



**AUTOMATIZACION EN EL PROCESO DE FABRICACION  
DE DESINFECTANTES Y CERAS PARA PISO**



**TRABAJO DE GRADUACION  
PREPARADO PARA LA FACULTAD  
DE INGENIERIA**

**PARA OPTAR AL GRADO DE:  
INGENIERO EN ELECTRONICA**

**POR  
CESAR ERNESTO CONTRERAS PALMA  
VICTOR CUCHILLAC CALLEJAS**

**NOVIEMBRE - 1997**

**SOYAPANGO - EL SALVADOR - CENTROAMERICA**

**UNIVERSIDAD DON BOSCO**

# UNIVERSIDAD DON BOSCO

*RECTOR*

ING. FEDERICO MIGUEL HUGUET RIVERA

*SECRETARIO GENERAL*

PBRO. PEDRO JOSE GARCIA CASTRO S.D.B



*DECANO DE LA FACULTAD DE INGENIERIA*

ING. CARLOS ALBERTO GUTIERREZ PEÑA

*ASESOR DE TRABAJO DE GRADUACION*

SAUL EDGARDO VELASCO LARREYNAGA

*JURADO EXAMINADOR*

ING. FEDERICO JOSE LAINEZ

ING ALEXANDER GUZMAN K.

**UNIVERSIDAD DON BOSCO**

**FACULTAD DE INGENIERIA**

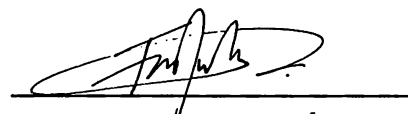
**DEPARTAMENTO DE INGENIERIA ELECTRONICA**

**JURADO EVALUADOR DEL TRABAJO DE GRADUACION**

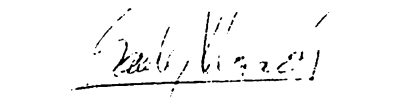
***“AUTOMATIZACION EN PROCESO DE FABRICACION DE  
DESINFECTANTES Y CERAS PARA PISO”***



**ING. ALEX GUZMÁN  
JURADO**



**ING. FEDERICO LAÍNEZ  
JURADO**



**ING. SAUL E. VELASCO  
ASESOR**

## **A. OBJETIVO GENERAL**

El objetivo fundamental de este trabajo es el diseñar y construir un dispositivo que esté basado en microcontroladores y que pueda proporcionar un mecanismo para impulsar sistemas autosostenidos en el área de control industrial y electrónico tanto para la industria nacional como para la Escuela de Ingeniería Electrónica.

## **B. OBJETIVOS ESPECÍFICOS**

1. Proveer a la Escuela de Ingeniería Electrónica de la Universidad Don Bosco con una herramienta bibliográfica elemental para la enseñanza de sistemas basados en microcontroladores Intel a nivel básico.
2. Brindar los conocimientos básicos de los microcontroladores de Intel para aquellos que se introducen a su estudio por primera vez.
3. Probar la aplicación de un controlador mínimo de bajo costo para la industria salvadoreña, para el caso, en la industria dedicada en la fabricación de desinfectantes y ceras para piso.
4. Promover el diseño de sistemas autosostenidos con microcontroladores en el área de Control y Automatización de la Escuela.

El hardware por si solo no trabaja en sistemas con microcontroladores, por tanto se creo un programa monitor para la ejecución de las diversas actividades. Por orden de prioridades se han considerado en cada etapa del diseño el factor económico, simplicidad y sencillez, flexibilidad, factibilidad y otros; como guías en las decisiones de lo que se debe incluir o no.

El sistema puede aumentar sus capacidades, por lo que no solo es aplicable para el proceso mencionado, dependerá de los usuarios lograr su evolución dado que puede modificarse para efectuar tareas diferentes a la cual fue concebida. Por lo tanto, se pretende promover en las personas que lean este documento no solo el brindar conocimientos generales de un proceso industrial específico o de un microcontrolador, sino, inculcar el deseo de crear sistemas basados en microcontroladores y explotar todas las bondades inherentes en ellos sistemas económicos, flexibles, robustos, seguros y confiables.

Debido a lo anterior, se puede afirmar que el sistema es versátil, de bajo costo y tiene proyección de desarrollo. Creemos que para la industria nacional tanto como los centros de educación superior representa una buena alternativa para el aprendizaje y aplicación.

## ÍNDICE

Introducción.....	1
1.Descripción del proceso	
1.1 Estado Actual.....	4
1.2 Proceso de fabricación.....	5
1.3 Propuesta.....	9
2 Metodología de la automatización	
2.1 Método para automatizar.....	13
2.2 Definición de entradas y salidas.....	16
2.3 Esquema de la planta.....	18
2.4 Especificaciones para materializar módulo.....	19
3. Sistema basado en microcontroladores	
3.1 Los microcontroladores.....	29
3.2 La familia MCS-51.....	34
3.2.1 Descripción de la familia MCS-51.....	34
3.2.2 Descripción de los pines del 8051.....	36
3.2.3 Organización de memoria.....	40
Ram de propósito general.....	41
Banco de registros.....	44
3.2.4 Registros especiales.....	44
Acumulador.....	44
Program status word.....	44
Registro B.....	46
Stack pointer.....	46
Data pointer.....	47
Registros de puerto.....	48
Registros de temporizadores.....	48
Registros de puerto serial.....	48
Registros de interrupciones.....	48
Registros de control de potencia	49
3.2.5 Memoria externa.....	50
Accesando la memoria externa de código...	51
Accesando datos de la memoria externa....	53

3.3	Temporizadores y contadores.....	56
3.3.1	Registro TMOD.....	56
3.3.2	Registro TCOM.....	58
3.3.3	Modos de operación.....	58
3.3.4	Fuentes de reloj.....	59
3.4	Puerto serial.....	60
3.4.1	Registro SCON.....	60
3.4.2	Modos de operación.....	61
3.4.3	Baud Rate del puerto serial.....	63
3.5	Interrupciones.....	65
3.5.1	Organización de las interrupciones.....	65
3.5.2	Habilitando y deshabilitando interrupciones..	66
3.6	Set de instrucciones.....	67
3.6.1	Modos de direccionamiento.....	68
	Direccionamiento por registro.....	69
	Direccionamiento directo.....	69
	Direccionamiento indirecto.....	69
	Direccionamiento inmediato.....	70
	Direccionamiento relativo.....	70
	Direccionamiento absoluto.....	70
	Direccionamiento largo.....	70
	Direccionamiento indexado.....	70
3.6.2	Tipo de instrucciones.....	71
	Aritmética.....	71
	Lógica.....	71
	Transferencia de datos.....	71
	Variable booleana.....	72
	Salto de programa.....	72
4.	Diseño del módulo controlador	
4.1	Diseño del módulo mínimo.....	73
4.1.1	Reloj del sistema.....	74
4.1.2	Reset del sistema.....	76
4.1.3	Decodificadores.....	79
4.1.4	Latches.....	80
4.1.5	Memorias.....	81
4.1.6	Comunicación serial vía RS- 232C.....	86

4.1.7 Fuente de alimentación.....	88
4.1.8 Mapeo de memoria.....	89
4.2 Teoría sobre interfaces y periféricos.....	90
4.2.1 Manejadores.....	90
4.2.2 Teclado.....	93
4.2.3 Pantalla de cristal líquido.....	96
5. Manual	
5.1 Descripción del circuito.....	102
5.2 Comunicación serial.....	106
5.3 Mapa de memoria.....	108
5.4 Recomendaciones para la tarjeta.....	110
5.5 Tabla de características.....	112
5.6 Instalación y mantenimiento del DTK-51.....	113
5.7 Ajuste y calibración.....	113
5.8 Manual para sistema mezclador basado MCS-51...	114
5.9 Pasos para programación.....	116
5.9.1 Proceso para dosificación de ingredientes....	118
5.9.2 Proceso por receta.....	122
5.9.3 Mensajes de error y stop.....	124
5.10 Especificaciones Técnicas.....	126
Bibliografía.....	127
Recomendaciones.....	130
Anexos	
Apéndice A.	
Sistema Controlador.....	133
Apéndice B	
Hojas de Datos.....	139
Rs-485.....	157
Uso de Ram estática no volátil.....	162
Apéndice C	
Flujograma.....	165
Programa.....	181
Set de instrucciones.....	194

## ÍNDICE DE ILUSTRACIONES

Diagrama en bloques del microcontrolador de Intel.....	31
Arquitectura del 8051.....	36
Estructura interna de puertos.....	37
Sumario de los espacios de memoria del 8051.....	41
Mapeo de memoria interna del microcontrolador.....	42
Uso del registro B.....	46
Uso del puntero de pila (Stack Pointer).....	47
Registro PCON.....	49
Multiplexación de memoria externa.....	51
Accesando memoria externa de código.....	51
Tiempo de lectura de la memoria de código externo.....	52
Instrucción MOVX.....	53
Conexión de Ram externa.....	54
Decodificación de dirección.....	55
Fuentes de reloj.....	59
Modo de operación 0.....	62
Selección de frecuencia.....	64

Registro IE.....	66
Estructura de las interrupciones en la familia MCS-51.....	67
Clasificación general de las instrucciones.....	68
Circuito de reloj para SBC utilizando MCS-51.....	75
Reset del SBC.....	78
Decodificador 74LS138.....	79
Utilización del 74LS373.....	81
Conexión típica de memorias en el 8051 sin el uso de decodificador.....	85
Relación eléctrica entre RS-232C y comunicación TTL.....	86
Mapeo de memoria.....	89
Etapas de potencia.....	92
Conexión de interface de teclado.....	95
Conexión del LCD.....	98
Pulsador de reset del sistema mínimo DTK-51.....	103
Header del DTK-51.....	104
Conexión de Energía.....	105
Configuración del conector DB9 para RS-232C.....	106
Diagrama de la Planta.....	116

**DEDICATORIA:****A DIOS:**

Por ser mi fortaleza. Este título pertenece a tu Gloria Padre. Porque sin tu apoyo nada de esto sería posible ayer, ahora y siempre.

**A MIS PADRES:**

Doris Celina Palma de Contreras y César Ernesto Contreras Cerna, por sacrificarse tanto por mis estudios. Ustedes más que ninguna otra persona merecen esta bendición.

**A MIS HERMANAS:**

Berta Ernestina y Doris Lorena Contreras Palma, por que siempre creyeron en mí.

**A KARLA:**

Por ser simplemente tú.

**A VICTOR:**

Por creer y depositar su confiar desde principio a fin de la tesis en mí.

**A MIS SERES QUERIDOS:**

A mis familiares, a mis amigos casi hermanos Marlon, Balmore, Carlos, Alex y a todos aquellos que de uno u otra forma alentaron y apoyaron mi carrera.

A ELLOS DEDICO ESTE TRIUNFO

César Ernesto Contreras Palma

## **DEDICATORIA**

### **A DIOS TODO PODEROSO**

Quien en su infinita misericordia permitió realizar este esfuerzo, sustentándome y proveyéndome en todo.

### **A MIS PADRES:**

Ana Victoria, por sus oraciones diarias; Milagros y Carlos Rivas, por su sacrificio, paciencia y guía para realizar esta carrera.

### **A MI COMPAÑERO Y AMIGO CESAR**

A quién agradezco sinceramente por su apoyo y amistad, sin las cuales no se hubiera podido terminar esta tesis.

### **A LINDA MARIA**

Mi amada, por su cariño, motivación y suficiente comprensión.

### **A MI ABUELA**

A la que siempre recordaré y llevaré en mi recuerdo.

Por todo Gracias Señor....

“ No a nosotros oh Señor, no ha nosotros; sino a tu nombre da gloria. Por tu misericordia y tu verdad” Sal. 115:1

## **AGRADECIMIENTOS:**

César Contreras y Víctor Cuchillac desean agradecer la ayuda, asistencia y apoyo brindado a las siguientes personas, reconociendo que sin su colaboración no hubiese sido posible esta tesis.

- Al asesor Ing. Saúl Velasco, por creer en este tema, por su tiempo y conocimientos.
- A los jurados Ing. Alexander Guzmán e Ing. Federico Laínez, por su disponibilidad y sugerencias.
- Al Ing. Carlos Ramírez, por su tiempo, conocimientos y apoyo.
- Al Ing. Walter Aguilar, Ing. Ricardo Medrano e Ing. Marcos Tulio Portillo, por su apoyo material.
- A la Lic. Linda María Bonilla, por su apoyo espiritual y material.
- Al Ing. Renato Leiva, por su apoyo técnico y material.
- Al señor Ricardo Amaya, por su tiempo y colaboración.

A estas personas y a todos los que siempre creyeron en esta tesis,

**MUCHAS GRACIAS**

## GLOSARIO

**Alfanumérico**: Cadenas de caracteres que constan de cualquier combinación de letras o números.

**Analógico**: Un valor numérico que representa cantidades medibles, tales como temperatura, peso, presión. Etc. Compare con Digital.

**And**: Una operación booleana que produce una salida verdadera sólo cuando todas las condiciones son Verdaderas y una salida Falsa si cualquiera de las condiciones es falsa.

**Aplicación**: Una máquina o proceso que requiere un sistema de control para su operación.

**Base de tiempo**: La unidad de tiempo usada por un temporizador para registrar eventos. Una base de tiempo de un segundo tiene una precisión al segundo más cercano. Según el controlador, dependerá la base de tiempo a operar.

**BCD**: Decimal codificado en binario - un sistema binario en el que cada dígito decimal de 0 a 9 se representa mediante cuatro dígitos binarios (bits).

**Binario**: Un sistema de numeración que sólo usa los dígitos 0 y 1. También se conoce como base 2.

**Bit**: La ubicación de almacenamiento más pequeña en memoria. Un bit contiene ya sea un 1 o un 0.

**Bus**: Un grupo de líneas usado para la transmisión de datos o control.

**Byte**: Un grupo de bits adyacentes generalmente operados como una unidad, como cuando se transfieren a o desde la memoria. Hay ocho bits en un byte. Un byte es capaz de almacenar y mostrar un equivalente numérico entre 0 y 255.

**Ciclo**: Una sola secuencia de operación.

**Contador**: Un dispositivo o instrucción de software que cuenta las apariciones de algún evento. Pueden ser impulsos que resulten tales como cierres de interruptores u otros eventos discretos.

**Controlador**: Un dispositivo programable donde residen la fuente de alimentación, procesador e interfaces de E/S, capaz de controlar o monitorear otro dispositivo.

**CPU**: Unidad central de procesamiento - La sección que hace decisiones de un controlador y que ejecuta las instrucciones contenidas en el programa del usuario.

**Dato**: En un controlador, un término general para cualquier tipo de información.

**Diagnóstico**: La detección e indicación de errores o malfuncionamiento.

**Digital**: Información presentada como un valor discreto, 1 ó 0.

**Dirección**: Una ubicación de memoria única, identificada mediante un carácter alfanumérico.

**Dispositivo de entrada**: Un dispositivo, tal como un botón pulsador, sensor o un interruptor de algún tipo que suministra señales al controlador.

**Dispositivo de salida:** Un dispositivo, tal como una luz piloto o una bobina de un arrancador de motor, que es gobernada por el controlador.

**E/S** (Entrada/Salida): Consta de dispositivos que proporcionan datos al (entrada) y reciben datos del controlador (salida).

**EIA:** Asociación de industrias Electrónicas. Agencia que establece estándares eléctricos/electrónicos como RS-232C, RS-422A, RS-485, etc.

**EPROM:** Memoria de lectura solamente programable y borrable mediante luz ultravioleta. En ellas residen programas, bases de datos, constantes, etc. No es alterable por el usuario.

**Fuente de alimentación:** Circuito eléctrico que filtra, acondiciona y suministra los voltajes correctos para los componentes y los circuitos del sistema.

**Hardware:** Incluye todos los componentes físicos del sistema de control, incluyendo el controlador programable, periféricos y cableado de interconexión.

**IC:** Siglas de "Integrated Circuit", se denomina a cualquier circuito integrado.

**Instrucción:** Un comando que define una operación a ser ejecutada por el controlador.

**Latch:** Significa enclavar. Dispositivo electrónico que se encarga de retener un valor digital media vez se suministren las señales de habilitación requeridas por éste mismo.

**LCD:** Siglas de "Liquid Crystal Display" Significa pantalla de cristal líquido.

**LED**: Diodo emisor de luz. Un diodo semiconductor, el cual emite luz cuando pasa una corriente. Los LED se usan como indicadores de diagnóstico en varios componentes de hardware del controlador.

**LIFO**: Ultimo en entrar es el primero en salir. Tipo de memoria.

**Manejador (Buffer)**: Circuito manejador con función de aislamiento entre dos de niveles diferentes como velocidad, corriente, etc.

**Memoria de usuario (aplicación)**: La porción del total de la memoria del sistema dedicada al almacenamiento de información proveniente del usuario o procesos donde sus resultados son necesarios para éste.

**Memoria de sistema**: El espacio total de memoria dentro del controlador que involucra base de datos, sistema operativo, etc.

**Memónico**: Un término fácil de recordar que se usa para representar un conjunto complejo o largo de información.

**Operadores Booleanos**: Operadores lógicos tales como Y (AND) y O (OR) que pueden usarse solos o en combinación para formar instrucciones lógicas o circuitos. Estas instrucciones deben tener una respuesta de salida la cual es verdadera o falsa.

**Pin**: Terminal de un dispositivo o circuito integrado.

**Periférico**: Dispositivos externos que están conectados mediante un puerto de comunicación (paralelo o serial) al controlador, generalmente para intercambio de datos, interface del operador, etc.

**Programa**: Conjunto de instrucciones almacenadas en memoria que es ejecutable en un orden predeterminado por la unidad central de procesamiento.

**RAM**: Memoria de acceso aleatorio. Comúnmente denominada memoria de lectura/escritura ya que puede ser escrita o leída.

**Registro**: Un espacio de almacenamiento temporal para varios tipos de información y datos, tales como valores para contadores, temporizadores, etc.

**SBC**: Controlador de una sola tarjeta (Single board controller). Circuito electrónico diseñado para controlar una aplicación específica.

**Sistema**: Conjunto de uno o más microcontroladores, que junto con dispositivos de E/S, memorias, software, periféricos, terminales, etc., proporcionan una manera de ejecutar el procesamiento de información para el control de máquinas y procesos.

**Software**: Conjunto de instrucciones propias del microcontrolador (para el caso) que sirve para desarrollar un lógica o secuencia.

**Transreceptor**: Dispositivo capaz de realizar las operaciones de manejador y receptor.

**UART**: Transmisor-Receptor Universal Asíncrono. Sirve como interface entre paralelo a serial y viceversa.

## INTRODUCCIÓN

Una empresa industrial de cualquier índole, como productora de bienes, se encuentra sometida a un ambiente altamente competitivo no sólo en cuanto que sirve al mercado estatal, sino en cuanto a que extiende su actividad a nivel internacional. Para poder crecer, o aún para sobrevivir, se encuentra con la necesidad de adaptarse con prontitud a los requerimientos del mercado, intentando estar un paso antes de sus competidores.

Hasta nuestros días, la automatización de máquinas y procesos ha demostrado ser la solución para mejorar la productividad, disminución de costos, reducción de desperdicios y obtención de altos grados de calidad. La automatización de una máquina o proceso consiste en la incorporación de dispositivos tecnológicos de decisión, sensores y actuadores que se encargan del control.

Hoy en día es verdaderamente sorprendente el número de diferentes sistemas de control que hay para la automatización de una fábrica de desinfectantes líquidos. Lo anterior hace muy difícil el proceso de selección del sistema a usar. Esta tarea se complicará aún más en el futuro al irse desarrollando dispositivos de control, equipos y sistemas más sofisticados, de acuerdo con los avances tecnológicos en esta área y las necesidades de ésta.

Actualmente las ceras y desinfectantes líquidos se fabrican básicamente mezclando manualmente los componentes o materias primas en un solvente que constituye más del cincuenta por ciento del producto, que para el caso es agua.

El sector que conforma la industria de desinfectantes y ceras líquidas en el país, lo constituye en su mayoría la micro empresa, la cual se instala en pequeños establecimientos o casas, por lo que no se prevee las condiciones ni técnicas mínimas necesarias para la manipulación de químicos, el manejo de desechos y

El sistema a seleccionar, debe romper el desafío técnico que consiste en integrar no solo el envasado, sino el dosificado y manejo de los componentes del producto desde el inicio del proceso, además de preparar al sector industrial de desinfectantes del país contra los retos que presentarán en un futuro muy cercano la globalización y la integración a tratados regionales de libre comercio que actualmente se impulsa.

Tomando en cuenta lo anterior, el controlador a diseñar pretende solventar las necesidades actuales y futuras del empresario dedicado a la elaboración de ceras y desinfectantes, cumpliendo los requerimientos tecnológicos actuales, con el fin de que no sea una solución obsoleta, un desperdicio de recursos y tiempo o una inversión mal justificada. El presente trabajo pretende brindar una alternativa que encierre las soluciones de los problemas que más aquejan a la industria nacional, así como ayudar a ésta en el proceso de incorporación a las Buenas Prácticas de Manufactura, apoyado de los conocimientos más modernos al servicio de la automatización.

El estudio se enfoca en las problemáticas de seguridad industrial, el alto desperdicio de producto fabricado, los tiempos muertos, control exacto en el proceso de dosificación y adecuado presupuesto según las posibilidades inherentes del sector o gremio productor de ceras y desinfectantes; sin descuidar las tendencias futuras de la tecnología y del mercado nacional y mundial.

Este trabajo busca ser un aporte para el gremio empresarial productor de desinfectantes, a la tecnificación y automatización de los procesos industriales a nivel nacional.

En el capítulo I, se hace un breve estudio del proceso de fabricación de desinfectantes y ceras para piso, con el fin de sugerir una propuesta que permita el acoplar un sistema completo automático para la fabricación de tales productos.

En el capítulo II, se definen los criterios para la automatización del proceso descrito en el capítulo anterior o similares. Se define las entradas y salidas, requerimientos tanto ambiente o entorno donde el sistema debe desenvolverse. Por último se resume las características del sistema controlador.

En el capítulo III, se detallan brevemente las partes fundamentales de los sistemas con microcontroladores, se describe brevemente la familia de microcontrolador MCS-51 (registros, banderas, timers, etc.). Este capítulo nace principalmente para minimizar la limitante nacional de escasa bibliografía y la inexistencia de técnicas propias en su desarrollo e implementación.

El capítulo IV detalla las partes fundamentales de los sistemas mínimos con microcontroladores, se describe su funcionamiento y de lo necesario para el buen desenvolvimiento de la planta o proceso. Llegándose así a un primer diseño, el que abrió el camino para construir el controlador. Luego se establecen los circuitos finales que se han implementado y se describe los elementos que lo conforman y el sistema de software.

El capítulo V es un manual de usuario, tanto para realizar la comunicación serial, cuidados con el sistema completo de control de la planta o proceso, recomendaciones para su mantenimiento y hoja con los datos técnicos del controlador.

En los capítulos subsiguientes se plantean conclusiones del trabajo, recomendaciones, modificaciones y demás sugerencias para explotar al máximo el módulo y todo lo que él representa.

## CAPITULO I

### DESCRIPCIÓN DEL PROCESO DE FABRICACIÓN DE DESINFECTANTES

#### 1.1 Estado actual

Ahora, a pesar del avance tecnológico alcanzado, en el país, existen procesos de producción, los cuales debido a sus características se realizan manualmente, tal es el caso de la elaboración de desinfectantes para pisos, ceras, jabones líquidos, limpiadores para superficies ( barnices para madera, limpiadores de vidrios), etc.

Si bien es cierto que estos procesos son sencillos, en la mayoría de ellos se presentan pérdidas de materia prima, dosificaciones incorrectas, atrasos en la producción, etc.

Actualmente, debido a la presencia más activa de control ecológico en los desechos, tratados de libre comercio con naciones cuyos productos se manufacturan con costos menores a los nacionales, y con mayores volúmenes que reducen el precio de venta, la eminente globalización de las economías, reducción de aranceles, estandarización, normalización mundial, la tecnificación y automatización de casi toda actividad industrial, obliga a las empresas nacionales a producir con más calidad, en menor tiempo, y con una mínima cantidad de desperdicio.

En la elaboración de desinfectantes y ceras de la industria salvadoreña, las microempresas poseen la mayor parte en el mercado. Y tales empresas se han colocado en esta situación, debido a que estas poseen una pequeña inversión tanto en la infraestructura como en la fabricación de sus productos. La fabricación artesanal en el país es la más común, debido al tipo de mercado nacional y lo barato de sus productos.

Un factor importante en la no automatización de los procesos industriales es que la implementación de los mismos demanda una fuerte inversión por la parte del empresario.

En el país sólo se cuenta con dos empresas muy reconocidas en esta área, por lo que los honorarios son elevados.

Otro factor involucrado, es la seguridad del obrero. Al realizar procesos con pocas medidas de prevención de accidentes y riesgos profesionales, aumentando la probabilidad de que ocurran, cuyo efecto puede generar hasta una incapacidad permanente o total para el empleado, el soportar los costos directos e indirectos del accidente profesional y perdiendo el producto que al momento del accidente se procesaba.

## **1.2 Proceso de fabricación**

La mayoría de industrias dedicadas a la fabricación de desinfectantes, ceras, jabones líquidos, limpiadores para superficies y lejías nacen debido a la pequeña inversión requerida tanto en infraestructura como materia prima.

El proceso de producción se realiza utilizando probetas, beakers, embudos, etc. en ambientes poco adecuados tanto para el obrero como las personas que habitan a los alrededores de la planta.

La secuencia de fabricación utilizada consiste:

1. Mezclar Ingrediente 1
2. Mezclar Ingrediente 2
3. Homogenizar
4. Mezclar Ingrediente 3
5. Homogenizar
6. Mezclar Ingrediente 4
- 7 Homogenizar

8. Mezclar Aroma

## 9. Homogenizar

## 10. Envasar

Los operarios toman probetas con las cantidades de cada ingrediente a mezclar, estos utilizan probetas con graduaciones cada 10 ml, por lo que se deja mucho al aprecio de éstos si se ha llegado a la marca o no del volumen requerido, conforme el tiempo transcurre, el cansancio en la vista hace que se permita errores de hasta 10 ml; si a todo esto, se requiere combinar seis ingredientes, es probable errores de hasta 60 ml por galón, lo que genera pérdidas elevadas y niveles de calidad poco constantes.

Al mezclar los ingredientes, se recurre a la agitación o mezcla de todos los elementos de la receta hasta que se obtiene un producto final homogéneo. Todo esto transcurre desde el primer elemento mezclado hasta su agitación en un lapso de dos minutos promedio, requiriendo un tiempo de 100 minutos (hora cuarenta minutos) la producción de 50 galones (un barril). Cabe destacar que la emanación de sustancias produce en el operario que sufra de cierto tipo de alucinaciones que acentúan tanto tiempos largos de mezclado, tiempos muertos, desperdicio como baja precisión. La secuencia es repetitiva, prolongándose si se incrementan los ingredientes. Esto permite introducir un controlador electrónico que realice la tarea en recipientes cerrados logrando en menor tiempo y sin emanaciones nocivas mejores resultados.

Los tiempos de mezclado no son de manera crítica, es decir, que si permanece un determinado tiempo sin mezclar produzca una reacción que descomponga alguno o algunos de los ingredientes en un resultado no deseado. Esto en ningún momento argumenta que se pueda permanecer horas sin mezclar, sencillamente se refiere a que el controlador no requeriría una regulación específica para manejo de materia prima, únicamente una secuencia continua. Si es necesario tener el proceso, la receta misma no es adulterada, por lo que se pueden hacer

Actualmente se pretende impulsar a nivel del gremio de fabricantes de desinfectantes, un sistema automático que realice la función en condiciones seguras, precisas y con menos tiempo.

Por lo que se diseña actualmente una planta piloto, compuesta de sensores de flujo tipo turbina, bombas, etc., con el fin de solventar la problemática. El controlador debe adaptarse a este plan, con el fin de obtener una propuesta barata y que permita que la gran mayoría de industrias dedicadas a producir desinfectantes puedan aplicarlo.

Algunos ejemplos de recetas y secuencia a continuación (todas son incluidas en el anexo):

**JABÓN . FORTEX (DESENGRASADOR)**  
**JABÓN LÍQUIDO INDUSTRIAL CONCENTRADO.**

### **DESCRIPCIÓN TÉCNICA**

Concentrado industrial para trabajos pesados en limpieza sobre la superficie de motores, maquinarias, planchas, hornillas y cocinas. Sus componentes no reaccionan con el plástico.

### **ADVERTENCIA**

- Evite el contacto con la piel, use guantes para su aplicación.
- En caso de ingestión accidental induzca al vómito y llame al médico.
- En caso de contacto con los ojos o partes delicadas de la piel enjuáguese con abundante agua.

### **INGREDIENTES**

Detergentes industriales, arrancagrasa, estabilizantes, colorante y agua.

a) Eter láurico de polietilenglicol	1.000 %
b) Carbonato de calcio	0.500 %
c) Carbonato de sodio	0.500 %
d) Carbómero	0.500 %
e) Colorante	0.001 %
f) Aroma	cs
g) Agua csp	100.000 %

- 1) Incorporar d) en 50 % de g) y homogenizar.
- 2) Incorporar b) y e) a 1) con agitación hasta que se encuentre bien mezclado.
- 3) Adicionar a) en 2) con agitación.
- 4) Mezclar e) y f) a 3) y homogenizar.
- 5) Llevar a volumen con g) homogenizando.

## **PRESENTACIONES**

Galón, Medio galón y litro.

## **COLORO . LIQUIDO**

BLANQUEADOR, DESINFECTANTE Y DESODORIZANTE.

## **DESCRIPCIÓN TÉCNICA**

Es una solución clorada elaborada especialmente para eliminar la suciedad, hongos, moho y malos olores en baños y servicios sanitarios. Sus componentes no deforman el plástico.

## **ADVERTENCIA**

- En caso de ingestión accidental induzca al vómito y consulte al médico.
- En caso de contacto con los ojos o partes delicadas de

la piel enjuáguese con abundante agua.

- En altas concentraciones evitar contacto con la piel.

## **INGREDIENTES**

Hipoclorito de sodio, y agua.

a) Hipoclorito de sodio	0.5 %
b) Agua csp	100.0 %

1) Disolver a) en 50 % de b) y homogenizar.

2) Adicionar b) hasta llevar a volumen.

## **PRESENTACIONES**

Galón y Medio Galón

**NOTA: EL CLORO LIQUIDO NO ES LEJIA.**

### **1.3 Propuesta**

Se pretende generar en base a este proyecto una propuesta, utilizando sensores de poco costo (baratos en relación a los tradicionales), ductería y demás sistema de inyección de bajo costo y poco sofisticados. El sistema será lo más hermético posible para evitar emanaciones dañinas, fuga y desperdicio de materia prima. El cerebro o controlador debe reducir tiempos muertos, permita realizar una infinidad de recetas y controle el dosificado con un alto grado de precisión.

El sistema permitirá el acople a futuro de etapa de llenado o envasado automático. Esto significa que permitirá tener un proceso automatizado completo desde la mezcla hasta el envasado.

Para mejor manejo de sustancias, se usará el concepto de premezcla, es decir, aquella sustancia que por su poca cantidad o presencia en el producto final, sea combinada con otras de similar cantidad para que juntas formen un mayor volumen, por lo que no será necesario tener contacto en ninguna etapa del proceso.

El controlador manejará la información proveniente de un sensor de flujo, cuyo dato determinará cuándo detener la inyección y el cambio para el siguiente ingrediente.

Aunque a un inicio parezca poco preciso, el control de pulsos permite precisiones hasta de 0.1% por ciento, aplicado especialmente en los sistemas de bombeo que encontramos comúnmente en las gasolineras.

Es así, que se configura una circuitería no muy complicada pero competitiva, lo que hace atractiva la implementación.

Un microcontrolador permitirá reducir aún más la parte de hardware, y permitiría obtener alto grado de control.

La selección del microcontrolador específico entre los existentes en el mercado, se basa en los criterios de:

- RAM integrada al chip para 30 localidades por lo menos
- Puertos integrados direccionables a bit
- Instrucciones de manejo de bit
- Velocidad acorde al muestreo de las señales del proceso
- Bajo consumo de potencia
- Accesible en el mercado
- Bajo costo
- De conocido uso

Actualmente en el mercado, se encuentran dos modelos ampliamente usados, tanto el Motorola 68HC11 como el Intel MCS-51.

La familia MCS-51 presenta todas las características descritas anteriormente, además presenta salidas de colector abierto, lo que lo hace más robusto en situaciones adversas. Los ingenieros de diseño de muchas corporaciones han seleccionado para sus equipos a este microcontrolador, para mencionar tenemos Red Lion, Foxboro, Toshiba, Kendall, etc.

Para una clara idea de lo que involucra la automatización del proceso, se presenta la siguiente cotización:

<b>DISPOSITIVO</b>	<b>PRECIO</b>	<b>MARCA</b>	<b>MODELO</b>	<b>VOLTAJE</b>
<b>BOMBA</b>	<b>\$300</b>	<b>OMEGA</b>	<b>FPUD311</b>	<b>110 VAC / ½ HP</b>
<b>SENSOR FLUJO</b>	<b>\$70</b>	<b>OMEGA</b>	<b>FPR132</b>	<b>110 VAC / 70 mA</b>
<b>VALVULAS</b>	<b>\$120</b>	<b>DAYTON</b>		<b>110 VAC</b>
<b>CONTROLADOR</b>	<b>\$57</b>			<b>110 VAC</b>
<b>MOTOR MIX</b>	<b>\$105</b>	<b>GENERAL ELECTRIC</b>	<b>5KC3511N70HX</b>	<b>110 VAC ¼ HP</b>
	<b>\$652</b>			

\* **NOTA:** La bomba es para manipulación de líquidos, construcción para soportar el ambiente industrial, resistente a la corrosión y permite viscosidades hasta 100 cps y temperatura máxima de 71.1° C. El sensor de flujo permite flujo de hasta 12 gpm, soporta hasta 100 psi y 82.2° C. Las válvulas tienen alta resistencia a químicos y temperatura máxima de 60° C. Todos estos dispositivos tienen recubrimiento de PVC.

<b>DISPOSITIVO</b>	<b>PRECIO</b>	<b>MARCA</b>	<b>MODELO</b>	<b>VOLTAJE</b>
<b>BANDA TRANSP.</b>	<b>\$105</b>	<b>GENERAL ELECTRIC</b>	<b>5KC3511N70HX</b>	<b>110 VAC ¼ HP</b>

<b>BERO CAPACTV. \$39</b>	<b>OMRON</b>	<b>E2K-X8ME1</b>	<b>110 VAC</b>
<b>VALVULA 3 vias \$38</b>	<b>DAYTON</b>		<b>110 VAC</b>
<b>\$182</b>			

Para un total de \$652 (¢5740) sin etapa automática de envasado. Total de \$834 (¢7340) con etapa de envasado automático.

## CAPITULO II

### METODOLOGÍA DE LA AUTOMATIZACIÓN

#### 2.1 Método para automatizar

El proceso presenta las siguientes características que influyeron para la automatización:

- \* Una operación repetitiva
- \* Operaciones accionadas por eventos
- \* Operaciones accionadas por tiempo
- \* Requisitos de introducción de datos

Para poder seleccionar un sistema de control, se deben determinar los requisitos de la aplicación, por lo que se definieron de la siguiente forma:

- \* Que el sistema controlador (SBC) sea económicamente viable para el sector químico industrial al evaluarse con los sistemas existentes.
- \* Confiabilidad Que permita un alto grado de exactitud, sin fallas en un largo período, mejorando así, los factores de competitividad.
- \* Flexibilidad Que las modificaciones del dosificado puedan hacerse mediante teclado y puedan almacenarse como receta en memoria.
- \* Funciones avanzadas Que permita realizar una variedad de tareas de control, desde acciones repetitivas hasta introducción vía teclado de datos.
- \* Comunicación Permita comunicación con la interface de operador, otros controladores (SBC) y el intercambio de datos con ordenadores.

\* **Velocidad** Que las actividades mejoren no sólo en calidad y reducción de desperdicio, sino en tiempo de procesado.

\* **Monitoreo** La capacidad de localizar problemas para que puedan ser corregidos fácilmente por el operario.

Las consideraciones de control para cualquier aplicación son

1. Definir la operación del sistema (desarrollado en el capítulo 1).

2. Definir los requisitos de dispositivos de entrada y salida.

3. La necesidad de operaciones especiales además de las de lógica discreta (On/Off o Encendido/Apagado), incluyendo:

\*Temporización

\*Conteo (baja y alta velocidad)

\*Secuenciales

\*Cálculo de datos de modo aritméticos

Las aplicaciones a menudo requieren operaciones adicionales a la sencilla lógica discreta (Encendido/Apagado). Estas funciones avanzadas incluyen temporización, conteo, secuencias, matemática, comparación y muchas otras operaciones que involucran manejo y cálculo de datos.

Antes de la concepción del autómata o controlador, es necesario listar las funciones avanzadas requeridas y como éstas serán usadas. Entre las que se pueden listar están las siguientes:

<b>Función</b>	<b>Uso</b>
Contador Progresivo	Cuenta ascendente de eventos
Contador Regresivo	Cuenta descendente de eventos
Temporización	Intervalos de tiempo
PID	Control Proporcional- Integral-Derivativo
Multiplexor	Conmutación digital
Decodificador	Convertidor de múltiples entradas a salida codificada (BCD-HEX y viceversa, ASCII)
Fuzzy	Control difuso
Operaciones Aritméticas	Suma, multiplicación, etc.

Es necesario aclarar que no siempre es prescindible que un controlador contenga todas estas funciones especiales o más, para ello se hace la selección del controlador y los requerimientos del proceso a automatizar.

4. Requisitos eléctricos para las entradas, salidas y alimentación eléctrica del sistema. Cuando se determinan los requisitos eléctricos de un sistema, es de considerar tres aspectos:

- 1) Energía eléctrica para el sistema de control.
- 2) Alimentación de los dispositivos de entrada.
- 3) Alimentación de los dispositivos de salida

5. Velocidad de operación del sistema de control.

6. Comunicación (Necesidad de compartir datos).

7 Si el sistema necesita control o interacción del operador.

8. Entorno físico en el cual el sistema de control estará ubicado.

## 2.2 Definición de entradas y salidas.

Sin importar el tipo de sistema o dispositivo de control que finalmente se seleccione, el primer paso a considerar para aproximarse a una situación de control es especificar los requisitos de la aplicación. Esto implica determinar los requisitos de entrada y salida del módulo.

Para ello se tiene una lista de variables, las que se clasifican según su funcionamiento, es decir, si éstos detectan o sensan la ocurrencia de un fenómeno físico, la presencia de un estado etc.

Para poder identificar una entrada, simplemente averiguar si detectan la ocurrencia de un evento; una salida es aquella la cual su condición acciona, enciende, apaga, controla un actuador, relé, etc.

Lista de requisitos de entradas y salidas:

<u>Función de Entrada</u>	<u>Dispositivo</u>
Solicitud de arranque	Pulsador
Solicitud de paro	Pulsador
Pulso del medidor flujo	Medidor flujo turbina
Indicador de nivel	Boya
Presencia envase	Bero Capacitivo*

\* Opcional del industrial que utilice el controlador

<u>Función de Salida</u>	<u>Dispositivo</u>
Verter ingrediente 1	Relé para válvula 1
Verter ingrediente 2	Relé para válvula 2

Verter ingrediente 3	Relé para válvula 3
Verter ingrediente 4	Relé para válvula 4
Verter ingrediente 5	Relé para válvula 5
Verter ingrediente 6	Relé para válvula 6
Anuncio de peligro o alarma	Bocina
Mezclar	Motor del mezclador
Transportar	Motor de banda transportador*

\* Opcional del industrial que utilice el controlador

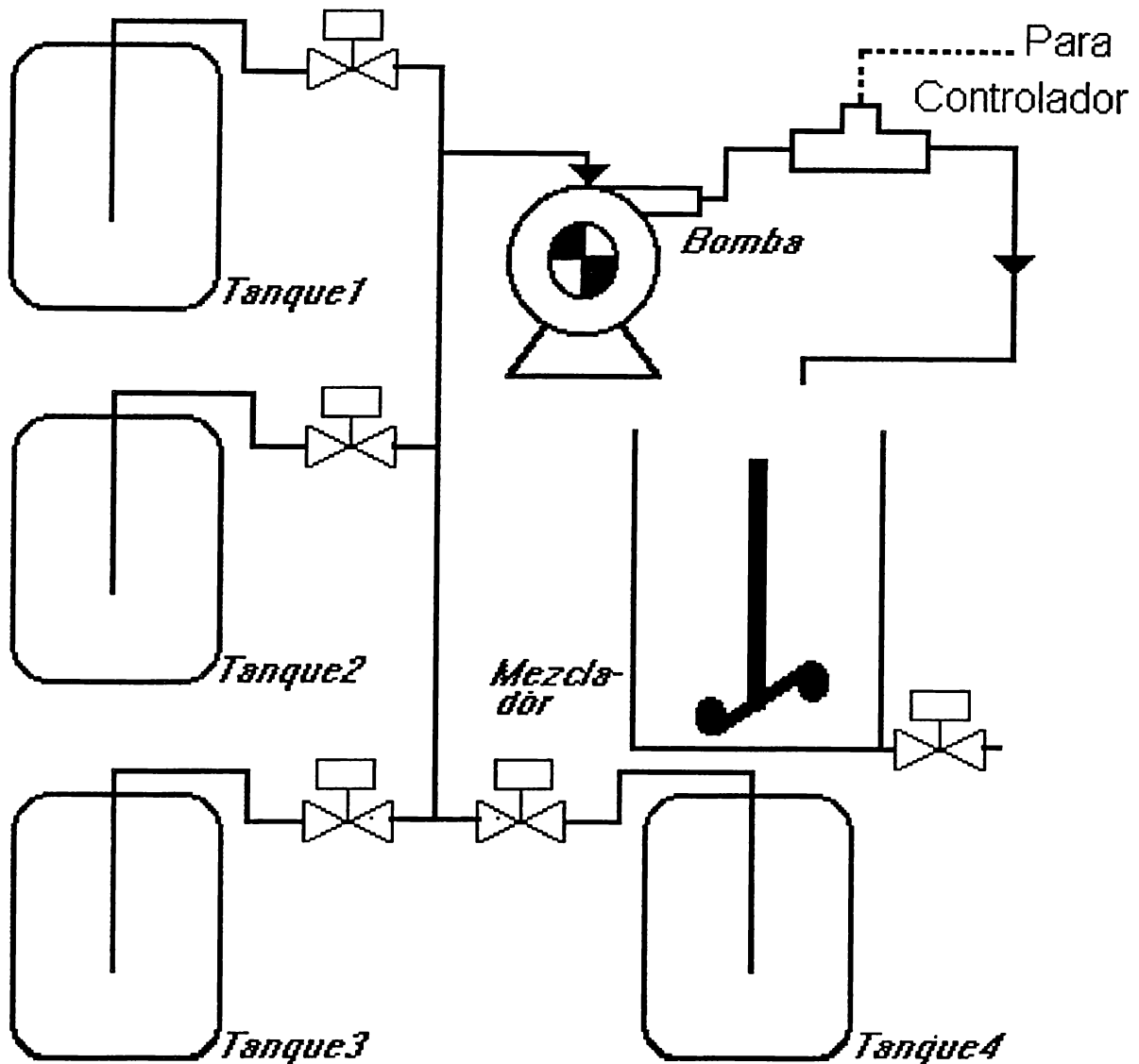
La planta constará de 4 recipientes conteniendo los ingredientes a mezclar. Todos los recipientes dependerán de una sola bomba, la que proporcionará un flujo de 6 galones por minuto. El sensor de flujo será común para cada ingrediente de la mezcla, lo que indica que se hará un proceso secuencial para la preparación del lote o batch de producción. Los pulsos aunque provienen de un sólo sensor, se asignará secuencialmente al ingrediente a mezclar, lo que determinará la dosis que se aplicará. Es necesario determinar el medio de envasado o presentación, por lo que se debe determinar si se almacenará en galones, litros o barriles.

El envasado será realizado por un operario, en recipientes con capacidad de un galón o litro. Aunque no es prescindible, el controlador permitirá la opción de mover una banda transportadora, detectar la presencia del recipiente y envasar la cantidad prevista automáticamente, si en un futuro así se desea. Si la presentación final son barriles se aplicará por medio de una válvula directamente de la mezcladora.

La mezcladora o mixer se encarga de combinar los ingredientes en una producto homogéneo, no siendo importante desde el momento que se vierte el primer ingrediente, que la mezcladora funcione. Al terminar de verter el último ingrediente,

éste será accionado por un período de un minuto. Luego de mezclar, se podrá disponer del producto.

### 2.3 Esquema de la planta.



### 2.4 Especificaciones para materializar el módulo controlador (mediante SBC).

Se ha detallado en el capítulo 1 y parte del capítulo 2, tanto el proceso como las entradas y salidas requeridas, pero falta analizar los requerimientos de control

mencionados en la sección anterior, para ello, definiremos nuestro módulo de control orientado a la constitución del microcontrolador y sus interfaces.

El SBC basado en microcontrolador, provee de las características y requerimientos necesarios para poder materializar un sistema capaz de controlar el proceso de dosificación y mezclado de desinfectantes o similares.

- Requisitos de funciones avanzadas:

Es necesario:

- \* Timer o temporizador para efectos de retardo de tiempo, tiempos de espera para aceptación de datos, estabilización del producto, etc.

- \* Contadores, los cuales permitirán obtener una relación de volumen constante, con lo que se pretende medir de forma precisa las cantidades vertidas de líquidos en base a las cantidades requeridas por la receta.

- \* Operaciones aritméticas de suma, resta, multiplicación y división, cuyos resultados arrojarán los datos necesarios de operación, con el fin de regular la cantidad a verter o a mezclar, las unidades volumétricas y respectivas conversión, etc.

- Requerimientos eléctricos para las entradas, salidas y alimentación.

Por lo tanto, es necesario cuestionar lo siguiente:

¿Tipo de energía eléctrica disponible?

La fuente de alimentación será provista por la alimentación residencial regulada a nivel nacional por la distribuidora eléctrica, para el caso, 110 VAC a 60 Hz.

¿Alimentación eléctrica utilizada por los dispositivos de campo?

Los dispositivos electrónicos encontrados a nivel comercial consumen corriente directa, sus valores nominales más comunes son +5, -5, +12, y -12 VDC. Para la

aplicación es necesario limitar la selección a +5 VDC, esto se debe a principalmente a la simplificación del circuito de alimentación que implicaría un voltaje standard.

Como conclusión, la fuente de alimentación proveerá 110 VAC, de la cual se convertirá a +5 VDC para todo aquel elemento que lo requiera. Por lo tanto, los elementos de conmutación deberán soportar dichos voltajes en sus terminales.

La fuente de alimentación, del controlador, tendrá la capacidad de soportar las deficiencias de línea que presenta las distribuidoras a nivel nacional. Para mantener costos bajos se implementará mediante circuitos integrados, los cuales presentan otras ventajas tales como poco espacio utilizado, poca circuitería y alto grado de confiabilidad.

Para facilitar la organización del sistema de control, se resumen los requisitos eléctricos.

<b>Función de entrada</b>	<b>Dispositivo</b>	<b>Voltaje</b>
Solicitud de arranque	Pulsador	5 VDC
Solicitud de paro	Pulsador	5 VDC
Pulso medidor flujo	Medidor flujo	5 VDC/Pulso TTL
Indicador de nivel	Boya	24 VDC/Entre terminales de switch
Presencia de Envase	Bero capacitivo	24 VDC*

<b>Función de salida</b>	<b>Dispositivo</b>	<b>Voltaje</b>
Verter ingrediente 1	Relé para válvula 1	110 VAC
Verter ingrediente 2	Relé para válvula 2	110 VAC
Verter ingrediente 3	Relé para válvula 3	110 VAC
Verter ingrediente 4	Relé para válvula 4	110 VAC
Verter ingrediente 5	Relé para válvula 5	110 VAC
Verter ingrediente 6	Relé para válvula 6	110 VAC

Anuncio peligro (alarma)	Bocina	5 VDC
Mezclar	Motor del mezclador	110 VAC
Transportar	Motor transportador	110 VAC*

\* Opcional del industrial que utilice el controlador

Funciones avanzadas	Dispositivo	Voltaje
Contador	T/C 0	5 VDC/Pulso TTL
Generador de Baud Rate	T/C 1	
Temporizador	T/C 2	
+,-,x,/	Implícito	

Los señalados con "---" indican a elementos internos en el microcontrolador.

- Velocidad de operación.

Para la determinación de la velocidad de operación, es importante considerar los siguientes puntos:

\*¿Qué tan rápido ocurre el proceso u opera la máquina?

\*¿Existencia operaciones o eventos "de tiempo crítico" que tienen que detectarse?

\*Lapso de tiempo que ocurre la acción más rápida (detectar y activar la salida respectiva)

\*¿Tiene el sistema que controlar pulsos provenientes de un codificador o medidor de flujo?

El sistema de control seleccionado necesita satisfacer los requisitos de velocidad del proceso de la máquina.

Claramente, el sistema de control del proceso de fabricación de desinfectantes para piso no requiere una respuesta rápida, tal como se plasma en el capítulo 1, ya que el proceso de dosificación y mezcla se realiza de forma manual, por lo que el tiempo de mezclado no influyen en el deterioro o deformación del producto final instantáneamente. Es importante no perder de vista, que parte de la automatización es reducir los períodos de mezclado no sólo con el fin de obtener un producto en el menor tiempo posible, sino con buena calidad, bajo nivel de desperdicio y bajo costo.

No existen tiempos críticos para el proceso, por lo que no es importante discutirlo. El lapso de tiempo promedio realizado por un operario de la industria de desinfectantes para un volumen de 50 galones (1 galón equivale a 3.785 litros) es de una hora y cuarenta minutos, precisando que puede extenderse dependiendo del tiempo que tiene realizando la misma rutina en el día (factores como cansancio), factores ambientales (temperatura), etc. El controlador permitirá realizar la misma tarea para 4 ingredientes (la fórmula de mayor mezcla) con un ahorro de tiempo del 400%, requiriendo un máximo de 12.5 minutos para el supuesto de bombas con un caudal de 6 gpm.

La cantidad del ingrediente vertida, se determinará mediante un medidor de flujo tipo turbina (ver anexos). La velocidad del fluido ejerce una velocidad directamente proporcional al caudal; la diferencia de presiones debida al cambio de área entre el rotor y el cono posterior ejerce una fuerza igual y opuesta. Posee un imán permanente, junto con el campo magnético giratorio que se origina induce una corriente en una bobina captora exterior, produciendo que la frecuencia que se genera en el rotor sea proporcional al caudal. Sus ventajas a mencionar están:

- Amplio rango de medición
- Precisos y de bajo costo
- Construidos de materiales que permiten aplicarse en casi cualquier proceso
- Simples, durables y de fácil reparación

- Operan en amplios rangos de temperatura y presión
- Flexibilidad al acoplarse con controladores electrónicos

Se recalcan las virtudes del medidor de flujo de turbina, en especial la última, debido a que el controlador debe presentar compatibilidad con el sensor con el fin de obtener los resultados deseados, en especial, en el proceso de dosificación, de lo contrario, no tendría sentido la introducción del controlador.

Es necesario profundizar más en el sensor de flujo, debido a que por éste, se determinará el paso o pulso comprenderá el galón, unidad de medida a la cual se sujetará de ahora en adelante los procesos de cálculo y decisiones en el controlador.

Para determinar el tipo de controlador, y en este caso, la frecuencia máxima que deberá contar para toma de decisiones, requeriremos definir

K = Pulsos por galón

Q = Caudal, en galones por minuto (gpm)

f = Pulsos por segundo (Hz)

$$\underline{f = kQ/60}$$

Primero, determinar en cuantas partes dividir el galón, es decir la magnitud de K; su función en el controlador es de determinar la cuenta válida de pulsos que determinan proporcionalmente la unidad de volumen fijada (para este caso el galón). Antes de proseguir es necesario delimitar al sensor, es decir, las características o especificaciones del fabricante: para el sensor al cual el controlador se acoplará es el siguiente: Rango de pulsos = 12 Hz escala baja

225 Hz escala máxima

Rango de flujo = 1.5 gpm escala baja

12 gpm escala máxima

El sensor mencionado, cumple con requisitos de presión máxima, temperatura máxima, precisión, repetibilidad, voltaje de interruptor, etc.; se presenta este apartado con el fin de aclarar que el sensor es un dispositivo al cual el controlador se adaptará, por lo que hace importante desarrollar el cálculo de K, f y Q.

Fijando K constante en tres valores:

- 1) Centésimas de galón (100 partes, 1/100, 0.01)
- 2) Milésimas de galón (1000 partes, 1/1000, 0.001)
- 3) Cien micropartes de galón (10000 partes, 1/10000, 0.0001)

Al graficar las tres condiciones anteriores, entre los límites de especificaciones f y Q del sensor obtenemos los siguientes resultados:

- 1)  $K=100$ , no se obtendrá mediciones correctas si no se aplica para el sistema de tubería el máximo caudal permitido en el sensor de flujo, lo que reduce su tiempo de vida, subutilizado, y pulsos por segundo bajos (20 Hz) lo que hará al sistema junto con el controlador lento para dosificar y mezclar.
- 2)  $K=10000$ , no es una condición práctica, debido a que no se llega a las condiciones o especificaciones de flujo dadas por el fabricante.
- 3)  $K=1000$ , se garantiza además de tener un flujo laminar, larga vida al sensor debido a que no se lleva a los extremos de operación, buen grado de precisión y se

lleva a condiciones en las cuales variaciones de viscosidad no afectarán en gran medida la dosificación correcta.

Un buen controlador, es aquel que se puede adaptar a una gran variedad de sensores, por lo que el controlador debe además de soportar frecuencias de 225 Hz, permitir mayores rangos de frecuencias, rangos de K superiores, etc., con el fin de que no se quede ligado a un sólo tipo de sensor. Es de aclarar que, para el sensor que se tiene previsto adaptar, fue necesario determinar K y f, para las posteriores ultimaciones del controlador. Para el sistema previsto con bomba de 6 gpm y tubería de 1/2" se limita el sistema a:

$$Q= 6 \text{ gpm}$$

$$K= 1000 \text{ pulsos por galón}$$

$$f= 100 \text{ pulsos por segundo (Hz)}$$

Como conclusión, es necesario determinar la velocidad de operación para justificar la utilización de un controlador basado en microcontroladores. El microcontrolador MCS-51 soporta frecuencias hasta 500000 Hz, permitiendo el adapte de una gran variedad de sensores, en el supuesto que se desee en un futuro incrementar el caudal actual de las bombas (6 gpm) a mayores para mejorar la velocidad de mezclado. Cabe recalcar que el sensor se adecua a los rangos puestos por las bombas, lo que determina un buen dimensionamiento de éste.

- Comunicación.

Al fin de obtener información sobre el estado de la máquina o proceso, o de permitir a un operador introducir datos, muchas aplicaciones requieren de comunicación con el operador. Las interfaces tradicionales incluyen botones pulsadores, interruptores preselectores rotatorios, luces pilotos, timbres y pantallas

numéricas con LED. Los sistemas más innovadores presentan teclados alfanuméricos y pantallas de cristal líquido con palabras o dígitos.

La comunicación involucra el compartir datos de la aplicación y su estado con otros dispositivos electrónicos, como computadoras u otro sistema electrónico.

- Interacción con el operador.

La interacción permite que el control se adecue a la receta del producto que se desea procesar, al permitir esta interacción, se garantiza un controlador versátil para introducir parámetros que permita fabricar productos que están en etapa de desarrollo o estudio. El teclado y la pantalla, como en los computadores actuales, es el medio más común por el cual un ordenador (o para el caso el controlador) manda mensajes y status del proceso para que sea el operario el que defina como se prosigue o se detenga.

- Entorno físico.

El controlador se encontrará en un ambiente residencial, debido a que la gran mayoría de la industria química orientada a la producción de desinfectantes, no posee la infraestructura adecuada. Por lo tanto la temperatura será la habitual de la ciudad (aprox. 35° C máxima). A pesar de lo anterior, los dispositivos utilizados soportan temperaturas de hasta 75°, permitiendo un amplio rango para su utilización.

El ambiente industrial de fabricación de desinfectantes es altamente húmedo (refiriéndonos a la presencia o desperdicio de líquidos), es por ello que se recomienda un sistema de ductería y recipientes impermeables con el fin de eliminar desperdicios y salpicaduras accidentales en los operarios; a pesar de la ductería, se requiere proteger el controlador con un chasis con resistencia a salpicaduras de líquidos. Para prevenir cualquier inconveniente con el sistema controlador, las

tarjetas serán recubiertas con una pintura aislante especial, esta pintura brinda aislamiento eléctrico, es a prueba de agua, y protege al cobre contra oxidaciones. La humedad ambiente no es preponderante si no se sobrepasa el 95%.,

Ciertas sustancias utilizadas en el proceso como materia prima, desprenden vapores altamente corrosivos, es por lo tanto necesario la utilización de recipientes de algún plástico altamente resistente a químicos, para evitar reacciones químicas que al combinarse con la materia prima produzca un producto final con características de calidad indeseable. Es por lo tanto justificado el sistema de inyección, ya que en ningún momento se dejan al intemperie alguna sustancia nociva que no sólo afecte al operario, sino corroa o reduzca el tiempo de vida de los sensores, controlador y demás dispositivos.

Parte del manual, incluirá recomendaciones para que el polvo, vibración, entorno del envoltorio de protección, etc., protejan al controlador y al mismo tiempo se asegure larga vida para el mismo.

Los siguientes numerales, en base a lo desarrollado en el capítulo, establecen puntualmente los requisitos primordiales para identificar las funciones del módulo:

1. Diseñar y construir el módulo mínimo fundamental del sistema, el cual incluirá por lo menos los componentes siguientes:

- Un microcontrolador familia MCS-51
- Una memoria ROM de 8k
- Una memoria RAM de 8k
- Un desplegador de cristal líquido alfanumérico
- Teclado para introducción de datos
- Circuitos de soporte

- Interfaces para manejo de entradas/salidas.
2. Diseño de un programa para control del proceso. El programa debe ser capaz de manejar datos desde teclado, enviarlos vía serial si es necesario y almacenarlos en memoria.
  3. El módulo quedará en circuito impreso, para mejor robustez y perdurabilidad. Una guía de usuario permitirá entender su aplicación, instalación y cuidado.

## CAPITULO III

### SISTEMA BASADO EN MICROCONTROLADORES

#### **3.1 Los microcontroladores.**

Hablar de microcontroladores es hablar de manera implícita de microprocesadores, ya que los microcontroladores son dispositivos similares. En 1976 Intel introdujo el 8048, el primer dispositivo de la familia de microcontroladores MCS-48™. Un microcontrolador es un circuito integrado en el cual se han incorporado elementos básicos que conforman un sistema computador; CPU, RAM, ROM (ó EPROM ), ENTRADAS Y SALIDAS, CONTADORES DE TIEMPO, CONVERTIDORES ANÁLOGO A DIGITAL Y VICEVERSA, COMUNICACIÓN SERIAL Y PARALELA.

Los microcontroladores se emplearon inicialmente en sustitución de circuitos muy complejos y extensos, pero de una manera mas específica se emplearon en: sistemas de estado, tales como máquinas de lavado, controladores de luces para tráfico (semáforo), periféricos para computadoras, teclados, controladores en automóviles, etc.

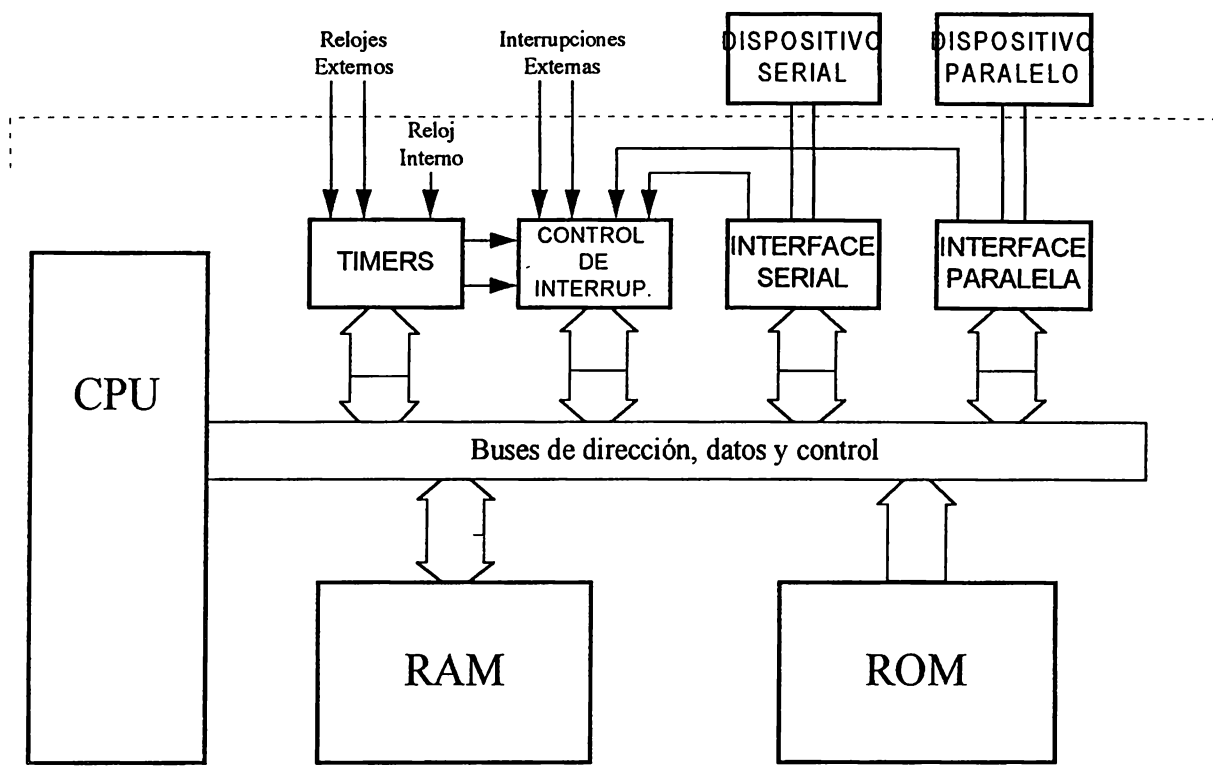
Compañías que desarrollan microprocesadores también desarrollan microcontroladores tal como Intel Corporation, Motorola, RCA, Toshiba, etc. De las cuales las dos primeras son las que en desarrollado una plataforma mucho mas estable que sus competidores, por lo que han logrado obtener una mejor presencia en el mercado.

La potencia y sus características dependen de la familia, las cuales han ido evolucionando de manera similar que los microprocesadores. Por ejemplo la primera familia MCS-48™, contiene más de 17,000 transistores, presentando las siguientes características 1Kbyte de memoria EPROM, 64 bytes de memoria RAM, 27 pines para entradas y salidas, y un timer o temporizador de 8 bits.

El primer dispositivo de la familia de MCS-51™ en contraste con el anterior; contiene mas de 60,000 transistores, 4Kbytes de memoria ROM, 128 bytes de memoria RAM, 32 líneas para entrada y salida, un puerto serial y dos timers.

La diferencia más significativa entre un microprocesador y microcontrolador es: que un microprocesador es solamente un CPU (Unidad, de Proceso Central), mientras que un microcontrolador es un sistema compuesto además de un CPU, de RAM, ROM, Timers, Interface Serial y Paralela, etc., como el que se presenta en la figura 3.1. Además la circuitería presenta mayor robustez que el simple CPU con microprocesador.

La arquitectura del microcontrolador con respecto al microprocesador, es fácil pensar que la elaboración de microprocesadores es subutilizada, sin embargo la característica más significativa en la elaboración de los microprocesadores es el manejo de una cantidad mayor de información que la que maneja un microcontrolador, debido a esta particularidad es que los microprocesadores son utilizados para ordenadores personales y los microcontroladores para sistemas en la industria.



*Figura 3.1. Diagrama en bloques de un microcontrolador de Intel*

## EL CPU

La unidad de procesamiento central, es el cerebro del sistema, administra toda la actividad en el sistema y realiza todas las operaciones en los datos.

Las dos actividades principales del CPU son:

- Realizar la extracción de las instrucciones de la memoria (a este proceso se le llama "Fetch" ).
- Ejecutar la instrucciones que fueron extraídas de la memoria.

En una forma simplificada el CPU, esta compuesto por:

- Los registros: que son un conjunto temporal de almacenamiento, en donde son colocados los datos a procesar.

- La unidad aritmética y lógica, ( *ALU: Arithmetic and logic unit* ): La cual se encarga de realizar las operaciones aritméticas ( suma, resta, etc. ) y lógicas ( and, or, not, etc. ) de los datos contenidos entre los registros.
- El Decodificador de instrucción y unidad de control: El cual determina la operación que realizará la ALU, así como la cantidad de pasos que se deben realizar para llevar a cabo tal instrucción.
- Registro de Instrucciones ( *IR: Instruction Register* ): almacena el código binario para la instrucción que es ejecutada.
- Contador de Programa ( *PC: Program Counter* ): Guarda la dirección de memoria de la siguiente instrucción que será ejecutada.

## **BUS DE DIRECCIÓN, DATOS Y CONTROL**

Un bus es un grupo de líneas que transportan información con un propósito común; para que el CPU pueda acceder cualquier elemento periférico a él; existen tres buses: uno para las **direcciones** de las localidades de memoria, otro para transportar los **datos** o información, y un tercero el de **control** para seleccionar los elementos con los cuales se establecerá la comunicación.

**Bus de control:** Para cada operación de escritura o lectura el CPU especifica la localidad de los datos, coloca una dirección en el bus de direcciones, activa una señal en el bus de control indicando si la instrucción es una operación de lectura o escritura. Si la operación es de lectura por ejemplo, el CPU lee el dato que se encuentra en el bus de datos y lo almacena en un registro interno, de forma análoga ocurre si la operación es de escritura.

**Bus de dirección:** La mayoría de computadoras pequeñas tienen 16 o 20 líneas en el bus de direcciones, en cada línea es posible encontrar uno de dos valores lógicos ya sea alto o bajo. Si definimos  $n$  como la cantidad de líneas,

entonces la cantidad de localidades que pueden ser accesadas son  $2^n$  por tanto podemos acceder  $2^{16} = 65,536$  localidades.

**Bus de datos:** El bus de datos transporta la información entre el CPU y la memoria o entre el CPU y los dispositivos de entrada y salida (I/O). Para una cantidad de líneas en el bus de datos de 20, se pueden acceder palabras de longitud de 20 bits. Cuando se menciona microcontroladores de 8 bits, se refiere a que el bus de datos es de 8 bits.

Los sistemas computadores utilizan dos o tres tiempos simples (pulsos de reloj), para mover los datos entre un registro interno del CPU, a una localidad externa en la ROM o la RAM, el número de líneas (ancho) del bus de datos determina el largo de las palabras, además este ancho es muy importante porque se comporta como un cuello de botella para la ejecución de ciertas instrucciones. Cuando se dice que un computador es de 16 bits, se está diciendo que el bus de datos es de 16 líneas, de igual forma existen computadores de 4, 8, 16 o 32 bits. El bus de datos es bidireccional; es decir que, los datos que transportan pueden ser leídos o escritos por los elementos que están conectados al bus. El bus de direcciones es unidireccional.

## PUERTOS DE I / O

Los dispositivos de entrada y salida, son llamados "*periféricos del computador*", proveen la vía de comunicación entre el sistema computador y el "*mundo real*". Sin estos elementos el sistema computador quedaría aislado y su función con los usuarios resultaría demasiado limitada. Existen tres tipos de dispositivos I/O:

- Dispositivos para almacenamiento: Tales como RAMs y ROMs, en donde se almacenan grandes cantidades de información (programas o datos), estos

dispositivos son de gran importancia, ya que la memoria principal (memoria interna) es muy pequeña comparada con el valor nominal de estas memorias.

De acuerdo a la facilidad de acceso, los dispositivos de almacenamiento pueden clasificarse en: Online o Archival:

Online: cuando el CPU obtiene la información requerida sin la intervención humana. Tales como EPROMS, o información en discos magnéticos.

Archival: En este tipo de dispositivos de almacenamiento, los datos son raramente requeridos por el CPU, y para ser cargada la información se necesita la intervención humana. Ejemplos de este tipo de almacenamiento son: los discos ópticos (o CD's), discos flexibles.

- Dispositivos para la interface humana: la unión del hombre y la máquina los dispositivos más comunes son los teclados, impresores, terminales de vídeo. Por medio de los cuales el hombre introduce y recibe la información.
- Dispositivos de Control / Monitor: Son los dispositivos por medio de los cuales el computador tiene la habilidad de realizar tareas de supervisión de elementos externos al mismo, para realizar esta tarea en particular es necesario la intervención de sensores que puedan registrar las variaciones de las propiedades que se desean monitorear.

### **3.2 La familia MCS-51**

#### **3.2.1 DESCRIPCIÓN DE LA FAMILIA MCS-51™**

La familia de microcontroladores MCS-51™, es fabricada y distribuida por Intel Corporation. Otras compañías tales como SIEMENS, ADVANCED MICRO DEVICES, FUJITSU y PHILLIPS tienen licencia de segunda fuente, es decir que estas compañías fabrican los integrados sin desarrollar avances en sus características. El

IC genérico de la familia MCS-51™ es el 8051, el cual fue el primer dispositivo que se ofreció comercialmente, sus características son presentadas a continuación:

- 4K bytes de ROM (enmascarado de fábrica)
- 128 bytes de RAM
- 4 puertos de I/O de 8 bits
- 2 Timers de 16 bits
- Interfaces Serial
- 64K ROM externo
- 64K RAM externa
- Procesador Booleano (operación a bit)
- 120 localidades de RAM direccionables a bit
- Multiplicación y división de 4 μseg (a 12 MHz)

A continuación se presenta una tabla comparativa entre los miembros más comunes de la familia MCS-51™.

ELEMENTO	MEMORIA PROG.	MEMORIA DATOS	TIMERS
8051	4K ROM	128 bytes	2
8031	0K	128 bytes	2
8751	4K EPROM	128 bytes	2
8052	8K ROM	256 bytes	3
8032	0K	256 bytes	3
8752	8K EPROM	256 bytes	3

### 3.2.2 DESCRIPCIÓN DE LOS PINES DEL 8051

El microcontrolador es un circuito integrado de 40 pines, 24 de estas líneas son pines de doble propósito (26 en el 8032/8052 ). Cada uno de estos pines puede operar como pin de I/O; como línea de control; o como bus de datos o direcciones. En la figura 3.2 se muestra el diagrama de bloques de la arquitectura del microcontrolador 8051.

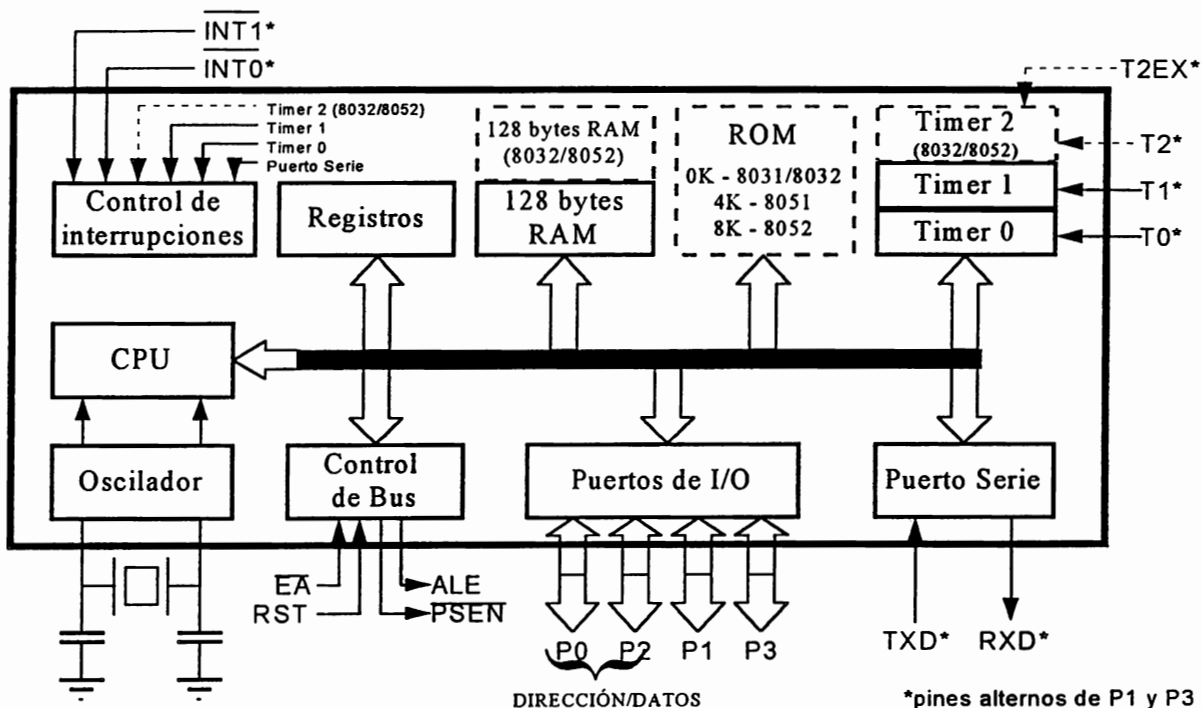


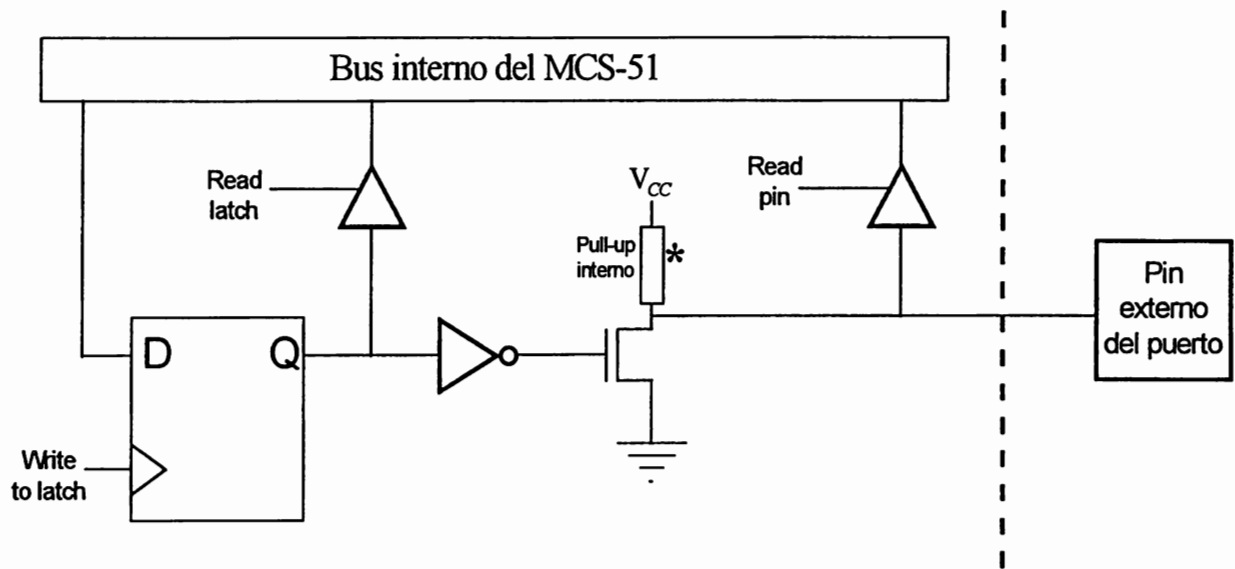
Figura 3.2 Arquitectura del 8051

#### PUERTO 0

El puerto 0 es un puerto de doble propósito, y se compone de los pines 32 al 39; en diseños de sistemas con componentes mínimos, este puerto es utilizado como un puerto de entrada y salida (I/O). Pero en diseños de sistemas más complejos con memoria externa, viene a ser un bus multiplexado para direcciones o datos.

## PUERTO 1

El puerto 1 esta dedicado como un puerto de entrada y salida, se encuentra en los pines 1 al 8, los cuales son llamados P1.0, P1.1, P1.2, y así sucesivamente. Estos pines están habilitados para conectarse con dispositivos externos, cada pin de este puerto es independiente, lo que permite que pueda conectarse a motores, led, una alarma, etc. Para el caso del 8032 y el 8052, el pin P1.0 y P1.1 pueden ser utilizados para el tercer timer.



\* Salida de colector abierto para puerto 0 cuando se usa como un puerto de I/O

Figura 3.3. Estructura Interna de puertos

## PUERTO 2

El puerto 2 se encuentra entre los pines 21 al 28, y es otro puerto de doble propósito, que se utiliza como puerto de propósito general (I/O) o como la parte alta del bus de direcciones, cuando el sistema posee memoria externa.

### PUERTO 3

Es un puerto de doble propósito, se encuentra entre los pines 10 al 17, y puede utilizarse como puerto de entrada y salida, o como un puerto con funciones especiales, algunas de estas funciones son: las interrupciones externas, señal de lectura y escritura, pin para los timer y pines para comunicación serial.

BIT	NOMBRE	FUNCIÓN ALTERNATIVA
P3.0	RXD	Recepción de datos serialmente
P3.1	TXD	Transmisión de datos serialmente
P3.2	'INT0	Interrupción externa 0
P3.3	'INT1	Interrupción externa 1
P3.4	T0	Timer/Contador 0
P3.5	T1	Timer/Contador 1
P3.6	'WR	Pin de escritura externa de memoria
P3.7	'RD	Pin de lectura externa de memoria
P1.0	T2	Timer/Contador 2 (8032/8052/8752)
P1.1	T2EX	Captura/Recarga timer/contador 2

### LÍNEAS DE CONTROL

#### PSEN ( PROGRAM STORE ENABLE )

Este pin (29), es uno de los cuatro pines que conforman señales de control, se encarga de indicarle al microcontrolador donde ejecutar al programa, para el caso si su valor es un cero lógico, el microcontrolador busca la instrucciones en la memoria externa, mientras que si se aplica un uno lógico, el microcontrolador ejecutará el programa en la ROM interna (para 8051 y 8052). Concretamente, es la señal de lectura del ROM externo, se activa cada vez que se realiza un Fetch de instrucción al ROM.

## **ALE ( ADDRESS LATCH ENABLE )**

La señal de ALE (30) puede ser familiar para aquellos quienes trabajaron con los microprocesadores Intel 8085, 8086 y 8088. Sirve en la demultiplexación del bus de direcciones y datos, cuando se opera de modo de programa externo. ALE es una señal que enclava la parte baja de la dirección en un dispositivo externo durante la primera mitad del ciclo de máquina. Una vez realizado, el Puerto 0 está en la capacidad de realizar funciones de Entrada/Salida de datos en la segunda mitad del ciclo de máquina.

## **EA ( EXTERNAL ACCESS)**

Nos referimos al pin 31, indica al microcontrolador donde debe ejecutar el programa, es decir, desde un dispositivo externo de memoria o el interno. Si es atada a una señal lógica igual a cero, el microcontrolador el programa a ejecutar reside en la ROM externo, de lo contrario (señal lógica igual a 1) ejecutará lo residente en la ROM interna.

## **RESET**

Ubicado en el pin 9, es el reset maestro del microcontrolador. Requiere de por lo menos dos ciclos de máquina en alto para que coloque los siguientes registros a los siguientes valores:

<b>REGISTROS</b>	<b>CONTENIDO</b>
Program Counter	0000h
Acumulador	00h
Registro B	00h
PSW	00h
Stack Pointer	07h
DPTR	0000h
Puertos: 0 al 3	FFh
Timer registros	00h

SCON	00h
SBUF	00h
IP (8031/51)	XXX00000b
IP (8032/52)	XX000000b
IE (8031/51)	0XX00000b
IE (8032/52)	0X000000b

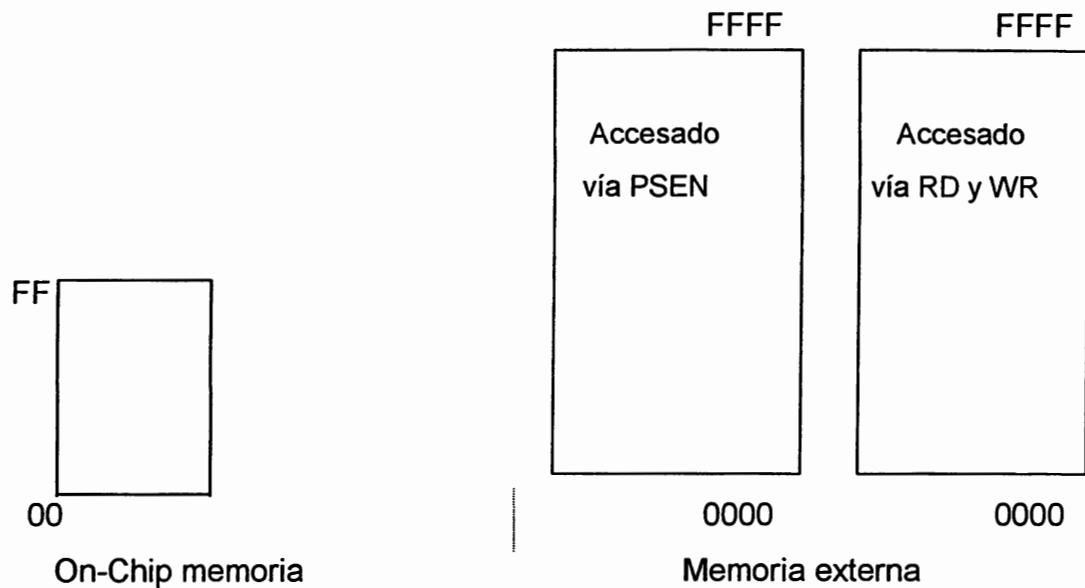
## OSCILADOR

La familia MCS-51 tiene una sistema interno que permite conectar un cristal directamente a sus pines (18 y 19), sin necesidad de otros elementos. Su frecuencia nominal es de 12 MHz, requiriendo nada más que unos condensadores de estabilización. Existen familias mejoradas que permiten osciladores hasta de 45 MHz.

### 3.2.3 ORGANIZACIÓN DE MEMORIA

La mayoría de microprocesadores implementan la arquitectura de memoria Von Newman, un espacio de memoria compartida para datos y programas. Esto es razonable, debido a que los programas son almacenados en discos y cargados en la RAM para su ejecución; tanto los datos como el programa residen en la RAM del sistema. Los microcontroladores, por el otro lado, son raramente usados como el CPU en las computadoras comerciales. Los microcontroladores son empleados como el componente central en los diseños de control orientado. Existe memoria limitada, y no existe un sistema operativo, el control reside en la ROM.

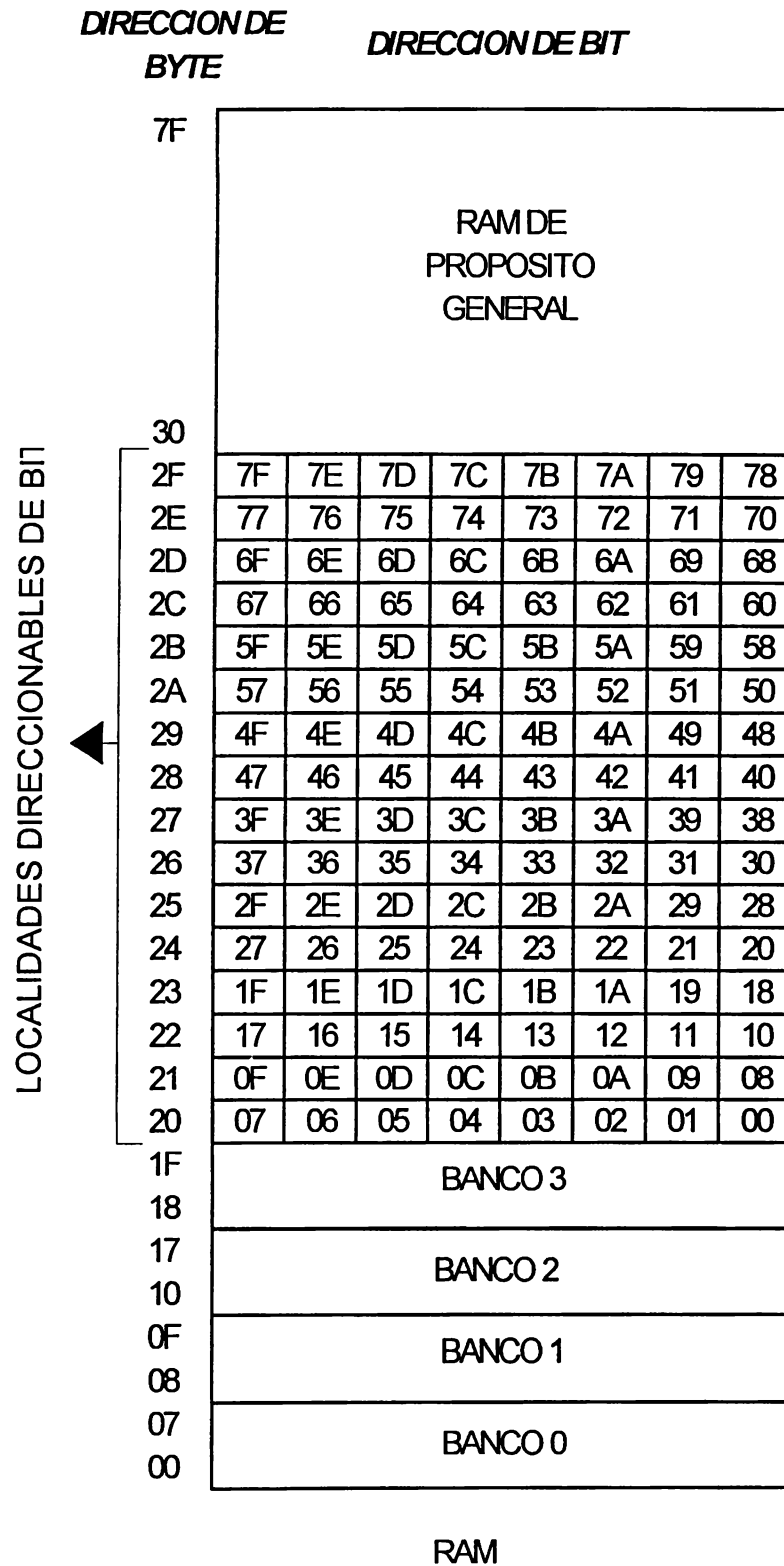
Por esta razón, el microcontrolador de la familia MCS-51 implementa una arquitectura Harvard para su memoria: Un espacio separado de memoria para programas y datos. Como se muestra en la figura 3.4, puede ser en su memoria interna como expandida a componentes externos.



*Figura 3.4. Sumario de los espacios de memoria del 8051*

- RAM DE PROPÓSITO GENERAL

La familia MCS-51 posee internamente 128 bytes de RAM, excepto las versiones 8052, 8032 y 8752 los cuales poseen 128 bytes extra. La dirección 30h hasta la 7Fh pertenecen a la parte de uso general, denotando desde la dirección 20h hasta 2Fh pertenecen a uso general con la ventaja que se pueden acceder a bit. Conforme se desarrolle el sistema mínimo, se pretende profundizar más en cada uno de estos elementos. Para tener una mejor idea, podemos observar la figura 3.5, donde tenemos una panorámica mejor de todos los detalles presentes en la memoria RAM interna (On-Chip data memory). Es importante recalcar, que cada registro especial (Acumulador, puertos, etc.), pueden ser accesados ya sean con su dirección respectiva mapeada en memoria como por su nombre genérico, dependiendo del ensamblador utilizado. El Program Counter y el Instruction Register no se encuentran mapeados como los anteriores, por lo que no son accesables.



<b>DIRECCION DE BYTE</b>	<b>DIRECCION DE BIT</b>								
FF									
F0	F7	F6	F5	F4	F3F	F2	F1	F0	B
E0	E7	E6	E5	E4	E3	E2	E1	E0	ACC
D0	D7	D6	D5	D4	D3	D2	-	D0	PSW
B8	-	-	-	BC	BB	BA	B9	B8	IP
B0	B7	B6	B5	B4	B3	B2	B1	B0	P3
A8	AF	-	-	AC	AB	AA	A9	A8	IE
A0	A7	A6	A5	A4	A3	A2	A1	A0	P2
99	BIT NO DIRECCIONABLE								SBUF
98	9F	9E	9D	9C	9B	9A	99	98	SCON
90	97	96	95	94	93	92	91	90	P1
8D	BIT NO DIRECCIONABLE								TH1
8C	BIT NO DIRECCIONABLE								TH0
8B	BIT NO DIRECCIONABLE								TL1
8A	BIT NO DIRECCIONABLE								TL0
89	BIT NO DIRECCIONABLE								TMOD
88	8F	8E	8D	8C	8B	8A	89	88	TCON
87	BIT NO DIRECCIONABLE								
83	BIT NO DIRECCIONABLE								DPH
82	BIT NO DIRECCIONABLE								DPL
81	BIT NO DIRECCIONABLE								SP
80	87	86	85	84	83	82	81	80	P0

## REGISTROS DE FUNCIONES ESPECIALES

Figura 3.5. Mapeo de memoria interna del microcontrolador

- **BANCO DE REGISTROS**

Al observar el mapeo de memoria interna del microcontrolador (On-Chip data memory) existen 32 locaciones conocidas como "Banco de registros". El set de instrucciones del microcontrolador soporta 8 registros, los cuales van desde R0 hasta R7, y por defecto (después de un reset del sistema), dichos registros están localizados entre las direcciones 00h-07h de la memoria interna. La ventaja de el uso de dichos registros, radica en lo corto y rápido que pueden efectuar operaciones comparado al usar instrucciones de direccionamiento directo.

### 3.2.4 REGISTROS ESPECIALES

Cada uno de los siguientes registros contiene su dirección en la memoria interna, su función como direccionamiento se explican a continuación:

- **ACUMULADOR**

Dirección en mapeo F0h, direccionable a bit, de propósito general, se comporta de la misma forma que los acumuladores de la mayoría de microprocesadores de 8 bits del mercado. Se ampliará más en el capítulo relacionado a modos de direccionamiento.

- **PROGRAM STATUS WORD**

Program Status Word o PSW (Estatus de palabras del programa), en la dirección D0h. Su función es informar en la ejecución de un programa, la situación lógica-aritmética respecto al acumulador. También permite la selección de registros.

<b>BIT</b>	<b>SÍMBOLO</b>	<b>DIRECCIÓN</b>	<b>DESCRIPCIÓN DEL BIT</b>
PSW.7	CY	D7h	Bandera de acarreo (Carry flag)
PSW.6	AC	D6h	Bandera de acarreo auxiliar (Auxiliary carry flag)
PSW.5	F0	D5h	Bandera de 0

PSW.4	RS1	D4h	Selección del Banco de Registro (1)
PSW.3	RS0	D3h	Selección del Banco de Registro (0) 00 = Banco 0; Direcciones 00-07h 01 = Banco 1; Direcciones 08-0Fh 10 = Banco 2; Direcciones 10-17h 11 = Banco 3; Direcciones 18-1Fh
PSW.2	OV	D2h	Bandera de Overflow o desborde
PSW.1		D1h	Reservado
PSW.0	P	D0h	Bandera de paridad

**Bandera de acarreo (CY):** Utilizada en operaciones aritméticas, se activa si el acarreo sobrepasa el bit 7 durante las operaciones. Además, se utiliza como el acumulador booleano para operaciones a bit.

**Banderas de acarreo auxiliar (AC):** Se utiliza en operaciones BCD, ésta es activada si hay desborde del bit 3 al bit 4 o si el resultado del nibble inferior está en el rango de 0A-0Fh.

**Bandera 0 (F0):** Bandera de 0 para el usuario.

**Selección del banco de registro (RS1-RS0):** Determinan el banco de registro activo, por default se ubica 00h respectivamente en estos bits.

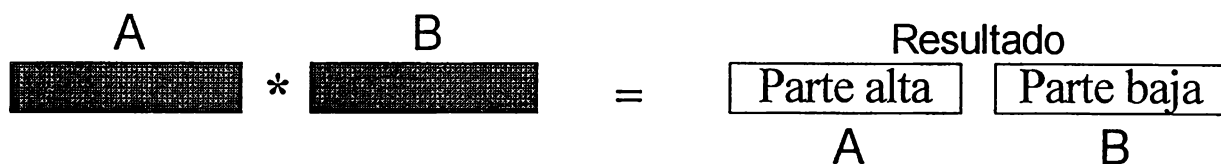
**Bandera de desborde (OV):** Es activada después de una adición o substracción si hubo un desborde aritmético, es decir, si el resultado es menor que -128d y mayor que 127d.

**Bit de paridad (P):** Activado o desactivado automáticamente cada ciclo de máquina para establecer la paridad del dato en el acumulador.

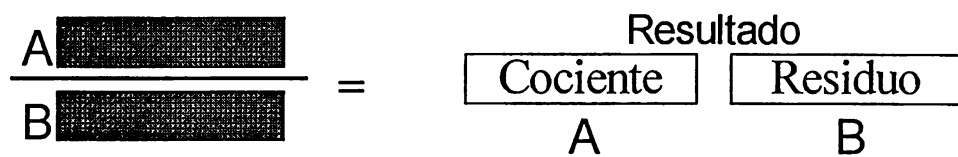
- REGISTRO B

El registro B en la dirección F0h se usa junto con el acumulador para operaciones de multiplicación y división. La operación se realiza como se observa en el siguiente gráfico (figura 3.6). Cabe recalcar que puede ser tratado o direccionado a bit.

Instrucción: **MUL AB**



Instrucción: **DIV AB**



*Figura 3.6. Uso del registro B*

- STACK POINTER

El registro de pila, stack pointer o SP es un registro de 8 bits localizado en la dirección 81h. Contiene la dirección del elemento de dato al tope de la pila. Es una memoria tipo LIFO (Last In First Out). Después de un Reset se direcciona la localidad 07h. Es conveniente reinicializar el SP para que inicie en los primeros bytes ubicados en la RAM de propósito general.

Al empujar datos en la pila se incrementa el puntero, mientras que al extraer datos, se decrementa la pila.

A=21H  
 B=45H  
 R0=00H

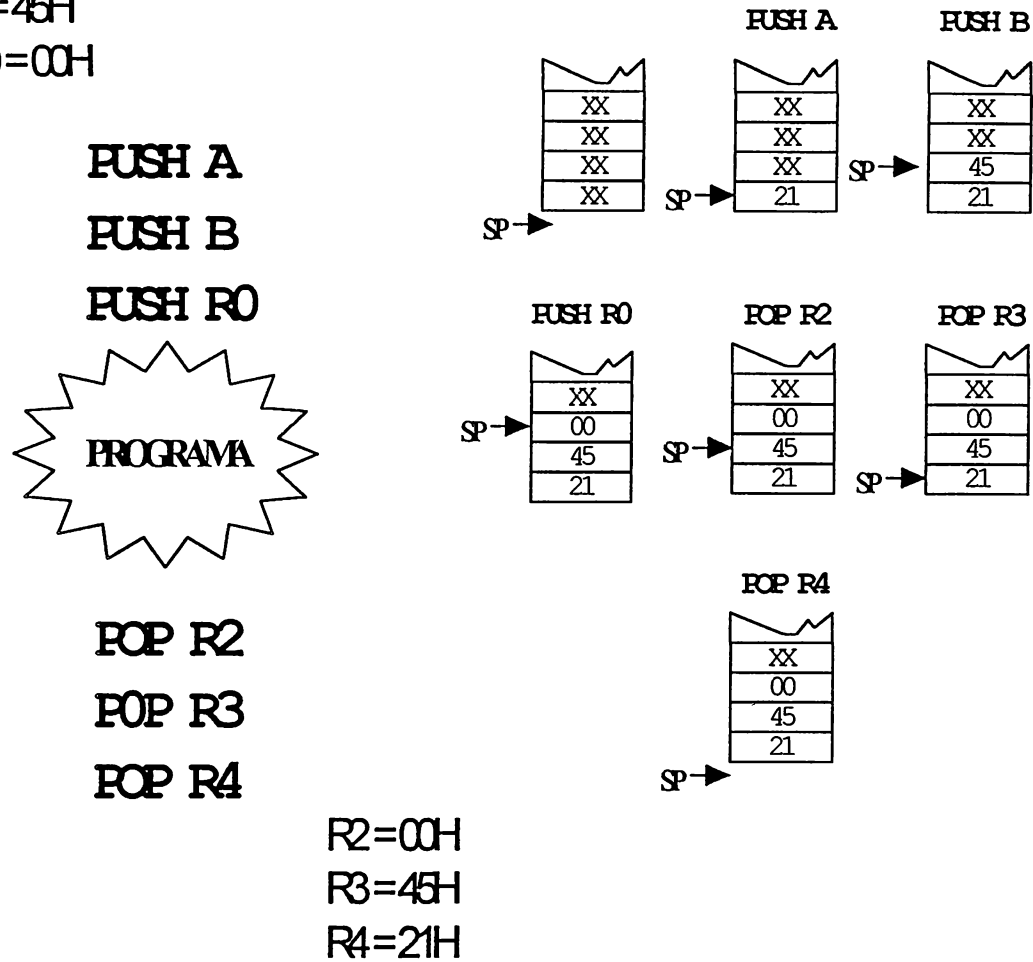


Figura 3.7. Uso de puntero de pila

- DATA POINTER

Puntero de datos, data pointer o DPTR es usado para acceso de datos de la memoria externa, ya que es un registro de 16 bits ubicado en las direcciones 82h (parte baja) y 83h (parte alta).

- **REGISTROS DE PUERTO**

MCS-51 contiene 4 puertos, Puerto 0 en la dirección 80h, Puerto 1 en la dirección 90h, Puerto 2 en la dirección a0h y el Puerto 3 en la dirección B0h. Si se usa memoria externa (RAM o ROM) no podrán estar disponibles los puertos 0, 2 y 3. Estos pueden ser accedidos por bit o byte.

- **REGISTROS DE TEMPORIZADORES**

El 8051, 8031 y 8751 poseen dos temporizadores/contadores de 16 bits usado para intervalos temporales o cuenta de eventos. El temporizador 0 está en la dirección 8Ah (parte baja o TL0) y 8Ch (parte alta o TH0). El temporizador 1 está en la dirección 8Bh (parte baja o TL1) y 8Dh (parte alta o TH1). El modo de operación es fijado por el registro TMOD ubicado en la dirección 89h y el registro de control TCON en la dirección 88h. La versión 8XX2 es la única que incluye 3 temporizadores/contadores.

- **REGISTROS DE PUERTO SERIAL**

La familia MCS-51 contiene un puerto serial para comunicación con terminales, MODEMS o interfaces para otros periféricos seriales como convertidores A/D, registros de desplazamiento, RAM no volátiles, etc. El registro SBUF ubicado en la dirección 99h, contiene tanto el dato a transmitir como el recibido. Su registro de control se conoce como SCON ubicado en la dirección 98h.

- **REGISTROS DE INTERRUPCIONES**

La familia MCS-51 posee cinco fuentes de interrupciones, dos niveles de prioridad de interrupción, estas son deshabilitadas después de un reset, habilitadas cuando es fijado el registro de habilitación IE (Interrupt Enable) en la dirección A8h. Los niveles de prioridad de interrupción son colocados en el registro IP en la dirección B8h.

- **REGISTRO DE CONTROL DE POTENCIA**

PCON, Power Control o registro de control de potencia, está ubicado en la dirección 87h, contiene una miscelánea de bits de control, especialmente los dos primeros bits de la parte baja (bit 0 y bit 1). Esta opción solo es disponible en los modelos CMOS. Este registro no es accesado a bit. Para contenido del registro observar figura 3.8.

- **Modo "Idle":**

- El reloj interno se desconecta del CPU
- Se mantienen las funciones de interrupción, timer y puerto serial.
- El estatus del CPU y los contenidos de los registros se conservan. Los puertos mantienen sus niveles lógicos.
- ALE y /PSEN se mantienen en nivel alto

El modo es terminado al activar una interrupción o con un reset del sistema; en ambos casos se coloca un cero lógico en el bit IDL.

- **Modo "Power Down":**

- El oscilador del chip se detiene
- Todas las funciones se detienen
- Los contenidos de la RAM interna se mantienen
- Los pines de los puertos mantienen sus niveles lógicos
- ALE y /PSEN se mantienen en nivel bajo
- La única salida es un reset del sistema

Durante este modo el valor de  $V_{CC}$  puede ser tan bajo como 2 V.

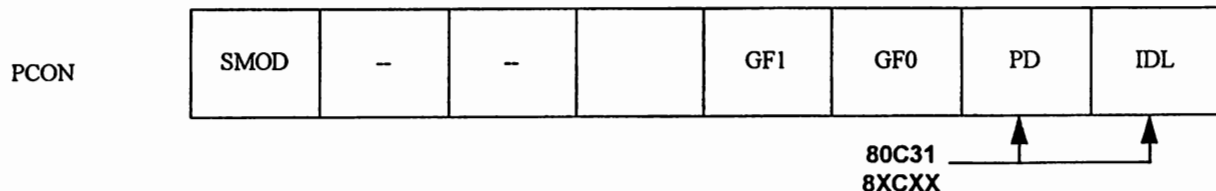


Figura 3.8. Registro PCON

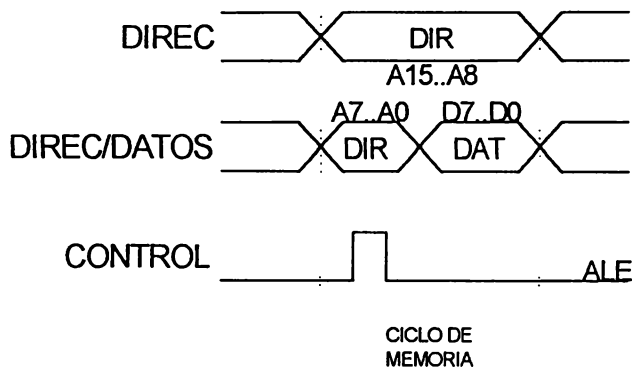
### 3.2.5 MEMORIA EXTERNA

Es importante que el microcontrolador tenga capacidades de expansión superiores a los recursos internos del chip (On-Chip memory). RAM y ROM externa pueden ser anexadas al sistema si es necesario, para ello es necesario un mapeo de memoria.

Cuando se utiliza memoria externa, el Puerto 0 no está disponible como puerto de Entrada/Salida. Este se convierte en el bus multiplexado de dirección (AB0-AB7) y de datos (DB0-DB7) combinados con la señal ALE para enclavar la parte baja de la dirección al inicio de cada ciclo de memoria externa. El Puerto 2 es utilizado (no siempre) para la parte alta del bus de dirección.

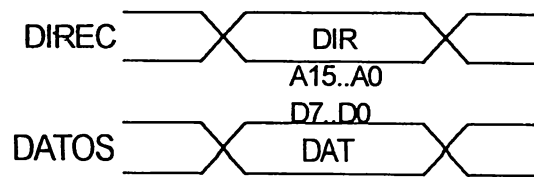
Antes de continuar discutiendo el proceso de multiplexación de dirección y datos realizados por el microcontrolador, es necesario aclarar el punto, observando la figura 3.9. Un arreglo no multiplexado como se muestra en la figura 3.9b, usa 16 líneas dedicadas para dirección y 8 líneas dedicadas para datos, para un total de 24 pines. El arreglo multiplexado (ver figura 3.9a) combina 8 líneas dedicadas par el bus de datos y la parte baja del bus de dirección, con otras 8 líneas dedicadas a la parte alta del bus de dirección, para un total de 16 pines. El ahorro en pines permite ofrecer en un DIP 40 (Dual inline package) otras funciones haciendo mejor al controlador.

Para la multiplexación, la primera mitad de cada ciclo de memoria, la parte baja de la dirección es dada por el Puerto 0 y es enclavada usando la señal ALE. Un dispositivo latch sostiene la parte baja de la dirección durante el resto del ciclo de memoria. Durante la segunda mitad del ciclo, Puerto 0 es usado como bus de datos, y los datos se escriben o leen según la operación.



*Sistema multiplexado*

3.9a



*Sistema sin multiplexar*

3.9b

3.9. Multiplexación para memoria externa

- ACCESANDO LA MEMORIA EXTERNA DE CÓDIGO

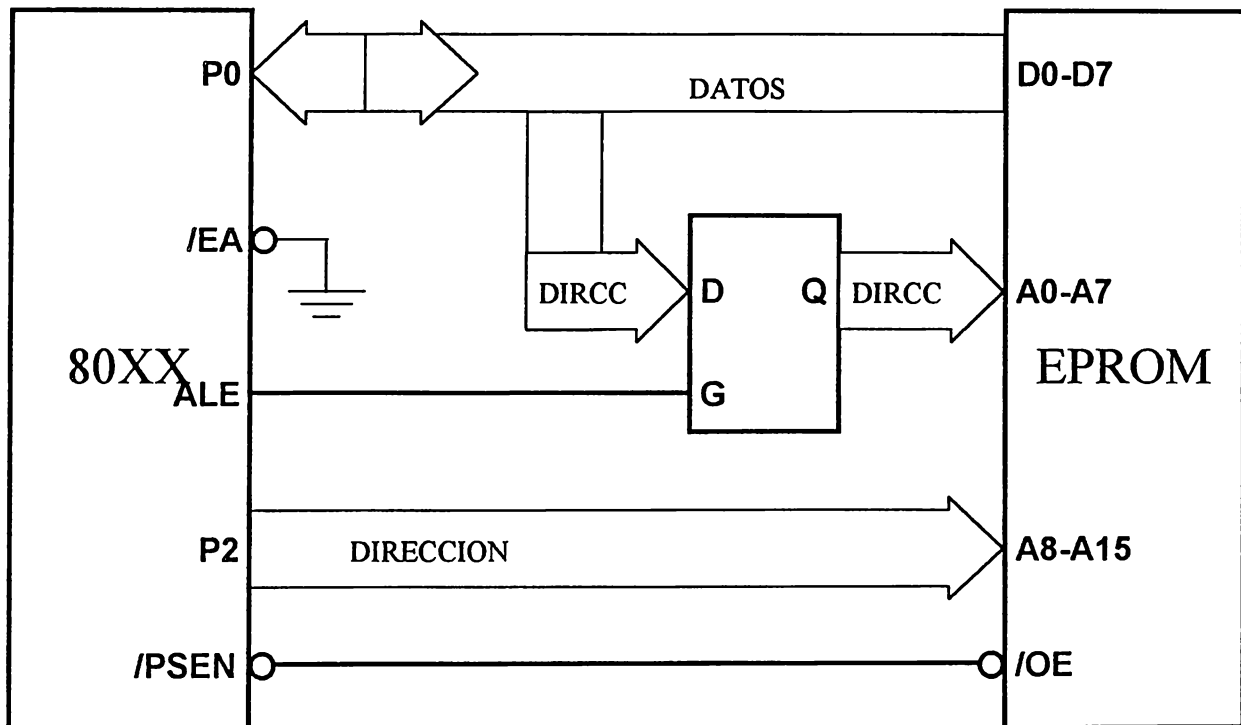


Figura 3.10. Accesando Memoria Externa de Código

La memoria externa de código es de tipo ROM o solo lectura, habilitada por la señal  $\overline{\text{PSEN}}$  (Figura 3.10). En este caso tanto el Puerto 0 y 2 no están disponibles como puertos de Entrada/salida de propósito general. Si el oscilador es de 12 MHz, un ciclo de máquina toma 1  $\mu\text{s}$  de duración. ALE es una señal que se dispara dos veces en el ciclo de memoria, esto significa la lectura de dos bytes desde la memoria del programa, pero si la corriente instrucción es de un byte, el segundo byte es descartado (fig. 3.11).

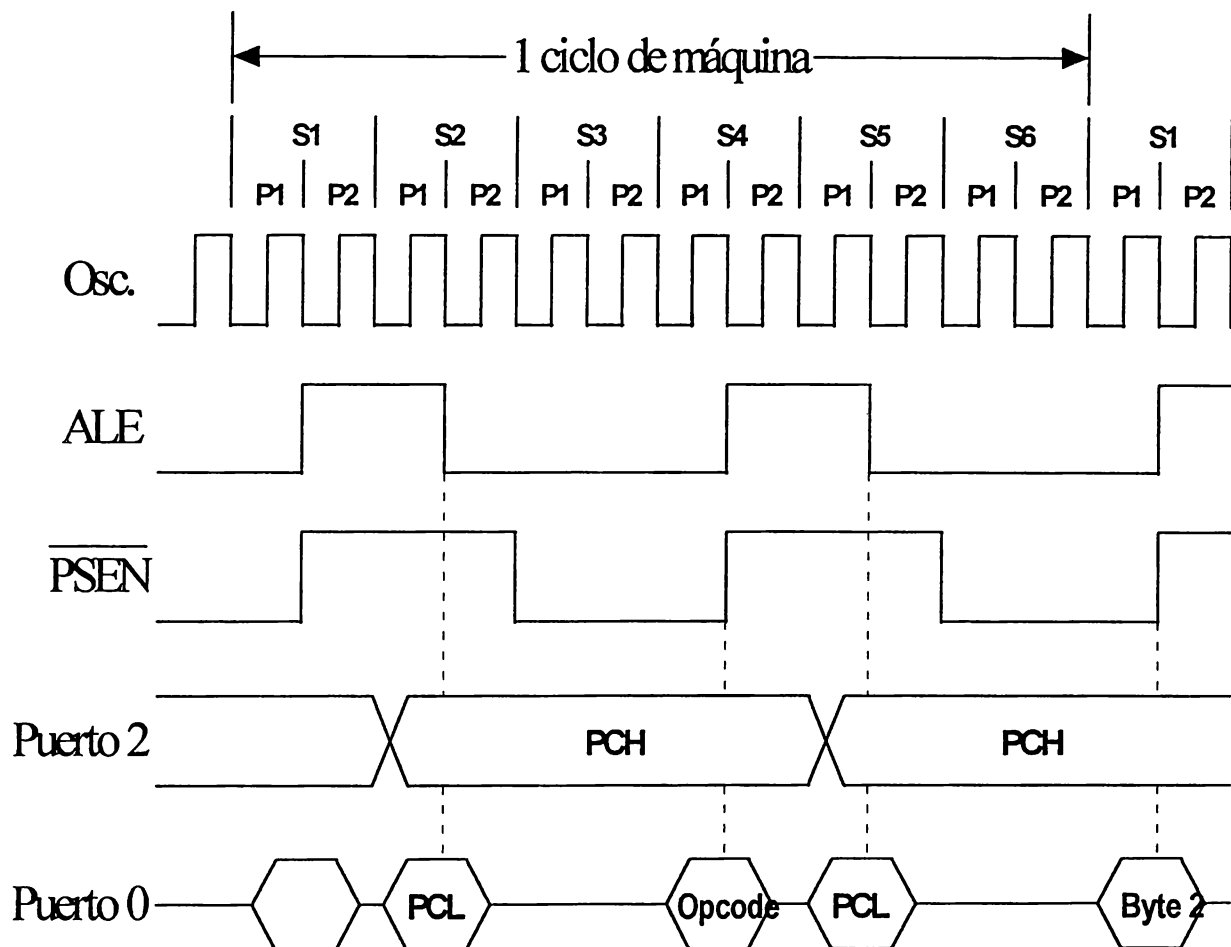


Figura 3.11. Tiempo de lectura de la memoria de código externa

- ACCESANDO DATOS DE LA MEMORIA EXTERNA

En este caso, se tiene una memoria de acceso aleatorio o RAM, por lo tanto se utilizan los pines  $\overline{RD}$  y  $\overline{WR}$  (P3.7 y P3.6 respectivamente). El único acceso a la memoria externa es mediante la instrucción MOVX, usando también el puntero de datos (DPTR), R0 o R1 como registro de dirección.

En la figura 3.12 se presenta la ejecución de la instrucción MOVX.

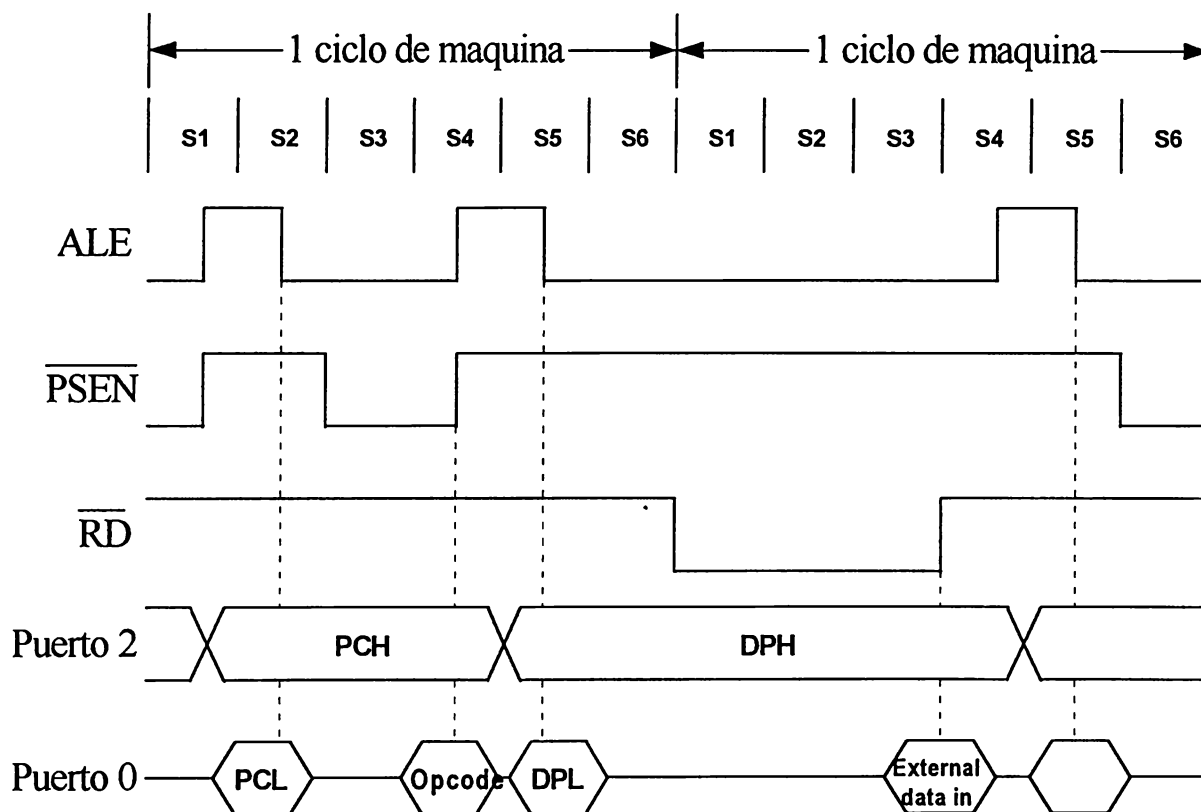


Figura 3.12. Instrucción MOVX

En la figura 3.13 se muestra la conexión con una RAM externa. Se nota que tiene conexión parecida a la utilizada con la ROM externa, excepto por la línea RD conectada al pin output enable ( $\overline{OE}$ ) de la RAM y WR al pin de la RAM write ( $\overline{W}$ ).

En la figura 3.13B se muestra la opción de conectar múltiples RAMs o ROMs a un mismo microcontrolador, para ello se utiliza la decodificación de dirección. Es similar a la utilizada por la mayoría de microprocesadores. Para la decodificación de bloques de 8K, se deben de fijar límites de 8K (0000-1FFFh, 2000-3FFFh, etc.), requiriendo un decodificador de 8 líneas de salida. Recordemos que pueden ser acomodados un máximo de 64K para la familia MCS-51.

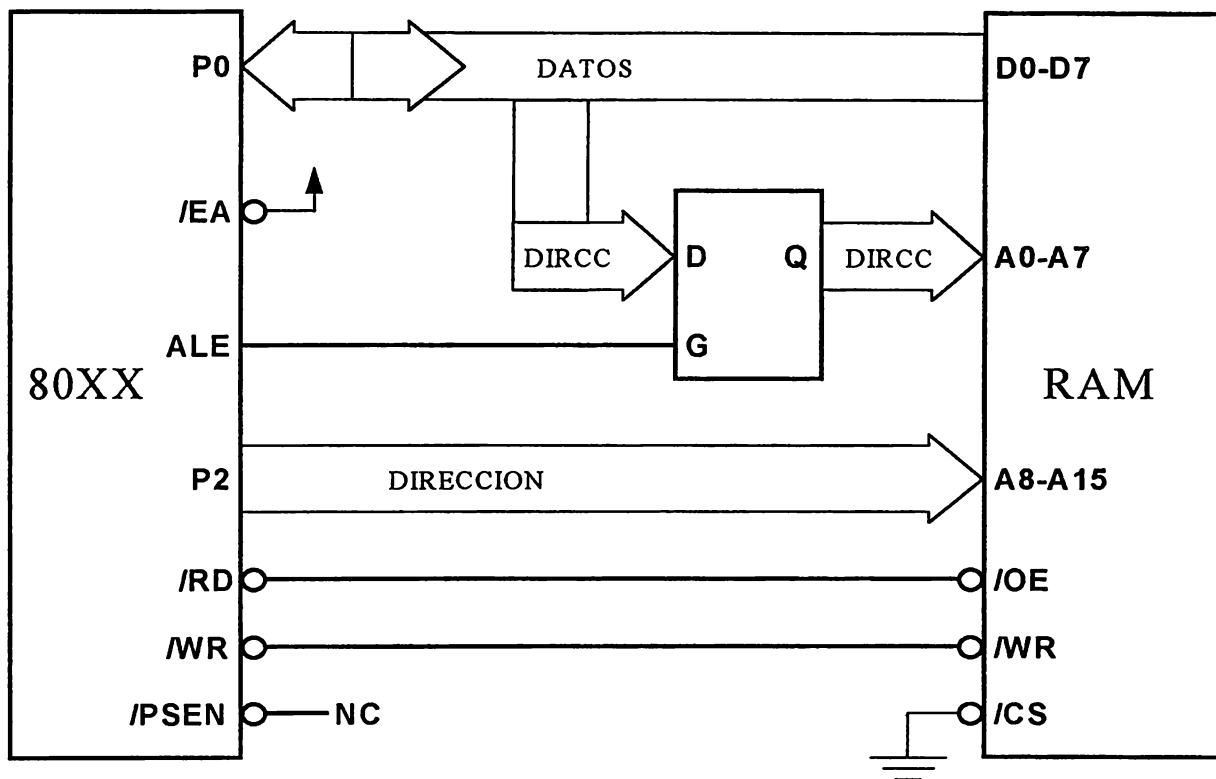


Figura 3.13. Conexión de RAM externa

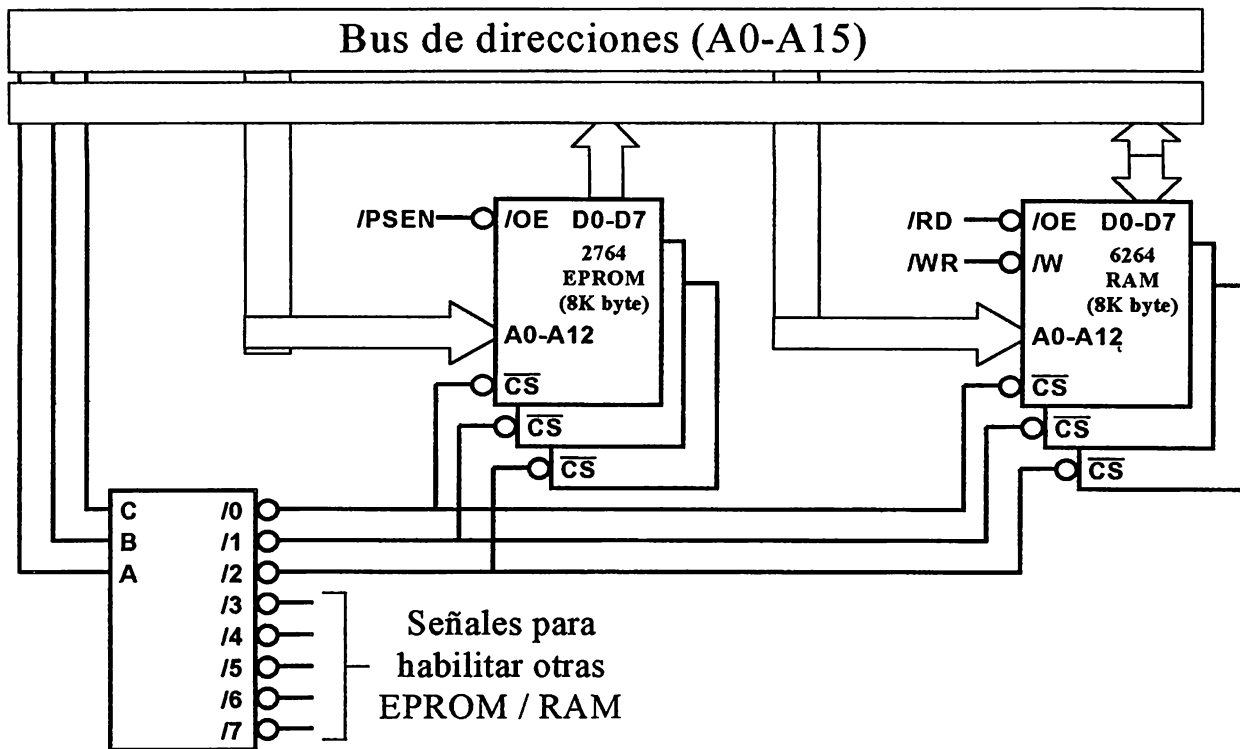


Figura 3.13B. Decodificación de dirección

### 3.3 TEMPORIZADORES (TIMERS) Y CONTADORES

Un timer es una serie de flip-flops divididos por dos, que reciben una señal de entrada proveniente de un reloj. La señal del reloj es aplicada al primer flip-flop, el cual divide la frecuencia del reloj entre dos. La salida del primer flip-flop alimenta al segundo flip-flop, el cual a su vez vuelve a dividir la frecuencia entre dos, y así sucesivamente. De modo que cada etapa divide la frecuencia entre dos, un circuito con  $n$  etapas, divide la frecuencia de entrada en  $2^n$ . La salida de la última etapa es el desborde del flip-flop, llamado a veces bandera de desborde, el cual al ser probado por software genera una interrupción. El valor binario en el temporizador puede considerarse como la cuenta del número de pulsos del reloj ( o eventos), desde el momento en que el timer es inicializado.

#### 3.3.1 REGISTRO TMOD

El registro TMOD contiene dos grupos de 4 bits que fijan el modo de operación para los temporizadores, la parte alta de este registro corresponde al timer 1 y la parte baja al timer 0. Es importante aclarar que es registro no es direccionable a bit, lo que indica que debe ser cargado el valor completo del registro.

En la siguiente tabla se presentan, los registros de funciones especiales, con su dirección.

REGISTRO	PROPÓSITO	DIRECCIÓN	DIREC/BIT
TCON	Control	88h	Si
TMOD	Modo	89h	No
TL0	Parte baja Timer 0	8Ah	No
TL1	Parte baja Timer 1	8Bh	No
TH0	Parte alta Timer 0	8Ch	No
TH1	Parte alta Timer 1	8Dh	No
T2CON*	Control Timer 2	C8h	Si

RCAP2L*	Parte baja Timer 2 / captura	CAh	No
RCAP2H*	Parte alta Timer 2 / captura	CBh	No
TL2*	Parte baja Timer 2	CCh	No
TH2*	Parte alta Timer 2	CDh	No

\*Solo en los 8XX2

En la siguiente tabla se presenta la descripción del registro TMOD, donde se muestra la función de cada bit:

BIT	NOMBRE	TIMER	DESCRIPCIÓN
7	GATE	1	GATE BIT, cuando su valor es uno, el timer es activado solamente por la interrupción INT1
6	C/T	1	Bit selector entre contador1 y timer0.
5	M1	X	Modo de operación para el timer 1
4	M2	X	Modo de operación para el timer 1
3	GATE	0	GATE BIT para el timer 0
2	C/T	0	Selector entre contador 0 y timer 0
1	M1	0	Modo de operación para el timer 0
0	M0	0	Modo de operación para el timer 0

Los bits de modo de operación M0 y M1, determinan el modo de operación, para el contador/timer. Los modos de operación se muestran en la siguiente tabla:

M1	M0	MODOS	DESCRIPCION
0	0	0	Timer de 13 bits ( compatible con la familia MCS-48)
0	1	1	timer / contador de 16 bits.
1	0	2	timer / contador de 8 bits autorecargable.
1	1	3	Timer 0: TL0 y TH0 son 8 bits timers/contadores separados

### 3.3.2 REGISTRO TCON

El registro TCON contiene los bits de estado y control para los timer 0 y 1.

BIT	SIMB.	DIR / BIT	DESCRIPCION
TCON.7	TF1	8Fh	Bandera de desborde para el timer 1
TCON.6	TR1	8Eh	bit de arranque para el timer 1
TCON.5	TF0	8Dh	bandera de desborde para el timer 0
TCON.4	TR0	8Ch	bit de arranque para el timer 0
TCON.3	IE1	8Bh	bandera interrupción externa 1 flanco
TCON.2	IT1	8Ah	bandera interrupción externa 1 tipo
TCON.1	IE0	89h	bandera interrupción externa 0 flanco
TCON.0	IT0	88h	bandera interrupción externa 0 tipo

### 3.3.3 MODOS DE OPERACIÓN

#### **Modo de operación 0: Contador / Timer de 13 Bits**

El modo 0 es un timer de 13 bits que provee la compatibilidad con el predecesor del 8051. Este tipo de modo es probable que no se emplee en futuros diseños.

#### **Modo de operación 1: Contador / Timer de 16 Bits**

El modo 1 es un timer de 16 bits, que trabaja de igual manera que el de 13 bits con la exención que este timer opera con los tres bits del TLX, que el de 13 bits no utiliza. El reloj está aplicado a los registros combinados del timer, bajo y alto. Como los pulsos del reloj son recibidos, el timer empieza a contar 0000h, 0001h, 0002h, etc.

#### **Modo de Operación 2: Timer de 8 bits autorecargables:**

La parte baja del timer (TLX) opera como un timer de 8 bits mientras la parte alta del timer (THX), sostiene el valor de recarga. Cuando la cuenta del desborde de FFh a 00h, la bandera de desborde es fijada a 1, y el valor en THX es cargado en

TLX; la cuenta continúa de este valor a la próxima transición de FFh a 00h y así sucesivamente.

### Modo de Operación 3: Contador / timer split.

En el modo 3, el timer es partido en dos diferentes timers de 8 bits cada uno. TL0 y TH0 son separados, Los bits TF0 y TF1 son utilizados respectivamente para las banderas de desborde.

#### 3.3.4 FUENTES DE RELOJ

Existen dos formas posibles de conectar la fuentes de reloj, estas se selecciona por la escritura del bit contador/timer (C/T), que se encuentra en registro TMOD cuando el timer es inicializado. Las fuentes de reloj son utilizadas para generar intervalos de tiempo o para contar eventos.

En la figura 3.14 se explica pictóricamente el uso.

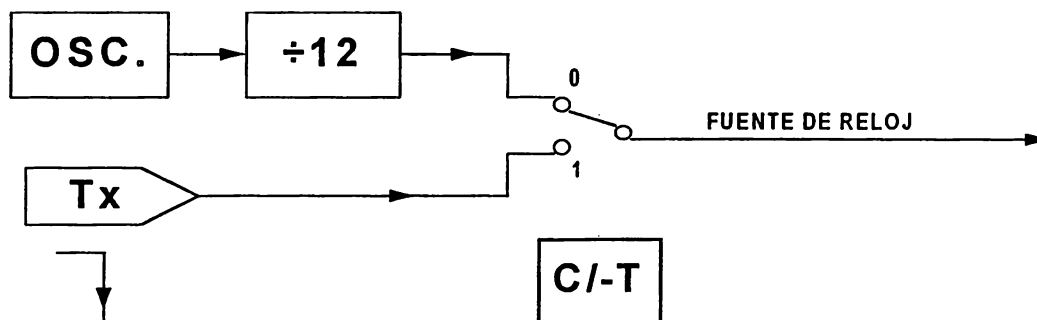


Figura 3.14. Fuentes de reloj

Si C/T tiene el valor de cero, el pin del reloj es conectado a la salida del chip oscilador. Si el C/T posee el valor de uno, el pin de reloj es conectado a un circuito que genera un pulso TTL, el cual es utilizado para la cuenta de eventos.

### 3.4 PUERTO SERIAL

El 8051 incluye un puerto serial en el chip, el cual opera en muchos modos sobre un amplio rango de frecuencias. La función esencial del puerto serial es realizar la conversión de paralelo a serie para los datos de salida y la conversión de serie a paralelo para los datos de entrada.

El Acceso por hardware para el puerto serie se realiza a través de los pines TXD y RXD. Estos pines funcionalmente están alternados con el puerto 3. En el bit P3.1 se ubica TXD y en el pin P3.0 se encuentra RXD.

La frecuencia de operación del puerto serial se denomina Baud rate, y puede ser fija ó variable. Si se usa un baud rate variable, el timer 1 suministra el reloj para el baud rate y este a su vez puede ser programado.

Dos registros con función especial, proporcionan el acceso por software al puerto serie, estos registros son el SBUF y el SCON. El buffer del puerto serial (SBUF) esta en la dirección 99H. Los datos a ser transmitidos se cargan el registro SBUF y son leídos en este registro cuando los datos son recibidos.

#### 3.4.1 REGISTRO SCON

El registro del puerto serial SCON se encuentra en la dirección 98H y es registro direccionable a bit, el cual contiene el status y los bits de control, para la comunicación. Los bits de control determinan el modo de operación y los bits de status indican el último bit para la transmisión o recepción. Los bits de status pueden ser probados por software o ser programados por medio de una interrupción.

En la siguiente tabla se presenta la descripción de los bits del registro SCON.

BIT	SIMB.	DIRECC	DESCRIPCIÓN
SCON.7	SM0	9Fh	bit para el modo 0
SCON.6	SM1	9Eh	bit para el modo 1
SCON.5	SM2	9Dh	bit para el modo 2
SCON.4	REN	9Ch	Receiver enable. Debe ser seteado para recibir los caracteres.
SCON.3	TB8	9Bh	Trasmit bit 8. Nueve bits son transmitidos en modo 2 y 3, fijados o limpiados por software
SCON.2	RB8	9Ah	Receive bit 8. Recibe nueve bits
SCON.1	T1	99h	Bandera para la interrupción en la transmisión
SCON.0	R1	98h	Bandera para la interrupción en la recepción

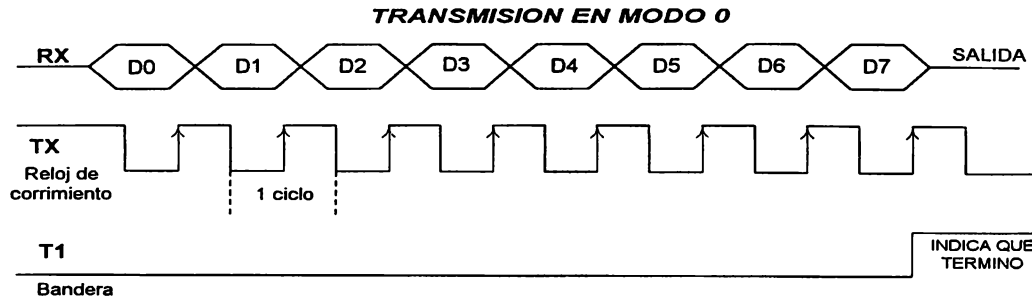
Modos de operación para el puerto serial:

SM0	SM1	MODE	DESCRIPCIÓN	BAUD RATE
0	0	0	Shift Register	Fija la frecuencia a (frec. Osc / 12)
0	1	1	UART de 8 bits	variable ( se setea por medio del timer )
1	0	2	UART de 9 bits	Fija, (frec. Osc. /12 o frec. Osc. /64 )
1	1	3	UART de 9 bits	variable ( se fija por medio del timer )

### 3.4.2 MODOS DE OPERACIÓN

#### MODO 0: SHIFT REGISTER DE 8 BITS

Los datos entran serialmente y salen a través de RXD y TXD. Ocho bits son transmitidos o recibidos con el bit menos significado (LSB) primero; y el baud rate se fija a un doceavo del valor de la frecuencia del oscilador. En este modo RXD es usado tanto para los datos de entrada como para los de salida. TXD es utilizada como el reloj.



*Figura 3.15. Modo de operación 0*

### **MODO 1: BAUD RATE VARIABLE CON UART DE 8 BITS**

En el modo 1 el puerto serial del 8051, opera como UART de ocho bits con una frecuencia variable. Una UART (Universal Asynchronous Receiver/Transmitter), es un dispositivo que recibe y transmite datos serialmente, con cada caracter del dato precedido por un bit de inicio (start bit) con un valor de cero y seguido por un bit de stop cuyo valor es un alto. Un bit de paridad algunas veces es insertado entre el último bit del dato y el bit de paro (stop bit).

La operación principal de la UART es la conversión paralelo a serie para los datos de salida y la conversión de serie a paralelo para los datos de entrada.

En modo 1, 10 bits son transmitidos en TXD o recibidos en RXD. Estos bits son: 1 bit para el bit de start (cuyo valor siempre es cero), 8 bits para los datos (enviándose el bit menos significativo primero, LSB), y por último el bit de paro (stop bit). Para una operación de recepción el bit de paro va a RB8 en SCON.

### **MODO 2: UART DE 9 BITS CON BAUD RATE FIJO**

Cuando  $SM1 = 1$  y  $SM0 = 0$ , el puerto serial opera en modo 2 como UART de 9 bits con baud rate fijo. Once bits son transmitidos o recibidos: Un bit de arranque, 8 bits de datos, noveno bit de datos programable y un bit de parada. En transmisión el noveno bit es fijado en TB8 del SCON (Como bit de paridad), mientras que, en

recepción es puesto en RB8. El baud rate en modo 2 es  $1/32$  o  $1/64$  de la frecuencia del oscilador.

### **MODO 3: UART DE 9 BITS .CON BAUD RATE VARIABLE**

Modo 3 es similar al modo anteriormente descrito excepto en que el baud rate es programado y proveído por el timer. En realidad los modos 1, 2 y 3 son muy similares.

#### **3.4.3 EL BAUD RATE DEL PUERTO SERIE**

Como se definió anteriormente el baud rate es de un valor fijo en los modos 0 y 2. En el modo 0 es siempre la frecuencia proveniente del chip oscilador entre 12. Usualmente un cristal se utiliza para el oscilador, pero se puede utilizar otro generador de pulsos también.

Con los modo 1 y 3 es posible generar las siguientes velocidades: 110, 150, 300, 600, 1200, 2400, 4800, 9600, 19200 (bits por segundo).

En la figura 3.16 se ilustra como son seleccionadas los valores para la frecuencia proveniente del oscilador, por medio del bit SMOD del registro PCON, seleccionamos el divisor para el modo determinado, dado que el registro PCON no es direccionable a bit es necesario modificar su contenido por medio del Acumulador.

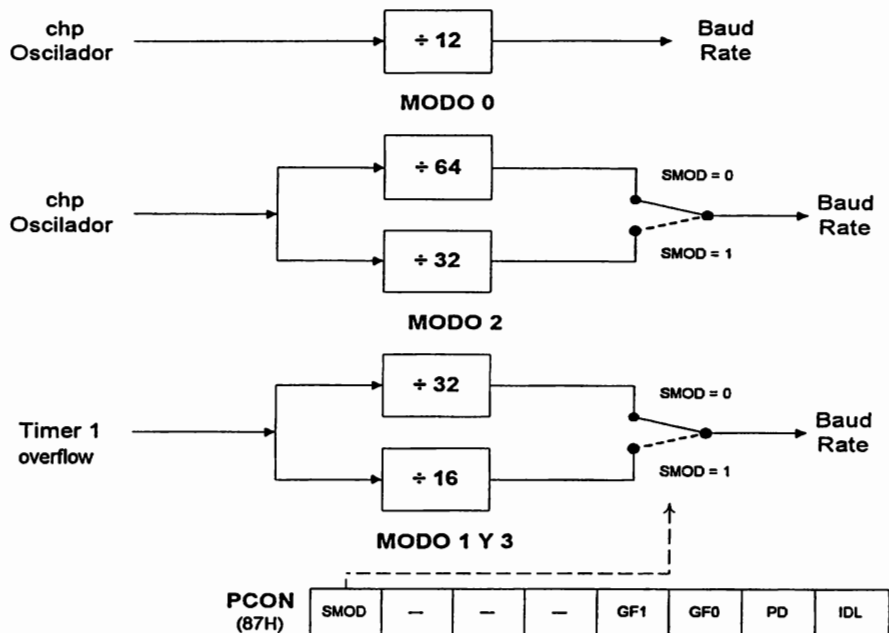


Figura 3.16. Selección de frecuencia

En la siguiente tabla se presenta los valores para las velocidades de transmisión, mas comunes, utilizando un oscilador de cristal de 12.000 MHz o 11.059 Mhz.

BAUD RATE	FREC. OSC.	SMOD	VALOR TH1	BAUD RATE REAL	ERROR (%)
9600	12.000 MHz.	1	-7 (F9H)	8923	0.07%
2400	12.000 MHz.	0	-13 (F3H)	2404	0.16%
1200	12.000 MHz.	0	-26 (E6H)	1202	0.16%
19200	11.059 MHz.	1	-3 (FDH)	19200	0.00%
9600	11.059 MHz.	0	-3 (FDH)	9600	0.00%
2400	11.059 MHz.	0	-12 (F4H)	2400	0.00%
1200	11.059 MHz.	0	-24 (E8H)	1200	0.00%

### 3.5 INTERRUPCIONES

Una interrupción es la ocurrencia de una condición o evento que causa una suspensión temporal de un programa mientras la condición es atendida por otro programa. Las interrupciones juegan un papel importante en el diseño e implementación de aplicaciones con microcontroladores.

Las interrupciones permiten a un sistema responder asíncronamente a un evento y tratar al evento mientras otro programa se encuentra en ejecución. Un sistema con manejo de interrupciones da la impresión o ilusión de estar realizando muchas cosas simultáneamente.

La Unidad Central de Proceso no puede ejecutar más de una instrucción a la vez; pero este puede suspender momentáneamente la ejecución de un programa, ejecutar otro, y luego retornar al primer programa. De alguna forma, puede decirse que es como una subrutina, conocida como "ISR" (Interrupt Service Routine). El CPU ejecuta otro programa (la subrutina) y luego retorna al programa principal. La diferencia es que un sistema con manejo de interrupciones, la interrupción no ocurre como un resultado de una instrucción (llamada de subrutina o "CALL"), sino al llamado asíncrono de un evento. Su ventaja radica en que no es conocido cuando el programa principal será interrumpido, por lo que no necesita rutinas de poleo (polling).

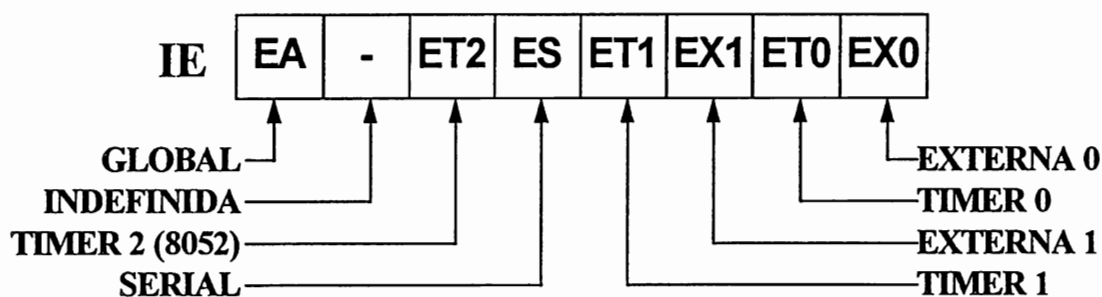
#### 3.5.1 ORGANIZACIÓN DE LAS INTERRUPCIONES

La familia MCS-51 posee cinco fuentes de interrupción: dos interrupciones externas, dos interrupciones por temporizadores (tres si es utilizado 8xx2), y una interrupción por puerto serial. Todas las interrupciones son deshabilitadas por un reset del sistema y habilitadas individualmente por software. En la figura 3.17 y 3.18 se muestra de forma completa el registro y estructura.

En el evento de dos o mas interrupciones simultáneamente o en la ocurrencia de una interrupción mientras otra interrupción es atendida, es examinada la prioridad y solo para conflictos de igual prioridad, se realiza una secuencia de poleo. La secuencia se realiza con el siguiente orden: Externa 0, Timer 0, Externa 1, Timer 1, Puerto Serial y Timer 2.

### 3.5.2 HABILITANDO Y DESHABILITANDO INTERRUPCIONES

Cada fuente de interrupción es habilitada o deshabilitada individualmente del registro especial de funciones direccionable a bit IE (Interrupt enable) localizado en la dirección de la memoria interna 0A8h.



Para habilitarlas hacer el bit = 1

Ejemplo: Habilitación de la interrupción de Timer 0 y Externa 0

```
SETB ET0 ;habilita interrupción del Timer 0
SETB EX0 ;habilita interrupción Externa 0
SETB EA ;habilitación global de interrupciones
```

Figura 3.17. Registro IE

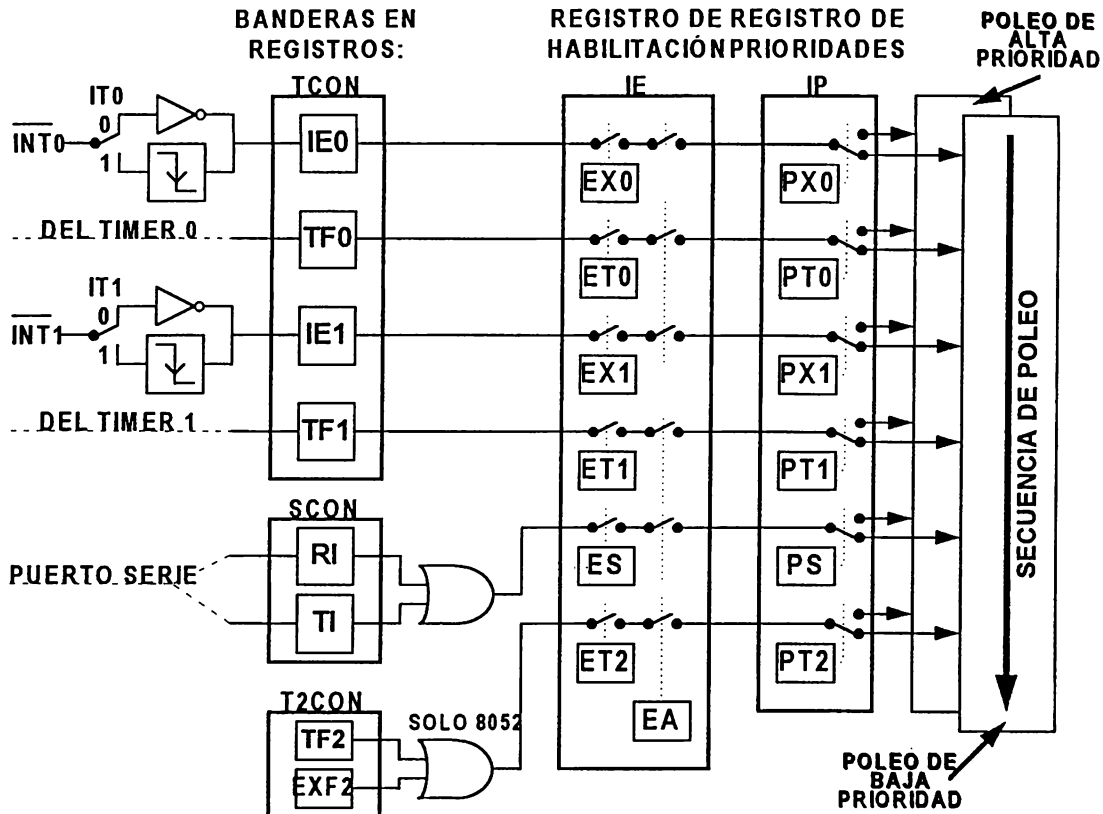


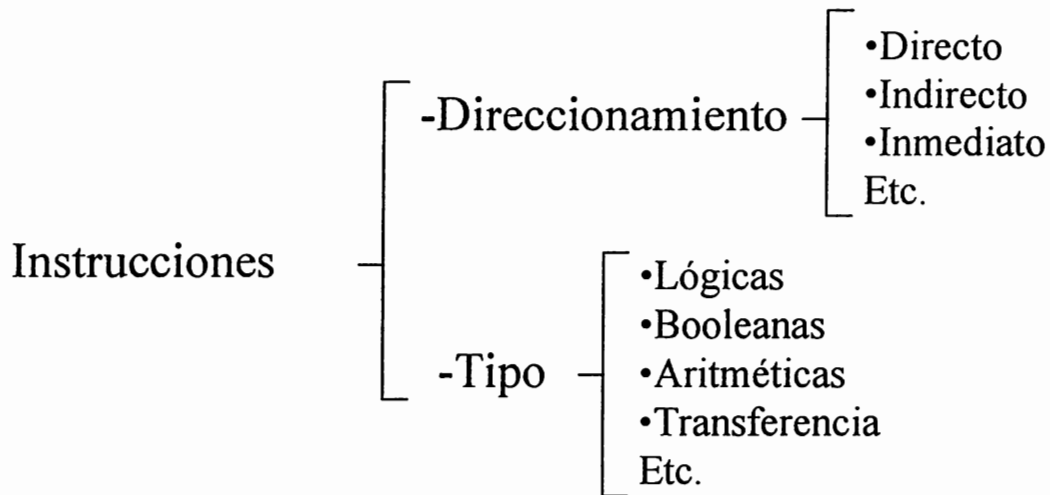
Figura 3.18. Estructura de las interrupciones en la familia MCS-51

### 3.6 SET DE INSTRUCCIONES

Un programa es el conjunto de instrucciones orientadas a establecer un proceder a ciertas situaciones relacionadas con el entorno de cualquier unidad de procesamiento. Las instrucciones son características inherentes de cada procesador o controlador. La familia MCS-51 posee instrucciones de opcodes de 8 bits, posibilitando un total de 255 instrucciones: 139 instrucciones de 1 byte, 92 instrucciones de 2 bytes y 24 instrucciones de 3 bytes. En la figura 3.18 se muestra la estructura completa de una instrucción y en la figura 3.19 se muestra la clasificación de las instrucciones.



Figura 3.18. Estructura de una instrucción



3.19. Clasificación general de las instrucciones

### 3.6.1 MODOS DE DIRECCIONAMIENTO

Cuando las instrucciones operan con datos, surge la pregunta “Donde están los datos?”, la respuesta se encuentra en los modos de direccionamiento. Los modos de direccionamiento son parte integral de cada set de instrucciones de cada ordenador, ellos permiten especificar la fuente y el destino de los datos en diferente forma dependiendo de la situación. La familia MCS-51 permite ocho modos de direccionamiento:

1. Registros
2. Directo
3. Indirecto
4. Inmediato
5. Relativo

- 6. Absoluto
- 7 Largo
- 8. Indexado

- **DIRECCIONAMIENTO POR REGISTROS**

Es el modo de direccionamiento que utiliza como destino o fuente a registros de trabajo ( R0 hasta R7, DPTR, etc.).

Algunas instrucciones son específicas para cada registro, como el acumulador, puntero de datos, etc., por lo que no es requerida una dirección, es decir, el opcode por si mismo indica el registro.

- **DIRECCIONAMIENTO DIRECTO**

El direccionamiento directo puede acceder cualquier variable situada en la memoria interna o registro. Un byte adicional es utilizado para especificar la localidad a utilizar.

- **DIRECCIONAMIENTO INDIRECTO**

Es la forma por la cual una variable indefinida puede su dirección determinada, computada o modificada mientras el programa es ejecutado. Es utilizada cuando se manipulan datos de localidades de memoria secuencialmente, números con múltiples precisiones, string o cadenas de caracteres, etc.

Los registros R0 y R1 pueden operar como punteros de datos, su contenido indica una dirección en memoria donde los datos serán leídos o escritos. En la familia MCS-51, el direccionamiento indirecto es representado por el símbolo comercial "@" (arroba o en inglés "at").

- DIRECCIONAMIENTO INMEDIATO

Cuando la fuente del operando es una constante (Un valor conocido), la constante puede ser incorporada en la instrucción como un byte de datos inmediatos. El símbolo por el cual se reconoce este tipo de instrucción es “#” (símbolo de número).

- DIRECCIONAMIENTO RELATIVO

El direccionamiento relativo es usado con ciertos tipos de instrucciones de salto (jump). La dirección relativa (o offset) es un valor de 8 bits con signo, un rango entre -128 y +127. Tiene la ventaja de ser independiente de la posición, aunque su rango es limitado.

- DIRECCIONAMIENTO ABSOLUTO

El direccionamiento absoluto es usado con las instrucciones ACALL y AJMP. Esta instrucción de 2kbytes permite saltos dentro de un rango de memoria de 2kbytes. Los 11 bits menos significativos determinan la dirección destino. El direccionamiento absoluto ofrece la ventaja de ser una instrucción corta, pero limita su rango hasta 2kbytes.

- DIRECCIONAMIENTO LARGO

Este tipo de direccionamiento es usado únicamente con las instrucciones LCALL y LJAMP. Su ventaja radica en saltos de 64k aunque depende de la posición y es una instrucción de 3 bytes.

- DIRECCIONAMIENTO INDEXADO

El direccionamiento indexado usa un registro base (tanto el contador del programa o el puntero de datos) y un offset (el acumulador) formando la dirección

efectiva para un JMP o MOVC. Su principal ventaja radica en que facilita el manejo de tabla.

### 3.6.2 TIPO DE INSTRUCCIONES

La clasificación del tipo de instrucción es la siguiente:

- **ARITMÉTICAS**

Las instrucciones aritméticas se presentan con más detalle en el apéndice. Este tipo de instrucción encierra todas las operaciones matemáticas básicas (+, -, x, /).

- **LÓGICAS**

Incluye las operaciones booleanas básicas (And, Or, Not, Xor) sea de bit, nibble o byte. (Ver apéndice)

- **TRASFERENCIA DE DATOS**

Se clasifican en:

- En RAM interna
- En RAM externa
- En Tabla

La instrucción MOV, permite acceso directo o indirecto, usada exclusivamente para uso de la memoria "on-chip" o RAM interna. Las de RAM externas son de tipo indirecto, es usada la instrucción MOVX y requiere del puntero de 16 bits (DPTR) o de punteros de 8 bits (solo para las localidades bajas :R0 y R1). Tabla es usada para acceso de códigos predeterminados con un significado distinto (por ejemplo el código ASCII), por lo que se utiliza la instrucción MOVC. (Ver apéndice)

- **VARIABLE BOOLEANA**

Incluyen todas las operaciones lógicas booleanas para un solo bit ( Booleanas como la prueba de bit), ver apéndice para más detalles.

- **SALTO DE PROGRAMA**

Son las instrucciones utilizadas para el control de flujo en un programa, incluyendo aquellas que llaman o retornan de una subrutina o saltos condicionales o incondicionales. Los saltos pueden ser:

- Saltos de tabla
- Salto de interrupciones y subrutinas
- Saltos condicionales

## CAPITULO IV

### DISEÑO DEL MÓDULO DE CONTROL

#### 4.1 DISEÑO DEL MÓDULO MÍNIMO

En las secciones anteriores se ha dado un bosquejo de uno de los muchos ambiente industriales donde podemos aplicar controladores basados en microcontroladores. En esta sección se procederá a establecer las especificaciones del diseño para tener una visión clara de las funciones que realizará y empezar de esta manera la materialización del equipo.

Los microcontroladores han influenciado profundamente el avance y vistosidad de muchas disciplinas; especialmente la de instrumentación y control. Su implementación no solo ha acarreado la simplificación del hardware y mejoramiento de la confiabilidad; sino también permitió la optimización de su funcionamiento y las capacidades de diagnóstico poderosos. Lo que no hubiese sido posible por hardware dedicado.

Las características de los microcontroladores cada vez son mejores en términos de arquitectura, velocidad, tamaño y otros. Y dichas tendencias continuarán.

Un sistema basado en microcontroladores por muy mínimo que sea requiere, además de éste, de otros elementos de apoyo para poder funcionar y desempeñar cualquier tarea útil. En general, estos elementos adicionales son: fuente(s) de alimentación, reloj, compuertas lógicas, latches, buffers y memoria externa.

La configuración básica, sin embargo, puede alejarse del caso generalizado dependiendo de las particularidades de cada microcontrolador; ya que mientras algunos incluyen ya varios de los elementos mencionados, otros necesitan, además, de algunos otros dispositivos electrónicos especiales al fin de estar acorde a la necesidad. En éste capítulo se presentarán con algún detalle éstos elementos; inicialmente tratando de ser lo más general posible, para finalizar con la presentación del sistema MCS-51, el cual se convertirá en la base de todo el trabajo posterior.

A fin de facilitar la construcción y depuración del sistema, éste ha sido diseñado en forma de módulos. El módulo principal contiene los elementos mínimos, es decir : CPU, memoria, decodificadores, reset, reloj y buffers. Además se incluyen algunas modificaciones que contribuyen a darle flexibilidad deseable en un sistema de control.

#### 4.1.1 RELOJ DEL SISTEMA

Las señales de reloj en un sistema basado en microcontroladores son las formas de onda que se utilizan para sincronizar la operación del sistema.

Ciertos microcontroladores necesitan de una circuitería un poco más complicada para generar las formas de onda de reloj, requiriendo hasta dos señales de reloj generadas externamente las cuales deben alternar entre 0 y +12V y que además no deban de traslaparse.

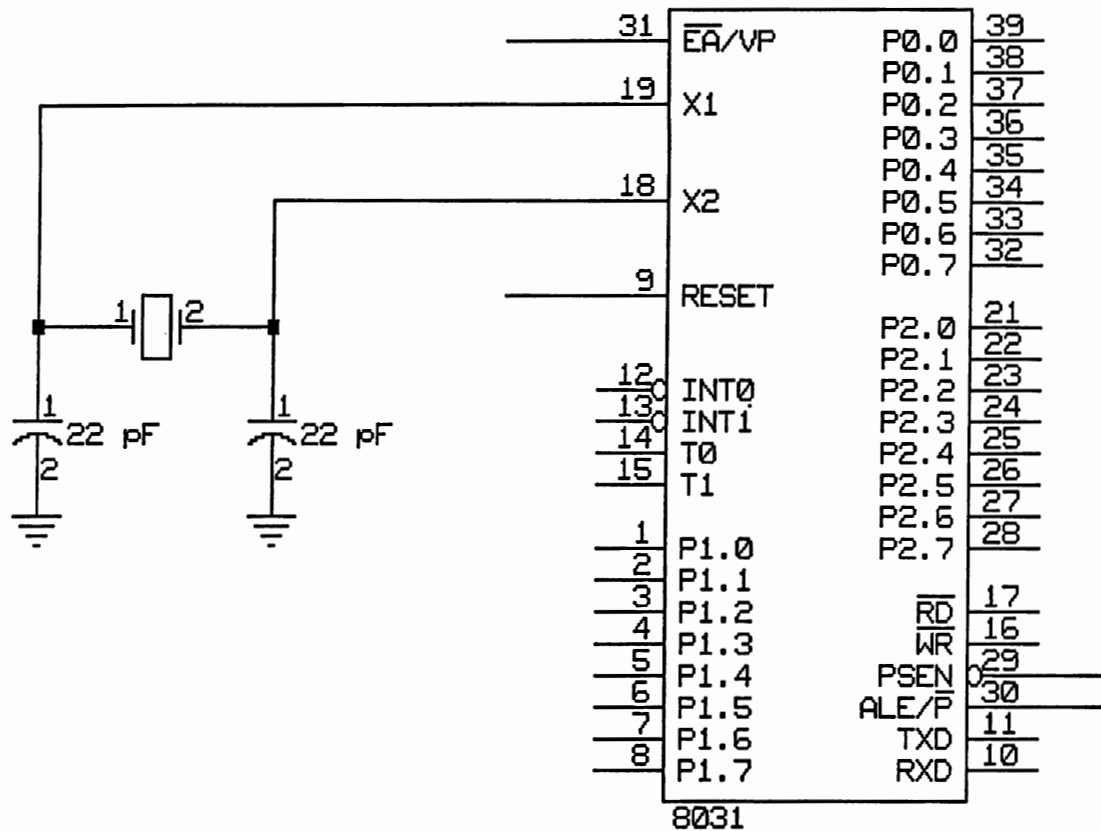


Figura 4.1. Circuito de reloj para SBC utilizando MCS-51

La familia MCS-51 tiene requerimientos de reloj poco sofisticados. Requiere de una señal generada externamente y que alterne de 0 a 5V, es decir una señal compatible con TTL. En sus terminales pueden conectarse un cristal con condensadores de 20 pF para estabilización (eliminar ruido) o un oscilador integrado.

El circuito de la figura 4.1, puede utilizarse para la aplicación. Lo único que faltaría por determinar en tal caso sería la frecuencia del cristal la cual quedará determinada en su límite superior por la máxima frecuencia que pueda soportar el microcontrolador. Para nuestro caso es de 12 MHz.

Aunque la frecuencia mínima puede ser del orden de 500 kHz, no es recomendable solo para fines de prueba en las etapas de desarrollo de un sistema. Cuando se opera cerca de la máxima frecuencia de trabajo, un oscilador es lo más recomendado para evitar irregularidades de temporización.

Para el módulo de control o SBC, se utilizará un cristal de 11.0592 MHz, debido a que con dicha frecuencia se puede obtener generación de baudios exactos. No es posible introducir un oscilador integrado debido a que el mercado no existen osciladores de 11.0592 MHz.

En conclusión, inicialmente se tomará la señal de reloj de un cristal de 11.0592 MHz, y una vez se esté en operación normal se utilizará el circuito de la figura 4.1. El motivo se debe a la obtención de baud rate con cero error. El circuito de reloj es particularmente sensible, por lo cual se recomienda mantener las patillas de los elementos lo más cortas posibles para evitar la creación de cargas inductivas y capacitivas parásitas.

#### 4.1.2 RESET DEL SISTEMA

Este circuito es muchas veces ignorado en los documentos que tratan acerca del diseño de sistemas con microcontroladores. Sin embargo tiene una gran importancia, la cual se manifiesta tanto al momento de energizar el sistema (función de inicialización) como al momento de la ejecución de programas que no se estén ejecutando de la forma esperada (reinicialización).

La función de inicialización es indispensable, ya que el microcontrolador exige que el terminal RESET (Pin 9) se mantenga en nivel alto por lo menos 2 ciclos de máquina.

Dado que no existe ninguna garantía de que dos fuentes de alimentación tengan el mismo tiempo de subida, el diseñar un circuito de inicialización que solo cumpla los requisitos mínimos es completamente insensato sobre todo teniendo en cuenta que no existe ningún tipo de limitación respecto al tamaño del pulso.

Es por ello que el circuito de reset a utilizar es este sistema tendrá un condensador de  $10\mu\text{F}$  y una resistencia de  $6.8\text{k}\Omega$ . Con este circuito, se puede lograr un pulso lo suficiente para la obtención de un reset tanto al iniciar como reiniciar el sistema. Aun cuando se pueda creer que con estos valores se obtiene un pulso excesivo visto desde la perspectiva del microcontrolador, para la perspectiva humana es despreciable. Es necesario aclarar que con este diseño no se ignora ningún requerimiento en el microcontrolador, se puede comprobar tomando los valores involucrados en los dispositivos, los niveles lógicos, tiempos, etc.

En la figura 4.2 se muestra el circuito de reset a emplear. Este circuito incorpora tanto el reset manual como el reset automático. El reset manual lo proporciona el pushbutton, el cual al ser presionado aplica el voltaje de la fuente al pin RESET del microcontrolador. La resistencia de  $100\ \Omega$  es opcional, sirve como limitante de corriente.

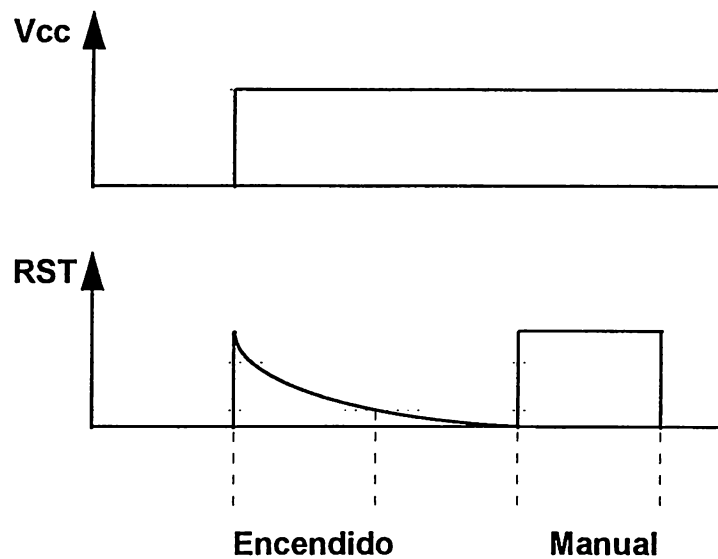
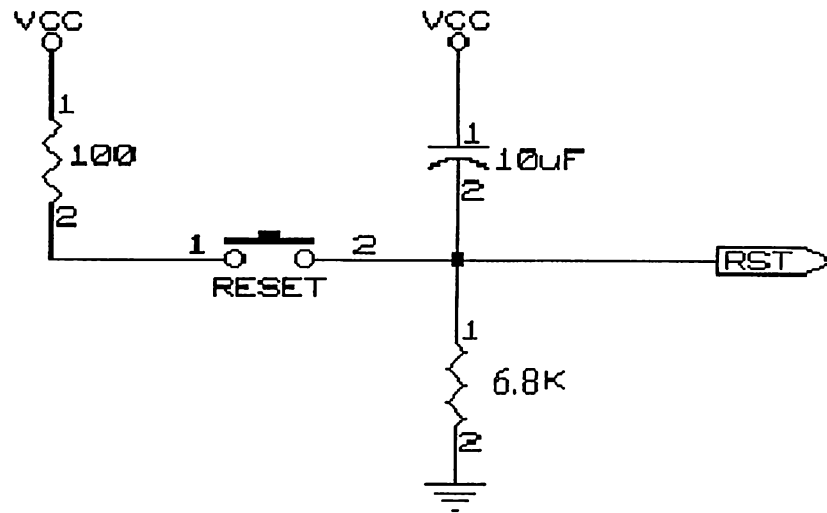
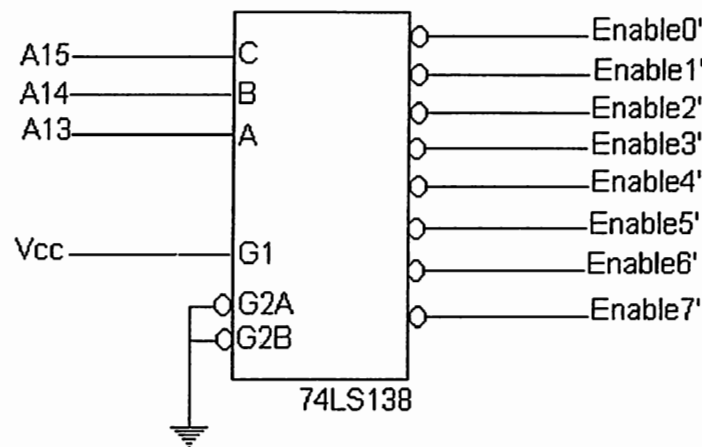


Figura 4.2. Reset del SBC

Las características del pulso de reset quedará determinada por el producto RC. Las gráficas son representaciones ideales del pulso en el pin de reset , con fines de brindar una idea del resultado del circuito descrito anteriormente.

### 4.1.3 DECODIFICADORES

Un decodificador es un circuito lógico de múltiples salidas que convierte entradas codificadas en salidas codificadas equivalentes de otro código. El código de entrada tiene, en general, menos bits que el código de salida y hay un mapeo uno a uno de las palabras de código entrantes en las palabras código salientes. En un mapeo uno a uno, cada palabra código entrante produce una palabra código saliente diferente.



*Figura 4.3. Decodificador 74LS138*

En el microcontrolador los impulsos de selección están asociados a las señales del puerto PSEN, RD y WR (todos de lógica negativa). Cuando se quiere realizar una selección, se pueden seguir dos estrategias: Decodificación ambigua (conocida como parcial) y Decodificación completa (o absoluta).

Utilizando codificación absoluta, garantiza que un dispositivo o posición de memoria será accesado por una única dirección colocada en el bus de direcciones. Su desventaja es que se necesitarán más compuertas y demultiplexores para generar los pulsos de selección por lo que se elevan los costos.

Por otra parte, la decodificación ambigua el circuito decodificador es simple, ya que se utilizan menos líneas de dirección, quedando las otras como estados “no importa” (it doesn't care). Esto permite que sea utilizado en sistemas industriales dedicados ya que por lo general estos no hacen uso de todo el mapa de direcciones.

Para la aplicación, como se muestra en la figura 4.3, se utiliza un decodificador 74LS138, permitiendo bloques de 8k.

Las líneas del bus de dirección A15, A14 y A13 forman las combinaciones posibles de decodificación, permitiendo conectar a la salida ya sean dispositivos de memoria como cualquier otro según la aplicación. Esta configuración reduce circuitería y al mismo tiempo permite optimizar la conexión de cualquier otro elemento periférico.

Es necesario aclarar que gracias al mapeo, es posible explotar el uso de los puertos al máximo, permitiendo media vez se utilicen las interfaces adecuadas, la conexión al puerto de lámparas, motores, relés, parlantes o speakers, etc.

#### 4.1.4 LATCHES

Una colección de dos o más flip-flops D con una entrada de reloj común se conoce como un registro o latch. Los registros se usan a menudo para almacenar una colección de bits relacionados.

La familia MCS-51 utiliza la multiplexación de uno de sus puertos como bus de datos y parte baja del bus de dirección, por lo que para la extracción de datos de memoria externa, se hace necesario retener la parte baja de la dirección para posterior uso.



Aun cuando los microcontroladores tienen una cierta cantidad de registros de almacenamiento, estos pueden ser utilizados solamente en la manipulación temporal de datos y no pueden todos los modelos almacenar instrucciones de programas.

Las instrucciones de programación deben ser almacenadas en elementos de memoria exterior, aunque en algunos casos, toda la memoria necesaria puede estar incluida en el chip del microcontrolador, pero en la mayoría de los casos se requiere de memoria externa a éste.

Aunque al hablar de memoria externa puede abarcarse muchas formas de almacenamiento que pueden ir desde las memorias de ultra-alta velocidad como las utilizadas en los caches de memoria hasta las formas de almacenamiento lentas como los cassettes o las tarjetas perforadas; cuando se habla de sistemas con microcontrolador por lo general se está refiriendo a formas de memoria que se encuentran ubicadas en medio de esos dos extremos de velocidad y densidad de almacenamiento.

En el ambiente de los microcontroladores se trabaja principalmente con dos niveles de memoria; memoria de corto plazo de alta velocidad (RAM MOS, núcleo magnético, ROM MOS, etc.) y memorias de almacenamiento masivo de mediano plazo (floppy disk, memoria de burbuja, memoria de tambor, CD ROM, etc.). Para un sistema mínimo lo que interesa es la memoria de trabajo de corto plazo. Las características importantes de estas unidades de almacenamiento incluyen tiempos de acceso rápido y capacidad de acceso aleatorio.

Dado que las memorias de núcleo magnético se han ido rezagando, en la actualidad las memorias semiconductoras se usan casi invariablemente en todas las microcomputadoras y controladores.

Las memorias semiconductoras se dividen en dos clases principales: ROM (Memoria de solo lectura) y RAM (RWM, memoria de escritura y lectura, memorias de acceso aleatorio o memorias de escritura/lectura aleatoria). La ROM se utiliza para almacenar datos específicos, pasos permanentes de programación o datos. Los contenidos de estas posiciones de memoria son considerados permanentes y no pueden cambiarse fácilmente.

La memoria de lectura/escritura, por otra parte, es utilizada para almacenar datos que cambian cuando la computadora está funcionando. Estos datos pueden ser resultados generados por un programa, o bien pueden ser parte de las instrucciones de un programa del usuario.

Por otra parte, la RAM se divide en dos categorías: RAM estática y RAM dinámica. La RAM estática tiene la característica de que conserva la información durante el tiempo que se mantenga la alimentación al circuito; en cambio las memorias dinámicas requieren que la información esté siendo regenerada periódicamente (conocido como "refresh"). La compensación a este inconveniente es que debido a su estructura más simple es posible obtener mayores densidades de almacenamiento, menor disipación de potencia y mayor velocidad.

Las formas de regenerar la información en una memoria dinámica puede ser a veces complicadas, ya que por lo general los microprocesadores y microcontroladores no proporcionan ningún soporte especial para este tipo de memorias. La familia MCS-51 es uno de los pocos microcontroladores de 8 bits que facilitan de alguna manera el uso de memorias dinámicas, aunque siempre requerirá de circuitería adicional para que el sistema funciones correctamente.

El uso de memoria dinámica se justifica únicamente cuando se necesite hacer uso de grandes cantidades de memoria. En sistemas pequeños, el uso de memoria

dinámica constituye más un problema que una ventaja. Es por ello que estos utilizan casi invariablemente memoria estática para el almacenamiento de datos cambiantes.

Sin importar que tan complejo o sencillo sea el sistema que se está desarrollando, éste es un elemento indispensable. La razón de ello es que un microcontrolador no hará nada útil si se carece de un lugar donde almacenar datos y programas.

La mayoría de sistemas pequeños como el que se quiere construir no necesitan gigantescas memorias para desempeñar su trabajo satisfactoriamente, aunque esto no significa que no sea posible aumentar considerablemente la memoria en caso que surja la necesidad en un futuro.

Por el momento, el sistema dispondrá de 8k de memoria Rom mediante el uso de una EPROM 2764 y, aunque no es tan necesario por ahora, 8k de RAM proporcionados por una RAM estática 6264. En la figura 4.5 se muestra el patillaje de las memorias a utilizar y una configuración posible para una necesidad de memoria en el sistema de 16k.

El motivo por el cual se seleccionan bloques con rangos de 8k, es con propósitos de desarrollo. Con esta cantidad de memoria, es posible construir un sistema que permita ejecutar programas de prueba sin necesidad de recurrir a la programación de memorias, además, permitirá modificarlo de una forma rápida en caso de anomalías en el programa. Esto implicará una ventaja económica, de hardware y de pruebas rutinarias.



Para el sistema mínimo a desarrollar, se utiliza un decodificador, por lo que el pin 20 (Chip select) de ambos IC se conectan a la línea respectiva del selector de habilitador de bloque de 8k realizado por 74LS138 (Y0 para EPROM y Y1 para Chip select1 de la RAM). De la EPROM, el pin 22 (Output enable) se conecta a la línea 'PSEN del bus de control y PGM a Vcc. En la RAM, el pin 26 (Chip select2) a Vcc, el pin 27 (Write) a la línea del bus de control 'WR y el pin 22 (Output Enable) conectado a la salida de la combinación de 'PSEN y 'RD. La combinación de 'PSEN y 'RD ('PSEN and 'RD) es necesaria, ya que permite ocupar a la RAM como memoria de programa, permitiendo sin necesidad de reprogramar continuamente EPROMs cada vez que se realicen cambios en el programa monitor.

#### 4.1.6 COMUNICACION SERIAL VÍA RS-232C

La conexión correcta entre equipos persigue permitir comunicación libre de errores. El CCITT es una entidad encargada de regular y estandarizar la conexión entre equipos sean terminal o de comunicación de datos. RS-232C nace como la recomendación V.24. En la actualidad todo sistema que permita la interconexión con otros similares utiliza esta interface. Sus características con respecto a las señales TTL se presentan en la figura 4.6.

	<b>RS-232C</b>	<b>TTL</b>
MARCA	-12 V	+5 V
ESPACIO	+12 V	0 V

*Figura 4.6. Relación eléctrica entre RS-232C y comunicación TTL*

Entre las características de la interface RS-232C son el control de flujo y una distancia máxima de 15 metros (50 pies). Si realmente se desea tener acceso con cualquier otro terminal, es necesario adaptar la señal TTL de comunicación brindado

por el microcontrolador a niveles eléctricos y distancia acordado en la recomendación V.24. Es posible la utilización de elementos discretos, con el inconveniente de involucrar muchos componentes y ocupar un espacio considerable.

Es por eso que se utiliza el circuito integrado MAXIM232CPE. Este dispositivo integra tanto los buffers y condicionadores de señales necesarios para interconectar dispositivos a niveles TTL RS-232C y viceversa, por lo tanto, se asegura comunicación bidireccional full o half duplex. Además incluye etapa de control de flujo, permitiendo comunicación con o sin handshake. La conexión se muestra en la figura 4.7.

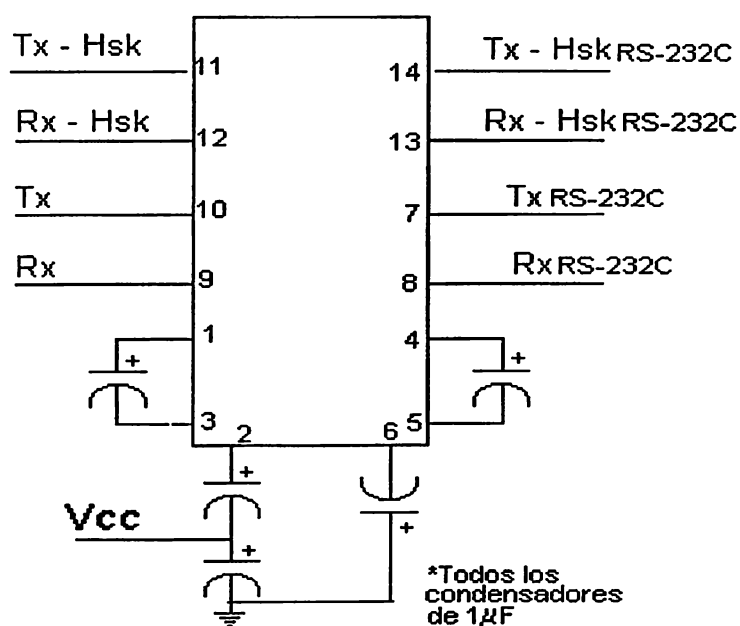


Figura 4.7. MAXIM232CPE

#### 4.1.7 FUENTE DE ALIMENTACION

Una fuente de poder puede construirse utilizando un transformador conectado a una alimentación de corriente alterna para reducir el voltaje al nivel deseado,

rectificando después con un circuito de media onda o de onda completa, filtrando el voltaje para eliminar el fenómeno de rizo y por último regulando con un transistor.

Pero es necesario incluir circuitería extra para referencia, amplificador de error, dispositivo de control, protección contra sobrecarga, inmunidad a efectos térmicos, limitador de corriente en caso de cortocircuitos y un amplio rango de voltajes de conexión sin dejar de regular.

Si se realizara todo la circuitería con componentes discretos, se requeriría de un gran espacio y un incremento sensible del costo. Por lo tanto, se utilizan circuitos integrados que integren todas las funciones o requisitos descritos anteriormente. En el mercado existen gran variedad de estos elementos o circuitos integrados, pero cabe recordar que, para simplificar diseño y reducir costo, se unifico el voltaje de alimentación en la tarjeta a +5 Vdc. El dispositivo 7805C, es un regulador fijo de +5 VDC con capacidad de 1 A. Requiere poca circuitería, dos condensadores, una para su entrada y otro en su salida.

Será necesario incluir la etapa de transformación y rectificado, por lo que el filtrado y la regulación serán realizados por el 7805C y su circuitería.

Su espacio es mínimo comparado con una fuente discreta de regulación. Parte importante que influyeron en la selección del microcontrolador es su robustez a la fuente de voltaje.

Por lo tanto, el circuito integrado 7805C se adapta a las exigencias del microcontrolador; brinda todas las características tanto de regulación como rechazo a fenómenos comunes provenientes de la línea de alimentación comercial, tales como picos y bajos de tensión.

#### 4.1.8 MAPEO DE MEMORIA

Se denomina mapeo de memoria a la forma en que está distribuida la selección de elementos de memoria cuya selección se realiza mediante el decodificador 74LS138.

Por medio de las líneas A13, A14 y A15 del bus de direcciones seleccionamos ocho posibles salidas las cuales corresponden desde Y0 a Y7

Las salidas son bajas activas, al habilitarse una determinada salida, se coloca un nivel lógico en bajo, por lo que si se necesita seleccionar un dispositivo por medio de un pulso positivo, ésta debe negarse.

Cada bloque esta formado por 13 líneas (  $2^{13} = 8192 = 8Kb$  ), las 12 primeras líneas ( A0 a A12 ), contienen la dirección de la localidad y la última selecciona el dispositivo.

Para el usuario quedan libres tres bloques de 8K, por lo que se tiene 24K (24576 localidades), ubicadas desde la dirección 4000<sub>H</sub> hasta la dirección 9000<sub>H</sub>.

En la siguiente tabla se muestra el mapeo de memoria para el sistema controlador:

<b><i>MAPEO DE MEMORIA</i></b>		
<b><i>YX</i></b>	<b><i>DESCRIPCIÓN</i></b>	<b><i>RANGO</i></b>
Y7	PANTALLA DE CRISTAL	E000 <sub>H</sub> ~ FFFF <sub>H</sub>
Y6	TECLADO	C000 <sub>H</sub> ~ DFFF <sub>H</sub>
Y5	INTERFAZ DE POTENCIA	A000 <sub>H</sub> ~ BFFF <sub>H</sub>
Y4	LIBRE	8000 <sub>H</sub> ~ 9FFF <sub>H</sub>

Y3	LIBRE	6000 <sub>H</sub> ~ 7FFF <sub>H</sub>
Y2	LIBRE	4000 <sub>H</sub> ~ 5FFF <sub>H</sub>
Y1	LIBRE/MEMORIA DE DATOS	2000 <sub>H</sub> ~ 3FFF <sub>H</sub>
Y0	MEMORIA DE PROGRAMA	0000 <sub>H</sub> ~ 1FFF <sub>H</sub>

## 4.2 TEORIA SOBRE INTERFACES Y PERIFÉRICOS

Un periférico es todo aquel dispositivo con funciones autónomas o semiautónomas específicas; a veces sirve como entrada y/o salida en un sistema basado en computadoras.

Una interface es todo aquel enlace entre dispositivos de características diferentes tanto eléctricamente, como en velocidad, etc.

Ambos elementos están involucrados en todo diseño, el periférico (para este caso pantalla de cristal líquido y teclado) se encargará del enlace entre operario y la tarea en que se encuentra envuelto el microcontrolador. La interface permite el adaptar y utilizar el periférico según las características de estos y las del microcontrolador.

### 4.2.1 MANEJADOR (BUFFER)

Los microcontroladores al igual que todo dispositivo electrónico, tiene especificados ciertos límites de operación, ya sea de voltajes máximos y mínimos, temperaturas extremas, disipación máxima, etc.

En cuanto la corriente que puede manejar en cada terminal, la familia MCS-51, depende del puerto en uso.

El puerto 0 está especificado para absorber dos cargas TTL estándar en cada uno de sus pines. En caso de cargas LS TTL, puede manejar hasta 8 directamente en cada pin.

Como el puerto 0 es de vital importancia para la conexión con otros dispositivos, con el fin de mejorar su fan out, se conectan resistencias de pull-up de  $10k\Omega$  (puede verificarse con los datos del fabricante anexados en los apéndices).

Los puertos 1, 2 y 3 están especificados para absorber una carga TTL estándar en cada uno de sus pines. En caso de cargar LS TTL, puede manejar hasta 4 directamente en cada pin. Este máximo especificado puede presentar de alguna manera una limitante al diseño.

Para ello, se podría diseñar un circuito tomando en consideración cada microamperio a fin de no sobrepasar el máximo manejable por el dispositivo en cuestión. Sin embargo si el circuito construido se encuentra en el límite de tolerancia sería muy arriesgado utilizar algún equipo de medición (sonda lógica, frecuencímetro, voltímetro, etc.), ya que la carga presentada por estos sumada a la de los otros dispositivos conectados a una línea particular, sobrepasará la capacidad de esa línea.

Una mejor manera de superar la limitación de excitación de las líneas de un microprocesador es utilizar manejadores (Buffers), con lo cual se puede llegar a un punto en el cual la carga ya no sea un factor que limite excesivamente al diseño del sistema.

Un manejador es un amplificador que puede utilizarse para incrementar la capacidad de manejo de corriente en una línea del microcontrolador. En la

actualidad, en el mercado se encuentran dispositivos de colector abierto, los cuales permiten la conexión de más elementos digitales a un solo pin.

Si se desea manejar cargas de mayor voltaje y corriente, será necesario la utilización de transistores y relés. El transistor será el accionador del rele. El rele realizará la función tanto de aislamiento entre niveles de voltajes diferentes como elemento final de Encendido/Apagado.

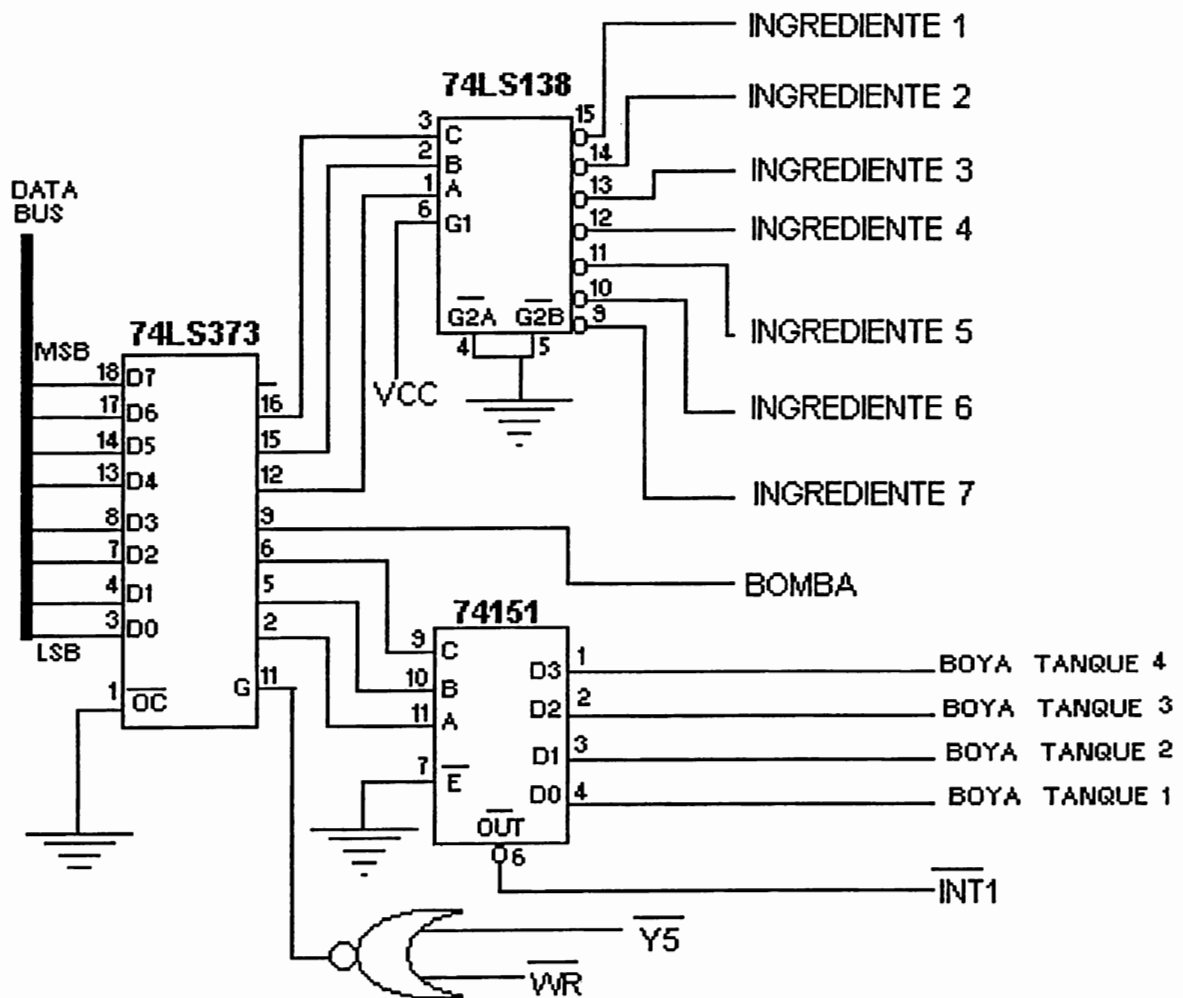
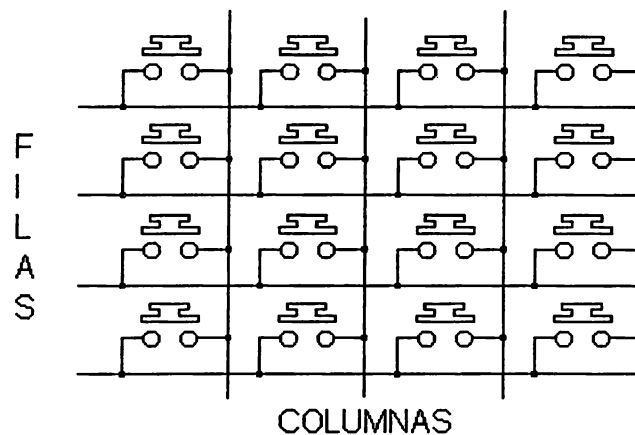


Figura 4.8. Etapa de potencia

## 4.2.2 TECLADO

El teclado es el periférico mediante el cual el usuario, se comunica con el sistema, este método es simple; mediante la presión de una tecla, un código es generado para la tecla pulsada, generado una vez este código es procesado por el microcontrolador.

Básicamente un teclado es una matriz de elementos interruptores. Cuando es presionado una tecla se crea un dato significativo, el teclado mediante una etapa lógica detecta este dato y lo convierte a un valor hexadecimal para que el microcontrolador pueda operarlo.



Cuando se presiona una tecla se cortocircuita una fila con una columna. El cierre es detectado por la aplicación de un nivel bajo lógico a las filas, tanto las filas como las columnas son rastreadas para encontrar el nivel bajo, el rastreo de las filas o columnas puede ser simultáneo o individual, depende del método que se emplee para esto.

Después que el proceso de detección e identificación ha sido completado, un pequeño retardo es colocado, antes de reanudar el escaneo. Este retardo de tiempo es necesario, para evitar el efecto que producen los contactos cuando la tecla es presionada. Si el retardo en el escaneo no se llevara a cabo, se tomarían medidas

erróneas, el tiempo que dura el efecto de los contactos es de 5 a 20 mseg. aproximadamente.

Como el microcontrolador trabaja con datos en forma hexadecimal, se necesita de un codificador de teclado, que convierta el dato proveniente de este a un valor en hexadecimal, tal conversión se realiza por medio de IC MM74C922, el cual es un codificador CMOS para teclados basados en interruptores SPST ( Single Pole Single Toggle: Simple posición, Simple Palanca).

Una característica importante de este dispositivo es que no preciso utilizar diodos para eliminar el rebote en el teclado. Por medio de un capacitor puede generarse el retardo de tiempo para el rastreo de los pines del teclado, pero si se desea puede conectarse un reloj al pin 5 (OSC), él cual determinara el tiempo de rastreo. El valor del capacitor se determina por medio de la curva que presenta el fabricante (ver anexos). Para el caso es de  $0.1\mu\text{F}$ , él cual genera una frecuencia de 600 Hz; es decir el tiempo en el cual se estará rastreando la señal de tecla es de 1.6 mseg. Tiempo suficiente para que el microcontrolador pueda leer los datos mientras el operario pulsa las teclas.

Otra característica es la de permitir almacenar la última tecla, en una especie de registro temporal.

El tiempo que tarda el IC en producir los datos esta determinado por la relación  $0.7 RC$  en donde R tiene el valor aproximado de  $10\text{K}\Omega$ , y C esta determinado por el valor del capacitor que se conecte al pin 6 (KMB), para un capacitor de  $1\mu\text{F}$  es de 7mseg.

El tiempo que emplea el IC para omitir los rebotes después que se presionado una tecla, esta determinado por el valor del capacitor que se conecta en el pin 6 (KBM), (ver en anexo hojas de fabricante para el 74C922).

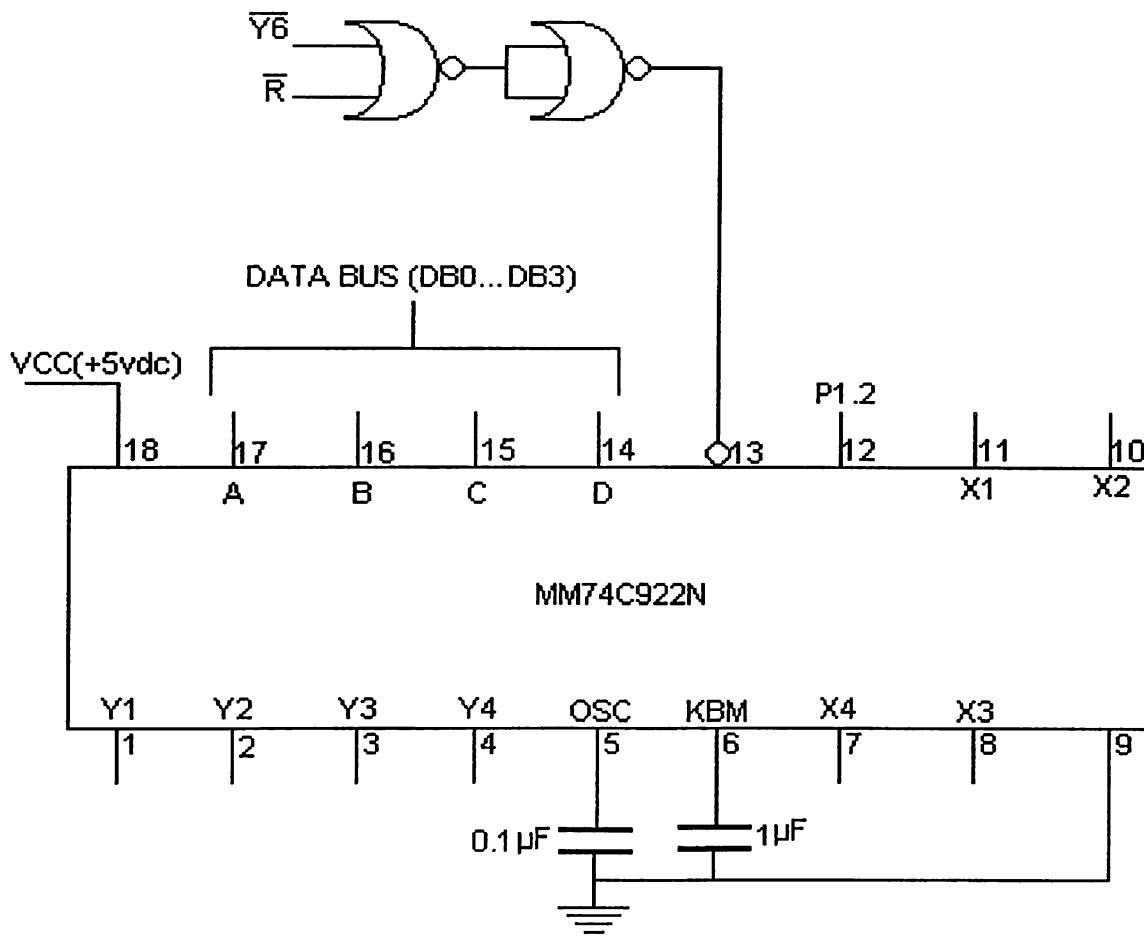


Figura 4.9. Conexión del interface de teclado

El teclado será manejado mediante un bit del Puerto 1. Se utiliza esta opción debido a que se utilizarán las interrupciones netamente para detener el proceso debido a una situación anómala. Si se utiliza vía interrupción, se requiere agregar

circuiteria extra y una rutina de poleo para determinar de quien proviene la petición. Debido a que no es requerido un reset por teclado (como en las computadoras), sino, simplemente la introducción de datos, es que se puede realizar la configuración expuesta en la figura 4.9.

#### 4.2.3 PANTALLA DE CRISTAL LIQUIDO

Las pantallas de cristal líquido, son dispositivos cuyo fenómeno se basa en la distorsión de luz reflejada a éste (regiones con diferentes índices de refracción) debido a la aplicación de un potencial. Este fenómeno fue descubierto en los laboratorios de RCA en 1968.

Entre sus características principales están:

- Requerimientos mucho menores de potencia versus a las luminosas (leds como ejemplo).
- Amplia variedad de diseño de caracteres.

Debido a que no producen luz propia, su consumo se reduce al orden de microwatts, además, permiten una mejor resolución a la generación de caracteres y gráficos, por lo que son ampliamente utilizados en calculadoras, relojes, etc.

Debido a que se puede lograr un ahorro de potencia considerable (comparado a un display por leds) y generación más clara de caracteres alfanuméricos para mensajes, influyó en la decisión de utilizarlos. Aunque su costo es elevado versus los leds, el ahorro de potencia a la larga justifica su inversión.

El display o pantalla seleccionada es el modelo DMC-16230 de la empresa OPTREX CORPORATION, con capacidad de desplegar dos líneas con dieciséis caracteres (un total de 32 caracteres en pantalla). Entre los caracteres que se pueden desplegar se encuentran parte del código ASCII, alfabeto japonés, alfabeto griego (las más importantes), símbolos aritméticos, etc. (ver apéndice para la tabla

de fonts disponibles). La resolución del caracter depende de una matriz de 5 x 7 dots o puntos. Por otra parte, es posible programar 16 caracteres según la necesidad y forma requerida por el usuario.

En la siguiente tabla se muestra función de cada pin del LCD.

<i>TERMINALES PARA PANTALLA DE CRISTAL LIQUIDO</i>			
<b>Pin No</b>	<b>Símbolo</b>	<b>Nivel</b>	<b>Descripción</b>
1	Vss		Tierra (Grnd)
2	Vcc		+Vcc (+5V)
3	Vee		Contraste
4	RS	H/L	Register Select
5	R/W	R/W	Read/Write
6	E	H⇌L	Enable o Habilitar
7	DB0	H/L	Bus de datos
8	DB1	H/L	Bus de datos
9	DB2	H/L	Bus de datos
10	DB3	H/L	Bus de datos
11	DB4	H/L	Bus de datos
12	DB5	H/L	Bus de datos
13	DB6	H/L	Bus de datos
14	DB7	H/L	Bus de datos

Este modelo de LCD, es programado por su propio set de instrucciones, es necesario un proceso de inicialización y cada caracter a escribir debe ser enviado según la tabla del apéndice de caracteres del fabricante.

Para su conexión con el microcontrolador se presenta la figura 4.10.

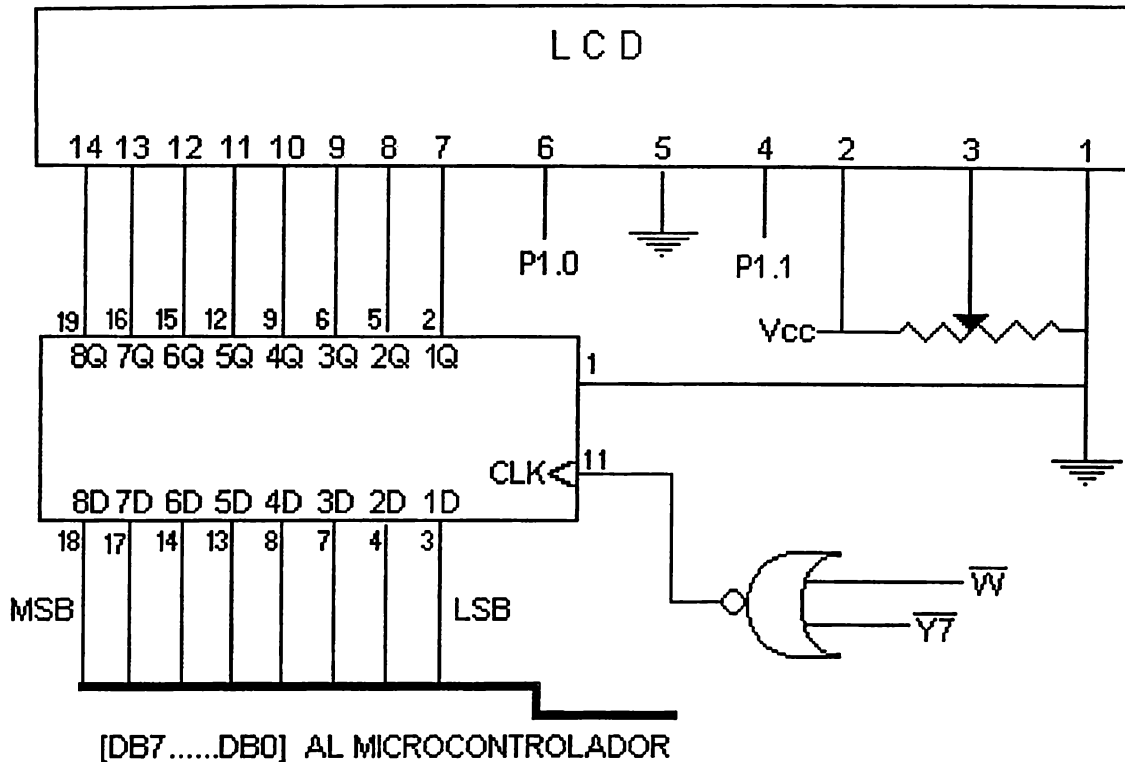


Figura 4.10. Conexión del LCD

Debido a que el LCD es un dispositivo más lento tanto para escritura y lectura (refiriéndonos a la velocidad con que realiza una operación similar el microcontrolador), además, requiere comandos o instrucciones para indicarle el tipo de operación a realizar y el destino de los datos, por lo que es necesario una latch 74LS373 con el fin de retener el dato el tiempo que fuese necesario para que éste (LCD) pueda reconocerlo y procesarlo según la instrucción. La compuerta NOR realiza la función de decodificación según el mapa de memoria. Por otro lado, es necesario las señales de habilitación, por lo que se utilizarán algunos pines del puerto 1, permitiendo ahorrar un conjunto de circuitería extra.

A continuación se presenta las instrucciones de programación y proceso de inicialización del display o pantalla.

Instrucción	R/ S	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
<i>Limpiar Display</i>	0	0	0	0	0	0	0	0	0	1
<i>Cursor al inicio</i>	0	0	0	0	0	0	0	0	1	x
<i>Cursor</i>	0	0	0	0	0	0	0	1	I/D	S
<i>On/Off Display</i>	0	0	0	0	0	0	1	D	C	B
<i>Desplazamiento</i>	0	0	0	0	0	1	S/C	R/L	x	x
<i>Función</i>	0	0	0	0	1	DL	N	F	x	x
<i>Dirección CGRAM</i>	0	0	0	1	Dirección en CGRAM					
<i>Dirección DDRAM</i>	0	0	1	Dirección en DDRAM						
<i>Busy flag</i>	0	1	BF	Address Counter						
<i>Escribir</i>	1	0	DATO							
<i>Leer</i>	1	1	DATO							

S/C=Display shift

R/L=Izquierda(0), derecha(1)

I/D= Decrementa(0), Incrementa(1)

S=Shift

N=Number lines

D=On/Off

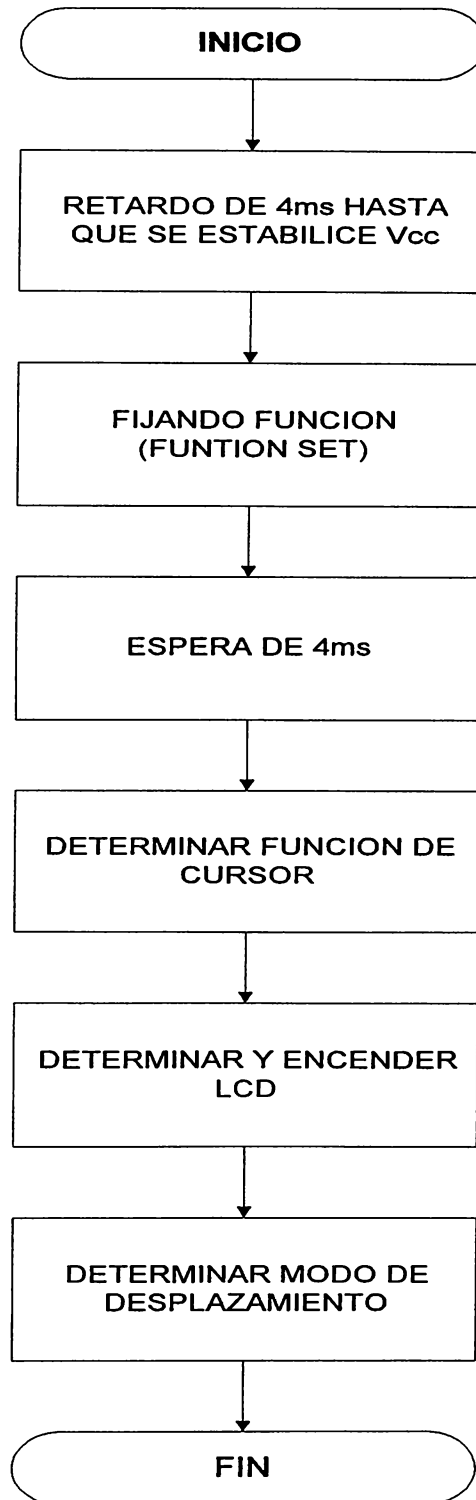
C=On/Off Cursor

B=On/Off Blink

DL=Longitud del dato

F=Font

Flujograma de inicialización del LCD:



## CAPITULO V

### MANUAL

El presente manual tiene como objeto presentar las características y recomendaciones para El sistema basado en un microcontrolador de la familia 8051 de Intel<sup>®</sup> Corporation.

#### *Sistema de Desarrollo DTK51*

El sistema de desarrollo DTK51, es un sistema basado en el microcontrolador Intel 8032, él cual permite su aplicación como:

1. *Sistema mínimo*: él cual puede ser utilizado para aplicaciones diversas: control automático de procesos secuenciales, adquisidor de datos, control de periféricos, manejo de datos, etc.
2. *Tarjeta de desarrollo*: él cual puede ser empleado para probar y evaluar programas, basados en los microcontroladores de la familia 8051 de Intel<sup>®</sup>

Para su operación como sistema mínimo utiliza una EPROM de 8Kx8 (2764), en donde queda almacenado el programa de aplicación, en formato hexadecimal. Para realizar la comunicación serial utiliza se utiliza el conector TB2, él cual está conectado al IC MAX232 de MAXIM<sup>®</sup> integrado que permite la interfaz para los niveles TTL a los niveles RS-232.

Como sistema de desarrollo, se agrega una memoria RAM de la misma dimensión. el sistema utiliza un programa especial que residirá en el EPROM; este programa (MONDTK51.HEX desarrollado únicamente con fines de prueba) es un sistema monitor que permite la comunicación de la tarjeta vía puerto serial con una computadora, el usuario puede entonces (utilizando un paquete de comunicación

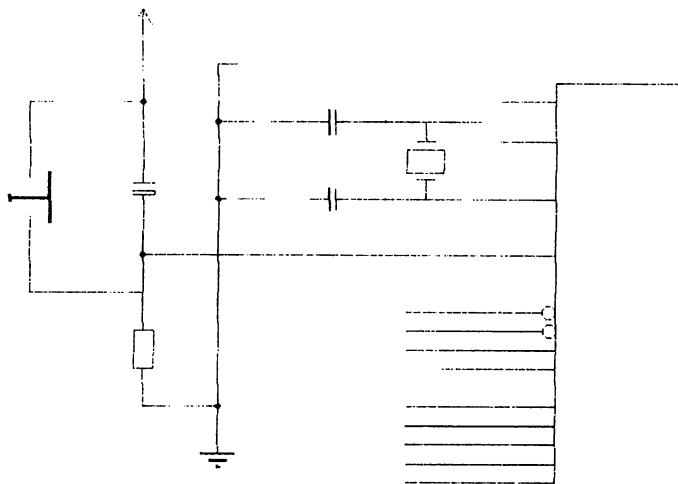
genérico, Laplink, FX, Terminal de Windows, etc. ) cargar los programas que desea ejecutar a la tarjeta. Estos programas quedan residentes en la memoria RAM.

Con esto se evita tener que quemar la EPROM cada vez que se requiera hacer una modificación al programa de aplicación. Cuando el usuario ha determinado que el programa funciona correctamente, el EPROM del monitor puede ser reemplazado por el programa final.

## **5.1 DESCRIPCIÓN DEL CIRCUITO**

- **Hardware:**

El circuito de la tarjeta DTK51 se muestra en el diagrama esquemático anexo. Los nombres que se referirán a partir de ahora a los dispositivos como por ejemplo "MICROCON" son para facilitar su búsqueda al momento de comparar con los anexos el diagrama esquemático. El circuito es alimentado por una fuente de reloj externa que genera una frecuencia de 11.0592 MHz la cual se utiliza para sincronizar el reloj interno, además de estabilizar su valor de tiempo y nivel de voltaje, el cristal (X1) se encuentra conectado a los capacitores C2 y C3 los cuales tienen como función la supresión de los picos de ruido. El reset del sistema es generado por medio del pulsador PUSH1, que se encuentra conectado a una red RC, R1 y C1.



*Figura 5.1 Pulsador de reset del sistema mínimo DTK-51*

El microcontrolador MICROCON se encuentra conectado en la configuración de acceso a memoria de programa externa, es decir para utilizar un  $\mu\text{C}$  8031 ó  $\mu\text{C}$  8032. El microcontrolador utiliza el puerto 0 para la parte baja del bus de direcciones y el puerto 2 para la parte alta. Los datos son colocados también por el puerto 0, por lo que se usa la latch 74LS373 (LATCH), la cual captura la parte baja del bus cuando el pin ALE/P $\bar{\Lambda}$  (pin 11) genera un pulso alto.

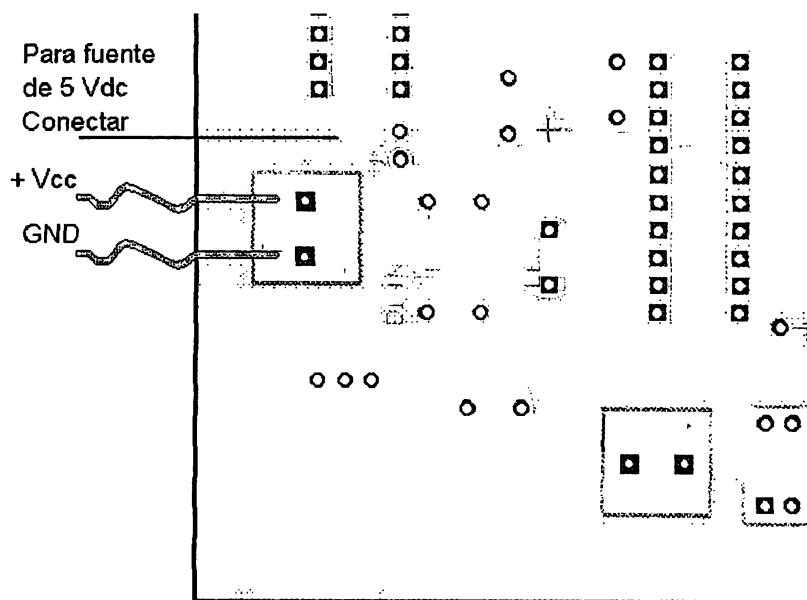
La memoria del programa 2764 (EPROM) es seleccionada mediante el decodificador de direcciones 74LS138 (DEC/DEMU), él cual nos provee ocho líneas de selección en bloques de 8K, las cuales son seleccionadas por las líneas (A13 a A15) del bus de direcciones. La memoria 6264 (RAM) es accesada para su escritura mediante la señal WR y la lectura puede ser mediante la señal PSEN y RD a través de la compuerta AND (AND A). Esto permite que la memoria de datos pueda ser accesada también como memoria de programa cuando se esta operando en la modalidad de sistema de desarrollo. El decodificador DEC/DEMU deja libres 6 señales de selección que podrán ser utilizadas según convenga.

La comunicación a través del puerto serial se realiza con el circuito IC2 (MAX232). Este circuito convierte los pulsos TTL de salida serial en pulsos de +12 y -12 V que son los niveles de señalización requeridos por el estándar RS232C y va a permitir la conectividad con cualquier equipo que tenga este tipo de puerto. Los pulsos recibidos (RXDR) son convertidos en forma inversa a señales TTL para que puedan ser leídos por el microcontrolador.

Muchas de las señales del sistema se encuentran conectadas a un conector tipo Header J1 (CON25X2). La finalidad es que el sistema pueda ser conectado a cualquier otra circuitería de apoyo. Las señales contenidas en el conector son las líneas del puerto de datos, de direcciones, el puerto 1 el cual se encuentra libre y puede ser usado como entradas y salidas, además son conectadas las salidas libres del decodificador así como las señales de control para la comunicación serial.



*Figura 5.2. Header del DTK-51.*



*Figura 5.3. Conexión de energía.*

La fuente de poder está diseñada con un regulador lineal IC1 (7805C) que puede ser alimentado por medio del conector (CON1), con 7.5 a 35 V de directa, produciendo a su salida una señal dc de 5 V regulada. Si se desea (por contar con una fuente de 5 V) se puede eliminar este circuito por medio de la conexión del Jumper (JUMP1).

Los capacitores de desacoplo (C1 a C7), tienen un valor de  $0.1\mu\text{F}$  y sirven para minimizar picos de ruido en la fuente de poder producidos por la conmutación de los circuitos. Es necesario colocar estos capacitores entre Vcc y Tierra de cada uno de los IC's y lo más cerca posible de los mismos. Al colocarlos lejos del circuito en cuestión se reduce su efectividad. Un condensador de desacoplo resuelve dos problemas: actúa como una fuente de alimentación cercana a la carga, aportando la energía necesaria para absorber los picos de corriente a través de una pequeña impedancia de línea y limita el tamaño del bucle con un área menor, con respecto a las componentes de alta frecuencia para impedir la creación de señales parásitas

indeseables de forma que, si existen, se derivan a masa. El condensador debe ser capaz de aportar los picos de corriente requeridos por la carga, la inductancia del bucle debe ser la menor posible.

## 5.2 COMUNICACIÓN SERIAL

Para los cables de comunicación con la computadora se tienen dos configuraciones posibles. Una es utilizando un conector DB25 para conectarlo al puerto RS232C de la PC, sin embargo en la actualidad la mayoría de las computadoras han reemplazado este conector con uno de 9 pines (DB9) por lo que se muestra la configuración para este conector. Es importante tener en cuenta que a menos que se emplee un cable especial la norma RS-232 permite una longitud de cable máxima de 50 pies ( 15 mts aprox. )

Para utilizar un conector DB9 (ver figura 5.4) es necesario, que ciertas terminales están conectadas entre sí; esto se hace con la finalidad de que la computadora vea un dispositivo de comunicación completo. Lo que se está haciendo realmente es engañar a la computadora retroalimentando su propia señalización. El extremo del cable serial se conecta al conector TB3A. Si se desea un conector DB25, a continuación se presentan un descripción de terminales.

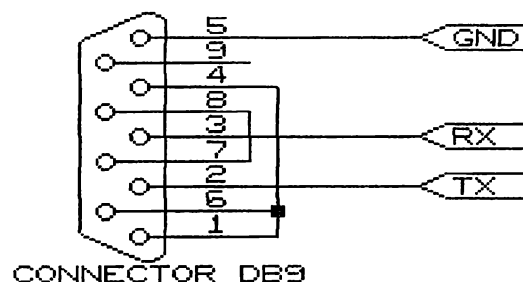


Figura 5.4. Configuración de conector DB9 para RS-232C

En la siguiente tabla se muestra las señales para los conectores DB9 y DB25:

<b>DESCRIPCIÓN DE TERMINALES PARA RS - 232C</b>			
<b>DB25</b>	<b>DB9</b>	<b>Nombre</b>	<b>DESCRIPCIÓN</b>
1		PG	Tierra para protección
2	3	TD	Transmisión de datos
3	2	RD	Recepción de datos.
4	7	RTS	Requisición para envío.
5	8	CTS	Limpia para enviar
6	6	DSR	Dato activo
7	5	SG	Señal de tierra
8	1	CD	Detector de Portadora
9		+ V	Probador de +V.
10		- V	Probador de -V.
11		NC	No Conectado.
12		SCD	Detector de portadora secundario.
13		SCS	Segundo CTS.
14		STD	Segundo transmisor de datos.
15		TC	Reloj Transmisor
16		SRD	Segundo receptor de datos.
17		RC	Reloj Receptor.
18		NC	No Conectado.
19		SRS	Segundo RTS
20	4	DTR	Indicador de Dato Terminal.
21		SQD	Señal Detectora de Calidad.
22	9	RI	Indicador de Ring.
23		DRS	Selector de velocidad de datos.
24		XTC	Reloj Externo.
25		NC	No Conectado.

### 5.3 MAPA DE MEMORIA

Se tiene un bloque de 8K para la memoria de programa, 8K para la memoria de datos que también puede ser accesada como memoria de programa y otros 6 bloques de 8K que el usuario puede utilizar mediante las líneas de selección extra del decodificador.

Cada bloque esta formado por 13 líneas (  $2^{13} = 8192 = 8Kb$  ), las líneas ( A0 a A12 ) contienen la dirección de la localidad del dispositivo seleccionado. Por medio de las líneas A13, A14 y A15 se selecciona el dispositivo o periférico que se mapea, por ejemplo si se necesita otra memoria de datos se puede mapear en el bloque A000h - BFFFh, en la siguiente tabla se muestra mapa de memoria para el sistema mínimo o la tarjeta de desarrollo.

<i>MAPEO DE MEMORIA PARA SISTEMA BASADO EN <math>\mu</math>C 8031 ,8032</i>					
<b>A15</b>	<b>A14</b>	<b>A13</b>	<b>BLOQUE</b>	<b>SALIDA</b>	<b>DESCRIPCIÓN</b>
1	1	1	E000h - FFFFh	Y7	Libre
1	1	0	C000h - DFFFh	Y6	Libre
1	0	1	A000h - BFFFh	Y5	Libre
1	0	0	8000h - 9999h	Y4	Libre
0	1	1	6000h - 7999h	Y3	Libre
0	1	0	4000h - 5999h	Y2	Libre
0	0	1	2000h - 3999h	Y1	Accesa memoria de datos ( RAM).
0	0	0	0000h - 1999h	Y0	Accesa memoria de programa

En base a lo anterior se puede notar que las direcciones para la memoria de datos están contenidas por el rango 2000h a 3999h y que si se precisa escribir un dato en la localidad 4052h el sistema perderá los datos.

Para seleccionar los periféricos se emplea un decodificador 74LS138 (DEC/DEMU), el cual tiene tres entradas y ocho salidas (Y0 - Y7), las cuales son las encargadas de habilitar los dispositivos mediante niveles bajo activos y corresponden a los pines (7,9..15) del decodificador. En el sistema las líneas Y2 a Y7 se encuentran conectadas al Conector tipo Header (CON25X2).

### **Ejemplo para el uso del mapeo de memoria.**

El uso de mapeo de periféricos nos permite tener libre el uso del puerto 1, el cual puede ser empleado para propósitos generales, por ejemplo, si es preciso utilizar un teclado, podemos accederlo por medio de la dirección C000h, ya que sólo necesitamos una línea, para este caso la línea a emplear es Y6 (pin 43 del Conector Header J1).

Si por el contrario queremos acceder a 10 dispositivos, mediante 12 líneas seleccionamos el rango C000h y por medio del conector Header (CON25X2) conectamos las líneas A0...A12 ( pines pares entre 2 y 32 ), a cada dispositivo, ( para proteger al sistema se recomienda el uso de manejadores (buffers), los cuales pueden ser habilitados por la línea Y6 ), no olvidar que los dispositivos se mantendrán encendidos mientras la dirección permanezca en alto a menos que los manejadores posean una forma de almacenar y mantener el último dato, si se necesitan más dispositivos será necesario el empleo de otro demultiplexor.

## 5.4 RECOMENDACIONES PARA LA TARJETA

### 1. HARDWARE

- Utilizar una buffers para las entradas o salidas, para evitar que el  $\mu\text{C}$  sea dañado al ocurrir un cortocircuito, puede utilizar Chips o transistores.
- Si la fuente del sistema alimenta alguna etapa adicionada, utilice una fuente que genere un 20% más del consumo necesitado, no olvidar una disipación adecuada.
- El uso de memorias RAM CMOS, disminuye el consumo de potencia, pero son más delicadas que las memorias RAM TTL, por lo que si la carga del sistema no excede el 80% de la máxima carga generada por el regulador de voltaje, es preferible utilizar RAM's TTL, ( Así mismo no olvidar la velocidad de respuesta que necesita el microcontrolador ).

### 2. COMUNICACIÓN E INTERFERENCIAS

- Colocar una etapa que permita la comunicación para el estándar RS-485, de esta manera se puede superar la limitante de longitud que posee el estándar RS - 232C.

Para esto se recomienda el uso del IC 74LS176B, el cual permitirá tener una comunicación con y sin Handshake.

- Utilizar resina o pintura para aplicaciones electrónicas, para aislar las caras, de la tarjeta, evitando de esta manera un corto, e interferencias por los bucles que forman las pistas.
- Utilizar un chasis metálico que cubra el Cristal, para evitar interferencias.
- No ubicar el sistema cerca de sistemas o líneas de alto voltaje, para evitar interferencia o ruido que pueda ser inducido en el bus de datos o de direcciones.

### 3. PRESENTACIÓN Y CAJA

- Para la creación de un caja que proteja al sistema del medio que la rodea, se recomienda el uso de una caja de plástico resistente a sustancias químicas, el

uso de plástico en lugar de aluminio o metal se debe a la no conducción de cargas eléctricas, así mismo hace liviano el sistema, ventaja si se utiliza como tarjeta de desarrollo o si es preciso montarlo en cualquier estructura, si se utiliza como tarjeta de aplicación.

- Utilizar un ventilador tipo computadora si el sistema es utilizado en un ambiente con un temperatura superior a los 40°C, ( será conveniente para esto un caja que permita la circulación de aire). Por lo que además será necesario considerar un espacio para la entrada de aire. ( puede utilizarse una malla metálica que impida al acceso a objetos pequeños así como a insectos.
- Conectar una regleta plástica a la salida del conector Header de 50 pines; esta facilitará la conexión con los dispositivos externos a la tarjeta.
- Se recomienda colocar en el panel frontal de la caja, el interruptor de encendido, indicador de encendido (LED), bornes para las líneas de alimentación, así como un interruptor que conmute el uso del regulador ( Jump 1).
- Las dimensiones recomendadas para la caja son: 6" X 6" x 4" (Nota la altura puede ser modificada dependiendo del tamaño del ventilador.

## 5.5 TABLA DE CARACTERÍSTICAS

<b>Especificaciones específicas:</b>		
Descripción:	Típico	Rango
Voltaje de Alimentación DC ( IC REG)	5.0	4.8 ~ 5.2
Rango de Voltaje de Alimentación externa.	9.0	7.5 ~ 35 V
Temperatura de trabajo	40°	0° ~ 50° C
Temperatura de almacenamiento	32°	-20° ~ 75° C
Humedad (sin condensación)	~	10% ~ 80%
Aterrizado	< 100 $\Omega$	~
Peso ( sólo tarjeta con dispositivos)	800 grs.	~
<b>Características:</b>		
Método de control	programa almacenado en ROM, programa cargado en RAM, mediante PC.	
Lenguaje de programación	Ensamblador para $\mu$ Controlador 8051	
Set de instrucciones:	256 instrucciones	
Tipo de instrucciones	Aritméticas, Booleanas, Datos, Salto.	
Capacidad máximo para programa	64 KB, 8192 Bytes, 65,536 Bits,	
Cantidad de RAM interna:	128 KB.	
Mapeo de memoria para usuario	6 Bloques de 8 KBytes cada uno.	
Temporizadores/ Contadores	3 (8032), dependiendo del modo de operación puede aumentar.	
Cuenta máxima.	FFFF <sub>H</sub> , 65,536 <sub>10</sub> , 11111111 11111111 <sub>2</sub>	
Entradas / Salidas ( I/O)	8 Bits direccionables ( Puerto 1)	
Tipo de comunicación serial	Estándar RS - 232C	
Baud rate máx.	19200bps (Para RS-232C)	
Distancia máx.	15 mts. ( 50 pies ).	
Vibración máx.	De 10 a 57 Hz.:0.075 mm de amplitud	
Fan out para puerto 1	4 compuertas TTL STD.	

## 5.6 INSTALACIÓN Y MANTENIMIENTO DE TARJETA DTK-51

La tarjeta de desarrollo DTK-51, con un adecuado gabinete, puede trabajar en ambientes toscos, sin que ello afecte la funcionalidad del mismo, pero la instalación en un sitio adecuado permitirá una vida útil mayor:

Condiciones para el lugar de instalación:

1. Un lugar donde exista una adecuada iluminación; solar de día y artificial de noche.
2. Un lugar en donde la temperatura ambiente no exceda el rango de 0° a 55° C.
3. Un lugar en donde la humedad relativa no exceda el rango de 10% a 90% RH
4. Un lugar con poca condensación.
5. Un lugar que evite golpes directos o vibración, así como la exposición a gases corrosivos, gases inflamables y sales.

Si la tarjeta se emplea como una tarjeta de aplicación, el mantenimiento se limita a la limpieza de los contactos de la bornera o regleta plástica, y las entradas de aire, y a la medición de los niveles de voltajes en las salidas y entradas de los periféricos.

## 5.7 AJUSTE Y CALIBRACIONES

El sistema no necesita calibraciones, ya que las modificaciones en los procesos se realizan por medio de software, es decir por medio del programa de prueba para la tarjeta de desarrollo o el programa monitor para el sistema mínimo, así mismo no se necesita regulaciones para la alimentación de los circuitos integrados, debido a que el regulador de voltaje se ajusta automáticamente ante los cambios. La selección de velocidad, cantidad de bits, etc. También se configura por medio de software.

## 5.8 MANUAL PARA SISTEMA MEZCLADOR BASADO EN $\mu$ C MCS-51

El presente manual tiene como objeto, presentar los procedimientos y características para realizar la automatización de un proceso de fabricación de ceras y desinfectantes para pisos. Esta aplicación del microcontrolador, demuestra su aplicación en procesos industriales de tipo secuencial. El proceso de fabricación se realiza de la siguiente manera:

1. Se vierte la cantidad de la primera substancia.
2. Se vierte la cantidad de la segunda substancia.
3. Se vierte la cantidad de la tercera substancia.
4. Se vierte la cantidad de la cuarta substancia.
5. Se mezclan, todas las cantidades de los ingredientes, según la fórmula o receta.
6. Se envasa el producto final.

El equipo de la planta está compuesto por:

1. Cuatro tanques o depósitos. Un depósito para cada uno de los ingredientes.
2. En cada tanque se encuentra localizada una electroválvula.
3. Una bomba, que realiza la tarea de transportar los ingredientes de los tanques de almacenamiento al tanque de mezclado.
4. Un medidor de flujo, él cual nos permite determinar la cantidad de líquido por unidad de tiempo (galones o litros por minuto).
5. El tanque de mezclado: él cual es un depósito en donde los ingredientes son almacenados para ser mezcladas por medio de las aspas de un agitador.
6. Una electroválvula para vaciar el contenido de tanque de mezclado hacia los envases.

El sistema controlador se divide en dos etapas:

1. Etapa de Potencia.

- Etapa en la cual se encuentra los dispositivos eléctricos, de mayor voltaje, motores, bombas, sensores, etc. es la parte del sistema que se encarga de realizar el proceso, se denominó potencia para identificarla con las características eléctricas que posee.
2. Etapa de control.
- Es la etapa que se encarga de monitorear y dirigir las etapas del proceso. Se encuentra formada por tres tarjetas electrónicas:
    - Una tarjeta electrónica para el sistema mínimo: para mayor información (ver sección anterior, "MANUAL DE SISTEMA MINIMO"
    - Una tarjeta electrónica para periféricos. ( ver capítulo 4).
    - Una tarjeta electrónica para interface. ( ver capítulo 4).

Para llevar a cabo el control del proceso, el microcontrolador debe realizar las siguientes tareas:

1. Abrir la electroválvula para el primer depósito y encender la bomba.
2. Cerrar la electroválvula y apagar la bomba cuando la cantidad del primer ingrediente se ha vertido en el tanque de mezclado, según la fórmula o receta del producto a fabricar.
3. Abrir la electroválvula para el segundo depósito y encender la bomba.
4. Cerrar la electroválvula y apagar la bomba cuando la cantidad del segundo ingrediente se a vertido en el tanque de mezclado.
5. Abrir la electroválvula para el tercer depósito y encender la bomba.
6. Cerrar la electroválvula y apagar la bomba cuando la cantidad del tercer ingrediente se a vertido en el tanque de mezclado.
7. Abrir la electroválvula para el cuarto depósito y encender la bomba.
8. Cerrar la electroválvula y apagar la bomba cuando la cantidad del cuarto ingrediente se ha vertido en el tanque de mezclado.

9. Encender el motor del mezclador para el iniciar mezclado de los ingredientes y obtener así una mezcla homogénea.
10. Esperar a vaciar tanque de mezclado para iniciar nuevo ciclo de fabricación.

En la siguiente figura se muestra el diagrama de la planta:

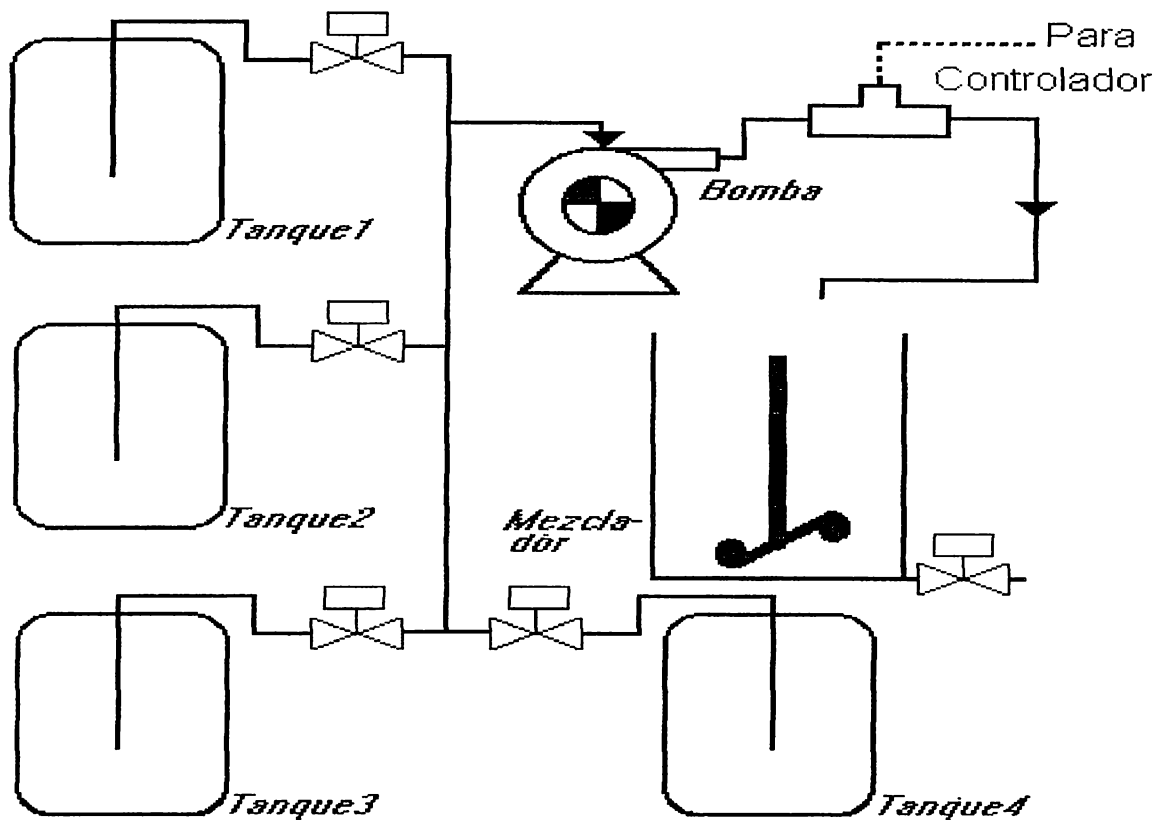


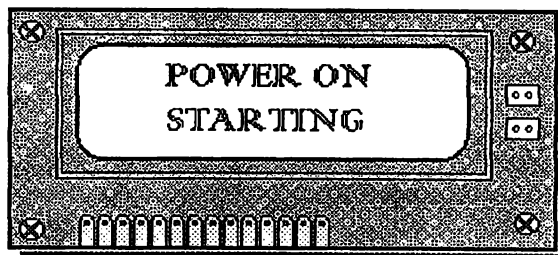
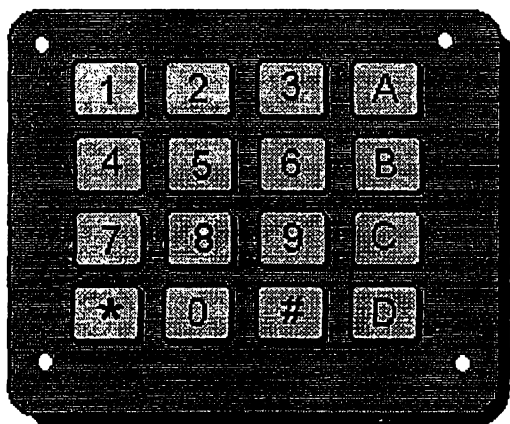
Figura 5.5. Diagrama de la planta

## 5.9 PASOS PARA PROGRAMACIÓN







El proceso puede realizarse por medio de dos métodos:

- Por la llamada de una receta previamente almacenada,
- Por la introducción de las cantidades de cada ingrediente a procesar.

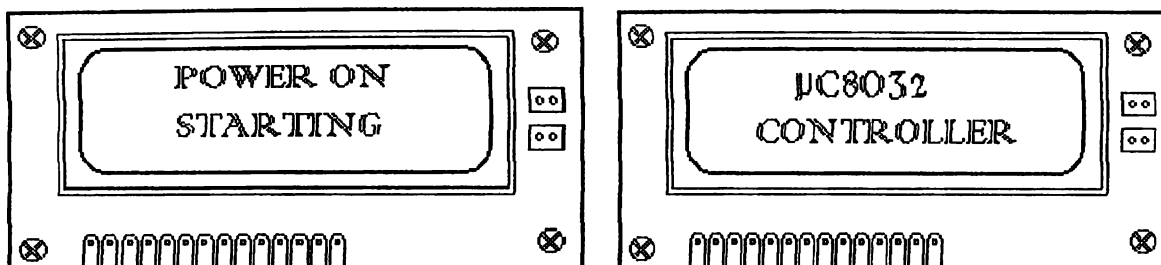
Para indicar la receta o las cantidades de los ingredientes se utiliza el teclado hexadecimal para introducir los datos y en la pantalla de cristal líquido se visualizan los mensajes. A continuación se muestra la forma real del teclado como de la pantalla de cristal líquido.



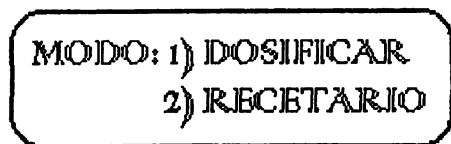
#### FUNCION DE TECLAS:

-  **A** Aceptar: Para validar un dato o cantidad.
-  **B** Retornar: Para indicar el regreso a una pantalla anterior.
-  **C** Continuar: Para continuar o indicar a una pantalla de proceso el proseguir.
-  **D** Denegar: Para dar como invalido o no correcto un valor o dato
-  **\*** .: Para indicar que la cantidad que se introducirá tendrá como unidades galones, o para introducir punto decimal cuando se introduce cantidades.
-  **#** #: Para indicar que la cantidad que se introducirá tendrá como unidades litros.

Al iniciar el proceso aparece las siguientes pantallas:



Luego el sistema presenta una pantalla en donde se pregunta el modo de operación, a esta pantalla se ha denominado pantalla de Modo.



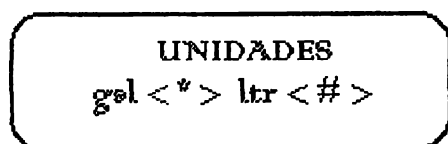
Dosificar: para indicar las cantidades manualmente.

Recetario: para llamar de memoria las cantidades para una determinada mezcla.

### 5.9.1 PROCESO PARA DOSIFICACIÓN DE INGREDIENTES

1.1 – Para realizar el proceso mediante dosificación se presiona la tecla “1” en la pantalla de Modo.

1.2 - Una pantalla en la cual se definen las unidades de las cantidades aparece. Cualquier otra tecla que no se “\*” y “#” no producirá algún efecto.



Al presionar la tecla “#” se seleccionan litros,

Al presionar la tecla “\*” se seleccionan galones.

1.3 - Definidas las unidades aparece automáticamente otra pantalla en la cual se introduce la cantidad para el ingrediente solicitado.

<b>INGREDIENTE # 1</b> <b>CANTIDAD _____</b>
---

El número de dígitos para las cantidades es de seis incluyendo el punto decimal.

Aparece una pantalla por ingrediente. ( Nota la cantidad máxima de ingredientes a mezclar es de cuatro).

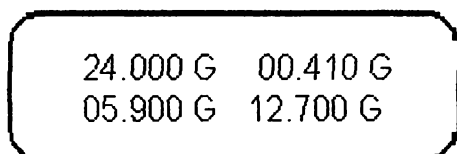
Si el dato es menor de los seis dígitos se presiona la tecla "A", si se desea introducir de nuevo el mismo ingrediente presionar la tecla "D" Media vez se presiona la tecla "A", el dato es aceptado y procede con el siguiente ingrediente.

1.4 - La cantidad que se introduce no indica si es en galones o en litros, por lo que sí la cantidad es en litros, el dato debe ser introducido con tres enteros y dos decimales. Si es galones, la cantidad debe ser introducida con dos enteros y tres decimales.

En la siguiente tabla se muestra la información necesaria para la correcta introducción de los datos.

	<b>Litros ( Lt )</b>	<b>Galones ( G )</b>
Valor Máximo	189.66	50.175
Cantidades Válidas	100, 100.25, 22.125, 0.90, .90, 00.50	30, 30.00, 12.525, 0.125, .125, 00.55
Cantidades No Válidas	192, 12..35, 12.#3, .256.2 mayor que 189.66	51, 1..12, 15.#2, *12.2 mayor que 50.175
Carácter asignado	*	#

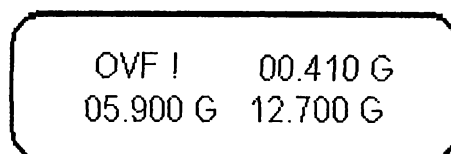
1.5 - Si los datos fueron introducidos correctamente aparece una pantalla en la cual los cuatro ingredientes son mostrados, durante seis segundos, luego aparece otra pantalla para confirmar el proceso. Pero si los datos fueron introducidos erróneamente o con cantidades no válidas aparece una pantalla indicando que se generó un sobreflujo en un ingrediente determinado, y el programa retorna a la pantalla de modo.



```

24.000 G  00.410 G
05.900 G  12.700 G
  
```

Pantalla para datos correctos

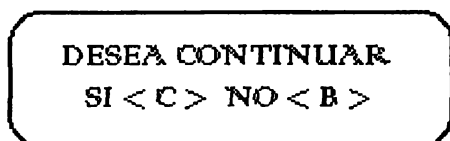


```

OVF !    00.410 G
05.900 G 12.700 G
  
```

Pantalla para datos no válidos

1.6 - Si los datos son cantidades válidas (ver tabla en 1.4) aparece una pantalla para continuar o para regresar al paso 1.1.-, y si los datos no son válidos automáticamente regresa a la pantalla del paso 1.2

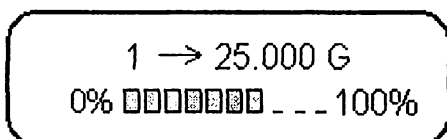


```

DESEA CONTINUAR
SI < C > NO < B >
  
```

Si se presiona la tecla " C " son tomados por válidos los datos y continua con el proceso, pero si se presiona "B", se retorna a la pantalla de modo.

1.7 - Si se presionó la tecla C para continuar con el proceso, aparecerá una pantalla mediante la cual se monitorea la salida de cada substancia de su respectivo depósito.



```

1 -> 25.000 G
0% ██████████ --- 100%
  
```

Por ejemplo si el primer ingrediente tiene una cantidad de 25 Galones, aparece la cantidad con sus unidades en la parte superior y en la parte inferior de la pantalla, se muestra su progreso.

Cuando se ha terminado de verter la substancia No.1 se procede automáticamente a verter la substancia No. 2 mostrando una pantalla similar en donde el único cambio es el número del ingrediente así como su cantidad.

1.8 - Cuando los cuatro ingredientes han sido vertidos en el tanque del mezclador, automáticamente se enciende el motor de este, cuando esto ocurre se presenta una pantalla en la cual se muestra el progreso del mezclado.

01 MIX ON 10  
 □□□□□□□□ - - -

El tiempo que estará encendido el motor del mezclador puede ser cambiado en el programa monitor, para este caso el tiempo es de 12 segundos.

9.- Si el proceso se ha realizado sin algún problema se presenta la pantalla en la cual se indica que el proceso ha finalizado y que el contenido del mezclador puede ser vaciado.

**MEZCLADORA LLENA  
 PUEDE ENVASAR**

Cuando el contenido del mezclador ha sido vaciado, el sensor que se encuentra en el tanque indica a la tarjeta controladora que el proceso ha terminado, retornando a la pantalla de "Modo"

## 5.9.2 PROCESO POR RECETA

2.1 Si se presiona la tecla "2" en la pantalla de modo, aparece otra pantalla preguntando por el tipo de producto a mezclar. ( El Programa tiene cuatro recetas; los cuatro productos son: "Brillantex", "Lejía", "Cloro" y "Cera" ).

1) BRILLANTEX  
2) LEJIA

3) CLORO  
4) CERA

En la primera pantalla que se presenta, se pregunta si se desea procesar el Brillantex o la Lejía, si desea escoger uno de ellos presiona el número correspondiente, pero si desea procesar otro producto presione la tecla "C"; entonces aparecerá otra pantalla mostrando las opciones para el Cloro o cera; para seleccionar cualquiera

de ellas se presiona el número de la izquierda. Si desea regresar a la pantalla anterior presione la tecla "B"

2.2 Si se escogió uno de los cuatro productos, inmediatamente aparece una pantalla mostrando las cantidades a mezclar, se preguntará si se desea continuar

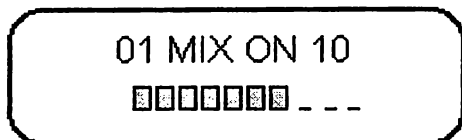
→ 25.000 G  
0% ██████████ - - - 100%

con el proceso tal como despues de dosificar, luego se muestra el progreso del vaciado de los ingredientes. Por ejemplo si el primer ingrediente tiene una cantidad de 25 Galones, aparece la

cantidad con sus unidades en la parte superior y en la parte inferior de la pantalla, se muestra su progreso.

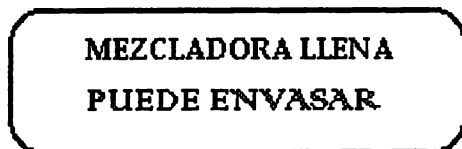
Quando se ha terminado de verter el primer ingrediente se procede automáticamente a verter el segundo ingrediente, mostrando una pantalla similar en donde el único cambio es la cantidad de este.

2.3 - Cuando los cuatro ingredientes han sido vertidos en el tanque del mezclador, Automáticamente se enciende el motor del agitador. Cuando esto ocurre se presenta una pantalla en la cual se muestra el progreso del mezclado.



El tiempo que estará encendido el motor del agitador puede ser cambiado en el programa monitor, para este caso el tiempo es de 12 segundos.

2.4 - Si el proceso se ha realizado satisfactoriamente, se presenta la pantalla en la cual se indica que el proceso ha finalizado y que el contenido del mezclador puede ser vaciado.

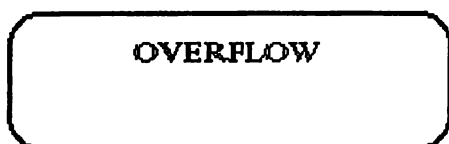


Cuando el contenido del mezclador ha sido vaciado, el sensor que se encuentra en el tanque de mezclado indica a la tarjeta controladora que el proceso ha terminado, retornando a la pantalla de "Modo", para continuar con un nuevo ciclo de producción.

### 5.9.3 MENSAJES DE ERROR Y STOP

- **PANTALLA DE SOBREFLUJO (OVERFLOW)**

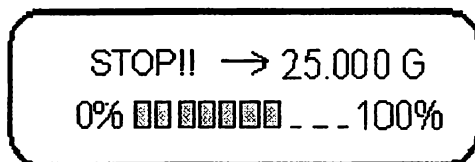
Aparece la pantalla de overflow cuando uno de los ingredientes tiene una cantidad mayor que la cantidad máxima del tanque de mezclado, (para esta aplicación el mezclador es de 50 Galones). También aparece esta pantalla cuando la suma de las cantidades de los ingredientes es mayor que la capacidad del tanque de mezclado. Esta pantalla aparece por un lapso de 6 segundos, luego se retorna al menú de modo.



Si se cambia el depósito, debe cambiarse el valor de su capacidad en el programa monitor, de la tarjeta controladora.

- **PANTALLA DE STOP**

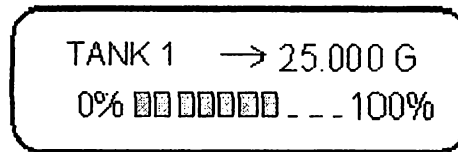
El objeto de esta pantalla es mostrar el progreso del mezclado o vertido de los ingredientes, cuando por algún motivo el operario necesite detener el proceso.



Para salir de la pantalla de stop es necesario presionar cualquier tecla del teclado hexadecimal.

- **PANTALLA DE AVISO PARA INGREDIENTE FALTANTE**

Esta pantalla aparece cuando la cantidad del ingrediente almacenado en su respectivo tanque, ha llegado al nivel mínimo. El nivel mínimo está determinado por la posición del sensor de nivel que se encuentra en cada una de los tanques de almacenamiento.



## 5.10 ESPECIFICACIONES

### ESPECIFICACIONES DEL CONTROLADOR

<b>ENTRADAS DISCRETAS</b>	Permite la conexión de terminales de accionamiento provenientes de un sensor con voltaje de 24 Vdc.
<b>FRECUENCIA DE ENTRADA</b>	Compatible con sensores de turbina o rueda, con un máximo de 200kHz.
<b>PULSO</b>	Compatible con TTL (0↔5 Vdc)
<b>MÍNIMA DURACIÓN DEL PULSO</b>	5 $\mu$ s
<b>SALIDAS DISCRETAS</b>	Ocho salidas de 2A, 220Vac,
<b>REQUERIMIENTOS DE POTENCIA</b>	<ul style="list-style-type: none"> <li>⊖ VIN: 110Vac <math>\pm</math> 10%</li> <li>⊖ FREQUENCY DE FUENTE: 60Hz</li> <li>⊖ POWER CONSUMPTION: 40VA</li> </ul>
<b>TEMPERATURA AMBIENTE</b>	5 a 60 °C (41 a 140 °F)
<b>LÍMITES DE HUMEDAD RELATIVA</b>	5 a 95%
<b>CANTIDAD DE TANQUES MÁXIMO</b>	4 TANQUES
<b>CAPACIDAD DE FUSIBLE</b>	2 A
<b>RETARDO DE REACCIÓN</b>	1 ms
<b>PRECISIÓN</b>	0.1%

\* Es necesario aclarar que muchas de las características inherentes del dispositivo dadas por el fabricante, no son aplicables a todas las marcas o reemplazo, por lo que se han realizado estimaciones y aproximaciones para las peores condiciones con el fin de tener un dato nada alejado de la realidad.

## **BIBLIOGRAFIA**

1. GEORGE LOVADEY, Manual de electrónica para ingenieros, Editorial Trillas, II Edición, México 1992.
2. TIMOTHY MALONEY, Electrónica Industrial, Prentice Hall, I Edición, México 1983.
3. WILLIAM D. COOPER, Instrumentación electrónica moderna, II Edición, México 1991.
4. EDUARD V. KIRK, Introducción a la Ingeniería y al diseño en la Ingeniería, Editorial Limusa, II Edición, México 1988.
5. SIEMENS, Automatización con S5-115U, Siemens publications, I Edición, Alemania 1987
6. SCOTT MACKENZIE, El microcontrolador 8051, Maxwell-Macmillan publications, I Edición, U. S. A. 1992.
7. CHARLES SLIPPL, The new Webster's computer terms, Editorial Lexicon, I Edición, U. S. A. 1990.
8. REMINGTON, Farmacía, Editorial Médica Panamericana, XVII Edición, Argentina 1987

9. IEEE MAGAZINES, Spectrum Octubre de 1996 "Microcontrollers, workhorses of the electronic era", IEEE Editoriales Inc., Unica Edición, U. S. A. 1996.
10. OMEGA, Flow Meters, Omega Editoriales, U .S. A. 1994.
11. ANTONIO CREUS, Instrumentación Industrial, Alfaomega Marcombo, IV Edición, México 1992.
12. JOHN F WAKERLY, Diseño digital Principios y prácticas, Prentice-Hall Hispano, I Edición, México 1992.
13. ALLEN-BRADLEY, Micromotor, Allen-Bradley / Rockwell publications, Unica Edición, U. S. A. 1995.
14. ALBERTO MAYOL BADIA, Automatas Programables, Alfaomega Marcombo, II Edición, España 1992.
15. TEXAS INSTRUMENT, TTL Logic Data Book, U. S. A. 1994.
16. INTEL CORPORATION, 8-Bit Embedded Controllers, 8031/8051/8031AH/8051AH/8032AH/8052AH/8751AH/8751H-8, Intel MCS -51@ Control-Oriented Microcomputers, U. S. A. 1991.
17. OPTREX CORPORATION, Optrex Data Book, Japón 1992.
18. NATIONAL SEMICONDUCTOR, Interfacing/IC Regulators, U. S. A. 1991.

19. AVOSET SYSTEM INC., XASM51: 8051 Cross Assembler, U. S. A. 1990.
20. AVOSET SYSTEM INC, AVSIM51: 8051 Family simulator, U. S. A. 1992.
21. A. D. SOFTWARE, 8051 Macro Assembler, U. S. A. 1985.
22. BALCELLS/DAURA/ESPARZA/PALLAS, Interferencias Electromagnéticas en Sistemas Electrónicos, Alfaomega Marcombo, Unica Edición, España 1992.

#### INTERACTIVE MART

- DIGI - KEY

e-mail: [sales@digiskey.com](mailto:sales@digiskey.com)

<http://www.digiskey.com>

fax: 218-681-3380

- JAMECO

e-mail: [info@jameco.com](mailto:info@jameco.com)

<http://www.jameco.com>

fax: 1-800-237-6948

## RECOMENDACIONES

1. Se recomienda ubicar el sistema en un lugar en donde la temperatura ambiente no exceda el rango de 10°C a 50°C.
2. Se recomienda colocar las tarjetas electrónicas dentro de un depósito que permita una adecuada ventilación.
3. El uso de un teclado revestido de una membrana, evitará el mal funcionamiento de este, provocado por residuos de polvo, sustancias, etc.
4. Para mejorar la visibilidad de la pantalla de cristal líquido, puede agregarse un led en los dos puntos de energía que se encuentran en la parte izquierda de la misma. ( para esto ver hojas del fabricante en anexos ).
5. Para un adecuado nivel de polarización y consumo de energía puede utilizarse una fuente conmutada de 5V con una corriente de 5 amperios la cual se conectará a la entradas de cada tarjeta electrónica.
6. Para mejorar la notificación de un error o un evento no deseado, puede colocarse una bocina en cualquier pin del puerto 1; de acuerdo a su incorporación al programa principal.
- 7 Para mejorar la comunicación serial en cuanto a la limitante de longitud, del standart RS-232 puede utilizar el estándar RS-485, para el cual, es necesario modificar el circuito de acuerdo al diagrama mostrado en los anexos.

8. Para reducir el consumo de energía en el microcontrolador, puede utilizar un  $\mu\text{C}$  80C52, o  $\mu\text{C}$  80C32. Además, estos permiten funcionar con valores de voltaje bajos, lo que puede ser explotado en situaciones de corte de energía eléctrica, para ello ver set de instrucciones, para poder programar esta función.
9. Si se desea incluir la función de monitoreo en tiempo real, se recomienda que el protocolo de comunicación serial utilizado por el programa sea el Xmodem, con los siguientes parámetros:  
Bit de stop, sin paridad, 9600 bps, configuración de control: ninguna, trama de 8 bits.
10. Para el uso de electroválvulas, motores, etc. Se recomienda que los relevadores soporten un 20% más de corriente que la que se demande.
11. El mantenimiento se limitará a remover el polvo, grasa, o cualquier sustancia que se encuentra en los dispositivos, internos ( integrados, resistores, etc. ) y externos (teclado, pantalla de cristal líquido).
12. Si desea programar el tiempo de mezclado, puede realizar en el programa principal las modificaciones pertinentes.
13. Se recomienda para cualquier modificación en el programa, utilizar un simulador de AVSM™ antes de intentar implementarlo..
14. Para disminuir los efectos magnéticos de otros dispositivos en el sistema controlador, es saludable utilizar lacas o pinturas aislantes, esto protegerá mejor contra la humedad y efectos magnéticos parásitos.

15. Agregar una Ram no volátil, permitirá desarrollar un menú de editor de recetas como sistema de respaldo (backup) para cualquier eventualidad. Permitirá llevar control de cantidad de batches producidos y otros valores que puedan arrojar estrategias futuras para el empresario que utilice el controlador.
  
16. Es importante recalcar que para el desarrollo de impresos se consideren factores como temperatura, humedad, vibración, capacitancias e inductancias parásitas, alta frecuencia, voltaje, etc. De lo contrario se predice poca confiabilidad a éste o cualquier sistema.

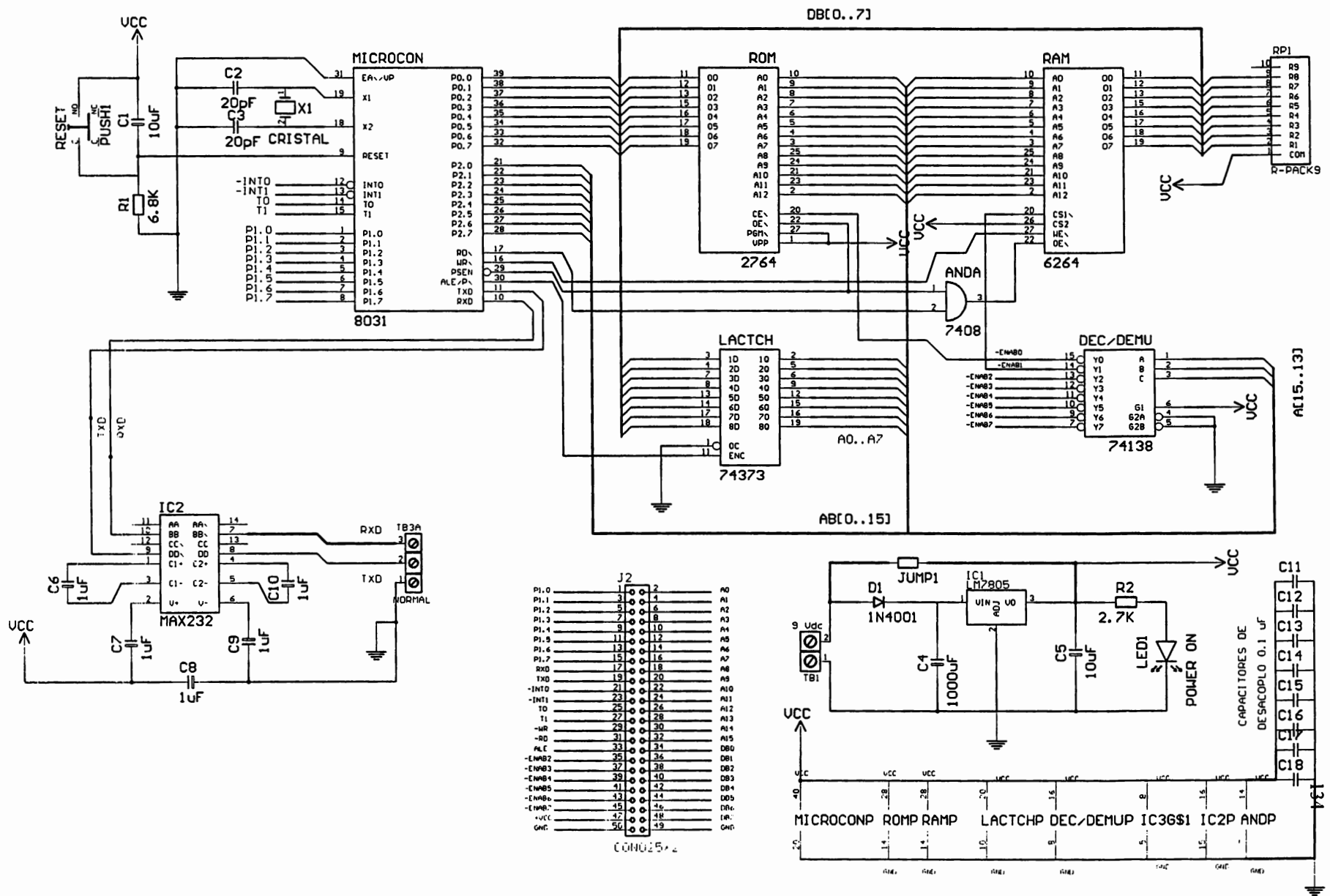
**APÉNDICE A**

### CONSTITUCIÓN DEL SISTEMA MÍNIMO

Un sistema constituido por la familia MCS-51 resulta ser un circuito atractivo, flexible, útil y fácil si se compara con cualquier otro circuito que implemente un microprocesador.

Para su constitución es necesario los siguientes elementos:

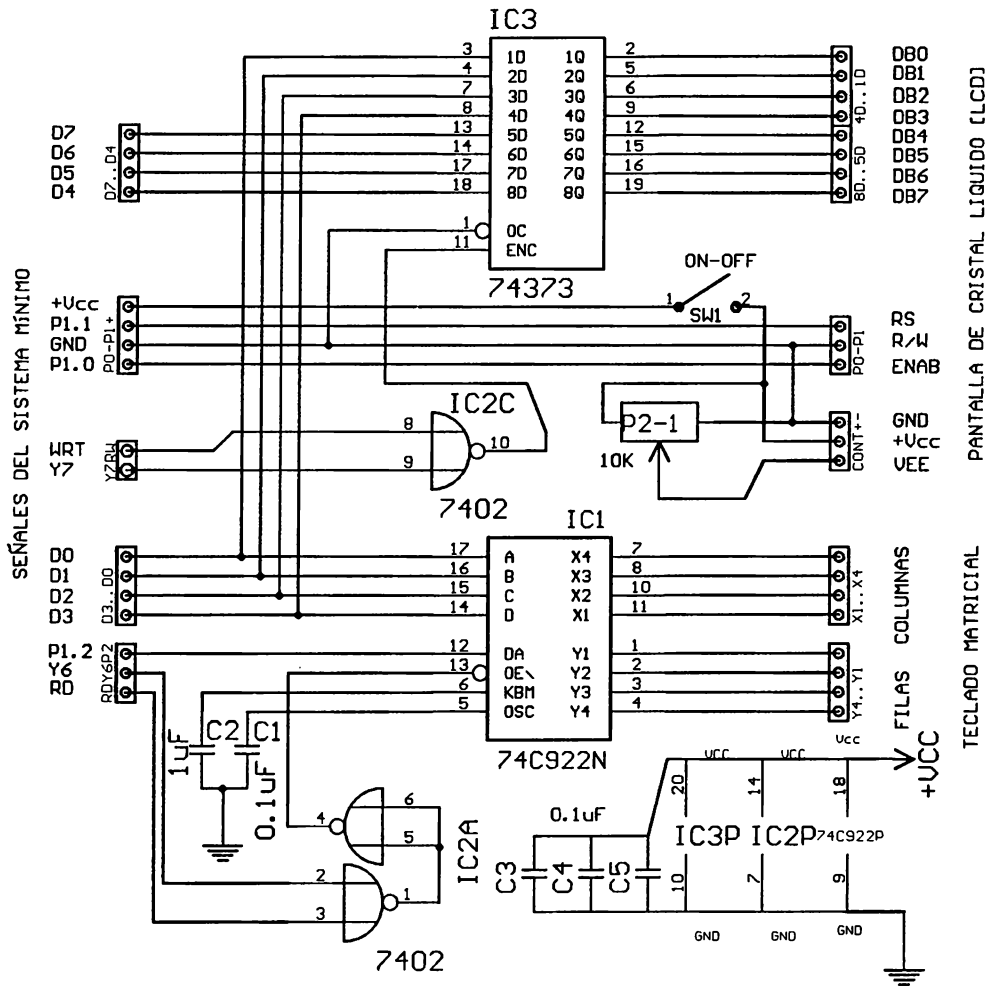
ÍTEM	CANTIDAD	PARTE
1	7	Condensador de cerámica de 0.1 $\mu$ F
2	5	Condensador electrolítico de 1 $\mu$ F
3	2	Condensadores electrolítico de 10 $\mu$ F
4	1	Condensadores electrolítico de 1000 $\mu$ F
5	2	Condensador de cerámica de 20 pF
6	1	Resistencia de 6.8 k $\Omega$
7	1	Resistencia de 2.7 k $\Omega$
8	8	Resistencia de 10 k $\Omega$
9	1	LED
10	1	7805C
11	1	Oscilador de 11.0592 MHz
12	1	Pushbutton NO
13	1	8032
14	1	74LS373
15	1	74LS138
16	1	74LS08
17	1	EPROM 2764
18	1	Static Ram 6264
19	1	MAXIM232CPE



### INTERFACE PARA TECLADO Y LCD

El siguiente diagrama muestra la conexión de un teclado hexadecimal y una pantalla de cristal líquido al sistema controlador. Los dispositivos a utilizar son:

ÍTEM	CANTIDAD	PARTE
1	1	74LS373
2	1	74LS02
3	1	74C922N
4	1	Potenciómetro 10 k $\Omega$
5	1	Interruptor NO
6	4	Condensadores de cerámica de 0.1 $\mu$ F
7	1	Condensador de cerámica de 1 $\mu$ F
8	1	Teclado Hexadecimal
9	1	LCD OPTREX DMC-16230



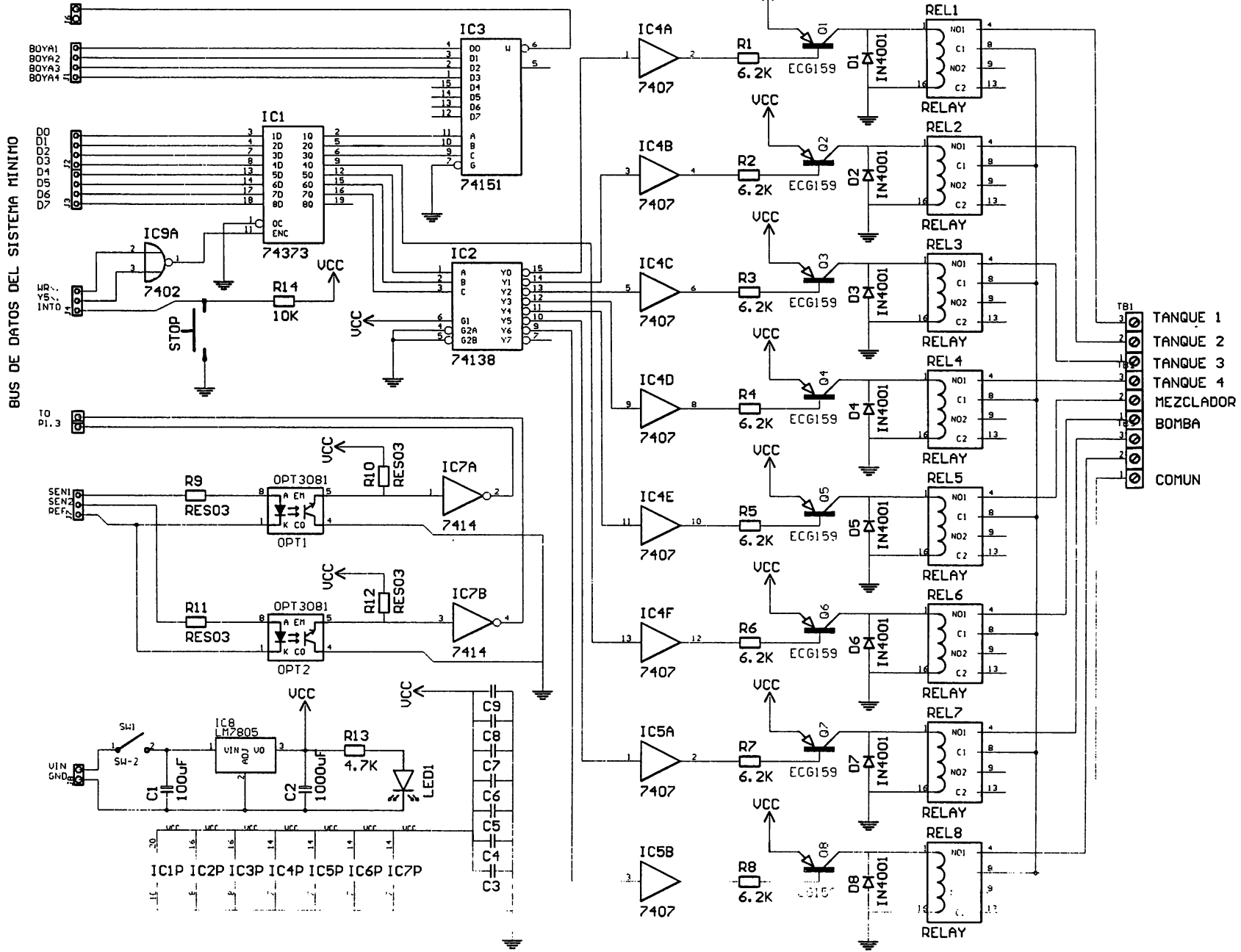
## INTERFACE PARA RELÉ

En este diagrama, se muestra la conexión del relé al sistema mínimo para servir como acturador, es de notar que el sistema es secuencial, por lo que si se desea más de una salida encendida al mismo tiempo, se puede utilizar otro contacto del relé.

ÍTEM	CANTIDAD	PRODUCTO
1	1	74LS373
2	1	74151
3	1	74LS02
4	1	74LS138
5	2	7407
6	8	2N3906
7	8	1N4001
8	8	6.2 k $\Omega$
9	1	7805C
10	1	Condensador de 1000 $\mu$ F electrolítico
11	1	Condensador de 10 $\mu$ F electrolítico
12	1	LED
13	1	Resistencia de 2.7 k $\Omega$
14	8	1N4001
15	8	RELAY 47SB057
16	7	Condensadores de cerámica de 0.1 $\mu$ F
17	1	7805C
18	1	Pushbutton
19	1	Resistencia de 10 k $\Omega$

Los octoacopladores y sus resistencias sirven para aislar las señales externas de pulsos de sensores como flujo, bera capacitivo, etc. No se presentan valores debido a que no se implementaron. Pero es necesario que se incluyan para dar mejor protección al sistema.

# ETAPA DE POTENCIA



RELEVADORES



**MCS<sup>®</sup>-51**  
**8-BIT CONTROL-ORIENTED MICROCOMPUTERS**  
**8031/8051**  
**8031AH/8051AH**  
**8032AH/8052AH**  
**8751H/8751H-8**

- High Performance HMOS Process
- Internal Timers/Event Counters
- 2-Level Interrupt Priority Structure
- 32 I/O Lines (Four 8-Bit Ports)
- 64K Program Memory Space
- Security Feature Protects EPROM Parts Against Software Piracy
- Boolean Processor
- Bit-Addressable RAM
- Programmable Full Duplex Serial Channel
- 111 Instructions (64 Single-Cycle)
- 64K Data Memory Space

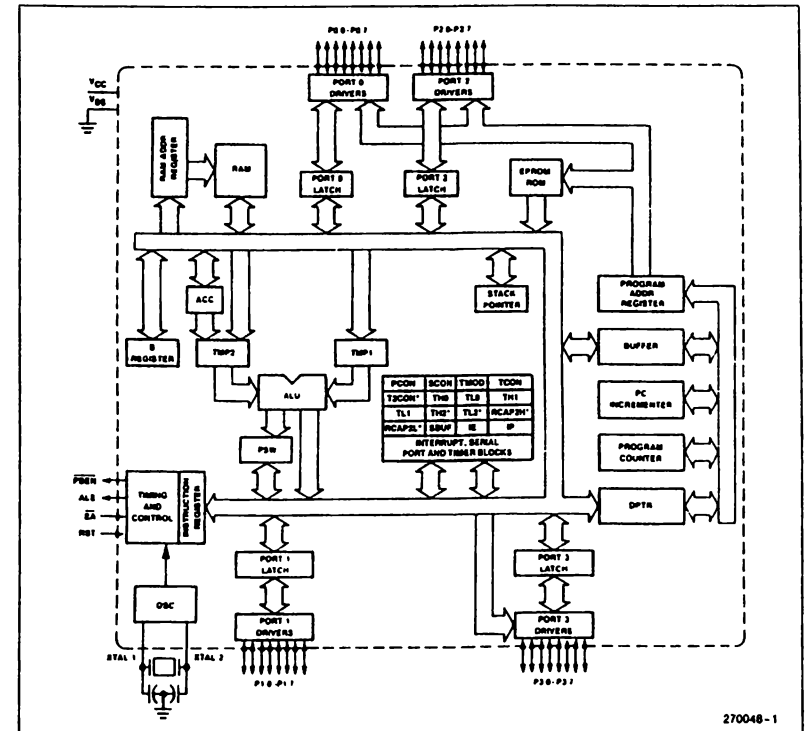
The MCS<sup>®</sup>-51 products are optimized for control applications. Byte-processing and numerical operations on small data structures are facilitated by a variety of fast addressing modes for accessing the internal RAM. The instruction set provides a convenient menu of 8-bit arithmetic instructions, including multiply and divide instructions. Extensive on-chip support is provided for one-bit variables as a separate data type, allowing direct bit manipulation and testing in control and logic systems that require Boolean processing.

The 8051 is the original member of the MCS-51 family. The 8051AH is identical to the 8051, but it is fabricated with HMOS II technology.

The 8751H is an EPROM version of the 8051AH; that is, the on-chip Program Memory can be electrically programmed, and can be erased by exposure to ultraviolet light. It is fully compatible with its predecessor, the 8751-8, but incorporates two new features: a Program Memory Security bit that can be used to protect the EPROM against unauthorized read-out, and a programmable baud rate modification bit (SMOD). The 8751H-8 is identical to the 8751H but only operates up to 8 MHz.

The 8052AH is an enhanced version of the 8051AH. It is backwards compatible with the 8051AH and is fabricated with HMOS II technology. The 8052AH enhancements are listed in the table below. Also refer to this table for the ROM, ROMless, and EPROM versions of each product.

Device	Internal Memory		Timers/ Event Counters	Interrupts
	Program	Data		
8052AH	8K x 8 ROM	256 x 8 RAM	3 x 16-Bit	6
8051AH	4K x 8 ROM	128 x 8 RAM	2 x 16-Bit	5
8051	4K x 8 ROM	128 x 8 RAM	2 x 16-Bit	5
8032AH	none	256 x 8 RAM	3 x 16-Bit	6
8031AH	none	128 x 8 RAM	2 x 16-Bit	5
8031	none	128 x 8 RAM	2 x 16-Bit	5
8751H	4K x 8 EPROM	128 x 8 RAM	2 x 16-Bit	5
8751H-8	4K x 8 EPROM	128 x 8 RAM	2 x 16-Bit	5

Figure 1. MCS<sup>®</sup>-51 Block Diagram**PACKAGES**

Part	Prefix	Package Type
8051AH/ 8031AH	P D N	40-Pin Plastic DIP 40-Pin CERDIP 44-Pin PLCC
8052AH/ 8032AH	P D N	40-Pin Plastic DIP 40-Pin CERDIP 44-Pin PLCC
8751H/ 8751H-8	D	40-Pin CERDIP

**PIN DESCRIPTIONS**

**V<sub>CC</sub>**: Supply voltage.

**V<sub>SS</sub>**: Circuit ground.

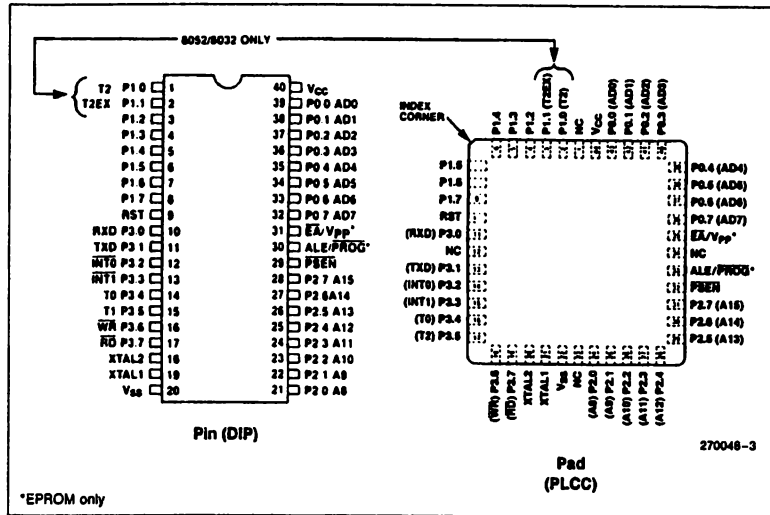
**Port 0**: Port 0 is an 8-bit open drain bidirectional I/O port. As an output port each pin can sink 8 LS TTL inputs.

Port 0 pins that have 1s written to them float, and i that state can be used as high-impedance inputs.

Port 0 is also the multiplexed low-order address and data bus during accesses to external Program and Data Memory. In this application it uses strong internal pullups when emitting 1s and can source and sink 8 LS TTL inputs.

Port 0 also receives the code bytes during programming of the EPROM parts, and outputs the code bytes during program verification of the ROM and EPROM parts. External pullups are required during program verification.

intel

MCS<sup>®</sup>-51Figure 2. MCS<sup>®</sup>-51 Connections

**Port 1:** Port 1 is an 8-bit bidirectional I/O port with internal pullups. The Port 1 output buffers can sink/source 4 LS TTL inputs. Port 1 pins that have 1s written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 1 pins that are externally being pulled low will source current ( $I_{IL}$  on the data sheet) because of the internal pullups.

Port 1 also receives the low-order address bytes during programming of the EPROM parts and during program verification of the ROM and EPROM parts.

In the 8032AH and 8052AH, Port 1 pins P1.0 and P1.1 also serve the T2 and T2EX functions, respectively.

**Port 2:** Port 2 is an 8-bit bidirectional I/O port with internal pullups. The Port 2 output buffers can sink/source 4 LS TTL inputs. Port 2 pins that have 1s written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 2 pins that are externally being pulled low will source current ( $I_{IL}$  on the data sheet) because of the internal pullups.

Port 2 emits the high-order address byte during fetches from external Program Memory and during accesses to external Data Memory that use 16-bit addresses (MOVX @DPTR). In this application it uses strong internal pullups when emitting 1s. Dur-

ing accesses to external Data Memory that use 8-bit addresses (MOVX @RI), Port 2 emits the contents of the P2 Special Function Register.

Port 2 also receives the high-order address bits during programming of the EPROM parts and during program verification of the ROM and EPROM parts.

**Port 3:** Port 3 is an 8-bit bidirectional I/O port with internal pullups. The Port 3 output buffers can sink/source 4 LS TTL inputs. Port 3 pins that have 1s written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 3 pins that are externally being pulled low will source current ( $I_{IL}$  on the data sheet) because of the pullups.

Port 3 also serves the functions of various special features of the MCS-51 Family, as listed below:

Port Pin	Alternative Function
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	INT0 (external interrupt 0)
P3.3	INT1 (external interrupt 1)
P3.4	T0 (Timer 0 external input)
P3.5	T1 (Timer 1 external input)
P3.6	WR (external data memory write strobe)
P3.7	RD (external data memory read strobe)

intel

MCS<sup>®</sup>-51

**RST:** Reset input. A high on this pin for two machine cycles while the oscillator is running resets the device.

**ALE/PROG:** Address Latch Enable output pulse for latching the low byte of the address during accesses to external memory. This pin is also the program pulse input (PROG) during programming of the EPROM parts.

In normal operation ALE is emitted at a constant rate of  $\frac{1}{6}$  the oscillator frequency, and may be used for external timing or clocking purposes. Note, however, that one ALE pulse is skipped during each access to external Data Memory.

**PSEN:** Program Store Enable is the read strobe to external Program Memory.

When the device is executing code from external Program Memory, PSEN is activated twice each machine cycle, except that two PSEN activations are skipped during each access to external Data Memory.

**EA/Vpp:** External Access enable EA must be strapped to VSS in order to enable any MCS-51 device to fetch code from external Program memory locations starting at 0000H up to FFFFH. EA must be strapped to VCC for internal program execution.

Note, however, that if the Security Bit in the EPROM devices is programmed, the device will not fetch code from any location in external Program Memory.

This pin also receives the 21V programming supply voltage (VPP) during programming of the EPROM parts.

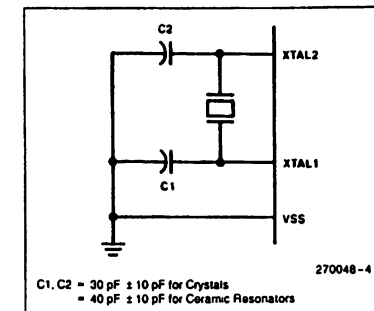


Figure 3. Oscillator Connections

**XTAL1:** Input to the inverting oscillator amplifier.

**XTAL2:** Output from the inverting oscillator amplifier.

## OSCILLATOR CHARACTERISTICS

XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier which can be configured for use as an on-chip oscillator, as shown in Figure 3. Either a quartz crystal or ceramic resonator may be used. More detailed information concerning the use of the on-chip oscillator is available in Application Note AP-155, "Oscillators for Microcontrollers."

To drive the device from an external clock source, XTAL1 should be grounded, while XTAL2 is driven, as shown in Figure 4. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is through a divide-by-two flip-flop, but minimum and maximum high and low times specified on the Data Sheet must be observed.

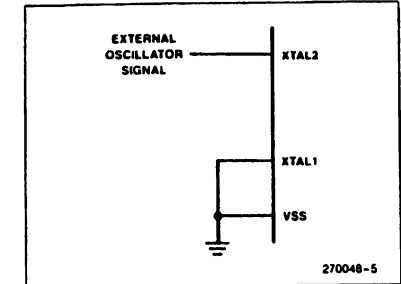


Figure 4. External Drive Configuration

## DESIGN CONSIDERATIONS

If an 8751BH or 8752BH may replace an 8751H in a future design, the user should carefully compare both data sheets for DC or AC Characteristic differences. Note that the  $V_{IH}$  and  $I_{IH}$  specifications for the EA pin differ significantly between the devices.

Exposure to light when the EPROM device is in operation may cause logic errors. For this reason, it is suggested that an opaque label be placed over the window when the die is exposed to ambient light.

MCS<sup>®</sup>-51**ABSOLUTE MAXIMUM RATINGS\***

Ambient Temperature Under Bias ..... 0°C to 70°C  
 Storage Temperature ..... -65°C to +150°C  
 Voltage on  $\overline{EA}/V_{PP}$  Pin to  $V_{SS}$  .... -0.5V to +21.5V  
 Voltage on Any Other Pin to  $V_{SS}$  .... -0.5V to +7V  
 Power Dissipation ..... 1.5W

NOTICE: This is a production data sheet. The specifications are subject to change without notice.

\*WARNING: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.

Operating Conditions:  $T_A$  (Under Bias) = 0°C to +70°C;  $V_{CC}$  = 5V  $\pm$  10%;  $V_{SS}$  = 0V

**D.C. CHARACTERISTICS** (Under Operating Conditions)

Symbol	Parameter	Min	Max	Units	Test Conditions
$V_{IL}$	Input Low Voltage (Except $\overline{EA}$ Pin of 8751H & 8751H-8)	-0.5	0.8	V	
$V_{IL1}$	Input Low Voltage to $\overline{EA}$ Pin of 8751H & 8751H-8	0	0.7	V	
$V_{IH}$	Input High Voltage (Except XTAL2, RST)	2.0	$V_{CC} + 0.5$	V	
$V_{IH1}$	Input High Voltage to XTAL2, RST	2.5	$V_{CC} + 0.5$	V	$XTAL1 = V_{SS}$
$V_{OL}$	Output Low Voltage (Ports 1, 2, 3)*		0.45	V	$I_{OL} = 1.6$ mA
$V_{OL1}$	Output Low Voltage (Port 0, ALE, PSEN)*				
	8751H, 8751H-8		0.60 0.45	V	$I_{OL} = 3.2$ mA $I_{OL} = 2.4$ mA
	All Others		0.45	V	$I_{OL} = 3.2$ mA
$V_{OH}$	Output High Voltage (Ports 1, 2, 3, ALE, PSEN)	2.4		V	$I_{OH} = -80$ $\mu$ A
$V_{OH1}$	Output High Voltage (Port 0 in External Bus Mode)	2.4		V	$I_{OH} = -400$ $\mu$ A
$I_{IL}$	Logical 0 Input Current (Ports 1, 2, 3, RST) 8032AH, 8052AH All Others		-800 -500	$\mu$ A $\mu$ A	$V_{IN} = 0.45$ V $V_{IN} = 0.45$ V
$I_{IL1}$	Logical 0 Input Current to $\overline{EA}$ Pin of 8751H & 8751H-8 Only		-15	mA	$V_{IN} = 0.45$ V
$I_{IL2}$	Logical 0 Input Current (XTAL2)		-3.2	mA	$V_{IN} = 0.45$ V
$I_{LI}$	Input Leakage Current (Port 0) 8751H & 8751H-8 All Others		$\pm 100$ $\pm 10$	$\mu$ A $\mu$ A	$0.45 \leq V_{IN} \leq V_{CC}$ $0.45 \leq V_{IN} \leq V_{CC}$
			500	$\mu$ A	$V_{IN} = 2.4$ V
$I_{IH1}$	Input Current to RST to Activate Reset		500	$\mu$ A	$V_{IN} < (V_{CC} - 1.5$ V)
$I_{CC}$	Power Supply Current: 8031/8051 8031AH/8051AH 8032AH/8052AH 8751H/8751H-8		160	mA	All Outputs Disconnected; $\overline{EA} = V_{CC}$
			125	mA	
			175	mA	
			250	mA	
$C_{IO}$	Pin Capacitance		10	pF	Test freq = 1 MHz

**\*NOTE:**

Capacitive loading on Ports 0 and 2 may cause spurious noise pulses to be superimposed on the  $V_{OL}$ s of ALE and Ports 1 and 3. The noise is due to external bus capacitance discharging into the Port 0 and Port 2 pins when these pins make 1-to-0 transitions during bus operations. In the worst cases (capacitive loading > 100 pF), the noise pulse on the ALE line may exceed 0.8V. In such cases it may be desirable to qualify ALE with a Schmitt Trigger, or use an address latch with a Schmitt Trigger STROBE input.

MCS<sup>®</sup>-51**A.C. CHARACTERISTICS** Under Operating Conditions;  
Load Capacitance for Port 0, ALE, and PSEN = 100 pF;  
Load Capacitance for All Other Outputs = 80 pF

Symbol	Parameter	12 MHz Oscillator		Variable Oscillator		Units	
		Min	Max	Min	Max		
1/TCLCL	Oscillator Frequency			3.5	12.0	MHz	
TLHLL	ALE Pulse Width	127		2TCLCL - 40		ns	
TAVLL	Address Valid to ALE Low	43		TCLCL - 40		ns	
TLLAX	Address Hold after ALE Low	48		TCLCL - 35		ns	
TLLIV	ALE Low to Valid Instr In 8751H All Others		183 233		4TCLCL - 150 4TCLCL - 100	ns ns	
TLLPL	ALE Low to PSEN Low	58		TCLCL - 25		ns	
TPLPH	PSEN Pulse Width 8751H All Others		190 215		3TCLCL - 60 3TCLCL - 35	ns ns	
TPLIV	PSEN Low to Valid Instr In 8751H All Others		100 125		3TCLCL - 150 3TCLCL - 125	ns ns	
TPXIX	Input Instr Hold after PSEN	0		0		ns	
TPXIZ	Input Instr Float after PSEN		63		TCLCL - 20	ns	
TPXAV	PSEN to Address Valid	75		TCLCL - 8		ns	
TAVIV	Address to Valid Instr In 8751H All Others		267 302		5TCLCL - 150 5TCLCL - 115	ns ns	
TPLAZ	PSEN Low to Address Float		20		20	ns	
TRLRH	$\overline{RD}$ Pulse Width	400		6TCLCL - 100		ns	
TWLWH	$\overline{WR}$ Pulse Width	400		6TCLCL - 100		ns	
TRLDV	$\overline{RD}$ Low to Valid Data In		252		5TCLCL - 165	ns	
TRHDX	Data Hold after $\overline{RD}$	0		0		ns	
TRHDZ	Data Float after $\overline{RD}$		97		2TCLCL - 70	ns	
TLLDV	ALE Low to Valid Data In		517		8TCLCL - 150	ns	
TAVDV	Address to Valid Data In		585		9TCLCL - 165	ns	
TLLWL	ALE Low to $\overline{RD}$ or $\overline{WR}$ Low	200	300	3TCLCL - 50	3TCLCL + 50	ns	
TAVWL	Address to $\overline{RD}$ or $\overline{WR}$ Low	203		4TCLCL - 130		ns	
TQVWX	Data Valid to $\overline{WR}$ Transition 8751H All Others		13 23		TCLCL - 70 TCLCL - 60	ns ns	
TQVWH	Data Valid to $\overline{WR}$ High	433		7TCLCL - 150		ns	
TWHQX	Data Hold after $\overline{WR}$	33		TCLCL - 50		ns	
TRLAZ	$\overline{RD}$ Low to Address Float		20		20	ns	
TWHLH	$\overline{RD}$ or $\overline{WR}$ High to ALE High 8751H All Others		33 43	133 123	TCLCL - 50 TCLCL - 40	TCLCL + 50 TCLCL + 40	ns ns

**NOTE:**

\*This table does not include the 8751-8 A.C. characteristics (see next page).

intel

MCS<sup>®</sup>-51

This Table is only for the 8751H-8

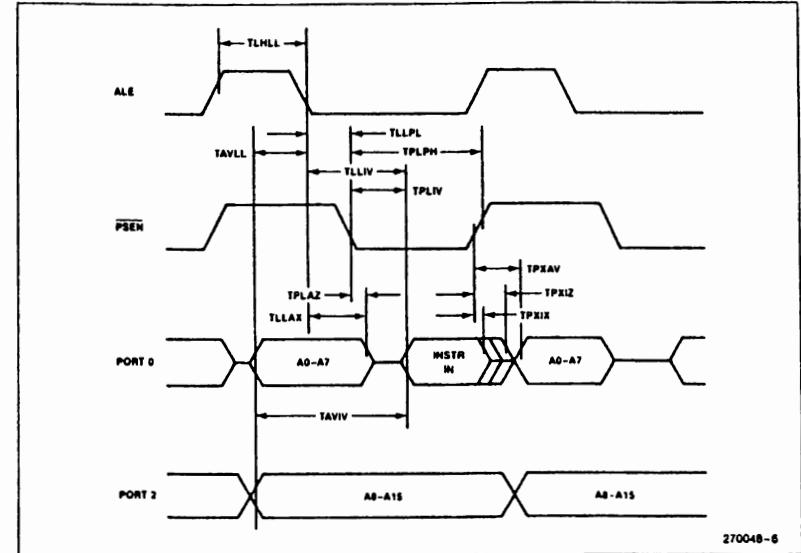
**A.C. CHARACTERISTICS** Under Operating Conditions;  
Load Capacitance for Port 0, ALE, and PSEN = 100 pF;  
Load Capacitance for All Other Outputs = 80 pF

Symbol	Parameter	8 MHz Oscillator		Variable Oscillator		Units
		Min	Max	Min	Max	
1/TCLCL	Oscillator Frequency			3.5	8.0	MHz
TLHLL	ALE Pulse Width	210		2TCLCL - 40		ns
TAVLL	Address Valid to ALE Low	85		TCLCL - 40		ns
TLLAX	Address Hold after ALE Low	90		TCLCL - 35		ns
TLLIV	ALE Low to Valid Instr In		350		4TCLCL - 150	ns
TLLPL	ALE Low to PSEN Low	100		TCLCL - 25		ns
TPLPH	PSEN Pulse Width	315		3TCLCL - 60		ns
TPLIV	PSEN Low to Valid Instr In		225		3TCLCL - 150	ns
TPXIX	Input Instr Hold after PSEN	0		0		ns
TPXIZ	Input Instr Float after PSEN		105		TCLCL - 20	ns
TPXAV	PSEN to Address Valid	117		TCLCL - 8		ns
TAVIV	Address to Valid Instr In		475		5TCLCL - 150	ns
TPLAZ	PSEN Low to Address Float		20		20	ns
TRLRH	RD Pulse Width	650		6TCLCL - 100		ns
TWLWH	WR Pulse Width	650		6TCLCL - 100		ns
TRLDV	RD Low to Valid Data In		460		5TCLCL - 165	ns
TRHDX	Data Hold after RD	0		0		ns
TRHDZ	Data Float after RD		180		2TCLCL - 70	ns
TLLDV	ALE Low to Valid Data In		850		8TCLCL - 150	ns
TAVDV	Address to Valid Data In		960		9TCLCL - 165	ns
TLLWL	ALE Low to RD or WR Low	325	425	3TCLCL - 50	3TCLCL + 50	ns
TAVWL	Address to RD or WR Low	370		4TCLCL - 130		ns
TQVWX	Data Valid to WR Transition	55		TCLCL - 70		ns
TQVWH	Data Valid to WR High	725		7TCLCL - 150		ns
TWHQX	Data Hold after WR	75		TCLCL - 50		ns
TRLAZ	RD Low to Address Float		20		20	ns
TWHLH	RD or WR High to ALE High	75	175	TCLCL - 50	TCLCL + 50	ns

intel

MCS<sup>®</sup>-51

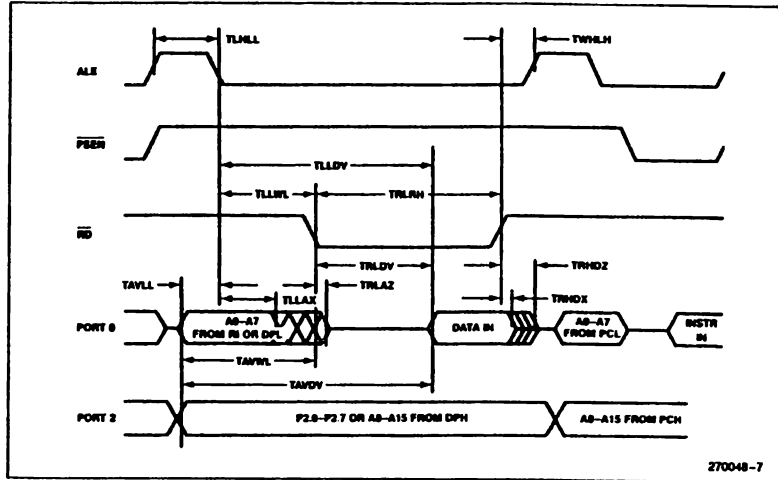
## EXTERNAL PROGRAM MEMORY READ CYCLE





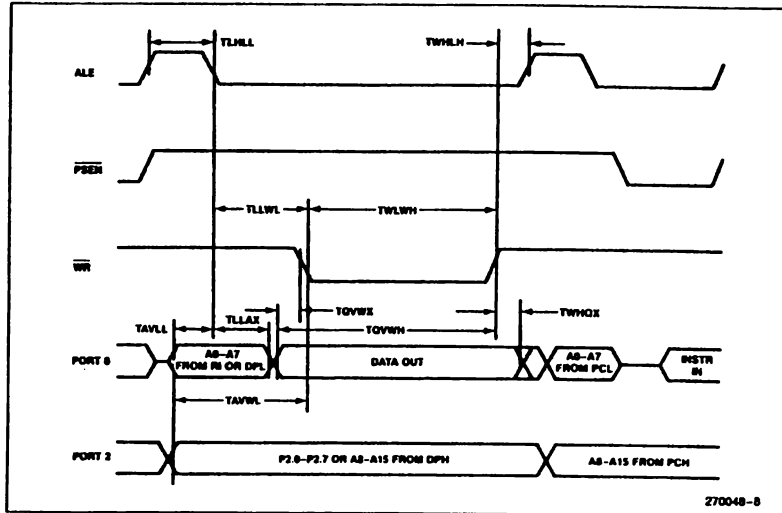
MCS<sup>®</sup>-51

EXTERNAL DATA MEMORY READ CYCLE



270048-7

EXTERNAL DATA MEMORY WRITE CYCLE



270048-8



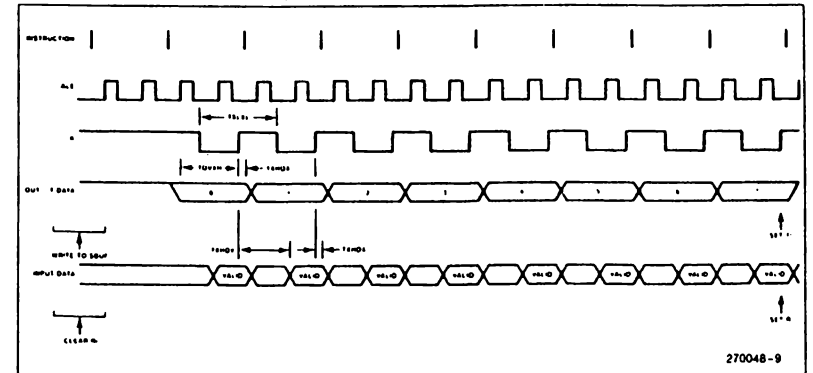
MCS<sup>®</sup>-51

SERIAL PORT TIMING—SHIFT REGISTER MODE

Test Conditions:  $T_A = 0^\circ\text{C to } 70^\circ\text{C}$ ;  $V_{CC} = 5V \pm 10\%$ ;  $V_{SS} = 0V$ ; Load Capacitance = 80 pF

Symbol	Parameter	12 MHz Oscillator		Variable Oscillator		Units
		Min	Max	Min	Max	
TXLXL	Serial Port Clock Cycle Time	1.0		12TCLCL		$\mu\text{s}$
TQVXH	Output Data Setup to Clock Rising Edge	700		10TCLCL - 133		ns
TXHOX	Output Data Hold after Clock Rising Edge	50		2TCLCL - 117		ns
TXHDX	Input Data Hold after Clock Rising Edge	0		0		ns
TXHDV	Clock Rising Edge to Input Data Valid		700		10TCLCL - 133	ns

SHIFT REGISTER TIMING WAVEFORMS



270048-9

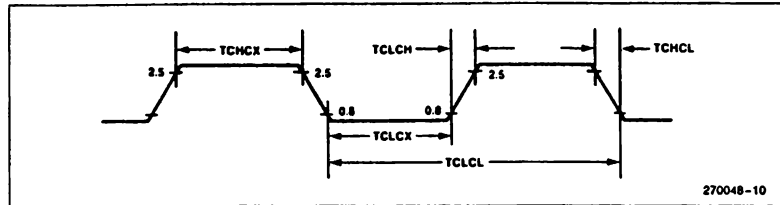
intel

MCS<sup>®</sup>-51

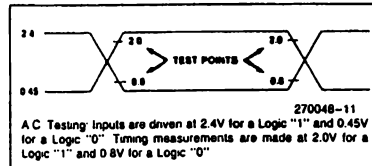
## EXTERNAL CLOCK DRIVE

Symbol	Parameter	Min	Max	Units
1/TCLCL	Oscillator Frequency (except 8751H-8)	3.5	12	MHz
	8751H-8	3.5	8	MHz
TCHCX	High Time	20		ns
TCLCX	Low Time	20		ns
TCLCH	Rise Time		20	ns
TCHCL	Fall Time		20	ns

## EXTERNAL CLOCK DRIVE WAVEFORM



## A.C. TESTING INPUT, OUTPUT WAVEFORM



intel

MCS<sup>®</sup>-51

## EPROM CHARACTERISTICS

Table 3. EPROM Programming Modes

Mode	RST	PSEN	ALE	EA	P2.7	P2.6	P2.5	P2.4
Program	1	0	0*	VPP	1	0	X	X
Inhibit	1	0	1	X	1	0	X	X
Verify	1	0	1	1	0	0	X	X
Security Set	1	0	0*	VPP	1	1	X	X

## NOTE:

"1" = logic high for that pin  
 "0" = logic low for that pin  
 "X" = "don't care"

"VPP" = +21V ± 0.5V

\*ALE is pulsed low for 50 ms.

## Programming the EPROM

To be programmed, the part must be running with a 4 to 8 MHz oscillator. (The reason the oscillator needs to be running is that the internal bus is being used to transfer address and program data to appropriate internal registers.) The address of an EPROM location to be programmed is applied to Port 1 and pins P2.0-P2.3 of Port 2, while the code byte to be programmed into that location is applied to Port 0. The other Port 2 pins, and RST, PSEN, and EA should be held at the "Program" levels indicated in Table 3. ALE is pulsed low for 50 ms to program the code byte into the addressed EPROM location. The setup is shown in Figure 5.

Normally EA is held at a logic high until just before ALE is to be pulsed. Then EA is raised to +21V, ALE is pulsed, and then EA is returned to a logic high. Waveforms and detailed timing specifications are shown in later sections of this data sheet.

Note that the EA/VPP pin must not be allowed to go above the maximum specified VPP level of 21.5V for any amount of time. Even a narrow glitch above that voltage level can cause permanent damage to the device. The VPP source should be well regulated and free of glitches.

## Program Verification

If the Security Bit has not been programmed, the on-chip Program Memory can be read out for verification purposes, if desired, either during or after the programming operation. The address of the Program Memory location to be read is applied to Port 1 and pins P2.0-P2.3. The other pins should be held at the "Verify" levels indicated in Table 3. The contents of the addressed location will come out on Port 0. External pullups are required on Port 0 for this operation.

The setup, which is shown in Figure 6, is the same as for programming the EPROM except that pin P2.7 is held at a logic low, or may be used as an active-low read strobe.

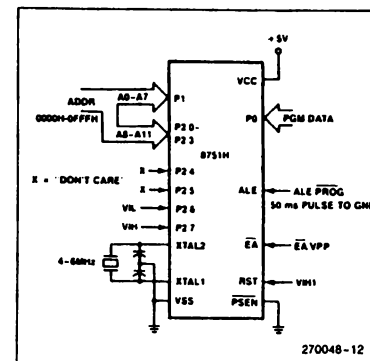


Figure 5. Programming Configuration

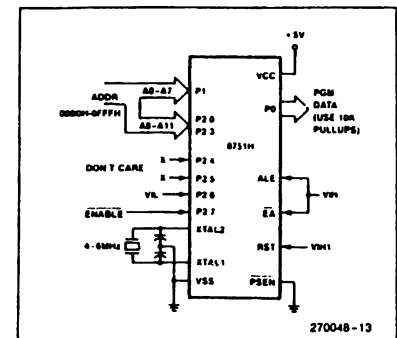


Figure 6. Program Verification



# DMC-16230 (16 Characters × 2 lines)

● Display Fonts 5 × 7 Dots + Cursor ● 1/16 Duty Drive

## ■ ABSOLUTE MAXIMUM RATINGS

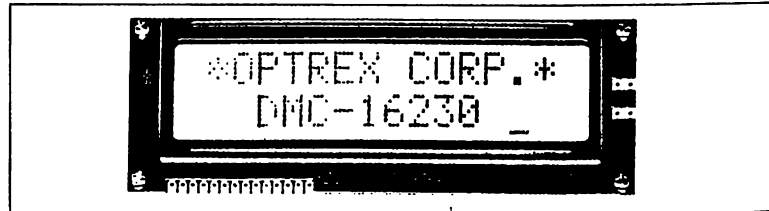
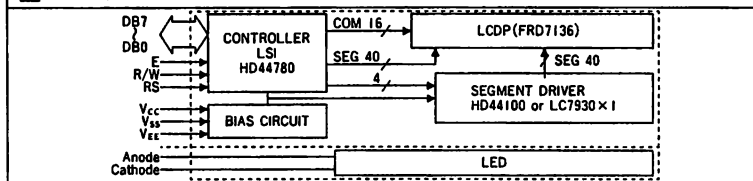
Item	Symbol	Test Condition	Standard Value		Unit
			min.	max.	
Supply Voltage for Logic	$V_{CC}-V_{SS}$	$T_a=25^{\circ}\text{C}$	-0.3	7	V
Supply Voltage for LCD Drive	$V_{CC}-V_{EE}$	$T_a=25^{\circ}\text{C}$	$V_{CC}-13.5$	$V_{CC}+0.3$	V
Input Voltage	$V_I$	$T_a=25^{\circ}\text{C}$	-0.3	$V_{CC}+0.3$	V
LED Forward Current	$I_F$	—	—	600	mA
LED Reverse Voltage	$V_R$	—	—	8	V
LED Power Loss	$P_D$	—	—	0.26	W
Operating Temperature	$T_{opr}$	—	0	+50	$^{\circ}\text{C}$
Storage Temperature	$T_{stg}$	—	-20	+70	$^{\circ}\text{C}$

## ■ ELECTRICAL CHARACTERISTICS

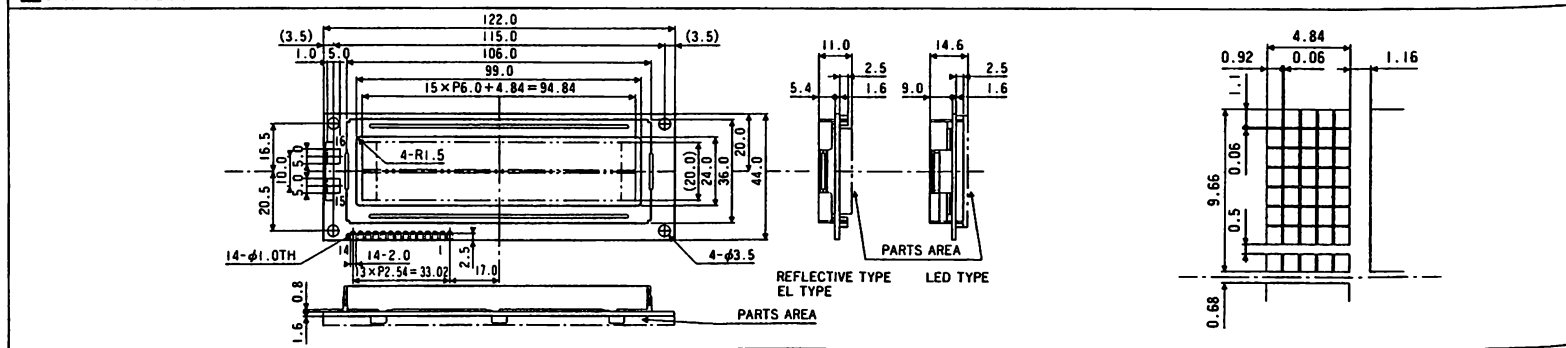
Item	Symbol	Test Condition	Standard Value			Unit
			min.	typ.	max.	
Input "High" Voltage	$V_{IH}$	—	2.2	—	$V_{CC}$	V
Input "Low" Voltage	$V_{IL}$	—	0	—	0.6	V
Output "High" Voltage	$V_{OH}$	$-I_{OH}=0.205\text{mA}$	2.4	—	$V_{CC}$	V
Output "Low" Voltage	$V_{OL}$	$I_{OL}=1.2\text{mA}$	0	—	0.4	V
LED Forward Voltage	$V_F$	$I_F=300\text{mA}$	3.8	4.0	4.2	V
Brightness *1	L	$I_F=300\text{mA}$	30	50	—	$\text{cd}/\text{m}^2$
Supply Current	$I_{CC}$	$V_{CC}=5.0\text{V}$	—	2.0	5.0	mA

\* $V_{CC}=5.0\text{V}\pm 5\%$ ,  $T_a=25^{\circ}\text{C}$  \*NOTE 1) Measured at the bare LED backlight unit.

## ■ BLOCK DIAGRAM



## ■ DIMENSION



# INSTRUCTIONS (Except for DMC40401 series)

Instruction	Code										Description	Executed Time(max.) fosc=250KHz
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
Clear Display	0	0	0	0	0	0	0	0	0	1	Clears all display and returns the cursor to the home position (Address 0).	1.64mS
Cursor At Home	0	0	0	0	0	0	0	0	1	*	Returns the cursor to the home position (Address 0). Also returns the display being shifted to the original position DDAM contents remain unchanged.	1.64mS
Entry Mode Set	0	0	0	0	0	0	0	1	I/D	S	Sets the cursor move direction and specifies or not to shift the display. These operations are performed during data write and read.	40μS
Display On/Off Control	0	0	0	0	0	0	1	D	C	B	Sets ON/OFF of all display (D) cursor ON/OFF (C), and blink of cursor position character (B).	40μS
Cursor/Display Shift	0	0	0	0	0	1	S/C	R/L	*	*	Moves the cursor and shifts the display without changing DDRAM contents.	40μS
Function Set	0	0	0	0	1	DL	N	F	*	*	Sets interface data length(DL), number of display lines(N) and character font(F).	40μS
CGRAM Address Set	0	0	0	1	A <sub>CG</sub>					Sets the CGRAM, data is sent and received after this setting.		40μS
DDRAM Address Set	0	0	1	A <sub>DD</sub>					Sets the CGRAM, data is sent and received after this setting.		40μS	
Busy Flag/Address Read	0	1	BF	AC					Reads Busy flag (FB) indicating internal operation is being performed and reads address counter contents.		0μS	
CGRAM/DDRAM Data Write	1	0	WRITE DATA					Writes data into DDRAM or CGRAM.		40μS		
CGRAM/DDRAM Data Read	1	1	READ DATA					Reads data into DDRAM or CGRAM.		40μS		

Code	Description	Executed Time(max.)
I/D=1: Increment I/D=0: Decrement S=1: With display shift S/C=1: Display shift S/C=0: Cursor movement R/L=1: Shift to the right R/L=0: Shift to the left DL=1:8-bit DL=0:4-bit I/16Duty I/8, I/11Duty F=1:5×10dots F=0:5×7dots BF=1:Internal operation is being performed BF=0:Instruction acceptable	DDRAM: Display Data RAM CGRAM: Character Generator RAM ACG: CGRAM Address ADD: DDRAM Address Corresponds to cursor address. AC: Address Counter, used for both DDRAM and CGRAM *: Invalid	fcp or fosc=250kHz However, when frequency changes, execution time also changes  Ex If fcp or fosc is 270kHz, $40\mu S \times \frac{250}{270} = 37\mu S$

## FONT TABLE (5×11Dots)

(5×8Dots)

Lower 4-bit	Upper 4-bit	0000	0010	0011	0100	0101	0110	0111	1010	1011	1100	1101	1110	1111
xxxx0000	CGRAM	G	a	P	'	P	-	9	3	0	P			
xxxx0001	(2)	!	1	H	Q	a	4	6	7	7	4	a	q	
xxxx0010	(3)	"	2	B	R	b	r	"	4	9	x	P	0	
xxxx0011	(4)	#	3	C	S	c	s	]	0	T	E	s	*	
xxxx0100	(5)	\$	4	D	T	d	t	.	!	!	!	!	!	!
xxxx0101	(6)	%	5	E	U	e	u	.	!	!	!	!	!	!
xxxx0110	(7)	&	6	F	V	f	v	!	!	!	!	!	!	!
xxxx0111	(8)	'	7	G	W	g	w	!	!	!	!	!	!	!
xxxx1000	(1)	(	8	H	X	h	x	!	!	!	!	!	!	!
xxxx1001	(2)	)	9	I	Y	i	y	!	!	!	!	!	!	!
xxxx1010	(3)	*	:	J	Z	j	z	!	!	!	!	!	!	!
xxxx1011	(4)	+	;	K	L	k	l	!	!	!	!	!	!	!
xxxx1100	(5)	,	<	L	#	1	!	!	!	!	!	!	!	!
xxxx1101	(6)	=	=	M	O	m	!	!	!	!	!	!	!	!
xxxx1110	(7)	.	>	N	^	n	!	!	!	!	!	!	!	!
xxxx1111	(8)	/	?	O	_	o	!	!	!	!	!	!	!	!

1110	1111
0	0
a	q
B	0
s	*
W	0
U	0
0	Z
q	π
J	X
!	!
!	!
!	!

\*CGRAM is Character Generator RAM which memorize characters that you can freely input by program.  
 \*32 characters stated under upper 4-bit of 1110 and 1111 are 5×10 dots, and part of which is cut when you use in display which display fonts is 5×7 dots. Please note.

5×11 dots type product:  
 DMC16106A, DMC24138, DMC32132, DMC40131

**MM54C922/MM74C922 16-Key Encoder**  
**MM54C923/MM74C923 20-Key Encoder**

**general description**

These CMOS key encoders provide all the necessary logic to fully encode an array of SPST switches. The keyboard scan can be implemented by either an external clock or external capacitor. These encoders also have on-chip pull-up devices which permit switches with up to 50 kΩ on resistance to be used. No diodes in the switch array are needed to eliminate ghost switches. The internal debounce circuit needs only a single external capacitor and can be defeated by omitting the capacitor. A Data Available output goes to a high level when a valid keyboard entry has been made. The Data Available output returns to a low level when the entered key is released, even if another key is depressed. The Data Available will return high to indicate acceptance of the new key after a normal debounce period; this two key roll over is provided between any two switches.

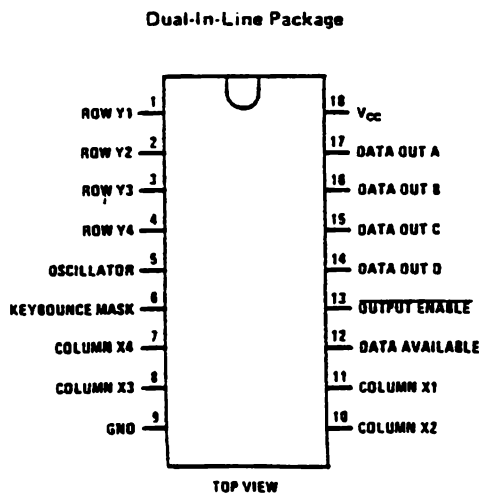
An internal register remembers the last key pressed even after the key is released. The TRI-STATE® outputs

provide for easy expansion and bus operation and are LPTTL compatible.

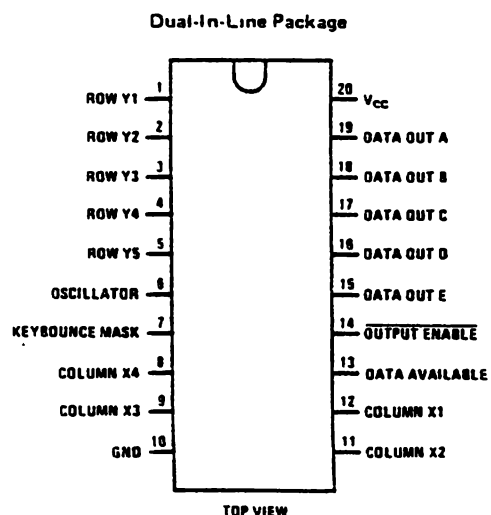
**features**

- 50 kΩ maximum switch on resistance
- On or off chip clock
- On chip row pull-up devices
- 2 key roll-over
- Keybounce elimination with single capacitor
- Last key register at outputs
- TRI-STATE outputs LPTTL compatible
- Wide supply range 3V to 15V
- Low power consumption

**connection diagrams**



Order Number MM54C922N  
or MM74C922N  
See Package 20



Order Number MM54C923N  
or MM74C923N  
See Package 20A

Voltage at Any Pin  
 Operating Temperature Range  
 MM54C922, MM54C923  
 MM74C922, MM74C923  
 Storage Temperature Range

$V_{CC} = 0.3V_{DD}$  to  $V_{DD} = 0.3V_{DD}$   
 $-55^{\circ}\text{C}$  to  $+125^{\circ}\text{C}$   
 $-40^{\circ}\text{C}$  to  $+85^{\circ}\text{C}$   
 $-65^{\circ}\text{C}$  to  $+150^{\circ}\text{C}$

Package Dissipation  
 Operating  $V_{CC}$  Range  
 $V_{CC}$   
 Lead Temperature (Soldering, 10 seconds)

500 mW  
 3V to 15V  
 18V  
 $300^{\circ}\text{C}$

### dc electrical characteristics

Min/max limits apply across temperature range unless otherwise noted

PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS
<b>CMOS TO CMOS</b>					
$V_{T+}$ Positive-Going Threshold Voltage at Osc and KBM Inputs	$V_{CC} = 5V, I_{IN} \geq 0.7\text{ mA}$	3	3.6	4.3	V
	$V_{CC} = 10V, I_{IN} \geq 1.4\text{ mA}$	6	6.8	8.6	V
	$V_{CC} = 15V, I_{IN} \geq 2.1\text{ mA}$	9	10	12.9	V
$V_{T-}$ Negative-Going Threshold Voltage at Osc and KBM Inputs	$V_{CC} = 5V, I_{IN} \geq 0.7\text{ mA}$	0.7	1.4	2	V
	$V_{CC} = 10V, I_{IN} \geq 1.4\text{ mA}$	1.4	3.2	4	V
	$V_{CC} = 15V, I_{IN} \geq 2.1\text{ mA}$	2.1	5	6	V
$V_{IN(1)}$ Logical "1" Input Voltage, Except Osc and KBM Inputs	$V_{CC} = 5V,$	3.5	4.5		V
	$V_{CC} = 10V,$	8	9		V
	$V_{CC} = 15V,$	12.5	13.5		V
$V_{IN(0)}$ Logical "0" Input Voltage, Except Osc and KBM Inputs	$V_{CC} = 5V,$		0.5	1.5	V
	$V_{CC} = 10V,$		1	2	V
	$V_{CC} = 15V,$		1.5	2.5	V
$I_{rp}$ Row Pull-Up Current at Y1, Y2, Y3, Y4 and Y5 Inputs	$V_{CC} = 5V, V_{IN} = 0.1 V_{CC}$		-2	-5	$\mu\text{A}$
	$V_{CC} = 10V$		-10	-20	$\mu\text{A}$
	$V_{CC} = 15V$		-22	-45	$\mu\text{A}$
$V_{OUT(1)}$ Logical "1" Output Voltage	$V_{CC} = 5V, I_O = -10\mu\text{A}$	4.5			V
	$V_{CC} = 10V, I_O = -10\mu\text{A}$	9			V
	$V_{CC} = 15V, I_O = -10\mu\text{A}$	13.5			V
$V_{OUT(0)}$ Logical "0" Output Voltage	$V_{CC} = 5V, I_O = 10\mu\text{A}$			0.5	V
	$V_{CC} = 10V, I_O = 10\mu\text{A}$			1	V
	$V_{CC} = 15V, I_O = 10\mu\text{A}$			1.5	V
$R_{on}$ Column "ON" Resistance at X1, X2, X3 and X4 Outputs	$V_{CC} = 5V, V_O = 0.5V$		500	1400	$\Omega$
	$V_{CC} = 10V, V_O = 1V$		300	700	$\Omega$
	$V_{CC} = 15V, V_O = 1.5V$		200	500	$\Omega$
$I_{CC}$ Supply Current	$V_{CC} = 5V, \text{Osc at } 0V$		0.55	1.1	mA
	$V_{CC} = 10V$		1.1	1.9	mA
	$V_{CC} = 15V$		1.7	2.6	mA
$I_{IN(1)}$ Logical "1" Input Current at Output Enable	$V_{CC} = 15V, V_{IN} = 15V$		0.005	1.0	$\mu\text{A}$
$I_{IN(0)}$ Logical "0" Input Current at Output Enable	$V_{CC} = 15V, V_{IN} = 0V$	-1.0	-0.005		$\mu\text{A}$
<b>CMOS/LPTTL INTERFACE</b>					
$V_{IN(1)}$ Logical "1" Input Voltage, Except Osc and KBM Inputs	54C, $V_{CC} = 4.5V$	$V_{CC}-1.5$			V
	74C, $V_{CC} = 4.75V$	$V_{CC}-1.5$			V
$V_{IN(0)}$ Logical "0" Input Voltage, Except Osc and KBM Inputs	54C, $V_{CC} = 4.5V$			0.8	V
	74C, $V_{CC} = 4.75V$			0.8	V
$V_{OUT(1)}$ Logical "1" Output Voltage	54C, $V_{CC} = 4.5V,$ $I_O = -360\mu\text{A}$	2.4			V
	74C, $V_{CC} = 4.75V,$ $I_O = -360\mu\text{A}$	2.4			V
$V_{OUT(0)}$ Logical "0" Output Voltage	54C, $V_{CC} = 4.5V,$ $I_O = -360\mu\text{A}$			0.4	V
	74C, $V_{CC} = 4.75V,$ $I_O = -360\mu\text{A}$			0.4	V

PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS
<b>OUTPUT DRIVE (See 54C/74C Family Characteristics Data Sheet)</b>					
I <sub>SOURCE</sub> Output Source Current (P-Channel)	V <sub>CC</sub> = 5V, V <sub>OUT</sub> = 0V, T <sub>A</sub> = 25°C	-1.75	-3.3		mA
I <sub>SOURCE</sub> Output Source Current (P-Channel)	V <sub>CC</sub> = 10V, V <sub>OUT</sub> = 0V, T <sub>A</sub> = 25°C	-8	-15		mA
I <sub>SINK</sub> Output Sink Current (N-Channel)	V <sub>CC</sub> = 5V, V <sub>OUT</sub> = V <sub>CC</sub> , T <sub>A</sub> = 25°C	1.75	3.6		mA
I <sub>SINK</sub> Output Sink Current (N-Channel)	V <sub>CC</sub> = 10V, V <sub>OUT</sub> = V <sub>CC</sub> , T <sub>A</sub> = 25°C	8	16		mA

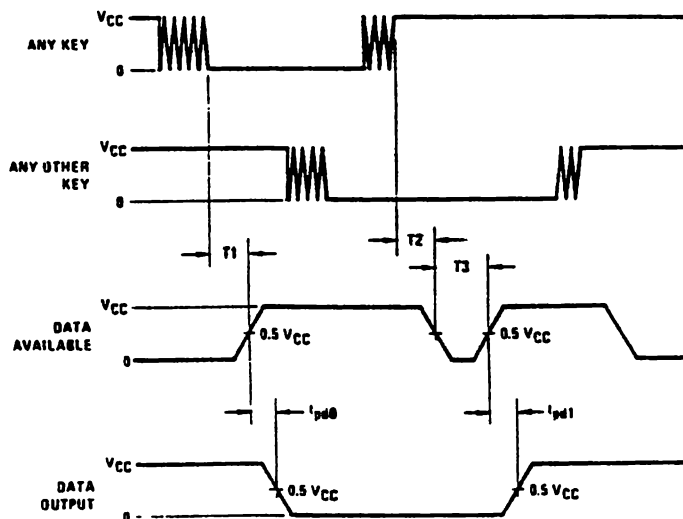
**ac electrical characteristics** T<sub>A</sub> = 25°C

PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS
t <sub>pd0</sub> , t <sub>pd1</sub> Propagation Delay Time to Logical "0" or Logical "1" from D.A.	C <sub>L</sub> = 50 pF, (Figure 1) V <sub>CC</sub> = 5V V <sub>CC</sub> = 10V V <sub>CC</sub> = 15V		60 35 25	150 80 60	ns ns ns
t <sub>OH</sub> , t <sub>1H</sub> Propagation Delay Time from Logical "0" or Logical "1" into High Impedance State	R <sub>L</sub> = 10k, C <sub>L</sub> = 5 pF, (Figure 2) V <sub>CC</sub> = 5V R <sub>L</sub> = 10k V <sub>CC</sub> = 10V C <sub>L</sub> = 10 pF V <sub>CC</sub> = 15V		80 65 50	200 150 110	ns ns ns
t <sub>H0</sub> , t <sub>H1</sub> Propagation Delay Time from High Impedance State to a Logical "0" or Logical "1"	R <sub>L</sub> = 10k, C <sub>L</sub> = 50 pF, (Figure 2) V <sub>CC</sub> = 5V R <sub>L</sub> = 10k V <sub>CC</sub> = 10V C <sub>L</sub> = 50 pF V <sub>CC</sub> = 15V		100 55 40	250 125 90	ns ns ns
C <sub>IN</sub> Input Capacitance	Any Input, (Note 2)		5	7.5	pF
C <sub>OUT</sub> TRI-STATE Output Capacitance	Any Output, (Note 2)		10		pF

**Note 1:** "Absolute Maximum Ratings" are those values beyond which the safety of the device cannot be guaranteed. Except for "Operating Temperature Range" they are not meant to imply that the devices should be operated at these limits. The table of "Electrical Characteristics" provides conditions for actual device operation.

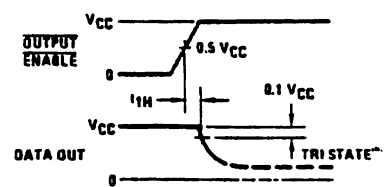
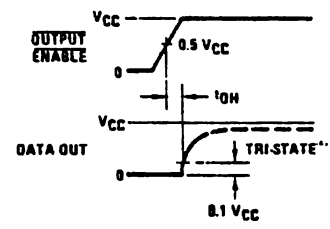
**Note 2:** Capacitance is guaranteed by periodic testing.

**switching time waveforms**

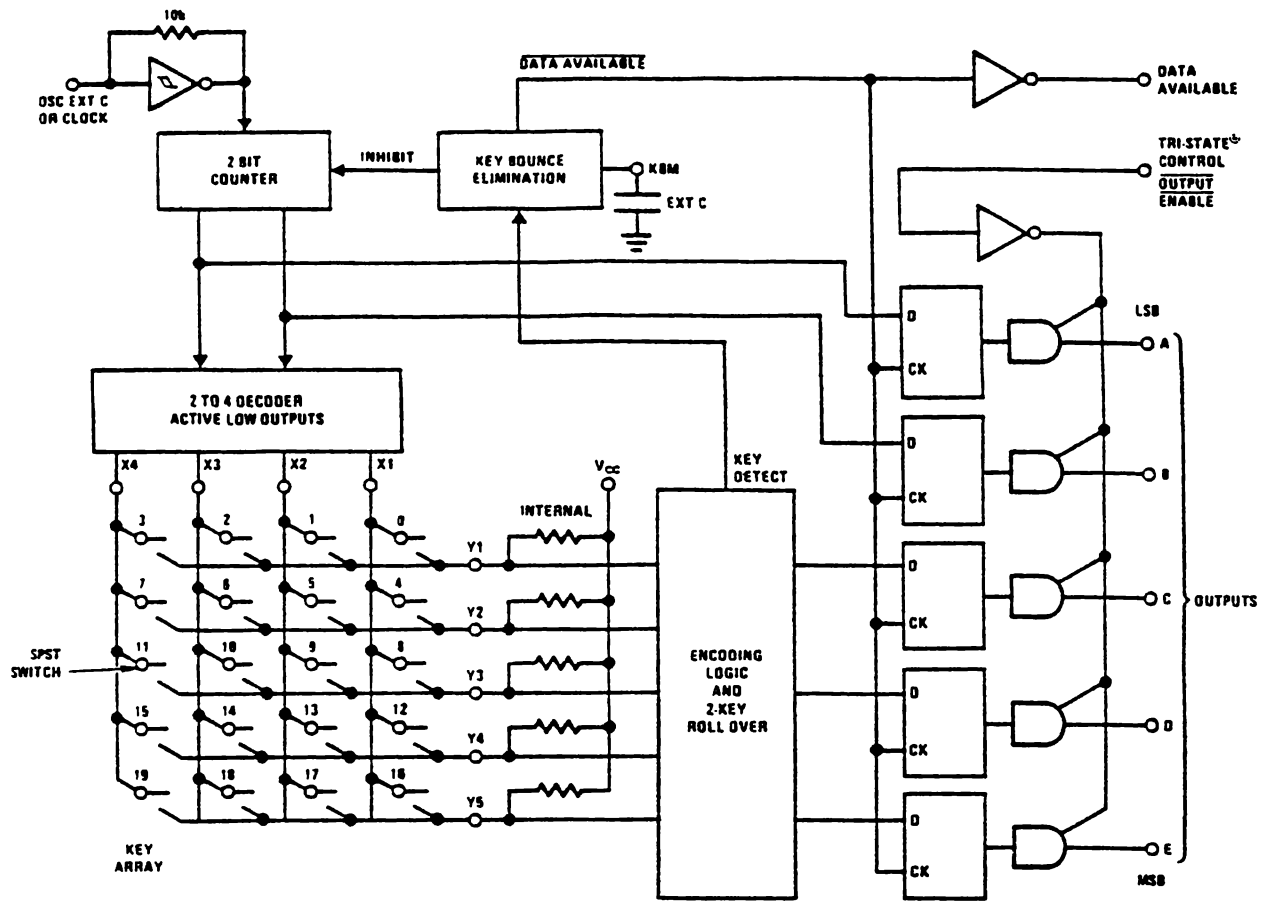


T<sub>1</sub> ≈ T<sub>2</sub> ≈ RC, T<sub>3</sub> ≈ 0.7 RC where R ≈ 10k and C is external capacitor at K<sub>B</sub>M input.

**FIGURE 1**



**FIGURE 2**

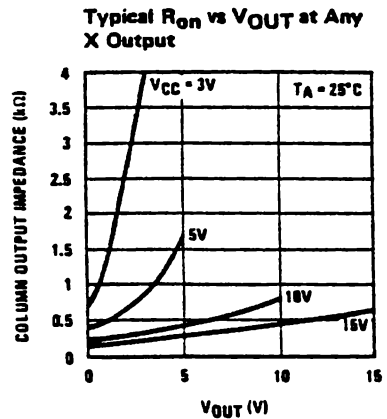
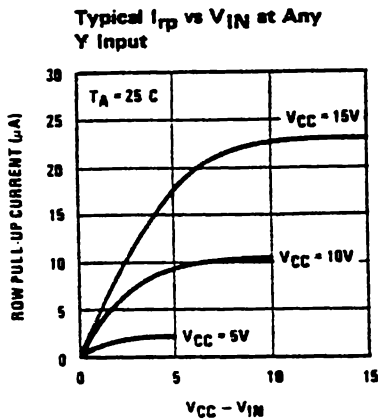


truth table

SWITCH POSITION	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
	Y1,X1	Y1,X2	Y1,X3	Y1,X4	Y2,X1	Y2,X2	Y2,X3	Y2,X4	Y3,X1	Y3,X2	Y3,X3	Y3,X4	Y4,X1	Y4,X2	Y4,X3	Y4,X4	Y5 <sup>0</sup> ,X1	Y5 <sup>0</sup> ,X2	Y5 <sup>0</sup> ,X3	Y5 <sup>0</sup> ,X4
D	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
A	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
B	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0
C	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0
D	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
E <sup>0</sup>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

\*Omit for MM54C922/MM74C922

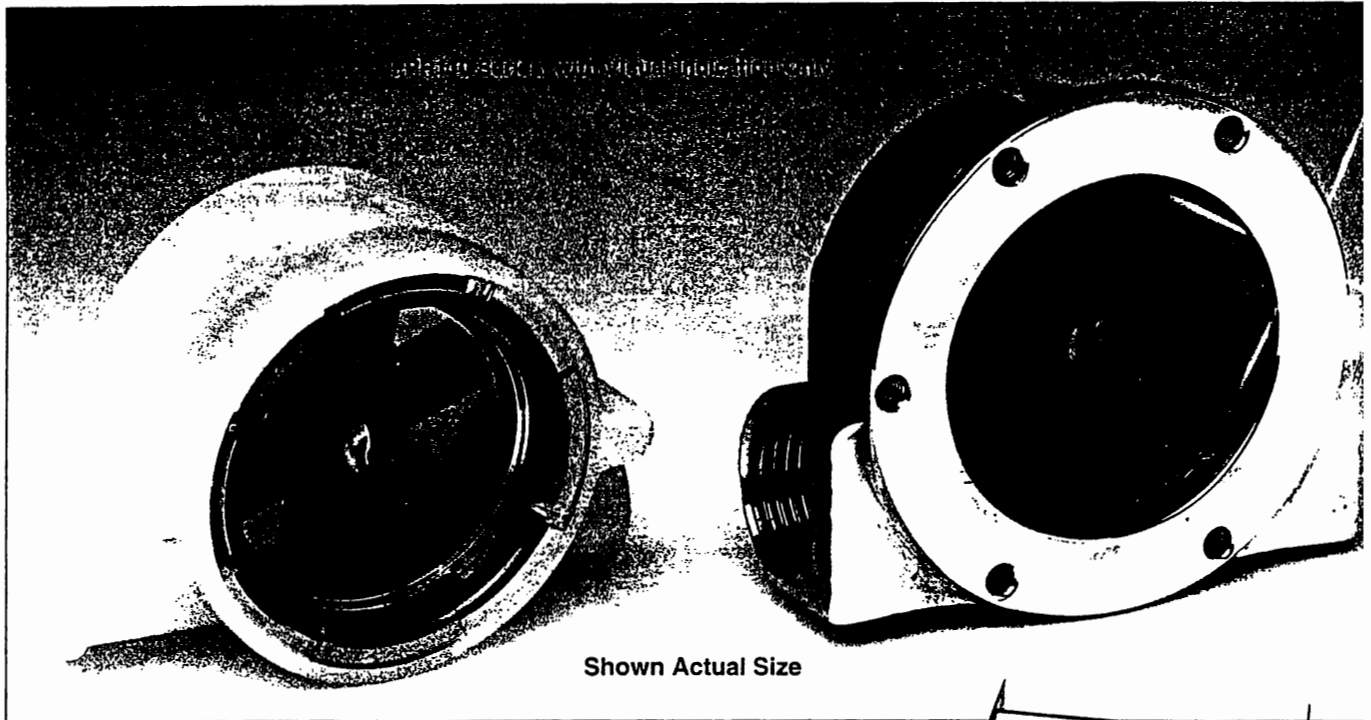
typical performance characteristics







# FLOW SENSORS WITH VISUAL INDICATION, REMOTE SENSING, AND FLOW ALARM



Shown Actual Size

FPR 100 Series

**\$36**

Basic Unit

- ✓ Very Economical
- ✓ Pulse or Alarm Output Optional
- ✓ For Flows from 0.1 to 30.0 GPM
- ✓ Flow Switch Rotor Cannot Seize in ON State
- ✓ Integral Mounting Holes Standard

The FPR100 Series sensors are designed with a one-piece composite rotor, sturdy unibody construction, ceramic shafts, and superior sealing to provide superior durability and economy. The FPR110 Series with bright orange spinning rotor provide visual indication of flow only; the FPR120 Series provide visual indication and an adjustable 0.5 amp @ 110 Vac SPDT mechanical flow switch; the FPR130 Series provide visual indication and a square wave pulse

output for use with remote ratemeters/totalizers, such as the DPF700 series meters.

For panel mounting, all polypropylene units come with mounting ears which accept #8 self-tapping screws; all brass units come with mounting holes which accept #8-32 UNC-2B screws. For all FPR100 sensors, incoming flow may be placed to either port; a minimum of 8 feet of straight inlet pipe is required. Filtration of 150 micron is recommended.

## SPECIFICATIONS

**Wetted Parts:** Brass or hydrolytically stable, glass reinforced polypropylene body; ceramic rotor pin; nylon composite rotor; polysulfone lens; and Buna-N O-ring with polypropylene body, Viton O-ring with brass body

**Max Pressure:**

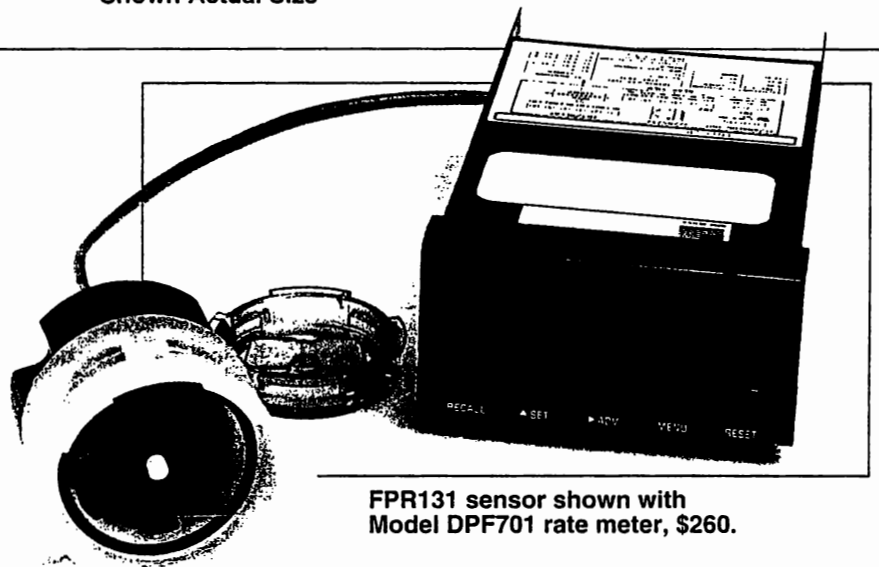
Polypropylene: 100 psig.

Brass: 200 psig

**Max Liquid Temperature:**

FPR120, FPR130: 82.2°C (180°F)

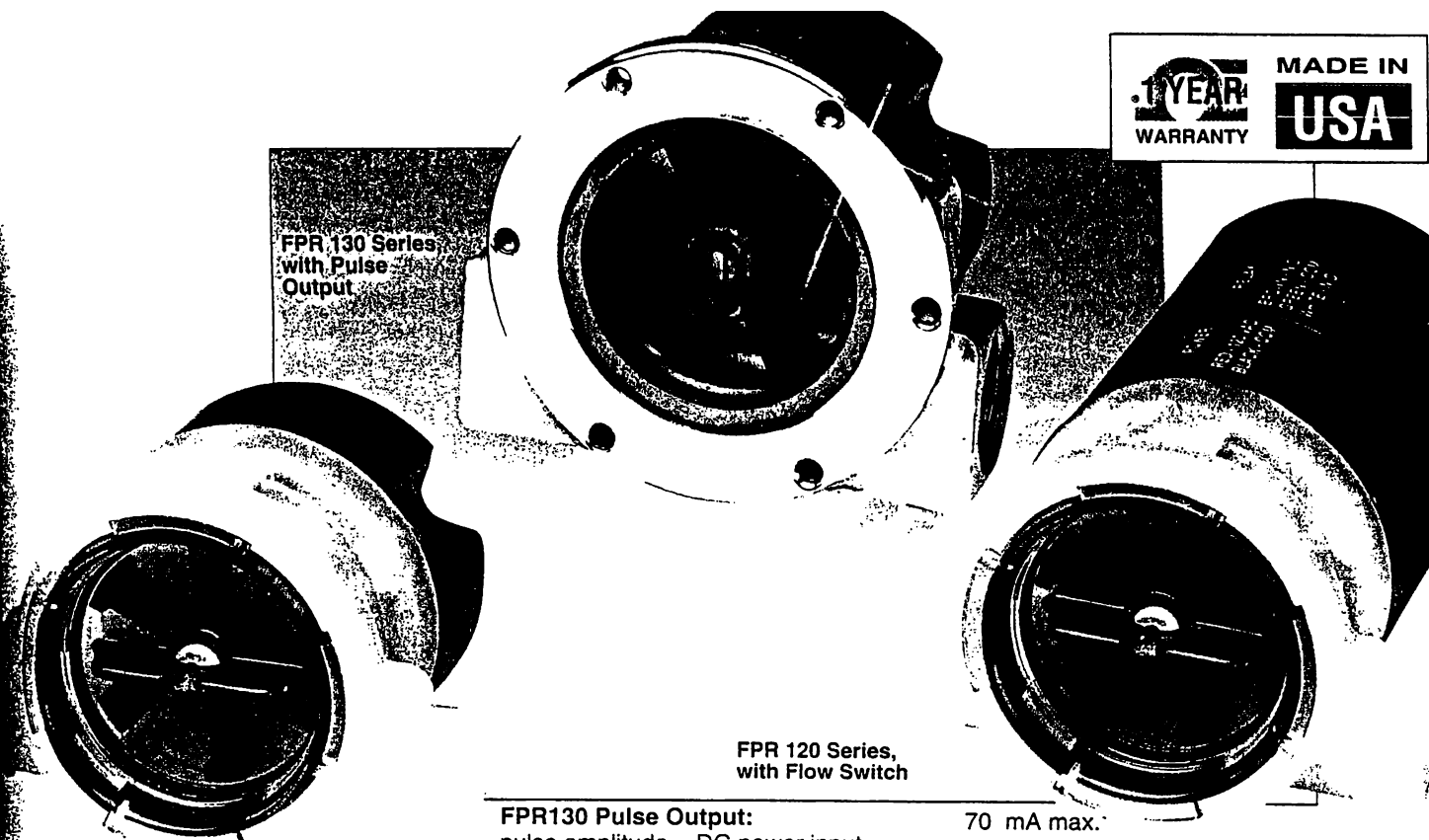
FPR110: 100°C (212°F)



FPR131 sensor shown with Model DPF701 rate meter, \$260.



FPR 130 Series with Pulse Output



FPR 120 Series, with Flow Switch

Max Pressure Drop: 11 psid  
 Max Ambient Temperature:  
 FPR120, FPR130: 65.6°C (150°F)  
 Max Liquid Viscosity: 200 SSU  
 FPR120 Power: 12 Vdc, 24 Vdc, or  
 110 Vac (per model number)

FPR130 Pulse Output:  
 pulse amplitude = DC power input  
 FPR130 Pulse Output Range: 12 Hz  
 @ lowest flowrate of both ranges;  
 225 Hz @ highest flowrate of both  
 ranges  
 FPR120, FPR130 Repeatability:  
 2% of rate  
 FPR130 Power: 4.5 to 24 Vdc @

70 mA max.  
 FPR120 SPDT Switch Rating  
 (resistive): 0.5 amps @ 110 Vac;  
 1 amp @ 24 Vdc  
 FPR120 Setpoint Deadband: 15% of  
 rate  
 Weight: PP body: FPR110, 12 oz;  
 FPR120, 1 lb; FPR130, 14 oz; Brass:  
 FPR110, 1.5 lb, FPR-120, 2 lb

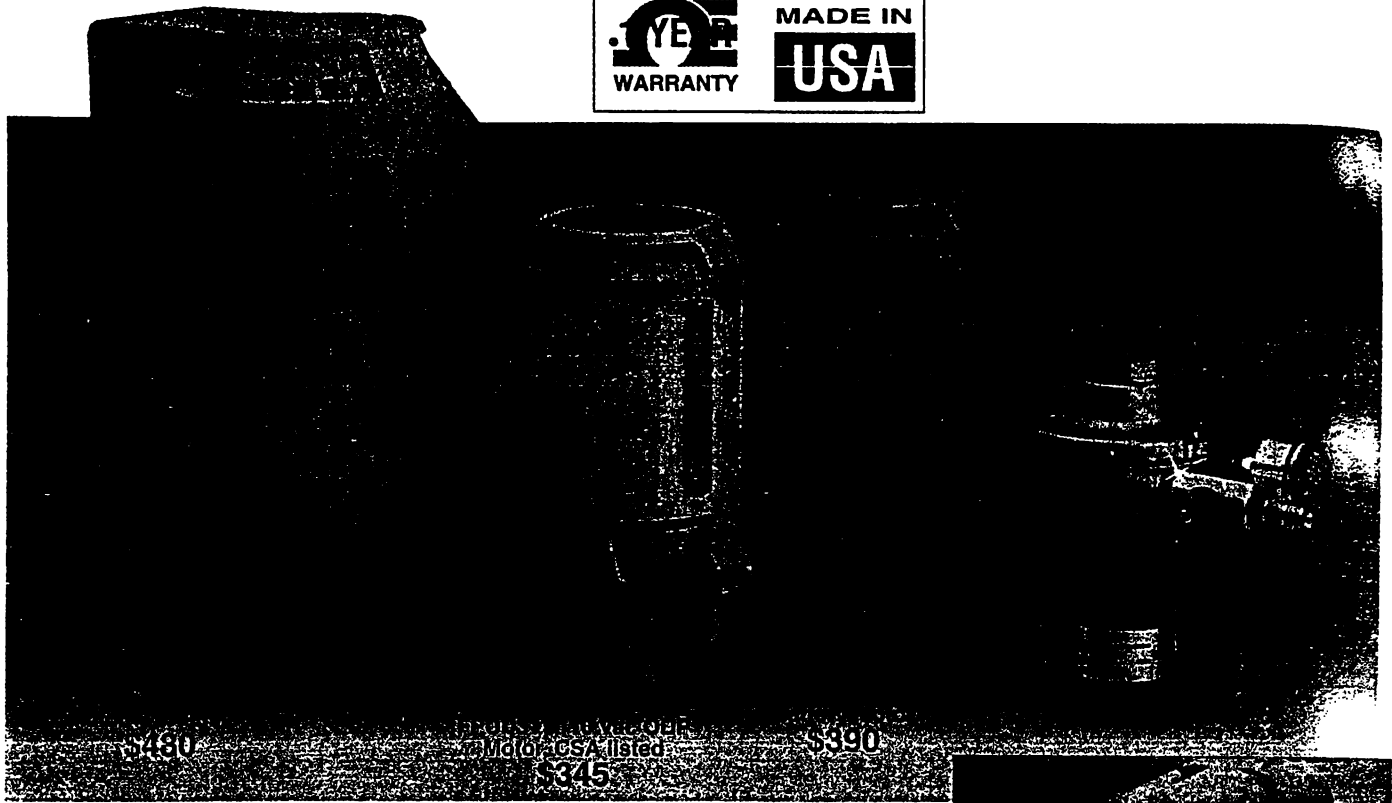
**To Order (Specify Model No.)** **HIGHLIGHTED MODELS IN STOCK FOR FAST DELIVERY!**

Sensor with Visual Indication Only						
Model No.	Price	Body Material	Port Size NPT	Input Power	Flow Ranges, GPM	
					Low* Range	Std. Range
FPR111	\$36	Polypropylene	0.25"	-	0.1 to 1.0	0.5 to 5.0
FPR112	36	Polypropylene	0.50"	-	1.5 to 12.0	4.0 to 20.0
FPR113	62	Brass	0.25"	-	0.1 to 1.0	1.0 to 6.0
FPR114	62	Brass	0.50"	-	1.5 to 12.0	4.0 to 20.0
FPR115	62	Brass	0.75"	-	-	6.0 to 30.0
Sensor with Visual Indication and Flow Switch						
FPR121	\$135	Polypropylene	0.25"	12 Vdc	0.1 to 1.0	0.5 to 5.0
FPR122	135			24 Vdc		
FPR123	160			110 Vac		
FPR124	135		0.50"	12 Vdc	1.5 to 12.0	4.0 to 20.0
FPR125	135		24 Vdc			
FPR126	160		110 Vac			
Sensor with Visual Indication and Pulse Output						
Accuracy (% of Rate)						
FPR131	\$70	Polypropylene	0.25"	-	0.1 to 1.0 (±7.0%)	0.5 to 5.0 (±7.0%)
FPR132	70	Polypropylene	0.50"	-	1.5 to 12.0 (±7.0%)	4.0 to 20.0 (±15.0%)
FPR133	90	Brass	0.25"	-	0.1 to 1.0 (±7.0%)	1.0 to 6.0 (±7.0%)
FPR134	90	Brass	0.50"	-	1.5 to 12.0 (±7.0%)	4.0 to 20.0 (±15.0%)
FPR135	90	Brass	0.75"	-	-	6.0 to 30.0 (±15.0%)

\*With use of low flow adaptor (supplied).  
 All units come with complete operator's manual.  
 Ordering Example: FPR124 flow sensor with visual indication and flow switch, polypropylene body, 0.50" port size, 1.5 to 12.0 GPM low range, 4.0 to 20.0 GPM standard range, powered by 12 Vdc, \$135.

# MOTOR DRUM PUMP

With Corrosion-Resistant Construction



- ✓ **Rugged Industrial Construction**
- ✓ **Wide Variety of Available Motors**
- ✓ **Pump Tubes and Motors Supplied Separately for Maximum Flexibility**
- ✓ **Corrosion-Resistant Polypropylene or 316SS Materials**
- ✓ **Sealless Pump Tubes for Leak-Proof Operation**

OMEGA FPUD300 Series Drum Pumps can empty a wide variety of containers in various applications.

Motor types available are single-phase, 110 or 220 Vac, 50/60 Hz, continuous duty; rechargeable battery operated; and air driven. (NOTE: Only the FPUD331 air motor is intended for pumping flammable or combustible materials.)

Both the FPUD311 and 321 electric motors are open drip-proof (ODP), while the FPUD362 is a totally enclosed fan-cooled (TEFC) motor. All three ac motors are thermal overload protected with a manual reset, and all three include an

internal cooling fan, a built-in on/off switch, a 12-ft, three-wire cord, and are CSA listed (except FPUD311-A, which is not CSA listed). Both 110 Vac motors include a grounded wall plug.

The FPUD350 series include a removable battery pack that can be recharged in the motor or externally. Buy additional battery packs for extended service.

The air operated unit has a 1/4" MNPT fitting for hooking up to a customer-supplied air hose, and includes a flow-regulating valve and muffler. An optional air lubricator/filter assembly is also available.

When pumping flammable or combustible materials or in explosive environments use the FPUD331 air motor with a stainless steel tube.

**CAUTION: DO NOT USE THE FPUD331 AIR MOTOR WITHOUT THE FPUD34-SK STATIC PROTECTION KIT.**

The FPUD3-P tubes include a variable orifice on the outlet to limit flows to 10, 20, 30, or 40 GPM. Refer to the performance curve graphs on next page for flow rate vs. discharge head data, and flow



**FPUD351 Rechargeable Battery Motor Shown with 110 Vac Recharger**

rate vs time for rechargeable battery model. All FPUD3 series pump tubes are sealless to insure leakproof operation. See pages L-53 through L-80 for tubing.

## SPECIFICATIONS

**Pump Tube Wetted Parts:** FPUD3-P: Polypropylene, Inconel®, and Teflon®; FPUD3-S: 316SS, Teflon, and Halar®

**Pump Tube Operating Temperature:** Polypropylene: 160°F (71.1°C); 316SS: 220°F (104°C)

### Electric and Air Motors

**Pump Capability:** 40 gpm, 80 ft TD 30-minute run-dry; air 30 gpm, 60 ft

**Max. Fluid Viscosity:** 500 cps, 1.8 specific gravity

### Rechargeable Battery Motor

**Pump Capability:** 9 gpm, 4 ft. TDH

**Max. Fluid Viscosity:** 100 cps, 1.6 specific gravity

## Pump Motors

Highlighted Models Stocked

### To Order (Specify Part Number)

Model No.	Price	Motor Type	Shp. Wt. Lbs.
<b>AC Powered (1/2 HP, 10,000 RPM)</b>			
FPUD311	\$345	110 Vac/ODP/CSA	8
FPUD311A	300	110 Vac/ODP/non-CSA	8
FPUD321	300	220 Vac/ODP/CSA	8
FPUD362	480	110 Vac/TEFC/CSA	11
<b>Rechargeable Battery (1/2 HP, 4000 RPM)</b>			
FPUD351	390	Rchg Batt/ODP w/Recharger For 110 Vac	7
FPUD352	415	Rchg Batt/ODP w/Recharger For 220 Vac	7
<b>Air Powered (requires 10-15 cfm @ 50 to 90 psig; 10,000 RPM)</b>			
FPUD331	375	Air Motor	5

## Accessories for Pump Motors

Model No.	Price	Description
FPUD35-BP	\$107	Spare Rechargeable Battery Pack for FPUD351, 352
FPUD35-BC	34	Spare Battery Charger - 110 Vac for FPUD351
FPUD352-BC	66	Spare Battery Charger - 220 Vac for FPUD352
FPUD345-FL	107	Filter-Lubricator Assembly for FPUD331
FPUD34-SK	220	Static Protection Kit for FPUD331

## Pump Tubes

Highlighted Models Stocked

### To Order (Specify Model Number)

Model No.	Price	Tube Outlet Connection	Length Inches	Ship. Wt. Lbs.
<b>Polypro (2" O.D.)</b>				
FPUD3-P27	\$235	Both 1" I.D. Hose and 1 1/4" MNPT	27	6
FPUD3-P36	245		36	6
FPUD3-P48	270		48	7
<b>316SS (2" O.D.)</b>				
FPUD3-S27	380	1" I.D. Hose Only	27	12
FPUD3-S36	425		36	14
FPUD3-S48	470		48	17

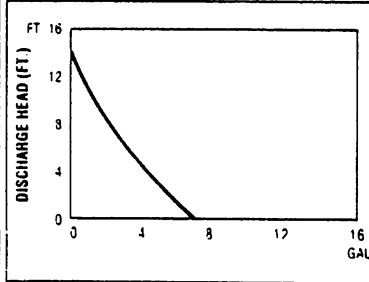
## Accessories for Pump Tubes

Model No.	Price	Description
FPUD3-SDA	\$115	316SS Drum Adaptor with 2" IPS thread
FPUD3-PDA	44	Polypropylene Drum Adaptor with 2" IPS thread
FPUD3-WB	22	Wallmount Bracket (grips tube)
FPUD3-PS	28	Polypropylene Inlet Strainer for FPUD3-P series ONLY

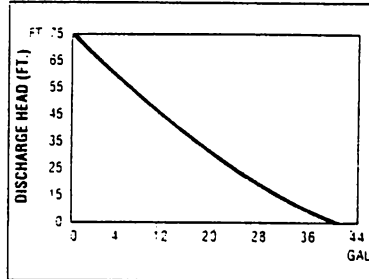
Ordering Example: FPUD362, 110 Vac TEFC powered pump motor with a FPUD3-S36 316 stainless steel pump tube, \$480 + 425 = \$905.

## PERFORMANCE CURVES

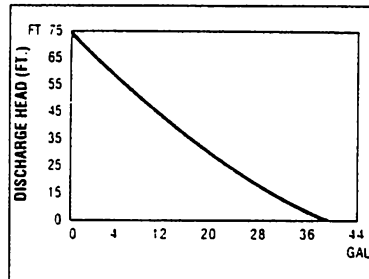
### Air Motor



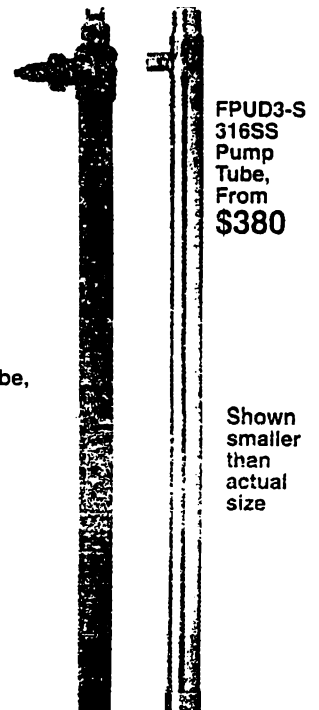
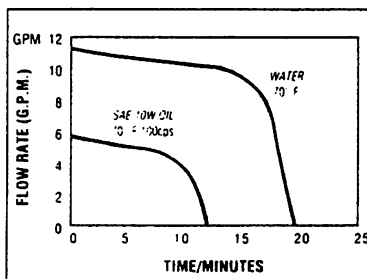
### ODP Electric Motor



### TEFC Electric Motor



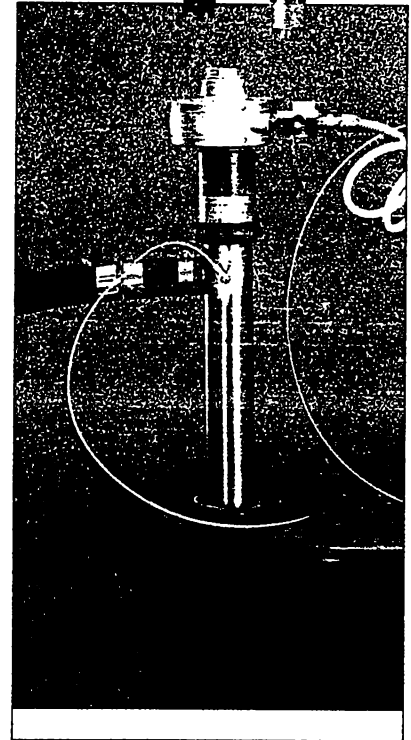
### Rechargeable Battery Motor



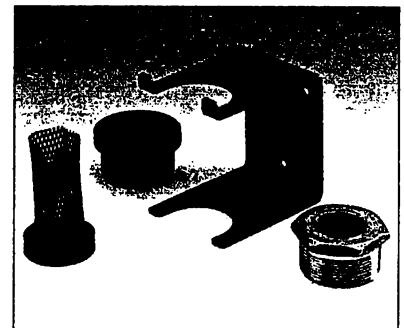
FPUD3-S  
316SS  
Pump  
Tube,  
From  
\$380

FPUD3-P  
Polypro  
Pump Tube,  
From  
\$235

Shown smaller than actual size



FPUD34-SK Static Protection Kit  
Shown with FPUD3-S Pump Tube \$220



Drum Adaptors, Wall Mount, Bracket and Strainer Accessories for Pump Tubes

**APÉNDICE B**

## COMUNICACIÓN SERIAL

El compartir datos entre varios dispositivos, es un requerimiento imprescindible en cualquier sistema controlador actual. Pero es necesario encaminarlos en características estándares internacionales con el fin de unificar criterios y permitir la interconectividad con cualquier otro dispositivo comercial o experimental.

Se ha explicado como realizar comunicación mediante el standard serial RS-232C. Pero existen requerimientos industriales los cuales se caracterizan por ser ambientes con alto grado de interferencias y ruidos de toda índole. Es por lo que la interface anterior no satisface lo difícil del medio industrial. Para ello se utiliza el standard recomendado por la EIA, la interface RS-485.

La transmisión de datos entre sistemas con computadores y periféricos a lo largo de grandes distancias, bajo condiciones de mucho ruido, usualmente prueba lo difícil que es la comunicación entre receptores y transmisores de un solo fin (single-ended). Para ello, las recomendaciones de la EIA (Electronic Industries Association = Asociación de Industrias Electrónicas) en interfaces digitales de voltaje balanceado proveen un diseño con una solución universal para sistemas con requerimientos de grandes distancias. La habilidad de acoplar voltajes de modo común permite a un receptor diferencial (balanceado) obtener y reproducir señales que de otra manera serían inutilizables. Por lo que a un sistema balanceado de transmisión de datos, es posible incorporarle un manejador de transmisión diferencial en interconexión de líneas balanceadas con un receptor que tenga entradas diferenciales.

En el siguiente cuadro se presentan comparaciones de las interfaces RS-232C y la RS-485:

<b>PARÁMETROS</b>	<b>RS-232C</b>	<b>RS-485</b>
Modo de operación	Un extremo	Diferencial
Numero de manejadores y receptores permitidos	1 Manejador 1 Receptor	32 Manejadores 32 Receptores
Máxima longitud de cable	15 mtr (50')	1219 mtr (4000')
Máximo baudrate	20 kbaudios	10 Mbaudios
Máximo voltaje de modo común	$\pm 25$ V	12V / -7V
Salida del manejador	$\pm 5$ Vmin $\pm 15$ Vmax	$\pm 1.5$ Vmin
Carga del manejador	3k hasta 7k $\Omega$	60 $\Omega$ min
Razón de cambio en manejador (slew rate)	30V/ $\mu$ s max	NA
Salida limite del manejador en cortocircuito	500 mA a Vcc o tierra	150 mA a tierra 250mA a 8-12V

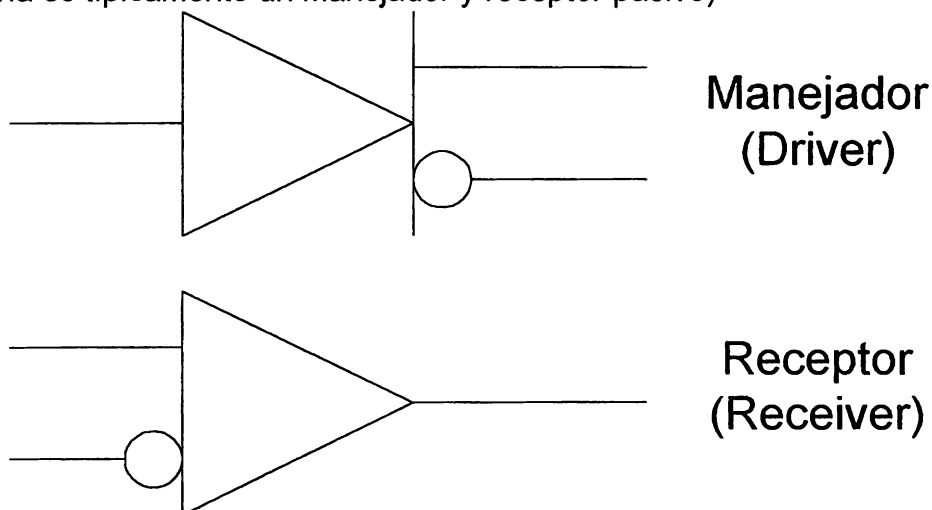
Resistencia salida del manejador	On	NA	120 k $\Omega$
	Off	300 $\Omega$	120 k $\Omega$
Resistencia del entrada del receptor		3k hasta 7k $\Omega$	12 k $\Omega$
Sensibilidad del receptor		$\pm 3$ V	$\pm 200$ mV

El estándar RS-485, introducida en 1983, es una mejora de la versión de la interface EIA RS-422A. Incrementa el uso de líneas de transmisión balanceadas en un solo par de líneas.

EIA RS-485 toma en consideración las características y requerimientos de la RS-422A para transmisión en líneas balanceadas, además, ventajas que permiten conectar múltiples receptores y transmisores..

El estándar RS-485 difiere de la RS-422A primeramente con permitir mayor número de puntos de comunicación. Entre otras ventajas que presenta están:

- Un manejador (Driver) puede interconectarse con 32 cargas unitarias (U. L.= Unit loads) y con un total de resistencia terminal de 60  $\Omega$  o más (una carga unitaria es típicamente un manejador y receptor pasivo)

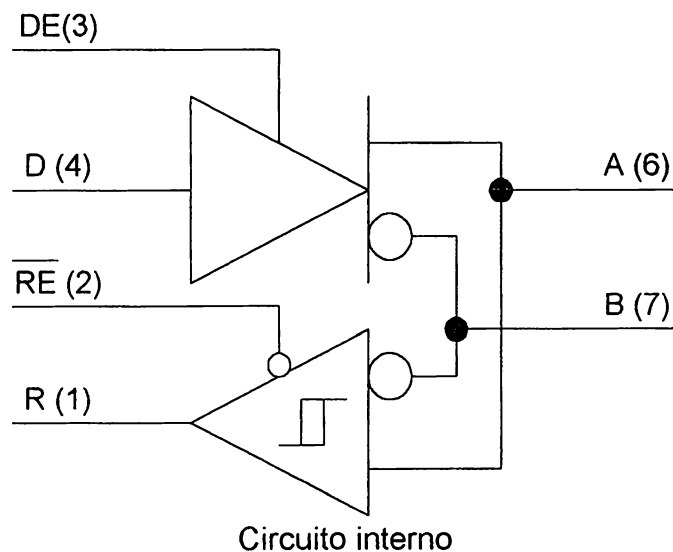
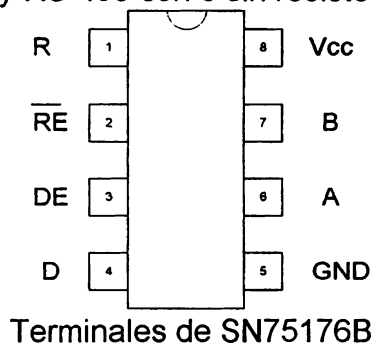


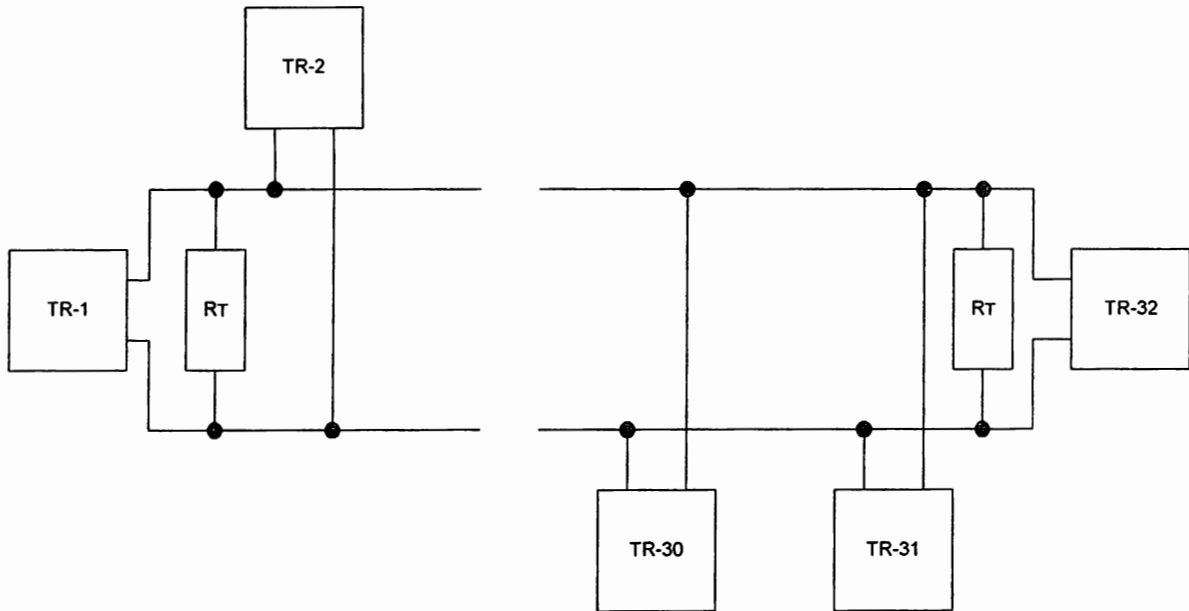
- La salida del manejador, estado de apagado y dispersión de corriente debe ser de 100  $\mu$ A o menos con cualquier línea de voltaje de -7 a 7 V.
- El manejador tiene capacidad de proveer un voltaje de salida diferencial de 1.5 hasta 5 V con líneas de mono común con voltaje desde -7 hasta 12 V.
- El manejador de tener protección interna contra la contienda (múltiples manejadores pretendiendo transmitir al mismo tiempo).
- La resistencia en alto a la entrada del receptor debe ser mínimo de 12 k $\Omega$ .
- El rango de recepción a la entrada en modo común debe ser de -7 hasta 12 V.
- La sensibilidad a la entrada diferencial de  $\pm 200$  mV en un rango de modo común de -7 hasta 12 V.

Los transreceptores son dispositivos que contienen tanto el receptor como el manejador, para el acople de comunicación mediante el sistema mínimo con microcontrolador se selecciona el IC SN75176B, las ventajas que este permite son:

1. Cumple los estándar EIA RS-485, RS-422A, CITT V.11 y X.27
2. Manejador de tres estados
3. Amplio rango de voltaje positivo y negativo para el bus.
4. Protección térmica.
5. Limitador de corriente
6. Resistencia del receptor mínima de 20 k $\Omega$ .

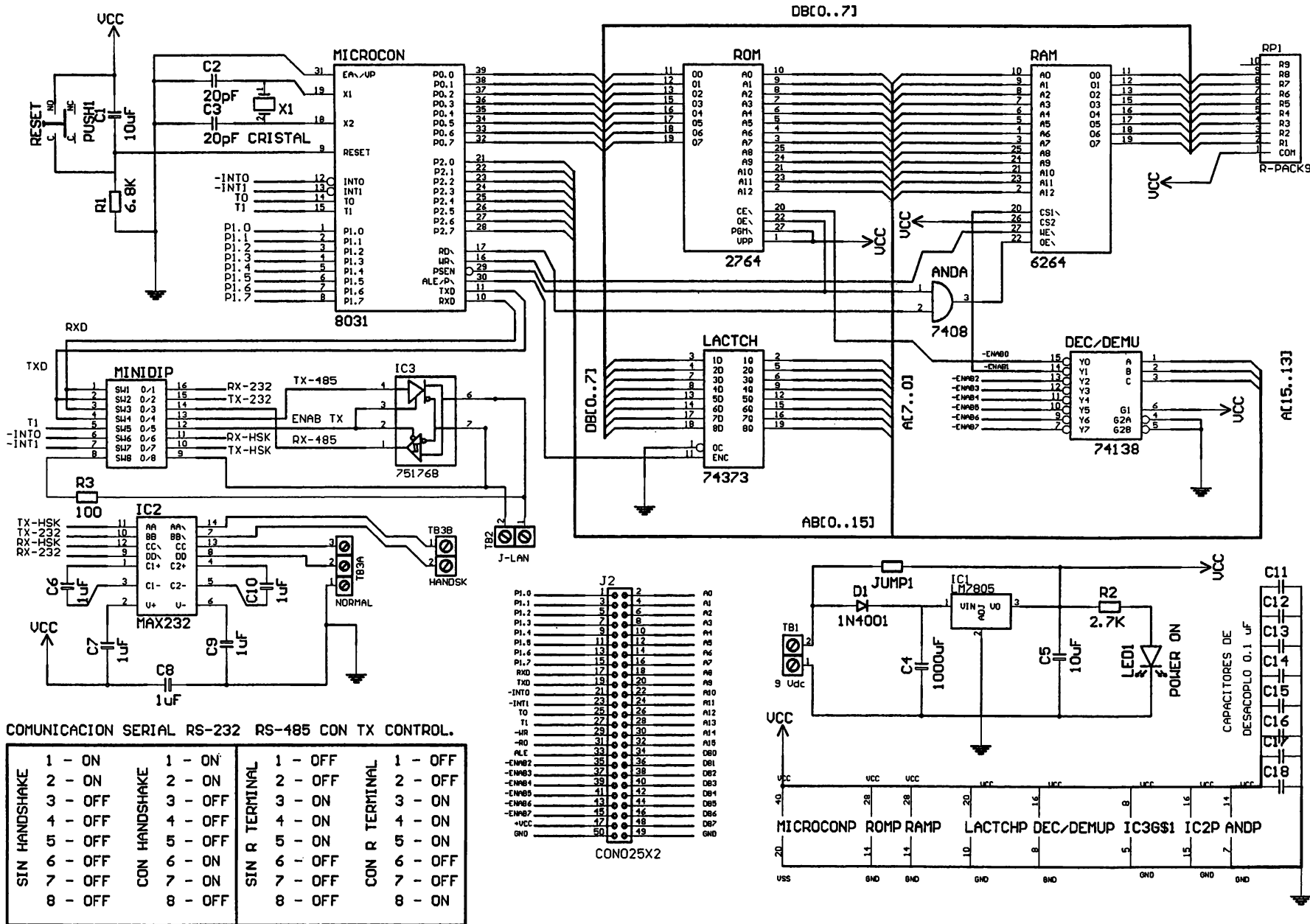
Es necesario aclarar que para cualquier aplicación se tenga presente tanto si se presenta resistencia terminal o no, velocidad de comunicación, etc. A continuación se presenta los terminales del IC, su circuitería interna y como se interconectaría el SN75176 en una red RS-485. Además se presenta un diagrama completo de la conexión del mismo con el microcontrolador, presentando RS-232C con o sin handshake y RS-485 con o sin resistencia terminal.





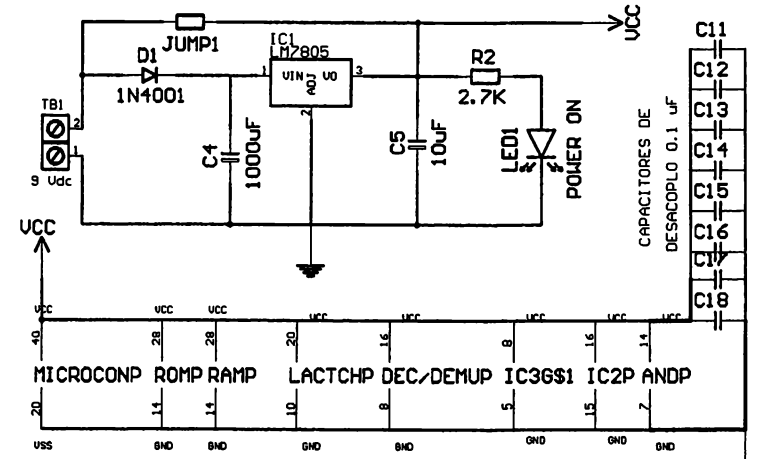
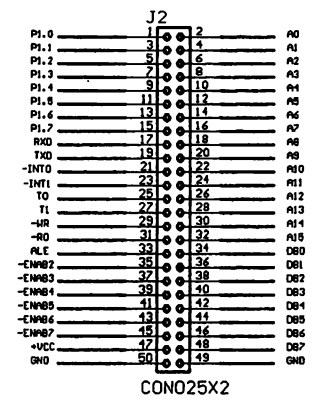
Transreceptor SN75176B en un Bus multiestación  
Diagrama de conexión con el sistema mínimo DTK-51:

A continuación se presenta la conexión del SN176B, un dip switch permite utilizar la interface RS-232C (con o sin handshake) y RS-485 (con o sin resistencia terminal).



COMUNICACION SERIAL RS-232 RS-485 CON TX CONTROL.

SIN HANDSHAKE	1 - ON	2 - ON	3 - OFF	4 - OFF	5 - OFF	6 - OFF	7 - OFF	8 - OFF
CON HANDSHAKE	1 - ON	2 - ON	3 - OFF	4 - OFF	5 - OFF	6 - ON	7 - ON	8 - OFF
SIN R TERMINAL	1 - OFF	2 - OFF	3 - ON	4 - ON	5 - ON	6 - OFF	7 - OFF	8 - OFF
CON R TERMINAL	1 - OFF	2 - OFF	3 - ON	4 - ON	5 - ON	6 - OFF	7 - OFF	8 - ON



## USO DE RAM NO VOLATIL

```

;*****
;**          EJEMPLO DE USO DE nvSRAM X244          **
;** Contiene subrutina tanto para salvar y recobrar **
;** datos entre la ram y el microcontrolador MC-51 **
;*****
;///////// Definicion de variables y registros ///////////
RECALL EQU 85H ;RECALL INSTRUCCION
WRITE EQU 84H ;WRITE ENABLE INSTRCC
STORE EQU 81H ;STORE INSTRUCCION
SLEEP EQU 82H ;SLEEP INSTRUCCION
W_DATA EQU 83H ;WRITE DATA INSTRUCCN
R_DATA EQU 87H ;READ DATA INSTRUCCIN
LENGTH EQU 32 ;32 BYTES MAXIMO
DIN REG P1.2 ;LINES DE INTERFACE
ENABLE REG P1.1
CLOCK REG P1.0
RAM CODE 8000H ;MAPEO
DSEG AT 60H
NVRAM DS LENGTH
CSEG AT RAM ;60-7FH EN RAM 8031
;SERA NECESARIO ACOMODAR LAS SUBRUTINAS
;SEGUN NECESIDADES DEL PROGRAMADOR
;+++++
;++ SUBROUTINA PARA SALVAR DATOS DESDE 60H ++
;++ HASTA 7FH EN MEMORIA INTERNA DE 8031 (0 ++
;++ MODELO SELECCIONADO MCS-51) PARA LA X2444++
;+++++
SAVE: MOV R0,#NVRAM ;R0 COMO PUNTERO
      CLR ENABLE ;DISABLE PARA X2444
      MOV A,#RECALL ;INSTRUCCION RECALL
      SETB ENABLE
      CALL W_BYTE
      CLR ENABLE
      MOV A,#WRITE ;INSTRUCCION WRITE
      SETB ENABLE ;ENABLE PREPARA
      CALL W_BYTE ;X2444 PARA ESCRTRA
      CLR ENABLE
      MOV R7,#00H ;INICIALIZAR R7
AG: MOV A,R7 ;FIJAR DIRECCION AC
    RL A ;DIRECCION EN BITS
    RL A ;3, 4, 5, 6
    RL A
    ORL A,#W_DATA BUILD WRITE
    SETB ENABLE
    CALL W_BYTE
    MOV R5,#02H
LOOP: MOV A,@R0 ;OBTENER DATO 8031

```

```

INC R0                ;SIGUIENTE DATO
CALL W_BYTE          ;ESCRIBIR EN X2444
DJNZ R5, LOOP
CLR ENABLE
INC R7                ;INCREMENTAR DIRECC
CJNE R7, #16D, AG    ;HACERLO 16 VECES
MOV A, #STORE
SETB ENABLE
CALL W_BYTE
CLR ENABLE
MOV A, #SLEEP        ;X2444 DESACTIVADA
SETB ENABLE
CALL W_BYTE
CLR ENABLE
RET

;+++++
;++ SUBROUTINA PARA RECOBRAR DATOS DE X2444 ++
;++ AL MICROCONTROLADOR EN LAS LOCALIDADES ++
;++ DE MEMORIA INTERNA 60H HASTA 7FH ++
;+++++
RECOVER: MOV R0, #NVRAM
CLR ENABLE
MOV A, #RECALL      ;RETOMAR INSTRUCCN
SETB ENABLE
CALL W_BYTE
CLR ENABLE
AN: MOV R7, #00H     ;R7 DIRECCION X2444
MOV A, R7           ;PONER DIRECC EN AC
RL A                ;BUILD READ
RL A
RL A
ORL A, #R_DATA
SETB ENABLE
CALL W_BYTE        ;MANDO COMANDO READ
MOV R5, #02H
LOOP1: CALL R_BYTE   ;LEER BYTE
MOV @R0, A         ;PONER EN RAM 8031
INC R0             ;PROXIMA DIRECCION
DJNZ R5, LOOP1
CLR ENABLE
INC R7
CJNE R7, #16D, AN
MOV A, #SLEEP
SETB ENABLE
CALL W_BYTE
CLR ENABLE
RET

;+++++
;++ LECTURA DE UN BYTE DE DATOS DEL X2444 ++
;+++++

```

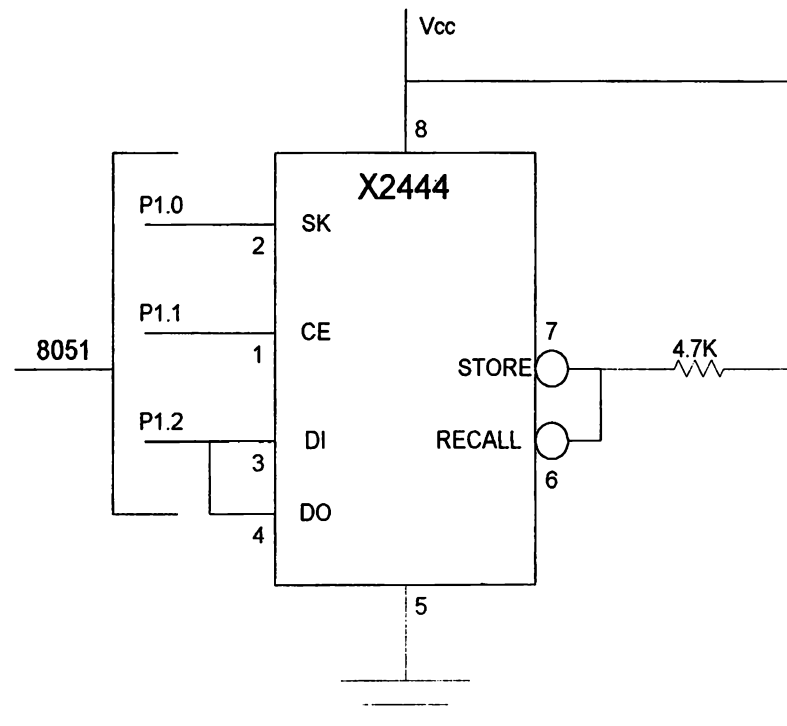
```

R_BYTE:  MOV R6,#08H      ;R6 CONTADOR DE BIT
AGAIN:   MOV C,DIN        ;DATA BIT EN CARRY
         RLC A            ;RECONSTRUIR DATO
         SETB CLOCK
         CLR CLOCK
         DJNZ R6,AGAIN
         RET

;+++++
;++ ESCRITURA DE UN BYTE DE DATO AL X2444 ++
;+++++
W_BYTE:  MOV R6,#08H      ;R6 CONTADOR DE BIT
AGAIN2:  RLC A            ;DATA BIT EN CARRY
         MOV DIN,C        ;PON DATO EN LINEA
         SETB CLOCK
         CLR CLOCK
         DJNZ R6,AGAIN2
         RET
END      ;FIN

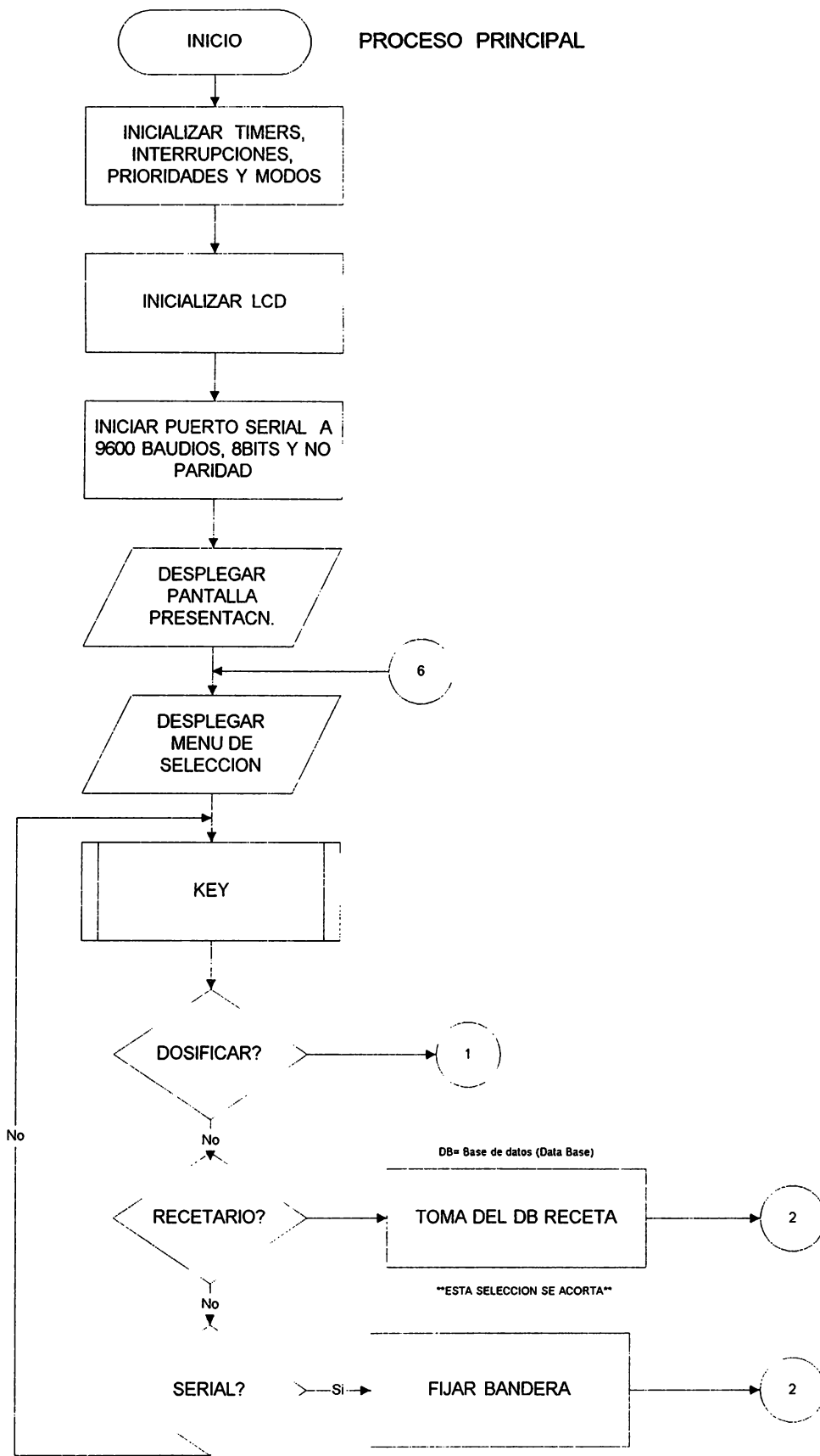
```

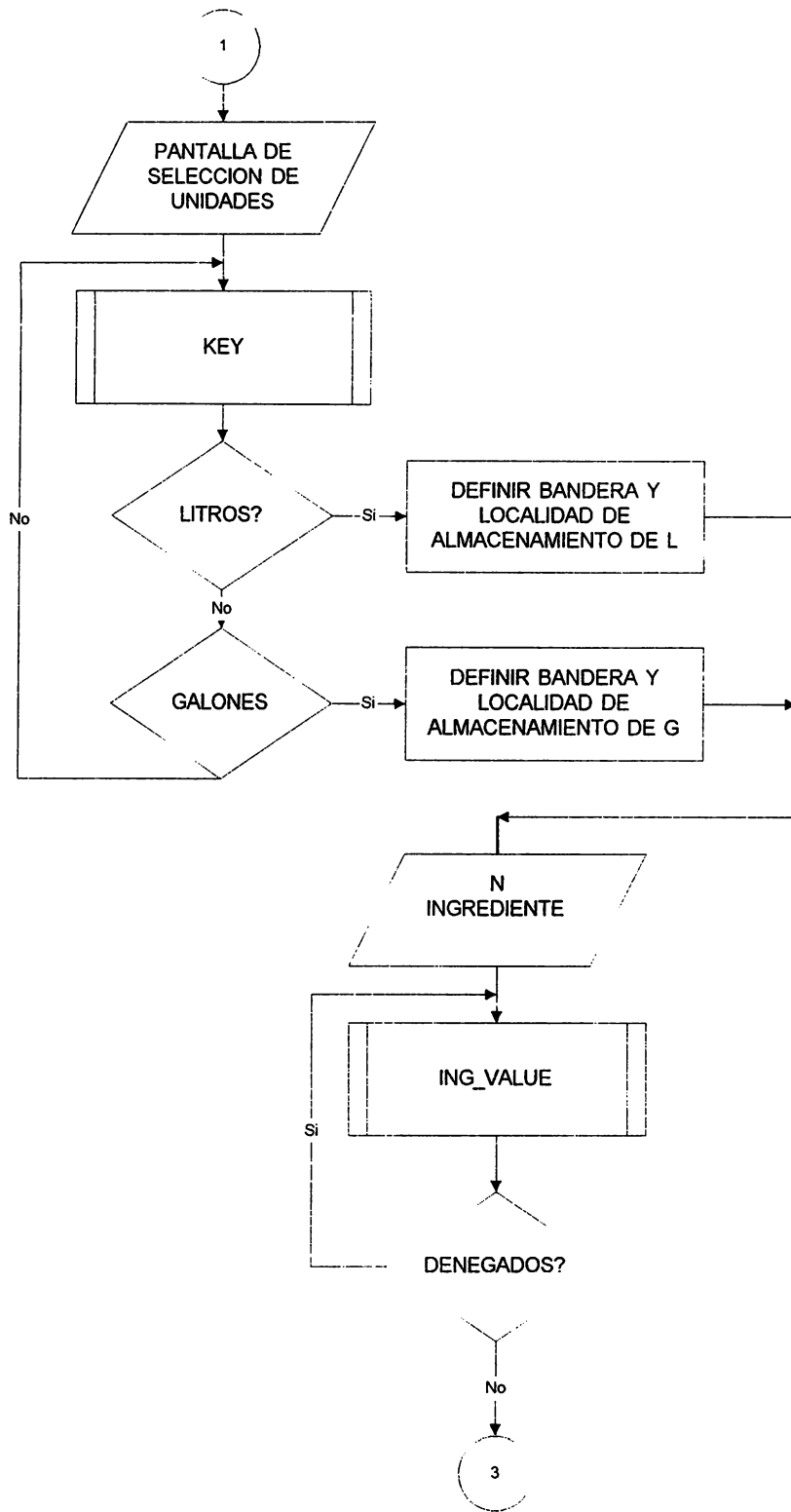
El programa anterior permite programar la nvSRAM x2444, es necesario aclarar que el programa sirve para dar una guía para poder accederla. A continuación se presenta un diagrama de conexión del microcontrolador a la RAM.

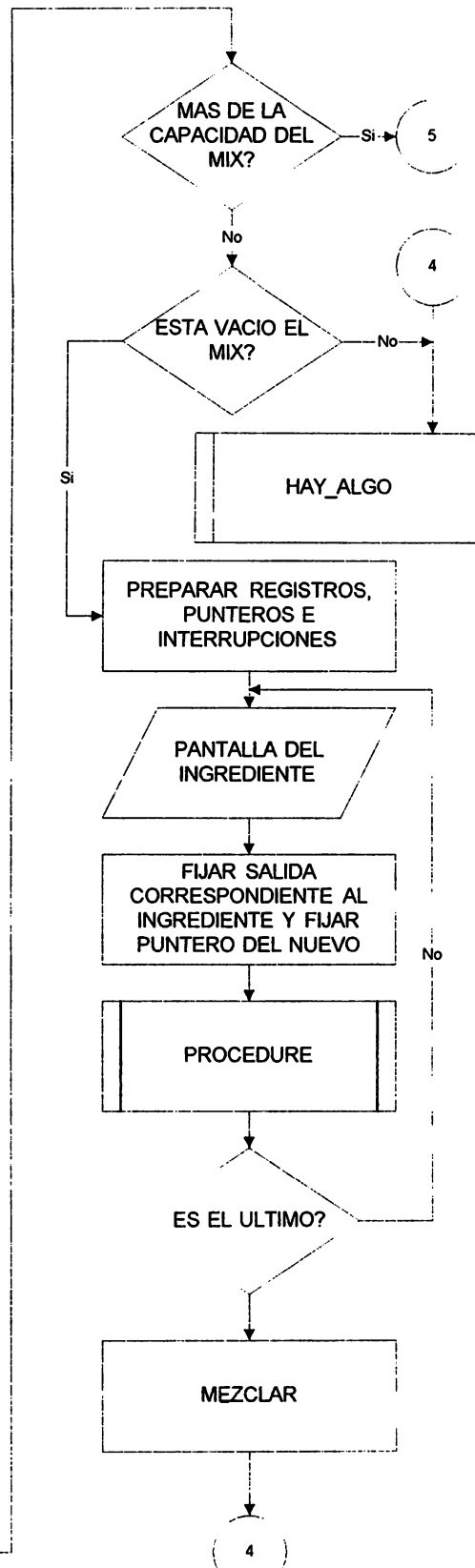
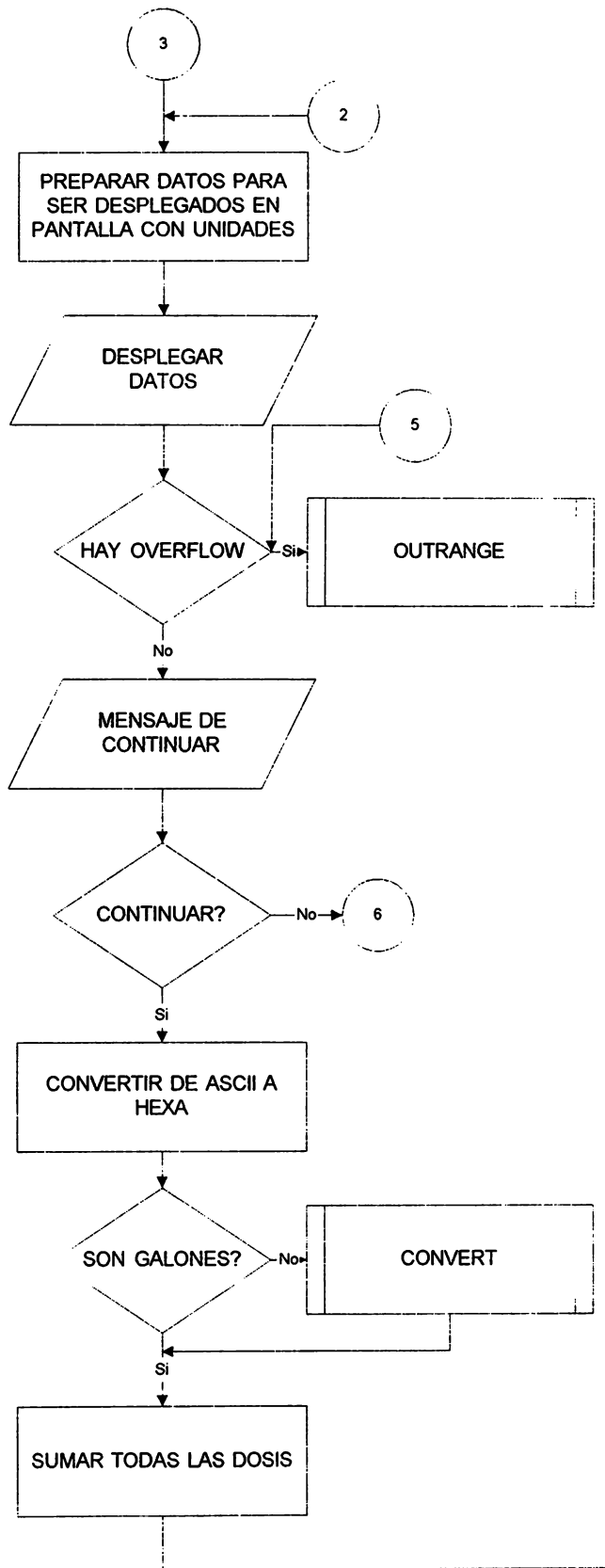


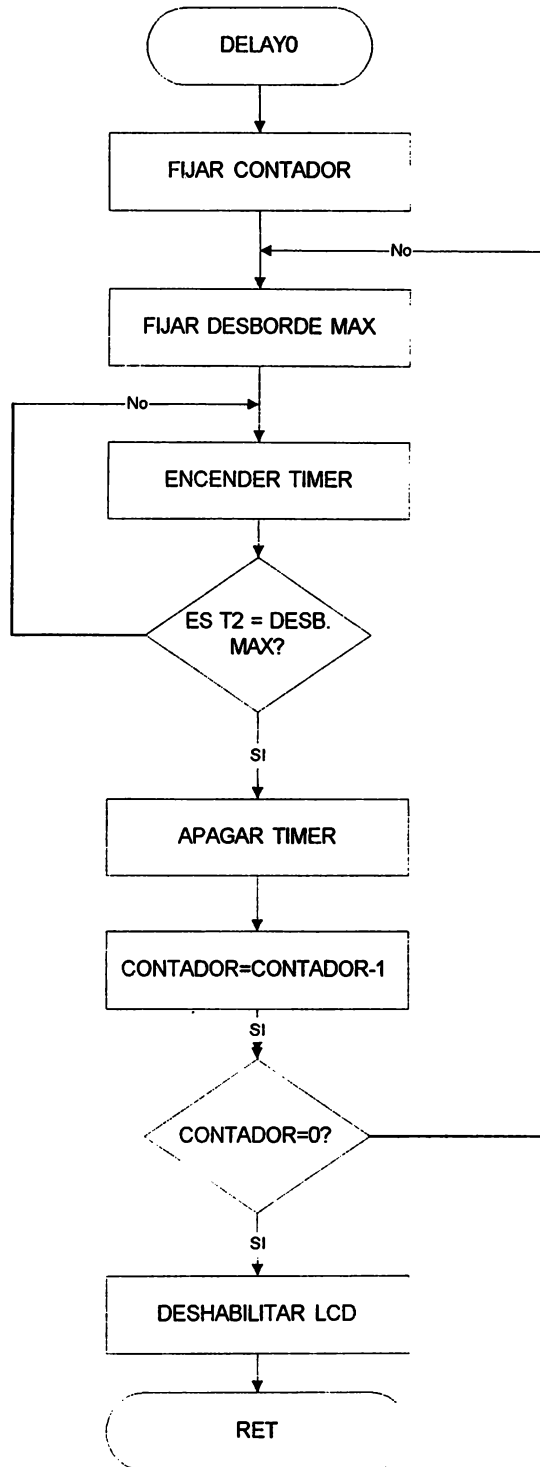
**APÉNDICE C**

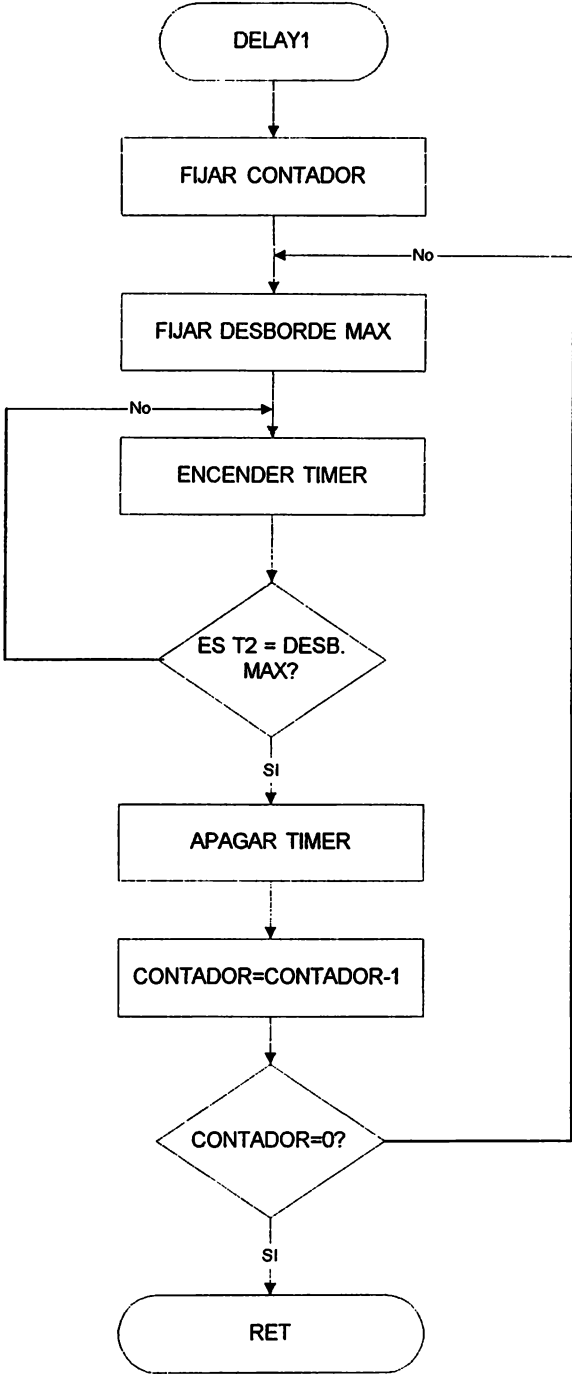
PROCESO PRINCIPAL



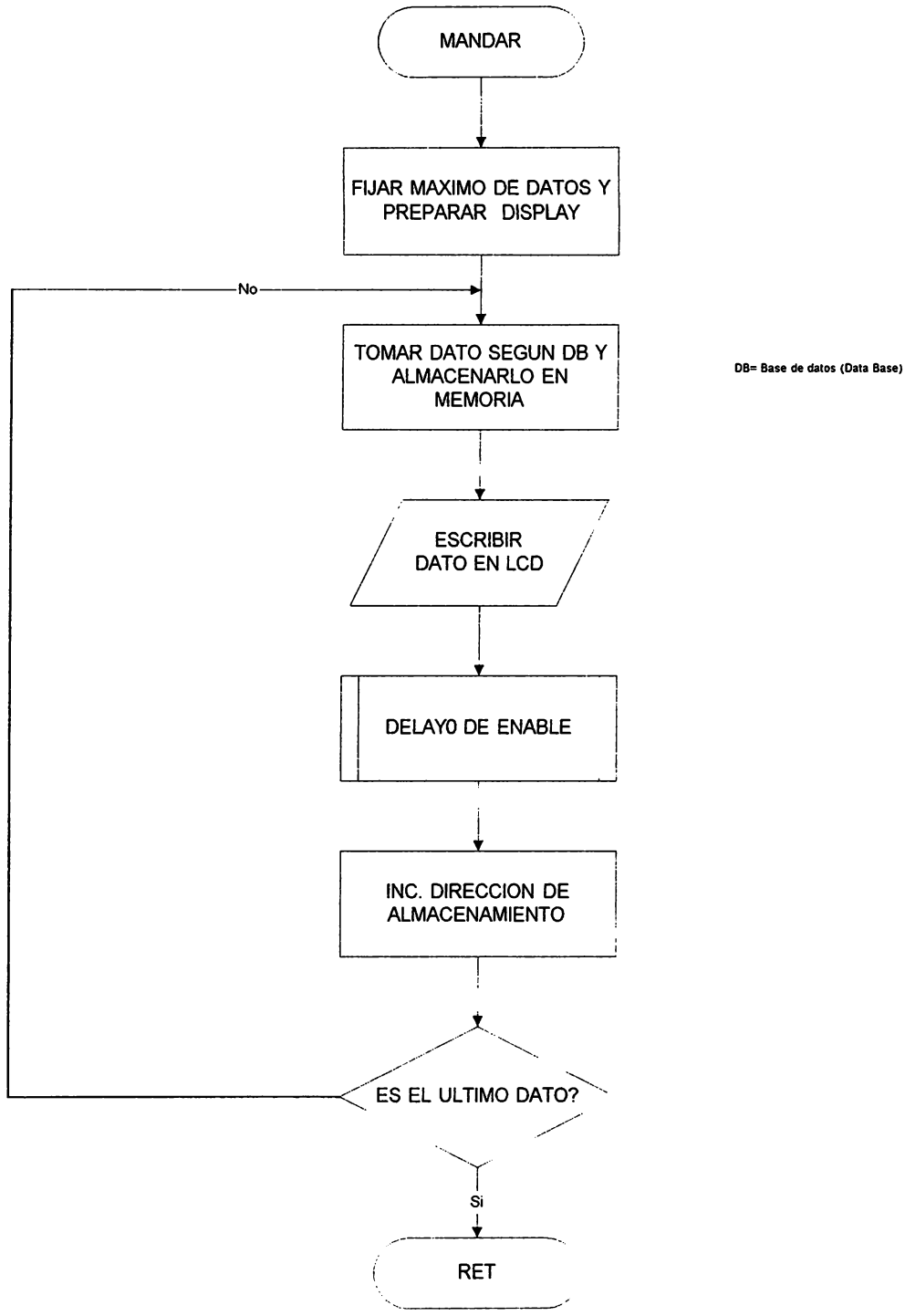




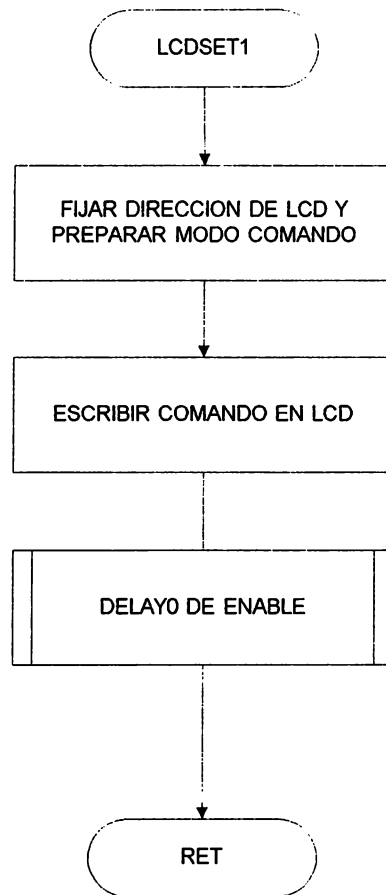




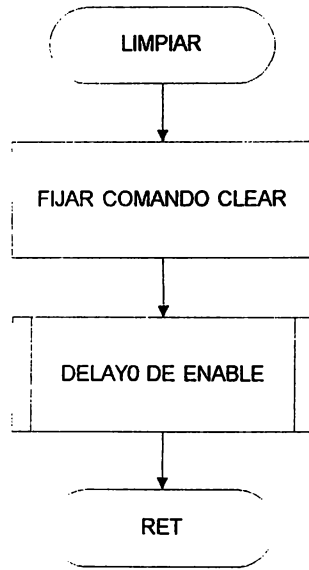
SUBROUTINA MANDAR



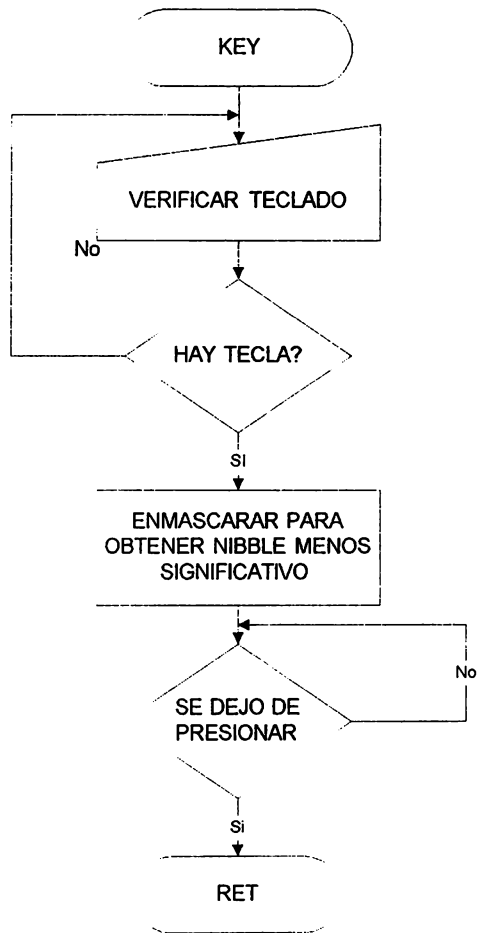
## SUBROUTINA LCDSET1



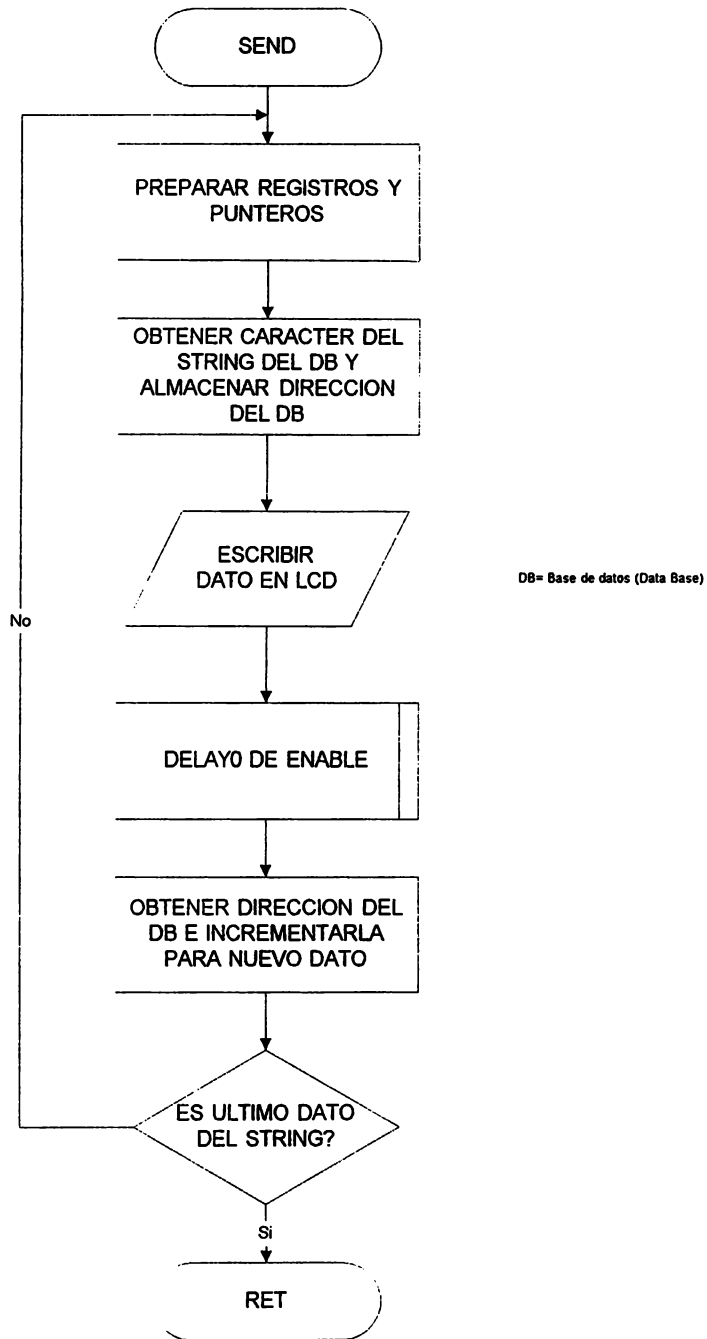
SUBROUTINA LIMPIAR



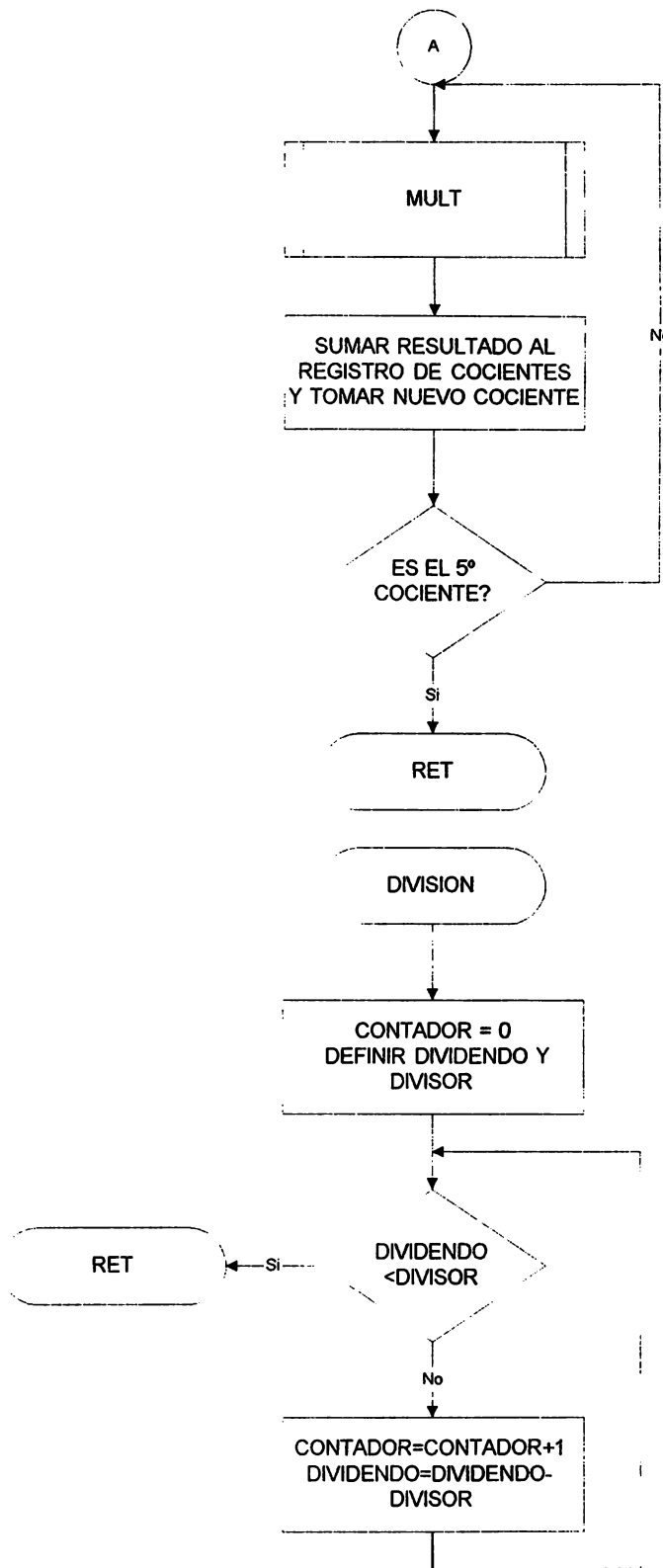
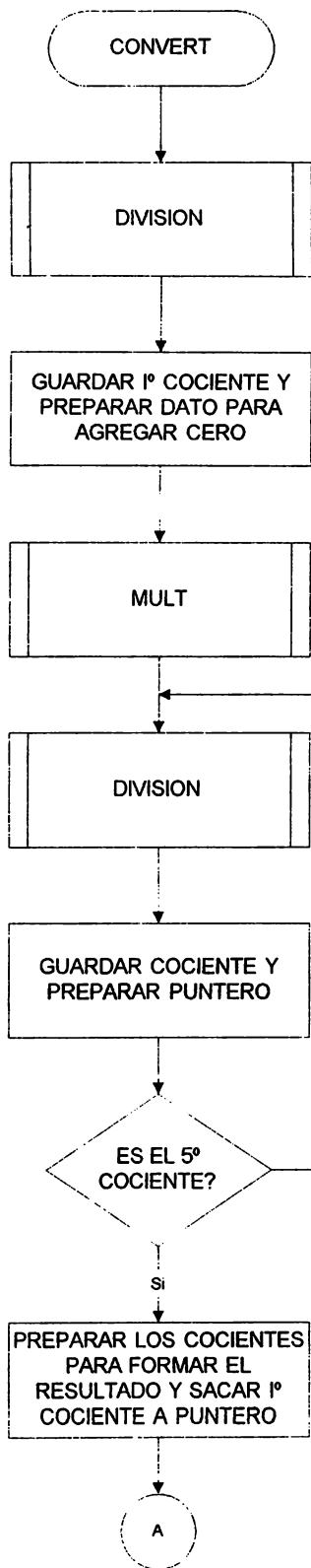
SUBROUTINA KEY

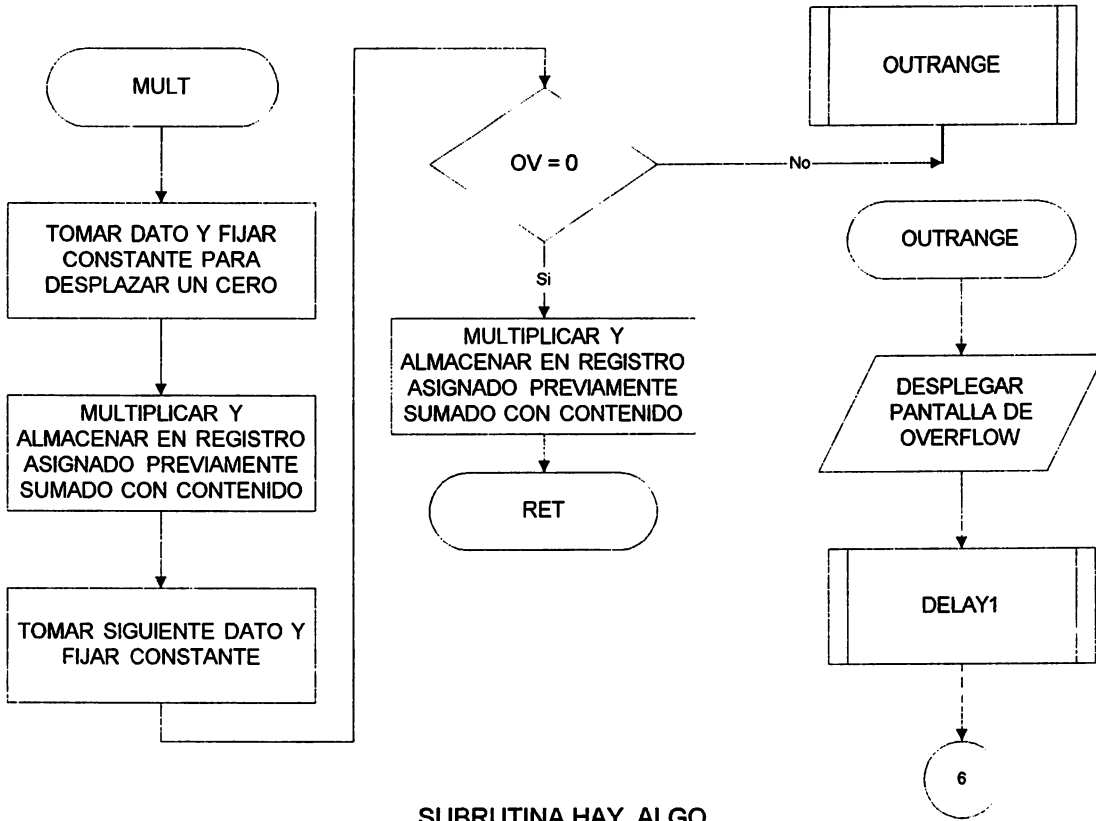


# SUBROUTINA SEND

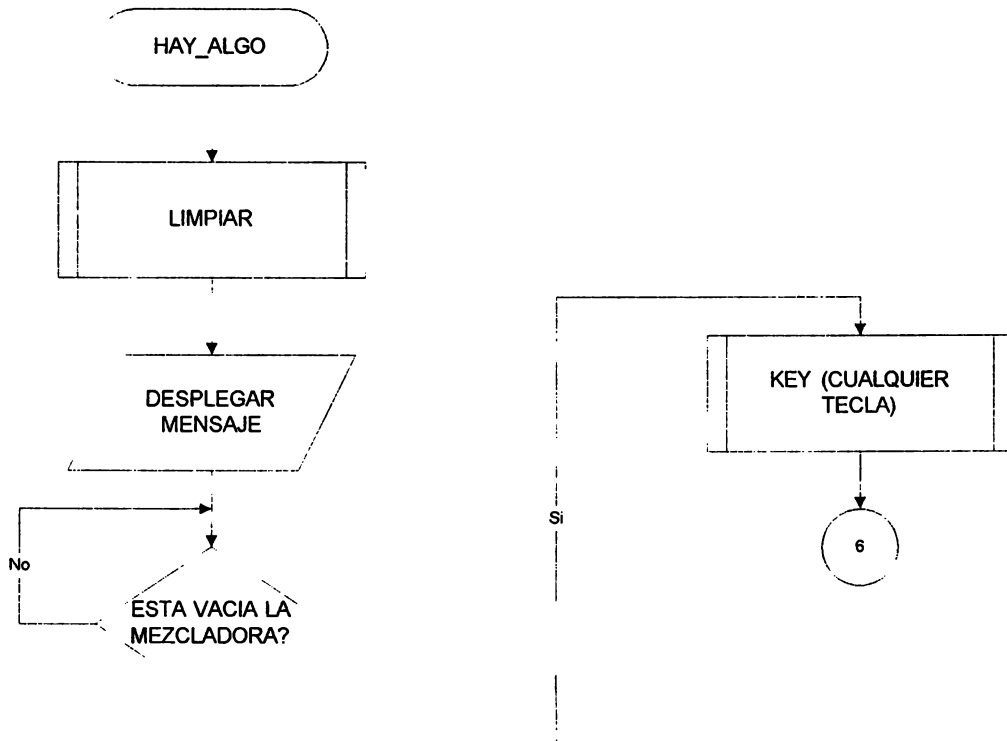


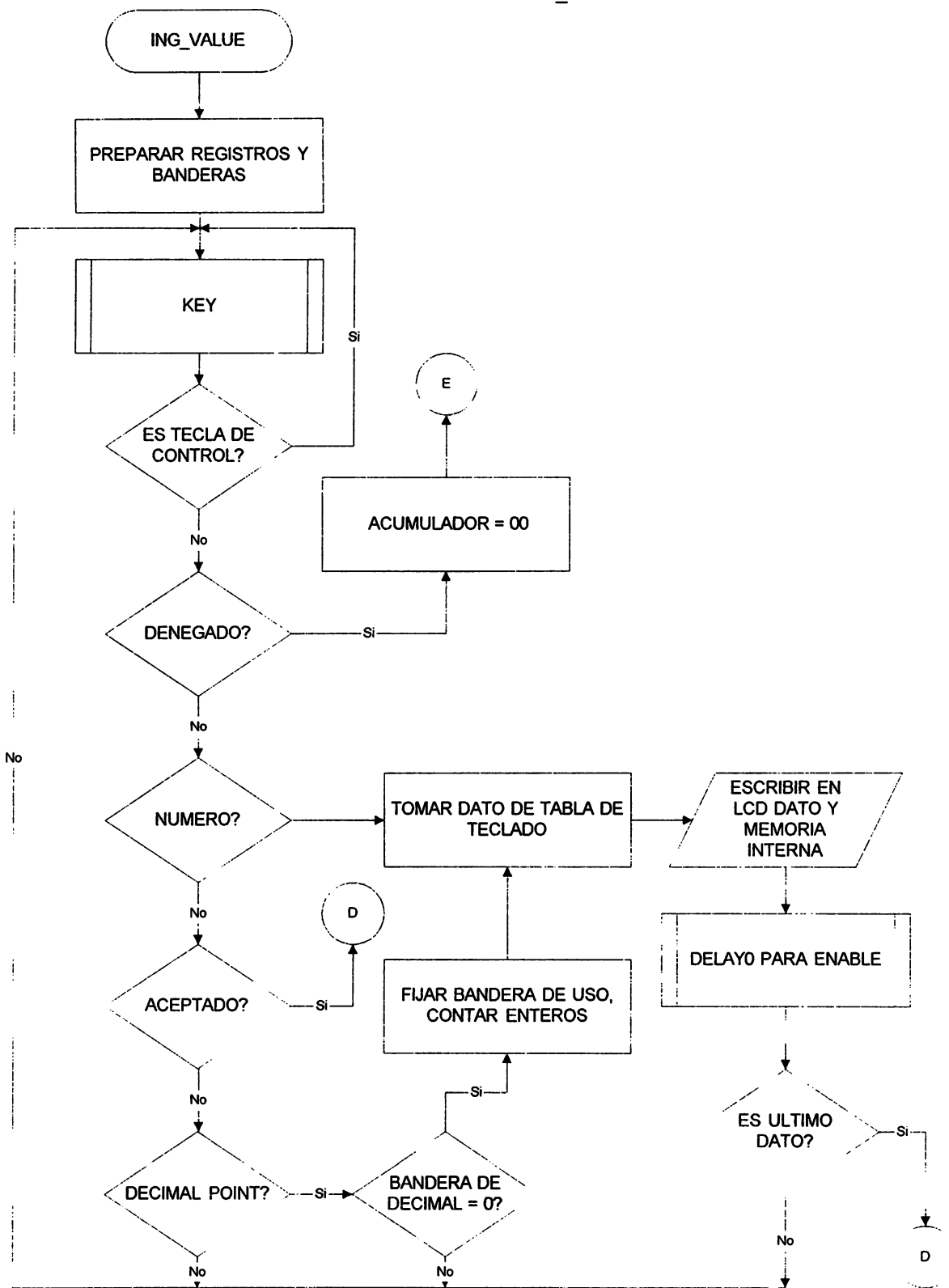
SUBROUTINE CONVERT

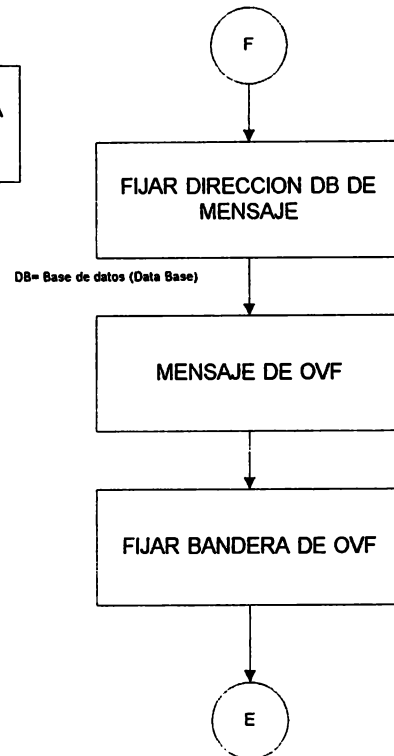
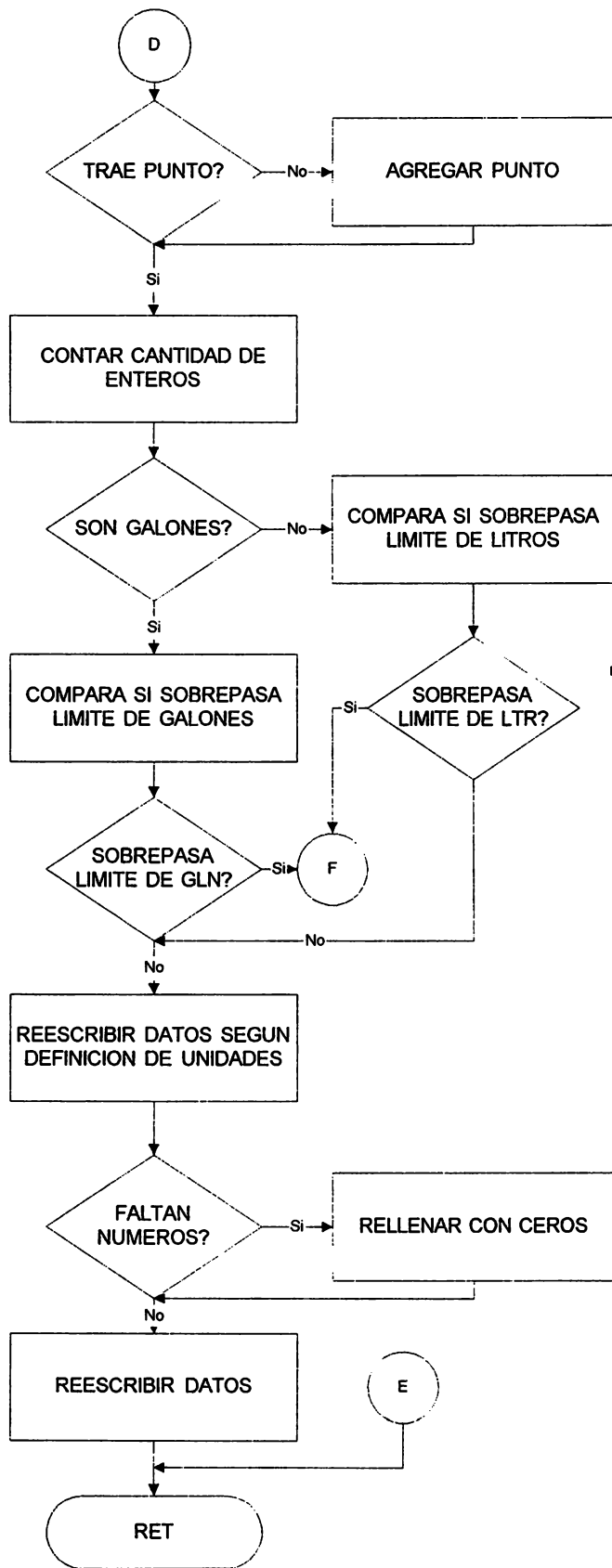




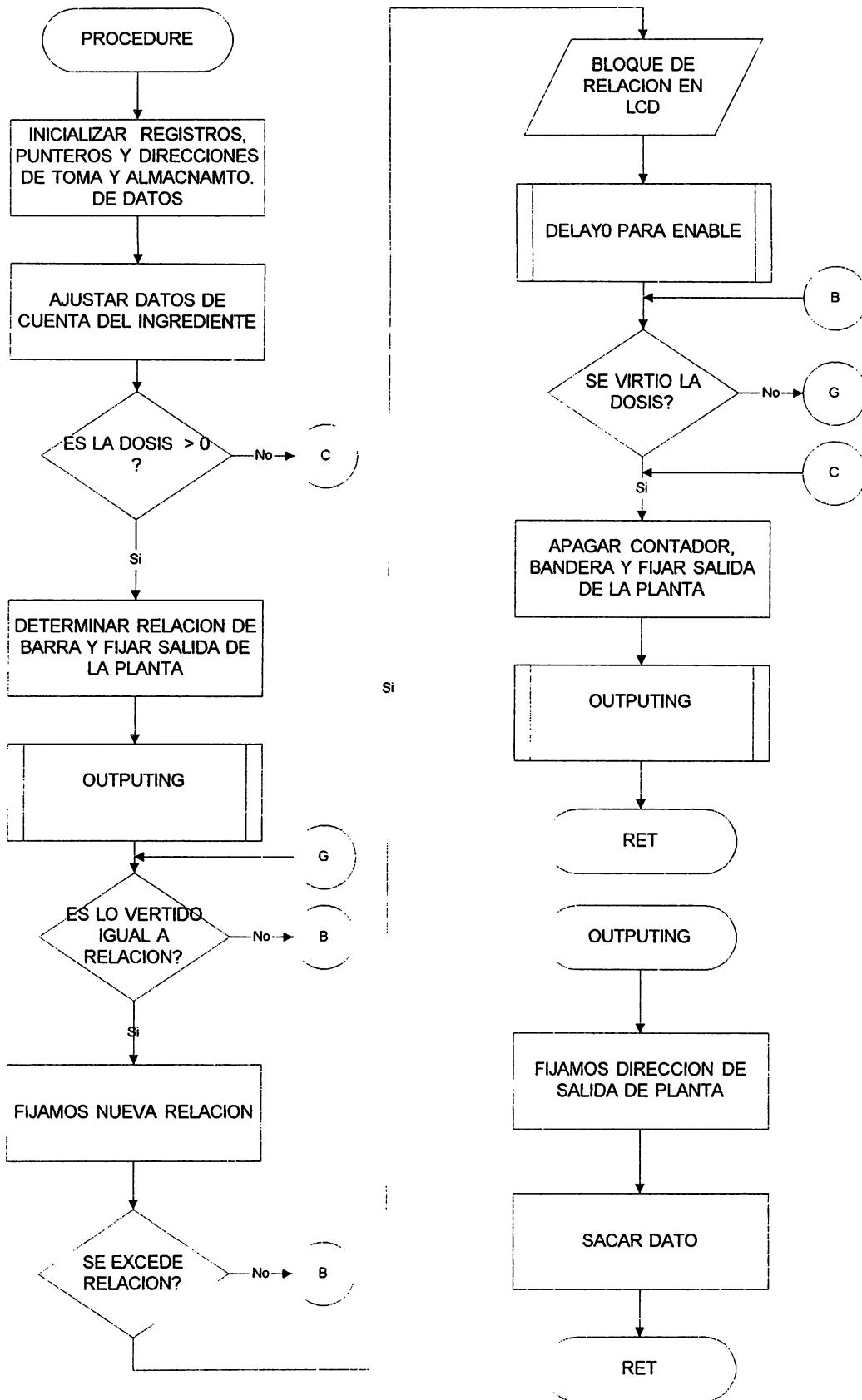
SUBROUTINA HAY\_ALGO



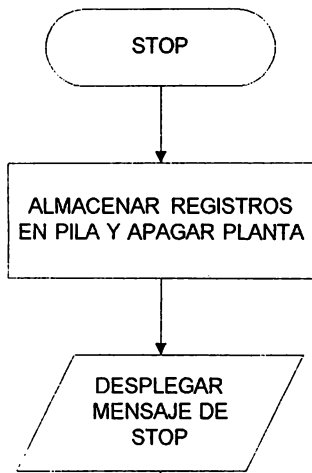




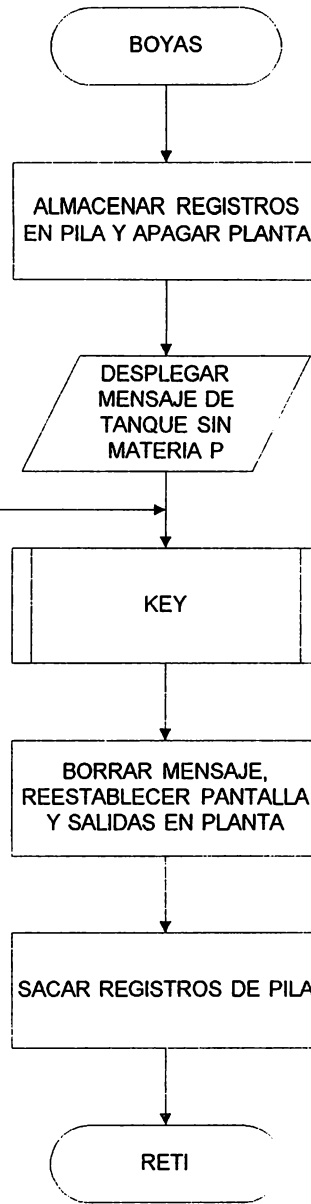
SUBROUTINA PROCEDURE

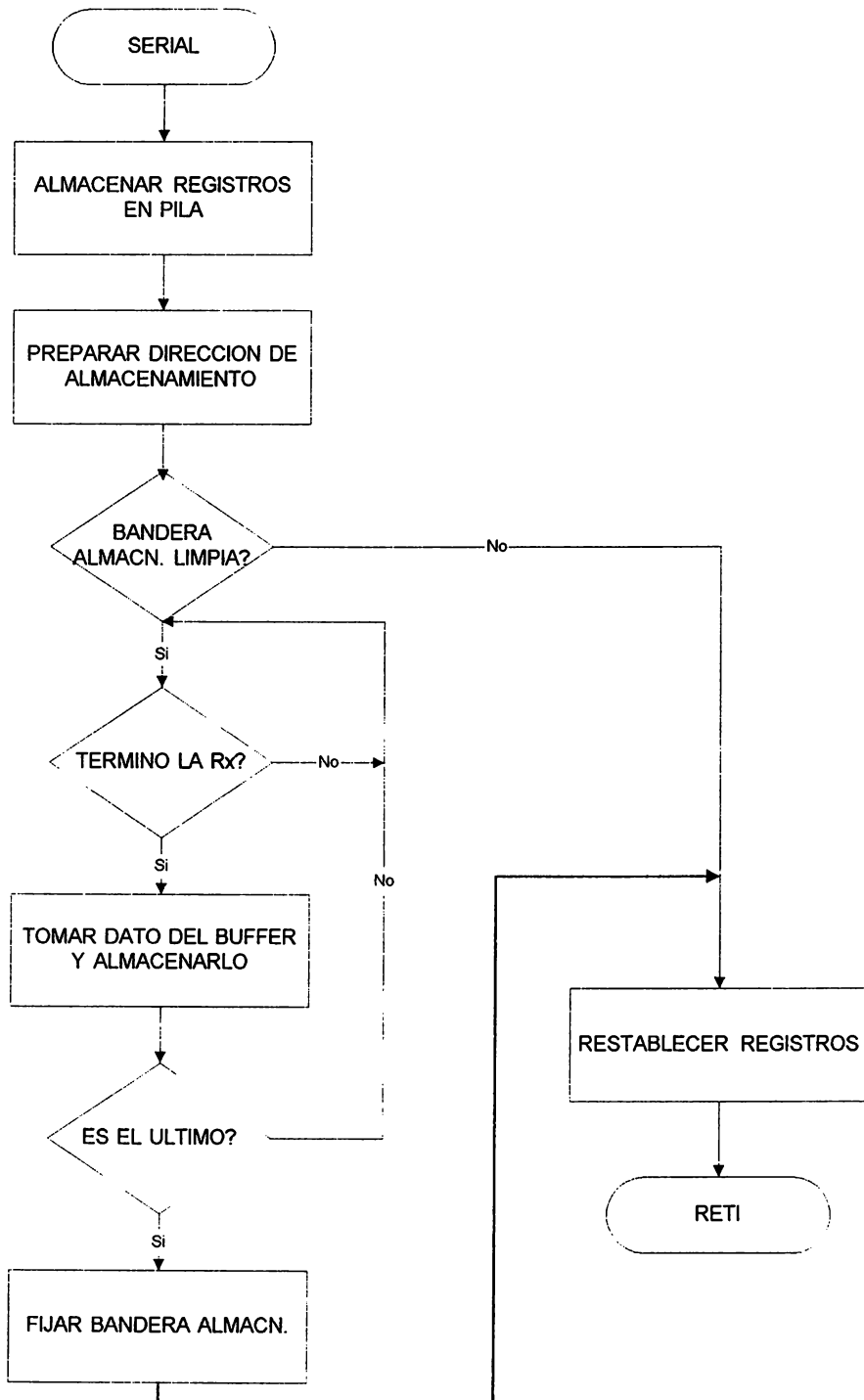


INTERRUPCION STOP



INTERRUPCION BOYAS







```

; EXTERNAL INTERRUPT 1          0013H
; SERIAL PORT                  0023H
; GENERAL STOP= DEFINIDA POR LA INT0, PARA
; O DETIENE EL PROCESO.
; BOYAS= ATIENDE LA NO PRSENCIA DE MATERIA
; PRIMA O INGREDIENTE QUE SE MEZCLA.
; SERIAL_PORT= PERMITE LA COMUNICACION VIA
; SERIAL POR LA INTERFAZ RS-232C.
;

```

```

.....INTERRUPT VECTORS.
ORG 0003H
JMP STOP          ;GENERAL STOP
ORG 0013H
JMP BOYAS         ;NO INGREDIENT
ORG 0023H
JMP SERIAL       ;R2-232C
ORG 0030H

```

### 1. RUTINA INICIO

```

; *****CONFIGURACION DEL 8052*****
INICIALIZA:MOV A,#FFH          ;LAS
              CALL OUTPUTING   ;SALIDAS
              MOV IP,#01H      ;INT0 MAX PRIORITY
              MOV T2CON,#01H    ;INICIALIZANDO T2
              MOV TMOD,#00100101B;MODOS PARA T1/0
*****

```

#### 1.1 INICIALIZANDO LCD

```

MOV DPTR,#DISPLAY ;DIRECCION DE LCD
MOV R0,#04H        ;RETARDO DE
CALL DELAY0        ;0.004 SEGUNDOS
LCDSET 38H, 01H    ;1
LCDSET 14H, 01H    ;2
LCDSET 0CH, 01H    ;3
LCDSET 06H, 01H    ;4
INICIO: CALL LIMPIAR ;CLEAR LCD
PANTALLA #1
LCDSET 84H, 01H    ;FIJANDO CURSOR
MENSAJE 08H, DAT011 ;PRIMER PANTALLA
LCDSET C4H, 01H    ;FIJANDO CURSOR
MENSAJE 08H, DAT012 ;PRIMER PANTALLA

```

#### 1.2 INICIALIZANDO SERIAL PORT

```

MOV TH1,#-3        ;FIJAR VELOCIDAD
MOV TL1,#-3        ;A 9600 BAUDIOS
MOV SCON,#01010000B;8 BITS BAUDRATE
SETB TR1           ;ARRANCA CONTADOR
MOV R0,#4EH        ;ESPERAMOS A ES-
CALL DELAY1        ;TABILIZAR BUFFER
PANTALLA #2
CALL LIMPIAR       ;CLEAR LCD
LCDSET 85H, 01H    ;FIJANDO CURSOR
MENSAJE 06H, DAT021 ;SEGUNDA PANTALLA
LCDSET C3H, 01H    ;FIJANDO CURSOR
MENSAJE 0AH, DAT022 ;SEGUNDA PANTALLA
MOV R0,#50H        ;4 SECONDS DELAY
CALL DELAY1
PANTALLA #3

```

```

A_DONDE: MENSAJE 16D, DAT042 ; CUARTA PANTALLA
          CALL KEY           ; READ SELECTION
          CJNE A, #00H, SERIA ; SE DESEA
          JMP BEGIN         ; DOSIFICAR?
SERIA:    JNB RS232C, RECET  ; CONFIRMA SI SE
          SETB GFLAG        ; ENVIAN LA RECETA
          JMP RVSND0        ; VIA SERIAL
RECET:    CJNE A, #01H, A_DONDE ; RECETAS DE ROM?
RECETA:   CALL LIMPIAR      ; CLEAR LCD
          LCDSET 80H, 01H    ; FIJANDO CURSOR
          MENSAJE 12D, DAT051 ; QUINATA PANTALLA
          LCDSET C0H, 01H    ; FIJANDO CURSOR
          MENSAJE 07H, DAT052 ; QUINTA PANTALLA
SERA:     CALL KEY           ; READ SELECTION
          CJNE A, #00H, RECETA2 ; RECETA #1?
          MOV DPTR, #BLLTX    ; SI: TOMAR DATOS
L:         MOV R7, #24D       ; DE MEMORIA ROM Y
          MOV R1, #26H        ; LOCALIZARLO EN
R:         MOVX A, @DPTR      ; AREA DE USO DE
          MOV @R1, A          ; RAM INTERNA DEL
          INC DPTR            ; MICROCONTROLADOR
          INC R1              ; Y FIJAR LA FLAG
          DJNZ R7, R          ; RESPECTIVA Y
          SETB GFLAG         ; LUEGO DESPLEGAR
          JMP RVSND0         ; LA RECETA EN LCD
RECETA2:  CJNE A, #01H, NO_RCTA ; RECETA #2?
          MOV DPTR, #CLORO
          JMP L
NO_RCTA:  CJNE A, #07H, RECETA3 ; BACK?
          JMP PATH
RECETA3:  CJNE A, #0BH, SERA    ; NEXT SCREEN?
          CALL LIMPIAR        ; CLEAR LCD
          LCDSET 80H, 01H    ; FIJANDO CURSOR
          MENSAJE 07H, DAT053 ; QUINTA PANTALLA
          LCDSET C0H, 01H    ; FIJANDO CURSOR
          MENSAJE 06H, DAT054 ; QUINTA PANTALLA
SERA1:    CALL KEY           ; READ SELECTION
          CJNE A, #02H, RECETA4 ; RECETA #3?
          MOV DPTR, #LEJIA
          JMP L
RECETA4:  CJNE A, #04H, NO_RCTA1 ; RECETA #4?
          MOV DPTR, #CERA
          JMP L
NO_RCTA1: CJNE A, #07H, SERA1   ; BACK?
          JMP RECETA

```

1.4 LECTURA DEL TECLADO Y DATOS

```

;
;
;
; PANTALLA #6
BEGIN:    CALL LIMPIAR        ; CLEAR LCD
          LCDSET 83H, 01H    ; FIJANDO CURSOR
          MENSAJE 10D, DAT061 ; SEXTA PANTALLA
          LCDSET C1H, 01H    ; FIJANDO CURSOR
          MENSAJE 14D, DAT062 ; SEXTA PANTALLA
UNIT:     CALL KEY           ; READ SELECTION
          CJNE A, #0EH, GAL   ; SON LITROS?
          CLR GFLAG          ; BANDERA
          MOV B, #3EH        ; DIRECCION EN RAM
          JMP READY          ; INTERNA
GAL:      CJNE A, #0CH, UNIT  ; SON GALONES?
          SETB GFLAG         ; BANDERA

```

```

MENSAJE 14D, DAT072 ;II  INGREDIENTE
LCDSET C0H, 01H      ;FIJANDO  CURSOR
MENSAJE 10D, DAT07C ;II  INGREDIETNE
CALL ING_VALUE
CJNE A,#00H,IN3
JMP IN2R
;TERCER INGREDIENTE
IN3:    MOV  A, #06H      ;POSICIONANDO EN
        ADD  A, B        ;NUEVA DIRECCION
        MOV  B, A        ;DE DATOS
IN3R:   CALL LIMPIAR    ;CLEAR LCD
        LCDSET 81H, 01H ;FIJANDO  CURSOR
        MENSAJE 14D, DAT073 ;III  INGREDIETNE
        LCDSET C0H, 01H ;FIJANDO  CURSOR
        MENSAJE 10D, DAT07C ;III  INGREDIETNE
        CALL ING_VALUE
        CJNE A,#00H,IN4
        JMP  IN3R
;CUARTO INGREDIENTE
IN4:    MOV  A, #06H      ;POSICIONANDO EN
        ADD  A, B        ;NUEVA DIRECCION
        MOV  B, A
IN4R:   CALL LIMPIAR    ;CLEAR LCD
        LCDSET 81H, 01H ;FIJANDO  CURSOR
        MENSAJE 14D, DAT074 ;IV   INGREDIENTE
        LCDSET C0H, 01H ;FIJANDO  CURSOR
        MENSAJE 10D, DAT07C ;IV   INGREDIENTE
        CALL ING_VALUE
        CJNE A,#00H,RVSND0
        JMP  IN4R
; PANTALLA #8
RVSND0: CALL LIMPIAR    ;CLEAR LCD
        JNB  GFLAG, ADEMAS ;SON GALONES?
        MOV  R1,#26H
        LCDSET 80H, 01H ;FIJANDO  CURSOR
        MENSAJE 16D, DAT08g ;ESCRIBIENDO LAS
        LCDSET C0H, 01 ;UNIDADES
        MENSAJE 16D, DAT08g
        JMP  LUEGO
ADEMAS: MOV  R1,#3EH      ;SON LITROS?
        LCDSET 80H, 01H ;FIJANDO  CURSOR
        MENSAJE 16D, DAT081 ;ESCRIBIENDO LAS
        LCDSET C0H, 01 ;UNIDADES
        MENSAJE 16D, DAT081
LUEGO:  LCDSET 80H, 01H ;SE  UBICA  CADA
        CALL MANDAR    ;DATO DE LOS #4
        LCDSET 89H, 01H ;INGREDIENTES EN
        CALL MANDAR    ;PANTALLA  PARA
        LCDSET C0H, 01H ;CORROBORAR SI
        CALL MANDAR    ;ESTOS FUERON LOS
        LCDSET C9H, 01H ;ESPERADOS PARA
        CALL MANDAR    ;LUEGO DARLOS POR
        MOV  R0,#78H ;ACEPTADOS O NO
        CALL DELAY1
        JNB  OFLAG, ES_P ;OVERFLOW?
        CLR  OFLAG      ;LIMPIAR BANDERA
        JMP  OUTRANGE
ES_P:   CALL LIMPIAR    ;CLEAR LCD
        LCDSET 80H, 01H ;FIJANDO  CURSOR
        MENSAJE 16D, DAT091 ;DECIMA PANTALLA
        LCDSET C2H, 01H ;FIJANDO  CURSOR

```

```

DE_NUEVO:  MOV  A, @R0           ;LITROS A GALONES
           CJNE A, #2EH, BUENO  ;SE REALIZA EN LA
           INC  R0              ;SIGUIENTE RUTINA
           DEC  R6
           JMP  DE_NUEVO
BUENO:     SUBB A, #30H
           MOV  DPL, A
ABACO:     CALL MULT
EL_SIGNTE: INC  R0
           MOV  A, @R0
           CJNE A, #2EH, NO_POINT
           JMP  EL_SIGNTE
NO_POINT:  SUBB A, #30H
           ADD  A, DPL
           MOV  DPL, A
           CLR  A
           ADDC A, DPH
           MOV  DPH, A
           DJNZ R6, ABACO
           INC  R0
           MOV  A, DPL
           MOV  @R1, DPL
           INC  R1
           MOV  A, DPH
           MOV  @R1, A
           INC  R1
           DJNZ R5, PREP_HEXAA ;FIN  CONVERSION
           JB   GFLAG, OVFF     ;DETERMINA SI SON
           MOV  R1, #56H        ;LITROS O GALONES
           MOV  R6, #04H
GALONEO:   MOV  DPL, @R1        ;FIJO  DATO  EN
           INC  R1              ;EN  DPTR
           MOV  DPH, @R1
           MOV  R0, #21H
           CALL CONVERT        ;CONVIRTIENDO  A
           DEC  R1              ;GALONES
           MOV  @R1, DPL
           INC  R1
           MOV  @R1, DPH
           INC  R1
           DJNZ R6, GALONEO
OVF:       MOV  DPTR, #0000H    ;ESTA RUTINA DARA
           MOV  R6, #04H        ;LUZ  VERDE  PARA
           MOV  R0, #56H        ;CONTINUAR O NO
           CLR  A               ;PERMITIR SEGUIR
           CLR  C               ;CON  EL  PROCESO
C_A:       MOV  A, DPL          ;POR  SOBREPASAR
           ADDC A, @R0          ;VOLUMEN MAXIMO
           MOV  DPL, A         ;DEL  MEZCLADOR
           INC  R0             ;CAPACIDAD MAXIMA
           CLR  A              ;PERMITIDA ES DE
           ADDC A, @R0         ;50.175  GALONES
           ADD  A, DPH
           JC   H_Y_D
           MOV  DPH, A
           INC  R0
           DJNZ R6, C_A
           MOV  R6, DPH
           CJNE R6, #C4H, $+3
           JC   BIT
           MOV  R6, #C4H

```

```

LCDSET C0H, 01H ;FIJANDO CURSOR
MENSAJE 16D, DAT10C ;DECIMA PANTALLA
LCDSET 88H, 01H ;FIJANDO CURSOR
JB GFLAG, G2LNS
MENSAJE 08H, DAT102 ;DECIMA PANTALLA
JMP L2TRS ;
G2LNS: MENSAJE 08H, DAT101 ;DECIMA PANTALLA
L2TRS: MOV R1, 5FH ;DIRECCION INGR X
LCDSET 89H, 01H ;FIJANDO CURSOR
CALL MANDAR ;ESCRIBO INGREDNT
MOV 5FH, R1 ;SALVAR ULTIMO DT
MOV DPTR, #OUTBL ;TABLA DE SALIDAS
CLR A ;LIMPIAR ACUMULAD
ADD A, R4 ;SUMAR EL OFFSET
INC R4 ;INCREMENTO OFFST
MOVC A, @A+DPTR ;CARGA DATO SALID
MOV R3, A ;LO FIJAMOS EN R3
CALL PROCEDURE ;SUBROUTINA MEZCLA
INC R2
CJNE R2, #04H, NO_FIN ;ULTIMO INGREDNT?
PANTALLA #11
MOV IE, #00H ;INTERRUPT ENABLE
CALL LIMPIAR ;CLEAR LCD
LCDSET 88H, 01H ;FIJANDO CURSOR
MENSAJE 07H, DAT111 ;ONCEAVA PANTALLA
MOV A, #4CH ;POWER ON THE MIX
MOV R4, A
CALL OUTPUTING
LCDSET C2H, 01H ;FIJANDO CURSOR
MOV DPTR, #DISPLAY ;DIRECCION LCD
MOV R2, #0DH ;13 SEGUNDOS MAX
MOV A, #FFH ;UTILIZO CHARACTER
SETB LCD_RS ;INPUTS ARE DATAS
TEN: MOV R0, #14H ;1 SECOND DELAY
CALL DELAY1
MOVX @DPTR, A
SETB LCD_EN
MOV R0, #01H
CALL DELAY0
DJNZ R2, TEN
CALL OUTPUTING
CALL HAY_ALGO
JMP PATH
;/**/**/**/**/**/**/**/**/**/**/**/**/**/**/**/**/**/**/**/**/**/**/
;
; ** SUBROUTINA DELAY0 **
; ASUME R0 = CUENTA, TIEMPO = CUENTA*1000µs
; UTILIZA EL TIMER2 (T2) MAXIMO = 127ms
;
DELAY0: MOV TH2, #>MILI ;CUENTA
MOV TL2, #<MILI
SETB TR2 ;ENCENDER TIMER2
JNB TF2, $ ;VERIFICAR FLAG
CLR TF2 ;BORRAR FLAG
CLR TR2 ;APAGAR TIMER2
DJNZ R0, DELAY0 ;CICLO DE CUENTA
CLR LCD_EN ;DESHABILITANDO
RET
;
; ** SUBROUTINA DELAY1 **
; ASUME R0 = CUENTA, TIEMPO = CUENTA*50000µs

```

```

MANDO:      MOV  A,@R1           ;FIJO EL PRIMER
            MOVX @DPTR,A        ;DATO, SE ESCRIBE
            SETB LCD_EN        ;EN LCD
            MOV  R0,#01H
            CALL DELAY0
            INC  R1
            DJNZ R6,MANDO      ;ES EL ULTIMO?
            RET

```

187

```

;
;
;      ** SUBROUTINA LCDSET1 **
;      SUBROUTINA PARA LA ESCRITURA DE COMANDOS EN
;      EL LCD PARA SU FUNCIONAMIENTO
;

```

```

LCDSET1:   MOV  DPTR,#DISPLAY   ;LCD
            CLR  LCD_RS        ;MODO SETTING
            MOVX @DPTR,A        ;PROGRAMO
            SETB LCD_EN        ;ENABLE
            CALL DELAY0        ;ENABLE DELAY
            RET

```

```

;
;
;      ** SUBROUTINA LIMPIAR **
;      LIMPIA DE TODO CARACTER LA PANTALLA (LCD)
;

```

```

LIMPIAR:   LCDSET CLEAR, 01H   ;CLEAR
            MOV  R0,#05H       ;CLEAR DELAY
            CALL DELAY0        ;CLEAR DELAY
            RET

```

```

;
;
;      ** SUBROUTINA SEND **
;      SUBROUTINA PARA ENVIAR UNA TABLA CON CODIGO
;      ASCII AL DISPLAY DE CRISTAL LIQUIDO (LCD)
;

```

```

SEND:      CLR  A              ;LIMPIAR ACC
            MOVC A,@A+DPTR     ;DB EQUIVALENTE
            PUSH DPL           ;GUARDAR
            PUSH DPH           ;DIRECCION
            MOV  DPTR,#DISPLAY ;DIRECCION LCD
            MOVX @DPTR,A       ;ESCRIBIRLO LCD
            SETB LCD_EN        ;ENABLE
            MOV  R0,#01H
            CALL DELAY0
            POP  DPH            ;RECUPERAR DPTR
            POP  DPL
            INC  DPTR
            DJNZ R7,SEND       ;ULTIMO DATO?
            RET

```

```

;
;
;      ** SUBROUTINA KEY **
;      SUBROUTINA QUE UNICAMENTE TOMA DATO INDICADO
;      POR EL KEY ENCODER 74C922
;

```

```

KEY:       JNB  KEYENB, KEY     ;ESPERA TECLA
            MOV  DPTR,#KEYBRD   ;DIRECCION
            MOVX A,@DPTR        ;READING KEY
            ANL  A, #0FH        ;MASCARA
            JB   KEYENB, $       ;FLANCO DE CAIDA
            RET

```

```

;
;
;      ** SUBROUTINA ING_VALUE **
;      SUBROUTINA PARA ENVIAR UNA TABLA CON CODIGO
;      ASCII AL DISPLAY DE CRISTAL LIQUIDO (LCD)
;

```

```

VL5:      CJNE A, #0CH, DND      ;DECIMAL POINT?
          JB  DFLAG, VLOUT      ;ONLY ONE
          PUSH A
          CALL WIP
          POP  A
          SETB DFLAG           ;FIRST AND LAST
DND:      SETB LCD_RS          ;MODO DATOS
          MOV  DPTR, #TECLA     ;FIJAR PUNTERO
          MOVC A, @A+DPTR      ;CARGA
          MOV  @R1, A          ;ALMACENARLO
          MOV  DPTR, #DISPLAY   ;DIRECCION LCD
          MOVX @DPTR, A        ;ESCRIBIRLO LCD
          SETB LCD_EN          ;ENABLE
          MOV  R0, #01
          CALL DELAY0
          INC  R1              ;NUEVA DIRECCION
          DJNZ R4, ING_VAL     ;ULTIMO DATO?
          JMP  MNB
L_A:      CJNE R5, #00H, EPN    ;NO HAY PUNTO?
          JB  DFLAG, EPN       ;HAY PUNTO?
          MOV  @R1, #2EH       ;UBICACION DEL
          CALL WIP             ;PUNTO DECIMAL Y
          DEC  R4              ;AJUSTE DE CEROS
          INC  R1              ;EN EL RESTO
EPN:      MOV  @R1, #30H       ;REUBICACION DE
          INC  R1              ;LOS DECIMALES Y
          DJNZ R4, EPN        ;ENTEROS DIGITADO
MNB:      JB  GFLAG, DE_TD     ;POR EL USUARIO
          CJNE R5, #04H, $+3
          JC  ABC
          JMP  PRS00
ABC:      CLR  C
          MOV  A, #03H
          SUBB A, R5
          MOV  R3, A
          MOV  R2, #03H
          JMP  S0
DE_TD:   CJNE R5, #03H, $+3
          JC  DEF
          JMP  PRS00
DEF:      CLR  C
          MOV  A, #02H
          SUBB A, R5
          MOV  R3, A
          MOV  R2, #04H
S0:      MOV  R6, #06H
          MOV  R0, #5EH
          MOV  R1, B
S1:      MOV  A, @R1
          MOV  @R0, A
          INC  R1
          INC  R0
          DJNZ R6, S1
          MOV  R1, B
          MOV  R0, #5EH
          MOV  R6, #06H
          CJNE R3, #00H, S2
          JMP  S3
S2:      MOV  @R1, #30H
          INC  R1
          DEC  R6

```

```

WIP:      MOV  A, #06H           ;DETERINA EL #
          SUBB A, R4             ;DE ENTEROS
          MOV  R5, A
PRS:      RET

;
;  ** SUBROUTINA CONVERSION LITROS A GALONES **
;  CUANDO LAS UNIDADES SEAN LITROS, FACTOR DE
;  CONVERSION 1G = 3.7851 ó 3785ml (L_TO_G)
;  CICLOS: 655 MAXIMO (SI OSCFREC =11.0592 MHZ
;  LA SUBROUTINA TOMARA MAX 707.46µs EJECUTARLA
;
CONVERT:  CALL DIVISION          ;DIVIDIR
          MOV  A, R5              ;I RESULTADO
          MOV  @R0, A            ;GUARDARLO
MIX:      CALL MULT              ;AGREGAR CERO
          CALL DIVISION          ;DIVIDIR
          INC  R0                ;INCRMNTO PUNTERO
          MOV  A, R5              ;n RESULTADO
          MOV  @R0, A            ;ES GUARDADO
          CJNE R0,#25H,MIX       ;ES 5° RESULTADO?
DESPLAZA: MOV  DPTR,#0000H       ;LIMPIAR DPTR
          MOV  R0, #21H          ;SACAR
          MOV  A, @R0            ;I RESULTADO
          MOV  DPL, A            ;CARGARLO AL DPL
MIX1:     CALL MULT              ;MULTIPLICAR
          INC  R0                ;INCRMNTO PUNTERO
          MOV  A, @R0            ;EL n RESULTADO
          ADD  A,DPL              ;SE SUMA A MIX1
          MOV  DPL, A            ;GUARDALO EN DPL
          CLR  A                 ;LIMPIAR ACC
          ADDC A, DPH             ;CARRY AL DPH
          MOV  DPH, A            ;GUARDARLO EN DPH
          CJNE R0,#25H,MIX1     ;ES 5° RESULTADO?
          RET

; ||||| SUBROUTINA DIVISION |||||
DIVISION: MOV  R2,#<L_TO_G       ;FIJANDO DIVISOR
          MOV  R3,#>L_TO_G       ;CTE L_TO_G
          MOV  R5,#00H           ;CONTADOR A 0
FALTAN:   MOV  A,DPL              ;SE RESTA DIVI-
          CLR  C                 ;DENDO CON EL
          SUBB A,R2               ;DIVISOR
          MOV  B, A
          MOV  A,DPH
          SUBB A,R3
          JC   NO_MORE
          MOV  DPH, A
          MOV  DPL, B            ;AJUSTE
          INC  R5
          CJNE R5,#00,FALTAN     ;INCREMENTAR
          JMP  FALTAN
NO_MORE:  CLR  C
TERMINO:  RET

; ||||| SUBROUTINA MULTIPLICACION |||||
MULT:     MOV  B,#0AH            ;FIJAR CONSTANTE
          MOV  A, DPL
          MUL  AB
          MOV  DPL, A            ;EL PRODUCTO ES
          MOV  R4, B             ;PARTE BAJA DPTR
          MOV  B, #0AH           ;FIJAR CONSTANTE
          MOV  A, DPH
          MUL  AB

```

```

        CLR    TLOW0
        MOV    A, #FFH
        CLR    C
        SUBB   A, 61H
        CJNE  A, #FFH, TML0
        SETB  TLOW0
TML0:   MOV    TL0, A
        INC   R1
        MOV   61H, @R1
        CLR   C
        MOV   A, #FFH
        SUBB  A, 61H
        MOV   60H, A
        CJNE A, #FFH, THG0
        INC   R1
        SETB  C
        ANL   C, TLOW0
        JC    NADA_DATO
        MOV   TH0, A
THG0:   INC   R1
        MOV   5EH, R1
        MOV   A, 61H
        MOV   B, #0AH           ;RELACION BARRA
        DIV  AB                 ;DE MEZCLADO VRS
        CJNE A, #00H, D_V
        INC   A
        CLR   C
D_V:    MOV   B, A               ;VOLUMEN VERTIDO
        ADD  A, 60H
        MOV   60H, A
        SETB TR0
        MOV   A, R3
        CALL OUTPUTING
        SETB EX1
        LCDSET C2H, 01H       ;FIJANDO CURSOR
        MOV  DPTR, #DISPLAY   ;DISPLAY ADDRESS
        MOV  A, TH0
        SETB LCD_RS           ;DATA SEND ENABLE
DOSIS:  CJNE A, 60H, AUN_NO
        ADD  A, B
        MOV  60H, A
        CJNE R6, #0AH, $+3
        JNC  AUN_NO
        INC  R6
        MOV  A, #FFH
        MOVX @DPTR, A
        SETB LCD_EN
        MOV  R0, #01H
        CALL DELAY0
AUN_NO: MOV  A, TH0
        JNB  TF0, DOSIS
        CLR  TR0
        CLR  TF0
        CLR  EX1
        MOV  A, #FFH           ;SELECTION
        CALL OUTPUTING
NADA_DATO: MOV  5EH, R1
        RET
OUTPUTING: MOV  DPTR, #OUTPUT  ;RELAY ADDRESS
        MOVX @DPTR, A         ;OUTPUT SELECTION
        RET

```

```
; O COMPLETAMENTE (ABORTAR Y REINICIAR)
```

191

```
STOP:      CLR  TR0
           PUSH A
           PUSH PSW
           MOV  A,#FFH           ;DATO
           CALL OUTPUTING       ;APAGANDO
PANTALLA STOP GENERAL #1
           LCDSET 81H, 01H      ;FIJANDO  CURSOR
           MENSAJE 06H, STOP01 ;PRIMERA PANTALLA
           JMP  RESTORE         ;STOP  AL  VERTIR
```

```
;
;          !! INTERRUPCION BOYAS!!
;          LLAMADO POR LA INTERRUPCION EXTERNA 1, PARA
;          DETENER EL PROCESO MOMENTANEAMENTE Y AVISAR
;          AL OPERARIO LA FALTA DE MATERIA PRIMA
;
```

```
BOYAS:     CLR  TR0
           PUSH A
           PUSH PSW
           MOV  A,#FFH           ;DATO
           CALL OUTPUTING       ;APAGANDO
PANTALLA NO INGREDIENTE #1
           LCDSET 81H, 01H      ;FIJANDO  CURSOR
           MENSAJE 05H, DATONI  ;PRIMERA PANTALLA
           LCDSET 86H, 01H      ;FIJANDO  CURSOR
           MOV  A, #31H
           ADD  A, R2            ;EL  NUMERO  DEL
           SETB LCD_RS          ;DATOS
           MOVX @DPTR, A
           SETB LCD_EN
           MOV  R0, #01H
           CALL DELAY0
```

```
RESTORE:   CALL KEY             ;READ  KEYBOARD
           LCDSET 81H, 01H      ;FIJANDO  CURSOR
           MENSAJE 06H, DATSPC ;DECIMA PANTALLA
           MOV  A, R3           ;DATO
           CALL OUTPUTING       ;APAGANDO
           MOV  DPTR, #DISPLAY
           MOV  R0, #01H
           MOV  A, #C2H
           ADD  A, R6
           CLR  LCD_RS
           MOVX @DPTR, A
           SETB LCD_EN
           CALL DELAY0
           SETB LCD_RS
           SETB TR0
           POP  PSW
           POP  A
           RETI                 ;FIN DE FALLA
```

```
;
;          !! INTERRUPCION SERIAL_PORT!!
;          ATIENDE EL LLAMADO DEL PUERTO SERIAL, PARA
;          PERMITIR INTERCAMBIO DE INFORMACION USANDO
;          LA INTERFACE RS-232C CON CARACTERISTICAS:
;          9600 BPS, 1 STOP BIT, NO PARITY, 8 BITS
;
```

```
SERIAL:    PUSH R1
           PUSH R4
           PUSH A
```

```

                                PUSH PSW
                                MOV  R1, #26H
                                MOV  R4, #24D
STORE:                          JB   RS232C,B_C
                                JNB  RI, $
                                CLR  RI
                                MOV  A, SBUF
                                MOV  @R1, A
                                INC  R1
                                CJNE A,#0DH,STORE
                                SETB RS232C
B_C:                             POP  PSW
                                POP  A
                                POP  R4
                                POP  R1
                                RETI

```

```

;
;
; ++ DATA BASE ++
; BASE DE DATOS QUE CONTIENE MENSAJES PARA EL
; DISPLAY DE CRISTAL LIQUIDO
;
; PANTALLA #1
DAT011:  DB 'POWER ON'          ;8
DAT012:  DB 'STARTING'         ;8
; PANTALLA #2
DAT021:  DB E4H,'C8032'        ;6
DAT022:  DB 'CONTROLLER'      ;10
; PANTALLA #3
DAT031:  DB 'C. CONTRERAS'    ;12
DAT032:  DB 'V. CUCHILLAC'    ;12
; PANTALLA #4
DAT041:  DB 'MODO:1)DOSIFICAR';16
DAT042:  DB '      2)RECETARIO';16
; PANTALLA #5
DAT051:  DB '1)BRILLANTEX'     ;12
DAT052:  DB '2)LEJIA'         ;7
DAT053:  DB '3)CLORO'         ;7
DAT054:  DB '4)CERA'          ;6
; PANTALLA #6
DAT061:  DB 'UNIDADES'        ;10
DAT062:  DB 'gal<*> ltr<#>' ;14
; PANTALLA #7
DAT071:  DB 'INGREDIENTE #1'  ;14
DAT072:  DB 'INGREDIENTE #2'  ;14
DAT073:  DB 'INGREDIENTE #3'  ;14
DAT074:  DB 'INGREDIENTE #4'  ;14
DAT07C:  DB 'CANTIDAD: '      ;10+6
; PANTALLA #8
DAT08g:  DB          G          G';16
DAT08l:  DB          l          l';16
; PANTALLA #9
DAT091:  DB 'DESEA CONTINUAR?';16
DAT092:  DB 'SI<C> NO<B>'     ;12
; PANTALLA #10
DAT101:  DB 7EH,'          G'  ;08
DAT102:  DB 7EH,'          l'  ;08
DAT10C:  DB '0%          100%';16
; PANTALLA #11
DAT111:  DB 7EH,'MIX ON'      ;07

```



```

                PUSH PSW
                MOV  R1, #26H
                MOV  R4, #24D
                JB   RS232C,B_C
STORE:         JNB  RI, $
                CLR  RI
                MOV  A, SBUF
                MOV  @R1, A
                INC  R1
                CJNE A,#0DH,STORE
                SETB RS232C
B_C:          POP  PSW
                POP  A
                POP  R4
                POP  R1
                RETI

```

```

;
;
;          ++ DATA BASE ++
;          BASE DE DATOS QUE CONTIENE MENSAJES PARA EL
;          DISPLAY DE CRISTAL LIQUIDO
;
; PANTALLA #1
DAT011:    DB 'POWER ON'          ;8
DAT012:    DB 'STARTING'         ;8
; PANTALLA #2
DAT021:    DB E4H, 'C8032'       ;6
DAT022:    DB 'CONTROLLER'      ;10
; PANTALLA #3
DAT031:    DB 'C. CONTRERAS'    ;12
DAT032:    DB 'V. CUCHILLAC'    ;12
; PANTALLA #4
DAT041:    DB 'MODO:1)DOSIFICAR';16
DAT042:    DB '      2)RECETARIO';16
; PANTALLA #5
DAT051:    DB '1)BRILLANTEX'    ;12
DAT052:    DB '2)LEJIA'        ;7
DAT053:    DB '3)CLORO'        ;7
DAT054:    DB '4)CERA'         ;6
; PANTALLA #6
DAT061:    DB 'UNIDADES : '     ;10
DAT062:    DB 'gal<*> ltr<#>' ;14
; PANTALLA #7
DAT071:    DB 'INGREDIENTE #1'  ;14
DAT072:    DB 'INGREDIENTE #2'  ;14
DAT073:    DB 'INGREDIENTE #3'  ;14
DAT074:    DB 'INGREDIENTE #4'  ;14
DAT07C:    DB 'CANTIDAD: '      ;10+6
; PANTALLA #8
DAT08g:    DB          G          G';16
DAT08l:    DB          1          1';16
; PANTALLA #9
DAT091:    DB 'DESEA CONTINUAR?';16
DAT092:    DB 'SI<C> NO<B>'     ;12
; PANTALLA #10
DAT101:    DB 7EH, '          G' ;08
DAT102:    DB 7EH, '          1' ;08
DAT10C:    DB '0%          100%';16
; PANTALLA #11
DAT111:    DB 7EH, 'MIX ON'     ;07
; PANTALLA #12
DAT121:    DB 'MEZCLADORA LLENA';16

```

```
DAT122:    DB 'PUEDE ENVASAR'    ;14
; PANTALLA STOP GENERAL
STOP01:    DB 'STOP!!'            ;06
; PANTALLA NO INGREDIENTE
DATONI:    DB 'TANK '             ;05+1
DATSPC:    DB '                   ;06
; PANTALLA DE INTRODUCCION DE DATOS
TECLA:     DB '123456 789 .0 '    ;16
; PANTALLA MENSAJE DE OVERFLOW
OVERFLOW:  DB ' OVFL '           ;06
OVR001:    DB 'OVERFLOW!!'       ;10
; RECETAS
CERA:      DB '27.00010.12512.87500.000'
CLORO:     DB '38.50011.50000.00000.000'
LEJIA:     DB '10.50016.75022.75000.000'
BLLTX:     DB '25.00004.10005.90015.000'
; OUTPUT TABLE
OUTBL:     DB 00H,11H,22H,33H
           END
```

## SET DE INSTRUCCIONES

INSTRUCCIONES ARITMÉTICAS

NEMONICO	OPERACION	DIRECCIONAMIENTO				CICLOS	BANDERAS AFECTADAS		
		DIR	IND	REG	INM		C	OV	AC
ADD A,<byte>	$A = A + \text{<byte>}$	X	X	X	X	1	X	X	X
ADDC A,<byte>	$A = A + \text{<byte>} + C$	X	X	X	X	1	X	X	X
SUBB A,<byte>	$A = A - \text{<byte>} - C$	X	X	X	X	1	X	X	X
INC A	$A = A + 1$	SOLO ACC				1			
INC <byte>	$\text{<byte>} = \text{<byte>} + 1$	X	X	X		1			
INC DPTR	$DPTR = DPTR + 1$	SOLO DPTR				2			
DEC A	$A = A - 1$	SOLO ACC				1			
DEC <byte>	$\text{<byte>} = \text{<byte>} - 1$	X	X	X		1			
MUL AB	$A:B = A \times B$	SOLO A y B				4	0	X	
DIV AB	$A = \text{int}(A/B), B = \text{res}(A/B)$	SOLO A y B				4	0	X	
DA A	AJUSTE DECIMAL (BCD)	SOLO ACC				1	X		

INSTRUCCIONES LÓGICAS

NEMONICO	OPERACION	DIRECCIONAMIENTO				CICLOS	BANDERAS AFECTADAS		
		DIR	IND	REG	INM		C	OV	AC
ANL A,<byte>	$A = A \text{ and } \text{<byte>}$	X	X	X	X	1			
ANL <byte>,A	$\text{<byte>} = \text{<byte>} \text{ and } A$	X				1			
ANL <byte>,#data	$\text{<byte>} = \text{<byte>} \text{ and } \#data$	X				2			
ORL A,<byte>	$A = A \text{ or } \text{<byte>}$	X	X	X	X	1			
ORL <byte>,A	$\text{<byte>} = \text{<byte>} \text{ or } A$	X				1			
ORL <byte>,#data	$\text{<byte>} = \text{<byte>} \text{ or } \#data$	X				2			
XRL A,<byte>	$A = A \text{ xor } \text{<byte>}$	X	X	X	X	1			
XRL <byte>,A	$\text{<byte>} = \text{<byte>} \text{ xor } A$	X				1			
XRL <byte>,#data	$\text{<byte>} = \text{<byte>} \text{ xor } \#data$	X				2			
CLR A	$A = 00h$	SOLO ACC				1			
CPL A	$A = \text{not } (A)$	SOLO ACC				1			
RL A	Rota ACC a izquierda 1 bit	SOLO ACC				1			
RLC A	Rota A izq a través de C 1 bit	SOLO ACC				1	X		
RR A	Rota ACC a derecha 1 bit	SOLO ACC				1			
RRC A	Rota A der a través de C 1 bit	SOLO ACC				1	X		
SWAP A	intercambia nibbles de A	SOLO ACC				1			

TRANSFERENCIA DE DATOS

NEMONICO	OPERACION	DIRECCIONAMIENTO				CICLOS
		DIR	IND	REG	INM	
MOV A,<fuente>	$A \leftarrow \text{<fuente>}$	X	X	X	X	1
MOV <dest>,A	$\text{<dest>} \leftarrow A$	X	X	X		1
MOV <dest>,<fuente>	$\text{<dest>} \leftarrow \text{<fuente>}$	X	X	X	X	2
MOV DPTR,#data16	$\text{<DPTR>} \leftarrow \text{dato de 16 bits}$				X	2
PUSH <fuente>	INC SP, MOV @SP,<fuente>	X				2
POP <dest>	MOV <dest>,@SP, DEC SP	X				2
XCH A,<byte>	$A \leftrightarrow \text{<byte>}$	X	X	X		1
XCHD A,@Ri	$A \leftrightarrow ((Ri))$ (sólo nibble bajo)		X			1

TRANSFERENCIA DE DATOS EXTERNOS

NEMONICO	OPERACION	DIRECCION CON	CICLOS
MOVX @Ri,A	$\text{WRITE EXTRAM}(Ri) \leftarrow A$	8 bits	2
MOVX A,@DPTR	$A \leftarrow \text{READ EXTRAM}(DPTR)$	16 bits	2
MOVX @DPTR,A	$\text{WRITE EXTRAM}(DPTR) \leftarrow A$	16 bits	2

MANEJO DE TABLAS

NEMONICO	OPERACION	CICLOS
MOVC A,@A + DPTR	$A \leftarrow \text{READ PGM MEM}(A + DPTR)$	2
MOVC A,@A + PC	$A \leftarrow \text{READ PGM MEM}(A + PC)$	2

INSTRUCCIONES BOOLEANAS

NEMONICO	OPERACION	CICLOS	BANDERAS AFECTADAS		
			C	OV	AC
ANL C,bit	C = C and bit	2	X		
ANL C,/bit	C = C and not bit	2	X		
ORL C,bit	C = C or bit	2	X		
ORL C,/bit	C = C or not bit	2	X		
MOV C,bit	C = bit	1	X		
MOV bit,C	bit = C	2			
CLR C	C = 0	1	0		
CLR bit	bit = 0	1			
SETB C	C = 1	1	1		
SETB bit	bit = 1	1			
CPL C	C = not C	1	X		
CPL bit	bit = not bit	1			
JC REL	salta REL localidades si C = 1	2			
JNC REL	salta REL localidades si C = 0	2			
JB bit,REL	salta REL localidades si bit = 1	2			
JNB bit,REL	salta REL localidades si bit = 0	2			
JBC bit,REL	salta REL localidades si bit = 1 y borra bit	2			

INSTRUCCIONES DE SALTO

NEMONICO	OPERACION	CICLOS
JMP addr	Salta a dirección addr	2
JMP @A + DPTR	Salta a dirección A + DPTR	2
CALL addr	Llama a subrutina en dirección addr	2
RET	Regresa de subrutina	2
RETI	Regresa de interrupción	2
NOP	No operación	1

INSTRUCCIONES DE SALTO

NEMONICO	OPERACION	DIRECCIONAMIENTO				CICLOS	BANDERAS AFECTADAS		
		DIR	IND	REG	INM		C	OV	AC
JZ rel	Salta si A=0	Sólo ACC				2			
JNZ rel	Salta si A≠0	Sólo ACC				2			
DJNZ <byte>,rel	Decrementa <byte> y salta si ≠0	X		X		2			
CJNE A,<byte>,rel	Salta si A ≠ <byte>	X			X	2	X		
CJNE A,#data,rel	Salta si A ≠ #data		X	X		2	X		

Instruction Code Summary

L	H	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0		NOP	JBC bit, rel	JB bit, rel	JNC rel	JZ rel	JNZ rel	SIMP rel	MOV DPTR, # data 16	ORL C, bit	ANL C, bit	PUSH dir	POP dir	MOVX @DPTR, A			
1		AJMP (P0)	ACALL (P0)	AJMP (P1)	ACALL (P2)	AJMP (P3)	ACALL (P5)	AJMP (P4)	ACALL (P4)	MOV C, bit	MOV C, bit	ACALL (P5)	ACALL (P6)	AJMP (P7)			
2		LJMP addr16	LCALL addr16	RET	ORL dir, A	XRL dir, A	ORL C, bit	ANL C, bit	MOVX A, @R1	MOVX A, @R0	INC DPTR	CLR bit	SETB bit	MOVX @R0, A			
3		RR A	RRC A	RL A	ANL dir, # data	XRL dir, # data	JMP @A+DPTR	MOVX A, @A+PC	MOVX A, @R1	INC DPTR	MOVX A, @R1	CLR C	SETB C	MOVX @R1, A			
4		INC A	DEC A	ADD A, # data	ADDC A, # data	ORL A, # data	ANL A, # data	XRL A, # data	MOV A, # data	MOV A, # data	MOV A, # data	SWAP A	DA A	CLR A			
5		INC dir	DEC dir	ADD A, dir	ADDC A, dir	ORL A, dir	ANL A, dir	XRL A, dir	MOV dir, # data	MOV dir, dir	SUBB A, # data	MOV dir, dir	DIV AB	MOV dir, rel			
6		INC @R0	DEC @R0	ADD A, @R0	ADDC A, @R0	ORL A, @R0	ANL A, @R0	XRL A, @R0	MOV @R0, # data	MOV @R0, dir	SUBB A, @R0	MOV @R0, dir	CJNE # data, rel	XCHD A, @R0	MOV A, @R0		
7		INC @R1	DEC @R1	ADD A, @R1	ADDC A, @R1	ORL A, @R1	ANL A, @R1	XRL A, @R1	MOV @R1, # data	MOV @R1, dir	SUBB A, @R1	MOV @R1, dir	CJNE @R1, # data, rel	XCHD A, @R1	MOV A, @R1		
8		INC R0	DEC R0	ADD A, R0	ADDC A, R0	ORL A, R0	ANL A, R0	XRL A, R0	MOV R0, # data	MOV R0, dir	SUBB A, R0	MOV R0, dir	CJNE R0, # data, rel	XCHD A, R0	MOV A, R0		
9		INC R1	DEC R1	ADD A, R1	ADDC A, R1	ORL A, R1	ANL A, R1	XRL A, R1	MOV R1, # data	MOV R1, dir	SUBB A, R1	MOV R1, dir	CJNE R1, # data, rel	XCHD A, R1	MOV A, R1		
A		INC R2	DEC R2	ADD A, R2	ADDC A, R2	ORL A, R2	ANL A, R2	XRL A, R2	MOV R2, # data	MOV R2, dir	SUBB A, R2	MOV R2, dir	CJNE R2, # data, rel	XCHD A, R2	MOV A, R2		
B		INC R3	DEC R3	ADD A, R3	ADDC A, R3	ORL A, R3	ANL A, R3	XRL A, R3	MOV R3, # data	MOV R3, dir	SUBB A, R3	MOV R3, dir	CJNE R3, # data, rel	XCHD A, R3	MOV A, R3		
C		INC R4	DEC R4	ADD A, R4	ADDC A, R4	ORL A, R4	ANL A, R4	XRL A, R4	MOV R4, # data	MOV R4, dir	SUBB A, R4	MOV R4, dir	CJNE R4, # data, rel	XCHD A, R4	MOV A, R4		
D		INC R5	DEC R5	ADD A, R5	ADDC A, R5	ORL A, R5	ANL A, R5	XRL A, R5	MOV R5, # data	MOV R5, dir	SUBB A, R5	MOV R5, dir	CJNE R5, # data, rel	XCHD A, R5	MOV A, R5		
E		INC R6	DEC R6	ADD A, R6	ADDC A, R6	ORL A, R6	ANL A, R6	XRL A, R6	MOV R6, # data	MOV R6, dir	SUBB A, R6	MOV R6, dir	CJNE R6, # data, rel	XCHD A, R6	MOV A, R6		
F		INC R7	DEC R7	ADD A, R7	ADDC A, R7	ORL A, R7	ANL A, R7	XRL A, R7	MOV R7, # data	MOV R7, dir	SUBB A, R7	MOV R7, dir	CJNE R7, # data, rel	XCHD A, R7	MOV A, R7		

2Byte	3Byte
2Cycle	4Cycle

Opcode map