborde en cualquiera de los dos temporizadores.

3) Interrupción Serial. Se genera por el puerto serie al terminar la recepción o transmisión de un byte. La rutina de servicio asociada a esta interrupción deberá determinar que tipo de evento ocurrió.

Las interrupciones cuentan con nivel de prioridad, con la finalidad de permitir un manejo más eficiente de ellas.

Cada una de las fuentes de interrupción provoca una llamada a la rutina de servicio ubicada en la posición de memoria establecida por el vector de interrupción. En este caso, el vector de interrupción así como el nivel de prioridad se muestran en la siguiente tabla.

FUENTE DE INTERRUPCION	VECTOR DE DIRECCION	PRIORIDAD
Interrupción Externa 0	03 H	1
Interrupción del Temporizador 0	0B H	2
Interrupción Externa 1	13 H	3
Interrupción del Temporizador 1	1B H	4
Interrupción del Puerto Serial	23 H	5

2.6 COMUNICACION SERIAL.

La interfaz serial del 8751 es del tipo "full duplex", esto significa que puede transmitir y recibir datos simultáneamente. Este puerto está formado por un "buffer", lo que hace posible que se pueda estar recibiendo un segundo byte de datos seriales antes de que el primer byte recibido sea leído. Sin embargo, el primero deberá ser leído antes que la recepción del segundo sea completada. La transmisión comienza con cualquier instrucción que tenga como destino el SBUF.

Las operaciones del puerto serial son controladas mediante el registro de control del puerto serial SCON y se tienen 4 modos de operación que a continuación se describen:

- MODO 0.** Los datos de recepción o transmisión son enviados mediante 8 corrimientos. El "baud rate" se fija a 1/12 de la frecuencia de oscilación del 8751.

- MODO 1.** El puerto serial recibe o transmite 10 bits; 1 bit de arranque, 8 de datos y 1 de paro. El "baud rate" se programa mediante el Temporizador 1 del 8751.

- MODO 2.** Recibe o transmite 11 bits; 1 bit arranque, 8 de datos, 1 de paro y 1 programable, el cual se toma del SCON.3. El "baud rate" se puede seleccionar a 1/32 o 1/64 de la frecuencia de oscilación del 8751.

MODO 3. Este funciona exactamente igual que el MODO 2 pero con la diferencia que en este el "baud rate" es variable. Este se fija a 1/12 de la frecuencia de oscilación del 8751.

2.7 INSTRUCCIONES.

El juego de instrucciones de la familia MCS-51 está formado por 111 instrucciones, 49 de las cuales son de un solo byte, 46 de dos bytes y 16 de tres bytes. El formato de la instrucción consiste de un mnemónico seguido de un campo "destino / fuente". Este campo especifica el tipo de dato y el modo de direccionamiento que se está utilizando.

Modos de Direccionamiento.

Los modos de direccionamiento que se presentan en el microcontrolador 8751 son los siguientes:

1) Direccionamiento Directo.

En este tipo de direccionamiento el operando se especifica con una dirección de 8 bits dentro de la instrucción. Es exclusivo para los Datos de la RAM interna y los SFR.

2) Direccionamiento Indirecto.

En este se utiliza un registro, en el que se encuentra la dirección del operando. Toda la memoria RAM tanto interna como externa se puede direccionar indirectamente.

El único registro de dirección de 16 bits es el DPTR.

3) Direccionamiento Inmediato.

En este tipo de direccionamiento, el código de instrucción es seguido de una constante.

4) Direccionamiento Indexado.

Este tipo de direccionamiento se utiliza para acceder únicamente a la memoria de programa y sólo permite su lectura.

5) Direccionamiento por Registro.

Los 8 registros (R0 a R7) de cualquiera de los 4 bancos de la memoria RAM se pueden acceder mediante ciertas instrucciones que simplifican sus códigos de operación y en la mayoría de los casos son más rápidas. La selección del banco de memoria que se utiliza está dada por los bits 3 y 4 del PSW.

JUEGO DE INSTRUCCIONES

El juego de instrucciones de la familia MCS-51 se puede dividir en 5 grupos funcionales:

2.7.1 Instrucciones Aritméticas.

MNEMONICO	DESCRIPCION	# BYTES
ADD A,Rn	Suma un Registro al Acumulador	1
ADD A,directo	Suma cualquier registro al Acumulador	2
ADD A,@Ri	Suma una localidad de memoria RAM en forma indirecta al Acumulador	1
ADD A,#dato	Suma el dato al Acumulador	2
ADDC A,Rn	Suma un Registro R0-R7 al Acumulador con acarreo	1
ADDC A,directo	Suma cualquier registro al Acumulador con acarreo	2
ADDC A,@Ri	Suma una localidad de memoria RAM en forma indirecta al Acumulador con acarreo	1
ADDC A,#dato	Suma el dato al Acumulador con acarreo	2
SUBB A,Rn	Resta el Registro R0-R7 del Acumulador con prestado	1
SUBB A,directo	Resta cualquier registro del Acumulador con prestado	2
SUBB A,@Ri	Resta una localidad de memoria RAM en forma indirecta del Acumulador con prestado	1
SUBB A,#dato	Resta el dato del Acumulador con prestado	2
INC A	Incrementa el Acumulador	1
INC Rn	Incrementa un Registro R0-R7	1
INC directo	Incrementa cualquier registro	2

INC @Ri	Incrementa una localidad de memoria RAM en forma indirecta	1
DEC A	Decrementa el Acumulador	1
DEC Rn	Decrementa un Registro R0-R7	1
DEC directo	Decrementa cualquier registro	2
DEC @Ri	Decrementa una localidad de memoria RAM en forma indirecta	1
INC DPTR	Incrementa el Apuntador de Datos	1
MUL AB	Multiplica A y B	1
DIV AB	Divide A entre B	1
DA A	Ajuste Decimal del Acumulador	1

2.7.2 Instrucciones Lógicas.

MNEMONICO	DESCRIPCION	# BYTES
ANL A,Rn	Ejecuta un AND entre el Acumulador y un Registro R0-R7	1
ANL A,directo	Ejecuta un AND entre el Acumulador y cualquier registro	2
ANL A,@Ri	Ejecuta un AND entre el Acumulador y una localidad de memoria RAM en forma indirecta	1
ANL A,#dato	Ejecuta un AND entre el Acumulador y un dato	2
ANL directo,A	Ejecuta un AND entre cualquier registro y el Acumulador	2

ANL directo,#dato	Ejecuta un AND entre cualquier registro y un dato	3
ORL A,Rn	Ejecuta un OR entre el Acumulador y un Registro R0-R7	1
ORL A,directo	Ejecuta un OR entre el Acumulador y cualquier registro	2
ORL A,@RI	Ejecuta un OR entre el Acumulador y una localidad de memoria RAM en forma indirecta	1
ORL A,#dato	Ejecuta un OR entre el Acumulador y un dato	2
ORL directo,A	Ejecuta un OR entre cualquier registro y el Acumulador	2
ORL directo,#dato	Ejecuta un OR entre cualquier registro y un dato	3
XRL A,Rn	Ejecuta un OR exclusivo entre el Acumulador y un Registro R0-R7	1
XRL A,directo	Ejecuta un OR exclusivo entre el Acumulador y cualquier registro	2
XRL A,@RI	Ejecuta un OR exclusivo entre el Acumulador y una localidad de memoria RAM en forma indirecta	1
XRL A,#dato	Ejecuta un OR exclusivo entre el Acumulador y un dato	2
XRL directo,A	Ejecuta un OR exclusivo entre un dato y el Acumulador	2
XRL directo,#dato	Ejecuta un OR exclusivo entre cualquier registro y un dato	3

CLR A	Limpia el Acumulador	1
CPL A	Complementa al Acumulador	1
RL A	Rota el Acumulador a la izquierda	1
RLC A	Rota el Acumulador a la izquierda con acarreo	1
RR A	Rota el Acumulador a la derecha	1
RRC A	Rota el Acumulador a la derecha con acarreo	1
SWAP A	Intercambia los primeros 4 bits con los segundos 4 bits del Acumulador	1

2.7.3 Instrucciones de Transferencia de Datos.

MNEMONICO	DESCRIPCION	# BYTES
MOV A,Rn	Mueve un Registro R0-R7 al Acumulador	1
MOV A,directo	Mueve cualquier registro al Acumulador	2
MOV A,@Ri	Mueve una localidad de memoria RAM en forma indirecta al Acumulador	1
MOV A,#dato	Mueve un dato al Acumulador	2
MOV Rn,A	Mueve el Acumulador a un Registro R0-R7	1
MOV Rn,directo	Mueve un byte a un Registro R0-R7	2
MOV Rn,#dato	Mueve un dato a un Registro R0-R7	2
MOV directo,A	Mueve el Acumulador a cualquier registro	2
MOV directo,Rn	Mueve un Registro R0-R7 a cualquier registro	2
MOV directo,directo	Mueve cualquier registro a cualquier otro registro	3

MOV directo,@Ri	Mueve una localidad de memoria RAM en forma indirecta a cualquier registro	2
MOV directo,#dato	Mueve un dato a cualquier registro	3
MOV @Ri,A	Mueve el Acumulador a una localidad de memoria RAM en forma indirecta	1
MOV @Ri,directo	Mueve cualquier registro a una localidad de memoria RAM en forma indirecta	2
MOV @Ri,dato	Mueve un dato a una localidad de memoria RAM en forma indirecta	2
MOV DPTR,#dato 16	Carga al Apuntador de Datos con una constante de 16 bits	3
PUSH directo	Manda un byte a la localidad indicada por el apuntador de pila y lo incrementa en 1	2
POP directo	Lee el byte de la localidad indicada por el apuntador de pila y lo decremента 1	2
XCH A,Rn	Intercambia valores entre el Acumulador y un Registro.	1
XCH A,directo	Intercambia los valores del Acumulador y un byte	2
XCH A,@Ri	Intercambia los valores del Acumulador y una localidad de memoria indirecta	1
XCHD A,@Ri	Intercambia los 4 LSB del Acumulador y una localidad de memoria indirecta	1
MOVC A,@A+DPTR	Mueve la suma del Acumulador y el DPTR al Acumulador	3

MOVC A,@A+PC	Mueve la suma del Acumulador y el PC al Acumulador	1
MOVX A,@Ri	Mueve memoria RAM externa al Acumulador (dirección de 8 bits)	1
MOVX A,@DPTR	Mueve memoria RAM externa al Acumulador (dirección de 16 bits)	1
MOVX A,@Ri,A	Mueve al Acumulador a memoria RAM externa (dirección de 8 bits)	1
MOVX @DPTR,A	Mueve el Acumulador a memoria RAM externa (dirección de 16 bits)	1

2.7.4 Instrucciones Booleanas.

MNEMONICO	DESCRIPCION	# BYTES
CLR C	Limpia el acarreo	1
CLR bit	Limpia un determinado bit	2
SETB C	Enciende el acarreo	1
SETB bit	Enciende un determinado bit	2
CPL C	Complementa el acarreo	1
CPL bit	Complementa un determinado bit	2
ANL C,bit	Ejecuta un AND entre el acarreo y un bit	2
ANL C,/bit	Ejecuta un AND entre el acarreo y el complemento de un bit	2
ORL C,bit	Ejecuta un OR entre el acarreo y un bit	2

ORL C,bit	Ejecuta un OR entre el acarreo y el complemento de un bit	2
MOV C,bit	Mueve un determinado bit al acarreo	2
MOV bit,C	Mueve el acarreo a un determinado bit	2
JC rel	Salta si el acarreo está encendido	2
JNC rel	Salta si el acarreo está apagado (no encendido)	2
JB rel	Salta si un determinado bit está encendido	2
JNB rel	Salta si un determinado bit está apagado	2
JBC bit,rel	Salta si un determinado bit está encendido y apaga a dicho bit	3

2.7.5 Instrucciones de Salto.

MNEMONICO	DESCRIPCION	# BYTES
ACALL addr11	Llama a una subrutina absoluta	2
LCALL addr16	Llamada larga de subrutina	3
RET	Regreso de una subrutina	1
RETI	Regreso de una interrupción	1
AJMP addr11	Salto absoluto	2
LJMP addr16	Salto largo	3
SJMP rel	Salto corto	2
JMP @a+DPTR	El PC salta al resultado de la suma del Acumulador y el DPTR	1
JZ rel	Salta si el Acumulador es cero	2
JNZ rel	Salta si el Acumulador no es cero	2

CJNE A,directo,rel	Compara cualquier registro con el Acumulador y si no son iguales salta	3
CJNE A,#dato,rel	Compara un dato con el Acumulador y si no son iguales salta	3
CJNE Rn,#dato,rel	Compara un dato con un Registro R0-R7 y si no son iguales salta	3
CJNE @Ri,#dato,rel	Compara un dato con una localidad de memoria RAM en forma indirecta y si no son iguales salta	3
DJNZ Rn,rel	Decrementa el Registro R0-R7 y si no es cero salta	2
DJNZ directo,rel	Decrementa cualquier registro y si no es cero salta	3
NOP	No hay operación	1

CAPITULO 3

DISEÑO DEL SISTEMA

El sistema propuesto se basa en el diseño de una tarjeta de adquisición de datos, cuya función será muestrear variables de un proceso que se ajuste a la capacidad de la tarjeta en cuestión, así mismo será capaz de comunicarse con una computadora, esto con la finalidad de poderle enviar los datos obtenidos durante el muestreo.

3.1 ESPECIFICACIONES DEL SISTEMA

TARJETA DE ADQUISICION DE DATOS.

- 1) Ocho canales de entrada de señales analógicas (0 - 5 volts).
- 2) Hasta 16 canales digitales de entrada.
- 3) La comunicación con la computadora será a través del puerto serial RS-232C.
- 4) Periodo de Muestreo de los canales analógicos y/o digitales configurable desde 1 hasta 15 segundos.

3.1.1. Descripción del sistema

1) La tarjeta de adquisición muestrea las señales analógicas y digitales a un periodo de muestreo constante y configurable desde la computadora personal. El periodo de muestreo puede ser desde 1 hasta 15 segundos para todos los canales de entrada. La operación de muestreo se realizará en forma continua hasta que la computadora indique, a través de un mensaje, otra operación.

- 2) La tarjeta estará esperando (mediante interrupción) algún mensaje de la computadora personal. Al recibir el mensaje, lo decodificará y realizará la operación indicada. Los mensajes y las operaciones que la computadora puede enviar a la tarjeta de adquisición de datos pueden ser:

Mensaje	Función
4CH	Envío de datos a la PC
5XH	Cambio de periodo de muestreo
47H	Sólo lectura de variables lógicas
3XH	Muestreo X líneas analógicas
6XH	Muestreo X líneas analógicas y lógicas

- 3) La computadora recibe los datos muestreados y los presenta al usuario en forma de gráficas de tendencia, en el caso de variables analógicas, y en el caso de variables lógicas en forma de gráfica de barras. El operador podrá seleccionar una de las gráficas de tendencia para ocupar toda la pantalla. La información que se presente en la pantalla se podrá almacenar en algún archivo.
- 4) La computadora podrá enviar cualquier mensaje en cualquier momento .
- 5) La tarjeta no podrá comunicarse con la computadora por sí sola. Siempre se comunicará como respuesta a un mensaje de la computadora.

3.1.2. Diagrama de bloques específico

El diagrama de bloques del sistema se observa en la siguiente figura :

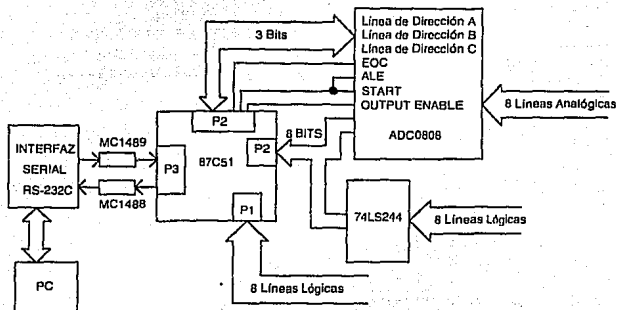


Figura 3.1 Diagrama de bloques específico.

3.2. REQUERIMIENTOS DE HARDWARE

3.2.1. Configuración del 8751.

- Como se requiere un reloj de tiempo real con base de tiempo de 1 segundo, el temporizador 0 se utilizará para este fin. Para ello, el temporizador 0 operará en modo 1 (temporizador de 16 bits).

- El modo de control del puerto serie será el modo 1, es decir, 1 bit de inicio, 8 bits de datos, 1 bit de paro y con una velocidad de transmisión o "baud rate" dado por el temporizador 1.
- Ya que se empleará el puerto serie, se requiere que el temporizador 1 opere en modo 2 para la generación del "baud rate", es decir, en modo recargable. El valor de conteo se encuentra fijo en el registro TH1, el cual se carga en TL1 cada vez que existe un desbordamiento. Debido a que se empleará una velocidad de transmisión baja SMOD estará cargado con 0, con el fin de no duplicar la velocidad de transmisión.
- Para asegurar que el microcontrolador comience a ejecutar el programa en el momento en que se alimente el circuito, se necesita tener una red de "reset" automático, de manera que la fuente de alimentación se estabilice.
- Se necesitan 8 bytes de la memoria RAM para poder almacenar los valores de las variables analógicas, y 2 bytes para los valores de las variables lógicas.
- Para poder lograr la comunicación con el convertidor analógico/digital se utilizarán los bits del puerto P0 para la comunicación de datos y el puerto P2 para el control y establecer las líneas de direccionamiento de las entradas analógicas del convertidor.
- Para poder recibir y transmitir datos entre el microcontrolador y una computadora personal se utilizará el puerto P3 del microcontrolador.

Circuito oscilador

El microcontrolador 8751 cuenta con un oscilador integrado en el mismo circuito. Este oscilador está compuesto por una etapa lineal inversora en la cual el cristal es operado como una reactancia inductiva en resonancia paralela con una capacitancia externa.

El circuito equivalente del oscilador se muestra en la figura 3.2.

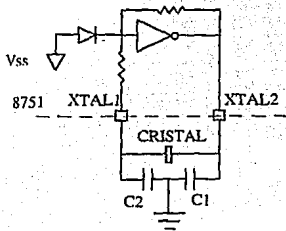


Figura 3.3 Oscilador Interno

Para poder utilizar este oscilador como un sistema controlado por cristal, se debe conectar un cristal entre los pines XTAL1 y XTAL2, que determinará el período de oscilación del circuito. En cada uno de estos pines, se conecta un capacitor a tierra con el propósito de provocar una resonancia en paralelo con dicho cristal.

La influencia de los capacitores conectados a las terminales XTAL1 y XTAL2, repercuten en el tiempo de establecimiento del oscilador y en su estabilidad. Dado que el controlador requiere de un reloj de tiempo real, se elegirán capacitores que proporcionen una máxima estabilidad en la frecuencia de oscilación. En estos casos, el fabricante recomienda que el valor de estos capacitores sea de 30 pF cada uno.

Debido a que esta aplicación no requiere de gran velocidad de proceso, se optó por seleccionar un cristal de baja frecuencia, con el objeto de no tener problemas en la construcción del circuito por efecto de capacitancias parásitas.

El fabricante especifica que el microcontrolador 8751 puede utilizar cristales desde 3.5 hasta 12 MHz., posteriormente se observarán los parámetros para la selección del cristal y se determinará el valor más adecuado.

El diagrama de conexiones del circuito oscilador se muestra en la Figura 3.3.

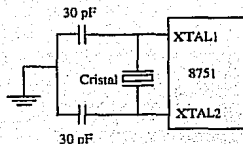


Figura 3.3 Conexión del circuito oscilador

Circuito de reset automático

Con el objeto de asegurar que el microcontrolador comience a ejecutar el programa en cuanto se allmente el circuito, es necesario construir una red de "reset" automático. Esta red, causará un estado de "reset" mientras la fuente de alimentación se estabiliza, inicializando los registros internos y configurando las señales de ALE y PSEN.

La entrada de "reset" del microcontrolador se activa en alto, y debe permanecer en este estado al menos por dos ciclos de máquina, es decir 24 oscilaciones de reloj, después de haberse estabilizado la fuente de alimentación.

Un circuito de "reset" automático puede obtenerse al encender el circuito conectando el pin RST hacia VCC a través de un capacitor de 10 μF , y hacia tierra por medio de una resistencia de 8.2 $\text{k}\Omega$. Estos valores son válidos siempre y cuando la fuente de alimentación tenga un tiempo de estabilización de menos de 1 ms y el tiempo de arranque del oscilador no exceda los 10 ms. El circuito de "reset" automático se observa en la Figura 3.4.

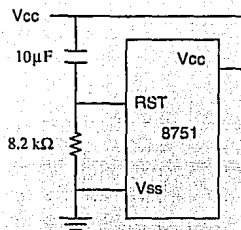


Figura 3.4 Circuito de "reset" automático

Cuando se alimenta al circuito por primera vez, la corriente que fluye a través de RST comienza a cargar el capacitor. El voltaje en RST es la diferencia entre VCC y el voltaje del capacitor y decrece conforme el capacitor se carga. Entre más grande sea el valor del capacitor, más lentamente disminuirá el voltaje de RST. El voltaje de RST debe permanecer abajo del umbral del disparador interno del microcontrolador lo suficiente para cubrir el tiempo de establecimiento de la fuente y el ciclo de "reset".

3.2.2 Convertidor analógico digital de 8 bits con 8 canales multiplexados ADC0808

Debido a que en las especificaciones de la tarjeta de adquisición de datos se menciona la capacidad de muestrear variables analógicas y el microcontrolador por sí sólo únicamente es capaz de intercambiar datos lógicos, es necesario complementar dicha tarjeta con un convertidor analógico/digital para lograr la comunicación entre las variables analógicas y el microcontrolador.

Descripción general

El ADC0808 es un convertidor analógico/digital (ADC) CMOS monolítico de 8 bits, con 8 canales multiplexados y una lógica de control compatible con microprocesadores. El convertidor A/D de 8 bits utiliza la aproximación sucesiva como la técnica de conversión. El convertidor está formado por un comparador, un divisor de voltaje 256R con un árbol de interruptores analógicos y un registro de aproximación sucesiva. El multiplexor de 8 canales puede acceder directamente a cualquiera de las 8 señales analógicas.

El dispositivo no necesita ajuste de escala completa ni de cero.

Características

El corazón de este circuito es el convertidor digital/analógico de 8 bits. El convertidor está diseñado para ofrecer conversiones exactas, rápidas y repetitivas en un amplio rango de temperaturas. El convertidor está dividido en 3 secciones principales; la red de escalera (convertidor D/A), el registro de aproximación sucesiva y el comparador.

El registro de aproximaciones sucesivas (SAR) realiza "n" iteraciones para aproximar al voltaje de entrada. Cualquier convertidor A/D de "n" bits por aproximaciones sucesivas requiere de "n" iteraciones para realizar la conversión.

La operación de un convertidor A/D por aproximaciones sucesivas es de la siguiente manera:

Con el primer pulso de inicio de conversión, el registro de aproximaciones sucesivas envía un "1" en el bit más significativo (MSB, Most Significant Bit) al convertidor D/A. La salida de este convertidor se envía al comparador junto con la proveniente de la variable analógica, si la señal analógica es mayor, se encenderá el siguiente bit, el bit 7 en el caso de un convertidor de 8 bits, por lo que ahora el D/A enviará un valor mayor al comparador. En caso de que la señal analógica sea menor que la enviada por el D/A se apagará el último bit encendido (puede ser incluso el primero) y se encenderá el inmediato inferior, para así continuar hasta tener los 8 bits. Una vez terminada la conversión el resultado en binario se encuentra en la salida en paralelo del registro de

aproximaciones sucesivas y este envía una señal de término de conversión (EOC End Of Conversion)

La conversión en proceso será interrumpida al recibir un nuevo pulso de inicio de conversión. La conversión continua se puede realizar conectando la salida del pulso de fin de conversión (EOC) a la entrada de la señal de inicio de conversión (SC Start Conversion). Si se usa de esta manera, se deberá dar un pulso externo de inicio de conversión después de encenderlo.

Este convertidor es fácilmente interconectado con microprocesadores y microcontroladores. Cuenta con 8 canales analógicos de entrada multiplexados, los cuales se pueden seleccionar mediante un decodificador de direcciones. El rango de entrada de estos canales es de 0 a 5 Volts.

El voltaje de alimentación para este circuito es de 5 Volts, sus salidas aceptan especificaciones de nivel de voltaje TTL y la frecuencia de reloj típica es de 640 KHz (mínima 10 KHz, máxima 1280 KHz).

Especificaciones

- Resolución de 8 bits
- Error total $\pm 1/2$ LSB y ± 1 LSB
- Alimentación 5 V_{CD}
- Consumo de energía 15 mW
- Tiempo de conversión 90 μ s (mínimo), 100 μ s (típico), 116 μ s (máximo)

- Frecuencia del reloj 10 kHz (mínimo), 640 kHz (típico), 1280 kHz (máximo)

En las figuras 3.5 y 3.6 se muestran los diagramas de bloques y de conexiones respectivamente.

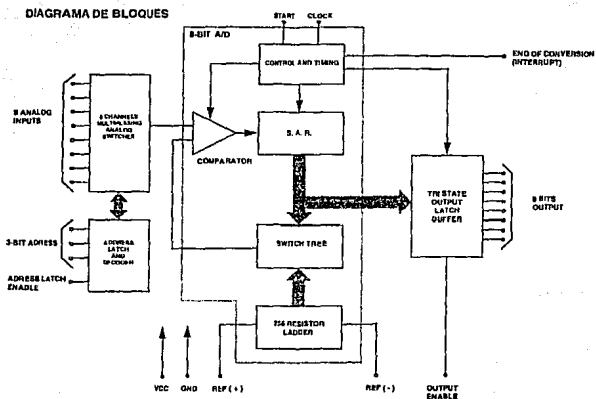


Figura 3.5. Diagrama de bloques del ADC0808

IN 3	1	28	IN 2
IN 4	2	27	IN 1
IN 5	3	26	IN 8
IN 6	4	25	ADD A
IN 7	5	24	ADD B
START	6	23	ADD C
EOC	7	22	ALE
D3	8	21	D7 MSB
OE	9	20	D6
CLOCK	10	19	D5
VCC	11	18	D4
REF (+)	12	17	D0 LSB
GND	13	16	REF (-)
D1	14	15	D2

Figura 3.6 Diagrama de conexiones del convertidor ADC0808

Donde :

IN1 - IN8 Son los canales analógicos de entrada.

ADD A,B,C Son los selectores del canal de entrada a convertir.

START Es el habilitador para la conversión.

EOC	Indica la finalización de la conversión (End Of Conversion).
ALE	Habilita el registro de direcciones (Address Latch Enable).
D0 - D7	Los datos de salida (1 byte).
OE	Habilita la salida.
CLOCK	La señal de reloj.
VCC	Alimentación.
REF (+)	Referencia Positiva.
REF (-)	Referencia Negativa.
GND	Tierra 0V referencia.

Este convertidor cuenta con un multiplexor de 8 canales de entrada. La entrada deseada se selecciona por medio del decodificador de direcciones. En la siguiente tabla se muestran los estados de las líneas de direcciones para seleccionar cualquier canal.

CANAL ANALOGICO	LINEA DE DIRECCIONES		
	SELECCIONADO	C	B
IN0	L	L	L
IN1	L	L	H
IN2	L	H	L
IN3	L	H	H
IN4	H	L	L
IN5	H	L	H
IN6	H	H	L
IN7	H	H	H

3.2.3. Transmisión Serial RS-232C.

A partir de los años 60's se propagó el uso de computadoras como terminales de tiempo compartido, por lo que se diseñaron modems (modulator-demodulator) de manera que dichas terminales pudieran comunicarse a través de líneas telefónicas con computadoras remotas. Los modems y otros dispositivos utilizados para la transmisión serial de datos son conocidos como equipo de comunicación de datos (DCE, Data Communication Equipment). Las terminales o computadoras que están enviando o recibiendo los datos se conocen como equipo de terminales de datos (DTE, Data Terminal Equipment). En respuesta a la necesidad de hacer compatible la señal entre los estándares DTE y DCE, la Asociación de Industrias Electrónicas (EIA, Electronic Industries Association) desarrolló el EIA RS232-C. Este estándar describe las funciones de

las 25 terminales de señales para la transferencia serial de datos. También describe los niveles de voltaje, niveles de impedancia, el máximo "bps" (bits por segundo) y la capacitancia máxima para estas líneas de señal. Para cuestiones de la presente tesis se empleará el estándar para 9 terminales de señal. Antes de empezar a trabajar con las funciones de los 9 terminales, se dará un breve vistazo a algunos otros aspectos del "hardware" del RS232-C.

El RS232-C especifica que deben ser 9 terminales para el manejo de señal, que el conector DTE debe ser macho y el conector DCE debe ser hembra. Cuando se están alamblando estos conectores es importante observar el orden de la numeración de dichas terminales.

Los niveles de voltaje para todas las señales RS232-C son los siguientes. Un alto lógico o "marca" es un voltaje entre -3 Volts y -15 Volts con carga (-25 V sin carga). Un bajo lógico o "espacio" es un voltaje entre +3V y +15V con carga (+25 V sin carga). Comúnmente se utilizan Voltajes de ± 12 Volts.

Interfaz entre RS-232C y TTL

Debido a que los niveles de señal de una computadora no son directamente compatibles con los niveles de TTL, es necesario implementar una interfaz para lograr que lo sean.

Una manera de lograr la interfaz entre los niveles RS232-C y TTL es a través de los circuitos MC1488 y MC1489, los cuales cuentan con 4 "drivers" TTL a RS-232C y 4 "drivers" RS-232C a TTL respectivamente tal y como se muestran en la figura 3.7. El MC1488 requiere de voltajes + y -, pero el MC1489 requiere sólo 5 volts. Las salidas de los "drivers" del MC1488 se deben conectar a tierra por medio de un capacitor.

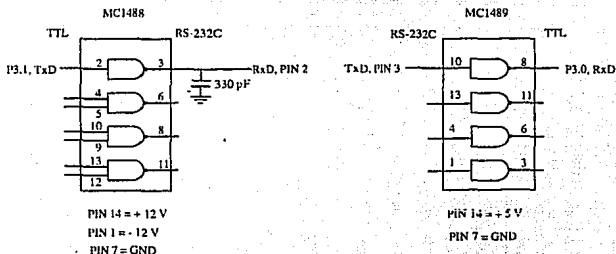


Figura 3.7 Conversión de señales TTL-RS-232C y RS-232C - TTL

Terminales de señal del RS-232C

En muchas aplicaciones sólo unas cuantas terminales de las 9 disponibles se utilizan. Para el presente trabajo se emplearán las terminales 2, 3 y 5, ya que estos son suficientes para cumplir con el propósito de transmitir y recibir tal como se puede observar en la siguiente tabla.

PIN	DESCRIPCION	SEÑAL
1	Carrier Detect	CD
2	Receiver Data	RxD
3	Transmitter Data	TxD
4	Data Terminal Ready	DTR
5	Signal Ground	GND
6	Data Set Ready	DSR
7	Request To Send	RTS
8	Clear To Send	CTS
9	Ring Indicator	RI

3.2.4. Buffer 74LS244.

Debido a que el puerto 0 del microcontrolador se utiliza como entrada tanto para los 8 bits de datos del convertidor A/D como para 8 variables digitales es necesario que al estar habilitada la entrada de uno de ellos, la otra se encuentre en estado de alta impedancia, esto con la finalidad de evitar posibles conflictos.

Características

- Salida de 3 estados.
- La corriente de la fuente que maneja es de 15 mA.
- Tiempo típico de habilitación/deshabilitación es de 18 ns.

El diagrama de conexiones del buffer se muestra en la Figura 3.8.

En la siguiente tabla se aprecian los diferentes estados que puede presentar la salida en función de las diferentes combinaciones de entradas.

Entradas		Salida
\bar{G}	A	Y
L	L	L
L	H	H
H	X	Z

L = Nivel lógico bajo

H = Nivel lógico alto

X = Cualquier nivel ya sea alto o bajo

Z = Alta impedancia

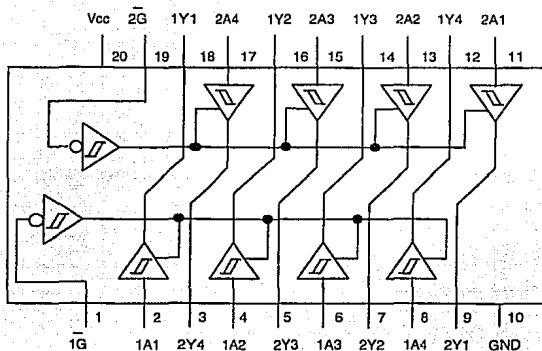


Figura 3.8 Diagrama de Conexiones del 74LS244.

3.2.5. Selección del cristal de cuarzo

Para la selección del cristal de cuarzo se deben tomar en cuenta las siguientes consideraciones:

1. Rango de frecuencias del oscilador. El cristal de cuarzo deberá estar dentro del rango establecido por el fabricante del microcontrolador. (3.5 MHz - 12 MHz.).

2. Selección del cristal mediante el diseño del reloj de tiempo real. Debido a que la frecuencia del oscilador es fija, será necesario conocer el número de ciclos de máquina necesarios para consumir un segundo.

El número de los ciclos deberá ser un entero, ya que de lo contrario el reloj será poco preciso y tenderá a atrasarse o adelantarse.

Para obtener el número de ciclos de máquina que se necesitan para consumir un segundo se efectúa el siguiente cálculo:

$$\# \text{ ciclos} = 1s \left(\frac{\text{FrecOsc}}{12} \right)$$

En la tabla que se presenta a continuación se observa que se seleccionan como posibles a utilizar aquellos cristales cuyas frecuencias cumplen con la premisa expuesta anteriormente.

Frecuencia de cristal de cuarzo (MHz)	Número de ciclos	Posible de utilizar
3.5795450	298295.42	
3.6864000	307200.00	✓
4.0000000	333333.33	
4.1943040	349525.33	
4.9152000	409600.00	✓
5.0000000	416666.67	
5.0688000	422400.00	✓
6.0000000	500000.00	✓
6.1440000	512000.00	✓
7.3728000	614400.00	✓
8.0000000	666666.67	
9.8304000	819200.00	✓
10.738635	894886.25	
11.000000	916666.67	
11.059200	921600.00	✓
12.000000	1000000.00	✓

3. Se debe considerar también la frecuencia de operación del convertidor analógico/digital ADC. (10 - 1280 KHz).

Otro parámetro para la selección del cristal de cuarzo es el considerar el rango del pulso de reloj del convertidor analógico/digital (10 - 1280 KHz, frecuencia típica de 640 kHz)

Debido a que el rango de cristales de cuarzo que se tiene hasta este momento es demasiado grande, se procederá a tomar la señal ALE en la terminal 30, la cual es 1/6 de la frecuencia del oscilador.

Frecuencia de cristal de cuarzo (MHz)	Frecuencia de la señal ALE (kHz)	Posible de utilizar
3.6864000	614.40	✓
4.9152000	819.20	✓
5.0688000	844.80	✓
6.0000000	1,000.00	✓
6.1440000	1,024.00	✓
7.3728000	1,228.80	✓
9.8304000	1,638.40	
11.059200	1,843.20	
12.000000	2,000.00	

4. Selección del cristal mediante la velocidad de transmisión del puerto serie. Es de gran utilidad que el cristal seleccionado permita la selección de varios "baud rates", ya que con esto proporciona mayor versatilidad en las aplicaciones.

Debido a que en el modo 1 de la transmisión serial el "baud rate" es generado por el temporizador 1 es importante que el valor de TH1 (valor de recarga para el temporizador 1 operando en modo 2) sea un número entero para el "baud rate" requerido, tal como se aprecia en la tabla mostrada a continuación:

$$TH1 = \frac{2^{SMOD} \text{ FREQ. DEL OSC.} - [(32)(BAUD RATE)(3072)]}{(-12)(32)(BAUD RATE)}$$

Cristal (MHz)	Bauds	TH1	Bauds	TH1	Bauds	TH1	Bauds	TH1	Bauds	TH1
3.68640	110		150	✓	300	✓	600	✓	1200	✓
4.91520	110		150		300		600		1200	
5.06880	110	✓	150	✓	300	✓	600	✓	1200	✓
6.00000	110		150		300		600		1200	
6.14400	110		150		300		600		1200	
7.37280	110		150	✓	300	✓	600	✓	1200	✓

Cristal (MHz)	Bauds	TH1	Bauds	TH1	Bauds	TH1	Bauds	TH1
3.68640	2400	✓	4800	✓	9600	✓	19200	
4.91520	2400		4800		9600		19200	
5.06880	2400		4800		9600		19200	
6.00000	2400		4800		9600		19200	
6.14400	2400		4800		9600		19200	
7.37280	2400	✓	4800	✓	9600	✓	19200	✓

De la tabla anterior se pueden seleccionar los cristales con los valores de 3.6864, 5.0688 y 7.3728 MHz, puesto que cumplen con valores enteros de TH1. Debido a que los cristales de 3.6864 y 7.3728 manejan mayor número de opciones en cuanto a "baud rates" se descarta el otro.

Se decidió utilizar el cristal de 3.6864 MHz ya que además de cumplir con todos los requisitos, proporcionará a través de la señal ALE del microcontrolador el valor de 614.4 KHz que es muy cercano al valor típico de 640 KHz requerido por el reloj del convertidor A/D.

En el Apéndice 2 se encuentra el diagrama final del circuito, el cual muestra los componentes utilizados así como sus conexiones.

3.2 DESARROLLO DE SOFTWARE

Antes de iniciar el desarrollo del programa es necesario entender el funcionamiento del sistema de adquisición de datos.

La tarjeta de adquisición muestrea las señales analógicas y lógicas durante un determinado periodo de muestreo constante configurable desde la computadora personal. El periodo de muestreo varía desde 1 hasta 15 segundos para todos los canales de entrada. Esto determina la utilización de un temporizador para determinar el periodo de muestreo.

La operación de muestreo se realizará en forma continua hasta que la computadora indique, a través de un mensaje enviado por el puerto serie, otra operación. La tarjeta almacenará los datos adquiridos en la memoria RAM. Además, al requerirse una comunicación serie entre microcontrolador y computadora personal se empleará el temporizador 1 para la generación del "baud rate".

Requerimientos de registros, banderas y memoria del microcontrolador

Ya que se empleará el puerto serie, se requiere que el temporizador 1 opere en modo 2 para la generación del "baud rate", es decir, en modo recargable. El valor de conteo se encuentra fijo en el registro TH1, el cual se carga cada vez que existe un sobreflujo.

Se requiere de un temporizador para producir los sobreflujos necesarios en la generación del reloj de tiempo real. Se empleará el temporizador 0 en modo 1 (temporizador de 16 bits). Serán necesarios por lo tanto los registros TH0 y TL0.

El modo de control del puerto serie será el modo 1, es decir, 1 bit de inicio, 8 bits de datos, 1 bit de paro y con una velocidad de transmisión o "baud rate" dado por el temporizador 1.

Se necesitan 8 bytes de la memoria RAM para poder almacenar los valores de las variables analógicas correspondientes en las direcciones respectivas y 2 bytes, para los valores de las variables lógicas.

Para poder lograr la comunicación con el convertidor analógico/digital se utilizarán los bits del puerto P0 para la comunicación de datos y el puerto P2 para el control y establecer las líneas de direccionamiento de las entradas analógicas del convertidor.

Como ya se mencionó anteriormente se requiere de dos banderas para determinar las siguientes tres formas de muestreo :

Muestreo de sólo variables lógicas

Muestreo de variables analógicas

Muestreo de variables analógicas y lógicas

Explicación general del programa

Se requiere una rutina principal la cual inicializará los registros, temporizadores, comunicación serial y banderas del programa. Posteriormente el programa entrará en un ciclo para la espera del envío de algún mensaje por parte de la computadora o del desbordamiento de los registros del temporizador 0.

Se activará la interrupción del puerto serie del microcontrolador para leer los mensajes de la computadora. Se utilizará una rutina de puerto serie la cual comenzará en el vector de direcciones 23H que corresponde a la dirección de la interrupción del puerto serie. Esta rutina leerá el mensaje de la computadora, lo decodificará y efectuará la instrucción.

Se activará la interrupción del temporizador 0 para determinar los sobreflujos del mismo temporizador con el objeto de llevar el control del tiempo real.

La rutina de tiempo real será llamada por la rutina de interrupción del temporizador 0. Esta llevará el control del reloj de 1 segundo. También llevará el control de los muestreos durante los periodos de muestreos determinados.

La rutina de adquisición de datos será llamada por la rutina de tiempo real si ya ha transcurrido el periodo de muestreo. Esta rutina efectuará el muestreo de variables analógicas y/o lógicas.

A continuación se presenta el diagrama de flujo general con las rutinas ya mencionadas.

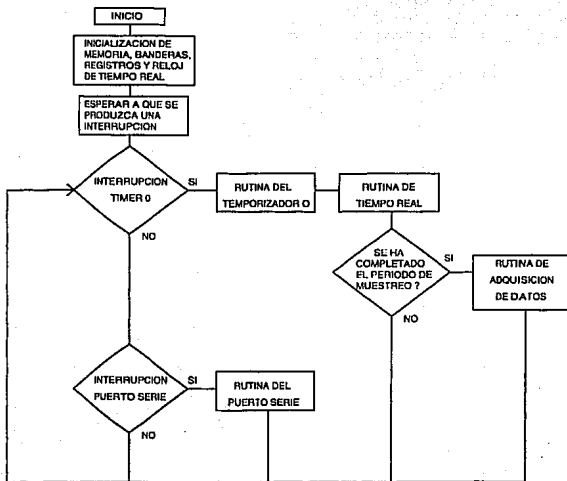


Figura 3.9 Diagrama de Flujo General

Mapa de memoria

Como ya se mencionó antes se requiere el uso de las siguientes áreas de memoria :

Dirección en RAM interna	Descripción
20H.1	Bandera. Si 20H.1=1, muestreo de "X" variables analógicas y lógicas. Si 20H.1=0, muestreo de "X" variables analógicas, con la condición de que F0 = 0.
F0	Bandera. Si F0 = 1, muestreo de sólo variables lógicas
30H	Mensaje enviado por la Computadora Personal
31H	"X" canales analógicos a muestrear
50H	Período de muestreo
120 - 127	Valores de las 8 Variables analógicas
118 - 119	Valores de las 16 variables lógicas

Mapa de bits

Bit	Descripción
P2.0, P2.1 y P2.2	Selección del canal a convertir en el A/D siendo P2.0 el LSB (A) y P2.2 MSB (C)
P2.3	Envía la señal de START y ALE al convertidor A/D.
P2.4	Envía la señal OE al convertidor A/D.
P2.5	Envía la señal EOC al convertidor A/D.

Descripción detallada del programa

Rutina Interrupción Serial

Esta rutina tiene por objeto leer el mensaje enviado por la computadora, decodificarlo y ejecutarlo.

Habiendo seleccionado el cristal de cuarzo de 3.6864 MHz se procede al cálculo del valor de recarga del temporizador 1.

Para llevar a cabo lo anterior es necesario primero seleccionar un "baud rate". Con el cristal seleccionado tenemos como posibles "baud rates" los siguientes: 150, 300, 600, 1200, 2400, 4800 y 9600. La única restricción que se tiene es enviar todos los datos a la computadora en menos de 1 segundo el cual es el límite inferior del periodo de muestreo. Ya que 1200 es una velocidad común en las computadoras personales y además de que cumple con lo anterior será el valor a utilizar.

$$TH1 = \frac{2^{MOD} \text{ FREQ. DEL OSC.} - [(32)(\text{BAUD RATE})(3072)]}{(-12)(32)(\text{BAUD RATE})}$$

$$TH1 = \frac{(2^0)(3686400 \text{ Hz}) - [(32)(1200 \text{ Hz})(3072)]}{(-12)(32)(1200 \text{ Hz})}$$

$$TH1 = 248$$

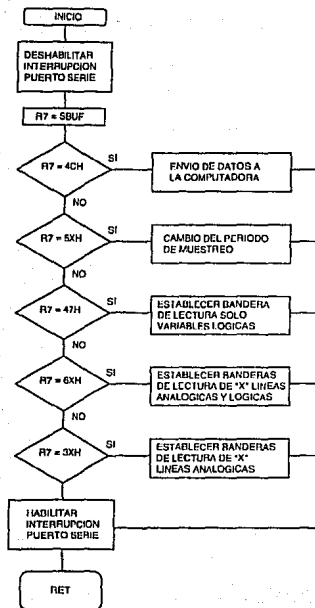


Figura 3.10 Rutina de Interrupción del Puerto Serie

Esta rutina de interrupción se inicia en la dirección 23H de la memoria de programa, deshabilitando la interrupción del puerto serie con el objeto de bloquear algún mensaje proveniente de la Computadora Personal.

Se guarda el valor de SBUF en R7 (mensaje proveniente de la Computadora Personal), se limpia RI (bandera de la interrupción de la recepción).

- ENVIO DE DATOS A LA COMPUTADORA PERSONAL. Se compara R7 con la constante #4CH, si es igual envía los datos muestreados a la Computadora Personal. Si es diferente salta al punto CAMBIO DEL PERIODO DE MUESTREO.

Si F0 es igual a 1, entonces se enviarán sólo las variables lógicas. De lo contrario se procederá al envío de "X" variables analógicas.

Antes de enviar las variables analógicas se carga R3 con el valor de "X" canales analógicos que se muestrearon y R0 con la dirección de la primer variable analógica.

Se entra a un ciclo para el envío de las "X" variables analógicas. Se carga el registro SBUF con la variable analógica, la dirección de este dato está en el registro R0, se incrementa R0 (R0 contendrá la dirección de la siguiente variable analógica a enviar).

ESTA TESIS NO DEBE SALIR DE LA BIBLIOTECA

A continuación se presenta un ciclo de espera para que el último bit del dato sea recibido por la PC. Para saber si la PC ha recibido el dato se apoyará en la bandera de transmisión. Si esta está en 1, significa que la PC ha recibido el último bit del dato; se limpiará esta bandera.

Se decrementará el registro R3 y brincaré el programa a la etiqueta de LOOP_TRANSM hasta que R3 valga 0. Nuevamente se carga el registro SBUF con la variable analógica por enviar (la dirección de esta variable está contenida en el registro R0).

Después de realizarse el envío de datos analógicos a la PC, se verificará la bandera 20H.1. Si 20H.1 es igual a 1, se enviarán las variables lógicas, si no saldrá inmediatamente de la rutina.

- **CAMBIO DEL PERIODO DE MUESTREO.** Se carga el valor de R7 en el acumulador y se realiza después un AND lógico de este último para obtener sólo los 4 bits más significativos y poner los bits menos significativos en 0. Si el resultado es igual a #50H se realizará lo siguiente. Si no el programa brincaré al siguiente punto.

El acumulador se cargará con el registro R7, se realizará un AND lógico del acumulador para obtener los 4 bits menos significativos y poner los 4 bits más significativos en 0, obteniéndose así el periodo de muestreo que se almacenará en la siguiente instrucción en la dirección 50H.

Posteriormente se cargará TL0 y TH0 con cero con el fin de inicializar el reloj, una vez cargado el valor del periodo de muestreo en la dirección 50H.

Se inicializa R6 (número de overflows) con 1 y R5 (número de segundos transcurridos) con el período de muestreo.

- ESTABLECER LECTURA DE SOLO VARIABLES LOGICAS. Se compara R7 con la constante #47H, si es igual pone en 1 a la bandera F0 (establece lectura de sólo variables lógicas). Si no es igual salta al siguiente punto.
- ESTABLECER LECTURA DE "X" LINEAS ANALOGICAS Y LOGICAS. Se carga el valor de R7 en el acumulador y se realiza después un AND lógico del acumulador para obtener sólo los 4 bits más significativos y poner los bits menos significativos en 0. Si es igual el resultado a #60H se realiza lo siguiente. Si no, el programa saltará al siguiente punto.

Se limpiará la bandera F0 (bandera de lectura de sólo variables lógicas).

Se pondrá en 1 el bit 20H.1 para establecer la lectura de "X" variables analógicas y lógicas.

Se guarda el valor de R7 en el acumulador, se realizará un AND lógico en el acumulador para obtener los 4 bits menos significativos y poner los 4 más significativos en 0. El valor del acumulador "X" se guardará en la dirección 31H.

- ESTABLECER LECTURA DE "X" LINEAS ANALOGICAS. Se carga el valor de R7 en el acumulador y se realiza después un AND lógico del acumulador para obtener sólo los 4 bits más significativos y poner los bits menos significativos en 0. Si es igual el resultado a #30H se realiza lo siguiente. Si no, termina la rutina de interrupción serial sin antes habilitar la interrupción del puerto serie nuevamente.

Se limpiará la bandera FO (bandera de lectura de sólo variables lógicas).

Se pondrá en 0 el bit 20H.1 para establecer la lectura de "X" variables analógicas.

Se guarda el valor de R7 en el acumulador, se realizará un AND lógico en el acumulador para obtener los 4 bits menos significativos y poner los 4 más significativos en 0. El valor del acumulador "X" se guardará en la dirección 31H.

Después se habilita la interrupción del puerto serie y finalmente la ejecución del programa continúa en la dirección de la instrucción siguiente al punto en el cual la interrupción fue detectada.

Rutina Inicio

Esta rutina tiene por objeto inicializar registros, banderas, temporizadores, comunicación serial, parámetros que se utilizarán en el resto del programa.

Además entra en un ciclo en espera de algún mensaje de la computadora o de la interrupción del temporizador 0.

Se iniciará con la habilitación de las Interrupciones del puerto serie y del temporizador 0. Se definirá el nivel de más alta prioridad a la interrupción del puerto serie. El temporizador 1 será utilizado en modo 2 (temporizador de 8 bits recargable) y el temporizador 0 estará en modo 1 (temporizador de 16 bits).

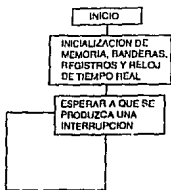


Figura 3.11 Rutina Inicio

Se configurará el puerto serie en el modo 1(UART 8 bits, frecuencia variable). Se activará la recepción por "software" a través de la bandera REN = 1.

Se establecerá el registro SMOD en 0 para no duplicar la velocidad del puerto serie cuando se utiliza el timer 1 en la generación del "baud rate".

Se inicializan los registros TH0 y TLO del temporizador 0 en ceros antes de habilitar al temporizador 0.

Se inicializan los registros TH1 y TL1 del temporizador 1 con el valor para la generación del "baud rate" ya establecido.

Limpia el pulso de OE (Output enable) del convertidor para que por omisión el microcontrolador direcciona a las 8 primeras líneas lógicas.

Se inicializa el período de muestreo y el registro R5 (número de segundos transcurridos) en 1 segundo.

Establece $F0 = 0$, $20H.1 = 1$ y $31H = \# 8$ para establecer el muestreo de 1-8 canales analógicos y lógicos

Se inicializa el registro R6 con el valor de 1 (número de desbordes) con el objeto de llevar el control del tiempo real de 1 segundo.

Se habilitarán los temporizadores 0 y 1

Y finalmente el programa entra a un ciclo en espera de algún mensaje de la computadora (interrupción puerto serie) o de la interrupción del temporizador 0.

Rutina Adquisición de Datos

Esta rutina tiene por objeto comparar las banderas F0 y 20H.1 para determinar las 3 formas de muestreo ya antes mencionadas y realizar el muestreo.

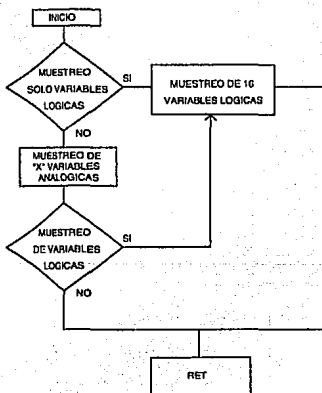


Figura 3.12 Diagrama de Flujo Rutina de Adquisición de Datos

Si se tiene la bandera F0 en "1" se lleva a cabo el muestreo de sólo las variables lógicas. Si se tiene la bandera F0 en "0" trabajará conjuntamente con la bandera 20H.1 de manera que si 20H.1 igual con "0", se ejecuta el muestreo de sólo "X" líneas analógicas, si 20H.1 igual con "1", se ejecuta el muestreo de las "X" líneas analógicas y las líneas lógicas. Ese valor "X" se encuentra almacenado en la dirección 31H

Antes de realizar la adquisición de las variables analógicas se requiere inicializar los registros R2, el cual contiene el valor de las "X" variables analógicas; R4 que inicializa las líneas de direccionamiento del convertidor ADC; enseguida, R1 guarda la dirección de la primera variable analógica.

En el comienzo del ciclo de la adquisición de variables analógicas se carga la dirección del convertidor en el puerto 2, posteriormente se pone en alto el pin P2.3 con el objeto de habilitar los pulsos START/ALE para el convertidor, después se pone en bajo el pin P2.3, enseguida se pone en alto el pin P2.5 para poder leer los datos del convertidor.

El programa entra en un ciclo de espera debido a que el convertidor requiere de un tiempo para la conversión. La señal del convertidor que determinará el poder leer ya el valor muestreado lo dará la señal EOC del convertidor hacia el pin P2.5.

Posteriormente el contenido del puerto 0 se guardará en la dirección que guarde R1 .

Se deshabilitará el pulso OE al convertidor a través del pin P2.4, se incrementará el registro R1 (R1 contendrá ahora la dirección de la siguiente variable analógica a guardar) y se incrementará R4 (R4 contendrá la dirección de la siguiente línea analógica del convertidor).

Se decrementará el registro R2 y saltará a la etiqueta repite_muestreo donde primeramente se carga la dirección del convertidor en el puerto 2 hasta que se haya concluido el muestreo de todas las "X" líneas analógicas.

Rutina Interrupción del Temporizador 0

Esta rutina es necesaria para llevar el control de las llamadas a la rutina de tiempo real cuando se presenten sobreflujos en el temporizador 0.

Esta rutina de interrupción inicia en la dirección 0BH de memoria de programa, deshabilitando la interrupción del puerto serie con el objeto de bloquear algún mensaje proveniente de la Computadora Personal.

Posteriormente se llama a la rutina de tiempo real; al finalizar ésta, se habilita la interrupción del puerto serie y finalmente la ejecución del programa continúa en la dirección de la instrucción siguiente al punto en el cual la interrupción fue detectada.

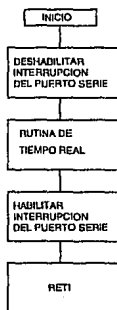


Figura 3.13 Diagrama de Flujo Rutina de Interrupción Temporizador 0

Rutina Tiempo Real

Esta rutina tiene la finalidad de llevar el control del reloj de tiempo real así como el de llamar a la rutina de muestreo al término del periodo de muestreo.

DISEÑO DEL RELOJ DE TIEMPO REAL

Para poder programar el reloj, se utilizará el contador/temporizador T0 que será programado como un temporizador de 16 bits.

Cuando el contador está programado para operar como temporizador de 16 bits, el registro de conteo (T0) se incrementa con cada ciclo de máquina, es decir, cada 12 ciclos de reloj.

Cada vez que el temporizador T0 excede su límite, volverá a cero y activará la bandera TF0. Si la bandera TF0 se encuentra activa y la interrupción de este contador está habilitada, el microcontrolador efectuará una llamada a una rutina de tiempo real que se encuentra en el vector de interrupciones correspondiente a la línea de interrupción 0BH.

Debido a que la frecuencia del oscilador es fija, es posible conocer el número de ciclos de máquina necesarios para consumir un segundo. Para ello se efectúa el siguiente cálculo:

$$\# \text{ ciclos} = 1s \left(\frac{\text{FrecOsc}}{12} \right)$$

El resultado de dicha operación deberá ser un entero, ya que de lo contrario el reloj será poco preciso y tenderá a atrasarse o adelantarse. Para que el resultado de esta operación sea un número entero, se seleccionó un cristal de 3.6864 MHz.

Una vez establecida la frecuencia de oscilación del cristal, se procede a calcular el tiempo que consume cada ciclo de máquina.

$$\text{Período}_{\text{ciclo de máquina}} = \frac{12}{3.6864 \times 10^6} = 3.2552 \times 10^{-6} \text{ s}$$

El contador programado como temporizador de 16 bits, generará una interrupción cada 65536 (2^{16}) ciclos de máquina. En base a esto, podemos calcular cuantas veces debemos entrar a la rutina de servicio del reloj para poder obtener un segundo. Este cálculo se realiza de la siguiente manera:

$$\# \text{ Desbordes} = \frac{307200}{65536} = 4.6875$$

Esto quiere decir que se requerirán 4 desbordes del temporizador y una fracción de 0.6875 veces el tiempo de sobreflujo para completar un segundo. El número de ciclos correspondiente a 0.6875 veces el régimen de desborde, se puede calcular de la siguiente manera:

$$\text{Ciclos adicionales} = 65536(0.6875) = 45056$$

Para poder contar fracciones del ciclo de sobreflujo del temporizador, se utiliza un procedimiento de recarga de los registros TH0 y TLO asociados a éste. Con este procedimiento se da un valor inicial a los registros de conteo, acortando el número de ciclos de máquina necesarios para alcanzar un desborde del temporizador.

En este caso, el valor de recarga de los registros de conteo se puede calcular de la siguiente forma:

$$V_{rec} = 65536 - 45056 = 20480 = 5000H$$

Por lo tanto:

$$TH0 = 50H = 32D$$

$$TLO = 00H = 00D$$

FUNCIONAMIENTO DE LA RUTINA DE TIEMPO REAL

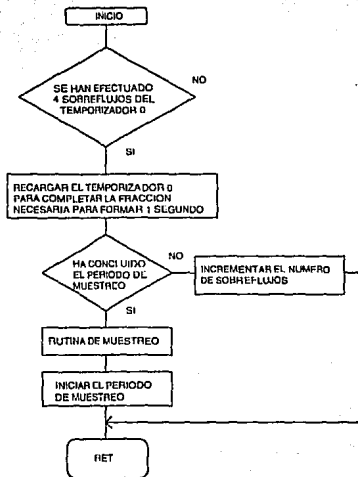


Figura 3.15 Rutina de Tiempo Real

La generación de la base de tiempo de un segundo consistirá en programar un temporizador de 16 bits, habilitar su interrupción y escribir en la dirección correspondiente del vector de interrupciones una rutina que permita el conteo de 4 ciclos de desborde completos y una recarga de contador para completar la fracción necesaria para formar 1 segundo. Si la rutina de tiempo real detecta que se han completado 5 ciclos incluyendo el ciclo que contiene la fracción correspondiente, se decrementará R5 (Período de Muestreo).

Para producir un segundo se requiere 4 desbordos del temporizador 0 en modo de 16 bits a partir de los valores TL0 = #00H y TH0=#00H. Y además de un quinto desborde pero con los valores de TL0 = #00H y TH0=#50H. El registro R5 contendrá el número de segundos que han transcurrido para que se produzca el período de muestreo (1 - 15 segundos).

Esta rutina comienza con la comparación del registro R6 con el valor #5 número de sobreflujos. Si es el quinto sobreflujo (R6 = 4) en presentarse entonces realiza lo siguiente, si no incrementa R6 y termina la rutina.

Carga TH0 con el valor de #50H y TL0 con el valor de #07H

Posteriormente se inicializa R6 con #1, ya que la próxima interrupción del timer 0 será el primer sobreflujo de 5 para producir un segundo.

Decrementa R5 y si no es igual a cero, termina la rutina de tiempo real. Si es cero (ya transcurrieron los "n" segundos para producir el periodo de muestreo) se llama a la rutina de muestreo. Después de realizarse el muestreo se carga el valor del período de muestreo (50H) en R5 y termina la rutina de tiempo real.

CAPITULO 4

INTERFAZ HOMBRE - MAQUINA

Debido a la rápida y constante evolución de los procesos, cada día es más necesario tener un eficiente control de dichos procesos, ya que pequeñas variaciones pueden ocasionar graves daños, así como costosas pérdidas. Con la versatilidad de una computadora se puede tener en ella toda la información que se requiera en un sólo punto en el mismo tiempo, cosa que reduce mucho los espacios requeridos para cuartos de control, así como de horas hombre de personal desplazándose de un lado a otro para poder obtener la información requerida, al mismo tiempo se gana mucho en precisión y tiempo de respuesta a una variación de alguna de las variables involucradas.

Después de evaluar diferentes paquetes comerciales se llegó a la definición de las opciones necesarias en pantalla para tener un manejo amigable y sencillo de las diferentes variables que se estarán visualizando mediante esta interfaz y su conexión con la Tarjeta de Adquisición de Datos .

La visualización de los Valores medidos para las variables seleccionadas se tendrá de forma gráfica en el monitor de una Computadora Personal ya que de esta manera la operación para el usuario es sencilla y amigable, además de tener centralizada toda la información. Debido a lo anterior, se deberá trabajar en un ambiente Windows ya que cuenta con gran versatilidad en el manejo de gráficas.

4.1 ESPECIFICACIONES

Dentro de la pantalla principal se podrán observar las gráficas de comportamiento de las 8 variables analógicas y de las 16 lógicas. En dicha pantalla se contará con tres ventanas, las cuales servirán para la manipulación de archivos y gráficas, así como para la comunicación con la tarjeta de adquisición tal como se muestra en la Figura 4.1.

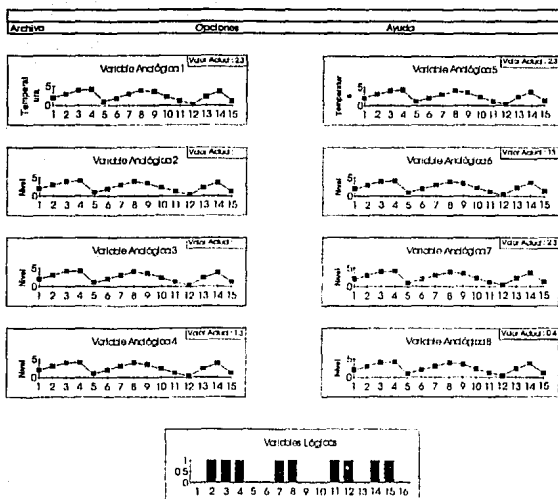


Figura 4.1 Pantalla Principal.

4.2 DESCRIPCION DE LA OPERACION.

Las tres ventanas contenidas dentro de la pantalla principal, así como las diferentes opciones que se presentan dentro de cada una de ellas se describen a continuación:

1. Una Ventana denominada **A**rchivo, la cual permitirá el manejo de los archivos consistente en:

- **N**uevo.- Con este comando se crea un nuevo archivo, es decir, a partir de ese momento y hasta que no se cree otro, éste será el encargado de almacenar los datos que estará enviando el microcontrolador.

- **A**brir.- Este permite traer a la pantalla algún archivo que ya se encuentre guardado en cualquier momento, ésto con el fin de hacer consultas posteriores del comportamiento durante el día, semana, etc. de alguna o todas las variables, según sea el caso.

- **G**uardar.- Por medio de esta instrucción se guarda en medio magnético el archivo tal y como se encuentre en el momento de la ejecución del comando.

- **I**mprimir.- Con esta opción se podrá mandar el archivo seleccionado a una impresora previamente definida.

- **S**alir.- Mediante la ejecución de esta instrucción se da término a la ejecución del programa, regresando a la pantalla previa de Windows.

2. Una ventana denominada **Opciones**, la cual permitirá la comunicación con la Tarjeta de Adquisición de Datos, es decir, peticiones de envío de Datos, cuales son las variables que se necesitan sean enviados sus valores, etc., como a continuación se verá con más detalle:

- **Envío de Datos.**- Con esta opción se mandará a través de puerto serial el equivalente a la letra L en hexadecimal "4C", con lo cual la Tarjeta de Adquisición de datos sabrá que se le está solicitando envíe los valores de las variables que tiene guardados en memoria.

- **Cambio del Período de Muestreo.**- Al seleccionar esta opción se abre una segunda opción, en la cual se pregunta el periodo deseado, limitando la respuesta al rango de 1 a 15 segundos, una vez proporcionado este valor la computadora enviará dicha instrucción a la Tarjeta.

- **Líneas Lógicas.**- Con esta opción se indicará que se desea obtener los datos de las variables lógicas única y exclusivamente.

- **X líneas Analógicas y Lógicas.**- Este comando muestra una segunda ventana en la cual se pregunta de cuantas variables analógicas se desean obtener datos, una vez proporcionado este dato la computadora manda la instrucción de obtener valores de las variables analógicas indicadas, así como de las lógicas.

- **X líneas analógicas.**- En esta opción al igual que en la anterior se debe alimentar a la computadora con el número de variables analógicas requeridas, con la diferencia que en ésta las lógicas no serán leídas.

- Envío de datos Automático.- Mediante este comando la computadora se encarga de pedir el envío de datos a la tarjeta de adquisición cada vez que se termine el ciclo indicado por el campo destinado al periodo de muestreo.

3. Una ventana asignada a la Ayuda del Usuario, en la cual se tiene una breve descripción de cada uno de los comandos y sus restricciones en caso de tenerlas.

Para el manejo de las pantallas la opción más viable es la utilización de un "mouse", tal y como se acostumbra dentro de las aplicaciones de Windows, y al igual que en este si no se dispone de uno de estos se podrá manejar a través del teclado por medio de la combinación de las teclas ALT y la letra subrayada de cada una de las tres opciones del menú principal, y una vez dentro de este moverse con las flechas hasta sombrear la deseada y oprimir ENTER o con la selección directa en teclado de la letra subrayada para cada uno de los submenús.

Es importante señalar que se tiene la posibilidad dentro de la pantalla principal de seleccionar alguna de las gráficas de cualquiera de las variables y mediante un doble ENTER con el "mouse" hacer que dicha gráfica se amplíe al tamaño de la pantalla (zoom).

Lo anteriormente mencionado se observa en la Figura 4.2 la cual muestra el diagrama general de la operación.

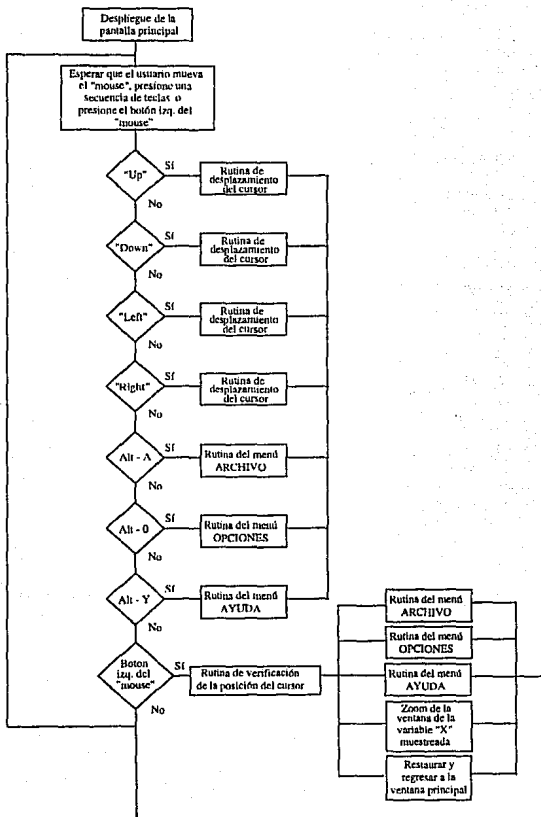


Figura 4.2 Diagrama de flujo general de la operación.

CONCLUSIONES

CONCLUSIONES

Con el presente trabajo se comprueba la gran y amplia utilidad que tienen los microcontroladores en el campo del desarrollo de proyectos que nos conducen a una mejoría tanto personal como material en muchos de los campos de acción con los que nos topamos día con día en nuestras vidas.

Se demuestra la versatilidad que se obtiene con diseños como el presentado ya que fácilmente se puede diseñar un sistema como el propuesto para aplicaciones específicas, dependiendo de las características deseadas, al mismo tiempo que un sistema ya implementado puede ser modificado o mejorado para la mayor satisfacción del usuario, siempre y cuando esto no represente cambios mayores.

Mediante la implementación del diseño planteado a través de los diferentes capítulos se obtendría:

- Reducción en los errores de lectura en la medición de las variables.
- Disminución en la cantidad de errores en las correcciones.
- Menor tiempo de respuesta para los ajustes en las variables en caso de ser necesarios.
- Reducción en la cantidad de personal dedicado a tomar lecturas.
- Personal más capacitado.
- Ganancia en la productividad.
- Minimización del área del cuarto de medición.

Es de gran importancia mencionar que para la exitosa implementación de un sistema como el propuesto es necesario sensibilizar a los operadores de las grandes ventajas, así como las mejoras que éste les va a proporcionar para que con esto y un buen plan de entrenamiento el aprovechamiento del mencionado sistema sea el óptimo.

El presente proyecto ha sido propuesto tomando como posibles aplicaciones procesos pequeños. Existe la posibilidad de formar una red de supervisión con varias tarjetas como estas conectadas a una misma computadora. Para esto, es posible utilizar otro tipo de interfaz serial como lo es la EIA-422A, o alguna otra similar.

La tendencia de proyectos de este tipo es el convertirse en sistemas de control, los cuales a su vez pueden interconectarse y así formar una red de control, con lo cual los beneficios son cada vez mayores.

APENDICE 1

\$ALLPUBLIC

JMP MAIN

.....
;
; RUTINA INTERRUPCION DEL TEMPORIZADOR 0
;

DEFSEG INTTIMER0,START=0BH,CLASS=CODE
SEG INTTIMER0

MOV IE,#82H ; Deshabilitar interrupción del puerto serie
ACALL TIEMPO_REAL ; Llamar subrutina tiempo real
MOV IE,#92H ; Habilitar interrupción puerto serie
RETI ; Salir de la interrupción temporizador 0

.....
;
; RUTINA INTERRUPCION SERIAL
;

DEFSEG INTSERIAL,START=23H,CLASS=CODE
SEG INTSERIAL

MOV IE,#82H ; Deshabilitar interrupción del puerto serie
MOV R7,SBUF ; Guardar el valor de SBUF en R7
CLR RI ; Limpiar bandera de recepción

CJNE R7,#4CH,SALTO_TRANSM ; Envío de datos
JB F0,ENVIAR_LOGICAS ; a la PC

	MOV	R3,31H	; R3 = X canales analógicos
	MOV	R0,#120	; R0 = Dirección 1er. variable ; analógica
LOOP_TRANSM:	MOV	SBUF,@R0	; Guardar variable analógica ; en SBUF
	INC	R0	; Incrementar dirección de las ; variables analógicas
RETARDO_TRANS1:	JB	TI,RETARDO_TRANS1	; Ciclo de espera envío de datos
	CLR	TI	; Limpiar bandera de transmisión
	DJNZ	R3,LOOP_TRANSM	; Decrementa X canales ; analógicos
	JB	20H.1,ENVIAR_LOGICAS	; Muestreo de líneas lógicas ; si 20H.1 = 1
	JMP	SALTO_NUM_ANALOG	
ENVIAR_LOGICAS:	MOV	SBUF,118	; Guardar la primera variable lógica ; en SBUF
RETARDO_TRANS2:	JB	TI,RETARDO_TRANS2	; Ciclo de espera envío de datos
	CLR	TI	; Limpiar bandera de transmisión
	MOV	SBUF,119	; Guardar la segunda variable ; lógica en SBUF

```

RETARDO_TRANS3:  JB      T1,RETARDO_TRANS3 ; Ciclo de espera envío de datos
                  CLR      TI              ; Limpiar bandera de transmisión
                  JMP      SALTO_NUM_ANALOG

```

.....

```

SALTO_TRANSM:   MOV      A,R7              ; Cambio del período de
                  ANL      A,#0F0H          ; de muestreo
                  CJNE     A,#50H,SALTO_PERIODO ;
                  MOV      A,R7              ;
                  ANL      A,#0FH          ;
                  MOV      50H,A           ; Guardar el período de
                                          ; muestreo
                  MOV      TL0,#00H        ; Inicializar registro TL0 del
                                          ; temporizador 0
                  MOV      TH0,#00H        ; Inicializar registro TH0 del
                                          ; temporizador 0
                  MOV      R6,#1H          ; Inicializar número de
                                          ; sobreflujos
                  MOV      R5,50H          ; Inicializar número de
                                          ; segundos transcurridos
                  JMP      SALTO_NUM_ANALOG ;

```

```

.....
SALTO_PERIODO:  CJNE      R7,#47H,SALTO_VAR_LOGICAL
                  SETB      F0                ;Establoc bandera de lectura de
                                                    ;sólo variables lógicas
                  JMP       SALTO_NUM_ANALOG

```

```

.....
SALTO_VAR_LOGICAL: MOV      A,R7                ; Establecer el muostroo
                  ANL      A,#0F0H          ; de X líneas analógicas
                  CJNE     A,#60H,SALTO_VAR_ANALOG ; y lógicas
                  CLR      F0                ; Limpiar bandera de sólo
                                                    ; lectura de variables lógicas
                  SETB     20H.1            ; Bandera 20H.1 (1) Lectura
                                                    ; de X variables analógicas
                                                    ; y lógicas
                  MOV      A,R7                ; Muestroar X
                  ANL      A,#0FH          ; canales
                  MOV      31H,A            ; analógicos
                  JMP      SALTO_NUM_ANALOG ;

```

```

.....
SALTO_VAR_ANALOG: MOV      A,R7      ; Establecer el muestreo
                   ANL      A,#0F0H   ; de X líneas analógicas
                   CJNE     A,#30H,SALTO_NUM_ANALOG
                   CLR      F0        ; Limpiar bandera lectura de variables lógicas
                   CLR      20H.1     ; Bnndera 20H.1 (0) Lectura de X variables
                                       ; analógicas

                   MOV      A,R7      ; Muestrear X
                   ANL      A,#0FH    ; canales
                   MOV      31H,A     ; analógicos

SALTO_NUM_ANALOG: MOV      IE,#92H   ; Habilitar interrupción puerto serio
                   RETI

```

.....
 INICIO


```

MAIN:      MOV      IE,#92H ; Registro de Habilitación de Interrupciones
          SETB    PS
          MOV     TMOD,#21H
          MOV     SCON,#50H
          MOV     PCON,#00H

          MOV     TH0,#00H
          MOV     TL0,#00H

          MOV     TH1,#0F8H
          MOV     TL1,#0F8H

          CLR     P2.4
          MOV     30H,#00H ; Dirección SBUFIN
          MOV     50H,#01H ; Dirección Período de Muestreo
          MOV     R5,50H ; Número de segundos transcurridos
          MOV     31H,#8 ; Muestrear 1-8 canales analógicos
          CLR     F0 ; Muestrear X canales analógicos
          SETB    20H.1 ; y lógicos
          MOV     R6,#1 ; Número de sobreflujos

          SETB    TR0 ; Habilitación del temporizador 0
          SETB    TR1 ; Habilitación del temporizador 1
  
```



```

LOOP_ADQUISICION:  NOP
                   AJMP    LOOP_ADQUISICION

```

```

.....
;.....
;.....  RUTINA  ADQUISICION DE DATOS  .....
;.....

```

```

MUESTREO:         JB      F0,MUESTREO_LOGICO ; Si F0=1 brinca a muestreo de
                   ; de variables lógicas
                   MOV    R2,31H           ; Guarda en R2 el valor de X
                   ; variables analógicas
                   MOV    R4,#00H         ; Inicializa las líneas del
                   ; convertidor
                   MOV    R1,#120        ; Guarda en R1 el valor de la
                   ; 1er. variable analógica
REPITE_MUESTREO:  MOV    P2,R4           ; Carga las direcciones del ADC
                   ; (ABC) en el puerto 2
                   SETB   P2.3           ; Habilita el pulso START/ALE
                   ; del ADC
                   NOP
                   NOP
                   CLR    P2.3           ; Deshabilita el pulso START/ALE (ADC)
                   SETB   P2.5           ; Poner en alto para leer EOC

```

ESPERA_TIME_CONV: NOP

JNB P2.5,ESPERA_TIME_CONV ; Ciclo para el tiempo de
; conversión

SETB P2.4 ; Habilitar Output Enable ADC
MOV @R1,P0 ; Carga el valor del puerto 0 a la
; dirección del registro 1

CLR P2.4 ; Deshabilita el pulso OE del ADC
INC R1 ; Incrementa el registro 1
INC R4 ; Incrementa el registro 4
DJNZ R2,REPITE_MUESTREO ; Decrementa el valor de X
; analógicas

JB 20H.1,MUESTREO_LOGICO ; Si 20H.1=1 realiza muestreo
; de variables lógicas

JMP TERMINA_MUESTREO ;

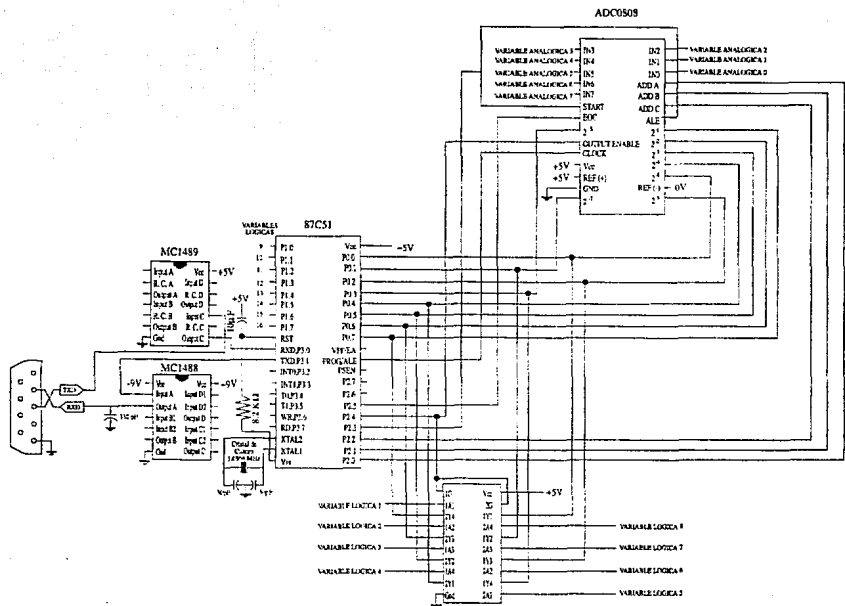
MUESTREO_LOGICO: **MOV** 118,P0 ; Guarda el valor del puerto 0 en la dirección 118
MOV 119,P1 ; Guarda el valor del puerto 1 en la dirección 119

TERMINA_MUESTREO: **RET** ; Fin de la subrutina de muestreo

.....
..... RUTINA TIEMPO REAL
.....

```
TIEMPO_REAL:  CJNE      R6,#4,SALTO_5OVERFLOW  
  
              MOV      TH0,#50H  
              MOV      TL0,#07H  
              MOV      R6,#1  
  
              DJNZ     R5,SALIDA  
  
              ACALL    MUESTREO  
              MOV      R5,50H      ; Periodo de muestreo  
              JMP      SALIDA  
  
SALTO_5OVERFLOW: INC      R6      ; INCREMENTA SOBREFLUJO  
  
SALIDA:      RET  
  
              END
```

APENDICE 2



74LS244

APENDICE 3

APENDICE 3

Simulador AVSIM51

El programa AVSIM51 se empleará como una herramienta de trabajo para el desarrollo del programa del microcontrolador ya que permite realizar la simulación de los microcontroladores de la familia 8051. Simula la arquitectura del CPU incluyendo los temporizadores, las interrupciones y los puertos de entrada/salida. También permite depurar fácilmente los programas.

El simulador despliega las condiciones de todos los registros, banderas, puertos y uso de áreas de memoria seleccionada durante la ejecución de los programas.

Las capacidad de depuración incluye el uso de puntos de interrupción dinámicos mediante condiciones, y ejecución paso a paso.

Se puede realizar la simulación directa de entradas y salidas al cambiar los valores de los puertos o se pueden usar archivos previamente creados que se cargan para simular las entradas al circuito. La salida del programa se puede salvar en un archivo para un análisis posterior.

La opción "Command Files" permite salvar secuencias de teclas para usarse en sesiones posteriores. Este comando se puede llamar en cualquier momento para establecer el ambiente de trabajo.

Operación del AVSIM51

Para ejecutar el AVSIM51 se teclea lo siguiente:

```
AVSIM51 [ -interruptor_opcional argumentos_opcionales ] RETURN
```

Los interruptores opcionales permiten optimizar el uso del AVSIM51 en computadoras que no sean compatibles con IBM. Los argumentos opcionales son una serie de comandos del AVSIM51 que ejecuta el simulador cuando el programa inicia.

Una vez que el simulador se carga, se deberá elegir el CPU a simular. Posteriormente aparecerá la pantalla del AVSIM51.

La versión 1.1 del AVSIM51 permite el uso de interruptores que optimizan su utilización en computadoras personales así como en ciertas configuraciones. Estos interruptores son los siguientes:

- -s inhibe el sonido del "beep".
- -i inhabilita las interrupciones 23H y 24H del sistema operativo. El interruptor 23H apagado evita la salida prematura del programa, el 24H permite un mayor control de los errores de disco no preparado o de protección de escritura.

- -d0 escribe sobre pantalla directamente . Esta opción incrementa la velocidad del simulador hasta en un 70%. Se usa únicamente en monitores CGA, monocromáticos o Hércules.
- -d1 escribe en la pantalla usando el ROM BIOS. Opción por default.
- -d2 escribe en la pantalla usando el ANSI.SYS. Esta opción se utiliza si se tienen problemas con el despliegue del AVSIM51.
- -k cambia el teclado para utilizar el SHIFT en lugar de CONTROL para enviar caracteres al AVSIM51.

Se pueden utilizar más de un interruptor a la vez.

Funcionamiento del AVSIM51

Cuando el AVSIM51 se ejecuta, simula en memoria todo el hardware del 8751, incluyendo los registros, puertos y memoria. Al realizar una instrucción se puede observar mientras se ejecuta, el estado del CPU y el efecto que se tiene sobre los registros, puertos y direcciones de memoria del microcontrolador simulado.

El AVSIM51 está provisto de un depurador simbólico en el que se pueden establecer puntos de interrupción o pase, ejecución paso a paso, comando "undo", y el examen y cambio de registros, memoria y banderas en cada operación. El AVSIM51 maneja la pantalla de despliegue como lo hace un editor

de texto, permitiendo el movimiento hacia casi cualquier parte del circuito y modificar su contenido, para posteriormente correr la modificación y ver su efecto. También se puede configurar para salvar automáticamente fragmentos de programa en un archivo, liberando al usuario de tener que recordar la modificación exacta que se realizó.

Con el fin de organizar las opciones disponibles, el AVSIM51 trabaja en dos modos:

- **MODO "DISPLAY"** : En modo "display", el simulador funciona como un editor de textos, es posible mover el cursor a cualquier parte del CPU, como lo son los registros, puertos o memoria y modificar el contenido de estas áreas.
- **MODO "COMMAND"** : En el modo "Command", se tiene acceso a un menú de comandos localizado en la parte inferior de la pantalla. Esto permite cargar archivos, examinar la memoria, introducir código de programa ("patch") etc.

Para realizar el intercambio de modos, se debe presionar la tecla de **ESCAPE**.

El simulador también está provisto de una serie de funciones que permiten controlar la manera de correr el programa. Estas funciones permiten establecer una operación a intervalos continuos o bien, paso a paso, deshacer operaciones, seleccionar la velocidad de simulación y las características de despliegue, establecer puntos de interrupción dinámicos, etc.

Pantalla del AVSIM51

La pantalla del AVSIM51 funciona como un "CPU visual", simula cada una de las características del CPU y despliega su contenido. Como resultado, se tiene una pantalla repleta de información, en la que se deben distinguir dos áreas importantes: una que posee la información de registros (1 en Figura A2.1) y aquella en la que se encuentran los valores del CPU (1 al 7 en Figura A2.1). Dentro de cada región, cada objeto, ya sea registro o pin, se trata como una ventana por separado. Con las teclas de movimiento de cursor se puede desplazar de una ventana a otra.

La pantalla del simulador se presenta en la Figura A2.1.

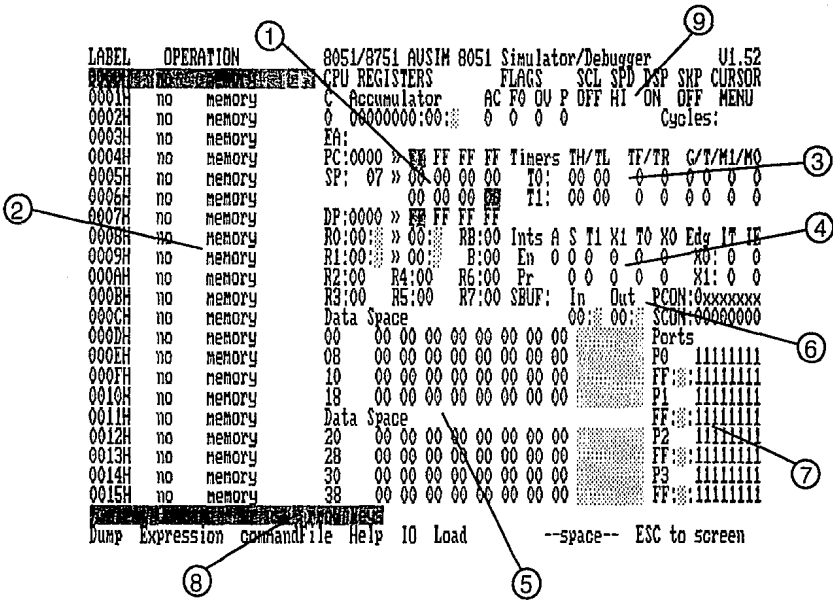


Figura A2.1 Pantalla del Simulador

Donde:

1. **REGISTROS Y BANDERAS.** Esta región despliega el contenido de cada registro asociado al CPU y presenta el estado de cada bandera en el CPU.
2. **PROGRAMA.** Esta área despliega la dirección, etiqueta, codificación y argumentos de la porción del programa que está siendo ejecutado. Mediante la combinación ALT + F5 se intercambia el despliegue para mostrar los símbolos o sus valores numéricos.
3. **TEMPORIZADORES.** Esta área muestra las características y el contenido de cada uno de los temporizadores.
4. **INTERRUPCIONES.** Esta área muestra las características de las interrupciones..
5. **ÁREAS DE MEMORIA.** Estas áreas se usan mediante el comando "Dump" para desplegar el contenido de memoria en hexadecimal y caracteres ASCII.
6. **PUERTO SERIAL.** Esta área muestra el contenido de los buffers de recepción y transmisión del puerto serial en hexadecimal y ASCII.

7. **PUERTOS.** Esta área muestra el contenido de cada uno de los puertos del CPU. Las características del "latch" se encuentran al lado del nombre del puerto. El contenido del puerto se presenta debajo de las características del "latch" hexadecimal, caracteres ASCII y bits.
8. **AREA DE COMANDOS.** Esta área despliega los menus utilizados para controlar el simulador. Se encuentra localizada en las dos últimas líneas de la pantalla.
9. **STATUS DEL SIMULADOR.** Esta área despliega el status del simulador como lo es la velocidad de simulación. El contador de ciclos es un pseudo-registro que se puede editar como cualquier otro registro. Incluso se pueden establecer puntos de interrupción en el contador. Para limpiar el contador, se debe mover al pseudo-registro de ciclos con CTRL Y y presionar CTRL HOME.

A continuación se describen las funciones de las teclas más importantes.

NOMBRE	TECLA	DESCRIPCIÓN
GO	F1	Inicia y termina la simulación del CPU.
BKUP	F2	Mueve el "breakpoint" una instrucción arriba en el programa.
BKST	F3	Establece un "breakpoint" dinámico en el lugar en que se encuentra el cursor.
BKDN	F4	Mueve el "breakpoint" una línea abajo.

SPD	F5	Selecciona la velocidad de ejecución de la simulación (HI, MEdium, LOW).
DSP	F6	Inicia y detiene la actualización continua del despliegue.
WMD	F7	Establece el tipo de ventana (BINaria, HEX o ASCII) a la cual se moverá el cursor cuando la ventana muestre datos en más de una base.
SKP	F8	Activa o desactiva (ON, OFF) el interruptor de salto en subrutinas. Si se esta operando en modo paso a paso y el interruptor está en ON no ejecutará la subrutina paso a paso. Cuando está en OFF sí la ejecutará paso a paso.
UNDO	F9	Deshace el último comando.
STEP	F10	Ejecuta una instrucción cada vez que se presiona esta tecla, es decir, ejecuta el programa paso a paso. No toma en cuenta los "breakpoints".
TRACE	ALT + F6	Activa o desactiva la actualización de las ventanas sin importar el estado del apuntador de la pila. (DSP).

SCLTOG	ALT + PgUp	Activa y desactiva el modo "Scroll". Cuando esta desactivado (OFF) las teclas de movimiento de cursor se mueven libremente en la pantalla. Cuando esta activado (ON), el cursor se mueve únicamente en la ventana actual.
--------	------------	---

BIBLIOGRAFIA

BIBLIOGRAFIA

GONZALEZ VAZQUEZ JOSE ADOLFO
INTRODUCCION A LOS MICROCONTROLADORES
McGRAW-HILL, 1992

HALL DOUGLAS V.
MICROPROCESSORS AND INTERFACING
PROGRAMMING AND HARDWARE
McGRAW-HILL, 1986

GILMORE CHARLES M.
MICROPROCESSORS PRINCIPLES AND APPLICATIONS
McGRAW-HILL PUBLISHING COMPANY, 1989

TOCCI RONALD
SISTEMAS DIGITALES
PRENTICE-HALL, 1987

APPLICATION NOTES FOR 80C51-BASED 8-BIT MICROCONTROLLERS
PHILIPS, 1993

AVMAC 8051 FAMILY USER'S MANUAL
AVOCET SYSTEMS, INC., 1988

**NATIONAL SEMICONDUCTOR CORPORATION
DATA ACQUISITION LINEAR DEVICES DATABOOK
CALIFORNIA, USA, 1989**

**DATA HANDBOOK 80C51-BASED 8-BIT MICROCONTROLLERS
PHILIPS, 1992**

**NATIONAL SEMICONDUCTOR CORPORATION
LS/S/TTL LOGIC DATABOOK
CALIFORNIA, USA, 1989**

**VEGA SALINAS ALEJANDRO
MANUAL Y APLICACIONES DEL MICROCONTROLADOR 8051
CENTRO DE INVESTIGACION Y DE ESTUDIOS AVANZADOS DEL IPN, 1993**

Manual ECS Inc. Cristales de Cuarzo

Manual CTS Microprocessors and Crystals

Manual Master Replacement Guide ECG Semiconductors

Revista DIGI-KEY Corporation