

29
19



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE INGENIERIA

**DISEÑO Y CONSTRUCCION DE UN
SISTEMA DE ADQUISICION DE
INFORMACION DIGITAL**

T E S I S

QUE PARA OBTENER EL TITULO DE
INGENIERO EN COMPUTACION
P R E S E N T A N :
MARIA IVONNE LOPEZ SOTELO
ARTURO VELEZ MORALES



Mexico, D. F.

DIRECTOR DE TESIS:
ING VICTOR ARMANDO CRUZ CEBALLOS

**EN
FALLA DE ORIGEN**

1991



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.

ANALISIS Y CONSTRUCCION DE UN SISTEMA PARA LA ADQUISICION DE INFORMACION DIGITAL.

INDICE

PAGINA	TEMA
1	INTRODUCCION
3	CAPITULO I - Panorama General
	I.1 Antecedentes.
	I.2 Planteamiento del problema
8	CAPITULO II - Analisis de Alternativas
	II.1 Estrategia de Solucion.
	II.2 Soluciones Alternativas.
	II.3 Propuesta de Solucion
21	CAPITULO III - Diseño y construcción del Hardware del Sistema.
	III.1 El microcontrolador 8751.
	III.2 Diseño del Módulo Principal
	III.2.1 El Computador de Flapo
	III.2.2 Accessal canal de Transmisión
	III.2.3 Alambrado del Módulo Principal

	III.3	Diseño de los Módulos Independientes
	III.3.1	El Procesador Principal
	III.3.2	Memoria de almacenamiento temporal
	III.3.3	Alambrado del módulo
36	CAPITULO IV	: Diseño e Implantación del Software del Sistema
	IV.1	Firmware para el microcontrolador
	IV.2	Software de comunicación entre el HOST y el CONTROLADOR
62	CAPITULO V	: Pruebas y Evaluación
66	CAPITULO VI	: Conclusiones
68	BIBLIOGRAFIA	

INTRODUCCION

El presente trabajo es el resultado de una constante búsqueda de modernización y actualización en los diferentes procedimientos y técnicas que tradicionalmente se llevan a cabo. Descansamos hacer un reconocimiento particular a una persona que a través de su coorte labor hace posible esa búsqueda. Nos referimos al Dr. Raúl Sierra Otero, quien desde un principio como es natural se nos brindó su apoyo, facilitando *coordinando* la elaboración de este proyecto. A él y a todo su equipo de trabajo queremos expresar nuestro más sincero agradecimiento.

La estructura del trabajo se ha dividido en seis capítulos. En el capítulo I, se presenta el panorama general de las necesidades, funciones y problemas operativos del Laboratorio de Control de Calidad de la Fabrica de Billetes del Banco de México.

En el Capítulo II, se expone un análisis de las alternativas y posibles opciones que surgen, así como una estrategia de solución. Además se plantea nuestra propuesta de solución, en la cual se basa todo el diseño y construcción del sistema.

Los capítulos III y IV describen el desarrollo de la etapa de diseño del Hardware del Sistema y del Software que permite la manipulación de la información.

Las pruebas y la evaluación del Sistema de Adquisición de Datos son contemplados en el Capítulo V.

Por último, el Capítulo VI expone las conclusiones establecidas al finalizar el trabajo.

CAPITULO I

PANORAMA GENERAL

1.1 Antecedentes

La Fábrica de Billetes es el organismo a través del cual el Banco de México imprime el papel moneda que manejamos cotidianamente. Como en todas las fábricas, el producto final debe cumplir con ciertas especificaciones mínimas de calidad. El uso frecuente y relativamente fuerte del billete, exige un control de calidad sumamente estricto.

El factor de principal interés en el proceso de control de calidad es el elemento medición, por esta razón, la Fábrica de Billetes cuenta con un laboratorio de control de calidad. Desde los inicios de este laboratorio se han establecido procesos específicos para llevar a cabo la cuantificación de parámetros que permitan la evaluación de las materias primas y del producto final.

El laboratorio de control de calidad cuenta con el equipo necesario para llevar a cabo su importante tarea. Anteriormente, algunas pruebas se realizaban con equipo mecánico e incluso manualmente, lo que provocaba falta de exactitud en los datos obtenidos. Actualmente el laboratorio ha sido equipado con instrumentos más modernos, exactos y confiables. La mayor parte del equipo proporciona una lectura en una pantalla digital e inclusive algunos de los equipos poseen capacidad para calcular datos de tipo estadístico, tales como promedios, desviaciones estándar, valores máximos y mínimos, etc. Facilitando con ello la anotación de las mediciones y haciendo más rápido el proceso de prueba.

Una vez obtenidos los resultados de las pruebas, los datos son anotados en un reporte mediante el cual se notifica la aceptación o el rechazo de los productos evaluados; cabe mencionar que la mayoría de las materias primas son de importación y que el tomar una decisión errónea implica un retraso en la producción, de aquí la necesidad de contar con un mecanismo eficiente, exacto y confiable para la obtención de resultados.

1.2 Planteamiento del Problema

El proceso que se sigue para elaborar un reporte consiste en realizar un conjunto de mediciones, anotar en una libreta los datos leídos y posteriormente manipular estos datos para presentar las conclusiones finales.

El laboratorio dispone de una computadora, la cual es empleada para el almacenamiento de los resultados obtenidos de las pruebas que se realizan en los equipos de medición, y por medio de una hoja de cálculo electrónica, se elaboran los reportes finales.

A continuación presentamos una breve descripción de algunos de los equipos utilizados en el laboratorio, y sus características propias:

A) Medidor de espesor de papel

También conocido como "Micrómetro", este equipo proporciona lecturas en milímetros del espesor de la muestra deseada.

- Conector RS232
- Plug D125 (fembra)
- Velocidad de transmisión: 9600 [bauds]
- Tamaño de la palabra: 8 [bits]
- Señal de paro: 2 [bits]
- Formato de la información: ASCII

B) DIGITIZER

Este equipo es un medidor de resistencia al rasgado, que proporciona una lectura en kilogramos igual a la fuerza de oposición por parte de la muestra.

- Conector RS232
- Plug DB25 (Hembra)
- Velocidad de transmisión: 9600 [bauds]
- Tamaño de la palabra: 8 [bits]
- Señal de paro: 2 [bits]
- Formato de la información: ASCII

C) Medidor de tensión

Este equipo es de los más completos, pues cuenta con una computadora dedicada e incluye un monitor, un disco duro y un sencillo teclado de comandos para ser operado. Los resultados de la prueba se presentan en el monitor y muestran la tensión en kilogramos soportada por el papel.

- Conector RS232
- Plug DB25 (Macho)
- Velocidad de transmisión: Programable
- Tamaño de la palabra: 8 [bits]
- Señal de paro: 2 [bits]
- Formato de la información: ASCII

Aunque actualmente los equipos mencionados anteriormente son los únicos que poseen una salida digital, cabe aclarar que se desea adquirir otros cuatro equipos con características similares a corto plazo, con lo que se elevará la cantidad de equipos a un número de siete.

Se puede observar que sólo se han modernizado las herramientas para la adquisición y la manipulación de los resultados, no así la transferencia de la información desde que es generada hasta que es capturada en la computadora, ya que se sigue empleando el registro manual como medio de almacenamiento temporal. Este procedimiento presenta algunos inconvenientes tales como:

- Pérdida de tiempo por parte del personal, tiempo que podría ser utilizado en el aumento de eventos de la muestra (por ejemplo:

- Errores de lectura
- Errores durante la captura de información
- Pérdida de la información
- Mezcla de información de muestras diferentes

Esta serie de retrasos e imprecisitudes conduce a la elaboración de reportes susceptibles de fallas, y por consiguiente, a posibles decisiones equivocadas, con sus conducentes implicaciones económicas.

CAPITULO II

ANALISIS DE ALTERNATIVAS

II.1 Estrategia de Solución.

Los instrumentos que se emplean en el laboratorio proporcionan además de las ventajas mencionadas anteriormente, la posibilidad de comunicarse a una computadora a través de un conector de comunicación serie en el estándar RS232. Como se habia mencionado anteriormente, el número total de equipos que actualmente poseen estas características es de tres, sin embargo se desea adquirir posteriormente más equipos y se calcula que en un corto plazo sean un mínimo de siete.

Aprovechando esta característica se puede solucionar el problema de transcribir los datos a una libreta y después a la computadora. Para lo tanto, se propone la utilización de un sistema para comunicar a través de un cable, el instrumento de medición con la computadora y utilizar un papelín de comunicación obteniendo de esta manera, los datos directamente en la PC.

Existe en el mercado una serie de dispositivos que ofrecen la posibilidad de conectar no sólo uno, sino varios equipos a una misma computadora, y que se pueden adaptar a los instrumentos de medición, con lo cual se logra un perfecto control en la transferencia de información.

Por otro lado, existe la posibilidad de desarrollar un sistema que cumpla con los requerimientos específicos de los equipos de medición y satisfaga las condiciones que plantea el problema.

H.2 Soluciones Alternativas.

Alternativas Comerciales.

A) Tarjeta Multipuerto

Esta tarjeta puede ser instalada en la computadora, permitiendo la expansión del número de puertos serie disponibles. Esta tarjeta ofrece de cuatro a ocho puertos serie y un microcontrolador. Es relativamente fácil de conseguir e instalar. Se debe desarrollar un programa que realice el muestreo de los equipos que se encuentren trabajando en ese momento.

sin embargo, esta tarjeta utiliza un "slot" de la computadora por cada cuatro equipos. Considerando que el número de slots de expansión en la computadora destinada para este uso es limitado, el número de equipos que se puede conectar también lo es.

B) Dispositivo Multiplexor de Información Serial

Existe una línea de dispositivos que permite conectar información entre varios equipos que posean puerto paralelo o serie.

Entre sus características más importantes es que permiten la comunicación automática de entre cinco y diez canales, cuenta con una cola de espera para la atención de peticiones. Es fácil de conseguir e instalar, pero requiere de software especial para la transmisión serie de información. Otra gran ventaja es que necesita protocolo de comunicación, y como consecuencia sólo puede atender un equipo de medición a la vez.

C) LABWINDOWS (Software para adquisición de datos)

Por medio del puerto serie y utilizando un multiplexor de RS-232, es posible realizar la adquisición de los datos. La manipulación estadística así como la presentación gráfica de la información es realizada por medio de un paquete conocido como LAB WINDOWS (NAT90). Este paquete es fácil de conseguir e instalar, además de que presenta la interfaz de manera clara y fácil de manejar. De la misma manera que la opción anterior, permite atender solo un equipo a la vez, y en caso de expansión se requiere la instalación de tarjetas adicionales.

Alternativas de Desarrollo.

A) Dispositivo de adquisición de datos controlado con un microprocesador.

Este sistema consiste en un microprocesador que monitorea cada uno de los puertos serie correspondientes a los equipos de medición.

Para su análisis tomemos el siguiente ejemplo :

Se tienen cuatro líneas de comunicación que transmiten a 9.6 KHz cada una. A su vez, cada una de estas líneas se asigna a cada equipo de medición. De acuerdo con el teorema de Nyquist, la frecuencia de muestreo deberá ser de por lo menos el doble de la máxima frecuencia de las señales a muestrear. En este caso la frecuencia de muestreo debe ser de por lo menos 19.2 KHz, es decir un período o tiempo máximo de muestreo de 0.0520833 ms.

Supongamos que utilizamos un microprocesador trabajando a 2 MHz, es decir, que tendrá ciclos de reloj de 0.0005 ms. Cada instrucción que realiza el microprocesador necesita entre 3 y 24 ciclos de reloj dependiendo de la operación.

Si dividimos el tiempo necesario para realizar el muestreo entre el tiempo de un ciclo de reloj, tendremos el total de ciclos de reloj máximos utilizados para el muestreo. Esto es :

$$\text{Max Num de Ciclos} = \frac{0.052083}{0.0005} = 104 \text{ clocks}$$

Supongamos ahora que utilizamos un microprocesador Z80 trabajando a 2 MHz. El algoritmo para realizar polling sobre 4 puertos, guardar el dato en el stack junto con un byte de identificación y transmitirlo a la computadora principal sin protocolo de comunicación, sería el siguiente :

PROGRAMA EJEMPLO PARA POLLING DE HASTA 7 PUERTOS Y UN PUERTO PARA PC

INSTRUCCION	CLOCKS
UD SP, 0711H	
BEGIN : LD A, (15451)	13
LD B, A	4
LD C, 01H	7
CHECA : IN A, (C)	11
AND 02H	4
JR NZ, LEEDATO	7 (Si continua)
	12 (Si salta)
CONT1 : INC C	4
INC C	4
DJNZ CHECA	8 (Si B = 0)
	13 (Si B <> 0)
JR SENDPC	12
LEEDATO : LD C, C	4
LD B, B	4
IN B, (C)	11
PUSH BC	11
LD B, D	4
JR CONT1	12
SENDPC : LD HL, 0F8011H	10
ADD HL, SP	11
JR Z, BEGIN	7 (Si continua)
	12 (Si salta)
IN A, (01H)	11
AND 81H	4
CP 81H	4
JR NZ, BEGIN	7 (Si continua)
	12 (Si salta)
PUSH BC	11
LD A, B	4
OUT (01H), A	11
LD B, 0FH	7
SIGUI : IN A, (01H)	11
AND 81H	4
CP 81H	4
JR Z, CONT2	7 (Si continua)
	12 (Si salta)
DJNZ SIGUI	8 (Si B = 0)
	13 (Si B <> 0)
	4
CONT2 : LD A, C	11
OUT (01H), A	11
JR BEGIN	12

Análisis :

Primer Caso : Se muestrean por Polling cuatro canales y no existió dato en ninguno de ellos

Número de Clocks necesarios : 236

Segundo Caso : Se muestrean por Polling cuatro canales y se encontraron datos en todos

Número de Clocks necesarios : 540

De acuerdo al análisis anterior y a la siguiente tabla se concluye que :

Velocidad del micro (MHz)	No. de ciclos mínimo	
	Z 80	8088
2	104	90
4	208	180
6	312	270
8	416	360
10	520	450
12	624	540

En la tabla se muestra una comparación entre los microprocesadores Z80 y 8088.

Para sensar por medio de Polling el caso crítico, es necesario utilizar un microprocesador que trabaje a 12 MHz, sin posibilidad de expansión, ya que la velocidad de operación debería aumentar proporcionalmente. Ninguno de los dos microprocesadores propuestos es capaz de funcionar a esta frecuencia, cualquier microprocesador que cumpla con esta restricción requiere de un diseño más complejo.

B) Sistema de adquisición de datos por multiprocesamiento

En este caso se trata de utilizar un procesador por cada dispositivo de medición, que permita trabajar independientemente de la demanda de los otros equipos.

Para este caso es posible diseñar un sistema que incluya un procesador maestro encargado de sincronizar las tareas de los demás procesadores, así como de comunicarse con la computadora principal estableciendo un protocolo de comunicación *predefinido* y asíncrono. Este sistema permite gran posibilidad de expansión debido a que la frecuencia de muestreo es independiente de la frecuencia del sistema. Sin embargo, el diseño se torna un poco más complejo.

11.3 Propuesta de Solución

La factibilidad de cada alternativa de solución propuesta esta establecida por un breve análisis de las restricciones de la estrategia de solución. Estas restricciones establecen las fronteras del espacio de soluciones.

- Es importante hacer notar que los equipos de medición a pesar de que proporcionan la posibilidad de comunicarse via puerto serie (RS232), no permiten el establecimiento de un protocolo de comunicación (ROSS90) es decir, únicamente se limitan a enviar los datos sin atender ninguna señal externa.

- Otra importante limitación es que se requiere la atención simultanea de varios equipos de medición. Ningun equipo deberá monopolizar la atención de la microcomputadora, pues los datos obtenidos por los equipos restantes se perderían.

De acuerdo a lo anterior, podemos descartar las alternativas comerciales, pues algunas de sus requerimientos para funcionar están limitados por las características propias de los equipos.

Con respecto a las alternativas de desarrollo, el sistema de adquisición de datos por multiprocesamiento es el único que además de cumplir con los requerimientos mencionados, proporciona la flexibilidad necesaria para la expansión de capacidad a medida que se vuelve necesaria.

El sistema de adquisición de datos por multiprocesamiento es una alternativa de desarrollo que surge como respuesta a requerimientos anteriormente detectados, y que a saber son : el control y la manipulación de los datos que reportan un conjunto de equipos en los cuales se llevan a cabo diferentes mediciones

Para explicar claramente la propuesta de solución, analicemos el esquema que se muestra en la figura 2.1

A) Una computadora con un puerto serie disponible, que dentro de nuestro análisis conoceremos con el nombre de HOST.

B) Un CONTROLADOR que tiene un puerto serie para comunicarse con el HOST y hasta 7 puertos serie, para recibir información de los EQUIPOS DE MEDICION

C) EQUIPO DE MEDICION, que posea puerto serie para enviar el resultado de las mediciones a través de éste. Cabe aclarar en este punto que los equipos pueden enviar información en cualquier momento e incluso simultáneamente

D) Un CANAL DE COMUNICACION, que permita la transferencia de información entre el HOST y el CONTROLADOR, así como entre éste y los EQUIPOS DE MEDICION

Internamente el controlador consta de las siguientes partes, como lo muestra la figura 2.2

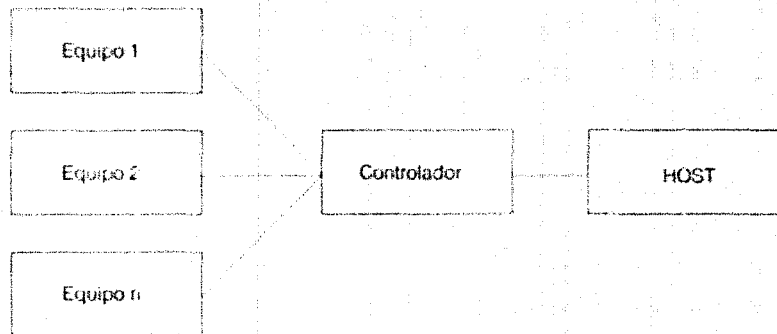


Figura 2.1

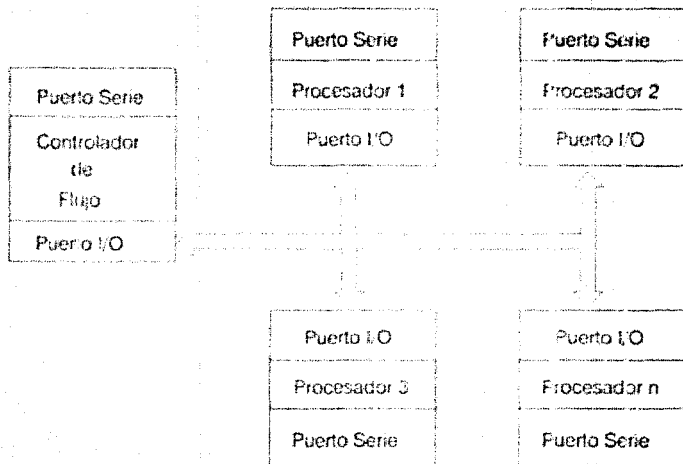


Figura 2.2

Cada procesador P_i se encarga de un equipo de medición en particular por medio de un puerto serie, de tal manera que este esquema de multiprocesamiento es capaz de trabajar en paralelo y atender simultáneamente a todos los equipos.

A su vez, cada procesador P_i posee un puerto de E/O para poder comunicarse con el controlador de flujo e indicarle que tiene información para transmitir hacia el HOST.

El controlador de flujo recibe información del procesador P_i que está listo para transmitir. Este controlador indica al procesador un controlador de flujo conectado con el HOST para establecer un CIRCUITO VIRTUAL entre este y el procesador P_i . Ya que existe una sola línea entre la computadora y el controlador para la transmisión y una sola para la recepción, constituyéndose así el canal de comunicación.

Es necesario hacer la comunicación entre el HOST y el procesador P_i por medio del controlador, ya que la computadora debe responder con una señal de aceptación. Esta señal no la puede recibir el procesador directamente, pues su línea de recepción se encuentra ocupada con el equipo de medición.

Una vez establecido el circuito virtual, el controlador le indica al procesador que el canal de comunicación está libre y asignado a él. En este momento el procesador es capaz de recibir y transmitir información proveniente del equipo de medición.

Es importante hacer notar que desde que se detecta la presencia de un dato en el puerto serie del procesador P_i , hasta que se le concede el acceso al canal de comunicación, el procesador necesita un medio de almacenamiento, una memoria RAM, para no perder los datos que van llegando, provenientes del equipo. La capacidad de ésta memoria está en función del tiempo que transcurre entre la petición y la asignación del canal de transmisión.

Como existen varios procesadores que necesitan tener acceso al canal de comunicación, es preciso seleccionar las líneas de transmisión de cada puerto serie, perteneciente a los procesadores. La figura 2.3 ilustra esta situación.

Aquí se muestra como es posible la comunicación directa entre cualquier procesador P_i y el HOST. El controlador de flujo selecciona el canal correspondiente en el selector.

Cabe hacer notar que el canal 0 se encuentra reservado para el controlador de flujo, pues como se había mencionado, este controlador necesita establecer el circuito virtual antes de permitir el acceso al canal de transmisión.

El controlador de flujo se comunica con los procesadores por medio de puertos de I/O como se muestra en la figura 2.4.

El controlador de flujo posee un puerto asignado a la recepción de una línea proveniente de cada procesador. Esto se puede representar con la figura 2.5.

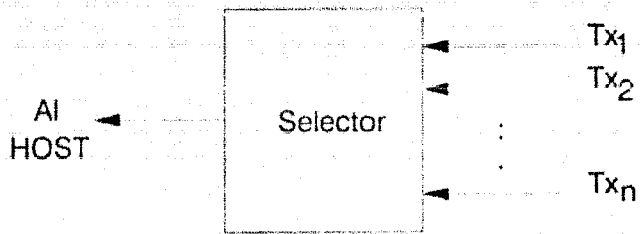


Figura 2.3

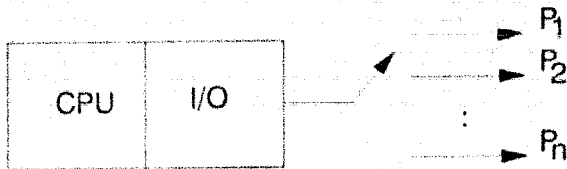


Figura 2.4

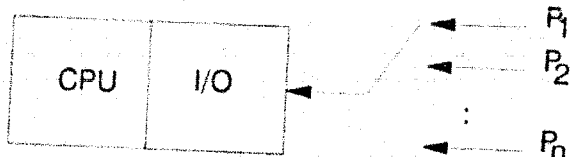


Figura 2.5

De tal forma que al revisar el puerto, un "1" lógico en el canal i , representaría que el procesador P_i está solicitando acceso al canal de transmisión. En caso de que dos o más procesadores soliciten acceso al mismo tiempo, se realizará un "polling" por prioridades, teniendo la prioridad más alta el procesador P_1 y la más baja el procesador P^N . El estado normal del puerto será cuando existan "0's" lógicos. Cuando el procesador atendido terminó su rutina de transmisión, envía un "0" lógico para que alguien más pueda acceder el canal. El controlador de flujo en ese momento establece el circuito virtual y concede el acceso al siguiente procesador en turno.

Por otro lado, el controlador de flujo posee la capacidad de recibir instrucciones provenientes del BOSC. Por ejemplo, si el BOSC necesita conocer el contenido de la memoria de un procesador P_i , instruye al controlador para que ejecute una interrupción sobre dicho procesador, el cual tendrá asignada una rutina de servicio que ejecute esta operación.

CAPITULO III

DISEÑO Y CONSTRUCCION DEL HARDWARE DEL SISTEMA

El hardware del sistema consiste en una serie de microcontroladores y circuitería externa que cumplen con todos los requerimientos planteados anteriormente. La familia MCS-51 de INTEL fue seleccionada por diversos motivos, pero principalmente debido al potencial que ofrece y a su fácil manejo.

En este tema se hablara brevemente de esta familia de INTEL, así como de las partes en las que se divide el sistema en cuanto a Hardware.

III.1 La Familia MCS-51 de Intel

¿Por qué un microcontrolador (BYT88)?

Es importante establecer la diferencia que existe entre una microcomputadora, un microprocesador y un microcontrolador.

Un MICROPROCESADOR es exclusivamente el CPU de una computadora, sin contar la memoria, Entrada/Salida, y los periféricos necesarios para un sistema completo. Por ejemplo, el 8088 ó el 80286 son microprocesadores. El prefijo "micro" indica que este elemento (CPU) se encuentra a nivel chip. Cualquier otro chip en una IBM PC se encuentra ahí para proveer otras características no incluidas en el microprocesador.

Cuando el microprocesador es combinado con IO y memoria, así como periféricos, esta combinación es conocida como MICROCOMPUTADORA. Muchas compañías añaden estas funciones periféricas al CPU dentro del mismo sustrato, creando una microcomputadora en un sólo chip.

Generalmente hablando, los chips de microcomputadoras son diseñados para dispositivos específicos que no necesitan todas las funciones de un sistema de computadora completo.

En aplicaciones de control donde el costo resulta elevado, aun los pocos chips que son necesarios para soportar un CPU como el 8088 ó el Z80 son diseñados. En estos casos, el diseñador utiliza una microcomputadora de un sólo chip para manejar situaciones particulares de control.

Cuando las microcomputadoras de un sólo chip son diseñadas o usadas en sistemas de control industrial, son normalmente conocidas como MICROCONTROLADORES. Básicamente, no existe diferencia entre las microcomputadoras y los microcontroladores, el nombre depende de como son usados.

La familia MCS-51 (INTEL) es un grupo de microcontroladores o microcomputadoras que cuenta con CPU, puertos paralelos de entrada/salida (I/O), oscilador, contadores / temporizadores (counter/timers), puerto serie, memoria de acceso aleatorio (RAM) y memoria de solo lectura ROM o EPROM, dependiendo de la versión del circuito (INT80). Los principales miembros de esta familia y sus características se presentan a continuación:

Dispositivo	Tecnología	ROM (bytes)	EPROM (bytes)	KAM (bytes)
8753H	NMOS	-	8 k	128
8053AH	NMOS	8 k	-	128
8751H	NMOS	-	4 k	128
8051AH	NMOS	4 k	-	128
8031AH	NMOS	-	-	128
80C51	CMOS	4 k	-	128
80C31	CMOS	-	-	128
80C52	CMOS	8 k	-	256
80C32	CMOS	-	-	256

Como se puede observar, la principal diferencia entre miembros de la misma familia es la cantidad de memoria de solo lectura (ROM) que tiene cada controlador, por ejemplo, el 8031 es la versión básica que contiene todos los elementos mencionados anteriormente con excepción de ROM, el 8051 es la versión del 8031 que contiene 4 kbytes de ROM, esta memoria se graba y se "enmascara" cuando el circuito es manufacturado. Una ROM de "mascara" es aquella que se solicita de antemano y se graba junto con la construcción del circuito; de tal manera que estos circuitos son mucho más baratos pues se construyen por cantidades industriales, tomando siempre en cuenta que el programa a grabar es una versión definitiva de lo que se requiere en el diseño original. En el 8751 la ROM se sustituye por una EPROM, con lo cual se da la ventaja de programar al dispositivo según convenga al diseñador y reprogramarlo si así se desea.

III 1.1 Arquitectura Interna

Como se mencionó anteriormente, un microcontrolador es una microcomputadora que está conformado por un CPU y circuitos periféricos. La estructura interna, en diagrama de bloques, es la que se muestra en la figura 3.1

Cada bloque puede representar uno o más circuitos integrados y cualesquiera componentes adicionales requeridos para hacer operaciones con estos circuitos.

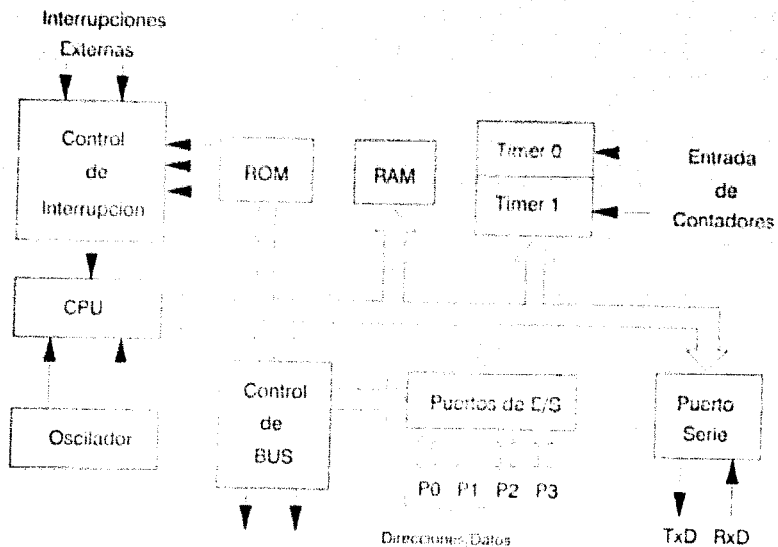


Figura 3.1 Estructura Interna del MCS-51

MEMORIA

El bloque de memoria contiene circuitos que permiten almacenar y recuperar los resultados de las operaciones efectuadas por el microcontrolador. También contiene la secuencia de instrucciones para efectuar dichas operaciones.

Todos los dispositivos de la familia M0 S-51 (8051) manejan sus operaciones en tres espacios de memoria. Estos espacios direccionables son: hasta 64 Kbytes de memoria de programa, 64 Kbytes de memoria externa de datos y 128 bytes de memoria de datos interna. De el espacio de memoria de programa los primeros cuatro u ocho Kbytes puede estar dentro del chip (en las versiones 8051 y 8054 respectivamente) esta memoria de programa solo se puede leer y se denomina como ROM.

El espacio de memoria de datos interna esta dividida en dos partes iguales de 128 bytes cada una. Una de esta partes es precisamente el espacio de memoria de datos de acceso aleatorio (RAM), cualquier dato almacenado en esta memoria, podrá ser accedido inmediatamente por el microcontrolador, el otro espacio esta ocupado por lo que se denominan Registros de Funciones Especiales (SFRs).

Los registros de función especial son :

SIMBOLO	NOMBRE	DIRECCION
• ACC	Acumulador	0E0H
• B	Registro B	0F0H
• PSW	Program Status Word	0F30H
SP	Stack Pointer	81H
DPTR	Data Pointer	
DPL	Low Byte	82H
DPH	High Byte	83H
• P0	Puerto 0	80H
• P1	Puerto 1	90H
• P2	Puerto 2	0A0H
• P3	Puerto 3	0B0H
• IP	Interrupt Priority	0B8H
• IE	Interrupt Enable	0A8H
TMOD	Timer Mode Control	89H
• TCON	Timer Control	88H
TH0	Timer 0 High Byte	8CH
TL0	Timer 0 Low Byte	8AH
TH1	Timer 1 High Byte	8DH
TL1	Timer 1 Low Byte	8BH
• SCON	Serial Control	98H
SBUF	Serial Data Buffer	99H
PCON	Power Control	87H

Aquellos registros indicados con un asterisco (*), son direccionables por bit, teniendo cada bit un significado y una dirección especial.

Los primeros treinta y dos bytes de RAM se dividen en cuatro bloques de ocho registros cada uno, y cada uno de estos registros se denominan desde R0 hasta R7, esto para permitir un uso más eficiente del espacio de código para programar.

En el espacio de SRAM se encuentran las palabras de control y de estado de los puertos de entrada/salida (I/O), puertos seriales, así como el control del sistema de interrupciones y programación de los modos en que operaran los timers/counters.

La figura 3.2 muestra el mapa de memoria que utilizan los microcontroladores de la familia MCS-51.

PUERTOS

El set de instrucciones de la familia MCS-51 permite manipular las treinta y dos líneas de entrada/salida como treinta y dos bits direccionables individualmente o como cuatro puertos de ocho bits cada uno, direccionables como P0, P1, P2 y P3.

Los puertos P0, P2 y P3 pueden asumir varias funciones. El P0 puede funcionar como transmisor del byte menos significativo de la dirección y como bus de datos, cuando el microcontrolador hace acceso a memoria externa, ya sea de datos o de programa, o cuando maneja a otro circuito periférico externo. El puerto P2 transmite el byte más significativo de la dirección cuando el microcontrolador hace acceso a un dispositivo externo.

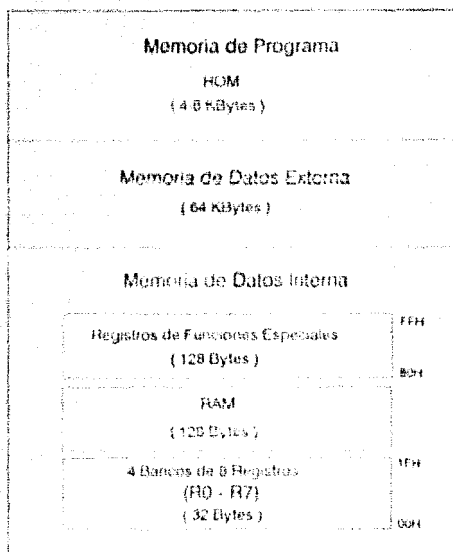


Figura 3.2 Mapa de memoria del MCS-51

Las terminales del puerto P3 pueden configurarse individualmente para desarrollar varias tareas como: la entrada y salida de señales en una transmisión serie, entradas para las señales de interrupción externa y para generar las señales de lectura y escritura para memorias externas.

SISTEMA DE INTERRUPTIONES

Una interrupción es una señal que indica al procesador que debe posponer lo que realiza para llevar a cabo otra acción de diferente prioridad. Las interrupciones pueden ser mascarables o no. Esto significa que una interrupción por hardware puede hacer por sí sola que el procesador pare la ejecución de la tarea actual y salte a una dirección donde generalmente se encuentra la rutina de servicio para atender la interrupción mencionada. Si la interrupción es mascarable, es decir, que únicamente se prenda una bandera, entonces el procesador decide si atiende o no a la interrupción solicitante.

Para que el CPU pueda atender a algún evento externo o a algún requerimiento de algún circuito periférico interno, fuera del programa, el microcontrolador cuenta con un sistema de interrupciones. La familia MCS-51 reconoce peticiones de interrupción provenientes de cinco fuentes: dos de fuentes externas, una de cada uno de los contadores internos y una más del puerto serie. Cada una de las interrupciones se le puede asignar cualquiera de los dos niveles de prioridad disponibles (alta o baja prioridad), y pueden ser habilitados y deshabilitados independiente o globalmente.

OSCILADOR

El microcontrolador cuenta con un amplificador inversor que puede ser usado como un oscilador interno. Puede utilizarse un cristal de cuarzo o la señal de un oscilador externo, haciendo una modificación en la conexión.

La conexión del cristal para el oscilador, ya sea externa o interna se muestra en la figura 3.3.

CONTADORES/TEMPORIZADORES

Los microcontroladores MCS-51 contiene dos contadores de 16 bits para medir intervalos de tiempo y anchos de pulso, para contar eventos así como para generar peticiones de interrupción periódicas y precisas. Cada uno de los contadores puede ser programado en cualquiera de los siguientes modos:

Modo 0: es un contador de ocho bits, cuenta hasta un número determinado dividido por la constante 32.

Modo 1: es un contador de 16 bits.

Modo 2: es un contador de 8 bits con recarga automática después de llegar a cuenta máxima.

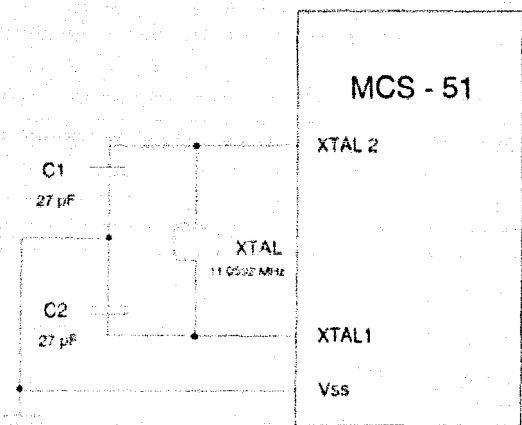


Figura 3.3 Conexion del cristal para el oscilador

Modo 3: es un contador doble

Los contadores son controlados por software, tanto para iniciar como para parar su operación. Cada contador enciende su propia bandera de petición de interrupción cuando llega a cuenta máxima o mínima.

III.2 Diseño del Módulo Principal

Se define como módulo principal aquel módulo que tiene como tarea prioritaria realizar el control de flujo, de las líneas de comunicación y de recepción de señales de los equipos de medición.

El módulo principal contiene la implementación necesaria para el buen funcionamiento y sincronización del uso del canal de comunicación, el cual es compartido por los módulos independientes; es el encargado de controlar las peticiones así como de generar las señales de autorización requeridas durante el proceso de selección y asignación.

III.2.1 El Controlador de Flujo

El controlador de flujo es aquella parte del módulo principal que se encarga de las funciones necesarias para establecer la comunicación entre los equipos de medición y el HOST. Tales funciones son:

- Selección del módulo independiente
- *Búsqueda del canal de comunicación*
- Establecimiento del circuito virtual
- Envío de la señal de reconocimiento al módulo independiente
- Selección del canal a transmitir.

Todas estas funciones se realizan por medio de ciertos circuitos, como multiplexores y decodificadores, que permiten el acoplamiento de las distintas señales que maneja el controlador de flujo como lo son las señales de petición de canal (CHRQ) y asignación de canal (CHAK). Para la selección de un módulo independiente que solicita entablar comunicación con el HOST, se utiliza un multiplexor de 8 líneas que funciona de acuerdo a la figura 3.4.

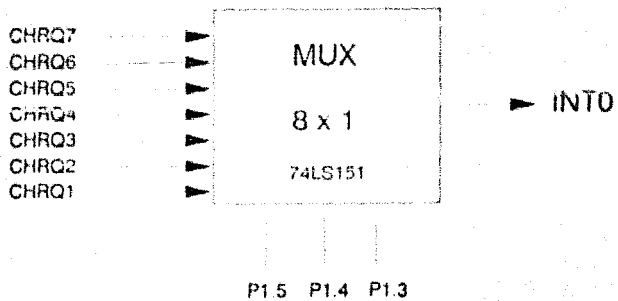


Figura 3.4

Las líneas de selección del MUX son controladas por tres bus del puerto 1 del controlador de flujo, específicamente los bus denominados P1 3, P1 4 y P1 5, los cuales funcionan secuencialmente como un contador de tres bits, como lo muestra la siguiente tabla :

Bus 5 4 3	Línea Selecciona
0 0 0	Ninguna
0 0 1	CHRQ1
0 1 0	CHRQ2
0 1 1	CHRQ3
1 0 0	CHRQ4
1 0 1	CHRQ5
1 1 0	CHRQ6
1 1 1	CHRQ7

La línea de salida del MUX se encuentra conectada a la línea de interrupción 0 (INT0) del controlador de flujo. Esta línea, activa baja, generará una señal de interrupción al controlador cuando los bus P1 3, P1 4 y P1 5 seleccionen una señal CHRQi, activa baja, que indica la solicitud del canal (channel request) a cargo del módulo apuntado por dichos bus, realizándose de esta manera un proceso de "polling" entre los procesadores actuales.

4H 2.2 Acceso al canal de Transmisión

Cada vez que un módulo independiente *i* realiza una petición de canal (CHRQi = 0), y es seleccionado a través del MUX por medio del controlador de flujo, es necesario indicar al HOST que un módulo *i* desea transmitir su información.

En este momento, el controlador de flujo establece un protocolo de comunicación con el HOST para indicarle que alguien desea enviarle información. El controlador de flujo espera hasta el arribo de la señal de reconocimiento por parte del HOST. Toda esta comunicación se realiza por medio del puerto serie en ambas partes.

Cuando el controlador de flujo ha recibido la señal de reconocimiento del HOST, realiza los siguientes procedimientos:

1) Selecciona el canal de transmisión del módulo independiente en el MUX de selección de transmisiones, por medio de los bits P1.0, P1.1 y P1.2 del puerto 1 como lo muestra la figura 3.5

2) Envía una señal de reconocimiento al módulo 1 (CHAKE) para indicarle que el canal de comunicación se encuentra libre y que el HOST esta esperando información. El bit P1.7 controla la habilitación del decodificador de reconocimiento como lo muestra la figura 3.6

3) Espera a que el módulo 1 termine la transmisión de datos para reanudar el conteo normal.

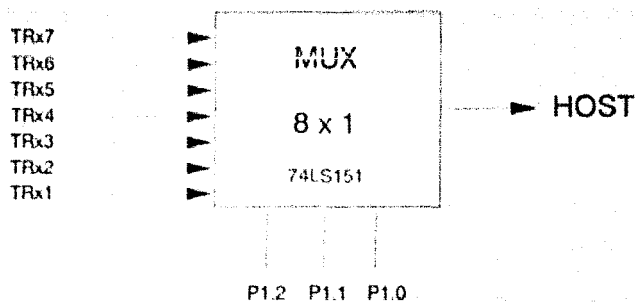


Figura 3.5

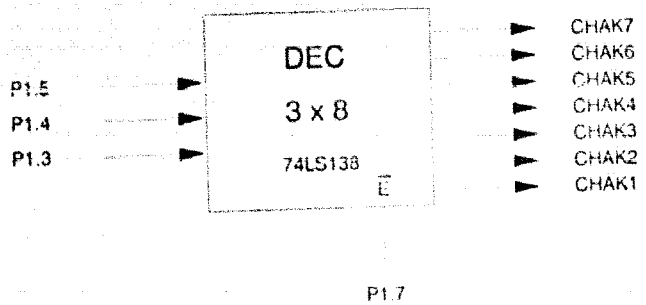
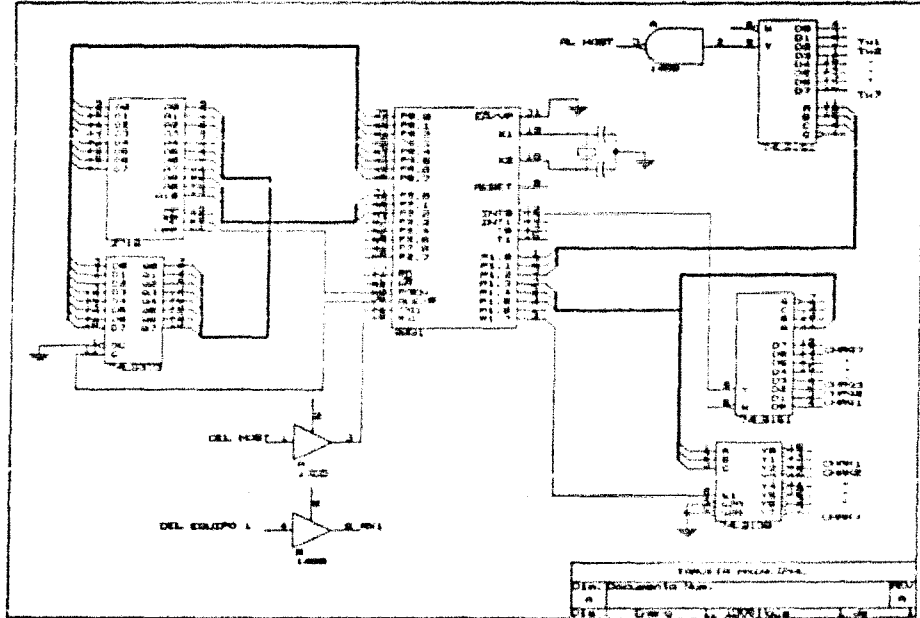


Figura 3.6

III.2.3 ALZAMIENTO DEL MÓDULO PRINCIPAL



III.3 Diseño de los Módulos Independientes

Se define como módulo independiente aquel módulo que se encarga de recibir la información que proviene de los equipos de medición, almacenarla en una memoria temporal, realizar una petición de canal y esperar una señal de reconocimiento para poder transmitir la información hacia el HONT.

III.3.1 El procesador principal

El procesador principal de los módulos independientes consiste en un microcontrolador 8751, que en un solo chip es capaz de realizar todas las funciones requeridas por el módulo.

La información proveniente de los equipos de medición es capturada a través de la línea serie de recepción (RXD).

La petición de canal (CHRG) se lleva a cabo poniendo en un nivel bajo (0) el bit P1.0 del puerto 1 de este microcontrolador.

Una vez solicitado el canal, el procesador se encarga de almacenar información y esperar un reconocimiento por medio del controlador de flujo, a través de su línea de interrupción 0 (INT0) cuando esta señal (CHAK) es recibida, el procesador es capaz de recibir, guardar y transmitir información al mismo tiempo.

III.3.2 Memoria de Almacenamiento Temporal

A pesar de que el microcontrolador 8751 posee todos los elementos necesarios para llevar a cabo absolutamente todas las funciones requeridas por el módulo independiente, su capacidad en memoria RAM es sumamente limitada, pues llega escasamente a los 128 bytes.

Para poder mantener la información proveniente del equipo de medición, en el lapso que ocurre entre la petición del canal y la adquisición del mismo, es necesario contar con una memoria temporal cuyo tamaño depende del tiempo mencionado. El análisis de la capacidad, considerando el peor de los casos, se estudia en el capítulo V.

Para tal efecto ha sido seleccionada una memoria RAM estática externa de 8K x 8, cuyas características son las siguientes :

Memoria en bytes : 8 K

Velocidad de acceso : 120 ns

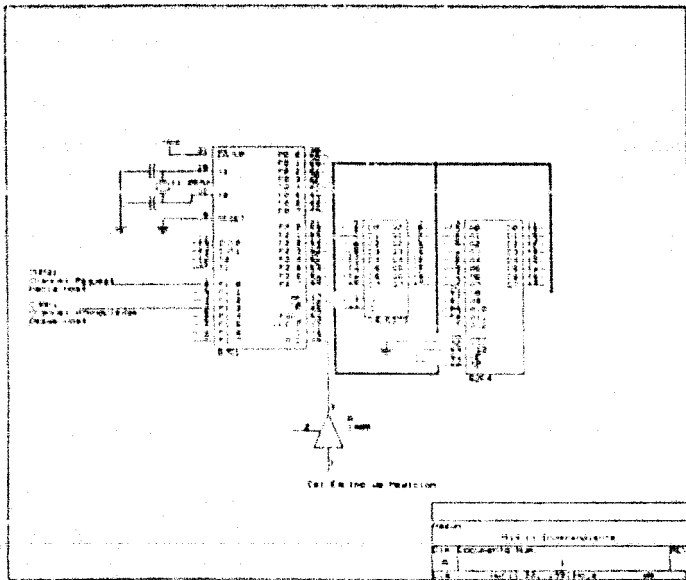


FIGURE 10 DRAWING OF WIRING CONNECTIONS

CAPITULO IV

DISEÑO E IMPLEMENTACIÓN DEL SOFTWARE DEL SISTEMA

IV.1 Firmware para el microcontrolador

Se define como Firmware a todos los programas escritos en un dispositivo de almacenamiento para ser la tarea como lo es una ROM. En este caso, el programa fue escrito en las EPROMs internas de cada microcontrolador.

IV.1.1 Elementos de procesamiento

Uno de los factores que determinan el cuánto le tomara a una microcomputadora completar una tarea determinada, es el número de instrucciones que debe ejecutar. Lo que permite a la arquitectura de un microcontrolador en particular poder ser eficiente para resolver un problema específico es que tan bien se adecue el set de instrucciones a la tarea que se desea realizar. Es decir, mientras mejor correspondan las microoperaciones a los pasos tomados por el algoritmo de control, menor número de instrucciones serán necesarias, así como mayor será la velocidad de ejecución.

Esta familia de microcontroladores de INTEL provee un amplio soporte para la manipulación de bits, lo que naturalmente conduce a la generación de programas mas eficientes que manejan las condiciones de entrada y salida binaria, inherentes a los problemas de control digital.

Existen cuatro elementos básicos que definen a una computadora digital, los cuales se encuentran integrados en este microcontrolador y que se explican brevemente a continuación:

- CPU

El CPU incorpora un procesador booleano dedicado a la ejecución de operaciones orientadas a manejo de bits. Este procesador incluye 17 instrucciones específicas, así como otras 9 dedicadas a manejo de bytes.

- Memoria de Programa

Las instrucciones de procesamiento de bits son tomadas de la misma memoria que utilizan las operaciones aritméticas y lógicas. En adición a esto, existen varias características muy sofisticadas para el control de los programas como lo son diferentes modos de direccionamiento, andamiaje de interrupciones, etc.

- Memoria de Datos

Existen instrucciones que pueden operar directamente sobre 144 bits de propósito general que forman la RAM del procesador, e inclusive manejar algunos de los registros de función especial (SFRs), bit por bit.

- Entrada / Salida

Existen 32 pins de I/O que pueden ser direccionados de manera individual como entradas, salidas o ambas en cualquier combinación. Existen, además, 53 bits que pueden reconfigurar el control, monitorear el estatus del CPU y cualquier otra función interna del chip como los timers, puerto serie, etc.

Gracias a estos cuatro elementos es posible desarrollar casi cualquier algoritmo de control digital y sobre todo con mucha flexibilidad debido al exhaustivo manejo de bits que proporciona el procesador bootstrap de esta familia.

Antes de abordar los algoritmos específicos de control para este sistema es conveniente dar una breve explicación acerca de los registros de función especial (SFRs) que serán utilizados más adelante.

PSW (Program Status Word)

Definición : Contiene el estado actual del procesador.

Organización

CY : Bandera de Carry

AC : Bandera de Carry Auxiliar

EO : Bandera 0. Definida por el usuario

RS1 : Bit de control 1 para selección del banco de registros

RS0 : Bit de control 0 para selección del banco de registros

OV : Bandera de Overflow

• : Reservada

P : Bandera de Paridad

PCON (Power CONTROL)

Definición : Contiene la configuración del bandaje del puerto serie y los modos de ahorro de energía en el caso de circuitos CMOS

Organización

SMOD : Indica bandaje doble (1) o sencillo (0)

• : Reservada

• : Reservada

• : Reservada

GF1 : Bandera de propósito general 1

GF2 : Bandera de propósito general 2

PD : Power Down

IDLE : Idle Mode

SCON (Serial port CONtrol)

Definición : Contiene la información para poder configurar el puerto serie así como las banderas para su control.

Organización :

SM0 : Bit de modo serie 0

SM1 : Bit de modo serie 1

SM2 : Bit de modo serie 2

REN : Habilitador de Recepción

TB8 : Transmit Bit 8 (9 bit Mode)

RB8 : Receive Bit 8 (9 bit Mode)

TI : Transmit Flag

RI : Receive Flag

NOTA: SM0 SM1 Significado

0	0	Registro de comentario sencillo
0	1	UART de 8 bits con baudaje variable
1	0	UART de 9 bits con baudaje fijo
1	1	UART de 9 bits con baudaje variable

IE (Interrupt Enable)

Definición: Contiene la información para configurar las interrupciones de manera independiente o global.

Organización:

EA: Enable All (Habilitador global)

RES: Reservada

ET2: Enable Timer 2 (solo para 8052)

ES: Enable Serial Port

ET1: Enable Timer 1

EX1: Enable External 1

ET0: Enable Timer 0

EX0 : Enable External 0

TCON (Timer CONTROL)

Definición : Contiene la información necesaria para el control de los timers y las interrupciones externas 0 y 1.

Organización :

TF1 : Bandera de Overflow del Timer 1

TR1 : Bandera para encendido (1) o apagado (0) del Timer 1

TF0 : Bandera de Overflow del Timer 0

TR0 : Bandera para encendido (1) o apagado (0) del Timer 0

IE1 : Bandera de detección de flanco de interrupción ext. 1

IF1 : Bit de control de flanco de interrupción externa 1

IE0 : Bandera de detección de flanco de interrupción ext. 0

IF0 : Bit de control de flanco de interrupción externa 0

TMOD (Timer MODe control)

Definición : Contiene la información necesaria para configurar los timers y contadores.

Organización :

GATE: Bit para configurar disparo por hardware (1) o software (0) del timer counter 1

CT: Selección de Contador1 (1) o Timer1 (0)

M1 : Bit 1 para modo de timer counter 1

M0 : Bit 0 para modo de timer counter 1

GATE: Bit para configurar disparo por hardware (1) o software (0) del timer counter 0

CT: Selección de Contador0 (1) o Timer0 (0)

M1 : Bit 1 para modo de timer counter 0

M0 : Bit 0 para modo de timer counter 0

Todos y cada uno de estos registros de función especial deben ser configurados antes de iniciar el algoritmo de control requerido, para asegurar el buen funcionamiento del procesador desde el inicio.

IV.1.2 Firmware para el Módulo Principal

El algoritmo que debe seguir el Módulo Principal para poder realizar las funciones que le son encomendadas, es fácilmente explicable por medio del diagrama de flujo que muestra la figura 4.1.

Para poder implementar este algoritmo, de acuerdo al diagrama, tenemos:

Después de un RESET, el procesador empieza a ejecutar a partir de la dirección 0000H. Para poder preservar la memoria asignada a los vectores de interrupción ya definidos, realizamos un salto hacia una localidad libre, por ejemplo la localidad 0100H.

NUEVO: IJMP INIT

En este momento es preciso inicializar los registros que sirven para configurar al procesador

INIT :	SETB SM1	;Elegimos un UART de 8 bits
	SETB REN	;Habilitamos la Recepción serie
	ORL TMOD,#20H	;Elegimos Timer 1 de 8 bits con
		; autorecarga
	ORL TH1,#0FDH	;Iniciamos la cuenta para
		; trabajar a 9600 bauds
	ANL IE,#00H	; Deshabilitar todas la

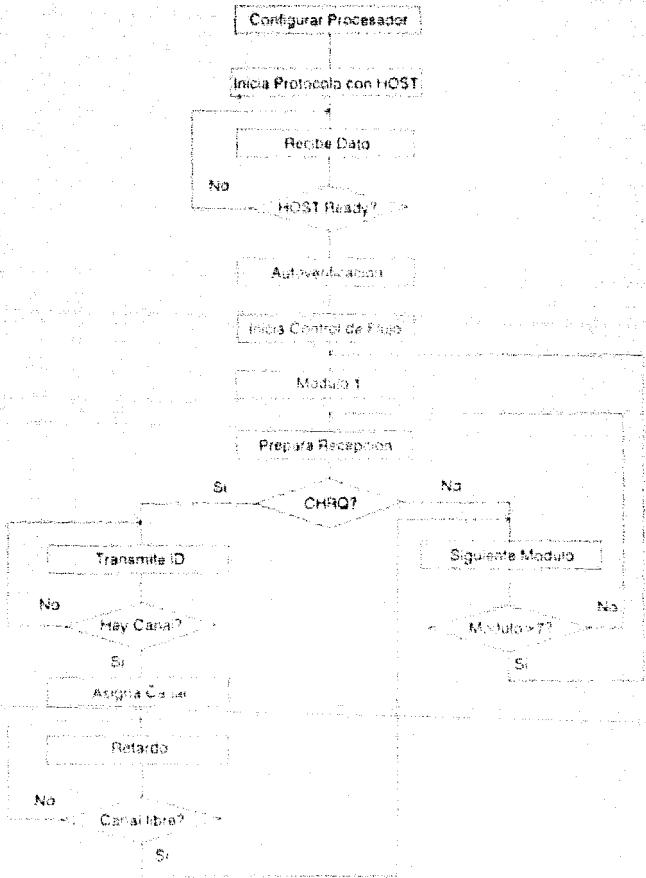


Figura 4.1 Diagrama de flujo del Modulo Principal

```

; interrupciones posibles
ANL  PCON,#00H ; Utilizar Baudaje sencillo para
                ; el puerto serie
SETB IT0       ; Configurar la interrupción
                ; externa 0 para trabajar con
                ; flanco de subida negativo
SETB TR1       ; Poner a trabajar el Timer 1

```

Una vez inicializados los registros correspondientes para trabajar adecuadamente, iniciamos el protocolo de comunicación con el HOST

```

RECIBE:  JBC  RI,CHECABIT ; Saltamos a preguntar que tipo
                ; de dato fue recibido en el
                ; UART

```

```

        SJMP RECIBE ; O bien esperamos hasta que
                ; exista un dato en el Buffer de
                ; Recepción

```

```

CHECABIT: MOV  A,SBUF ; Leemos el dato proveniente
                ; del HOST
        CJNE A,#11H,RECIBE ; y lo comparamos contra un

```

```

; (X ON), el cual indica que
; el HOST esta listo para
; recibir información
; Cualquier otro carácter es
; descartado y esperamos por
; un carácter válido

```

En el momento en que es detectado un carácter válido que proviene del HOST, se inicia una rutina de autoverificación del controlador. La cual asegura que el equipo esta funcionando en un 100%.

```

MOV R1,#09H ; Iniciamos contador de elementos
CICL MOV A,#00H ; Limpiamos el acumulador
MOV P1A ; Seleccionamos canal libre de
; comunicación entre el controlador
; de bus y el HOST
MOV A,R1 ; Iniciamos el acumulador con el
; indicador de modulo en turno
TRANSM MOV SBUF,A ; Indicamos al HOST el modulo
; elegido
MOV P1A ; Seleccionamos canal libre entre
; el modulo apuntado por R1 y el
; HOST

```


	MOV R0,#0FH	
ESP1:	DJNZ R0,ESP1	; Esperamos un momento
	JBC R1,ASIG1	; Preguntamos por la respuesta del
		; HOST para empezar a enviar
		; informacion
	SJMP TRANSM	; Saliramos en caso contrario
ASIG1:	SETB P1.7	; Indicamos al modulo apuntado por
		; R1 que puede empezar su
		; transmision
ESP2:	JBC IE0,SUM1	; Preguntamos por su terminacion
	SJMP ESP2	; esperamos en caso contrario
		;
SUM1:	CLR P1.7	; Limpiamos la ultima asignacion
	MOV A,R1	; Actualizamos los apuntadores
	ADD A,#0FH	
	MOV R1,A	
	CJNE A,#1SH,C01	; Comparamos contra el ultimo
		; modulo e iniciamos
		; brevemente el ciclo hasta
		; concluir con todos los
		; modulos disponibles
		;
	MOV R0,0FH	; Esperamos un momento
ESP3:	DJNZ R0,ESP3	; antes de iniciar con el
		; algoritmo de control

Una vez terminado el ciclo de autoverificación iniciamos el algoritmo de control de flujo entre módulos independientes y el HOST. Este algoritmo es sumamente parecido a la rutina de autoverificación, solo que aquí el controlador de flujo utiliza "polling" para saber que modulo posee información la cual debe comunicar al HOST

```

CICLO2:  MOV  A #08H
CICLO2:  MOV  P1 A
        JBC  I/O.SICHRQ      ; Preguntamos por CHIRQ

SUMA:    ADD  A,#08H
        CJNE A,#0BH,CICLO1  ; Actualizamos Apuntadores
        SJMP CICLO2

SICHRQ:  MOV  SBUF A          ; Transmitimos identificación
        MOV  RO,#0FH
        DJNZ RO,ESPERA
        JBC  RLAMCNA
        SJMP SICHIRQ

ASIGNA:  MOV  RO,#03H        ; Asignamos Canal de Com.
ROTA:    RR   A
        DJNZ RO,ROTA
        ORL  P1 A
        SETB P1.7

```

```

WAITEA:   JBC   IE0,LIBRE           ; Esperamos liberación de canal
          SJMP  WAITEA

LIBRE:    CLR   PL.7
          STTB  ITO
          MOV   A,PI
          ANI   A,#350H
          SJMP  SUMA           ; Continuamos ciclo de manera
                               ; indefinida

```

IV.13 Firmware para el Módulo Independiente.

Este algoritmo funciona igual para cualquier módulo independiente, lo que implica una muy alta flexibilidad e independencia con respecto al equipo de medición que se encuentre susando.

El algoritmo que debe seguir el Módulo Independiente para poder realizar las funciones que le son encomendadas, es mostrado en el diagrama de flujo indicado en la figura 4.2.

Para poder implementar este algoritmo, de acuerdo al diagrama, tenemos:

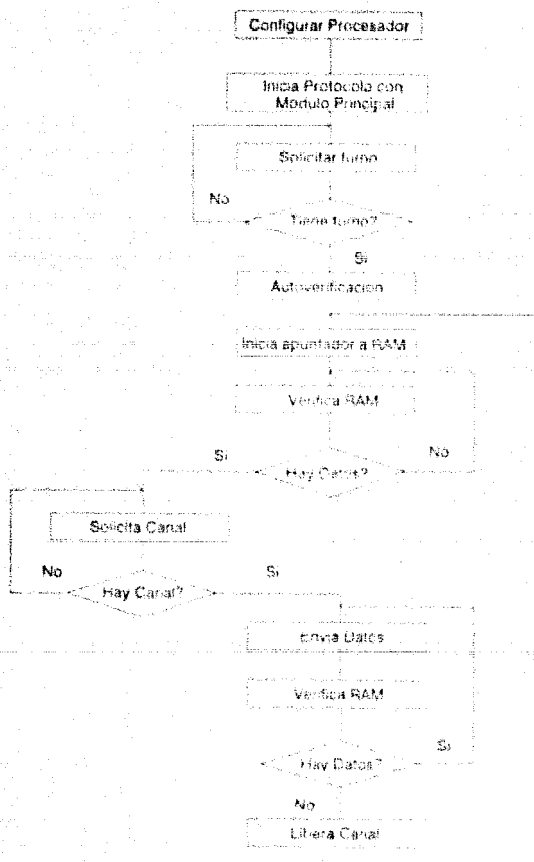


Figura 4.2 Diagrama de flujo del Modulo Independiente

Como ya se había indicado anteriormente, Después de que se lleva a cabo un RESET, el procesador empieza a ejecutar a partir de la dirección 0000H. Para poder preservar la memoria asignada a los vectores de interrupción ya definidos, realizamos un salto hacia una localidad libre, por ejemplo la localidad 0100H

SEUVO JUMP INIT

Sin embargo, el espacio para el código perteneciente a los vectores de interrupción se encuentra entre el RESET y el inicio del programa principal. En este caso, el único vector de interrupción que se va a utilizar es el perteneciente a la interrupción proveniente del puerto serie

ORG	023H	; Este vector inicia en la dirección 23H
	PUSH PSW	; Salvamos el estatus del procesador
	ACALL PTOSERIE	; Llamamos a la rutina de atención
	POP PSW	; Devolvemos los banderas de estatus
	RETI	; regresamos a donde fue llamada la interrupción.
ORG	0100H	; Iniciamos el programa principal
INIT:	SETB ES	; Habilitamos la interrupción por puerto serie
	SETB EA	; Habilitamos las interrupciones
	SETB SM1	; Configuramos el puerto serie como un UART de 8 bits

```

SETB REN ;Habilitamos la Recepción
ORL TMOD,#00H ; Iniciamos el Timer 1 como de
; 8 bits y autorecarga
ORL TH1,#0FDH ; Configuramos la cuenta para
; trabajar a 9600 bauds
SETB T10 ; Configuramos la interrupción
; externa con flanco de onda
; creciente
ANL PCON,#00H ; Utilizamos Baudrate sencillo
SETB TR1 ; Iniciamos el Timer
;
AUTO TEST: JBC IE0,ENVI0 ; Preguntamos por el turno para
; iniciar autoverificación
SIMP AUTO TEST ; o esperamos turno
;
ENVI0: MOV SBUF,PSW ; Enviamos los principales
; registros de función especial
; a modo de autoverificación
ACALL WAIT
MOV SBUF,SP
ACALL WAIT
MOV SBUF,LD
ACALL WAIT
MOV SBUF,TMOD
ACALL WAIT
MOV SBUF,TCON
ACALL WAIT

```

	MOV	SBUF,SCON	
	ACALL	WAIT	
	SETB	P1.0	; Indicamos la terminación de
	CLR	P1.0	; Autoverificación, continuar
	SETB	P1.0	; con el algoritmo de
			; recolección de información
INITRAM:	ANL	DPL #00H	; Iniciamos los apuntes para
	ANL	DPH #00H	; manejo de memoria externa.
	ANL	R0 #00H	
	ANL	R1 #00H	
CHKRAM:	MOV	A,DPL	; Iniciamos el ciclo de
			; verificación de la RAM externa
	CJNE	A,R0,HAYDATO	
	MOV	A,DPH	
	CJNE	A,R1,HAYDATO	
	SJMP	CHKRAM	; Hasta que exista un dato
HAYDATO:	CLR	P1.0	; Solicitamos canal
			; CHRQ = 0
TRYAGAIN:	JBC	IE0,HAYANAL	; Esperamos hasta la señal de
			; reconocimiento de canal
			; CHAK = 0
	SJMP	TRYAGAIN	

```

HAYCANAL: CLR EA ; Si hay canal deshabilitamos
; cualquier interrupción

MOV R2,DPL
MOV R3,DPIH
MOV DPL,R0
MOV DPIH,R2
MOVXA,0,DPIH
MOV SBI,EA ; Enviamos dato al HOST
INC DPIH
MOV R0,DPL
MOV R1,DPIH
MOV DPL,R2 ; Reestructuramos los
MOV DPIH,R3 ; apuntadores

SETB EA ; Habilitamos interrupciones
MOV A,DPL
CJNE A,R0,HAYCANAL ; Si todavia hay datos
MOV A,DPIH ; los enviamos
CJNE A,R1,HAYCANAL
SETB P10 ; Si ya no hay datos,
CLR P10 ; liberamos el canal
SETB P10
SJMP CHECARAM ; Iniciamos el ciclo

PEOSFRIB: CLR EA ; Esta es la subrutina

```


	CLR RI	: de atención para el
	PUSH ACC	: puerto serie
	MOV A, SBUF	
	MOVX @DPTR, A	: Se lee el dato del puerto
	INC DPTR	: y se guarda en RAM
	POP ACC	
	SETB EA	
	RET	
WAIT:	MOV R0, #0FH	: Esta es la subrutina de
ESPERA:	DECZ R0, ESPERA	: espera
	RET	

IV.2 Software de comunicación entre el HOST y el CONTROLADOR

Se define como software de comunicación a aquel programa que es ejecutado en el HOST y que permite establecer una línea directa con el CONTROLADOR a través del puerto serie de ambas partes.

Este programa está basado en el BIOS del HOST y se puede integrar a un algoritmo más complejo.

El BIOS (Basic Input/Output System) es una serie de servicios de software que habilitan a los programas para utilizar el hardware que constituye una

computadora personal. Esta serie de servicios esta constituida por rutinas de bajo nivel que forman una especie de "escudo" entre el hardware y otros programas. Normalmente estos servicios son conocidos como interrupciones y cada interrupcion posee una o varias funciones diferentes de acuerdo a lo que se necesite de esa interrupcion.

Existe una interrupcion especifica conocida con el nombre de interrupcion 14H, la cual se encarga de controlar el o los puertos de comunicacion serie. Esta interrupcion posee cuatro funciones diferentes, las cuales son seleccionadas por medio del registro AH como lo muestra la siguiente tabla.

AH	Función
00H	Inicializar Puerto Serie
01H	Enviar un carácter
02H	Recibir un carácter
03H	Obtener el estatus del puerto

Aunque los registros y las funciones de esta interrupcion son llevadas a cabo a bajo nivel, es posible realizar una interfase con un lenguaje de alto nivel. En este caso en particular, revisaremos la opción de interfase utilizando Turbo Pascal Versión 5.0.

• Inicialización del puerto serie.

```
Procedure Inicia( Palabra : Byte;
```

```
    Var Regs : Registers);
```

```
Begin
```

```
    Regs.AH := 00;
```

```
    Regs.DX := 00;
```

```
    Regs.AL := Palabra;
```

```
    Inr($14 Regs);
```

```
End;
```

En este procedimiento se utiliza el parámetro "Palabra" para inicializar el puerto serie. Esta Palabra se puede manipular bit por bit en otro procedimiento, de tal manera que contemple todas las posibilidades que proporciona el estándar 8088, como baudíe, paridad, número de bits de stop y longitud del carácter.

• Verificación del estado del puerto

```
Procedure Status( Var Estado : Byte;
```

```
    Var Regs : Registers);
```

```
Begin
```

```
    Regs.AL := Estado;
```

```

Regs.AH = 03,
Regs.DX = 00,
Intr($14,Regs),
Estado = Regs.AH;

```

End;

Este procedimiento regresa el estatus del puerto serie en la variable "Estado". Cabe aclarar que es sumamente importante verificar el estatus del puerto ANTES de leer un carácter del mismo. El no realizarlo así puede ocasionar resultados inesperados.

• Recepción de un carácter.

```

Procedure Receive( Var Carac : Byte);

```

```

    Var Regs : Registers;

```

Begin

```

    Regs.AL = Carac,
    Regs.AH = 02,
    Regs.DX = 00,
    Intr($14,Regs),
    Carac = Regs.AL,

```

End;

La variable "Carac" regresa un carácter que fue leído desde el puerto serie, utilizando la función 0211

• Transmisión de un carácter.

```

Procedure Send(   Var Dato : Byte;
                 Var Regs : Registers);
Begin
    Regs.AL := Dato;
    Regs.AH := 01;
    Regs.DX := 00;
    Intr($14 Regs);
    Dato := Regs.AL;
End;

```

Este procedimiento permite enviar cualquier carácter o número que contenga el parámetro "Dato", utilizando la función 0111

En cualquiera de los casos anteriores, se define una variable llamada "Regs", previamente definida por Turbo Pascal que contiene a todos los registros

pertencientes al procesador de la computadora personal. Después de ejecutar la interrupción, uno o más de estos registros adquiere un valor de acuerdo al resultado de la operación.

Si se utilizan los procedimientos anteriormente definidos, el siguiente segmento de código permitirá revisar el puerto serie e imprimir en pantalla el valor del dato leído, así como transmitir un carácter leído del teclado y terminar la sesión si se presiona la tecla de «ESC». En cualquier caso, se presupone que se realizó la inicialización previamente.

```
Repeat
  Status(Estado, Regs);
  Estado = Estado And 01;
  If Estado = 01 Then
    Begin
      Receive(Dato, Regs);
      WriteLn( Caracter = Char(Dato), Valor = Dato);
    End;
  (*End_If*)
  If KeyPressed Then
    Begin
      Tecla = ReadKey;
      If Tecla <> #27 Then
        Begin
          WriteLn;
```

```
WriteLn('Enviando : ',Tecla);  
Dato:=Ord(Tecla);  
Send(Dato,Regs);  
End;  
(*End_II*)  
End;  
(*End_II*)  
Until Tecla=#27;
```

Ahora bien, una vez que se establece el protocolo de comunicación entre el Sistema de Adquisición de Datos y la computadora, la información proveniente de cada módulo independiente se guarda en un archivo propio y diferente para cada equipo. Estos archivos presentan el formato necesario para ser procesados por una hoja electrónica de datos (EOTSI), de tal manera que cualquier operador sería capaz de leerlo e integrarlo a su hoja de trabajo.

CAPITULO V

PRUEBAS Y EVALUACION

Los objetivos de esta etapa son valorar y mejorar la calidad del producto generado durante la fase de desarrollo y modificación del sistema. Como principales atributos de la calidad hemos considerado la consistencia, la confiabilidad y la eficacia, que nos permiten determinar el grado en que se cumplen las especificaciones establecidas durante el planteamiento del problema y la conformidad con los requisitos.

Por rendimiento queremos expresar la manera o la eficiencia con que un sistema cumple sus metas. Así pues, el rendimiento es una cantidad relativa, aunque en ocasiones suele hablarse de "medidas absolutas de rendimiento" (DIB*) como el número de trabajos por hora que un sistema determinado puede hacer.

Algunas medidas de rendimiento como el tiempo de respuesta por ejemplo, se dice que están orientadas al usuario.

Entre las medidas de rendimiento que realizamos, tenemos las que se listan a continuación

- Capacidad de almacenamiento.

Existe una memoria RAM asociada a cada equipo, la cual tiene una capacidad de almacenamiento de 8192 bytes. Si la computadora no está en condición de atender al Sistema de Adquisición, existirá entonces un número límite de mediciones que se puedan realizar antes de que se agote la capacidad de la RAM. La siguiente tabla muestra esta situación:

Equipo	Bytes por medición	Número Máximo de mediciones permitidas	Total ocupado en RAM (Bytes)
Medidor de tensión	100	81	8100
Medidor de Espesor	10	819	8190
Medidor de resistencia al rasgado	63	130	8190

Las pruebas que aquí se presentan fueron realizadas con los equipos que actualmente se encuentran trabajando en el laboratorio y como se puede observar, esta capacidad varía de acuerdo a las características particulares de cada equipo.

• Capacidad de ejecución

Es la medida de la ejecución de trabajo por unidad de tiempo. Los equipos de medición envían los datos a una tasa de 9600 bauds, cada procesador asociado, así como la computadora, son programados para recibir y transmitir información a la misma tasa de 9600 bauds.

Particularmente, cada instrucción que realiza el procesador asociado se realiza entre 12 y 24 periodos de oscilador, de tal manera que cada una de estas operaciones se efectúa en un tiempo de 1 a 2 microsegundos.

El sistema es capaz de sensar la llegada de datos, almacenarlos en memoria RAM, establecer un protocolo de comunicación con la computadora y transmitir los datos almacenados sin ninguna restricción ni limitación.

Carga de trabajo

La carga de trabajo es la medida de la cantidad de trabajo que ha sido introducida en el sistema, y que el sistema debe procesar normalmente para funcionar de una manera aceptable.

La carga de trabajo dependerá directamente del número de muestras que se realizan en un periodo, y del tiempo que es requerido para realizar una prueba: montar, probar y desmontar cada una de las muestras. La siguiente tabla muestra esta relación de tiempos para los equipos asociados.

Equipos	Tiempo para montar y probar (seg)	Tiempo para desmontar y cambiar (seg)
Medidor de Tensión	15	15
Medidor de Espesor	5	5
Medidor de Resistencia al rasgado	10	10

El número de pruebas que se realizan en un equipo es de un promedio de 20 muestras por lote. Si analizamos la relación de esta tabla con la tabla de capacidad de almacenamiento, se puede observar que existe un factor de seguridad que permite la liberación de la computadora para atender otras tareas antes de permitir el acceso al Sistema de Adquisición de Datos. Este factor dependerá del equipo en cuestión, y suponiendo que el sistema se encuentre trabajando con siete equipos simultáneamente, el tiempo total de recuperación del sistema es de seis segundos, tiempo suficiente para continuar efectuando pruebas.

CAPITULO VI

CONCLUSIONES

Todo elemento desarrollado por el hombre, surge como una idea en su mente. Los sistemas se desarrollan como una respuesta a requerimientos detectados, y el diseño de este sistema puede hacerse de diversas formas, la tendencia de utilizar la primera solución que se nos presenta, es un problema muy frecuente, y una forma de evitarlo, es desarrollando una estrategia de solución, enumerando las posibles soluciones y diferentes opciones con que se cuenta. La factibilidad de cada una de estas posibilidades se establece por el análisis de las restricciones del problema en particular.

Con este proyecto se brinda una solución al problema de comunicar varios equipos de medición, de manera simultánea, a una sola computadora utilizando su puerto serie y presentando las siguientes características:

- Posibilidad de ampliar la capacidad para atender hasta siete equipos de medición simultáneamente, sin comprometer el canal de comunicación con la computadora, utilizando un solo puerto serie.
- El diseño modular facilita la expansión de la capacidad, permitiendo añadir más módulos independientes a medida que se vuelven necesarios.

- Otro de los atractivos de este diseño es que si un módulo independiente falla, no interfiere en el funcionamiento del sistema general

- El sistema permite la realización de pruebas en cualquier momento, sin la posibilidad de perder un solo dato, independientemente de que los equipos de medición permitan o no establecer un protocolo de comunicación

- No se requiere la atención plena de la computadora, y esta podrá estar realizando otra tarea sin involucrarse en el proceso de prueba

- El sistema ofrece la posibilidad de expansión de funciones para proyectos futuros

Algunos de los aspectos del diseño de sistemas sólo pueden aprenderse a través de la experiencia. Nos damos el placer de presentar un enfoque moderno del diseño de sistemas digitales, dado que la velocidad de cambio ha sido extraordinaria, nuestro principal objetivo ha sido realizar un trabajo tan eficiente como sea posible

BIBLIOGRAFIA

- AB187** Abel, Peter
IBM PL Assembler Language and Programming
Prentice Hall Inc. 1987
- ARM86** Armbrast, Steven
Programming & Reference Manual for ARM PC.
Dow Jones-Irwin 1986
- IN180** Intel Corporation
8 Bit Embedded Controller Handbook
Intel Corp. Santa Clara Ca. 1980
- NA190** National Instruments
ELC-1984 Control, Data Acquisition and Analysis for your Computer
National Instruments Corp. 1980
- RO189** Rosch, Wern H.
The Wern Rosch Hardware Bible
Ed Brady, N.Y. 1989
- TIW88** Flwing Albert Instrument Company
Technical Reference Handbook
Flwing Albert Co. 1988
- DI186** DICE Magazine
McGraw Hill
1988

ESTA TESIS NO DEBE
SALIR DE LA BIBLIOTECA

APENDICE A

Alambriado del Módulo Principal con un
Módulo independiente Epico

