

885216



**UNIVERSIDAD AMERICANA DE ACAPULCO**  
EXCELENCIA PARA EL DESARROLLO

---

**FACULTAD DE INGENIERÍA EN COMPUTACIÓN**

INCORPORADA A LA  
UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

**DISEÑO DE UN INTÉRPRETE GRÁFICO  
MEDIANTE UN SISTEMA DEDICADO,  
IMPLEMENTADO CON EL  
MICROCONTROLADOR MCS-51**

**T E S I S**

QUE PARA OBTENER EL TÍTULO DE:  
**INGENIERA EN COMPUTACIÓN**

PRESENTA:  
**MARIA MAGDALENA BARBERENA ZERMEÑO**

DIRECTOR DE TESIS  
**M. EN C. MARTÍN SANTIAGO DOMÍNGUEZ GONZÁLEZ**



ACAPULCO, GRO.

ABRIL DE 2004



Universidad Nacional  
Autónoma de México



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

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

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

ESTA TESIS NO SALE  
DE LA BIBLIOTECA

## **AGRADECIMIENTOS**

---

---



## **GRACIAS:**

A Dios por TODO.

A mi Father, por ser mi ángel, quererme, cuidarme y estar siempre conmigo.

A mi hermosa y queridísima familia, Mother, Tomaco, Fermín, Pili y Fidel, Lety y tío Armando, por su gran cariño y valiosísimo apoyo siempre incondicionales. Sépanse personajes protagónicos de mi vida y por supuesto de este trabajo.

A mis abuelos por todo lo que dejaron en mi.

A mi Pirrot, por tu terca obsesión que aquí se cumple.

A mi querido "Profe", por su amistad, paciencia y excelente dirección.

A la Universidad Americana de Acapulco por su valiosa contribución en mi formación.

A la Facultad de Ingeniería por su apoyo y buena disposición.

A la Universidad Nacional Autónoma de México por la Beca otorgada.

A cada uno de mis maestros por sus enseñanzas profesionales y personales.

A Gaby Berlanga por esta maravillosa idea.

A mis queridas "chicas", Lencha, Mary y Roge, por todos estos años de consentimiento.

A mi amiga Ale y mi compa Noe, por su amistad, apoyo y tantos aventones a la casa.

Al Lic. Pedro Checa, por haber sido mensajero de un regalo tan especial.

A Gustavo Aguilar Montes, por tu generosidad, cambiaste mi forma de ver la vida.

Al Lic. Marcelo Ferrufino, a Nacho Arévalo y a la alegre Lili, por su amistad, apoyo y optimismo.

**TÍTULO**

---

---

**DISEÑO DE UN INTÉRPRETE GRÁFICO  
MEDIANTE UN SISTEMA DEDICADO,  
IMPLEMENTADO CON EL  
MICROCONTROLADOR MCS-51**

# CONTENIDO

---

---

|  |    |
|--|----|
| INTRODUCCIÓN.....                                  | 1  |
| CAPÍTULO 1: PRESENTACIÓN.....                      | 1  |
| 1.1 Planteamiento del problema .....               | 2  |
| 1.2 Justificación del tema .....                   | 5  |
| 1.3 Objetivos.....                                 | 5  |
| 1.4 Hipótesis.....                                 | 5  |
| 1.5 Metodología .....                              | 6  |
| CAPÍTULO 2: MARCO TEÓRICO .....                    | 8  |
| 2.1 La Comunicación Alternativa y Aumentativa..... | 9  |
| 2.1.1 Productos en el mercado .....                | 10 |
| 2.2 Familias Lógicas .....                         | 12 |
| 2.2.1 Clasificación de las familias lógicas .....  | 15 |
| 2.2.2 ASIC y FPGA .....                            | 17 |
| 2.3 Sistema Dedicado.....                          | 19 |
| 2.4 Microprocesador y microcontrolador .....       | 20 |
| 2.4.1 Microcontrolador 8751H.....                  | 22 |
| 2.4.1.1 Memoria de Programa .....                  | 25 |
| 2.4.1.2 Memoria de Datos .....                     | 26 |
| 2.4.1.3 Modos de direccionamiento .....            | 28 |
| 2.4.1.4 Transferencia de datos .....               | 31 |
| 2.4.1.5 Instrucciones booleanas .....              | 34 |
| 2.4.1.6 Instrucciones de salto .....               | 35 |
| 2.4.1.7 Registros de función especial (SFR) .....  | 36 |
| 2.5 Dispositivos Lógicos Programables .....        | 43 |
| 2.5.1 Matriz Genérica Programable .....            | 44 |
| 2.6 Memoria.....                                   | 45 |
| 2.6.1 Memoria RAM .....                            | 46 |
| 2.6.2 Memoria ROM.....                             | 48 |
| 2.6.3 Memoria FLASH.....                           | 49 |

|  |           |
|--|-----------|
| 2.7 Características de la voz.....                               | 51        |
| 2.8 ¿Qué es muestrear?.....                                      | 52        |
| 2.8.1 Teorema del muestreo.....                                  | 53        |
| <b>CAPÍTULO 3: DESARROLLO.....</b>                               | <b>55</b> |
| 3.1 Diseño de un Intérprete Gráfico.....                         | 56        |
| 3.1.1 Características del intérprete gráfico.....                | 56        |
| 3.1.2 Elementos de Entrada.....                                  | 57        |
| 3.1.3 Elementos de Salida.....                                   | 57        |
| 3.1.4 Elementos del Proceso.....                                 | 58        |
| 3.1.4.1 Microcontrolador (8751H).....                            | 61        |
| 3.1.4.2 Mapeo de Memoria y Entrada/Salida.....                   | 66        |
| 3.1.4.3 Habilitación de periféricos.....                         | 68        |
| 3.1.4.4 Convertidor Analógico Digital (ADC0809).....             | 71        |
| 3.1.4.5 Generador de "CLK" para el circuito ADC.....             | 72        |
| 3.1.4.6 Convertidor Digital Analógico (DAC0800).....             | 75        |
| 3.1.4.7 Amplificador Operacional (TL081).....                    | 76        |
| 3.1.4.8 Controlador de teclado (74C922).....                     | 77        |
| 3.1.4.9 Memoria externa.....                                     | 79        |
| 3.1.4.10 Latch (DM74LS573).....                                  | 81        |
| 3.1.4.11 Display Alfanumérico (AND491).....                      | 83        |
| 3.1.5 Interconexión del intérprete gráfico.....                  | 84        |
| <b>CAPÍTULO 4: RESULTADOS.....</b>                               | <b>85</b> |
| 4.1 Sistema Básico.....  | 86        |
| 4.1.1 Decodificador de 8 líneas (SN74LS138).....                 | 88        |
| 4.1.2 Buffer octal inversor con salidas 3-State (SN74LS540)..... | 90        |
| 4.2 Pruebas esenciales.....                                      | 91        |
| <b>CONCLUSIONES.....</b>   | <b>97</b> |
| <b>LITERATURA CONSULTADA.....</b>                                | <b>99</b> |
| Bibliografía.....  | 100       |
| Páginas en Internet.....   | 100       |
| Hojas de especificación.....                                     | 103       |
| Otros.....   | 106       |

|   |     |
|---|-----|
| APÉNDICE A:   |     |
| DIAGRAMA ELÉCTRICO DEL PREAMPLIFICADOR .....                              | 107 |
| APÉNDICE B:   |     |
| DIAGRAMA ELÉCTRICO DEL AMPLIFICADOR.....                                  | 109 |
| APÉNDICE C:   |     |
| DIAGRAMA ELÉCTRICO DE LA FUENTE DE ALIMENTACIÓN.....                      | 111 |
| APÉNDICE D:   |     |
| CIRCUITO ESQUEMÁTICO DEL MÓDULO PRINCIPAL DEL<br>INTÉRPRETE GRÁFICO ..... | 113 |
| APÉNDICE E:   |     |
| DIAGRAMAS DE FLUJO DEL INTÉRPRETE GRÁFICO .....                           | 115 |
| APÉNDICE F:   |     |
| CIRCUITO ESQUEMÁTICO DEL SISTEMA BÁSICO .....                             | 122 |
| APÉNDICE G:   |     |
| DIAGRAMAS DE FLUJO DEL SISTEMA BÁSICO .....                               | 124 |
| APÉNDICE H:   |     |
| RUTINA DE PROGRAMACIÓN PARA EL SISTEMA BÁSICO.....                        | 130 |
| TABLA DE FIGURAS.....   | 133 |

# INTRODUCCIÓN

---

---

Dentro de la ingeniería en computación se desarrollan aplicaciones útiles a diversos ámbitos del que hacer humano. En la actualidad la creación de Sistemas Dedicados para resolver problemas específicos, de una gran cantidad de campos de acción, es una realidad.

Existen personas con dificultades para comunicarse verbalmente que requieren de sistemas llamados de Comunicación Alternativa y Aumentativa que proporcionan ayuda especial mediante métodos designados para suplir cualquier habilidad del habla.

Por tal motivo, se pensó en el diseño de un Sistema Dedicado a la interpretación gráfica, que reproduzca un mensaje audible cuando sea presionada una imagen, facilitando así la comunicación ya que no será necesario ver la imagen que fue seleccionada, simplemente escuchar tal como se haría con cualquier otra persona. Generando una mayor confianza y la posibilidad de comunicarse con cualquier persona.

EL diseño del Intérprete Gráfico mediante un sistema dedicado emplea el microcontrolador MCS-51 de Intel.

Es un tablero del tamaño de una hoja oficio dividido en 8 partes iguales, cada una de ellas es considerada una tecla. Cada tecla contiene una imagen que puede ser cambiada según la actividad que se realice. A cada imagen se le grabará un mensaje hablado que exprese su significado, así, si la imagen se cambia el mensaje también deberá cambiar.

Además del microcontrolador, se utilizaron dispositivos de almacenamiento, de amplificación y conversión de señales entre otros.

Se obtuvo el diseño y la comprobación de su funcionamiento mediante la implementación de un sistema básico que consta de una sola tecla (un botón).

Este trabajo consta de cuatro capítulos:

En el Capítulo 1 se plantea el problema, se justifica el tema, se mencionan los objetivos y la hipótesis y se explica la metodología utilizada en la realización de este trabajo.

En el Capítulo 2 se presenta el marco teórico o revisión de literatura, es decir, los fundamentos teóricos que sustentan la investigación.

En el Capítulo 3 se mencionan las características de todos los dispositivos que forman parte del diseño del intérprete gráfico y su interconexión.

En el Capítulo 4 se explica la implementación del diseño con un solo botón llamado Sistema Básico, que comprueba la factibilidad del trabajo. Y se presentan los resultados de las pruebas realizadas.

# **CAPÍTULO 1**

---

---

## **PRESENTACIÓN**



## **1.1 Planteamiento del problema**

La comunicación es una necesidad básica del ser humano, y dentro de todos los tipos de comunicación humana la más potente es la comunicación oral.

Cuando una persona tiene afectados los órganos que participan en la comunicación oral puede potenciar o suplir las funciones de éstos mediante gran variedad de ayudas técnicas dentro del área de la Comunicación Aumentativa y Alternativa (CAA).

Se denominan sistemas de Comunicación Aumentativa a las ayudas especiales y los métodos designados para suplir cualquier habilidad del habla.

Los "tableros de comunicación" forman parte de los sistemas de CAA, los cuales son utilizados por personas, principalmente niños, "no verbales", es decir, individuos que debido a algún síndrome, enfermedad o discapacidad, han perdido o no han logrado adquirir una comunicación oral eficiente. Dentro de estos individuos "no verbales" se incluyen pacientes con parálisis cerebral, retraso mental, afasia, síndrome de Down, autismo, etc.

Estos niños comprenden todo lo que sucede a su alrededor, pero debido a la dificultad motriz que presentan no pueden coordinar sus estructuras orofaciales (mandíbula, labios, lengua, etc.) para poder hablar.

Haciendo un análisis de su capacidad de movimiento, se encontró, en la mayoría de los casos, que la parte del cuerpo en la que presentan una mayor coordinación es la mano.

Actualmente, algunos de estos niños, que cuentan con la ayuda de un terapeuta con conocimiento en sistemas de CAA trabajan con una hoja, donde se les presentan imágenes que representan mensajes que los niños podrían comunicar, de acuerdo a la actividad que se este realizando, y de esta manera, los niños, con su mano, señalan la imagen que indica lo que quieren decir.

Por ejemplo, si están comiendo los niños tienen en su hoja imágenes que representan "mensajes" como: "quiero mas", "me gusta", "no me gusta", "ya no quiero", "tengo sed", "me ensucié", "gracias", etc.

De esta manera, se ha logrado que estos niños logren comunicar algunas ideas dentro de su vida diaria como en la hora de comer, de vestir, de ir al baño, de ver la TV, de jugar, de oír música, etc. E incluso algunos han comenzado a realizar actividades académicas como el conocimiento y clasificación de animales y el aprendizaje de la lectoescritura.

No obstante, su comunicación no resulta del todo eficiente, ya que este método de la "hoja" presenta las siguientes desventajas:

1. Los niños se pueden comunicar únicamente con las personas que conocen bien las imágenes de la hoja, como son su terapeuta, sus padres y familia cercana. Ya que ellos, conocen bien las imágenes y el significado de cada una de ellas, e incluso saben en que orden se encuentran.

2. Los niños requieren que al menos una persona los este viendo en el momento en que señalan la imagen, de lo contrario no serviría de nada.

Es conveniente desarrollar un intérprete gráfico que de solución a estas desventajas, haciendo que los "mensajes" representados en imágenes, sean audibles y así:

1. Los niños podrán comunicarse con todas las personas (que puedan escuchar y que entiendan su idioma) sin importar que éstas conozcan o no las imágenes de su hoja.
2. Un mensaje audible llamará la atención de las personas que estén cerca, sin necesidad de estarlos observando en todo momento.

Actualmente, en países de primer mundo como Estados Unidos, Canadá, Australia, etc. existen tableros de comunicación electrónicos; Sin embargo, debido a su alto costo y el trámite de importación, resulta difícil su adquisición para una familia mexicana. Es importante desarrollar un tablero hecho en México que esté al alcance de cualquiera de estas familias.

Con la ayuda de los avances tecnológicos en computación y electrónica, se diseñará un intérprete gráfico como sistema de CAA.

Un sistema dedicado, está enfocado exclusivamente a la realización de una tarea. En este caso, fungir como intérprete gráfico, con la posibilidad de grabar y reproducir los mensajes representados en las imágenes.

## **1.2 Justificación del tema**

Poder colaborar en la realización de dispositivos, basados en tecnología actual y el conocimiento que de ella he obtenido durante la carrera, que puedan mejorar la calidad de vida de algunas personas. Este es el principal motivo para la realización de este trabajo que está inspirado en la siguiente frase "El que no pueda hablar, no significa que no tenga nada que decir"

## **1.3 Objetivos**

Contribuir al desarrollo de equipos e instrumentación de tecnología actual, que facilite la comunicación de personas que debido a algún síndrome, enfermedad o discapacidad han perdido o no han logrado adquirir una comunicación oral eficiente.

Proponer el diseño de un Sistema Dedicado, que reciba como entrada el empuje táctil sobre una de ocho imágenes, y que emita como salida la reproducción de una grabación o mensaje audible que corresponda a la imagen seleccionada.

## **1.4 Hipótesis**

- Un intérprete gráfico que haga audibles los mensajes visuales, es posible realizarlo utilizando el microcontrolador MCS-51 de la familia INTEL.
- Una memoria FLASH de 4Mbits cubre las necesidades de almacenamiento para 2 niveles de 8 mensajes (16 mensajes) de 3 seg. cada uno con una resolución aceptable.

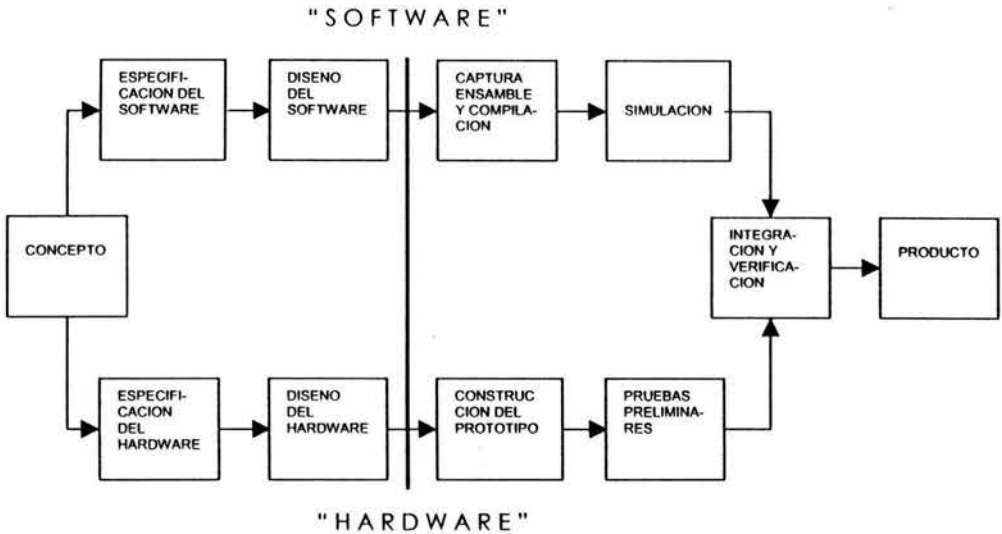
- Se construirá un circuito electrónico que funcione de intérprete gráfico empleando materiales, dispositivos electrónicos, circuitos integrados, etc.

## **1.5 Metodología**

La metodología utilizada en el desarrollo de este proyecto es la siguiente:

- Establecer el concepto del proyecto
- Especificación del "hardware" y "software"
- Diseño del "hardware" •  
Construcción del prototipo  
Pruebas preliminares de pistas y conectividad
- Diseño del "software"  
Captura de programas  
Captura de archivos de programación  
Simulación de rutinas de programación
- Integración y verificación de "hardware" y "software"
- Obtención del producto

Gráficamente se ve de la siguiente manera:



**Figura 1-1 Metodología para el desarrollo del intérprete gráfico**

Este proceso completo (desde el Concepto hasta el Producto) solamente se aplicó en el desarrollo del sistema básico (capítulo IV) como prueba esencial que ayuda a validar el proyecto en su totalidad. El alcance de este último está marcado con la línea vertical.

## **CAPÍTULO 2**

---

---

### **MARCO TEÓRICO**

## **2.1 La Comunicación Alternativa y Aumentativa**

La Comunicación Alternativa y Aumentativa (CAA) se refiere a toda forma de comunicación que apoya, complementa y/o suplementa el habla. La CAA es un área de la práctica clínica que intenta compensar (temporal o permanentemente) la invalidez de los patrones de discapacidad del individuo con desordenes severos en la comunicación expresiva.

La comunicación basada en símbolos forma parte de los sistemas de CAA, los cuales son utilizados por personas, principalmente niños, “no verbales”, es decir, individuos que debido a algún síndrome, enfermedad o discapacidad, han perdido o no han logrado adquirir una comunicación oral eficiente. Dentro de estos individuos “no verbales” se incluyen pacientes con parálisis cerebral, retraso mental, afasia, síndrome de Down, autismo, etc. (Berlanga, 2003)

Algunos sistemas formales de signos y de símbolos pueden representar tanto ideas complejas como simples. Cuanto más simbólico o ideográfico es un sistema, más capacidad combinatoria y de expresión tiene. Las ayudas especiales y los métodos designados para suplir cualquier habilidad del habla que el individuo tenga, se denominan sistemas aumentativos de comunicación. El objetivo de un sistema de comunicación es proporcionar a una persona no hablante un medio eficiente y eficaz de comunicación, de modo que pueda convertirse en un comunicador activo, temporalmente o a largo plazo. También provee de un medio para facilitar el desarrollo del habla. (Larraz, 2003)



## 2.1.1 Productos en el mercado




Mayer-Johnson, Inc., una compañía dedicada a desarrollar productos de comunicación aumentativa (CA), situado en San Diego, California, maneja un catálogo de 50 páginas con mas de 170 productos. Comenzó vendiendo libros con imágenes, después se digitalizaron estas imágenes junto con un software para su manejo y actualmente desarrolla hardware tal como interruptores, pantallas sensibles y dispositivos de habla. (Mayer-Johnson, 2003)

La siguiente tabla muestra tres dispositivos de comunicación aumentativa desarrollados por Mayer-Johnson, todos en esencia hacen lo mismo y lo que varia entre ellos, es la cantidad y tamaño de imágenes así como el número de niveles.

| Tech/Four™   | Tech/TALK™   | Tech/Speak™   |
|--|--|---|
|  |  |  |
| 1 nivel-4 mensajes<br>(4 seg/msg)  | 8 mensajes/múltiples niveles<br>(4 seg/msg)  | 32 mensajes/múltiples niveles<br>(2 seg/msg)  |
| \$ 198.00 US   | 6 niveles \$ 395.00 US<br>8 niveles \$ 445.00 US                                   | 2 niveles \$ 445.00 US<br>4 niveles \$ 545.00 US<br>6 niveles \$ 645.00 US          |

Figura 2-1 Ejemplos de productos de CA Mayer-Johnson

Enabling devices, una división de Toys for Special Children, Inc., es otra compañía que desde hace muchos años se dedica a proveer recursos para el aprendizaje y dispositivos de asistencia para los desafíos físicos que se presentan en la gente discapacitada. La siguiente tabla muestra, el nombre comercial, la imagen, una breve descripción y el precio de algunos dispositivos que enabling devices tiene a la venta. (Enabling Devices, 2003)

| Talking Plate (5"x8")   | Talking Picture Card Holder  | Cheap Talk 8, Say It Play It   |
|---|--|--|
|    |                           |    |
| <p>Simplemente presionando el súper botón de 5"x 8" se activará cualquier mensaje (de mas de 20 segundos) que se haya grabado previamente.</p> <p>\$ 69.95 US</p> | <p>Presionar el botón permitirá al usuario escuchar el mensaje previamente grabado.</p> <p>\$ 76.95 US</p> | <p>Ocho imágenes fácilmente intercambiables que al tacto reproducirá el mensaje previamente grabado.</p> <p>\$ 179.95 US</p> |

**Figura 2-2 Ejemplos de productos de CA Enabling Devices**

Las páginas web de donde se obtuvo esta información no mencionan el uso específico de cada uno de estos dispositivos, pero sí que forman parte de la Comunicación Aumentativa.

Estas grandes industrias, al igual que otras, cuentan con derechos reservados (copyright) de sus productos, y por supuesto que no mencionan ni el diseño ni los componentes internos de los circuitos de

sus productos. Sin embargo es muy probable que utilicen técnicas de minimización que logren integrar en una sola pastilla todos los componentes interconectados y programados que estén listos para realizar las tareas que determinado producto requiere.

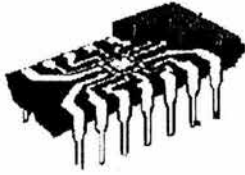
Algunas de estas técnicas de minimización de circuitos se mencionan a continuación.

## **2.2 Familias Lógicas**

La era de la electrónica con semiconductores comenzó con la invención del transistor en 1948 y a partir de ahí la evolución de la tecnología electrónica inicia una rápida carrera. En 1952, se sustituye el empleo de germanio por el silicio y en 1958 se fabricó el primer JFET (transistor de unión de efecto campo), lo que condujo a la aparición del transistor metal-óxido semiconductor de efecto campo (MOSFET). Las continuas mejoras en el diseño y fabricación de los sistemas de computación, han hecho de los MOSFET los dispositivos más universalmente empleados.

Para mejorar las conexiones de diferentes componentes electrónicos se propuso la fabricación de todos los componentes del circuito, junto con su interconectado, sobre una misma oblea de silicio. A esta solución se la denominó circuito integrado monolítico, y en 1959 la empresa Texas Instruments ® desarrolla el primer circuito integrado con tecnología RTL (lógica resistencia-transistor bipolar).

La Figura 2-3 muestra una sección de un circuito integrado, donde se ve el chip del circuito dentro del encapsulado. Los terminales del chip se conectan a los pines del encapsulado para permitir las conexiones con las entradas y salidas del mundo exterior.



**Figura 2-3 Sección de un circuito integrado**

Los sistemas digitales actuales utilizan casi exclusivamente circuitos integrados en su diseño debido a su reducido tamaño, alta fiabilidad, bajo costo y reducido consumo de potencia.

La evolución experimentada en el campo de los circuitos integrados ha sido extraordinaria, pasándose por crecientes escalas de integración. Hoy es posible colocar millones de componentes tanto activos como pasivos dentro de un solo chip, lo que permite construir computadoras sobre un área de silicio de sólo unos pocos milímetros cuadrados.

A comienzos de los sesenta es cuando empiezan a aparecer los primeros circuitos integrados SSI (Figura 2-4), con menos de 100 componentes y no más de 30 compuertas por chip.

En 1966 surgen los circuitos integrados MSI, que permiten hasta 1000 componentes, lo que supone entre 30 y 300 compuertas por chip. Tres años más tarde fue posible fabricar circuitos LSI, lo que permitió los primeros microprocesadores. El incremento en la escala de integración hasta los circuitos integrados VLSI, en 1975, facilita que aparezcan microprocesadores mayores y memorias RAM de 16Kb. El término ULSI es de reciente aparición y llega hasta el límite de la tecnología actual, es decir, hasta los centenares de millones de dispositivos por chip.

| <b>Nivel de integración</b>           | <b>No. de compuertas</b> | <b>Aplicaciones</b>                       |
|---------------------------------------|--------------------------|---|
| Pequeña escala de integración (SSI)   | 1 – 30                   | Compuertas básicas y <i>flip-flops</i> .  |
| Media escala de integración (MSI)     | 30 – 300                 | Contadores, registros, memorias pequeñas. |
| Gran escala de integración (LSI)      | 300 – 10,000             | Memorias y microprocesadores sencillos.   |
| Muy alta escala de integración (VLSI) | 10,000 – 1,000,000       | Memorias grandes, microprocesadores.      |
| Escala de integración ultra (ULSI)    | > 1,000,000              |   |

**Figura 2-4 Niveles de integración para dispositivos digitales**

Aunque los modernos componentes electrónicos digitales son el resultado de años de desarrollo y evolución, no hay un conjunto ideal de circuitos que satisfaga todos los requerimientos. Por tanto existen varias familias lógicas, cada una de las cuales ofrecen ventajas particulares. La velocidad, consumo de potencia y densidad de componentes son cuestiones a tener en cuenta. Algunas familias trabajan a velocidades muy altas, mientras que otras poseen bajo consumo. Parte de la función del diseñador consiste en seleccionar una familia lógica apropiada para una aplicación dada.

La primera familia lógica fue la RTL. De estas primeras celdas lógicas se derivó la lógica TTL (transistor-transistor) cuya principal característica es el empleo de transistores bipolares con múltiples emisores.

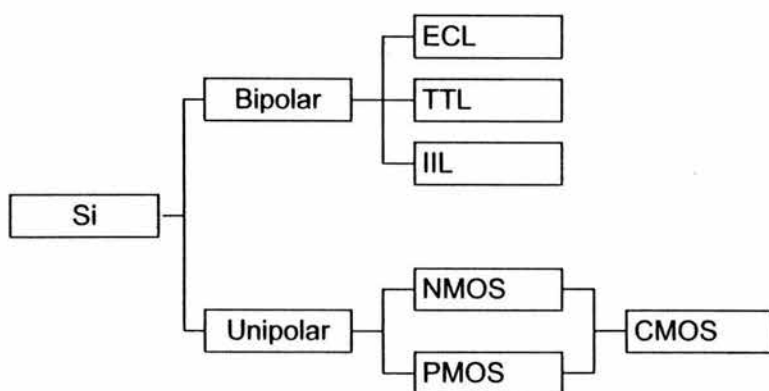
La utilización de MOSFET resultaba interesante debido a que permite mayores densidades de componentes (dispositivos de menor tamaño y mucho menor consumo que los bipolares). Originalmente se fabricaron PMOS (MOSFET de canal p), pero la mejora en los procesos de fabricación condujo a los NMOS (MOSFET de canal n) por presentar mayor velocidad.

Las configuraciones MOS de simetría complementaria (CMOS), que emplean ambos tipos de dispositivos (PMOS y NMOS) se han situado a la cabeza de las tecnologías digitales en la actualidad.

### **2.2.1 Clasificación de las familias lógicas**

Los circuitos integrados (CI) digitales se dividen en familias lógicas. Cada familia particular está basada en un tipo particular de circuito. Todos los elementos de una familia lógica son compatibles entre sí, es decir, operan con los mismos niveles lógicos, pudiendo la salida de un elemento alimentar la entrada de otro.

Existen por un lado familias lógicas basadas en silicio (la dominante en la actualidad) y basadas en arseniuro de galio (GaAs), para aplicaciones de muy alta velocidad. En la Figura 2-5 se muestra una clasificación de circuitos integrados basados en silicio según la tecnología de fabricación.



**Figura 2-5 Clasificación de CI según la tecnología de fabricación**

Dentro de los circuitos integrados basados en este material semiconductor se definieron dos categorías, la tecnología bipolar y la unipolar.

Los dispositivos de tecnología bipolar se caracterizan porque presentan unas altas velocidades de operación gracias a los transistores de unión (BJTs), pero también son elevados los consumos de potencia. Pertenecen a esta categoría las familias lógicas ECL (lógica de emisorés acoplados), TTL (Lógica transistor-transistor) e IIL (Lógica de Inyección Integrada).

Por lo que respecta a los dispositivos unipolares, los más importantes son los que se basan en los dispositivos de efecto campo MOS. Dentro de esta tecnología se incluyen la MOS de canal n (NMOS), la de canal P (PMOS, que ha quedado obsoleta) y la tecnología MOS de simetría complementaria (CMOS), en la que se integran los dos tipos de

canales. La más empleada es la tecnología CMOS por consumo de potencia y velocidad de operación.

Entre ambos tipos de tecnologías, unipolares y bipolares, se encuentra otra reciente denominada BiCMOS, que permite disponer en un mismo circuito integrado dispositivos bipolares (para las entradas y las salidas) y estructuras CMOS. El costo es superior al de la tecnología CMOS, pero proporciona mejores corrientes de salida y mayor velocidad. (UPC, 2003)

Dentro de la muy alta escala de integración, existen diversas posibilidades de diseño. Por ejemplo los Circuitos Integrados de Aplicación Específica (ASIC Application Specific Integrated Circuits), circuitos que son realizados a medida y para una aplicación concreta, cuya utilización es cada vez más general. (Alcubilla, 1996)

### **2.2.2 ASIC y FPGA**

Hay dos métodos para la ejecución de algoritmos en la computación tradicional. El primero es usar un Circuito Integrado de Aplicación Específica o ASIC que opera en hardware. Se diseñan ASICs para realizar una aplicación específica. Estos son muy rápidos y eficaces siempre que realicen exactamente la aplicación para la que han sido diseñados. Sin embargo, después de la fabricación el circuito no puede alterarse. Si parte del circuito necesita una modificación será necesario un nuevo diseño y fabricación de otro chip. Éste es un proceso caro, sobre todo cuando uno considera las dificultades de reemplazar los ASICs de un gran número de sistemas desplegados.



Los microprocesadores presentan una solución más flexible. Para realizar las operaciones pertinentes los procesadores ejecutan un juego de instrucciones. Si cambiamos las instrucciones del software, modificaremos la funcionalidad del sistema sin cambiar el hardware. Sin embargo, la desventaja de esta flexibilidad es que la velocidad de actuación está por debajo del de un ASIC. El procesador debe leer cada instrucción de la memoria, determinar su significado, y sólo entonces lo ejecuta. Adicionalmente, el juego de instrucciones que pueden usarse por un programa se determina en el momento de la fabricación del procesador. Cualquier otro funcionamiento que sea llevado a cabo deberá construirse fuera de las instrucciones ya existentes. Esto produce un tiempo alto de ejecución.

La computación reconfigurable llena el hueco entre el hardware (ASIC) y el software (Microprocesador), mientras consigue una actuación, potencialmente, más rápida que los microprocesadores mantiene un nivel más alto de flexibilidad que los ASICs. Este tipo de computación se consigue con la utilización de FPGAs (Field Programmable Gate Arrays). Estos dispositivos contienen una serie de bloques lógicos cuya funcionalidad se determina a través de múltiples celdas de SRAM. Estos elementos, también conocidos como bloques lógicos, se conectan entre sí mediante recursos de interconexión programables. De este modo, se pueden trazar los circuitos con las funciones lógicas dentro de los bloques lógicos y posteriormente programar las conexiones entre dichos bloques lógicos formando el circuito necesario dentro de la FPGA. (Aguirre y Ezquerro, 2003)

Aunque todo los componentes del diseño en este trabajo se podrían integrar en una solo circuito, ya sea un ASIC o un FPGA, esta opción no se consideró debido a que éstos son económicamente redituables cuando se producen cientos o miles de ellos.

En el diseño del intérprete gráfico se pretende utilizar dispositivos comercialmente disponibles, en este caso un microprocesador, que en conjunto formen un Sistema Dedicado.

### **2.3 Sistema Dedicado**

Un sistema dedicado se puede definir como sistema electrónico, compuesto por hardware y software, diseñado para cumplir con una tarea específica. Usualmente son parte de un sistema más complejo. Al igual que una computadora personal, la arquitectura de hardware de un sistema dedicado usualmente se elabora en torno a un microprocesador o microcontrolador. Además estos sistemas se diferencian de sistemas de propósito general (como las computadoras personales) en que buscan explotar de mejor manera el uso del espacio, el consumo, la eficiencia y el costo.

Los sistemas dedicados están detrás de otros innumerables equipos de uso cotidiano como microondas, lavadoras, automóviles, teléfonos celulares, sistemas de navegación, plataformas de juegos gráficos, reproductores MP3, sistemas de radionavegación GPS (Global Position System), equipos VCR, televisores, equipos DVD, periféricos de computadores, equipos médicos, maquinas expendedoras, etc.

Componente vital de un sistema dedicado es el software. El contenido de software de los sistemas dedicados ha ido en aumento con los años

porque simplifica el diseño y da flexibilidad. Además se brinda una oportunidad única para incorporar inteligencia detrás de los sistemas. (Gold-Chile, 2003)

En este caso se propone elaborar el hardware del sistema con un microcontrolador de la familia MCS-51 de INTEL que se menciona a continuación y para el software su compilador avmac51.

## 2.4 Microprocesador y microcontrolador

A cualquier circuito integrado que contenga una Unidad de Procesamiento Central (CPU) se denomina con el término de microprocesador.

En la siguiente figura vemos la estructura de la Unidad de Procesamiento Central, CPU.

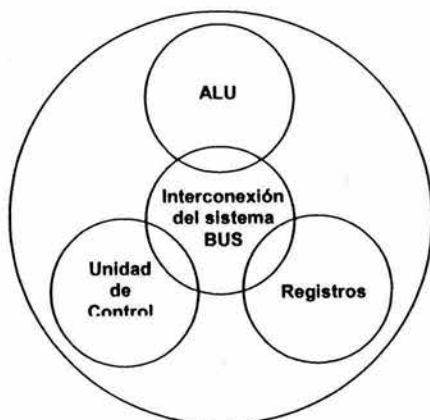


Figura 2-6 Estructura de la CPU

En el año de 1976, gracias al aumento de la densidad de integración de componentes en un circuito integrado, se desarrolló la primera

computadora en un chip, es decir, se integraron junto con el microprocesador los subsistemas que anteriormente formaban unidades especializadas e independientes, pero unidas al microprocesador por las pistas de un circuito impreso, formando lo que se conoce como sistema.

A este nuevo circuito integrado se lo denominó microcomputador monopastilla o **microcontrolador**. (Jiménez y Pérez, 2003)

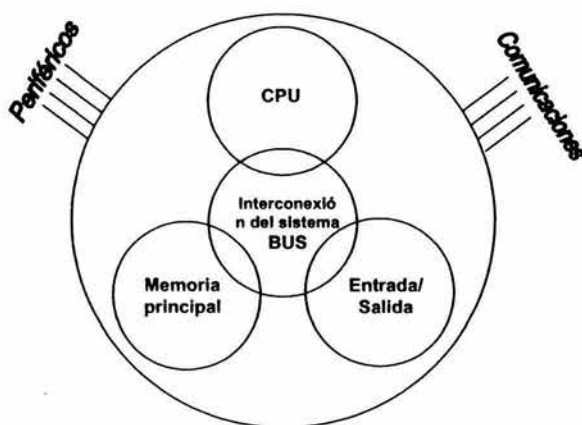


Figura 2-7 Organización de una computadora

Un microcontrolador es un circuito integrado en el que se aloja elementos de una computadora completa interconectados entre sí, con prestaciones limitadas, que se suele destinar a tareas específicas y que se comunica con el exterior a través de sus patas.

El microcontrolador integra los siguientes subsistemas, aunque no siempre los presenta todos relacionados: CPU, Memoria principal (RAM, ROM o EPROM), entradas / salidas, gestión de interrupciones.

Las limitaciones tecnológicas impiden que los módulos lógicos contenidos en un microcontrolador tengan una elevada potencia y capacidad, por lo que existe una gran cantidad de modelos para seleccionar en cada caso el que se ajuste mejor a los requerimientos del problema. (Fajardo, 2003)

## 2.4.1 Microcontrolador 8751H

Como ya se mencionó, el diseño de este sistema se llevará a cabo utilizando el **microcontrolador 8751H** que pertenece a la familia MCS-51 de INTEL. Decir que un circuito integrado pertenece a cierta “familia” significa que comparte una estructura interna muy similar con otros circuitos integrados donde todos forman una familia: así, al hablar de la familia 51 fabricado por INTEL, se habla del 8051, 8031, 8751, entre otros.

| Nombre       | Memoria Interna   |              | Temporiza/<br>Contadores | Interrup-<br>ciones | Líneas<br>E/S | Bancos de<br>registros |
|--------------|-------------------|--------------|--------------------------|---------------------|---------------|------------------------|
|              | Programa          | Datos        |                          |                     |               |                        |
| 8031         | NO                | 128x8        | 2 x 16 bits              | 5                   | 32            | 4                      |
| 8031AH       | NO                | 128x8        | 2 x 16 bits              | 5                   | 32            | 4                      |
| 8051         | 4Kx8 ROM          | 128x8        | 2 x 16 bits              | 5                   | 32            | 4                      |
| 8051AH       | 4Kx8 ROM          | 128x8        | 2 x 16 bits              | 5                   | 32            | 4                      |
| <b>8751H</b> | <b>4Kx8 EPROM</b> | <b>128x8</b> | <b>2 x 16 bits</b>       | <b>5</b>            | <b>32</b>     | <b>4</b>               |
| 8751H-12     | 4Kx8 EPROM        | 128x8        | 2 x 16 bits              | 5                   | 32            | 4                      |
| 8032AH       | NO                | 256x8        | 3 x 16 bits              | 6                   | 32            | 4                      |
| 8052AH       | 8Kx8 ROM          | 256x8        | 3 x 16 bits              | 6                   | 32            | 4                      |

**Figura 2-8 Características de los dispositivos de la familia MCS-51**

La figura 2-8 muestra una tabla con las diferentes características de los microcontroladores que forman parte de la familia MCS-51.

En la siguiente figura se observa que el 8751H es considerado microcontrolador ya que cuenta con varios elementos interconectados.

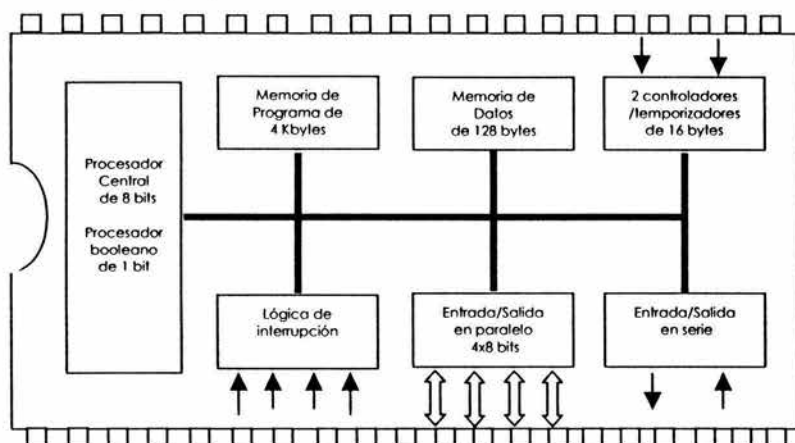


Figura 2-9 Esquema a bloques del 8751H

La forma en que los pines están distribuidos se muestra a continuación:

|            |    |    |             |
|------------|----|----|-------------|
| P1.0       | 1  | 40 | Vcc         |
| P1.1       | 2  | 39 | P0.0 AD0    |
| P1.2       | 3  | 38 | P0.1 AD1    |
| P1.3       | 4  | 37 | P0.2 AD2    |
| P1.4       | 5  | 36 | P0.3 AD3    |
| P1.5       | 6  | 35 | P0.4 AD4    |
| P1.6       | 7  | 34 | P0.5 AD5    |
| P1.7       | 8  | 33 | P0.6 AD6    |
| RST        | 9  | 32 | P0.7 AD7    |
| TXD P3.0   | 10 | 31 | /EA - Vpp   |
| TXD P3.1   | 11 | 30 | ALE - /PROG |
| /INT0 P3.2 | 12 | 29 | /PSEN       |
| /INT1 P3.3 | 13 | 28 | P2.7 A15    |
| TO P3.4    | 14 | 27 | P2.6 A14    |
| T1 P3.5    | 15 | 26 | P2.5 A13    |
| /WR P3.6   | 16 | 25 | P2.4 A12    |
| /RD P3.7   | 17 | 24 | P2.3 A11    |
| XTAL2      | 18 | 23 | P2.2 A10    |
| XTAL1      | 19 | 22 | P2.1 A9     |
| Vss        | 20 | 21 | P2.0 A8     |

Figura 2-10 Diagrama de Conexión del MCS-51

Relacionando las dos figuras anteriores tenemos:

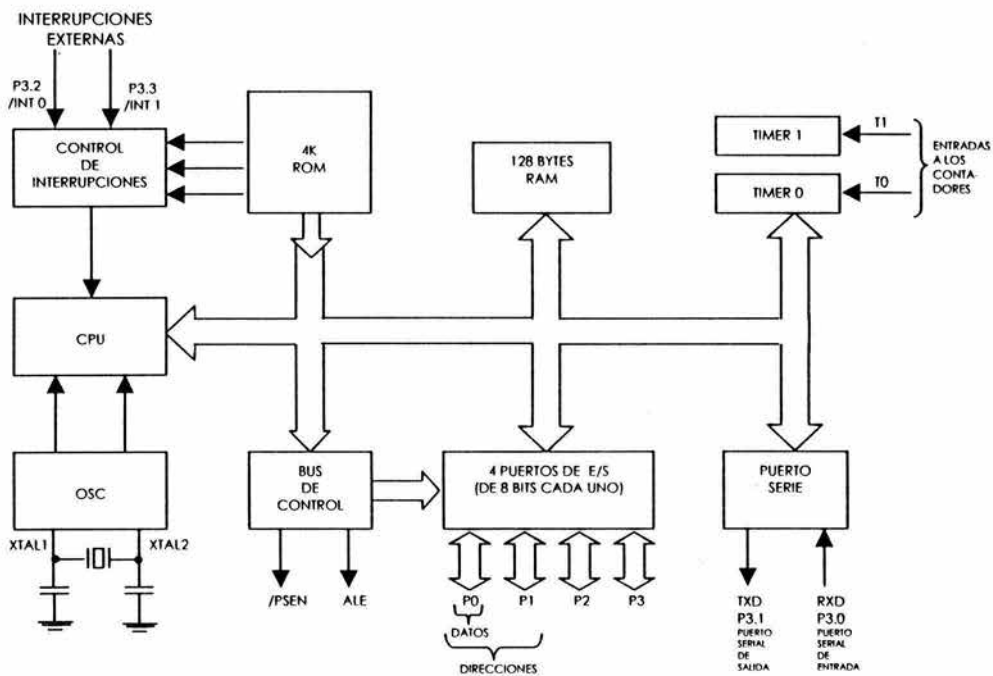


Figura 2-11 Diagrama a Bloques de la esencia del 8051

La figura 2-12 nos muestra la relación de las áreas de almacenamiento. La columna de la extrema derecha muestra el valor en decimal (d) y el de la extrema izquierda en hexadecimal (h).

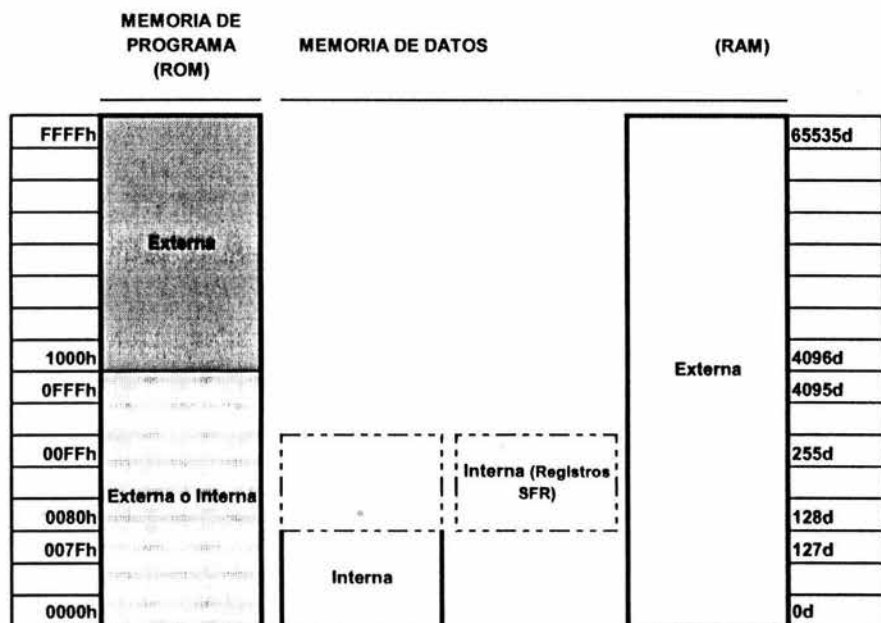


Figura 2-12 Relación de áreas de almacenamiento (8051)

Como se puede ver en la figura 2-11, con los pines de direccionamiento 0 y 1 se tienen 16 pines de direccionamiento, esto es puede direccionar hasta 65,535 bytes ( $2^{16} = 65,535$  bytes), tanto de ROM como de RAM externas.

### 2.4.1.1 Memoria de Programa

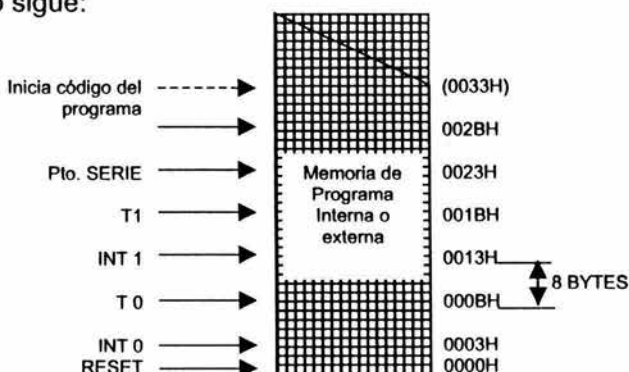
La memoria de programa (o ROM) se puede utilizar 100% externa, una parte interna y otra externa o solo la interna, que como se ve en la figura 2-12 tiene 4,095 bytes de memoria de programa (ROM) interna.



**Tabla de direcciones de subrutinas de interrupción**

| Fuente                    | Dirección |
|---------------------------|-----------|
| Interrupción externa 0    | 0003H     |
| Desbordamiento en Timer 0 | 000BH     |
| Interrupción externa 1    | 0013H     |
| Desbordamiento en Timer 1 | 001BH     |
| Puerto serie              | 0023H     |

En la parte baja de la memoria de programa se encuentran las localidades de interrupciones que se mencionan en la **Tabla de direcciones de subrutinas de interrupción** y que gráficamente se verían como sigue:

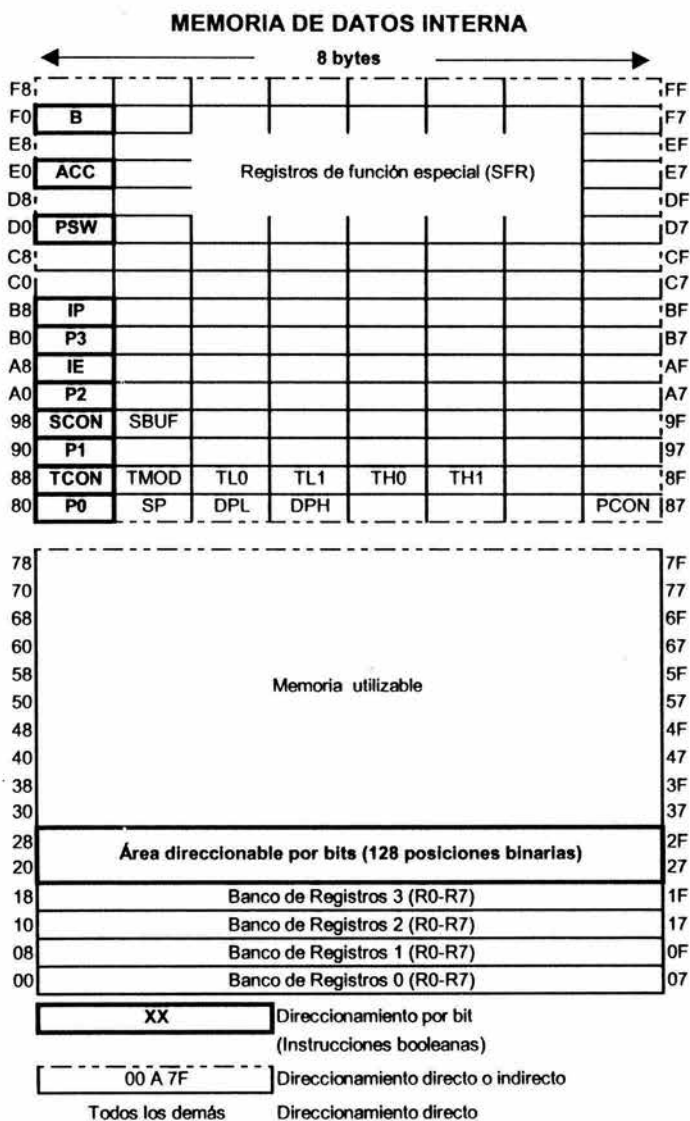


**Figura 2-13 Memoria de Programa (8051)**

### 2.4.1.2 Memoria de Datos

De la misma forma que a la memoria de programa, se puede direccionar una Memoria de Datos (o RAM) Externa de hasta 65,535 bytes, y además tiene una Memoria de Datos Interna de 255 bytes, de los cuales solo se considera una parte de los 128 bytes inferiores como utilizables ya que algunos tienen funciones específicas como se muestra en la figura 2-14 y los 128 bytes superiores son registros de

función especial SFR (por sus siglas en inglés Special Function Register).



**Figura 2-14 Mapa de memoria de datos interna (8051)**

También podemos ver en la figura 2-14 que los SFR utilizan únicamente direccionamiento directo y solo en algunos de ellos (registros de la primera columna de izquierda a derecha) también pueden ser direccionados bit a bit (utilizando las instrucciones booleanas).

### 2.4.1.3 Modos de direccionamiento

#### a) Direccionamiento inmediato

##### Ejemplo:

El valor de una constante que se proporciona en código (memoria de programa). En la sintaxis del código fuente, el direccionamiento inmediato utiliza el carácter "#".

|                 |  |
|-----------------|--|
| MOV A, #100     | El Acumulador es cargado con el dato 100 decimal.  |
| MOV A, #64H     | El Acumulador es cargado con el dato 100 decimal (que es igual a 64 hexadecimal).          |
| ADD A,#120      | El Acumulador es sumado al número decimal 120 y el resultado se almacena en el Acumulador. |
| MOV DPTR,#1245H | El DPTR es cargado con el dato 1245H en forma.   |

#### b) Direccionamiento directo

En el direccionamiento directo se especifica la dirección del operando mediante un campo de dirección de 8-bits en la instrucción. Solo la RAM interna y los registros SFR pueden ser direccionados directamente.

### Ejemplo:

|              |   |
|--------------|---|
| ADD A, 7FH   | El Acumulador es sumado al dato que se encuentra en la dirección 7FH de la RAM interna y el resultado será almacenado en el Acumulador. |
| MOV A, 2EH   | El Acumulador es cargado con el dato que se encuentra en la dirección 2EH de la memoria RAM interna.                                    |
| MOV 3DH, 4EH | El contenido de la dirección 3DH es cargado con el dato que se encuentra en la dirección 4EH.   |

### c) Direccionamiento indirecto

En el direccionamiento indirecto la instrucción especifica un registro que contiene la dirección del operando. La memoria RAM, interna y externa ser puede direccionar indirectamente.

El registro de dirección, para direcciones de 8-bits puede ser R0 o R1 del bando de registros, o el Stack Pointer (que forma parte de los registros de función especial).

El registro de dirección, para direcciones de 16-bits solo puede ser el registro de 16-bits, DPTR ("Data Pointer").

En la sintaxis del código fuente, el direccionamiento indirecto utiliza el carácter "@". **Ejemplo:**

|              |   |
|--------------|---|
| ADD A,@R0    | El Acumulador es sumado con el dato que se encuentra en la dirección apuntada por R0.       |
| MOV A,@R0    | El Acumulador es cargado con el dato que se encuentra en la dirección apuntada por R0.      |
| MOVX @DPTR,A | El contenido del Acumulador es guardado en la dirección apuntada por el DPTR                |
| MOVX A,@DPTR | El Acumulador es cargado con el dato que se encuentra en la dirección apuntada por el DPTR. |

#### d) Direccionamiento indexado

Solo la memoria de programa se puede acceder con direccionamiento indexado y en modo de solo lectura.

Este modo de direccionamiento es utilizado en las lecturas de tablas de la memoria de programa o datos que se encuentran como constantes.

Un registro de 16 bits ( el DPTR o el PC), apunta la base de la tabla y mediante el acumulador se establece el número de la entrada de la tabla.

La dirección de la entrada de la tabla en la memoria de programa está formada por la suma del acumulador y el Apuntador de Base (DPTR o PC). **Ejemplo:**

|               |  |
|---------------|--|
| MOV A,@A+DPTR | Mueve una constante que se encuentra en la memoria de programa. El Acumulador es cargado con el dato que se encuentra apuntado por la dirección formada por la suma del Acumulador y el Apuntador de Datos (DPTR). |
| MOV A,@A+PC   | El Acumulador es cargado con dato que se encuentra en la dirección formada por la suma del mismo Acumulador y el Contador del Programa (PC).   |

#### e) Direccionamiento por registro

Los 8 registros (R0-R7), pueden ser accedidos mediante ciertas instrucciones que simplifican sus códigos de operación (opcode) y en la mayoría de los casos son más rápidas.

### Ejemplo:

|          |  |
|----------|--|
| ADD A,R7 | El acumulador es cargado con el resultado de suma del Acumulador y el contenido del registro R7. |
| DEC R0   | Decrementa el registro R0.   |

## 2.4.1.4 Transferencia de datos

- RAM interna

Para poder mover datos de la memoria de datos interna RAM o SFR, existen 8 instrucciones principales:

| Mnemónico              | Operación                              | Modo de direccionamiento |     |     |     |
|------------------------|--|--------------------------|-----|-----|-----|
|                        |  | Dlr                      | Ind | Reg | Inm |
| MOV A, <fuente>        | A = <fuente>                           | X                        | X   | X   | X   |
| MOV <destino>,A        | <destino> = A                          | X                        | X   | X   |     |
| MOV <destino>,<fuente> | <destino> = <fuente>                   | X                        | X   | X   | X   |
| MOV DPTR,#dato 16 bits | DPTR = constante inmediata de 16 bits. |                          |     |     | X   |
| PUSH <fuente>          | INC SP: "@SP", <fuente>                | X                        |     |     |     |
| POP <destino>          | MOV <destino>, "@SP"; DEC SP           | X                        |     |     |     |
| XCH A, <byte>          | ACC y <byte> intercambia sus datos.    | X                        | X   | X   |     |
| XCHD A, @Ri            | ACC y @Ri intercambian el nibble bajo. |                          | X   |     |     |

La instrucción MOV <dest>,<fuente> ; permite transferir datos de la memoria interna RAM y del SFR sin pasar a través del Acumulador, mientras que las instrucciones MOV A,<fuente> y MOV <dest>,A utilizan al Acumulador para el movimiento de datos dentro de la memoria interna RAM.

La instrucción PUSH, primeramente incrementa el Stack Pointer (SP), y después guarda el dato dentro de la localidad de memoria apuntada por el Stack. La instrucción POP, primero toma el dato de la memoria y después decrementa el SP.

Cabe señalar que si el SP apunta en la localidad 7FH, no se podrán seguir almacenando datos. Arriba de la dirección 80H los datos almacenados con la instrucción PUSH se pierden y los datos tomados con POP, son indeterminados.

La instrucción XCH A, <dirección> permite al Acumulador y al dato apuntado por la dirección de intercambiarse entre sí. La instrucción XCHD A,@R0 intercambia los 4 bits menos significativos (low nibble), de Acumulador con los 4 bits menos significativos del dato apuntado por el registro R0.

- **RAM externa**

Las instrucciones para la transferencia de datos en la memoria RAM externa, son 4 básicamente:

| Dirección de: | Mnemónico    | Operación                        |
|---------------|--------------|----------------------------------|
| 8 bits        | MOVX A, @Ri  | Lee RAM externa @Ri, A=<@Ri>     |
| 8 bits        | MOVX @Ri,A   | Escribe RAM externa @Ri, <@Ri>=A |
| 16 bits       | MOVX A,@DPTR | Lee RAM externa @Ri, A=<@DPTR>   |
| 16 bits       | MOVX @DPTR,A | Escribe RAM externa, <@DPTR>=A   |

En todos los accesos a RAM externa, siempre se utiliza el acumulador como destino o como fuente.

Las habilitaciones de lectura y escritura en RAM externa son activadas solamente durante la activación de una instrucción MOVX.

▪ **Movimiento de tablas localizadas en memoria**

La mayoría de las veces dentro de nuestro programa principal se tienen algunas constantes, como son palabras de control, contadores, banderas iniciales, tablas de datos, etc. Por tal motivo, el 8051 utiliza las siguientes instrucciones solamente en lectura (ya que en escritura no tendrá caso por ser una memoria EPROM o ROM).

| Mnemónico         | Operación                             |
|-------------------|---------------------------------------|
| MOVC A, @A + DPTR | Lee Memoria de Programa en (A + DPTR) |
| MOVC A, @A + PC   | Lee Memoria de Programa en (A + PC)   |



### 2.4.1.5 Instrucciones booleanas

El microcontrolador 8051 integra un procesador booleano para el cual el papel de acumulador es realizado por el bit de acarreo CY (Carry). Este bit que se encuentra en el registro PSW (Program Status Word) y es usado como acumulador del procesador booleano.

El microcontrolador posee una serie de instrucciones que permiten la realización de operaciones lógicas sobre bits, es decir, puede obtenerse una manipulación bit a bit. La dirección de estos bits se encuentra, por una parte, en la tabla de registros de funciones especiales (SFR) y, por otra, en la RAM interna (Dirección 20H-2FH).

Las operaciones booleanas (o de direccionamiento por bit) son:

| Mnemónico |          | Operación                       |
|-----------|----------|---------------------------------|
| ANL       | C,bit    | C = C. AND. bit                 |
| ANL       | C,/bit   | C = C. AND. NOT.bit             |
| ORL       | C,bit    | C = C. OR. bit                  |
| ORL       | C,/bit   | C = C. OR. NOT. bit             |
| MOV       | C,bit    | C = bit                         |
| MOV       | bit,C    | bit = C                         |
| CLR       | C        | C = 0                           |
| CLR       | bit      | bit = 0                         |
| SETB      | C        | C = 1                           |
| SETB      | bit      | bit = 1                         |
| CPL       | C        | C = NOT.C                       |
| CPL       | bit      | bit = NOT.bit                   |
| JC        | rel      | Si C=1, brinca a rel            |
| JNC       | rel      | Si C=0, brinca a rel            |
| JB        | bit, rel | Si bit=1, brinca a rel          |
| JNB       | bit, rel | Si bit=0, brinca a rel          |
| JBC       | bit,rel  | Si bit=1, brinca a rel y limpia |

## 2.4.1.6 Instrucciones de salto

### ▪ Saltos incondicionados

Las instrucciones que permiten hacer los saltos incondicionados son las siguientes:

| Mnemónico      | Operación                         |
|----------------|-----------------------------------|
| JMP direc      | Salta a la dirección direc.       |
| JMP @A + DPTR  | Salta a la dirección A + DPTR     |
| CALL dirección | Llama a la subrutina "dirección". |
| RET            | Regreso de la subrutina.          |
| RETI           | Regreso de la interrupción.       |
| NOP            | Sin operación.                    |

La instrucción RETI es usada para el regreso de una rutina de servicio de interrupción. La diferencia con la instrucción RET es que RETI llama al sistema de control de interrupción mientras que la interrupción está en proceso. Si no existe ninguna interrupción en proceso, entonces la instrucción RETI es igual a RET.

### ▪ Saltos condicionados

El juego de instrucciones con saltos que están condicionados a la activación o desactivación de algunas de las banderas del PSW son las siguientes:

| Mnemónico | Operación              |
|-----------|------------------------|
| JZ rel    | Salta si A es cero.    |
| JNZ rel   | Salta si A no es cero. |

|                       |  |
|-----------------------|--|
| DJNZ <byte>,rel       | Decrementa y salta si no es cero.              |
| CJNE A,<byte>,rel     | Compara y salta si A no es igual a <byte>.     |
| CJNE <byte>,#data,rel | Compara y salta si <byte> no es igual a #data. |

### 2.4.1.7 Registros de función especial (SFR)

Los registros de función especial (SFR) se encuentran en la RAM Interna entre las direcciones 80H y FFH.



**Figura 2-15 Registros de función especial**

En la figura 2-16 se observa una tabla que resume las características los registros de función especial.

| Símbolo | Descripción            | Dirección | Dirección bit a bit, símbolo, o alternativa de función de puerto |                  |     |     |      |                  |      |     | Valor después del Reset |
|---------|------------------------|-----------|--|------------------|-----|-----|------|------------------|------|-----|-------------------------|
|         |                        |           | MSB  |                  |     |     | LSB  |                  |      |     |                         |
| ACC*    | Acumulador             | E0H       |  |                  |     |     |      |                  |      |     | 00H                     |
| B*      | registro B             | F0H       |  |                  |     |     |      |                  |      |     | 00H                     |
| DPTR    | Data pointer (2 bytes) |           |  |                  |     |     |      |                  |      |     |                         |
| DPH     | Data pointer high      | 83H       |  |                  |     |     |      |                  |      |     | 00H                     |
| DPL     | Data pointer low       | 82H       |  |                  |     |     |      |                  |      |     | 00H                     |
| IE*     | Interrupt enable       | A8H       | EA   | -                | -   | ES  | ET1  | EX1              | ET0  | EX0 | 0x000000B               |
| IP*     | Interrupt priority     | B8H       | -  | -                | -   | PS  | PT1  | PX1              | PT0  | PX0 | xx000000B               |
| P0*     | Puerto 0               | 80H       | AD7  | AD6              | AD5 | AD4 | AD3  | AD2              | AD1  | AD0 | FFH                     |
| P1*     | Puerto 1               | 90H       | -  | -                | -   | -   | -    | -                | T2EX | T2  | FFH                     |
| P2*     | Puerto 2               | A0H       | A15  | A14              | A13 | A12 | A11  | A10              | A9   | A8  | FFH                     |
| P3*     | Puerto 3               | B0H       | RD   | WR               | T1  | T0  | INT1 | INT0             | TxD  | RxD | FFH                     |
| PCON    | Power control          | 87H       | SMOD   | -                | -   | -   | GF1  | GF0              | PD   | IDL | 0xxxxxxxB               |
| PSW*    | Program status word    | D0H       | CY   | AC               | F0  | RS1 | RS0  | OV               | -    | P   | 00H                     |
| SBUF    | Serial data buffer     | 99H       |  |                  |     |     |      |                  |      |     | xxxxxxxxB               |
| SCON*   | Serial controller      | 98H       | SM0  | SM1              | SM2 | REN | TB8  | RB8              | T1   | R1  | 00H                     |
| SP      | Stack pointer          | 81H       |  |                  |     |     |      |                  |      |     | 07H                     |
| TCON*   | Timer control          | 88H       | TF1  | TR1              | TF0 | TR0 | IE1  | IT1              | IE0  | IT0 | 00H                     |
| TH0     | Timer high 0           | 8CH       |  |                  |     |     |      |                  |      |     | 00H                     |
| TH1     | Timer high 1           | 8DH       |  |                  |     |     |      |                  |      |     | 00H                     |
| TL0     | Timer low 0            | 8AH       |  |                  |     |     |      |                  |      |     | 00H                     |
| TL1     | Timer low 1            | 8BH       |  |                  |     |     |      |                  |      |     | 00H                     |
| TMOD    | Timer mode             | 89H       | GATE   | $\overline{C/T}$ | M1  | M0  | GATE | $\overline{C/T}$ | M1   | M0  | 00H                     |

Figura 2-16 Características de los SFR

Como se había señalado en el mapa de memoria de datos (figura 2-14) algunos de los SFR son direccionables bit a bit, en la figura 2-16, éstos están marcados con “\*”.

A continuación se describen mas ampliamente los SFR que son utilizados en este trabajo:

- **ACC (Acumulador, Dirección E0H)**

Es el registro más utilizado, y sobre el que recaen la mayoría de las operaciones, al estar conexionado con la Unidad Lógico Aritmética (ALU). Es direccionable a nivel Bit. Interviene en todas las instrucciones que llevan la letra A (Por ejemplo: ADD A,#34H). Cuando una instrucción admite una dirección puede utilizarse el símbolo ACC, que equivale al valor 0E0H, o dirección del registro A.

- **B (B Register, Dirección 0F0H, Direccionable por bit)**

Es un registro de propósito general, que colabora con el Acumulador en las instrucciones de multiplicación y división (MUL AB y DIV AB).

- **DPL/DPH (Data Pointer Low/High, Direcciones 82H y 83H)**

El registro de 16 bits DPTR, está formado por la unión de los registros DPH y DPL, cuya misión principal consiste en apuntar a cualquier dirección dentro de los 64 Kbytes del espacio de memoria tanto de código como de datos. Para ello se dispone de instrucciones específicas.

### Ejemplo:

|                |  |
|----------------|--|
| CLR A          | Borra el Acumulador.   |
| MOV DPTR,#1234 | Carga el DPTR con el valor 1234H.  |
| MOVC A,@A+DPTR | El contenido de la dirección 1234H de memoria de código, se copia en el Acc. |
| MOVX @DPTR,A   | El contenido del Acc se copia en la dirección 1234H de la memoria de datos.  |

- **IE (Interrupt Enable, Dirección 0A8H)**

Es el registro de habilitación de interrupción. Cada interrupción puede ser habilitada individualmente activando el bit que tiene asignado en este SFR. A continuación se detalla cada bit del registro.

|     |              | MSB   |   |   |    |     |     |     | LSB |
|-----|--------------|---|---|---|----|-----|-----|-----|-----|
|     |              | EA  | - | - | ES | ET1 | EX1 | ET0 | EX0 |
| EA  | IE.7         | Habilitación global de las interrupciones. Si EA = 0 no se atenderá ninguna interrupción. |   |   |    |     |     |     |     |
| ES  | IE.4         | Habilita o deshabilita la interrupción del puerto serie.                                  |   |   |    |     |     |     |     |
| EXx | IE.2<br>IE.0 | Habilita o deshabilita la interrupción externa INTx.                                      |   |   |    |     |     |     |     |
| ETx | IE.3<br>IE.1 | Habilita o deshabilita la interrupción del Timer Tx.                                      |   |   |    |     |     |     |     |

- **IP (Interrupt Priority, Dirección 0B8H)**

Cada fuente de interrupción dispone de 2 niveles de prioridad. El nivel de prioridad de una interrupción se establece mediante el bit asignado a la misma en este registro. Una interrupción solo se verá interrumpida por otra de un nivel superior de prioridad.

| MSB |              |   |    | LSB |     |     |     |
|-----|--------------|---|----|-----|-----|-----|-----|
| -   | -            | -   | PS | PT1 | PX1 | PT0 | PX0 |
| PS  | IP.4         | Prioridad de la interrupción del puerto serie. (1 = ALTA, 0 = BAJA) |    |     |     |     |     |
| PTx | IP.3<br>IP.1 | Prioridad de la interrupción del Timer Tx.                          |    |     |     |     |     |
| PXx | IP.2<br>IP.0 | Prioridad de la interrupción externa INTx.                          |    |     |     |     |     |

- **P0, P1, P2, P3 (Direcciones 80H, 90H, A0H y B0H)**

Existen cuatro registros de 8 bits encargados de gestionar los puertos paralelos en el microcontrolador 8051 que son, P0 (Dirección 80H), P1 (Dirección 90H), P2 (Dirección 0A0H) y P3 (Dirección 0B0H). Estos puertos confieren 32 pines al microcontrolador. Cada puerto consiste en un latch, un driver de salida, y un buffer de entrada.

Los pines de los puertos además de servir como líneas de entrada y salida de propósito general, admiten las siguientes funciones alternativas:

1. Los drivers de salida de los puertos P0 y P2, y el buffer de entrada del puerto P0 son usados en el acceso a la memoria externa.
2. En cuanto al puerto P3, sus hilos proporcionan diferentes funciones alternativas al microcontrolador 8051.

- **PSW (Program Status Word, Dirección D0H)**

Contiene información del estado de la CPU que se actualiza tras la ejecución de cada instrucción. En la siguiente tabla se explica detenidamente:

|            |                | MSB   |            |                                       |     | LSB |    |   |   |
|------------|----------------|---|------------|---------------------------------------|-----|-----|----|---|---|
|            |                | CY  | AC         | FO                                    | RS1 | RS0 | OV | - | P |
| CY         | PWS.7          | Flag de acarreo. Se pone a 1 si hay llevada en sumas o restas.  |            |                                       |     |     |    |   |   |
| AC         | PWS.6          | Flag de acarreo intermedio. Se pone a 1 en las sumas y restas si hay llevada desde el nibble bajo hacia el nibble alto.             |            |                                       |     |     |    |   |   |
| FO         | PWS.5          | Flag de propósito general a disposición del programador.  |            |                                       |     |     |    |   |   |
| RS1<br>RS0 | PWS.4<br>PWS.3 | <b>RS1</b>  | <b>RS0</b> | <b>Selección de banco de registro</b> |     |     |    |   |   |
|            |                | 0   | 0          | Banco 0 Direcciones: 00H – 07H        |     |     |    |   |   |
|            |                | 0   | 1          | Banco 1 Direcciones: 08H – 0FH        |     |     |    |   |   |
|            |                | 1   | 0          | Banco 2 Direcciones: 10H – 17H        |     |     |    |   |   |
|            |                | 1   | 1          | Banco 3 Direcciones: 18H – 1FH        |     |     |    |   |   |
| OV         | PWS.2          | Flag de desbordamiento. Cuando se utiliza aritmética entera en complemento a dos, OV = 1 indica que el resultado no cabe en 8 bits. |            |                                       |     |     |    |   |   |
| -          | PWS.1          | Indefinida  |            |                                       |     |     |    |   |   |
| P          | PWS.0          | Flag de paridad. (P = 1 si el acumulador tiene número impar de unos. P = 0 si el acumulador tiene un número par de unos.)           |            |                                       |     |     |    |   |   |

- **SP (Stack Pointer, Dirección 81H)**

Durante una instrucción CALL o PUSH, el contenido del registro SP se incrementa antes de almacenar los datos en la dirección apuntada por SP.

Durante una instrucción POP, RET, o RETI, el contenido del registro SP se decrementa después de recuperar los datos desde la dirección apuntada por SP (Salvo en "POP SP", en el cual SP se decrementa antes de ser cargado con el valor sacado de la pila).



Después de un Reset, el registro SP queda cargado con el valor 07H. Su misión es apuntar a la RAM interna donde serán guardadas las direcciones de retorno de las subrutinas llamadas con instrucciones CALL, y desde donde dichas direcciones serán leídas al encontrarse con instrucciones RET de retorno de subrutina. Asimismo, las instrucciones POP permiten recoger de la PILA los valores de las variables depositados anteriormente con instrucciones PUSH.

- **TCON (Timer Control Register, Dirección 88H)**

Configura y modifica las funciones de los Timers T0 y T1.

En la siguiente tabla se utiliza el termino "x" haciendo referencia a la utilización tanto del Timer0 (T0) como del Timer1 (T1) indistintamente.

|     |                  | MSB   |     |     |     |     |     |     | LSB |
|-----|------------------|---|-----|-----|-----|-----|-----|-----|-----|
|     |                  | TF1   | TR1 | TF0 | TR0 | IE1 | IT1 | IE0 | IT0 |
| TFx | TCON.7<br>TCON.5 | Flag de desbordamiento del timer Tx, se borra automáticamente una vez atendida la interrupción.   |     |     |     |     |     |     |     |
| TRx | TCON.6<br>TCON.4 | Bit de arranque Tx<br>= "1" si permite que el timer Tx cuente<br>= "0" no permite que el timer Tx cuente                                  |     |     |     |     |     |     |     |
| IEx | TCON.3<br>TCON.1 | Bit de transición de INTx<br>Flag de petición de la interrupción externa INTx, se borra automáticamente una vez atendida la interrupción. |     |     |     |     |     |     |     |
| ITx | TCON.2<br>TCON.0 | Modo de control de INTx<br>= "1" Interrupción INTx por flanco de Bajada<br>= "0" por nivel bajo.  |     |     |     |     |     |     |     |

- **TLx/THx (Timer x Low/High, Direcciones 8AH/8BH y 8CH/8DH)**

Un timer puede pedir una interrupción con carácter periódico (modo temporizador), o después de recibir un determinado número de flancos de bajada en una patilla (modo contador). Los dos modos incrementan los registros TLx y THx que piden una interrupción en el momento en que se desbordan.

- **TMOD (Timer Mode Control Register, Dirección 89H)**

Configura el modo de trabajo de los timers entre los cuatro posibles de la siguiente tabla.

|      |                            | MSB   |                                  |    |                                    | LSB     |     |    |    |      |                  |    |           |   |   |   |                  |   |   |   |                  |   |   |   |                                  |    |                            |   |   |   |                                    |
|------|----------------------------|---|----------------------------------|----|------------------------------------|---------|-----|----|----|------|------------------|----|-----------|---|---|---|------------------|---|---|---|------------------|---|---|---|----------------------------------|----|----------------------------|---|---|---|------------------------------------|
|      |                            | GATE  | C/T                              | M1 | M0                                 | GATE    | C/T | M1 | M0 |      |                  |    |           |   |   |   |                  |   |   |   |                  |   |   |   |                                  |    |                            |   |   |   |                                    |
|      |                            | TIMER 1   |                                  |    |                                    | TIMER 0 |     |    |    |      |                  |    |           |   |   |   |                  |   |   |   |                  |   |   |   |                                  |    |                            |   |   |   |                                    |
| GATE | TMOD.7<br>TMOD.3           | = "1" Permite que el pin INTx actúe sobre la señal de CONTROL del Temporizador. (El bit TRx debe estar a "1").  |                                  |    |                                    |         |     |    |    |      |                  |    |           |   |   |   |                  |   |   |   |                  |   |   |   |                                  |    |                            |   |   |   |                                    |
| C/T  | TMOD.6<br>TMOD.2           | Selecciona modo Contador (1) o modo Temporizador (0).   |                                  |    |                                    |         |     |    |    |      |                  |    |           |   |   |   |                  |   |   |   |                  |   |   |   |                                  |    |                            |   |   |   |                                    |
| M1   | TMOD.5                     | <table border="1"> <thead> <tr> <th>MODO</th> <th>M1</th> <th>M0</th> <th>OPERACIÓN</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>Timer de 13 bits</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>Timer de 16 bits</td> </tr> <tr> <td>2</td> <td>1</td> <td>0</td> <td>Timer de 8 bits con Auto-Recarga</td> </tr> <tr> <td>M0</td> <td>TMOD.4<br/>TMOD.1<br/>TMOD.0</td> <td>3</td> <td>1</td> <td>1</td> <td>T0 desdoblado, T1 parado en modo 3</td> </tr> </tbody> </table> |                                  |    |                                    |         |     |    |    | MODO | M1               | M0 | OPERACIÓN | 0 | 0 | 0 | Timer de 13 bits | 1 | 0 | 1 | Timer de 16 bits | 2 | 1 | 0 | Timer de 8 bits con Auto-Recarga | M0 | TMOD.4<br>TMOD.1<br>TMOD.0 | 3 | 1 | 1 | T0 desdoblado, T1 parado en modo 3 |
| MODO | M1                         |   |                                  |    |                                    |         |     |    |    | M0   | OPERACIÓN        |    |           |   |   |   |                  |   |   |   |                  |   |   |   |                                  |    |                            |   |   |   |                                    |
| 0    | 0                          |   |                                  |    |                                    |         |     |    |    | 0    | Timer de 13 bits |    |           |   |   |   |                  |   |   |   |                  |   |   |   |                                  |    |                            |   |   |   |                                    |
| 1    | 0                          |   |                                  |    |                                    |         |     |    |    | 1    | Timer de 16 bits |    |           |   |   |   |                  |   |   |   |                  |   |   |   |                                  |    |                            |   |   |   |                                    |
| 2    | 1                          | 0   | Timer de 8 bits con Auto-Recarga |    |                                    |         |     |    |    |      |                  |    |           |   |   |   |                  |   |   |   |                  |   |   |   |                                  |    |                            |   |   |   |                                    |
| M0   | TMOD.4<br>TMOD.1<br>TMOD.0 | 3   | 1                                | 1  | T0 desdoblado, T1 parado en modo 3 |         |     |    |    |      |                  |    |           |   |   |   |                  |   |   |   |                  |   |   |   |                                  |    |                            |   |   |   |                                    |
|      | TMOD.4                     |   |                                  |    |                                    |         |     |    |    |      |                  |    |           |   |   |   |                  |   |   |   |                  |   |   |   |                                  |    |                            |   |   |   |                                    |
|      | TMOD.1                     |   |                                  |    |                                    |         |     |    |    |      |                  |    |           |   |   |   |                  |   |   |   |                  |   |   |   |                                  |    |                            |   |   |   |                                    |
|      | TMOD.0                     |   |                                  |    |                                    |         |     |    |    |      |                  |    |           |   |   |   |                  |   |   |   |                  |   |   |   |                                  |    |                            |   |   |   |                                    |

## 2.5 Dispositivos Lógicos Programables

Para que el microcontrolador pueda habilitar los dispositivos externos, necesarios para la creación del intérprete gráfico, necesita de un circuito decodificador, que para este caso se propone utilizar un

GAL16V8 que es una matriz genérica programable que forma parte de los Dispositivos Lógicos Programables.

Un dispositivo lógico programable, o PLD (Programmable Logic Device), es un dispositivo cuyas características pueden ser modificadas y almacenadas mediante programación. El dispositivo programable más simple es el PAL (Programmable Array Logic). El circuito interno de un PAL consiste en una matriz de conexiones, una matriz de compuertas AND y un arreglo de compuertas OR.

Una matriz de conexiones es una red de conductores distribuidos en filas y columnas con un fusible en cada punto de intersección, mediante la cual se seleccionan cuales entradas del dispositivo serán conectadas al arreglo AND cuyas salidas son conectadas al arreglo OR y de esta manera obtener una función lógica en forma de suma de productos.

La mayoría de los PLDs están formados por una matriz de conexiones, una matriz de compuertas AND, y una matriz de compuertas OR y algunos, además, con registros. Las matrices pueden ser fijas o programables. Con estos recursos se implementan las funciones lógicas deseadas mediante un software especial y un programador de dispositivos.

### **2.5.1 Matriz Genérica Programable**

Una Matriz Genérica Programable, GAL (Generic Array Logic), en su forma básica es un PLD con una matriz AND reprogramable, una matriz OR fija y una matriz lógica de salida programable mediante una macro celda.

Esta estructura permite implementar cualquier función lógica en forma de suma de productos con un número de términos definido.

En los PLDs no reprogramables la síntesis de las ecuaciones lógicas se realiza mediante la quema de fusibles en cada punto de intersección de los pines de entrada con las compuertas. En un GAL el fusible se reemplaza por una celda CMOS eléctricamente borrable (EECMOS) y mediante programación se activa o desactiva cada celda EECMOS.

Una celda activada conecta su correspondiente intersección de fila y columna, y una celda desactivada desconecta dicha intersección. Con esta estructura se puede aplicar cualquier combinación de variables de entrada, o sus complementos, a una compuerta AND para generar cualquier operación producto que se desee.

Para la realización de este trabajo se propone utilizar un GAL16V8 como decodificador para las habilitaciones externas al microcontrolador.

## **2.6 Memoria**

Las memorias son dispositivos semiconductores que sirven para guardar información: datos, comandos de programas o programas enteros.

En una memoria existen líneas de datos, líneas de direccionamiento y líneas de control, además de las de alimentación.

Mediante las líneas de direccionamiento el microprocesador selecciona una posición de memoria donde leer o escribir un dato.

Con las líneas de datos (o bus de datos) el microprocesador recibe o envía información desde o hacia la memoria, cuando lee o escribe.

Con las líneas de control, el microcontrolador selecciona, habilita e indica si se leerá o se escribirá en determinada memoria.

La siguiente figura presenta una tabla con las características de las memorias.

| Tipo de memoria                             | Clase              | Borrado                       | Grabado                          | ¿Volátil? |
|---|--------------------|-------------------------------|----------------------------------|-----------|
| RAM<br>(de acceso aleatorio)                | lec / esc          | eléctricamente<br>(por bytes) | eléctricamente                   | si        |
| ROM<br>(de solo lectura)                    | lec                | no es posible                 | con una máscara                  | no        |
| PROM<br>(ROM programable)                   | lec                | no es posible                 | eléctricamente<br>(una sola vez) | no        |
| EPROM<br>(PROM borrable)                    | lec                | luz ultravioleta              | eléctricamente                   | no        |
| EEPROM<br>(PROM eléctricamente<br>borrable) | lec<br>esc (lenta) | eléctricamente<br>(por bytes) | eléctricamente                   | no        |

**Figura 2-17 Características de las memorias semiconductoras**

### 2.6.1 Memoria RAM

La memoria RAM (por sus siglas en inglés Random Access Memory) es muy importante en el diseño de una computadora, con este tipo de memoria se implementa la memoria caché y la principal. Lo que distingue a las memorias RAM de las ROM, no es el acceso aleatorio

(como su nombre lo indica, ya que ambas manejan esa forma de acceso) sino el poder escribir o no en ellas de forma eléctrica y la volatilidad (retener la información cuando deja de estar alimentada).

Este tipo de memorias se clasifican en 2 grandes grupos: SRAM y DRAM.

### **Memorias SRAM**

La memoria SRAM o RAM estática (Static RAM) se conocen con este nombre ya que es capaz de mantener sus estados mientras esté alimentada. Está constituida por unidades básicas de memoria llamadas celdas (flip-flops). Otra característica de estas memorias es que las líneas de direcciones seleccionan directamente un byte o palabra. Debido a esto, este tipo de memorias se utilizan para implementar memorias caché.

### **Memorias DRAM**

En una memoria DRAM o RAM dinámica (Dinamic RAM), la información es guardada en forma de carga en un capacitor. Una DRAM es capaz de guardar información por tan solo unos milisegundos, por este motivo es necesario "refrescar" constantemente la carga del capacitor a su valor total y esto a su vez provoca que los procesos de lectura y/o escritura sean mucho más lentos que en una SRAM.

## **2.6.2 Memoria ROM**

El otro gran grupo de tipos de memoria es el de sólo lectura ROM (por sus siglas en inglés Read Only Memory). Esto significa que el microprocesador puede acceder a la información grabada, pero no modificarla.

Son conocidas como memorias no volátiles ya que conservan su contenido aunque se desactive la alimentación.

Como podemos ver en la figura 2-17, existen subgrupos de ROM que, partiendo de su forma de grabación, borrado y regrabación, se diferencian entre sí aunque conserven la característica principal que es la de ser de sólo lectura.

### **Memorias ROM y PROM**

Las memorias ROM no modificables son la ROM y la PROM. Este tipo de memorias, una vez grabadas, no se pueden reprogramar.

Las ROM (Read only Memory) son memorias que salen grabadas de fábrica con una programación especificada por el cliente.

Las PROM (Programmable ROM) son memorias ROM vírgenes que pueden ser programadas una sola vez por el adquisidor para alguna aplicación específica.

### **Memorias EPROM**

Otro tipo de memorias de solo lectura son las reprogramables, entre ellas existen diferencias en su modo de borrado y reprogramación.

Las PROM borrables, EPROM (Eraseable PROM) se distinguen físicamente por tener una ventana directamente sobre la matriz del chip. Para modificar el programa en el chip, primero debe exponerse a una radiación de luz ultravioleta, que borra el contenido almacenado, y después grabar eléctricamente el nuevo contenido utilizando un programador.

### **Memorias EEPROM**

Las PROM eléctricamente borrables, EEPROM ó E<sup>2</sup>PROM (Electrically Eraseable PROM), cuya ventaja es la de ser regrabables durante un procesamiento de datos, sin sacarlas de su placa para exponerlas a una radiación de luz ultravioleta. Se siguen considerando de solo lectura aunque pueden ser usadas como de escritura "lenta" (en mseg.) para el almacenamiento prolongado de nuevos datos.

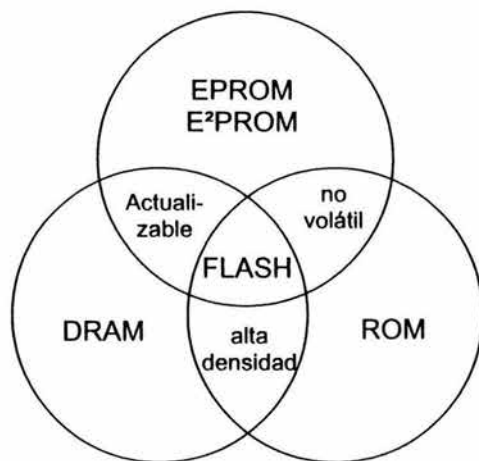
### **2.6.3 Memoria FLASH**

Las memorias FLASH se derivan de las EEPROM, los chips Flash son menos costosos y de mayor densidad (capacidad de almacenamiento). Son no volátiles ya que mantienen su contenido aunque no estén alimentadas.

El término Flash se refiere a su capacidad para borrarse "en un instante". Deben ser borradas en bloques de tamaños fijos, por lo general van de 512 bytes hasta 256KB.

Las memorias Flash se están convirtiendo en una alternativa que sustituye a otros tipos de memoria. La figura 2-18 muestra los tipos de memoria y las características que comparten.





**Figura 2-18 Tipos de memoria**

La siguiente figura muestra, en resumen, las características de los principales tipos de memoria.

| TIPO DE MEMORIA  | CARACTERÍSTICAS   |
|--|---|
| <b>FLASH</b>   | Bajo-costo, bajo-consumo, alta-densidad, arquitectura de alta-velocidad, alta-confiabilidad       |
| <b>ROM</b><br>Read-Only Memory                                       | Bajo-costo, alta-densidad, confiable, conveniente para alta producción con código estable         |
| <b>SRAM</b><br>Static Random-Access Memory                           | Alta-velocidad, alto-consumo, memoria de baja-densidad; la densidad limitada maneja altos costos. |
| <b>EPROM</b><br>Electrically Programmable Read-Only Memory           | Memoria de alta densidad; debe ser expuesta a luz ultravioleta para ser borrada.                  |
| <b>EEPROM</b><br>Electrically Erasable Programmable Read-Only Memory | Alto-costo, la más baja-densidad, baja confiabilidad; Byte borrable eléctricamente.               |
| <b>DRAM</b><br>Dynamic Random Access Memory                          | Bajo-costo, alto-consumo, alta-densidad, alta-velocidad.  |

Figura 2-19 Características de los principales tipos de memoria.

## 2.7 Características de la voz

La emisión de la voz se mide mediante tres parámetros básicos:

1. Intensidad: Medida en decibelios. Determina las voces fuertes o débiles.
2. Tono o frecuencia de la voz: Medida en ciclos por segundo o hertz. Determina las voces graves o agudas.

3. Timbre: Son los armónicos. Determina las características propias de la voz. Así tenemos voces metálicas, voces apagadas, etc. Determinada, en parte, por las cavidades de resonancia.

## **2.8 ¿Qué es muestrear?**

Dicho de una forma muy simple, muestrear es hacer fotos de un sonido a mucha velocidad. La comparación más fácil es el cine: el ojo humano necesita unas 24 imágenes o "fotos" por segundo para crear la ilusión de movimiento continuo. El oído es bastante más fino y necesita unas 40.000 tomas de audio por segundo para percibir un sonido grabado como real.

Por motivos prácticos, los grabadores digitales hablan en lenguaje binario. La unidad básica con la que trabaja el sistema binario es el bit, que es la unidad de información más pequeña que se puede manejar (o cero o uno). Del mismo modo, con cuatro bits puedo representar 16 valores (2 elevado a 4). Pues bien, la precisión más común a utilizar en los sistemas de grabación digital es de 16 bits, que representa un total de 65.536 valores para cada muestra. A esta precisión en la medida se le llama resolución de muestreo, y junto con la frecuencia de muestreo constituyen los valores más importantes a tratar cuando se graba audio digitalmente.

### 2.8.1 Teorema del muestreo

Harry Nyquist dedujo el teorema sobre el que se sustenta toda grabación digital de audio. El teorema tiene este enunciado "La frecuencia de muestreo para convertir audio analógico debe ser al menos el doble de la frecuencia máxima a convertir si no queremos que se pierda información esencial o que aparezcan falsas frecuencias denominadas alias". En audio digital se denomina "alias" a una frecuencia "fantasma" que se produce en los grabadores si la frecuencia del sonido a grabar (o algún componente armónico del sonido) es mayor que la mitad de la frecuencia de muestreo (frecuencia de Nyquist).

La razón de esto es simple (por lo menos en teoría). Se necesita un mínimo de dos muestras por ciclo para representar los movimientos positivo y negativo de una forma de onda. Si la señal no se muestrea al menos 2 veces en cada ciclo de la forma de onda, la traducción digital parecerá representar una onda de menor frecuencia.

El "aliasing" ocurre cuando el circuito de muestreo y retención (sample & hold) contiene una muestra de un ciclo de la forma de onda, pero en el momento de captar la segunda muestra, el ciclo ya ha pasado y ha empezado uno nuevo. La segunda muestra entonces, se ha tomado en un ciclo completamente diferente al primero. En lugar de representar correctamente la frecuencia de la forma de onda, una frecuencia "alias" es lo que se obtiene.

La solución al aliasing es simple: filtrar las frecuencias por encima de la frecuencia de Nyquist antes de ser muestreadas. Por este motivo todos los grabadores digitales tienen un filtro pasa bajos (algunas veces llamados anti-aliasing) con una envolvente de corte precisa en el primer lugar de la cadena de grabación. La frecuencia de corte del filtro se ajusta un poco por debajo de la frecuencia de Nyquist para producir un espacio de seguridad.

## **CAPÍTULO 3**

---

---

### **DESARROLLO**

### 3.1 Diseño de un Intérprete Gráfico

En este trabajo se diseña un intérprete gráfico que ayude a las personas con dificultades en el habla a comunicarse. Reproduciendo el audio que exprese el objeto, la frase y/o la acción de una imagen (que podrá variar según la actividad que se realice).

Es el diseño de un sistema dedicado, que solo podrá realizar la tarea de intérprete gráfico y deberá ser utilizado en la forma que se indique.

Como ya se ha mencionado, es un diseño que interconecta dispositivos comercialmente disponibles, como el MCS-51, microcontrolador que "llevará el control" ejecutando instrucciones.

#### 3.1.1 Características del intérprete gráfico

El siguiente diagrama representa, de una forma general, los bloques que son utilizados en el sistema.

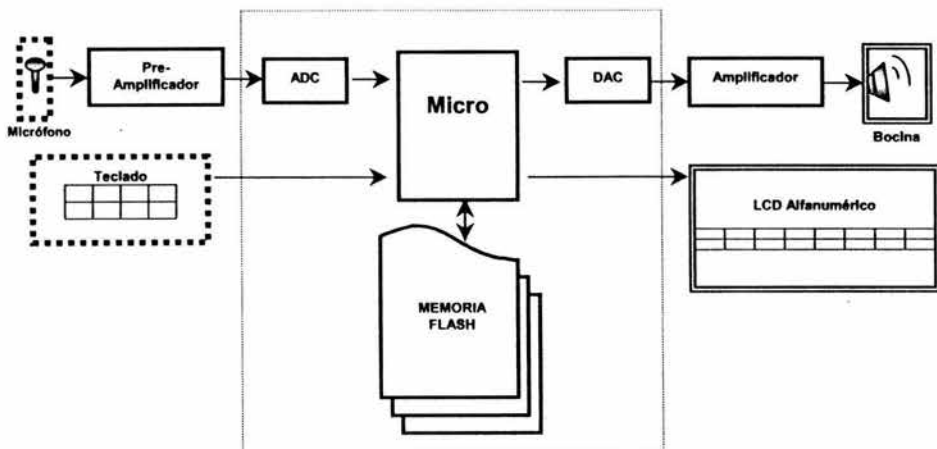
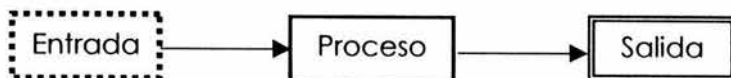


Figura 3-1 Diagrama a bloques del sistema

Donde el código de líneas representa:



Y las flechas representan el flujo de datos.

### 3.1.2 Elementos de Entrada

Como se muestra en la figura 3-1, existen 2 elementos de entrada, un micrófono y un teclado.

Puede utilizarse cualquier micrófono unidireccional de baja impedancia

El teclado consta de 8 botones "un polo dos tiros". Para este trabajo se construyó en un tablero como se muestra en la siguiente imagen.

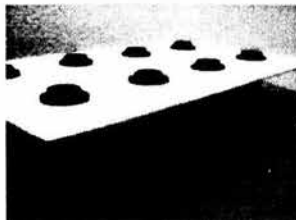


Figura 3-2 Teclado de 8 botones

### 3.1.3 Elementos de Salida

Los 2 elementos de salida que se muestran en la figura 3-1, son la bocina y el display alfanumérico.

Una bocina de computadora sirvió para realizar las pruebas básicas del sistema.



El display alfanumérico AND491, es una pantalla de cristal liquido de 2 renglones y 16 columnas, tiene la función de facilitar la comunicación con el usuario ya que en ella se visualizará el número de nivel, mensaje y el modo en que se encuentra: grabando o reproduciendo.

### **3.1.4 Elementos del Proceso**

Finalmente los elementos del proceso, como se puede ver en figura 3-1, forman la mayor parte del sistema. En este trabajo, todos los bloques que conforman el proceso, con excepción del pre-amplificador y el amplificador, forman parte de lo que llamaremos "módulo principal" debido a que se encuentran en la misma placa (línea punteada en la figura 3-1). El pre-amplificador y el amplificador serán tomados como módulos independientes ya que cada uno de ellos tiene su placa independiente, ambos son kits PROAM y fueron armados en base a los diagramas eléctricos que se muestran en el Apéndices A y B.

El módulo principal contiene los siguientes circuitos:

- Un microcontrolador 8751
- Una memoria FLASH Am29F040
- Un ADC0808, convertidor analógico-digital
- Un DAC0800, convertidor digital-analógico
- Un TL081, amplificador operacional configurado como seguidor de voltaje después del DAC
- Dos circuitos 74LS573, uno captura las líneas de direcciones parte baja A0-A7, y el otro captura los datos para el DAC0800
- Un 74C922, controlador de teclado de 16 teclas

- Un GAL16V8 como decodificador para la habilitación de dispositivos externos al microcontrolador
- Display Alfanumérico AND491 (16 Columnas x 2 renglones)

La figura 3-3 representa, en forma de bloques, la relación que existe entre los circuitos que se utilizarán en todo el sistema. Algunos de estos circuitos trabajan de una manera síncrona, esto significa que requieren de un pulso de reloj para su correcto funcionamiento, estos pulsos se generan utilizando cristales de cuarzo y algunos circuitos que no se especifican en el diagrama pero que serán mencionados en su momento.

Como se puede ver en la figura 3-3, todo el sistema estará alimentado por una fuente de alimentación de +5, +15, -15 y 24 Volts, debido a los requerimientos de cada dispositivo. El diagrama eléctrico y la lista de partes están representados en el Apéndice C.

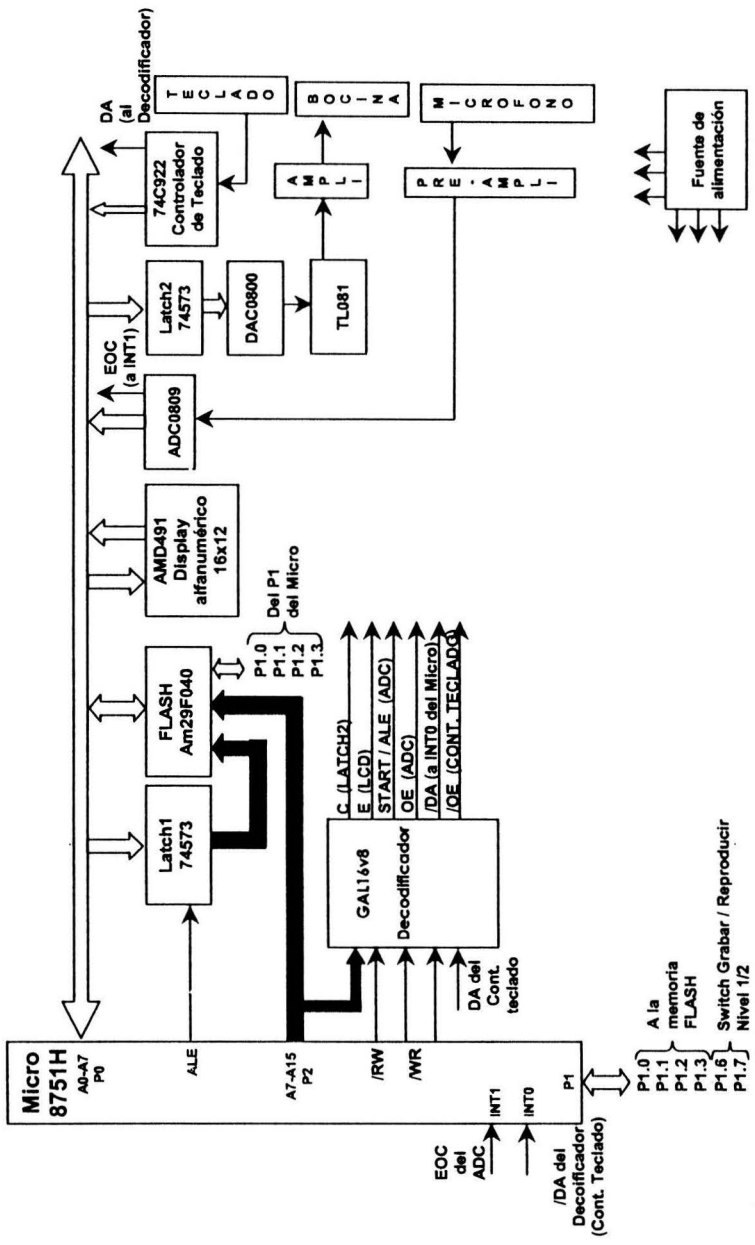


Figura 3-3 Diagrama de Interconexión del sistema

### 3.1.4.1 Microcontrolador (8751H)

En el capítulo II hay una amplia explicación de las áreas de almacenamiento, registros y modos de direccionamiento del microcontrolador MSC-51 de INTEL.

A continuación se mencionan sus características principales y una breve descripción de cada uno de sus pines o patas:

Características principales:

- Proceso de Alto Rendimiento HMOS (high-density, short-channel MOS)<sup>1</sup>
- Contadores internos de Tiempos / Eventos
- 32 Líneas de Entrada / Salida (Cuatro puertos de 8-Bits)
- 4K bytes de Espacio de Memoria de Programa Interna EPROM
- 128 bytes de Espacio de Memoria de Datos Interna
- Procesador Booleano
- CPU de 8 bits
- Estructura de Prioridad de Interrupción de 2-Niveles
- 5 Fuentes de interrupción:
  - 2 interrupciones externas
  - 2 interrupciones de los contadores
  - 1 interrupción de la comunicación serie

---

<sup>1</sup> HMOS (alta densidad, MOS del canal-corto), que fue desarrollada originalmente por Intel Corporation. El proceso de HMOS proporciona densidades del circuito del doble de los NMOS estándar, y un producto en velocidad y consumo cuatro veces mejor que los NMOS estándar.

Breve descripción de los pines del microcontrolador 8751H:

VCC (40): Alimentación positiva de 5 voltios.

VSS (20): Conexión a tierra o referencia cero voltios.

PUERTOS: Todos los microcontroladores poseen puertos bidireccionales (pueden programarse como entrada o salida), que son los que permiten la comunicación con el medio exterior. El MCS-8751 Intel, posee 4 puertos de 8 bits capacitados para realizar las siguientes funciones:

Puerto 0 (P0): Es un puerto bidireccional con salidas en colector abierto. Cuando el puerto tiene 1's escritos, las salidas flotan y pueden servir como entradas. El puerto 0 es también multiplexado para obtener el DATO y la parte baja de la dirección.

Puerto 1 (P1): Es un puerto quasibidireccional<sup>2</sup>, cuando se escribe 1's en el puerto, el puerto puede ser utilizado como entrada.

Puerto 2 (P2): Es un puerto quasibidireccional con fijadores de nivel internos (PULL-UP). Cuando se escriben 1's sobre el puerto, las líneas pueden ser utilizadas como entradas o como salidas. El puerto 2 es utilizado además para direccionar memoria externa. Este puerto, emite el byte más alto de la dirección durante la búsqueda de datos en la memoria del programa externo y durante el acceso a memorias de datos externos que usan direccionamientos de 16 bits.

---

<sup>2</sup> Un puerto de Entrada/Salida quasibidireccional puede o no tener pull-ups internos, no es igual leer del latch que leer del pin del puerto y en él se realizan operaciones de lectura-escritura y modificación como: ANL, ORL, XRL, CPL, INC y DEC.  
[www.cs.itu.edu.tr](http://www.cs.itu.edu.tr)

Puerto 3 (P3): Es un puerto quasibidireccional con fijadores de nivel internos (PULL-UP). Cuando se escriben 1's sobre el puerto, las líneas pueden ser utilizadas como entradas o como salidas.

El puerto 3 es utilizado además para producir señales de control de dispositivos externos como son las siguientes:

P3.0 RXD Entrada del puerto serial de comunicaciones

P3.1 TXD Salida del puerto serial de comunicaciones

P3.2 INT0 Interrupción externa 0

P3.3 INT1 Interrupción interna 1

P3.4 T0 Entrada externa del temporizador 0

P3.5 T1 Entrada externa del temporizador 1

P3.6 /WR Autorización de escritura en memoria de datos externa

P3.6 /RD Autorización de lectura en memoria de datos externa

Como salida los drivers del puerto 0 pueden soportar una cargabilidad (número de entradas aplicadas a sus pines de salida) de 8 cargas TTL-LS, el resto de puertos solo admiten 4 cargas TTL-LS. Además el puerto 0 puede adoptar el estado de alta impedancia (triestado), por la disposición de los transistores en estado de no-conducción, mientras que en los otros tres puertos, el estado de reposo es en valor alto (1 lógico) para la salida, a través de la resistencia de carga (pull-up) integrada.

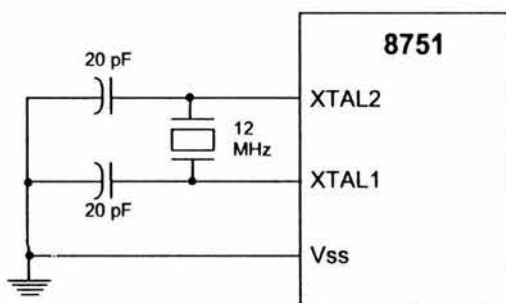
ALE/PROG (30): Este pin tiene doble función, cuando actúa como ALE (address Latch Enable) es un pulso que emite el microcontrolador, para enclavar el byte bajo del bus de direcciones en el acceso a la memoria

externa. ALE se emite con una frecuencia de 1/6 de la frecuencia de emisión del reloj. En cambio PROG es el pin de entrada de pulsos de programación de la memoria de programa interna.

/PSEN (29): (Program Store Enable) Es la señal de habilitación para leer en la memoria de programa externo. PSEN no se activa cuando se está ejecutando el programa de la "EPROM" interna.

/EA ( 31): (External Access) Cuando se mantiene en nivel alto, se ejecuta sólo el programa de la "ROM" interna.

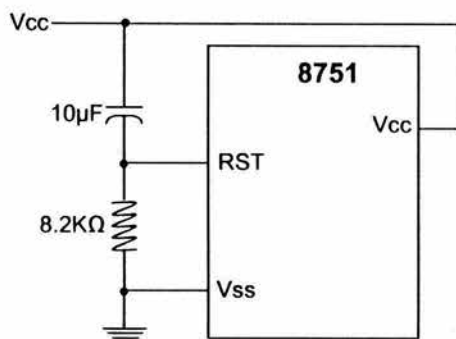
XTAL1 (19) y XTAL2 (18): Son la entrada y salida respectivamente, de un amplificador inversor que puede ser configurado para su uso como un "chip" oscilador. Se uso utilizando un cristal de cuarzo de 12 Mhz como se muestra en la siguiente figura:



**Figura 3-4** Circuito de reloj para el microcontrolador

RST (9): En este pin de entrada RESET se propicia la operación de inicialización de los dispositivos internos del microcontrolador. El

RESET automático del microcontrolador se produce si es conectado como se muestra en la siguiente figura:



**Figura 3-5 Circuito de RESET automático al encendido del módulo**

Los pines de un microcontrolador se clasifican en tres grupos de acuerdo a su función: dirección, datos y control. Los que están conectados a pines similares en las pastillas de memoria y de Entrada / Salida por medio de un BUS.

Para extraer una instrucción, el microprocesador pone primero la dirección de memoria de dicha instrucción en los pines de direccionamiento. Enseguida (por lo general), activa la línea de control para informar que desea leer una palabra. La memoria responde poniendo la palabra solicitada en los pines de datos.

No es posible realizar otro tipo de comunicación, entre la CPU y la memoria o los dispositivos de E/S, que esta, presentando y aceptando señales a través de sus pines.



### 3.1.4.2 Mapeo de Memoria y Entrada/Salida

El microcontrolador puede direccionar hasta 64kbytes de memoria de datos externa, es decir desde la dirección 0000h hasta la FFFFh.

Utilizando 16 líneas de direcciones A0 a A15 (correspondientes a los puertos 0 y 2) y 3 líneas de control ALE, /WR y /RD, el microcontrolador establece el control de los dispositivos externos de entrada y salida.

Las primeras 32K direcciones (0000h a 7FFFh) se destinan para manipular la memoria flash en 16 bloques (definidos por la combinación de 4 bits del puerto 1) de 32K bytes cada uno. Consecutivamente a los dispositivos periféricos como se muestra en la figura 3-6.

Con las direcciones de 0000h a 7FFFh se puede direccionar 32Kbytes y la memoria FLASH es de 512Kbytes, por esta razón se recurre a la paginación utilizando cuatro líneas del puerto 1, que pueden direccionar hasta 16 páginas ( $2^4 = 16$ ).

Cada una de estas páginas de 32Kbytes almacenarán los datos correspondientes a un mensaje de 3 segundos. Así podemos guardar los 16 mensajes (2 niveles de 8 imágenes).

|       |  |
|-------|--|
| FFFFh |  |
|       |  |
| C001h |  |
| C000h | Habilitación de Lectura del Teclado (/OE)                |
| BFFFh |  |
|       |  |
| B001h |  |
| B000h | Habilitación de Lectura de Datos del ADC (OE)            |
| AFFFh |  |
|       |  |
| A001h |  |
| A000h | Habilitación de Inicio de Conversión del ADC (Start/ALE) |
| 9FFFh |  |
|       |  |
| 9005h |  |
| 9004h | Habilitación de Lectura de Datos del LCD                 |
| 9003h | Habilitación de Lectura Bussy Flag del LCD               |
| 9002h | Habilitación de Escritura de Datos del LCD               |
| 9001h | Habilitación de Escritura de Instrucción del LCD         |
| 8FFFh |  |
|       |  |
| 8001h |  |
| 8000h | Habilitación del Latch2 que va al DAC (C)                |
| 7FFFh |  |
|       |  |
| 0000h |  |

**Figura 3-6 Mapa de memoria y dispositivos externos**

### 3.1.4.3 Habilitación de periféricos

Se utiliza una GAL16V8 como decodificador para la habilitación de dispositivos externos al microcontrolador.

La GAL16V8 es un PLD de alto desempeño, con tecnología CMOS y eléctricamente borrable E<sup>2</sup>CMOS. Se borra y reprograma a alta velocidad y tiene un bajo consumo de energía.

Tiene 1 pin para el clock que temporizara sus acciones (que en este trabajo no se utiliza) 8 pines de E/S y 8 mas de Entrada.

| ENTRADAS      |     | SALIDAS       |       |
|---------------|-----|---------------|-------|
| DESDE         | PIN | HACIA         | PIN   |
| Micro         | A15 | LATCH2        | C     |
| Micro         | A14 | (Pin 6) LCD   | E     |
| Micro         | A13 | ADC           | START |
| Micro         | A12 |               | ALE   |
| Micro         | WR  |               | OE    |
| Micro         | RD  | Micro         | /DA   |
| Micro         | ALE | Cont. Teclado | /OE   |
| Cont. Teclado | DA  |               |       |

Figura 3-7 Cuadro de E/S del GAL16V8

La figura 3-7 indica las señales que entran y las que salen del decodificador, así como también el nombre de cada una de ellas indicando de y hacia cual dispositivo esta conectado.

En la figura 3.8 se observa de una manera gráfica lo antes mencionado. Es importante mencionar que dentro de la flecha que representa el bus

de direcciones, que va hacia el decodificador, la RAM y la EEPROM, están consideradas las señales A15, A14, A13, A12, RD y WR.

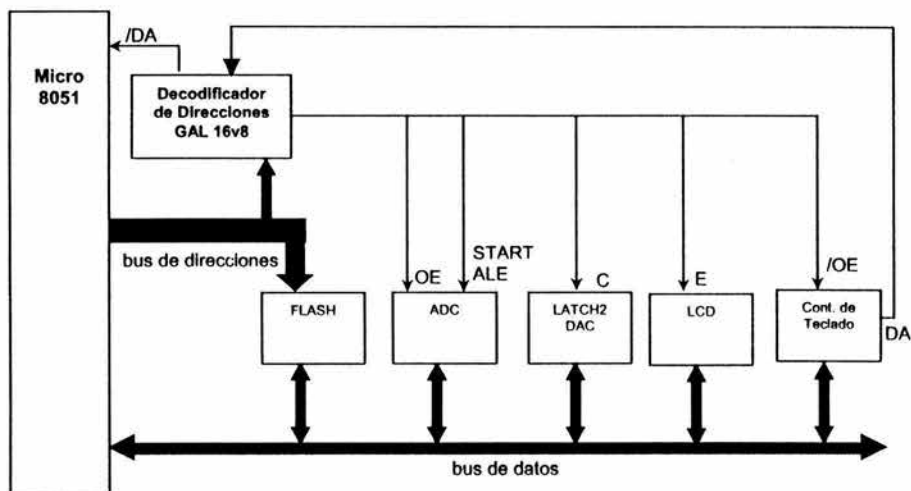


Figura 3-8 Diagrama de E/S del GAL16V8

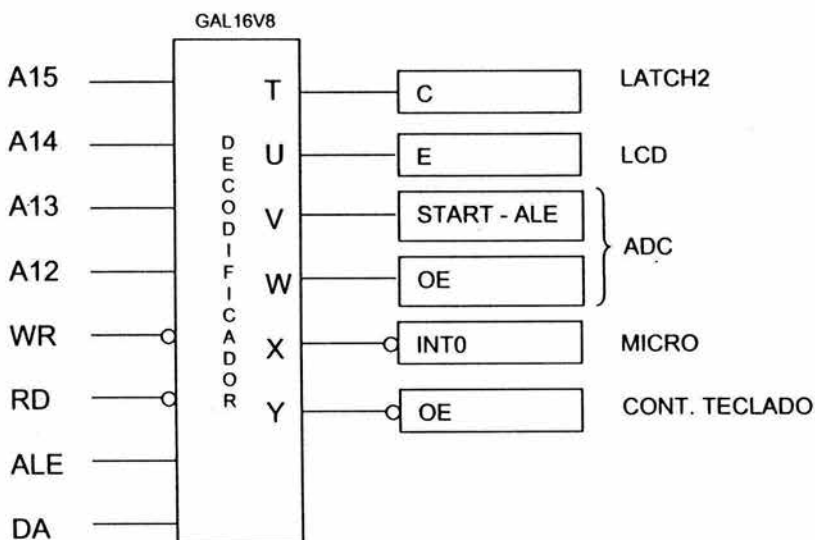


Figura 3-9 Ecuaciones de E/S del GAL16V8

Este dispositivo debe ser programado como decodificador en base a las siguientes ecuaciones que se basan en el circuito esquemático del Apéndice D.

$$T = A15 \cdot A14 \cdot /A13 \cdot A12 \cdot /WR \cdot /ALE$$

$$U = A15 \cdot A14 \cdot /A13 \cdot /A12 \cdot /WR \cdot /ALE +$$

$$A15 \cdot A14 \cdot /A13 \cdot /A12 \cdot /RD \cdot /ALE$$

$$V = A15 \cdot /A14 \cdot A13 \cdot A12 \cdot /WR \cdot /ALE$$

$$W = A15 \cdot /A14 \cdot A13 \cdot /A12 \cdot /RD \cdot /ALE$$

$$/X = DA$$

$$/Y = A15 \cdot /A14 \cdot /A13 \cdot A12 \cdot /RD \cdot /ALE$$

#### **3.1.4.4 Convertidor Analógico Digital (ADC0809)**

Debido a que los circuitos digitales trabajan con señales que pueden adoptar uno de dos valores posibles, alto o bajo (unos o ceros) y la información que queremos almacenar es una señal analógica (voz), existen los convertidores analógico-digital que como su nombre lo indica convierten la información analógica en digital.

Es importante mencionar que entre el micrófono y el convertidor analógico-digital, existe la etapa de preamplificación que amplifica la señal de la voz para que sean representativos los cambios de valores al momento de pasar al convertidor.

Las características principales de este convertidor son:

- Dispositivo CMOS monolítico
- Fácil interfaz con todos los microprocesadores
- Salida tri-state TTL
- Tiene una resolución de 8 bits
- 5 Volts de alimentación
- Tiempo de conversión de 100 $\mu$ S
- Cuenta con 8-canales multiplexados, que en este trabajo solo se utiliza un canal (IN0)
- Chip de 28 pines

Para realizar la conversión, este circuito requiere de una señal de reloj que puede ser desde 10 Khz. hasta 1280 Khz. y que a continuación se menciona.

### 3.1.4.5 Generador de “CLK” para el circuito ADC

Un flip-flop JK sirve como divisor de frecuencia de la señal ALE del microcontrolador para generar una frecuencia de 1Mhz.

Es posible usar la configuración que se muestra en la figura 3-10

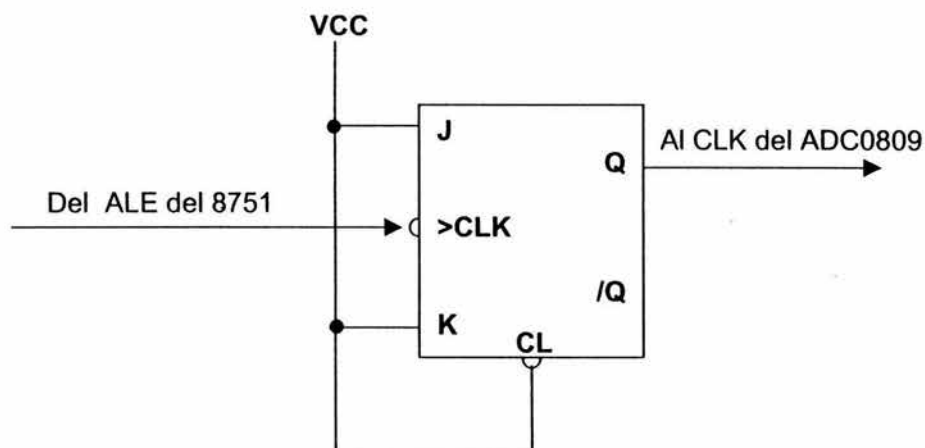


Figura 3-10 Diagrama de la señal de 1Mhz del CLK del ADC

El SN74LS73 es un chip de 14 pines que contiene 2 flip-flops JK con clock y clear independientes y cuya tabla de funciones se presenta a continuación.

TABLA DE VERDAD

| ENTRADAS |     |   |   | SALIDAS        |     |
|----------|-----|---|---|----------------|-----|
| /CLR     | CLK | J | K | Q              | /Q  |
| L        | X   | X | X | L              | H   |
| H        |     | L | L | Q0             | /Q0 |
| H        |     | H | L | H              | L   |
| H        |     | L | H | L              | H   |
| H        |     | H | H | COMPLEMENTARIO |     |
| H        | H   | X | X | Q0             | /Q0 |

H=HIGH nivel de voltaje alto

L=LOW nivel de voltaje bajo

X=NO IMPORTA

Figura 3-11 Tabla de verdad del flip-flop JK 74LS73

En las figuras 3-10 y 3-11 se observa que para este caso tan solo se utiliza una combinación cuya salida es el valor complementario, ya que el clear (CL o CLR), J y K están conectados a 5 volts (H) logrando así que cada vez que se genere el clock (CLK) en las salidas tendremos un cambio de estado.

El pin 30 del microcontrolador es ALE que se usa para habilitar el latch de la parte baja de las direcciones y que emite una señal a una tasa constante de 1/6 de la frecuencia de oscilación (12MHz). Por tal motivo se suele utilizar como reloj de propósito externo como en este caso que se utiliza para generar la frecuencia necesaria de 1MHz. para el CLK del ADC0809. En la figura 3-12 se observa claramente la secuencia de estados de las señales cuando está configurado como se mostró en la figura 3.10.



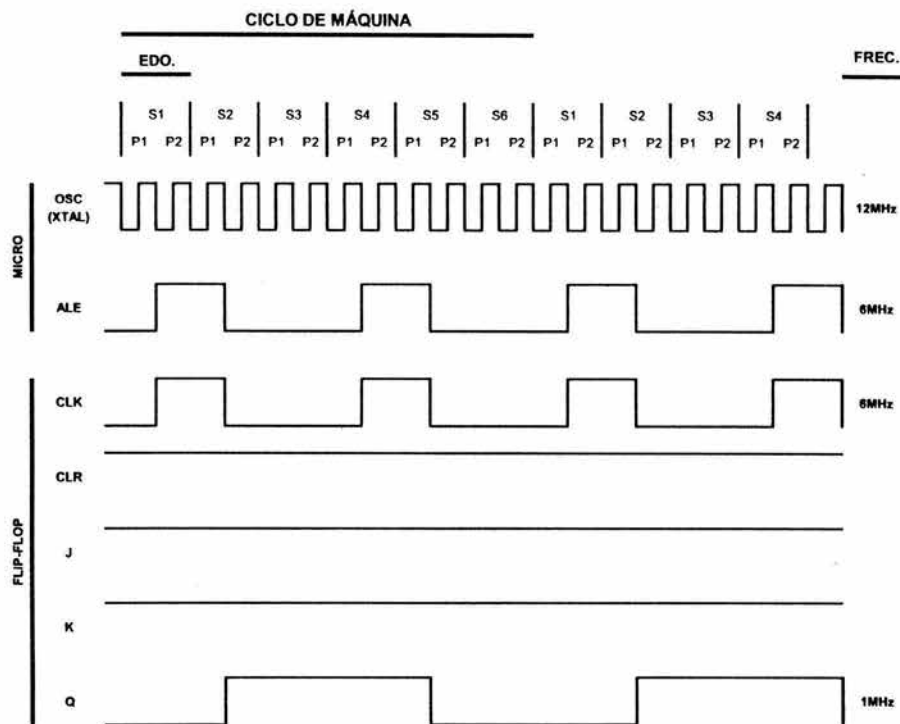


Figura 3-12 Secuencia de estados que explica el CLK del ADC

### 3.1.4.6 Convertidor Digital Analógico (DAC0800)

Este convertidor realiza el proceso inverso al ADC. Debido a que los circuitos digitales trabajan con señales digitales (unos o ceros) y la información que queremos en la salida es una señal analógica (voz), existen los convertidores digital-analógico que como su nombre lo indica convierten la información digital en analógica.

Las características principales de este convertidor son:

- Alto rango de suministro de energía  $\pm 4.5V$  a  $\pm 18V$
- Permite la interfase directa con circuitos TTL, CMOS, PMOS y otros
- Bajo costo
- Chip de 16 pines

El diagrama eléctrico de conexión utilizado, es uno propuesto como aplicación típica en la hoja de especificación del DAC0800 y se muestra en la siguiente figura.

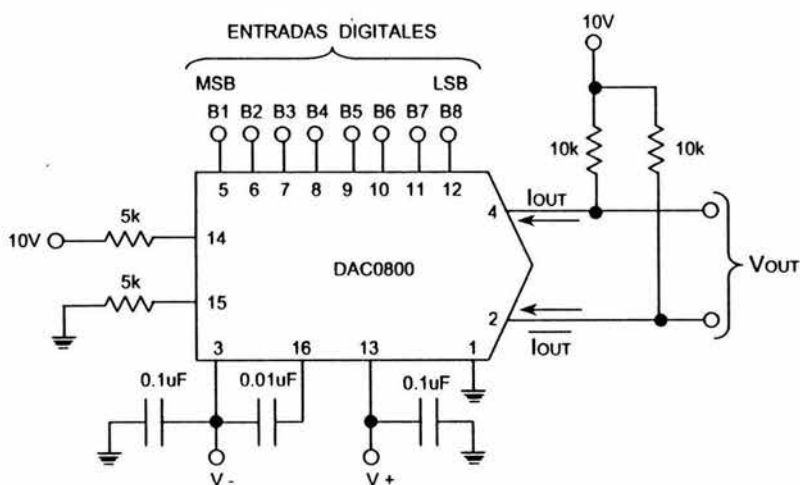


Figura 3-13 Diagrama eléctrico del convertidor digital-analógico

### 3.1.4.7 Amplificador Operacional (TL081)

Debido a que el tipo de salida del DAC0800 es de corriente, pin 2 y 4 de la figura 3-13, se requiere de un amplificador operacional a la salida.

La característica principal del amplificador operacional es que tiene una alta impedancia en su entrada y una baja impedancia en su salida, esto evita efecto de carga al ser conectado a otro dispositivo.

La figura 3-14 muestra la manera en que el amplificador operacional TL081 debe ser conectado al DAC0800:

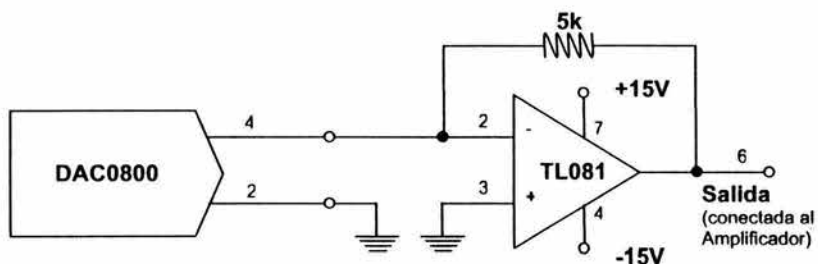


Figura 3-14 Diagrama salida de baja impedancia positiva

Para que la señal de salida del TL081 pueda ser escuchada, debe ir conectada al amplificador y del amplificador a la bocina.

### 3.1.4.8 Controlador de teclado (74C922)

Como ya se especificó, dentro de los elementos de entrada, el teclado es un arreglo de 4 x 2 botones tipo “un polo dos tiros”, que para poder tener comunicación con el sistema requiere un controlador de teclado. En este caso se propone un 74C922 que puede identificara hasta 16 teclas.

Cuando cualquier botón se oprima, el circuito controlador de teclado emitirá una señal DA (dato listo) que generará la interrupción externa 0 (INT0) del microcontrolador mediante un inversor implementado en la GAL16V8. Ocurrido lo anterior, se envía la señal de habilitación de salida del dato del controlador de teclado (/OE). Así, el microcontrolador realizará las operaciones específicas de acuerdo a la tecla que haya sido oprimida.

La siguiente figura muestra el diagrama general del controlador de teclado.

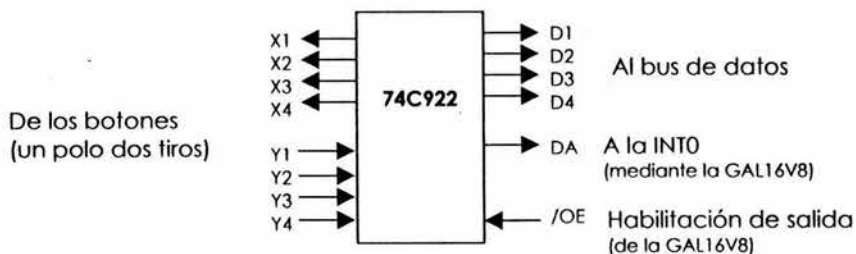


Figura 3-15 Controlador de teclado 74C922

Las características principales del controlador de teclado son:

- Codificador de 16-Teclas de la familia CMOS
- El escaneo del teclado puede ser implementado por un clock externo o por un capacitor externo
- No requiere diodos en el arreglo de switches para eliminar los teclas fantasmas
- Cuando una entrada de tecla es válida, el Dato Disponible (Data Available) se va a estado alto y regresa a estado bajo cuando deja de presionarse la tecla, incluso si se presiona otra tecla. Estos cambios de alto a bajo siempre después de un periodo normal de debouncer para evitar los rebotes
- Tiene un registro interno que almacena la última tecla presionada

### **3.1.4.9 Memoria externa**

En la memoria externa se guardan los datos (3 segundos de grabación de voz por mensaje). En este trabajo se propone utilizar la memoria Flash de alta densidad Am29F040.

Para este trabajo también pudo haberse utilizado una memoria EEPROM en lugar de la memoria Flash. Para este caso se requeriría de una memoria RAM que funcione como memoria caché.

Al momento de grabar un mensaje, éste debería almacenarse primeramente en la RAM, ya que los "tiempos de escritura" de la EEPROM no son lo suficientemente rápidos para ir almacenando las muestras, y posteriormente transferir los datos a la EEPROM (tiempo imperceptible por el usuario) donde permanecerán almacenados aunque deje de estar alimentado el sistema. La lectura sí se realizaría directamente de la EEPROM ya que sus "tiempos de lectura" son lo suficientemente rápidos para reproducir un mensaje.

Por tal motivo, la diferencia entre utilizar una Flash y una EEPROM es eliminar el uso de la RAM. Esto genera ahorro de dinero y de espacio.

Considerando que el tiempo de conversión del ADC0809 es de 100  $\mu$ s, el periodo de muestreo no puede ser mayor a 10,000 muestras/seg (1/100e-6). Si queremos grabar 3 segundos por mensaje y son 8 mensajes necesitamos 24 seg. de grabación en 240,000 muestras. Y si además queremos que no solo sea un nivel de ocho mensajes sino dos niveles, necesitamos una memoria que pueda almacenar 480,000 muestras, por este motivo se propone utilizar una de 512,000 bytes, que se describe a continuación.

### **3.1.4.9.1 Memoria Flash (Am29F040)**

El circuito integrado Am29F040 es una memoria Flash que tiene la capacidad de almacenar hasta 4Mbits (512Kx8 bits) de información, que mantiene aunque no esté energizada.

Es una memoria de bajo costo, arquitectura de alta densidad, bajo consumo y alta confiabilidad. Estas características hacen que sea considerada como la mejor opción para este trabajo.

Estas son sus principales características técnicas:

- 5 Volts de alimentación para operaciones de lectura y escritura
- Compatible con los estándares JEDEC
- Garantía de al menos 100,000 ciclos de escribir / leer
- Tiempo de acceso máximo 55ns
- Cualquier combinación de sectores puede ser borrada. Incluso puede borrar el chip por completo
- Arquitectura de sectores uniformes de borrado es de 64k bytes cada uno
- Método de hardware que deshabilita cualquier operación de escritura o borrado de los sectores
- Bajo consumo de energía  
20mA corriente típica para lectura  
30mA corriente típica para programar/borrar  
1 $\mu$ A corriente típica para modo de espera
- Chip de 32 pines

### 3.1.4.10 Latch (DM74LS573)

Dos circuitos LATCH son utilizados en el módulo principal. El LATCH1 que se utiliza para capturar la parte baja de las direcciones que sirven para direccionar la memoria de datos externa. Y el LATCH2 que captura los datos para el convertidor digital-analógico.

Sus características técnicas principales son:

- Latch octal tipo-D con salidas 3-state
- Las entradas y las salidas en lados opuestos del chip facilitan la interconexión con el microprocesador
- Totalmente compatible con TTL y con CMOS
- Chip de 20 pines

Como se puede ver en la siguiente figura 3-16, si en /OE existe un "1" lógico (o estado alto H) sin importar que valores tengan LE y D, en la salida se tiene una alta impedancia. Si existe en /OE un "0" lógico y en LE un "1" lógico, el mismo valor que se encuentre en los pines de entrada se tendrá en los pines de salida. Y si existe en /OE un "0" lógico y LE cambia a un "0" lógico, se captura la información de 'D' al momento del cambio y permanece con esa información mientras LE no cambie.



| Output Enable | Latch Enable | D | Output O       |
|---------------|--------------|---|----------------|
| L             | H            | H | H              |
| L             | H            | L | L              |
| L             | L            | X | Q <sub>0</sub> |
| H             | X            | X | Z              |

L = estado bajo  
H = estado alto  
X = no importa  
Z = estado de alta impedancia  
Q<sub>0</sub> = condición previa de O

**Figura 3-16** Tabla de función del 74LS573

Relacionando lo que se mencionó anteriormente con la figura 3-16, tenemos que:

Los dos circuitos 'LATCH' que se utilizarán en el módulo principal, están conectados: en sus pines de entrada al bus de datos P0 y en sus pines de salida: uno de ellos (Latch1) a la memoria flash y el otro (Latch2) al convertidor digital-analógico (DAC0800).

Para ambos capturadores /OE (Output Enable) esta conectada a tierra (L), pues no necesitamos una alta impedancia (Z) a la salida en ningún momento, y LE (Latch Enable) es el que servirá para habilitarlos. En el LATCH1 estará conectado a la señal ALE del microcontrolador y en el LATCH2 a una de las salidas del decodificador.

### 3.1.4.11 Display Alfanumérico (AND491)

Como ya se especificó, uno de los elementos de salida, es la pantalla de cristal líquido LCD ya que en ella podemos visualizar los números, tanto de nivel como de mensaje y si se está grabando o reproduciendo.

La razón por la cual también se considera dentro de los elementos de proceso es por el tipo de conexión que requiere y que formaría parte del módulo principal, dentro de éste mismo se colocaría un header de 14 pines para conectar la pantalla de cristal líquido, AND 491.

La siguiente figura muestra el diagrama de conexión.

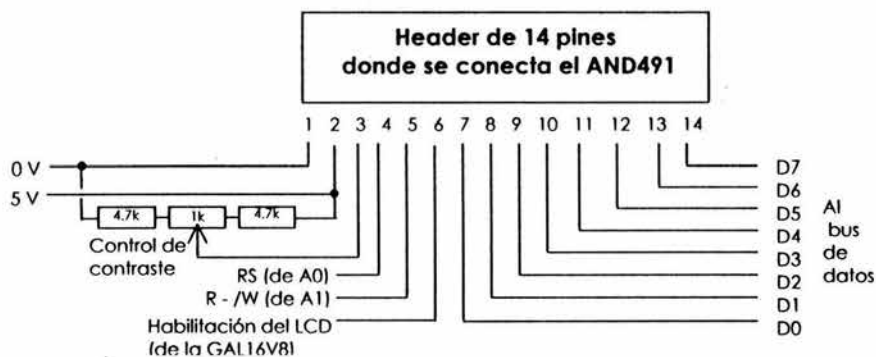


Figura 3-17 Pantalla de cristal líquido AND 491

Donde RS indica, si es "H" entrada de dato y si es "L" entrada de comando. Y, R - /W indica si es lectura o escritura.

### 3.1.5 Interconexión del intérprete gráfico

Anteriormente se mencionó cada uno de los dispositivos y su interrelación con otros dispositivos.

La mejor manera en que se puede apreciar la interconexión en conjunto del el módulo principal es en el circuito esquemático que se encuentra en el Apéndice D.

En base al circuito esquemático, se propone una rutina de programación que, teóricamente, en su conjunto forman el intérprete gráfico. Dicha rutina está en el Apéndice E.

La siguiente figura representa de forma general como la placa del módulo principal se conecta con los dispositivos.

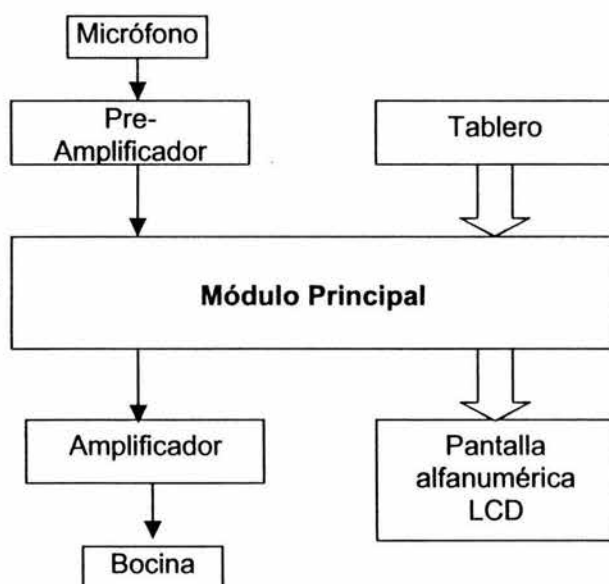


Figura 3-18 Conexión del Módulo Principal

## **CAPÍTULO 4**

---

---

### **CAPÍTULO 4: RESULTADOS**

## 4.1 Sistema Básico

Para comprobar la viabilidad del proyecto se creo lo que llamaremos **sistema básico**, que es la reducción del diseño a un solo botón.

Para la realización del sistema básico se utilizó el diseño de la placa desarrollada en la materia de microcomputadoras del décimo semestre de la carrera.

La figura 4-1 muestra el diagrama de sistema básico en el que se observan las siguientes diferencias con el diseño propuesto para este trabajo (Capítulo 3):

- No tiene controlador de teclado, debido a que solo se utilizará un botón.
- No tiene display, ya que es suficiente poner unos leds como indicadores de tiempos de lectura y escritura.
- Utiliza otro número de microcontrolador pero que pertenece a la misma familia MSC-51, la diferencia entre ellos es la memoria ROM interna. El utilizado en el sistema básico no tiene ya que está conectado a una memoria externa EPROM en donde se graba el código.
- Tiene una RAM donde se almacena el mensaje y una EPROM donde está el código del programa.
- Se utilizó un circuito TTL como decodificador y no un PLD (GAL16v8), se requirió de un circuito negador (NOT) para las habilitaciones que requieren de activos altos.

- Se utilizó un chip de compuertas NAND para adaptar los estados lógicos necesarios para la habilitación del decodificador y de la interrupción INT1. Dibujados en circuito esquemático del Apéndice F.

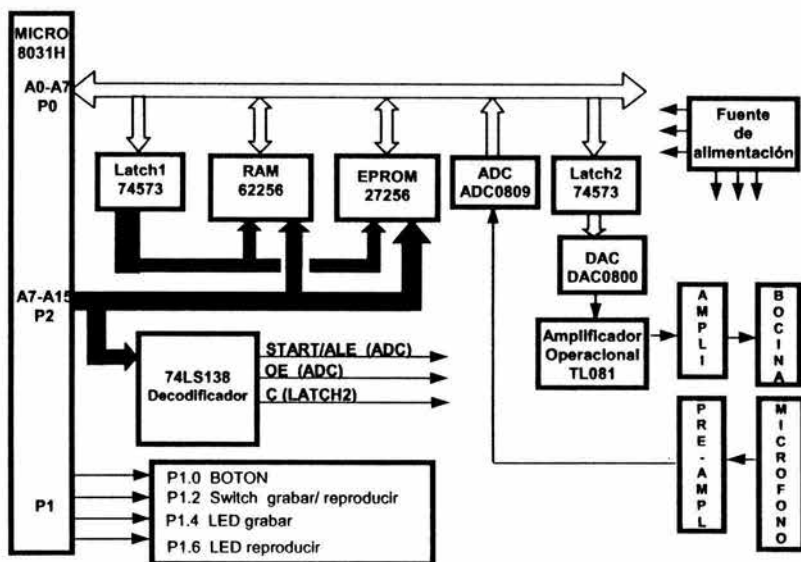


Figura 4-1 Diagrama del sistema básico

Como ya se mencionó, para el sistema básico utiliza 3 circuitos que no se mencionan en el capítulo 3 pero que fueron utilizados para la creación del sistema básico que son: el Flip-Flop JK (SN74LS73), el decodificador demultiplexor de 8 líneas (SN74LS138) y el buffer octal con salidas 3-State (SN74LS540) que se mencionan brevemente a continuación.

### 4.1.1 Decodificador de 8 líneas (SN74LS138)

Este circuito de 16 pines se utilizó como decodificador. Tiene 3 pines de entrada, 3 de habilitación, 8 de salida, alimentación (Vcc) y tierra (GND).

**TABLA DE VERDAD**

| No. | ENTRADAS |      |    |   |   |   | SALIDAS |     |     |     |     |     |     |     |
|-----|----------|------|----|---|---|---|---------|-----|-----|-----|-----|-----|-----|-----|
|     | /G2B     | /G2A | G1 | A | B | C | /O0     | /O1 | /O2 | /O3 | /O4 | /O5 | /O6 | /O7 |
| 1   | H        | X    | X  | X | X | X | H       | H   | H   | H   | H   | H   | H   | H   |
| 2   | X        | H    | X  | X | X | X | H       | H   | H   | H   | H   | H   | H   | H   |
| 3   | X        | X    | L  | X | X | X | H       | H   | H   | H   | H   | H   | H   | H   |
| 4   | L        | L    | H  | L | L | L | L       | H   | H   | H   | H   | H   | H   | H   |
| 5   | L        | L    | H  | H | L | L | H       | L   | H   | H   | H   | H   | H   | H   |
| 6   | L        | L    | H  | L | H | L | H       | H   | L   | H   | H   | H   | H   | H   |
| 7   | L        | L    | H  | H | H | L | H       | H   | H   | L   | H   | H   | H   | H   |
| 8   | L        | L    | H  | L | L | H | H       | H   | H   | L   | H   | H   | H   | H   |
| 9   | L        | L    | H  | H | L | H | H       | H   | H   | H   | L   | H   | H   | H   |
| 10  | L        | L    | H  | L | H | H | H       | H   | H   | H   | H   | L   | H   | H   |
| 11  | L        | L    | H  | H | H | H | H       | H   | H   | H   | H   | H   | H   | L   |

H=HIGH nivel de voltaje alto

L=LOW nivel de voltaje bajo

X=NO IMPORTA

**Figura 4-2** Tabla de verdad del decodificador 74LS138

En la columna de la extrema izquierda de la Figura 4-2 se numeran cada una de las combinaciones posibles. En la configuración del sistema básico /G2A y /G2B siempre están en "L", por tal motivo, las combinaciones 1 y 2 no pueden presentarse. Cuando G1 es "L" solo existe una posibilidad y es la combinación 3, que sin importar el valor de A, B y C todas las salidas están en "H".

G1 es "H" cuando se desea habilitar algún periférico, donde A, B y C designarán el dispositivo a habilitarse. Así, en el código del sistema básico (Apéndice H) tenemos que la dirección 09000H se usa para habilitar (OE) el ADC, la 0A000H para activar START/ALE del ADC y la 0C000H para habilitar (C) el LATCH2 que se conecta con el DAC.

Todo lo anterior se representa en la siguiente tabla.

| Puerto                |       | P2   |      |      |      |      |      |      |      | P0   |      |      |      |      |      |      |      |
|-----------------------|-------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
|                       |       | P2.7 | P2.6 | P2.5 | P2.4 | P2.3 | P2.2 | P2.1 | P2.0 | P0.7 | P0.6 | P0.5 | P0.4 | P0.3 | P0.2 | P0.1 | P0.0 |
| Direcciones (Address) |       | A15  | A14  | A13  | A12  | A11  | A10  | A9   | A8   | A7   | A6   | A5   | A4   | A3   | A2   | A2   | A0   |
| C<br>(LATCH2)         | C000H | 1    | 1    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
|                       | A000H | 1    | 0    | 1    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
|                       | 9000H | 1    | 0    | 0    | 1    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |

Figura 4-3 16 Bits de direcciones en la habilitación de periféricos



### 4.1.2 Buffer octal inversor con salidas 3-State (SN74LS540)

Como se puede ver, las salidas en la figura 4-2 son activos bajos. Y las 3 señales que se habilitan a través del decodificador (C, START/ALE y OE) se activan con bajos, por tal motivo es necesario utilizar un circuito negador, para este caso se utilizó este buffer octal inversor ya que sus pines de entrada y salida están en lados opuestos del chip lo que facilitan la interconexión con el decodificador.

| No. | ENTRADAS |     |    | SALIDAS |
|-----|----------|-----|----|---------|
|     | /G1      | /G2 | Ax | Yx      |
| 1   | L        | L   | H  | L       |
| 2   | H        | X   | X  | Z       |
| 3   | X        | H   | X  | Z       |
| 4   | L        | L   | L  | H       |

H=HIGH nivel de voltaje alto

L=LOW nivel de voltaje bajo

X=NO IMPORTA

Z=alta impedancia

**Figura 4-4** Tabla de verdad del buffer octal (SN74LS540)

De la tabla de verdad de la figura 4-4 podemos concluir que solo las combinaciones no. 1 y 4 se utilizan en el sistema básico, ya que /G1 y /G2 están conectados a tierra como se observa en el circuito esquemático del Apéndice F.

## 4.2 Pruebas esenciales

Se buscó una forma económica y práctica de interconectar los dispositivos que en conjunto formen el diagrama representado en la figura 4-1.

Como primer paso, se buscaron los kits de preamplificación y amplificación para conectarlos al micrófono y a la bocina respectivamente. Los diagramas eléctricos y la lista de partes de cada uno de ellos están especificados en los Apéndices A y B.

Se utilizó una placa del diseño visto en la materia de microcomputadoras, del décimo semestre.

Al sistema básico lo conforman de 5 placas y los dispositivos de Entrada/Salida. Así, en cada una de las placas tenemos: el módulo principal, la fuente de poder, el pre-amplificador, el amplificador y el convertidor digital-analógico.

Utilizando un header y un cable plano de 8 líneas se interconectan el módulo principal y el convertidor digital-analógico (DAC0800). Mediante alambres se conectan a la placa del módulo principal, la alimentación, la entrada 0 (IN0) del convertidor analógico-digital, la señal (voz) proveniente del pre-amplificador y las 4 señales de los puertos P0.0, P0.2, P0.4 y P0.6, que son: el botón, el switch de selección Grabar/Reproducir, el led rojo de grabación y el led verde de reproducción respectivamente.

Debido a que únicamente se va a guardar la información de un mensaje de 3 segundos, son suficientes 32Kbits de la RAM que tiene el módulo principal, sin embargo, debido a que es una memoria volátil la información se perderá si deja de alimentarse el circuito. Para que esto no suceda podría conectarse el pin de alimentación (VCC) a una pila.

El circuito esquemático del sistema básico se encuentra en el Apéndice F, el diagrama de flujo en el Apéndice G y el código de programación del microcontrolador en el Apéndice H, para este último se utilizó un programador universal XELTEC SUPERPRO III.

Primeramente se comprobó que los kits de preamplificación y amplificación funcionaran correctamente.

De manera independiente se conectó el convertidor digital analógico y se hicieron pruebas variando el voltaje manualmente en los pines de entrada ("1" lógico = 5V y "0" lógico = 0V) dando como resultado la siguiente tabla:

| No.<br>decimal | ENTRADAS<br>(DAC0800) |   |   |   |   |   |   |     | SALIDAS<br>(TL081)<br>en Volts |
|----------------|-----------------------|---|---|---|---|---|---|-----|--------------------------------|
|                | MSB                   |   |   |   |   |   |   | LSB |                                |
| 0              | 0                     | 0 | 0 | 0 | 0 | 0 | 0 | 0   | -5.2                           |
| 1              | 0                     | 0 | 0 | 0 | 0 | 0 | 0 | 1   | -5.1                           |
| 3              | 0                     | 0 | 0 | 0 | 0 | 0 | 1 | 1   | -5.0                           |
| 7              | 0                     | 0 | 0 | 0 | 0 | 1 | 1 | 1   | -4.8                           |
| 15             | 0                     | 0 | 0 | 0 | 1 | 1 | 1 | 1   | -4.6                           |
| 31             | 0                     | 0 | 0 | 1 | 1 | 1 | 1 | 1   | -4.0                           |
| 63             | 0                     | 0 | 1 | 1 | 1 | 1 | 1 | 1   | -2.8                           |
| 127            | 0                     | 1 | 1 | 1 | 1 | 1 | 1 | 1   | -0.5                           |
| 128            | 1                     | 0 | 0 | 0 | 0 | 0 | 0 | 0   | -0.4                           |
| 192            | 1                     | 1 | 0 | 0 | 0 | 0 | 0 | 0   | 1.8                            |
| 224            | 1                     | 1 | 1 | 0 | 0 | 0 | 0 | 0   | 2.9                            |
| 240            | 1                     | 1 | 1 | 1 | 0 | 0 | 0 | 0   | 3.5                            |
| 248            | 1                     | 1 | 1 | 1 | 1 | 0 | 0 | 0   | 3.8                            |
| 252            | 1                     | 1 | 1 | 1 | 1 | 1 | 0 | 0   | 3.9                            |
| 254            | 1                     | 1 | 1 | 1 | 1 | 1 | 1 | 0   | 4.0                            |
| 255            | 1                     | 1 | 1 | 1 | 1 | 1 | 1 | 1   | 4.2                            |

Gracias a que los resultados de la tabla fueron los esperados, se continuó con la conexión, utilizando caimanes, de las 5 placas.

Primeramente se probó con un código que únicamente grababa un mensaje, esperaba 3 segundos y lo reproducía constantemente hasta que el dispositivo dejara de alimentarse. Con este código se hicieron una serie de pruebas modificando únicamente el número de muestras y el intervalo entre cada una de ellas.

De estas pruebas se encontró que dos datos importantes son: la frecuencia de muestreo, relacionada directamente con el ancho de banda, y el tiempo de conversión del convertidor analógico digital.

Esto es, si el rango de ancho de banda (BW) de la voz humana está entre los 500 y los 4000 Hz. Según el teorema de muestreo es suficiente con tener una frecuencia de muestreo mayor o igual a 8000 Hz. ( $F_m=2*BW$ ) para poder recuperar esta señal. Partiendo de ello, tenemos un periodo de muestreo de  $125\mu S$  ( $T_m=1/F_m$ ).

Considerando que el tiempo de conversión del ADC0809 es de  $100\mu S$ , el periodo de muestreo antes calculado, se ajusta a esta condición. Y tenemos:

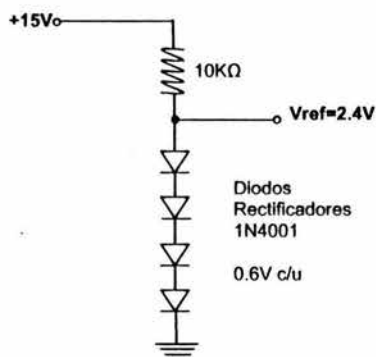
Si 1 muestra se toma cada  $125\mu S$ , ¿cuántas muestras se tomarán en 3 segundos? (duración del mensaje) R= En 24,000 muestras.

Muy probablemente utilizando un BW mayor se recuperaría un mensaje con mejor definición pero esto implicaría un periodo de muestreo cada vez mayor, limitada por el tiempo de conversión del convertidor.

Otra observación importante fue que la señal máxima de voz procedente del pre-amplificador es de aproximadamente 2 volts pico a pico, por esta razón el voltaje de referencia positivo no podría ser de 5 volts.

El ADC0809 realiza la conversión basándose en el "Voltaje de referencia -" y el "Voltaje de referencia +" que se indican en los pines 16 y 12 respectivamente. En este trabajo el  $V_{ref(-)}$  es de 0V y el  $V_{ref(+)}$  es de 2.4V, así, considerando la resolución de 8-bits, 0V estarán representados por 00000000 binario y 2.4V por 11111111 binario, este último representa el 255 decimal, entonces cada 9.4mV representará un cambio ( $2.4/255=9.4mV$ ).

A continuación, el diagrama eléctrico que ofrece los 2.4V utilizados como Vref(+) y que se encuentra en la placa del convertidor digital analógico.



**Figura 4-5 Diagrama eléctrico 15V - 2.4V**

Se realizaron algunas pruebas con un par de micrófonos y bocinas distintos, sin notarse un cambio significativo en la calidad del audio.

Finalmente se modificó el código que logró exitosamente la grabación y reproducción de un mensaje utilizando un botón y un switch de selección Grabar/Reproducir siguiendo los siguientes pasos:

1. Encender el dispositivo
2. Poner el switch en modo de grabar
3. Oprimir el botón
4. Grabar el mensaje (micrófono)
5. Poner el switch en modo de reproducir
6. Oprimir el botón
7. Escuchar el mensaje (bocina)

Es importante saber que:

- √ Mientras el dispositivo esté encendido y se haya grabado un mensaje, si el botón es presionado, el mensaje será escuchado.
- √ Una vez grabado el mensaje, si no se cambia de modo grabar a modo reproducir y se oprime nuevamente el botón, se sobrescribirá el mensaje.
- √ Durante los 3 segundos de grabación, un led rojo se encenderá. Si se oprime el botón en este tiempo, será ignorado esta última acción.
- √ Durante los 3 segundos de reproducción, un led verde se encenderá. Si se oprime el botón en este tiempo, será ignorado esta última acción.

## CONCLUSIONES

---

---



Se pudo construir un prototipo que funciona para grabar y reproducir un mensaje, lo cual garantiza que puede hacerse con un mayor número de botones. Existen muchas formas en que esto podría hacerse, desde la interconexión de más módulos de memoria hasta la creación de un ASIC (Circuito Integrado de Aplicación Específica) con el microcontrolador, las memorias y los convertidores integrados.

A pesar de que el microcontrolador MSC-51 cuenta tan solo con 16 líneas para direccionar memoria. Fue posible conectarlo con una memoria de 19 líneas (512Kbytes) utilizando 4 líneas del puerto 1, logrando así una paginación de 16 bloques de 32Kbytes cada uno, que es el tamaño requerido para el almacenamiento de un mensaje con duración de 3 segundos.

La memoria Flash Am29F040 es una excelente opción para este trabajo gracias a su gran capacidad de almacenamiento (4Mbits), a sus rápidos tiempos de acceso tanto de lectura como de escritura (55nseg) y a que es una memoria del tipo: no volátil.

Es posible la implementación del interprete gráfico de 8 botones x 2 niveles siguiendo el diseño propuesto en este trabajo.

## LITERATURA CONSULTADA

---

---

## **Bibliografía**

Linear Circuits (Amplifiers, Comparators and Special Functions)  
Data Book Volume 1  
Texas Instruments, 1989

Linear Circuits (Data Acquisition and Conversion)  
Data Book Volume 2  
Texas Instruments, 1989

Linear Circuits (Voltage Regulators and Supervisors)  
Data Book Volume 3  
Texas Instruments, 1989

Diseño Digital  
R. Alcubilla, J. Pons, D. Bardés  
Ediciones UPC, 1996

## **Páginas en Internet**

"80C51 family programmer's guide and instruction set"  
Philips Semiconductors (Tabla de SFR)  
<<http://www.eng.warwick.ac.uk/staff/elh/es153/00/8051soft.pdf>>  
[Consulta: 29 Enero 2004]

Aguirre Iñigo y Ezquerro José Ángel  
"Metodologías de diseño para computación reconfigurable"  
Universidad del País Vasco. Euskal Herriko Unibertsitatea.  
<[http://vgweb.upc.vg.eupvg.upc.es/web\\_eupvg/xic/arxiu\\_ponencias/R0101.pdf](http://vgweb.upc.vg.eupvg.upc.es/web_eupvg/xic/arxiu_ponencias/R0101.pdf)>  
[Consulta: 24 Noviembre 2003]

"Aplicación Industrial de Micros"  
Universidad Antonio Nariño  
Código: 444405, Capítulo 1  
<[http://www.geocities.com/micros\\_uan/cap11.html](http://www.geocities.com/micros_uan/cap11.html)>  
[Consulta: 07 Noviembre 2003]

Beltrán M. Nicolás y Domingo Morales L.  
"Modernización de la Enseñanza en Electrónica en la Facultad"  
Departamento de Ingeniería Eléctrica.  
Área de Telecomunicaciones y Electrónica  
<<http://www.dcc.uchile.cl/~ppoblete/cgi-bin/archivos-ed2002/20021001172509/EnsenElectronica.doc>>  
[Consulta: 07 Noviembre 2003]

"Características de la voz"  
Información Médica  
<<http://www.sinfomed.org.ar/mains/infopaci/preven2.htm>>  
[Consulta: 05 Noviembre 2003]

Corona Villaseñor Salvador, Palacios Escobar José Luis, Salas Barajas Fernando  
"Detector de frecuencias cardiacas"  
(Características de los Puertos del microcontrolador)  
<[http://proton.ucting.udg.mx/expodec/abr99/e35/e\\_35.htm](http://proton.ucting.udg.mx/expodec/abr99/e35/e_35.htm)>  
[Consulta: 02 Marzo 2004]

Enabling Devices  
Toys for Special Children, Inc.  
<<http://www.enablingdevices.com>>  
[Consulta: 07 Noviembre 2003]

Fajardo Contreras Waldo, Dr.  
Dpto. Ciencias de la Computación e I. A.  
Universidad de Granada  
<[http://www.etsi2.ugr.es/depar/ccia/Inteligencia\\_Artificial\\_Practica\\_1/microcontroladores.htm](http://www.etsi2.ugr.es/depar/ccia/Inteligencia_Artificial_Practica_1/microcontroladores.htm)>  
[Consulta: 07/Noviembre/2003]

"Introducción a las familias lógicas"  
Profesora Responsable: Ana Toledo Moreo  
UPC, Universidad Politécnica de Cartagena, España  
Departamento de Tecnología Electrónica  
<[http://www.dte.upct.es/docencia/temarios/tecn\\_electricidad\\_elect\\_dig/Tema4.PDF](http://www.dte.upct.es/docencia/temarios/tecn_electricidad_elect_dig/Tema4.PDF)>  
[Consulta: 17 Noviembre 2003]

Jiménez Edward y Pérez Luis  
"Introducción a microcontroladores"  
Unidad de Investigación y Desarrollo Tecnológico  
<<http://petroecuador.com.ec/uidt>>  
[Consulta: 28 Octubre 2003]

Larraz Isturiz Cristina, Logopeda del Ceapat  
"Ayudas técnicas para el diálogo"  
<[http://www.ceapat.org/6\\_centro/boletin/boletin\\_19.doc](http://www.ceapat.org/6_centro/boletin/boletin_19.doc)>  
[Consulta: 26 Noviembre 2003]

Mayer-Johnson, Inc.  
<<http://www.mayer-johnson.com>>  
[Consulta: 19 Octubre 2003]

Rodero Antón Emma, Universidad Pontificia de Salamanca  
"El tono de la voz masculina y femenina en los informativos  
radiofónicos: un análisis comparativo"  
(Frecuencia de la voz)  
<<http://bocc.ubi.pt/pag/rodero-emma-tono-voz-femenina.pdf>>  
[Consulta: 05 Noviembre 2003]

"Teorema del Muestreo"  
Breve historia y teoría del Sampler. Archivo Resonancias  
<[http://www.ccapitalia.net/reso/articulos/historia\\_sampler/historiasampler.pdf](http://www.ccapitalia.net/reso/articulos/historia_sampler/historiasampler.pdf)>  
[Consulta: 23 Noviembre 2003]

Vega S. Alejandro, Dr.  
"Manual del Microcontrolador 8051"  
<[www.die.udec.cl/~diseno/materiales/Cap5.pdf](http://www.die.udec.cl/~diseno/materiales/Cap5.pdf)>  
[Consulta: 28 Enero 2004]

"What is Flash Memory?"  
Intel Corporation  
<<http://www.intel.com/design/flash/articles/what.htm>>  
[Consulta: 23 Febrero 2004]

## **Hojas de especificación**

74LS540

OCTAL BUFFER/LINE DRIVER  
WITH 3-STATE OUTPUTS

© Motorola Inc., 1992

Disponible en: <<http://www.ee.mut.ac.th/datasheet/sn74ls540.pdf>>

74LS73

DUAL J-K FLIP-FLOPS WITH CLEAR

© 1988, Texas Instruments Incorporated

Disponible en: <<http://focus.ti.com/docs/prod/folders/print/sn7473.html>>

Am29F040

4 Megabit (524,288 x 8-Bit) CMOS 5.0 Volt-only,  
Sector Erase Flash Memory

Copyright © 1996 Advanced Micro Devices, AMD

Disponible en: <[http://www.amd.com/us-en/assets/content\\_type/white\\_papers\\_and\\_tech\\_docs/17113.pdf](http://www.amd.com/us-en/assets/content_type/white_papers_and_tech_docs/17113.pdf)>

AND - Intelligent Alphanumeric Displays

Application Notes

© 1999 Purdy Electronics Corporation

Disponible en: <

<http://www.purdyelectronics.com/ANDDisplays/Intchar.cfm>>

DAC0800/DAC0802

8-Bit Digital-to-Analog Converters

© 2001 National Semiconductor Corporation

Disponible en: < <http://www.national.com/pf/DA/DAC0800.html>>

DM74LS573

Octal D-Type Latch with 3-STATE Outputs

© 2000 Fairchild Semiconductor Corporation

Disponible en:

<<http://www.fairchildsemi.com/pf/DM/DM74ALS573B.html>>

GAL22V10

High Performance E2CMOS PLD

Generic Array Logic™

© 1997 Lattice Semiconductor Corporation

Disponibile en: <

<http://www.latticesemi.com/account/download.cfm?AMID=106>>

HM62256B Series

256k SRAM (32-kword X 8-bit)

© 1997 Hitachi, Ltd

Disponibile en: < [http://www.geocities.com/nazlen\\_2000/web/62256.pdf](http://www.geocities.com/nazlen_2000/web/62256.pdf)>

LM387/LM387A

Low Noise Dual Preamplifier

© 1995 National Semiconductor Corporation

Disponibile en: <<http://www.national.com/pf/LM/LM387.html>>

LM78XX

Series Voltage Regulators

© 2000 National Semiconductor Corporation

Disponibile en: < <http://www.national.com/ds/LM/LM7512C.pdf>>

LM79XX Series

3-Terminal Negative Regulators

© 2001 National Semiconductor Corporation

Disponibile en: < [www.national.com/pf/LM/LM7905.html](http://www.national.com/pf/LM/LM7905.html)>

MCS® -51

8-Bit Control-Oriented Microcontrollers

© 1993 INTEL Corporation

Disponibile en: < <http://www.intel.com/design/mcs51/>>

MM74C922 • MM74C923

16-Key Encoder • 20-Key Encoder

© 2001 Fairchild Semiconductor Corporation

Disponibile en: < <http://www.fairchildsemi.com/ds/MM/MM74C923.pdf>>

SN74LS00

Quad 2-Input NAND Gate

© 2001 Semiconductor Components Industries, LLC

Disponible en:

<

<http://www.onsemi.com/site/products/summary/0,4450,SN74LS00,00.html>>

SN74LS138

1-of-8 Decoder / Demultiplexer

© Semiconductor Components Industries, LLC, 1999

<[www.onsemi.com/site/products/summary/0,4450,SN74LS138,00.html](http://www.onsemi.com/site/products/summary/0,4450,SN74LS138,00.html)>

TDA2030

14W Hi-Fi AUDIO AMPLIFIER

® 1993 SGS-THOMSON Microelectronics

Disponible en: <<http://www.st.com/stonline/books/pdf/docs/1458.pdf>>

TL081 Wide Bandwidth JFET

Input Operational Amplifier

© 1995 National Semiconductor Corporation

Disponible en: <<http://www.national.com/pf/TL/TL081.html>>



## Otros

Berlanga R. Gabriela, 2003 "Apuntes sobre CAA y Dibujos para pruebas"

Inédito, Comunicación Personal

M en C. Martín Santiago Domínguez González, agosto-diciembre 2001

"Diseño Digital", apuntes

Universidad Americana de Acapulco

Séptimo semestre

M en C. Martín Santiago Domínguez González, enero-julio 2002

"Diseño de Sistemas Digital", apuntes

Universidad Americana de Acapulco

Octavo semestre

Ing. Carlos Cruz Arizmendi, agosto-diciembre 2002

"Comunicaciones Digitales", apuntes

Universidad Americana de Acapulco

Noveno semestre

Ing. Luis Rubén Neri Martínez, agosto-diciembre 2002

"Organización de Computadoras", apuntes

Universidad Americana de Acapulco

Noveno semestre

M en C. Martín Santiago Domínguez González, enero-julio 2003

"Microcomputadoras", apuntes

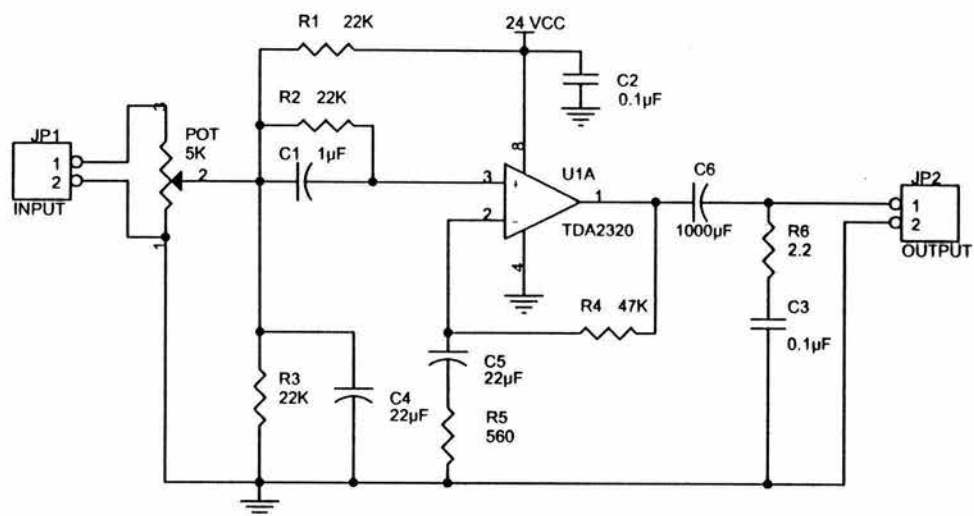
Universidad Americana de Acapulco

Décimo semestre

## **APÉNDICE A**

---

### **DIAGRAMA ELÉCTRICO DEL PREAMPLIFICADOR**



| PRE-AMPLIFICADOR |                             |
|------------------|-----------------------------|
| CLAVE            | COMPONENTES                 |
| CI - 1           | Circuito Integrado LM 387   |
| R1,R2            | 100KΩ @ ¼ W                 |
| R4               | 56KΩ @ ¼ W                  |
| R3               | 39KΩ @ ¼ W                  |
| R5               | 330Ω @ ¼ W                  |
| R6               | 4.7KΩ @ ¼ W                 |
| C1               | Cerámico de 0.0022uf @ 10V  |
| C2,C7,C8         | Cerámico de 0.1uf @ 10v     |
| C3,C4,C5         | Cerámico de 0.01uF @ 10v    |
| C6               | Electrolítico de 47uf @ 10v |

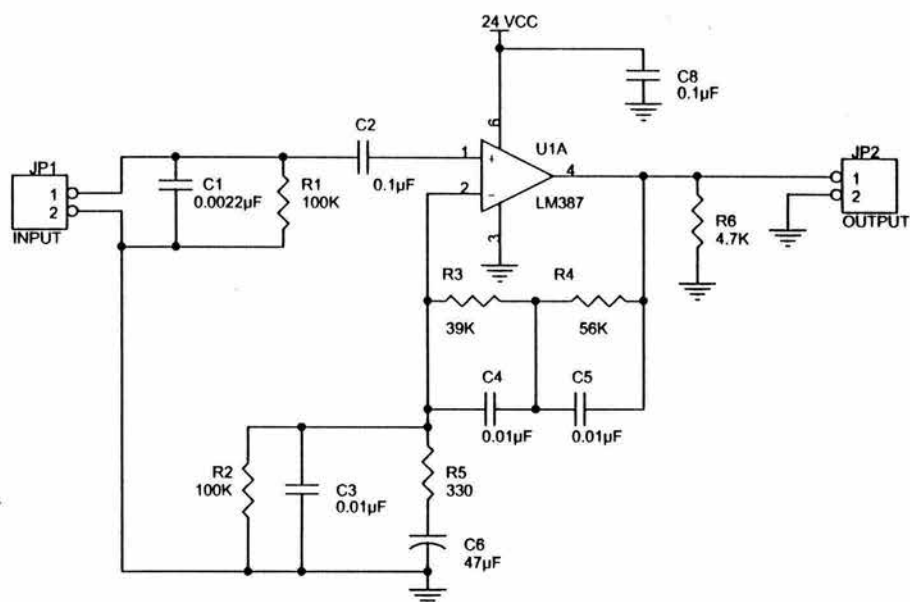
Figura A-1 Diagrama eléctrico y lista de partes del pre-amplificador

## **APÉNDICE B**

---

---

### **DIAGRAMA ELÉCTRICO DEL AMPLIFICADOR**



| AMPLIFICADOR |                               |
|--------------|-------------------------------|
| CLAVE        | COMPONENTES                   |
| CI - 1       | Circuito Integrado TDA 2030   |
| R1,R2,R3     | 22K $\Omega$ @ ½ W            |
| R4           | 47K $\Omega$ @ ½ W            |
| R5           | 560 $\Omega$ @ ½ W            |
| R6           | 2.2 $\Omega$ @ ½ W            |
| C1           | Electrolítico de 1Uf @ 16V    |
| C2,C3        | Cerámico de 0.1uf @ 16V       |
| C4,C5        | Electrolítico de 22uf @ 16v   |
| C6           | Electrolítico de 1000uf @ 16v |
| Pot          | Potenciómetro de 5K $\Omega$  |

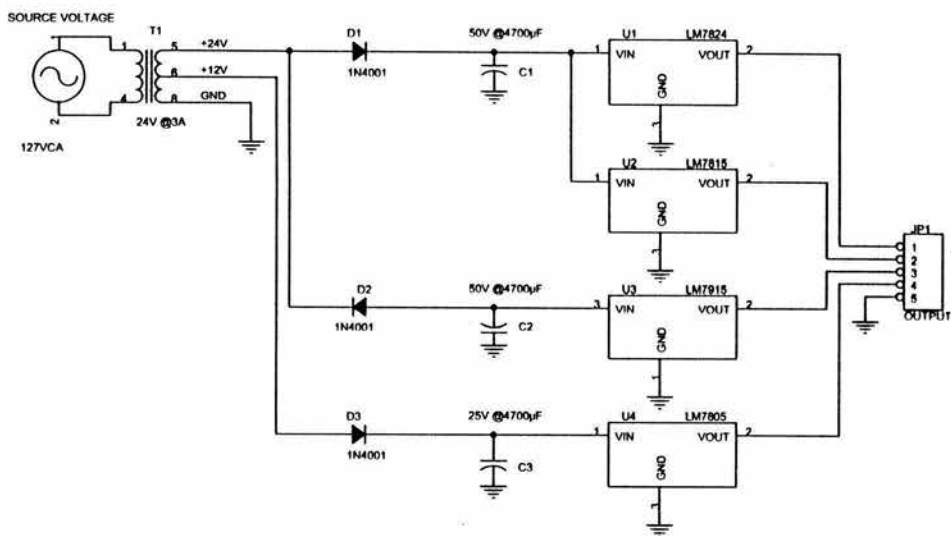
Figura B-1 Diagrama eléctrico y lista de partes del amplificador

## APÉNDICE C

---

---

### DIAGRAMA ELÉCTRICO DE LA FUENTE DE ALIMENTACIÓN



| FUENTE DE ALIMENTACION |                              |
|------------------------|------------------------------|
| CLAVE                  | COMPONENTES                  |
| IN4001                 | Diodo rectificador @1A       |
| C1                     | 50V @ 4700uF                 |
| C2                     | 50V @ 4700uF                 |
| C3                     | 25V @ 4700uF                 |
| 7824                   | Regulador de voltaje de 24V  |
| 7815                   | Regulador de voltaje de 15V  |
| 7915                   | Regulador de voltaje de -15V |
| 7805                   | Regulador de voltaje de 5V   |
| Transformador          | 24V @3A                      |

Figura C-1 Diagrama eléctrico y lista de partes de la fuente de alimentación

## **APÉNDICE D**

---

---

### **CIRCUITO ESQUEMÁTICO DEL MÓDULO PRINCIPAL DEL INTÉRPRETE GRÁFICO**



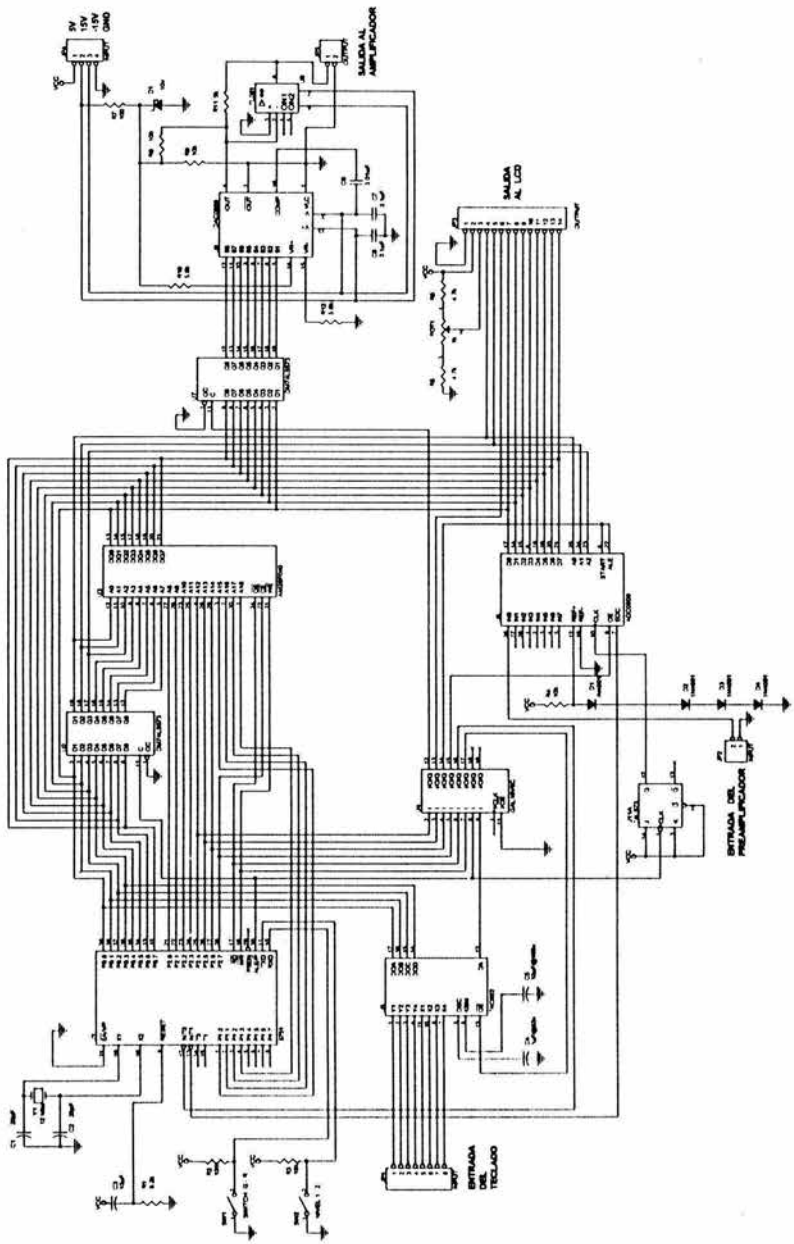


Figura D-1 Circuito esquemático del Intérprete Gráfico

## APÉNDICE E

---

---

### DIAGRAMAS DE FLUJO DEL INTÉRPRETE GRÁFICO

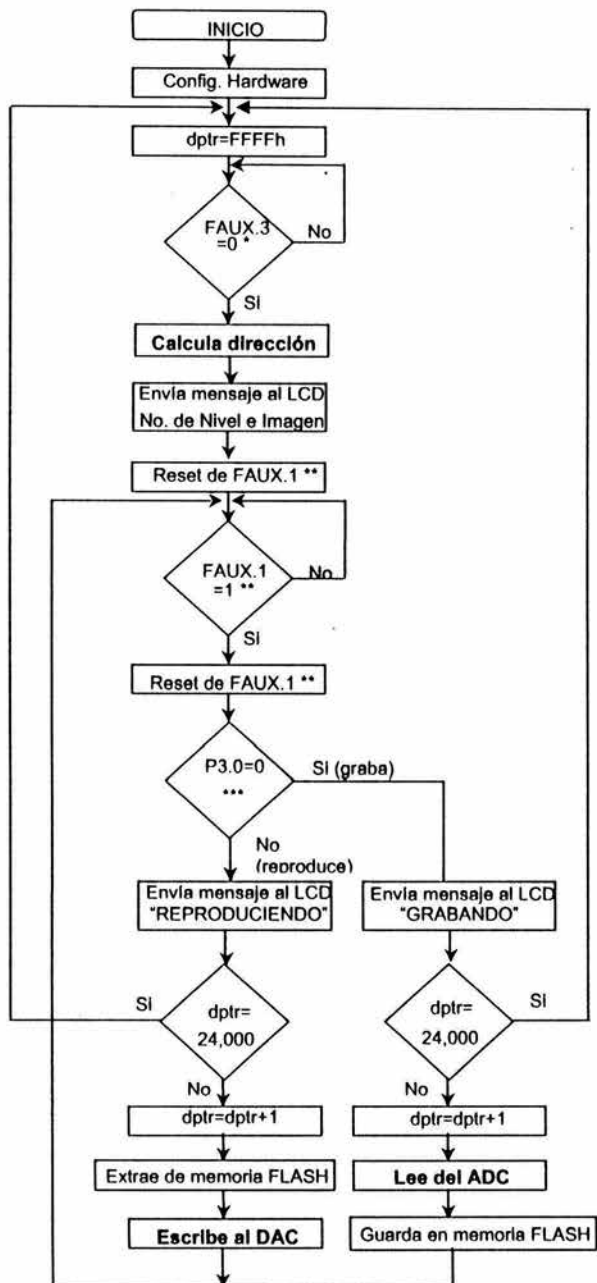


Figura E-1 Diagrama de Flujo del Intérprete Gráfico

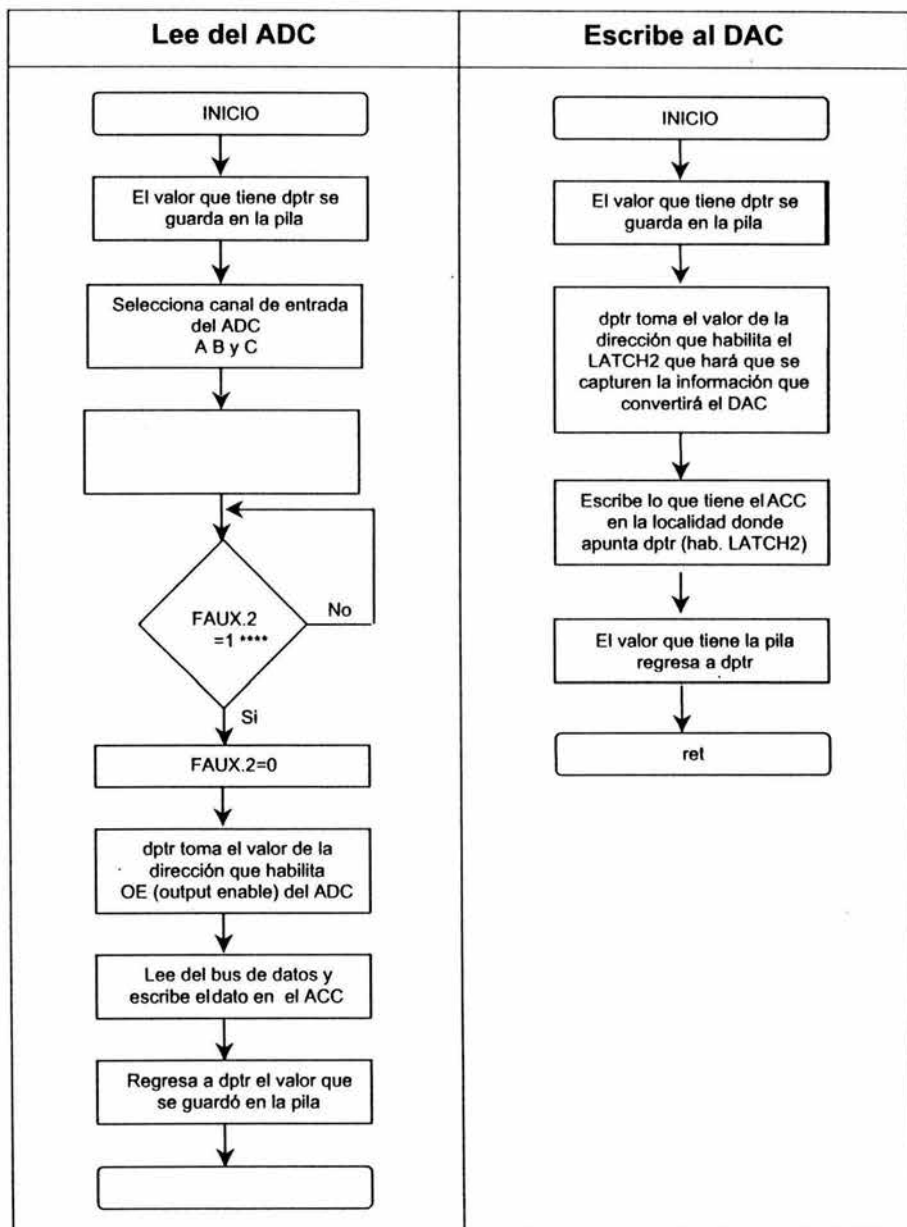


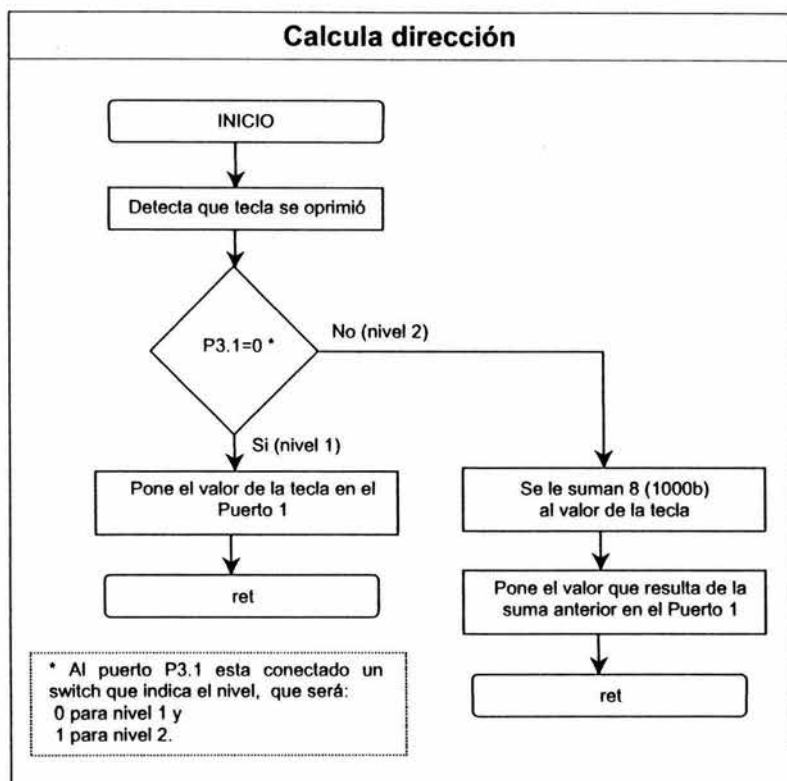
Figura E-2 Subrutinas de programa

\*La bandera FAUX.3 se enciende cuando se genera la interrupción externa 0, al oprimirse alguna de las 8 teclas. (ver subrutina de interrupción INT0)

\*\* La bandera FAUX.1 se enciende cuando se genera la interrupción del timer 0, cada 125 $\mu$ seg. (ver subrutina de interrupción T0)

\*\*\* Al puerto P3.0 esta conectado un switch que puede estar en 0 para grabar o en 1 para reproducir.

\*\*\*\* La bandera FAUX.2 se enciende cuando se genera la interrupción externa 1, cada vez que el convertidor analógico-digital genera un "fin de conversión" (EOC).



**Figura E-3 Subrutina de programa**

En las figuras E-2 y E-3 se presentan los diagramas de flujo de las subrutinas de programa a las cuales se hace referencia en el diagrama de flujo del programa principal en la figura E-1.

La subrutina de interrupción T0 se genera cada vez que hay un sobre flujo del Timer 0 (T0), esto es cada 125µseg. Para que esto suceda, en la configuración del hardware debe establecerse un valor de precarga de 131d, de no ser así la interrupción se generaría cada 256useg.

La subrutina de interrupción INT0 se genera cada vez que alguna de las 8 teclas es oprimida.

La subrutina de interrupción INT1 se genera cada vez que se genere un fin de conversión (EOC) del convertidor analógico-digital.

La figura E-4 muestra los diagramas de flujo de tres subrutinas de interrupción, en todas ellas se utiliza un bit de registro FAUX como bandera que enciende (se pone en 1) cada vez que la interrupción es generada.

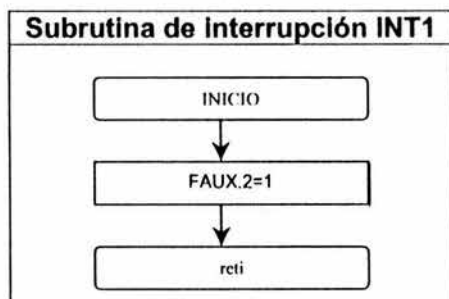
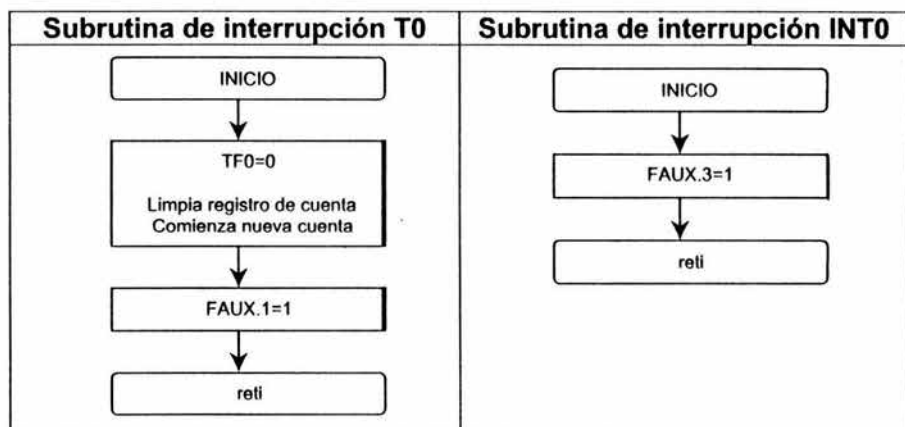


Figura E-4 Subrutinas de Interrupción

En las figuras E-3 y E-5 se puede ver que en el nivel 1 el valor de la tecla se pone en el Puerto 1 directamente, y en el nivel 2 al valor de la tecla se le debe sumar el número ocho binario (1000) para continuar la cuenta hasta el número 1111 binario (15 decimal) y así poder paginar la memoria Flash para guardar los 16 mensajes de 3 segundos cada uno (2 niveles de 8 imágenes).

| Valor de la Tecla | Nivel (P3.1) | Valor que va al Puerto 1 |     |     |
|-------------------|--------------|--------------------------|-----|-----|
|                   |              | bin                      | hex | dec |
| 0000              | 1            | 0000                     | 0   | 0   |
| 0001              | 1            | 0001                     | 1   | 1   |
| 0010              | 1            | 0010                     | 2   | 2   |
| 0011              | 1            | 0011                     | 3   | 3   |
| 0100              | 1            | 0100                     | 4   | 4   |
| 0101              | 1            | 0101                     | 5   | 5   |
| 0110              | 1            | 0110                     | 6   | 6   |
| 0111              | 1            | 0111                     | 7   | 7   |
| 0000              | 2            | 1000                     | 8   | 8   |
| 0001              | 2            | 1001                     | 9   | 9   |
| 0010              | 2            | 1010                     | A   | 10  |
| 0011              | 2            | 1011                     | B   | 11  |
| 0100              | 2            | 1100                     | C   | 12  |
| 0101              | 2            | 1101                     | D   | 13  |
| 0110              | 2            | 1110                     | E   | 14  |
| 0111              | 2            | 1111                     | F   | 15  |

**Figura E-5** Tabla de valores de tecla con relación al nivel



## APÉNDICE F

---

---

### CIRCUITO ESQUEMÁTICO DEL SISTEMA BÁSICO

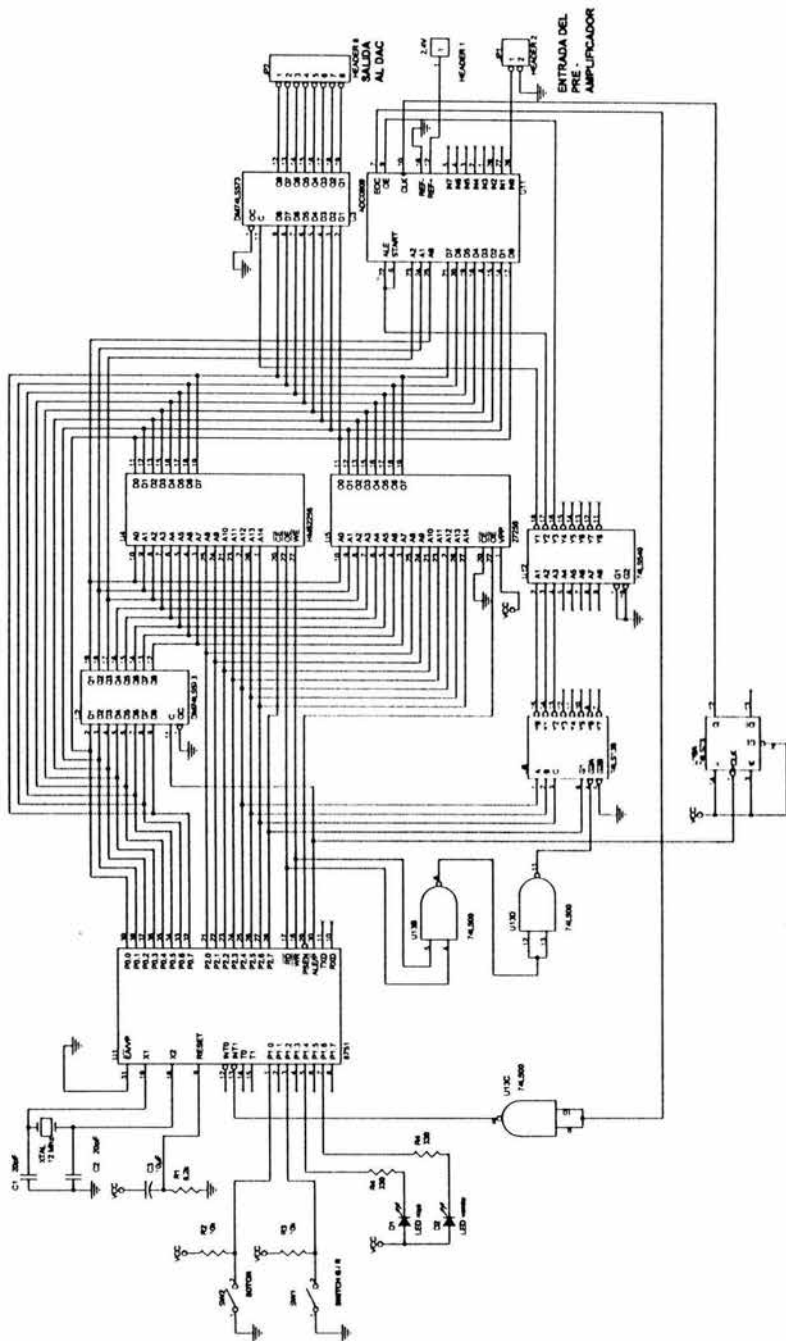


Figura F-1 Circuito esquemático del Sistema Básico

## **APÉNDICE G**

---

---

### **DIAGRAMAS DE FLUJO DEL SISTEMA BÁSICO**

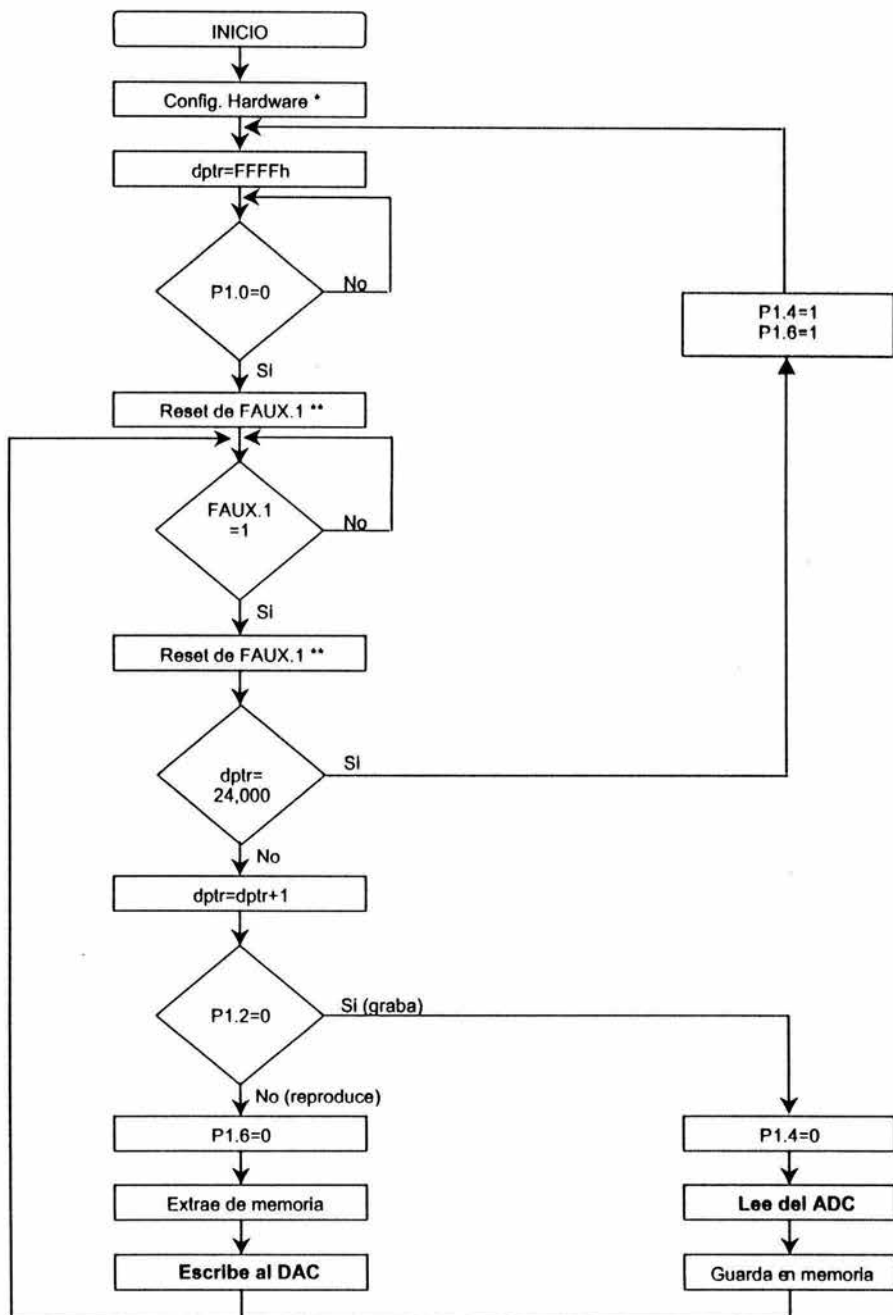


Figura G-1 Diagrama de Flujo del Sistema Básico

\* En la configuración del hardware:

1. Se habilitan los bits 1,2 y 7 del registro IE.
  - IE.1 Que habilitará la interrupción de sobre flujo del Timer 0 (T0)
  - IE.2 Que habilitará la interrupción externa 1 (INT1) para el fin de conversión (EOC) del convertidor analógico-digital.
  - IE.7 Que necesita estar en 1 para que el microprocesador reaccione ante las interrupciones anteriores (habilitación de interrupciones de sobre flujo de T0 e INT1)
2. Se habilita el bit 1 de registro IP
  - IP.1 Que habilita la interrupción del T0 como nivel prioritario
3. Se habilitan los bits 2 y 4 de registro TCON
  - TCON.2 Es el bit de control de la INT1 e indica si la interrupción será considerada siempre y cuando esté en nivel bajo o simplemente que haya habido una detección de flanco descendente
  - TCON.4 Controla el arranque del T0

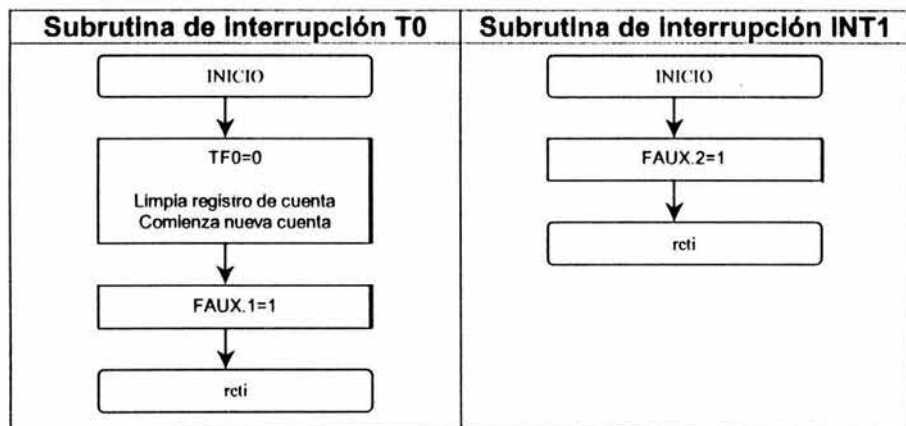
4. Se habilita el TMOD en modo de auto recarga de 8 bits de control interno para el T0 (02H). De tal forma que cada vez que se genere la interrupción comience nuevamente. En este caso, se generará la interrupción cada 125µseg, por este motivo es necesario indicarle un valor de recarga ya que de no ser así la interrupción se generaría cada 256useg (8 bits). El valor de recarga se indica en el registro TH0, para este caso 083H, por que:

$$256 - 125 = 131, \quad 131 \text{ decimal} = 83 \text{ hexadecimal}$$

\*\* FAUX es una bandera auxiliar, un registro direccionable bit a bit (localidad 2FH de la memoria de datos interna). Se utilizan 2 bits de este registro como banderas. Una de ellas nos indica si la interrupción del T0 se generó (FAUX.1) y la otra nos indica si la INT1 se generó (FAUX.2). Reset de FAUX.1 "limpia" el bit, poniéndolo en cero, de este modo en cuanto se genere la interrupción FAUX.1 se pondrá en uno justo cuando se limpie el registro de cuenta del timer y reinicie el conteo. Así se logra que se generen siempre muestras en tiempos iguales. FAUX.2 es uno cuando se genera la INT1 que es el fin de conversión del ADC (EOC End Of Conversion), esto es cuando termina de convertir un dato analógico en digital y tiene listo el valor o muestra.

En puerto el P1.0 esta conectado el botón (la imagen), en P1.2 el switch de selección Grabar o Reproducir, en P1.4 y P1.6 están conectados al led rojo y verde respectivamente. Así, mientras se esta grabando, se enciende el led rojo y mientras se esta reproduciendo, se enciende el led verde.

La figura G-2 muestra los diagramas de flujo de las dos subrutinas de interrupción utilizadas en el sistema básico. Como se puede ver, en ambas se utiliza un bit de registro FAUX como bandera que enciende (se pone en 1) cada vez que la interrupción es generada.



**Figura G-2 Subrutinas de Interrupción**

La figura G-3 presentan los diagramas de flujo de las subrutinas de programación a las cuales se hace referencia en el diagrama de flujo del programa principal en la figura G-1.

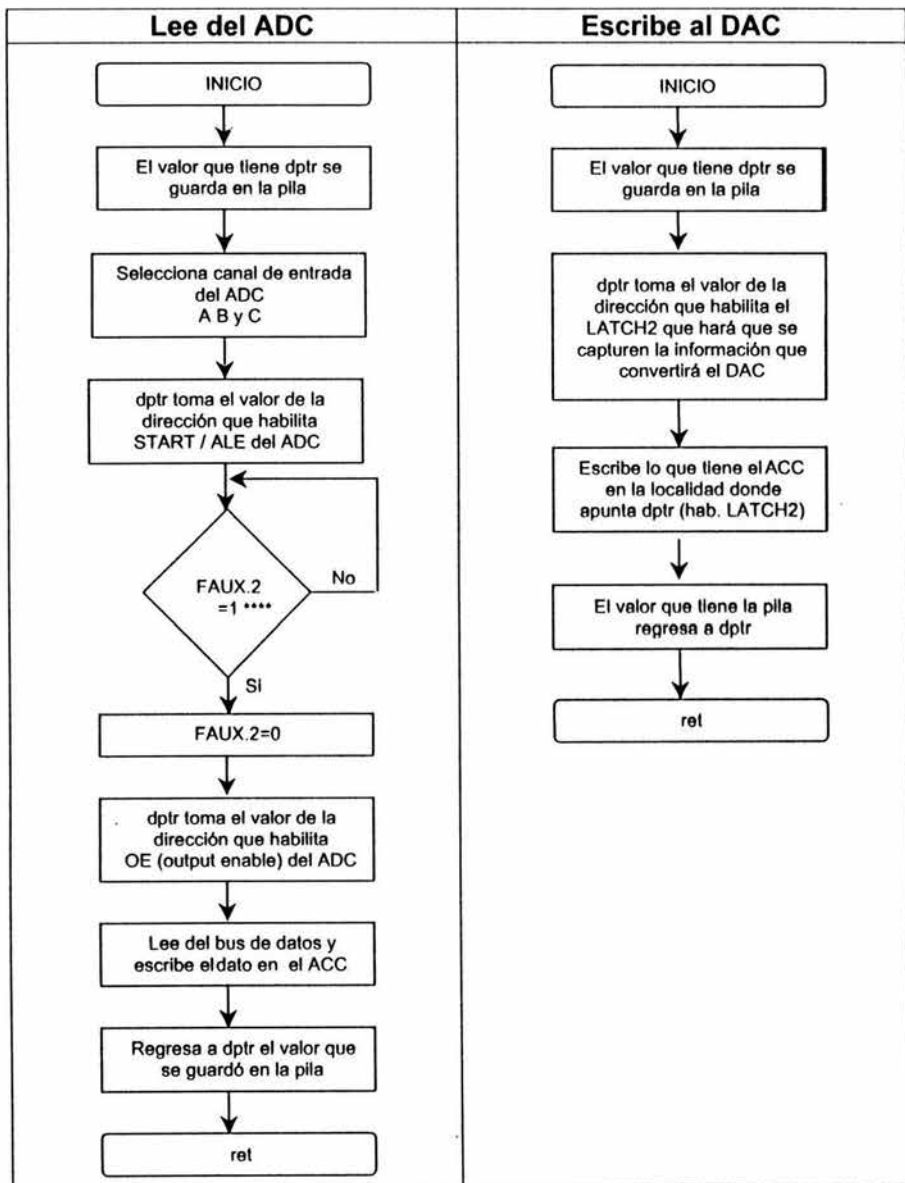


Figura G-3 Subrutinas de programa



## **APÉNDICE H**

---

### **RUTINA DE PROGRAMACIÓN PARA EL SISTEMA BÁSICO**

defseg codigo, absolute

seg codigo

.....declaración de puertos

LATCH: equ 0C000H

OE: equ 09000H

ALE: equ 0A000H

.....declaración de variables

FAUX: equ 2FH ;localidad de memoria direccionable bit a bit  
;que se le asigna el valor de la variable FAUX  
org 00h ;reset de la memoria de programa  
ajmp inicio ;manda al inicio del programa principal  
  
org 0Bh ;vector de interrupción T0 (timer 0)  
clr TF0 ;limpia el registro de cuenta y vuelve a comenzar  
setb FAUX.1 ;se activa poniéndose en 1 la bandera FAUX.1 que  
;indica que se ha generado un sobreflujo del T0  
reti ;regreso de subrutina de interrupción T0  
  
org 13h ;vector de interrupción INT1 (EOC del ADC)  
setb FAUX.2 ;se activa poniéndose en 1 la bandera FAUX.2 que  
;indica que el ADC tiene una muestra.  
reti ;regreso de subrutina de interrupción INT1  
  
org 30h ;a partir de aquí comienza a guardar el código

.....inicia programa principal

inicio: ;inicia el programa principal  
mov FAUX,#00h ;limpia el registro FAUX utilizado como banderas de interrupción  
mov IE,#10000110b ;habilita bits 1(T0), 2(INT1) y 7(int en gen.) del registro IE  
mov IP,#00000010b ;habilita bit 1(T0 como prioridad) del registro IP  
setb tcon.2 ;habilita bit de control de la INT1  
mov TMOD,#02h ;habilita timer interno 8bit-auto-recarga  
mov TH0,#083h ;PRECARGA, registro de cuenta (muestras cada 125useg)  
setb tcon.4 ;RUN, habilita bit de arranque del timer 0  
regres: setb p1.4 ;apaga led de grabación  
setb p1.6 ;apaga led de reproducción  
mov dptr,#0FFFFh ;pone en 1 los 16 bits del registro dptr  
again: jb p1.0,again ;espera el botonazo, en pin 1 del micro  
clr FAUX.1 ;borra la bandera que indica que la interrupción T0 fue generada  
muest: jnb FAUX.1,muest ;espera a que se genere el siguiente T0  
clr FAUX.1 ;borra la bandera que indica que la interrupción T0 fue generada  
mov r7,dph ;guarda el valor de r7 en dph  
mov r6,dpl ;guarda el valor de r6 en dpl  
cjne r7,#05Dh,confir ;p.alta, compara hasta llegar a 24000d=5DC0h  
cjne r6,#0C0h,confir ;p.baja, compara hasta llegar a  
ajmp regres ;regresa al inicio  
  
confir: inc dptr ;incrementa en uno el registro dptr  
jb p1.2,repo ;si es cero graba(sigue),sino brinca a repro  
clr p1.4 ;se enciende el led de grabar, rojo  
acall lecladc ;llama a subrutina que lee dato del ADC  
movx @dptr,a ;escribe en memoria de datos externa  
;lo que tiene el acc en donde apunta dptr  
clr FAUX.1 ;borra la bandera que indica que la interrupción T0 fue generada

```

    ajmp muest          ;brinca a muest, para repetir el proceso

repro:  clr p1.6        ;se enciende el led de reproducir, verde
        movx a,@dptr   ;lee de memoria de datos externa
                        ;el acc toma el valor de donde apunta dptr
        acall escdac   ;llama a subrutina que escribe dato al DAC
        clr FAUX.1     ;borra la bandera que indica que la interrupción T0 fue generada
        ajmp muest     ;brinca a muest, para repetir el proceso

;*****termina programa principal

;*****inicia subrutina que lee dato del ADC

lectadc: push dph      ;guarda en la pila el valor de dph
         push dpl      ;guarda en la pila el valor de dpl
         mov dptr,#ALE ;habilita START-ALE (0F000h)
         mov dpl,#00h  ;la señal entrara por IN0 del ADC
         movx @dptr,a  ;escribe en memoria de datos externa
                        ;lo que tiene el acc en donde apunta dptr

eadc:   jnb FAUX.2,eadc ;espera a que se genere la INT1
        clr FAUX.2     ;borra la bandera que indica que la INT1 fue generada
        mov dptr,#OE  ;habilita OE del ADC (0D000h)
        movx a,@dptr  ;lee de memoria de datos externa
                        ;el acc toma el valor de donde apunta dptr
        pop dpl       ;regresa de la pila el valor que tenia dpl
        pop dph       ;regresa de la pila el valor que tenia dph
        ret           ;regreso de subrutina lectadc

;*****termina subrutina que lee dato del ADC

;*****inicia subrutina que escribe dato al DAC

escdac: push dph      ;guarda en la pila el valor de dph
         push dpl      ;guarda en la pila el valor de dpl
         mov dptr,#LATCH ;habilita LATCH (conectado al DAC),(0A00h)
         movx @dptr,a  ;escribe en memoria de datos externa
                        ;lo que tiene el acc en donde apunta dptr
        pop dpl       ;regresa de la pila el valor que tenia dpl
        pop dph       ;regresa de la pila el valor que tenia dph
        ret           ;regreso de subrutina lectadc

;*****termina subrutina que escribe dato al DAC

end          ;fin del programa

```

**Figura H-1 Rutina de programación para el Sistema Básico**

# TABLA DE FIGURAS

---

---

## Capítulo 1

|            |  |   |
|------------|--|---|
| Figura 1-1 | Metodología para el desarrollo del intérprete gráfico----- | 7 |
|------------|--|---|

## Capítulo 2

|             |   |    |
|-------------|---|----|
| Figura 2-1  | Ejemplos de productos de CA Mayer-Johnson -----             | 10 |
| Figura 2-2  | Ejemplos de productos de CA Enabling Devices-----           | 11 |
| Figura 2-3  | Sección de un circuito integrado-----                       | 13 |
| Figura 2-4  | Niveles de integración para dispositivos digitales -----    | 14 |
| Figura 2-5  | Clasificación de CI según la tecnología de fabricación ---- | 16 |
| Figura 2-6  | Estructura de la CPU -----                                  | 20 |
| Figura 2-7  | Organización de una computadora -----                       | 21 |
| Figura 2-8  | Características de los dispositivos de la familia MCS-51 -- | 22 |
| Figura 2-9  | Esquema a bloques del 8751H-----                            | 23 |
| Figura 2-10 | Diagrama de Conexión del MCS-51 -----                       | 23 |
| Figura 2-11 | Diagrama a Bloques de la esencia del 8051 -----             | 24 |
| Figura 2-12 | Relación de áreas de almacenamiento (8051)-----             | 25 |
| Figura 2-13 | Memoria de Programa (8051)-----                             | 26 |
| Figura 2-14 | Mapa de memoria de datos interna (8051) -----               | 27 |
| Figura 2-15 | Registros de función especial -----                         | 36 |
| Figura 2-16 | Características de los SFR-----                             | 37 |
| Figura 2-17 | Características de las memorias semiconductoras -----       | 46 |
| Figura 2-18 | Tipos de memoria -----                                      | 50 |
| Figura 2-19 | Características de los principales tipos de memoria. -----  | 51 |

## Capítulo 3

|             |  |    |
|-------------|--|----|
| Figura 3-1  | Diagrama a bloques del sistema-----                    | 56 |
| Figura 3-2  | Teclado de 8 botones -----                             | 57 |
| Figura 3-3  | Diagrama de interconexión del sistema-----             | 60 |
| Figura 3-4  | Circuito de reloj para el microcontrolador -----       | 64 |
| Figura 3-5  | Circuito de RESET automático al encendido del módulo-- | 65 |
| Figura 3-6  | Mapa de memoria y dispositivos externos -----          | 67 |
| Figura 3-7  | Cuadro de E/S del GAL16V8 -----                        | 68 |
| Figura 3-8  | Diagrama de E/S del GAL16V8-----                       | 69 |
| Figura 3-9  | Ecuaciones de E/S del GAL16V8 -----                    | 69 |
| Figura 3-10 | Diagrama de la señal de 1MHz del CLK del ADC-----      | 72 |
| Figura 3-11 | Tabla de verdad del flip-flop JK 74LS73-----           | 73 |

|             |  |    |
|-------------|--|----|
| Figura 3-12 | Secuencia de estados que explica el CLK del ADC      | 74 |
| Figura 3-13 | Diagrama eléctrico del convertidor digital-analógico | 75 |
| Figura 3-14 | Diagrama salida de baja impedancia positiva          | 76 |
| Figura 3-15 | Controlador de teclado 74C922                        | 77 |
| Figura 3-16 | Tabla de función del 74LS573                         | 82 |
| Figura 3-17 | Pantalla de cristal líquido AND 491                  | 83 |
| Figura 3-18 | Conexión del Módulo Principal                        | 84 |

#### Capítulo 4

|            |  |    |
|------------|--|----|
| Figura 4-1 | Diagrama del sistema básico                              | 87 |
| Figura 4-2 | Tabla de verdad del decodificador 74LS138                | 88 |
| Figura 4-3 | 16 Bits de direcciones en la habilitación de periféricos | 89 |
| Figura 4-4 | Tabla de verdad del buffer octal (SN74LS540)             | 90 |
| Figura 4-5 | Diagrama eléctrico 15V - 2.4V                            | 95 |

#### Apéndices

|            |   |     |
|------------|---|-----|
| Figura A-1 | Diagrama eléctrico y lista de partes del pre-amplificador         | 108 |
| Figura B-1 | Diagrama eléctrico y lista de partes del amplificador             | 110 |
| Figura C-1 | Diagrama eléctrico y lista de partes de la fuente de alimentación | 112 |
| Figura D-1 | Circuito esquemático del Intérprete Gráfico                       | 114 |
| Figura E-1 | Diagrama de Flujo del Intérprete Gráfico                          | 116 |
| Figura E-2 | Subrutinas de programa  | 117 |
| Figura E-3 | Subrutina de programa   | 119 |
| Figura E-4 | Subrutinas de interrupción  | 120 |
| Figura E-5 | Tabla de valores de tecla con relación al nivel                   | 121 |
| Figura F-1 | Circuito esquemático del Sistema Básico                           | 123 |
| Figura G-1 | Diagrama de Flujo del Sistema Básico                              | 125 |
| Figura G-2 | Subrutinas de interrupción  | 128 |
| Figura G-3 | Subrutinas de programa  | 129 |
| Figura H-1 | Rutina de programación para el Sistema Básico                     | 132 |